Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

support.sas.com/bookstore for additional books and resources.
## Contents

_About This Book_ .......................................................... vii  
_Recommended Reading_ ................................................ ix

**Chapter 1 • Introduction to SAS Enterprise Miner** .................................................. 1  
Starting SAS Enterprise Miner .................................................. 1  
Setting Up the Initial Project and Diagram ..................................... 1  
Identifying the Interface Components ........................................ 3  
Data Mining and SEMMA ..................................................... 4  
Accessing SAS Data through SAS Libraries ................................. 17

**Chapter 2 • Predictive Modeling** .................................................. 19  
Problem Formulation .......................................................... 20  
Creating a Process Flow Diagram ............................................. 21  
Data Preparation and Investigation ........................................... 33  
Fitting and Comparing Candidate Models ................................... 45  
Generating and Using Scoring Code ......................................... 54  
Generating a Report Using the Reporter Node ............................. 57

**Chapter 3 • Variable Selection** .................................................. 59  
Introduction to Variable Selection ............................................. 59  
Using the Variable Selection Node ............................................ 59

**Chapter 4 • Clustering Tools** .................................................. 63  
Problem Formulation .......................................................... 63  
Cluster Analysis .............................................................. 64

**Chapter 5 • Association Analysis** .................................................. 71  
Problem Formulation .......................................................... 71  
Association Analysis .......................................................... 71

**Chapter 6 • Link Analysis** ...................................................... 75  
Problem Formulation .......................................................... 75  
Examining Web Log Data ..................................................... 76
About This Book

Audience

This book is intended primarily for users who are new to SAS Enterprise Miner. The documentation assumes familiarity with graphical user interface (GUI) based software applications and basic, but not advanced, knowledge of data mining and statistical modeling principles. Although this knowledge is assumed, users who do not have this knowledge will still be able to complete the example that is described in this book end-to-end. In addition, SAS code is displayed in some result windows that are produced during the course of the example. However, SAS programming knowledge is not necessary to perform any task outlined in this book.
Recommended Reading

Here is the recommended reading list for this title:

• The online Help for SAS Enterprise Miner
• *Getting Started with SAS Enterprise Miner 13.1*
• “Customer Segmentation and Clustering Using SAS Enterprise Miner”
• “Data Preparation for Analytics Using SAS”
• “Predictive Modeling with SAS Enterprise Miner: Practical Solutions for Business Applications”
• “Decision Trees for Analytics Using SAS Enterprise Miner”

For a complete list of SAS books, go to support.sas.com/bookstore. If you have questions about which titles you need, please contact a SAS Book Sales Representative:

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-3228
Fax: 1-919-677-8166
E-mail: sasbook@sas.com
Web address: support.sas.com/bookstore
Recommended Reading
Chapter 1
Introduction to SAS Enterprise Miner

Starting SAS Enterprise Miner

To start SAS Enterprise Miner, click Start ⇒ SAS ⇒ SAS Enterprise Miner Client.
For more information, see Getting Started with SAS Enterprise Miner.

Setting Up the Initial Project and Diagram

SAS Enterprise Miner organizes data analysis into projects and diagrams. Each project can have several process flow diagrams, and each diagram can contain several analyses. Typically, each diagram contains an analysis of one data set.

To create a new project and diagram, complete the following steps:

1. On the menu bar, select File ⇒ New ⇒ Project.

2. In the Create New Project — Select a SAS Server window, specify the SAS server where this project is located. Click Next.

3. In the Create New Project — Specify Project Name and Server Directory window, specify a Project Name and the SAS Server Location. Click Next.
4. In the Create New Project — Register the Project window, the SAS Folder Location identifies where your project-specific information is stored. If you want to change this directory, click Browse and navigate to a different folder. Click Next.

5. The Create New Project — New Project Information window contains a summary of your new project. Click Finish.

6. To create a new diagram, select File $\Rightarrow$ New $\Rightarrow$ Diagram on the main menu.

7. In the Create New Diagram window, enter the name of your diagram. Click OK.
Identifying the Interface Components

Figure 1.1 SAS Enterprise Miner User Interface

Toolbar
The SAS Enterprise Miner Toolbar is a graphic set of node icons and tools that you use to build process flow diagrams in the Diagram Workspace. The text name of any node or tool icon is displayed when you position your mouse pointer over the button.

Toolbar Shortcut Buttons
The SAS Enterprise Miner toolbar shortcut icons are a graphic set of user interface tools that you use to perform common computer functions and frequently used SAS Enterprise Miner operations. The text name of any tool icon is displayed when you position your mouse pointer over the icon.
Project Panel
Use the Project Panel to manage and view data sources, diagrams, results, and project users.

Properties Panel
Use the Properties Panel to view and edit the settings of data sources, diagrams, nodes, results, and users.

Diagram Workspace
Use the Diagram Workspace to build, edit, run, and save process flow diagrams. In this workspace, you graphically build, order, and sequence the nodes that you use to mine your data and generate reports.

Property Help Panel
The Help Panel displays a short description of the property that you select in the Properties Panel. Extended help can be found on the Help main menu.

Data Mining and SEMMA

Definition of Data Mining
This document defines data mining as advanced methods for exploring and modeling relationships in large amounts of data.

Overview of the Data
A typical data set has many thousands of observations. An observation can represent an entity such as an individual customer, a specific transaction, or a certain household. Variables in the data set contain specific information such as demographic information, sales history, or financial information for each observation. How this information is used depends on the research question of interest.

When discussing types of data, consider the measurement level of each variable. You can generally classify each variable as one of the following:

- Interval — a continuous variable that contains values across a range. For these variables, the mean (or average) is interpretable. Examples include income, temperature, or height.

- Categorical — a classification variable with a finite number of distinct, discrete values. Examples include gender (male or female) and drink size (small, medium, large). Because these variables are noncontinuous, the mean is not interpretable. Categorical data can be grouped in several ways. SAS Enterprise Miner uses the following groupings:
  - Unary — a variable that has the same value for every observation in the data set.
  - Binary — a variable that has two possible values (for example, gender).
  - Nominal — a variable that has more than two levels, but the values of each level have no implied order. Examples are pie flavors such as cherry, apple, and peach.
  - Ordinal — a variable that has more than two levels. The values of each level have an implied order. Examples are drink sizes such as small, medium, and large.
Note: Ordinal variables can be treated as nominal variables when you are not interested in the ordering of the levels. However, nominal variables cannot be treated as ordinal variables because, by definition, there is no implied ordering.

To obtain a meaningful analysis, you must construct an appropriate data set and specify the correct measurement level for each variable in that data set.

**Predictive and Descriptive Techniques**

Predictive modeling techniques enable you to identify whether a set of input variables is useful in predicting some outcome, or target, variable. For example, a financial institution might try to determine whether knowledge of an applicant’s income and credit history (input variables) helps predict whether the client is likely to default on a loan (outcome variable).

To distinguish the input variables from the outcome variables, set the model for each variable in the data set. Identify outcome variables with the variable role **Target**, and identify input variables with the variable role **Input**. Other variable roles include **Cost**, **Frequency**, **ID**, and **Rejected**. The **Rejected** variable role specifies that the identified variable is not included in the model building process. The **ID** variable role indicates that the identified variable contains a unique identifier for each observation.

Predictive modeling techniques require one or more outcome variables of interest. Each technique attempts to predict the outcome as accurately as possible, according to a specified criterion such as maximizing accuracy or minimizing loss. This document shows you how to use several predictive modeling techniques through SAS Enterprise Miner, including regression models, decision trees, and neural networks. Each of these techniques enables you to predict a binary, nominal, ordinal, or continuous variable from any combination of input variables.

Descriptive techniques enable you to identify underlying patterns in a data set. These techniques do not have a specific outcome variable of interest. This document explores how to use SAS Enterprise Miner to perform the following descriptive analyses:

- Cluster analysis — This analysis attempts to find natural groupings of observations in the data, based on a set of input variables. After grouping the observations into clusters, you can use the input variables to attempt to characterize each group. When the clusters have been identified and interpreted, you can decide whether to treat each independently.

- Association analysis — This analysis identifies groupings of products or services that tend to be purchased at the same time, or at different times by the same customer. This analysis answers questions such as the following:
  - What proportion of the people who purchased eggs and milk also purchased bread?
  - What proportion of the people who have a car loan with some financial institution later obtain a home mortgage from the same institution?

**Overview of SEMMA**

SEMMA is an acronym used to describe the SAS data mining process. It stands for Sample, Explore, Modify, Model, and Assess. SAS Enterprise Miner nodes are arranged on tabs with the same names.

- **Sample** — These nodes identify, merge, partition, and sample input data sets, among other tasks.
Explore — These nodes explore data sets statistically and graphically. These nodes plot the data, obtain descriptive statistics, identify important variables, and perform association analysis, among other tasks.

Modify — These nodes prepare the data for analysis. Examples of the tasks that you can complete for these nodes are creating additional variables, transforming existing variables, identifying outliers, replacing missing values, performing cluster analysis, and analyzing data with self-organizing maps (SOMs) or Kohonen networks.

Model — These nodes fit a predictive model to a target variable. Available models include decision trees, neural networks, least angle regressions, support vector machines, linear regressions, and logistic regressions.

Assess — These nodes compare competing predictive models. They build charts that plot the percentage of respondents, percentage of respondents captures, lift, and profit.

SAS Enterprise Miner also includes the Utility and Applications tabs for nodes that provide necessary tools but that are not easily categorized on a SEMMA tab. One such example is the SAS Code node. This node enables you to insert custom SAS code into your process flow diagrams. Another example is the Score Code Export node, which exports the files that are necessary for score code deployment in production systems.

Overview of the Nodes

Sample Nodes

The Append node enables you to append data sets that are exported by two or more paths in a single SAS Enterprise Miner process flow diagram. The Append node can append data according to the data role, such as joining training data to training data, transaction data to transaction data, score data to score data, and so on. The Append node can append data that was previously partitioned in train, test, and validate roles into one large training data set.

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 weights during estimation and is also used for model assessment. The test data set is an additional hold-out data set that you can use for model assessment. This node uses simple random sampling, stratified random sampling, or user-defined partitions to create partitioned data sets.

The File Import node enables you to import data that is stored in external formats into a data source that SAS Enterprise Miner can interpret. The File Import node currently can
process CSV flat files, JMP tables, Microsoft Excel and Lotus spreadsheet files, Microsoft Access database tables, and DBF, DLM, and DTA files.

The **Filter** node enables you to apply a filter to the training data set in order to exclude outliers or other observations that you do not want to include in your data mining analysis. Outliers can greatly affect modeling results and, subsequently, the accuracy and reliability of trained models.

The **Input Data** node enables you to access SAS data sets and other types of data. The **Input Data** node represents the introduction of predefined metadata into a Diagram Workspace for processing. You can view metadata information about your source data in the **Input Data** node, such as initial values for measurement levels and model roles of each variable.

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. In addition, you have the option to rename certain variables (for example, predicted values and posterior probabilities) depending on the settings of the node.

The **Sample** node enables you to take random, stratified random, and cluster samples of data sets. Sampling is recommended for extremely large databases because it can significantly decrease model training time. If the sample is sufficiently representative, then relationships found in the sample can be expected to generalize to the complete data set.

The **Time Series** node enables you to understand trends and seasonal variations in the transaction data that you collect from your customers and suppliers over the time, by converting transactional data into time series data. Transactional data is timestamped data that is collected over time at no particular frequency. By condensing the information into a time series, you can discover trends and seasonal variations in customer and supplier habits that might not be visible in transactional data.
Exploring Nodes

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

The **Cluster** node enables you to segment your data by grouping observations that are statistically similar. 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 other tools for use as an input, ID, or target variable. It can also be used as a group variable that enables automatic construction of separate models for each group.

The **DMDB** node creates a data mining database that provides summary statistics and factor-level information for class and interval variables in the imported data set. The DMDB is a metadata catalog that is used to store valuable counts and statistics for model building.

The **Graph Explore** node is an advanced visualization tool that enables you to explore large volumes of data graphically to uncover patterns and trends and to reveal extreme values in the database. For example, you can analyze univariate distributions, investigate multivariate distributions, and create scatter and box plots and constellation and 3-D charts. Graph Explore plots are fully interactive and are dynamically linked to highlight data selections in multiple views.

The **Link Analysis** node transforms unstructured transactional or relational data into a model that can be graphed. Such models can be used to discover fraud detection, criminal network conspiracies, telephone traffic patterns, website structure and usage, database visualization, and social network analysis. Also, the node can be used to recommend new products to existing customers.
The Market Basket node performs association rule mining over transaction data in conjunction with item taxonomy. This node is useful in retail marketing scenarios that involve tens of thousands of distinct items, where the items are grouped into subcategories, categories, departments, and so on. This is called item taxonomy. The Market Basket node uses the taxonomy data and generates rules at multiple levels in the taxonomy.

The MultiPlot node is a visualization tool that enables you to explore larger volumes of data graphically. The MultiPlot node automatically creates bar charts and scatter plots for the input and target variables without making several menu or window item selections. The code that is created by this node can be used to create graphs in a batch environment.

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

The SOM/Kohonen node enables you to perform unsupervised learning by using Kohonen vector quantization (VQ), Kohonen self-organizing maps (SOMs), or batch SOMs with Nadaraya-Watson or local-linear smoothing. Kohonen VQ is a clustering method, but SOMs are primarily dimension-reduction methods.

The StatExplore node is a multipurpose node that you use to examine variable distributions and statistics in your data sets. Use the StatExplore node to compute standard univariate statistics, to compute standard bivariate statistics by class target and class segment, and to compute correlation statistics for interval variables by interval input and target. You can also use the StatExplore node to reject variables based on target correlation.
The **Variable Clustering** node is a useful tool for selecting variables or cluster components for analysis. Variable clustering removes collinearity, decreases variable redundancy, and helps reveal the underlying structure of the input variables in a data set. Large numbers of variables can complicate the task of determining the relationships that might exist between the independent variables and the target variable in a model. Models that are built with too many redundant variables can destabilize parameter estimates, confound variable interpretation, and increase the computing time that is required to run the model. Variable clustering can reduce the number of variables that are required to build reliable predictive or segmentation models.

The **Variable Selection** node enables you to evaluate the importance of input variables in predicting or classifying the target variable. The node uses either an R-square or a Chi-square selection (tree-based) criterion. The R-square criterion removes variables that have large percentages of missing values, and removes 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 tools in the process flow diagram, these variables are not used as model inputs by modeling nodes such as the Neural Network and Decision Tree tools. If a variable interest is rejected, you can force that variable into the model by reassigning the variable role in any modeling node.

**Modify Nodes**

The **Drop** node enables you to drop selected variables from your scored SAS Enterprise Miner data sets. You can drop variables that have the roles of Assess, Classification, Frequency, Hidden, Input, Rejected, Residual, and Target from your scored data sets. Use the **Drop** node to trim the size of data sets and metadata during tree analysis.

The **Impute** node enables you to replace missing values for interval variables with the mean, median, midrange, mid-minimum spacing, distribution-based replacement, or use a replacement M-estimator such as Tukey's biweight, Hubers, or Andrew's Wave, or 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 **Interactive Binning** node is used to model nonlinear functions of multiple modes of continuous distributions. The interactive tool computes initial bins by quintiles, and then you can split and combine the initial quintile-based bins into custom final bins.
The Principal Components node enables you to perform a principal components analysis for data interpretation and dimension reduction. The node generates principal components that are uncorrelated linear combinations of the original input variables and that depend on the covariance matrix or correlation matrix of the input variables. In data mining, principal components are usually used as the new set of input variables for subsequent analysis by modeling nodes.

The Replacement node enables you to replace selected values for class variables. The Replacement node summarizes all values of class variables and provides you with an editable variables list. You can also select a replacement value for future unknown values.

The Rules Builder node enables you to create ad hoc sets of rules for your data that result in user-definable outcomes. For example, you might use the Rules Builder node to define outcomes named Deny and Review based on rules such as the following:

```
IF P Default Yes > 0.4 then do
  EM_OUTCOME="Deny";
IF AGE > 60 then
  EM_OUTCOME="Review";
END;
```

The Transform Variables node enables you to create new variables that are transformations of existing variables in your data. Transformations can be used to stabilize variances, remove nonlinearity, improve additivity, and correct nonnormality in variables. You can also use the Transform Variables node to transform class variables and to create interaction variables. Examples of variable transformations include taking the square root of a variable, maximizing the correlation with the target variable, or normalizing a variable.

Model Nodes

The AutoNeural node can be used to automatically configure a neural network. The AutoNeural node implements a search algorithm to incrementally select activation functions for a variety of multilayer networks.
The **Decision Tree** node enables you to fit decision tree models to your data. The implementation includes features that are found in a variety of popular decision tree algorithms (for example, CHAID, CART, and C4.5). 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 can be used to select variables for use in subsequent modeling. You can override any automatic step with the option to define a splitting rule and prune explicit tools or subtrees. Interactive training enables you to explore and evaluate data splits as you develop them.

The **Dmine Regression** node enables you to compute a forward stepwise, least squares regression model. In each step, the independent variable that contributes maximally to the model R-square value is selected. The tool can also automatically bin continuous terms.

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 with automatic selection of an activation function.

The **Ensemble** node enables you to create new models by combining the posterior probabilities (for class targets) or the predicted values (for interval targets) from multiple predecessor models.

The **Gradient Boosting** node uses tree boosting to create a series of decision trees that together form a single predictive model. Each tree in the series is fit to the residual of the prediction from the earlier trees in the series. The residual is defined in terms of the derivative of a loss function. For squared error loss with an interval target, the residual is simply the target value minus the predicted value. Boosting is defined for binary, nominal, and interval targets.
The **LARS** node enables you to use Least Angle Regression algorithms to perform variable selection and model fitting tasks. The **LARS** node can produce models that range from simple intercept models to complex multivariate models that have many inputs. When the **LARS** node is used to perform model fitting, it uses criteria from either least angle regression or the LASSO regression to choose the optimal model.

The **MBR** (Memory-Based Reasoning) node enables you to identify similar cases and to apply information that is obtained from these cases to a new record. The **MBR** node uses k-nearest neighbor algorithms to categorize or predict observations.

The **Model Import** node enables you to import models into the SAS Enterprise Miner environment that were not created by SAS Enterprise Miner. For example, models that were created by using SAS PROC LOGISTIC can now be run, assessed, and modified in SAS Enterprise Miner.

The **Neural Network** node enables you to construct, train, and validate multilayer feedforward neural networks. Users can select from several predefined architectures or manually select input, hidden, and target layer functions and options.

The **Partial Least Squares** node is a tool for modeling continuous and binary targets based on SAS/STAT PROC PLS. The **Partial Least Squares** node produces DATA step score code and standard predictive model assessment results.

The **Regression** node enables you to fit both linear and logistic regression models to your data. You can use continuous, ordinal, and binary target variables. You can use both continuous and discrete variables as inputs. The node supports the stepwise, forward, and backward selection methods. A point-and-click interaction builder enables you to create higher-order modeling terms.
The **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 that 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. After all levels of the target variable are modeled, the score code is combined into a SAS DATA step.

The **SVM** node uses supervised machine learning to perform binary classification problems, including polynomial, radial basis function, and sigmoid nonlinear kernels. The standard SVM problem solves binary classification problems by constructing a set of hyperplanes that maximize the margin between two classes. The **SVM** node does not support multi-class problems or support vector regression.

The **TwoStage** node enables you to compute a two-stage model for predicting a class and interval target variables at the same time. The interval target variable is usually a value that is associated with a level of the class target.

**Assess Nodes**

The **Cutoff** node provides tabular and graphical information to help you determine the best cutoff point or points for decision making models that have binary target variables.

The **Decisions** node enables you to define target profiles to produce optimal decisions. You can define fixed and variable costs, prior probabilities, and profit or loss matrices. These values are used in model selection steps.

The **Model Comparison** node provides a common framework for comparing models and predictions from any of the modeling tools (such as Regression, Decision Tree, and Neural Network tools). The comparison is based on standard model fit statistics as well as potential expected and actual profits or losses that would result from implementing
the model. The node produces the following charts that help describe the usefulness of
the model: lift, profit, return on investment, receiver operating curves, diagnostic charts,
and threshold-based charts.

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 cannot 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 Segment Profile node enables you to assess and explore segmented data sets.
Segmented data is created from data BY values, clustering, or applied business rules.
The Segment Profile node facilitates data exploration to identify factors that
differentiate individual segments from the population, and to compare the distribution of
key factors between individual segments and the population. The Segment Profile node
outputs a Profile plot of variable distributions across segments and the population, a
Segment Size pie chart, a Variable Worth plot that ranks factor importance within each
segment, and summary statistics for the segmentation results. The Segment Profile node
does not generate score code or modify metadata.

Utility Nodes

The Control Point node establishes a nonfunctional connection point to clarify and
simplify process flow diagrams. For example, suppose three Input Data 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 nodes to all of the modeling
nodes. However, if a Control Point node is used, only six connections are required.

The End Groups node terminates a group processing segment in the process flow
diagram. If the group processing function is stratified, bagging, or boosting, the Ends
Groups node will function as a model node and present the final aggregated model.
(Ensemble nodes are not required as in SAS Enterprise Miner 4.3.) Nodes that follow the
Ends Groups node continue data mining processes normally.
The **ExtDemo** node illustrates the various UI elements that can be used by SAS Enterprise Miner extension 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 **Reporter** node tool uses SAS Output Delivery System (ODS) capabilities to create a single document for the given analysis in PDF or RTF format. The document includes important SAS Enterprise Miner results, such as variable selection, model diagnostic tables, and model results plots. The document can be viewed and saved directly and will be included in SAS Enterprise Miner report package files.

The **SAS Code** node tool enables you to incorporate SAS code into process flows that you develop using SAS Enterprise Miner. The **SAS Code** node extends the functionality of SAS Enterprise Miner by making other SAS System procedures available in your data mining analysis. You can also write a SAS DATA step to create customized scoring code, to conditionally process data, and to concatenate or to merge existing data sets.

The **Score Code Export** node tool enables you to extract score code and score metadata to an external folder. The **Score Code Export** node must be preceded by a Score node.

The **Start Groups** node initiates a group processing segment in the process flow diagram. The **Start Groups** node performs the following types of group processing:

- **Stratified** group processing that repeats processes for values of a class variable, such as GENDER=M and GENDER=F.
- **Bagging**, or bootstrap aggregation via repeated resampling.
- **Boosting**, or boosted bootstrap aggregation, using repeated resampling with residual-based weights.
- **Index processing**, which repeats processes for a fixed number of times. Index processing is normally used with a Sampling node or with user's code for a sample selection.
Applications Nodes

The **Incremental Response** node directly models the incremental impact of a treatment (such as a marketing action or incentive) and optimizes customer targeting in order to obtain the maximal response or return on investment. You can use incremental response modeling to determine the likelihood that a customer purchases a product, uses a coupon, or to predict the incremental revenue that is realized during a promotional period.

The **Ratemaking** node builds a generalized linear model, which is an extension of a traditional linear model. Using class and binned interval input variables, these models enable the population mean to depend on a linear predictor through a nonlinear link function.

The **Survival** node performs survival analysis on mining customer databases when there are time-dependent outcomes. Some examples of time-dependent outcomes are customer churn, cancellation of all products and services, unprofitable behavior, and server downgrade or extreme inactivity.

Some General Usage Rules for Nodes

These are some general rules that govern how you place nodes in a process flow diagram.

- The Input Data node cannot be preceded by a node that exports a data set.
- The Sample node must be preceded by a node that exports a data set.
- The Assessment node must be preceded by one or more model nodes.
- The Score node and the Score Code Export node must be preceded by a node that produces score code. Any node that modifies the data or builds models generates score code.
- The SAS Code node can be used in any stage of a process flow diagram. It does not require an input data set.

Accessing SAS Data through SAS Libraries

SAS uses libraries to organize files. These libraries point to folders where data and programs are stored. To view existing libraries, select **View → Explorer** on the main menu.
You can view the files in a library by selecting the library name from the list of libraries in the left panel of the Explorer window. To create a new library, select File ⇒ New ⇒ Library on the main menu.

In the Library Wizard — Selection Action window, Create New Library is automatically selected. Click Next. In the Library Wizard — Create or Modify window, enter the Name, Path, and Options for your library. In the Library Wizard — Confirm Action window, a summary is displayed. Click Finish.

SAS Enterprise Miner automatically assigns several libraries when you start it. One of those libraries is SAMPSIO and contains sample data sets. Any data set in the library can be referenced by the two-level name that is constructed by using the SAS library name and the SAS data set name. For example, the HMEQ data set in the SAMPSIO library is identified by the two-level name SAMPSIO.HMEQ.
# Chapter 2

## Predictive Modeling

### Problem Formulation
- Overview of the Predictive Modeling Case ........................................... 20
- Input Data Source .................................................................................. 20

### Creating a Process Flow Diagram
- Identifying the Input Data ................................................................. 21
- Adding the Nodes ................................................................................. 22
- Using the Cursor .................................................................................. 23
- Understanding the Metadata ............................................................... 24
- Inspecting Distribution ....................................................................... 24
- Investigating Descriptive Statistics ..................................................... 25
- Inspecting the Default Settings of the Data Partition Node ................. 26
- Fitting and Evaluating a Regression Model with Data Replacement .... 26
- Understanding Data Imputation .......................................................... 28
- Fitting and Evaluating a Regression Model with Data Imputation ....... 30

### Data Preparation and Investigation
- Preliminary Investigation ..................................................................... 33
- Performing Variable Transformation .................................................. 35
- Understanding Interactive Binning .................................................... 41
- Performing Interactive Binning .......................................................... 42

### Fitting and Comparing Candidate Models
- Fitting a Regression Model ................................................................. 45
- Evaluating the Models ........................................................................ 47
- Fitting a Default Decision Tree .......................................................... 49
- Exploring the Tree Model Results ....................................................... 50
- Fitting a Default Neural Network ....................................................... 51
- Investigating the Regression and Neural Network Models .................. 53

### Generating and Using Scoring Code
- Overview ............................................................................................. 54
- Scoring Using Base SAS ...................................................................... 55
- Scoring with SAS Enterprise Miner ..................................................... 56

### Generating a Report Using the Reporter Node
- ............................................................................................................ 57
Problem Formulation

Overview of the Predictive Modeling Case

A financial services company offers a home equity line of credit to its clients. The company has extended several thousand lines of credit in the past, and many of these accepted applicants (approximately 20%) have defaulted on their loans. By using geographic, demographic, and financial variables, the company wants to build a model to predict whether an applicant will default.

Input Data Source

After analyzing the data, the company selected a subset of 12 predictor (or input) variables to model whether each applicant defaulted. The response (or target) variable BAD indicates whether an applicant defaulted on the home equity line of credit. These variables, along with their model role, measurement level, and description are shown in the following table.

Note: This book uses uppercase for variable names. SAS accepts mixed case and lowercase variable names as well.

<table>
<thead>
<tr>
<th>Name</th>
<th>Model Role</th>
<th>Measurement Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAD</td>
<td>Target</td>
<td>Binary</td>
<td>A value of 1 indicates that the client defaulted on the loan or is seriously delinquent. A value of 0 indicates that the client paid off the loan.</td>
</tr>
<tr>
<td>CLAGE</td>
<td>Input</td>
<td>Interval</td>
<td>Age of the oldest credit line, measured in months</td>
</tr>
<tr>
<td>CLNO</td>
<td>Input</td>
<td>Interval</td>
<td>Number of credit lines</td>
</tr>
<tr>
<td>DEBTINC</td>
<td>Input</td>
<td>Interval</td>
<td>Debt-to-income ratio</td>
</tr>
<tr>
<td>DELINQ</td>
<td>Input</td>
<td>Interval</td>
<td>Number of delinquent credit lines</td>
</tr>
<tr>
<td>DEROG</td>
<td>Input</td>
<td>Interval</td>
<td>Number of major derogatory reports</td>
</tr>
<tr>
<td>JOB</td>
<td>Input</td>
<td>Nominal</td>
<td>Occupational categories</td>
</tr>
<tr>
<td>Name</td>
<td>Model Role</td>
<td>Measurement Level</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LOAN</td>
<td>Input</td>
<td>Interval</td>
<td>Amount requested for the loan</td>
</tr>
<tr>
<td>MORTDUE</td>
<td>Input</td>
<td>Interval</td>
<td>Amount due on the existing mortgage</td>
</tr>
<tr>
<td>NINQ</td>
<td>Input</td>
<td>Interval</td>
<td>Number of recent credit inquiries</td>
</tr>
<tr>
<td>REASON</td>
<td>Input</td>
<td>Binary</td>
<td>The value DebtCon indicates that the loan was intended for debt consolidation. The value HomeImp indicates that the loan was for home improvement.</td>
</tr>
<tr>
<td>VALUE</td>
<td>Input</td>
<td>Interval</td>
<td>Value of the current property</td>
</tr>
<tr>
<td>YOJ</td>
<td>Input</td>
<td>Interval</td>
<td>Years at the applicant’s current job</td>
</tr>
</tbody>
</table>

The SAMPSIO.HMEQ data set contains 5,960 observations for building and comparing competing models. The data set is split into training, validation, and test data sets for analysis.

---

**Creating a Process Flow Diagram**

**Identifying the Input Data**

To begin analyzing HMEQ, you must first add the data source to your project. In the Project Panel, right-click Data Sources and click Create Data Source. This opens the Data Source Wizard. Follow the steps below to complete the Data Source Wizard.

1. In the Data Source Wizard — Metadata Source window, SAS Table is automatically selected for the Source field. Click Next.
2. In the Data Source Wizard — Select a SAS Table window, enter SAMPSIO.HMEQ in the Table field. Click Next.
3. The Data Source Wizard — Table Information window displays some summary information about the HMEQ data set. Click Next.
4. In the Data Source Wizard — Metadata Advisor Options window, the Basic advisor is automatically selected. Click Next.
5. In the Data Source Wizard — Column Metadata window, make the following changes to the Role and Level for each variable listed.
   - BAD — Set the Role to Target. Set the Level to Binary.
• DELINQ — Set the Level to Ordinal
• REASON — Set the Level to Binary.

Click Next.

6. In the Data Source Wizard — Decision Configuration window, No is automatically selected. This indicates that SAS Enterprise Miner will not conduct decision processing for this data source. Click Next.

7. In the Data Source Wizard — Create Sample window, No is automatically selected. This indicates that a sample is not created. Instead, the entire data set is used for analysis. Click Next.

8. In the Data Source Wizard — Data Source Attributes window, you can rename the data source, change its role, segment the data source, or add notes to the data source. Set the Role of the data source to Train. Click Next.

9. The Data Source Wizard — Summary window provides a summary of the data source. To create the data source, click Finish.

Adding the Nodes

Begin building the first process flow diagram to analyze this data. Use the SAS Enterprise Miner toolbar to access the nodes that are used in your process flow diagram.

To add the input data, drag the HMEQ data set from the Data Sources section of the Project Panel to the diagram workspace. Because this is a predictive modeling process flow diagram, you will add a Data Partition node. From the Sample tab of the toolbar, drag a Data Partition node to the diagram workspace. You can also add a node by right-clicking in the diagram workspace and selecting Add Node. The nodes in the Add Node menu are grouped in the same categories as the tabs in the toolbar.

Click the Data Partition node to select it. Your process flow diagram should resemble the above image. Note the border around the Data Partition node, which indicates that the node is selected. To deselect every node, click any open space in the project workspace.
Using the Cursor

The cursor shape changes depending on where it is positioned. The behavior of the mouse commands depends on the cursor shape in addition to the selection state of the node over which the cursor is positioned. Right-click in an open area to see the pop-up menu shown below.

```
Add Node
Paste
Select All
Select Nodes
Layout
Zoom
Copy Diagram to Clipboard
Reset Diagram
```

When you position the mouse over a node, the cursor becomes a set of intersecting lines. This indicates that you are able to click and drag the node in order to reposition it in the diagram workspace. Note that the node will remain selected after you have placed it in a new location.

The default diagram layout positions the nodes horizontally. This means that the nodes in the process flow diagram run from left to right. You can change the diagram layout to position the nodes vertically by right-clicking the diagram workspace and selecting **Layout ➞ Vertically**. This selection causes the nodes in the process flow diagram to run from top to bottom.

If you position the mouse over the right edge of a node, the cursor will turn into a pencil. This indicates that you can create a connection from this node (the beginning node) to another node (the ending node). To create a connection, position the mouse over the right edge of a beginning node, and then left-click and drag the mouse to the ending node. After the mouse is positioned above the ending node, release the left-click.

Connect your **HMEQ** node to the **Data Partition** node.
Understanding the Metadata

All analysis packages must determine how to use variables in the analysis. SAS Enterprise Miner uses metadata in order to make a preliminary assessment of how to use each variable. SAS Enterprise Miner computes simple descriptive statistics that you can access in the Variables window.

To view the variables and their metadata, right-click the HMEQ node and select **Edit Variables**. In the Variables window, you can evaluate and update the assignments that were made when the data source was added to your project. The image below shows only the default metadata information. Use the **Columns** check boxes to view more metadata information for each variable.

Notice that the column **Name** has a gray background, which indicates that you cannot edit the values in this column. Variable names must conform to the same naming standards that were described earlier for libraries.

The first variable listed is BAD. Although BAD is a numeric variable, it is assigned the **Level** of **Binary** because it has only two distinct nonmissing values.

The variables LOAN, MORTDUE, and VALUE are assigned the **Level** of **Interval** because they are numeric variables with more than 10 distinct values.

The variables REASON and JOB are both character variables. However, you set their **Level** to different values. Because REASON has only two distinct, nonmissing values, it is best modeled as a **Binary** variable. JOB is best modeled as a **Nominal** variable because it is a character variable with more than two distinct, non-missing values.

Two other values for **Level** are available, **Unary** and **Ordinal**. A unary variable is a variable that has only one value. An ordinal variable is a numeric variable where the relative order of each value is important. SAS Enterprise Miner automatically assigns numeric variables with more than 2 and fewer than 10 unique values a **Level** of **Ordinal**. Ordinal variables often contain counts, such as the number of children.

Inspecting Distribution

You can inspect the distribution of values in the input data set for each variable. To view the distribution for the variable BAD, complete the following steps:

1. In the Variables window, select the variable BAD.
2. Click the Explore icon on the bottom right corner of the screen.
3. The Explore window appears. The BAD window displays the distribution of the variable BAD.

![Bar chart showing the distribution of BAD variable]

On the bar chart, you can see that approximately 80% of the observations in BAD have a value of 0 and 20% have a value of 1. This means that approximately 20% of the customers in this sample data set defaulted on their loans.

4. Close the Explore window and the Variables window.

**Investigating Descriptive Statistics**

Use the HMEQ data source to calculate and explore descriptive statistics for the input data. To do so, complete the following steps:

1. In your project workspace, select the HMEQ data source.
2. Set the value of the Summarize property to Yes.
3. Right-click the HMEQ data source and click Run. In the Confirmation window, click Yes. After the node has successfully run, click Results in the Run Status window.
4. The Results window appears. The Statistics Table table window contains the descriptive statistics.
Investigate the minimum value, maximum value, mean, standard deviation, percentage of missing values, number of levels, skewness, and kurtosis of the variables in the input data. In this example, there does not appear to be any unusual minimum or maximum values. Notice that, at approximately 22%, DEBTINC has a high percentage of missing values.

5. Close the Results window.

**Inspecting the Default Settings of the Data Partition Node**

In your diagram workspace, select the Data Partition node. The default Partitioning Method for the Data Partition node is stratified because BAD is a class variable. The available methods for partitioning are as follows:

- Simple Random — Every observation in the data set has the sample probability to be selected for a specific partition.
- Stratified — When you use stratified partitioning, specific variables form strata (or subgroups) of the total population. Within each stratum, all observations have an equal probability of being assigned to one of the partitioned data sets.
- Cluster — When you use cluster partitioning, distinct values of the cluster variable are assigned to the various partitions using simple random partitioning. Note that with this method, the partition percentages apply to the values of the cluster variable, and not to the number of observations that are in the partition data sets.

Use the Random Seed property to specify the seed value for random numbers that generated to create the partitions. If you use the same data set and the same Random Seed, excluding the seed value 0, in different process flow diagrams, your partitioning results will be identical. Note that re-sorting the data results in a different ordering of the data, and there a different partitioning of the data. This can lead to slightly different modeling results.

The Data Set Allocations property subgroup enables you to specify what percentage of the input data set is included in each partition. These values must sum to 100.

<table>
<thead>
<tr>
<th>Data Set Allocations</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>40.0</td>
</tr>
<tr>
<td>Validation</td>
<td>30.0</td>
</tr>
<tr>
<td>Test</td>
<td>30.0</td>
</tr>
</tbody>
</table>

For this example, use the default settings.

**Fitting and Evaluating a Regression Model with Data Replacement**

Now that you have decided how the input data is partitioned, you are ready to add a modeling node to the process flow diagram. On the Model tab, drag a Regression node to the diagram workspace. Connect the Data Partition node to the Regression node.

Modeling nodes, such as the Regression node, require a target variable. The Regression node fits models for interval, ordinal, nominal, and binary target variables. Because the target variable BAD is a binary variable, the default model is a binary logistic regression.
model with main effects for each input variable. By default, the node codes your grouping variables with deviation coding. You can specify GLM coding if desired.

In your process flow diagram, right-click the Regression node and click Run. In the Confirmation window, click Yes. After the node has successfully run, click Results in the Run Status window.

The Effects Plot window contains a bar chart of the absolute value of the model effects. The greater the absolute value, the more important that variable is to the regression model. In this example, the most important variables, and thus best predictor variables, are DELINQ, JOB, NINQ, and DEROG.

The Score Rankings Overlay window enables you to view assessment charts. The default chart is the Cumulative Lift chart. Open the drop-down menu in the upper left corner of the Score Rankings Overlay window and click Cumulative % Response. This chart arranges the observations into deciles based on their predicted probability of response. It then plots the actual percentage of respondents.
In this example, the individuals are sorted in descending order of their predicted probability of defaulting on a loan. The plotted values are the cumulative actual probabilities of loan defaults. The Score Rankings Overlay window displays information for both the training and the validation data sets. If the model is useful, the proportion of individuals that defaulted on a load is relatively high in the top deciles and the plotted curve is decreasing. Because this curve steadily increases throughout the middle deciles, the default regression model is not useful.

Recall that the variable DEBTINC has a high percentage of missing values. Because of this, it is not appropriate to apply a default regression model directly to the training data. Regression models ignore observations that have a missing value for at least one input variable. Therefore, you should consider variable imputation before fitting a regression model. Use the Impute node to impute missing values in the input data set.

Understanding Data Imputation

The Impute node enables you to impute missing values in the input data set. Imputation is necessary to ensure that every observation in the training data is used when you build a regression or neural network model. Tree models are able to handle missing values directly. But regression and neural network models ignore incomplete observations. It is more appropriate to compare models that are built on the same set of observations. You should perform variable replacement or imputation before fitting any regression or neural network model that you want to compare to a tree model.

The Impute node enables you to specify an imputation method that replaces every missing value with some statistic. By default, interval input variables are replaced by the mean of that variable. Class input variables are replaced by the most frequent value for that variable. This example will use those default values.

On the Modify tab, drag an Impute node to your diagram workspace. Connect the Data Partition node to the Impute node as shown in the diagram below.

The default imputation method for class variables is Count. This means that missing values are assigned the modal value for each variable. If the modal value is missing, then SAS Enterprise Miner uses the second most frequently occurring level.

The complete set of imputation methods for class variables is as follows:

- **Count** — Use the Count setting to replace missing class variable values with the most frequently occurring class variable value.
- **Default Constant Value** — Use the Default Constant setting to replace missing class variable values with the value that you enter in the Default Character Value property.
- **Distribution** — Use the Distribution setting to replace missing class variable values with replacement values that are calculated based on the random percentiles of the variable's distribution. In this case, the assignment of values is based on the probability distribution of the nonmissing observations. The distribution imputation method typically does not change the distribution of the data very much.
- **Tree** — Use the Tree setting to replace missing class variable values with replacement values that are estimated by analyzing each input as a target. The remaining input and rejected variables are used as predictors. Use the Variables window to edit the status of the input variables. Variables that have a model role of target cannot be used to impute the data. Because the imputed value for each input variable is based on the other input variables, this imputation technique might be more accurate than simply using the variable mean or median to replace the missing tree values.

- **Tree Surrogate** — Use the Tree Surrogate setting to replace missing class variable values by using the same algorithm as Tree Imputation, except with the addition of surrogate splitting rules. A surrogate rule is a backup to the main splitting rule. When the main splitting rule relies on an input whose value is missing, the next surrogate is invoked. If missing values prevent the main rule and all the surrogates from applying to an observation, the main rule assigns the observation to the branch that is assigned to receive missing values.

- **None** — Missing class variable values are not imputed under the None setting.

The following imputation methods are available for interval target variables:

- **Mean** — Use the Mean setting to replace missing interval variable values with the arithmetic average, calculated as the sum of all values divided by the number of observations. The mean is the most common measure of a variable's central tendency. It is an unbiased estimate of the population mean. The mean is the preferred statistic to use to replace missing values if the variable values are at least approximately symmetric (for example, a bell-shaped normal distribution). Mean is the default setting for the Default Input Method for interval variables.

- **Median** — Use the Mean setting to replace missing interval variable values with the 50th percentile, which is either the middle value or the arithmetic mean of the two middle values for a set of numbers arranged in ascending order. The mean and median are equal for a symmetric distribution. The median is less sensitive to extreme values than the mean or midrange. Therefore, the median is preferable when you want to impute missing values for variables that have skewed distributions. The median is also useful for ordinal data.

- **Midrange** — Use the Midrange setting to replace missing interval variable values with the maximum value for the variable plus the minimum value for the variable divided by two. The midrange is a rough measure of central tendency that is easy to calculate.

- **Distribution** — Use the Distribution setting to replace missing interval variable values with replacement values that are calculated based on the random percentiles of the variable's distribution. The assignment of values is based on the probability distribution of the nonmissing observations. This imputation method typically does not change the distribution of the data very much.

- **Tree** — Use the Tree setting to replace missing interval variable values with replacement values that are estimated by analyzing each input as a target. The remaining input and rejected variables are used as predictors. Use the Variables window to edit the status of the input variables. Variables that have a model role of target cannot be used to impute the data. Because the imputed value for each input variable is based on the other input variables, this imputation technique might be more accurate than simply using the variable mean or median to replace the missing tree values.

- **Tree Surrogate** — Use the Tree Surrogate setting to replace missing interval variable values by using the same algorithm as Tree Imputation, except with the addition of surrogate splitting rules. A surrogate rule is a backup to the main splitting...
rule. When the main splitting rule relies on an input whose value is missing, the next surrogate is invoked. If missing values prevent the main rule and all the surrogates from applying to an observation, the main rule assigns the observation to the branch that is assigned to receive missing values.

- **Mid-Minimum Spacing** — Use the Mid-Minimum setting to replace missing interval variable values with mid-minimum spacing. The mid-minimum spacing method uses a numeric constant to specify the proportion of the data to be contained in the spacing.

- **Tukey's Biweight** — Use the Tukey's Biweight setting to replace missing interval variable values with the Tukey's Biweight robust M-estimator value.

- **Huber** — Use the Huber setting to replace missing interval variable values with the Huber's robust M-estimator value.

- **Andrew's Wave** — Use the Andrew's Wave setting to replace missing interval variable values with the Andrew's Wave robust M-estimator value.

- **Default Constant** — Use the Default Constant Value setting to replace missing interval variable values with the value that you enter in the Default Character Value property.

- **None** — Specify the None setting if you do not want to replace missing interval variable values.

Select the Impute node and find the Indicator Variables property subgroup in the properties panel. Set the value of Type to Single to create a single indicator variable that indicates whether any variable was imputed for each observation. Set the value of Type to Unique to create an indicator variable for each original variable that identifies if that specific variable was imputed.

For this example, set the value of Type to None.

An indicator variable is a special variable that is created automatically by SAS Enterprise Miner. The value of this variable is 1 when an observation contained a missing value and 0 otherwise. The Regression and Neural Network nodes can use the indicator variables to identify observations that had missing values before the imputation.

### Fitting and Evaluating a Regression Model with Data Imputation

Next, you will build a new regression model based on the imputed data set. From the Model tab, drag a Regression node to your diagram workspace. Connect the Impute node to the new Regression node.

Right-click the Regression (2) node and click Run. In the Confirmation window, click Yes. After the node has successfully run, click Results in the Run Status window.
In the Score Rankings Overlay window, click **Cumulative % Response** on the drop-down menu in the upper left corner. Notice that this model shows a smooth decrease in the cumulative percent response, which contrasts with the previous model. This indicates that the new model is much better than the first model at predicting who will default on a loan.

The discussion of the remaining charts refers to those who defaulted on a loan as defaults or respondents. This is because the target level of interest is BAD=1.

Position the mouse over a point on the **Cumulative % Response** plot to see the cumulative percent response for that point on the curve. Notice that at the first decile (the top 10% of the data), approximately 69% of the loan recipients default on their loan.

On the drop-down menu in the upper left corner of the Score Rankings Overlay window, click **% Response**. This chart shows the non-cumulative percentage of loan recipients that defaulted at each level of the input data.

On the drop-down menu, click **Cumulative Lift**.
Lift charts plot the same information about a different scale. As discussed earlier, the overall response rate is approximately 20%. You calculate lift by dividing the response rate in a given group by the overall response rate. The percentage of respondents in the first decile was approximately 69%, so the lift for that decile is approximately 69/20 = 3.45. Position the cursor over the cumulative lift chart at the first decile to see that the calculated lift for that point is 3.4. This indicates that the response rate in the first decile is more than three times greater than the response rate in the population.

Instead of asking the question, “What percentage of those in a bin were defaulters?”, you could ask the question, “What percentage of the total number of defaulters are in a bin?” The latter question can be evaluated by using the Cumulative % Captured Response curve. Click Cumulative % Captured Response on the drop-down menu.

You can calculate lift from this chart as well. If you were to take a random sample of 10% of the observations, you would expect to capture 10% of the defaulters. Likewise, if you take a random sample of 20% of the data, you would expect to capture 20% of the defaulters. To calculate lift, divide the proportion of the defaulters that were captured by the percentage of those whom you have chosen for action (rejection of the loan application).

Note that at the 20th percentile, approximately 55% of those who defaulted are identified. At the 30th percentile, approximately 68% of those who defaulted are
identified. The corresponding lift values for those two percentiles are approximately 2.75 and 2.27, respectively. Observe that lift depends on the proportion of those who have been chosen for action. Lift generally decreases as you choose larger proportions of the data for action. When you compare two models on the same proportion of the data, the model that has the higher lift is often preferred, excluding issues that involve model complexity and interpretability.

Note: A model that performs best in one decile might perform poorly in other deciles. Therefore, when you compare competing models, your choice of the final model might depend on the proportion of individuals that you have chosen for action.

As with the initial Regression node results, you can view the Effects Plot for this model. Note that in this model, the most important variables are DELINQ, JOB, DEROG, NINQ, and REASON.

Data Preparation and Investigation

Preliminary Investigation

In your process flow diagram, select the Data Partition node. You can use the Variables property to access several summary statistics and statistical graphs for the input data set. Click the ... button next to the Variable property of the Data Partition node. Select all 13 the variables and click Explore.

TIP To select all of the variables, you can select one variable and press Ctrl+A on your keyboard.

The Explore window appears. This window contains the sample statistics for every variable, a histogram for each of the interval variables, and a bar chart for each class variable. This example highlights only a few of these plots, but you are encouraged to explore the rest on your own.

Maximize the Sample Properties window. This window contains information about the data set sample that was used to create the statistics and graphics in the Explore window. The Fetched Rows property indicates the number of observations that were used in the
sample. The SAMPSIO.HMEQ data set is small enough that the entire data set is used. Close the Sample Properties window.

Maximize the Sample Statistics window. This window contains the calculated mean, maximum, and minimum for the interval variables and the number of class levels, modal value, and percentage of observations in the modal value for class variables. The percentage of missing values is calculated for every variable. Close the Sample Statistics window.

Maximize the CLAGE window. The variable CLAGE indicates the age of the client’s oldest credit line in months.

The gray bar on the left side of the histogram represents the missing values. Notice that the vast majority of the observations are less than 350. The CLAGE data set is skewed right. Close the CLAGE window.

Maximize the REASON window. This variable represents the stated reason why the client took out the loan.
The unlabeled bar in this bar chart indicates the missing values. Notice that significantly more people listed debt consolidation as the reason that they borrowed instead of home improvement. Close the REASON window.

It is highly encouraged that you view the distributions for each of the remaining variables. When you are finished, close the Explore window. Next, close the Variables window.

**Performing Variable Transformation**

After you have viewed the sample statistics and variable distributions, it is obvious that some variables have highly skewed distributions. In highly skewed distributions, a small percentage of the data points can have a large amount of influence on the final model. Sometimes, performing a transformation on an input variable can yield a better fitting model. The Transform Variables node enables you to perform variable transformation.

From the Modify tab, drag a Transform Variables node to your diagram workspace.

Connect the Data Partition node to the Transform Variables node. Click next to the Variables property of the Transform Variables node. The Variables window appears.

In the Variables window, select the Statistics option in the upper right corner of the screen. Scroll the variables list all the way to the right. You should see the Skewness and Kurtosis statistics.
The Skewness statistic indicates the level of skewness and the direction of skewness for a distribution. A Skewness value of 0 indicates that the distribution is perfectly symmetrical. A positive Skewness value indicates that the distribution is skewed to the right, which describes all of the variables in this data set. A negative value indicates that the distribution is skewed to the left.

The Kurtosis statistic indicates the peakedness of a distribution. However, this example focuses only on the Skewness statistic.

The Transform Variables node enables you to rapidly transform interval variables using standard transformations. You can also create new variables whose values are calculated from existing variables in the data set. Note that the most skewed variables are, in order, DEROG, DELINQ, VALUE, DEBTINC, NINQ, and LOAN. These five variables all have a Skewness value greater than 2. Close the Variables window.

This example applies a log transformation to all of the input variables. The log transformation creates a new variable by taking the natural log of each original input variable. In your diagram workspace, select the Transform Variables node. Set the value of the Interval Inputs property to Log.

Right-click the Transform Variables property and click Run. Click Yes in the Confirmation window. In the Run Status window, click Results.

Maximize the Transformations Statistics window. This window provides statistics for the original and transformed variables. The Formula column indicates the expression used to transform each variable. Notice that the absolute value of the Skewness statistic for the transformed values is typically smaller than that of the original variables. Close the Results window.

The Transform Variables enables you to perform a different transformation on each variable. This is useful when your input data contains variables that are skewed in different ways. In your diagram workspace, select the Transform Variables node. Click next to the Variables property of the Transform Variables node. The Variables window appears. In the Variables window, note the Method column. Use this column to set the transformation for each variable individually.

Before doing so, you want to recall the distribution for each variable. Select the variable DEROG and click Explore. Note that nearly all of the observations have a value of 0. Close the Explore window.
Repeat this process for the **DELINQ** variable. Nearly all of the values for **DELINQ** are equal to 0. The next largest class is the missing values.

In situations where there is a large number of observations at one value and relatively few observations spread out over the rest of the distribution, it can be useful to group the levels of an interval variable. Close the Variables window.

Instead of fitting a slope to the whole range of values for **DEROG** and **DELINQ**, you need to estimate the mean in each group. Because most of the applicants in the data set had no delinquent credit lines, there is a high concentration of observations where **DELINQ**=0.

In your process flow diagram, select the **Transform Variables** node. Click \[\text{Property} \] next to the **Formulas** property. The Formulas window appears.
The Formulas window enables you to create custom variable transformations. Select the **DELINQ** variable and click the **Create** variable in the upper left corner of the Formulas window. The Add Transformation window appears.
Complete the following steps to transform the DELINQ variable:

1. Enter INDELINQ for the Name property. The default values are acceptable for the other properties.

2. In the Formula dialog box, enter \( \text{DELINQ} > 0 \).

   This definition is an example of Boolean logic and illustrates one way to dichotomize an interval variable. The statement is either true or false for each observation. When the statement is true, the expression evaluates as 1. Otherwise, the expression evaluates as 0. In other words, when DELINQ>0, INDELINQ=1. Likewise, when DELINQ=0, INDELINQ=0. If the value of DELINQ is missing, the expression evaluates to 0, because missing values are treated as being smaller than any nonmissing values in a numerical comparison. Because a missing value of DELINQ is reasonably imputed as DELINQ=0, this does not pose a problem for this example.

3. Click OK. The formula now appears in the Formulas window.

4. Repeat the above steps for the variable DEROG. Name the new variable INDEROG.
5. Click **Preview** in the lower left corner of the screen.
6. Click **OK**.

Even though DEROG and DELINQ were used to construct the new variables, the original variables are still available for analysis. You can modify this if you want, but this example keeps the original variables. This is done because the transformed variables contain only a portion of the information that is contained in the original variables. Specifically, the new variables identify whether DEROG or DELINQ is greater than zero.

**Understanding Interactive Binning**

An additional processing technique to apply before modeling is binning, also referred to as grouping. The Interactive Binning node enables you to automatically group variable values into classes based on the node settings. You have the option to modify the initially generated classes interactively. By using the Interactive Grouping node, you can manage the number of groups of a variable, improve the predictive power of a variable, select predictive variables, generate the Weight of Evidence (WOE) for each group of a variable, and make the WOE vary smoothly (or linearly) across the groups.

The WOE for a group is defined as the logarithm of the ratio of the proportion of nonevent observations in the group over the proportion of event observations in the group. For the binary target variable BAD in this example, BAD=1 (a client who
defaulted) is the event level and BAD=0 (a client who repaid the loan) is the nonevent level. WOE measures the relative risk of a group. Therefore, high negative values of WOE correspond to high risk of loan default. High positive values correspond to low risk.

After the binning of variable values has been defined for a variable, you can assess the predictive power of the variable. The predictive power of a variable is the ability of the variable to distinguish event and nonevent observations. In this example, it is the ability to separate bad loan clients from good loan clients. You can assess the predictive power by using one of the following criteria:

- Information Value — is the weighted sum of WOE over the groups. The weight is the difference between the proportion of nonevents and the proportion of events in each group.
- Gini Score — is the same as the Gini index in the Tree node. See the Interactive Grouping node in the SAS Enterprise Miner Help for more information.

The WOE variables are usually used as inputs in successor modeling nodes.

**Performing Interactive Binning**

Recall the distribution of NINQ. NINQ is a counting variable, but the majority of the observations have the value of either 0, 1, or 2. It might be useful to create a grouped version of NINQ by pooling all of the values larger than 2 (of which there are very few) into a new level. This would create a new three-level grouping variable from NINQ. Creating a grouping variable that has three levels causes loss of information about the exact number of recent credit inquiries. But it does enable you to handle nonlinearity in the relationship between NINQ and the response variable.

In order to bin NINQ, you need to add an Interactive Binning node to your process flow diagram. However, because you plan to bin NINQ, you do not want to transform it with the Transform Variables node. Right-click the Transform Variables node and click Variables. Set the value of Method for the variable NINQ to None. The Transform Variables node will not transform NINQ.

On the Modify tab, drag an Interactive Binning node to your process flow diagram. Connect the Transform Variables node to Interactive Binning node.

Right-click the Interactive Binning node and click Run. In the Confirmation window, click Yes. In the Run Status window, click OK.

![Diagram of process flow]

In your diagram workspace, select the Interactive Binning node. Next, click next to the Interactive Binning property in the properties panel.
Use the Interactive Binning window to manually bin the input variables. Before binning NINQ, first notice that most of the variables have a Calculated Role of Rejected. Because you want to use these variables in the data mining process, set their New Role to Input. To do so, select all of the variables. Then right-click anywhere in the selection and click Input. This result is shown in the image above.

Next, on the Selected Variable drop-down menu, located in the upper left corner of the screen, click NINQ. Next, click the Groupings tab to see details about the variable NINQ.

Automatically, the Interactive Binning node created five groups for NINQ. Notice, however, that the first two groups contain the missing values and negative values, respectively. Also notice that the group that contains the negative values is empty. For
In this example, you assume that a missing value indicates that no credit inquiry was made. Therefore, you want to merge both of these groups into the third group.

First, select the **MISSING** group and right-click that row. Click **Assign To**. In the Group Selection window, select 3 for the **Select a Group** selection.

![Group Selection Window](image)

Notice that the groups were renumbered.

Next, repeat this process for the empty bin that contains the negative values. Select the group **NINQ < 0** and right-click that row. Click **Group = 2**. Once again, this renumbers the groups.

You now have three groups for the variable NINQ. But you want to add a fourth group that contains all of the values greater than two and retain a group that contains just 2. To do so, select the row **NINQ >= 2** and right-click that row. Click **Split Bin**. Enter 2.5 in the **Enter New Cutoff Value** dialog box of Split Bin window. This creates a separate bin that still belongs to the same group. To assign this new bin to its own group, right-click the row **2.5 <= NINQ** and click **Group = 4**.
Note: Because the values of NINQ are only whole numbers, you could have selected any value between 2 and 3, exclusive as the new cutoff value.

On the Select Variable drop-down menu, click INDEROG. Notice that the Interactive Grouping node decided to split INDEROG at 0, which means that every observation now belongs to the same bin. Select the row INDEROG >= 0, right-click that row and click Split Bin. Enter 0.5 in the Enter New Cutoff Value dialog box. Click OK. Now right-click the row 0.5 <= INDEROG and click GROUP = 4.

You need to repeat this same process for the variable INDELINQ. On the Select Variable drop-down menu, click INDELINQ. Right-click the row INDELINQ >= 0, and click Split Bin. Enter 0.5 in the Enter New Cutoff Value dialog box. Click OK. Now right-click the row 0.5 <= INDELINQ and click GROUP = 4.

For both INDELINQ and INDEROG, group 3 corresponds to an original value of 0 and group 4 corresponds to an original value of 1.

Click Close to close the Interactive Binning window. Click Yes in the Save Changes window.

---

**Fitting and Comparing Candidate Models**

**Fitting a Regression Model**

On the Model tab, drag a Regression node to your diagram workspace. Connect the Interactive Binning node to the Regression (3) node.

In the Properties Panel, locate the Model Selection subgroup. These properties determine whether and which type of model of variable selection is performed when generating the model. You can choose from the following variable selection techniques:

- **Backward** — Variable selection begins with all candidate effects in the model and systematically removes effects that are not significantly associated with the target. This continues until no other effect in the model meets the Stay Significance Level,
or until the **Stop Variable Number** is met. This method is not recommended for binary or ordinal targets when there are many candidate effects or when there are many levels for some classification input variables.

- **Forward** — Variable selection begins with no candidate effects in the model and then systematically adds effects that are significantly associated with the target. This proceeds until none of the remaining effects meet the **Entry Significance Level**, or until the **Stop Variable Number** is met.

- **Stepwise** — Variable selection begins with no candidate effects in the model and then systematically adds effects that are significantly associated with the target. After an effect is added to the model, it can be removed if it is deemed that the effect is no longer significantly associated with the target.

- **None** — No model selection is performed and all candidate effects are included in the final model.

For this example, set the value of the **Selection Model** property to **Forward**. Set the value of the **Use Selection Defaults** property to **No**. After you do this, notice that the **Selection Options** property becomes available.

Click ![ ] next to the **Selection Options** property. The **Selection Options** window contains the **Entry Significance Level**, **Stay Significance Level**, **Start Variable Number**, and **Stop Variable Number** properties that were just mentioned. These properties enable you to control the significance level required for an effect to remain in or be removed from a model and the number of effects in a model. For **Backward** selection, the **Stop Variable Number** determines the minimum number of effects. For **Forward** selection, the **Stop Variable Number** determines the maximum number of effects.

The **Maximum Number of Steps** property enables you to set the maximum number of steps before the **Stepwise** method stops. The default value is twice the number of effects in the model.

The **Hierarchy Effects** property determines which variables are subject to effect hierarchies. Specify **Class** if you want to include just the classification variables and **All** if you want to use both the classification and interval variables.

Model hierarchy refers to the requirement that for any effect in the model, all effects that it contains must also be in the model. For example, in order for the interaction A*B to be in the model, the main effects A and B must also be in the model.

For this example, set the value of the **Stay Significance Level** to 0.025. Set the value of the **Start Variable Number** to 10. This ensures that at least 10 effects will be used by the regression model. Use the default settings for every other property.
Evaluating the Models

On the Assess tab, drag a Model Comparison node to your diagram workspace. Connect the Regression, Regression (2), and Regression (3) nodes to the Model Comparison node.

Right-click the Model Comparison node and click Run. In the Confirmation window click Yes. In the Run Status window click Results.
The Model Comparison node Results window displays a lift chart that contains all three models. Maximize the Score Rankings Overlay window. On the drop-down menu, click Cumulative % Response.

Recall that this chart groups individuals based on the predicted probability of response, and then plots the percentage of respondents. Notice that the model created by the Regression (3) node is better at almost every depth.
The model created in the Regression (2) node was created with no effect selection, because it used the default settings of the Regression node. Therefore, that model contains every effect.

Minimize the Score Rankings Overlay window.

Maximize the Output window. Scroll down until you find the Fit Statistics Table. This table lists all of the calculated assessment statistics for each model that was compared. Statistics are computed separately for each partition of the input data. These statistics are also displayed in the Fit Statistics window.

Explore the rest of the charts and tables in the Results window. Close the Results window when you are finished.

**Fitting a Default Decision Tree**

Now that you have modeled the input data with three different regression models, you want to fit a tree model to determine whether it will make better predictions. On the Model tab, drag a Decision Tree node to your diagram workspace. Connect the Data Partition node to the Decision Tree node. Connect the Decision Tree node to the Model Comparison node.

Tree models handle missing values differently than regression models, so you can directly connect the Data Partition node to the Decision Tree node.

Monotonic transformations of interval numeric variables are not likely to improve the tree model because the tree groups numeric variables into bins. In fact, the tree model might perform worse if you connect it after you group the input variables. The bins created in the Interactive Binning node reduce the number of splits that the tree model can consider, unless you include both the original and binned variables.

Right-click the Model Comparison node and click Run. In the Confirmation window, click Yes. In the Run Status window, click Results.
In the Score Rankings Overlay window, notice that Cumulative % Response graphs for the Decision Tree model and the Regression (3) are fairly similar. However, the Regression (3) model is better as the depth increases.

The Fit Statistics window confirms that the Regression (3) model is the better model based on the selection criteria of the Model Comparison node.

Explore the rest of the charts and tables in the Results window. Close the Results window when you are finished.

**Exploring the Tree Model Results**

In your diagram workspace, right-click the Decision Tree node and click Results.
The Tree Map and Tree windows present two different views of the decision tree. The Tree window displays a standard decision tree, where the node splits are provided on the links between the nodes. Inside the nodes, you can find the node ID, the training and validation classification rates, and the number of observations in the node for each partition. The Tree Map window uses the width of each node to visually display the percentage of observations in each node on that row. Selecting a node in one window selects the corresponding node in the other window.

The color of each node in the decision tree is based on the target variable. The more observations where BAD=1, the darker the node color is.

The Leaf Statistics window displays the training and validation classification rates for each leaf node in the tree model. Leaf nodes are terminal nodes in the decision tree. Select a bar in the Leaf Statistics window to select the corresponding leaf node in the Tree and Tree Map windows.

The Fit Statistics window displays several computed statistics for the tree model. These are computed for all partitions of the training data.

Explore the rest of the charts and tables in the Results window. Close the Results window when you are finished.

**Fitting a Default Neural Network**

Next, you want to compare how well a neural network model compares to the regression and tree models. On the Model tab, drag a Neural Network node to your diagram workspace. Connect the Interactive Binning node to the Neural Network node. Connect the Neural Network node to the Model Comparison node.
By default, the **Neural Network** node creates a multilayer perceptron (MLP) model with no direct connections, and the number of hidden layers is data-dependent.

Right-click the **Neural Network** node and click **Run**. In the Confirmation window, click **Yes**. Click **Results** in the Run Status window.

The **Fit Statistics** window displays various computed statistics for the neural network model. The **Iteration Plot** window displays various statistics that were computed at each iteration during the creation of the neural network node.

Explore the rest of the charts and tables in the **Results** window. Close the **Results** window when you are finished.

In the diagram workspace, select the **Model Comparison** node. Right-click the **Model Comparison** node and click **Run**. In the Confirmation window, click **Yes**. In the Run Status window, click **Results**.

Notice that the predictive power **Neural Network** model is very similar to the **Decision Tree** and **Regression (3)** models. However, both models are considered slightly better by the **Model Comparison** node based on the default assessment criteria. In the **Fit Statistics** window, the models are listed in order with best model being at the top of the table and the worst model at the bottom.
Explore the rest of the charts and tables in the Results window. Close the Results window when you are finished.

Investigating the Regression and Neural Network Models

The default neural network model does not perform any better than the regression model. If both models performed poorly compared to the decision tree model, poor performance might be due to how missing values are handled. The tree model directly handles observations with missing values while the regression and neural network models ignore those observations. This is why the Regression model is significantly worse than the Decision Tree model.

In the Neural Network and Regression (3) model, you transformed and binned the variables before creating the regression. When the variables were binned, classification variables were created and missing values were assigned to a level for each of the new classification variables.

The Regression (2) model uses imputation to handle missing values. The effect of this replacement is that you replace a missing value (perhaps an unusual value for the variable) with an imputed value (a typical value for the variable). Imputation can change an observation from being somewhat unusual with respect to a particular variable to very typical with respect to that variable.

For example, if someone applied for a loan and had a missing value for INCOME, the Impute node would replace that value with the mean value of INCOME by default. In practice, however, someone who has an average value for INCOME would often be evaluated differently than someone with a missing value for INCOME. Any models that follow the Impute node would not be able to distinguish the difference between these two applicants.

One solution to this problem is to create missing value indicator variables. These variables indicate whether an observation originally had a missing value before imputation is performed. The missing value indicators enable the Regression and Neural Network nodes to differentiate between observations that originally had missing values and observations with no missing values. The addition of missing value indicators can greatly improve a neural network or regression model.

Recall that you chose not to use indicator variables earlier. You are going to reverse that decision now. In your diagram workspace, select the Impute node. Locate the Type property in the Indicator Variables subgroup. Set the value of this property to Unique. Set the value of the Source property to Missing Values. Set the value of the Role property to Input.

In the diagram workspace, select the Model Comparison node. Right-click the Model Comparison node and click Run. In the Confirmation window, click Yes. In the Run Status window, click Results.

Notice that the Regression (2) model now outperforms the Neural Network model. In fact, it is the best model in the first decile of the training data.

Close the Results window.

In general, it is impossible to know which model will provide the best results when it is applied to new data. For this data, or any other, a good analyst considers many variations of each model and identifies the best model according to their own criteria. In this case, assume that Regression (3) is the selected model.
Generating and Using Scoring Code

Overview

After deciding on a model, you often need to use your model to score new or existing observations. The Score node can be used to evaluate, save, and combine scoring code from different models. In this example, you want to score a data set using the Regression (3) model.

On the Assess tab, drag a Score node to your diagram workspace. Connect the Regression (3) node to the Score node.

First, set the value of the Type of Scored Data property to Data. Right-click the Score node and click Run. In the Confirmation window, click Yes. Click Results in the Run Status window.

In the Results window, maximize the Optimized SAS Code window. This is the optimized score code for the Regression (3) model.

You can save the optimized score code by selecting File → Save As on the main menu. In Save window, navigate to an appropriate location for you to save the score code and enter scoreCode.sas in the File name field. This enables you to make changes to a model without losing the score code for an accurate model.
Close the Results window.

**Scoring Using Base SAS**

You can use the saved score code to score a data set by using Base SAS. The program requires only Base SAS to run. Therefore, you can run the program on any of the systems in which you have installed Base SAS, regardless of whether SAS Enterprise Miner is installed.

SAS Enterprise Miner runs on top of a SAS session, and you can use this SAS session at any time. Use this SAS session to score the DMAHMEQ data set in the SAMSIO library. This data set contains all the same inputs as the HMEQ data set, but it also contains response information. This enables you to compare the predicted outcome to the actual outcome.

To score the data set using Base SAS, complete the following steps:

1. From the main menu, select **View ⇒ Program Editor**. This opens the Program Editor — Editor window in place of the diagram workspace.

2. From the main menu, select **File ⇒ Open**. In the Open window, navigate to the location where you saved your score code. Select your code and click **Open**.

3. At the beginning of the code, enter the following lines:

   ```sas
   data myPredicted;
   set sampsio.dmahmeq;
   ```

   These lines create the myPredicted data set and choose SAMSIO.DMAHMEQ as the data set that is scored.

4. At the end of the code, enter the following lines:

   ```sas
   title "The First 10 Observations";
   PROC PRINT DATA=myPredicted(obs=10);
   VAR BAD P_BAD1 P_BAD0;
   run;
   ```
This code prints the value of BAD as well as P_BAD1 and P_BAD0. P_BAD1 is the predicted probability of default, and P_BAD0 is predicted probability of not defaulting.

Note: Although it is not necessary, you should consider sorting the observations by one of the variables, such as P_BAD1. To do so, add the code that follows before the PROC PRINT statement. By sorting the observations by descending values of P_BAD1, you arrange the most likely to default observations at the top of the printed results. The example in this book does sort the data.

```
PROC SORT DATA=myPredicted;
   BY DESCENDING P_BAD1;
run;
```

5. Submit the score code by selecting **Actions ➤ Run** on the main menu.

6. After the code has finished running, click the **Output** tab at the bottom of the Program Editor window. Notice that the 10 observations with the greatest value of P_BAD1 were printed.

<table>
<thead>
<tr>
<th>Obs</th>
<th>bad</th>
<th>P_BAD1</th>
<th>P_BAD0</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1</td>
<td>0.99992</td>
<td>0.000075084</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>0.99984</td>
<td>0.000159540</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>0.99968</td>
<td>0.000319392</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>0.99962</td>
<td>0.000377014</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>0.99913</td>
<td>0.000868449</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>0.99904</td>
<td>0.000956572</td>
</tr>
<tr>
<td>12</td>
<td>7</td>
<td>0.99903</td>
<td>0.000974321</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>0.99899</td>
<td>0.001010456</td>
</tr>
<tr>
<td>14</td>
<td>9</td>
<td>0.99892</td>
<td>0.001082280</td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>0.99892</td>
<td>0.001082321</td>
</tr>
</tbody>
</table>

Observe that because BAD has two levels, and the value of P_BAD1+P_BAD0 is always 1.

Close the Program Editor window to return to your process flow diagram.

**Scoring with SAS Enterprise Miner**

When you attach an input data set to the **Score** node, SAS Enterprise Miner knows to score that data set. However, you have not added the SAMPSIO.DMAHMEQ data set to your project yet. To do so, complete the following steps:

1. In the Project Panel, right-click **Data Sources** and click **Create Data Source**.
2. In the Data Source Wizard, click **Next**.
3. Enter **SAMPSIO.DMAHMEQ** in the **Table** field. Click **Next**.
4. In the Table Information window, click **Next**.
5. In the Metadata Advisor Options window, click **Basic**. Click **Next**.
6. In the Column Metadata window, click **Next**.
7. In the Create Sample window, click Next.
8. In the Data Source Attributes window, set the value of Role to Score. Click Next.
9. In the Summary window, click Finish.

This creates the **All: Home-Equity Loan Scoring Data** data source in your Project Panel. Drag this data source to the diagram workspace. Connect the **All: Home-Equity Loan Scoring Data** data source to the **Score** node.

Set the value of the **Type of Scored Data** property to View. In the **Score Data** properties subgroup, set the value of the **Validation** and **Test** properties to Yes.

Right-click the **Score** node and click Run. In the Confirmation window, click Yes. Click OK in the Run Status window.

On the Utility tab, drag a SAS Code node to your diagram workspace. Connect the **Score** code to the **SAS Code** node. Click the ellipsis button next to the **Code Editor** property of the **SAS Code** node. In the **Training Code** field, enter the following code:

```sas
PROC SORT DATA=<diagramID>.Score_SCORE out=defaults;
   BY DESCENDING P_BAD1;
run;

PROC PRINT DATA=DEFAULTS;
   VAR BAD P_BAD1 P_BAD0;
run;
```

You must replace `<diagramID>` with the value of the **ID** property. This value is displayed when you select the diagram in the Project Panel. Close the Training Code window. Click Yes in the Save Changes window.

Right-click the **SAS Code** node and click Run. In the Confirmation window, click Yes. Click Results in the Run Status window.

The scored values here should match those obtained in the previous section. Close the Results window.

---

**Generating a Report Using the Reporter Node**

To create a PDF report of this example, add a Reporter node. On the Utility tab, drag a Reporter node to your diagram workspace. Connect the **SAS Code** node to the Reporter node.
Right-click the **Reporter** node and click **Run**. In the Confirmation window, click **Yes**. Click **Results** in the Run Status window.

The Custom Report window provides the location where the PDF was saved. Use the **View** icon in the bottom right corner of the Custom Report window to open the report. This PDF contains many of the results and graphs that were discussed throughout the example.
Chapter 3
Variable Selection

Introduction to Variable Selection

Input data often contains an extremely large number of variables. In general, using all of the variables in a predictive model is not practical, so variable selection plays a critical role in modeling. The previous chapter used stepwise regression to perform variable selection. However, this method might not perform well when you are evaluating data sets that contain hundreds of potential input variables. Furthermore, keep in mind that the stepwise selection method is available only in the Regression node. Variable selection is often more critical for the Neural Network node than it is for the other modeling nodes. This is because of the large number of parameters that are generated relative to using the same number of variables in a regression model.

Because variable selection is a critical phase in model building, SAS Enterprise Miner provides the Variable Selection node. Variables selected for analysis in the Variable Selection node are available to any subsequent nodes. No single method of variable selection is universally better than any other method. It is often useful to consider many types of variable selection methods when evaluating the importance of each variable.

This chapter demonstrates how to use the Variable Selection node to identify important variables. For convenience, consider the process flow diagram that you created in the previous chapter. In this chapter, you perform only variable selection before creating a neural network model. This model is then compared to the other models created in the previous chapter.

Using the Variable Selection Node

Selecting Variables Using the R-Square Criterion

On the Explore tab, drag a Variable Selection node to your diagram workspace. Connect the Data Partition node to the Variable Selection node.
Set the value of the **Max Missing Percentage** property to 10. This eliminates variables that have more than 10% of their values missing.

Set the value of the **Target Model** property to **R-Square**. This indicates that the r-square criterion is used to evaluate and select variables. Notice that the **Chi-Square Options** properties subgroup is now unavailable. Use the default values of the properties in the **R-Square Options** subgroup.

The **R-Square** criterion uses a goodness-of-fit criterion to evaluate variables. It uses a stepwise method of selecting variables that stops when the improvement in the r-square value is less than 0.0005. By default, this method rejects variables whose contribution to the r-square value is less than 0.005.

The following three-step process is done when you apply the **R-Square** criterion to a binary target. When the target is non-binary, only the first two steps are performed.

1. SAS Enterprise Miner computes the squared correlation for each variable with the target and then assigns the rejected role to those variables that have a value less than **Minimum R-Square** value.
2. SAS Enterprise Miner evaluates the remaining variables with a forward stepwise r-square regression. Variables that have a stepwise r-square improvement less than the **Stop R-Square** value are rejected.
3. SAS Enterprise Miner performs a logistic regression with the predicted values that are output from the forward stepwise regression used as the independent input variable.

Right-click the **Variable Selection** node and click **Run**. In the Confirmation window, click **Yes**. Click **Results** in the Run Status window.
In the Variable Selection window, notice that CLAGE, DEBTINC, DELINQ, DEROG, G_JOB, NINQ, and YOJ have their **Role** set to **Input**. This indicates that they were the variables selected by the node for inclusion in the preceding neural network model.

Close the Results window.

**Creating and Evaluating a Neural Network Model**

On the **Model** tab, drag a **Neural Network** node to your diagram workspace. Connect the **Variable Selection** node to the **Neural Network (2)** node. Connect the **Neural Network (2)** node to the **Model Comparison** node.

It is highly recommended that you perform some type of variable selection before building neural network models. Neural network models are very flexible, but they are also very computationally intense. Failure to reduce the number of input variables can result in the following:

- an overfit model that does not perform well in practice
- a tremendous increase in the computational time that is necessary to fit a model
- computational difficulties in obtaining good parameter estimates
As in the previous chapter, use the default settings for the Neural Network (2) node. Right-click the Model Comparison node and click Run. In the Confirmation window, click Yes. Click Results in the Run Status window.

Notice that the Neural Network (2) model, while significantly better than the initial Regression model, is considered worse than the other models. Close the Results window.

As an exercise, consider adjusting the R-Square Options or setting the Target Model property to Chi-Square.
Problem Formulation

Consider the following scenario. A baseball manager wants to identify and group players on the team who are very similar with respect to several statistics of interest. Note that there is no response variable in this example. The manager simply wants to identify different groups of players. The manager also wants to learn what differentiates players in one group from players in a different group.

The data set for this example is located in SAMPSON.DMABASE. The following table contains a description of the important variables.

<table>
<thead>
<tr>
<th>Name</th>
<th>Model Role</th>
<th>Measurement Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>ID</td>
<td>Nominal</td>
<td>Player name</td>
</tr>
<tr>
<td>TEAM</td>
<td>Rejected</td>
<td>Nominal</td>
<td>Team at the end of 1986</td>
</tr>
<tr>
<td>POSITION</td>
<td>Rejected</td>
<td>Nominal</td>
<td>Positions played at the end of 1986</td>
</tr>
<tr>
<td>LEAGUE</td>
<td>Rejected</td>
<td>Binary</td>
<td>League at the end of 1986</td>
</tr>
<tr>
<td>DIVISION</td>
<td>Rejected</td>
<td>Binary</td>
<td>Division at the end of 1986</td>
</tr>
<tr>
<td>NO_ATBAT</td>
<td>Input</td>
<td>Interval</td>
<td>Times at bat in 1986</td>
</tr>
<tr>
<td>NO_HITS</td>
<td>Input</td>
<td>Interval</td>
<td>Hits in 1986</td>
</tr>
<tr>
<td>Name</td>
<td>Model Role</td>
<td>Measurement Level</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
<td>-------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>NO_HOME</td>
<td>Input</td>
<td>Interval</td>
<td>Home runs in 1986</td>
</tr>
<tr>
<td>NO_RUNS</td>
<td>Input</td>
<td>Interval</td>
<td>Runs in 1986</td>
</tr>
<tr>
<td>NO_RBI</td>
<td>Input</td>
<td>Interval</td>
<td>RBIs in 1986</td>
</tr>
<tr>
<td>NO_BB</td>
<td>Input</td>
<td>Interval</td>
<td>Walks in 1986</td>
</tr>
<tr>
<td>YR_MAJOR</td>
<td>Input</td>
<td>Interval</td>
<td>Years in the major leagues</td>
</tr>
<tr>
<td>CR_ATBAT</td>
<td>Input</td>
<td>Interval</td>
<td>Career times at bat</td>
</tr>
<tr>
<td>CR_HITS</td>
<td>Input</td>
<td>Interval</td>
<td>Career hits</td>
</tr>
<tr>
<td>CR_HOME</td>
<td>Input</td>
<td>Interval</td>
<td>Career home runs</td>
</tr>
<tr>
<td>CR_RUNS</td>
<td>Input</td>
<td>Interval</td>
<td>Career runs</td>
</tr>
<tr>
<td>CR_RBI</td>
<td>Input</td>
<td>Interval</td>
<td>Career RBIs</td>
</tr>
<tr>
<td>CR_BB</td>
<td>Input</td>
<td>Interval</td>
<td>Career walks</td>
</tr>
<tr>
<td>NO_OUTS</td>
<td>Input</td>
<td>Interval</td>
<td>Put outs in 1986</td>
</tr>
<tr>
<td>NO_ASSTS</td>
<td>Input</td>
<td>Interval</td>
<td>Assists in 1986</td>
</tr>
<tr>
<td>NO_ERROR</td>
<td>Input</td>
<td>Interval</td>
<td>Errors in 1986</td>
</tr>
<tr>
<td>SALARY</td>
<td>Rejected</td>
<td>Interval</td>
<td>1987 salary in thousands</td>
</tr>
<tr>
<td>LOGSALAR</td>
<td>Input</td>
<td>Interval</td>
<td>Log of 1987 salary</td>
</tr>
</tbody>
</table>

For this example, you set the model role for TEAM, POSITION, LEAGUE, DIVISION, and SALARY to Rejected. SALARY is rejected because this information is contained in the LOGSALAR variable. No target variables are used in a cluster analysis or self-organizing map (SOM). If you want to identify groups based on a target variable, consider a predictive modeling technique and specify a categorical target variable.

## Cluster Analysis

### Overview

Cluster analysis 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. These
groups, or clusters, are assigned numbers. However, the cluster number cannot be used to evaluate the proximity between clusters.

Self-organizing maps attempt to create clusters and plot the resulting clusters on a map so that cluster proximity can be evaluated graphically. This example does not contain a self-organizing map. However, the SOM/Kohonen node is used to create self-organizing maps.

Building the Process Flow Diagram

This example uses the same diagram workspace that you created in Chapter 2. You have the option to create a new diagram for this example, but instructions to do so are not provided in this example. First, you need to add the SAMPSIO.DMABASE data source to project.

1. In the Project Panel, right-click Data Sources and click Create Data Source.
2. In the Data Source Wizard — Metadata Source window, click Next.
3. In the Data Source Wizard — Select a SAS Table and enter SAMPSIO.DMABASE in the Table field. Click Next.
4. In the Data Source Wizard — Table Information window, click Next.
5. In the Data Source Wizard — Metadata Advisor Options window, select Advanced. Click Next.
6. In the Data Source Wizard — Column Metadata window, make the following changes:
   • For the variable NAME, set the Role to ID.
   • For the variables TEAM, POSITION, LEAGUE, DIVISION, and SALARY set the Role to Rejected.
   • Ensure that all other variables have a Role of Input.
   Click Next.
7. In the Data Source Wizard — Create Sample window, click Next.
8. In the Data Source Wizard — Data Source Attributes window, click Next.
9. In the Data Source Wizard — Summary window, click Finish.

This creates the All: Baseball Data data source in your Project Panel. Drag the All: Baseball Data data source to your diagram workspace.

If you explore the data in the All: Baseball Data data source, you will notice that SALARY and LOGSALAR have missing values. Although it is not always necessary to impute missing values, at times the amount of missing data can prevent the Cluster node from obtaining a solution. The Cluster node needs some complete observations in order to generate the initial clusters. When the amount of missing data is too extreme, use the Replacement or Impute node to handle the missing values. This example uses imputation to replace missing values with the median.

On the Modify tab, drag an Impute node to your diagram workspace. Connect the All: Baseball Data data source to the Impute node. In the Interval Variables property subgroup, set the value of the Default Input Method property to Median.

On the Explore tab, drag a Cluster node to your diagram workspace. Connect the Impute node to the Cluster node.
By default, the Cluster node uses the Cubic Clustering Criterion (CCC) to approximate the number of clusters. The node first makes a preliminary clustering pass, beginning with the number of clusters that is specified in the Preliminary Maximum value in the Selection Criterion properties. After the preliminary pass completes, the multivariate means of the clusters are used as inputs for a second pass that uses agglomerative, hierarchical algorithms to combine and reduce the number of clusters. Then, the smallest number of clusters that meets all four of the following criteria is selected.

• The number of clusters must be greater than or equal to the number that is specified as the Minimum value in the Selection Criterion properties.

• The number of clusters must have cubic clustering criterion statistic values that are greater than the CCC Cutoff that is specified in the Selection Criterion properties.

• The number of clusters must be less than or equal to the Final Maximum value.

• A peak in the number of clusters exists.

**Note:** If the data to be clustered is such that the four criteria are not met, then the number of clusters is set to the first local peak. In this event, the following warning is displayed in the Cluster node log: 

**WARNING:** The number of clusters selected based on the CCC values may not be valid. Please refer to the documentation of the Cubic Clustering Criterion. There are no number of clusters matching the specified minimum and maximum number of clusters. The number of clusters will be set to the first local peak. After the number of clusters is determined, a final pass runs to produce the final clustering for the Automatic setting.

You are now ready to cluster the input data.

**Using the Cluster Node**

Right-click the Cluster node and click Run. In the Confirmation window, click Yes. Click Results in the Run Status window.
Notice in the Segment Size window that the Cluster node created 4 clusters.

On the main menu, select View ⇒ Cluster Profile ⇒ Variable Importance. The Variable Importance window displays each variable that was used to generate the clusters and their relative importance. Notice that NO_ASSTS, NO_ERROR, and NO_OUTS have an Importance of 0. These variables were not used by the Cluster node when the final clusters were created.

On the main menu, select View ⇒ Summary Statistics ⇒ Input Means Plot.
This plot displays the normalized mean value for each variable, both inside each cluster and for the complete data set. Notice that the in-cluster mean for cluster 1 is always less than the overall mean. But, in cluster 4, the in-cluster mean is almost always greater than the overall mean. Clusters 2 and 3 each contain some in-cluster means below the overall mean and some in-cluster means above the overall mean.

From the Input Means Plot, you can infer that the players in cluster 1 are younger players that are earning a below average salary. Their 1986 and career statistics are also below average. Conversely, the players in cluster 4 are veteran players with above average 1986 and career statistics. These players receive an above average salary. The players in clusters 2 and 3 excel in some areas, but perform poorly in others. Their average salaries are slightly above average.

Close the Results window.

**Examining the Clusters**

Select the Cluster node in your process flow diagram. Click the ellipsis button next to the Exported Data property. The Exported Data — Cluster window appears. Click TRAIN and click Explore.

The Clus_TRAIN window contains the entire input data set and three additional columns that are appended to the end. Scroll to the right end of the window to locate the Segment ID, Distance, and Segment Description columns. The Segment ID and Segment Description columns display what cluster each observation belongs to.
Next, create a plot that compares the number of hits for each cluster.

1. On the main menu, select **Actions ➔ Plot**.
2. In the Select a Chart Type window, click **Box**. Click **Next**
3. In the Select Chart Roles window, find the _SEGMENT_LABEL_ variable. Set the **Role** for _SEGMENT_LABEL_ to **X**. Next, find the NO_HITS variable. Set the **Role** for NO_HITS to **Y**. Click **Next**.
4. In the Data WHERE Clause window, click **Next**.
5. In the Chart Titles window, enter **Number of Hits in Each Cluster** in the **Title** field. Leave the **Footnote** field blank. Enter **Segment** for the **X Axis Label**. Enter **Hits** for the **Y Axis Label**. Click **Next**.
6. In the Chart Legends window, click **Finish**.
Notice that the average number of hits in clusters 1 and 3 is significantly lower than the average number of hits in clusters 2 and 4. You should repeat these steps to create box plots for several other variables.
Chapter 5
Association Analysis

Problem Formulation

Consider the following scenario. A store wants to examine its customer base and to understand which of its products tend to be purchased together. It has chosen to conduct a market basket analysis on a sample of its customers.

The ASSOCS data set lists 20 grocery products that are purchased by 1,001 customers. Seven items were purchased by each of the 1,001 customers, which yields 7,007 rows in the input data set. Each row of the data set represents a customer-product combination. In most data sets, not all customers have the same number of products.

Association Analysis

Building the Process Flow Diagram

This example uses the same diagram workspace that you created in Chapter 2. You have the option to create a new diagram for this example, but instructions to do so are not provided in this example. First, you need to add the SAMPSIO.ASSOCS data source to project.

1. In the Project Panel, right-click Data Sources and click Create Data Source.
2. In the Data Source Wizard — Metadata Source window, click Next.
3. In the Data Source Wizard — Select a SAS Table and enter SAMPSIO.ASSOCS in the Table field. Click Next.
4. In the Data Source Wizard — Table Information window, click Next.
5. In the Data Source Wizard — Metadata Advisor Options window, click Advanced. Click Next.
6. In the Data Source Wizard — Column Metadata window, make the following changes:
   • For the variable CUSTOMER, set the Role to ID.
   • For the variable PRODUCT, set the Role to Target.
   • For the variable TIME, set the Role to Rejected.

   Note: The variable TIME identifies the sequence in which the products were purchased. In this example, all of the products were purchased at the same time, so the order relates only to the order in which they are scanned at the register. When order is taken into account, association analysis is known as sequence analysis. Sequence analysis is not demonstrated here.

   Click Next.

7. In the Data Source Wizard — Decision Configuration window, click Next.

8. In the Data Source Wizard — Create Sample window, click Next.

9. In the Data Source Wizard — Data Source Attributes window, set the Role of the data source to Transaction. Click Next.

10. In the Data Source Wizard — Summary window, click Finish.

   In the Project Panel, drag the ASSOCS data source to your diagram workspace. On the Explore tab, drag an Association node to your diagram workspace. Connect the ASSOCS data source to the Association node.

---

Understanding Analysis Modes

To perform association discovery, the input data set must have a separate observation for each product purchased by each customer. You must assign the ID role to one variable and the Target model role to another variable when you create the data source.

To perform sequence discovery, the input data set must have a separate observation for each product purchased by each customer at each visit. In addition to assignment of ID and Target roles, your input data must contain a Sequence variable. The sequence variable is used for timing comparisons. It can have any numeric value including date/time values. The time or span from observation to observation in the input data set must be on the same scale.

Because you set the role of TIME to Rejected, the Association node performs an association analysis in this example.

Observe the Association properties subgroup of the Association node. These properties determine how large each association can be and how association rules are formed. Set the value of the Maximum Items property to 2. This indicates that only associations between pairs of products are generated.

---

Running the Association Node

In your diagram workspace, right-click the Association node and click Run. In the Confirmation window, click Yes. Click Results in the Run Status window.
In the Results window, select View ⇒ Rules ⇒ Rules Table from the main menu. The Rules Table displays information about each rule that was created. This includes the confidence, support, lift, number of occurrences, and the items in the rule. To explain confidence, support, and lift, consider the rule A => B where A and B each represent one product.

- The support percentage for A => B is the percentage of all customers who purchased both A and B. Support is a measure of how frequently the rule occurs in the database.

- The confidence percentage for A => B is the percentage of all customers who purchased both A and B, divided by the number of customers who purchased A.

- The lift of A => B is a measure of the strength of the association. For example, if the lift is 2 for A => B, then a customer who purchased A is twice as likely than a customer chosen at random to purchase B.

Sort the Rules Table by Support (%). Notice that the top two rules are heineken ==> cracker and cracker ==> heineken, both with a support of 36.56%. This indicates that 36.56% of all customers purchased beer and crackers together.

The confidence for heineken ==> cracker is 61%, which indicates that 61% of customers who purchased Heineken then purchased crackers. For the rule cracker ==> heineken, the confidence is 75%. This means that 75% of the customers who purchased crackers then purchased Heineken.

Lift, in the context of association rules, is the ratio of the confidence of a rule to the expected confidence of the rule. The expected confidence is calculated under the assumption that the left hand side of a rule is independent from the right hand side of the rule. Consequently, lift is a measure of association between the left hand side and right hand side of the rule. Values that are greater than one represent positive association between the left and right hand sides. Values that are equal to one represent independence. Values that are less than one represent negative association between the left and right hand sides.

Sort the Rules Table by Lift. Notice that the rule ice_crea ==> coke has the greatest lift. This indicates that customers who buy ice cream are 2.38 times more likely to buy Coke than a customer chosen at random.

Close the Results window.
Problem Formulation

Consider the following scenario. As part of your business, you sell books and other publications from your website. You web log data contains information about the navigation patterns that visitors make within your website. You want to examine the files that are requested. In addition, you want to determine the most commonly occurring paths that visitors take from your home page to the check-out page. In other words, you want to determine what paths visitors take when they make a purchase.

The data set SAMPSON.WEBPATH contains a set of hypothetical web log data. The data set contains the following information:

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Model Role</th>
<th>Measurement Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REFERRER</td>
<td>Rejected</td>
<td>Nominal</td>
<td>Referring URL</td>
</tr>
<tr>
<td>REQUESTED_FILE</td>
<td>Target</td>
<td>Nominal</td>
<td>File that the visitor clicked</td>
</tr>
<tr>
<td>SESSION_IDENTIFIER</td>
<td>ID</td>
<td>Nominal</td>
<td>Unique ID code for the session</td>
</tr>
<tr>
<td>SESSION_SEQUENCE</td>
<td>Sequence</td>
<td>Interval</td>
<td>Order in which files were requested within a session</td>
</tr>
</tbody>
</table>
Examining Web Log Data

Building the Process Flow Diagram

This example uses the same diagram workspace that you created in Chapter 2. You have the option to create a new diagram for this example, but instructions to do so are not provided in this example. First, you need to add the SAMPSIO.WEBPATH data source to project.

1. In the Project Panel, right-click Data Sources and click Create Data Source.
2. In the Data Source Wizard — Metadata Source window, click Next.
3. In the Data Source Wizard — Select a SAS Table and enter SAMPSIO.WEBPATH in the Table field. Click Next.
4. In the Data Source Wizard — Table Information window, click Next.
5. In the Data Source Wizard — Metadata Advisor Options window, click Advanced. Click Next.
6. In the Data Source Wizard — Column Metadata window, make the following changes:
   • For the variable REFERRER, set the Role to Input.
   • For the variable REQUESTED_FILE, set the Role to Target.
   • For the variable SESSION_ID, set the Role to ID.
   • For the variable SESSION_SEQUENCE, set the Role to Sequence.
   Click Next.
7. In the Data Source Wizard — Decision Configuration window, click Next.
8. In the Data Source Wizard — Create Sample window, click Next.
9. In the Data Source Wizard — Data Source Attributes window, set the Role of the data source to Transaction. Click Next.
10. In the Data Source Wizard — Summary window, click Finish.

In the Project Panel, drag the WEBPATH data source to your diagram workspace. From the Explore tab, drag a Link Analysis node to your diagram workspace. Connect the WEBPATH data source to the Link Analysis node.

Select the Link Analysis node. Set the value of the Association Support Type property to Count. Set the value of the Association Support Count property to 1. This ensures that all paths are captured by the Link Analysis node, including visitors that requested just a single page. Set the Minimum Confidence (%) property to 50.
Running the Link Analysis Node

In your diagram workspace, right-click the **Link Analysis** node and click **Run**. In the Confirmation window, click **Yes**. Click **Results** in the Run Status window.

Maximize the Items Constellation Plot window. The Items Constellation Plot shows all of the links to and from each page. The arrows indicate the direction of travel. In the upper left corner, set the **Node** value on the drop-down menu to `/Cart.jsp`. This puts the `/Cart.jsp` node in the center of the diagram. The diagram now contains only the nodes that are connected to `/Cart.jsp`. 
The thickness of each arrow represents the relative frequency of that link. In this example, the most frequent links appear to be from /Cart.jsp to /Confirm.jsp, /Summary.jsp, /Billing.jsp, and /Shipping.jsp.

Because SESSION_SEQUENCE is a sequence variable, these links are directed. If you set the role of SESSION_SEQUENCE to Rejected, then the links are undirected. Undirected links would not detect the beginning and ending web page, but would instead treat each direction as the same path. For comparison, the undirected Items Constellation Plot is shown below.
In addition to creating undirected links, the **Link Analysis** node performed item-cluster detection because there was no sequence variable. The color of each node in the Items Constellation Plot indicates the cluster that the node belongs to. Minimize the Items Constellation Plot.

Maximize the Node Frequency Histogram (by Item-cluster) window.
Notice that the first cluster contains the page /Cart.jsp. This indicates that the rest of the pages located in the first cluster are very similar to /Cart.jsp.

Close the Results window.