

# **Applying Data Mining Techniques Using Enterprise Miner<sup>TM</sup>**

**Course Notes**

*Applying Data Mining Techniques Using Enterprise Miner™ Course Notes* was developed by Sue Walsh. Some of the course notes is based on material developed by Will Potts and Doug Wielenga. Additional contributions were made by John Amrhein, Kate Brown, Iris Krammer, and Bob Lucas. Editing and production support was provided by the Curriculum Development and Support Department.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.

### **Applying Data Mining Techniques Using Enterprise Miner™ Course Notes**

Copyright © 2002 by SAS Institute Inc., Cary, NC 27513, USA. All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher,  
SAS Institute Inc.

---

Book code 58801, course code ADMT, prepared date 05Apr02.

## Table of Contents

Course Description .....	v
Prerequisites .....	vi
General Conventions .....	vii
<b>Chapter 1      Introduction to Data Mining.....</b>	<b>1-1</b>
1.1    Background.....	1-3
1.2    SEMMA.....	1-15
<b>Chapter 2      Predictive Modeling Using Decision Trees .....</b>	<b>2-1</b>
2.1    Introduction to Enterprise Miner .....	2-3
2.2    Modeling Issues and Data Difficulties.....	2-20
2.3    Introduction to Decision Trees.....	2-37
2.4    Building and Interpreting Decision Trees .....	2-46
<b>Chapter 3      Predictive Modeling Using Regression.....</b>	<b>3-1</b>
3.1    Introduction to Regression.....	3-3
3.2    Regression in Enterprise Miner .....	3-8
<b>Chapter 4      Variable Selection.....</b>	<b>4-1</b>
4.1    Variable Selection and Enterprise Miner .....	4-3
<b>Chapter 5      Predictive Modeling Using Neural Networks .....</b>	<b>5-1</b>
5.1    Introduction to Neural Networks .....	5-3
5.2    Visualizing Neural Networks .....	5-9

<b>Chapter 6 Model Evaluation and Implementation .....</b>	<b>6-1</b>
6.1 Model Evaluation: Comparing Candidate Models.....	6-3
6.2 Ensemble Models.....	6-10
6.3 Model Implementation: Generating and Using Score Code .....	6-16
<b>Chapter 7 Cluster Analysis .....</b>	<b>7-1</b>
7.1 K-Means Cluster Analysis.....	7-3
7.2 Self-Organizing Maps.....	7-24
<b>Chapter 8 Association and Sequence Analysis .....</b>	<b>8-1</b>
8.1 Introduction to Association Analysis .....	8-3
8.2 Interpretation of Association and Sequence Analysis .....	8-7
8.3 Dissociation Analysis (Self-Study) .....	8-24
<b>Appendix A References .....</b>	<b>A-1</b>
A.1 References.....	A-3
<b>Appendix B Index .....</b>	<b>B-1</b>

## Course Description

This course provides extensive hands-on experience with Enterprise Miner and covers the basic skills required to assemble analyses using the rich tool set of Enterprise Miner. It also covers concepts fundamental to understanding and successfully applying data mining methods.

After completing this course, you should be able to

- identify business problems and determine suitable analytical methods
- understand the difficulties presented by massive, opportunistic data
- assemble analysis-flow diagrams
- prepare data for analysis, including partitioning data and imputing missing values
- train, assess, and compare regression models, neural networks, and decision trees
- perform cluster analysis
- perform association and sequence analysis.

### To learn more...



---

#### SAS Education

A full curriculum of general and statistical instructor-based training is available at any of the Institute's training facilities. Institute instructors can also provide on-site training.

For information on other courses in the curriculum, contact the SAS Education Division at 1-919-531-7321, or send e-mail to [training@sas.com](mailto:training@sas.com). You can also find this information on the Web at [www.sas.com/training/](http://www.sas.com/training/) as well as in the SAS Training Course Catalog.



---

#### SAS Publishing

For a list of other SAS books that relate to the topics covered in this Course Notes, USA customers can contact our SAS Publishing Department at 1-800-727-3228 or send e-mail to [sasbook@sas.com](mailto:sasbook@sas.com). Customers outside the USA, please contact your local SAS office.

Also, see the Publications Catalog on the Web at [www.sas.com/pubs](http://www.sas.com/pubs) for a complete list of books and a convenient order form.

## Prerequisites

Before selecting this course, you should be familiar with Microsoft Windows and Windows-based software. No previous SAS software experience is necessary.

## General Conventions

This section explains the various conventions used in presenting text, SAS language syntax, and examples in this book.

### Typographical Conventions

You will see several type styles in this book. This list explains the meaning of each style:

UPPERCASE ROMAN	is used for SAS statements, variable names, and other SAS language elements when they appear in the text.
<i>italic</i>	identifies terms or concepts that are defined in text. Italic is also used for book titles when they are referenced in text, as well as for various syntax and mathematical elements.
<b>bold</b>	is used for emphasis within text.
monospace	is used for examples of SAS programming statements and for SAS character strings. Monospace is also used to refer to field names in windows, information in fields, and user-supplied information.
<u>select</u>	indicates selectable items in windows and menus. This book also uses icons to represent selectable items.

### Syntax Conventions

The general forms of SAS statements and commands shown in this book include only that part of the syntax actually taught in the course. For complete syntax, see the appropriate SAS reference guide.

```
PROC CHART DATA= SAS-data-set;  
      HBAR | VBAR chart-variables </ options>;  
      RUN;
```

This is an example of how SAS syntax is shown in text:

- **PROC** and **CHART** are in uppercase bold because they are SAS keywords.
- **DATA=** is in uppercase to indicate that it must be spelled as shown.
- *SAS-data-set* is in italic because it represents a value that you supply. In this case, the value must be the name of a SAS data set.
- **HBAR** and **VBAR** are in uppercase bold because they are SAS keywords. They are separated by a vertical bar to indicate they are mutually exclusive; you can choose one or the other.
- *chart-variables* is in italic because it represents a value or values that you supply.
- </ *options*> represents optional syntax specific to the HBAR and VBAR statements. The angle brackets enclose the slash as well as *options* because if no options are specified you do not include the slash.
- **RUN** is in uppercase bold because it is a SAS keyword.



# Chapter 1 Introduction to Data Mining

1.1	Background.....	1-3
1.2	SEMMA .....	1-15



## 1.1 Background

### Objectives

- Discuss some of the history of data mining.
- Define data mining and its uses.

2

### Defining Characteristics

#### 1. The Data

- Massive, operational, and opportunistic

#### 2. The Users and Sponsors

- Business decision support

#### 3. The Methodology

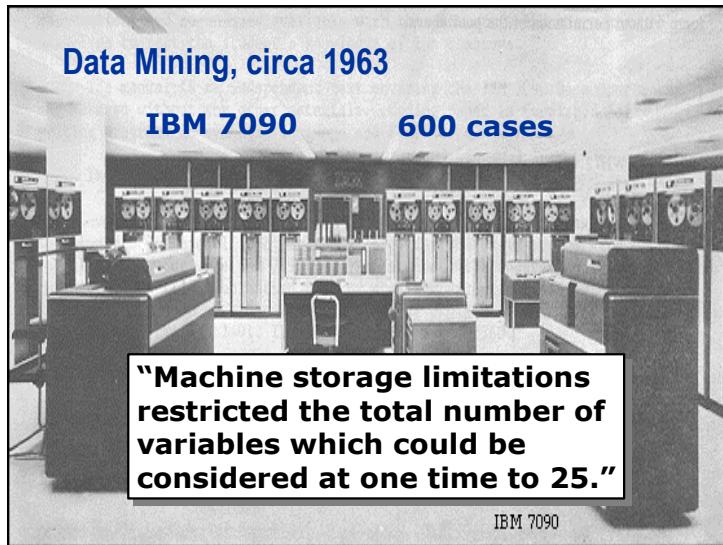
- Computer-intensive “ad hockery”
- Multidisciplinary lineage

3

SAS defines data mining as

*advanced methods for exploring and modeling relationships in large amounts of data.*

There are other similar definitions. However, exploring and modeling relationships in data has a much longer history than the term *data mining*.



Data mining analysis was limited by the computing power of the time. The IBM 7090 is a transistorized mainframe introduced in 1959. It cost approximately 3 million dollars. It had a processor speed of approximately 0.5 MHz and roughly 0.2 MB of RAM using ferrite magnetic cores. Data sets were stored on punch cards and then transferred to magnetic tape using separate equipment. A data set with 600 rows and 4 columns would have used approximately 3,000 cards. Tape storage was limited by the size of the room. The room pictured above contains the tape drives and controllers for the IBM 7090. The computer itself would need a larger room.

## Since 1963

### Moore's Law:

The information density on silicon-integrated circuits doubles every 18 to 24 months.

### Parkinson's Law:

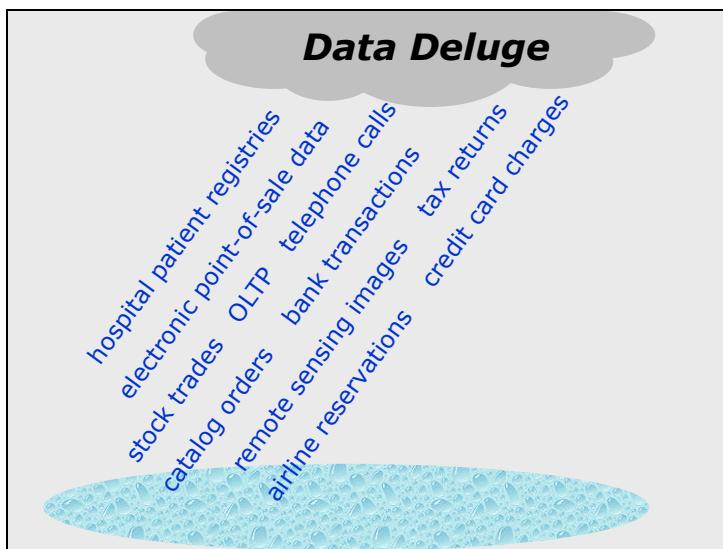
Work expands to fill the time available for its completion.

5

Computer performance has been doubling every 18 to 24 months (Gordon Moore, co-founder of Intel, 1965). This has led to technological advances in storage structures and a corresponding increase in MB of storage space per dollar. Parkinson's law of data, a corollary of Parkinson's law (Cyril Northcote Parkinson, 1955), states that

*Data expands to fill the space available for storage.*

In fact, the amount of data in the world has been doubling every 18 to 24 months. Multi-gigabyte commercial databases are now commonplace.



The data deluge is the result of the prevalence of automatic data collection, electronic instrumentation, and online transactional processing (OLTP). There is a growing recognition of the untapped value in these databases. This recognition is driving the development of data mining and data warehousing.

## The Data

	<u>Experimental</u>	<u>Opportunistic</u>
<b>Purpose</b>	Research	Operational
<b>Value</b>	Scientific	Commercial
<b>Generation</b>	Actively controlled	Passively observed
<b>Size</b>	Small	Massive
<b>Hygiene</b>	Clean	Dirty
<b>State</b>	Static	Dynamic

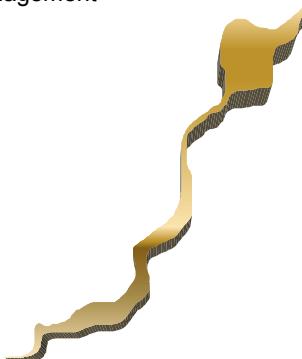
7

Historically, most data was generated or collected for research purposes. Today, businesses have massive amounts of operational data. This operational data was not generated with data analysis in mind. It is aptly characterized as *opportunistic* (Huber 1997). This is in contrast to experimental data where factors are controlled and varied in order to answer specific questions.

## Business Decision Support

- Database Marketing
  - Target marketing
  - Customer relationship management
- Credit Risk Management
  - Credit scoring
- Fraud Detection
- Healthcare Informatics
  - Clinical decision support

8



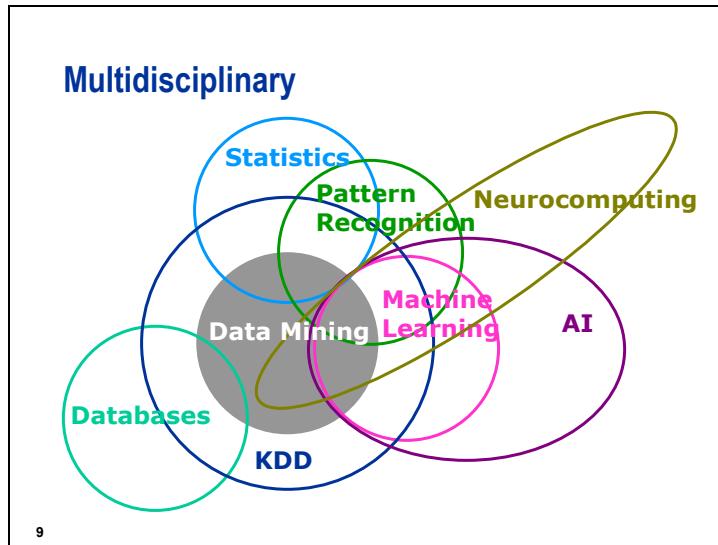
The owners of the data and sponsors of the analyses are typically not researchers. The objectives are usually to support crucial business decisions.

Database marketing makes use of customer and transaction databases to improve product introduction, cross-sell, trade-up, and customer loyalty promotions. In *target marketing*, segments of customers that are most likely to respond to an offer are identified so that campaign efforts can be focused on that group. One of the facets of *customer relationship management* is concerned with identifying and profiling customers who are likely to switch brands or cancel services (*churn*). These customers can then be targeted for loyalty promotions.

*Credit scoring* (Rosenberg and Gleidt 1994, Hand and Henley 1997) is chiefly concerned with whether to extend credit to an applicant. The aim is to anticipate and reduce defaults and serious delinquencies. Other credit risk management concerns are the maintenance of existing credit lines (should the credit limit be raised?) and determining the best action to be taken on delinquent accounts.

The aim of *fraud detection* is to uncover the patterns that characterize deliberate deception. These patterns are used by banks to prevent fraudulent credit card transactions and bad checks, by telecommunication companies to prevent fraudulent calling card transactions, and by insurance companies to identify fictitious or abusive claims.

*Healthcare informatics* (medical informatics) is concerned with management and analysis of the computer-based patient record (CPR). Decision-support systems relate clinical information to patient outcomes. Practitioners and healthcare administrators use the information to improve the quality and cost effectiveness of different therapies and practices.



The analytical tools used in data mining were developed mainly by statisticians, artificial intelligence (AI) researchers, and database system researchers.

*KDD* (knowledge discovery in databases) is a newly formed (1989), multidisciplinary research area concerned with the extraction of patterns from large databases. KDD is often used synonymously with data mining. More precisely, data mining is considered a single step in the overall discovery process.

Machine learning is a branch of AI concerned with creating and understanding semiautomatic learning methods.

Pattern recognition has its roots in engineering and is typically concerned with image classification. Pattern recognition methodology crosses over many areas.

Neurocomputing is, itself, a multidisciplinary field concerned with neural networks.

## Tower of Babel

“Bias”

STATISTICS: the expected difference between an estimator and what is being estimated



NEUROCOMPUTING: the constant term in a linear combination

MACHINE LEARNING: a reason for favoring any model that does not fit the data perfectly

10

One consequence of the multidisciplinary lineage of data mining methods is confusing terminology. The same terms are often used in different senses, and synonyms abound.

## Steps in Data Mining/Analysis

### 1. Specific Objectives

- In terms of the subject matter

### 2. Translation into Analytical Methods

### 3. Data Examination

- Data capacity
- Preliminary results

### 4. Refinement and Reformulation

11

Problem formulation is central to successful data mining. The following are examples of objectives that are inadequately specified:

*Understand our customer base.*

*Re-engineer our customer retention strategy.*

*Detect actionable patterns.*

Objectives such as these leave many essential questions unanswered. For example, what specific actions will result from the analytical effort? The answer, of course, may depend on the result, but the inability to speculate is an indication of inadequate problem formulation. Unless the purpose of the analysis is to write a research paper, “understanding” is probably not the ultimate goal.

A related pitfall is to specify the objectives in terms of analytical methods:

*Implement neural networks.*

*Apply visualization tools.*

*Cluster the database.*

The same analytical tools may be applied (or misapplied) to many different problems. The choice of the most appropriate analytical tool often depends on subtle differences in the objectives. The objectives eventually must be translated in terms of analytical methods. This should occur only after they are specified in ordinary language.

## Required Expertise

- Domain
- Data
- Analytical Methods



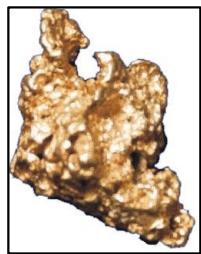
12

- The *domain expert* understands the particulars of the business or scientific problem; the relevant background knowledge, context, and terminology; and the strengths and deficiencies of the current solution (if a current solution exists).
- The *data expert* understands the structure, size, and format of the data.
- The *analytical expert* understands the capabilities and limitations of the methods that may be relevant to the problem.

The embodiment of this expertise might take one, two, three, or more people.

## Nuggets

**"If you've got terabytes of data, and  
you're relying on  
data mining to find  
interesting things  
in there for you,  
you've lost before  
you've even begun."**



— Herb Edelstein

13

The passage continues (Beck 1997):

*...You really need people who understand what it is they are looking for – and what they can do with it once they find it.*

Many people think data mining means magically discovering hidden nuggets of information without having to formulate the problem and without regard to the structure or content of the data. This is an unfortunate misconception.

## What Is Data Mining?

### ■ IT

- Complicated database queries

### ■ ML

- Inductive learning from examples

### ■ Stat

- What we were taught not to do

14

The database community has a tendency to view data mining methods as more complicated types of database queries. For example, standard query tools can answer questions such as “how many surgeries resulted in hospital stays longer than 10 days?” But data mining is needed for more complicated queries such as “what are the important preoperative predictors of excessive length of stay?” This view has led many to confuse data mining with query tools. For example, many consider OLAP (online analytical processing), which is software for interactive access, query, and summarization of multidimensional data warehouses, to be data mining. To some, the objective of data mining is merely to implement query tools. In this case, there is no specific problem to formulate, and sophisticated analytical methods are not relevant.

## Problem Translation

### ■ Predictive Modeling

- Supervised classification

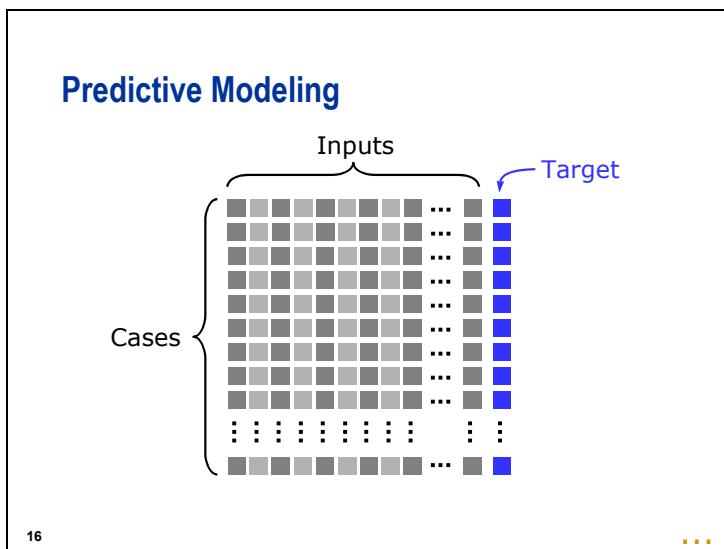
### ■ Cluster Analysis

### ■ Association Rules

### ■ Something Else

15

The problem translation step involves determining what analytical methods (if any) are relevant to the objectives. The requisite knowledge is a wide array of methodologies and what sorts of problems they solve.



*Predictive modeling* (aka supervised prediction, supervised learning) is the fundamental data mining task. The *training data set* consists of *cases* (aka observations, examples, instances, records). Associated with each case is a vector of *input variables* (aka predictors, features, explanatory variables, independent variables) and a *target variable* (aka response, outcome, dependent variable). The training data is used to construct a model (rule) that can predict the values of the target from the inputs.

The task is referred to as *supervised* because the prediction model is constructed from data where the target is known. If the targets are known, why do we need a prediction model? It allows you to predict *new* cases when the target is unknown. Typically, the target is unknown because it refers to a future event. In addition, the target may be difficult, expensive, or destructive to measure.

The measurement scale of the inputs can be varied. The inputs may be numeric variables such as income. They may be nominal variables such as occupation. They are often binary variables such as home ownership.

## Types of Targets

- Supervised Classification
  - Event/no event (binary target)
  - Class label (multiclass problem)
- Regression
  - Continuous outcome
- Survival Analysis
  - Time-to-event (possibly censored)

17

The main differences among the analytical methods for predictive modeling depend on the type of target variable.

In *supervised classification*, the target is a class label (categorical). The training data consists of labeled cases. The aim is to construct a model (classifier) that can allocate cases to the classes using only the values of the inputs.

*Regression analysis* is supervised prediction where the target is a continuous variable. (The term *regression* can also be used more generally; for example, *logistic regression* is a method used for supervised classification.) The aim is to construct a model that can predict the values of the target from the inputs.

In *survival analysis*, the target is the time until some event occurs. The outcome for some cases is censored; all that is known is that the event has not yet occurred. Special methods are usually needed to handle censoring.

## 1.2 SEMMA

### Objectives

- Define SEMMA.
- Introduce the tools available in Enterprise Miner.

**SEMMA**

- Sample
- Explore
- Modify
- Model
- Assess

20

The tools in the Enterprise Miner are arranged according the SAS process for data mining, SEMMA.

SEMMA stands for

**Sample** - identify input data sets (identify input data, sample from a larger data set, partition data set into training, validation, and test data sets).

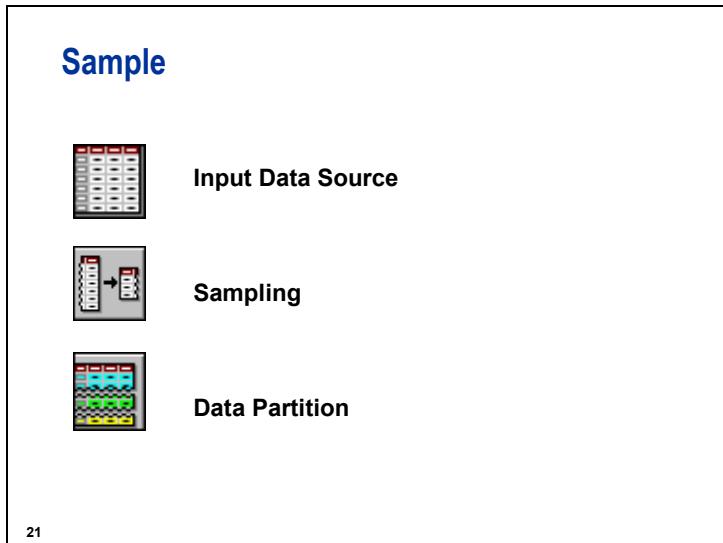
**Explore** - explore data set statistically and graphically (plot the data, obtain descriptive statistics, identify important variables, perform association analysis).

**Modify** - prepare the data for analysis (create additional variables or transform existing variables for analysis, identify outliers, impute missing values, modify the way in which variables are used for the analysis, perform cluster analysis, analyze data with SOMs or Kohonen networks).

**Model** - fit a predictive model (model a target variable using a regression model, a decision tree, a neural network, or a user-defined model).

**Assess** - compare competing predictive models (build charts plotting percentage of respondents, percentage of respondents captured, lift charts, profit charts).

Additional tools are available under the Utilities group.



## Sample Nodes

The **Input Data Source node** reads data sources and defines their attributes for later processing by Enterprise Miner. This node can perform various tasks:

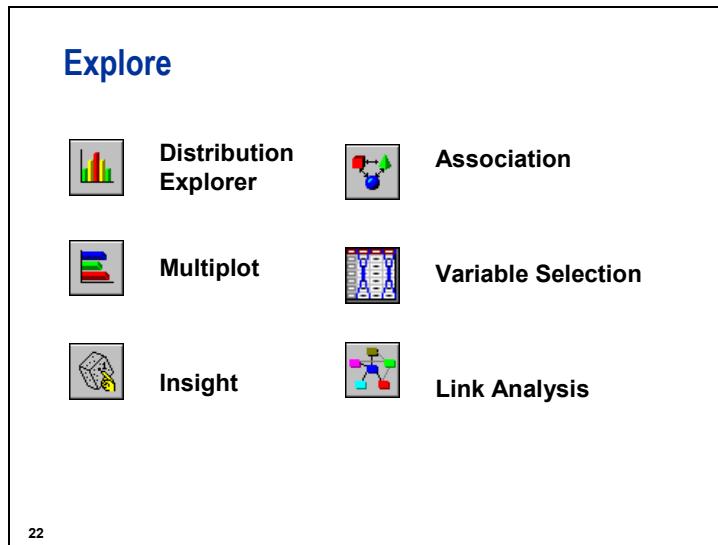
1. It enables you to access SAS data sets and data marts. Data marts can be defined using SAS/Warehouse Administrator software and set up for Enterprise Miner using the Enterprise Miner Warehouse Add-Ins.
2. It automatically creates the metadata sample for each variable in the data set.
3. It sets initial values for the measurement level and the model role for each variable. You can change these values if you are not satisfied with the automatic selections made by the node.
4. It displays summary statistics for interval and class variables.
5. It enables you to define target profiles for each target in the input data set.

 For the purposes of this document, *data sets* and *data tables* are equivalent terms.

The **Sampling node** enables you to take random samples, stratified random samples, and cluster samples of data sets. Sampling is recommended for extremely large databases because it can significantly decrease model training time. If the sample is sufficiently representative, relationships found in the sample can be expected to generalize to the complete data set. The Sampling node writes the sampled observations to an output data set and saves the seed values that are used to generate the random numbers for the samples so that you may replicate the samples.

The **Data Partition node** enables you to partition data sets into training, test, and validation data sets. The training data set is used for preliminary model fitting. The validation data set is used to monitor and tune the model during estimation and is also used for model assessment. The test data set is an additional holdout data set that you

can use for model assessment. This node uses simple random sampling, stratified random sampling, or user-defined partitions to create partitioned data sets.



### Explore Nodes

The **Distribution Explorer node** is a visualization tool that enables you to explore large volumes of data quickly and easily in multidimensional histograms. You can view the distribution of up to three variables at a time with this node. When the variable is binary, nominal, or ordinal, you can select specific values to exclude from the chart. To exclude extreme values for interval variables, you can set a range cutoff. The node also generates summary statistics for the charting variables.

The **Multiplot node** is another visualization tool that enables you to explore larger volumes of data graphically. Unlike the Insight or Distribution Explorer nodes, the Multiplot node automatically creates bar charts and scatter plots for the input and target variables without making several menu or window item selections. The code created by this node can be used to create graphs in a batch environment, whereas the Insight and Distribution Explorer nodes must be run interactively.

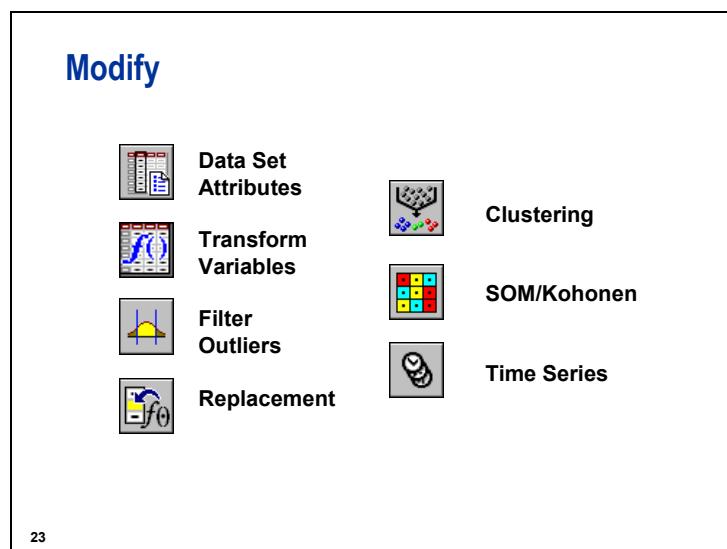
The **Insight node** enables you to open a SAS/INSIGHT session. SAS/INSIGHT software is an interactive tool for data exploration and analysis. With it you explore data through graphs and analyses that are linked across multiple windows. You can analyze univariate distributions, investigate multivariate distributions, and fit explanatory models using generalized linear models.

The **Association node** enables you to identify association relationships within the data. For example, if a customer buys a loaf of bread, how likely is the customer to also buy a gallon of milk? The node also enables you to perform sequence discovery if a time stamp variable (a sequence variable) is present in the data set.

The **Variable Selection node** enables you to evaluate the importance of input variables in predicting or classifying the target variable. To select the important inputs, the node uses either an R-square or a Chi-square selection (tree-based) criterion. The R-square criterion enables you to remove variables in hierarchies, remove variables that have large percentages of missing values, and remove class

variables that are based on the number of unique values. The variables that are not related to the target are set to a status of rejected. Although rejected variables are passed to subsequent nodes in the process flow diagram, these variables are not used as model inputs by a more detailed modeling node, such as the Neural Network and Tree nodes. You can reassign the input model status to rejected variables.

*Link Analysis* is the examination of the linkages between effects in a complex system to discover patterns of activity that can be used to derive useful conclusions. Some applications include forms of fraud detection, criminal network conspiracies, telephone traffic patterns, Web site structure and usage, database visualization, and social network analysis. The **Link Analysis node** transforms data from differing sources into a data model that can be graphed. The data model supports simple statistical measures, presents a simple interactive graph for basic analytical exploration, and generates cluster scores from raw data that can be used for data reduction and segmentation. Graphics from the node show the relationships between variable levels.



## Modify Nodes

The **Data Set Attributes node** enables you to modify data set attributes, such as data set names, descriptions, and roles. You can also use this node to modify the metadata sample that is associated with a data set and specify target profiles for a target. An example of a useful Data Set Attributes application is to generate a data set in the SAS Code node and then modify its metadata sample with this node.

The **Transform Variables node** enables you to transform variables; for example, you can transform variables by taking the square root of a variable, by taking the natural logarithm, maximizing the correlation with the target, or normalizing a variable. Additionally, the node supports user-defined formulas for transformations and provides a visual interface for grouping interval-valued variables into buckets or quantiles. This node also automatically bins interval variables into buckets using an algorithm based on a decision tree. Transforming variables to similar scale and variability may improve the fit of models and, subsequently, the classification and prediction precision of fitted models.

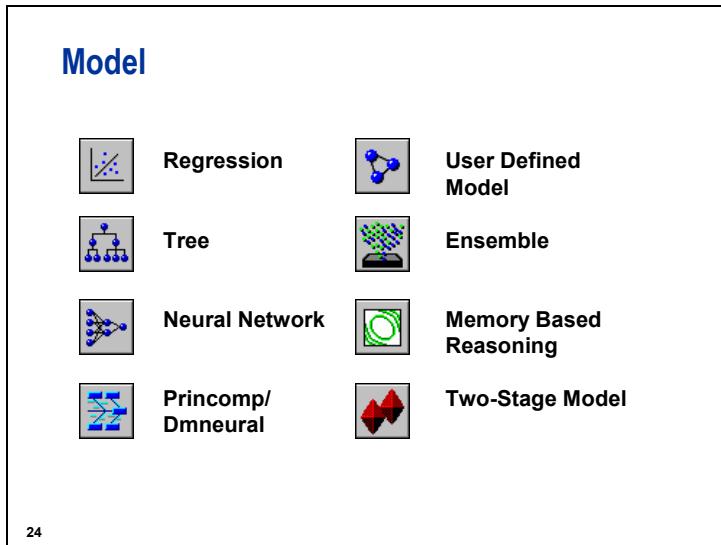
The **Filter Outliers node** enables you to apply a filter to your training data set to exclude observations, such as outliers or other observations, that you do not want to include in your data mining analysis. The node does not filter observations in the validation, test, or score data sets.

The **Replacement node** enables you to impute (fill in) values for observations that have missing values. You can replace missing values for interval variables with the mean, median, midrange, midminimum spacing, distribution-based replacement, or you can use a replacement M-estimator such as Tukey's biweight, Huber's, or Andrew's Wave. You can also estimate the replacement values for each interval input by using a tree-based imputation method. Missing values for class variables can be replaced with the most frequently occurring value, distribution-based replacement, tree-based imputation, or a constant.

The **Clustering node** enables you to segment your data; that is, it enables you to identify data observations that are similar in some way. Observations that are similar tend to be in the same cluster, and observations that are different tend to be in different clusters. The cluster identifier for each observation can be passed to subsequent nodes in the diagram.

The **SOM/Kohonen node** generates self-organizing maps, Kohonen networks, and vector quantization networks. Essentially, the node performs unsupervised learning in which it attempts to learn the structure of the data. As with the Clustering node, after the network maps have been created, the characteristics can be examined graphically using the results browser. The node provides the analysis results in the form of an interactive map that illustrates the characteristics of the clusters. Furthermore, it provides a report indicating the importance of each variable.

The **Time Series node** enables you to understand trends and seasonal variation in your customers' buying patterns. You may have many suppliers and many customers as well as transaction data that is associated with both. The size of each set of transactions may be very large, which makes many traditional data mining tasks difficult. By condensing the information into a time series, you can discover trends and seasonal variations in customer and supplier habits that may not be visible in transactional data. The node converts transactional data to time series data. Transactional data is time-stamped data that is collected over time at no particular frequency. By contrast, time series data is time-stamped data that is summarized over time at a specific frequency.



24

## Model Nodes

The **Regression node** enables you to fit both linear and logistic regression models to your data. You can use continuous, ordinal, and binary target variables. You can use both continuous and discrete variables as inputs. The node supports the stepwise, forward, and backward selection methods. A point-and-click interaction builder enables you to create higher-order modeling terms.

The **Tree node** enables you to perform multiway splitting of your database based on nominal, ordinal, and continuous variables. The node supports both automatic and interactive training. When you run the Tree node in automatic mode, it automatically ranks the input variables based on the strength of their contribution to the tree. This ranking may be used to select variables for use in subsequent modeling. In addition, dummy variables can be generated for use in subsequent modeling. You can override any automatic step with the option to define a splitting rule and prune explicit nodes or subtrees. Interactive training enables you to explore and evaluate a large set of trees as you develop them.

The **Neural Network node** enables you to construct, train, and validate multilayer feed-forward neural networks. In general, each input is fully connected to the first hidden layer, each hidden layer is fully connected to the next hidden layer, and the last hidden layer is fully connected to the output. The Neural Network node supports many variations of this general form.

The **Princomp/Dmneural node** enables you to fit an additive nonlinear model that uses the bucketed principal components as inputs to predict a binary or an interval target variable. The node also performs a principal components analysis and passes the scored principal components to the successor nodes. The target variable must be binary or interval for dmneural network training, but no target variable is required for a principal components analysis.

The **User-Defined Model node** enables you to generate assessment statistics using predicted values from a model that you built with the SAS Code node (for example, a logistic model using the LOGISTIC procedure in SAS/STAT) or the Variable

Selection node. The predicted values can also be saved to a SAS data set and then imported into the process flow with the Input Data Source node.

The **Ensemble node** enables you to combine models. Ensemble models are expected to exhibit greater stability than individual models. They are most effective when the individual models exhibit lower correlations. The node creates three different types of ensembles:

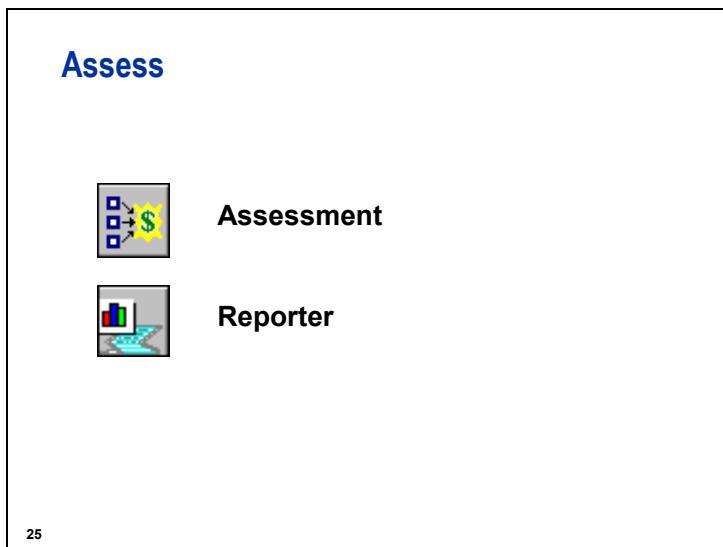
1. Combined model - For example, combining a decision tree and a neural network model. The combination function is the mean of the predicted values.
2. Stratified model - Performing group processing over the values of one or more variables. In this case, there is no combination function because each row in the data set is scored by a single model that is dependent on the value of one or more variables.
4. Bagging/Boosting models – Bagging and boosting models are created by resampling the training data and fitting a separate model for each sample. The predicted values (for interval targets) or the posterior probabilities (for a class target) are then averaged to form the ensemble model. Bagging uses random sampling with replacement to create the  $n$  samples. Each observation is weighted equally. The probability of selecting the next observation is  $1/N$ , where  $N$  represents the number of observations in the sample. Boosting adaptively reweights each training observation. The weights in the resampling are increased for those observations most often misclassified in the previous models. Therefore, the distribution of the observation weights is based on the model performance of the previous samples. Boosting requires a categorical target.

Memory-based reasoning is a process that identifies similar cases and applies the information that is obtained from these cases to a new record. In Enterprise Miner, the **Memory-Based Reasoning node** is a modeling tool that uses a  $k$ -nearest neighbor algorithm to categorize or predict observations. The  $k$ -nearest neighbor algorithm takes a data set and a probe, where each observation in the data set is composed of a set of variables and the probe has one value for each variable. The distance between an observation and the probe is calculated. The  $k$  observations that have the smallest distances to the probe are the  $k$ -nearest neighbors to that probe. In Enterprise Miner, the  $k$ -nearest neighbors are determined by the Euclidean distance between an observation and the probe. Based on the target values of the  $k$ -nearest neighbors, each of the  $k$ -nearest neighbors votes on the target value for a probe. The votes are the posterior probabilities for the class target variable.

The **Two Stage Model node** computes a two-stage model for predicting a class target and an interval target. The interval target variable is usually the value that is associated with a level of the class target. For example, the binary variable PURCHASE has two levels, Yes and No, and the interval variable AMOUNT can be the amount of money that a customer spends on the purchase. The node automatically recognizes the class target and the value target, as well as the probability, classification, and prediction variables. A class model and a value model are fitted for the class target and the interval target, respectively, in the first and the second stages. By defining a transfer function and using the filter option, you are able to specify how the class prediction for the class target is applied and whether to use all or a subset of the training data in the second stage for interval prediction. The prediction

of the interval target is computed from the value model and optionally adjusted by the posterior probabilities of the class target through the bias adjustment option. It also runs a posterior analysis that displays the value prediction for the interval target by the actual value and prediction of the class target. The score code of the Two Stage Model node is a composite of the class and value models. The value model is used to create the assessment plots in the Model Manager and also in the Assessment node.

-  These modeling nodes utilize a directory table facility, called the Model Manager, in which you can store and assess models on demand. The modeling nodes also enable you to modify the target profile(s) for a target variable.



## Assess Nodes

The **Assessment node** provides a common framework for comparing models and predictions from any of the modeling nodes (Regression, Tree, Neural Network, and User Defined Model nodes). The comparison is based on the expected and actual profits or losses that would result from implementing the model. The node produces several charts that help to describe the usefulness of the model such as lift charts and profit/loss charts.

The **Reporter node** assembles the results from a process flow analysis into an HTML report that can be viewed with your favorite Web browser. Each report contains header information, an image of the process flow diagram, and a separate report for each node in the flow. Reports are managed in the Reports tab of the Project Navigator.

## Other Types of Nodes – Scoring Nodes



**Score**



**C\*Score**

26

### Scoring Nodes

The **Score node** enables you to generate and manage predicted values from a trained model. Scoring formulas are created for both assessment and prediction. Enterprise Miner generates and manages scoring formulas in the form of SAS DATA step code, which can be used in most SAS environments even without the presence of Enterprise Miner.

The **C\*Score node** translates the SAS DATA step score code that is generated by Enterprise Miner tools into a score function in the C programming language, as described in the **ISO/IEC 9899 International Standard for Programming Languages - C** handbook. You can save the score function to a plain text output file, enabling you to deploy the scoring algorithms in your preferred C or C++ development environment. The node produces a header file that is used to compile the C code. The C code compiles with any C compiler that supports the **ISO/IEC 9899 International Standard for Programming Languages - C**. It can be linked as a callable C function, for example, as a dynamic link library (DLL). While C\*Score output could be used as the core of a scoring system, it is not a complete scoring system. It provides only the functionality that is explicit in the SAS DATA step scoring code that is translated. The node runs only in Enterprise Miner, and will only translate the score code that is produced by Enterprise Miner nodes. The intended users of the C\*Score node are programmers with experience writing code in the C or C++ languages.

## Other Types of Nodes – Utility Nodes



**Group Processing**



**Data Mining Database**



**SAS Code**



**Control Point**



**Subdiagram**

27

### Utility Nodes

The **Group Processing node** enables you to perform BY-group processing for class variables. You can also use this node to analyze multiple targets, and process the same data source repeatedly by setting the group-processing mode to index.

The **Data Mining Database node** enables you to create a data mining database (DMDB) for batch processing. For nonbatch processing, DMDBs are automatically created as they are needed.

The **SAS Code node** enables you to incorporate new or existing SAS code into process flow diagrams. The ability to write SAS code enables you to include additional SAS System procedures into your data mining analysis. You can also use a SAS DATA step to create customized scoring code, to conditionally process data, and to concatenate or to merge existing data sets. The node provides a macro facility to dynamically reference data sets used for training, validation, testing, or scoring and variables, such as input, target, and predict variables. After you run the SAS Code node, the results and the data sets can then be exported for use by subsequent nodes in the diagram.

The **Control Point node** enables you to establish a control point to reduce the number of connections that are made in process flow diagrams. For example, suppose three Input Data Source nodes are to be connected to three modeling nodes. If no Control Point node is used, then nine connections are required to connect all of the Input Data Source nodes to all of the modeling nodes. However, if a Control Point node is used, only six connections are required.

The **Subdiagram node** enables you to group a portion of a process flow diagram into a subdiagram. For complex process flow diagrams, you may want to create subdiagrams to better design and control the process flow.

Following are some general rules that govern the placement of nodes in a process flow diagram:

- The **Input Data Source** node cannot be preceded by any other nodes.
- All nodes except the **Input Data Source** and **SAS Code** nodes must be preceded by a node that exports a data set.
- The **SAS Code** node can be defined in any stage of the process flow diagram. It does not require an input data set that is defined in the **Input Data Source** node. If you create a SAS data set with code you write in the **SAS Code** node (for example, a data set that contains target and predict variables), you can use a successor **Data Set Attributes** node to assign model roles to the variables in the SAS data set.
- The **Assessment** node must be preceded by one or more modeling nodes.
- The **Score** node must be preceded by a node that produces score code. For example, the modeling nodes produce score code.
- The **Ensemble** node must be preceded by a modeling node.
- The **Reporter** node generates HTML reports for all of the predecessor nodes. It does not generate reports for the successor nodes. When you run the flow from the **Reporter** node, Enterprise Miner makes a first pass through the flow to analyze the data. After the analysis is complete, Enterprise Miner makes a second pass to generate the HTML reports. The node icons are colored green during the analysis pass and yellow during the reporting pass.
- You can have one **Group Processing** node per process flow. Group processing occurs when you run the process flow path. The **Group Processing** node can be connected to any successor node, but the only nodes that accumulate results from each pass are the modeling nodes and the **Score** node.

# Chapter 2 Predictive Modeling Using Decision Trees

<b>2.1</b>	<b>Introduction to Enterprise Miner.....</b>	<b>2-3</b>
<b>2.2</b>	<b>Modeling Issues and Data Difficulties .....</b>	<b>2-20</b>
<b>2.3</b>	<b>Introduction to Decision Trees.....</b>	<b>2-37</b>
<b>2.4</b>	<b>Building and Interpreting Decision Trees .....</b>	<b>2-46</b>



## 2.1 Introduction to Enterprise Miner

### Objectives

- Open Enterprise Miner.
- Explore the workspace components of Enterprise Miner.
- Set up a project in Enterprise Miner.
- Conduct initial data exploration using Enterprise Miner.



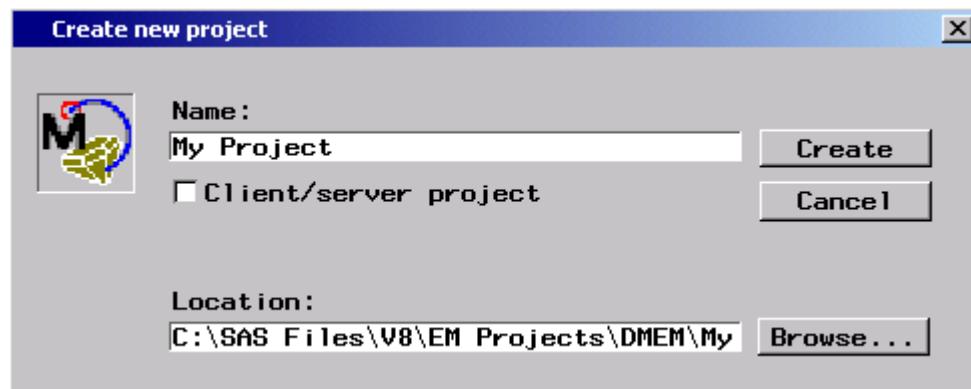
## Introduction to Enterprise Miner

### Opening The Enterprise Miner

1. Start a SAS session. Double-click on the SAS icon on your desktop or select **Start**  $\Rightarrow$  **Programs**  $\Rightarrow$  **The SAS System**  $\Rightarrow$  **The SAS System for Windows V8**.
2. To start Enterprise Miner, type **miner** in the command box or select **Solutions**  $\Rightarrow$  **Analysis**  $\Rightarrow$  **Enterprise Miner**.

### Setting Up the Initial Project and Diagram

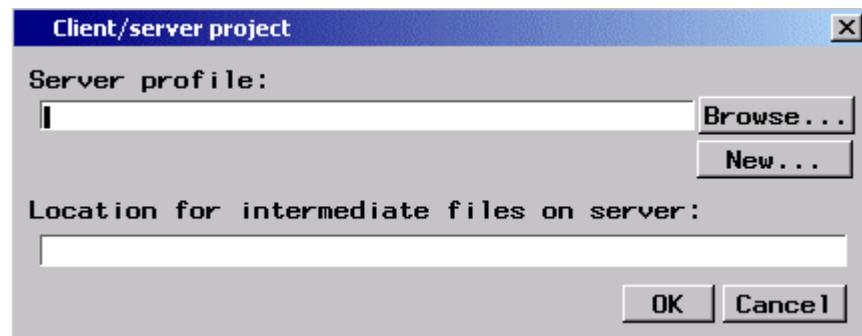
1. Select **File**  $\Rightarrow$  **New**  $\Rightarrow$  **Project...**.
2. Modify the location of the project folder if desired by selecting **Browse...**.
3. Type the name of the project (for example, **My Project**).



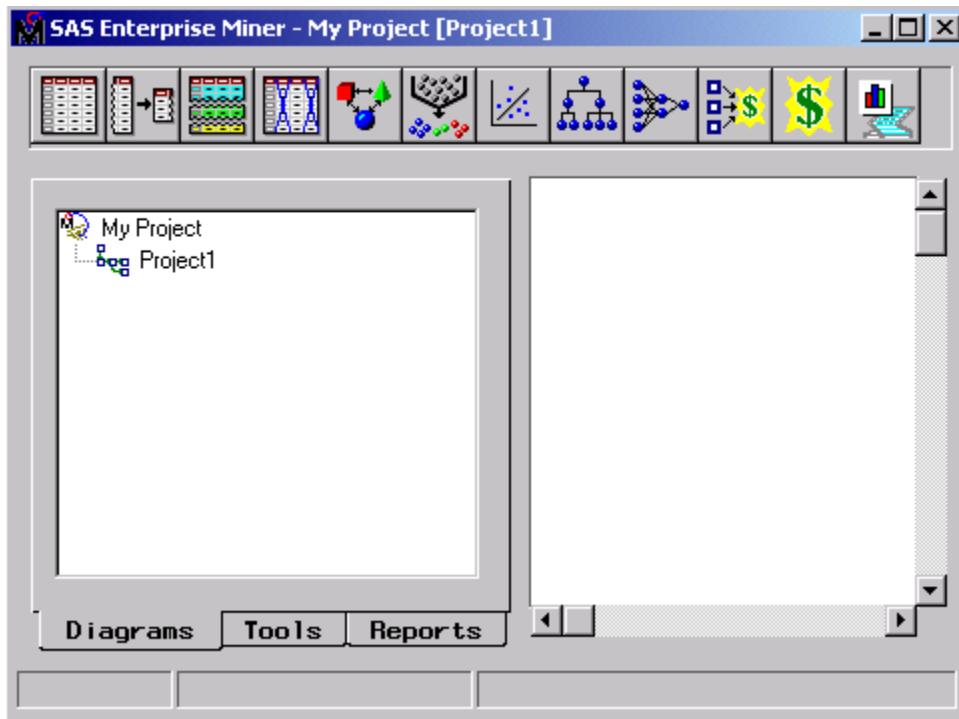
4. Check the box for Client/server project if needed. Do not check this box unless instructed to do so by the instructor.



You must have access to a server running the same version of Enterprise Miner. This allows you to access databases on a remote host or distribute the data-intensive processing to a more powerful remote host. If you create a client/server project, you will be prompted to provide a server profile and to choose a location for files created by Enterprise Miner.

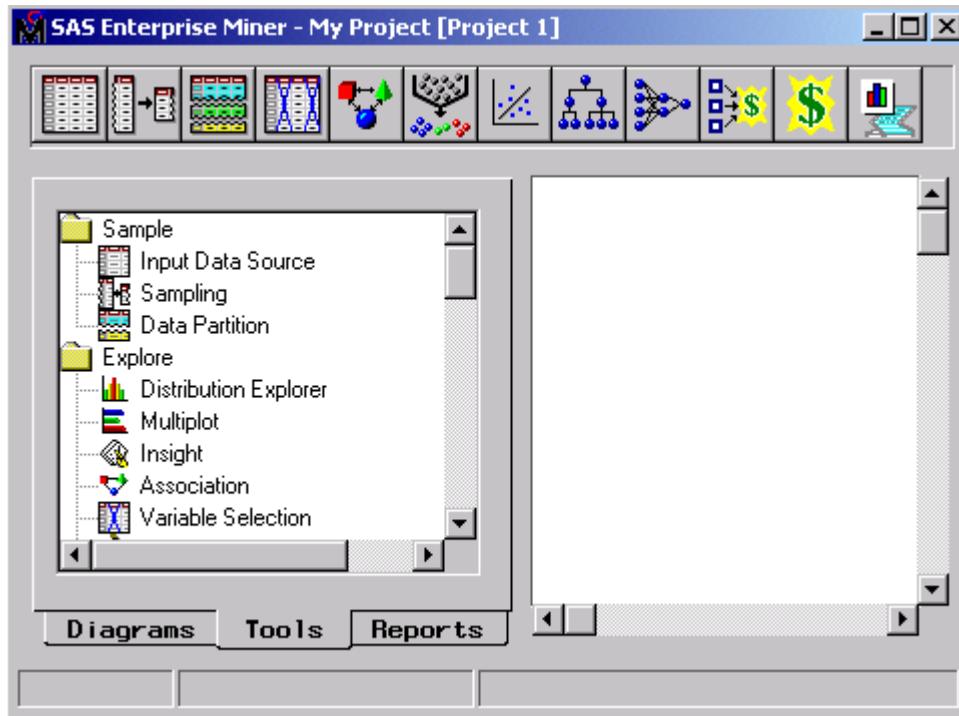


5. Select **Create**. The project opens with an initial untitled diagram.
6. Click on the diagram title and type a new title if desired (for example, **Project1**).



### Identifying the Workspace Components

1. Observe that the project window opens with the Diagrams tab activated. Select the **Tools** tab located to the right of the Diagrams tab in the lower-left portion of the project window. This tab enables you to see all of the tools (or nodes) that are available in Enterprise Miner.



Many of the commonly used tools are shown on the toolbar at the top of the window. If you want additional tools on this toolbar, you can drag them from the window above onto the toolbar. In addition, you can rearrange the tools on the toolbar by dragging each tool to the desired location on the bar.

2. Select the **Reports** tab located to the right of the Tools tab. This tab reveals any reports that have been generated for this project. This is a new project, so no reports are currently available.

The open space on the right is your diagram workspace. This is where you graphically build, order, and sequence the nodes you use to mine your data and generate reports.

## The Scenario

- Determine who should be approved for a home equity loan.
- The target variable is a binary variable that indicates whether an applicant eventually defaulted on the loan.
- The input variables are variables such as the amount of the loan, amount due on the existing mortgage, the value of the property, and the number of recent credit inquiries.

4

The consumer credit department of a bank wants to automate the decision-making process for approval of home equity lines of credit. To do this, they will follow the recommendations of the Equal Credit Opportunity Act to create an empirically derived and statistically sound credit scoring model. The model will be based on data collected from recent applicants granted credit through the current process of loan underwriting. The model will be built from predictive modeling tools, but the created model must be sufficiently interpretable so as to provide a reason for any adverse actions (rejections).

The HMEQ data set contains baseline and loan performance information for 5,960 recent home equity loans. The target (BAD) is a binary variable that indicates if an applicant eventually defaulted or was seriously delinquent. This adverse outcome occurred in 1,189 cases (20%). For each applicant, 12 input variables were recorded.

Name	Model Role	Measurement Level	Description
BAD	Target	Binary	1=defaulted on loan, 0=paid back loan
REASON	Input	Binary	HomeImp=home improvement, DebtCon=debt consolidation
JOB	Input	Nominal	Six occupational categories
LOAN	Input	Interval	Amount of loan request
MORTDUE	Input	Interval	Amount due on existing mortgage
VALUE	Input	Interval	Value of current property
DEBTINC	Input	Interval	Debt-to-income ratio
YOJ	Input	Interval	Years at present job
DEROG	Input	Interval	Number of major derogatory reports
CLNO	Input	Interval	Number of trade lines
DELINQ	Input	Interval	Number of delinquent trade lines
CLAGE	Input	Interval	Age of oldest trade line in months
NINQ	Input	Interval	Number of recent credit inquiries

The credit scoring model computes a probability of a given loan applicant defaulting on loan repayment. A threshold is selected such that all applicants whose probability of default is in excess of the threshold are recommended for rejection.

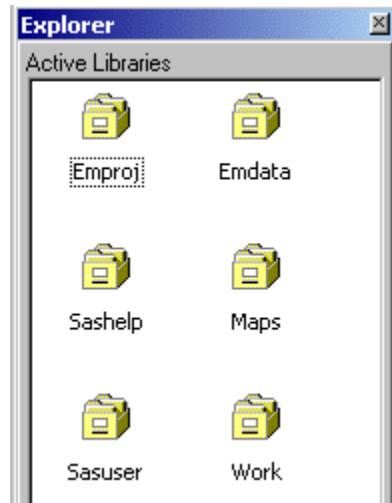


## Project Setup and Initial Data Exploration

### Using SAS Libraries

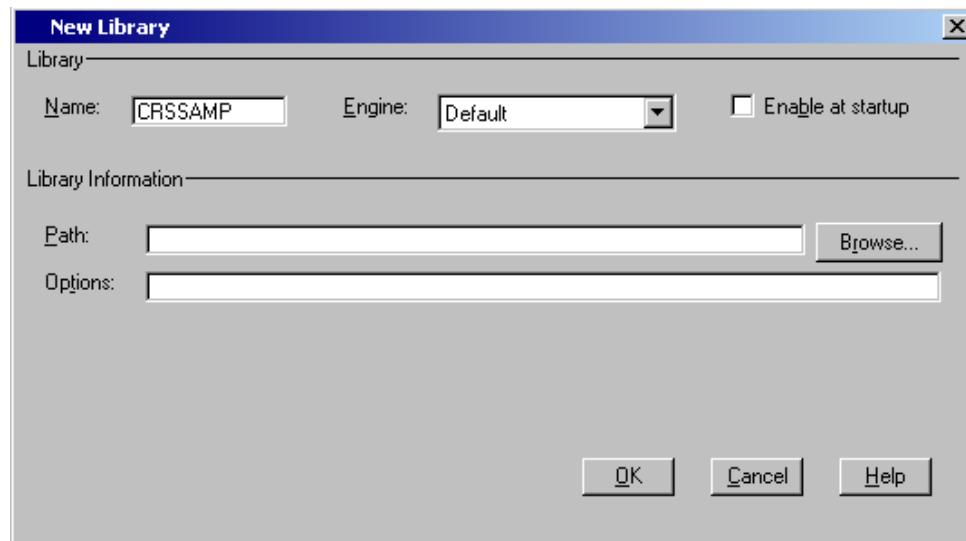
To identify a SAS data library, you assign it a library reference name, or *libref*. When you open Enterprise Miner, several libraries are automatically assigned and can be seen in the Explorer window.

1. Double-click on the Libraries icon in the Explorer window.

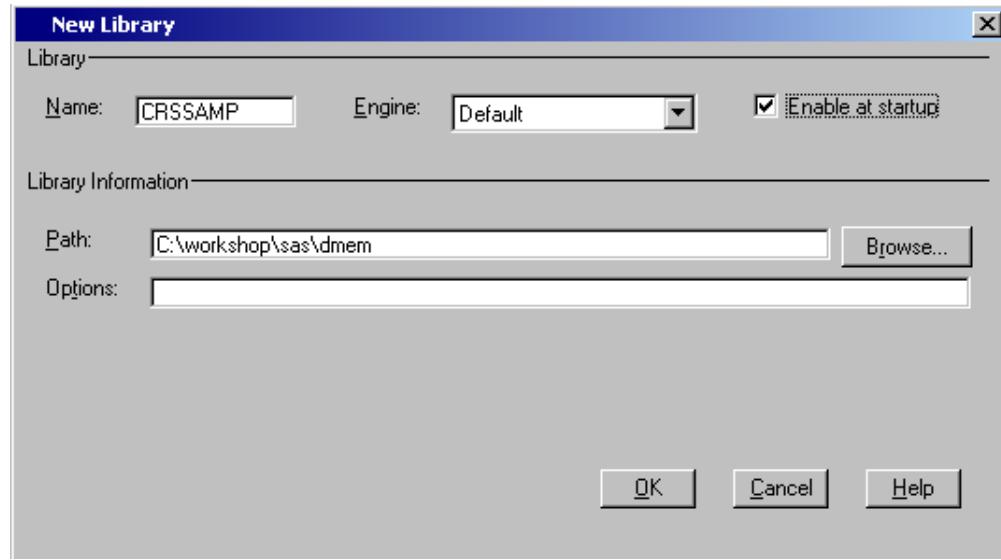


To define a new library:

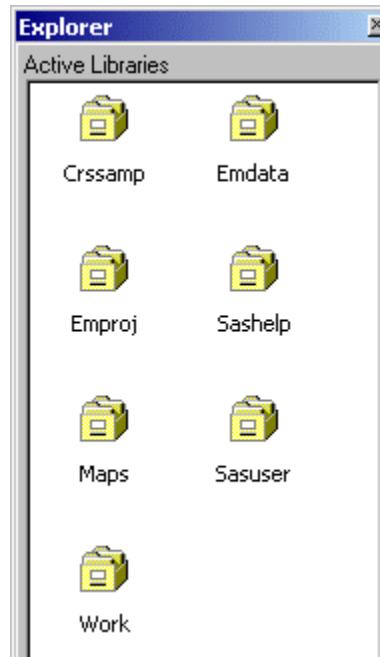
2. Right-click in the Explorer window and select **New**.



3. In the New Library window, type a name for the new library. For example, type CRSSAMP.
4. Type in the path name or select **Browse** to choose the folder to be connected with the new library name. For example, the chosen folder might be located at C:\workshop\sas\dmem.
5. If you want this library name to be connected with this folder every time you open SAS, select **Enable at startup**.



6. Select **OK**. The new library is now assigned and can be seen in the Explorer window.

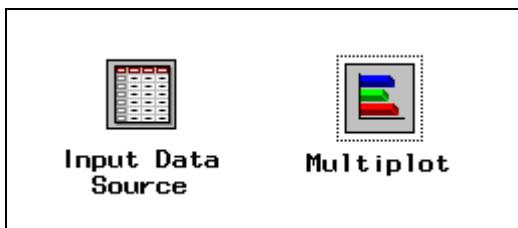


- To view the data sets that are included in the new library, double-click on the icon for Crssamp.



### **Building the Initial Flow**

- Presuming that the diagram Project1 in the project named My Project is open, add an Input Data Source node by dragging the node from the toolbar or from the Tools tab to the diagram workspace.
- Add a Multiplot node to the workspace to the right of the Input Data Source node. Your diagram should appear as shown below.



Observe that the Multiplot node is selected (as indicated by the dotted line around it), but the Input Data Source node is not selected. If you click in any open space on the workspace, all nodes become deselected.

In addition to dragging a node onto the workspace, there are two other ways to add a node to the flow. You can right-click in the workspace where you want the node to be placed and select **Add node** from the pop-up menu, or you can double-click where you want the node to be placed. In either case, a list of nodes appears, enabling you to select the desired node.

The shape of the cursor changes depending on where it is positioned. The behavior of the mouse commands depends on the shape as well as the selection state of the node over which the cursor is positioned. Right-click in an open area to see the menu. The last three menu items (Connect items, Move items, Move and Connect) enable you to modify the ways in which the cursor can be used. Move and Connect is selected by default, and it is highly recommended that you do not change this setting. If your

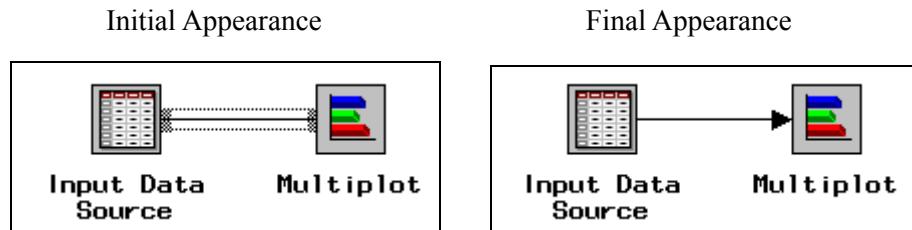
cursor is not performing a desired task, check this menu to make sure that Move and Connect is selected. This selection allows you to move the nodes around the workspace as well as connect them.

Observe that when you put your cursor in the middle of a node, the cursor appears as a hand. To move the nodes around the workspace:

1. Position the cursor in the middle of the node until the hand appears.
2. Press the left mouse button and drag the node to the desired location.
3. Release the left mouse button.

To connect the two nodes in the workspace:

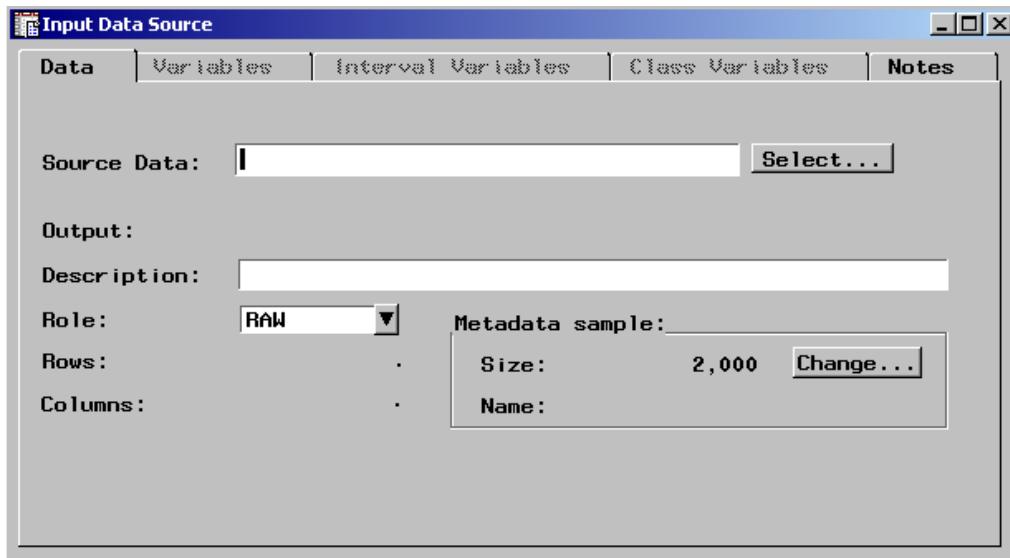
1. Ensure that the Input Data Source node is deselected. It is much easier to drag a line when the node is deselected. If the beginning node is selected, click in an open area of the workspace to deselect it.
2. Position the cursor on the edge of the icon representing the Input Data Source node (until the crosshair appears).
3. Press the left mouse button and immediately begin to drag in the direction of the Multiplot node. If you do not begin dragging immediately after pressing the left mouse button, you select only the node. Dragging a selected node generally results in moving the node; that is, no line forms.
4. Release the mouse button after reaching the edge of the icon that represents the ending node.
5. Click away from the line and the finished arrow forms as shown below.



## Identifying the Input Data

This example uses the HMEQ data set in the CRSSAMP library.

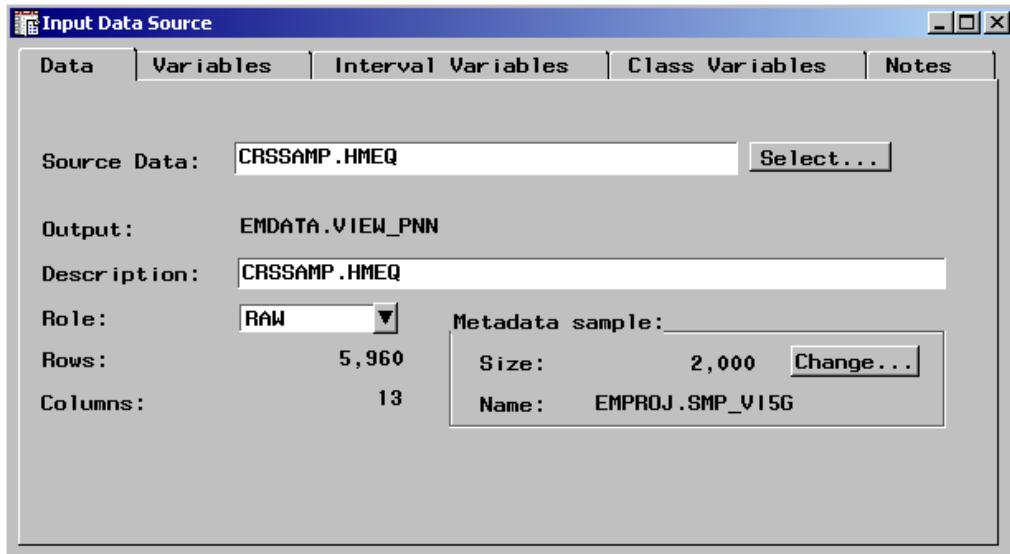
- To specify the input data, double-click on the **Input Data Source** node or right-click on this node and select **Open...**. The Data tab is active. Your window should appear as follows:



- Click on **Select...** to select the data set. Alternatively, you can enter the name of the data set.
- The SASUSER library is selected by default. To view data sets in the CRSSAMP library, click on the **▼** and select **CRSSAMP** from the list of defined libraries.



4. Select **HMEQ** from the list of data sets in the CRSSAMP library and then select **OK**. The dialog shown below opens.



Observe that this data set has 5,960 observations (rows) and 13 variables (columns). CRSSAMP.HMEQ is listed as the source data. You could have typed in this name in the field instead of selecting it through the dialog. Note that the lower-right corner indicates a metadata sample of size 2,000.

All analysis packages must determine how to use variables in the analysis. Enterprise Miner utilizes metadata in order to make a preliminary assessment of how to use each variable. By default, it takes a random sample of 2,000 observations from the data set of interest and uses this information to assign a model role and a measurement level to each variable. To take a larger sample, you can select the Change... button in the lower-right corner of the dialog. However, that is not shown here.

1. Click on the **Variables** tab to see all of the variables and their respective assignments.
2. Click on the first column heading, labeled Name, to sort the variables by their name. You can see all of the variables if you enlarge the window. The following table shows a portion of the information for each of the 13 variables.

Input Data Source					
Data	Variables	Interval Variables	Class Variables	Notes	
Name	Model Role	Measurement	Type	Format	
BAD	input	binary	num	BEST12.	
CLAGE	input	interval	num	BEST12.	
CLNO	input	interval	num	BEST12.	
DEBTINC	input	interval	num	BEST12.	
DELINQ	input	interval	num	BEST12.	
DEROG	input	interval	num	BEST12.	
JOB	input	nominal	char	\$7.	
LOAN	input	interval	num	BEST12.	
MORTDUE	input	interval	num	BEST12.	
NINQ	input	interval	num	BEST12.	
REASON	input	binary	char	\$7.	
VALUE	input	interval	num	BEST12.	
YOJ	input	interval	num	BEST12.	

Observe that two of the columns are grayed out. These columns represent information from the SAS data set that cannot be changed in this node. Type is either character (**char**) or numeric (**num**), and it affects how a variable can be used. The value for Type and the number of levels in the metadata sample of 2,000 is used to identify the model role and measurement level.

The first variable is BAD, which is the target variable. Although BAD is a numeric variable in the data set, Enterprise Miner identifies it as a **binary** variable because it has only two distinct nonmissing levels in the metadata sample. The model role for all **binary** variables is set to **input** by default. You need to change the model role for BAD to target before performing the analysis.

The next five variables (CLAGE through DEROG) have the measurement level **interval** because they are numeric variables in the SAS data set and have more than 10 distinct levels in the metadata sample. The model role for all **interval** variables is set to **input** by default.

The variables JOB and REASON are both character variables in the data set, but they have different measurement levels. REASON is binary because it has only two distinct nonmissing levels in the metadata sample. The model role for JOB, however, is nominal because it is a character variable with more than two levels.

For the purpose of this analysis, treat the remaining variables as interval variables.

- At times, variables such as DEROG and DELINQ will be assigned the model role of **ordinal**. A variable is listed as ordinal when it is a numeric variable with more than two but no more than ten distinct nonmissing levels in the metadata sample. This often occurs with counting variables, such as a variable for the number of children. Because this assignment depends on the metadata sample, the measurement level of DEROG or DELINQ for your analysis might be set to **ordinal**. All ordinal variables are set to have the **input** model role; however, you treat these variables as interval inputs for the purpose of this analysis.

## Identifying Target Variables

BAD is the response variables for this analysis. Change the model role for BAD to **target**.

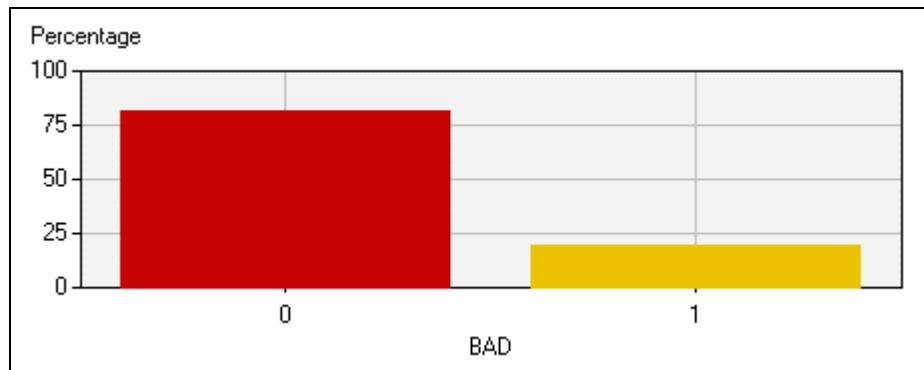
To modify the model role information, proceed as follows:

1. Position the tip of your cursor over the row for BAD in the Model Role column and right-click.
2. Select **Set Model Role**  $\Rightarrow$  **target** from the pop-up menu.

## Inspecting Distributions

You can inspect the distribution of values in the metadata sample for each of the variables. To view the distribution of BAD:

1. Position the tip of your cursor over the variable BAD in the Name column.
2. Right-click and observe that you can sort by name, find a name, or view the distribution of BAD.
3. Select **View Distribution of BAD** to see the distribution of values for BAD in the metadata sample.



To obtain additional information, select the View Info tool, , from the toolbar at the top of the window and click on one of the bars. Enterprise Miner displays the level and the proportion of observations represented by the bar. These plots provide an initial overview of the data. For this example, approximately 20% of the observations were loans where the client defaulted. Because the plots are based on the metadata sample, they may vary slightly due to the differences in the sampled observations, but the bar for BAD=1 should represent approximately 20% of the data. Close the Variable Histogram window when you are finished inspecting the plot. You can evaluate the distribution of other variables as desired.

## Modifying Variable Information

Ensure that the remaining variables have the correct model role and measurement level information. If necessary, change the measurement level for DEROG and DELINQ to **interval**. To modify the measurement level information:

1. Position the tip of your cursor over the row for DEROG in the measurement column and right-click.
2. Select **Set Measurement**  $\Rightarrow$  **interval** from the pop-up menu.
3. Repeat steps 1 and 2 for DELINQ.

Alternatively, you can update the measurement level information for both variables at the same time by highlighting the rows for DEROG and DELINQ simultaneously before following steps 1 and 2 above.

### Investigating Descriptive Statistics

The metadata is used to compute descriptive statistics. Select the **Interval Variables** tab.

Name	Min	Max	Mean	Std Dev.	Missing %
CLAGE	3.0444	1168.2	181.09	91.193	5%
CLNO	0	65	21.378	10.154	3%
DEBTINC	0.7203	143.95	33.691	8.5965	21%
DELINQ	0	11	0.4535	1.13	10%
DEROG	0	9	0.2457	0.8187	13%
LOAN	1100	89900	18500	10918	0%
MORTDUE	2619	399412	75474	44923	9%
NINQ	0	13	1.1188	1.5882	9%
VALUE	8000	854114	102290	56044	2%
YOJ	0	41	8.854	7.4462	10%

Investigate the minimum value, maximum value, mean, standard deviation, percentage of missing observations, skewness, and kurtosis for interval variables. Based on business knowledge of the data, inspecting the minimum and maximum values indicates no unusual values. Observe that DEBTINC has a high percentage of missing values (21%).

Select the **Class Variables** tab.

Name	Values	Missing %	Order
BAD	2	0%	Descending
JOB	6	4%	Ascending
REASON	2	5%	Ascending

Investigate the number of levels, percentage of missing values, and the sort order of each variable. Observe that the sort order for BAD is descending, whereas the sort order for all the others is ascending. This occurs because you have a binary target

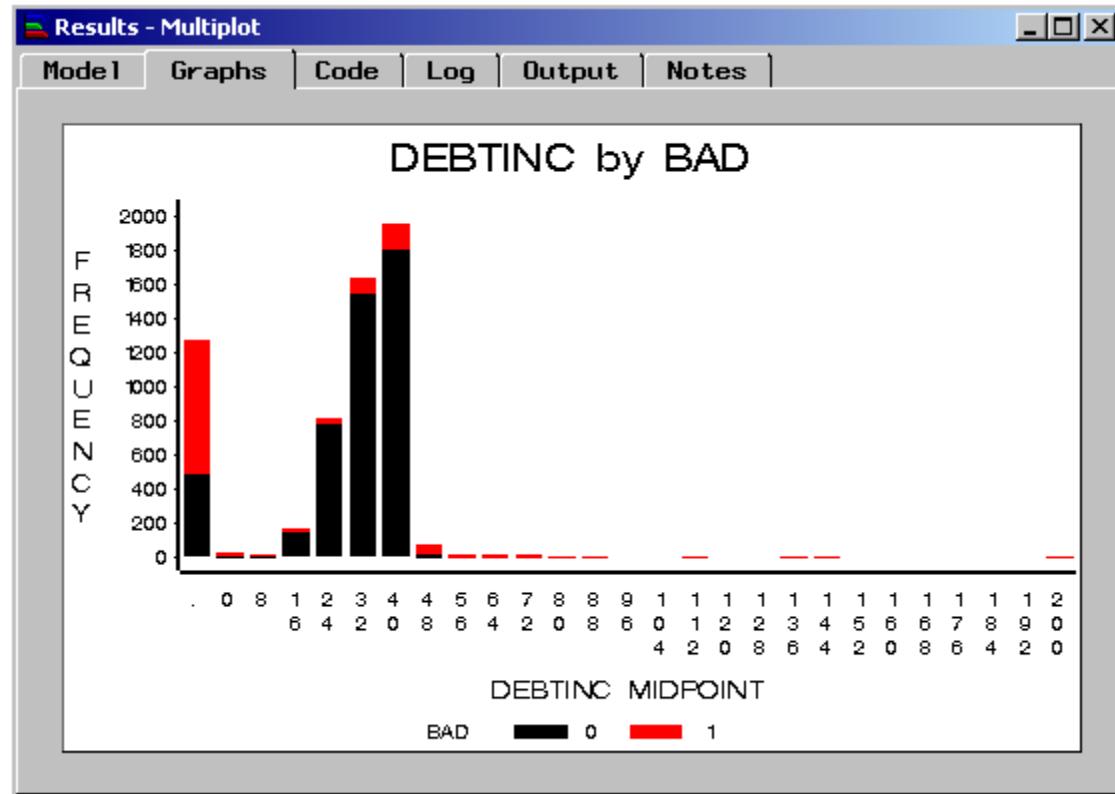
event. It is common to code a binary target with a 1 when the event occurs and a 0 otherwise. Sorting in descending order makes level 1 the first level, which is the target event for a binary variable. It is useful to sort other similarly coded binary variables in descending order for interpreting parameter estimates in a regression model. Close the Input Data Source node, saving changes when prompted.

### Additional Data Exploration

Other tools available in Enterprise Miner enable you to explore your data further. One such tool is the Multiplot node. The Multiplot node creates a series of histograms and bar charts that enable you to examine the relationships between the input variables and the binary target variable.

1. Right-click on the Multiplot node and select **Run**.
2. When prompted, select **Yes** to view the results.

By using the Page Down button on your keyboard, you can view the histograms generated for this data.



From this histogram, you can see that many of the defaulted loans were by homeowners with either a high debt-to-income ratio or an unknown debt-to-income ratio.

- When you open a project diagram in Enterprise Miner, a lock is placed on the diagram to avoid the possibility of more than one person trying to change the diagram at the same time. If Enterprise Miner or SAS terminates abnormally,

the lock files are not deleted and the lock remains on the diagram. If this occurs, you must delete the lock file to gain access to the diagram.

To delete a lock file:

1. Right-click on the project name in the diagrams tab of the workspace and select Explore....



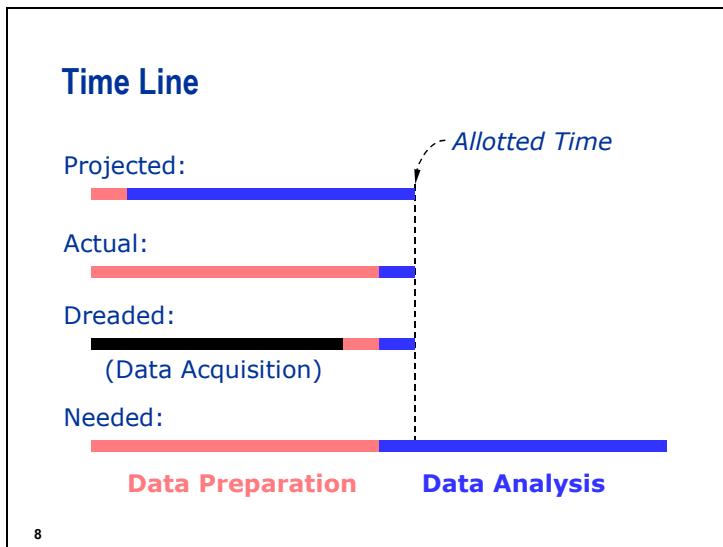
2. In the toolbar of the explorer window that opens, click on .
3. In the Search for files or folders named field, type **\*.lock**.
4. Select .
5. Once the lock file has been located, right-click on the filename and select Delete.

This deletes the lock file and makes the project accessible again.

## 2.2 Modeling Issues and Data Difficulties

### Objectives

- Discuss data difficulties inherent in data mining.
- Examine common pitfalls in model building.



It is often remarked that data preparation takes 90% of the effort for a given project. The truth is that the modeling process could benefit from more effort than is usually given to it, but after a grueling data preparation phase there is often not enough time left to spend on refining the prediction models.

The first step in data preparation is data acquisition, where the relevant data is identified, accessed, and retrieved from various sources; converted; and then consolidated. In many cases, the data acquisition step takes so long that there is little time left for other preparation tasks such as cleaning.

A data warehouse speeds up the data acquisition step. A *data warehouse* is a consolidation and integration of databases designed for information access. The source data usually comes from a transaction-update system stored in operational databases.

## Data Arrangement

<u>Acct</u>	<u>type</u>
2133	MTG
2133	SVG
2133	CK
2653	CK
2653	SVG
3544	MTG
3544	CK
3544	MMF
3544	CD
3544	LOC

*Long-Narrow*

<u>Acct</u>	<u>CK</u>	<u>SVG</u>	<u>MMF</u>	<u>CD</u>	<u>LOC</u>	<u>MTG</u>
2133	1	1	0	0	0	1
2653	1	1	0	0	0	0
3544	1	0	1	1	1	1

*Short-Wide*

10

The data often must be manipulated into a structure suitable for a particular analysis-by-software combination. For example, should this banking data be arranged with multiple rows for each account-product combination or with a single row for each account and multiple columns for each product?

## Derived Inputs

<u>Claim Date</u>	<u>Accident Time</u>	<u>Delay</u>	<u>Season</u>	<u>Dark</u>
11nov96	102396/12:38	19	fall	0
22dec95	012395/01:42	333	winter	1
26apr95	042395/03:05	3	spring	1
02jul94	070294/06:25	0	summer	0
08mar96	123095/18:33	69	winter	0
15dec96	061296/18:12	186	summer	0
09nov94	110594/22:14	4	fall	1

11

The variables relevant to the analysis rarely come prefabricated with opportunistic data. They must be created. For example, the date that an auto accident took place and insurance claim was filed might not be useful predictors of fraud. Derived variables such as the time between the two events might be more useful.

## Roll Up

<u>HH</u>	<u>Acct</u>	<u>Sales</u>
4461	2133	160
4461	2244	42
4461	2773	212
4461	2653	250
4461	2801	122
4911	3544	786
5630	2496	458
5630	2635	328
6225	4244	27
6225	4165	759

<u>HH</u>	<u>Acct</u>	<u>Sales</u>
4461	2133	?
4911	3544	?
5630	2496	?
6225	4244	?

12

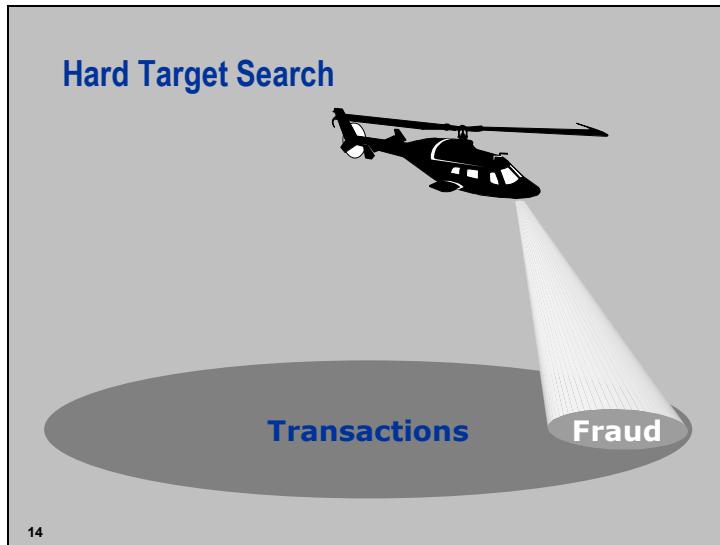
Marketing strategies often dictate rolling up accounts from a single household into a single record (case). This process usually involves creating new summary data. How should the sales figures for multiple accounts in a household be summarized? Using the sum, the mean, the variance, or all three?

## Rolling Up Longitudinal Data

<u>Frequent Flier</u>	<u>Month</u>	<u>Flying Mileage</u>	<u>VIP Member</u>
10621	Jan	650	No
10621	Feb	0	No
10621	Mar	0	No
10621	Apr	250	No
33855	Jan	350	No
33855	Feb	300	No
33855	Mar	1200	Yes
33855	Apr	850	Yes

13

In some situations it may be necessary to roll up longitudinal data into a single record for each individual. For example, suppose an airline wants to build a prediction model to target current frequent fliers for a membership offer in the “Very Important Passenger” club. One record per passenger is needed for supervised classification. How should the flying mileage be consolidated if it is to be used as a predictor of club membership?



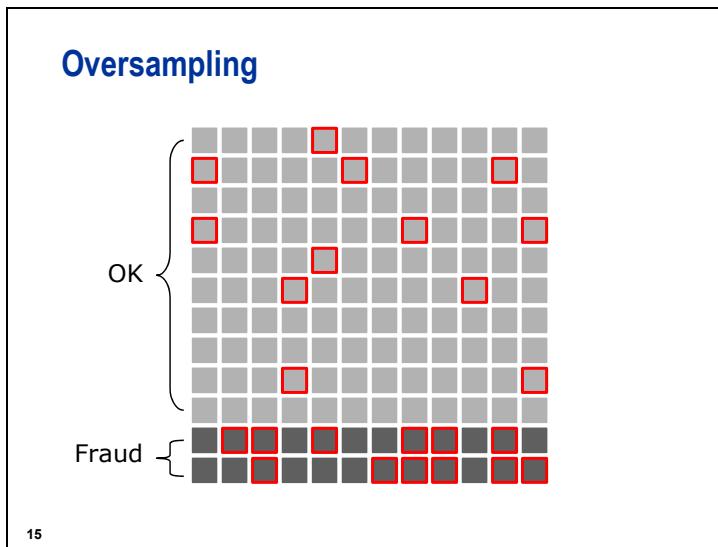
The lack of a target variable is a common example of opportunistic data not having the capacity to meet the objectives. For instance, a utility company may have terabytes of customer usage data and a desire to detect fraud, but it does not know which cases are fraudulent. The data is abundant, but none of it is supervised.

Another example would be healthcare data where the outcome of interest is progress of some condition across time, but only a tiny fraction of the patients were evaluated at more than one time point.

In direct marketing, if customer history and demographics are available but there is no information on response to a particular solicitation of interest, a test mailing is often used to obtain supervised data.

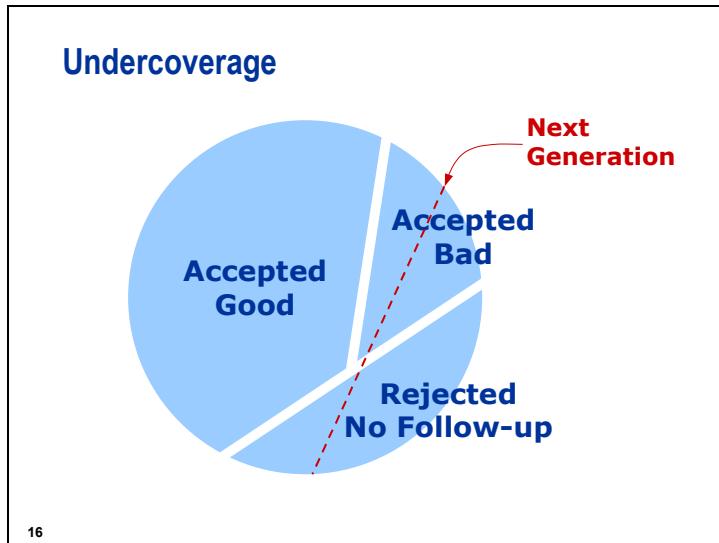
When the data does not have the capacity to solve the problem, the problem needs to be reformulated. For example, there are unsupervised approaches to detecting anomalous data that might be useful for investigating possible fraud.

Initial data examination and analysis does not always limit the scope of the analysis. Getting acquainted with the data and examining summary statistics often inspires more sophisticated questions than were originally posed.



Instead of the lack of a target variable, at times there are very rare target classes (credit card fraud, response to direct mail, and so on). A stratified sampling strategy useful in those situations is *choice-based sampling* (also known as case-control sampling). In choice-based sampling (Scott and Wild 1986), the data are stratified on the target and a sample is taken from each strata so that the rare target class will be more represented in the training set. The model is then built on this biased training set. The effects of the input variables on the target are often estimated with more precision with the choice-based sample even when a smaller overall sample size is taken, compared to a random sample. The results usually must be adjusted to correct for the oversampling.

In assessing how much data is available for data mining, the rarity of the target event must be considered. If there are 12 million transactions, but only 500 are fraudulent, how much data is there? Some would argue that the effective sample size for predictive modeling is much closer to 500 than to 12 million.



The data used to build the model often does not represent the true target population. For example, in credit scoring, information is collected on all applicants. Some are rejected based on the current criterion. The eventual outcome (good/bad) for the rejected applicants is not known. If a prediction model is built using only the accepted applicants, the results may be distorted when used to score future applicants. Undercoverage of the population continues when a new model is built using data from accepted applicants of the current model. Credit-scoring models have proven useful despite this limitation.

*Reject inference* refers to attempts to include the rejected applicants in the analysis. There are several ad hoc approaches, all of which are of questionable value (Hand 1997). The best approach (from a data analysis standpoint) is to acquire outcome data on the rejected applicants by either extending credit to some of them or by purchasing follow-up information on the ones who were given credit by other companies.

## Errors, Outliers, and Missings

<u>cking</u>	<u>#cking</u>	<u>ADB</u>	<u>NSF</u>	<u>dirdep</u>	<u>SVG</u>	<u>bal</u>
Y	1	468.11	1	1876	Y	1208
Y	1	68.75	0	0	Y	0
Y	1	212.04	0	6		0
.	.	.	0	0	Y	4301
Y	2	585.05	0	7218	Y	234
Y	1	-47.69	2	1256		238
Y	1	4687.7	0	0		0
.	.	.	1	0	Y	1208
Y	.	.	.	1598		0
1		0.00	0	0		0
Y	3	89981.12	0	0	Y	45662
Y	2	585.05	0	7218	Y	234

17

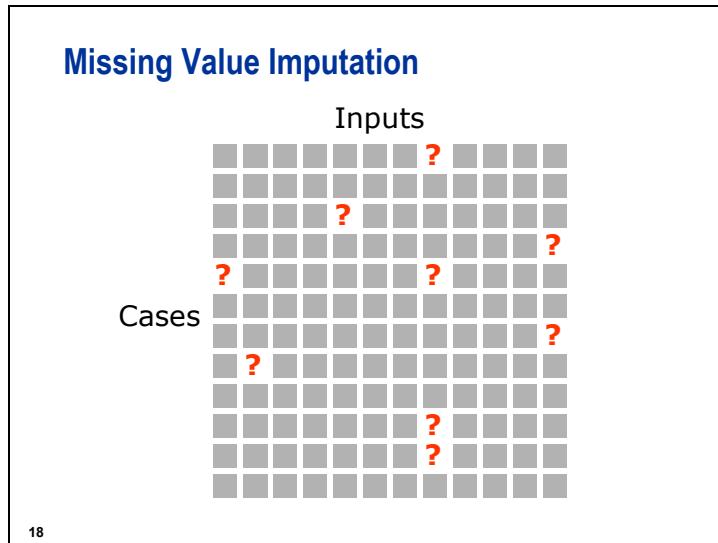
Are there any suspicious values in the above data?

Inadequate data scrutiny is a common oversight. Errors, outliers, and missing values must be detected, investigated, and corrected (if possible). The basic data scrutiny tools are raw listings, if/then subsetting functions, exploratory graphics, and descriptive statistics such as frequency counts and minimum and maximum values.

Detection of such errors as impossible values, impossible combinations of values, inconsistent coding, coding mistakes, and repeated records require persistence, creativity, and domain knowledge.

Outliers are anomalous data values. They may or may not be errors (likewise errors may or may not be outliers). Furthermore, outliers may or may not be influential on the analysis.

Missing values can be caused by a variety of reasons. They often represent unknown but knowable information. Structural missing data represent values that logically could not have a value. Missing values are often coded in different ways and sometimes miscoded as zeros. The reasons for the coding and the consistency of the coding must be investigated.



Two analysis strategies are

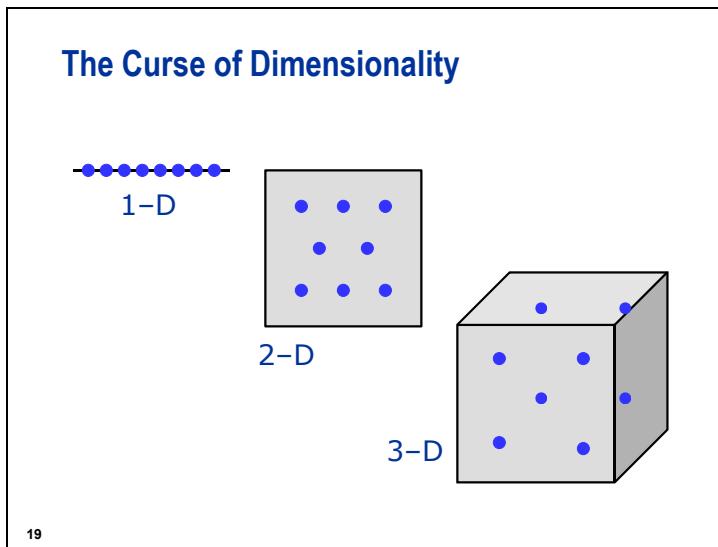
1. complete-case analysis. Use only the cases that have complete records in the analysis. If the “missingness” is related to the inputs or to the target, then ignoring missing values can bias the results.

In data mining, the chief disadvantage with this strategy is practical. Even a smattering of missing values in a high dimensional data set can cause a disastrous reduction in data. In the above example, only 9 of the 144 values (6.25%) are missing, but a complete-case analysis would only use 4 cases—a third of the data set.

2. imputation. Fill in the missing values with some **reasonable** value. Run the analysis on the full (filled-in) data.

The simplest types of imputation methods fill in the missing values with the mean (mode for categorical variables) of the complete cases. This method can be refined by using the mean within homogenous groups of the data.

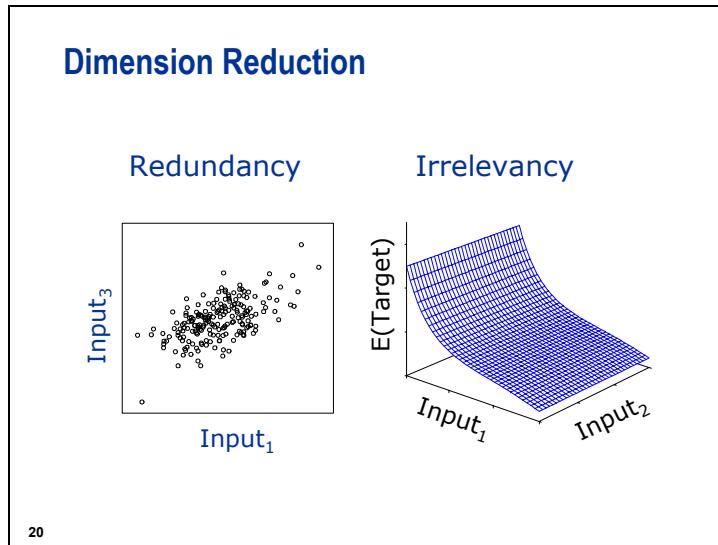
The missing values of categorical variables could be treated as a separate category. For example, type of residence might be coded as own home, buying home, rents home, rents apartment, lives with parents, mobile home, and unknown. This method would be preferable if the missingness is itself a predictor of the target.



The *dimension* of a problem refers to the number of input variables (actually, degrees of freedom). Data mining problems are often massive in both the number of cases and the dimension.

The *curse of dimensionality* refers to the exponential increase in data required to densely populate space as the dimension increases. For example, the eight points fill the one-dimensional space but become more separated as the dimension increases. In 100-dimensional space, they would be like distant galaxies.

The curse of dimensionality limits our practical ability to fit a flexible model to noisy data (real data) when there are a large number of input variables. A densely populated input space is required to fit highly complex models. In assessing how much data is available for data mining, the dimension of the problem must be considered.



Reducing the number of inputs is the obvious way to thwart the curse of dimensionality. Unfortunately, reducing the dimension is also an easy way to disregard important information.

The two principal reasons for eliminating a variable are redundancy and irrelevancy. A redundant input does not give any new information that has not already been explained. Unsupervised methods such as principal components, factor analysis, and variable clustering are useful for finding lower dimensional spaces of nonredundant information.

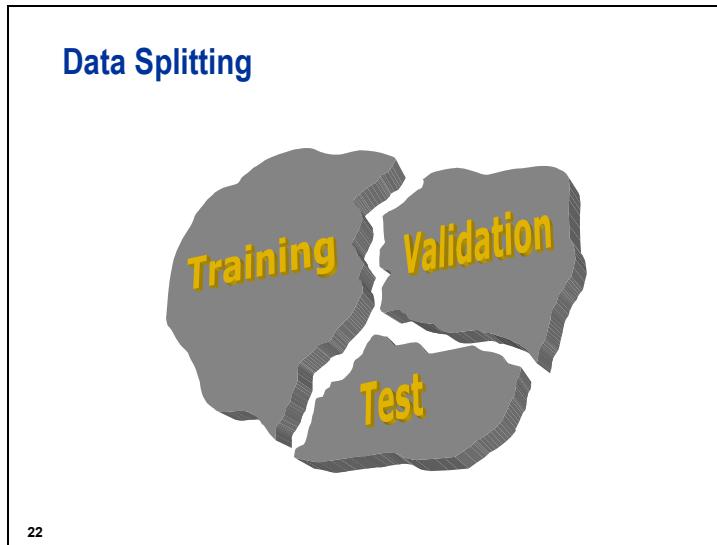
An irrelevant input is not useful in explaining variation in the target. Interactions and partial associations make irrelevancy more difficult to detect than redundancy. It is often useful to first eliminate redundant dimensions and then tackle irrelevancy.

Modern multivariate methods such as neural networks and decision trees have built-in mechanisms for dimension reduction.



*Testing the procedure on the data that gave it birth is almost certain to overestimate performance, for the optimizing process that chose it from among many possible procedures will have made the greatest use of any and all idiosyncrasies of those particular data.*

— Mosteller and Tukey (1977)

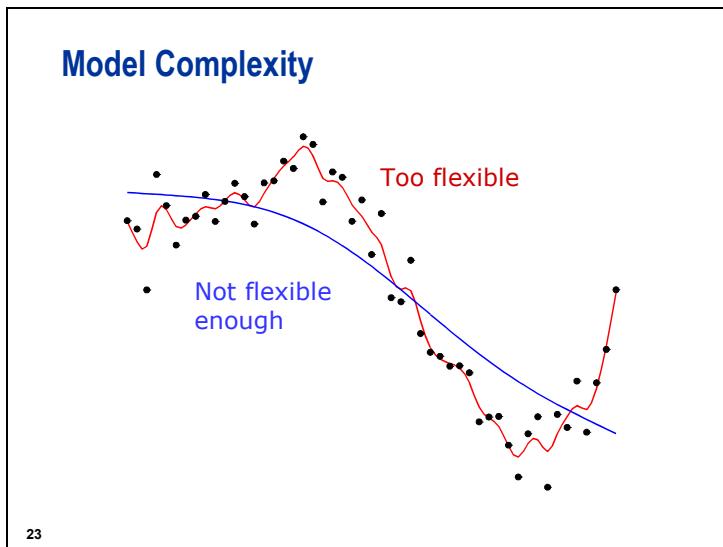


In data mining, the standard strategy for honest assessment of generalization is *data splitting*. A portion is used for fitting the model—the training data set. The rest is held out for empirical validation.

The *validation data set* is used for monitoring and tuning the model to improve its generalization. The tuning process usually involves selecting among models of different types and complexities. The tuning process optimizes the selected model on the validation data. Consequently, a further holdout sample is needed for a final, unbiased assessment.

The *test data set* has only one use: to give a final honest estimate of generalization. Consequently, cases in the test set must be treated just as new data would be treated. They cannot be involved whatsoever in the determination of the fitted prediction model. In some applications, there may be no need for a final honest assessment of generalization. A model can be optimized for performance on the test set by tuning it on the validation set. It may be enough to know that the prediction model will likely give the best generalization possible without actually being able to say what it is. In this situation, no test set is needed.

With small or moderate data sets, data splitting is inefficient; the reduced sample size can severely degrade the fit of the model. Computer-intensive methods such as cross-validation and the bootstrap have been developed so that all the data can be used for both fitting and honest assessment. However, data mining usually has the luxury of massive data sets.



23

Fitting a model to data requires searching through the space of possible models. Constructing a model with good generalization requires choosing the right complexity.

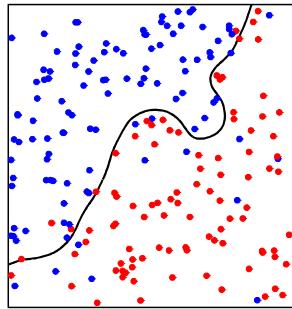
Selecting model complexity involves a trade-off between bias and variance. An insufficiently complex model might not be flexible enough. This leads to underfitting – systematically missing the signal (high bias).

A naive modeler might assume that the most complex model should always outperform the others, but this is not the case. An overly complex model might be too flexible. This will lead to overfitting – accommodating nuances of the random noise in the particular sample (high variance). A model with just enough flexibility will give the best generalization.

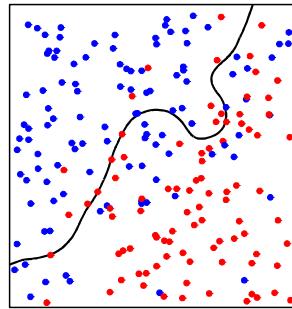
The strategy for choosing model complexity in data mining is to select the model that performs best on the validation data set. Using performance on the training data set usually leads to selecting too complex a model. (The classic example of this is selecting linear regression models based on  $R^2$ .)

## Overfitting

Training Set



Test Set

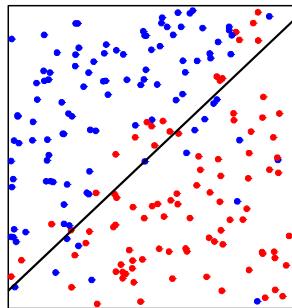


24

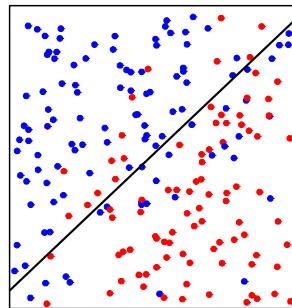
A very flexible model was used on the above classification problem where the goal was to discriminate between the blue and red classes. The classifier fit the training data well, making only 19 errors among the 200 cases (90.5% accuracy). On a fresh set of data, however, the classifier did not do as well, making 49 errors among 200 cases (75.5% accuracy). The flexible model snaked through the training data accommodating the noise as well as the signal.

## Better Fitting

Training Set



Test Set



25

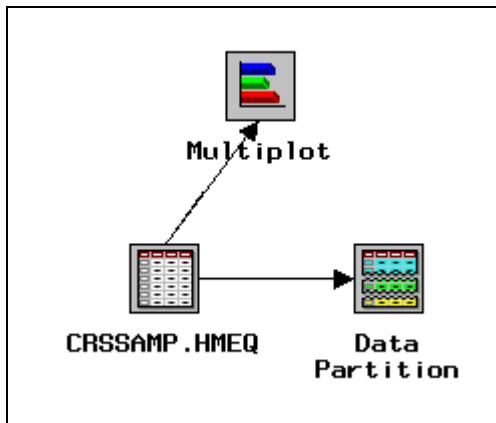
A more parsimonious model was fit to the training data. The apparent accuracy was not quite as impressive as the flexible model (34 errors, 83% accuracy), but it gave better performance on the test set (43 errors, 78.5% accuracy).



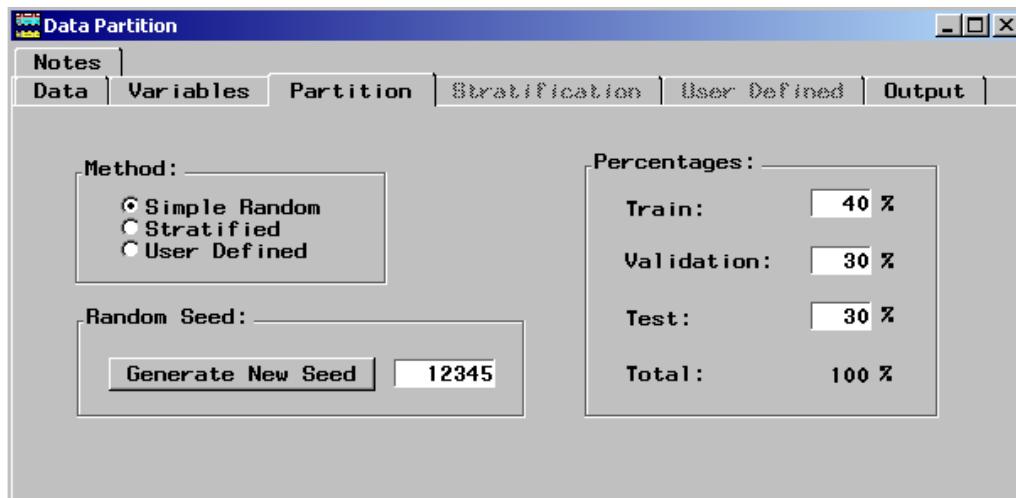
## Exploring the Data Partition Node

### Inspecting Default Settings in the Data Partition Node

1. Add a Data Partition node to the diagram.
2. Connect the Data Partition node to the CRSSAMP.HMEQ node.



3. Open the Data Partition node either by double-clicking on the node, or by right-clicking and selecting [Open...](#).



You choose the method for partitioning in the upper-left section of the tab.

- By default, Enterprise Miner takes a simple random sample of the input data and divides it into training, validation, and test data sets.
- To perform stratified sampling, select the Stratified radio button and then use the options in the Stratified tab to set up your strata.

- To perform user-defined sampling, select the User Defined button and then use the options on the User Defined tab to identify the variable in the data set that identifies the partitions.

You can specify a random seed for initializing the sampling process in the lower-left section of the tab. Randomization within computer programs is often started by some type of seed. If you use the same data set with the same seed in different flows, you get the same partition. Observe that re-sorting the data will result in a different ordering of data and, therefore, a different partition, which will potentially yield different results.

The right side of the tab enables you to specify the percentage of the data to allocate to training, validation, and test data.

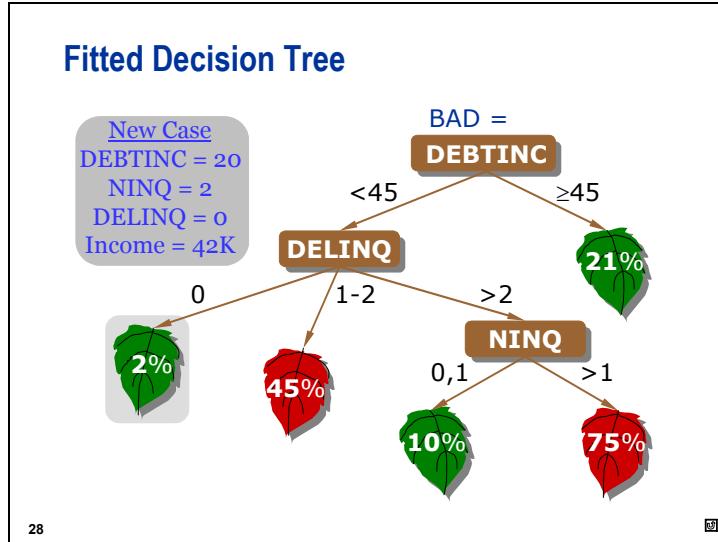
Partition the HMEQ data for modeling. Based on the data available, create training and validation data sets and omit the test data.

4. Set Train, Validation, and Test to 67, 33, and 0, respectively.
5. Close the Data Partition node, and select Yes to save changes when prompted.

## 2.3 Introduction to Decision Trees

### Objectives

- Explore the general concept of decision trees.
- Understand the different decision tree algorithms.
- Discuss the benefits and drawbacks of decision tree models.



Banking marketing scenario:

Target = default on a home-equity line of credit (BAD)

Inputs = number of delinquent trade lines (DELINQ)

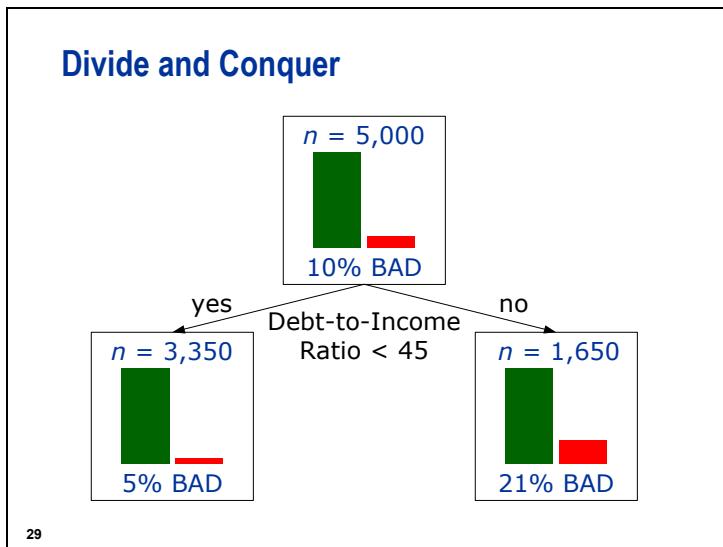
number of credit inquiries (NINQ)

debt to income ratio (DEBTINC)

possibly many other inputs

Interpretation of the fitted decision tree is straightforward. The *internal nodes* contain rules that involve one of the input variables. Start at the *root node* (top) and follow the rules until a terminal node (*leaf*) is reached. The leaves contain the estimate of the expected value of the target – in this case the posterior probability of BAD. The probability can then be used to allocate cases to classes. In this case, green denotes BAD and red denotes otherwise.

When the target is categorical, the decision tree is called a *classification tree*. When the target is continuous, it is called a *regression tree*.



The tree is fitted to the data by *recursive partitioning*. Partitioning refers to segmenting the data into subgroups that are as homogeneous as possible with respect to the target. In this case, the binary split (Debt-to-Income Ratio  $< 45$ ) was chosen. The 5,000 cases were split into two groups, one with a 5% BAD rate and the other with a 21% BAD rate.

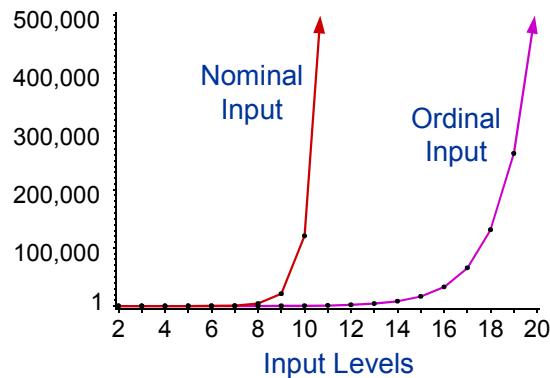
The method is recursive because each subgroup results from splitting a subgroup from a previous split. Thus, the 3,350 cases in the left child node and the 1,650 cases in the right child node are split again in similar fashion.

## The Cultivation of Trees

- Split Search
  - Which splits are to be considered?
- Splitting Criterion
  - Which split is best?
- Stopping Rule
  - When should the splitting stop?
- Pruning Rule
  - Should some branches be lopped off?

30

## Possible Splits to Consider



31

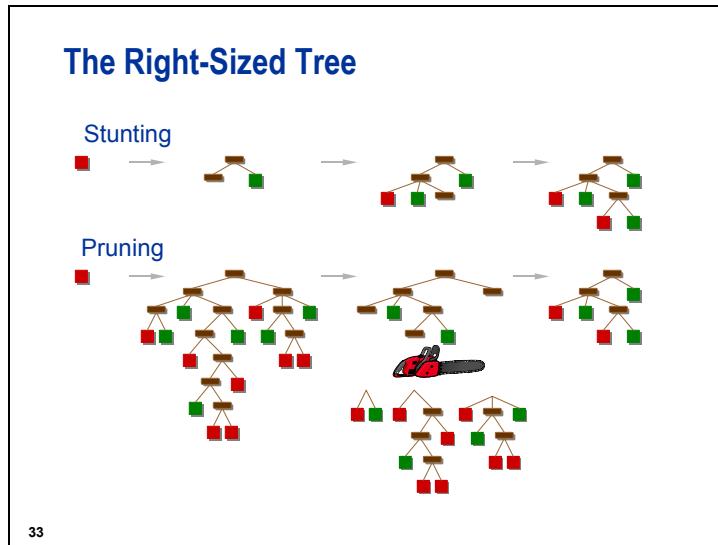
The number of possible splits to consider is enormous in all but the simplest cases. No split search algorithm exhaustively examines all possible partitions. Instead, various restrictions are imposed to limit the possible splits to consider. The most common restriction is to look at only binary splits. Other restrictions involve binning continuous inputs, stepwise search algorithms, and sampling.

Splitting Criteria					
	Left	Right			
Not Bad	3196	1304	4500		Debt-to-Income Ratio < 45
Bad	154	346	500		
	Left	Center	Right		
Not Bad	2521	1188	791	4500	A Competing Three-Way Split
Bad	115	162	223	500	
Not Bad	4500	0	4500		Perfect Split
Bad	0	500	500		

32

How is the best split determined? In some situations, the worth of a split is obvious. If the expected target is the same in the child nodes as in the parent node, no improvement was made, and the split is worthless.

In contrast, if a split results in pure child nodes, the split is undisputedly best. For classification trees, the three most widely used splitting criteria are based on the Pearson chi-squared test, the Gini index, and entropy. All three measure the difference in class distributions across the child nodes. The three methods usually give similar results.

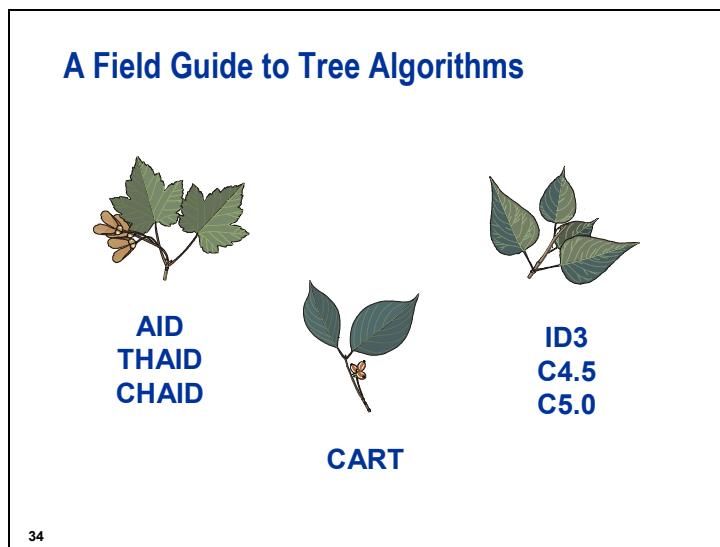


For decision trees, model complexity is the number of leaves. A tree can be continually split until all leaves are pure or contain only one case. This tree would give a perfect fit to the training data but would probably give poor predictions on new data. At the other extreme, the tree could have only one leaf (the root node). Every case would have the same predicted value (no-data rule). There are two approaches to determining the right-sized tree:

1. Using forward-stopping rules to stunt the growth of a tree (prepruning).
- A universally accepted prepruning rule is to stop growing if the node is pure. Two other popular rules are to stop if the number of cases in a node falls below a specified limit or to stop when the split is not statistically significant at a specified level.
2. Growing a large tree and pruning back branches (postpruning).

Postpruning creates a sequence of trees of increasing complexity. An assessment criterion is needed for deciding the best (sub) tree. The assessment criteria are usually based on performance on holdout samples (validation data or with cross-validation). Cost or profit considerations can be incorporated into the assessment.

Prepruning is less computationally demanding but runs the risk of missing future splits that occur below weak splits.



Hundreds of decision tree algorithms have been proposed in the statistical, machine learning, and pattern recognition literature. The most commercially popular are CART, CHAID, and C4.5 (C5.0).

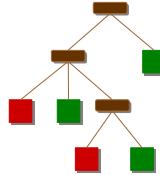
There are many variations of the CART (classification and regression trees) algorithm (Breiman et al. 1984). The standard CART approach is restricted to binary splits and uses post-pruning. All possible binary splits are considered. If the data is very large, within-node sampling can be used. The standard splitting criterion is based on the Gini index for classification trees and variance reduction for regression trees. Other criteria for multiclass problems (the twoing criterion) and regression trees (least absolute deviation) are also used. A maximal tree is grown and pruned back using  $v$ -fold cross-validation. Validation data can be used if there is sufficient data.

CHAID (chi-squared automatic interaction detection) is a modification of the AID algorithm that was originally developed in 1963 (Morgan and Sonquist 1963, Kass 1980). CHAID uses multiway splits and prepruning for growing classification trees. It finds the best multiway split using a stepwise agglomerate algorithm. The split search algorithm is designed for categorical inputs, so continuous inputs must be discretized. The splitting and stopping criteria are based on statistical significance (Chi-squared test).

The ID3 family of classification trees was developed in the machine learning literature (Quinlan 1993). C4.5 only considers  $L$ -way splits for  $L$ -level categorical inputs and binary splits for continuous inputs. The splitting criteria are based on information (entropy) gain. Postpruning is done using pessimistic adjustments to the training set error rate.

## Benefits of Trees

- Interpretability
  - tree-structured presentation
- Mixed Measurement Scales
  - nominal, ordinal, interval
- Regression trees
- Robustness
- Missing Values



35



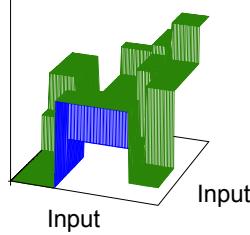
The tree diagram is useful for assessing which variables are important and how they interact with each other. The results can often be written as simple rules such as: If  $(DEBTINC \geq 45)$  or  $(Debits < 45 \text{ and } 1 \leq DELINQ \leq 2)$  or  $(Debits < 45 \text{ and } ADB > 2 \text{ and } NINQ > 1)$ , then  $BAD=yes$ , otherwise no.

Splits based on numeric input variables depend only on the rank order of the values. Like many nonparametric methods based on ranks, trees are robust to outliers in the input space.

Recursive partitioning has special ways of treating missing values. One approach is to treat missings as a separate level of the input variable. The missings could be grouped with other values in a node or have their own node. Another approach is to use surrogate splits; if a particular case has a missing value for the chosen split, you can use a nonmissing input variable that gives a similar split instead.

## Benefits of Trees

Prob

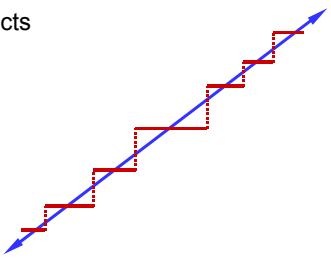


- Automatically
  - Detects interactions (AID)
  - Accommodates nonlinearity
  - Selects input variables

36

## Drawbacks of Trees

- Roughness
- Linear, Main Effects
- Instability



37

The fitted model is composed of discontinuous flat surfaces. The predicted values do not vary smoothly across the input space like other models. This roughness is the trade-off for interpretability.

A step function fitted to a straight line needs many small steps. Stratifying on an input variable that does not interact with the other inputs needlessly complicates the structure of the model. Consequently, linear additive inputs can produce complicated trees that miss the simple structure.

Trees are unstable because small perturbations in the training data can sometimes have large effects on the topology of the tree. The effect of altering a split is compounded as it cascades down the tree and as the sample size decreases.

## 2.4 Building and Interpreting Decision Trees

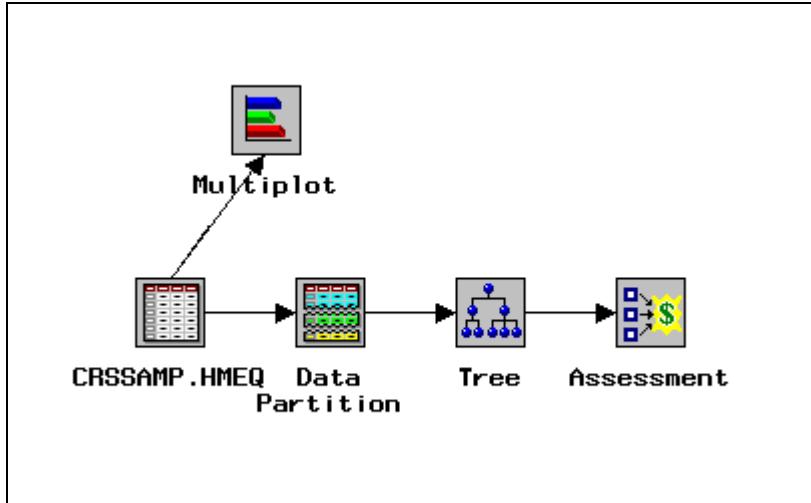
### Objectives

- Explore the types of decision tree models available in Enterprise Miner.
- Build a decision tree model.
- Examine the model results and interpret these results.
- Choose a decision threshold theoretically and empirically.



## Building and Interpreting Decision Trees

To complete the first phase of your first diagram, add a Tree node and an Assessment node to the workspace and connect the nodes as shown below:



Examine the default setting for the decision tree.

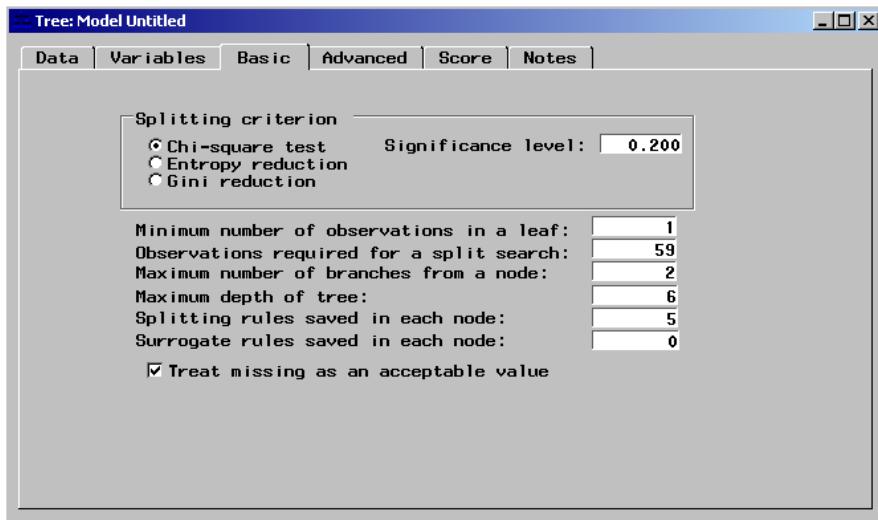
1. Double-click on the Tree node to open it.
2. Examine the Variables tab to ensure all variables have the appropriate status, model role, and measurement level.

**Tree: Model Untitled**

Name	Status	Model Role	Measurement	Type	Format	Label
BAD	use	target	binary	num	BEST12.	
CLAGE	use	input	interval	num	BEST12.	
CLNO	use	input	interval	num	BEST12.	
DEBTINC	use	input	interval	num	BEST12.	
DELINQ	use	input	interval	num	BEST12.	
DEROG	use	input	interval	num	BEST12.	
JOB	use	input	nominal	char	\$7.	
LOAN	use	input	interval	num	BEST12.	
MORTDUE	use	input	interval	num	BEST12.	
NINQ	use	input	interval	num	BEST12.	
REASON	use	input	binary	char	\$7.	
VALUE	use	input	interval	num	BEST12.	
Y0J	use	input	interval	num	BEST12.	

- If the model role or measurement level were not correct, it could not be corrected in this node. You would return to the input data source node to make the corrections.

3. Select the **Basic** tab.



Many of the options discussed earlier for building a decision tree are controlled in this tab.

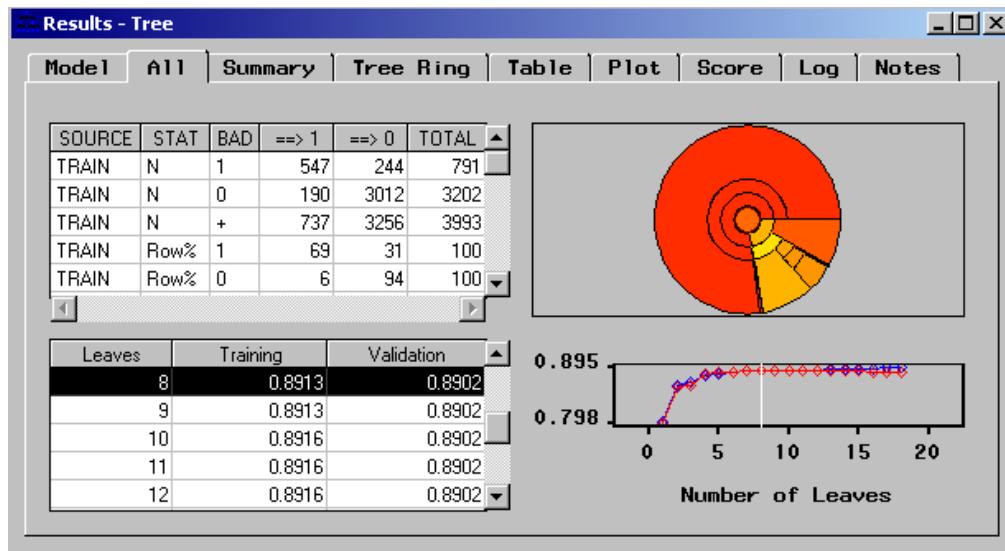
The splitting criteria available depend on the measurement level of the target variable. For binary or nominal target variables, the default splitting criterion is the chi-square test with a significance level of 0.2. Alternately, you could choose to use entropy reduction or Gini reduction as the splitting criterion. For an ordinal target variable, only entropy reduction or Gini reduction are available. For an interval target variable, you have a choice of two splitting criteria: the default F test or variance reduction.

The other options available in this tab affect the growth and size of the tree. By default, only binary splits are permitted, the maximum depth of the tree is 6 levels, and the minimum number of observations in a leaf is 1. However, there is also a setting for the required number of observations in a node in order to split the node. The default is the total number of observations available in the training data set divided by 100.

There are additional options available in the Advanced tab. All of the options are discussed in greater detail in the Decision Tree Modeling course.

4. Close the Tree node.
5. Run the diagram from the Tree node. Right-click on the **Tree** node and select **Run**.
6. When prompted, select **Yes** to view the results.

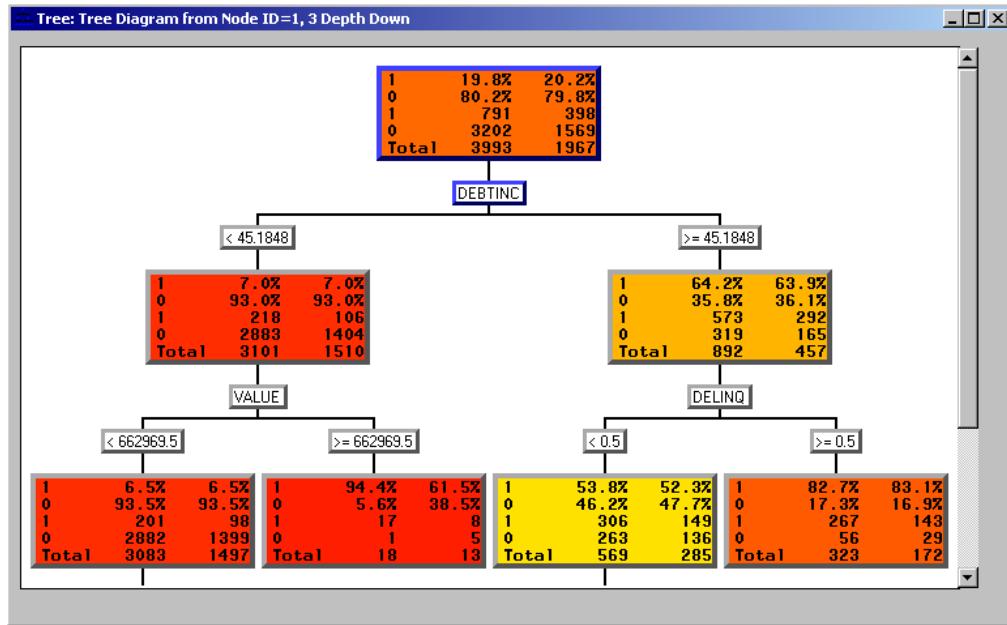
When you view the results of the tree node, the All tab is active and displays a summary of several of the subtabs.



From the graph in the lower-right corner, you can see that a tree with 18 leaves was originally grown based on the training data set and pruned back to a tree with 8 leaves based on the validation data set. The table in the lower-left corner shows that the 8-leaf model has an accuracy of 89.02% on the validation data set.

7. View the tree by selecting **View**  $\Rightarrow$  **Tree** from the menu bar.

A portion of the tree appears below.



Although the selected tree was supposed to have eight leaves, not all eight leaves are visible. By default, the decision tree viewer displays three levels deep.

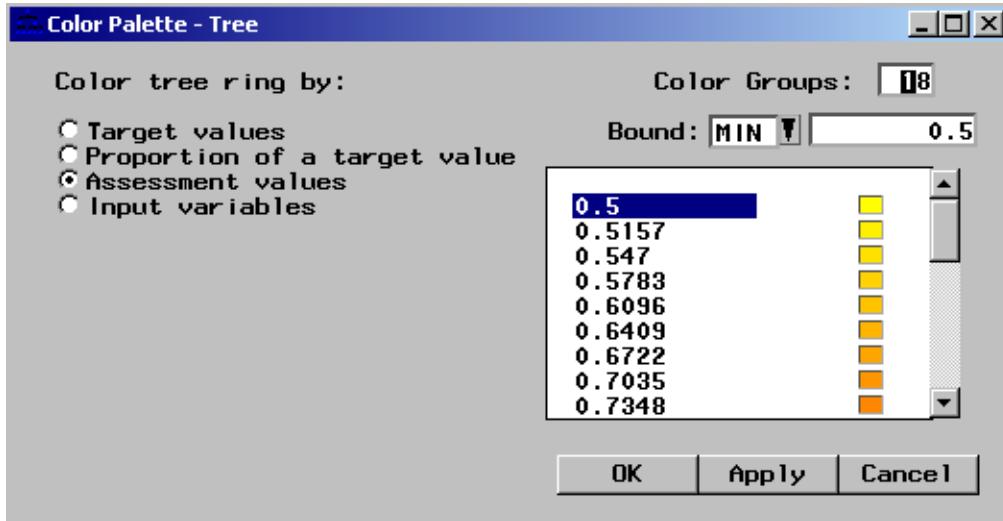
To modify the levels that are visible, proceed as follows:

1. Select **View**  $\Rightarrow$  **Tree Options...**
2. Type **6** in the Tree depth down field.
3. Select **OK**.
4. Verify that all eight leaves are visible.

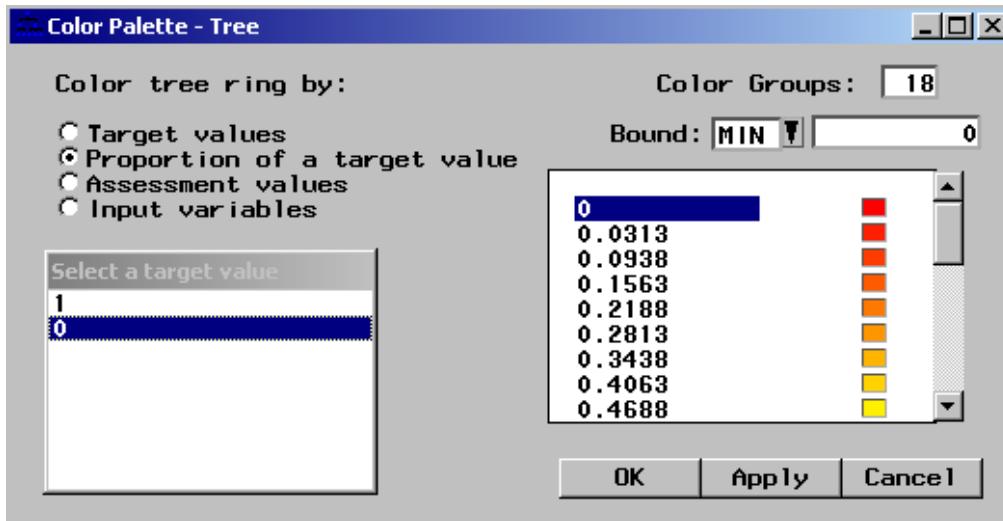
The colors in the tree ring diagram and the decision tree itself indicate node purity by default. If the node contains all ones or all zeros, the node is colored red. If the node contains an equal mix of ones and zeros, it is colored yellow.

You can change the coloring scheme as follows:

1. Select **Tools**  $\Rightarrow$  **Define Colors**.



2. Select the **Proportion of a target value** radio button.

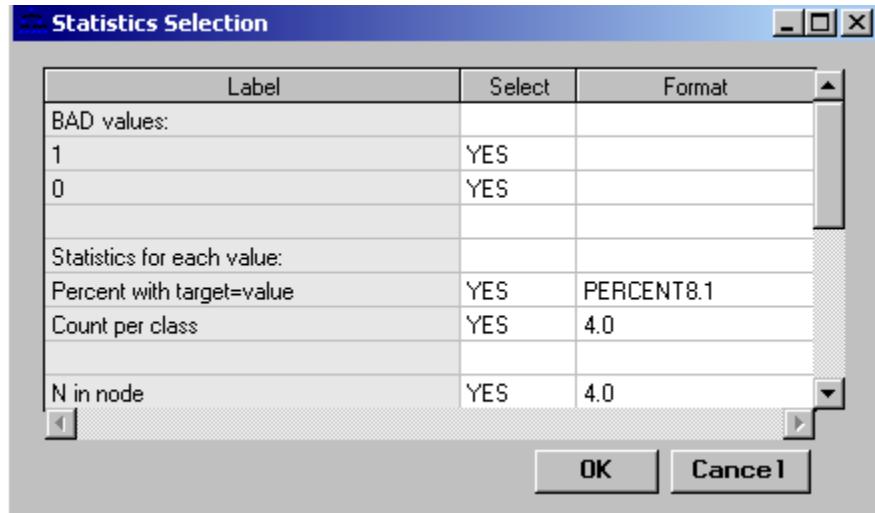


3. Select **0** in the Select a target value table. Selecting zero as the target value makes the leaves with all zeros green and those with no zeros (that is, all ones) red. In other words, leaves that include only individuals who will default on their loan will be red.
4. Select **OK**.

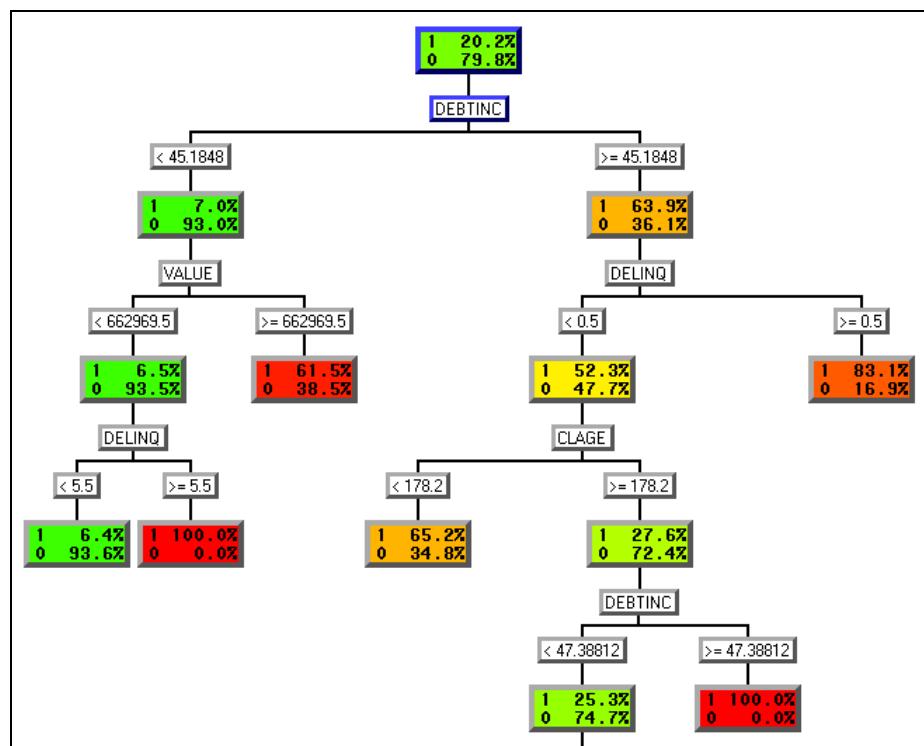
Inspect the tree diagram to identify the terminal nodes with a high percentage of bad loans (colored red) and those with a high percentage of good loans (colored green).

You can also change the statistics displayed in the tree nodes.

1. Select View  $\Rightarrow$  Statistics...

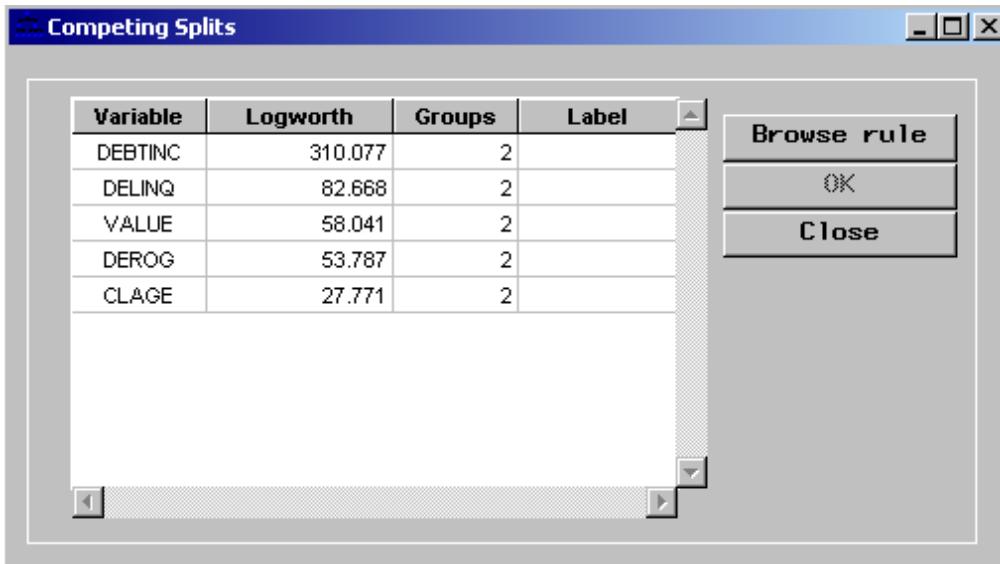


2. To turn off the count per class, right-click in the Select column in the Count per class row. Select Set Select  $\Rightarrow$  No from the pop-up menus.
3. Turn off the N in node, Predicted Value, Training Data, and Node ID rows in the same way. This enables you to see more of the tree on your screen.
4. Select OK.

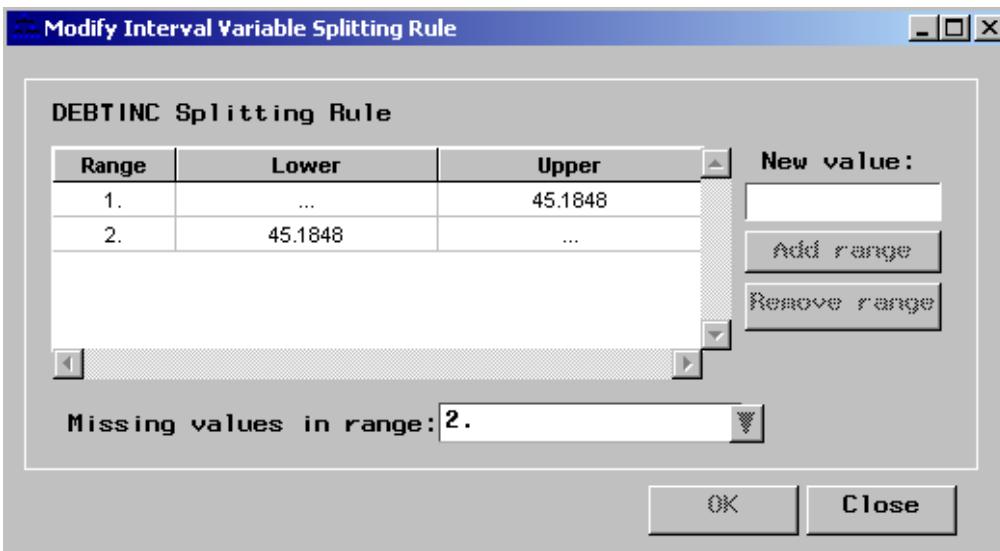


Note that the initial split on DEBTINC has produced two branches. Do the following to determine which branch contains the missing values:

1. Position the tip of the cursor above the variable name DEBTINC directly below the root node in the tree diagram.
2. Right-click and select **View competing splits...**. The Competing Splits window opens. The table lists the top five inputs considered for splitting as ranked by a measure of worth.



3. Select the row for the variable DEBTINC.
4. Select **Browse rule**. The Modify Interval Variable Splitting Rule window opens.



The table presents the selected ranges for each of the branches as well as the branch number that contains the missing values. In this case the branch that contains the values greater than 45.1848 also contains the missing values.

5. Close the Modify Interval Variable Splitting Rule window, the Competing Splits window, and the tree diagram.



You can also see splitting information using the Tree Ring tab in the Results-Tree window. Using the View Info tool, you can click on the partitions in the tree ring plot to see the variable and cutoff value used for each split. The sizes of the resulting nodes are proportional to the size of the segments in the tree ring plot. You can see the split statistics by selecting **View**  $\Rightarrow$  **Probe tree ring statistics**. You can view a path to any node by selecting it and then selecting **View**  $\Rightarrow$  **Path**.

You can also determine the variables that were important in growing the tree in the Score tab.

1. Select the **Score** tab.
2. Select the **Variable Selection** subtab.

Name	Importance	Role	Rules	Variable Label
DEBTINC	1.0000	input	2	
DELINQ	0.4947	input	2	
CLAGE	0.4030	input	1	
VALUE	0.2195	input	2	
DEROG	0.0000	rejected	0	
JOB	0.0000	rejected	0	
REASON	0.0000	rejected	0	
MORTDUE	0.0000	rejected	0	
NINQ	0.0000	rejected	0	
LOAN	0.0000	rejected	0	
YOJ	0.0000	rejected	0	
CLNO	0.0000	rejected	0	

This subtab gives the relative importance of variables used in growing the tree. It also can be used to export new variable roles, which is discussed later in the course.

3. Close the Results window and save the changes when prompted.



### New Tree Viewer

A new tree viewer will be available in a future version of Enterprise Miner. To obtain access to this new viewer,

1. In the command bar, type the statement `%let emv4tree=1`.



2. Press the return key.
3. Return to the Enterprise Miner window.
4. Right-click on the Tree node and select New view....

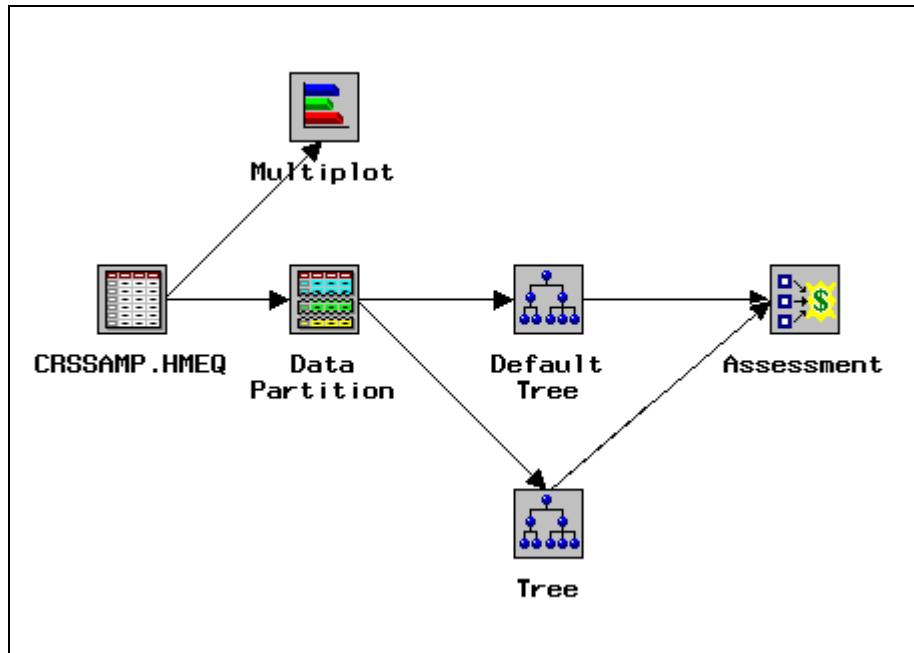
### Using Tree Options

You can make adjustments to the default tree algorithm that causes your tree to grow differently. These changes do not necessarily improve the classification performance of the tree, but they may improve its interpretability.

The Tree node splits a node into two nodes by default (called *binary splits*). In theory, trees using multiway splits are no more flexible or powerful than trees using binary splits. The primary goal is to increase interpretability of the final result.

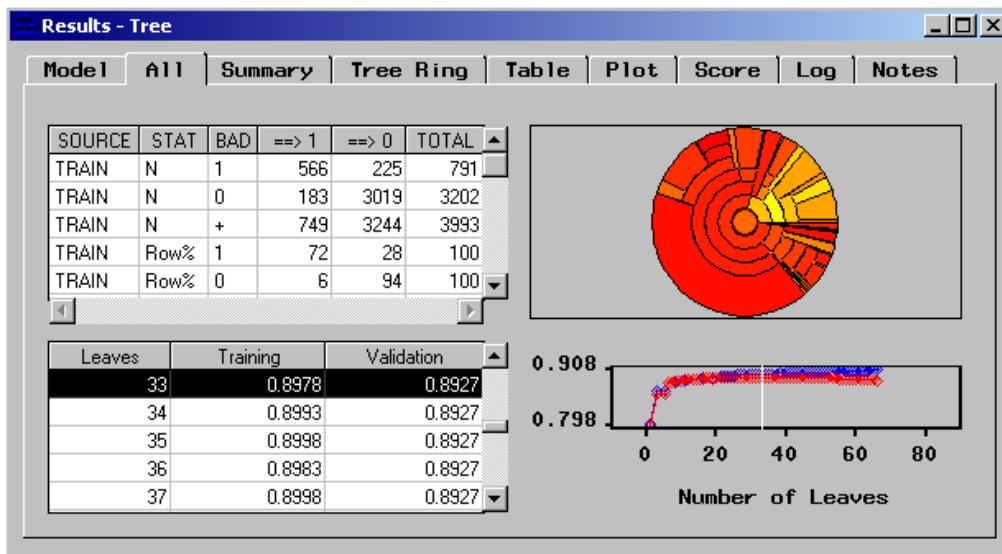
Consider a competing tree that allows up to 4-way splits.

1. Click on the label for the tree in the diagram, and change the label to **Default Tree**.
2. Add another Tree node to the workspace.
3. Connect the Data Partition node to the Tree node.
4. Connect the Tree node to the Assessment node.



5. Open the new Tree node.
  6. Select the **Basic** tab.
  7. Enter **4** for the Maximum number of branches from a node field. This option will allow binary, 3-way, and 4-way splits to be considered.
- Maximum number of branches from a node: **4**
8. Close the Tree node, saving changes when prompted.
  9. Enter **DT4way** as the model name when prompted. This will remind you that you specified up to 4-way splits.
  10. Select **OK**.
  11. Run the flow from this Tree node and view the results.

The number of leaves in the selected tree has increased from 8 to 33. It is a matter of preference as to whether this tree is more comprehensible than the binary split tree. The increased number of leaves suggests to some a lower degree of comprehensibility. The accuracy on the validation set is only 0.25% higher than the default model in spite of greatly increased complexity.



If you inspect the tree diagram, there are many nodes containing only a few applicants. You can employ additional cultivation options to limit this phenomenon.

12. Close the Results window.

### Limiting Tree Growth

Various stopping or stunting rules (also known as prepruning) can be used to limit the growth of a decision tree. For example, it may be deemed beneficial not to split a node with fewer than 50 cases and require that each node have at least 25 cases.

Modify the most recently created Tree node and employ these stunting rules to keep the tree from generating so many small terminal nodes.

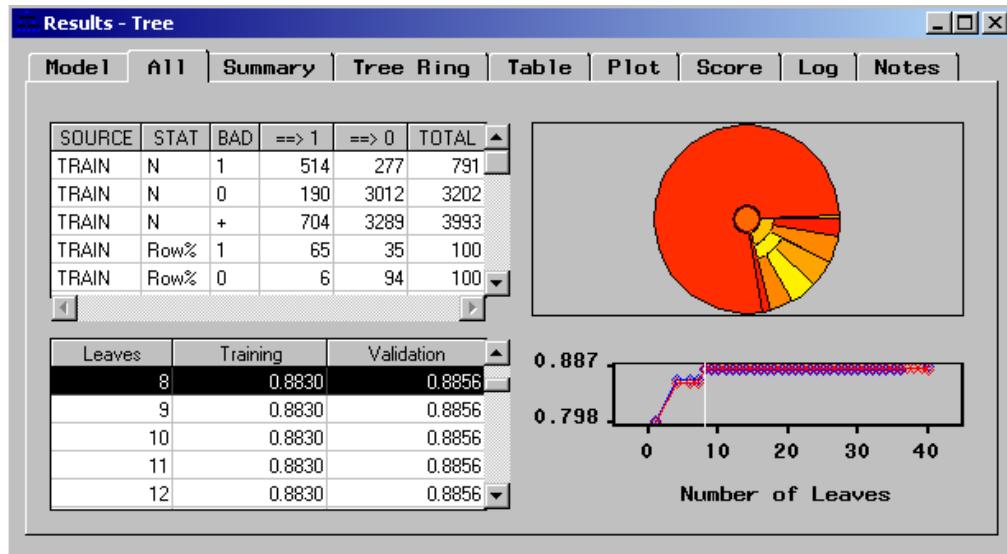
1. Open the Tree node.
2. Select the **Basic** tab.
3. Type **25** for the minimum number of observations in a leaf and then press the Enter key.
4. Type **50** for the number of observations required for a split search and then press the Enter key.

The Decision Tree node requires that (Observations required for a split search)  $\geq 2 * (\text{Minimum number of observations in a leaf})$ . In this example, the observations required for a split search must be greater than  $2 * 25 = 50$ . A node with fewer than 50 observations cannot be split into two nodes with each having at least 25 observations. If you specify numbers that violate this requirement, you will not be able to close the window.

5. Close and save your changes to the Tree node.

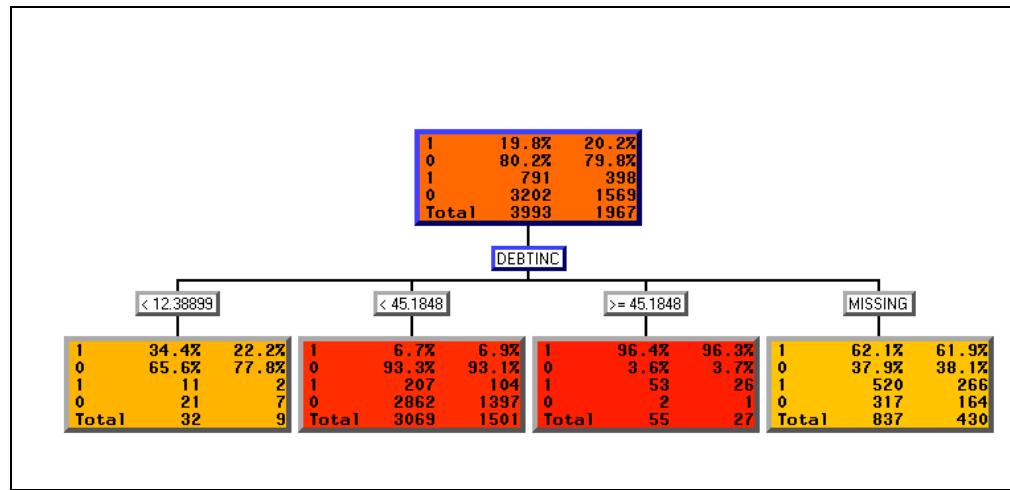
If the Tree node does not prompt you to save changes when you close, the settings have not been changed. Reopen the node and modify the settings again.

6. Rerun the Tree node and view the results as before.



The optimal tree now has 8 leaves. The validation accuracy has dropped slightly to 88.56%.

7. Select View  $\Rightarrow$  Tree to see the tree diagram.



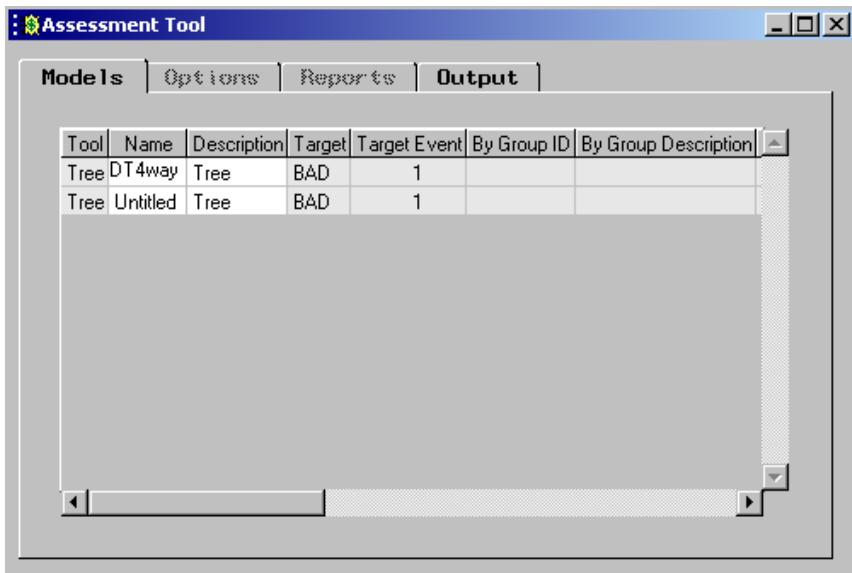
Note that the initial split on DEBTINC has produced four branches.

8. Close the tree diagram and results when you are finished viewing them.

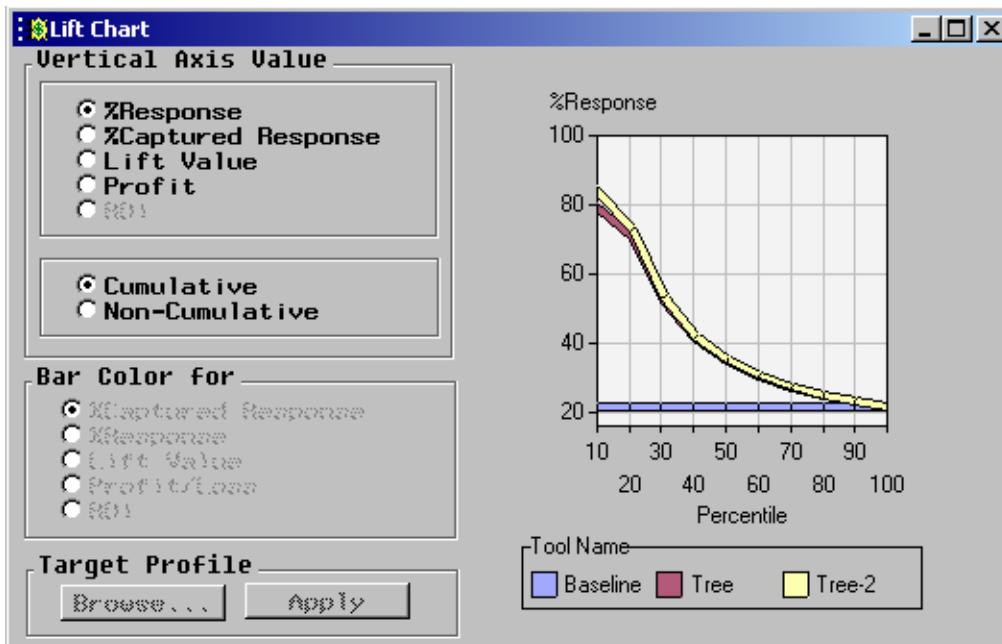
### Comparing Models

The Assessment node is useful for comparing models.

1. To run the diagram from the Assessment node, right-click on the Assessment node and select Run.
2. When prompted, select Yes to view the results.



3. In the Assessment Tool window, click and drag to select both of the models.
4. Select **Tools**  $\Rightarrow$  **Lift Chart**.



A Cumulative %Response chart is shown by default. By default, this chart arranges people into deciles based on their predicted probability of response, and then plots the actual percentage of respondents. To see actual values, click on the View Info tool and then click on one of the lines for the models. Clicking on the Tree-2 line near the upper-left corner of the plot indicates a %Response of 82.06, but what does that mean?

To interpret the Cumulative %Response chart, consider how the chart is constructed.

- For this example, a responder is defined as someone who defaulted on a loan (BAD=1). For each person, the fitted model (in this case, a decision tree) predicts

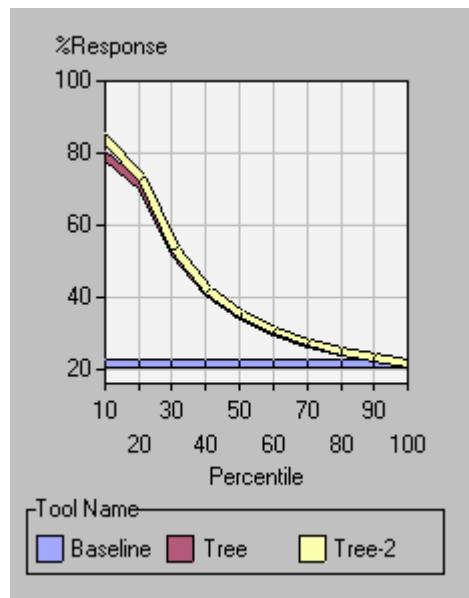
the probability that the person will default. Sort the observations by the predicted probability of response from the highest probability of response to the lowest probability of response.

- Group the people into ordered bins, each containing approximately 10% of the data in this case.
- Using the target variable BAD, count the percentage of actual responders in each bin.

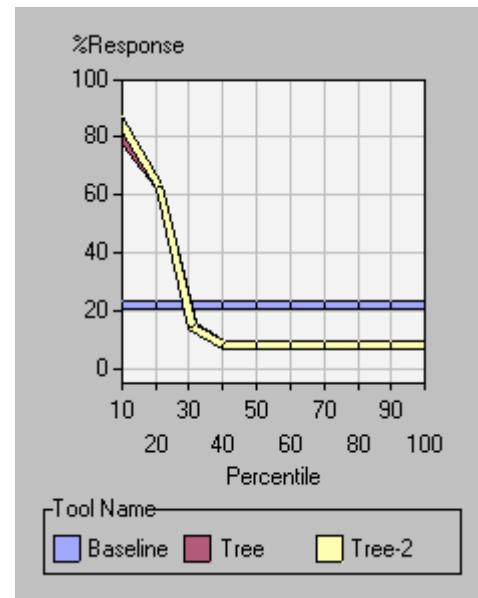
If the model is useful, the proportion of responders (defaulters) will be relatively high in bins where the predicted probability of response is high. The cumulative response curve shown above shows the percentage of respondents in the top 10%, top 20%, top 30%, and so on. In the top 10%, over 80% of the people were defaulters. In the top 20%, the proportion of defaulters has dropped to just over 72% of the people. The horizontal line represents the baseline rate (approximately 20%) for comparison purposes, which is an estimate of the percentage of defaulters that you would expect if you were to take a random sample. The plot above represents cumulative percentages, but you can also see the proportion of responders in each bin by selecting the radio button next to Non-Cumulative on the left side of the graph.

Select the radio button next to Non-Cumulative and inspect the plot.

*Cumulative %Response*

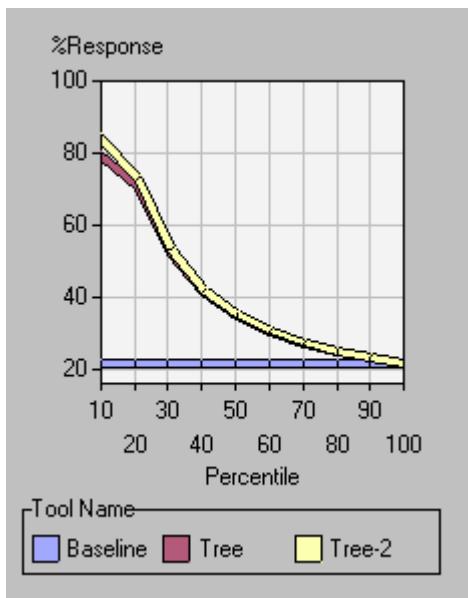
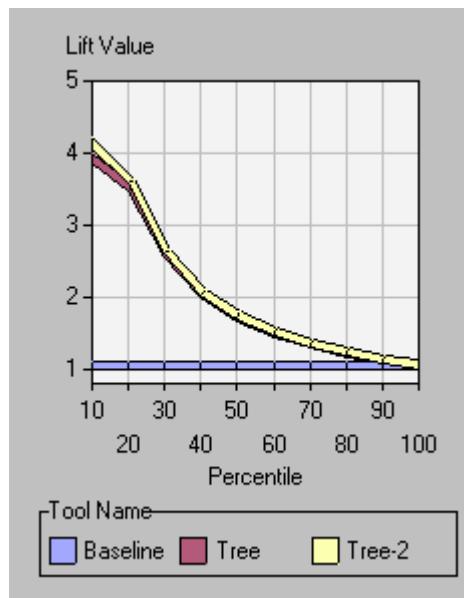


*Non-Cumulative %Response*



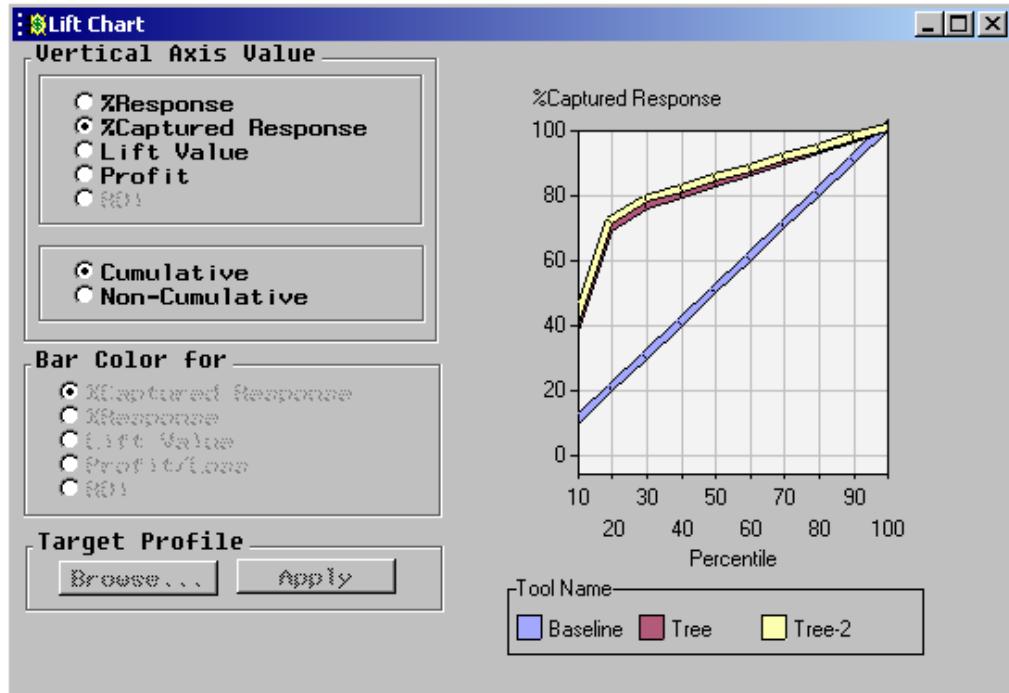
The Non-Cumulative chart shows that once you get beyond the 20<sup>th</sup> percentile for predicted probability, the default rate is lower than what you would expect if you were to take a random sample.

Select the **Cumulative** button and then select **Lift Value**. Lift charts plot the same information on a different scale. Recall that the population response rate is about 20%. A lift chart can be obtained by dividing the response rate in each percentile by the population response rate. The lift chart, therefore, plots relative improvement over baseline.

**Cumulative %Response****Cumulative Lift Value**

Recall that the percentage of defaulters in the top 10% was 82.06%. Dividing 82.06% by about 20% (baseline rate) results in a number slightly higher than 4, which indicates that you would expect to get over 4 times as many defaulters in this top group as you would from taking a simple random sample of the same size.

Instead of asking the question "What percentage of observations in a bin were responders?", you could ask the question "What percentage of the total number of responders are in a bin?" This can be evaluated using the Captured Response curve. To inspect this curve, select the radio button next to **%Captured Response**. Use the View Info tool to evaluate how the model performs.



Observe that if the percentage of applications chosen for rejection were approximately

- 20%, you would have identified about 70% of the people who would have defaulted (a lift of about 3.5!).
- 40%, you would have identified over 80% of the people who would have defaulted (a lift of over 2!).

Close the Lift Chart and Assessment Tool windows.

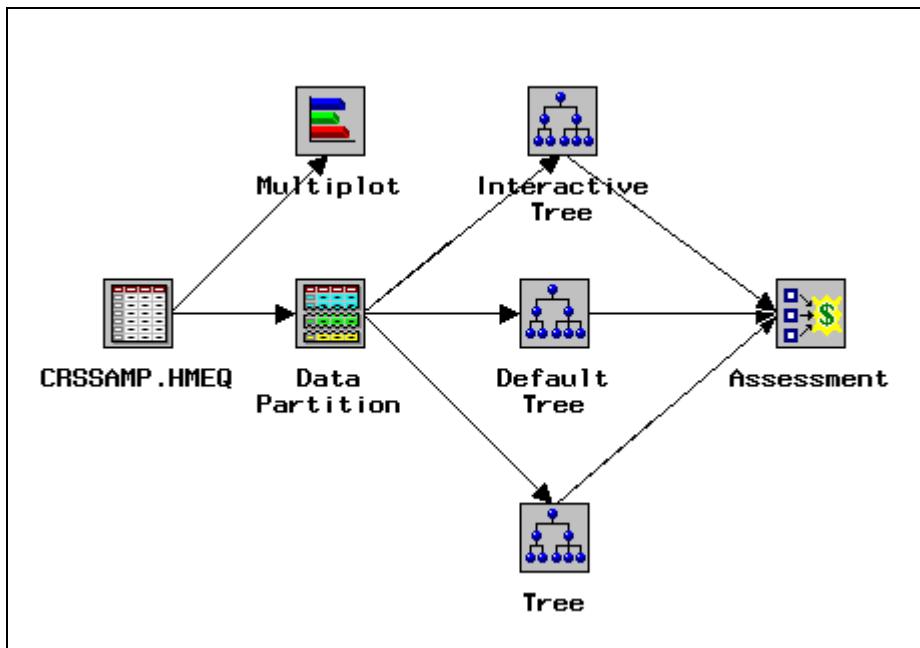
### Interactive Training

Decision tree splits are selected on the basis of an analytic criterion. Sometimes it is necessary or desirable to select splits on the basis of a practical business criterion. For example, the best split for a particular node may be on an input that is difficult or expensive to obtain. If a competing split on an alternative input has a similar worth and is cheaper and easier to obtain, it makes sense to use the alternative input for the split at that node.

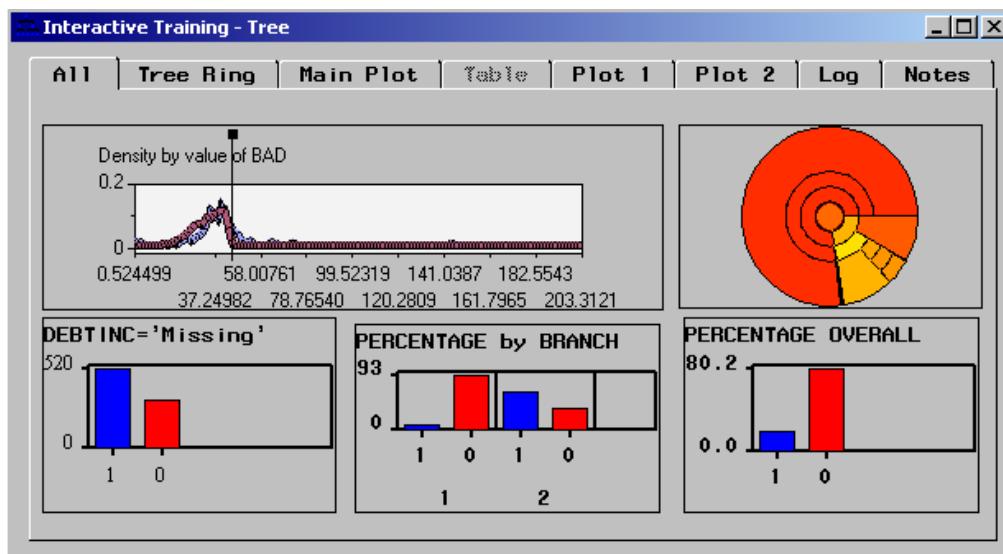
Likewise, splits may be selected that are statistically optimal but may be in conflict with an existing business practice. For example, the credit department may treat applications where debt-to-income ratios are not available differently from those where this information is available. You can incorporate this type of business rule into your decision tree using interactive training in the Tree node. It might then be

interesting to compare the statistical results of the original tree with the changed tree. In order to accomplish this, first make a copy of the Default Tree node.

1. Select the Default Tree node with the right mouse button and then select **Copy**.
2. Move your cursor to an empty place above the Default Tree node, right-click, and select **Paste**. Rename this node **Interactive Tree**.
3. Connect the Interactive Tree node to the Data Partition node and the Assessment node as shown.

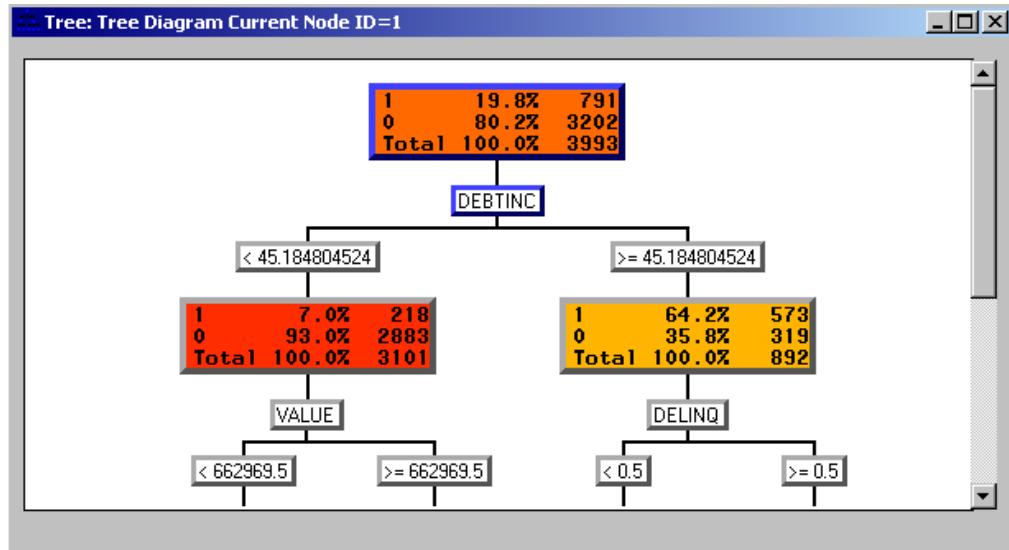


4. Right-click on the Interactive Tree node and select **Interactive...**. The Interactive Training window opens.



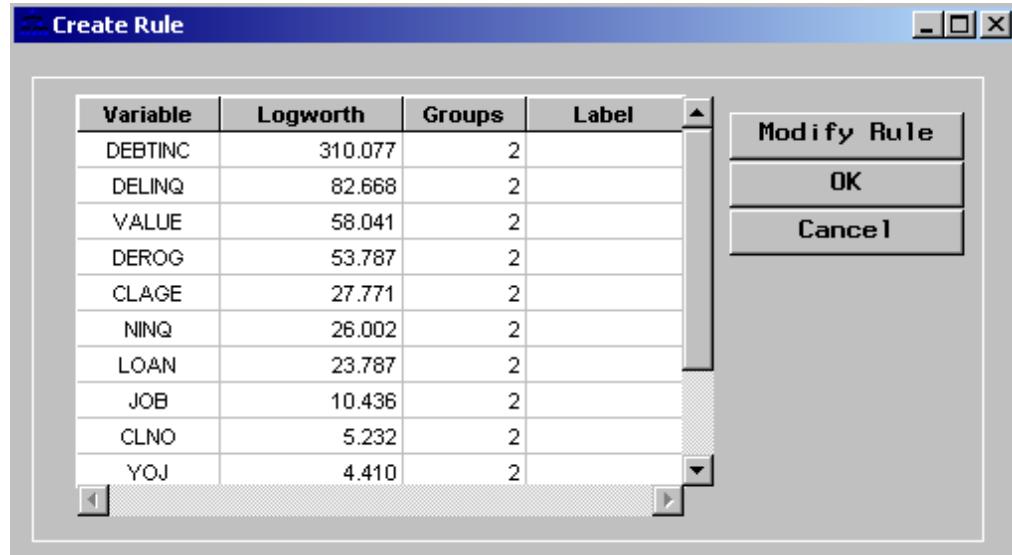
5. Select **View**  $\Rightarrow$  **Tree** from the menu bar.

The default decision tree is displayed.

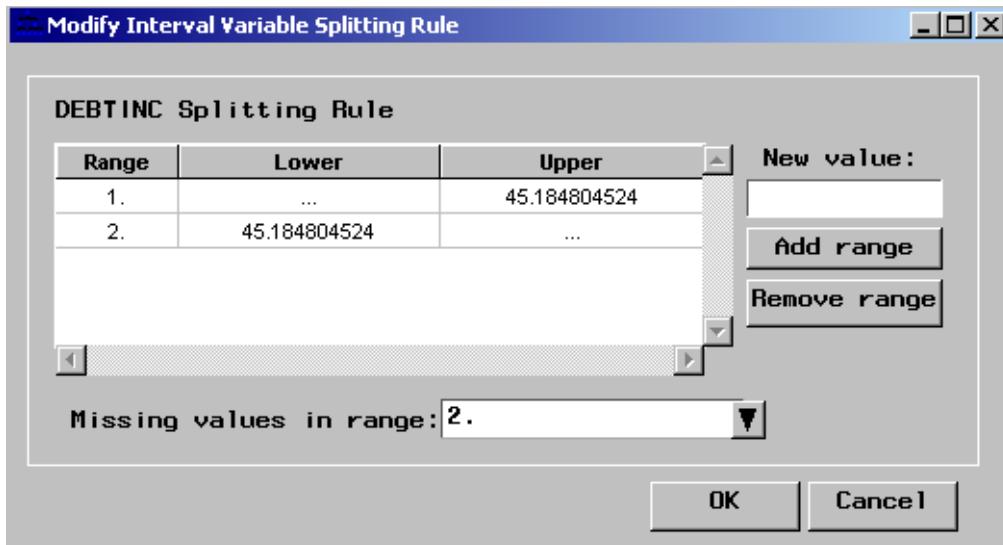


Your goal is to modify the initial split so that one branch contains all the applications with missing debt-to-income data and the other branch contains the rest of the applications. From this initial split, you will use the decision tree's analytic method to grow the remainder of the tree.

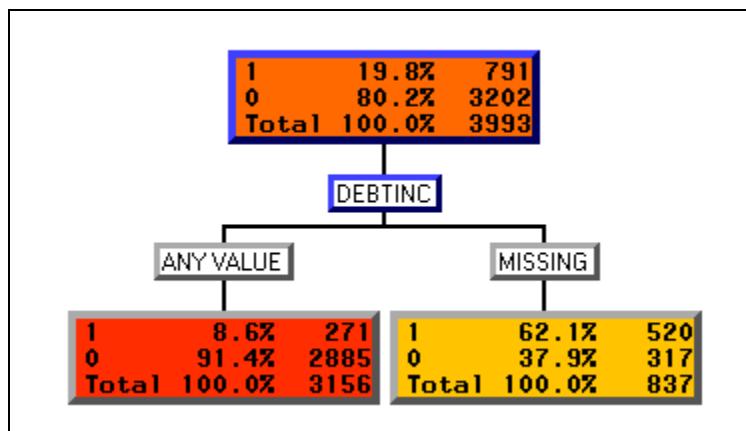
1. Select the Create Rule icon, , on the toolbar.
2. Select the root node of the tree. The Create Rule window opens listing potential splitting variables and a measure of the worth of each input.



3. Select the row corresponding to DEBTINC.
4. Select **Modify Rule**. The Modify Interval Variable Splitting Rule window opens.



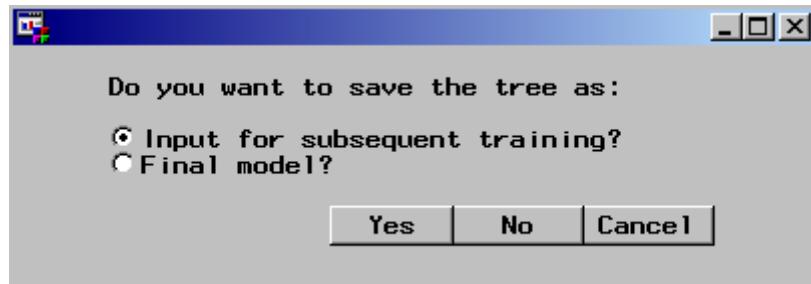
5. Select the row for range 2.
6. Select **Remove range**. The split is now defined to put all nonmissing values of DEBTINC into node 1 and all missing values of DEBTINC into node 2.
7. Select **OK** to close the Modify Interval Variable Splitting Rule window.
8. Select **OK** in the Create Rule window. The Create Rule window closes and the tree diagram is updated as shown.



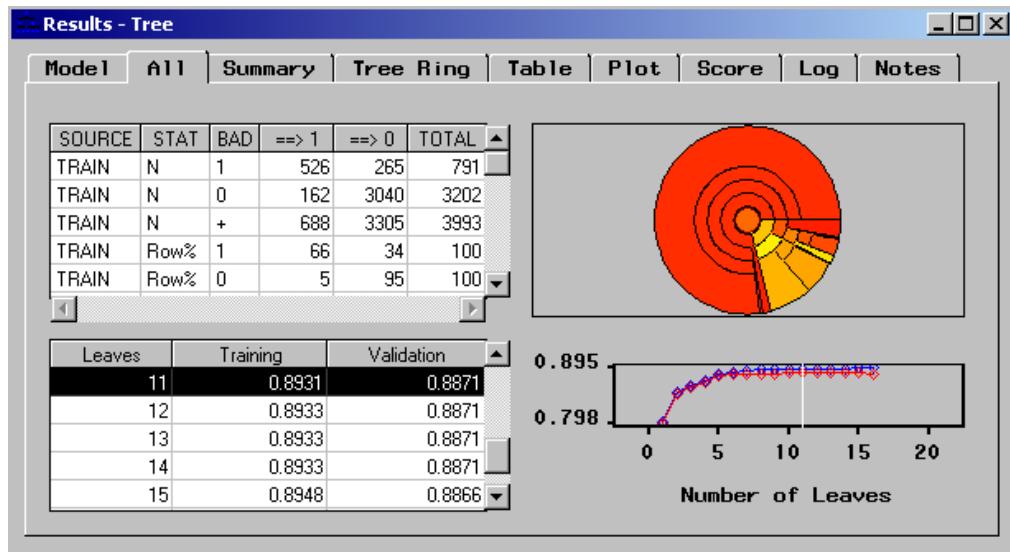
The left node contains any value of DEBTINC, and the right node contains only missing values for DEBTINC.

9. Close the tree diagram and the Interactive Training window.

10. Select **Yes** to save the tree as input for subsequent training.

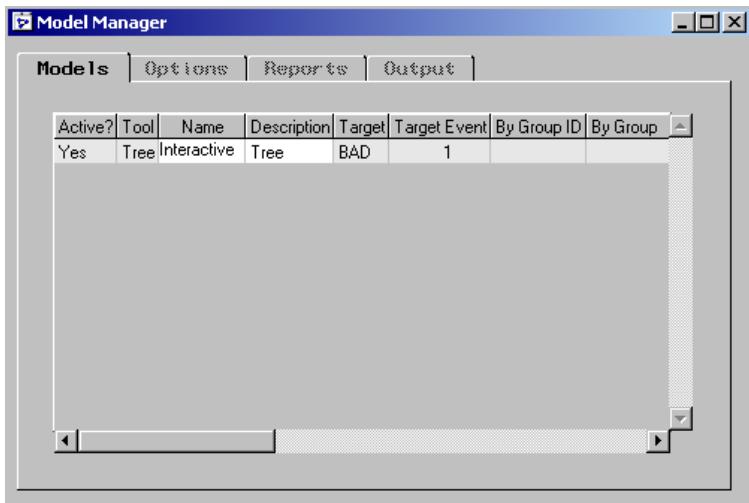


11. Run the modified Interactive Tree node and view the results. The selected tree has 11 nodes. Its validation accuracy is 88.71%.

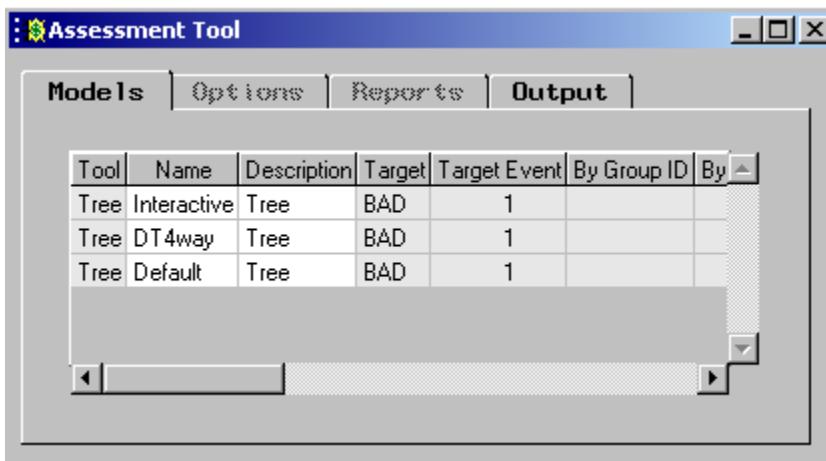


To compare the tree models:

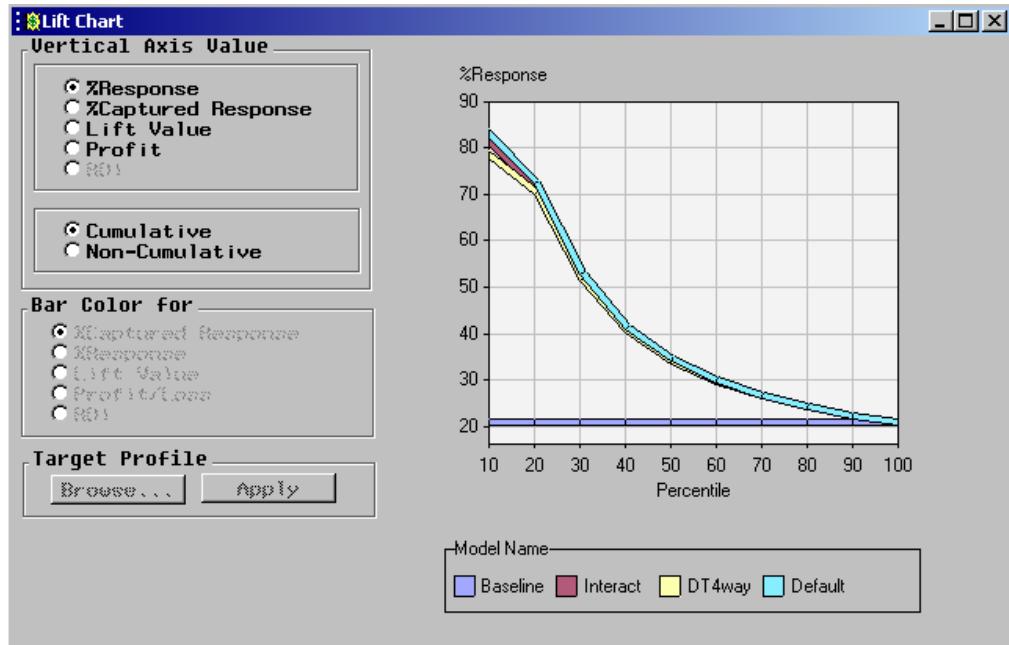
1. Close the Results window.
2. To rename the new model, right-click on the Interactive Tree node and select **Model Manager...**.
3. Change the name from Untitled to **Interactive**.



4. Close the Model Manager window. Right-click on the Assessment node and select **Results...**.
5. Enter **Default** as the name for the default tree model (currently Untitled).



6. Click and drag to select all three rows that correspond to the tree models.
  7. Select **Tools**  $\Rightarrow$  **Lift Chart**.
  8. Select **Format**  $\Rightarrow$  **Model Name**.
- You may have to maximize the window or resize the legend in order to see the entire legend.



The performance of the three tree models is not appreciably different. Close the lift chart when you are finished inspecting the results.

## Consequences of a Decision

	Decision 1	Decision 0
Actual 1	<i>True Positive</i>	<i>False Negative</i>
Actual 0	<i>False Positive</i>	<i>True Negative</i>

41

In order to choose the appropriate threshold to classify observations positively or negatively, the cost of misclassification must be considered. In the home equity line of credit example, you are modeling the probability of a default, which is coded as a 1. Therefore, Enterprise Miner sets up the profit matrix as shown above.

## Example

Recall the home equity line of credit scoring example. Presume that every two dollars loaned eventually returns three dollars if the loan is paid off in full.

42

Assume that every two dollars loaned returns three dollars if the borrower does not default. Rejecting a good loan for two dollars forgoes the expected dollar profit. Accepting a bad loan for two dollars forgoes the two-dollar loan itself (assuming that the default is early in the repayment period).

## Consequences of a Decision

	Decision 1	Decision 0
Actual 1	True Positive	False Negative (cost=\$2)
Actual 0	False Positive (cost=\$1)	True Negative

43

The costs of misclassification are shown in the table.

## Bayes Rule

$$\theta = \frac{1}{1 + \frac{\text{cost of false negative}}{\text{cost of false positive}}}$$

44

One way to determine the appropriate threshold is a theoretical approach. This approach uses the plug in Bayes rule. Using simple decision theory, the optimal threshold is given by  $\theta$ .

Using the cost structure defined for the home equity example, the optimal threshold is  $1/(1+(2/1)) = 1/3$ . That is, reject all applications whose predicted probability of default exceeds 0.33.

## Consequences of a Decision

	Decision 1	Decision 0
Actual 1	<i>True Positive (profit=\$2)</i>	<i>False Negative</i>
Actual 0	<i>False Positive (profit=\$-1)</i>	<i>True Negative</i>

45

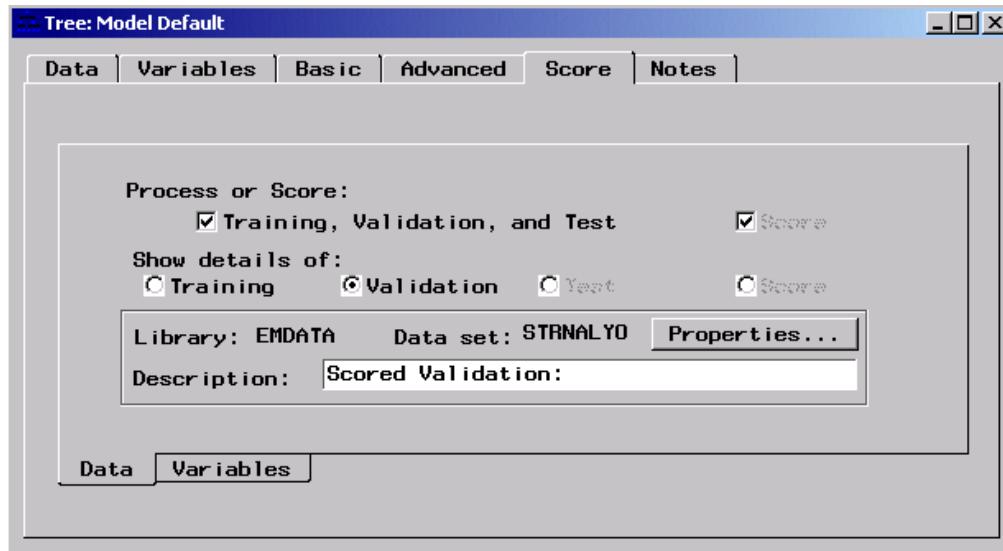
You can obtain the same result using the Assessment node in Enterprise Miner by using the profit matrix to specify the profit associated with the level of the response being modeled (in this case, a loan default or a 1). As a bonus, you can estimate the fraction of loan applications you must reject when using the selected threshold.



## Choosing a Decision Threshold

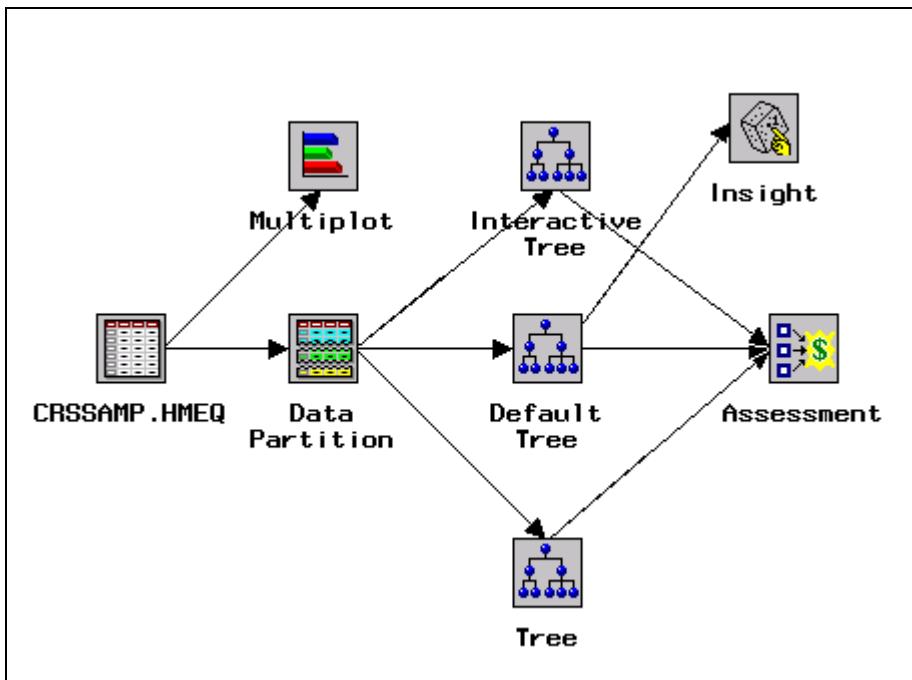
First, consider the decision threshold determined theoretically.

1. Return to the Project1 diagram, open the **Default Tree** node, and select the **Score** tab.
2. Check the box next to Training, Validation, and Test. This adds predicted values to the data sets.

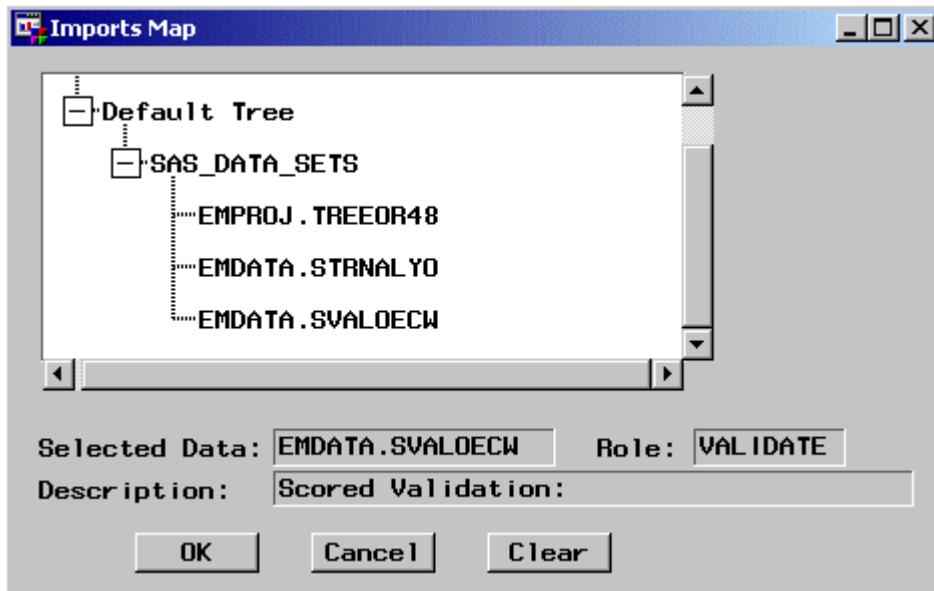


3. Close the tree node, saving changes when prompted.

4. Add an Insight node after the Default Tree node.



5. Open the Insight node.  
 6. In the Data tab, select the Select... button to see a list of predecessor data sets.  
 7. Choose the validation data set from the Default Tree node.



8. Select OK.  
 9. In the Data tab of the Insight Settings window, select the radio button next to Entire Data Set so that Insight will use the entire validation data set.

10. Close the node, saving changes when prompted.

11. Run the Insight node and view the results when prompted.

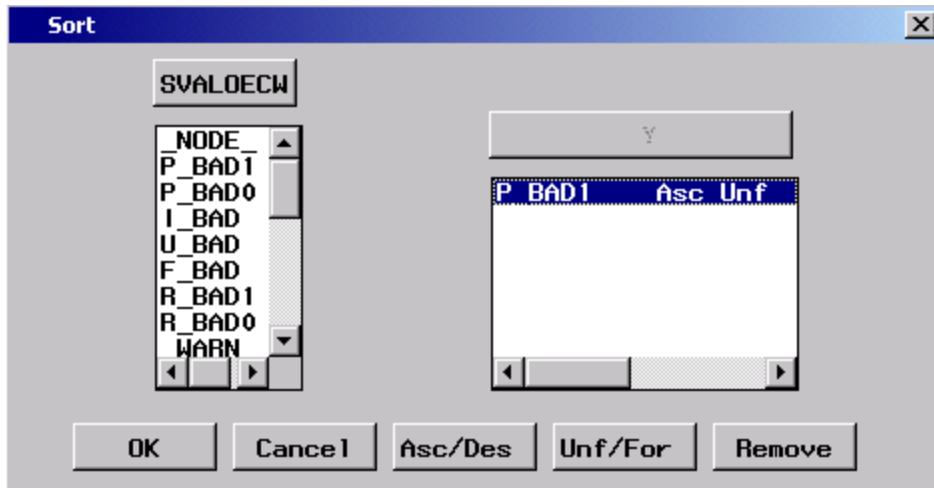
EMDATA.SVALOECW												
1967	22	Int	Int	Int	Nom	Int	Nom	Int	Int	R_BAD1	R_BAD0	
	NODE	P_BAD1	P_BAD0	I_BAD	U_BAD	F_BAD		R_BAD1	R_BAD0			
	1	7	0.8266	0.1734	1		1	1	0.1734	-0.1734		
	2	10	0.6471	0.3529	1		1	1	0.3529	-0.3529		
	3	8	0.0625	0.9375	0		0	1	0.9375	-0.9375		
	4	8	0.0625	0.9375	0		0	1	0.9375	-0.9375		
	5	7	0.8266	0.1734	1		1	1	0.1734	-0.1734		
	6	10	0.6471	0.3529	1		1	1	0.3529	-0.3529		
	7	7	0.8266	0.1734	1		1	1	0.1734	-0.1734		
	8	7	0.8266	0.1734	1		1	1	0.1734	-0.1734		
	9	7	0.8266	0.1734	1		1	1	0.1734	-0.1734		
	10	10	0.6471	0.3529	1		1	1	0.3529	-0.3529		
	11	10	0.6471	0.3529	1		1	1	0.3529	-0.3529		
	12	8	0.0625	0.9375	0		0	0	-0.0625	0.0625		
	13	7	0.8266	0.1734	1		1	1	0.1734	-0.1734		
	14	7	0.8266	0.1734	1		1	1	0.1734	-0.1734		
	15	7	0.8266	0.1734	1		1	1	0.1734	-0.1734		
	16	26	0.2857	0.7143	0		0	0	-0.2857	0.2857		
	17	10	0.6471	0.3529	1		1	0	-0.6471	0.6471		
	18	7	0.8266	0.1734	1		1	1	0.1734	-0.1734		
	19	10	0.6471	0.3529	1		1	1	0.3529	-0.3529		
	20	26	0.2857	0.7143	0		0	0	-0.2857	0.2857		
	21	8	0.0625	0.9375	0		0	0	-0.0625	0.0625		
	22	8	0.0625	0.9375	0		0	0	-0.0625	0.0625		
	23	7	0.8266	0.1734	1		1	1	0.1734	-0.1734		
	24	7	0.8266	0.1734	1		1	1	0.1734	-0.1734		
	25	8	0.0625	0.9375	0		0	0	-0.0625	0.0625		

One of the new variables in the data set is P\_BAD1, which is the predicted probability of a target value of 1 (a loan default). To sort the data set based on this variable:

12. Click on the triangle in the top left of the data table and select Sort....

EMDATA.SVALOECW												
196	22	Int	Int	Int	Nom	Int	Nom	Int	Int	R_BAD1	R_BAD0	
		BAD0	I_BAD	U_BAD	F_BAD		R_BAD1	R_BAD0				
		Find Next	1734	1		1	1	0.1734	-0.1734			
		Move to First	3529	1		1	1	0.3529	-0.3529			
		Move to Last	9375	0		0	1	0.9375	-0.9375			
		Sort...	9375	0		0	1	0.9375	-0.9375			
			1734	1		1	1	0.1734	-0.1734			
			3529	1		1	1	0.3529	-0.3529			
		New Observations	1734	1		1	1	0.1734	-0.1734			
		New Variables	1734	1		1	1	0.1734	-0.1734			
		Define Variables...	3529	1		1	1	0.3529	-0.3529			
		Fill Values...	3529	1		1	1	0.3529	-0.3529			
		Extract	9375	0		0	0	-0.0625	0.0625			
			1734	1		1	1	0.1734	-0.1734			
			1734	1		1	1	0.1734	-0.1734			
		Data Options...	1734	1		1	1	0.1734	-0.1734			
	16	26	0.2857	0.7143	0		0	0	-0.2857	0.2857		
	17	10	0.6471	0.3529	1		1	0	-0.6471	0.6471		
	18	7	0.8266	0.1734	1		1	1	0.1734	-0.1734		
	19	10	0.6471	0.3529	1		1	1	0.3529	-0.3529		
	20	26	0.2857	0.7143	0		0	0	-0.2857	0.2857		
	21	8	0.0625	0.9375	0		0	0	-0.0625	0.0625		
	22	8	0.0625	0.9375	0		0	0	-0.0625	0.0625		
	23	7	0.8266	0.1734	1		1	1	0.1734	-0.1734		
	24	7	0.8266	0.1734	1		1	1	0.1734	-0.1734		
	25	8	0.0625	0.9375	0		0	0	-0.0625	0.0625		

13. In the Sort window, select P\_BAD1  $\Rightarrow$  Y.



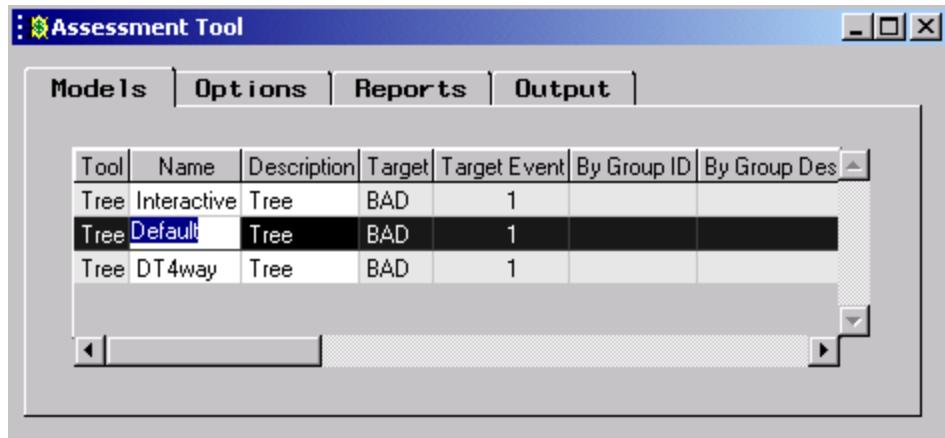
14. Highlight **P\_BAD1** in the Y column and select **Asc/Des** to sort in descending order.
15. Select **OK**.
16. Scroll through the data table and note that 380 of the observations have a predicted probability of default greater than 1/3.

	22	Int	Int	Int	Nom	Int	Nom	Int	I
	1967	NODE	P_BAD1	P_BAD0	I_BAD	U_BAD	F_BAD	R_BAD1	R_BA
■	373	10	0.6471	0.3529	1	1	1	0.3529	-.35
■	374	10	0.6471	0.3529	1	1	1	0.3529	-.35
■	375	10	0.6471	0.3529	1	1	1	0.3529	-.35
■	376	10	0.6471	0.3529	1	1	1	0.3529	-.35
■	377	10	0.6471	0.3529	1	1	1	0.3529	-.35
■	378	10	0.6471	0.3529	1	1	1	0.3529	-.35
■	379	10	0.6471	0.3529	1	0	1	-.6471	0.64
■	380	10	0.6471	0.3529	1	0	1	-.6471	0.64
■	381	26	0.2857	0.7143	0	0	0	-.2857	0.28
■	382	26	0.2857	0.7143	0	1	0	0.7143	-.71
■	383	26	0.2857	0.7143	0	0	0	-.2857	0.28
■	384	26	0.2857	0.7143	0	0	0	-.2857	0.28
■	385	26	0.2857	0.7143	0	0	0	-.2857	0.28
■	386	26	0.2857	0.7143	0	0	0	-.2857	0.28
■	387	26	0.2857	0.7143	0	0	0	-.2857	0.28
■	388	26	0.2857	0.7143	0	1	0	0.7143	-.71
■	389	26	0.2857	0.7143	0	0	0	-.2857	0.28

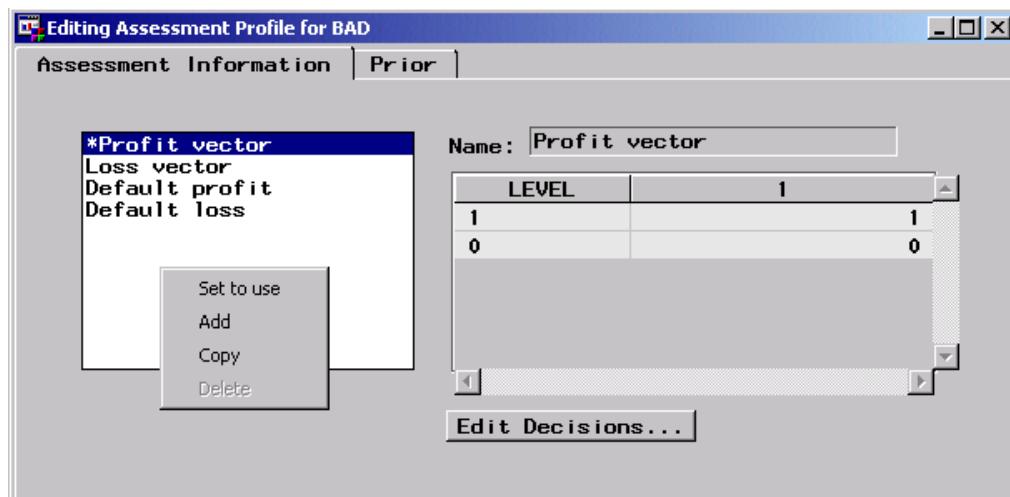
Therefore, based on the theoretical approach, 380 out of 1967 applications, or approximately 19%, should be rejected.

You can obtain the same result using the Assessment node.

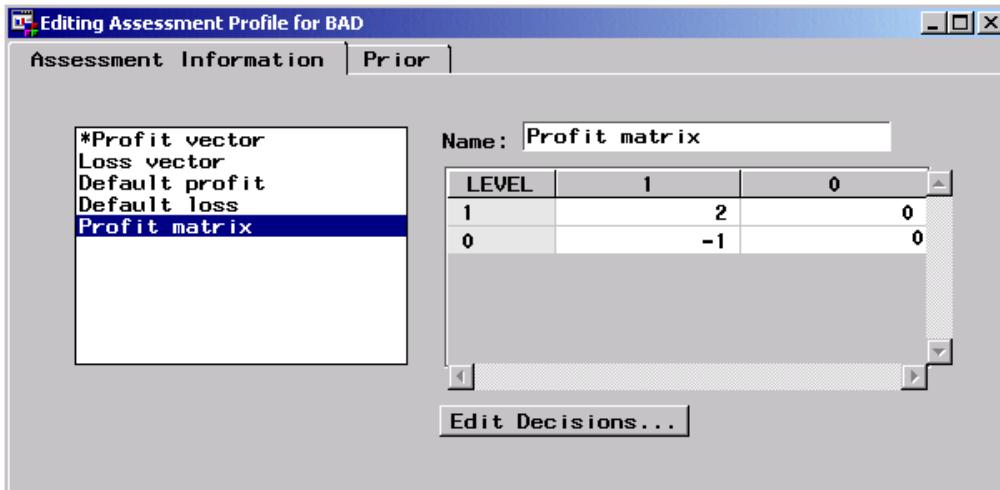
1. Close the Insight data table.
2. Right-click on the Assessment node and select **Results...**
3. Select the default model in the Assessment node.



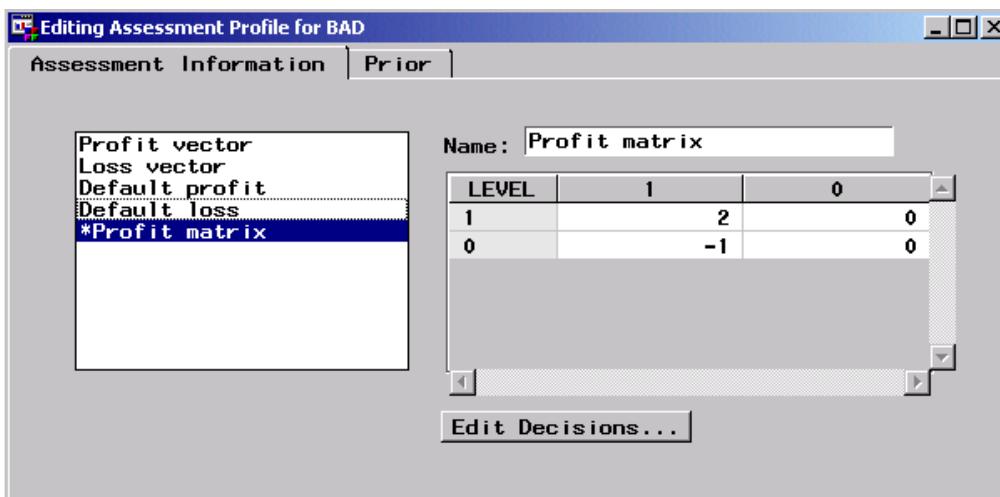
4. Select Tools  $\Rightarrow$  Lift Chart from the menu bar.
5. In the bottom-left corner of the Lift Chart window, select Edit... to define a target profile.
6. In the Editing Assessment Profile for BAD window, right-click in the open area where the vectors and matrices are listed and select Add.



7. Highlight the new Profit matrix, and enter the values in the matrix as pictured below.



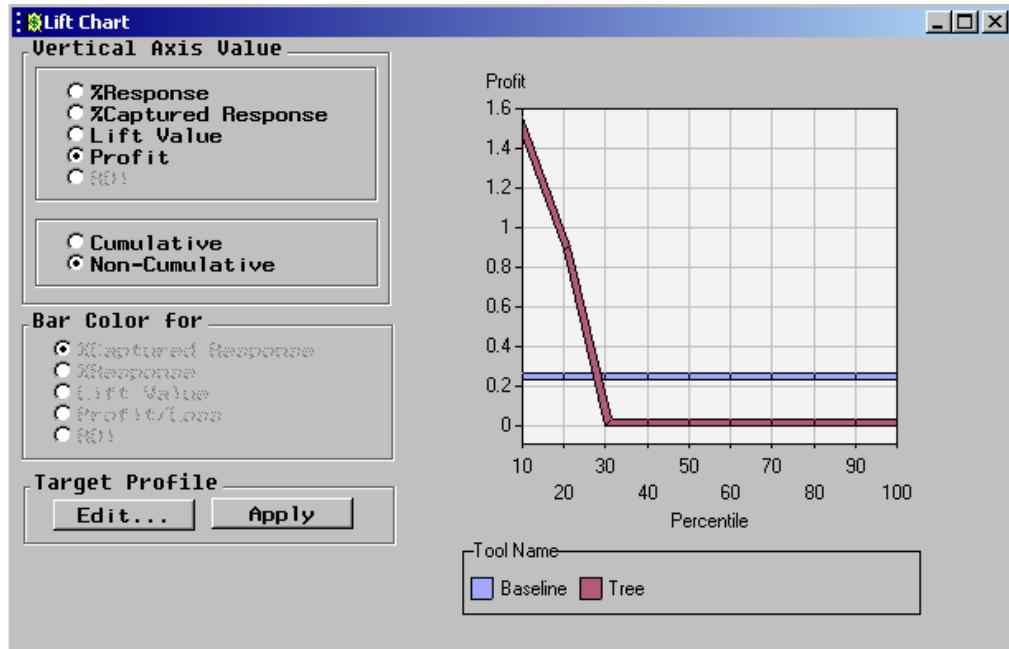
- For credit screening, a target value of 1 implies a default and, thus, a loss. A target value of 0 implies a paid repaid loan and, thus, a profit. The fixed cost of processing each loan application is insubstantial and taken to be zero.
8. Right-click on the Profit matrix and select Set to Use. The profit matrix is now active as indicated by the asterisk next to it.



9. Close the Profit Matrix Definition window, saving changes when prompted.
10. Select Apply.



11. Select the Profit radio button.
12. Select the Non-Cumulative radio button.



The plot shows the actual profit for each percentile of loan applications as ranked by the decision tree model. Percentiles up to and including the 20<sup>th</sup> percentile show a profit for rejecting the applicants. Therefore, it makes sense to reject the top 20% of loan applications. This agrees with the results obtained theoretically.

- In Enterprise Miner, the Non-Cumulative profit chart never dips below zero. This is because a cutoff value is chosen and there is no cost below this level because there is no action. As a result, the cumulative profit chart can be misleading.

# Chapter 3 Predictive Modeling Using Regression

<b>3.1 Introduction to Regression.....</b>	<b>3-3</b>
<b>3.2 Regression in Enterprise Miner .....</b>	<b>3-8</b>



## 3.1 Introduction to Regression

### Objectives

- Describe linear and logistic regression.
- Explore data issues associated with regression.
- Discuss variable selection methods.

## Linear versus Logistic Regression

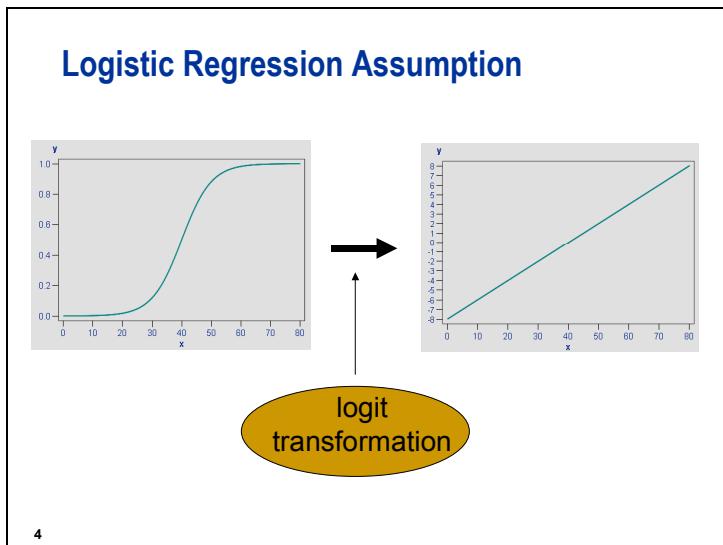
Linear Regression	Logistic Regression
Target is an interval variable.	Target is a discrete (binary or ordinal) variable.
Input variables have any measurement level.	Input variables have any measurement level.
Predicted values are the mean of the target variable at the given values of the input variables.	Predicted values are the probability of a particular level(s) of the target variable at the given values of the input variables.

3

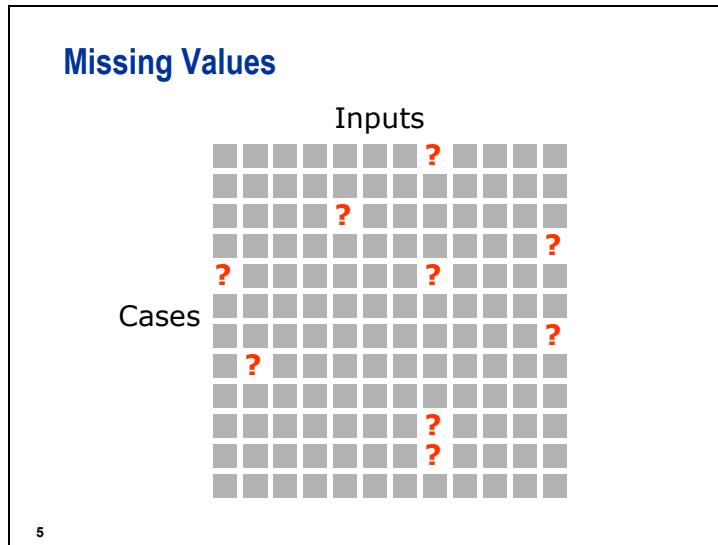
The Regression node in Enterprise Miner does either linear or logistic regression depending upon the measurement level of the target variable.

Linear regression is done if the target variable is an interval variable. In linear regression the model predicts the mean of the target variable at the given values of the input variables.

Logistic regression is done if the target variable is a discrete variable. In logistic regression the model predicts the probability of a particular level(s) of the target variable at the given values of the input variables. Because the predictions are probabilities, which are bounded by 0 and 1 and are not linear in this space, the probabilities must be transformed in order to be adequately modeled. The most common transformation for a binary target is the logit transformation. Probit and complementary log-log transformations are also available in the regression node.



Recall that one assumption of logistic regression is that the logit transformation of the probabilities of the target variable results in a linear relationship with the input variables.



Regression uses only full cases in the model. This means that any case, or observation, that has a missing value will be excluded from consideration when building the model. As discussed earlier, when there are many potential input variables to be considered, this could result in an unacceptably high loss of data. Therefore, when possible, missing values should be imputed prior to running a regression model.

Other reasons for imputing missing values include the following:

- Decision trees handle missing values directly, whereas regression and neural network models ignore all observations with missing values on any of the input variables. It is more appropriate to compare models built on the same set of observations. Therefore, before doing a regression or building a neural network model, you should perform data replacement, particularly if you plan to compare the results to results obtained from a decision tree model.
- If the missing values are in some way related to each other or to the target variable, the models created without those observations may be biased.
- If missing values are not imputed during the modeling process, observations with missing values cannot be scored with the score code built from the models.

## Stepwise Selection Methods



Forward Selection



Backward Selection



Stepwise Selection

6

There are three variable selection methods available in the Regression node of Enterprise Miner.

**Forward** first selects the best one-variable model. Then it selects the best two variables among those that contain the first selected variable. This process continues until it reaches the point where no additional variables have a  $p$ -value less than the specified entry  $p$ -value.

**Backward** starts with the full model. Next, the variable that is least significant, given the other variables, is removed from the model. This process continues until all of the remaining variables have a  $p$ -value less than the specified stay  $p$ -value.

**Stepwise** is a modification of the forward selection method. The difference is that variables already in the model do not necessarily stay there. After each variable is entered into the model, this method looks at all the variables already included in the model and deletes any variable that is not significant at the specified level. The process ends when none of the variables outside the model has a  $p$ -value less than the specified entry value and every variable in the model is significant at the specified stay value.



The specified  $p$ -values are also known as *significance levels*.

## 3.2 Regression in Enterprise Miner

### Objectives

- Conduct missing value imputation.
- Examine transformations of data.
- Generate a regression model.



## Imputation, Transformation, and Regression

The data for this example is from a nonprofit organization that relies on fundraising campaigns to support their efforts. After analyzing the data, a subset of 19 predictor variables was selected to model the response to a mailing. Two response variables were stored in the data set. One response variable related to whether or not someone responded to the mailing (TARGET\_B), and the other response variable measured how much the person actually donated in U.S. dollars (TARGET\_D).

Name	Model Role	Measurement Level	Description
AGE	Input	Interval	Donor's age
AVGGIFT	Input	Interval	Donor's average gift
CARDGIFT	Input	Interval	Donor's gifts to card promotions
CARDPROM	Input	Interval	Number of card promotions
FEDGOV	Input	Interval	% of household in federal government
FIRSTT	Input	Interval	Elapsed time since first donation
GENDER	Input	Binary	F=female, M=Male
HOMEOWNR	Input	Binary	H=homeowner, U=unknown
IDCODE	ID	Nominal	ID code, unique for each donor
INCOME	Input	Ordinal	Income level (integer values 0-9)
LASTT	Input	Interval	Elapsed time since last donation
LOCALGOV	Input	Interval	% of household in local government
MALEMILI	Input	Interval	% of household males active in the military
MALEVET	Input	Interval	% of household male veterans
NUMPROM	Input	Interval	Total number of promotions
PCOWNERS	Input	Binary	Y=donor owns computer (missing otherwise)
PETS	Input	Binary	Y=donor owns pets (missing otherwise)
STATEGOV	Input	Interval	% of household in state government
TARGET_B	Target	Binary	1=donor to campaign, 0=did not contribute

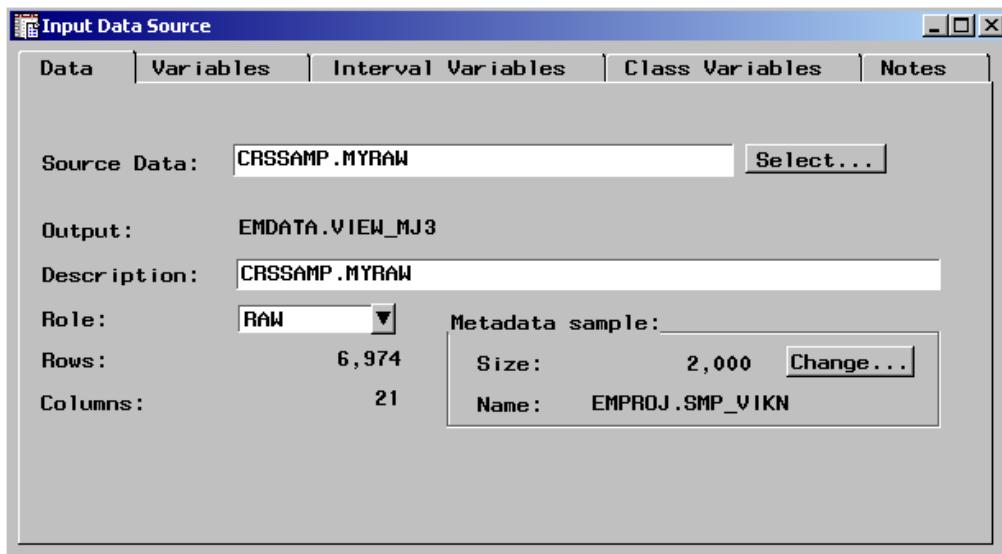
TARGET_D	Target	Interval	Dollar amount of contribution to campaign
TIMELAG	Input	Interval	Time between first and second donation

- ✍ The variable TARGET\_D is not considered in this chapter, so its model role will be set to **Rejected**.
- ✍ A card promotion is one where the charitable organization sends potential donors an assortment of greeting cards and requests a donation for them.

The MYRAW data set in the CRSSAMP library contains 6,974 observations for building and comparing competing models. This data set will be split equally into training and validation data sets for analysis.

### Building the Initial Flow and Identifying the Input Data

1. Open a new diagram by selecting **File** ⇒ **New** ⇒ **Diagram**.
2. On the Diagrams subtab, name the new diagram by right-clicking on **Untitled** and selecting **Rename**.
3. Name the new diagram **Non-Profit**.
4. Add an **Input Data Source** node to the diagram workspace by dragging the node from the toolbar or from the Tools tab.
5. Add a **Data Partition** node to the diagram and connect it to the Input Data Source node.
6. To specify the input data, double-click on the **Input Data Source** node.
7. Click on **Select...** in order to choose the data set.
8. Click on the and select **CRSSAMP** from the list of defined libraries.
9. Select the **MYRAW** data set from the list of data sets in the CRSSAMP library and then select **OK**.



Observe that this data set has 6,974 observations (rows) and 21 variables (columns). Evaluate (and update, if necessary) the assignments that were made using the metadata sample.

1. Click on the **Variables** tab to see all of the variables and their respective assignments.
2. Click on the **Name** column heading to sort the variables by their name. A portion of the table showing the first 10 variables is shown below.

The screenshot shows the 'Variables' tab of the 'Input Data Source' dialog box. The table lists the following variables:

Name	Model Role	Measurement	Type
AGE	input	interval	num
AVGGIFT	input	interval	num
CARDGIFT	input	interval	num
CARDPROM	input	interval	num
FEDGOV	input	interval	num
FIRSTT	input	interval	num
GENDER	input	binary	char
HOMEOWNR	input	binary	char
IDCODE	id	nominal	char
INCOME	input	ordinal	num

The first several variables (AGE through FIRSTT) have the measurement level **interval** because they are numeric in the data set and have more than 10 distinct levels in the metadata sample. The model role for all **interval** variables is set to **input** by default. The variables GENDER and HOMEOWNR have the measurement level **binary** because they have only two different nonmissing levels in the metadata sample. The model role for all **binary** variables is set to **input** by default.

The variable IDCODE is listed as a nominal variable because it is a character variable with more than two nonmissing levels in the metadata sample. Furthermore, because it is nominal and the number of distinct values is at least 2000 or greater than 90% of the sample size, the IDCODE variable has the model role **id**. If the ID value had been stored as a number, it would have been assigned an **interval** measurement level and an **input** model role.

The variable INCOME is listed as an ordinal variable because it is a numeric variable with more than two but no more than ten distinct levels in the metadata sample. All ordinal variables are set to have the **input** model role.

Scroll down to see the rest of the variables.

Name	Model Role	Measurement	Type	Format
LASTT	input	interval	num	BEST1
LOCALGOV	input	interval	num	BEST1
MALEMIL1	input	interval	num	BEST1
MALEVET	input	interval	num	BEST1
NUMPROM	input	interval	num	BEST1
PCOWNERS	rejected	unary	char	\$8.
PETS	rejected	unary	char	\$8.
STATEGOV	input	interval	num	BEST1
TARGET_B	input	binary	num	BEST1
TARGET_D	input	interval	num	BEST1
TIMELAG	input	interval	num	BEST1

The variables PCOWNERS and PETS both are identified as **unary** for their measurement level. This is because there is only one nonmissing level in the metadata sample. It does not matter in this case whether the variable was character or numeric, the measurement level is set to **unary** and the model role is set to **rejected**.

These variables do have useful information, however, and it is the way in which they are coded that makes them seem useless. Both variables contain the value **Y** for a person if the person has that condition (pet owner for PETS, computer owner for PCOWNERS) and a missing value otherwise. Decision trees handle missing values directly, so no data modification needs to be done for fitting a decision tree; however, neural networks and regression models ignore any observation with a missing value, so you will need to recode these variables to get at the desired information. For example, you can recode the missing values as a **U**, for unknown. You do this later using the Replacement node.

### Identifying Target Variables

Note that the variables TARGET\_B and TARGET\_D are the response variables for this analysis. TARGET\_B is binary even though it is a numeric variable since there are only two non-missing levels in the metadata sample. TARGET\_D has the interval measurement level. Both variables are set to have the **input** model role (just like

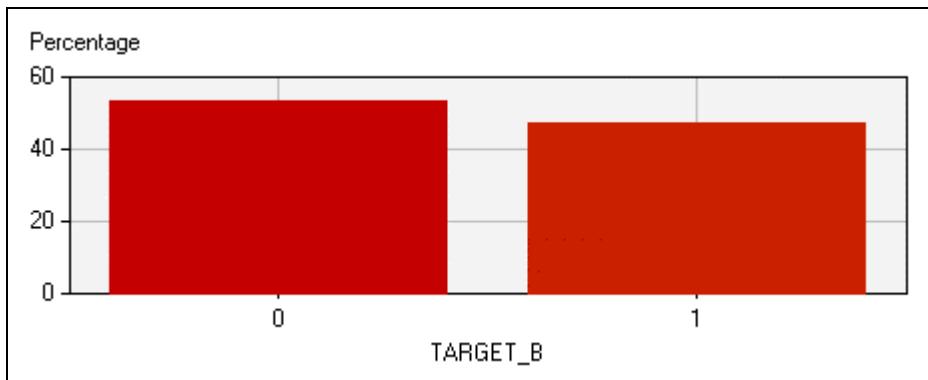
any other binary or interval variable). This analysis will focus on TARGET\_B, so you need to change the model role for TARGET\_B to **target** and the model role TARGET\_D to **rejected** because you should not use a response variable as a predictor.

1. Right-click in the Model Role column of the row for TARGET\_B.
2. Select **Set Model Role**  $\Rightarrow$  **target** from the pop-up menu.
3. Right-click in the Model Role column of the row for TARGET\_D.
4. Select **Set Model Role**  $\Rightarrow$  **rejected** from the pop-up menu.

### Inspecting Distributions

You can inspect the distribution of values in the metadata sample for each of the variables. To view the distribution of TARGET\_B:

1. Right-click in the name column of the row for TARGET\_B.
2. Select **View distribution of TARGET\_B**.



Investigate the distribution of the unary variables, PETS and PCOWNERS. What percentage of the observations have pets? What percentage of the observations own personal computers? Recall that these distributions depend on the metadata sample. The numbers may be slightly different if you refresh your metadata sample; however, these distributions are only being used for a quick overview of the data.

Evaluate the distribution of other variables as desired. For example, consider the distribution of INCOME. Some analysts would assign the interval measurement level to this variable. If this were done and the distribution was highly skewed, a transformation of this variable may lead to better results.

## Modifying Variable Information

Earlier you changed the model role for TARGET\_B to **target**. Now modify the model role and measurement level for PCOWNERS and PETS.

1. Click and drag to select the rows for PCOWNERS and PETS.
2. Right-click in the Model Role column for one of these variables and select **Set Model Role**  $\Leftrightarrow$  **input** from the pop-up menu.
3. Right-click in the measurement column for one of these variables and select **Set Measurement**  $\Leftrightarrow$  **binary** from the pop-up menu.

## Understanding the Target Profiler for a Binary Target

When building predictive models, the "best" model often varies according to the criteria used for evaluation. One criterion might suggest that the best model is the one that most accurately predicts the response. Another criterion might suggest that the best model is the one that generates the highest expected profit. These criteria can lead to quite different results.

In this analysis, you are analyzing a binary variable. The accuracy criteria would choose the model that best predicts whether someone actually responded; however, there are different profits and losses associated with different types of errors.

Specifically, it costs less than a dollar to send someone a mailing, but you receive a median of \$13.00 from those that respond. Therefore, to send a mailing to someone that would not respond costs less than a dollar, but failing to mail to someone that would have responded costs over \$12.00 in lost revenue.

-  In the example shown here, the median is used as the measure of central tendency. In computing expected profit, it is theoretically more appropriate to use the mean.

In addition to considering the ramifications of different types of errors, it is important to consider whether or not the sample is representative of the population. In your sample, almost 50% of the observations represent responders. In the population, however, the response rate was much closer to 5% than 50%. In order to obtain appropriate predicted values, you must adjust these predicted probabilities based on the prior probabilities. In this situation, accuracy would yield a very poor model because you would be correct approximately 95% of the time in concluding that nobody will respond. Unfortunately, this does not satisfactorily solve your problem of trying to identify the "best" subset of a population for your mailing.

-  In the case of rare target events, it is not uncommon to oversample. This is because you tend to get better models when they are built on a data set that is more balanced with respect to the levels of the target variable.

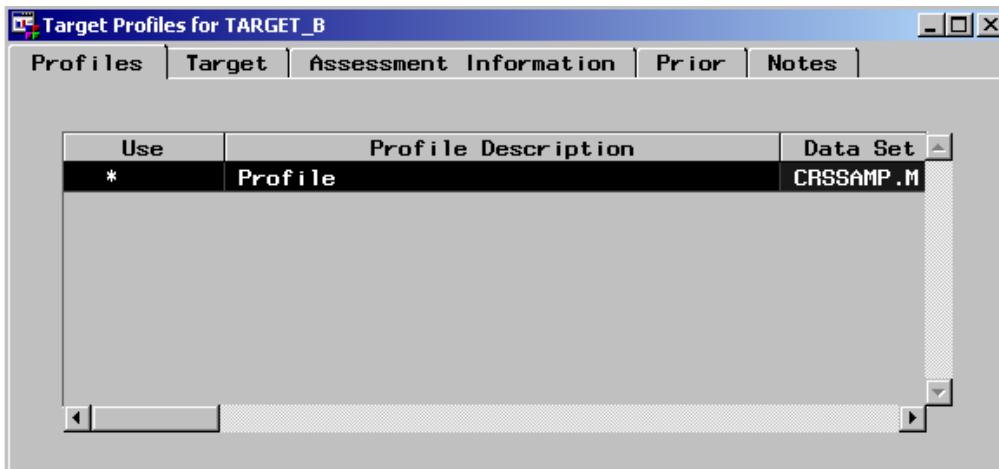
## Using the Target Profiler

When building predictive models, the choice of the "best" model depends on the criteria you use to compare competing models. Enterprise Miner allows you to specify information about the target that can be used to compare competing models. To generate a target profile for a variable, you must have already set the model role

for the variable to *target*. This analysis focuses on the variable TARGET\_B. To set up the target profile for this TARGET\_B, proceed as follows:

1. Right-click over the row for TARGET\_B and select **Edit target profile...**.
2. When the message stating that no target profile was found appears, select **Yes** to create the profile.

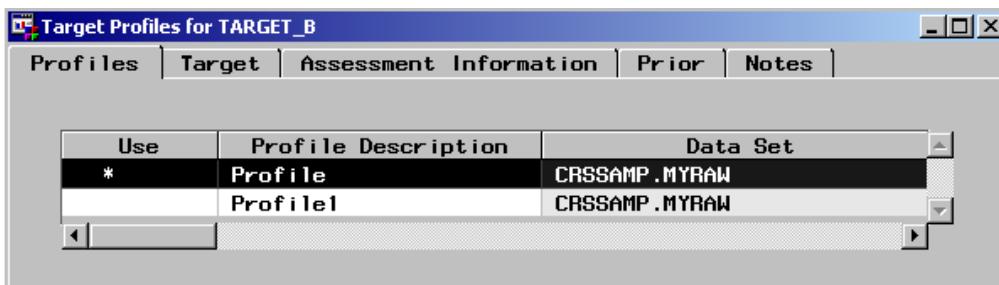
The target profiler opens with the Profiles tab active. You can use the default profile or you can create your own.



The screenshot shows the 'Target Profiles for TARGET\_B' dialog box. The 'Profiles' tab is selected. A table lists a single profile:

Use	Profile Description	Data Set
*	Profile	CRSSAMP.M

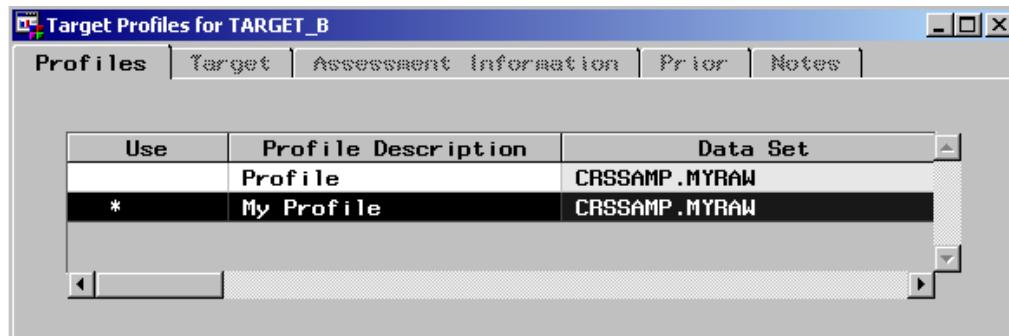
3. Select **Edit**  $\Rightarrow$  **Create New Profile** to create a new profile.



The screenshot shows the 'Target Profiles for TARGET\_B' dialog box. The 'Profiles' tab is selected. A table lists two profiles:

Use	Profile Description	Data Set
*	Profile	CRSSAMP.MYRAW
	Profile1	CRSSAMP.MYRAW

4. Type **My Profile** as the description for this new profile (currently named Profile1).
5. To set the newly created profile for use, position your cursor in the row corresponding to your new profile in the Use column and right-click.
6. Select **Set to use**.



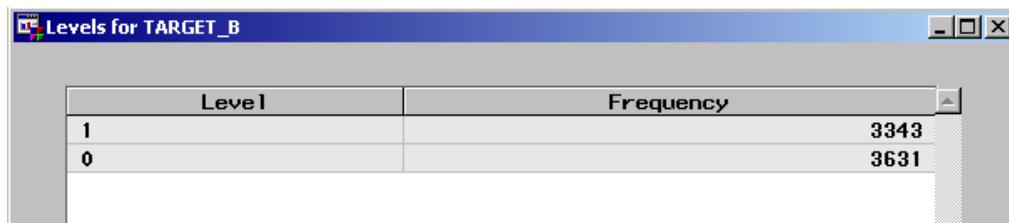
The values stored in the remaining tabs of the target profiler may vary according to which profile is selected. Make sure that your new profile is selected before examining the remainder of the tabs.

7. Select the Target tab.

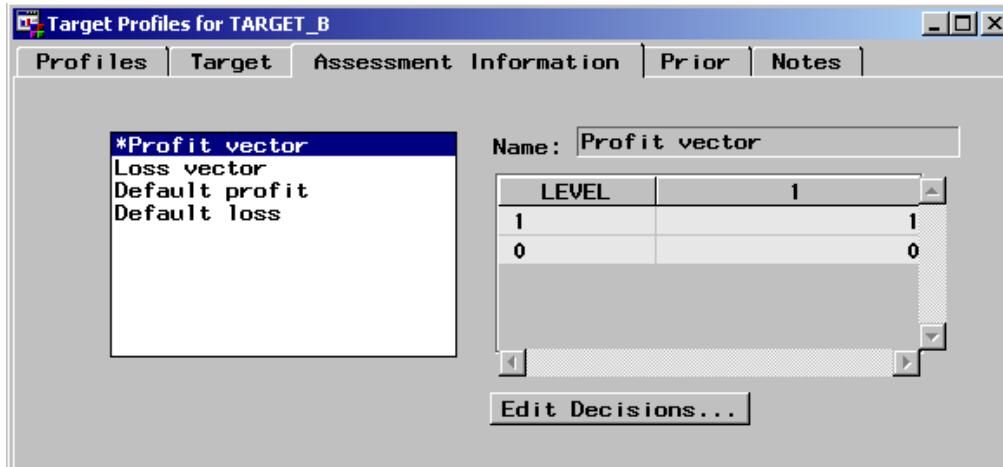


This tab shows that TARGET\_B is a binary target variable that uses the BEST12 format. It also shows that the two levels are sorted in descending order, and that the first listed level and modeled event is level 1 (the value next to Event).

8. To see the levels and associated frequencies for the target, select Levels.... Close the Levels window when you are done.

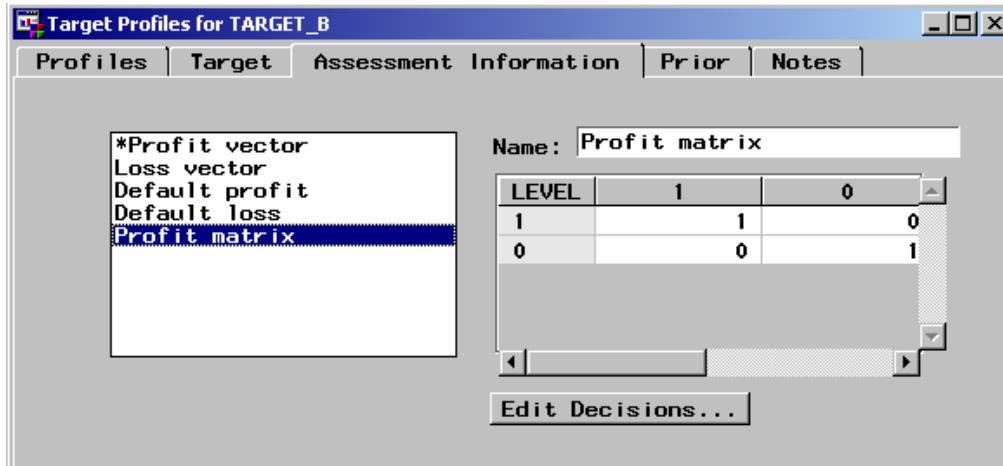


9. To incorporate profit and cost information into this profile, select the **Assessment Information** tab.



By default, the target profiler assumes you are trying to maximize profit using the default profit vector. This profit vector assigns a profit of 1 for each responder you correctly identify and a profit of 0 for every nonresponder you predict to respond. In other words, the best model maximizes accuracy. You can also build your model based on loss, or you can build it to minimize misclassification.

10. For this problem, create a new profit matrix by right-clicking in the open area where the vectors and matrices are listed and selecting **Add**.



A new matrix is formed. The new matrix is the same as the default profit matrix, but you can edit the fields and change the values, if desired. You can also change the name of the matrix.

11. Type **My Matrix** in the name field and press the Enter key.

For this problem, responders gave a median of \$13.00, and it costs approximately 68 cents to mail to each person; therefore, the net profit for

- mailing to a responder is  $13.00 - 0.68 = 12.32$
  - mailing to a nonresponder is  $0.00 - 0.68 = -0.68$
12. Enter the profits associated with the vector for action (LEVEL=1). Your matrix should appear as shown below. You may need to maximize your window to see all of the cells simultaneously. Do not forget to change the bottom right cell of the matrix to 0.

The screenshot shows a software window titled "Target Profiles for TARGET\_B". The menu bar includes "Profiles", "Target", "Assessment", "Information", "Prior", and "Notes". A toolbar with icons for "New", "Open", "Save", "Print", and "Exit" is visible. On the left, a list of options includes "\*Profit vector", "Loss vector", "Default profit", "Default loss", and "My Matrix" (which is selected and highlighted in blue). To the right, there is a "Name:" field containing "My Matrix" and a table editor. The table has a header row "LEVEL" with columns "1" and "0". Below this are two data rows: one for LEVEL 1 with values 12.32 and 0, and one for LEVEL 0 with values -0.68 and 0. A "Edit Decisions..." button is at the bottom.

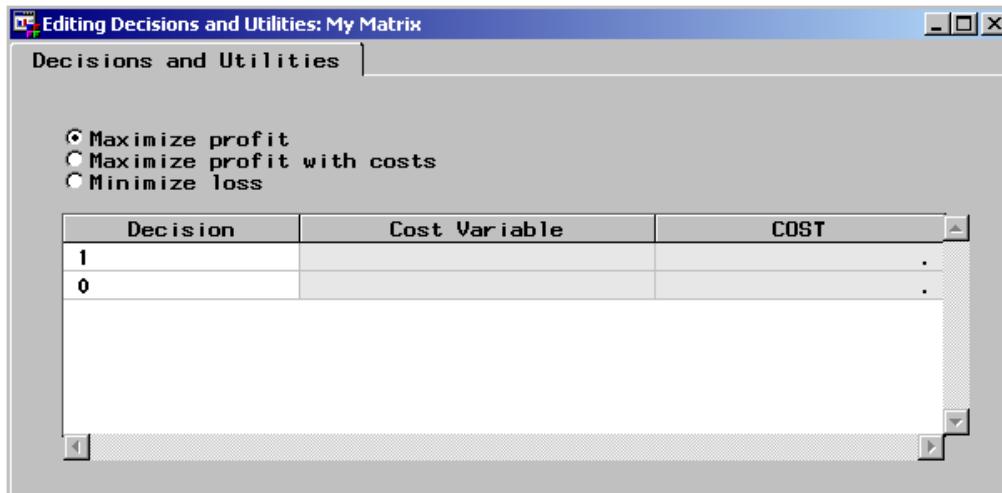
LEVEL	1	0
1	12.32	0
0	-0.68	0

13. To make the newly created matrix active, click on My Matrix to highlight it.
14. Right-click on My Matrix and select Set to use.

This screenshot is identical to the previous one, showing the "Target Profiles for TARGET\_B" software window. The matrix "My Matrix" is now highlighted in blue in the list of options on the left. The matrix table and "Edit Decisions..." button remain the same.

LEVEL	1	0
1	12.32	0
0	-0.68	0

15. To examine the decision criteria, select Edit Decisions....

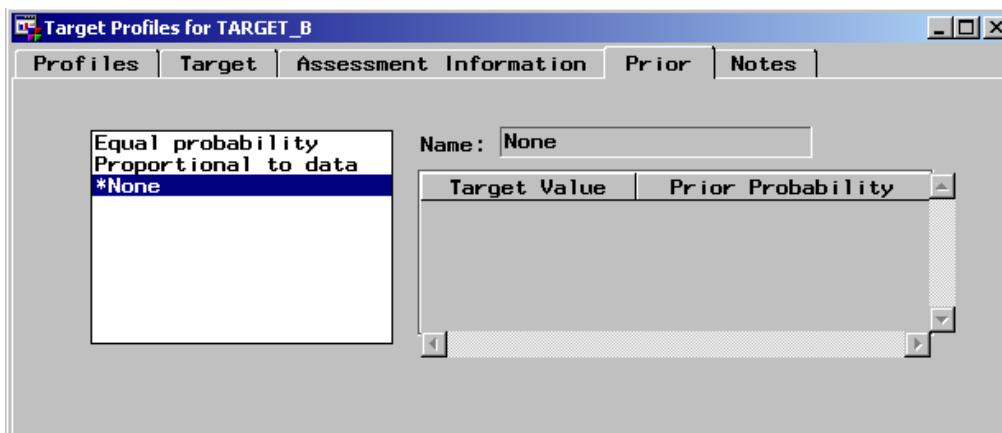


By default, you attempt to maximize profit. Because your costs have already been built into your matrix, do not specify them here. Optionally, you could specify profits of **13** and **0** (rather than 12.32 and -0.68) and then use a fixed cost of **0.68** for Decision=1 and **0** for Decision=0, but that is **not** done in this example. If the cost is not constant for each person, Enterprise Miner allows you to specify a cost variable. The radio buttons enable you to choose one of three ways to use the matrix or vector that is activated. You can choose to

- maximize profit (default) - use the active matrix on the previous page as a profit matrix, but do not use any information regarding a fixed cost or cost variable.
- maximize profit with costs - use the active matrix on the previous page as a profit matrix in conjunction with the cost information.
- minimize loss - consider the matrix or vector on the previous page as a loss matrix.

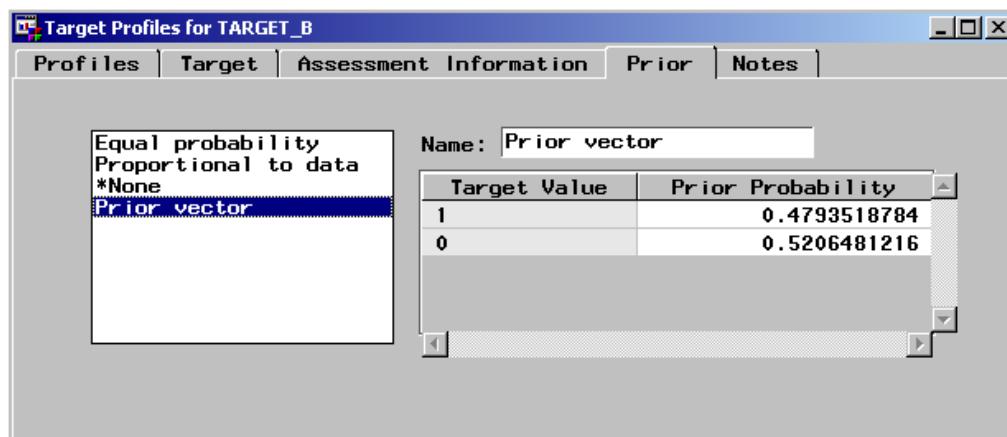
16. Close the Editing Decisions and Utilities window without modifying the table.

17. As discussed earlier, the proportions in the population are not represented in the sample. To adjust for this, select the **Prior** tab.

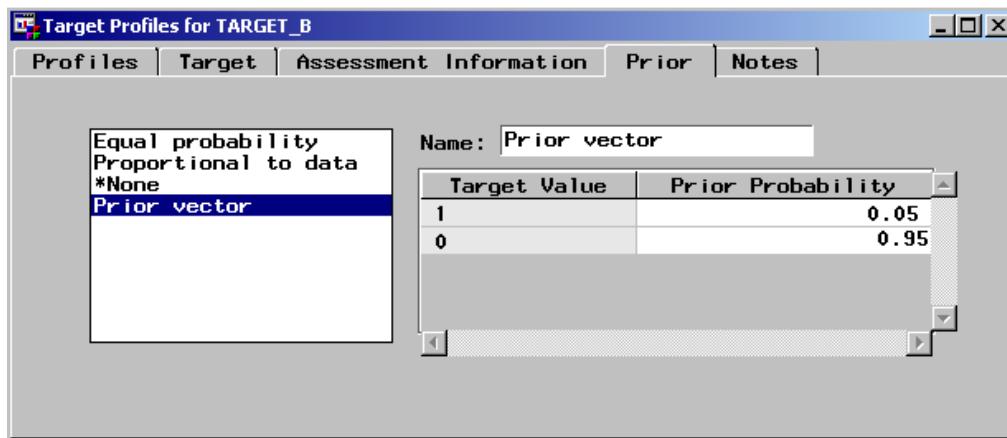


By default, there are three predefined prior vectors in the Prior tab:

- Equal Probability - contains equal probability prior values for each level of the target.
  - Proportional to data - contains prior probabilities that are proportional to the probabilities in the data.
  - None - (default) does not apply prior class probabilities.
18. To add a new prior vector, right-click in the open area where the prior profiles are activated and select **Add**. A new prior profile is added to the list, named Prior vector.
19. To highlight the new prior profile, select **Prior vector**.



20. Modify the prior vector to represent the true proportions in the population.



21. To make the prior vector the active vector, select **Prior vector** in the prior profiles list to highlight it.
22. Right-click on **Prior vector** and select **Set to use**.
23. Close the target profiler. Select **Yes** to save changes when prompted.

## Investigating Descriptive Statistics

The metadata is used to compute descriptive statistics for every variable.

1. Select the **Interval Variables** tab.

Name	Min	Max	Mean	Std Dev.
AGE	8	97	61.521	16.23
AVGGIFT	1.2857	336.67	12.804	12.013
CARDGIFT	0	26	5.37	4.678
CARDPROM	3	51	18.934	8.6094
FEDGOV	0	38	3.047	3.9233
FIRSTT	457	8552	2152.4	1158.2
LASTT	120	823	547.34	128.25
LOCALGOV	0	55	6.76	4.3968
MALEMIL1	0	99	0.853	3.6877
MALEVET	0	79	30.365	11.497
NUMPROM	8	140	48.421	23.253
STATEGOV	0	58	4.4495	5.1556
TARGET_D	0	150	7.154	10.914
TIMELAG	0	47	8.1385	6.4563

Investigate the descriptive statistics provided for the interval variables. Inspecting the minimum and maximum values indicates no unusual values (such as AGE=0 or TARGET\_D<0). AGE has a high percentage of missing values (26%). TIMELAG has a somewhat smaller percentage (9%).

2. Select the **Class Variables** tab.

Name	Values	Missing %	Order
GENDER	2	5%	Ascending
HOMEOWNR	2	24%	Ascending
IDCODE	128	0%	Ascending
INCOME	7	23%	Ascending
PCOWNERS	1	88%	Ascending
PETS	1	85%	Ascending
TARGET_B	2	0%	Descending

Investigate the number of levels, percentage of missing values, and the sort order of each variable. Observe that the sort order for TARGET\_B is descending whereas the sort order for all the others is ascending. This occurs because you have a binary target event. It is common to code a binary target with a 1 when the event occurs and a 0 otherwise. Sorting in descending order makes the 1 the first level, and this identifies the target event for a binary variable. It is useful to sort other similarly coded binary variables in descending order as well for interpreting results of a regression model.



If the maximum number of distinct values is greater than or equal to 128, the Class Variables tab will indicate 128 values.

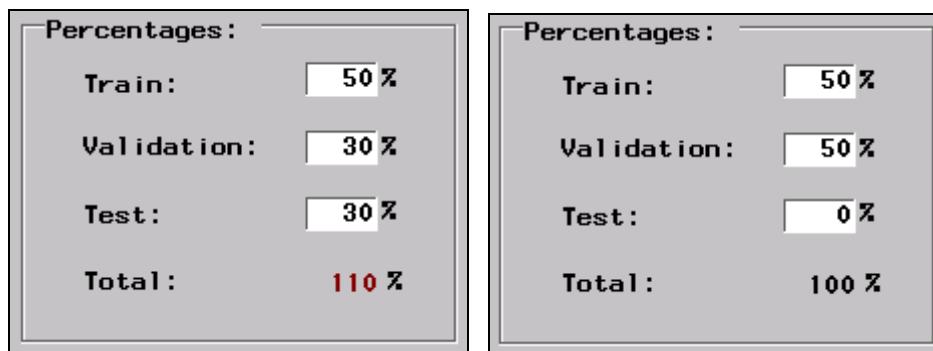
Close the Input Data Source node and save the changes when prompted.

### The Data Partition Node

1. Open the Data Partition node.
2. The right side enables you to specify the percentage of the data to allocate to training, validation, and testing data. Enter **50** for the values of training and validation.



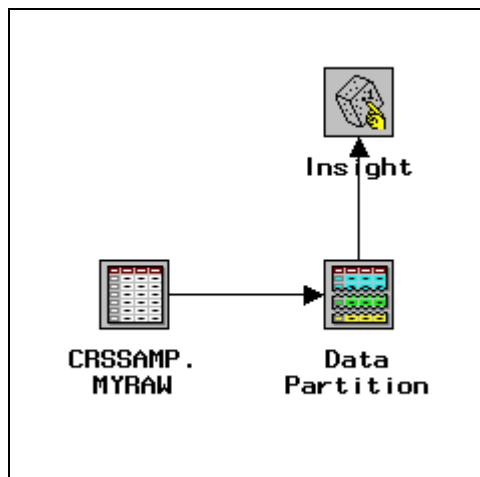
Observe that when you enter the 50 for training, the total percentage (110) turns red, indicating an inconsistency in the values. The number changes color again when the total percentage is 100. If the total is not 100%, the data partition node will not close.



3. Close the Data Partition node. Select **Yes** to save changes when prompted.

### Preliminary Investigation

1. Add an Insight node to the workspace and connect it to the Data Partition node as illustrated below.



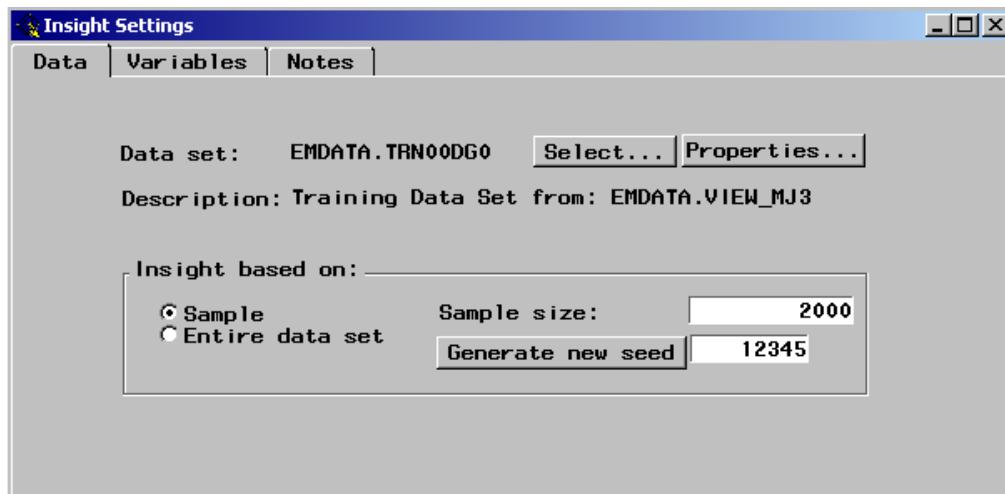
2. To run the flow from the Insight node, right-click on the node and select **Run**.

3. Select **Yes** when prompted to see the results. A portion of the output is shown below.

	21	Int	Nom	Int	Nom	Int	Int	Int	Int	
	2000	AGE	HOMEOWNR	INCOME	GENDER	MALEMIL	MALEVET	LOCALGOV	S	
■	1	64	H	5	M	0	31	8		
■	2	.		5	F	0	41	8		
■	3	54	H	7	F	0	32	4		
■	4	42	H	6	F	14	45	9		
■	5	58	H	4	M	43	35	3		
■	6	46	H	4	F	0	30	10		
■	7	86	H	1	F	0	30	4		
■	8	66	H	4	M	6	34	14		
■	9	44	H	5	F	1	34	13		
■	10	82	H	6	M	1	37	3		
■	11	64	H	7	M	0	33	7		
■	12	53	H	7	F	1	31	7		
■	13	84	H	2	F	0	0	0		
■	14	53	H	5	F	0	31	10		
■	15	68	H	4	M	2	42	6		
■	16	66	H	5	M	0	33	6		
■	17	78	H	6	M	0	33	6		

Observe that the upper-left corner has the numbers 2000 and 21, which indicate there are 2000 rows (observations) and 21 columns (variables). This represents a sample from either the training data set or the validation data set, but how would you know which one?

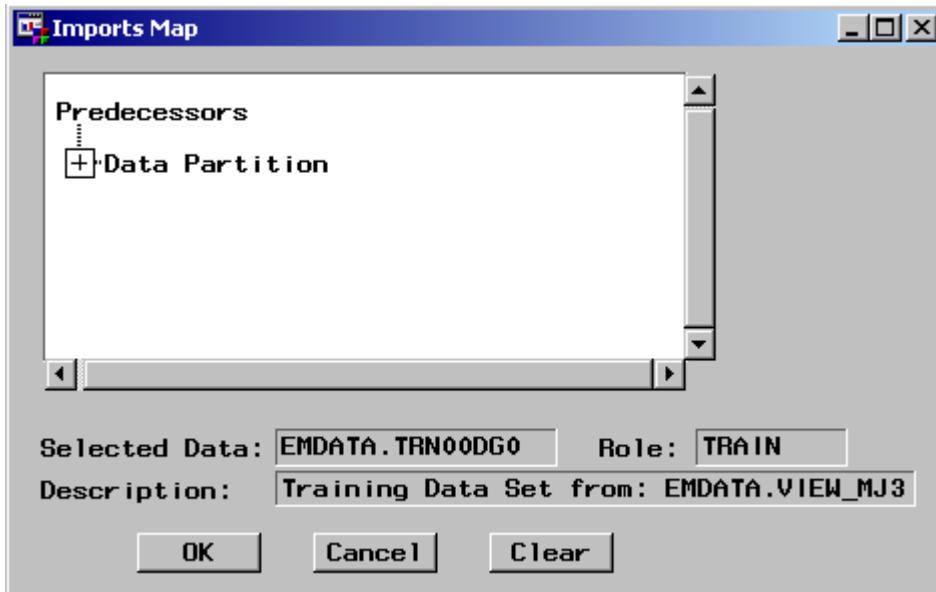
1. Close the Insight data set to return to the workspace.
2. To open the Insight node, right-click on the node in the workspace and select **Open...**. The Data tab is initially active and is displayed below.



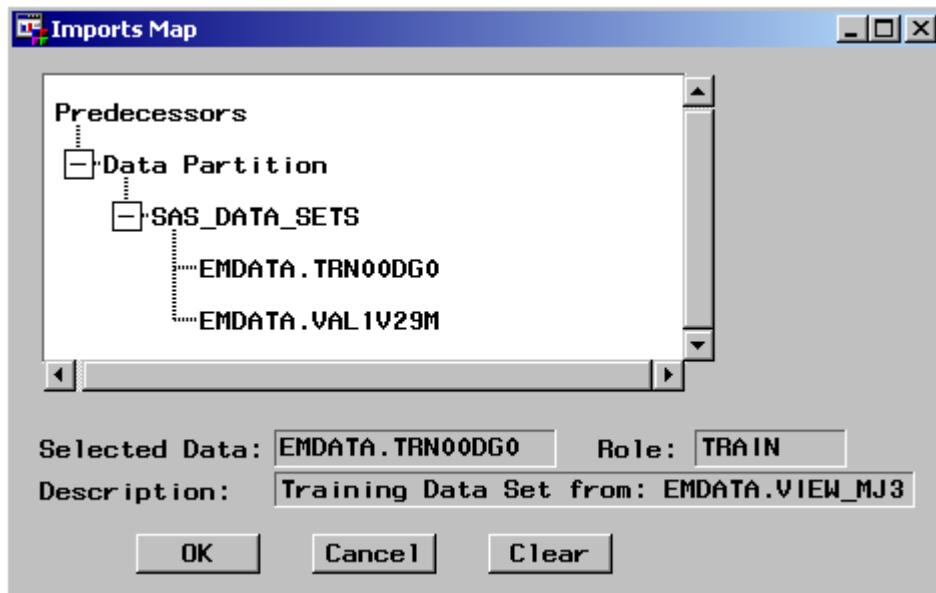
Observe that the selected data set is the training data set. The name of the data set is composed of key letters (in this case, TRN) and some random alphanumeric characters (in this case, 00DG0). The TRN00DG0 data set is stored in the EMDATA library. The bottom of the tab indicates that Insight, by default, is generating a random sample of 2000 observations from the training data based on the random seed 12345.

3. To change which data set Insight is using, choose **Select....**

You can see the predecessor nodes listed in a table. The Data Partition node is the only predecessor.

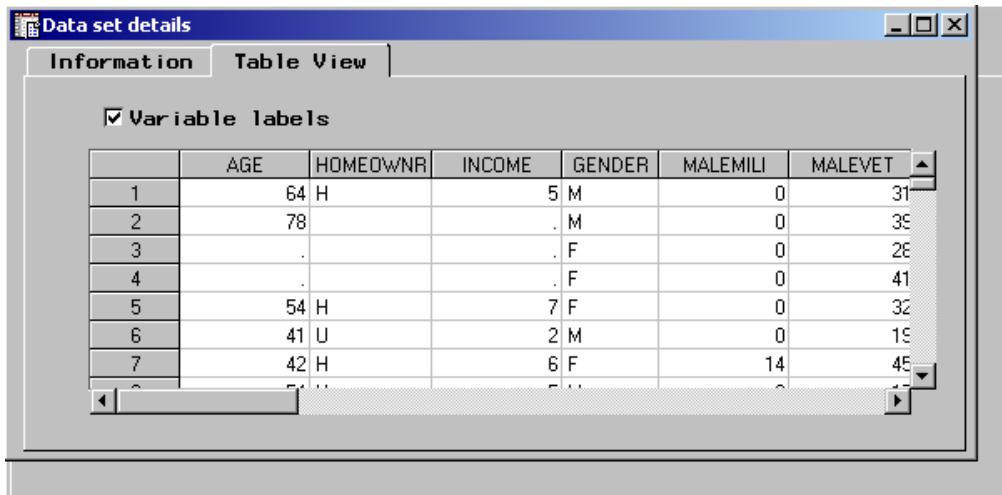


4. Click on the **+** next to Data Partition and then click on the **+** next to SAS\_DATA\_SETS. Two data sets are shown that represent the training and validation data sets.



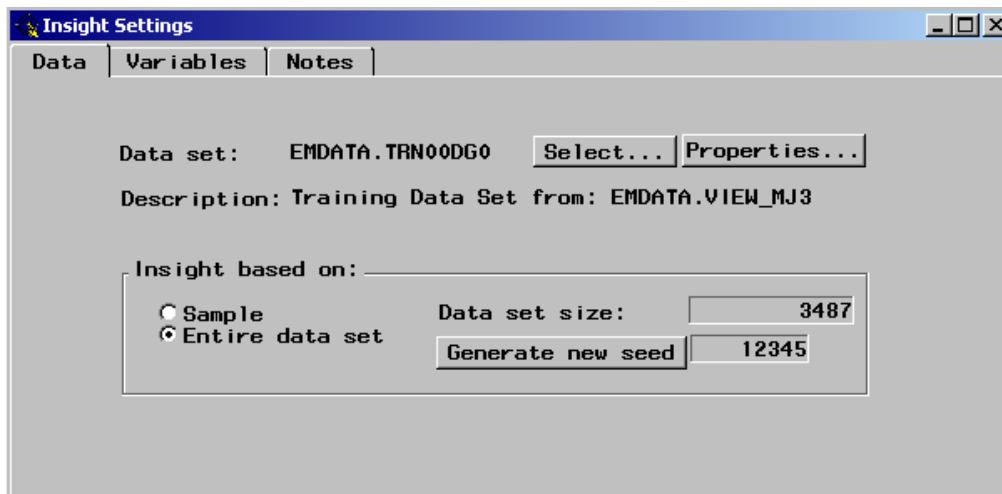
5. Leave the training data set as the selected data and select **OK** to return to the Data tab.
6. Select **Properties....** The Information tab is active. This tab provides information about when the data set was constructed as well as the number of rows and columns.

7. Select the **Table View** tab.



This tab enables you to view the data for the currently selected data set in tabular form. The check box enables you to see the column headings using the variable labels. Unchecking the box would cause the table to use the SAS variable names for column headings. If no label is associated with the variable, the column heading cell displays the SAS variable name.

8. Close the Data set details window when you are finished to return to the main Insight dialog.
9. Select the radio button next to **Entire data set** to run Insight using the entire data set.



You can run Insight with the new settings by proceeding as follows:

1. Close the Insight Settings window and select **Yes** when prompted to save changes.
2. Run the diagram from the Insight node.
3. Select **Yes** when prompted to see the results.

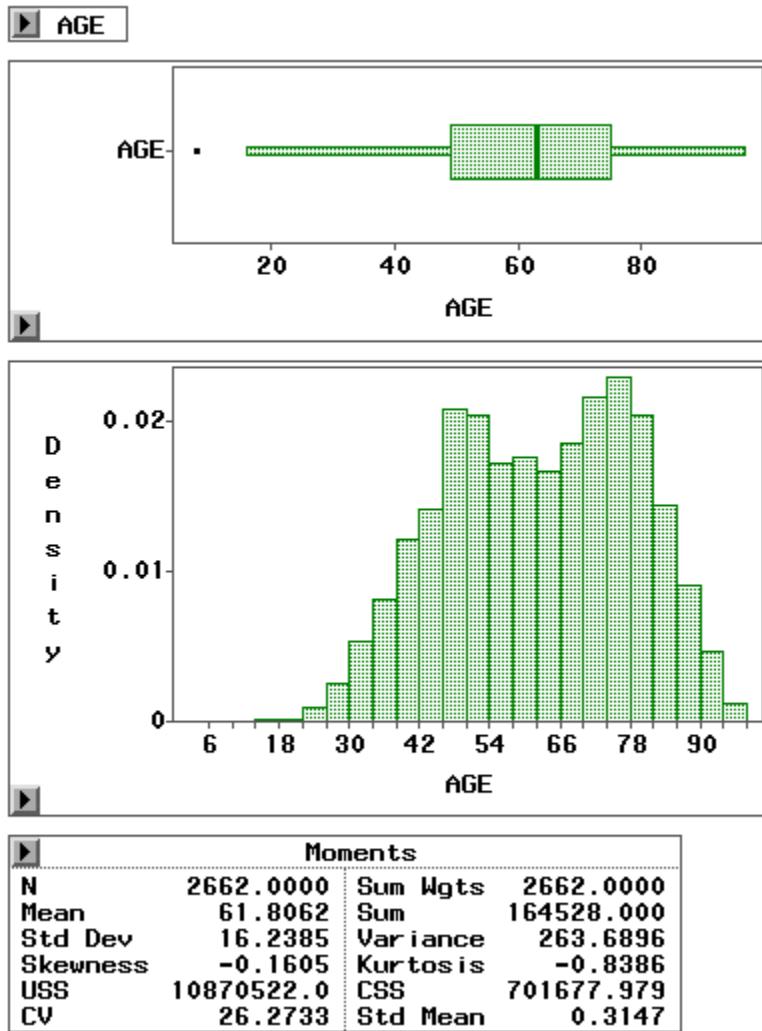


You can also run Insight without closing the main dialog by selecting the run icon ( ) from the toolbar and selecting **Yes** when prompted to see the results.

Use Insight to look at the distribution of each of the variables:

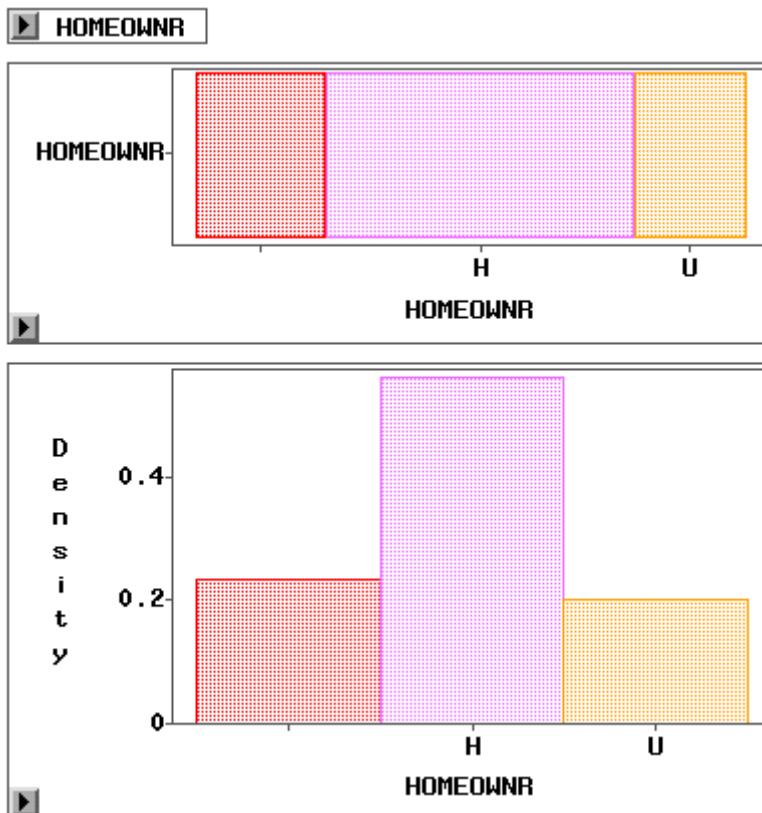
1. Select **Analyze**  $\Rightarrow$  **Distribution (Y)**.
2. Highlight all of the variables except IDCODE in the variable list (IDCODE is the last variable in the list).
3. Select **Y**.
4. Select **IDCODE**  $\Rightarrow$  **Label**.
5. Select **OK**.

Charts for numeric variables include histograms, box and whisker plots, and assorted descriptive statistics.



The distribution of AGE is not overly skewed, so no transformation seems necessary.

Charts for character variables include mosaic plots and histograms.



The variable HOMEOWNR has the value **H** when the person is a homeowner and a value of **U** when the ownership status is unknown. The bar at the far left represents a missing value for HOMEOWNR. These missing values indicate that the value for HOMEOWNR is unknown, so recoding these missing values into the level **U** would remove the redundancy in this set of categories. You do this later in the Replacement node.

Some general comments about other distributions appear below.

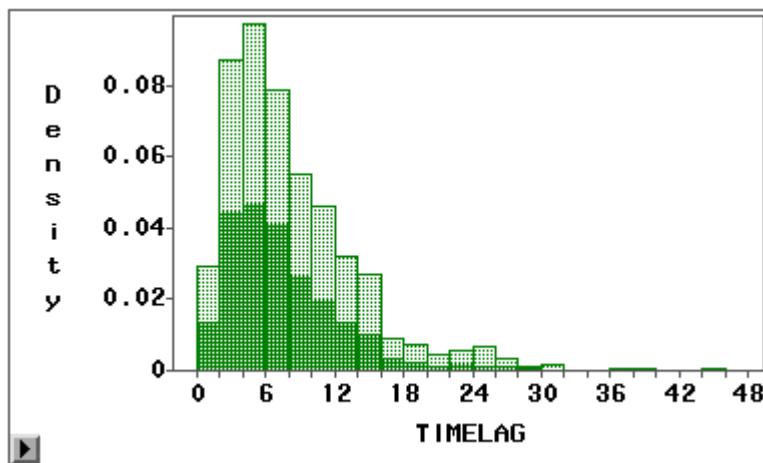
- INCOME is treated like a continuous variable because it is a numeric variable.
- There are more females than males in the training data set, and the observations with missing values for GENDER should be recoded to **M** or **F** for regression and neural network models. Alternatively, the missing values could be recoded to **U** for unknown.
- The variable MALEMILI is a numeric variable, but the information may be better represented if the values are binned into a new variable.
- The variable MALEVET does not seem to need a transformation, but there is a spike in the graph near MALEVET=0.
- The variables LOCALGOV, STATEGOV, and FEDGOV are skewed to the right, so they may benefit from a log transformation.
- The variables PETS and PCOWNERS only contain the values Y and missing. Recoding the missing values to **U** for unknown would make these variable more useful for regression and neural network models.

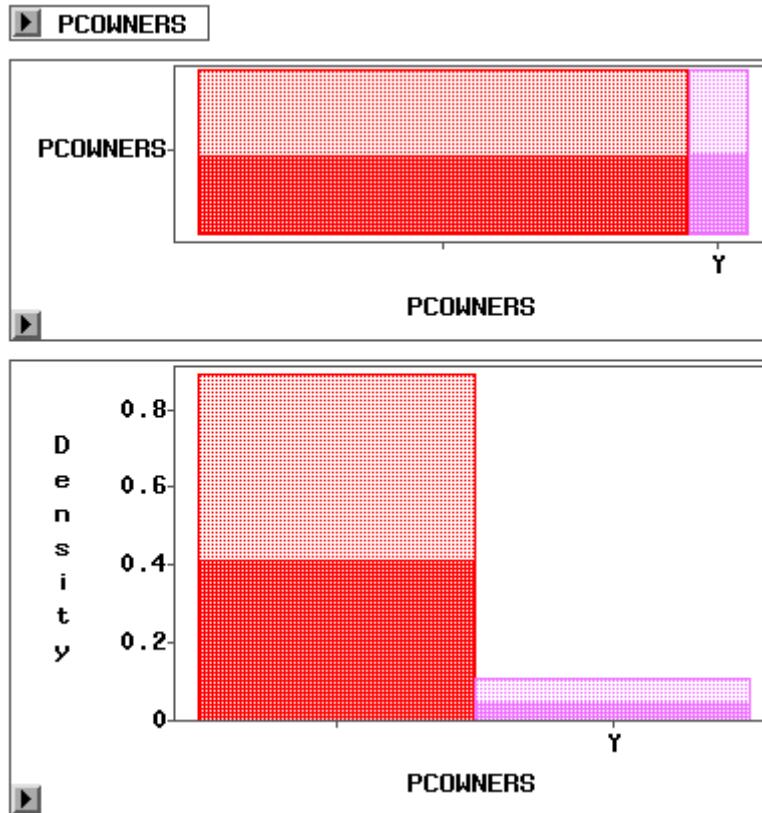
- The distributions of CARDPROM and NUMPROM do not need any transformation.
- The variables CARDGIFT and TIMELAG may benefit from a log transformation.
- The variable AVGGIFT may yield better results if its values are binned.

You can use Insight to see how responders are distributed.

1. Scroll to the distribution of TARGET\_B.
2. Select the bar corresponding to TARGET\_B=1
3. Scroll to the other distributions and inspect the highlighting pattern.

Examples of the highlighting pattern for TIMELAG and PCOWNERS are shown. These graphs do not show any clear relationships.

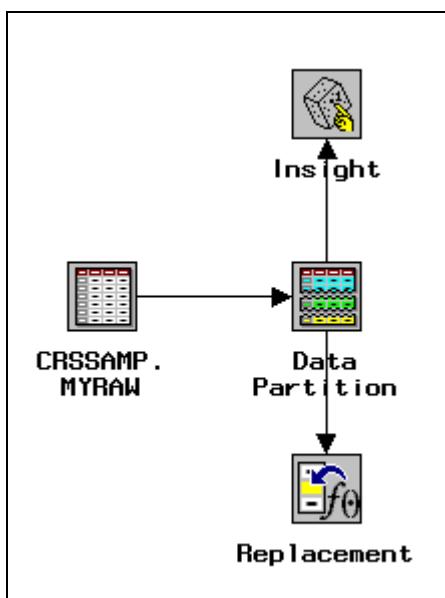




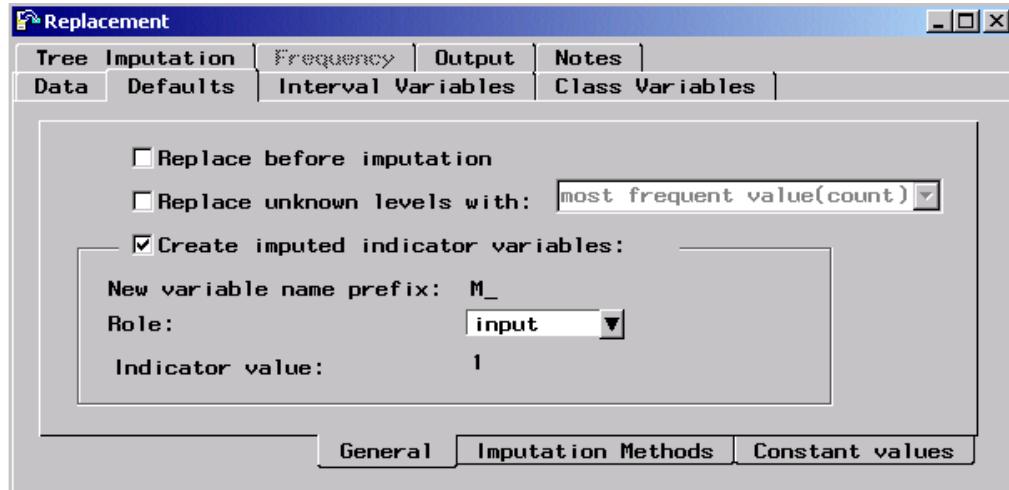
When you are finished, return to the main process flow diagram by closing the Insight windows.

### Understanding Data Replacement

1. Add a Replacement node to the diagram. Your new diagram should appear as follows:



2. Open the Replacement node.
3. The Defaults tab is displayed first. Check the box for **Create imputed indicator variables** and use the arrow to change the Role field to **input**.



This requests the creation of new variables, each having a prefix **M\_** followed by the original variable name. These new variables have a value of **1** when an observation has a missing value for the associated variable and **0** otherwise. If the “missingness” of a variable is related to the response variable, the regression and the neural network model can use these newly created indicator variables to identify observations that had missing values originally.

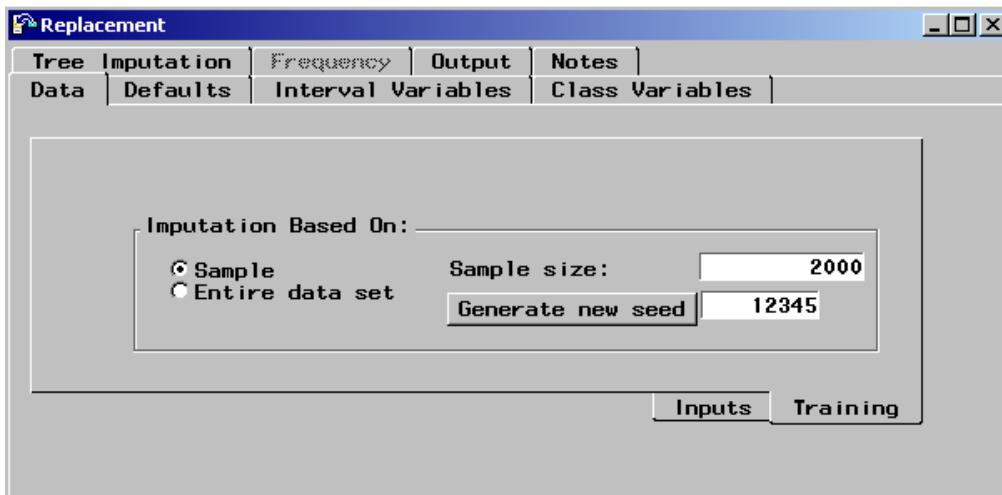
The Replacement node allows you to replace certain values before imputing. Perhaps a data set has coded all missing values as 999. In this situation, select the Replace before imputation check box and then have the value replaced before imputing.

When the class variables in the score data set contain values that are not in the training data set, these unknown values can be imputed by the most frequent values or missing values. To do this, select the Replace unknown level with check box and then use the drop-down list to choose either most frequent value (count) or missing value.

## Using Data Replacement

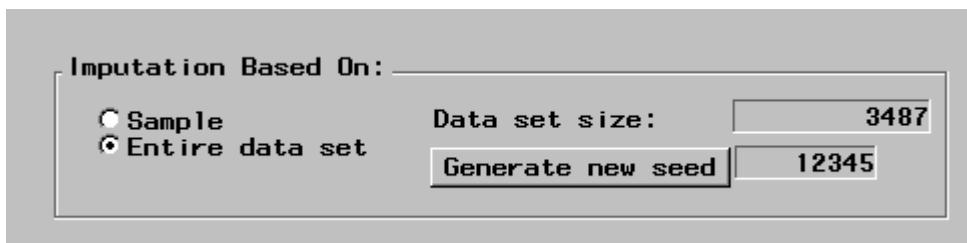
1. Select the **Data** tab. Most nodes have a Data tab that enables you to see the names of the data sets being processed as well as a view of the data in each one. The radio button next to Training is selected.

2. Select the **Training** subtab under the Data tab.

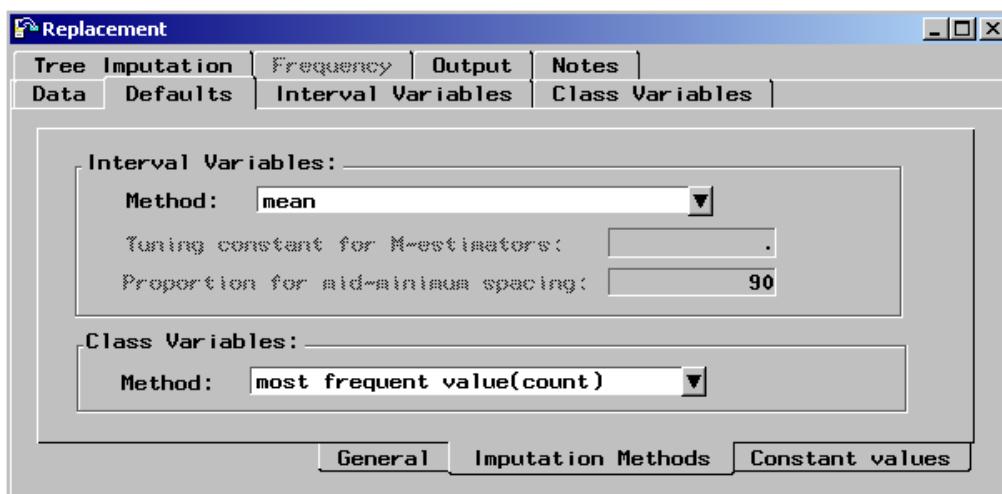


By default, the imputation is based on a random sample of the training data. The seed is used to initialize the randomization process. Generating a new seed creates a different sample.

3. To use the entire training data set, select the button next to Entire data set. The subtab information now appears as pictured below.



4. Return to the Defaults tab and select the Imputation Methods subtab.



This shows that the default imputation method for Interval Variables is the mean (of the random sample from the training data set or the entire training data set, depending on the settings in the Data tab). By default, imputation for class variables is done

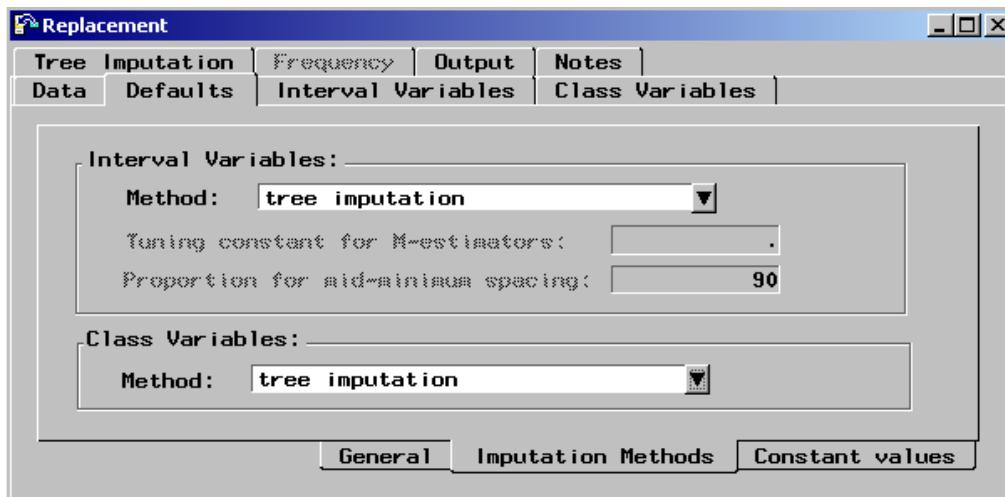
using the most frequently occurring level (or mode) in the same sample. If the most commonly occurring value is missing, it uses the second most frequently occurring level in the sample.

Click on the arrow next to the method for interval variables. Enterprise Miner provides the following methods for imputing missing values for interval variables:

- Mean – uses the arithmetic average. This is the default.
- Median – uses the 50<sup>th</sup> percentile.
- Midrange – uses the maximum plus the minimum divided by two.
- Distribution-based – calculates replacement values based on the random percentiles of the variable's distribution.
- Tree imputation – estimates replacement values with a decision tree using the remaining input and rejected variables that have a status of use as the predictors.
- Tree imputation with surrogates – is the same as above except that surrogate variables are used for splitting whenever a split variable has a missing values. This prevents forcing everyone with a missing value for a variable into the same node.
- Mid-min spacing – uses the mid-minimum spacing statistic. To calculate this statistic, the data is trimmed using  $N$  percent of the data as specified in the Proportion for mid-minimum spacing entry field. By default, 90% of the data is used to trim the original data. In other words, 5% of the data is dropped from each end of the distribution. The mid-range is calculated from this trimmed data.
- Tukey's biweight, Huber's, and Andrew's wave – are robust M-estimators of location. This class of estimators minimize functions of the deviations of the observations from the estimate that are more general than the sum of squared deviations or the sum of absolute deviations. M-estimators generalize the idea of the maximum-likelihood estimator of the location parameter in a specified distribution.
- Default constant – enables you to set a default value to be imputed for some or all variables.
- None – turns off the imputation for interval variables.

Click on the arrow next to the method for class variables. Enterprise Miner provides several of the same methods for imputing missing values for class variables including distribution-based, tree imputation, tree imputation with surrogates, default constant, and none.

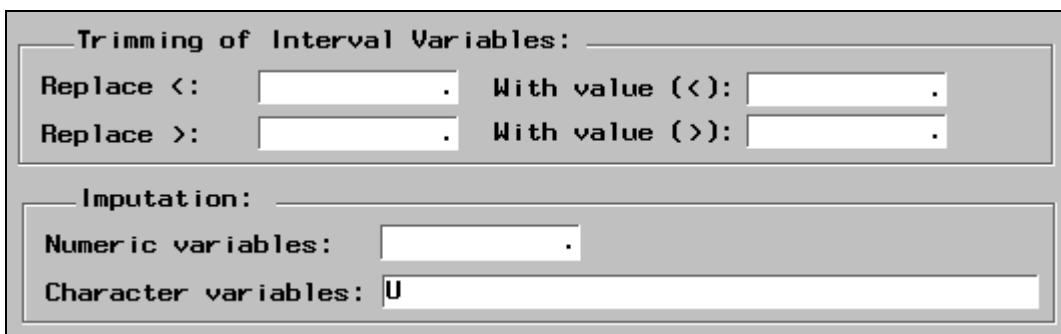
5. Select **Tree imputation** as the imputation method for both types of variables.



When using the tree imputation for imputing missing values, use the entire training data set for more consistent results.

Regardless of the values set in this section, you can select any imputation method for any variable. This tab merely controls the default settings.

6. Select the **Constant values** subtab. This subtab enables you to replace certain values (before imputing, if desired, using the check box on the Defaults tab). It also enables you to specify constants for imputing missing values.
7. Enter **U** in the field for character variables.



8. Select the **Tree Imputation** tab. This tab enables you to set the variables that will be used when using tree imputation. Observe that target variables are not available, and rejected variables are not used by default. To use a rejected variable, you can set the Status to **use**, but that would be inappropriate here because the rejected variable TARGET\_D is related to the target variable TARGET\_B.

Name	Status	Model Role	Measurement	Type	Format
AGE	use	input	interval	num	BEST12.
AVGGIFT	use	input	interval	num	BEST12.
CARDGIFT	use	input	interval	num	BEST12.
CARDPROM	use	input	interval	num	BEST12.
FEDGOV	use	input	interval	num	BEST12.
FIRSTT	use	input	interval	num	BEST12.
LASTT	use	input	interval	num	BEST12.
LOCALGOV	use	input	interval	num	BEST12.

Suppose you want to change the imputation method for AGE to mean and CARDPROM to 20.

1. Select the **Interval Variables** tab.
2. To specify the imputation method for AGE, position the tip of your cursor on the row for AGE in the Imputation Method column and right-click.
3. Select **Select Method...**  $\Rightarrow$  **mean**.
4. To specify the imputation method for CARDPROM, position the tip of your cursor on the row for CARDPROM in the Imputation Method column and right-click.
5. Select **Select Method...**  $\Rightarrow$  **set value...**
6. Type **20** for the new value.
7. Select **OK**.
8. Specify **none** as the imputation method for TARGET\_D in like manner.

Inspect the resulting window. A portion of the window appears below.

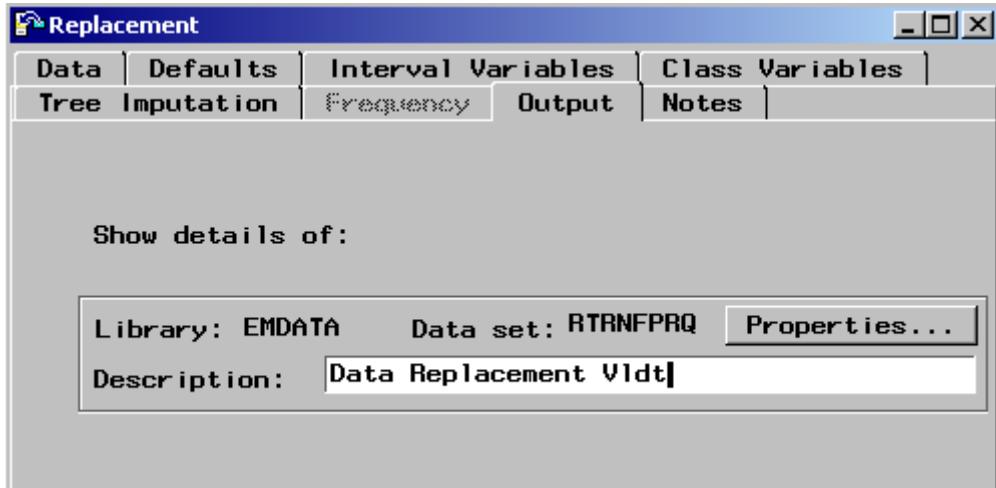
F Replacement					
Tree Imputation		Frequency	Output	Notes	
Data		Defaults		Interval Variables	
<b>Imputation Method</b>					
Name	Status	Model Role			
AGE	use	input	mean		
AVGGIFT	use	input	tree imputation		
CARDGIFT	use	input	tree imputation		
CARDPROM	use	input	set value - 20		
FEDGOV	use	input	tree imputation		
FIRSTT	use	input	tree imputation		
LASTT	use	input	tree imputation		
LOCALGOV	use	input	tree imputation		
MALEMIL	use	input	tree imputation		
MALEVET	use	input	tree imputation		
NUMPROM	use	input	tree imputation		
STATEGOV	use	input	tree imputation		
TARGET_D	don't use	rejected	none		

Recall that the variables HOMEOWNR, PCOWNERS, and PETS should have the missing values set to **U**.

1. Select the **Class Variables** tab.
2. Control-click to select the rows for **HOMEOWNR**, **PCOWNERS**, and **PETS**.
3. Right-click on one of the selected rows in the Imputation Method column.
4. Select **Select Method...**  $\Rightarrow$  **default constant**.
5. To change the imputation for TARGET\_B to none, right-click on the row for TARGET\_B in the Imputation Method column.
6. Choose **Select method...**  $\Rightarrow$  **none**.

F Replacement					
Tree Imputation		Frequency	Output	Notes	
Data		Defaults		Interval Variables	
<b>Imputation Method</b>					
Name	Status				
GENDER	use	tree imputation			
HOMEOWNR	use	default constant - U			
INCOME	use	tree imputation			
PCOWNERS	use	default constant - U			
PETS	use	default constant - U			
TARGET_B	don't use	none			

7. Select the **Output** tab. While the Data tab shows the input data, the Output tab shows the output data set information.

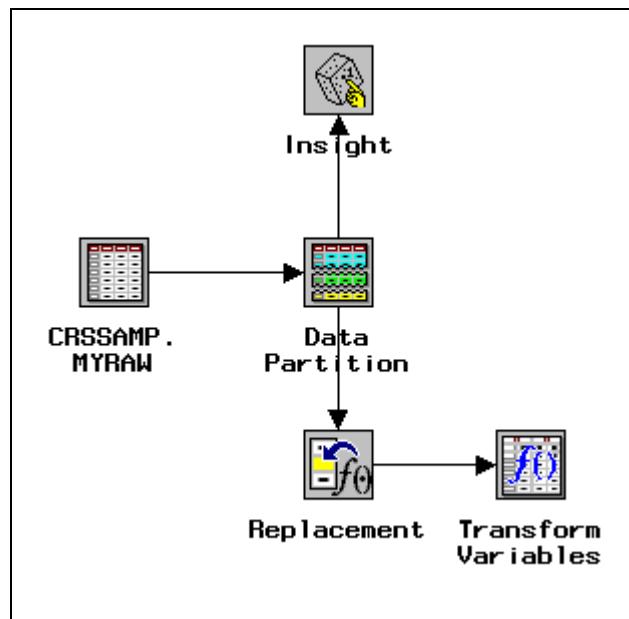


8. Close the Replacement node saving the changes when prompted.

### Performing Variable Transformations

Some input variables have highly skewed distributions. In highly skewed distributions, a small percentage of the points may have a great deal of influence. On occasion, performing a transformation on an input variable may yield a better fitting model. This section demonstrates how to perform some common transformations.

Add a Transform Variables node to the flow as shown below.

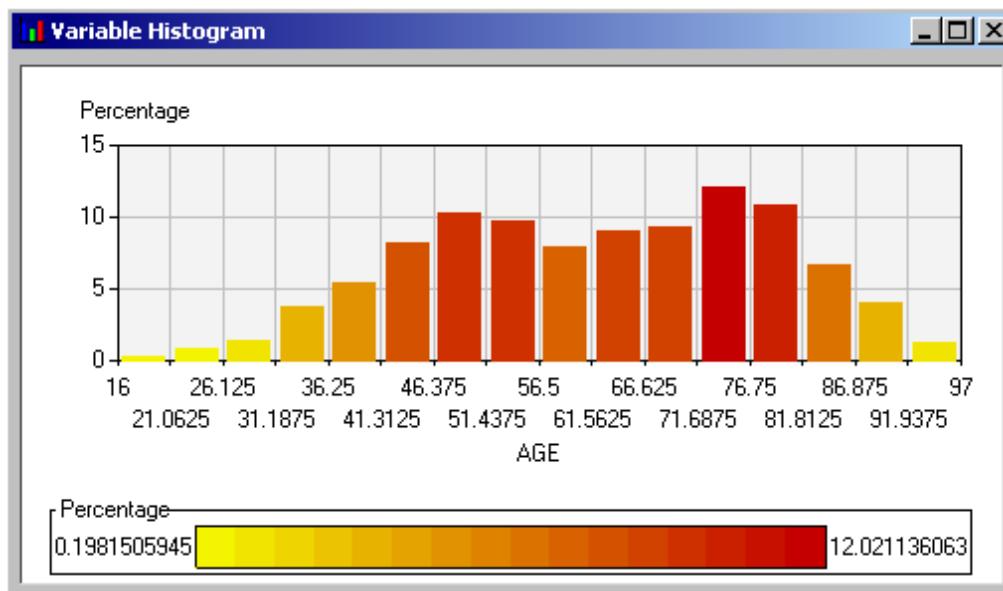


Open the Transform Variables node by right-clicking on it and selecting **Open...**. The Variables tab is shown by default. It displays statistics for the interval-level variables including the mean, standard deviation, skewness, and kurtosis (calculated from the metadata sample). The Transform Variables node enables you to rapidly transform

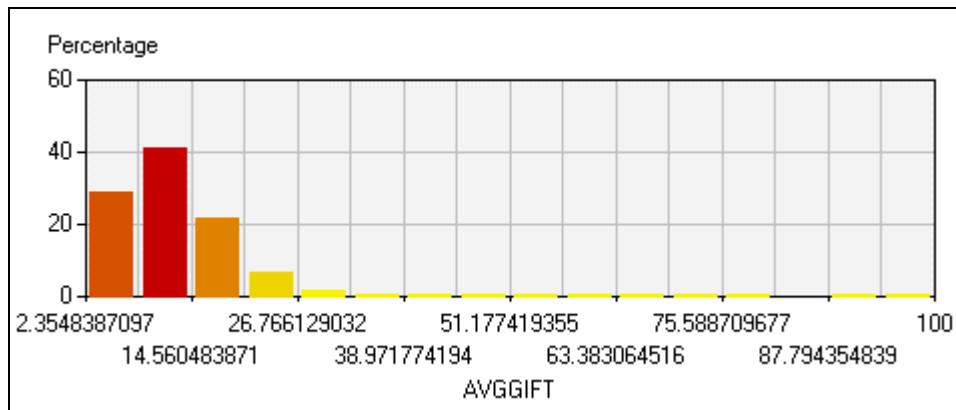
interval-valued variables using standard transformations. You can also create new variables whose values are calculated from existing variables in the data set. Observe that the only nongreyed column in this dialog is the Keep column.

Transform Variables					
		Data	Variables	Output	Notes
Name	Keep	Role	Formula	Mean	Std Dev
AGE	Yes	input		62.112945839	16.193021235
AVGGIFT	Yes	input		12.81512315	8.1052913573
CARDGIFT	Yes	input		5.3205	4.5657642133
CARDPROM	Yes	input		18.7555	8.4437769058
FEDGOV	Yes	input		3.1765	4.483569889
FIRSTT	Yes	input		2114.6415	1122.2359341
LASTT	Yes	input		549.7225	129.94511672
LOCALGOV	Yes	input		6.881	4.5244971585
MALEMIL	Yes	input		1.073	4.7894822862
MALEVET	Yes	input		30.543	11.516399816
NUMPROM	Yes	input		47.9305	22.855543259
STATEGOV	Yes	input		4.5635	5.4170693248
TARGET_D	Yes	rejected		7.65396	12.035944562
TIMELAG	Yes	input		7.8914473684	6.1033567457

You can view the distribution of each of the variables just as you did in the Input Data Source node. Begin by viewing the distribution of AGE. The distribution of AGE is not highly skewed, so no transformation is performed. Close the distribution of AGE.



Investigate the distribution of AVGGIFT.



This variable has the majority of its observations near zero, and very few observations appear to be higher than 30. Consider creating a new grouping variable that creates bins for the values of AVGGIFT. You can create just such a grouping variable in several different ways.

- Bucket - creates cutoffs at approximately equally spaced intervals.
- Quantile - creates bins with approximately equal frequencies.
- Optimal Binning for Relationship to Target - creates cutoffs that yield optimal relationship to target (for binary targets).

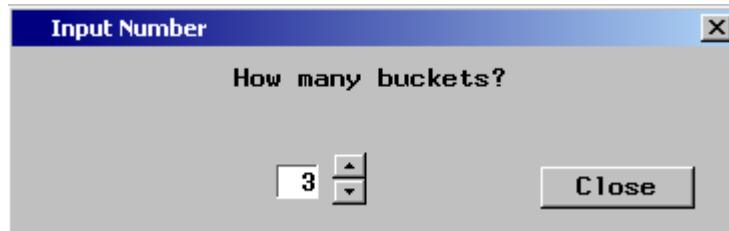
The Optimal Binning for Relationship to Target transformation uses the DMSPLIT procedure to optimally split a variable into  $n$  groups with regard to a binary target. This binning transformation is useful when there is a nonlinear relationship between the input variable and the binary target. An ordinal measurement level is assigned to the transformed variable.

To create the  $n$  optimal groups, the node applies a recursive process of splitting the variable into groups that maximize the association with the target values. To determine the optimum groups and to speed processing, the node uses the metadata as input.

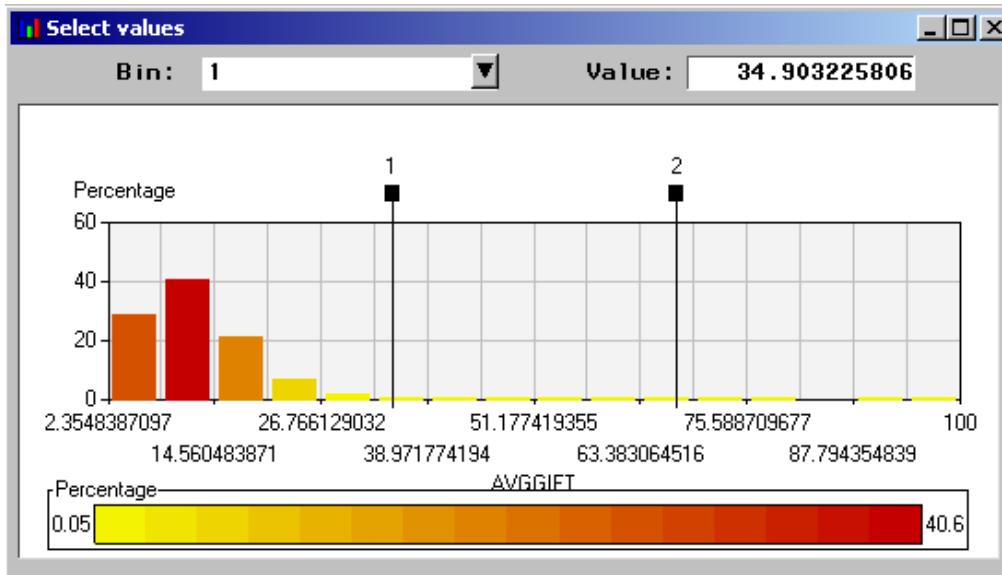
Close the distribution of AVGGIFT.

Create bins for AVGGIFT. Suppose your earlier analysis suggested binning the values into the intervals 0-10, 10-20, and 20+.

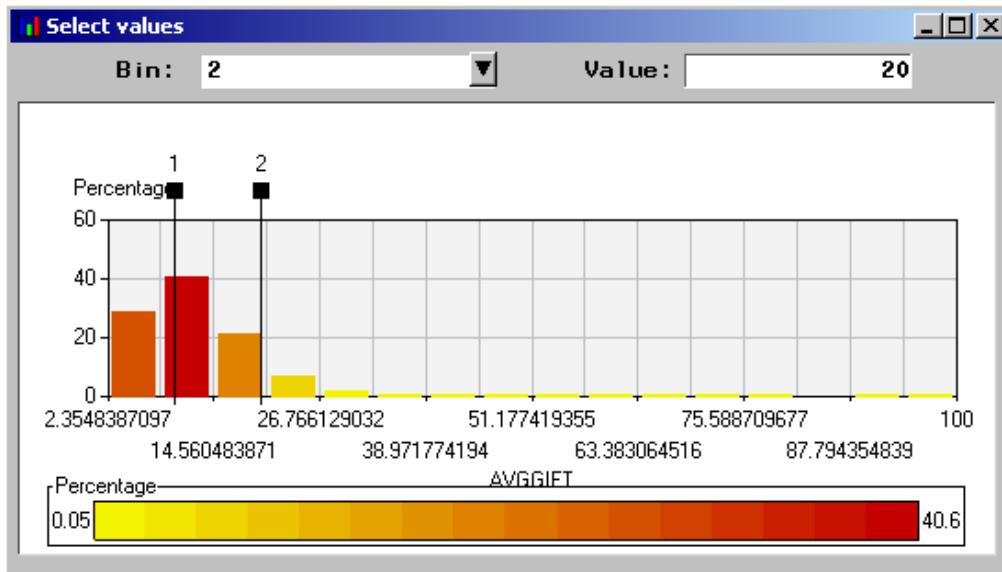
1. Right-click on the row for AVGGIFT and select Transform...  $\Rightarrow$  Bucket.
2. The default number of buckets is 4. Change this value to 3 using the arrows.



3. Select Close.



4. Enter **10** in the Value field for Bin 1. Press the Enter key.
5. Use the **▼** to change from Bin 1 to Bin 2.
6. Enter **20** in the Value field for Bin 2 and press the Enter key. The result appears as pictured below:

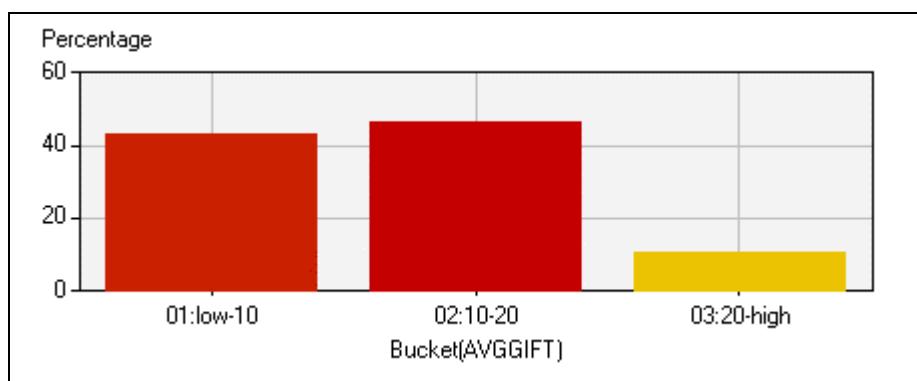


7. Close the plot and select **Yes** to save the changes when prompted.

A new variable is added to the table. The new variable has the truncated name of the original variable followed by a random string of digits. Note that the Enterprise Miner set the value of Keep to **No** for the original variable. If you wanted to use both the binned variable and the original variable in the analysis, you would need to modify this attribute for AVGGIFT and the set the value of Keep to **Yes**, but that is not done here.

Name	Keep	Role	Formula	Mean
AGE	Yes	input		62.112945839
AVGGIFT	No	input		12.81512315
AVGG_NDA	Yes	input	AVGGIFT	12.81512315
CARDGIFT	Yes	input		5.3205

Examine the distribution of the new variable.



The View Info tool reveals that there is over 40% of the data in each of the two lowest categories and there is approximately 10% of the data in the highest category.

Recall that the distributions of LOCALGOV, STATEGOV, FEDGOV, CARDGIFT, and TIMELAG were highly skewed to the right. A log transformation of these variables may provide more stable results.

Begin by transforming CARDGIFT.

1. Position the tip of the cursor on the row for CARDGIFT and right-click.
2. Select Transform...  $\Rightarrow$  log.

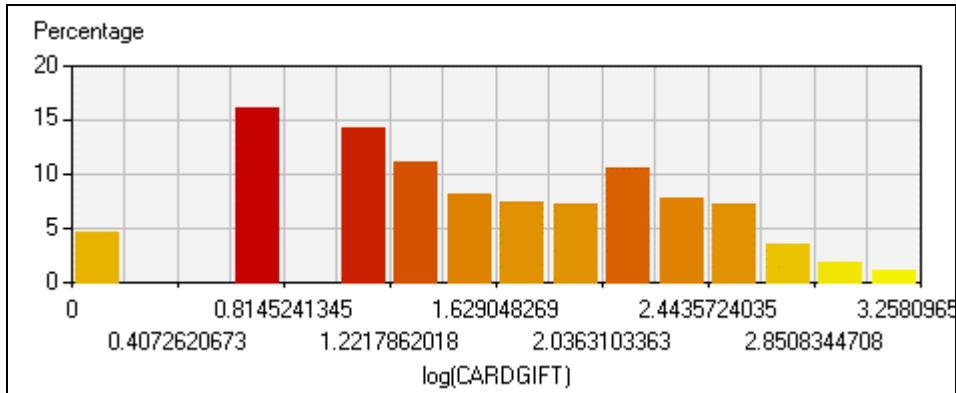
Inspect the resulting table.

Name	Keep	Role	Formula
AGE	Yes	input	
AVGGIFT	No	input	
AVGG_NDA	Yes	input	AVGGIFT
CARDGIFT	No	input	
CARD_8EF	Yes	input	log((CARDGIFT + 1))

The formula shows that Enterprise Miner has performed the log transformation after adding 1 to the value of CARDGIFT. Why has this occurred? Recall that CARDGIFT has a minimum value of zero. The logarithm of zero is undefined, and the logarithm of something close to zero is extremely negative. The Enterprise Miner takes this

information into account and actually uses the transformation  $\log(\text{CARDGIFT}+1)$  to create a new variable with values greater than or equal to zero (because the  $\log(1)=0$ ).

Inspect the distribution of the transformed variable. It is much less skewed than before.



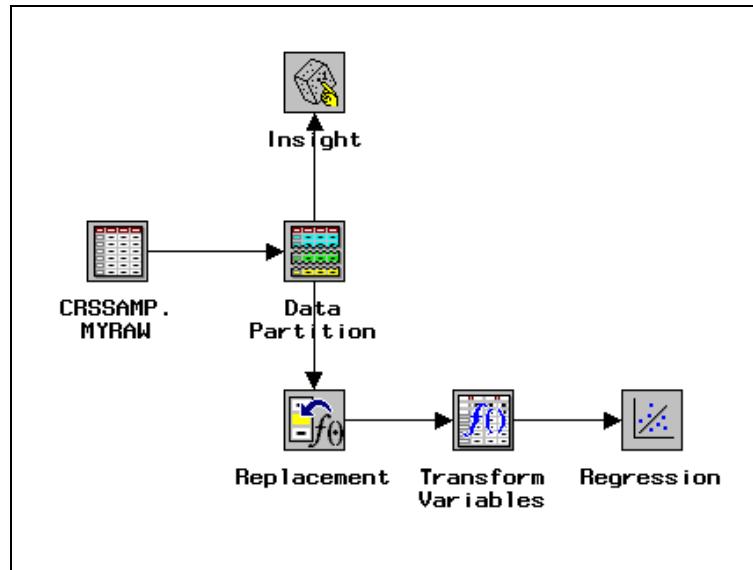
Perform log transformations on the other variables (FEDGOV, LOCALGOV, STATEGOV, and TIMELAG).

1. Press and hold the Ctrl key on the keyboard.
2. While holding the Ctrl key, select each of the variables.
3. When all have been selected, release the Ctrl key.
4. Right-click on one of the selected rows and select **Transform...  $\Rightarrow$  log**.
5. View the distributions of these newly created variables.

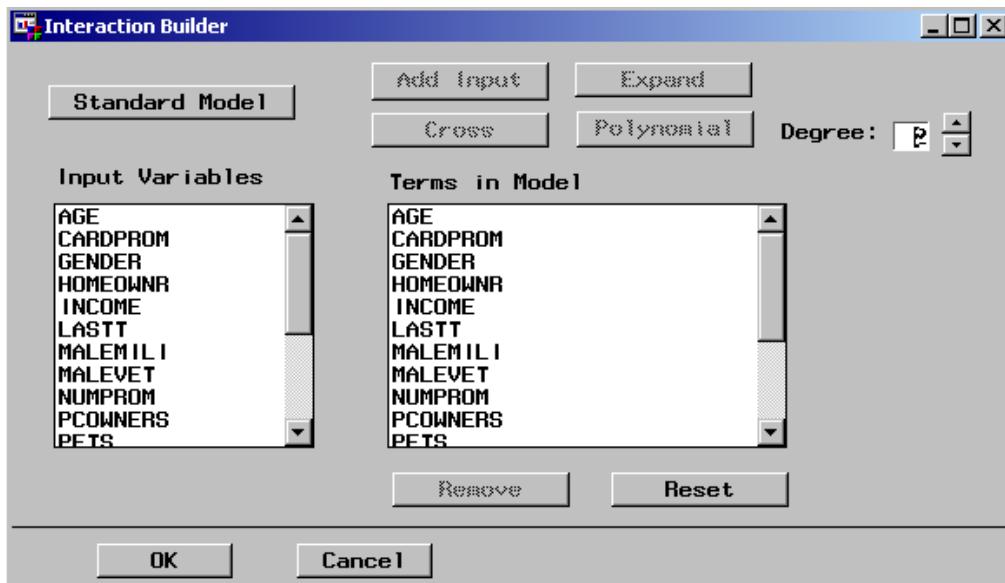
Close the node when you are finished, saving changes when prompted.

### Fitting a Regression Model

1. Connect a Regression node to the diagram as shown.

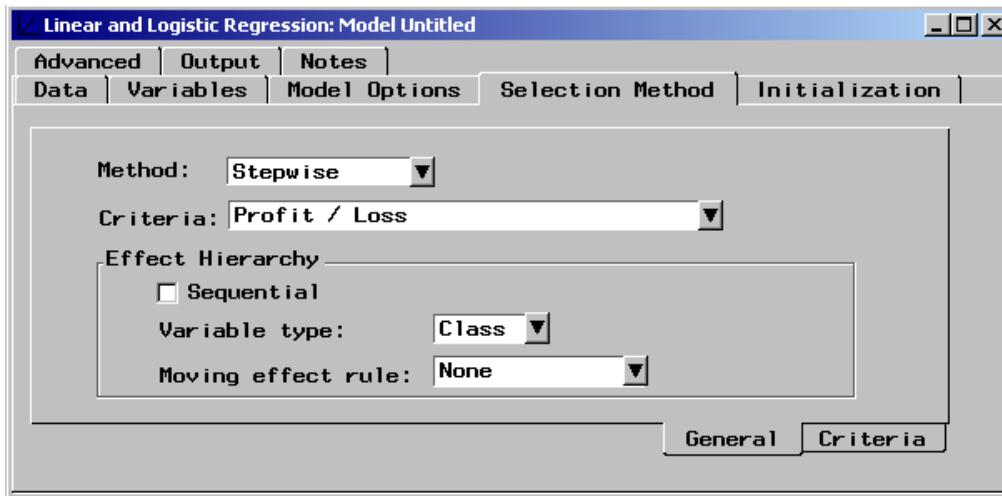


2. Open the Regression node.
3. Find the Tools menu on the top of the session window and select **Tools**  $\Rightarrow$  **Interaction Builder...**. This tool enables you to easily add interactions and higher-order terms to the model, although you do not do so now.



The input variables are shown on the left, and the terms in the model are shown on the right. The Regression node fits a model containing all main effects by default.

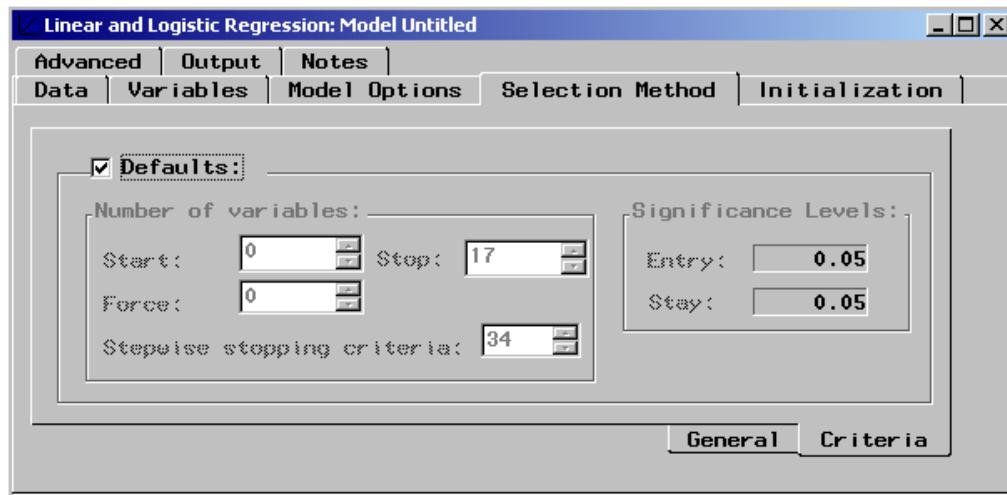
4. Select **Cancel** to close the Interaction Builder window when you are finished inspecting it.
5. Select the **Selection Method** tab. This tab enables you to perform different types of variable selection using various criteria. You can choose backward, forward, or stepwise selection. The default in Enterprise Miner is to construct a model with all input variables that have a status of use.
6. Select **Stepwise** using the arrow next to the Method field.



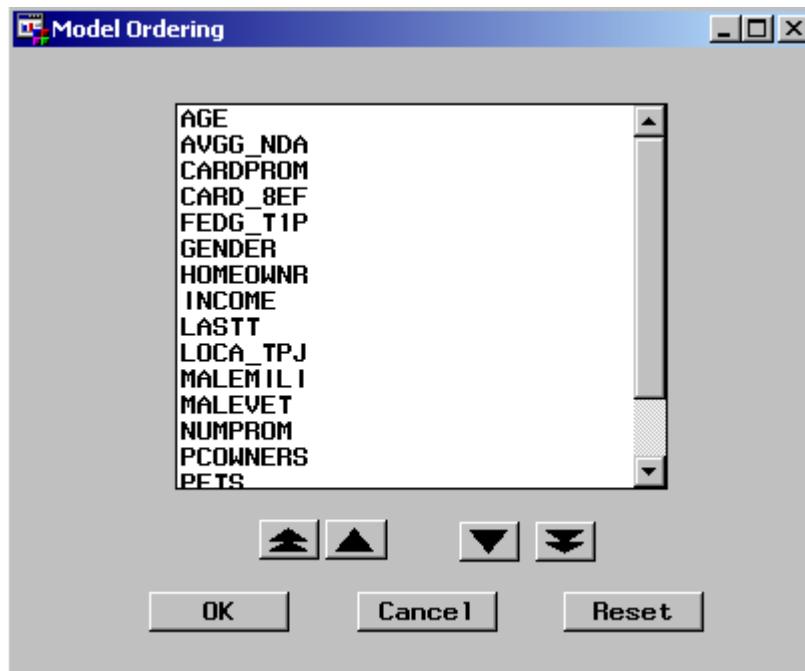
If you choose the **Forward**, **Backward**, or **Stepwise** effect selection method, then you can use the Criteria field to specify a selection criterion to be used to select the final model. The node first performs the effect selection process, which generates a set of candidate models corresponding to each step in the process. Effect selection is done based on the **Entry or Stay Significance Levels** found in the Criteria subtab of the Model Selection tab. Once the effect selection process terminates, the candidate model that optimizes the selection criterion on the validation data set is chosen as the final model.

Inspect the Effect Hierarchy options in the lower-left corner of the window. Model hierarchy refers to the requirement that for any effect in the model, all effects that it contains must also be in the model. For example, in order for the interaction A\*B to be in the model, the main effects A and B must also be in the model. The Effect Hierarchy options enable you to control how a set of effects is entered into or removed from the model during the variable selection process.

7. Select the **Criteria** subtab.



The list of candidate effects can be seen by selecting from the main menu **Tools** ⇒ **Model Ordering....** This opens the Model Ordering window.



The Start value,  $n$ , selects the first  $n$  effects from the beginning of this list as the first model. For the Forward and Stepwise selection methods, the default Start value is 0. For the Backward selection method, the default is the total number of candidate effects.

The Force value,  $n$ , is used to specify the effects that must be included in the final model regardless of the selection method. The first  $n$  variables in the Model Ordering list will be included in every model.

The Stop value for the Forward method is the maximum number of effects to appear in the final model. For the Backward method, the Stop value is the minimum number

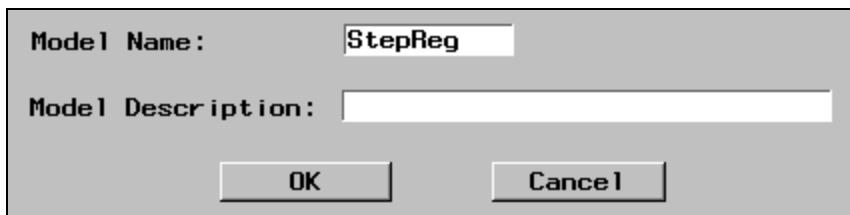
of effects to appear in the final model. For the Backward selection method, the default Stop value is 0. For the Forward selection method, the default is the total number of input variables. The Stop option is not used in the Stepwise selection method.

The Stepwise stopping criteria field enables you to set the maximum number of steps before the Stepwise methods stops. The default is set to twice the number of effects in the model.

The Stepwise method uses cutoffs for the variables entering the model and for the variables leaving the model. The default significance levels are 0.05 for both entry and stay.

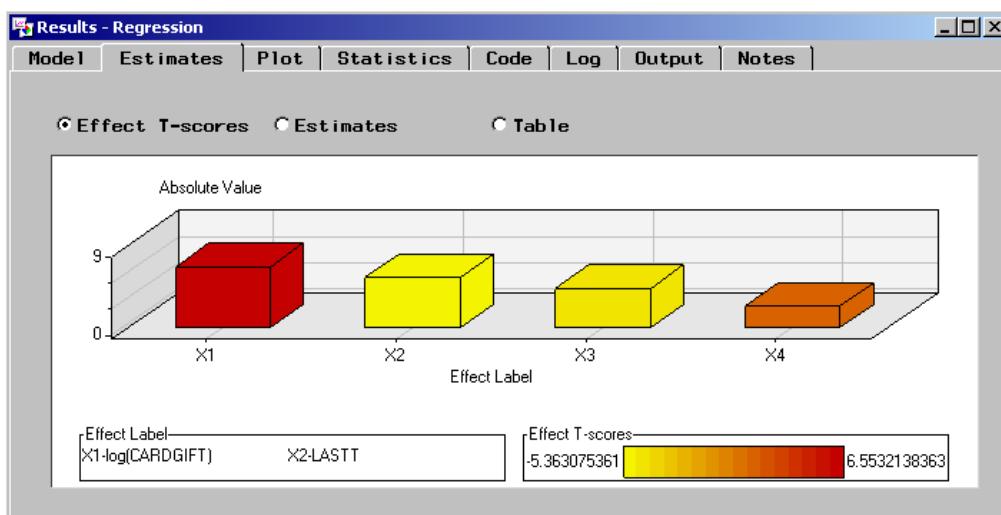
Changing these values may impact the final variables included in the model.

8. Close the Regression node saving the changes when prompted.
9. Because you have changed the default settings for the node, it prompts you to change the default model name. Enter **StepReg** for the model name.



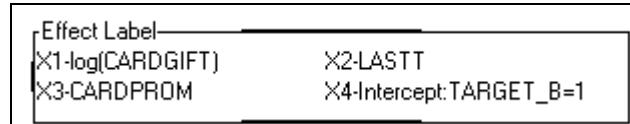
10. Select **OK**.
11. Run the diagram from the regression node.
12. Select **Yes** to view the results when prompted.

The results open with the Estimates tab active and provide a graph of the effect T-scores.

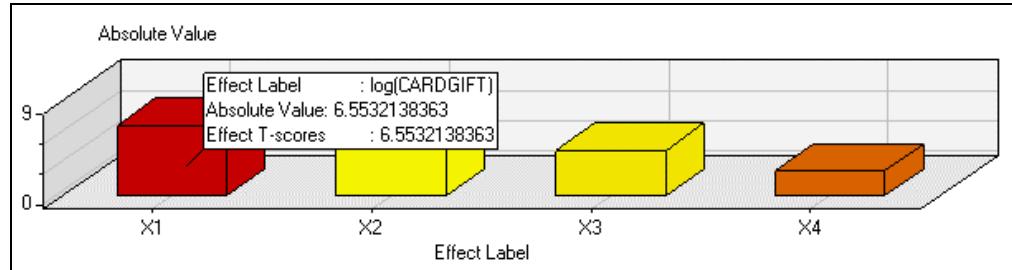


The graph shows the relative significance of the model parameter estimates. There are several ways to determine what variables the bars represent.

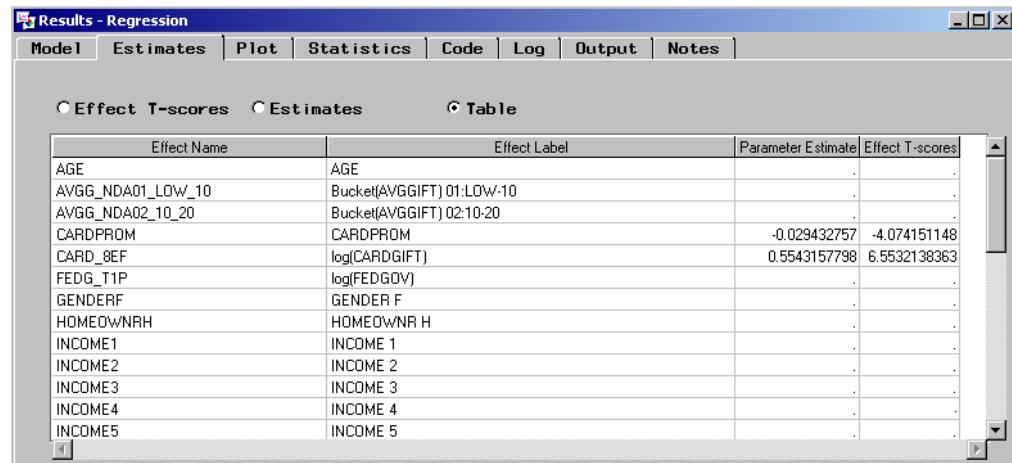
1. You can expand the legend. Select the Move and Resize Legend tool, , point at the dark part of the legend's top border until your cursor becomes a double-headed arrow. Drag the legend border up to make the legend area bigger.



2. As an alternative to making the legend bigger, you can use the View Info tool,  . Select the View Info tool and point and click on any of the bars in the graph. Information about that bar will be displayed.



3. Select the Table radio button in the Estimates tab. Parameter estimates and effect T-scores will only be shown for variables in the selected model.



Regardless of the method chosen to determine the variables in the model and their relative importance, for the model to predict TARGET\_B, the important variables are

- log(CARDGIFT) – the natural log of the donor's gifts to previous card promotions
- LASTT – the elapsed time since the last donation
- CARDPROM – the number of card promotions previously received.

4. Select the **Statistics** tab.

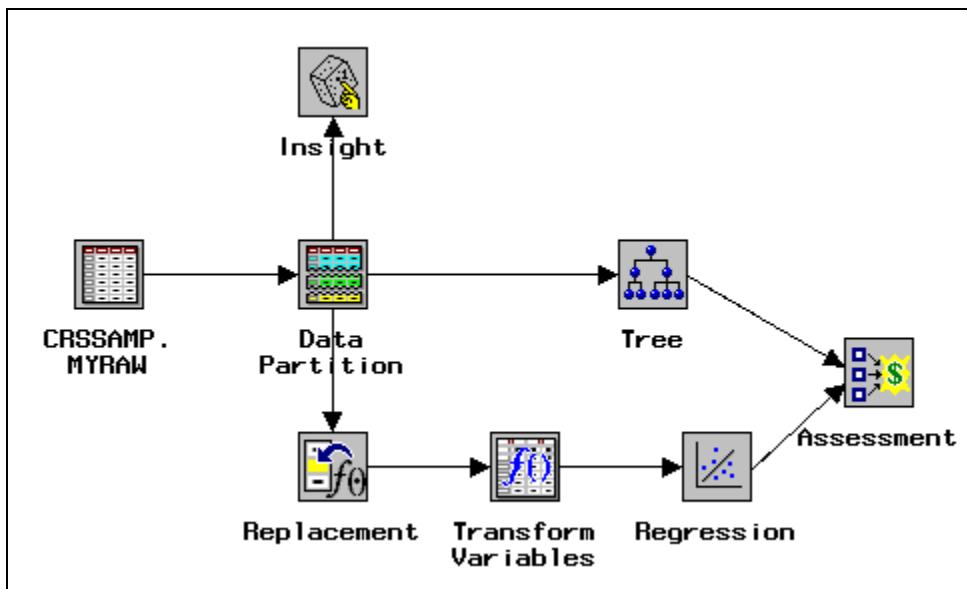
Fit Statistic	Label	Training	Validation	Test
_AIC_	Akaike's Information Criterion	4739.6960985	.	.
_ASE_	Average Squared Error	0.2427303072	0.2400464636	.
_AVERR_	Average Error Function	0.6784766416	0.6729983142	.
_DFE_	Degrees of Freedom for Error	3483	.	.
_DFM_	Model Degrees of Freedom	4	.	.
_DFT_	Total Degrees of Freedom	3487	.	.
_DIV_	Divisor for ASE	6974	6974	.
_ERR_	Error Function	4731.6960985	4693.4902431	.
_FPE_	Final Prediction Error	0.2432878273	.	.
_MAX_	Maximum Absolute Error	0.7851316003	0.7511775148	.
_MSE_	Mean Square Error	0.2430090673	0.2400464636	.
_NOBS_	Sum of Frequencies	3487	3487	.
_NW_	Number of Estimate Weights	4	.	.
RASE	Root Average Sum of Squares	0.4926766762	0.489945368	.

The Statistics tab lists fit statistics, in alphabetical order, for the training data, validation data, and test data analyzed with the regression model. In this example, you only have training and validation data sets.

5. Close the regression results.

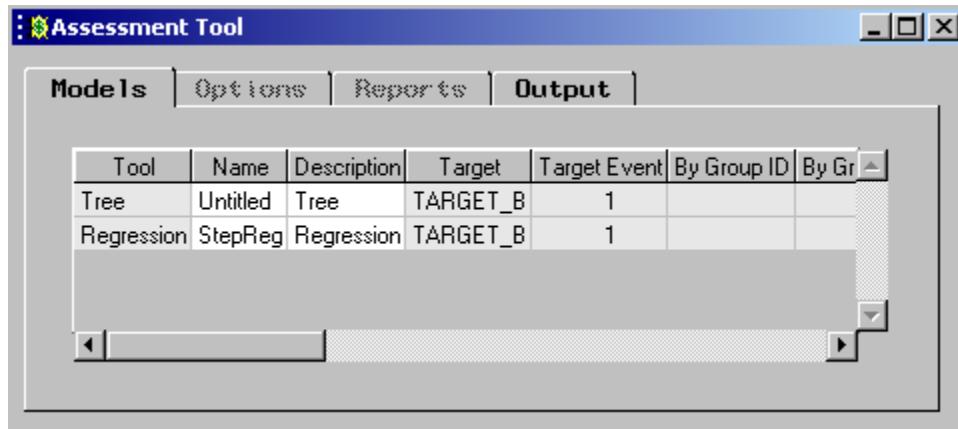
### Fitting a Default Decision Tree

1. Add a default Tree node to the workspace. Connect the Data Partition to the Tree.
2. Add an Assessment node to the workspace and then connect the Tree node and the Regression node to the Assessment node. The flow should now appear like the one pictured below.



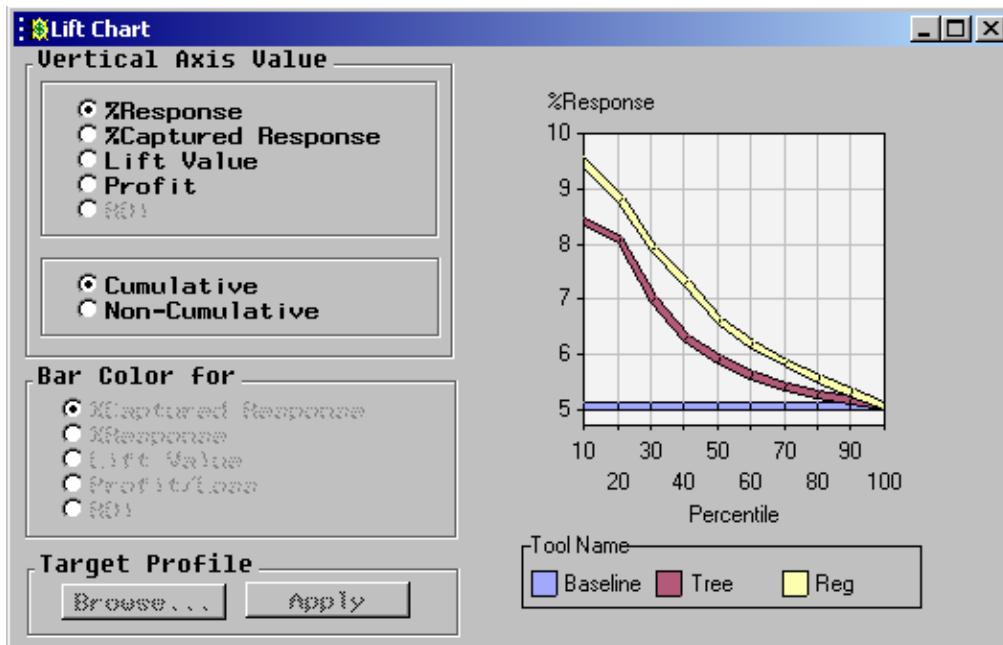
A decision tree handles missing values directly, so it does not need data replacement. Monotonic transformations of interval variables will probably not improve the tree fit because the tree bins numeric variables. The tree may perform worse if you connect it after binning a variable in the Transform Variables node, because binning reduces the splits the tree can consider (unless you include the original variable and the binned variable in the model).

- Run the flow from the Assessment node and select **Yes** when you are prompted to view the results. The Assessment node opens with two models displayed.



You can change the name for the model by editing the Name column. This feature is especially useful when you fit several competing models of the same type.

- Enter the name **DefTree** in the Name column for the Tree tool to indicate that you have fit a default tree.
- To generate a lift chart, highlight both rows in the Assessment node. You can do this by selecting the row for one model and then Ctrl-clicking on the row for the other model. You can also drag through both rows to highlight them simultaneously.
- Select **Tools**  $\Rightarrow$  **Lift Chart** to compare how the models perform on the validation data set.



Observe that the regression model outperforms the default tree throughout the graph.

7. Close the Assessment node when you have finished inspecting the various lift charts.



# Chapter 4 Variable Selection

4.1 Variable Selection and Enterprise Miner .....	4-3
---	-----



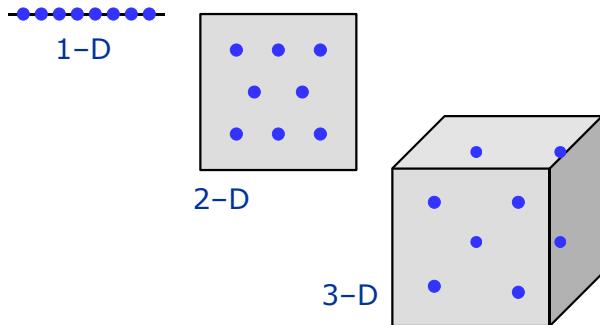
## 4.1 Variable Selection and Enterprise Miner

### Objectives

- Discuss the need for variable selection.
- Explain the methods of variable selection available in Enterprise Miner.
- Demonstrate the use of different variable selection methods.

2

### The Curse of Dimensionality



3

Recall that as the number of input variables to a model increases there is an exponential increase in the data required to densely populate the model space. If the modeling space becomes too sparsely populated, the ability to fit a model to noisy (real) data is hampered. In many cases although the number of input variables is large, some variables may be redundant and others irrelevant.

The difficulty is in determining which variables can be disregarded in modeling without leaving behind important information.

## Methods of Variable Selection

- Stepwise Regression
- Decision Trees
- Variable Selection Node

4

Three possible methods of variable selection available in Enterprise Miner are

- stepwise regression
- decision trees
- the variable selection node.

 Another possible method for dimension reduction is principal components analysis. Principal components are uncorrelated linear combinations of the original input variables; they depend on the covariance matrix or the correlation matrix of the original input variables.

## Stepwise Regression

- Uses multiple regression  $p$ -values to eliminate variables.
- May not perform well with many potential input variables.

5

As you saw earlier, stepwise regression methods can be used for variable selection. However, this method was not designed for use in evaluating data sets with dozens (or hundreds) of potential input variables and may not perform well under such conditions.

## Decision Trees

- Grow a large tree.
- Retain only the variables important in growing the tree for further modeling.

6

In the tree node, a measure of variable importance is calculated. The measure incorporates primary splits and any saved surrogate splits in the calculation. The importance measures are scaled between 0 and 1, with larger values indicating greater importance. Variables that do not appear in any primary or saved surrogate splits have zero importance.

By default, variables with importance less than 0.05 are given a model role of rejected, and this status is automatically transferred to subsequent nodes in the flow.

No particular tree-growing or pruning options are preferable for variable selection. However, in pruning, it is usually better to err on the side of complexity as opposed to parsimony. Severe pruning often results in too few variables being selected. When presented with a range of trees of similar performance, selecting a bushier tree is often more useful.

## Variable Selection Node

Selection based on one of two criteria:

- R-square
- chi-square – for binary targets only.

7

The variable selection node provides selection based on one of two criteria. By default, the node removes variables unrelated to the target.

When you use the R-square variable selection criterion, a three-step process is followed:

1. Enterprise Miner computes the squared correlation for each variable and then assigns the rejected role to those variables that have a value less than the squared correlation criterion (default 0.005).
2. Enterprise Miner evaluates the remaining (not rejected) variables using a forward stepwise  $R^2$  regression. Variables that have a stepwise  $R^2$  improvement less than the threshold criterion (default 0.0005) are assigned the rejected role.
3. For binary targets, Enterprise Miner performs a final logistic regression using the predicted values that are output from the forward stepwise regression as the only input variable.

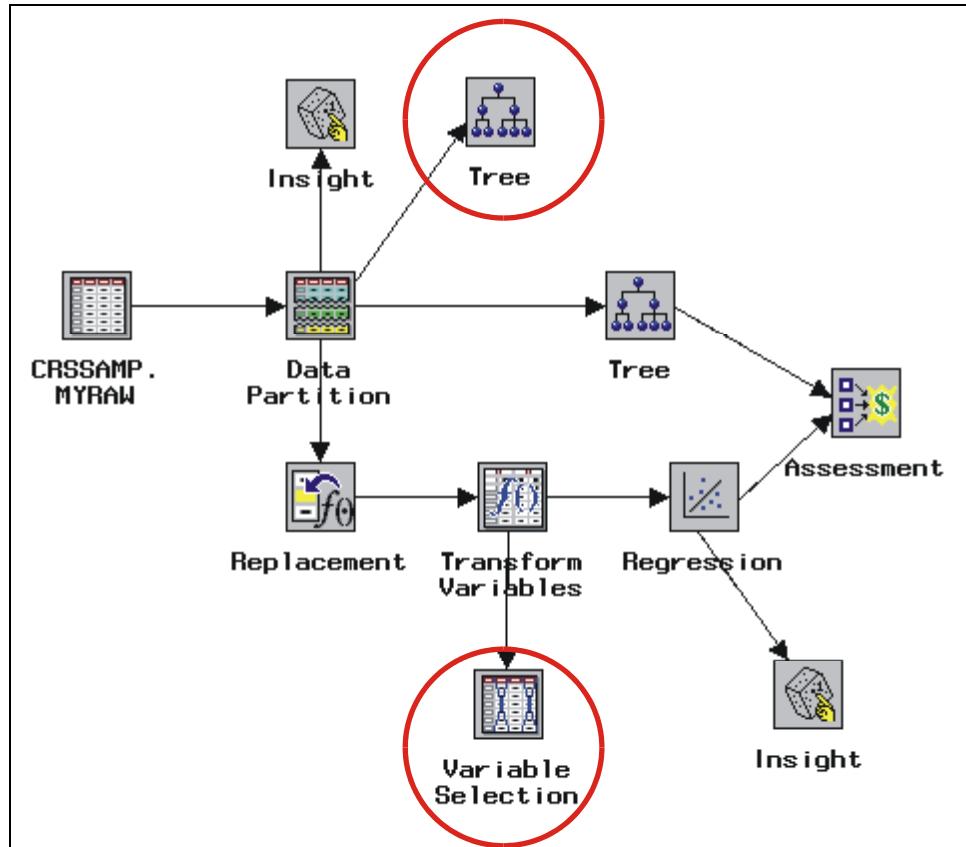
If the target is nonbinary, only the first two steps are performed.

When you use the chi-square selection criterion, variable selection is performed using binary splits for maximizing the chi-square value of a 2x2 frequency table. Each level of the ordinal or nominal variables is decomposed into binary dummy variables. The range of each interval variable is divided into a number of categories for splits. These bins are equally sized intervals and, by default, interval inputs are binned into 50 levels.



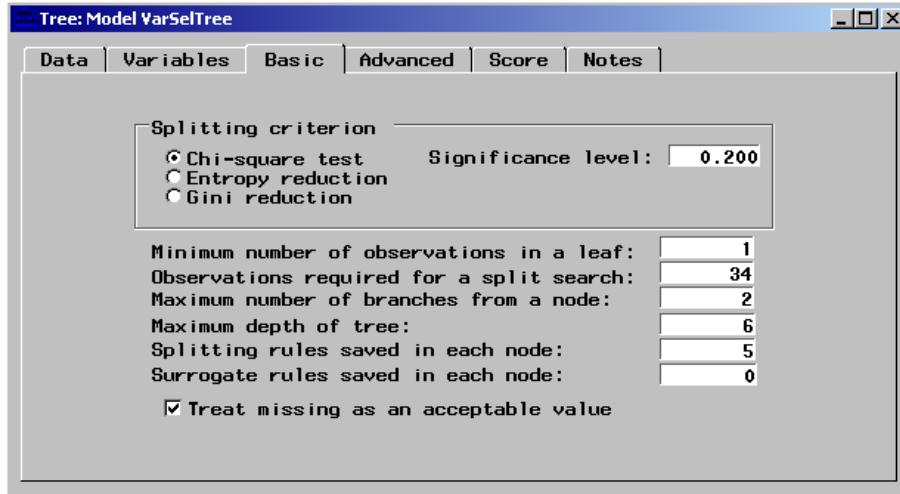
## Variable Selection with Decision Trees and the Variable Selection Node

Return to the nonprofit diagram created in Chapter 3, “Predictive Modeling Using Regression.” Add a Tree node after the Data Partition node, and add a Variable Selection node after the Transform Variables node. Your workspace should appear as shown below:

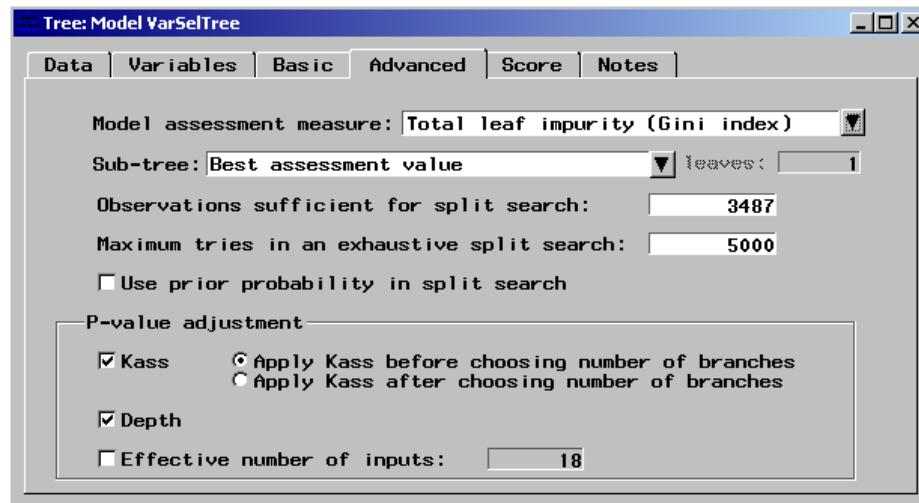


### Variable Selection Using a Decision Tree

1. Open the new Tree node.
2. Select the **Basic** tab.

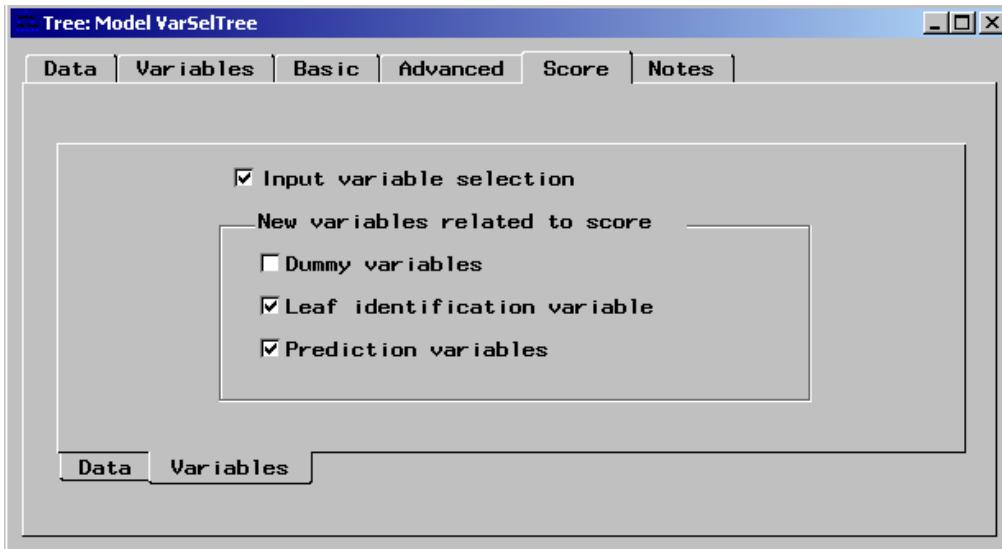


3. Select **Gini reduction** for the splitting criterion.
4. Change the maximum depth of the tree to **8** to allow a larger tree to grow.
5. Select the **Advanced** tab and change the model assessment measure to **Total Leaf Impurity (Gini index)**.

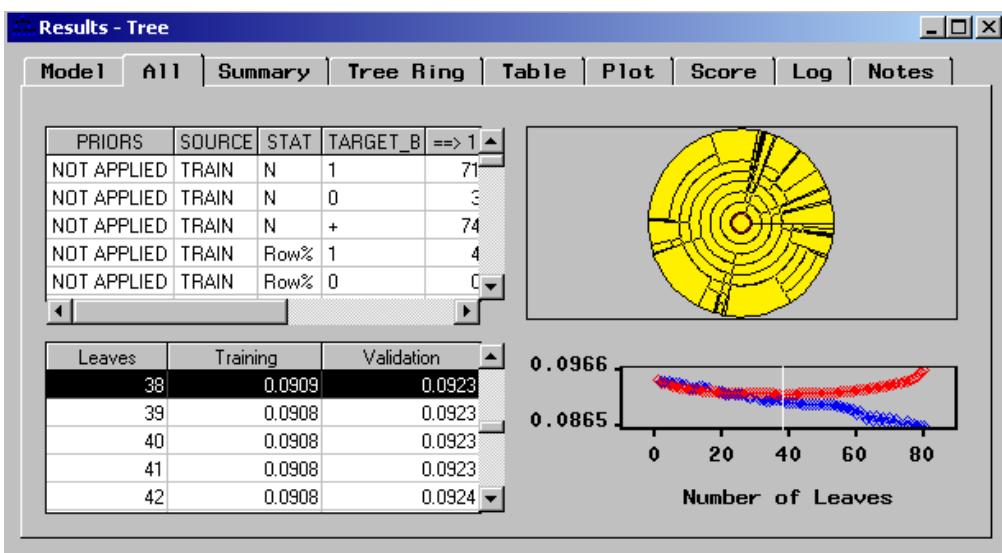


These selections for splitting criterion and model assessment measure are chosen because they tend to result in larger (bushier) trees, which is desirable for variable selection.

6. Select the **Score** tab and then the **Variables** subtab. This tab controls in part how the Tree node will modify the data set when the node is run.



7. Close the tree node, saving changes when prompted. It is unnecessary to specify a different name for the model, but you may do so if desired when prompted. Select **OK** to exit.
8. Run the flow from the Tree node and select **Yes** to view the results when prompted.



The 38-leaf tree has the lowest total impurity on the validation data set, so it is selected by default.

9. Select the **Score** tab and then select the **Variable Selection** subtab. This subtab shows you which variables have been retained and which have been rejected.

The screenshot shows the 'Results - Tree' dialog box with the 'Score' tab selected. A sub-dialog titled 'Variable Selection' is open, containing a table with the following data:

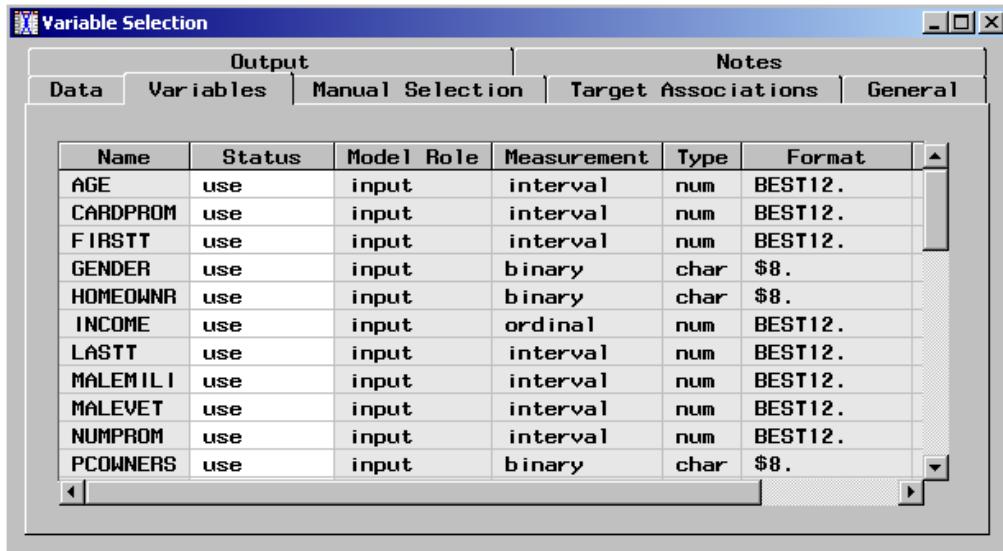
Name	Importance	Role	Rules
AVGGIFT	1.0000	input	5
TIMELAG	0.7395	input	2
LASTT	0.6401	input	3
INCOME	0.6350	input	1
MALEVET	0.5337	input	2
CARDPROM	0.4774	input	3
CARDGIFT	0.4319	input	3
NUMPROM	0.3830	input	4
HOMEOWNR	0.2692	input	1
FIRSTT	0.2250	input	2
AGE	0.2217	input	2
MALEMILI	0.2198	input	2
LOCALGOV	0.2176	input	4
FEDGOV	0.1396	input	1
STATEGOV	0.1287	input	2
GENDER	0.0000	rejected	0
PETS	0.0000	rejected	0
PCOWNERS	0.0000	rejected	0

The chosen tree retains 15 of the 18 variables for analysis. You could add a Regression node or Neural Network node to the flow following this Tree node. However, because no data replacement or variable transformations have been performed, you should consider doing these things first (for the input variables identified by the variable selection). In general, a tree with more leaves retains a greater number of variables, whereas a tree with fewer leaves retains a smaller number of variables.

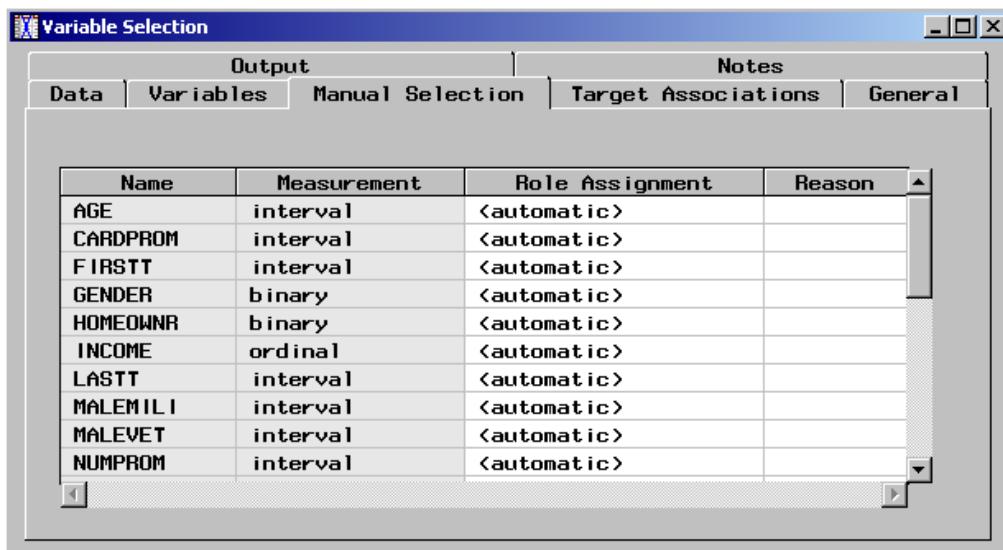
10. Close the tree results.

## Variable Selection Using the Variable Selection Node

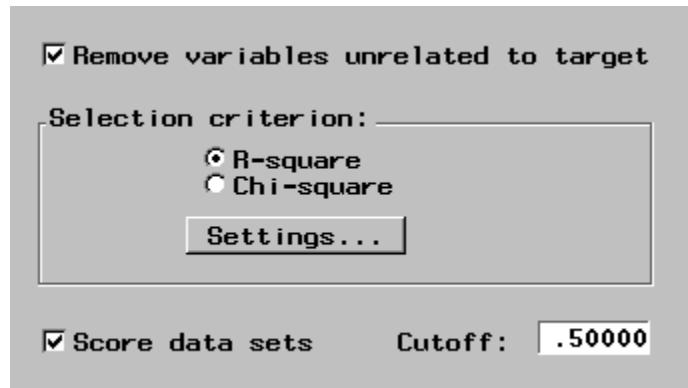
1. Open the Variable Selection node.



2. Select the **Manual Selection** tab. This tab enables you to force variables to be included or excluded from future analyses. By default, the role assignment is automatic, which means that the role is set based on the analysis performed in this node.

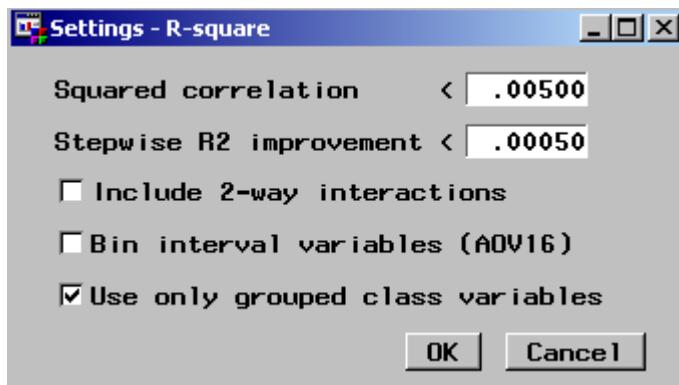


3. Select the **Target Associations** tab. This tab enables you to choose one of two selection criteria and specify options for the chosen criterion. By default, the node removes variables unrelated to the target (according to the settings used for the selection criterion) and scores the data sets.



### Selection Using the R-square Criterion

1. Consider the settings associated with the default R-square criterion first. Because R-square is already selected as the selection criterion, click on the [Settings...](#) button on the Target Association tab.



Recall that a three-step process is performed when you apply the  $R^2$  variable selection criterion to a binary target. The Setting window allows you to specify the cutoff squared correlation measure and the necessary  $R^2$  improvement for a variable to remain as an input variable.

Additional available options enable you to

- test 2-way interactions. When this option is selected, Enterprise Miner evaluates 2-way interactions for categorical inputs. If this option is not selected, Enterprise Miner still evaluates the 2-way interactions, but it does not allow them to be passed to successor nodes as input variables.
- bin interval variables in up to 16 bins. When selected, this option requests Enterprise Miner to bin interval variables into 16 equally spaced groups (AOV16). The AOV16 variables are created to help identify nonlinear relationships with the target. Bins with zero observations are eliminated, meaning an AOV16 variable can have fewer than 16 bins

- use only grouped class variables. When this option is selected, Enterprise Miner uses only the grouped class variable to evaluate variable importance. If this option is not selected, Enterprise Miner uses the grouped class variable as well as the original class variable in evaluating variable importance. This may greatly increase processing time.
- Leave the default settings and close the node.
  - Run the flow from the Variable Selection node and view the results.
  - Click on the **Role** column heading to sort the variable by their assigned roles. Then click on the **Rejection Reason** column heading. Inspect the results.

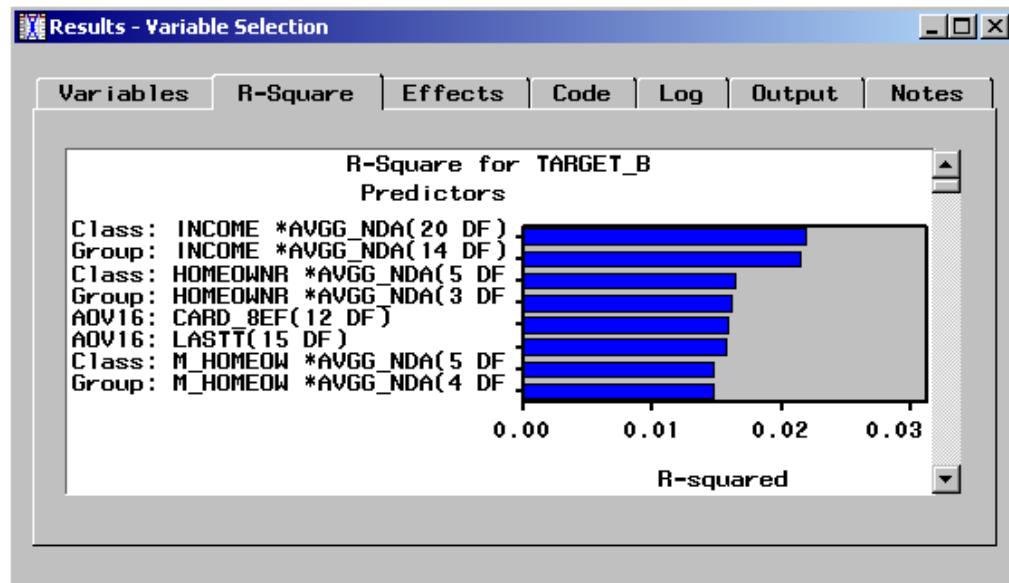
**Results - Variable Selection**

The screenshot shows a software interface titled "Results - Variable Selection". At the top, there is a menu bar with options: File, Edit, View, Insert, Tools, Options, Help, and Exit. Below the menu bar is a toolbar with icons for New, Open, Save, Print, and others. The main area contains a table with the following columns: Name, Role, and Rejection Reason. The table lists nine variables:

Name	Role	Rejection Reason
LASTT	input	
CARD_8EF	input	
G_AVGG_NDA	input	
AVGG_NDA	rejected	Group variable G_AVGG_NDA pre
AGE	rejected	Low R2 w/ target
CARDPROM	rejected	Low R2 w/ target
GENDER	rejected	Low R2 w/ target
HOMEOWNR	rejected	Low R2 w/ target
INCOME	rejected	Low R2 w/ target
MALEMILI	rejected	Low R2 w/ target

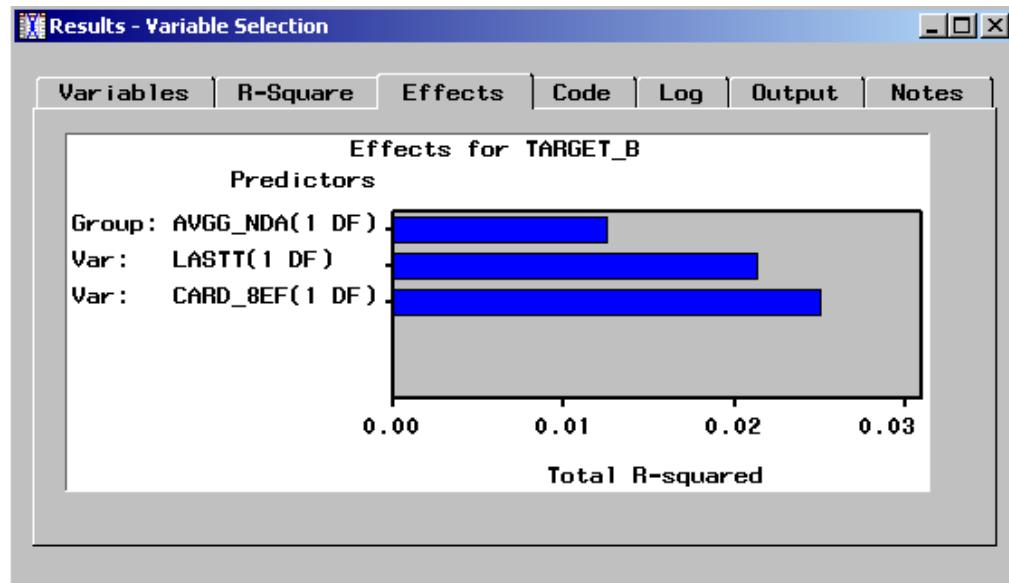
LASTT, CARD8EF, and G\_AVGG\_NDA are retained. If you scroll to the right to look at the Label column, you see that CARD8EF is the log of CARDGIFT and G\_AVGG\_NDA is the variable AVGGIFT bucketed. The variable CARD8EF was created in the Transform Variables node, and G\_AVGG\_NDA was created here in the Variable Selection node.

5. Select the **R-square** tab.



This tab shows the R-square for each effect with TARGET\_B. Note that some of the effects are interaction effects that will not be passed to the successor node.

6. Select the **Effects** tab.

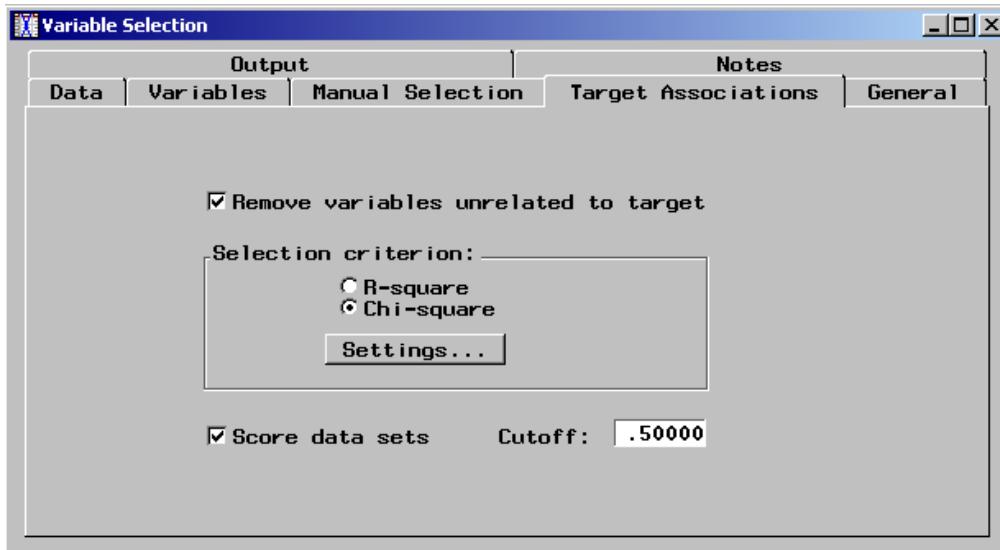


This tab shows the total R-squared as each selected variable is added into the model.

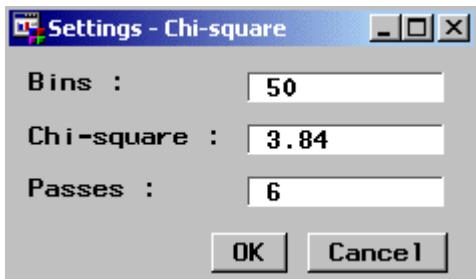
7. Close the Results window.

### Selection Using the Chi-square Criterion

1. Open the Variable Selection node. Select the **Target Associations** tab and then select the **Chi-square** criterion.



2. Click on the **Settings...** button and inspect the options.



As discussed earlier, variable selection is performed using binary splits for maximizing the chi-square values of a 2x2 frequency table. There are three settings that control parts of this process:

- |            |   |
|------------|---|
| Bins       | This option determines the number of categories in which the range of each interval variable is divided for splits. By default, interval inputs are binned into 50 levels.  |
| Chi-square | This option governs the number of splits that are performed. This value is a minimum bound for the chi-square value to decide whether it is eligible for making a variable split. By default, the chi-square value is set to 3.84. As you increase the chi-square value, the procedure performs fewer splits. |
| Passes     | By default, the node makes 6 passes through the data to determine the optimum splits.   |

3. Select **Cancel** to close the Settings window without making any changes.
4. Close the Variable Selection node. Select **Yes** to save changes when prompted.

5. Rerun the flow from the Variable Selection node and select **Yes** to view the results when prompted.
6. Click on the **Role** column heading to sort the variable by their assigned roles. Then click on the **Rejection Reason** column heading. Inspect the results.

**Results - Variable Selection**

**Notes**

Variables	R-Square	Effects	Code	Log	Output
Name	Role	Rejection Reason			
AGE	input				
FIRSTT	input				
HOMEOWNR	input				
INCOME	input				
LASTT	input				
MALEMILI	input				
MALEVET	input				
NUMPROM	input				
PETS	input				
AVGG_0TM	input				
CARD_W72	input				
FEDG_NDJ	input				
LOCA_G0X	input				
STAT_1EA	input				
TIME_FV6	input				
CARDPROM	rejected	Small chi-square			
GENDER	rejected	Small chi-square			
PCOWNERS	rejected	Small chi-square			
M_AGE	rejected	Small chi-square			
M_TIMELA	rejected	Small chi-square			
M_GENDER	rejected	Small chi-square			
M_HOMEOW	rejected	Small chi-square			
M_INCOME	rejected	Small chi-square			
M_PCOWNE	rejected	Small chi-square			
M_PETS	rejected	Small chi-square			

Fifteen variables have been retained as input variables.

If the resulting number of variables is too high, consider increasing the chi-square cutoff value. Increasing the chi-square cutoff value generally reduces the number of retained variables.

# Chapter 5 Predictive Modeling Using Neural Networks

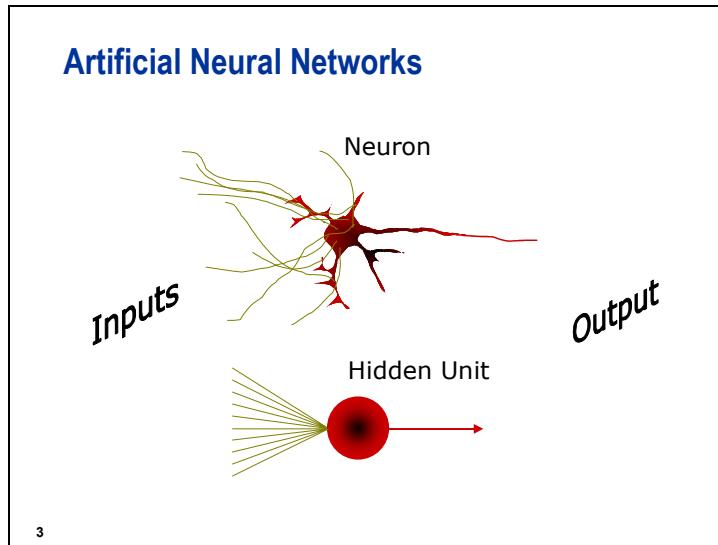
<b>5.1 Introduction to Neural Networks .....</b>	<b>5-3</b>
<b>5.2 Visualizing Neural Networks .....</b>	<b>5-9</b>



## 5.1 Introduction to Neural Networks

### Objectives

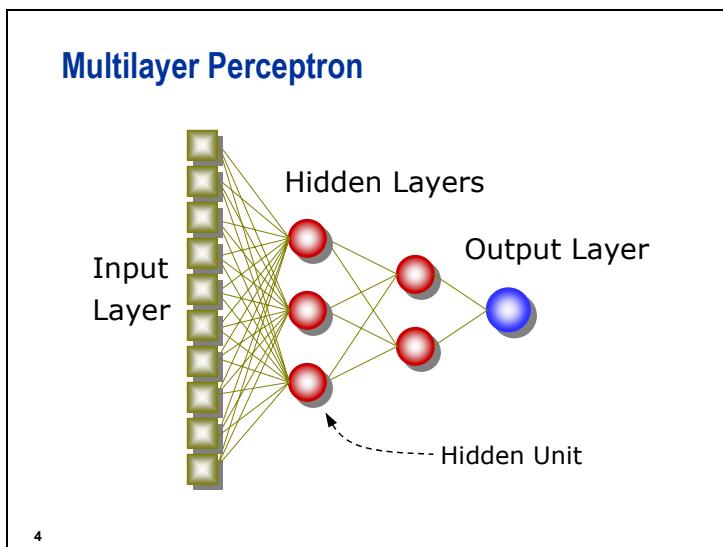
- Define a neural network.
- List the components of a neural network.
- Define an activation function.
- Discuss the concept of an optimization algorithm.



An *organic* neural network has 10 billion highly interconnected neurons acting in parallel. Each neuron may receive electrochemical signals (through synapses) from as many as 200,000 other neurons. These connections can be altered by environmental stimuli. If the right signal is received by the inputs, the neuron is activated and sends inhibitory or excitatory signals to other neurons.

In data analysis, artificial neural networks are a class of flexible nonlinear models used for supervised prediction problems. Yet, because of the ascribed analogy to neurophysiology, they are usually perceived to be more glamorous than other (statistical) prediction models.

The basic building blocks of an artificial neural network are called *hidden units*. Hidden units are modeled after the neuron. Each hidden unit receives a linear combination of input variables. The coefficients are called the (synaptic) weights. An activation function transforms the linear combinations and then outputs them to another unit that can then use them as inputs.



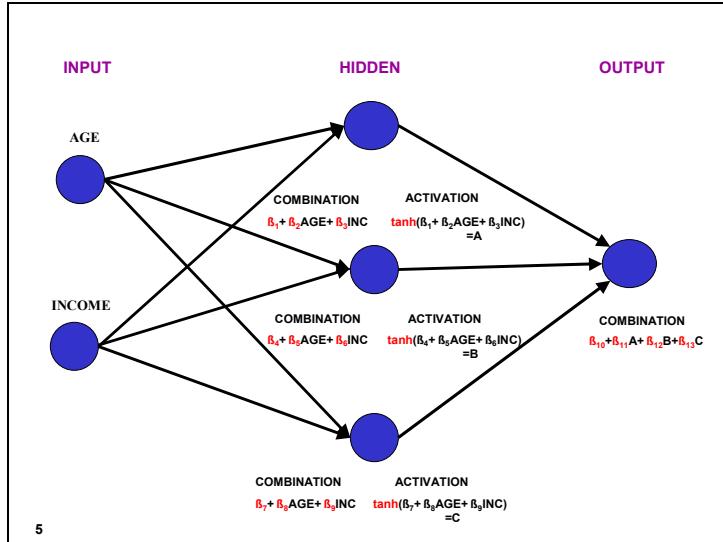
An *artificial* neural network is a flexible framework for specifying a variety of models. The most widely used type of neural network in data analysis is the *multilayer perceptron* (MLP). A MLP is a feed-forward network composed of an input layer, hidden layers composed of hidden units, and an output layer.

The input layer is composed of units that correspond to each input variable. For nominal inputs with  $C$  levels,  $C-1$  input units will be created. Consequently, the number of input units may be greater than the number of inputs.

The hidden layers are composed of hidden units. Each hidden unit outputs a nonlinear function of a linear combination of its inputs – the *activation function*.

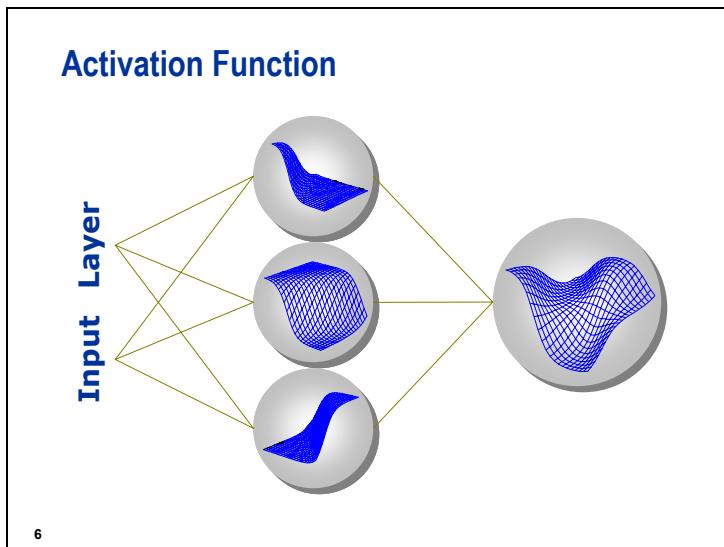
The output layer has units corresponding to the target. With multiple target variables or multiclass ( $>2$ ) targets, there are multiple output units.

The network diagram is a representation of an underlying statistical model. The unknown parameters (weights and biases) correspond to the connections between the units.

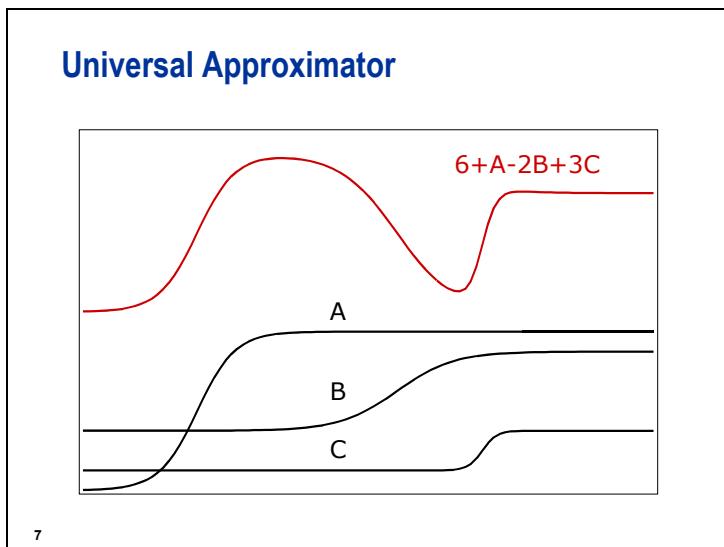


Each hidden unit outputs a nonlinear transformation of a linear combination of their inputs. The linear combination is the net input. The nonlinear transformation is the activation function. The activation functions used with MLPs are sigmoidal curves (surfaces).

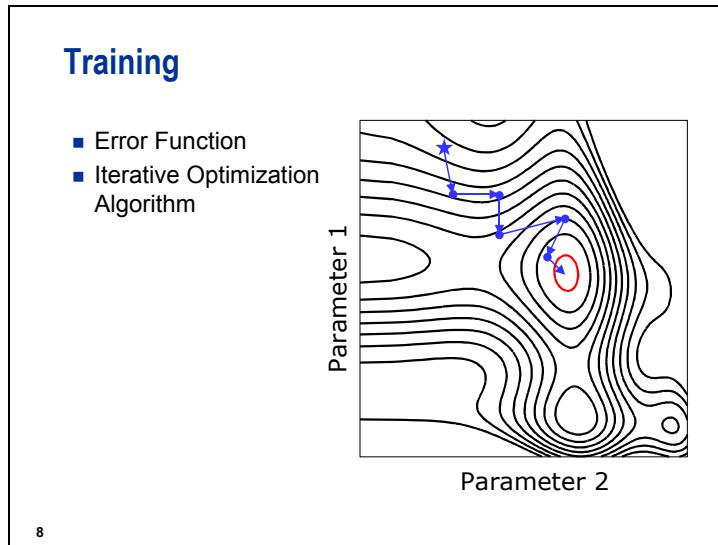
A hidden layer can be thought of as a new (usually) lower-dimensional space that is a nonlinear combination of the previous layer. The output from the hidden units is linearly combined to form the input of the next layer. The combination of nonlinear surfaces gives MLPs their modeling flexibility.



An output activation function is used to transform the output into a suitable scale for the expected value of the target. In statistics, this function is called the inverse *link function*. For binary targets, the logistic function is suitable because it constrains the output to be between zero and one (the expected value of a binary target is the posterior probability). The logistic function is sometimes used as the activation function for the hidden units as well. This sometimes gives the false impression that they are related. The choice of output activation function depends only on the scale of the target.



An MLP with one hidden layer is a *universal approximator*. That is, it can theoretically approximate any continuous surface to any degree of accuracy (for some number of hidden units). In practice, a MLP may not achieve this level of flexibility because the weights and biases need to be estimated from the data. Moreover, the number of hidden units that are required for approximating a given function might be enormous.



A regression model, such as an MLP, depends on unknown parameters that must be estimated using the data. Estimating the weights and biases (parameters) in a neural network is called *training the network*. The *error function* is the criterion by which the parameter estimates are chosen (learned). Every possible combination of parameter estimates corresponds to a prediction of the expected target. Error functions can be thought of as measures of the distance between these predictions and the actual data. The objective is to find the set of parameter estimates that optimize (minimize) the error function.

For some simple regression models, explicit formulas for the optimal estimates can be determined. Finding the parameter values for neural networks, however, is more difficult. Iterative numerical optimization methods are used. First, starting values are chosen. The starting values are equivalent to an initial guess at the parameter values. These values are updated to improve the estimates and reduce the error function. The updates continue until the estimates converge (in other words, there is no further progress).

Optimization can be thought of as searching for a global optimum (minimum) on a multidimensional surface. The contours of the above surface represent level values of the error function. Every pair of values of the two parameters is a location on the surface. There are many algorithms for determining the direction and distance of the update step.

Multiple minima, saddle points, flat regions, and troughs can complicate the optimization process.

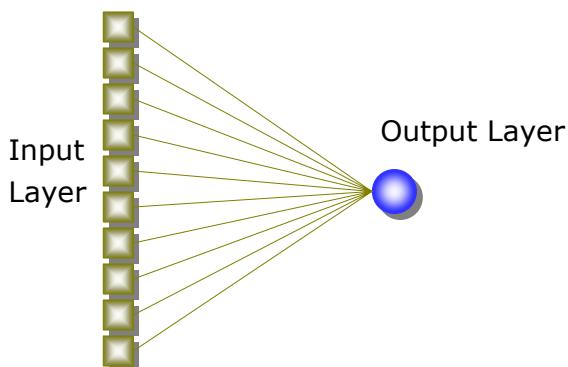
## 5.2 Visualizing Neural Networks

### Objectives

- Relate a generalized linear model and logistic regression to neural networks.
- Fit a neural network using Enterprise Miner.

10

### Generalized Linear Models

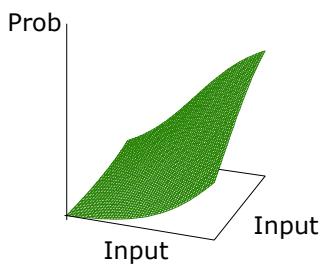


11

Generalized linear models can be represented as feed-forward neural networks without any hidden layers. Standard linear regression (continuous target) and logistic regression (binary target) are important special cases. The simple structure makes them easier to interpret and less troublesome to train.

## Logistic Regression/Discrimination

$\text{logit}(\text{probability of target event}) = \text{linear combination of the inputs}$



12

The simplicity of the linear-logistic model makes it attractive but also limits its flexibility. The effect of each input on the logit is assumed to be linear and assumed not to interact with the other inputs. For example, a unit increase in an input variable corresponds to the same constant increase in the logit for all values of the other inputs.

## The Scenario

- Determine who will respond to a mail promotion.
- The target variable is a binary variable that indicates whether an individual responded to a recent promotion.
- The input variables are items such as age, income, marital status, and number of purchases in the last six months.

13

The BUY data set consists of 10,000 customers and whether or not they responded to a recent promotion (RESPOND). On each customer, 12 input variables were recorded. The variables in the data set are shown below:

Name	Model Role	Measurement Level	Description
RESPOND	Target	Binary	1=responded to promotion, 0=did not respond
AGE	Input	Interval	Age of individual in years
INCOME	Input	Interval	Annual income in thousands of dollars
MARRIED	Input	Binary	1=married, 0=not married
FICO	Input	Interval	Credit score from outside credit agency
GENDER	Input	Binary	F=Female, M=Male
OWNHOME	Input	Binary	1=owns home, 0=does not own home
LOC	Input	Nominal	Location of residence coded A through H
BUY6	Input	Interval	Number of purchases in the last 6 months
BUY12	Input	Interval	Number of purchases in the last 12 months
BUY18	Input	Interval	Number of purchases in the last 18 months
VALUE24	Input	Interval	Total value of purchases in the past 24 months
COA6	Input	Binary	Change of address in the last 6 months (1=address changed, 0=address did not change)

The analysis goal is to build a model that can predict the target (RESPOND) from the inputs. This model can then be used to find new customers to target for a similar promotion.

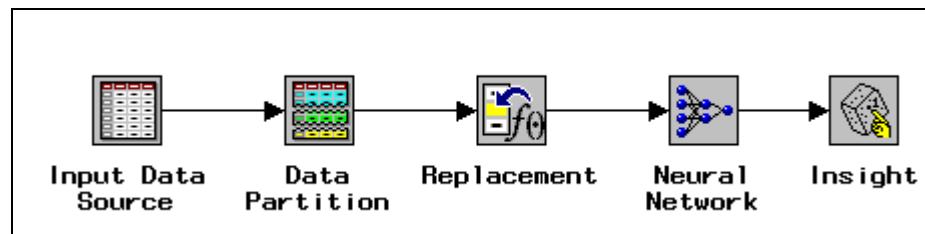


## Fitting a Neural Network Model

To allow visualization of the output from an MLP, a network will be constructed with only two inputs. Two inputs permit direct viewing of the trained prediction model and speed up training.

### Select the Data

1. To insert a new diagram in the course project, select **File**  $\Rightarrow$  **New**  $\Rightarrow$  **Diagram**.
2. Assemble the diagram shown below.



3. To select the data for this example, open the Input Data Source node.
4. Select the **BUY** data set from the CRSSAMP library.
5. Set the model role of RESPOND to **target**.
6. Set the model role of all other variables except AGE and INCOME to **rejected**. The model role of AGE and INCOME should be **input**.

Input Data Source				
	Data	Variables	Interval Variables	Class Variables
	Name	Model Role	Measurement	Type
RESPOND	target	binary	num	
AGE	input	interval	num	
INCOME	input	interval	num	
GENDER	rejected	binary	char	
MARRIED	rejected	binary	num	
FICO	rejected	interval	num	
OWNHOME	rejected	binary	num	
LOC	rejected	nominal	char	
BUY6	rejected	ordinal	num	
BUY12	rejected	ordinal	num	
BUY18	rejected	ordinal	num	
VALUE24	rejected	interval	num	
COA6	rejected	binary	num	

7. Select the **Interval Variables** tab. Note that there are very few missing values for the variables AGE and INCOME.

- Close and save changes to the Input Data Source node.

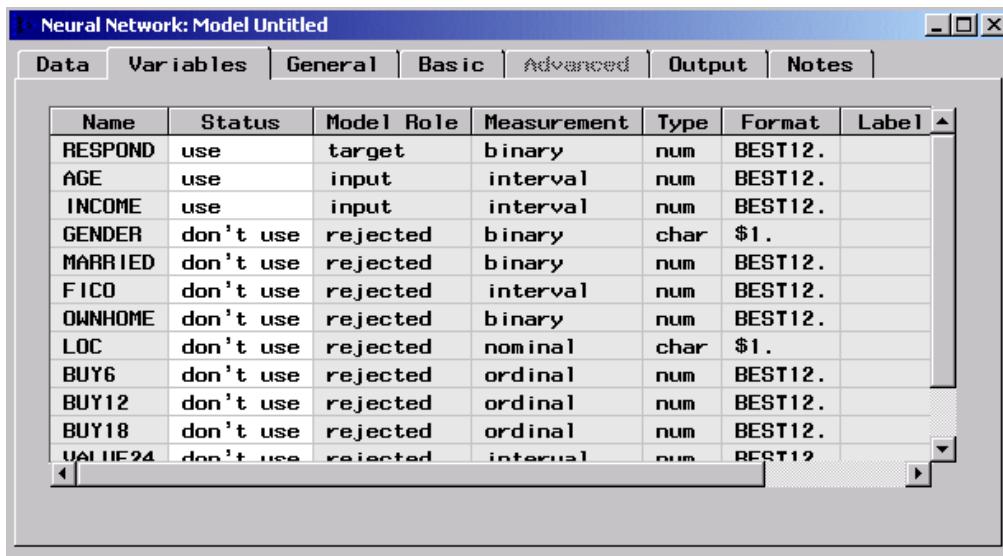
### Partition the Data

- To partition the data, open the Data Partition node.
- Set Train to **40**, Validation to **60**, and Test to **0**.
- Close and save changes to the Data Partition node.

 The Replacement node will be used with the default settings because there are so few missing values.

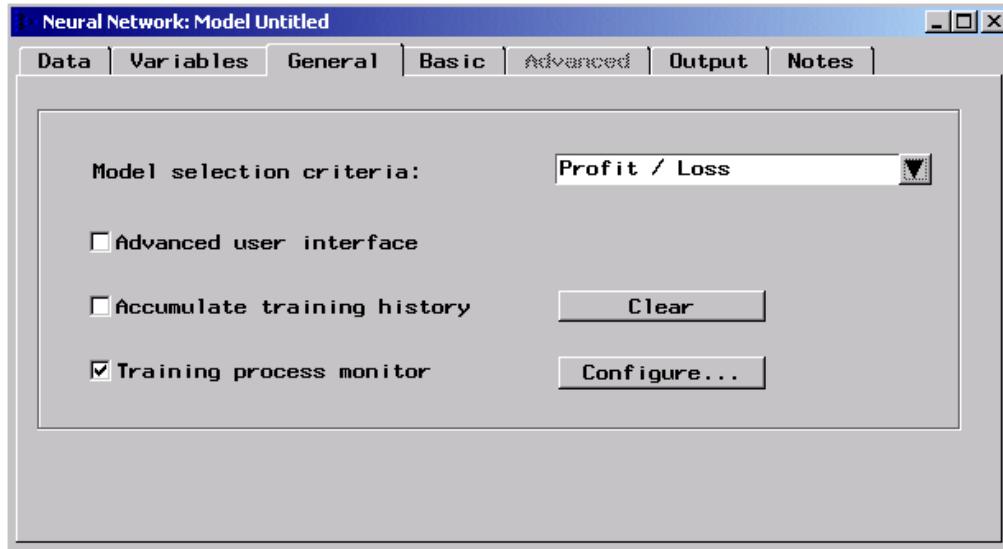
### Construct the Multilayer Perceptron

- Open the Neural Network node. The Variables tab is active.



Name	Status	Model Role	Measurement	Type	Format	Label
RESPOND	use	target	binary	num	BEST12.	
AGE	use	input	interval	num	BEST12.	
INCOME	use	input	interval	num	BEST12.	
GENDER	don't use	rejected	binary	char	\$1.	
MARRIED	don't use	rejected	binary	num	BEST12.	
FICO	don't use	rejected	interval	num	BEST12.	
OWNHOME	don't use	rejected	binary	num	BEST12.	
LOC	don't use	rejected	nominal	char	\$1.	
BUY6	don't use	rejected	ordinal	num	BEST12.	
BUY12	don't use	rejected	ordinal	num	BEST12.	
BUY18	don't use	rejected	ordinal	num	BEST12.	
VALUE24	don't use	rejected	interval	num	BEST12	

- Select the General tab.



Model selection criteria: Profit / Loss

Advanced user interface

Accumulate training history

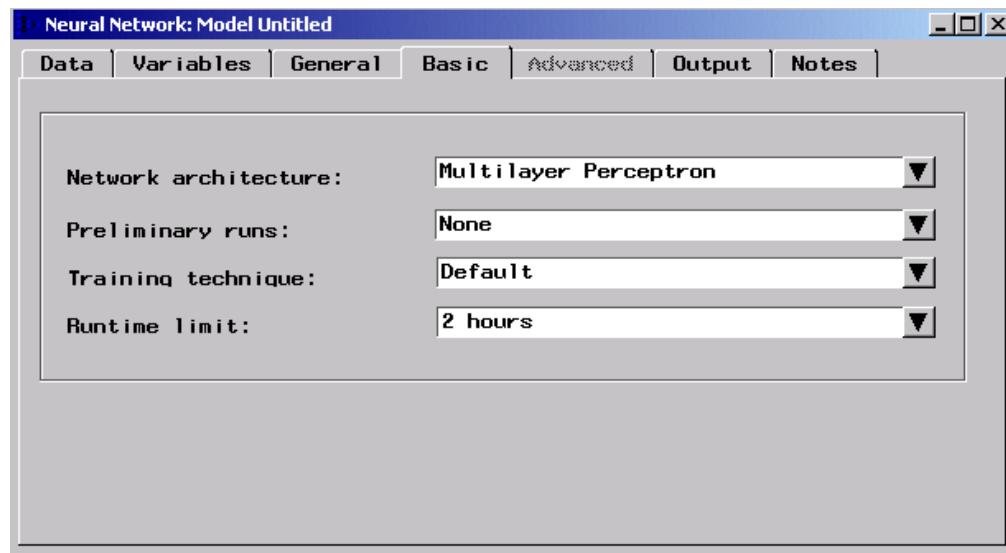
Training process monitor

You can specify one of the following criteria for selecting the best model:

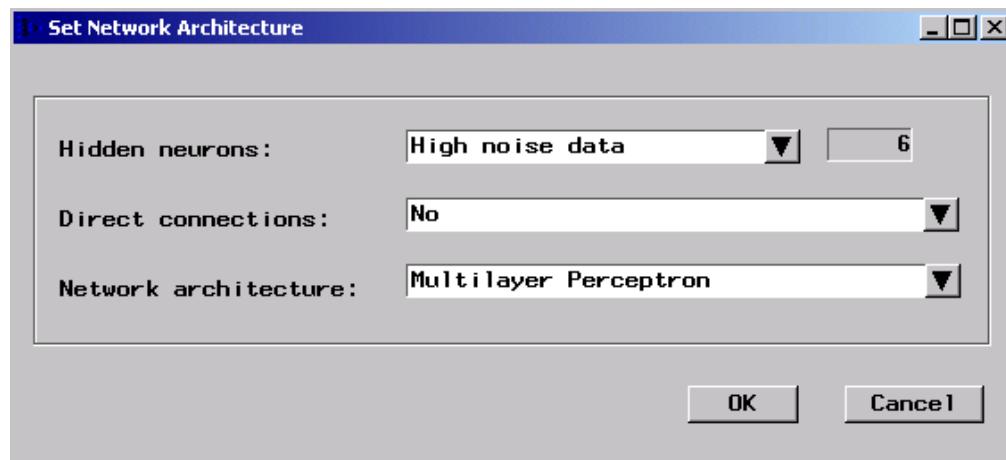
- |                        |   |
|------------------------|---|
| Average Error          | chooses the model that has the smallest average error for the validation data set.                          |
| Misclassification Rate | chooses the model that has the smallest misclassification rate for the validation data set.                 |
| Profit/Loss            | chooses the model that maximizes the profit or minimizes the loss for the cases in the validation data set. |

You can also specify options regarding the training history and the training monitor.

3. Because you have not created a profit/loss vector for this data, select **Average Error** as the model selection criterion.
4. Select the **Basic** tab. The Basic tab contains options for specifying network architecture, preliminary runs, training technique, and runtime limits.



5. Select the arrow next to Network architecture. The default network is a Multilayer Perceptron.



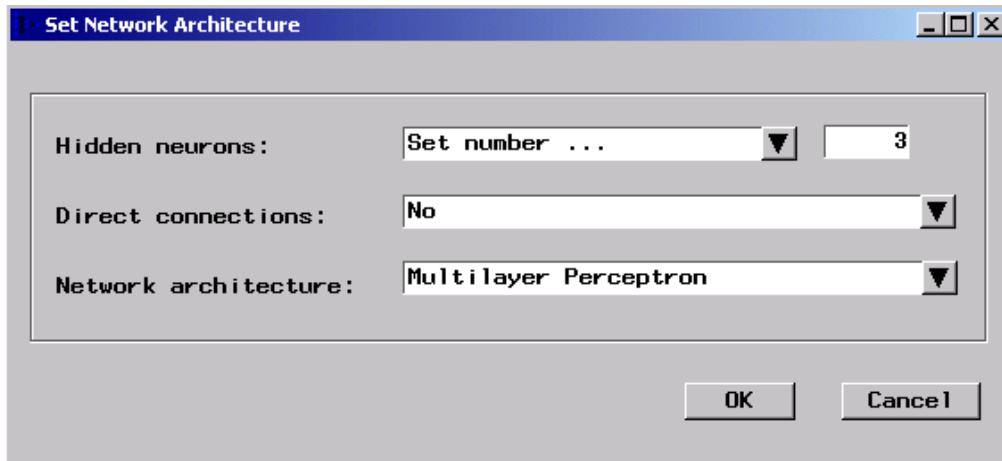
Hidden neurons perform the internal computations, providing the nonlinearity that makes neural networks so powerful. To set the number of hidden neurons criterion, select the Hidden neurons drop-down arrow and select one of the following items:

- High noise data
- Moderate noise data
- Low noise data
- Noiseless data
- Set number.

If you select the number of hidden neurons based on the noise in the data (any of the first four items), the number of neurons is determined at run time and is based on the total number of input levels, the total number of target levels, and the number of training data rows in addition to the noise level.

For this example, specify a multilayer perceptron with three hidden neurons.

1. Select the drop-down arrow next to Hidden neurons and select **Set Number...**.
2. Enter **3** in the field to the right of the drop-down arrow. Your dialog should now look like the one pictured below.



By default, the network does not include direct connections. In this case, each input unit is connected to each hidden unit and each hidden unit is connected to each output unit. If you set the Direct connections value to Yes, each input unit is also connected to each output unit. Direct connections define linear layers, whereas hidden neurons define nonlinear layers. Do not change the default setting for direct connections for this example.

The network architecture field allows you to specify a wide variety of neural networks including

- Generalized linear model
- Multilayer perceptron (default)
- Ordinary radial basis function with equal widths
- Ordinary radial basis function with unequal widths
- Normalized radial basis function with equal heights
- Normalized radial basis function with equal volumes
- Normalized radial basis function with equal widths
- Normalized radial basis function with equal widths and heights
- Normalized radial basis function with unequal widths and heights.

 Usage of the neural networks is discussed at length in the Neural Network Modeling course. Therefore, these architectures are not discussed here.

3. Select **OK** to return to the Basic tab.

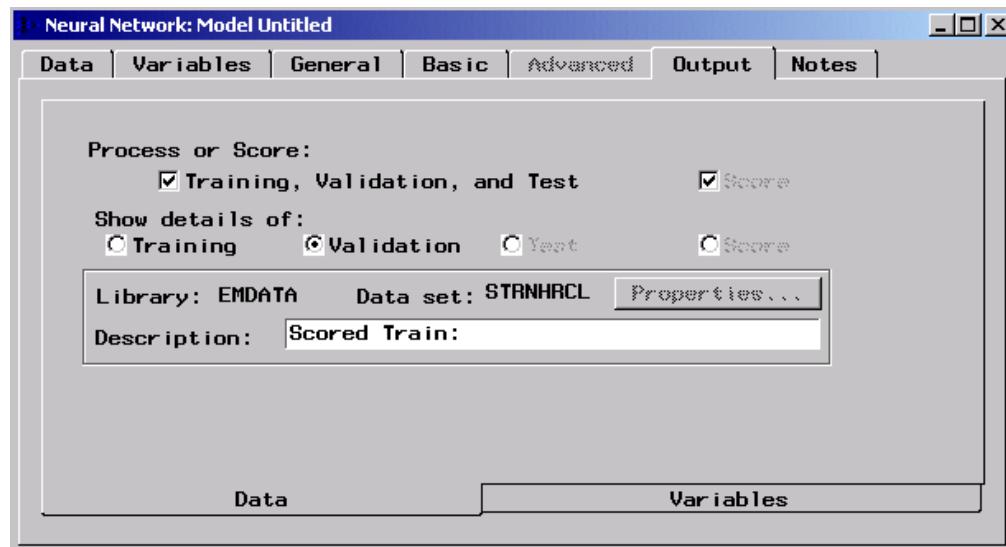
The remaining options on the Basic tab enable you to specify the following options:

- |                    |   |
|--------------------|---|
| Preliminary runs   | are preliminary runs that attempt to identify good starting values for training the neural network. |
| Training technique | is the methodology used to iterate from the starting values to a solution.                          |
| Runtime limit      | limits the time spent training the network.   |

Use the default options for this analysis.

4. Select the **Output** tab.

5. Select the **Training, Validation, and Test** checkbox.



6. Close the Neural Network node, saving changes when prompted.

7. Enter the name **NN3** in the model name field when prompted.
8. Select **OK**.

### Examine the Model

1. Run the flow from the Neural Network node and view the results when prompted.

The Tables tab is displayed first. Additional information about the estimates, the statistics, and the data sets is available from the drop-down menu.

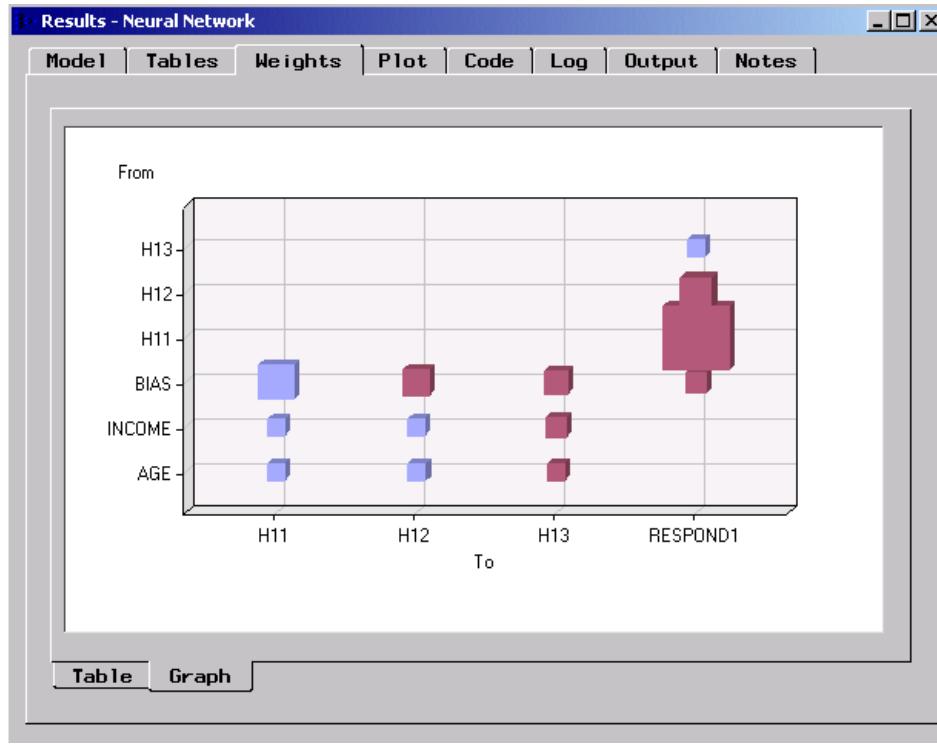
	Fit Statistic	Training	Validation
1	[TARGET=RESPOND]	.	.
2	Average Profit	0.079	0.0751666667
3	Misclassification Rate	0.079	0.0751666667
4	Average Error	0.2748578321	0.2642408627
5	Average Squared Error	0.0725131083	0.069151799E
6	Sum of Squared Errors	580.10486677	829.8215946E
7	Root Average Squared Error	0.2692825808	0.2629672975
8	Root Final Prediction Error	0.270159176	.
9	Root Mean Squared Error	0.2697212345	0.2629672975

2. Select the **Weights** tab. You may need to maximize or resize the window in order to see all of the weights. This table shows the coefficients used to construct each piece of the neural network model.

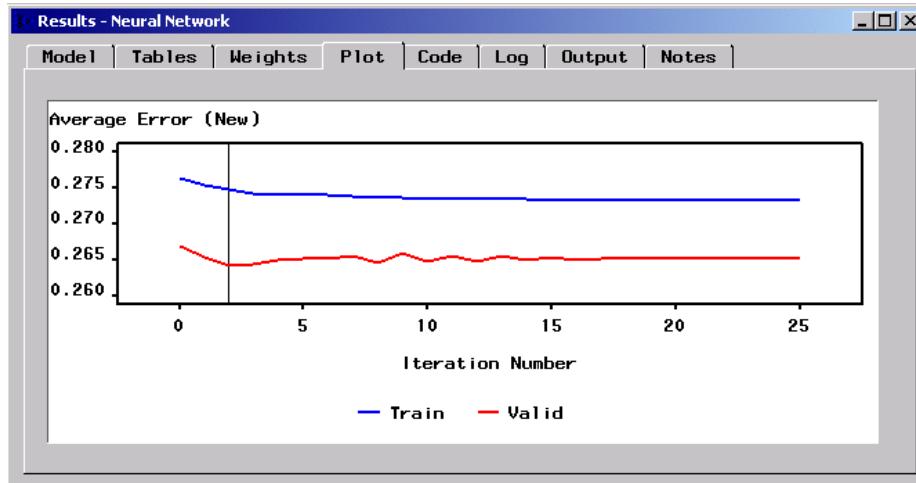
	From	To	Weight
1	AGE	H11	-0.095403849
2	INCOME	H11	-0.049448229
3	AGE	H12	-0.138118135
4	INCOME	H12	-0.047279827
5	AGE	H13	0.0598369607
6	INCOME	H13	0.3523536662
7	BIAS	H11	-1.529160668
8	BIAS	H12	0.7944280042
9	BIAS	H13	0.5747162064
10	H11	RESPOND1	3.9929180701
11	H12	RESPOND1	1.2359685746
12	H13	RESPOND1	-0.100015375
13	BIAS	RESPOND1	0.3929450195

**Table** **Graph**

3. Select the **Graph** subtab. The size of each square is proportional to the weight, and the color indicates sign. Red squares indicate positive weights, and blue squares indicate negative weights.



4. Select the **Plot** tab. This plots the error on the training and validation data sets. While additional iterations improve the fit on the training data set (top line) slightly, the performance on the validation does not continue to improve beyond the first few iterations. A line is drawn at the model that performs best on the validation data set.



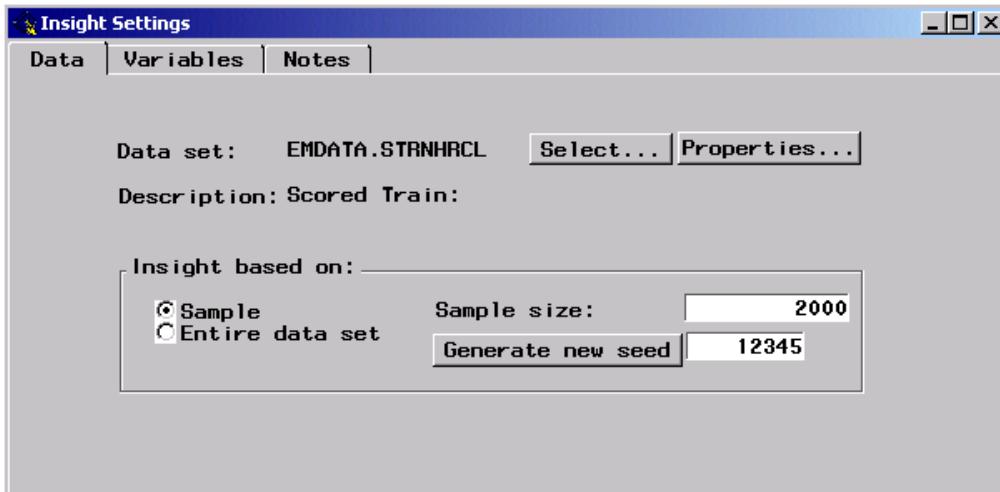
This plot is best viewed with the window maximized, although that was not done for the plot pictured above.

5. Close the results window.

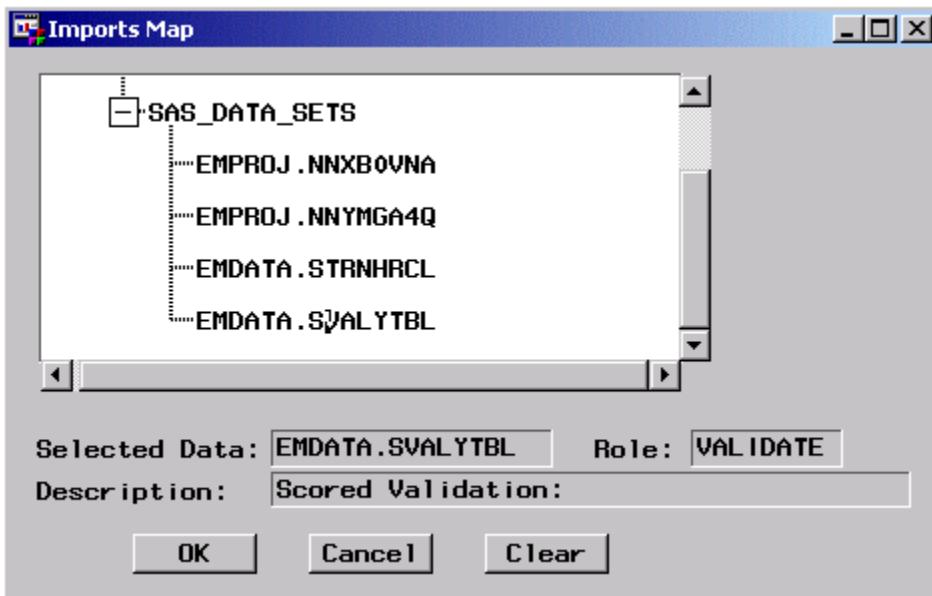
## Visualize the Model with Insight

You can use Insight to visualize the surface of this neural network

1. Open the Insight node.

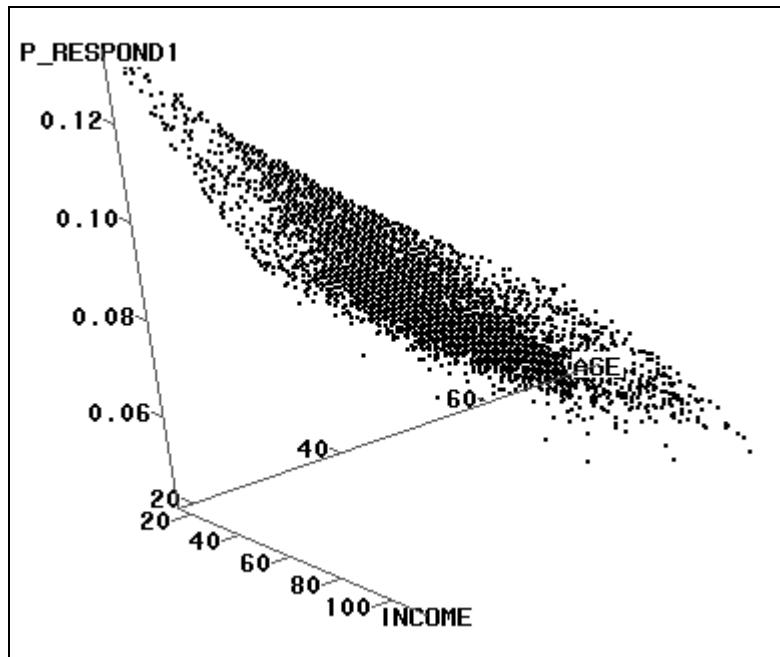


2. To use the validation data set, select Select....
3. Expand the list of predecessor data sets and select the validation data set.



4. Select OK to return to the Insight Settings window.
5. Select the Entire data set radio button.
6. Close the Insight Settings window, saving changes when prompted.
7. Run the flow from the Insight node and select Yes to view the results when prompted.
8. Select Analyze ⇒ Rotating Plot (Z Y X).

9. The values in the P\_RESPOND1 column are the predicted probabilities from the neural network model that RESPOND is equal to 1. Select **P RESPOND1**  $\Rightarrow$  **Y**.
10. Select **AGE**  $\Rightarrow$  **Z**.
11. Select **INCOME**  $\Rightarrow$  **X**.
12. To ensure that the axes in the graph meet at the minimum values, select **Output** and then select **At Minima**.
13. Select **OK** to return to the main rotating plot dialog.
14. Select **OK** to generate the plot.
15. Resize the display as desired.
16. Right-click in the plot and select **Marker Sizes**  $\Rightarrow$  **3**.

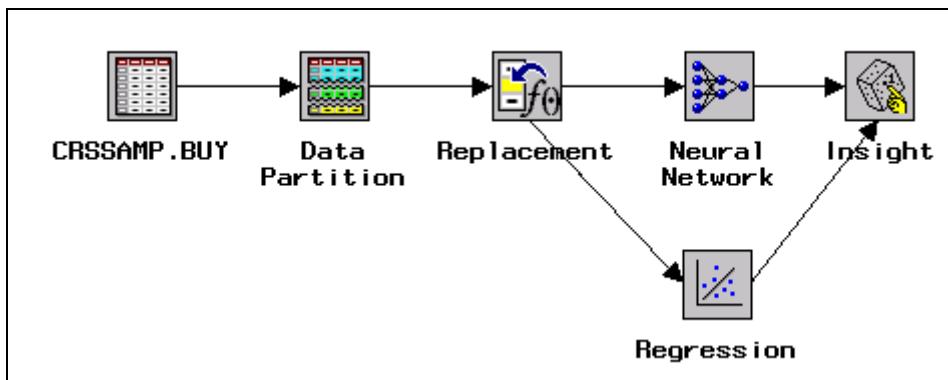


17. Close the rotating plot window and the data set window to exit from Insight and return to the Enterprise Miner workspace.

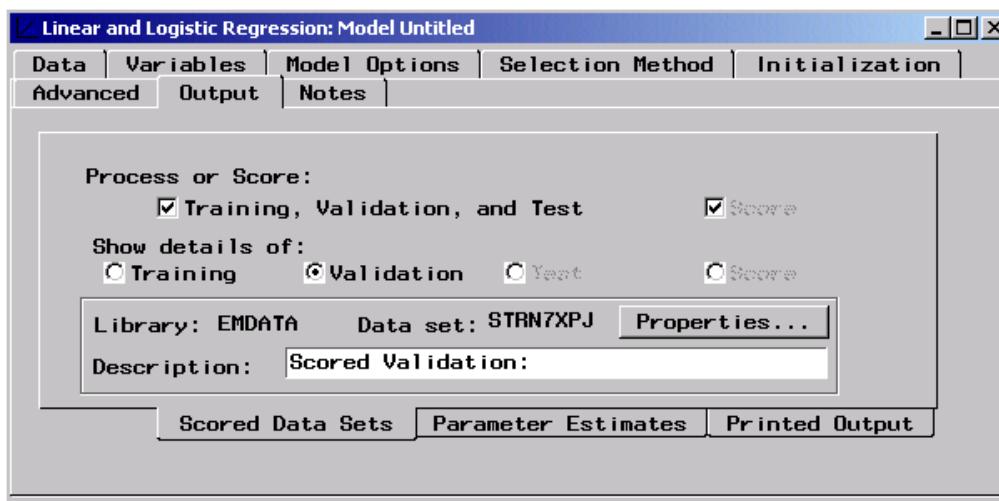
## Visualizing Logistic Regression

A standard logistic regression model is a MLP with zero hidden layers and a logistic output activation function.

- To visualize a fitted logistic regression surface, drag a Regression node onto the workspace and connect it as shown below.



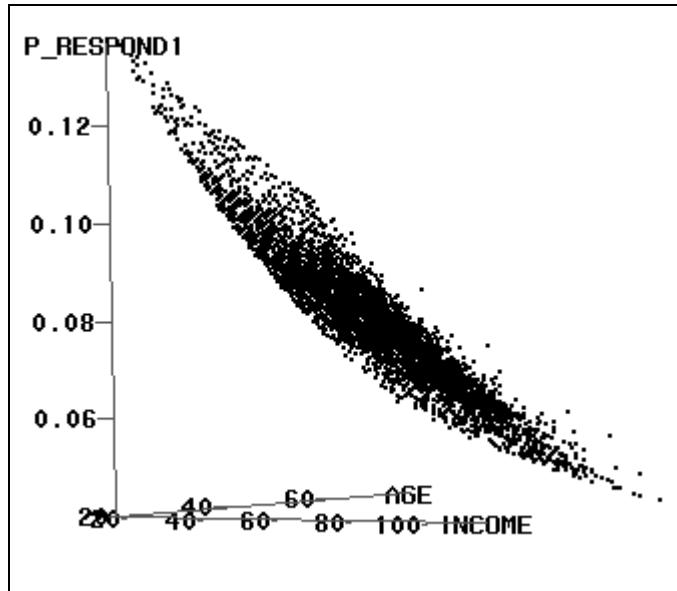
- To modify the regression node to add prediction information to the data sets, open the Regression node.
- Select the **Output** tab.
- Select the **Training, Validation, and Test** check box.



- Close and save changes to the Regression node. By default, the Regression model is named Untitled. You can edit this name.
- Run the diagram from the Regression node but do not view the results.
- Open the Insight node.
- Select the name of the scored validation data set for the regression model. You will open this data set from within Insight.
- Choose the option to use the entire data set if it is not already selected.

10. Close Insight, saving changes when prompted.
11. Run the flow from the Insight node.
12. Generate a rotating scatter plot as you did in the previous section.

 To see the plots for the regression and neural network models simultaneously, you must note the name of each data set. Select one of the data sets from within the Insight node, and open the other data set from within Insight.



# Chapter 6 Model Evaluation and Implementation

<b>6.1 Model Evaluation: Comparing Candidate Models .....</b>	<b>6-3</b>
<b>6.2 Ensemble Models .....</b>	<b>6-10</b>
<b>6.3 Model Implementation: Generating and Using Score Code .....</b>	<b>6-16</b>



## 6.1 Model Evaluation: Comparing Candidate Models

### Objectives

- Review methods of comparing candidate models.
- Generate and compare different models.

2

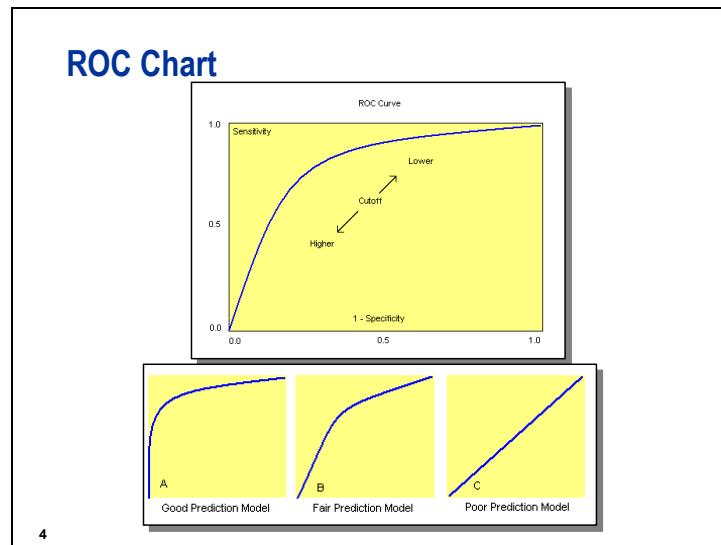
### Comparing Candidate Models

- Percent response chart
- Lift chart
- Profit chart
- ROC chart

3

As discussed earlier, the percent response, lift, and profit charts display and compare assessment statistics for groups of observations within the scored data set.

The receiver operating characteristic (ROC) chart is a graphical display that gives the measure of the predictive accuracy of a logistic model. It displays the sensitivity (a measure of accuracy for predicting events that is equal to the true positive / total actual positive) and specificity (a measure of accuracy for predicting nonevents that is equal to the true negative / total actual negative) of a classifier for a range of cutoffs. ROC charts require a binary target.



The receiver operating characteristic (ROC) curve is a graphical display that gives a measure of the predictive accuracy of a logistic regression model. The ROC curve displays the sensitivity and specificity of the model for a range of cutoffs. The cutoff choice represents a trade-off between sensitivity and specificity. A lower cutoff gives more false positives and fewer false negatives. A high cutoff gives more false negatives, a low sensitivity, and a high specificity.

The extremes (0,0) and (1,1) represent cutoffs of 1.0 and 0.0 respectively. If you use the rule that all observations with a predicted probability of 1 be classified as events, none of the event observations are correctly classified, but all of the nonevent observations are correctly classified (sensitivity = 0, specificity = 1). If you use a rule that all observations with a predicted probability of 0 or higher be classified as events, all the events are correctly classified by none of the nonevents are correctly classified (sensitivity = 1, specificity = 0). The horizontal axis is 1 – specificity, so the ROC curve starts at the point (0,0) and ends at the point (1,1).

The performance quality of a model is demonstrated by the degree the ROC curve pushes upward and to the left. This can be quantified by the area under the curve. The area will range from 0.50, for a poor model, to 1.00, for a perfect classifier. For a logistic regression model with high predictive accuracy, the ROC curve would rise quickly (sensitivity increases rapidly, specificity stays at 1). Therefore, the area under the curve is closer to 1 for a model with high predictive accuracy. Conversely, the ROC curve rises slowly and has a smaller area under the curve for logistic regression models with low predictive accuracy.

A ROC curve that rises at 45 degrees is a poor model. It represents a random allocation of cases to the classes and should be considered a baseline model.

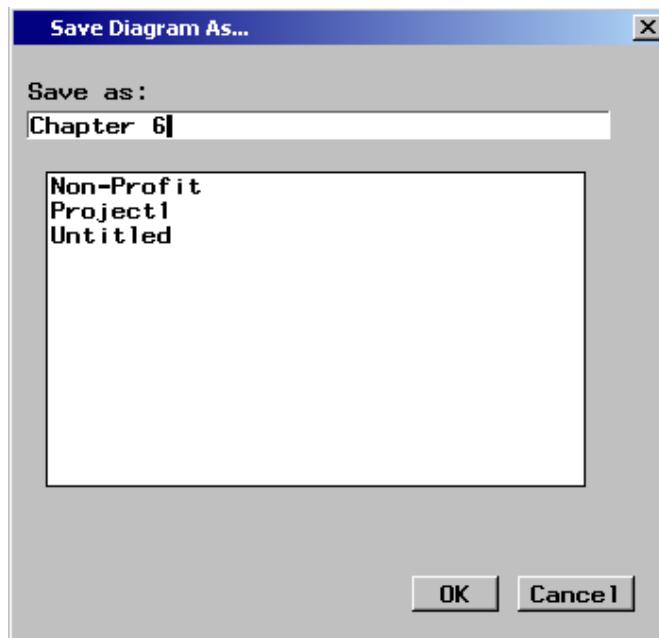


## Comparing Candidate Models

Recall that earlier in the course you generated a decision tree model and a regression model to predict who would donate to a nonprofit organization. Consider generating an alternative decision tree and a neural network for the same target and comparing all four of the models.

There are times when you do not want to change a diagram but do want to consider some additional analyses for the same data. In such situations, it is useful to create a copy of your Enterprise Miner diagram.

1. Open the Non-Profit diagram created earlier.
2. To create a copy of the diagram, select **File**  $\Rightarrow$  **Save diagram as...**
3. In the Save Diagram As... dialog window, type in a new name, such as **Chapter 6**.

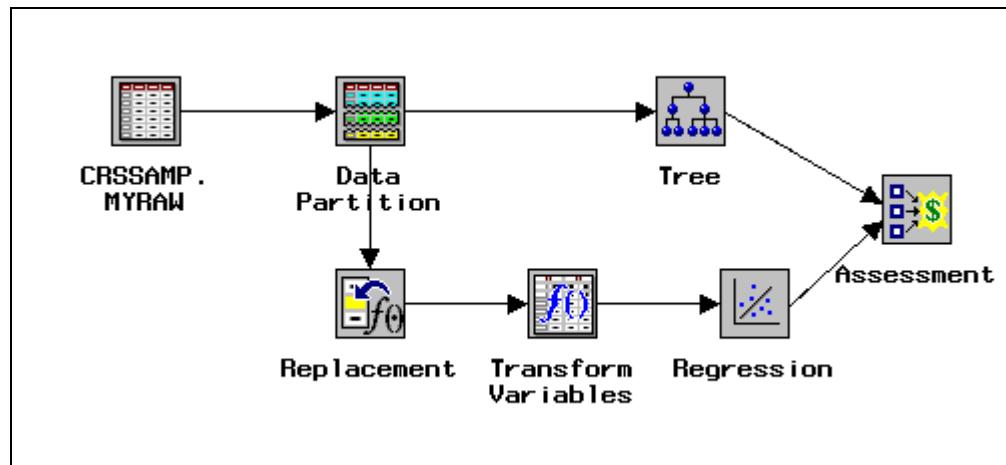


4. Select **OK**. A copy of the original diagram is made with the new name.



Be patient because it may take a while to copy the diagram, especially if there are a large number of nodes. You can observe the progress by watching the notes that appear in the lower-left corner of the SAS window.

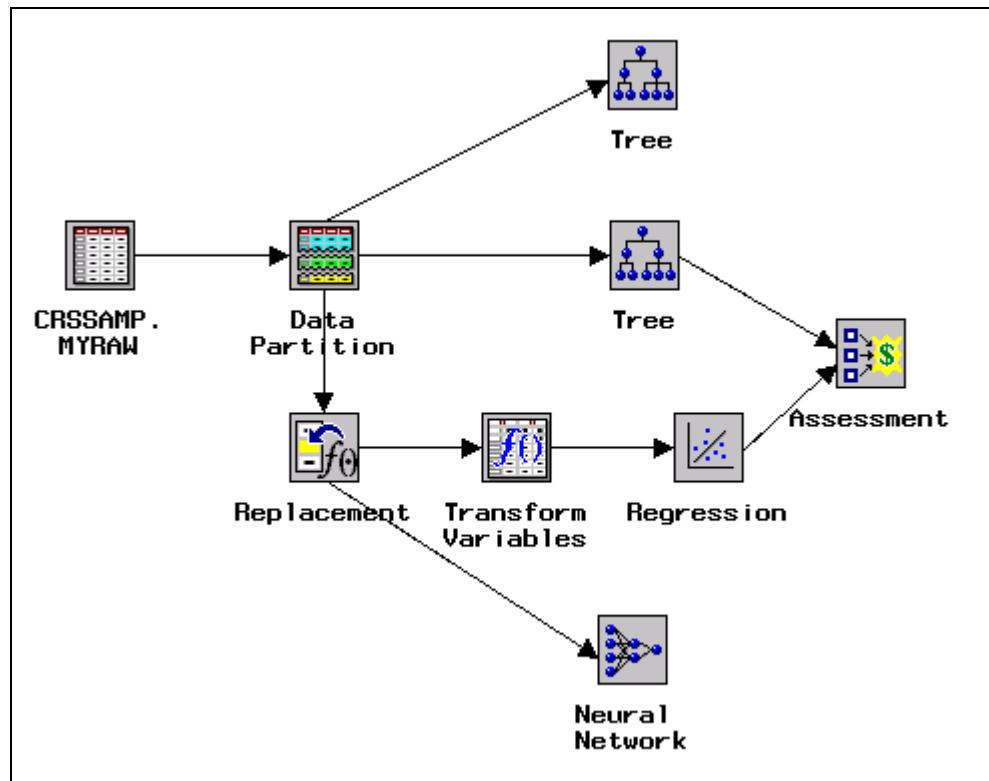
5. To unclutter the diagram workspace, delete the two Insight nodes, the Variable Selection node, and the Tree node used earlier for variable selection.



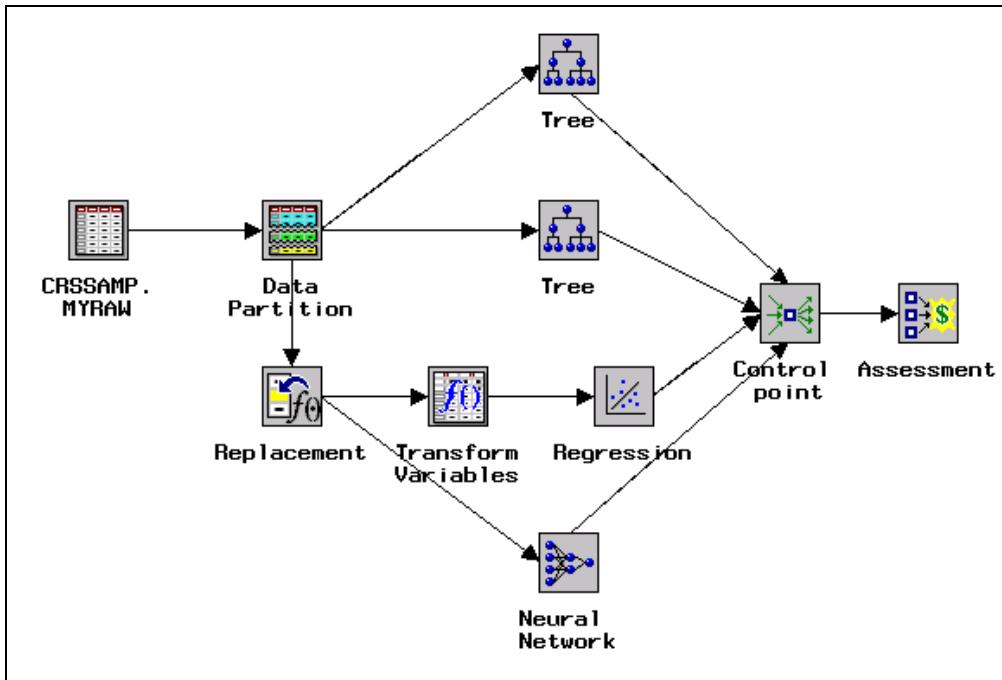
6. Add a Neural Network node and connect it to the Replacement node.

Because of the transformations that occur in a neural network model, the model is inherently flexible. As mentioned earlier, a multilayer perceptron with one hidden layer is a universal approximator. Therefore, when generating a neural network model, there is not the same necessity as with a linear regression model to transform variables to ensure a better model fit.

7. Add a new Tree node and connect it to the Data Partition node.



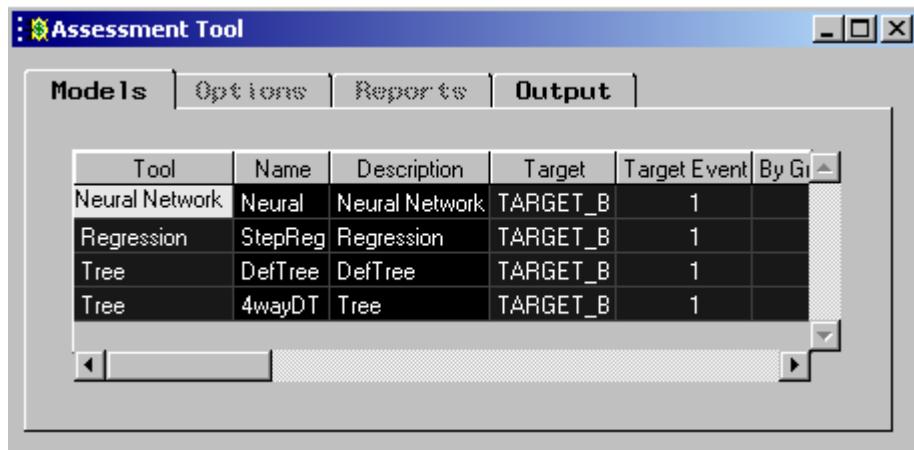
8. Rather than connecting all of the modeling nodes to the Assessment node, connect them through a Control Point. To do this, first delete the connecting arrows to the Assessment node. Select the arrow, right-click on the arrow, and select **Delete**. Repeat this process for each of the two arrows.
9. Add a Control Point to the diagram. Connect the four modeling nodes to the control point and connect the control point to the Assessment node.



The original decision tree created for this data was done with the Enterprise Miner default settings. This allows for only two-way splits. For the new tree model, allow up to four-way splits.

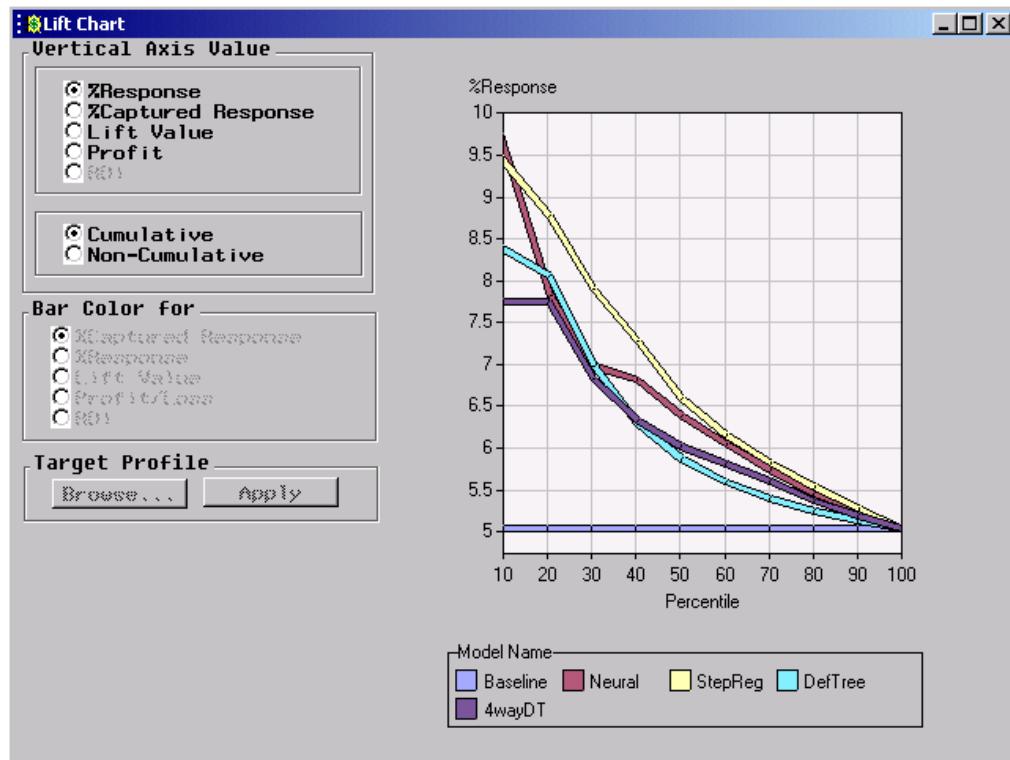
10. Right-click on the new Tree node and select **Open...**.
11. Select the **Basic** tab.
12. To allow for up to four-way splits, change the maximum number of branches from a node to **4**.
13. Close the Tree node, saving changes when prompted.
14. When prompted, name the model **4wayDT** and select **OK**.
15. Change the name of the node in the diagram to **4-way Tree**.
16. Because our primary purpose here is to compare models, leave the neural network node set with its defaults and run the diagram from the Assessment node.
17. Select **Yes** to view the results when prompted.
18. Change the name of the Neural Network to **Neural**.

19. Click and drag to select all four models.



20. Select Tools  $\Rightarrow$  Lift Chart.

21. Select Format  $\Rightarrow$  Model Name.



Based on the % Response cumulative chart, the stepwise regression model is the overall best model in this case, with 8.8% response in the first two deciles as compared with the 7.7 to 8 % response in the first two deciles for the other models and 5% in the population as a whole. In the first decile, the neural network is slightly better than the regression model.



A regression model provides a better fit than other, more flexible, modeling methods when the relationship between the target and the inputs is linear in nature.

22. Close the Lift Chart and Assessment Tool windows to return to the workspace.

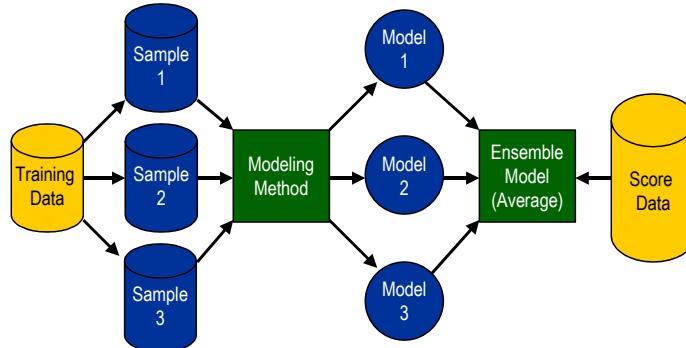
## 6.2 Ensemble Models

### Objectives

- List the different types of ensemble models available in Enterprise Miner.
- Discuss different approaches to combined models.
- Generate and evaluate a combined model.

7

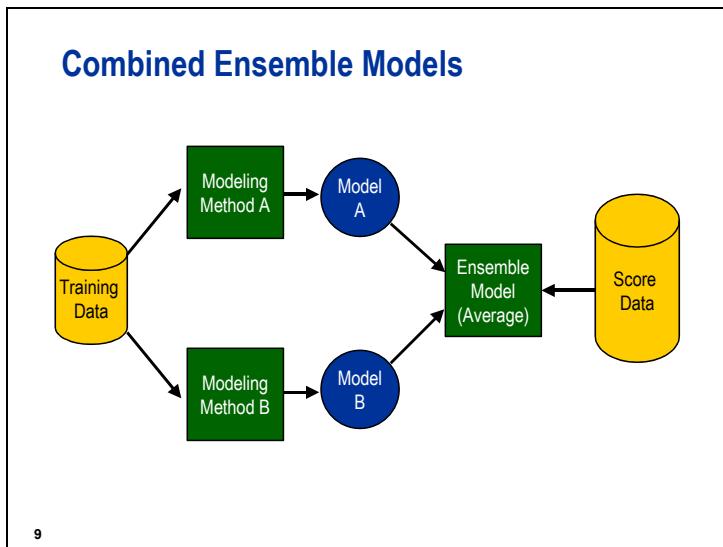
### Combined Ensemble Models



8

The Ensemble node creates a new model by averaging the posterior probabilities (for class targets) or the predicted values (for interval targets) from multiple models. The new model is then used to score new data.

One common ensemble approach is to resample the training data and fit a separate model for each sample. The Ensemble node then integrates the component models to form a potentially stronger solution.



Another common approach is to use multiple modeling methods, such as a neural network and a decision tree, to obtain separate models from the same training data set. The Ensemble node integrates the component models from the two complementary modeling methods to form the final model solution.

It is important to note that the ensemble model created from either approach can only be more accurate than the individual models if the individual models disagree with one another. You should always compare the model performance of the ensemble model with the individual models.

## Other Types of Ensemble Models

- Stratified
- Bagging
- Boosting

10

The Ensemble node can also be used to combine the scoring code from stratified models. The modeling nodes generate different scoring formulas when operating on a stratification variable (for example, a group variable such as GENDER) that you define in a Group Processing node. The Ensemble node combines the scoring code into a single DATA step by logically dividing the data into IF-THEN-DO/END blocks.

Bagging and boosting models are created by resampling the training data and fitting a separate model for each sample. The predicted values (for interval targets) or the posterior probabilities (for a class target) are then averaged to form the ensemble model.



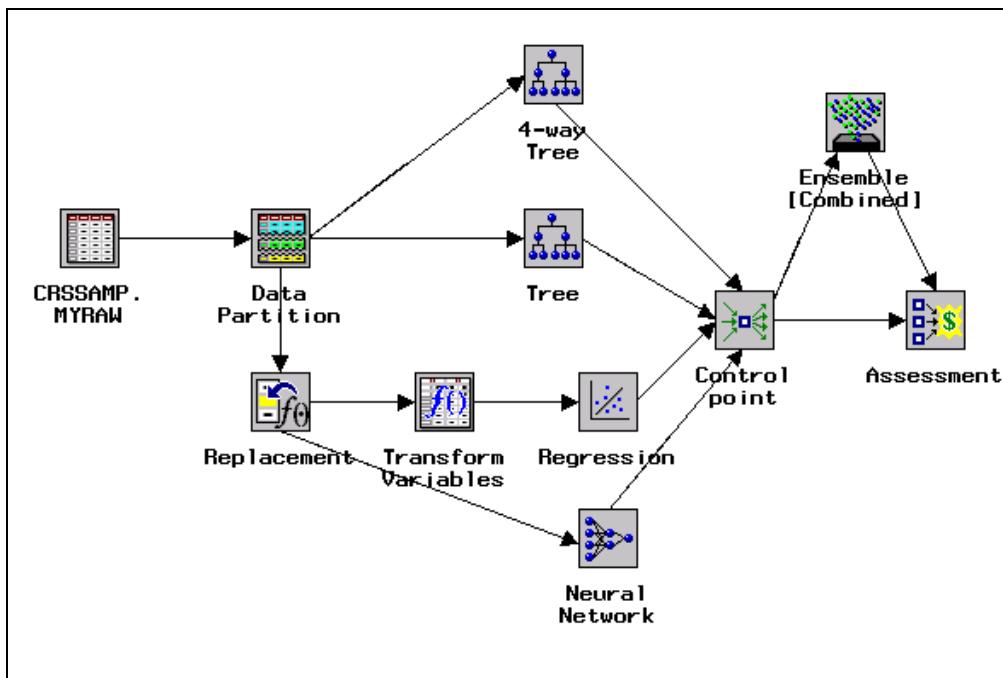
Bagging and boosting models are discussed in detail in the Decision Tree Modeling course.



## Combined Ensemble Models

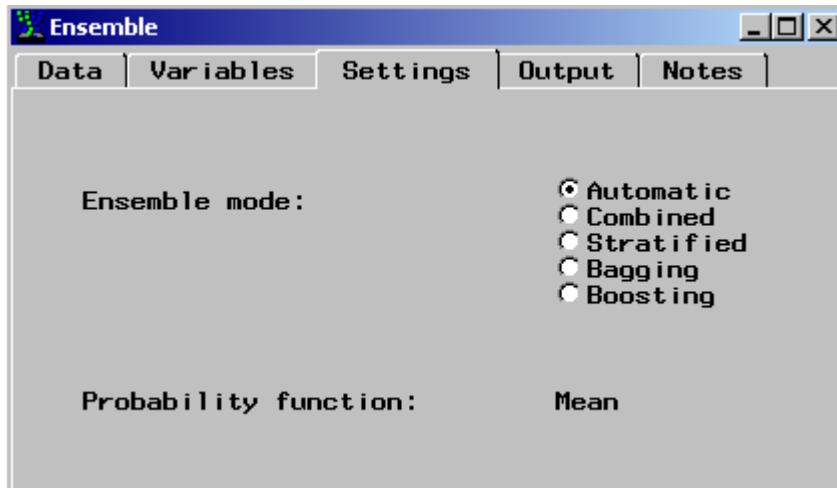
You have fit four separate models to the nonprofit organization data: two decision trees, a regression, and a neural network. In comparing these models, you determined that the regression model appears to be the best model on the validation data set. Combine these models to form an ensemble model and determine if this is a better model than each of the individual models generated.

1. Add an Ensemble node to the diagram.
2. Make a connection from the Control Point to the Ensemble node and from the Ensemble node to the Assessment node.



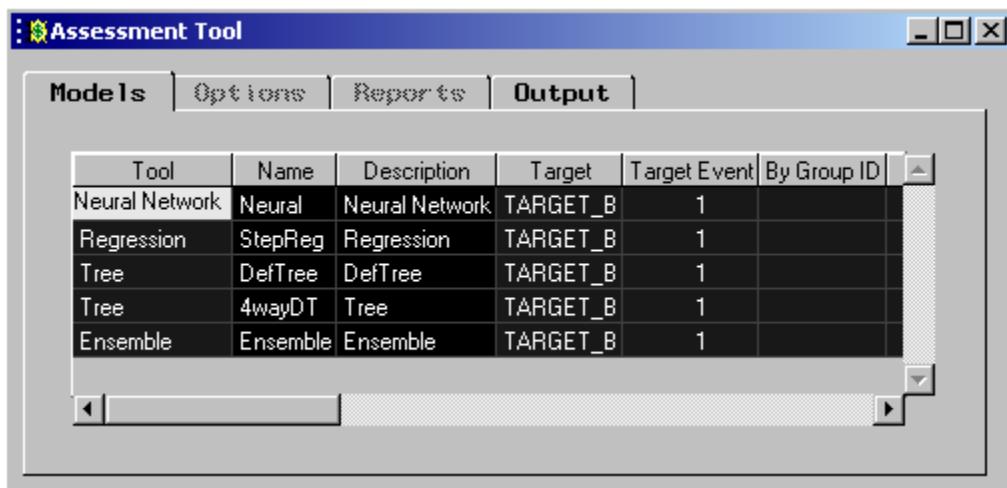
3. To open the Ensemble node, right-click and select [Open...](#).

4. The Variables tab is active and shows the active target variable, in this case TARGET\_B. Select the **Settings** tab.

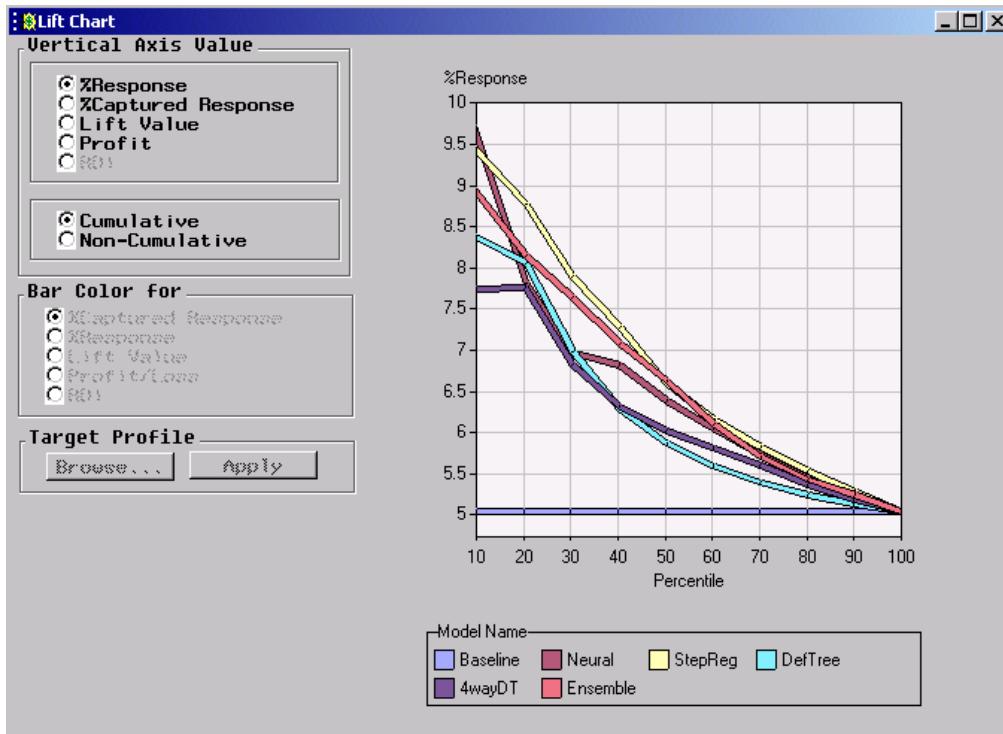


The default selection is Automatic. In this case, because there is no Group Processing node in the diagram, the Ensemble node will create a combined model. If there is a Group Processing node in the diagram, the type of model built under the automatic setting depends upon the setting in the Group Processing node.

5. Close the Ensemble node and run the diagram from the Assessment node.
6. Select **Yes** to view the results when prompted.
7. Change the name of the Ensemble model to **Ensemble**.
8. Click and drag to select all five models.



9. Select **Tools**  $\Rightarrow$  **Lift Chart**.
10. Select **Format**  $\Rightarrow$  **Model Name**.



The neural network model appears to do a slightly better job with the first decile, but the regression model is better than or the same as the neural network model after the first decile. In this case, choosing a model may depend upon your business use. If you know that you are only planning on mailing to the top 10% for this card promotion, then you might choose to use the neural network model. If you were planning on sending the mailing to a greater percentage of the potential donors, you might choose to do so based on scores produced by the regression model. Note that in this case the ensemble model is not an improvement over the best of the individual models.

11. Close the Lift Chart and Assessment Tool windows to return to the workspace.

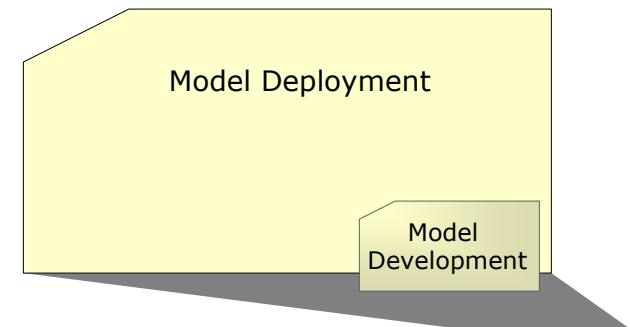
## 6.3 Model Implementation: Generating and Using Score Code

### Objectives

- Discuss the tasks involved in scoring data.
- Be aware of some pitfalls in developing and deploying models.
- Generate and use score code within Enterprise Miner.
- Use score code outside of Enterprise Miner.

13

### Scoring



14

The predictive modeling task is not completed once a model and allocation rule is determined. The model must be practically applied to new cases. This process is called *scoring*.

In database marketing, this process can be tremendously burdensome, because the data to be scored may be many times more massive than the data that was used to develop the model. Moreover, the data may be stored in a different format on a different system using different software.

In other applications, such as fraud detection, the model may need to be integrated into an online monitoring system.

## Scoring Recipe

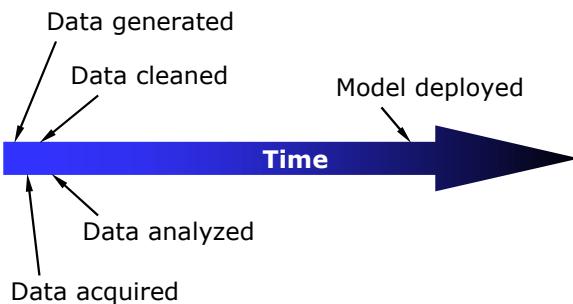
- Model
  - Formula
- Data Modifications
  - Derived inputs
  - Transformations
  - Missing value imputation

- Scoring Code
  - ≠ Scored data
  - ≠ Original computation algorithm

15

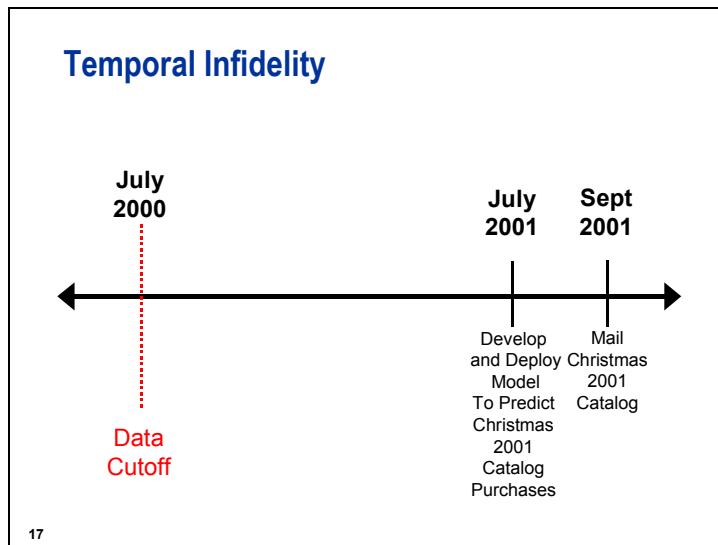
The score code must be more than just the model equation. It must incorporate all data manipulation tasks done before generating the model. In the Enterprise Miner score code, this is done automatically.

## Population Drift



16

There is almost always a lag between model development and deployment. However, the population is dynamic. The data used to build a model might not adequately reflect the population at future times. Predictive models should be monitored, revalidated, and periodically refitted.



*Temporal infidelity* (John 1997) occurs when the input variables contain information that will be unavailable at the time that the prediction model is deployed. For example, in July 2001 you are developing a model to deploy for predicting purchases from your Christmas catalog. You will build the model with data from Christmas of 2000 and then score data to predict Christmas of 2001. In building the model, you must cut your input data off at July 2000 because when you score your new data you will only have information up to July 2001.

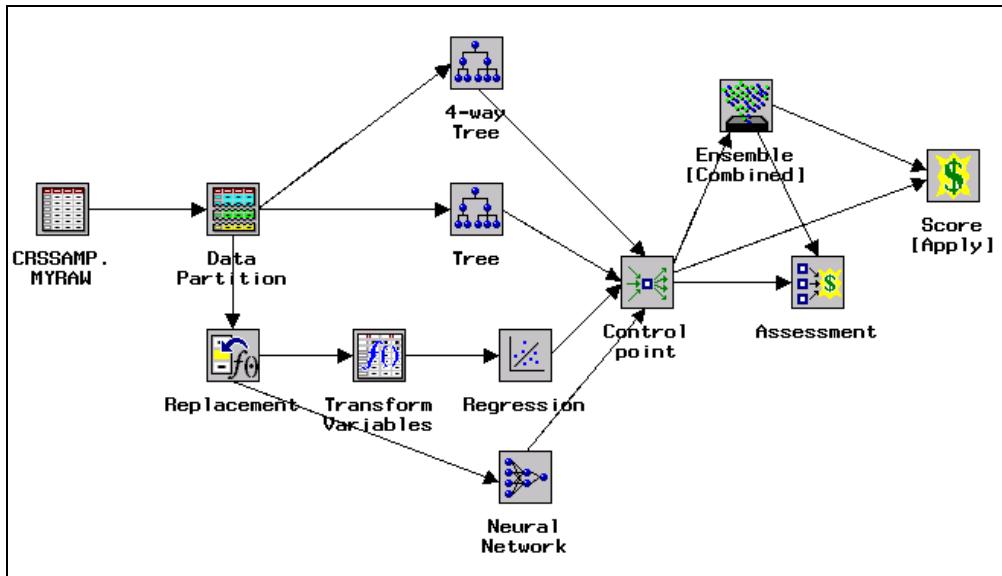
Another example of temporal infidelity is using intraoperative and postoperative information to predict surgical outcomes, when the purpose of the model is to predict patient outcomes preoperatively.



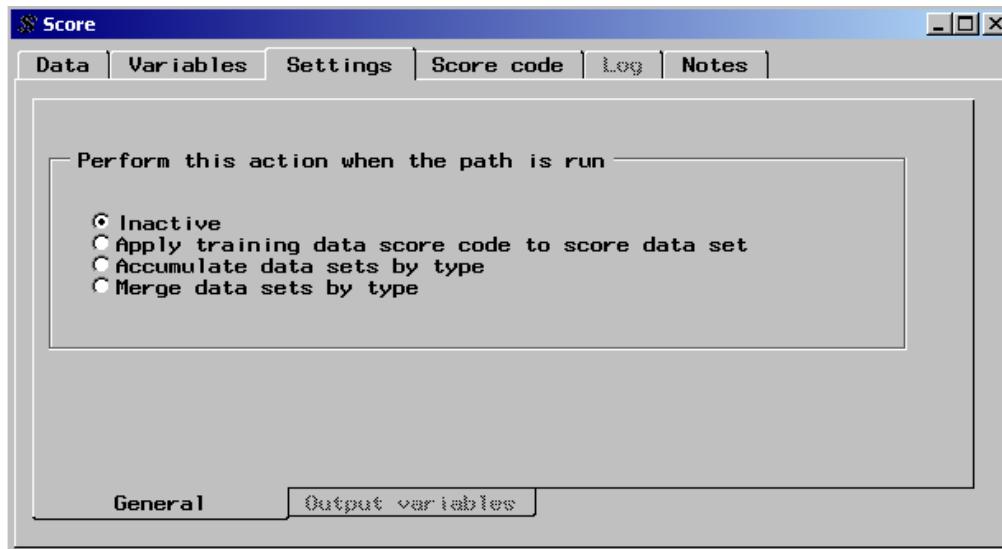
## Generating and Using Score Code

The Score node can be used to evaluate, save, and combine scoring code from different models.

1. Add a Score node to the diagram and connect it to the control point and the Ensemble node as shown.

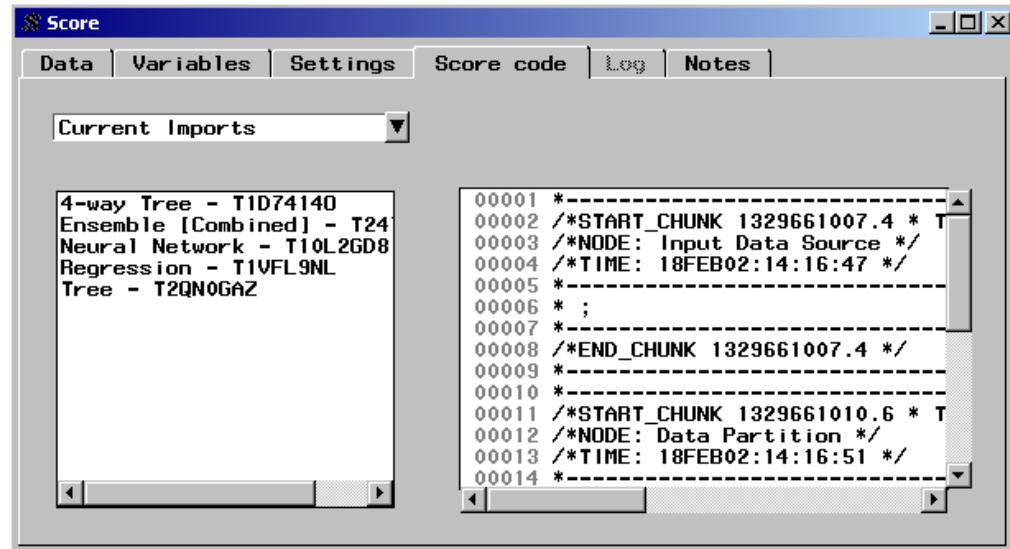


2. Open the Score node. The Settings tab is active.



The Settings tab provides the following options when you run the Score node in a path:

- Inactive (default) - exports the most recently created scored data sets.
  - Apply training data score code to score data set - applies scoring code to the score data set.
  - Accumulate data sets by type - copies and exports data sets imported from predecessor nodes. If you use this action in a path that contains a Group Processing node, the output data sets are concatenated.
  - Merge data sets by type - merges data sets imported from predecessor nodes. For example, you can use this action to merge the training data sets from two modeling nodes to compare the predicted values. If the number of observations in each score data set is not the same, an error condition is created.
3. Select the **Score code** tab. This tab shows the scoring code for each model connected to the Score node.

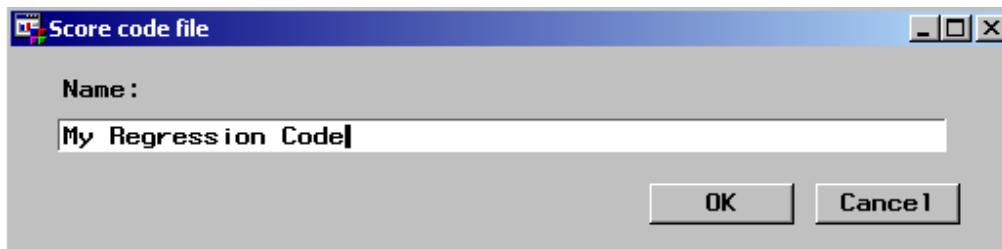


4. Click on the arrow next to Current Imports to see the available management functions. The options include
  - Current imports - (default) lists the scoring code currently imported from node predecessors.
  - Accumulated runs - lists scoring code that was exported by the node's predecessors during the most recent path run (training action). If the training action involves group processing, a separate score entry is listed for each group iteration for each predecessor node. This is the only access to score code generated from group processing.
  - Saved - lists saved or merged score code entries.
  - All - lists all score code entries that are managed by the node.

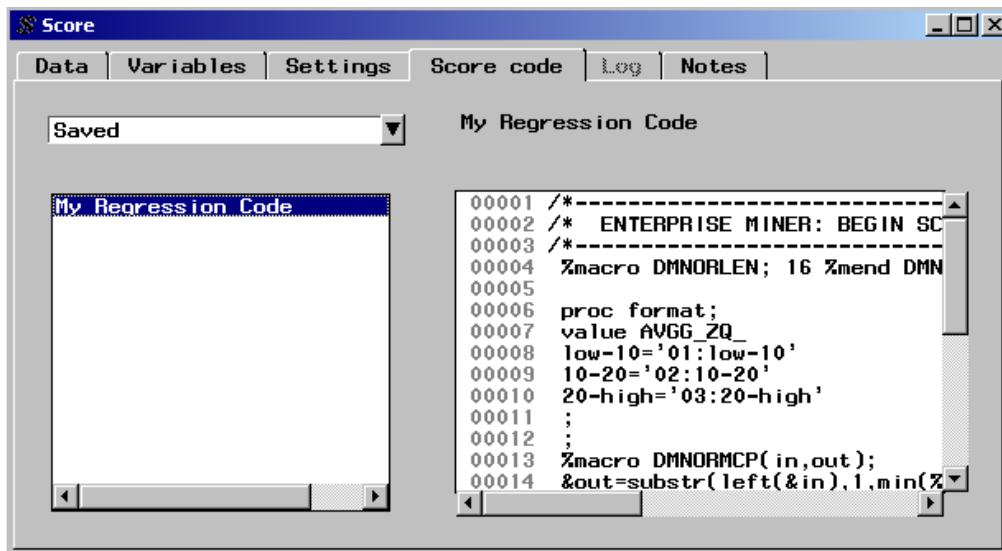
To see the scoring code for a model, double-click on the desired model in the list on the left, and the associated scoring code is displayed in the window on the right. The code is a SAS program that performs a SAS DATA step. You can use the scoring code on any system having base SAS.

If you modify the settings in a modeling node and run the flow, the scoring code associated with the affected model is updated. To keep modifications in the workspace from affecting the scoring code, you can save the scoring code. For this example, presume that you want to retain the code from the regression model.

1. Select the regression model from the list on the left side of the window.
2. Right-click on the selected model and select Save....
3. A dialog window opens that enables you to name the saved source file. You can enter a name if desired, although this is not necessary. Type in a name, such as **My Regression Code**.



4. Select OK. Enterprise Miner now displays the saved runs.



The code is now saved within Enterprise Miner. To use the code outside of Enterprise Miner in a SAS session, you need to export the scoring code from Enterprise Miner. You can export the scoring code as follows:

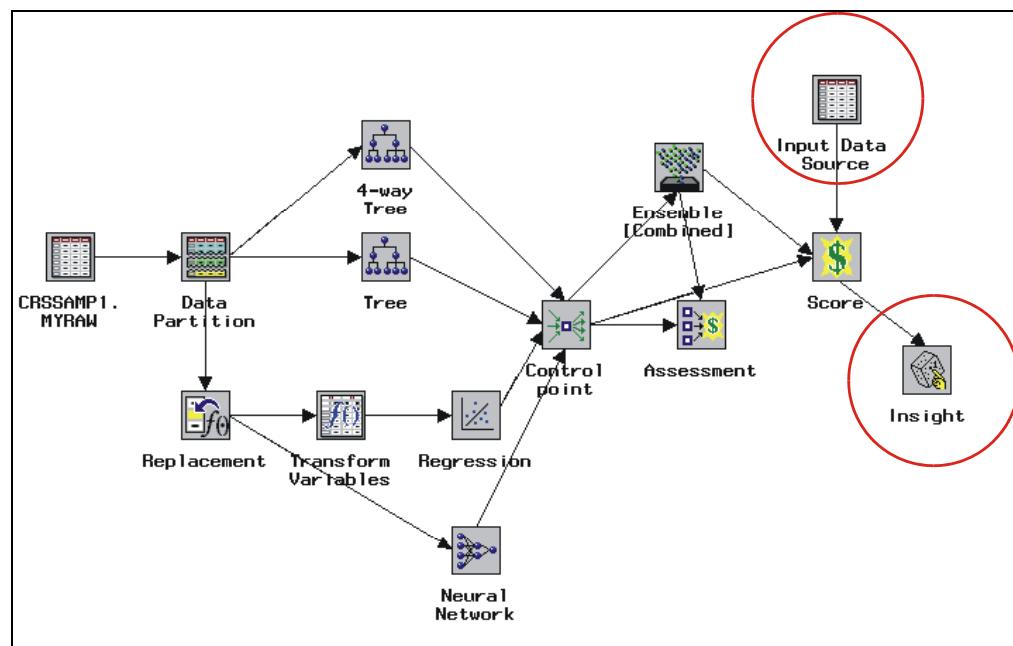
1. Highlight the name representing the desired code in the list on the left side.
2. Right-click on the highlighted name and select **Export...**
3. Enter a name for the saved program, such as **MyCode**, and select **Save**.
4. Close the Score node.

 You cannot export score code until it has first been saved.

### Scoring within Enterprise Miner

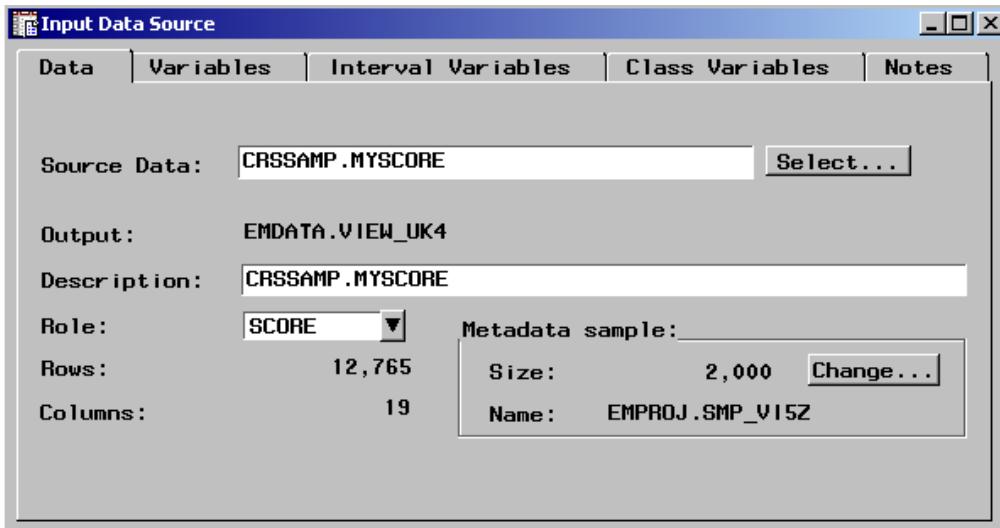
The saved scoring code can be used in base SAS to score a data set, but you can score a new data set within an Enterprise Miner diagram.

1. Add another Input Data Source node to the flow and connect it to the Score node.
2. Add an Insight node and connect the Score node to it as pictured below.



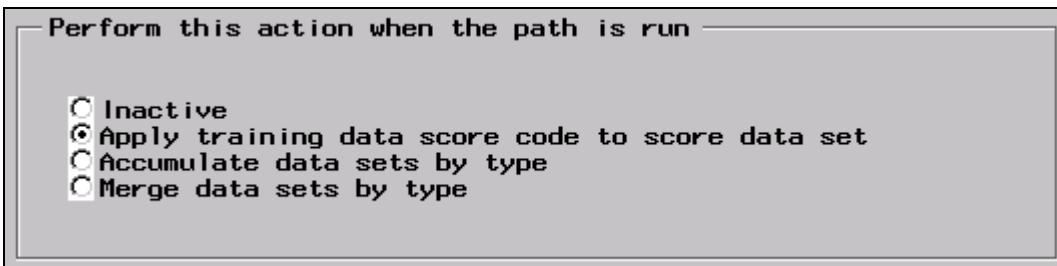
3. Open the Input Data Source node and select the **MYSCORE** data set from CRSSAMP library.

4. Change the role of the data set from RAW to SCORE.

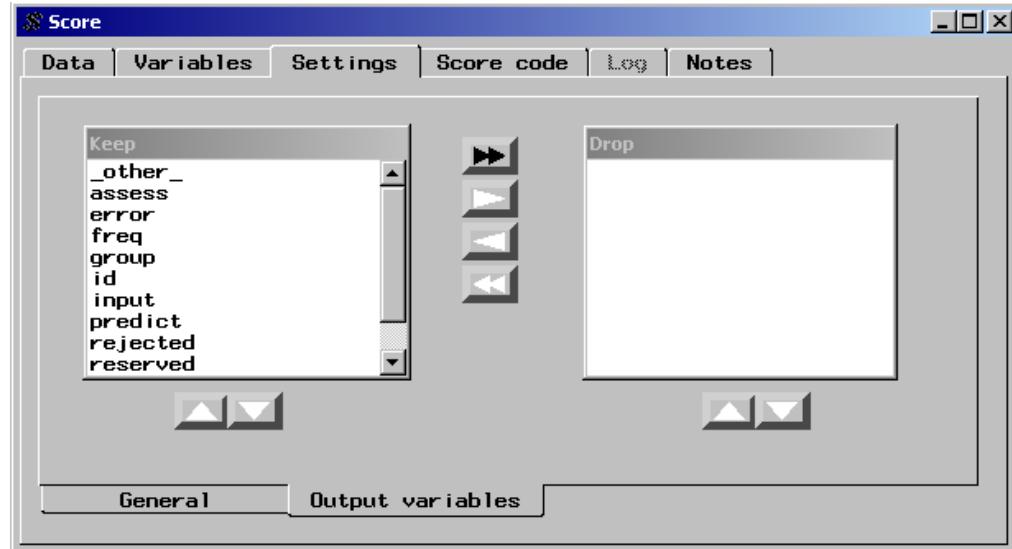


Inspect the variables if you desire. There is no need to modify the variables here because the role and level of each variable is built into the scoring code.

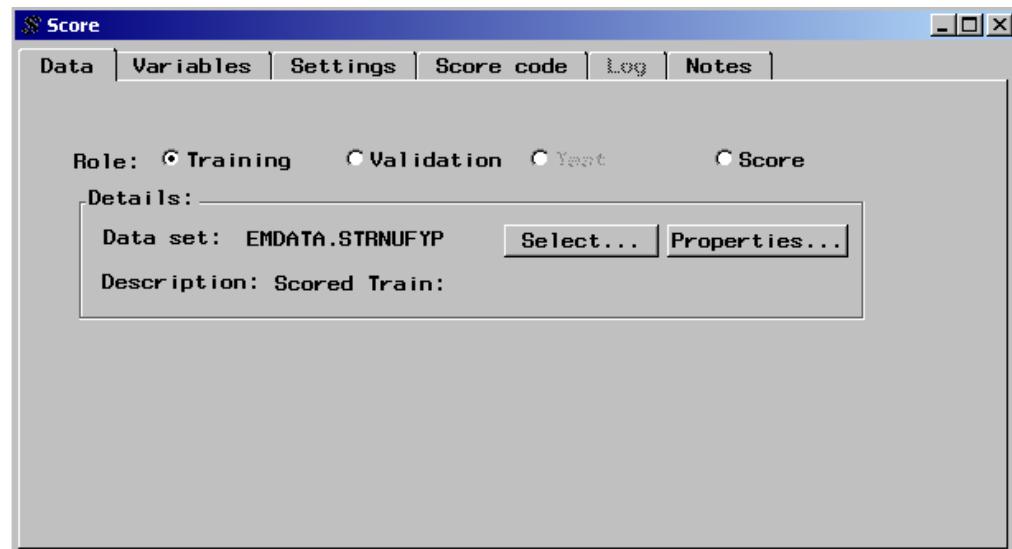
5. After inspection, close this Input Data Source, saving changes when prompted.
6. Open the Score node. By default the Score node is inactive when running items in a path. Select the Apply training data score code to score data set radio button. The Score node will now add prediction information to the data set that it is scoring.



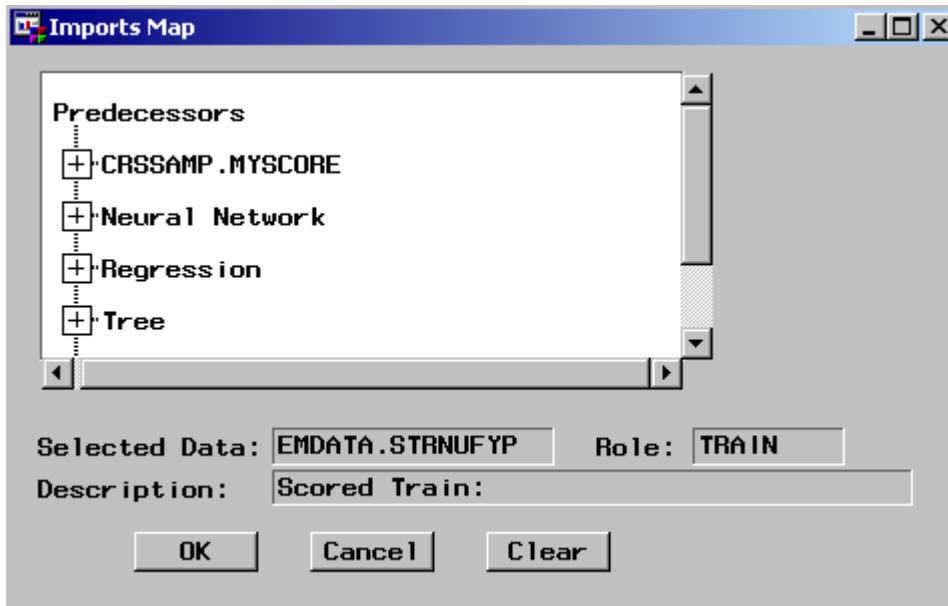
7. After changing the action of the Score node, the Output variables subtab becomes available. Select the **Output variables** subtab. This subtab enables you to control what values are added to the scored data set. All variables are included by default, but the options shown below allow you to drop certain variables, if desired. No variables are dropped in this example.



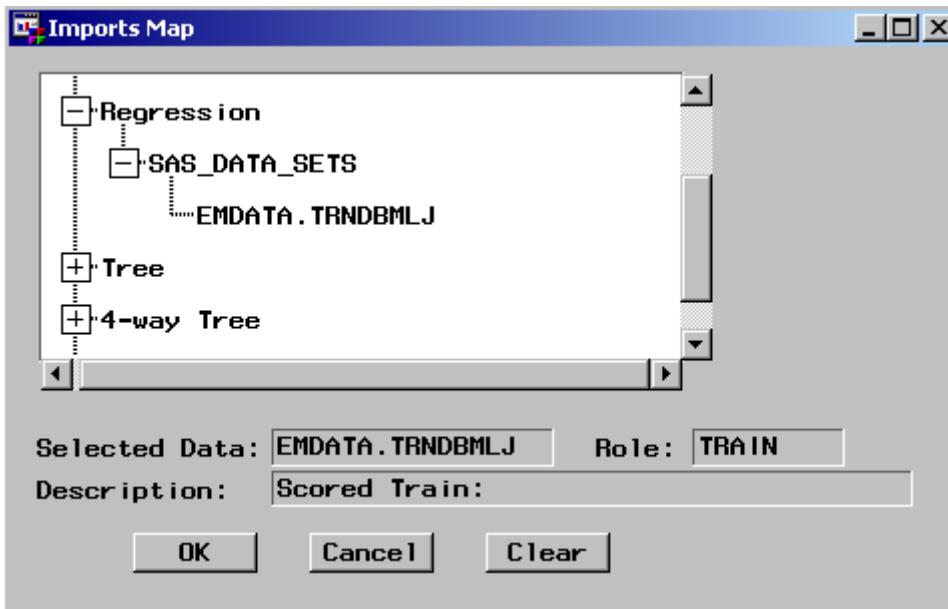
8. Recall that you fit several different models. You can control which of the models will be used for scoring by specifying the desired model in the Data tab. Select the **Data** tab.



9. Select the **Select...** button to see the list of predecessors. An Imports Map window opens, displaying the predecessor nodes.

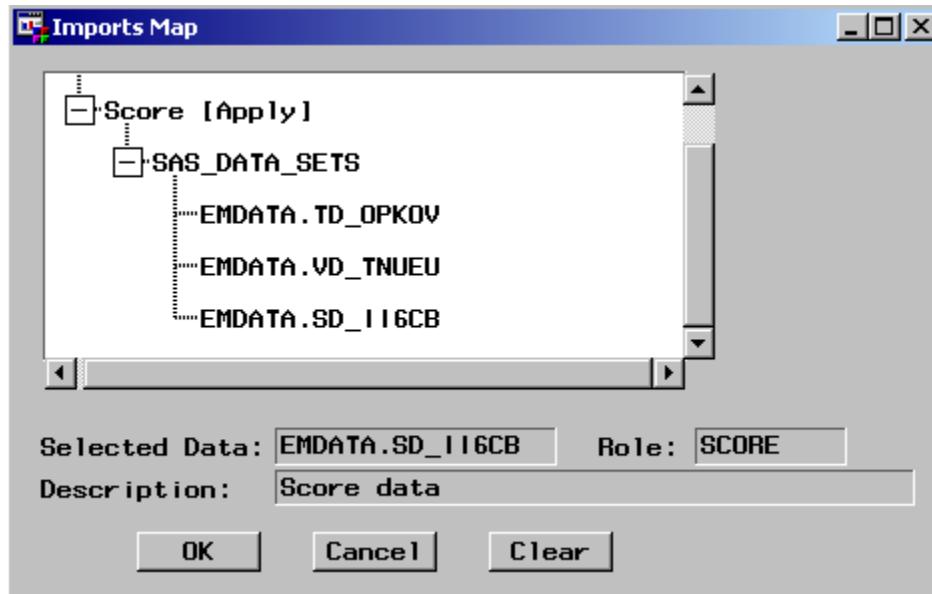


10. For this example, presume that you have determined that the regression model is the best model for your business purposes. Find the data set associated with the regression node and select it as shown below.



11. Select **OK** to accept this selection. The regression code will now be used for scoring the new data set.  
 12. Close the Score node, saving changes when prompted.  
 13. Run the diagram from the Score node. You do not need to view the results.  
 14. Open the Insight node.

15. Select the **Select...** button on the Data tab to select the data set associated with the score data. This data set will typically have an SD prefix followed by a string of random alphanumeric characters. Also note that the description will be **Score data**.



16. Select **OK** to return to the Data tab.
17. Select **Entire data set**.
18. Close the Insight Settings window, saving changes when prompted.
19. Run the diagram from the Insight node and select **Yes** to view the results when prompted.

The scored data set now has 50 variables. Only 19 variables were in the original data set, so the scoring code has added additional variables to this data set.

IDCODE	AGE	HOMEOWNR	Int		Nom		Int		Nom		Int		Int		Nom		Nc	
			AGE	HOMEOWNR	INCOME	GENDER	MALEMIL	MALEVET	PETS	PCOWNER								
12765	1	62.0000	H		3	F		2	25	U								
	2	66.0000	H		5	F		0	33	U								
	3	58.0000	U		3	F		0	38	U								
	4	61.8062	U		5	F		1	38	U								
	5	61.8062	U		5	F		0	35	U								
	6	48.0000	H		7	M		0	43	Y								
	7	70.0000	U		5	F		0	33	U								
	8	61.0000	H		1	F		0	24	Y								
	9	75.0000	U		2	M		0	24	U								
	10	60.0000	H		4	M		5	35	U								
	11	74.0000	H		1	F		0	34	U								
	12	68.0000	H		4	F		0	55	U								
	13	72.0000	U		1	M		0	20	U								
	14	76.0000	U		2	F		13	46	U								
	15	38.0000	U		7	F		0	32	U								
	16	74.0000	H		3	F		0	27	U								
	17	47.0000	H		4	F		0	17	U								
	18	61.8062	U		1	F		0	24	U								
	19	86.0000	H		6	F		2	32	U								
	20	41.0000	H		5	M		2	20	U								
	21	78.0000	H		6	F		0	21	U								
	22	61.8062	U		5	M		0	30	U								

You can see some of the newly created variables by scrolling to the right. Some of the new variables are from the transformations and data replacement that were done in the original flow. Other new variables have the predicted values from the regression model. The variable P\_TARGET\_B1 contains the predicted probability of responding to a card promotion.

20. Close the Insight data table when you are finished examining the data set.

-  This data set contains identifying information for each of the cases in the scored data set (the variable IDCODE). This identifying information could be extracted for the cases with the highest predicted probability of responding, matched with name and address information, and used to mail the next promotion.

### Scoring Using Base SAS (Self-Study)

As an alternative to scoring new data within Enterprise Miner, you can use the scoring code you saved earlier to score a data set using base SAS. Enterprise Miner runs on top of a SAS session. You can use this SAS session regardless of the current window in Enterprise Miner. Use SAS to score the MYSCORE data set in the CRSSAMP library.

- At the bottom of the SAS window, select the Program Editor button,  

- Select File  $\Rightarrow$  Open.
- Find and select the program that you saved (named MYCODE in this example).



If you use the default folder when saving the code, it will be in the same folder that opens when you select **File**  $\Rightarrow$  **Open**.

4. Select **Open**. The scoring code appears in the Program Editor of the SAS session. A portion of the code appears below.

```
/*-----*/
/*  ENTERPRISE MINER: BEGIN SCORE CODE */
/*-----*/
%macro DMNORLEN; 16 %mend DMNORLEN;

proc format;
value AVGG_ZQ_
low-10='01:low-10'
10-20='02:10-20'
20-high='03:20-high'
;
;
%macro DMNORMCP(in,out);

&out=substr(left(&in),1,min(%dmnorlen,length(left(&in)));
&out=upcase(&out);
%mend DMNORMCP;

%macro DMNORMIP(in);
&in=left(&in);
&in=substr(&in,1,min(%dmnorlen,length(&in)));
&in=upcase(&in);
%mend DMNORMIP;

DATA &_PREDICT ; SET &_SCORE ;
```

The data set \_PREDICT is the data set that is created containing the predicted values. The data set represented by \_SCORE is the data set you want to score. Because these data sets are being used in a macro (preceded by &\_), the data sets need to be initialized.

5. Score the MYSCORE data set in the CRSSAMP library. To do so, first initialize \_PREDICT and \_SCORE. Type the following code at the beginning of the program:

```
%let _score=crssamp.myscore;
%let _predict=x;
```

The second line will initialize \_PREDICT. There is actually no X data set. It is just a dummy name. The actual \_PREDICT data set is re-created by the scoring code.

6. To see the results of scoring, add the following code at the end of the program:

```
proc print data=&_predict;
  var idcode p_target_b1;
run;
```

This code will print IDCODE and P\_TARGET\_B1. Recall that P\_TARGET\_B1 is the predicted probability of a response.

7. Submit the scoring code by selecting **Run**  $\Rightarrow$  **Submit** or by selecing the Submit icon, , from the toolbar. Inspect the resulting output.

Obs	IDCODE	P_TARGET_B1
1	42176	0.06480
2	42177	0.06921
3	42179	0.03558
4	42180	0.06887
5	42181	0.04468
6	42185	0.04345
7	42187	0.06546
8	42188	0.07658
9	42189	0.15254
10	42190	0.05031
11	42192	0.06292
12	42196	0.05304
13	42201	0.02628
14	42202	0.07001
15	42203	0.04728
16	42204	0.10529
17	42206	0.04305
18	42207	0.06198
19	42208	0.08702
20	42209	0.04443
21	42210	0.03664
22	42213	0.05765
23	42214	0.12883
24	42219	0.04712
25	42222	0.06480
26	42223	0.04118
27	42224	0.05532
28	42225	0.05202
29	42228	0.09529
30	42229	0.07708
31	42231	0.02301
32	42234	0.04669
33	42235	0.05362
34	42236	0.07062

The cutoff for the regression model was approximately 0.055. To implement this model, send the mailing to people with a predicted probability of response (P\_TARGET\_B1) greater than or equal to 0.055.

Return to the Enterprise Miner workspace when you finish inspecting the results.

-  In addition to using the Score node for scoring data within SAS, the C-Score node can be used to generate a score function in the C programming language. This enables you to deploy the scoring algorithms in your preferred C or C++ development environment.



# Chapter 7 Cluster Analysis

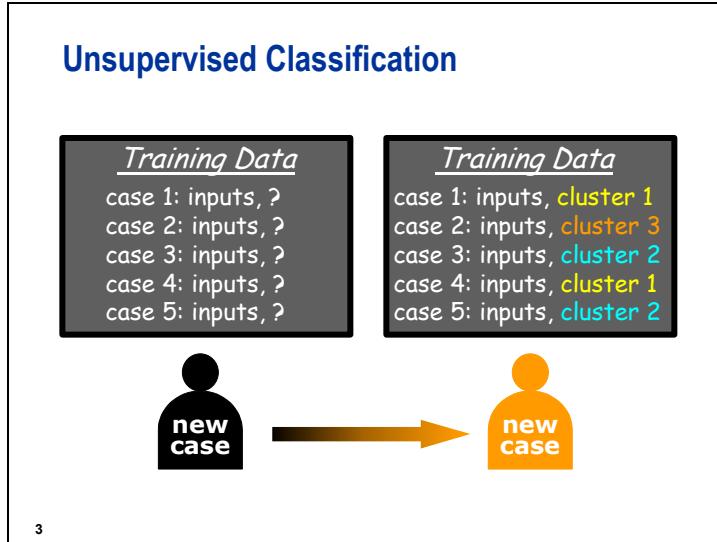
<b>7.1 K-Means Cluster Analysis.....</b>	<b>7-3</b>
<b>7.2 Self-Organizing Maps.....</b>	<b>7-24</b>



## 7.1 K-Means Cluster Analysis

### Objectives

- Discuss the concept of  $k$ -means clustering.
- Define measures of distance in cluster analysis.
- Understand the dangers of forced clustering.
- Generate a cluster analysis and interpret the results.

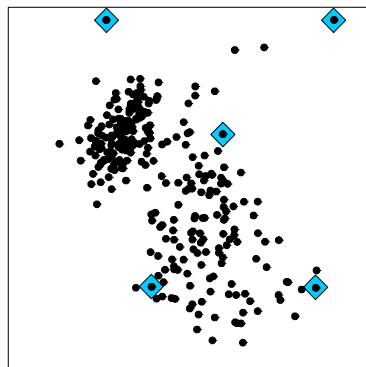


*Unsupervised classification* (aka *clustering*) is classification with an unknown target. That is, the class of each case is unknown. Furthermore, the total number of classes is unknown. The aim is to segment the cases into disjoint classes that are homogenous with respect to the inputs.

The purpose of clustering is often description. For example, segmenting existing customers into groups and associating a distinct profile with each group could help future marketing strategies. However, there is no guarantee that the resulting clusters will be meaningful or useful.

Unsupervised classification is also useful as a step in a supervised prediction problem. For example, customers could be clustered into homogenous groups based on sales of different items. Then a model could be built to predict the cluster membership based on some more easily obtained input variables.

## K-means Clustering

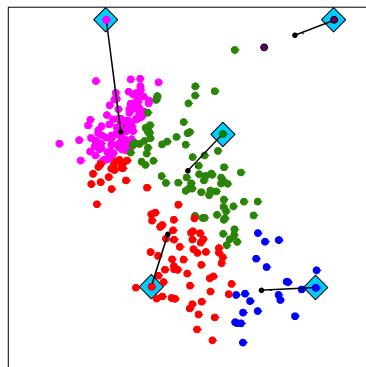


4

*K*-means clustering is a computationally efficient unsupervised classification method. The first two steps of the method are as follows:

1. Specify the number of clusters (classes)  $k$ . In Enterprise Miner, the number of clusters is determined by the cubic clustering criterion.
2. Choose  $k$  initial cluster seeds.

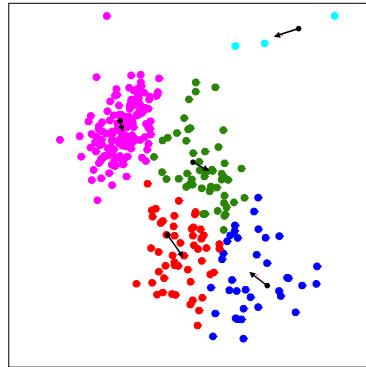
## Assignment



5

3. Assign cases closest to seed  $i$  as belonging to cluster  $i$ ;  $i = 1, \dots, k$ .
4. Calculate the mean of the cases in each cluster, and move the  $k$  cluster seeds to the mean of their cluster.

### Reassignment

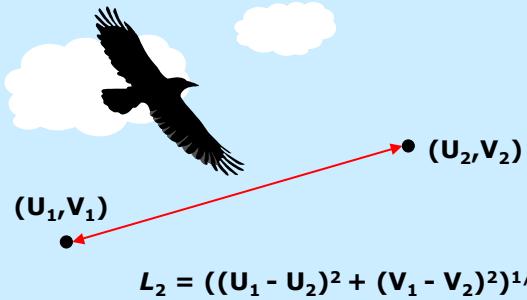


6

5. Reassign cases closest to the new seed  $i$  as belonging to cluster  $i$ .
6. Take the mean of the cases in each cluster as the new cluster seed.

This process can be further iterated, but this is rarely necessary, provided the initial cluster seeds are placed intelligently.

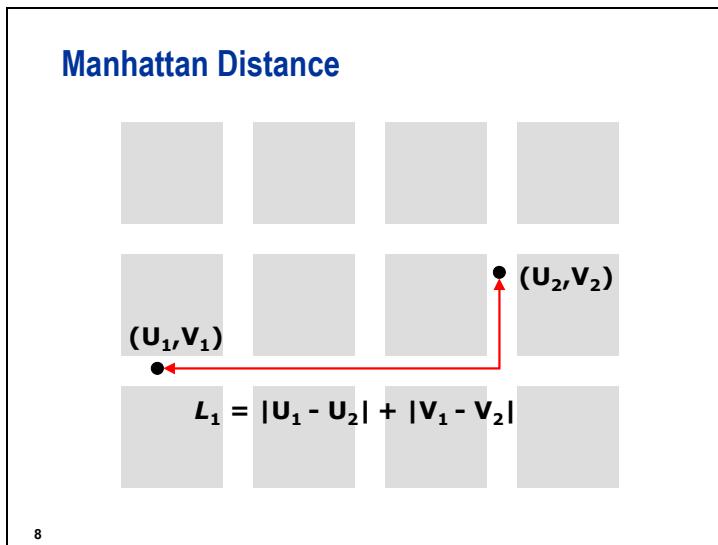
### Euclidean Distance



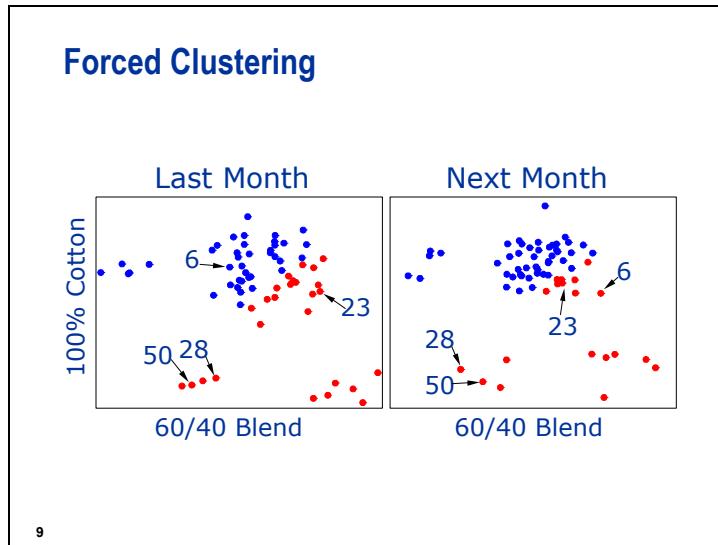
7

Clustering methods depend on a measure of distance or similarity between points. Different distance metrics used in  $k$ -means clustering can give different types of clusters.

The most widely used metric is Euclidean distance ( $L_2$  norm). The Euclidean distance between two points is the length of the straight line that joins them. Clusters formed using Euclidean distance tend to be spherical in nature.



The Manhattan distance ( $L_1$  norm) between two points is the length of the shortest axis-parallel connection between them. The formation of clusters using the Manhattan metric is relatively insensitive to outlying data points. Clusters formed using Manhattan distances tend to be more cubical in shape.



One pitfall of problem formulation is failing to appreciate the limitations of the analytical methods.

Consider a retail allocation-planning problem. The objective is to find two roughly equal-sized clusters of stores that are similar with respect to the sales of two shirt styles. The results will be used to allocate styles to the stores. The hope is that this will be an improvement over the current system of treating every store the same.

In reality, there were four well-separated clusters of stores: one large cluster and three small ones. Cluster analysis forced the stores into two roughly equal-sized groups (red and blue). The clusters are artificial. For example, based on the analysis of last month's data, stores 6 and 23 received different allocations. However, their apparent difference in sales was just random variation within their cluster. In contrast, assignment of the stores to the same true clusters would have produced stable results.

Failure of the cluster analysis to give useful results is not the fault of the analytical method. The problem was not formulated realistically. It would have been better to first attempt to discover what natural clusters exist and then determine whether they are practically useful.



## Cluster Analysis

A catalog company periodically purchases lists of prospects from outside sources. They want to design a test mailing to evaluate the potential response rates for several different products. Based on their experience, they know that customer preference for their products depends on geographic and demographic factors. Consequently, they want to segment the prospects into groups that are similar to each other with respect to these attributes.

After the prospects have been segmented, a random sample of prospects within each segment will be mailed one of several offers. The results of the test campaign will allow the analyst to evaluate the potential profit of prospects from the list source overall as well as for specific segments.

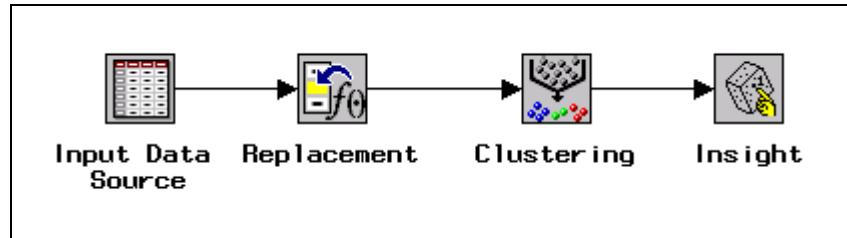
The data that was obtained from the vendor is in the table below. The prospects' name and mailing address (not shown) were also provided.

Name	Model Role	Measurement Level	Description
AGE	Input	Interval	Age in years
INCOME	Input	Interval	Annual income in thousands
MARRIED	Input	Binary	1=married, 0=not married
GENDER	Input	Binary	F=female, M=male
OWNHOME	Input	Binary	1=homeowner, 0=not a homeowner
LOCATION	Rejected	Nominal	Location of residence (A-H)
CLIMATE	Input	Nominal	Climate code for residence (10,20, & 30)
FICO	Input	Interval	Credit score
ID	ID	Nominal	Unique customer identification number

Observe that all variables except ID and LOCATION should be set to input. No target variables are used in a cluster analysis or SOMs. If you want to identify groups based on a target variable, consider a predictive modeling technique and specify a categorical target. This type of modeling is often referred to as *supervised classification* because it attempts to predict group or class membership for a specific categorical response variable. Clustering, on the other hand, is referred to as *unsupervised classification* because it identifies groups or classes within the data based on all the input variables.

### Building the Initial Flow

1. Open a new diagram and title it **Cluster Analysis**.
2. Assemble the following diagram and connect the nodes as shown.



The Replacement node is not critical in cluster analysis. If missing value imputation is not done, clustering is based on the non-missing inputs.

### Setting Up the Input Data Source

1. Open the Input Data Source node.
2. Select the **PROSPECT** data set from the CRSSAMP library.
- Because the CLIMATE variable is a grouping of the LOCATION variable, it is redundant to use both. CLIMATE was chosen because it had fewer levels (3 versus 8) and business knowledge suggested that these 3 levels were sufficient.
3. Set the model role of LOCATION to rejected.
4. Explore the distributions and descriptive statistics as desired.

Input Data Source						
Data	Variables	Interval Variables	Class Variables	Notes		
Name	Model Role	Measurement	Type	F		
ID	id	nominal	char	\$		
AGE	input	interval	num	B		
INCOME	input	interval	num	B		
GENDER	input	binary	char	\$		
MARRIED	input	binary	num	B		
FICO	input	interval	num	B		
OWNHOME	input	binary	num	B		
LOCATION	rejected	nominal	char	\$		
CLIMATE	input	nominal	char	\$		

5. Select the Interval Variables tab and observe that there are only a few missing values for AGE, INCOME, and FICO.
6. Select the Class Variables tab and observe that only a small percent of demographic variables are missing.
7. Close the Input Data Source window, saving changes when prompted.



Because of the small number of missing values, use the defaults for the Replacement node.

### Setting Up the Clustering Node

1. Open the Clustering node.

The Variables tab is active when you open the Cluster node. *K*-means clustering is very sensitive to the scale of measurement of different inputs. Consequently, it is advisable to use one of the standardization options if the data has not been standardized previously in the flow.

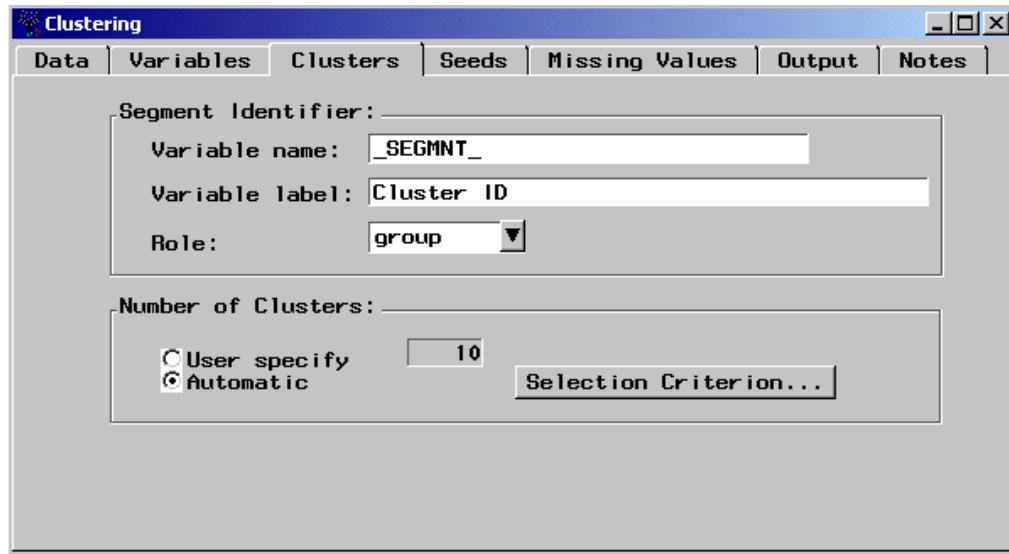
2. Select the Std Dev. radio button on the Variables tab.

Name	Status	Model Role	Measurement	Type	Format	Label
ID	use	id	nominal	char	\$9.	
AGE	use	input	interval	num	BEST12.	
INCOME	use	input	interval	num	BEST12.	
GENDER	use	input	binary	char	\$1.	
MARRIED	use	input	binary	num	BEST12.	
FICO	use	input	interval	num	BEST12.	
OWNHOME	use	input	binary	num	BEST12.	
LOCATION	don't use	rejected	nominal	char	\$1.	
CLIMATE	use	input	nominal	char	\$2.	



It should be noted that categorical variables tend to dominate cluster analysis because of their pure separation. When dealing with a mixture of categorical and continuous variables as cluster analysis inputs, it may be better to use multidimensional scaling on the categorical variables and use the results from that as input to the cluster analysis.

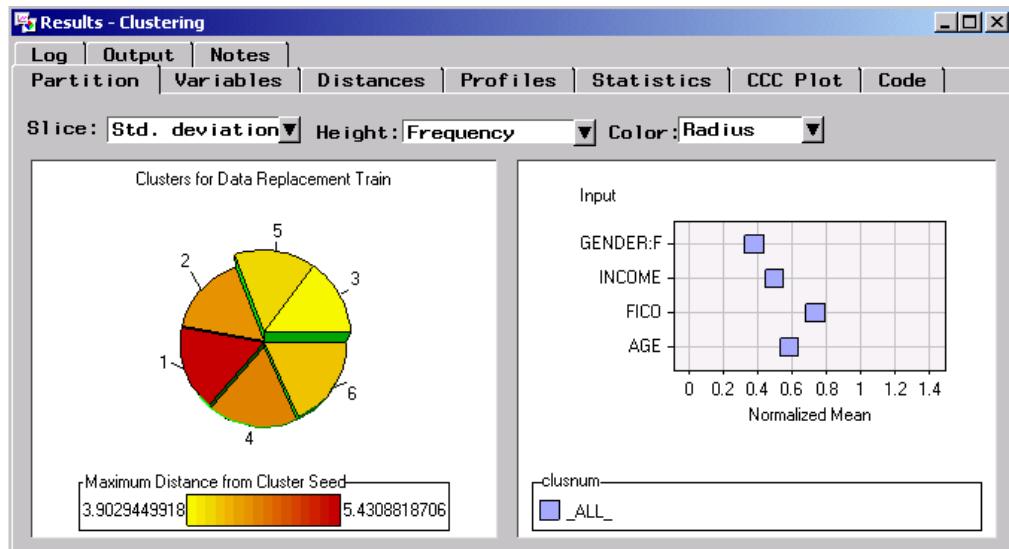
3. Select the **Clusters** tab.



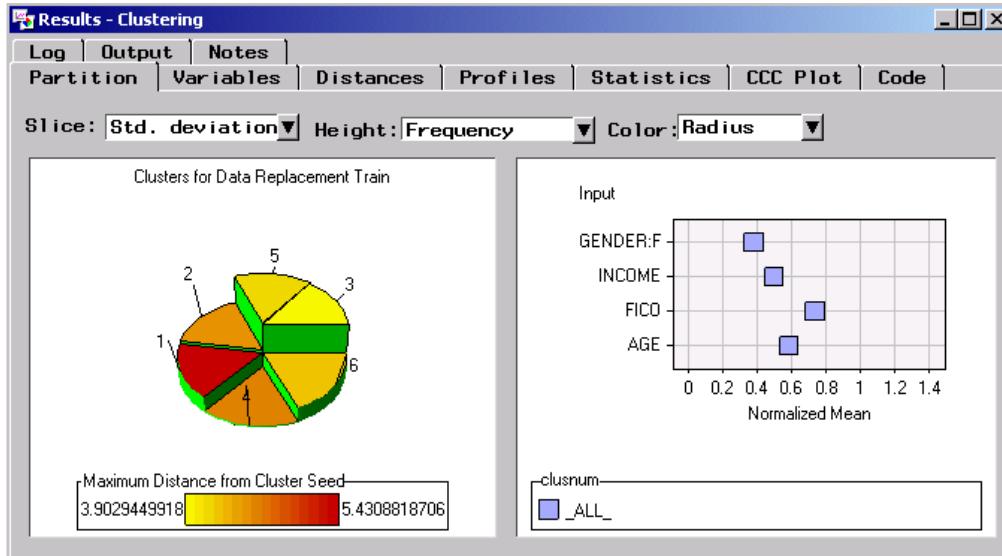
Observe that the default method of determining the number of clusters is automatic. This means the Cubic Clustering Criterion (CCC) based on a sample of 2,000 observations will be used to estimate the appropriate number of clusters. You can change the default sample size by selecting the Data tab and then selecting the Preliminary Training and Profiles tab. The Automatic selection of the number of clusters can be overridden by selecting the User specify radio button.

4. Note that the default name of the cluster identification in the output data set will be `_SEGMENT_`. Change this variable name to **CLUSTER**.
5. Close the Clustering node and save the changes when prompted.
6. Run the diagram from the Clustering node and view the results.

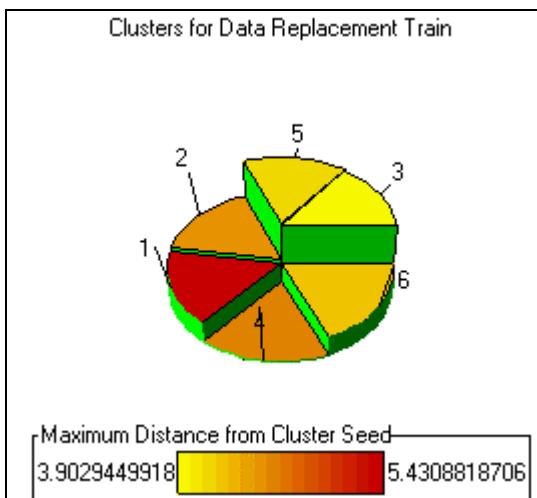
### Exploring the Cluster Node Results



- Select the Tilt icon, , from the toolbar and tilt the pie chart as shown below.



Inspect the chart in the left window of the Partition tab.



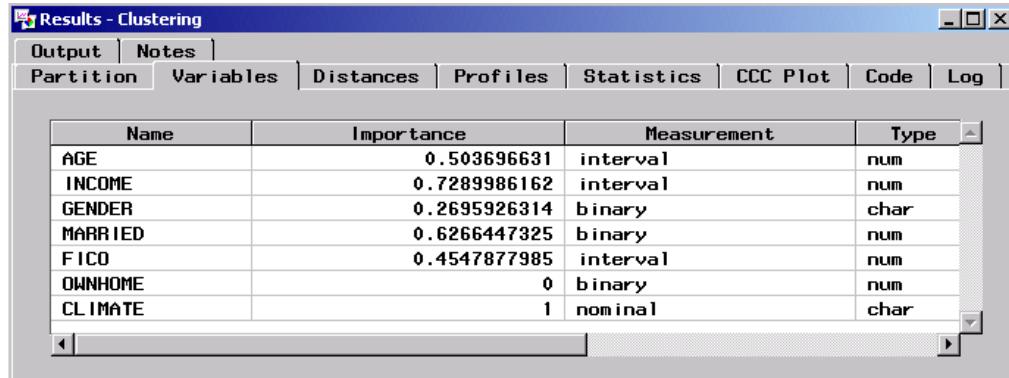
This chart summarizes three statistics for each of the six clusters. The height of the slice indicates the number of cases in each cluster. Clusters 3 and 5 contain the most cases, and cluster 4 contains the fewest. The width of each slice is set to Std. deviation, which is the root-mean-square standard deviation (root-mean-square distance) between cases in the cluster. The color is set to radius, which is the distance of the farthest cluster member from the cluster seed.



The plot at the right side of the window is a normalized mean plot. As documented in SAS Note SN-006805, if any input variables are categorical and Std. deviation standardization is used, the normalized mean plot is incorrect.

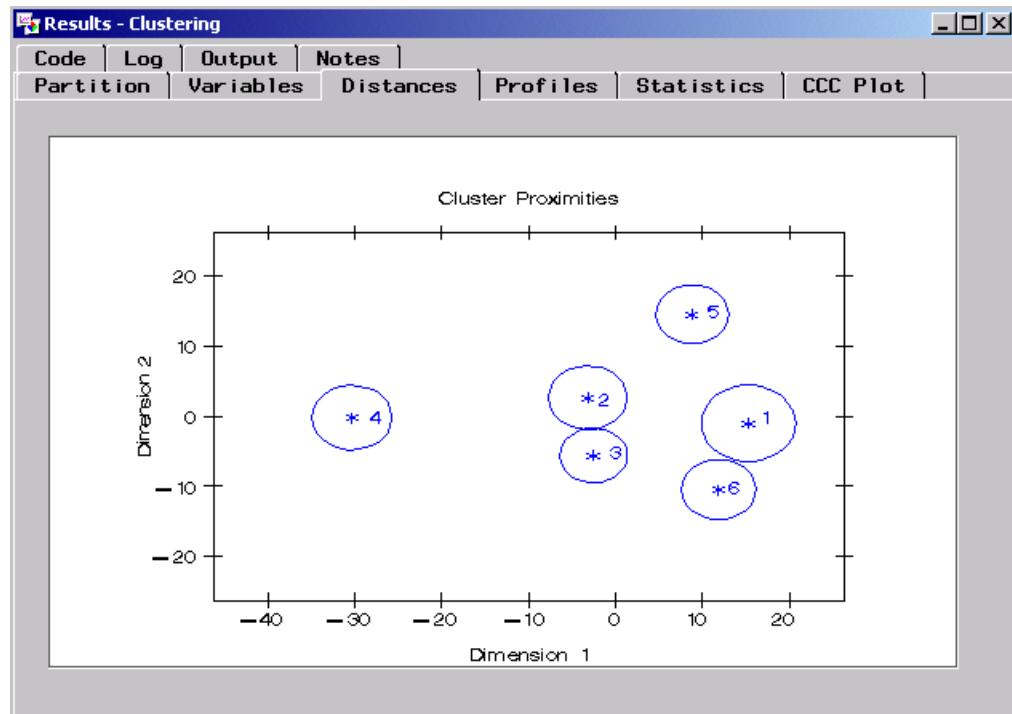
Explore the other result tabs.

2. Select the **Variables** tab.



This tab shows the relative importance of the variables in determining the clusters. The importance measures range between 0 and 1, with higher numbers indicating a more important, or influential, variable. In this case, CLIMATE appears to have been the most important variable.

3. Select the **Distances** tab.



The Distances tab provides a graphical representation of the size of each cluster and the relationship among clusters. The axes are determined from multidimensional scaling analysis using a matrix of distances between cluster means as input. The asterisks are the cluster centers, and the circles represent the cluster radii. A cluster that contains only one case is displayed as an asterisk. The radius of each cluster depends on the most distant case in that cluster, and cases may not be distributed

uniformly within clusters. Hence, it may appear that clusters overlap, but in fact each case is assigned to only one cluster.

4. Select the **Statistics** tab.

CLUSTER	Frequency of Cluster	Root-Mean-Square Standard Deviation	Maximum Distance from Cluster Seed	Nearest Cluster
1	656	0.7903085529	5.4308818706	4
2	719	0.7881703633	4.5342267794	5
3	1341	0.7083995568	3.9029449918	5
4	317	0.8449183857	4.5881037346	1
5	1380	0.7387256638	4.1668516015	3
6	642	0.8718606232	4.2594857685	5

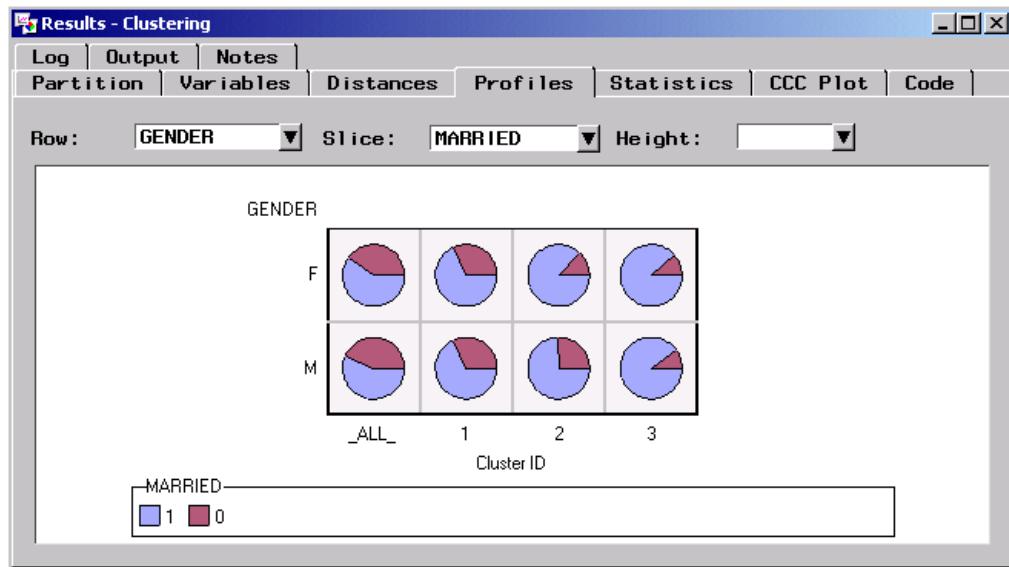
The tab gives descriptive statistics and other information about the clusters such as the frequency of the cluster, the nearest cluster, and the average value of the input variables in the cluster.

5. Scroll to the right to see the statistics provided for the input variables.

CLUSTER	AGE	INCOME	FICO	GENDER:F	MARRIED:0	OWNHOME:0
1	46.616389467	44.543049532	706.07758396	0.4634146341	0.3125	0.6844512195
2	46.773007503	49.195285933	687.79525753	0.4422809458	0.1988873435	0.8386648122
3	50.643216204	40.822037289	687.72287135	0.5219985086	0.1029082774	0.7233407905
4	37.320930097	48.176498909	661.14477943	0.3217665615	0.7003154574	0.7160883281
5	40.962872187	60.605565485	700.85984081	0.2847826087	0.5833333333	0.5137681159
6	35.540498442	35.660436137	702.82661252	0.7585669782	0.9143302181	0.738317757

Notice that

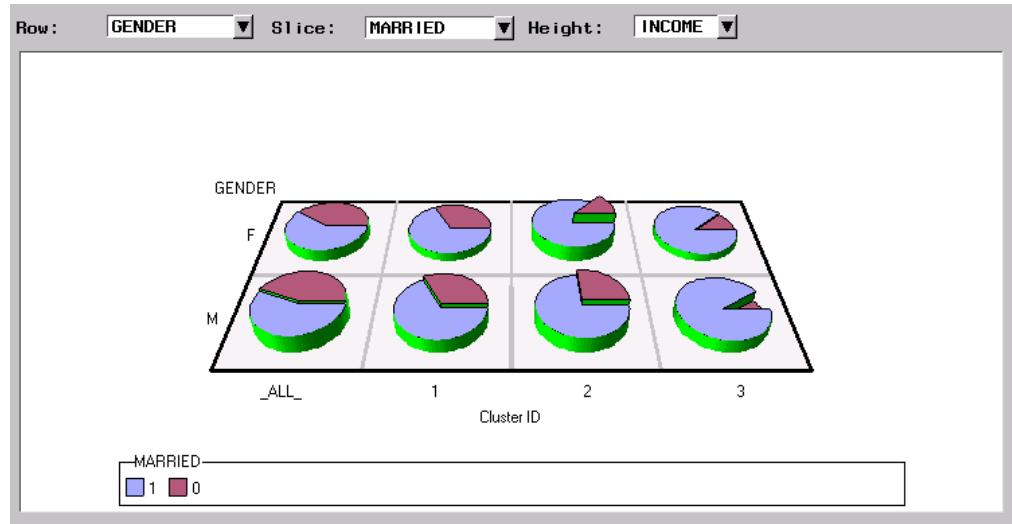
- the prospects in clusters 4 and 6 are younger on the average
- the prospects in cluster 6 have lower average incomes, and the prospects in cluster 5 have higher average incomes
- prospects in cluster 6 tend to be married females.

6. Select the **Profiles** tab.

The Profiles tab allows you to explore three variables at a time for each cluster: two class variables and an interval variable. The rows and slices each represent a class variable, and the heights of the slices can represent an interval variable. Explore the relationships between variables and clusters.

7. Select the drop-down arrow next to height and select **INCOME** as the height variable.

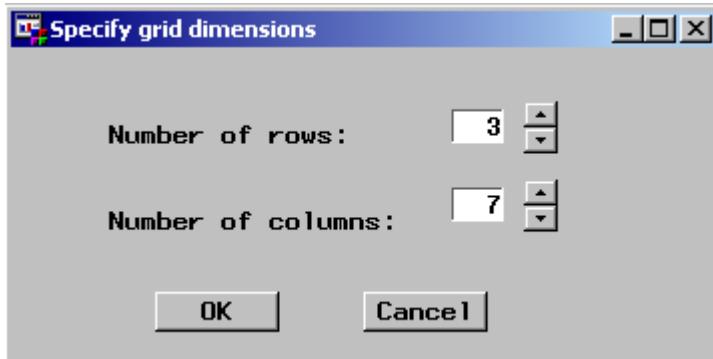
8. Select the Tilt tool, , and use it to tilt the grid as shown below.



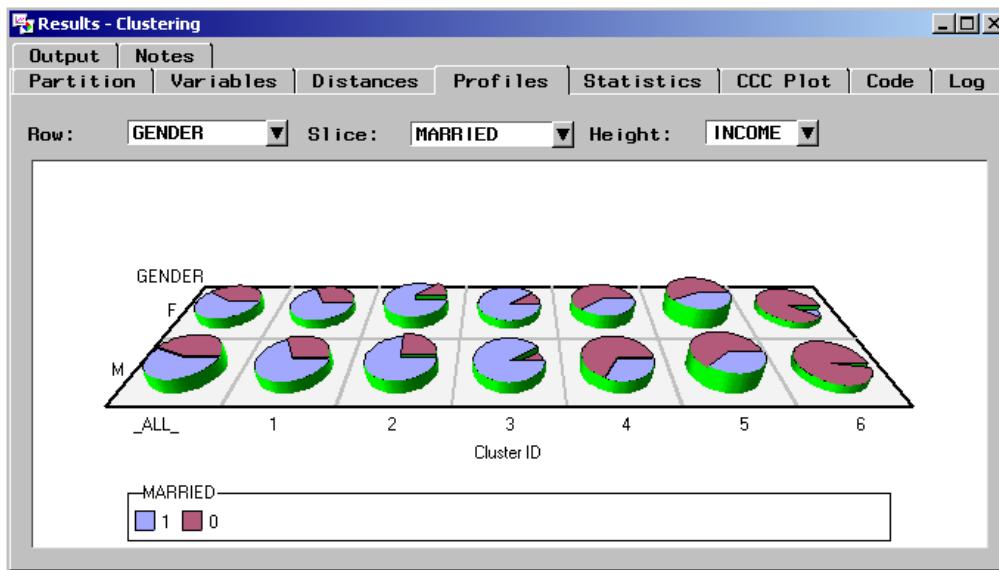
Looking at the grid you can form some general conclusions about some of the clusters. For example, cluster 2 has a smaller percentage of single people than the overall population, but the single people in cluster 2 have higher incomes than the married people in the cluster.

To view the remaining clusters not currently shown in the grid, select the Scroll Data tool, , and use it to scroll the horizontal axis to the right.

9. To view all of the clusters at the same time, select **Format**  $\Rightarrow$  **Set Grid Dimensions...**
10. Set the number of rows to 3 and the number of columns to 7. This will enable you to view the overall pie chart and all six clusters and also any categorical variable with up to 3 levels, such as CLIMATE.

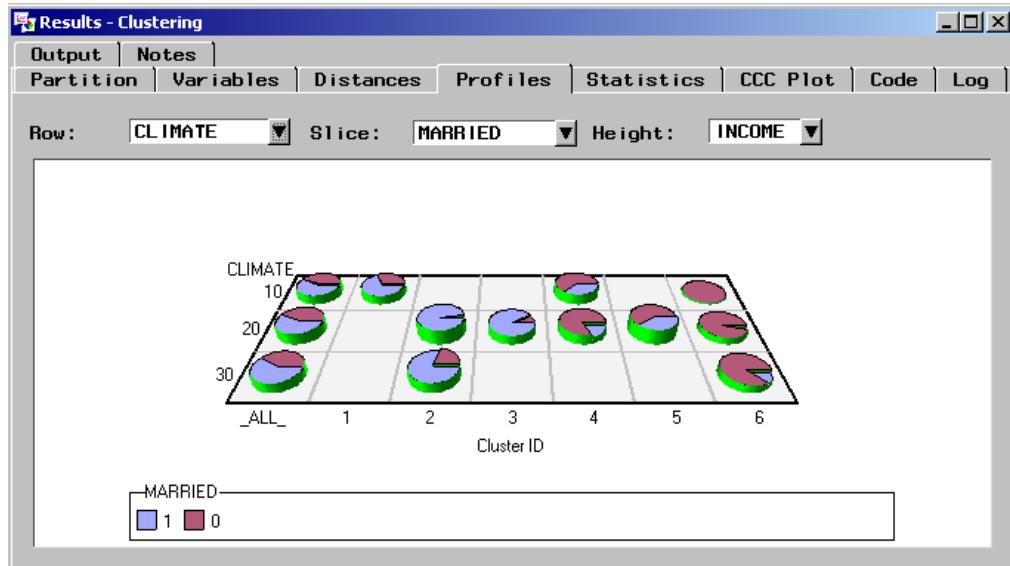


11. Select **OK**.



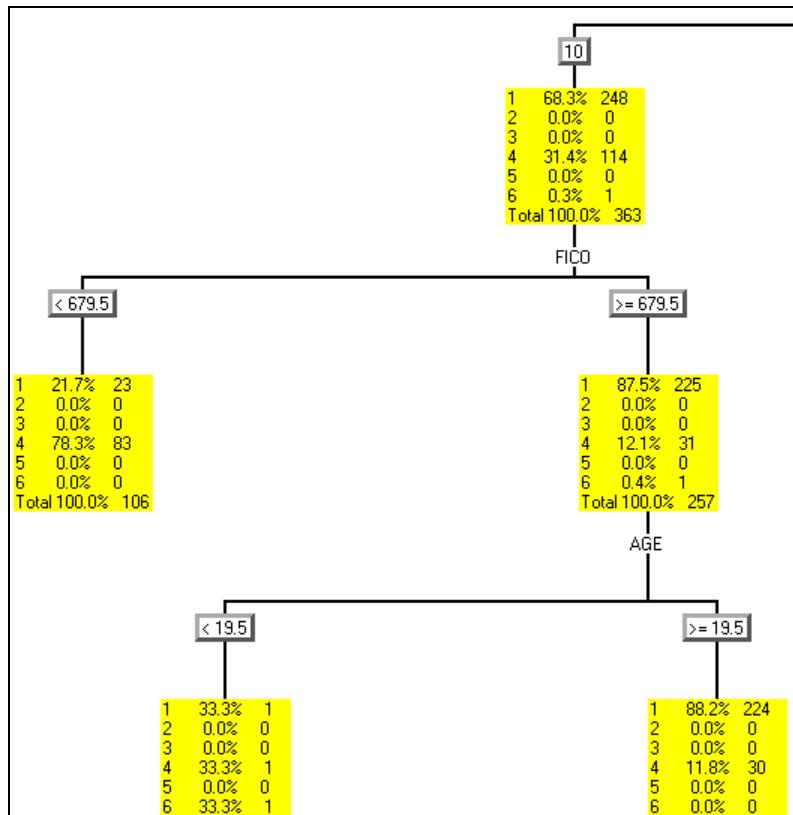
This picture shows only two rows because GENDER has only two values.

12. Change the row variable to CLIMATE. The grid is large enough to accommodate all three levels of CLIMATE.



Another way to examine the clusters is with the cluster profile tree.

13. Select View  $\Rightarrow$  Cluster Profile Tree.



The profile tree lists the percentages and numbers of cases assigned to each cluster and the threshold values of each input variable displayed as a hierarchical tree. It

enables you to see which input variables are most effective in grouping cases into clusters.

14. Close the Cluster node results window when you have finished exploring the results.

### Exploring the Results Using the Insight Node

The Insight node can also be used to compare the differences among the attributes of the prospects in the clusters.

1. Open the Insight node and choose the option to use the entire data set.
2. Close the Insight node, saving the changes when prompted.
3. Run the flow from the Insight node and view the results.

	11	Nom	Int	Int	Nom	Int	FICO
	5055	ID	AGE	INCOME	GENDER	MARRIED	
■	1	000595865	44	54	M	1	
■	2	001038701	44.14696814	47.390030832	M	1	
■	3	001158405	38	40	F	0	
■	4	001365475	25	65	M	0	
■	5	001446031	45	55	M	1	
■	6	001522174	43	53	M	1	
■	7	001641733	47	65	M	0	
■	8	001796166	45	19	F	0	
■	9	001866714	45	74	F	0	
■	10	002384317	55	35	F	1	
■	11	003210135	27	20	M	1	
■	12	003395270	54	73	M	0	
■	13	003470679	50	21	F	1	
■	14	003615261	46	53	M	1	
■	15	003631239	36	55	F	0	
■	16	003757911	60	51	M	0	
■	17	004181057	36	42	F	0	
■	18	004426926	41	50	M	1	

All of the observations in the original data set are present, but the number of columns has increased from 9 to 11.

4. Right-click in the data table and select **Data Options...** from the pop-up menu.
5. In the Data Options window, select **Show Variable Labels**.
6. Select **OK** to return to the data table.
7. Scroll to identify the two new columns.

	11	Int	Nom	Nom	Int		Int
	OWNHOME	LOCATION	CLIMATE	CLUSTER Cluster ID		distance	Distance to Cluster Seed
5055	1	0 G	30	2		2.1343221312	
	2	0 F	20	5		2.0162742361	
	3	0 E	20	6		2.0841092653	
	4	1 B	20	5		2.8782000573	
	5	0 B	20	3		1.7718862874	
	6	0 G	30	2		2.6206001303	
	7	1 F	20	5		2.3343740794	
	8	0 F	20	6		2.5318148893	
	9	1 E	20	5		3.0647856104	
	10	0 F	20	3		2.1828710338	

The column CLUSTER identifies the cluster, and the column DISTANCE identifies the distance from each observation to the cluster mean.

You can use the analytical tools within Insight to evaluate and compare the clusters. The following steps represent one way to make these comparisons.

In Insight, there are two measurement levels, nominal and interval. By default, all numeric variables are set to interval and all character variables are set to nominal. This is not always appropriate, and in this case some of these assignments must be changed.

1. Change the measurement scale for MARRIED to nominal. Select the measurement scale, Int, directly above the variable name, and then select Nominal.
2. Repeat step 1 for the variables OWNHOME and CLUSTER.

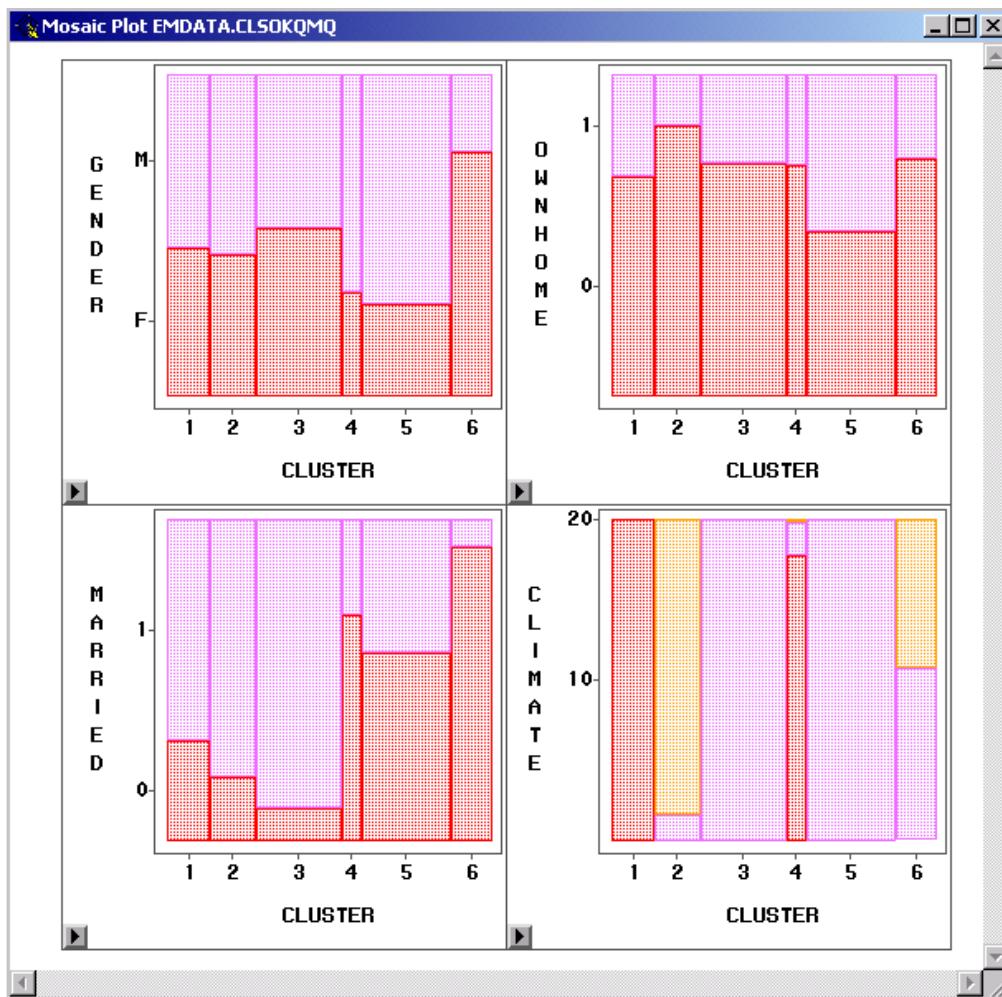
	11	Nom	Int	Nom	Nom	Nom	Nom	Nom	CLUSTER Cluster ID	
	MARRIED	FICO	OWNHOME	LOCATION	CLIMATE		distance	Distance to Cluster Seed		
5055	1	1	694	0 G	30	2	2	2		
	2	1	719	0 F	20	5	2	2		
	3	0	692	0 E	20	6	2	2		
	4	0	649	1 B	20	5	2	2		
	5	1	683	0 B	20	3	1	1		
	6	1	727	0 G	30	2	2	2		
	7	0	655	1 F	20	5	2	2		
	8	0	687	0 F	20	6	2	2		
	9	0	642	1 E	20	5	3	3		
	10	1	724	0 F	20	3	2	2		
	11	1	715	0 C	10	1	2	2		
	12	0	735	0 B	20	5	2	2		
	13	1	662	0 A	10	1	2	2		

3. Select Analyze  $\Rightarrow$  Box Plot/Mosaic Plot (Y).
4. Highlight CLUSTER  $\Rightarrow$  X.
5. Select GENDER.
6. Press and hold the Ctrl key.
7. Select MARRIED, OWNHOME, and CLIMATE.

8. Select Y.

9. Select OK.

The Mosaic plots should appear as below. The width of the columns indicates the number of cases in each cluster. The colors indicate the percentage of cases for each level of the variable on the vertical axes.



CLIMATE is important in distinguishing among the clusters, with five of the six clusters entirely or almost entirely containing cases that live in only one climate zone. Note that clusters 3 and 5 contain prospects that live only in climate zone 20. Similarly, clusters 1 and 4 contain prospects that live mostly in climate zone 10. Consequently, they must differ by other attributes.

Clusters 3 and 5 differ substantially by the percent of married persons as do clusters 1 and 4. Cluster 6 appears to be evenly distributed between climate zones 10 and 30. Cluster 6, however, has a much higher percentage of females and unmarried persons than most of the other clusters.

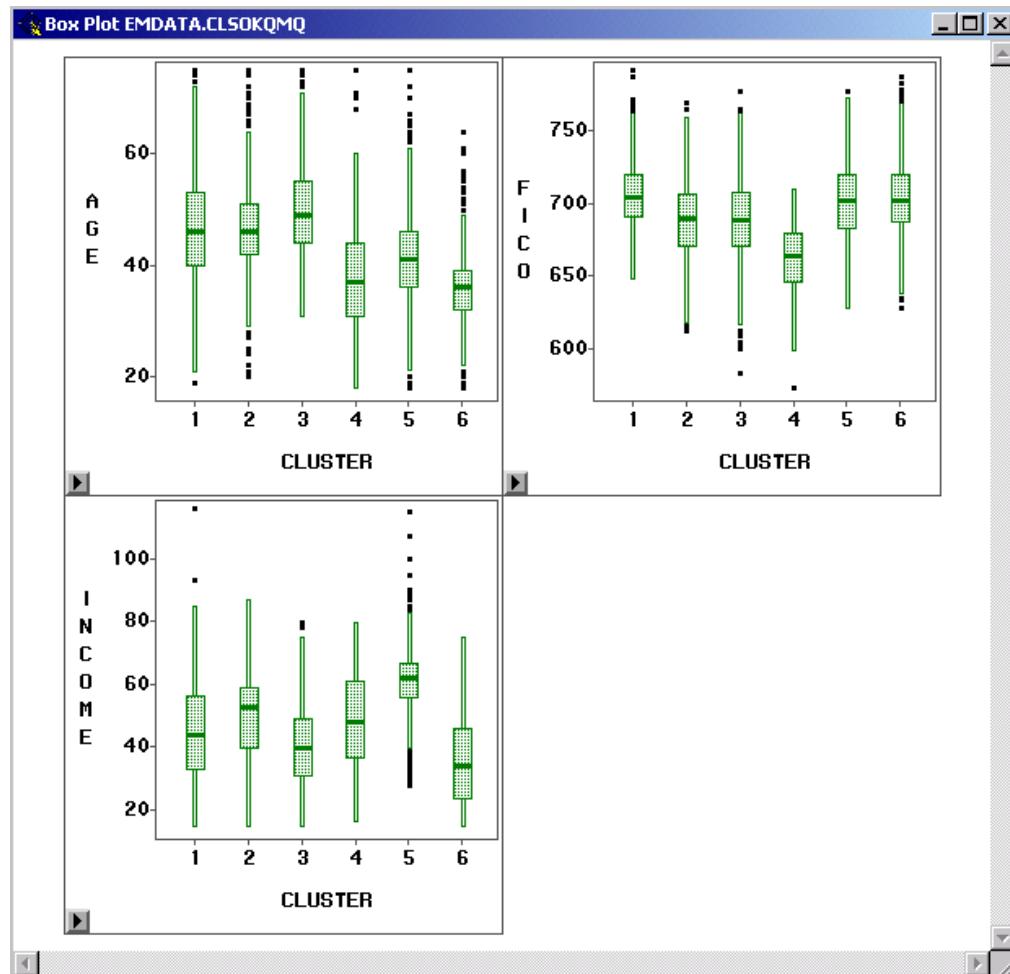


At times it can be difficult to determine which levels of the categorical variables are represented by the different colors, as in the mosaic plot for CLIMATE. When in doubt, you can double-click on any section of the graph. This opens a window that contains information about all of the observations represented by that segment, which in turn enables you to see what level of CLIMATE is represented.

The Insight node can also be used to compare the distributions of the interval inputs among the clusters to ascertain their contribution to explaining the cluster differences.

1. Select **Analyze**  $\Rightarrow$  **Box Plot/Mosaic Plot (Y)**.
2. Select **CLUSTER**  $\Rightarrow$  **X**.
3. Control-click to select **AGE**, **INCOME**, **FICO**  $\Rightarrow$  **Y**.
4. Select **OK**.

The Box plots should appear as below.



Cluster 6 appears to be the lowest income group and is among the clusters with younger members.

Cluster 4 contains members with lower FICO scores.

In summary, the six clusters can be described as follows:

- Cluster 1      married persons living in climate zone 10
- Cluster 2      married persons living in climate zone 30
- Cluster 3      married persons living in climate zone 20
- Cluster 4      younger, unmarried persons with lower FICO scores, living in climate zone 10
- Cluster 5      younger, unmarried men with higher incomes, living in climate zone 20
- Cluster 6      younger, unmarried women living in climate zone 20 or 30.

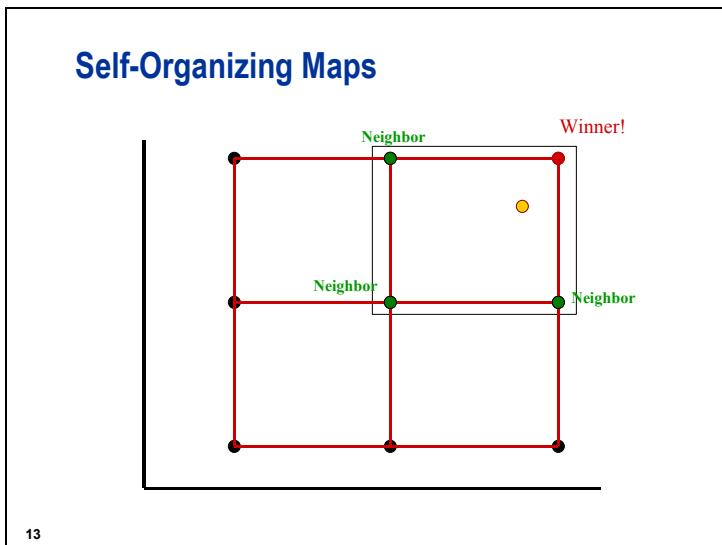
These clusters may or may not be useful for marketing strategies, depending on the line of business and planned campaigns.

5. Close the plot and data table windows to return to the Enterprise Miner workspace.

## 7.2 Self-Organizing Maps

### Objectives

- Discuss the concept of self-organizing maps.
- Generate a self-organizing map and interpret the results.

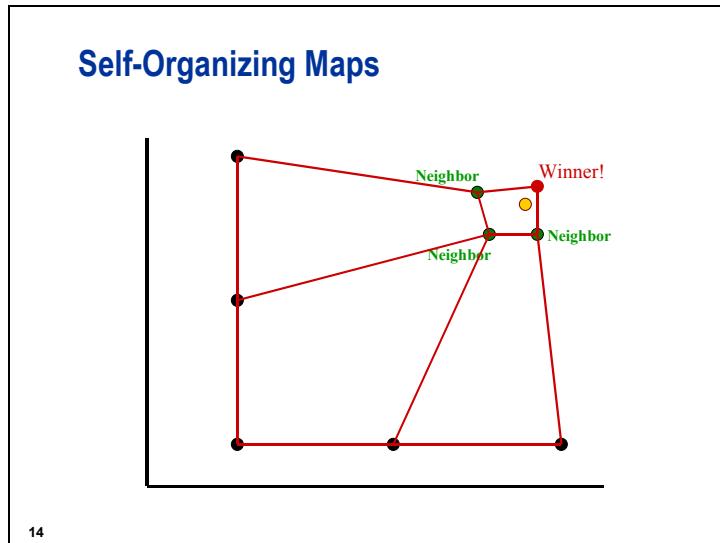


A self-organizing map (SOM) is an analytical tool that provides a mapping from the input space to the clusters. In a SOM, the clusters are organized into a grid. The grid is usually two-dimensional, but sometimes it is one-dimensional, and (rarely) three-dimensional or higher. In Enterprise Miner, only one-dimensional and two-dimensional grids are available.

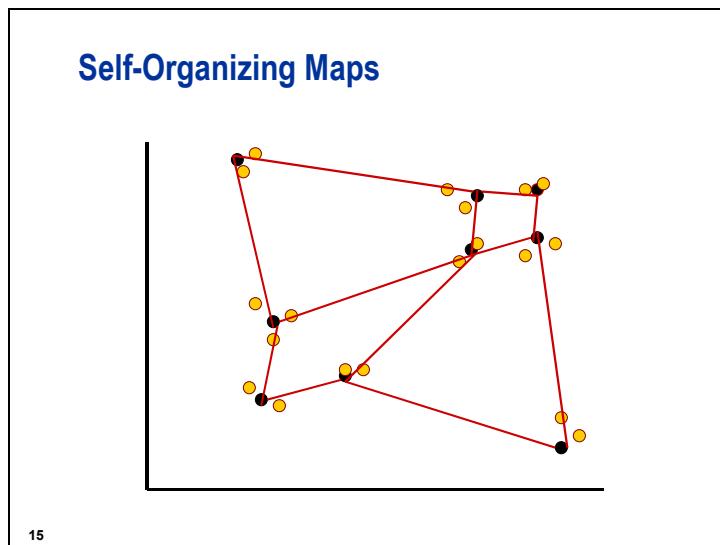
The grid exists in a space that is separate from the input space; any number of inputs can be used, but the dimension of the input space should be larger than the dimension of the grid. In this situation, dimension is not referring to the number of input variables, but the number of cases or observations. Smaller grid dimensions result in fewer clusters and make interpretation easier. However, ease of interpretation and the usefulness of the clusters need to be balanced with the homogeneity of the clusters. Larger grid dimensions may be better as long as most clusters have at least 5-10 cases, but these larger maps take longer to train. Choosing a useful map size generally requires trial and error. The usefulness of the map is generally more sensitive to the shape of the grid than the size of the grid.

SOMs differ from  $k$ -means clusters because in  $k$ -means clustering, cases are grouped together based on their Euclidean distance from each other in the input space. An SOM tries to find clusters such that any two clusters that are close together in the grid space have seeds that are close in the input space.

Unlike  $k$ -means clustering, the building of an SOM involves a neighborhood function that defines an area around each cluster (seed) in the grid. Observations can influence the position of all cluster seeds within this neighborhood, not just the position of the closest seed. This results in a final cluster solution in which clusters that are close together on the grid are similar with respect to the input variables.

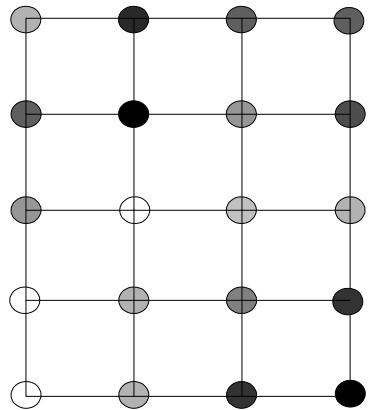


The “winning” unit is moved closer to the input. The amount of movement is proportional to the Euclidean distance between the unit and the input. Other units in the neighborhood are also moved closer to the input. A neighbor unit’s distance from the input determines the amount of movement of the unit. The closer to the input the unit is, the more it moves.



This process is repeated with the neighborhood size becoming smaller with each iteration.

### Self-Organizing Maps



17

The result is a grid in which the clusters closer to one another contain observations that are more similar than clusters farther apart.

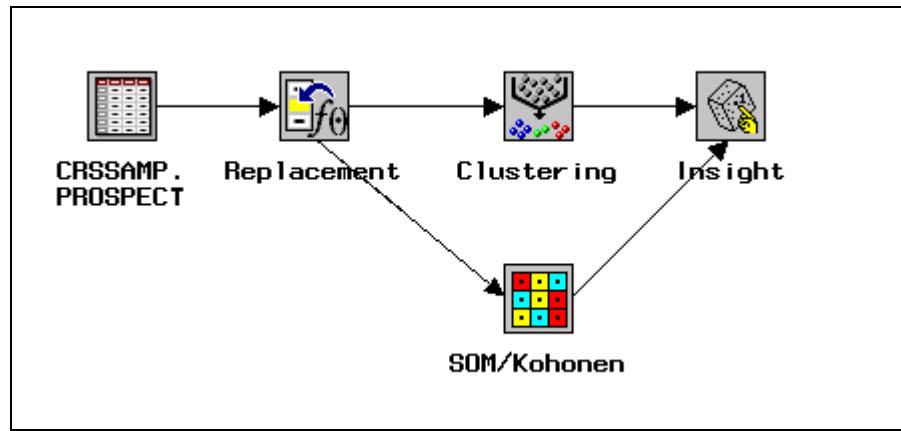


## Self-Organizing Maps

### Setting Up the SOM/Kohonen Node

As was previously mentioned, in Enterprise Miner, only one- and two-dimensional self-organizing maps are available. In addition, the default grid size is 4 by 6, or 24 clusters. Consequently, when working on small problems, you should consider reducing the grid dimensions.

1. Add a SOM/Kohonen node and connect it between the Replacement node and the Insight node. Your diagram should now look like the one pictured below.



2. Open the SOM/Kohonen node.

The Variables tab appears first. Inspect the options.

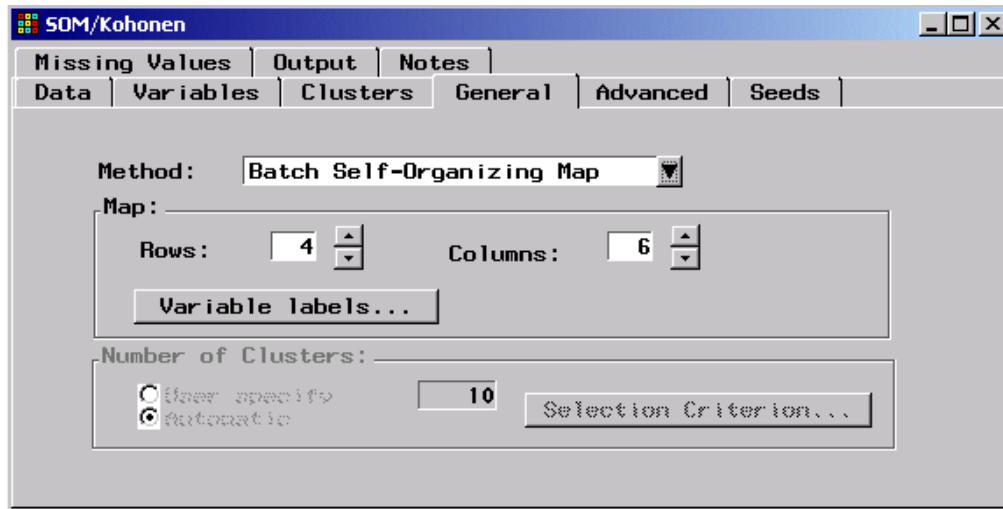
Name	Status	Model	Role	Measurement	Type	Format	Label
ID	use	id	nominal	char	\$9.		
AGE	use	input	interval	num	BEST12.		
INCOME	use	input	interval	num	BEST12.		
GENDER	use	input	binary	char	\$1.		
MARRIED	use	input	binary	num	BEST12.		
FICO	use	input	interval	num	BEST12.		
OWNHOME	use	input	binary	num	BEST12.		
LOCATION	don't use	rejected	nominal	char	\$1.		
CLIMATE	use	input	nominal	char	\$2.		

As in  $k$ -means clustering, the scale of the measurements can heavily influence the determination of the clusters. Standardizing the inputs is recommended.

- Select the **Standardize** radio button from the Variables tab to standardize the input variables.

Standardization:		<input type="radio"/> None	<input type="radio"/> Range	<input checked="" type="radio"/> Standardize		
Name	Status	Model Role	Measurement	Type	Format	Label
ID	use	id	nominal	char	\$9.	

- Select the **General** tab. The default method is a Batch Self-Organizing Map.



You can specify three options in the method field using the drop-down arrow including Batch Self-Organizing Map, Kohonen Self-Organizing Map, and Kohonen Vector Quantization (VQ), which is a clustering method.

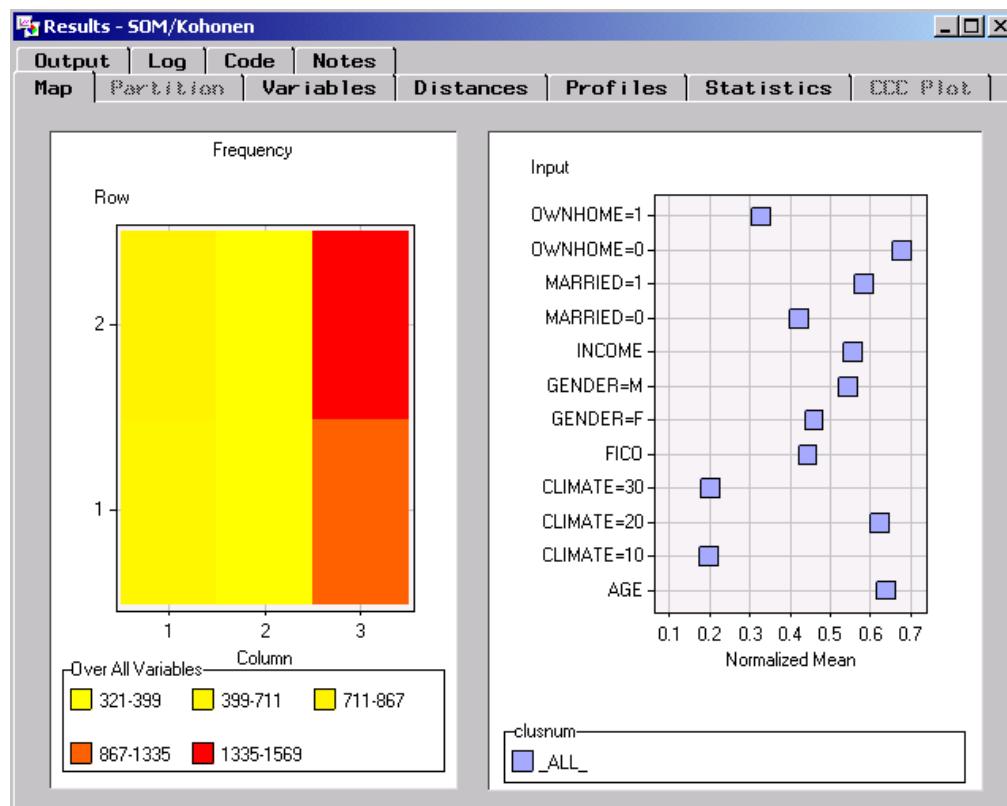
The Cluster node is recommended over Kohonen VQ for clustering. Also, for many situations, Batch SOMs obtain satisfactory result and are computationally more efficient. However, Kohonen SOMs are recommended for highly nonlinear data.

- To facilitate comparing the clusters from the SOM/Kohonen node to those determined in the Cluster node, choose a grid space that corresponds to six clusters. Use the arrows to specify 2 for the number of rows and 3 for the number of columns.

- Close the SOM/Kohonen node and save the settings.
- Run the diagram from the SOM/Kohonen node and view the results.

### Exploring the SOM/Kohonen Node Results

The SOM/Kohonen results window contains two parts. The left side displays the grid. The colors of the rectangles in the grid indicate the number of cases in each cluster. Clusters with lighter colors have lower frequency counts. Clusters with darker colors have higher frequency counts.



The plot on the right shows the normalized means for each input variable. The means are normalized using a scale transformation function. You may need to maximize or resize the window to see the complete plot.

Note that there are three variables associated with the variable CLIMATE. In general, the SOM/Kohonen node constructs  $n$  dummy variables for a categorical variable with  $n$  levels.

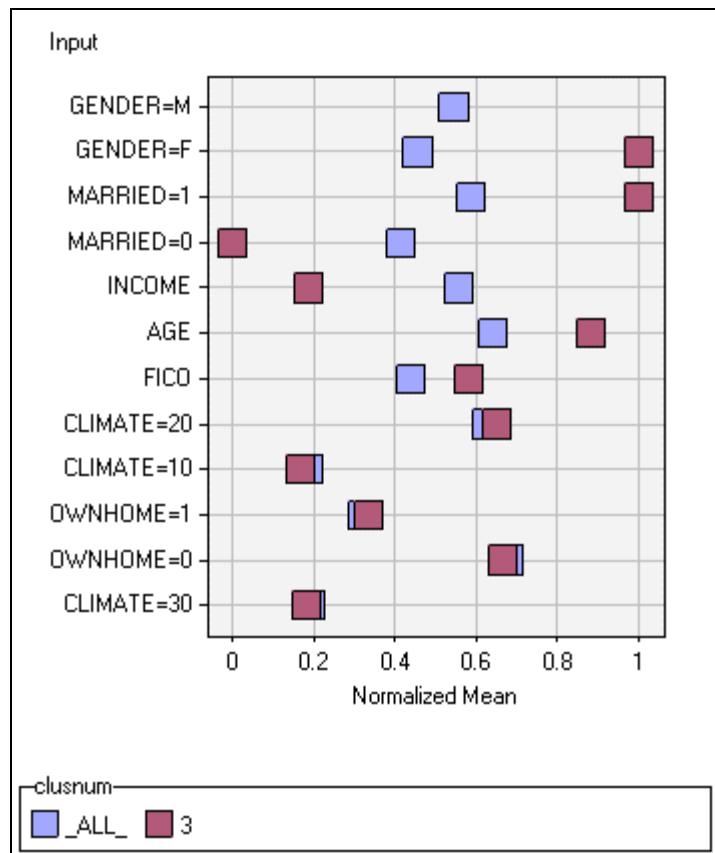
Initially, the overall normalized means are plotted for each input; however, if the window is not large enough or if there are many input variables you will not be able to see the entire plot. In that case, you can use the scroll icon, from the toolbar and scroll to view the others.

The Normalized Mean plot can be used to compare the overall normalized means with the normalized means in each cluster.

1. Select the Select Points icon, from the toolbar.
2. Select the section for row 1, column 3 in the map. The section turns gray, indicating it has been selected.

3. Select the Refresh Input Means Plot icon, , from the toolbar.

Inspect the Normalized Mean plot.



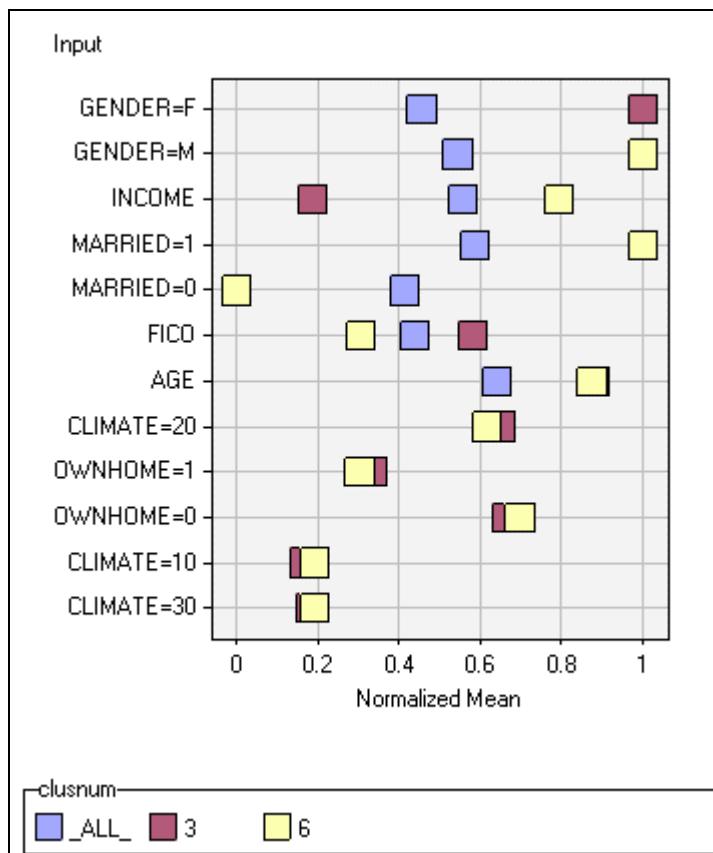
Note that cluster three

- has lower than average incomes
- consists of married females
- has higher than average ages
- has slightly higher than average credit scores.

The other clusters can be compared with the overall average by repeating these steps. For example, inspect the normalized mean plot for row 2, column 3.

4. Select the Select Points icon, , from the toolbar.
  5. Control-click to select the section for row 2, column 3 in the grid. The section turns gray, indicating it has been selected.
  6. Select the Refresh Input Means Plot icon, , from the toolbar.
-  If you control-click to select the second cluster, both clusters will then be selected and the information for both clusters will appear on the plot as

shown here. If you click on another cluster without using the control key, the first cluster will be deselected and only the information for the new cluster will appear on the plot.



Observe that prospects in cluster 6

- are male
- are married
- have higher than average incomes.

7. Select the Variables tab.
8. Click on the column heading Importance to sort the variables in the order of their relative importance in determining the clusters.

Name	Importance	Measurement	Type	Label
FICO	0	interval	num	
CLIMATE	0	nominal	char	
INCOME	0.0622563118	interval	num	
OWNHOME	0.168311178	binary	num	
AGE	0.64804929	interval	num	
MARRIED	0.9812919094	binary	num	
GENDER	1	binary	char	

The variables GENDER, MARRIED, and AGE were the most important variables in determining the clusters. Recall that CLIMATE, INCOME, and MARRIED were the important variables in the *k*-means cluster analysis.

9. Select the Statistics tab to examine the descriptive statistics for these clusters. Scroll to the right to see how the rows and columns of the map relate to the cluster number.

_SEGMENT_	Row	Column	SOM ID	AGE	INCOME	FICO	GENDER=F	GENDER=M	M
1	1	1 1:1	33.265553869	45.403641882	695.90256474		1	5.927875E-16	**
2	1	2 1:2	50.052959502	41.866043614	694.4417139		1	5.927875E-16	1
3	1	3 1:3	48.262084592	43.844410876	695.13094447		1	-1.93495E-14	**
4	2	1 2:1	33.667481663	52.381418093	690.35439496	-3.04688E-16		1	**
5	2	2 2:2	50.291208791	49.678571429	698.5467033	-3.04688E-16		1	1
6	2	3 2:3	48.16137366	50.219651757	692.8504592	-3.04688E-16		1	**

The clusters are numbered beginning with the bottom left cluster on the map, across the bottom row from left to right, and then up to the next row and across it from left to right.

Examining the statistics reveals that the

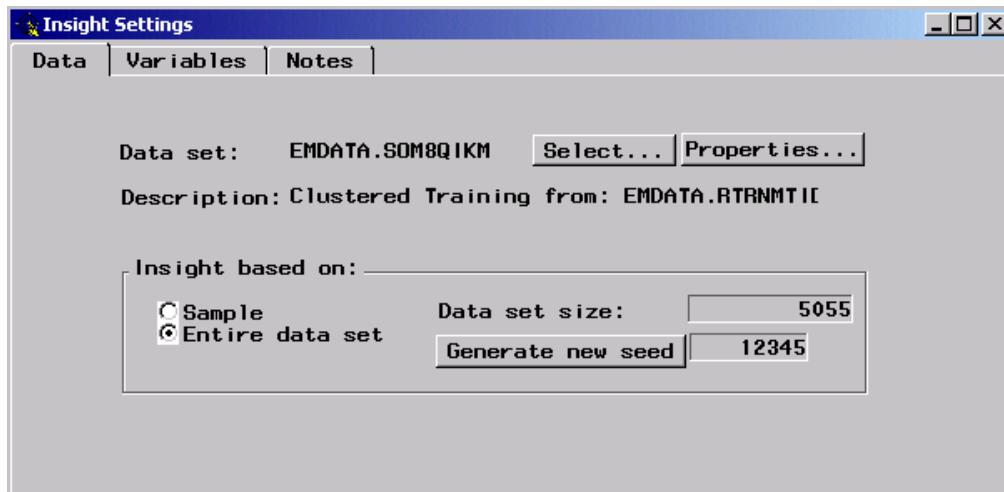
- average age of the prospects in clusters in the first column of the map is younger than the other clusters
- clusters in the first row of the map consist of women while the clusters in the second row consist of men
- clusters in the first two columns of the map are primarily unmarried individuals.

- Close the SOM/Kohonen node after inspecting these results.

### Exploring the Results Using Insight

Use the Insight node to compare the distribution of the categorical and interval inputs among the clusters in the SOM/Kohonen node.

- Open the Insight node and confirm that the option to use the Entire data set is still selected.
- Select Select... to choose the data set associated with the SOM/Kohonen node.



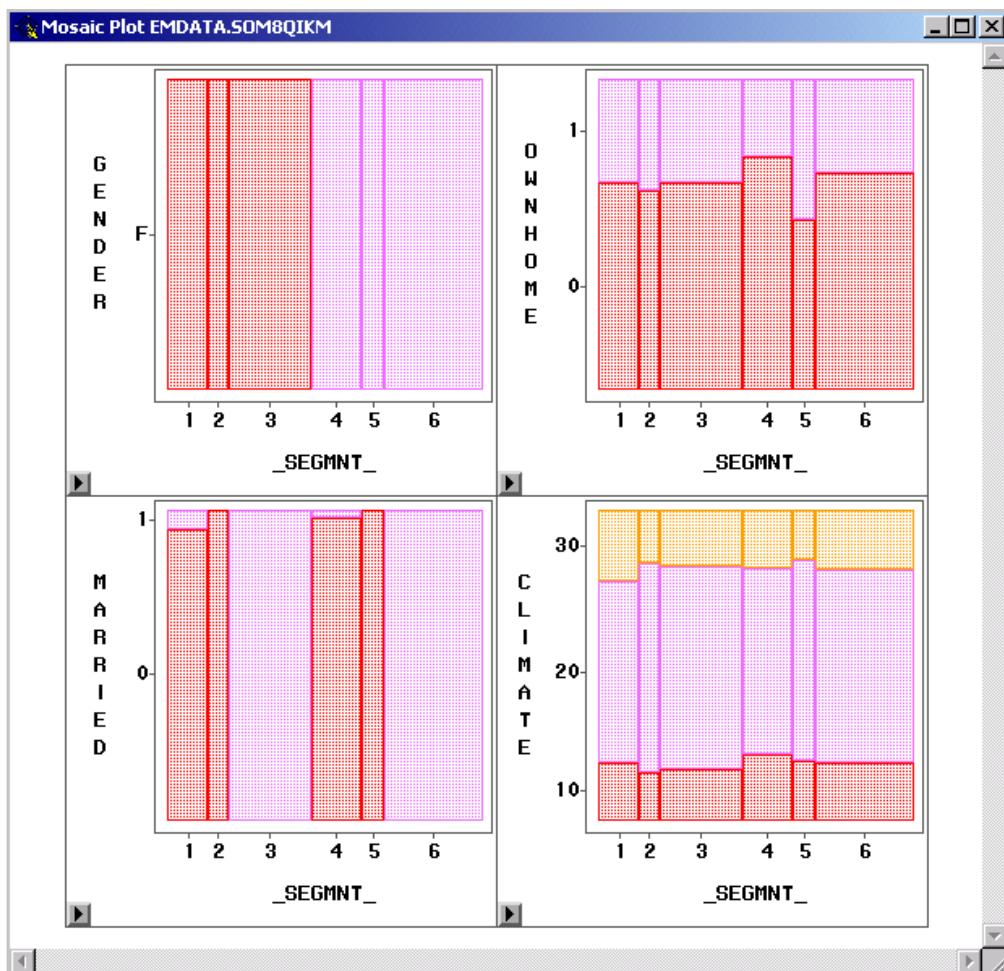
- Close the Insight node and save the changes when prompted.
- Run the flow from the Insight node and view the results. Scroll to the right in the data table to see the new columns that have been added to the original data set.

	14	Nom	Int	Int	Int	Int	Nom	
	CLIMATE	_SEGMENT_	Distance	Row	Column	Int	SOM_ID	
■	1	30	6	1.6378	2	3	2:3	
■	2	20	6	1.3479	2	3	2:3	
■	3	20	1	1.1478	1	1	1:1	
■	4	20	4	2.5321	2	1	2:1	
■	5	20	6	1.0451	2	3	2:3	
■	6	30	6	2.0447	2	3	2:3	
■	7	20	5	2.2724	2	2	2:2	
■	8	20	2	1.8107	1	2	1:2	
■	9	20	2	3.1497	1	2	1:2	
■	10	20	3	1.6115	1	3	1:3	
■	11	10	6	3.3237	2	3	2:3	
■	12	20	5	2.2772	2	2	2:2	
■	13	10	3	2.4605	1	3	1:3	
■	14	10	6	1.6991	2	3	2:3	
■	15	10	1	2.4834	1	1	1:1	
■	16	20	5	1.4536	2	2	2:2	
■	17	20	1	1.3121	1	1	1:1	

- Change the measurement scale for MARRIED, OWNHOME, and \_SEGMENT\_ to nominal by changing the measurement scale directly above the variable name.
- Select Analyze  $\Rightarrow$  Box Plot/Mosaic Plot (Y).

7. Select SEGMENT  $\Rightarrow$  X.
8. Select GENDER.
9. Press and hold the Ctrl key.
10. Highlight MARRIED, OWNHOME, and CLIMATE.
11. Select Y.
12. Select OK.

Examine the resulting mosaic plots.

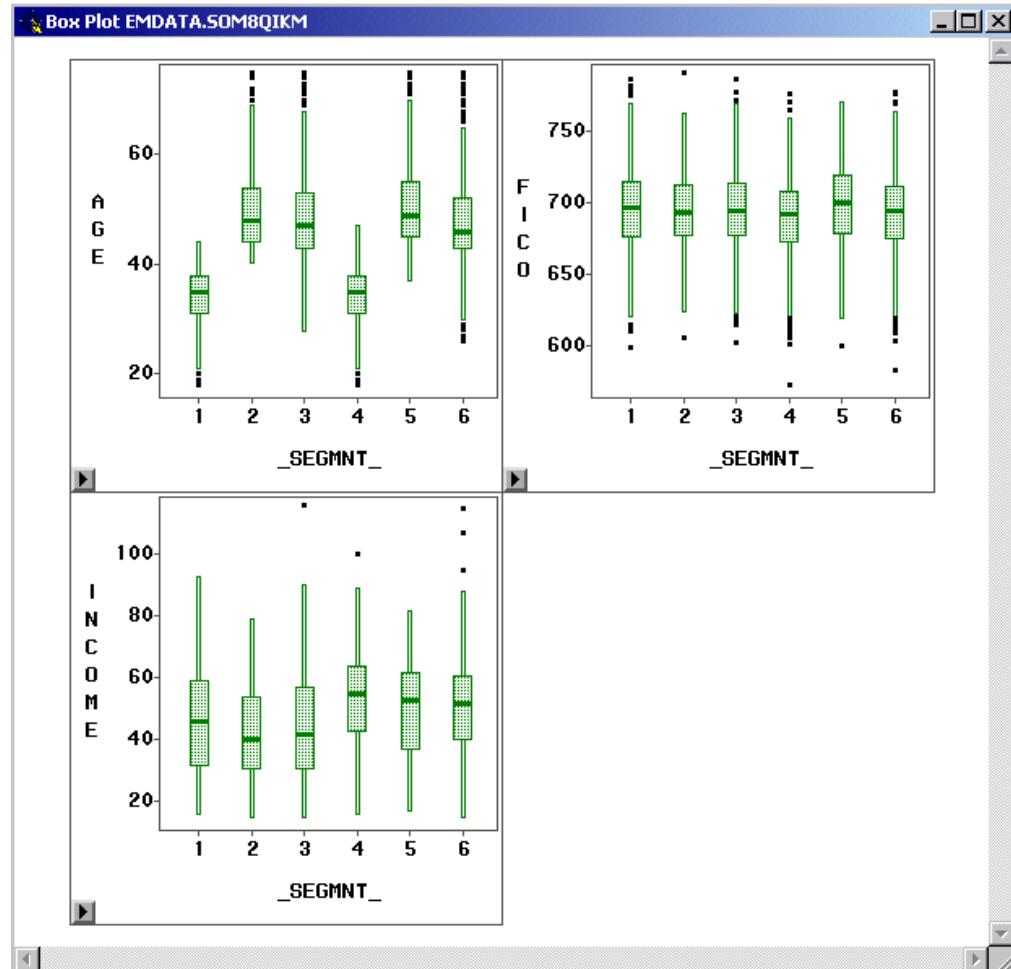


The clusters appear to be generally equivalent with respect to location and homeownership with the separation clearly coming from gender and marital status.

Now compare the distributions of the interval inputs to understand the cluster differences.

1. Select Analyze  $\Rightarrow$  Box Plot/Mosaic Plot (Y).
2. Select SEGMENT  $\Rightarrow$  X.

3. Select **AGE**.
4. Press and hold the control key.
5. Select **INCOME** and **FICO**, and then select **Y**.
6. Select **OK**.



The box plots confirm that clusters 1 and 4 have younger prospects than the other clusters. FICO and INCOME do not appear to differ among clusters.

7. Close the plot and data table windows to return to the Enterprise Miner workspace.

## Self-Organizing Map Results

younger, unmarried males	unmarried males	married males
younger, unmarried females	unmarried females	married females

19

To summarize the attributes of the six clusters:

1. Map(1,1) – younger, unmarried females
2. Map(1,2) –unmarried females
3. Map(1,3) – married females
4. Map(2,1) – younger, unmarried males
5. Map(2,2) – unmarried males
6. Map(2,3) – married males.



# Chapter 8 Association and Sequence Analysis

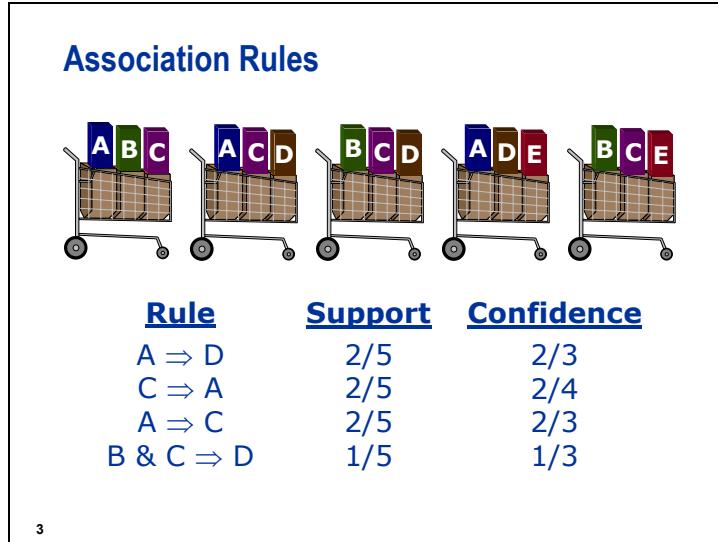
8.1	Introduction to Association Analysis .....	8-3
8.2	Interpretation of Association and Sequence Analysis.....	8-7
8.3	Dissociation Analysis (Self-Study) .....	8-24



## 8.1 Introduction to Association Analysis

### Objectives

- Define an association rule.
- Define support, confidence, and lift.
- Discuss difficulties in obtaining or acting upon results.



*Association rule discovery* (aka market-basket analysis, affinity analysis) is a popular data mining method. In the simplest situation, the data consists of two variables: a *transaction* and an *item*.

For each transaction, there is a list of items. Typically, a transaction is a single customer purchase and the items are the things that were bought. An *association rule* is a statement of the form (item set  $A$ )  $\Rightarrow$  (item set  $B$ ).

The aim of the analysis is to determine the strength of all the association rules among a set of items.

The strength of the association is measured by the *support* and *confidence* of the rule. The support for the rule  $A \Rightarrow B$  is the probability that the two item sets occur together. The support of the rule  $A \Rightarrow B$  is estimated by

$$\frac{\text{transactions that contain every item in } A \text{ and } B}{\text{all transactions}}.$$

Note that support is reflexive. That is, the support of the rule  $A \Rightarrow B$  is the same as the support of the rule  $B \Rightarrow A$ .

The confidence of an association rule  $A \Rightarrow B$  is the conditional probability of a transaction containing item set  $B$  given that it contains item set  $A$ . The confidence is estimated by

$$\frac{\text{transactions that contain every item in } A \text{ and } B}{\text{transactions that contain the items in } A}.$$

		Checking Account		
		No	Yes	
Saving Account	No	500	3,500	4,000
	Yes	1,000	5,000	6,000
		Support(SVG $\Rightarrow$ CK) = 50%		10,000
		Confidence(SVG $\Rightarrow$ CK) = 83%		
		Lift(SVG $\Rightarrow$ CK) = 0.83/0.85 < 1		

4

The interpretation of the implication ( $\Rightarrow$ ) in association rules is precarious. High confidence and support does not imply cause and effect. The rule is not necessarily interesting. The two items might not even be correlated. The term *confidence* is not related to the statistical usage; therefore, there is no repeated sampling interpretation.

Consider the association rule (saving account)  $\Rightarrow$  (checking account). This rule has 50% support (5,000/10,000) and 83% confidence (5,000/6,000). Based on these two measures, this might be considered a strong rule. On the contrary, those **without** a savings account are even more likely to have a checking account (87.5%). Saving and checking are in fact negatively correlated.

If the two accounts were independent, then knowing that one has a saving account does not help in knowing whether one has a checking account. The expected confidence if the two accounts were independent is 85% (8,500/10,000). This is higher than the confidence of SVG  $\Rightarrow$  CK.

The *lift* of the rule  $A \Rightarrow B$  is the confidence of the rule divided by the expected confidence, assuming the item sets are independent. The lift can be interpreted as a general measure of association between the two item sets. Values greater than 1 indicate positive correlation; values equal to 1 indicate zero correlation; and values less than 1 indicate negative correlation. Note that lift is reflexive. That is, the lift of the rule  $A \Rightarrow B$  is the same as the lift of the rule  $B \Rightarrow A$ .

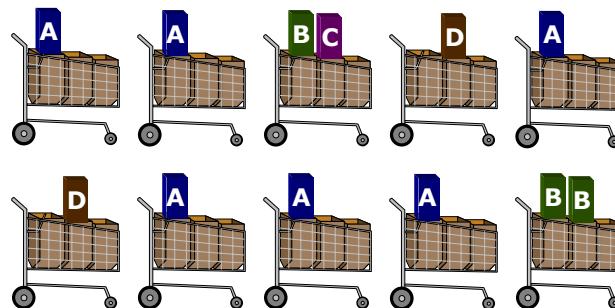
### Barbie® $\Rightarrow$ Candy

1. Put them closer together in the store.
2. Put them far apart in the store.
3. Package candy bars with the dolls.
4. Package Barbie + candy + poorly selling item.
5. Raise the price on one, lower it on the other.
6. Barbie accessories for proofs of purchase.
7. Do not advertise candy and Barbie together.
8. Offer candies in the shape of a Barbie Doll.

5

*Forbes* (Palmeri 1997) reported that a major retailer has determined that customers who buy Barbie dolls have a 60% likelihood of buying one of three types of candy bars. The confidence of the rule  $\text{Barbie} \Rightarrow \text{candy}$  is 60%. The retailer was unsure what to do with this nugget. The online newsletter *Knowledge Discovery Nuggets* invited suggestions (Piatesky-Shapiro 1998).

### Data Capacity



7

In data mining, the data is not generated to meet the objectives of the analysis. It must be determined whether the data, as it exists, has the capacity to meet the objectives. For example, quantifying affinities among related items would be pointless if very few transactions involved multiple items. Therefore, it is important to do some initial examination of the data before attempting to do association analysis.

## 8.2 Interpretation of Association and Sequence Analysis

### Objectives

- Conduct an association analysis and interpret the results.
- Distinguish between association analysis and sequence analysis.
- Conduct a sequence analysis and interpret the results.

## Scenario – Banking Services

- ATM
- Automobile Loan
- Credit Card
- Certificate of Deposit
- Check/Debit Card
- Checking Account
- Home Equity Line of Credit
- Individual Retirement Account
- Money Market Deposit Account
- Mortgage
- Personal/Consumer Installment Loan
- Savings Account
- Personal Trust Account

10

A bank wants to examine its customer base and understand which of its products individual customers own in combination with one another. It has chosen to conduct a market-basket analysis of a sample of its customer base. The bank has a data set that lists the banking products/services used by 7,991 customers. Thirteen possible products are represented as shown above.

There are three columns in the data set.

Name	Model Role	Measurement Level	Description
ACCT	ID	Nominal	Account Number
SERVICE	Target	Nominal	Type of Service
VISIT	Sequence	Ordinal	Order of Product Purchase



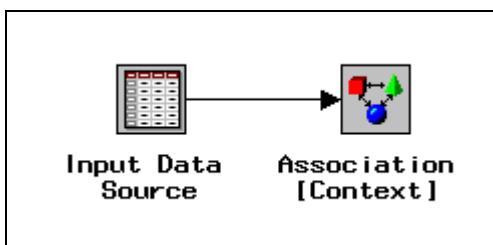
## Association Analysis

The BANK data set has over 32,000 rows. Each row of the data set represents a customer-service combination. Therefore, a single customer can have multiple rows in the data set, each row representing one of the products he or she owns. The median number of products per customer is three.

The 13 products are represented in the data set using the following abbreviations:

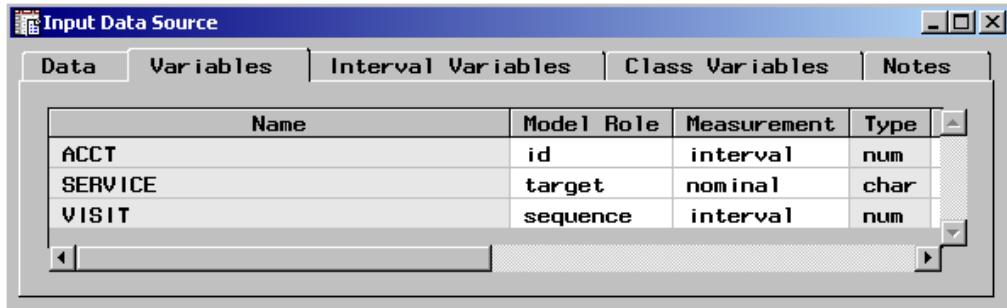
ATM	automated teller machine debit card
AUTO	automobile installment loan
CCRD	credit card
CD	certificate of deposit
CKCRD	check/debit card
CKING	checking account
HMEQLC	home equity line of credit
IRA	individual retirement account
MMDA	money market deposit account
MTG	mortgage
PLOAN	personal/consumer installment loan
SVG	saving account
TRUST	personal trust account

1. To open a new diagram workspace, select **File**  $\Rightarrow$  **New**  $\Rightarrow$  **Diagram**.
2. Name the diagram **Associations**.
3. Add nodes to the workspace as shown below.

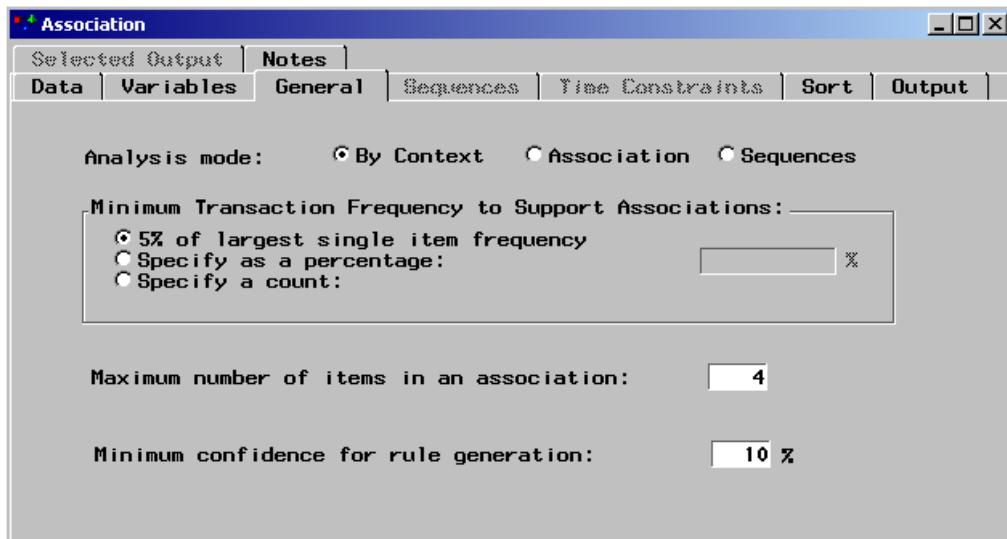


4. Open the Input Data Source node.
5. Select the **BANK** data set from the CRSSAMP library.

6. Select the **Variables** tab.
7. Set the model role for ACCT to **id**, for SERVICE to **target**, and for VISIT to **sequence**.



8. Close and save changes to the Input Data Source node.
9. Open the Association node. The Variables tab is active by default and lists the same information that is found in the Variables tab in the Input Data Source node.
10. Select the **General** tab. This tab enables you to modify the analysis mode and control how many rules are generated.



Inspect the Analysis mode options.

The default analysis mode is By Context. This mode uses information specified in the input data source to determine the appropriate analysis. If the input data set contains

- an ID variable and a target variable, the node automatically performs an association analysis
- a sequence variable that has a status of **use**, the node performs a sequence analysis.

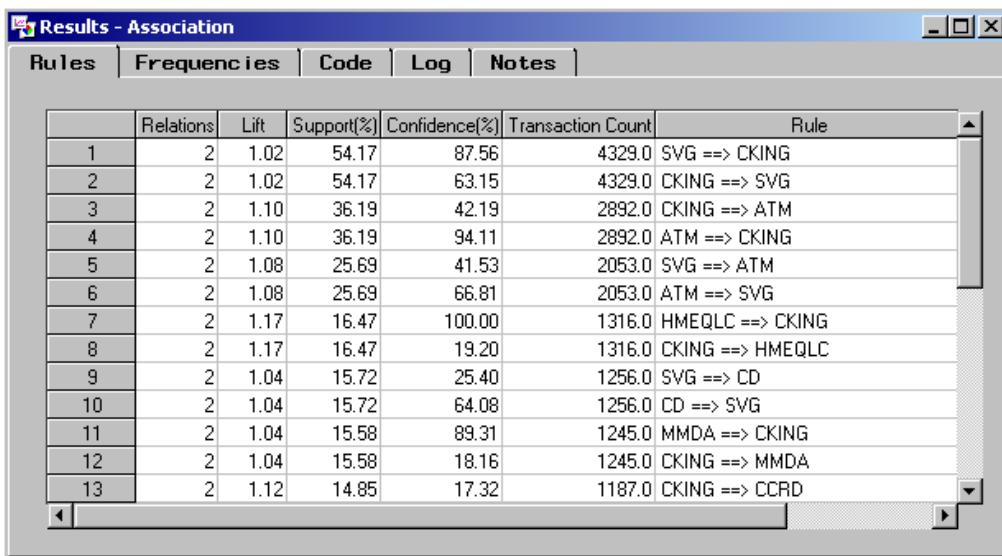
Because there is a sequence variable in the input data set, the default analysis mode would be a sequence analysis. Sequence analysis is discussed later in this section, but for the moment, perform an association analysis.

11. Change the Analysis mode to **Association**.
12. Close the Association node, saving changes when prompted.

Other options include

- Minimum Transaction Frequency to Support Associations - specifies a minimum level of support to claim that items are associated (that is, they occur together in the database). The default frequency is 5%.
- Maximum number of items in an association - determines the maximum size of the item set to be considered. For example, the default of four items indicates that a maximum of 4 items will be included in a single association rule.
- Minimum confidence for rule generation - specifies the minimum confidence level to generate a rule. The default level is 10%. This option is grayed out if you are performing a sequence discovery.

13. Run the diagram from the Association node and view the results. The Rules tab is displayed first.



The screenshot shows a software interface titled "Results - Association". The window has a menu bar at the top with icons for File, Edit, View, Insert, Tools, Help, and a zoom control. Below the menu is a toolbar with icons for New, Open, Save, Print, and others. The main area is a table titled "Rules" with the following columns: Relations, Lift, Support(%), Confidence(%), Transaction Count, and Rule. The table contains 13 rows of data. The last row is partially visible.

	Relations	Lift	Support(%)	Confidence(%)	Transaction Count	Rule
1	2	1.02	54.17	87.56	4329.0	SVG ==> CKING
2	2	1.02	54.17	63.15	4329.0	CKING ==> SVG
3	2	1.10	36.19	42.19	2892.0	CKING ==> ATM
4	2	1.10	36.19	94.11	2892.0	ATM ==> CKING
5	2	1.08	25.69	41.53	2053.0	SVG ==> ATM
6	2	1.08	25.69	66.81	2053.0	ATM ==> SVG
7	2	1.17	16.47	100.00	1316.0	HMEQLC ==> CKING
8	2	1.17	16.47	19.20	1316.0	CKING ==> HMEQLC
9	2	1.04	15.72	25.40	1256.0	SVG ==> CD
10	2	1.04	15.72	64.08	1256.0	CD ==> SVG
11	2	1.04	15.58	89.31	1245.0	MMDA ==> CKING
12	2	1.04	15.58	18.16	1245.0	CKING ==> MMDA
13	2	1.12	14.85	17.32	1187.0	CKING ==> CCRD

The Rules tab contains information for each rule. Consider the rule  $A \Rightarrow B$ . Recall that the

- support of  $A \Rightarrow B$  is the probability that a customer has both A and B
- confidence of  $A \Rightarrow B$  is the probability that a customer has B given that the customer has A.
- lift of  $A \Rightarrow B$  is a measure of strength of the association. If the Lift=2 for the rule  $A \Rightarrow B$ , then a customer having A is twice as likely to have B than a customer chosen at random.

14. Right-click on the Support(%) column and select Sort  $\Rightarrow$  Descending.

The screenshot shows a software interface titled "Results - Association". At the top, there are tabs: Rules, Frequencies, Code, Log, and Notes. The "Frequencies" tab is selected. Below the tabs is a table with the following columns: Relations, Lift, Support(%), Confidence(%), Transaction Count, and Rule. The table contains 13 rows of data. The "Support(%)" column is sorted in descending order, with values ranging from 16.47% to 54.17%.

	Relations	Lift	Support(%)	Confidence(%)	Transaction Count	Rule
1	2	1.02	54.17	63.15	4329.0	CKING ==> SVG
2	2	1.02	54.17	87.56	4329.0	SVG ==> CKING
3	2	1.10	36.19	94.11	2892.0	ATM ==> CKING
4	2	1.10	36.19	42.19	2892.0	CKING ==> ATM
5	2	1.08	25.69	66.81	2053.0	ATM ==> SVG
6	2	1.08	25.69	41.53	2053.0	SVG ==> ATM
7	3	1.19	24.85	45.88	1986.0	SVG & CKING ==> ATM
8	3	1.19	24.85	64.63	1986.0	ATM ==> SVG & CKING
9	3	1.11	24.85	68.67	1986.0	CKING & ATM ==> SVG
10	3	1.13	24.85	96.74	1986.0	SVG & ATM ==> CKING
11	3	1.13	24.85	28.97	1986.0	CKING ==> SVG & ATM
12	3	1.11	24.85	40.17	1986.0	SVG ==> CKING & ATM
13	2	1.17	16.47	19.20	1316.0	CKING ==> HMEQLC

The support is the percentage of customers who have all the services involved in the rule. For example, approximately 54% of the 7,991 customers have a checking and savings account and approximately 25% have a checking account, savings account, and an ATM card.

15. Right-click on the Confidence(%) column and select Sort  $\Rightarrow$  Descending.

The screenshot shows a software interface titled "Results - Association". At the top, there are tabs: Rules, Frequencies, Code, Log, and Notes. The "Frequencies" tab is selected. Below the tabs is a table with the following columns: Relations, Lift, Support(%), Confidence(%), Transaction Count, and Rule. The table contains 13 rows of data. The "Confidence(%)" column is sorted in descending order, with values ranging from 4.84% to 100.00%.

	Relations	Lift	Support(%)	Confidence(%)	Transaction Count	Rule
1	3	1.17	8.53	100.00	682.00	HMEQLC & ATM ==> CKING
2	2	1.17	11.30	100.00	903.00	CKCRD ==> CKING
3	2	1.17	16.47	100.00	1316.0	HMEQLC ==> CKING
4	3	1.17	7.97	100.00	637.00	SVG & CKCRD ==> CKING
5	3	1.17	11.15	100.00	891.00	SVG & HMEQLC ==> CKING
6	4	1.17	6.09	100.00	487.00	SVG & HMEQLC & ATM ==> CKING
7	3	1.17	4.63	100.00	370.00	HMEQLC & CCRD ==> CKING
8	3	1.17	5.58	100.00	446.00	CKCRD & CCRD ==> CKING
9	3	1.14	7.01	97.90	560.00	CD & ATM ==> CKING
10	2	1.14	7.27	97.81	581.00	MTG ==> CKING
11	3	1.14	9.99	97.67	798.00	SVG & CCRD ==> CKING
12	4	1.14	5.26	97.67	420.00	SVG & CD & ATM ==> CKING
13	3	1.14	4.84	97.48	387.00	CCRD & ATM ==> CKING

The confidence represents the percentage of customers who have the right-hand-side (RHS) item among those who have the left-hand-side (LHS) item. For example, all customers who have a checking account also have a check card. Among those customers that have both a savings account and a credit card, over 97% have a checking account.

16. Right-click on the Lift column and select **Sort  $\Rightarrow$  Descending**.

	Relations	Lift	Support(%)	Confidence(%)	Transaction Count	Rule
1	3	3.33	5.58	49.39	446.00	CKCRD ==> CKING & CCRD
2	3	3.33	5.58	37.57	446.00	CKING & CCRD ==> CKCRD
3	3	3.19	5.58	36.05	446.00	CCRD ==> CKING & CKCRD
4	3	3.19	5.58	49.39	446.00	CKING & CKRD ==> CCRD
5	2	3.19	5.58	49.39	446.00	CKCRD ==> CCRD
6	2	3.19	5.58	36.05	446.00	CCRD ==> CKCRD
7	3	1.89	4.63	31.17	370.00	CKING & CCRD ==> HMEQLC
8	3	1.89	4.63	28.12	370.00	HMEQLC ==> CKING & CCRD
9	3	1.82	4.63	28.12	370.00	HMEQLC & CKING ==> CCRD
10	2	1.82	4.63	28.12	370.00	HMEQLC ==> CCRD
11	2	1.82	4.63	29.91	370.00	CCRD ==> HMEQLC
12	3	1.82	4.63	29.91	370.00	CCRD ==> HMEQLC & CKING
13	4	1.51	6.09	16.84	487.00	CKING & ATM ==> SVG & HMEQLC

Lift, in the context of association rules, is the ratio of the confidence of a rule to the confidence of a rule assuming the RHS was independent of the LHS. Consequently, lift is a measure of association between the LHS and RHS of the rule. Values greater than 1 represent positive correlation between the LHS and RHS. Values equal to 1 represent independence. Values less than 1 represent negative correlation between the LHS and RHS.

The lift of the relationship CKRD ==> CCRD is 3.19. Therefore, if you select a customer who has a check/debit card, the relative frequency of that customer having a credit card is more than 3 times higher than an individual chosen at random.



By default, only rules with a lift greater than 1 are displayed in the results. You can change this by selecting **View  $\Rightarrow$  When Confidence > Expected Confidence**.

17. Select the **Frequencies** tab.

The screenshot shows a software window titled "Results - Association". The window has a menu bar with "File", "Edit", "View", "Rules", "Frequencies" (which is highlighted in blue), "Code", "Log", and "Notes". Below the menu is a toolbar with icons for "New", "Open", "Save", "Print", "Exit", "Copy", "Paste", "Delete", "Find", "Replace", "Select All", "Clear", "Help", and "About". The main area contains a table with 12 rows, each representing an item and its count. The columns are labeled "Count" and "Item". The table data is as follows:

	Count	Item
1	6855	CKING
2	4944	SVG
3	3073	ATM
4	1960	CD
5	1394	MMDA
6	1316	HMEQLC
7	1237	CCRD
8	903	CKCRD
9	866	IRA
10	742	AUTO
11	594	MTG
12	390	TRUST

This tab shows the number of customers who have each of the products. This number is not the same as a simple frequency count. For example, a customer may have more than one checking account but is counted only once here.

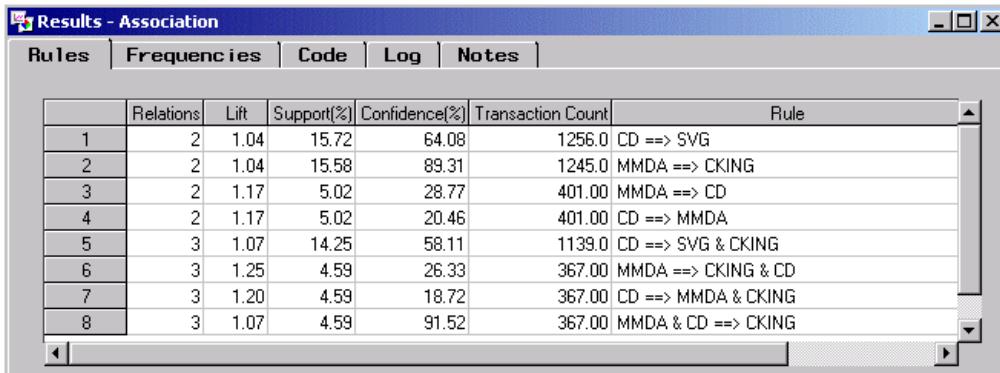
This information can be important when trying to determine why a particular item did not appear in the rules. Recall that, by default, the minimum transaction frequency to support an association is 5% of the largest frequency item. In this case the largest frequency item is a checking account with a frequency of 6855. Therefore, if any group of products is not held by at least 343 customers (5% of 6855), they are not included together in a rule.

If you are interested in associations involving fairly rare products, you should consider reducing the minimum transaction frequency when you run the association node. If you obtain too many rules to be practically useful, you should consider raising the minimum transaction frequency as one possible solution.

Suppose you are particularly interested in looking at customers who have a money market deposit account (MMDA) and/or a certificate of deposit (CD) to determine what other types of products they also own.

1. Select the **Rules** tab.
2. Select **View**  $\Rightarrow$  **Subset Table...**
3. In the Predecessor  $\Rightarrow$  Successor tab, control-click to select both **CD** and **MMDA** under Left Hand Side.
4. In the drop-down menu for Type under Left Hand Side, select **Combinations & Single**.

5. In the drop-down menu for Type under Right Hand Side, select **Find Any**.
6. Select **Process** and examine the resulting table.



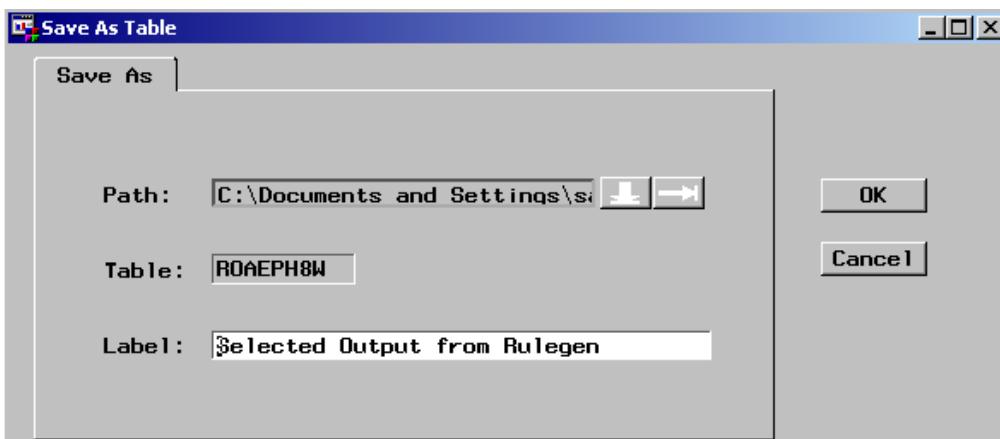
The screenshot shows the SAS Results - Association window. The title bar says "Results - Association". Below it is a menu bar with "Rules", "Frequencies", "Code", "Log", and "Notes". The main area is a table with the following data:

	Relations	Lift	Support(%)	Confidence(%)	Transaction Count	Rule
1	2	1.04	15.72	64.08	1256.0	CD => SVG
2	2	1.04	15.58	89.31	1245.0	MMDA => CKING
3	2	1.17	5.02	28.77	401.00	MMDA => CD
4	2	1.17	5.02	20.46	401.00	CD => MMDA
5	3	1.07	14.25	58.11	1139.0	CD => SVG & CKING
6	3	1.25	4.59	26.33	367.00	MMDA => CKING & CD
7	3	1.20	4.59	18.72	367.00	CD => MMDA & CKING
8	3	1.07	4.59	91.52	367.00	MMDA & CD => CKING

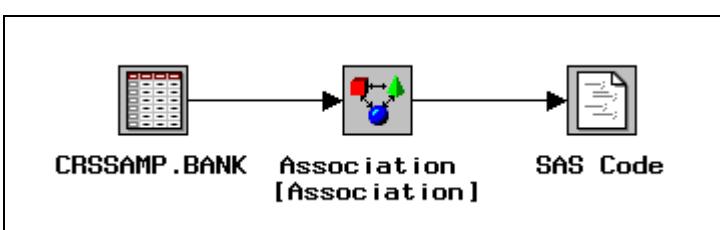
There are eight rules that contain MMDA, CD, or both on the left hand side.

Suppose instead that you are particularly interested in those associations that involve automobile loans. You would like to create a subset of the rules, to include only those rules with the product AUTO. The easiest way to accomplish this goal is to save the rules as a data set and then use a SAS Code node to subset that data.

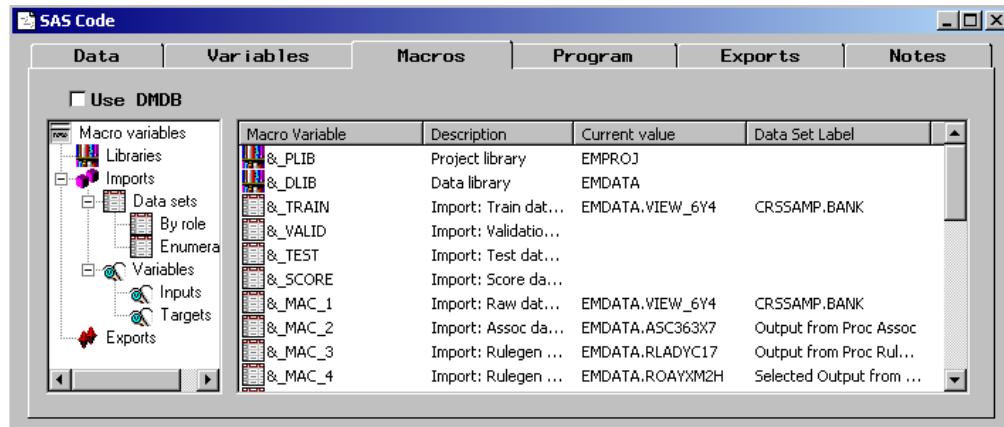
7. Select **View**  $\Rightarrow$  **Reset Table**.
8. Select **File**  $\Rightarrow$  **Save As Data Set...**



9. Note that the data set will be saved with the label Selected Output from Rulegen. Select **OK** to save the data set.
10. Close the Association node results window and add a SAS Code node to the diagram.



11. Open the **SAS Code** node and select the **Macros** tab.



Enterprise Miner automatically creates many macro variables. In this case, note the variable &\_MAC\_4, which is the data set you saved from the Association node results.

12. Select the **Program** tab.

13. Type in the following program:

```
data work.auto;
  set &_MAC_4;
  if item1='AUTO' or item2='AUTO' or item3='AUTO'
    or item4='AUTO' or item5='AUTO';
run;

proc print data=work.auto;
run;
```

14. Select the Run SAS Code button, , to submit the program.

15. Select **Yes** to run the SAS code now.

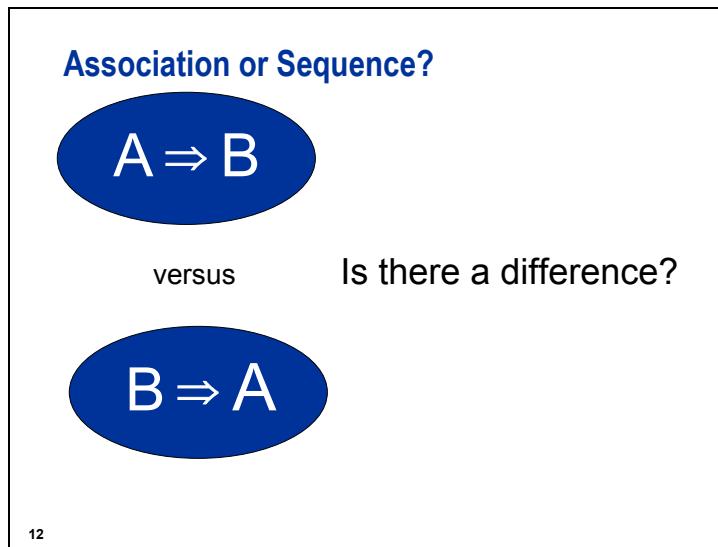
16. Select **Yes** to view the results, when prompted.

17. Select the **Output** tab.

Obs	SET_SIZE	EXP_CONF	CONF	SUPPORT	LIFT	COUNT	RULE	_LHAND
1	2	85.78	91.78	8.52	1.07	681.00	AUTO ==> CKING	AUTO
2	2	61.87	66.17	6.14	1.07	491.00	AUTO ==> SVG	AUTO
3	2	38.46	48.11	4.47	1.25	357.00	AUTO ==> ATM	AUTO
4	2	9.29	11.62	4.47	1.25	357.00	ATM ==> AUTO	ATM
5	3	54.17	64.29	5.97	1.19	477.00	AUTO ==> SVG & CKING	AUTO
6	3	9.29	11.02	5.97	1.19	477.00	SVG & CKING ==> AUTO	SVG & CKING
7	3	85.78	97.15	5.97	1.13	477.00	SVG & AUTO ==> CKING	SVG & AUTO
8	3	61.87	70.04	5.97	1.13	477.00	CKING & AUTO ==> SVG	CKING & AUTO
9	3	36.19	46.90	4.35	1.30	348.00	AUTO ==> CKING & ATM	AUTO
10	3	8.52	11.32	4.35	1.33	348.00	ATM ==> CKING & AUTO	ATM
11	3	38.46	51.10	4.35	1.33	348.00	CKING & AUTO ==> ATM	CKING & AUTO
12	3	9.29	12.03	4.35	1.30	348.00	CKING & ATM ==> AUTO	CKING & ATM
13	3	85.78	97.48	4.35	1.14	348.00	AUTO & ATM ==> CKING	AUTO & ATM
Obs	_RHAND	ITEM1	ITEM2	ITEM3	ITEM4	ITEM5		
1	CKING	AUTO	=====>	CKING				
2	SVG	AUTO	=====>	SVG				
3	ATM	AUTO	=====>	ATM				
4	AUTO	ATM	=====>	AUTO				
5	SVG & CKING	AUTO	=====>	SVG	CKING			
6	AUTO	SVG	CKING	=====>	AUTO			
7	CKING	SVG	AUTO	=====>	CKING			
8	SVG	CKING	AUTO	=====>	SVG			
9	CKING & ATM	AUTO	=====>	CKING	ATM			
10	CKING & AUTO	ATM	=====>	CKING	AUTO			
11	ATM	CKING	AUTO	=====>	ATM			
12	AUTO	CKING	ATM	=====>	AUTO			
13	CKING	AUTO	ATM	=====>	CKING			

There are 13 rules that involve automobile loans as shown in the output.

18. Close the SAS Code node results and the SAS code node when you are finished examining the output.



Association analysis is designed to determine the relationships between products offered for sale. In other words, what products are likely to appear together in a customer's basket?

Sequence analysis takes this a step further in that it examines the order in which products are purchased. This can be helpful in answering such questions as: if a customer purchased product A this week, is he likely to purchase product B next week?

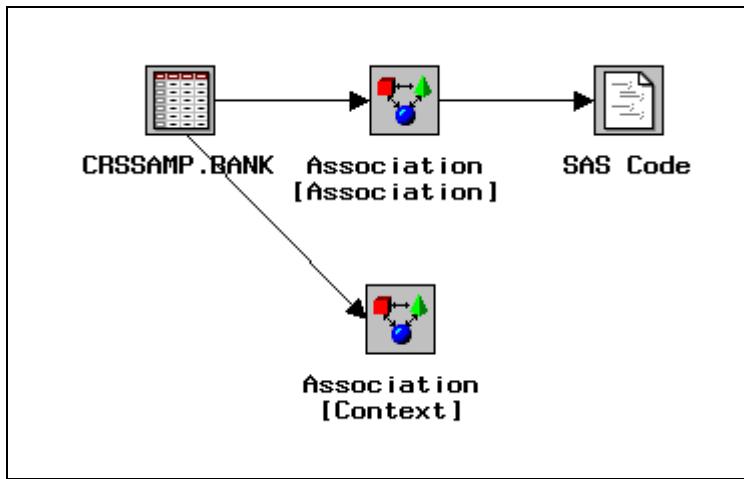
A sequence analysis requires the specification of a variable whose model role is sequence. An association analysis ignores a sequence variable.



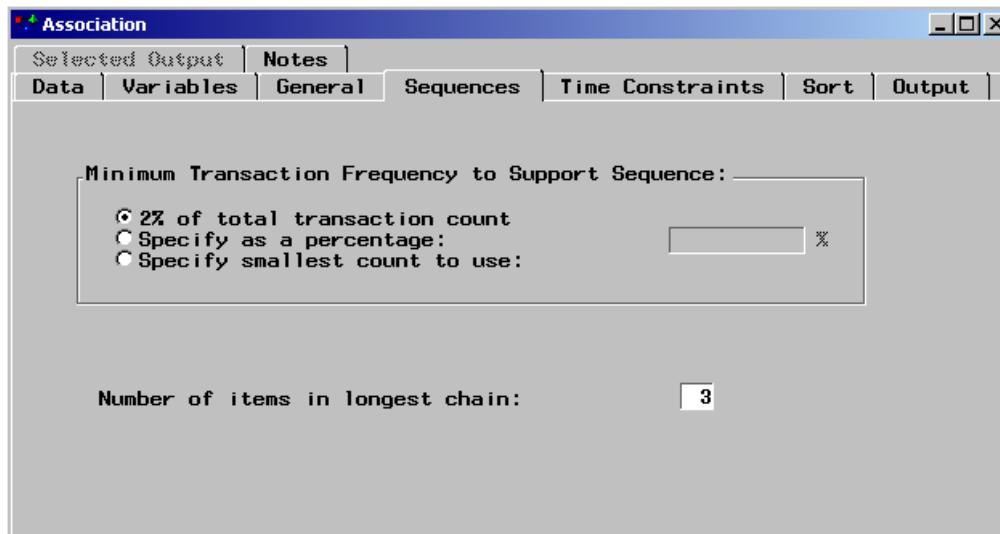
## Sequence Analysis

In addition to the products owned by its customers, the bank is interested in examining the order in which the products are purchased. The sequence variable in the data set allows you to conduct a sequence analysis.

1. Add an Association node to the diagram workspace and connect it to the Input Data Source node.

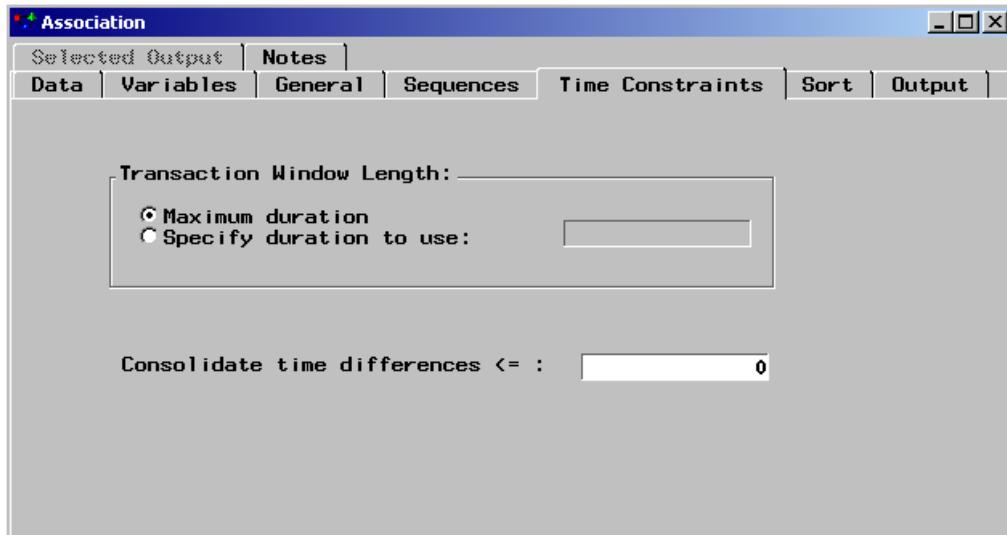


2. Open the new Association node.
3. Select the **General** tab. Note that because there is a sequence variable with a status of **use** in the input data set, by default, the analysis mode will be a sequence analysis.
4. Select the **Sequences** tab.



The options in the Sequences tab enable you to specify the minimum transaction frequency required to report a sequence and the total number of items that can be in the longest chain reported. The maximum number of items that can be specified is 10.

5. Select the **Time Constraints** tab.



The options here allow you to specify the maximum length of time for a series of transactions to be considered a sequence. For example, you might want to specify that the purchase of two products more than three months apart from each other does not constitute a sequence.

Another option is to consolidate time differences. In other words, two products purchased less than a day apart from each other are actually the same transaction.

6. Close the Association node, leaving the default settings.
7. Run the diagram from the Association node and select **Yes** to view the results when prompted.

	Chain Length	Support(%)	Confidence(%)	Transaction Count	Rule
1	2	54.17	63.15	4329	CKING ==> SVG
2	2	36.19	42.19	2892	CKING ==> ATM
3	2	25.69	41.53	2053	SVG ==> ATM
4	2	21.39	55.61	1709	ATM ==> ATM
5	2	20.99	24.46	1677	CKING ==> CD
6	2	16.47	19.20	1316	CKING ==> HMEQLC
7	2	15.72	25.40	1256	SVG ==> CD
8	2	15.58	18.16	1245	CKING ==> MMDA
9	2	14.85	17.32	1187	CKING ==> CCRD
10	2	13.58	21.95	1085	SVG ==> SVG
11	2	11.30	13.17	903	CKING ==> CKCRD
12	2	11.30	100.00	903	CKCRD ==> CKCRD
13	2	11.15	18.02	891	SVG ==> HMEQLC
14	2	10.22	16.53	817	SVG ==> CCRD
15	2	9.21	37.55	736	CD ==> CD
16	2	8.82	10.28	705	CKING ==> IRA
17	2	8.81	53.50	704	HMEQLC ==> HMEQLC
18	2	8.53	22.19	682	ATM ==> HMEQLC

8. Right-click on the Confidence(%) column and select Sort  $\Rightarrow$  Descending.

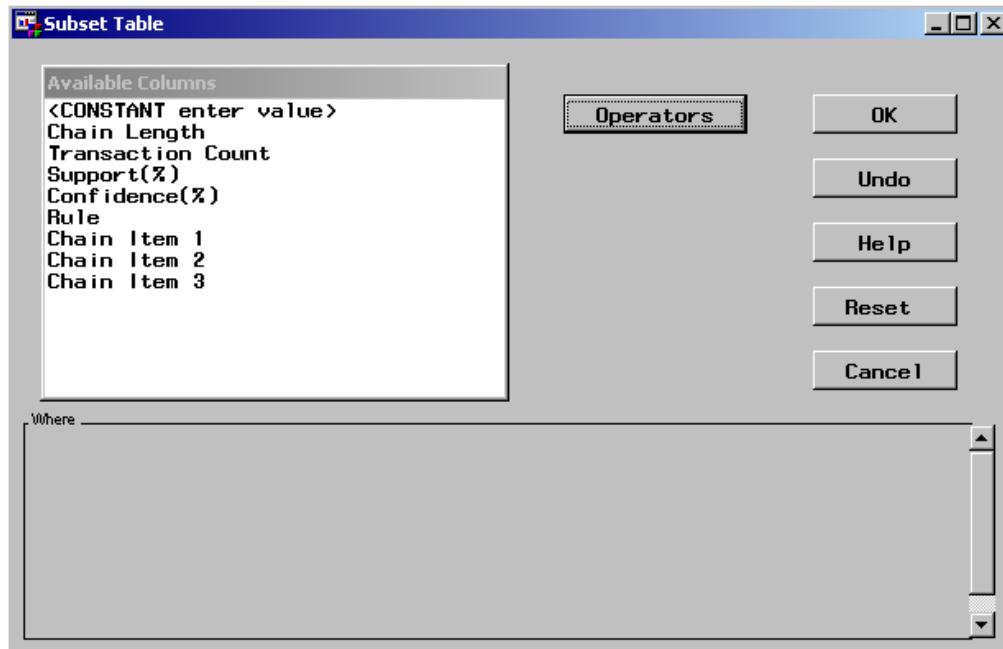
	Chain Length	Support(%)	Confidence(%)	Transaction Count	Rule
1	3	4.47	100.00	357	ATM ==> AUTO ==> AUTO
2	3	11.30	100.00	903	CKING ==> CKCRD ==> CKCRD
3	3	2.60	100.00	208	CD ==> CKCRD ==> CKCRD
4	3	3.45	100.00	276	HMEQLC ==> CKCRD ==> CKCRD
5	3	5.58	100.00	446	CCRD ==> CKCRD ==> CKCRD
6	2	11.30	100.00	903	CKCRD ==> CKCRD
7	3	2.63	100.00	210	MMDA ==> CKCRD ==> CKCRD
8	3	7.97	100.00	637	SVG ==> CKCRD ==> CKCRD
9	3	3.29	87.38	263	ATM ==> MTG ==> MTG
10	3	2.10	82.76	168	CD ==> AUTO ==> AUTO
11	2	7.63	82.21	610	AUTO ==> AUTO
12	3	6.87	80.62	549	CKING ==> AUTO ==> AUTO
13	3	4.87	79.23	389	SVG ==> AUTO ==> AUTO
14	2	3.85	78.97	308	TRUST ==> TRUST
15	3	2.52	78.82	201	CD ==> TRUST ==> TRUST
16	3	3.65	78.28	292	CKING ==> TRUST ==> TRUST
17	3	2.29	75.00	183	SVG ==> TRUST ==> TRUST
18	2	5.26	70.71	420	MTG ==> MTG

The *transaction count* is the total number of customers that have purchased products in this order. The percent *support* is the transaction count divided by the total number of customers, which would be the maximum transaction count. The percent *confidence* is the transaction count divided by the transaction count for the left side of the sequence (which can be determined by looking at the Frequencies tab). So, for

example, of the customers that got an automobile loan, 82.21% got a second automobile loan.

Suppose you only want to see those sequences that involve automobile loans.

1. Select View  $\Rightarrow$  Subset Table...



2. From the Available Columns list, select Chain Item 1.
3. From the Operators list, select EQ.
4. From the Available Columns list, select <LOOKUP distinct values>.
5. Select AUTO.
6. Select Operators  $\Rightarrow$  OR.
7. From the Available Columns list, select Chain Item 2.
8. From the Operators list, select EQ.
9. From the Available Columns list, select <LOOKUP distinct values>.
10. Select AUTO.
11. Select Operators  $\Rightarrow$  OR.
12. From the Available Columns list, select Chain Item 3.
13. From the Operators list, select EQ.
14. From the Available Columns list, select <LOOKUP distinct values>.
15. Select AUTO.
16. Select OK.

**Results - Sequence**

	Rules	Frequencies	Code	Log	Notes
1	Chain Length	Support(%)	Confidence(%)	Transaction Count	Rule
1	3	4.47	100.00	357	ATM ==> AUTO ==> AUTO
2	3	2.10	82.76	168	CD ==> AUTO ==> AUTO
3	2	7.63	82.21	610	AUTO ==> AUTO
4	3	6.87	80.62	549	CKING ==> AUTO ==> AUTO
5	3	4.87	79.23	389	SVG ==> AUTO ==> AUTO
6	2	2.53	15.35	202	HMEQLC ==> AUTO
7	3	2.53	15.35	202	CKING ==> HMEQLC ==> AUTO
8	3	2.18	14.66	174	CKING ==> CCRD ==> AUTO
9	2	2.21	14.31	177	CCRD ==> AUTO
10	3	3.30	12.86	264	SVG ==> ATM ==> AUTO
11	3	4.35	12.03	348	CKING ==> ATM ==> AUTO
12	2	4.47	11.62	357	ATM ==> AUTO
13	3	2.40	11.45	192	CKING ==> CD ==> AUTO
14	3	5.97	11.02	477	CKING ==> SVG ==> AUTO
15	2	2.54	10.36	203	CD ==> AUTO
16	2	8.52	9.93	681	CKING ==> AUTO
17	2	6.14	9.93	491	SVG ==> AUTO

Seventeen rules involve automobile loans as shown above.

17. Close the sequence analysis results when you are finished examining the output.

## 8.3 Dissociation Analysis (Self-Study)

### Objectives

- Define dissociation analysis.
- Generate a dissociation analysis within Enterprise Miner.
- Clone a node in Enterprise Miner.

15

### Dissociation Analysis

*Dissociation analysis* is used to determine what products do **not** appear together in market baskets.

16

A *dissociation rule* is a rule involving the negation of some item. For example, the left side may represent **no** checking account (~CKING) and the right side might be an auto loan. In other words, customers who do not have checking accounts tend to have auto loans. Dissociation rules may be particularly interesting when the items involved are highly prevalent.



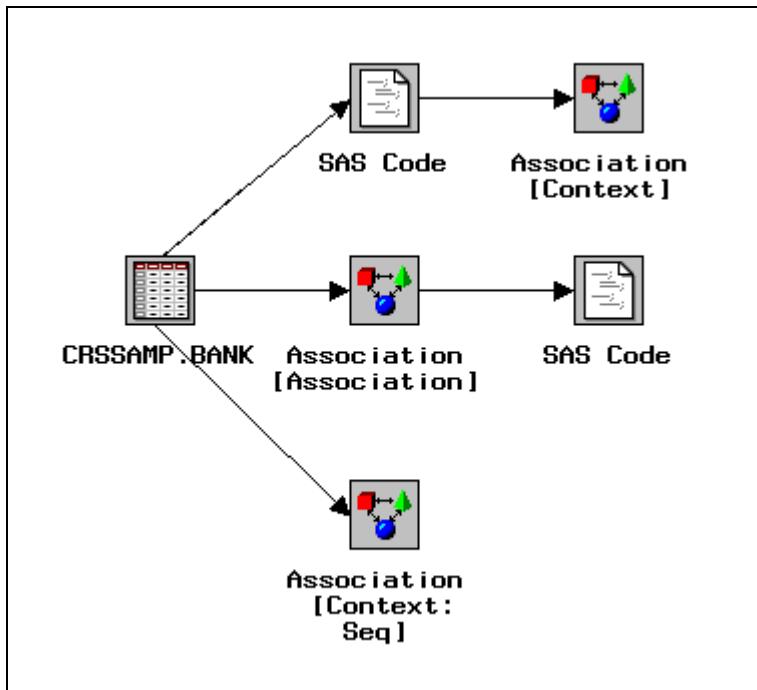
## Dissociation Analysis

The Association node will include dissociation rules if the data is modified to include the negation of selected items. The SAS Code node can be used for such data modification.

### Creating Dissociations

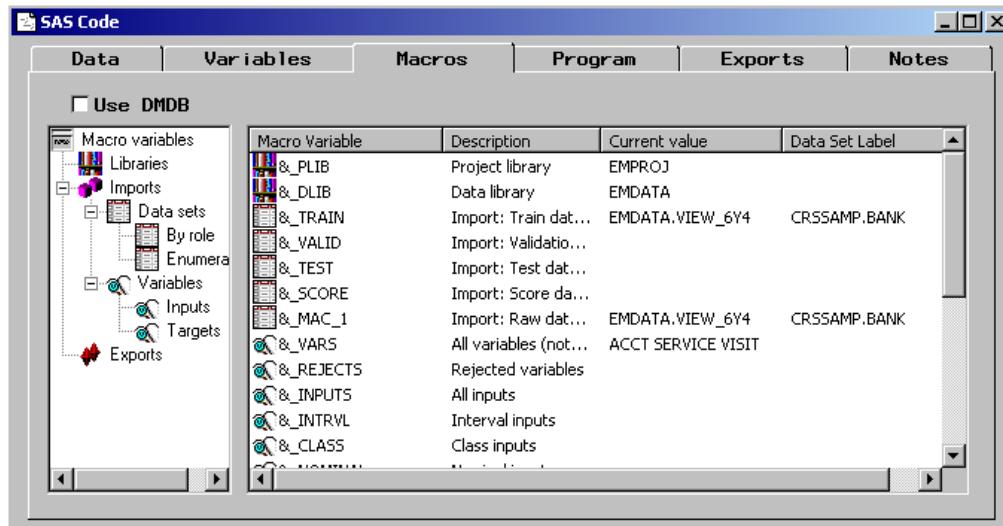
Augment the data with services not present in each account.

1. Add a SAS Code node to the workspace and connect it to the Input Data Source node.
2. Add another Association node to the diagram and connect it to the SAS Code node.

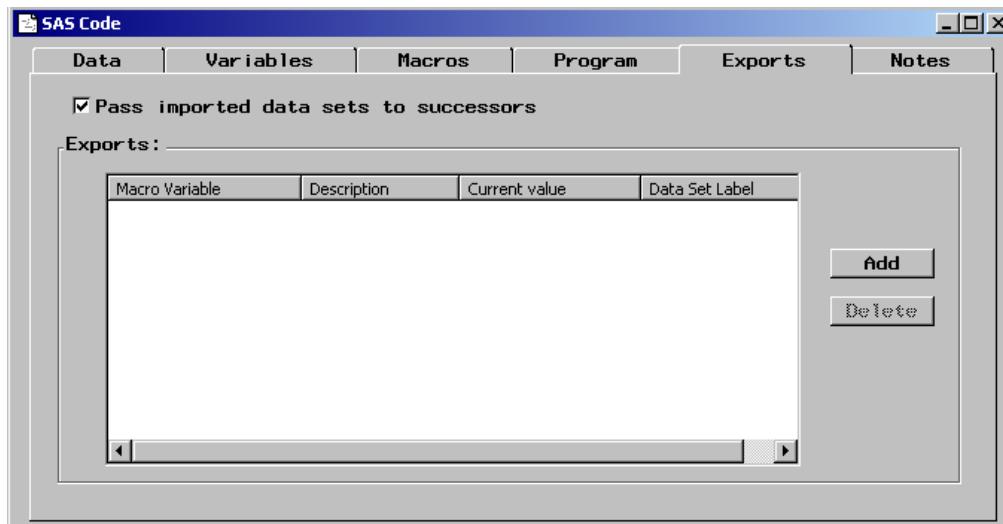


3. Open the new SAS Code node.

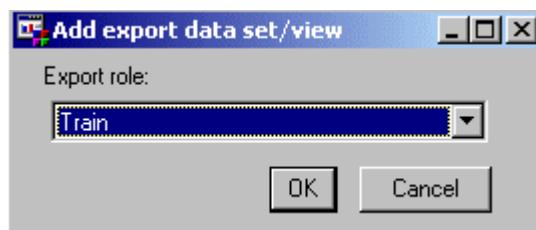
4. Select the **Macros** tab. Observe that the name of the training data set is &\_TRAIN. As seen earlier, you can use the macro names in the programs used in this node. It is unnecessary to know the exact name that the Enterprise Miner has assigned to each data set.



5. Select the **Exports** tab.



6. Select **Add**.



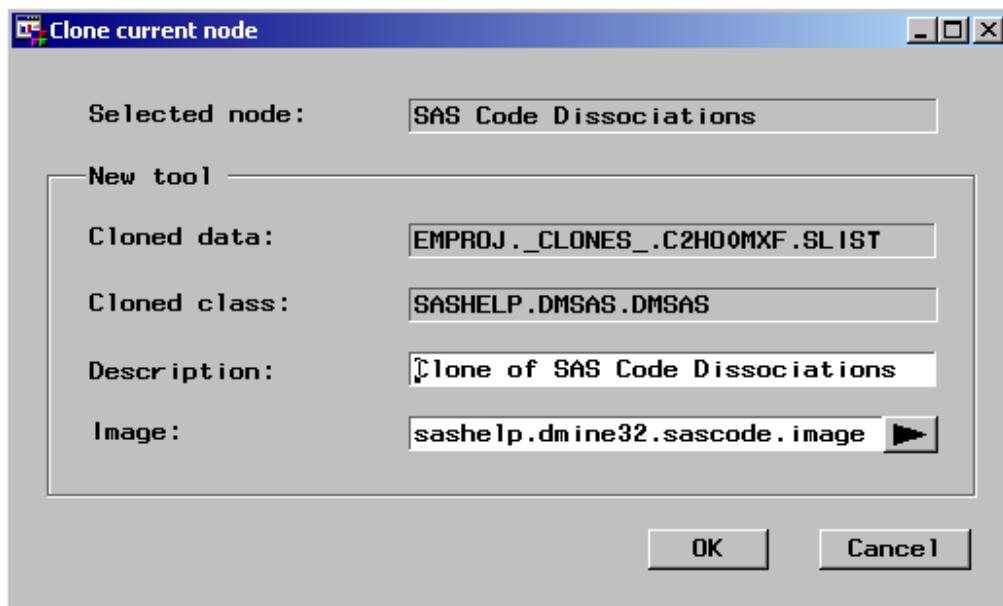
7. Select **OK** to choose the training data for export. Note that the name of the data set is &\_TRA.

8. Deselect **Pass imported data sets to successors.**
9. Select the **Program** tab.
10. Select **File**  $\Rightarrow$  **Import File** and a browser opens. Use the browser to locate the program Dissoc.sas.
11. Highlight the program Dissoc.sas.
12. Select **OK**.
13. Close the SAS code node and save changes when prompted.
14. Rename the node in the diagram to **SAS Code Dissociations**.

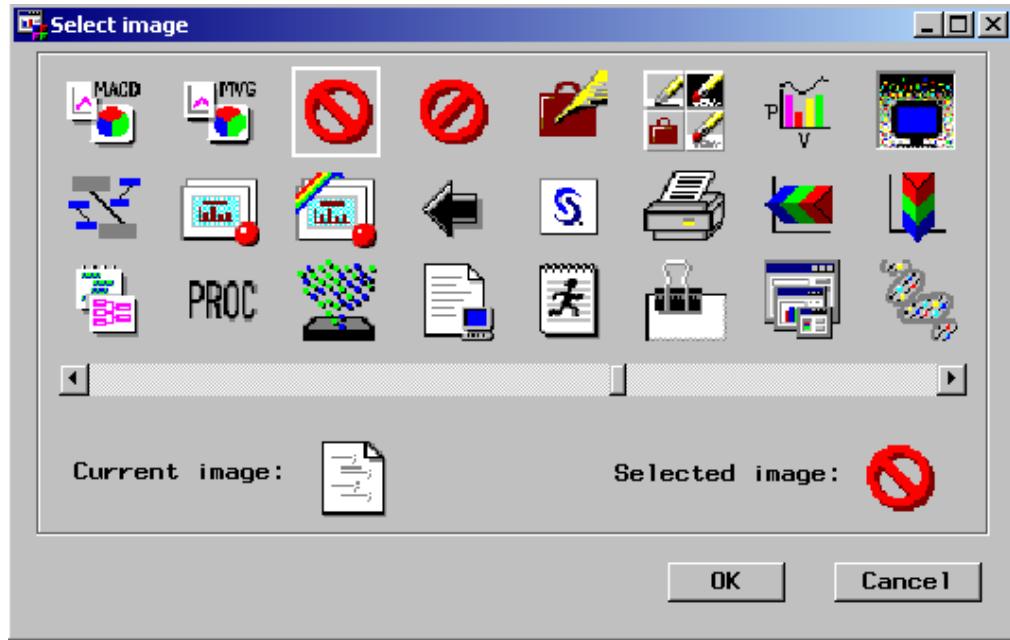
### Node Cloning

At this point, the SAS Code node can be used for dissociation analysis for any data set. If you anticipate doing more dissociation analyses, it would be helpful to save this code node with the other nodes in the Tools tab. By cloning a node, you can add it as a custom node to the tools palette. Clone the SAS Code node and add it to the Node types palette.

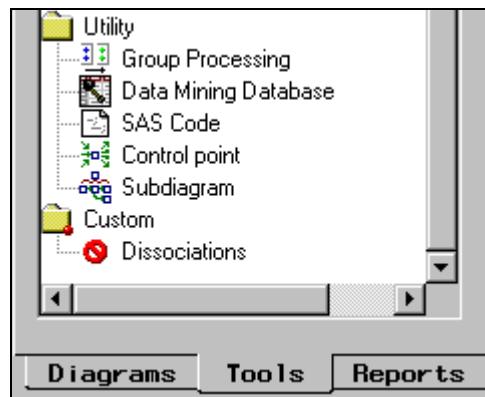
1. Right-click on the **SAS Code Dissociations** node and select **Clone...**



2. Change the Description to **Dissociations**.
3. Select the right arrow next to the Image field.
4. Select an appropriate icon from the palette.



5. Select OK to accept the image.
6. Select OK to close the Clone Current Node window.
7. Select the Tools tab from the left side of the Enterprise Miner application window. Scroll down to the bottom of the tools. A new tool appears at the bottom of the Tools palette with the icon you selected.



The cloned tool can be used in the diagram in place of the SAS Code node. A cloned tool is saved in the project library. Consequently, every diagram created within the project will have the Dissociations node available for use. Note that this cloned node has to be modified with variable and level names that are specific to the data set with which it will be used.

## Modifying the Dissociations Node

1. Open the SAS Code Dissociations node. The Program tab is active.
2. Modify the first three lines of the imported program as shown below:

```
%let values='SVG','CKING','MMDA';
%let in=&_TRAIN;
%let out=&_TRA;
```

The first line identifies the values of the target for which negations are created. The values must be enclosed in quotes and separated by commas. This program scans each ID (ACCT) to see if the items (services) specified in the **values** are present. If not, the data is augmented with the negated items. In this case you will create negations only for savings accounts, checking accounts, and money market accounts.

The second and third lines provide macro names for the training data and the augmented (exported) data.

3. Close the SAS Code Dissociations node and save the changes.
4. Run the diagram from the SAS Code Dissociations node, but do not view the results.
5. Open the Association node.
6. Select the **Data** tab.
7. Select **Properties...** and then select the **Table View** tab. The listing is of the augmented data that was exported from the SAS Code node.
8. Close the Association node.
9. Run the Association node and view the results.



Because the sequence variable was not passed to the new Association node by the SAS Code Dissociations node, the default analysis will be an association analysis rather than a sequence analysis. This is appropriate because there is no sequence to **not** purchasing a product.

The results now list association and dissociation rules. For example, among customers without a money market account, 65.58% have a savings account (rule 4).

Results - Association						
Rules	Frequencies	Code	Log	Notes		
	Relations	Lift	Support(%)	Confidence(%)	Transaction Count	Rule
1	2	1.02	54.17	63.15	4329.0	CKING ==> SVG
2	2	1.02	54.17	87.56	4329.0	SVG ==> CKING
3	2	1.06	54.14	87.50	4326.0	SVG ==> ~MMDA
4	2	1.06	54.14	65.58	4326.0	~MMDA ==> SVG
5	2	1.10	36.19	94.11	2892.0	ATM ==> CKING
6	2	1.10	36.19	42.19	2892.0	CKING ==> ATM
7	2	1.04	33.12	86.14	2647.0	ATM ==> ~MMDA
8	2	1.04	33.12	40.12	2647.0	~MMDA ==> ATM
9	2	1.08	25.69	66.81	2053.0	ATM ==> SVG
10	2	1.08	25.69	41.53	2053.0	SVG ==> ATM
11	2	1.17	16.47	19.20	1316.0	CKING ==> HMEQLC
12	2	1.17	16.47	100.00	1316.0	HMEQLC ==> CKING
13	2	1.04	15.72	64.08	1256.0	CD ==> SVG
14	2	1.04	15.72	25.40	1256.0	SVG ==> CD
15	2	1.04	15.58	18.16	1245.0	CKING ==> MMDA
16	2	1.04	15.58	22.21	1245.0	MMDA ==> CKING

Close the Association node when you have finished examining the results.

# Appendix A References

A.1 References .....	A-3
----------------------	-----



## A.1 References

- Beck, A. 1997. "Herb Edelstein discusses the usefulness of datamining." *DS Star*, Vol. 1, N0. 2. Available <http://www.tgc.com/dsstar/>.
- Berry, M. J. A. and G. Linoff. 1997. *Data Mining Techniques for Marketing, Sales, and Customer Support*. New York: John Wiley & Sons, Inc.
- Bishop, C. M. 1995. *Neural Networks for Pattern Recognition*. New York: Oxford University Press.
- Bigus, J. P. 1996. *Data Mining with Neural Networks: Solving Business Problems - from Application Development to Decision Support*. New York: McGraw-Hill.
- Breiman, L., et al. 1984. *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group.
- Chatfield, C. 1995. "Model uncertainty, data mining and statistical inference (with discussion)." *JRSS B* 419-466.
- Einhorn, H. J. 1972. "Alchemy in the behavioral sciences." *Public Opinion Quarterly* 36:367-378.
- Hand, D. J. 1997. *Construction and Assessment of Classification Rules*. New York: John Wiley & Sons, Inc.
- Hand, D. J. and W. E. Henley. 1997. "Statistical classification methods in consumer credit scoring: a review." *Journal of the Royal Statistical Society A* 160:523-541.
- Hand, David, Heikki Mannila, and Padraig Smyth. 2001. *Principles of Data Mining*. Cambridge, Massachusetts: The MIT Press.
- Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag New York, Inc.
- Hoaglin, D. C., F. Mosteller, and J. W. Tukey. 1983. *Understanding Robust and Exploratory Data Analysis*. New York: John Wiley & Sons, Inc.
- Huber, P. J. 1997. "From large to huge: A statisticians reaction to KDD and DM." *Proceedings, Third International Conference on Knowledge Discovery and Data Mining*. AAAI Press.
- John, G. H. 1997. *Enhancements to the Data Mining Process*. Ph.D. thesis, Computer Science Department, Stanford University.
- Kass, G. V. 1980. "An exploratory technique for investigating large quantities of categorical data." *Applied Statistics* 29:119-127.
- Little, R. J. A. and D. B. Rubin. 1987. *Statistical Analysis with Missing Data*. New York: John Wiley & Sons, Inc.
- Little, R. J. A. 1992. "Regression with missing X's: A review." *Journal of the American Statistical Association* 87:1227-1237.

- Lovell, M. C. 1983. "Data Mining." *The Review of Economics and Statistics*. Vol. LXV, number 1.
- Michie, D., D. J. Spiegelhalter, and C. C. Taylor. 1994. *Machine Learning, Neural and Statistical Classification*. New York: Ellis Horwood.
- Morgan, J. N. and J. A. Sonquist. 1963. "Problems in the analysis of survey data, and a proposal." *Journal of the American Statistical Association* 58:415-434.
- Mosteller, F and J. W. Tukey. 1977. *Data Analysis and Regression*. Reading, MA: Addison-Wesley.
- Palmeri, C. 1997. "Believe in yourself, believe in merchandise." *Forbes* Vol. 160, No. 5:118-124
- Piatesky-Shapiro, G. 1998. "What Wal-Mart might do with Barbie association rules." *Knowledge Discovery Nuggets*, 98:1. Available <http://www.kdnuggets.com/>.
- Quinlan, J. R. 1993. *C4.5 Programs for Machine Learning*. Morgan Kaufmann.
- Ripley, B. D. 1996. *Pattern Recognition and Neural Networks*. New York: Cambridge University Press.
- Rosenberg, E. and A. Gleit. 1994. "Quantitative methods in credit management." *Operations Research*, 42:589-613.
- Rud, Olivia Parr. 2001. *Data Mining Cookbook: Modeling Data, Risk, and Customer Relationship Management*. New York: John Wiley & Sons, Inc.
- Sarle, W. S. 1997. "How to measure the importance of inputs." SAS Institute Inc. Available <ftp://ftp.sas.com/pub/neural/importance.html>.
- Sarle, W.S. 1994a. "Neural Networks and Statistical Models," *Proceedings of the Nineteenth Annual SAS® Users Group International Conference*. Cary: NC, SAS Institute Inc., 1538-1550.
- Sarle, W.S. 1994b. "Neural Network Implementation in SAS® Software," *Proceedings of the Nineteenth Annual SAS® Users Group International Conference*. Cary: NC, SAS Institute Inc., 1550-1573.
- Sarle, W.S. 1995. "Stopped Training and Other Remedies for Overfitting." *Proceedings of the 27th Symposium on the Interface*.
- SAS Institute Inc. 1990. *SAS® Language: Reference, Version 6, First Edition*. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 1990. *SAS® Procedures Guide, Version 6, Third Edition*. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 1990. *SAS/STAT® User's Guide, Version 6, Fourth Edition, Volumes 1 and 2*. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 1995. *Logistic Regression Examples Using the SAS® System, Version 6, First Edition*. Cary, NC: SAS Institute Inc.

SAS Institute Inc. 1995. *SAS/INSIGHT® User's Guide, Version 6, Third Edition*. Cary, NC: SAS Institute Inc.

Smith, M. 1993. *Neural Networks for Statistical Modeling*. New York: Van Nostrand Reinhold.

Weiss, S.M. and C. A. Kulikowski. 1991. *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*. San Mateo, CA: Morgan Kaufmann.

Zhang, Heping, and Burton Singer. 1999. *Recursive Partitioning in the Health Sciences*. New York: Springer-Verlag New York, Inc.



# Appendix B Index

## A

Analysis mode options, 8-10–8-11

analytical expert, 1-11

analytical tools, 1-8

artificial neural networks, 5-5

Assess nodes

    Enterprise Miner, 1-23

Assessment node

    Enterprise Miner, 1-23

association analysis

    compared with sequence analysis, 8-18

    Enterprise Miner, 8-9–8-17

    overview, 8-4–8-6

Association node

    Enterprise Miner, 1-18

association rules, 8-4

## B

backward selection method

    Regression node, 3-7, 3-43–3-45

bagging models, 6-12

base SAS

    generating scoring code, 6-27–6-29

Bayes rule, 2-70

boosting models, 6-12

## C

C\*Score node

    Enterprise Miner, 1-24

candidate models

    comparing, 6-5–6-9

CART algorithm, 2-43

case-control sampling, 2-25

CHAID algorithm, 2-43

chi-square criterion

    Variable Selection node, 4-15–4-16

choice-based sampling, 2-25

classification trees, 2-38

cloning nodes, 8-27–8-28

cluster analysis, 7-3–7-37

    K-means, 7-5–7-8

clustering, 7-4

K-means, 7-5–7-8

Clustering node, 7-11

    Enterprise Miner, 1-20

Control Point node

    Enterprise Miner, 1-25

credit risk management, 1-7

credit scoring, 1-7

Cumulative %Response charts

    Enterprise Miner, 2-59–2-61

curse of dimensionality, 2-29

customer relationship management, 1-7

## D

data expert, 1-11

data mining

    analytical tools, 1-8

    definition, 1-3

    KDD, 1-8

    machine learning, 1-8–1-9

    neurocomputing, 1-8–1-9

    overview, 1-3–1-14

    pattern recognition, 1-8

    problem formulation, 1-10

    problem translation, 1-12

    required expertise, 1-11

    SEMMA process, 1-15–1-26

    steps, 1-10

Data Mining Database node

    Enterprise Miner, 1-25

Data Partition node, 3-22–3-24

    Enterprise Miner, 1-18, 2-35–2-36

data replacement, 3-29–3-30

Data Set Attributes node

    Enterprise Miner, 1-19

data splitting, 2-32

data warehouses, 2-21

database marketing, 1-7

decision trees

    algorithms, 2-43

    benefits, 2-44

    building, 2-47–2-59

    classification trees, 2-38

    drawbacks, 2-45

    fitted, 2-38

- interactive training, 2-62–2-67
  - limiting growth, 2-57–2-58
  - options, 2-55
  - possible splits, 2-40
  - pruning, 2-42
  - recursive partitioning, 2-39, 2-44
  - regression trees, 2-38
  - splitting criteria, 2-41
  - stunting, 2-42
  - variable selection, 4-4–4-5, 4-7–4-10
  - dissociation analysis, 8-24
    - Enterprise Miner, 8-25–8-30
  - Distribution Explorer node
    - Enterprise Miner, 1-18
  - distributions
    - inspecting, 3-13
  - domain expert, 1-11
- E**
- ensemble models
    - bagging, 6-12
    - boosting, 6-12
    - combined, 6-10–6-11, 6-13–6-15
    - stratified, 6-12
  - Ensemble node, 6-10–6-14
    - Enterprise Miner, 1-22
  - Enterprise Miner
    - adding nodes, 2-11–2-12
    - Assess nodes, 1-23
    - association analysis, 8-9–8-17
    - building decision trees, 2-47–2-59
    - building the initial flow, 2-11–2-12
    - choosing a decision threshold, 2-69–2-71
    - combined ensemble models, 6-13–6-15
    - comparing candidate models, 6-5–6-9
    - computing descriptive statistics, 2-17
    - Cumulative %Response charts, 2-59–2-61
    - Data Partition node, 2-35–2-36
    - decision tree options, 2-55
    - defining new libraries, 2-9–2-11
    - dissociation analysis, 8-25–8-30
    - Explore nodes, 1-18–1-19
    - fitting decision trees, 3-47–3-49
    - fitting neural network models, 5-12–5-22
    - fitting regression models, 3-42–3-47
    - generating scoring code, 6-19–6-27
    - identifying input data, 3-10–3-13
    - identifying target variables, 3-12–3-13
    - imputing missing values, 3-31–3-36
    - Input Data Source node, 2-13
  - inspecting distributions, 2-16, 3-13
  - K-means cluster analysis, 7-9–7-23
  - limiting decision trees growth, 2-57–2-58
  - lock files, 2-19
  - Model nodes, 1-21–1-23
  - model roles, 2-15
  - Modify nodes, 1-19–1-20
  - modifying variable information, 2-17, 3-14
  - Multiplot node, 2-18–2-19
  - opening, 2-4
  - performing regressions using, 3-9–3-49
  - placement of nodes in process flow
    - diagrams, 1-26
  - profit matrix, 2-69–2-71
  - Reports tab, 2-6
  - Sample nodes, 1-17
  - Scoring nodes, 1-24
  - self-organizing maps (SOMs), 7-28–7-37
  - sequence analysis, 8-19–8-23
  - setting up projects, 2-4–2-5
  - specifying input data, 2-13–2-15
  - target profiler, 3-14–3-21
  - target variables, 2-16
  - Tools tab, 2-6
  - Utility nodes, 1-25
  - variable selection, 4-3–4-16
  - variable selection methods, 4-4–4-16
  - variable transformations, 3-36–3-41
  - Variables tab, 2-14–2-15
- Explore nodes**
- Enterprise Miner, 1-18–1-19
- F**
- Filter Outliers node
    - Enterprise Miner, 1-20
  - fitted decision trees, 2-38
  - fitting default, 3-47–3-49
  - fitting models, 2-33–2-34
  - forward selection method
    - Regression node, 3-7, 3-43–3-45
  - fraud detection, 1-7
- G**
- generalized linear models, 5-9
  - Group Processing node
    - Enterprise Miner, 1-25
- H**
- healthcare informatics, 1-7
  - hidden units

- neural networks, 5-4
- I**
- ID3 algorithm, 2-43
- imputing missing values, 3-31
  - methods, 3-32
  - reasons for, 3-6
- Input Data Source node
  - Enterprise Miner, 1-17, 2-13
- Insight node, 3-22–3-26
  - Enterprise Miner, 1-18
- K**
- KDD, 1-8
- K-means cluster analysis, 7-5–7-8
  - Enterprise Miner, 7-9–7-23
- L**
- linear regression
  - compared with logistic regression, 3-4
- Link Analysis node
  - Enterprise Miner, 1-19
- link functions
  - neural networks, 5-7
- lock files, 2-19
- logistic regression
  - compared with linear regression, 3-4
  - logit transformation, 3-5
  - visualizing, 5-21
- M**
- machine learning, 1-8–1-9
- Memory-Based Reasoning node
  - Enterprise Miner, 1-22
- missing values, 2-27
  - analysis strategies, 2-28
  - imputing, 3-31–3-36
  - reasons for imputing, 3-6
- MLPs. See multilayer perceptrons (MLPs)
- Model nodes
  - Enterprise Miner, 1-21–1-23
- model roles, 2-15
- models
  - comparing, 2-58–2-61
  - fitting, 2-33–2-34
  - model complexity, 2-33
  - overfitting, 2-34
  - underfitting, 2-33
- Modify nodes
- Enterprise Miner, 1-19–1-20
- multilayer perceptrons (MLPs), 5-5–5-8
  - constructing, 5-13–5-17
  - universal approximators, 5-7
- Multiplot node
  - Enterprise Miner, 1-18, 2-18–2-19
- N**
- Neural Network node
  - Enterprise Miner, 1-21
- neural networks
  - activation functions, 5-5–5-7
  - artificial, 5-5
  - hidden units, 5-4
  - link functions, 5-7
  - multilayer perceptrons, 5-5–5-8
  - organic, 5-4
  - training, 5-8
  - visualizing, 5-12–5-22
- neurocomputing, 1-8–1-9
- nodes
  - cloning, 8-27–8-28
- O**
- outliers, 2-27
- oversampling, 2-25
- P**
- pattern recognition, 1-8
- predictive modeling, 1-13
- Princomp/Dmneural node
  - Enterprise Miner, 1-21
- Prior tab
  - predefined prior vectors, 3-20
- prior vectors, 3-20
- profit matrices, 2-69–2-71
- R**
- recursive partitioning, 2-39, 2-44
- regression analysis, 1-14
  - performing using Enterprise Miner, 3-9–3-49
- Regression node, 3-4
  - Enterprise Miner, 1-21
  - variable selection methods, 3-7
- regression trees, 2-38
- Replacement node, 3-29–3-30
  - Enterprise Miner, 1-20
- Reporter node

- Enterprise Miner, 1-23
- Reports tab
  - Enterprise Miner, 2-6
- ROC charts, 6-3–6-4
- S
- Sample nodes
  - Enterprise Miner, 1-17
- Sampling node
  - Enterprise Miner, 1-17
- SAS Code node
  - Enterprise Miner, 1-25
- Score node, 6-19
  - Enterprise Miner, 1-24
- scoring code, 6-17
  - generating using base SAS, 6-27–6-29
  - generating using Enterprise Miner, 6-19–6-27
- scoring data, 6-16–6-29
- Scoring nodes
  - Enterprise Miner, 1-24
- self-organizing maps (SOMs), 7-25
  - Enterprise Miner, 7-28–7-37
- SEMMA process, 1-15–1-26
- sequence analysis
  - compared with association analysis, 8-18
  - Enterprise Miner, 8-19–8-23
- SOM/Kohonen node
  - Enterprise Miner, 1-20
- SOMs. See self-organizing maps (SOMs)
- stepwise regression, 4-4
- stepwise selection method
  - Regression node, 3-7, 3-45
- stratified models, 6-12
- Subdiagram node
  - Enterprise Miner, 1-25
- supervised classification, 1-13–1-14, 7-9
- survival analysis, 1-14
- T
- target marketing, 1-7
- target profiler, 3-14–3-21
- target variables, 2-16
  - identifying, 3-12–3-13
  - lack of, 2-24–2-25
- temporal infidelity, 6-18
- test data sets, 2-32
- Time Series node
  - Enterprise Miner, 1-20
- Tools tab
  - Enterprise Miner, 2-6
- training neural networks, 5-8
- transaction count, 8-21
- Transform Variables node, 3-36
  - Enterprise Miner, 1-19
- Tree node
  - Enterprise Miner, 1-21
- Two Stage Model node
  - Enterprise Miner, 1-22
- U
- unsupervised classification, 7-4, 7-9
- User-Defined Model node
  - Enterprise Miner, 1-22
- Utility nodes
  - Enterprise Miner, 1-25
- V
- validation data sets, 2-32
- variable selection, 4-3–4-16
- Variable Selection node, 4-4, 4-6, 4-11–4-15
  - Chi-square criterion, 4-15–4-16
  - Enterprise Miner, 1-19
- variables
  - measurement levels, 2-15
  - model roles, 2-15
  - modifying information, 2-17
  - target, 2-16
  - types, 2-15
- Variables tab
  - Enterprise Miner, 2-14–2-15