

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

**Course Notes**

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

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

**Applying Data Mining Techniques Using SAS® Enterprise Miner™ Course Notes**

Copyright © 2005 by SAS Institute Inc., Cary, NC 27513, USA. All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

---

Book code 59758, course code ADMT5, prepared date 03Feb05.

# Table of Contents

Course Description .....	vi
Prerequisites .....	vii
General Conventions .....	viii
<b>Chapter 1 Introduction to Data Mining .....</b>	<b>1-1</b>
1.1 Background .....	1-3
1.2 SEMMA .....	1-14
<b>Chapter 2 Predictive Modeling Using Decision Trees .....</b>	<b>2-1</b>
2.1 Introduction to SAS Enterprise Miner .....	2-3
2.2 Modeling Issues and Data Difficulties .....	2-19
2.3 Introduction to Decision Trees .....	2-36
2.4 Building and Interpreting Decision Trees .....	2-45
2.5 Exercises .....	2-75
2.6 Solutions to Exercises .....	2-79
<b>Chapter 3 Predictive Modeling Using Regression .....</b>	<b>3-1</b>
3.1 Introduction to Regression .....	3-3
3.2 Regression in SAS Enterprise Miner .....	3-8
3.3 Exercises .....	3-37
3.4 Solutions to Exercises .....	3-39
<b>Chapter 4 Variable Selection .....</b>	<b>4-1</b>
4.1 Variable Selection and SAS Enterprise Miner .....	4-3

<b>Chapter 5</b>	<b>Predictive Modeling Using Neural Networks .....</b>	<b>5-1</b>
5.1	Introduction to Neural Networks .....	5-3
5.2	Visualizing Neural Networks .....	5-9
5.3	Exercises .....	5-22
5.4	Solutions to Exercises .....	5-23
<b>Chapter 6</b>	<b>Model Evaluation and Implementation .....</b>	<b>6-1</b>
6.1	Model Evaluation: Comparing Candidate Models.....	6-3
6.2	Ensemble Models.....	6-9
6.3	Model Implementation: Generating and Using Score Code .....	6-14
6.4	Exercises .....	6-25
6.5	Solutions to Exercises .....	6-26
<b>Chapter 7</b>	<b>Cluster Analysis .....</b>	<b>7-1</b>
7.1	Using $k$ -Means Cluster Analysis .....	7-3
7.2	Exercises .....	7-20
7.3	Solutions to Exercises .....	7-22
<b>Chapter 8</b>	<b>Association and Sequence Analysis .....</b>	<b>8-1</b>
8.1	Introduction to Association Analysis .....	8-3
8.2	Interpretation of Association and Sequence Analysis .....	8-7
8.3	Dissociation Analysis (Self-Study) .....	8-28
8.4	Exercises .....	8-33
8.5	Solutions to Exercises .....	8-35
<b>Appendix A</b>	<b>References .....</b>	<b>A-1</b>
A.1	References.....	A-3

<b>Appendix B Case Study .....</b>	<b>B-1</b>
B.1 Case Study Exercise.....	B-3
B.2 Solutions to Case Study Exercise .....	B-6
<b>Appendix C Index .....</b>	<b>C-1</b>

## Course Description

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

After completing this course, you should be able to

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

### To learn more...



SAS Education

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

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



SAS Publishing

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

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

## Prerequisites

Before selecting this course, you should be familiar with Microsoft Windows and Windows-based software. Previous SAS software experience is helpful but not necessary.

## General Conventions

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

## Typographical Conventions

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

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

## Syntax Conventions

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

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

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

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



# Chapter 1 Introduction to Data Mining

1.1	Background.....	1-3
1.2	SEMMA .....	1-14



## 1.1 Background

### Objectives

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

3

### Defining Characteristics

#### 1. The Data

- Massive, operational, and opportunistic

#### 2. The Users and Sponsors

- Business decision support

#### 3. The Methodology

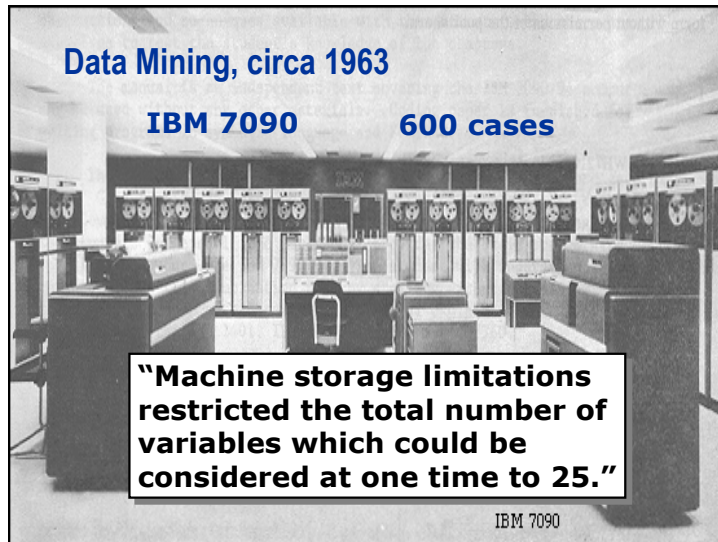
- Computer-intensive “ad hockery”
- Multidisciplinary lineage

4

SAS defines data mining as

*the process of sampling, exploring, **modifying**, **modeling**, and **assessing** (SEMMA) large amounts of data to uncover previously unknown patterns, which can be utilized as a business advantage.*

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



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

### Since 1963

#### Moore's Law:

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

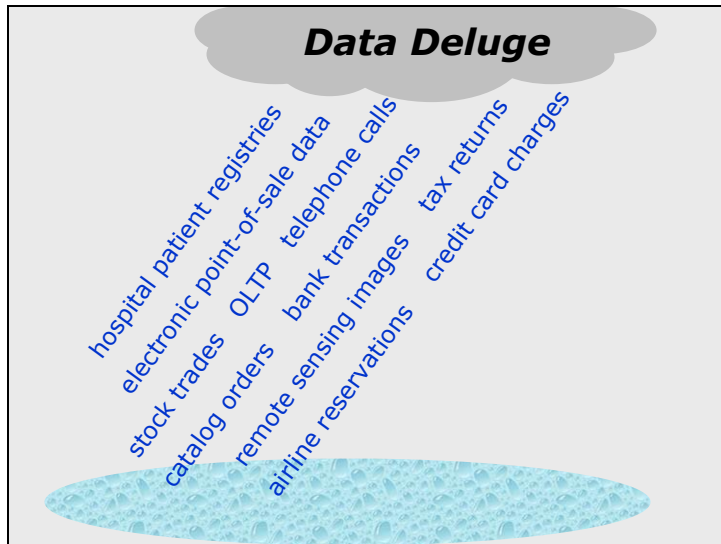
#### Parkinson's Law:

Work expands to fill the time available for its completion.

6

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

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



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

## The Data

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

8

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



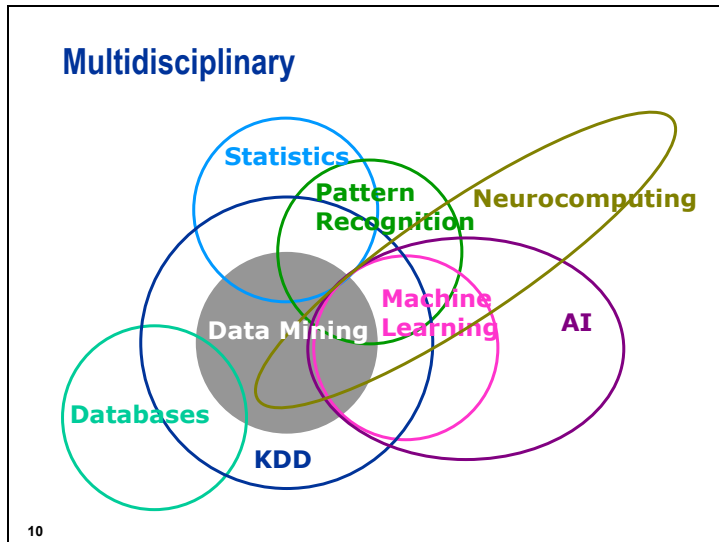
The owners of the data and sponsors of the analyses are typically not researchers. The objectives are usually to support crucial business decisions.

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

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

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

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



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

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

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

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

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

## Tower of Babel

### “Bias”

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

NEUROCOMPUTING: the constant term in a linear combination

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



11

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



### Steps in Data Mining/Analysis

#### 1. Specific Objectives

- In terms of the subject matter

#### 2. Translation into Analytical Methods

#### 3. Data Examination

- Data capacity
- Preliminary results

#### 4. Refinement and Reformulation

12

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

- *Understand our customer base.*
- *Re-engineer our customer retention strategy.*
- *Detect actionable patterns.*

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

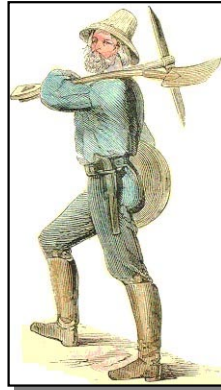
A related pitfall is to specify the objectives in terms of analytical methods:

- *Implement neural networks.*
- *Apply visualization tools.*
- *Cluster the database.*

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

## Required Expertise

- Domain
- Data
- Analytical Methods

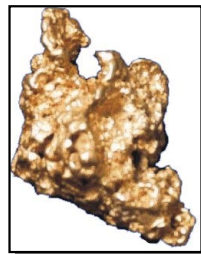


13

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

The embodiment of this expertise might take up to three or more people.

## Nuggets



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

— Herb Edelstein

14

The passage continues (Beck 1997):

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

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

## What Is Data Mining?

- **IT**
  - Complicated database queries
- **ML**
  - Inductive learning from examples
- **Stat**
  - What we were taught not to do

15

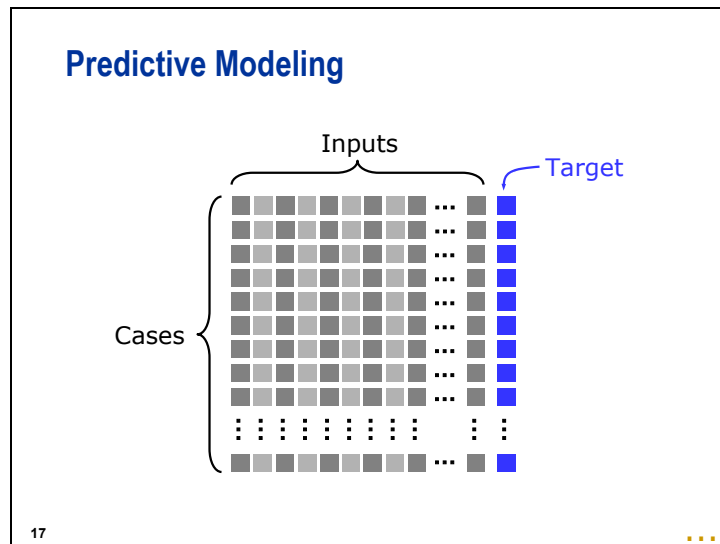
The database community has a tendency to view data mining methods as more complicated types of database queries. For example, standard query tools can answer questions such as “how many surgeries resulted in hospital stays longer than 10 days?” But data mining is needed for more complicated queries such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

## Problem Translation

- **Predictive Modeling**
  - Supervised classification
- **Cluster Analysis**
- **Association Rules**
- **Something Else**

16

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



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

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

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

### Types of Targets

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

18

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

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

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

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

## 1.2 SEMMA

### Objectives

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

## SEMMA

- Sample
- Explore
- Modify
- Model
- Assess

21

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

SEMMA stands for

Sample - the data by creating one or more data tables. The samples should be large enough to contain the significant information, yet small enough to process.

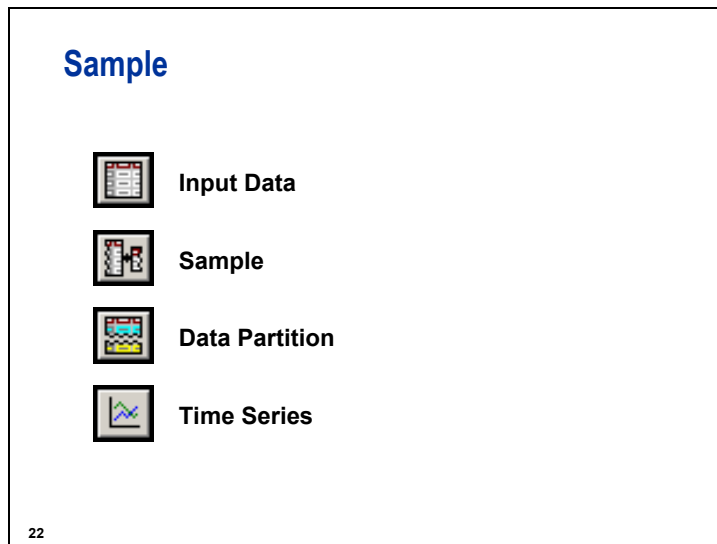
Explore - the data by searching for anticipated relationships, unanticipated trends, and anomalies in order to gain understanding and ideas.

Modify - the data by creating, selecting, and transforming the variables to focus the model selection process.

Model - the data by using the analytical tools to search for a combination of the data that reliably predicts a desired outcome.

Assess - compare competing predictive models (build charts to evaluate the usefulness and reliability of the findings from the data mining process).

Additional tools are available under the Utilities group.



### Sample Nodes

The **Input Data node** represents the data source that you choose for your mining analysis and provides details (metadata) about the variables in the data source that you want to use.

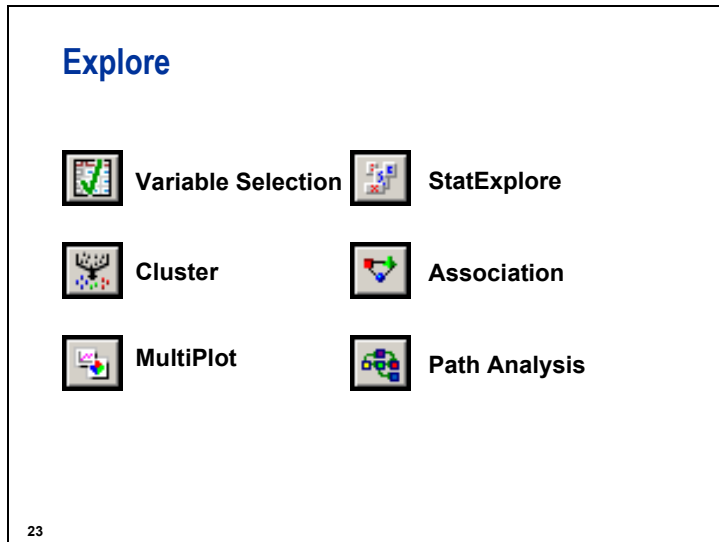
The **Sample node** enables you to take simple random samples,  $n^{\text{th}}$  observation samples, stratified random samples, first- $n$  samples, and cluster samples of data sets. For any type of sampling, you can specify either a number of observations or a percentage of the population to select for the sample. If you are working with rare events, the Sample node can be configured for oversampling, or stratified sampling.

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

The **Data Partition node** enables you to partition data sets into training, test, and validation data sets. The training data set is used for preliminary model fitting. The validation data set is used to monitor and tune the model during estimation and is also used for model assessment. The test data set is an additional holdout data set that you can use for model assessment. This node uses simple random sampling, stratified random sampling, or cluster sampling to create partitioned data sets.

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





## Explore Nodes

The **Variable Selection node** enables you to evaluate the importance of input variables in predicting or classifying the target variable. To select the important inputs, the node uses either an R-square or a Chi-square selection criterion. The R-square criterion enables you to remove variables in hierarchies, remove variables that have large percentages of missing values, and remove class variables that are based on the number of unique values. The variables that are not related to the target are set to a status of *rejected*. Although rejected variables are passed to subsequent nodes in the process flow diagram, these variables are not used as model inputs by more detailed modeling nodes, such as the Neural Network and Decision Tree nodes. You can reassign the input model status to rejected variables.

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

The **MultiPlot node** is a visualization tool that enables you to explore large volumes of data graphically. The MultiPlot node automatically creates bar charts and scatter plots for the input and target. The code created by this node can be used to create graphs in a batch environment.

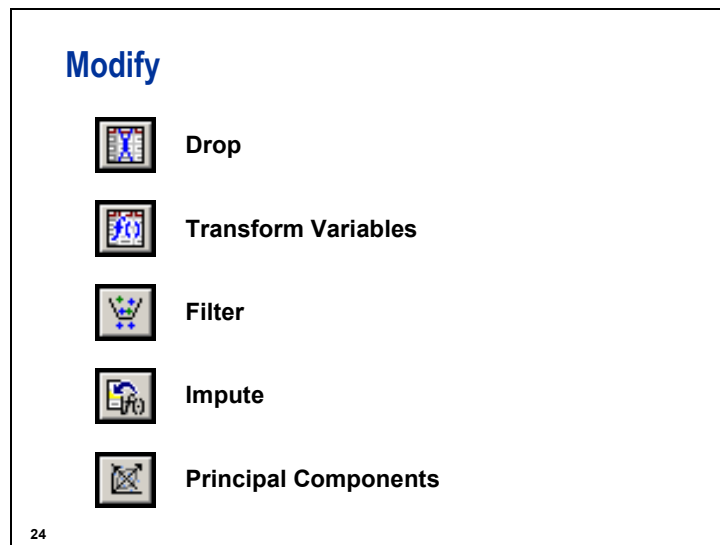
The **StatExplore node** is a multipurpose tool used to examine variable distributions and statistics in your data sets. The node generates summarization statistics. You can use the StatExplore node to:

- Select variables for analysis, for profiling clusters, and for predictive models
- Compute standard univariate distribution statistics
- Compute standard bivariate statistics by class target and class segment
- Compute correlation statistics for interval variables by interval input and target.

The **Association node** enables you to perform association discovery to identify items that tend to occur together within the data. For example, if a customer buys a loaf of

bread, how likely is the customer to also buy a gallon of milk? This type of discovery is also known as market basket analysis. The node also enables you to perform sequence discovery if a time stamp variable (a sequence variable) is present in the data set. This enables you to take into account the ordering of the relationships among items.

The **Path Analysis node** enables you to analyze Web log data to determine the paths that visitors take as they navigate through a Web site. You can also use the node to perform sequence analysis.



### Modify Nodes

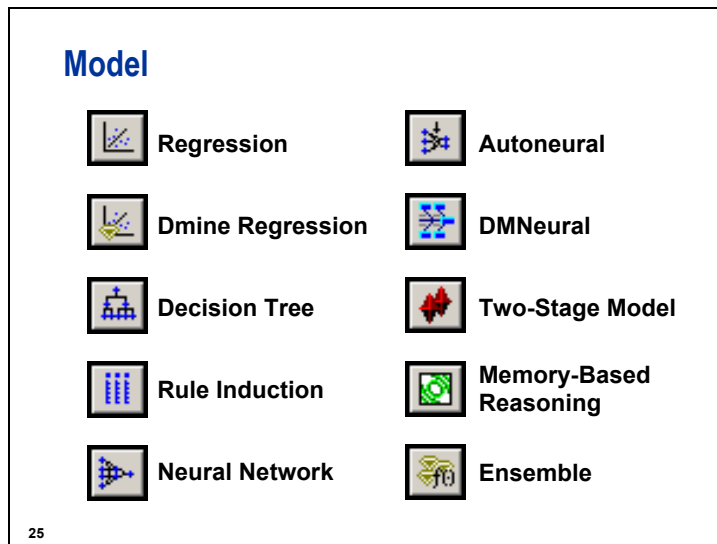
The **Drop node** is used to remove variables from scored data sets. You can remove all variables with the role type that you specify or you can manually specify individual variables to drop. For example, you could remove all hidden, rejected, and residual variables from your exported data set, or just a few variables that you identify yourself.

The **Transform Variables node** enables you to create new variables that are transformations of existing variables in your data. Transformations are useful when you want to improve the fit of a model to the data. For example, transformations can be used to stabilize variances, remove nonlinearity, improve additivity, and correct nonnormality in variables. The Transform Variables node supports various transformation methods. The available methods depend on the type and the role of a variable.

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

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

The **Principal Components node** calculates eigenvalues and eigenvectors from the uncorrected covariance matrix, corrected covariance matrix, or the correlation matrix of input variables. Principal components are calculated from the eigenvectors and are usually treated as the new set of input variables for successor modeling nodes. A principal components analysis is useful for data interpretation and data dimension reduction.



### Model Nodes

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

The **Dmine Regression node** performs a regression analysis on data sets that have a binary or interval level target variable. The Dmine Regression node computes a forward stepwise least-squares regression. In each step, an independent variable is selected that contributes maximally to the model R-square value. The node can compute all 2-way interactions of classification variables and it also can use AOV16 variables to identify non-linear relationships between interval variables and the target variable. In addition, the node can use group variables to reduce the number of levels of classification variables.



If you want to create a regression model on data that contains a nominal or ordinal target, then you should use the Regression node.

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

The **Rule Induction node** enables you to improve the classification of rare events in your modeling data. The Rule Induction node creates a Rule Induction model which uses split techniques to remove the largest pure split node from the data. Rule Induction also creates binary models for each level of a target variable and ranks the levels from the most rare event to the most common.

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

The **AutoNeural node** can be used to automatically configure a neural network. It conducts limited searches for a better network configuration.

The **DMNeural node** is another modeling node that you can use to fit an additive nonlinear model. The additive nonlinear model uses bucketed principal components as inputs to predict a binary or an interval target variable.

The **TwoStage node** enables you to model a class target and an interval target. The interval target variable is usually the value that is associated with a level of the class target. For example, the binary variable **PURCHASE** is a class target that has two levels: Yes and No, and the interval variable **AMOUNT** can be the value target that represents the amount of money that a customer spends on the purchase.

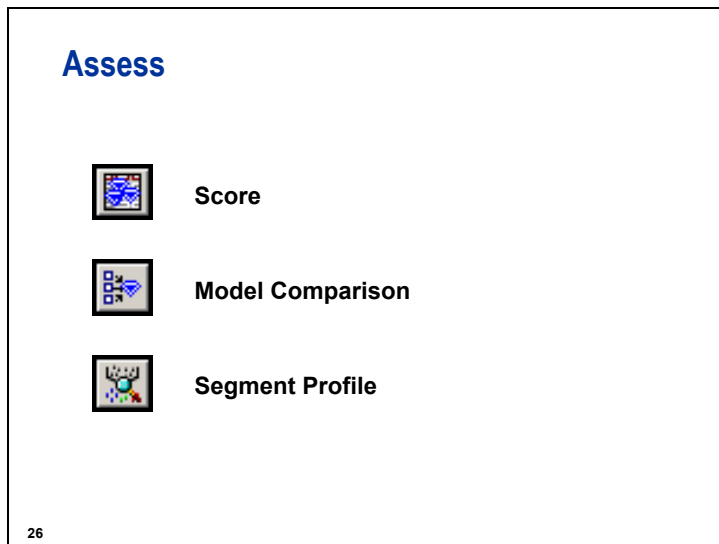
The TwoStage node supports two types of modeling: sequential and concurrent. For sequential modeling, a class model and a value model are fitted for the class target and the interval target, respectively, in the first and the second stages. For concurrent modeling, the values of the value target for observations that contain a non-event value for the class target are set to missing prior training. Then, the TwoStage node fits a neural network model for the class and value target variables simultaneously. The default TwoStage node fits a sequential model that has a decision tree class model and a neural network value model. Posterior probabilities from the decision tree class model are used as input for the regression value model.

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

The **Ensemble node** creates new models by combining the posterior probabilities (for class targets) or the predicted values (for interval targets) from multiple predecessor models. The new model is then used to score new data. One common ensemble approach is to use multiple modeling methods, such as a neural network and a decision tree, to obtain separate models from the same training data set. The component models from the two complementary modeling methods are integrated by the Ensemble node to form the final model solution. It is important to note that the ensemble model can only be more accurate than the individual models if the individual models disagree with one another. You should always compare the model performance of the ensemble model with the individual models. You can compare models in a Model Comparison node.



In SAS Enterprise Miner 5.1, the Ensemble node combines different model outputs. The SAS Enterprise Miner 5.1 Ensemble node does not yet support resampling, bagging, or boosting.



### Assess Nodes

The **Score node** enables you to manage, edit, export, and execute scoring code that is generated from a trained model. Scoring is the generation of predicted values for a data set that may not contain a target variable. The Score node generates and manages scoring formulas in the form of a single SAS DATA step, which can be used in most SAS environments even without the presence of SAS Enterprise Miner. The Score node can also generate C Score code and Java Score code.

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

The **Segment Profile node** enables you to examine segmented or clustered data and identify factors that differentiate data segments from the population. The node generates various reports that aid in exploring and comparing the distribution of these factors within the segments and population.

### Other Types of Nodes – Utility Nodes



**Metadata**



**SAS Code**



**Control Point**



**Merge**

27

### Utility Nodes

The **Metadata node** enables you to modify the columns metadata information at some point in your process flow diagram. You can modify attributes such as roles, measurement levels, and order.

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

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

The **Merge node** enables you to merge observations from two or more data sets into a single observation in a new data set. The Merge node supports both one-to-one and match merging.



# Chapter 2 Predictive Modeling Using Decision Trees

<b>2.1</b>	<b>Introduction to SAS Enterprise Miner .....</b>	<b>2-3</b>
<b>2.2</b>	<b>Modeling Issues and Data Difficulties .....</b>	<b>2-19</b>
<b>2.3</b>	<b>Introduction to Decision Trees .....</b>	<b>2-36</b>
<b>2.4</b>	<b>Building and Interpreting Decision Trees .....</b>	<b>2-45</b>
<b>2.5</b>	<b>Exercises .....</b>	<b>2-75</b>
<b>2.6</b>	<b>Solutions to Exercises .....</b>	<b>2-79</b>



## 2.1 Introduction to SAS Enterprise Miner

### Objectives

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



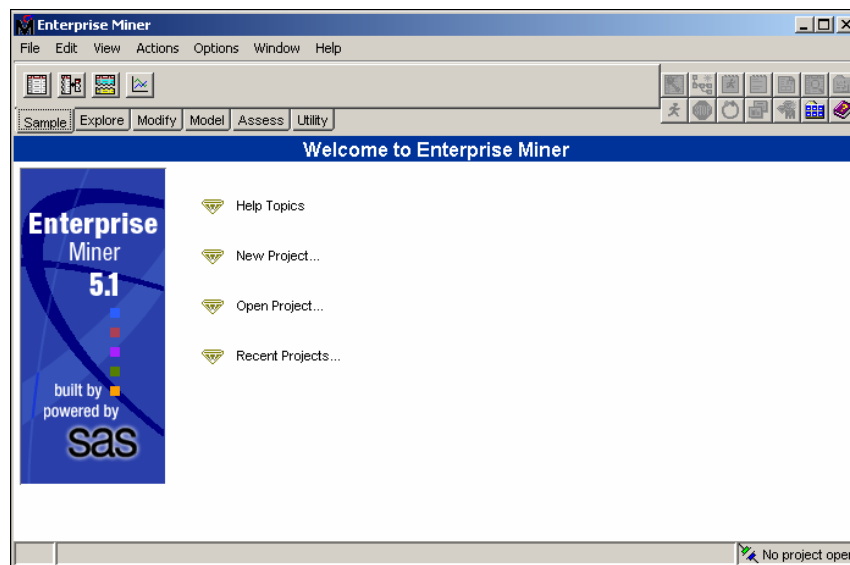
## Introduction to SAS Enterprise Miner

### Opening SAS Enterprise Miner

1. To begin, select **Start** ⇒ **Programs** ⇒ **SAS** ⇒ **Enterprise Miner**. This opens the Start Enterprise Miner window.
2. Ensure that Personal Workstation is selected, and then select **Start**.

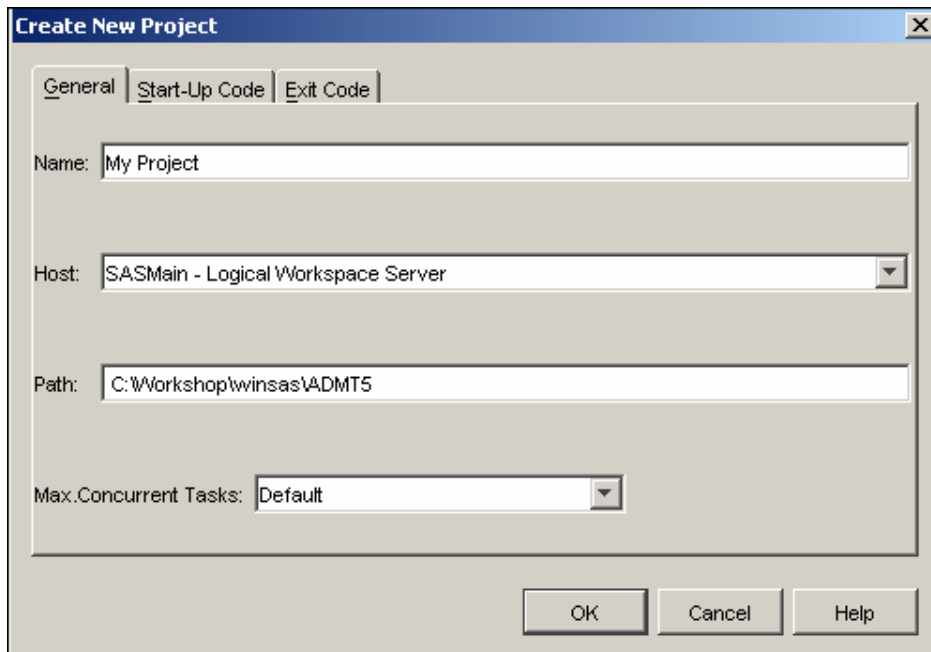
### Setting Up the Initial Project and Diagram

1. Select **New Project...** in the Welcome to Enterprise Miner screen.



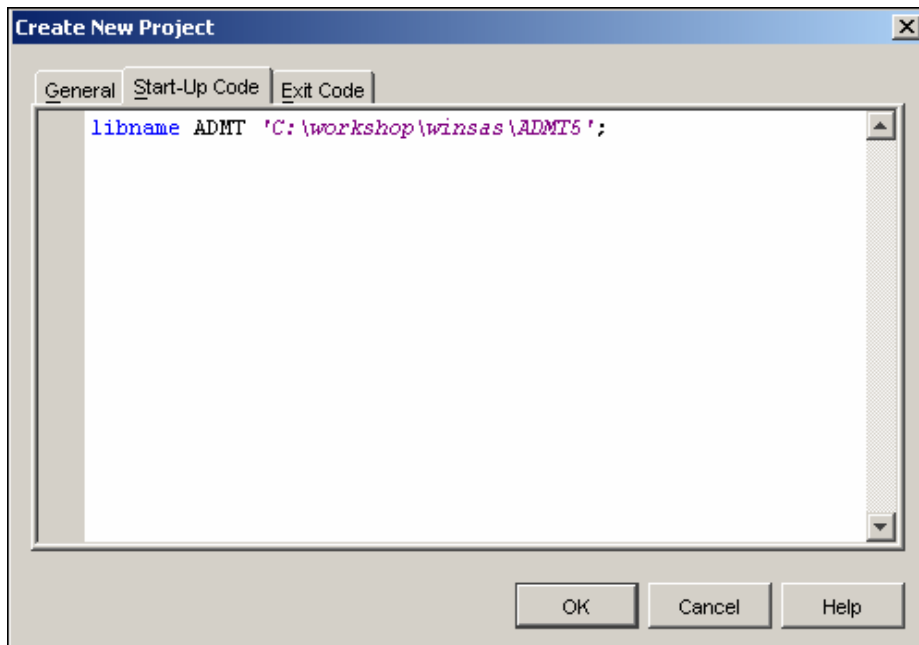
2. Type the name of the project (for example, **My Project**).

3. Type in the location of the project folder, for example,  
**C:\Workshop\winsas\AMDT5.**



The screenshot shows the 'Create New Project' dialog box with the 'General' tab selected. The 'Name' field contains 'My Project'. The 'Host' dropdown menu is set to 'SASMain - Logical Workspace Server'. The 'Path' field contains 'C:\Workshop\winsas\AMDT5'. The 'Max. Concurrent Tasks' dropdown menu is set to 'Default'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

4. Select the Start-Up Code tab.
5. Type in the appropriate code to define the library for the course data.



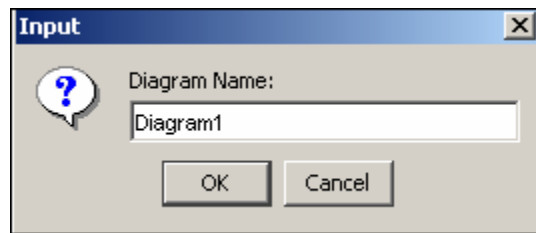
The screenshot shows the 'Create New Project' dialog box with the 'Start-Up Code' tab selected. The code editor contains the text: `libname ADMT 'C:\workshop\winsas\AMDT5';`. At the bottom are 'OK', 'Cancel', and 'Help' buttons.



This will create the library every time you open the project

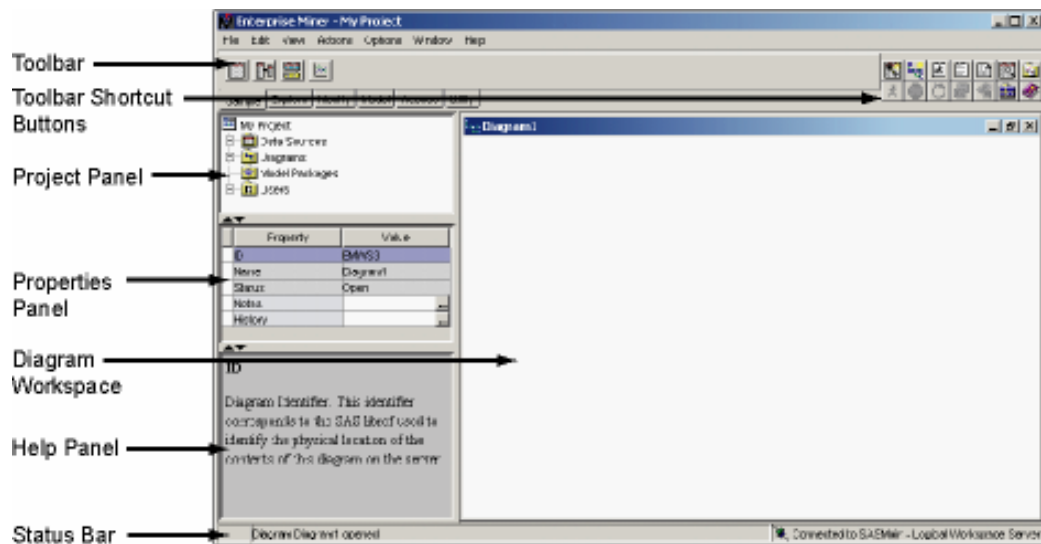
6. Select **OK**.
7. Select **File ⇒ New ⇒ Dialog...**

8. Type in the diagram name, for example **Diagram1**.



9. Select **OK**.

### Identifying the Components of the Enterprise Miner Window



In addition to the menu and toolbars, which will be explored throughout the course, the Enterprise Miner window has four major areas.

On the top left is the Project Panel. This is used to manage and view data sources, diagrams, results, and project users. The Data Sources folder displays the data sources that have been defined for the project. Selecting a data source in the directory displays the data source properties in the Properties Panel below. The Diagrams folder lists the various diagrams that are associated with the project. To open a diagram in the Diagram Workspace, right-click on a diagram name in the Project Panel and select **Open**. The Model Packages folder lists all the results you have generated by creating a report. The Users folder lists the users who are currently working with the open project.

The middle-left panel is the Properties Panel. This is used to view and edit the settings of data sources, diagrams, nodes, results, and users.

The bottom-left panel is the Help Panel. This displays a short description of the property that you select in the Properties Panel.

On the right side is the Diagram Workspace. This workspace is used to build, edit, run, and save process flow diagrams. The workspace is where you graphically build, order, and sequence the nodes that you use to mine your data and generate reports.

### The Scenario

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

5

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

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

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

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



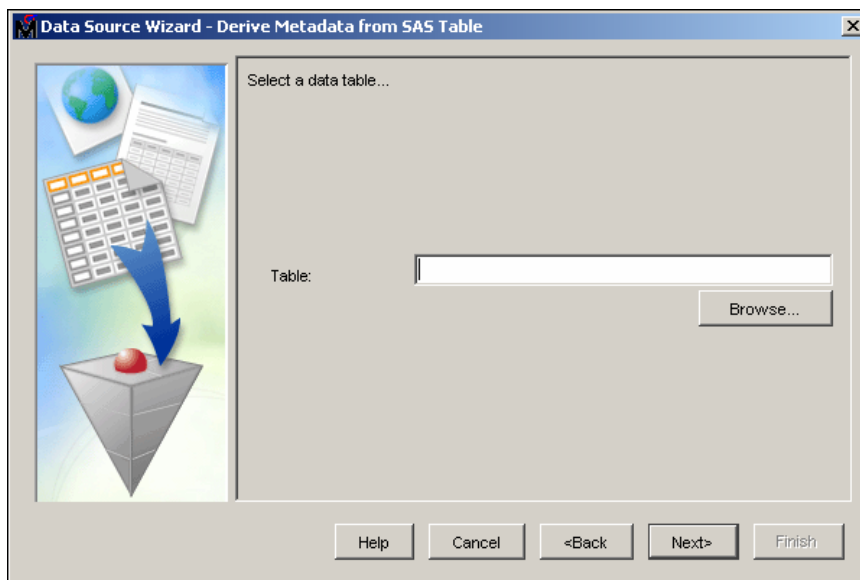


## Project Setup and Initial Data Exploration

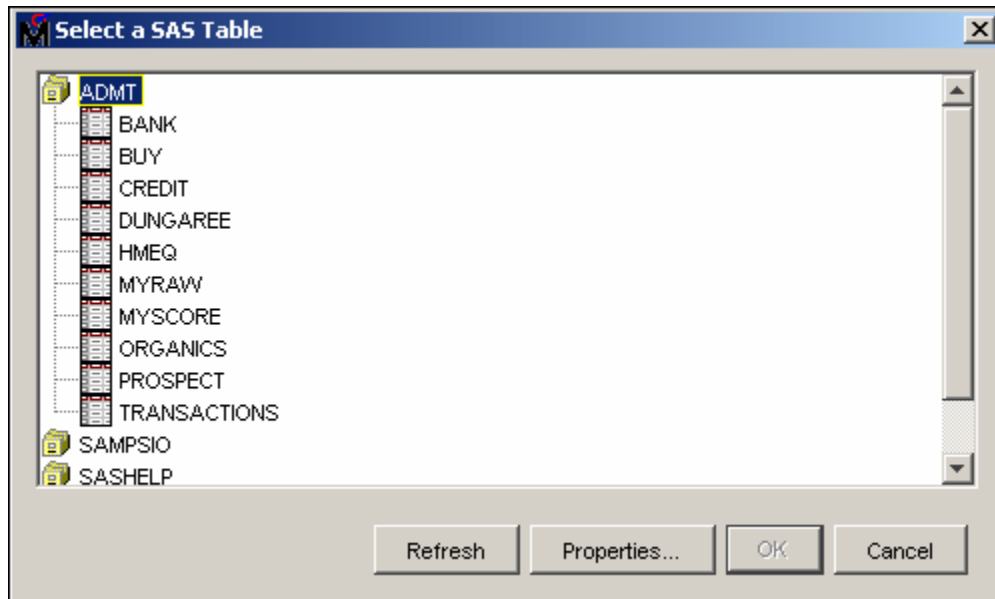
### Defining a Data Source

This example uses the **HMEQ** data set in the **ADMT** library.

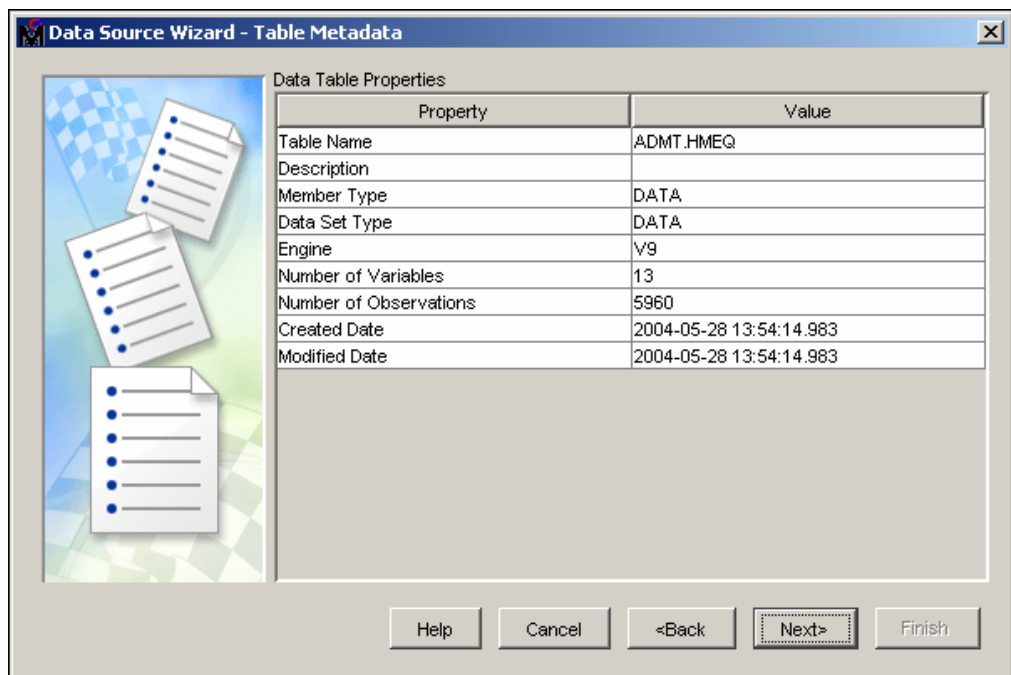
1. To define a data source, right-click on **Data Sources** in the Project Panel and select **Create Data Source**.
2. In the Data Source Wizard – Metadata Source window, be sure **SAS Table** is selected as the source and select **Next>**.



3. To choose the desired data table select **Browse...**.
4. Double-click on the **ADMT** library to see the data tables in the library.

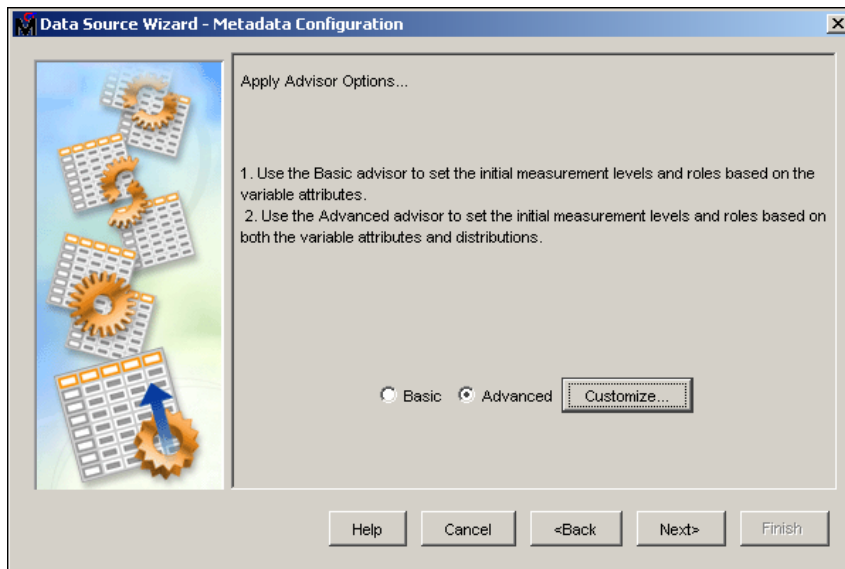


5. Select the **HMEQ** data set, and then select **OK**.
6. Select **Next>**.



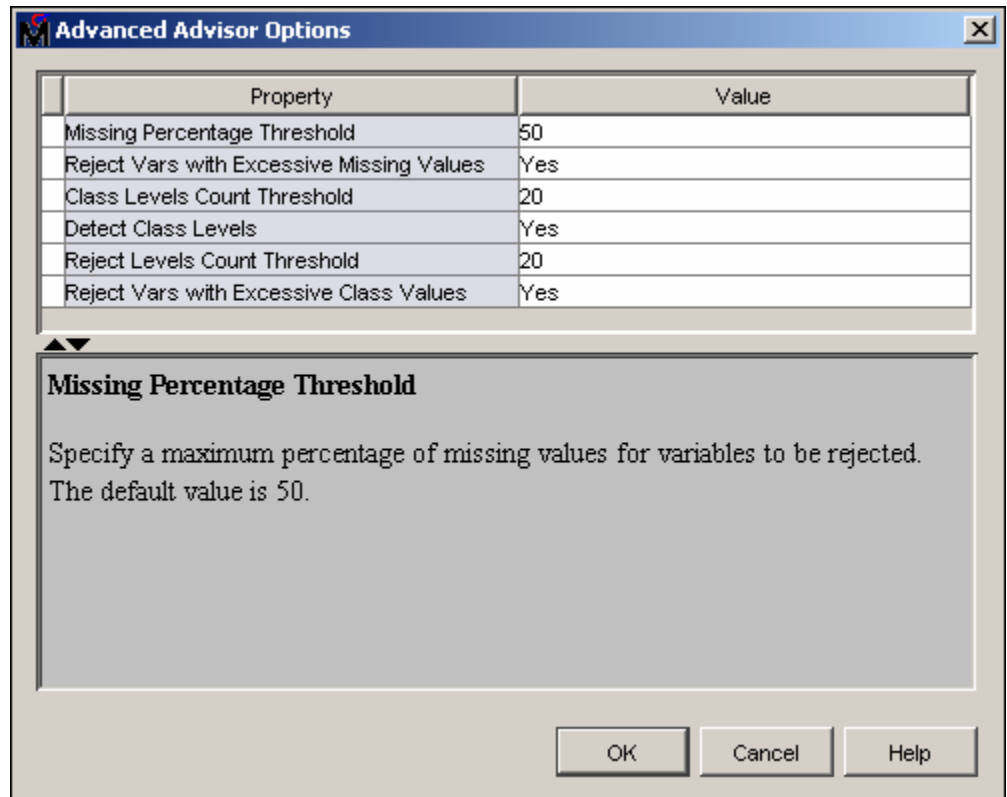
Observe that this data table has 5,960 observations (rows) and 13 variables (columns).

7. After examining the data table properties select **Next>**.



All analysis packages must determine how to use variables in the analysis. If you choose the basic option here, the initial role and level of the variables will be determined by the variable type and format values. If you choose the advanced option here, initial roles and levels are based on the variable type, format values, and the number of distinct values contained in the variable.

8. Select **Advanced** to use the Advanced advisor.
9. Select **Customize...** to view the details of the options available.

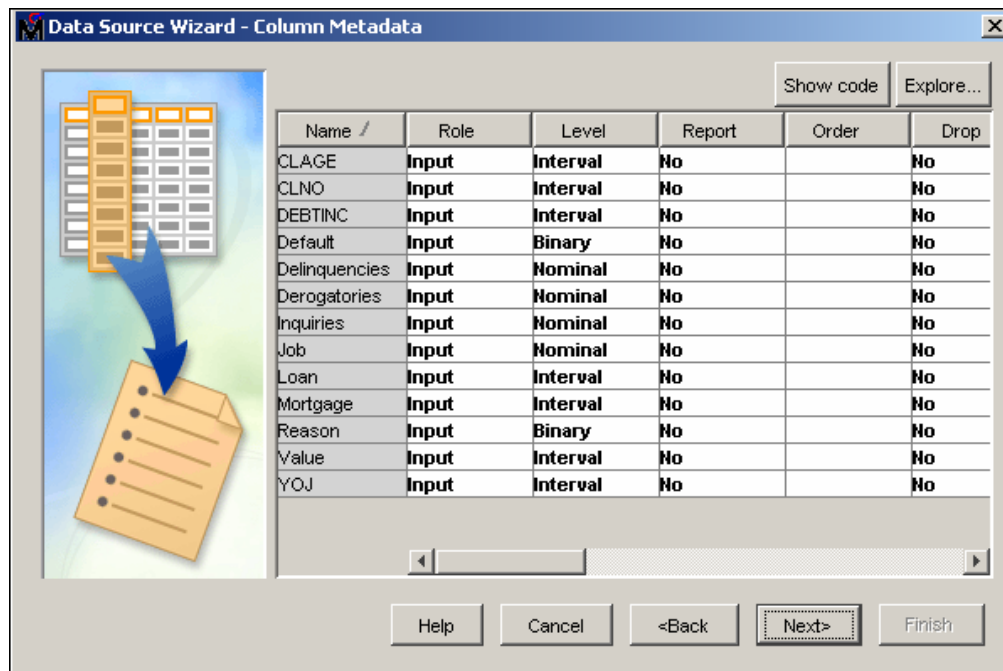


The first two choices address the number of missing values in a variable. The default settings are to reject variables that have more than 50% of their values missing. The next two choices address numeric variables. The default setting is that numeric variables with fewer than 20 unique values will be assigned as nominal variables. Finally, the last two choices address the maximum number of levels in a class variable. Any character variable with more than 20 levels will be rejected. You can override all of the decisions made by these rules when you examine the column metadata.

10. To leave the default values in place, select **Cancel**.

11. Select **Next>**.

12. Click on the first column heading, labeled Name, to sort the variables by their name. You can see all of the variables if you enlarge the window. The following table shows a portion of the information for each of the 13 variables.



Observe that some of the columns are grayed out. These columns represent information from the SAS data set that cannot be changed with the Data Source Wizard.

The first three variables (**CLAGE** through **DEBTINC**) have the measurement level interval because they are numeric variables in the SAS data set and have more than 20 distinct values in the data table. The model role for all interval variables is set to input by default.

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

The next three variables **Delinquencies**, **Derogatories**, and **Inquiries** are all numeric variables with fewer than 20 distinct values in the data table. Therefore, their level has been set to nominal by default. This frequently occurs with counting variables such as these. For the purpose of analysis, these variables should be treated as interval variables, so their level will need to be changed.

The variables **Job** and **Reason** are both character variables in the data set, but they have different levels. **Reason** is binary because it has only two distinct nonmissing levels. The level for **Job**, however, is nominal because it is a character variable with more than two levels.

The level for remaining variables is interval, which is appropriate for this data.

### Identifying Target Variables

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

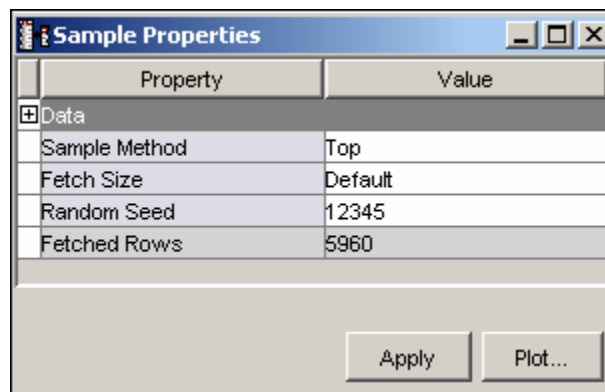
To modify the role information, proceed as follows:

1. Position the tip of your cursor over the row for **Default** in the Role column.
2. Click and select **Target** from the drop down menu.

### Inspecting Distributions

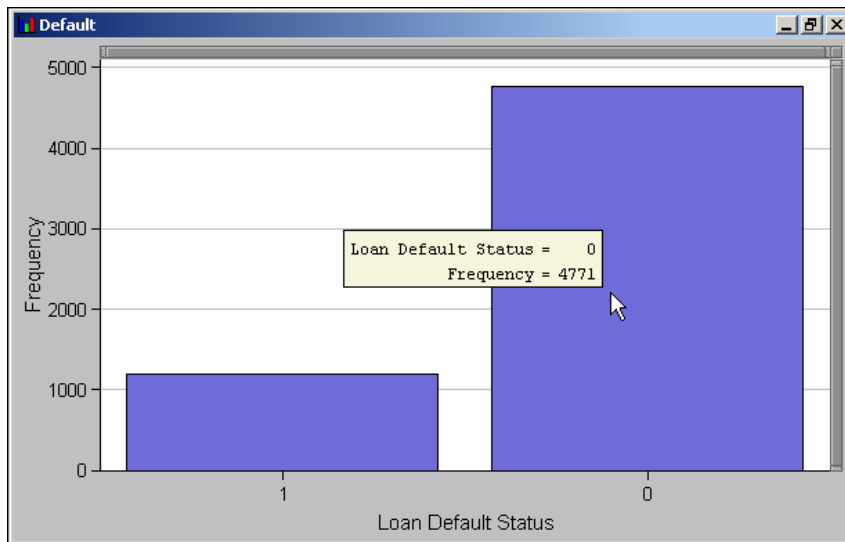
You can inspect the distribution of values for each of the variables. To view the distribution of **Default**:

1. Highlight the row for the variable **Default**.
2. Select **Explore...**.
3. Examine the Sample Properties window.



By default, the number of rows used in the exploration of variables depends on the record length and the number of rows in the data table. When the data table is large, fewer rows are selected for use. Presuming that all rows are not used for variable exploration, the top rows are selected by default. You can control this behavior, by changing the Sample Method. After a particular sample method has been selected, you can determine how large a sample you wish to use for exploration purposes. In this case, all of the rows are used, so no changes are necessary.

#### 4. Examine the bar chart for **Default**.



If you place your cursor over a bar in the chart the specific frequency of the value represented by that bar is displayed.

Close the Explore window when you are finished inspecting the plot. You can evaluate the distribution of other variables as desired.

#### Modifying Variable Information

Ensure that the remaining variables have the correct role and level information. Change the level for **Delinquencies**, **Derogatories**, and **Inquiries** to interval.

1. Position the tip of your cursor over the row for **Delinquencies** in the level column.
2. Click and select **Interval** from the drop down menu.
3. Repeat steps 1 and 2 for **Derogatories** and **Inquiries**.

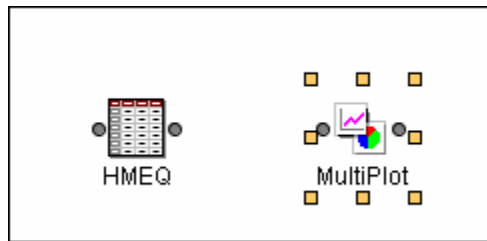


Alternatively, you can update the level information for all three variables at the same time by highlighting all three rows simultaneously before following steps 1 and 2 above.

4. When you have finished exploring the distribution of the variables, return to the Data Source Wizard – Column Metadata window and select **Next>**.
5. For now, skip Decision Processing by selecting **Next>**.
6. If desired, add notes about the data set, and then select **Finish**. The **HMEQ** data set has been added as a data source for this project.

### Building the Initial Flow


1. Presuming that Diagram1 in the project named My Project is open, add the Data Source node for the **HMEQ** data source from the Project Panel to the diagram workspace.
2. Select the Explore tab in the toolbar and drag a MultiPlot node to the workspace to the right of the **HMEQ** Data Source node. Your diagram should appear as shown below.



Observe that the MultiPlot node is selected (as indicated by the yellow squares around it), but the **HMEQ** node is not selected. If you click in any open space on the workspace, all nodes become deselected.

In addition to dragging a node onto the workspace, there is another way to add a node to the flow. You can right-click in the workspace where you want the node to be placed and select **Add node** from the pop-up menu. A pop-up menu appears, enabling you to select the desired node.

The shape of the cursor changes depending on where it is positioned. The behavior of the mouse commands depends on the shape as well as the selection state of the node over which the cursor is positioned.

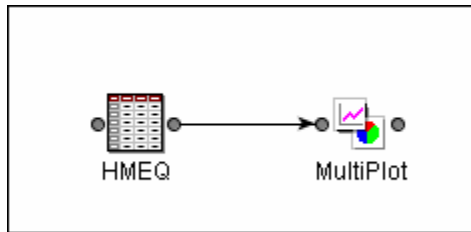
Observe that when you put your cursor in the middle of a node, the cursor appears as a four-headed arrow, . To move the nodes around the workspace:

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



To connect the two nodes in the workspace:

1. Position the cursor on the circle at the edge of the icon representing the **HMEQ** Data Source (until the pencil appears).
2. Press the left mouse button and drag in the direction of the MultiPlot node.
3. Release the mouse button after reaching the circle at the edge of the icon that represents the MultiPlot node.

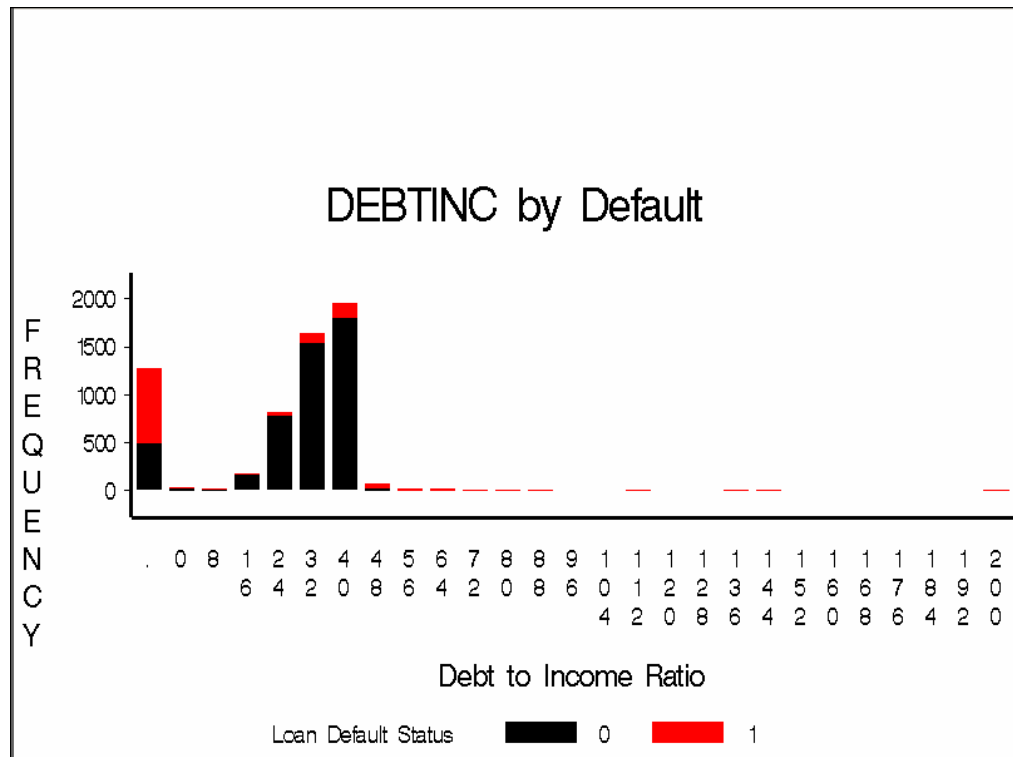


### Additional Data Exploration

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

1. Right-click on the MultiPlot node and select **Run**.
2. When prompted, select **Yes** to run the path.
3. Select **OK** to acknowledge the completion of the run.
4. Right-click on the MultiPlot node and select **Results...** to view the results.

By using the **Next>** button on the display, you can view the histograms generated for this data. You can also move to a specific graph using the drop down menu.

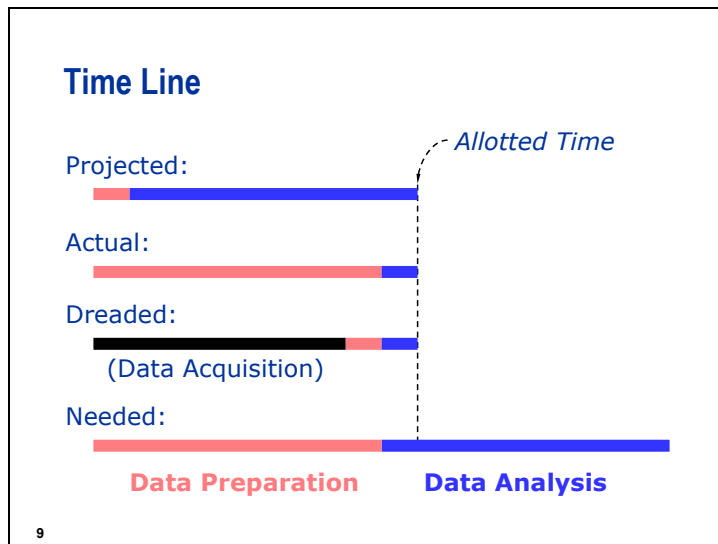


From this histogram, you can see that many of the defaulted loans were by homeowners with either a high debt-to-income ratio or an unknown debt-to-income ratio.

## 2.2 Modeling Issues and Data Difficulties

### Objectives

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



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

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

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

### Data Arrangement

<u>Acct</u> <u>type</u>		<i>Long-Narrow</i>						
2133	MTG							
2133	SVG							
2133	CK							
2653	CK							
2653	SVG							
3544	MTG							
3544	CK							
3544	MMF							
3544	CD							
3544	LOC							

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

11

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

### Derived Inputs

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

12

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

### Roll Up

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

13

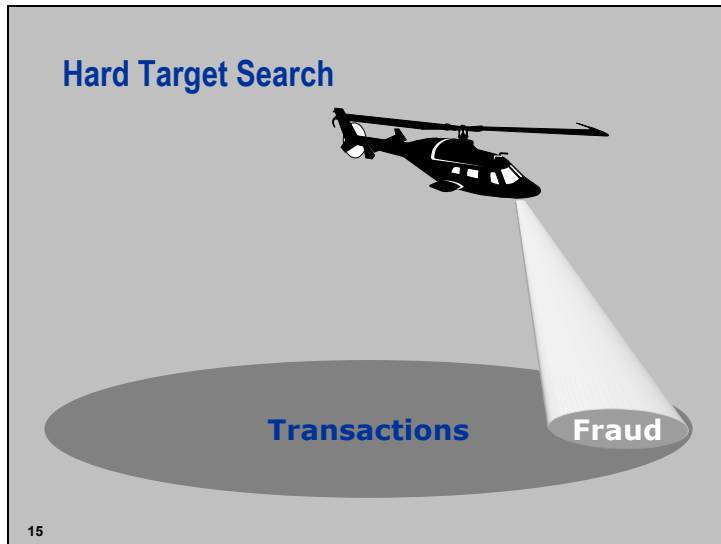
Marketing strategies often dictate rolling up accounts from a single household into a single record (case). This process usually involves creating new summary data. How should the sales figures for multiple accounts in a household be summarized? Using the sum, the mean, the variance, or all three?

### Rolling Up Longitudinal Data

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

14

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



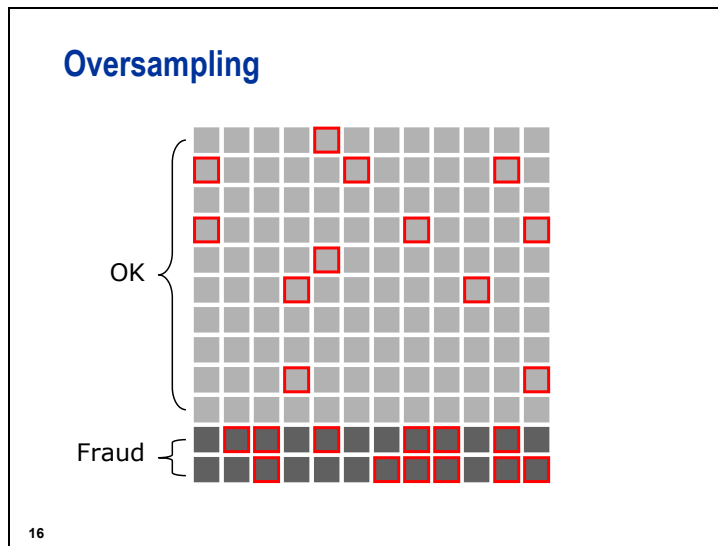
The lack of a target variable is a common example of opportunistic data not having the capacity to meet the objectives. For instance, a utility company may have terabytes of customer usage data and a desire to detect fraud, but it does not know which cases are fraudulent. The data is abundant, but none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

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

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

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

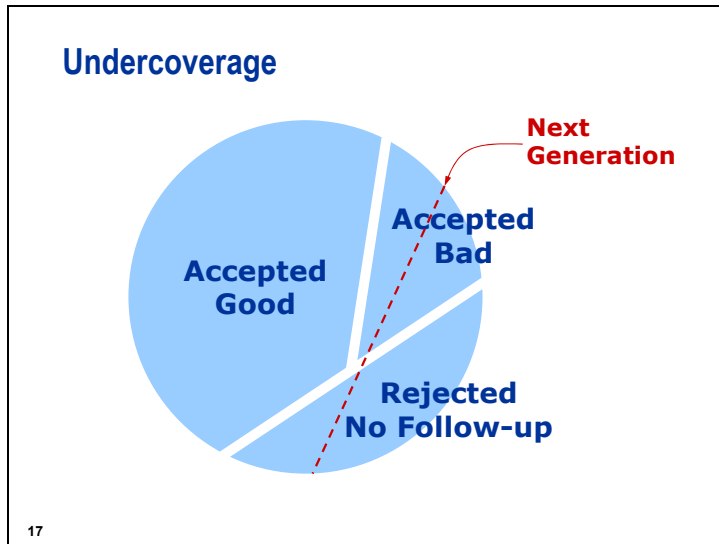
Initial data examination and analysis does not always limit the scope of the analysis. Getting acquainted with the data and examining summary statistics often inspires more sophisticated questions than were originally posed.



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

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





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

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

### Errors, Outliers, and Missings

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

18

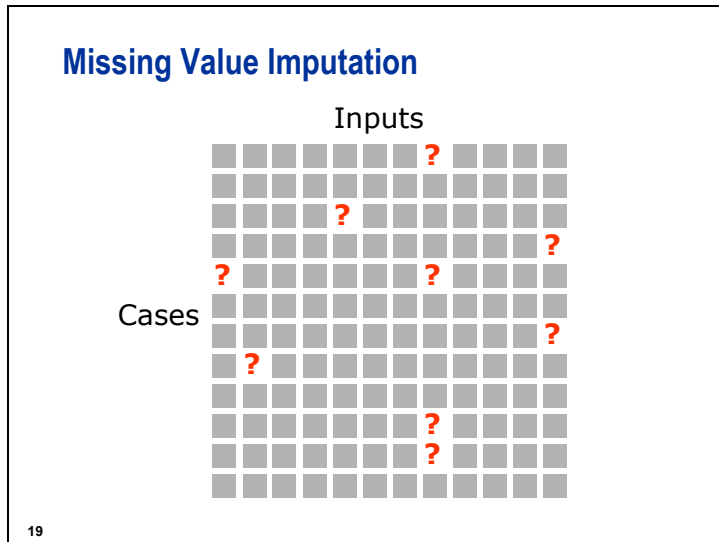
Are there any suspicious values in the above data?

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

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

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

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



Two analysis strategies are

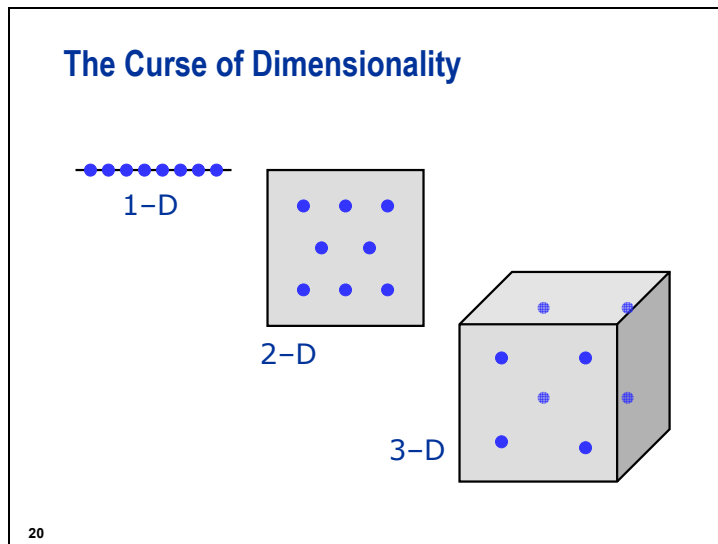
1. complete-case analysis. Use only the cases that have complete records in the analysis. If the “missingness” is related to the inputs or to the target, then ignoring missing values can bias the results.

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

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

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

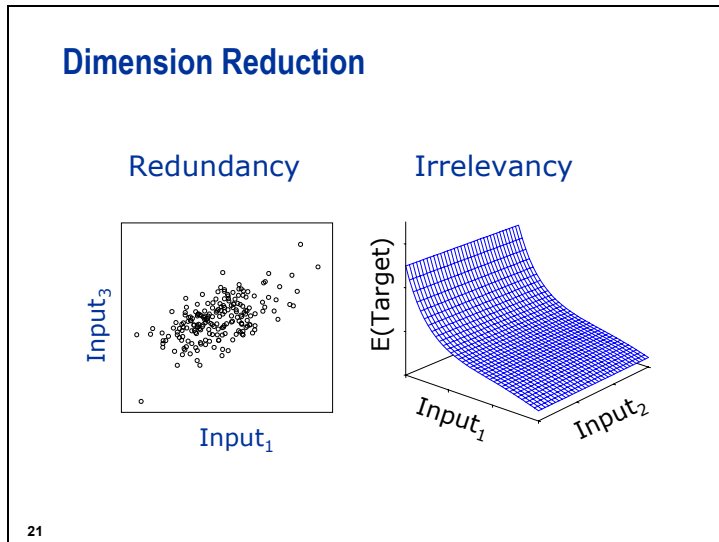
The missing values of categorical variables could be treated as a separate category. For example, type of residence might be coded as own home, buying home, rents home, rents apartment, lives with parents, mobile home, and unknown. This method would be preferable if the missingness is itself a predictor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



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

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

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

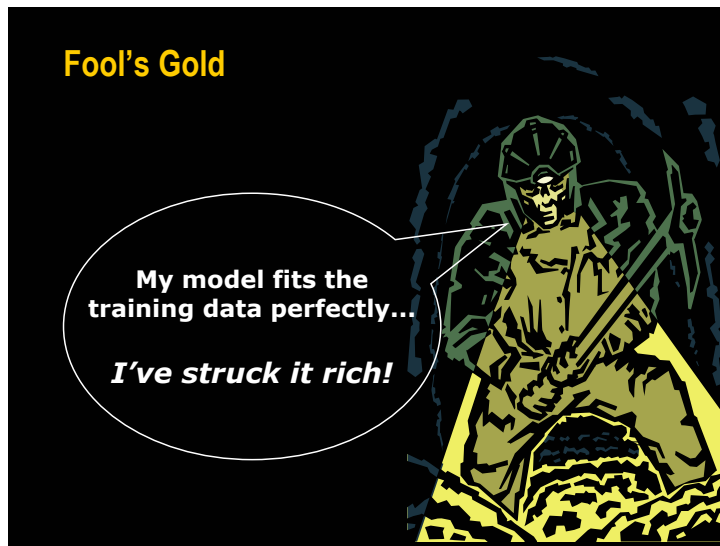


Reducing the number of inputs is the obvious way to thwart the curse of dimensionality. Unfortunately, reducing the dimension is also an easy way to disregard important information.

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

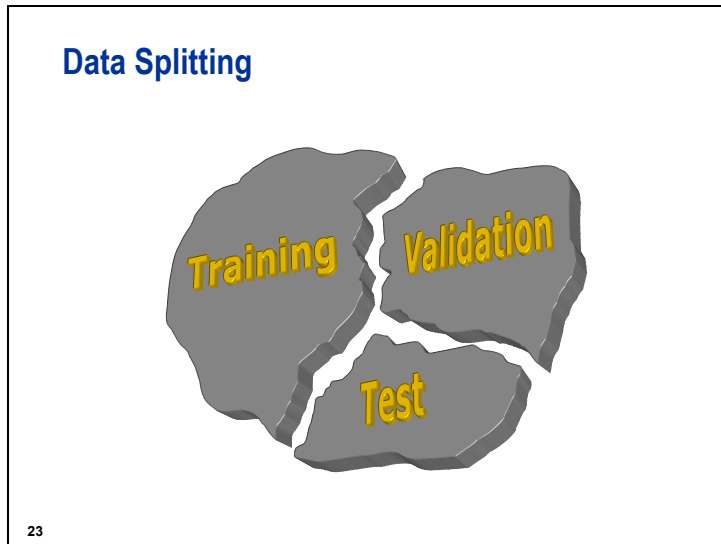
An irrelevant input is not useful in explaining variation in the target. Interactions and partial associations make irrelevancy more difficult to detect than redundancy. It is often useful to first eliminate redundant dimensions and then tackle irrelevancy.

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



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

— Mosteller and Tukey (1977)

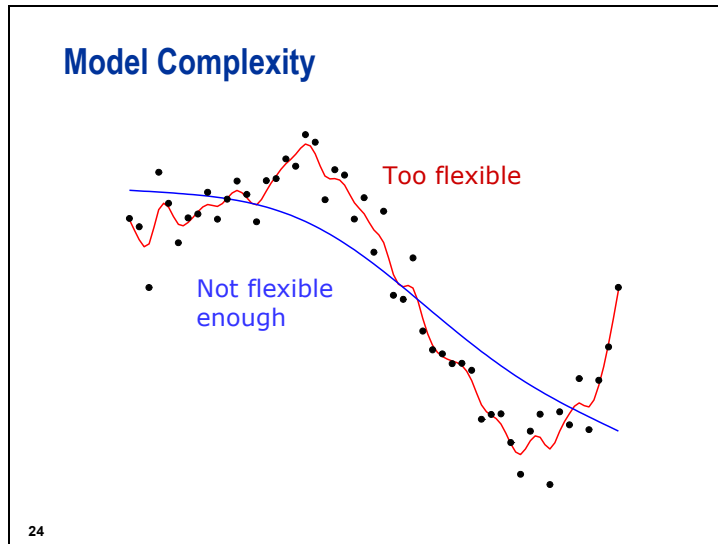


In data mining, the standard strategy for honest assessment of generalization is *data splitting*. A portion is used for fitting the model—the training data set. The rest is held out for empirical validation.

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

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

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



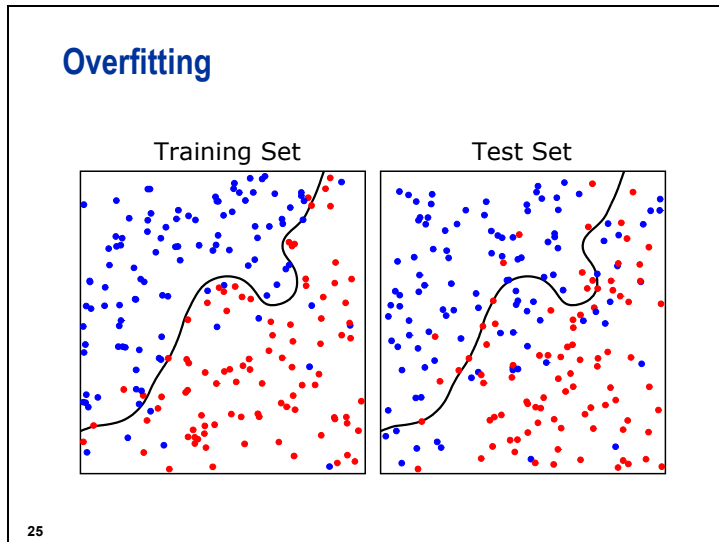
Fitting a model to data requires searching through the space of possible models. Constructing a model with good generalization requires choosing the right complexity.

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

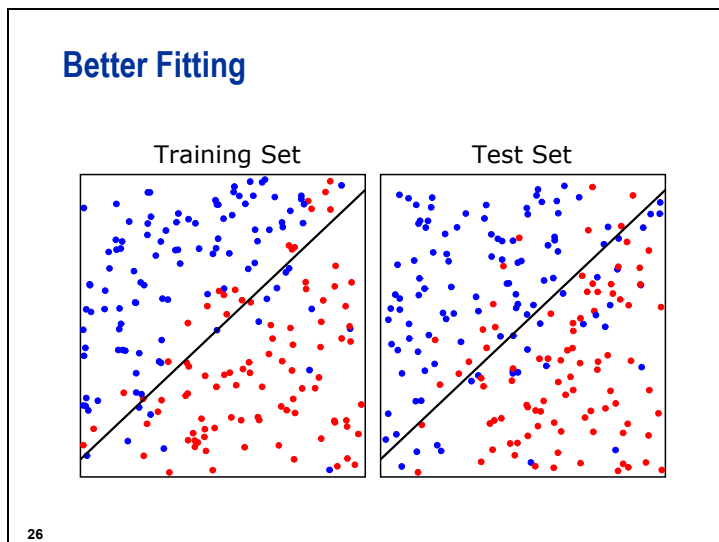
A naïve modeler might assume that the most complex model should always outperform the others, but this is not the case. An overly complex model might be too flexible. This will lead to overfitting – accommodating nuances of the random noise in the particular sample (high variance). A model with just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

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





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



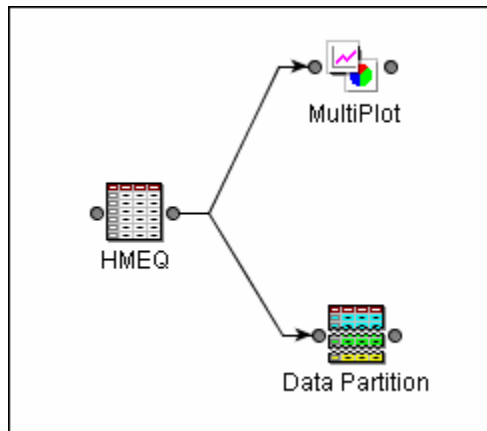
A more parsimonious model was fit to the training data. The apparent accuracy was not quite as impressive as the flexible model (34 errors, 83% accuracy), but it gave better performance on the test set (43 errors, 78.5% accuracy).



## Exploring the Data Partition Node

### Inspecting Default Settings in the Data Partition Node

1. Right-click in an empty part of the diagram workspace and select **Add Node** ⇒ **Sample** ⇒ **Data Partition**.
2. Connect the Data Partition node to the **HMEQ** node.



3. Select the Data Partition node in the workspace and examine the Properties Panel.

Property	Value
Node ID	Part
Imported Data	...
Variables	...
Partitioning Method	Default
Random Seed	12345
Data Set Percentages	
Training	40.0
Validation	30.0
Test	30.0
Status	
Last Error	
Last Status	
Needs Updating	Yes
Needs to Run	Yes
Time of Last Run	
Run Duration	

You choose the method for partitioning in the top section of the panel. By default, if the target variable is a class variable, SAS Enterprise Miner takes a stratified random sample to divide the input data table into training, validation, and test data sets. The sampling is stratified on the target variable. If the target variable is not a class variable, then a simple random sample of the input data is used.

Other sampling methods that can be chosen are

- Random – use simple random sampling regardless of the nature of the target variable.
- Cluster – use simple cluster sampling to sample from a cluster of observations that are similar in some way. For example, you might want to take a random sample of customers and then include all records of the selected customers in the sample.
- Stratify – use stratified sampling by specifying variables from the input data set to form strata (or subsets) of the total population. Within each stratum, a simple random sample is chosen.

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

The center part of the Properties Panel enables you to specify the percentage of the data to allocate to training, validation, and test data.

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

4. Set Train, Validation, and Test to **67**, **33**, and **0**, respectively.

Data Set Percentages	
Training	67.0
Validation	33.0
Test	0.0

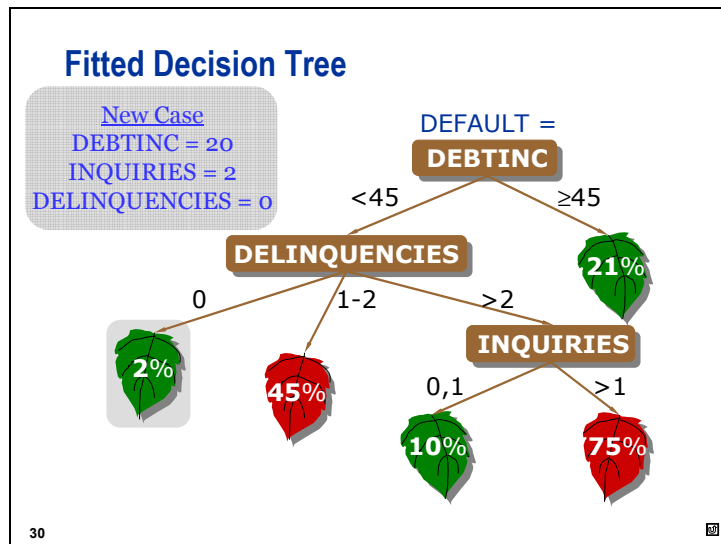


The percentage values that you configure for Training, Validation, and Test must add up to 100%. If they do not, then the percentage values for each data set are reset to the default values.

## 2.3 Introduction to Decision Trees

### Objectives

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



Banking marketing scenario:

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

Inputs = number of delinquent trade lines (**Delinquencies**)

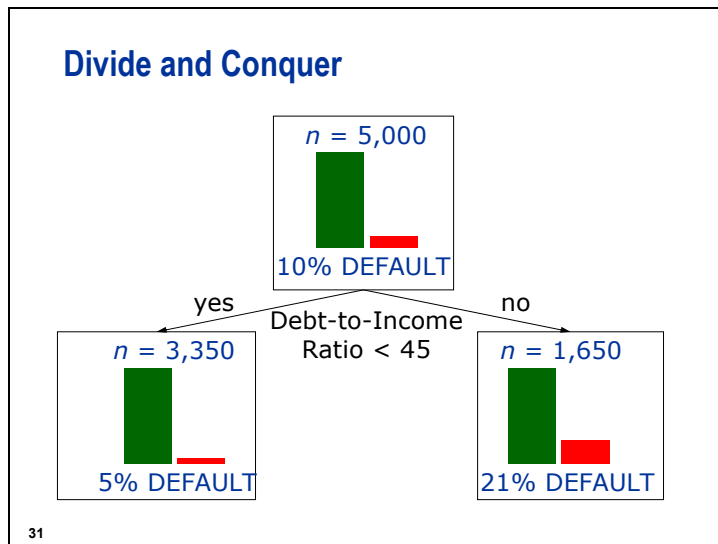
number of credit inquiries (**Inquiries**)

debt-to-income ratio (**DEBTINC**)

possibly many other inputs

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

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



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

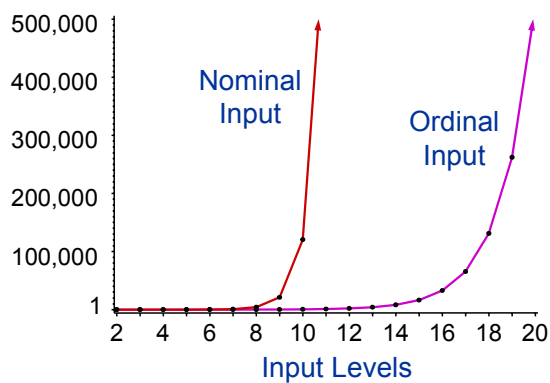
The method is recursive because each subgroup results from splitting a subgroup from a previous split. Thus, the 3,350 cases in the left child node and the 1,650 cases in the right child node are split again in similar fashion.

### The Cultivation of Trees

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

32

### Possible Splits to Consider



33

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

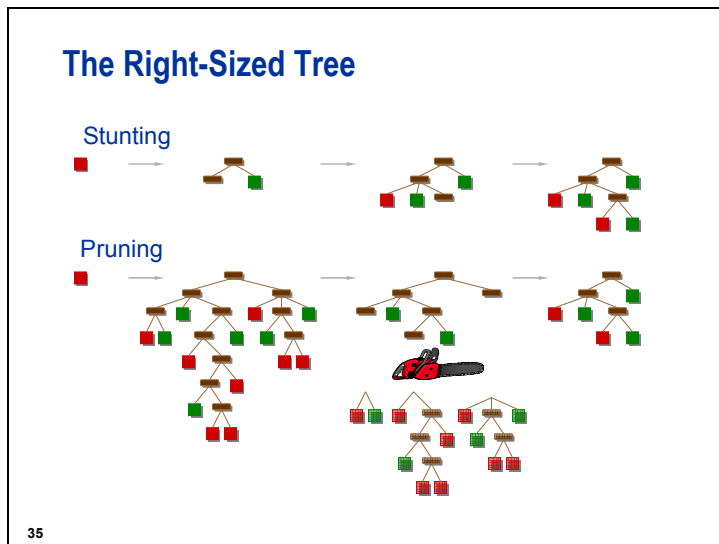
Splitting Criteria				
	Left	Right		
Not Default	3196	1304	4500	Debt-to-Income Ratio < 45
Default	154	346	500	
	Left	Center	Right	
Not Default	2521	1188	791	4500
Default	115	162	223	500
				A Competing Three-Way Split
	Left	Right		
Not Default	4500	0	4500	Perfect Split
Default	0	500	500	

34

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

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





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

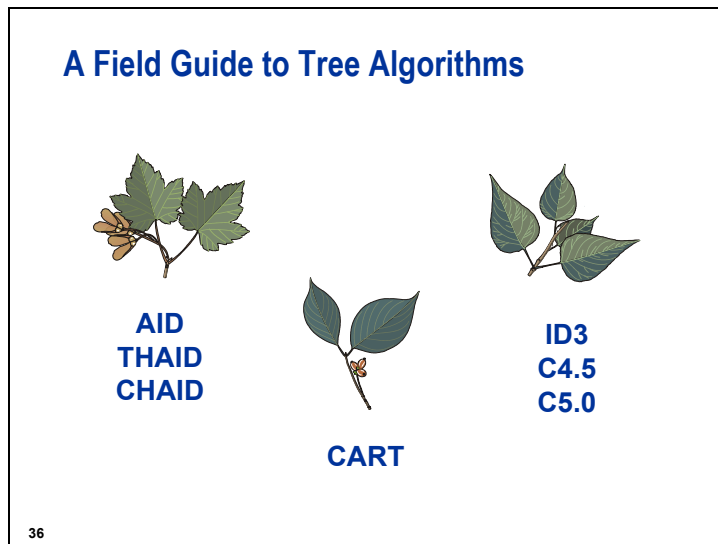
1. Using forward-stopping rules to stunt the growth of a tree (prepruning).

A universally accepted prepruning rule is to stop growing if the node is pure. Two other popular rules are to stop if the number of cases in a node falls below a specified limit or to stop when the split is not statistically significant at a specified level.

2. Growing a large tree and pruning back branches (postpruning).

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

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



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

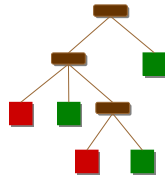
There are many variations of the CART (classification and regression trees) algorithm (Breiman et al. 1984). The standard CART approach is restricted to binary splits and uses post-pruning. All possible binary splits are considered. If the data is very large, within-node sampling can be used. The standard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

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

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

## Benefits of Trees

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



37



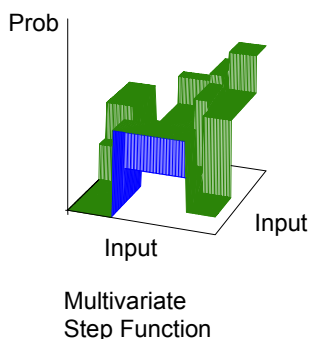
The tree diagram is useful for assessing which variables are important and how they interact with each other. The results can often be written as simple rules such as:

If (**DEBTINC**  $\geq 45$ ) or (**Debits**  $< 45$  and  $1 \leq \text{Delinquencies} \leq 2$ ) or (**Debits**  $< 45$  and **ADB**  $> 2$  and **Inquiries**  $> 1$ ), then **Default** = yes, otherwise no.

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

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

## Benefits of Trees

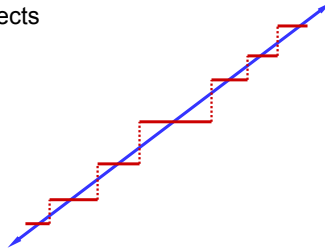


- Automatically
  - Detects interactions (AID)
  - Accommodates nonlinearity
  - Selects input variables

38

### Drawbacks of Trees

- Roughness
- Linear, Main Effects
- Instability



39

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

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

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

## 2.4 Building and Interpreting Decision Trees

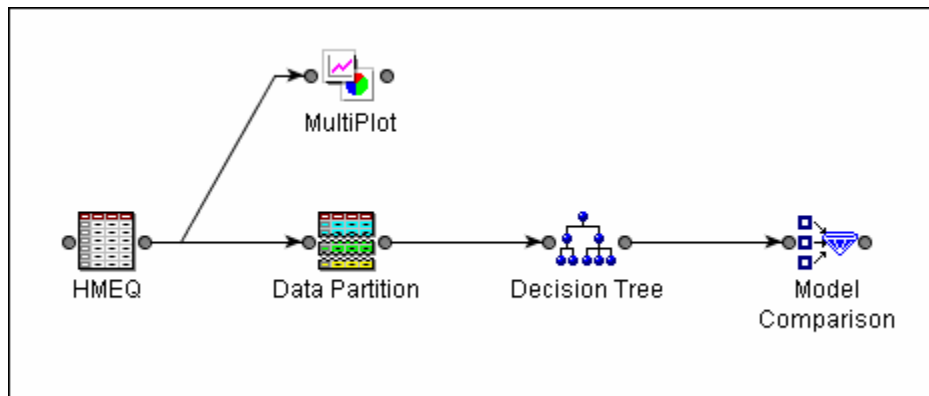
### Objectives

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



## Building and Interpreting Decision Trees


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

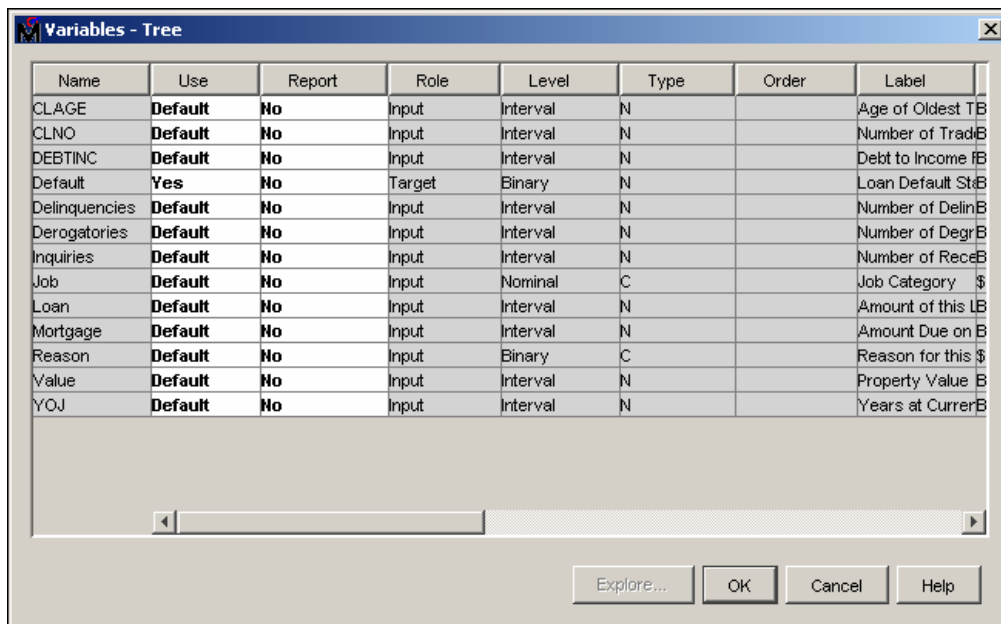


Examine the default setting for the decision tree.

1. Select the Decision Tree node and examine the Property Panel.

Property	Value
Node ID	Tree
Imported Data	...
Variables	...
Interactive Training	...
Splitting Criterion	Default
Significance Level	0.2
Missing Values	Use in search
Leaf Size	5
Maximum Branch	2
Maximum Depth	6
Minimum Categorical Size	5
Number of Rules	5
Number of Surrogate Rules	0
Split Size	

2. Select the  button in the Variables row to examine the variables to ensure all variables have the appropriate status, role, and level.



Name	Use	Report	Role	Level	Type	Order	Label
CLAGE	Default	No	Input	Interval	N		Age of Oldest TB
CLNO	Default	No	Input	Interval	N		Number of TradB
DEBTINC	Default	No	Input	Interval	N		Debt to Income fB
Default	Yes	No	Target	Binary	N		Loan Default StaB
Delinquencies	Default	No	Input	Interval	N		Number of DelinB
Derogatories	Default	No	Input	Interval	N		Number of DegrB
Inquiries	Default	No	Input	Interval	N		Number of ReceB
Job	Default	No	Input	Nominal	C		Job Category \$
Loan	Default	No	Input	Interval	N		Amount of this LB
Mortgage	Default	No	Input	Interval	N		Amount Due on B
Reason	Default	No	Input	Binary	C		Reason for this \$
Value	Default	No	Input	Interval	N		Property Value B
YOJ	Default	No	Input	Interval	N		Years at CurreB



If the role or level were not correct, it could not be corrected in this node. You would return to the Input Data node to make the corrections.

### 3. Continue to examine the Property Panel.

Many of the options discussed earlier for building a decision tree are controlled in the Property Panel.

The splitting criteria available depend on the measurement level of the target variable. For binary or nominal target variables, the default splitting criterion is the chi-square test. For ordinal target variables, Entropy is the default splitting criterion, and for interval targets the default splitting criterion is the F test. Other available methods include Gini reduction and Variance reduction.

The significance level property is used to specify the threshold  $p$ -value for the worth of a candidate splitting rule. For the Chi-square and F-test criteria, the threshold is the maximum acceptable  $p$ -value. For other criteria, the threshold is the minimum acceptable increase in the measure of worth.

The missing value property determines how splitting rules handle observations that contain a missing value for a variable. The options available include

- Use in search – uses the missing values during the split search. This is the default.
- Most correlated branch – assigns the observation with the missing value to the branch that minimizes the Sum of Squared Error among observations that contain missing values.
- Largest branch – assigns the observations that contain missing values to the largest branch.

The other options available in the Property Panel affect the growth and size of the tree. By default, only binary splits are permitted, the maximum depth of the tree is 6 levels, and the minimum number of observations in a leaf is 5.

Other properties that are shown in the Property Panel that affect the growth of the tree include:

- The minimum categorical size that specifies the minimum number of observations a category variable level must have before the level can be used in a split search.
- The number of rules property that specifies the number of splitting rules that will be saved with each node.
- The number of surrogate rules that specifies the maximum number of surrogate rules the decision tree will seek in each non-leaf node.
- The split size that specifies the required number of observations in a node in order to split the node. The default is two times the leaf size.

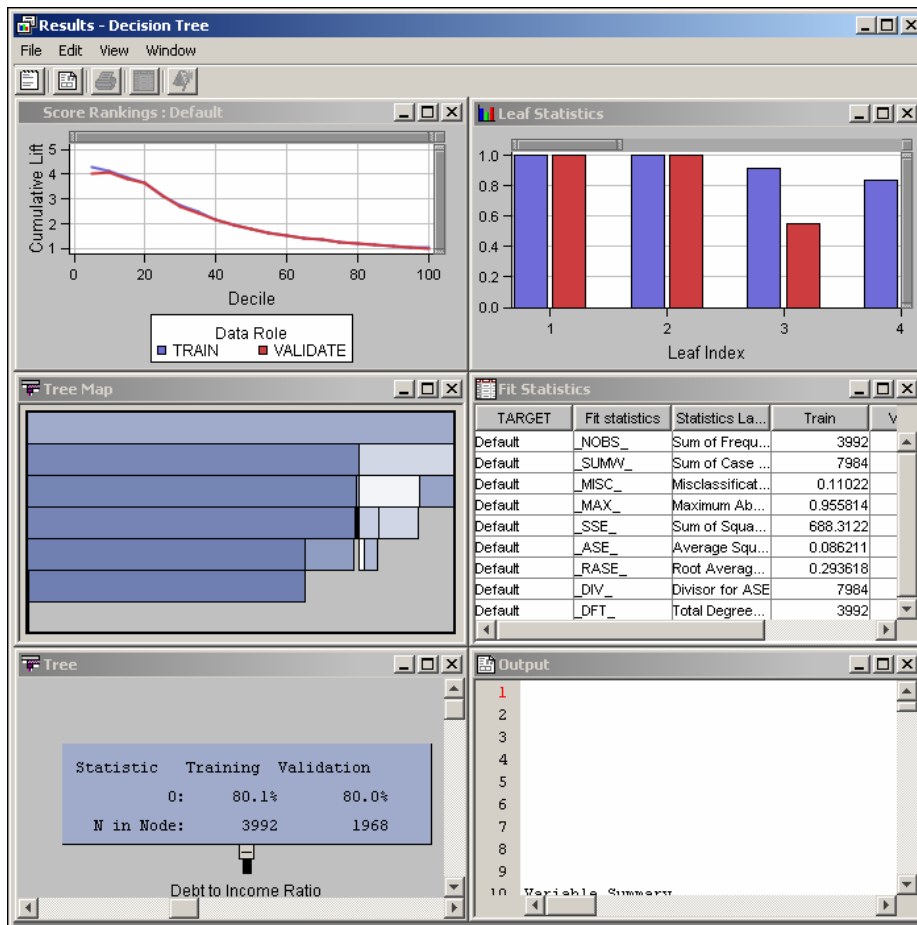


There are additional options available in the advanced Property Panel. All of the options are discussed in greater detail in the Decision Tree Modeling course.

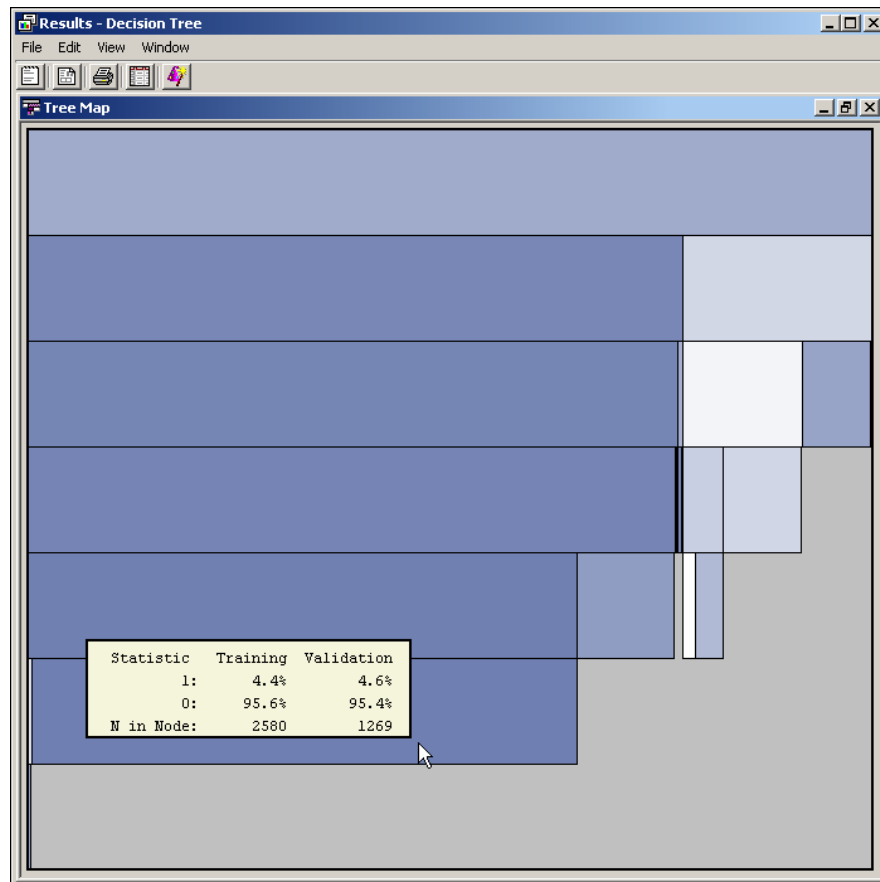
4. Leave the default setting for the tree and run the diagram from the Decision Tree node. Right-click on the Decision Tree node and select **Run**.
5. When prompted, select **Yes** to run the diagram.
6. When prompted, select **OK** to acknowledge the completion of the run.
7. Right-click on the Decision Tree node and select **Results...**.



When you view the results of the Decision Tree node, several different displays are shown by default.



8. Maximize the Tree Map window to examine it more closely.



The tree map shows the way the tree was split. The final tree appears to have 8 leaves, but it is difficult to be sure because some leaves may be so small that they are almost invisible on the map. You can use your cursor to display information on each of the leaves as well as the intermediate nodes of the tree.

9. Maximize the Output window.

The first part of the output gives information on the types of variables from the data set used in the tree.

Variable Summary		
ROLE	LEVEL	COUNT
INPUT	BINARY	1
INPUT	INTERVAL	10
INPUT	NOMINAL	1
TARGET	BINARY	1

In this case there were twelve input variables: one binary, ten interval, and one nominal. There was one binary target variable.

10. Scroll down the Output window to view the Variable Importance table.

Variable Importance						
Obs	NAME	LABEL	NRULES	IMPORTANCE	VIMPORTANCE	RATIO
1	DEBTINC	Debt to Income Ratio	1	1.00000	1.00000	1.00000
2	Delinquencies	Number of Delinquent Trade Lines	3	0.35881	0.35666	0.99399
3	Value	Property Value	2	0.27354	0.14083	0.51482
4	CLAGE	Age of Oldest Trade Line (months)	1	0.23777	0.28376	1.19341
5	YOJ	Years at Current Job	1	0.10516	0.15907	1.51258
6	Loan	Amount of this Loan	1	0.09667	0.07393	0.76476
7	Mortgage	Amount Due on First Mortgage	1	0.09585	0.04889	0.51000

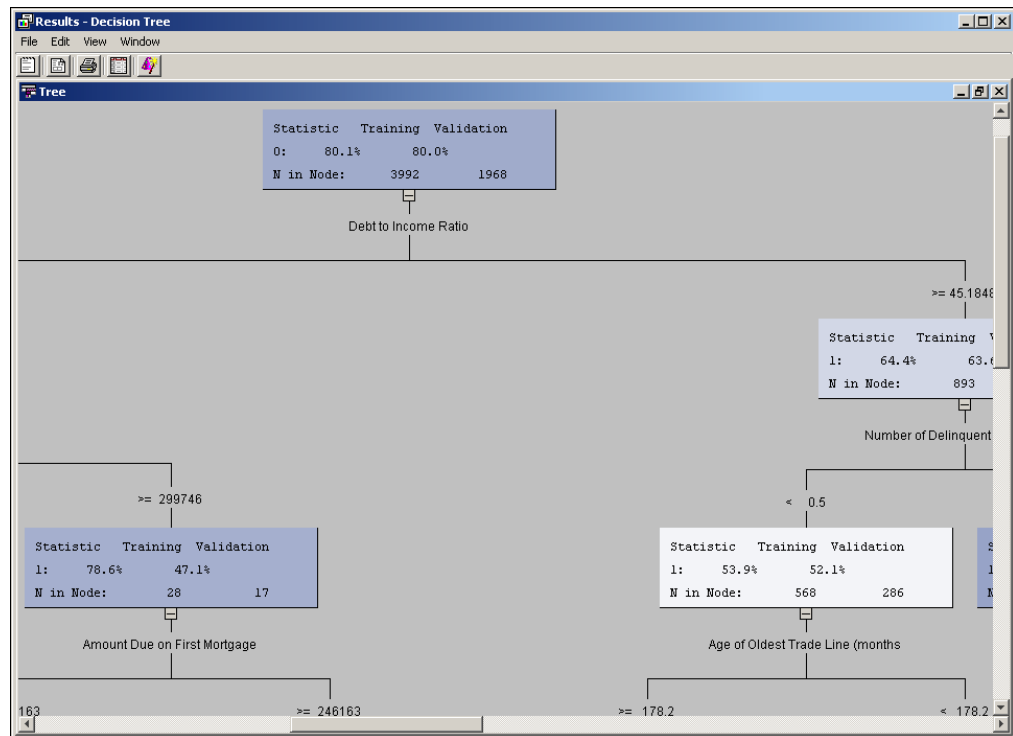
The table shows the variable name and label, the number of rules (or splits) in the tree that involve the variable (**NRULES**), the importance of the variable computed with the training data (**IMPORTANCE**), the importance of the variable computed with the validation data (**VIMPORTANCE**), and the ratio of **VIMPORTANCE** to **IMPORTANCE**. The calculation of importance is a function of the number of splits the variable is used in, the number of observations at each of those splits, and the relative purity of each of the splits. In this tree, the most important variable is the debt-to-income ratio.

11. Scroll down further in the output to examine the Tree Leaf Report.

Tree Leaf Report					
Node	Depth	Training Observations	% 1	Validation Observations	% V 1
25	5	2580	0.04	1269	0.05
17	4	464	0.15	218	0.14
12	3	374	0.65	187	0.65
7	2	325	0.83	170	0.83
19	4	134	0.25	81	0.19
18	4	60	0.50	18	0.67
10	3	23	0.91	11	0.55
31	6	13	0.15	3	0.00
9	3	9	1.00	3	1.00
30	6	5	1.00	2	1.00
11	3	5	0.20	6	0.33

This report shows that the tree actually has 11 leaves, rather than the 8 that it appeared to have from looking at the tree map. The leaves in the table are in order from the largest number of training observations to the fewest training observations. Notice that details are given for both the training and validation data sets.

Maximize the Tree window to examine the decision tree itself.



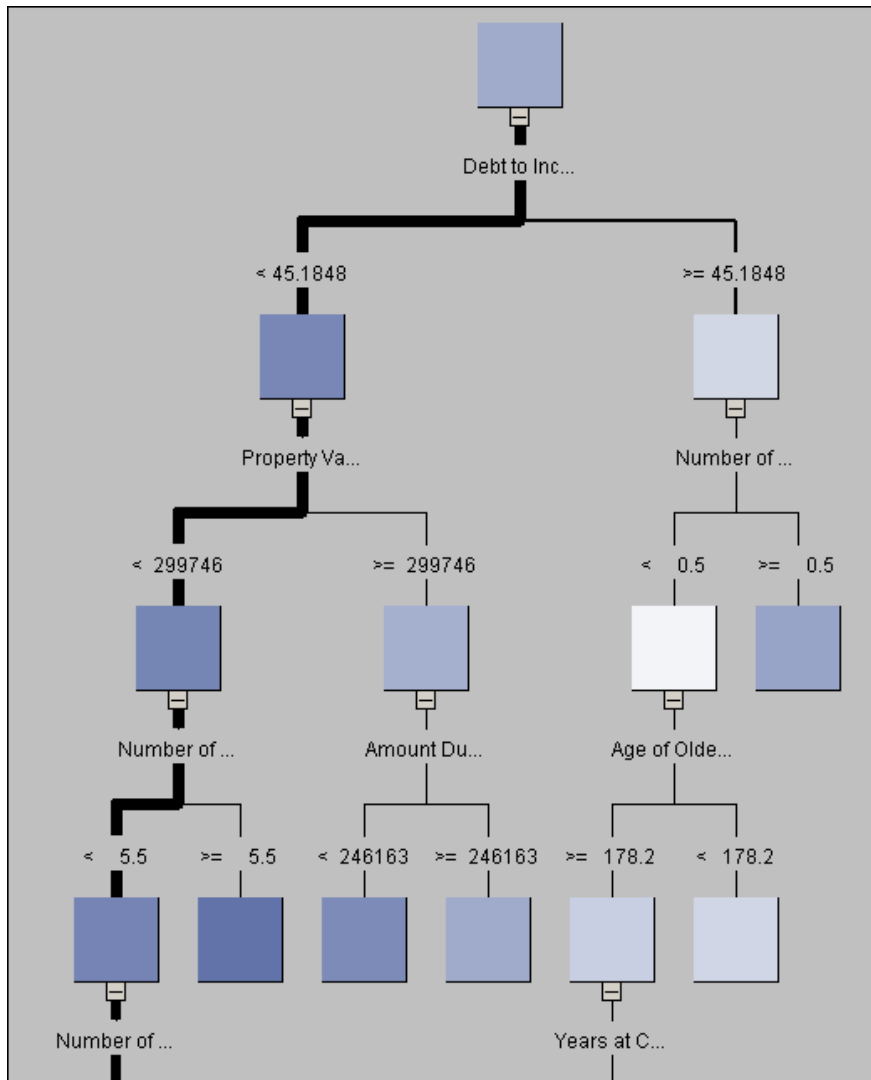
The default decision has the following properties:

- It is displayed in vertical orientation.
- The nodes are colored by the proportion of a categorical target value or the average of an interval target.
- The line width is proportional to the ratio of the number of observations in the branch to the number of observations in the root, or top, node.
- The line color is constant.

All of these default properties can be changed.

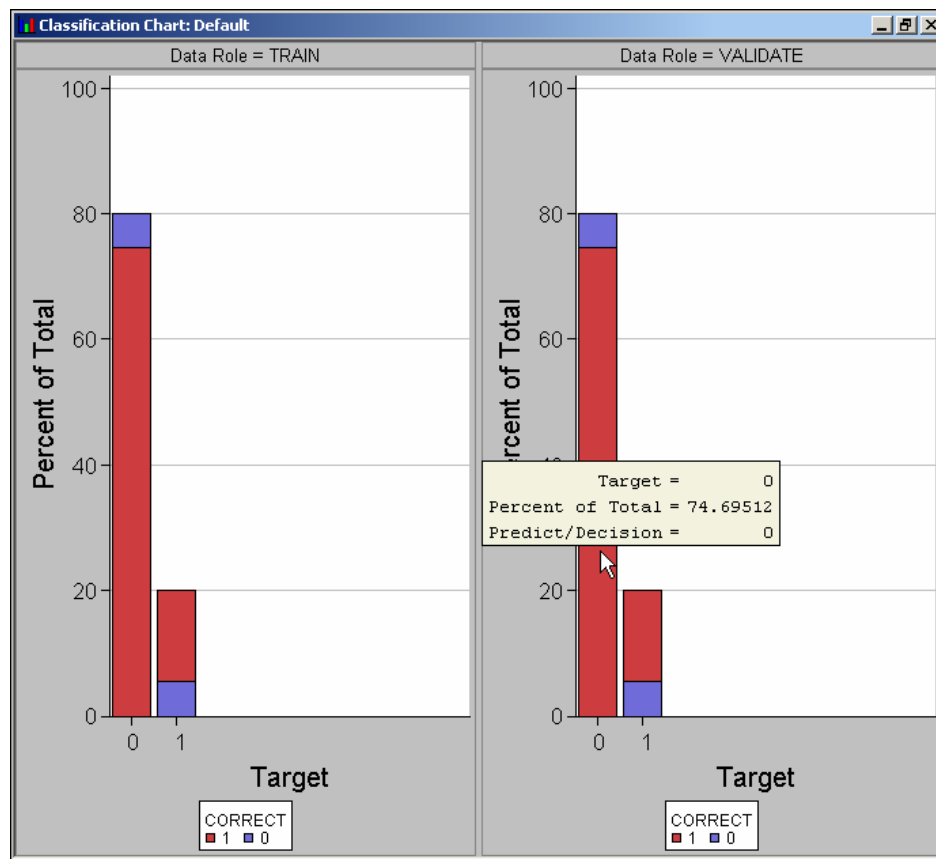
In this case you can see that the first split was based on the debt-to-income ratio, but you cannot see much of the tree because of its size. Adjust some of the tree settings to get a better picture of the tree.

1. Right-click in a blank area of the Tree window and select **Graph Properties....**
2. Select the Nodes tab in the Tree Properties window.
3. Change the Node Text option to **No Text**.
4. Select **OK**.



Although you still cannot view the entire tree, you can get a more comprehensive picture of it. You can use your cursor to examine the data values of the target variable at each of the nodes.

There are additional results that assist you in evaluating how well the tree fits the data. View the Classification Table by selecting **View** ⇒ **Model** ⇒ **Classification Chart: Default**.



The horizontal axis displays the actual target levels of the observations, and the colors indicate the classification of the observations based on the model. For example, in the validation data set, 74.7% of the observations were customers who did not default on their loan and the model predicts that they will not default.

The English Rules provide a written description of the leaves of the tree. To view the English Rules select **View** ⇒ **Model** ⇒ **English Rules**.

```

1  IF      0.5 <= Number of Delinquent Trade Lines
2  AND 45.184804524 <= Debt to Income Ratio
3  THEN
4      NODE      :      7
5      N          :      325
6      1          :      82.8%
7      0          :      17.2%
8
9  IF      5.5 <= Number of Delinquent Trade Lines
10 AND Property Value <      299746
11 AND Debt to Income Ratio < 45.184804524
12 THEN
13     NODE      :      9
14     N          :      9
15     1          :      100.0%
16     0          :      0.0%
17
18 IF  Amount Due on First Mortgage <      246163
19 AND      299746 <= Property Value
20 AND Debt to Income Ratio < 45.184804524
21 THEN
22     NODE      :      10
23     N          :      23
24     1          :      91.3%
25     0          :      8.7%
26
27 IF      246163 <= Amount Due on First Mortgage
28 AND      299746 <= Property Value
29 AND Debt to Income Ratio < 45.184804524
30 THEN
31     NODE      :      11

```

After you are finished examining the tree results, close the results and return to the diagram.

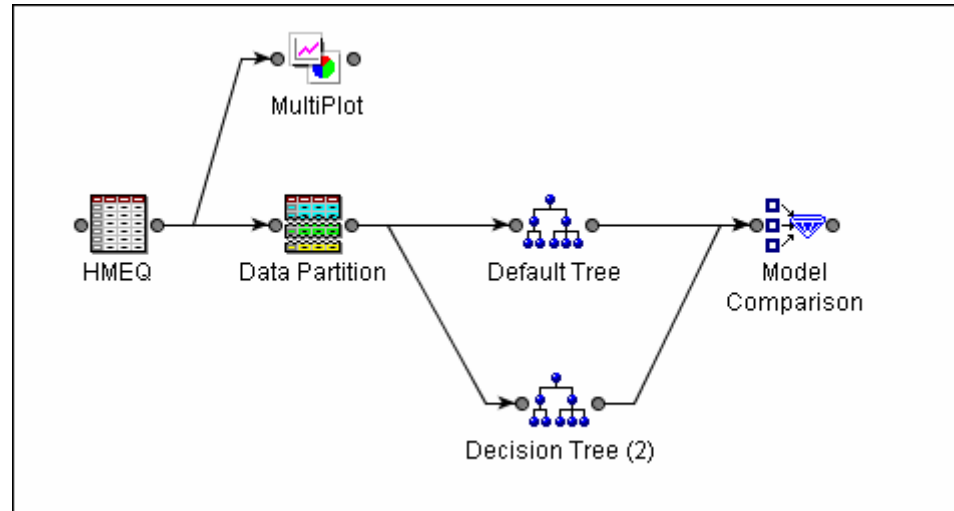
### Using Tree Options

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

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

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

1. Right-click on the Decision Tree node in the diagram and select **Rename**.
2. Change the name to **Default Tree** and select **OK**.
3. Add another Decision Tree node to the workspace.
4. Connect the Data Partition node to the Decision Tree node.
5. Connect the new Decision Tree node to the Model Comparison node.



6. Select the new Decision Tree node.
7. Examine the Property Panel.
8. Change the Maximum Branch value to **4**. This will allow binary, 3-way, and 4-way splits to be considered when the tree is growing.
9. Change the name of the node to **4-way Tree (2)** in the diagram.
10. Run the flow from this Decision Tree node and view the results.

The number of leaves in the new tree has increased from 11 to 26. It is a matter of preference as to whether this tree is more comprehensible than the binary split tree. The increased number of leaves suggests to some a lower degree of comprehensibility. Both the average squared error and the misclassification rate for the validation data set have increased slightly with the larger tree.

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

11. Close the Results window.

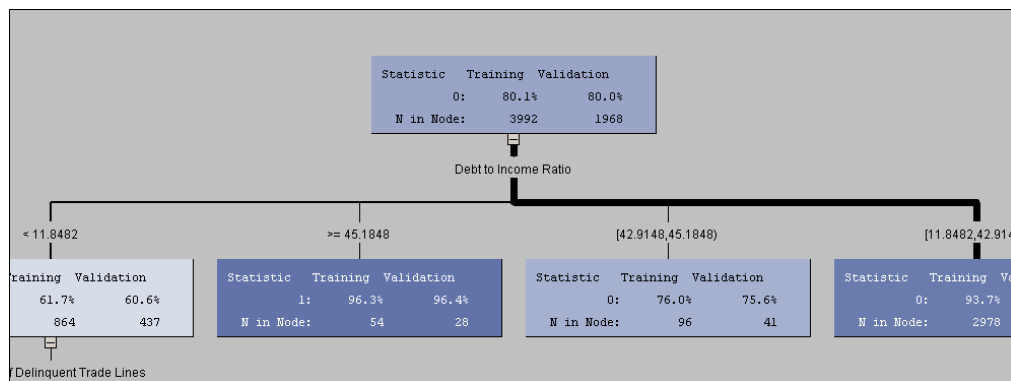


## Limiting Tree Growth

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

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

1. Select the 4-way Tree node in the diagram.
2. In the Property Panel, change Leaf Size to 25, and then press the ENTER key.
3. Rerun the 4-way Tree node and view the results as before. The optimal tree now has 10 leaves.
4. View the resulting tree.



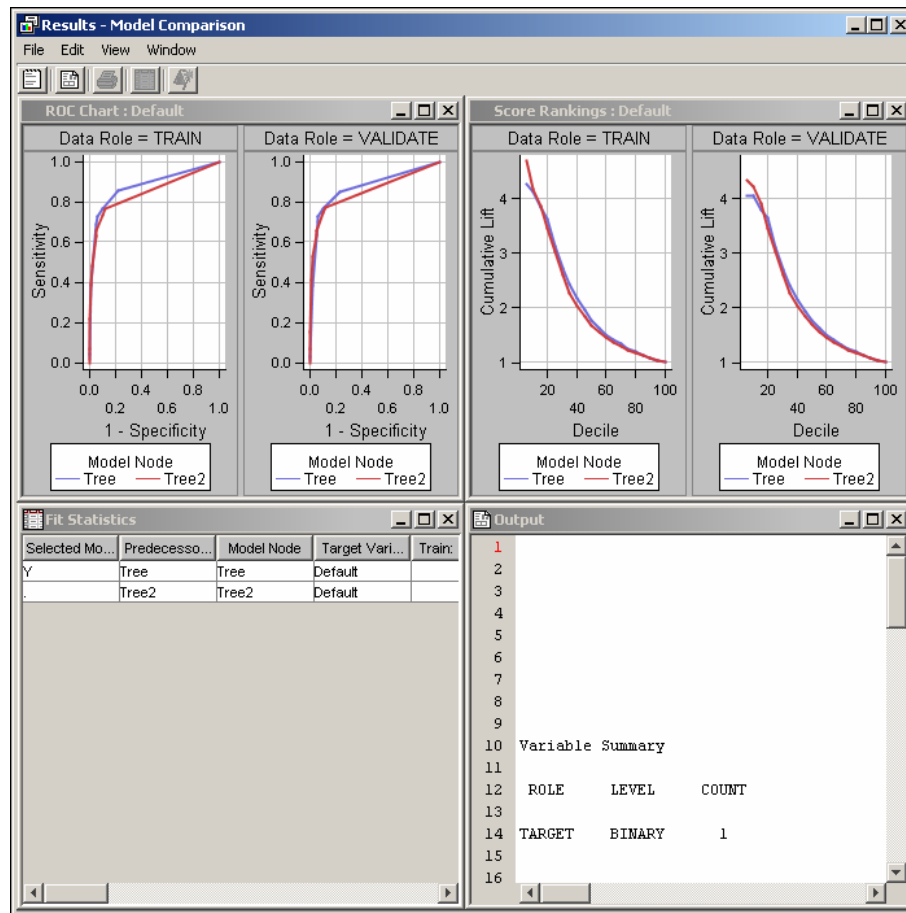
Notice that the initial split on **DEBTINC** has produced four branches.

5. Close the results when you are finished viewing them.

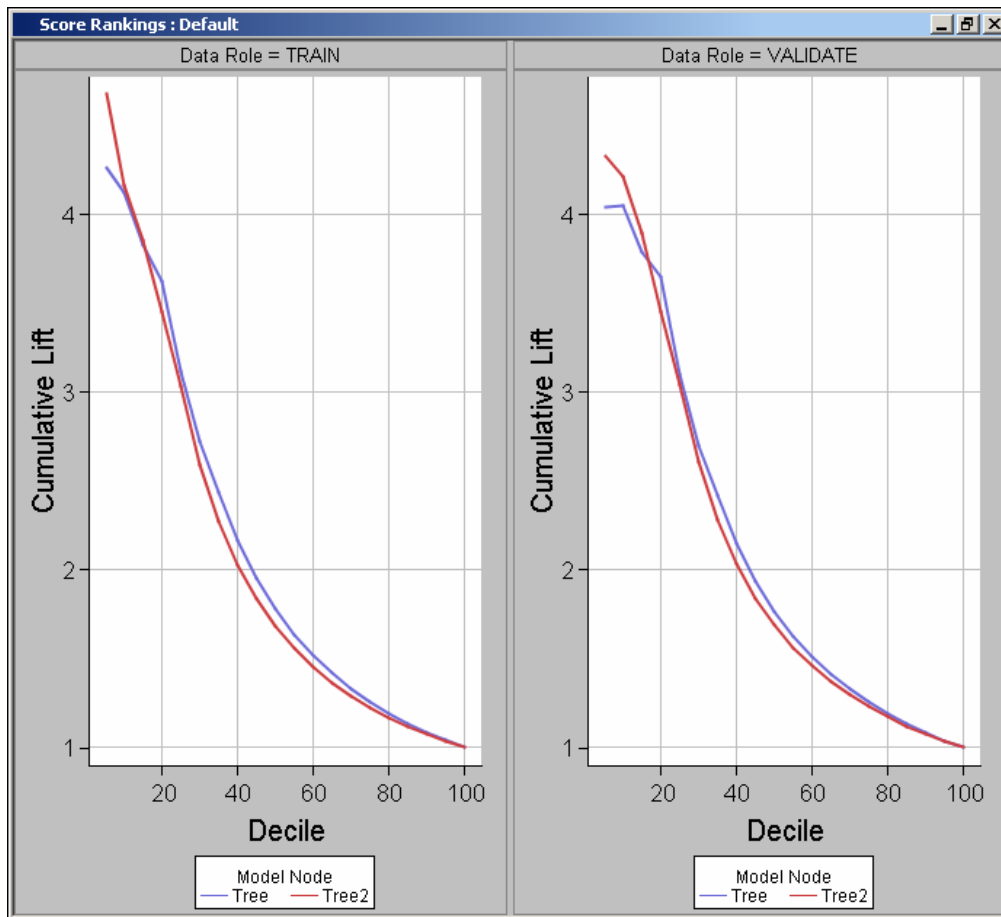
## Comparing Models

1. To run the diagram from the Model Comparison node, right-click on the node and select **Run**.
2. When prompted, select **Yes** to run the path.
3. Select **OK** to confirm that the run was completed.

4. Right-click on the Model Comparison node and select **Results...**



Maximize the Score Rankings graphs and examine them more closely.



A Cumulative Lift chart is shown by default. To see actual values, place the cursor over the plot at any decile. For example, placing the cursor over the plot for Tree2 in the validation graph indicates a lift of 4.2 for the 10<sup>th</sup> percentile.

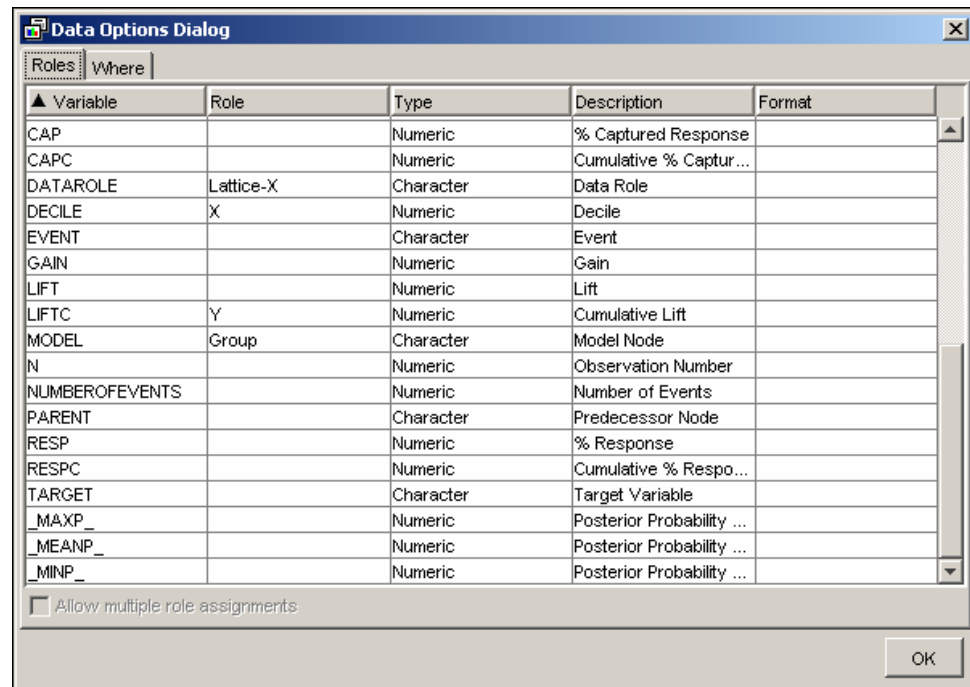
To interpret the Cumulative Lift chart, consider how the chart is constructed.

- For this example, a responder is defined as someone who defaulted on a loan (**Default** = 1). For each person, the fitted model (in this case, a decision tree) predicts the probability that the person will default. Sort the observations by the predicted probability of response from the highest probability of response to the lowest probability of response.
- Group the people into ordered bins, each containing approximately 5% of the data in this case.
- Using the target variable **Default**, count the percentage of actual responders in each bin and divide that by the population response rate (in this case, approximately 20%).

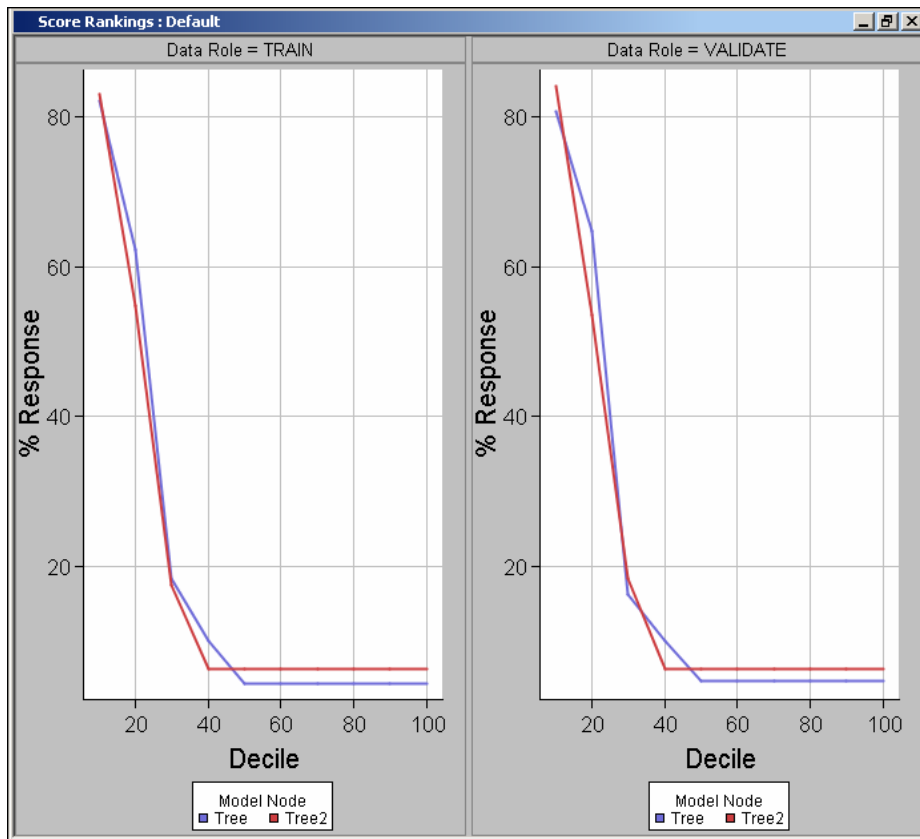
If the model is useful, the proportion of responders (defaulters) will be relatively high in bins where the predicted probability of response is high. The lift chart curve shown above shows the percentage of respondents in the top 10% is four times higher than the population response rate of 20%. Therefore, the lift chart plots the relative improvement over the expected response if you were to take a random sample.

You have the option to change the values that are graphed on the plot. For example, if you are interested in the percent of responders in each decile, you might change the vertical axis to percent response.

1. With the Score Rankings window as the active window, select **Edit** ⇒ **Data Options...**



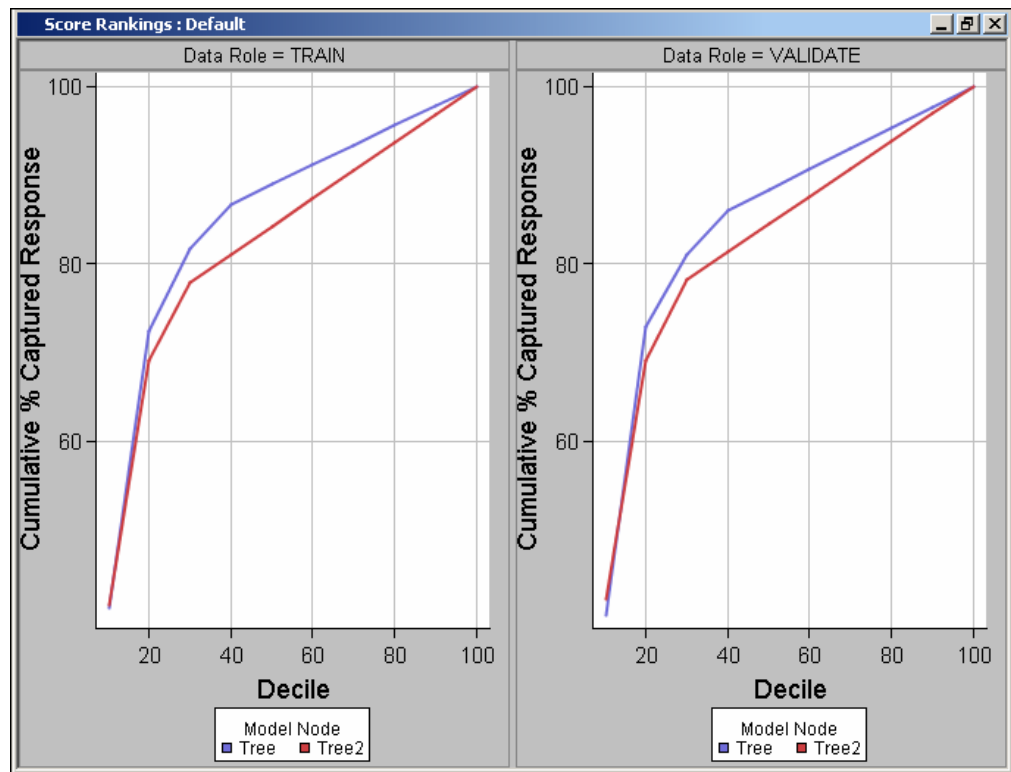
2. Scroll down the the Data Options Dialog and notice that **LIFTC** has the role Y. Scroll down and find the variable **RESP**, which represents percent response.
3. Assign **RESP** the role of the Y variable in the graph by clicking in the role column of the **RESP** row and selecting **Y**.
4. Select **OK**.



This non-cumulative percent response chart shows that after you get beyond the 30<sup>th</sup> percentile for predicted probability, the default rate is lower than what you would expect if you were to take a random sample.

Instead of asking the question, "What percentage of observations in a bin were responders?", you could ask the question, "What percentage of the total number of responders are in a bin?" This can be evaluated using the cumulative percent captured response as the response variable in the graph.

1. With the Score Rankings window as the active window, select **Edit** ⇒ **Data Options...**.
2. Scroll down the the Data Options Dialog and assign **CAPC** the role of the Y variable by clicking in the role column of the **CAPC** row and selecting **Y**.
3. Select **OK**.



Observe that using the validation data set and original tree model, if the percentage of applications chosen for rejection were approximately

- 20%, then you would have identified over 70% of the people who would have defaulted.
- 40%, then you would have identified over 80% of the people who would have defaulted.



ROC Charts will be discussed later in this course.

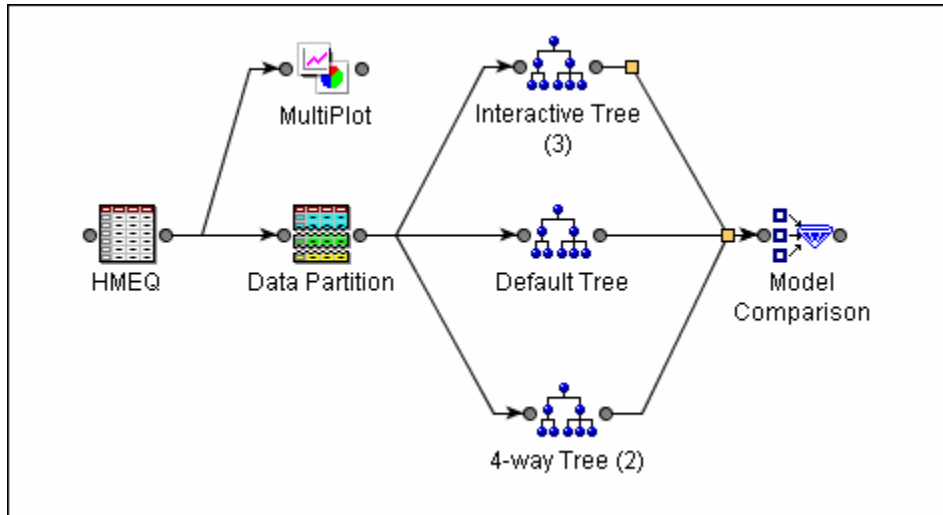
### Interactive Training

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

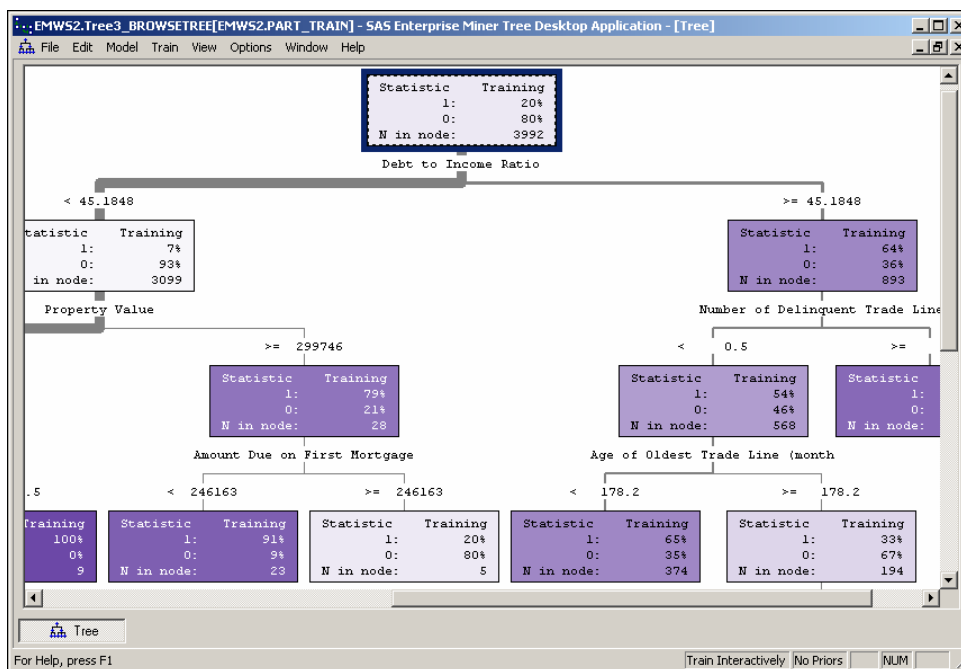
Likewise, splits may be selected that are statistically optimal but may be in conflict with an existing business practice. For example, the credit department may treat applications where debt-to-income ratios are not available differently from those where this information is available. You can incorporate this type of business rule into your decision tree using interactive training in the Decision Tree node. It might then be interesting to compare the statistical results of the original tree with the changed tree. In order to accomplish this, first make a copy of the Default Tree node.

1. Select the Default Tree node with the right mouse button and then select **Copy**.

2. Move your cursor to an empty place above the Default Tree node, right-click, and select **Paste**. Rename this node **Interactive Tree (3)**.
3. Connect the Interactive Tree node to the Data Partition node and the Model Comparison node as shown.



4. In order to do interactive training of the tree you must first run the Decision Tree node. Right-click on the new Decision Tree node and select **Run**.
5. When prompted, select **Yes** to run the diagram and select **OK** to acknowledge the completion of the run.
6. Select the Interactive Tree (3) node and in the Property Panel, select the button in the Interactive Training row. The Tree Desktop Application will open.



7. Select **View** ⇒ **Competing Rules** from the menu bar.

Variable	-Log(p)	Branches
Debt to Income Ratio	309.414856	2
Number of Delinquent Trade Lines	81.374315	2
Property Value	60.260774	2
Number of Degrogatory Reports	55.022579	2
Age of Oldest Trade Line (months)	26.746714	2

The Competing Rules window lists the top five inputs considered for splitting as ranked by their logworth (negative the log of the  $p$ -value of the split).

8. To view the actual values of the splits considered, select **View** ⇒ **Competing Rules Details** from the menu bar.

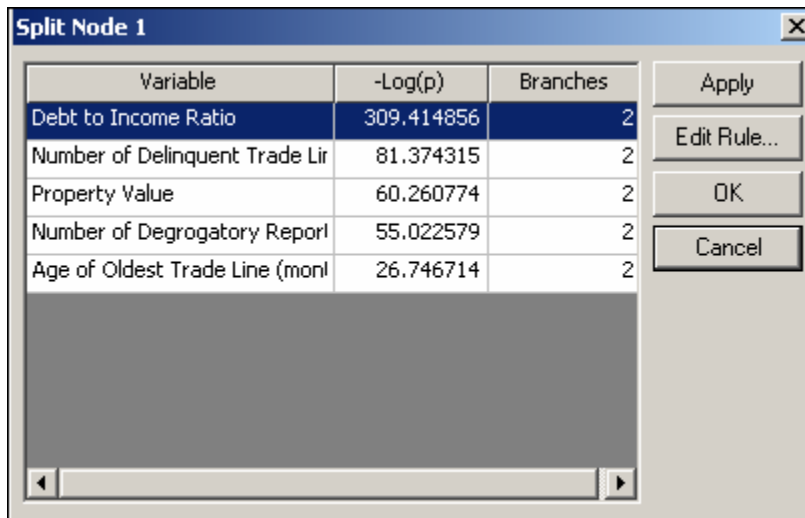
Branch	Variable	Values
1	Debt to Income Ratio	< 45.1848
2		>= 45.1848 or Missing
1	Number of Delinquent Trade Lines	< 0.5 or Missing
2		>= 0.5
1	Property Value	All except missing
2		Missing only
1	Number of Degrogatory Reports	< 0.5 or Missing
2		>= 0.5
1	Age of Oldest Trade Line (months)	< 150.205 or Missing
2		>= 150.205

As you can see in the table, the first split for the tree was on the debt-to-income ratio. Those observations with a debt-to-income ratio less than 45.1848 are in one branch. Those observations with a higher debt-to-income ratio or where the debt-to-income ratio is unknown are in the other branch.

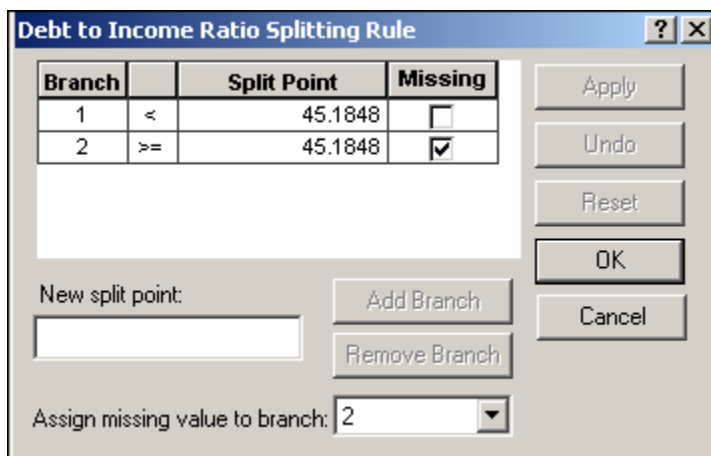
Your goal is to modify the initial split so that one branch contains all the applications with missing debt-to-income data and the other branch contains the rest of the applications. From this initial split, you will use the decision tree's analytic method to grow the remainder of the tree.



1. Right-click on the root node of the tree and select **Split Node...**. A window opens listing potential splitting variables and a measure of the worth of each input.



2. Select the row corresponding to **Debt to Income Ratio**.
3. Select **Edit Rule**.



4. At the bottom of the window, change the Assign missing value to branch field to **Missing only**. This creates a third branch for the split that only contains the missing values.

**Debt to Income Ratio Splitting Rule** [?] [X]

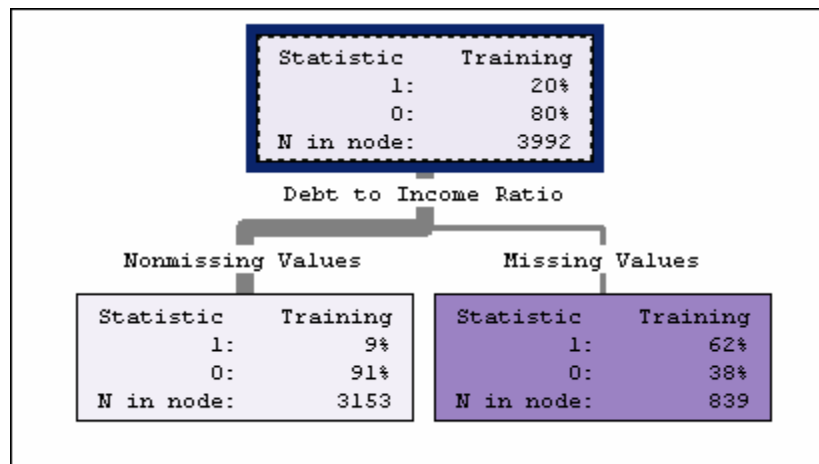
Branch		Split Point	Missing
1	<	45.1848	<input type="checkbox"/>
2	>=	45.1848	<input type="checkbox"/>
3		Missing only	<input checked="" type="checkbox"/>

New split point:  Add Branch Remove Branch

Assign missing value to branch: **Missing only** [v]

Apply Undo Reset OK Cancel

- Select the row for Branch 2, and then select **Remove Branch**. The split is now defined to put all nonmissing values of **DEBTINC** into Branch 1 and all missing values of **DEBTINC** into Branch 2.
- Select **Apply** to apply the new split to the tree.
- Select **OK** to close the Splitting Rule window.
- Select **OK** in the Split Node 1 window. The window closes and the tree diagram is updated as shown.



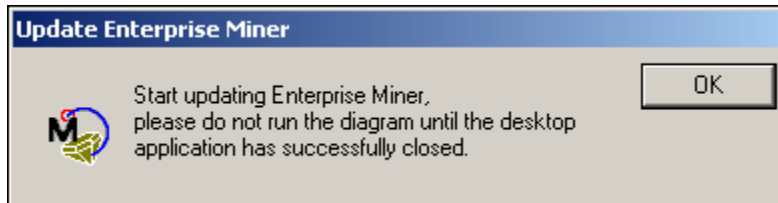
The left node contains all observations with a nonmissing value of **DEBTINC**, and the right node contains only observations with missing values for **DEBTINC**.

- To rerun the tree with this as the first split, select **Train** ⇌ **Train**.

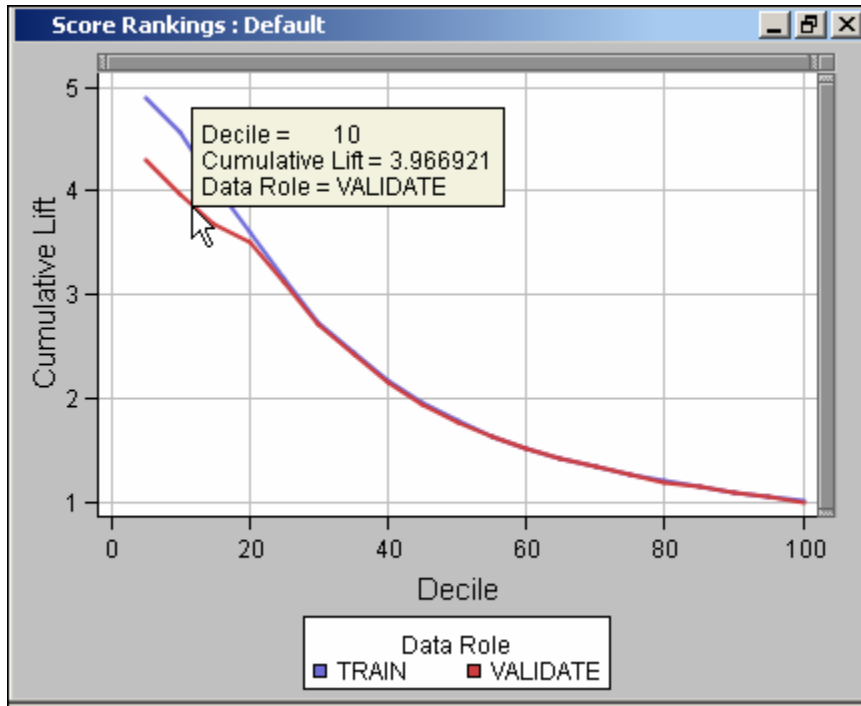
At this point you can examine the resulting tree. However, to evaluate the tree you must close the tree desktop application and run the tree in interactive mode.

- Close the Tree Desktop Application and select **Yes** when prompted to save the changes.

11. Select **OK** to acknowledge the need to run the tree in the interactive mode after the desktop application closes.



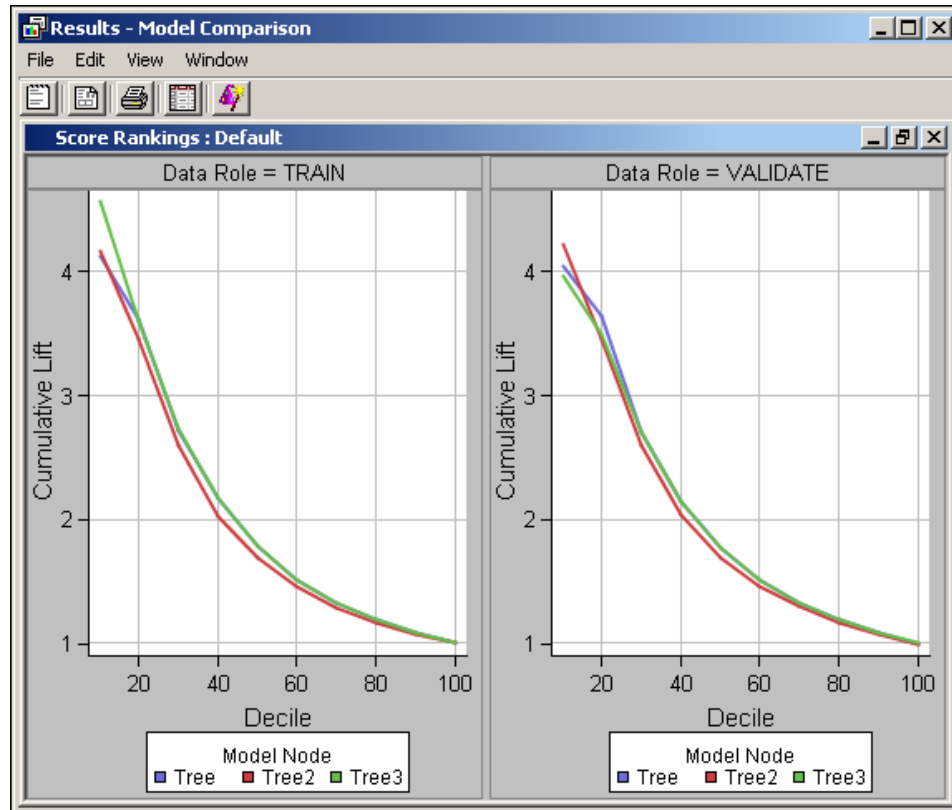
12. Run the modified Interactive Tree node and view the results.



The lift at the 10<sup>th</sup> percentile of the validation data set for this tree is just under 4, which is similar to, but not quite as good as, the trees constructed earlier.

To compare the tree models:

1. Close the Results window.
2. Run the diagram from the Model Comparison node.
3. After the diagram run is completed, right-click on the Model Comparison node and select **Results...**. Examine the cumulative lift chart.



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

### Consequences of a Decision

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

43

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

### Example

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

44

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

### Consequences of a Decision

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

45

The costs of misclassification are shown in the table.

### Bayes Rule

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

46

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

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

### Consequences of a Decision

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

47

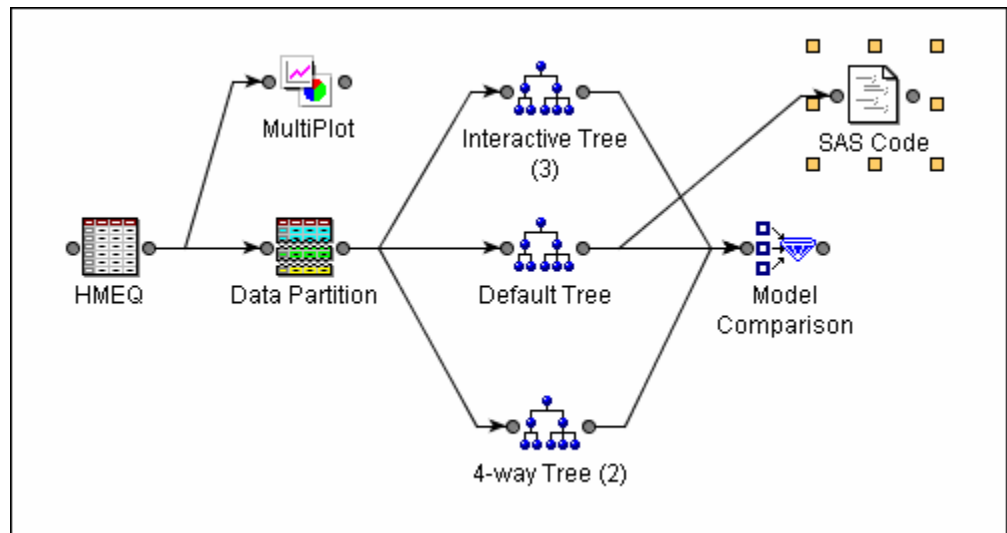
You can obtain the same result in SAS Enterprise Miner by using the profit matrix to specify the profit associated with the level of the response being modeled (in this case, a loan default or a 1). You will see this in the next chapter.




## Choosing a Decision Threshold

Consider the decision threshold determined theoretically. Because there is very little difference between the trees, use the default tree for this demonstration.

1. Return to Diagram1 and add a SAS Code node.
2. Connect the SAS Code node to the Default Tree node.




3. With the SAS Code node selected, examine the Property Panel.
4. Select the  button in the Variables row of the Property Panel.



Name	Role	Level	Type	Order	Label	Form
CLAGE	Input	Interval	N		Age of Oldest Trade Line	BEST12.
CLNO	Rejected	Interval	N		Number of Trade Lines	BEST12.
DEBTINC	Input	Interval	N		Debt to Income Ratio	BEST12.
Default	Target	Binary	N		Loan Default Status	BEST12.
Delinquencies	Input	Interval	N		Number of Delinquent Tra	BEST12.
Derogatories	Rejected	Interval	N		Number of Degrogatory R	BEST12.
F_Default	Classification	Nominal	C		From: Default	
Inquiries	Rejected	Interval	N		Number of Recent Credit	BEST12.
I_Default	Classification	Nominal	C		Into: Default	
Job	Rejected	Nominal	C		Job Category	\$F7.
Loan	Input	Interval	N		Amount of this Loan	BEST12.
Mortgage	Input	Interval	N		Amount Due on First Mort	BEST12.
P_Default0	Prediction	Interval	N		Predicted: Default=0	
P_Default1	Prediction	Interval	N		Predicted: Default=1	
Reason	Rejected	Binary	C		Reason for this Loan	\$F7.
R_Default0	Residual	Interval	N		Residual: Default=0	
R_Default1	Residual	Interval	N		Residual: Default=1	

Buttons: Explore... OK Cancel Help


Notice that the variable **P\_Default1** is the predicted probability of a default from the tree model.

5. Select **OK** to close the Variables – EMCODE window.
6. Select the  button in the Imported Data row of the Property Panel.

Port	Source	Table	Role
DATA	TRAIN port of Default Tree	EMWS.Tree_TRAIN	Train
VALIDATE	VALIDATE port of Default Tree	EMWS.Tree_VALIDATE	Validate
TEST			
SCORE			
TRANSACTION			
DOCUMENT			
RULES			
REPORTFIT	REPORTFIT port of Default Tree	EMWS.Tree_EMREPORTFIT	Report Fit
RANK	RANK port of Default Tree	EMWS.Tree_EMRANK	Rank
SCOREDIST	SCOREDIST port of Default Tree	EMWS.Tree_EMSCOREDIST	Score Dist.
ESTIMATE			
TREE	TREE port of Default Tree	EMWS.Tree_EMTREE	Tree

Buttons: Browse... Explore... Properties... OK

Notice that the validation data table coming from the tree has been named **EMWS.Tree\_VALIDATE**. The library name may vary depending upon what libraries have been previously created.

7. Select **OK** to close the Imported Data – SAS code window.
8. Select the  button in the SAS Code row of the Property Panel. A SAS Code window opens.

9. Type in the SAS code shown below.

```
proc means data=emws.tree_validate n;
  var p_default1;
  title 'Total Observations in Validation Data Table';
run;

data work.cutoff;
  set emws.tree_validate;
  where p_default1 ge 0.33;
run;

proc means data=work.cutoff n;
  var p_default1;
  title 'Number of Observations where P_Default1 '
    'greater than or equal to 0.33';
run;
```

10. Select **OK** to save the code.

11. Run the SAS code and examine the results of the MEANS procedures.

```
Total Observations in the Validation Data Table

The MEANS Procedure

Analysis Variable : P_Default1 Predicted: Default=1

      N
-----
      1968
-----

Number of Observations where P_Default1 greater than or equal to 0.33

The MEANS Procedure

Analysis Variable : P_Default1 Predicted: Default=1

      N
-----
      391
-----
```

Therefore, based on the theoretical approach, 391 out of 1968 applications, or approximately 19%, should be rejected.

As you will see in the next chapter, you can obtain the same result using the Model Comparison node if you input the profits associated with the decisions.

## 2.5 Exercises

### 1. Initial Data Exploration

A supermarket is beginning to offer a line of organic products. The supermarket's management would like to determine which customers are likely to purchase these products.

The supermarket has a customer loyalty program. As an initial buyer incentive plan, the supermarket provided coupons for the organic products to all of their loyalty program participants and have now collected data that includes whether or not these customers have purchased any of the organic products.

The **ORGANICS** data set contains over 22,000 observations and 18 variables. The variables in the data set are shown below with the appropriate roles and levels.

Name	Model Role	Measurement Level	Description
CUSTID	ID	Nominal	Customer loyalty identification number
GENDER	Input	Nominal	M = male, F = female, U = unknown
DOB	Rejected	Interval	Date of birth
EDATE	Rejected	Unary	Date extracted from the daily sales data base
AGE	Input	Interval	Age, in years
AGEGRP1	Input	Nominal	Age group 1
AGEGRP2	Input	Nominal	Age group 2
TV_REG	Input	Nominal	Television region
NGROUP	Input	Nominal	Neighborhood group
NEIGHBORHOOD	Input	Nominal	Type of residential neighborhood
LCDATE	Rejected	Interval	Loyalty card application date
LTIME	Input	Interval	Time as loyalty card member
ORGANICS	Target	Interval	Number of organic products purchased
BILL	Input	Interval	Total amount spent
REGION	Input	Nominal	Geographic region
CLASS	Input	Nominal	Customer loyalty status: tin, silver, gold, or platinum
ORGYN	Target	Binary	Organics purchased? 1 = Yes, 0 = No
AFFL	Input	Interval	Affluence grade on a scale from 1 to 30

Although two target variables are listed, these exercises concentrate on the binary variable **ORGYN**.

- a. Set up a new project for this exercise with Exercise as the project name.
- b. Create a new diagram called **Organics**.
- c. Define the data set **ADMT.ORGANICS** as a data source for the project.
- d. Set the model role for the target variable and examine the distribution of the variable. What is the proportion of individuals who purchased organic products?
- e. Do you have any reservations about any of the other variables in the data set? Are there any variables that should not be included as input variables in your analysis?
- f. The variables **AGE**, **AGEGRP1**, and **AGEGRP2** are all different measurements for the same information. Presume that, based on previous experience, you know that **AGE** should be used for this type of modeling. Set the model role for **AGEGRP1** and **AGEGRP2** to rejected.
- g. The variable **NGROUP** contains collapsed levels of the variable **NEIGHBORHOOD**. Therefore, only one of these variables should be used in a model. Presume that, based on previous experience, you believe that **NGROUP** is sufficient for this type of modeling effort. Set the model role for **NEIGHBORHOOD** to rejected.
- h. The variables **LCDATE** and **LTIME** essentially measure the same thing. Set the model role for **LCDATE** to rejected, retaining the variable **LTIME** as an input variable.
- i. The variable **ORGANICS** contains information that would not be known at the time you are developing a model to predict the purchase of organic products. Set the model role for **ORGANICS** to rejected.
- j. Add the **ADMT.ORGANICS** data source to the diagram workspace.
- k. Add a Data Partition node to the diagram and connect it to the Data Source node. Assign 70% of the data for training and 30% for validation.

**2. Predictive Modeling Using Decision Trees**

- a.** Return to the Organics diagram in the Exercise project. Add a Decision Tree node to the workspace and connect it to the Data Partition node.
- b.** Run the diagram from the Decision Tree node with the default values for the decision tree.
- c.** Examine the tree results. How many leaves are in the tree that is selected based on the validation data set?
- d.** View the tree. Which variable was used for the first split? What were the competing splits for this first split?
- e.** Add a second Decision Tree node to the diagram and connect it to the Data Partition node.
- f.** In the Properties Panel of the new Decision Tree node, change the maximum number of branches from a node to 3 to allow for 3-way splits.
- g.** Run the diagram from the new Decision Tree node and examine the tree results. How many leaves are in the tree that is selected based on the validation data set?
- h.** Which variables were important in growing this tree?
- i.** View the tree. Which variable was used for the first split?
- j.** Close the tree results and add a Model Comparison node to the diagram. Connect both Decision Tree nodes to the Model Comparison node.
- k.** Using the Model Comparison node, which of the decision tree models appears to be better?

## 2.6 Solutions to Exercises

### 1. Initial Data Exploration

a. To set up a new project for this exercise:

- 1) Select **File** ⇒ **New** ⇒ **Project...**.
- 2) Type the name of the project (for example, **Exercise**).
- 3) Type in the location of the project folder, for example,  
**C:\Workshop\winsas\AMDT5\Exercises**.
- 4) Select the Start-Up Code tab.
- 5) Type in the appropriate code to define the library for the course data. For example:

```
libname ADMT 'C:\workshop\winsas\AMDT5' ;
```

6) Select **OK**.

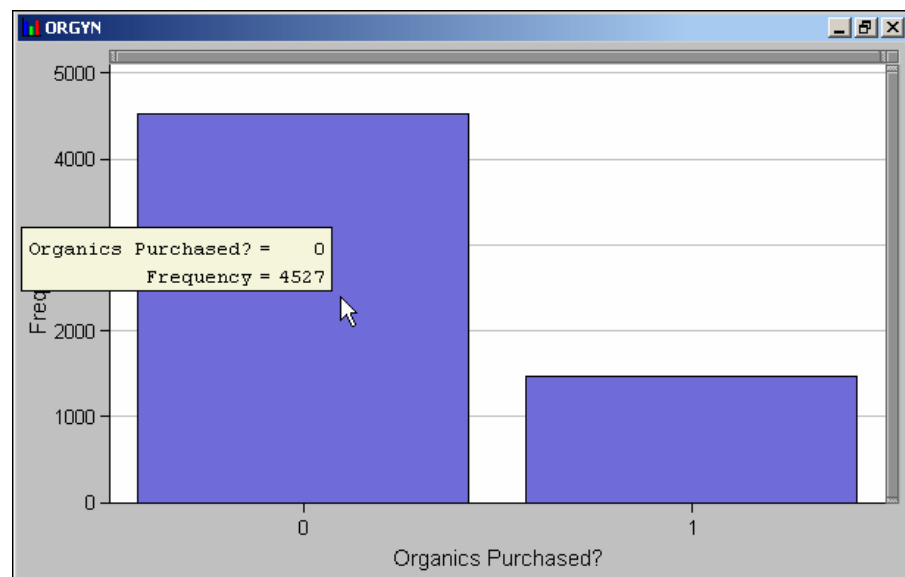
b. To create the diagram Organics:

- 1) Select **File** ⇒ **New** ⇒ **Diagram**.
- 2) Type in the diagram name, for example **Organics**.
- 3) Select **OK**.

c. To define a Data Source for the project:

- 1) Right-click on **Data Sources** in the Project Panel and select **Create Data Source**.
- 2) In the Data Source Wizard – Metadata Source window, be sure **SAS Table** is selected as the source and select **Next>**.
- 3) To choose the desired data table select **Browse...**.
- 4) Double-click on the **ADMT** library to see the data tables in the library.
- 5) Select the **ORGANICS** data set, and then select **OK**.
- 6) Select **Next>**. Observe that this data table has 22,223 observations (rows) and 18 variables (columns).
- 7) After examining the data table properties select **Next>**.
- 8) Select **Advanced** to use the Advanced advisor.
- 9) Select **Next>**.

- d. To set the model role for the target variable and examine the distribution of the variable:
- 1) Position the tip of your cursor over the row for **ORGYN** in the Role column.
  - 2) Click and select **Target** from the drop down menu.
  - 3) Highlight the row for the variable **ORGYN**.
  - 4) Select **Explore...**.
  - 5) Click on the **Value** column of the Sample Method row and select **Random** from the drop down menu.
  - 6) Select **Apply** to apply this change in the sample to the bar chart for the variable **ORGYN**.



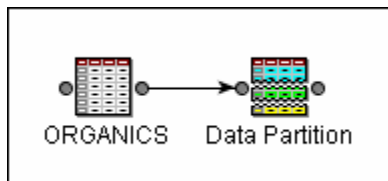
If you place your cursor over a bar in the chart, then the specific frequency of the value represented by that bar is displayed.

Close the Explore window when you are finished inspecting the plot. You can evaluate the distribution of other variables as desired.

- e. When you consider the other variables in the data set and examine their distributions, you might have the following reservations about them:
- The variable **GENDER** has the three values F, M, and U, plus missing values. U is essentially the same as missing, so it might make sense to replace the missing values for this variable with U.
  - The variables **AGE**, **AGEGRP1**, and **AGEGRP2** are all different measurements for the same information. Probably only one of these variables should be used for modeling.



- The variables **NGROUP** contains collapsed levels of the variable **NEIGHBORHOOD**. Probably only one of these variables should be used for modeling.
  - The variable **ORGANICS** contains information that would not be known at the time you are developing a model to predict the purchase of organic products. To avoid temporal infidelity, this variable should not be used as an input variable in any model to predict the probability of purchasing organic products.
  - The variables **AFFL**, **BILL**, and **LTIME** are all skewed to the right and might need to be transformed for models other than decision trees.
- f. To change the model roles for **AGEGRP1** and **AGEGRP2**:
- 1) Control-click to select the rows for both variables.
  - 2) Click in the Role column for one of the variables and select **Rejected**.
- g. It is quite possible that the role for **NEIGHBORHOOD** is already set to rejected because it exceeds the default maximum number of levels for a class variable. If this variable is not already rejected, change the role for **NEIGHBORHOOD**, by clicking in the Role column in the row for **NEIGHBORHOOD** and selecting **Rejected**.
- h. The model role for **LCDATE** should already be set as rejected. If it is not, click in the Role column in the row for **LCDATE** and select **Rejected**.
- i. To change the model role for **ORGANICS**, click in the Role column in the row for **ORGANICS** and select **Rejected**.
- j. After you have made all of the necessary changes to the roles of the variables, complete the definition of the data source by selecting **Next>**. Then select **Next>** and **Finish**. To add the data source **ORGANICS** to the workspace, drag it from the Project Panel to the workspace.
- k. To add a Data Partition node to the diagram and assign data for training and validation:
- 1) Select the Sample tab in the toolbar and drag a Data Partition node to the workspace to the right of the **ORGANICS** Data Source node.
  - 2) Connect the Data Partition node to the **ORGANICS** node.

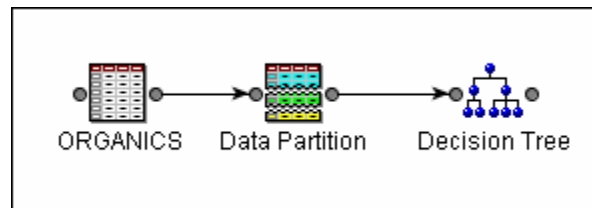


- 3) Select the Data Partition node and examine the Properties Panel.
- 4) In the Properties Panel, set the percentages for Training, Validation, and Test to 70, 30, and 0, respectively.

Data Set Percentage:	
Training	70.0
Validation	30.0
Test	0.0

## 2. Predictive Modeling Using Decision Trees

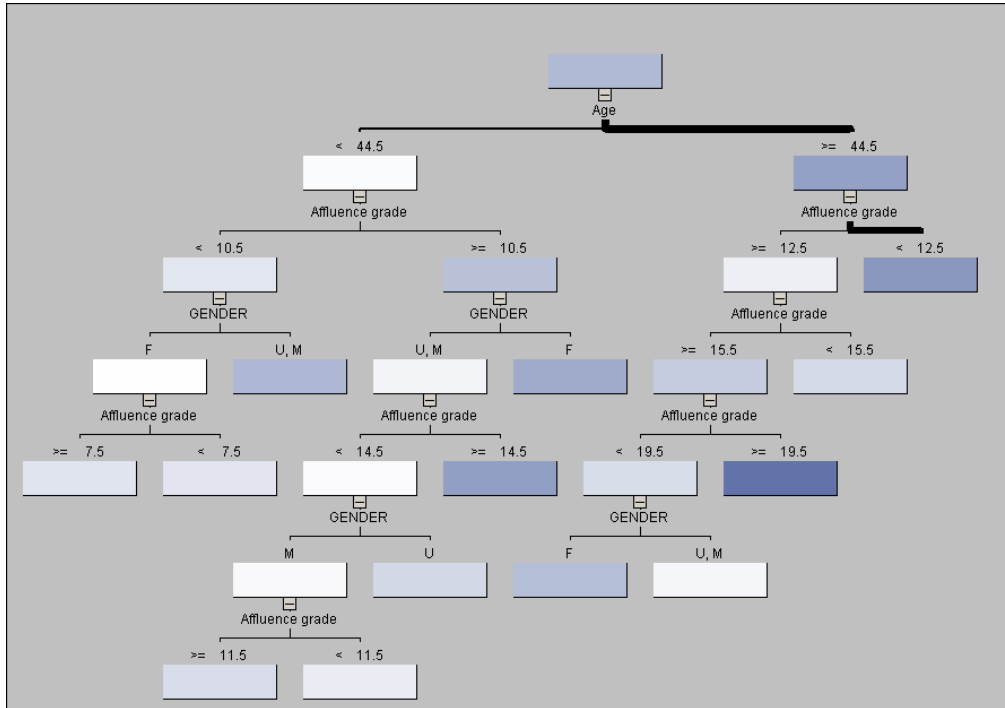
- a. Adding a Decision Tree node connected to the Data Partition node results in the diagram shown below.




- b. To run the diagram from the Decision Tree node, right-click on the node and select **Run**. Select **Yes** to confirm that you want to run the path. Select **OK** to acknowledge the completion of the run.
- c. To view the results, right-click on the Decision Tree node and select **Results...**. Examining the output reveals that the tree has 13 leaves.

Tree Leaf Report					
Node	Depth	Training Observations	% 1	Validation Observations	% V 1
6	2	10604	0.13	4530	0.13
11	3	1012	0.80	402	0.80
9	3	948	0.24	456	0.22
14	3	864	0.36	377	0.43
17	4	765	0.61	321	0.65
16	4	668	0.40	296	0.44
54	5	134	0.74	49	0.69
57	6	122	0.63	53	0.55
56	6	120	0.43	43	0.40
36	5	110	0.35	45	0.24
21	4	85	0.85	43	0.86
55	5	85	0.46	35	0.29
31	4	40	1.00	16	0.94

- d. To view the tree, maximize it in the results window. To get a better view of the entire tree, select **Edit** ⇒ **View** ⇒ **Fit to Page**.



The variable **AGE** was used for the first split.

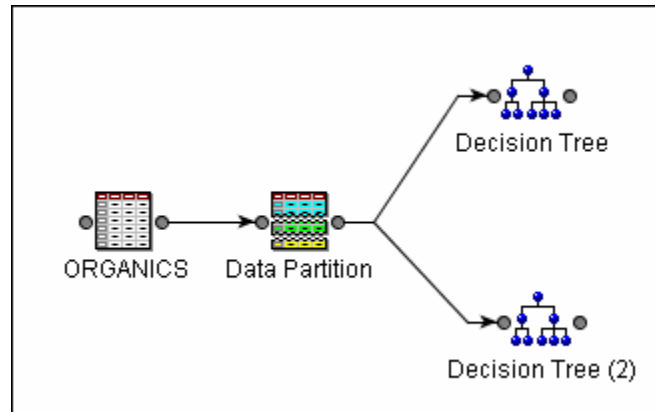
To examine the competing splits you must switch to interactive mode. Close the results window for the tree and be certain that the Decision Tree node is selected in the workspace. In the Interactive Training row of the Properties Panel, select . With the initial node selected, select **View** ⇒ **Competing Rules**.

Variable	-Log(p)	Branches
Age	156.722451	2
Affluence grade	78.116147	2
GENDER	69.271644	2
Total Amount Spent	15.667918	2
Customer LoyaltyStatus	9.941148	2

The other variables competing for this split were **AFFL**, **GENDER**, **BILL**, and **CLASS**.

Close the Competing Rules window and the Interactive Tree window to return to the diagram workspace.

- e. Adding a second Decision Tree node connected to the Data Partition node results in the diagram shown below:



- f. Select the new Decision Tree node to see the Properties Panel: Change the Maximum Branch value to **3** to allow for 3-way splits in addition to binary splits.

Property	Value
Node ID	Tree2
Imported Data	...
Variables	...
Interactive Training	...
Splitting Criterion	Default
Significance Level	0.2
Missing Values	Use in search
Leaf Size	5
Maximum Branch	3
Maximum Depth	6
Minimum Categorical S	5
Number of Rules	5
Number of Surrogate F0	
Split Size	

- g. To run the diagram from the Decision Tree node and examine the results:

- 1) Right-click on the node and select **Run**.
- 2) Select **Yes** to confirm that you want to run the path.
- 3) Select **OK** to acknowledge the completion of the run.

- 4) To view the results, right-click on the Decision Tree node and select **Results...**. Examining the output reveals that the tree has 18 leaves.

Tree Leaf Report					
Node	Depth	Training Observations	% 1	Validation Observations	% V 1
11	2	6039	0.08	2579	0.08
12	2	5429	0.21	2328	0.23
8	2	767	0.28	379	0.26
7	2	634	0.86	259	0.82
19	3	562	0.71	233	0.74
16	3	548	0.51	243	0.58
27	3	325	0.63	123	0.68
18	3	268	0.41	98	0.38
15	3	231	0.32	83	0.35
14	3	149	0.14	86	0.16
56	4	134	0.74	49	0.69
26	3	134	0.43	68	0.32
10	2	99	0.81	36	0.92
25	3	72	0.25	36	0.19
57	4	54	0.52	24	0.29
17	3	41	0.22	15	0.20
35	3	40	1.00	16	0.94
58	4	31	0.35	11	0.27

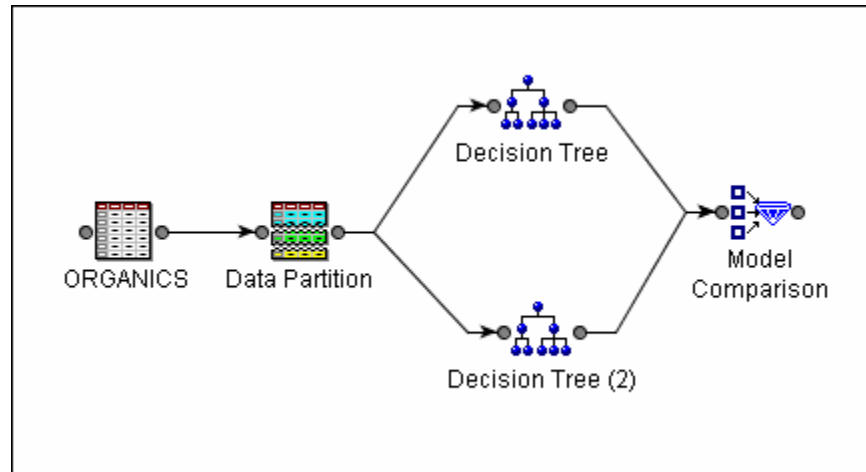
- h. To determine the relative importance of the variables in growing the tree, examine the Variable Importance table in the Output window.

Variable Importance					
Obs	NAME	LABEL	NRULES	IMPORTANCE	VIMPORTANCE
1	AGE	Age	1	1.00000	1.00000
2	AFFL	Affluence grade	4	0.75896	0.70027
3	GENDER		4	0.36206	0.44386

The variables that were important in growing the tree were **AGE**, **AFFL**, and **GENDER**.

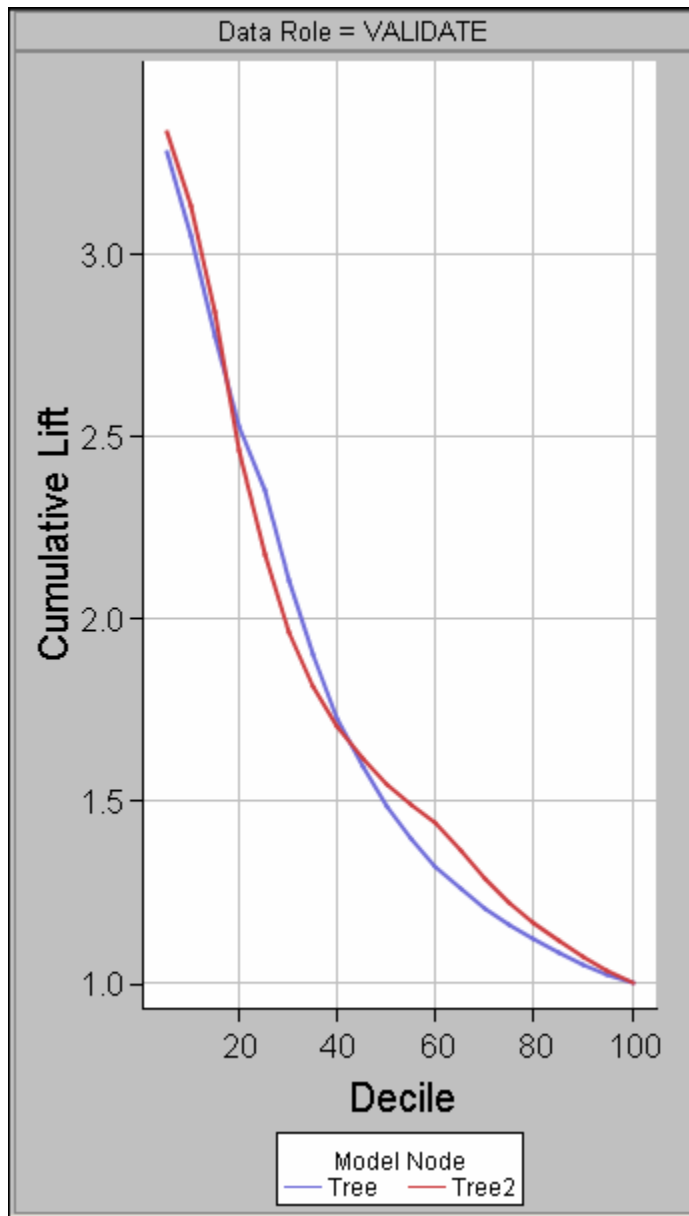
- i. Examining the tree in the output, the first split is a 3-way split on the variable **AGE**.

- j. Close the results to return to the diagram workspace. Add a Model Comparison node to the workspace.



- k. To compare the two decision tree models, use the Model Comparison node.
- 1) To run the diagram from the Model Comparison node, right-click on the node and select **Run**.
  - 2) Select **Yes** to confirm that you want to run the path. Select **OK** to acknowledge that the run is completed.
  - 3) To view the results, right-click on the Model Comparison node and select **Results...**.

4) Examine the Score Rankings graph for the validation data set.



The two models appear to be relatively similar.





# Chapter 3 Predictive Modeling Using Regression

<b>3.1</b>	<b>Introduction to Regression.....</b>	<b>3-3</b>
<b>3.2</b>	<b>Regression in SAS Enterprise Miner .....</b>	<b>3-8</b>
<b>3.3</b>	<b>Exercises.....</b>	<b>3-37</b>
<b>3.4</b>	<b>Solutions to Exercises .....</b>	<b>3-39</b>



## 3.1 Introduction to Regression

### Objectives

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

### Linear versus Logistic Regression

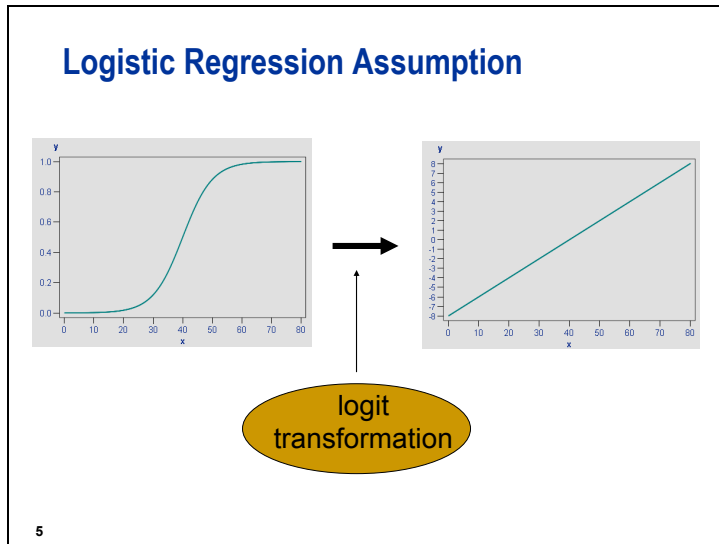
Linear Regression	Logistic Regression
Target is an interval variable.	Target is a discrete (binary or ordinal) variable.
Input variables have any measurement level.	Input variables have any measurement level.
Predicted values are the mean of the target variable at the given values of the input variables.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

4

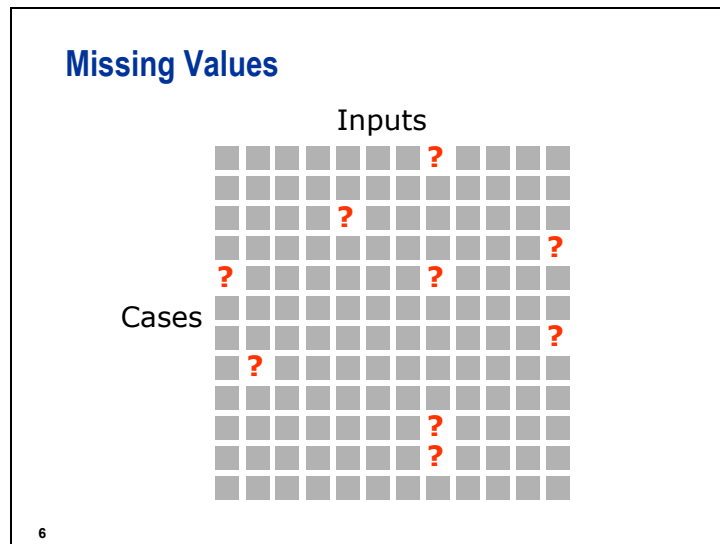
The Regression node in SAS Enterprise Miner does either linear or logistic regression depending upon the measurement level of the target variable.

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

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



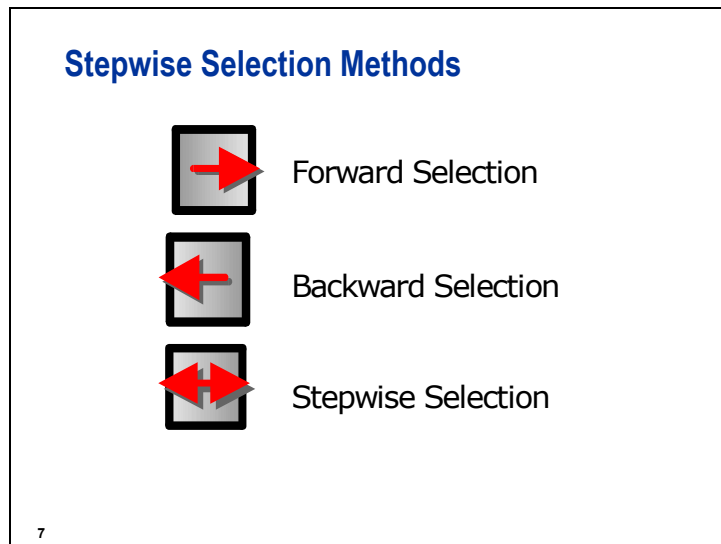
Recall that one assumption of logistic regression is that the logit transformation of the probabilities of the target variable results in a linear relationship with the input variables.



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

Other reasons for imputing missing values include the following:

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



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

Forward	selects the best one-variable model first. Then it selects the best two variables among those that contain the first selected variable. This process continues until it reaches the point where no additional variables have a $p$ -value less than the specified entry $p$ -value.
Backward	starts with the full model. Next, the variable that is least significant, given the other variables, is removed from the model. This process continues until all of the remaining variables have a $p$ -value less than the specified stay $p$ -value.
Stepwise	is a modification of the forward selection method. The difference is that variables already in the model do not necessarily stay there. After each variable is entered into the model, this method looks at all the variables already included in the model and deletes any variable that is not significant at the specified level. The process ends when none of the variables outside the model has a $p$ -value less than the specified entry value and every variable in the model is significant at the specified stay value.



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

## 3.2 Regression in SAS Enterprise Miner

### Objectives

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



### The Scenario

A nonprofit organization wants to send greeting cards to lapsed donors to encourage them to make a new donation. The organization wants to build a model to predict those most likely to donate. The types of information available include

- personal information such as age, gender, and income
- past donation information such as average gift and time since first donation
- census tract information based on the donor's address such as the percent of households with at least one member that works for the federal, state, or local government.

10

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

Name	Model Role	Measurement Level	Description
AGE	Input	Interval	Donor's age
AVGGIFT	Input	Interval	Donor's average gift
CARDGIFT	Input	Interval	Donor's gifts to card promotions
CARDPROM	Input	Interval	Number of card promotions
FEDGOV	Input	Interval	% of households with members working in federal government
FIRSTT	Input	Interval	Elapsed time since first donation
GENDER	Input	Binary	F=female, M=Male
HOMEOWNR	Input	Binary	H=homeowner, U=unknown
IDCODE	ID	Nominal	ID code, unique for each donor
INCOME	Input	Ordinal	Income level (integer values 0-9)
LASTT	Input	Interval	Elapsed time since last donation
LOCALGOV	Input	Interval	% of households with members working in local government

MALEMILI	Input	Interval	% of households with males active in the military
MALEVET	Input	Interval	% of households with male veterans
NUMPROM	Input	Interval	Total number of promotions
PCOWNER	Input	Binary	Y=donor owns computer (missing otherwise)
PETS	Input	Binary	Y=donor owns pets (missing otherwise)
STATEGOV	Input	Interval	% of households with members working in state government
TARGETB	Target	Binary	1=donor to campaign, 0=did not contribute
TARGETD	Target	Interval	Dollar amount of contribution to campaign
TIMELAG	Input	Interval	Time between first and second donation



The variable **TargetD** is not considered in this chapter, so its model role will be set to **Rejected**.

A card promotion is one where the charitable organization sends potential donors an assortment of greeting cards and requests a donation for them.

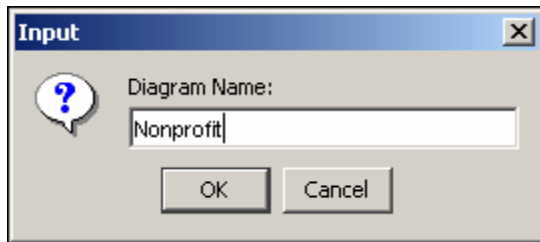


## Imputation, Transformation, and Regression

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

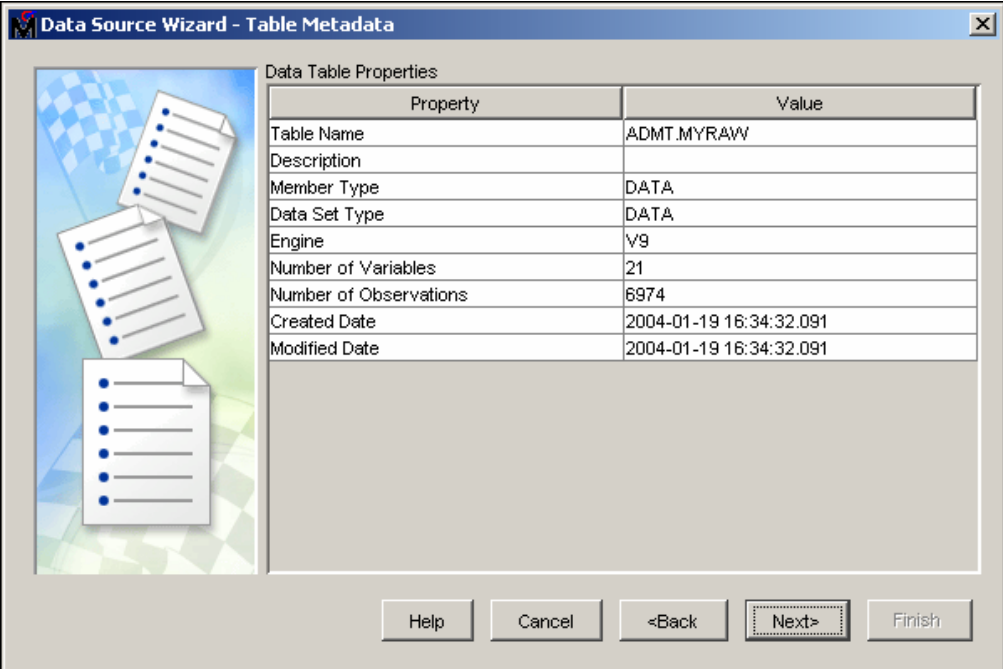
### Defining a New Data Source and Building the Initial Flow

1. Open a new diagram by selecting **File** ⇒ **New** ⇒ **Diagram...**.



2. Name the new diagram **Nonprofit** and select **OK**.
3. Define a new data source for the project by right-clicking on **Data Sources** in the Project Panel and selecting **Create Data Source**.
4. In the Data Source Wizard – Metadata Source window, be sure **SAS Table** is selected as the source and select **Next>**.
5. To choose the desired data table, select **Browse...**.
6. Double-click on the **ADMT** library to see the data tables in the library.
7. Select the **MYRAW** data set, and then select **OK**.

8. Select **Next>**.



The image shows a screenshot of the 'Data Source Wizard - Table Metadata' dialog box. The dialog has a title bar with the text 'Data Source Wizard - Table Metadata' and a close button. On the left side, there is a graphic of three overlapping document icons. The main area is titled 'Data Table Properties' and contains a table with two columns: 'Property' and 'Value'. The table lists the following properties and values:

Property	Value
Table Name	ADMT.MYRAW
Description	
Member Type	DATA
Data Set Type	DATA
Engine	V9
Number of Variables	21
Number of Observations	6974
Created Date	2004-01-19 16:34:32.091
Modified Date	2004-01-19 16:34:32.091

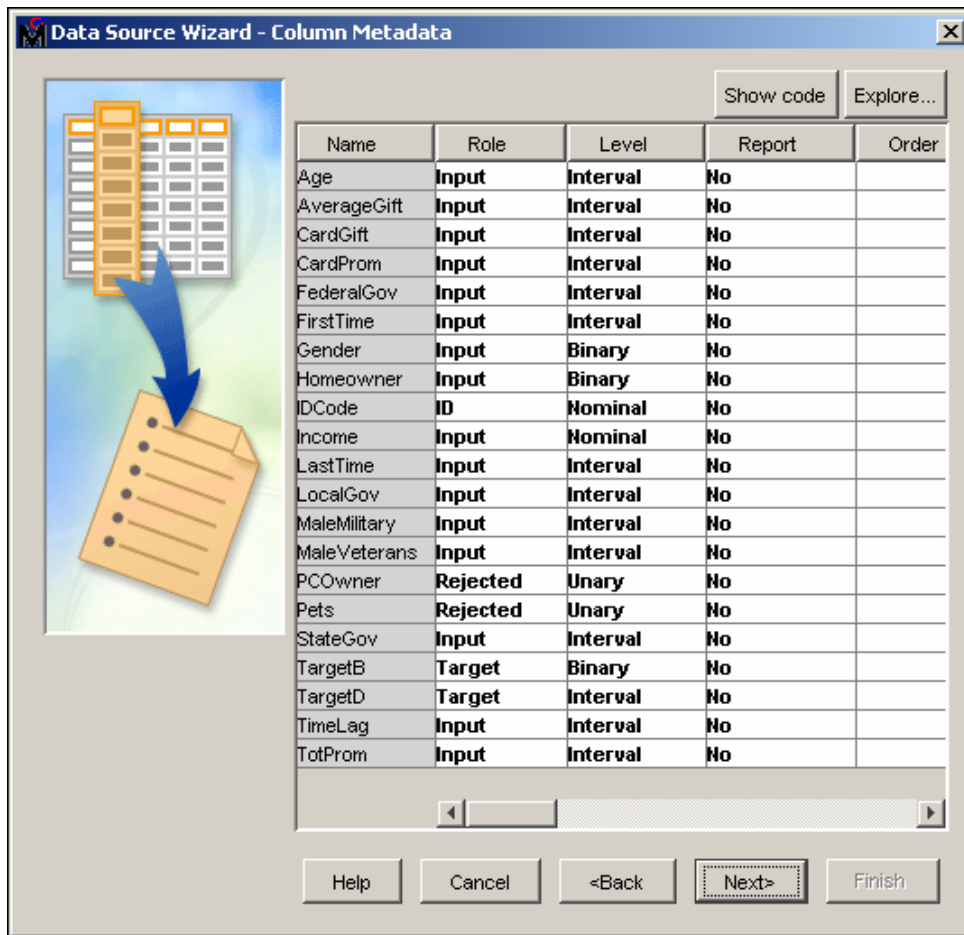
Below the table, there is a large empty rectangular area. At the bottom of the dialog, there are five buttons: 'Help', 'Cancel', '<Back', 'Next>' (which is highlighted with a dashed border), and 'Finish'.

Observe that this data set has 6,974 observations (rows) and 21 variables (columns).

9. Select **Next>**.

10. Select **Advanced** to use the Advanced advisor to initially determine the roles and levels for the variables.

11. Select **Next>**.



The first several variables (**Age** through **FirstTime**) have the measurement level interval because they are numeric in the data set and have more than 20 distinct levels. The model role for all interval variables is set to input by default.

The variables **Gender** and **Homeowner** have the measurement level binary because they have only two different nonmissing levels. The model role for all binary variables is set to input by default.

The variable **IDCode** is listed as a nominal variable because it is a character variable with more than two nonmissing levels. Furthermore, because it is nominal and the number of distinct values is greater than 20, the **IDCode** variable has the model role ID. If **IDCode** had been stored as a number, it would have been assigned an interval measurement level and an input model role.

The variable **Income** is listed as a nominal variable because it is a numeric variable with more than two but no more than ten distinct levels. All nominal variables are set to have the input model role. You will change the level to ordinal.

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

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

### Identifying Target Variables

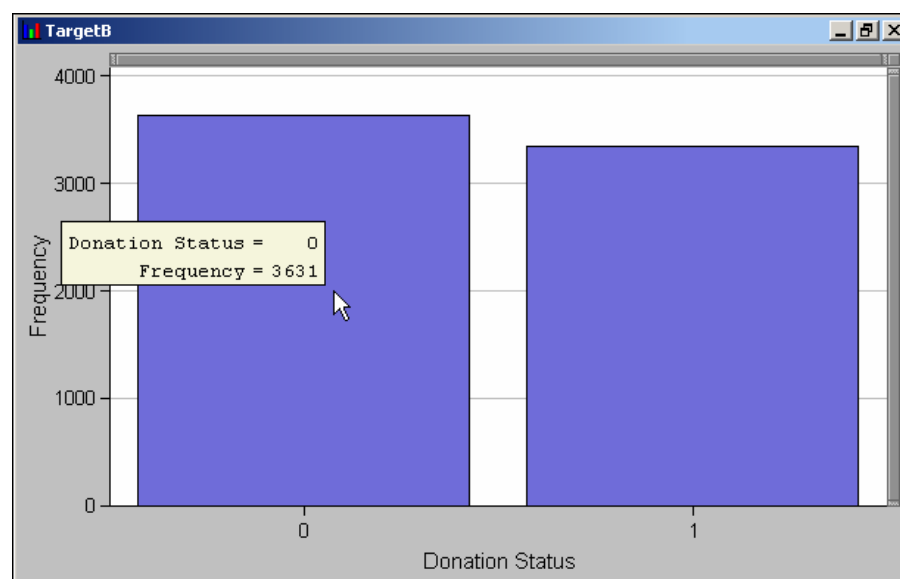
Note that the variables **TargetB** and **TargetD** are the response variables for this analysis. **TargetB** is binary even though it is a numeric variable since there are only two non-missing levels. **TargetD** has the interval measurement level. Both variables are set to have the input model role (just like any other binary or interval variable). This analysis will focus on **TargetB**, so you need to change the model role for **TargetB** to target and the model role for **TargetD** to rejected.

1. Position the tip of your cursor over the row for **TargetD** in the Role column.
2. Click and select **Rejected** from the drop-down menu.

### Inspecting Distributions

You can inspect the distribution of values for each of the variables. To view the distribution of **TargetB**:

1. Select the row for **TargetB**.
2. Select **Explore...**



Investigate the distribution of the unary variables, **PCOwner** and **Pets**. What percentage of the observations have pets? What percentage of the observations own personal computers?

Evaluate the distribution of other variables as desired.

### Modifying Variable Information

Earlier you changed the model role for **TargetB** to target. Now modify the role and level for **PCOwner** and **Pets**. Also, change the level for **Income** to ordinal.

1. Control-click to select the rows for **PCOwner** and **Pets**.
2. Click in the Role column for one of these variables and select **Input** from the drop-down menu.
3. Click in the Level column for one of these variables and select **Binary** from the drop-down menu.
4. Click in the Level column of the row for the variable **Income** and select **Ordinal** from the drop-down menu.
5. All of the necessary changes have been made. Select **Next>** to continue defining the data source.

### Understanding Decision Processing for a Binary Target

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

In this analysis, you are analyzing a binary variable. The accuracy criterion would choose the model that best predicts whether someone actually responded; however, there are different profits and losses associated with different types of errors. Specifically, it costs less than a dollar to send someone a mailing, but you receive a median of \$13.00 from those that respond. Therefore, to send a mailing to someone that would not respond costs less than a dollar, but failing to mail to someone that would have responded costs over \$12.00 in lost revenue.



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

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

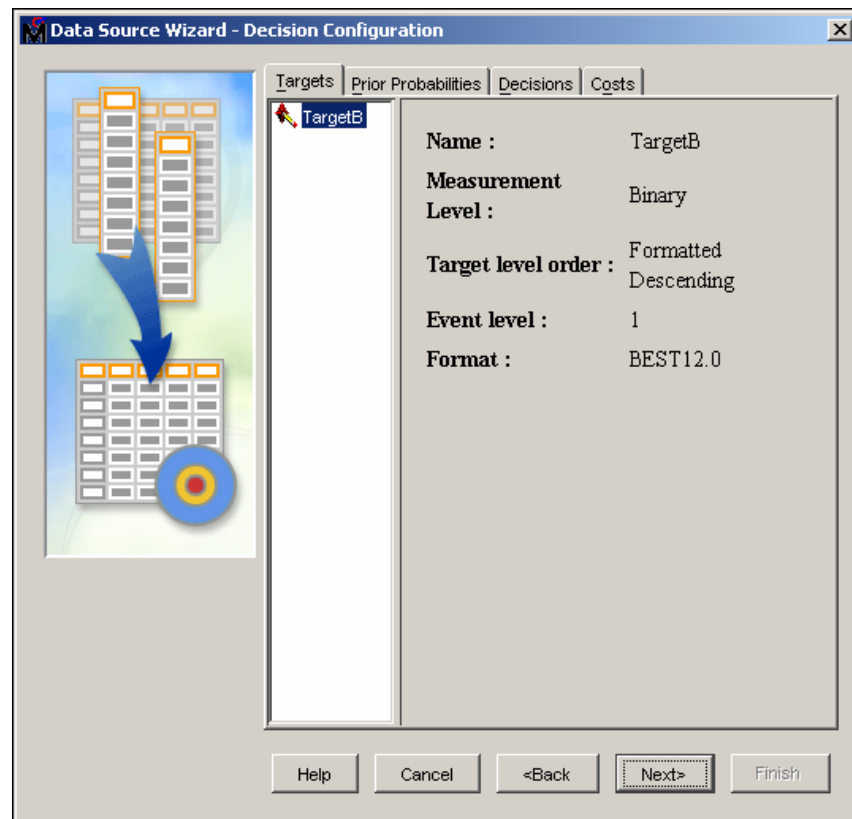


In the case of rare target events, it is not uncommon to oversample. This is because you tend to get better models when they are built on a data set that is more balanced with respect to the levels of the target variable.

## Using a Target Profile

When building predictive models, the choice of the "best" model depends on the criteria you use to compare competing models. SAS Enterprise Miner allows you to specify information about the target that can be used to compare competing models. To generate a target profile for a variable, you must have already set the model role for the variable to target. This analysis focuses on the variable **TargetB**. To set up the target profile for this **TargetB**, proceed as follows:

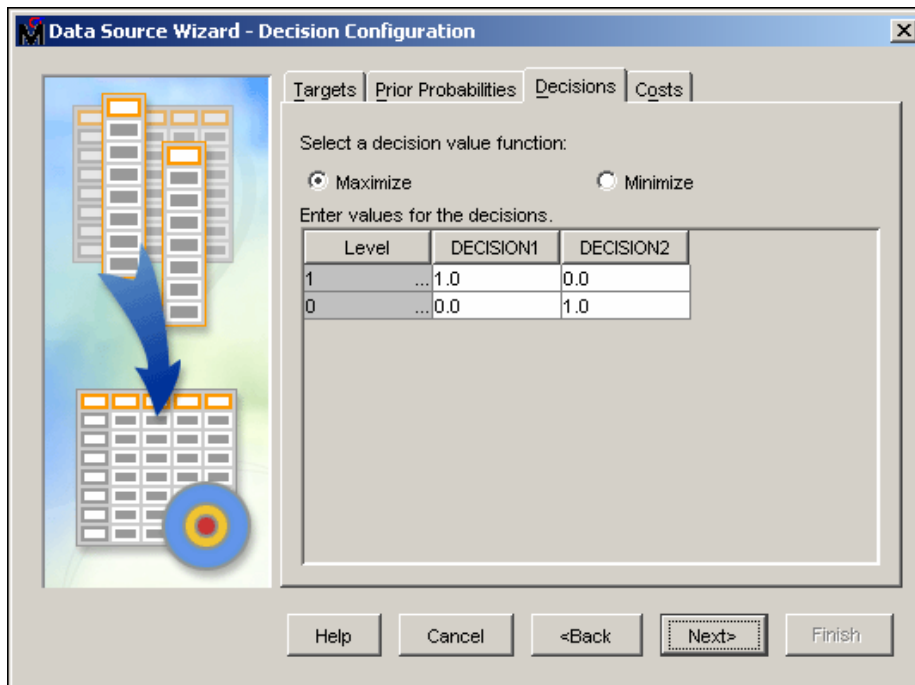
1. In the Data Source Wizard – Decision Processing window, select **Yes** to build models based on the values of decisions.
2. Select **Next>** to move to the Decision Configuration window.



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



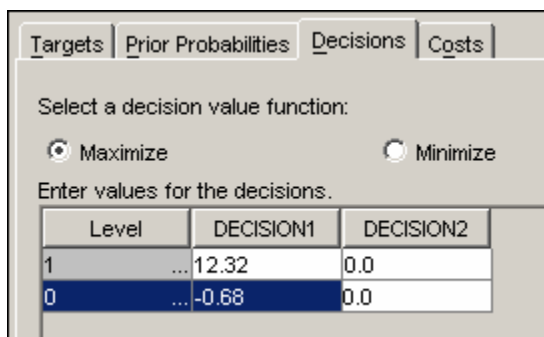
- To incorporate profit and cost information into this profile, select the **Decisions** tab.



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

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

- mailing to a responder is  $13.00 - 0.68 = 12.32$
  - mailing to a nonresponder is  $0.00 - 0.68 = -0.68$
- Enter the profits associated with the vector for action (DECISION1). Your matrix should appear as shown below. Do not forget to change the bottom-right cell of the matrix to 0.



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

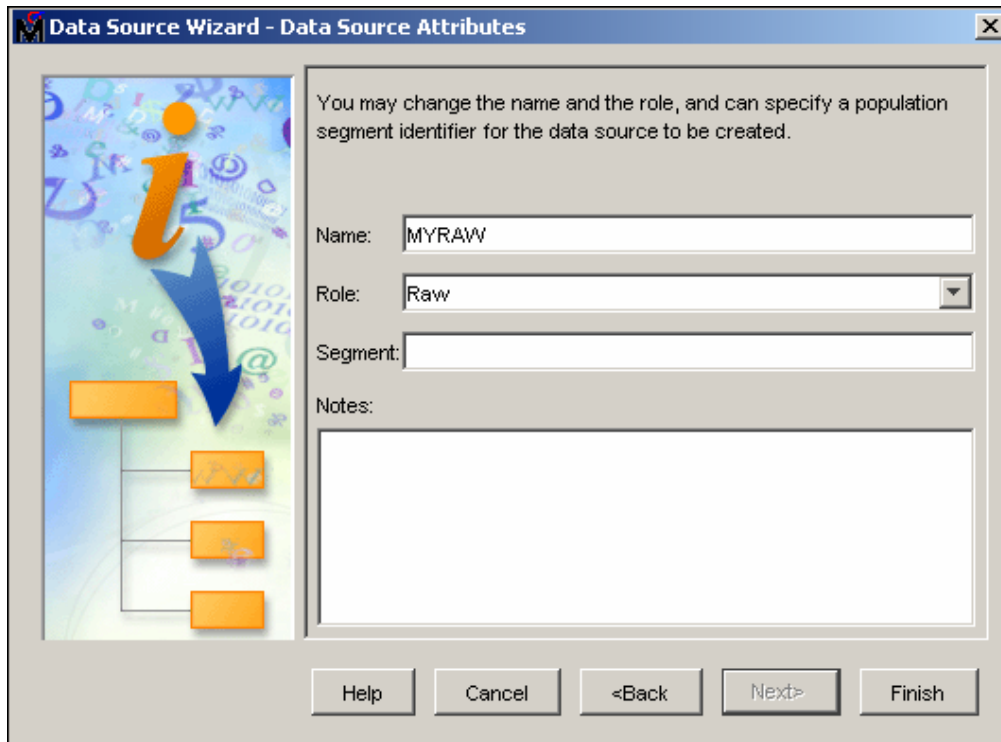
5. As discussed earlier, the proportions in the population are not represented in the sample. To adjust for this, select the **Prior Probabilities** tab.

Level	Count	Prior
1	3343	0.4794
0	3631	0.5206

6. To add a prior probabilities, select **Yes**. This adds the Adjusted Prior column.
7. Modify the Adjusted Prior column to reflect the true proportions in the population.

Level	Count	Prior	Adjusted Prior
1	3343	0.4794	0.05
0	3631	0.5206	0.95

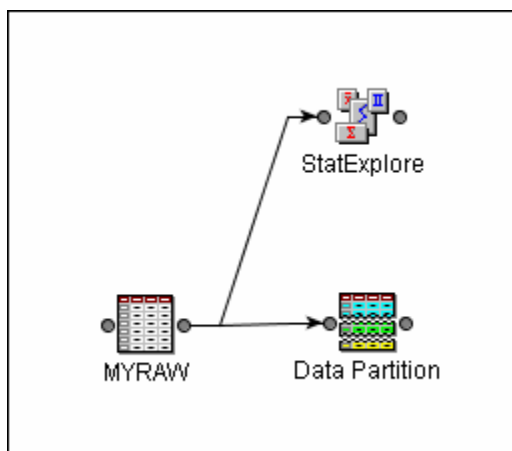
8. After all of the target variable information has been entered, select **Next>**.



9. If desired, add notes about the data set, and then select **Finish**. The **MYRAW** data set has been added as a data source for this project.

### Building the Initial Flow

1. Presuming the diagram Nonprofit in the project named My Project is open, add the **MYRAW** data source to the diagram workspace.
2. Also add a Data Partition node and a StatExplore node to the workspace and connect them to the data source as shown below.



### The Data Partition Node

1. Select the Data Partition node in the workspace.
2. You can specify the percentage of the data to allocate to training, validation, and testing data in the Property Panel. Enter **70** for training, **30** for validation, and **0** for test.

Data Set Percentage:	
Training	70.0
Validation	30.0
Test	0.0



If you allocate more or less than 100% of the data, then SAS Enterprise Miner will ignore the percentages you specify and use the default values.

### Preliminary Investigation

The StatExplore node can be used to examine the distributions of variables in the data set. In addition, the node can be used to select variables to be used in future analyses, such as predictive models. In this case, you would like to examine the distributions of the variables.

1. Select the StatExplore node in the workspace and examine the properties in the Property Panel.

Property	Value
Node ID	Stat
Imported Data	...
Variables	...
Chi-Square	Yes
Correlations	Yes
Use Segment Variable	No
Variable Selection	
Hide Rejected Variable	Yes
Number of Selected Variables	1000
Status	
Last Error	
Last Status	
Needs Updating	No
Needs to Run	Yes
Time of Last Run	
Run Duration	

2. Because you are interested in examining all variables, use the drop-down menu to change the Hide Rejected Variables field to **No**.
3. Right-click on the StatExplore node and select **Run**.
4. Select **Yes** when prompted to confirm that you want to run the path.
5. Select **OK** to confirm that the run is complete.

6. Right-click on the StatExplore node and select **Results...**

7. Examine the results in the Output window.

Class Variable Summary Statistics (maximum 500 observations printed)							
Variable	Role	Numcat	NMiss	Mode	Mode Pct	Mode2	Mode2Pct
Gender	INPUT	3	395	F	53.74	M	40.59
Homeowner	INPUT	3	1656	H	55.29		23.75
Income	INPUT	8	1576	.	22.60	5	16.89
PCOwner	INPUT	2	6196		88.84	Y	11.16
Pets	INPUT	2	5865		84.10	Y	15.90
TargetB	TARGET	2	0	0	52.06	1	47.94

The output includes information about the modes of each of the class variables in the data table. For example, you can see that the variable **PCOwner** has almost 90% missing values and the variable **Pets** has over 84% missing values. As discussed earlier, these missing values will be recoded as U. Also, notice that the variable **Homeowner** has almost 24% missing values. You might consider recoding this as U also.

You can also examine summary statistics such as the mean, minimum value, maximum value, and median of the interval variables in the data table.

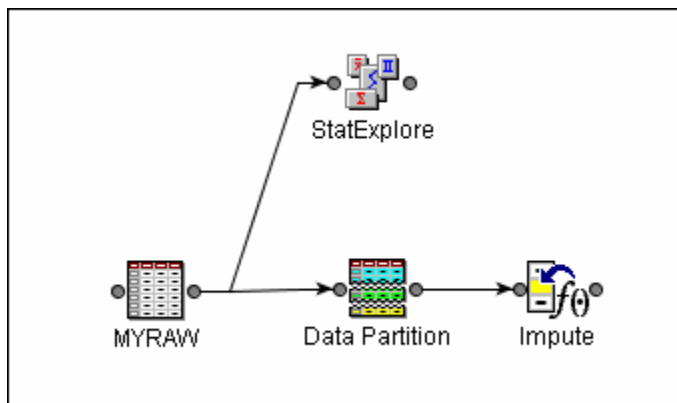
You can see the same information for each of these variables by level of the target variable, **TargetB**, as well.



You can view some of this same information in a table form by selecting **View** ⇒ **Summary Statistics** ⇒ **Class Variables** or **View** ⇒ **Summary Statistics** ⇒ **Interval Variables**.

## Understanding and Using Data Replacement

1. Add an Impute node to the diagram. Your new diagram should appear as follows:



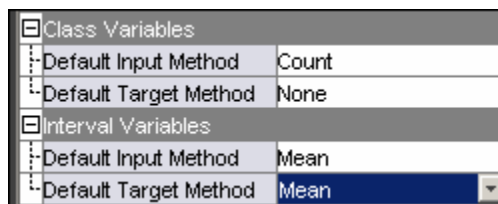
2. Select the Impute node in the workspace and examine the Property Panel.

3. In the Indicator Variables section of the Property Panel, use the drop-down menu to change the value of the Indicator Variable row to **Unique**.
4. Change the Indicator Variable Role to **Input**.



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

5. Examine the Interval Variables and Class Variables sections of the Property Panel.



This shows that the default imputation method for Interval Variables is the mean. By default, imputation for class variables is done using the most frequently occurring level (also referred to as the count or the mode). If the most commonly occurring value is missing, it uses the second most frequently occurring level in the sample.

Click in the cell showing the Default Input Method for interval variables. SAS Enterprise Miner provides the following methods for imputing missing values for interval variables:

- Mean – uses the arithmetic average. This is the default.
- Median – uses the 50<sup>th</sup> percentile.
- Midrange – uses the maximum plus the minimum divided by two.
- Distribution-based – calculates replacement values based on the random percentiles of the variable's distribution.
- Tree imputation – estimates replacement values with a decision tree using the remaining input and rejected variables that have a status of use as the predictors.
- Tree imputation with surrogates – is the same as above except that surrogate variables are used for splitting whenever a split variable has a missing value. This prevents forcing everyone with a missing value for a variable into the same node.
- Mid-Minimum spacing – uses the mid-minimum spacing statistic. To calculate this statistic, the data is trimmed using  $N$  percent of the data as specified in the Proportion for mid-minimum spacing entry field. By default, 90% of the data is used to trim the original data. In other words, 5% of the data is dropped from each end of the distribution. The mid-range is calculated from this trimmed data.

- Tukey's biweight, Huber's, and Andrew's wave – are robust M-estimators of location. This class of estimators minimize functions of the deviations of the observations from the estimate that are more general than the sum of squared deviations or the sum of absolute deviations. M-estimators generalize the idea of the maximum-likelihood estimator of the location parameter in a specified distribution.
- Default constant value – enables you to set a default value to be imputed for some or all variables.
- None – turns off the imputation for interval variables.

Click in the cell showing the Default Input Method for class variables. SAS Enterprise Miner provides several of the same methods for imputing missing values for class variables including distribution-based, tree imputation, tree imputation with surrogates, default constant value, and none.

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

Class Variables	
Default Input Method	Tree
Default Target Method	None
Interval Variables	
Default Input Method	Tree
Default Target Method	Mean


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

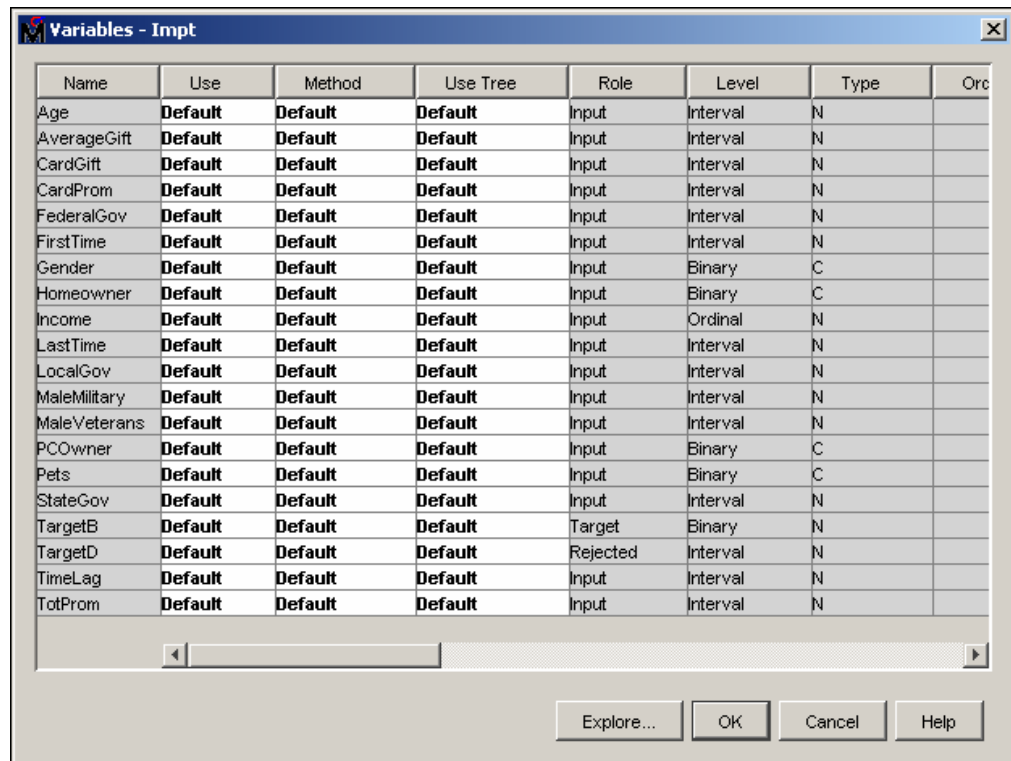
7. Note the Default Constant Value section of the Property Panel. Enter **U** in the field for character values.
8. To examine other options available in the Impute node select **View** ⇒ **Property Sheet** ⇒ **Advanced**.

The Non Missing Variables property is set to No by default. This indicates that imputation will not be done for variables that have no missing values in the input data set. In addition, missing value indicator variables will not be created for those variables that are not imputed. While this prevents the creation of unary missing value indicator variables, it can also create a scoring issue if the data to be scored contains missing values where the training data has none. This can degrade model performance if this happens often.

Also note that when you choose tree as an imputation method you have options to control the growth of the trees.

At this point, you have stipulated the imputation method for interval variables as a tree. Suppose you want to change the imputation methods for **Age** and **CardProm**. Impute the missing values for **Age** with the mean and impute the missing values for **CardProm** with the constant value 20.

9. Select  in the Variables row of the Property Panel.



10. To specify a different imputation method for **Age**, click on the row for **Age** in the Method column and select Mean from the drop-down menu.

11. To specify the imputation method for **CardProm**, click on the row for **CardProm** in the Method column and select Constant from the drop-down menu.

Recall that the variables **Homeowner**, **PCOwner**, and **Pets** should have the missing values set to **U**.

12. Control-click to select the rows for **Homeowner**, **PCOwner**, and **Pets**.

13. Click on one of the selected rows in the Method column and select Constant from the drop-down menu.

14. Select OK to confirm the changes and close the Variables window.

15. To complete the settings for variable imputation enter **20** for the Default Number Value in the Property Panel.



16. Run the diagram from the Impute node and view the results.

Imputation Summary							
Variable	Imputation Method	Imputed Variable	Indicator Variable	Role	LEVEL	Type	Variable Label
Age	MEAN	IMP_Age	M_Age	INPUT	INTERVAL	N	Donor's Age
Gender	TREE	IMP_Gender	M_Gender	INPUT	BINARY	C	.
Homeowner	CONSTANT	IMP_Homeowner	M_Homeowner	INPUT	BINARY	C	.
Income	TREE	IMP_Income	M_Income	INPUT	ORDINAL	N	Income Level
PCOwner	CONSTANT	IMP_PCOwner	M_PCOwner	INPUT	BINARY	C	Personal Compute...
Pets	CONSTANT	IMP_Pets	M_Pets	INPUT	BINARY	C	Pet(s) in Household
TimeLag	TREE	IMP_TimeLag	M_TimeLag	INPUT	INTERVAL	N	Time between Don...

The Imputation Summary shows that there were seven variables with missing values that were imputed.

17. Select **View** ⇒ **Output Data Sets** ⇒ **Train**.

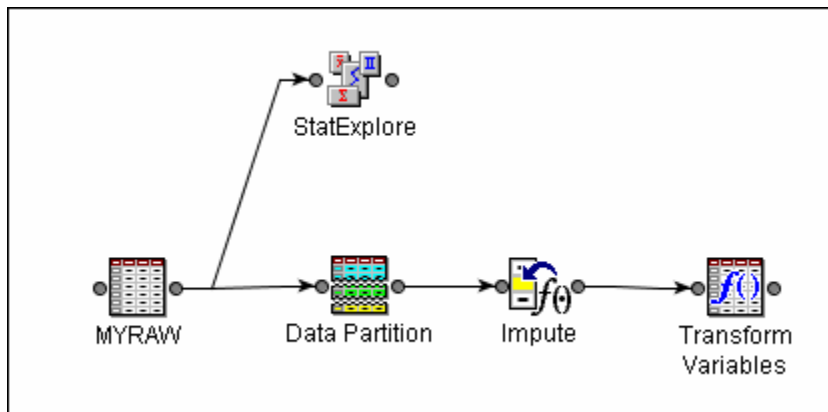
18. Scroll to the right to view the new columns that were created in the data table.

EMWS4.Impt_TRAIN								
tion In...	Imputed: Per...	Imputation In...	Imputed: Pet...	Imputation In...	Imputed: Do...	Imputation In...	Imputed Ge...	V
1U		0Y		0	64		0M	.
1U		1U		0	78		0M	.
1U		1U		1	61.797619048		0F	.
1U		1U		1	61.797619048		0F	.
1U		0Y		0	54		0F	.
1U		1U		0	41		0M	.
1U		1U		0	42		0F	.
1U		1U		0	54		0M	.

### Performing Variable Transformations

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


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

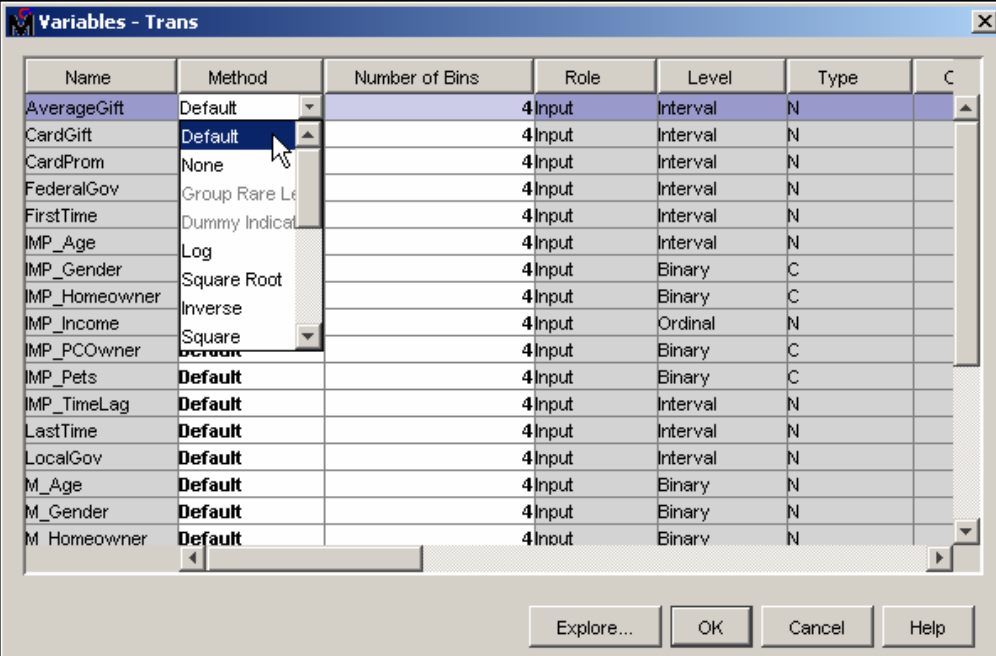


2. Select the Transform Variables node and examine the Property Panel.

Property	Value
Node ID	Trans
Imported Data	...
Variables	...
Default Methods	
Interval Inputs	None
Interval Targets	None
Class Inputs	None
Class Targets	None
Interactions Editor	...
Summary Variables	Transformed and New

Note that by default no transformations are done.

3. Select  in the Variables row of the Property Panel.
4. Click in the Method column of any interval variable row in the Variables window.



Name	Method	Number of Bins	Role	Level	Type	C
AverageGift	Default	4	Input	Interval	N	
CardGift	Default	4	Input	Interval	N	
CardProm	None	4	Input	Interval	N	
FederalGov	Group Rare Le	4	Input	Interval	N	
FirstTime	Dummy Indica	4	Input	Interval	N	
IMP_Age	Log	4	Input	Interval	N	
IMP_Gender	Square Root	4	Input	Binary	C	
IMP_Homeowner	Inverse	4	Input	Binary	C	
IMP_Income	Square	4	Input	Ordinal	N	
IMP_PCOwner	Default	4	Input	Binary	C	
IMP_Pets	Default	4	Input	Binary	C	
IMP_TimeLag	Default	4	Input	Interval	N	
LastTime	Default	4	Input	Interval	N	
LocalGov	Default	4	Input	Interval	N	
M_Age	Default	4	Input	Binary	N	
M_Gender	Default	4	Input	Binary	N	
M_Homeowner	Default	4	Input	Binary	N	

Buttons: Explore... OK Cancel Help

The Transform Variables node enables you to rapidly transform interval-valued variables using standard transformations such as the log, square root, inverse, and square.

In addition to these standard transformations, you can collapse an interval variable into create a grouping variable that creates bins for the values of an interval variable. This can be done in several different ways.

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

Finally, best power transformations are also available for interval variables. These best power transformations are a subset of the general class of transformations that are known as Box-Cox transformations. The Transform Variables node supports the following best power transformations:

- **Maximize Normality** — This method chooses the transformation that yields sample quantiles that are closest to the theoretical quantiles of a normal distribution. This method requires an interval variable.
- **Maximize Correlation with Target** — This method chooses the transformation that has the best squared correlation with the target. This method requires an interval target.
- **Equalize Spread with Target Levels** — This method chooses the transformation that has the smallest variance of the variances between the target levels. This method requires a class target.
- **Optimal Maximum Equalize Spread with Target Level** — This method chooses the transformation that equalizes spread with target levels. This method requires a class target.

The Transform Variables node can also be used for class variables. The transformations available for class variables are to group rare levels or to create indicator variables.

You can examine the distributions of variables within the Transform Variables node by highlighting the row for the variable(s) of interest and selecting **Explore...**. The distributions of **CardGift**, **LocalGov**, **StateGov**, **FedGov**, and **TimeLag** are highly skewed to the right. A log transformation of these variables may provide more stable results.

1. To transform **CardGift**, click in the Method of the row for the variable **CardGift** and select **Log** from the drop-down menu.
2. You can repeat this process to specify the transformation for each of the remaining variables one at a time or you can do all of the variables together. To specify the transformation of the variables together Press and hold the Ctrl key on the keyboard.
3. While holding the Ctrl key, select each of the variables, **LocalGov**, **StateGov**, **FedGov**, and **IMP\_TimeLag**.
4. When all have been selected, release the Ctrl key.
5. Click in the Method column of one of the selected rows and select **Log**.
6. Run the diagram from the Transform Variables node and view the results.

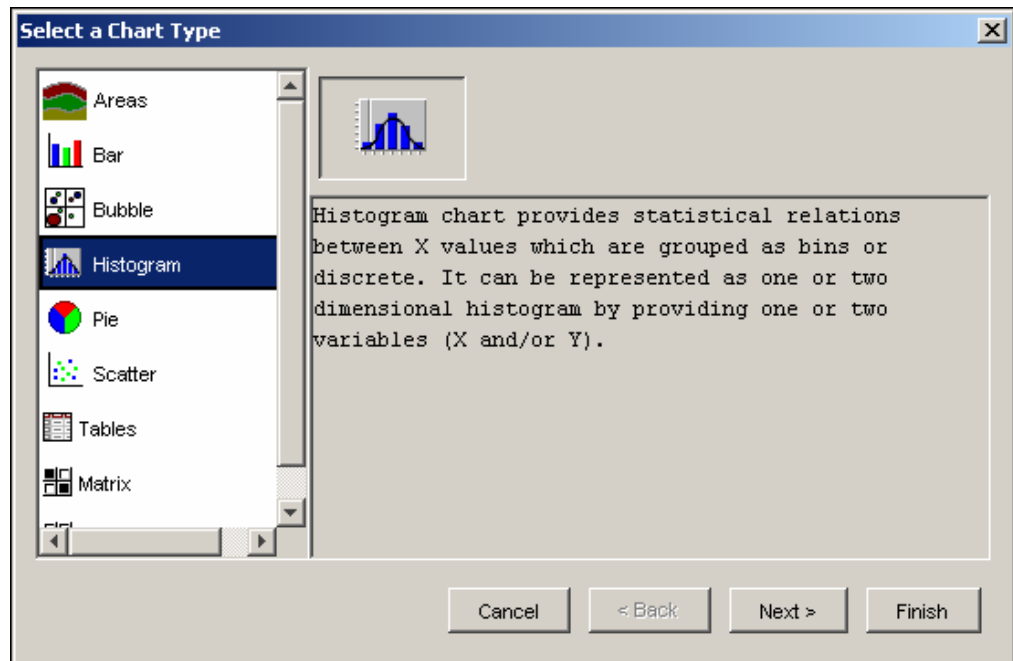
7. Maximize the Transformations window in the results and expand the Formula column.

Transformations								
Input Name	Output Name	Output Level	Power	Formula	LABEL	Output Type	Role	Input
CardGift	LOG_CardGift	INTERVAL		$\text{Olog}(\text{CardGift} + 1)$	Transformed: ...N		INPUT	INTERV
FederalGov	LOG_Federal...	INTERVAL		$\text{Olog}(\text{FederalGov} + 1)$	Transformed: ...N		INPUT	INTERV
IMP_TimeLag	LOG_IMP_Ti...	INTERVAL		$\text{Olog}(\text{IMP\_TimeLag} + 1)$	Transformed: ...N		INPUT	INTERV
LocalGov	LOG_LocalGov	INTERVAL		$\text{Olog}(\text{LocalGov} + 1)$	Transformed: ...N		INPUT	INTERV
StateGov	LOG_StateGov	INTERVAL		$\text{Olog}(\text{StateGov} + 1)$	Transformed: ...N		INPUT	INTERV

The formula shows that SAS Enterprise Miner has performed the log transformation after adding 1 to the value of each of these variables. Why has this occurred? As an example, **CardGift** has a minimum value of zero. The logarithm of zero is undefined, and the logarithm of something close to zero is extremely negative. SAS Enterprise Miner takes this information into account and actually uses the transformation  $\text{log}(\text{CardGift} + 1)$  to create a new variable with values greater than or equal to zero (because the  $\text{log}(1)=0$ ).

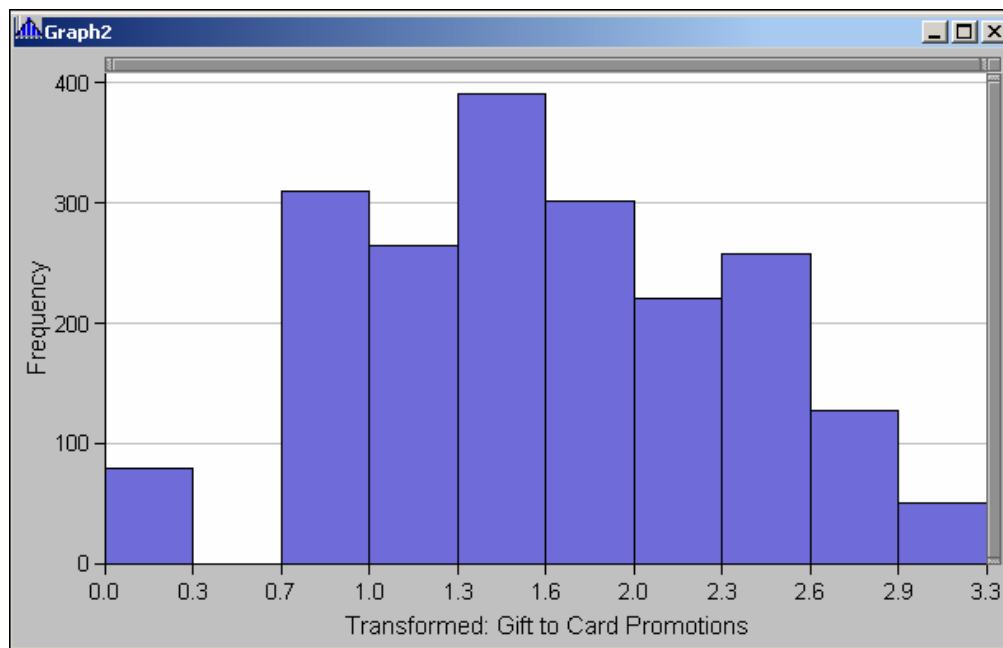
You can inspect the distribution of the transformed variables.

8. While examining the results, select **View** ⇒ **Output Data Sets** ⇒ **Train**.
9. In the Explore window, select **Actions** ⇒ **Plot**.



10. Select **Histogram** as the type of chart, then select **Next>**.
11. Change the Role of **LOG\_CARDGIFT** to **X**.

12. Select **Finish**.

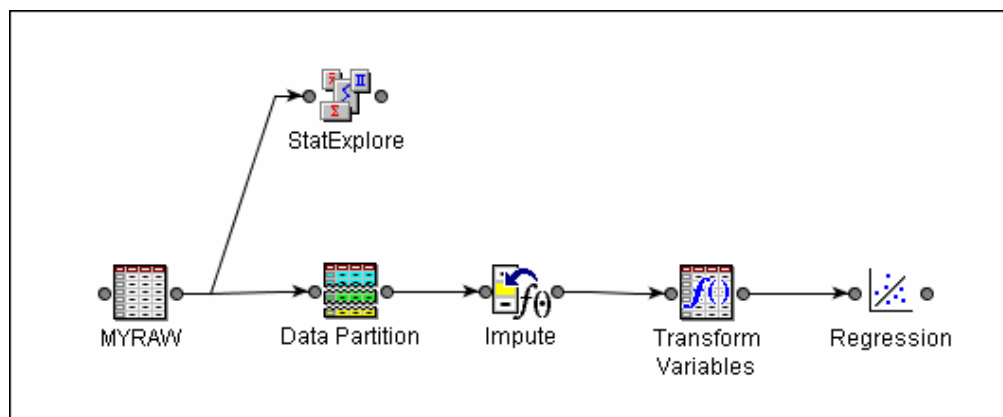


The variable is much less skewed than before.

It may be appropriate at times to keep the original variable and the created variable. Commonly this is not done when the original variable and the transformed variable have the same measurement level. The default behavior of SAS Enterprise Miner is to hide and reject the original variables. This behavior can be changed using the advanced features of the Property Panel.


### Fitting a Regression Model

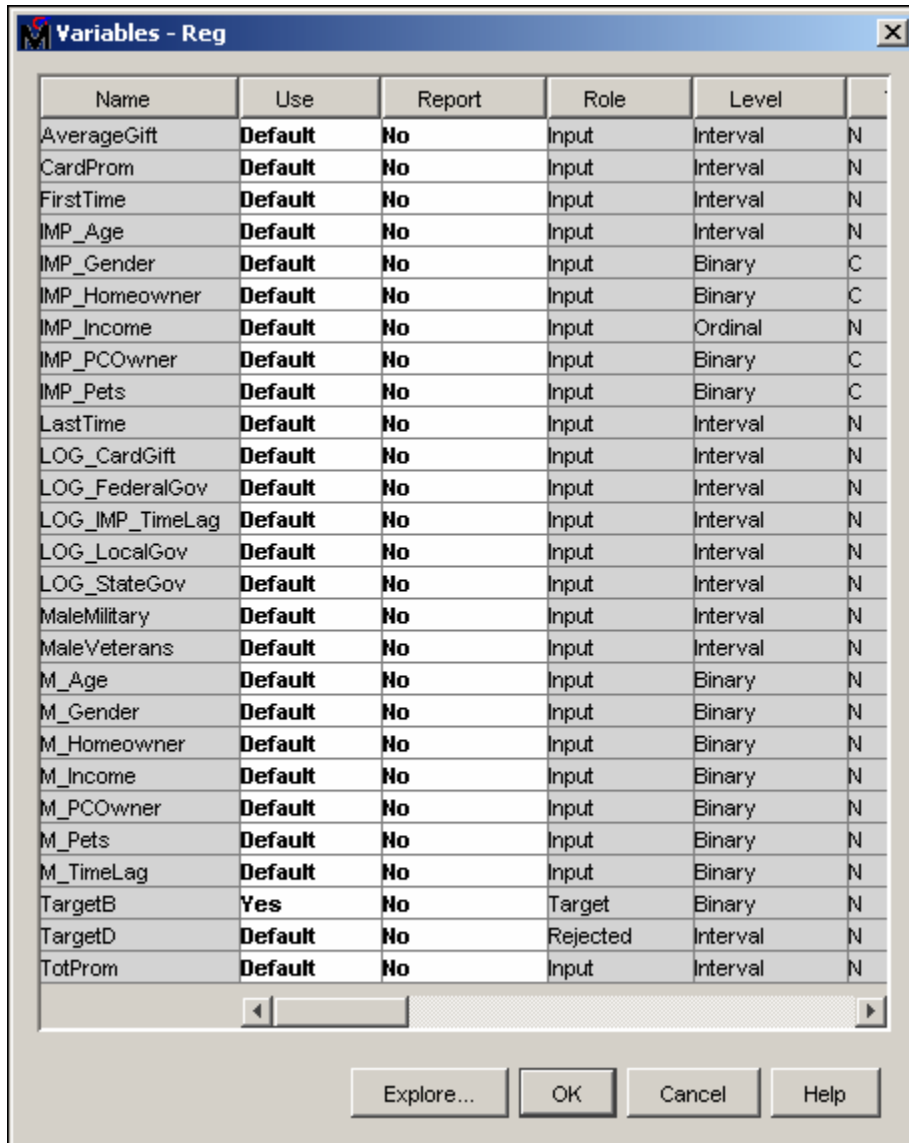
1. Connect a Regression node to the diagram as shown.



2. Select the Regression node in the diagram and examine the Property Panel.

Property	Value
Node ID	Reg
Imported Data	...
Variables	...
Class Targets	
Regression Type	Logistic Regression
Link Function	Logit
Model Selection Options	
Selection Model	None
Selection Criterion	Default
Equation	
Main Effects	Yes
Two-Factor Interactions	No
Polynomial Terms	No
Polynomial Degree	2
User Terms	No
Term Editor	...
Status	
Last Error	
Last Status	
Needs Updating	Yes
Needs to Run	Yes
Time of Last Run	
Run Duration	

3. Select  in the Variables row of the Property Panel.



Note that the pre-transformed variables do not appear in the variable list because the default behavior is to hide those variables.

4. Select **OK** to close the Variables window.

Examine the Class Target section of the Property Panel. Based on the binary target variable, the Regression node has automatically set to do a logistic regression with a logit transformation. If you click on each of these you will see that the node can also do linear regression and can use two alternate forms of transformations, the probit and the complementary log-log transformations.

5. Examine the Model Selection Options of the Property Panel. Change the Selection Model option from None to **Stepwise** using the drop-down menu.

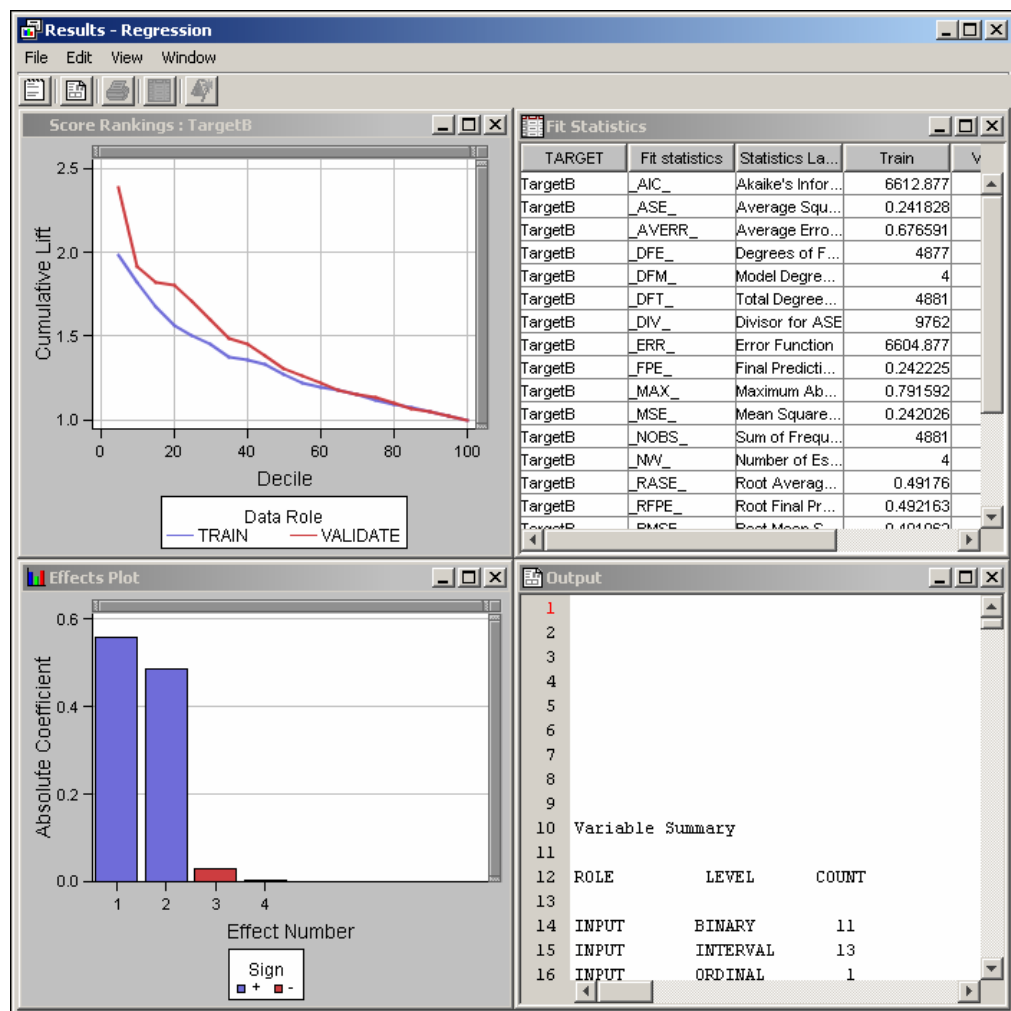
When you specify a selection criterion in this section, the node first performs the effect selection process, which generates a set of candidate models corresponding to each step in the process. Effect selection is done based on the Entry and/or Stay Significance Levels, which are 0.05 by default and can be changed using the advanced Property Panel. In this case a profit/loss matrix has been specified. Therefore, after the effect selection process terminates, the candidate model that optimizes profit/loss in the validation data set is chosen as the final model.

The equation section of the Property Panel enables the user to include all two-way interaction terms for class variables and polynomial terms in the model. It also enables the user to identify other specific higher-order terms to be included in the model. This allows for maximum flexibility when desired.



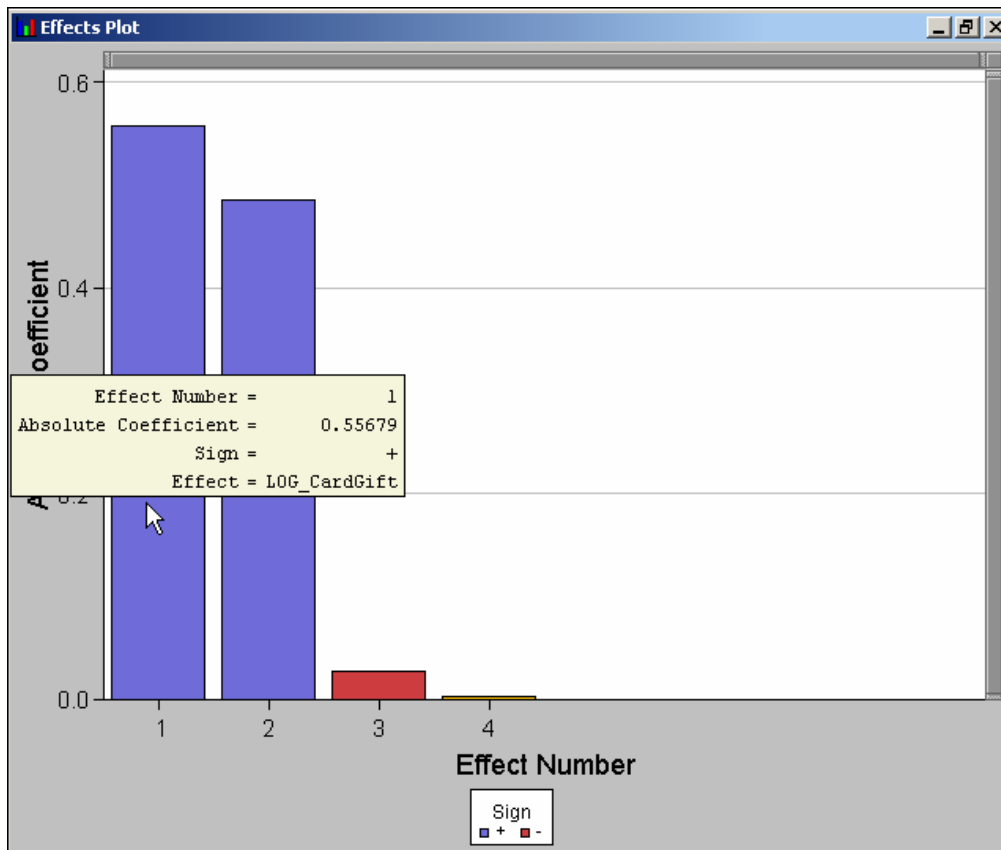
Additional model selection options, optimization options, and convergence criteria can be specified using the advanced Property Panel.

#### 6. Run the flow from the Regression node and examine the results.





Maximize and examine the effects plot.



The plot shows the model parameter estimates. When you hold your cursor over a bar in the plot the information about the coefficient and the variable it represents is displayed.



You can also determine the variables in the final model by examining the Output window or the parameter estimates. You can view the parameter estimate by selecting **View** ⇒ **Output Data Sets** ⇒ **Parameter Estimates**.

Regardless of the method chosen to determine the variables in the model for the model to predict **TargetB**, the variables in this model are

- **LOG\_CardGift** – the natural log of the donor's gifts to previous card promotions
- **CardProm** – the number of card promotions previously received
- **LastTime** – the elapsed time since the last donation.

7. Maximize the Fit Statistics window in the results.

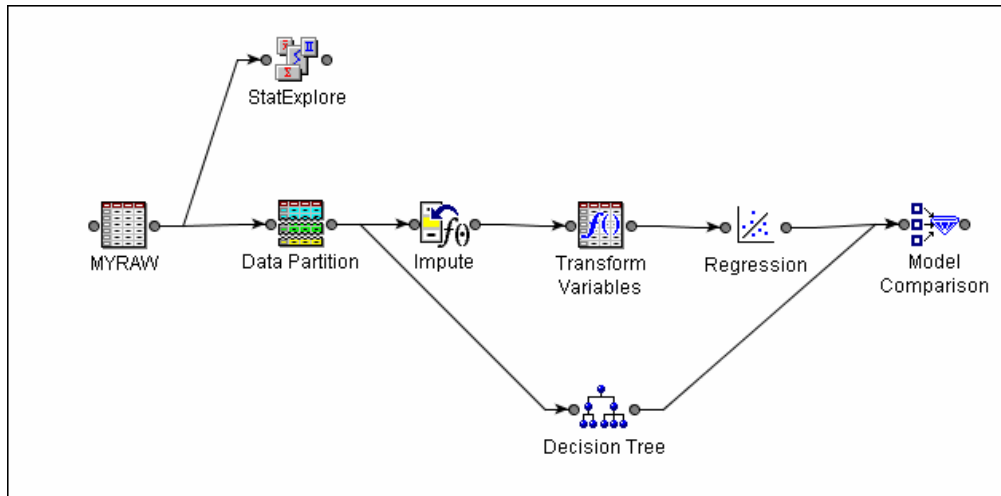
Fit Statistics					
TARGET	Fit statistics	Statistics La...	Train	Validation	Test
TargetB	_AIC_	Akaike's Infor...	6612.877	.	.
TargetB	_ASE_	Average Squ...	0.241828	0.239571	.
TargetB	_AVERR_	Average Erro...	0.676591	0.672128	.
TargetB	_DFE_	Degrees of F...	4877	.	.
TargetB	_DFM_	Model Degre...	4	.	.
TargetB	_DFT_	Total Degree...	4881	.	.
TargetB	_DIV_	Divisor for ASE	9762	4186	.
TargetB	_ERR_	Error Function	6604.877	2813.526	.
TargetB	_FPE_	Final Predicti...	0.242225	.	.
TargetB	_MAX_	Maximum Ab...	0.791592	0.764319	.
TargetB	_MSE_	Mean Square...	0.242026	0.239571	.
TargetB	_NOBS_	Sum of Frequ...	4881	2093	.
TargetB	_NW_	Number of Es...	4	.	.
TargetB	_RASE_	Root Averag...	0.49176	0.48946	.
TargetB	_RFPE_	Root Final Pr...	0.492163	.	.
TargetB	_RMSE_	Root Mean S...	0.491962	0.48946	.
TargetB	_SBC_	Schwarz's B...	6638.85	.	.
TargetB	_SSE_	Sum of Squa...	2360.726	1002.846	.
TargetB	_SUMWV_	Sum of Case ...	9762	4186	.
TargetB	_MISC_	Misclassificat...	0.47941	0.479216	.
TargetB	_PROF_	Total Profit fo...	384.3121	210.4971	.
TargetB	_APROF_	Average Prof...	0.078736	0.100572	.

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

8. Close the regression results.

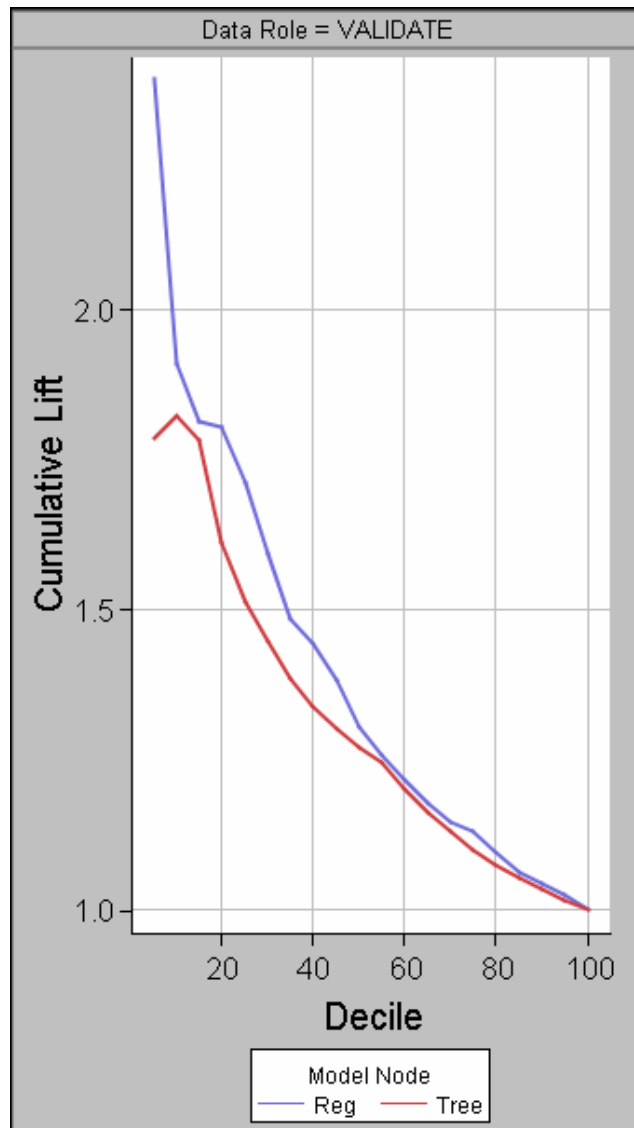
### Fitting a Default Decision Tree

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



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

3. Run the flow from the Model Comparison node and view the results.



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

4. Close the Model Comparison node results when you have finished inspecting the various results.

## 3.3 Exercises

### 1. Predictive Modeling Using Regression

- a. Return to the Organics diagram in the Exercise project. Use the Input Data Node and/or a StatExplore node to examine the distribution of the variables in the **ORGANICS** data set.
- b. In preparation for regression, is any imputation of missing values needed for this data set? If yes, should this imputation have been done before generating the decision tree models? Why or why not?
- c. Add an Impute node to the diagram and connect it to the Data Partition node.
- d. In the imputation, use tree imputation as the default method. Create missing value indicator variables. Replace missing values for **GENDER** with **U** for unknown.
- e. Add a Regression node to the diagram and connect it to the Replacement node.
- f. Choose the stepwise selection for the regression and run the diagram from the Regression node.
- g. Which variables are included in the final model? Which variable appears to be the most important variable in this model?
- h. In preparation for regression are any transformations of the data warranted? Why or why not?
- i. Add a Transform Variables node to the diagram and connect it to the Impute node.
- j. The variable **AFFL** appears to be skewed to the right. Use a square root transformation for **AFFL**. The variables **BILL** and **LTIME** also appear to be skewed to the right. Let SAS Enterprise Miner transform these variables to maximize normality.
- k. After running the Transform variables node, did the transformation of **AFFL** appear to result in a less skewed distribution? What transformation was chosen for the variables **BILL** and **LTIME**?
- l. Add another Regression node to the diagram and connect it to the Transform Variables node.
- m. Choose the stepwise selection method for the regression and run the diagram from the new Regression node.
- n. Which variables are included in the final model? Which variable appears to be the most important variable in the model?
- o. Connect the two Regression nodes to the Model Comparison node and rerun the Model Comparison node.

- p.** Use the Model Comparison node results to compare the regression models to the two decision tree models. Which model appears to be the best model?

## 3.4 Solutions to Exercises

### 1. Predictive Modeling Using Regression

- a. If you choose to add a StatExplore node to the diagram, it should be connected to the Input Data Source node.
- b. In general, imputation of missing values should be considered **before** constructing a regression model. Any observation with missing values for any of the variables in the regression model will not be used in the regression model. One way to determine the extent of missing values in this data set is to use the StatExplore node.
  - 1) Select the StatExplore node in the diagram and examine the Property Panel.
  - 2) Change the option to Hide Rejected Variables to **No** and change the option for Interval Variables to **Yes** using the drop-down menus.

Property	Value
Node ID	Stat
Imported Data	...
Variables	...
Use Segment Variables	No
Variable Selection	
Hide Rejected Variables	No
Number of Selected Variables	1000
Chi-Square Statistics	
Chi-Square	Yes
Interval Variables	Yes
Number of Bins	5
Correlation Statistics	
Correlations	Yes
Pearson Correlations	Yes
Spearman Correlations	No

- 3) Run the StatExplore node and examine the Output window in the Results.

Class Variable Summary Statistics (maximum 500 observations printed)				
Variable	Role	Numcat	NMiss	Mode
CLASS	INPUT	4	0	Silver
GENDER	INPUT	4	2512	F
NGROUP	INPUT	8	674	C
REGION	INPUT	6	465	South East
TV_REG	INPUT	14	465	London
ORGYN	TARGET	2	0	0

The variable **GENDER** has a relatively large number of missing values.

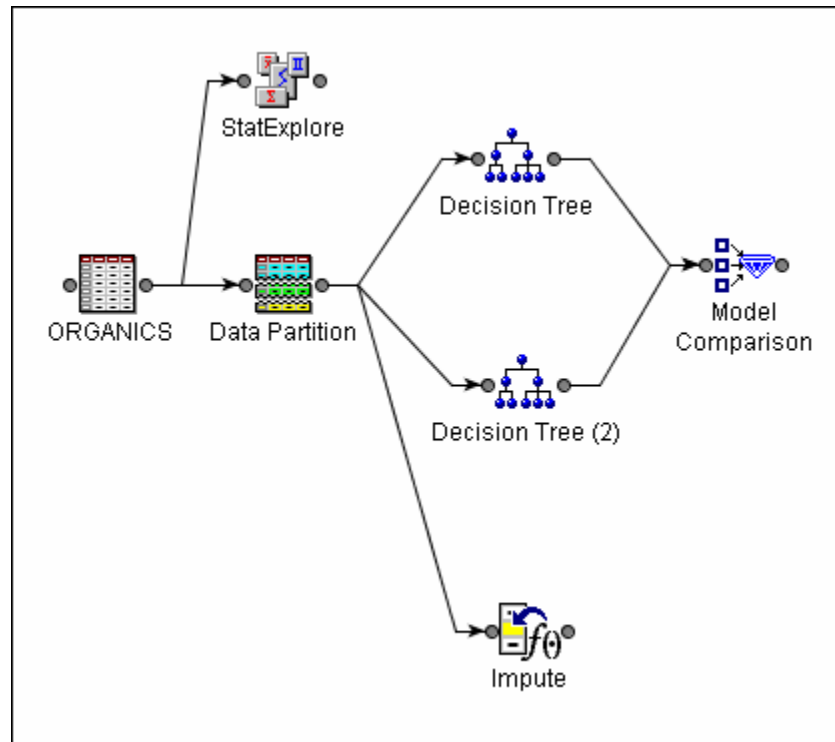
Interval Variable Summary Statistics (maximum 500 variables printed)					
Variable	ROLE	Mean	StdDev	Non Missing	Missing
AFFL	INPUT	9	3	21138	1085
AGE	INPUT	54	13	20715	1508
BILL	INPUT	4421	7559	22223	0
LTIME	INPUT	7	5	21942	281

The variables **AFFL** and **AGE** have over 1000 missing values each.

Imputation of missing values should be done prior to generating a regression model. However, imputation is **not** necessary before generating a decision tree model because a decision tree can use missing values just like any other data value in the data set.

4) Close the StatExplore node results and return to the diagram workspace.

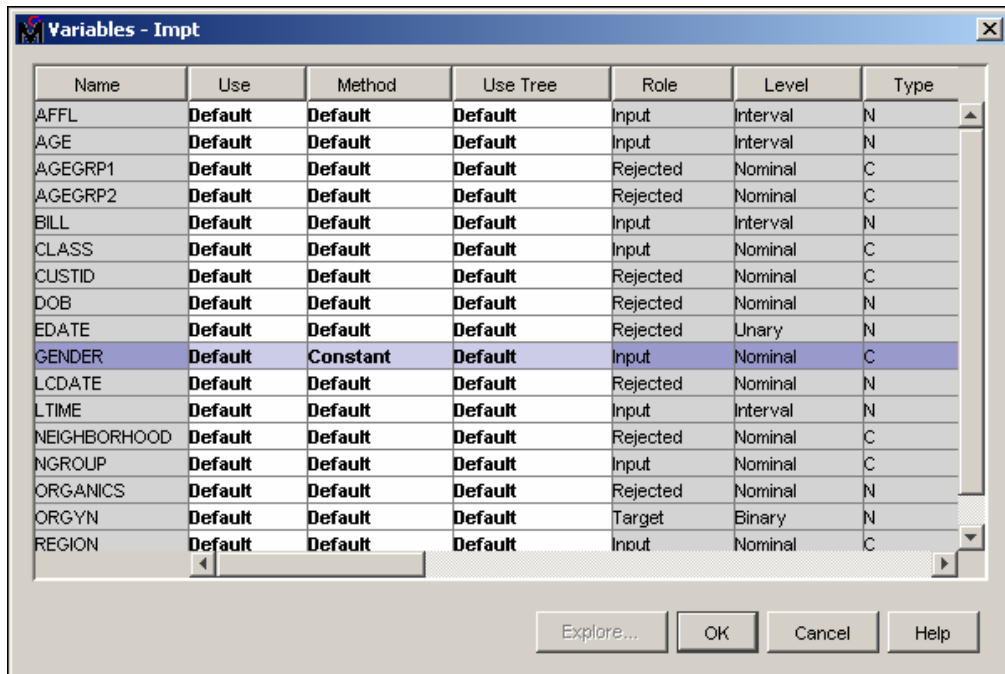
c. After adding an Impute node, the diagram should appear as shown below.





## d. Choose the imputation methods.

- 1) Select the Impute node in the diagram.
- 2) To use tree imputation as the default method of imputation change the Default Input Method for both class and interval variables to **Tree** using the drop-down menus.
- 3) To create missing value indicator variables, change the Indicator Variable property to **Unique** and the Indicator Variable Role to **Input** using the drop-down menus in the Property Panel.
- 4) To replace missing values for the variable **GENDER** with U, first change the Default Character Value to **U**. Select **...** in the Variables row of the Property Panel. Change the Method for the variable **GENDER** to **Constant** as shown below.

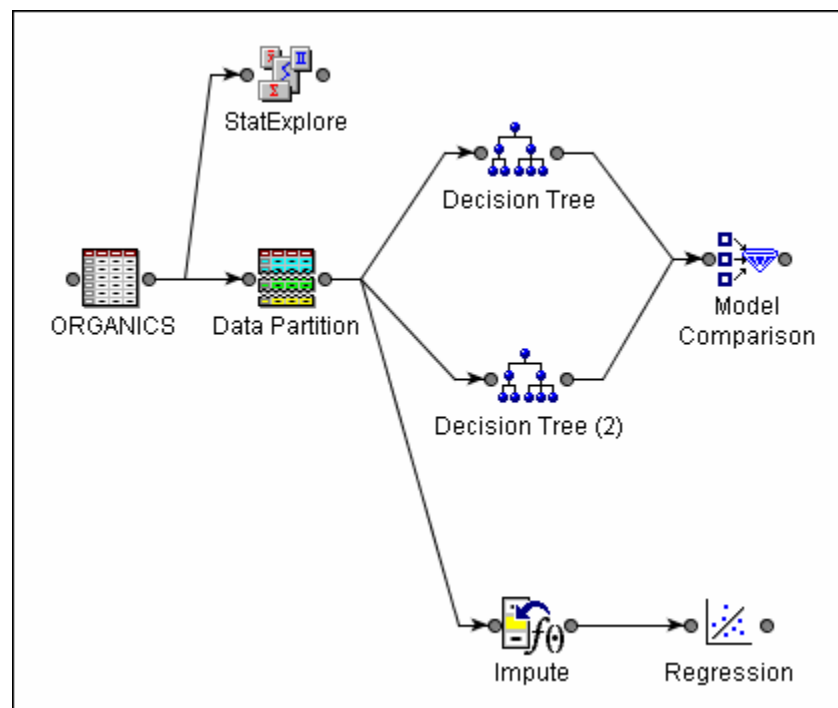


- 5) Select **OK** to confirm the change.

- 6) The Property Panel for the Impute node should now appear as shown below.

Property	Value
Node ID	Impt
Imported Data	...
Variables	...
Class Variables	
Default Input Method	Tree
Default Target Method	None
Interval Variables	
Default Input Method	Tree
Default Target Method	None
Default Constant Value	
Default Character Value	J
Default Number Value	.
Indicator Variables	
Indicator Variable	Unique
Indicator Variable Role	Input
Status	
Last Error	
Last Status	
Needs Updating	No
Needs to Run	Yes
Time of Last Run	
Run Duration	

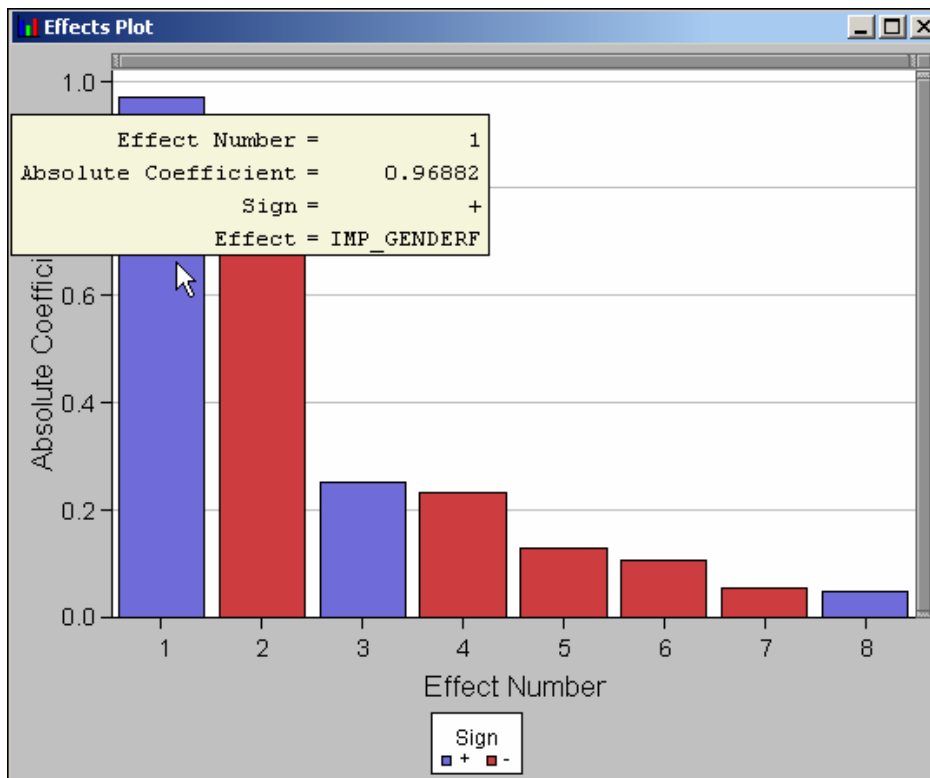
- e. Add a Regression node to the diagram.



- f. Choose the stepwise regression method and run the diagram from the Regression node.
- 1) Select the Regression node in the diagram.
  - 2) In the Property Panel, change the Selection Model property to **Stepwise** using the drop-down menu.
  - 3) To run the regression, right-click on the Regression node and select **Run**. Select **Yes** to confirm that you want to run the path.
  - 4) Select **OK** to confirm completion of the run.
  - 5) To view the results, right-click on the Regression node and select **Results...**
- g. Determine which variables are in the final model and which variable appears to be the most important variable.
- 1) Maximize the Output window.
  - 2) Scroll toward the bottom of the output to determine the selected model.

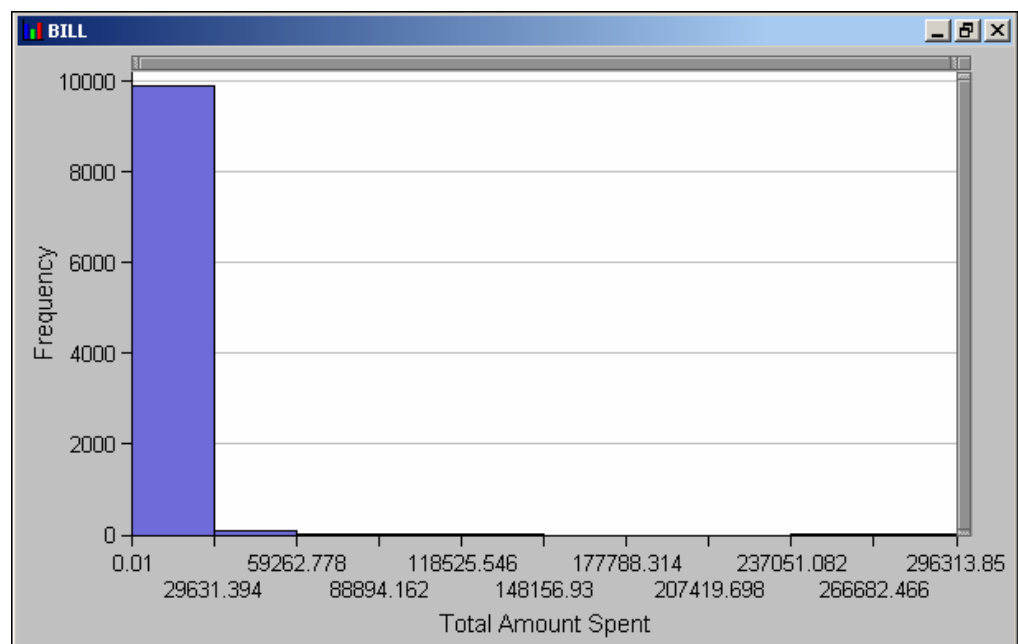
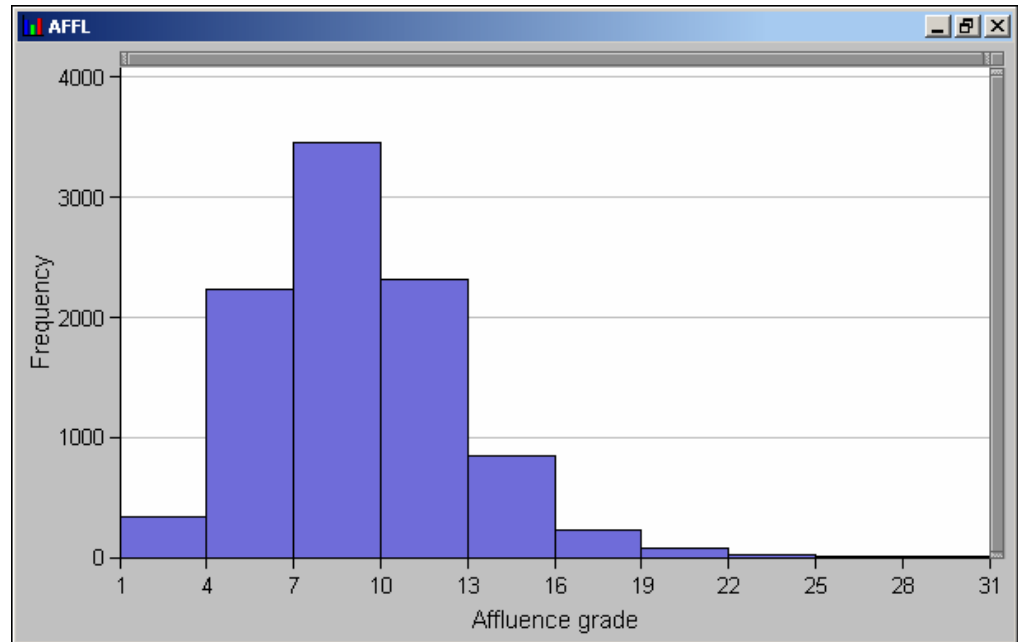
The selected model includes the variable **IMP\_AFFL**, **IMP\_AGE**, **IMP\_GENDER**, and the missing value indicator variables for **AFFL**, **AGE**, and **GENDER**.

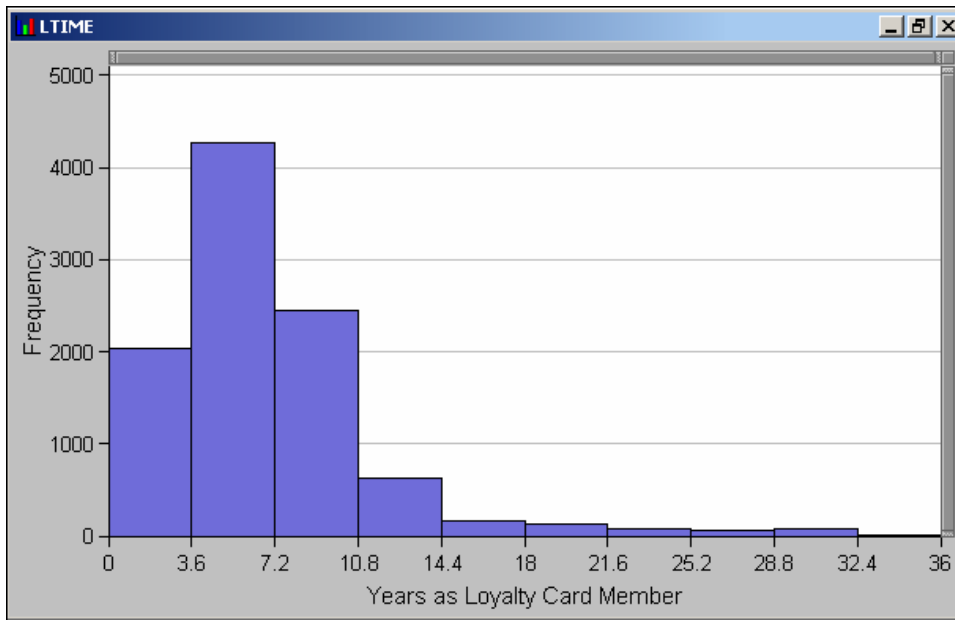
- 3) Examine the Effects Plot.



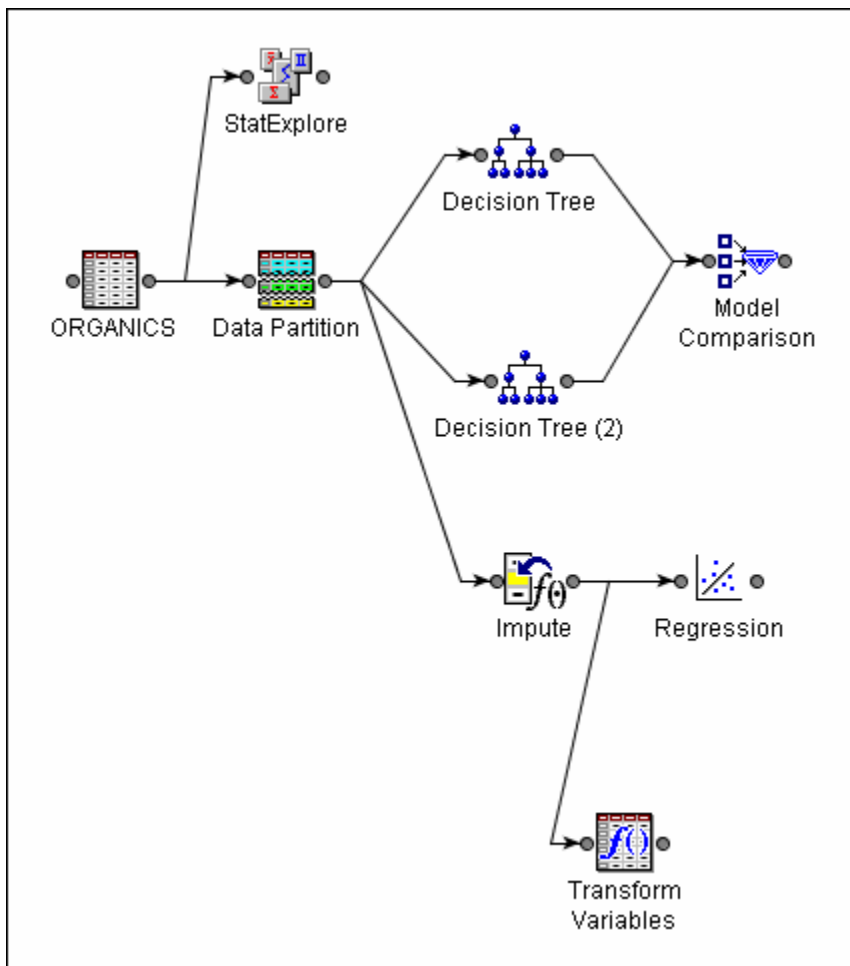
The variable with the largest absolute coefficient is **IMP\_GENDERF**. This is an indication that **GENDER** is the most important variable in this model.

- 4) Close the regression results and return to the diagram workspace.
- h. The variables **AFFL**, **BILL**, and **LTIME** are skewed to the right and may need to be transformed for a regression. This is evident in the histograms that can be viewed in the exploration tool available in the Input Data node and other nodes.




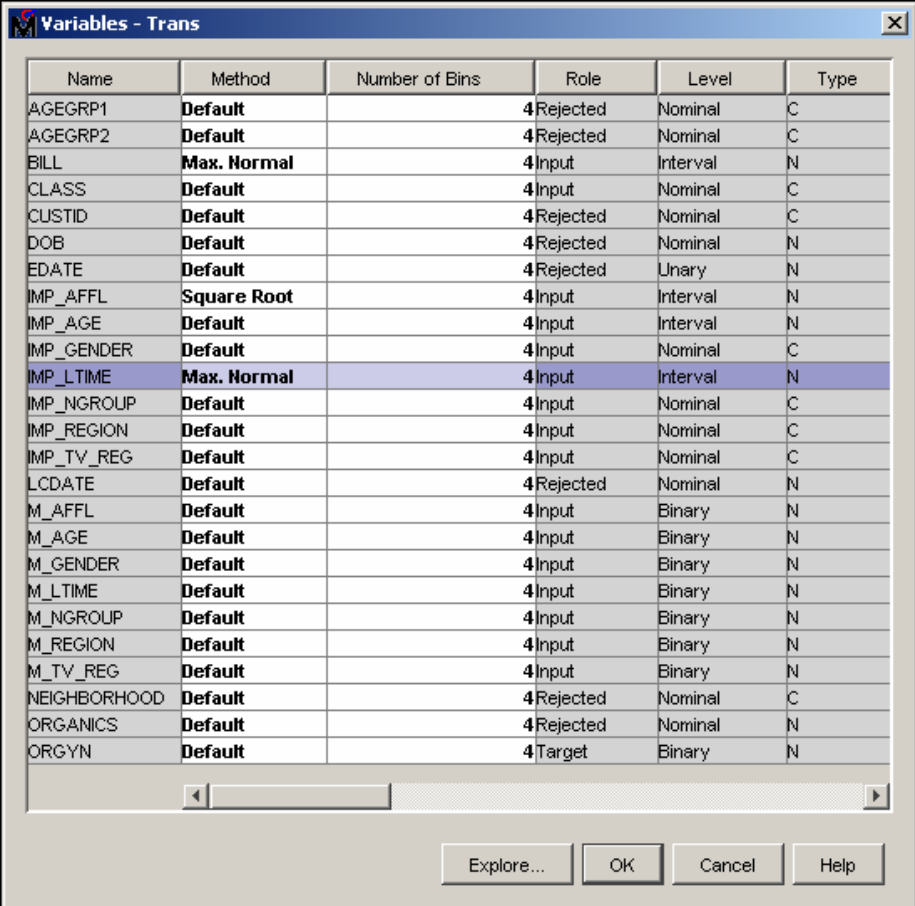


i. Add a Transform Variables node to the diagram.



j. Transform the variables **AFFL**, **BILL**, and **LTIME**.

- 1) Select the Transform Variable node in the diagram and select  in the Variables row of the Property Panel.
- 2) In the Method column for the variable **IMP\_AFFL**, select **Square Root** from the drop-down menu. In the Method column for the variables **BILL** and **IMP\_LTIME**, select **Max. Normal** from the drop-down menu. The Variables window should now appear as shown below.

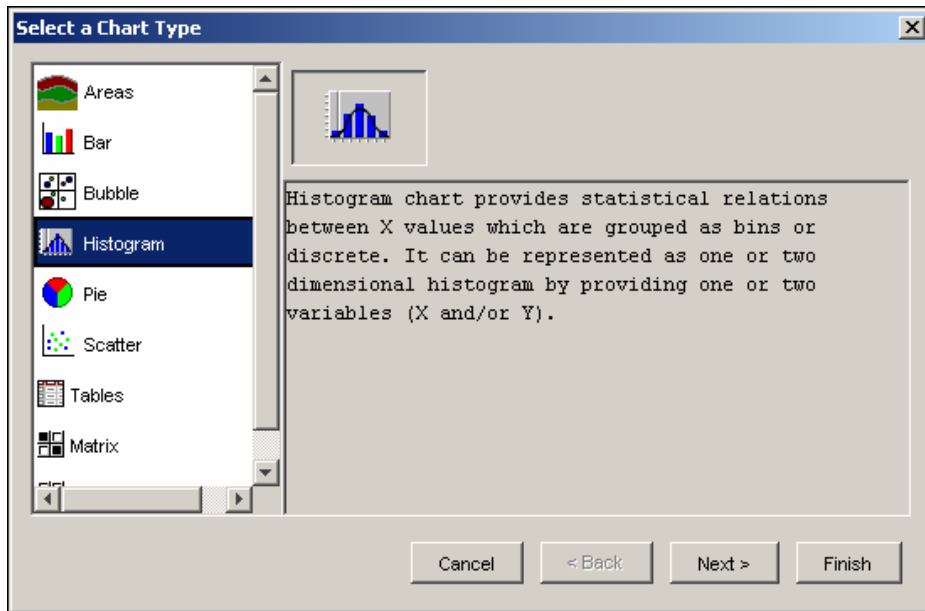


Name	Method	Number of Bins	Role	Level	Type
AGEGRP1	Default		4 Rejected	Nominal	C
AGEGRP2	Default		4 Rejected	Nominal	C
BILL	Max. Normal		4 Input	Interval	N
CLASS	Default		4 Input	Nominal	C
CUSTID	Default		4 Rejected	Nominal	C
DOB	Default		4 Rejected	Nominal	N
EDATE	Default		4 Rejected	Unary	N
IMP_AFFL	Square Root		4 Input	Interval	N
IMP_AGE	Default		4 Input	Interval	N
IMP_GENDER	Default		4 Input	Nominal	C
IMP_LTIME	Max. Normal		4 Input	Interval	N
IMP_NGROUP	Default		4 Input	Nominal	C
IMP_REGION	Default		4 Input	Nominal	C
IMP_TV_REG	Default		4 Input	Nominal	C
LCDATE	Default		4 Rejected	Nominal	N
M_AFFL	Default		4 Input	Binary	N
M_AGE	Default		4 Input	Binary	N
M_GENDER	Default		4 Input	Binary	N
M_LTIME	Default		4 Input	Binary	N
M_NGROUP	Default		4 Input	Binary	N
M_REGION	Default		4 Input	Binary	N
M_TV_REG	Default		4 Input	Binary	N
NEIGHBORHOOD	Default		4 Rejected	Nominal	C
ORGANICS	Default		4 Rejected	Nominal	N
ORGYN	Default		4 Target	Binary	N

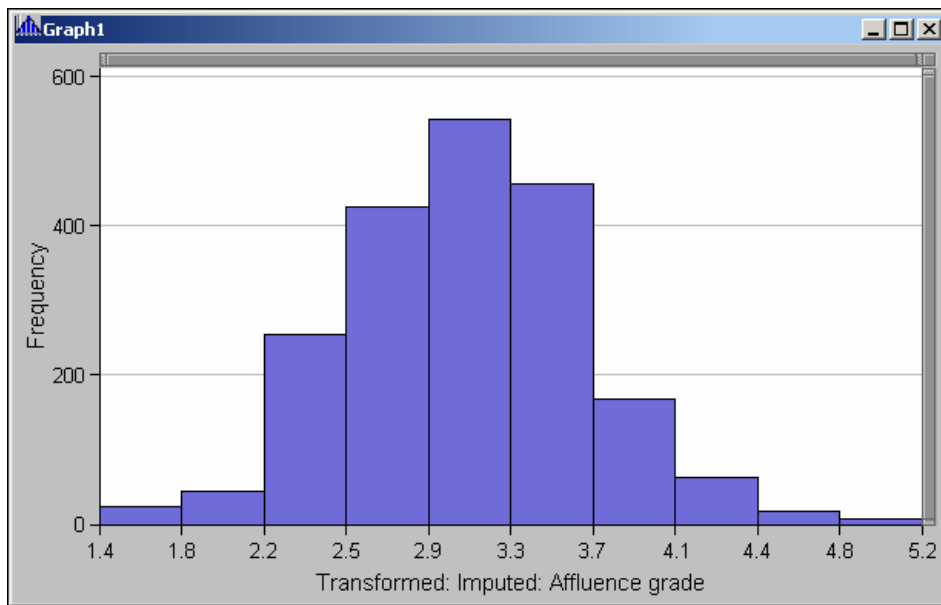
- 3) Select **OK** to confirm the changes.

k. Run the Transform Variables node and examine the results.

- 1) After the results from the Transform Variables node are open, select **View** ⇒ **Output Data Sets** ⇒ **Train**.
- 2) In the Explore window, select **Actions** ⇒ **Plot...**



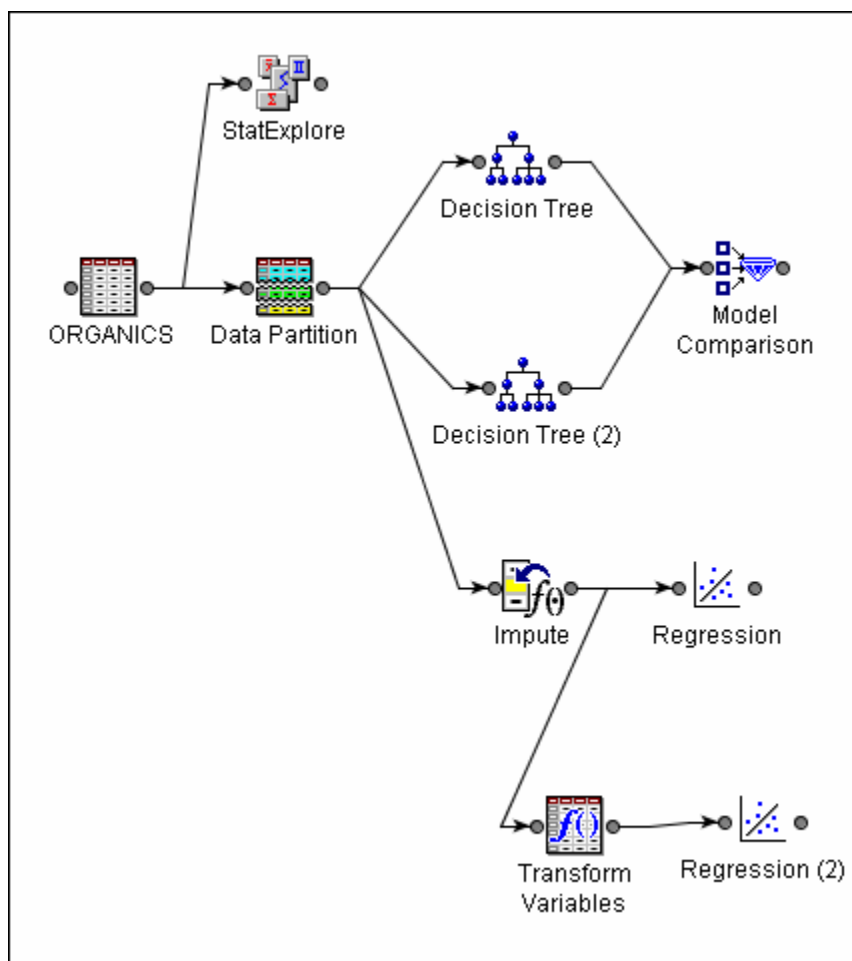
- 3) Select **Histogram** as the type of chart, then select **Next>**.
- 4) Change the Role of **SQRT\_IMP\_AFFL** to **X**.
- 5) Select **Finish**.



The transformed variable appears to be more normally distributed than the original variable.

- 6) To view the transformations for the other variables, close the graph and examine the Transformations window in the results. The transformation chosen for the variable **BILL** was a square root transformation, and the transformation for **IMP\_LTIME** was the fourth root.
- 7) Close the results and return to the diagram workspace.

1. Add a second Regression node to the diagram.



m. Choose the stepwise regression method and run the diagram from the new Regression node.

- 1) Select the Regression node in the diagram.
- 2) In the Property Panel, change the Selection Model property to **Stepwise** using the drop-down menu.
- 3) To run the regression, right-click on the Regression node and select **Run**. Select **Yes** to confirm that you want to run the path.
- 4) Select **OK** to confirm completion of the run.
- 5) To view the results, right-click on the Regression node and select **Results...**

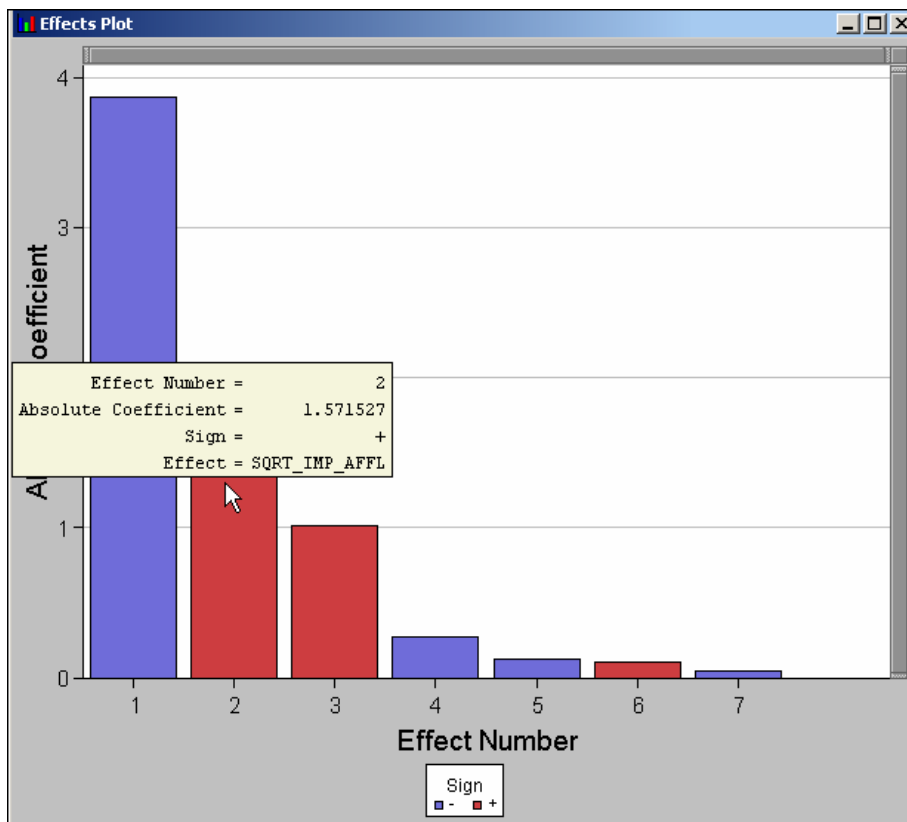


- n. Determine which variables are in the final model and which variable appears to be the most important variable.

- 1) Maximize the Output window.
- 2) Scroll toward the bottom of the output to determine the selected model.

The selected model includes the variables **IMP\_AGE**, **IMP\_GENDER**, the missing value indicator variables for **AFFL** and **GENDER.**, and the variable **SQRT\_IMP\_AFFL**.

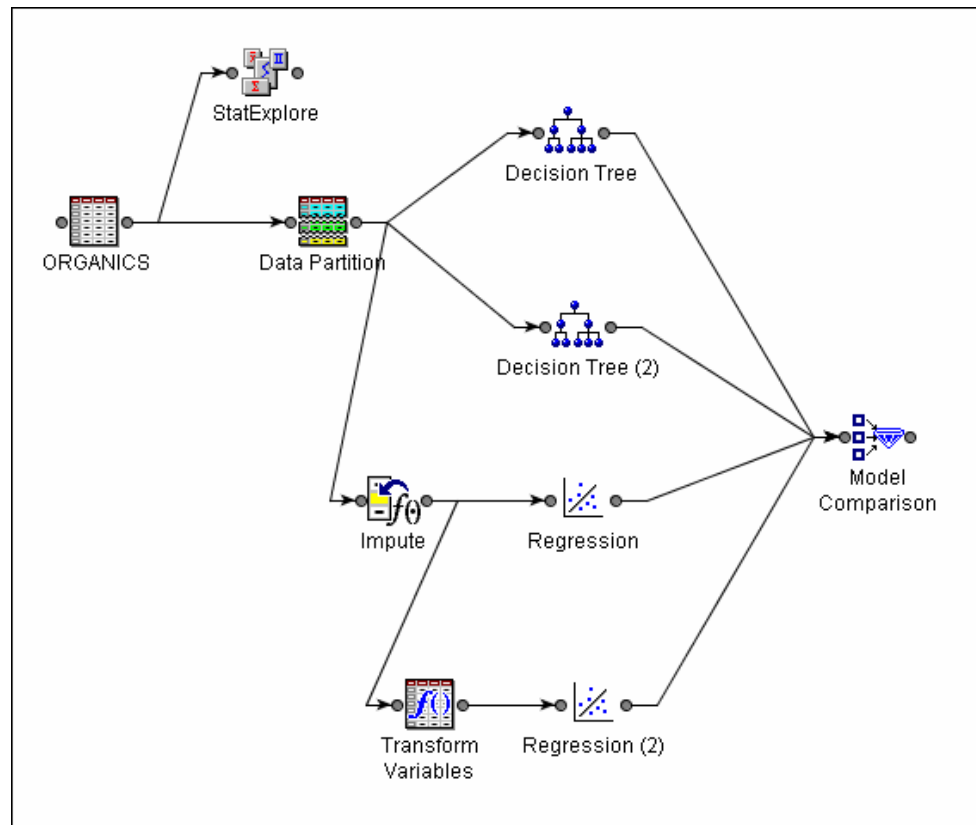
- 3) Examine the Effects Plot.



The variable with the largest absolute coefficient is **SQRT\_IMP\_AFFL**. This is an indication that **AFFL** is the most important variable in this model.

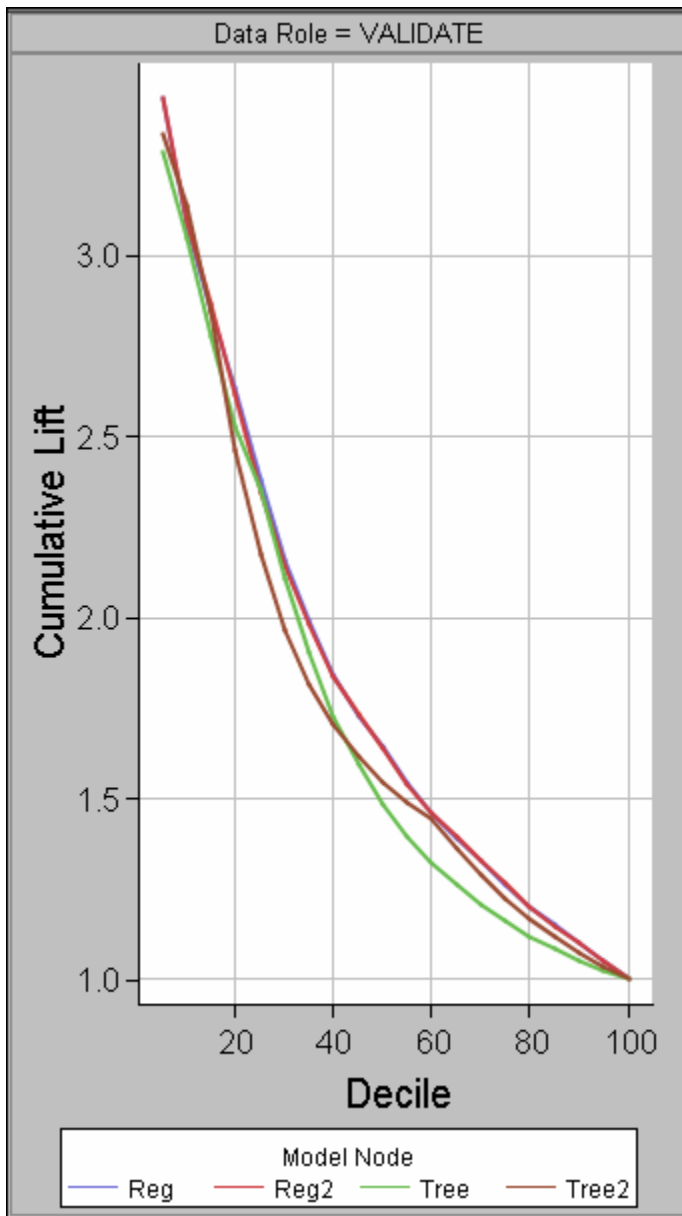
- 4) Close the regression results and return to the diagram workspace.

- o. Connect the two Regression nodes to the Model Comparison node.



- 1) To run the diagram from the Model Comparison node, right-click on the node and select **Run**.
- 2) Select **Yes** to confirm that you want to run the path.
- 3) Select **OK** to confirm that the run is complete.
- 4) To view the results, right-click on the Model Comparison node and select **Results...**.

- p. Compare the regression models to the two decision tree models.



Based upon the lift chart, the four models are similar in their predictive ability. No one model is clearly superior to the others.



# Chapter 4 Variable Selection

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



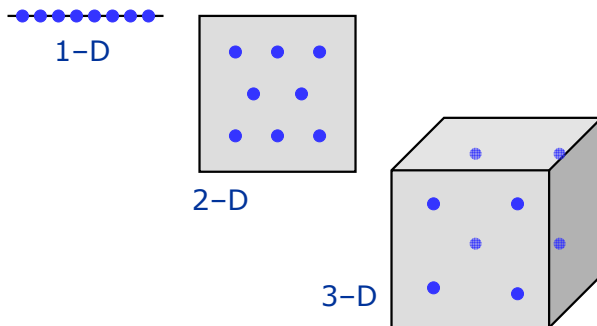
## 4.1 Variable Selection and SAS Enterprise Miner

### Objectives

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

3

### The Curse of Dimensionality



4

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

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

### Methods of Variable Selection

- Stepwise Regression
- Decision Trees
- Variable Selection Node

5

Three possible methods of variable selection available in SAS Enterprise Miner are

- stepwise regression
- decision trees
- the Variable Selection node.



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

### Stepwise Regression

Stepwise regression methods

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

6

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



### Decision Trees

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

7

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

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

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

### Variable Selection Node

Selection is based on one of two criteria:

- R-square
- chi-square, which is for binary targets only.

8

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

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

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

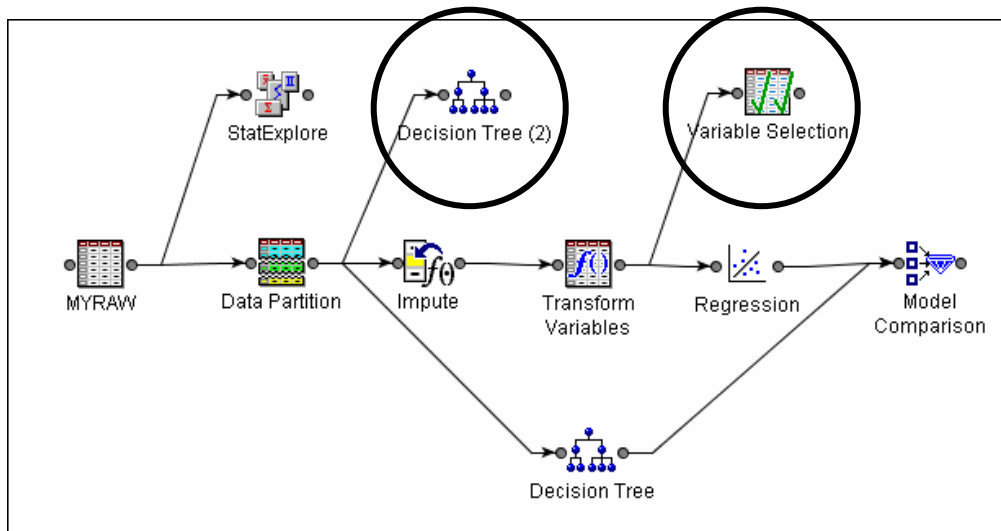
If the target is nonbinary, only the first two steps are performed.

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



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

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



### Variable Selection Using a Decision Tree

1. Select the new Decision Tree node, **Decision Tree (2)** and examine the Property Panel.

Property	Value
Node ID	Tree2
Imported Data	...
Variables	...
Interactive Training	...
Splitting Criterion	Default
Significance Level	0.2
Missing Values	Use in search
Leaf Size	5
Maximum Branch	2
Maximum Depth	6
Minimum Categorical S	5
Number of Rules	5
Number of Surrogate F	0
Split Size	

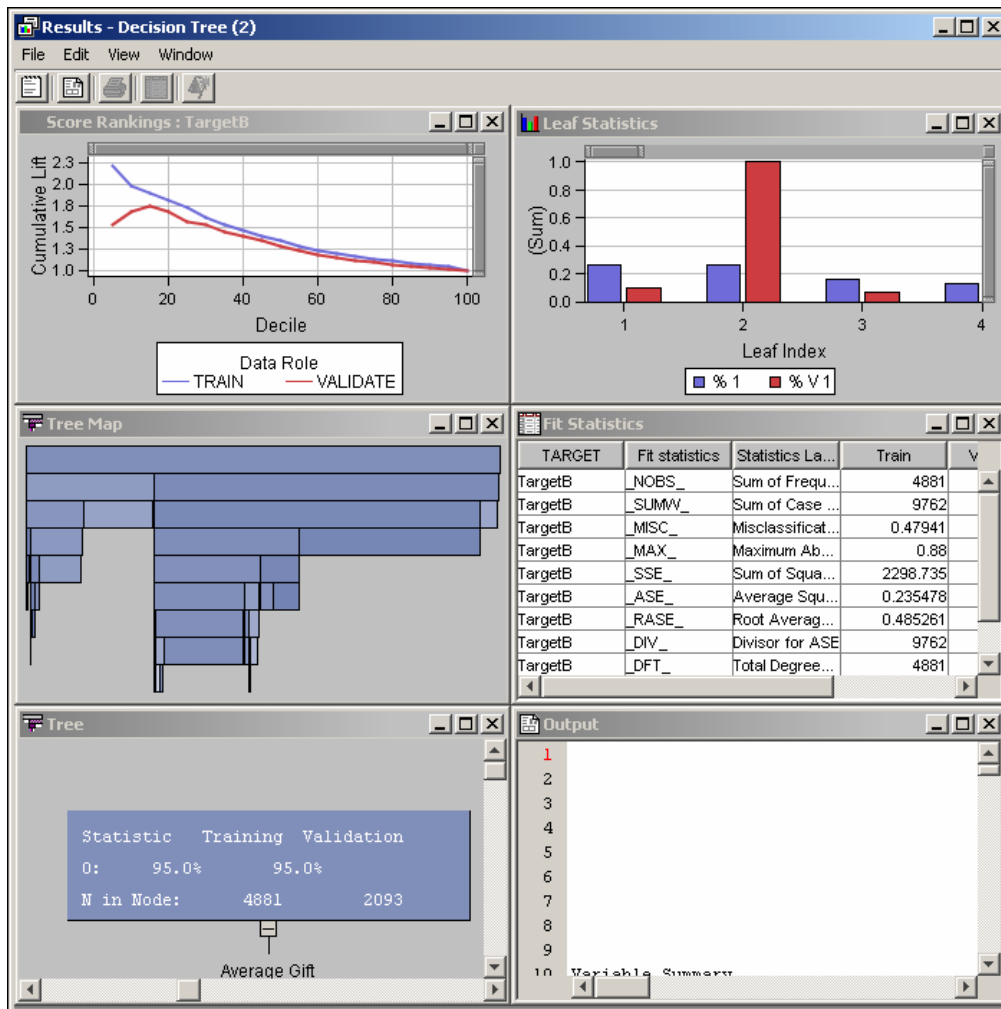
2. Change the Splitting Criterion to **Gini**.
3. Change the maximum depth of the tree to **8** to allow a larger tree to grow.

Property	Value
Node ID	Tree2
Imported Data	...
Variables	...
Interactive Training	...
Splitting Criterion	Gini
Significance Level	0.2
Missing Values	Use in search
Leaf Size	5
Maximum Branch	2
Maximum Depth	8
Minimum Categorical S	5
Number of Rules	5
Number of Surrogate F	0
Split Size	



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

4. Run the flow from the Decision Tree (2) node and view the results.



5. Expand the Output window and scroll to the Tree Leaf Report.

108	Tree Leaf Report					
109						
110			Training		Validation	% V
111	Node	Depth	Observations	% 1	Observations	1
112						
113	14	3	1866	0.04	788	0.04
114	141	7	823	0.05	334	0.05
115	4	2	719	0.09	327	0.09
116	22	4	441	0.08	175	0.06
117	57	5	267	0.03	123	0.04
118	6	2	191	0.09	74	0.09
119	56	5	134	0.06	47	0.09
120	144	7	81	0.13	39	0.06
121	201	8	56	0.16	36	0.07
122	44	5	56	0.06	28	0.05
123	143	7	45	0.07	21	0.06
124	200	8	37	0.05	25	0.05
125	77	6	31	0.01	12	0.07
126	20	4	29	0.04	13	0.02
127	40	5	25	0.01	11	0.03
128	208	8	18	0.03	5	0.04
129	138	7	16	0.00	9	0.10
130	118	7	8	0.03	5	0.01
131	142	7	8	0.00	7	0.02
132	119	7	7	0.26	3	0.10
133	209	8	7	0.26	1	1.00
134	198	8	6	0.01	5	0.04
135	41	5	5	0.08	2	0.05
136	199	8	5	0.08	3	0.10

The final tree has 24 leaves.

6. Scroll up in the Output window to the Variable Importance table.

84	Variable Importance						
85							
86	Obs	NAME	LABEL	NRULES	IMPORTANCE	VIMPORTANCE	RATIO
87							
88	1	AverageGift	Average Gift	3	1.00000	1.00000	1.00000
89	2	LastTime	Time Since Last Donation	4	0.74474	0.63073	0.84692
90	3	CardGift	Gift to Card Promotions	2	0.46049	0.61595	1.33760
91	4	TotProm	Total Number of Promotions	3	0.42606	0.00000	0.00000
92	5	Income	Income Level	2	0.40026	0.22060	0.55113
93	6	CardProm	Number of Card Promotions	2	0.38575	0.00000	0.00000
94	7	Age	Donor's Age	1	0.32611	0.00000	0.00000
95	8	FirstTime	Time Since First Donation	1	0.28548	0.00000	0.00000
96	9	LocalGov	Local Government (% of households)	1	0.27420	0.00000	0.00000
97	10	Gender		1	0.22925	0.00000	0.00000
98	11	MaleVeterans	Male Veterans (% of households)	1	0.22113	0.00000	0.00000
99	12	FederalGov	Federal Government (% of households)	1	0.18305	0.00000	0.00000
100	13	StateGov	State Government (% of households)	1	0.14110	0.00000	0.00000

These 13 variables out of the original 18 variables have been retained for future modeling. The other 5 variables will remain in the data set as rejected variables.

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

7. Close the tree results.

### Variable Selection Using the Variable Selection Node

1. Select the Variable Selection node in the flow and examine the Property Panel.

Property	Value
Node ID	Varsel
Imported Data	...
Variables	...
Max Class Level	100
Max Missing Percentag	50
Target Model	Default
Hide Rejected Variable	Yes
Reject Unused Variabl	Yes

2. To view additional properties of the node, select **View** ⇒ **Property Sheet** ⇒ **Advanced**.

Property	Value
Node ID	Varsel
Imported Data	...
Variables	...
Max Class Level	100
Max Missing Percent	50
Target Model	Default
Hide Rejected Variab	Yes
Reject Unused Variab	Yes
Chi-Square Options	
Number of Bins	50
Maximum Pass Numb	6
Minimum Chi-Square	3.84
R-Square Options	
Maximum Variable Nu	3000
Minimum R-Square	0.0050
Stop R-Square	5.0E-4
Use AOV16 Variable	No
Use Group Variables	Yes
Use Interactions	No
SPDS	Yes

The Target Model property is set to Default. With this default setting, if the target is binary and the model has more than 400 degrees of freedom, the chi-squared method is used. Otherwise, the R-square method of variable selection is used.

### Selection Using the R-square Criterion

1. Consider the settings associated with the R-square criterion first. Change the Target Model property to R-Square.

Property	Value
Node ID	Varsel
Imported Data	...
Variables	...
Max Class Level	100
Max Missing Percentag	50
Target Model	R-Square
Hide Rejected Variable	Yes
Reject Unused Variable	Yes
Chi-Square Options	
R-Square Options	
Maximum Variable Num	3000
Minimum R-Square	0.0050
Stop R-Square	5.0E-4
Use AOV16 Variables	No
Use Group Variables	Yes
Use Interactions	No
SPDS	Yes

Recall that a three-step process is performed when you apply the R-square variable selection criterion to a binary target. The Property Panel enables you to specify the maximum number of variables that can be selected, the cutoff minimum squared correlation measure for an individual variable to be selected, and the necessary  $R^2$  improvement for a variable to remain as an input variable.

Additional available options include

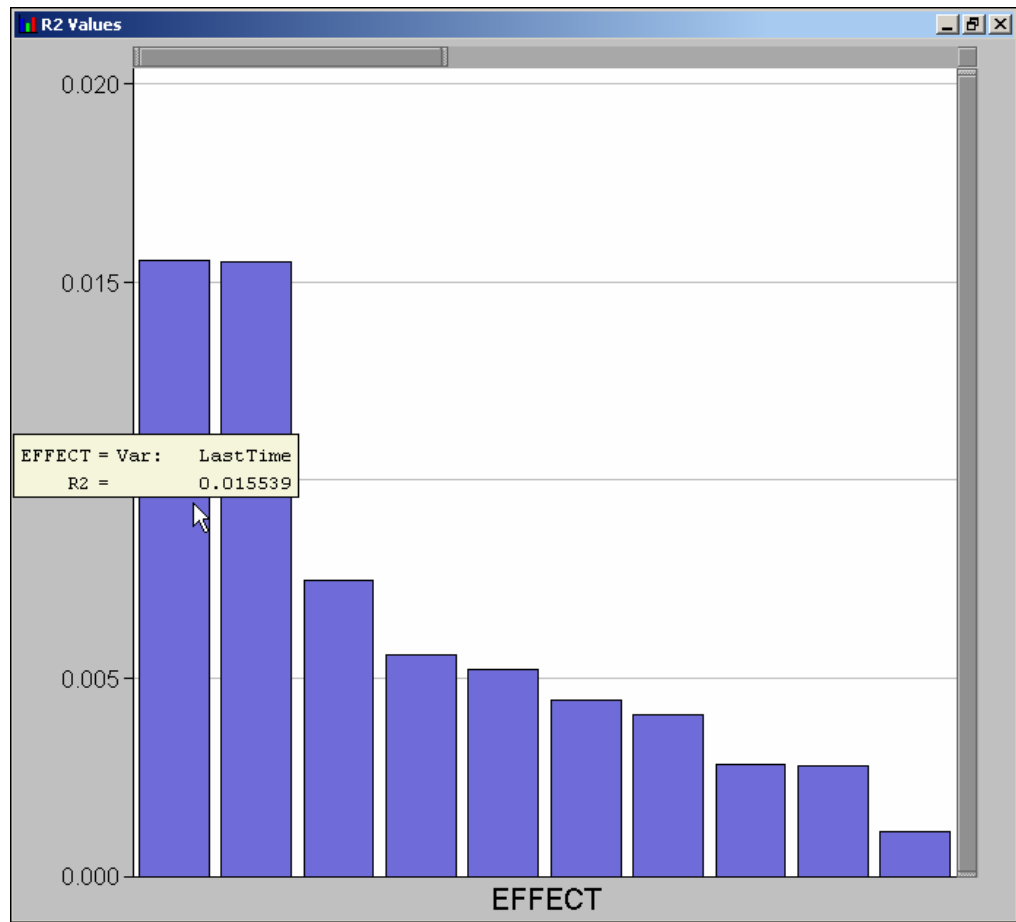
- Use AOV16 Variables. When selected, this option requests SAS Enterprise Miner to bin interval variables into 16 equally spaced groups (AOV16). The AOV16 variables are created to help identify nonlinear relationships with the target. Bins with zero observations are eliminated, meaning an AOV16 variable can have fewer than 16 bins.
  - Use Group Variables. When set to Yes, this option allows the number of levels of a class variable to be reduced based on the relationship of the levels to the target variable.
  - Use Interactions. When this option is selected, SAS Enterprise Miner evaluates 2-way interactions for categorical inputs.
2. Leave the R-Square options with their default settings.
  3. Run the flow from the Variable Selection node and view the results.
  4. Examine the Variable Selection window. Click on the **Role** column heading to sort the variable by their assigned roles. Then click on the **Comment** column heading. Inspect the results.



Variable Selection					
Name	Role	Level	Type	Label	Comment
AverageGift	Input	Interval	N	Average Gift	.
LastTime	Input	Interval	N	Time Since Last Donation	.
LOG_CardGift	Input	Interval	N	Transformed: Gift to Card Promotions	.
TotProm	Input	Interval	N	Total Number of Promotions	.
TargetD	Rejected	Interval	N	Amount Donated	.
CardProm	Rejected	Interval	N	Number of Card Promotions	Varsel: Small R-square value
FirstTime	Rejected	Interval	N	Time Since First Donation	Varsel: Small R-square value
IMP_Age	Rejected	Interval	N	Imputed: Donor's Age	Varsel: Small R-square value
IMP_Gender	Rejected	Binary	C	Imputed Gender	Varsel: Small R-square value
IMP_Homeow...	Rejected	Binary	C	Imputed Homeowner	Varsel: Small R-square value
IMP_Income	Rejected	Ordinal	N	Imputed: Income Level	Varsel: Small R-square value
IMP_PCOwner	Rejected	Binary	C	Imputed: Personal Computer Owner	Varsel: Small R-square value
IMP_Pets	Rejected	Binary	C	Imputed: Pet(s) in Household	Varsel: Small R-square value
LOG_Federal...	Rejected	Interval	N	Transformed: Federal Government (%...	Varsel: Small R-square value
LOG_IMP_Ti...	Rejected	Interval	N	Transformed: Imputed: Time between ...	Varsel: Small R-square value
LOG_LocalGov	Rejected	Interval	N	Transformed: Local Government (% o...	Varsel: Small R-square value
LOG_StateGov	Rejected	Interval	N	Transformed: State Government (% o...	Varsel: Small R-square value
MaleMilitary	Rejected	Interval	N	Active Duty Military Males (% of hous...	Varsel: Small R-square value
MaleVeterans	Rejected	Interval	N	Male Veterans (% of households)	Varsel: Small R-square value
M_Age	Rejected	Binary	N	Imputation Indicator for Age	Varsel: Small R-square value
M_Gender	Rejected	Binary	N	Imputation Indicator for Gender	Varsel: Small R-square value
M_Homeowner	Rejected	Binary	N	Imputation Indicator for Homeowner	Varsel: Small R-square value
M_Income	Rejected	Binary	N	Imputation Indicator for Income	Varsel: Small R-square value
M_PCOwner	Rejected	Binary	N	Imputation Indicator for PCOwner	Varsel: Small R-square value
M_Pets	Rejected	Binary	N	Imputation Indicator for Pets	Varsel: Small R-square value
M_TimeLag	Rejected	Binary	N	Imputation Indicator for TimeLag	Varsel: Small R-square value

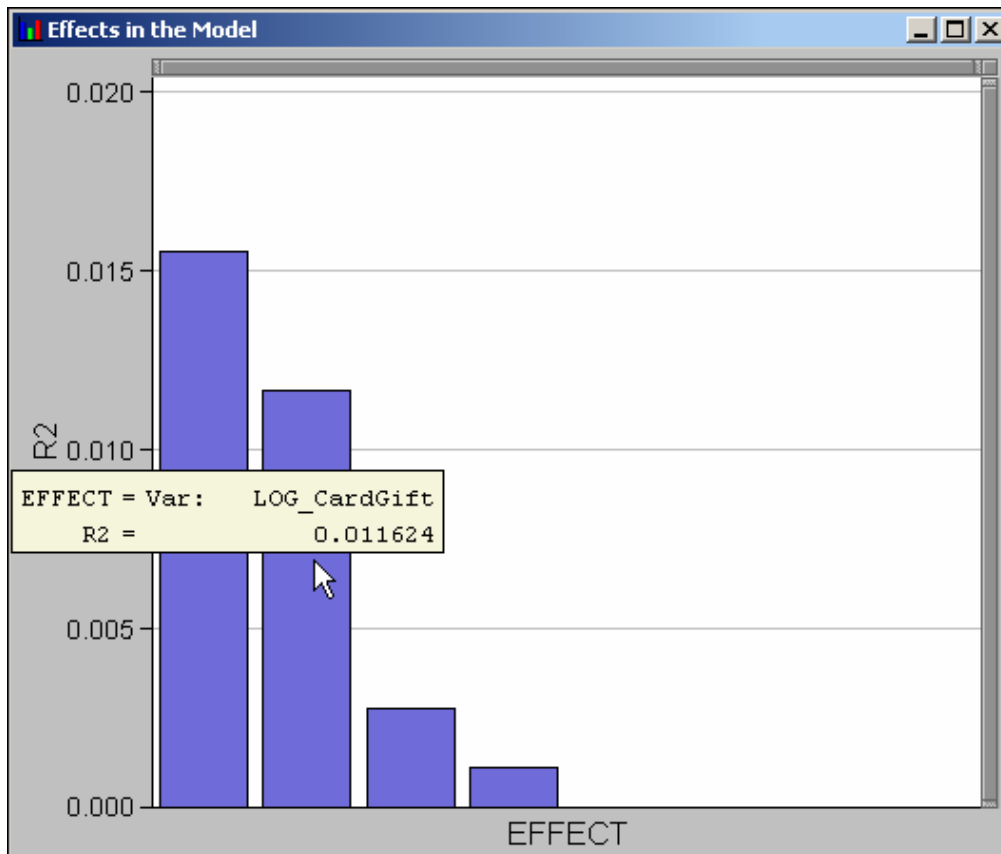
The variables **AverageGift**, **LastTime**, **LOG\_CardGift**, and **TotProm** are retained for future modeling. The remaining potential input variables were rejected because of a small R-square value.

5. Examine the R2 Values window.



This plot shows the R-square for each effect with **TargetB**.

6. Select **View** ⇒ **R-Square:Plots** ⇒ **Effects in the Model**.



This plot shows the improvement in R-square as each selected variable is added into the model.

7. Close the Results window.

### Selection Using the Chi-square Criterion

1. Select the Variable Selection node in the diagram.
2. Change the Target Model property to **Chi-Square** in the Property Panel.

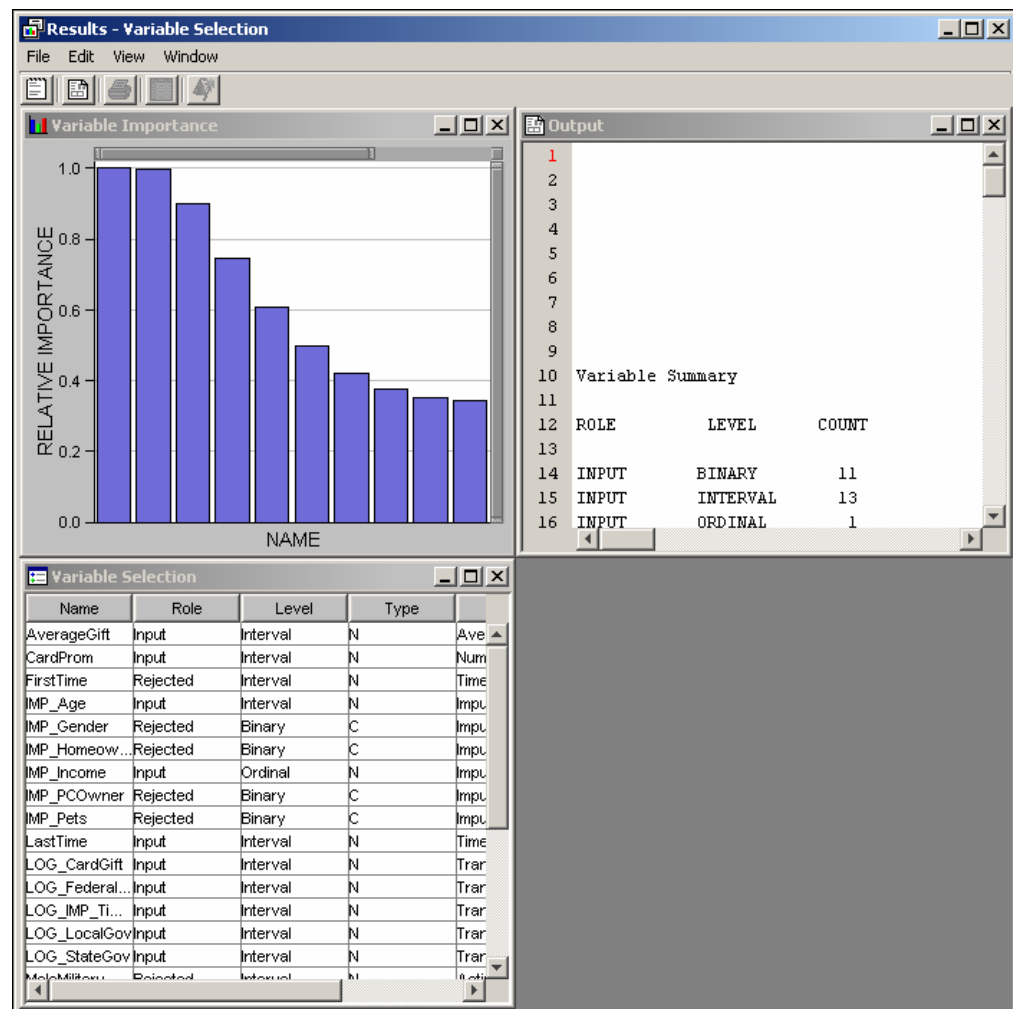
Property	Value
Node ID	Varsel
Imported Data	...
Variables	...
Max Class Level	100
Max Missing Percentag	50
Target Model	Chi-Square
Hide Rejected Variable	Yes
Reject Unused Variable	Yes
Chi-Square Options	
Number of Bins	50
Maximum Pass Number	6
Minimum Chi-Square	3.84

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

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

3. Retain the default settings and run the flow from the Variable Selection node.

4. View the results.



- Examine the Variable Selection window. Click on the **Role** column heading to sort the variable by their assigned roles. Then click on the **Comment** column heading. Inspect the results.

Variable Selection					
Name	Role	Level	Type	Label	Comment
AverageGift	Input	Interval	N	Average Gift	.
CardProm	Input	Interval	N	Number of Card Promotions	.
IMP_Age	Input	Interval	N	Imputed: Donor's Age	.
IMP_Income	Input	Ordinal	N	Imputed: Income Level	.
LastTime	Input	Interval	N	Time Since Last Donation	.
LOG_CardGift	Input	Interval	N	Transformed: Gift to Card Promotions	.
LOG_Federal...	Input	Interval	N	Transformed: Federal Government (% ...	.
LOG_IMP_Ti...	Input	Interval	N	Transformed: Imputed: Time between D...	.
LOG_LocalGov	Input	Interval	N	Transformed: Local Government (% of ...	.
LOG_StateGov	Input	Interval	N	Transformed: State Government (% of ...	.
MaleVeterans	Input	Interval	N	Male Veterans (% of households)	.
M_Gender	Input	Binary	N	Imputation Indicator for Gender	.
M_Homeowner	Input	Binary	N	Imputation Indicator for Homeowner	.
TotProm	Input	Interval	N	Total Number of Promotions	.
TargetD	Rejected	Interval	N	Amount Donated	.
FirstTime	Rejected	Interval	N	Time Since First Donation	Varsel: Small Chi-square value
IMP_Gender	Rejected	Binary	C	Imputed Gender	Varsel: Small Chi-square value
IMP_Homeow...	Rejected	Binary	C	Imputed Homeowner	Varsel: Small Chi-square value
IMP_PCOwner	Rejected	Binary	C	Imputed: Personal Computer Owner	Varsel: Small Chi-square value
IMP_Pets	Rejected	Binary	C	Imputed: Pet(s) in Household	Varsel: Small Chi-square value
MaleMilitary	Rejected	Interval	N	Active Duty Military Males (% of house...	Varsel: Small Chi-square value
M_Age	Rejected	Binary	N	Imputation Indicator for Age	Varsel: Small Chi-square value
M_Income	Rejected	Binary	N	Imputation Indicator for Income	Varsel: Small Chi-square value
M_PCOwner	Rejected	Binary	N	Imputation Indicator for PCOwner	Varsel: Small Chi-square value
M_Pets	Rejected	Binary	N	Imputation Indicator for Pets	Varsel: Small Chi-square value
M_TimeLag	Rejected	Binary	N	Imputation Indicator for TimeLag	Varsel: Small Chi-square value

Fourteen variables have been retained as input variables.

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



# Chapter 5 Predictive Modeling Using Neural Networks

<b>5.1</b>	<b>Introduction to Neural Networks .....</b>	<b>5-3</b>
<b>5.2</b>	<b>Visualizing Neural Networks .....</b>	<b>5-9</b>
<b>5.3</b>	<b>Exercises .....</b>	<b>5-22</b>
<b>5.4</b>	<b>Solutions to Exercises .....</b>	<b>5-23</b>

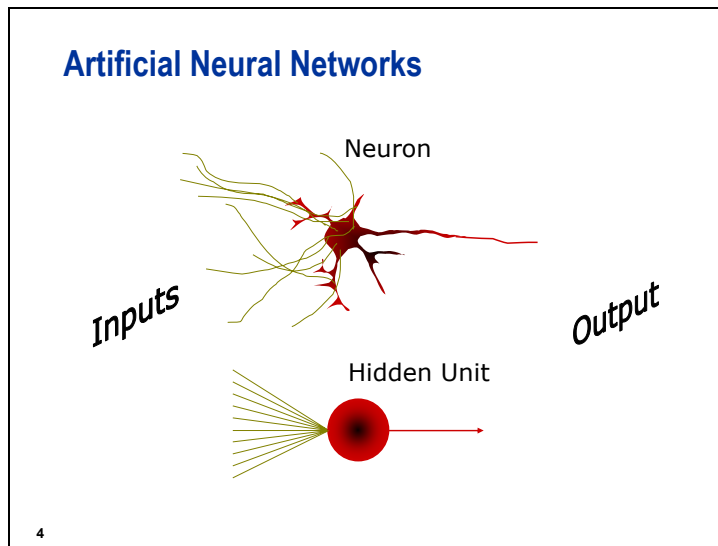




## 5.1 Introduction to Neural Networks

### Objectives

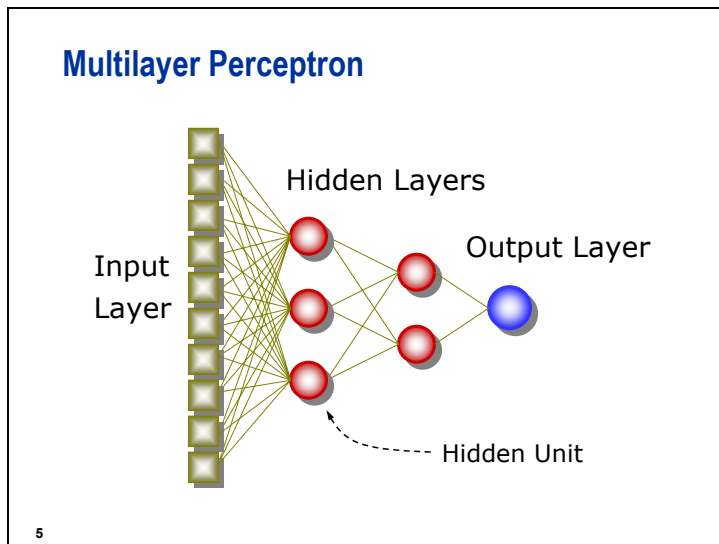
- Define a neural network.
- List the components of a neural network.
- Define an activation function.
- Discuss the concept of an optimization algorithm.



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

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

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



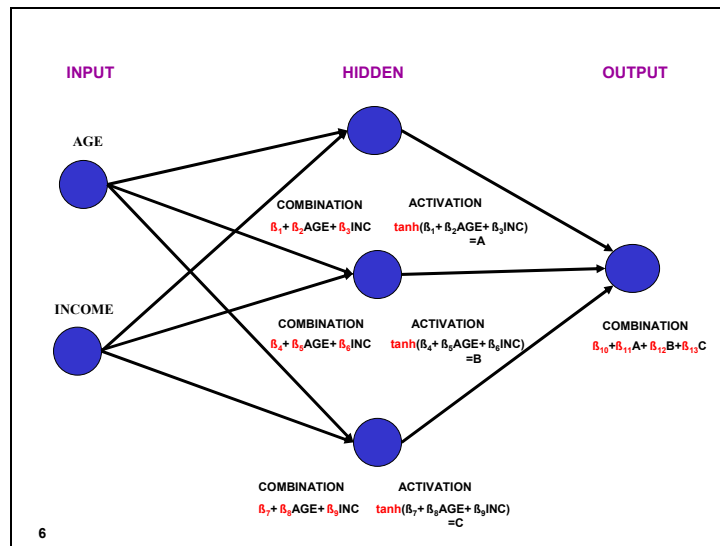
An *artificial* neural network is a flexible framework for specifying a variety of models. The most widely used type of neural network in data analysis is the *multilayer perceptron* (MLP). An MLP is a feed-forward network composed of an input layer, hidden layers composed of hidden units, and an output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

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

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

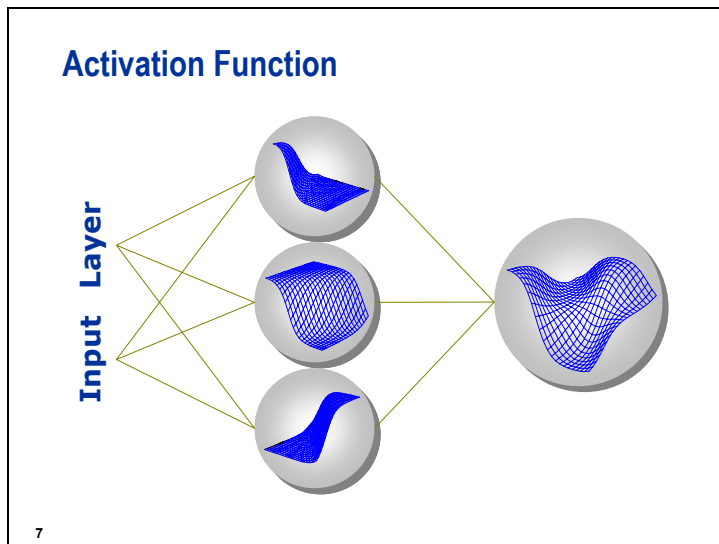
The output layer has units corresponding to the target. With multiple target variables or multiclass ( $>2$ ) targets, there are multiple output units.

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

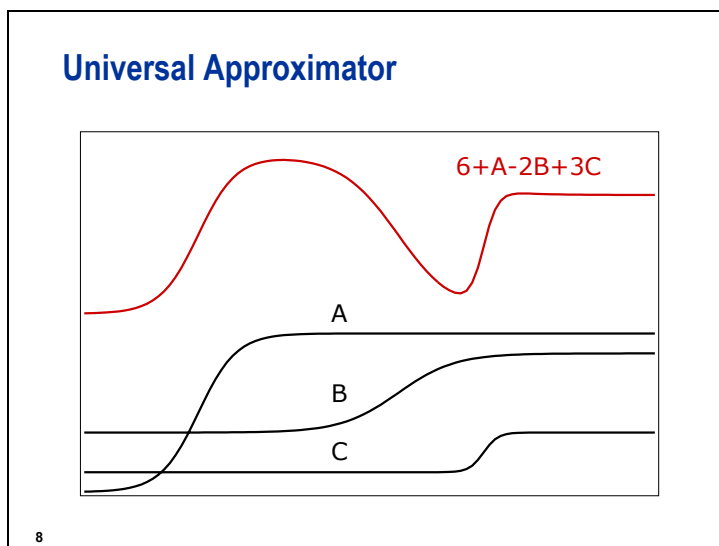


Each hidden unit outputs a nonlinear transformation of a linear combination of their inputs. The linear combination is the net input. The nonlinear transformation is the activation function. The activation functions used with MLPs are sigmoidal curves (surfaces).

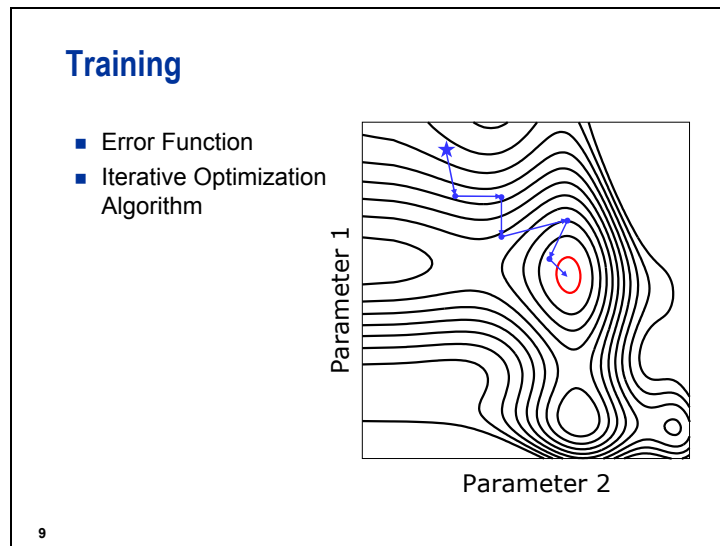
A hidden layer can be thought of as a new (usually) lower-dimensional space that is a nonlinear combination of the previous layer. The output from the hidden units is linearly combined to form the input of the next layer. The combination of nonlinear surfaces gives MLPs their modeling flexibility.



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



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



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

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

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

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

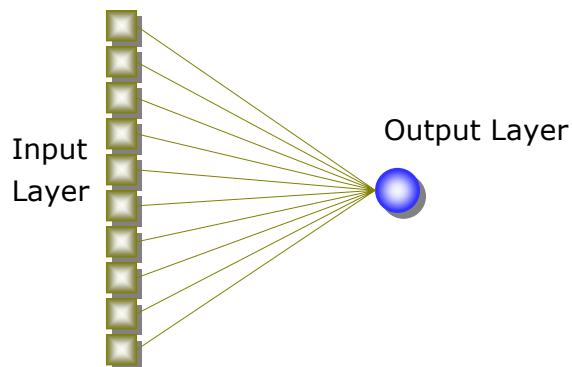
## 5.2 Visualizing Neural Networks

### Objectives

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

11

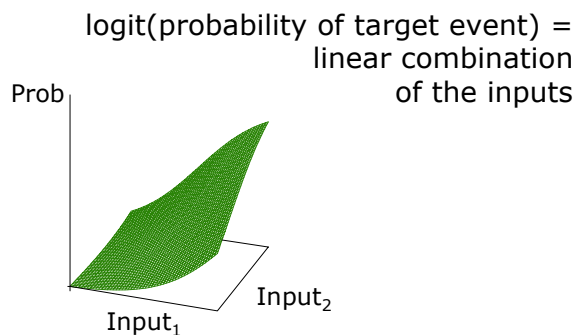
### Generalized Linear Models



12

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

### Logistic Regression/Discrimination



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

### The Scenario

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



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

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

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



## Fitting a Neural Network Model

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

### Defining a New Data Source and Building the Initial Flow

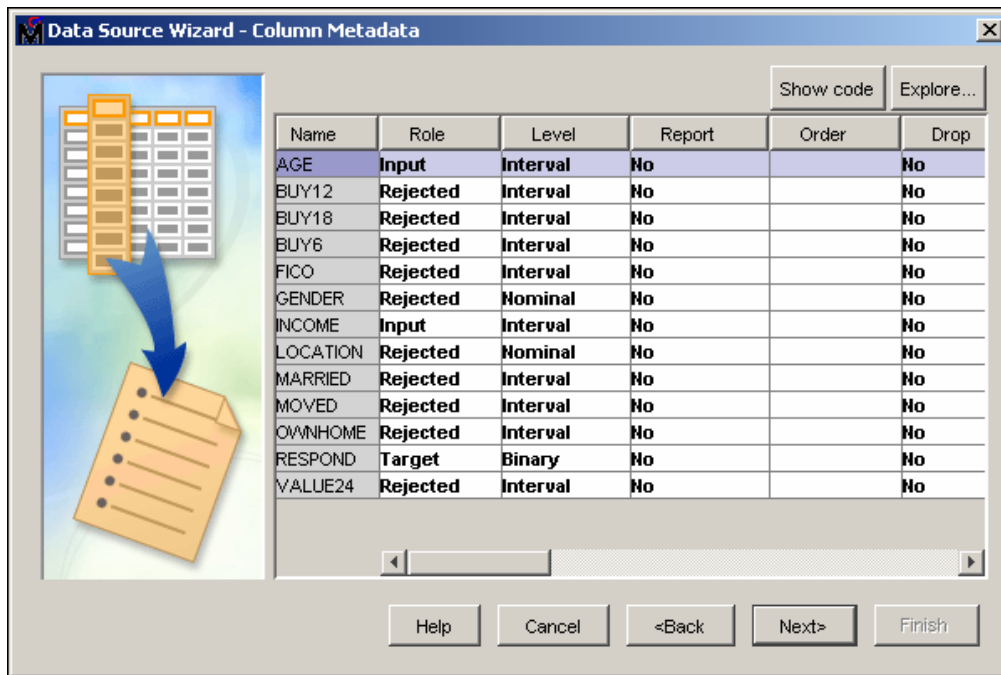
1. To insert a new diagram in the course project, select **File** ⇒ **New** ⇒ **Diagram....**
2. Name the new diagram **Neural Network**, and then select **OK**.
3. Define a new data source for the project by right-clicking on **Data Sources** in the Project Panel and selecting **Create Data Source**.
4. In the Data Source Wizard – Metadata Source window, be sure **SAS Table** is selected as the source and select **Next>**.
5. To choose the desired data table, select **Browse....**
6. Double-click on the **ADMT** library to see the data tables in the library.
7. Select the **BUY** data set, and then select **OK**.
8. Select **Next>**.

Property	Value
Table Name	ADMT.BUY
Description	
Member Type	DATA
Data Set Type	DATA
Engine	V9
Number of Variables	13
Number of Observations	10000
Created Date	2004-01-19 16:34:32.001
Modified Date	2004-01-19 16:34:32.001

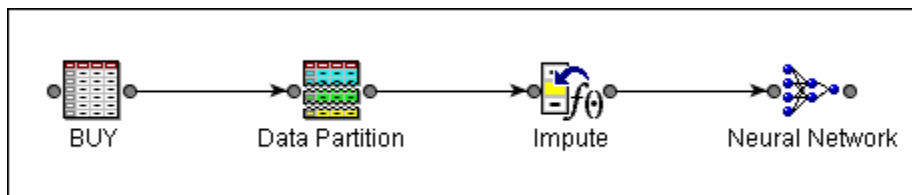
Observe that this data set has 10,000 observations (rows) and 13 variables (columns).

9. Select **Next>**.

10. Retain the Basic advisor option and select **Next>**.
11. Set the role of **RESPOND** to **Target**.
12. Notice that all numeric variables are given the Level Interval when using the Basic advisor. Change the level of **RESPOND** to **Binary**.
13. Control-click to select all of the other variables except **AGE** and **INCOME**. Change their role to **Rejected**. The role of **AGE** and **INCOME** should be **Input**.



14. Select **Next>**.
15. Select **Next>**.
16. Select **Finish**.
17. Assemble the diagram shown below.



### Partition the Data

1. To partition the data, select the Data Partition node in the diagram.
2. In the Property Panel, change the data set percentages for Training to **70** and for Test to **0**. Leave the percentage for Validation at 30.



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

### Construct the Multilayer Perceptron

1. Select the Neural Network node.
2. To see additional options, select **View** ⇒ **Property Sheet** ⇒ **Advanced**.
3. Examine the general properties of the node.

Property	Value
Node ID	Neural
Imported Data	...
Variables	...
Use Current Estimates	No
Architecture	MLP
Direct Connection	No
Model Selection Criterion	Profit/Loss
Number of Hidden Units	3

In the `Use Current Estimates` field you specify whether you want to use the current estimates as the starting values for training. When you set this property to Yes, the estimates from the previous run of the node will be used as the initial values for training. To use this property, an estimates data set must have been created by the node before you set this property to Yes. The default value of the property is No.

The `Architecture` field enables you to specify a wide variety of neural networks including

- Generalized linear model (GLIM)
- Multilayer perceptron (MLP, which is the default)
- Ordinary radial basis function with equal widths (ORBFEQ)
- Ordinary radial basis function with unequal widths (ORBFUN)
- Normalized radial basis function with equal heights (NRBFEH)
- Normalized radial basis function with equal volumes (NRBFEV)
- Normalized radial basis function with equal widths (NRBFEW)
- Normalized radial basis function with equal widths and heights (NRBFEQ)
- Normalized radial basis function with unequal widths and heights (NRBFUN).

The User option in the field enables the user to define a network with a single hidden layer.



Discussion of these architectures is beyond the scope of this course.

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

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

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

The Number of Hidden Units property enables you to specify the number of hidden units that you want to use in each hidden layer. Permissible values are integers between 2 and 64. The default value is 3.

#### 4. Examine the Training Options properties of the node.

Training Options	
Maximum Iterations	20
Maximum Time	4 Hours
Training Technique	Default

The training options include

- the maximum number of iterations allowed during the neural network training. The permissible values are integers from 1 to 500. The default value is 20.
- the maximum time the maximum amount of CPU time that you want to use during training. Permissible values are 5 minutes, 10 minutes, 30 minutes, 1 hour, 2 hours, 4 hours, or 7 hours. The default setting for the Maximum Time property is 1 hour.
- the training technique is the methodology used to iterate from the starting values to a solution.

#### 5. Examine the Preliminary Training Options properties of the node.

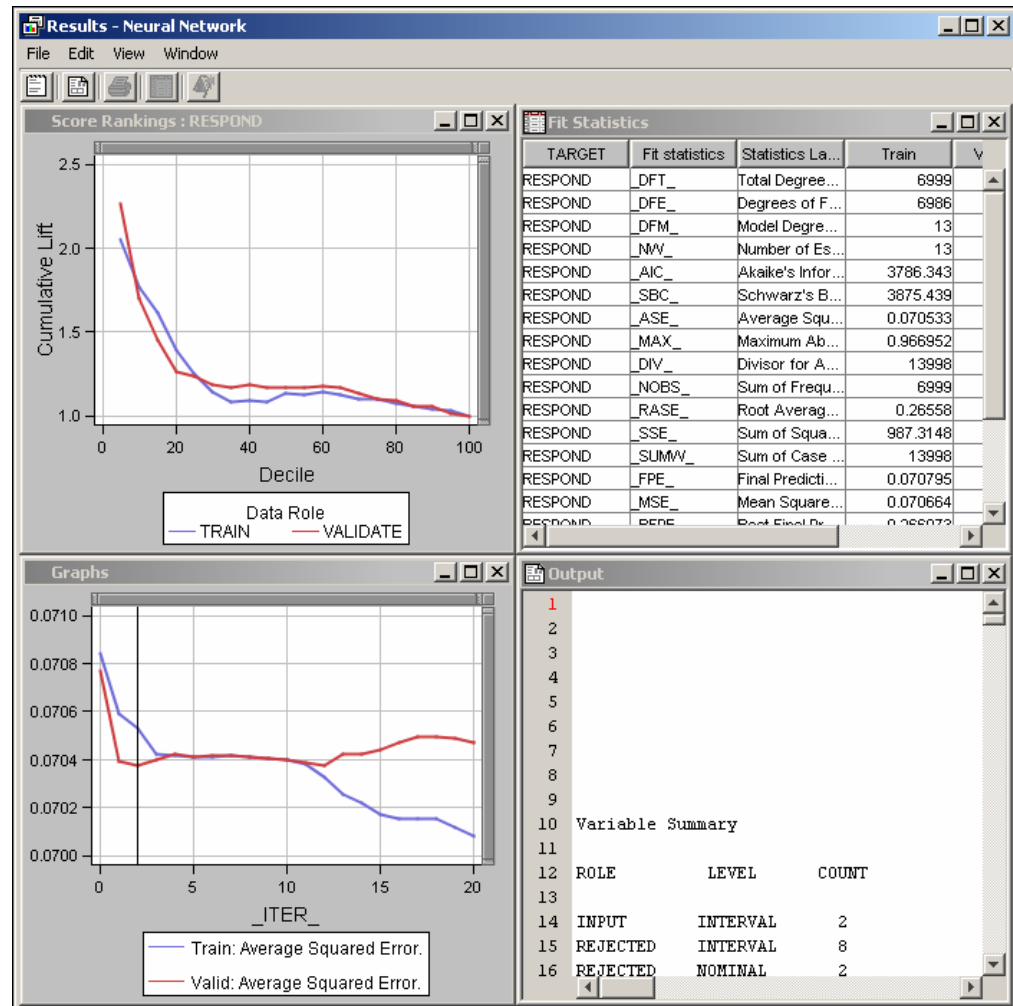
Preliminary Training Options	
Preliminary Training	No
Maximum Iterations	10
Maximum Time	1 Hour
Number of Runs	5

Preliminary training performs preliminary runs to determine parameter starting values. The default is No. If preliminary training is set to Yes, the options available include

- the maximum iterations
- the maximum time
- the number of runs.

## Examine the Model

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

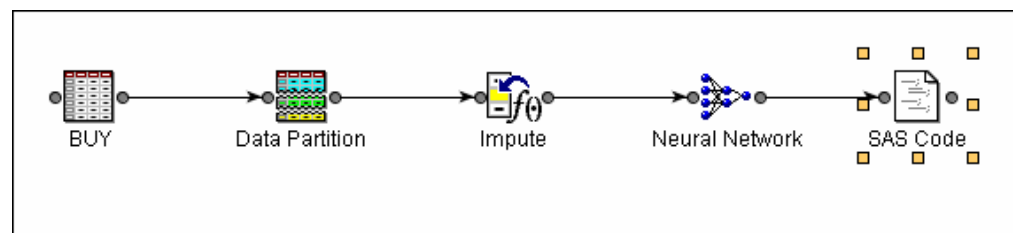


The Graphs window shows that the final neural network chosen is from the second iteration.


## Visualize the Model

You can use a SAS Code node to visualize the surface of this neural network.

1. Add a SAS Code node to the diagram.



2. Select the SAS Code node and examine the Property Panel.

3. Select  from the SAS Code property.
4. Right-click in the SAS Code window and select **File** ⇒ **Open**.
5. Select **neural network surface plot.sas**.
6. Select **Open**.



To view this plot you must have a browser and ActiveX plug-in installed on your client computer.

The code first registers an HTML file with SAS Enterprise Miner.

```
%em_register(key=myplot,type=FILE,extension=HTM) ;
```

Next, a call to the %EM\_ODSLISTON utility macro is executed. This macro opens an HTML output destination (procedure output is captured to the specified HTML file) and closes the listing destination (procedure output is not placed in the Output window). The macro variable &EM\_USER\_MYPLOT references the file registered in the first line of code.

```
%em_odsliston(file=&EM_USER_MYPLOT) ;
```

A graphics options statement resets all current graph settings and specifies that the output device is an ActiveX plug-in. The BORDER option draws a border around the output.

```
goptions device=ACTIVEX border;
```

The G3GRID procedure creates a rectangular grid of interpolated or smoothed values from the irregularly spaced observations in the scored training data set (&EM\_IMPORT\_DATA) for use in a three-dimensional surface or contour plot.

```
proc g3grid data=&EM_IMPORT_DATA out=surface;
  grid AGE*INCOME=P_RESPOND1;
run;
```

The G3D procedure creates a surface plot of the input variables versus the model predictions using the output data set from the G3GRID procedure.

```
proc g3d data=surface;
  plot AGE*INCOME=P_RESPOND1;
run;
quit;
```

The HTML destination is closed and the listing destination is restored.

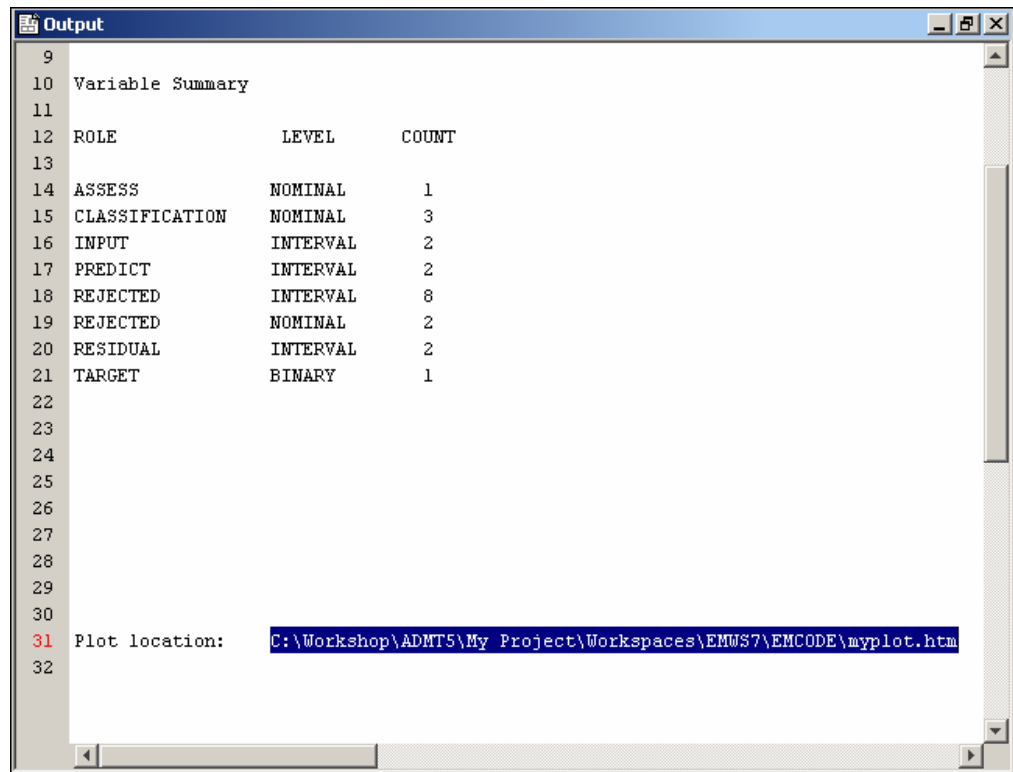
```
%em_odslistoff;
```

A DATA step outputs the location of the surface plot file to the Output window. By copying this location to the clipboard, you can view and interact with the created surface plot.

```
data _null_;
  file PRINT;
  put "Plot location:    &EM_USER_MYPLOT";
run;
```

7. Select **OK** to close the SAS Code window.

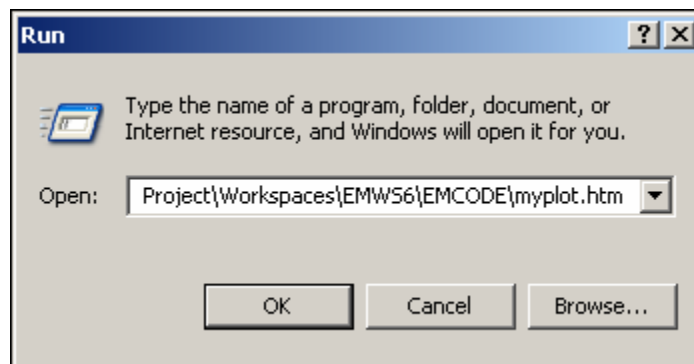
8. Run the SAS Code node and view the results.



9. Copy the plot location appearing in the Output window to the clipboard.

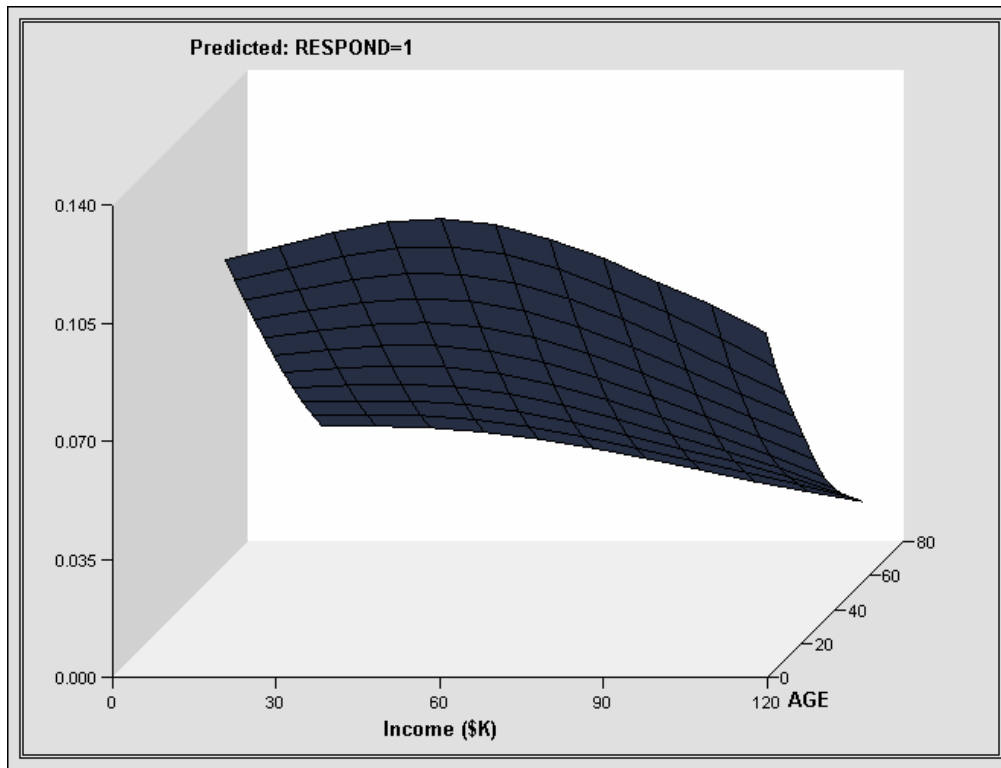
10. Select **Start** ⇒ **Run...** from the windows tool bar. The Run window opens.

11. Paste the plot location into the Open field.



12. Select **OK**. The surface plot opens in a browser window.



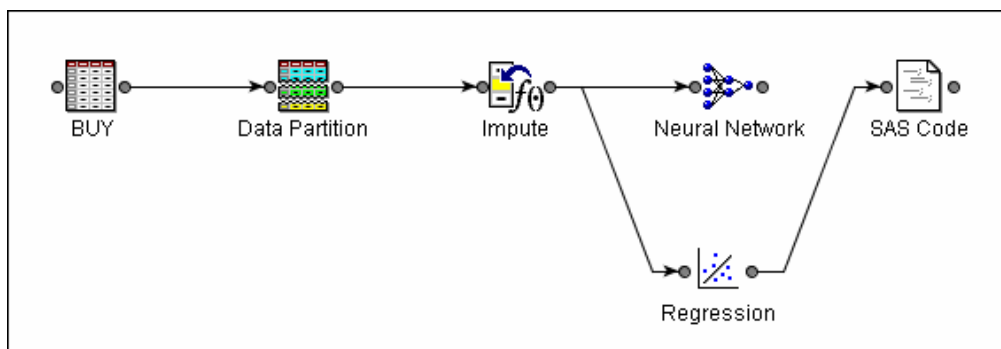


You can rotate this plot by holding the ALT key while clicking and dragging the pointer within the plot window.

### Visualizing Logistic Regression

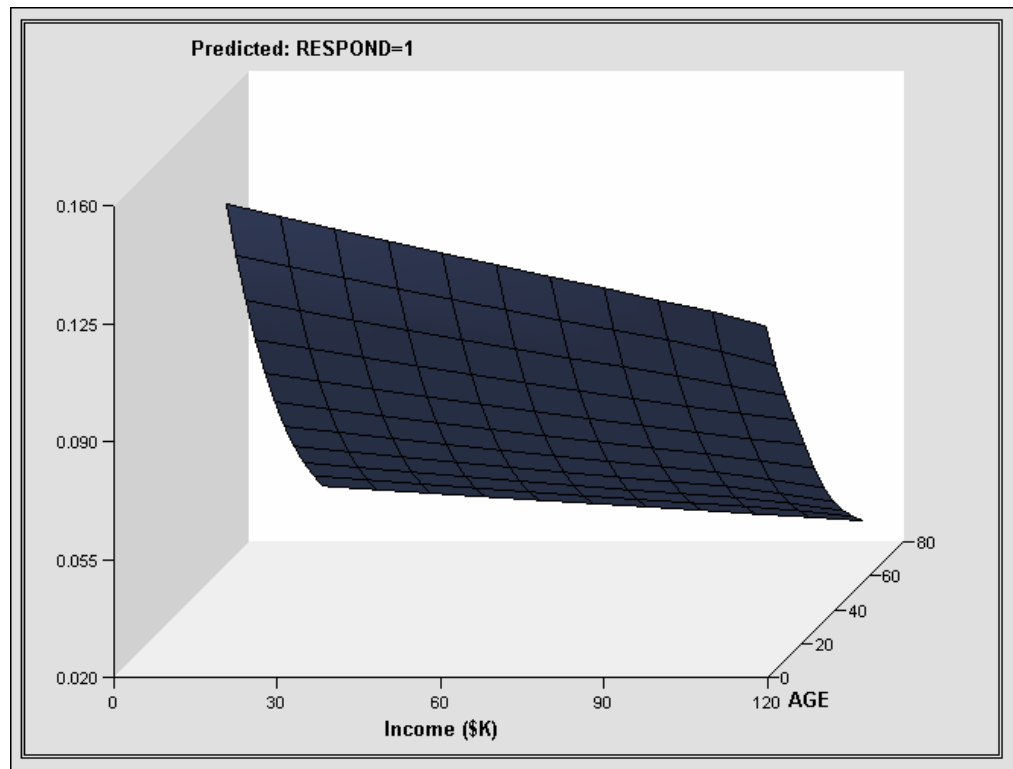
A standard logistic regression model is a multilayer perceptron with zero hidden layers and a logistic output activation function.

1. To visualize a fitted logistic regression surface, drag a Regression node onto the workspace.
2. Connect the Regression node to the SAS Code node and disconnect the neural network from the SAS Code node as shown below.



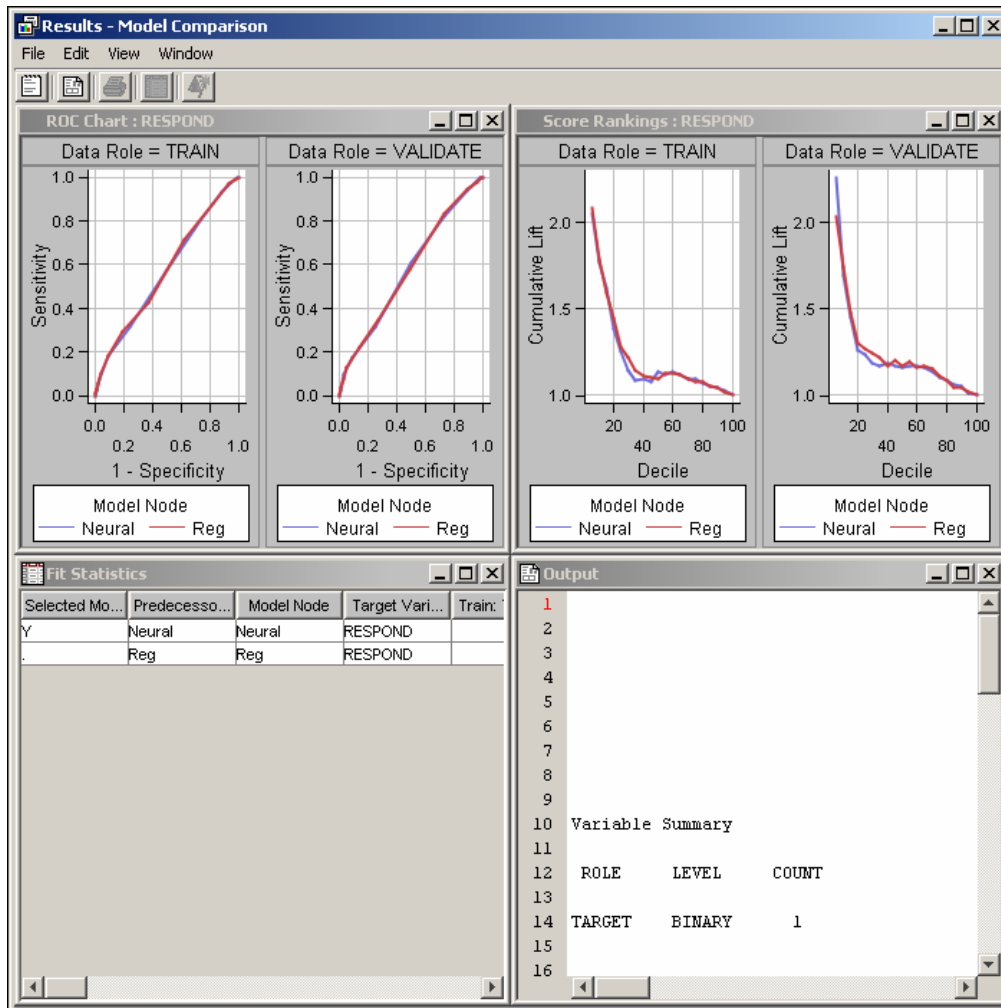
3. Run the diagram from the SAS Code node.
4. Select **Start** ⇒ **Run...** from the Windows tool bar. The Run window opens with the file location previously pasted.

5. Select **OK**. The browser window now displays the predictions of the regression.



The surface is not as complex as the surface generated as a result of the neural network model.

6. Add a Model Comparison node to the diagram and connect it to both the Neural Network node and the Regression node.
7. Run the flow from the Model Comparison node and view the results.



Based on the lift and ROC charts, there is not much difference in predictive capability of the two models.

## 5.3 Exercises

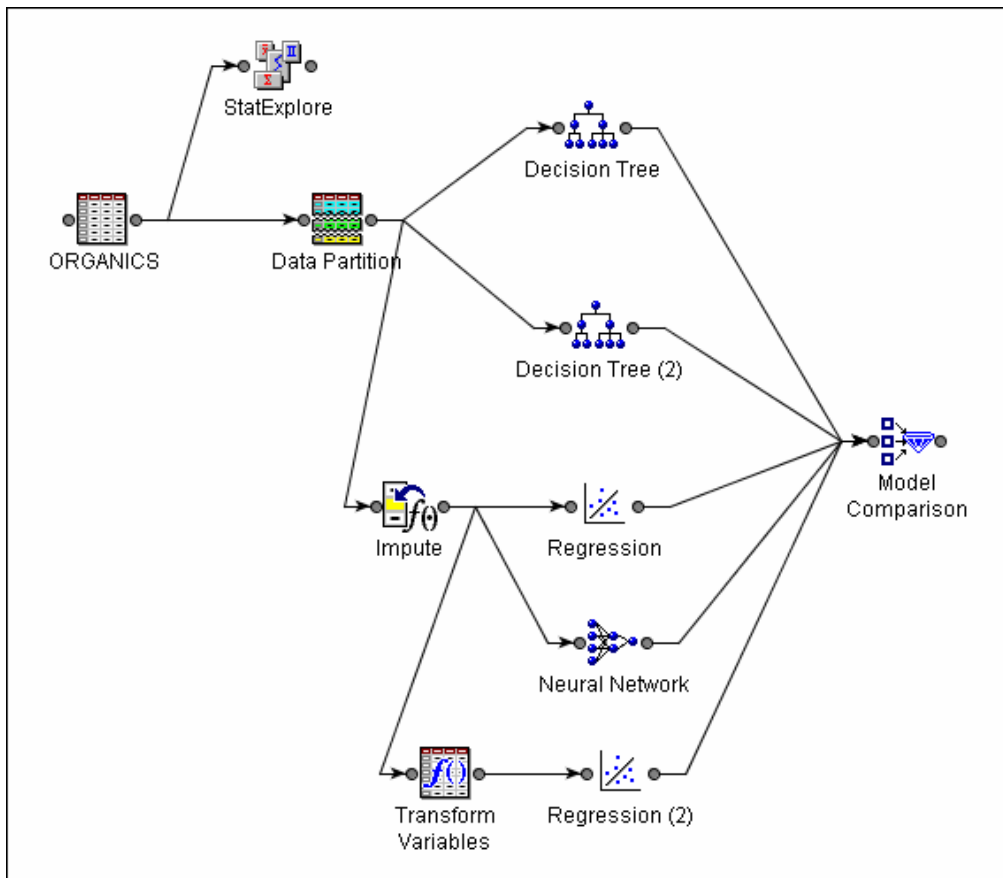
### 1. Predictive Modeling Using Neural Networks

- a. In preparation for a neural network model, is imputation of missing values needed? Why or why not?
- b. In preparation for a neural network model, is data transformation generally needed? Why or why not?
- c. Add a Neural Network node to the Organics diagram and connect it to the Impute node. Connect the Neural Network node to the Model Comparison node.
- d. Rerun the Model Comparison node and compare all of the models generated. Which model appears to be the best model?

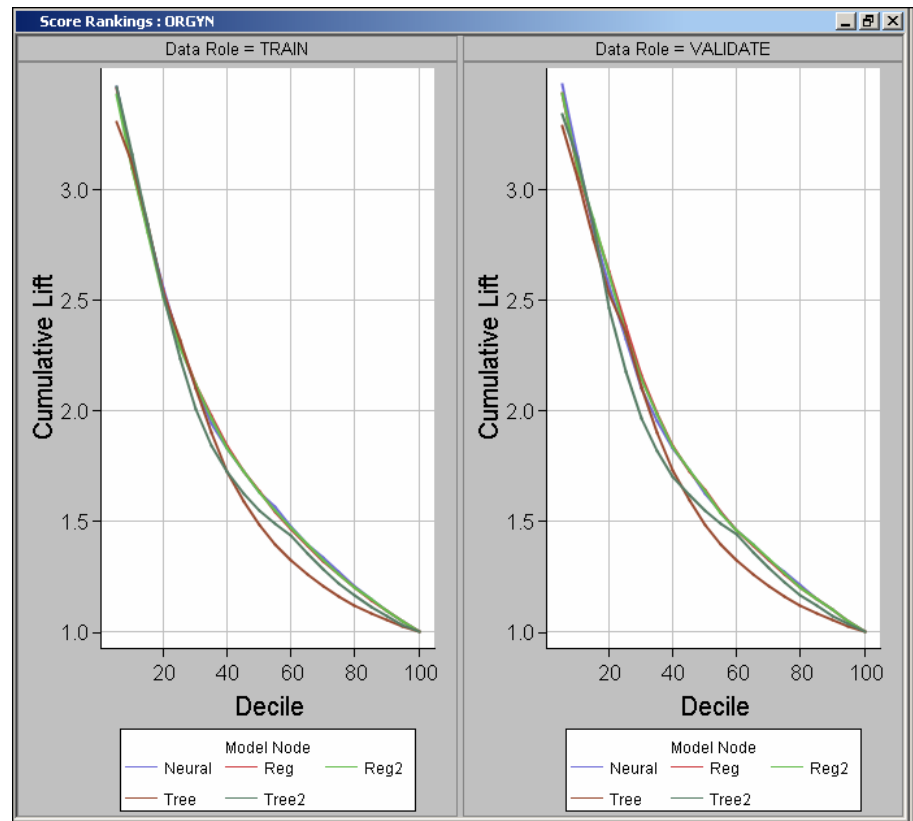
## 5.4 Solutions to Exercises

### 1. Predictive Modeling Using Neural Networks

- a. Imputation of missing values is needed prior to generating a neural network model because observations that have missing values for any of the variables cannot be used to train the model.
- b. Data transformation is not generally necessary prior to generating a neural network model because the model itself does transformations of the linear combinations of inputs.
- c. After adding the Neural Network node, the diagram should appear as shown.



- d. Rerun the Model Comparison and compare the models.
- 1) Right-click on the Model Comparison node and select **Run**.
  - 2) Select **Yes** to confirm that you want to run the path.
  - 3) After confirming the completion of the run, view the results by right-clicking on the Model Comparison node and selecting **Results...**.



Based upon the lift chart, the four models are similar in their predictive ability. The second decision tree appears to be slightly better than some of the other models.

# Chapter 6 Model Evaluation and Implementation

<b>6.1</b>	<b>Model Evaluation: Comparing Candidate Models .....</b>	<b>6-3</b>
<b>6.2</b>	<b>Ensemble Models .....</b>	<b>6-9</b>
<b>6.3</b>	<b>Model Implementation: Generating and Using Score Code .....</b>	<b>6-14</b>
<b>6.4</b>	<b>Exercises .....</b>	<b>6-25</b>
<b>6.5</b>	<b>Solutions to Exercises .....</b>	<b>6-26</b>





## 6.1 Model Evaluation: Comparing Candidate Models

### Objectives

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

3

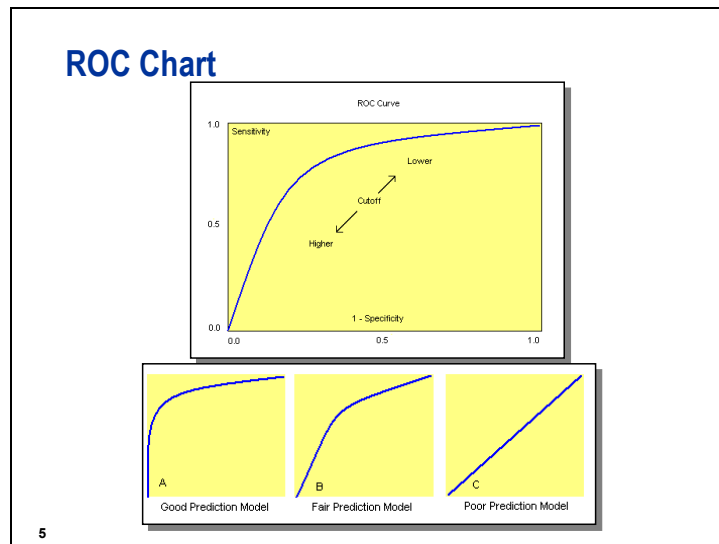
### Comparing Candidate Models

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

4

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

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



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

The extremes (0,0) and (1,1) represent cutoffs of 1.0 and 0.0 respectively. You could choose a rule that all observations with a predicted probability of 1 are classified as events. In other words, you would not solicit anyone in the nonprofit example. In that case none of the event observations are correctly classified, but all of the nonevent observations are correctly classified (sensitivity = 0, specificity = 1).

Alternatively, you could choose a rule that all observations with a predicted probability of 0 or higher are classified as events. In other words, you would solicit everyone in the nonprofit example. In that case all the events are correctly classified but none of the nonevents are correctly classified (sensitivity = 1, specificity = 0). The horizontal axis is 1 – specificity, so the ROC curve starts at the point (0,0) and ends at the point (1,1).

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

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

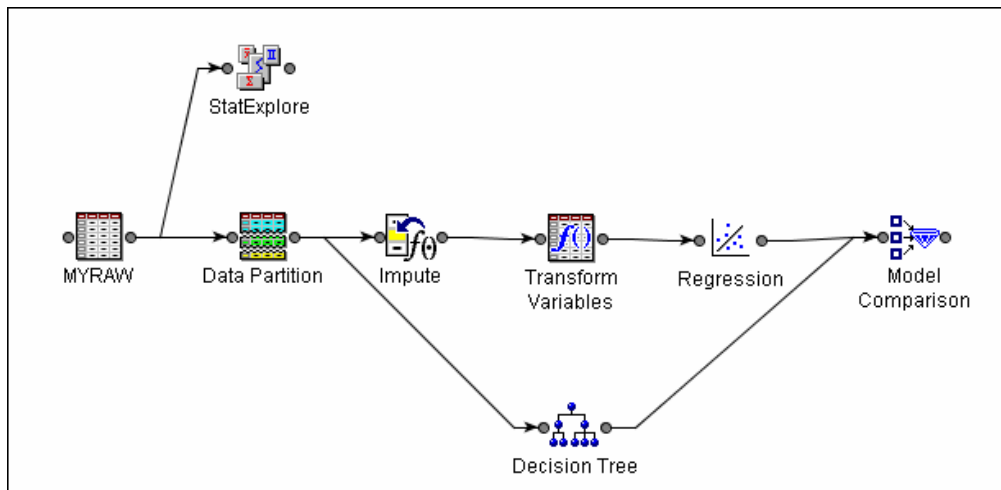


## Comparing Candidate Models

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

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

1. Open the Non-Profit diagram created earlier.
2. To unclutter the diagram workspace, delete the Variable Selection node and the Tree node used earlier for variable selection.

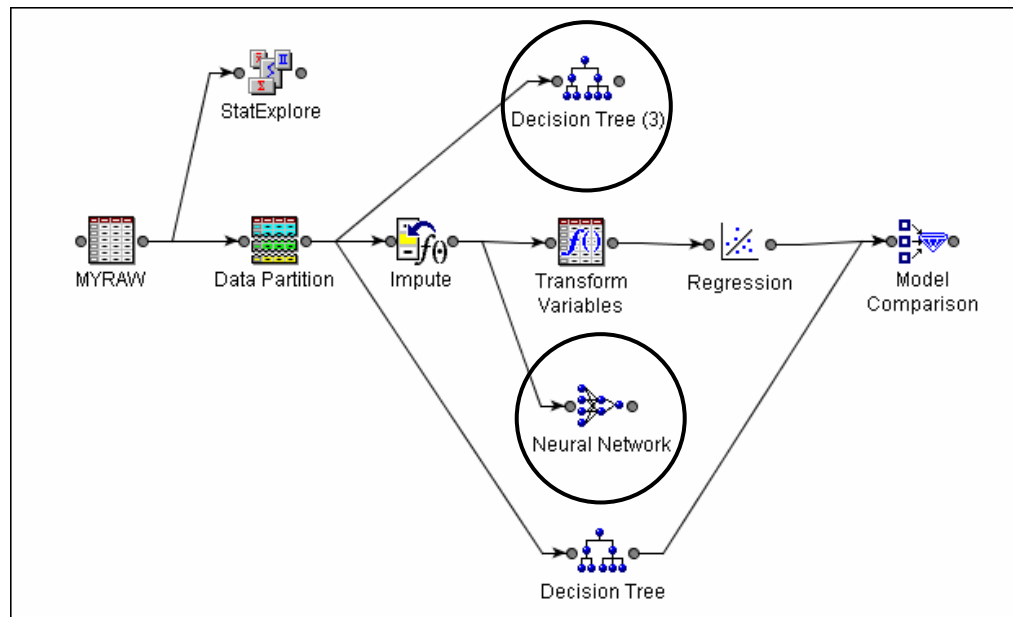


3. Add a Neural Network node and connect it to the Impute node.

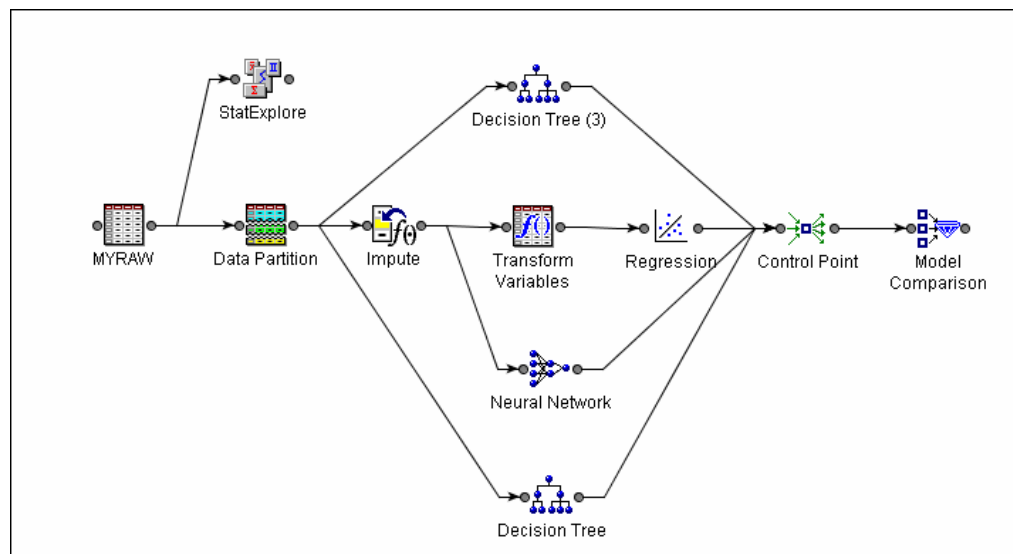


Because of the transformations that occur in a neural network model, the model is inherently flexible. As mentioned earlier, a multilayer perceptron with one hidden layer is a universal approximator. Therefore, when generating a neural network model, there is not the same necessity as with a linear regression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

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



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



The original decision tree created for this data was done with the SAS Enterprise Miner default settings. This allows for only 2-way splits. For the new tree model, allow up to 4-way splits.

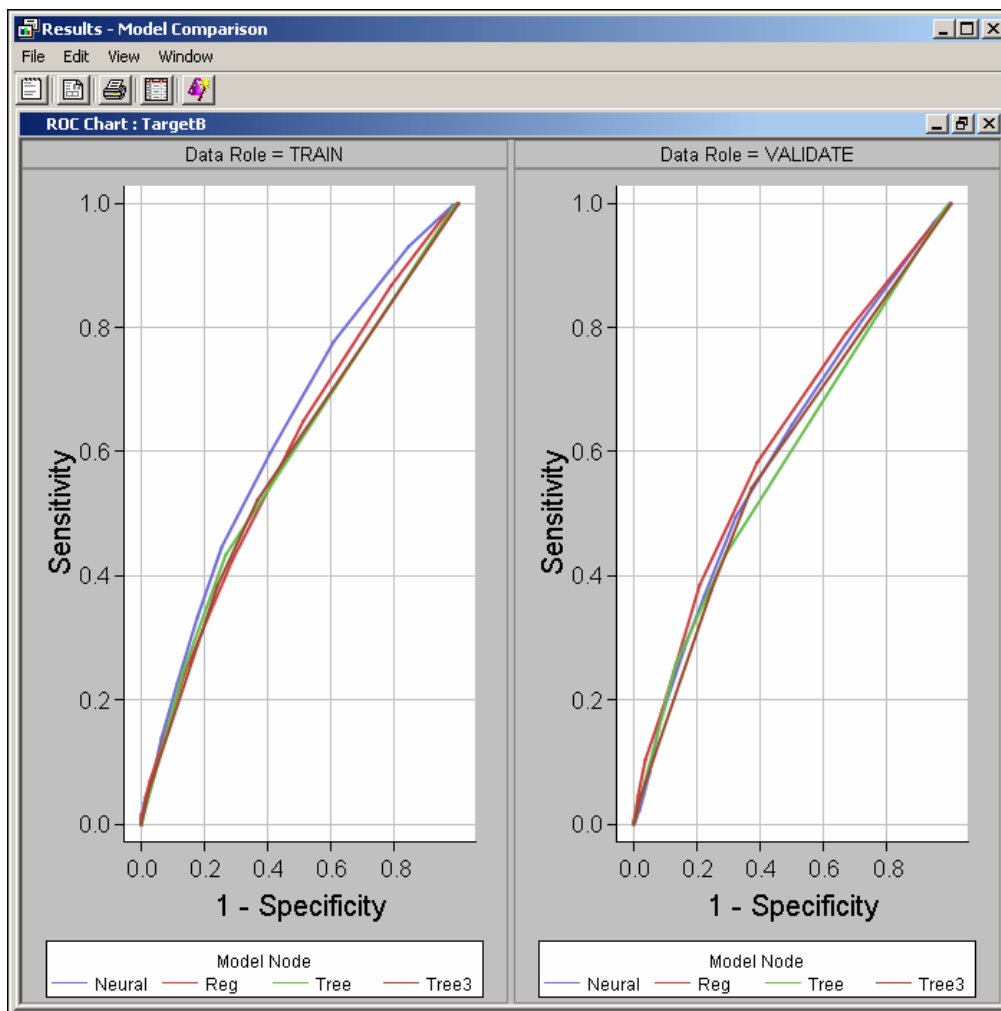
7. Select the new Tree node and examine the Property Panel.

8. To allow for up to 4-way splits, change the Maximum Branch property to **4**.
9. Change the name of the node in the diagram to **4-way Tree (3)**.
10. Because our primary purpose here is to compare models, leave the neural network node set with its defaults and run the diagram from the Model Comparison node.
11. View the results when the run is completed.

Examine the fit statistics window. This window shows that the regression model has been selected as the best model.

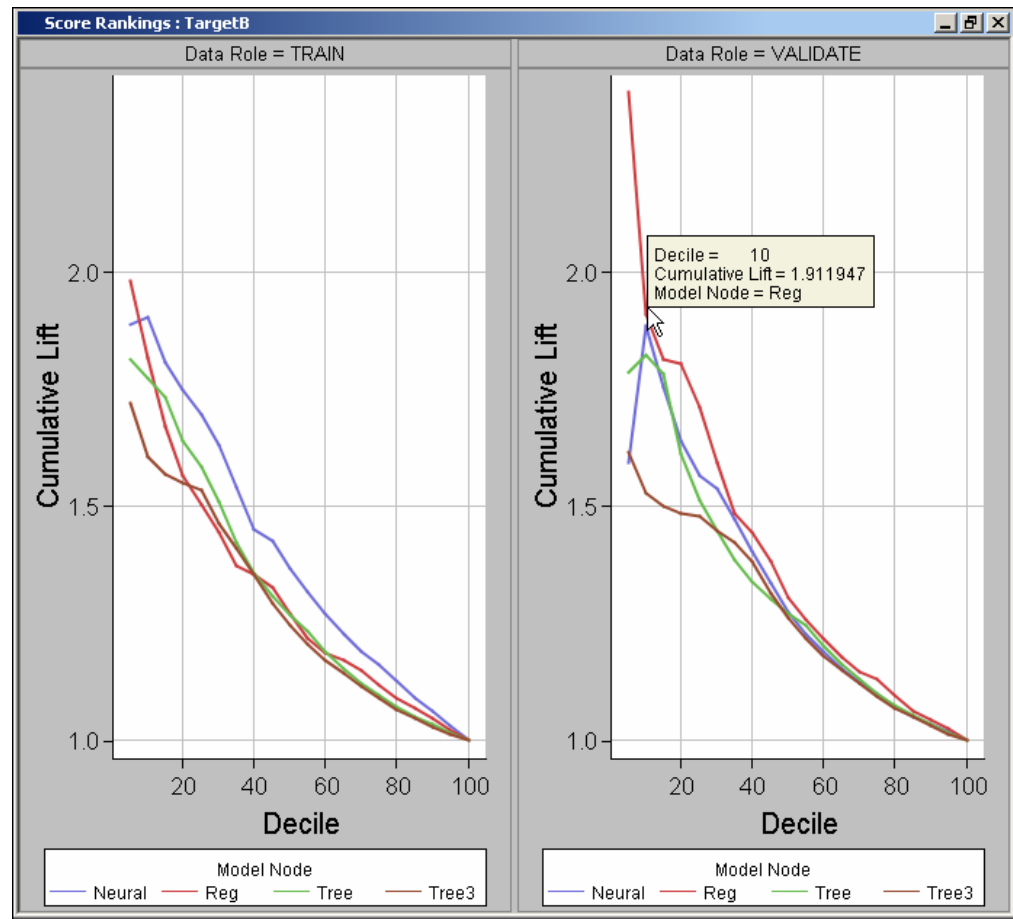
Fit Statistics				
Selected Mo...	Predecesso...	Model Node	Target Vari...	Train:
.	Neural	Neural	TargetB	
Y	Reg	Reg	TargetB	
.	Tree	Tree	TargetB	
.	Tree3	Tree3	TargetB	

Examine the ROC charts.



While the neural network model appears to have the best fit in the training data, when you examine the chart for the validation data, the regression model is the best model. While the neural network fits the training data the best, in this case it does not generalize to other data as well as the regression model.

Examine the cumulative lift charts.



Based on the cumulative lift chart, the stepwise regression model is the overall best model in this case, with a lift of almost 2 in the first decile.



If the relationship between the target and the inputs is linear, then a model with more flexibility than a linear regression is unnecessarily complex and may not provide as good a fit on new data.

12. Close the results to return to the workspace.

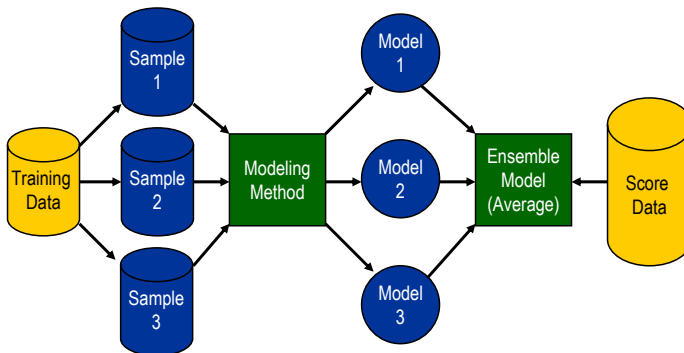
## 6.2 Ensemble Models

### Objectives

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

8

### Combined Ensemble Models



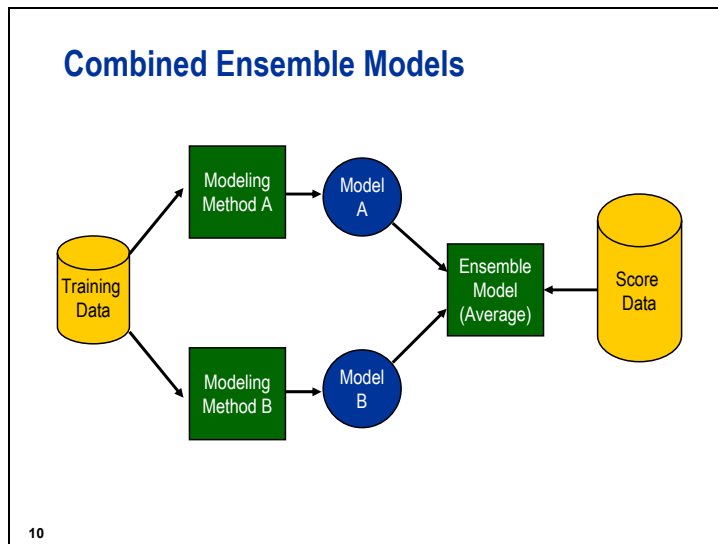
9

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

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



In order to combine three models from the same modeling method as shown here, you must use three separate modeling nodes in SAS Enterprise Miner.



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

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

### Other Types of Ensemble Models

- Bagging
- Boosting

11

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



Bagging and boosting are not yet supported in SAS Enterprise Miner 5.1.

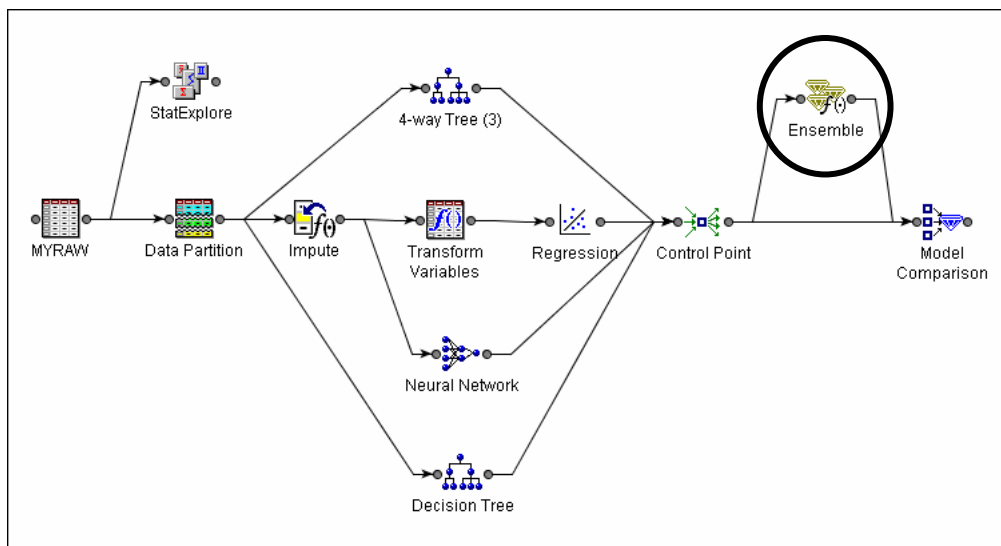




## Combined Ensemble Models

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

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



3. Select the Ensemble node and examine the Property Panel.

Property	Value
Node ID	Ensmbl
Imported Data	...
Variables	...
Interval Target	
Predicted Values	Average
Class Target	
Posterior Probabilities	Average
Voting Posterior Probabilities	Average
Status	
Last Error	
Last Status	
Needs Updating	Yes
Needs to Run	Yes
Time of Last Run	
Run Duration	

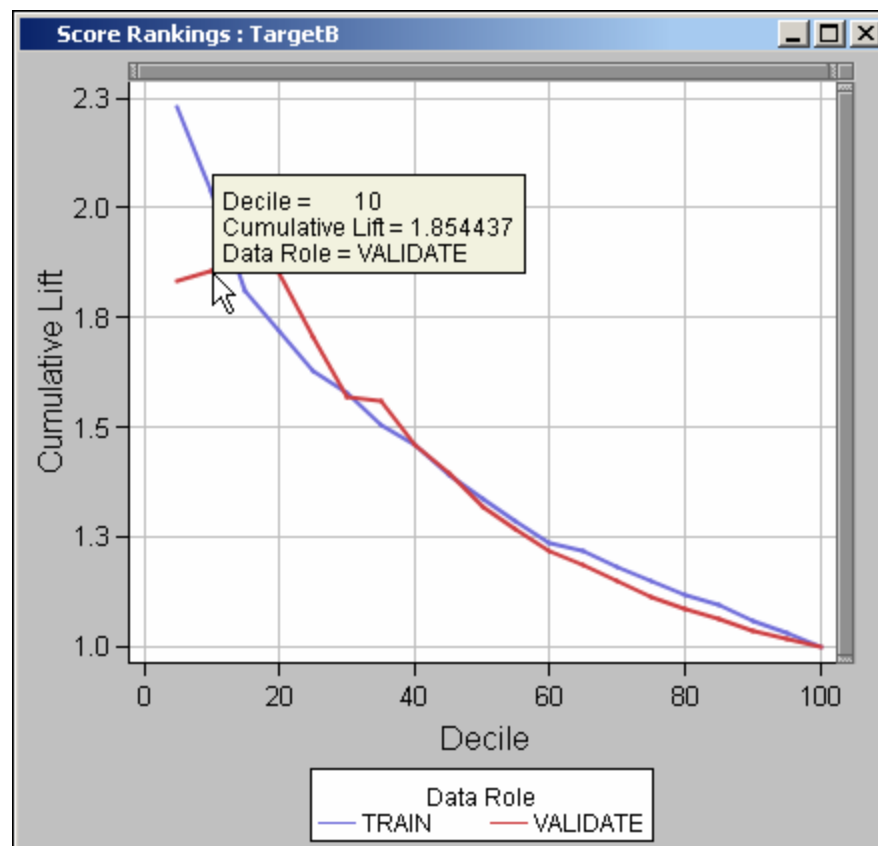
The properties differ depending on the type of target variable. If you have an interval target variable you can choose to combine models using the following functions:

- Average** takes the average of the predicted values from the different models as the prediction from the Ensemble node. This is the default method.
- Maximum** takes the maximum of the predicted values from the different models as the prediction from the Ensemble node.

If you have a class target variable you can choose to combine models using the following functions:

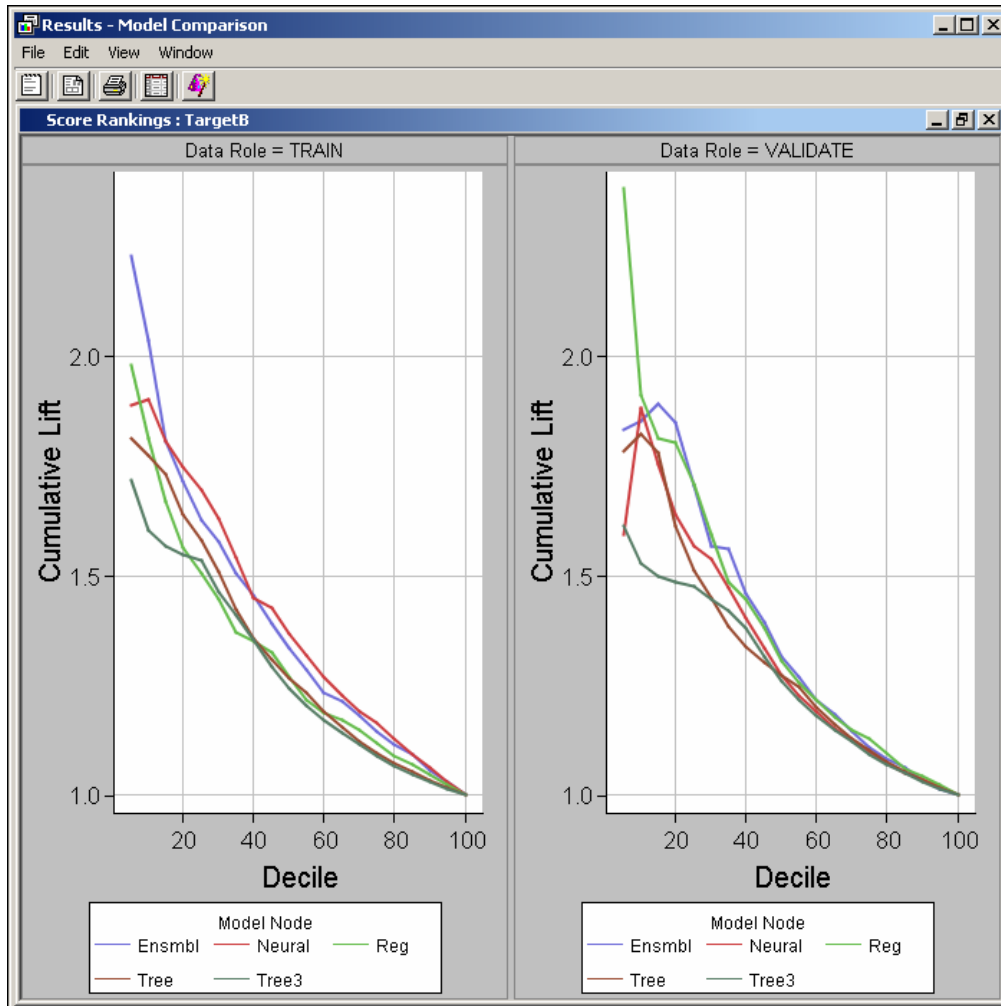
- Average** takes the average of the posterior probabilities from the different models as the prediction from the Ensemble node. This is the default method.
- Maximum** takes the maximum of the posterior probabilities from the different models as the prediction from the Ensemble node.
- Voting** There are two voting methods available, average and proportion. The average uses the posterior probabilities from the models that predict the event level of interest. Using this method any model that predicts a different event level is ignored. Using the proportion method, the posterior probabilities are ignored and the prediction from the Ensemble model is the proportion of models that predict the target level of interest. When the Posterior Probabilities property is set to voting, the Voting Posterior Probabilities property determines which of these two voting methods will be used.

4. Run the flow from the Ensemble node and view the results.



The cumulative lift from the Ensemble model at the first decile is approximately 1.85. This is not quite as good as what was seen earlier with the regression model.

5. To compare the Ensemble model on the same graph as the other models, rerun the Model Comparison node and view the results.



The results from the models are quite similar to each other, with no model much better than the others. In this case the ensemble model is not a big improvement over the best of the individual models, although the ensemble model is the selected model. In any case, choosing a model may depend upon your business use. You may have business rules or needs, which may determine which model is best.

6. Close the results and return to the workspace.

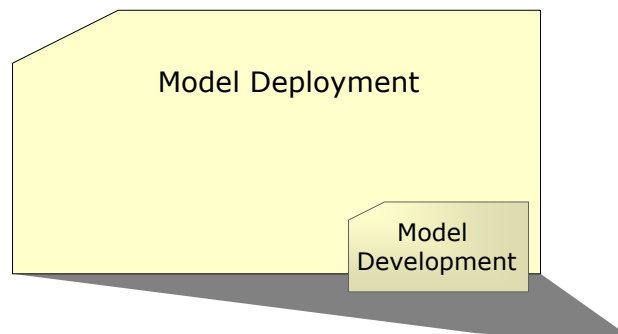
## 6.3 Model Implementation: Generating and Using Score Code

### Objectives

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

14

### Scoring



15

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

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

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

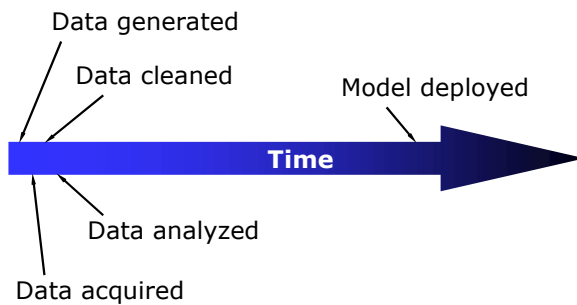
### Scoring Recipe

- |                            |                |
|----------------------------|----------------|
| ■ Model                    | ■ Scoring Code |
| ■ Formula                  | ≠ Scored data  |
| ■ Data Modifications       | ≠ Original     |
| ■ Derived inputs           | computation    |
| ■ Transformations          | algorithm      |
| ■ Missing value imputation |                |

16

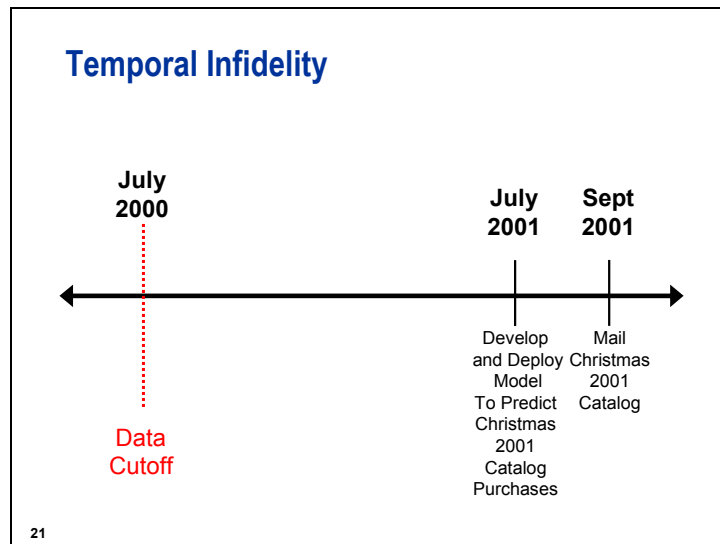
The score code must be more than just the model equation. It must incorporate all data manipulation tasks done before generating the model. In the SAS Enterprise Miner score code, this is done automatically.

### Population Drift



17

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



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

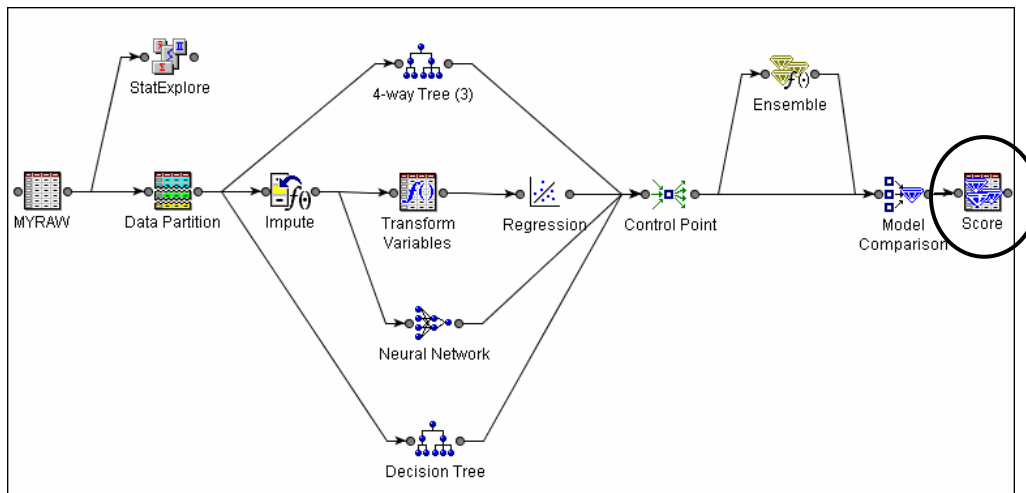
Another example of temporal infidelity is using intraoperative and postoperative information to predict surgical outcomes, when the purpose of the model is to predict patient outcomes preoperatively.



## Generating and Using Score Code

The Score node can be used to manage, edit, export, and execute scoring code from different models. It generates and manages scoring formulas in the form of a single SAS DATA step, which can be used in most SAS environments, with or without the presence of SAS Enterprise Miner. The scoring code can also create scoring code in C, Predictive Model Markup Language (PMML), and Java.

1. Add a Score node to the diagram and connect it to the Model Comparison node.



2. Select the Score node in the diagram.
3. Select **View** ⇒ **Property Sheet** ⇒ **Advanced**.

Examine the properties of the node.

Property	Value
Node ID	Score
Imported Data	...
Variables	...
Type of Scored Data	View
Use Fixed Output Names	Yes
Hide Variables	No

The Score node general properties include the following options:

Type of Scored Data	you can specify whether a data set or a data view will be scored. The default is a view.
Use Fixed Output Names	determines if score code will be generated to map the output variables to pre-specified output names. For example, EM_DECISION and EM_EVENTPROBABILITY contain the predicted classification and the probability of that classification, respectively.
Hide Variables	specifies if the original variables should be hidden in the exported metadata when the data set is scored. The default value is No.

Hide Selection	
Input	Yes
Target	Yes
Rejected	Yes
Frequency	Yes
Assess	Yes
Predict	Yes
Residual	Yes
Classification	Yes
Other	Yes

The Hide Selection properties are used when you set the Hide Variables property to Yes. These properties allow you to override the Hide Variables property for specific types of variables that will be hidden.

Score Data	
Validation	No
Test	No

The Score Data properties determine whether or not the validation and test data sets are scored.

Score Code Conversion	
C Score	Yes
Java Score	Yes

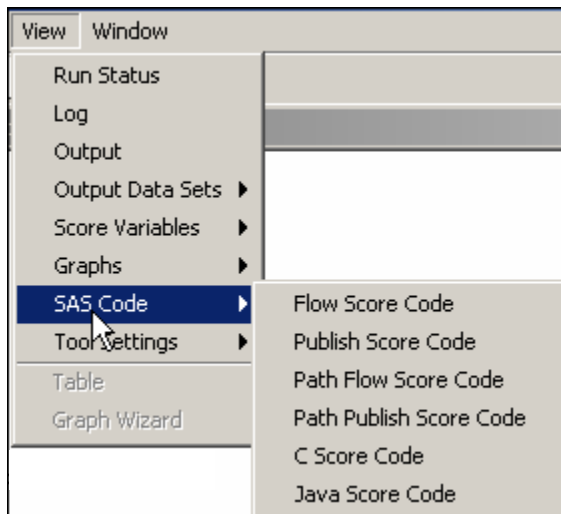
The Score Code Conversion properties specify if C score code and Java score code will be generated. If you want these score codes suppressed, set these properties to No.

4. Leave all of the default settings and run the flow from the Score node.
5. View the results after the run has completed.

The Output window of the results provides some summary statistics, which will be of more interest after you have scored a data set.



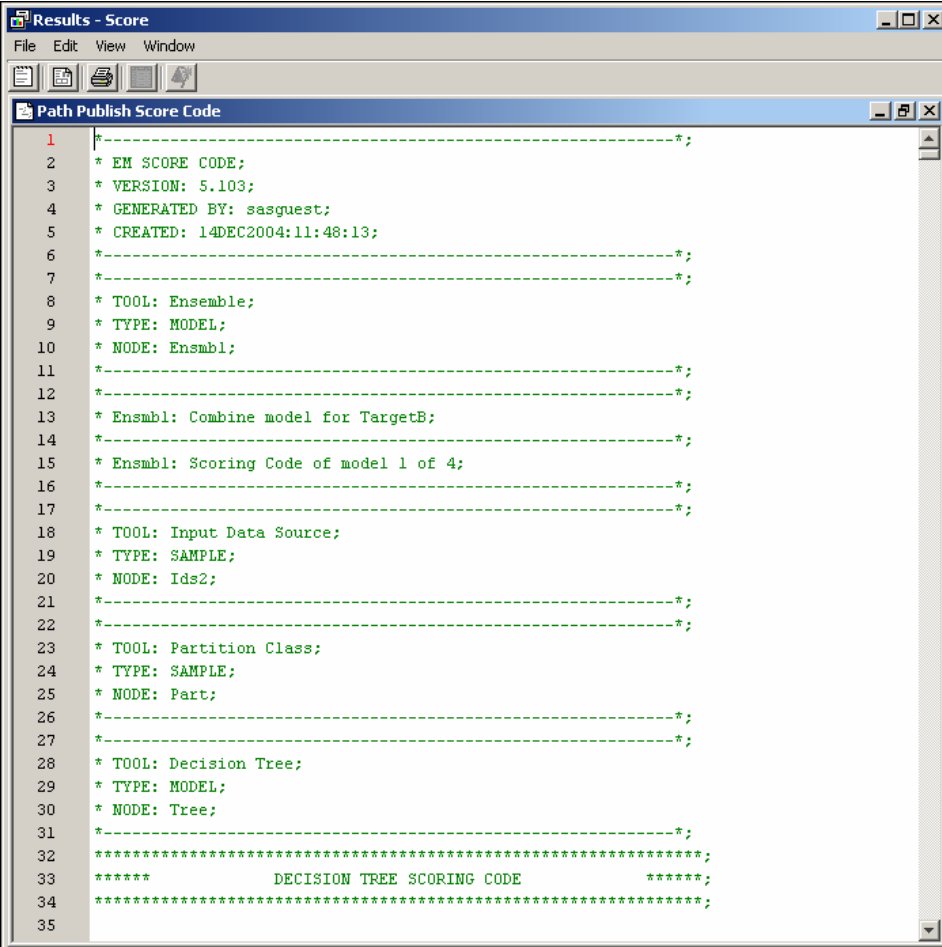
6. Select **View** ⇒ **SAS Code**.



There are six options for score code. The flow and publish score codes contain only the SAS code created by the score node. The path flow and path publish score codes contain the SAS score code for the entire process flow for the model.



The publish score codes do not include code to generate residual variables.

7. Select **View** ⇒ **SAS Code** ⇒ **Path Publish Score Code**.


```

1  *-----*;
2  * EM SCORE CODE;
3  * VERSION: 5.103;
4  * GENERATED BY: sasquest;
5  * CREATED: 14DEC2004:11:48:13;
6  *-----*;
7  *-----*;
8  * TOOL: Ensemble;
9  * TYPE: MODEL;
10 * NODE: Ensembl;
11 *-----*;
12 *-----*;
13 * Ensembl: Combine model for TargetB;
14 *-----*;
15 * Ensembl: Scoring Code of model 1 of 4;
16 *-----*;
17 *-----*;
18 * TOOL: Input Data Source;
19 * TYPE: SAMPLE;
20 * NODE: Ids2;
21 *-----*;
22 *-----*;
23 * TOOL: Partition Class;
24 * TYPE: SAMPLE;
25 * NODE: Part;
26 *-----*;
27 *-----*;
28 * TOOL: Decision Tree;
29 * TYPE: MODEL;
30 * NODE: Tree;
31 *-----*;
32 *****;
33 *****      DECISION TREE SCORING CODE      *****;
34 *****;
35

```

Scroll through the Path Publish Score Code window to see the code. The code is for the Ensemble model because this is the model selected by the Model Comparison node. If you wanted to get the score code for one of the other models developed, you should connect the Score node to the appropriate modeling node.

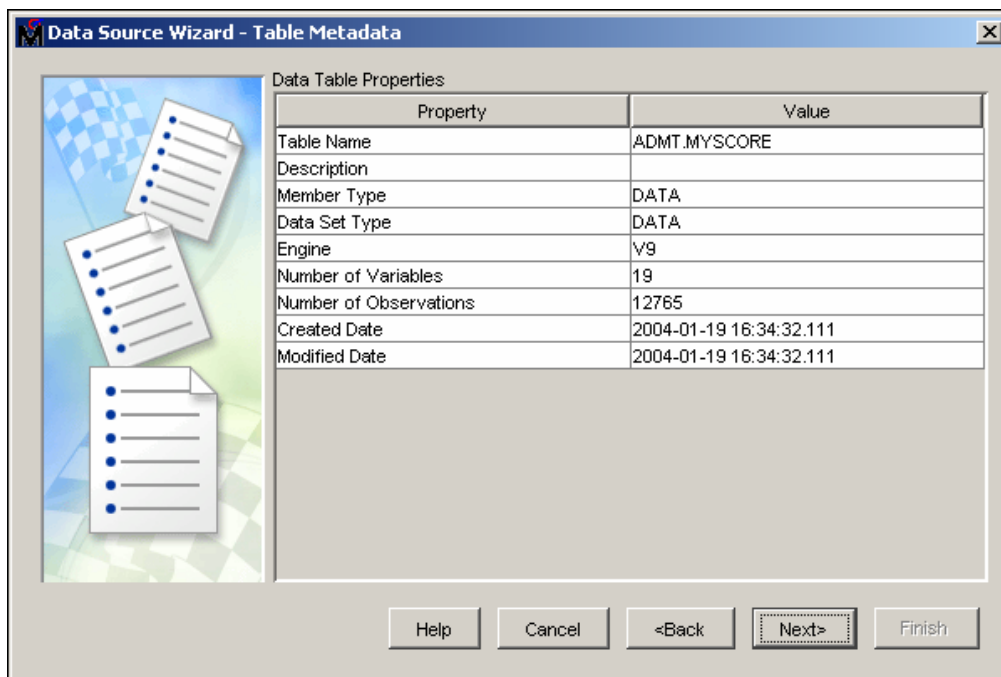
You can score a new data set either within SAS Enterprise Miner or outside of SAS Enterprise Miner. In order to use the code outside of SAS Enterprise Miner, you need to save the code by selecting **File** ⇒ **Save As...**. This will save the code as a \*.sas file for use with Base SAS software.

You can also view and save the C code and the Java code for use outside of SAS.

## Scoring within SAS Enterprise Miner

The saved scoring code can be used in Base SAS to score a data set, but you can also score a new data set within a SAS Enterprise Miner diagram. First, you must create a new data source for the data to be scored. The data to be scored is in the SAS data set **ADMT.MYSCORE**. This data set contains the same columns as the **MYRAW** data set except it does not have the target variable information.

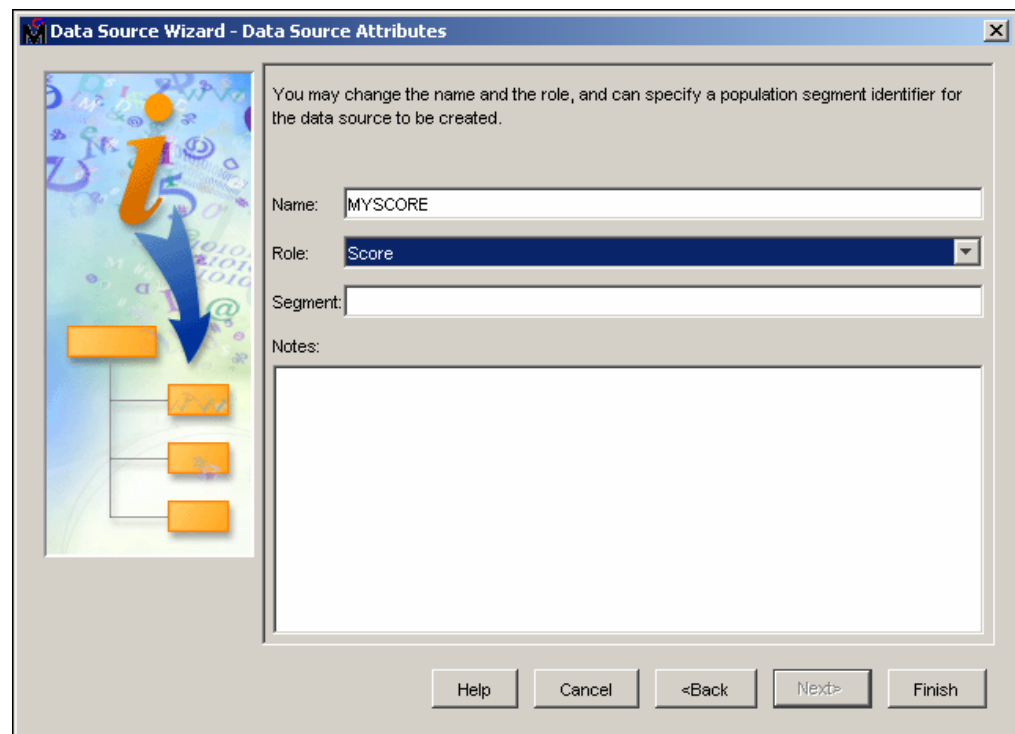
1. Right-click on **Data Sources** in the Project Panel and select **Create Data Source**.
2. In the Data Source Wizard – Metadata Source window, be sure **SAS Table** is selected as the source and select **Next>**.
3. To choose the desired data table select **Browse...**.
4. Double-click on the **ADMT** library to see the data tables in the library.
5. Select the **MYSCORE** data set, and then select **OK**.
6. Select **Next>**.



Observe that this data table has almost 13,000 observations (rows) to be scored.

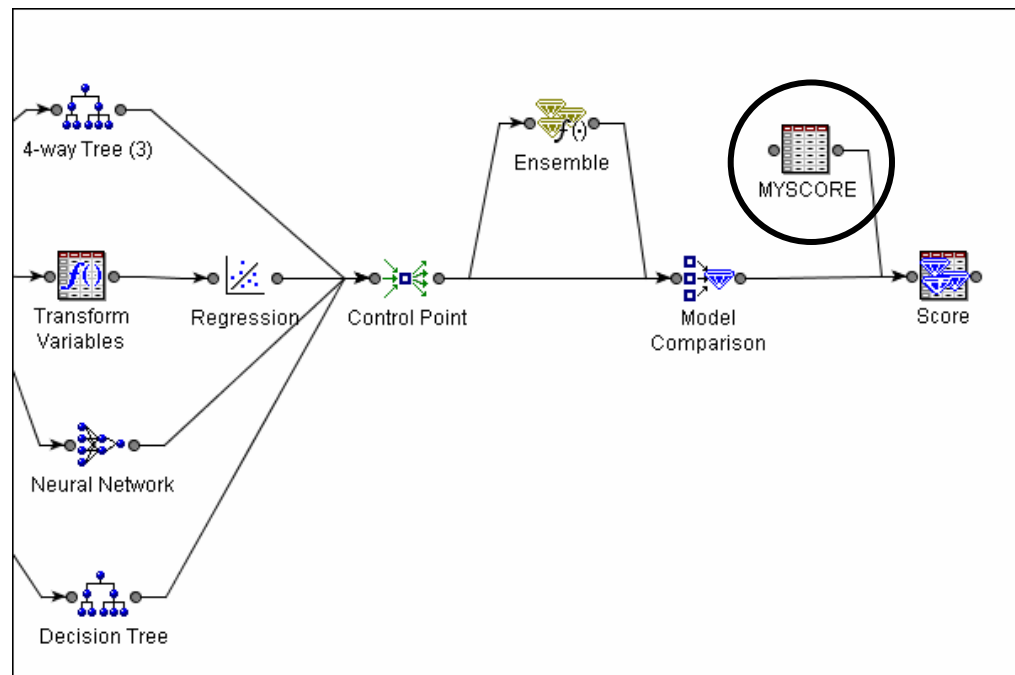
7. After examining the data table properties select **Next>**.
8. There is no need to modify any of the variables in the data set to be scored because the role and level of each variable is built into the scoring code. Select **Next>** to view the variables in the data table.
9. After looking at the variables in the data set, select **Next>**.

10. Change the role of the data set to **Score**.



11. Select **Finish**.

12. Add the **MYSCORE** data source node to the diagram and connect it to the Score node as shown below.



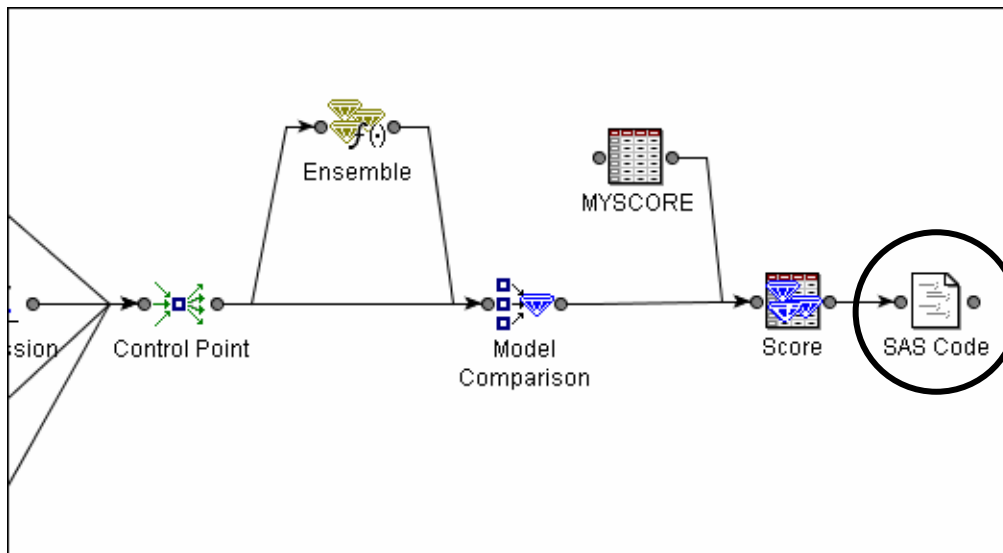
13. Run the diagram from the Score node and view the results when the run is completed.
14. Scroll down in the Output window to examine the distribution of the posterior probabilities of responding to the mailing, **P\_TARGETB1**.


Variable=P_TargetB1				
Statistics	Label	TRAIN	VALIDATE	SCORE
MEAN	Mean	0.05	0.05	0.05
STD	Standard Deviation	0.02	0.02	0.02
N		4881.00	2093.00	12765.00
MIN	Minimum	0.03	0.03	0.02
P25	25th Percentile	0.04	0.04	0.04
MEDIAN	Median	0.05	0.05	0.05
P75	75th Percentile	0.07	0.07	0.07
MAX	Maximum	0.12	0.13	0.13

Information is provided for all three of the data sets.

You can use a SAS Code node to create a report of those most likely to respond to a future mailing. Suppose you make a business decision to mail to 25% of the individuals on the list. You would choose to mail to those with the highest probability of responding. Based on the report shown above, this would be those in the score data set with predicted probabilities of at least 0.07.

15. Add a SAS Code node to the diagram as shown below



16. Select the SAS Code node.
17. Select  in the SAS Code property of the Training section in the Property Panel.

18. Type the following code into the SAS Code window.

```
data mailing;
  set &EM_IMPORT_SCORE;
  if p_targetb1 lt 0.07 then delete;
run;
proc sort data=mailing;
  by idcode;
run;
proc print data=mailing;
  var idcode p_targetb1;
  title 'Mailing List for Potential Donors';
run;
```

The code creates a temporary data set, **mailing**, containing only those observations with predicted probability of responding of at least 0.07. It sorts the resulting data set by **idcode** for ease of identification and prints the report.

19. Run the code and view the results.

Partial results are shown below.

Mailing List for Potential Donors

Obs	IDCode	P_ TargetB1
1	42176	0.07742
2	42177	0.08391
3	42180	0.07921
4	42187	0.07165
5	42188	0.08129
6	42189	0.11809
7	42190	0.07062
8	42202	0.08517
9	42204	0.09467
10	42208	0.08829
11	42214	0.08755
12	42222	0.07707
13	42228	0.08675

## 6.4 Exercises

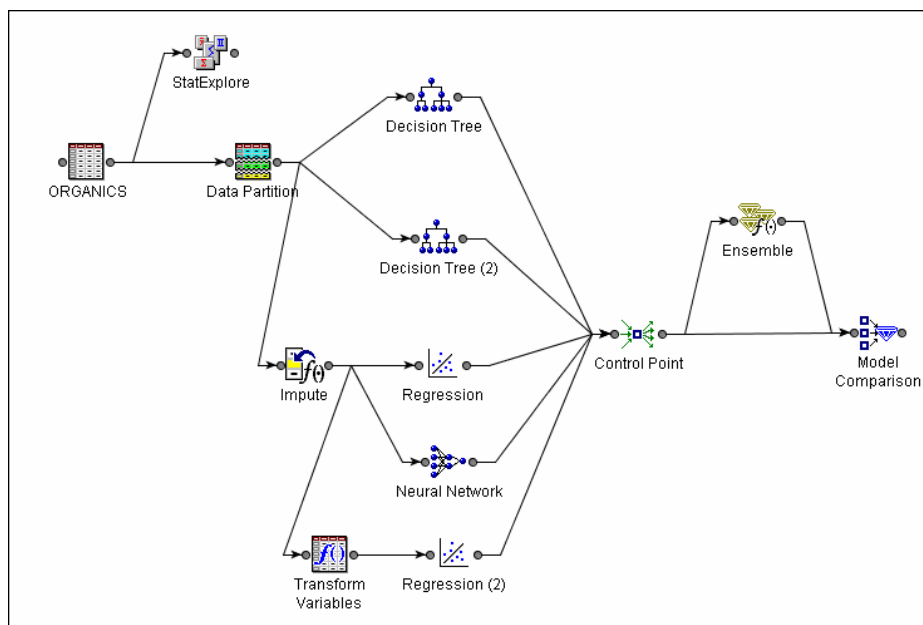
### 1. Generating Ensemble Models

- a. Add an Ensemble node to the Organics diagram in the Exercise project. Connect the Ensemble node to the nodes that generate the two regression models, the two decision tree models, and the neural network model. You might want to run these connections through a Control Point.
- b. Connect the Ensemble node to the Model Comparison node and connect each of the modeling nodes to the Model Comparison node. If you have chosen to use a Control Point, you need only connect the Control Point and the Ensemble node to the Model Comparison node. Rerun the diagram from the Model Comparison node.
- c. Use the Model Comparison node to compare all of the models. Does the Ensemble model appear to be better than the other models?
- d. Under what circumstances might an Ensemble model outperform other types of models?

## 6.5 Solutions to Exercises

### 1. Generating Ensemble Models

- a. Add an Ensemble node to the diagram and connect the other five modeling nodes to the Ensemble node. Rather than creating many crossing connections in the diagram, you may choose to disconnect all of the modeling nodes from the Assessment node and connect them to a Control Point.
- b. If you have chosen to use a Control Point, you need only to connect the Control Point and the Ensemble model to the Assessment node. The diagram using a Control Point is shown below:

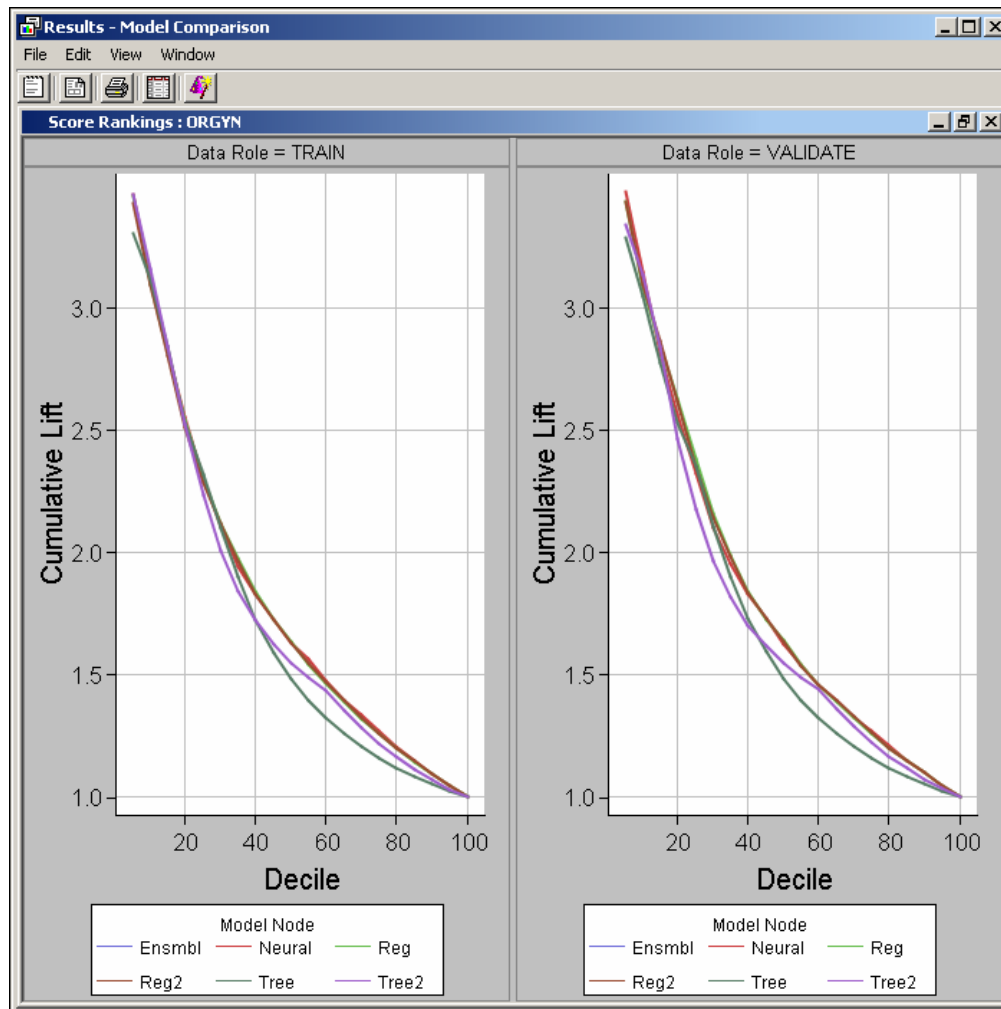


If you did not use a Control Point, all of the modeling nodes are already connected to the Model Comparison node and you need only connect the Ensemble model to the Model Comparison node. To rerun the diagram:

- 1) Right-click on the Model Comparison node and select **Run**.
- 2) View the results when the run completes.

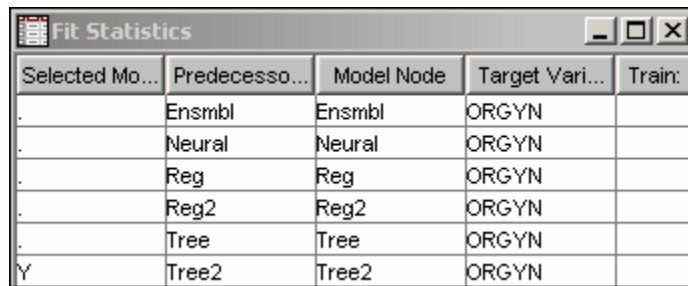


- c. Compare the models.
- 1) Examine the cumulative lift charts in the results.



The Ensemble model does not appear to be better than the other models.

2) Examine the Fit Statistics window in the results.



The image shows a software window titled "Fit Statistics". It contains a table with five columns: "Selected Mo...", "Predecesso...", "Model Node", "Target Vari...", and "Train:". The table lists six rows of model results. The first five rows have a "." in the "Selected Mo..." column, and the last row has a "Y". The "Model Node" column lists "Ensmbl", "Neural", "Reg", "Reg2", "Tree", and "Tree2". The "Target Vari..." column lists "ORGYN" for all rows. The "Train:" column is empty for all rows.

Selected Mo...	Predecesso...	Model Node	Target Vari...	Train:
.	Ensmbl	Ensmbl	ORGYN	
.	Neural	Neural	ORGYN	
.	Reg	Reg	ORGYN	
.	Reg2	Reg2	ORGYN	
.	Tree	Tree	ORGYN	
Y	Tree2	Tree2	ORGYN	

The second decision tree is the model that has been selected.

- d. An Ensemble model can only be better than the individual models if the individual models disagree with one another.

# Chapter 7 Cluster Analysis

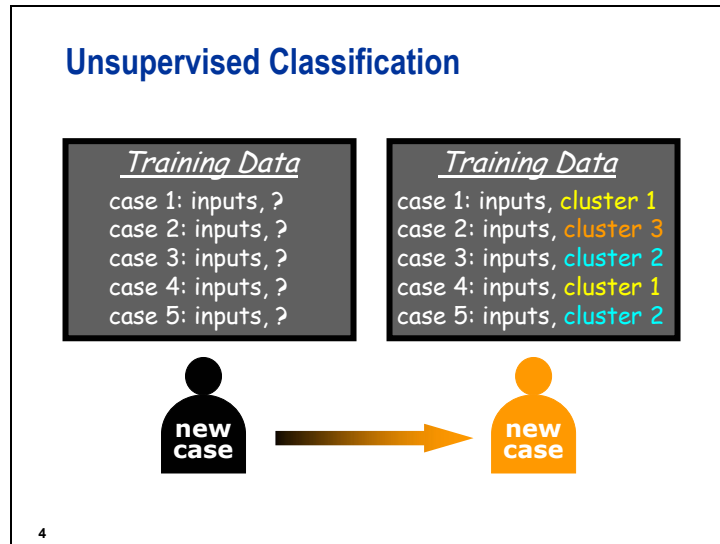
<b>7.1</b>	<b>Using <i>k</i>-Means Cluster Analysis .....</b>	<b>7-3</b>
<b>7.2</b>	<b>Exercises .....</b>	<b>7-20</b>
<b>7.3</b>	<b>Solutions to Exercises .....</b>	<b>7-22</b>



## 7.1 Using $k$ -Means Cluster Analysis

### Objectives

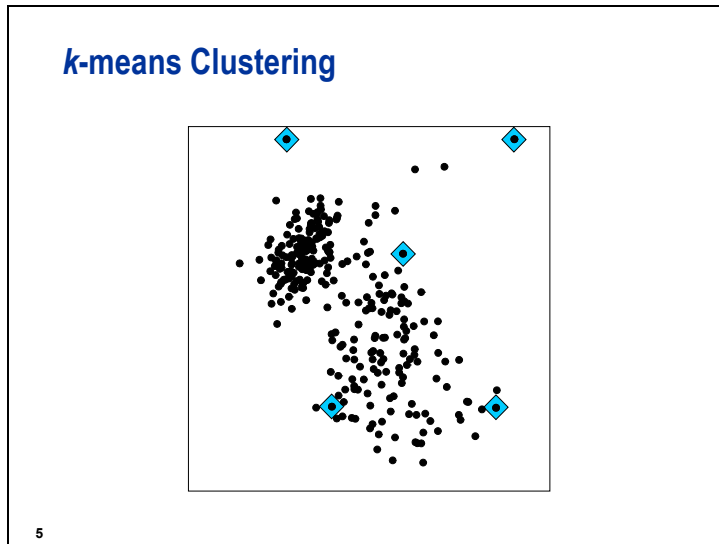
- Discuss the concept of  $k$ -means clustering.
- Define measures of distance in cluster analysis.
- Understand the dangers of forced clustering.
- Generate a cluster analysis and interpret the results.



*Unsupervised classification* (also known as *clustering*) is classification with an unknown target. That is, the class of each case is unknown. Furthermore, the total number of classes is unknown. The aim is to segment the cases into disjoint classes that are homogenous with respect to the inputs.

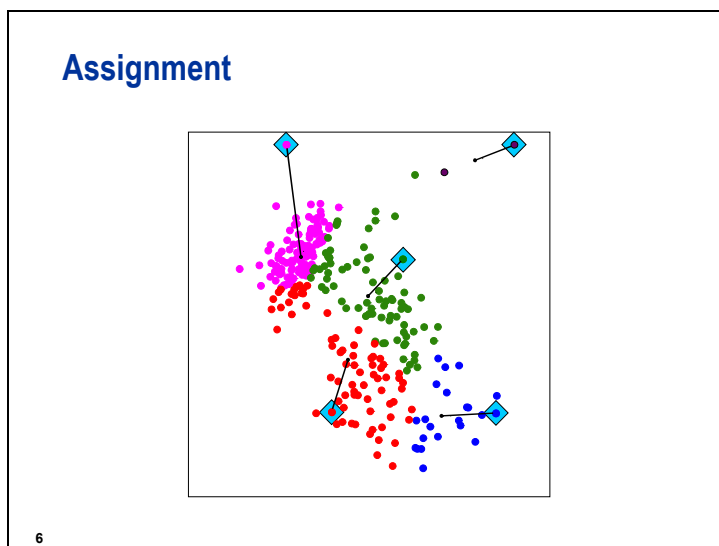
The purpose of clustering is often description. For example, segmenting existing customers into groups and associating a distinct profile with each group could help future marketing strategies. However, there is no guarantee that the resulting clusters will be meaningful or useful.

Unsupervised classification is also useful as a step in a supervised prediction problem. For example, customers could be clustered into homogenous groups based on sales of different items. Then a model could be built to predict the cluster membership based on some more easily obtained input variables.

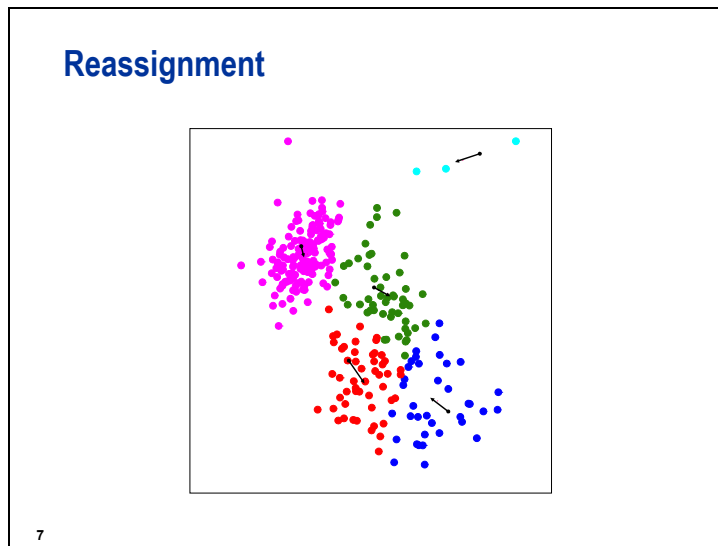


*k*-means clustering is a computationally efficient unsupervised classification method. The first two steps of the method are as follows:

1. Specify the number of clusters (classes)  $k$ . In SAS Enterprise Miner, the number of clusters is determined by the cubic clustering criterion.
2. Choose  $k$  initial cluster seeds.

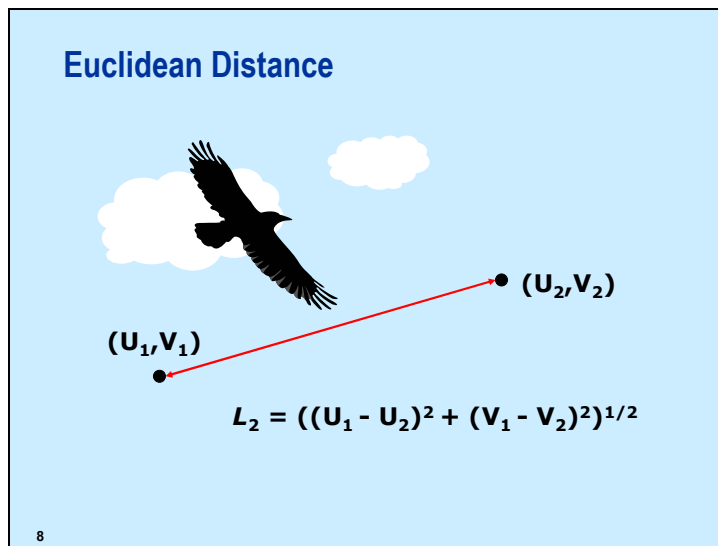


3. Assign cases closest to seed  $i$  as belonging to cluster  $i$ ;  $i = 1, \dots, k$ .
4. Calculate the mean of the cases in each cluster, and move the  $k$  cluster seeds to the mean of their cluster.



5. Reassign cases closest to the new seed  $i$  as belonging to cluster  $i$ .
6. Take the mean of the cases in each cluster as the new cluster seed.

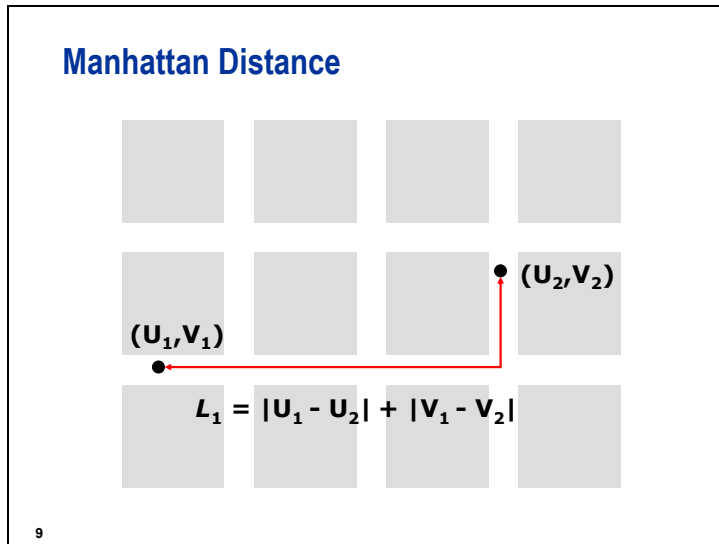
This process can be further iterated, but this is rarely necessary, provided the initial cluster seeds are placed intelligently.



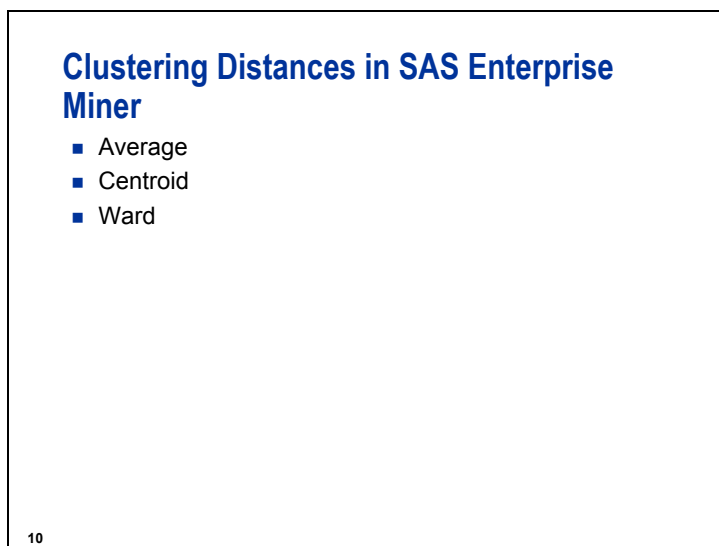
Clustering methods depend on a measure of distance or similarity between points. Different distance metrics used in  $k$ -means clustering can give different types of clusters.

The most widely used metric is Euclidean distance ( $L_2$  norm). The Euclidean distance between two points is the length of the straight line that joins them. Clusters formed using Euclidean distance tend to be spherical in nature.



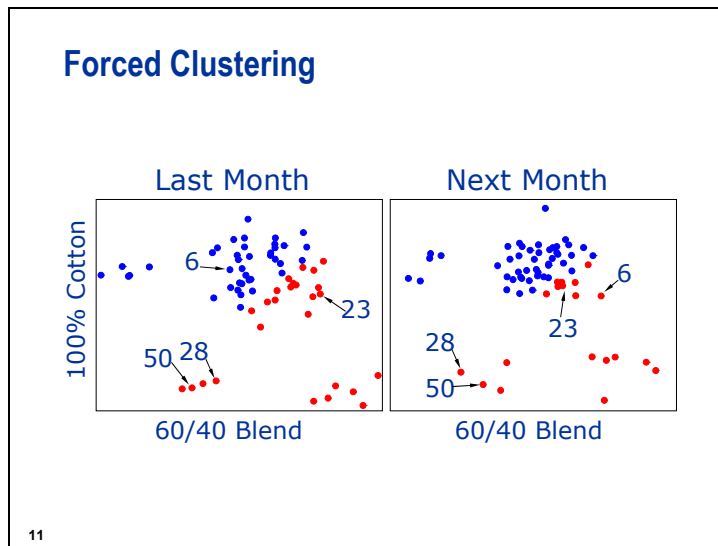


The Manhattan distance ( $L_1$  norm) between two points is the length of the shortest axis-parallel connection between them. The formation of clusters using the Manhattan metric is relatively insensitive to outlying data points. Clusters formed using Manhattan distances tend to be more cubical in shape.



SAS Enterprise Miner has three methods for calculating cluster distances:

- |          |   |
|----------|---|
| Average  | the distance between two clusters is the average distance between pairs of observations, one in each cluster.           |
| Centroid | the distance between two clusters is the Euclidean distance between their centroids or means.                           |
| Ward     | the distance between two clusters is the ANOVA sum of squares between the two clusters added up over all the variables. |



One pitfall of problem formulation is failing to appreciate the limitations of the analytical methods.

Consider a retail allocation-planning problem. The objective is to find two roughly equal-sized clusters of stores that are similar with respect to the sales of two shirt styles. The results will be used to allocate styles to the stores. The hope is that this will be an improvement over the current system of treating every store the same.

In reality, there were four well-separated clusters of stores: one large cluster and three small ones. Cluster analysis forced the stores into two roughly equal-sized groups (red and blue). The clusters are artificial. For example, based on the analysis of last month's data, stores 6 and 23 received different allocations. However, their apparent difference in sales was just random variation within their cluster. In contrast, assignment of the stores to the same true clusters would have produced stable results.

Failure of the cluster analysis to give useful results is not the fault of the analytical method. The problem was not formulated realistically. It would have been better to first attempt to discover what natural clusters exist and then determine whether they are practically useful.

### The Scenario

- The goal is to segment potential customers based on geographic and demographic attributes.
- Known attributes include such things as age, income, marital status, gender, and home ownership.

12

A catalog company periodically purchases lists of prospects from outside sources. They want to design a test mailing to evaluate the potential response rates for several different products. Based on their experience, they know that customer preference for their products depends on geographic and demographic factors. Consequently, they want to segment the prospects into groups that are similar to each other with respect to these attributes.

After the prospects have been segmented, a random sample of prospects within each segment will be mailed one of several offers. The results of the test campaign will allow the analyst to evaluate the potential profit of prospects from the list source overall as well as for specific segments.

The data that was obtained from the vendor is in the table below. The prospects' names and mailing addresses (not shown) were also provided.

Name	Model Role	Measurement Level	Description
AGE	Input	Interval	Age in years
INCOME	Input	Interval	Annual income in thousands
MARRIED	Input	Binary	1=married, 0=not married
GENDER	Input	Binary	F=female, M=male
OWNHOME	Input	Binary	1=homeowner, 0=not a homeowner
LOCATION	Rejected	Nominal	Location of residence (A through H)
CLIMATE	Input	Nominal	Climate code for residence (10, 20, and 30)
FICO	Input	Interval	Credit score
ID	ID	Nominal	Unique customer identification number

Observe that all variables except **ID** and **LOCATION** should be set to input. No target variables are used in a cluster analysis. If you want to identify groups based on a target variable, consider a predictive modeling technique and specify a categorical target. This type of modeling is often referred to as supervised classification because it attempts to predict group or class membership for a specific categorical response variable. Clustering, on the other hand, is referred to as unsupervised classification because it identifies groups or classes within the data based on all the input variables.

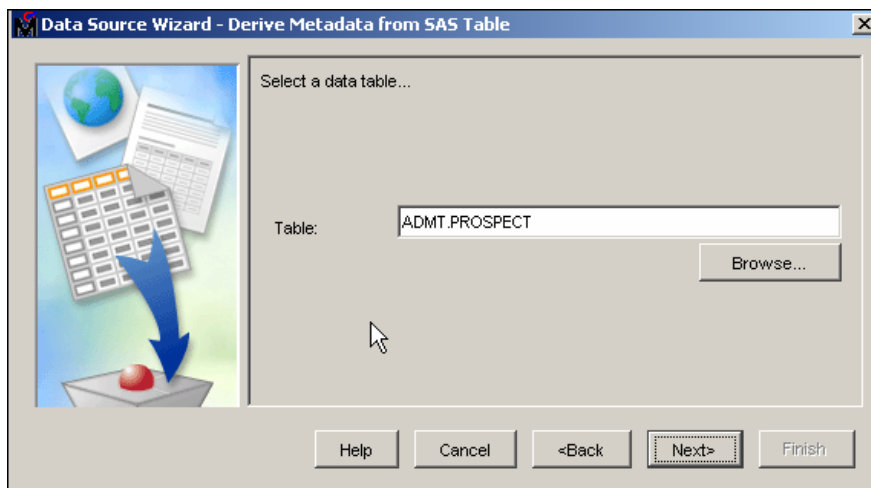


## Cluster Analysis

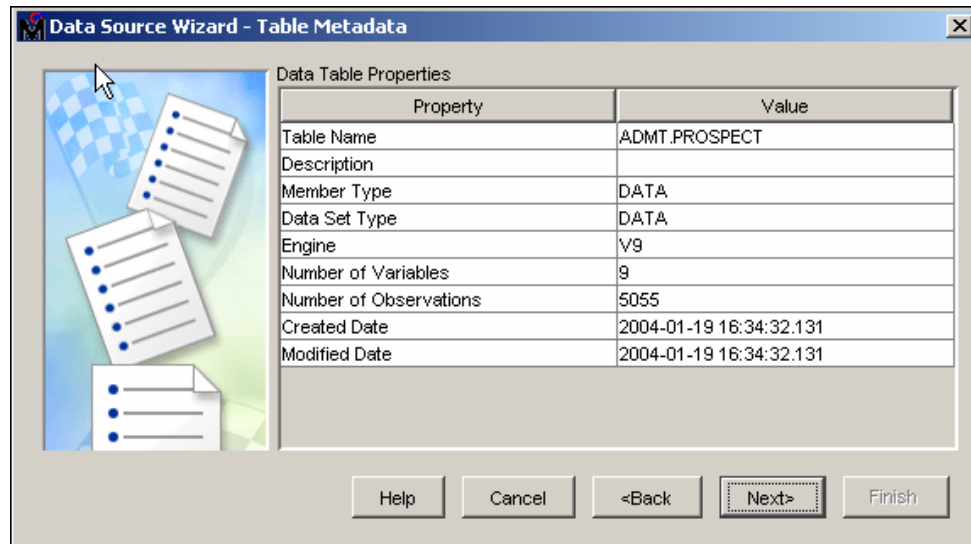
Begin the cluster analysis by creating a new diagram and data source. Open a new diagram and title it **Cluster Analysis**.

### Defining the New Data Source

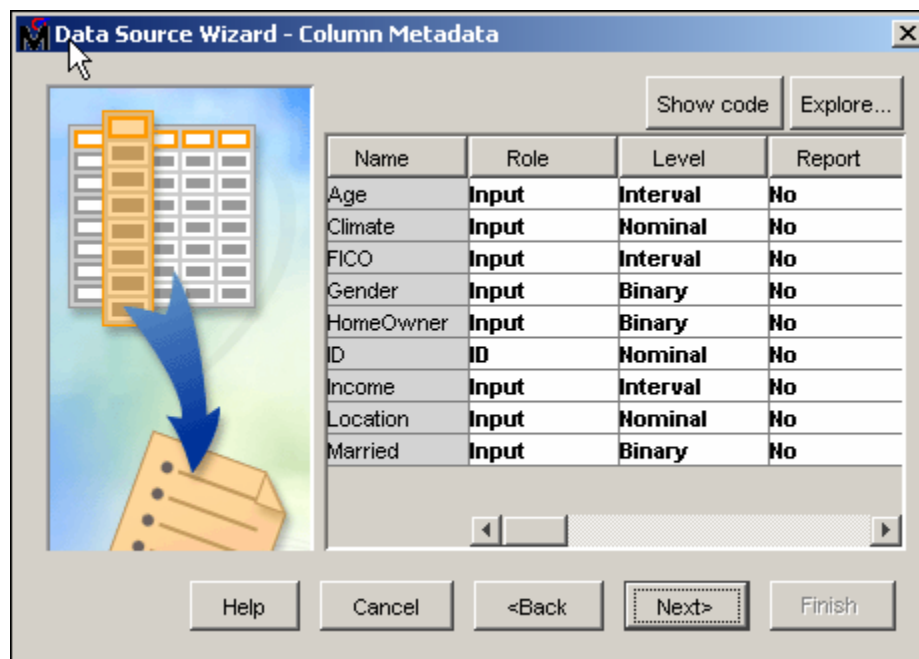
1. Select **File** ⇒ **New** ⇒ **Data Source...**. The new data source will be the SAS data set **PROSPECT** in the ADMT library.
2. In the Data Source Wizard – Metadata Source window, be sure **SAS Table** is selected as the source and select **Next>**.
3. To choose the desired data table, select **Browse...**.
4. Double-click on the **ADMT** library to see the data tables in the library.
5. Select the **PROSPECT** data set, and then select **OK**.



6. Select **Next>**.



7. Examine the data table properties, and then select **Next>**.
8. Select **Advanced** to use the Advanced advisor, and then select **Next>**.
9. Examine the column metadata.

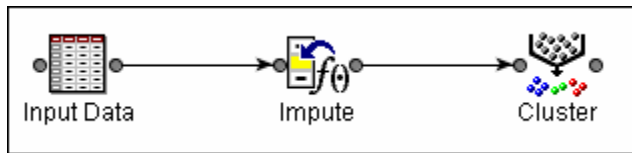


Because the **CLIMATE** variable is a grouping of the **LOCATION** variable, it is redundant to use both. **CLIMATE** was chosen because it had fewer levels (three versus eight) and business knowledge suggested that these three levels were sufficient.


10. Set the model role of **LOCATION** to rejected.
11. Select **Next>**, and then select **Finish**. Notice that the **PROSPECT** data table has been added as a data source for this project.

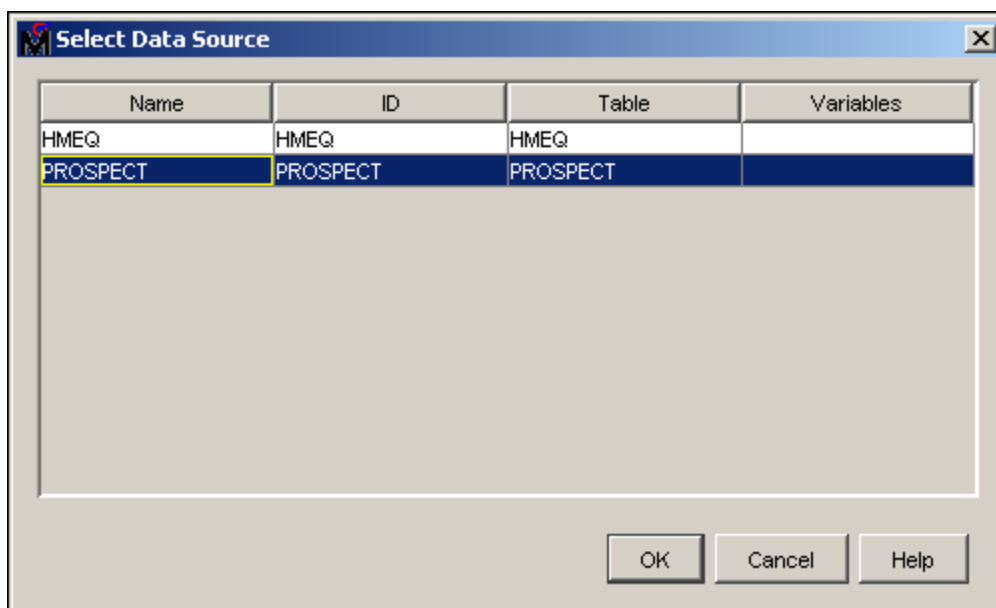
## Building the Initial Flow

1. Assemble the following diagram and connect the nodes as shown.

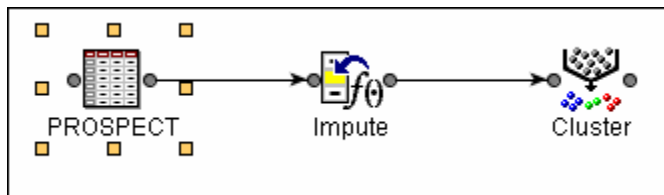


The Impute node is not critical in cluster analysis. If missing value imputation is not done, clustering is based on the non-missing inputs.

2. Select the **Input Data** node in the diagram.
3. In the Property Panel, select  in the Data Source row.



4. In the Select Data Source window, select the **PROSPECT** data table as the data source. Then select **OK**.



## Setting Up the Impute Node

1. Select the **Impute** node in the diagram.
2. In the Property Panel, change the Default Input Method for both Interval and Class variables to **Tree**.

Property	Value
Node ID	Impt
Imported Data	...
Variables	...
Class Variables	
Default Input Method	Tree
Default Target Method	None
Interval Variables	
Default Input Method	Tree
Default Target Method	None

### Setting Up the Cluster Node

1. Select the **Cluster** node.
2. To view additional Cluster node options, select **View** ⇒ **Property Sheet** ⇒ **Advanced**.

*k*-means clustering is very sensitive to the scale of measurement of different inputs. Consequently, it is advisable to use one of the standardization options if the data has not been standardized previously in the flow.

3. Change the Internal Standardization option to **Standardization**.

Property	Value
Node ID	Clus
Imported Data	...
Variables	...
Cluster Variable Role	Segment
Internal Standardization	Standardization



It should be noted that categorical variables tend to dominate cluster analysis because of their pure separation.

4. Observe the other setting options available in the node.

Number of Clusters	
Maximum Number of Clusters	10
Specification Method	Automatic
Selection Criterion	
Clustering Method	Ward
Maximum	50
Minimum	2
CCC Cutoff	3

The default method of determining the number of clusters is automatic. The clustering method property specifies the method used to calculate the clustering distances. In the Ward method the distance is defined as the ANOVA sum of squares between the two clusters added up over all of the variables.

The maximum and minimum values are the maximum and minimum number of clusters that the automatic method will create.



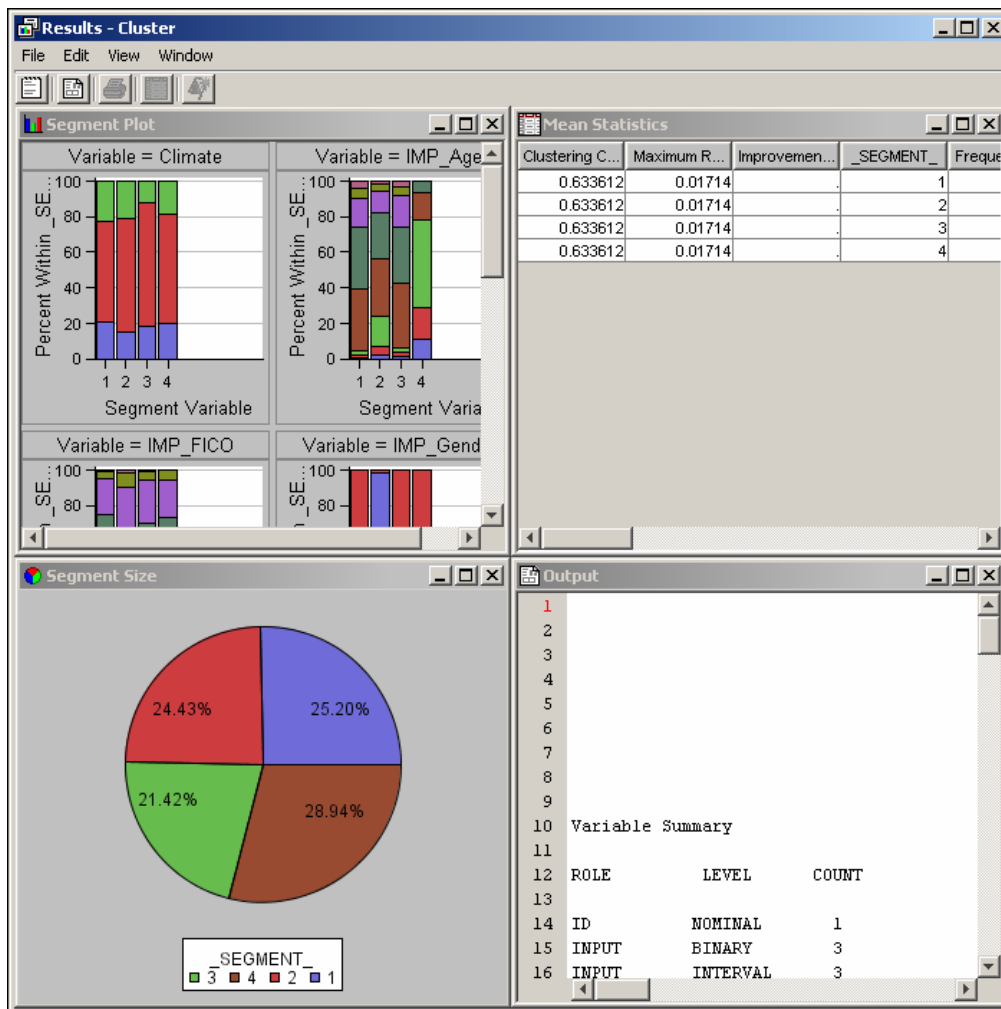
The Cubic Clustering Criterion (CCC) is used to estimate the appropriate number of clusters and the default cutoff value is 3.

You can override these defaults and specify the number of clusters to be created by changing the specification method to user specify. In that case the maximum number of clusters property is used to manually specify the number of clusters to be created.

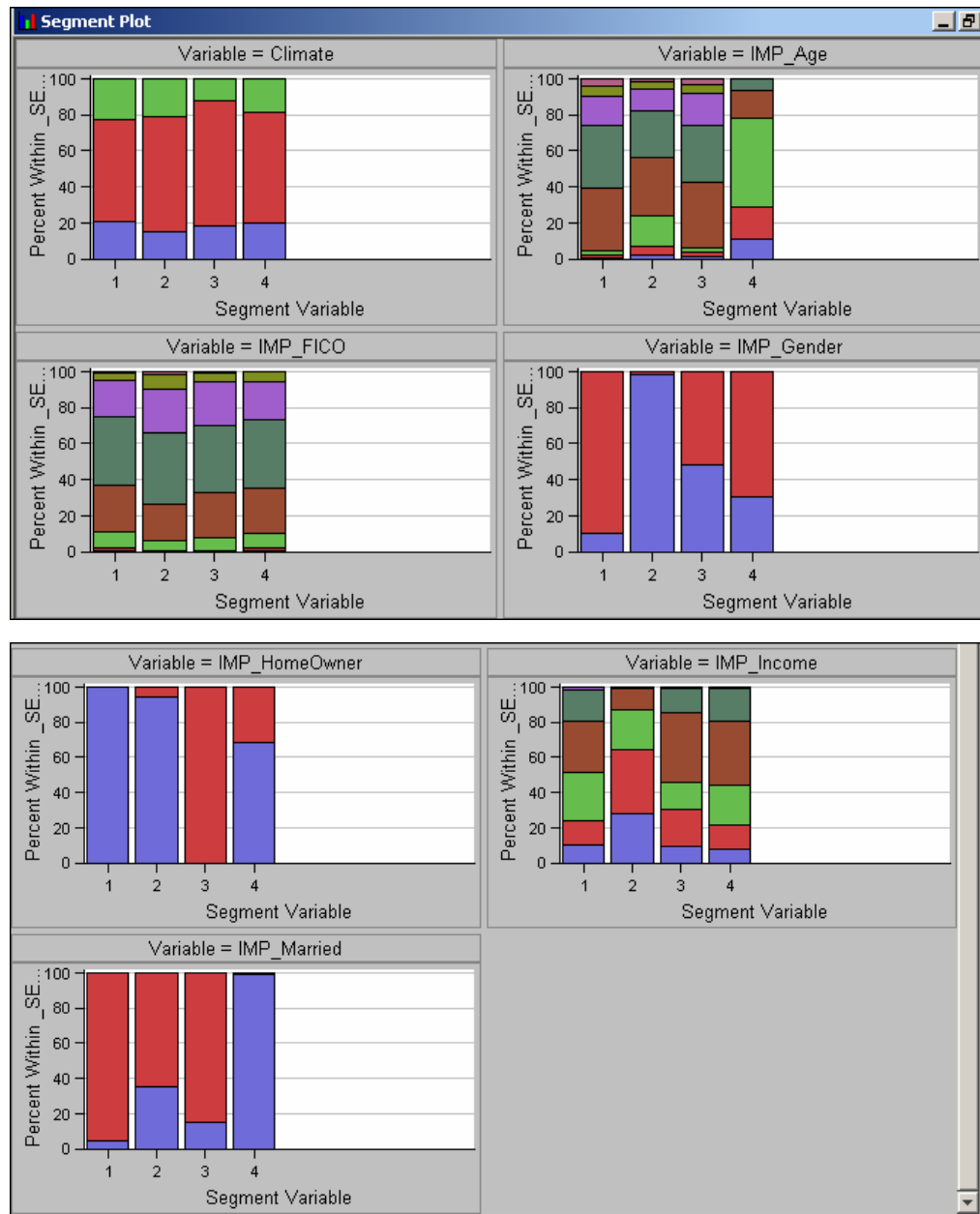
5. Run the diagram from the Cluster node and view the results.

### Exploring the Cluster Node Results

1. Examine the Segment Size pie chart. The observations are divided fairly evenly between the four segments.



2. Maximize the Segment Plot in the Results window to begin examining the differences between the clusters.



When you use your cursor to point at a particular section of a graph, information on that section appears in a pop-up window. Some initial conclusions you might draw from the segment plot are:

- The segments appear to be similar with respect to the variables **CLIMATE** and **FICO**.
- The individuals in segment 3 are all homeowners. There are some homeowners in segment 4, and a few homeowners in segment 2.
- Most of the individuals in segment 1 are married, while most of those in segment 4 are unmarried.


- Younger individuals are in segment 4.
  - Most of the individuals in segment 1 are males while most of those in segment 2 are females.
  - Income appears to be lower in segment 2.
3. Restore the Segment Plot window to its original size and maximize the Mean Statistics window.

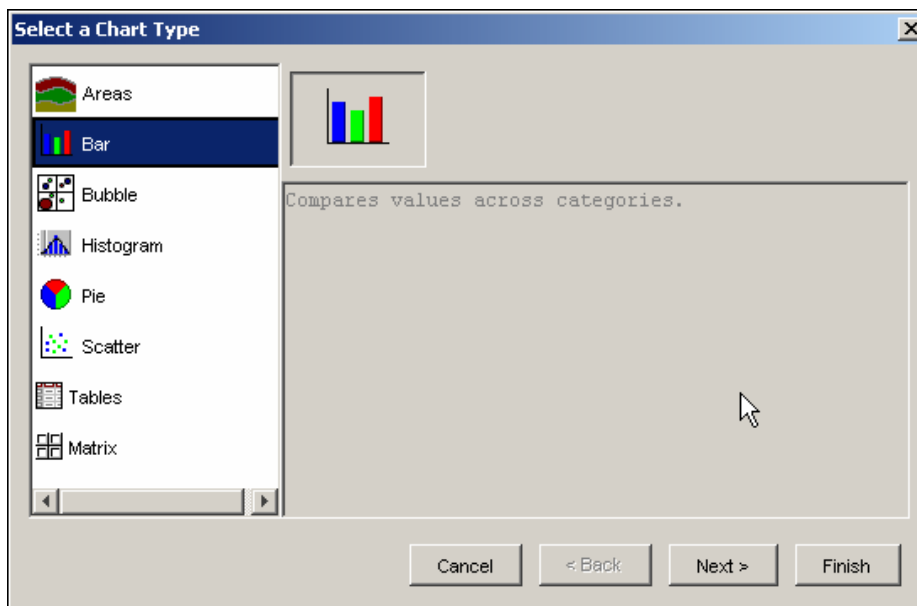
-Mean-...	Maximum Di...	Nearest Clu...	Distance to ...	Imputed Age	Imputed: Cr...	Imputed: Inc...	Climate=10	Climate=20	Clim...
0.594937	5.262233	2	2.134875	49.22141	690.6637	50.84306	0.209576	0.562009	0
0.626185	4.204139	1	2.134875	45.53855	699.2272	36.47568	0.145749	0.644534	0
0.644301	4.468338	1	2.300241	48.81717	694.423	50.48753	0.180979	0.693444	0
0.663799	4.935838	2	2.427328	35.48325	692.0206	52.54545	0.20164	0.611757	0

The window gives descriptive statistics and other information about the clusters such as the frequency of the cluster, the nearest cluster, and the average value of the input variables in the cluster.

Scroll to the right to view the statistics for each variable for each cluster. These statistics confirm some of the conclusions drawn based on the graphs. For example, the average age of individuals in cluster 4 is approximately 35.5, while the average ages in clusters 1, 2, and 3 are approximately 49.2, 45.5, and 48.8 respectively.

You can use the Plot feature to graph some of these mean statistics. For example, create a graph to compare the average income in the clusters.

4. With the Mean Statistics window selected, select **View** ⇒ **Graph Wizard**, or select the plot button .



5. Select **Bar** as the Chart Type and then select **Next>**.
6. Select **Response** as the Role for the variable **IMP\_INCOME**.
7. Select **Category** as the Role for the variable **\_SEGMENT\_**.

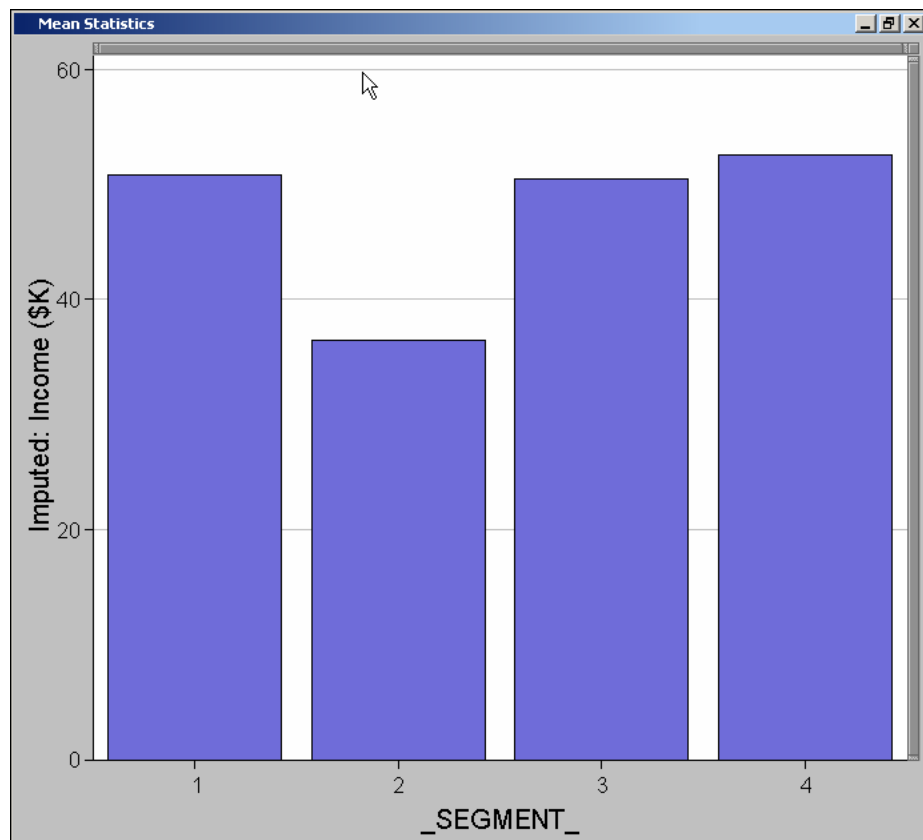
**Select Chart Roles**

Variable	Role	Type	Description	Format
IMP_HOMEOWNER0		Numeric	IMP_HomeOwner=0	
IMP_HOMEOWNER1		Numeric	IMP_HomeOwner=1	
IMP_INCOME	Response	Numeric	Imputed: Income (...)	
IMP_MARRIED0		Numeric	IMP_Married=0	
IMP_MARRIED1		Numeric	IMP_Married=1	
_CRIT_		Numeric	Clustering Criterion	
_FCONV_		Numeric	Improvement in Cl...	
_FREQ_		Numeric	Frequency of Clus...	
_GAP_		Numeric	Distance to Neare...	
_NEAR_		Numeric	Nearest Cluster	
_RADIUS_		Numeric	Maximum Distance...	
_RMSSTD_		Numeric	Root-Mean-Squar...	
_SEGMENT_	Category	Numeric	_SEGMENT_	
XCONV		Numeric	Maximum Relative	

Response statistic: Frequency

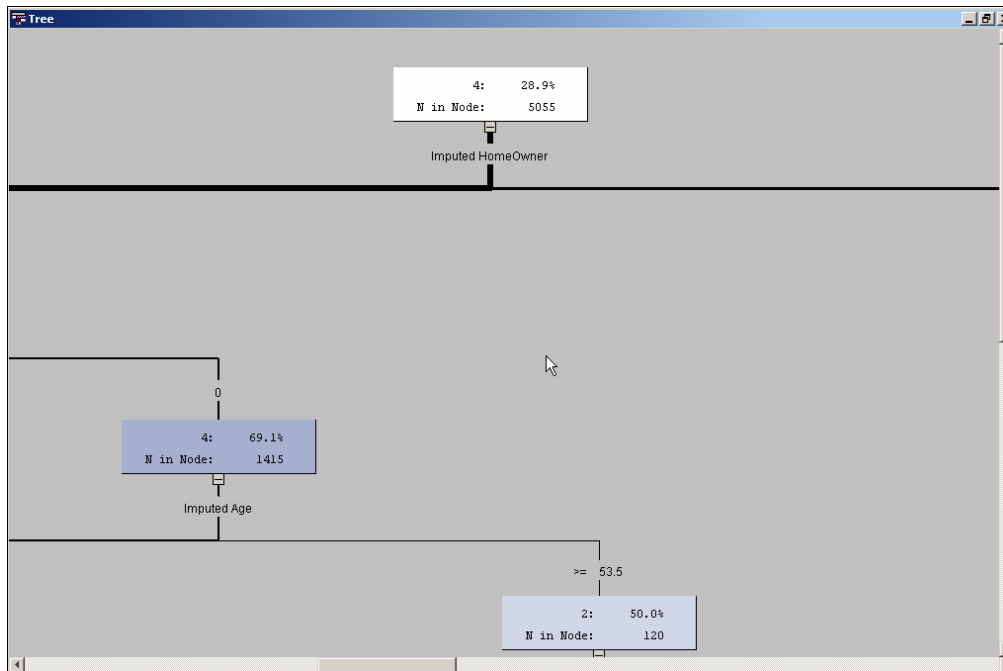
Cancel < Back Next > Finish

8. Select **Finish**.



Another way to examine the clusters is with the cluster profile tree.

9. Select **View** ⇒ **Cluster Profile** ⇒ **Tree**.



You can use the ActiveX feature of the graph to see the statistics for each node and you can control what is displayed with the Edit menu. The tree lists the percentages and numbers of cases assigned to each cluster and the threshold values of each input variable displayed as a hierarchical tree. It enables you to see which input variables are most effective in grouping cases into clusters.

10. Close the Cluster results window when you have finished exploring the results.

In summary, the four clusters can be described as follows:

- Cluster 1      married males
- Cluster 2      lower income females
- Cluster 3      married homeowners
- Cluster 4      younger, unmarried people.

These clusters may or may not be useful for marketing strategies, depending on the line of business and planned campaigns.

## 7.2 Exercises

### 1. Conducting Cluster Analysis

The **DUNGAREE** data set gives the number of pairs of four different types of dungarees sold at stores. Each row represents an individual store. There are five columns in the data set. One column is the store identification number, and the remaining columns contain the number of pairs of each type of jeans sold.

Name	Model Role	Measurement Level	Description
STOREID	ID	Nominal	Identification Number of Store
FASHION	Input	Interval	Number of pairs fashion jeans sold at the store
LEISURE	Input	Interval	Number of pairs of leisure jeans sold at the store
STRETCH	Input	Interval	Number of pairs of stretch jeans sold at the store
ORIGINAL	Input	Interval	Number of pairs of original jeans sold at the store
SALESTOT	Rejected	Interval	Total number of pairs of jeans sold (the sum of FASHION, LEISURE, STRETCH, and ORIGINAL)



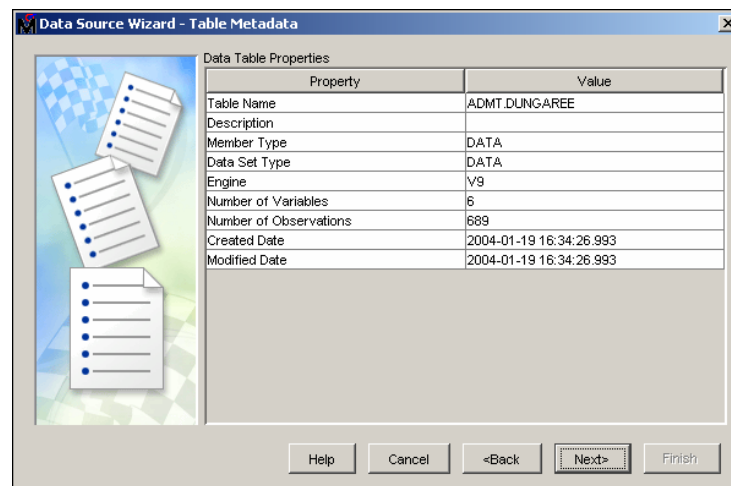
The numbers of pairs of jeans sold are all over a specified time period.

- a. Open a new diagram in your Exercise project. Name the diagram **Jeans**.
- b. Define the data set **DUNGAREE** as a Data Source.
- c. Are the model roles and measurement levels assigned to the variables appropriate? Examine the distribution of the variables. Are there any unusual data values? Are there missing values that should be replaced?
- d. Assign the variable **STOREID** the model role ID and the variable **SALESTOT** the model role rejected. Be sure that the remaining variables have the input model role and interval measurement level. Why should the variable **SALESTOT** be rejected?
- e. Add an Input Data node to the diagram workspace and select the **DUNGAREE** data table as the data source.
- f. Add a Cluster node to the diagram workspace and connect it to the Input Data node.
- g. Select the Cluster node, view the advanced property sheet, and choose the standardization method.
- h. Run the diagram from the Clustering node and examine the results.
- i. After examining the results, summarize the nature of the clusters.

## 7.3 Solutions to Exercises

### 1. Conducting Cluster Analysis

- a. Open a new diagram in the Exercise project.
  - 1) If the Exercise project is open, skip to step 4 below. Otherwise, if the Exercise project is not open, first open the project by selecting **File** ⇒ **Open Project...**.
  - 2) Expand the projects under the appropriate server.
  - 3) Double-click on **Exercise** to open the project.
  - 4) To open a new diagram in the Exercise project, select **File** ⇒ **New** ⇒ **Diagram**.
  - 5) Type in the name of the new diagram, **Jeans**, and select **OK**.
- b. Define the data set **DUNGAREE** as a data source.
  - 1) Select **File** ⇒ **New** ⇒ **Data Source...**.
  - 2) In the Data Source Wizard – Metadata Source window, be sure **SAS Table** is selected as the source and select **Next>**.
  - 3) To choose the desired data table, select **Browse...**.
  - 4) Double-click on the **ADMT** library to see the data tables in the library.
  - 5) Select the **DUNGAREE** data set, and then select **OK**.
  - 6) Select **Next>**.

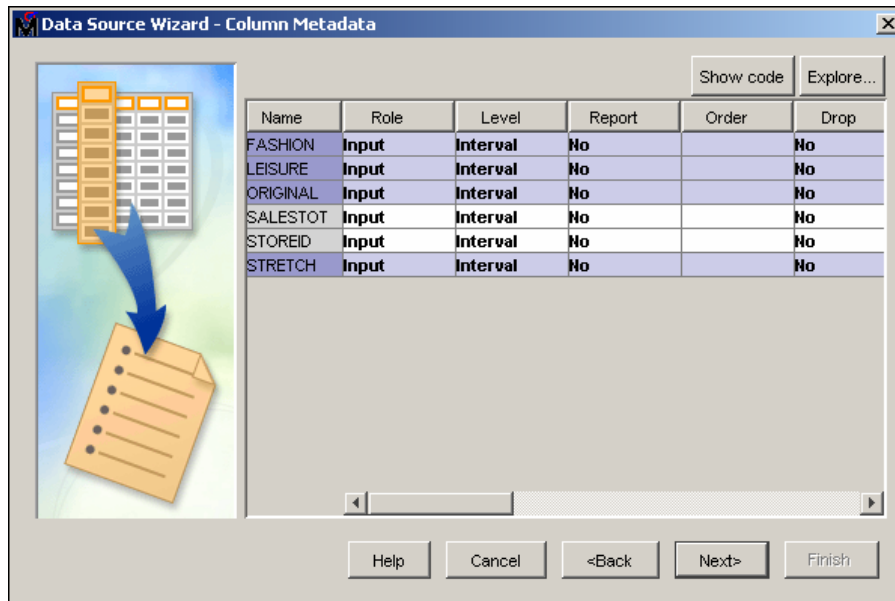


- 7) Select **Next>**.
- 8) Select **Advanced** to use the Advanced advisor, and then select **Next>**.



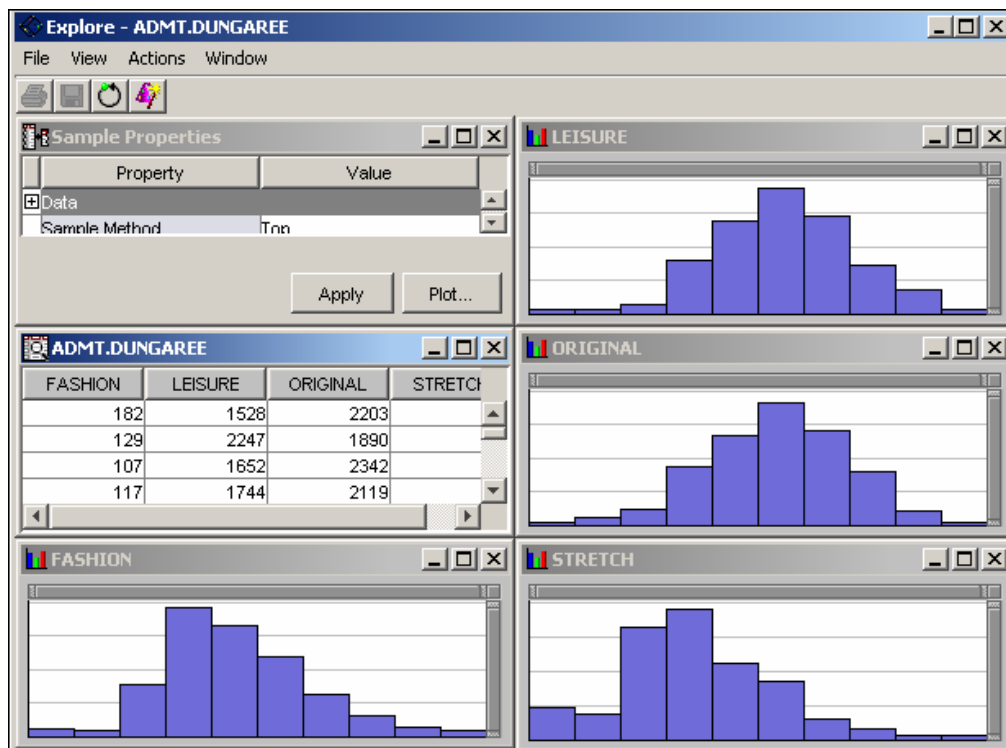
- c. The variable **STOREID** should have the ID model role and the variable **SALESTOT** should have the rejected model role. Examine the distribution of the variables.

1) Control-click to select the variables of interest.



2) Select

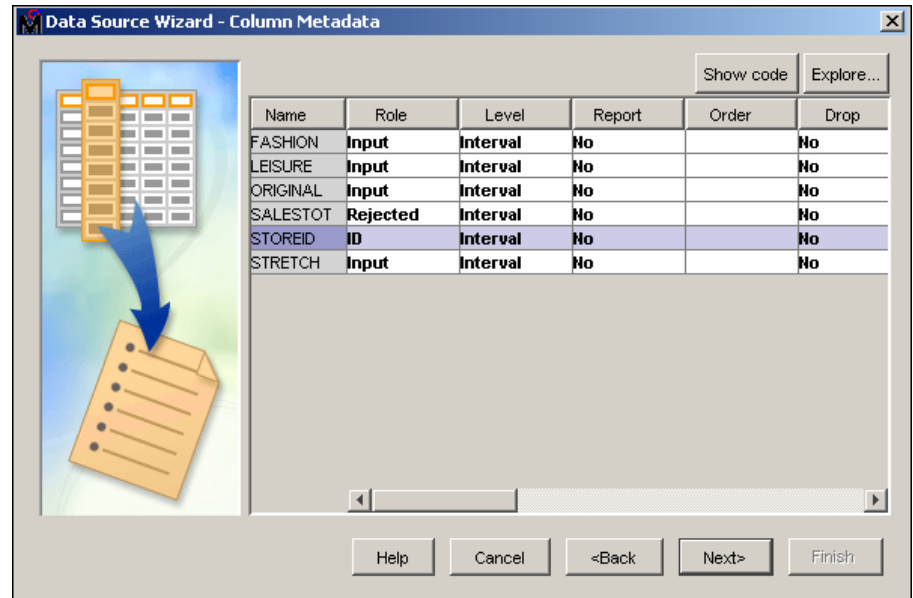
Explore...



There do not appear to be any unusual data values.

d. Assign appropriate model roles to the variables.

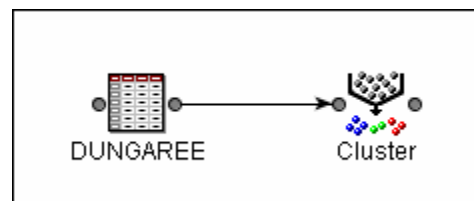
- 1) Click in the Role column of the **STOREID** row and select **ID**.
- 2) Click in the Role column of the **SALESTOT** row and select **Rejected**.



The variable **SALESTOT** should be rejected because it is the sum of the other input variables in the data set. Therefore, it should not be considered as an independent input value.

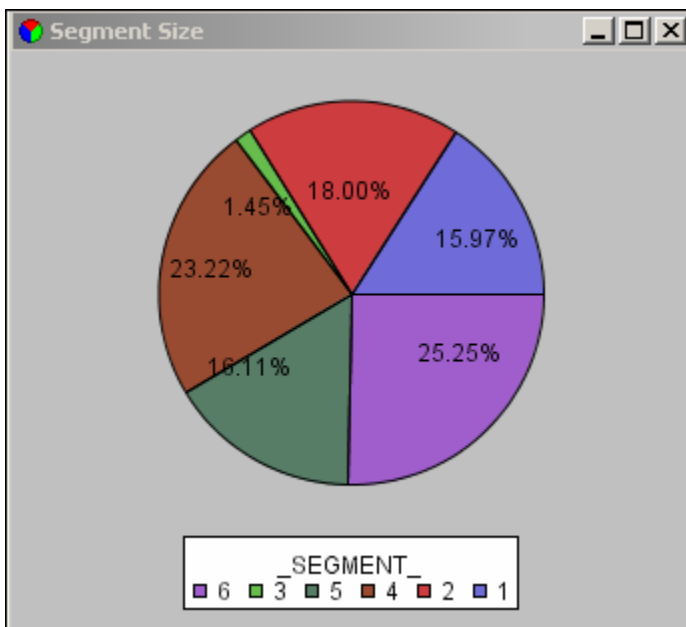
3) Select **Next>** and then **Finish** to complete the data source creation.

- e. To add an Input Data node to the diagram workspace and select the **DUNGAREE** data table as the data source, drag the **DUNGAREE** data source onto the workspace.
- f. Add a Cluster node to the diagram workspace. The workspace should appear as shown.

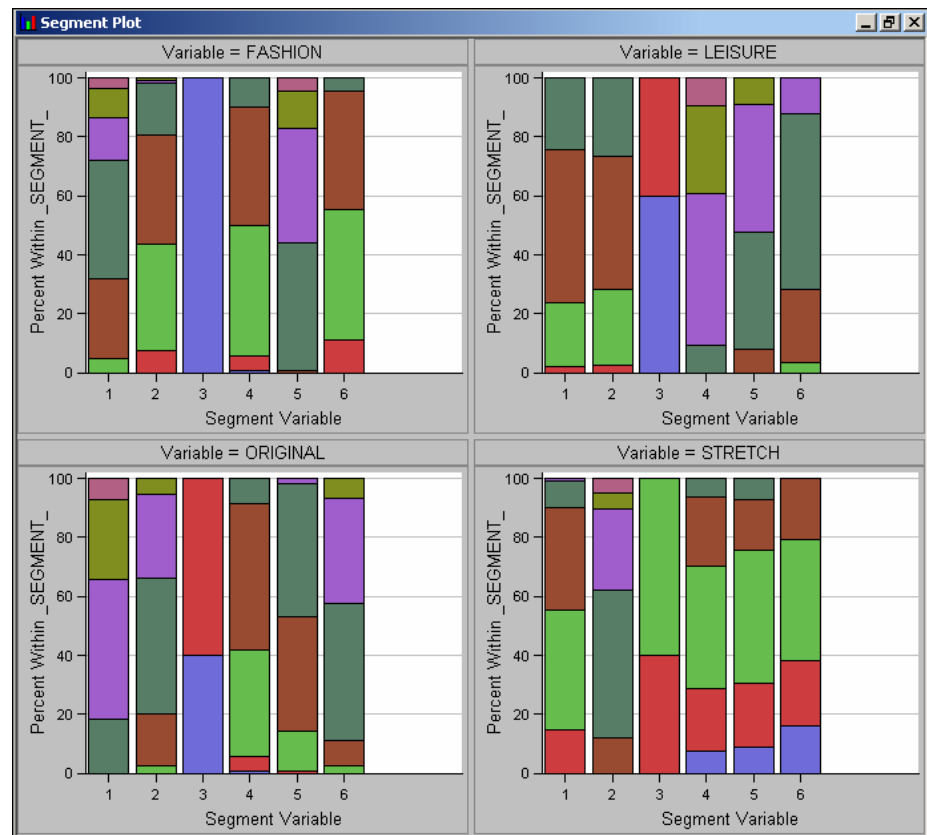


- g. Select the Cluster node, view the advanced property sheet, and choose the standardization method.
- 1) Select the **Cluster** node.
  - 2) Select **View** ⇒ **Property Sheet** ⇒ **Advanced**.
  - 3) In the property sheet, change the Internal Standardization option to **Standardization**.
- h. Run the diagram from the Clustering node and examine the results.
- 1) Right-click on the Cluster node and select **Run**.
  - 2) Select **Yes** when asked if you want to run the path.
  - 3) When the run is completed, select **OK**.
  - 4) To view the results, right-click on the Cluster node and select **Results...**.

_SEGMENT_	Frequency ...
1	110
2	124
3	10
4	160
5	111
6	174



Six clusters were created. Cluster 3 has very few observations.



FASHION	LEISURE	ORIGINAL	STRETCH
117.5455	1646.718	2183.045	454.6545
82.79839	1639.169	1925.202	752.7661
4.2	840	1039.6	315.8
78.11875	2314.669	1561.5	384.4563
133.3784	2080.333	1731.045	365.2432
74.86207	1875.592	1971.172	329.7471

- i. Clusters 1 and 5 appear to be the stores with higher sales of fashion jeans than the other clusters. Cluster 2 has the highest sales of stretch jeans. Cluster 3 appears to be the stores with much lower sales for all of the types of jeans except stretch jeans. Cluster 4 appears to be the stores with high leisure jeans sales. Clusters 5 and 6 do not appear to be remarkable in any category.

# Chapter 8 Association and Sequence Analysis

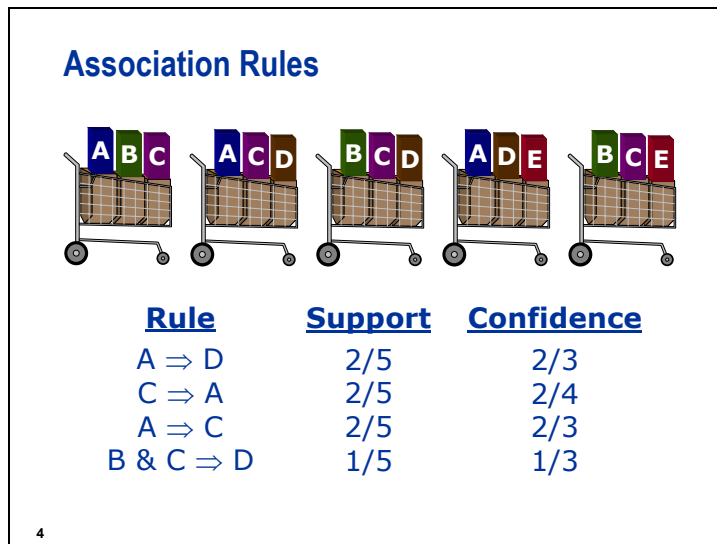
<b>8.1</b>	<b>Introduction to Association Analysis .....</b>	<b>8-3</b>
<b>8.2</b>	<b>Interpretation of Association and Sequence Analysis.....</b>	<b>8-7</b>
<b>8.3</b>	<b>Dissociation Analysis (Self-Study) .....</b>	<b>8-28</b>
<b>8.4</b>	<b>Exercises .....</b>	<b>8-33</b>
<b>8.5</b>	<b>Solutions to Exercises .....</b>	<b>8-35</b>



## 8.1 Introduction to Association Analysis

### Objectives

- Define an association rule.
- Define support, confidence, expected confidence and lift.
- Discuss difficulties in obtaining or acting upon results.



*Association rule discovery* (market-basket analysis, affinity analysis) is a popular data mining method. In the simplest situation, the data consists of two variables: a *transaction* and an *item*.

For each transaction, there is a list of items. Typically, a transaction is a single customer purchase and the items are the things that were bought. An *association rule* is a statement of the form (item set  $A$ )  $\Rightarrow$  (item set  $B$ ).

The aim of the analysis is to determine the strength of all the association rules among a set of items.

The strength of the association is measured by the *support* and *confidence* of the rule. The support for the rule  $A \Rightarrow B$  is the probability that the two item sets occur together. The support of the rule  $A \Rightarrow B$  is estimated by

$$\frac{\text{transactions that contain every item in } A \text{ and } B}{\text{all transactions}}.$$

Notice that support is reflexive. That is, the support of the rule  $A \Rightarrow B$  is the same as the support of the rule  $B \Rightarrow A$ .

The confidence of an association rule  $A \Rightarrow B$  is the conditional probability of a transaction containing item set  $B$  given that it contains item set  $A$ . The confidence is estimated by

$$\frac{\text{transactions that contain every item in } A \text{ and } B}{\text{transactions that contain the items in } A}.$$



**Implication?**

		Checking Account		
		No	Yes	
Saving Account	No	500	3,500	4,000
	Yes	1,000	5,000	6,000
				10,000

Support(SVG  $\Rightarrow$  CK) = 50%  
 Confidence(SVG  $\Rightarrow$  CK) = 83%  
 Expected Confidence(SVG  $\Rightarrow$  CK) = 85%  
 Lift(SVG  $\Rightarrow$  CK) =  $0.83/0.85 < 1$

5

The interpretation of the implication ( $\Rightarrow$ ) in association rules is precarious. High confidence and support does not imply cause and effect. The rule is not necessarily interesting. The two items might not even be correlated. The term confidence is not related to the statistical usage; therefore, there is no repeated sampling interpretation.

Consider the association rule (saving account)  $\Rightarrow$  (checking account). This rule has 50% support (5,000/10,000) and 83% confidence (5,000/6,000). Based on these two measures, this might be considered a strong rule. On the contrary, those **without** a savings account are even more likely to have a checking account (87.5%). Saving and checking are in fact negatively correlated.

If the two accounts were independent, then knowing that one has a saving account does not help in knowing whether one has a checking account. The *expected confidence* if the two accounts were independent is 85% (8,500/10,000). This is higher than the confidence of SVG  $\Rightarrow$  CK.

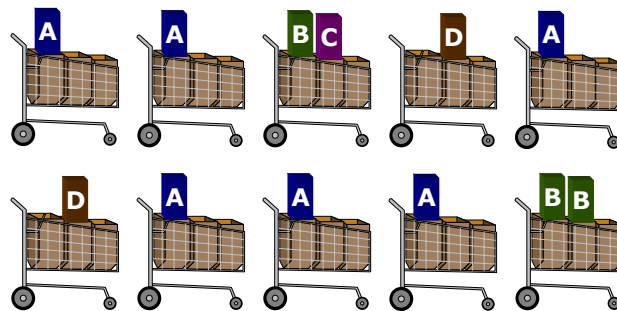
The *lift* of the rule  $A \Rightarrow B$  is the confidence of the rule divided by the expected confidence, assuming the item sets are independent. The lift can be interpreted as a general measure of association between the two item sets. Values greater than 1 indicate positive correlation; values equal to 1 indicate zero correlation; and values less than 1 indicate negative correlation. Notice that lift is reflexive. That is, the lift of the rule  $A \Rightarrow B$  is the same as the lift of the rule  $B \Rightarrow A$ .

**Barbie® ⇒ Candy**

1. Put them closer together in the store.
2. Put them far apart in the store.
3. Package candy bars with the dolls.
4. Package Barbie + candy + poorly selling item.
5. Raise the price on one, lower it on the other.
6. Barbie accessories for proofs of purchase.
7. Do not advertise candy and Barbie together.
8. Offer candies in the shape of a Barbie doll.

6

*Forbes* (Palmeri 1997) reported that a major retailer has determined that customers who buy Barbie dolls have a 60% likelihood of buying one of three types of candy bars. The confidence of the rule  $\text{Barbie} \Rightarrow \text{candy}$  is 60%. The retailer was unsure what to do with this nugget. The online newsletter *Knowledge Discovery Nuggets* invited suggestions (Piatesky-Shapiro 1998).

**Data Capacity**

8

In data mining, the data is not generated to meet the objectives of the analysis. It must be determined whether the data, as it exists, has the capacity to meet the objectives. For example, quantifying affinities among related items would be pointless if very few transactions involved multiple items. Therefore, it is important to do some initial examination of the data before attempting to do association analysis.

## 8.2 Interpretation of Association and Sequence Analysis

### Objectives

- Conduct an association analysis and interpret the results.
- Distinguish between association analysis and sequence analysis.
- Conduct a sequence analysis and interpret the results.

### Scenario – Banking Services

- ATM
- Automobile Loan
- Credit Card
- Certificate of Deposit
- Check/Debit Card
- Checking Account
- Home Equity Line of Credit
- Individual Retirement Account
- Money Market Deposit Account
- Mortgage
- Personal/Consumer Installment Loan
- Savings Account
- Personal Trust Account

11

A bank wants to examine its customer base and understand which of its products individual customers own in combination with one another. It has chosen to conduct a market-basket analysis of a sample of its customer base. The bank has a data set that lists the banking products/services used by 7,991 customers. Thirteen possible products are represented as shown above.

There are three variables in the data set.

Name	Model Role	Measurement Level	Description
ACCOUNT	ID	Nominal	Account Number
SERVICE	Target	Nominal	Type of Service
VISIT	Sequence	Ordinal	Order of Product Purchase



## Association Analysis

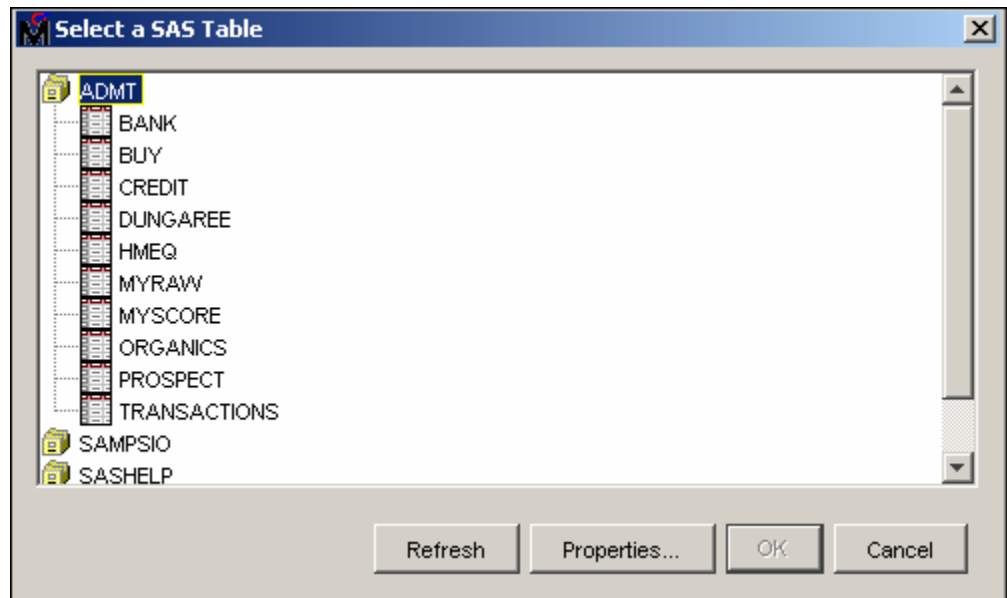
The BANK data set has over 32,000 rows. Each row of the data set represents a customer-service combination. Therefore, a single customer can have multiple rows in the data set, each row representing one of the products he or she owns. The median number of products per customer is three.

The 13 products are represented in the data set using the following abbreviations:

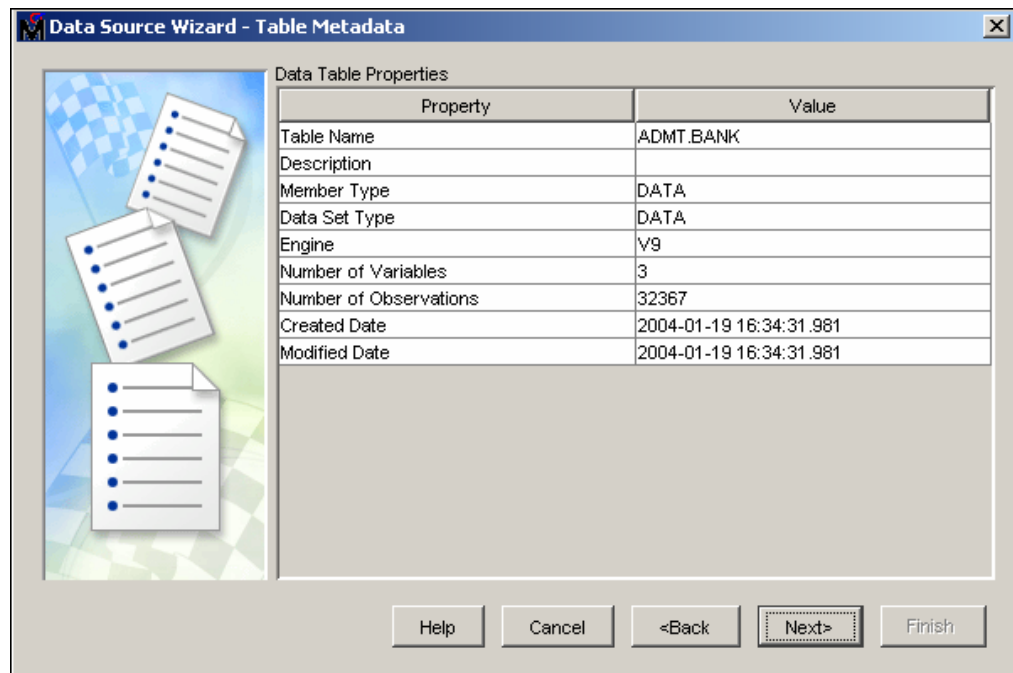
ATM	automated teller machine debit card
AUTO	automobile installment loan
CCRD	credit card
CD	certificate of deposit
CKCRD	check/debit card
CKING	checking account
HMEQLC	home equity line of credit
IRA	individual retirement account
MMDA	money market deposit account
MTG	mortgage
PLOAN	personal/consumer installment loan
SVG	saving account
TRUST	personal trust account

1. Open the project called My Project. To open a new diagram workspace, select **File** ⇒ **New** ⇒ **Diagram...**.
2. Name the diagram **Associations** and select **OK**.
3. To add a new data source to the project, right-click on **Data Sources** in the project tree and select **Create Data Source**.
4. In the Data Source Wizard – Metadata Source window, be sure **SAS Table** is selected as the source and select **Next>**.
5. To choose the desired data table, select **Browse...**.

6. Double-click on the **ADMT** library to see the data tables in the library.

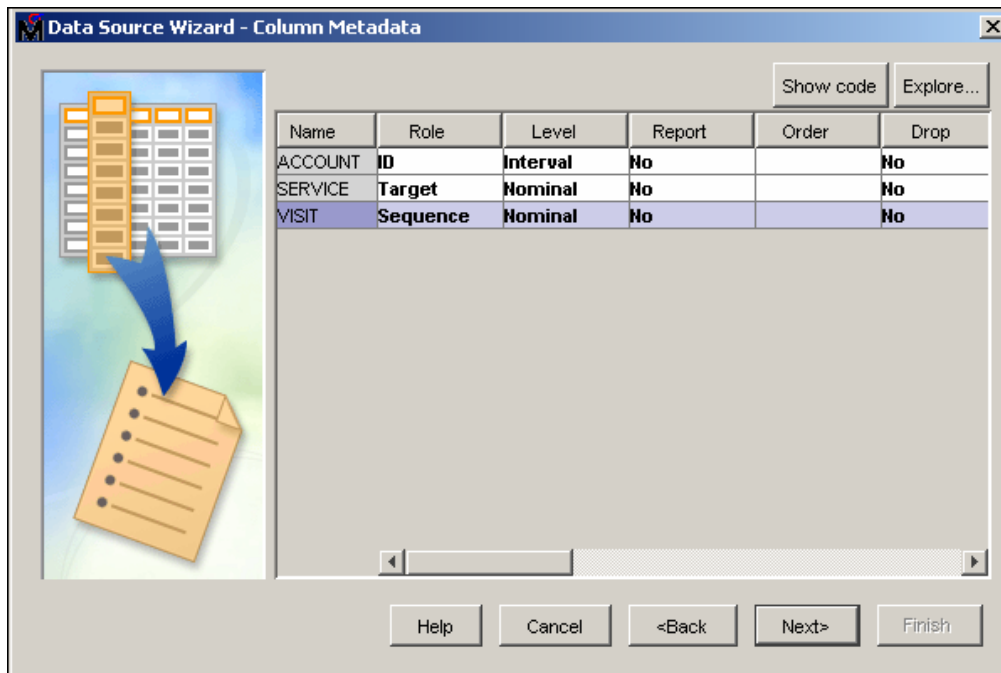


7. Select the **BANK** data set, and then select **OK**.
8. Select **Next>**.

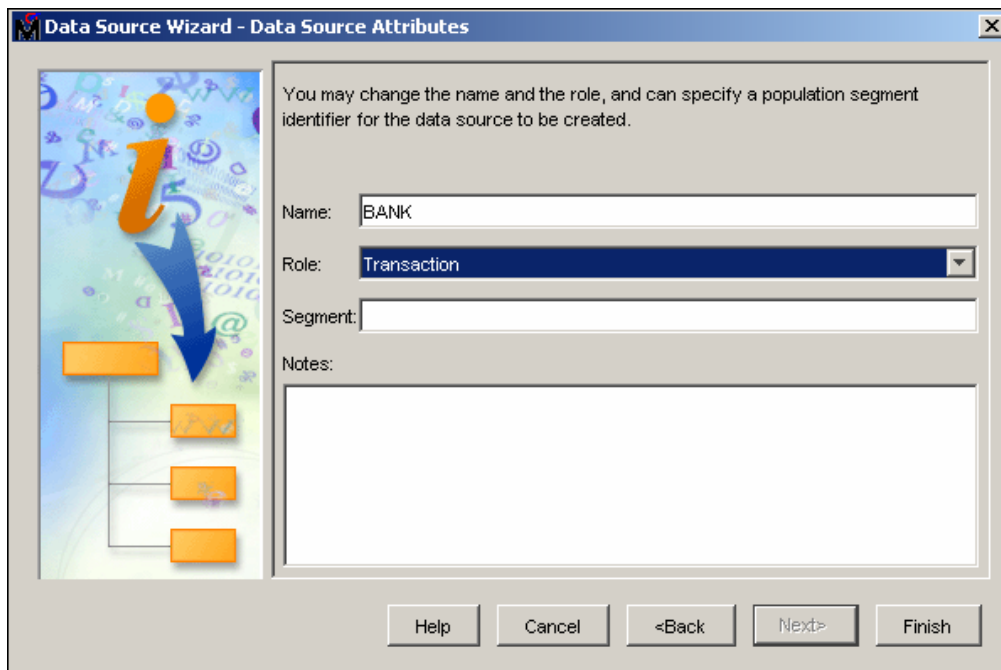


9. Examine the data table properties, then select **Next>**.
10. Select **Advanced** to use the Advanced advisor, and then select **Next>**.

11. Set the role for **ACCOUNT** to **ID**, for **SERVICE** to **Target**, and for **VISIT** to **Sequence**.

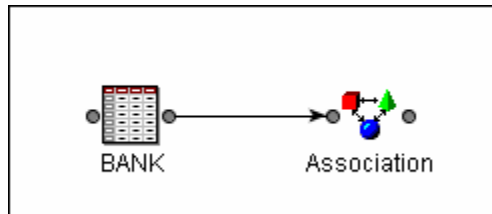


12. Select **Next>**, and then select **Next>** again.
13. Because you are interested in an association analysis, change the Role to **Transaction**.



14. Select **Finish**.


15. Add the node for the **BANK** data set and an Association node to the diagram workspace as shown below.

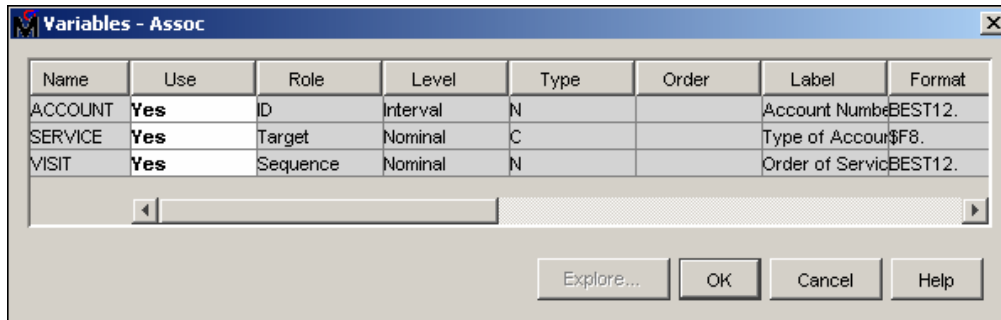



16. Select the Association node and examine the Property Panel.

Property	Value
Node ID	Assoc
Imported Data	...
Variables	...
Association	
Minimum Confidence L	10
Support Type	Percent
Support Count	2
Support Percentage	5.0
Maximum Items	4
Sequence	
Chain Count	3
Consolidate Time	0.0
Maximum Transaction	
Support Type	Percent
Support Count	2
Support Percentage	2.0
Rules	
Number to Keep	200
Sort Criterion	DEFAULT
Number to Transpose	200
Export Rule by ID	No
Status	
Last Error	
Last Status	
Needs Updating	Yes
Needs to Run	Yes
Time of Last Run	
Run Duration	



17. To examine the variables to be used in the analysis, click  in the Variables row.



 Because the data table has a sequence variable with a status of use, the Association node will perform a sequence analysis by default. Sequence analysis is discussed later in this section, so for the moment, perform an association analysis.


18. Change the Use status of the variable **VISIT** to **No**. The default analysis will now be an association analysis.

19. Select **OK**.

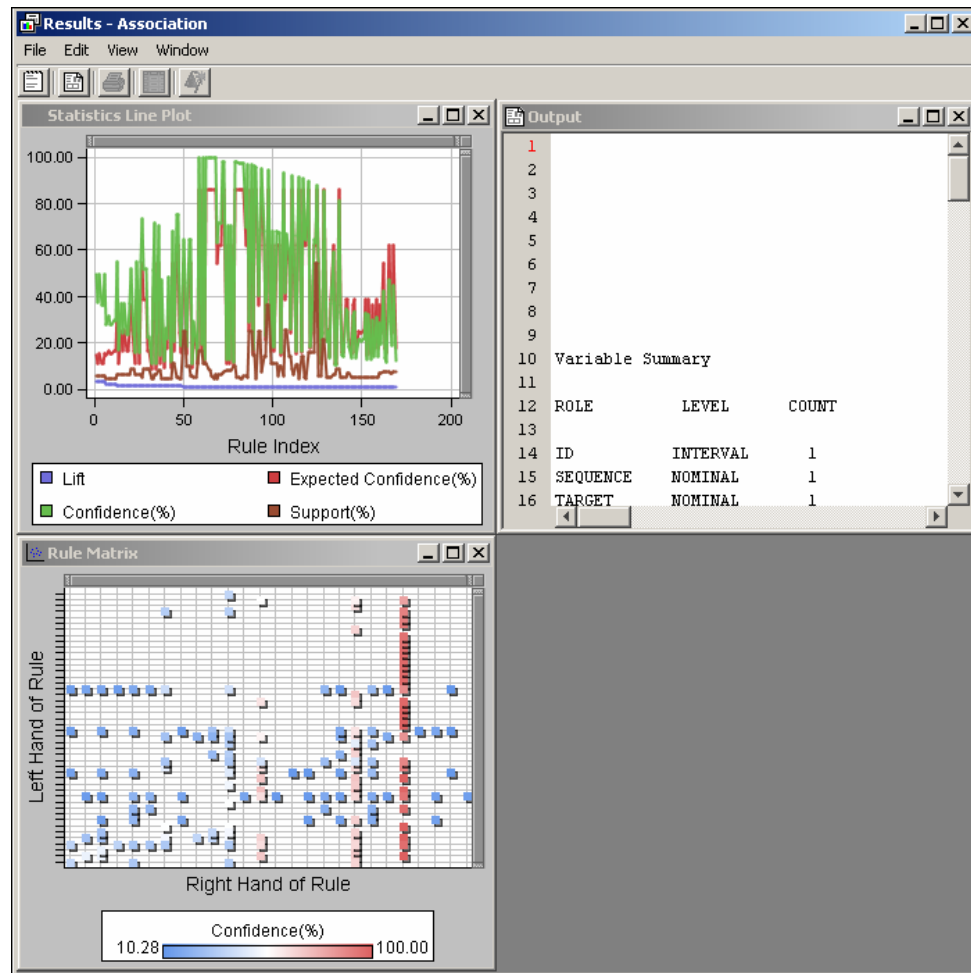
20. The Export Rule by ID property determines if the Rule-by-ID data is exported from the node and if the Rule Description table will be available for display in the Results window. Set Export Rule by ID to **Yes**.

Other options in the Property Panel include

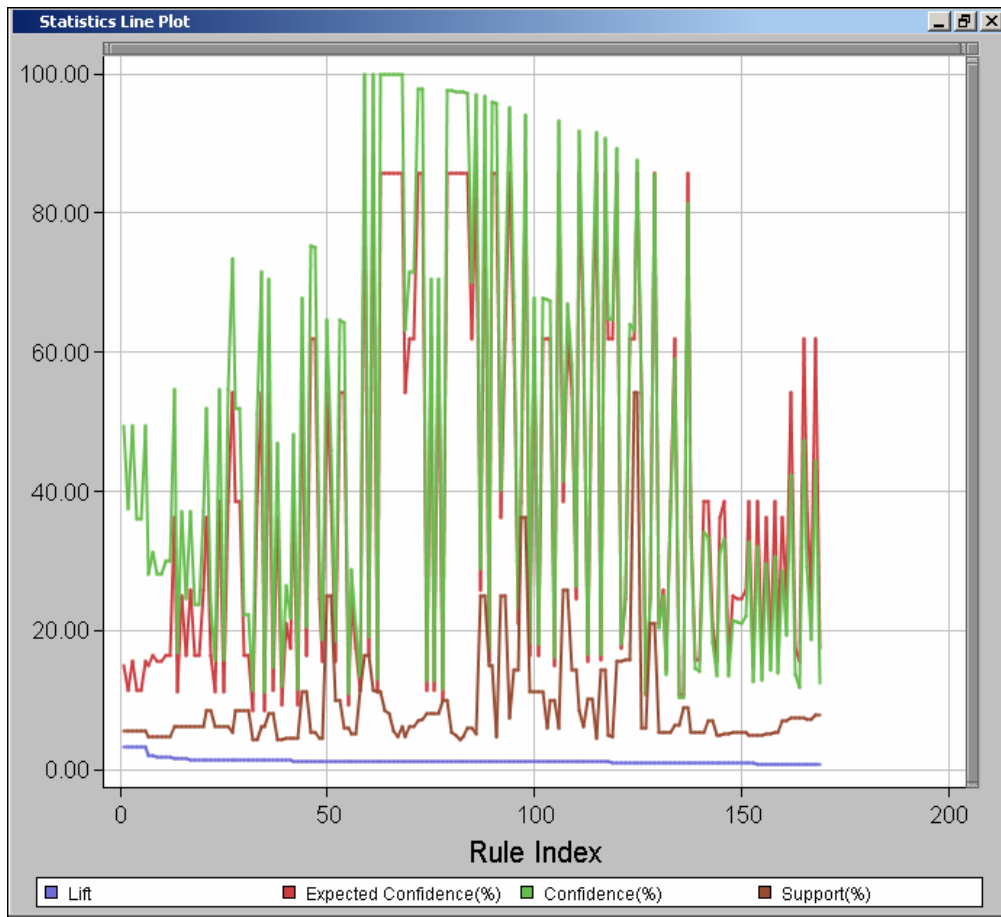
- Minimum confidence level, which specifies the minimum confidence level to generate a rule. The default level is 10%.
- Support Type, which specifies whether the analysis should use the support count or support percentage property. The default setting is Percent.
- Support Count, which specifies a minimum level of support to claim that items are associated (that is, they occur together in the database). The default count is 2.
- Support Percentage, which specifies a minimum level of support to claim that items are associated (that is, they occur together in the database). The default frequency is 5%.
- Maximum items, which determines the maximum size of the item set to be considered. For example, the default of four items indicates that a maximum of 4 items will be included in a single association rule.

 If you are interested in associations involving fairly rare products, you should consider reducing the support count or percentage when you run the Association node. If you obtain too many rules to be practically useful, you should consider raising the minimum support count or percentage as one possible solution.

21. Run the diagram from the Association node and view the results. The Statistics Line Plot, Rule Matrix, and Output windows are visible.



## 22. Maximize the Statistics Line Plot window.



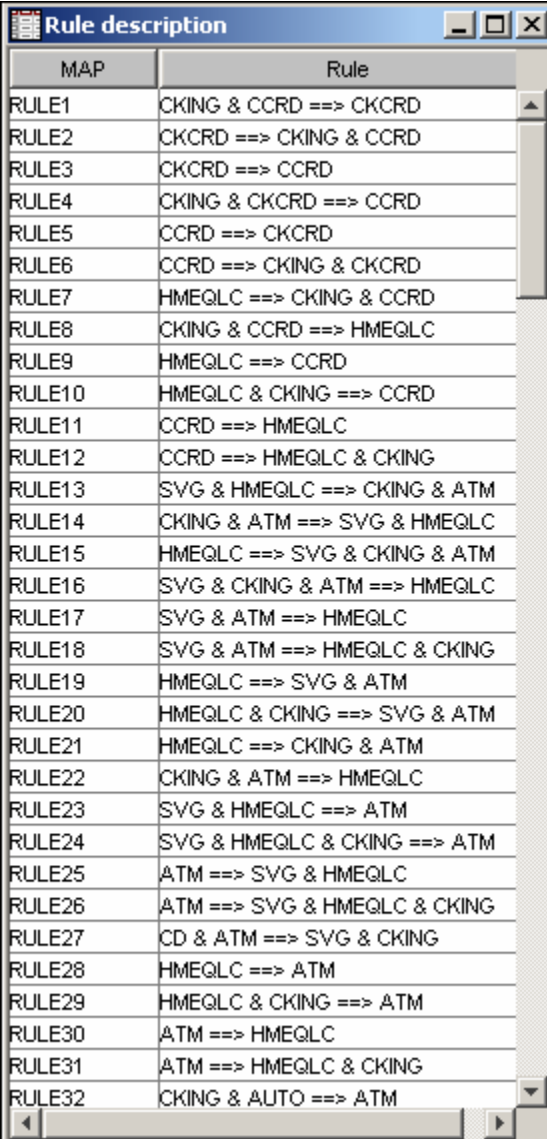
The Statistics Line Plot graphs the lift, expected confidence, confidence, and support for each of the rules by rule index number.

Consider the rule  $A \Rightarrow B$ . Recall that the

- support of  $A \Rightarrow B$  is the probability that a customer has both A and B.
- confidence of  $A \Rightarrow B$  is the probability that a customer has B given that the customer has A.
- expected confidence of  $A \Rightarrow B$  is the probability that a customer has B.
- lift of  $A \Rightarrow B$  is a measure of strength of the association. If the Lift=2 for the rule  $A \Rightarrow B$ , then a customer having A is twice as likely to have B than a customer chosen at random. Lift is the confidence divided by the expected confidence.

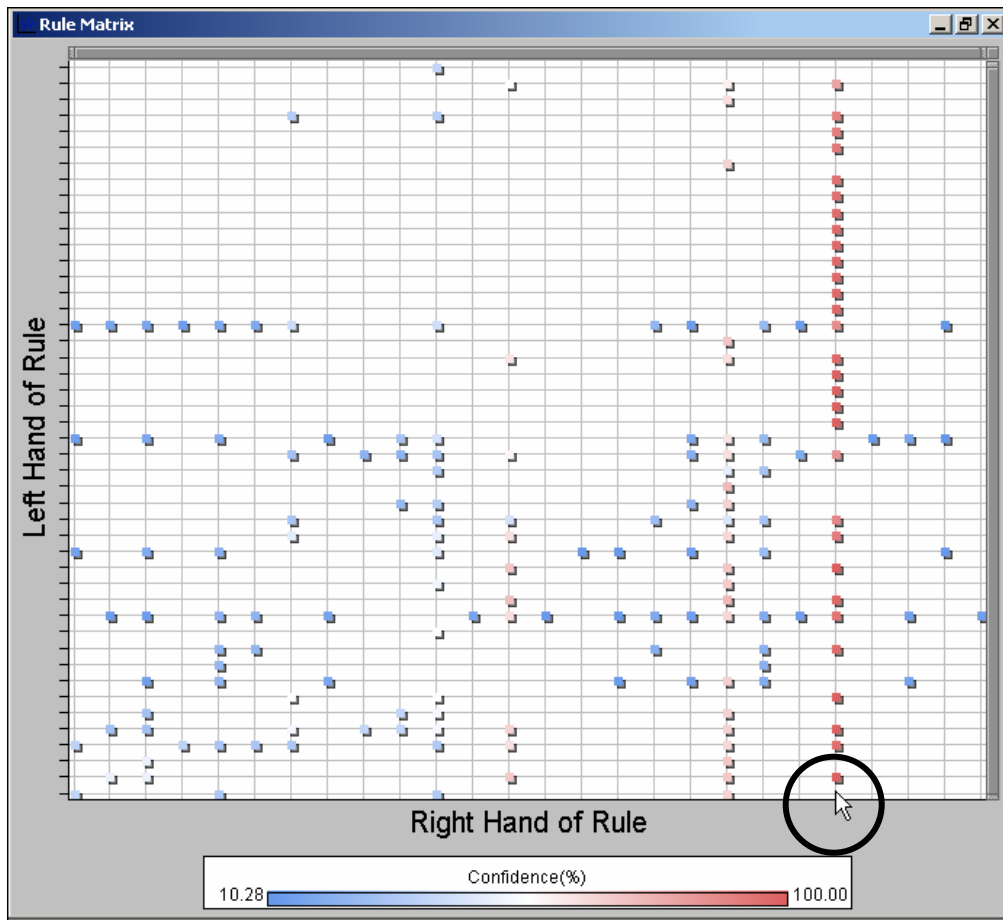
Notice that the rules are ordered in descending order of lift.

23. To view the descriptions of the rules, select **View** ⇒ **Rules** ⇒ **Rule description**.



MAP	Rule
RULE1	CKING & CCRD ==> CKCRD
RULE2	CKCRD ==> CKING & CCRD
RULE3	CKCRD ==> CCRD
RULE4	CKING & CKCRD ==> CCRD
RULE5	CCRD ==> CKCRD
RULE6	CCRD ==> CKING & CKCRD
RULE7	HMEQLC ==> CKING & CCRD
RULE8	CKING & CCRD ==> HMEQLC
RULE9	HMEQLC ==> CCRD
RULE10	HMEQLC & CKING ==> CCRD
RULE11	CCRD ==> HMEQLC
RULE12	CCRD ==> HMEQLC & CKING
RULE13	SVG & HMEQLC ==> CKING & ATM
RULE14	CKING & ATM ==> SVG & HMEQLC
RULE15	HMEQLC ==> SVG & CKING & ATM
RULE16	SVG & CKING & ATM ==> HMEQLC
RULE17	SVG & ATM ==> HMEQLC
RULE18	SVG & ATM ==> HMEQLC & CKING
RULE19	HMEQLC ==> SVG & ATM
RULE20	HMEQLC & CKING ==> SVG & ATM
RULE21	HMEQLC ==> CKING & ATM
RULE22	CKING & ATM ==> HMEQLC
RULE23	SVG & HMEQLC ==> ATM
RULE24	SVG & HMEQLC & CKING ==> ATM
RULE25	ATM ==> SVG & HMEQLC
RULE26	ATM ==> SVG & HMEQLC & CKING
RULE27	CD & ATM ==> SVG & CKING
RULE28	HMEQLC ==> ATM
RULE29	HMEQLC & CKING ==> ATM
RULE30	ATM ==> HMEQLC
RULE31	ATM ==> HMEQLC & CKING
RULE32	CKING & AUTO ==> ATM

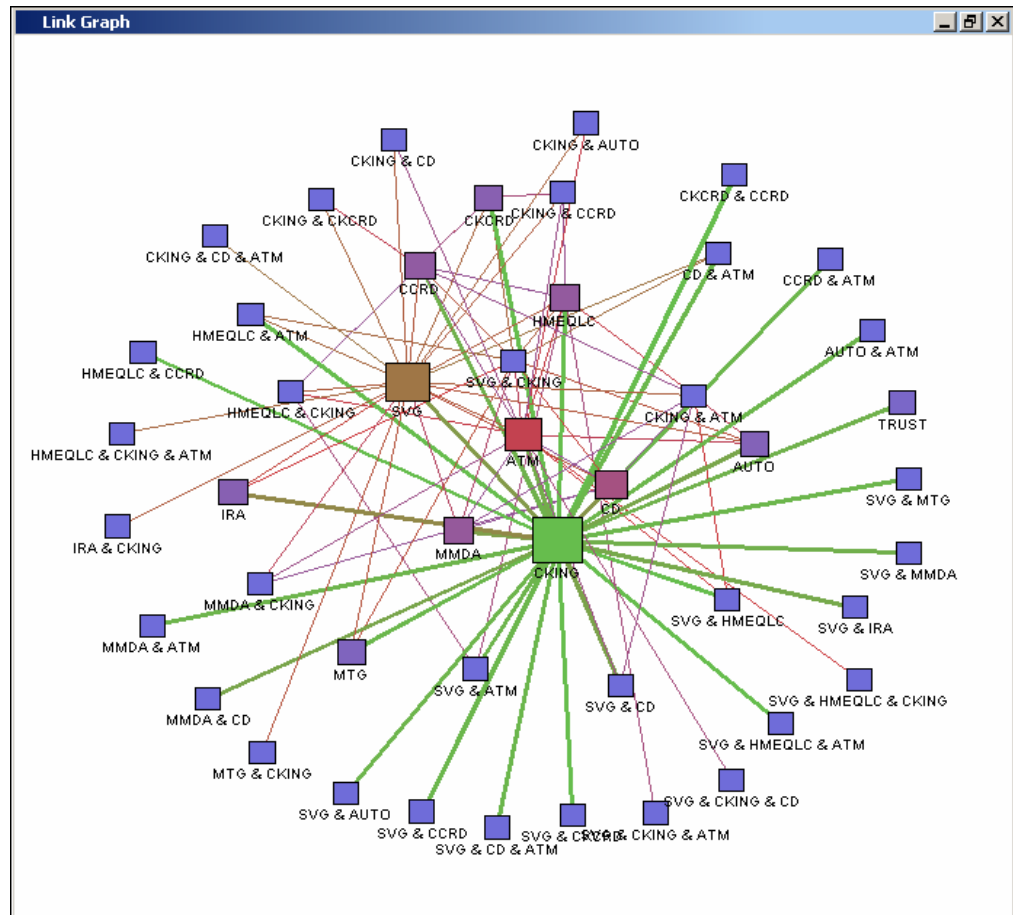
Examine the Rule Matrix.



The rule matrix plots the rules based on the items on the left side of the rule and the items on the right side of the rule. The points are colored based on the confidence of the rules. For example, the rules with the highest confidence are in the column indicated by the cursor in the picture above. Using the ActiveX feature of the graph, you discover that these rules all have checking on the right-hand side of the rule.

Finally, explore the associations by viewing the link graph.

24. To view the link graph, select **View** ⇒ **Rules** ⇒ **Link Graph**.

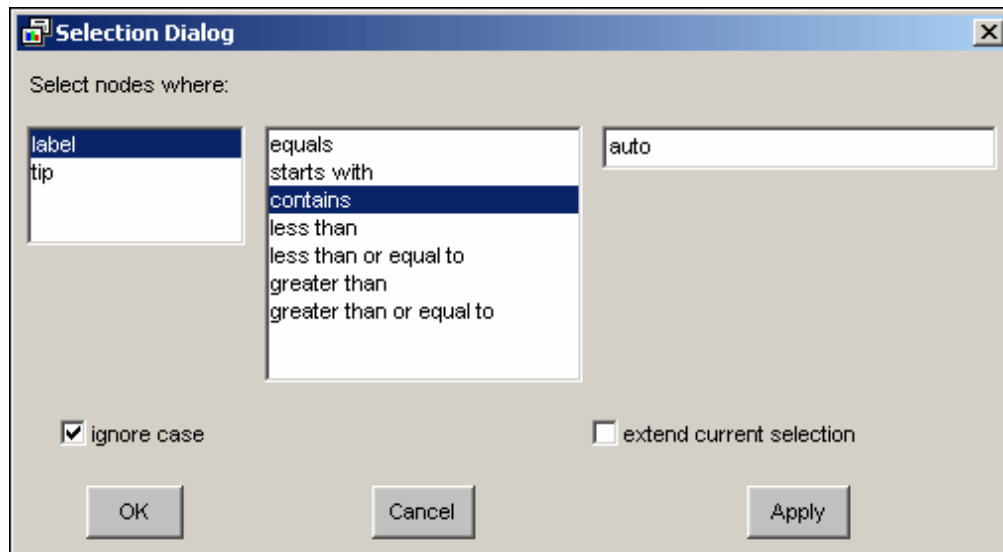


The link graph displays association results by using nodes and links. The size and color of a node indicate the transactions counts in the **Rules** data set. Larger nodes have greater counts than smaller nodes. The color and thickness of a link indicate the confidence level of a rule. The thicker the links are, the higher confidence the rules have.

Suppose you are particularly interested in those associations that involve automobile loans. One way to accomplish that visually in the link graph is to select those nodes whose label contains **AUTO** and then show only those links involving the selected nodes.

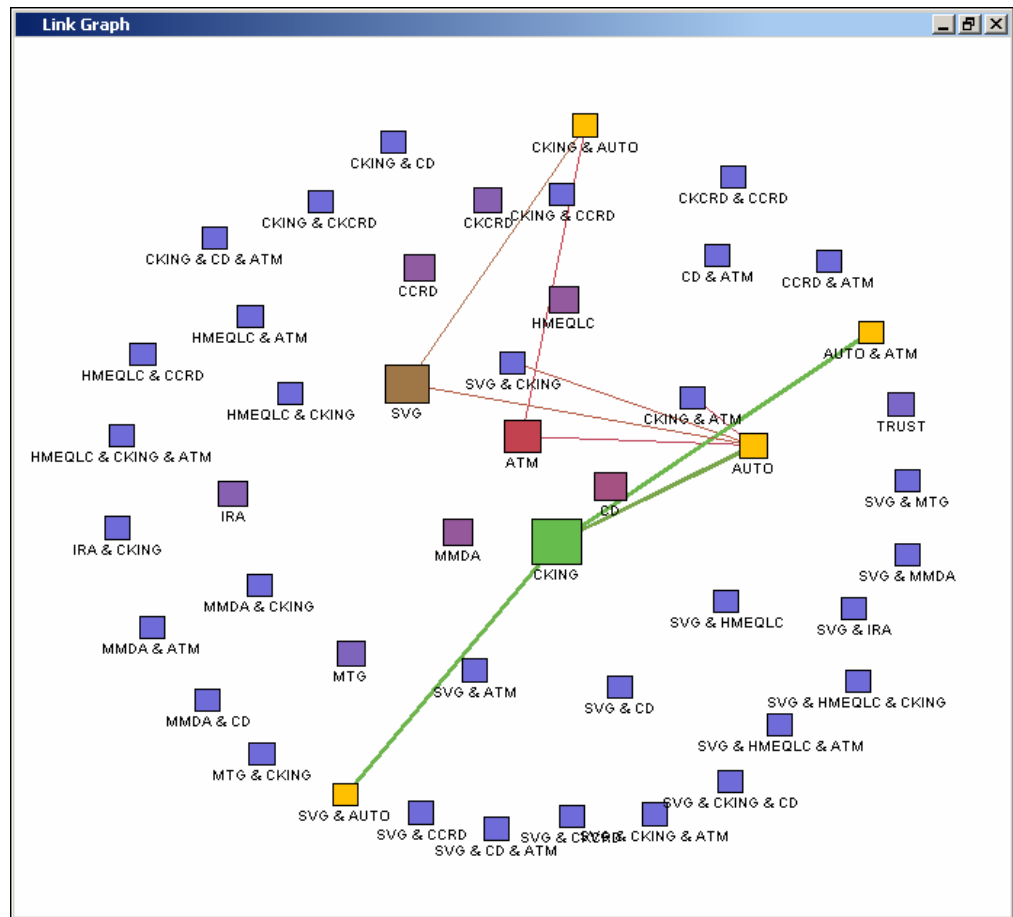
1. Right-click in the Link Graph window and select **Select...**.

2. In the Selection Dialog window, change the options to select the nodes where the label contains auto as shown below:



3. Select **OK**.

4. In the link graph, the nodes with auto are now selected. Right-click in the link graph and deselect **Show all links**.

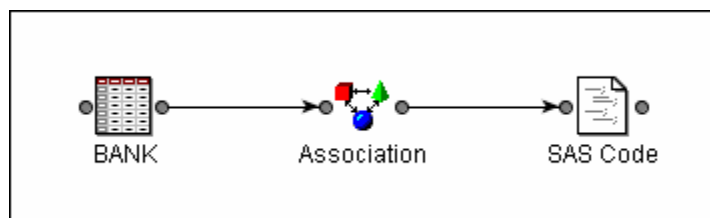



The only links shown are those that involve automobile loans.

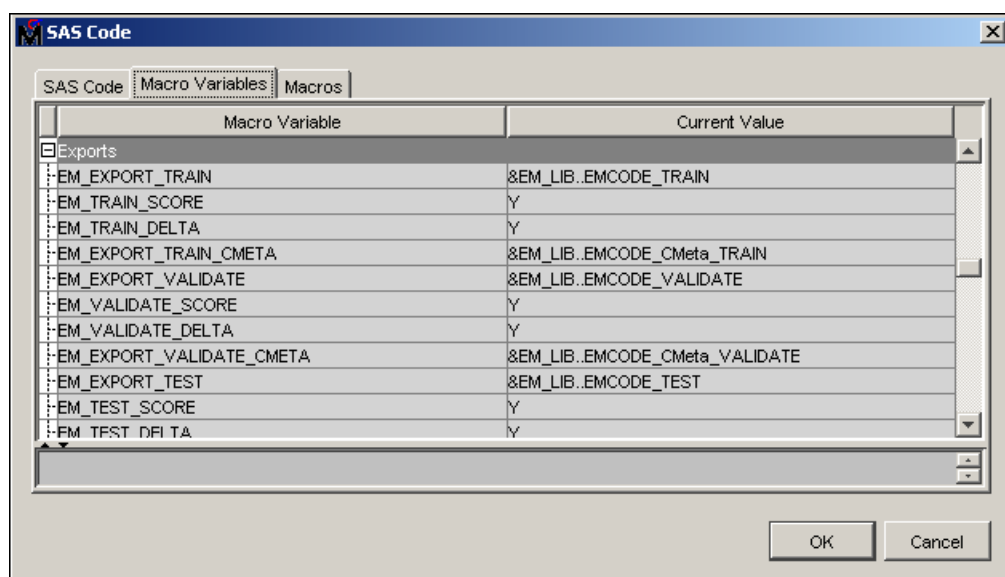


It is also possible that you would like to create a subset of the rules, to include only those rules with the product AUTO. Because the rules have been saved as a SAS data set, the easiest way to accomplish this goal is to use a SAS Code node to subset the data.

1. Close the Association node results window and add a SAS Code node to the diagram.




2. In the Property Panel, open the SAS Code window by clicking  in the SAS Code row.
3. Select the **Macro Variables** tab. Examine the contents of this tab and notice the many macro variables that SAS Enterprise Miner automatically creates.



4. Select the **SAS Code** tab.
5. Type in the following program:

```
Options ls=75;
data work.auto;
    set &EM_IMPORT_RULES;
    where _LHAND contains 'AUTO'
    or _RHAND contains 'AUTO';
run;

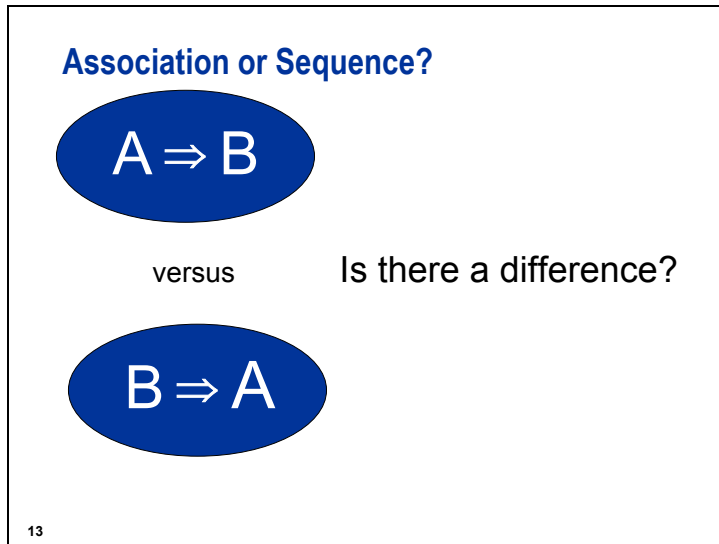
proc print data=work.auto;
run;
```

6. Select **OK** to close the SAS Code window.
7. Select the Run button, , to submit the program.
8. Select **Yes** to run the path.
9. Select **OK** to acknowledge that the run was completed.
10. Right-click on the SAS Code node and select **Results...** to view the output from the code.
11. Scroll down in the Output window to view the output from the PRINT procedure.

Obs	SET_SIZE	EXP_CONF	CONF	SUPPORT	LIFT	COUNT	RULE
1	3	38.46	51.10	4.35	1.33	348.00	CKING & AUTO ==> ATM
2	3	8.52	11.32	4.35	1.33	348.00	ATM ==> CKING & AUTO
3	3	9.29	12.03	4.35	1.30	348.00	CKING & ATM ==> AUTO
4	3	36.19	46.90	4.35	1.30	348.00	AUTO ==> CKING & ATM
5	2	38.46	48.11	4.47	1.25	357.00	AUTO ==> ATM
6	2	9.29	11.62	4.47	1.25	357.00	ATM ==> AUTO
7	3	54.17	64.29	5.97	1.19	477.00	AUTO ==> SVG & CKING
8	3	9.29	11.02	5.97	1.19	477.00	SVG & CKING ==> AUTO
9	3	85.78	97.48	4.35	1.14	348.00	AUTO & ATM ==> CKING
10	3	85.78	97.15	5.97	1.13	477.00	SVG & AUTO ==> CKING
11	3	61.87	70.04	5.97	1.13	477.00	CKING & AUTO ==> SVG
12	2	85.78	91.78	8.52	1.07	681.00	AUTO ==> CKING
13	2	61.87	66.17	6.14	1.07	491.00	AUTO ==> SVG

There are thirteen rules that involve automobile loans as shown in the partial output above.

12. Close the SAS Code node results when you are finished examining the output.



Association analysis is designed to determine the relationships between products offered for sale. In other words, what products are likely to appear together in a customer's basket?

Sequence analysis takes this a step further in that it examines the order in which products are purchased. This can be helpful in answering such questions as: if a customer purchased product A this week, is he likely to purchase product B next week?

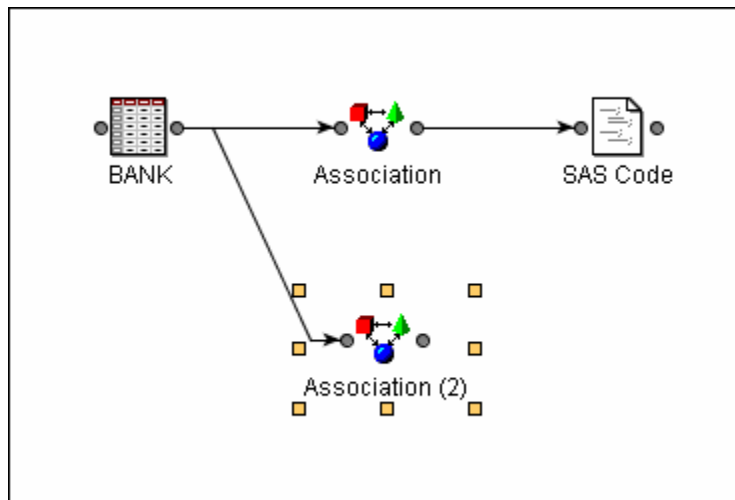
A sequence analysis requires the specification of a variable whose model role is sequence. An association analysis ignores a sequence variable.




## Sequence Analysis

In addition to the products owned by its customers, the bank is interested in examining the order in which the products are purchased. The sequence variable in the data set allows you to conduct a sequence analysis.

1. Add an Association node to the diagram workspace and connect it to the Input Data Source node.



2. Examine the Property Panel for the new Association node.
3. To examine the variables to be used in the analysis, click  in the Variables row.

**Variables - Assoc2**

Name	Use	Role	Level	Type	Order
ACCOUNT	Yes	ID	Interval	N	
SERVICE	Yes	Target	Nominal	C	
VISIT	Yes	Sequence	Nominal	N	

Explore... OK Cancel Help

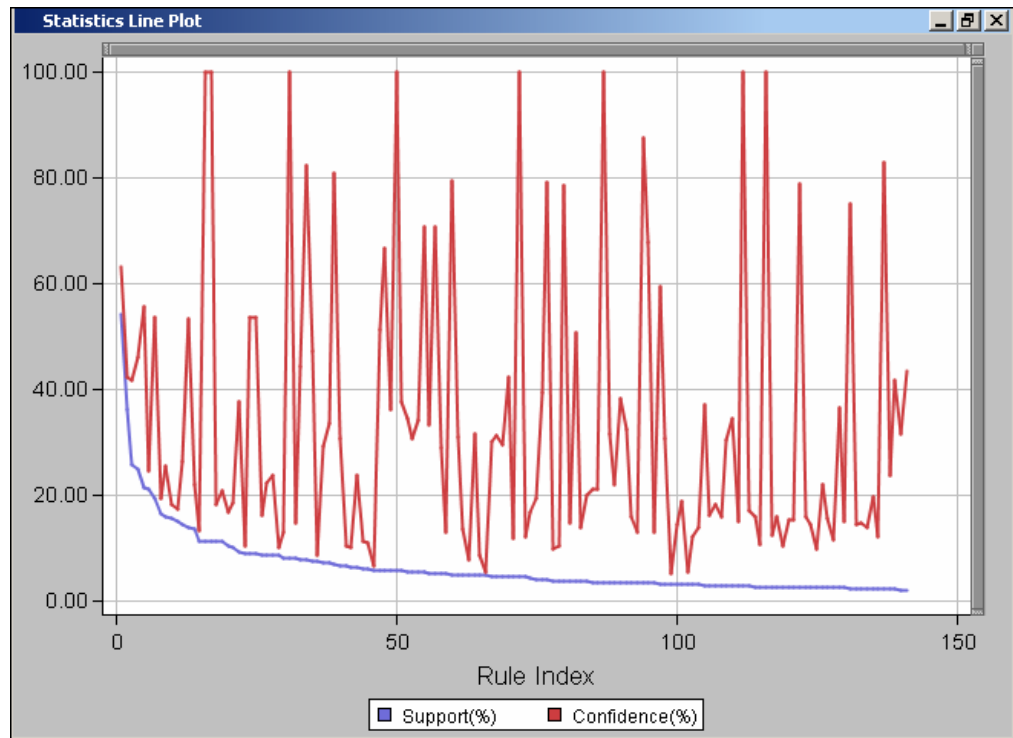
4. Because there is a sequence variable with Use set to Yes in the input data set, by default, the analysis mode will be a sequence analysis. Select **OK** to close the Variables window.
5. Set Export Rule by ID to **Yes**.
6. Examine the Sequence Panel in the Property Panel.

Sequence	
Chain Count	3
Consolidate Time	0.0
Maximum Transaction	
Support Type	Percent
Support Count	2
Support Percentage	2.0

The options in the Sequence Panel enable you to specify the following properties:

- Chain Count, which is the maximum number of items that can be included in a sequence. The default value is three and the maximum value is 10.
  - Consolidate Time, which enables you to specify whether consecutive visits to a location or consecutive purchases over a given interval can be consolidated into a single visit for analysis purposes. For example, two products purchased less than a day apart might be considered to be a single transaction.
  - Maximum Transaction Duration, which allows you to specify the maximum length of time for a series of transactions to be considered a sequence. For example, you might want to specify that the purchase of two products more than three months apart from each other does not constitute a sequence.
  - Support Type, which specifies whether the sequence analysis should use the Support Count or Support Percentage property. The default setting is Percent.
  - Support Count, which specifies the minimum frequency required to include a sequence in the sequence analysis when the Sequence Support Type is set to Count. If a sequence has a count less than the specified value, that sequence is excluded from the output. The default setting is 2.
  - Support Percentage, which specifies the minimum level of support to include the sequence in the analysis when the Sequence Support Type is set to Percentage. If a sequence has a frequency that is less than the specified percentage of the total number of transactions then that sequence is excluded from the output. The default percentage is 2%. Permissible values are real numbers between 0 and 100.
7. Leaving the default settings, run the diagram from the new Association node.
  8. After the run is completed, view the results.

## 9. Maximize the Statistics Line Plot window.



The Statistics Line Plot graphs the confidence and support for each of the rules by rule index number.

The percent support is the transaction count divided by the total number of customers, which would be the maximum transaction count. The percent confidence is the transaction count divided by the transaction count for the left side of the sequence.

10. To view the descriptions of the rules, select **View** ⇒ **Rules** ⇒ **Rule description**.

Rule description	
MAP	Rule
RULE1	CKING ==> SVG
RULE2	CKING ==> ATM
RULE3	SVG ==> ATM
RULE4	CKING ==> SVG ==> ATM
RULE5	ATM ==> ATM
RULE6	CKING ==> CD
RULE7	CKING ==> ATM ==> ATM
RULE8	CKING ==> HMEQLC
RULE9	SVG ==> CD
RULE10	CKING ==> MMDA
RULE11	CKING ==> CCRD
RULE12	CKING ==> SVG ==> CD
RULE13	SVG ==> ATM ==> ATM
RULE14	SVG ==> SVG
RULE15	CKING ==> CKCRD
RULE16	CKCRD ==> CKCRD
RULE17	CKING ==> CKCRD ==> CKCRD
RULE18	SVG ==> HMEQLC
RULE19	CKING ==> SVG ==> HMEQLC
RULE20	SVG ==> CCRD
RULE21	CKING ==> SVG ==> CCRD
RULE22	CD ==> CD
RULE23	CKING ==> IRA
RULE24	HMEQLC ==> HMEQLC
RULE25	CKING ==> HMEQLC ==> HMEQLC
RULE26	CKING ==> SVG ==> SVG
RULE27	ATM ==> HMEQLC
RULE28	CKING ==> ATM ==> HMEQLC
RULE29	CKING ==> AUTO

11. Close the sequence analysis results when you are finished examining the output.
12. Rename the second association analysis node to identify it as a sequence analysis. Right-click on the node and select **Rename**.
13. In the Input window that opens, type **Association Sequence** as the new name for the node.

The screenshot shows a dialog box titled "Input" with a question mark icon. It contains a text field labeled "Node Name:" with the text "Association Sequence" entered. Below the text field are two buttons: "OK" and "Cancel".

14. Select **OK**.

## 8.3 Dissociation Analysis (Self-Study)

### Objectives

- Define dissociation analysis.
- Generate a dissociation analysis within SAS Enterprise Miner.

16

### Dissociation Analysis

*Dissociation analysis* is used to determine what products do not appear together in market baskets.

17

A *dissociation rule* is a rule involving the negation of some item. For example, the left side may represent **no** checking account ( $\sim$ CKING) and the right side might be an auto loan. In other words, customers who do not have checking accounts tend to have auto loans. Dissociation rules may be particularly interesting when the items involved are highly prevalent.





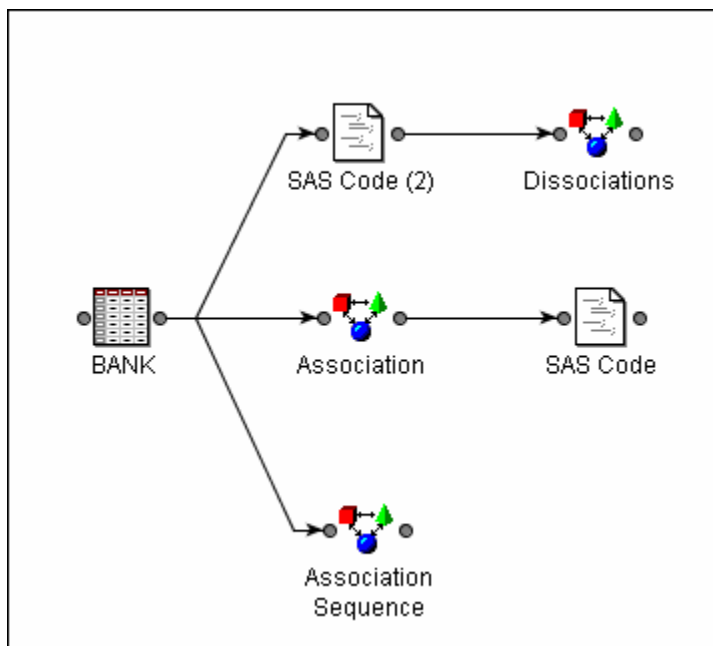
## Dissociation Analysis


The Association node will include dissociation rules if the data is modified to include the negation of selected items. The SAS Code node can be used for such data modification.

### Creating Dissociations

Augment the data with services not present in each account.

1. Add a SAS Code node to the workspace and connect it to the Input Data node.
2. Add another Association node to the diagram and connect it to the SAS Code node. Rename the new Association node as Dissociations.

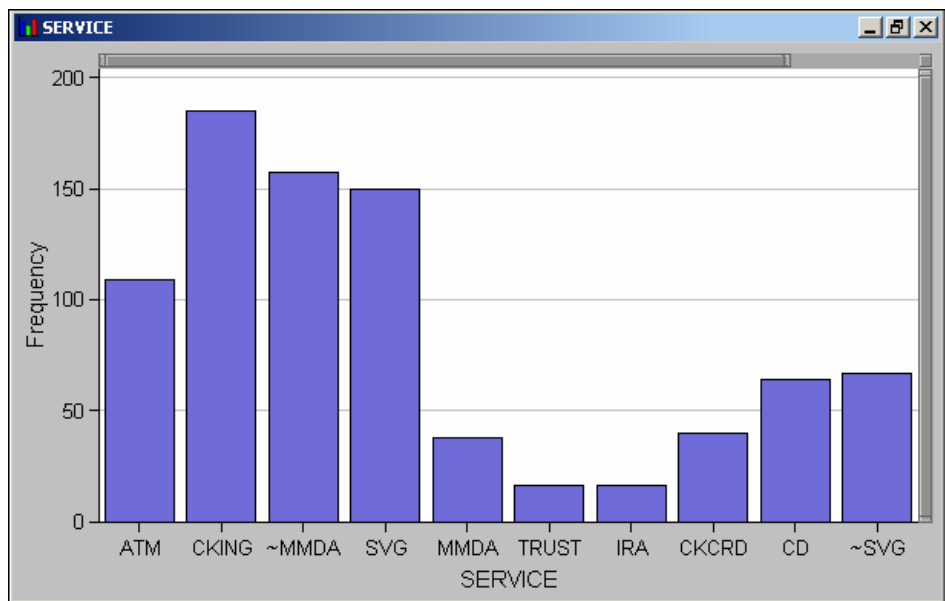


3. Select the new SAS Code node.
4. In the Training section of the Properties Panel, change the Code Location to **External File**.
5. In the External File field, type in the location and name of the external file to be used. For example, `c:\workshop\winsas\admt5\disso.c.sas`.
6. Run the SAS code. If you like, you may examine the log in the results to be sure there were no errors in the submitted code.
7. Select the Dissociations node in the diagram.
8. Examine the variables to be used in the analysis by clicking  in the Variables row of the Property Panel.

Variables - Assoc3							
Name	Use	Role	Level	Type	Order	Label	Format
ACCOUNT	Yes	ID	Interval	N		Account Number	BEST12.
SERVICE	Yes	Target	Nominal	C			
VISIT	Yes	Sequence	Nominal	N		Order of Service	BEST12.

Explore... OK Cancel Help

9. Change the Use status of the variable **VISIT** to **No**.
10. Explore the distribution of the variable **SERVICE** by selecting the row for that variable and then select **Explore...**.



Notice that the new values of **SERVICE** have been included in the data set for the association analysis.



### Modifying the Dissociations Code

The first three lines of the included code are as follows:

```
%let values= 'SVG' , 'CKING' , 'MMDA' ;  
%let in=&EM_IMPORT_TRANSACTION;  
%let out=&EM_EXPORT_TRANSACTION;
```

The first line identifies the values of the target for which negations are created. The values must be enclosed in quotes and separated by commas. This program scans each ID (ACCT) to see if the items (services) specified in the **values** are present. If not, the data is augmented with the negated items. In this case you will create negations only for savings accounts, checking accounts, and money market accounts.

The second and third lines provide macro names for the training data and the augmented (exported) data.

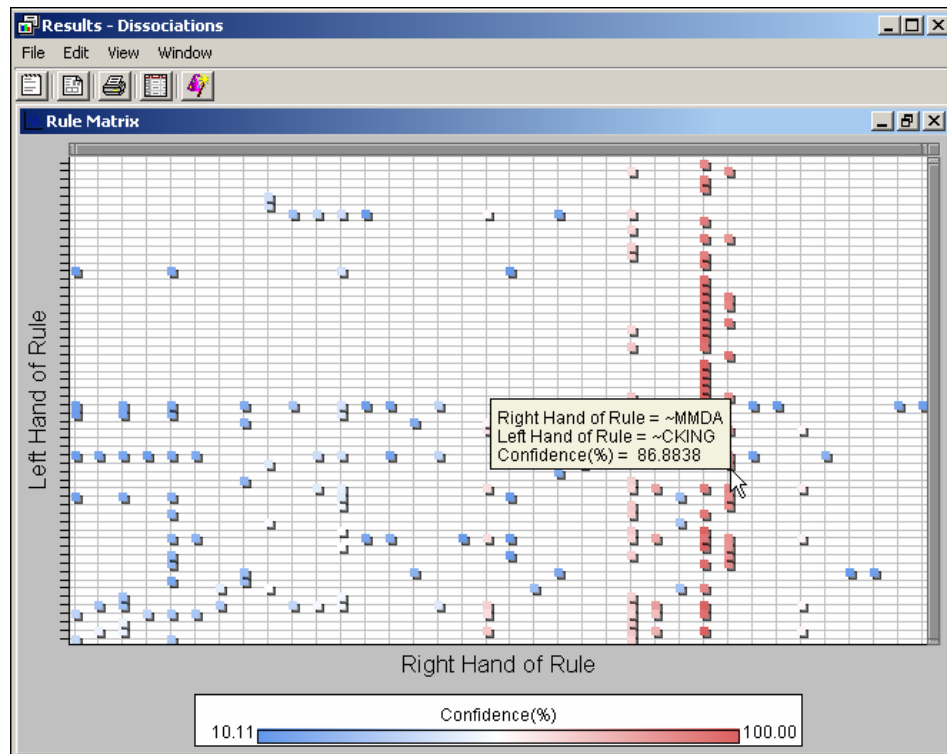
These three lines of code would need to be changed for analysis of a different data set with different products.

11. Close the Explore window.
12. Select **OK** to close the Variables window and continue with the analysis.
13. In the Property Panel, change the maximum items for an association to **3**.
14. Run the Dissociations node and view the results.



Because the use status of the sequence variable was set to No, the default analysis will be an association analysis rather than a sequence analysis. This is appropriate because there is no sequence to **not** purchasing a product.

The results now list association and dissociation rules. You can examine some of these rules by looking at the Rule Matrix.



For example, among customers without a checking account, 86.88% also do not have a money market account.

## 8.4 Exercises

### 1. Conducting an Association Analysis

A store is interested in determining the associations between items purchased from the health and beauty aids department and the stationary department. The store has chosen to conduct a market basket analysis of specific items purchased from these two departments. The **ASSOCIATIONS** data set contains information on over 400,000 transactions made over the past three months. The following products are represented in the data set:

- bar soap
- bows
- candy bars
- deodorant
- greeting cards
- magazines
- markers
- pain relievers
- pencils
- pens
- perfume
- photo processing
- prescription medications
- shampoo
- toothbrushes
- toothpaste
- wrapping paper.

There are four variables in the data set:

Name	Model Role	Measurement Level	Description
STORE	Rejected	Nominal	Identification Number of Store
TRANSACTION	ID	Nominal	Transaction Identification Number
PRODUCT	Target	Nominal	Product purchased
QUANTITY	Rejected	Interval	Quantity of this product purchased

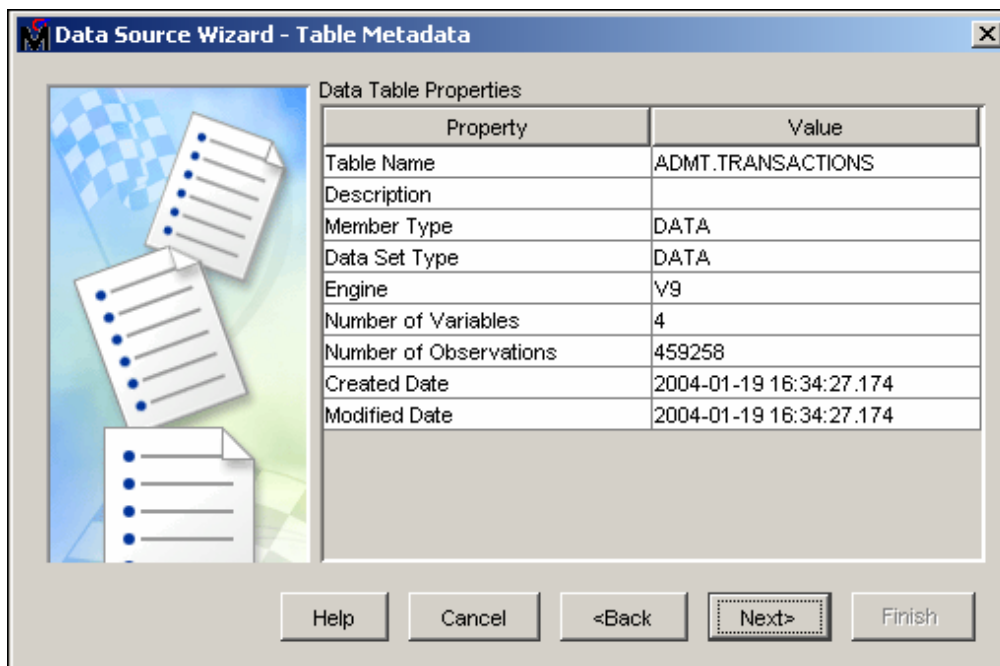
- a. Open a new diagram in your Exercise project. Name the diagram **Transactions**.
- b. Create a new Data Source for the data set **TRANSACTIONS**.

- c. Assign the variables **STORE** and **QUANTITY** the model role rejected. These variables will not be used in this analysis. Assign the ID model role to the variable **TRANSACTION** and the target model role to the variable **PRODUCT**.
- d. Add the node for the **TRANSACTIONS** data set and an Association node to the diagram.
- e. Change the setting for Export Rule by ID to Yes.
- f. Leave the remaining default settings for the Association node and run the analysis.
- g. Examine the results of the association analysis. What is the highest lift value for the resulting rules? Which rule has this value?
- h. You are particularly interested in the other products that individuals purchase when they come to your store to buy greeting cards. Use the link graph to examine the associations that involve greeting cards. Based on the rules, what are the other products these individuals are most likely to purchase?

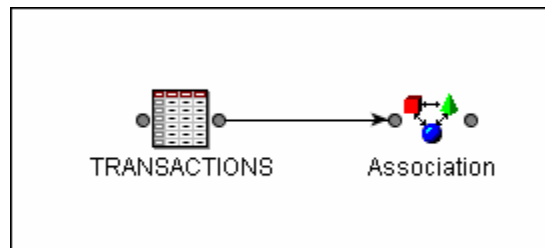
## 8.5 Solutions to Exercises

### 1. Conducting an Association Analysis

- a. Open a new diagram in the Exercise project.
  - 1) If the Exercise project is open, skip to step 3 below. If the Exercise project is **not** open, first open the project by selecting **File** ⇒ **Open Project...**
  - 2) Locate the correct server and expand the project list under that server. Double-click on **Exercise** to open the project.
  - 3) To open a new diagram in the project, select **File** ⇒ **New** ⇒ **Diagram...**
  - 4) Name the new diagram **Transactions** and select **OK**.
- b. Create a new Data Source for the data set **ADMT . TRANSACTIONS**.
  - 1) Right-click on **Data Sources** in the project tree and select **Create Data Source**.
  - 2) In the Data Source Wizard – Metadata Source window, be sure **SAS Table** is selected as the source and select **Next>**.
  - 3) Select **Browse...** to choose a data set.
  - 4) Double-click on the **ADMT** library and select the **TRANSACTIONS** data set.
  - 5) Select **OK**.
  - 6) Select **Next>**.



- 7) Examine the data table properties, and then select **Next>**.
- 8) Select **Advanced** to use the Advanced advisor, and then select **Next>**.
- c. Assign appropriate model roles to the variables.
  - 1) Control-click to select the rows for the variables **STORE** and **QUANTITY**. In the Role column of one of these rows, select **Rejected**.
  - 2) Select the **TRANSACTION** row and select **ID** as the role.
  - 3) Select the **PRODUCT** row and select **Target** as the role.
  - 4) Select **Next>**.
  - 5) To skip decision processing, select **Next>**.
  - 6) Change the Role to **Transaction**.
  - 7) Select **Finish**.
- d. Add the node for the **TRANSACTIONS** data set and an Association node to the diagram workspace. The workspace should appear as shown.

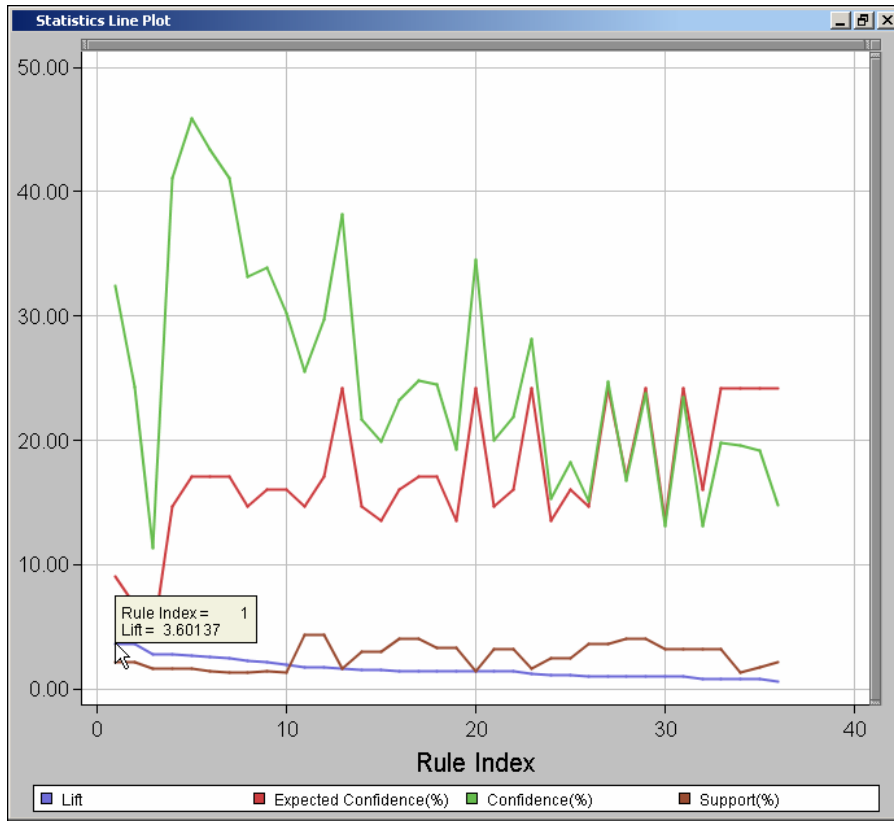


- e. Change the setting for Export Rule by ID to Yes.
  - 1) Select the Association node and examine the Property Panel.
  - 2) Set the Export Rule by ID property to **Yes**.
- f. Run the analysis.
  - 1) Right-click on the Association node and select **Run**.
  - 2) Select **Yes** to run the analysis when prompted.
  - 3) Select **OK** to confirm the completion of the run.



g. Examine the results of the association analysis.

- 1) Right-click on the Association node and select **Results...**
- 2) Examine the Statistics Line Plot.

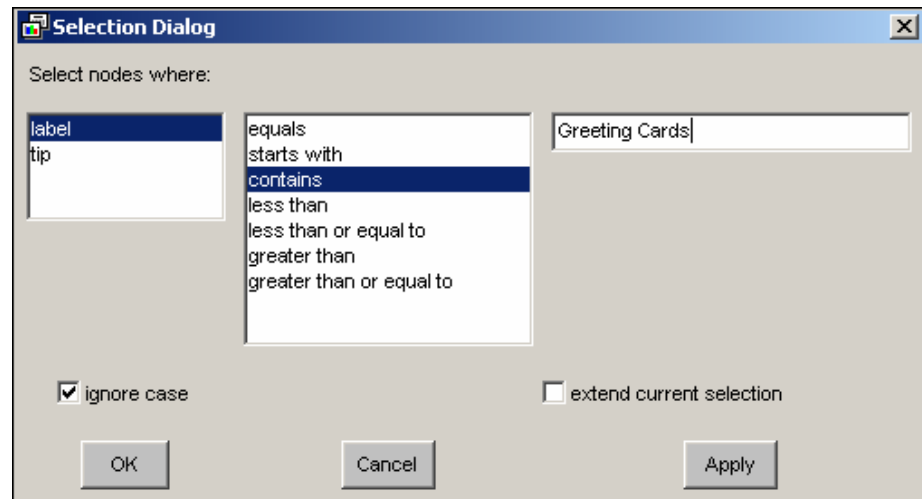


Rule 1 has the highest lift value, 3.60. Looking at the Output reveals that Rule 1 is the rule Toothbrush → Perfume.

h. Use the link graph to examine the associations that include greeting cards.

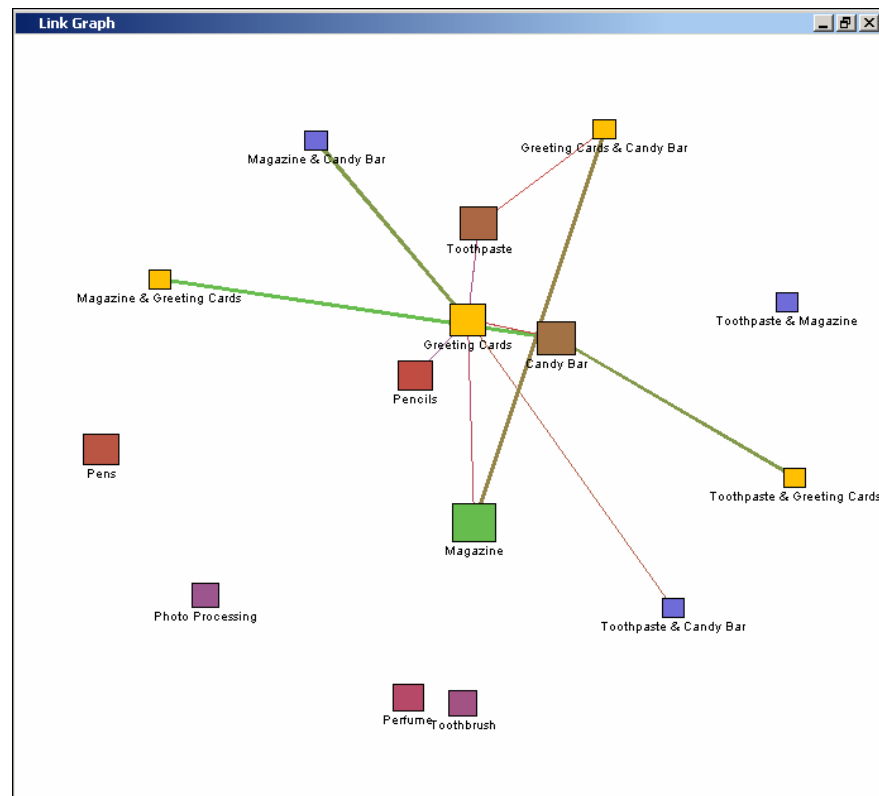
- 1) In the Results – Association window, select **View** ⇒ **Rules** ⇒ **Link Graph**.
- 2) Right-click in the Link Graph window and select **Select...**

- 3) In the Selection Dialog window, change the options to select the nodes where the label contains Greeting Cards as shown below:



- 4) Select **OK**.

- 5) In the link graph, right-click and deselect **Show all links**.



It appears that individuals who come to the store to purchase greeting cards are also likely to purchase candy bars, magazines, toothpaste, and pencils.

# Appendix A References

A.1	References .....	A-3
-----	------------------	-----



## A.1 References

- Beck, A. 1997. "Herb Edelstein discusses the usefulness of data mining." *DS Star*, Vol. 1, N0. 2. Available <http://www.tgc.com/dsstar/>.
- Berry, M. J. A. and G. Linoff. 1997. *Data Mining Techniques for Marketing, Sales, and Customer Support*. New York: John Wiley & Sons, Inc.
- Bigus, J. P. 1996. *Data Mining with Neural Networks: Solving Business Problems - from Application Development to Decision Support*. New York: McGraw-Hill.
- Bishop, C. M. 1995. *Neural Networks for Pattern Recognition*. New York: Oxford University Press.
- Breiman, L., et al. 1984. *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group.
- Chatfield, C. 1995. "Model uncertainty, data mining and statistical inference (with discussion)." *JRSS B* 419-466.
- Einhorn, H. J. 1972. "Alchemy in the behavioral sciences." *Public Opinion Quarterly* 36:367-378.
- Hand, D. J. 1997. *Construction and Assessment of Classification Rules*. New York: John Wiley & Sons, Inc.
- Hand, D. J. and W. E. Henley. 1997. "Statistical classification methods in consumer credit scoring: a review." *Journal of the Royal Statistical Society A* 160:523-541.
- Hand, David, Heikki Mannila, and Padraic Smyth. 2001. *Principles of Data Mining*. Cambridge, Massachusetts: The MIT Press.
- Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag New York, Inc.
- Hoaglin, D. C., F. Mosteller, and J. W. Tukey. 1983. *Understanding Robust and Exploratory Data Analysis*. New York: John Wiley & Sons, Inc.
- Huber, P. J. 1997. "From large to huge: A statisticians reaction to KDD and DM." *Proceedings, Third International Conference on Knowledge Discovery and Data Mining*. AAAI Press.
- John, G. H. 1997. *Enhancements to the Data Mining Process*. Ph.D. thesis, Computer Science Department, Stanford University.
- Kass, G. V. 1980. "An exploratory technique for investigating large quantities of categorical data." *Applied Statistics* 29:119-127.
- Little, R. J. A. and D. B. Rubin. 1987. *Statistical Analysis with Missing Data*. New York: John Wiley & Sons, Inc.
- Little, R. J. A. 1992. "Regression with missing X's: A review." *Journal of the American Statistical Association* 87:1227-1237.

- Lovell, M. C. 1983. "Data Mining." *The Review of Economics and Statistics*. Vol. LXV, number 1.
- Michie, D., D. J. Spiegelhalter, and C. C. Taylor. 1994. *Machine Learning, Neural and Statistical Classification*. New York: Ellis Horwood.
- Morgan, J. N. and J. A. Sonquist. 1963. "Problems in the analysis of survey data, and a proposal." *Journal of the American Statistical Association* 58:415-434.
- Mosteller, F and J. W. Tukey. 1977. *Data Analysis and Regression*. Reading, MA: Addison-Wesley.
- Palmeri, C. 1997. "Believe in yourself, believe in merchandise." *Forbes* Vol. 160, No. 5:118-124.
- Piatetsky-Shapiro, G. 1998. "What Wal-Mart might do with Barbie association rules." *Knowledge Discovery Nuggets*, 98:1. Available <http://www.kdnuggets.com/>.
- Quinlan, J. R. 1993. *C4.5 Programs for Machine Learning*. Morgan Kaufmann.
- Ripley, B. D. 1996. *Pattern Recognition and Neural Networks*. New York: Cambridge University Press.
- Rosenberg, E. and A. Gleit. 1994. "Quantitative methods in credit management." *Operations Research*, 42:589-613.
- Rud, Olivia Parr. 2001. *Data Mining Cookbook: Modeling Data, Risk, and Customer Relationship Management*. New York: John Wiley & Sons, Inc.
- Sarle, W.S. 1994a. "Neural Networks and Statistical Models," *Proceedings of the Nineteenth Annual SAS® Users Group International Conference*. Cary: NC, SAS Institute Inc., 1538-1550.
- Sarle, W.S. 1994b. "Neural Network Implementation in SAS® Software," *Proceedings of the Nineteenth Annual SAS® Users Group International Conference*. Cary: NC, SAS Institute Inc., 1550-1573.
- Sarle, W.S. 1995. "Stopped Training and Other Remedies for Overfitting." *Proceedings of the 27th Symposium on the Interface*.
- Sarle, W. S. 1997. "How to measure the importance of inputs." SAS Institute Inc. Available <ftp://ftp.sas.com/pub/neural/importance.html>.
- SAS Institute Inc. 1990. *SAS® Language: Reference, Version 6, First Edition*. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 1990. *SAS® Procedures Guide, Version 6, Third Edition*. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 1990. *SAS/STAT® User's Guide, Version 6, Fourth Edition, Volumes 1 and 2*. Cary, NC: SAS Institute Inc.
- SAS Institute Inc. 1995. *Logistic Regression Examples Using the SAS® System, Version 6, First Edition*. Cary, NC: SAS Institute Inc.

SAS Institute Inc. 1995. *SAS/INSIGHT® User's Guide, Version 6, Third Edition*. Cary, NC: SAS Institute Inc.

Smith, M. 1993. *Neural Networks for Statistical Modeling*. New York: Van Nostrand Reinhold.

Weiss, S.M. and C. A. Kulikowski. 1991. *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*. San Mateo, CA: Morgan Kaufmann.

Zhang, Heping, and Burton Singer. 1999. *Recursive Partitioning in the Health Sciences*. New York: Springer-Verlag New York, Inc.





# Appendix B Case Study

<b>B.1 Case Study Exercise .....</b>	<b>B-3</b>
<b>B.2 Solutions to Case Study Exercise.....</b>	<b>B-6</b>



## B.1 Case Study Exercise

### 1. Generating Predictive Models

A German bank is interested in evaluating the credit worthiness of prospective customers. They will use information on their current customers to develop a model to predict the probability of the customers having a bad credit rating status.

The data set **CREDIT** contains information on 1000 customers. There are 21 variables in the data set.

Name	Model Role	Measurement Level	Description
AGE	Input	Interval	Age in years
AMOUNT	Input	Interval	Amount of credit requested
CHECKING	Input	Nominal or Ordinal	Balance in existing checking account: 1 = less than 0 DM 2 = more than 0 but less than 200 DM 3 = at least 200 DM 4 = no checking account
COAPP	Input	Nominal	Other debtors or guarantors: 1 = none 2 = co-applicant 3 = guarantor
DEPENDS	Input	Interval	Number of dependents
DURATION	Input	Interval	Length of loan in months
EMPLOYED	Input	Ordinal	Time at present employment: 1 = unemployed 2 = less than 1 year 3 = at least 1, but less than 4 years 4 = at least 4, but less than 7 years 5 = at least 7 years
EXISTCR	Input	Interval	Number of existing accounts at this bank
FOREIGN	Input	Binary	Foreign worker: 1 = Yes 2 = No
GOOD_BAD	Target	Binary	Credit Rating Status (good or bad)

HISTORY	Input	Ordinal	Credit History: 0 = no loans taken / all loans paid back in full and on time 1 = all loans at this bank paid back in full and on time 2 = all loans paid back on time until now 3 = late payments on previous loans 4 = critical account / loans in arrears at other banks
HOUSING	Input	Nominal	Rent/Own: 1 = rent 2 = own 3 = free housing
INSTALLP	Input	Interval	Debt as a percent of disposable income
JOB	Input	Ordinal	Employment status: 1 = unemployed / unskilled non-resident 2 = unskilled resident 3 = skilled employee / official 4 = management / self-employed / highly skilled employee / officer
MARITAL	Input	Nominal	Marital status and gender 1 = male – divorced/separated 2 = female – divorced/separated/married 3 = male – single 4 = male – married/widowed 5 = female – single
OTHER	Input	Nominal or Ordinal	Other installment loans: 1 = bank 2 = stores 3 = none
PROPERTY	Input	Nominal or Ordinal	Collateral property for loan: 1 – real estate 2 = if not 1, building society savings agreement / life insurance 3 = if not 1 or 2, car or others 4 = unknown / no property

PURPOSE	Input	Nominal	Reason for loan request: 0 = new car 1 = used car 2 = furniture/equipment 3 = radio / television 4 = domestic appliances 5 = repairs 6 = education 7 = vacation 8 = retraining 9 = business x = other
RESIDENT	Input	Interval	Years at current address
SAVINGS	Input	Nominal or Ordinal	Savings account balance: 1 = less than 100 DM 2 = at least 100, but less than 500 DM 3 = at least 500, but less than 1000 DM 4 = at least 1000 DM 5 = unknown / no savings account
TELEPHON	Input	Binary	Telephone: 1 = none 2 = yes, registered under the customer's name

- a. Analyze this data set and build a model to predict the probability of a bad credit rating. Be prepared to share with the class what you did and why. (**Hint:** You want to model the probability of a bad credit rating. Because bad is before good alphanumerically, make sure the variable **GOOD\_BAD** is sorted in ascending order.)

## **B.2 Solutions to Case Study Exercise**

### **1. Generating Predictive Models**

- a.** The answers to this exercise will vary. The intent is to allow you to explore the analysis on your own and compare your results with other students.

# Appendix C Index

## A

- activation functions
  - neural networks, 5-5–5-7
- Analysis mode options, 8-13–8-14
- analytical expert, 1-10
- analytical tools, 1-7
- artificial neural networks, 5-5
- Assess nodes
  - SAS Enterprise Miner, 1-23
- association analysis
  - compared with sequence analysis, 8-23
  - Enterprise Miner, 8-9–8-22
  - overview, 8-4–8-6
- Association node
  - SAS Enterprise Miner, 1-18
- association rules, 8-4
- AutoNeural node
  - SAS Enterprise Miner, 1-21

## B

- backward selection method
  - Regression node, 3-7
- bagging models, 6-10
- Bayes rule, 2-70
- boosting models, 6-10

## C

- candidate models
  - comparing, 6-3, 6-5–6-8
- CART algorithm, 2-42
- case-control sampling, 2-24
- CHAID algorithm, 2-42
- chi-square criterion
  - Variable Selection node, 4-15–4-17
- choice-based sampling, 2-24
- classification trees, 2-37
- cluster analysis
  - k-means, 7-5–7-8
- Cluster node
  - SAS Enterprise Miner, 1-17
- clustering, 7-4

- k-means, 7-5–7-8

- confidence, 8-4
- Control Point node
  - SAS Enterprise Miner, 1-24
- credit risk management, 1-6
- credit scoring, 1-6
- curse of dimensionality, 2-28, 4-3
- customer relationship management, 1-6

## D

- data capacity, 8-6
- data expert, 1-10
- data mining
  - analytical tools, 1-7
  - definition, 1-3
  - KDD, 1-7
  - machine learning, 1-7–1-8
  - neurocomputing, 1-7–1-8
  - overview, 1-3–1-13
  - pattern recognition, 1-7
  - problem formulation, 1-9
  - problem translation, 1-11
  - required expertise, 1-10
  - steps, 1-9
- Data Partition node
  - SAS Enterprise Miner, 1-16, 2-34–2-35
- data splitting, 2-31
- data warehouses, 2-20
- database marketing, 1-6
- Decision Tree node
  - SAS Enterprise Miner, 1-20, 2-46–2-58
- decision trees
  - algorithms, 2-42
  - benefits, 2-43
  - building, 2-46–2-58
  - classification trees, 2-37
  - drawbacks, 2-44
  - fitted, 2-37
  - limiting growth, 2-57
  - options, 2-55

- possible splits, 2-39
- pruning, 2-41
- recursive partitioning, 2-38, 2-43
- regression trees, 2-37
- splitting criteria, 2-40
- stunting, 2-41
- variable selection, 4-4-4-5, 4-7-4-11

#### dissociation analysis

- Enterprise Miner, 8-29-8-32

- dissociation rules, 8-28

#### distributions

- inspecting, 3-14

#### Dmine Regression node

- SAS Enterprise Miner, 1-20

#### DMNeural node

- SAS Enterprise Miner, 1-21

- domain expert, 1-10

- Drop, 1-18

### E

#### ensemble models

- bagging, 6-10

- boosting, 6-10

- combined, 6-9-6-10, 6-11-6-13

#### Ensemble node

- SAS Enterprise Miner, 1-22

#### Enterprise Miner

- association analysis, 8-9-8-22

- combined ensemble models, 6-11-6-13

- comparing candidate models, 6-5-6-8

- dissociation analysis, 8-29-8-32

- fitting neural network models, 5-12-5-20

- k-means cluster analysis, 7-11-7-19

- sequence analysis, 8-24-8-27

- variable selection, 4-3-4-17

- Euclidean distance, 7-6

- expected confidence, 8-5

#### Explore nodes

- SAS Enterprise Miner, 1-17-1-18

### F

#### Filter node

- SAS Enterprise Miner, 1-18

- fitted decision trees, 2-37

- fitting decision trees, 3-35-3-36

- fitting models, 2-32-2-33

- forward selection method

  - Regression node, 3-7

- fraud detection, 1-6

### G

- generalized linear models, 5-9

### H

- healthcare informatics, 1-6

- hidden units

  - neural networks, 5-4

### I

- ID3 algorithm, 2-42

- Impute node

  - SAS Enterprise Miner, 1-19

- imputing missing values

  - methods, 3-22

  - reasons for, 3-6

- Input Data node

  - SAS Enterprise Miner, 1-16

- items, 8-4

### K

- KDD, 1-7

- k-means cluster analysis, 7-5-7-8

  - Enterprise Miner, 7-11-7-19

### L

- lift, 8-5

- linear regression

  - compared with logistic regression, 3-4

- link functions

  - neural networks, 5-7

- logistic regression

  - compared with linear regression, 3-4

  - discrimination, 5-10

  - logit transformation, 3-5

  - visualizing, 5-19

### M

- machine learning, 1-7-1-8

- Manhattan distance, 7-7



Memory-Based Reasoning node  
SAS Enterprise Miner, 1-21

Merge node  
SAS Enterprise Miner, 1-24

Metadata node  
SAS Enterprise Miner, 1-24

missing values, 2-26  
analysis strategies, 2-27  
imputing, 3-22–3-24  
reasons for imputing, 3-6

MLPs. *See* multilayer perceptrons (MLPs)

Model Comparison node  
SAS Enterprise Miner, 1-23

model roles, 2-13

models  
fitting, 2-32–2-33  
model complexity, 2-32  
overfitting, 2-33  
underfitting, 2-32

Modify nodes  
SAS Enterprise Miner, 1-18–1-19

multilayer perceptrons (MLPs), 5-5–5-8  
constructing, 5-14–5-15  
universal approximators, 5-7

Multiplot node  
SAS Enterprise Miner, 1-17

## N

Neural Network node  
SAS Enterprise Miner, 1-21

neural networks  
activation functions, 5-5–5-7  
artificial, 5-5  
hidden units, 5-4  
link functions, 5-7  
multilayer perceptrons, 5-5–5-8  
training, 5-8  
visualizing, 5-12–5-20

neurocomputing, 1-7–1-8

## O

outliers, 2-26  
oversampling, 2-24

## P

Path Analysis node  
SAS Enterprise Miner, 1-18

pattern recognition, 1-7

predictive modeling, 1-12

Principal Components node  
SAS Enterprise Miner, 1-19

## R

recursive partitioning, 2-38, 2-43

regression analysis, 1-13  
performing using SAS Enterprise Miner,  
3-11–3-36

Regression node, 3-4  
SAS Enterprise Miner, 1-20  
variable selection methods, 3-7

regression trees, 2-37

reject referencing, 2-25

ROC charts, 6-3–6-4

R-square criterion  
Variable Selection node, 4-12–4-15

Rule Induction node  
SAS Enterprise Miner, 1-20

## S

Sample node  
SAS Enterprise Miner, 1-16

Sample nodes  
SAS Enterprise Miner, 1-16

SAS Code node  
SAS Enterprise Miner, 1-24

SAS Enterprise Miner  
adding nodes, 2-16–2-17  
Assess nodes, 1-23  
building decision trees, 2-46–2-58  
building the initial flow, 2-16–2-17  
Data Partition node, 2-34–2-35  
Decision Tree node, 2-46–2-58  
decision tree options, 2-55  
Explore nodes, 1-17–1-18  
fitting decision trees, 3-35–3-36  
fitting regression models, 3-29–3-34  
identifying input data, 3-11–3-14  
identifying target variables, 3-14

- imputing missing values, 3-22–3-24
- inspecting distributions, 2-14, 3-14
- limiting decision trees growth, 2-57
- model roles, 2-13
- Modify nodes, 1-18–1-19
- modifying variable information, 2-15, 3-15
- opening, 2-4
- performing regressions using, 3-11–3-36
- Sample nodes, 1-16
- target variables, 2-14
- Utility nodes, 1-24
- variable transformations, 3-25–3-29

Score node, 6-17

- SAS Enterprise Miner, 1-23

scoring, 6-14

scoring code, 6-15, 6-17–6-24

Segment Profile node

- SAS Enterprise Miner, 1-23

sequence analysis

- compared with association analysis, 8-23
- Enterprise Miner, 8-24–8-27

StatExplore node

- SAS Enterprise Miner, 1-17

stepwise regression methods, 4-4

stepwise selection method

- Regression node, 3-7

supervised classification, 1-12–1-13, 7-10

support, 8-4

survival analysis, 1-13

## T

target marketing, 1-6

target variables, 2-14

- identifying, 3-14
- lack of, 2-23–2-24

temporal infidelity, 6-16

test data sets, 1-16, 2-31

Time Series node

- SAS Enterprise Miner, 1-16

training data sets, 1-12, 1-16, 2-31

training neural networks, 5-8

transaction count, 8-26

transactions, 8-4

Transform Variables node, 3-25

- SAS Enterprise Miner, 1-18

TwoStage node

- SAS Enterprise Miner, 1-21

## U

unsupervised classification, 7-4, 7-10

Utility nodes

- SAS Enterprise Miner, 1-24

## V

validation data sets, 1-16, 2-31

variable selection, 4-3–4-17

Variable Selection node, 4-4, 4-6, 4-11

- chi-square criterion, 4-15–4-17
- R-square criterion, 4-12–4-15
- SAS Enterprise Miner, 1-17

variables

- measurement levels, 2-13
- model roles, 2-13
- modifying information, 2-15
- target, 2-14
- types, 2-13