

Altair® Knowledge Studio®

Advanced Modelling with KnowledgeSTUDIO

Altair

February 14, 2020



Altair

1 Course Introduction	1-1
1.1 Introduction	1-1
1.2 Data	1-1
1.3 Course Objectives	1-2
2 Introduction to Data Mining	2-1
2.1 Introduction	2-1
2.2 What is Data Mining	2-1
2.2.1 Data Mining Interpretations	2-1
2.3 Data Mining Consideration	2-3
2.4 A Strategy for Data Mining	2-3
2.5 CRISP-DM – Cross-Industry Standard Process for Data Mining	2-4
2.6 Classification of Data Mining Techniques	2-6
2.7 Predictive Models	2-7
2.7.1 Clustering Techniques	2-12
2.8 Data Mining Solutions to Business Problems	2-14
2.9 Conclusion	2-16
3 Introduction to KnowledgeSTUDIO	3-1
3.1 Introduction	3-1
3.2 Starting KnowledgeSTUDIO	3-1
3.3 Menu System and Toolbar	3-2
3.4 Setting Connections and the Working Directory	3-3
3.5 Projects	3-4
3.6 Nodes and Palettes	3-6
3.7 Filtering, Searching and Node Availability	3-6
3.7.1 Filtering	3-6
3.7.2 Searching	3-6
3.7.3 Node Availability	3-7

3.8	Creating Workflows	3-7
3.8.1	Adding Nodes	3-8
3.8.2	Node States	3-12
3.8.3	Accessing Node Options	3-12
3.9	Additional Workflow Features	3-14
3.9.1	Complex Workflows	3-14
3.10	Help Options	3-15
3.11	Conclusion	3-16
4	Data Exploration and Profiling	4-1
4.1	Introduction	4-1
4.2	Data	4-1
4.3	Data Profiling with KnowledgeSTUDIO	4-2
4.4	Overview Report	4-3
4.5	Dataset Chart Tab	4-4
4.6	Data Tab	4-5
4.7	Segment Viewer Tab	4-5
4.8	Cross Tabs Tab	4-7
4.9	Characteristic Analysis	4-9
4.10	Correlations Tab	4-10
4.11	Saved Charts Tab	4-11
4.12	Split Screen	4-12
4.13	Summary	4-13
4.14	Appendix: Summary Statistics Groups	4-14
5	Data Preparation	5-1
5.1	Introduction	5-1
5.2	The Manipulate Palette	5-1
5.3	Dataset Operations	5-2

5.3.1	Scenario	5-2
5.3.2	Appending Datasets	5-3
5.3.3	Aggregating	5-8
5.3.4	Removing Duplicates	5-11
5.3.5	Merging Records across Datasets	5-14
5.4	Variable Transformations	5-17
5.4.1	The Dataset Editor	5-18
5.4.2	Field Transformations: Identifying Special Values	5-19
5.5	The Expression Editor	5-22
5.5.1	Field Transformations: Binning	5-24
5.5.2	Field Transformations: Transforming Multiple Fields	5-27
5.6	LOS Code Generation	5-30
5.7	Conclusion	5-31
6	Variable Selection, Sampling and Partitioning	6-1
6.1	Variable Selection	6-1
6.1.1	The Segment Viewer	6-2
6.1.2	Variable Selection Node	6-2
6.2	Partitioning Features in KnowledgeSTUDIO	6-5
6.2.1	Creating a Randomly Sampled Partition	6-6
6.2.2	Appendix: Creating a Stratified Sample	6-8
6.3	Conclusion	6-11
7	Understanding Decision Trees	7-1
7.1	Introduction	7-1
7.2	The Basics	7-1
7.3	Components of a Decision Tree	7-2
7.4	Dependent and Independent Variables	7-2
7.5	Altair Decision Tree Algorithms	7-3

7.6	Decision Tree Modelling in KnowledgeSTUDIO	7-3
7.7	Setting the Scene: Data	7-5
7.8	Creating Partitions and Adding a Decision Tree Node	7-6
7.9	Growing the Decision Tree	7-12
7.9.1	Find Split	7-13
7.9.2	Force Split	7-14
7.9.3	Edit Split	7-15
7.9.4	Automatic Grow	7-17
7.10	Automated Versus Manual	7-18
7.11	Tree Object Tabs	7-18
7.12	Improving the Model	7-19
7.12.1	Robust: Nodes Size	7-20
7.12.2	Robust: Number of Branches	7-20
7.12.3	Model Accuracy	7-21
7.12.4	A Simple Model	7-22
7.12.5	Explainable Models and Simpson's Paradox	7-22
7.12.6	Version Control; Creating a Model Instance	7-23
7.13	Conclusion	7-24
8	Ensemble Models	8-1
8.1	Introduction	8-1
8.2	Boosting	8-1
8.3	Bagging	8-2
8.4	Random Forest	8-2
8.5	Demonstration	8-2
8.5.1	Boosting	8-3
8.5.2	Bagging	8-5
8.5.3	Random Forest	8-7

8.5.4 Comparing Results	8-8
9 Model Evaluation and Validation	9-1
9.1 Introduction	9-1
9.2 Evaluate Palette	9-1
9.3 Statistical Validation	9-2
9.3.1 Re-substitution Statistics	9-2
9.3.2 Model Validation Node	9-2
9.4 Thorough Validation	9-9
9.5 Business Validation	9-11
9.5.1 Cumulative tab and Report	9-13
9.5.2 Lift Chart tab and Report	9-15
9.5.3 K-S Chart tab	9-16
9.5.4 ROC Chart Tab	9-17
9.5.5 GOF Statistic tab	9-19
9.5.6 Profit Curve Tab	9-20
9.6 Assessing Variable Importance	9-23
9.7 Conclusion	9-26
9.8 Appendix I - Cross Validation	9-27
9.9 Appendix II - Picking the Best Model	9-30
9.10 Appendix III - Model Analyzer with a Continuous Dependent Variable	9-31
9.10.1 Bias Chart	9-32
9.10.2 Accuracy Chart	9-32
9.10.3 Scatter Plot	9-33
9.10.4 Error Chart	9-34
10 Model Deployment	10-1
10.1 Introduction	10-1
10.1.1 Model Deployment	10-1

10.1.2 Directly Applying Models	10-2
10.1.3 Code Generation	10-5
10.2 Exporting Results	10-7
10.3 Text File Export	10-8
10.4 Conclusion	10-10
11 Introduction to Strategy Trees	11-1
11.1 Introduction	11-1
11.2 Strategy Trees	11-1
11.2.1 Adding Calculations to a Strategy Tree	11-3
11.3 Treatments	11-6
11.3.1 Managing Treatments	11-6
11.4 Treatments from Calculation	11-7
11.4.1 Colour Coding Treatments	11-9
11.4.2 Treatments Report	11-10
11.4.3 Node Report	11-11
11.5 Demonstrations	11-11
11.5.1 Building a Strategy Tree based on a Decision Tree	11-11
11.5.2 Building a Strategy Tree based on a Dataset	11-16
11.6 Conclusion	11-21
12 Strategy Validation and Deployment	12-1
12.1 Introduction	12-1
12.2 Strategy Validation	12-1
12.3 Validating Strategy Trees	12-2
12.3.1 Report Tab	12-4
12.3.2 Tree tab	12-5
12.3.3 Tree Map Tab	12-6
12.3.4 Node Design/Validation Tab	12-6

12.3.5 Node Report Tab	12-6
12.3.6 Profile Chart Tab	12-6
12.4 Strategy Deployment	12-7
12.4.1 Score Current Project Dataset	12-8
12.4.2 Automatic Code Generation	12-10
12.5 Conclusion	12-12
13 Linear Regression	13-1
13.1 Introduction	13-1
13.2 Description	13-1
13.2.1 Simple Linear Regression	13-1
13.2.2 Applying Simple Linear Regression in Practice	13-3
13.2.3 Multiple Linear Regression	13-4
13.2.4 Linear Regression Assumptions	13-5
13.2.5 Model Accuracy: FIT	13-9
13.2.6 Model Accuracy: Variable Coefficients	13-10
13.2.7 Steps when Developing Linear Regression Models	13-10
13.3 Linear Regression in KnowledgeSTUDIO	13-11
13.4 Data Exploration	13-12
13.4.1 Data Preparation	13-14
13.4.2 Building the Linear Regression Model in KnowledgeSTUDIO	13-14
13.4.3 Linear Regression Model Results	13-19
13.4.4 Re-running the Model	13-24
13.4.5 Model Validation, Accuracy and Residual Analysis	13-25
13.4.6 Linear Regression Model Deployment	13-33
13.5 Summary	13-36
14 Logistic Regression	14-1
14.1 Introduction	14-1

14.2 Description	14-2
14.2.1 Logistic Regression Example	14-2
14.2.2 Steps when Developing Logistic Regression Models	14-4
14.3 Logistic Regression in KnowledgeSTUDIO	14-5
14.3.1 Data Partitioning	14-5
14.3.2 Building the Logistic Regression Model in KnowledgeSTUDIO	14-6
14.3.3 Logistic Regression Model Results	14-9
14.3.4 Logistic Regression Model Validation	14-16
14.3.5 Logistic Regression Model Deployment	14-23
14.4 Summary	14-25
15 Neural Networks	15-1
15.1 Introduction	15-1
15.2 Description	15-2
15.2.1 ANN Emulation	15-4
15.2.2 Layers in a Neural Network	15-4
15.2.3 Training a Neural Network	15-4
15.2.4 ANNs in Practice	15-4
15.2.5 Data Requirements	15-5
15.3 Neural Network in KnowledgeSTUDIO	15-5
15.3.1 Model Results	15-11
15.3.2 Model Evaluation and Validation	15-13
15.3.3 Model Deployment	15-15
15.3.4 Understanding the Neural Network and Comparing with other Methods .	15-16
15.3.5 Summary	15-16
16 Cluster Analysis	16-1
16.1 Introduction	16-1
16.2 Description	16-1

16.2.1 Application of Cluster Models	16-2
16.2.2 Cluster Analysis Process	16-2
16.2.3 Distance Measures	16-5
16.2.4 Cluster Analysis with KnowledgeSTUDIO	16-7
16.2.5 Building the Cluster Model in KnowledgeSTUDIO	16-7
16.2.6 Cluster Analysis Outputs Results	16-10
16.2.7 Renaming Clusters	16-14
16.2.8 Scoring Data to Further Characterize Clusters	16-14
16.3 Validating a Cluster Analysis Model: Discussion	16-17
16.4 Summary	16-18
17 Principal Component Analysis	17-1
17.1 Introduction	17-1
17.2 Description	17-2
17.3 Demonstration	17-6
17.3.1 Principal Component Analysis in KnowledgeSTUDIO	17-8
17.3.2 Summary	17-13
18 Market Basket Analysis	18-1
18.1 Introduction	18-1
18.2 Description	18-1
18.2.1 Itemsets and Association Rules	18-2
18.2.2 Rule Statistics	18-3
18.2.3 Market Basket Analysis with KnowledgeSTUDIO	18-4
18.3 Demonstration	18-4
18.3.1 MBA Modelling	18-5
18.4 MBA Model Deployment	18-12
18.5 Summary	18-17

19 Course Summary	19-1
19.1 Introduction	19-1
19.2 Hints and Tips	19-1
19.3 Course Objectives	19-1

Chapter 1: Course Introduction

1.1 Introduction

Welcome to the course; **Advanced Modelling with Altair KnowledgeSTUDIO!**

This course is designed to provide a solid understanding of prevalent *Data Mining* techniques and how to apply these to solve business problems using **Altair KnowledgeSTUDIO**.

Chapters are laid out in a progressive manner providing an understanding of *Data Mining* in general, data import, data preparation, model building, evaluation, validation and deployment, using **KnowledgeSTUDIO**.

Chapters can be taken individually or as a whole and apply to the advanced user and novice alike and cover the following topics:

- Introduction to *Data Mining*
- Introduction to **KnowledgeSTUDIO**
- Creating and Managing **Projects** and **Importing Data**
- **Data Profiling** using **KnowledgeSTUDIO**
- **Dataset Operations** and **Variable Transformations**
- Variable Reduction, Sampling & Partitioning
- Understanding **Decision Trees**
- **Ensemble Models**
- Model **Evaluation** and **Validation**
- Model **Deployment**
- Introduction to **Strategy Trees**
- **Strategy Validation** and **Deployment**
- **Linear Regression**
- **Logistic Regression**
- **Neural Networks**
- **Cluster Analysis**
- **Principal Component Analysis**
- **Market Basket Analysis**

A final chapter provides some concluding statements and a hints and tips section.

1.2 Data

Data used throughout the course will be made available prior to or, at the beginning of the course. This includes some generic datasets fit for purpose and sufficient to explain the concepts delivered throughout. Data should be stored in an accessible location.

1.3 Course Objectives

On completion of this course, attendees should be able to:

- Explain the concept of *Data Mining*
- Navigate the **KnowledgeSTUDIO** interface
- Create and manage projects using **KnowledgeSTUDIO**
- Import data from a variety of sources and file formats
- Analyse and profile data using **KnowledgeSTUDIO** capabilities
- Prepare and transform data including deriving new variables
- Explain, build, evaluate, validate and deploy:
 - **Altair Decision Trees**
 - **Altair Strategy Trees**
 - **Linear Regression models**
 - **Neural Network models**
 - **Cluster Analysis models**
 - **Market Basket Analysis models**
 - **Principal Component Models**

Thank you for your attendance and we wish you a pleasant experience! Further information on all **Altair** products and services can be found at: <http://www.altair.com>. Alternatively, email us at: info@altair.com.

Chapter 2: Introduction to Data Mining

2.1 Introduction

The era of *Big Data* is awakening companies to the opportunities to use customer data for competitive advantage[1]. Broad availability of data and its high level of complexity have made it impossible to simply rely on traditional decision making techniques.

These conventional approaches predominantly use tools such as spreadsheets, database queries and other business intelligence tools. Instead, increasing interest in methods for extracting meaningful information and knowledge from *Big Data* has made a shift from a gut-based decision making style to data-driven decision making style.

At the same time the convergence of various phenomena including powerful computers and computer networks, internet technologies, high volumes of data and efficient algorithms have given rise to data science principles and data mining techniques with the ultimate goal to improve and speed up the decision making process and consequently establish a competitive advantage.

This chapter aims to outline a general definition of data mining, the **CRISP-DM** process as well as a variety of *Data Mining* techniques that can be applied to a variety of business problems.

2.2 What is Data Mining

Myth: Data mining is a computer-driven process that looks for patterns in huge, complex databases and automatically gives results.

Fact: Data mining is a **user-driven process** that uses computers to wade through enormous amounts of data in order to **discover useful patterns**.

2.2.1 Data Mining Interpretations

Many and varied definitions of Data Mining exist, some insightful examples follow.

"A knowledge discovery process of extracting previously unknown, actionable information from very large databases."

Aaron Zornes, The META Group

"The process of discovering meaningful new correlations, patterns and trends by sifting through large amounts of data stored in repositories, using pattern recognition technologies as well as statistical and mathematical techniques."

Erick Brethenoux, Gartner Group

"Data mining is used to discover patterns and relationships in your data in order to help you make better business decisions."

Robert Small, Two Crows

There are a number of synonyms used interchangeably in relation to data mining, examples of which are: data science, predictive analytics, data analysis, pattern analysis/recognition, business intelligence, knowledge discovery, knowledge extraction, Big Data analytics, and text analytics.

Nevertheless, they all have a similar meaning; knowledge extraction from data, and therefore, they all come under the same umbrella of *Data Mining*.

Data mining is a multidimensional concept which requires consideration of four different perspectives:

- **Data perspective**
- **Knowledge perspective**
- **Technique Utilisation**
- **Application perspective[2]**

Data perspective considers different data sources such as relational databases, data warehouses, XML databases, multi-media and streaming data, sensor data etc.

Knowledge perspective includes data classification, segmentations, clustering, trend analysis, outlier analysis, etc.

Techniques utilization incorporates machine learning algorithms, statistics, and data visualization among others.

Application perspective is very broad and includes a range of sectors such as: telecommunication; e.g. churn and customer retention. Retail; e.g. customer segmentation, cross/up sell, market basket analysis, customer lifetime value. Financial; e.g. credit risk, scorecard, fraud detection. Government; education, science etc.

Figure 2.1: Knowledge Management Pyramid and Data Mining

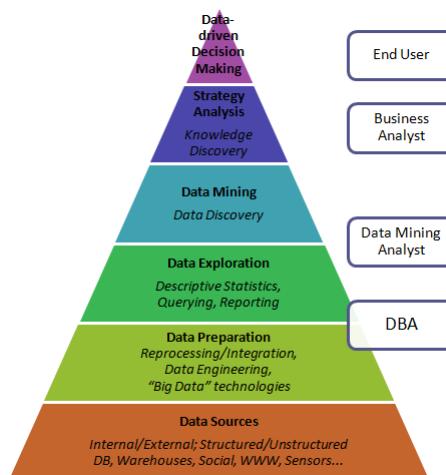


Figure 2.1 depicts the position of the data mining process from a data management perspective

the commonly associated job roles. In progressing towards the pinnacle of the pyramid, data becomes more parsimonious and has increasing potential to support critical business decisions.

2.3 Data Mining Consideration

Data is the most valuable business asset[3] and a prerequisite to *Data Mining*. Valuable sources of information can be found throughout organisations and well beyond their borders i.e. internal and external data sources, both structured and unstructured.

Internal data sources such as databases, data warehouses, reports and spreadsheets store a wealth of useful data. External databases and various Internet sources are also excellent ways of obtaining business intelligence that can be combined with internal sources to get better insights into the problem under investigation.

Data quality is imperative for effective *Data Mining*. Without the appropriate data it would be impossible to derive any valuable knowledge even when employing the best Data mining specialists. The most important data quality requirements are:

- **Relevance** - fitness for purpose
- **Availability** - either internally or externally
- **Completeness** - noisy or missing data are the two the most common degrading factors to data completeness
- **Consistency** - information silos are the main problem of inconsistent data, which could be overcome by using an integrated information system
- **Presentation** - visual presentation; portals, mash-ups, dashboards, graphs, spreadsheets etc
- **Trust** - using reliable data sources and dealing with subjective data in an efficient way; having enough data so the sample size can be a good approximation of the entire population
- **Timeliness** - minimizing delays between data collection and data processing

2.4 A Strategy for Data Mining

Prior to the start of a data mining project it is necessary to answer the following questions:

- What is the substantive problem to be solved?
- What data is available, and what parts are relevant?
- What kind of pre-processing and cleaning is needed before starting the project?
- What data mining technique(s) will be used?
- How will the results of the data mining analysis be evaluated?
- How to get the most out of the information obtained from the Data mining analysis?

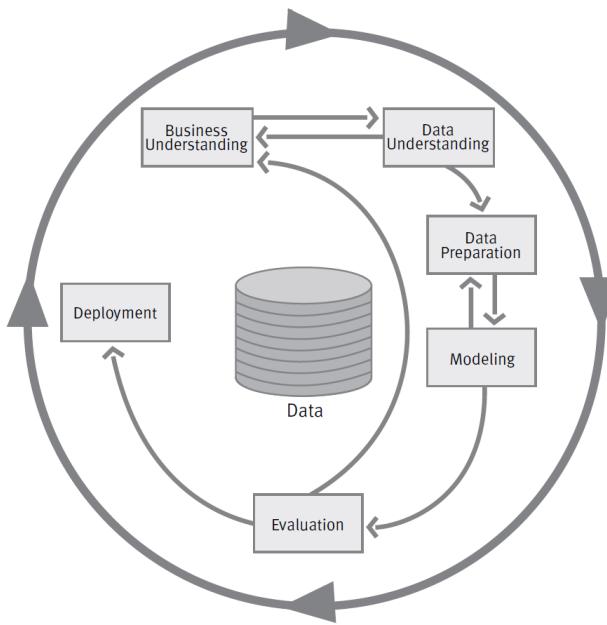
The best way to answer these questions is to follow a structured approach. The **CRoss Industry Standard Process for Data Mining; CRISP-DM**, provides an appropriate framework to capture these aspects.

2.5 CRISP-DM – Cross-Industry Standard Process for Data Mining

CRISP-DM[4] is an industry standard and a framework for the *Data Mining* process that describes a commonly used approach that data scientists utilize to tackle problems.

The **CRISP-DM** reference model encourages best practices and offers organizations the structure needed to realize better and faster results from *Data Mining*.

Figure 2.2: Phases of the CRISP-DM Reference Model



The *Data Mining* lifecycle consists of six phases as illustrated in figure 2.2. The sequence of phases is not firm and it is often necessary to move forwards and backwards during the process.

The outcome of each phase is the input into the next one – as indicated by the arrows. The outer cycle represents the cyclical nature of *Data Mining* itself – the lessons learned from a deployed solution could trigger new business questions. In summary, the **CRISP-DM** phases are:

- | | |
|---|---|
| <ul style="list-style-type: none">• Business Understanding• Data Understanding• Data Preparation• Modelling• Evaluation/Validation• Deployment | <ul style="list-style-type: none">– determine business objectives and data mining goals– collect, describe, explore and verify quality of data– select, clean, construct, integrate and format data– select, generate and build model(s)– do results achieve business objectives?– integrate new knowledge into business processes |
|---|---|

Business Understanding is the initial phase which focuses on understanding the project objectives and requirements from a business perspective.

This knowledge is then defined as a data mining problem and a preliminary plan is designed to achieve the objectives.

The **Data Understanding** phase starts with initial data collection and proceeds with activities that enable data scientists to become familiar with the data, identify data quality problems, discover first insights into the data, and detect interesting subsets to form hypotheses regarding hidden information.

The **Data Preparation** phase covers all activities needed to construct the final dataset, data that will be fed into the modelling tools, from the initial raw data. Data preparation tasks are likely to be performed multiple times and not in any prescribed order. Tasks include, table, record and attribute selection, as well as transformation and cleaning of data for modelling tools.

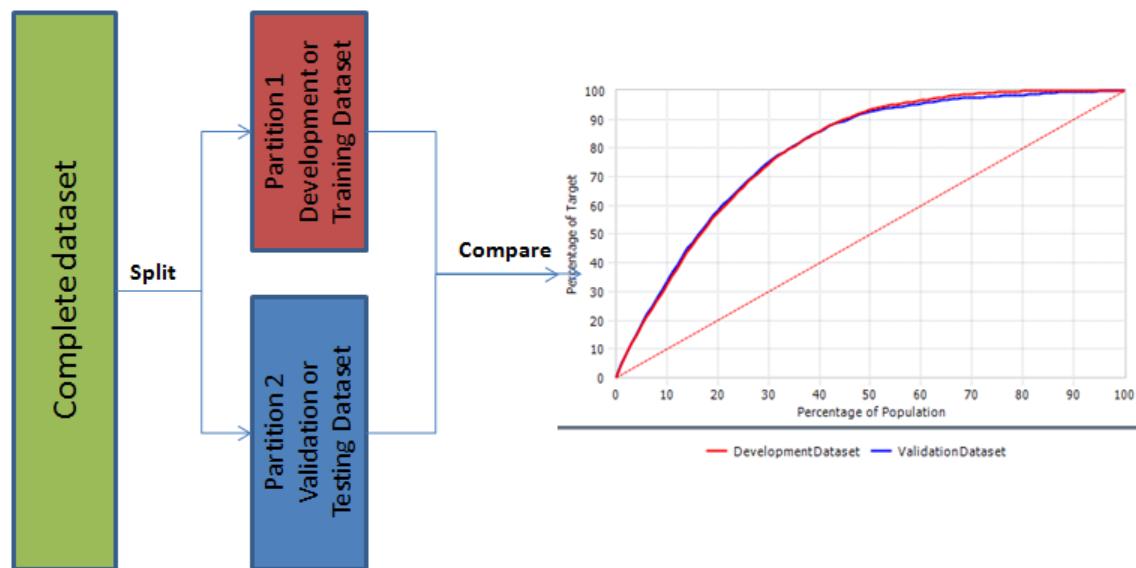
The **Modelling** phase selects and applies one or multiple modelling techniques by specifying and calibrating parameters to optimal values. Typically, there are several techniques for the same data mining problem type. Some techniques require data in specific formats, hence, iterative data preparation is necessary.

The **Evaluation/Validation** phase, is of immense importance. Prior to proceeding to final deployment of the model it is important to thoroughly evaluate the model and review the steps executed to create the model in order to be certain the model properly achieves the business objectives.

A key objective is to determine if there is some important business issue that has not been sufficiently considered. At the end of this phase a decision on the use of data mining results should be reached.

The business validation is of special importance as it assesses the business benefit of the model. The aim of the business validation is to test if the model generalizes on an independent dataset which is different from the one used to build the model. The best way to test this requirement is to use a testing or validation dataset as represented in figure 2.3.

Figure 2.3: Business Model Validation



The **Deployment** phase puts the validated data model into operation.

Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable *Data Mining* process across the enterprise; for example, scoring live data with the created model for immediate decision making; e.g. accept/reject loan.

In many cases, it is the end user, not the data analyst who carries out the deployment steps; therefore, it is important for the end-user to understand what actions need to be carried out in order to make use of the created model.

Figure 2.4 depicts an outline of each phase accompanied by generic tasks in **bold** and associated outputs in *italic*.

Figure 2.4: CRISP-DM Model Tasks

Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Determine Business Objectives <i>Background</i> <i>Business Objectives</i> <i>Business Success Criteria</i> Assess Situation <i>Inventory of Resources Requirements, Assumptions, and Constraints</i> <i>Risks and Contingencies</i> <i>Terminology</i> <i>Costs and Benefits</i> Determine Data Mining Goals <i>Data Mining Goals</i> <i>Data Mining Success Criteria</i> Produce Project Plan <i>Project Plan</i> <i>Initial Assessment of Tools and Techniques</i>	Collect Initial Data <i>Initial Data Collection Report</i> Describe Data <i>Data Description Report</i> Explore Data <i>Data Exploration Report</i> Verify Data Quality <i>Data Quality Report</i>	Select Data <i>Rationale for Inclusion/Exclusion</i> Clean Data <i>Data Cleaning Report</i> Construct Data <i>Derived Attributes Generated Records</i> Integrate Data <i>Merged Data</i> Format Data <i>Reformatted Data</i> <i>Dataset</i> <i>Dataset Description</i>	Select Modeling Techniques <i>Modeling Technique</i> <i>Modeling Assumptions</i> Generate Test Design <i>Test Design</i> Build Model <i>Parameter Settings</i> <i>Models</i> <i>Model Descriptions</i> Assess Model <i>Model Assessment</i> <i>Revised Parameter Settings</i>	Evaluate Results <i>Assessment of Data Mining Results w.r.t. Business Success Criteria</i> <i>Approved Models</i> Review Process <i>Review of Process</i> Determine Next Steps <i>List of Possible Actions Decision</i>	Plan Deployment <i>Deployment Plan</i> Plan Monitoring and Maintenance <i>Monitoring and Maintenance Plan</i> Produce Final Report <i>Final Report</i> <i>Final Presentation</i> Review Project <i>Experience Documentation</i>

2.6 Classification of Data Mining Techniques

Figure 2.5 is a simplistic classification of data mining techniques.

The top level division separates techniques with and without a dependent variable. In the figure, this is referred to as a response variable. Supervised methods are those with a response variable, whereas unsupervised methods are those without a response variable.

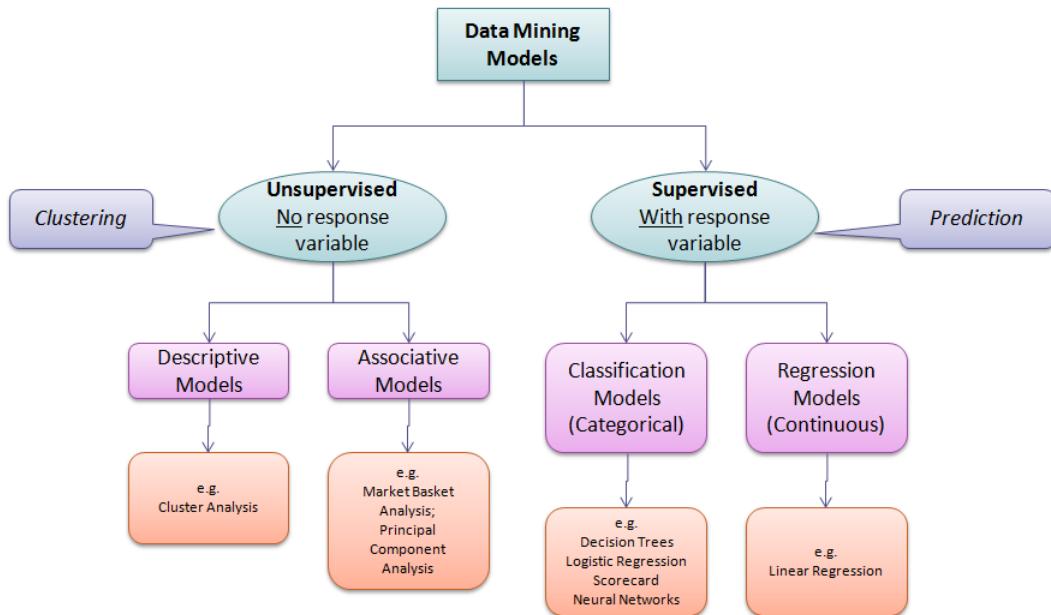
At the second level there are four different models, two for each of the higher level groupings.

For methods oriented to prediction, the main distinction is regarding the nature of the response variable. Classification models relate to a categorical response variable and linear regression models relate to a continuous response variable.

The bottom level of the flow chart shows a set of the most popular data mining algorithms such as **Cluster**

Analysis, Market Basket Analysis, Decision Trees, Logistic Regression, Scorecards and Linear Regression.

Figure 2.5: Classification of DataMining Techniques [5]

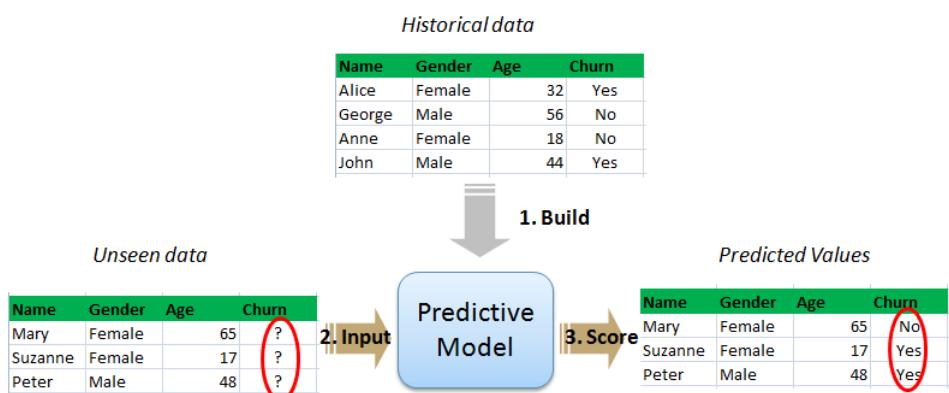


2.7 Predictive Models

A predictive model detects and identifies patterns in existing data in order to forecast future outcome.

Building a predictive model involves the application of statistical techniques to capture and expose the information contained in the data.

Figure 2.6: Predictive Modelling



Decision Trees

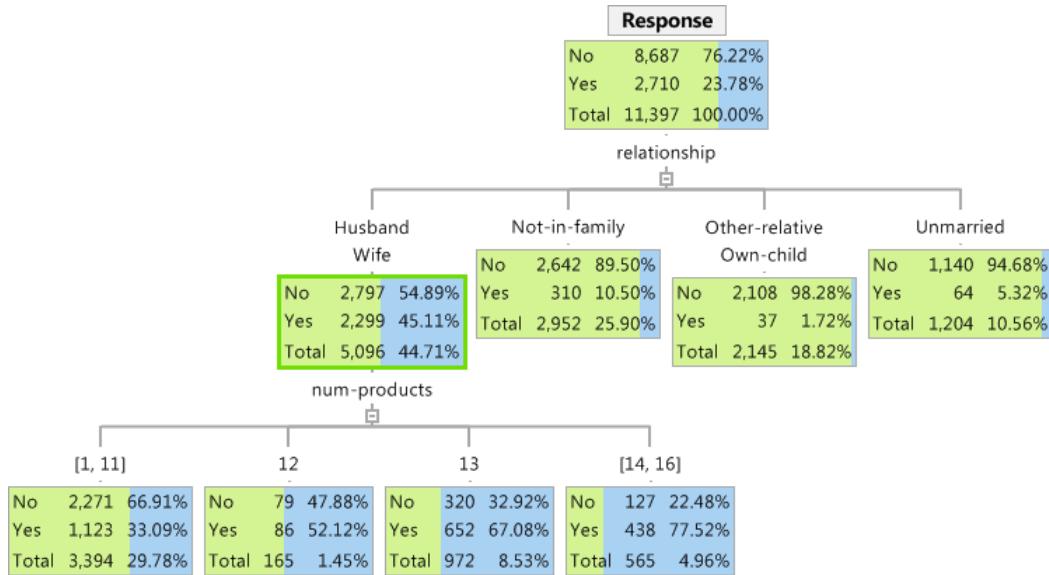
Decision Trees are one of the most versatile modelling techniques. These models can be used in their own right for prediction or they can be used as a pre-modelling technique in the development of other models.

For example **Decision Trees** can be used for initial selection of a set of variables appropriate to feed into an alternative modelling technique. **Decision Trees** can also be used after modelling to explain the workings of a more complex model. For example; to explain how a neural network has made its decisions.

Decision Trees are very intuitive, highly visual, easy to use and understand and generally very efficient predictors. As classification models they are appropriate for categorical dependent variables but can also be used to predict continuous dependent variables. They can handle a large number of independent variables.

Decision Trees can be generated using a variety of methods. **Decision Trees** enable direct extraction into a set of decision rules, with ability to incorporate business rules. They also provide implicit feature weightings making them suitable for initial feature selection to feed into the other modelling techniques.

Figure 2.7: A Decision Tree



Linear Regression

Linear Regression is a common statistical modelling technique with many practical uses in prediction and forecasting, where the former is a more general term relating to any kind of prediction and the latter is the process of making statements about future events whose actual outcomes have not yet been observed.

Linear Regression is appropriate for continuous variables. In particular, the dependent variable (*DV*) must be continuous. The independent variables (*IVs*) should typically be continuous as well, however it is often allowable to include some categorical *IVs* by dummy coding them, although this should usually be done for only a subset of the *IVs*, and not the majority of them.

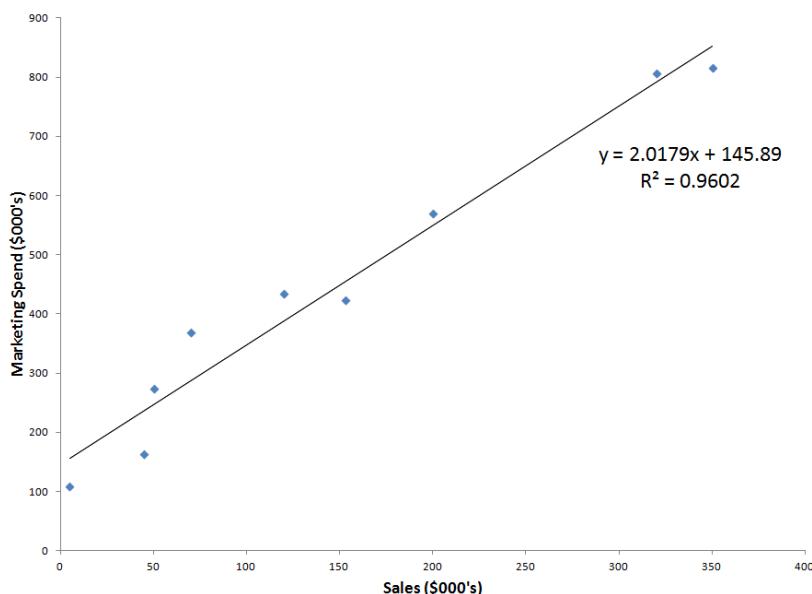
The output function, i.e. the model, is a straight line which approximates a linear relationship between the

dependent and independent variables.

Simple Linear Regression includes one and only one independent variable to predict the dependent variable. **Multiple Linear Regression** takes more than one independent variable to predict the dependent variable.

An important reference statistic to assess overall model performance is given by the (R Square) statistic. This reflects the proportion of the dependent variable explained by the model.

Figure 2.8: Linear Regression Model



Logistic Regression

Logistic regression is a modelling technique used for predicting the outcome of a categorical dependent variable.

Frequently the dependent variable is binary with two available categories: 0/1, Bad/Good, Yes/No. For example, a gold card holder can be represented as 1, and a non-gold card holder can be represented by 0. In this case, a logistic regression model will provide an estimate of the probability that a new customer will become a gold card holder.

This type of logistic regression is called binary logistic regression as there are two categories of the dependent variable. Problems with more than two dependent variable categories, for example low/medium/high can be dealt with using **Multinomial Logistic Regression**.

Logistic Regression measures the relationship between a categorical dependent variable and one or more independent variables, which are usually, but not necessarily, continuous. However, **Logistic Regression** will generally perform better when the independent variables are continuous.

Logistic Regression treats categorical independent variables in the same fashion as **Linear Regression**. From a statistical perspective, **Logistic Regression** is an extension of **Linear Regression** to represent the case of a discrete dependent variable.

Here there is interest in the odds of being in one category as opposed to the other. This can be framed as a ratio of probabilities; where p is the probability of being in one category, generally the category of interest, and $1 - p$ is the probability of being in the other category.

In the equation, Y , represents the *logistic function* which is the natural logarithm of the odds: *In(odds)*:

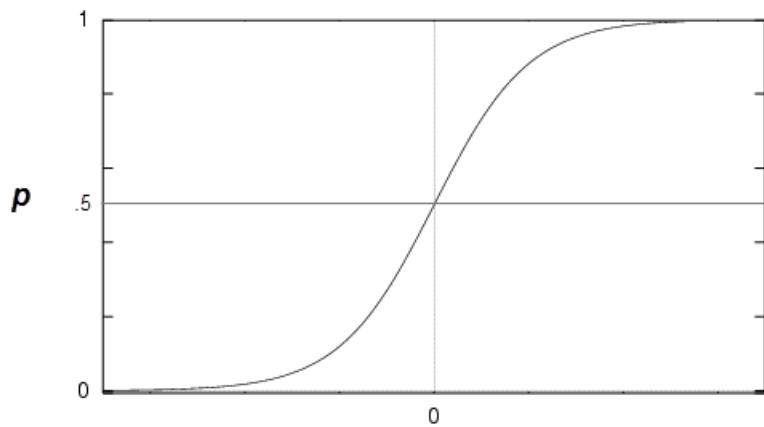
$$Y = \ln \left(\frac{p}{1-p} \right) = a + bx \quad (2.1)$$

From the above equation the probability of the DV being in the category of interest is given by the equation:

$$p = \frac{e^{(a+bx)}}{1 + e^{(a+bx)}} \quad (2.2)$$

The above equation is an s-shaped function called the *Sigmoid function*. A common and useful application of **Binary Logistic Regression** is when building credit scorecards, as **Binary Logistic Regression** is one of the key steps.

Figure 2.9: Sigmoid Function



Neural Networks

Neural Networks, (NN), are predictive models, inspired by the central nervous system, that are capable of machine learning and pattern recognition. They are usually presented as systems of interconnected *neurons* that can compute values from inputs by feeding information through the network.

The **Multi-Layered Neural Network**, (MLNN), is a simple feed-forward network with three layers.

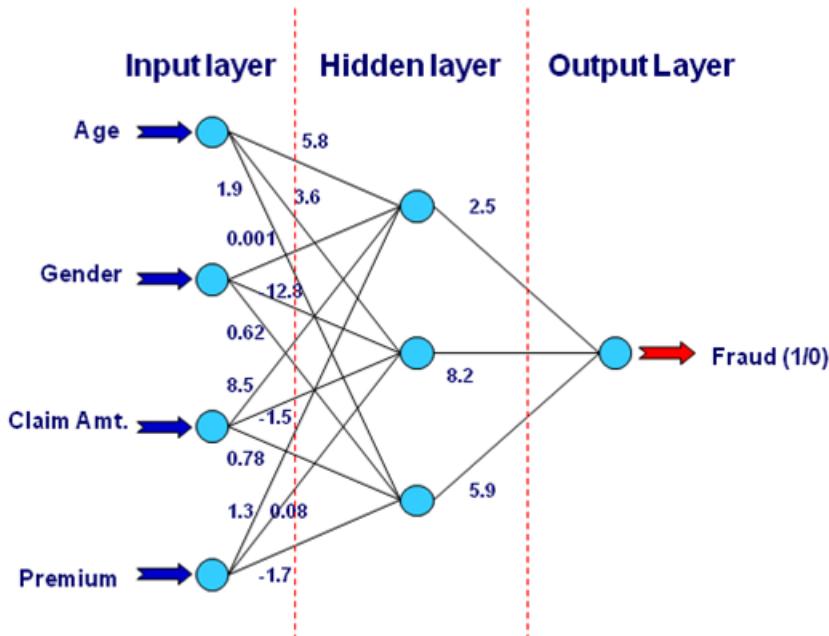
The neurons of the input layer receive the inputs, they do not change the input signals. Their activation function is a simple linear function with a slope of one. The output layer neuron(s) receive their inputs from the hidden layer(s).

It has been shown in literature that such a simple network can approximate any function to any required degree of accuracy. The training algorithm of the MLNN normalises all input signals to the range (0, 1). MLNN networks can be used in most tasks for both continuous and categorical dependent variables and independent variables.

Therefore, the dependent variable can be either discrete/categorical such as purchase; Y/N, or continuous such as sales revenue.

Neural Networks are very complex and can be difficult to interpret. Sometimes, to aid understanding, it is necessary to use another model (for example a decision tree) prior to modelling in order to eliminate less predictive independent variables, or post modelling to explain how the network has made its decisions. **Neural Networks** are popular for modelling fraud.

Figure 2.10: Multi-layered Neural Network



Scorecards

A **Scorecard** is a type of predictive model that sums points associated with variable characteristics to produce a score. **Scorecards** are usually used by lenders to support credit decisions such as scoring new lending

applications, changes in credit limits, over-limit approvals on transactions, etc.

Scorecards are used as a common format to build credit risk models as they are easy to understand, manage and deploy, they allow straightforward communication between stakeholders and they provide simple diagnosis and monitoring. Scorecards offer great flexibility to adapt data to linear modeling assumptions.

Scorecards in general come in two flavours; **Application or Behavioural Scorecards**. The former are used to assess whether to accept or reject applications for a service or product. The latter are useful predictive models in account management and collections, where scorecards can be combined with probability-based models.

In figure 2.11, a simple two-variable scorecard is illustrated.

Figure 2.11: Credit Scorecard Based on Two Variables

Residential Status	Live with Parents	
	Other	66
	Own Home	70
	Rent	39
Years at Address	0 to <1	51
	1 to <3	56
	3 to <5	58
	5 to <10	66
	>=10	71

Using the values, renting and living at current address less than 1 year, the scorecards points total:

$$39 + 51 = 90 \quad (2.3)$$

Similarly, scores can be calculated for any combination of variable values. In order to provide credit based on a scorecard, a threshold score can be agreed upon. Applicants with scores lower than the threshold are rejected and above are accepted.

2.7.1 Clustering Techniques

Clustering is the process of creating homogeneous segments. For example, clusters can be based on the buying habits of customers.

Clustering is an unsupervised data mining technique as there is no dependent variable involved in the model – all variables are considered as independent variables (*I/V's*).

Cluster Analysis

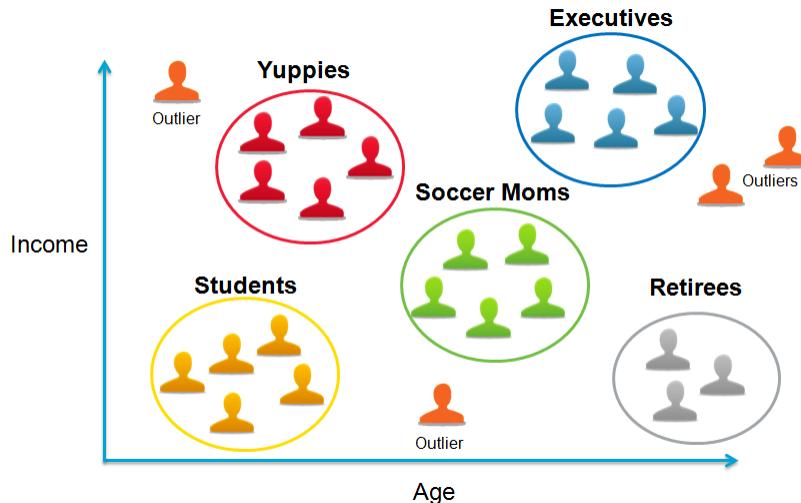
Cluster Analysis is the most common clustering technique. Although, it works best with scale/continuous variables, it can also include categorical variables.

In Altair KnowledgeSEEKER or KnowledgeSTUDIO, clusters are identified by using one of two available algorithms: **K-Means** or **Expectation Maximisation**.

Cluster Analysis is an appropriate technique for customer segmentation, product targeting, building credit behaviour segments and product purchase segments.

An example of customer segmentation is illustrated in figure 2.12. Each point on the scatter plot represents a customer in terms of age and income. The results identify five different segments. In addition, some data points which have extreme values may be interpreted as outliers.

Figure 2.12: Customer Segmentation



The clustering algorithm finds clusters in the data and labels each record with the cluster that it belongs to. Based on the understanding of the meaning of the variables that characterize each cluster an analyst would assign a name or a meaning to each of these clusters.

For example, the cluster in the lower left corner with younger *age* and lower *income* may be assigned a label of *Students*, or the cluster with younger *age* and higher *income* may be thought of as *Yuppies*. These labels which describe or summarize the characteristics of the cluster can be used to tailor the product and services offered to each segment.

Customer segmentation is applied in practice with two objectives:

- Segment the customer base into smaller groups to better target these segments
- Generate an index; cluster number or label, to use in further modelling or exploration

Market Basket Analysis

Market Basket Analysis, (MBA), is an association technique with similarities to clustering, used for discovering associations between items and deriving rules that indicate the likelihood of items occurring together in groups of specific types. *MBA* is generally used when there is interest in promoting other products and services as a next best product.

A typical *MBA* problem is to determine which products or product categories are likely to be purchased together. The resulting association rules can be used to build strategies for product promotions, product placement, cross-sell, and so on.

Other possible applications of association rules include health sciences, fraud detection, and many other areas where identifying patterns of events or behaviour from transactional data is required.

MBA models work by analysing the contents of sales i.e. baskets, or groups of products purchased together.

The *MBA* algorithm extracts rules in the form:

if(A and B), then C

Rules are then used to score customers based on previous purchases to recommend products they are likely to purchase. For example *Amazon's Customers Who Bought This Item Also Bought...*

Another use of *MBA* is to find product bundles. Product bundles are a by-product of *MBA*. For example, *MBA* may find that products *A*, *B* and *C* are sold together in a high volume. This suggests that bundling these products would be welcomed by customers.

Figure 2.13: Product Recommendation



2.8 Data Mining Solutions to Business Problems

Large volumes of available data have opened new opportunities and challenges to companies in getting better insights to customers' behaviour [6].

In general, *Data Mining* models may be used for prediction of future events, customer segmentation or to get better insights into the voice of the customer. A number of different model types are summarised, and some are further explained below:

- **Prediction**
 - Next Product/Recommendation
 - Customer Acquisition

- Upsell/Cross Sell
- Customer Retention/Loyalty/Churn
- Customer lifetime value
- Credit Risk
- Collections
- Fraud
- Forecasting
- **Segmentation**
 - Customer Segmentation
 - Product Segmentation
- **Text Analytics**
 - Voice of the Customer
 - Leveraging social media/text response in reporting and modelling

Next product recommendation aims to promote additional products to existing customers when the time is right. When a company has many products to offer they have to determine which of those should be offered to a customer based on the existing products the customer owns.

Customer acquisition is used to acquire new customers and increase market share, and often involves offering products to a large number of prospects. Figure 2.14, shows the possible acquisition models and the appropriate data mining techniques to be used.

Figure 2.14: Modelling Business Scenarios

Data Mining Model	Acquisition Model		
	Propensity	Revenue	Time to Respond
Decision Trees	✓	✓	✓
Neural Networks	✓	✓	✓
Linear Regression	✗	✓	✓
Logistic Regression	✓	✓	✗

Upsell and **Cross-sell** models aim to provide existing customers with additional or more valued products.

Upsell is the practice of selling more expensive products, upgrades or add-ons to an existing customer.
Cross-sell is the practice of selling additional products to existing customers.

Appropriate data mining techniques for these models are **Decision Trees**, **Logistic Regression**, **Market Basket Analysis** and **Neural Networks**.

Customer Retention strategies and **Churn** models aim at maintaining and rewarding customer loyalty. In the case of Churn, the focus is on customers who will cancel a product within a certain time frame. There are 4 types of churn:

- *Customer Churn* - customers that are leaving
- *Product Churn* - customer product cancellations
- *Downgrading* - customers who reduce their level of product usage
- *Product Replacement* - customers who replace one product with another

Customer Lifetime Value represents the expected revenue that is to be earned from a customer over their lifetime considering all of the possible products that this customer could purchase.

Customer Lifetime Value can also represent an index that represents such expected revenue. Figure 2.15 shows the possible acquisition models and appropriate data mining techniques to be used.

Figure 2.15: Modelling Scenarios

Acquisition Model		
Data Mining Model	Propensity	Revenue
Decision Trees**	✓	✓
Linear Regression	✓	✗
Logistic Regression	✗	✓
Neural Networks*	✓	✓

*more accurate but difficult to interpret

**most popular

Customer Segmentation allows better understanding of the market landscape in terms of customer characteristics and whether they can naturally be grouped into segments that have something in common.

A common *Data Mining* technique for this purpose is cluster analysis. The model output is a set of clusters that can be used as additional inputs into other models such as **Decision Trees**, **Linear Regression**, **Logistic Regression** and **Neural Networks**.

Product Segmentation allows recommendations of product bundles using product affinity, in most cases using **Market Basket Analysis**.

Voice of the Customer using **Text Analytics** allows for the analysis of text-based data sources and turns unstructured data into structured fields containing the *Entities*, *Themes* and *Topics* that customers are talking about as well as the *Sentiment*, a positive or negative score, associated with these.

2.9 Conclusion

With the advent of increasing amounts of data and the increased adoption of statistics and model building in the business world, the traditional statistical approach to model building underwent a rebirth of sorts, and data mining emerged.

The progress of computer technology provided a means to process larger amounts of data, faster, using more complex manipulations and methodologies.

As a result of completing this chapter attendees should be familiar with the following:

- The concept of *Data Mining*
- **CRISP-DM**; a set of best practice steps to guide through any data mining project
- Various techniques applicable to different data mining situations to address varying business problems

References

- [1] Altair Software Ltd. <http://www.altair.com>
- [2] Han, J. (2011) "Data Mining: Concepts and Techniques", Morgan Kaufmann
- [3] Provost, F, Fawcett, T. (2013) "Data Science for Business: What you need to know about Data Mining and data-analytic thinking", O'Reilly Media
- [4] CRISP-DM, available online at: <ftp://ftp.software.ibm.com/software/analytics/spss/support/Modeler/Documentation/14/UserManual/CRISP-DM.pdf>
- [5] Giberta, K., Sàncchez-Marrèa, M., Codinaa, V. (2010) "Choosing the Right Data Mining Technique: Classification of Methods and Intelligent Recommendation", International Congress on Environmental Modelling and Software Modelling for Environment's Sake
- [6] Altair Customer Analytics Roadmap series, available online at: https://www.youtube.com/watch?v=Z_HPFe_hXSY

Chapter 3: Introduction to KnowledgeSTUDIO

3.1 Introduction

The aim of this chapter is to provide an overview of **KnowledgeSTUDIO** and its capabilities including:

- **Project Pane** and **Working Directory**
- **Menu System** and **Toolbar**
- Creating new projects
- **Workflows, Nodes and Palettes**
- Importing files and connecting nodes to create a simple **Workflow**
- Augmenting, extending and adding comments to **Workflows**

3.2 Starting KnowledgeSTUDIO

To start **KnowledgeSTUDIO**:

Click the Windows **Start** button and choose: **All Programs... KS Workstation 10.x**

Figure 3.1: Windows Start Menu

Alternatively, click the desktop shortcut:

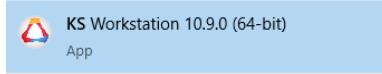
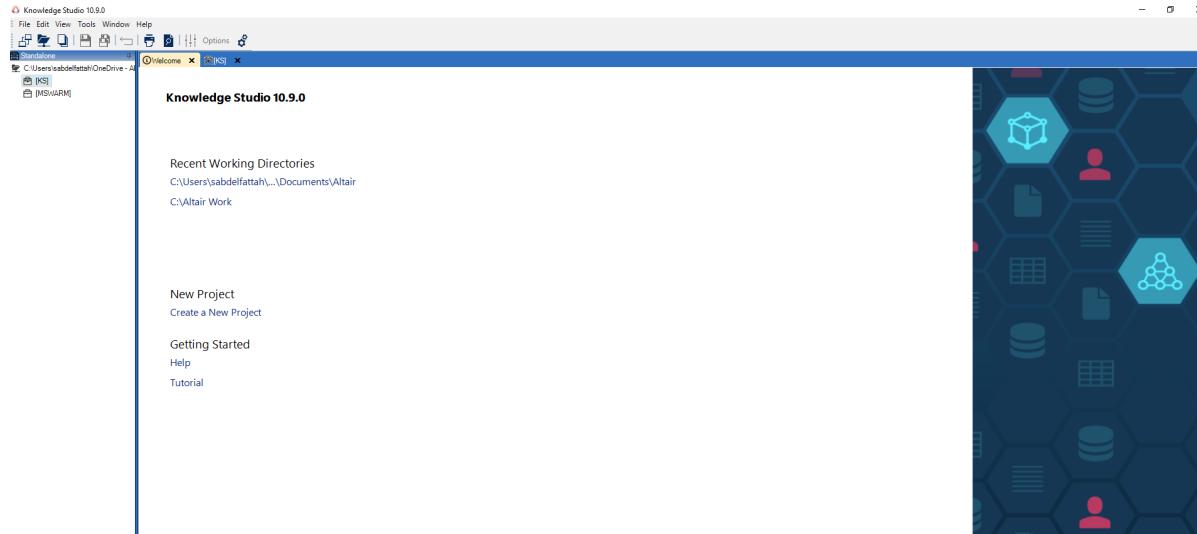


Figure 3.2: Desktop Icon



Once complete the initial product view opens with the **Project Pane** on the left hand side indicating the default **Working Directory** and the **Welcome to KnowledgeSTUDIO** screen on the right hand side.

Figure 3.3: Welcome Screen



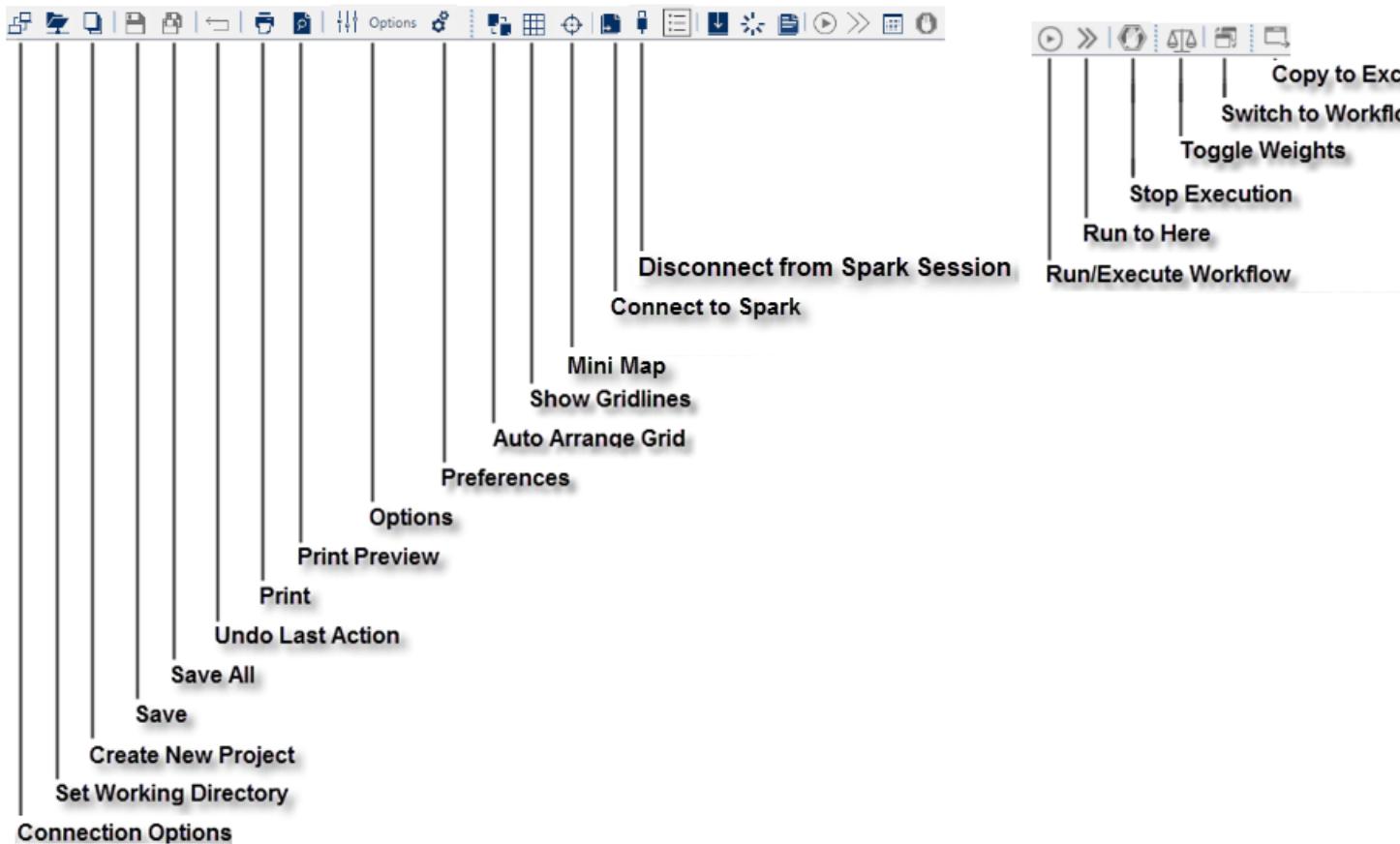
3.3 Menu System and Toolbar

KnowledgeSTUDIO provides eight menus with available options:

- **File** Local machine vs client/server connections, set the working directory, create projects, print output, access recent working directories, additional project options and exit the product
- **Edit** Standard editing options such as undo, copy, select all, delete and rename
- **View** Access to the activity and query logs
- **Insert** **Measures of Predictive Power**
- **Workflow** Create, rename and delete **workflows**
- **Tools** Access to various options and preferences within the product
- **Window** Modify view, for example; **Close All Except Active Document**
- **Help** Generic help, about and contact details. Contains tutorials as well as access to sample data used in tutorials. **License Manager** is also accessed from this menu.

The **Toolbar** contains icons that are shortcuts to frequently used menu options, specifically:

Figure 3.4: Toolbar Options

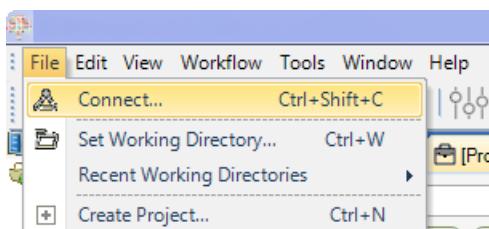


NOTE: Fig 3.4 shows the most commonly used icons. Additionally, some icons are only available within specific views.

3.4 Setting Connections and the Working Directory

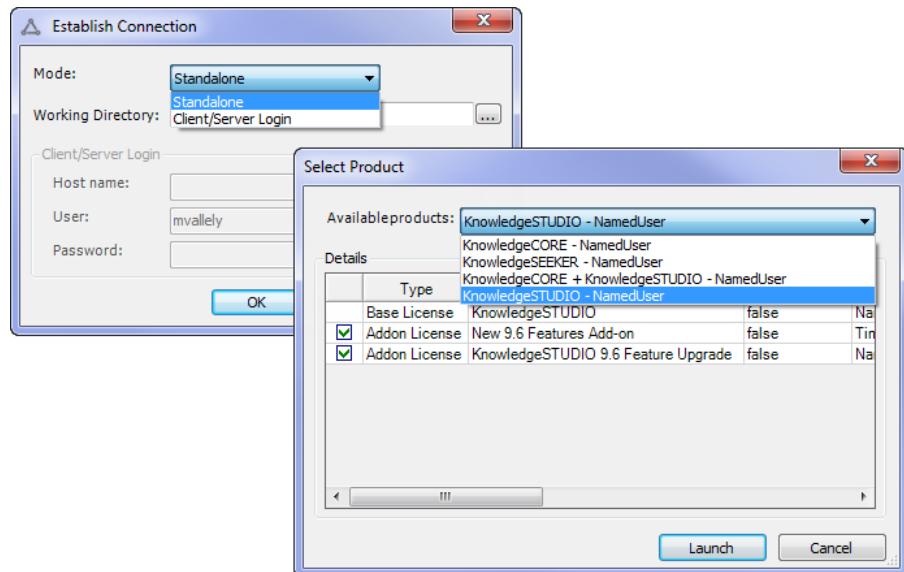
Prior to starting any *Data Mining* project, connections and the working directory must be specified correctly. Both are accessible from the **File** menu.

Figure 3.5: File menu; Connections and Working Directory Options



Use **Connect...** to specify local machine or client/server connections. Once complete an additional dialog appears requesting selection of the appropriate product to run.

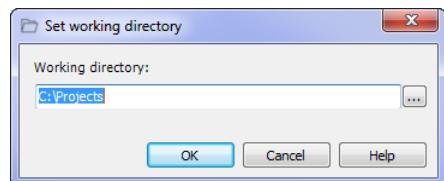
Figure 3.6: Connect Options



NOTE: If **Client/Server Login** is selected, an additional dialog prior to **Select Product** appears to enter server login details. Additionally the **Select Product** window may or may not contain more than one option depending on licensing.

The **Working Directory** is where all **Altair Projects** are stored. On first installation this is usually set to the **My Documents** folder. Set this to a specific location to more easily locate projects.

Figure 3.7: Set Working Directory



3.5 Projects

Once connections and the **Working Directory** are set, a project must be created. The project resides in the **Working Directory**.

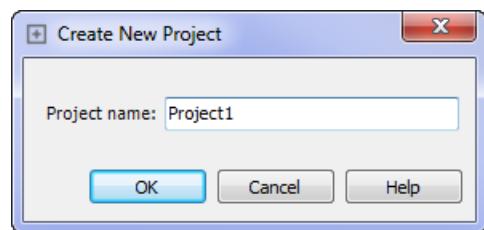
Creating a project is the first step before conducting any analysis in **KnowledgeSTUDIO**, and is a prerequisite to importing data.

A project is simply a folder in the file system where all project contents are stored. A benefit of this organizational structure is that the project can be backed up or moved by simply copying the entire project folder.

NOTE: Do not change or alter any files included in the project folder with any software other than **KnowledgeSTUDIO**, otherwise the project may not load properly.

To create a new project: from the **File** menu choose **Create Project** and assign the name **SeekerProject**.

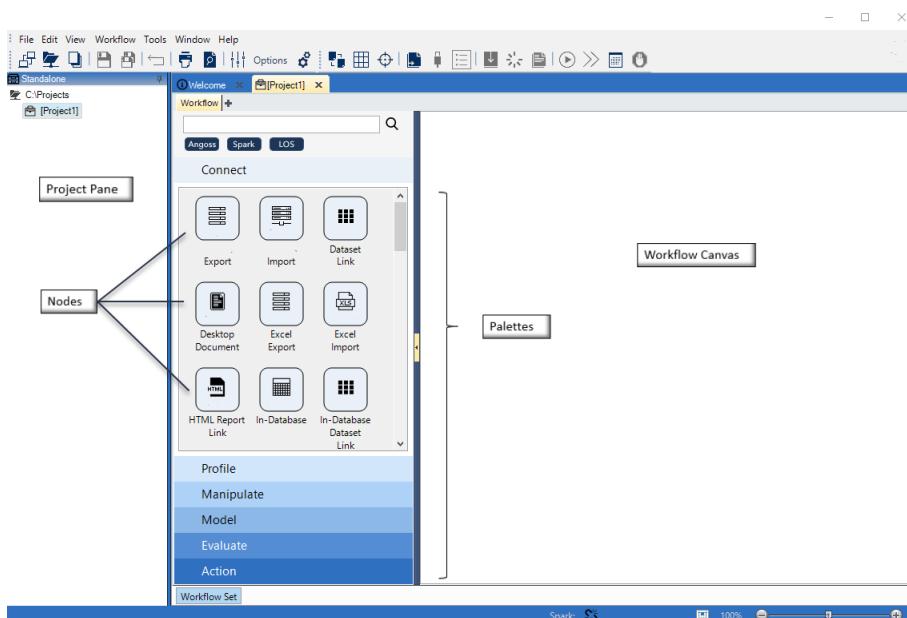
Figure 3.8: Create New Project



Once complete the new project appears in the **Project Pane** and initiates a blank **Workflow**.

The view contains the **Menu Bar**, **Toolbar**, the **Workflow Canvas** and a series of **Palettes** to the left hand side of the **Workflow** canvas, containing grouped colour coded process nodes.

Figure 3.9: Interface Components



NOTE: The project pane can be docked/undocked by clicking its associated pushpin icon .

3.6 Nodes and Palettes

There are six palettes of grouped colour coded processing nodes. The colour denotes the processing group the node belongs to. The processing capabilities of the various node groups are detailed in table 3.1.

Table 3.1: Node Group Descriptions

Node Group	Colour	Description
Connect		Nodes for importing and exporting from/to a variety of data and model sources
Profile		Nodes for profiling data
Manipulate		Dataset and record level operations such as joining, appending, de-duplicating, aggregation, new variable creation
Model		Modeling nodes
Evaluate		Nodes for assessing models performance
Action	Dark Blue	Nodes for translating models to code. Documentation nodes, optimization, scheduling and coding nodes

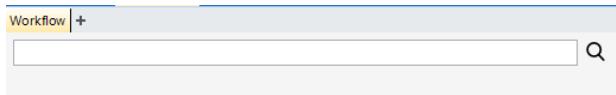
3.7 Filtering, Searching and Node Availability

3.7.1 Filtering

Nodes can be filtered to show only those nodes for working in a specific platform. Filtering and search options are available at the top of the palette section.

Filtering options for Altair suite of products include: **Python**, **R**, **Spark** and **LOS**.

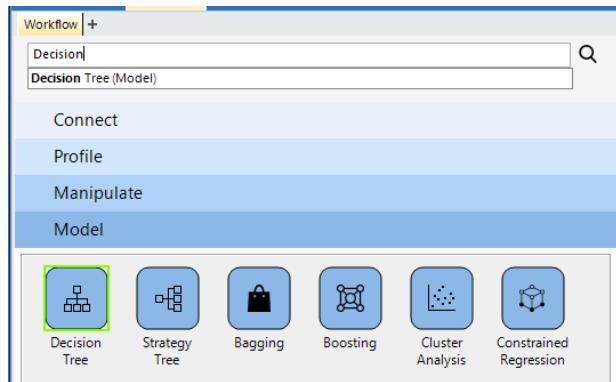
Figure 3.10: Filtering and Search Bar



3.7.2 Searching

Nodes can easily be sought using the search facility. Specifying the node name opens its associated palette and highlights the node.

Figure 3.11: Searching

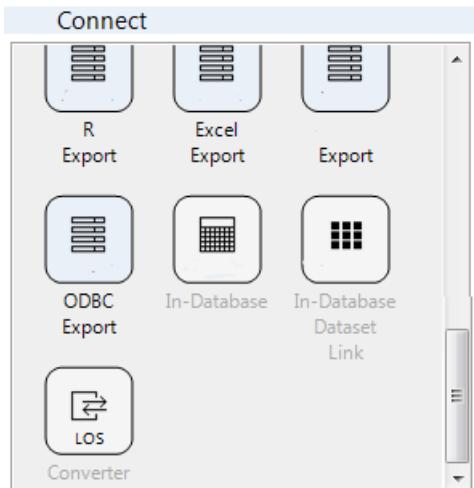


3.7.3 Node Availability

When a palette is selected, all nodes are visible however those that are not available for use are greyed out.

Figure 3.12 illustrates unavailable nodes from the **Connect** palette. Note that product license governs available nodes.

Figure 3.12: Unavailable Nodes



3.8 Creating Workflows

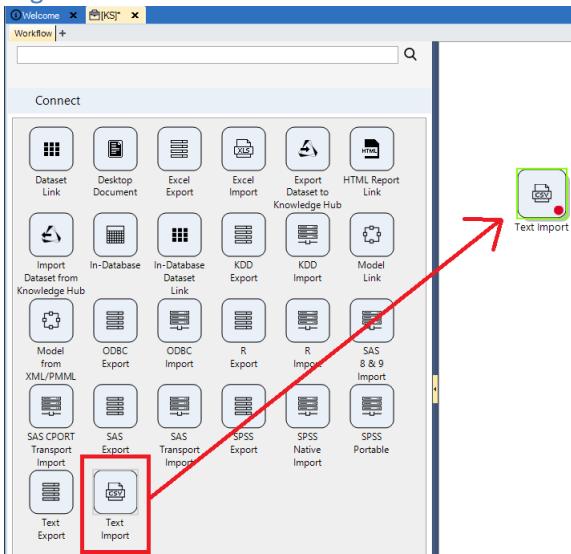
Workflows are a visual representation of data processing steps. They are created by dragging and dropping nodes from palettes and connecting together to form a process flow.

A **Workflow** must start with a **Connect** or **Link** node. Additional nodes can be connected to process the data further. **Workflows** can be complex, however they can be annotated for easy reference.

3.8.1 Adding Nodes

To import a text file; add a **Node** to a **Workflow**, click and drag the node from the **Connect** palette to the canvas.

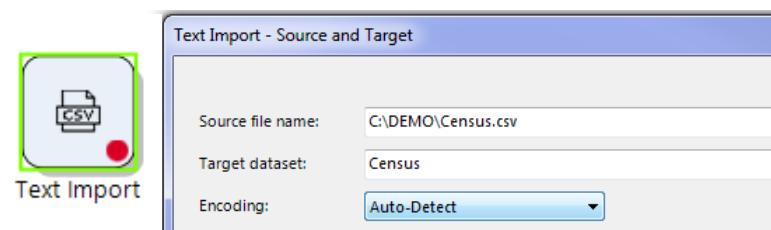
Figure 3.13: Connect Palette



Once placed on the canvas, an indicator appears at the bottom right corner of the node, at this point, as a red circle indicating that the node is as of yet **Undefined**.

Double click the node to access a wizard for options associated with the specific node; in figure 3.14 the wizard contains options for reading the specific file type. Clicking **Run** completes the process.

Figure 3.14: Text Import Node Options



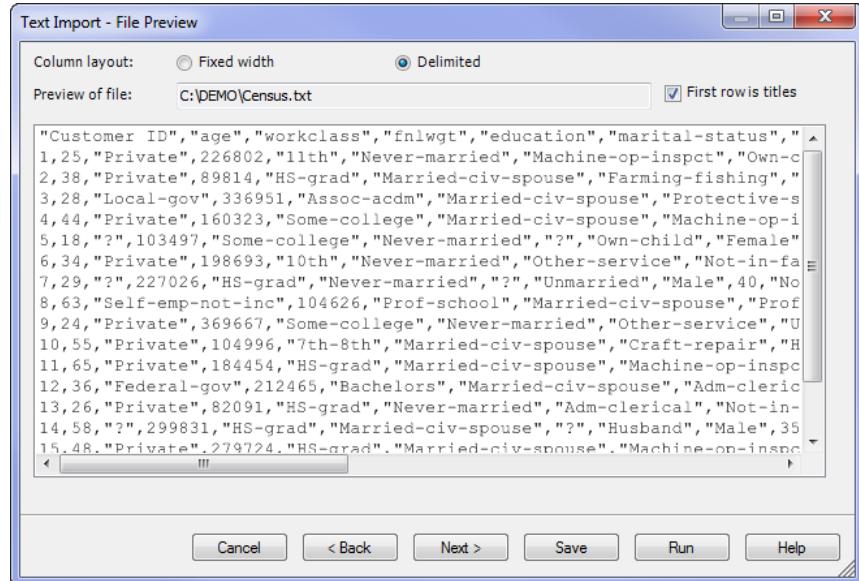
At the point of double clicking, a wizard opens. A number of dialogs are available to specify options:

- The first screen **Import - Source and Target**, as illustrated in figure 3.14, requires specification of the file location, the name to give to the target dataset. Additionally chose an encoding format, if known, otherwise leave at the default of **Auto-Detect**
- Click **Next>** to access the **Text Import - File Preview** dialog, this allows specification of whether the text file is *fixed width* or *delimited* and allows specification of column headers.

This dialog also provides a preview of the data when importing. Visually inspect to assess whether field

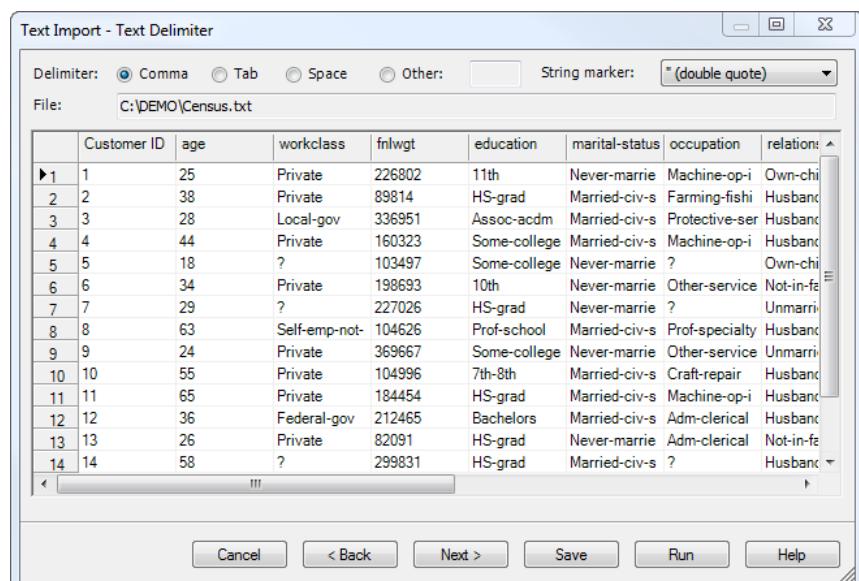
names are contained in the first row of data and whether the chosen formatting options are appropriate

Figure 3.15: Text Import - File Preview



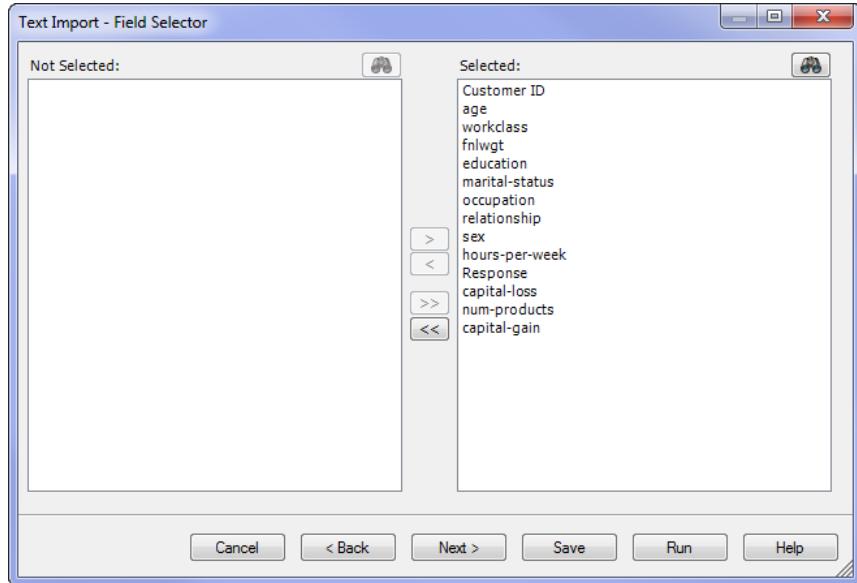
- Click **Next >** to access the **Text Import — Text Delimiter** dialog. Identify the file delimiter by visually inspecting the preview to assess whether the selected option is appropriate.

Figure 3.16: Text Import - Text Delimiter



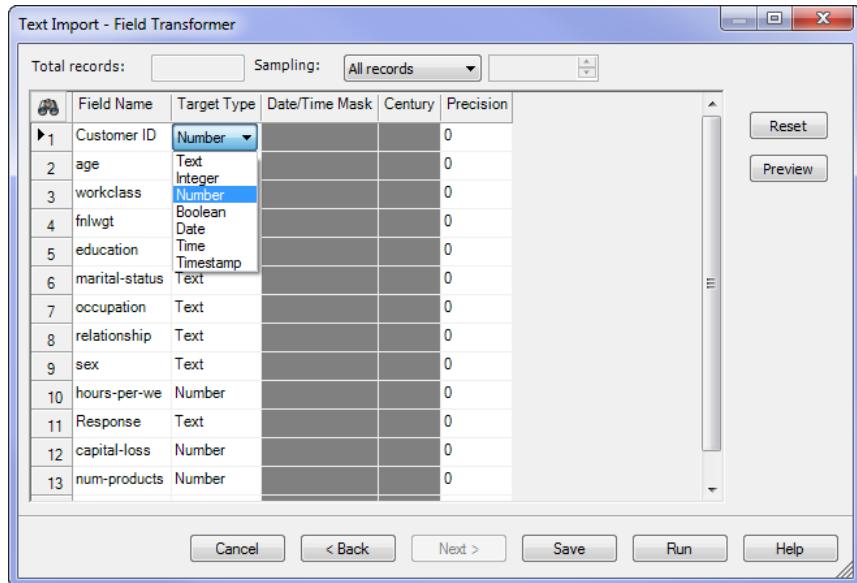
- Click **Next >** to step to the **Text Import - Field Selector**. Here, imported fields can be filtered.

Figure 3.17: Text Import - Field Selector



Click **Next>** to access the **Text Import - Field Transformer** dialog.

Figure 3.18: Text Import - Field Transformer



This dialog is used to assign variable type, precision and sampling options. Examine data using the **Preview** button to aid in correct assignment.

When importing from a text file, the first 30 observations from each field are used to determine data type. In most cases this is successful. However, it is advisable to sense check using the **Preview** button.

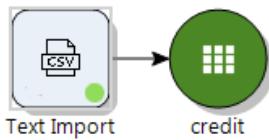
Set precision using the **Precision** column. Default is 0, meaning honour source data representations.

The **Sampling** dropdown menu allows sample selection. Options are: *All records*, *Top N records*, a specified *Number of records* or *Percentage of records*. Sampling does not speed up import as the entire file must be read prior to sampling.

Click **Run** and the data set is read into **KnowledgeSTUDIO**. The imported file appears in the **Project Pane** and is represented on the canvas as an oval, in the centre of which is the file name.

Additional process nodes from other palettes can now be added and connected to extend the process flow. Note also that the imported object appears on both the canvas and the **Project Pane** and that the node indicator now shows a green check mark.

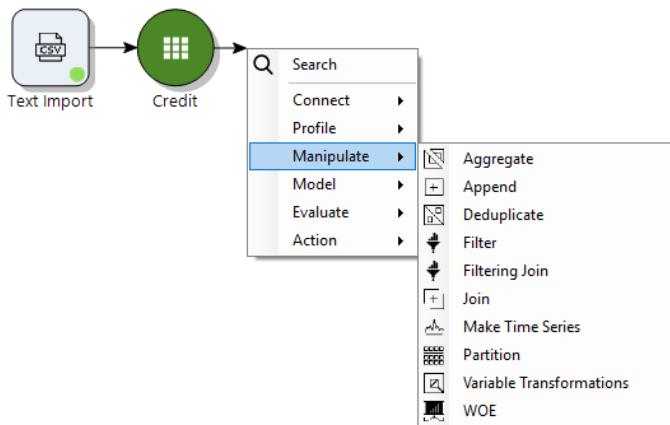
Figure 3.19: Completed Import Process



The **Workflow** can be extended by adding additional nodes in one of two ways:

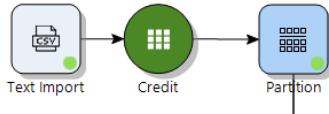
- Dragging nodes from an appropriate palette and connecting
- Hovering over any node until the hand icon appears 
 - Once visible, click and drag to an empty canvas space and release. A menu becomes available depicting all palettes. Select an appropriate node and it is added. Figure 3.20 depicts the addition of a **Partition** node

Figure 3.20: Extending Workflows



The node is added and automatically connected to the **Workflow**.

Figure 3.21: Extending Workflows Further



3.8.2 Node States

Node states are indicated by a small circle in the bottom right hand corner. Two states are possible:

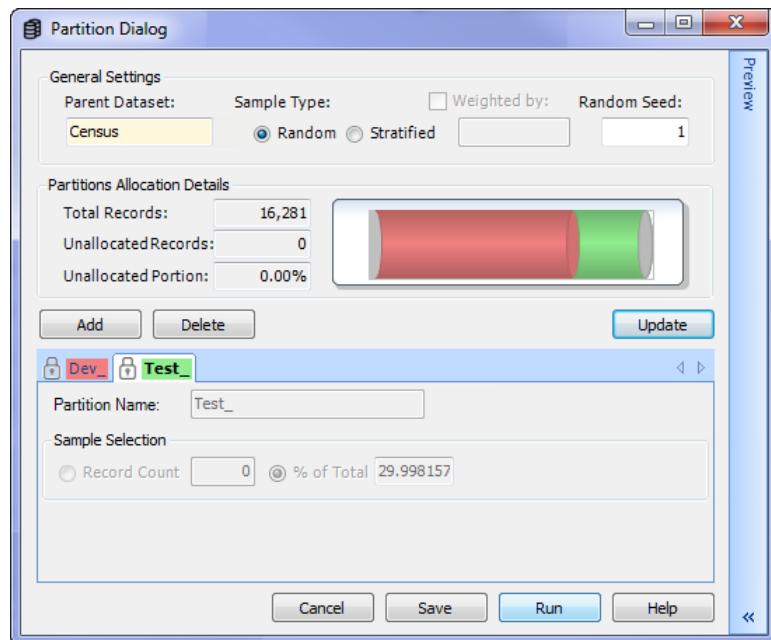
- **Unprocessed** A red circle indicates an **Unprocessed** state. The node remains as such if node options are modified and saved but not run
- **Processed** A green check mark indicates a **Processed** state

The **Partition** node state is **Unprocessed**. This means no options have been set for this node.

3.8.3 Accessing Node Options

Access node option either by double clicking the node or right clicking and selecting the option **Modify**. This opens the node dialog, here, the **Partition** node.

Figure 3.22: Accessing node Options

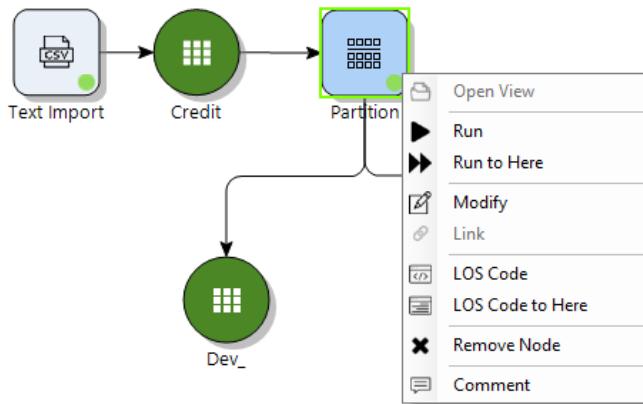


Modifications can be cancelled, saved or run. Choosing **Run** accepts node options and produces output.

Save retains modifications to run at a later time. If the option **Save** is selected, two partitions are created, but not populated.

To create the partitions, right click the **Partition** and select either **Run** or **Run to Here**.

Figure 3.23: Node Options



As can be seen, some options are available and some are greyed out. Table 3.2 describes options.

Table 3.2: Node Option Descriptions

Option	Description
Open View	Open view for object. Available for datasets and models
Run	Runs entire Workflow
Run to Here	Run Workflow to selected node
Modify	Opens options dialog related to selected node
Link	Reference project datasets. Only active when using Link node from the Manipulate palette
LOS Code	Generate <i>LOS</i> code for selected nodes from the Workflow
LOS Code to Here	Generate <i>LOS</i> code from start up to and including the selected node
Remove Node	Delete the selected node
Comment	Add a comment to the selected node
Link on another Workflow	Create a new Workflow with the selected node present

*Some options only display for dataset nodes

3.9 Additional Workflow Features

Additional **Workflow** features are detailed in table 3.3. Features are found either as a task bar icon or by selecting nodes, or an entire **Workflow**, and right clicking.

Table 3.3: Additional Workflow Features

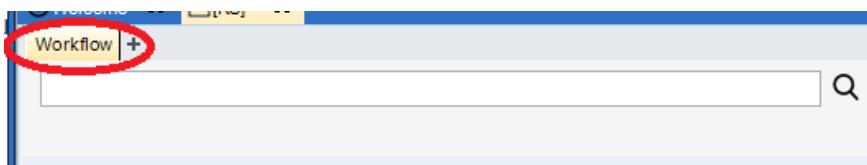
Option	Description
Auto-Arrange	Automatically moves all nodes in the Workflow to align them horizontally with minimal space between them (Taskbar icon)
Show Grid and Auto-Snap	Toggle Workflow grid. Any nodes added when toggled on are auto-snapped to align to the grid (Taskbar icon)
Mini Map	View map of current Workflow
Undo Last Action	Undo the last Workflow configuration change. (Taskbar icon or ctrl-z)
Copy Layout	Copy and paste existing Workflow or set of nodes to a new Workflow or already existing Workflow . (Available through right click)
Cut Nodes	Delete selected nodes. (Available through right click)
Align Horizontally	Align selected nodes horizontally. (Available through right click)
Align Vertically	Align selected nodes vertically. (Available through right click)
Assess Connections	Although not available as an option, hovering over any node will highlight its connections

3.9.1 Complex Workflows

Workflows can become complex and sometimes additional nodes may clutter or distort the **Workflow**.

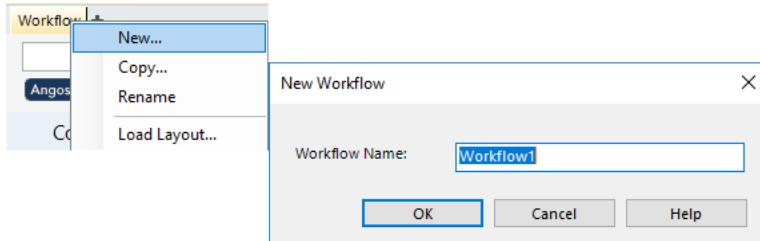
Thankfully multiple **Workflows** can exist in the same project enabling processing sections to be isolated for clarity but also connected! All **Workflows** are visible at the top of the project.

Figure 3.24: Project Workflows



Adding a new **Workflow** is straightforward: right click an existing **Workflow** tab and choose: **New...**

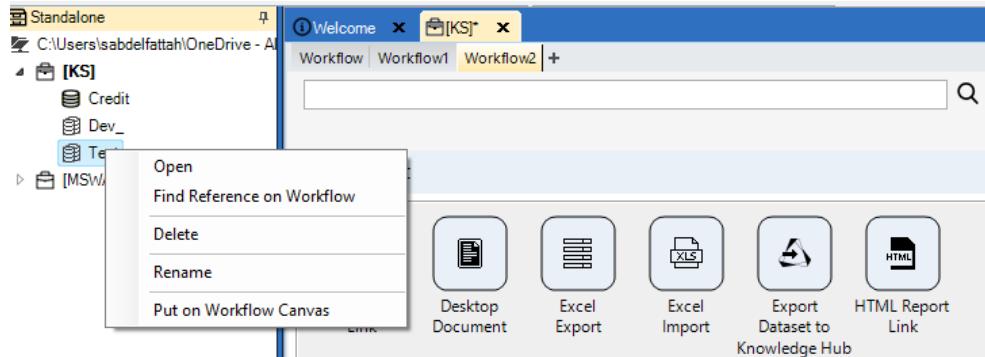
Figure 3.25: Additional Workflows



This initiates a dialog prompting for a name for the new **Workflow**. Once added, data and model results can be communicated between **Workflows** using either:

- **Dataset Link** and **Model Link** nodes. These are found in the **Connect** palette
- Right clicking the dataset or model in the **Project Pane** and selecting the option: **Put on Workflow Canvas**

Figure 3.26: Communicating Data and Models Across Workflows



These nodes operate in such a way as to allow data or model results to be referenced multiple times.

To further enhance the **Workflow** visual audit trail, comments can be added to any node. This is accomplished by right clicking any node and selecting the option; **Comment**.

3.10 Help Options

Altair provides many help options. In general these can be spoken of in three ways:

- **Generic Help** - available from the Help menu by selecting the option: **Help**.

Figure 3.27: Generic Help

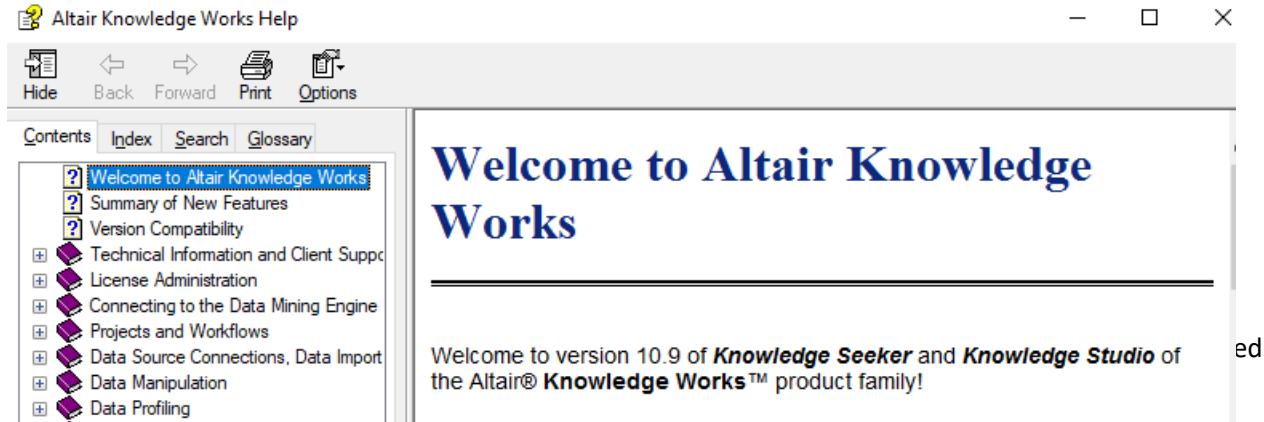
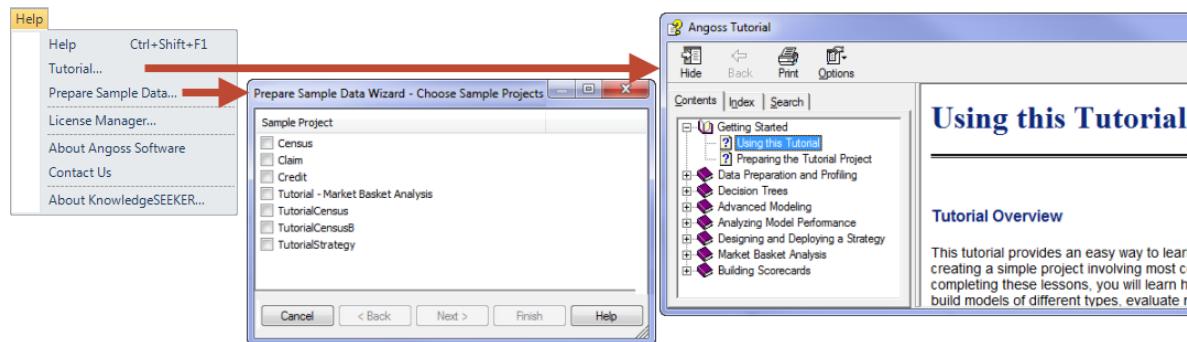
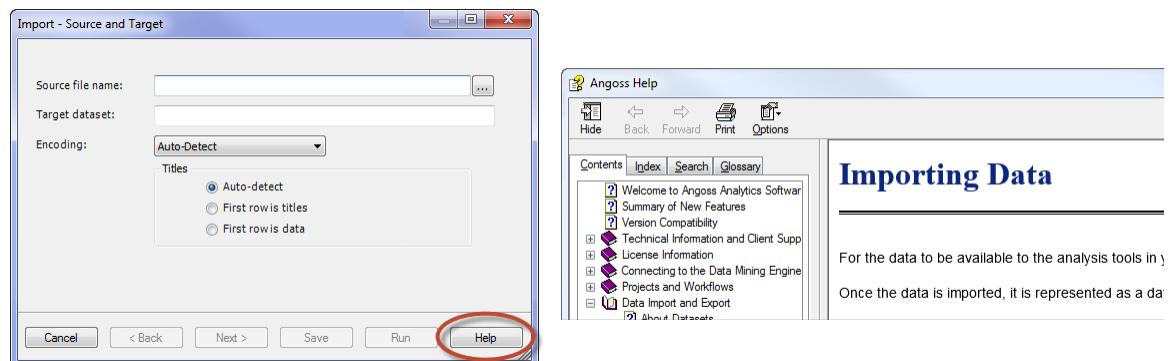


Figure 3.28: Tutorials And Prepare Sample Data



- **Context Sensitive Help.** This option is available from within dialogs, and once clicked opens up the Help pages for that specific dialog.

Figure 3.29: Context Sensitive Help for Import Node Options



3.11 Conclusion

This chapter introduced **KnowledgeSTUDIO** and its capabilities. On completion of this chapter users should be familiar with:

- **Project Pane and Working Directory**
- **Menu System and Toolbar**
- Creating new projects
- **Workflows, Nodes and Palettes**
- Importing files and connecting nodes to create a simple **Workflow**
- Augmenting, extending and adding comments to **Workflows**

Exercises

1. Start **KnowledgeSTUDIO**
2. Using the **File** menu
 - (a) Set an appropriate **Working Directory** to store projects. This can be any location of your choosing
 - (b) Create a new **Project** and give it a name of your choosing
3. Explore the different node palettes and become familiar with the nodes contained therein
4. To become familiar with the various import node procedures, import the following files using the appropriate node from the **Connect** palette:
 - (a) *Census.xlsx*
 - (b) *Census.txt*
 - (c) *Census.csv*
5. Right click the **Excel Import** node and explore available options
6. Add some other nodes and connect to form an extended **Workflow**
7. Explore the filtering and search options
 - (a) Filter to show only **Altair** nodes
 - (b) Search for the **Decision Tree** node
8. Open the **Help** files and locate information on the topics:
 - (a) **Working with projects**
 - (b) **Importing Data (Overview)**

NOTE: These can be found from the **Contents** tab.
9. Access **Context Sensitive Help** for the **Import Node**

Chapter 4: Data Exploration and Profiling

4.1 Introduction

Data exploration, as a preliminary to further modelling or simply to attempt to better understand data, is an essential and integral part of *Data Mining*.

Exploration of data involves examining the characteristics and distributions of the dataset fields to identify potential issues and ultimately important factors and patterns that distinguish the data.

KnowledegSTUDIO provides an array of tools to explore and profile data. This chapter outlines how to utilize and understand core concepts in relation to exploring and profiling data using **KnowledegSTUDIO**.

As a result of completing this chapter, users should be familiar with and be able to use the following native **Altair** profiling features:

- Univariate statistics using the **Overview Report**
- Graphic representations using **Dataset Charts**
- Using the **Data tab** to view a spreadsheet style display of the data
- Identifying potentially good predictors of a target variable using the **Segment Viewer**, **Measure of Predictive Power** and the **Variable Selection** node
- Using and understanding correlations and crosstabulations including **KnowledegSTUDIO Characteristic Analysis**

These aspects are not only used to explore a dataset but also as a preliminary set of tools to identify potentially good predictors of an outcome, target, or **Dependent Variable**, prior to modelling.

4.2 Data

The dataset used in all demonstrations from this point onwards is an excel file; *Census.xlsx*. This file is based on a set of 16,281 records. The *Data Mining* steps and modelling in following chapters are tuned to marketing operations.

Fields included in this dataset are:

- Demographic information such as *age*, *education-level*, *marital-status* and *sex*
- Some financial information such as *capital-gain* and *capital-loss*
- The **Dependent Variable** is the variable *Response*, with categories *Yes* and *No*. *Response* records whether a campaign has been responded to in the past or not

4.3 Data Profiling with KnowledgeSTUDIO

KnowledgeSTUDIO provides the following data profiling features accessed through tabs from any dataset:

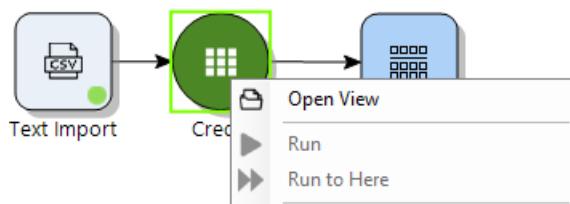
• Overview Report	Data structure and univariate summary statistics
• Dataset Charts	Graphic representation of each variable
• Data	Spreadsheet style display of the raw data
• Segment Viewer	Data visualizations segmented by a variable of interest
• Cross Tabs	Cross-tabulations
• Characteristic Analysis	Extends cross-tabulations to include binned continuous variables
• Correlations	Four correlation statistics to assess variable associations
• Saved Charts	Store and organize charts of interest

To work with a dataset, it must exist in the project. Once an appropriate **Working Directory** has been specified and a **Project** created:

- Drag an **Excel Import** node from the **Source** palette onto the canvas, locate the source data: *Census.xlsx*, and accept the default **Target Dataset** name assigned.

Once the file is successfully imported, profiling features can be accessed through either double clicking, or right clicking and selecting the **Open View** option as illustrated in figure 4.1

Figure 4.1: Open View



The **Dataset View** contains a number of tabs related to the profiling features outlined previously and, by default, opens on the first tab, the **Overview Report** tab.

Figure 4.2: Data View

Screenshot of the KnowledgeSTUDIO Data View interface showing the [Project1].[Census] dataset.

#		Field Name	Field Label	Data Type	Cardinality	Unique Count	# of Missing Values	% of Missing Values	Minimum	Maximum
►	1	Customer ID	Customer ID	Number	16281					
	2	age	age	Number	73					
	3	workclass	workclass	String	9					
	4	fnlwgt	fnlwgt	Number	12787					
	5	education	education	String	16					
	6	marital-status	marital-status	String	7					
	7	occupation	occupation	String	15					
	8	relationship	relationship	String	6					
	9	sex	sex	String	2					
	10	hours-per-week	hours-per-week	Number	89					
	11	Response	Response	String	2					
	12	capital-loss	capital-loss	Number	82					
	13	num-products	num-products	Number	16					
	14	capital-gain	capital-gain	Number	110					

Below the table:

- Calculate (button)
- Calculate All (button)
- Dataset: [Project1].[Census]
- Weight: [No weight]
- Records: 16,281
- Overview Report (selected tab)
- Dataset Chart
- Data
- Segment Viewer
- Cross Tabs
- Characteristic Analysis
- Correlations
- Saved Charts

The following sections describe each of these tabs in turn.

4.4 Overview Report

The **Overview Report** describes the number of imported records and fields, and lists the field name along with **Data Type** and **Cardinality**.

Default summary statistics are; unique count, number and % of missing values, minimum, maximum, mean and standard deviation and are calculated for each field by clicking **Calculate All**.

To calculate statistics for user selected fields: select fields to assess and click **Calculate**.

The **Options** button from the **Taskbar** provides additional summary statistics. Available statistics are grouped into eight expandable sections:

- **Basic Measures**
- **Measures of Central Tendency**
- **Measures of Dispersion**
- **Quantiles**
- **Extreme Observations**
- **Tests for Normality**
- **Tests for Location**
- **Confidence Intervals for Population Mean**

A shaded indicator means that some measures from that group have been selected. Expand to view and select items from each group.

NOTE: Appendix section describes available statistics.

4.5 Dataset Chart Tab

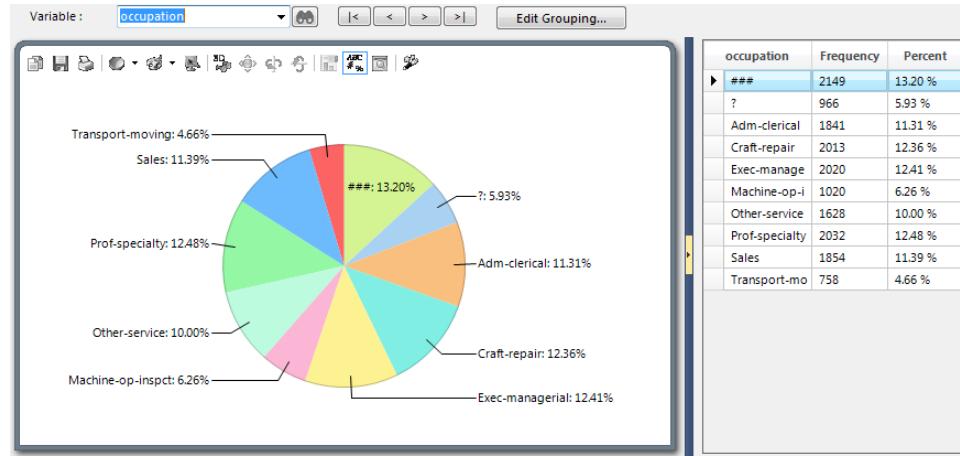
The **Dataset Chart** tab is used to visualize dataset variables. Visualizations provide a rich source of detail and can be used to become familiar with data quickly. Common uses include:

- Categorical variables:
 - Describe the distribution of the data
 - Identify categories that are potential candidates for merging.
 - Identify categories that might dominate a distribution, such as missing values which can then be substituted or removed
- Continuous variables:
 - Describe and identify the distribution of the data
 - Identify the degree of skew, and determine if corrective action is necessary

By default, continuous variables are displayed using histograms and categorical variables using pie charts.

Numeric variables with less than 10 entries are treated as categorical by default. Variables can be viewed by selecting via a dropdown or iterating through variables using the navigation buttons.

Figure 4.3: Dataset Chart Tab

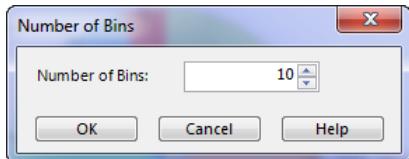


Ten evenly-spaced bins are used to construct histograms and ten categories for pie charts.

NOTE: To the right of the dropdown list is a binoculars icon. Use this feature to quickly search for variables. Clicking this button will launch a variable list with a **Find...** dialog. This feature is available throughout KnowledgeSTUDIO and not exclusive to the **Dataset Chart** tab.

Change bins for categorical variables by clicking the **Edit Grouping...** button and input a value.

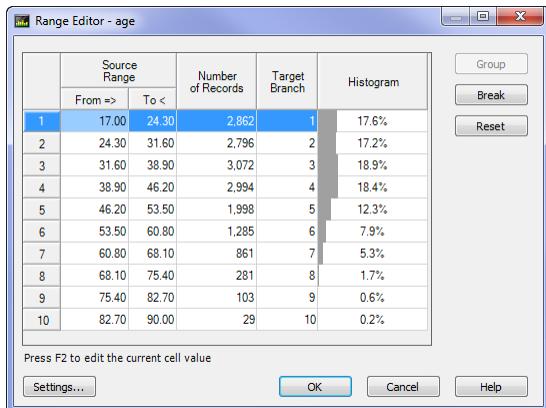
Figure 4.4: Variable Grouping



This feature is useful when there are more than ten groups. Since, by default, the number of categories to display is set to ten, if a variable has more than ten categories the nine most populated categories are displayed, and the remaining observations will be included in a single category labelled as ####.

For numeric data, the bin ranges of histograms can be modified using the **Edit Ranges...** button. To change the appearance of a histogram, click **Edit Ranges...**. The **Range Editor** opens.

Figure 4.5: Range Editor



	Source Range		Number of Records	Target Branch	Histogram
	From =>	To <			
1	17.00	24.30	2,862	1	17.6%
2	24.30	31.60	2,796	2	17.2%
3	31.60	38.90	3,072	3	18.9%
4	38.90	46.20	2,994	4	18.4%
5	46.20	53.50	1,998	5	12.3%
6	53.50	60.80	1,285	6	7.9%
7	60.80	68.10	861	7	5.3%
8	68.10	75.40	281	8	1.7%
9	75.40	82.70	103	9	0.6%
10	82.70	90.00	29	10	0.2%

Press F2 to edit the current cell value

Settings... OK Cancel Help

The **Range Editor** is used to define variable binning. Manually input values using columns: **From =>** & **To <**. Additionally, bins can be defined using **Group** to merge across ranges and **Break** to split at a specific value.

Options to customize any graph are available by right-clicking. Additional customization options are available by selecting **Properties**.

4.6 Data Tab

A spreadsheet style display of the data. Columns represent dataset fields and rows are records.

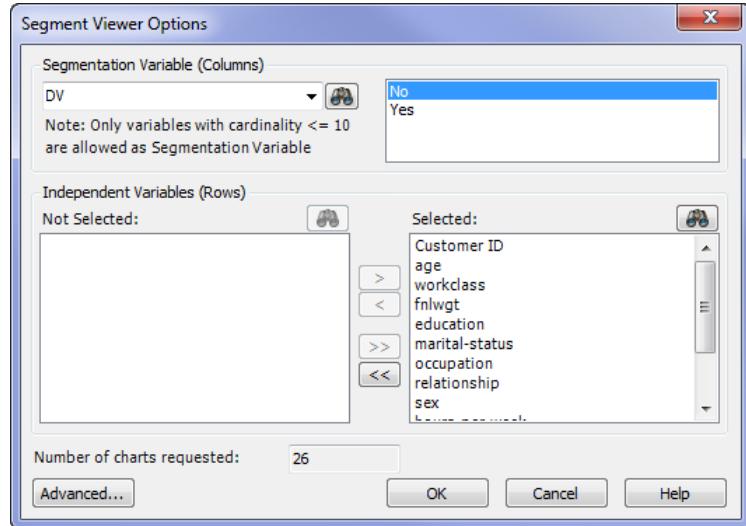
Data cannot be edited. Only the first 100 observations are initially displayed. The **Options** button provides additional row and column options. Click **Show more...** to increase no. of records displayed.

4.7 Segment Viewer Tab

The **Segment Viewer** charts a set of **Independent Variables**, segmented by the categories of a selected **Segmentation Variable**.

Clicking the **Segment Viewer** tab launches the **Options** dialog as illustrated in figure 4.6.

Figure 4.6: Segment Viewer Options



The **Advanced...** button allows customization of chart binning.

Results are displayed in figure 4.7. There are three columns, one representing the distribution for all cases for each variable, and one for each category of the segmentation variable.

Figure 4.7: Segment Viewer



To aid analysis and visualization use **Taskbar** buttons:  to expand charts to available width, and  to view tabular representations of charts. Additionally double-click on any graph to open a larger representation.

tation.

The **Segment Viewer** is used to characterize the **Segmentation Variable** categories. In cases where there are lots of variables this may incur considerable time and effort. To address this, two additional columns relaying **Information Value** and **Entropy Variance** statistics are also available.

These statistics are calculated from the first split of a **Decision Tree** using the **Entropy Variance Non P-Value Information Gain Measure** and reflect the relationship of each variable to the **Segmentation Variable**. Higher values of the **Info Value** and **Entropy Ratio** indicate **Independent Variables** with greater predictive power.

Either static can be sorted in ascending or descending order by clicking the appropriate column header. Sorting in descending order has the resulting effect of pushing the most contrasted variables to the top. This shortcuts the process of characterizing the **Segmentation Variable**.

The Yes category is characterized by: older married males, working longer hours per week.

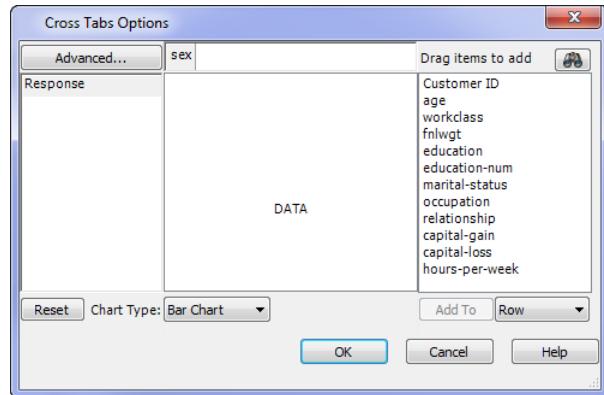
Use the export to **Excel** button: , to send any profiling view to **Excel**.

NOTE: that the **Segment Viewer** can be used not only as a means to characterize a **Segmentation Variable** but also to identify potentially good predictors of a target variable.

4.8 Cross Tabs Tab

Cross-tabulations allow visualization of data as **Surface Chart**, **Scatter Plot** or **Bar Chart**. Options are accessed via **Options**. Drag and drop **Row** and **Column** variables.

Figure 4.8: Crosstabulations Options



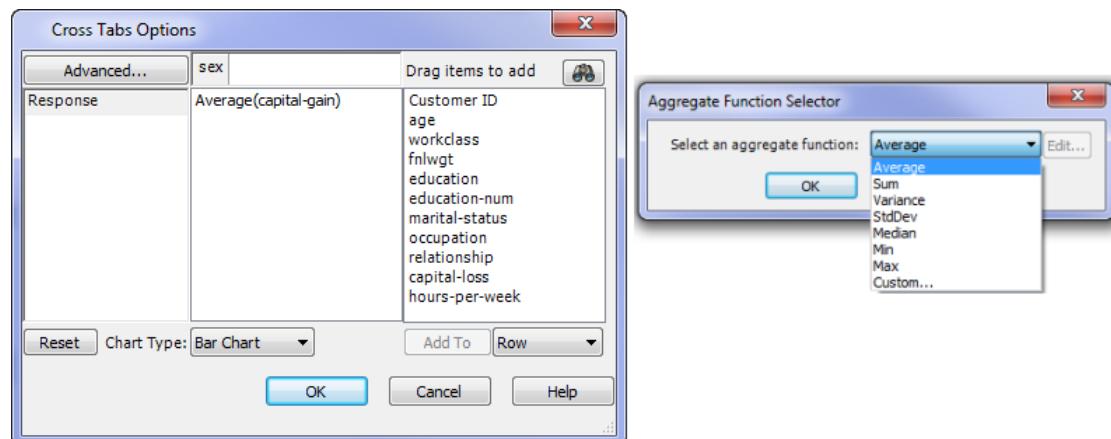
Results display the no. cases in each category of *Sex* within each *Response* category. The default illustration is a clustered bar chart. Access chart customizations and the chart **Properties** dialog by right-clicking.

Figure 4.9: Results



Include continuous variables by dragging them to the central pane. Here, *capital-gain* is used.

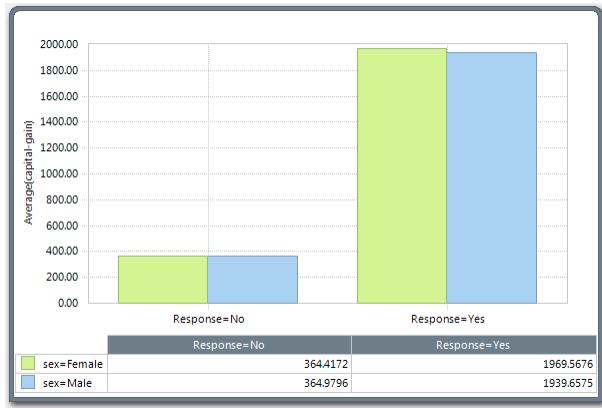
Figure 4.10: Adding a Continuous Variable



Alternative summary statistics are available and can be accessed by double clicking on the summary statistic and selecting an alternative from the available list. Custom functions can also be created.

Results show the average *capital-gain* for each category of *sex* within the *Response* variable categories. The average *capital-gain* for those in the *Yes* category is greater than those in the *No* category, regardless of *sex*.

Figure 4.11: With Summary



4.9 Characteristic Analysis

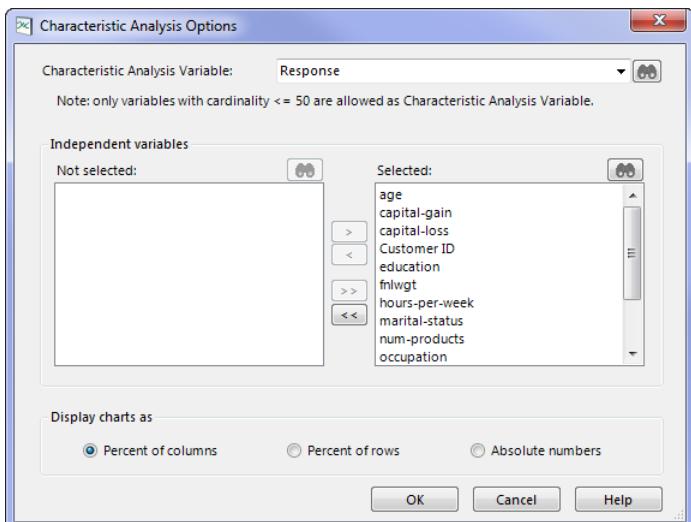
Characteristic Analysis extends cross-tabulations by enabling the inclusion of both categorical and continuous variables.

Continuous variables are binned into 10 approximately equal width intervals where possible, or can be user specified. Categorical variables are taken as is.

Characteristic Analysis generates cross tabulation tables with counts and row and column percentages. Graphs are also generated based on one of these user selected representations.

The graphs can be used to assess the degree of interaction between each variable and the dependent.

Figure 4.12: Characteristic Analysis Options



Bins can be customised for each variable by selecting the **Edit Binning** button.

4.10 Correlations Tab

KnowleddegSTUDIO generates four correlation statistics:

- Pearson
- Spearman's Rho
- Kendall's Tau-b
- Hoeffding

The **Pearson** correlation coefficient is based on the raw values of the variables, while **Spearman's rho** is based on ranking the values.

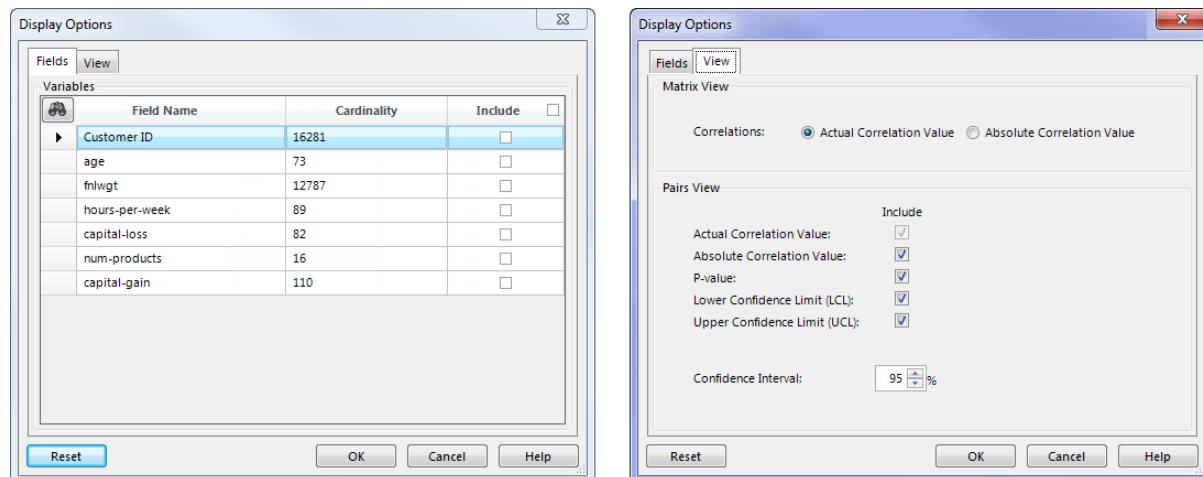
Spearman's rho is less vulnerable to the presence of outliers and extreme values which contrasts with the **Pearson** coefficient which is inherently affected by them.

Kendall's Tau-b coefficient is preferable in determining whether two non-parametric data samples with ties are correlated.

Hoeffding's measure of dependence (**Hoeffding's D**) can pick up on nonlinear relationships, and lies on the interval $[-.5, 1]$ if there are no tied ranks, with larger values indicating a stronger relationship. With a large number of ties in a small data set, the statistic might be less than -0.5 .

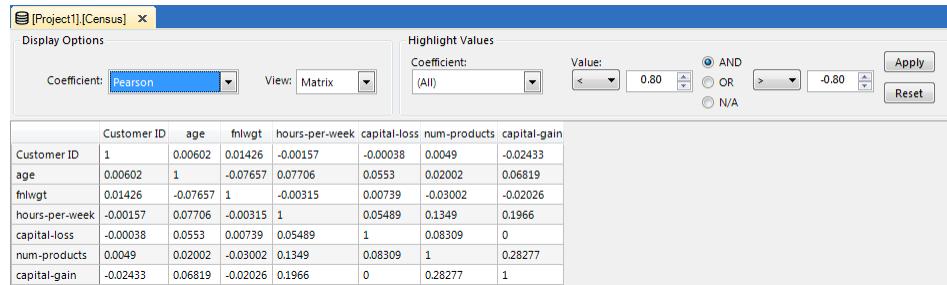
Selecting the **Correlations** tab in **KnowleddegSTUDIO** launches the **Display Options** wizard. From here variables can be selected for analysis as well as determining whether to view actual or absolute correlations, p-values and confidence intervals. Only numeric fields are available for selection.

Figure 4.13: Display Options



Click **OK** to display results.

Figure 4.14: Matrix View



The screenshot shows the KnowledgeSTUDIO interface with the title bar [Project1][Census]. The main area displays a correlation matrix for the Census dataset. The columns and rows are labeled: Customer ID, age, fnlwgt, hours-per-week, capital-loss, num-products, and capital-gain. The matrix values are as follows:

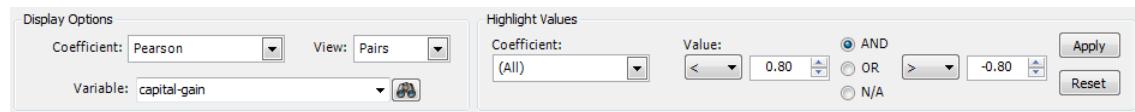
	Customer ID	age	fnlwgt	hours-per-week	capital-loss	num-products	capital-gain
Customer ID	1	0.00602	0.01426	-0.00157	-0.00038	0.0049	-0.02433
age	0.00602	1	-0.07657	0.07706	0.0553	0.02002	0.06819
fnlwgt	0.01426	-0.07657	1	-0.00315	0.00739	-0.03002	-0.02026
hours-per-week	-0.00157	0.07706	-0.00315	1	0.05489	0.1349	0.1966
capital-loss	-0.00038	0.0553	0.00739	0.05489	1	0.08309	0
num-products	0.0049	0.02002	-0.03002	0.1349	0.08309	1	0.28277
capital-gain	-0.02433	0.06819	-0.02026	0.1966	0	0.28277	1

By default all selected variables are displayed alongside all coefficients in a matrix style. Alternative display options are available from the **Coefficient:** and **View:** dropdown lists.

Notably with correlation tables there is duplication of the displayed statistics. Using the available drop-down menus allows restructuring of the tables to reduce redundancy.

Using the **Display Options** settings as per figure 4.15 produces the restructured matrix as per figure 4.16.

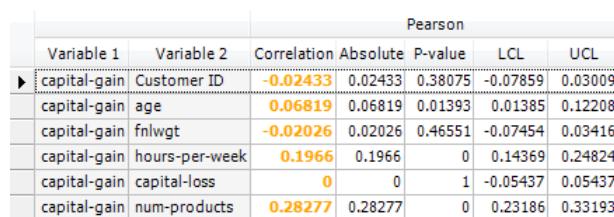
Figure 4.15: Display Settings



The screenshot shows the KnowledgeSTUDIO interface with the title bar [Project1][Census]. The main area displays the same correlation matrix as Figure 4.14. The **Display Options** dialog box is open, showing the following settings:

- Coefficient:** Pearson
- View:** Pairs
- Variable:** capital-gain
- Highlight Values** settings: Coefficient: (All), Value: < 0.80 AND, Apply button

Figure 4.16: Restructured Matrix



The screenshot shows the KnowledgeSTUDIO interface with the title bar [Project1][Census]. The main area displays the restructured correlation matrix. The columns are labeled Variable 1 and Variable 2, and the rows are labeled by the variable names. The matrix values are as follows:

		Pearson				
Variable 1	Variable 2	Correlation	Absolute	P-value	LCL	UCL
capital-gain	Customer ID	-0.02433	0.02433	0.38075	-0.07859	0.03009
capital-gain	age	0.06819	0.06819	0.01393	0.01385	0.12208
capital-gain	fnlwgt	-0.02026	0.02026	0.46551	-0.07454	0.03416
capital-gain	hours-per-week	0.1966	0.1966	0	0.14369	0.24824
capital-gain	capital-loss	0	0	1	-0.05437	0.05437
capital-gain	num-products	0.28277	0.28277	0	0.23186	0.33193

NOTE: Clicking any column will sort results. Identical sorting capabilities are available from any similar representation of data in KnowledgeSTUDIO.

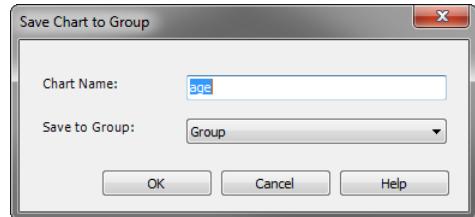
4.11 Saved Charts Tab

The **Saved Charts** tab is a great way to save and organize charts of interest.

The facility to save charts to the **Saved Charts** tab is accessed either by selecting the **Save Chart** icon , found on the **Taskbar**, or by right-clicking anywhere in the **Dataset Chart** tab to reveal the context menu and selecting **Save Chart**.

This activates the **Save Chart to Group** dialog, which provides the option of naming the chart and selecting the folder in which to save the chart.

Figure 4.17: Save Chart



Click **OK** and the chart is saved to the **Saved Charts** tab (Not Shown).

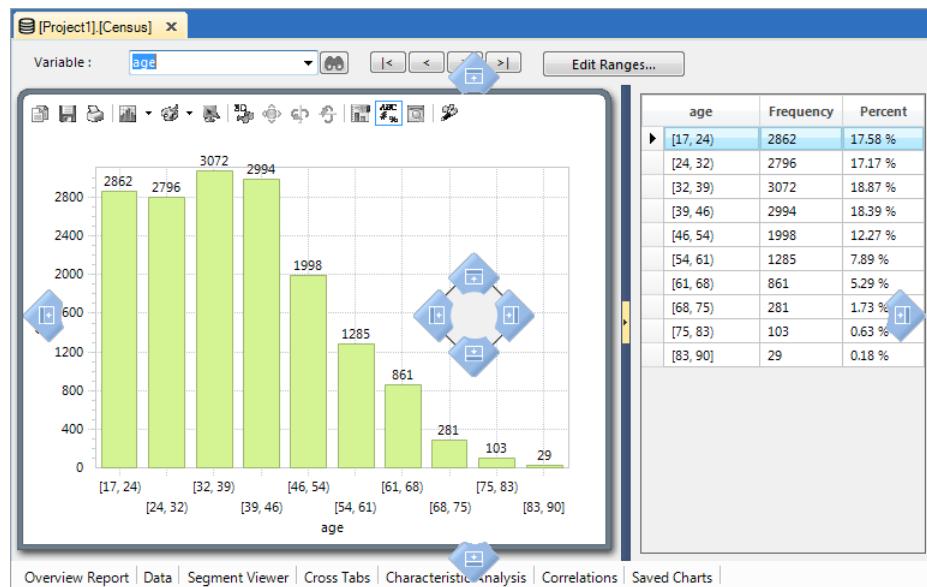
By default, only one chart group exists, called **Group**. Others can be created by activating the **Saved Charts** tab, right clicking on the **Group** folder name and choosing one of the available options to create a **New...** folder or **Copy** or **Rename** the currently selected folder.

4.12 Split Screen

KnowledgeSTUDIO includes split screen functionality for any tabbed window set.

To activate split screen, drag any tab onto the window display and release on one of the crosshair arrows that appears.

Figure 4.18: Split Screen



Repeat for any additional tabs to have a multiple split screen view; alternatively, right-click on a tab and

select either **New Vertical Tab Group** or **New Horizontal Tab Group**.

4.13 Summary

Data exploration and profiling is made easy and accessible with **KnowledgeSTUDIO**.

An array of features makes variable assessment and relationship analysis straightforward. As a result of completing this chapter users should be comfortable with using the following features to assess data:

- Univariate statistics using the **Overview Report**
- Graphical representations using **Dataset Charts**
- Using the **Data** tab to view a spreadsheet style display of the data
- Characterizing the categories of a variable using the **Segmentation Variable**
- Using and understanding correlations and cross-tabulations including **KnowledgeSTUDIO Characteristic Analysis**

4.14 Appendix: Summary Statistics Groups

This section describes the statistics available in the **Overview Report** tab. All statistics are accessible in expandable sections from the **Options** dialog.

Table 4.1: Basic Measures

Measure	Description
Cardinality	Number of distinct values
Unique Count	Number of values occurring exactly once
# of Missing Values	Number of missing values
% of Missing Values	Percentage of missing values
# of Non-Missing Values	Number of non-missing values
% of Non-Missing Values	Percent non-missing values
Minimum	Minimum value. Also applicable to String fields wrt lexicographic order
Maximum	Maximum value. Also applicable to String fields as before
Sum	Sum

Table 4.2: Measures of Central Tendency

Measure	Description
Mean	Average
Median	The value above and below which there are equal numbers of records
Mode 1	Most common value
Mode 2	Second most common value

Table 4.3: Measures of Dispersion

Measure	Description
Standard Deviation	Standard deviation
Range	Max minus Min
Interquartile Range	Range of 25th – 75th percentile
Coefficient of Variation	Standard deviation/mean
Std Error Mean	Standard error of the mean
Variance	Variance of the field values
Skewness	The asymmetry of the distribution. Values close to zero desirable
Kurtosis	Peaked/plateaued; Leptokurtic/Platykurtic . Values close to zero desirable

Table 4.4: Quantiles

Measure	Description
Customize	User specified
1%	Max 1st percentile
5%	5th percentile
10%	First, lowest decile 99th percentilee
25%	First, lower quartile
75%	Last, upper quartile
90%	Last, upper decile
95%	95th percentile
99%	99th percentile

Table 4.5: Extreme Observations

Measure	Description
Lowest Observations	n lowest values, specify n in parameters column. Default is 5
Highest Observations	n highest values, specify n in parameters column. Default is 5

Table 4.6: Tests for Normality

Measure	Description
Shapiro-Wilk Test	Tests for the null hypothesis that the field is normally distributed. Used for small sample sizes: 500 – 2000
Cramér-Von Mises Criterion	<p>Goodness-of-fit test. Used for judging the goodness of fit of a cumulative distribution function compared to a given empirical distribution function.</p> <p>When used as a test for normality; evaluates the squared difference between data EDF and standard normal distribution function F.</p> <p>Let $x[1], \dots, x[n]$ be the ordered set of all distinct non-missing values in the field x in ascending order.</p> <p>Then the statistic is computed as:</p> $\frac{1}{12n} + \sum_{i=1}^n \left(F(x_i) - \frac{2i-1}{2n} \right)^2$
Kolmogorov-Smirnov Test	<p>A goodness-of-fit test based on the empirical distribution function, EDF.</p> <p>When used as a test for normality, measures the largest vertical difference between the EDF of the data in the field and the standard normal distribution function F with the same mean and variance.</p> <p>Let $x[1], \dots, x[n]$ be the ordered set of all distinct non-missing values in the field x in ascending order.</p> <p>The statistic is the larger of the two values:</p> $\max_i \left \frac{i}{n} - F(x_i) \right \text{ and } \max_i \left F(x_i) - \frac{i-1}{n} \right $

Anderson-Darling Test Goodness-of-fit test based on the empirical distribution function, *EDF*.

When used as a test for normality, evaluates the weighted squared difference between the *EDF* of the data in the field and the standard normal distribution function *F*.

Let $x[1], \dots, x[n]$ be the ordered set of all distinct non-missing values in the field *x* in ascending order and $y[i]$ be their standardized values.

Then the statistic is computed as:

$$-n - \frac{1}{n} \sum_{i=1}^n \left[(2i-1) \ln F(x_i) + (2(n-1)+1) \ln(1 - F(x_i)) \right]$$

The null hypothesis that the data has standard normal distribution *F* is rejected if the value is larger than the chosen threshold.

Table 4.7: Tests for Location

Measure	Description
Student's t-test	Tests the null hypothesis that there is no difference between the average value of the selected field and a hypothesized value
Sign Test	Non-parametric equivalent of the students t-test
Wilcoxon signed-rank test	Non-parametric statistical test for the median

Table 4.8: Confidence Interval for Population Mean

Measure	Description
Confidence Interval for Population Mean	Generates the estimated range of values within which 95% of average values from samples of <i>this size</i> will lie. <i>this size</i> relates to the size of the current sample/data

Exercises

1. Start **KnowledgeSTUDIO**, if not already open.
2. If not already open, import one of the **Census** files.
3. From the **Overview Report** tab:
 - (a) Assess the number of records and variables
 - (b) Calculate statistics for all variables
 - (c) Become familiar with the following statistical measures:
 - i. Cardinality
 - ii. Extent of missing values
 - iii. Minimum and Maximum and other statistics
 - (d) Modify the extent of visible statistics using the **Options** button.
 - (e) Copy the visible results to any *MS Office* application (try *Excel*).
4. Use the **Dataset Charts** tab to assess the distributions of the variables.
 - (a) Select variables to view using the dropdown or arrow buttons.
 - (b) Assess the distributions of each variable and note any of interest.
 - (c) Access and modify some Chart Properties.
 - (d) Assess the distribution of the variable; Response
5. From the **Data** tab:
 - (a) View the data.
 - (b) Modify the view to show only categorical variables. (**HINT:** use the **Options** button).
6. From the **Segment Viewer** tab:
 - (a) Choose the variable *Response* as the **Segmentation Variable**.
 - (b) Select variables to profile and move to the **Selected** pane.
 - (c) Fit the charts to the width of the page.
 - (d) Assess whether there are any differences across the categories of the **Dependent Variable** for any **Independent Variables**
 - (e) Use the **Information Value** and **Entropy Variance** columns to aid in identifying variables that characterize the **Dependent Variable**
7. From the **Cross tabs** tab:
 - (a) Run cross tabulations with some categorical variables.

- (b) Investigate some continuous variables with the variable; *Response*.
HINT: Try *capital-gain* and *capital-loss*.

8. From the **Correlations** tab:

- (a) Select all continuous variables and choose a coefficient.
- (b) Create the correlation matrix.
- (c) Choose **pairs** from the **View** dropdown and focus the view by selecting one of the variables.
- (d) Can you re-run and add the variable *Response*? Why not?
- (e) Modify the view to list correlation in descending absolute order.

Chapter 5: Data Preparation

5.1 Introduction

There are two main groups of functions to aid in preparing data in **KnowledgeSTUDIO**:

- Dataset operations
- Field transformations

Dataset operations refer to actions at the dataset level such as merging two files or removing duplicates. Field transformations work at the case level, creating new fields or transforming existing ones.

As a result of completing this chapter users will be able to use and apply **Altair's** data preparation nodes including:

- Append
- Aggregate
- Join
- De-duplicate
- Variable Transformations

5.2 The Manipulate Palette

Nodes for performing dataset operations and field transformations can be found in the **Manipulate** palette.

There are a considerable number of nodes in this palette. This chapter will focus only on **Altair's** native capabilities and only nodes available with a **KnowledgeSTUDIO** license.

To view only these nodes, the **Altair** filter button is selected.

Figure 5.1: Filter Altair Nodes

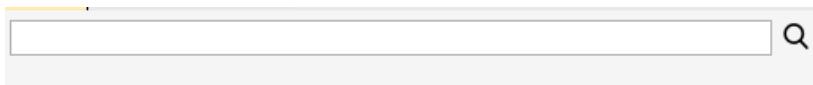
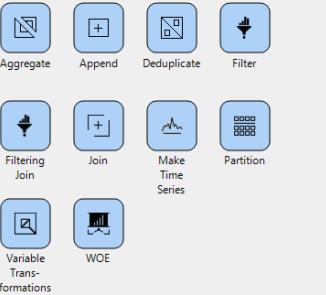


Table 5.1: Manipulate Palette

Palette	Node	Description
	Aggregate	Group records on one or more attributes and generate summary statistics
	Append	Add Records from two datasets
	Deduplicate	Remove rows identical on all fields
	Join	Merge datasets on a single column
	Partition*	Create Samples
	Variable Transformations	Use to derive new fields
	WOE*	Use to create WOE variables
	Variable Selection*	Assess relationships between Dependent Variable and Independent Variable

*Elaborated on in further chapter

5.3 Dataset Operations

Four common dataset operations are used frequently when preparing data for analysis.

- Appending
- Merging
- Aggregating
- De-duplicating

Each operation can be used in isolation but in many instances are combined in varying ways to address data preparation issues.

The following sections demonstrate each through the lens of a common usage scenario which utilizes each of these operations at various stages in the data preparation process.

5.3.1 Scenario

Customer transactions are compiled weekly and stored in separate files. A customer base file exists containing demographic information. There is interest in enriching the transactional information with associated demographics to potentially use for modelling.

Since the information desired applies on a customer level, the final file should have one line per customer. A number of data preparation steps are needed to get to the desired result.

Figure 5.2: Common Data Prep Elements

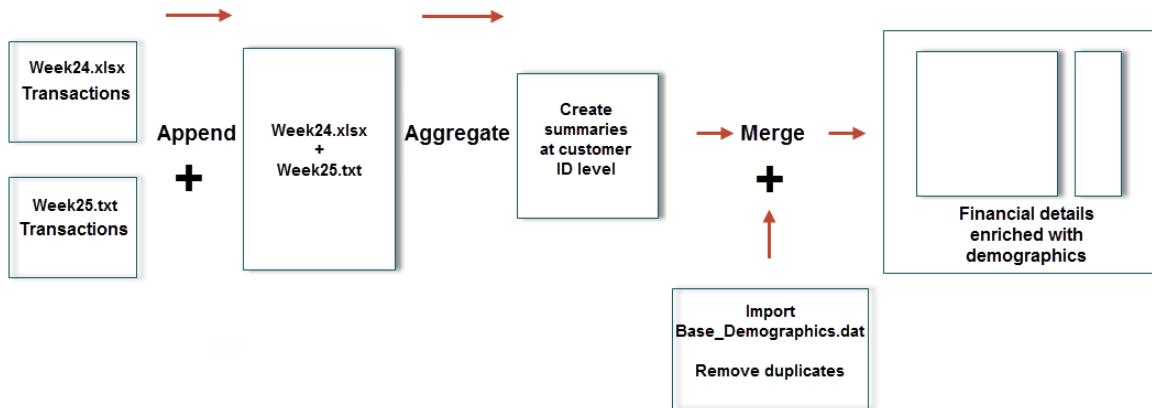


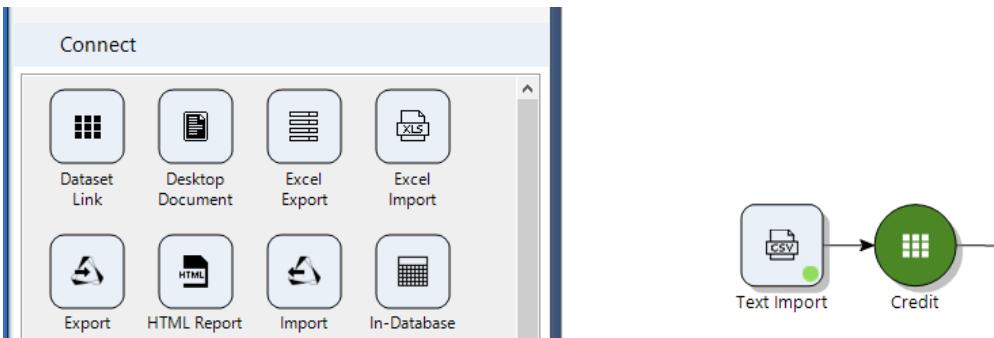
Figure 5.2 illustrates the steps involved to combine transactional and demographic data. Once complete the results are aggregated to the customer level.

The results are enriched with demographic detail from a de-duplicated customer file. The final file contains the transactional details enriched with demographics. Files used are:

- Week24.xlsx
- Week25.txt
- Base_Demographics.txt

The first step is to create a new project and import the files. As the files are in separate formats, a **Text** node and an **Excel** node from the **Source** palette are used to read each.

Figure 5.3: File Import



Once the import process is complete the first processing step, combining records across files using an append node, can proceed.

5.3.2 Appending Datasets

Append is a term used in statistics and *Data Mining* to refer to the concatenation of datasets where the records in one file are combined with the records in a second file.

The number of records in the appended results should equal the sum of the number of records in both files.

Viewing the **Overview Report** for both **Week24** and **Week25**, not shown, shows record counts of 300 and 430 records respectively. The appended results should then have 730 records.

Exploring both files from the **Data** tab on each provides additional useful insight:

Figure 5.4: Data Records

ID	No_Products	Cost	ID	No_Sales	Price
1	3	219.87	301	10	365.53
1	7	330.58	301	13	705.63
1	9	306.45	301	7	515.38
1	5	80.87	301	14	260.05
1	3	58.3	301	8	116.14
1	10	152.24	301	6	129.35
1	6	652.32	301	8	693.07
1	7	66.85	308	14	780.16
1	2	14.12	308	12	285.77
1	2	122.76	308	13	1116.4

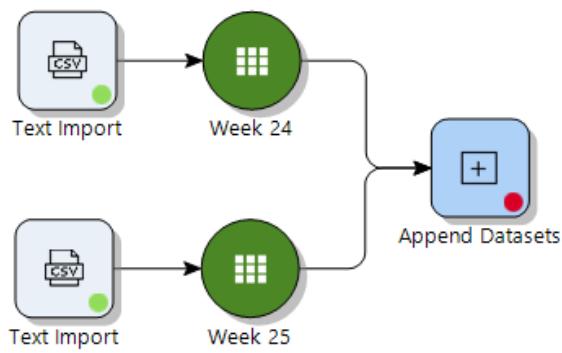
The field *ID* identifies customers and is named the same across both files. Note that there are multiple transactions per customer; hence the need for aggregation to obtain one record for each customer *ID*.

Some field names are different for **Week 24** and **Week 25**. *No_Sales* and *No_Products* record the same information as **Cost** and **Price** respectively. However they are named differently across files.

During the **Append** process these fields can be mapped to indicate that they are the same.

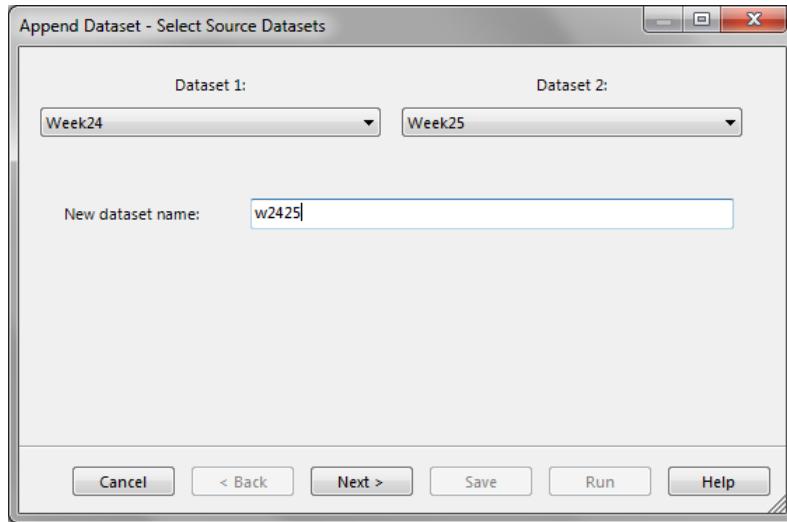
To initiate the append, drag an **Append** node from the **Manipulate** palette onto the canvas and connect **Week24** and **Week25** as illustrated in figure 5.5

Figure 5.5: Appending Files



Either double click the **Append Datasets** node or right click and choose **Modify** to access the append dialog.

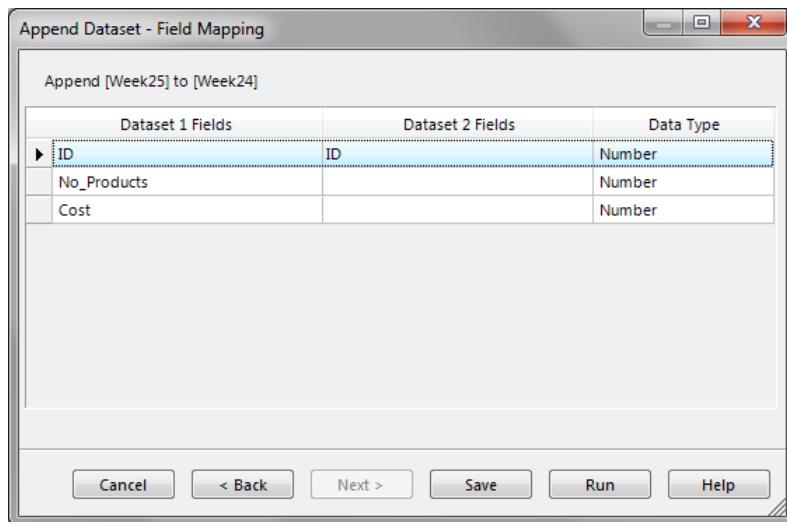
Figure 5.6: Append Dataset - Select Source Datasets



The first dialog, **Append Dataset – Select Source Datasets**, provides options to choose the datasets to append and assign a name to the newly created dataset.

Connected datasets are automatically identified. Assign the name **W2425** as the new dataset name. Clicking **Next >** opens the **Append Dataset – Field Mapping** dialog.

Figure 5.7: Append Dataset - Field Mapping



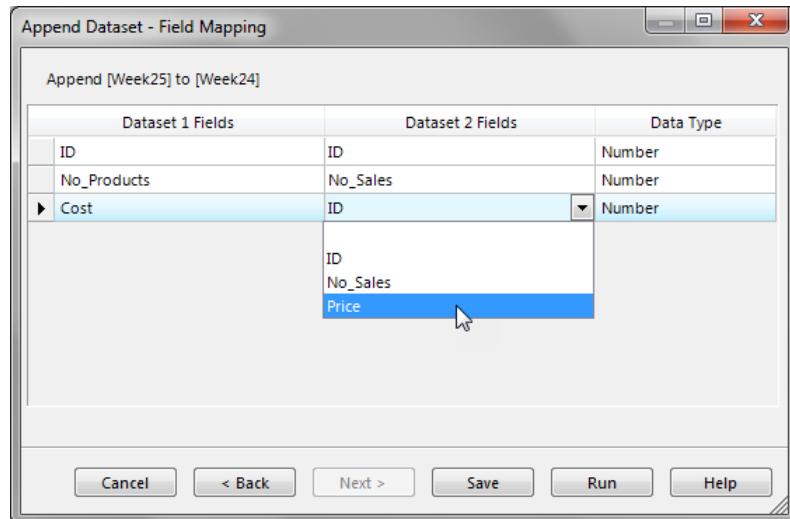
This dialog is used to map fields across files. All **Dataset 1 Fields** are present but only *ID* from **Dataset2** is initially present. This is as a result of two default aspects:

- The file selected as **Dataset 1** determines the default fields in the appended results
- **Dataset 2** fields that match names evident in **Dataset 1** are retained, all others are dropped

Current settings mean all fields from **Dataset 1** are retained but only *ID* for **Dataset 2**.

KnowledgeSTUDIO provides a drop down facility to match fields. Click the corresponding blank row in **Dataset 2 Fields** column to activate the dropdown and match *No_Products* to *No_Sales* and *Cost* to *Price*.

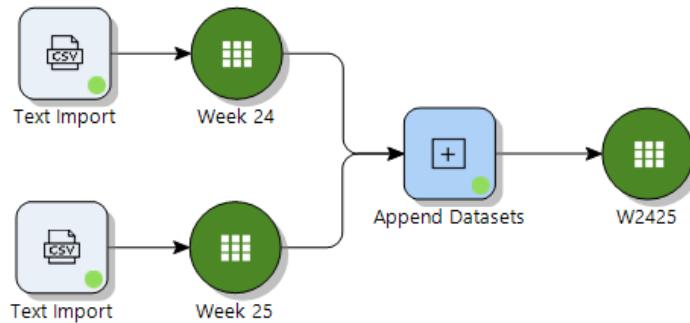
Figure 5.8: Override Defaults



Once options are set, click **Run** to complete the process.

NOTE: If the fields are of different data types, the match will fail. To prevent dropping fields when appending, create matching fields populated with null values. This can be accomplished with the expression: **CAST(null AS DATA_TYPE)**. **DATA_TYPE** must be replaced with an actual data type, e.g. **STRING** or **DOUBLE**, etc...

Figure 5.9: Appended Results



Note that a new node has been created on the canvas reflecting the appended results. This refers to the dataset created as a result of the process, and is located in the **Project Pane**.

Either double click **W2425** or right click and select **Open View** to assess results.

Figure 5.10: Append Dataset Report



Append Datasets Report		
Append [Week25] to [Week24]		
Input Datasets		
	[Week24]	[Week25]
Record count	300	430
Field Mapping		
[Week24]		[Week25]
ID	ID	
No_Products	No_Sales	
Cost	Price	
Result Dataset		
Name	[w2425]	
Record count	730	

The results open on the **Report** tab illustrating the record count of individual and final datasets and field mapping.

The record count of the final file corresponds to the sum of the records in the other two files. Also note that the appended dataset **W2425** has been added to the **Workflow** canvas and **Project Pane**.

The next step in the process is to aggregate results to generate one record for each unique *ID*. Summaries can be generated for total and average numbers of products purchased and costs.

5.3.3 Aggregating

Aggregation is used to summarize records based on a common identifier, also known as a grouping field.

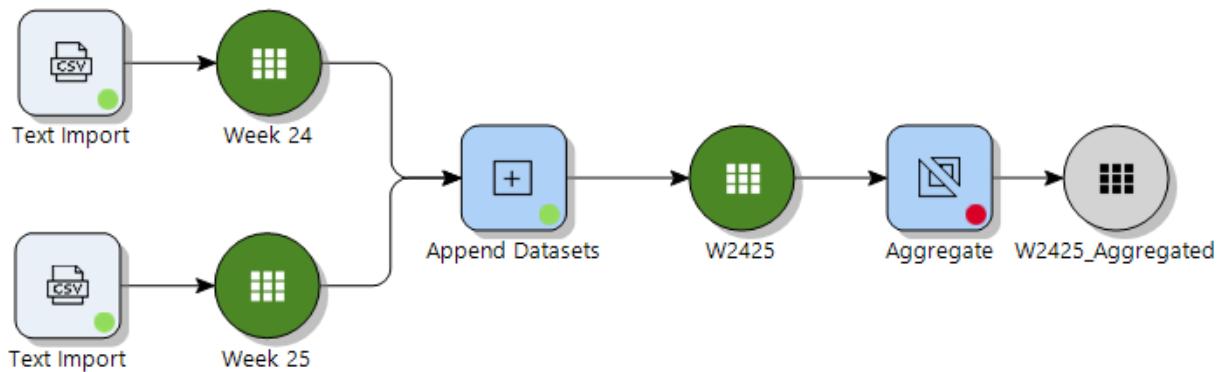
Aggregation is a common feature of data preparation. For example, individual transactions data can be aggregated to the customer level and various summaries produced to reflect average spend, total spend, total number of products purchased, total number of visits etc.

KnowledgeSTUDIO allows multiple grouping fields. Continuous fields can be aggregated using, among other functions, sum, average or maximum/minimum. Categorical fields are aggregated using the mode.

Multiple summaries can be generated for the same continuous field and results are stored in a new dataset containing the aggregated summary fields and the grouping field.

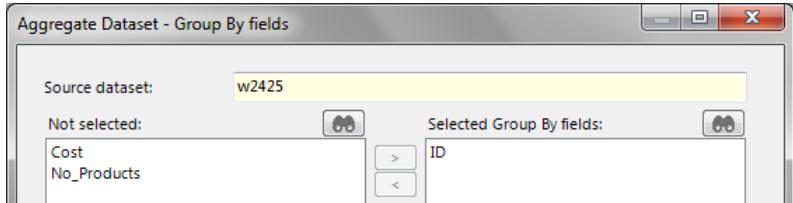
Drag the **Aggregate** node from the **Manipulate** palette and add as illustrated in figure 5.11.

Figure 5.11: Connect Aggregate Node



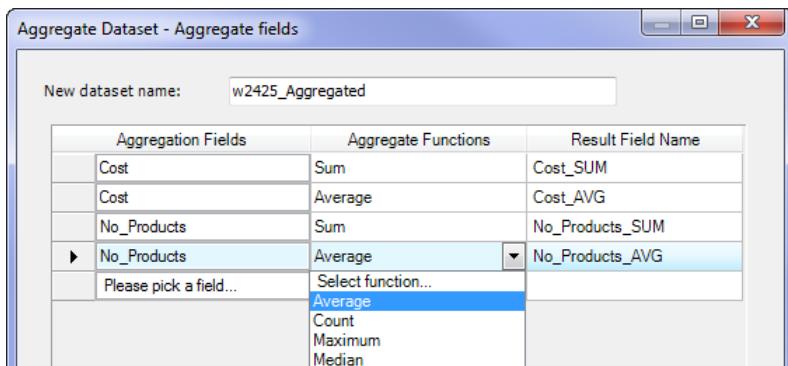
Either double click the **Aggregate** node or right click and select **Modify**, to access options.

Figure 5.12: Aggregate Dataset - Group By Fields



In the first dialog the grouping field(s) are identified. In this example fields are grouped by the field **ID**. Clicking **Next >** opens the **Aggregate Dataset - Aggregate Fields** dialog.

Figure 5.13: Aggregate Fields



In this dialog summary calculations are specified. Click **Please pick a field...** from the **Aggregation Fields** column and select the field to summarize from the dropdown.

In the **Aggregation Functions** column choose the appropriate summary function. **KnowledgeSTUDIO** automatically names the summary field, adding a suffix determined by the aggregate function selected. This can be changed by clicking in the **Result Field Name** slot and modifying.

Here, the **Sum** and **Average** are created for both *Cost* and *No_Products*. Default names are accepted.

Click **Run** to generate the aggregated dataset **W2425_Aggregated**,(not shown). Open the node to view results. The data opens on the **Report** tab as illustrated in figure 5.14.

Figure 5.14: Aggregated Dataset

Aggregate Datasets Report	
Input Datasets	
Name:	[w2425]
Record count:	730
Group By Fields	
ID	
Result Dataset	
Name:	[w2425_Aggregated]
Record count:	554
Aggregation Fields	
Aggregation	
SUM(Cost)	Cost_SUM
AVG(Cost)	Cost_AVG
SUM(No_Products)	No_Products_SUM
AVG(No_Products)	No_Products_AVG

The **Report** tab shows the input and output number of records, the grouping and aggregation fields. Note the number of records in the resulting file of 554. This contrasts with the input number of records of 730 and implies multiple purchases for the same *ID*.

Click the **Data** tab to view the created fields.

Figure 5.15: Aggregated Results Data Tab

	ID	Cost_AVG	Cost_SUM	No_Products_AVG	No_Products_SUM
►1	1	223.0073684210526	4237.139999	6.1578947368421053	117
2	20	297.6033333333333	892.81	4.6666666666666667	14
3	23	213.46	213.46	4	4
4	24	213.44	213.44	4	4
5	25	328.2479999999993	1641.239999	8.4	42
6	26	401.1340000000007	2005.670000	9	45
7	27	243.0079999999995	1215.039999	4.8	24
8	28	288.7199999999997	1443.6	5.8	29
9	29	31.02	31.02	2	2
10	34	98.25	98.25	2	2

As can be seen in figure 5.15 only the aggregated summary and grouping field are present in the resulting dataset.

NOTE: Selecting more than one field to aggregate on will result in all possible combinations of field values and summaries generated for these combinations.

This is illustrated using the **Census** dataset where both *sex* and *relationship* are selected as grouping fields and averages are generated for both *age* and *education-num*.

Figure 5.16: Aggregated Results with Two Grouping Fields

#	relationship	sex	age_AVG	education-num_AVG
1	Husband	Male	44.0829372987889	10.261996014103939
2	Not-in-family	Female	40.38095238095238	10.444110275689223
3	Not-in-family	Male	36.984231274638631	10.265440210249672
4	Other-relative	Female	36.656370656370655	9.40926640926641
5	Other-relative	Male	31.218045112781954	8.5338345864661651
6	Own-child	Female	24.573828470380196	9.6498673740053054
7	Own-child	Male	24.850217076700435	9.3972503617945016
8	Unmarried	Female	40.981946624803768	9.66326530612245
9	Unmarried	Male	38.3604938271605	9.4345679012345673
10	Wife	Female	40.5249343832021	10.547244094488189
11	Wife	Male	64	6

The next steps in the process are to merge the aggregated results with the file containing customer demographic details; *Base_Demographics.txt*.

In this case, prior to merging, it is best to ensure that there are no duplicate entries in the file containing demographic detail.

This can be accomplished using the **Deduplicate** node from the **Manipulate** palette.

5.3.4 Removing Duplicates

Use the **Text** node from the **Source** palette to read the file; *Base_Demographics.txt*. Once imported, either double click the **Base_Demographics** node or right click and select **Open View** to view the dataset.

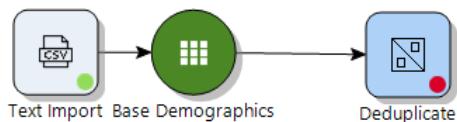
Figure 5.17: Overview Report

[Project1].[Base_Demographics]												
Calculate		Calculate All		Dataset:		Weight:		Records:		Fields:		10
#	Field Name	Field Label	Data Type	Cardinality	Unique Count	# of Missing Values	% of Missing Values	Minimum	Maximum	Mean	Standard Deviation	Skewness
1	ID	ID	Number	15882	15849	0	0.00 %	1.00	16,281.00	8,139.19	39.02	-0.000
2	age	age	Number	73	0	0	0.00 %	17.00	90.00	39.02	39.02	-0.000
3	workclass	workclass	String	9	0	0	0.00 %	?	Without-pay	Some-college	Some-college	-0.000
4	education	education	String	16	0	0	0.00 %	10th	16.00	10.02	10.02	-0.000
5	education-num	education-num	Number	16	0	0	0.00 %	1.00	16.00	10.02	10.02	-0.000
6	marital-status	marital-status	String	7	0	0	0.00 %	Divorced	Widowed	Widowed	Widowed	-0.000
7	occupation	occupation	String	15	0	0	0.00 %	?	Transport-moving	Transport-moving	Transport-moving	-0.000
8	relationship	relationship	String	6	0	0	0.00 %	Husband	Wife	Wife	Wife	-0.000
9	sex	sex	String	2	0	0	0.00 %	Female	Male	Male	Male	-0.000
10	hours-per-week	hours-per-week	Number	89	4	0	0.00 %	1.00	99.00	40.34	40.34	-0.000

The dataset contains 16,281 records and ten fields; nine of which are varied demographics from *age* to *hours-per-week*, additionally there is an *ID* field, this will be used to link records to the aggregated results generated previously.

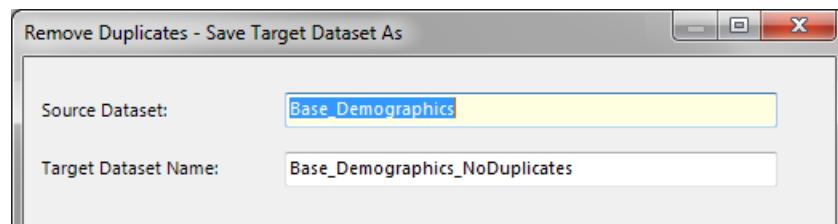
To perform the de-duplication, add and connect a **Deduplicate** node from the **Manipulate** palette to the **Base_Demographics** data node.

Figure 5.18: Deduplicate Node Added



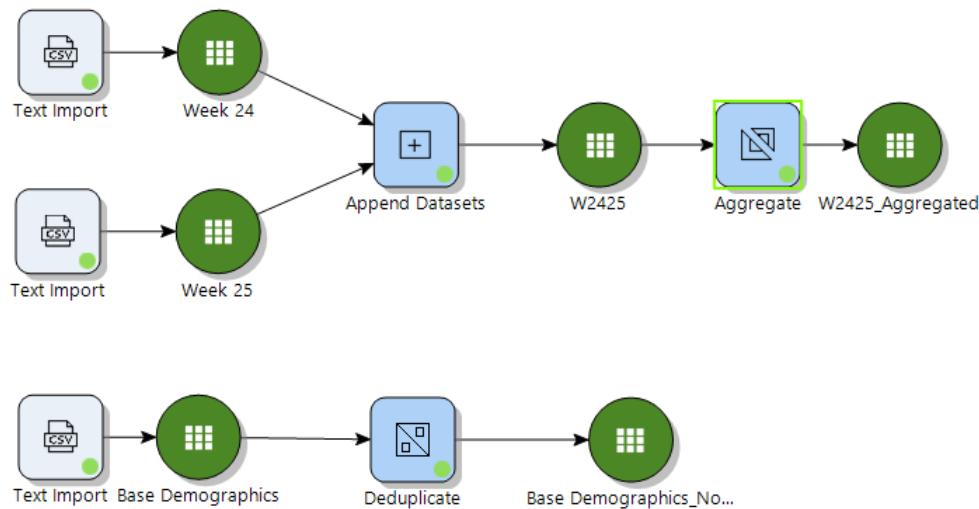
Once added open the **Deduplicate** node to access options.

Figure 5.19: Remove Duplicates Dialog



De-duplication in **KnowledgeSTUDIO** deletes only those records that are exact duplicates, i.e. records whose values are identical for all fields. Therefore options are nil and the process swift. Click **Run** to complete the process. The **Workflow** is depicted in figure 5.20.

Figure 5.20: Current Workflow



Either double click on the create **Base_Demographics_NoDuplicates** node or right click and choose **Open View** to view results.

Figure 5.21: Deduplicated Dataset Report - Partial View

Remove Duplicates Report
 Altair

Input Datasets	
Name	[base_demographics]
Record count	16,281

Grouping Fields	
age	
education	
education-num	
hours-per-week	
ID	
marital-status	
occupation	
relationship	
sex	
workclass	

The results open on the **Report** tab as illustrated in figure 5.21 showing the number of records in the source and resulting file as well as the number of duplicates found and removed, here 194.

The final step in the process is to merge results.

5.3.5 Merging Records across Datasets

Merging records enables information in one file to be enriched with that in another. A field common to both files must exist to match records across files.

KnowledgeSTUDIO supports three types of merge as illustrated in table 5.2.

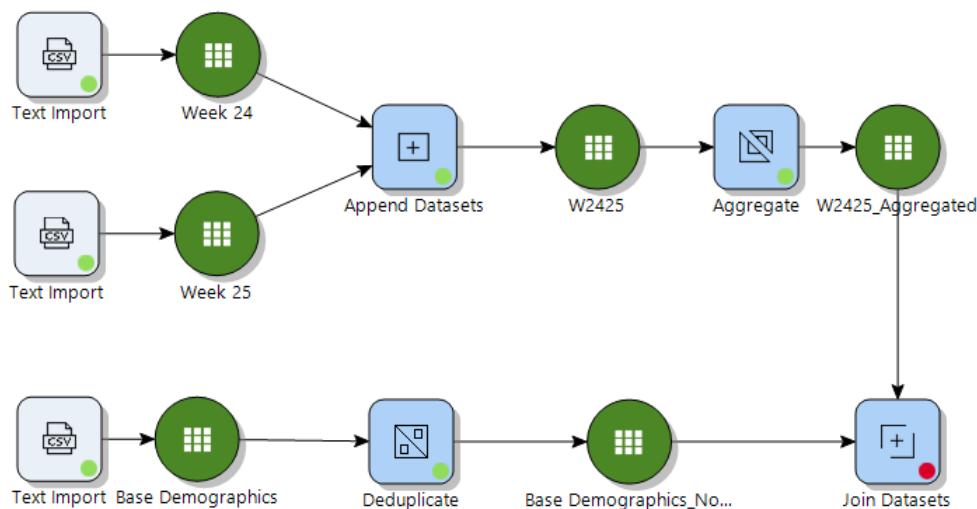
Table 5.2: Merge Options

Merge Option	Description
	Retain records that exist in both datasets
	Retain all records from the left dataset and matching records from the right
	Retain all records from the right dataset and matching records from the left

All records in the file **w2425_Aggregated** dataset shall be retained and only matching records from **Base_Demographics_NoDuplicates**.

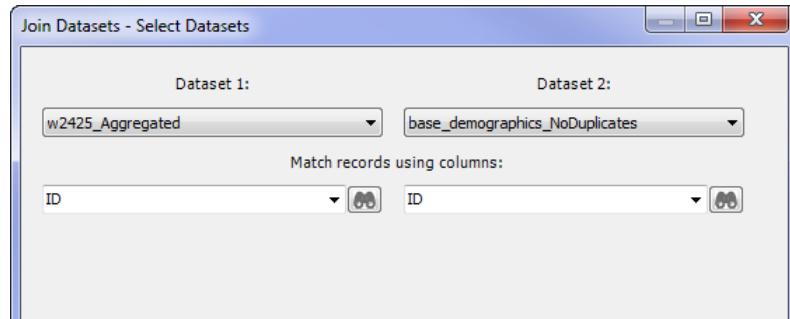
To initiate the process; drag a **Join** node from the **Manipulate** palette, connect both datasets **W2425_Aggregated** and **Base_Demographics_NoDuplicates**.

Figure 5.22: Add Join Node



Once complete open the **Join** node to access options.

Figure 5.23: Join Datasets - Select Datasets



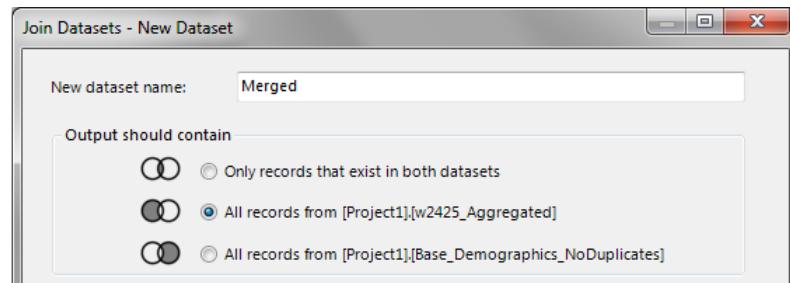
The first dialog; **Join Datasets – Select Datasets**, allows selection of **Dataset 1** and **Dataset 2**. These are automatically populated by the connected datasets.

Here the matching variable is selected for each datasets in **Match records using column:** area. If a common variable exists in both datasets it is automatically selected.

NOTE: KnowledgeSTUDIO provides the facility to merge datasets on one field only. If a merge based on two or more fields is required, it is advisable to create a new field that concatenates values.

Clicking **Next >** opens the **Join Datasets – New Dataset** dialog.

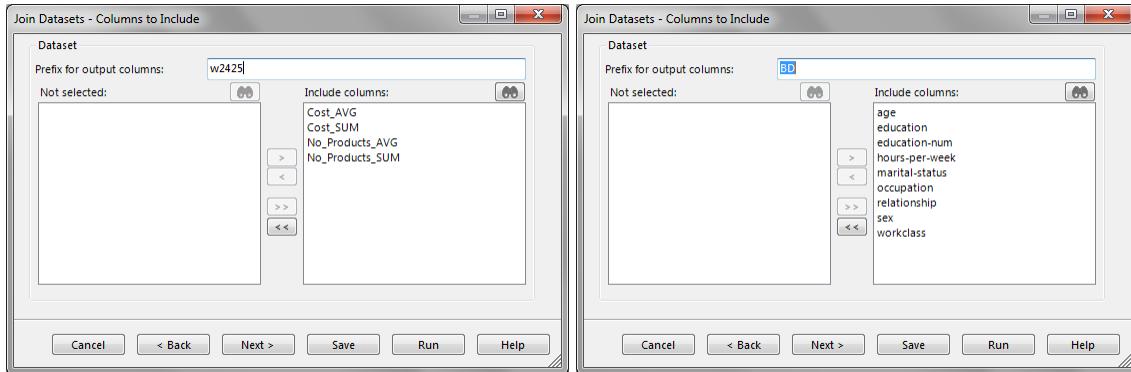
Figure 5.24: Join Datasets - New Dataset



This dialog allows selection of the merge type and assigns a name to the merged results.

The appropriate merge type in this example is **All records from [Project1].[w2425_Aggregated]**. Assign the new dataset name as **Merged** and click **Next >**.

Figure 5.25: Columns to Include



The next two dialogs require selection of fields to include in the final dataset.

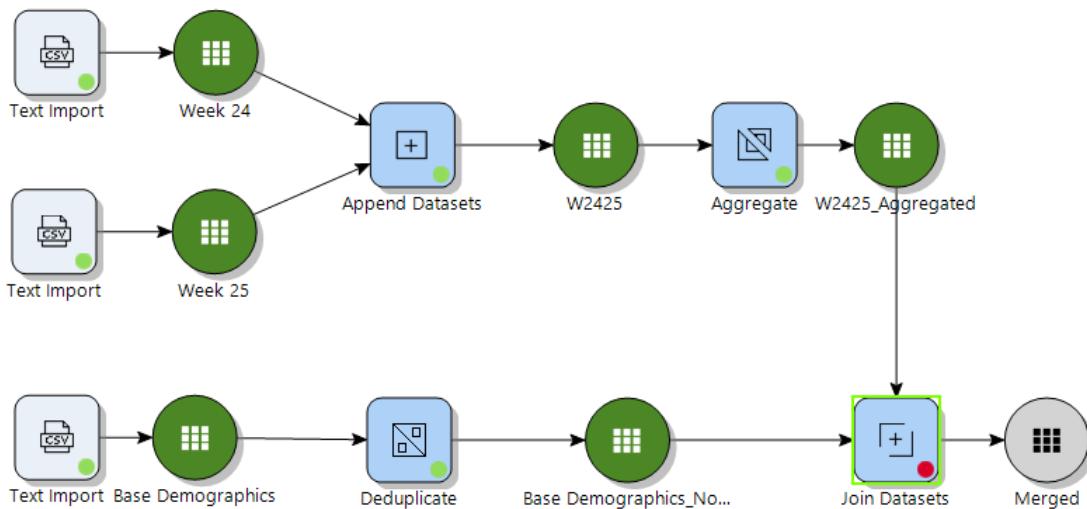
A prefix can be created to identify from which dataset each field comes from. The default prefix is the dataset name, which can be quite long, so here are changed as follows:

w2425_Aggregated changed to **w2425** and **Base_Demographics_NoDuplicates** changed to **BD**.

Click **Next >** and a final dialog appears illustrating the initial datasets and fields on the left hand side and a preview of the new fields and names in the right hand side, not shown.

Click **Run** to complete the process and generate results. The resulting **Workflow** is illustrated in figure 5.26.

Figure 5.26: Joined Datasets



To view results either double click the node **Merged** or right click and select **Open View**.

Figure 5.27: Join Report - Partial View



Input Datasets		
	[w2425_Aggregated]	[base_demographics_NoDuplicates]
Record matched using column	ID	ID
Record count	554	16,087
Prefix for output columns	w2425_Aggregated	base_demographics_NoDuplicates

Result Dataset	
Join type	Only records from [w2425_Aggregated]
Record count	554

Columns Included in Output	
From [w2425_Aggregated]:	From [base_demographics_NoDuplicates]:
Cost_AVG	age
Cost_SUM	education
No_Products_AVG	education-num
No_Products_SUM	hours-per-week
	marital-status
	occupation
	relationship

The **Report** tab provides information on the input and output datasets, and the resulting output fields.

Note the output dataset contains 554 records, this is identical to the number of cases in the **W2425_Aggregated** file and the desired result.

Click on the **Data** tab to view the resulting aggregated fields and associated demographics.

5.4 Variable Transformations

Variable transformations are a means to create new fields based on the values of others.

The following demonstrations use the file *Census.xls* and illustrates the following typical examples:

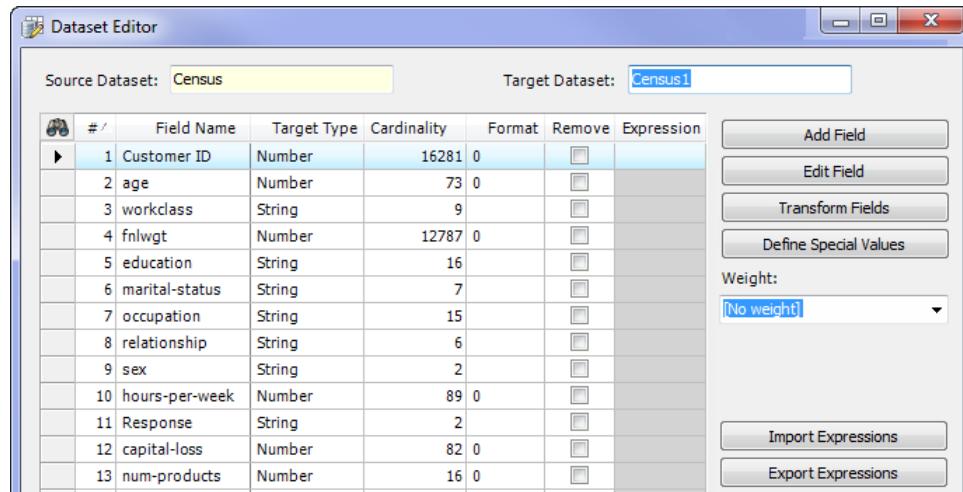
- Identify special variable values
- Binning values of an already existing field
- Applying the same transformation to multiple fields

Variable transformations are facilitated in **KnowledgeSTUDIO** using the **Dataset Editor**. This is accessed via the **Variable Transformations** node, found in the **Manipulate** palette.

5.4.1 The Dataset Editor

Create a new project and import the excel file: *Census.xls* using the **Excel** import node. Next attach a **Data Transformations** node to the **Census** dataset and open as illustrated in figure 5.28.

Figure 5.28: Dataset Editor



The **Dataset Editor** displays the **Source Dataset**: Additional options are detailed in table 5.3

Table 5.3: Dataset Editor Options

Option	Description
Target Dataset:	Specify a name for the dataset to be created
Add Field	Click this button to open the Expression Editor and add new fields
Edit Field	Modify newly derived field expressions or pre-existing dataset field formats. The values in pre-existing dataset fields cannot be changed
Transform Fields	Apply the same transformation to multiple variables simultaneously
Define Special Values	Id values with special meaning, e.g. if -999, convert to null
Weight:	Select a weighting field
Import Expressions	Transformation expressions can be imported from another open dataset or in XML format from an Altair XML format expression file
Export Expressions	Export as an Altair Expression Format (XML) file or as Plain Text
Remove	Remove columns

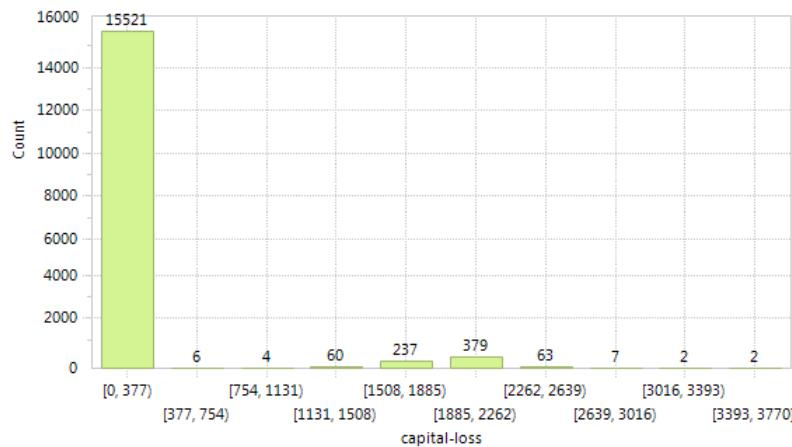
5.4.2 Field Transformations: Identifying Special Values

KnowledgeSTUDIO provides a **Define Special Values** radio button on the **Variable Transformations** node. This allows identification of one or more special values for any selected variable.

For example, a code of -999 may exist to represent a specific state, e.g.: *value illegible*. Proceeding with the value unchanged means it will be included in any calculations, an undesirable outcome.

Special Values can be either replaced with *null* or simply highlighted as being special! For this demonstration, the variable *capital-loss* is used. Figure 5.29 illustrates its distribution.

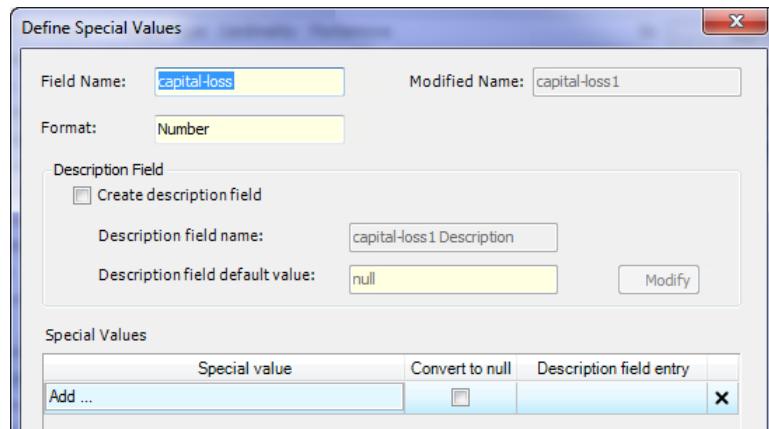
Figure 5.29: Distribution of *capital-loss*



The distribution is dominated by records with a value of 0, reflecting no *capital-loss*. This value is an error and should in fact appear as null rather than 0. To ensure this is the case the **Define Special Values** facility is an ideal remedy.

The first step, once the **Dataset Editor** is accessed, is to select the variable *capital-loss*. Once complete, click the **Define Special Values** radio button. The resulting view is illustrated in figure 5.30

Figure 5.30: Define Special Values Dialog



The **Special Values** area provides options to either simply highlight or modify **Special Values**.

A **Description Field** can be created relaying information about the **Special Value(s)** highlighted or modified.

If the focus is to modify values, a new field is created with those values changed. The new variable name is set via the **Modified Name:** slot. The default name comprises the variable to modify coupled with the number 1.

NOTE: This option becomes available only when a **Special Value** is identified and a replacement specified.

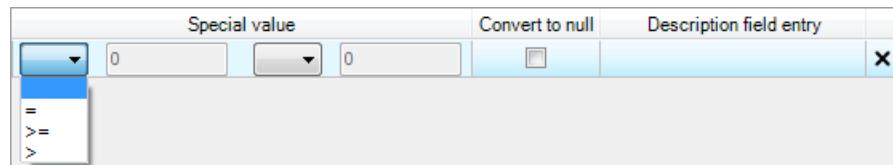
Some options in the dialog are not modifiable and are coloured yellow. These include the selected **Field Name:** and **Format:**

NOTE: The **Description field default value** is also yellow, but can be modified.

To address the 0's in the field *capital-gain* and replace with *null* is straightforward. In the **Special Values** area, click **Add ...** in the **Special value** column.

This provides options to specify either one or a range of values via a dropdown as illustrated in figure 5.31

Figure 5.31: Special Values Area



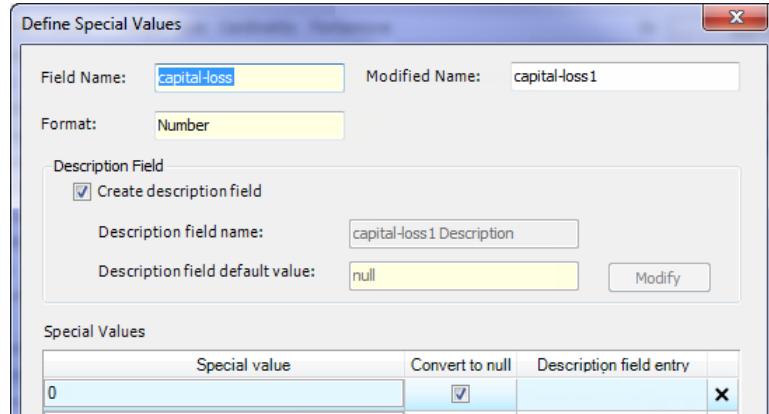
For this example, the appropriate option to select is **=**. Once selected, add the value 0.

Figure 5.32: Identifying Zero

Special value	Convert to null	Description field entry
0	<input type="checkbox"/>	<input type="text"/>
Add ...	<input type="checkbox"/>	<input type="text"/>

NOTE: Click **Add** to define more **Special Values**.

At this point the value 0 has been identified, notice that the **Modified Name** is still unavailable. This becomes available once the **Convert to null** check box is selected.

Figure 5.33: Replacing with *null*

For this example the default name is accepted. Finally, a **Description Field** can be created. This can be used to input a numeric or string value describing the modification.

For example, if there is interest in assigning a value of 1 if a record contains the **Special Value**, then the value 1 is entered in the **Description field entry** slot. Similarly, a string value such as *0 replaced by null* can be entered. Bear in mind that any string value must be quoted!

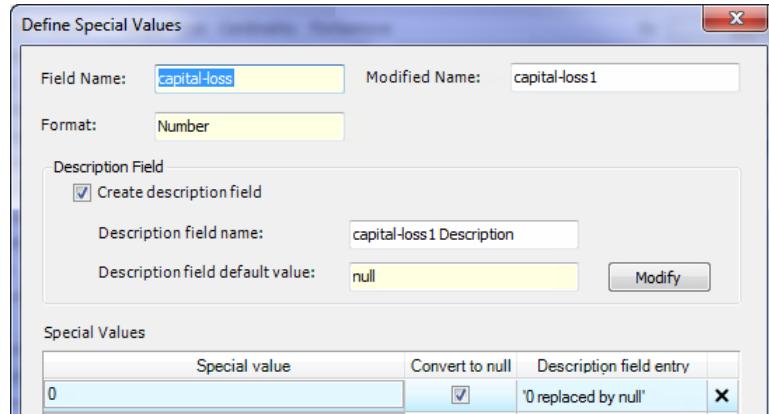
To add the string descriptor the **Create description field** checkbox must be selected. Once complete, options to modify the **Description field name:** and **Description field default value:** become available.

The default **Description field default value:** is *null*. This assigns *null* to records that do not contain the **Special Value**. This can easily be changed by clicking the **Modify** button. Once clicked this opens the familiar **Expression Editor**, where the default value can be modified.

NOTE: To create a dichotomous variable: set the default value to 0 and set the **Description field entry** value for the **Special Variable** to 0.

For this demonstration, a string descriptor: *0 replaced by null*, is added to identify modified records. All other records have the value *null*. The completed dialog is illustrated in figure 5.34

Figure 5.34: Completed Dialog



Clicking **OK** returns to the **Dataset Editor**. Notice the new additions: *capital-loss1* and *capital-loss1 Description*, (not shown). Click **Run** to create the new fields. Figure 5.35 shows the original and newly created variables in the **Data** tab.

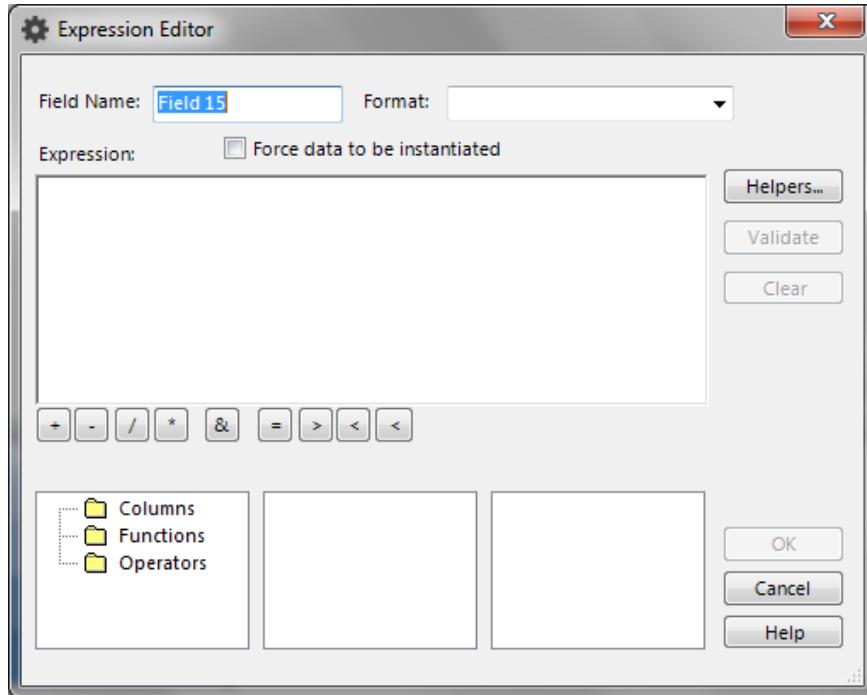
Figure 5.35: New Fields Added

	capital-loss	capital-loss1	capital-loss1 Description
296	1590	1590	(null)
297	0	(null)	0 replaced by null
298	0	(null)	0 replaced by null
299	0	(null)	0 replaced by null
300	2057	2057	(null)
301	0	(null)	0 replaced by null

5.5 The Expression Editor

The **Expression Editor** provides functionality to create new fields. The **Expression Editor** is access from the **Dataset Editor** dialog by clicking the **Add** button.

Figure 5.36: Expression Editor



The **Expression Editor** provides functionality to create new fields including:

- Assign a name to the new field using the **Field Name:** area

- Check **Force data to be instantiated**, if the original variable on which the transformation is based is removed from the dataset once the transformation is applied
- SQL code for a transformation can be typed directly entered and validated via the **Validate** radio button
- Expression creation is aided by predefined lists contained in folders on the left hand side of the editor. These are efficient ways to build expressions and avoid typing errors. Available folders are:
 - **Columns** Containing a list of all fields in the dataset
 - **Functions** Function categories are listed in the **Column Type:** area. Select any category to list functions in the **Column Name:** area.
 - **Operators** A list of common operators such as logical and bitwise operators, addition, subtraction, multiplication, division, etc

In addition, the **Expression Editor** contains a **Helpers...** button that launches an **Expression Helpers Wizard** that automates many common transformations and automatically writes the appropriate *SQL* code. Available transformations are summarized in table 5.4.

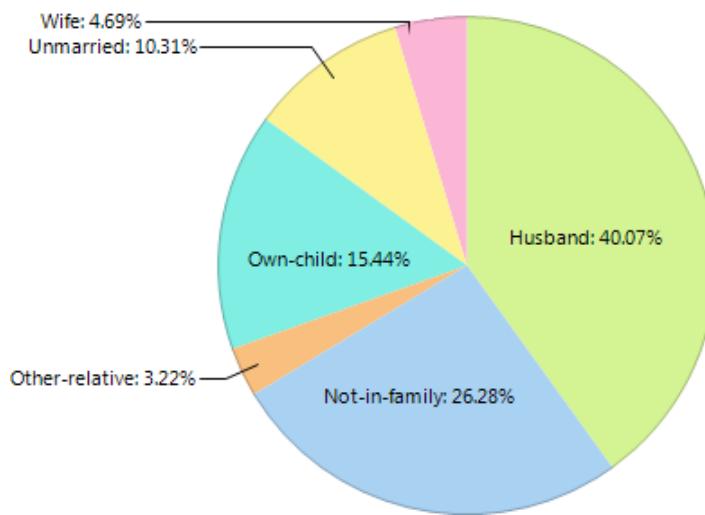
Table 5.4: Helpers Wizard Expression Types

Option	Description
Binning	Categorize continuous or categorical fields
Sum/Difference/Ratio of two numeric columns	Add, subtract or divide. Requires two fields
Generation of dummy fields	Creates a set of dummy, or indicator fields from a discrete field.
Interaction terms	Creates new fields based on the product of the selected fields and/or their squares
Logarithm transform	Creates a new column by taking the logarithm of an existing column
Optimal Binning	Uses Decision Tree to direct binning, based on a target variable
Outliers clipper	Coerces extremes values to a specific minimum and maximum
Power transform	Creates a new column by taking a base to the power of an existing column
Substitute missing values	Substituting missing values with user defined values
Weight	Creates the weight field in the case of weighted sample

5.5.1 Field Transformations: Binning

A pie chart of the field *relationship* is illustrated in figure 5.37.

Figure 5.37: Pie Chart of Relationship



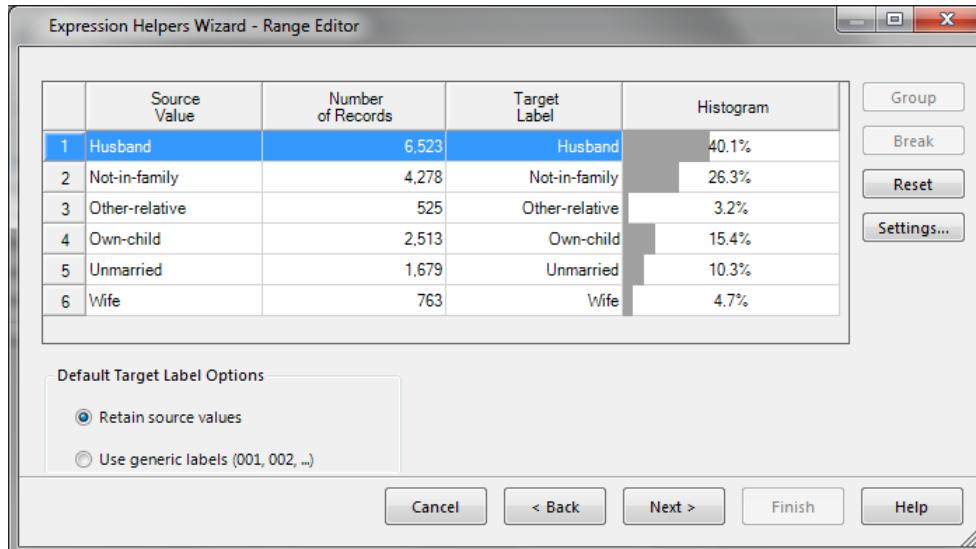
There are six categories, for modelling this may be useful and there are certainly not too many, however for the purposes of demonstration a new variable with two bins will be created.

The new variable will collapse the categories **Husband** and **Wife** into a new category called **Married**. All other categories will be collapsed into a category called **Other**.

The following steps outline the process. From the **Expression Editor**:

1. Assign the name *Rel2* to the new field
2. Activate the **Expression Helpers Wizard** by clicking the **Helpers...** button
3. From the **Expression Helpers Wizard – Select Expression Type** screen, select **Binning** from the **Expression type:** dropdown
4. Click **Next >**
5. In the following screen; **Expression Helpers Wizard – Binning Type**, click the **Discrete Transform** radio button
6. In the **Field** pane select the field *relationship*
7. Click **Next >** to launch the **Range Editor**

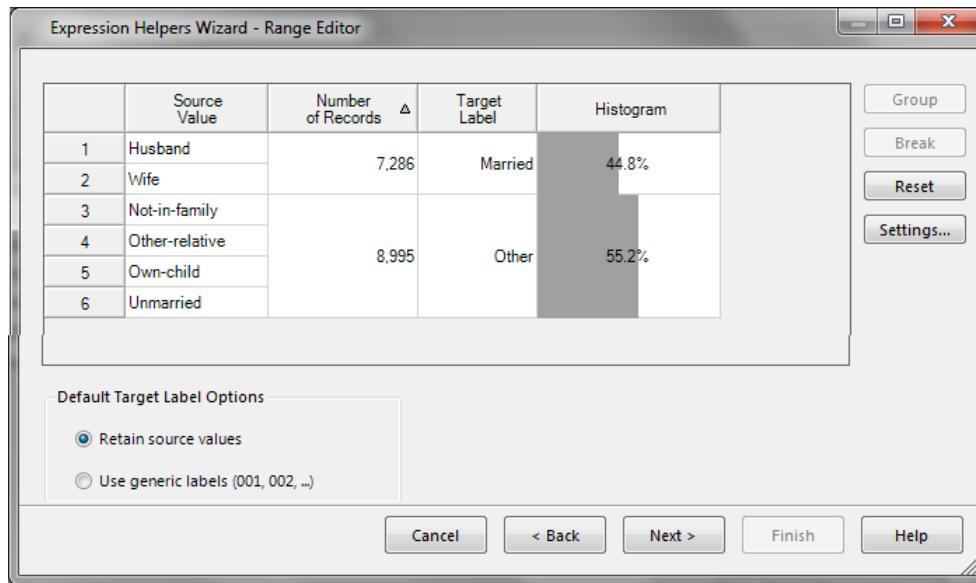
Figure 5.38: Expression Helpers Wizard - Range Editor



The **Range Editor** is used to group the categories *Husband* and *Wife* together into a new category called *Married*. All other categories can be grouped into a second category called *Other*.

1. **Ctrl-Select** the categories: *Husband* and *Wife*, then click the **Group** button
2. Assign the label *Married* in the **Target Label** area
3. Merge remaining categories and assign the label *Other*

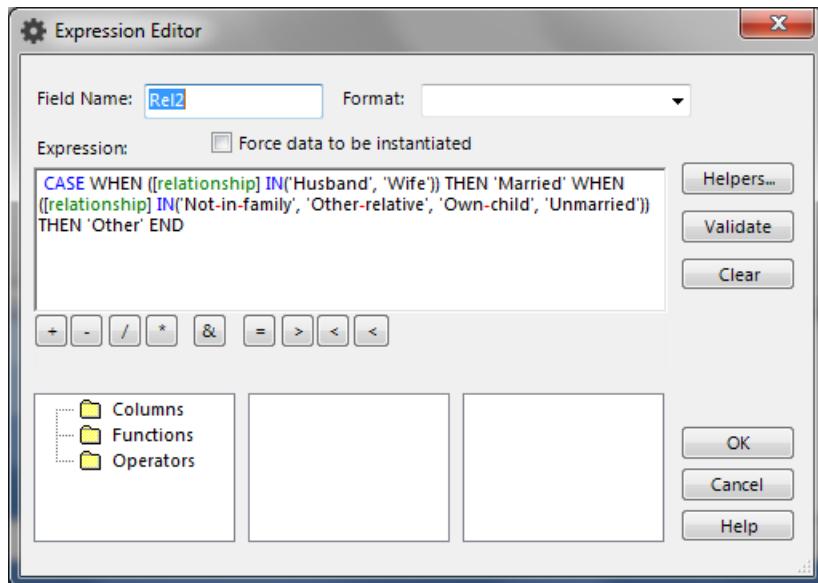
Figure 5.39: Binned Results



Clicking **Next >** launches the **Expression Helpers Wizard – Preview Expression** pane. This window illustrates the code generated from the binning specified previously (not shown).

Once complete, click **Finish** and the **Expression Editor** window will resume.

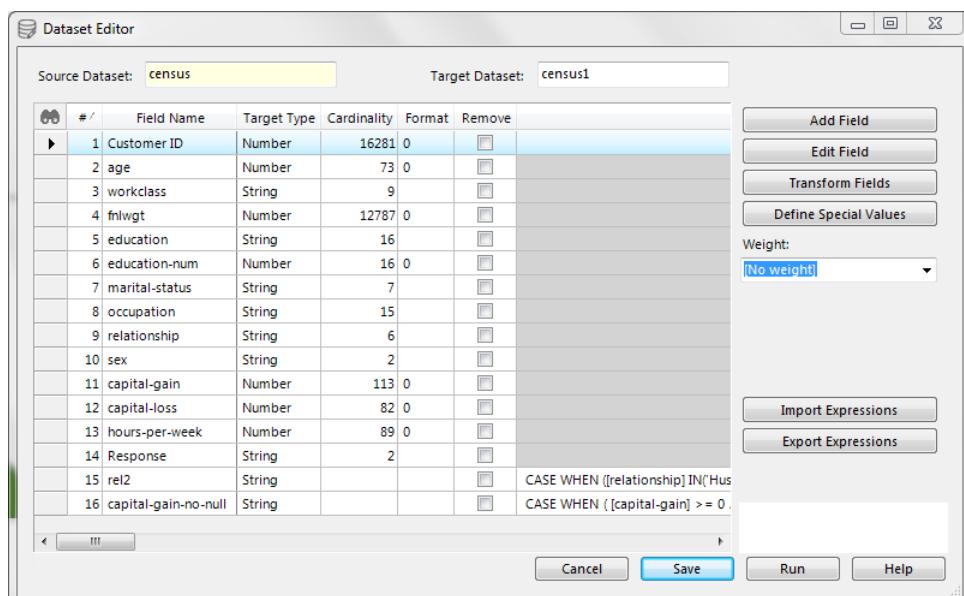
Figure 5.40: SQL Code for Binning Transformation



The correct *SQL* expression for the transformation appears in the **Expression Editor** pane. The expression could have been written manually, but here the **Helpers** feature automates code creation.

NOTE: If *SQL* code is typed directly without recourse to **Helpers**, the *SQL* syntax can be checked using the **Validate** button. Click **OK** to return to the **Dataset Editor** window.

Figure 5.41: Dataset Editor with Newly Created Field

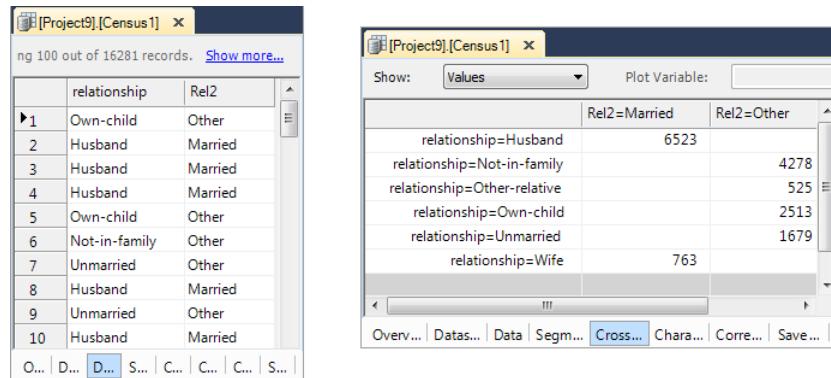


The new field appears with associated expression visible. The final step is to click the **Run** button in the **Dataset Editor** in order to generate a new dataset containing the computed field.

A new dataset node; **Census1**, is created on the canvas (not shown). This has the same name and symbolically references the created dataset in the **Project Pane**.

Viewing the data it can be seen that the new field exists, and running a cross-tabulation shows the binning generated as expected.

Figure 5.42: Data View and Crosstabs



5.5.2 Field Transformations: Transforming Multiple Fields

Applying the same transformation to multiple fields is common in *Data Mining*. For example, in the financial sector many monetary fields are log transformed to address skewed distributions.

Recall the distribution of the field *capital-loss* in figure 5.29. Fields exhibiting skewed distributions can have undesirable and unwanted effects on model coefficients. Fields such as these are generally addressed by log transforming their values.

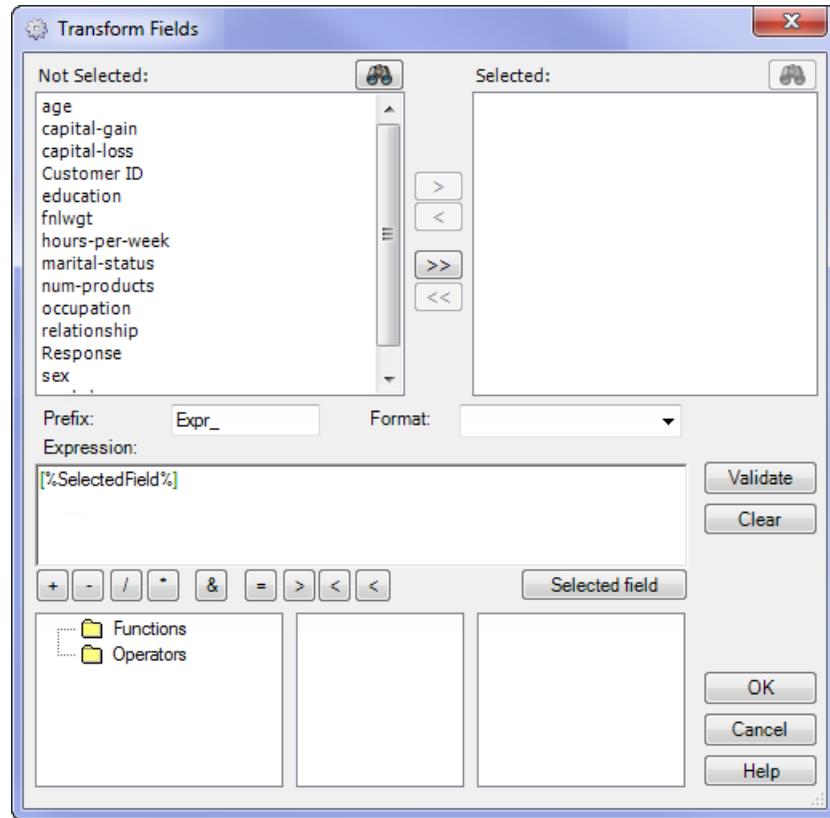
Log transforming fields compresses the range. Higher values are compressed to a greater degree than lower values. This has the desirable effect of modifying extreme values to a greater extent than smaller values while compressing the range and creating a more desirable distribution for modelling and reporting.

Applying such a transformation to multiple variables, 10's or even 100's, can be a time consuming process if applied on a one-by-one basis.

KnowledgeSTUDIO provides the capability to apply the same transformation to multiple fields simultaneously.

This capability is accessed via the **Transform Fields** radio button on the **Expression Editor** dialog.

Figure 5.43: Transform Multiple Fields

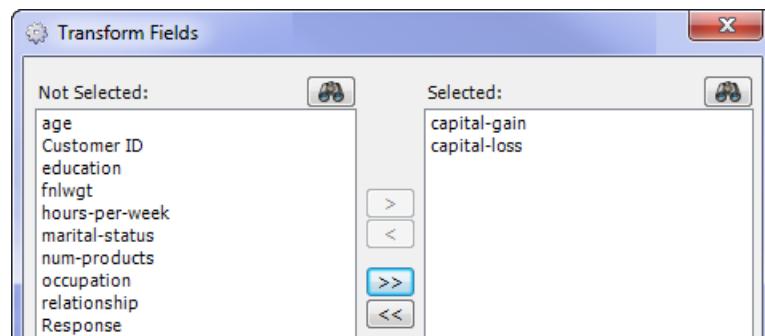


The dialog has distinct areas designed to aid the easy application of a transformation to multiple variables simultaneously.

The first area requires selection of the variables to transform. In this example a log transformation is applied to two fields: *capital-gain* and *capital-loss*.

Selecting both fields and move to the **Selected:** area as illustrated in figure 5.44.

Figure 5.44: Select Fields



The central section provides the ability to assign a prefix and format to the newly created variables. The

default **Prefix:** is **Expr_**, and the default **Format:** is initially blank. Both can be changed easily but for this demonstration, the default values are accepted, (not shown).

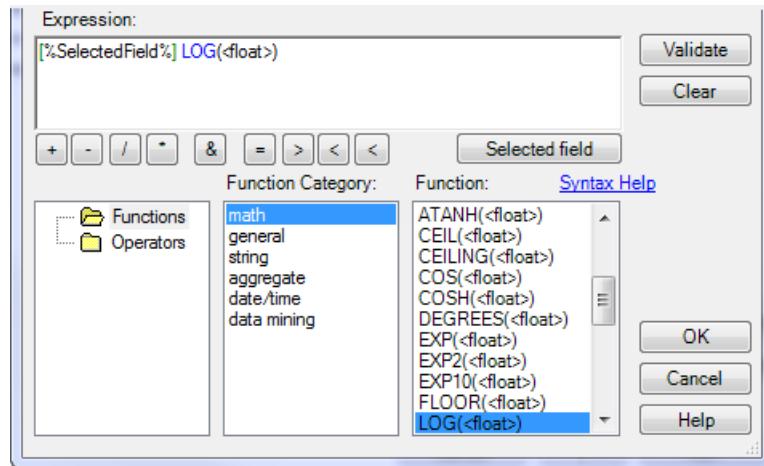
NOTE: In general, given the transformation, **KnowledgeSTUDIO** will assign an appropriate **Format:**.

The final section provides a space to specify the transformation. The options are identical to those presented when modifying variables in general with the addition of two elements:

- **[%SelectedField%]** A pattern that is a proxy for the variables to transform. The transformation is applied to each element in the list. Must be present for the transformation to proceed
- **Selected Field** Clicking generates the pattern: **[%SelectedField%]**

To apply the log transformation, select the **LOG<float>** function from the **math Function Category:** and double click to move to the **Expression:** pane as shown in figure 5.45.

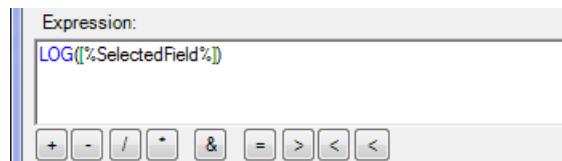
Figure 5.45: Selecting the LOG Function



At this point there is a need to move the pattern inside the brackets to apply the transformation to all fields.

First, highlight and delete the pattern **[%SelectedField%]**. Next, highlight all elements inside the **LOG** brackets and click the **Selected field** radio button to paste the pattern.

Figure 5.46: Applying the Transformation



NOTE: Copying and pasting the pattern also suffices!

One thing to bear in mind is that the **LOG** of zero is undefined, if any record has a value of 0 then the transformation will return a null. To address this, a small constant, 1, is added as illustrated in figure 5.47.

Figure 5.47: Applying the Transformation



As the **LOG** of 1 is zero, this will ensure values of 0 are retained as is.

Once complete, click **OK** to return to the **Dataset Editor**. Notice that two new fields are added. The **Expression** column contains the transformation applied.

Figure 5.48: New Fields Added

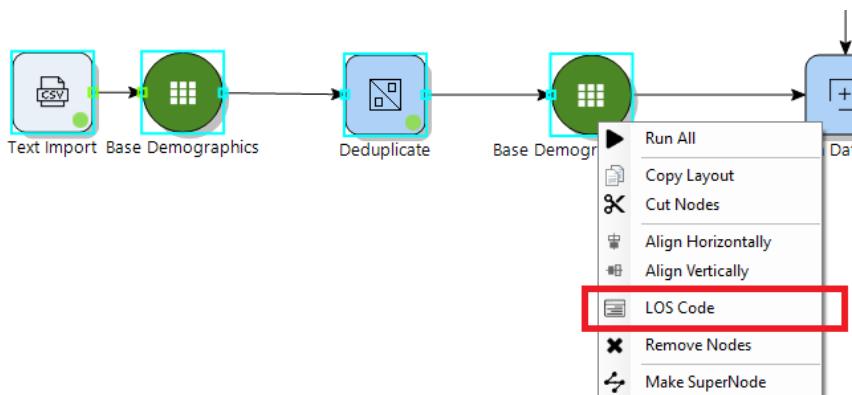
12	capital-loss	Number	82	0	
13	num-products	Number	16	0	
14	capital-gain	Number	110		
15	Expr_capital-gain	Number			LOG(1+[capital-gain])
▶	16 Expr_capital-loss	Number			LOG(1+[capital-loss])

Clicking **Run** adds the new fields to the dataset. Examining one of the fields, *capital-loss*, it can be seen that the distribution is not as skewed as before and the range is much tighter, not shown.

5.6 LOS Code Generation

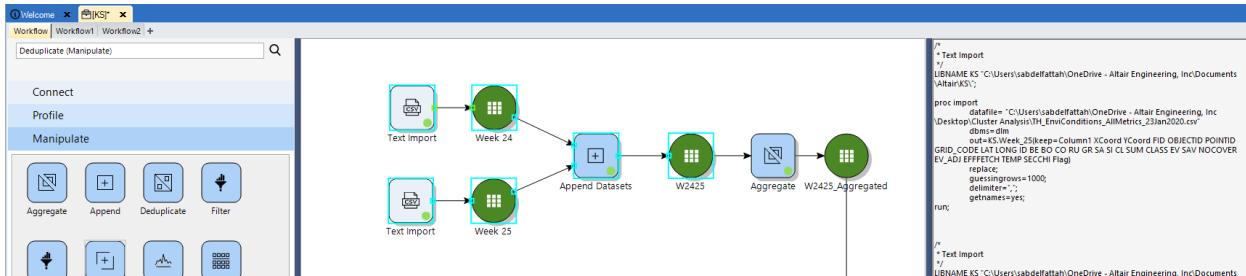
LOS code can be generated for all or part of a **Workflow**, including transformations. To create the **LOS** code, highlight all or part of a **Workflow**, right click and select the option **LOS Code**

Figure 5.49: LOS Code Generation



The **LOS** code is generated and evident in a new pane to the right hand side of the **Workflow**. The new pane can be sized or opened and closed by double clicking its associated handle.

Figure 5.50: Workflow with LOS Code



5.7 Conclusion

This chapter introduced **KnowledgeSTUDIO** functionality for manipulating data. The **Manipulate** palette contains a host of process nodes suitable for most data manipulation tasks.

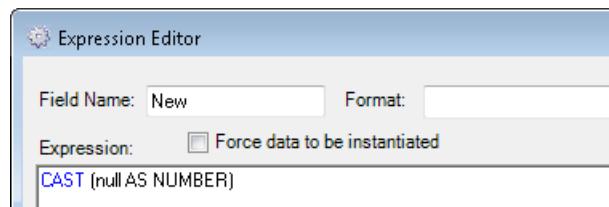
As a result of completing of this chapter, users should be aware of **KnowledgeSTUDIO** functionality to transform data at the dataset and field level. Specifically:

- Append files
- Merge files
- Remove duplicates
- Aggregate data
- Field transformation capabilities through the use of the **Dataset Editor**, **Expression Editor** and **Helpers**

Exercises

1. Append Datasets:

- (a) Import the files:
 - i. *FileLeft.xlsx*
 - ii. *FileRight.xlsx*
- (b) Profile both files using **KnowledgeSTUDIO** functionality
- (c) How many cases and fields are in each?
- (d) Append the files, retain fields from both
 - i. If fields exist in one but not the other, create new fields with null values
 - ii. Any new *NULL* field is automatically String, to ensure numeric, a cast statement is required as:



- (e) Are the results as expected?

2. Join Datasets:

- (a) Import the files:
 - i. *FileOne.xls*
 - ii. *FileTwo.csv*
- (b) Explore the files using **KnowledgeSTUDIO** functionality
- (c) Is there a common field across files to enable a join?
- (d) Merge both files retaining all records in *FileOne.xls*

3. Aggregate

- (a) Import the file *Census.xls*
- (b) Aggregate the file on the field *Response*
- (c) Choose some continuous and categorical fields selecting an appropriate aggregate in each
 - i. **NOTE:** For categorical fields choose the **MODE**
- (d) Explore the results using **KnowledgeSTUDIO** functionality

4. Define Special Values

- (a) Using the variable *hours-per-week* create **ONLY** a **Description Field** identifying records with a value of 99. Leave the **Description field default value:** as *null*.
- (b) Using the same variable, add a field that replaces the value 99 with *null*. Create a dichotomous descriptor with values of 0 if the **Special Value** is not present and 1 if it is.
- (c) What proportion of records have a value of 99 for the variable *hours-per-week*.

5. Field Transformations:

- (a) Using the Helper functions to create some new fields such as:
 - i. New binning fields
e.g. Bin the continuous field *capital_gain* into three **Equal Width** bins
 - ii. Something else of your choosing
- (b) Create a weight variable based on the field *sex*
- (c) Ensure that all categories of *sex* are equally represented when the weight is applied

6. LOS Code Generation

- (a) Highlight all or part of the **Workflow** and generate the LOS code

Chapter 6: Variable Selection, Sampling and Partitioning

Variable selection and partitioning are essential aspects of any *Data Mining* endeavour.

Statistical modelling techniques can assess the nature and extent of the relationship between a set of **Independent Variables** and a **Dependent Variable**. Of course this is elementary, however, determining which variables should be included from a larger pool is not as clear cut.

Some analysts rely on specific techniques to weed out variables not related to the **Dependent Variable**, for example, **Stepwise** methods.

In the case where a lot of potential predictors exist and the dataset size is relatively large, this can be computationally time consuming. It is not advised to solely rely on these methods but use them in conjunction with other variable selection techniques to highlight potential predictors.

Partitioning and sampling are synonymous and are well used and documented and the reasoning behind partitioning is practical.

A modelling dataset is divided into at least two parts. One part is used to create a model and the other(s), hidden from the modelling process, used to determine not only how well the model will predict new information, but also whether the model has been overfit on the data used in its development.

KnowledgeSTUDIO includes both variable selection methods and partitioning/sampling capabilities. The objectives of this chapter are:

- To outline the various variable selection methods available in **KnowledgeSTUDIO**
- Illustrate how to generate samples and create partitions using the **Partition** node

6.1 Variable Selection

Variable selection is a conceptual approach with many possible implementations. For example, variables may be removed from a dataset based on screening measures such as:

- Min = max
- Low or no variability
- All records concentrated in one category
- High proportion of records with missing values

These options represent some of the possibilities to take into account when assessing variables for inclusion in a model. **KnowledgeSTUDIO** complements this approach with additional statistical and graphical methods to aid in variable selection, namely:

- | | |
|-----------------------------|---|
| • Segment Viewer | Available from the Segment Viewer tab of any dataset |
| • Variable Selection | A node available from the Manipulate palette |

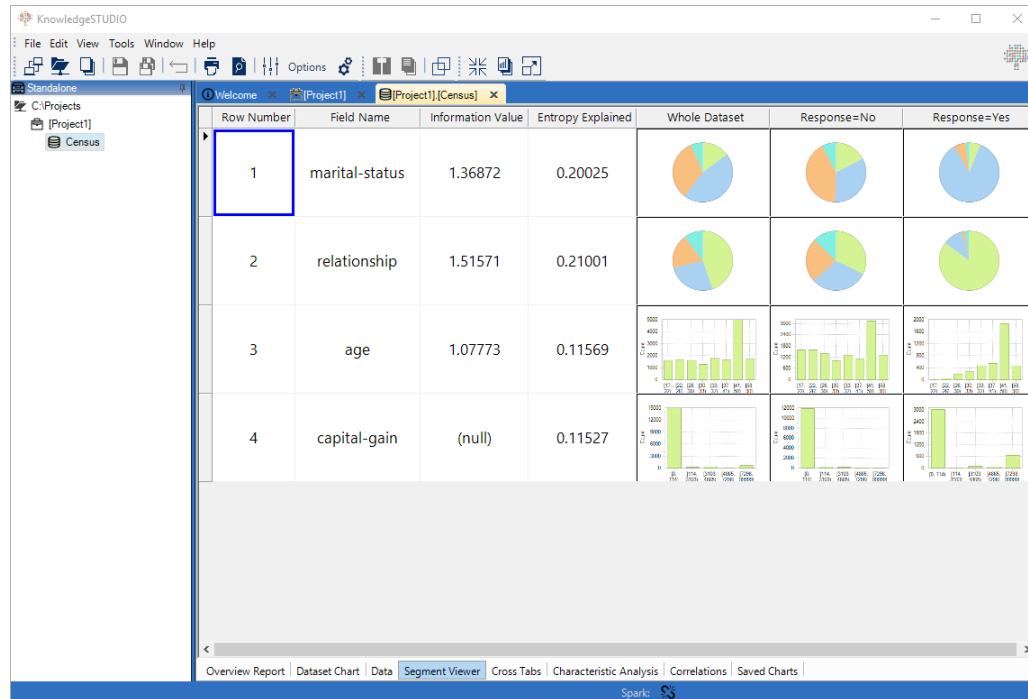
NOTE: The **Variable Importance** node, available from the **Evaluate** palette, can be used to visualize the relative ranked importance of potential predictors. It can also be used as a means of **Variable Selection**. This node is introduced in a later chapter.

6.1.1 The Segment Viewer

The **Segment Viewer** is a primary characterizing and investigative method that can be applied to identify potentially good predictors of a **Dependent Variable**.

For example, assuming there is interest in modelling the variable *Response* from the **Census** dataset, the **Segment Viewer** is an ideal starting point to assess potential predictors.

Figure 6.1: The Segment Viewer



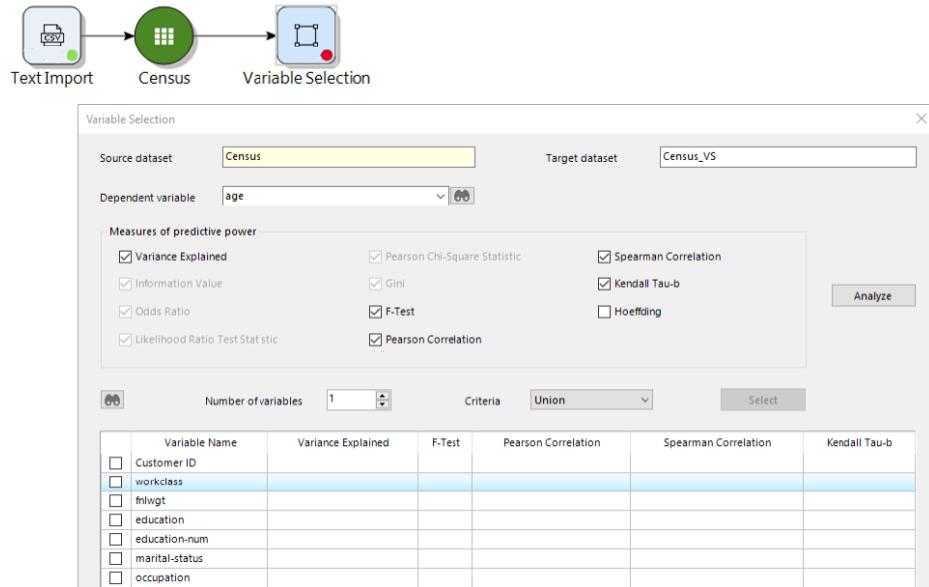
The visual representations added to the capability to sort based on **Information Value** or **Entropy Variance**, which pushes the best predictors to the top, provide a neat mechanism to identify good predictors.

For example, for the variable *age*, it can be said that those in the *No* category are generally younger than those in the *Yes* Category. Given this contrast, *age* may be useful in predicting the **Dependent Variable**.

6.1.2 Variable Selection Node

The **Variable Selection** node is found on the **Manipulate** palette. Connecting to the **Census** dataset and opening reveals available options.

Figure 6.2: Variable Selection Node



The node outputs a new dataset including selected variables based on generated statistics available from the **Measures of Predictive Power**. The statistics change based on the **Dependent Variable** type.

The current **Dependent Variable** is *age*, this is alphabetically the first variable in the dataset and the reason for its selection. It is a continuous variable and all available statistics are appropriate for this type of **Dependent Variable**.

Clicking the **Analyse** button calculates statistics for all variables.

Variables can be listed in ascending or descending order of any statistic simply by clicking the associated column header.

At this point variables can be selected for inclusion in a new dataset by checking the box to the left of each. This is of course identical to using the **Measures of Predictive Power**.

The additional elements available here are:

- A new dataset containing only selected predictors can be created
- Variable selection can be automated based on the **Number of Variables** and **Criteria** options

The creation of a new dataset is self explanatory. Automating variable selection is based on the use of **Number of Variables** and **Criteria** in conjunction with the **Select** radio button.

The application of these elements is explained in table 6.1.

Table 6.1: Automate Variable Selection Options

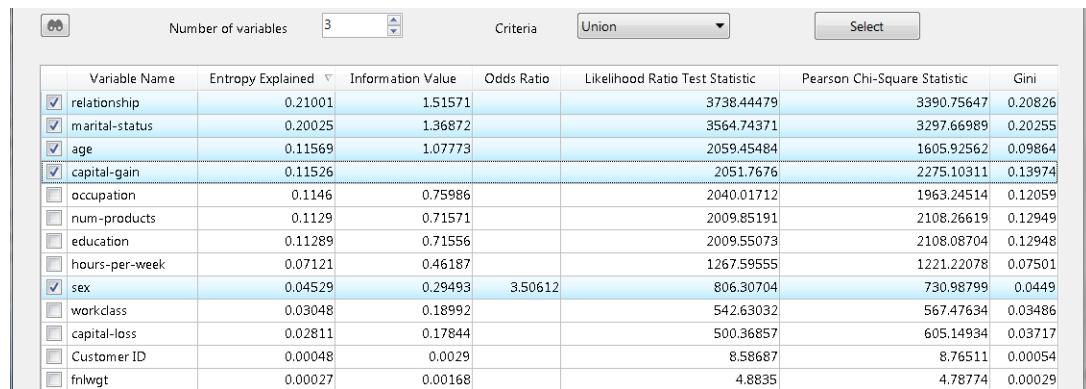
Option	Description
Number of Variables	Number of variables to select
Criteria	<p>Determines how the Number of variables are identified:</p> <ul style="list-style-type: none"> • Union An OR option: Select variables that are the top Number of variables for any measure. A value of 2 will select the top 2 variables for each measure • Intersection An AND option: Select variables that are in the top Number of variables across all measures. A value of 5 will select a maximum of 5 variables in total

For this demonstration, **Response** is selected as the **Dependent Variable**. All statistics are calculated using the **Analyse** radio button.

Once complete, the **Number of variables** value is set to 3 and the criteria set to **Union**.

Clicking **Select** automatically selects the top 3 predictors across each measure. This results in a total of 5 variables selected.

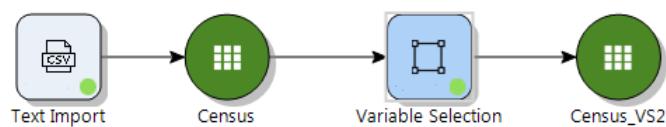
Figure 6.3: Automating Variable Selection



Variable Name	Entropy Explained	Information Value	Odds Ratio	Likelihood Ratio Test Statistic	Pearson Chi-Square Statistic	Gini
relationship	0.21001	1.51571		3738.44479	3390.75647	0.20826
marital-status	0.20025	1.36872		3564.74371	3297.66989	0.20255
age	0.11569	1.07773		2059.45484	1605.92562	0.09864
capital-gain	0.11526			2051.7676	2275.10311	0.13974
occupation	0.1146	0.75986		2040.01712	1963.24514	0.12059
num-products	0.1129	0.71571		2009.85191	2108.26619	0.12949
education	0.11289	0.71556		2009.55073	2108.08704	0.12948
hours-per-week	0.07121	0.46187		1267.59555	1221.22078	0.07501
sex	0.04529	0.29493	3.50612	806.30704	730.98799	0.0449
workclass	0.03048	0.18992		542.63032	567.47634	0.03486
capital-loss	0.02811	0.17844		500.36857	605.14934	0.03717
Customer ID	0.00048	0.0029		8.58687	8.76511	0.00054
fnlwgt	0.00027	0.00168		4.8835	4.78774	0.00029

Clicking **Run** creates the dataset **Census_VS**. Opening the dataset reveals 6 variables: the 5 selected and the **Dependent Variable**

Figure 6.4: New Dataset with Selected Variables



6.2 Partitioning Features in KnowledgeSTUDIO

Partitioning features available in **KnowledgeSTUDIO** are provided through the **Partition** node contained in the **Manipulate** palette.

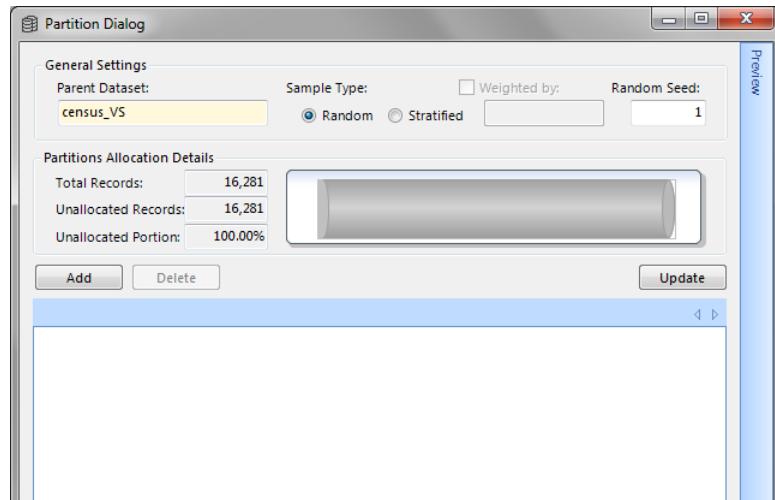
The **Partition** node provides the following functionality:

- Create single or multiple; random, or stratified, partitions
- Set the random seed, to allow repeatability of results by replicating partition structure
- Maintain a variable weighting within a partition
- Graphically view partition allocation details, including the number and proportion of cases assigned to each partition, and unallocated cases

To use the **Partition** node, create a new project and import the file: *Census.xlsx*. Drag the **Partition** node from the **Manipulate** palette on to the **Workflow** canvas and connect to the imported file.

Access the **Partition** node dialog by either double click the **Partition** node or right clicking and selecting the option **Modify**.

Figure 6.5: Partition Dialog



The **Partition** dialog provides the ability to create either randomly sampled or stratified partitions. Additional features are described in table 6.2.

Table 6.2: Partition Dialog Options

Option	Description
Parent Dataset	The dataset to be partitioned

Sample Type	Random: Dataset records are selected at random for each partition. Stratified: A sample with a specific user-defined distribution of a selected field (called the stratification variable). Note that the default Sample Type is Random
Random Seed	The purpose of the random seed is to provide repeatable results of procedures that involve randomness. All factors being equal, partitions created with the same Random Seed will be identical
Weighted by	If a weight variable is defined in the parent dataset, then the name of that field will be displayed. The partitions being defined can be based on either weighted or non-weighted record counts, the default being weighted. The weighted option is available only for the random sample type. See the Weight topic in the help menu for further details
Partition Name	The name of the partition being defined
Sample Selection	The requested partition size can be specified either as a number of records or the percentage of the total
Add / Delete	Add or delete the selected partition
Update	Updates partition details to populate or apply changes
Partitions Allocation Details	Total Records: total number of records in the parent dataset Unallocated Records: records not assigned to any partition Unallocated Portion (%): available records as a percentage

The following demonstrations illustrate how to create **Random** and **Stratified** partitions.

6.2.1 Creating a Randomly Sampled Partition

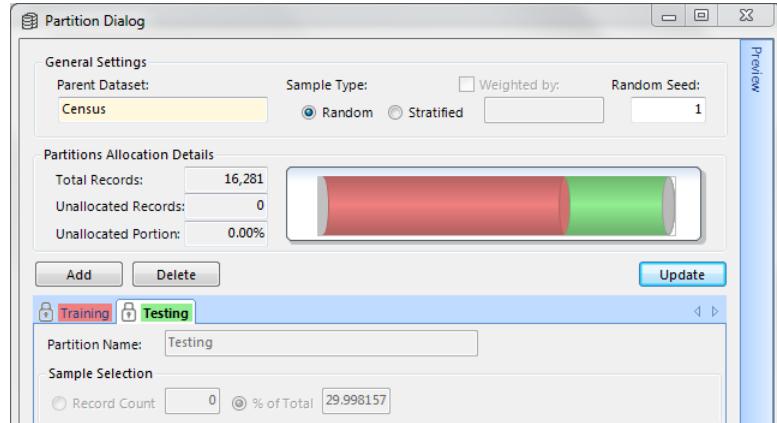
In this example two partitions are created, where 70% and 30% of the parent dataset are randomly assigned to each.

This is a common approach when modelling, as models can be developed on one partition and validated on the other.

To create the partitions, from the **Partition** node dialog:

1. Select **Random** as the **Sample Type**. This is the default setting
2. Click **Add** twice to create two new partitions
3. Two new tabs appear in the lower part of the dialog

Figure 6.6: Tabs Added

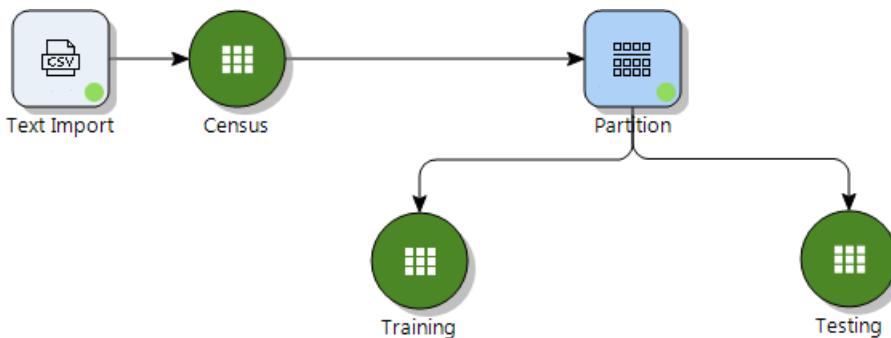


1. Select the **Census_P1** partition tab
2. Rename this to **Training**
3. Select % of Total under **Sample Selection**, and assign 70% of cases to the **Training** partition
4. Click **Update** to populate the partition
5. Select the **Census_P2** partition tab
6. Rename this to **Testing**
7. Assign 30% of cases to the **Testing** partition
8. Click **Update** to populate the partition

Once options have been set the cylinder representing dataset records is now colour coded to reflect the proportion assigned to each new partition. Since the partitions were created at the same time, they are disjoint and the records were sampled without replacement.

Click **Run** to create the partitions.

Figure 6.7: Partitions Added to Workflow



NOTE: Once partitions are generated they are visible in the **Project Pane** and symbolically represent on the **Workflow** canvas, as previously illustrated.

6.2.2 Appendix: Creating a Stratified Sample

Stratified partitions are used to ensure adequate representation of smaller population subgroups in a sample.

The creation of a stratified sample requires:

- The identification of a stratification variable
- Specification of sample size for each of the stratification variables' subgroups, or
- A sample size for the partition, and percent representation for each of the stratification variables' subgroups

Options available when selecting **Stratified** as the **Sample Type** from the **Partition** dialog are detailed in table 6.3.

Table 6.3: Stratified Partition Options

Option	Description
Field	Stratification field. Only fields of cardinality ≤ 100 can be selected
Stratify by Records / Percentage	Records: specify number of records requested for each subgroup Percentage: specify partition size as percentage of parent dataset
Partition Size	Size of resulting partition as a number or percentage of the parent dataset
Partition Name	Partition name
Unique Value	Stratification field categories/values
% of Total	Stratification variable distribution in parent dataset
# Available Records	No. of records in stratification variable category
# Requested	Number of cases from subgroup in resulting partition. Available only if Stratify by Records option chosen
% Requested	Percentage of resulting partition assigned to selected subgroup. Available only if Stratify by Percentage chosen. Note: must specify partition size
Auto-calculate	If Percentage selected as Stratify option. Select this option to automatically calculate partition size as number of records or percentage of parent. Note, that strata percentages must be input

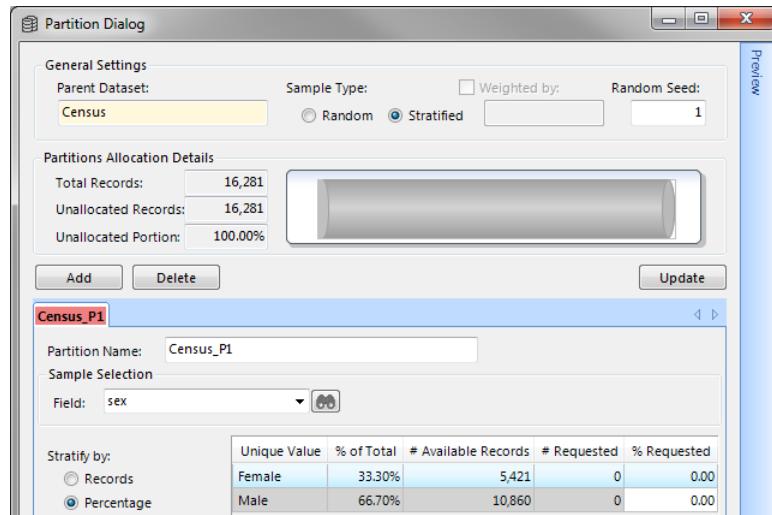
To illustrate stratified partitioning, a sample of size **6000** is drawn from the parent dataset **Census**. The resulting sample will be stratified by *sex* with equal representation in the resulting sample for males and females.

Use the current project or create a new one and either import the file *Census.xlsx* or use the **Dataset Link** node from the **Manipulate** palette to include the **Census** dataset in the **Workflow**.

Connect the **Partition** node and activate its dialog. Set the following options:

1. Select **Stratified** as the **Sample Type**: validation, datasets
2. Click **Add** to create a new partition
3. Name the partition; **Sex_Strat**
4. From the field dropdown choose **sex**

Figure 6.8: Stratified Partition



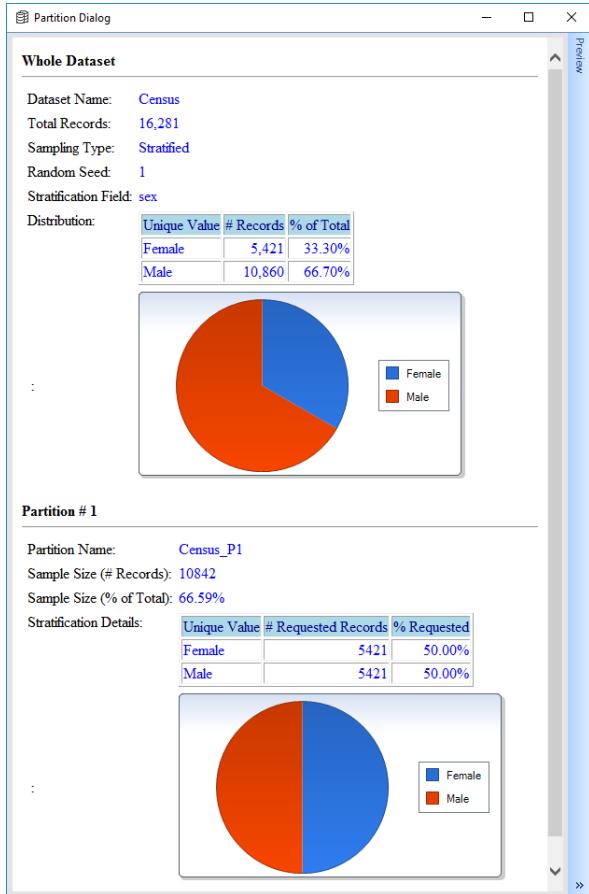
Next, specify the resulting partition size and assign subgroup proportions.

1. Choose **Percentage** in the **Stratify by:** area
2. Specify 6000 in the **# Records** slot in the Partition Size area
3. Type 50 in the % Requested slot for both **Male** and **Female** groups
4. From the field dropdown choose **sex**

At any point while specifying the partition parameters, you can **Preview** the partition summary report by clicking the arrow at the bottom of the vertical blue bar on the right side of the dialog.

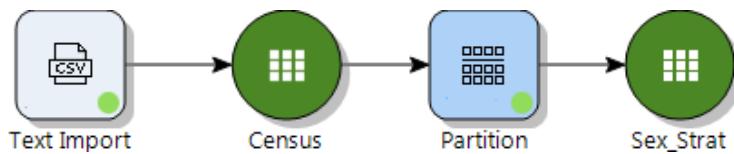
In the case of **Stratified** partitions, the summary includes the pie charts of the requested distribution of the stratification variable for each partition defined.

Figure 6.9: Partition Preview



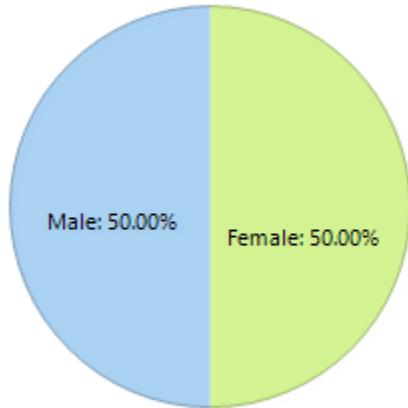
The preview shows the before and after results as well as the partition size. Click **Run** to generate the partition. Figure 6.10 illustrates the results.

Figure 6.10: Stratified Partition Added



Using the **Dataset Chart** tab and illustrating the variable *sex*, the stratification is confirmed: 50% of cases in the resulting sample are **Male** and 50% are **Female**, with sample size 6000.

Figure 6.11: Distribution of Sex



sex	Frequency	Percent
► Female	3000	50.00 %
Male	3000	50.00 %

6.3 Conclusion

This chapter introduced the variable selection and partitioning features available in **KnowledgeSTUDIO**.

On completion of this chapter users should be able to:

- Use the various variable selection methods available in **KnowledgeSTUDIO** including the **Segment Viewer** and the **Variable Selection** node
- Illustrate how to generate samples and create partitions using the **Partition** node

Exercises

1. Variable Selection:

- (a) Import the files *Census.xlsx* if it is not already in the project.
- (b) Investigate and identify variables that are potentially good predictors of the variable *Response*.
 - i. Use the **Segment Viewer** as a preliminary tool to visualize distributions and identify potentially good predictors.
 - ii. Add a **Variable Selection** node from the **Manipulate** palette and use it to create a dataset of the top 5 predictors using the **Criteria: Union**. How many variables have been selected?
 - iii. Re-run but change the **Target Dataset** name to something of your choosing, and change the **Criteria: to Intersection**. Note that changing the dataset name will not overwrite the previous selections and makes comparison easier.
 - iv. Compare the results to that when **Union** was selected.
 - v. Compare the results to the list generated from using the **Measures of Predictive Power**.

2. Random Partitions:

- (a) Import the files *Census.xlsx* if it is not already in the project.
- (b) Drag the **Partition** node from the **Data Transformations** area onto the **Workflow** canvas.
- (c) Familiarize yourself with its functionality. Use the **Help** button!
- (d) Create two random partitions of equal proportion:
 - i. Name them appropriately.
 - ii. Use the **Update** button to view allocations.

3. Stratified Partitioning:

- (a) Create a stratified partition based on the variable *relationship*.
- (b) Stratify by **Records** and ensure categories are equally represented with 100 records in each.
- (c) Verify results using the **Dataset Chart** tab.

Chapter 7: Understanding Decision Trees

7.1 Introduction

A **Decision Tree** is a segmentation model that successively splits a dataset based on the relationship between a dependent, or target variable and a set of independent, or predictor variables.

Decision Trees are a versatile *Data Mining* technique accommodating categorical or continuous dependent variables. Similarly, inputs, can also be of any type. As **Decision Trees** make no assumptions about the data, they are an ideal and easy to use method, not only to model an outcome, but also to:

- Explore an unfamiliar dataset
- Identify potentially good predictors to use in other models
- Compliment alternative modelling techniques such as **Logistic Regression** and **Neural Networks**

Decision Trees classify records into homogeneous subgroups by successively segmenting data using known predictor values. The results of a **Decision Tree** are represented in a tree-like structure. Trees are comprised of *nodes* which represent and contain subsets of the data.

As a result of completing this chapter users should be able to:

- Build **Decision Trees** with **Altair KnowledgeSTUDIO**
- Understand and employ the various growing methods to iteratively, interactively and automatically build **Decision Trees**
- Prune tree nodes to increase acceptability
- Create Model Instances to store successive models

7.2 The Basics

Decision Trees are a popular predictive modelling technique due to:

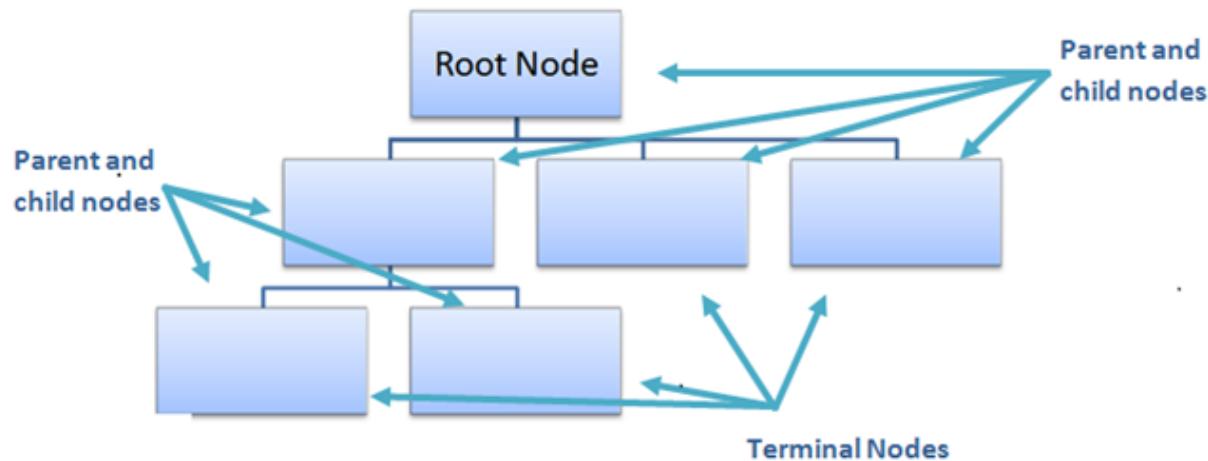
- Ease of use
- Ease of understanding
- The ability to incorporate both categorical and continuous independent and dependent variables
- There are no data assumptions
- Large numbers of independent variables can be incorporated
- Non-linear relationships and interactions can be described and incorporated more easily than classical methods
- Ease of deployment

7.3 Components of a Decision Tree

There are different types of nodes within a **Decision Tree** defined as follows:

- **Root node:** Contains all cases in the dataset, the first node in any tree
- **Parent node:** A node that is further split to form child or terminal nodes
- **Child node:** Arises from a parent node, can be split to form further child or terminal nodes variables
- **Terminal node:** Arises from a parent node, does not undergo further splitting

Figure 7.1: Decision Tree Structure



NOTE: Nodes may be of more than one type, e.g. a node can be both a parent and a child simultaneously.

7.4 Dependent and Independent Variables

Decision Trees aim to predict a dependent variable from a series of independent variables.

The dependent variable, also referred to as the target or *DV*, is the focus of the analysis. Ideally this variable should have no missing values. The dependent variable can be categorical or continuous.

The independent variables, also referred to as predictors or *IVs*, are those used to predict the dependent variable. The independent variables can be a mixture of categorical and continuous variables.

Prior to developing any type of model, including **Decision Trees**, it is wise to remove any independent variables exhibiting the following characteristics:

- Contain a large percentage of missing values
 - **KnowledgeSTUDIO Decision Trees** can incorporate and deal with missing values. However if the values represent uncertainty rather than having useful business meaning, it is wise to exclude them

- Fields that have all/majority of records in the same category
- Variables with little or no variability
- Unique identifiers, such as customer id, telephone number etc
- Direct correlations with the dependent variable, those that can be described as a proxy measure of the dependent variable
- Fields derived from the dependent variable
- Independent variables that are highly correlated with each other. Note that this is not an issue for **Decision Trees**

7.5 Altair Decision Tree Algorithms

There are a variety of **Decision Tree** algorithms available. Some algorithms use a purely mathematical basis to assess predictor variables and segment the dataset while some incorporate statistical methodologies.

Regardless of the algorithm used their goal is the same: to segment the dataset into subgroups that homogenize the target variable.

KnowledgeSTUDIO provides the following **Decision Tree** algorithms.

Table 7.1: Decision Tree Algorithms Supported in KnowledgeSTUDIO

Algorithm	Description
Unadjusted – Raw P-value Measure*	The p-value of a Chi-Squared statistical test is used to build the trees. This is the equivalent to CHAID Trees. KnowledgeSTUDIO provides two versions of this algorithm
Adjusted P-value – Bonferroni Adjustment Measure*	The Bonferroni adjustment uses multiple test comparisons and adjusts the significance level accordingly
Entropy Variance – Non P-value Information Gain Measure**	The value of entropy is a number in the interval [0, 1]. Entropy of one indicates high dispersion whereas values closer to 0 indicate increased homogeneity
Gini Variance – Non P-value Gini Measure***	This measure uses the Gini coefficient, or Gini index, to determine the importance of each split in a tree. Value varies between 0 and 1. Values closer to zero indicates increased homogeneity

*Equivalent to **CHAID** variants

Similar to **C5 methods

***Equivalent to **CART** variants

7.6 Decision Tree Modelling in KnowledgeSTUDIO

KnowledgeSTUDIO **Decision Tree** capabilities are found in the **Model** palette. Table 7.2 details the **Decision Tree** modelling nodes available. Note that only modelling nodes available with a

KnowledgeSTUDIO license are illustrated and detailed.

Table 7.2: Model Palette

Palette	Node	Description
Decision Tree	Decision Tree	Use to create Decision Tree models
Strategy Tree	Strategy Tree*	Use to create a Strategy Tree model
Bagging	Random Forest*	Use to create a Random Forest ensemble mode
Boosting	Bagging*	Use to create a Bagging ensemble model
Cluster Analysis	Boosting*	Use to create a Boosting ensemble model
Constrained Regression	Logistic Regression*	Use to create a Logistic Regression model
Deep Learning	Regularization**	Apply regularization to model a linear or discrete outcome
Factor Analysis	Constrained Regression**	Use to create a Constrained Regression model
Linear Regression	Scorecard**	Scale a Logistic Regression model to a standard scorecard
Logistic Regression	Scorecard Editor**	Use to calibrate a standard scorecard
Market Basket Analysis	Reject Inference**	Use to include rejected records when developing a scorecard
PLS Regression	Deep Learning*	Use to develop a Neural Network model
Principal Component Analysis	Linear Regression*	Use to develop a Linear Regression model
Random Forest	Cluster Analysis*	Use to develop a Cluster Analysis model
Regularization	Market Basket Analysis*	Use to develop association models
Reject Inference	Principal Component Analysis*	Use to reduce a set of variables and assess latent factors
Scorecard	Regularization**	Use to perform regularization
Scorecard Editor	PLS Regression**	Use to develop PLS Regression model
Survival Analysis		

*Covered in a later chapter

**Not covered in this course

NOTE: Some nodes are greyed out. This is related to the license and functionality unlocked as a result of the currently installed license. Grey nodes are not part of the **KnowledgeSTUDIO** license.

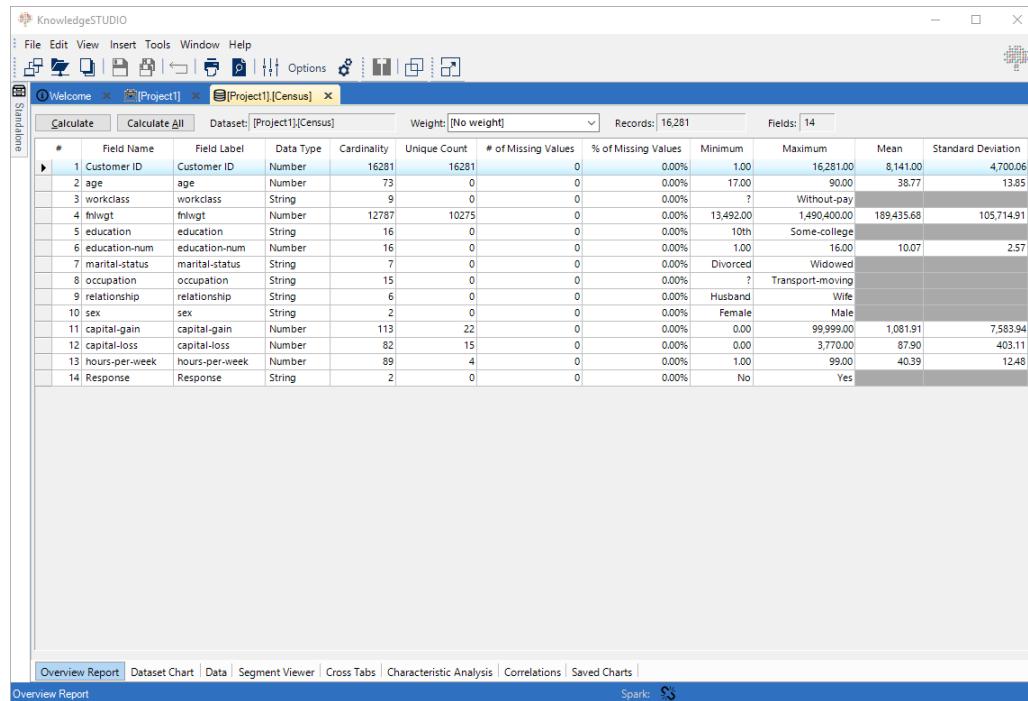
7.7 Setting the Scene: Data

The data for this example and following chapters is contained in an excel file: *Census.xlsx*.

Create a new project and, using the *Excel* node from the **Source** palette, locate and import the file: *Census.xlsx*.

Once imported, initial data profiling can commence. The **Overview Report** tab provides a host of univariate statistics to assess variables.

Figure 7.2: Census Overview Report

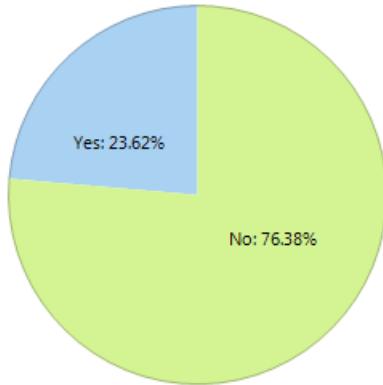


#	Field Name	Field Label	Data Type	Cardinality	Unique Count	# of Missing Values	% of Missing Values	Minimum	Maximum	Mean	Standard Deviation
1	Customer ID	Customer ID	Number	16281	16281	0	0.00%	1.00	16,281.00	8,141.00	4,700.06
2	age	age	Number	73	0	0	0.00%	17.00	90.00	38.77	13.85
3	workclass	workclass	String	9	0	0	0.00%	?	Without-pay		
4	fnlwgt	fnlwgt	Number	12787	10275	0	0.00%	13,492.00	1,490,400.00	189,435.68	105,714.91
5	education	education	String	16	0	0	0.00%	10th	Some-college		
6	education-num	education-num	Number	16	0	0	0.00%	1.00	16.00	10.07	2.57
7	marital-status	marital-status	String	7	0	0	0.00%	Divorced	Widowed		
8	occupation	occupation	String	15	0	0	0.00%	?	Transport-moving		
9	relationship	relationship	String	6	0	0	0.00%	Husband	Wife		
10	sex	sex	String	2	0	0	0.00%	Female	Male		
11	capital-gain	capital-gain	Number	113	22	0	0.00%	0.00	99,999.00	1,081.91	7,583.94
12	capital-loss	capital-loss	Number	82	15	0	0.00%	0.00	3,770.00	87.90	403.11
13	hours-per-week	hours-per-week	Number	89	4	0	0.00%	1.00	99.00	40.39	12.48
14	Response	Response	String	2	0	0	0.00%	No	Yes		

It can be seen from figure 7.2, that this dataset has 14 fields and 16,281 records. The fields include some demographics, financials and a variable called *Response* with two values: Yes and No.

This field reflects whether a recent campaign has been responded to. Notice also that all fields are complete bar the field *capital-gain*, which has in excess of 92% missing values. The focus is to model the variable: *Response*. Its distribution is illustrated in figure 7.3

Figure 7.3: Response Distribution



It can be seen that 23.62% of records have a value of Yes, i.e.: have responded to a recent marketing campaign, and 76.38% have not.

7.8 Creating Partitions and Adding a Decision Tree Node

Prior to modelling it is recommended to partition the data. In general two partitions are created.

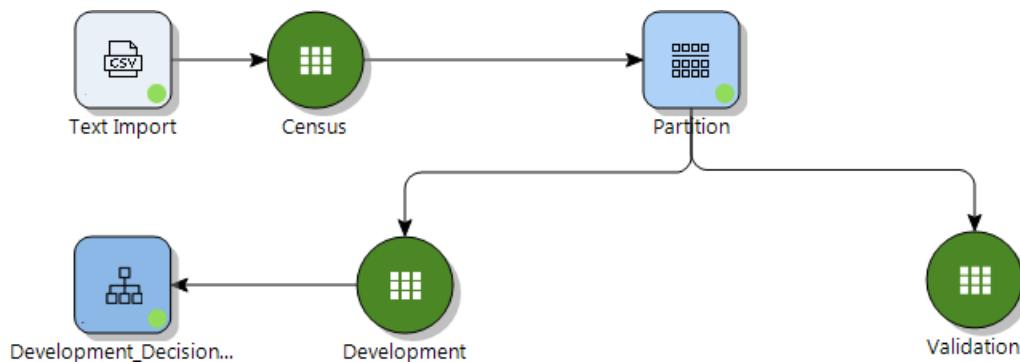
The model is developed and evaluated on one partition, usually referred to as the **Development** or **Training** partition, and validated on another, usually referred to as the **Validation** or **Testing** partition.

The **Testing** partition should reflect the population the model is ultimately applied to, and therefore acts as a proxy to determine model accuracy in the population.

To create the partitions drag the **Partition** node from the **Manipulate** palette to the canvas and link it to the **Census** dataset. Create two randomly sampled partitions called Development and Validation containing 70% and 30% of the parent dataset respectively.

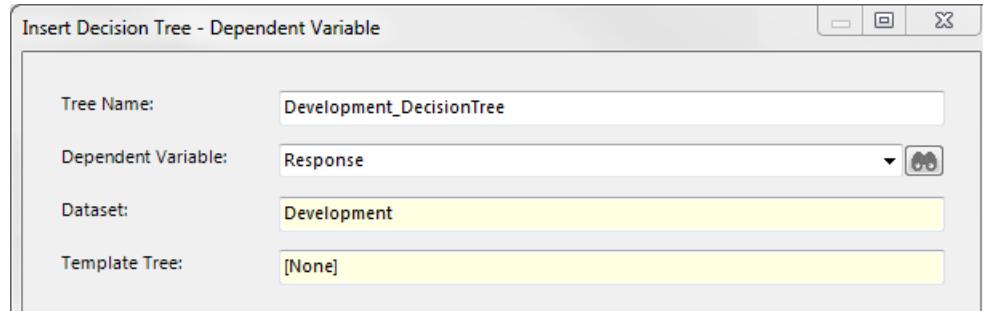
Once complete add and connect a **Decision Tree** node from the **Model** palette to the **Development** partition. The results should resemble figure 7.4.

Figure 7.4: Partitions and Decision Tree Added



Double click the **Decision Tree** or right click and select **Modify**, to access options.

Figure 7.5: Insert Decision Tree - Dependent Variable



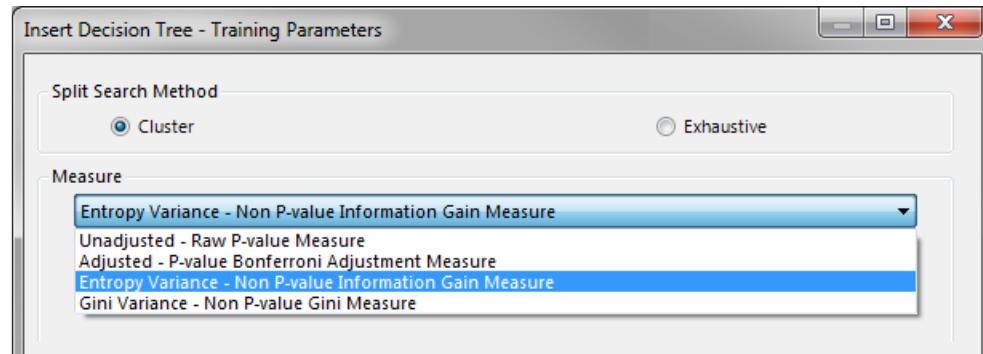
The initial dialog **Insert Decision Tree – Dependent Variable** illustrated in figure 7.5 provides options to name the resulting tree model and select the dependent variable.

The options **Dataset** and **Template Tree** are automatically populated with connected dataset names.

NOTE: Models can be generated as a **Model Instance** and used as a basis for further models. Connecting a **Model Instance** to the **Decision Tree** automatically populates as the **Template Tree**. Additionally if a template is being used, tree settings as per the template are applied and the **Dependent Variable** dropdown is not available.

For this demonstration, select *Response* as the **Dependent Variable**, all other options can be left as default. Clicking **Next >** opens the **Insert Decision Tree – Training Parameters** dialog.

Figure 7.6: Insert Decision Tree - Training Parameters



From here select the **Decision Tree** algorithm and the **Split Search Method**. The **Split Search Method** provides two options:

- **Cluster** Finds groups that maximize similarity within and dissimilarity between nodes. It is the default **Split Search Method**, tends to find more natural patterns than **Exhaustive**
- **Exhaustive** Finds groups that maximize statistical significance. Takes longer to train

The **Measure option** enables selection of the **Decision Tree** algorithm.

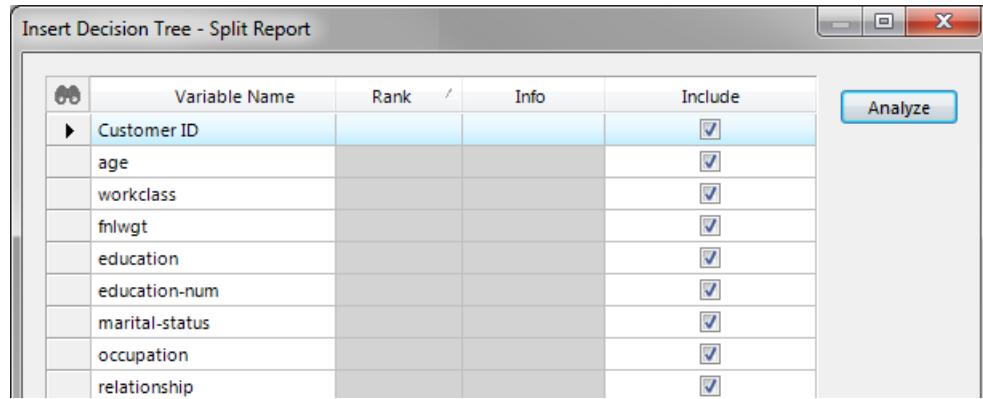
The default and recommended algorithm is **Entropy Variance - Non P-value Information Gain Measure**. This method generalizes well in most circumstances irrespective of data.

NOTE: The training parameters set in this wizard page govern the splitting mechanism of the **Find Split** and **Automatic Grow** operations.

Here the defaults of **Cluster for Split Search Method** and **Entropy Variance - Non P-value Information Gain Measure** for the **Decision Tree** algorithm will suffice.

Click **Next >** to proceed to the **Insert Decision Tree – Split Report** dialog

Figure 7.7: Insert Decision Tree - Split Report

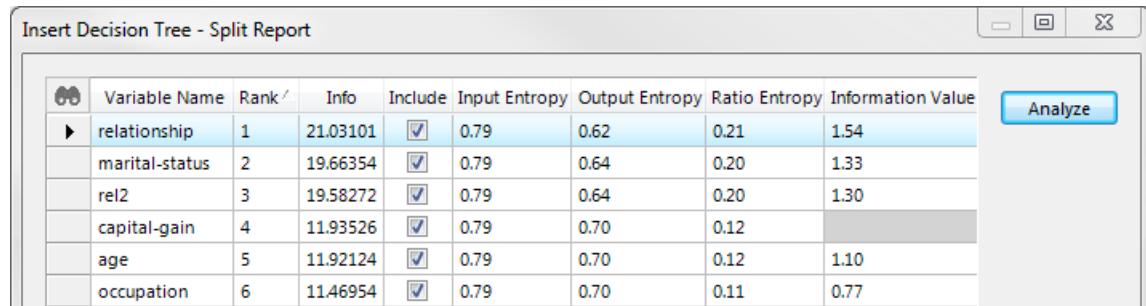


This dialog allows selection of independent variables. By default, all variables are included.

The **Analyze** button calculates and displays split report statistics. This can be a useful tool when deciding on variables to include in the model and are depicted in figure 7.8.

NOTE: Variable reduction is generally completed before this point.

Figure 7.8: Calculated Statistics



	Variable Name	Rank	Info	Include	Input Entropy	Output Entropy	Ratio Entropy	Information Value
►	relationship	1	21.03101	<input checked="" type="checkbox"/>	0.79	0.62	0.21	1.54
	marital-status	2	19.66354	<input checked="" type="checkbox"/>	0.79	0.64	0.20	1.33
	rel2	3	19.58272	<input checked="" type="checkbox"/>	0.79	0.64	0.20	1.30
	capital-gain	4	11.93526	<input checked="" type="checkbox"/>	0.79	0.70	0.12	
	age	5	11.92124	<input checked="" type="checkbox"/>	0.79	0.70	0.12	1.10
	occupation	6	11.46954	<input checked="" type="checkbox"/>	0.79	0.70	0.11	0.77

The report calculates statistics related to the algorithm chosen and ranks the results by variable

importance, the most important, or best predictor is given a rank of 1.

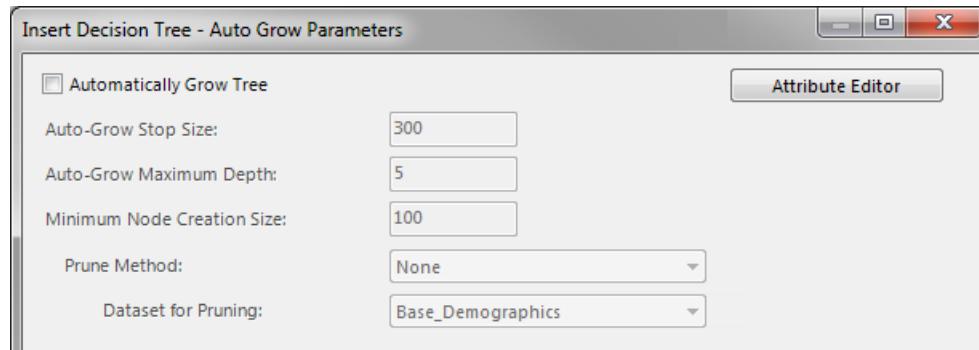
Visible statistics are listed and described in table 7.3.

Table 7.3: Calculated Statistics

Statistic	Description
Rank	Ranks variables. A value of 1 indicates the most significantly/importantly related to the dependent variable
Info	100 x ratio entropy
Input Entropy	E_i achieves a maximum of 1 for a uniform distribution and a minimum of 0 for a degenerate/peaked distribution
Output Entropy	See Decision Tree in Mathematical Formulations in Help files for detailed equations
Ratio Entropy	1 – Output Entropy / Input Entropy

Click **Next >** to open the **Insert Decision Tree – Auto Grow Parameters** dialog.

Figure 7.9: Insert Decision Tree - Auto Grow Parameters



This dialog becomes active by checking the **Automatically Grow Tree** check box.

If **Automatically Grow Tree** is not selected, clicking finish creates the tree as a root node, and the model must be grown interactively. If **Automatically Grow Tree** is selected, the tree is grown as per options set, detailed in table 7.4.

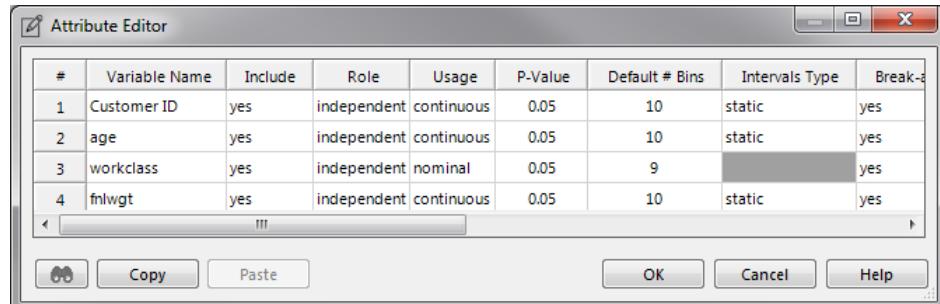
Table 7.4: Insert Decision Tree – Auto Grow Parameters

Option	Description
Auto-Grow Stop Size:	Minimum number of records in a node to attempt to grow that node
Auto-Grow Maximum Depth:	Maximum number of levels in the tree, including the root node. Therefore a value of 1 will result in no tree growth. A value of 0 means no maximum
Minimum Node Creation Size:	Minimum number of records required for a child node
Prune Method:	Requires at least one other dataset One dataset is used to grow the tree and the other is utilized by the prune method

This dialog also includes an **Attribute Editor** button. The **Attribute Editor** allows modification of individual variable attributes prior to tree modeling.

Attribute values can be set and modified when inserting the tree from the tree wizard or modified once the tree has been created by choosing **Attribute Editor...** from the **Tools** menu.

Figure 7.10: Attribute Editor



NOTE: Some analysts prefer to build **Decision Trees** using only binary splits. Binary splits have the advantage that they are easier to read than multiple splits.

In addition, binary splits avoid sub-segmenting too much at early stages of a tree build. To use binary splits, use the **Attribute Editor** to set the **Max Branches = 2** for all variables.

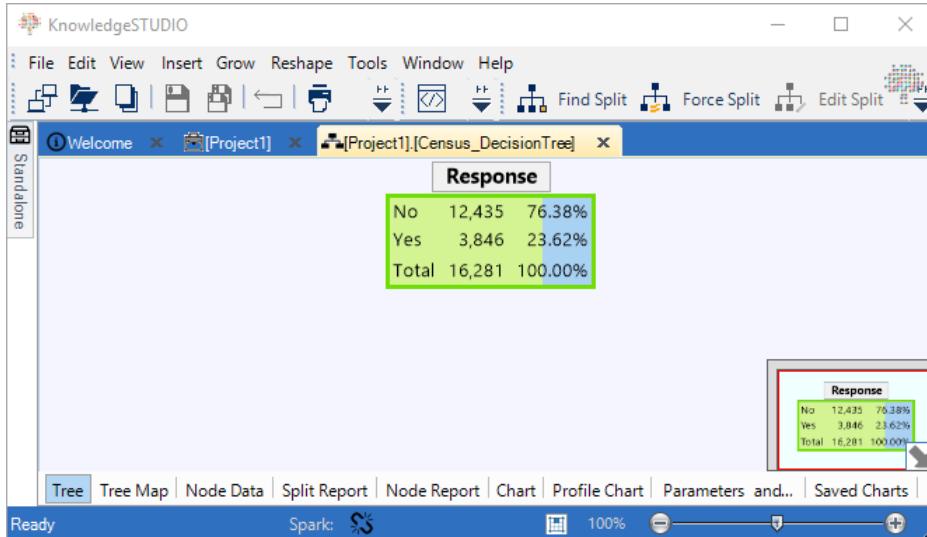
Attribute values for multiple variables can be changed simultaneously, simply *Ctrl-Select* or click and drag to highlight multiple variables. Changing any column value is applied to all selected variables.

For the purposes of this demonstration, deselect **Automatically Grow Tree** and click **Run**.

A **Decision Tree** object has now been created in the **Project Pane**. This is symbolically referenced as a node on the **Workflow** canvas with a green check mark (not shown).

Opening the **Decision Tree** node reveals the model as illustrated in figure 7.11.

Figure 7.11: Initial Results

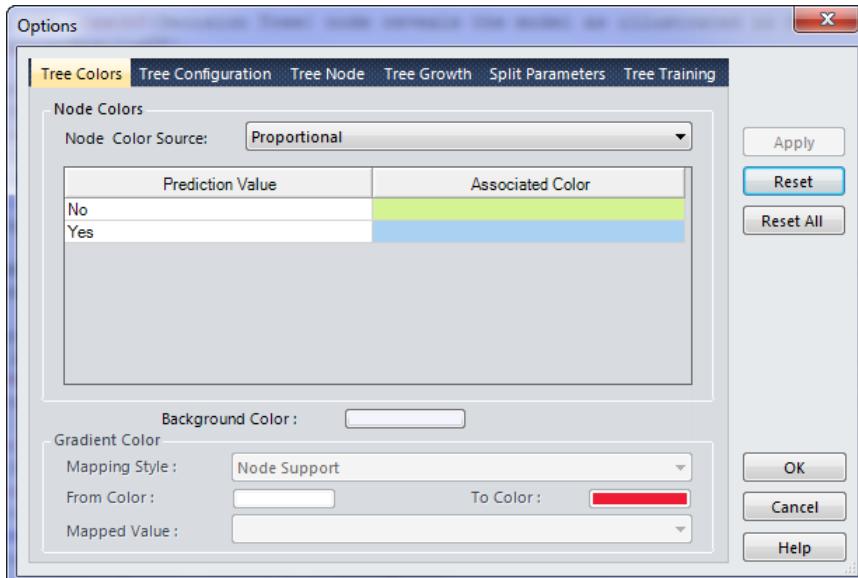


Nine tabs to aid in model exploration are visible on opening the **Decision Tree**. The tree opens on the **Tree** tab, and the root node appears.

The root node displays colour-coded distributions of the categories in the dependent variable for the entire dataset. In this example, green represents the proportion of No: 76.22%, and blue corresponds to the proportion of Yes: 23.8%.

NOTE: If **Automatically Grow Tree** was selected a fully grown tree would appear. The **Options** button provides access to tree and display parameters. Additional access via right-clicking any white space and selecting **Options** from the visible **Menu**.

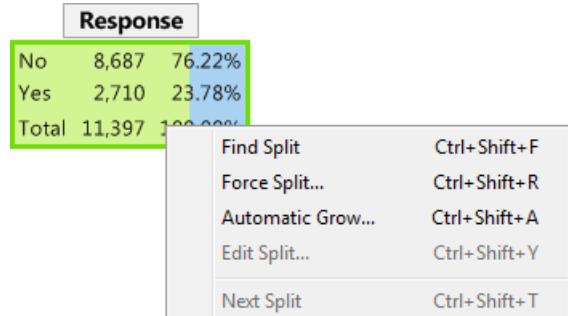
Figure 7.12: Decision Tree Options



7.9 Growing the Decision Tree

Tree growing options are available from the **Task Bar** or by right clicking the root node.

Figure 7.13: Tree Growth Options



Some available options are described in table 7.5.

Table 7.5: Tree Growth Options

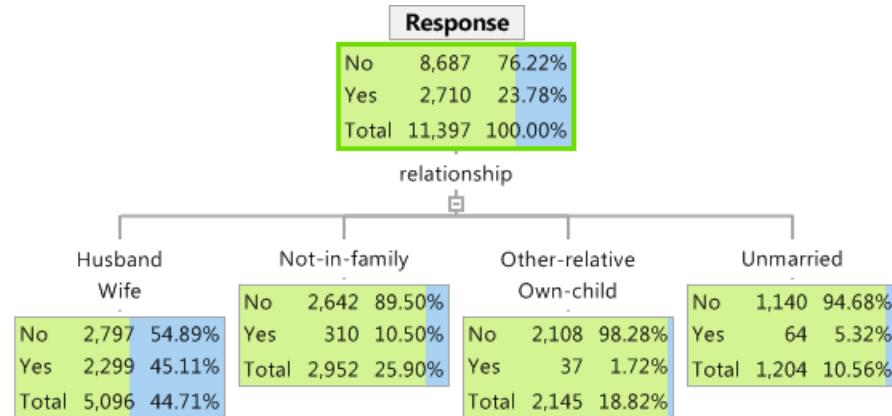
Option	Description
Find split	Algorithm searches for the best predictor given the selected node based on ranked information value. The predictor is displayed as a set of child nodes and automatically binned to optimize results. Other splits can be viewed by right clicking and selecting Next Split , Previous Split or Go to Split . The Split Report tab is populated once a split has been found and can be referenced to guide predictor selection
Force Split...	User specified predictor and split selection. Optimal binning is optional
Automatic grow...	Tree grows automatically as per predefined algorithm parameters. Results are identical to choosing Automatically Grow Tree from the Insert Decision Tree – Auto Grow Parameters dialog
Edit Split...	Available only for existing splits. Allows further binning or splitting of resulting child nodes

The following illustrations show the results of selecting each of these options in turn. A combination of these methods is often used when building models.

7.9.1 Find Split

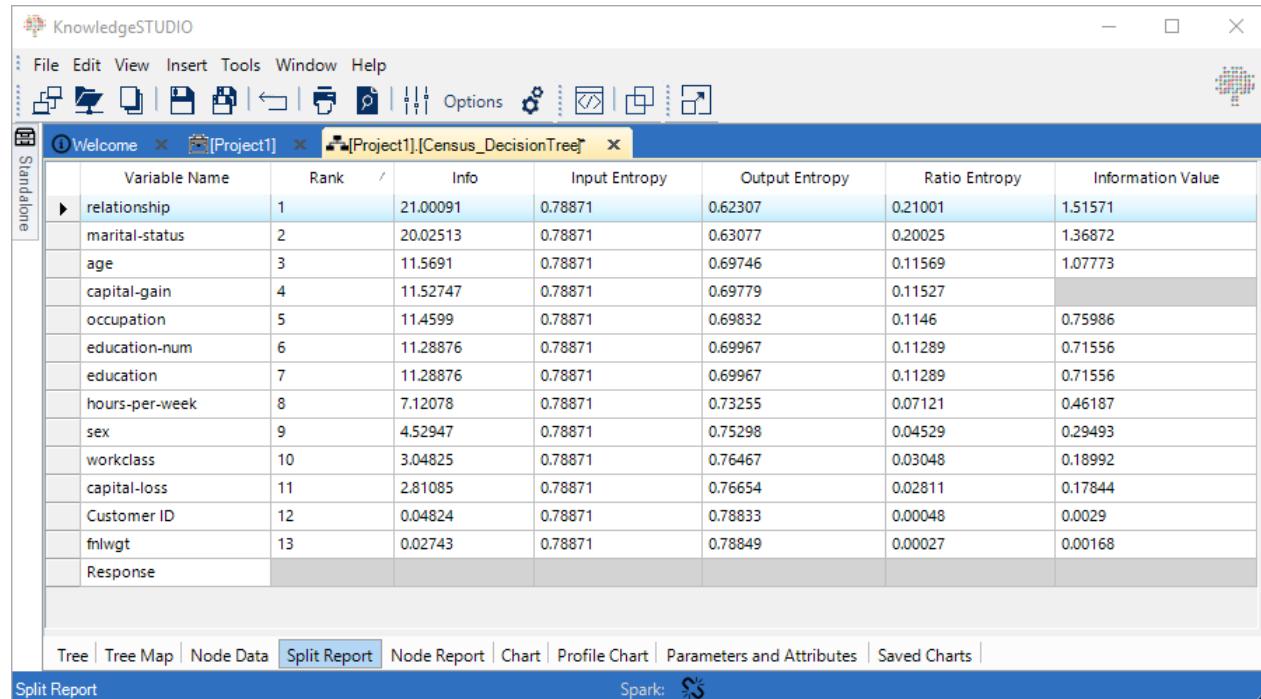
Using **Find Split** selects the variable *relationship* and bins categories as illustrated in figure

Figure 7.14: Find Split



Determining why *relationship* was selected can be understood by viewing the **Split Report** tab.

Figure 7.15: Split Report Window

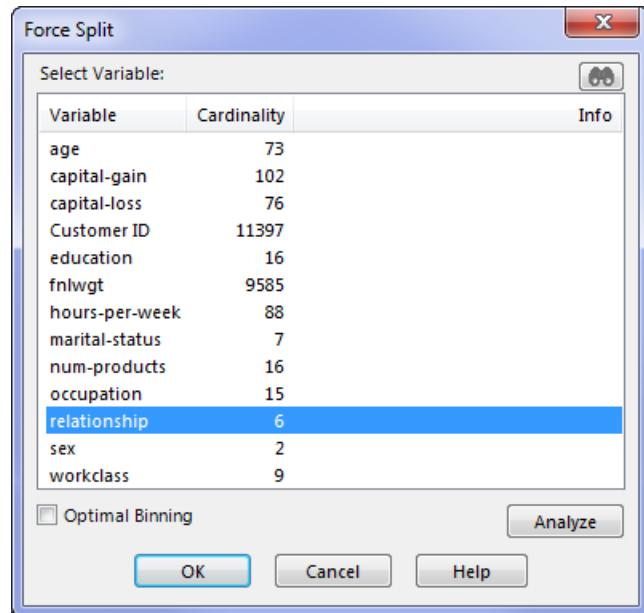


The **Split Report** provides a list of variables ranked by predictive power for any selected node. Notice that *relationship* has a rank of 1, hence it being selected as the split variable.

7.9.2 Force Split

Right clicking the root node and choosing **Force Split...** opens the **Force Split** dialog.

Figure 7.16: Force Split



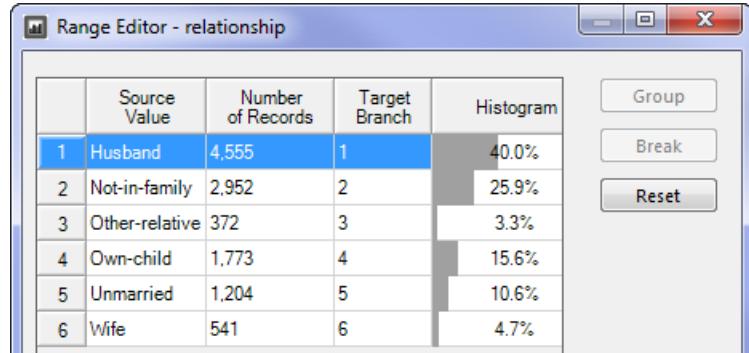
Force Split allows complete control of the model structure. Any variable can be chosen and forced into the model, irrespective of its importance or statistical significance in relation to the dependent variable.

Clicking **Analyze** shows the **Information Value (Info)** for each variable. Checking the box for **Optimal Binning** allows the program to try to reduce the number of bins for the selected variable by combining bins without statistically significant differences in the distribution of the dependent variable.

For more information about how this works, see the **Optimal Binning Helper** in the **Variable Transformations** wizard.

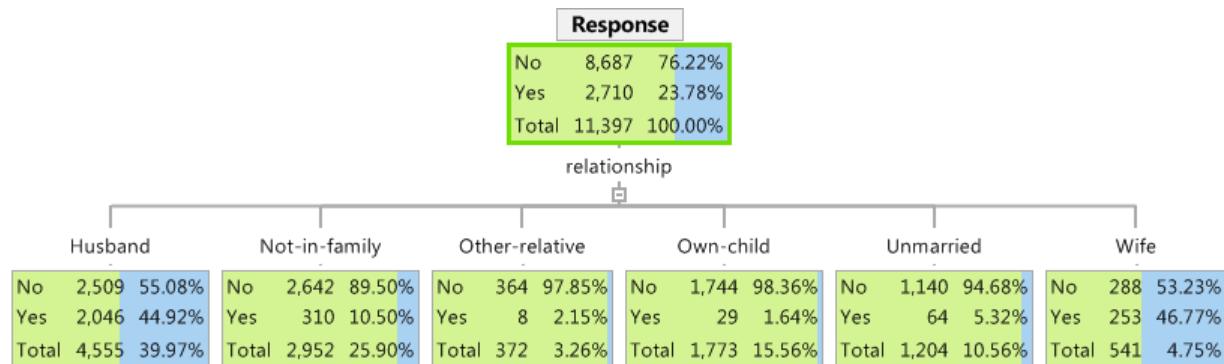
Once a variable is selected, the **Range Editor** opens allowing control over category grouping.

Figure 7.17: Force Split Range Editor



If no grouping is enforced the variable is added to the tree with no binning applied.

Figure 7.18: Forced Split

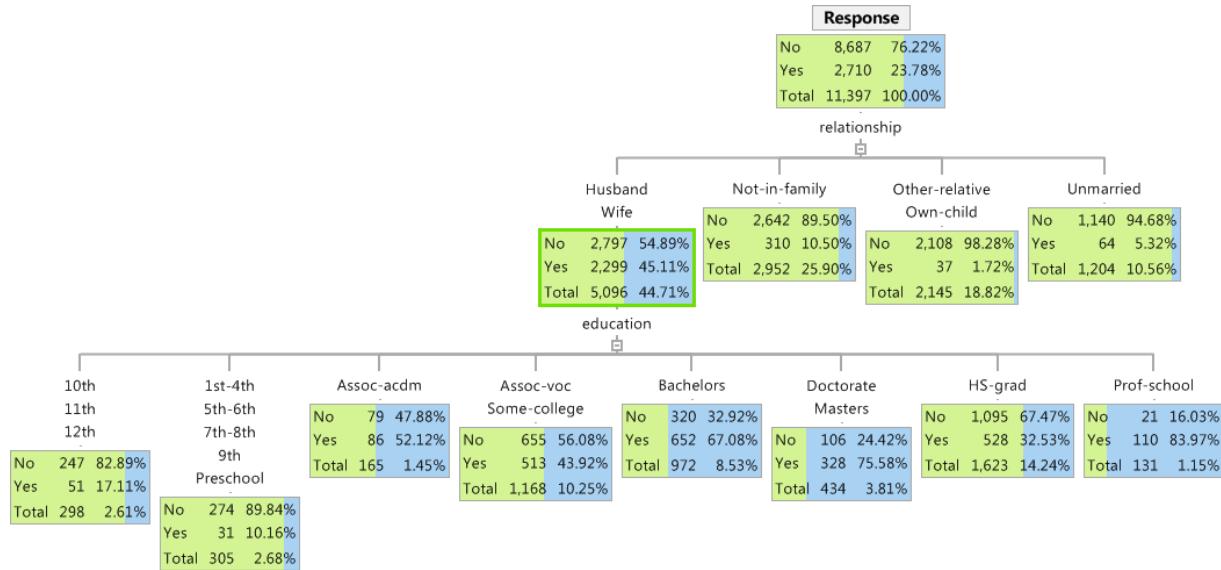


7.9.3 Edit Split

Binning can be modified whether a field is selected automatically or forced. Right click any parent node and select **Edit Split**. This opens the **Range Editor**, as illustrated previously, providing access to current variable binning with options to modify.

To illustrate, build a tree as depicted in figure 7.19 using only **Find Split..**

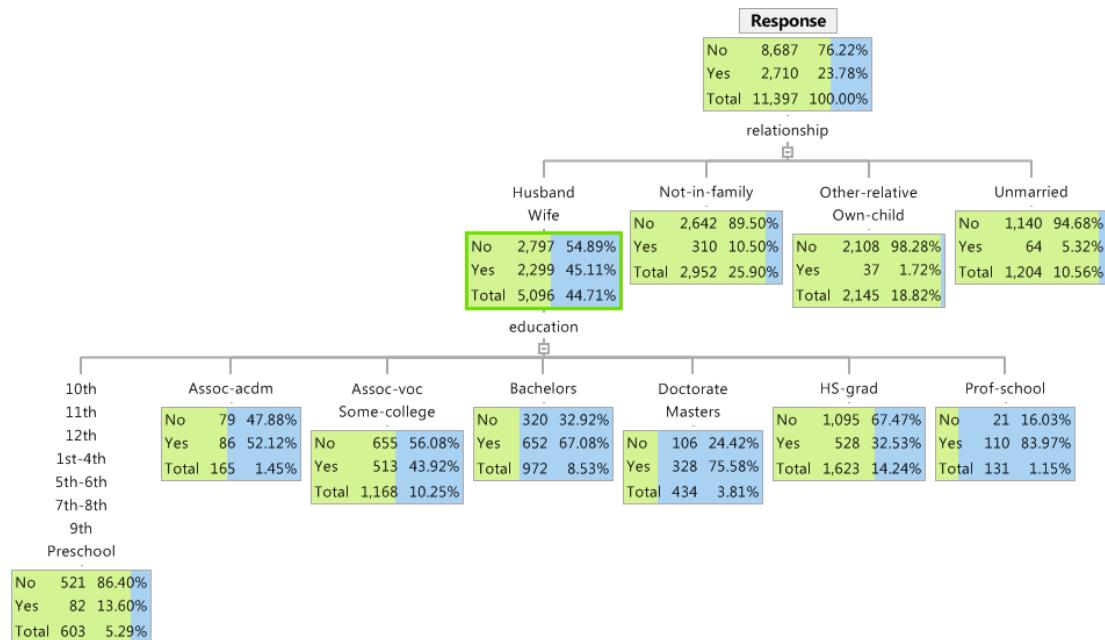
Figure 7.19: Additional Growth



It may be that the business does not consider those with lower education levels any differently, i.e. from *pre-school* to *12th*.

To combine these categories, right click and select the option **Edit Split**. *Ctrl-select* categories to merge and click **Group**. Clicking **OK** applies the modifications as illustrated in 7.20.

Figure 7.20: Edited Split



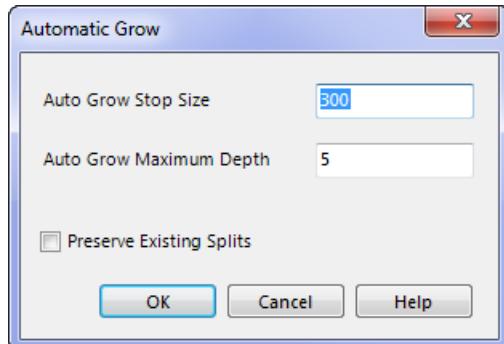
The growing process can continue using the methods as previously discussed.

7.9.4 Automatic Grow

The **Automatic Grow** option grows the tree according to pre-defined build parameters.

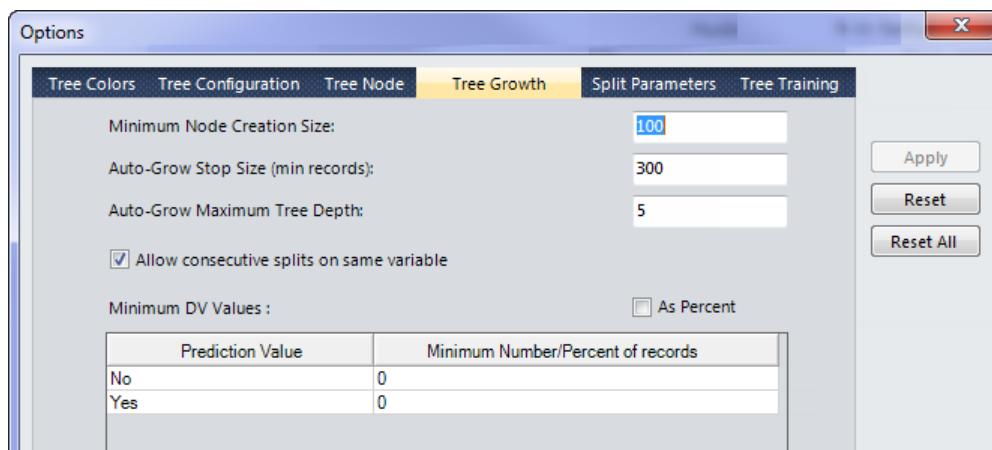
Right click any node and select **Automatic Grow** to open the **Automatic Grow** dialog.

Figure 7.21: Auto Grow



Automatic Grow options are limited. More extensive options are available from the **Tree Growth** tab accessed from the **Options** button on the **Task bar**.

Figure 7.22: Tree Growth Tab



Available options are described in table 7.6.

Table 7.6: Complete Tree Growth Options

Option	Description
Minimum Node Creation Size:	The minimum number of records that a node must contain before it is displayed as a separate node. A good model will contain between 1% - 2% of records of the training dataset.
Auto-Grow Stop Size (min records):	The minimum number of records in a node required before Automatic Grow will attempt to split the node. A good model will have between 3% - 5% of the training dataset
Auto-Grow Maximum Tree Depth:	Max number of levels in the tree. This number includes the root node. A value of one will not grow a tree beyond the root node A value of zero stands for "unlimited"
Minimum DV Values:	Limits for minimum number or percentage of records in each Dependent Variable category across each node

NOTE: Most options are available when using the wizard to initially build the tree on the **Insert Decision Tree – Auto Grow Parameters** dialog.

Selecting **Automatic Grow** may result in a large tree. **KnowlegeSEEKER** provides an easy to use navigation pane in the bottom right hand corner to better assess, understand and explore the model(not shown).

7.10 Automated Versus Manual

As highlighted in the previous sections, **Decision Trees** are easily created and manipulated using the various build methods.

Using algorithms alone to build models maximizes predictive power, however interactively growing a tree allows business strategies to be incorporated.

NOTE: It is highly recommended that manual build methods are applied in a judicious manner.

7.11 Tree Object Tabs

Nine tabs aid in exploring the tree. Each contains unique information or profiles of the **Decision Tree**.

Table 7.7: Tree Object Tabs

Statistic	Description
Tree	Displays the tree
Tree Map	Can help navigate larger trees and identify nodes of interest
Node Data	Displays records associated with any selected node in the tree. Data can be exported using the Export option from the File menu
Split Report	For each Split, the Split Report provides a statistical report of predictors evaluated for that split
Node Report	Terminal nodes are ranked in descending order of the target category
Chart	Dynamically graphs variable distributions for any node. Select any node and the chart tab updates to illustrate that variable
Profile Chart	Graphic representation of terminal nodes. Each node is represented as a bar. Bar width represents the no. of cases in a node relative to other nodes. Height represents target category probability
Parameters and Attributes	Predictor variable parameters and attributes
Saved Charts	Any charts of interest can be saved to this tab

7.12 Improving the Model

In order to improve a model it is wise to understand what makes a good model. A good model should possess following characteristics:

- **Robust** A good model should generalize well and not overfit the training data, unless there is interest only in classifying cases in the current file. To ensure a robust model, the tree should be pruned to avoid small nodes sizes and large branches
- **Accurate** Purer terminal nodes are more desirable. Ideal terminal nodes are predominantly concentrated in one category. Observe terminal nodes and compare the % of the dependent variable target category to the root node. Each split should further homogenize the dependent variable target category
- **Simple** A good decision tree should be parsimonious and easy to use. The number of levels should not exceed 5 and the total number of terminal nodes should be less than or equal to 50. This aspect can be easily governed through observation if manually building a model, and setting options if using automatic methods
- **Explainable** The model should have characteristics that make sense

This section will focus mainly on building a robust model. Other aspects are introduced in this section and developed in following chapters.

7.12.1 Robust: Nodes Size

In order to build a robust model that performs well on new data, the tree should be pruned to avoid small nodes and large branches.

The terminal nodes should contain an adequate number of records to avoid over-fitting. It is recommended to have a minimum number of records equal to at least 1% to 2% of the training dataset in each node.

NOTE: The following illustrations are for informative purposes and may not reproduce with current data.

Figure 7.23: Adequate Node Size



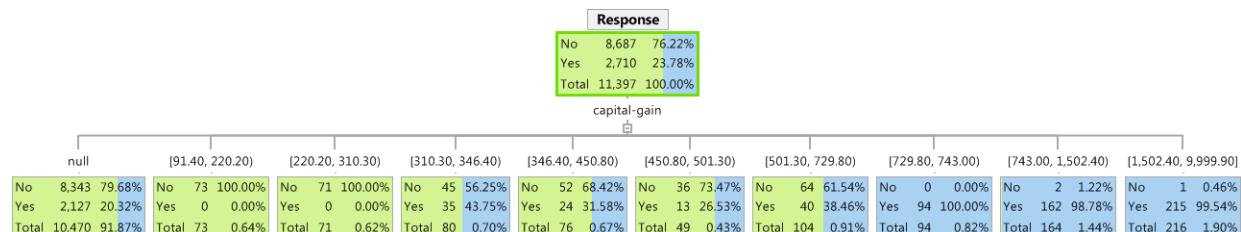
The right-most node of the tree on the left hand side in the above image is extremely small, keeping it in the model is probably not wise: it will likely lead to overfitting. To prevent this, merge or remove the node, here the node is merged to another node using the **Edit Split** option.

7.12.2 Robust: Number of Branches

As with very small nodes, very wide trees will lead to over-fitting. In general, a robust model should have six or fewer splits regardless of whether the tree is grown manually or interactively.

To illustrate, force a split on the root node using the variable *capital-gain* and accept the default of 10 bins.

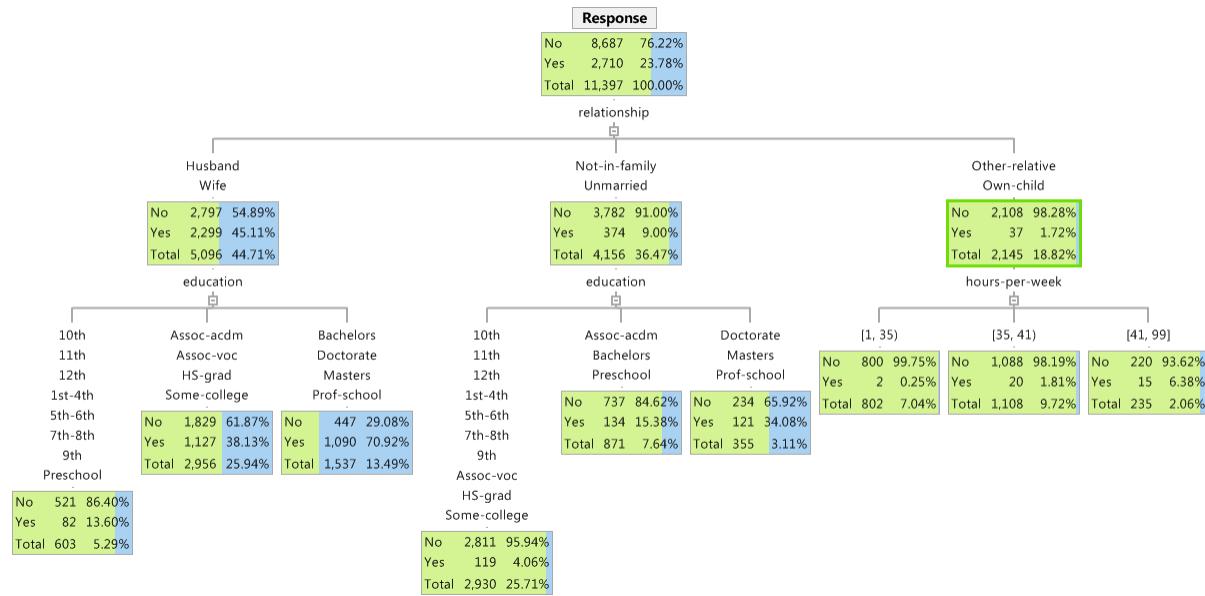
Figure 7.24: Forced Split with 10 Bins



This of course is also quite complex and may not be very reflective of the population. In cases like this use the **Range Editor** to reduce splits to a more acceptable number of nodes: somewhere between 4 - 6.

If using **Automatic Grow** or **Find Split** the **Max Branches** can be set from the **Attribute Editor**. This will ensure that the number of splits is within reasonable and acceptable bounds. Figure 7.25 show results using **Find Split with Max Branches** set to three.

Figure 7.25: Find Split with Max Branches = 3



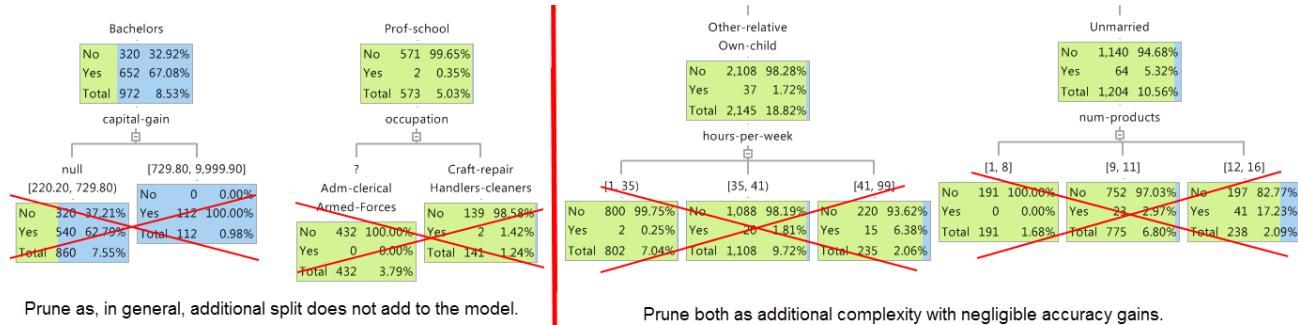
7.12.3 Model Accuracy

An accurate model should have nodes that concentrate cases into one of the dependent variable categories. In general nodes that have 100% of cases in one category should be treated with suspicion as this seldom happens in real life.

In practice, nodes with a high concentration into one of the dependent variable categories are perfectly acceptable. Additionally, splits that do not better explain the dependent variable should also be sought out and pruned.

NOTE: The following illustrations are for informative purposes and may not reproduce with current data.

Figure 7.26: Pruning Nodes



7.12.4 A Simple Model

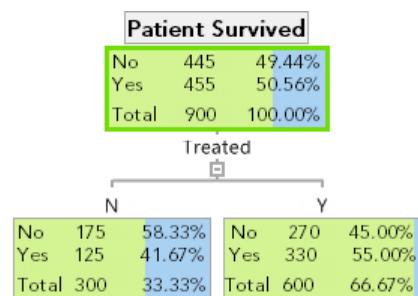
A good **Decision Tree** should always be easy to understand and explainable. Splits can be assessed during interactive growth or empirically if grown automatically. In general, although not set in stone, trees should be grown to a maximum of 5 levels.

7.12.5 Explainable Models and Simpson's Paradox

An explainable model is a model that has rules that make sense. This can sometimes be a bit tricky because an inconsistency may be detected called **Simpson's Paradox**.

Simpson's Paradox occurs when a correlation present in all groups of data is reversed when the groups are combined. Figure 7.27 illustrates treatment results for patients affected by the **Bird Flu Virus**.

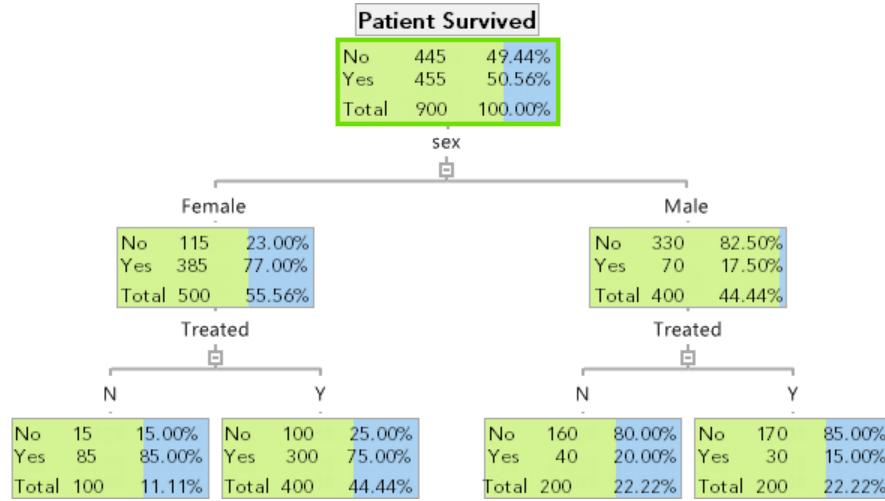
Figure 7.27: A Simple Tree Yesterday



From the tree; the treated group had increased chances of survival over untreated groups by about 15%.

However, introducing another variable can affect the conclusion. In general, men and women react differently to medicines and viruses, and including *Gender* produces contradictory results.

Figure 7.28: Simpsons Paradox



Focussing on the first split shows that women are much more likely to survive than men. However when each group is split further by the variable it can be seen that a higher proportion of females survived when not treated: 85%, in comparison to 75% survival when treated.

This is also evident for males! A higher % of males survive when not treated: 20% vs 15% when treated.

This contradicts the initial supposition that treatment increases chances of survival. This is **Simpson's Paradox**; a relationship that seemed evident from the data completely reversed when the data was split by another variable. Variables that display **Simpson's Paradox** are said to be part of **Simpson's Triangle**.

It is estimated that 40% of business data has this problem. Where possible, trees should exclude variables that are part of **Simpson's Triangle**. At a minimum, variables that are part of **Simpson's Triangle** should be examined further for more thorough understanding.

NOTE: **Simpson's Paradox** illustrates something that seems illogical. It is easy to miss cases of **Simpson's Paradox** because they seem to contradict conclusions that appear obvious. When building models, it is important to be vigilant and make sure to avoid making false conclusions about such variables.

Simpson's Paradox is one pitfall to be aware of when developing **Decision Tree** model. There may be other aspects of data that are subtle and hard to spot. Question all splits to ensure they make sense and are appropriate.

7.12.6 Version Control; Creating a Model Instance

Once a final tree has been arrived at, a **Model Instance** can be created. This enables the specifications associated with a particular model to be retained.

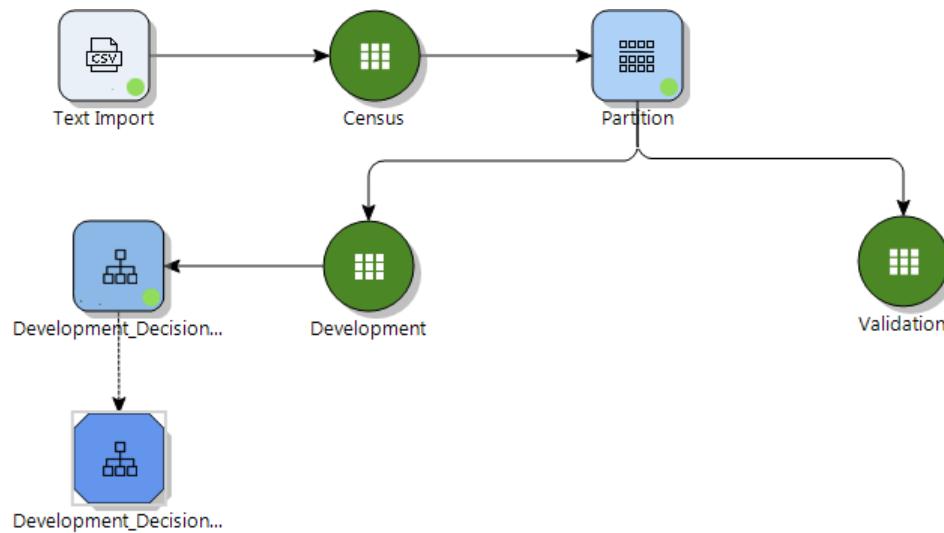
The same generating node can be used to create further instances with different parameters settings. To create a model instance right click the generating node, and select the option **Save Model Instance**, (not shown).

This opens the **Save Model Instance** dialog, where the default name can be modified, (note shown).

If the default name is not changed, a model instance is created using the following syntax:

Here the model instance is called **Development_DecisionTree_06Dec**

Figure 7.29: Model Instance



The model instance is created as an object in the **Project Pane** and its corresponding canvas node is connected to the generating node by way of a dotted line.

If model parameters are changed and the model re-run, another instance can be generated. Any model instance after the first is assigned a default name using the syntax:

`<partition_name>_<Decision_Tree>_<Date><n>` ...where n begins at 1

7.13 Conclusion

KnowledgeSTUDIO Decision Trees can be used to explore unfamiliar datasets, build models and illustrate relationships. **Decision Trees** as a modelling technique can accommodate categorical or continuous dependent variables. Similarly, inputs (predictors or independent variables) can also be of any type. As a result of completing this chapter users should be able to:

- Build **Decision Trees** with **Altair KnowledgeSTUDIO**
- Understand and employ the various growing methods to iteratively, interactively and automatically build **Decision Trees**
- Prune tree nodes to increase acceptability
- Assess variable importance
- Create **Model Instances** to store successive models

Exercises

For these exercises, use the Census data from one of the different files provided (e.g. Census.csv).

1. If partitions do not already exist, create two random partitions using the **Partition** node. Create a **Development** partition, and allocate 10,000 records to it. Create a **Validation** partition, and allocate 6,000 records to it
2. Create a **Decision Tree** by dragging a **Decision Tree** node onto the **Workflow** canvas
3. Use the **Development** partition to train the model by linking it to the **Decision Tree** node
 - (a) Select the variable *Response*, as the **Dependent Variable**
 - (b) Name the model appropriately
 - (c) Use the default algorithm: **Entropy Variance – Non P-value Information Gain Measure** to develop the model
4. Open the **Decision Tree**. The first tab displays the tree view
5. Assess the dependent variable distribution
6. Use the taskbar icon or right click and choose Find Split to find the first split variable.
7. What variable was chosen? Why? Refer to the Split Report tab to assess.
8. Use Find Split on the resulting nodes to further build the tree.
9. If any splits result in more than six child nodes use Edit Split to merge nodes together, use your judgement to assess which should be merged.
10. Explore the other tabs to assess the information they contain.
11. Split the screen to have the model, some charts and the Tree Map visible simultaneously.
12. Create a dynamic chart of a single variable, e.g. age. Click on any node to see the charts reflect the distribution of cases in the selected node.
13. Return to single view.
14. Use the **Automatic Grow** option to allow the algorithm to automatically grow the tree.
15. Assess variable attributes using the **Attribute Editor** from the **Tools** menu to assess whether any variables should be excluded.
16. Should all variables be used in building the model? Consider excluding the variable *fnlwgt*. Why?

17. Use the **Force Split** option and choose any variable to force a split. How does this differ from the other methods?
18. View the node report for a tabular representation of the tree.
19. View the **Parameters and Attributes** tab to assess current variable attributes.
20. Explore creating more trees using a different algorithm, by modifying the **Decision Tree** generating node.
21. Create a model instance.

Chapter 8: Ensemble Models

8.1 Introduction

Ensemble Models are a class of models that involve building a large number of simple models and combining their results to make predictions.

KnowledgeSTUDIO implements three different types of **Ensemble Model** to allow users multiple options when modelling. The three available model types are:

- **Boosting**
- **Bagging**
- **Random Forests**

When building such models, it is typical to have anywhere from hundreds to tens of thousands of decision trees involved, so the simple decision trees in these models are not built interactively, but instead take advantage of the auto-growth functionality for decision trees in **KnowledgeSTUDIO**.

On completion of this chapter users should be able to:

- Understand and use the ensemble models available in **KnowledgeSTUDIO**.

8.2 Boosting

In general, **Boosting** models begin with a single model and iteratively create new models based on the previous model, assigning a greater weight to those records incorrectly predicted in the previous iteration.

In practice, after each tree in the **Boosting** model is trained, the error from the previous model is computed. A weight vector is assigned which depends on how well the previous model performed; where the previous model made a correct prediction the weight for the next model is decreased, and where the previous model made an error the weight for the next model is increased.

New trees are trained on the weighted data, meaning that each tree will be more likely to do a good job on the subsets of the data where the previous tree did poorly. However, each tree in the process is more specialized and should therefore have less weight than previous trees when the predictions are all averaged.

This is handled by a shrinkage factor, which is user-editable in **KnowledgeSTUDIO** – larger values of the shrinkage factor result in more aggressive model. The process that **KnowledgeSTUDIO** uses to create a Boosting model works as follows:

1. Grow an initial decision tree. Create an initial weight vector; all records get equal weight
2. Use the tree to score the training data, and detect which records are misclassified
3. Update the weight vector: more weight to incorrectly predicted records in the previous model, and reduce the total weight with the shrinkage factor
4. Grow a new decision tree on the training data, but taking the weighting into account
5. Repeat 2-4

6. Combine results

The end result is a single **Boosting** model with the ability to make predictions on new data. New data is scored by averaging the predictions across all trees.

8.3 Bagging

Bagging is short hand for **Bootstrap Aggregation**. Of the 3 types of **Ensemble Models** implemented in **KnowledgeSTUDIO**, **Bagging** models are the simplest. In **Bagging** models, a number of simple decision trees are grown using automatic growth features.

The simple decision trees are not all identical because the data that are used to train the trees are not the same for each tree. Say that there are N records in the training data to build the **Bagging** model. For each decision tree, N records are chosen with replacement. That means that some records may appear more than once, and some records won't appear at all. This technique is known as **Bootstrapping**.

NOTE: The **Bagging** wizard allows the user to modify the sample size and to turn off **Sampling with Replacement**. Sampling with replacement with a sample size equal to the total number of records is the default. Result of this process is that a large number of trees are built based on bootstrapped samples of the training data. Since these trees are trained on different datasets, they likely won't be identical. The predictions of these trees are averaged to create the final **Bagging** model.

8.4 Random Forest

Random Forest models are similar to **Bagging** models, but slightly more complicated.

Bagging involves the creation of a large number of decision trees using bootstrapped data and variables are the same for all trees. In **Random Forest** models, the variables included in the model are also sampled. Each simple decision tree will be trained with only a subset of all the variables available to that specific tree.

While this sampling of variables makes the individual trees in a **Random Forest** model simpler, and likely weaker on average, it can also mean that there is more variation between the trees in the model, and in many cases this can lead to better predictions on new data used when the model is actually being deployed.

NOTE: Similar to the **Bagging** wizard, the **Random Forest** wizard allows the user to modify the sample size and to turn off **Sampling with Replacement**.

In addition, the user can specify the proportion of variables to use for each sample.

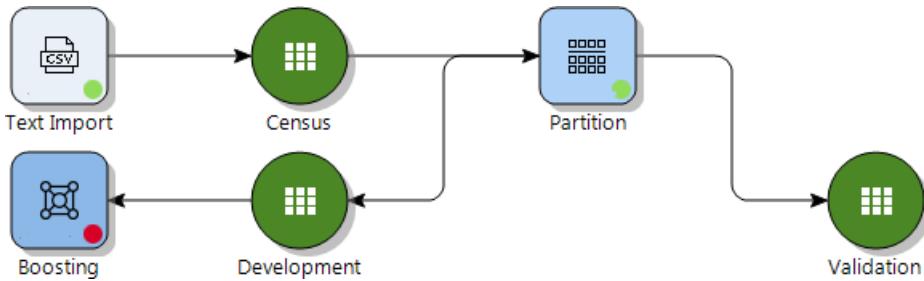
8.5 Demonstration

The following examples illustrate the creation of one of each type of ensemble model; **Boosting**, **Bagging** and **Random Forests** using the file: *Census.txt*.. Two partitions are created called **Development** and **Validation**, the proportion of the dataset assigned to each is 70% and 30% respectively.

8.5.1 Boosting

Drag a **Boosting** node from the **Model** palette onto the canvas, and connect it to the **Development** partition as illustrated in figure 8.1.

Figure 8.1: Boosting Model

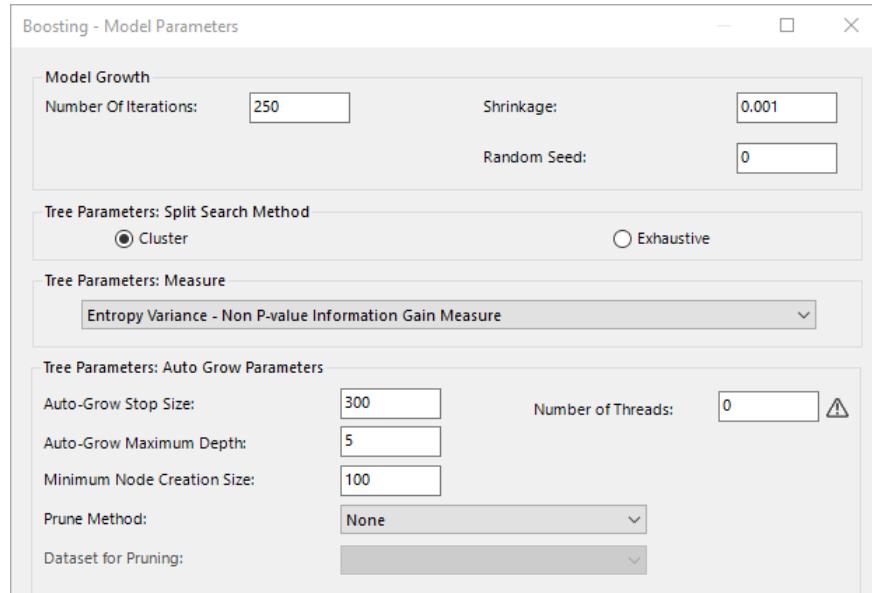


Access the **Boosting** node to set options. The first dialog provide the facility to assign a name to the model, the connected dataset and whether any template model is used are also evident, not shown.

The second page of the wizard, **Boosting – Variable and Model Selection**, allows selection of the dependent and independent variables and the dependent variable target category.

The third and final page is **Boosting - Model Parameters**. This include parameters that govern model build.

Figure 8.2: Boosting - Model Parameters



Bearing in mind that ensembling methods are applied using a **Decision Tree** algorithm, so, options are identical to those presented when growing a **Decision Tree**. Additional aspects specifically related to **Boosting** are:

- **Number of Iterations** Number of models to grow
- **Shrinkage** How much the weights decrease at each iteration

Here, options are set as follows:

- The **Dependent variable** is *Response*, with the **Target Category** selected as **Yes**
- All fields bar *Customer ID* and *fnlwgt* are included
- The **Number of Iterations** is changed from 250 to 100

NOTE: Depending on the number of iterations, model building may take time. The smaller the number of iterations, the faster the process, but, potentially, the weaker the model.

Once options have been specified, click **Run** to build the model and once complete, double click the node to access results. Model results are minimal and output is available from two tabs:

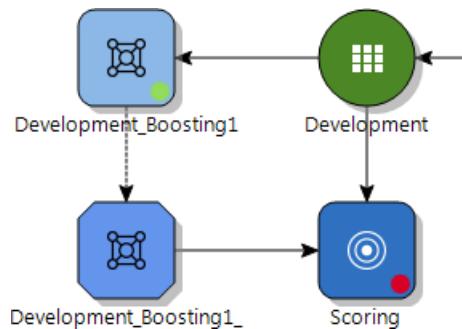
- **Results**
- **Parameters and Attributes**

Both sets of output results are simply reflecting the model set up prior to running. As a direct result of building many models and combining results, no additional output is created.

In fact the output from ensemble models is minimal to such an extent that scoring and model evaluation are the only recourse to assess model performance. Scoring the model is accomplished by adding a **Scoring** node as illustrated in figure 8.3. Remember to create a model instance first!

NOTE: The same scored results are achieved using a **Model Validation** node to score the data.

Figure 8.3: Scoring The Model



Opening the **Scoring** node and navigating to the **Scoring – Scoring Fields** page reveals that only four new fields are added to the results:

- **Response Prediction**
- **Response Probability of Prediction**

- Response No Probability
- Response Yes Probability

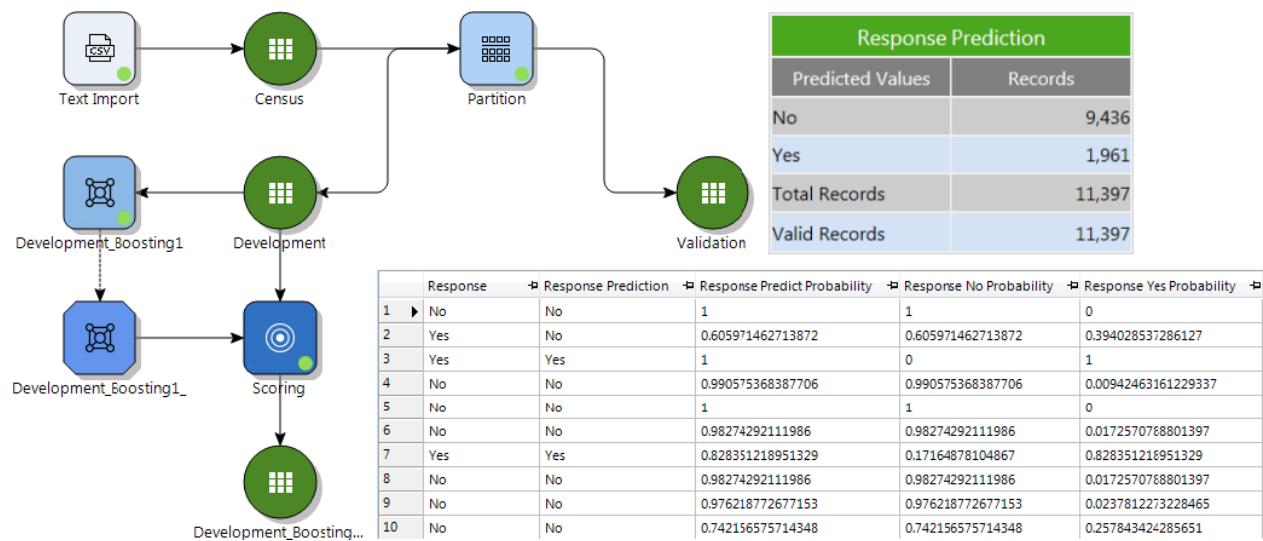
Figure 8.4: Scoring Fields

Item	Field Name	Include	Cut Off
Response Prediction	Response Prediction	<input checked="" type="checkbox"/>	
Response Probability of Prediction	Response Predict Prob	<input checked="" type="checkbox"/>	
Response No Probability	Response No Prob	<input checked="" type="checkbox"/>	0.5
Response Yes Probability	Response Yes Prob	<input checked="" type="checkbox"/>	0.5

These fields are the only outcomes generated for ensemble models. These fields can be used to further evaluate and assess the resulting model.

Once scored, the **Report** tab provides minimal insight into model performance by relaying the number of records assigned Yes or No, the dependent variable outcomes.

Figure 8.5: Boosted Model Scored



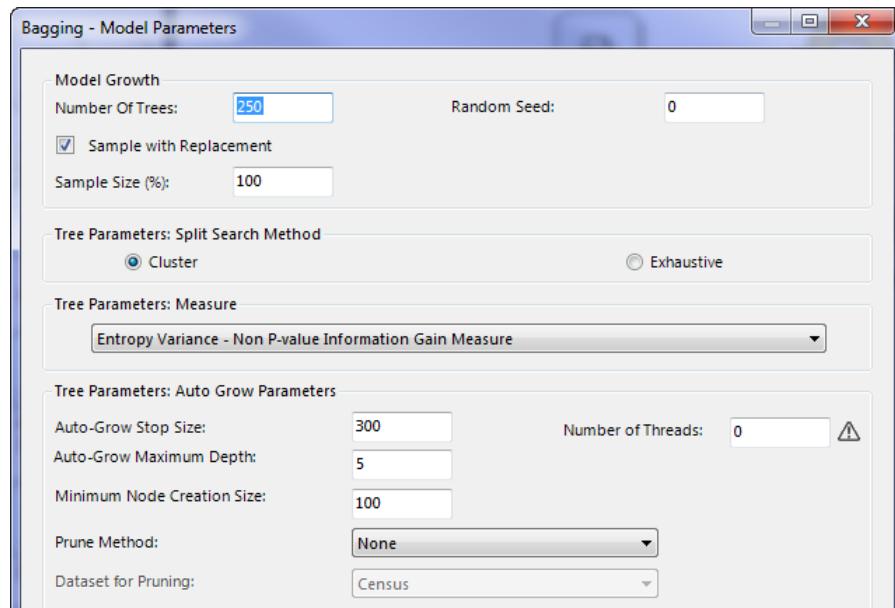
8.5.2 Bagging

For this example the same data & project are used. Create a new **Workflow** in the same project. Link the **Development** partition, and a **Bagging** node, not shown.

The first page provides the opportunity to name the model, here, the default is accepted. The second dialog, allows specification of the dependent and independent variables, here these are set as per the **Boosting** model where the **Dependent variable** is *Response*, and all fields bar *Customer ID* and *fnlwgt* are included.

The third dialog provides parameters that govern model growth and is illustrated in figure 8.6.

Figure 8.6: Bagging - Model Parameters



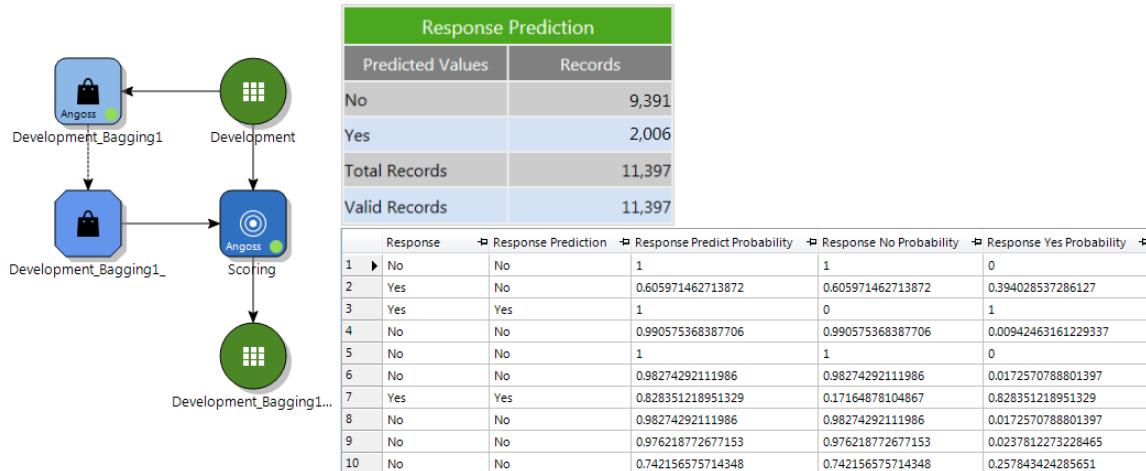
Available options, beyond the default decision tree parameters, are minimal. Additional options are:

- **Sample with Replacement** Selected by default
- **Sample Size(%)** Proportion of parent to use in model building

Here, the **Number of Trees** is changed to 100. All other options are accepted at their defaults. Clicking **Run** completes the process and outputs results.

As for boosting models, output is minimal and reflects setup options. The final score for each record is determined by averaging results from individual models and can only be accessed by scoring the data as illustrated in figure 8.7.

Figure 8.7: Scored Bagging Model and Data



8.5.3 Random Forest

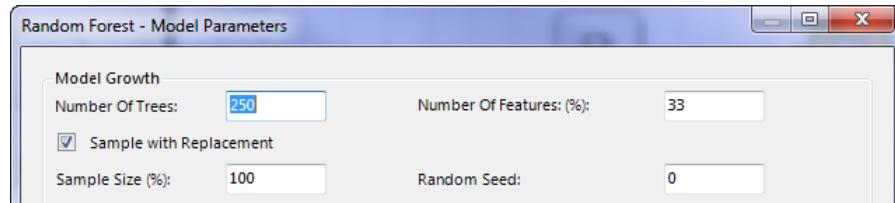
Using the same project: create a new **Workflow**, link the **Development** partition and connect and open a **Random Forest** node, not shown.

The first dialog shows connections and allows specification of a model name, here the defaults are accepted. The second dialog allows specification of the dependent and independent variables. Here these are set as per the **Boosting** and **Bagging** models where the **Dependent variable** is *Response*, and all fields bar *Customer ID* and *fnlwgt* are included.

Click **Next** to navigate to the third dialog. This dialog contains all modifiable parameters.

As with **Boosting** and **Bagging** model parameters, most options relate to **Decision Tree** options. Additional aspects related to **Random Forests** are illustrated in figure 8.8.

Figure 8.8: Random Forest - Model Parameters



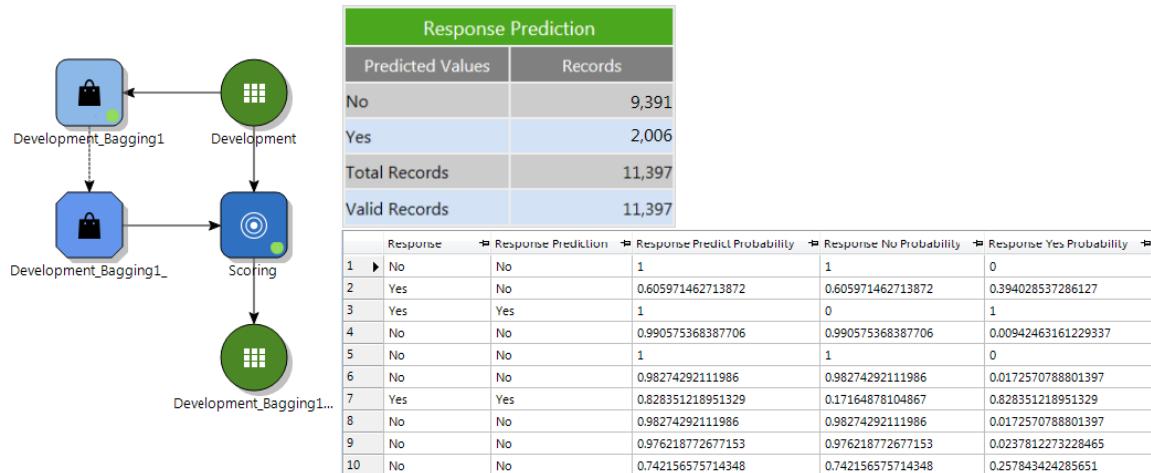
As **Random Forests** are a development of **Bagging**, options are identical with the additional of **Number of Features: (%)**; which determines what percentage of independent predictors are randomly selected at each model iteration.

Here, as with the previous models, the **Number of Trees** is changed to 100. All other options are accepted at their defaults.

Click **Run** to build the model. Again, output is minimal and scoring of the data is required to proceed to

further evaluate the model. Note that scored results are identical to other ensemble methods.

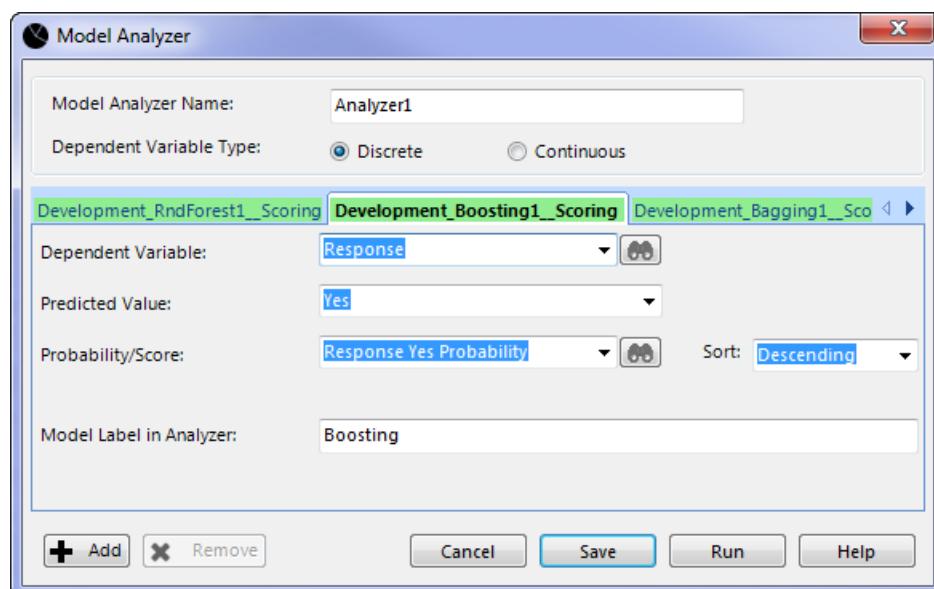
Figure 8.9: Scored Random Forest



8.5.4 Comparing Results

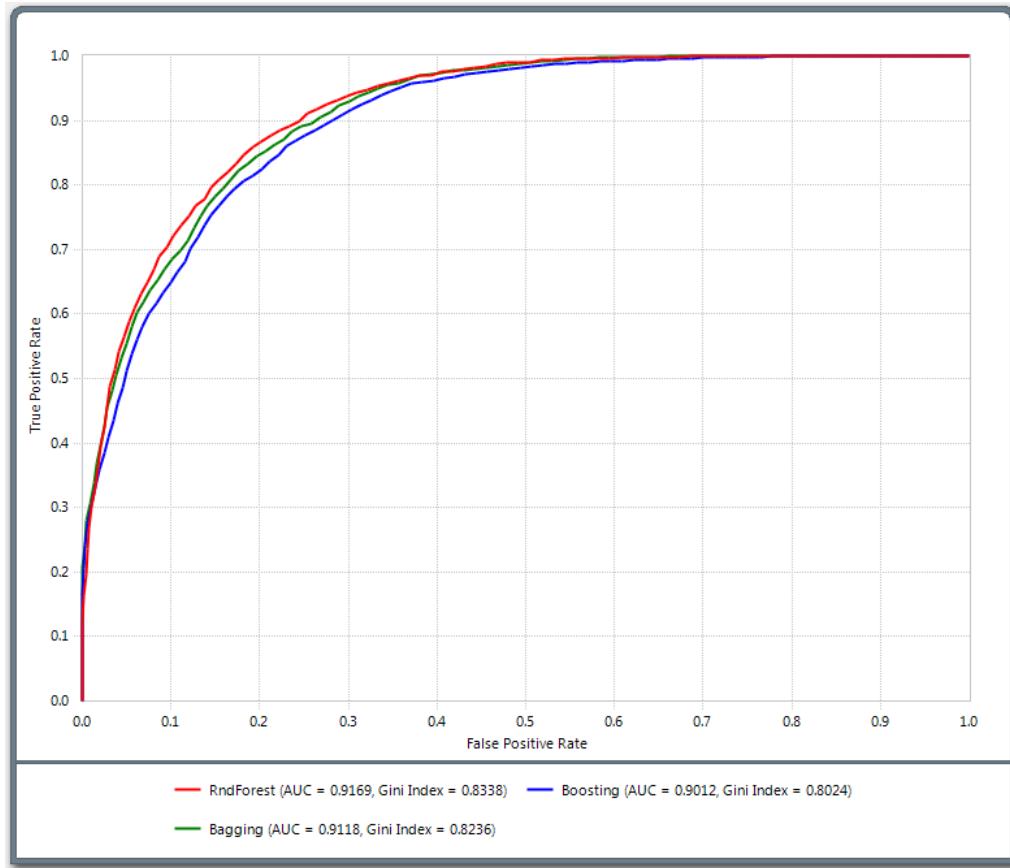
The models can be compared using various methods. Here, the **Model Analyzer** is used. Connect the scored dataset for each model. Options are set as illustrated in figure 8.10 where the **Predicted Value:** is set to **Yes** and the **Probability/Score:** is set to the variable: *Response Yes Probability*. Assign an appropriate label to easily identify each model results in the analyser. Here the labels: **Boosting**, **Bagging** and **RndForest** are used.

Figure 8.10: Setting Analyser Options



Once options have been set appropriately, click **Run** to generate. Open the **Analyser** results on the **ROC Chart** tab.

Figure 8.11: Comparing Results



As can be seen the models are very close in terms of performance. Here the **Random Forest** model wins by a whisker with an **AUC** value of 0.9169.

Exercises

1. Create a new tab, and either import or create a link to the **Census** dataset
2. Drag a **Boosting** node onto the canvas. Connect it to the Census dataset and open the wizard
3. Select the dependent variable and appropriate independent variables of your choice.
4. Get a stopwatch or a timer ready. Set the **Number of Trees** option to 100. Click **Run**, and time how long the model takes to run
5. When complete, record model build time. Right click the **Boosting** node and click **Modify**. This time set the **Number of Trees** option to 250. Click **Run**, and note the model build time
6. Repeat with 500 trees. If it's not taking too long on your hardware, try repeating with some bigger numbers.
7. What is the largest number of trees for which the model will run in under 5 minutes?
8. Repeat 2-7 for **Bagging**
 - (a) Note, when setting up the **Bagging** model, the previous **Boosting** model can be used as a template. This will auto select the dependent and independent variables. As a result of doing this, all that needs to be addressed and set, as are the model parameters
 - (b) To use the previous model as a template; simply add the **Bagging** node and connect the previously run **Boosting** model. When accessing the **Bagging** dialog, note that the dependent and independent variables are automatically selected
9. Time permitting; repeat 2-7 for **Random Forest** models
10. Once a satisfactory model or models exist, validate them. Which of the three ensemble methods is most accurate?

Chapter 9: Model Evaluation and Validation

9.1 Introduction

Prior to modelling, a dataset is typically partitioned into a **Development** and **Validation** partition.

Models can then be tested on both the dataset that was used to build the model; a process called *Model Evaluation*, and, on a holdout or test dataset, called *Model Validation*.

The purpose of *Model Validation* is to test how well a model performs on new data, and in general, the validation or testing dataset acts as a proxy for the population the model is ultimately deployed on.

Assessing whether the rules/segments found using the model are applicable to the population is determined by assessing their validity on the validation or testing dataset.

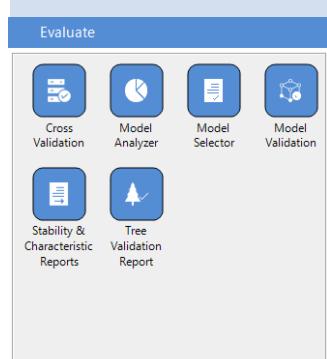
As a result of completing this chapter users should be able to:

- Evaluate and validate **Decision Tree** models using statistics and charts
- Assess the structural integrity of the model to identify segments/rules that do not validate
- Determine predictor importance using the **Variable Importance** node

9.2 Evaluate Palette

KnowledgeSTUDIO model evaluation capabilities are available via a series of nodes found in the **Evaluate** palette. Relevant nodes are described in table 9.1.

Table 9.1: Evaluate Palette

Palette	Node	Description
	Model Analyzer	Graphs and statistics to assess model accuracy
	Model Validation	Scores data, outputs statistics and confusion matrix
	Model Selector	Select best performing model based on a selected Evaluation Metric
	Cross Validation	Traditionally used with small dataset, but can be applied regardless
	Tree Validation Report	Compares tree structure across datasets
	Stability & Characteristic Report*	Use to assess scorecards

*Not covered on this course

KnowledgeSTUDIO evaluation and validation features can be further classified as:

- Statistical Validation
- Thorough Validation
- Business Validation

NOTE: This chapter explains the following concepts by automatically growing a **Decision Tree** with all available independent variables considered.

9.3 Statistical Validation

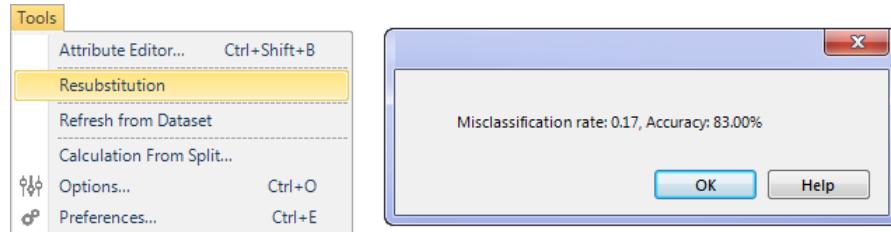
Statistical validation can be assessed using the re-substitution statistics and, more thoroughly, using the **Model Validation** node.

9.3.1 Re-substitution Statistics

Resubstitution, available from the **Tools** menu of the **Tree View** tab, can be used to quickly assess accuracy and the misclassification rate of a **Decision Tree** model as applied to the development dataset.

The Accuracy value reports the approximate correct classification rate of records of the current tree when applied to the development dataset.

Figure 9.1: Resubstitution



Re-substitution results are minimal, displayed as illustrated and not saved. They are used as a quick assessment of overall model accuracy.

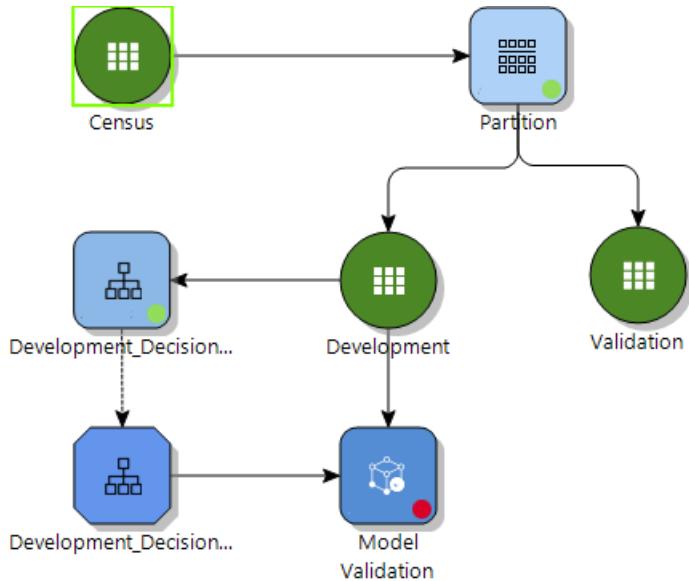
A more thorough assessment can be conducted using the **Model Validation** node, from the **Evaluate** palette.

9.3.2 Model Validation Node

The **Model Validation** node generates a new dataset containing new fields relating to the model scored, a confusion matrix and statistical report are also generated.

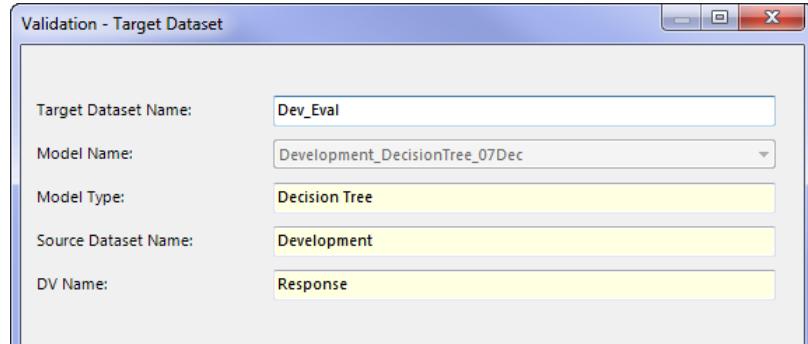
To generate results using the **Model Validation** node; drag it from the **Evaluate** palette to the **Workflow** canvas and connect as illustrated in figure 9.2.

Figure 9.2: Model Validation Node



Either double click the **Model Validation** node or right click and select **Modify** to access options. The first dialog is **Validation – Target Dataset**.

Figure 9.3: Validation - Target Dataset



Most options are preset and dependent on connections; for example, **Model Name**, **Model Type** and **DV Name** are taken from the **Model Instance**. The source dataset is taken from connected nodes; in this case the **Development** partition.

The only modifiable option is the **Target Dataset Name**; the default syntax when applying names to instances is:

<Model_Instance_Name>_<Validation>

Depending on previous assignment the name can get quite long. In this example the name is changed to **Dev_Eval**. Click **Next >** to move to the **Validation – Field Mapping** dialog.

Figure 9.4: Validation - Field Mapping

	Model Field Names	Dataset Field Names
▶	age	age
	education	education
	occupation	occupation
	relationship	relationship
	hours-per-week	hours-per-week
	Response	Response
	num-products	num-products
	capital-gain	capital-gain

In this dialog fields names from the model instance are mapped to corresponding fields in the attached dataset. If evaluating the model on the data used to create it, this step can be skipped as the correct field name will be automatically mapped.

However if the model is being applied to an alternative dataset and field names do not correspond, then clicking any field name in the **Dataset Field Names** column activates a drop-down that can be used to map fields correctly.

Click **Next >** to move to the **Validation – Validation Fields** dialog.

Figure 9.5: Validation - Validation Fields

Item	Field Name	Include	Cut Off	Hit Range
Response Prediction	Response Prediction	<input checked="" type="checkbox"/>		
Response Probability of Prediction	Response Predict Prob	<input checked="" type="checkbox"/>		
Response No Probability	Response No Prob	<input checked="" type="checkbox"/>	0.5	
Response Yes Probability	Response Yes Prob	<input checked="" type="checkbox"/>	0.5	
Response Prediction Correct	Response Correct	<input checked="" type="checkbox"/>		
Response Node ID	Response Node ID	<input checked="" type="checkbox"/>		
Response Node Number	Response Node Number	<input checked="" type="checkbox"/>		

The **Validation – Validation Fields** dialog lists all fields created. Fields can be deselected.

If the dependent variable is binary the **Cut Off** value can be specified to determine the prediction. Values will always sum to 1. Available fields are listed table 9.2.

Table 9.2: Validation Fields Created

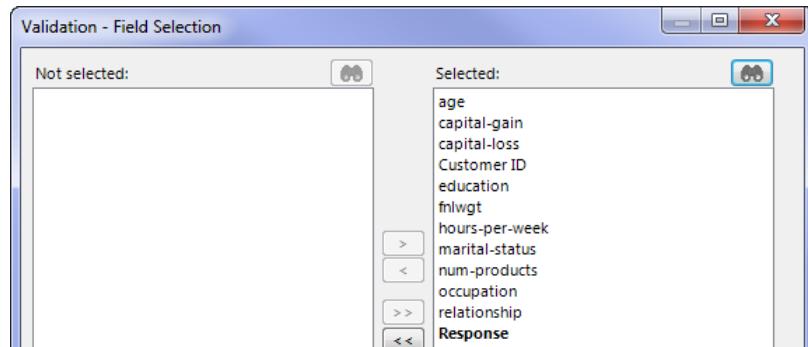
Field	Description
Prediction:	Predicted outcome value**
Probability of Prediction	Probability/confidence of the predicted value
No/Yes Probability	Category probability – vary cut-off with binary dependent variable*
Prediction Correct	Whether the predicted value was correct
Node ID	Node ID for each record
Node Number	Node number for each record

*Probabilities will update & sum to 1

**Predicted value will change based on cut-off value

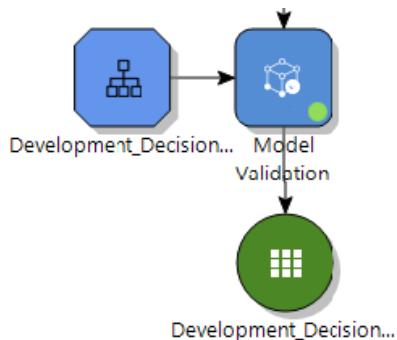
Once options have been set, click **Next >** to move to the **Validation – Field Selection** dialog.

Figure 9.6: Validation - Field Selection



This dialog provides options to determine which fields from the original dataset will be included in the output dataset. Click **Run** to create the scored dataset.

Figure 9.7: Workflow with Scored Data



To open the results; either double-click on the dataset node or right click and select **Open View**. The output opens on the **Report** tab containing the **Confusion Matrix** for the dependent variable and an associated **Statistics** table.

The **Confusion Matrix** provides a count of **Predicted** and **Actual** values and includes the category correct classification rate.

Figure 9.8: Confusion Matrix

Confusion Matrix - DV		
		Predicted
Actual	No	7842 (90.27%)
	Yes	1039
	True Negative	845
	False Negative	1671 (61.66%)
	False Positive	
	True Positive	

Of the 8669 records in the *No* category; 8240 were correctly predicted as *No*, and 447 incorrectly predicted as *Yes*. Similarly, of the 2710 records in the *Yes* category 1620 are correctly predicted and 1090 incorrectly predicted.

Summing the values in the diagonal; i.e. the correct classifications, and dividing by the total number of cases, returns the correct classification rate, in this case, 81.86%. This value is also evident in the statistics table along with other informative measures used to evaluate the model.

Figure 9.9: Validation Report Statistics Table

Statistics	
Total Records	11,397
Correctly Predicted	9,513
Percentage	83.47
Valid Records	11,397
Entropy Explained	0.37
K-L divergence	0
Cross Entropy	0.55
Entropy of predict	0.53
Entropy of actual	0.55

A description of each statistic is provided in table 9.3

Table 9.3: Validation Report Statistics

Statistic	Description
Total Records	Number of records in the validation dataset
Correctly Predicted	Number of correctly predicted records
Percentage	Percentage representation of the no. correctly predicted
Valid Records	No. valid records in the data. Missing values is most common reason for invalid records
Entropy Explained	Pseudo R-Square statistic, value varies between 0 – 1. Values closer to 1 more desirable
K-L divergence	How different the predicted distribution is from the actual distribution, 0 means identical, very large number means totally different
Cross Entropy	Measures differences between distributions, values closer to 0 more desirable
Entropy of predicted	Entropy of predicted dataset. Measured on interval [0,1]. 0 = all records in one category, for equal distribution entropy = 1
Entropy of actual	Entropy of design dataset, same as Entropy of predicted

The overall accuracy for the model at 81.86% is promising, but can be slightly misleading. The *No* category is predicted with 94.85% accuracy and the *Yes* at 40.22%.

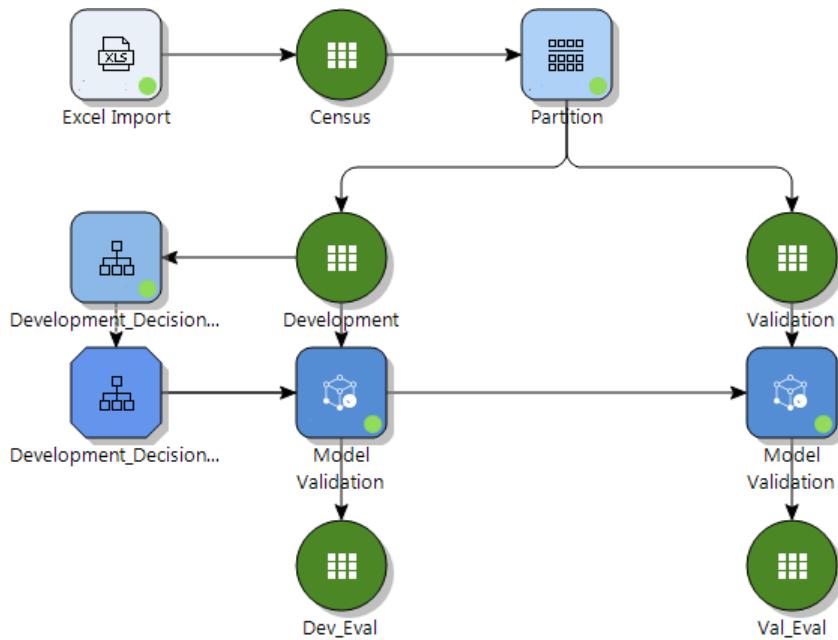
If there is more interest in accurately predicting a specific category, the results should be further assessed.

Once validation statistics are generated for the **Development** dataset, they should be generated for the **Validation** dataset also.

If the model performs significantly better on the **Development** dataset than on the **Validation** dataset, this is a sign of possible over-fitting, more on this later.

Connect a **validation** node as before, but this time to the **Validation** dataset as depicted. Assign the name **Val_Eval**.

Figure 9.10: Model Validation Applied to Partitions



Once created, results can be compared side by side using split screen representations.

Figure 9.11: Comparing Model Results

Validation Report	 Altair	Validation Report	 Altair
Input Model Name: DT_Model_Instance		Input Model Name: DT_Model_Instance	
Input Dataset:Development		Input Dataset: Validation	
Confusion Matrix - Response		Confusion Matrix - Response	
		Predicted	
		No	Yes
Actual	No	7842 (90.27%)	845
	Yes	1039	1671 (61.66%)
Statistics		Statistics	
Total Records	11,397	Total Records	4,884
Correctly Predicted	9,513	Correctly Predicted	4,111
Percentage	83.47	Percentage	84.17
Valid Records	11,397	Valid Records	4,884
Entropy Explained	0.37	Entropy Explained	0.37
K-L divergence	0	K-L divergence	0
Cross Entropy	0.55	Cross Entropy	0.54
Entropy of predict	0.53	Entropy of predict	0.52
Entropy of actual	0.55	Entropy of actual	0.54

NOTE: A point of interest is that some analysts will always refer to model validation statistics and disregard evaluation results, as the validation results estimate how well the model will perform on new data.

The next aspect of model assessment is **Thorough Validation** using the **Tree Validation Report** node.

9.4 Thorough Validation

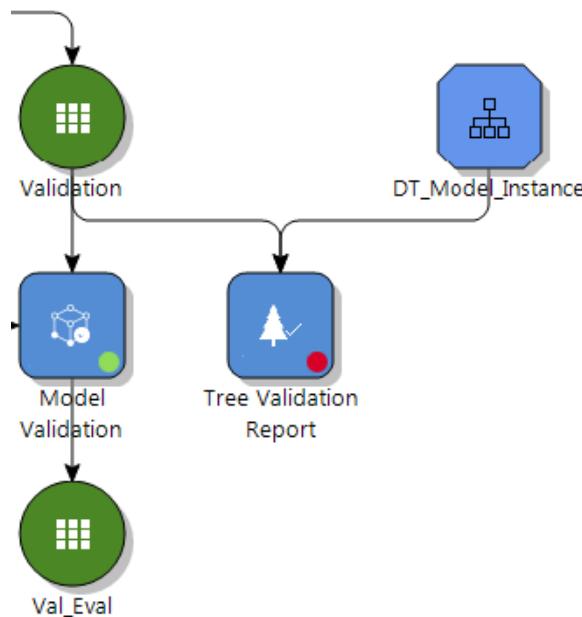
Thorough Validation compares tree structure across both the development and validation partitions. It is used as a means to assess whether the model rules/segments found on the **Development** partition are robust when applied to the **Validation** partition.

The dependent variable distributions are compared node by node and a simple report is generated showing differences between partitions.

To create the report, add a **Tree Validation Report** node from the **Evaluate** palette to the **Workflow** canvas and connect both the **Decision Tree Model Instance** and the **Validation** partition as illustrated in figure 9.12.

Here, rather than connect the model from its current position, the **Model Link** node from the **Model** palette is used. This ensures clarity and neater **Workflows**.

Figure 9.12: Workflow with Tree Validation Report



Open the **Tree Validation Report** node to access options,(not shown). Options are modest and the **Base Tree** and **Testing Dataset** are pre-determined based on connections to the node. Only the **Report Name** can be set.

Accept the defaults and click **Run** to create the report. once complete, open the **Tree Validation Report** node to view results.

NOTE: The results are created as an object in the **Project Pane**.

Figure 9.13: Validation Report Partial View

Tree Validation Report					
Testing Data Source: Validation					
Misclassification rate: 0.17, Accuracy: 83.00%					
All Data					
Node Number = 1					
Category	Design		Validation		
	Freq	%	Freq	%	Diff
No	8,687	76.22%	3,748	76.74%	0.52%
Yes	2,710	23.78%	1,136	23.26%	-0.52%
Total	11,397		4,884		
relationship = Husband , Wite					
Node Number = 2					
Category	Design		Validation		
	Freq	%	Freq	%	Diff
No	2,797	54.89%	1,213	55.39%	0.50%
Yes	2,299	45.11%	977	44.61%	-0.50%
Total	5,096		2,190		

Results are presented depending on the dependent variable type: discreet or continuous. In both cases, a node comparison is applied and displayed.

The **Development** dataset is referred to in the **Design** columns, and the **Validation** dataset is referred to in the **Validation** columns.

A separate section is produced for each node showing the distribution of the dependent variable and the overall difference across partitions.

The node rule is shown in the blue bar at the top of each rectangle. Acceptable differences are generally within a few percent: <= 5%.

The report also shows the misclassification rate and accuracy of the tree model as a whole. In the example above, the accuracy is 85%.

9.5 Business Validation

Business Validation is a broad term that can be applied to model acceptability in data mining. The term is open to interpretation but can be loosely understood as the business benefits of applying the model.

For example:

Targeting the top 30% of records contains 70% of those likely to respond/churn/purchase etc

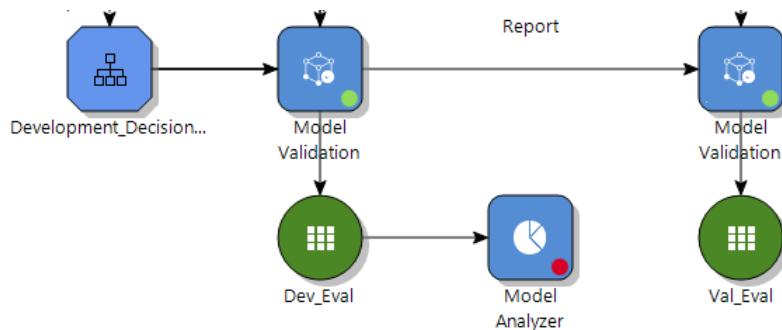
Model observations such as these can translated to bottom line results and real value when metrics such as the cost of contacting customers, cost of acquisition vs. retention, etc. are taken into account.

The **Model Analyzer** generates a series of graphs, tables and statistics that can be used to further assess model acceptability and business benefits. The **Model Analyzer** can be applied to both discrete and continuous dependent variables.

Additionally graphs can be generated simultaneously for multiple partitions and multiple models.

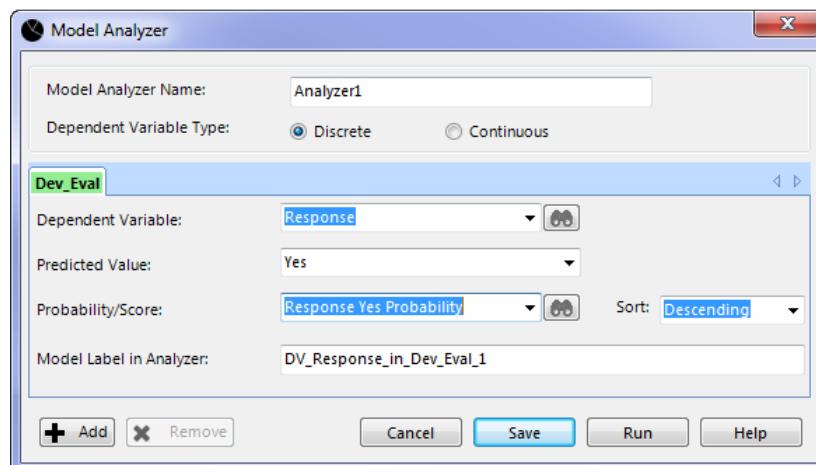
To use the **Model Analyzer** to assess the **Decision Tree** model created earlier; drag the **Model Analyzer** node from the **Evaluate** palette to the **Workflow** canvas and connected the dataset node; **Dev_Eval** as illustrated in figure 9.14.

Figure 9.14: Adding Model Analyzer Node



Accessing the **Model Analyzer** dialog reveals one page with multiple tabs; one for each connected dataset. Currently only one dataset is connected; the **Dev_Eval** partition, hence only one tab is evident.

Figure 9.15: Model Analyser



Model Analyser options are listed in table 9.4.

Table 9.4: Model Analyzer Options

Option	Description
Model Analyzer Name	Assign a name to the output
Dependent Variable Type	Specify the dependent variable type. By default the type is automatically set based on prior models and connections
Dependent Variable	Identify the dependent variable field. By default automatically set based on prior models and connections
Predicted Value	Select the target category to illustrate results for
Probability/Score	The field containing the Predicted Value scores. If the Predicted Value is changed, the corresponding Probability/Score field will also need to be identified
Sort	Sort result in ascending/descending order, by default: descending
Model Label in Analyzer	Label for model as it appears in the analyzer results

In the example, there is interest in assessing the Yes category predictions, i.e. those likely to respond. The **Predicted Value** is changed to Yes and the **Probability/Score** field is correspondingly changed to *Response Yes Prob*. All other options can be left as is.

Click **Run** to generate the results. A **Model Analyzer** named **Analyzer1** is created in the **Project Pane**. To access results either double click the **Project Pane** object or right click the generating node on the **Workflow** canvas and select **Open View**.

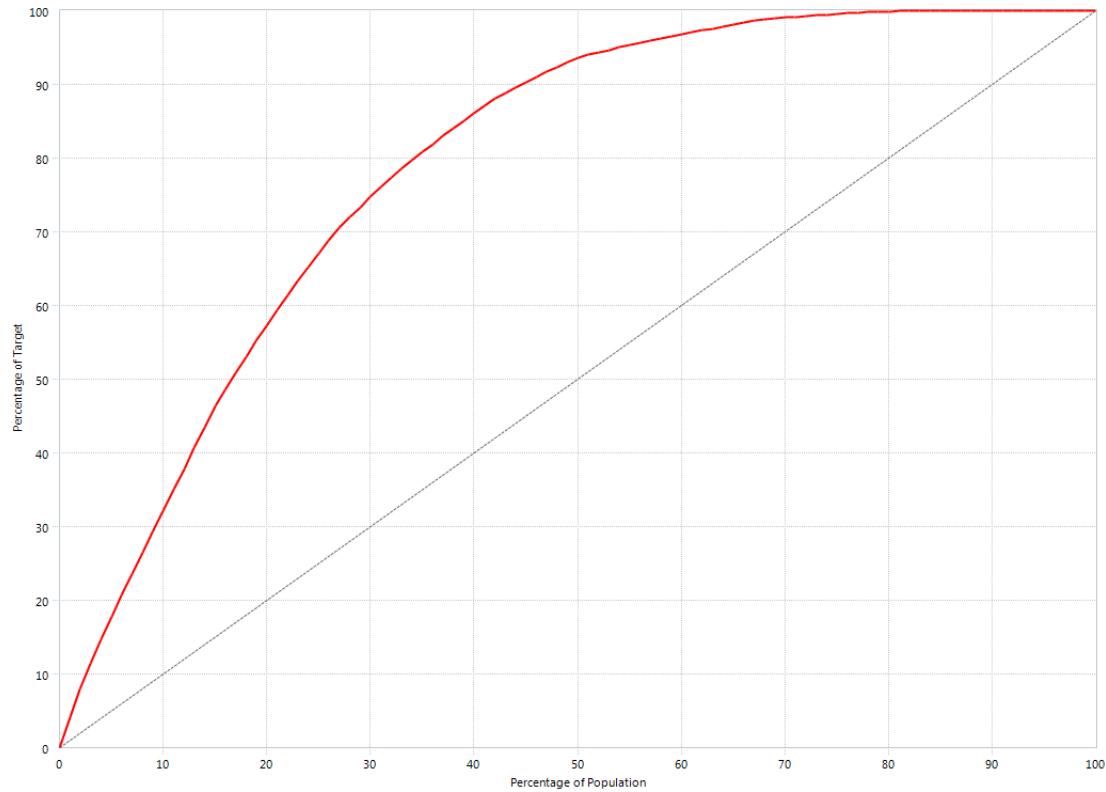
The **Model Analyzer** contains a series of graphs, tables and statistics spread across a series of tabs. Each of these graphs is explained in turn in the following sections.

9.5.1 Cumulative tab and Report

The default tab view is the **Cumulative** tab, this illustrates the **Cumulative** gains charts for the selected category, *Yes*, of the dependent variable.

This has a corresponding tabular representation found in the **Cumulative Lift Report** tab.

Figure 9.16: Cumulative Tab



The **Cumulative** gains chart compares model performance to random selection.

The horizontal **x-axis** refers to the **Percentage of Population** selected. The vertical **y-axis** refers to the **Percentage of Target** hit.

Simply put: the graph shows the proportion of those in the target category contained in the percentage of the population selected. Here the target category is the *Yes* category of the *Dependent Variable Response*.

The diagonal line represents pure chance or random selection, i.e.: if 50% are randomly selected, it should contain 50% of the total number of those in the *Yes* category, and conversely 50% of the total number of those in the *No* category also.

The curve reflects model performance. Selecting the top 40% based on the model predicted scores would contain approximately 78.1% of the total number of those in the *Yes* category. In this case the model is performing approximately 38% better than chance.

The **Cumulative Lift Report** tab provides the same information in tabular form. This table gives the model lift in decile increments. The 4th decile shows the value identified from the **Cumulative Lift** chart.

Figure 9.17: Cumulative Lift Report

Decile	Target Volume	Lift	Cumulative Lift
1	1139	32.15	32.15
2	1140	25.13	57.28
3	1140	17.54	74.82
4	1139	11.31	86.13
5	1140	7.50	93.63
6	1140	3.16	96.79
7	1140	2.28	99.06
8	1139	0.84	99.90
9	1140	0.10	100.00
10	1140	0.00	100.00

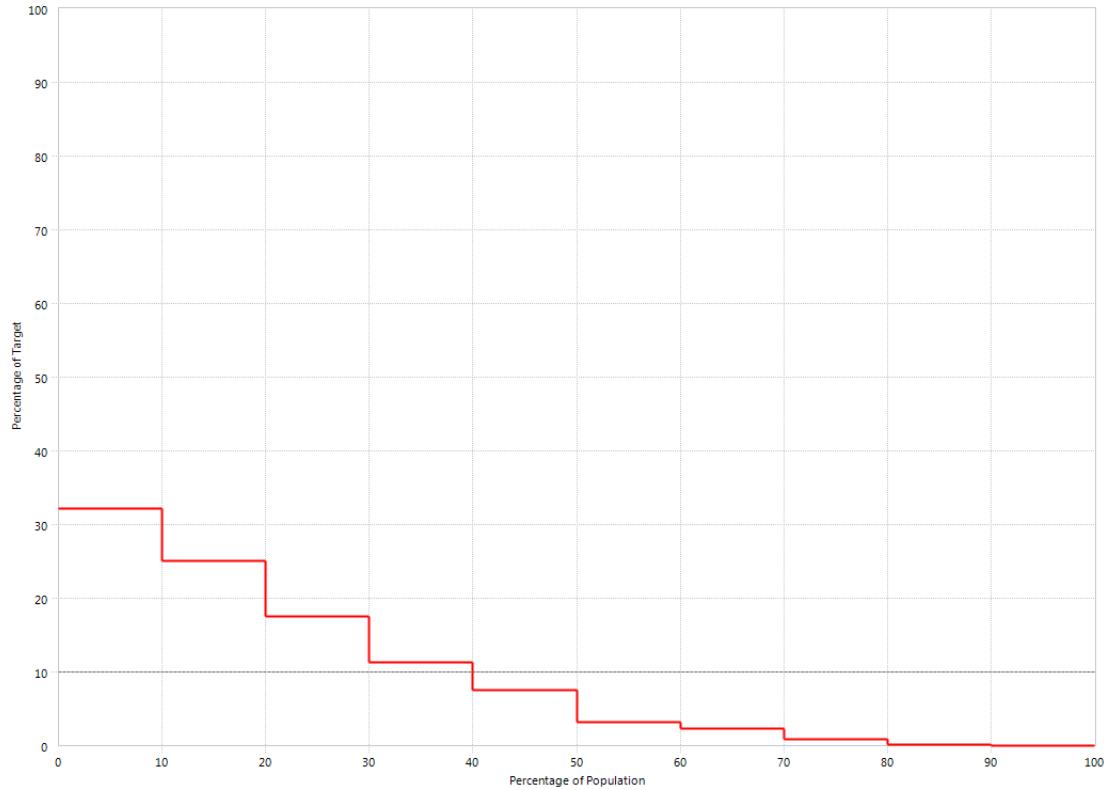
9.5.2 Lift Chart tab and Report

The **Lift Chart** tab shows the lift chart for the model versus random selection. It is a scaled ratio representation of model performance versus random selection at each decile.

For example, referring back to the cumulative gains chart; draw an imaginary vertical line beginning at the value: 10, on the x-axis. This cuts the diagonal at 10 and the model line at approximately 34.

These values represent random selection and model performance for the first decile. Divide model performance by random selection, gives approximately 3.4, multiply by 10 gives 34, hence the first decile value on the **Lift Chart**.

Figure 9.18: Lift Chart Tab



The interpretation of the **Lift Chart** is straightforward. For example: If I choose the top 10%, my model is more than three times as likely to identify a **Yes** category member, than random selection.

This chart is also useful for detecting hidden problems with the model. The model when plotted should be a monotonically decreasing function, i.e. each decile should have a lower lift than the previous decile. If this is not the case, i.e. the lift decreases and then subsequently increases, then the model should be treated with caution and not used.

NOTE: You can see there is a horizontal line drawn at the value 10 - why? This line represents random selection, which always has a 1/1 ratio.

The intersection of the model line and the random selection line reflects the point at which model performance is no better than random selection. This happens at approximately the 40th percentile.

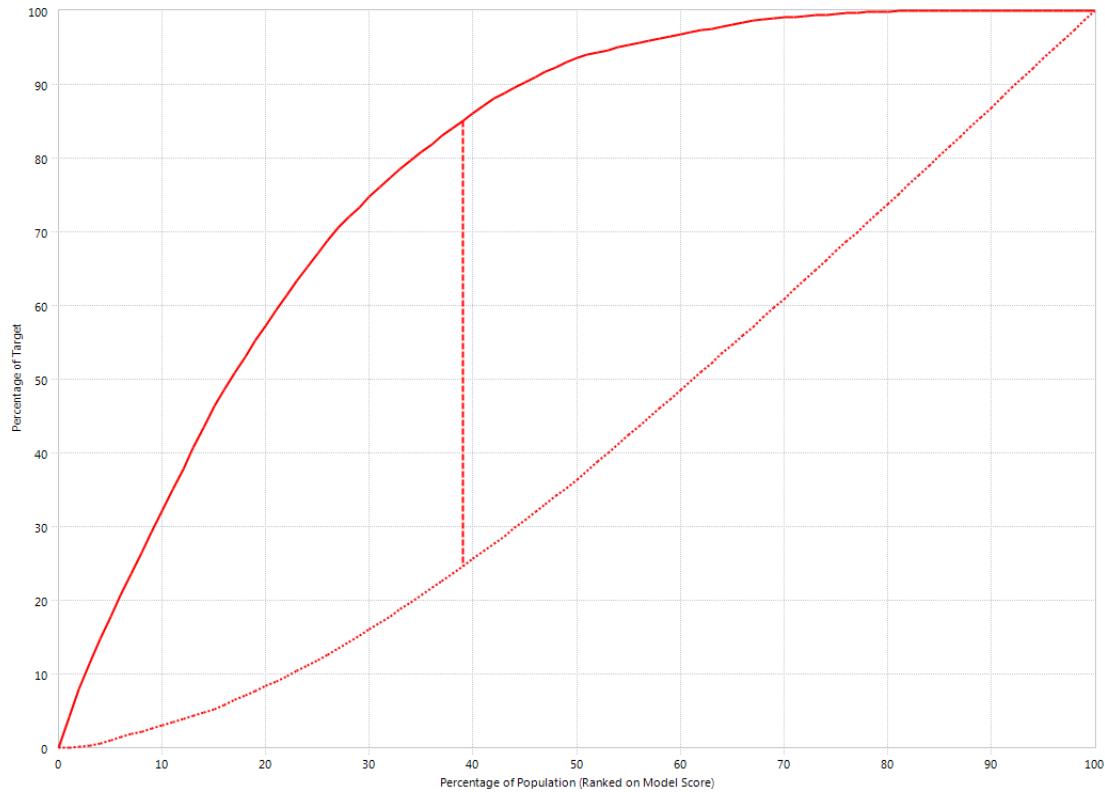
This information is also given in tabular form on the **Lift Report** tab (not shown).

9.5.3 K-S Chart tab

The **Kolmogorov-Smirnov** curve shows the cumulative distribution of the dependent variable categories.

The **K-S statistic** measures the difference between the two functions and returns the maximum value, i.e. where the model maximizes separation between **Yes** and **No** categories.

Figure 9.19: KS Chart Tab



The **K-S** statistic is also reported in brackets to the right of the model name, and ranges from 0 (random scores) to 1 (perfectly predicted scores).

Here the maximum value of the **K-S** statistic is 0.614. This is calculated as the difference between the Percentage of Target value for each of the curves at the maximum separation point; $(81.8 - 21.4)/100 = 0.614$.

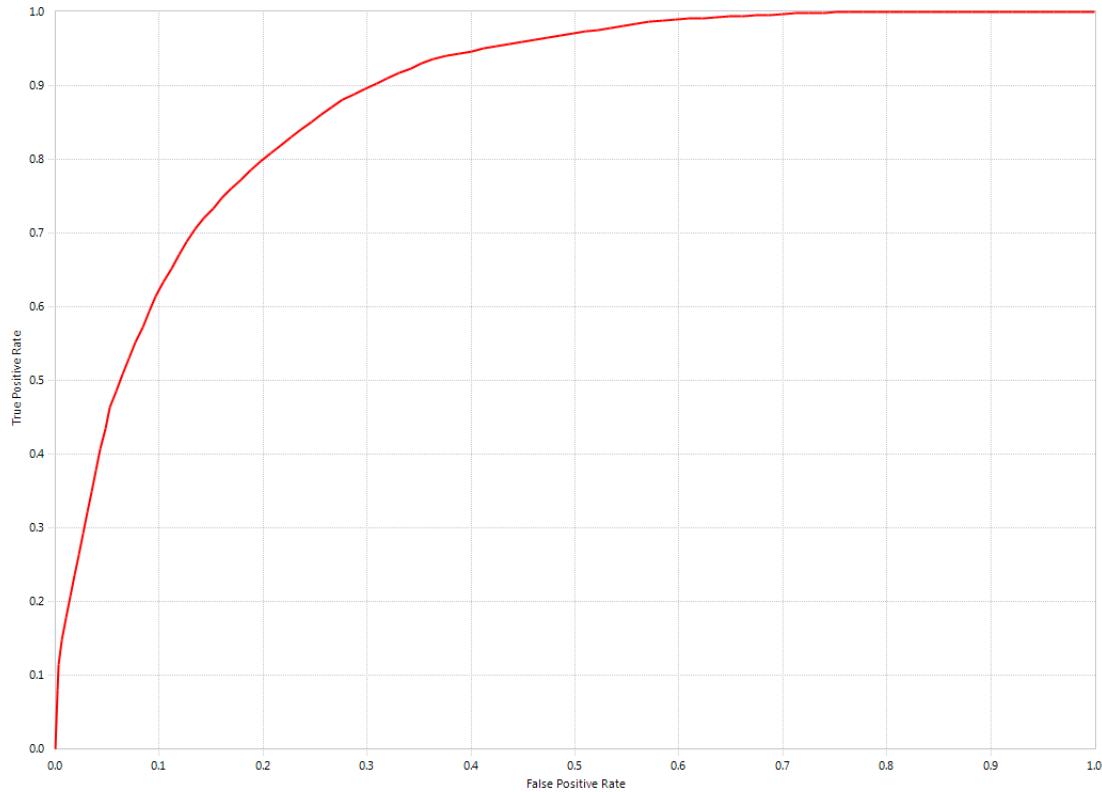
The optimum population selection percentage based on the model is approx. 36%, i.e. selecting the top 36% as identified by the model will give the greatest separation between *No* and *Yes*.

NOTE: A criticism of the **K-S** statistic is that the costs of misclassification are equal; this means that missing a *Yes* responder has the same penalty as targeting a *No*. In reality, this is often not the case in practice.

9.5.4 ROC Chart Tab

The **ROC Chart, Receiver Operating Characteristic**, compares the **True Positive Rate** to the **False Positive Rate** as the discriminant, or classification threshold changes.

Figure 9.20: ROC Chart Tab



The *x-axis* shows the proportion of the non-target category misclassified as a result of obtaining the *y-axis* accuracy for the target category of the dependent variable.

The graph values highlighted can be interpreted as:

To obtain an accuracy of 80% for the *Yes* means 20% of the *No* category will be misclassified.

The area under the curve is known as the **AUC** or **C-Statistic**. This ranges between 0.5 (for random classification) and 1 (for perfect classification). Any value in excess of .70 means the model is acceptable. The AUC here is 0.8899 which is more than adequate for a reliable model.

The **Gini Index** is also displayed and is equal to $(2 \times AUC) - 1$. The **Gini Index** is no better or worse than the **AUC** statistic, but ranges between 0 and 1 instead of 0.5 and 1.

NOTE: The **True Positive Rate** and the **False Positive Rate** are calculated from a confusion matrix. Consider the cells in a **Confusion Matrix** as illustrated in figure 9.21.

Figure 9.21: Confusion Matrix

		Predicted Value	
		No	Yes
Actual Value	No	<i>True Negatives (TN)</i>	<i>False Positives (FP)</i>
	Yes	<i>False Negatives (FN)</i>	<i>True Positives (TP)</i>

The **ROC** curve essentially shows the variation in predictive accuracy of a specific category of the dependent variable at different cut-off values.

The prediction accuracy is expressed by the following metrics in terms of *TN*, *TP*, *FN*, *FP*:

- **Sensitivity** $\text{True Positive Rate} = \frac{TP}{(TP+FN)}$
- **Specificity** $\frac{TN}{(TN+FP)}$
- **False Positive Rate** $\frac{FP}{(TN+FP)} = 1 - \text{Specificity}$

The **ROC** curve is created by plotting the **True Positive Rate**; *TPR*, or Sensitivity, on the vertical axis versus the **False Positive Rate**; *FPR*, or $1 - \text{Specificity}$, on the horizontal axis.

The cut-off used to construct the confusion matrix is varied from 0 to 1. Both *TPR* and *FPR* vary within the interval [0,1].

9.5.5 GOF Statistic tab

The **GOF Statistic** tab gives the **Hosmer-Lemeshow** goodness of fit test for the model. This measures whether or not the predictions of the model differ significantly from the actual data.

Figure 9.22: GOF Statistics Tab

	Group	(Response = Yes) Observed	(Response = Yes) Estimated	(Response != Yes) Observed	(Response != Yes) Estimated	Total count
▶	1	0	0	1959	1959	1959
	2	6	6.00000000000002	589	589	595
	3	42	41.999999999998	1356	1356.000000000002	1398
	4	63	63.000000000006	912	911.999999999943	975
	5	120	120	1028	1028	1148
	6	173	173	709	709	882
	7	310	310	805	805	1115
	8	478	478.000000000001	636	635.999999999999	1114
	9	381	380.999999999999	298	298.000000000001	679
	10	1137	1137	395	395	1532

The **Hosmer-Lemeshow** goodness of fit test is a variation of the **Chi-Squared** test. It compares the test statistic, seen in figure 9.23, based on the difference between the observed and expected frequencies, to a χ^2 distribution.

Figure 9.23: Hosmer-Lemeshow Goodness-of-Fit Test

$$\sum_{i=1}^g \frac{(O_i - N_i \bar{p}_i)^2}{N_i \bar{p}_i (1 - \bar{p}_i)}$$

The test divides the population into deciles based on descending predicted probabilities before comparing the expected and observed frequencies.

Creating 10 groups consumes 1 extra degree of freedom compared to a regular **Chi-Squared** test, therefore, there are only 8 degrees of freedom, for the **Hosmer-Lemeshow** test[1].

NOTE: If some of the deciles have no observations, then there may be less than 8 degrees of freedom.

The **Null Hypothesis** being tested is that the observed frequencies, (e.g. the actual values) do not differ significantly from estimated frequencies (e.g. those predicted from the model).

If the significance is less than 0.05, the **Null Hypothesis** is rejected, meaning there is a (statistically) significant difference between model predictions and actual dependent variable values.

If the test statistic is greater than 0.05, then there is not enough evidence to reject the **Null Hypothesis**, in other words we cannot say that model predictions differ from actual observed values [1].

Most often, when doing hypothesis testing it is desirable to observe a significant difference. In this case the opposite is true and the **Null Hypothesis** is desirable, in other words, it is desirable for their to be no significant difference between the predicted and actual values, meaning the model predicts the dependent variable well.

NOTE: As dataset size increases, this test becomes more and more likely to show a difference between predicted and observed frequencies; in other words, for large dataset sizes (>400 records), it is more likely to observe a significant difference.

Also, note that the test statistic will always be 0 for the development partition for **Decision Tree** models.

9.5.6 Profit Curve Tab

The **Profit Curve** provides a quick **Return-on-Investment, ROI**, calculation based on model performance, business specific costs and expected returns. This curve can aid in determining the optimum cut-off.

The Profit Curve requires user input for the following parameters:

- Revenue per unit
- Fixed cost
- Cost per unit
- Population size

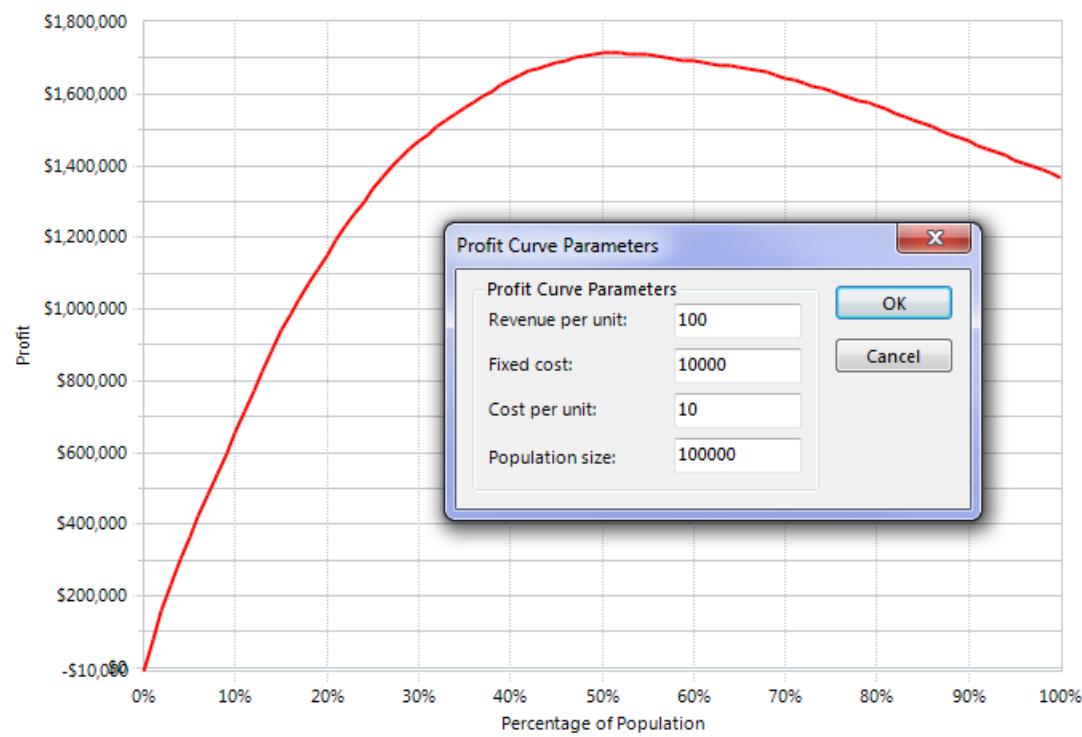
These values are more easily interpreted in relation to a marketing campaign; however they can be used

in any scenario. The table provides an explanation of the parameters based on a marketing campaign.

Table 9.5: Profit Curve Parameters

Parameter	Description
Revenue per unit	How much will be made from a responder
Fixed cost	Overall costs of marketing campaign
Cost per unit	How much it costs per unit of marketing material, e.g. mailer
Population size	Population size model will be deployed on

Figure 9.24: Profit Curve Parameters & Chart

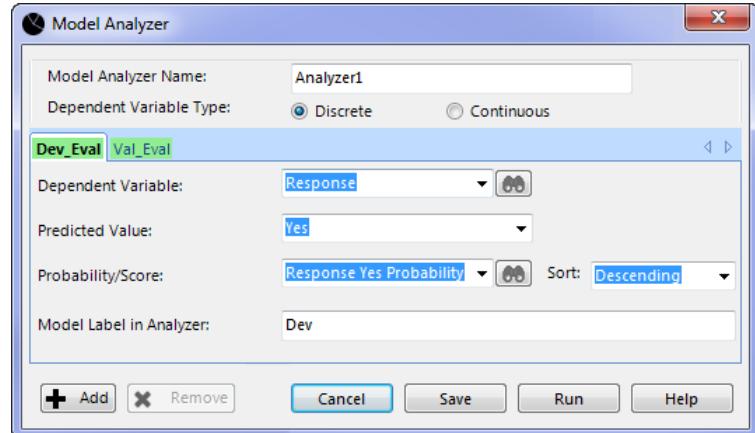


Notice that the highest profit can be gained by targeting the top 45% of the population. In a marketing campaign scenario, less profit will be gained by contacting the whole population, since money will be spent advertising to people that are unlikely to respond.

The **Model Analyzer** is a valuable tool for model evaluation. Most analysts and business users focus on a selected number of graphs while others may use more. Model Analyzer can be used for model validation & comparing models applied to the same dependent variable.

Connecting the **Validation** partition is straightforward. Once connected accessing the **Model Analyzer** dialog now shows two tabs, one for each connected partition.

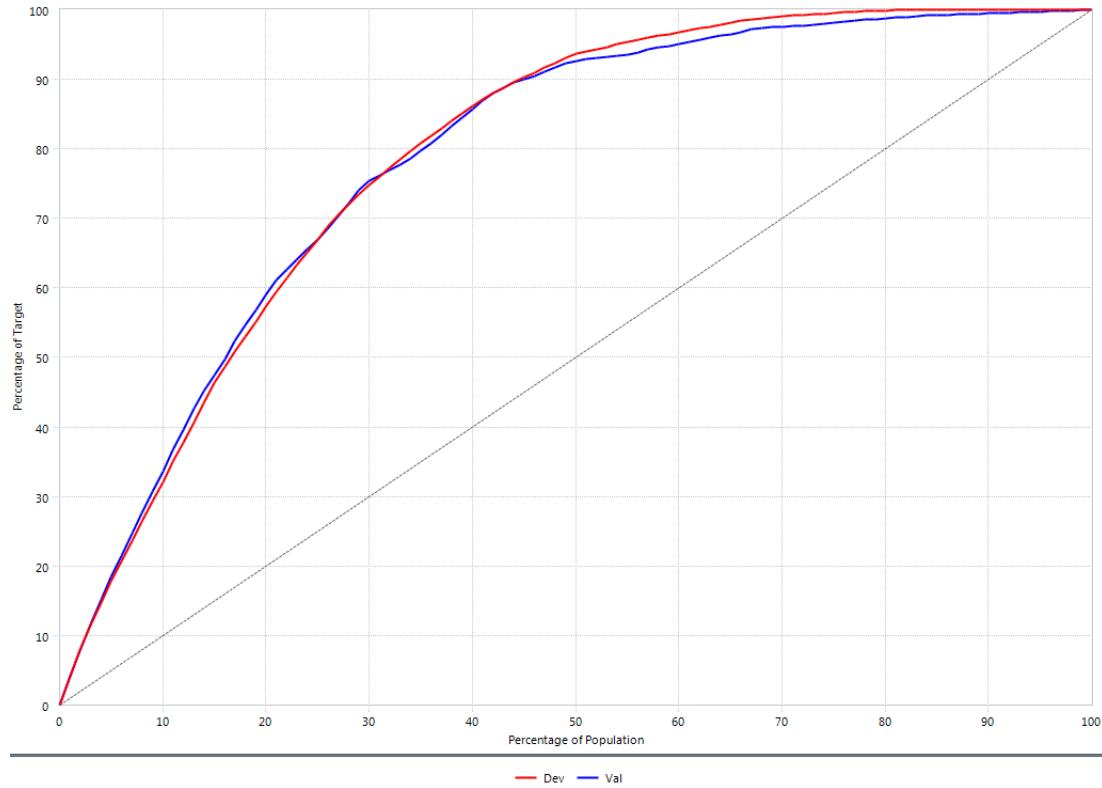
Figure 9.25: Multiple Connections



The default **Predicted Value** and **Probability/Score** fields must be changed from their default of the first dependent variable category, in this instance; *No*, to the category of interest; *Yes*.

Once completed, clicking **Run** adds an additional set of representations for the **Val_Eval** partition.

Figure 9.26: Model Analyser with Added Results



As can be seen there are separate lines for the model applied to both partitions. Both models track each other across all charts implying that the model validates well.

9.6 Assessing Variable Importance

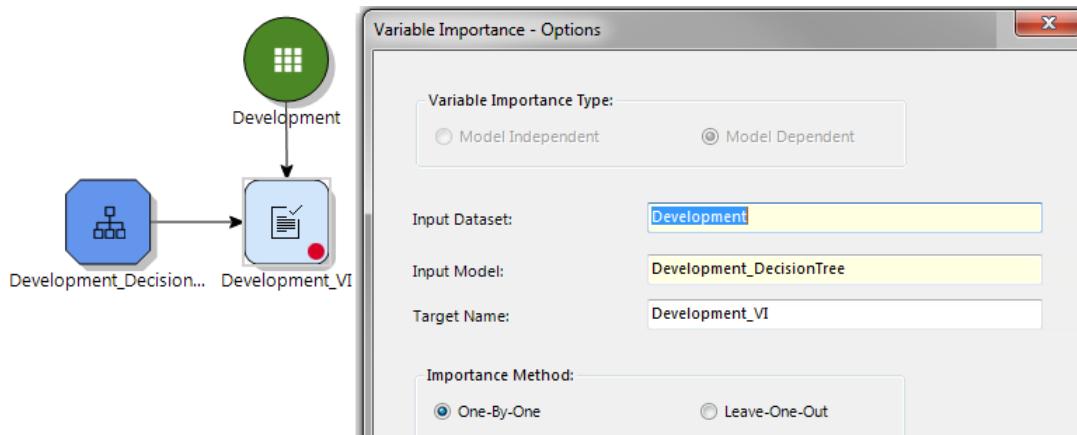
One element that is not as easily assessed for **Decision Trees**, in comparison to other modelling techniques, is variable importance. This is essential to understand the influence of each **Independent Variable**.

The **Variable Importance** node is ideal for this purpose and outputs graphic representations illustrating the ranked relative importance of each variable in the model, the most important given a value of 100.

To begin, a new **Workflow** is created and the **Decision Tree** model and **Development** partition are included via the **Model Link** and **Dataset Link** nodes respectively.

A **Variable Importance** node is added, connected and opened as illustrated in figure 9.27.

Figure 9.27: Variable Importance Node



The node provides two **Variable Importance Types**: **Model Dependent** and **Model Independent**, each automatically selected based on whether a model is connected or not.

If a model is connected, as in this case, **Model Dependent** is automatically selected. The model **Independent Variables** are assessed against the predicted **Dependent Variable** values.

If no model is connected, a **Dependent Variable** must be selected.

Once a **Variable Importance Types**: is set, the method of determining variable importance must be chosen. Two options are provided: **One-by-One**, or **Leave-One-Out**.

One-by-One assesses the univariate relationship between each predictor and the **Dependent Variable**, this of course, does not take into account the effect of other predictors.

The **Leave-One-Out** method on the other hand builds two models: one with all variables and one with all variables minus the predictor being assessed.

In either case, regression models and a scaled **Likelihood Ratio** are used to assess relative predictor importance.

For this example, **Leave-One-Out** is selected as the **Importance Method**. This is appropriate as there is interest in assessing the relative importance of each predictor in the model.

Clicking **Next >** opens the **Variable Importance - Field Mapping** dialog.

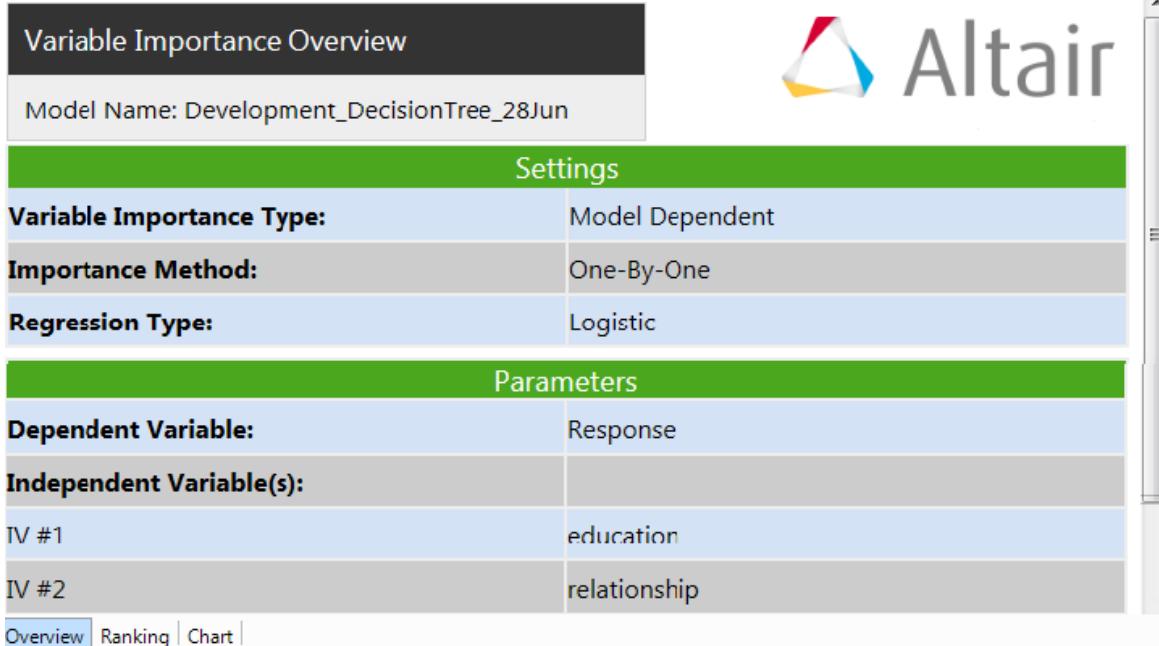
Figure 9.28: Variable Importance - Field Mapping

Variable Importance - Field Mapping		
	Model Fields	Dataset Fields
▶	education	education
	relationship	relationship
	Response	Response

The implication of this page means that any dataset can be assessed against any model. This is a valuable asset if there is interest in determining whether variable importance is consistent on, generally, the most recent data or on a **Validation** partition.

Click **Run** to generate results and once complete, open to assess.

Figure 9.29: Variable Importance - Results



The screenshot shows the 'Variable Importance Overview' interface. At the top, it displays the 'Model Name: Development_DecisionTree_28Jun'. Below this is a 'Settings' section with the following parameters:

Variable Importance Type:	Model Dependent
Importance Method:	One-By-One
Regression Type:	Logistic

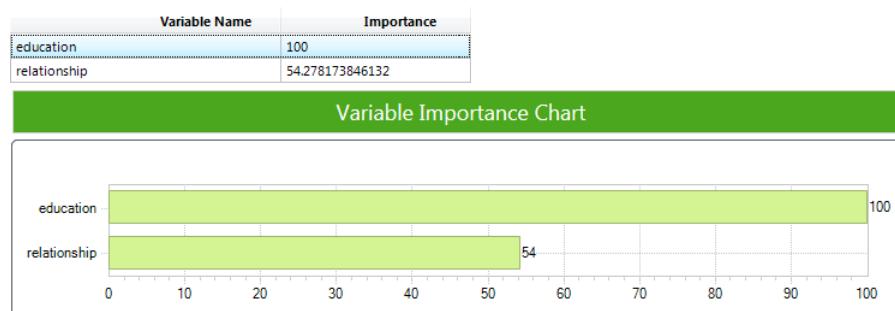
Below the settings is a 'Parameters' section with the following details:

Dependent Variable:	Response
Independent Variable(s):	
IV #1	education
IV #2	relationship

At the bottom, there are three tabs: 'Overview' (selected), 'Ranking', and 'Chart'.

Results are spread across three tabs; **Overview**, **Ranking** and **Chart**. The **Overview** tab details information in relation to initial settings. The **Ranking** and **Chart** tab illustrates the ranked relative order of variable importance. The most important predictor is given a rank of 100.

Figure 9.30: Variable Ranking



For this model it can be seen that the top 5 predictors are: *relationship*, *occupation*, *education*, *age* and *capital-gain*.

9.7 Conclusion

KnowledgeSTUDIO provides an array of features to assess model performance including statistics, tables, reports and graphs.

For **Decision Trees** the process can be divided into three:

- **Statistical Validation**
- **Thorough Validation**
- **Business Validation**

As a result of completing this chapter, users should be able to understand and explain the three elements of **Decision Tree** assessment and how to determine variable importance, as well as being able to use:

- Evaluate and validate **Decision Tree** models using statistics and charts
- Assess the structural integrity of the model to identify segments/rules that do not validate
- Determine predictor importance using the **Variable Importance** node

References

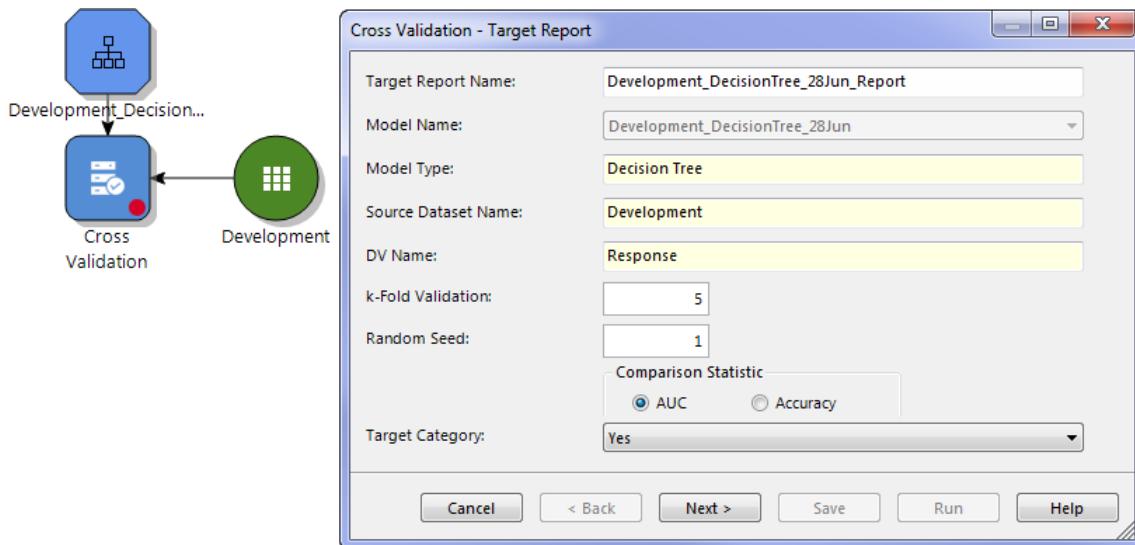
- [1] Tufféry, Stéphane, (2011). **Data Mining and Statistics for Decision Making**. John Wiley & Sons Ltd., United Kingdom.

9.8 Appendix I - Cross Validation

In situations where the dataset is not of a size to adequately support partitioning, cross validation as a means of model assessment should be considered. Cross validation divides a dataset into n partitions or *folds*, where n is any number, for example, 5.

n models are developed, each using $n-1$ partitions when building the model. The model is applied to the partition not included at the model building stage. Figure 9.31 illustrates the process for 1 model.

Figure 9.31: Cross Validation Process



Each partition in turn is used as a validation partition. The remaining partitions used to develop the model. At every iteration the data used to validate the model is not used in its development. Comparison statistics are calculated for each partition to assess model stability.

The process outlined describes a general approach to cross validation. Cross validation in **KnowledgeSTUDIO** differs slightly and requires a developed model and a dataset.

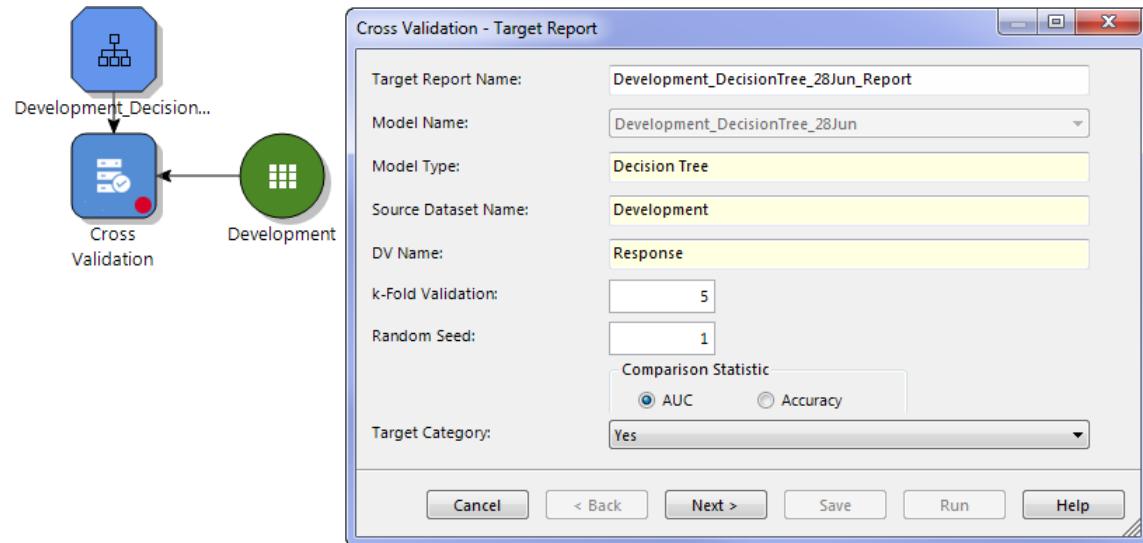
The operation of cross validation in **KnowledgeSTUDIO** is similar to that outlined above however, as a result of requiring a developed model, the model is not created based on $n-1$ partitions.

The model coefficients are updated based on the data in the $n-1$ partitions, once complete the model is then applied to the hold out partition.

This is the process for all model types except **Decision Trees**. With a **Decision Tree**, the model structure is simply applied to each hold out partition in turn.

Figure 9.32 illustrates the use of the **Cross Validation** node applied to a **Decision Tree** model with a discrete **Dependent Variable**, **Response**, with two categories: **Yes** and **No**.

Figure 9.32: Cross Validation



The first dialog of the node is also illustrated. As can be seen some options are yellow, these cannot be modified from within the dialog and are determined by connections and convey the following information:

- **Model Type:** Type of model connected
- **Source Dataset Name:** Name of connected dataset
- **DV Name:** **Dependent Variable** name

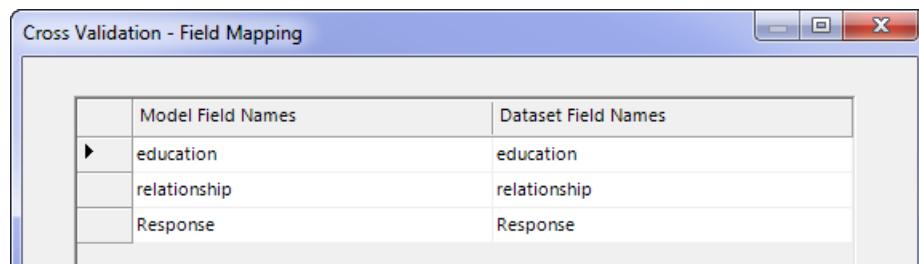
Modifiable options include naming the report created, the number of *folds* and random seed settings.

Comparison statistics are generated for each *fold* and two are available; **AUC**, which requires selection of the **Dependent Variable Target Category:** or **Accuracy**.

For this demonstration, the **AUC** option is selected and the **Target Category:** is set to **Yes**.

Clicking **Next >** moves to the **Cross Validation - Field Mapping** dialog. This is a familiar dialog and allows model fields to be mapped to dataset fields.

Figure 9.33: Cross Validation - Field Mapping



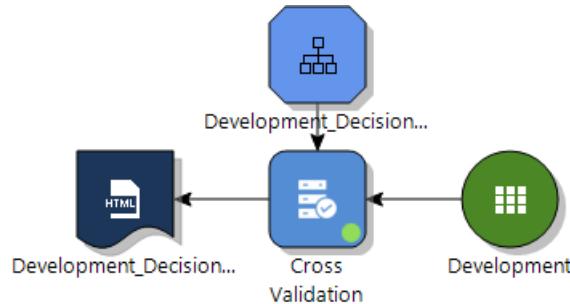
The dialog is titled "Cross Validation - Field Mapping". It contains a table with two columns: "Model Field Names" and "Dataset Field Names". The table rows are:

Model Field Names	Dataset Field Names
education	education
relationship	relationship
Response	Response

The implication of this dialog means that any dataset containing model fields can be used with the cross validation node. Here, the model fields are identified in the data automatically.

Clicking **Run** generates results as a *HTML* report

Figure 9.34: *HTML* Results



Opening the results reveals minimal output.

Figure 9.35: Cross Validation Report



Cross Validation Report	
Input Model Name:	
Development DecisionTree 28Jun	
Input Dataset: Development	
Comparison Statistic: AUC	
Validation fold #	Likelihood function value
1	0.8219
2	0.8313
3	0.8324
4	0.8077
5	0.8280
Max	0.8324
Min	0.8077
Avq	0.8243
© COPYRIGHT 2017 ANGOSS SOFTWARE CORPORATION	
Created: Wednesday, Jun 28, 2017 05:39:15 PM	

As can be seen, a **Likelihood function value** is generated for each partition. Additional values are the minimum, maximum and overall average.

The likelihood function is an assessment of the chosen **Comparison Statistic**, chosen previously. The **Likelihood function value** does not require direct interpretation beyond the understanding, that in general, a lower likelihood value reflects a better model fit.

When assessing model stability using **KnowledgeSTUDIO Cross Validation**, the focus should be on whether the comparison statistics are not extremely different. This is certainly the case here, implying the model validates well.

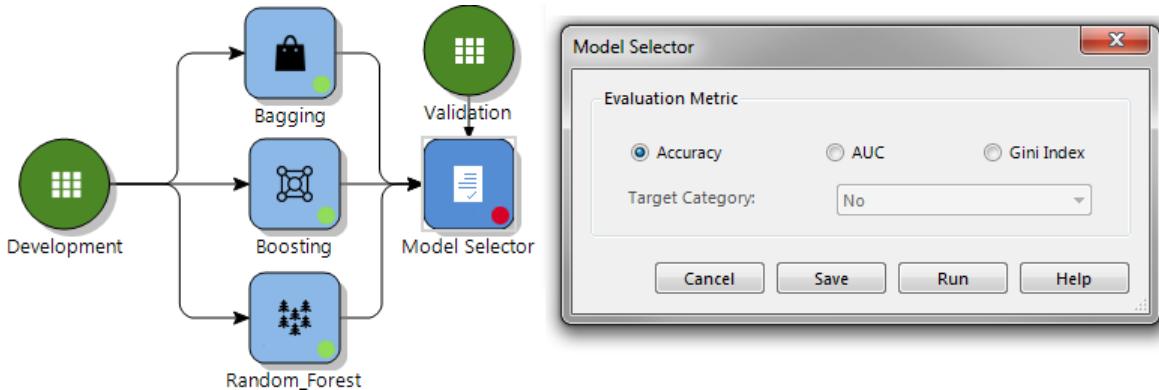
9.9 Appendix II - Picking the Best Model

In some scenarios multiple models may be developed. The question then arises: which one to choose? This can be easily addressed with the **KnowledgeSTUDIO Model Selector** node, located in the **Model** palette.

This node provide the option to compare models on one of three statistics; **Accuracy**, **AUC** or **Gini Index**. The nodes requires at least three connections: two models and a dataset.

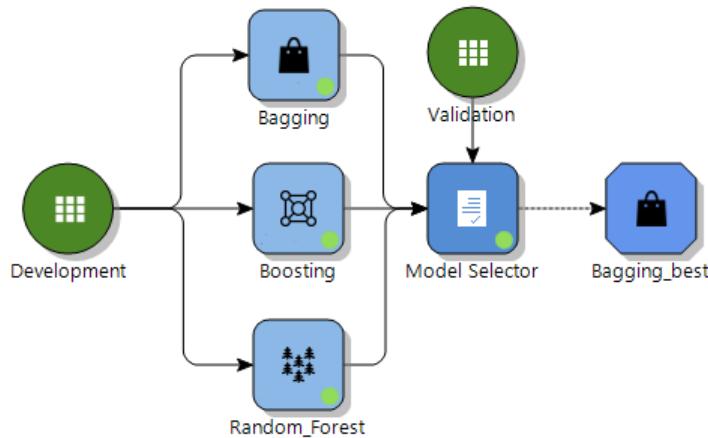
In figure 9.36, three **Decision Tree** models and a dataset are connected to the **model Selector** node. The node options are also shown.

Figure 9.36: Model Selector



As can be seen, the options are minimal. If **AUC** or **Gini Index** is chosen, a target category must be selected. Here, **Accuracy**, the default, is chosen. Clicking **Run** outputs the chosen model as illustrated in figure 9.37.

Figure 9.37: Best Model Selected



Note that the chosen model name is its original name with a suffix of **_best**. Opening the results reveals the chosen model. The **Model Selector** simply duplicates the chosen models output results, with an additional tab **Model Selector**.

This tab illustrates the **Model Selector** chosen statistic for each model, in this case the **Accuracy** value.

Figure 9.38: Model Selector Report

Model name	Accuracy
Bagging	85.40%
Boosting	85.18%
Random_Forest	82.86%

9.10 Appendix III - Model Analyzer with a Continuous Dependent Variable

The **Model Analyzer** can also be used to assess models that have a continuous dependent variable.

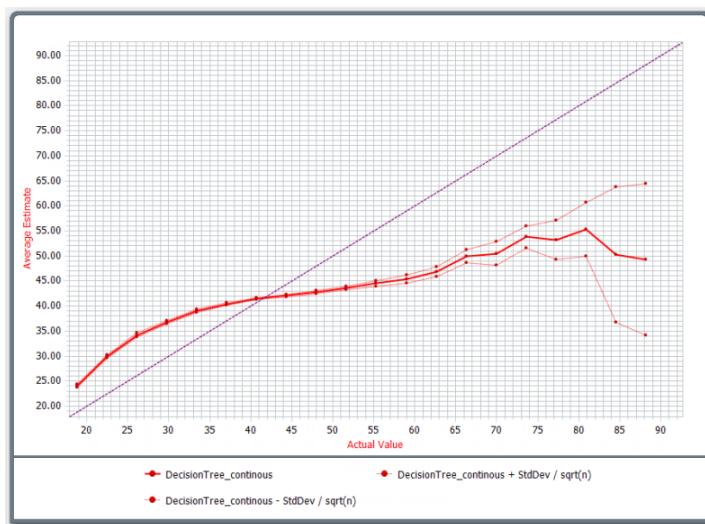
Once appropriate connections are made and options set, again, a set of charts are created but this time appropriate for a continuous dependent variable. The resulting charts are illustrated and explained in the following sections.

9.10.1 Bias Chart

The first chart generated is the **Bias Chart**. The *x-axis* illustrates the range of the dependent variable values. The *y-axis* shows the **Average Estimate** of the dependent variable, and measured in the same units as the dependent variable.

The grid plots estimated values of the dependent variable for a binned range of the actual dependent variable. The graph also shows confidence intervals for each bin as $\pm \frac{StdDev}{\sqrt{n}}$, where n is the number of observations in a given interval.

Figure 9.39: Bias Chart



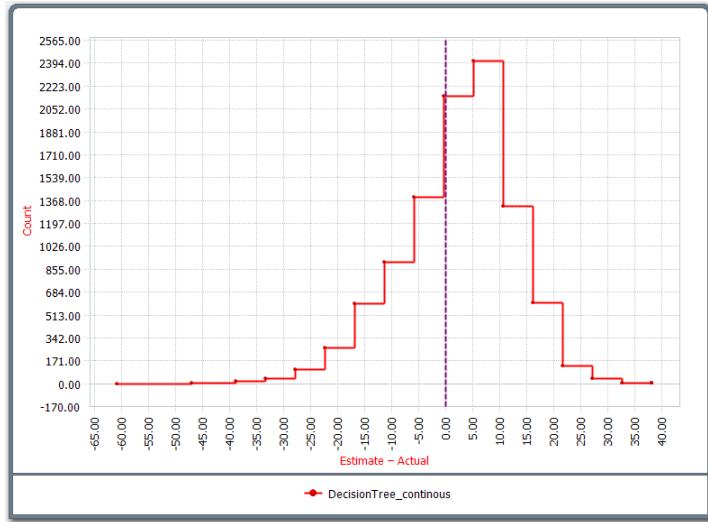
An ideal model, is represented by the diagonal line; $X=Y$. Deviation from the dialog represent the bias in the model.

Here the model values generally deviate from actuals and only correspond at one value: 40.

9.10.2 Accuracy Chart

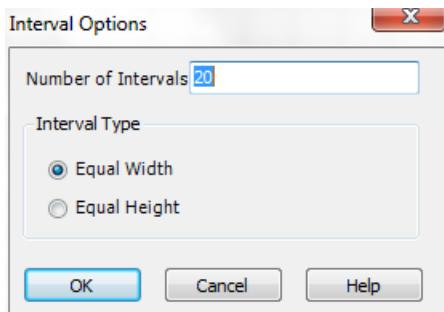
The second tab provides the **Accuracy Chart**.

Figure 9.40: Accuracy Chart



The **Accuracy Chart** provides a histogram of the residuals. Intervals can be adjusted from **Tools... Options**.

Figure 9.41: Interval Options



The x-axis scale ranges from the highest negative difference to the highest positive difference between the estimate and actual dependent variable values.

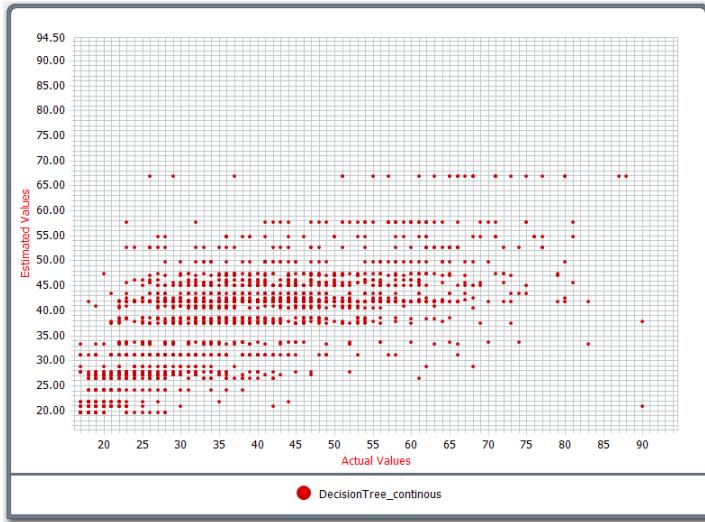
The interpretation of these values is that negative values refer to under predictions and positive values to over predictions. As can be seen the distribution is skewed to the left, meaning the model has a greater tendency to under-predict.

9.10.3 Scatter Plot

The third tab provides a scatter plot of the actual vs. estimates, where each point represents a single observation.

This representation can be used to assess whether there is constant variance of the residuals across the predicted values. An acceptable model would exhibit a constant range either side of the diagonal.

Figure 9.42: Scatter Plot

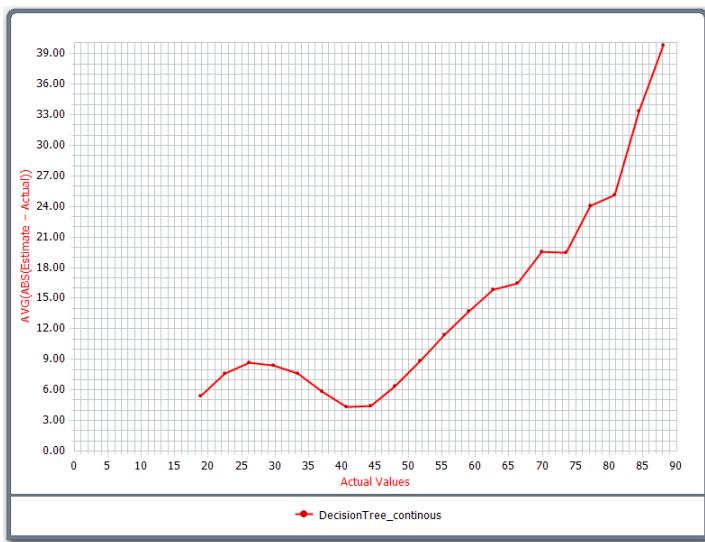


NOTE: If there are more than 2000 records, the data is sampled. Sampling options can be specified from the **Options** dialog found in the **Tools** menu.

9.10.4 Error Chart

Graphs the average absolute difference between the actual values and predictions.

Figure 9.43: Error Chart



For example, for all records with an actual value of 20, the range of predictions is assessed and the average absolute difference between the estimates and actual calculated.

Vertical axis values closer to zero are desirable. Here it can be seen that the higher the actual value of the

dependent variable, the predictions deteriorate, i.e. the predictions get worse.

Predictions are most accurate around the range 40 - 45.

The **Options** button provides access to interval selection (not shown).

Exercises

1. Using the **Automatic Grow** option or a tree, assess model accuracy using **Resubstitution** from the **Tools** menu.
2. Using the **Model Validation** node, generate a validated dataset containing the statistical report and **Confusion Matrix**.
 - (a) What is the classification rate for each of the categories of the dependent variables?
 - (b) Look at the Overview Report and Data tabs. What new variables have been created, and what are they?
3. Evaluate the model by connecting the validated dataset from the previous step to the **Model Analyzer** node. Ensure you select the **Yes** category of the dependent variable, and the **Yes Prob** from the validated dataset.
4. Assess each graph in turn ensuring you understand what it means.
 - (a) What is the purpose of the Cumulative chart and how is it interpreted and used?
 - (b) What is the purpose of the Lift chart and how is it interpreted and used?
 - (c) What is the purpose for the ROC and K-S charts?
5. Using the profit curve, assign some values and assess the outcome.
6. Considering all of these charts, what is the optimum cut-off point in this model?
7. Use the **Tree Validation Report** node to apply the model created with the **Development** partition to the **Validation** partition. Produce a **Validation Report**
8. Use the **Model Analyzer** to assess how well the model performs on the **Validation** partition.
 - (a) Do the curves look similar?
 - (b) Do you think the model validates? Use the shape of the curves and the generated statistics to assess.
9. Use the **Variable Importance** node from the **Evaluate** palette to determine ranked variable importance.

Chapter 10: Model Deployment

10.1 Introduction

Model deployment and exporting results are the final steps in the modelling process. **Deployment** relates to scoring an existing dataset within the project, or creating model code to deploy on an appropriate platform.

Modified or scored datasets may need to be exported to an external location or database and in a specific format, and **KnowledgeSTUDIO** provides an array of export possibilities to address these needs.

The following chapter outlines **KnowledgeSTUDIO** capabilities related to deploying model results and exporting either modified or scored datasets. This chapter details the following topics:

- Score current project datasets using the Scoring node
- Create code in a variety of formats using the code nodes
- Export datasets to a specific location in an available format, or export to a database

10.1.1 Model Deployment

Once a satisfactory model has been developed, evaluated and validated, it is ready for deployment.

Model deployment means applying the model to a population or a new dataset and is commonly referred to as scoring. **KnowledgeSTUDIO** provides two methods for model deployment:

- Directly applying the model
- Generating code for the model

KnowledgeSTUDIO provides an array of nodes for model deployment; one for applying the model to a dataset and ten nodes for generating code in a variety of formats.

Deployment nodes can be applied to any model type and are contained in the **Action** palette:

Table 10.1: Action Palette

Palette	Node	Description
Action	Generate English	Generate English code
	Generate Generic	Generate Generic code
	Generate Java	Generate Java code
	Generate LOS	Generate LOS code
	Generate PMML	Generate PMML code. Note that some advanced properties of trees, such as open-ended intervals in splits on continuous variables, may not be expressed in PMML
	Generate SPSS	Generate SPSS code
	Generate SQL	SQL code
	Generate SQL Function	Creates an SQL function that takes attributes as arguments and returns a score or prediction
	Generate SQL Select	For Decision Trees only: generates rules to select records in a segment
	Generate XML	Altair XML code
	Scoring	Score open or external datasets
	Generate Python	Generate Python code
	Generate R	Generate R code

*Only nodes available with a **KnowledgeSTUDIO** license are described. Other nodes for documentation and DMX code generation may be available but are not covered on this course

NOTE: SQL code generated follows **Microsoft SQL Server** standards. Some adjustments in SQL syntax may be necessary if the code is deployed in other database systems, such as **Oracle**.

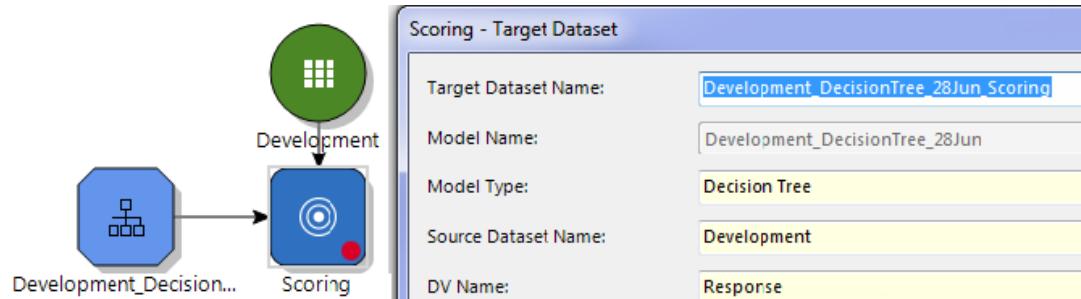
10.1.2 Directly Applying Models

Directly applying the model refers to scoring a current project dataset and can be performed using the **Scoring** node, found on the **Action** palette. The output from the **Scoring** node is a new dataset.

The **Scoring** node requires two connections: the **Model Instance** to apply and the dataset to score. Here the **Validation** dataset is scored using the **Decision Tree Model Instance** created previously.

Figure 10.1 illustrates a partial view of a **Workflow** with the **Scoring** node connected and opened on first dialog; **Scoring - Target Dataset**.

Figure 10.1: Scoring Node Attached



The **Scoring – Target Dataset** dialog provides mostly unmodifiable information in relation to connections made. The only option that can be specified is **Target Dataset Name**, here the default name is accepted. Clicking **Next >** moves to the **Scoring – Field Mapping** dialog.

Figure 10.2: Scoring - Field Mapping

Scoring - Field Mapping		
	Model Field Names	Dataset Field Names
▶	education	education
	relationship	relationship

This dialog has two columns: **Model Field Names**, the fields used when creating the model, and **Dataset Field Names**, the corresponding fields in the dataset being scored to use to generate model scores.

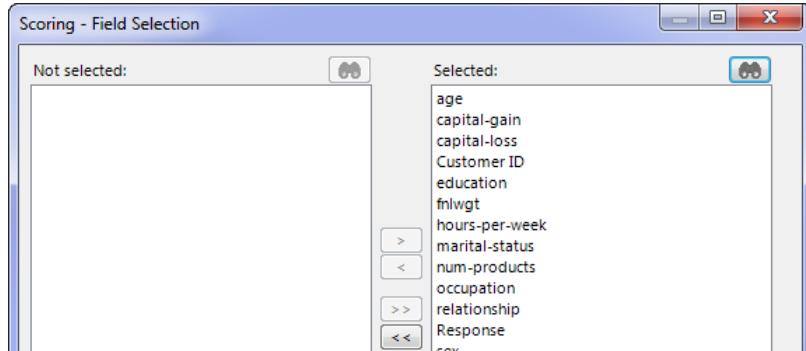
If field names are identical in both, they will be mapped automatically, otherwise they can be mapped by clicking the **Dataset Field Names** column for any field and selecting the appropriate field to map from the dropdown. Click **Next >** to move to the **Scoring – Scoring Fields** dialog.

Figure 10.3: Scoring - Scoring Fields

Item	Field Name	Include	Cut Off
Response Prediction	Response Prediction	<input checked="" type="checkbox"/>	
Response Probability of Prediction	Response Predict Probability	<input checked="" type="checkbox"/>	
Response No Probability	Response No Probability	<input checked="" type="checkbox"/>	0.5
Response Yes Probability	Response Yes Probability	<input checked="" type="checkbox"/>	0.5
Response Node ID	Response Node ID	<input checked="" type="checkbox"/>	
Response Node Number	Response Node Number	<input checked="" type="checkbox"/>	

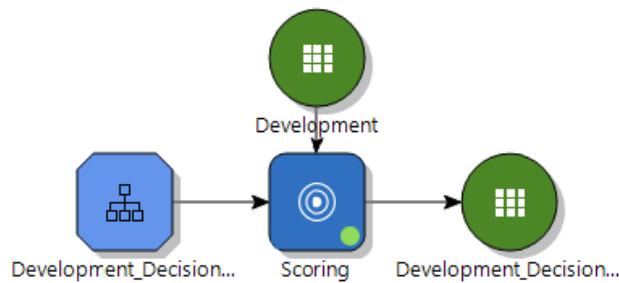
The **Scoring – Scoring Fields** dialog provides options for the new fields created when the dataset is scored. Available options are identical to those present when validating a model. Clicking **Next >** navigates to the final dialog; **Scoring – Field Selection**.

Figure 10.4: Scoring - Field Selection



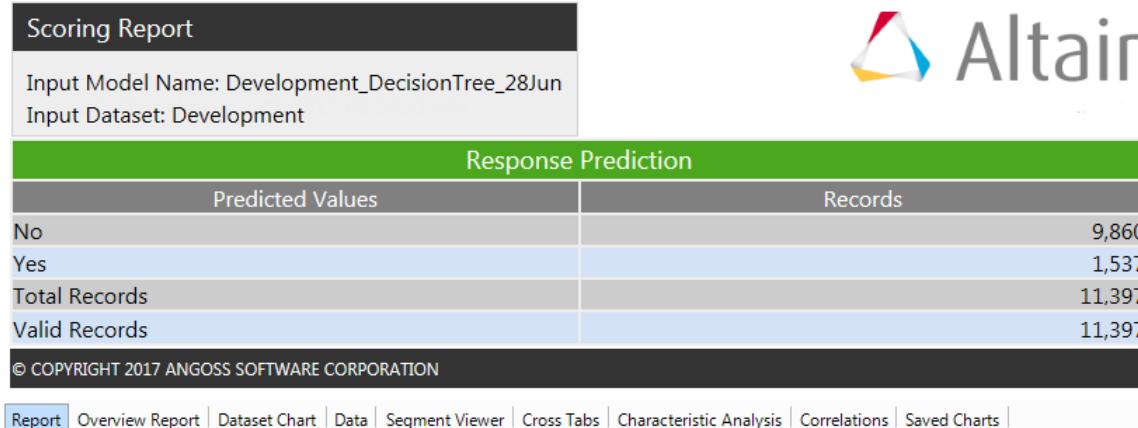
The **Scoring – Field Selection** dialog provides options to select fields from the input dataset to appear in the new dataset alongside those listed in the **Scoring - Scoring Fields** dialog. The default is to include all fields.

Figure 10.5: Scored Dataset



The scored dataset is depicted as a node on the **Workflow** canvas and represents the created dataset on the **Project Pane**. Open results by either double clicking the dataset on the **Project Pane**, or right click the **Workflow** node and select **Open View**.

Figure 10.6: Scored Results



The screenshot shows the 'Scoring Report' interface. At the top, it displays the 'Input Model Name: Development_DecisionTree_28Jun' and 'Input Dataset: Development'. Below this is a table titled 'Response Prediction' with columns 'Predicted Values' and 'Records'. The table shows data for 'No' (9,860 records), 'Yes' (1,537 records), 'Total Records' (11,397), and 'Valid Records' (11,397). At the bottom, it says '© COPYRIGHT 2017 ANGOSS SOFTWARE CORPORATION' and includes a navigation bar with links like 'Report', 'Overview Report', 'Dataset Chart', etc.

The scored dataset contains all the tabs normally available when using the dataset viewer, and one addition: the **Report** tab. This tab relays information in relation to the dataset scored, model used, records scored, and the date the scoring was performed.

The **Scoring Report** also shows the total number of records provided as the scoring input as **Total Records**, and the number of records for which a valid score was produced; **Valid Records**.

The latter may be less than the former in the case where a required field in the scoring dataset has missing values. Usually, for a categorical outcome, the probability of the category of interest is the only field necessary to for any further steps.

Scores are compared to the **Probability Cut Off** value, specified when scoring the data, which determines the predicted value of the dependent variable field; **Response Prediction**, for each record.

If the cut off was defined as 0.5, the default, then any score value greater than or equal to 0.5, results in a prediction of **Yes**. A score value below 0.5 will result in a prediction of **No**.

Figure 10.7: Scored Data Tab

	Response	Response Prediction	Response Predict Probability	Response No Probability	Response Yes P
1	No	No	0.982750582750583	0.982750582750583	0.017249417249
2	Yes	No	0.660297836470919	0.660297836470919	0.339702163529
3	Yes	No	0.660297836470919	0.660297836470919	0.339702163529
4	No	No	0.982750582750583	0.982750582750583	0.017249417249
5	No	No	0.894986449864499	0.894986449864499	0.10501355013
6	No	No	0.946843853820598	0.946843853820598	0.05315614617
7	Yes	Yes	0.83969465648855	0.16030534351145	0.83969465648855

10.1.3 Code Generation

Automatic code generation is available for a variety of formats including *SQL*, *SAS*, *SPSS*, *Java*, *XML* and *PMMI*. Code can subsequently be applied and used as needed.

Model code can be generated using nodes found in the **Action** palette as described previously.

To generate code; drag an appropriate node from the **Action** palette to the **Workflow** canvas, connect a **Model Instance** and open as illustrated in figure 10.8.

Figure 10.8: SQL Code Generation



Options are limited and in general, informative. The code file generated can be assigned a name in the **Target Name** slot. Accept the default name and click **Run** to generate the code.

The resulting code file is generated in the **Project Pane**. Double click to open and view.

Figure 10.9: SQL Code Generated - Partial View

```
-- SQL Predictive Model

--Block # 1: Calculates the probability that '(Response)' equals 'No'
(CASE
WHEN ("relationship" = 'Husband' or "relationship" = 'Wife') THEN
    (CASE
        WHEN ("education" = '10th' or "education" = '11th' or "education" = '12th' or
            "education" = '1st-4th' or "education" = '5th-6th' or "education" = '7th-8th' or "education" =
            '9th' or "education" = 'Assoc-acdm' or "education" = 'Assoc-voc' or "education" = 'HS-grad' or
            "education" = 'Preschool' or "education" = 'Some-college') THEN
            0.6602978364709188
        WHEN "education" = 'Bachelors' THEN
            0.3292181069958848
        WHEN ("education" = 'Doctorate' or "education" = 'Masters') THEN
            0.24423963133640553
        WHEN "education" = 'Prof-school' THEN
            0.16030534351145037
        ELSE
            0.548861852433281
    END)
WHEN "relationship" = 'Not-in-family' THEN
    0.8949864498644986
WHEN ("relationship" = 'Other-relative' or "relationship" = 'Own-child') THEN
    0.9827505827505828
WHEN "relationship" = 'Unmarried' THEN
    0.946843853820598
ELSE
    0.7622181275774327
END)

--Block # 2: Calculates the probability that '(Response)' equals 'Yes'
```

Code Generation

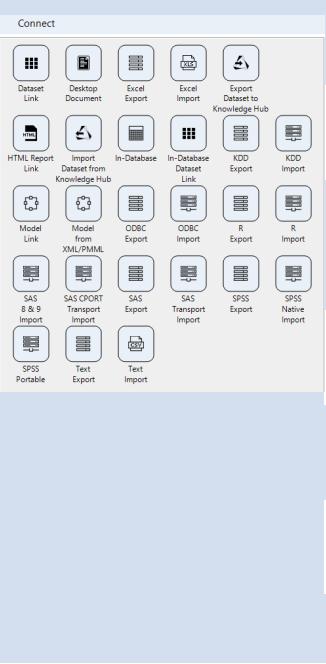
The code can be exported to an *SQL* file and should run error free on appropriate platforms. To export the code, select the file from the **Project Pane** and from the **File** menu select **Save As**. The correct file format *.sql* is automatically selected. *LOS* code can be generated and exported using a similar process.

10.2 Exporting Results

Creating a dataset with scoring fields or modifying a dataset may require that results are exported to a specific format or location. **KnowledgeSTUDIO** provides an array of export capabilities from excel, text and *LOS* file export to communicating with a database and creating new tables.

The **Connect** palette provides seven export nodes. The nodes are listed and detailed in table 10.2.

Table 10.2: Source Palette

Palette	Node	Description
	Export Dataset to Knowledge Hub	Export to Altair file form for Knowledge Hub
	Excel Export	Export to Excel file format
	ODBC Export	Connect and export to all database systems accessible via ODBC file format
	R Export	Export to R file format
	SAS Export	Export to SAS file format
	SPSS Export	Export to SPSS file format
	Text Export	Export to Text file format

To export data, add the appropriate node to the **Workflow** canvas and connect a data source. Options are presented, set and data is exported.

Exporting to **Excel**, **R**, **SAS**, **SPSS**, and **Altair** file format is straightforward. You only need to specify the output file name and the fields to be included.

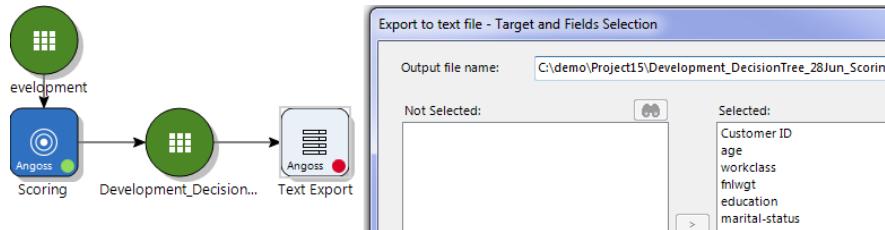
Additional options are presented when exporting to **Text** file format such as the column delimiter. **ODBC** export requires specification of the database type to connect to and the appropriate driver being installed, among other aspects.

The process in general is straightforward and an example is given below to export to text.

10.3 Text File Export

Add a **Text** node from the **Source** palette to the **Workflow** canvas and connect the dataset scored using the **Decision Tree Model Instance** as depicted in figure 10.10.

Figure 10.10: Text File Export

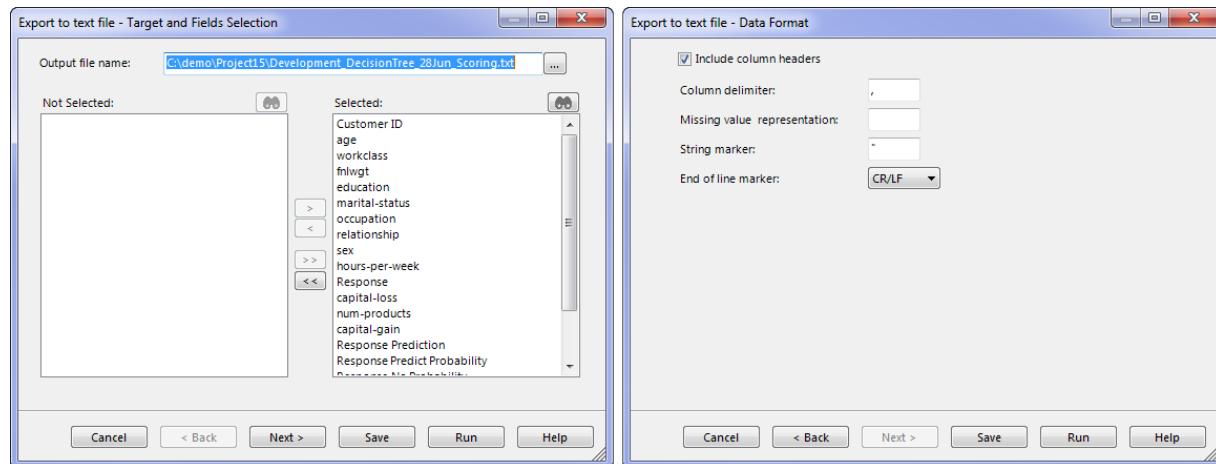


Access the **Text Export** options. Two dialogs are available to step through to set options.

The first, **Export to text file – Target and Field Selection**, provides options to specify the location and name of the text file to export and the fields to include. The second, **Export to text file – Data Format**, provides options to:

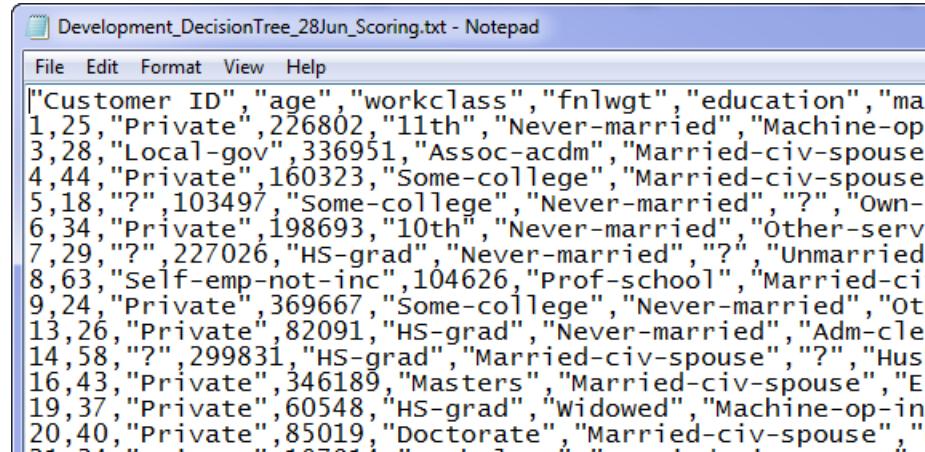
- Include field names as column headers
- Specify delimiter
- Determine how missing values are represented
- Specify the string wrapper
- Specify the end of line marker

Figure 10.11: Text Export Dialog



Accepting the defaults and running results in a text file being exported. Results can be opened with any program capable of reading text. Figure illustration shows the file opened with **Notepad**.

Figure 10.12: Notepad Results



```
Development_DecisionTree_28Jun_Scoring.txt - Notepad
File Edit Format View Help
["Customer ID", "age", "workclass", "fnlwgt", "education", "ma
1,25, "Private", 226802, "11th", "Never-married", "Machine-op
3,28, "Local-gov", 336951, "Assoc-acdm", "Married-civ-spouse
4,44, "Private", 160323, "Some-college", "Married-civ-spouse
5,18, "?", 103497, "Some-college", "Never-married", "?", "Own-
6,34, "Private", 198693, "10th", "Never-married", "?", "Other-serv
7,29, "?", 227026, "HS-grad", "Never-married", "?", "Unmarried
8,63, "Self-emp-not-inc", 104626, "Prof-school", "Married-ci
9,24, "Private", 369667, "Some-college", "Never-married", "Otl
13,26, "Private", 82091, "HS-grad", "Never-married", "Adm-cle
14,58, "?", 299831, "HS-grad", "Married-civ-spouse", "?", "Husl
16,43, "Private", 346189, "Masters", "Married-civ-spouse", "E
19,37, "Private", 60548, "HS-grad", "Widowed", "Machine-op-in
20,40, "Private", 85019, "Doctorate", "Married-civ-spouse", "I
```

10.4 Conclusion

KnowledgeSTUDIO provides ample capabilities to deploy model results by scoring a project dataset or creating model code in a variety of formats for use on other platforms using the **Scoring** or code nodes available from the **Action** palette.

Additionally, if there is a need to export results to a specific file format or database, the **Source** palette provides an array of suitable export nodes.

As a result of completing this chapter, users should be able to:

- Score current project datasets using the Scoring node
- Create code in a variety of formats using the code nodes
- Export datasets to a specific location in an available format, or export to a database

Exercises

1. Create a **Decision Tree Model Instance** and use it to score and create a new dataset
 - (a) Which new fields are created by **Scoring**? Use the **Overview Report** to explore
 - (b) What is the range of predicted probabilities that were generated? Use a **Dataset Chart** to explore
 - (c) Export the scored dataset to .xlsx format
2. Choose a code node, and export code for the model. Try a few different formats

Chapter 11: Introduction to Strategy Trees

11.1 Introduction

KnowledgeSTUDIO Strategy Trees are a unique feature enabling additional node calculations to be added to model results to better determine how best to treat a group of records.

Additional calculations can be simple or complex calculations but are usually business critical **KPIs**, and are used as a means to augment model scores to more intelligently assign outcomes to records.

Leveraging models traditionally is a matter of:

- Scoring data, generally a large database
- Adding additional measures, such as **KPIs**
- Using external products, such as excel, to slice and dice results to assess how best to proceed and treat records

KnowledgeSTUDIO provides the facility to augment model scores with additional measures and assign treatments using a process feature called a **Strategy Tree**.

The objectives of this chapters are:

- Understand and build **Strategy Trees** from an existing model or on a dataset
- Add additional node calculations
- Assign **Treatments**
- Generate and evaluate assigned **Treatments** using reports
- Modify treatments given limitations on number of assignable treatments
- Include additional dependent variables

11.2 Strategy Trees

Strategy Trees can be built on a dataset or an existing **Decision Tree** model and although the **Strategy Tree** includes and extends **Decision Tree** functionality, its raison d'etre differs.

Decision Trees are designed to segment data into identifiable and distinguished groups. A **Strategy Tree** takes these groups, adds additional measures and applies a **Treatment**.

Treatments are the actions taken based on segment characteristics.

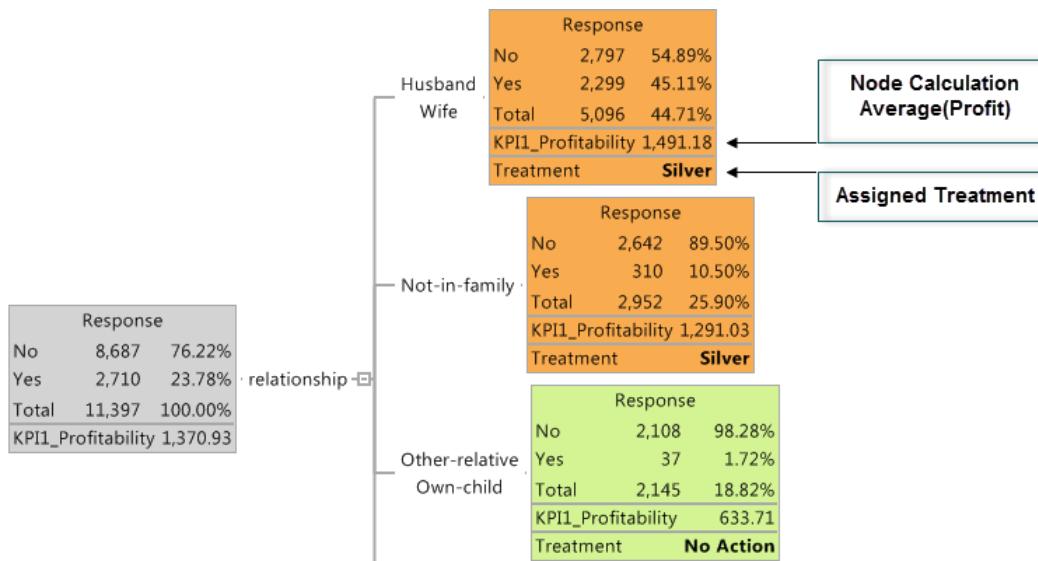
A **Strategy Tree** builds upon a **Decision Tree** by:

- Including additional measures to complement the propensity scores
 - A **Decision Tree** has a single segmented dependent variable
 - A **Strategy Tree** allows additional calculations to compliment the dependent variable distribution

- Prescribing a treatment
 - A **Strategy Tree** aims to assign actions or **Treatments**
 - A **Strategy Tree** explicitly assigns a business action, or treatment, to each node.
- Interactively determining the dependent variable
 - A **Decision Tree** has a single dependent variable that is predicted at every split
 - The dependent variable in a **Strategy Tree** can vary at each node
- **Strategy Trees** have the ability to combine scores from multiple models as well as business rules
 - By contrast, a predictive model can only generate a single score per model

Strategy Trees are initiated using the **Strategy Tree** node found in the **Model** palette.

Figure 11.1: Strategy Tree Example



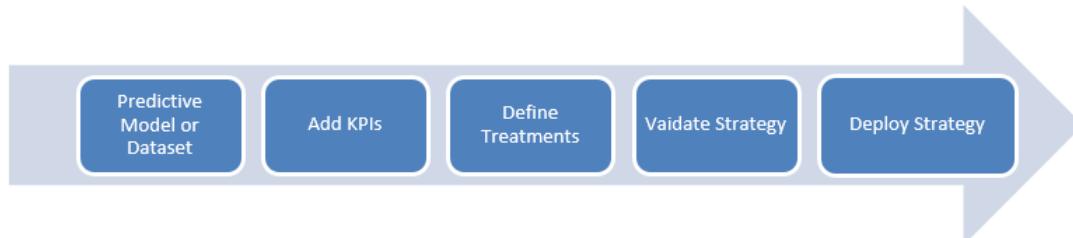
The process of building a **Strategy Tree** involves:

- A **Model Instance**, i.e. **Decision Tree**, or a dataset
- Defining independent node calculations, the **Key Performance Indicators: KPIs**
- Applying treatments to **Strategy Tree** nodes
- Deploying the **Strategy Tree**
- Validating the strategy with independent data, or data gathered over time once the strategy is deployed

NOTE: The final two points can be performed in either order. For some it is important to validate a strategy on a validation dataset prior to deployment, and for others the strategy is deployed, and after a period of time, the performance of the strategy assessed.

Either approach is valid. In addition to this, a strategy can also be assessed on a validation dataset, deployed, and then monitored over time!

Figure 11.2: Business Process Flow



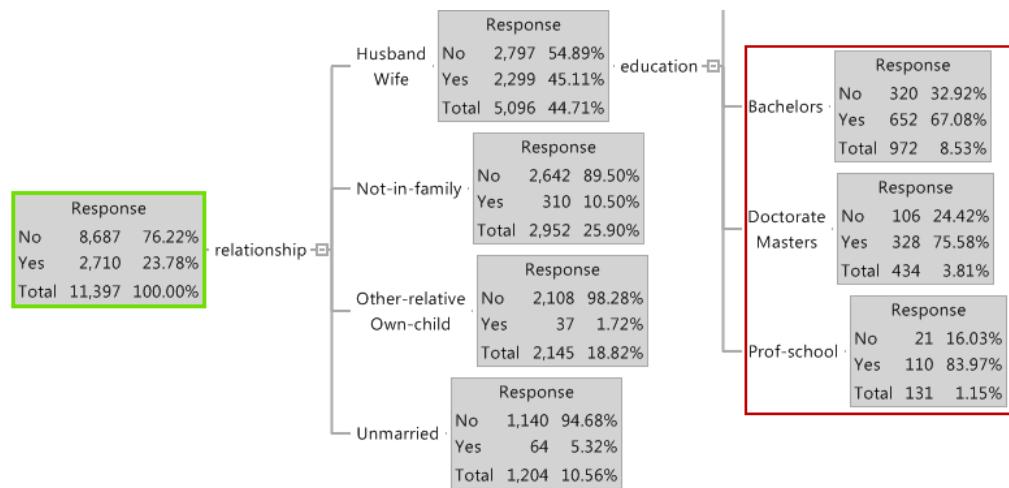
11.2.1 Adding Calculations to a Strategy Tree

Calculations are aggregated values of one or more variables based on the cases in a node. Using calculations within a **Strategy Tree** is a powerful way to determine appropriate treatments for specific segments of the data.

Rather than simply using probabilities to determine appropriate treatments, treatments can be assigned by referring to **Key Performance Indicators (KPIs)** to augment the probability scores.

Consider the scenario depicted in figure 11.3.

Figure 11.3: Guiding Treatment



In a traditional scenario, probabilities are used to determine the appropriate course of action.

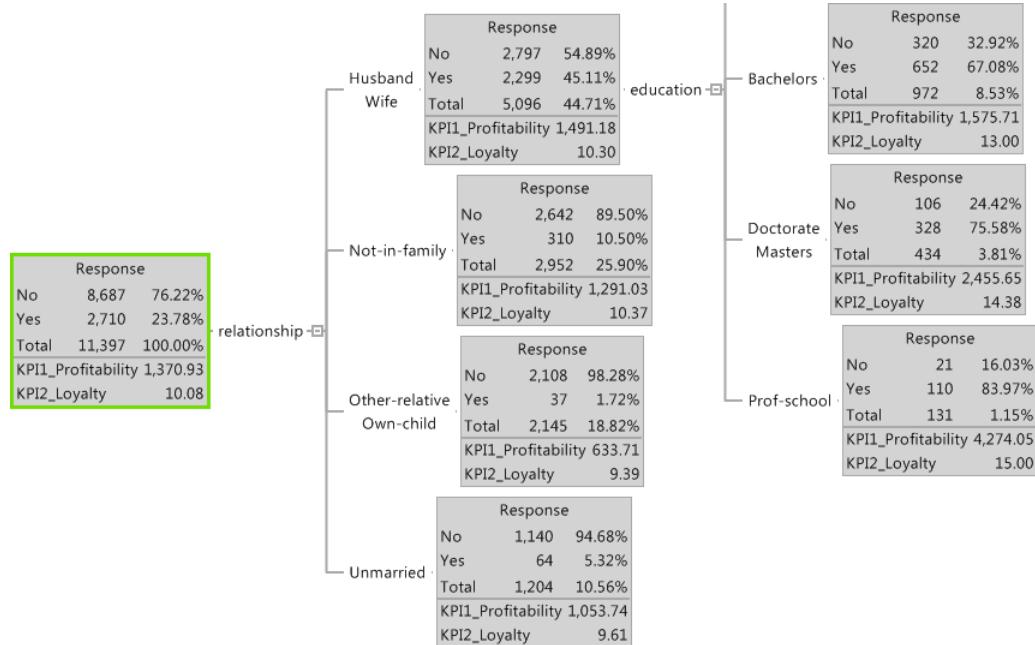
The highlighted nodes in the **Decision Tree** depicted above have relatively high rates for the Yes category, i.e. these records are more likely to respond.

If probabilities alone are used to determine a course of action then the same **Treatment** may be applied to all.

KnowledgeSTUDIO provides the capability to incorporate additional measures that can be used together with the probability to more intelligently assign treatments.

For example, two measures that reflect the **Profitability** and **Loyalty** are added to the **Root** node, cascade to all other nodes. This is illustrated in figure 11.4.

Figure 11.4: KPIs Added



Notice that **KPI1_Profitability** and **KPI2_Loyalty** differ across the nodes.

This would certainly have an effect on how these segments are treated and puts into perspective the value of incorporating additional calculations into a **Strategy Tree** to guide **Treatment**.

Calculations can be added to the **Strategy Tree** directly from the **Strategy Tree** view by selecting the **Tree Calculations** icon  from the **Task Bar**, or select **Calculations** from the **Tools** menu.

NOTE: To access option for calculations, the **Strategy Tree** view must be open, also depending on whether calculations are added from the **Strategy Tree** dialog or from the **Task Bar/Tools** menu, the **Tree Calculations** dialog will differ slightly in terms of look, however functionality is identical.

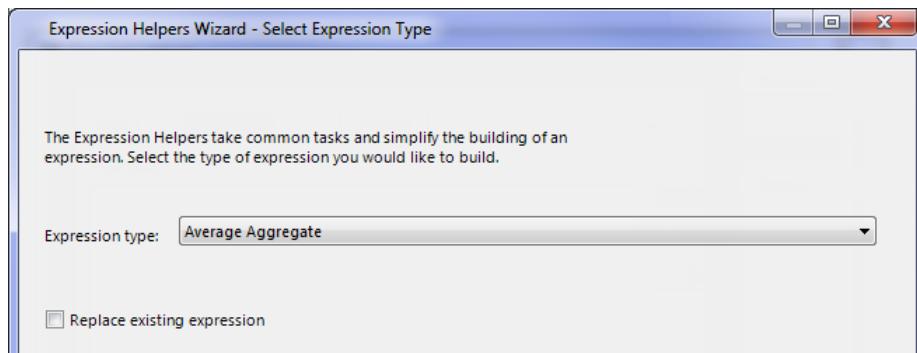
Click the **Add** button to add a calculation.

Figure 11.5: Defining Calculations



The familiar **Expression Editor** opens providing the facility to add calculations manually or via a **Helper** function. Here a **Helper** is selected to generate a calculation.

Figure 11.6: Helpers Wizard for Strategy Trees

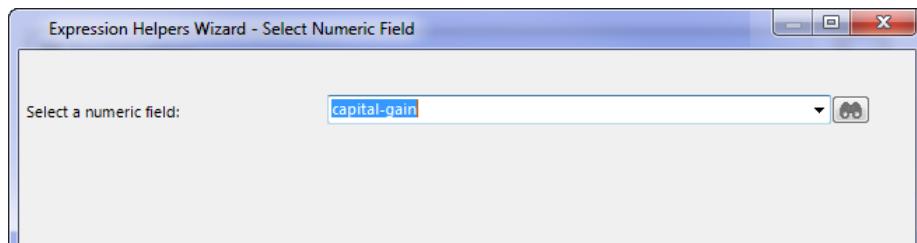


The **Average Aggregate Helper** is a commonly used function to summarize continuous variable values and here is used to add two new calculations:

- KPI1-Profit, based on the variable capital-gain
- KPI2-Loyaty, based on the variable num-products

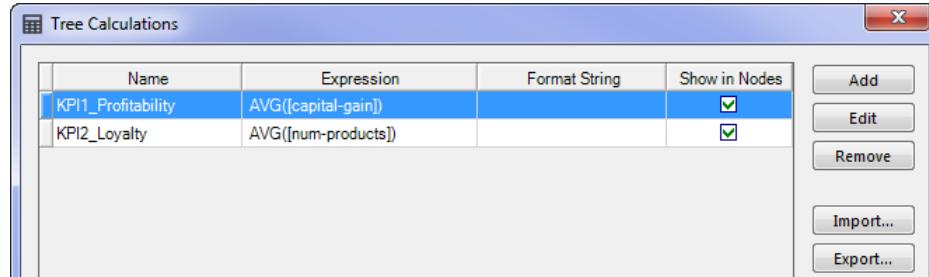
Select the **Average Aggregate** function and click **Next** to open the **Expressions Helpers Wizard – Select Numeric Field** screen. Select *capital-gain* from the Select a numeric field dropdown.

Figure 11.7: Expressions Helpers Wizard – Select Numeric Field



Click **Finish** and then **OK**. The calculation is added to the **Tree Calculations** dialog. Repeat for **KPI2**.

Figure 11.8: Added Calculations



Once both **KPIs** have been added, click **OK** to add the calculations to the **Strategy Tree** and proceed to inspect each node.

Treatments can be assigned by referring to the percentage of the category of interest for the dependent variable, and/or the additional calculations.

Note that additional **KPIs** can be added or edited at any point using the **Tree Calculations** dialog via **Tools...**

Calculations, or by selecting the **Tree Calculations** icon from the **Task Bar** .

11.3 Treatments

A **Treatment** is an action, output, or goal associated with the terminal node of a **Strategy Tree**. Once a **Strategy Tree** has been generated, **Treatments** can be created and assigned.

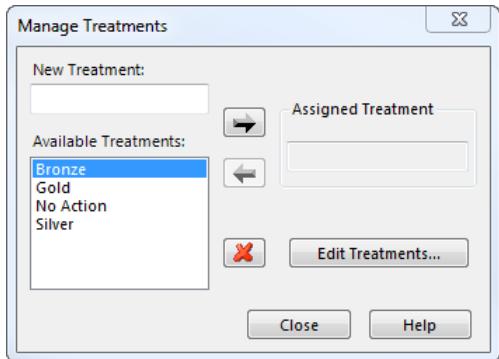
Only one **Treatment** can be associated to a given terminal node. **Treatments** can be assigned manually or assigned automatically based on one or several conditions.

11.3.1 Managing Treatments

Treatments are created, assigned and managed using the **Manage Treatments** dialog found in the **Tools** menu or by clicking the  icon in **Task bar**.

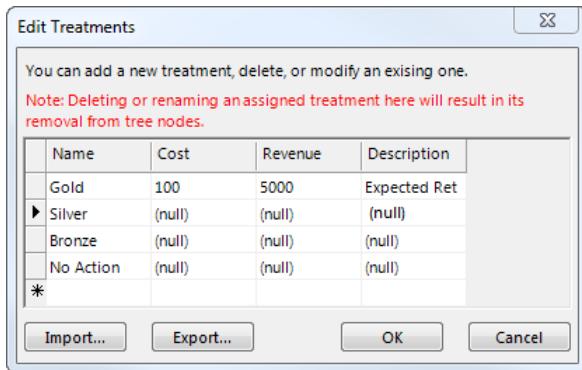
To assign a **Treatment**, right-click a terminal node in the **Strategy Tree** and select the option **Treatments**. This opens the **Manage Treatments** dialog. Create a new treatment by typing a name for the treatment and click the right arrow button to assign to the selected node.

Figure 11.9: Manage Treatments



Treatments can be deleted with the  button. **Treatments** can also be edited via **Edit Treatments** button from the **Manage Treatments** dialog. Additionally, **Cost** and **Revenue** can be associated with an individual **Treatment**.

Figure 11.10: Editing Treatments and Assigning Cost and Revenue



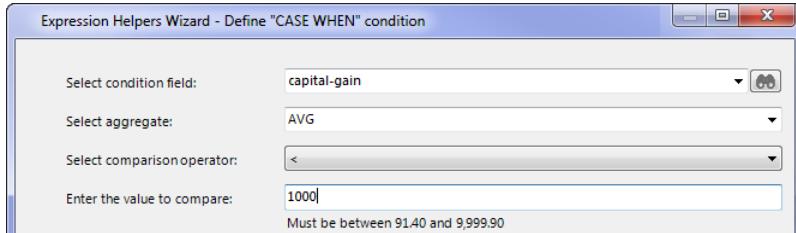
11.4 Treatments from Calculation

Manually assigning treatments based on dependent variable proportions and multiple **KPIs** may be time consuming if the tree is large. **Treatments** can be assigned from a set of rules defined by a calculation in the tree.

To begin, open the **Tree Calculations** dialog, click **Add** then **Helpers**. From the **Helpers** menu select **Computed Conditional Treatment**. Required fields for setting conditions are:

- Field for condition: choose variable with which to define the condition
- Aggregate for field: choose how to aggregate the variable. Within the **Helper AVG** and **SUM** are available for continuous variables
- Select comparison operator: choose between less than, less than or equal to, greater than, or greater than or equal to
- Enter the value for comparison: select the value for which the operator will apply

Figure 11.11: Defining Conditions for Treatments



Click **Next** to give a name to the **Treatment** defined. Add additional treatment conditions using checkbox **Add**. Click **Finish** when all **Treatments** have been defined.

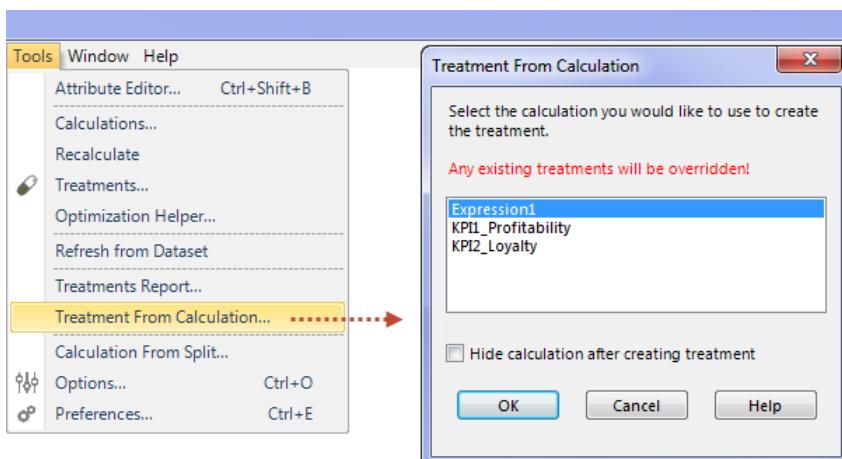
Figure 11.12: Name Treatment



NOTE: SQL code will be generated upon clicking **Finish**. You can manually edit this code to build more complex conditions for treatments.

The calculation is added as an expression to each segment, not shown. Use the **Treatments from Calculations** dialog from the **Tools** menu in **Strategy Tree** view, to apply as a treatment to tree segments. Note that the expression can be hidden by selecting the **Hide calculation after creating treatment**, tick box.

Figure 11.13: Treatment from Calculation Option



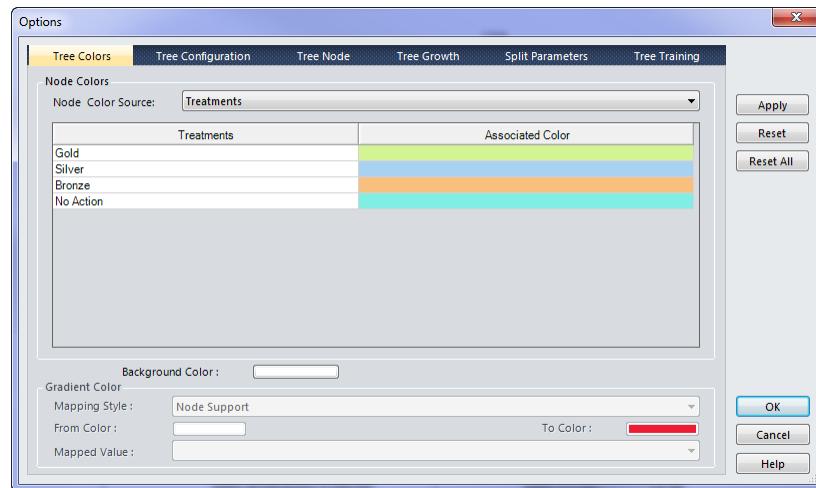
11.4.1 Colour Coding Treatments

Treatments can be assigned colours to improve visualization and clarity. This is accomplished by modifying **Treatment** properties from the **Options** dialog.

Once activated, click the **Tree Colours** tab. Select **Treatments** from the **Node Colour Source** dropdown.

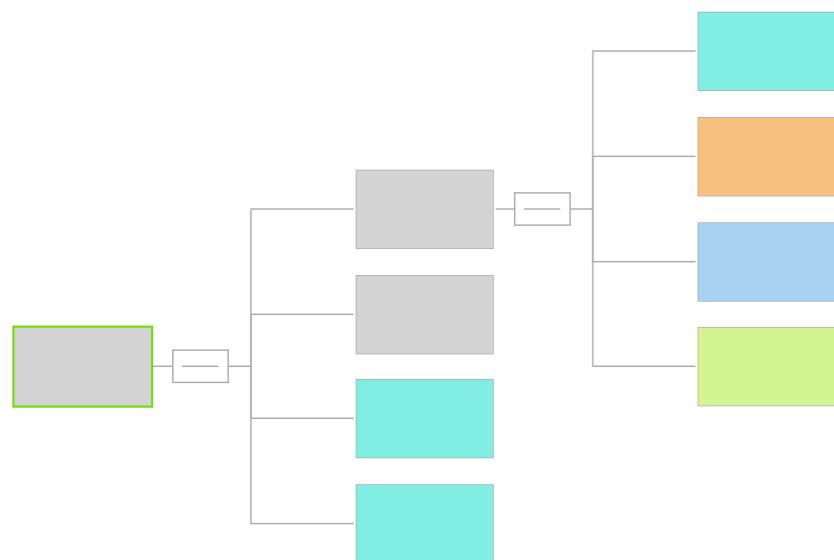
Each **Treatment** is assigned a colour. Here, treatments are kept simple for illustrative purpose and listed in the table below:

Figure 11.14: Assign Colours



The tree map accessed from the **Tree Map** tab also adopts the associated colours and can be used as a means to navigate the **Strategy Tree**.

Figure 11.15: Colour Coded Strategy Tree Map



11.4.2 Treatments Report

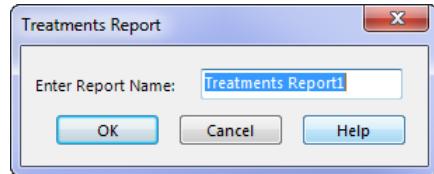
Once treatments have been assigned, a **Treatments Report** can be created. The **Treatments Report** is a tabular representation of the number of treatments assigned.

The **Treatment** report contains:

- The **Treatment** applied
- Values of all node calculations
- **Treatment** attributes from the Edit Treatments dialog; cost and revenue, if defined

To generate a treatment report, from the **Tools** menu select **Treatments Report**

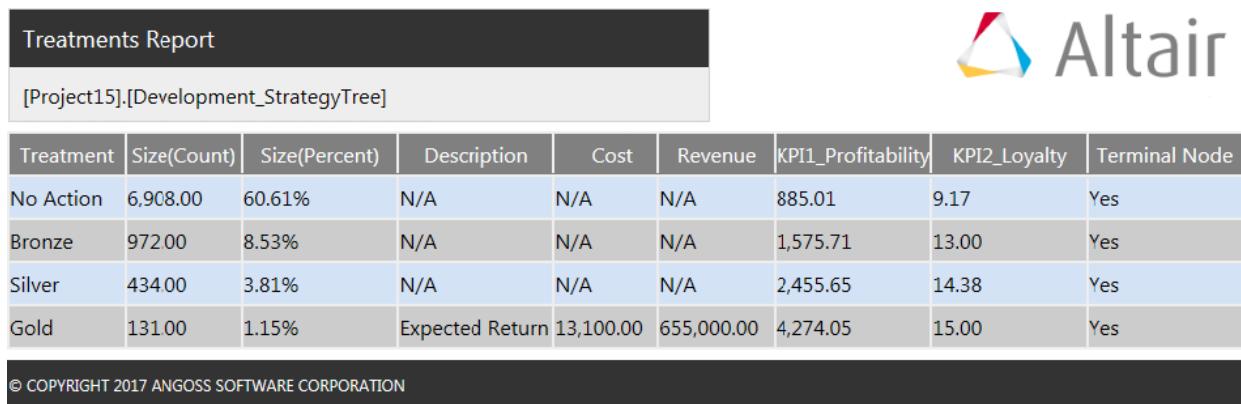
Figure 11.16: Treatments Report Wizard



The **Treatments Report** produces summaries of **Treatments** and **KPIs**. It is a means to measure the effectiveness of the strategy put in place and can address business questions such as:

- What's the bottom line when it comes to this strategy?
- Will the strategy meet expectations?
- In addition, the report can provide a summary of the cost and revenue associated with each treatment if specified when defining **Treatments**
- The report can be copied **Excel**, **PowerPoint**, and **Word**, and also exported as a **PDF** document

Figure 11.17: Treatments Report



Treatment	Size(Count)	Size(Percent)	Description	Cost	Revenue	KPI1_Profitability	KPI2_Loyalty	Terminal Node
No Action	6,908.00	60.61%	N/A	N/A	N/A	885.01	9.17	Yes
Bronze	972.00	8.53%	N/A	N/A	N/A	1,575.71	13.00	Yes
Silver	434.00	3.81%	N/A	N/A	N/A	2,455.65	14.38	Yes
Gold	131.00	1.15%	Expected Return	13,100.00	655,000.00	4,274.05	15.00	Yes

Based on the **Treatment Report** additional actions may be taken, for example, if the number of a specific treatment is limited, say only 350 **Gold Treatments** are available.

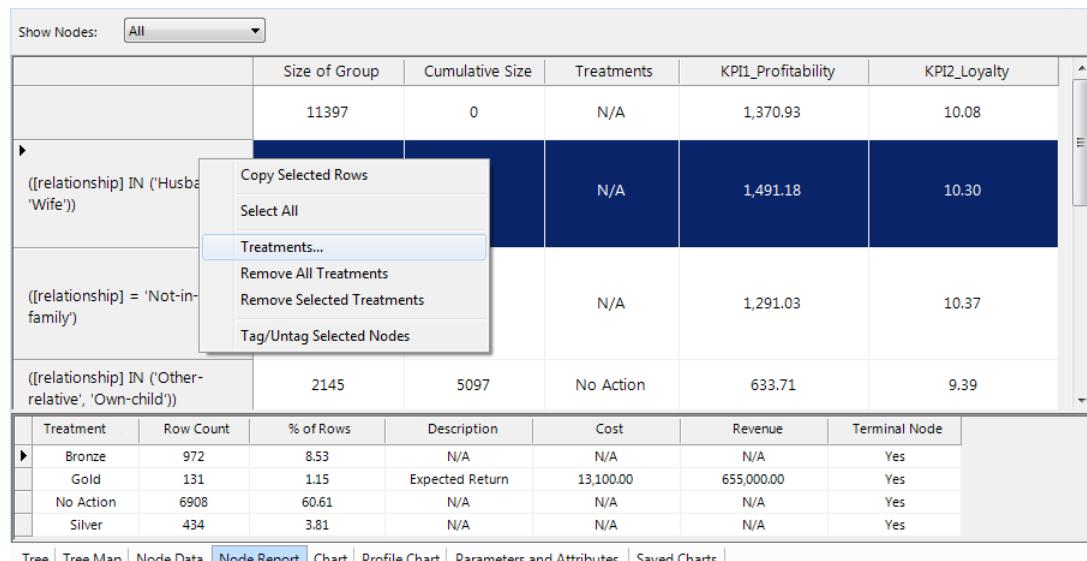
Since the **Strategy Tree** contains the same modification capabilities as a **Decision Tree**, the tree can be further segmented to ensure thresholds are respected.

11.4.3 Node Report

The **Node Report** summarizes nodes based on **Treatments & KPIs**. It also reports node size and cumulative size. **Treatments** are generally assigned from a **Strategy Tree**. The **Node Report** contains added functionality that enables **Treatments** to be viewed and assigned from the **Node Report**.

Right clicking on any node row and choosing **Treatments** opens the **Manage Treatments** dialog where **Treatments** can be assigned.

Figure 11.18: Node Report Tab



Node Report						
Show Nodes:		Size of Group	Cumulative Size	Treatments	KPI1_Profitability	KPI2_Loyalty
	All	11397	0	N/A	1,370.93	10.08
▶	([relationship] IN ('Husband/Wife'))			N/A	1,491.18	10.30
	([relationship] = 'Not-in-family')			N/A	1,291.03	10.37
	([relationship] IN ('Other-relative', 'Own-child'))	2145	5097	No Action	633.71	9.39
Treatment						
▶	Bronze	972	8.53	N/A	N/A	Yes
	Gold	131	1.15	Expected Return	13,100.00	Yes
	No Action	6908	60.61	N/A	N/A	Yes
	Silver	434	3.81	N/A	N/A	Yes

11.5 Demonstrations

The demonstrations will focus on two possible scenarios:

- Building a **Strategy Tree** based on a **Decision Tree** model to leverage all functionalities
- Building a **Strategy Tree** based on a dataset and defining the resulting strategy by combining different objectives in different segments

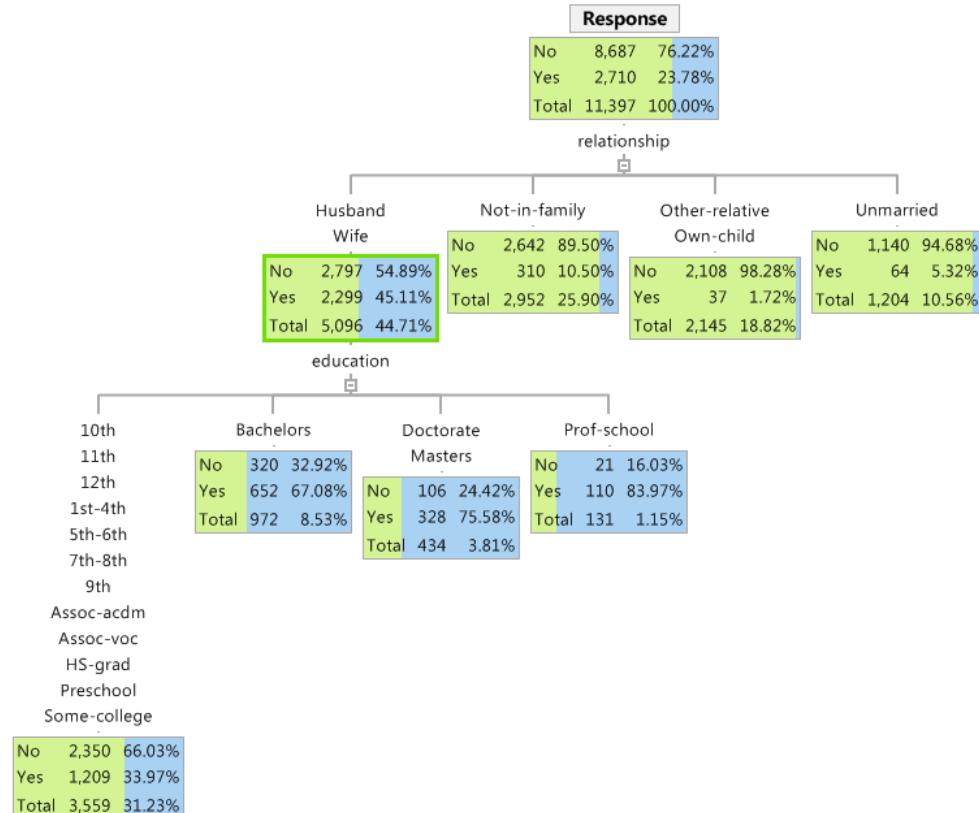
11.5.1 Building a Strategy Tree based on a Decision Tree

The following demonstration uses the file *Census.xlsx*. To follow the demonstration:

- Create a new project
- Import the file *Census.xlsx* different segments

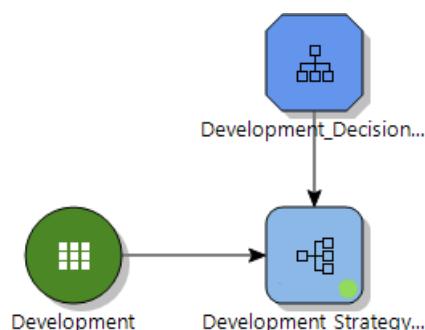
- Partition the data 70/30
- Add a **Decision Tree** node to the larger partition
- Use **Response** as the **Dependent Variable** with **Find Split** and **Edit Split** to re-create figure 11.19

Figure 11.19: Decision Tree



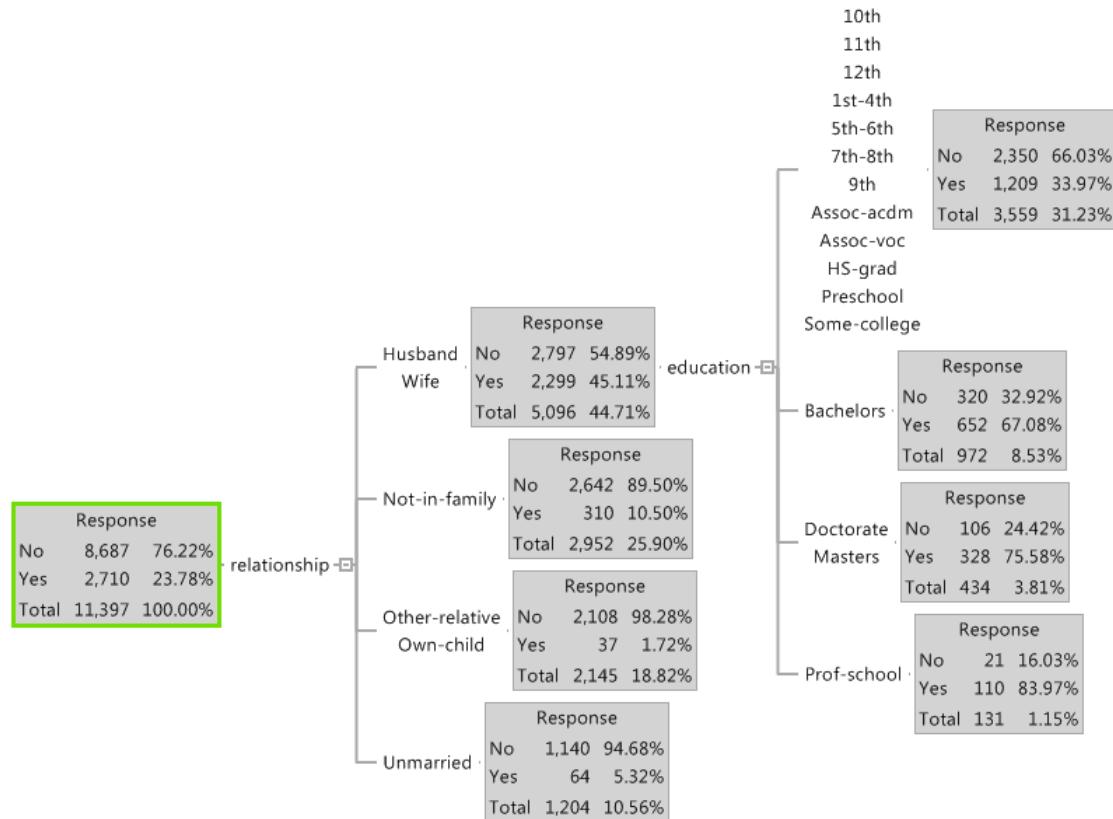
Once complete, generate a **Model Instance**. Drag the **Strategy Tree** node from the **Model** palette and connect the **Model Instance** and the dataset used in development of the model, to the **Strategy Tree** node as illustrated in figure 11.20.

Figure 11.20: Workflow with Strategy Tree



Either double click the **Strategy Tree** node or right click and select **Modify**. As calculations can be defined once the tree has been created, skip this step and click **Run** to generate initial results, and open to view.

Figure 11.21: Strategy Tree from Decision Tree



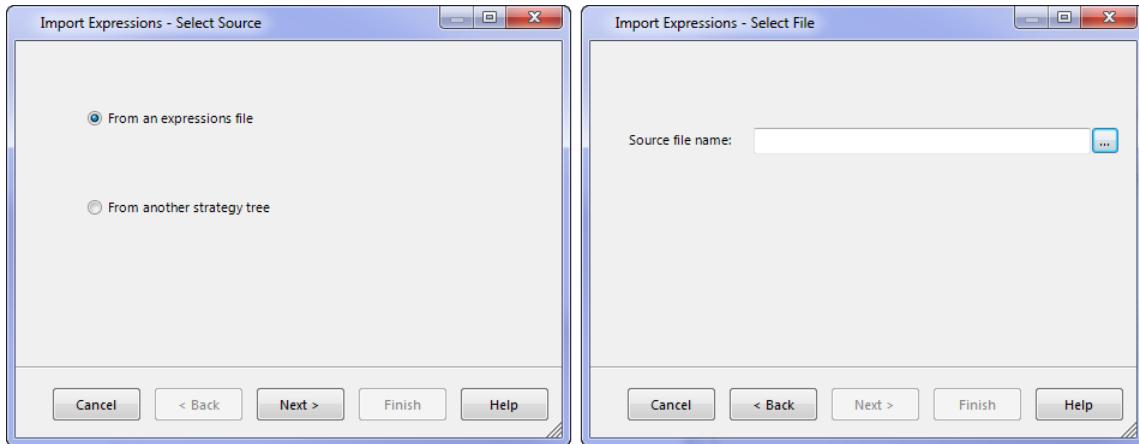
The **Strategy Tree** duplicates the **Decision Tree**. To add calculations, click the **Tree Calculations** icon in the Task Bar.

Notice the **Import...** and **Export...** buttons on the **Tree Calculations** dialog.

Import calculations from an **XML** expression file or other project dataset or **Strategy Tree**. **Export** option allows exporting of calculations to **Altair Expression Format (XML)** or **Plain Text**.

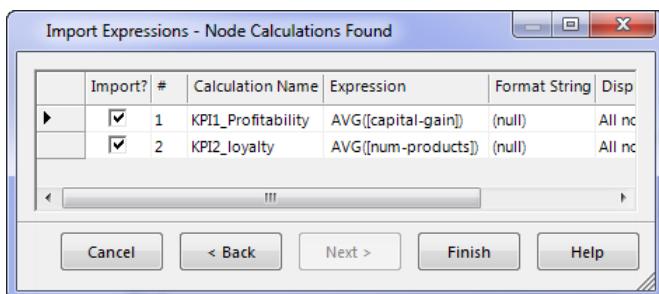
- Click **Import...**
- Choose to import From an expression file in the **Import Expressions – Select Source** dialog
- Locate the calculations file: *KPIs for Marketing.xml*
- Click **Finish**

Figure 11.22: Import Screens



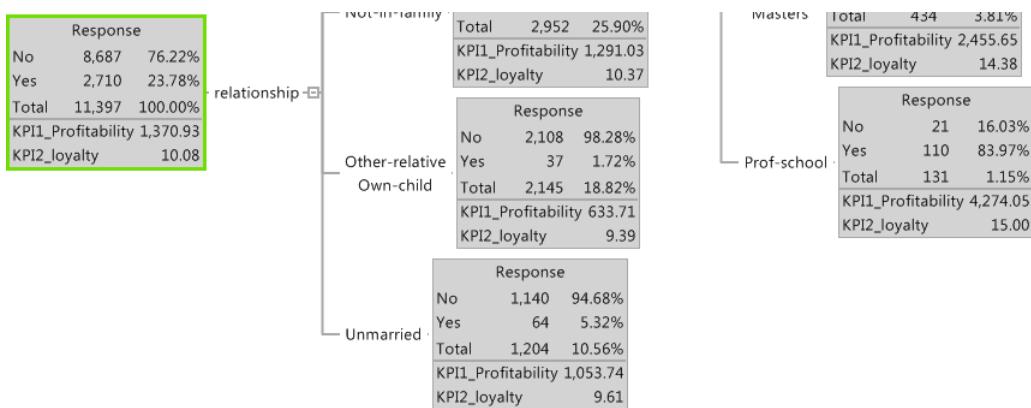
The calculations contained in the located file are displayed on the **Import Expressions – Node Calculations Found** dialog. An **Import?** column allows inclusion of individual **KPIs** by ensuring a tick exists beside those to import.

Figure 11.23: Import Expressions – Node Calculations Found



Click **Finish** and then **OK** to add selected calculations to the **Strategy Tree** as shown in figure 11.24.

Figure 11.24: Strategy Tree with Added Calculations; Partial View

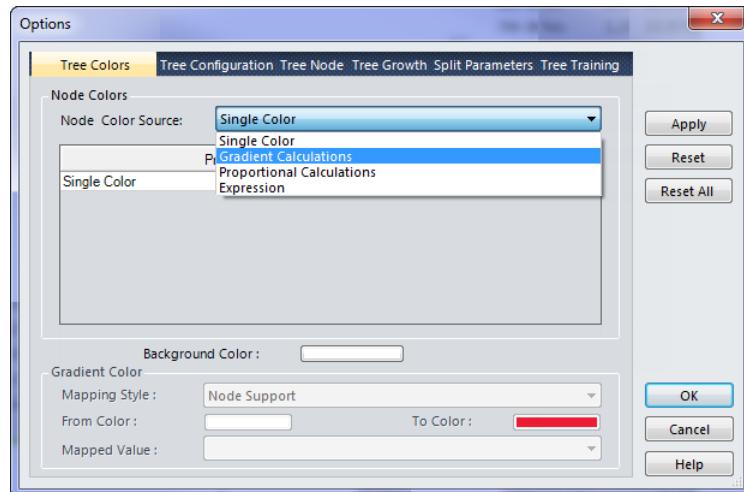


Once **KPIs** have been added, treatments can be applied.

In complex tree with many nodes it may be difficult to identify nodes with higher values for a specific **KPI**. To address this a gradient colouring can be applied to nodes based on a **KPI** value; higher values = deeper colour.

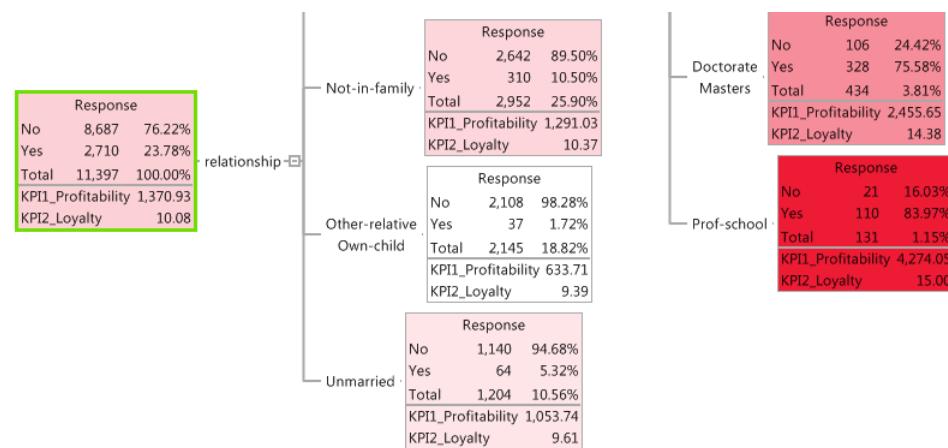
From the **Tree Colours** tab in the **Options** dialog ,select **Gradient Calculation** from the **Node Colour Source** dropdown.

Figure 11.25: Tree Options



Once selected, the **Mapped Value** dropdown become available. Choose the **From Colour** and **To Colour** gradient options for the nodes in the tree based on the value of the selected **KPI**.

Figure 11.26: Gradient Mapped Tree Section



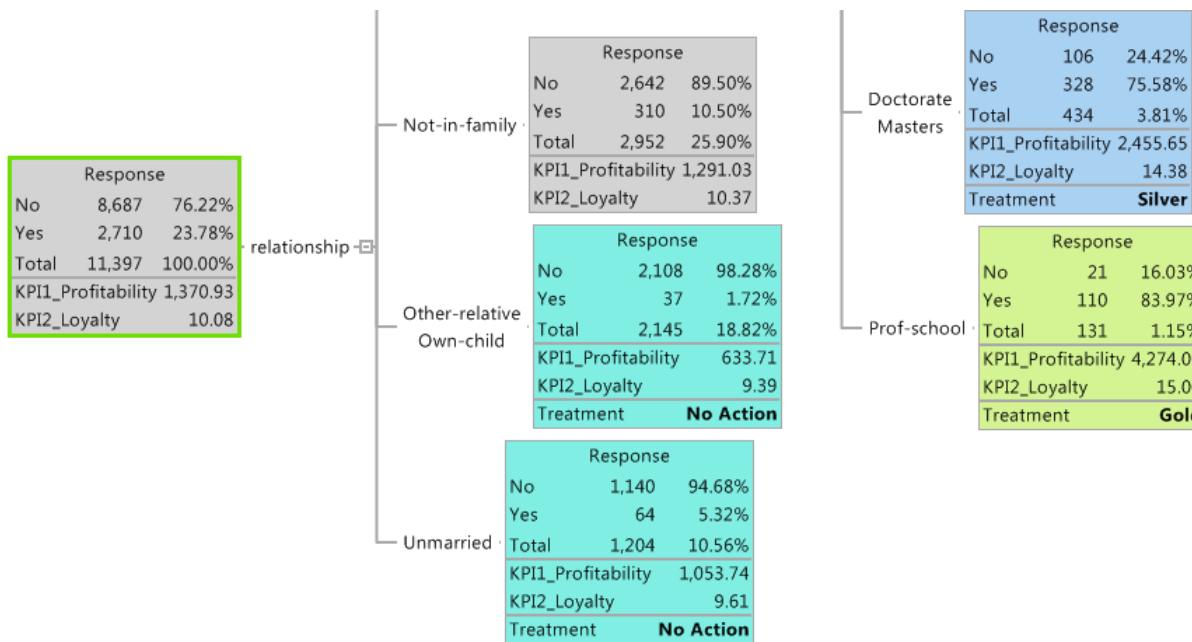
This makes node identification easier and aids in the application of treatments. The next step in the process is of course, the application of treatments, this aspect was introduced earlier, and the treatments

in this instance are identical;

- **Gold**
- **Silver**
- **Bronze**
- **No Action**

The final tree with treatments assigned and colour coded can be seen in figure 11.27.

Figure 11.27: Treatments Applied; Partial View



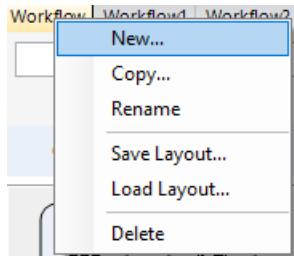
11.5.2 Building a Strategy Tree based on a Dataset

Here, a **Strategy Tree** is created using a dataset and defined by combining objectives for different segments.

Although it is possible to create the new **Strategy Tree** in the same Workflow, it is best to use a clean sheet.

Thankfully multiple **Workflows** can exist in the same project; create a new **Workflow** in the same project by right clicking the current **Workflow** title.

Figure 11.28: Add New Workflow

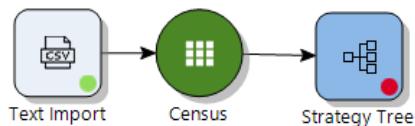


Accept the default name, and using the **Dataset Link** node from the **Manipulate** palette, link to the **Census** dataset.

Rather than partitioning and modelling, this demonstration will jump straight to the creation of the **Strategy Tree** with the entire dataset.

Once a **Strategy Tree** node is added and connected, the **Workflow** should look as depicted in figure 11.29.

Figure 11.29: New Workflow Created

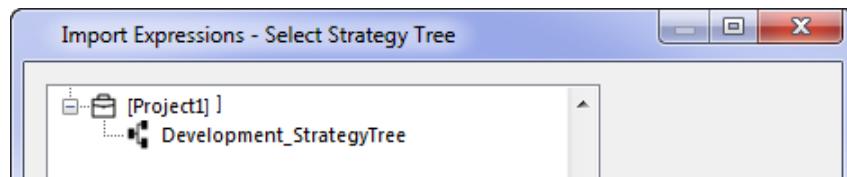


Either by double clicking the **Strategy Tree** node or by right clicking and selecting **Modify**, open the **Insert Strategy Tree - Define Calculations** dialog.

This dialog enables the addition of calculations directly or via **Helpers**. Calculations can also be imported from another **Strategy Tree** or from an **Expressions File (XML)**.

To copy **KPIs** from the previous tree select **Import** and then the option **From another strategy tree**.

Figure 11.30: Import KPIs from another Strategy Tree



Select the **Strategy Tree** created previously and click **Next >**. The **Import Expression – Node Calculations Found** dialog appears and enables selection of the **KPIs** to include.

Figure 11.31: Import Expressions

	Import?	#	Calculation Name	Expression	Format String	Display	Type
▶	<input checked="" type="checkbox"/>	1	KPI1_Profitability	AVG([capital-gain])	(null)	All nodes	NodeCal
	<input checked="" type="checkbox"/>	2	KPI2_loyalty	AVG([num-products])	(null)	All nodes	NodeCal

Click **Finish** to create the **Strategy Tree** and once created open it.

As a result of basing the tree on a dataset and not an existing **Decision Tree** model, only the root node appears with the **KPIs** added.

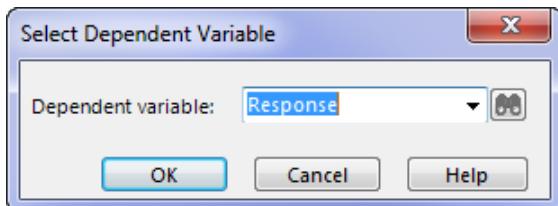
Figure 11.32: Strategy Tree Root Node

Total	16,281	100.00%
KPI1_Profitability	1,354.03	
KPI2_loyalty	10.07	

KnowledgeSTUDIO provides and extends **Decision Tree** functionality to interactively create and build a segmented tree.

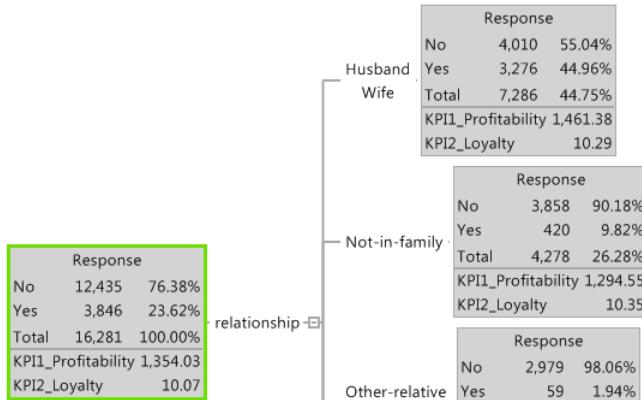
The first step is to use the **Find Split** function at the root node. **Find Split** command prompts for selection of a dependent variable. In this example, **Response** is selected.

Figure 11.33: Select Dependent Variable for Find Split



Based on the dependent variable selected, the tree finds the independent variables that separate the *Yes* and *No* categories of the variable **Response**, just as per a **Decision Tree**. In this instance, the field returned is relationship.

Figure 11.34: Find Split



As has been stated, **Strategy Trees** include and extend **Decision Tree** functionality.

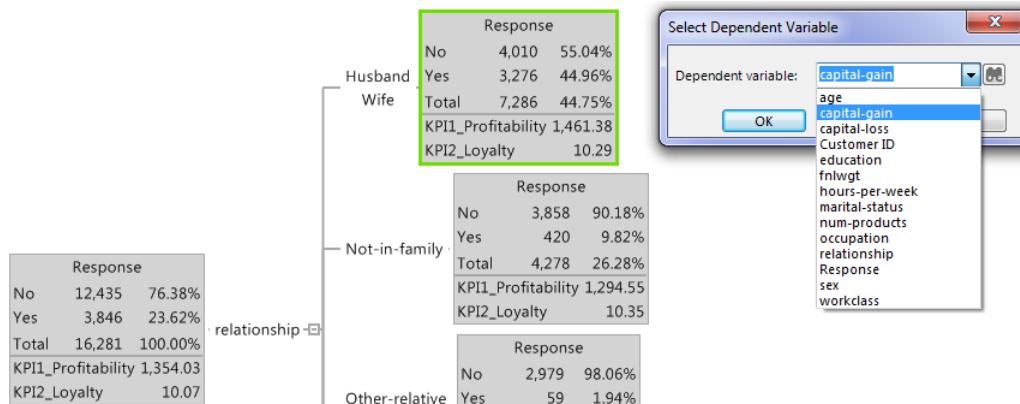
Extended functionality is provided by way of allowing multiple dependent variables and the ability to change the dependent variable for any node. Any node can be selected and again, a split variable can be sought based on an additionally selected dependent variable.

In this example the segment **Husband/Wife** is selected and **Find Split** is chosen. Once more a dialog appears to select a dependent variable.

Selecting capital-gain initiates the **Strategy Tree** to assess all variables and returns a variable and splits that homogenize the dependent variable values.

In the case of a categorical dependent variable, the software attempts to concentrate the proportion of cases in each node into one category of the dependent variable.

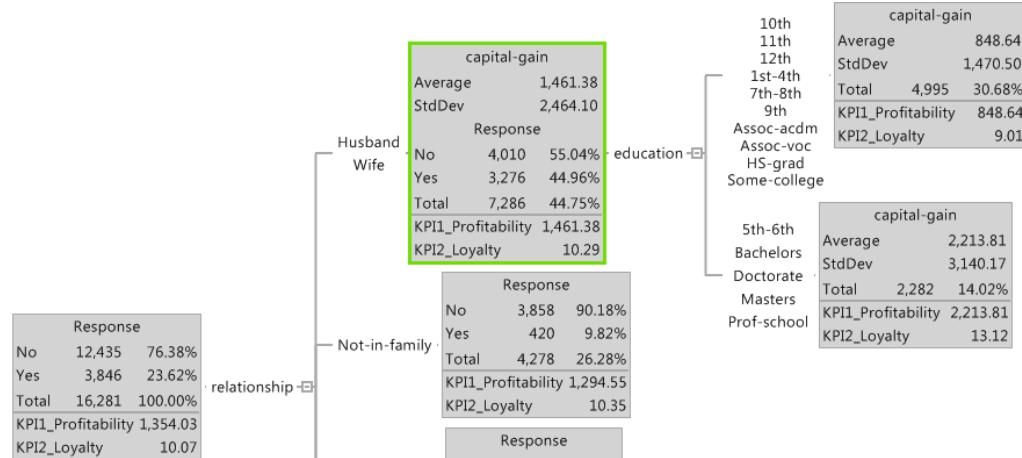
Figure 11.35: Including Additional Dependent Variable



In this instance, the variable *capital_gain* has been selected. The **Root Node**, in this case **Husband/Wife**, shows both the distributions across the first dependent variable selected; *Response*, and also the **Average** and **Standard Deviation** for the dependent variable selected at that node; *capital-gain*.

Notice the resulting splits for *education* show only the **Average** and **Standard Deviation**, i.e. only the values associated with the dependent variable selected at that point. In this way, varying dependent variables can be selected for differing segments.

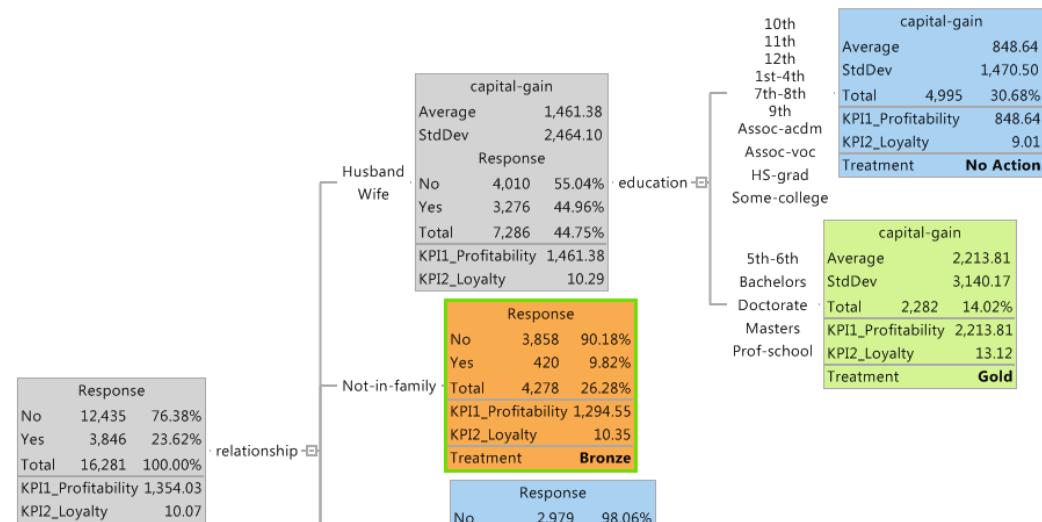
Figure 11.36: Segmenting on *capital-gain*



This capability provides unlimited scope and versatility to assess segments in different ways. For example, a selected set of KPIs may not provide any clarity in distinguishing segments and segmenting further on an additional dependent measure may be the solution.

Additionally, the same dependent variable may not be as applicable or reveal any useful segmentation for one group in comparison to another so having the ability to select an alternative greatly increases the chances of distinguishing segments and enables better treatment. Finally, treatments can be assigned in the usual manner.

Figure 11.37: Treatments Assigned



11.6 Conclusion

KnowledgeSTUDIO Strategy Trees are a unique feature enabling additional node calculations to be added to model results to better determine how best to treat a group of records.

Strategy Trees can be applied to already existing models or to a dataset. **Strategy Trees** include and extend the functionality found in **KnowledgeSTUDIO Decision Trees** and enable further splitting and multiple dependent variables.

Strategy Trees are versatile, with many applications and extensions. As a result of completing this chapter, users should be able to:

- Understand and build **Strategy Trees** from an existing model or on a dataset
- Add additional node calculations
- Assign **Treatments**
- Generate and evaluate assigned **Treatments** using reports
- Modify treatments given limitations on number of assignable treatments
- Include additional dependent variables

Exercises

1. Locate or create a **Decision Tree** for the **Census** dataset
2. Prior to creating a **Strategy Tree**, simplify the **Decision Tree** model
 - (a) Use the **Find Split** option on the root node to grow the first level
 - (b) Grow the tree by splitting on **Husband/Wife** using **Find Split**
 - (c) Flip the model horizontally from the **View** menu select **Horizontal Tree**
3. From the **Model** palette, select the **Strategy Tree** node and drag it onto the **Workflow** canvas
 - (a) Link the appropriate dataset and **Model Instance** to the **Strategy Tree** node
 - (b) Create the **Strategy Tree**
4. Add some **KPIs**
 - (a) Create **KPI1 – Profitability**
 - i. Use the **Average Aggregate** function and the variable *capital_gain*
 - (b) Create **KPI2 – Loyalty**
 - i. Use the **Average Aggregate** function and the variable *num_products*
5. Assess the overall averages for each **KPI** at the root node
6. Compare the **KPIs** at each node to the average **KPIs** and overall response rate to assess which records are of most interest
7. Apply treatments of your choice using the pill icon on the task bar.
8. Using the **Options** button apply colours to the **Treatments** and to the **Strategy Tree**
9. If time permits: Build a **Strategy Tree** based on a dataset. Use the steps in the relevant section of the manual to guide you

Chapter 12: Strategy Validation and Deployment

Chapter 12: Strategy Validation and Deployment

12.1 Introduction

Strategy Validation is a means by which a created strategy can be assessed either prior to deployment or as a means to monitor an already deployed model.

A strategy can be deployed in a similar fashion to a **Decision Tree** or other model, either directly on an open or external data source or as code in a variety of formats for use on other platforms. All code generating and direct deployment nodes are contained in the **Action** palette.

The objectives of this chapter are to ensure users can:

- Validate a **Strategy Tree Model Instance** using the **Model Validation** node
- Monitor a deployed strategy to assess continued viability
- Understand, use, and create varying comparisons to assess strategy performance
- Assess where a strategy is effective and ineffective
- Use a **Strategy Tree Model Instance** to score an existing project dataset
- Export the scored results to a database or specific file format
- Create code for a **Strategy Tree Model Instance**

12.2 Strategy Validation

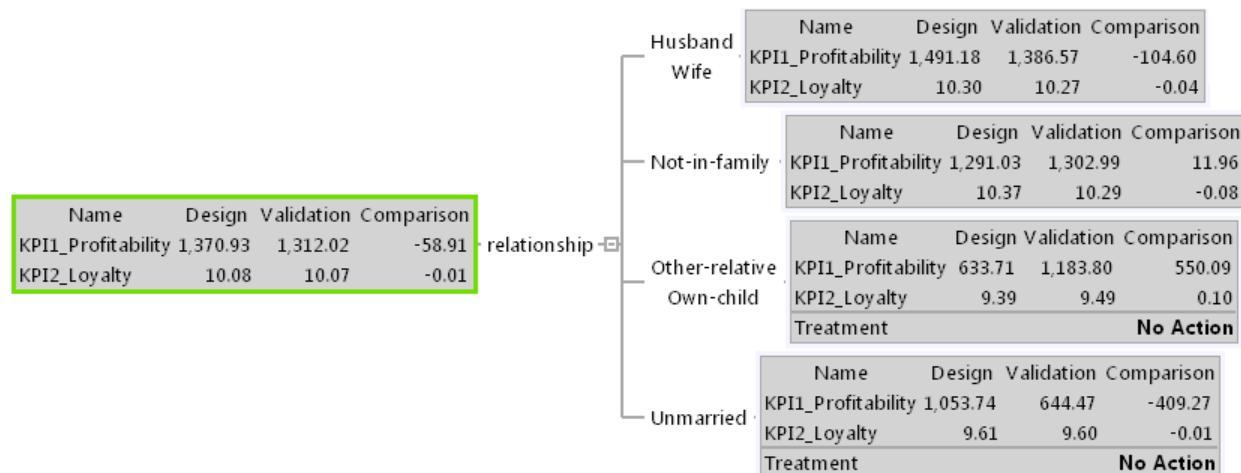
Like **Decision Trees**, **Strategy Trees** can and should be validated. **Validation** of a strategy is the process of applying the same node calculations to a **Validation** dataset and comparing the results to the calculations from the original design dataset.

The results of the comparison are displayed in a **Validation Tree**.

Figure 12.1: Validation Tree Comparing Node Calculations

		Name	Design	Validation	Comparison
		KPI1_Profitability	1,491.18	1,386.57	-104.60
		KPI2_Loyalty	10.30	10.27	-0.04
			Name	Design	Validation
			KPI1_Profitability	1,291.03	1,302.99
			KPI2_Loyalty	10.37	10.29
					-0.08
			Name	Design	Validation
			KPI1_Profitability	633.71	1,183.80
			KPI2_Loyalty	9.39	9.49
			Treatment	No Action	
			Name	Design	Validation
			KPI1_Profitability	1,053.74	644.47
			KPI2_Loyalty	9.61	9.60
			Treatment	No Action	

relationship



```

graph TD
    Husband[Wife] --> Husband
    Husband --> NotInFamily[Not-in-family]
    Husband --> OtherRelative[Other-relative]
    Husband --> Unmarried[Unmarried]
    NotInFamily --> NotInFamily
    NotInFamily --> Treatment[Treatment]
    Treatment --> NoAction[No Action]
    OtherRelative --> OtherRelative
    Unmarried --> Unmarried
  
```

Validation of a Strategy is used to:

- Assess the business validity of the strategy design before deployment
- Monitor the performance of strategy already in production to determine whether it is still relevant
- Assess the performance and validity of a strategy on a node by node basis

12.3 Validating Strategy Trees

Validating a strategy is accomplished using the same processes applied when validating a **Decision Tree**; first a **Strategy Tree Model Instance** is generated. Once complete, add the **Model Validation** node from the **Evaluate** palette and connect the **Model Instance** and the **Validation dataset**.

Figure 12.2: Connections Made



NOTE: **Workflows** can get quite busy but bear in mind that multiple **Workflows** can exist in the same project and datasets and model results can be linked across **Workflows** as well as within the same **Workflow** using the **Dataset Link** and **Model Link** nodes respectively.

To access the **Model Validation** node dialog either double click or right click and select **Modify** (not shown). **This opens the Strategy Validation – Target Tree dialog.**

Figure 12.3: Strategy Validation - Target Tree

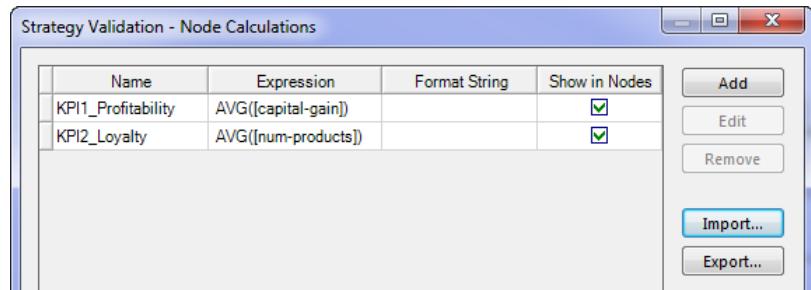


The only modifiable option is the option for assigning a name to the validation results in the **Validation Tree Name** field.

All other aspects are pre-defined as a result of connections and are presented as confirmation of the same.

Click **Next >** to access the **Validation Strategy – Node Calculations** dialog.

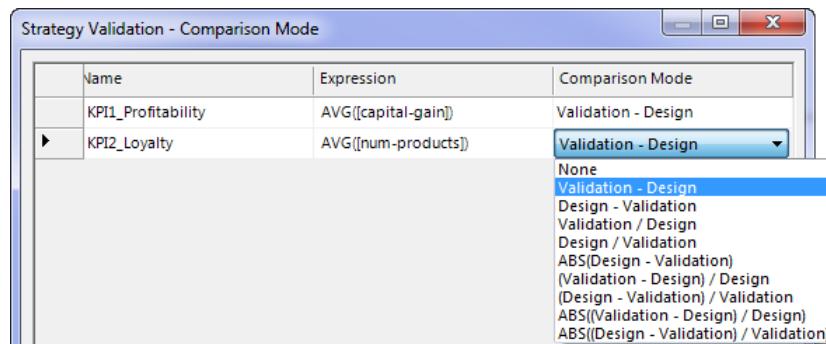
Figure 12.4: Validation Strategy - Node Calculations



The **Validation Strategy – Node Calculations** dialog displays the calculations from the **Model Instance**. These are the **KPIs** added previously and will be applied and compared to the **Validation** dataset.

Additional calculations can be added, removed, imported and exported. Click **Next >** to access the **Strategy Validation – Comparison Mode** dialog.

Figure 12.5: Strategy Validation - Comparison Mode



A variety of comparison modes are available. A dropdown listing the methods is available by clicking the **Comparison Mode** column for any node calculation.

The **Model Instance** and consequently the original **Strategy Tree** is referred to as **Design**, and the **Validation** dataset is referred to as **Validation**.

The default comparison is the difference between the node calculations for the **Validation** dataset minus the calculations for the **Design** dataset.

Table 12.1: Comparison Modes

Comparison Mode	Description
None	No comparison is performed
Validation – Design	Difference between KPIs
Design – Validation	Subtracts the results in the test dataset (Validation) from the data in the learning dataset (Design)
Validation / Design	Ratio of KPI values
Design / Validation	Divides the results in the learning dataset (Design) by the results in the test dataset (Validation)
ABS(Design - Validation)	Absolute value of the differences between KPIs
(Validation - Design) / Design	Produces a ratio of the difference between Validation and Design datasets with Design KPI values as denominator
ABS ((Validation - Design) / Design)	Absolute value of the preceding expression
(Design - Validation) / Validation	Difference between KPIs , with Validation KPI values as denominator
ABS ((Design - Validation) / Validation)	Absolute value of the difference between KPIs , with Validation KPI values as denominator

Clicking **Next >** opens the **Validation Strategy – Field Mapping** dialog. Use this dialog to match fields if necessary. Once complete, click **Run** to create results.

The validation results appear as an object in the **Project Pane**, nested underneath the **Strategy Tree Model Instance**, not shown. Double click results to open or right click the **Model Validation** node on the **Workflow** canvas and click **Open View**.

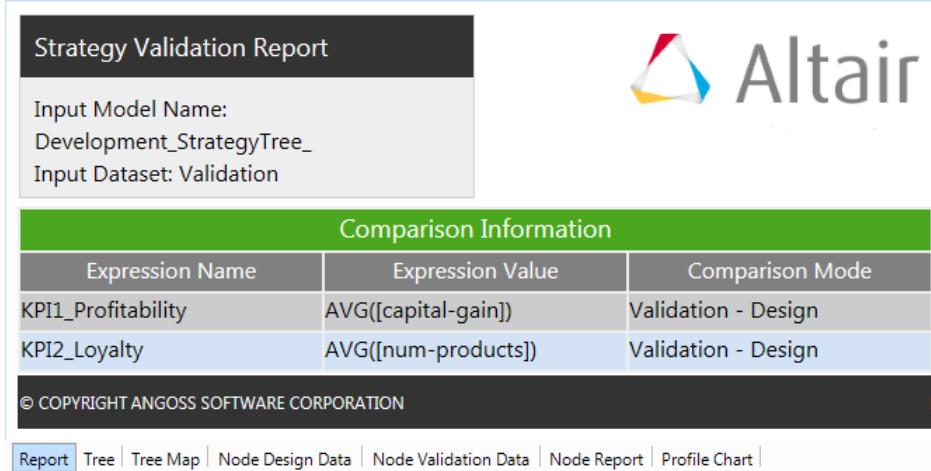
The **Validation** results are provided in a tabbed structure. Each tab provides information in relation to the validated **Strategy Tree**.

The results open on the **Report** tab providing information in relation to the **Strategy Tree KPIs** and the comparison mode selected.

12.3.1 Report Tab

The **Report** tab provides information in relation to the **Strategy Tree KPIs**; the expression used to generate each, and the comparison mode.

Figure 12.6: Strategy Validation Report



The screenshot shows the 'Strategy Validation Report' interface. At the top left, it displays the input model name 'Development_StrategyTree_' and the input dataset 'Validation'. On the right, the Altair logo is visible. Below this, a table titled 'Comparison Information' lists two expressions: 'KPI1_Profitability' and 'KPI2_Loyalty', each with its expression value and comparison mode. The table has three columns: 'Expression Name', 'Expression Value', and 'Comparison Mode'. The 'Comparison Mode' for both entries is 'Validation - Design'. The bottom of the report includes a copyright notice for ANGOSS SOFTWARE CORPORATION and a navigation bar with links to Report, Tree, Tree Map, Node Design Data, Node Validation Data, Node Report, and Profile Chart.

Comparison Information		
Expression Name	Expression Value	Comparison Mode
KPI1_Profitability	AVG([capital-gain])	Validation - Design
KPI2_Loyalty	AVG([num-products])	Validation - Design

© COPYRIGHT ANGOSS SOFTWARE CORPORATION

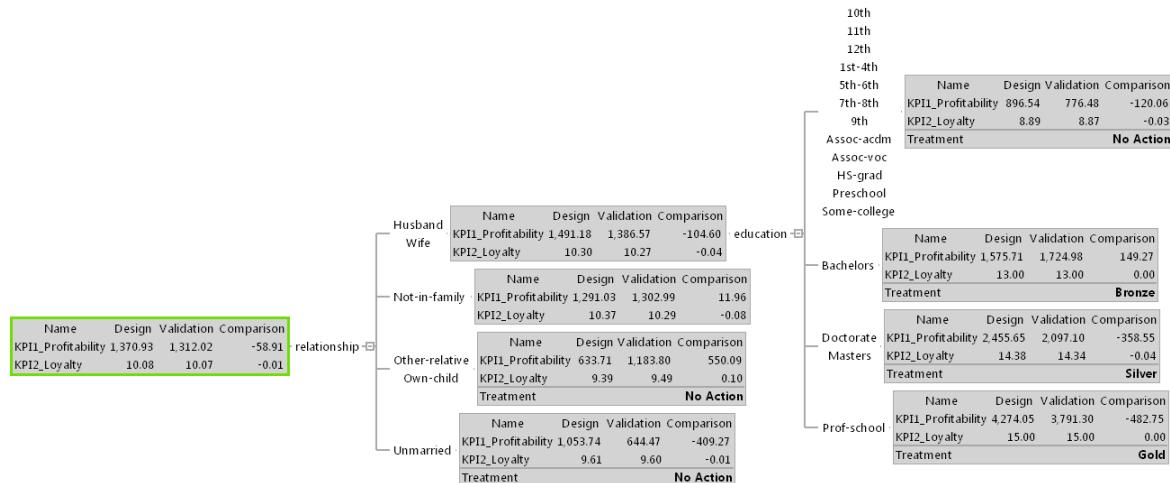
Report | Tree | Tree Map | Node Design Data | Node Validation Data | Node Report | Profile Chart |

12.3.2 Tree tab

The **Tree tab** provides access to the strategy **Validation Tree**.

This displays the results of the node calculations from both the **Design** and **Validation** datasets and retains and displays treatments assigned.

Figure 12.7: Validation Results



The **KPI** comparisons are provided at each node, which includes an overall comparison at the root node. As can be seen, overall the strategy results in a negative value for both **KPI** comparisons.

NOTE: If the validation data is collected after the strategy is implemented and have had time to take effect, then it might be desirable to see positive changes in the **KPI** values, which could indicate that the strategies put in place have worked and improved the business situation.

In this case, both the design and validation data were collected at the same time, so changes would never

be able to indicate an effective strategy. Instead, small changes in **KPI** indicate a stable strategy.

For this example, imagine that the validation data was collected after the strategy was implemented. Since comparisons are provided at each node, strategy strengths and weaknesses can be highlighted.

For example, the **Bronze** treatment for the **Unmarried** segment appears unsuccessful; average profit has significantly decreased.

However, taking **No Action** for the **Other-relative/Own-child** and **Husband/Wife – 1st-4th ... Preschool** segments resulted in positive profitability.

Negative differences may mean a greater need to focus on that specific segment to determine whether an alternative strategy, further segmentation or additional modelling; on the whole or on that specific segment, is required.

12.3.3 Tree Map Tab

The **Tree Map** tab provides a similar representation to its **Decision Tree** counterpart and can be used to navigate large trees.

Selecting any node in the **Tree Map** will highlight the corresponding node information in other tabs (not shown).

12.3.4 Node Design/Validation Tab

The **Node Design Data** and **Node Validation Data** tabs provide a data view of the currently selected node, either from the **Tree** or **Tree Map** tab, not shown.

12.3.5 Node Report Tab

The **Node Report** tab provides a tabular representation of the validation figures and can be easily copied to other tools.

Figure 12.8: Node Report Tab

Show Nodes:	All	Treatments	Design - KPI1_Profitability	Validation - KPI1_Profitability	Comparison - KPI1_Profitability	Design - KPI2_Loyalty	Validation - KPI2_Loyalty
Node Rules							
([relationship] = 'Not-in-family')	N/A	1,291.03	1,302.99	11.96	10.37	10.29	
([relationship] IN ('Husband', 'Wife')) AND ([education] IN ('Doctorate', 'Masters'))	Silver	2,455.65	2,097.10	-358.55	14.38	14.34	
([relationship] IN ('Husband', 'Wife'))	N/A	1,491.18	1,386.57	-104.60	10.30	10.27	

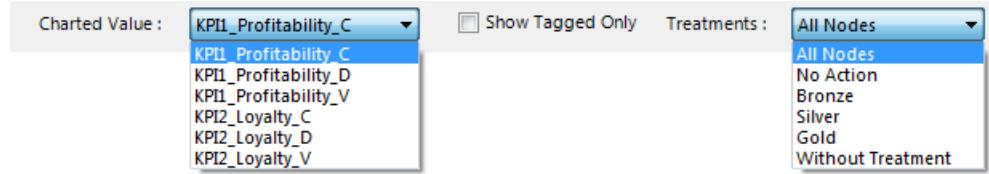
Report | Tree | Tree Map | Node Design Data | Node Validation Data | **Node Report** | Profile Chart |

12.3.6 Profile Chart Tab

The final tab is the **Profile Chart** tab, this provides information on the performance of each node on a selectable **Charted Value** for all or specific **Treatments**.

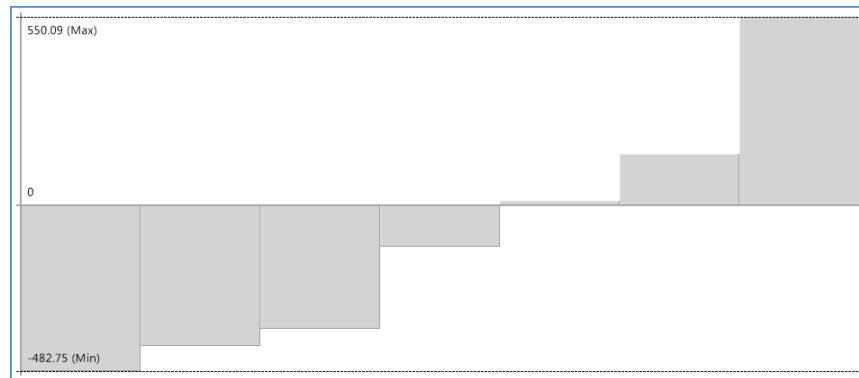
The **Charted Values** available are the **Design**, **Validation** or **Difference** for each calculation included.

Figure 12.9: Profile Chart Options



Node performance is illustrated in ascending order with the worst performing node to the extreme right hand side. Clicking any node bar will again, identify that node in other tabs

Figure 12.10: Profile Chart Tab



Once a strategy has been validated, the final step is deployment.

12.4 Strategy Deployment

The final step of the strategy development process is to deploy the strategy to put it into action. The process to deploy a **Strategy Tree** is identical to that of deploying a **Decision Tree**. To recap, **Strategy Trees** can be deployed in one of two ways;

- Score a current project dataset using the Scoring node
- Generate code for the **Strategy Tree** for use on other platforms

All nodes to aid in the above are located on the **Action** palette. Additionally, if a **Strategy Tree** is used to score a current project dataset, the scored dataset can be exported to a specific file format or sent to a database using the **Data Export** palette.

12.4.1 Score Current Project Dataset

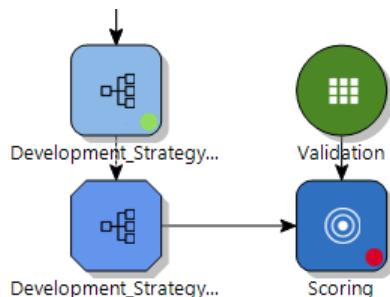
The **Scoring** node from the **Action** palette is used to score an existing project dataset. This will create a new scored dataset with selected fields from the base dataset and additional scoring fields from the **Strategy Tree**.

To begin the process, drag a **Scoring** node from the **Action** palette to the **Workflow** canvas.

Connect the **Strategy Tree Model Instance** and a dataset to score; for this demonstration, the validation partition created previously is used.

NOTE: the **Dataset Link** node is used to reference the **Validation** dataset

Figure 12.11: Scoring Node Added



To access and set options either double click the **Scoring** node or right click and select **Modify**.

Options are identical to the process for scoring a **Decision Tree**, the first dialog, **Strategy Scoring – Target Dataset**, provides information in relation to connections made to the **Scoring** node. The only modifiable option is to assign a name to the target dataset created. Notably the **DV Name** is absent.

The next dialog, **Strategy Scoring – Field Mapping**, provides generic field mapping as previously introduced. The **Strategy Scoring – Treatments** dialog provides the facility to include/exclude treatments from the scored dataset.

Figure 12.12: Strategy Scoring - Treatments



One or more treatments can be selected from those created in the **Strategy Tree**.

The default is to include all treatments when scoring. **Treatments** can be easily moved from the panes using the arrow buttons or by double-clicking on the treatment name.

NOTE: (null) is an available option for scoring and is not selected by default. If included, null is applied records or segments not assigned a treatment in the **Strategy Tree**, if omitted, then records or segments not assigned a treatment are omitted from the created dataset.

Once treatments have been selected, click **Next >** to move to the **Score Strategy – Scoring Fields** dialog.

Figure 12.13: Strategy Scoring - Scoring Fields

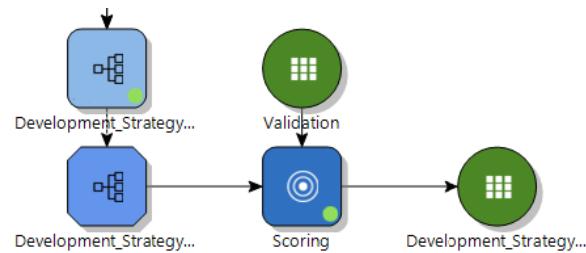
Item	Field Name	Include
Node ID	Node ID	<input checked="" type="checkbox"/>
Node Number	Node Number	<input checked="" type="checkbox"/>
Treatment	Treatment	<input checked="" type="checkbox"/>
Calculation 1	KPI1_Profitability	<input type="checkbox"/>
Calculation 2	KPI2_Loyalty	<input type="checkbox"/>

This dialog provides information on the scoring fields added to the created dataset. The outcome is the **Treatment**, no fields are created with probabilities or category assignment.

The next dialog, **Strategy Scoring – Field Selection**, provides options to specify the fields included in the resulting dataset(not shown).

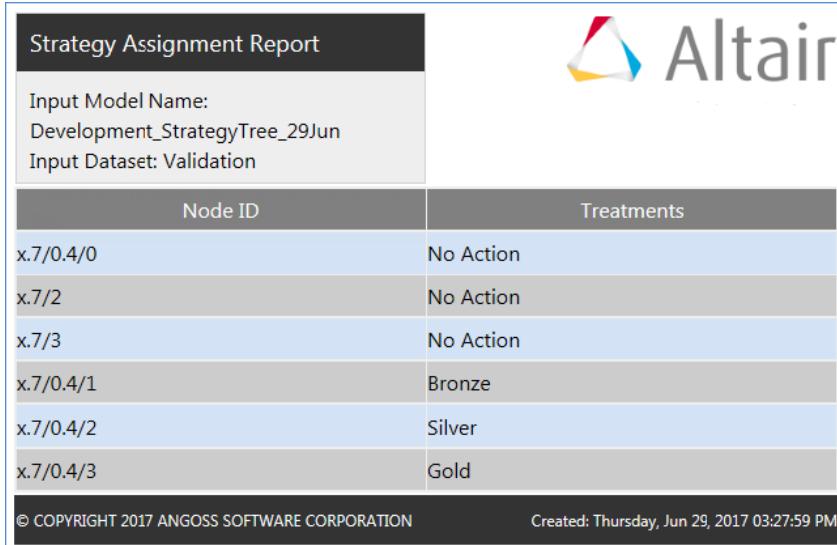
Click **Run** to complete the process. The scored dataset is created in the **Project Pane** and a corresponding node is evident in the **Workflow**

Figure 12.14: Scored Dataset Added to Workflow



Opening the scored results shows the new dataset contains nine tabs. This includes an additional **Report** tab on which the dataset opens by default.

Figure 12.15: Report Tab



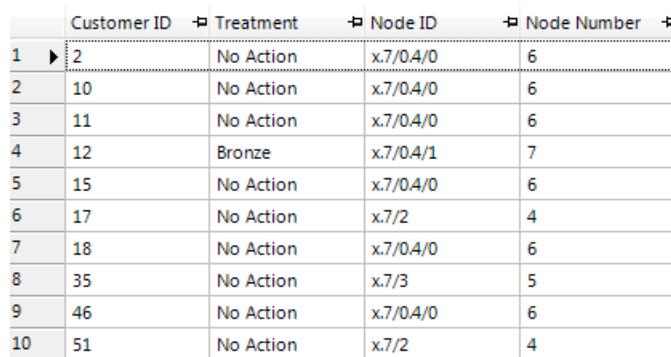
The screenshot shows the 'Strategy Assignment Report' tab. It displays the input model name as 'Development_StrategyTree_29Jun' and the input dataset as 'Validation'. A table lists node IDs and their assigned treatments:

Node ID	Treatments
x.7/0.4/0	No Action
x.7/2	No Action
x.7/3	No Action
x.7/0.4/1	Bronze
x.7/0.4/2	Silver
x.7/0.4/3	Gold

At the bottom, it shows the copyright information '© COPYRIGHT 2017 ANGOSS SOFTWARE CORPORATION' and the creation date 'Created: Thursday, Jun 29, 2017 03:27:59 PM'.

The **Report** tab provides information about the input and output datasets, model scored, treatments assigned and date created. View the treatment field created from the **Data** tab.

Figure 12.16: Data Tab with Treatment field



The screenshot shows the 'Data' tab with a table of data:

	Customer ID	Treatment	Node ID	Node Number
1	2	No Action	x.7/0.4/0	6
2	10	No Action	x.7/0.4/0	6
3	11	No Action	x.7/0.4/0	6
4	12	Bronze	x.7/0.4/1	7
5	15	No Action	x.7/0.4/0	6
6	17	No Action	x.7/2	4
7	18	No Action	x.7/0.4/0	6
8	35	No Action	x.7/3	5
9	46	No Action	x.7/0.4/0	6
10	51	No Action	x.7/2	4

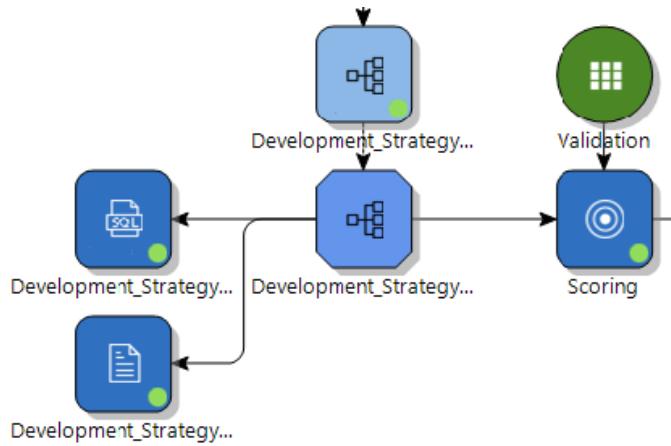
Once the scored dataset has been created it can be exported to an appropriate format using the available nodes from the **Data Export** tab.

12.4.2 Automatic Code Generation

As with **Decision Trees**, code can also be produced for **Strategy Trees**. Again, code can be generated in a variety of formats.

The following illustrations generate code in *SQL* and *LOS* formats using the **Generate SQL** and **Generate LOS** nodes found in the **Deployment** palette, both are added to the **Workflow** simultaneously as depicted.

Figure 12.17: Code Nodes Added to Workflow



Opening the code is a matter of double clicking the items in the Project Pane or right clicking the respective **Workflow** node and selecting **Open View**.

Figure 12.18: *SQL* and *LOS* Code Snippets

<pre> -- SQL Predictive Model -- Block # 1: Calculates the treatments -- Treatments -- (CASE WHEN ("relationship" = 'Husband' or "relationship" = 'Wife') THEN (CASE WHEN ("education" = '10th' or "education" = '11th' or "education" = '12th' or "education" = '1st-4th' or "education" = '5th-6th' or "education" = '7th-8th' or "education" = '9th' or "education" = 'Assoc-acdm' or "education" = 'Assoc-voc' or "education" = 'HS-grad' or "education" = 'Preschool' or "education" = 'Some-college') THEN 'No Action' WHEN "education" = 'Bachelors' THEN 'Bronze' WHEN ("education" = 'Doctorate' or "education" = 'Masters') THEN 'Silver' WHEN "education" = 'Prof-school' THEN 'Gold') </pre>	<pre> /* * This LOS program consists of two parts. The first part deals with the * missing values. The second part runs the rules to get score. */ options symbolgen; options mprint; options mlogic; %macro TreeRule(DsIn, DsOut); data &DsOut; set &DsIn; length Treatment \$80; NODENUMBER = 1; Treatment = ''; IF relationship = 'Husband' </pre>
---	---

The code snippets above can be easily exported using the **Save As** option from the **File** menu or copied and pasted to an appropriate file format or application.

Notice the *LOS* code generates a macro that requires specification of the input and output datasets to use.

12.5 Conclusion

Strategy **Validation** and **Deployment** are the final stages in any *Data Mining* project. The processes are made easy to access and easy to use with **KnowledgeSTUDIO** functionality.

As a result of completing this chapter users should be able to:

- Validate a **Strategy Tree Model Instance** using the **Model Validation** node
- Monitor a deployed strategy to assess continued viability
- Understand, use, and create varying comparisons to assess strategy performance
- Assess where a strategy is effective and ineffective
- Use a **Strategy Tree Model Instance** to score an existing project dataset
- Export the scored results to a database or specific file format
- Create code for a **Strategy Tree Model Instance**

Exercises

1. Validate a **Strategy Tree Model Instance** using the **Model Validation** node
2. Choose a partition/existing dataset to validate
3. Explore the different comparison methods to fully understand each, use the Help menu if necessary
4. Assess results on the whole and for each node
5. Identify where the strategy is weak and where it strong. Can anything be done to assess weakness (e.g. further segmentation/different action)?
6. Deploy the **Strategy Tree** by scoring an existing project dataset
7. Assess the generated dataset **Report** tab
8. Familiarize yourself with the variables created by referring to the Data tab
9. Export the results to a text file format in a convenient location
10. Deploy the **Strategy Tree** as code
 - (a) Create code in a suitable format
 - (b) Compare the code results to that of a **Decision Tree** code
11. Save the code results to a file using the **Save As** option from the **File** menu

Chapter 13: Linear Regression

13.1 Introduction

Linear Regression is a primary statistical technique designed to model the linear relationship between an outcome or dependent variable and a set of inputs; predictors, drivers or **Independent Variables**.

The **Dependent Variable** must be a scale or continuous variable whereas the inputs, although traditionally and preferably continuous, can be either categorical or continuous.

Regression modelling provides a means to assess not only the degree to which the outcome or **Dependent Variable** can be determined, but also the effect of each predictor on that outcome.

Linear Regression comes in many forms, two are considered here:

- Simple linear regression
- Multiple linear regression

As a result of completing this chapter, users should be able to:

- Describe **Simple** and **Multiple Linear Regression**
- Develop, evaluate, validate and deploy **Linear Regression** models using **KnowledgeSTUDIO**

13.2 Description

The following sections describe both simple and multiple linear regression.

13.2.1 Simple Linear Regression

In **Simple Linear Regression**, only two variables exist in the model; the outcome or dependent variable and a single input predictor.

The equation used to represent the relationship takes the form:

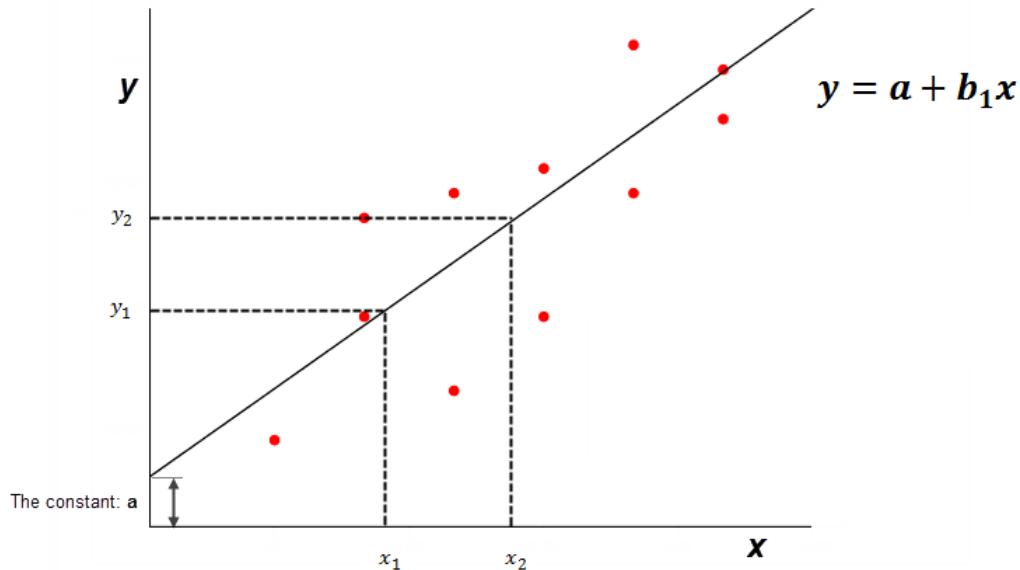
$$y = a + b_1x$$

Where:

- y is the dependent variable
- a is the constant or intercept coefficient
- b_1 is the regression coefficient for the independent variable: x

A regression line can be fit to a scatterplot of any two continuous variables to further clarify and understand the elements of the regression equation.

Figure 13.1: Regression Line fit to Scatterplot



Revisiting the elements of the equations and using the graph as a means to qualify, it can be said that:

- y is value of the dependent variable for a given value of x
- a , the constant or intercept coefficient, is the value of y when the value of x is set to zero
- b_1 is the regression coefficient for the independent variable: x . This can be interpreted in a variety of ways:
 - The gradient or slope
 - How y changes for a constant change in x
 - The change/impact on y for a unit change in x

The slope is calculated using the formula:

$$b_1 = \frac{y_2 - y_1}{x_2 - x_1}$$

NOTE: Regression lines can be fit to data using a number of different methods. The most popular being the method of least squares.

This method minimizes the sum of squared distances between the points and the line. Other methods include minimizing based on the lack of fit. Least squares can also be used to fit non-linear models.

13.2.2 Applying Simple Linear Regression in Practice

Simple Linear Regression can be put into a more practical framework by applying it to a real world example.

Let's assume there is interest in predicting monthly spend. Only one predictor is used to determine this; *No.Children*. The regression equation that results from this model is:

$$\text{monthlyspend} = 240 + 145 * \text{No.Children}$$

$$y = a + b_1 * x$$

here:

- y is the predicted spend
- 240 is the constant, and is the value of y when the value of x is set to zero
- 145 is the regression coefficient for the independent variable; x

Interpreting the results it can be said that:

- 240 is the monthly spend, when the value of the independent variable is set to 0. Practically, this is predicted spend of those with no children
- The regression coefficient is positive, meaning that the value of the dependent variable increases with x . This is as expected, as extra mouths to feed requires extra spend
- The coefficient value of 145 is interpreted as: for every additional child, monthly spend increases by 145

The regression equation can be applied in practice to predict monthly spend given number of children. Regression also allows for 'what-if' scenario's. For example:

what if there are four children, what is the predicted monthly spend?

Using the equation and inputting the value 4 for x gives:

$$y = 240 + 145 * 4$$

This returns the value: 820.

13.2.3 Multiple Linear Regression

Multiple Linear Regression is an extension of simple linear regression. Whereas **Simple Linear Regression** has one independent variable, **Multiple Linear Regression** includes more than one. In mathematical form:

$$y = a + b_1 + b_2x_2 + \dots + b_nx_n$$

There is a separate coefficient for each independent variable. To put this into practical terms, and using the relationship introduced previously: when predicting monthly spend it may be that other variables such as *gender*, *region*, *income*, *no. cars*, *credit cards*, etc. associate with, and help determine monthly spend.

Extending the simple regression equation with real values and including *IncomeDifference* in thousands, gives:

$$\text{monthlyspend} = 240 + 145 * \text{No.Children} + 56 * \text{IncomeDifference}$$

NOTE: *IncomeDifference* refers to the difference in thousands, between *income* and average *income*. A value of 0 means average *income*, 1 equals 1000 above average, 2, 2000, etc.

This is referred to as centering variable and a commonly used device to aid interpretation.

Each coefficient is interpreted as before, but now there are two.

- The dependent variable, y , is the predicted monthly spend
- 240 is the constant, and is the value of y when both independent variable values are set to zero
- 145 is the regression coefficient for the independent variable: *No.Children*

Interpreting the equation it can be said that:

- 240 (dollars/pounds/euro) is the monthly spend, when the value of both independent variables are set to 0. This is the monthly spend by those with no children earning average income
- The regression coefficients are both positive, meaning that the value of the dependent variable increases when these variable values also increase. This is as expected
- The coefficient value of 145 is interpreted as: for every additional child, monthly spend increases by 145

Again, what-if scenarios can be assessed: Given four children and income is 2000 above the average, monthly spend is predicted to be 932 and calculated as:

$$y = 240 + (145 * 4) + (56 * 2)$$

A peculiar situation arises in the situation where the *No.Children* is 0 and *IncomeDifference* is -5, i.e. 5000 below average income.

The predicted value of the dependent variable is negative:

$$y = 240 + (56 * -5) = 240 - 280 = -40$$

This highlights not only the usefulness of the equation to identify anomalies but also the need to evaluate the:

- Range of independent variable values used in the model
- Range of resulting predictions

NOTE: The model constant can be suppressed to force a predicted value of zero when all inputs are set to zero, but this will not stop negative predictions.

13.2.4 Linear Regression Assumptions

Linear regression models make the following assumptions:

- The dependent and independent variables are continuous
- Each independent variable is linearly associated with the dependent variable
- The independent variables are not related to each other (collinear)
- The residuals are:
 - Random
 - Homoscedastic
 - Independent of the prediction

The Dependent and Independent Variables are Continuous

A primary assumption of any linear regression model is that the dependent and independent variables should be continuous and linearly related.

The linearity assumption is violated in general, in the case of categorical predictors. However these variables can be included if they are dummy coded.

This may seem impossible but consider the example of modelling number of years of education with the variable gender, coded such that 0 is male and 1 is female.

This is a binary variable and can be included as its coefficient can be spoken of in terms of a constant or unit change; the impact on the dependent variable as the binary variable moves from 0 to 1; in the case of *gender; male to female*.

The binary or dichotomous nature of the variable gender enables its inclusion and interpretation. The resulting equation becomes:

$$\text{No.yrs Educ} = 13 + .25 * \text{gender}$$

The results are interpreted as usual:

- The constant is value of the dependent variable when *gender* is set to 0. Here, this is 13 and is the predicted number of years of education for the category *male*
- The coefficient for gender, 0.25, is the change/impact on number of years of education for a unit change in *gender*. In this instance, moving from the value 0 (*male*) to 1 (*female*). Therefore when *gender* is 1 the value of the dependent variable is 13.25. This is the predicted number of years of education for *gender = female*, and of course, the coefficient is the average number of years education difference between the sexes

As a direct result of the characteristics of binary variables, any categorical variable, with any number of categories, can be included in any regression model by creating a set of binary, or dichotomous variables to replace it. The process of creating the binary equivalents is called dummy coding.

Take for example the variable *Marital_Status*. This variable has three values: *Married*; *Divorced*, *Separated*, or *Widowed*; and *Never Married*.

This variable cannot be included in a model as is, but it can be represented using a set of dummy coded variables, and they can be included in its place. A matrix can be used to understand the coding and is illustrated in figure 13.2

Figure 13.2: Dummy Coding Matrix

		Resulting Dummy Coded Variables		
		Married	DSW	NeverMarried
Original	Married	1	0	0
Variable	Divorced, Separated, Widowed	0	1	0
Values	Never Married	0	0	1

As can be seen from the matrix, the number of dummy coded variables created equals the number of values in the original variable, in this case, three.

Conventionally, $n - 1$ dummy coded variables are needed to represent the original variable. Also, if all three are included, the model becomes unstable and multicollinearity issues arise, therefore one of the categories is chosen as the reference category and its dummy coded equivalent is excluded.

A simple example should suffice to illustrate results. Assuming a model predicting monthly spend using the dummy coded predictions with reference category *NeverMarried*, gives the following equation:

$$\text{monthlyspend} = 60 + 10 * \text{DSW} + 30 * \text{Married}$$

- The constant is the value of the dependent variable when all other variables are set to zero. When both *DSW* and *Married* are set to zero, the value 60 is the predicted monthly spend for those *NeverMarried*.
- The outcome for any category can be found by setting the value for all others equal to zero. For example, for the category; *DSW* the result is: 70
- The coefficients for both *DSW* and *Married*, are positive and reflect the difference in spend in relation to the reference category *NeverMarried*

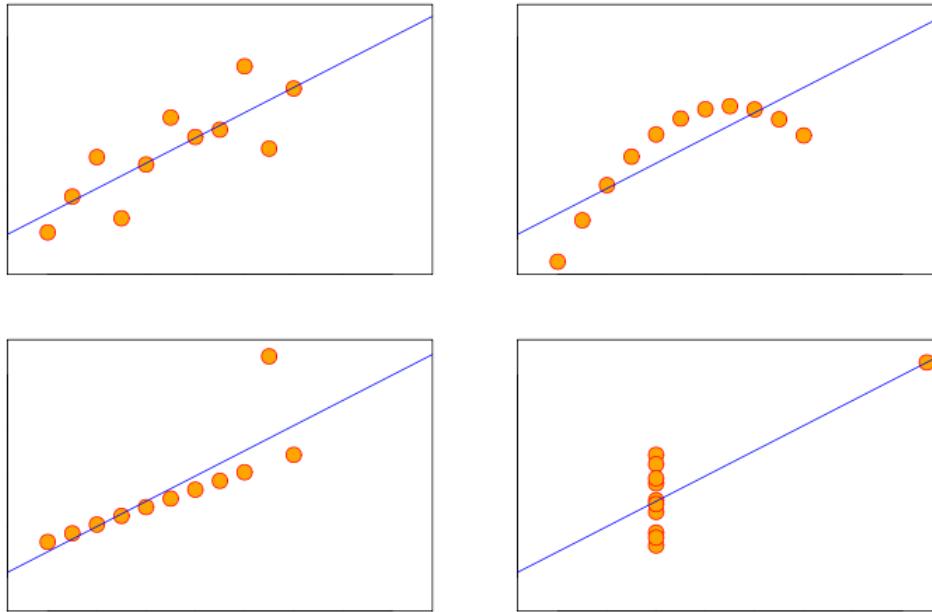
Each Independent Variable is Linearly Associated with the Dependent Variable

The linearity of the association between the dependent and each independent variable can be assessed using graphs and statistics. The most common statistic being the correlation coefficient: r .

The correlation coefficient quantifies the degree of association between pairs of continuous variables and varies between -1 to +1. Values closer to -1 or +1 represent stronger relationships. The closer the value is to 0, the weaker the relationship.

This is an invaluable statistic in linear regression and in modelling in general. However, deferring to statistics alone can lead to errors. This was laid bare by the statistician *Francis Anscombe* using four simple datasets and graphs more commonly known as **Anscombes quartet**.

Figure 13.3: Anscombes Quartet



The figures show how residuals can affect the linearity of the association and lead to errors, even with large correlation coefficients. For each scatterplot, the correlation coefficients are identical at 0.816.

The first scatterplot, top left, shows an acceptable association. The top right is clearly non-linear. In the bottom left; the association is underestimated due to an outlier. The final plot shows that one outlier is

enough to produce a high correlation, even though the relationship is not linear, possibly non-existent.

This demonstrates the use of graphs to broaden understanding of data when assessing associations. It also highlights errors due to outliers and the need for careful assessment prior to model building.

The method used to identify potential predictors is also applied to the independent variables to reveal associations between pairs of independent variables.

High correlations between pairs of predictor variables, spurious relationships and interactions can lead to inaccurate models. This can be referred to, in general, as multicollinearity.

Multicollinearity is made easier to detect with **Variance Inflation Factors**, **VIF**, however assessing relationships between predictors using **KnowledgeSTUDIO** functionality is much advised.

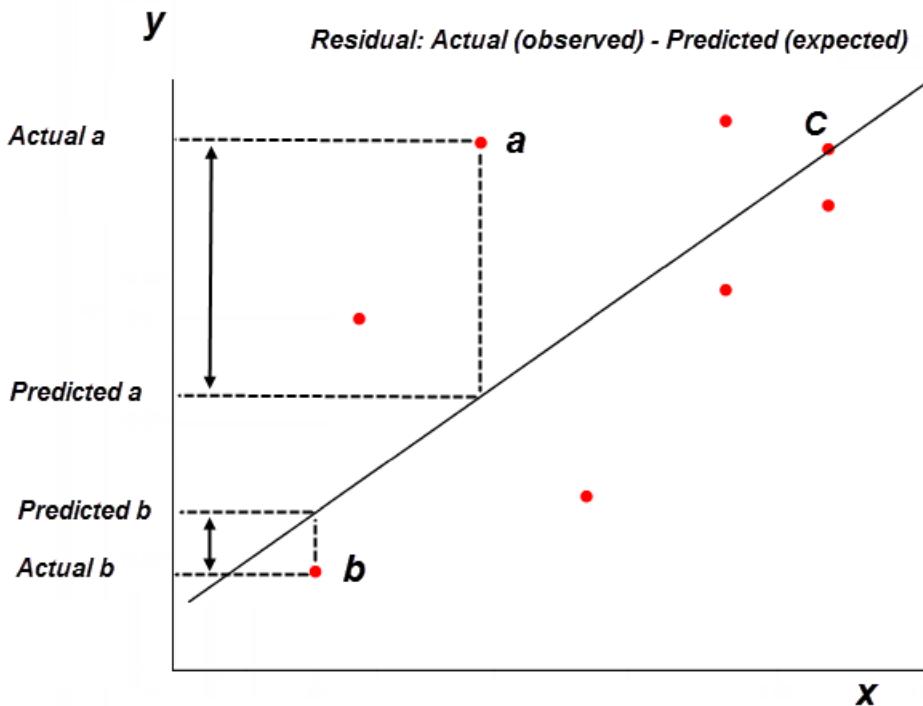
Residuals

The difference between the actual or observed value of the dependent variable and the corresponding predicted or expected value of the dependent variable is referred to as the error or residual.

$$\text{Residual} = \text{Actual (Observed)} - \text{Predicted (Expected)}$$

Generally, predictions contain some degree of inaccuracy; over or under predicting the dependent

Figure 13.4: Residuals: over and under predictions



For the three points: **a, b, c:**

- **c** is on the line. Therefore its actual and predicted values are identical
- The points **a & b**, are not on the line. For these cases, the predicted values contain some degree of inaccuracy:
 - The actual value of point **a** is greater than its predicted value and results in a positive residual. Positive residuals relate to under predictions
 - The actual value of the point **b** is lower than its predicted value and results in a negative residual. Negative residuals relate to over predictions

Residual analysis can be used to assess whether a model has been *well specified*. This is a generic statistical term and in plain English can be loosely interpreted as:

Has everything that affects the dependent variable been included in the model?

For a well specified model, the residuals should be:

- **Random** the errors should be normally distributed
- **Homoscedastic** error variance should be constant over time and over the predicted values
- **Independent** the errors should not be related to the predictions

These aspects can be assessed by referring to the graphs generated by the **Model Analyser**. If a model violates these assumptions it cannot be said to be a well specified model. To give a simple example;

If it is known that monthly spend can be determined accurately by disposable income and number of children, but a regression model including only disposable income is fit, then since number of children is excluded from the model, the effect of number of children will show up in the residual. Residual analysis should then reveal that the model is not well specified. In this way, residual analysis can lead to a better understanding of whether all relevant predictors have been included in the model.

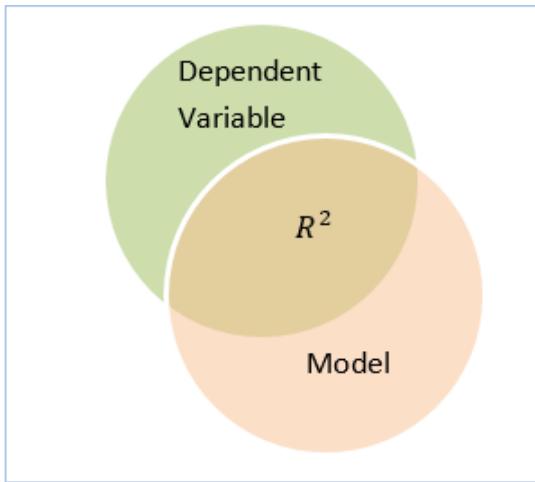
13.2.5 Model Accuracy: FIT

Traditionally model accuracy is assessed using a combination of aspects, including residual analysis.

Typically, the **R-Square** statistic is looked at more closely than just **r**. This is a primary indicator of **Goodness of Fit**. This value varies between 0 and 1 and interpreted as a proportion.

A higher **R-square** value means a greater proportion of the dependent variable can be explained by the model. This can be represented graphically.

Figure 13.5: Regression Model R-Square



Although a higher **R-Square** is preferable it may be limited by data, model specifications or other factors.

13.2.6 Model Accuracy: Variable Coefficients

Assessing individual variable coefficients is also referred to when assessing model accuracy.

All coefficients must make sense in relation to their effect on the dependent variable and should be significantly affecting the dependent variable; unless there is an underlying need for their retention.

For example, when modelling monthly spend it would make sense to assume that increased income increases monthly spend and similarly with number of children. Therefore their coefficients should be positive.

If this is not the case and the coefficients are counterintuitive, it may be necessary to remove the predictor or consider a further transformation such as logs, differences or interaction terms.

NOTE: The **Dataset Editor** provides a **Helper** for automatic generation of interaction terms

13.2.7 Steps when Developing Linear Regression Models

Linear regression is a fundamental statistical technique and has been developed considerably since its inception.

Table 13.1 lists, in general, the steps involved when developing a linear regression model, Altair functionality at each step, and points to note at different stages in model development.

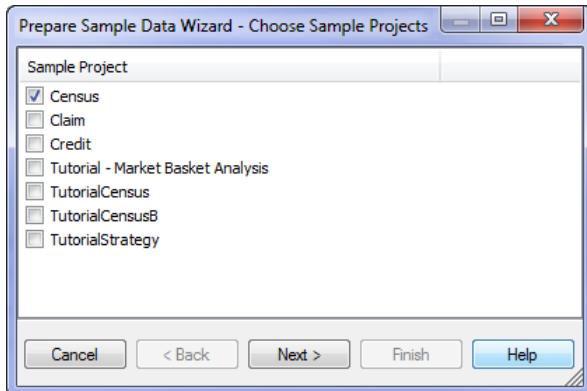
Table 13.1: Model Stages

Stage	Objective	Functionality	Points to Note
Data Exploration	Variable summaries & relationships	Overview Report tab Charts tab Data tab Segment Viewer tab MPP Crosstabulations Scatterplots Characteristic Analysis Correlations Decision Trees	Missing values Transformations Interactions Spurious relationships Relationships with the dependent variable Relationships within the independent variables
Data Preparation	Create dataset partitions	Insert Dataset Partition...	Partition sizes
Modelling	Accuracy: Model evaluation	Model output R-square Significance	R-Square value Appropriate variables included Predictors are significant Coefficients are explainable No multicollinearity Average & range of predictions
Residuals	Accuracy: Residual Analysis	Statistics and Bias charts	Min Max Average Residuals are random Independent Homogenous variance
Validation	Stable model	Model Analyser	Model validates well
Deployment	Score data	Score open dataset Code generation Export to file/database	

13.3 Linear Regression in KnowledgeSTUDIO

The following demonstration uses the **Census** dataset. The file is located in the **Prepare Sample Data...** dialog from the **Help menu**.

Figure 13.6: Prepare Sample Data Dialog



This project contains a number of elements and a pre-prepared **Workflow**. Clear the **Workflow** canvas and remove all elements except the **Census** dataset from the **Project Pane**. The project should be empty with only the **Census** dataset node on the canvas.

The dataset contains a mixture of string and numeric variables. The dependent variable is; *hours-per-week*.

13.4 Data Exploration

Initial exploration begins with data understanding using statistics and graphs. The **Overview Report** contains information that can be used to gain insight prior to any modelling.

Figure 13.7: Census Overview Report

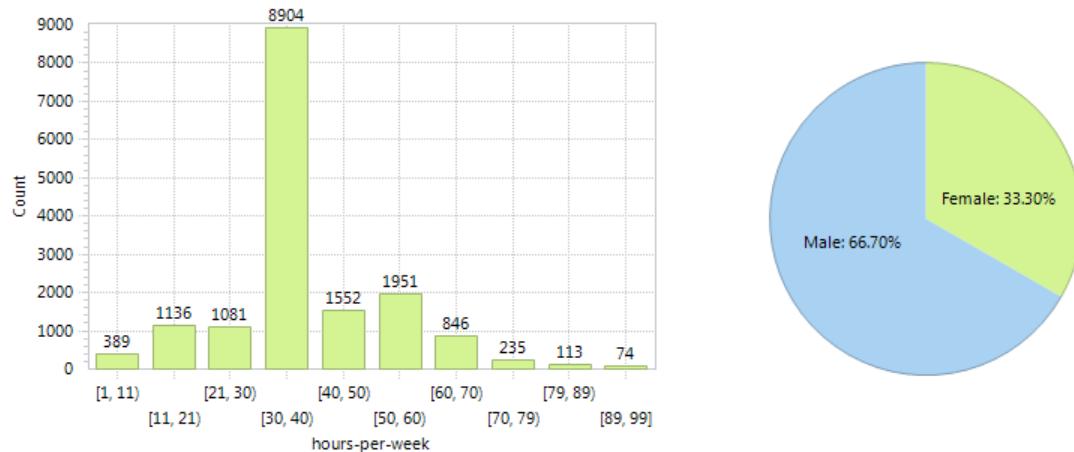
[Census].[Census] X										
Calculate		Calculate All		Dataset: [Census].[Census]		Weight: [No weight]		Records: 16,281		Fields: 14
#	Field Name	Field Label	Data Type	Cardinality	Unique Count	# of Missing Values	% of Missing Values	Minimum	Maximum	Mean
1	age	age	Number	73	0	0	0.00 %	17.00	90.00	38.77
2	workclass	workclass	String	9	0	0	0.00 %	?	Without-pay	
3	fnlwgt	fnlwgt	Number	12787	10275	0	0.00 %	13,492.00	1,490,400.00	189,435.68
4	education	education	String	16	0	0	0.00 %	10th	Some-college	
5	education-num	education-num	Number	16	0	0	0.00 %	1.00	16.00	10.07
6	marital-status	marital-status	String	7	0	0	0.00 %	Divorced	Widowed	
7	occupation	occupation	String	15	0	0	0.00 %	?	Transport-moving	
8	relationship	relationship	String	6	0	0	0.00 %	Husband	Wife	
9	sex	sex	String	2	0	0	0.00 %	Female	Male	
10	capital-gain	capital-gain	Number	113	22	0	0.00 %	0.00	99,999.00	1,081.91
11	capital-loss	capital-loss	Number	82	15	0	0.00 %	0.00	3,770.00	87.90
12	hours-per-week	hours-per-week	Number	89	4	0	0.00 %	1.00	99.00	40.39
13	native-country	native-country	String	41	0	0	0.00 %	?	Yugoslavia	

For example:

- The dataset contains 14 variables & 16281 cases
- Demographics and some financial variables
- There is a mixture of string and number variables
- There are no missing values
- Minimum *age* is 17
- Average number of hours worked per week is slight in excess of 40, the maximum value is 99

The **Dataset Chart** tab can be used to generate variable distributions to gain additional insights as illustrated in figure 13.8.

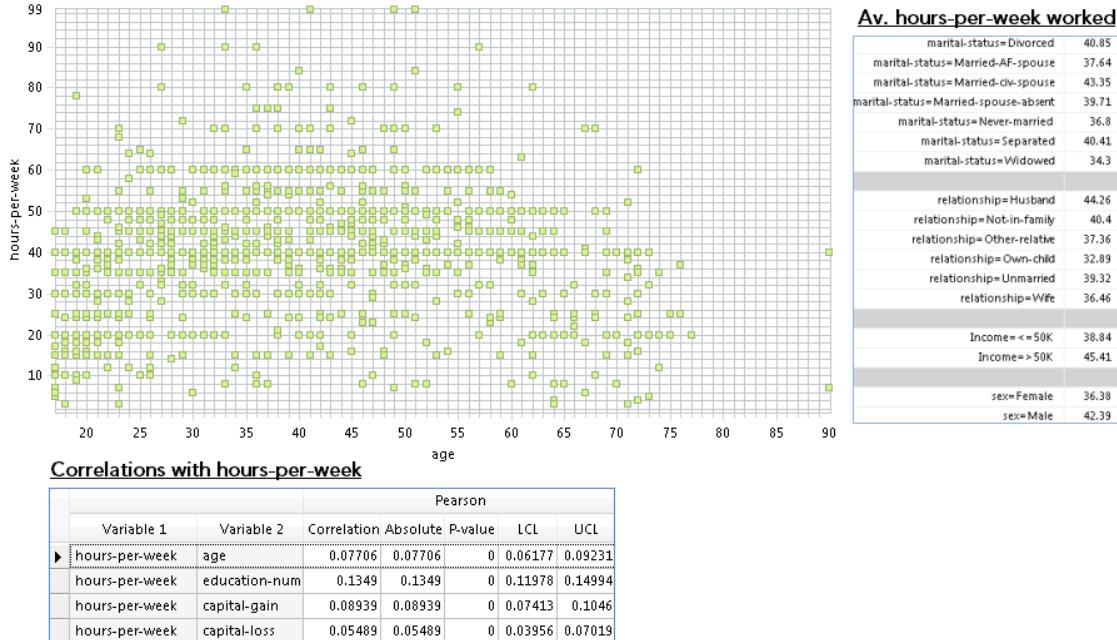
Figure 13.8: Variable Charts



Once univariate analysis has been conducted, other profiling tabs can be used to identify potentially good predictors of the dependent variable. For Example figure 13.9 highlights the following:

- The scatterplot of *age* and *hours-per-week* highlights a potentially non-linear relationship and may lead to further investigation and variable transformations, such as binning
- The table of averages highlights differences across the dependent variable categories
- The correlations table shows associations between *hours-per-week* and other continuous variables

Figure 13.9: Exploration



In addition to using the methods listed above to assess potentially good predictors, other complementary elements can be used such as:

- **Measures of Predictive Power node**
- **Variable Selection node**
- **Decision Tree**

Using these methods, an initial set of candidate predictors can be selected for inclusion in the model. For this demonstration, the following variables are selected:

- *age*
- *sex*
- *marital_status*
- *relationship*
- *education-num*

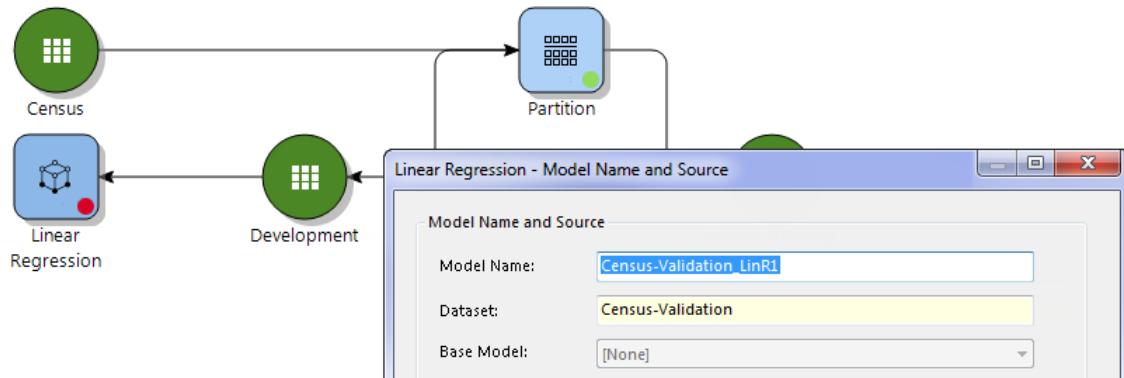
13.4.1 Data Preparation

For this demonstration, two partitions are created with a 70/30 split. Partitions are called **Development** and **Validation** respectively (not shown).

13.4.2 Building the Linear Regression Model in KnowledgeSTUDIO

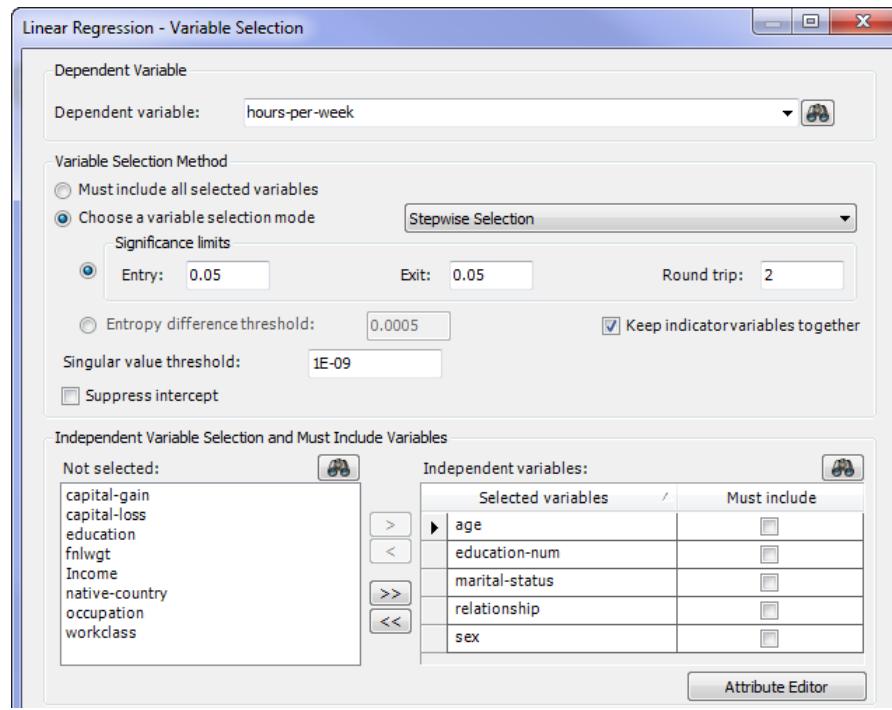
Drag a **Linear Regression** node from the **Modeling** palette to the **Workflow** canvas, connect to the **Development** partition and open as illustrated in figure 13.10.

Figure 13.10: Linear Regression Model Configuration



The first dialog provides options to specify the **Model Name** and choose the **Base Model** if applicable. The **Dataset** option is predetermined and populated based on connections made. Click **Next >** to open the **Linear Regression – Variable Selection** dialog.

Figure 13.11: Linear Regression - Variable Selection



The **Linear Regression – Variable Selection** dialog provides three distinct areas:

- **Dependent Variable**
 - Provides options to specify the **Dependent Variable**
- **Variable Selection Method**

- Options for including variables in the model; see table 13.2
- **Independent Variable Selection and Must Include Variables**
 - Candidate variable specification, including force options and **Attribute Editor**
 - The Attribute Editor enables access to modifiable variable properties including missing value treatment and reference category selection for categorical variables

The **Variable Selection Methods**, **Significance Limits** and **Attribute Editor** options are explained in table 13.2.

Table 13.2: Predictive Model - Variable Selection Method Options

Option	Description
Must include all selected variables	Default Builds a model with all variables regardless of significance
Choose a variable selection mode	Stepwise Selection Variables are included sequentially based on statistical association or Entropy difference Forward Selection Variables are continually assessed. If other variables join, current variables are retained if still significant Model building stops when no variable meets inclusion criteria
	Backward Selection Begins with all variables in the model and discards most insignificant predictors sequentially Model building stops when no remaining variables can be removed
R-square selection	Variables are included based on significantly increasing R-Square Model building stops when no remaining variables significantly increase R-Square
Variable Sequence Of My Choice	Variables are included based on a user set of sequences and significance

NOTE: Variable selection is assessed by referring to either **Significance Limits** or an **Entropy Difference Threshold**, both can be user-defined.

Options related to significance limits are detailed in table 13.3.

Table 13.3: Significance Limits

Significance Limit	Description
Entry:	Significant threshold for inclusion in model
Exit:	Significance cut for removal. Applies only to variables in the model at the point of further variable addition
Round Trip:	Number of entry-exit-entry-exit sequences before a variable is disqualified and excluded from selection
Keep indicator variables together:	When this option is unchecked, each category is evaluated separately and will be included or excluded individually
Print iteration details:	Prints output of each iteration of the variable selection process during training

The **Attribute Editor** provides access to modifiable variable properties. and are detailed in table 13.4.

Table 13.4: Attribute Editor Options

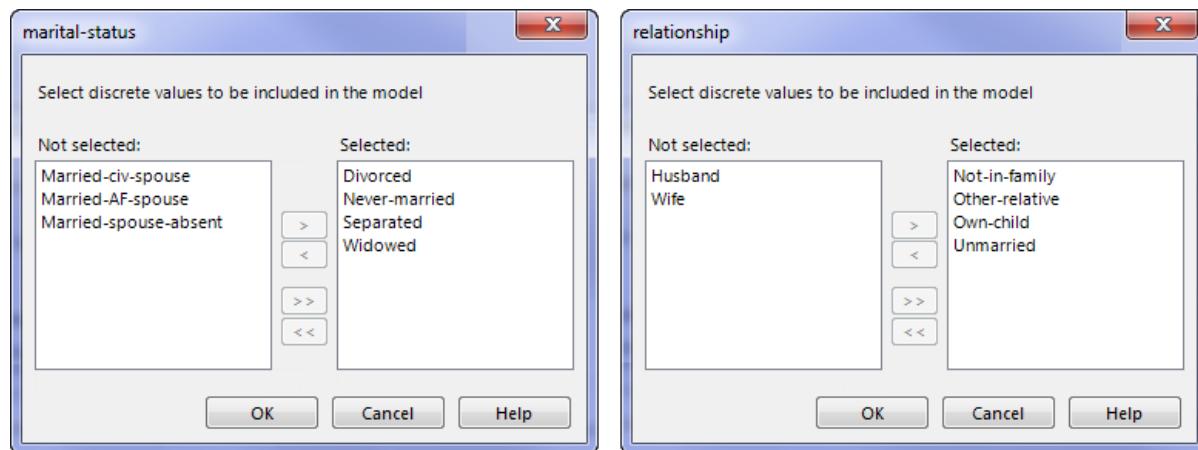
Field	Description
Variable Name	The name of the variable
Include	Determines if a variable is included in the model
Role	Determines how a variable is used in the analysis
Usage	Depending on selection, inputs are either transformed into a common range of values, usually between -1 and 1, or left unchanged, see Help files for further information
Missing values	Applies to independent variables only. Specify how missing values are handled
Dummy Variables	Available for Discrete variables only. Specify the reference category for dummy coding
Cardinality	Displays the Cardinality ; no. of unique field values
# of Missing Values	Number of missing values in field

For this demonstration, **Stepwise Selection** is chosen as the variable selection mode using the fields:

- *age*
- *sex*
- *marital_status*
- *relationship*
- *education-num*

As *marital_status* and *relationship* are categorical, reference categories must be set. This can be done by selecting the *Dummy Variables* column for each field from the *Attribute Editor*. Coding is set as illustrated in figure 13.12.

Figure 13.12: Dummy Coding



The reference category for *marital_status* is the combination of the categories:

- *Married-AF-spouse*
- *Married-civ-spouse*
- *Married-spouse-absent*

This coding has the effect of including four additional fields in the model. For the variable *relationship* the combination of the categories *Husband* and *Wife* is selected as the reference category, and again, will add four dummy coded variables.

Note also that the field *sex* is categorical. There is no need to dummy code as there are only two categories. In this example *Male* is chosen as the reference category.

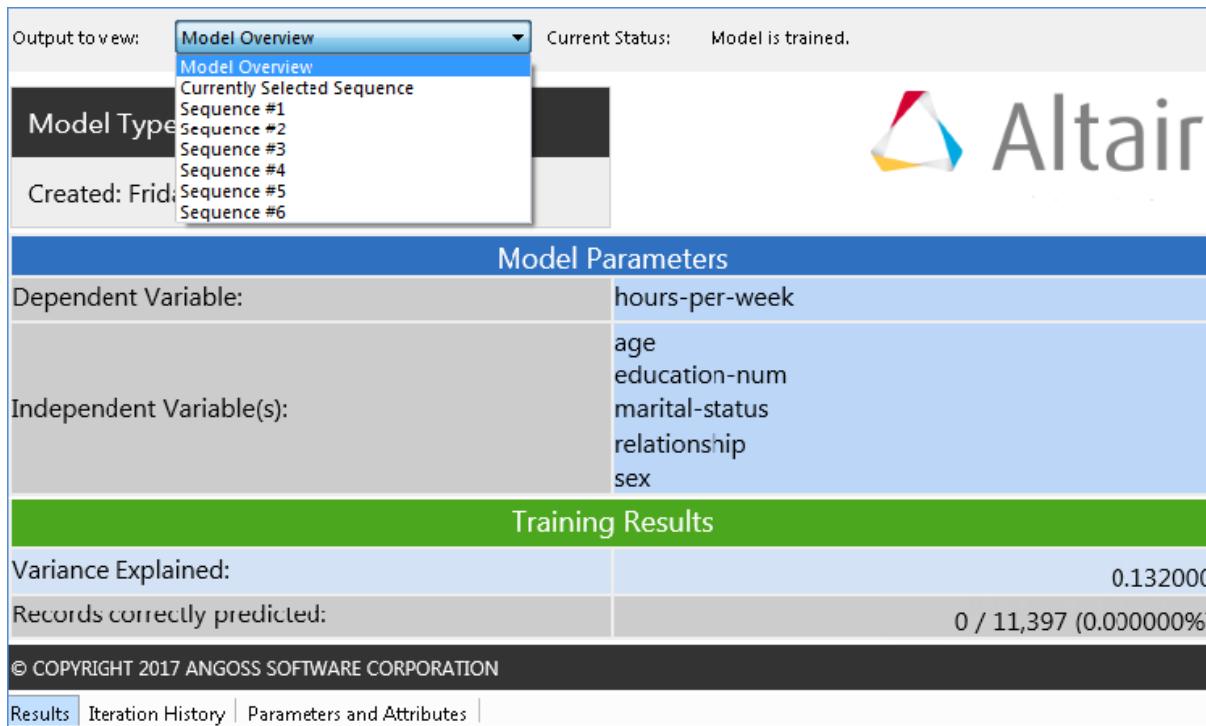
Additional options are retained at their default, these include intercept suppression and threshold settings for collinearity detection; **Singular Value Threshold**.

Click **Run** to build the model and generate results. The results are visible in the **Project Pane** nested beneath the **Development** dataset used to build the model, not shown.

13.4.3 Linear Regression Model Results

Double click model results to open.

Figure 13.13: Model Results



The screenshot shows the KnowledgeSTUDIO interface for a Linear Regression model. The top navigation bar includes 'File', 'Edit', 'View', 'Analysis', 'Tools', 'Help', and a user icon. A dropdown menu 'Output to view:' is open, showing 'Model Overview' (selected), 'Model Overview', 'Currently Selected Sequence', 'Sequence #1', 'Sequence #2', 'Sequence #3', 'Sequence #4', 'Sequence #5', and 'Sequence #6'. The status bar indicates 'Current Status: Model is trained.' The interface features a sidebar with 'Model Type' (set to 'Linear Regression') and 'Created: Friday, April 28, 2017'. The main content area is divided into sections:

- Model Parameters:**

Dependent Variable:	hours-per-week
Independent Variable(s):	age education-num marital-status relationship sex
- Training Results:**

Variance Explained:	0.132000
Records correctly predicted:	0 / 11,397 (0.000000%)

At the bottom, there are tabs for 'Results' (selected), 'Iteration History', 'Parameters and Attributes', and a footer with '© COPYRIGHT 2017 ANGOSS SOFTWARE CORPORATION'.

KnowledgeSTUDIO generates three tabs containing results: **Results**, **Iteration History**, **Parameters and Attributes**:

- **Results** Summary of model results
- **Iteration History** Displays each iteration of the variable selection process
- **Parameters and Attributes** Displays a complete summary of the model configuration

Most detail is found in the **Results** tab. The results tab contains a series of views. Views are selected using the **Output to view:** dropdown, and the default view displayed is the **Model Overview**.

Model Overview

The **Model Overview** relays summary information in relation to the model performance, records predicted and model parameters. Additional detail is provided when a **Stepwise** view is selected.

Stepwise Views

As **Stepwise Selection** was chosen as the variable selection mode, a number of models are generated. Each model is referred to as a **Sequence** and each additional **Sequence** either adds or removes a predictor. For this model there were six steps, or **Sequences**.

Each **Sequence** can be examined in turn and is accessible from the **Output to view:** dropdown. For example **Sequence #1** shows the intercept and can be ignored, **Sequence #2** shows the variable *relationship* was included first.

As this is the first variable selected and included in the model it can be said to be the most important predictor and so on, for subsequent sequences.

This is generally the case, but as predictors can also be excluded this may not be the rule. Referring to the standardised values provides a better assessment of relative importance.

The **Model Overview** is the default view and illustrates the model **R-square as Variance Explained:** and the dependent and independent variable settings in the **Model Parameters** section. Here a modest amount of variance is explained at 13.2%.

Selecting **Currently Selected Sequence** or the last **Sequence, Sequence #6** returns the same results and will provide overall model assessment and parameters for included variables.

The output in this view is formed of tables across four sections:

- **Stepwise Actions**
- **Analysis of Variance** for *hours-per-week*
- **Independent Variable Statistics**
- **Variance Inflation Factors**

The **Stepwise Actions** tables details the field included at that specific stage. As **Currently Selected Sequence** is selected, this refers to the last step in the model. At this stage *age* was included.

The **Analysis of Variance** for *hours-per-week* table provides statistics to assess overall performance

[Figure 13.14: Analysis of Variance output](#)

Analysis of Variance for hours-per-week				
Variance Explained:	0.132000			Adjusted Variance Explained: 0.131161
F-Ratio:	157.396417			F-Ratio Degrees Of Freedom 1/2: 11/11385
P-Value:	0.000000			Generalized R^2: 0.132000
				AIC: 55,840.7937
				BIC: 55,928.8870
	Sum-Of-Squares	DF		Mean-Square
Regression	232,167.668922	11		21,106.151720
Error	1,526,677.305194	11385		134.095503
Total	1,758,844.974116	11396		154.338801

Available statistics are described in table 13.5.

Table 13.5: Analysis of Variance Parameters

Statistic	Description
Variance Explained:	R-square. Proportion of variance explained
Adjusted Variance Explained:	Adjusted proportion of variance explained reflects explained variability that takes into account the complexity of the model and will always be <= Variance Explained Use adjusted figure to compare models with varying numbers of predictors included
F-Ratio & P-Value	The P-Value tests the significance of the F-Ratio The Null Hypothesis can be described in a number of ways. One being that there is no relationship between the model and the dependent variable P-Values <= 0.05 are desirable, meaning the model significantly predicts the dependent variable
F-Ratio Degrees of Freedom 1 / 2:	Degrees of freedom of the model used to calculate the F-Ratio 1 relates to the no. parameters in the model. NB: this does not count the constant, 2 = (number of records – (1 + constant)) Refer to the generated table for specific values
Generalized R2	Interpreted as R-Square coefficient
AIC/BIC	Information criteria values. Lower values are more desirable. Useful when comparing models

The model is significant; as determined by the **P-Value** being <= 0.05. The proportion of variance explained overall is modest at 13.2%. All others statistics are used when comparing models.

The second section displays **Independent Variable Statistics** for the model.

Figure 13.15: Independent Variable Statistics

Independent Variable Statistics							
hours-per-week							
Variable Name / Value	DF	Model Parameter	Parameter Standard Error	Wald Chi-Square	p-value	Standardized Parameter	Standardized Parameter Error
age	1	-0.070038	0.009964	49.412173 0.000000		-0.078348	0.011146
education-num	1	0.529809	0.043090	151.174630 0.000000		0.109405	0.008898
marital-status	Divorced	1	0.889001	0.832167	1.141256 0.285387	0.024424	0.022862
	Never-married	1	-2.198725	0.798895	7.574646 0.005919	-0.083592	0.030373
	Separated	1	1.368882	0.995673	1.890161 0.169184	0.018879	0.013732
	Widowed	1	-3.072098	1.013488	9.188246 0.002436	-0.043194	0.014250
sex	Female	1	-5.007089	0.263850	360.128987 0.000000	-0.189966	0.010010
relationship	Not-in-family	1	-0.341792	0.811316	0.177477 0.673550	-0.012053	0.028611
	Other-relative	1	-2.518011	0.907607	7.696962 0.005531	-0.036017	0.012982
	Own-child	1	-7.711184	0.851490	82.013069 0.000000	-0.224980	0.024843
	Unmarried	1	-0.048429	0.868758	0.003107 0.955545	-0.001198	0.021496
#INTERCEPT#		1	41.500635	0.642259	4,175.314062 0.000000	-	-

This table is used to assess predictor variable impact on the dependent variable; significance, magnitude and direction. Each column represented is detailed in table.

Table 13.6: Independent Variable Statistics Description

Statistic	Description
DF (Degree of Freedom):	Degrees of freedom for each of the tests of the coefficients. Can be ignored
Model parameters:	The coefficients that describe the impact each predictor has on the Dependent Variable . Assess magnitude and direction
Parameter Standard Error:	Estimate of the Standard Deviation of the coefficient Model Parameter
Wald Chi-Square: / Significance:	Both are used to assess significance of relationship between each independent variable and the dependent variable Values below .05 are preferable
Standardized Parameter: / Standardized Parameter Error	A means to assess the relative effect of each predictor and its associated error

As the **Model Parameters** column denotes the magnitude and direction of the effect of each predictor on the dependent variable, it can be seen that:

- The coefficient for *age* is negative meaning that getting older means working less hours per week
- The negative parameter for *sex* where *Male* is the reference category is also negative meaning

males work more hours per week than females

Both these variables are significant as denoted by their **Significance** values.

From the standardized parameters it can be seen that the dummy variable; *Own-child* has the greatest relative impact on the dependent variable followed by sex and *education-num*.

Some *marital_status* categories are significant, all are retained as this variable, entirely, is significant.

The intercept value of approximately 41.5 is the predicted *hours-per-week* when all other variables are set to 0. This conceptually does not make sense as it relates to someone with a value of 0 for *age* and *education-num*.

NOTE: It may be necessary to create transformed variables to aid interpretation of the constant. For example the values for these values can be replaced by a value representing difference from the average, referred to as centering, it means that a zero value for a centered variable equals the average, positive values are greater than the average and negative values, less than the average.

The final area of output is the **Variance Inflation Factors** section.

This is used to identify potential multicollinearity issues. Multicollinearity reflects relationships between pairs of independent variables. High values: <= 10, reflect highly related variables. In weaker models values above 2.5 may be a concern.

The table lists each variable and its collinearity statistic called the **VIF**. This statistic can be found in the **Value** column.

Figure 13.16: Variance Inflation Factors

Variance Inflation Factors	
Variable	Value
[age]	1.629
[education-num]	1.039
([marital-status]=='Divorced')	6.856
([marital-status]=='Never-married')	12.100
([marital-status]=='Separated')	2.473
([marital-status]=='Widowed')	2.663
([relationship]=='Not-in-family')	10.737
([relationship]=='Other-relative')	2.211
([relationship]=='Own-child')	8.095
([relationship]=='Unmarried')	6.061
([sex]=='Female')	1.314

The table reflects multicollinearity between some of the relationship and marital-status dummy coded

variables. On inspection this is hardly surprising as both are conceptually similar.

The last section of output contains the model equation. This equation is used to score new data and, although granular, can be summarised to perform quick what-if scenarios.

Figure 13.17: Model Formula

Formula(s)
[hours-per-week] = -0.07003824418612596 * [age] + 0.5298090613417833 * [education-num] + 0.8890008145636562 * ([marital-status] == 'Divorced') - 2.19872512611075 * ([marital-status] == 'Never-married') + 1.3688824467833216 * ([marital-status] == 'Separated') - 3.07209838721143 * ([marital-status] == 'Widowed') - 5.007089467938949 * ([sex] == 'Female') - 0.3417918035846998 * ([relationship] == 'Not-in-family') - 2.518011330933824 * ([relationship] == 'Other-relative') - 7.711184269245915 * ([relationship] == 'Own-child') - 0.04842850682663747 * ([relationship] == 'Unmarried') + 41.50063457892246

13.4.4 Re-running the Model

Models may be run a number of times to address issues. As some multicollinearity issues were identified, the variable *relationship* is removed and *marital_status* retained. The model is re-run using a **Stepwise** method. and partial results are illustrated in 13.18.

Figure 13.18: Final Model Parameters

Independent Variable Statistics							
hours-per-week							
Variable Name / Value	DF	Model Parameter	Parameter Standard Error	Wald Chi-Square	p-value	Standardized Parameter	Standardized Parameter Error
age	1	-0.025465	0.009860	6.670432	0.009803	-0.028486	0.011030
education-num	1	0.626087	0.043331	208.774878	0.000000	0.129287	0.008948
marital-status	Divorced	1	0.241391	0.365476	0.436236 0.508945	0.006632	0.010041
	Never-married	1	-4.934667	0.302822	265.547228 0.000000	-0.187609	0.011513
	Separated	1	0.593224	0.669267	0.785667 0.375414	0.008182	0.009230
	Widowed	1	-4.059449	0.691442	34.468556 0.000000	-0.057076	0.009722
sex	Female	1	-4.966962	0.262298	358.584376 0.000000	-0.188443	0.009951
#INTERCEPT#	1	38.475625	0.628327	3,749.726026 0.000000	-	-	-
Variance Inflation Factors							
Variable	Value						
[age]	1.540						
[education-num]	1.014						
([marital-status] == 'Divorced')	1.276						
([marital-status] == 'Never-married')	1.678						
([marital-status] == 'Separated')	1.079						
([marital-status] == 'Widowed')	1.197						
([sex] == 'Female')	1.254						

As can be seen:

- All variables are significant
- The coefficient for *age* is positive, changed since the previous model; increasing *age* means shorter hours worked
- The coefficient for *education_num* is positive; better educated, longer working week
- The variable *sex* has the greatest relative impact on the dependent variable followed
- No signs of multicollinearity, the model formula can be applied to new data

13.4.5 Model Validation, Accuracy and Residual Analysis

Validation is an essential stage of the modelling process. **KnowledgeSTUDIO** provides two methods to assess accuracy, residuals and validate **Linear Regression** models:

- **Statistical Validation** using statistics and reports
- **Business Validation** using a series of charts

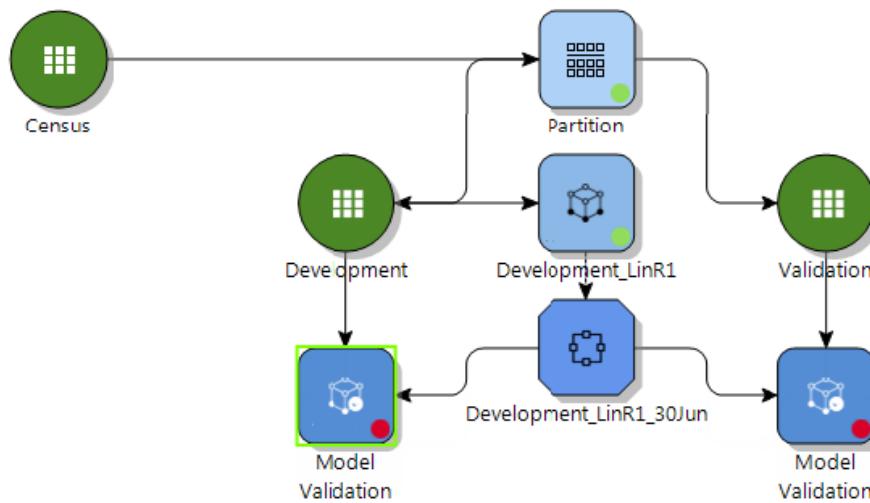
These capabilities are available through the **Model Validation** and **Model Analyser** nodes respectively.

Statistical Validation

Statistical Validation requires that data are scored. This can be applied to the **Development** partition to further evaluate the model, and to the **Validation** partition to validate the model.

To assess model performance on each partition, first, a **Model Instance** is created. Once created, add **Model Validation** nodes from the **Evaluate** palette and connect as illustrated in figure 13.19.

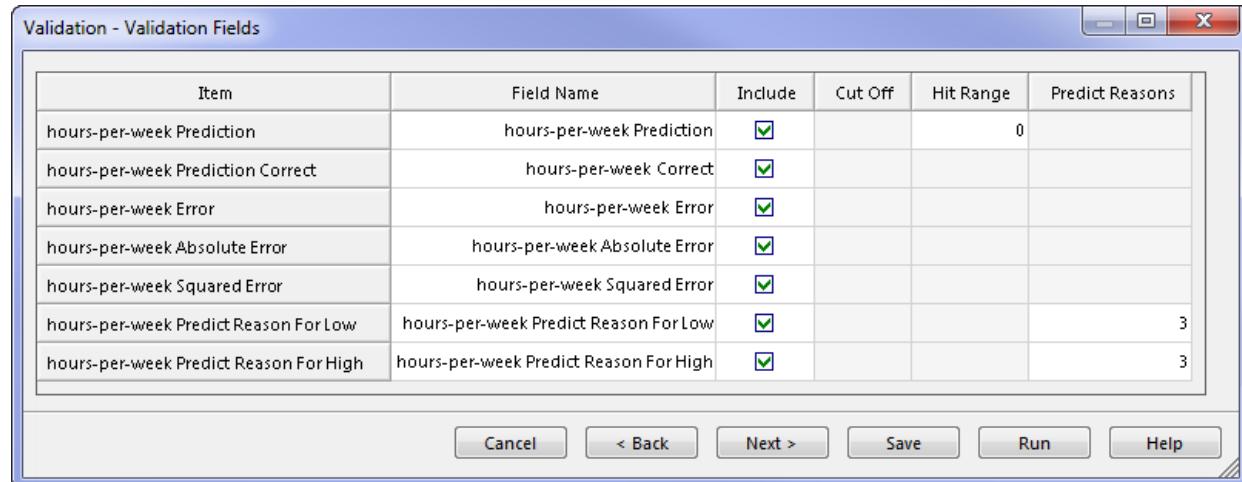
Figure 13.19: Model Validation Nodes Added



To access **Validation** options; either double click on the **Model Validation** node or right click and select **Modify**. A number of dialog screens are available providing information on connections and new fields created.

Click **Next >** until the **Validation – Validation Fields** dialog appears.

Figure 13.20: Validation Fields



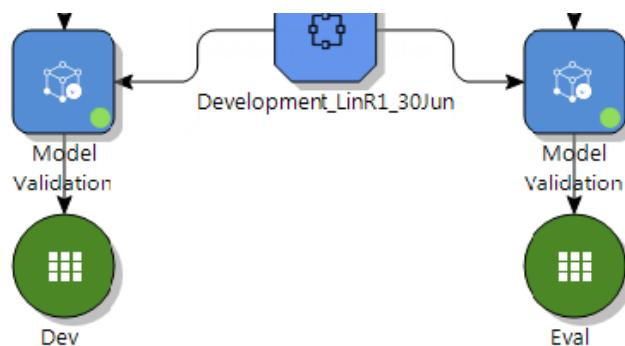
Up to seven new variables can be created. Interestingly an *hours_per_week Prediction Correct* variable can be created.

This is a dichotomous variable that is given a 1 if the prediction is within a specific user specified **Hit Range**, and 0 otherwise.

This can be used as a coarse means to assess model accuracy. The **Hit Range** value is in the same units as the dependent variable and based on +/- *hours_per_week Prediction*.

Setting the **Hit Range** to 8 and clicking **Run** scores the data and creates a new dataset. Repeat the process for the **Validation** dataset.

Figure 13.21: Development and Validation Results



To view results either double click either of the created datasets or right click and select **Open View**. Results open on the **Report** tab.

Figure 13.22: Validation Results




Validation Report	
Input Model Name: Validation_LinR1	
Input Dataset: Development	
Variable - hours-per-week	
Total Validated Records	11,397
Records Correctly Predicted	7,834
Percentage of correctly predicted	68.7374
Valid Records	11,397
Variance Explained	0.1362
Mean Deviation	7.9444
Variable - hours-per-week	
Total Validated Records	4,884
Records Correctly Predicted	3,293
Percentage of correctly predicted	67.4242
Valid Records	4,884
Variance Explained	0.1368
Mean Deviation	8.1

© COPYRIGHT 2017 ANGOSS SOFTWARE CORPORATION © COPYRIGHT 2017 ANGOSS SOFTWARE CORPORATION

Focussing on the **Development** partition results:

- Approx. 67 % – 69% of records are correctly predicted within 8 hours
- The **R-Square** statistic (**Variance Explained**) is roughly 8%

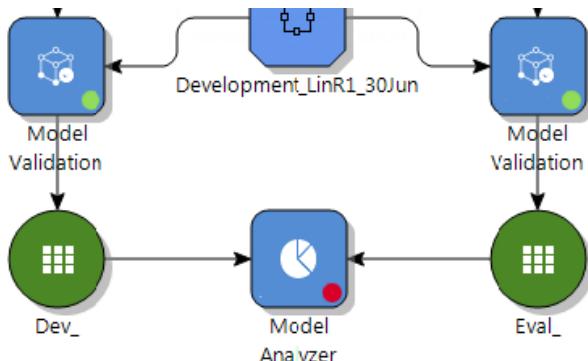
Notice that the results for the **Validation** partition are almost identical. This reflects that the mode should give similar accuracy when deployed in the population. Model stability and accuracy can be further assessed using the **Model Analyser**.

Business Validation: The Model Analyser

Business Validation helps assess the business benefit of the linear regression model. The **Model Analyser** creates a series of graphs and statistics that can be used to further validate model stability and assess residuals.

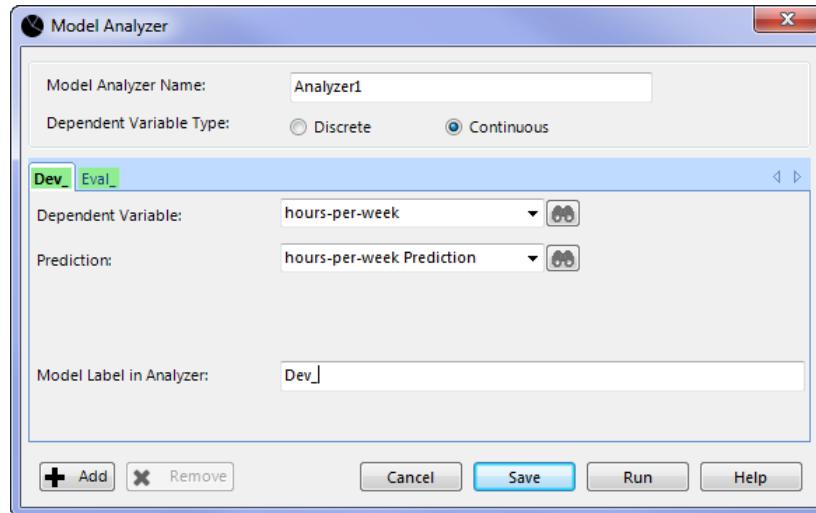
To generate the charts, drag the **Model Analyser** from the **Evaluate** palette onto the **Workflow** canvas and connect both validated datasets as illustrated in figure 13.23.

Figure 13.23: Model Analyser Added



To access the **Model Analyser** options, either double click or right click the **Model Analyser** node and select **Modify**.

Figure 13.24: Model Analyser for Continuous Dependent Variable



NOTE: The default **Dependent Variable Type** is set to **Discrete** and ensures that options are not populated as the dependent variable in this example is continuous. Changing the option to **Continuous** automatically populates options appropriately.

The **Model Analyser** display a tab for each connected dataset. For this example, two datasets are connected and therefore two tabs are visible with the **Dependent Variable** and **Prediction** field dropdowns populated.

Clicking **Run** generates results.

Results are displayed in the form of charts. For **Linear Regression** models the **Model Analyzer** supports four charts:

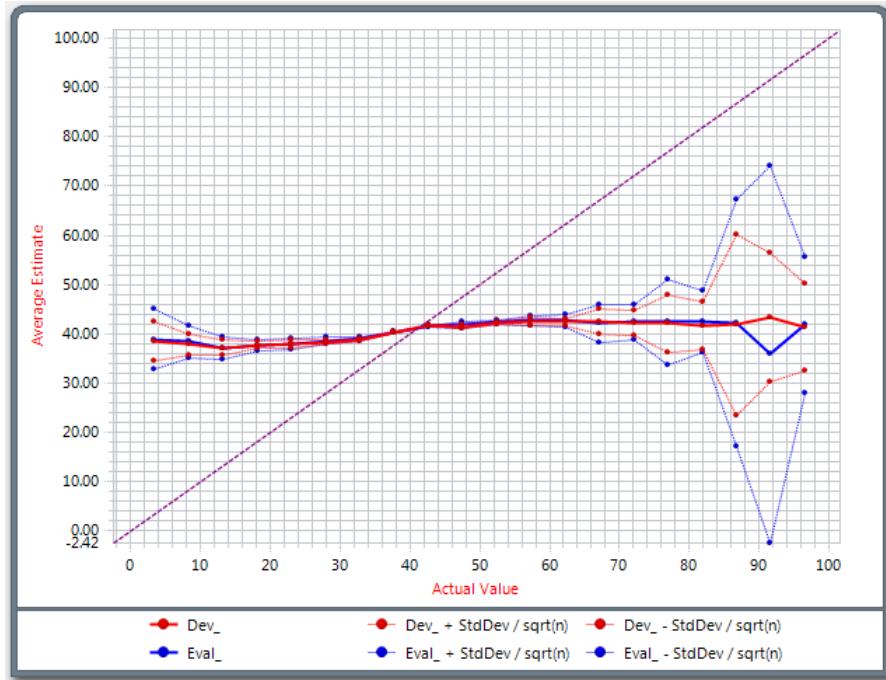
- **Bias**
- **Accuracy**
- **Error**
- **Scatter Plot**

Each chart is denoted by a separate tab, not shown. The first chart is the **Bias** chart.

Bias Chart

Bias charts provide a means to assess the variance of the residuals, along with the **Error** chart, and the accuracy of predictions. A good model exhibits values that snake around the diagonal, reflecting accurate predictions with little error.

Figure 13.25: Bias Chart



The model performance is similar across both partitions although errors are not *Homoscedastic*. The increased standard errors in higher values of number of hours worked may be a function of two few cases with these values.

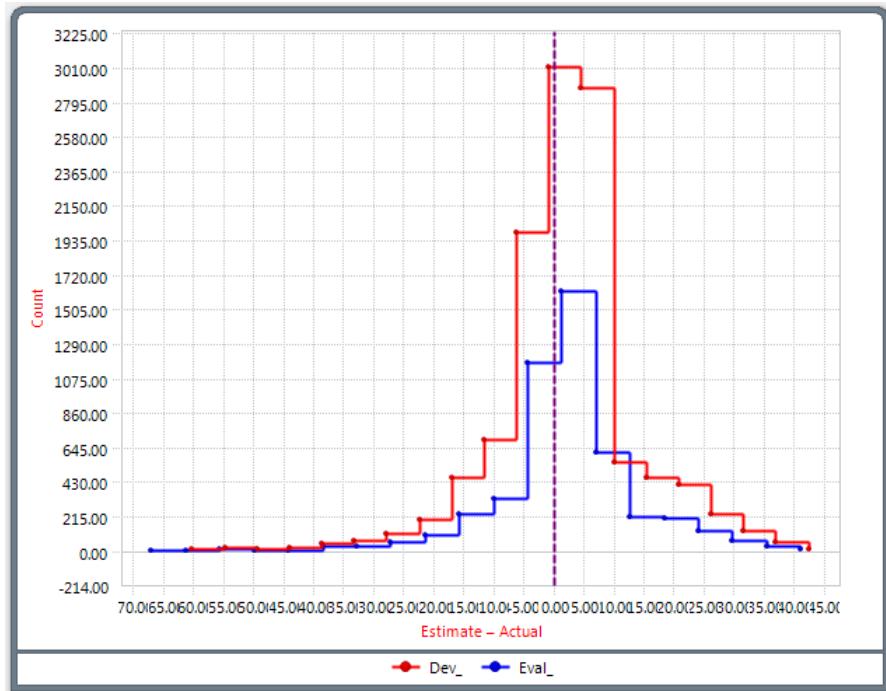
A possible modification in subsequent iterations may consider a transformation using outlier clipping to address this severe deviation.

NOTE: Modify the number of bins from the **Options** dialog. The default is 20.

Accuracy Chart

The **Accuracy Chart** plots the residuals: the difference between the estimated value of the dependent variable and its actual value, as a histogram.

Figure 13.26: Accuracy Chart



Accuracy Charts provide a means to assess whether the residuals are random. As can be seen there is a peak around 0, meaning the majority of predictions are accurate.

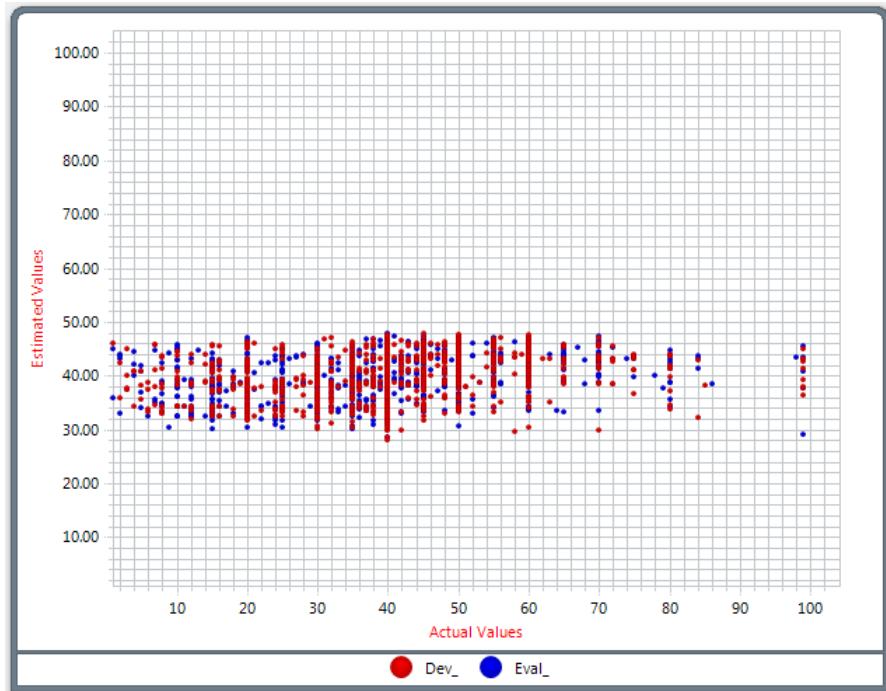
The distribution is negatively skewed meaning there is a slight tendency to overestimate the dependent variable.

All in all, the distributions appear approximately normally distributed. The difference in frequency counts for the model applied to the **Development** and **Validation** partitions is expected and correct as there are more records in the **Development** partition.

Scatter Plot

The Scatter Plot shows the Actual Values versus the Estimated Values. The Estimated Values are the predictions.

Figure 13.27: Scatter Plot



Ideally the scatterplot should jitter around the diagonal. It is clear that the distribution clusters around the predicted value of 40 with slight deviations. There are very few cases with large values for *hours_per_week*.

Given the results, the model accuracy is questionable. To be more precise, extremes; low and high values for the **Dependent Variable** *hours_per_week* are very poor predicted.

Here, the model predicts a very tight range of values between 35 – 45 and does not predict actual values outside of this range with accuracy.

Error Chart

The **Error Chart** plots the average absolute value of the difference between the predicted and actual values, in bins, against the actual values.

Figure 13.28: Error Chart



In general, a very low horizontal line is preferable reflecting small deviations across actual and predicted values. In this model inaccuracy is greatest at either side of an actual value of 40 *hours_per_week* worked.

As pointed out previously the greatest prediction error occurs at the extremes. Higher values are more inaccurately predicted than low values, again, a reflection of the small number of records in the model with higher value for hours-per-week.

These graphs reveal that the residuals, although being approximately normally distributed are not independent and the error variance is not *Homoscedastic*.

In summary, this model:

- Included four predictors: *age*, *sex*, *marital_status_2* and *education_num*
- The model **R-Square** is low, validating at 8.8%
- Although model evaluation and validation highlight a model that has a limited range of predictions and large errors, it does this consistently for both the **Development** and **Validation** partitions

NOTE: Other variables and transformations of these should be considered during the exercises in an attempt to improve the model.

13.4.6 Linear Regression Model Deployment

The final stage, model deployment can be performed by:

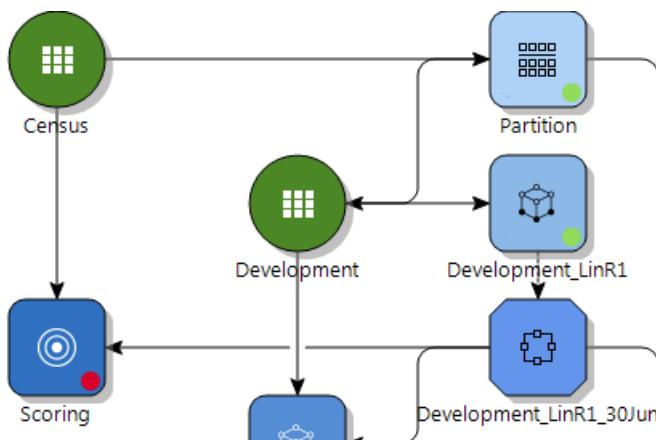
- Automatically scoring an existing dataset
- Generating code for the model
- Exporting to a file or database

Deploying using either of these methods is initiated using appropriate nodes found on the **Action** and **Connect** palettes.

Automatically Score an Existing Dataset

Scoring an existing dataset is performed using the **Scoring** node from the **Action** palette. Drag the Score node onto the **Workflow** canvas and connect the model instance and the data to score as illustrated in figure 13.29.

Figure 13.29: Score Node Added



The **Scoring** node provides default settings, model field identification and mapping and scoring fields to create. Clicking forward to the **Scoring – Scoring Fields** illustrates the minimal number of fields generated.

Figure 13.30: Scoring - Scoring Fields

Item	Field Name	Include	Cut Off	Predict Reasons
hours-per-week Prediction	hours-per-week Prediction	<input checked="" type="checkbox"/>		
hours-per-week Predict Reason For Low	hours-per-week Predict Reason For Low	<input checked="" type="checkbox"/>	3	
hours-per-week Predict Reason For High	hours-per-week Predict Reason For High	<input checked="" type="checkbox"/>	3	

Clicking **Run** creates an additional file with scoring fields added and a report on number of records scored, not shown.

Model Code Generation

Code generation for **Linear Regression** is available in four formats.

Each format can be produced by selecting the appropriate node from the **Action** palette. The available formats are listed in the table below.

Table 13.7: Linear Regression Code Formats

Node	Description
Generate LOS	Language of SAS code for the model
Generate SQL	<i>SQL</i> function that takes attributes as arguments and returns a score or prediction. Makes it easier to deploy models in a production environment. The code follows the <i>Microsoft SQL Server</i> standards. Some adjustments in <i>SQL</i> syntax may be necessary if the code is deployed in other database systems, such as Oracle
Generate XML	Generates <i>XML</i> code for the model. Equivalent to displaying the contents of the Altair file representing it
Generate PMML	<i>PMML</i> format

Figure 13.31 illustrates the process for generating the language of SAS code for the model.

Figure 13.31: Generate LOS node connected



Once connected code generation is straightforward and simply a matter of accessing the **Code Generation** dialog and clicking **Run**.

A snippet of the generated **LOS** code is illustrated in figure 13.32.

Figure 13.32: SAS code snippet

```
%MACRO KST_MODEL(DSInput, DSOOutput);
DATA &DSOutput;
SET &DSInput;

/* **** */
/* **** */
/* **** */
/* **** */
/* model summary */
/* **** */

/* dependent variables */
/* [hours-per-week] */

/* independent variables */

/* [age] -> age */
/* [education-num] -> education_num */
/* [marital-status] -> marital_status */
/* [sex] -> sex */

/* link function: linear */

/* **** */
/* **** */
/* **** */
```

Note that any **KnowledgeSTUDIO** model generated as **SAS** code is packaged in the form of a macro. The user need only supply the input and output datasets to deploy.

The generated code can be saved by selecting the **Save As...** option from the **File** menu.

When saved, the code will have an appropriate extension. For example a saved **SAS** code file will have the extension ***.sas**, which can be run within any **SAS** environment. Additionally the code can be copied and pasted.

Export to a File or Database

A model can score an existing dataset and then be exported to a variety of file formats. The available formats are available from the **Connect** palette and reflect the **Data Import** capabilities.

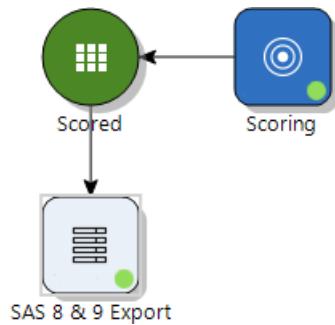
Results can be exported as:

- **Text**
- **SAS**
- **SPSS**
- **R**
- **Excel & Excel 2003**
- **ODBC ... for database export**

Exporting to any format is a matter of dragging the appropriate node onto the canvas, connecting the dataset to export and stepping through and selecting options from the available dialogs.

Once complete clicking **Run** exports to the desired format. The illustrations shows the scored data exported to **SAS** file format and the Resulting **SAS** file.

Figure 13.33: Export to SAS



13.5 Summary

This chapter looked at a fundamental statistical technique: **Linear Regression**.

The two most common forms; **Simple** and **Multiple Linear Regression** were described, developed and deployed using **KnowledgeSTUDIO** functionality.

Once developed accuracy was assess using statistics. Further assessment and residual analysis was conducted using the **Model Analyser**.

Finally, this chapter described the deployment facility of **KnowledgeSTUDIO** to automatically score data or generate code for the same purposes.

As a result of completing this chapter, users should be able to:

- Describe **Simple** and **Multiple Linear Regression**
- Develop, evaluate, validate and deploy **Linear Regression** models using **KnowledgeSTUDIO**

Exercises

The exercises continue the model building process initiated during the chapter demonstration.

Source data used is the **Census** file. This can be found by loading the **Census Sample Project** from the **Prepare Sample Data...** dialog found in the **Help** menu.

All elements can be deleted, retaining only the **Census** dataset. The *Dependent Variable* is *hours_per_week* and the aim is to model this variable using as much available data as possible.

1. Explore the data using the profiling features available in **KnowledgeSTUDIO**
 - (a) Use the **Data** tab to view some cases for each variable
 - (b) Use the **Overview** tab to assess variable summaries
 - (c) Use the **Dataset Chart** tab to further understand and qualify variable characteristics
 - (d) What do the distributions look like?
 - (e) Should any be transformed?
 - (f) The **Crosstabs** tab can be used to generate crosstabulations, scatterplots & means tables between the dependent and independent variables as well as between the independent variables.
 - (g) Are any variables linearly related to the *Dependent Variable*?
 - (h) What variable is most strongly associated with the dependent variable?
 - (i) Are there any other potentially useful predictors?
 - (j) Notice the relationship between *age* and *hours_per_week*? Is the relationship linear? Should *age* be transformed?
 - (k) Use the **Correlations** tab to further identify potential predictors.
 - (l) Use the Measures of Predictive Power facility to identify good predictors and save the top 5 – 7 as a list for ease of recall.
2. Some variables have a large number of categories. Assess whether they can be reduced meaningfully to a small number and included in the model. Once initial exploration is complete, variable characteristics noted, data preparation can proceed.
3. Create two or more dataset partitions to develop and test the model. Choose appropriate split proportions, and name the partitions **Development** and **Testing**, or any name of your choosing.
4. Of the variables identified for inclusion in the model, select only the one you deemed to be most strongly associated with the *Dependent Variable*.
5. Use the Predictive Models... to develop a linear regression model. In this instance any inclusion method is acceptable.

6. Assess the model results. What is the **R-Square** value? What is the effect of the predictor on the dependent variable? Is it significant? Does the interpretation of the constant make sense?
7. Use the equation to generate some values by hand.
8. Assess the model statistically on the **Testing** partition using the **Model Validation** node
 - (a) Become familiar with the variables created
 - (b) Input 4 for **Hit Range**. This will assess whether a prediction is within +/- 4 hours of the actual value
 - (c) What does the **Correct** variable mean, how is it calculated? What is its distribution?
9. Generate corresponding graphs using the Model Analyser
 - (a) Is the error random? Are the errors independent and of constant variance? What is the range of the errors? Does the model predict poorly in some areas as opposed to others? Why?
10. Re-run the model, including other potential predictors. Use **Stepwise Selection** as the selection mode and run the model. Make sure to select the correct reference category for any discrete variables.
11. Assess the model accuracy. What is the **R-Square** value? Assess the significance of each predictor and also the direction of the effect. Does the interpretation of the constant make sense? Should any variables be centered to address this?
12. Which variable has the greatest relative impact on the dependent variable?
13. Are there any multicollinearity issues? If so, what can be done? Remove a predictors, recode.
14. The modelling process is iterative and should lead to better ability to tweak parameters and consider modifying variables to improve the model. Explore a variety of selection modes to assess the best possible model. Making sure all coefficients are significant and make sense. Once a final model is selected, the model should be validated statistically using the **Model Analyser**.
15. Deploy the model using the **Score...** dialog from the Tools menu. Score the **Census** dataset. Notice the newly created variables
16. Explore the code generation options of **KnowledgeSTUDIO**

Chapter 14: Logistic Regression

14.1 Introduction

Logistic Regression is a modelling technique used for predicting the outcome of a categorical dependent variable. Frequently the Dependent Variable, is binary with two categories, for example

- 0, 1
- Bad, Good
- Yes, No

For example, a gold card holder can be represented as 1, and a non-gold card holder can be represented by 0. In this case, a **Logistic Regression** will provide an estimate of the probability that a new customer will become a gold card holder.

This type of regression is called **Binary Logistic Regression** as there are two categories of the **Dependent Variable**. Problems with more than two dependent variable categories, for example *low, medium, high*, can be dealt with using **Multinomial Logistic Regression**, which, although available in **KnowledgeSTUDIO**, is beyond the scope of this chapter.

As a prediction model **Logistic Regression** can be utilised in a number of business applications including: customer acquisition, next product recommendation, customer churn/retention, up-sell/cross-sell, building credit scorecards etc.

Binary Logistic Regression has the following characteristics:

- Does not assume linearity of relationship between the *Independent* and *Dependent Variables*
- Does not require normally distributed variables
- The *Dependent Variable*, can be categorical; yes, no, or numeric; 0, 1, but can have only two categories
- The *Independent Variable(s)*, can be discrete, continuous or a mixture of both
- **Logistic Regression** treats categorical independent variables in the same fashion as **Linear Regression**

This chapter describes **Binary Logistic Regression** and introduces the **Logistic** regression capabilities of **KnowledgeSTUDIO**; how to effectively create, test, modify and deploy **Logistic Regression** models.

On completion of this chapter and accompanying exercises the reader should be able to:

- Understand **logistic Regression** and its applicability
- Build **logistic Regression** models using **KnowledgeSTUDIO**
- Analyze and interpret the model outputs
- Validate the model from statistical and business perspective
- Apply and deploy logistic regression models

14.2 Description

Logistic Regression is a type of model that predicts an outcome for a discrete dependent variable. For **Binary Logistic Regression** the dependent variable can only have two categories.

Output from a **logistic regression** is in the form of log odds. These can be easily converted to odds and probabilities.

The **logistic regression** equation is an extension of the **Linear Regression** equation. A **Linear Regression** model cannot be applied in cases where the dependent variable is categorical, as this violates one of its primary assumptions.

Recall: **Linear Regression** assumes a linear association between the predictors and the dependent variable.

Making a similar assumption wouldn't make sense for a binary dependent variable, the type of variable that **Logistic Regression** aims to model. This is because in a linear model, small changes in the independent variables will lead to small changes in the dependent variable, but small changes in a binary (dependent) variable aren't possible. Instead, **Logistic Regression** assumes that the natural log of the odds of the dependent variable is linearly related to the independent variables. Therefore, the linear regression equation for one **Independent Variable** can be modified:

$$\ln(\text{odds}(y)) = \ln\left(\frac{p}{1-p}\right) = a + b_1x$$

The equation can be further extended to incorporate additional independent variables:

$$\ln(\text{odds}(y)) = \ln\left(\frac{p}{1-p}\right) = a + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

The left hand side of the equation outputs an odds, and these odds can easily be converted back to a probability of a specific outcome for the **Dependent Variable**.

14.2.1 Logistic Regression Example

Let's assume we have a **Logistic Regression** model based on one **Independent Variable**, *Income*, with three categories: *Low*, *Medium* and *High*.

The **Dependent Variable**, *Response*, has two values: *Yes* and *No*, representing whether someone responded or not to a previous marketing campaign.

NOTE: Prior to applying **Logistic Regression** it is necessary to test the hypothesis that *Income* has an effect on *Response*. The hypothesis can be tested using a non-parametric statistical test such as the **Chi-square** test. If the test result shows that *Income* does affect *Response*, **Logistic Regression** can be applied to model the relationship and calculate the probability of a *Yes* or a *No*, given the *Income* variable values.

The example below demonstrates the modelling process.

Figure 14.1: Crosstabulation

		Response	
		Yes	No
Income	Low	4	1
	Medium	3	2
	High	3	1

In order to build a **Logistic Regression** model, it is necessary to calculate the following:

- Yes and No probabilities for each *Income* state; *Low*, *Medium* and *High*
- The odds, e.g. the ratio of Yes and No probabilities
- The corresponding natural logarithm function

Results are illustrated in figure 14.2.

Figure 14.2: Logistic Regression Calculations

IV[cat]	→	DV[cat]	Outcome	Probability of Outcome	Odds	Natural Odds
						In[Odds]
Low		Yes		4/5 = 0.8	0.8/0.2 = 4	1.4
Low		No		1/5 = 0.2	0.2/0.8 = 1/4	-1.4
Medium		Yes		4/6 = 0.67	0.67/0.33 = 2	0.7
Medium		No		2/6 = 0.33	0.33/0.67 = 1/2	-0.7
High		Yes		1/4 = 0.25	0.25/0.75 = 1/3	-1.1
High		No		3/4 = 0.75	0.75/0.25 = 3	1.1

The **Odds** are always positive, which is not ideal for a dependent variable modeled by a linear relationship. Taking the logarithm produces a quantity which has no numeric restrictions (it can vary between negative and positive infinity), a much better quantity to model.

The **Dependent Variable** in a **Logistic Regression** is the natural log of the odds; $\ln(\frac{p}{1-p})$. This equation is known as the **Link** function, also referred to as a *logit*.

At first glance, **Logistic Regression** may look familiar, especially to those users of **Linear Regression**: there is a regression equation, complete with coefficients for all the variables.

However, these regress against the *logit*, not the **Dependent Variable** itself. To find the probability of a *Yes*, the *logit* must be converted.

The initial **Logistic Regression** equation generates the In odds, the natural log of the odds:

$$\ln(\text{Odds}(y)) = a + b_1x$$

To convert to odds, the \ln from the left hand side is removed giving:

$$\text{Odds}(y) = e^{a+b_1x}$$

This results in the odds; to convert to a probability is simple as the relationship between odds and probabilities is known:

$$P = \frac{\text{Odds}}{1 + \text{Odds}}$$

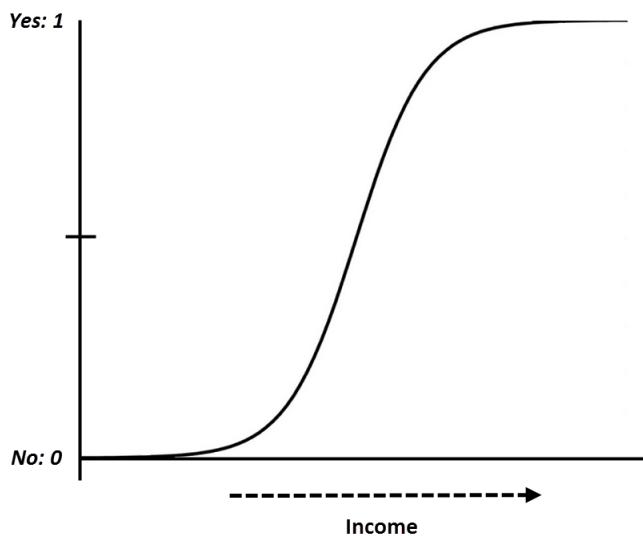
Substituting the odds from the **Logistic Regression** equation gives:

$$P = \frac{e^{a+b_1x}}{1 + e^{a+b_1x}}$$

This final equation is an **S-shaped** function called the **Sigmoid function**, where the **Independent Variable** is *income* and the **Dependent Variable** is the probability of responding to the campaign.

NOTE: This can be written in short form as: $\frac{e^X}{1+e^X}$, where X is the regression equation.

Figure 14.3: The Sigmoid Function, representing Yes and No probabilities



14.2.2 Steps when Developing Logistic Regression Models

Logistic Regression is a primary statistical technique and although the mathematics remain constant, the output and ways to assess data and include independent variables have changed over the years.

The steps below outline the general process when developing models with **KnowledgeSTUDIO**.

- Data Exploration
- Data Preparation
- Modelling
- Validation
- Deployment

14.3 Logistic Regression in KnowledgeSTUDIO

This section explains the process of creating, analyzing, validating and deploying a **Logistic Regression** model using **KnowledgeSTUDIO**. The file: *Census.xlsx* is used throughout.

Initiate a new project and import the file **Census.xlsx** as illustrated. Once complete open the **Census** dataset to explore the data.

From the **Overview Report** tab; the dataset contains 14 variables and 16281 cases and comprises demographics such as *age*, *education*, *sex*, some financial variables such as *capital-gain*, *capital-loss* and *income*.

There is interest in modelling the variable *Response*; a binary variable with two categories: *No* and *Yes*. This variable comes from a marketing department and represents whether someone responded to a previous marketing campaign or not.

The distribution across its categories can be viewed from the **Dataset Chart** tab

It is assumed data exploration and **Independent Variable** selection has been conducted and will not be stepped through in this demonstration, however, to recap, data exploration steps should include:

- Calculation of univariate statistics and graphs for all fields
- Derivation of new variables if necessary
- Identifying candidates for model inclusion. This step can be assessed using all exploratory methods including the **Segment Viewer**, **Characteristic Analysis**, **Measures of Predictive Power** and **KnowledgeSTUDIO Decision Trees** to visualise relationships

NOTE: Once an adequate set of predictors has been identified, create a list using the **Variable List** option. The list can be easily modified and recalled at the point of modelling.

14.3.1 Data Partitioning

Prior to modelling and given an adequate sample size, it is advisable to create partitions to develop and further test the validity of any model.

Partitioning ensures a rigorous validation process by creating independent datasets used exclusively for model testing. For this demonstration, two partitions are created; **Development** and **Validation** using a 70/30 split respectively.

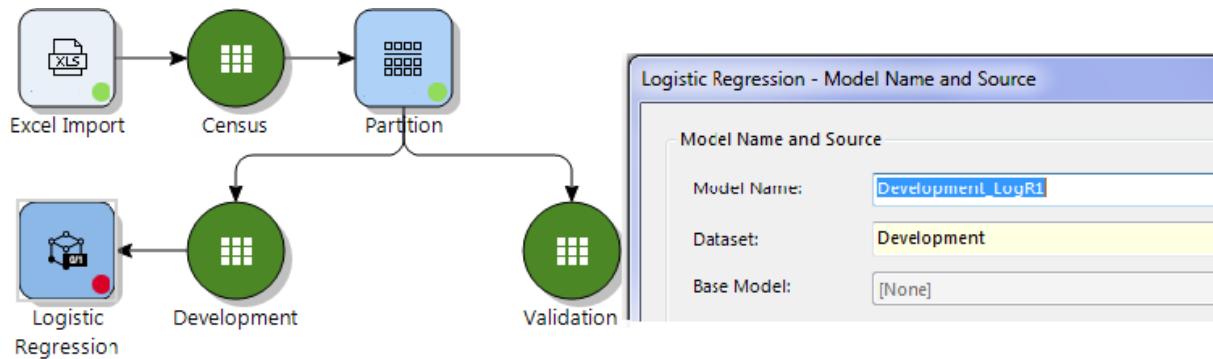
Once the partitions have been added the logistic regression model can be developed.

14.3.2 Building the Logistic Regression Model in KnowledgeSTUDIO

The **KnowledgeSTUDIO Logistic Regression** node can be found in the **Model** palette. Drag the **Logistic Regression** node to the canvas and connect the **Development** partition.

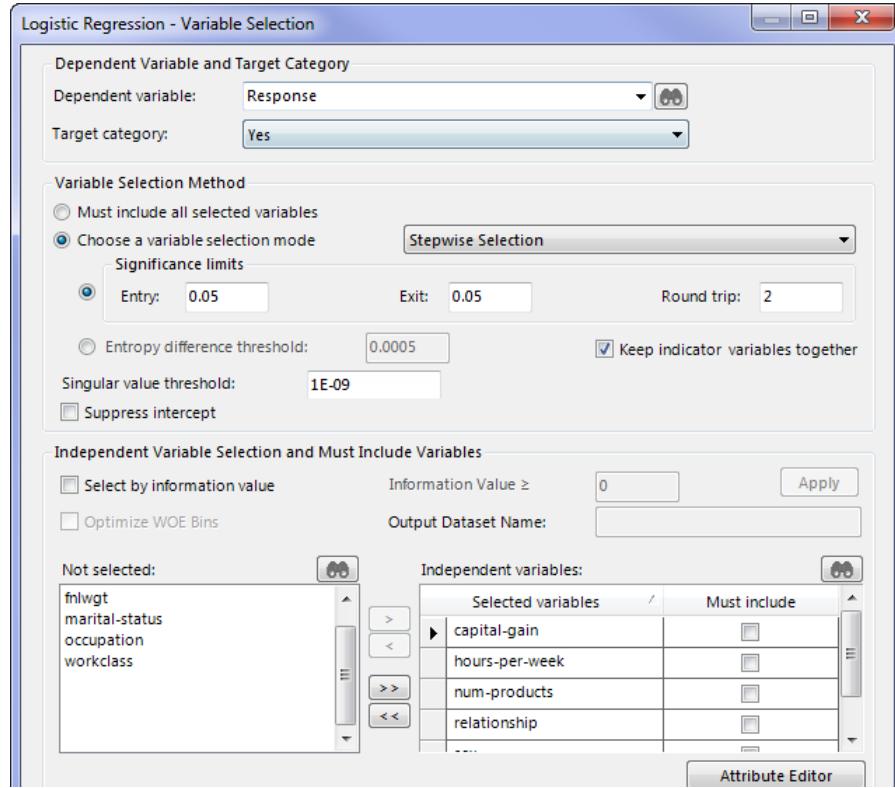
Open the **Logistic Regression** node to set options. The first dialog is: **Logistic Regression – Model Name and Source**.

Figure 14.4: Logistic Regression - Model Name and Source



This dialog page reflects connections and a modifiable model name. Clicking **Next >** opens the **Logistic Regression – Variable Selection** dialog.

Figure 14.5: Logistic Regression - Variable Selection



This dialog has three distinct areas:

- **Dependent Variable and Target Category**
 - Provides options to specify the **Dependent Variable** and select the **Target Category**
- **Variable Selection Method**
 - Options for including variables in the model
- **Independent Variable Selection and Must Include Variables**
 - Candidate variable specification, including force options and **Attribute Editor**
 - The **Attribute Editor** enables access to modifiable variable properties including missing value treatment and reference category selection for categorical variables

Variable Selection Methods are identical to those available for **Linear Regression**. Variable selection is determined via **Significance Limits** or an **Entropy Difference Threshold**. Both can be accepted at their default or user defined.

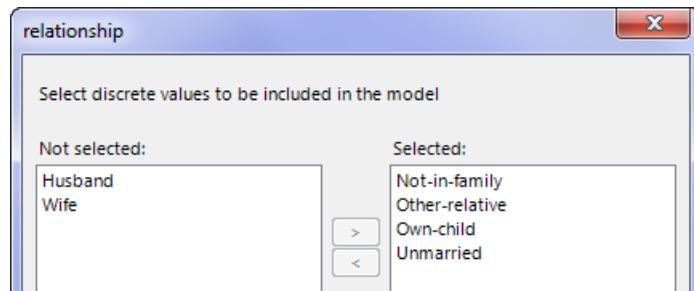
In this example, the objective is to predict customers who are more likely to respond to a direct mailing campaign. The **Dependent Variable** is *Response*, with a target category of *Yes*. The variable selection method is **Stepwise** and predictors used are:

- *capital_gain*

- *hours_per_week*
- *relationship*
- *sex*
- *num-products*

As *relationship* and *sex* are categorical, reference categories must be set. This can be accomplished by selecting the **Dummy Variables** column for each field from the **Attribute Editor**. Default coding is accepted for *sex*. For the variable *relationship* **Husband** and **Wife** are set as the reference category as illustrated in figure 14.6.

Figure 14.6: Logistic Regression Dummy Coding



All other options are set to their default. Click **Next >** to go to the **Logistic Regression – Model Fitting Parameters and Optimizers** dialog. This dialog contains two areas to specify model options:

- **Model Fitting Parameters**
- **Model Optimizers**

These areas are detailed in tables 14.1

Table 14.1: Logistic Regression - Model Configuration Options

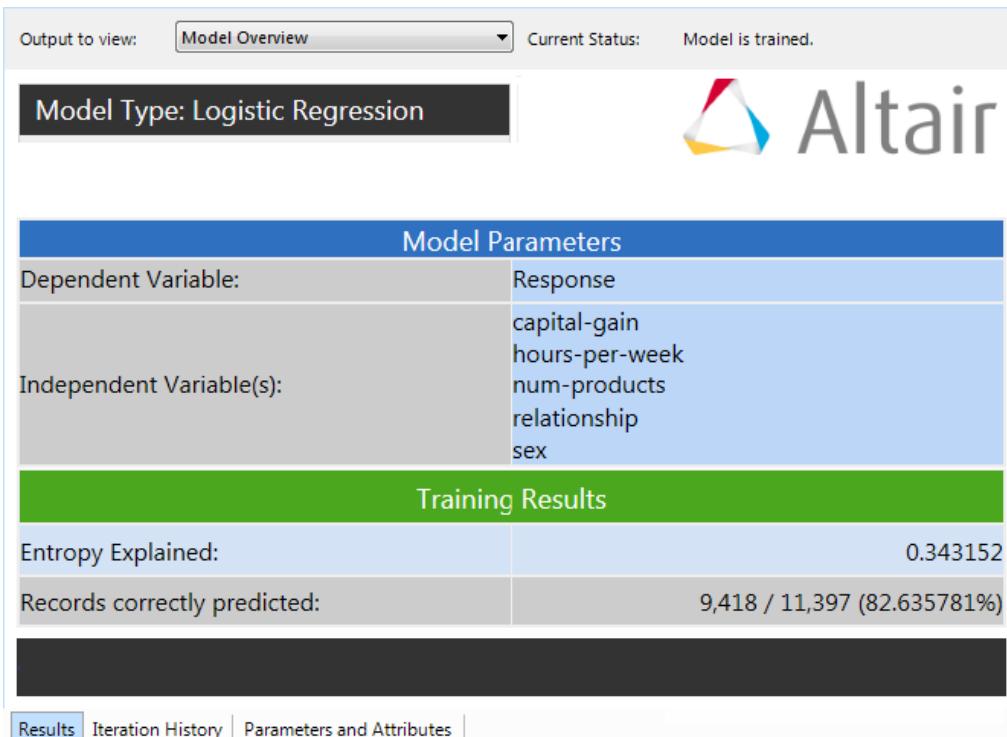
Area	Element	Description
Model Fitting Parameters	Link Function	Non-linear transformation applied to predicted values so to accommodate a specific exponential distribution Options available are Logistic (the default), Inverse Cumulative Normal , Complementary Log-Log , and Log-Log
	Network Configuration	Not applicable to Logistic Regression
	Maximum iterations	No. times model iterates to calculate coefficients
	Compute Errors	Standard Errors are computed and reported (this should almost never be disabled for Logistic Regression)

Model Optimizers	Select an Optimizer	Select a model optimizer for calculating coefficients. Default: Levenberg Marquardt
	Grad epsilon	Convergence criteria for slope
	Line epsilon	Similar to Grad epsilon but used by line optimizer

14.3.3 Logistic Regression Model Results

Once options have been specified click **Run** to generate the model. Either double click the model results in the **Project Pane** or right click the generating node on the **Workflow** canvas and select **Open View** to access results.

Figure 14.7: Logistic Regression Results Tab



The screenshot shows the 'Results' tab of the KnowledgeSTUDIO interface for a Logistic Regression model. At the top, there is a header bar with 'Output to view:' dropdown set to 'Model Overview', 'Current Status:' showing 'Model is trained.', and the Altair logo. Below the header, the 'Model Type: Logistic Regression' is displayed. The main area is divided into sections: 'Model Parameters' and 'Training Results'. In 'Model Parameters', the 'Dependent Variable:' is 'Response' and the 'Independent Variable(s):' are 'capital-gain', 'hours-per-week', 'num-products', 'relationship', and 'sex'. In 'Training Results', the 'Entropy Explained:' is 0.343152 and the 'Records correctly predicted:' is 9,418 / 11,397 (82.635781%). At the bottom, there are three tabs: 'Results' (selected), 'Iteration History', and 'Parameters and Attributes'.

KnowledgeSTUDIO generates three tabs containing results: **Results**, **Iteration History**, **Parameters and Attributes**:

- **Results** Summary of model parameters and training results
- **Iteration History** Displays each iteration of the variable selection process
- **Parameters and Attributes** Displays a complete summary of the model configuration

Most detail is found in the **Results** tab. The default view displayed via the **Output to view** dropdown is

the **Model Overview**.

Model Overview

The **Model Overview** relays summary information in relation to model performance, records predicted and model parameters. Additional detail is provided when a **Stepwise** view is selected.

Stepwise Views

Stepwise Selection potentially generates a number of models. Each model is referred to as a sequence and each additional sequence either adds or removes a predictor. Here there are six sequences.

Each sequence can be examined in turn and is accessible from the **Output to view:** dropdown. For example **Sequence #1** shows the intercept, **Sequence #2** shows *relationship* was included first.

The first variable selected is the most important predictor, and so on, for subsequent sequences. This is generally the case, but as predictors can be excluded this may not be the rule. Refer to standardised values to determine relative importance. Model results can be examined in more detail by viewing the **Currently Selected Sequence** view.

Figure 14.8: Logistic Regression Model Output

Stepwise Actions						
Variable	Action	Score	Wald	Separated	Collinearity	Convergence
[capital-gain]	Entered	0.000000	0.000000	-	-	-
Model Fitting Summary for Response						
Chi-Square: 4,290.338019 P-Value: 0.000000 Entropy Explained: 0.343152				Chi-Square Degrees Of Freedom: 7 Generalized R^2: 0.313703 AIC: 8,228.390309 BIC: 8,287.119153		
Percent Concordant: 87.035329 Percent Discordant: 11.878172 Percent Tied: 1.086499 Total Pairs: 23,541,770.000000				Somer's D: 0.751572 Gamma: 0.759827 tau a: 0.272456 c: 0.875786		
Global Null Hypothesis Testing (BETA=0)						
Statistic		Chi-Square	DF	p-value	Negative 2(Log-Likelihood)	
Likelihood Ratio		4,290.338019	7	0.000000	Null Model	12,502.728329
Score		3,513.057831	7	0.000000	Full Model	8,212.390309
Wald		2,118.705216	7	0.000000		7
Residual Chi Square		0.103897	1	0.747203		
Independent Variable Statistics						
Response = Yes						
Variable Name / Value	DF	Model Parameter	Parameter Standard Error	Wald Chi-Square	p-value	-95%
relationship						Odds
Not-in-family	1	-2.267636	0.075206	909.168904	0.000000	0.089000
Other-relative	1	-3.411028	0.370469	84.774886	0.000000	0.016000
Own-child	1	-3.630402	0.195807	343.759283	0.000000	0.018000
Unmarried	1	-2.697536	0.142174	359.994238	0.000000	0.051000
hours-per-week	1	0.025134	0.002350	114.392288	0.000000	1.021000
num-products	1	0.343171	0.012431	762.143745	0.000000	1.375000
capital-gain	1	0.003771	0.000316	142.156557	0.000000	1.003000
#INTERCEPT#	1	-7.637768	0.285561	715.376158	0.000000	<0.001000
					<0.001000	<0.001000
						-
Variance Inflation Factors						
Variable	Value					
(relationship) == 'Not-in-family'	1.183					
(relationship) == 'Other-relative'	1.052					
(relationship) == 'Own-child'	1.248					
(relationship) == 'Unmarried'	1.121					
[hours-per-week]	1.107					
[num-products]	1.047					
[capital-gain]	1.017					
Formula(s)						
Z0 = -2.2676361872657953 * ([relationship] == 'Not-in-family') - 3.4110277320809046 * ([relationship] == 'Other-relative') - 3.6304017700373565 * [relationship] == 'Own-child') - 2.6975361848372366 * ([relationship] == 'Unmarried') + 0.025133860159173543 * [hours-per-week] + 0.34317138656224894 * [num-products] + 0.003771432384381691 * [capital-gain] - 7.637768292218533						
Probability([Response] == 'Yes') = $\frac{\exp(Z0)}{1 + \exp(Z0)}$	$\exp(Z0)/(1 + \exp(Z0))$					

The **Currently Selected Sequence** sections are:

- **Model Fitting Summary For [dependent variable]**
- **Global Null Hypothesis Testing**
- **Independent Variables Statistics**
- **Variance Inflation Factors**
- **Formula(s)** equations of the model written in *SQL* notation

The **Model Fitting Summary For [dependent variable]** table details statics used to assess overall model performance. Each statistic is described in table 14.2.

Table 14.2: Model Fitting Summary – Statistical Parameters

Parameter	Description
Chi-Square	<p>The Likelihood Ratio Chi-Square test, tests the model as a whole; that at least one of the predictors' regression coefficients is not zero</p> <p>The value is calculated as the difference between deviances, i.e. Log-Likelihood, L, of the NULL model, the intercept, and the FULL model:</p> $\text{Chi-Square} = -2*L(\text{INTERCEPT model}) - (-2*L(\text{FULL model}))$ <p>In this example the Chi-square is equal to $12,502.73 - 8,204.83 = 4,297.90$</p>
P-value	Probability of the Chi-Square statistic. Used to assess model significance on the whole. Values less than 0.05 are desirable
Entropy Explained	Entropy Explained assesses goodness of fit and is often referred as a Pseudo-R² statistic. Values closer to 1 desirable
Percent Concordant	<p>Looks at all possible pairs of actual and predicted observations</p> <p>A pair is concordant if the observation with the larger value of X also has the larger value of Y</p> <p>A pair is discordant if the observation with the larger value of X has the smaller value of Y; here, X and Y are predicted and actuals values respectively</p>
Percent Discordant	Opposite of Concordant Concordant
Percent Tied	Percent of pairs tied; note that (Percent Concordant + Percent Discordant + Percent Tied) = 1
Total Pairs	All possible combination of pairs of actual and predicted values
Chi-Square Degrees of Freedom	Defined by the number of predictors in the model NOTE: each class of a categorical variable is considered as a predictor
Generalized R²	Model fit statistic
Somer's D, Gamma, Tau a, c	Measures of association between model and dependent variable, values close to zero reflect little or no relationship
AIC and BIC	<p>AIC, Akaike's Information Criterion and BIC, Bayesian Information Criterion, are useful for comparing with other regression models</p> <p>When comparing models lower AIC and BIC statistics indicate a better fit</p>

The Global Null Hypothesis Testing table provides information used to determine whether the null

hypothesis that there is no relationship between the dependent variable and the model is rejected.

This table generates three statistics with associated **Chi-Square**, degrees of freedom and **p-values**.

Also provided are **-2LogLikelihood** statistics for the intercept or **Null**, and saturated or **Full**, model.

These aspects are described in the table below.

Table 14.3: -2LogLikelihood Table Parameters

Parameter	Description
Null Model	Describes a model with no predictor variables and simply fits an intercept to predict the outcome variable
Full Model	Describes a model that includes the specified predictor variables and has been arrived at through an iterative process that maximizes the log likelihood of the outcomes seen in the outcome variable By including the predictor variables and maximizing the log likelihood of the outcomes seen in the data, the Final model should improve upon the Intercept Only model by having higher Entropy Explained and Chi-Square values
Negative 2(Log Likelihood)	The product of -2 time the log likelihoods of the Null , intercept only, model and fitted Final model The likelihood of the model is used to test of whether all predictors' regression coefficients in the model are simultaneously zero

The **Independent Variable Statistics** table displays statistics are used to assess variables in the model in terms of their significance and direction.

Available statistics are detailed in table 14.4.

Table 14.4: Independent Variable Statistics – Parameters

Parameter	Description
DF	The Degrees of Freedom for each of the tests of the coefficients For each parameter estimated in the model, one DF is required. DF defines the Chi-Square distribution used to assess each model coefficient for significance
Model Parameters	The coefficients that describe the size of the effect a predictor variable is having on the dependent variable

Parameter Standard Error	An estimate of the Standard Deviation of the coefficient
Wald Chi-Square	Tests that the predictor regression coefficient is not equal to zero in the model. As it is calculated as: $(Modelparameters / ParameterStandardError)^2$ it can be used to test the true value of the model parameter
Significance	Tests whether parameters are significantly different from 0 If a parameter is close to 0, and significance is > 0.05 then the predictor has no effect in the model
Standardized Parameter	Coefficients obtained by standardising the predictors and the outcome variables As a consequence of standardizing, coefficients can be directly compared using the standardized regression coefficient and relative impact can be assessed
Standardized Parameter Error	Use to generate confidence intervals for estimated parameters
Confidence Interval (CI)	Confidence intervals calculated using Standardized Parameter Error
Odds Ratio	The odds ratios are exponentiated Model Parameters and show how a one unit increase in the independent variable affects the odds of being in the selected category of the Dependent Variable

The **Variance Inflation Factors (VIF)** section displays statistics used to detect multicollinearity or associations between the **Independent Variables** in the model.

A good model will have *VIF* values for each independent variable close to 1.

Finally the **Formula(s)** section provides the model equation in *SQL* notation.

Interpreting Output Results

From the **Model Fitting Summary for Response** table the statistics to focus on are:

- **P-Value**
- **Entropy Explained**
- **Generalized R2**

By convention, we usually assume that if the **p-value** is less than 0.05 than the model is (statistically) significant.

The model explains somewhere between 31.4% - 34.3% of the **Dependent Variable** and there is a positive relationship in general between the model and the **Dependent Variable** outcome; higher model values reflect more the target category, in this instance, the Yes category.

The **Global Null Hypothesis Testing** table reflects the **p-value** from the previous section. When all **p-value** statistics are less than 0.05, the null hypothesis of no significant relationship between the model and the **dependent variable** can be rejected, therefore the model statistically significantly predicts the **Dependent Variable**.

The **Independent Variable Statistics** table enables assessment of variables included in the model and their impact on the **Dependent Variable**.

For example the **Model Parameter** for *num_products* is positive meaning the higher the number of products purchased the more likely to respond to the marketing campaign, i.e. *Response* = Yes.

By calculating the odds (manually) things can be quantified a little further to say that increasing products purchased will increase the odds of response by 1.4.

There are some negative **Model Parameters**. These have the inverse effect of positive values meaning that increasing the values of these variables decreases the odds of *Response*.

From the **Standardized Parameters** it can be seen that the capital-gain has the greatest relative impact on the dependent variable followed by the dummy variable *Own_Child* and *Not_in_family*.

All included variables and dummy coded fields are significant as reflected in the **Significance** column.

As **Stepwise Selection** was chosen, only significant predictors are included in the model. However if a field is categorical and contains dummy coded equivalents, not all may be significant.

As the variable is assessed on the whole, and if some dummy codes are significant and others are not, but the whole variable is significant, then all dummy coded fields are included for that variable.

The **Variance Inflation Factors (VIF)** table reflects a stable model as all values are close to 1.

When to Re-run a Logistic Regression Model

Re-running the logistic regression model might be necessary when:

- There are insignificant variables in the model i.e. significance is greater than 0.05
- There is multicollinearity between predictors, assessed from **Variance Inflation Factors** table
- The number of dummy variables for categorical variables is too large
- Coefficients are unexplainable; the sign of parameter coefficients contradicts the actual data, showing the increase in the dependent variable, whilst the actual data shows decrease in the dependent variable. This effect could be due to confounding variables or other interactions

NOTE: Using **KnowledgeSTUDIO Decision Trees** is an easy means to assess and illustrate relationship.

Once a satisfactory model has been arrived at the next steps are to further assess model accuracy using statistical and business validation methods.

14.3.4 Logistic Regression Model Validation

KnowledgeSTUDIO provides two methods to further assess accuracy, and validate **Logistic Regression Models**:

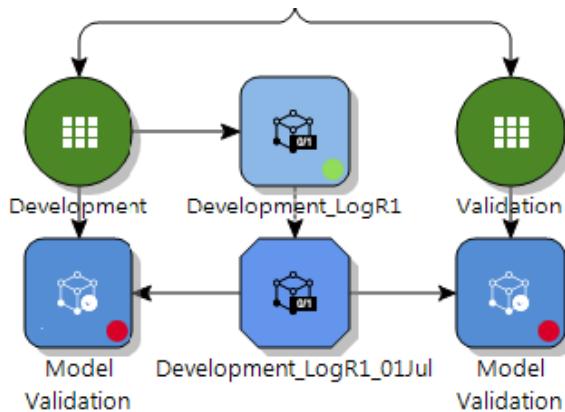
- **Statistical Validation** Using statistics and reports
- **Business Validation** Using a series of charts

Statistical Validation

Statistical validation requires the data to be scored, creating new variables. Statistical validation can be applied to both the **Development** partition, to further evaluate the model, and to the **Validation partition** to validate the model.

First a model instance is created. Once created, add **Model Validation** nodes from the **Evaluate** palette and connect as illustrated in figure 14.9.

Figure 14.9: Model Validation Nodes Added



From the **Model Validation** node, a number of dialog screens are available providing information on connections and new fields created.

Click **Next** until the **Validation – Validation Fields** dialog appears.

Figure 14.10: Validation Fields

Item	Field Name	Include	Cut Off	Hit Range	Predict Reasons
Response Prediction	Response Prediction	<input checked="" type="checkbox"/>			
Response Probability of Prediction	Response Predict Prob	<input checked="" type="checkbox"/>			
Response No Probability	Response No Prob	<input checked="" type="checkbox"/>	0.5		
Response No Predict Reason	Response No Predict Reason	<input checked="" type="checkbox"/>		3	
Response Yes Probability	Response Yes Prob	<input checked="" type="checkbox"/>	0.5		
Response Yes Predict Reason	Response Yes Predict Reason	<input checked="" type="checkbox"/>		3	
Response Prediction Correct	Response Correct	<input checked="" type="checkbox"/>			

Up to seven new variables can be created. Interestingly a **Response Prediction Correct** variable is included. This is a dichotomous variable that is given a 1 if the prediction is correct and 0 otherwise.

Predictions are determined by probability scores. The **Cut Off** determines the predicted outcome. For example if the cut off is set to 0.6 for **Response Yes Probability**, the **Response Prediction** is Yes for any records reaching or exceeding this value.

NOTE: **Cut Off** values will always sum to 1.

Accept the default values and click **Run** to generate a new dataset with scoring fields added for both the **Development** and **Validation** datasets. Scored datasets are named **Dev_** and **Val_** respectively.

Figure 14.11 shows the **Report** tab results for both partitions.

Figure 14.11: Development and Validation Report Tabs

Confusion Matrix - Response				Confusion Matrix - Response					
	Predicted				Predicted				
Actual	No	Yes		Actual	No	Yes			
	8092 (93.15%)	595			3489 (93.09%)	259			
Statistics				Statistics					
Total Records	11,397			Total Records	4,884				
Correctly Predicted	9,418			Correctly Predicted	4,070				
Percentage	82.64			Percentage	83.33				
Valid Records	11,397			Valid Records	4,884				
Entropy Explained	0.34			Entropy Explained	0.35				
K-L divergence	0.02			K-L divergence	0.01				
Cross Entropy	0.56			Cross Entropy	0.55				
Entropy of predict	0.45			Entropy of predict	0.46				
Entropy of actual	0.55			Entropy of actual	0.54				

The **Report** tab details **Entropy** and also includes a **Confusion/Classification** or **Mis-classification** matrix.

The **Percentage** correctly predicted across partitions are similar, as are other statistics. This is indicative of model stability.

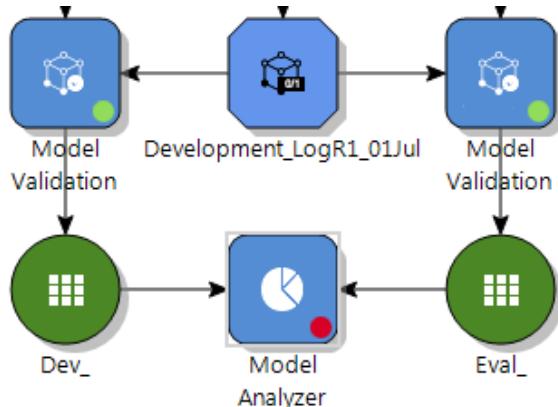
Statistical Validation is used as a means to assess overall model performance of the model on new data. **Business Validation** using charts is a complementary process to assess model performance.

Business Validation using the Model Analyzer

Business Validation helps assess the business benefit of the model. The **Model Analyzer** creates a series of graphs, tables and statistics that can be used to further validate model stability and robustness.

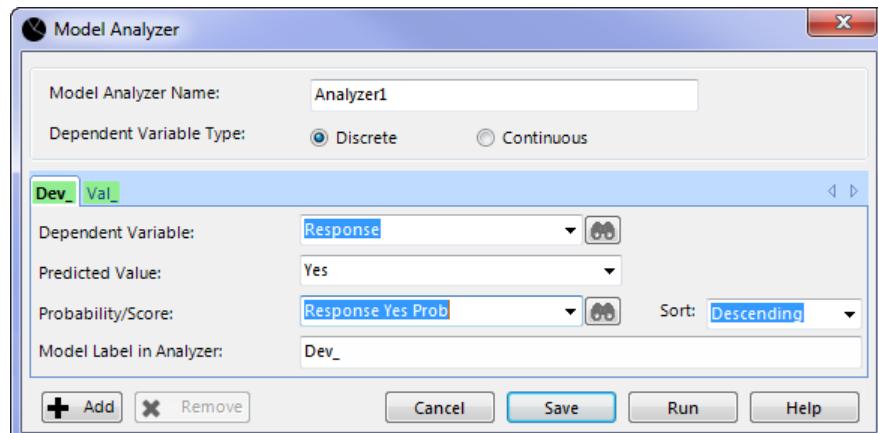
Connect both datasets to a **Model Analyzer** node as illustrated in figure 14.12.

Figure 14.12: Model Analyzer Added



Open the **Model Analyzer**, two tabs are visible, one for each connected dataset.

Figure 14.13: Model Analyzer Options



The default **Dependent Variable Type** is set to Discrete and ensures that options are auto populated as the **Dependent Variable** in this example is discrete.

However, the **Predicted Value** field value is set to the first category of the **Dependent Variable**, in this instance: *No*. The corresponding **Probability/Score** field corresponds to the *No* category.

As there is interest in assessing the model performance for the *Yes* category, these can be easily changed and set to the correct values of *Yes* and *Response Yes Prob* respectively.

Once options have been set appropriately, click **Run** to generate results.

The Model Analyzer creates five different charts: **Cumulative Chart**, **Lift Chart**, **K-S Chart**, **ROC Chart** and a **Profit Curve**. The charts are used to:

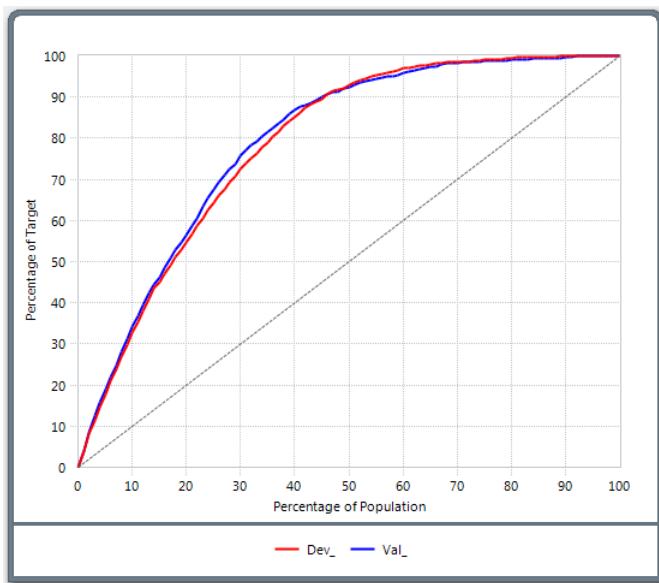
- Evaluate how well the model classifies the **Development** and **Validation** partitions
- Compare different models to determine which provides the best performance

Each of the chart types are explained in succession.

Cumulative Chart Tab

The default tab view is the **Cumulative** tab, this illustrates the **Cumulative Gains** chart for the selected category, Yes, of the **Dependent Variable**, this has a corresponding tabular representation found in the **Cumulative Lift Report** tab.

Figure 14.14: Cumulative Lift Chart



The curve reflects model performance. Selecting the top 40% of records based on Yes probability contains approximately 85% of those in the Yes category.

The **Cumulative Lift Report** tab provides the same information in tabular form. This table gives the model lift in decile increments. Notice the 4th decile showing the value determined from charts previously.

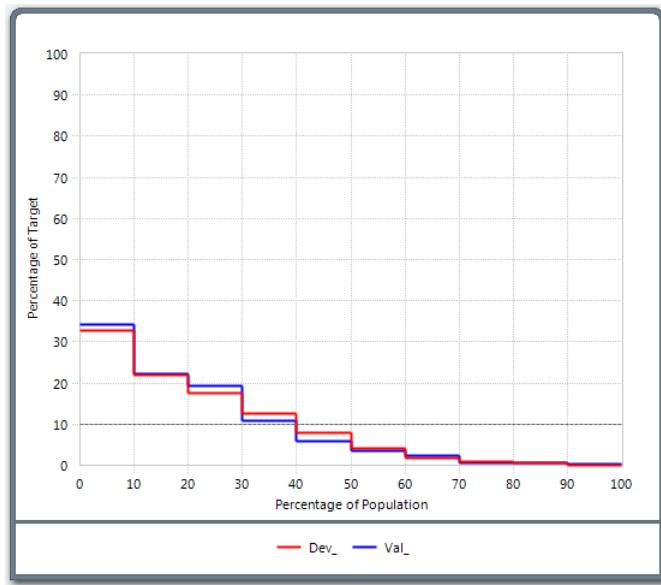
Figure 14.15: Cumulative Lift Report

Decile	Target Volume	Lift	Cumulative Lift
1	1140	32.68	32.68
2	1139	22.05	54.73
3	1140	17.66	72.39
4	1139	12.70	85.09
5	1140	7.79	92.88
6	1140	4.06	96.94
7	1139	1.67	98.61
8	1140	0.80	99.41
9	1140	0.48	99.89
10	1140	0.11	100.00

Lift Chart Tab

The **Lift Chart** tab shows the lift chart for the model versus random selection. It is a scaled ratio representation of model performance versus random selection at each decile.

Figure 14.16: Lift Chart



Here choosing the top 10%, the model is more than three times as likely to identify a Yes category member than random selection.

This information is also given in tabular form in the **Lift Report** tab as illustrated in figure 14.17.

Figure 14.17: Lift Report tab

Decile	Number of Records	# of Actual Target Values	% of Actual Target Values	# of Estimated Target Values	% of Estimated Target Values	Lift (Index)
1	1140	886	77.72 %	895.59	78.56 %	330.36
2	1139	597	52.41 %	636.15	55.85 %	234.86
3	1140	479	42.02 %	434.51	38.12 %	160.3
4	1139	344	30.20 %	340.49	29.89 %	125.69
5	1140	211	18.51 %	195.09	17.11 %	71.95
6	1140	110	9.65 %	95.23	8.35 %	35.11
7	1139	46	4.04 %	54.44	4.78 %	20.1
8	1140	21	1.84 %	34.53	3.03 %	12.74
9	1140	13	1.14 %	16.8	1.47 %	6.18
10	1140	3	0.26 %	7.16	0.63 %	2.65
TOTAL	11397	2710	23.78 %	2710	23.78 %	100

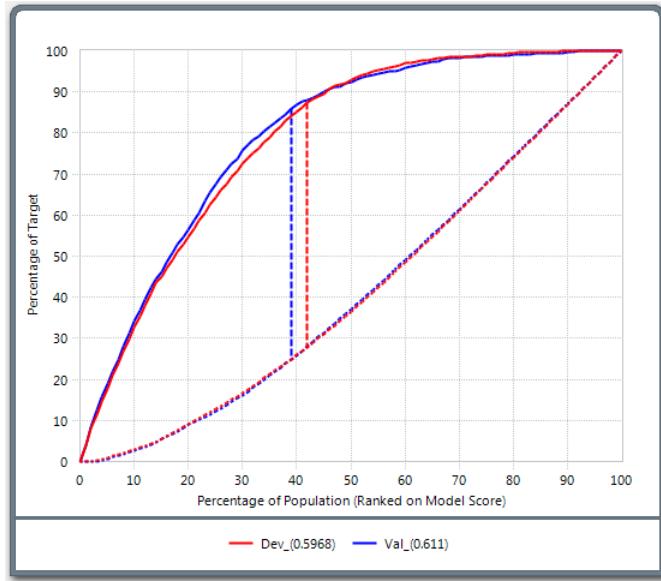
K-S Chart Tab

The **Kolmogorov-Smirnov** curve chart shows the cumulative distribution of the **Dependent Variable** categories.

The **K-S** statistic measures the difference between the two functions and returns the maximum value i.e.

where the model maximizes separation between Yes and No categories.

Figure 14.18: K-S Chart on Development (red) and Validation (blue) partitions



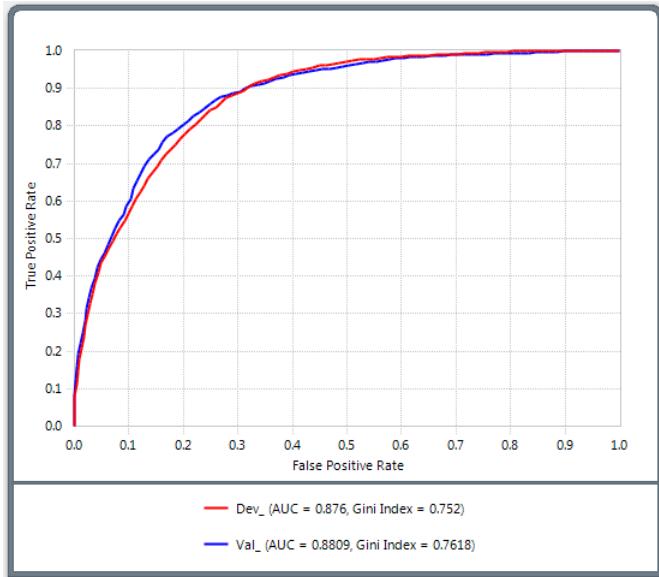
Here the maximum value of the **K-S** statistic is 0.5956 for the **Development** partition. This is calculated as the difference between the **Percentage of Target** value for each of the curves at the maximum separation point.

The optimum population selection percentage based on the model is approx. 39%, i.e. selecting the top 39% as identified by the model gives optimum separation between *No* and *Yes*.

ROC Chart Tab

The **ROC Chart, Receiver Operating Characteristic**, compares the **True Positive Rate** to the **False Positive Rate** as the discriminant, or classification, threshold changes.

Figure 14.19: ROC Chart on Development (red) and Validation (blue) partitions



The x-axis shows the proportion of the non-target category misclassified as a result of obtaining the y-axis accuracy for the target category of the dependent variable.

Here the **AUC** values are in excess of .75 and reflect an adequate model.

GOF Statistic Tab

The **GOF Statistic** tab gives the **Hosmer-Lemeshow** goodness of fit test for the model. This can be used to determine whether the model significantly predicts the **Dependent Variable**.

Figure 14.20: GOF Statistic tab

	Group	(Response = Yes) Observed	(Response = Yes) Estimated	(Response != Yes) Observed	(Response != Yes) Estimated	Total count
▶	1	3	6.29682344453694	1052	1048.703176555463	1055
	2	13	17.4079893293982	1199	1194.5920106706019	1212
	3	21	33.6643582492643	1103	1090.3356417507357	1124
	4	46	55.5045149559069	1121	1111.4954850440931	1167
	5	110	95.1624291403134	1030	1044.8375708596866	1140
	6	211	194.746141616097	928	944.253858383903	1139
	7	343	339.61399590737	794	797.38600409263	1137
	8	470	428.033471845998	656	697.96652815400194	1126
	9	588	624.668512743222	541	504.331487256778	1129
	10	905	914.901384517692	263	253.098615482308	1168
		Chi-Square	df	P-Value		
▶		25.6487122480673	8	0.00120593624108483		

The **Hosmer-Lemeshow** goodness of fit test is a variation of the **Chi-Square** test. Therefore it compares the observed and expected frequencies to a χ^2 distribution.

In this example the test statistic for the **Validation** dataset leads to a rejection of the null hypothesis of no difference between the model predictions and actuals **Dependent Variable** values.

Therefore the model is not very good, however the **Hosmer-Lemeshow** test is very sensitive to sample size; small deviations are significant with large sample sizes > 400.

Business Validation Assessment

Across all graphs the model tracks well for both the **Development** and **Validation** partitions, again reflecting stability and a reliable model.

The **AUC Statistic** is higher on the **Validation** partition in comparison to the **Development** partition; .881 vs .876, again providing good grounds to assume stability on new data and the profit curve provides attractive estimates for deployment.

The **Model Analyzer** is a valuable tool for model evaluation. Most analysts and business users focus on a selected number of graphs while others may use more. The **Model Analyzer** can be, and frequently is used for model validation and comparing separate models applied to the same dependent variable.

14.3.5 Logistic Regression Model Deployment

Once a satisfactory model has been developed, evaluated and validated, it is ready for deployment.

KnowledgeSTUDIO provides three methods for model deployment:

- Automatically scoring an existing dataset
- Generating code for the model
- Exporting to a file or database

Scoring an existing dataset is performed using the **Scoring** node from the **Action** palette. This operates in much the same way as a **Validation** node; creates a new dataset and adds scoring variables.

Code generation for **Logistic Regression** is available in four formats; *PMML*, *SAS*, *SQL Function*, *XML*. Each format can be produced by selecting the appropriate node from the **Action** palette.

Once connected code generation is straightforward and simply a matter of adding the appropriate code generation node, accessing the **Code Generation** dialog and clicking **Run**.

A snippet of *LOS* code is illustrated in figure 14.21.

Figure 14.21: SAS Code Snippet

```
/* **** */
/* section II: Model (Scoring Equations) */
/* **** */

Y1 = -2.135245337767731*relationship_1 - 3.261699354270741*relationship_2
- 3.5026721205406477*relationship_3 - 2.4809422585573015*relationship_4
+ 0.42780707522434586*relationship_5 - 0.2629318279193461*sex_0 + 0.025541980812921455
*_hours_per_week
+ 0.34289076475634284*_num_products + 0.003762232950434676*_capital_gain
- 7.66367827228117;

Prob1 = 1/(1 + EXP(-Y1));

/* **** */
/* section III: Denormalization and Generation of Scores */
/* **** */

Response_No_Prob = 1 - Prob1;
Response_Prediction = "No"; _KS_MAX = Response_No_Prob;
Response_Yes_Prob = Prob1; IF Prob1 > _KS_MAX THEN DO; _KS_MAX = Prob1; Response_Prediction =
"Yes"; END;

/* **** */
/* section IV: cleanup */
/* **** */

DROP _hours_per_week _num_products _capital_gain _KS_MAX Y1 Prob1 relationship_1
relationship_2 relationship_3 relationship_4 relationship_5 sex_0;
RUN;
%mend KST_MODEL;
```

Note that any **KnowledgeSTUDIO** model generated as *LOS* code is packaged in the form of a macro. The user need only supply the input and output datasets to deploy.

The generated code can be saved by selecting the **Save As...** option from the **File** menu. When saved, the code will have an appropriate extension.

For example a saved *LOS* code file will have the extension **.sas*, which can be run within an appropriate environment.

A model can be used to score an existing dataset and then be exported to a variety of file formats.

Available formats are found on the **Source** palette and reflect the **Data Import** capabilities.

Exporting to any format is a matter of dragging the appropriate node onto the canvas, connecting the dataset and setting options.

Once complete clicking the obligatory **Run** exports to the desired format.

14.4 Summary

This chapter described **Logistic Regression**, and the **Logistic Regression** capabilities of **KnowledgeSTUDIO**; how to effectively create, test, modify and deploy logistic regression models.

As a result of completing this chapter the user should be able to:

- Understand **logistic Regression** and its applicability
- Build **logistic Regression** models using **KnowledgeSTUDIO**
- Analyze and interpret the model outputs
- Validate the model from statistical and business perspective
- Apply and deploy logistic regression models

Exercises

The exercises use the file: *Census.xlsx*. This file should be provided by the trainer.

This file contains some demographics such as *age*, *gender*, etc, some variables recording financial information such as *capital_gain* and *capital_loss* and a **Dependent Variable**; *Response*.

1. Create a project and insert the *Census.xlsx* file.
2. Explore the data and familiarize yourself with the variables and distributions.
 - (a) Use the **Overview Report** tab to generate univariate statistics.
 - (b) Use the **Charts** tab to generate visual representations.
 - (c) Use the **Segment Viewer** as a means to identify good potential predictors.
 - (d) Use **Crosstabulations** and **Correlations** where necessary.
 - (e) What is the distribution of the **Dependent Variable**?
 - (f) Are any variables good predictors of the **Dependent Variable**?
 - (g) Can any of the discrete variables with large numbers of categories be transformed to a small number of categories?
 - (h) Use **Characteristic Analysis** to assess interactions with the **Dependent Variable**.
 - (i) Generate **Measures of Predictive Power** to further assess potential predictors.
 - (j) Try using **KnowledgeSTUDIO Decision Trees** as a means to identify predictors. Try generating a list using **Decision Trees** and referencing the list from within **KnowledgeSTUDIO**.
3. Once the data has been explored and potential predictors identified, create two partitions to develop and test the model. Partition the data into two datasets, named appropriately, use a 70%, 30% split, or any of your choosing for partitions.
 - (a) Create a **Logistic Regression** model using **Stepwise Selection** as the **Variable Selection Mode**.
4. Once generated, assess the output.
 - (a) Is the model significant?
 - (b) What amount of **Entropy** is explained?
 - (c) What order were variables included in the model? Does this reflect the order suggested by the **Decision Tree**?
 - (d) Are all included variables significant? ... they should be if a **Stepwise Selection** was chosen!
 - (e) Are there any multicollinearity issues?
5. Repeat the model steps and try a selection of other variable selection modes besides **Stepwise Selection**; is there the same set of predictors in the final model?
6. Once a final model has been created, use appropriate methods to further evaluate and validate results.

7. Compare the results to those obtained using a **Decision Tree** using the same **Dependent Variable** and **Independent Variables**.
8. Explore the code generation options for the model. Generate code and save the code as an external file.
9. Finally, use a Strategy Tree adding additional measures to determine an appropriate deployment strategy.

Chapter 15: Neural Networks

15.1 Introduction

Neural Networks, are a relatively new development in statistics and data mining. They are based on brain functionality and attempt to emulate the process by which we make a decision.

In scientific terms, this process is determined by electrical impulses discharged by neurons in the brain.

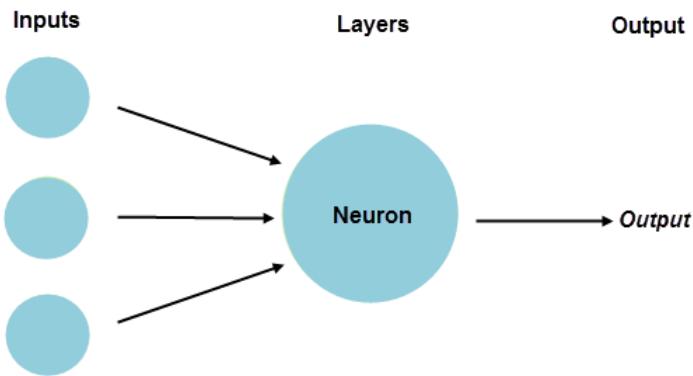
Neurons are specialized cells that conduct electrical impulses. The connections between neurons are referred to as synapses and these are the pathways along which the impulses travel. The average number of neurons in the brain runs into billions. Each one of these neurons is connected to, on average, 1,000 other neurons (University of Bristol, 2011).

As each neuron receives electrical input from on average 1000 other neurons, simultaneous impulses are summed and, if strong enough, discharge.

This forms the input to the next neuron in the network, eventually leading to an action or, for the purposes of statistical interpretation, an output.

Figure 15.1 illustrates a simplistic **Neural Network**.

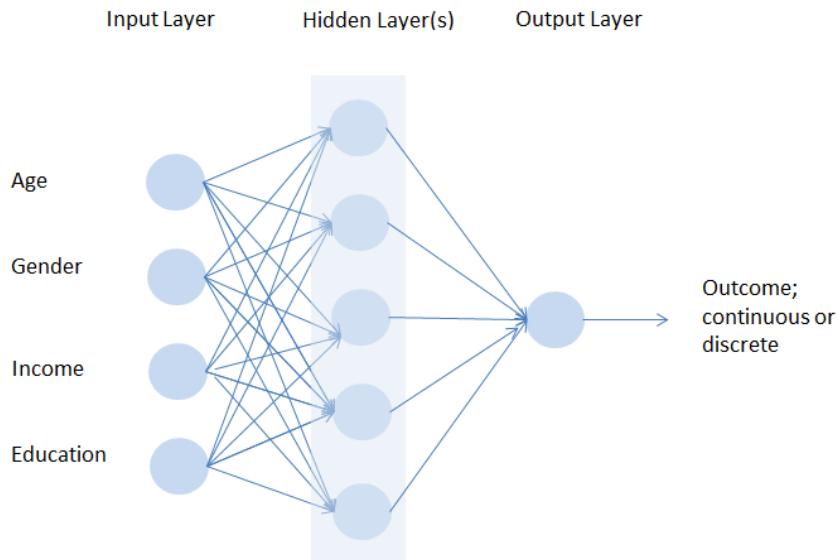
Figure 15.1: A Neural Network Illustration



As a direct result of the attempt to model brain functionality, the algorithmic emulation of a **Neural Network** is more precisely referred to as an **Artificial Neural Network (ANN)**.

An **Artificial Neural Network** has an input layer; a set of variable values, one or more hidden layers, and an output layer.

Figure 15.2: Artificial Neural Network (ANN), Basic Architecture



Artificial Neural Networks can be applied to data as a means to model an outcome. They can be used in much the same instance as a regression; **Linear**, **Logistic** and **Multinomial Logistic** or a **Decision Tree**, as the **Dependent Variable** can be binary, discrete or continuous.

The versatility of **Neural Networks** cannot be denied, however, due to the interconnectedness of elements, the exact reasoning behind the outcome is opaque.

There are many types of **Neural Network** available with many adjustable parameters. **KnowledgeSTUDIO** provides one method called a **Multi-Layer Neural Network**.

This is a simple **Feed-Forward** network with three layers: input, hidden and output. The **Feed-Forward Neural Network** was the first and simplest type of **Artificial Neural Network** devised.

In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes to the output. There are no cycles or loops in the network.

This chapter aims to develop a **Neural Network** model in **KnowledgeSTUDIO** and provides exercises to reinforce learning at the end of the chapter.

As a result of completing this chapter users should be able to:

- Describe **Neural Networks**
- Develop, evaluate and validate a **Neural Network** model using **Altair KnowledgeSTUDIO**
- Compare results to alternative techniques such as **Logistic Regression** and **Decision Trees**

15.2 Description

To understand an **Artificial Neural Network**, a framework is required. To not only determine how the outcome is arrived at but also, how the learning process develops.

Take the example of a child and an apple. If this is the first time the child has seen an apple, many factors will characterise the object encountered. Let us focus on three: **Shape**, **Colour** and **Taste**:

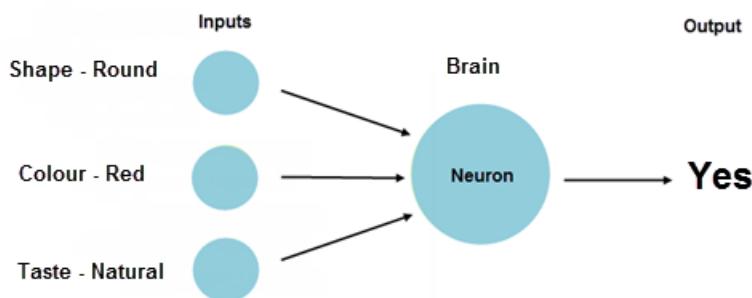
- **Shape** Round
- **Colour** Red
- **Taste** Natural

Lets say the child automatically stores these attribute values as what defines this object as an apple, with different importance, or weighting, associated with each attribute value.

If, at a later time the child encounters a similar object, there are now three things that can be used to determine whether it is in fact, an apple.

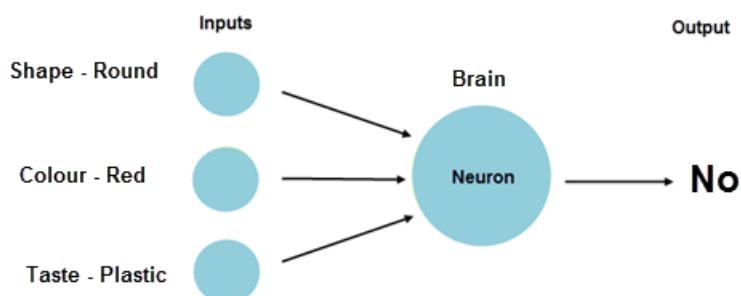
Of course, weightings are applied to each attribute, these are then summed to produce and outcome; in this case, the outcome is a **Yes**, as depicted.

Figure 15.3: Assessing the Object



Other objects will be assessed using these criteria, for example if a small plastic red ball is presented.

Figure 15.4: Additional Assessment



In this case, the inputs are fed into the system and a **No** returned.

The error in determining the outcome is fed back along the system and the weightings associated with each attribute are adjusted to some degree to account for the error.

For a human, this is straightforward and minimal training is required to determine the correct outcome.

15.2.1 ANN Emulation

To emulate the same process, **Artificial Neural Networks** require more than one object to make an assessment of an outcome. These objects come in the form of data; records and variable values.

The model must be trained to predict an outcome. This is achieved by passing more data through the network. The input is referred to as the Input Layer and is the value of each records variable attributes.

Random weights are initially assigned to values of input layer variables, these are then summed and passed to the next neuron in the network and so on until an outcome results.

Comparing the predicted and actual values results in a correction begin fed back along the system. The weights associated with each input attribute value are adjusted.

This continues for all records, and cycles through the dataset many times until a stable solution is found.

Passing many items through an **Artificial Neural Network** many times is the means by which the network learns; continually updating attribute weights until a stable solution is arrived at.

15.2.2 Layers in a Neural Network

All ANNs will have inputs, generate an output and have at least one hidden layer.

Neural Networks can be designed with a user defined number of hidden layers and neurons and **KnowledgeSTUDIO** provide the ability to specify network configuration.

NOTE: : Default ANN parameters usually suffice in practice. Further reading regarding Neural Networks is desirable if the need to specify additional nodes and layers is required.

15.2.3 Training a Neural Network

Training a **Neural Network** is a matter of providing data to pass through it. In general it is advised to have a very large training set to allow the network to identify and pick out patterns.

To identify patterns, data may be passed through a network hundreds of times, increasing the risk of overfitting.

To address this, **KnowledgeSTUDIO** provides an option to use a testing data set during the building process to prevent overfitting. The testing partition can be referenced and used during model building.

NOTE: This method of overfit prevention is only recommended for use with small files sizes. Applying to large datasets will significantly increase processing time and is not recommended.

15.2.4 ANNs in Practice

The reasoning behind predictions for **Logistic Regression** or **Decision Trees** is easily determined.

As a result of the nuances of a **Neural Network** it is very difficult to determine how and why the outcome was arrived at, and is the main thrust behind referring to a **Neural Network** as a *black-box* technique.

Table 15.1 lists some advantages and disadvantages of using **Neural Networks**

Table 15.1: Pros and Cons of Neural Networks

Pro	Con
Easy to use in KnowledgeSTUDIO	Require lots of training data
Can be used for a discrete or categorical dependent variable	May overfit to data
Ability to detect complex relationships	Little understanding in relation to predictions
Can be more accurate than alternatives. Ideal if a prediction is all that is needed	Can be memory intensive

15.2.5 Data Requirements

KnowledgeSTUDIO Neural Networks require scale/continuous data.

Categorical variables can be included via dummy coding, and in this case the reference category can be set using the **Attribute Editor**.

The training algorithm of the **MLNN** normalizes all input signals to the range [0,1]. The normalized data can be exported and an option to do so is available in the wizard dialog.

MLNN Networks can be used for both continuous or categorical **Dependent** and **Independent Variables** (via dummy-coding where necessary).

15.3 Neural Network in KnowledgeSTUDIO

This section explains the process of creating, analysing, validating and deploying a **Neural Network** model, and uses the file: *Census_DV.xlsx*.

This file contains some demographics, some financials and a generic **Dependent Variable** called *DV*.

Steps when developing a **Neural Network** model are identical to other modelling process steps:

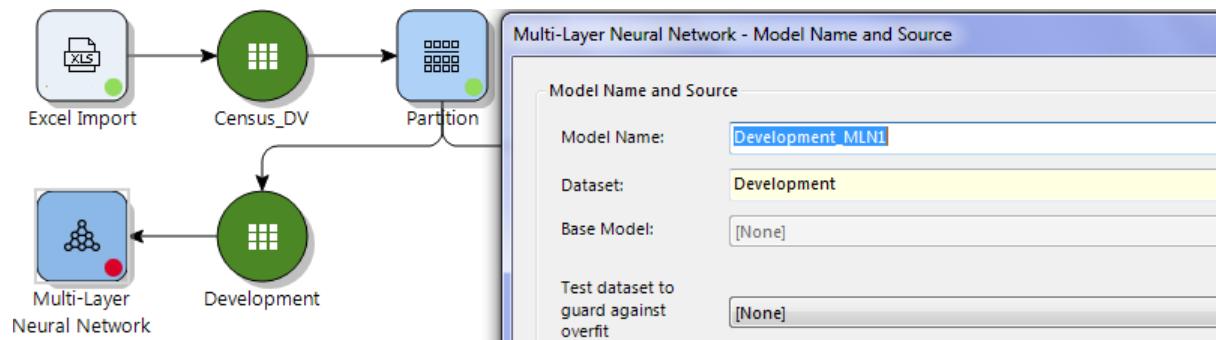
- Data Exploration
- Candidate Variable identification and selection
- Partitioning
- Modelling
- Evaluation
- Deployment

For this demonstration it is assumed that exploration and candidate variable selection has been performed.

Insert the file *Census_DV.xlsx*. Create two partitions called **Development** and **Validation** using a 70/30 split respectively.

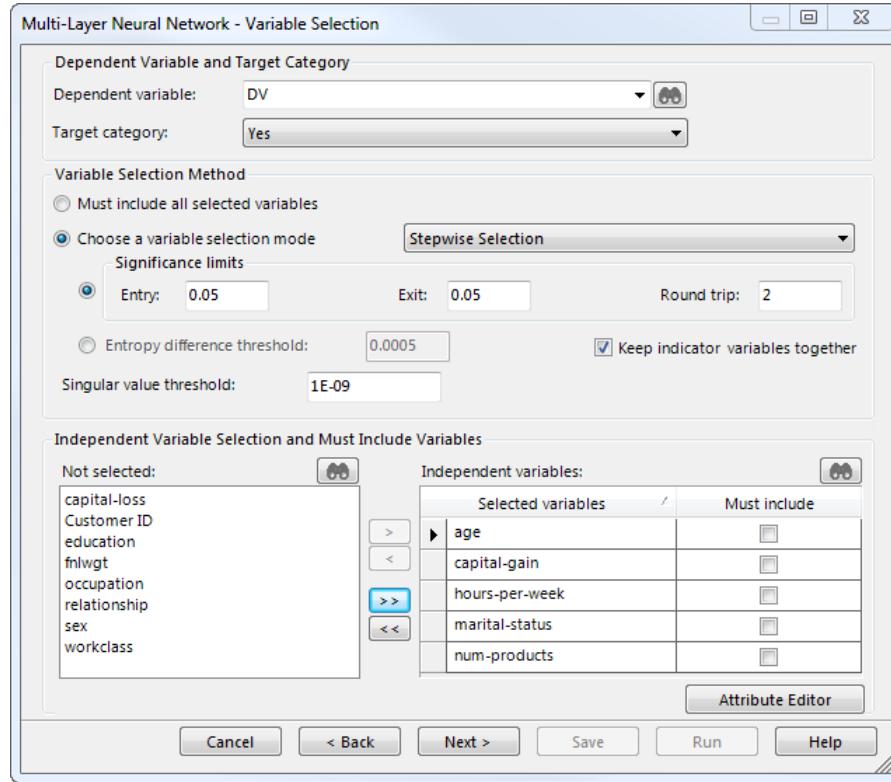
Add a **Deep Learning** modelling node from the **Model** palette, connect to the **Development** partition and open as illustrated in figure 15.5

Figure 15.5: Multi-Layer Neural Network – Model Name and Source



The **Model Name and Source** area provides generic naming aspects and dataset connections and also an additional option specific to **Neural Network** modelling: **Test Dataset to guard against overfit**.

Figure 15.6: Multi-Layer Neural Network – Variable Selection



This dialog has three distinct areas:

- **Dependent Variable and Target Category**
 - Provides options to specify the **Dependent Variable** and select the **Target Category**
- **Variable Selection Method**
 - Options for including variables in the model
- **Independent Variable Selection and Must Include Variables**
 - Candidate variable specification, including force options and **Attribute Editor**
 - The **Attribute Editor** enables access to modifiable variable properties including missing value treatment and reference category selection for categorical variables

Variable Selection Methods are identical to those available for **Linear Regression** and **Logistic Regression** and include enter and stepwise methods based on user default or user defined defined **Significance Limits** or an **Entropy Difference Threshold**.

In this example, the objective is to predict customers who are more likely to respond to a direct mailing campaign. The **Dependent Variable** is **DV**, with a target category of **Yes**.

The variable selection method is **Must Include all selected variables**, with **Independent Variables**:

- *capital_gain*

- *hours_per_week*
- *marital_status*
- *age*
- *num_products*

NOTE: Using a stepwise variable selection method for a **Neural Network** model can be a lengthy operation. To address this, **Decision Trees** or the **Measures of Predictive Power** can be used as complimentary methods to determine an initial list of candidate predictors.

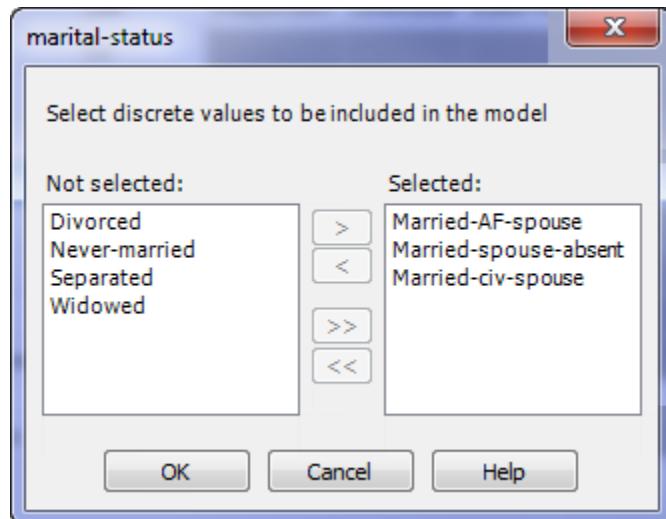
NOTE: Overfit checking is not recommended as a result of the amount of time it takes to validate but may be necessary when using a very small number of records.

Additionally the **Attribute Editor** provides access to modify independent variable properties including dummy coding options.

In this dialog, the **Dependent Variable** target category is Yes, and five variables are selected to build the model using the **Must Include all selected variables** option.

The default reference category specifications for the field *marital_status* are set as per figure 15.7.

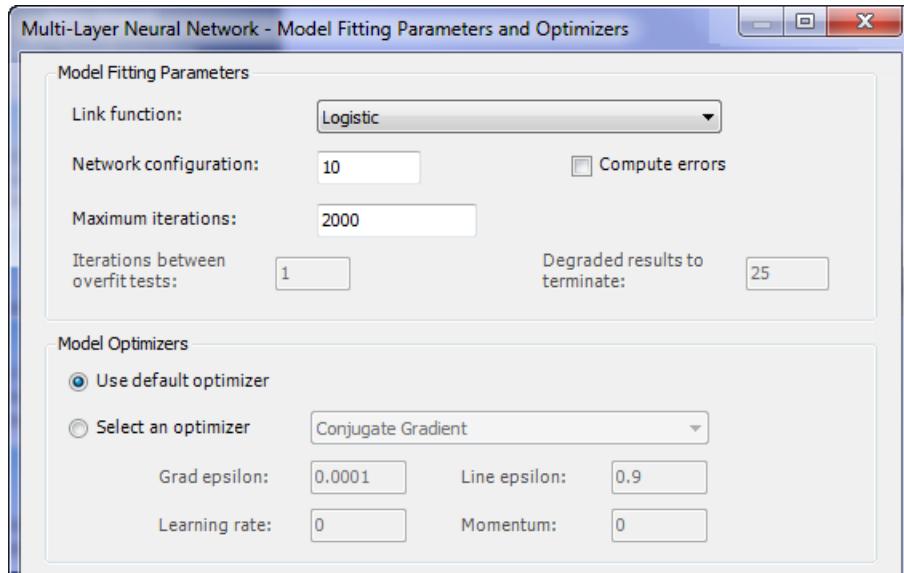
Figure 15.7: *marital_status* Settings



Here the categories referring to those unmarried states are selected as a combined reference category and can be referred to as *Other*.

Click **Next >** to open the **Multi-Layer Neural Network – Model Fitting Parameters and Optimizers** dialog.

Figure 15.8: Multi-Layer Neural Network - Model Fitting Parameters and Optimizers



Model Fitting Parameter options are detailed in table 15.2.

Table 15.2: Model Fitting Parameters

Option	Definition
Maximum Iterations:	Max no. of times the data passes through the network to ensure correct identification of patterns and to ensure parameter estimates converge
Link Function:	The nature of the transformation; predictions follow known distribution
Network Configuration:	Specify the configuration of the hidden layers in a neural network. For example: 4/6/8: three hidden layers with 4, 6, & 8 neurons respectively
Compute Errors	Standard Errors are computed and displayed in results. This may increase model run time.
Iterations between overfit tests	After each data pass, the model is validated against the test dataset Performing this test after each iteration is time consuming so users can specify how many iterations should be cycled before validating
Degraded results to terminate	No. times model results, compared to validation, continue to degrade before terminating training and reverting to the model with the best results

The second section details **Model Optimizers**. Three model optimizers are available. These are usually left at their defaults but can be modified. Available options are listed in table 15.3.

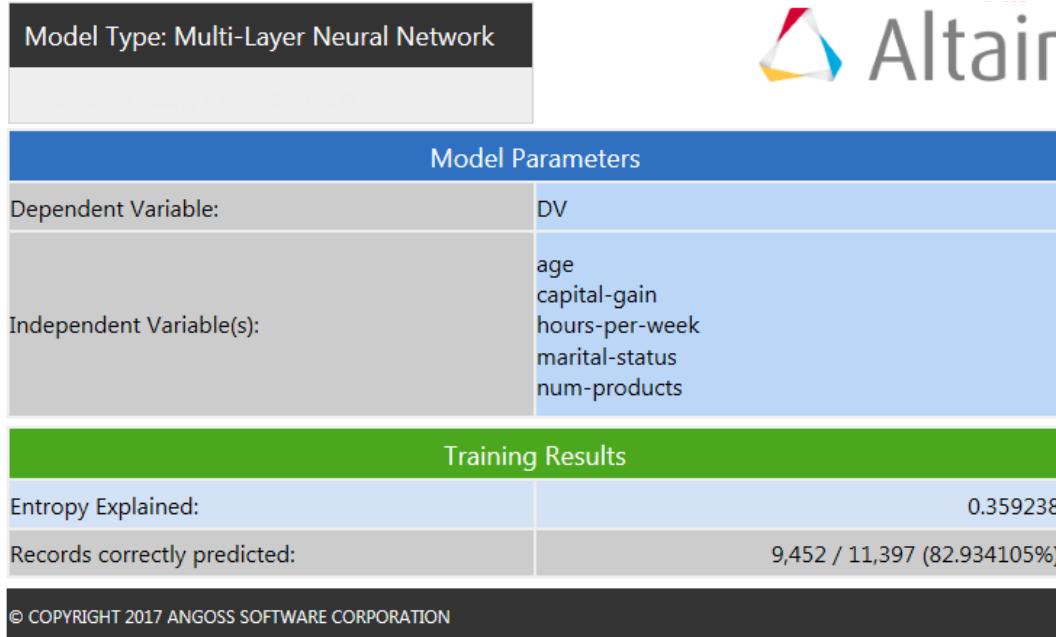
Table 15.3: Model Optimizers Available Options

Option	Definition
Conjugate Gradient (default)	Most efficient method
BFGS	Broyden-Fletcher-Goldfarb-Shanno. Use if small no. variables. With increased numbers of variables (& weights), BFGS requires storage and memory
Classic Back Propagation	Adjusts weight based on previous weights and on moving in the direction of the greatest rate of decrease of the error. Not as accurate as the other algorithms
Grad Epsilon	Epsilon for gradient, should be close to 0 as a starting point
Line Epsilon	Similar to Grad Epsilon but used by line optimizer. Initial value should be close to 1

15.3.1 Model Results

Once options have been set, run the model, once complete access results.

Figure 15.9: Model Results - Overview



The screenshot shows a software interface for model results. At the top, it says "Model Type: Multi-Layer Neural Network". Below this are three main tabs:

- Model Parameters**: Shows "Dependent Variable: DV" and "Independent Variable(s): age, capital-gain, hours-per-week, marital-status, num-products".
- Training Results**: Shows "Entropy Explained: 0.359238" and "Records correctly predicted: 9,452 / 11,397 (82.934105%)".
- Results**: This tab is partially visible at the bottom.

At the bottom left of the interface, it says "© COPYRIGHT 2017 ANGOSS SOFTWARE CORPORATION".

Three generic tabs are provided, relaying model results. The default tab displayed is the **Results** tab.

NOTE: **Iteration History** is only produced if a variable selection mode is included.

Results Tab

The results tab contains a series of views. Views are selected using the **Output to view:** dropdown, and the default view displayed is the **Model Overview**.

Here the **Entropy** explained is modest to good at around 0.37. There are a large proportion of cases correctly predicted by the model.

The selectable dropdown is organized in a similar fashion for any predictive model output. If a stepwise method was selected when specifying model parameters, the stepwise results are listed. The final model is always listed as the **Currently Selected Sequence** from the dropdown.

The **Currently Selected Sequence** contains a lot of content split across a number of sections. Most of which are generated for any predictive model.

For example there is a **Model Fitting Summary for DV** and a **Global Null Hypothesis Testing** section, relaying overall fit statistics and model hypothesis testing.

These are generated and interpreted in the same way for most modelling techniques.

Figure 15.10: Model Fitting Summary for DV

[Project15].[Development_MLN1]

Sequence #1

Model Fitting Summary for DV			
Chi-Square: 4,491.455123	Chi-Square Degrees Of Freedom: 100	P-Value: 0.000000	Generalized R^2: 0.325707
Entropy Explained: 0.359238	AIC: 8,213.273206		BIC: 8,954.724855
Percent Concordant: 88.232427	Somer's D: 0.764932	Percent Discordant: 11.739245	Gamma: 0.765149
Percent Tied: 0.028328	tau a: 0.277299	Total Pairs: 23,541,770.000000	c: 0.882466
Global Null Hypothesis Testing (BETA=0)			
Statistic	Chi-Square	DF	p-value
Likelihood Ratio	4,491.455123	100	0.000000
Score		100	
Wald		100	
	Negative 2(Log-Likelihood)	DF	
Null Model	12,502.728329	-	
Full Model	8,011.273206	100	

Again, this relays some statistics from the **Model Overview** view but also provides additional measures.

Note the **Generalized R2** and **Entropy Explained** statistics, both suggesting that the percentage of the overall proportion of the variation of the **Dependent Variable** explained by the model is roughly in the mid 30's.

The **P-Value** for the **Chi-Square** statistic in the **Model Fitting Summary for DV** and the **P-Value** for the **Likelihood Ratio** test in the **Global Null Hypothesis Testing** section are both well below the typical cutoff value of 0.05. The null hypothesis of no relationship between the model and the **Dependent Variable** is thus rejected.

The next section; **Independent Variable Statistics**, illustrates model parameters for each variable for each neuron in the hidden layer, not shown. Equations are generated for all 10 default neurons. Each neuron is denoted as: **H1_n**, where **n**, denotes the neuron number. There are 10 in total, relating to the total number of neurons in the hidden layer.

These equations can be assessed to determine, to some degree, the inner workings of the model

NOTE: The model equations in *SQL* format for each of the 10 neurons are presented in a section at the end of the **Currently Selected Sequence**, not shown.

Iteration History

The Iteration History is available only if a variable selection method chosen, and illustrates the model sequences and the variables entered and removed at each step. In this example a **Stepwise Selection** method was not chosen so this simply lists all model variables, (not shown).

Parameters and Attributes

This tab relays model settings and some output parameters, (not shown). Once output has been fully assessed, next steps are to evaluate and validate the model.

15.3.2 Model Evaluation and Validation

The **Neural Network** model can be evaluated using statistics and graphs to assess its performance on both the development and training sample and these can be spoken of in two ways:

- **Statistical Validation** Using statistics and reports
- **Business Validation** Using a series of charts

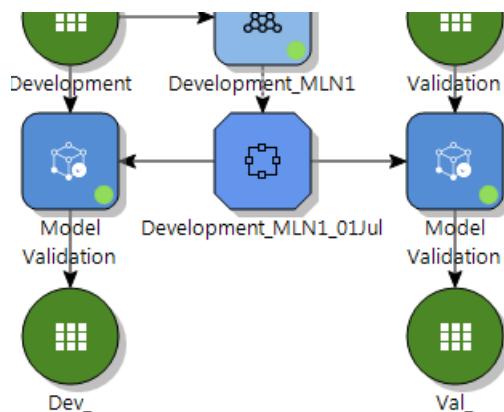
Statistical Validation

Statistical validation requires the data to be scored, creating new variables, as per any predictive model assessment. This can be applied to the **Development** partition to further evaluate the model, and to the **Validation** partition to validate the model.

Create a **Model Instance** from the **Neural Network** model. Add **Validation** nodes from the **Evaluate** palette, connect each partition to one of the **Validation** nodes and connect the **Model Instance** to both.

Name the resulting scored partitions **Dev_** and **Val_** and score both partitions with the **Neural Network** model. Once complete, the results should resemble that in figure 15.11.

Figure 15.11: Neural Network Evaluation and Validation



To view results either double click either of the created datasets or right click and select **Open View**. Results open on the **Report** tab. The illustration below shows the **Report** tab results for both partitions.

Figure 15.12: Neural Network Model Validation Statistics

Confusion Matrix - DV				Confusion Matrix - DV			
Actual	Predicted		Yes	Actual	Predicted		Yes
	No	Yes			No	Yes	
No	8111 (93.37%)	576		No	3506 (93.54%)	242	
Yes	1369	1341 (49.48%)		Yes	545	591 (52.02%)	
Statistics				Statistics			
Total Records		11,397		Total Records		4,884	
Correctly Predicted		9,452		Correctly Predicted		4,097	
Percentage		82.93		Percentage		83.89	
Valid Records		11,397		Valid Records		4,884	
Entropy Explained		0.36		Entropy Explained		0.37	
K-L divergence		0.02		K-L divergence		0.01	
Cross Entropy		0.56		Cross Entropy		0.55	
Entropy of predict		0.45		Entropy of predict		0.46	
Entropy of actual		0.55		Entropy of actual		0.54	

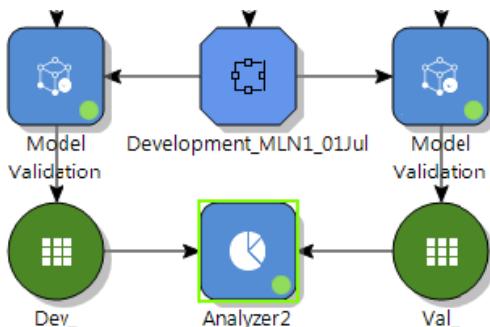
The **Report** tabs provide a statistical report detailing the amount of **Entropy** explained, in the **Dependent Variable** by the model for both the **Development** and **Validation** partitions.

The output also includes a **Confusion** matrix of the model. Across the **Development** and **Validation** partitions the percentages correctly predicted are similar as are all other evaluative statistics. This provides good evidence to suggest the model will be stable when applied to new data.

Business Validation using the Model Analyser

The **Model Analyser** can be used to generate a series of graphs to evaluate and validate the model. Drag the **Model Analyser** onto the **Workflow** canvas and connect both validated datasets as illustrated in figure 15.13.

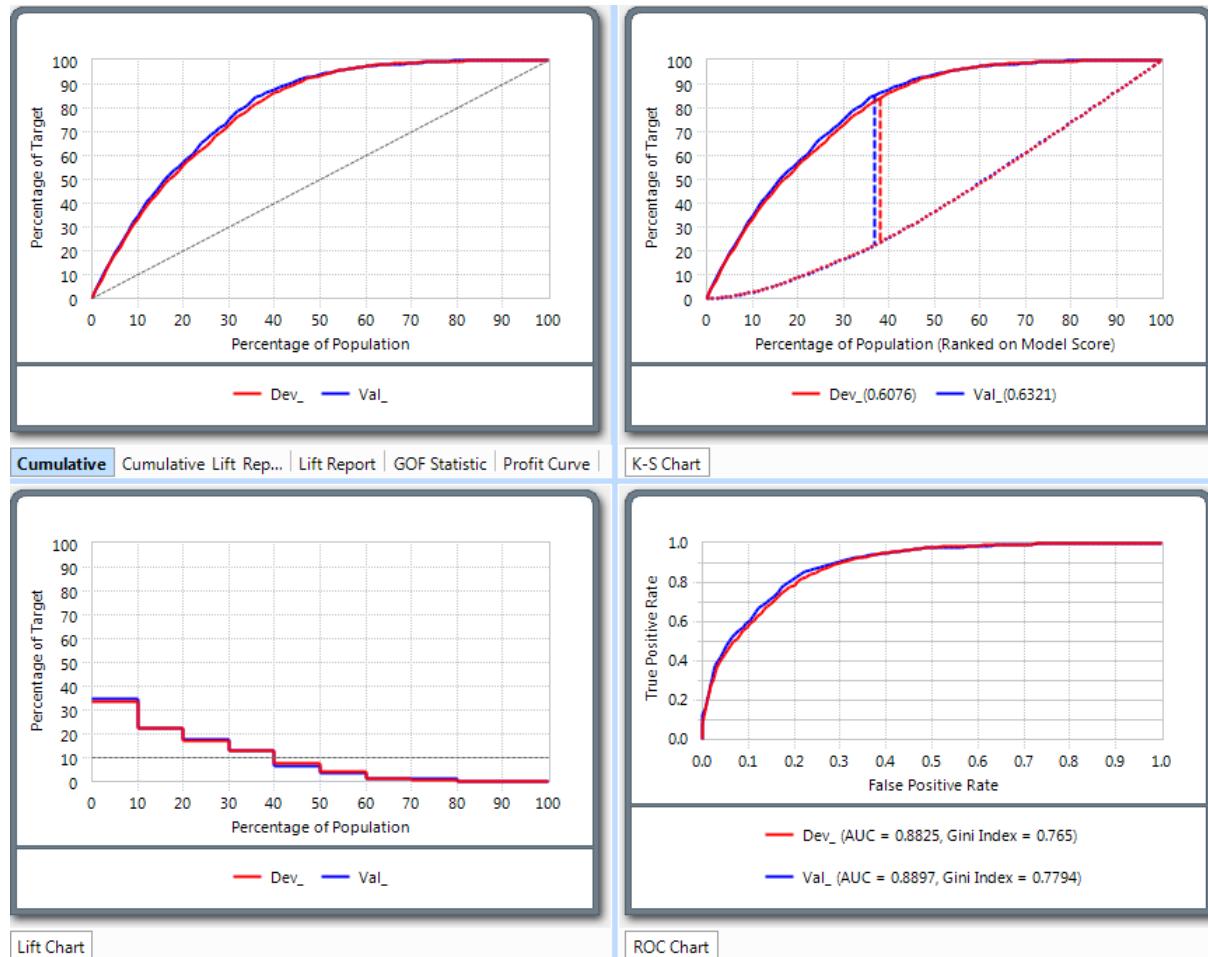
Figure 15.13: Adding a Model Analyser Node



To access the **Model Analyser** options, either double click or right click the **Model Analyser** node and select **Modify**. Tabs are visible for each connected dataset. Set the **Predicted Value** to Yes and the

Probability/Score to DV Yes Prob for both. Click **Run** to generate results.

Figure 15.14: Model Analyser Graphs



The **Model Analyser** results are interpreted as usual. The model is consistent across both datasets in all graphs. The values of the **K-S Statistic** and **AUC** are higher for the **Validation** partition in both instances, which certainly validates the model and bodes well for deployment.

15.3.3 Model Deployment

Once a satisfactory model has been developed, evaluated and validated, it is ready for deployment.

KnowledgeSTUDIO provides three methods for model deployment; automatically scoring an existing dataset using the **Scoring** node, generating code for the model or exporting to a file/database.

These methods are not illustrated here but can be accomplished by attaching the relevant node to generate desired results. Code generation for **Neural Networks** is available in four formats: *SAS*, *SQL*, *XML* & *PMML*.

15.3.4 Understanding the Neural Network and Comparing with other Methods

Given the output generated, the shortcomings of **Neural Networks** in comparison to other modelling techniques is laid bare; there is no significance associated with predictors overall and the equations do not lend themselves to easy interpretation, nor is there a tree based output to understand segments.

These shortcomings can be addressed to some degree in a number of ways:

- Using **Regression** to assess predictor impact
- Using **Decision Trees** to assess and visualize predictor importance

These methods can be applied to both the original **Dependent Variable** and the predicted results.

15.3.5 Summary

This chapter details the **KnowledgeSTUDIO** functionality for developing, evaluating, validating and deploying **Neural Network** models. On completion of this chapter the user should be able to:

- Describe **Neural Networks**
- Develop, evaluate and validate a **Neural Network** model using **Altair KnowledgeSTUDIO**

Exercises

The data used is the **Census** file.

This can be found by loading the **Census Sample Project** from the **Prepare Sample Data...** dialog found in the **Help** menu. All elements can be deleted, retaining only the **Census** dataset.

The dependent variable is *income* with two categories; $\geq 50k$ and $<50k$.

1. Explore the data using the profiling features available in **KnowledgeSTUDIO**
 - (a) Use the **Data** tab to view some cases for each variable
 - (b) Use the **Overview** tab to assess variable summaries
 - (c) Use the **Dataset Chart** tab to further understand and qualify variable characteristics
 - (d) The **Crosstabs** tab can be used to generate crosstabulations, scatterplots & means tables between the **Dependent** and **Independent Variables** as well as between **Independent Variables**
 - (e) Are any variables related to the **Dependent Variable**?
 - (f) What variable is most strongly associated with the **Dependent Variable**?
 - (g) Are there any other potentially useful predictors?
 - (h) Use the **Correlations** tab to further identify potential multicollinearity issues
2. Some variables have a large number of categories. Assess whether they can be reduced meaningfully to a small number and included in the model.
3. Once initial exploration is complete, variable characteristics noted, data preparation can proceed.
4. Create two (or more) dataset partitions to develop and test the model. Use a 70/30 split and name the partitions **Development** and **Validation**, or any name of your choosing.
5. Of the variables identified for inclusion in the model, select only the one you deemed to be most strongly associated with the dependent variable.
6. Use the **Predictive Models...** to develop a **Neural Network** model. In this instance accept all default values.
7. Assess the model results. What is the **Entropy** value and prediction rate?
8. Assess the modelling equations. Do they shed light on the working of the model?
9. Assess the model statistically on the **Validation** partition using **Decision Tree**
10. Try to further understand the model by using its predictions as the **Dependent Variable** in a **Decision Tree**.

11. Use the model analyser to further assess the model.
12. Re-run the model specifying different **Network Configurations**.

For example in the **Predictive Model - Model Fitting Parameters** dialog specify: 3/4/2, this is a network with three hidden layers. The number of neurons in each layer is 3, 4 and 2 respectively.
13. Does the model take longer to run, is it more accurate?
14. Try a number of different configurations.
15. Score the data and use a **Decision Tree** with the predictions as the **Dependent Variable**.
16. Develop a **Logistic Regression** and a **Decision Tree** model and compare results to the **Neural Network**.

Chapter 16: Cluster Analysis

16.1 Introduction

Clustering is a method of grouping objects with similar characteristics into groups. The main aim of a cluster analysis is to generate *homogenous* groups where variability within clusters is minimized and the variability between clusters is maximized.

The aim of clustering is to segment a *heterogeneous* population into a number of more *homogeneous* sub-groups or clusters. Records within the same cluster should be similar to each other and should be distinct from records in other clusters.

Clustering is an unsupervised *Data Mining* technique as the grouping is not driven by any specific purpose and therefore there is no **Dependent Variable** involved in the model. Instead, all variables are considered independent variables.

Cluster Analysis is a useful technique for uncovering *homogenous* groups. As such, it is mainly used as an exploratory technique in a number of areas including pattern recognition, image analysis, information retrieval and bioinformatics alongside business and marketing.

Within the business and marketing domain **Cluster Analysis** is an appropriate technique for customer segmentation, product targeting, credit behaviour, product segmentation and fraud.

This chapter describes **Cluster Analysis** and introduces the **Clustering** capabilities of **KnowledgeSTUDIO** alongside how to effectively create, explore and utilise created clusters.

On completion of this chapter and accompanying exercises users should be able to:

- Understand **Cluster Analysis** and its applicability
- Apply **Cluster Analysis** techniques using **KnowledgeSTUDIO**
- Analyse and interpret the clusters
- Deploy the **Clustering** model
- Validating a cluster analysis model

16.2 Description

Cluster Analysis is a general technique that accommodates a number of different algorithms that can be applied to solve a specific problem by grouping objects of interest in a meaningful way.

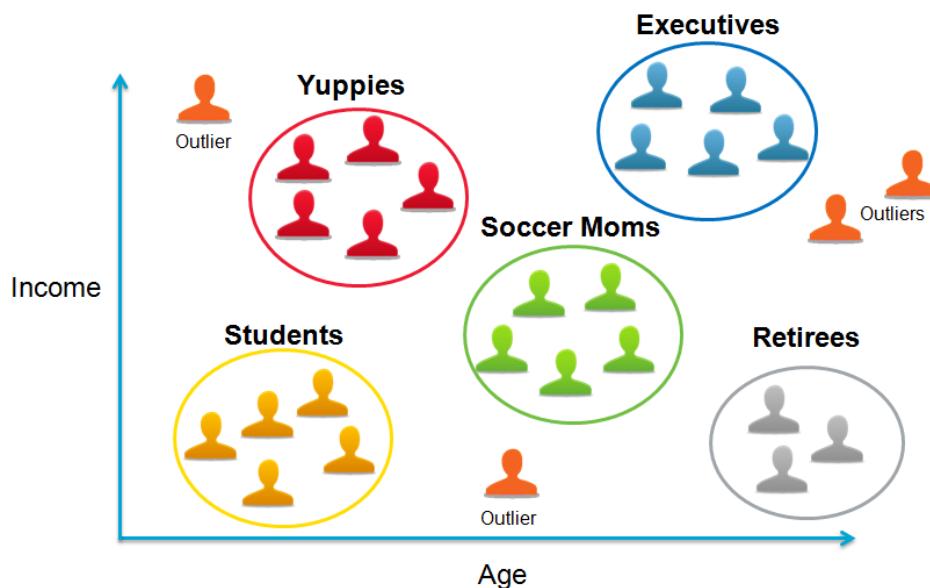
Cluster Analysis has the following characteristics:

- Unsupervised as there is no **Dependent Variable**
- Discovers previously unknown structures in a given data set that will contribute to better understanding of the data
- Produces classification segments/clusters/groups
- Reveals similarities/differences among records

An example of customer segmentation is illustrated in figure 16.1. Each point on the scatter plot represents a customer in terms of *age* and *income*. Here, five segments have been identified.

In addition, some data points have some extreme values, which may be interpreted as outliers. In clustering, classification is not based on a single variable but on the combination of multiple variables simultaneously.

Figure 16.1: Customer Segmentation



Cluster analysis works best with continuous/scale variables however **KnowledgeSTUDIO** provides the ability to incorporate categorical variables by way of dummy coding.

16.2.1 Application of Cluster Models

Cluster Analysis is applied in practice with two objectives:

- Segmentation of the data into distinct groups to more easily understand, identify and distinguish *homogenous* groups
- Generation of an index i.e. the segment number or label, to use in other models or for further exploration

16.2.2 Cluster Analysis Process

Cluster Analysis consists of the following steps:

- Select clustering variables
- Decide on clustering procedure
- Specify the number of clusters to produce
- Validate and interpret the solution

Select Clustering Variables

This step should be conducted with great care as improper segmentation may lead to poor strategies. Including relevant variables is essential and the statistical maxim of rubbish in rubbish out should be respected.

Decide on Clustering Procedure

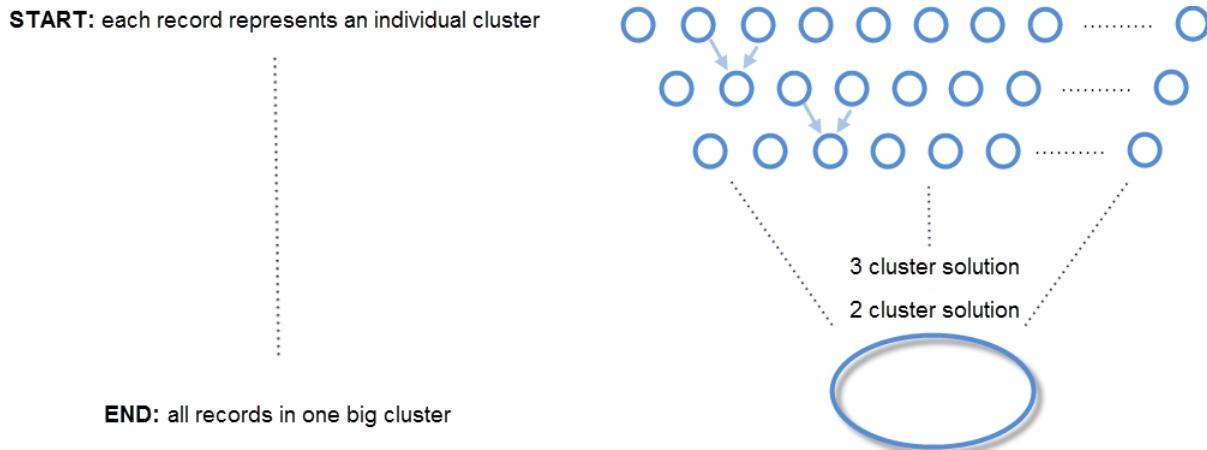
Cluster Analysis has many methods, which can be classified as: **Hierarchical**, **Non Hierarchical** and **Probabilistic**.

Hierarchical methods possess the characteristic that once a record has been assigned a cluster, it remains in that cluster.

Each case begins as a separate cluster and through an iterative process cases are joined together to form larger and larger clusters. This process continues until all cases are merged into one big cluster.

Generally, solutions prior to the final step are examined. **Hierarchical** methods are processor & memory intensive & not recommended if greater than 3000 – 5000 records.

Figure 16.2: Hierarchical Clustering



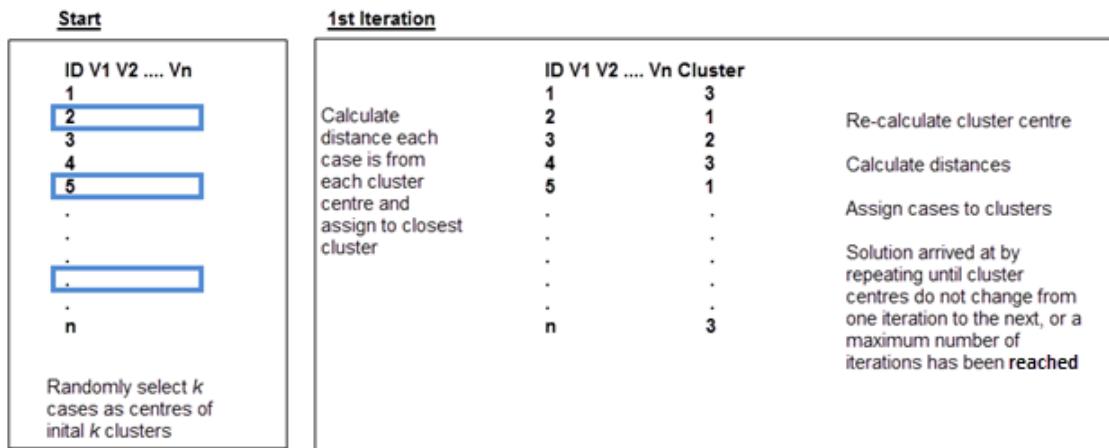
In **Non-hierarchical** methods records can dynamically change the cluster they belong to in successive iterations. One common method is **K-Means**.

K-Means starts by randomly selecting **k** records. Here, **k** is the user-defined number of clusters.

Distances between cases & cluster centres are calculated. Records are assigned the nearest cluster. The cluster centres are recalculated based on the records assigned.

This process of distance calculation and cluster assignment continues for a user-specified number of iterations or until the cluster centres do not change from one iteration to the next, i.e. **Convergence** is reached.

Figure 16.3: K-Means Clustering



NOTE: Although **Convergence** is not provided as a selectable option in **KnowledgeSTUDIO** it is referred to when determining a cluster solution.

A solution is arrived at either when a maximum number of iterations has been reached or the cluster centres converge; whichever comes first.

Probabilistic methods attempt to maximize or minimize a function and include a reference to probability distributions when assigning cases to clusters.

There are many methods and quite a few variants. More common probabilistic methods are **Two-Step**, **Expectation-Maximization** and **Latent Class analysis**.

The **Expectation-Maximization** clustering algorithm computes probabilities of cluster memberships based on one or more probability distributions. The goal is to maximize the overall probability or likelihood of the data belonging to a particular cluster.

In summary, there are many clustering methods and algorithms available.

As **Cluster Analysis** is an exploratory technique, the interpretation of resulting clusters is held in much higher regard than the clustering algorithm chosen; as many techniques can result in very similar solutions.

How Many Clusters?

The number of clusters sought is based on need. The process can be exploratory and determined by the desire to assess naturally forming clusters, or can be predetermined by a business objective, e.g.

"we have a new product and a budget to market it to three audiences" ... find three clusters.

Alternatively an optimum number of clusters may be sought where the data scientist looks for a best fit solution, from a range of possibilities.

KnowledgeSTUDIO provides this capability and uses a **Pseudo F Statistic** to assess the best solution fit. See the technical documentation for a description of the Pseudo-F statistic.

Validate and Interpret the Solution

Validation is usually assessed by splitting the dataset into two partitions and applying the same clustering algorithm on both partitions.

The outputs such as cluster distances and relative numbers of records should produce similar results to confirm the validity of the clustering solution.

The final step of any cluster analysis is the interpretation of the clusters by examining the cluster centroids and assessing whether the segments are conceptually distinguishable.

Based on individual **Cluster Analysis**, a meaningful name or label should be assigned to each cluster. Labels should be assigned that reflect the characteristics of the records in each cluster and should serve to identify each and distinguish between them.

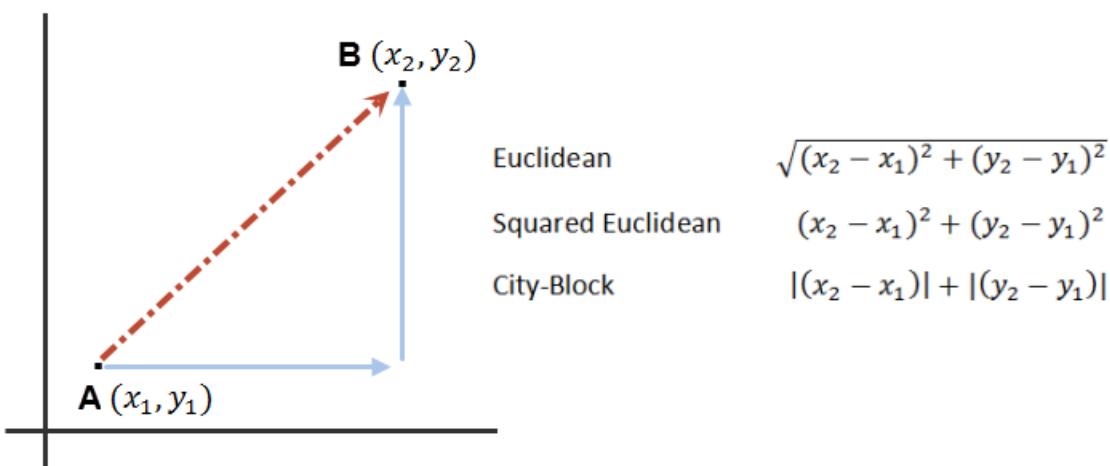
16.2.3 Distance Measures

In general when employing clustering techniques a distance measure is used to calculate the proximity between cases, or between cases and clusters, to determine cluster membership.

This is the case for **Hierarchical** and **Non-Hierarchical** methods, **Probabilistic** methods of course use probabilities!

Depending on the method, varied distance measures are available. Common distance measures are **Euclidean**, **Squared Euclidean** and **City-Block** distances. These are illustrated in figure 16.4.

Figure 16.4: Distance Measures



NOTE: The diagram illustrates clustering based on two variables. This can easily be extended to include many variables:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 + \dots + (n_2 - n_1)^2}$$

Where x, y, z, \dots, n , represent individual variables.

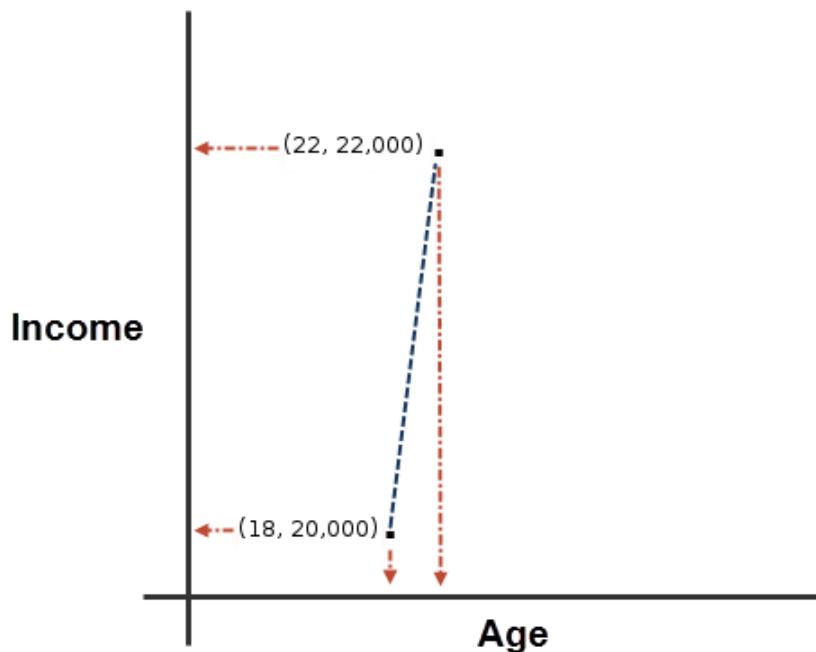
Once distances are calculated, a variety of clustering methods are available to assign cases to clusters, for example; **Nearest-Neighbour**, **Furthest-Neighbour**, **Centroid** methods, **Sum of Squares** etc.

Further discussion of these methods is not warranted in this instance as they relate mainly to hierarchical clustering techniques, which are not available in **KnowledgeSTUDIO**.

Variable Transformations when Clustering in KnowledgeSTUDIO

Any cluster solution may be dominated by variables measured on large scales. For example; two variables can be used to illustrate the effect. Take *age* in years and *income* in pounds/dollars as illustrated in figure 16.5.

Figure 16.5: Dominant Variable Measured on Large Scale



If one cluster centroid has values: (*age*: 22, *income*: 22,000) and one case has values: (*age*: 18, *income*: 20,000).

The **Euclidean** distance calculation is:

$$\sqrt{(18 - 22)^2 + (20,000 - 22,000)^2}$$

Giving:

$$\sqrt{4 + 4,000,000} = 2000.00099$$

Here, the distance between the case and the cluster centre (in fact any cluster centre) is dominated and determined solely by the difference between *income* values.

This is an aspect that must be borne in mind when including variables in a **Cluster Analysis**.

If there are many variables measured on varying scales, one or two variables may dominate if they are measured on relatively larger scales. In cases such as this, there may be a need to transform variables prior to running the **Cluster Analysis**.

KnowledgeSTUDIO deals with this by automatically transforming all variables included in a **Cluster Analysis** such that:

- Continuous variables are standardised; mean = 0 & standard deviation = 1
- Ordered variables are standardised in the same way as continuous variables
- Categorical variables are converted into dummy binary variables

16.2.4 Cluster Analysis with KnowledgeSTUDIO

This section explains the process of creating, analysing, and deploying clusters created using **KnowledgeSTUDIO** and uses the file: *Census.xlsx*.

Initiate a new project and import the file *Census.xlsx*.

16.2.5 Building the Cluster Model in KnowledgeSTUDIO

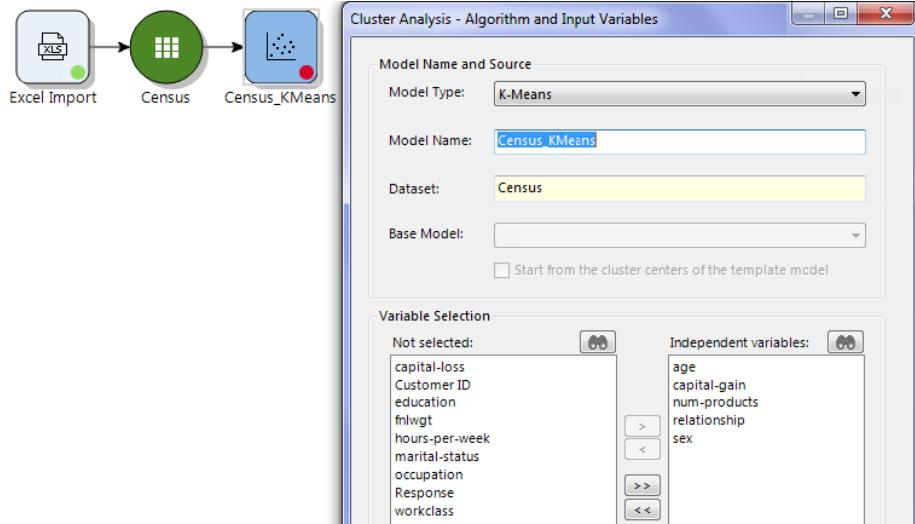
Two clustering algorithms are available in **KnowledgeSTUDIO**:

- **K-means**
- **Expectation-Maximization**

Both techniques are available through the **Cluster Analysis** node found in the **Model** palette.

Drag a **Cluster Analysis** node to the canvas and connect the **Census** dataset. Open the **Cluster Analysis** node. The first dialog is **Cluster Analysis – Algorithm and Input Variables**.

Figure 16.6: Cluster Analysis Node Added



The first dialog have two sections: **Model Name and Source** and **Variable Selection**. Available options are detailed in the table 16.1.

Table 16.1: Cluster Analysis – Algorithm and Input Variables

Option	Description
Model Type	Choose from K-Means and Expectation-Maximization
Dataset	Dataset to use when training the model. Pre-defined as per connections
Model Name	Provide a name for the model
Base Model	Training parameters of the template model are inherited from the base model If included, the option: Start from the cluster centres of the template model becomes available

Additionally, the **Attribute Editor** allows modification of variable attributes such as missing value treatment and, for discrete fields; categories to include.

Note that continuous variable values are by default standardised and ordinal variables are interpolated to the range [0,1]. For this demonstration **K-Means** is selected as the **Model Type** and five variables are selected:

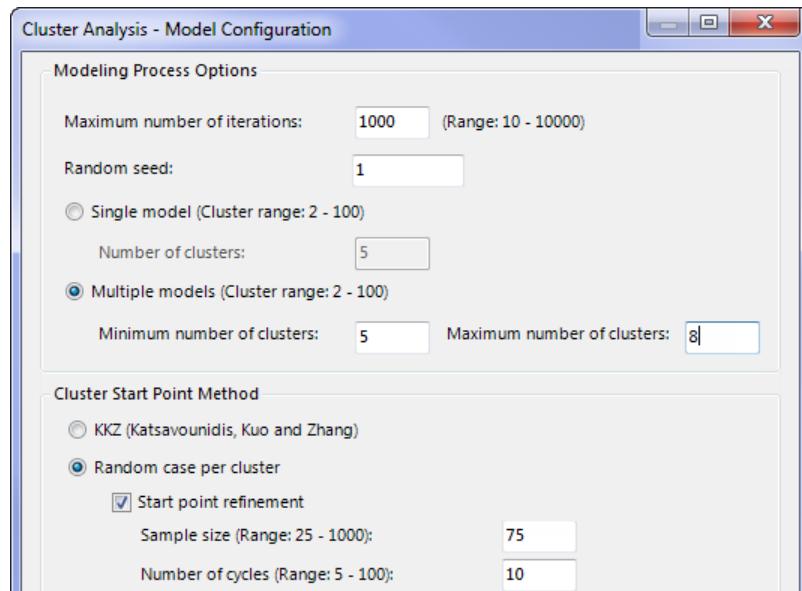
- age
- *capital_gain*

- *num_products*
- *relationship*
- *sex*

NOTE: K-Means calculates distances using the **Euclidean** method.

Click **Next >** to move to the **Cluster Analysis - Model Configuration** dialog.

Figure 16.7: Cluster Analysis - Model Configuration



The **Cluster Analysis – Model Configuration** dialog is divided into two with various parameters.

The **Modeling Process Options** provides options to choose whether to create a **Single model** with a specified number of clusters or to create **Multiple models** and specify the **Minimum** and **Maximum** values.

If **Multiple Models** is chosen, models are automatically compared and the best selected based on a goodness of fit measure; **Pseudo F Statistic**.

The **Maximum number of iterations** is the number of times the algorithm iterates through the clustering process, if convergence is reached prior to this number of iterations a cluster solution is generated.

If convergence is not reached at this point a cluster solution is generated. Output results will relay whether convergence was reached, if not, the model can be re-run and this value increased.

Each time a model is trained, a starting value or seed is used. Models trained on the same data but with different seed values may produce slightly different results.

To ensure the repeatability of results the same **Random Seed** should be used each time when running the cluster analysis.

Options related to the **Cluster Start Point Method** are detailed in table 16.2.

Table 16.2: Cluster Start Point Method – Available Options

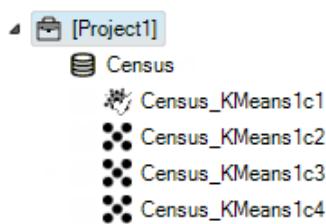
Option	Description
KKZ(Katsavounidis, Kuo and Zhang)	Optimises the distance between the seed clusters for inter-cluster separation
Random case per cluster	K records are chosen at random to be the starting point for the centres of the clusters
Start point refinement	Uses a sub-sampling procedure to attempt to find intelligent start points for the cluster centres, rather than picking purely at random.
Sample size	Determines how many subsamples are defined Value is data-dependent but generally a lower value is preferable in the case of few variables and many records, and higher values are preferable with many variables and fewer records
Number of cycles	The default setting of 10 is usually sufficient for the refinement process

Here, **Multiple Models** is selected with the **Minimum** and **Maximum** values set to 5 and 8 respectively.

Four solutions are evaluated. Click **Run** to generate results.

The best fitting model is identified with a checkmark in the **Project Pane**.

Figure 16.8: Cluster Analysis Output in the Project Pane



In this demonstration, the first model is selected. This is the five cluster model.

16.2.6 Cluster Analysis Outputs Results

Open the node to assess results. Output is spread across three tabs: **Results**, **Segment Viewer** & **Parameters and Attribute**. Of the output the **Results** and **Segment Viewer** tabs are most informative.

Results Tab

The results tab illustrates output across four tables: **Training Results**, **Cluster Distances**, **Cluster Sizes** and **Model Parameters**

Figure 16.9: Cluster Analysis Results Tab

Training Results										
Sample Size:	16,281									
Number of Clusters:	5									
Max Number of Iterations:	1000									
Goodness Of Clusters:	0.232971152596766									
Cubic Clustering Criterion:	-7.52340535550937									
R ² :	0.482370740285595									
Pseudo F Statistic:	3791.83847393986									
Cluster Distances										
	1	2	3	4	5	Variance				
1	0.000000	1.482492	2.108801	1.396387	1.532644	2.192414				
2	1.482492	0.000000	1.986682	1.397803	1.452782	1.661626				
3	2.108801	1.986682	0.000000	2.114997	2.137347	2.233179				
4	1.396387	1.397803	2.114997	0.000000	1.243874	1.362197				
5	1.532644	1.452782	2.137347	1.243874	0.000000	1.668320				
Variance	2.192414	1.661626	2.233179	1.362197	1.668320	-				
Cluster Sizes										
Cluster			Record Count							
1			3,560							
2			3,837							
3			85							
4			5,100							
5			3,699							
Model Parameters										
Independent Variable(s):	age capital-gain num-products relationship sex									

The **Training** results table contains metrics to assess the model. These are detailed in table 16.3.

Table 16.3: Cluster Analysis Results Tab

Option	Description
Sample Size	The number of records used in the model
Number of Clusters	The number of clusters in the model

Max Number of Iterations	The max number of times the algorithm will re-calculate the clusters with new centres
Goodness of Clusters	This value is a normalised Calinski-Harabasz pseudo F statistic and is equal to Pseudo F divided by (N - K) , where N is the total number of observations and K is the number of clusters. The best model has the highest Goodness of Clusters value
Cubic Clustering Criterion (CCC)	Shows how much the clusters deviate from the uniform data distribution. Larger values are more desirable. The maximum value of CCC across the hierarchy levels can be used to indicate the optimal number of clusters. For reference these numbers are usually below 2.0, and are often negative. However, as they are relative numbers, this rarely matters, as the goal is to find the number of clusters producing the largest values of the CCC
R-squared	Proportion of variance accounted for by the clusters
Pseudo F Statistic	Reflects the tightness of clusters as the ratio of the mean sum of squares between groups to the mean sum of squares within group. Larger values of the Pseudo-F indicate a better solution. It is used as a criterion for the best number of clusters

The **Cluster Distances** table shows the **Euclidean** distances between the cluster centroids. Each cluster is denoted by **Ki**.

Variance represents the intra-cluster measure of tightness. Smaller values are more desirable.

The **Cluster Sizes** table reflects no. records in each cluster. Notably there is one small cluster; **K3**, containing 85 records.

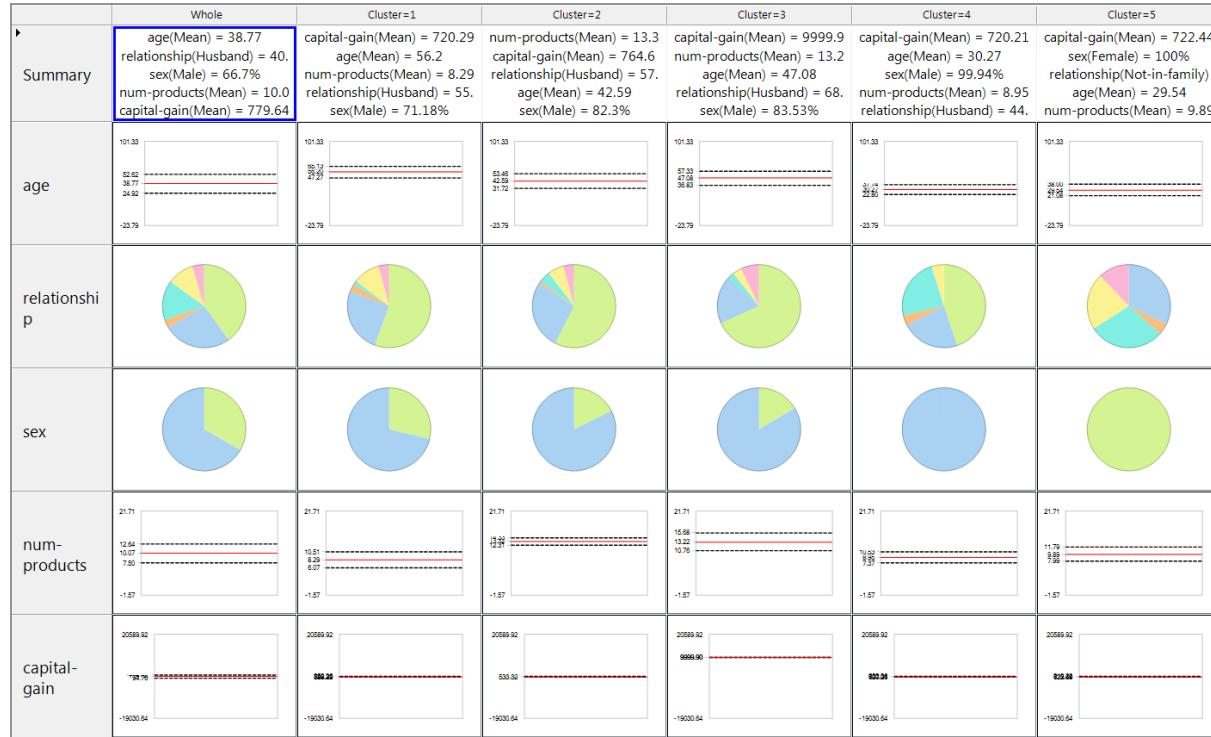
K-Means is good at detecting and creating outlier clusters and given the small size of **K3** there must be something quite specific about those assigned to it.

The **Model Parameters** table reflects the variables used to generate the cluster model.

Segment Viewer Tab

The **Segment Viewer** tab shows the distribution of each variable's data within each cluster in the form of charts.

Figure 16.10: Cluster Analysis Segment Viewer Results



The **Summary** row contains a short description of the clusters, i.e., the summary of each variable's distribution within each cluster.

For discrete variables, the value that has the highest percentage within a given cluster is shown. For any continuous variable, the mean value over a given cluster is shown.

Discrete variables are represented as pie-charts and continuous variables as solid red lines representing the mean values of the variables over a given cluster. The dashed black lines represent the standard deviation.

Double-click on any chart to enlarge. All charts for a variable use the same scale for convenient visual comparison.

The order of the variables is different for each segment and is based on the contribution of the variable within a specific cluster. The first variable in each segment has the highest contribution.

To see the values of relevance for each variable, select the **Option** button from the **Toolbar** then select the **Show Relevance** radio button.

The value of relevance is the **K-L divergence**, also called **Information Divergence**, or **Relative Entropy**. It measures the degree of relevance of a variable to each cluster; the higher the value, the greater contribution of the variable in the cluster.

The user can specify the relevance threshold to show only the variables with relevance greater than the specified threshold.

16.2.7 Renaming Clusters

Once the clusters have been characterized, KnowledgeSTUDIO gives users the option to assign useful names to the clusters for easier recognition and analysis. This can be done using the Rename Clusters button in the toolbar:

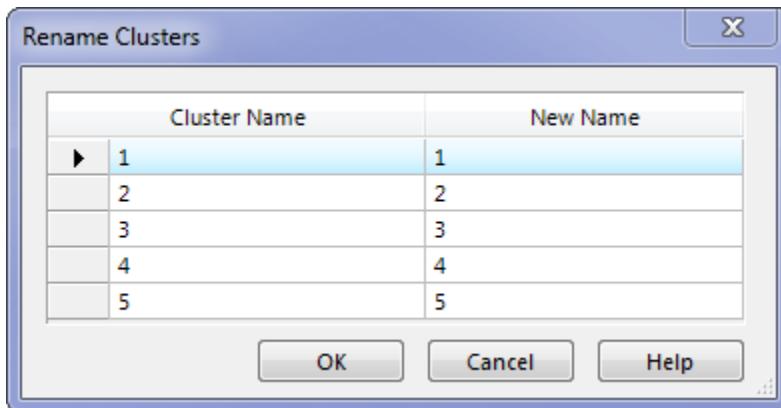
Figure 16.11: Rename Clusters Button



Note that this button is only available from the Cluster Model View.

Once clicked, the user will be presented with a pop-up window to rename any of the existing clusters. Such changes will propagate through the Cluster Model View, and also into the model itself. Importantly, if a cluster model is scored, the updated cluster names will be used in the scoring.

Figure 16.12: Rename Clusters Window

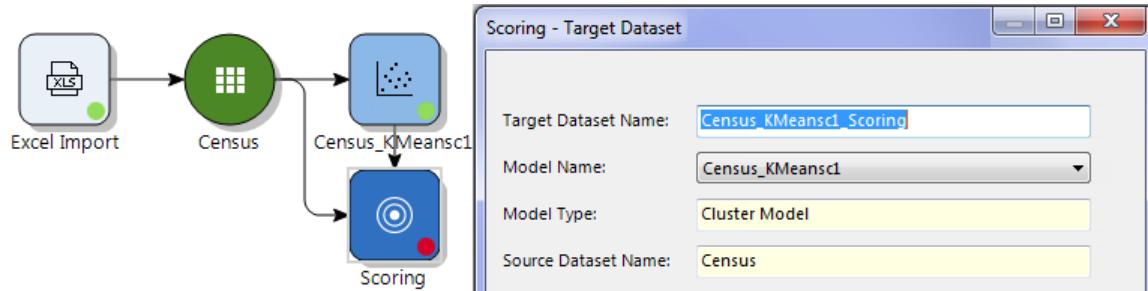


16.2.8 Scoring Data to Further Characterize Clusters

To assess the business value of the results, the model is used to score a dataset and further assessment can take place to determine characteristics and differences.

To score data, add a **Scoring** node from the **Action** palette, and connect the **Cluster Analysis Model Instance**. Dialogs are provided for naming results, field mapping and resulting scored variable selection.

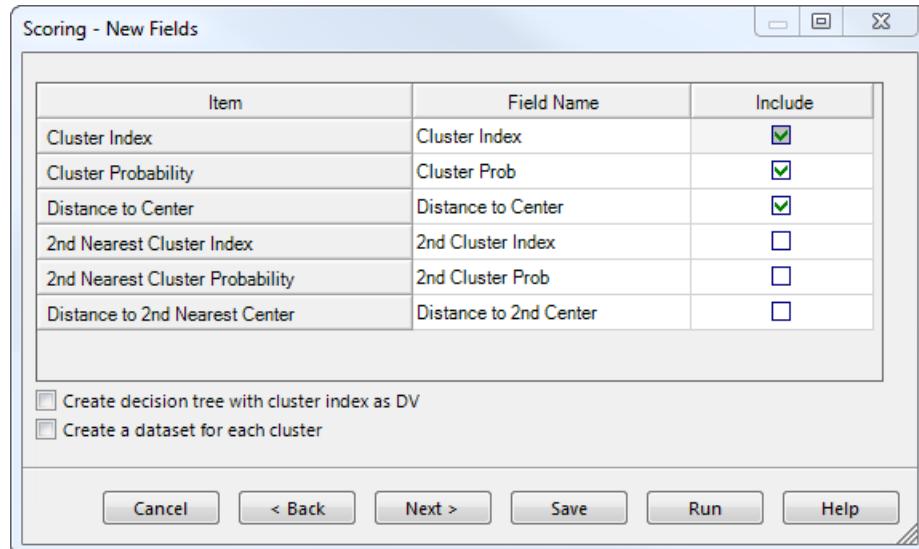
Figure 16.13: Model Scoring



NOTE: The default model used for scoring is the best fit solution. To use a specific model; use the **Model Link** node. Alternatively, right click the model in the **Project Pane** and select the option: **Put on Workflow Canvas**.

Navigating to the **Scoring – Scoring Fields** dialog provides aspects for field and additional output creation.

Figure 16.14: Scoring - New Fields



The **Scoring – New Fields** dialog provides options to generate up to six new fields:

- **Cluster Index** Indices range from 1 to **N**, where **N** = total number of clusters
- **Cluster Probability** The probability for the assigned cluster
- **Distance to Centre** The distance the record is from the cluster centre

Three additional fields are available, identical to those indicated but in relation to the next nearest cluster for each record. Note that the **Cluster Index** is the only mandatory field.

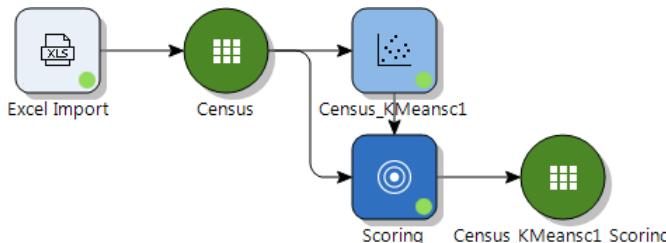
Two additional options are also available from this dialog;

- **Create decision tree with cluster index as DV**
 - Builds a **Decision Tree** with the cluster index field as the **DV**. Using a **Find Split** the tree is grown one level
- **Create a dataset for each cluster**
 - A separate dataset for each cluster is created

NOTE: If these options are selected, output results will appear nested underneath the resulting scored dataset in the **Project Pane**.

Accept the defaults and click **Run** to generate the scored dataset as illustrated.

Figure 16.15: Scored Dataset



Opening the scored dataset reveals nine tabs. The usual eight along with a **Report** tab relaying cluster distribution information, not shown. Viewing the data from **Data** tab reveals three new fields shown in the figure below; namely, **Distance to Center**, **Cluster Prob** and **Cluster Index**.

Figure 16.16: Data Tab

	Customer ID	Cluster Index	Cluster Prob	Distance to Center
1	1	4	0.999793271038387	1.2473185134416
2	2	4	0.996462822209641	0.852486284218445
3	3	2	0.619462036984878	1.30407383870853
4	4	4	0.616161641993463	1.2514071105799
5	5	5	0.999913998726372	1.16767836791925
6	6	4	0.992874037992801	1.50104628150122
7	7	4	0.997902293972344	1.1065517440574
8	8	2	0.99997749363051	1.72049875723355
9	9	5	0.999318577000296	1.49771113938889
10	10	1	0.999999989900106	1.80168894063158

The **Segment Viewer** tab is the most useful as it helps visually identify cluster characteristics by showing the distribution of each variable within each cluster in the form of charts.

Results can be sorted in ascending or descending order of any column. The **Information Value** and **Entropy Value** columns provide additional insight into variables contrasted across clusters.

Here the charts are illustrated in descending order of **Entropy Variance** with the variable *education* as the first variable in the list. Note that this should be reflected in the **Decision Tree** if selected from the **Scoring – New Fields** dialog.

Figure 16.17: Segment Viewer



Assessing the clusters it can be seen that **Cluster 5** is characterized by young, female customers, never married, having their own child. This contrasts with **Cluster 1** that contains predominantly older, married, male customers.

The small cluster with 85 records is characterised as being composed totally of those with a value of Yes for the variable *Response*.

Additional characteristics and points of difference can be derived and the results can be further assessed using **Decision Trees** and **Strategy Trees**.

The model can be deployed as code in one of four coding formats:

- **SQL function**
- **SAS**
- **XML**
- **PMML**

Additionally results can be exported to a range of file formats or to a database using the appropriate node from the **Source** palette.

16.3 Validating a Cluster Analysis Model: Discussion

Validating a cluster analysis takes the same generic approach as validating other model types by assessing results across a development and validation partition.

Validating the results is a matter of determining whether the cluster proportions, statistics and characteristics replicate across the partitions.

This process requires a little more manual intervention as there is no model validation node currently available to apply to a cluster analysis to automatically generate a comparison.

16.4 Summary

This chapter described cluster analysis modelling process using **KnowledgeSTUDIO**, in particular the process of creating, analysing, and scoring cluster models.

As a result of completing this chapter the user should be able to:

- Understand **Cluster Analysis** and its applicability
- Build **Clustering** models using **KnowledgeSTUDIO**
- Analyse and interpret the clusters
- Deploy the **Clustering** model
- Validating a cluster analysis model

Exercises

The data used is the **Census** file.

This can be found by loading the **Census Sample Project** from the **Prepare Sample Data...** dialog found in the **Help** menu. All elements can be deleted, retaining only the **Census** dataset.

1. Create a new project and import the file *Census.xlsx*.
2. Explore the data and familiarize yourself with the variables and distributions.
3. Once the data has been explored and potential predictors identified, create two partitions to develop and test the model.
NOTE: this was not covered in the chapter, however is an appropriate way to evaluate clusters.
4. Using the **Cluster Analysis** option create a model.
 - (a) Use **K-Means** algorithm.
 - (b) Select the variables to use for clustering
 - (c) Specify a cluster range.
5. Once generated, assess the output.
 - (a) Compare cluster size and characteristics, including variable importance.
6. Score the model and from the **Segment Viewer** tab use the *Cluster Index* variable as the segmentation variable.
 - (a) Characterise each segment by exploring its variables
 - (b) Identify the smallest segment. What is the common feature of the records belonging to that segment?
 - (c) **Label** the segments in a meaningful way based on their characteristics.
7. Explore the code generation options for the cluster model. Generate code and save the code as an external file.
8. Validate the cluster model by applying the same clustering algorithm on the **Validation** partition.
 - (a) Compare cluster distances, number of records and the variable contribution within each cluster in the **Development** and **Validation** partitions; both partitions should produce similar results to confirm the validity of the clustering solution.
9. Create a **Decision Tree** with *Cluster Index* as the **Dependent Variable**. Use the **Decision Tree** to explore the significant variables that caused the clusters to be created.

- (a) How do they compare with the cluster analysis results?
10. Use a **Strategy Tree** adding additional measures to determine an appropriate deployment strategy. Refer to **KnowledgeSTUDIO** manual if necessary.
 - (a) Create two **Key Performance Indicators**: *Profitability* and *Loyalty* using *capital-gain* & *num-products* respectively.
11. Create code using the available nodes from the **Action** palette.

Chapter 17: Principal Component Analysis

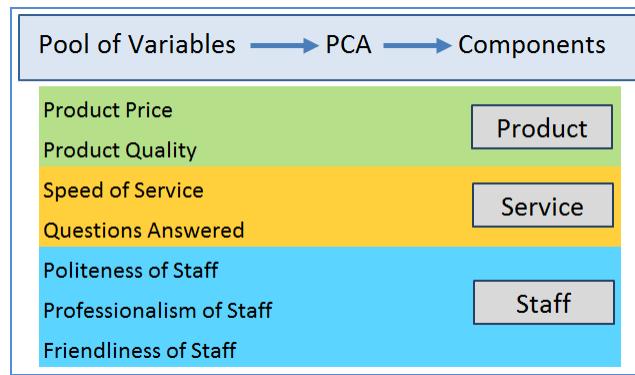
17.1 Introduction

Principal Component Analysis, PCA, is a statistical technique that applies transformations to a set of possibly correlated variables to derive a set of linearly uncorrelated variables called **Principal Components**.

The technique can be used for two main purposes:

- To understand the underlying structure of a set of variables
- To reduce a set of variables to a smaller subset; for modelling or for further exploration

Figure 17.1: PCA



Principal Component Analysis shares common ground with other data reduction techniques and is frequently mentioned alongside **Factor Analysis**.

In fact the two can be combined as **Principal Component Factor Analysis**. **Principal Component Analysis** focuses on the **Variance** of a set of observed variables while **Factor Analysis** focuses on the relationships between the variables; their **Covariances**.

NOTE: The starting point for a **Principal Component Analysis** is a **Covariance Matrix**, however a **Correlation Matrix** is generally referred to as it is more commonly understood and widely known.

Principal Component Analysis focuses on the Variances of each variable, the diagonal. **Factor Analysis** focuses on the relationships between variables; their Covariances, the off-diagonal.

As a result of this difference, traditionally **Principal Component Analysis** has been used as a data reduction technique to extract **Principal Components** and **Factor Analysis** as a data understanding techniques to extract **Latent Variables**.

As **Factor Analysis** applies an additional assumption that some variability cannot be explained by the extracted components, the overall amount of variance available as a starting point is smaller.

Regardless of their conceptual and mathematical differences, both can be used in general for the same purposes in Data Mining. With any extraction method the questions that a good solution should try to answer are:

- How many components are needed to represent the observed variables?
- What do the components represent? i.e. how should they be interpreted?

As a result of completing this chapter users should be able to:

- Describe **Principal Component Analysis**
- Describe the differences between **Principal Component Analysis** and **Factor Analysis**
- Run and evaluate a Principal Component Analysis using **KnowledgeSTUDIO**

17.2 Description

Visitors to a retail outlet were asked to rate different aspects of their experience. The correlation matrix in figure shows the relationships between the data gathered.

It is clear there are relationships between two groups of variables. One set: *Polite, Knowledgeable, Efficient* and *Well Stocked, Nice Layout* and *Good Location*.

Figure 17.2: Correlation Matrix of Variables

	Polite	Knowledgeable	Efficient	Well Stocked	Nice Layout	Good Location
Polite	1.0					
Knowledgeable	0.6	1.0				
Efficient	0.7	0.6	1.0			
Well Stocked	0.2	0.1	0.2	1.0		
Nice Layout	0.3	0.2	0.1	0.6	1.0	
Good Location	0.1	0.4	0.3	0.7	0.6	1.0

In some modelling techniques, multicollinearity issues may arise if variables are highly associated. A solution is to apply data reduction, extract the **Principal Components** and use these in place of the original variables in a model.

Principal Component Analysis begins by extracting the **Eigenvectors** and **Eigenvalues** of the covariance matrix of a set of variables. As the variables are standardized, the process is essentially conducted on a correlation matrix where each variable contains 1 unit of variance.

Figure 17.3: Table of Eigenvalues and Eigenvectors

	EigenVector1	EigenVector2	EigenVector3	EigenVector4	EigenVector5	EigenVector6
Polite	0.9	0.4	-0.9	-0.1	0.3	0.5
Knowledgeable	0.8	0.3	-0.1	-0.4	-0.5	0.3
Efficient	0.9	0.5	0.1	-0.3	-0.4	-0.3
Well Stocked	0.7	0.4	0.5	-0.2	-0.3	0.8
Nice Layout	0.6	0.3	0.1	-0.2	-0.2	-0.1
Good Location	0.8	-0.2	0.1	0.1	-0.3	-0.3
Eignevalues	3.5	1.5	0.4	0.3	0.2	0.1

The **Eigenvectors** are the underlying dimensions in the data and the **Eigenvalues** reflect how many units of variability each **Eigenvector** explains.

The **Eigenvectors** and **Eigenvalues** provide information about the structure of the data and are in turn used to derive the set of **Principal Components**.

In this example there are six variables, so in total there are six units of variance to explain. **Eigenvector1** explains 3.5 units or approximately 60% of this.

As one variable equals one unit of variance, a rule of thumb is to extract only **Eigenvectors** explaining at least 1 unit of variance as this must means they explain variance from more than one variable.

An alternative is to base component extraction on the total proportion of variability explained. For example: extract the first **n** components that together explain at least **x** percent of the total variance.

This is known as the **Variance Cutoff** method and an available option in **KnowledgeSTUDIO**. In this example, if the **Variance Cutoff** value is set at 70% the first two eigenvectors are extracted and two **Principal Components** created.

Figure 17.4: Extracted Components Loadings Matrix

	PComponent1	PComponent2
Polite	0.9	0.4
Knowledgeable	0.8	0.5
Efficient	0.8	0.4
Well Stocked	0.7	-0.5
Nice Layout	0.6	-0.3
Good Location	0.7	-0.4
% Variance	52.0	29.0

The original set of variables is displayed as a correlation matrix. The correlation between each variable and extracted component is referred to as a **Loading**.

At this stage the extracted components should be interpreted in relation to the variables that load onto them and if any variable has low loadings or loadings that are similar across components can be removed

and the analysis re-run.

In this example, all variables load heavily onto the first component which explains the bulk of the variability at 52%. The second component explains a lot less variability, but notice that variables load onto it in generally the same way and to the same degree.

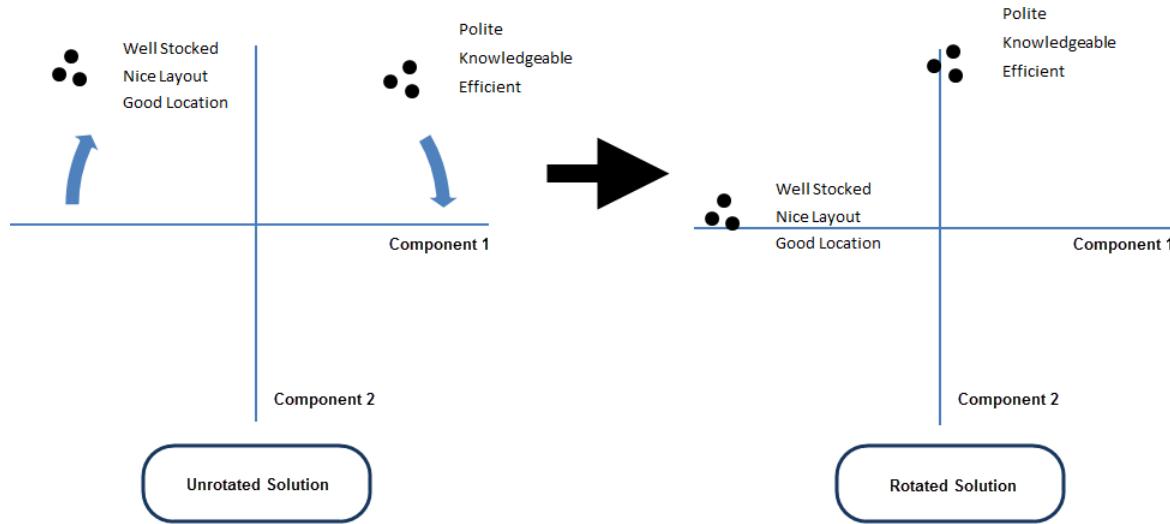
This is a common occurrence in any data reduction methodology: there may be little clarity in distinguishing and naming factors and all variables load most heavily onto the first component. The answer to this is to rotate the solution.

This is a mathematical transformation of the results that redistributes the variance associated with each component while maintaining **Orthogonality**.

There are many ways to rotate a solution and **KnowledgeSTUDIO** provides **Varimax**. A method that attempts to more equally distribute the total amount of variance accounted for across the components.

Rotating the solution can clarify the relationships between components and variables and allow for more appropriate names to be applied to the extracted components

Figure 17.5: Rotating the Solution



As can be seen from figure 17.5, if an unrotated solution provides little clarity in understanding and distinguishing components, then rotating the solution might clarify and aid in improving interpretability of the extracted components.

Rotated solutions retain the same amount of variability while spreading variability explained more evenly among the components. Results are presented as before as loadings.

Figure 17.6: Rotated Solution Loadings Matrix

	PComponent1	PComponent2
Polite		0.9
Knowledgeable		0.8
Efficient		0.8
Well Stocked	0.9	
Nice Layout	0.8	
Good Location	0.8	
% Variance	42.0	39.0

As a result of the rotations, relationships between the set of variables and each component is clarified. **Component1** clearly relates to *Convenience*, and **Component2** to *Service*.

Low loadings can also been removed from the results to de-clutter.

Figure 17.7: Scores Matrix

	PComponent1	PComponent2
Polite	0.035	0.114
Knowledgeable	0.047	0.567
Efficient	0.271	0.815
Well Stocked	0.550	-0.016
Nice Layout	0.534	-0.024
Good Location	0.540	-0.003

Finally a set of linear equations relating variables and components is generated, again, in the form of matrix, but can easily be expressed using more familiar representations.

PComponent1

$$\begin{aligned}
 &= 0.035 * \text{Polite} + 0.047 * \text{Knowledgeable} + 0.271 * \text{Efficient} + 0.55 \\
 &\quad * \text{Well Stocked} + 0.534 * \text{Nice Layout} + 0.54 * \text{Good Location}
 \end{aligned}$$

PComponent2

$$\begin{aligned}
 &= 0.114 * \text{Polite} + 0.567 * \text{Knowledgeable} + 0.851 * \text{Efficient} - 0.016 \\
 &\quad * \text{Well Stocked} - 0.024 * \text{Nice Layout} - 0.003 * \text{Good Location}
 \end{aligned}$$

The resulting equations can be used to score the data adding scores for each case based on the variable values. As all variables are standardized the resulting equations relate to the change, in standard deviation units, in the component, for a one standard deviation unit change in the observed variable.

These can be used for interpretation and assessing the relative importance of each predictor for each component.

The steps below outline the process and points to note when undertaking a principal component analysis using **KnowledgeSTUDIO**:

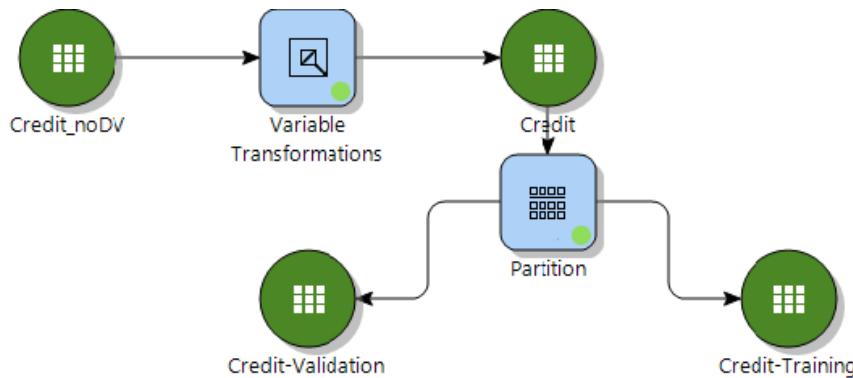
- Prepare and explore the data
 - Data requirements for **Principal Component Analysis** in **KnowledgeSTUDIO** requires that all input variables are numeric
- Assess overall number of components needed to adequately represent the data
- Assess the variable loadings; rotate the solution if necessary and interpret the components
 - This is an important step in the process and if components cannot be adequately understood and named appropriately in regard to the variables that load onto it, should be removed
- Score the data and use the results in other modelling techniques or for further exploration

17.3 Demonstration

Principal Component Analysis can be used in a variety of circumstances to aid in data interpretation, exploration and model building. A common use is to reduce a large set of variables to a smaller subset. The extracted components are then used in place of the original set of variables when modeling.

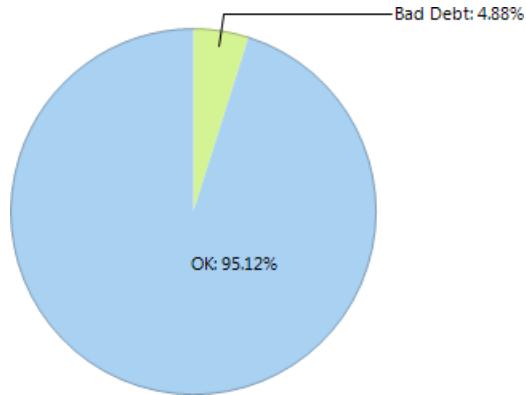
This demonstration builds a **Logistic Regression** model and uses **PCA** to address multicollinearity and uses the dataset; **Credit**. This can be found by loading the **Credit** sample project from the **Prepare Sample Data...** dialog found in the **Help** menu. All elements can be deleted, retaining only the **Credit** dataset and its partitions as illustrated in figure 17.8.

Figure 17.8: Initial Project



Viewing the dataset it can be seen it contains information from the financial services sector including some demographics. A limited number of these variables shall be used in the demonstration.

The **Dependent Variable** for the initial **Logistic Regression** is **Status2**. This contains two values: **OK** and **Bad Debt** distributed as illustrated in figure 17.9.

Figure 17.9: *Status2*

Add a **Logistic Regression** model node to the **Workflow** and connect to the **Credit-Training** partition, not shown. The **Dependent Variable** is ***Status2*** with the target category set to ***OK***. The selection method **Must include all selected variables** is chosen and seven independent variables are included:

- *age*
- *Avg Monthly Balance*
- *Credit Line*
- *Recommended Line*
- *Total Amount Purchased*
- *Total Service Charge Revenue*
- *Year at Address*

Run the model and open the results. Although the **Entropy Explained** is high at .69395, the **Variance Inflations Factors** table from the **Results** tab shows some signs of multicollinearity.

Figure 17.10: Variance Inflation Factors

Variance Inflation Factors	
Variable	Value
[Credit Line]	1.045
[Recommended Line]	1.125
[Year at Address]	1.110
[Avg Monthly Balance]	7.182
[Total Amount Purchased]	1.473
[Total Service Charge Revenue]	6.687
[age]	1.140

Running a correlation matrix of the variables further clarifies the interrelatedness of the variables included in the regression model.

Figure 17.11: Independent Variables

	Credit Line	Recommended Line	Year at Address	Avg Monthly Balance	Total Amount Purchased	Total Service Charge Revenue	age
Credit Line	1	0.06694	0.1905	-0.02762	0.13283	-0.0221	0.28084
Recommended Line	0.06694	1	0.19524	0.02772	0.0344	0.02089	0.24648
Year at Address	0.1905	0.19524	1	-0.08898	-0.04486	-0.08565	0.32373
Avg Monthly Balance	-0.02762	0.02772	-0.08898	1	0.53574	0.92247	-0.11512
Total Amount Purchased	0.13283	0.0344	-0.04486	0.53574	1	0.45917	-0.05193
Total Service Charge Revenue	-0.0221	0.02089	-0.08565	0.92247	0.45917	1	-0.10954
age	0.28084	0.24648	0.32373	-0.11512	-0.05193	-0.10954	1

It can be seen that *age* and *Year at Address* are related and, to a greater degree, variables relaying financial detail; *Average Monthly Balance*, *Total Amount Purchased* and *Total Service Charge Revenue*.

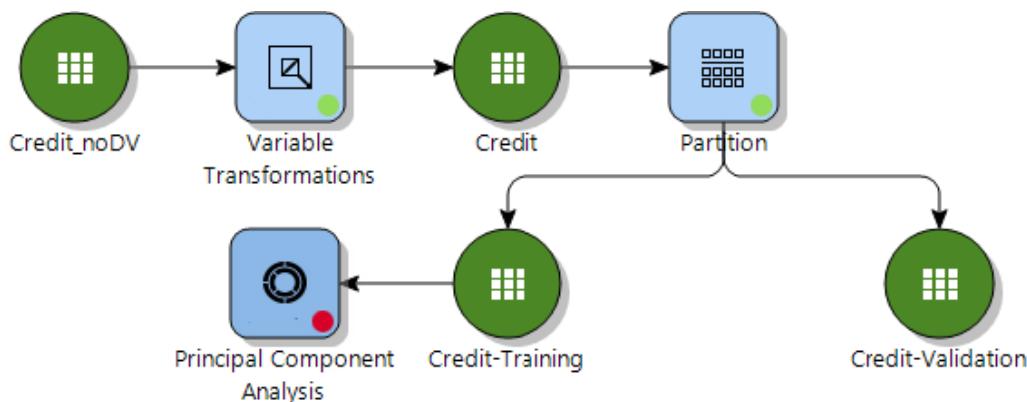
NOTE: Correlations greater than or equal to 0.3 and less than or equal to -0.3 have been highlighted.

A **Principal Component Analysis** is used to assess whether the variables can be explained and replaced by a smaller number of components.

17.3.1 Principal Component Analysis in KnowledgeSTUDIO

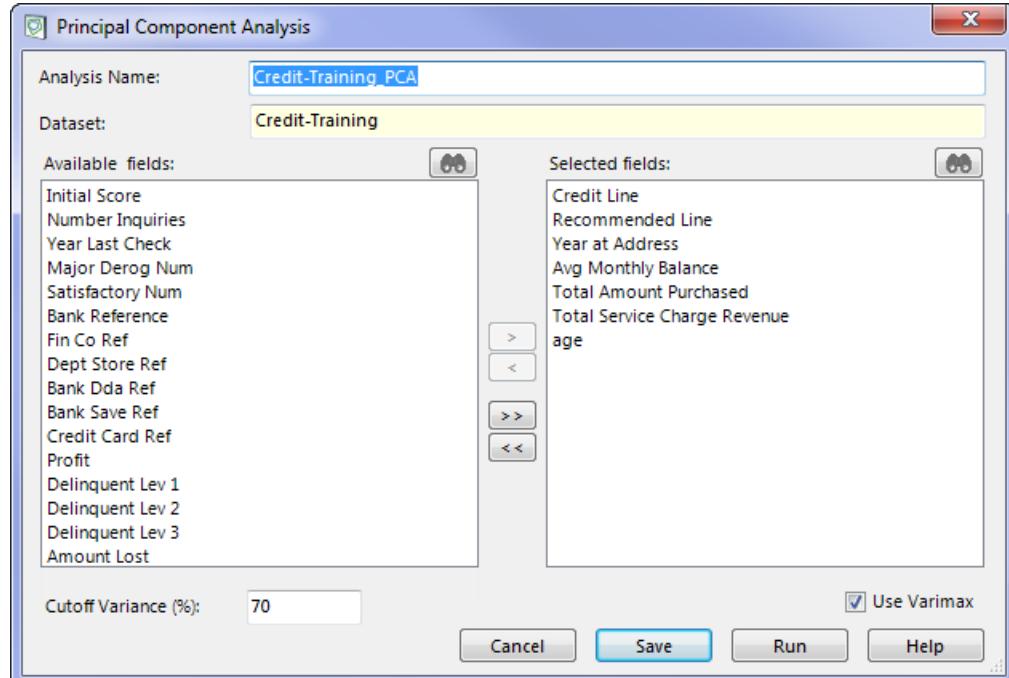
The **Principal Component Analysis** node is located in the **Model** palette. Add to the **Workflow** and connected to the **Credit-Training** partition as illustrated in figure 17.12.

Figure 17.12: Adding a Principal Component Analysis Node



To access options, either double click the **Principal Component Analysis** node or right click and select **Modify**.

Figure 17.13: Principal Component Analysis Dialog



All options are conveniently contained in one dialog and explained below.

Table 17.1: Principal Component Analysis - Dialog Options

Option	Description
Analysis Name	Assign a name to the results
Dataset	The focus of the analysis. Determined by connections
Available & Selected Fields	Fields in the dataset and those used in the analysis respectively
Cutoff Variance (%)	Threshold setting for cumulative % of variance explained by candidate components
Use Varimax	Select to apply a Varimax rotation to the solution

Seven variables are included in the model, all other options are left at their defaults. Click **Run** to generate results.

Accessing the results reveals four tabs: **Summary**, **Eigenvectors**, **Loadings**, **Scores**.

The **Summary** tab lists the dataset, variables and settings. The **Eigenvectors** tab shows the extracted components, the relationship between each and the input variables along with associated **Eigenvalues**.

Figure 17.14: Eigenvectors Tab

	Eigen1	Eigen2	Eigen3	Eigen4	Eigen5	Eigen6	Eigen7
age	0.23851	0.32413	0.46047	-0.78993	-0.04512	0.00139	-0.00102
Avg Monthly Balance	-0.55578	0.25566	0.08097	0.00089	-0.31283	-0.72203	0.00096
Credit Line	0.23748	0.56952	-0.32198	0.12066	-0.03571	-0.00257	-0.70692
Recommended Line	0.23679	0.57011	-0.32118	0.11931	-0.03549	0.00004	0.70729
Total Amount Purchased	-0.42138	0.26914	0.05873	-0.03169	0.86184	0.05279	-0.00061
Total Service Charge Revenue	-0.54893	0.24857	0.07868	0.00581	-0.39342	0.68984	-0.00162
Year at Address	0.20706	0.21442	0.75163	0.58837	0.00472	-0.00049	0.00015
Eigenvalues	2.43098	2.30362	0.99851	0.65209	0.53473	0.07604	0.00403

As there are seven variables there are seven units of variance to account for. Table 17.15 below illustrates the amount of variability explained by each **Eigenvector**.

Figure 17.15: Eigenvectors, Eigenvalues and Percent Explained

Eigenvector	Eigenvalues	Percent Explained	Cumulative
Eigen1	2.430	34.7%	34.7%
Eigen2	2.303	32.9%	67.6%
Eigen3	0.998	14.3%	81.9%
Eigen4	0.652	9.3%	91.2%
Eigen5	0.534	7.6%	98.9%
Eigen6	0.076	1.1%	99.9%
Eigen7	0.004	0.1%	100.0%
Total Units	7		

NOTE: The percentage explained may vary slightly when rotations are applied.

As can be seen from the table the first three components explain in excess of 80% of the variability of the original set of variables. As a result of the **Cutoff Variance** setting these three Eigenvectors are extracted, rotated and presented in the **Loadings** tab.

Figure 17.16: Loadings Tab

Absolute Display Threshold: 0.4			
	PComponent1	PComponent2	PComponent3
age			0.70183
Avg Monthly Balance	0.95082		
Credit Line		0.98120	
Recommended Line		0.98117	
Total Amount Purchased	0.77201		
Total Service Charge Revenue	0.93624		
Year at Address			0.87859
Variance	2.38168	2.03023	1.32120
% Variance	34.02395	29.00332	18.87434

The **Loadings** are the correlations between each component and the variables included in the analysis. The table in figure 17.16 has been modified to only show those correlations in excess of .4.

This de-clutters the table enabling a clear understanding of the relationships. At this stage components should be interpreted in relation to the variables that load onto them. In this example:

Pcomponent1 relates to *Purchases*, **PComponent2** relates to **Credit Line** and **PComponent3** relates to **Demographics**. All variables are positively related to their respective component.

NOTE: It is generally recommended to extract components with eigenvalues greater than or equal to 1 (or extremely close to 1). The component can then be said to explain variability from more than one full input variable, and the goal of PCA is to use fewer variables to contain more of the variability. This does not mean that components with eigenvalues less than one should always be excluded, for example a component with an eigenvalue of 0.9 may explain 30% of the variability of three separate variables, which is still a major gain on having to include all 3 raw variables.

Finally, the **Scores** tab illustrates the regression coefficients for each variable for each component.

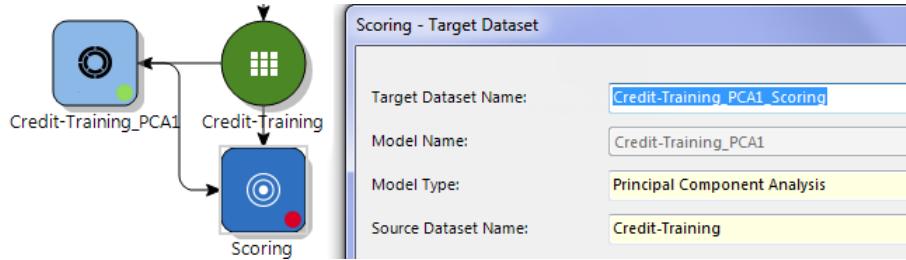
Figure 17.17: Scores Tab

	PComponent1	PComponent2	PComponent3
age	0.00136	0.00065	0.03617
Avg Monthly Balance	0.00106	-0.00006	0.00005
Credit Line	-0.00002	0.00130	-0.00024
Recommended Line	-0.00002	0.00129	-0.00024
Total Amount Purchased	0.00050	0.00003	0.00005
Total Service Charge Revenue	0.00510	-0.00033	0.00024
Year at Address	0.00515	-0.02337	0.09091
Constant Coefficient	-0.81049	-1.69884	-1.68551

The magnitude of the coefficients corresponds to their loading for that component. For example the largest coefficients for **PComponent3** are *Years at Address* and *age*.

The resulting components can be created as new variables and used in place of the original set of variables in the model scenario introduced earlier. This is achieved by scoring the data with the resulting model via a **Scoring** node.

Figure 17.18: PCA Scoring



Navigating to the **Scoring - Scoring Fields** dialog provides the capability to assign names to the components.

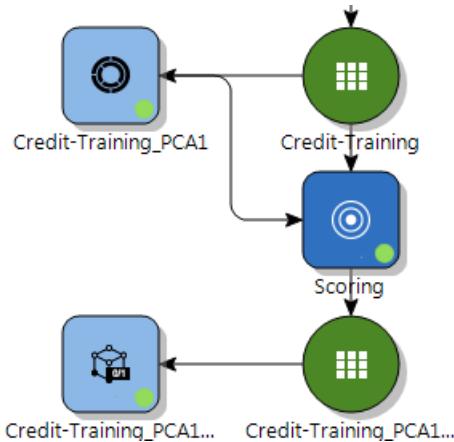
Here the following names are assigned:

- PComponent1 is renamed to *Purchases*
- PComponent2 is renamed to *Credit_Line*
- PComponent3 is renamed to *Demographics*

This dialog also provides the capability to export expressions as either **Altair Expression Format (XML)** or **Plain Text**.

Once the data has been scored and the new fields added, they can be used in place of the original set in the **Logistic Regression**. Figure 17.19 illustrates the setup.

Figure 17.19: Re-run Model



Status2 is set as **Dependent Variable & Target Category = OK**.

The variable selection method is; **Must include all selected variables** with the following variables chosen: *Credit_Line, Demographics and Purchases*.

Run the model and once complete, view the **Logistic Regression** results.

Figure 17.20: Training Results

Model Parameters	
Dependent Variable:	Status2
Independent Variable(s):	Credit_Line Demographics Purchases
Training Results	
Entropy Explained:	0.095246
Records correctly predicted:	19,393 / 20,417 (94.984572%)

The Results tab of the **Logistic Regression** output show an **Entropy Explained** value of 0.09.

This is low in comparison to the previous model with a value of 0.69344, however the proportion and number of Records correctly predicted is similar. Selecting **Currently Selected Sequence** from the **Output to View:** dropdown gives further model results.

The coefficients can be interpreted as normal and the variance inflation factors are now comfortably acceptable.

Figure 17.21: Rerun Model - Variance Inflation Factors

Variance Inflation Factors	
Variable	Value
[Purchases]	1.007
[Credit_Line]	1.007
[Demographics]	1.005

If deemed acceptable the **Logistic Regression** model can be deployed using the available methods.

17.3.2 Summary

This chapter detailed KnowledgeSTUDIO functionality for **Principal Component Analysis**. On completion of this chapter the user should be able to:

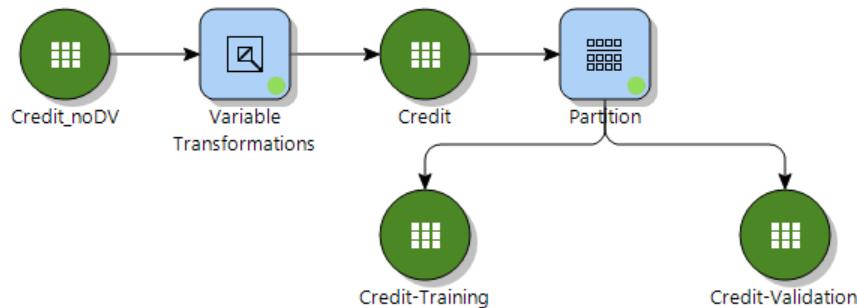
- Describe **Principal Component Analysis**
- Describe the differences between **Principal Component Analysis** and **Factor Analysis**
- Run and evaluate a Principal Component Analysis using **KnowledgeSTUDIO**

Exercises

This demonstration uses the dataset **Credit**.

This can be found by loading the **Credit** sample project from the **Prepare Sample Data...** dialog found in the **Help** menu.

All elements can be deleted, retaining only the **Credit** dataset and its partitions as illustrated.



The dataset contains information from the financial services sector. A variable list is illustrated.

#	Field Name
1	Credit Line
2	Initial Score
3	Recommended Line
4	AgentID
5	Number Inquiries
6	Worst Rating Des
7	Year Last Check
8	Major Derog Num
9	Satisfactory Num
10	Year at Address
11	Bank Reference
12	Fin Co Ref
13	Dept Store Ref
14	Bank Dda Ref
15	Bank Save Ref
16	Credit Card Ref
17	Region
18	Avg Monthly Balance
19	Total Amount Purchased
20	Total Service Charge Revenue
21	Profit
22	Delinquent Lev 1
23	Delinquent Lev 2
24	Delinquent Lev 3
25	Amount Lost
26	age
27	Residential Status
28	Status
29	Status2

The purpose of the exercise is to develop a **Logistic Regression** model to predict the variable *Status2* and use **PCA** as means to not only reduce the number of variables but also to understand relationships.

1. Explore the data using the profiling features available in **KnowledgeSTUDIO**.
2. Explore variable relationships with *Status2*.
3. Use the Correlations tab to assess relationships between continuous variables.
4. Assess an initial set of candidate predictors for use in a model to predict *Status2*. **Decision Trees** can be used as a means to do this. Once identified save the candidate variables as a **Variable List**.
5. Develop a **Logistic Regression** with *Status2* as the **Dependent Variable**. Include any variable deemed to be relevant.
6. Assess the model results. What is the entropy value? Are there any signs of multicollinearity?
7. If so, re-run the model and replacing the initial candidate variable list with a smaller subset. Use a **Principal Component Analysis** node from the **Model** palette.
8. How many components are extracted? How much variability do they explain? Can they be interpreted?
9. Create the resulting components as new variables and score the dataset **Credit**.
10. Re-run the **Logistic Regression** model using only the components used?
11. What is the **Entropy** value and what proportion of records are correctly predicted? How do these values compare to the previous model?
12. Use the **Model Analyser** to further assess and compare the models.
13. Can you think of any other ways that **PCA** may be applied to this data? Can the data be clustered? If so, can **PCA** be used to reduce the number of variable used?
14. Time permitting: Re-run the analysis and only replace highly correlated variables with a component. Does this have an impact on the **Entropy Explained**?

Chapter 18: Market Basket Analysis

18.1 Introduction

Also called, *Affinity Analysis* or *Association Analysis*, **Market Basket Analysis (MBA)** is a data mining technique for determining associations between items.

Items can be anything; processes, activities, products, services etc. The main thrust of any analysis in business is to assess, in general, whether items are purchased together.

Once associations have been determined upsell, cross sell, promotional and other strategies can be implemented.

Other possible applications of association rules include: financial services, insurance, health sciences, fraud detection and other areas where identifying patterns of events or behaviour from transactional data is required.

Market Basket Analysis was first used in retail to understand the purchase behaviour of customers, basically to assess what is in their baskets. Based on this information, new products that may appeal can be promoted based on basket items and their associations.

For example, a retailer finds that items **A**, **B** and **C** are generally purchased together. A cross-sell strategy might be to offer item **C** to those who purchased only items **A** and **B**.

Alternatively, a communications provider might find that there is association between branded products and upgrades, that is, those who have more products from the same brand, generally have more expensive handsets.

This is an ideal upsell opportunity when more expensive version of the same products can be promoted based on the associations with other items.

This chapter will describe *MBA* and also provide an illustrative demonstration of the application and use of **Market Basket Analysis**. Exercises are also provided at the end of the chapter.

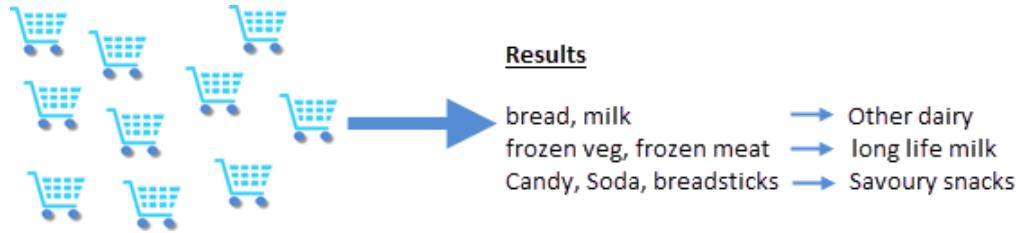
As a result of completing this chapter, users should be able to:

- Describe **Market Basket Analysis**
- Develop a market basket analysis model using **KnowledgeSTUDIO**
- Deploy **Market Basket Analysis** results

18.2 Description

Market Basket Analysis is a means to assess associations between itemsets: processes, products, services and more, and was first used in retail to assess whether products in individual baskets “associated” with other products, in other words, did knowledge of one product being in a basket imply information about other products being in the same basket.

Figure 18.1: Basket Analysis



Once determined, strategies can be designed around the results. For example for those who purchase *bread* and *milk*, other dairy products could be discounted if purchased alongside these products.

Promoting additional items to entice a purchase based on results such as those from **MBA** is referred to as cross-sell or next best offer. Promoting items that are superior versions of current purchases is referred to as an up-sell.

Although **Market Basket Analysis** was derived and is commonly used in retail, it can be applied to good effect in other areas.

For example, in the financial services sector; a bank has low cross-sell rates of its products and would like to increase the uptake of complimentary products in its customer base.

Market Basket Analysis includes many benefits to business, some of which are:

- Increased basket size
- More relevant and personalized offers
- Dynamism and visibility of product
- Better understanding of customer behaviour
- Increased revenue with the right strategy

18.2.1 Itemsets and Association Rules

Any **Market Basket Analysis** aims to derive groups of associated items, these are referred to as itemsets. These are groups of items that are generally found together in the same basket. In figure 18.1 above there are three itemsets:

- *bread, milk*
- *frozen veg, frozen meat*
- *Candy, Soda, breadsticks*

Once a set of associated items, an itemset, has been identified, **Association Rules** or simply **Rules** can be derived. A rule includes an outcome, in the diagram above there are three rules:

- *bread, milk* → *Other dairy*
- *frozen veg, frozen meat* → *long life milk*
- *Candy, Soda, breadsticks* → *Savoury snacks*

The itemsets are also referred to as the **Left Hand** of the rule or **Antecedents**. The outcome is referred to as the **Right Hand** of the rule or the **Consequent**. Items can exist as both **Antecedents** or **Consequents**.

18.2.2 Rule Statistics

Rules are understood by referring to four rule statistics; **Rule Support**, **Confidence**, **Lift** and **Expectation**.

Rule Support

Support is the frequency of an itemset or rule expressed as a percentage. For example if an itemset containing **Item A** and **Item B** has support of 2.5%, then 2.5% of baskets contain both items.

Confidence

Confidence applies to **Rules** and is the ratio of the support of an item set to its output. For example if 10% of baskets contain **Item A** and 5% of baskets contain **Item A** and **Item B**, the confidence for the rule A - B is:

$$\frac{5}{10} = 0.5$$

Confidence varies from 0 to 1 and in simple terms, **Confidence** is a means to assess the proportion of itemset cases the outcome applies to.

In the example above the statistic is interpreted as; 50% of those with **Item A** also have **Item B**.

Lift

Lift is also referred to as importance and is the ratio of the support for the **Rule** divided by the support of the individual items. This, is a ratio of probabilities, where the numerator is the probability of both evident in a basket. The denominator is the product of the support for the individual item.

In the example above, given the rule support is 5%, the Support for **Item A** and **Item B** are 10% and 25% respectively, the **Lift** is calculated as:

$$\left(\frac{5}{10 * 25} \right) * 100 = 2$$

The resulting **Lift** value is interpreted as: the probability of finding someone with **B** is twice as probable if they have **A** in comparison to a randomly selected basket.

In laymans terms: twice as likely to find **Item B** if **Item A** is present, in comparison to selecting a basket at random.

Expectation

A simple statistic referring to the basket frequency of the **Right Hand** of the rule; the outcome. All of these statistics can be used when assessing rules. The most focus is applied to the **Support**, **Confidence**

and **Lift** statistics.

Threshold values can be set for **Support**, **Confidence** and the maximum number of items that can appear in the **Left Hand** of the rule.

18.2.3 Market Basket Analysis with KnowledgeSTUDIO

Market Basket Analysis in KnowledgeSTUDIO requires transactional data.

Data should contain at least one field identifying transactions and a field that identifies the item. An example transactional dataset containing a *TransactionID* variable and an *Item* variable is illustrated in figure 18.2.

Figure 18.2: Transactional Dataset

TransactionID	Item
1	Bread
1	Milk
2	Frozen foods
2	Long life Milk
2	Water

KnowledgeSTUDIO extends traditional **Market Basket Analysis** output and includes a host of additional visual tools to assess results. These will be illustrated in the following demonstration.

18.3 Demonstration

The data used for this demonstration is accessed by loading the Tutorial; **Market Basket Analysis**, from the **Prepare Sample Data** dialog found in the **Help** menu.

This project contains two transactional datasets:

- **Retail Transactions**
- **Transactions To Score**

The **Retail Transactions** dataset contains approximately 50,000 records relating to customer transactions in a supermarket. These will be used as the basis for the model. Rules found are applied to a second dataset: **Transactions To Score**.

The **Retail Transactions** dataset is transactional and contains a *Transaction ID* and *Item Category*, these are essential to building the model and identify the transaction the item belongs to and the item itself.

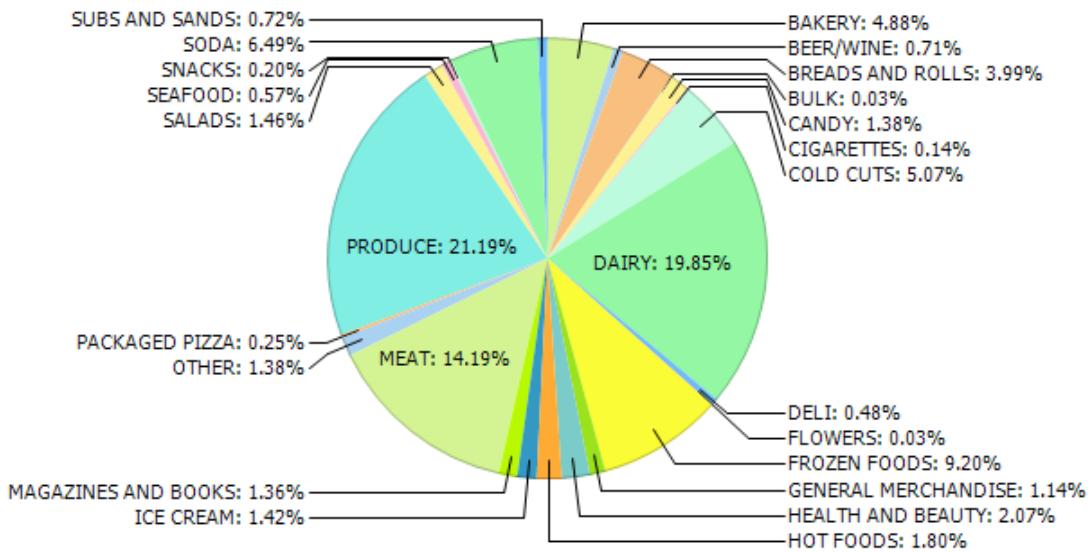
Figure 18.3: Retail Transactions Dataset -Showing Three Variables

	Customer ID	Transaction ID	Item Category
1	2474	3723404	MAGAZINES AND BOOKS
2	2476	3723406	FROZEN FOODS
3	2480	3723410	BAKERY
4	2484	3723414	MAGAZINES AND BOOKS
5	2485	3723415	MAGAZINES AND BOOKS
6	2485	3723415	BAKERY
7	2487	3723417	CANDY
8	2487	3723417	DAIRY
9	2487	3723417	BAKERY
10	2488	3723418	MAGAZINES AND BOOKS

The dataset contains 50,307 records. There are 10,667 unique transactions performed by 3929 customers on 25 individual items. These figures are derived from the dataset size and **Cardinality** of the fields: *Transaction ID*, *Customer ID* and *Item Category* respectively.

A graph of *Item Category* reveals the most common purchases; *Produce*, *Dairy*, *Meat* and *Frozen Foods*.

Figure 18.4: Distribution of Item Category

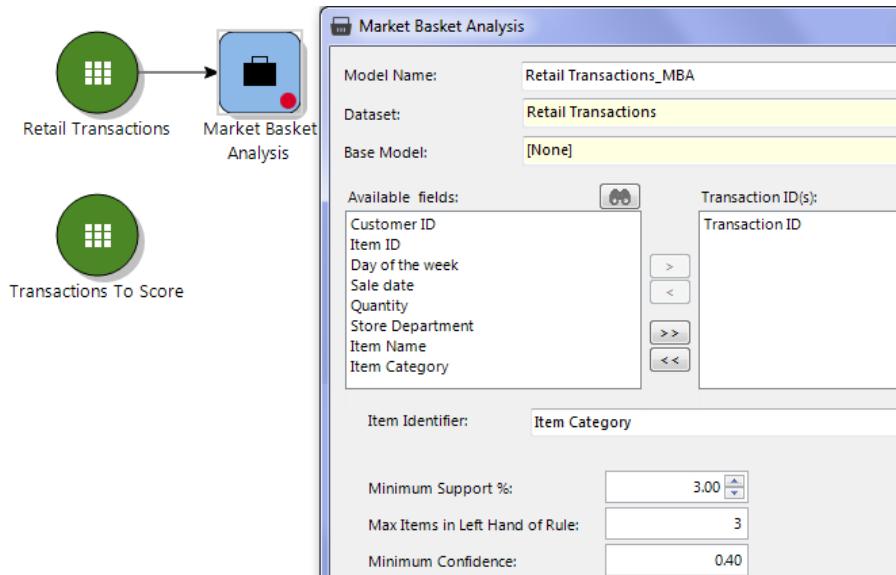


The dataset; **Transactions To Score**, contains 6000 records. To score new records the fields used in the model must be present.

18.3.1 MBA Modelling

A **Market Basket Analysis** modelling node is contained in the **Model** palette. Drag the node to the **Workflow**, connect and open as illustrated in figure 18.5.

Figure 18.5: Adding a Market Basket Analysis Node



All options are conveniently contained in one dialog and explained in table 18.1.

Table 18.1: Market Basket Analysis Options

Option	Description
Model Name	Assign a name to the model
Dataset	Focus of the analysis. Determined by Workflow connections
Base Model	Supply a template model
Available Fields/Transaction ID(s)	Select the variable(s) containing the unique Transaction ID(s)
Item Identifier	Select the variable containing items in each transaction
Minimum Support %	Min proportion of baskets the rule must apply to, to be extracted
Max Items in Left Hand of Rule	Maximum number of items in the left hand of rule
Minimum Confidence	Minimum confidence to extract a rule

The variables **Transaction ID** and **Item Category** are selected as the **Transaction ID(s)** and **Item Identifier** variables respectively. The **Minimum Support** is set at 3.00%. All other options are left at their defaults. Click **Run** to build the model. Once complete view the results.

As can be seen model results are spread across a number of tabs. The output opens automatically on the

Results tab. This provides information in relation to model set up and the number of item sets and rules discovered.

Figure 18.6: Results Tab

Model Parameters	
Transaction IDs in input dataset	Transaction ID
Item identifier field in input dataset	Item Category
Minimum support	3.00%
Maximum items in left hand side	3
Minimum confidence	0.40

Training Results	
Total number of itemsets discovered	96
Total number of association rules discovered	124

Here a total of 96 item sets and 124 rules were generated.

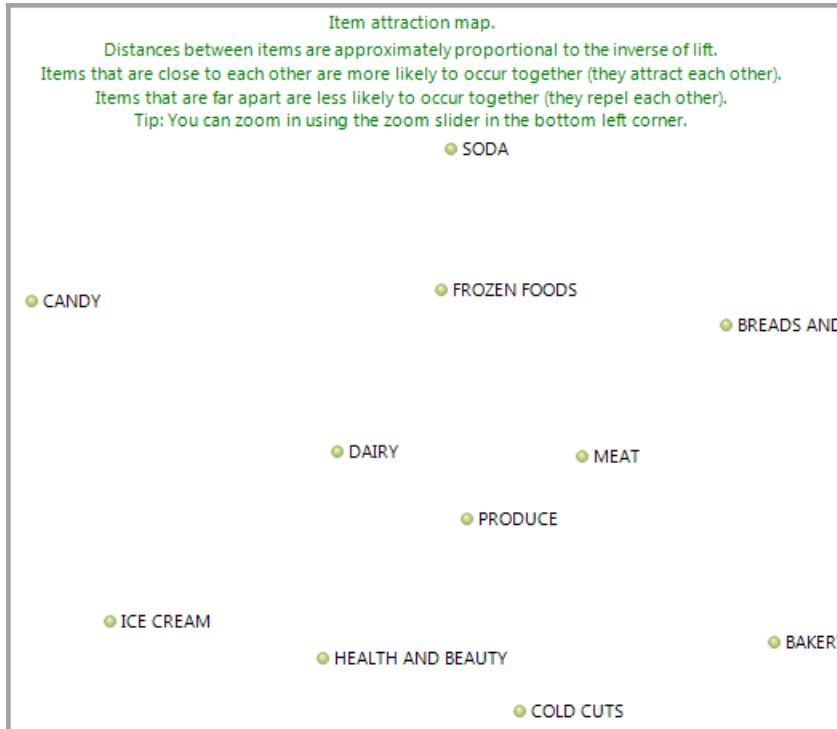
The **Map** tab provides an **Attraction Map** as a means to not only initially assess associations between individual items but also to clarify extracted rules.

The **Association Map** is a standard tool in **Market Research** and displays a graphic representation of the relative distance between individual items.

The closer a group of items are, the more they attract each other, the higher the association and the greater the likelihood they are found in the same basket.

The greater the distance between items, the less likely they are found together, basically they repel.

Figure 18.7: Map Tab

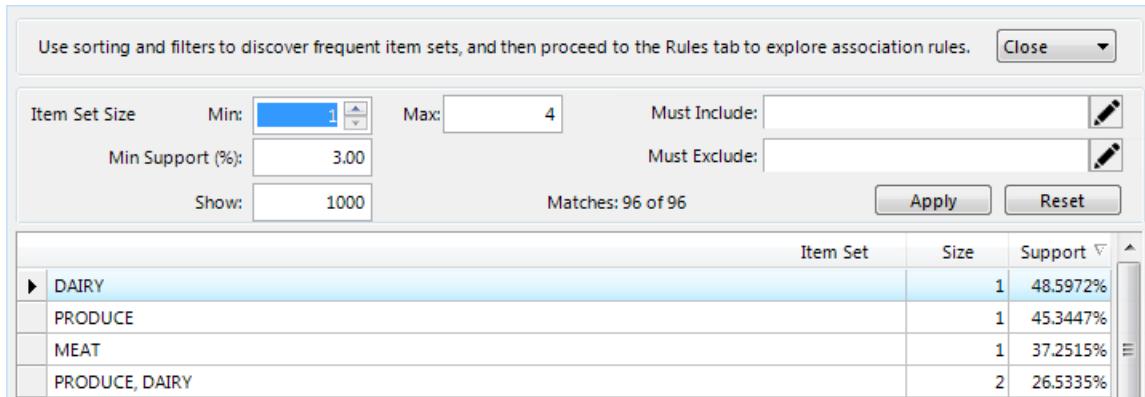


In this example, among other things:

- *FROZEN FOODS, DAIRY, PRODUCE* and *MEAT* are all quite close and attract each other
- *CANDY* and *BREAD AND ROLLS* do not associate. The same can be said for *SODA* and *COLD CUTS*

The **Item Sets** tab lists the extracted **Left Hand** of the rule.

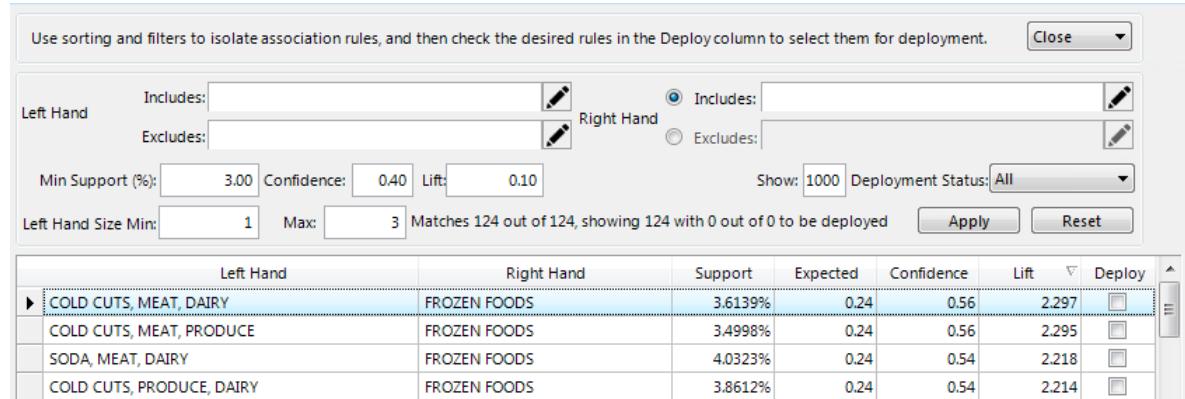
Figure 18.8: Item Sets Tab



DAIRY is the most common followed by *PRODUCE* and *MEAT*. KnowledgeSTUDIO provides filtering and sorting capabilities to better identify interesting item sets.

The **Rules** tab contains the extracted association rules with associated statistics.

Figure 18.9: Rules Tab



Left Hand	Right Hand	Support	Expected	Confidence	Lift	Deploy
COLD CUTS, MEAT, DAIRY	FROZEN FOODS	3.6139%	0.24	0.56	2.297	<input checked="" type="checkbox"/>
COLD CUTS, MEAT, PRODUCE	FROZEN FOODS	3.4998%	0.24	0.56	2.295	<input checked="" type="checkbox"/>
SODA, MEAT, DAIRY	FROZEN FOODS	4.0323%	0.24	0.54	2.218	<input checked="" type="checkbox"/>
COLD CUTS, PRODUCE, DAIRY	FROZEN FOODS	3.8612%	0.24	0.54	2.214	<input checked="" type="checkbox"/>

Sorting and filtering is also provided in the **Rules** tab. There is an additional column; **Deploy**, this allows for easy selection of rules for deployment.

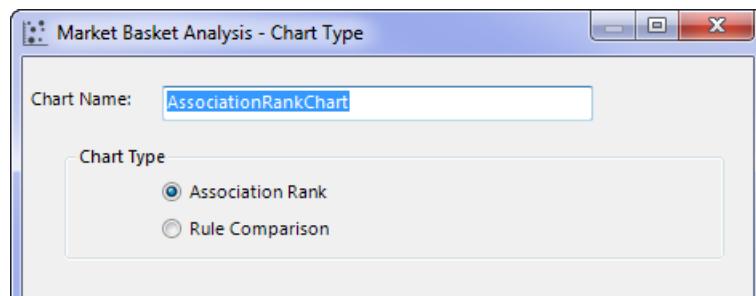
Although rules can be examined and selected from the rules tab, further investigation is possible through the use of charts available from the **Charts** tab.

Two types of chart are available:

- **Association Rank**
- **Rule Comparison**

Charts are added using the add icon:  from the task bar.

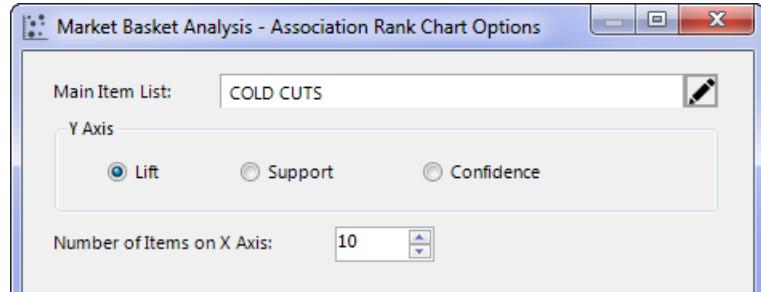
Figure 18.10: Market Basket Analysis - Chart Type



Options are available to specify a name and the chart type. The next dialog provides options related to the chart type selected. The **Association Rank** chart is useful to identify those items most strongly associated with one or more selected items in the form of a bar chart.

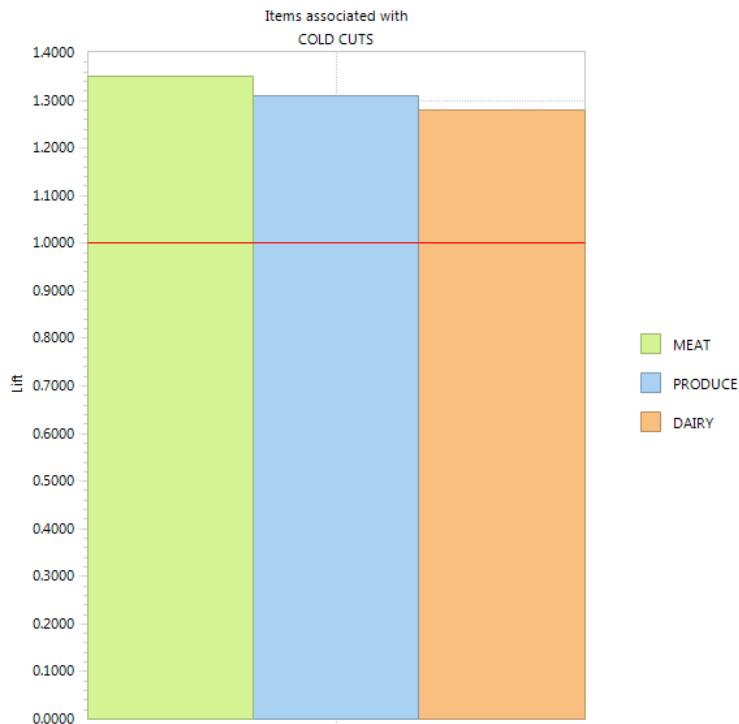
The *y-axis* can display **Lift**, **Support** or **Confidence** statistics and limitations can be put on the maximum number of items to display on the *x-axis*. More than one item can be chosen.

Figure 18.11: Market Basket Analysis - Association Rank Chart Options



Click **Finish** to create the chart.

Figure 18.12: Association Rank Chart



Results are easily interpreted. For each *x-axis* item, if the bars are above the reference point for item independence; the red line, then there is an association, if the bar is below the red line, the items repel.

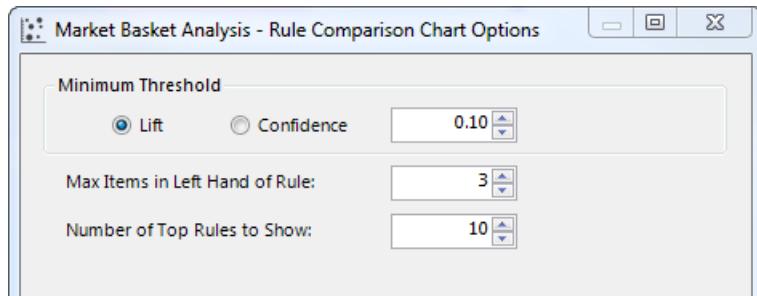
For the item *COLD CUTS*, the model has identified *MEAT*, *PRODUCE* and *DAIRY*. As these items are above the red line, they each associate with *COLD CUTS*.

Rule Comparison charts produce a bubble chart to compare rules. Options are available to:

- Illustrate the **Lift** or **Confidence** statistics, and provide a minimum threshold for appearance on the graph
- Max no. items in **Left Hand** of the rule

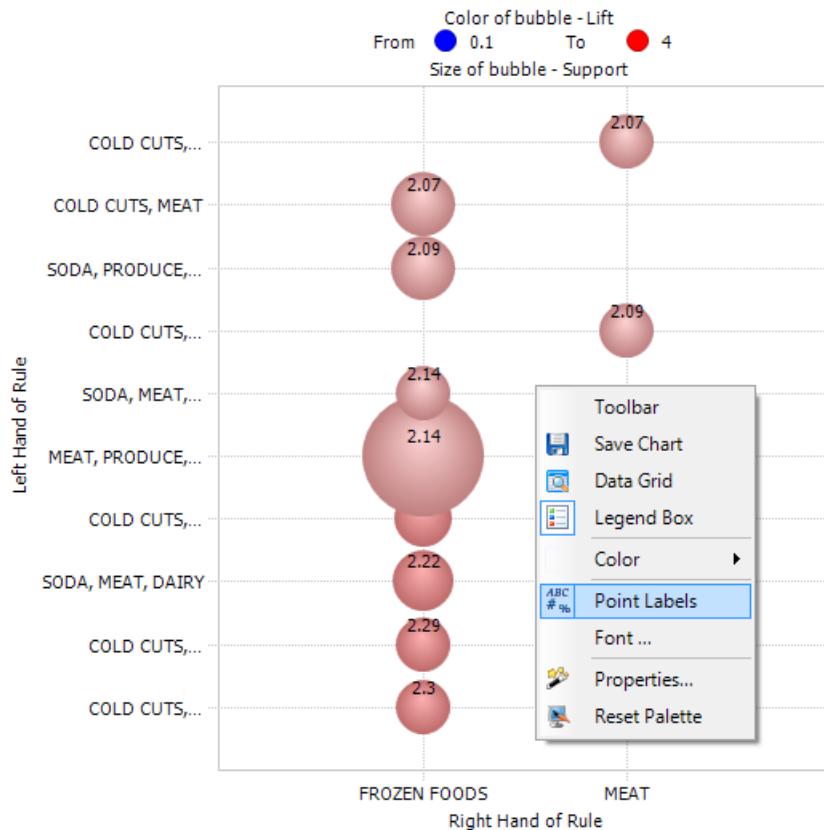
- Limit the number of rules shown (Top N rules)

Figure 18.13: Market Basket Analysis - Rule Comparison Chart Options



The **y-axis** displays the **Left Hand** of the rule and the **x-axis** displays the **Right Hand** of the rule. The bubble size relates to the rule **Support** and the colour depth relates to **Lift**.

Figure 18.14: Rule Comparison Chart



Here the rule with greatest support is: *MEAT, PRODUCE, DAIRY - FROZEN FOODS*. The rule with greatest lift is: *COLD CUTS, MEAT, PRODUCE - FROZEN FOODS*.

NOTE: The **Rule Comparison** chart does not by default display lift values. To show **Lift** values, right click anywhere on the chart and select the option **Point Labels**.

18.4 MBA Model Deployment

Deploying model can be performed in one of two ways in **KnowledgeSTUDIO**:

- Score an existing dataset
- Generate code for use on other platforms

The rules of a **Market Basket Analysis** model consists in applying the rules to new data to generate recommendations. In terms of association rules, the task of producing product recommendations can be formulated as follows:

- For each customer, find all association rules whose Left Hand sides are subsets of at least one basket
- Select top 5 most important and recommend the Right Hand side item

This is ultimately the most thorough means to generate a list of recommendations. **KnowledgeSTUDIO** can generate multiple recommendations and rank these the **Lift** value.

First of all create a **Model Instance** and from the **Rules** tab, select the rules for deployment.

In this example the top 10 rules are selected based on **Lift**. Note that across these 10 rules there are only two recommendations; *FROZEN FOODS* or *MEAT*.

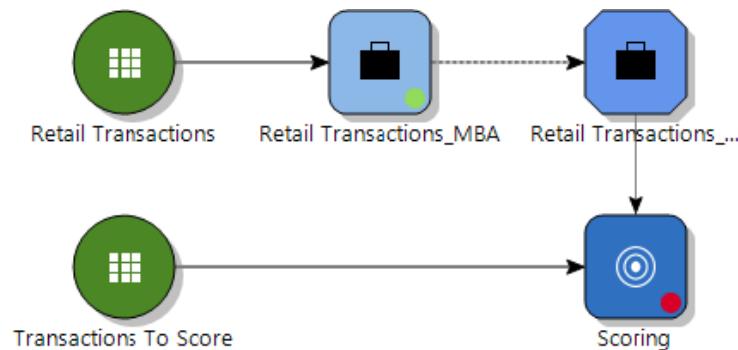
Figure 18.15: Selecting rules to deploy

Left Hand	Right Hand	Support	Expected	Confidence	Lift	▼	Deploy
COLD CUTS, MEAT, DAIRY	FROZEN FOODS	3.6139%	0.24	0.56	2.297	<input checked="" type="checkbox"/>	
COLD CUTS, MEAT, PRODUCE	FROZEN FOODS	3.4998%	0.24	0.56	2.295	<input checked="" type="checkbox"/>	
SODA, MEAT, DAIRY	FROZEN FOODS	4.0323%	0.24	0.54	2.218	<input checked="" type="checkbox"/>	
COLD CUTS, PRODUCE, DAIRY	FROZEN FOODS	3.8612%	0.24	0.54	2.214	<input checked="" type="checkbox"/>	
MEAT, PRODUCE, DAIRY	FROZEN FOODS	8.2834%	0.24	0.52	2.143	<input checked="" type="checkbox"/>	
SODA, MEAT, PRODUCE	FROZEN FOODS	3.6614%	0.24	0.52	2.139	<input checked="" type="checkbox"/>	
COLD CUTS, FROZEN FOODS, PRODUCE	MEAT	3.4998%	0.37	0.78	2.093	<input checked="" type="checkbox"/>	
SODA, PRODUCE, DAIRY	FROZEN FOODS	4.1655%	0.24	0.51	2.092	<input checked="" type="checkbox"/>	
COLD CUTS, MEAT	FROZEN FOODS	4.1940%	0.24	0.50	2.070	<input checked="" type="checkbox"/>	
COLD CUTS, FROZEN FOODS, DAIRY	MEAT	3.6139%	0.37	0.77	2.069	<input checked="" type="checkbox"/>	
SODA, FROZEN FOODS, PRODUCE	MEAT	3.6614%	0.37	0.77	2.055	<input type="checkbox"/>	

Once the rules to deploy have been selected, the next step is to score the **Transactions to Score** dataset.

To do this add a **Scoring** node from the **Action** palette and connect both the **Model Instance** and the **Transactions to Score** dataset as illustrated in figure.

Figure 18.16: Scoring the Transactions to Score dataset



Options are as usual and dialogs are available to map fields and name the results. Skipping to the **Scoring – Scoring Fields** dialog provide insight into the fields to be added to the scored dataset.

Figure 18.17: Scoring – Scoring Fields



Up to five fields including the mandatory **Recommendation** are available for selection. Additional fields ranking the recommendations on either **Lift** or **Confidence** values can also be created.

A tickbox option to recommend new items only is provided in this dialog also. This will limit recommendations to new products only, i.e. do not recommend items already appearing in the target basket.

By default **KnowledgeSTUDIO** outputs only results for those to whom a recommendation is made. If all records should be included in the final file, the option **Generate a null record for IDs that have no recommendation** should be checked.

Click **Run** to generate the new scored dataset, opening the dataset shows it contains all the usual tabs with the addition of the **Report** tab.

Figure 18.18: Scored Data Report Tab

Market Basket Analysis Scoring Report	
Input Model Name: Retail Transactions_MBA_02Jul1	
Input Dataset: Transactions To Score	
Scoring Settings	
Transaction ID(s) in input dataset	Transaction ID
Item identifier field in input dataset	Item Category
Recommend new items only	Yes
Generate null record for IDs with no recommendation	No
Scoring Statistics	
Total number of input records	6,000
Number of ID(s) with no recommendation	3,930
Recommendation Frequencies	
FROZEN FOODS	103
MEAT	8

The **Report** tab contains a summary of the scoring settings and basic scoring statistics, including recommendation frequencies. The report shows up to 100 most frequent recommendations ordered by frequency.

As can be seen from the report tab of the 6,000 input records, 4,013 received no recommendation. 111 recommendations were made in total: 103 for *FROZEN FOODS* and 3 for *MEAT*. Increasing the rules to apply will increase recommendations made.

The **Data** tab provides access to the resulting recommendations and selected output statistics.

Figure 18.19: Scored Dataset

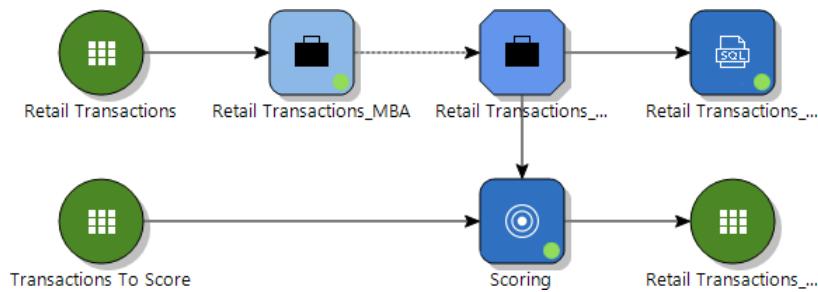
	Transaction ID	Recommendation	Lift	Rank_Lift	Confidence	
1	3723904	FROZEN FOODS	2.07035	1	0.50228	
2	3723949	FROZEN FOODS	2.092	1	0.50753	
3	3724049	FROZEN FOODS	2.2948	1	0.55673	
4	3724054	FROZEN FOODS	2.13873	1	0.51887	
5	3724282	FROZEN FOODS	2.14339	1	0.52	
6	3724688	FROZEN FOODS	2.29667	1	0.55718	
7	3724693	FROZEN FOODS	2.14339	1	0.52	
8	3724752	FROZEN FOODS	2.07035	1	0.50228	
9	3724754	FROZEN FOODS	2.07035	1	0.50228	
10	3724819	FROZEN FOODS	2.2948	1	0.55673	

The model can also be deployed in code format for use on other platforms. Available formats for an **MBA** model are **English** and **SQL Select**.

English language rules are simply a natural language expression of association rules. They are generated for illustrative purposes rather than deployment. The *SQL* code is intended to produce recommendations and can be deployed in a database system.

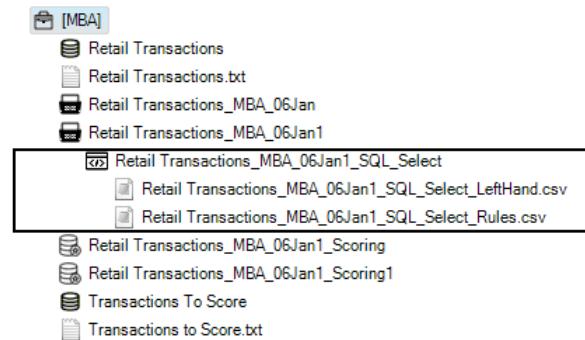
To illustrate **SQL** code for the model choose drag an **Generate SQL Select** node from the **Action** palette onto the **Workflow** canvas and connect to the **Model Instance** as illustrated in figure 18.20.

Figure 18.20: Generating SQL Select code



Once connected, run the code. The **SQL** code is generated along with two comma-separated files that represent the association rules selected for deployment. These are visible in the **Project Pane**

Figure 18.21: Code Generation Output



Together with the scoring code, the content of these files is necessary when deploying the rules in a database. The files are named:

- <Model_name>_SQL_Select_LeftHand.csv
- <Model_name>_SQL_Select_Rules.csv

As the **SQL** code uses the contents of these files when deploying an **MBA** a strict process must be followed when deploying to a database. The steps and an explanation of the code are outlined.

Figure 18.22: Code Generation Output

```
create table [__left_hand_side__] (
    [left_hand_side_id] int not null,
    [left_hand_side_cardinality] int not null,
    [Item Category] nvarchar(19) null)

create table [__rule__] (
    [left_hand_side_id] int not null,
    [rule_id] nvarchar(1024) not null,
    [support] float not null,
    [expected] float not null,
    [confidence] float not null,
    [lift] float not null,
    [Item Category] nvarchar(19) null)

-- 
-- load the file 'C:\DEMO\Tutorial - Market Basket Analysis\Retail Transactions_MBA_14Jul_SQL_Select_LeftHand.csv' into the
[__left_hand_side__] table
--

-- 
-- load the file 'C:\DEMO\Tutorial - Market Basket Analysis\Retail Transactions_MBA_14Jul_SQL_Select_Rules.csv' into the [__rule__]
table
-- 

-- 
-- SQL rules for association
-- 

-- 
-- as an example, the [__left_hand_side__] table is converted
-- into a table of scorable transactions
-- 
select [left_hand_side].[left_hand_side_id] [Transaction ID], [left_hand_side].[Item Category]
    into [__scoring_example__]
    from [__left_hand_side__] [left_hand_side]
-- 
-- recommendations
```

The first part of the code contains the commands to load the content of the generated .csv files into database tables. The second part contains the scoring code that uses these tables to produce recommendations.

Detailed comments in the scoring code provide useful guidance and explanation of the process.

In the first part, the create table statements creates the schema for two tables that together represent the association rules to be deployed. Populate these tables by loading the generated text file content.

To do this: right-click the project name in the **Project Outline** and select **Open In Explorer** from the context menu. This will open the project folder in **Windows Explorer**.

Copy the files:

- <Model_name>_SQL_Select_LeftHand.csv
- <Model_name>_SQL_Select_Rules.csv

from the project folder to a preferred location and load their content into the database tables.

The second part of the code contains an example scoring code that converts the first table to a set of input transactions and produces recommendations for them.

It serves as an example code snippet that the user can modify as necessary to target other transaction tables.

For each input ID, the *SQL* code only produces the top-ranked recommendation based on **Lift**.

NOTE: that scoring options: **Recommend new items only** and **Generate a null record for IDs that have no recommendation** are not used in *SQL* code generation, however, these can be added manually.

18.5 Summary

This chapter illustrated and demonstrated the **KnowledgeSTUDIO** facility for **Market Basket Analysis**. **MBA** can be applied to data to derive product recommendations regardless of the sector; retail, financial services, marketing, insurance etc.

As a result of completing this chapter, users should be able to:

- Describe **Market Basket Analysis**
- Develop a market basket analysis model using **KnowledgeSTUDIO**
- Deploy **Market Basket Analysis** results

Exercises

The data used for these exercises are accessed by loading the Tutorial – **Market Basket Analysis** from the **Prepare Sample Data** dialog found in the **Help** menu.

This project contains two transactional datasets:

- **Retail Transactions**
- **Transactions to Score**

The **Retail Transactions** dataset contains approx. 50k records relating to customer transactions in a supermarket and the basis for the model. Rules are applied to a second dataset: **Transactions To Score**.

The aim is to develop a model based on the *Item Category* field with *Customer ID* as the unique identifier.

#	Field Name	Field Label	Data Type	Cardinality	Unique Count	# of Missing Values	% of Missing Values
1	Customer ID	Customer ID	Number	3929	111	0	0.00 %
2	Transaction ID	Transaction ID	Number	10667	2356	0	0.00 %
3	Item ID	Item ID	Number	3534	875	0	0.00 %
4	Day of the week	Day of the week	String	7	0	0	0.00 %
5	Sale date	Sale date	String	7	0	0	0.00 %
6	Quantity	Quantity	Number	9	0	0	0.00 %
7	Store Department	Store Department	String	10	0	0	0.00 %
8	Item Name	Item Name	String	3492	845	0	0.00 %
9	Item Category	Item Category	String	25	0	0	0.00 %

1. Explore the data using the profiling features available in **KnowledgeSTUDIO**.
2. Insert a **Market Basket Analysis** node from the **Model** palette. Select *Customer ID* as the **Transaction ID(s)** and *Item Category* as the **Item Identifier**. Leave all other parameters at their default. NB: Minimum Support %: should be set at 2%.
3. Assess the output results. How many rules and itemsets have been discovered?
4. Use the **Map** and **Charts** tab to understand the relationships and most prominent rules.
5. Assess the resulting itemsets and rules from the **Item Sets** and **Rules** tabs
6. Use the filtering options to identify the best rules with adequate itemsets i.e. in excess of 1.
7. As there are over 1,000 rules generated, re-run the model and increase the minimum support to 10% and assess the results.
8. What are the highest **Lift** and **Confidence** values? How would you interpret the other available statistics in the **Rules** tab?
9. Select the top 5 rules and score the **Transactions To Score** dataset using the **Scoring** node from the **Action** palette.

10. How many recommendations have been made? Have any customers multiple recommendations? Were any ranking variables included in the scoring output?
11. Increase the number of recommendations made by selecting more rules. Base rule inclusion on the **Lift** value. What is an appropriate cutoff?
12. If applicable, create *SQL* code for the model using the **SQL Select** node from the **Action** palette. Refer to the manual and the **Help** file for further assistance in understanding how to deploy the code.

Chapter 19: Course Summary

19.1 Introduction

Thank you for your attendance during this course: **Advanced Modelling with Altair KnowledgeSTUDIO!**

You should now have a solid understand of *Data Mining* techniques prevalent in the field and how to apply these using **Altair KnowledgeSTUDIO**.

Although this course aims at covering the breadth and scope of *Data Mining* and its applicability to business problems, addressing every possible scenario would increase document size inordinately. Most chapters contain hints and tips, some additions are provided and even though these are not exhaustive, they may provide further insight.

19.2 Hints and Tips

The following hints and tips may provides some useful insight when working with data using **Altair KnowledgeSTUDIO**.

- Select all **Workflow** nodes of interest, copy and paste to a new **Workflow**
- Projects exist in folders in their entirety, they can be moved, zipped up and emailed
- Align, auto arrange and undo options are also available when creating **Workflows**
- Remember variable expressions are created in *SQL* format, these can be shared between datasets in a project or exported to a text file for use on other platforms
- Use **Decision Trees** to explore data, not only to create models!
- Update processes by pointing the **Workflow Import** node to new data
- Create **LOS** code for **Workflows** by selecting one or more **Workflow** nodes, right clicking and choosing LOS Code
- A representative sample may be as few as 3000 cases and representation, rather than size, is the key
- Many **Decision Trees** can generate the same accuracy, create more than one and combine results
- Model deployment: make sure fields used in the model are named as such in the deployment database or dataset
- Use multiple models to determine the same outcome and combine results
- Use a **Decision Tree** to compliment a **Logistic Regression**
- Use a **Decision Tree** to understand the results of a **Neural Network**
- Validation can also be used to confirm **Cluster Analysis** results!
- Models can be imported via *XML* and *PMML*
- Use the **Model Selector** node to automatically select the best performing model
- Use the **Variable Importance** node as a means to identify predictors prior to modelling

19.3 Course Objectives

As a result of attending and completing this course, attendees should be able to:

- Explain the concept of *Data Mining*
- Navigate the **KnowledgeSTUDIO** interface
- Create and manage projects using **KnowledgeSTUDIO**
- Import data from a variety of sources and file formats
- Analyse and profile data using **KnowledgeSTUDIO** capabilities
- Prepare and transform data including deriving new variables
- Explain, build, evaluate, validate and deploy:
 - **Altair Decision Trees**
 - **Altair Strategy Trees**
 - **Linear Regression models**
 - **Linear Regression models**
 - **Neural Network models**
 - **Cluster Analysis models**
 - **Market Basket Analysis models**
 - **Principal Component Analysis Models**

Thank you for your attendance and we wish you a pleasant experience! Further information on all **Altair** products and services can be found at: <http://www.altair.com>. Alternatively, email us at: info@altair.com.