THE POWER TO KNOW.

Data Mining Using SAS® Enterprise Miner™: A Case Study Approach, Fourth Edition

SAS® Documentation
March 26, 2018

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2018. *Data Mining Using SAS® Enterprise Miner™: A Case Study Approach, Fourth Edition*. Cary, NC: SAS Institute Inc.

Data Mining Using SAS® Enterprise Miner™: A Case Study Approach, Third Edition

Copyright © 2018, SAS Institute Inc., Cary, NC, USA

ISBN 978-1-61290-525-9 (electronic book)
ISBN 978-1-61290-638–6

All rights reserved. Produced in the United States of America.

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

ISBN 978-1-61290-525-9
Electronic book 1, February 2018

ISBN 978-1-61290-638–6
Printing 1, February 2018

Contents

About This Book v

Chapter 1 • Introduction to SAS Enterprise Miner **1**
- Starting SAS Enterprise Miner 1
- Setting Up the Initial Project and Diagram 3
- Identifying the Interface Components 4
- Data Mining and SEMMA 5
- Accessing SAS Data through SAS Libraries 27

Chapter 2 • Predictive Modeling **29**
- Problem Formulation 30
- Creating a Process Flow Diagram 31
- Data Preparation and Investigation 44
- Fitting and Comparing Candidate Models 56
- Generating and Using Scoring Code 64
- Generating a Report Using the Reporter Node 68

Chapter 3 • Variable Selection **71**
- Introduction to Variable Selection 71
- Using the Variable Selection Node 71

Chapter 4 • Clustering Tools **75**
- Problem Formulation 75
- Cluster Analysis 76

Chapter 5 • Association Analysis **85**
- Problem Formulation 85
- Association Analysis 85

Chapter 6 • Link Analysis **91**
- Problem Formulation 91
- Examining Web Log Data 92

Recommended Reading **99**

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.

Chapter 1

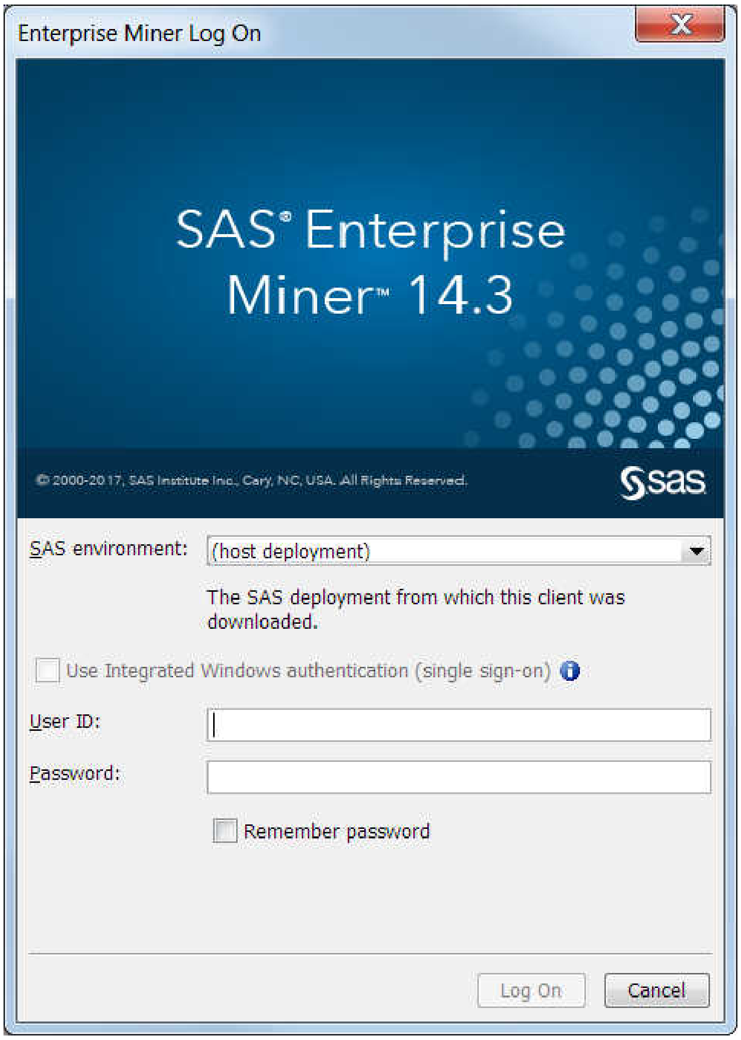
Introduction to SAS Enterprise Miner

Starting SAS Enterprise Miner 1
Setting Up the Initial Project and Diagram 3
Identifying the Interface Components 4
Data Mining and SEMMA 5
Definition of Data Mining 5
Overview of the Data 5
Predictive and Descriptive Techniques 6
Overview of SEMMA 6
Overview of the Nodes 7
Some General Usage Rules for Nodes 26
Accessing SAS Data through SAS Libraries 27

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*.

Some SAS Enterprise Miner installations provide a Java Web Start facility. When you start a SAS Enterprise Miner session from Java Web Start, the client logon resembles the following:



Enterprise Miner Log On

SAS® Enterprise Miner™ 14.3

© 2000-2017, SAS Institute Inc., Cary, NC, USA. All Rights Reserved.

SAS environment: (host deployment)

The SAS deployment from which this client was downloaded.

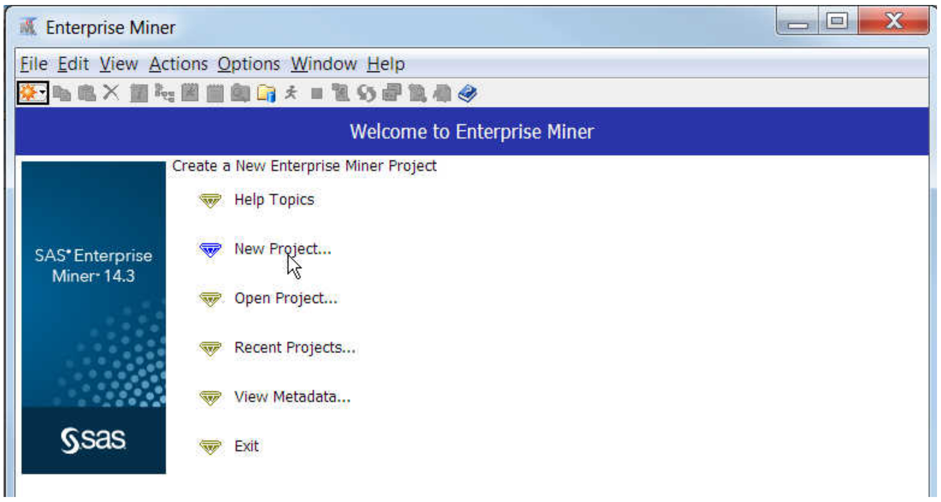
Use Integrated Windows authentication (single sign-on)

User ID:

Password:

Remember password

The client start takes you to a Welcome to Enterprise Miner page where you can start a new project, open an existing project, browse recent projects, or view Help topics or SAS Metadata. If you reach this page, select **New Project** to begin the *Data Mining with SAS Enterprise Miner: A Case Study Approach* exercises.



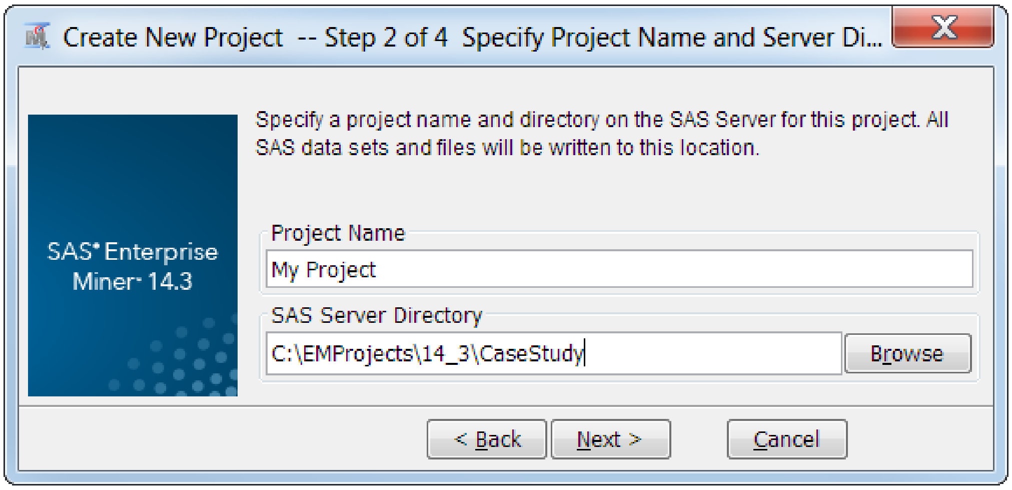
Enterprise Miner
File Edit View Actions Options Window Help
Welcome to Enterprise Miner
SAS® Enterprise Miner™ 14.3
Create a New Enterprise Miner Project
Help Topics
New Project...
Open Project...
Recent Projects...
View Metadata...
Exit

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**.


Create New Project -- Step 2 of 4 Specify Project Name and Server Di...

Specify a project name and directory on the SAS Server for this project. All SAS data sets and files will be written to this location.

Project Name: My Project

SAS Server Directory: C:\EMProjects\14_3\CaseStudy

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** ⇨ **New** ⇨ **Diagram** on the main menu.
7. In the Create New Diagram window, enter the name of your diagram. Click **OK**.

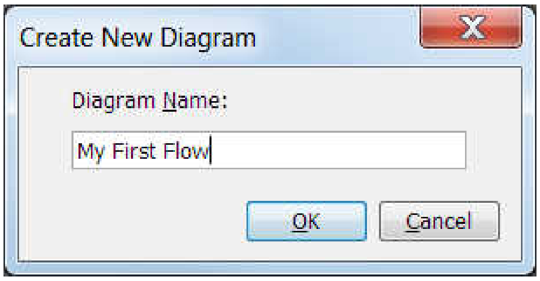
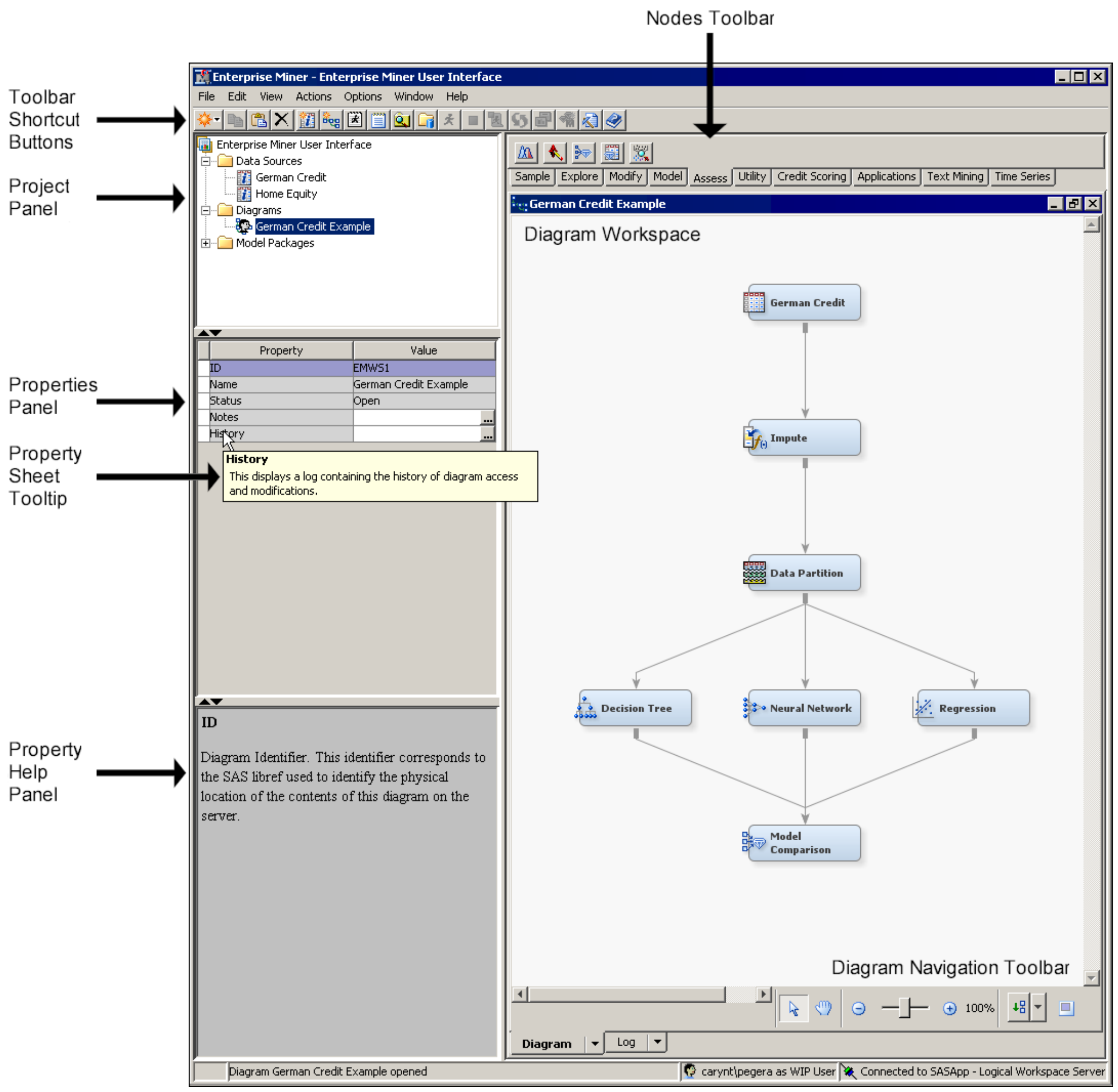

Create New Diagram

Diagram Name: My First Flow

Identifying the Interface Components

Figure 1.1 *SAS Enterprise Miner User Interface*



Property	Value
ID	EMWS1
Name	German Credit Example
Status	Open
Notes	
History	

History
This displays a log containing the history of diagram access and modifications.

ID

Diagram Identifier. This identifier corresponds to the SAS libref used to identify the physical location of the contents of this diagram on the server.

Diagram German Credit Example opened

caryntl\pegera as WIP User

Connected to SASApp - Logical Workspace Server

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 interval or 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 **HPDM**, **Utility**, **Time Series**, 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.

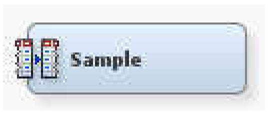
The **Data Partition** node enables you to partition data sets into training, test, and validation data sets. The training data set is used for preliminary model fitting. The validation data set is used to monitor and tune the model weights during estimation and is also used for model assessment. The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

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.



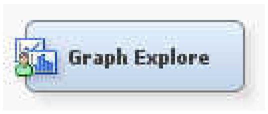
The **Sample** node enables you to take random, stratified random, and cluster samples of data sets. Sampling is recommended for extremely large databases because it can significantly decrease model training time. If the sample is sufficiently representative, then relationships found in the sample can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

Explore 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 **MultPlot** 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.

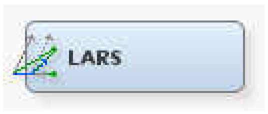
The **Decision Tree** node enables you to fit decision tree models to your data. The implementation includes features that are found in a variety of popular decision tree algorithms (for example, CHAID, CART, and C4.5). The node supports both automatic and interactive training. When you run the Decision 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.

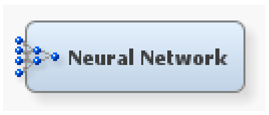
The **Gradient Boosting** node uses tree boosting to create a series of decision trees that together form a single predictive model. Each tree in the series is fit to the residual of the prediction from the earlier trees in the series. The residual is defined in terms of the derivative of a loss function. 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.

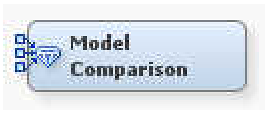
The **Rule Induction** node enables you to improve the classification of rare events in your modeling data. The **Rule Induction** node creates a Rule Induction model that uses split techniques to remove the largest pure split node from the data. Rule Induction also creates binary models for each level 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 **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.

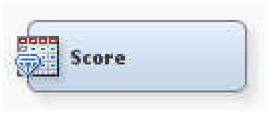
Assess Nodes

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 creates 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 Open Source Integration node enables you to write code in the R language inside SAS Enterprise Miner. The Open Source Integration node makes SAS Enterprise Miner data and metadata available to your R code and returns R results to SAS Enterprise Miner. In addition to training and scoring supervised and unsupervised R models, the Open Source Integration node allows for data transformation and data exploration. Further, when you run an appropriate R package with Output Mode set to PMML, your results should contain the standard assessment plots and tables that are associated with the corresponding modeling node.

The Register Model node enables you to register segmentation, classification, or prediction models to the SAS Metadata Server. Models registered in metadata can be submitted to SAS Model Manager, used to score data in SAS Enterprise Guide, or used to score data in SAS Enterprise Miner. Information about input variables, output variables, target variables, target levels, mining function, training data, and SAS score code is registered to the metadata.

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 SAS Viya Code node is a modified SAS Code node. You are expected to write your own code to be executed in SAS Viya and CAS. The node is template-based and is populated with basic training code. You should modify this code with your desired code. The template contains utility macros to simplify the integration of your code with SAS Enterprise Miner.

Saving data from a process flow diagram once required you to use the SAS Code node in conjunction with an external SAS library to export your data. Alternatively, you can use the operating system to copy the data out of the project directories. The Save Data node combines these methods and enables you to save training, validation, test, score, or transaction data from a node to either a previously defined SAS library or a specified file path. The Save Data node can export JMP, Excel 2010, CSV, and tab-delimited files. The default options are designed so that the node can be deployed in SAS Enterprise Miner batch programs without user input.

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.

Credit Scoring Nodes

Note: The Credit Scoring for SAS Enterprise Miner solution is not included with the base version of SAS Enterprise Miner. If your site has not licensed Credit Scoring for SAS Enterprise Miner, the credit scoring node tools do not appear in your SAS Enterprise Miner software.

The **Credit Exchange** node enables you to exchange the data that is created in SAS Enterprise Miner with the SAS Credit Risk Management solution.

The **Credit Exchange** node creates the following data sources:

- Statistics Table
- Mapping Table

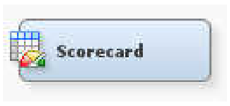
You use the **Interactive Grouping** node to perform grouping and initial characteristic screening. A characteristic is an observable trait, quality, or property of a credit applicant. Scorecard characteristics can be selected from any of the sources of data that are available to the lender. Examples of such characteristics are demographics (for example, age, time at residence, time at job, and postal code), existing relationship (for example, time at bank, number of products, payment performance, and previous claims), credit bureau (for example, inquiries, trades, delinquencies, and public records), real estate data, and so on. Thus, characteristics can be represented in your data as interval, nominal, ordinal, or binary variables. Each possible value of a characteristic is called an attribute of the characteristic. Thus, before any binning or grouping is performed, binary, ordinal, and nominal variables have a fixed, finite number of attributes while interval characteristics have an infinite number of attributes.

Credit scoring models are built with a fundamental bias (selection bias). The sample data that was used to develop a credit scoring model is structurally different from the "through-the-door" population to which the credit scoring model is applied. The non-event/event target variable that is created for the credit scoring model is based on the records of applicants who were all accepted for credit. However, the population to which the credit scoring model is applied is composed includes applicants who would have been rejected under the scoring rules that were used to generate the initial model.

One remedy for this selection bias is to use reject inference. The reject inference approach uses the model that was trained using the accepted applications to score the rejected applications. The observations in the rejected data set are classified as inferred non-event and inferred event using one of three available methods. The inferred observations are then added to the accepts data set, which contains the actual non-event and event records, to form an augmented data set. This augmented data set, which represents the "through-the-door" population, serves as the training data set for a second scorecard model.

When you create inferred data from the rejects data set, you predict how the rejected customer would have performed if they had been accepted. The Enterprise Miner **Reject Inference** node provides three different methods that you can use to create inferred data from the rejects data set: Fuzzy, Hard Cutoff, and Parceling.

Once the rejects data has been scored and the observations have been classified as non-events or events, the rejects data is appended to the accepts data to form an augmented data set. If you over-sampled the events in the accepts data and generated frequency weights, those weights are used in the augmented data set for the accepts observations. If you did not use frequency weights for the accepts data, the frequency weight is set to 1 for the accepts observations in the augmented data set.



Credit scores are designed to reflect the odds of an applicant being a "good" credit risk versus being a "bad" credit risk. However, "good" and "bad" are defined. The Scorecard node computes credit scores in a two-step process.

In the first step, the **Scorecard** node fits a logistic regression model, which estimates the ln(odds) as a linear function of the characteristics. The **Scorecard** node uses the data set that is exported by the **Interactive Grouping** node as the model's input data set. You can choose to use either the characteristics' Weight Of Evidence variables or the group variables as inputs for the logistic regression model. The node's Model Selection properties enable you to choose from a variety of model selection methods and selection criteria.

In the second step, the Scorecard node applies a linear transformation to the predicted ln(odds) to compute a score for each attribute of each characteristic. The score for an attribute of a characteristic satisfies the following equation: score = ln(odds) * factor + offset.

HPDM Nodes

A Bayesian network is a directed, acyclic graphical model in which the nodes represent random variables, and the links between the nodes represent conditional dependency between two random variables. The structure of a Bayesian network is based on the conditional dependency between the variables. As a result, a conditional probability table is created for each node in the network. Because of these features, Bayesian networks can be effective predictive models during supervised data mining.

The **HP Bayesian Network Classifier** node supports several network structures. In this context, the network structure specifies both which nodes are connected and the direction in which they are connected. For the **HP Bayesian Network Classifier** node, a child node is conditionally dependent on a parent node, and connections are made from parent nodes to child nodes.

Use the **HP Cluster** node to perform observation clustering, which can be used to segment databases. Clustering places objects into groups or clusters suggested by the data. The objects in each cluster tend to be similar to each other in some sense, and objects in different clusters tend to be dissimilar. If obvious clusters or groupings can be developed prior to the analysis, then the clustering analysis can be performed by simply sorting the data.

The clustering methods in the **HP Cluster** node perform disjoint cluster analysis on the basis of Euclidean distances. Euclidean distances are computed from one or more quantitative variables and seeds that are generated and updated by the algorithm. You can specify the clustering criterion that is used to measure the distance between data observations and seeds. The observations are divided into clusters so that every observation belongs to at most one cluster.

After clustering is performed, the characteristics of the clusters can be examined graphically. The consistency of clusters across variables is of particular interest. The multidimensional charts and plots enable you to graphically compare the clusters.

Most data mining projects use large volumes of sampled data. After sampling, the data is usually partitioned before modeling.

Use the **HP Data Partition** node to partition your input data into one of the following data sets:

Train
: used for preliminary model fitting. The analyst tries to find the best model weights by using this data set.

Validation
: used to assess the adequacy of the model in the Model Comparison node.

The validation data is also used for fine-tuning models in the following nodes:

- **HP Forest** node — to create a decision tree model.

- **HP Neural** node — to choose among network architectures or for the early-stopping of the training algorithm.
- **HP Regression** node — to choose a final subset of predictors from all the subsets that are computed during stepwise regression.

The **HP Explore** node enables you to obtain descriptive information about a training data set. Statistics such as mean, standard deviation, minimum value, maximum value, and percentage of missing values are generated for each input variable and displayed in tabular and graphical form. These descriptive statistics are important tools in the pre-model building process. For example, if the **HP Explore** node identifies variables with missing values, you can impute these values using the **HP Impute** node.

The **HP Forest** node creates a predictive model called a forest. A forest consists of several decision trees that differ from each other in two ways. First, the training data for a tree is a sample without replacement from all available observations. Second, the input variables that are considered for splitting a node are randomly selected from all available inputs. In other respects, trees in a forest are trained like standard trees. ⇨ The **HP Forest** node accepts interval and nominal target variables. For an interval target, the procedure averages the predictions of the individual trees to predict an observation. For a categorical target, the posterior probabilities in the forest are the averages of the posterior probabilities of the individual trees. The node makes a second prediction by voting: the forest predicts the target category that the individual trees predict most often.

A generalized linear model (GLM) is an extension of traditional linear model that allows the population mean to depend on a linear predictor through a nonlinear link function. For example, GLMs can be used to model traditional insurance measures such as claim frequency, severity, or pure premium. Claim frequency is typically modeled with a Poisson distribution and a logarithmic link function. Claim severity is typically modeled with a gamma distribution and a logarithmic link function. Pure premiums are modeled with the Tweedie distribution.

Note that when you specify a discrete distribution for the Interval Target Probability Distribution property model assessment is not performed and model comparison with the Model Comparison node is unavailable.

The **HP GLM** node is able to fit models for standard distributions in the exponential family. The **HP GLM** node fits zero-inflated Poisson and negative binomial models for count data.

Use the **HP Impute** node to replace missing values in data sets that are used for data mining. The HP Impute node is typically used during the modification phase of the Sample, Explore, Modify, Model, and Assess (SEMMA) SAS data mining methodology.

Data mining databases often contain observations that have missing values for one or more variables. Missing values can result from the following: data collection errors; incomplete customer responses; actual system and measurement failures; or from a revision of the data collection scope over time (such as tracking new variables that were not included in the previous data collection schema).

If an observation contains a missing value, then by default that observation is not used for modeling by nodes such as **HP Neural** or **HP Regression**. However, rejecting all incomplete observations might ignore useful or important information that is still contained in the nonmissing variables. Rejecting all incomplete observations might also bias the sample, since observations that are missing values might have other things in common as well.

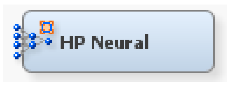
Choosing the "best" missing value replacement technique inherently requires the researcher to make assumptions about the true (missing) data. For example, researchers often replace a missing value with the mean of the variable. This approach assumes that the variable's data distribution follows a normal population response. Replacing missing values with the mean, median, or another measure of central tendency is simple. But it can greatly affect a variable's sample distribution. You should use these replacement statistics carefully and only when the effect is minimal.

The **HP Impute** node provides the following imputations for missing interval variables:

- Default Constant Value
- Mean
- Maximum
- Minimum
- Midrange
- None

The **HP Impute** node provides the following imputations for missing class variables:

- Count
- Default Constant Value
- Distribution
- None



The **HP Neural** node was designed with two goals. First, the **HP Neural** node aims to perform efficient, high-speed training of neural networks. Second, the **HP Neural** node attempts to create accurate, generalizable models in an easy to use manner. With these goals in mind, most parameters for the neural network are selected automatically. This includes standardization of input and target variables, activation and error functions, and termination of model training.

The **HP Neural** node creates multilayer neural networks that pass information from one layer to the next in order to map an input to a specific category or predicted value. The **HP Neural** node enables this mapping to take place in a distributed computing environment, which enables you to build neural networks on massive data sets in a relatively short amount of time.

The **HP Principal Components** node calculates eigenvalues and eigenvectors from the covariance matrix or the correlation matrix of input variables. Principal components are calculated from the eigenvectors and can be used as inputs for successor modeling nodes in the process flow diagram. Since interpreting principal components is often problematic or impossible, it is much safer to view them simply as a mathematical transformation of the set of original variables.

A principal components analysis is useful for data interpretation and data dimension reduction. It is usually an intermediate step in the data mining process. Principal components are uncorrelated linear combinations of the original input variables; they depend on the covariance matrix or the correlation matrix of the original input variables. Principal components are usually treated as the new set of input variables for successor modeling nodes.

The **HP Regression** node fits a linear regression or a logistic regression for an interval or binary target variable that is specified in the training data set. Linear regression attempts to predict the value of an interval target as a linear function of one or more independent inputs. Logistic regression attempts to predict the probability that a binary target will acquire the event of interest based on a specified link function of one or more independent inputs.

The **HP Regression** node supports interval, binary, nominal, and ordinal target variables. For example, when modeling customer profiles, a variable named Purchase that indicates whether a customer made a purchase can be modeled as a binary target variable. If your customer profiles also contain a variable named Value that indicates the amount spent, this variable can be modeled as an interval target variable. The **HP Regression** node does not supports the modeling of more than one target variable.

A support vector machine (SVM) is a supervised machine-learning method that is used to perform classification and regression analysis. Vapnik (1995) developed the concept of SVM in terms of hard margin, and later he and his colleague proposed the SVM with slack variables, which is a soft margin classifier. The standard SVM model solves binary classification problems that produce non-probability output (only sign +1/-1) by constructing a set of hyperplanes that maximize the margin between two classes. Most problems in a finite dimensional space are not linearly separable. In this case, the original space needs to be mapped into a much higher dimensional space or an infinite dimensional space, which makes the separation easier. SVM uses a kernel function to define the larger dimensional space.

The **HP SVM** node supports only binary classification problems, including polynomial, radial basis function, and sigmoid nonlinear kernels. The **HP SVM** node does not perform multi-class problems or support vector regression.

The **HP Text Miner** node enables you to build predictive models for a document collection in a distributed computing environment. Data is processed in two phases: text parsing and transformation. Text parsing processes textual data into a term-by-document frequency matrix. Transformations such as singular value decomposition (SVD) alter this matrix into a data set that is suitable for data mining purposes. A document collection of millions of documents and hundreds of thousands of terms can be represented in a compact and efficient form.

You can connect the **HP Text Miner** node with other HP nodes to perform modeling. The **HP Text Miner** node supports Chinese, Dutch, English, Finnish, French, German, Italian, Japanese, Korean, Portuguese, Russian, Spanish, and Turkish.

The **HP Transform** node enables you to make transformations to your interval input and target variables. Interval input transformations are important for improving the fit of your model. Transformation of variables can be used to stabilize variances, remove nonlinearity, and correct non-normality in variables.

The **HP Tree** node enables you to create and visualize a tree model and determine input variable importance. Trees are created using PROC HPSPLIT, which was designed for the high-performance data mining environment.

The **HP Tree** node includes many of the tools and results found in the Decision Tree node. For interval targets, the **HP Tree** node supports the Variance and F Test splitting criteria and the Average Square Error pruning criterion. For categorical targets, the **HP Tree** node supports the Entropy, Fast CHAID, Gini, and Chi-Square splitting criteria and the Gini, Entropy, Average Square Error, and Misclassification rate pruning criteria. Also available are standard visualization and assessments plots such as the tree diagram, treemap, leaf statistics, subtree assessment plot, and Node Rules.

Many data mining databases have hundreds of potential model inputs (independent or explanatory variables) that can be used to predict the target (dependent or response variable). The **HP Variable Selection** node reduces the number of inputs by identifying the input variables that are not related to the target variable, and rejecting them. Although rejected variables are passed to subsequent nodes in the process flow, these variables are not used as model inputs by a successor modeling nodes.

The **HP Variable Selection** node quickly identifies input variables that are useful for predicting the target variables. The use status Input is assigned to these variables. You can override the automatic selection process by assigning the status Input to a rejected variable or the status Rejected to an input variable. The information-rich inputs are then evaluated in more detail by one of the modeling nodes.

The **HP Variable Selection** node provides both unsupervised and supervised variable selection. The input interval variables that have more missing data than desired or input class variables that have more levels than desired are excluded in the pre-selection stage. The unsupervised model performs variable selection by identifying a set of variables that jointly explain the maximal amount of data variance. Supervised variable selection includes LASSO, LARS, and stepwise regression analysis. The node can be run prior to any other analysis and the results passed to any SAS Enterprise Miner node or any procedure in the SAS System.

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 **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.

Time Series Nodes

The `Time Series Correlation` node provides auto-correlation and cross-correlation analysis for time stamped data. Cross-correlation analysis is a similarity measure of two time series as a function of time lag, which indicates the time points where the two series are correlated. Auto-correlation analysis is a special case of cross-correlation analysis where the input time series is cross-correlated with itself. Therefore, the auto-correlation function similarity peaks at lag zero.

Over time, businesses collect large amounts of data related to the business activities and transactions that they conduct with their customers, suppliers, and monitoring devices. Time Series analysis enables you to better understand trends and seasonal variations in time series data, such as the buying patterns of your customers, your buying patterns from your suppliers, or the ongoing performance of your equipment and machinery.

The **TS Data Preparation** node helps you organize and format your data for time series data mining. For example, a business might have many suppliers and many customers, both with very large sets of associated transactional data. Traditional data mining tasks become difficult because of the size of the data set. By condensing the information into a time series, you can discover trends and seasonal variations in the data, such as customer and supplier habits that might not have been visible in the transactional data.

The **TS Decomposition** node is used for classical seasonal decomposition of the time series data.

As a time series grows longer, its dimensionality increases and the sparsity of the data increases. High-dimension data requires large amounts of storage. It might be dissimilar to low-dimension data due to the sparsity of the data. Due to these conditions, similarity analysis, clustering, or both might not work well with high-dimension data. These issues, however, can be addressed with dimension reduction techniques.

The **TS Dimension Reduction** node currently supports five time series dimension reduction techniques. These techniques are the Discrete Wavelet Transform, Discrete Fourier Transform, Singular Value Decomposition, Line Segment Approximation with the Mean, and Line Segment Approximation with the Sum. The goal is to approximate the original time series with a more compact time series that retains as much information from the original series as possible.

The **TS Exponential Smoothing** node generates forecasts and some outputs that are useful for data mining. The node uses exponential smoothing models that have optimized smoothing weights for time series data and transaction data.

The **TS Exponential Smoothing** node offers the following forecasting models:

- Simple (single) exponential smoothing
- Double (Brown) exponential smoothing
- Linear (Holt) exponential smoothing
- Damped trend exponential smoothing
- Additive seasonal exponential smoothing
- Multiplicative seasonal exponential smoothing
- Winters multiplicative method
- Winters additive method

The **TS Similarity** node computes similarity measures for timestamped data with respect to time. The tool does so by accumulating the data into a time series format, and then it computes several similarity measures for sequentially ordered numeric data by respecting the ordering of the data.

Some General Usage Rules for Nodes

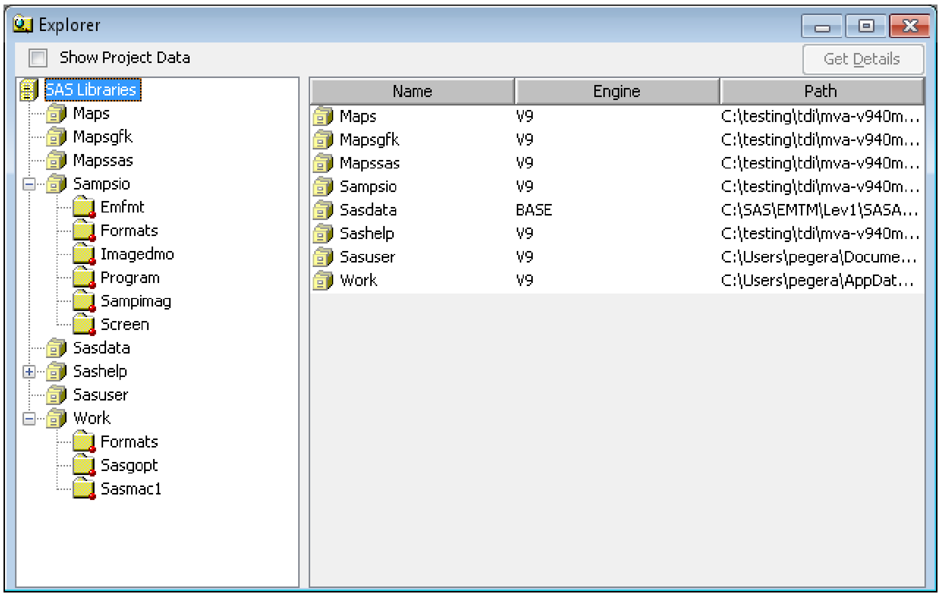
These are some general rules that govern how you place nodes in a process flow diagram.

- 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.

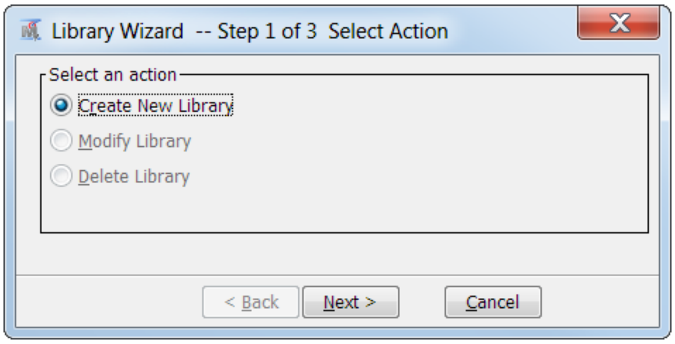
Accessing SAS Data through SAS Libraries

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.



Name	Engine	Path
Maps	V9	C:\testing\tdi\mva-v940m...
Mapsgfk	V9	C:\testing\tdi\mva-v940m...
Mapssas	V9	C:\testing\tdi\mva-v940m...
Sampsio	V9	C:\testing\tdi\mva-v940m...
Sasdata	BASE	C:\SAS\EMTM\Lev1\SASA...
Sashelp	V9	C:\testing\tdi\mva-v940m...
Sasuser	V9	C:\Users\pegera\Docume...
Work	V9	C:\Users\pegera\AppDat...

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 **30**
Overview of the Predictive Modeling Case 30
Input Data Source 30

Creating a Process Flow Diagram **31**
Identifying the Input Data 31
Adding the Nodes 33
Using the Cursor 33
Understanding the Metadata 34
Inspecting Distribution 35
Investigating Descriptive Statistics 35
Inspecting the Default Settings of the Data Partition Node 36
Fitting and Evaluating a Regression Model 37
Understanding Data Imputation 39
Fitting and Evaluating a Regression Model with Data Imputation 41

Data Preparation and Investigation **44**
Preliminary Investigation 44
Performing Variable Transformation 46
Understanding Interactive Binning 51
Performing Interactive Binning 51

Fitting and Comparing Candidate Models **56**
Fitting a Regression Model 56
Evaluating the Models 58
Fitting a Default Decision Tree 59
Exploring the Tree Model Results 60
Fitting a Default Neural Network 61
Investigating the Regression and Neural Network Models 63

Generating and Using Scoring Code **64**
Overview 64
Scoring Using Base SAS 65
Scoring with SAS Enterprise Miner 67

Generating a Report Using the Reporter Node **68**

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.

Name	**Model Role**	**Measurement Level**	**Description**
BAD	Target	Binary	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.
CLAGE	Input	Interval	Age of the oldest credit line, measured in months
CLNO	Input	Interval	Number of credit lines
DEBTINC	Input	Interval	Debt-to-income ratio
DELINQ	Input	Interval	Number of delinquent credit lines
DEROG	Input	Interval	Number of major derogatory reports
JOB	Input	Nominal	Occupational categories (six categories).

Name	Model Role	Measurement Level	Description
LOAN	Input	Interval	Amount of the loan request
MORTDUE	Input	Interval	Amount due on the existing mortgage
NINQ	Input	Interval	Number of recent credit inquiries
REASON	Input	Binary	DebtCon=debt consolidation loan. HomeImp=home improvement loan.
VALUE	Input	Interval	Value of the current property
YOJ	Input	Interval	Years at the applicant's current job

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.

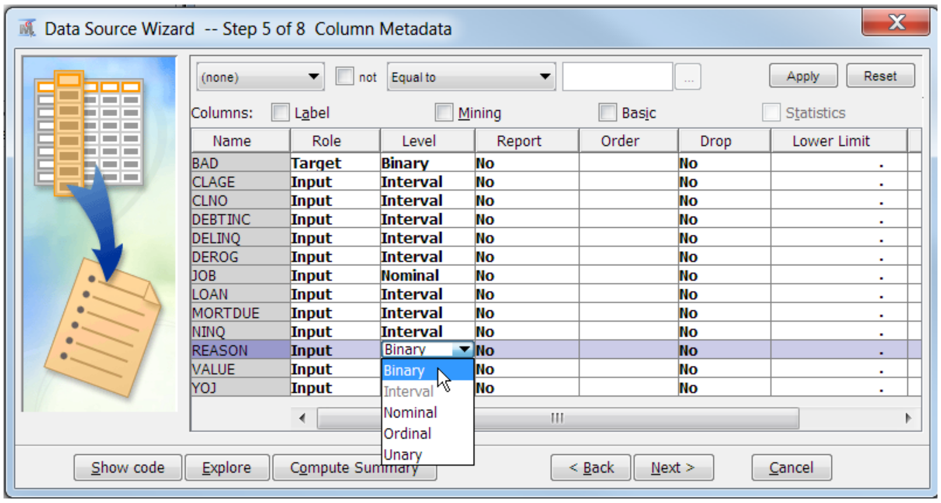
Creating a Process Flow Diagram

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 — Ensure that the **Level** is set to **Interval**.
   - DEROG — Ensure that the **Level** is set to **Interval**.

- REASON — Set the **Level** to **Binary**.



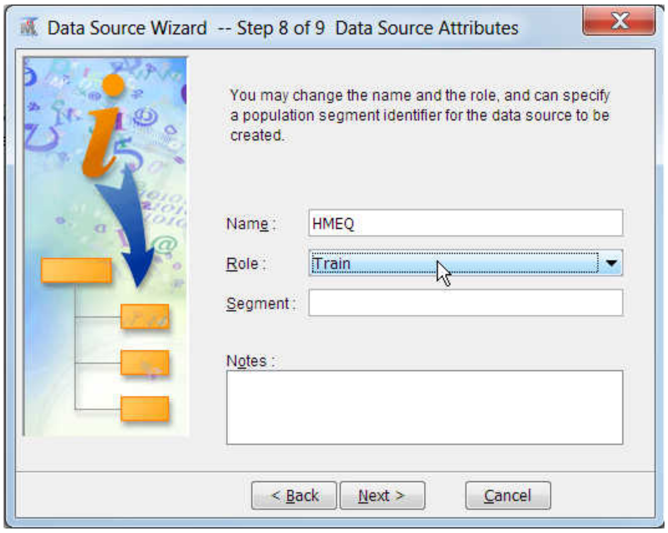
Name	Role	Level	Report	Order	Drop	Lower Limit
BAD	Target	Binary	No		No	.
CLAGE	Input	Interval	No		No	.
CLNO	Input	Interval	No		No	.
DEBTINC	Input	Interval	No		No	.
DELINQ	Input	Interval	No		No	.
DEROG	Input	Interval	No		No	.
JOB	Input	Nominal	No		No	.
LOAN	Input	Interval	No		No	.
MORTDUE	Input	Interval	No		No	.
NINQ	Input	Interval	No		No	.
REASON	Input	Binary	No		No	.
VALUE	Input		No		No	.
YOJ	Input		No		No	.

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. The HMEQ data source contains known values for the binary target value BAD, so we will use the HMEQ data to train our predictive model. Set the **Role** of the data source to **Train**. Click **Next**.



You may change the name and the role, and can specify a population segment identifier for the data source to be created.

Name : HMEQ

Role : Train

Segment :

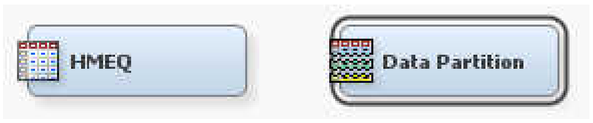
Notes :

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.

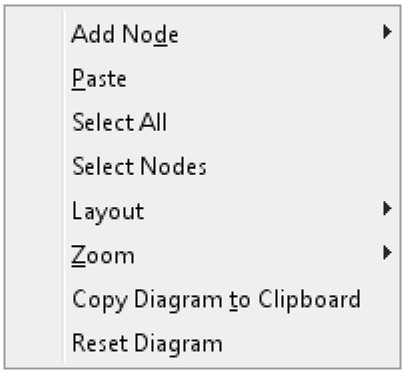
To add the input data, drag the HMEQ data set from the **Data Sources** section of the Project Panel to the diagram workspace. Because this is a predictive modeling process flow diagram, you will add a Data Partition node. From the **Sample** tab of the toolbar, drag a **Data Partition** node to the 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

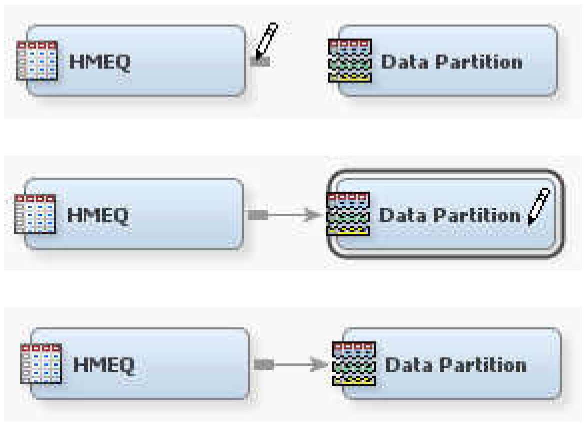
The cursor shape changes depending on where it is positioned. The behavior of the mouse commands depends on the cursor shape in addition to the selection state of the node over which the cursor is positioned. Right-click in an open area to see the pop-up menu shown below.



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.

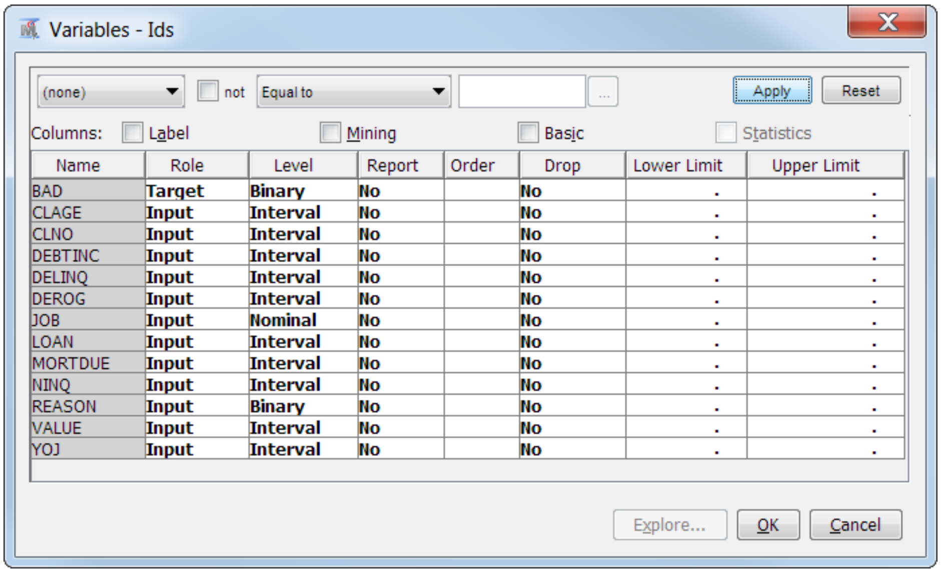


Connect your **HMEQ** node to the **Data Partition** node.

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.



Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
BAD	Target	Binary	No		No	.	.
CLAGE	Input	Interval	No		No	.	.
CLNO	Input	Interval	No		No	.	.
DEBTINC	Input	Interval	No		No	.	.
DELINQ	Input	Interval	No		No	.	.
DEROG	Input	Interval	No		No	.	.
JOB	Input	Nominal	No		No	.	.
LOAN	Input	Interval	No		No	.	.
MORTDUE	Input	Interval	No		No	.	.
NINQ	Input	Interval	No		No	.	.
REASON	Input	Binary	No		No	.	.
VALUE	Input	Interval	No		No	.	.
YOJ	Input	Interval	No		No	.	.

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 CLAGE, CLNO, DEBTINC, DELINQ, and DEROG are assigned the **Level** of **Interval**.

The variable JOB is a character variable. We will model JOB as a **Nominal** variable because it is a character variable with more than two distinct, nonmissing values.

The variables LOAN, MORTDUE, and NINQ are assigned the **Level** of **Interval**.

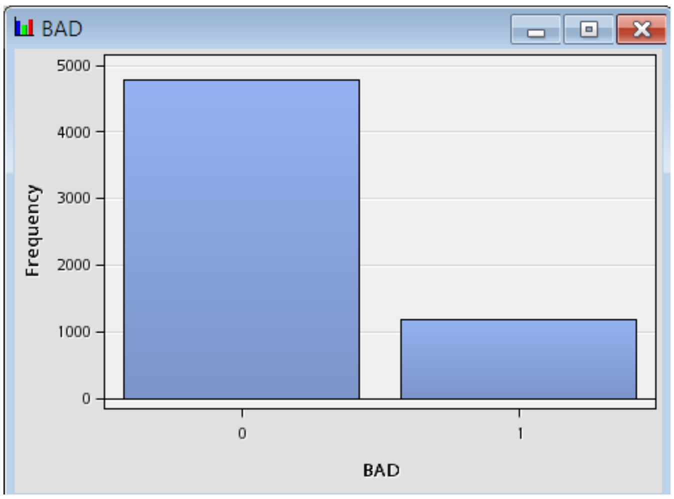
The variable REASON is a character variable. Unlike the character variable JOB, the character variable REASON has only two distinct, nonmissing values. REASON is best modeled as with a **Level** of **Binary**.

The variables VALUE and YOJ are assigned the **Level** of **Interval**for the purposes of this analysis.

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.



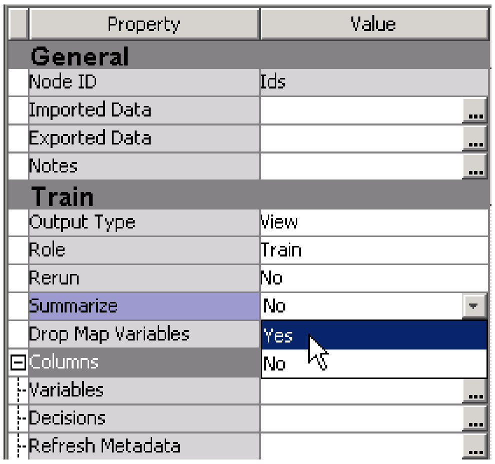
On the bar chart, you can see that approximately 80% of the observations in BAD have a value of 0 and 20% have a value of 1. This means that approximately 20% of the customers in this sample data set defaulted on their loans.

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:

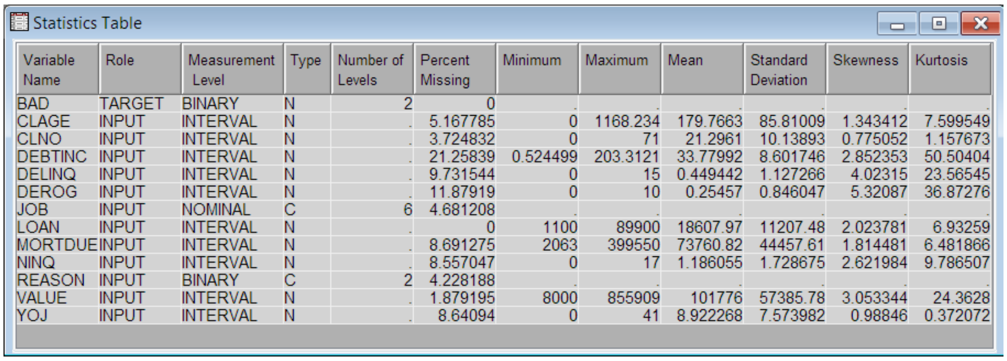
1. In your project workspace, select the **HMEQ** data source.
2. In the Properties panel, set the value of the **Summarize** property to **Yes**.



Property	Value
General	
Node ID	Ids
Imported Data	...
Exported Data	...
Notes	...
Train	
Output Type	View
Role	Train
Rerun	No
Summarize	No
Drop Map Variables	Yes / No
⊟ Columns	
Variables	...
Decisions	...
Refresh Metadata	...

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.



Statistics Table

Variable Name	Role	Measurement Level	Type	Number of Levels	Percent Missing	Minimum	Maximum	Mean	Standard Deviation	Skewness	Kurtosis
BAD	TARGET	BINARY	N	2	0	.	.	.	.	.	.
CLAGE	INPUT	INTERVAL	N	.	5.167785	0	1168.234	179.7663	85.81009	1.343412	7.599549
CLNO	INPUT	INTERVAL	N	.	3.724832	0	71	21.2961	10.13893	0.775052	1.157673
DEBTINC	INPUT	INTERVAL	N	.	21.25839	0.524499	203.3121	33.77992	8.601746	2.852353	50.50404
DELINQ	INPUT	INTERVAL	N	.	9.731544	0	15	0.449442	1.127266	4.02315	23.56545
DEROG	INPUT	INTERVAL	N	.	11.87919	0	10	0.25457	0.846047	5.32087	36.87276
JOB	INPUT	NOMINAL	C	6	4.681208	.	.	.	.	.	.
LOAN	INPUT	INTERVAL	N	.	0	1100	89900	18607.97	11207.48	2.023781	6.93259
MORTDUE	INPUT	INTERVAL	N	.	8.691275	2063	399550	73760.82	44457.61	1.814481	6.481866
NINQ	INPUT	INTERVAL	N	.	8.557047	0	17	1.186055	1.728675	2.621984	9.786507
REASON	INPUT	BINARY	C	2	4.228188	.	.	.	.	.	.
VALUE	INPUT	INTERVAL	N	.	1.879195	8000	855909	101776	57385.78	3.053344	24.3628
YOJ	INPUT	INTERVAL	N	.	8.64094	0	41	8.922268	7.573982	0.98846	0.372072

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 values in the minimum or maximum columns. Notice that DEBTINC indicates a high percentage (21.25%) of missing values.

5. Close the Results window.

Inspecting the Default Settings of the Data Partition Node

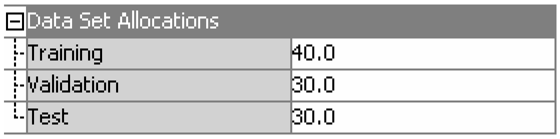
In your diagram workspace, select the **Data Partition** node. The default **Partitioning Method** for the **Data Partition** node is stratified because BAD is a class variable. The available methods for partitioning are as follows:

- 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.

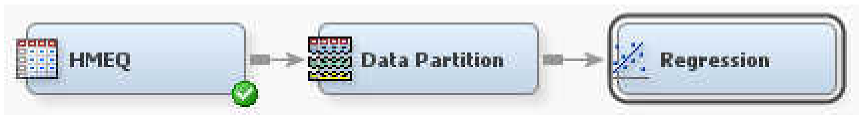


Data Set Allocations	
Training	40.0
Validation	30.0
Test	30.0

For this example, use the default settings.

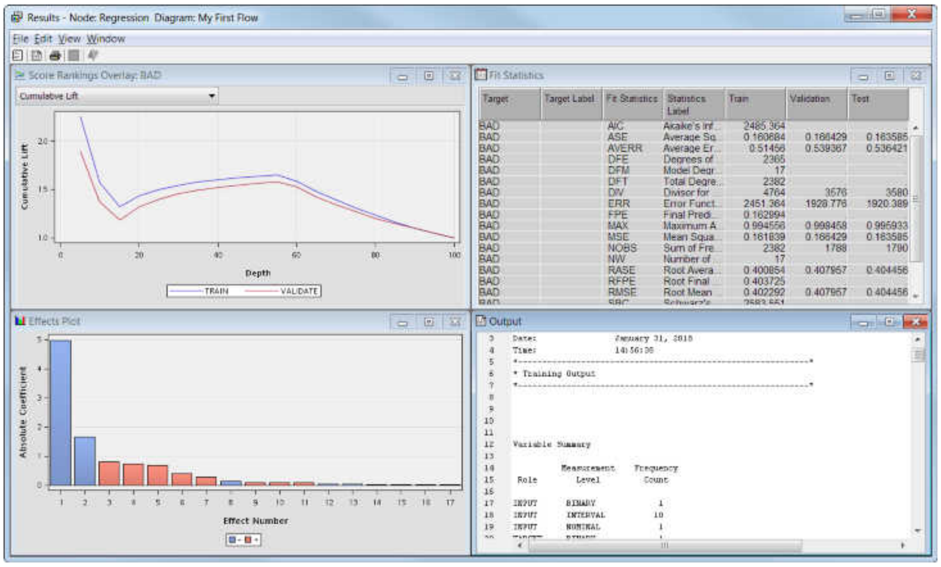
Fitting and Evaluating a Regression Model

Now that you have configured 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. You can verify the Regression configuration by selecting the node in the workspace, and then examining the Property panel settings. In the Equation group, inspect the Main Effects property. In the Class Targets group, inspect the Regression Type property. By default, the node codes your grouping variables with deviation coding. Under Model Options, verify the Input Coding property setting.

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.

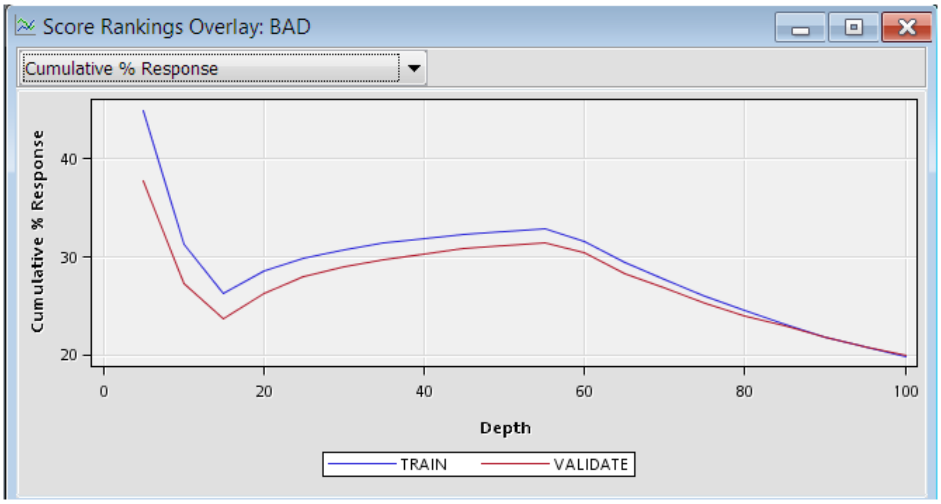


Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
BAD		AIC	Akaike's Inf...	2485.364		
BAD		ASE	Average Sq...	0.160684	0.166429	0.163585
BAD		AVERR	Average Er...	0.51456	0.539367	0.536421
BAD		DFE	Degrees of ...	2365		
BAD		DFM	Model Degr...	17		
BAD		DFT	Total Degre...	2382		
BAD		DIV	Divisor for ...	4764	3576	3580
BAD		ERR	Error Funct...	2451.364	1928.776	1920.389
BAD		FPE	Final Predi...	0.162994		
BAD		MAX	Maximum A...	0.994556	0.998458	0.995933
BAD		MSE	Mean Squa...	0.161639	0.166429	0.163585
BAD		NOBS	Sum of Fre...	2382	1788	1790
BAD		NW	Number of ...	17		
BAD		RASE	Root Avera...	0.400854	0.407957	0.404456
BAD		RFPE	Root Final ...	0.403725		
BAD		RMSE	Root Mean ...	0.402292	0.407957	0.404456
BAD		SBC	Schwarz's ...	2583.551		

```
Date:                January 31, 2018
Time:                14:56:36
*------------------------------------------------------------*
* Training Output
*------------------------------------------------------------*



Variable Summary

          Measurement    Frequency
 Role        Level         Count

INPUT       BINARY           1
INPUT       INTERVAL        10
INPUT       NOMINAL          1
```

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.

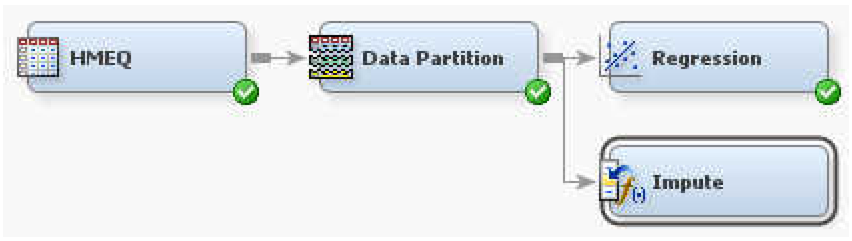
Recall that the variable DEBTINC has a high percentage of missing values. Because of this, it is not appropriate to apply a default regression model directly to the training data. Regression models ignore observations that have a missing value for at least one input variable. Therefore, you should consider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

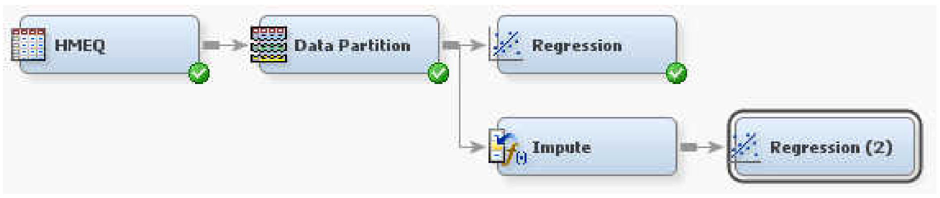
Select the **Impute** node and find the **Indicator Variables** property subgroup in the properties panel. If you want to create a single indicator variable that indicates whether any variable was imputed for each observation, set **Type** to **Single**. If you want to create an indicator variable for each original variable that specifies if imputation was performed, set **Type** to **Unique**.

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

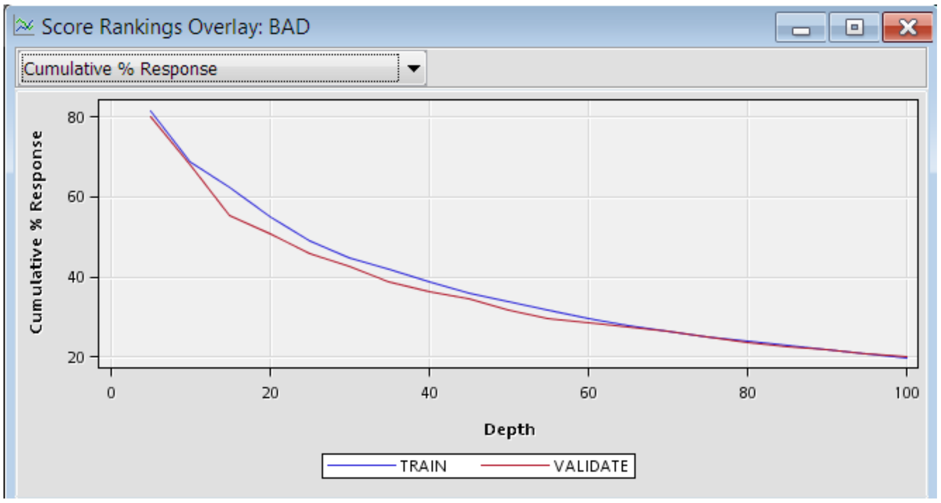
An indicator variable is a special variable that is created automatically by SAS Enterprise Miner. The value of this variable is 1 when an observation contained a missing value and 0 otherwise. The Regression and Neural Network nodes can use the indicator variables to identify observations that had missing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



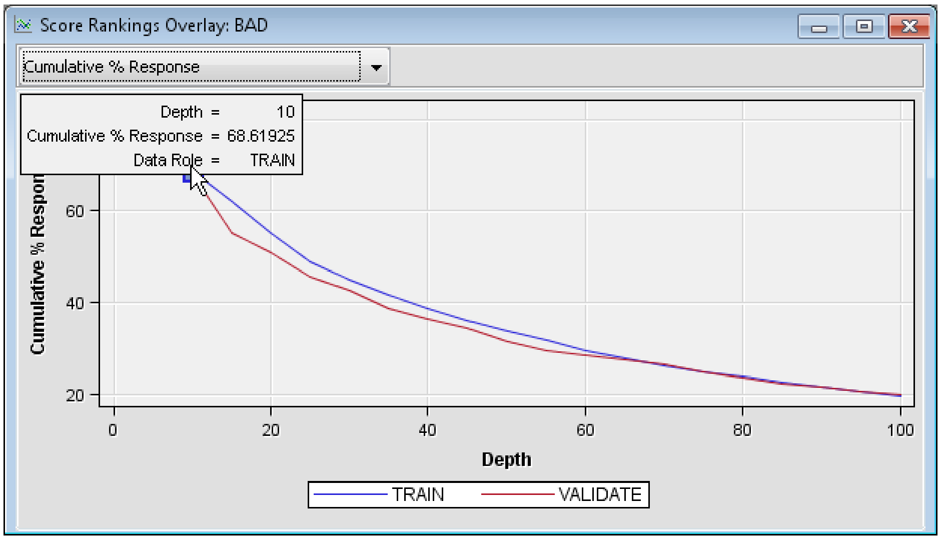
Right-click the **Regression (2)** node and click **Run**. In the Confirmation window, click **Yes**. After the node has successfully run, click **Results** in the Run Status window.



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.

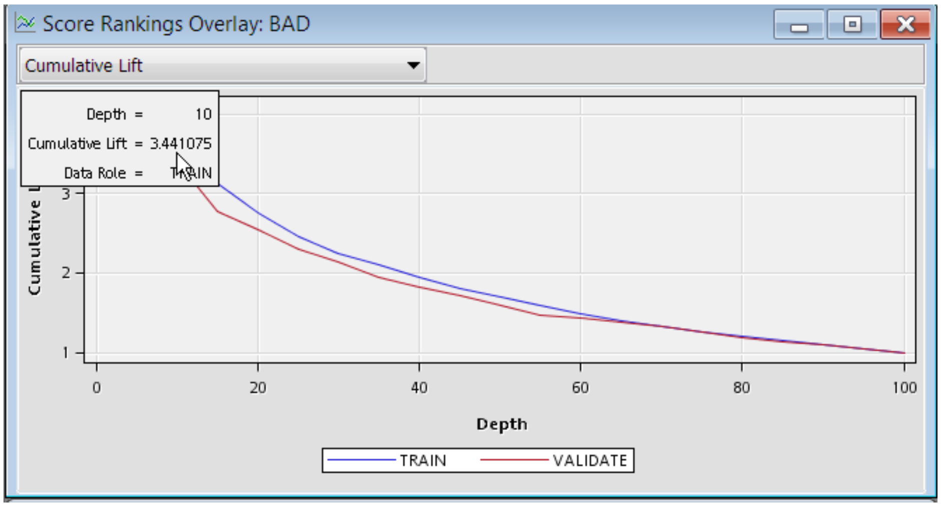
The discussion of the remaining charts refers to those who defaulted on a loan as defaults or respondents. This is because the target level of interest is BAD=1.

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.

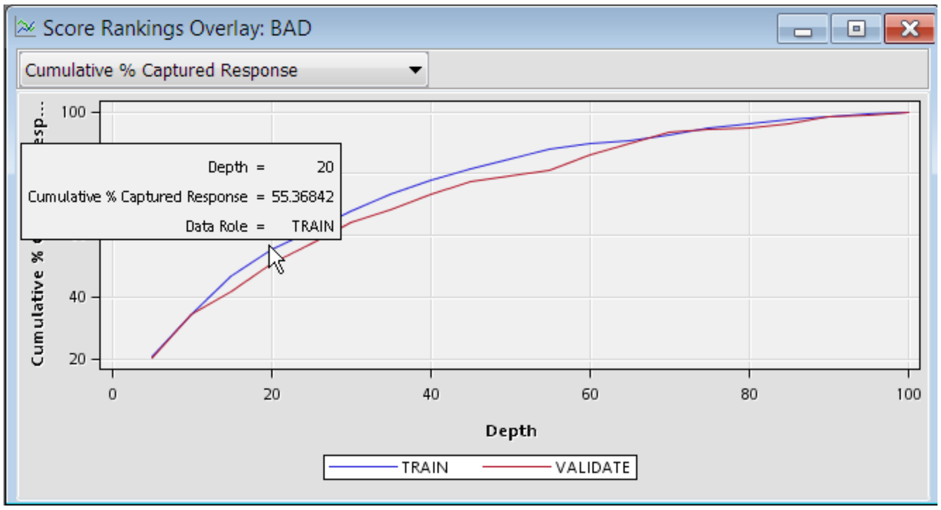
On the drop-down menu, click **Cumulative Lift**.



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 =

3.44. 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.


Score Rankings Overlay: BAD
Cumulative % Captured Response
Cumulative % Captured Resp...
Depth
TRAIN
VALIDATE

Depth = 20
Cumulative % Captured Response = 55.36842
Data Role = TRAIN

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. The Effects Plot contains values for variables with imputed values. Imputed variables are identified by the prefix, “IMP_”. Note that in this model, the most important variables in the Effects Plot are DELINQ (IMP_DELINQ), JOB (IMP_JOBOFFICE, IMP_JOBSALES, IMP_JOBPROFEXE, IMP_JOBMGR), DEROG (IMP_DEROG), NINQ (IMP_NINQ), and REASON (IMP_REASONDEBTCON).

Data Preparation and Investigation

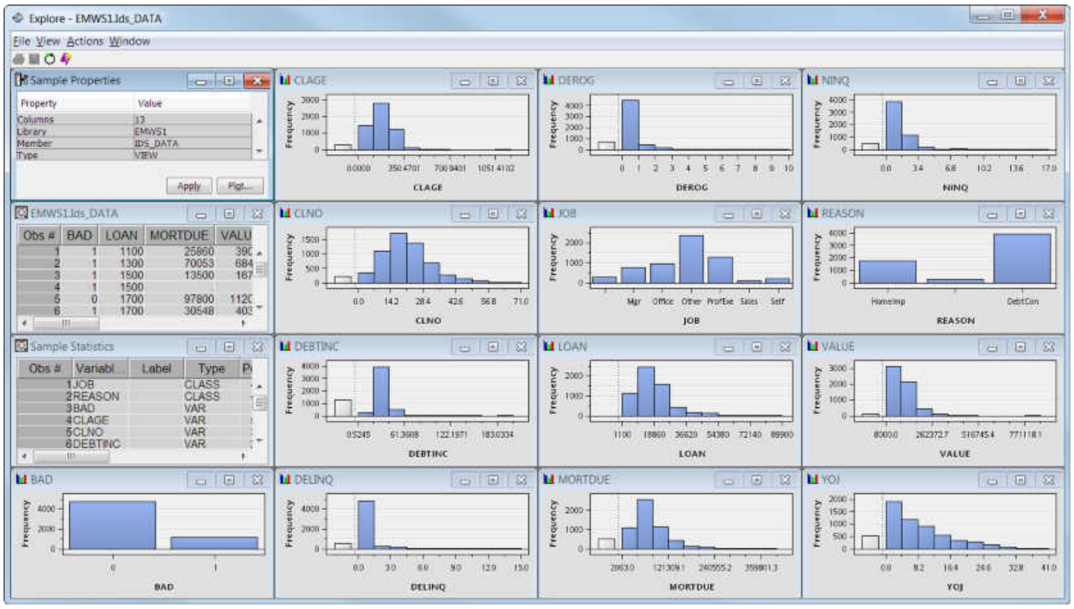
Preliminary Investigation

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.



Property	Value
Columns	13
Library	EMWS1
Member	IDS_DATA
Type	VIEW

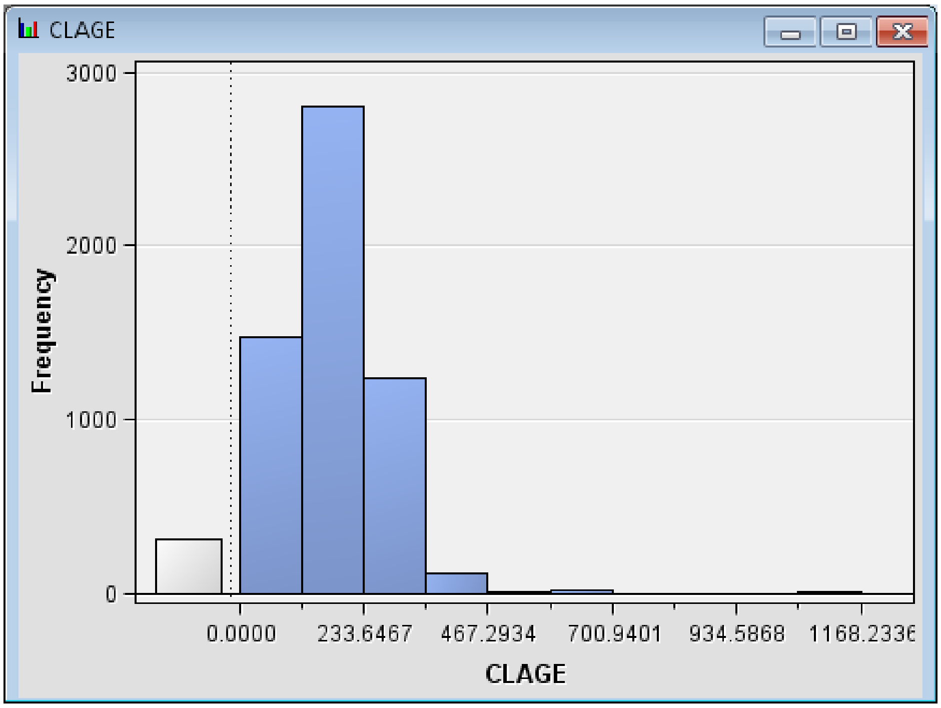
Obs #	BAD	LOAN	MORTDUE	VALU
1	1	1100	25860	390
2	1	1300	70053	684
3	1	1500	13500	167
4	1	1500		
5	0	1700	97800	112C
6	1	1700	30548	403

Obs #	Variabl	Label	Type	P
1	JOB		CLASS	
2	REASON		CLASS	
3	BAD		VAR	
4	CLAGE		VAR	
5	CLNO		VAR	
6	DEBTINC		VAR	

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: 5,960. The SAMPSIO.HMEQ data set is small enough that we declined to create a sample in the Input Data Wizard, and we chose to use the entire data set. Close the Sample Properties window.

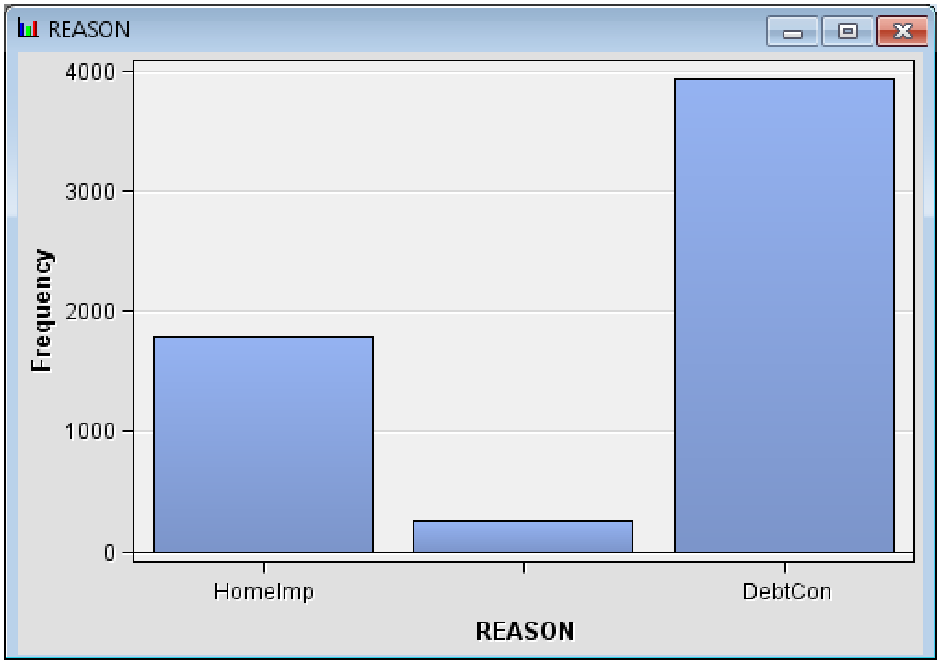
Maximize the Sample Statistics window. This window contains the calculated mean, maximum, and minimum for the interval variables and the number of class levels, modal value, and percentage of observations in the modal value for class variables. The percentage of missing values is calculated for every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

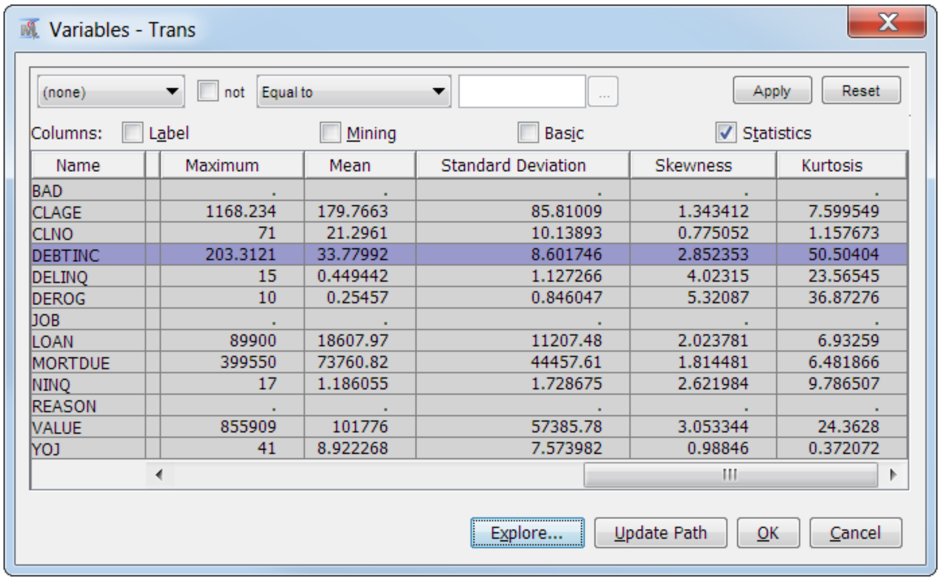
Performing Variable Transformation

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-Trans window appears.

In the Variables-Trans 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.



Columns: ☐ Label ☐ Mining ☐ Basic ☑ Statistics

Name	Maximum	Mean	Standard Deviation	Skewness	Kurtosis
BAD	.	.	.	.	.
CLAGE	1168.234	179.7663	85.81009	1.343412	7.599549
CLNO	71	21.2961	10.13893	0.775052	1.157673
DEBTINC	203.3121	33.77992	8.601746	2.852353	50.50404
DELINQ	15	0.449442	1.127266	4.02315	23.56545
DEROG	10	0.25457	0.846047	5.32087	36.87276
JOB	.	.	.	.	.
LOAN	89900	18607.97	11207.48	2.023781	6.93259
MORTDUE	399550	73760.82	44457.61	1.814481	6.481866
NINQ	17	1.186055	1.728675	2.621984	9.786507
REASON	.	.	.	.	.
VALUE	855909	101776	57385.78	3.053344	24.3628
YOJ	41	8.922268	7.573982	0.98846	0.372072

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. Click the **Skewness** column heading until the table values are sorted by ascending skewness. 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.

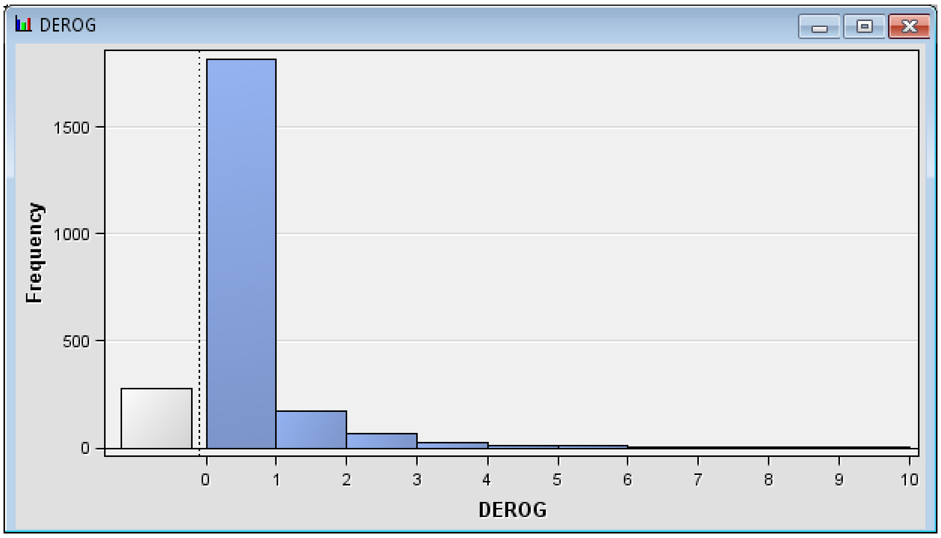
This example applies a log transformation to all of the input variables. The log transformation creates a new variable by taking the natural log of each original input variable. In your diagram workspace, select the **Transform Variables** node. In the Default Methods group, set the value of the **Interval 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**.

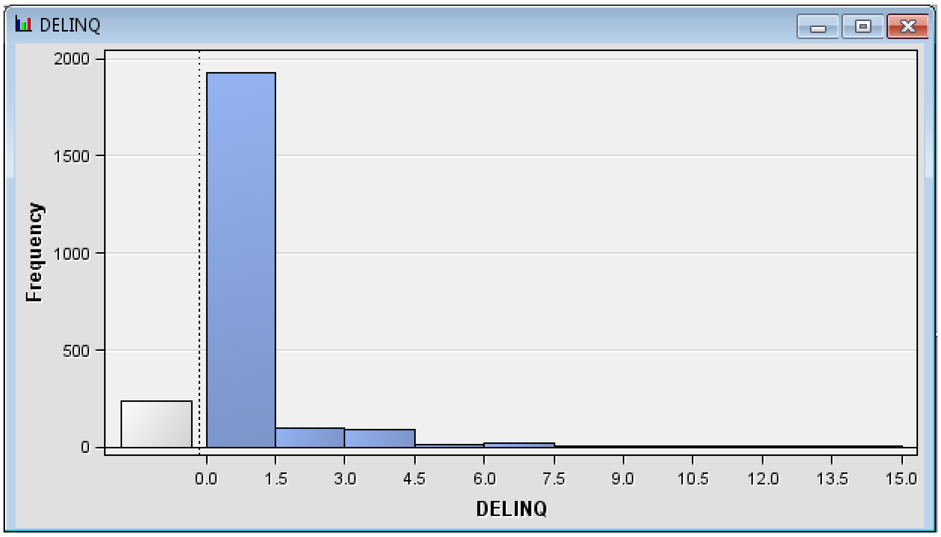
Maximize the Transformations Statistics window. This window provides statistics for the original and transformed variables. The **Formula** column indicates the expression used to transform each variable. Notice that the absolute value of the **Skewness** statistic for the transformed values is typically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



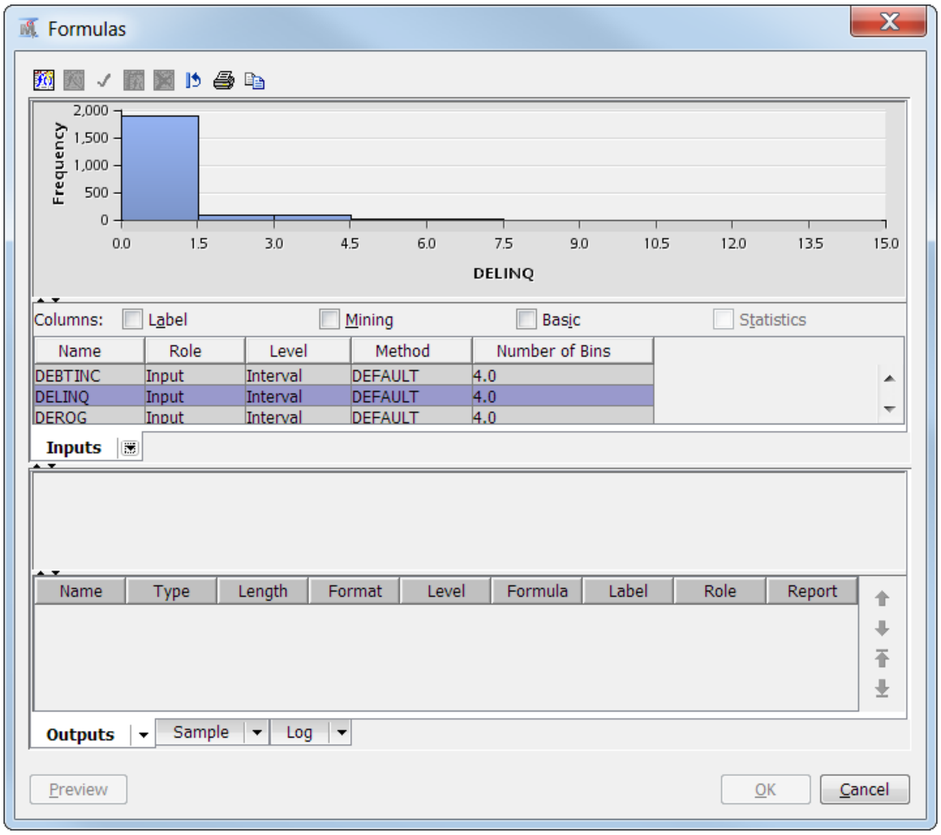
Repeat this process for the **DELINQ** variable. Nearly all of the values for DELINQ are equal to 0. The next largest class is the missing values.



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.

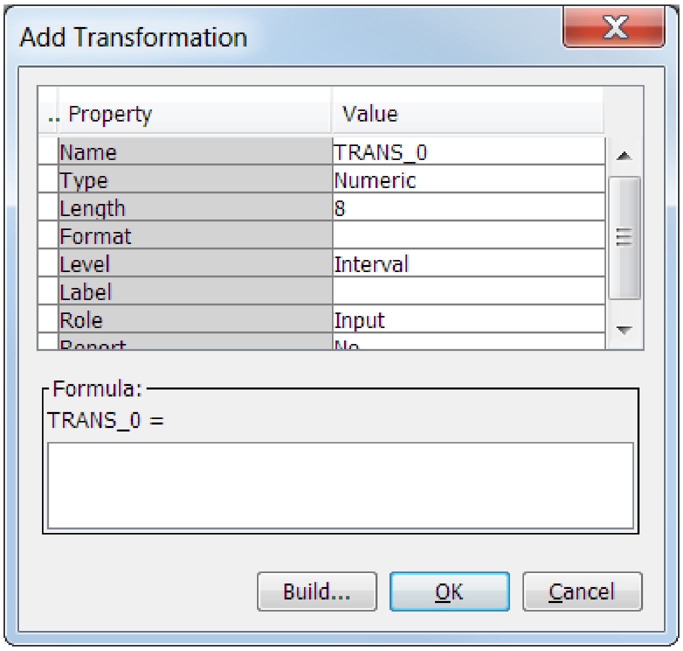
In your process flow diagram, select the **Transform Variables** node. Click ... next to the **Formulas** property. The Formulas window appears.



Name	Role	Level	Method	Number of Bins
DEBTINC	Input	Interval	DEFAULT	4.0
DELINQ	Input	Interval	DEFAULT	4.0
DEROG	Input	Interval	DEFAULT	4.0

Name	Type	Length	Format	Level	Formula	Label	Role	Report

The Formulas window enables you to create custom variable transformations. Select the **DELINQ** variable and click the **Create** tool in the upper left corner of the Formulas window. The Add Transformation window appears.



Property	Value
Name	TRANS_0
Type	Numeric
Length	8
Format	
Level	Interval
Label	
Role	Input

Formula:
TRANS_0 =

Complete the following steps to transform the DELINQ variable:

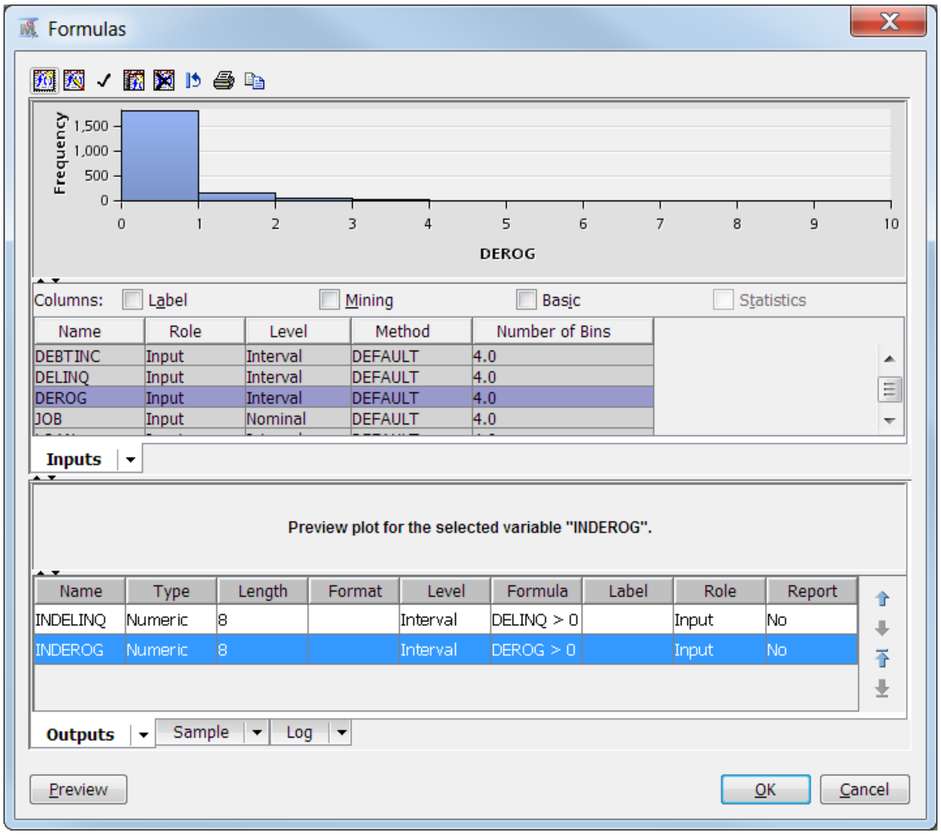
1. Enter **`INDELINQ`** for the **Name** property. (You need to select and over-write the default TRANS0 value.) The default values are acceptable for the other Add Transformation properties.

2. In the **Formula** dialog box at the bottom of the window, enter **`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`**.

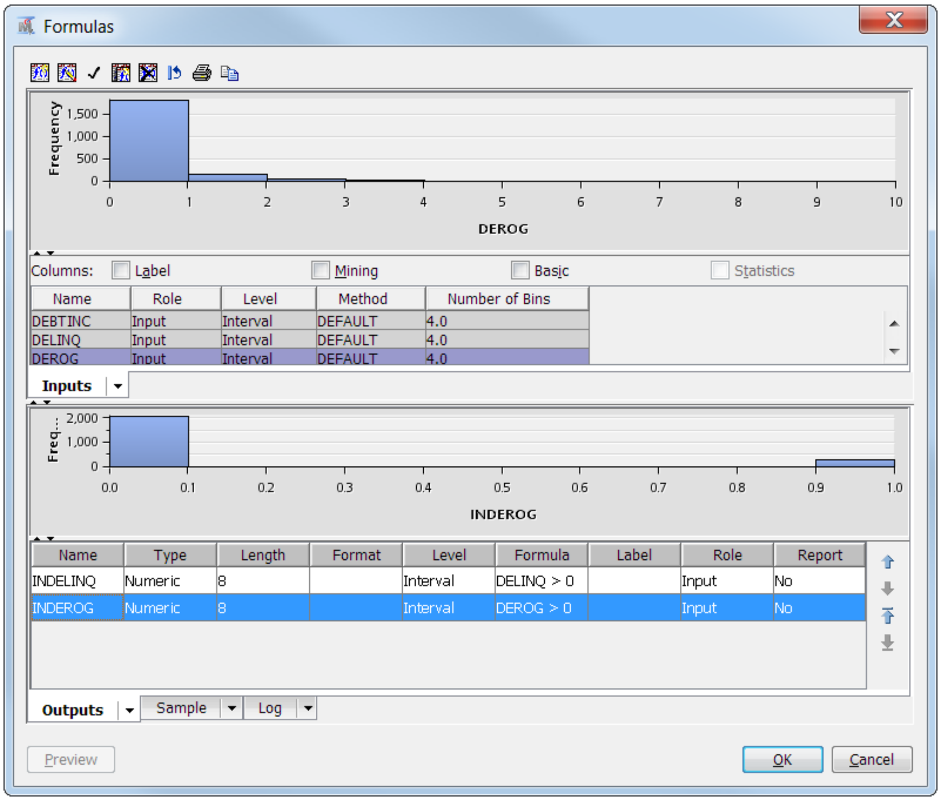


Name	Role	Level	Method	Number of Bins
DEBTINC	Input	Interval	DEFAULT	4.0
DELINQ	Input	Interval	DEFAULT	4.0
DEROG	Input	Interval	DEFAULT	4.0
JOB	Input	Nominal	DEFAULT	4.0

Preview plot for the selected variable "INDEROG".

Name	Type	Length	Format	Level	Formula	Label	Role	Report
INDELINQ	Numeric	8		Interval	DELINQ > 0		Input	No
INDEROG	Numeric	8		Interval	DEROG > 0		Input	No

5. Click **Preview** in the lower left corner of the screen.



Name	Role	Level	Method	Number of Bins
DEBTINC	Input	Interval	DEFAULT	4.0
DELINQ	Input	Interval	DEFAULT	4.0
DEROG	Input	Interval	DEFAULT	4.0

Name	Type	Length	Format	Level	Formula	Label	Role	Report
INDELINQ	Numeric	8		Interval	DELINQ > 0		Input	No
INDEROG	Numeric	8		Interval	DEROG > 0		Input	No

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.

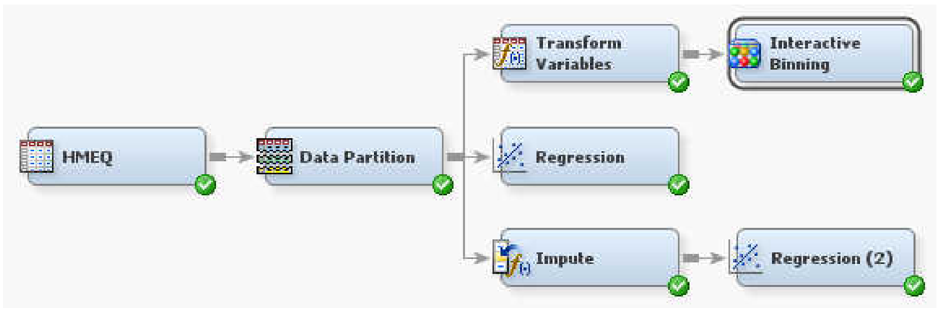
Performing Interactive Binning

Recall the distribution of NINQ, the number of recent credit inquiries associated with an individual. 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 “2+”. This would create a new three-level grouping variable from NINQ. It is true that creating a grouping variable that condenses multiple levels into a combined level will result in some loss of information about the exact number of recent credit inquiries. On the other hand, the small change does enable you to handle nonlinearity in the relationship between NINQ and the response variable. The trade-off is worth exploring.

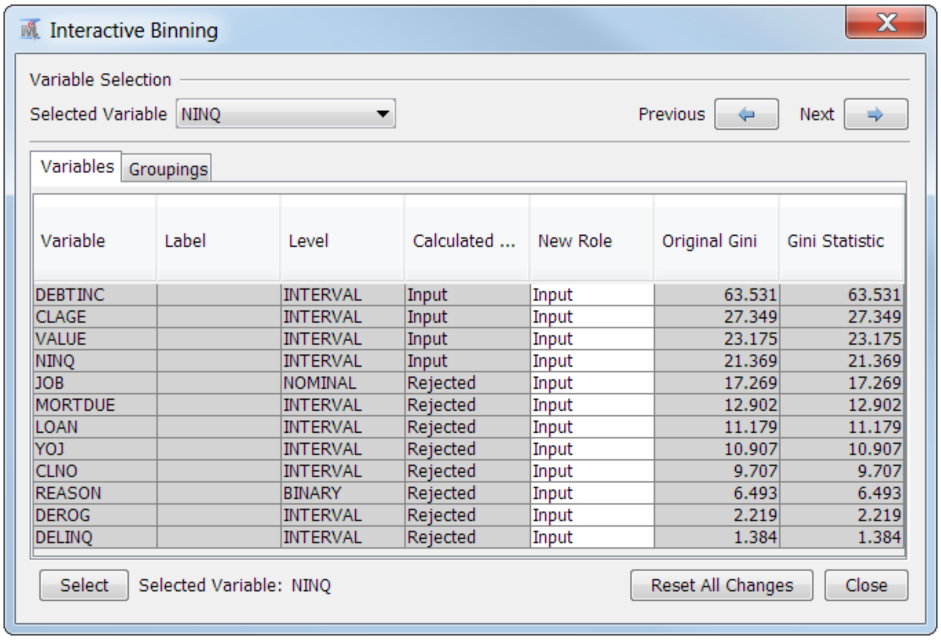
In order to bin NINQ, you need to add an **Interactive Binning** node to your process flow diagram. However, because you plan to bin NINQ, you do not want to transform it with the **Transform Variables** node. Right-click the **Transform Variables** node and click **Edit Variables**. In the Variables-Trans table, set the **Method** for the variable NINQ from to **Default** to **None**, and then click **OK**. The **Transform Variables** node will not transform NINQ.

From 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**.



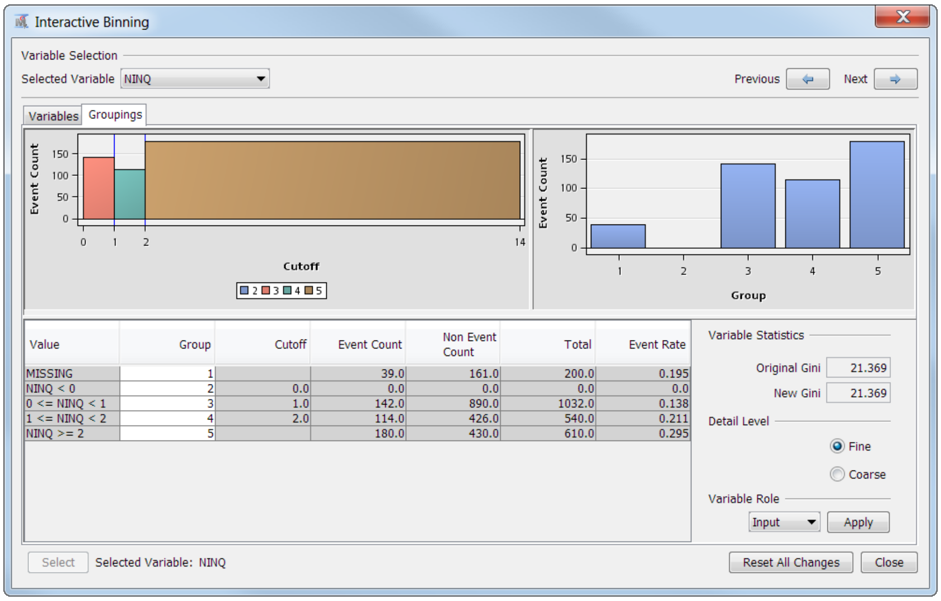
In your diagram workspace, select the **Interactive Binning** node. Next, click ... next to the **Interactive Binning** property in the properties panel.



Variable	Label	Level	Calculated ...	New Role	Original Gini	Gini Statistic
DEBTINC		INTERVAL	Input	Input	63.531	63.531
CLAGE		INTERVAL	Input	Input	27.349	27.349
VALUE		INTERVAL	Input	Input	23.175	23.175
NINQ		INTERVAL	Input	Input	21.369	21.369
JOB		NOMINAL	Rejected	Input	17.269	17.269
MORTDUE		INTERVAL	Rejected	Input	12.902	12.902
LOAN		INTERVAL	Rejected	Input	11.179	11.179
YOJ		INTERVAL	Rejected	Input	10.907	10.907
CLNO		INTERVAL	Rejected	Input	9.707	9.707
REASON		BINARY	Rejected	Input	6.493	6.493
DEROG		INTERVAL	Rejected	Input	2.219	2.219
DELINQ		INTERVAL	Rejected	Input	1.384	1.384

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, click the **Groupings** tab to display individual variable bins. Use the **Selected Variable** drop-down menu in the upper left corner of the window to select the variable **NINQ**.



Selected Variable: NINQ

Value	Group	Cutoff	Event Count	Non Event Count	Total	Event Rate
MISSING	1		39.0	161.0	200.0	0.195
NINQ < 0	2	0.0	0.0	0.0	0.0	0.0
0 <= NINQ < 1	3	1.0	142.0	890.0	1032.0	0.138
1 <= NINQ < 2	4	2.0	114.0	426.0	540.0	0.211
NINQ >= 2	5		180.0	430.0	610.0	0.295

Variable Statistics

Original Gini: 21.369

New Gini: 21.369

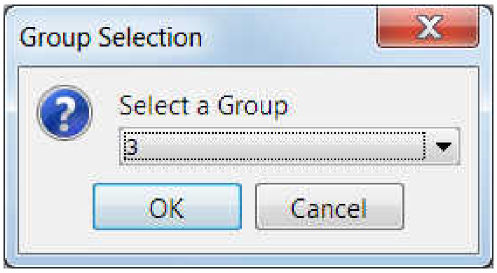
Detail Level: [✓ Fine] Coarse

Variable Role: Input

Selected Variable: NINQ

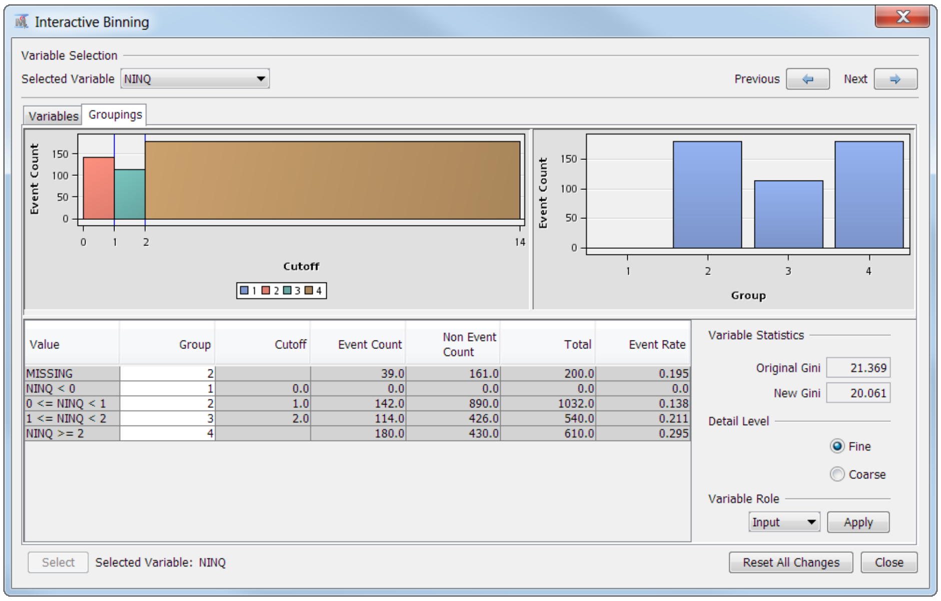
Automatically, the **Interactive Binning** node created five groups for NINQ. Notice, however, that the first two groups contain the missing values and negative values, respectively. Also notice that the group that contains the negative values is empty. For this example, you assume that a missing value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

In the table, select the **MISSING** group (Group 1) and right-click that row. Click **Assign To**. In the Group Selection window, select **3** for the **Select a Group** selection.



Select a Group: 3

Notice that the groups were renumbered.



Selected Variable: NINQ

Value	Group	Cutoff	Event Count	Non Event Count	Total	Event Rate
MISSING	2		39.0	161.0	200.0	0.195
NINQ < 0	1	0.0	0.0	0.0	0.0	0.0
0 <= NINQ < 1	2	1.0	142.0	890.0	1032.0	0.138
1 <= NINQ < 2	3	2.0	114.0	426.0	540.0	0.211
NINQ >= 2	4		180.0	430.0	610.0	0.295

Variable Statistics

Original Gini 21.369

New Gini 20.061

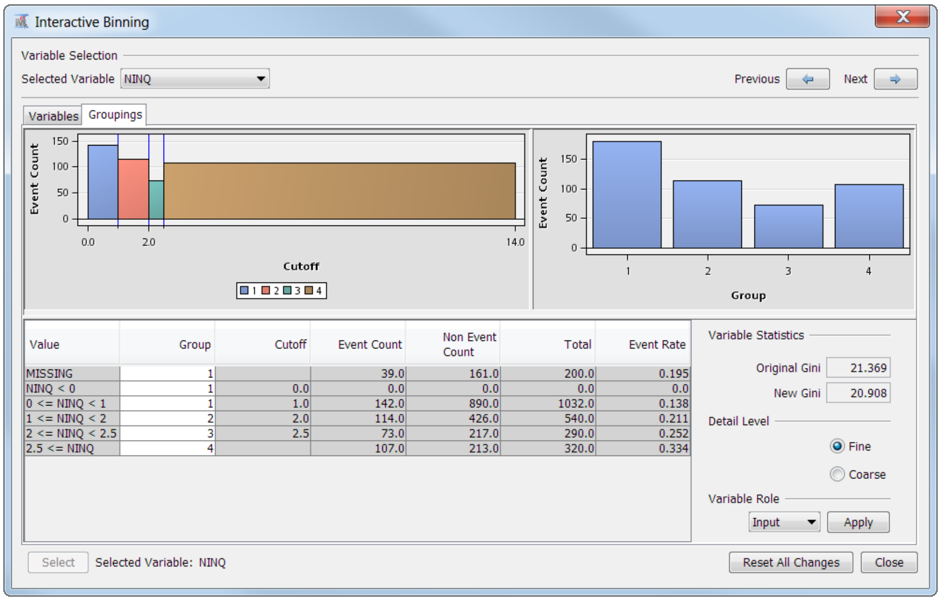
Detail Level: [✓ Fine] Coarse

Variable Role: Input Apply

Selected Variable: NINQ

Next, repeat this process for the empty bin that contains the negative values. In the table, select the group **NINQ < 0** (Group 1) 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**.



Selected Variable: NINQ

Value	Group	Cutoff	Event Count	Non Event Count	Total	Event Rate
MISSING	1		39.0	161.0	200.0	0.195
NINQ < 0	1	0.0	0.0	0.0	0.0	0.0
0 <= NINQ < 1	1	1.0	142.0	890.0	1032.0	0.138
1 <= NINQ < 2	2	2.0	114.0	426.0	540.0	0.211
2 <= NINQ < 2.5	3	2.5	73.0	217.0	290.0	0.252
2.5 <= NINQ	4		107.0	213.0	320.0	0.334

Variable Statistics

Original Gini 21.369

New Gini 20.908

Detail Level: [✓ Fine] Coarse

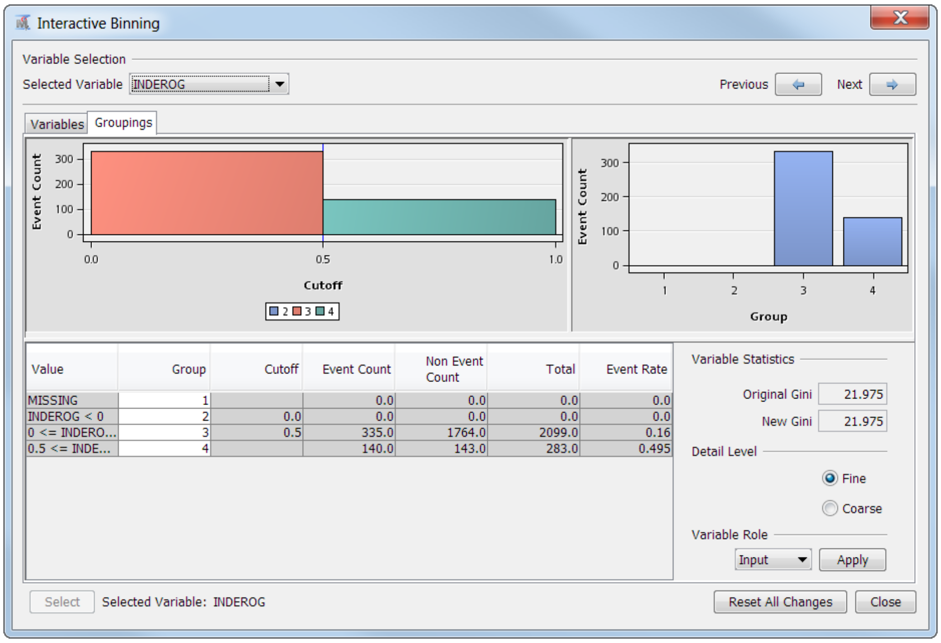
Variable Role: Input Apply

Selected Variable: NINQ

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.

After you make your changes to the bins for NINQ, click the **Apply** button at lower right to save your **NINQ** changes.

On the **Select Variable** drop-down menu at upper left, 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**.



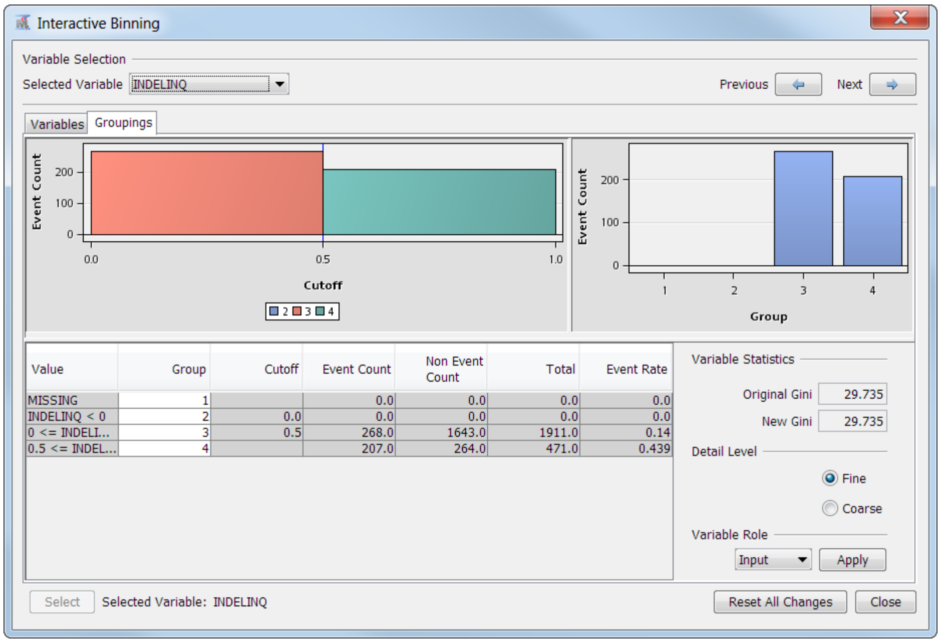
Value	Group	Cutoff	Event Count	Non Event Count	Total	Event Rate
MISSING	1		0.0	0.0	0.0	0.0
INDEROG < 0	2	0.0	0.0	0.0	0.0	0.0
0 <= INDERO...	3	0.5	335.0	1764.0	2099.0	0.16
0.5 <= INDE...	4		140.0	143.0	283.0	0.495

Original Gini 21.975
New Gini 21.975
Detail Level: [✓ Fine] Coarse
Variable Role: Input
Selected Variable: INDEROG

After you make your changes to the bins for **INDEROG**, click the **Apply** button at lower right to save your **INDEROG** changes.

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.



Selected Variable: INDELINQ

Value	Group	Cutoff	Event Count	Non Event Count	Total	Event Rate
MISSING	1		0.0	0.0	0.0	0.0
INDELINQ < 0	2	0.0	0.0	0.0	0.0	0.0
0 <= INDELI...	3	0.5	268.0	1643.0	1911.0	0.14
0.5 <= INDEL...	4		207.0	264.0	471.0	0.439

Variable Statistics
Original Gini 29.735
New Gini 29.735

Detail Level: [✓ Fine] Coarse

Variable Role: Input

After you make your changes to the bins for **INDELINQ**, click the **Apply** button at lower right to save your **INDELINQ** changes.

Click the **Variables** tab to verify that all 12 variables in the **New Role** column of the Interactive Binning variables table show **Input** status. If necessary, select all of the variables, right-click in the **New Role** column, and re-assign all of the variables to **Input** status.

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.

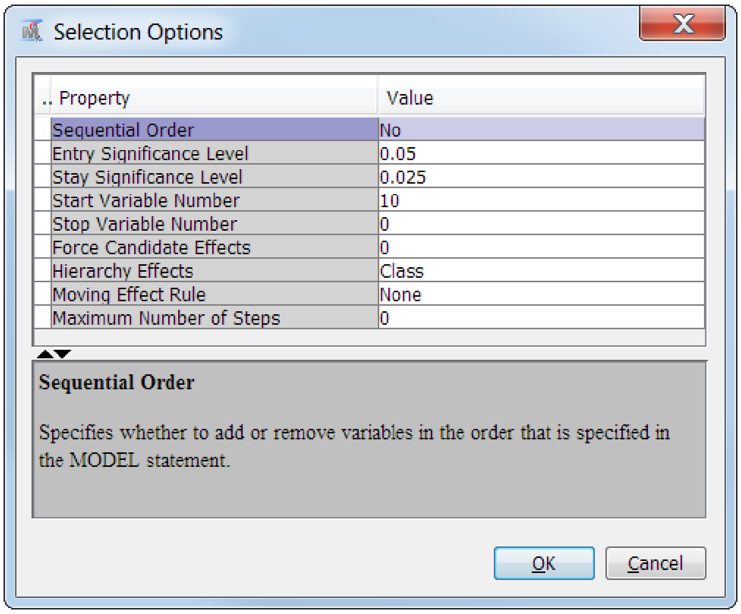
Click ... next to the **Selection Options** property. The Selection Options window contains the **Entry Significance Level**, **Stay Significance Level**, **Start Variable Number**, and **Stop Variable Number** properties that were just mentioned. These properties enable you to control the significance 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.



Property	Value
Sequential Order	No
Entry Significance Level	0.05
Stay Significance Level	0.025
Start Variable Number	10
Stop Variable Number	0
Force Candidate Effects	0
Hierarchy Effects	Class
Moving Effect Rule	None
Maximum Number of Steps	0

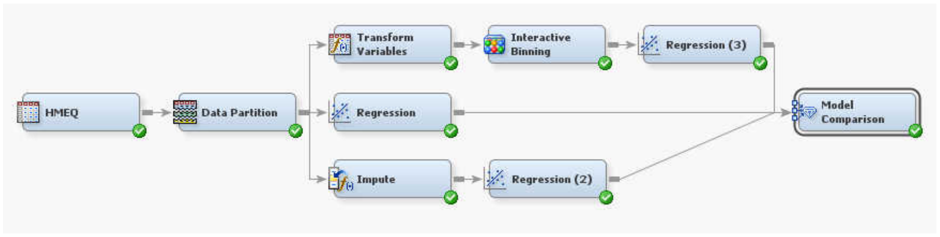
Sequential Order

Specifies whether to add or remove variables in the order that is specified in the MODEL statement.

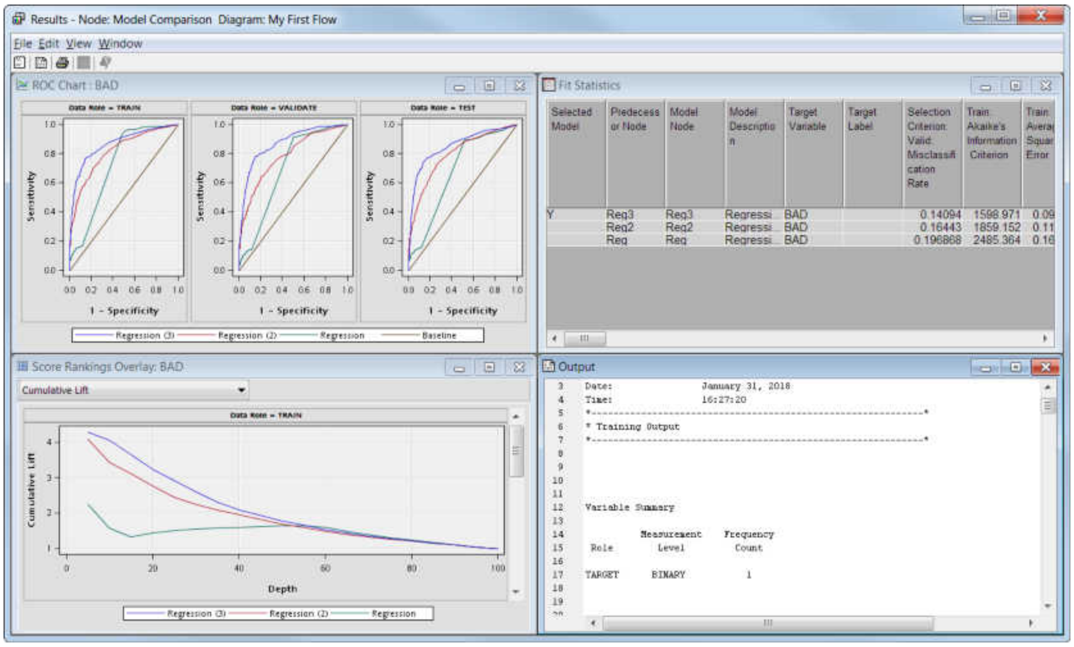
Click **OK**.

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.



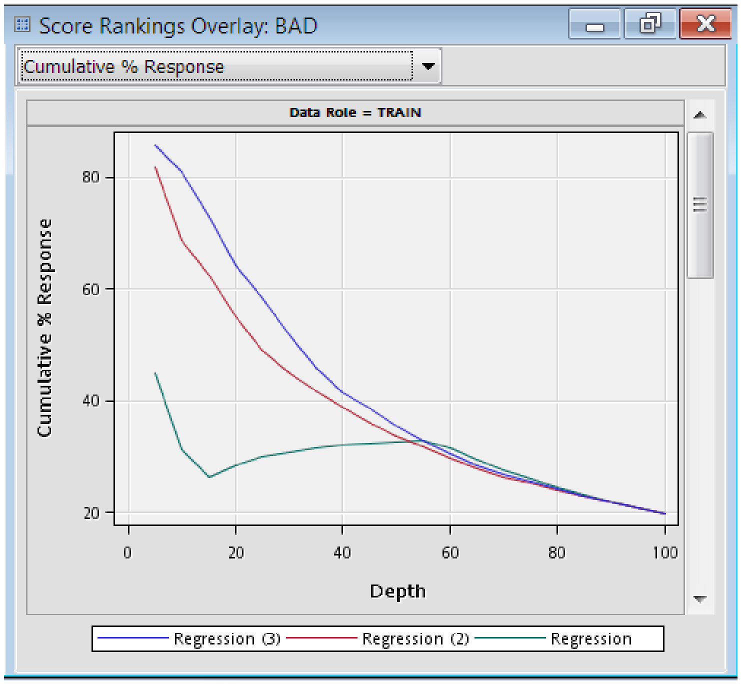
Select the **Model Comparison** node in the diagram,and in the Properties Panel, set the **Selection Statistic** property to **Misclassification Rate**, and set the **Selection Table** property to **Test**. Right-click the **Model Comparison** node, and then click **Run**. In the Confirmation window, click **Yes**. In the Run Status window click **Results**.



Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate	Train: Akaike's Information Criterion	Train: Average Squared Error
Y	Reg3	Reg3	Regressi...	BAD		0.14094	1598.971	0.09
	Reg2	Reg2	Regressi...	BAD		0.16443	1859.152	0.11
	Reg	Reg	Regressi...	BAD		0.196868	2485.364	0.16

```
Date:                January 31, 2018
Time:                16:27:20
*------------------------------------------------------------*
* Training Output
*------------------------------------------------------------*



Variable Summary

          Measurement    Frequency
 Role        Level         Count

TARGET      BINARY           1


```

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.

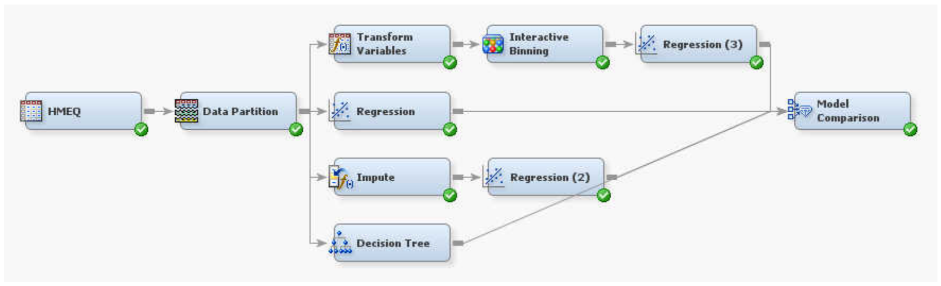
Minimize the Score Rankings Overlay window.

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

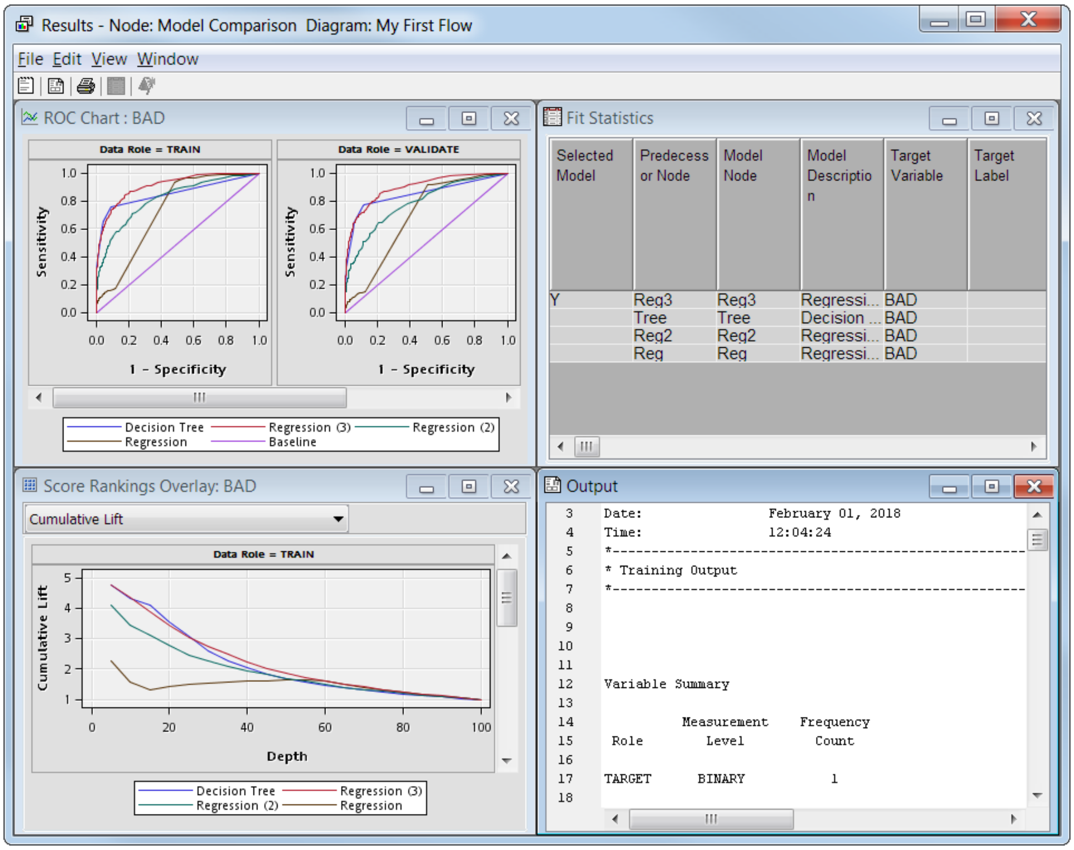
Now that you have modeled the input data with three different regression models, you want to fit a tree model to determine whether it will make better predictions. On the **Model** tab, drag a **Decision Tree** node to your diagram workspace. Connect the **Data Partition** node to the **Decision Tree**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



Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label
Y	Reg3	Reg3	Regressi...	BAD	
	Tree	Tree	Decision ...	BAD	
	Reg2	Reg2	Regressi...	BAD	
	Reg	Reg	Regressi...	BAD	

```
Date:                  February 01, 2018
Time:                  12:04:24
*------------------------------------------------------
* Training Output
*------------------------------------------------------



Variable Summary

              Measurement     Frequency
 Role            Level          Count

TARGET        BINARY             1

```

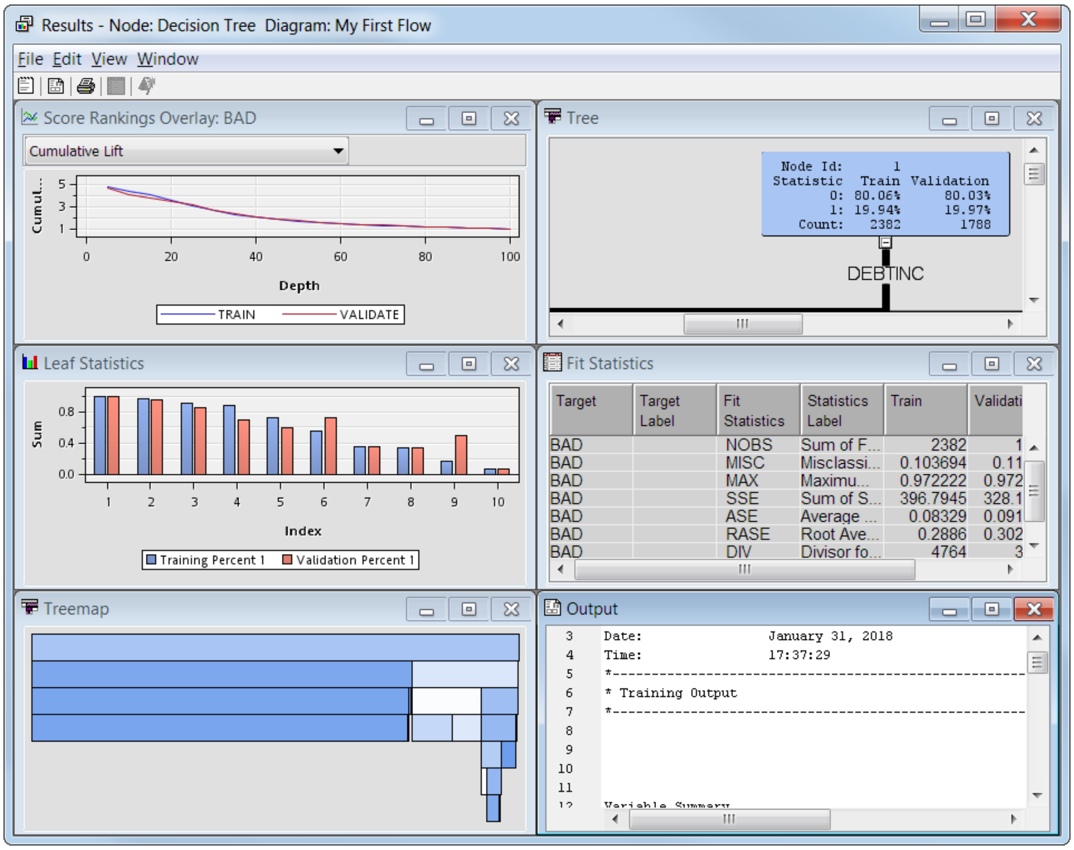
In the Score Rankings Overlay window, notice that **Cumulative % Response** graphs for the **Decision Tree** model and the **Regression (3)** are fairly similar. However, the **Regression (3)** model is better as the depth increases.

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**.



```
Node Id:     1
Statistic  Train  Validation
     0:  80.06%     80.03%
     1:  19.94%     19.97%
 Count:    2382       1788
```

Target	Target Label	Fit Statistics	Statistics Label	Train	Validati
BAD		NOBS	Sum of F...	2382	1
BAD		MISC	Misclassi...	0.103694	0.11
BAD		MAX	Maximu...	0.972222	0.972
BAD		SSE	Sum of S...	396.7945	328.1
BAD		ASE	Average ...	0.08329	0.091
BAD		RASE	Root Ave...	0.2886	0.302
BAD		DIV	Divisor fo...	4764	3

```
Date:                January 31, 2018
Time:                17:37:29
*------------------------------------------------------------
* Training Output
*------------------------------------------------------------




Variable Summary
```

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.

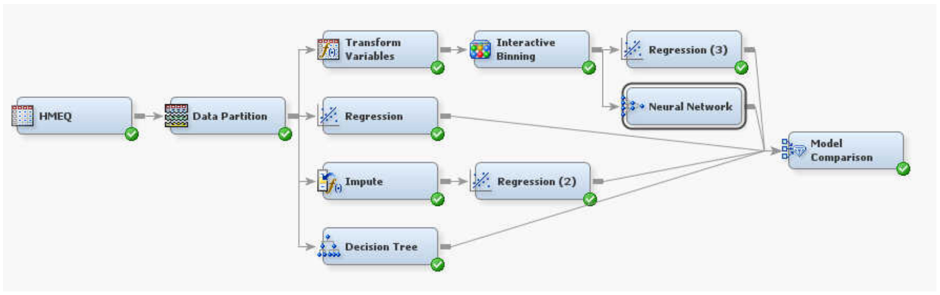
The Leaf Statistics window displays the training and validation classification rates for each leaf node in the tree model. Leaf nodes are terminal nodes in the decision tree. Select a bar in the Leaf Statistics window to select the corresponding leaf node in the Tree and Tree Map windows.

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.

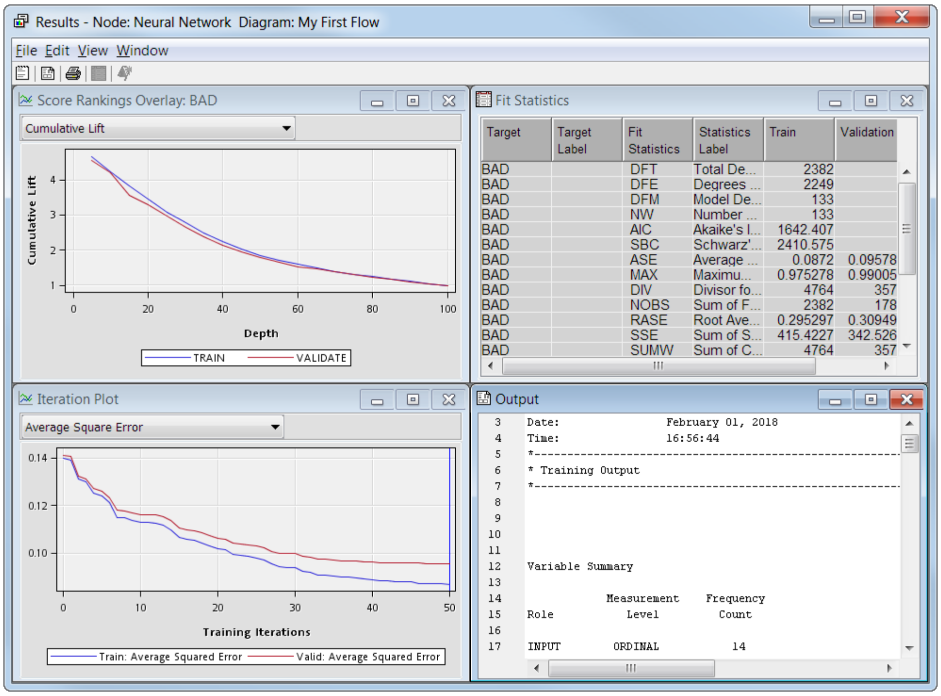
Fitting a Default Neural Network

Next, you want to compare how well a neural network model compares to the regression and decision 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.



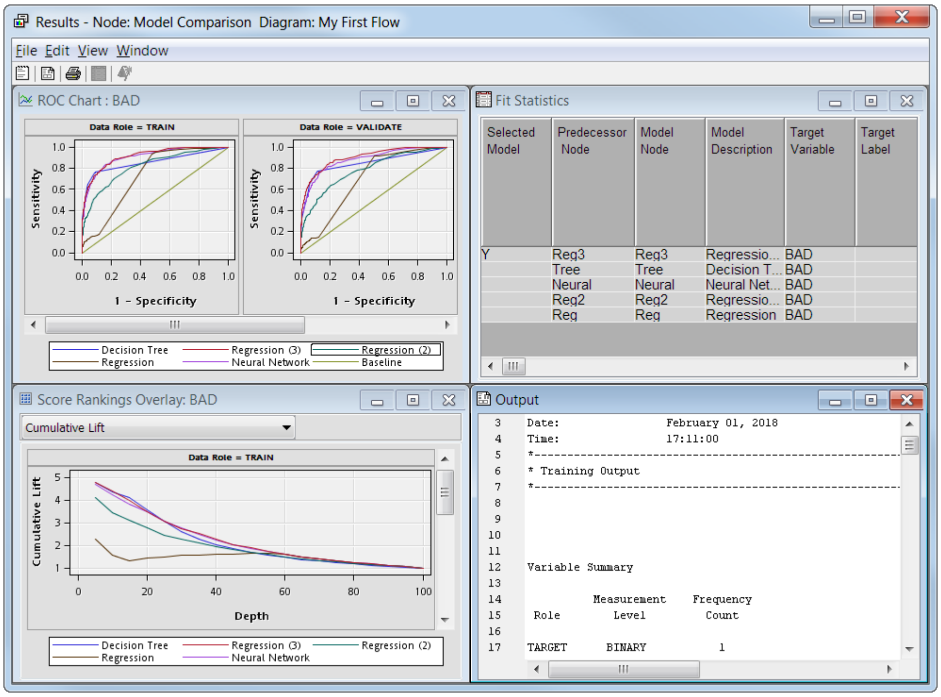
Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
BAD		DFT	Total De...	2382	
BAD		DFE	Degrees ...	2249	
BAD		DFM	Model De...	133	
BAD		NW	Number ...	133	
BAD		AIC	Akaike's I...	1642.407	
BAD		SBC	Schwarz'...	2410.575	
BAD		ASE	Average ...	0.0872	0.09578
BAD		MAX	Maximu...	0.975278	0.99005
BAD		DIV	Divisor fo...	4764	357
BAD		NOBS	Sum of F...	2382	178
BAD		RASE	Root Ave...	0.295297	0.30949
BAD		SSE	Sum of S...	415.4227	342.526
BAD		SUMW	Sum of C...	4764	357

```
Date:                     February 01, 2018
Time:                     16:56:44
*------------------------------------------------------------
* Training Output
*------------------------------------------------------------




Variable Summary

            Measurement    Frequency
Role           Level         Count

INPUT        ORDINAL          14
```

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**.



Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label
Y	Reg3	Reg3	Regressio...	BAD	
	Tree	Tree	Decision T...	BAD	
	Neural	Neural	Neural Net...	BAD	
	Reg2	Reg2	Regressio...	BAD	
	Reg	Reg	Regression	BAD	

```
Date:                    February 01, 2018
Time:                    17:11:00
*------------------------------------------------------------
* Training Output
*------------------------------------------------------------




Variable Summary

          Measurement    Frequency
 Role        Level         Count

TARGET      BINARY           1
```

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 model selection criterion. 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 out-perform the **Regression (3)** model. If both models performed poorly compared to the decision tree model, poor performance might be due to how missing values are handled. Decision tree models directly handle observations that have missing values, while regression and neural network models ignore observations that have missing values. This is why the original **Regression** model performance is significantly worse than the **Decision Tree** model.

In the **Neural Network** and **Regression (3)** models, 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. In the Properties Panel, set the **Type** property (in the **Indicator Variables** subgroup under **Score**) to **Unique**. Set the value of the **Source** property to **Missing Variables**. Set the value of the **Role** property to **Input**.

In the diagram workspace, select the **Model Comparison** node. Right-click the node and select **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** and **Decision Tree** models. In fact, the **Regression (2)** model outperforms the **Regression (3)** model, by a very thin margin. The model selection criterion uses the misclassification rate (Test data) to choose the champion model. The difference between the selection criterion scores for the **Regression (2)** model (0.111173) and the **Regression (3)** model (0.111732) is very small: 0.000002. The very small margin, indicates that the **Regression (2)** and **Regression (3)** models are incredibly close in performance. The **Regression (2)** model outperforms the **Regression (3)** model in the first 5 deciles of the training data, and the **Regression (3)** model outperforms the **Regression (2)** model on the last 5 deciles of the training data. If validation data is used to choose the champion model, the **Regression (3)** model outperforms the **Regression (2)** model, by a very thin margin. The two models are very close performers.

In cases like this, it is impossible to know which model will provide the best results when it is applied to new data. For this example, or any other, a good analyst considers many variations of each model and identifies the best model according to their own criteria. In our case, for simplicity, we will assume that **Regression (3)** is the selected model.

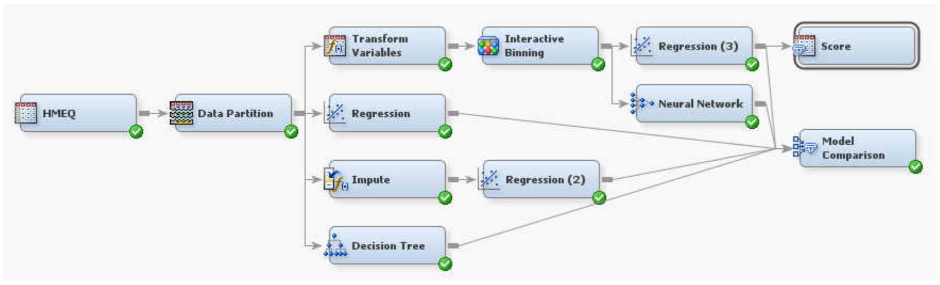
Close the Results window.

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.

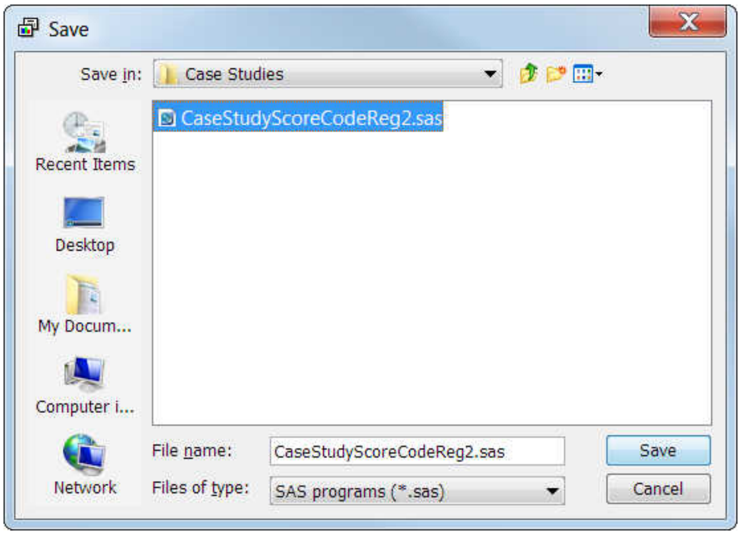
On the **Assess** tab, drag a **Score** node to your diagram workspace. Connect the **Regression (3)** node to the **Score** node.



With the **Score** node selected, in the Properties Panel, 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 (2)** model.

You can save the optimized score code by selecting **File** ⇨ **Save As** from the main menu on the Results - Node: Score window. In the Save window, navigate to an appropriate location for you to save the score code and enter `CaseStudyScoreCodeReg3.sas` in the **File name** field. This enables you to experiment with changes to a model without losing the score code for your original model.



Save

Save in: Case Studies

CaseStudyScoreCodeReg2.sas

File name: CaseStudyScoreCodeReg2.sas

Files of type: SAS programs (*.sas)

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. It does not matter 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 SAMPSIO 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. Go to line 1 of the code, and insert a blank line or two. In this space at the beginning of the code, enter the following lines:

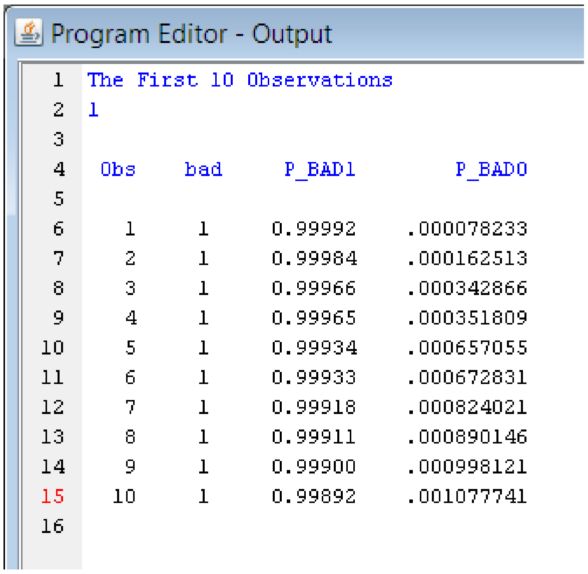
   ```
   data myPredicted;
       set sampsio.dmahmeq;
   ```

   These lines create the myPredicted data set and choose SAMPSIO.DMAHMEQ as the data set that is scored.
4. Scroll down to the end of the code, and insert a blank line. In this space at the end of the code, enter the following lines:

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

   This code will sort the observations by the values in a named variable. Here, we are sorting the observations by the values of P_BAD1. By sorting the observations by descending values of P_BAD1, you arrange the observations for the individuals that are most likely to default on a loan at the top of the printed results. After this code, insert a blank line, and enter the following lines:

   ```
   title "The First 10 Observations";
   PROC PRINT DATA=myPredicted(obs=10);
       VAR BAD P_BAD1 P_BAD0;
   run;
   ```
5. Submit the score code by selecting **Actions** ⇨ **Run** on the main menu.
6. After the code runs, 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.

   ```
   The First 10 Observations
   1

    Obs    bad     P_BAD1        P_BAD0

      1     1     0.99992    .000078233
      2     1     0.99984    .000162513
      3     1     0.99966    .000342866
      4     1     0.99965    .000351809
      5     1     0.99934    .000657055
      6     1     0.99933    .000672831
      7     1     0.99918    .000824021
      8     1     0.99911    .000890146
      9     1     0.99900    .000998121
     10     1     0.99892    .001077741
   ```

   BAD has two levels. The values of P_BAD1 and the values of P_BAD0 should always sum to 1 (after accounting for significant digits in summary values).

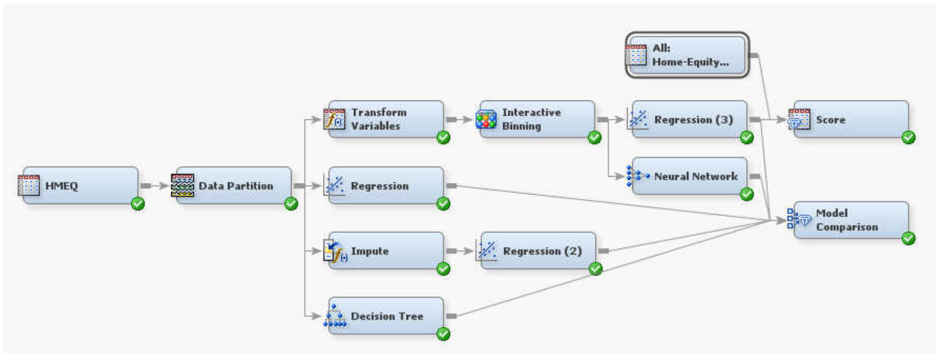
Close the Program Editor window to return to your process flow diagram.

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, set the **Level** for the variable REASON to **Binary**. 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.



Select the **Score** node in the diagram. In the Property Panel, 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.

In the Project Panel, select the name of your diagram under the **Diagrams** folder. This example has been using **My First Flow**. In the Properties Panel, note the value for the ID property. It should be a value similar to “EMWS1”. You will need to use this ID value in the following step.

Click the ellipsis button next to the **Code Editor** property of the **SAS Code** node. In the **Training Code** field, enter the following code:

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

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

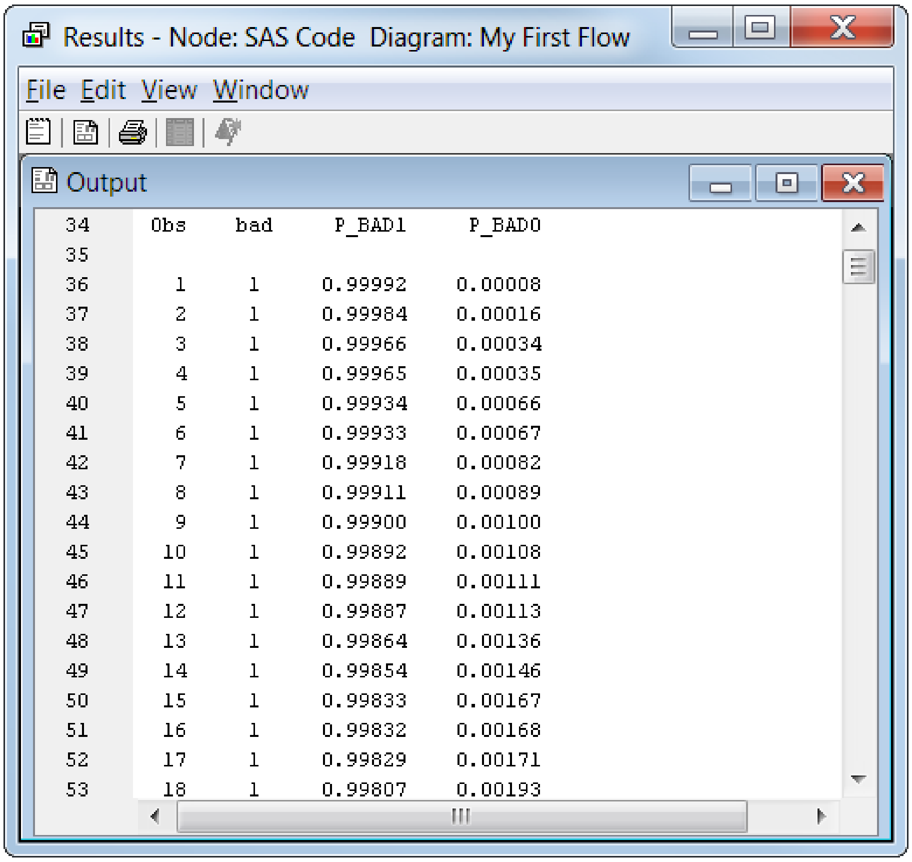
You must replace <diagramID> with the value of the **ID** property.

If your diagram ID is EMWS1, for example, your PROC SORT statement should become: **`PROC SORT DATA=EMWS1.Score_SCORE out=defaults;`**.

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.

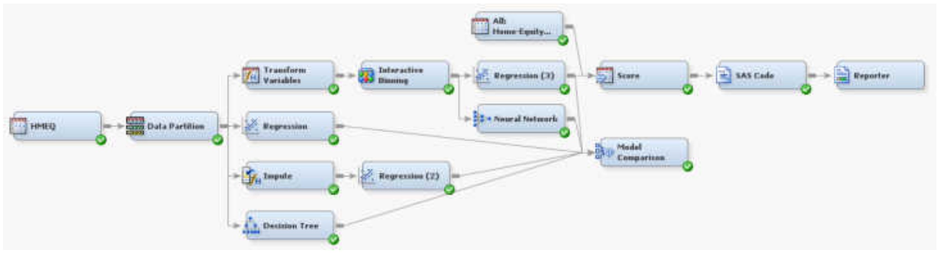
The scored values here should match those obtained in the previous section. Close the Results window.



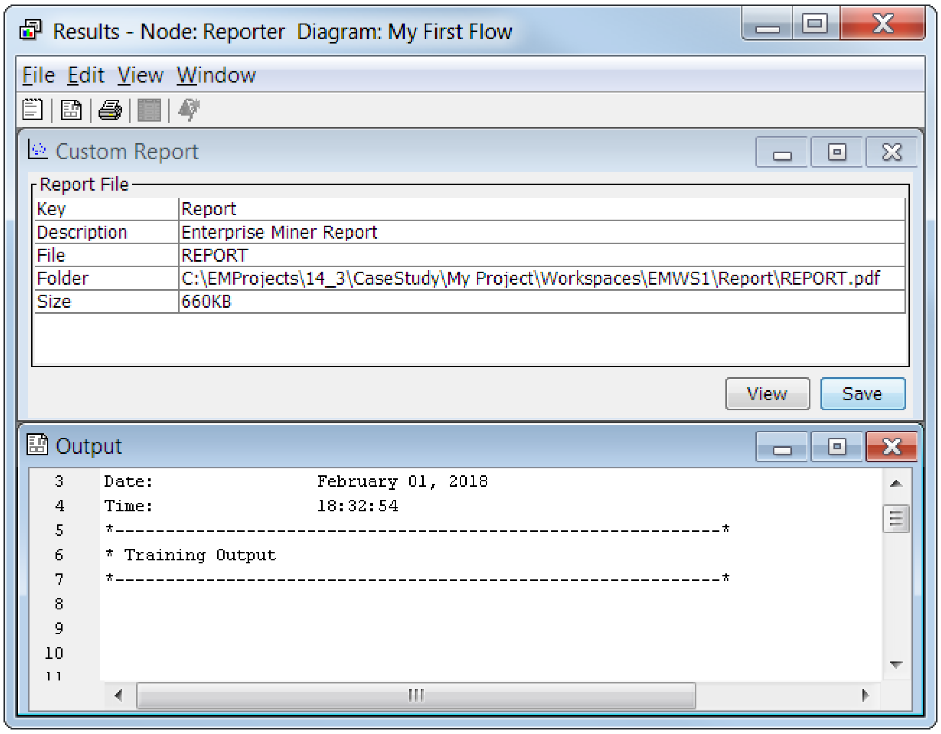
```
34   Obs   bad     P_BAD1     P_BAD0
35
36     1     1    0.99992    0.00008
37     2     1    0.99984    0.00016
38     3     1    0.99966    0.00034
39     4     1    0.99965    0.00035
40     5     1    0.99934    0.00066
41     6     1    0.99933    0.00067
42     7     1    0.99918    0.00082
43     8     1    0.99911    0.00089
44     9     1    0.99900    0.00100
45    10     1    0.99892    0.00108
46    11     1    0.99889    0.00111
47    12     1    0.99887    0.00113
48    13     1    0.99864    0.00136
49    14     1    0.99854    0.00146
50    15     1    0.99833    0.00167
51    16     1    0.99832    0.00168
52    17     1    0.99829    0.00171
53    18     1    0.99807    0.00193
```

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.



Report File

Key	Report
Description	Enterprise Miner Report
File	REPORT
Folder	C:\EMProjects\14_3\CaseStudy\My Project\Workspaces\EMWS1\Report\REPORT.pdf
Size	660KB

```
Date:                February 01, 2018
Time:                18:32:54
*------------------------------------------------------------*
* Training Output
*------------------------------------------------------------*
```

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 **71**

Using the Variable Selection Node **71**
Selecting Variables Using the R-Square Criterion 71
Creating and Evaluating a Neural Network Model 73

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.

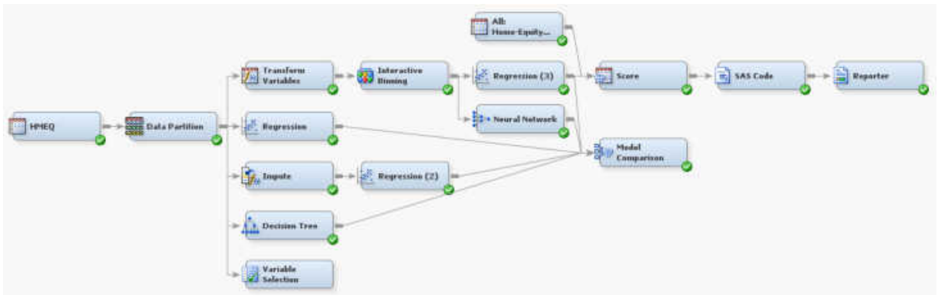
Because variable selection is a critical phase in model building, SAS Enterprise Miner provides the Variable Selection node. Variables selected for analysis in the Variable Selection node are available to any subsequent nodes. No single method of variable selection is universally better than any other 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.

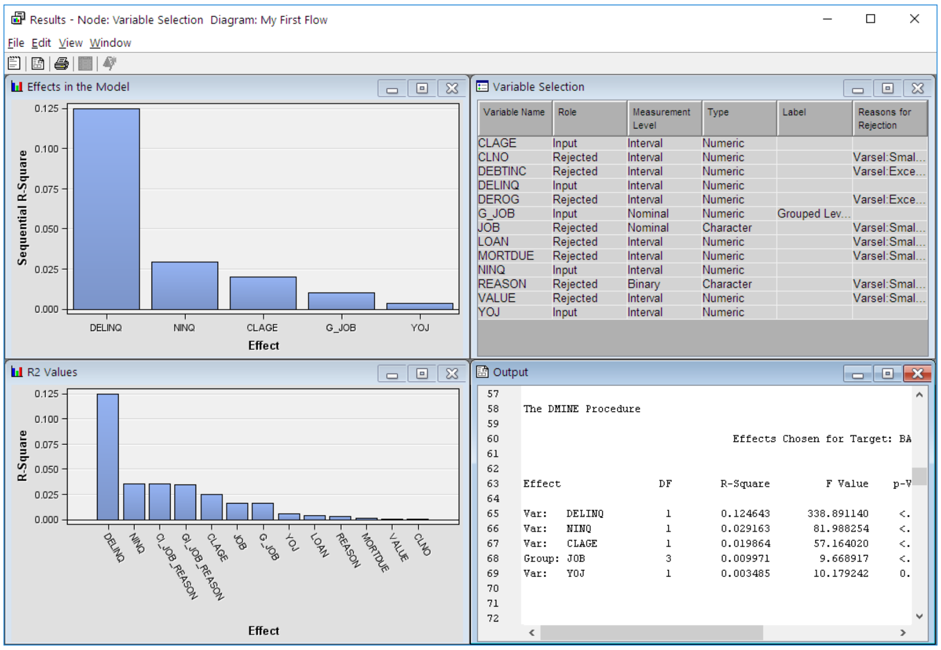
Set the value of the **Target Model** property to **R-Square**. This indicates that the r-square criterion is used to evaluate and select variables. Notice that the **Chi-Square Options** properties subgroup is now unavailable. Use the default values of the properties in the **R-Square Options** subgroup.

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 generated 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.



Variable Name	Role	Measurement Level	Type	Label	Reasons for Rejection
CLAGE	Input	Interval	Numeric		
CLNO	Rejected	Interval	Numeric		Varsel:Smal...
DEBTINC	Rejected	Interval	Numeric		Varsel:Exce...
DELINQ	Input	Interval	Numeric		
DEROG	Rejected	Interval	Numeric		Varsel:Exce...
G_JOB	Input	Nominal	Numeric	Grouped Lev...	
JOB	Rejected	Nominal	Character		Varsel:Smal...
LOAN	Rejected	Interval	Numeric		Varsel:Smal...
MORTDUE	Rejected	Interval	Numeric		Varsel:Smal...
NINQ	Input	Interval	Numeric		
REASON	Rejected	Binary	Character		Varsel:Smal...
VALUE	Rejected	Interval	Numeric		Varsel:Smal...
YOJ	Input	Interval	Numeric		

```

The DMINE Procedure

                                    Effects Chosen for Target: BA



Effect                  DF        R-Square          F Value     p-V

Var:    DELINQ           1        0.124643       338.891140     <.
Var:    NINQ             1        0.029163        81.988254     <.
Var:    CLAGE            1        0.019864        57.164020     <.
Group:  JOB              3        0.009971         9.668917     <.
Var:    YOJ              1        0.003485        10.179242     0.



```

In the Variable Selection table view at upper right, notice that CLAGE, DELINQ, G_JOB, NINQ, and YOJ show their **Role** set to **Input**. The list of inputs in the table is sorted alphabetically by variable name.

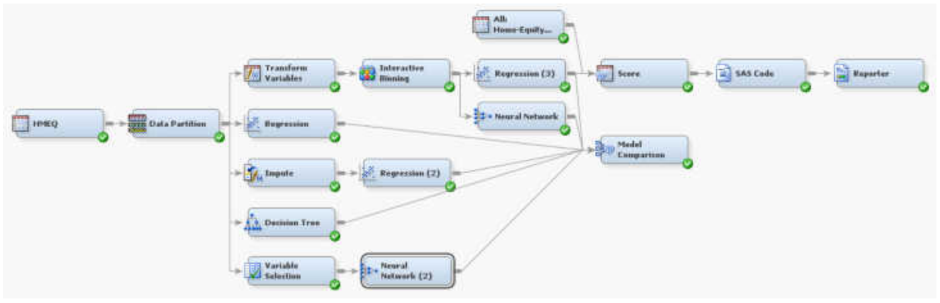
In the Output window, scroll down until you see the list of effects that were chosen for the target. The list is sorted by the descending selection criteria, R-Square values: DELINQ, NINQ, CLAGE, (Group) JOB, and YOJ. You can see the R-Square values for the variables in the R2 Effects effects plot.

DELINQ, NINQ, CLAGE, (Group) JOB, and YOJ are the five variables chosen by the Variable Selection node for inclusion in the as inputs for the following neural network model.

Close the Results window.

Creating and Evaluating a Neural Network Model

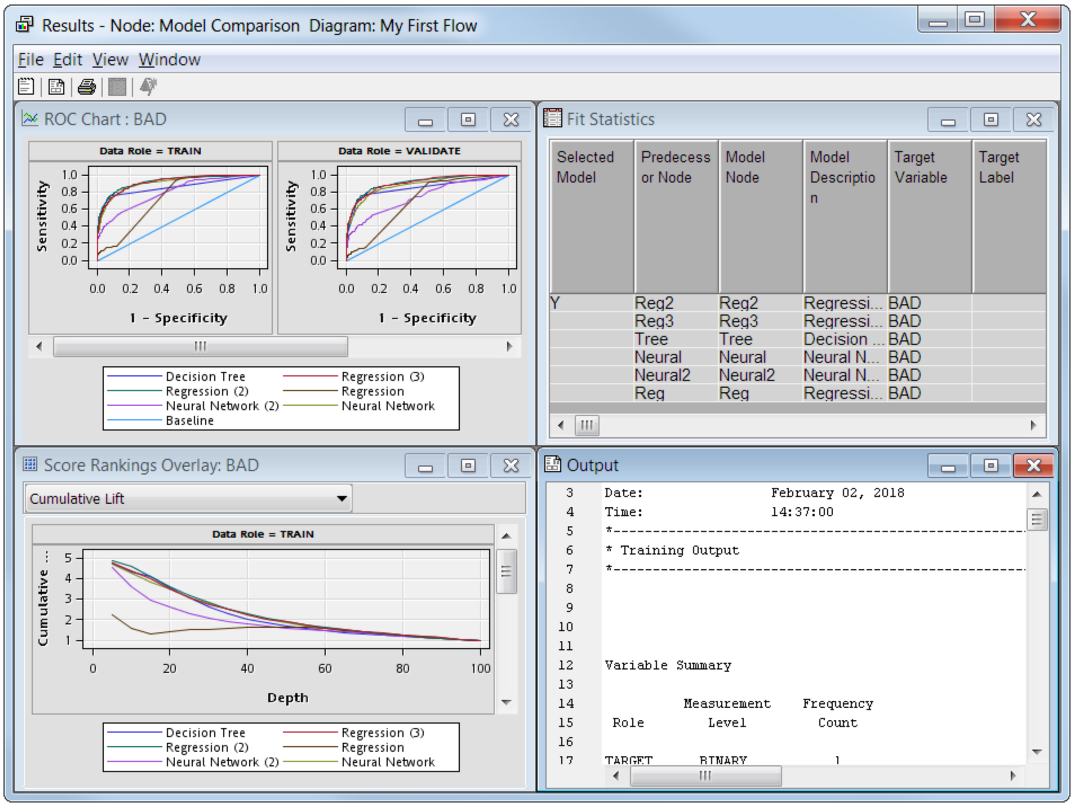
On the **Model** tab, drag a **Neural Network** node to your diagram workspace. Connect the **Variable Selection** node to the **Neural Network (2)** node. Connect the **Neural Network (2)** node to the **Model Comparison** node.



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.



Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label
Y	Reg2	Reg2	Regressi...	BAD	
	Reg3	Reg3	Regressi...	BAD	
	Tree	Tree	Decision ...	BAD	
	Neural	Neural	Neural N...	BAD	
	Neural2	Neural2	Neural N...	BAD	
	Reg	Reg	Regressi...	BAD	

```
Date:                February 02, 2018
Time:                14:37:00
*------------------------------------------------------------
* Training Output
*------------------------------------------------------------




Variable Summary

          Measurement     Frequency
 Role        Level          Count

TARGET      BINARY            1
```

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**.

Chapter 4
Clustering Tools

Problem Formulation **75**

Cluster Analysis **76**
Overview 76
Building the Process Flow Diagram 77
Using the Cluster Node 79
Examining the Clusters 81

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 SAMPSIO.DMABASE. The following table contains a description of the important variables.

Name	Model Role	Measurement Level	Description
NAME	ID	Nominal	Player name
TEAM	Rejected	Nominal	Team at the end of 1986
POSITION	Rejected	Nominal	Positions played at the end of 1986
LEAGUE	Rejected	Binary	League at the end of 1986
DIVISION	Rejected	Binary	Division at the end of 1986
NO_ATBAT	Input	Interval	Times at bat in 1986

Name	Model Role	Measurement Level	Description
NO_HITS	Input	Interval	Hits in 1986
NO_HOME	Input	Interval	Home runs in 1986
NO_RUNS	Input	Interval	Runs in 1986
NO_RBI	Input	Interval	RBIs in 1986
NO_BB	Input	Interval	Walks in 1986
YR_MAJOR	Input	Interval	Years in the major leagues
CR_ATBAT	Input	Interval	Career times at bat
CR_HITS	Input	Interval	Career hits
CR_HOME	Input	Interval	Career home runs
CR_RUNS	Input	Interval	Career runs
CR_RBI	Input	Interval	Career RBIs
CR_BB	Input	Interval	Career walks
NO_OUTS	Input	Interval	Put outs in 1986
NO_ASSTS	Input	Interval	Assists in 1986
NO_ERROR	Input	Interval	Errors in 1986
SALARY	Rejected	Interval	1987 salary in thousands
LOGSALAR	Input	Interval	Log of 1987 salary

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.

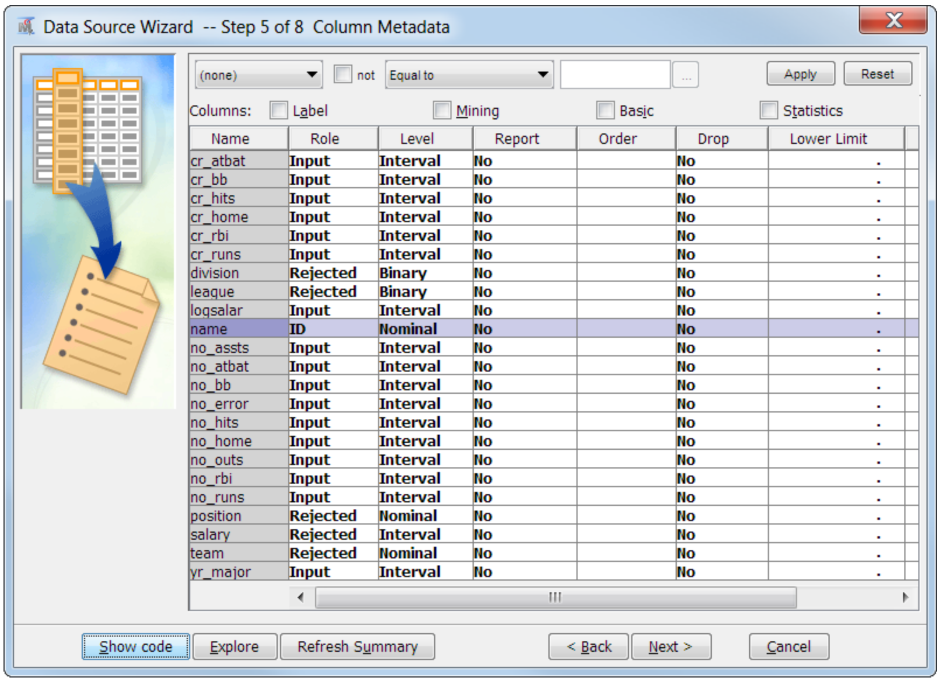
Clustering, on the other hand, is referred to as unsupervised classification because it identifies groups or classes within the data based on all the input variables. These groups, or clusters, are assigned numbers. However, the cluster number cannot be used to evaluate the proximity between clusters.

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**.



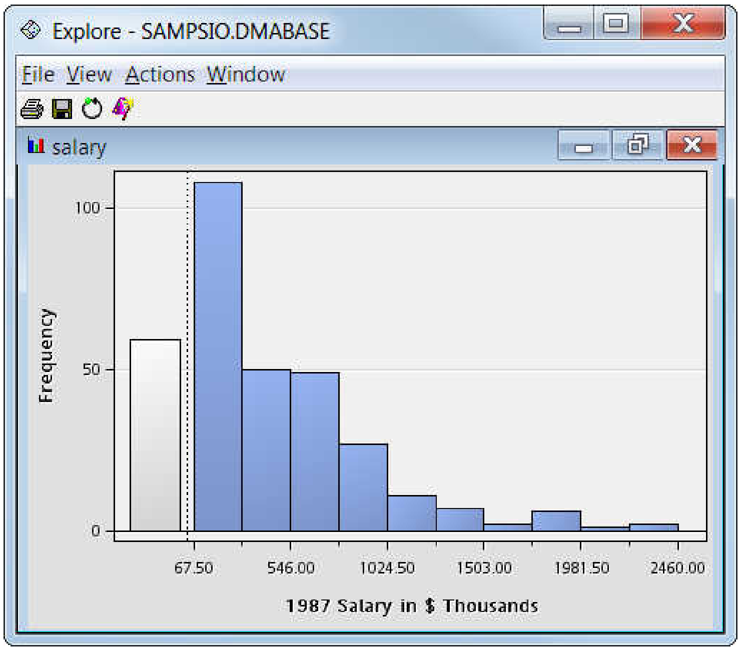
Name	Role	Level	Report	Order	Drop	Lower Limit
cr_atbat	Input	Interval	No		No	.
cr_bb	Input	Interval	No		No	.
cr_hits	Input	Interval	No		No	.
cr_home	Input	Interval	No		No	.
cr_rbi	Input	Interval	No		No	.
cr_runs	Input	Interval	No		No	.
division	Rejected	Binary	No		No	.
league	Rejected	Binary	No		No	.
logsalar	Input	Interval	No		No	.
name	ID	Nominal	No		No	.
no_assts	Input	Interval	No		No	.
no_atbat	Input	Interval	No		No	.
no_bb	Input	Interval	No		No	.
no_error	Input	Interval	No		No	.
no_hits	Input	Interval	No		No	.
no_home	Input	Interval	No		No	.
no_outs	Input	Interval	No		No	.
no_rbi	Input	Interval	No		No	.
no_runs	Input	Interval	No		No	.
position	Rejected	Nominal	No		No	.
salary	Rejected	Interval	No		No	.
team	Rejected	Nominal	No		No	.
yr_major	Input	Interval	No		No	.

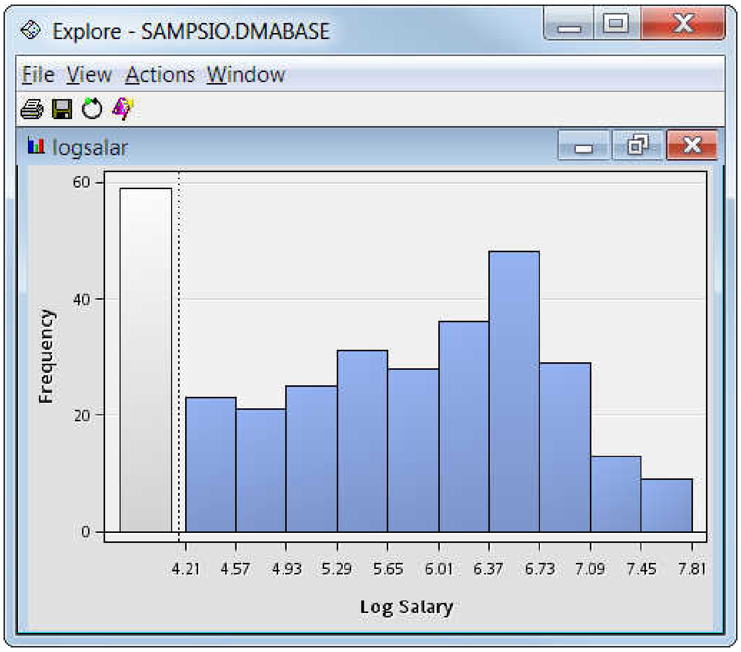
Click **Next**.

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.

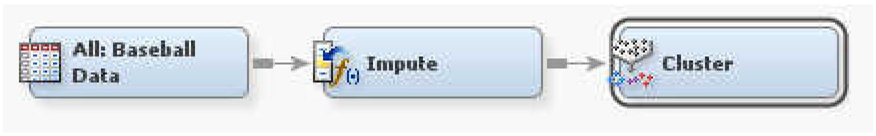




Although it is not always necessary to impute missing values, at times the amount of missing data can prevent the **Cluster** node from obtaining a solution. The **Cluster** node needs some complete observations in order to generate the initial clusters. When the amount of missing data is too extreme, 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**.

From 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.

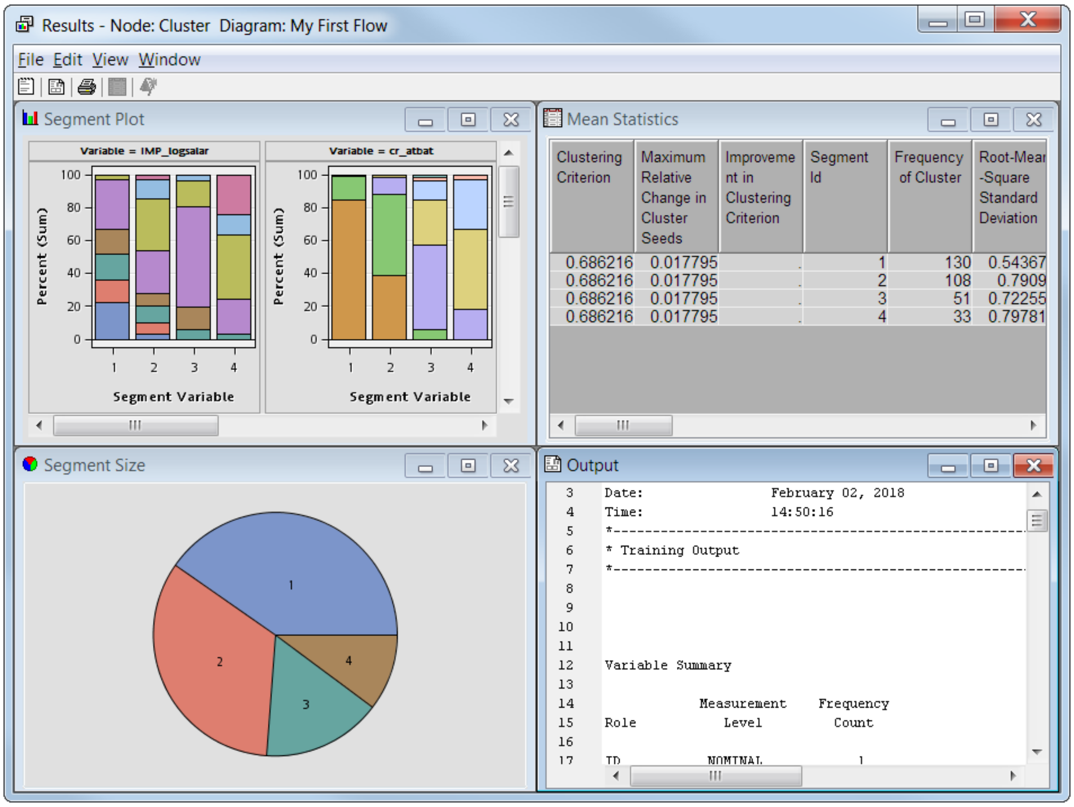
- The number of clusters must be greater than or equal to the number that is specified as the Minimum value in the Selection Criterion properties.
- The number of clusters must have cubic clustering criterion statistic values that are greater than the CCC Cutoff that is specified in the Selection Criterion 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.

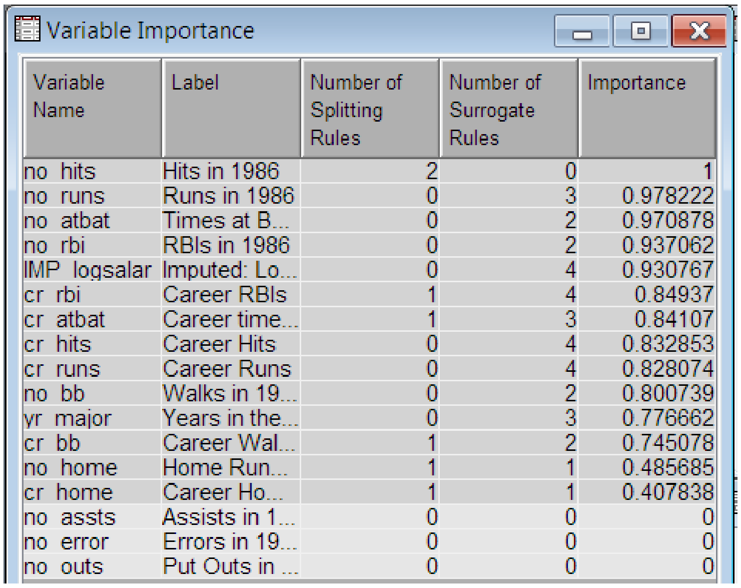


Clustering Criterion	Maximum Relative Change in Cluster Seeds	Improvement in Clustering Criterion	Segment Id	Frequency of Cluster	Root-Mean-Square Standard Deviation
0.686216	0.017795	.	1	130	0.54367
0.686216	0.017795	.	2	108	0.7909
0.686216	0.017795	.	3	51	0.72255
0.686216	0.017795	.	4	33	0.79781

```
Date:                   February 02, 2018
Time:                   14:50:16
*------------------------------------------------------------
* Training Output
*------------------------------------------------------------




Variable Summary

                Measurement    Frequency
Role               Level         Count

ID                NOMINAL          1
```

Notice in the Segment Size window that the Cluster node created 4 clusters.

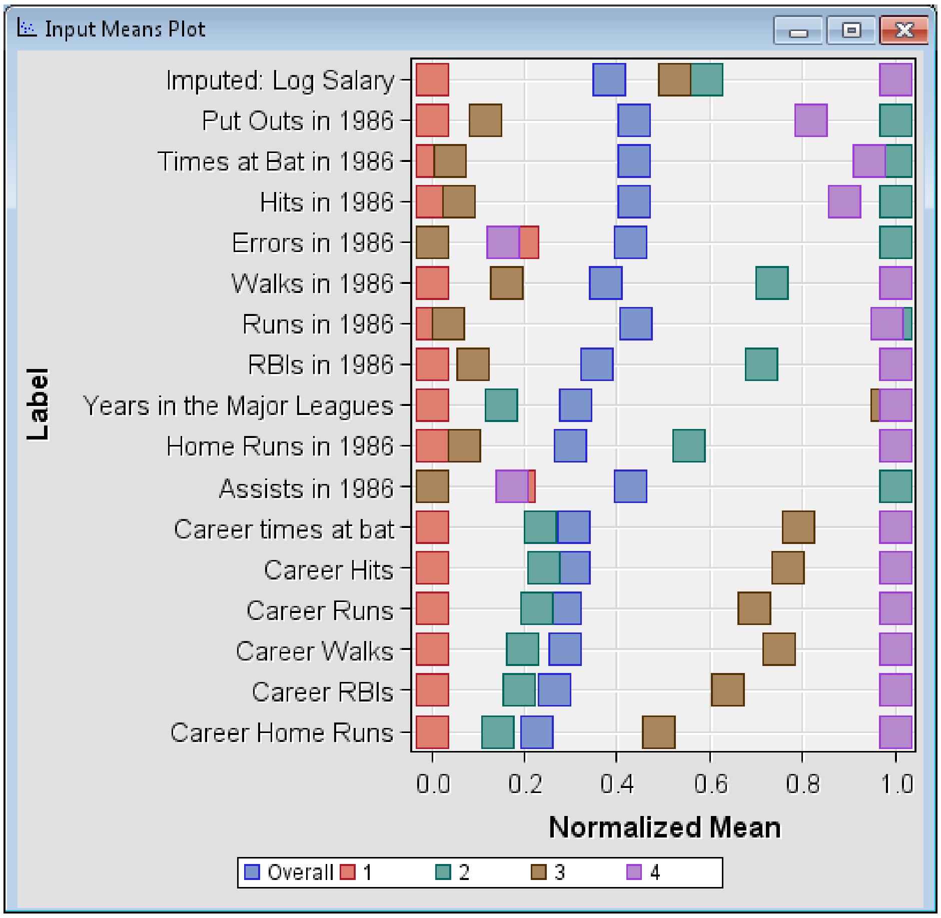
From the main menu of the Cluster Results window, select **View** ⇨ **Cluster Profile** ⇨ **Variable Importance**. The Variable Importance window displays each variable that was used to generate the clusters and their relative importance.


Variable Importance

Variable Name	Label	Number of Splitting Rules	Number of Surrogate Rules	Importance
no_hits	Hits in 1986	2	0	1
no_runs	Runs in 1986	0	3	0.978222
no_atbat	Times at B...	0	2	0.970878
no_rbi	RBIs in 1986	0	2	0.937062
IMP_logsalar	Imputed: Lo...	0	4	0.930767
cr_rbi	Career RBIs	1	4	0.84937
cr_atbat	Career time...	1	3	0.84107
cr_hits	Career Hits	0	4	0.832853
cr_runs	Career Runs	0	4	0.828074
no_bb	Walks in 19...	0	2	0.800739
yr_major	Years in the...	0	3	0.776662
cr_bb	Career Wal...	1	2	0.745078
no_home	Home Run...	1	1	0.485685
cr_home	Career Ho...	1	1	0.407838
no_assts	Assists in 1...	0	0	0
no_error	Errors in 19...	0	0	0
no_outs	Put Outs in ...	0	0	0

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**.



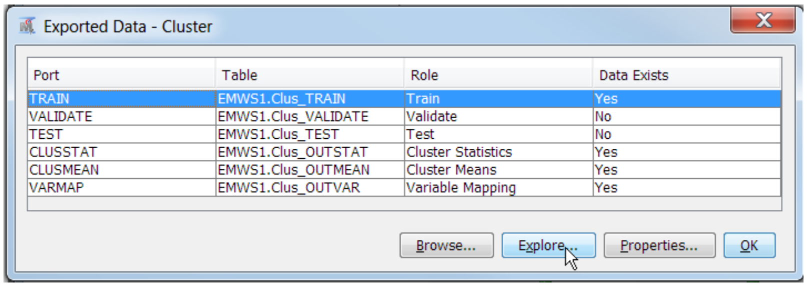
This plot displays the normalized mean value for each variable, both inside each cluster and for the complete data set. Notice that the in-cluster mean for cluster 1 is always less than the overall mean. But, in cluster 4, the in-cluster mean is almost always greater than the overall mean. Clusters 2 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.

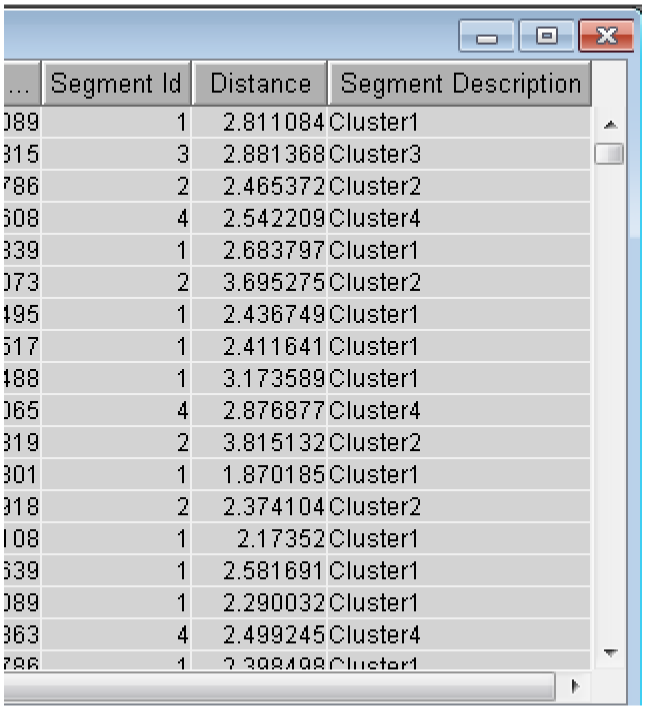
Examining the Clusters

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**.



Port	Table	Role	Data Exists
TRAIN	EMWS1.Clus_TRAIN	Train	Yes
VALIDATE	EMWS1.Clus_VALIDATE	Validate	No
TEST	EMWS1.Clus_TEST	Test	No
CLUSSTAT	EMWS1.Clus_OUTSTAT	Cluster Statistics	Yes
CLUSMEAN	EMWS1.Clus_OUTMEAN	Cluster Means	Yes
VARMAP	EMWS1.Clus_OUTVAR	Variable Mapping	Yes

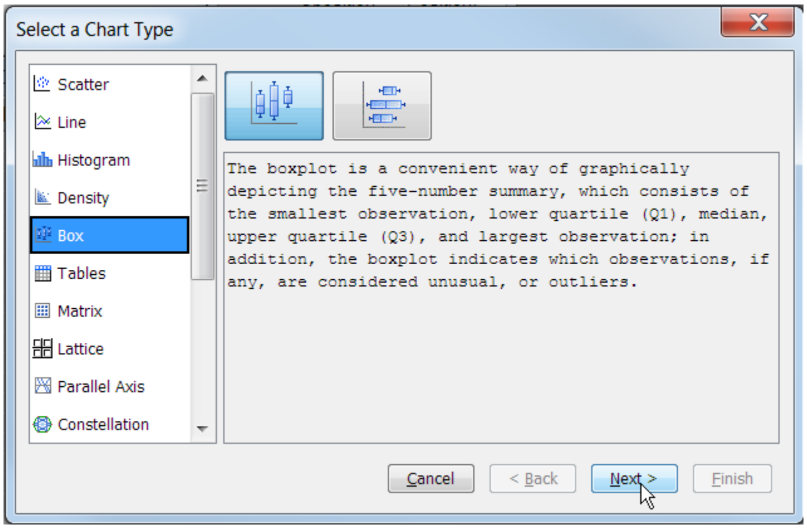
The Clus_TRAIN window contains the entire input data set and three additional columns that are appended to the end. Scroll to the right end of the window to locate the **Segment ID**, **Distance**, and **Segment Description** columns. The **Segment ID** and **Segment Description** columns display what cluster each observation belongs to.



...	Segment Id	Distance	Segment Description
089	1	2.811084	Cluster1
315	3	2.881368	Cluster3
786	2	2.465372	Cluster2
508	4	2.542209	Cluster4
339	1	2.683797	Cluster1
073	2	3.695275	Cluster2
495	1	2.436749	Cluster1
517	1	2.411641	Cluster1
488	1	3.173589	Cluster1
065	4	2.876877	Cluster4
319	2	3.815132	Cluster2
301	1	1.870185	Cluster1
318	2	2.374104	Cluster2
108	1	2.17352	Cluster1
539	1	2.581691	Cluster1
089	1	2.290032	Cluster1
363	4	2.499245	Cluster4
786	1	2.398498	Cluster1

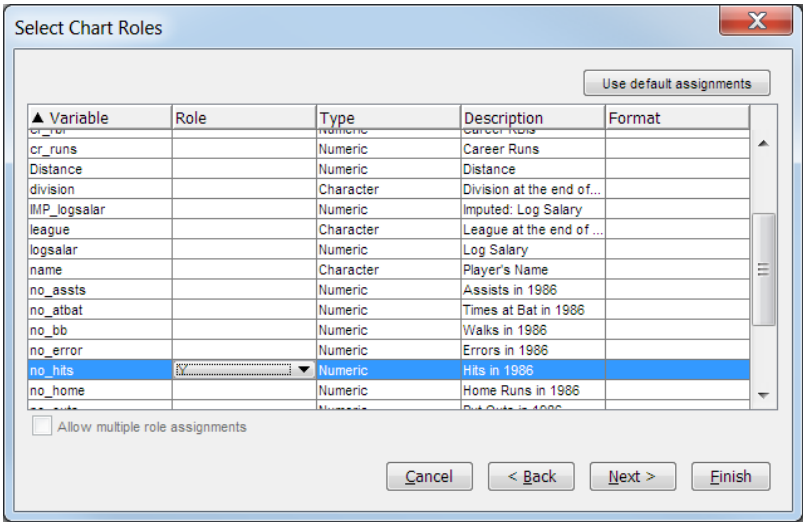
Next, create a plot that compares the number of hits for each cluster.

1. From the main menu of the Clus_TRAIN window, select **Actions** ⇨ **Plot**.
2. In the Select a Chart Type window, click **Box**. Click **Next**



```
The boxplot is a convenient way of graphically
depicting the five-number summary, which consists of
the smallest observation, lower quartile (Q1), median,
upper quartile (Q3), and largest observation; in
addition, the boxplot indicates which observations, if
any, are considered unusual, or outliers.
```

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**.

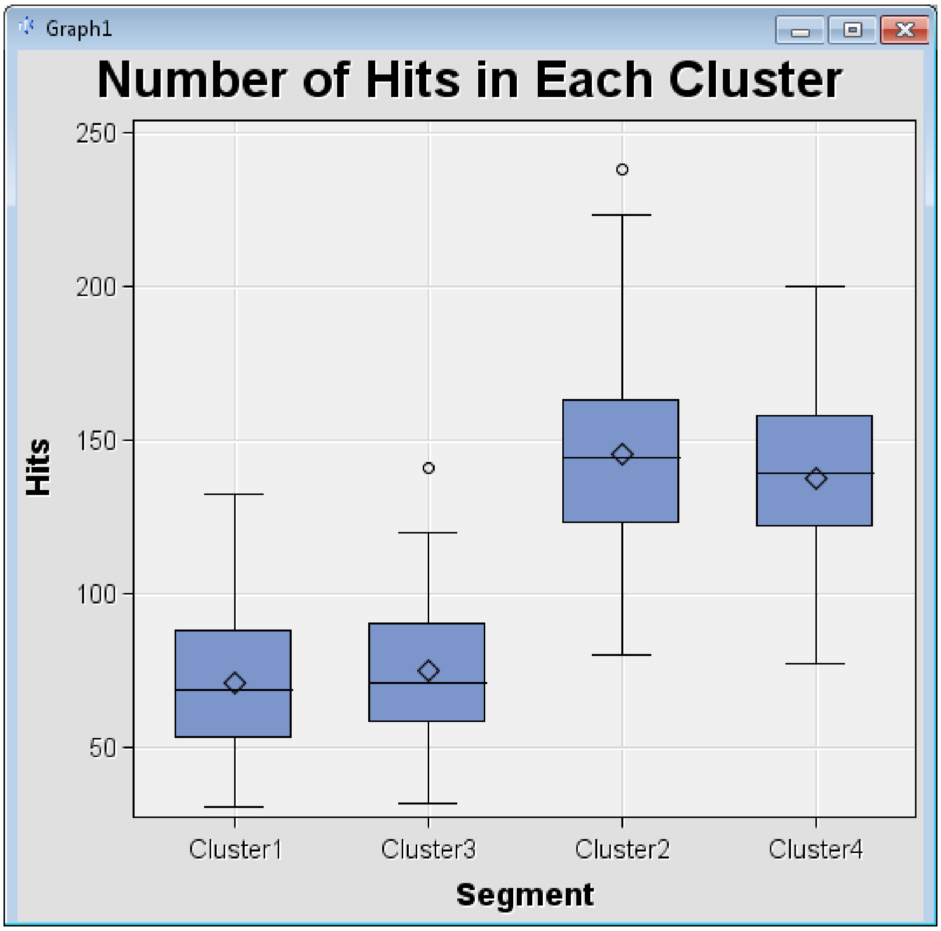


▲ Variable	Role	Type	Description	Format
cr_runs		Numeric	Career Runs	
Distance		Numeric	Distance	
division		Character	Division at the end of...	
IMP_logsalar		Numeric	Imputed: Log Salary	
league		Character	League at the end of ...	
logsalar		Numeric	Log Salary	
name		Character	Player's Name	
no_assts		Numeric	Assists in 1986	
no_atbat		Numeric	Times at Bat in 1986	
no_bb		Numeric	Walks in 1986	
no_error		Numeric	Errors in 1986	
no_hits	Y	Numeric	Hits in 1986	
no_home		Numeric	Home Runs in 1986	

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 **85**

Association Analysis **85**
Building the Process Flow Diagram 85
Understanding Analysis Modes 87
Running the Association Node 87

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

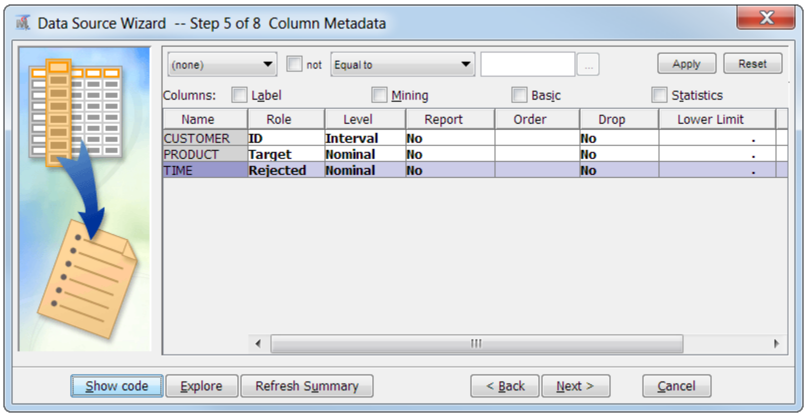
This example uses the same diagram workspace that you created in Chapter 2. You have the option to create a new diagram for this example, but instructions to do so are not provided in this example. First, you need to add the SAMPSIO.ASSOCS data source to project.

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**.

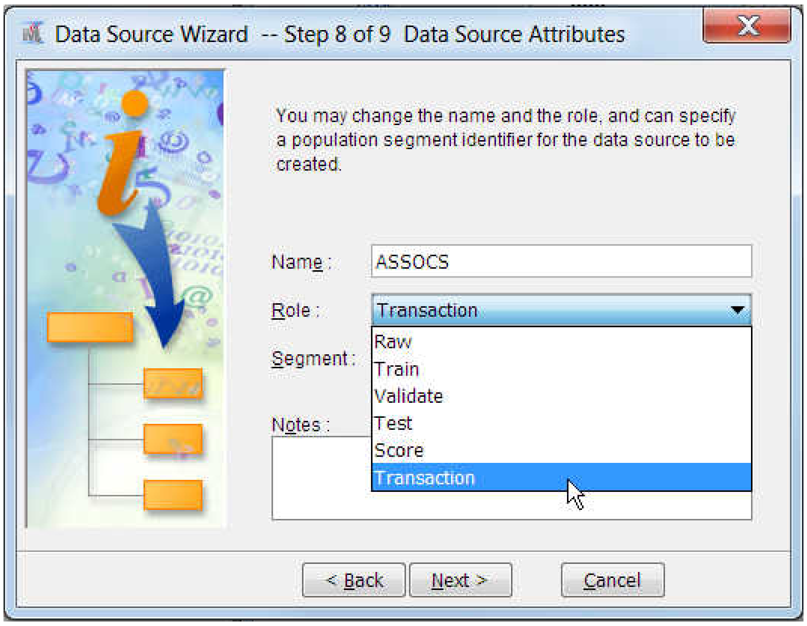
   Note: The variable TIME identifies the sequence in which the products were purchased. In this example, all of the products were purchased at the same time, so the order relates only to the order in which they are scanned at the register. When order is taken into account, association analysis is known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



Name	Role	Level	Report	Order	Drop	Lower Limit
CUSTOMER	ID	Interval	No		No	.
PRODUCT	Target	Nominal	No		No	.
TIME	Rejected	Nominal	No		No	.

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**.



You may change the name and the role, and can specify a population segment identifier for the data source to be created.

Name : ASSOCS

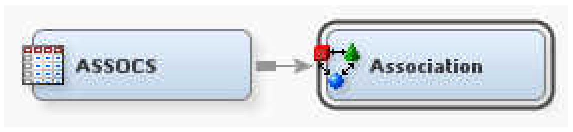
Role : Transaction

Segment :

Notes :

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.

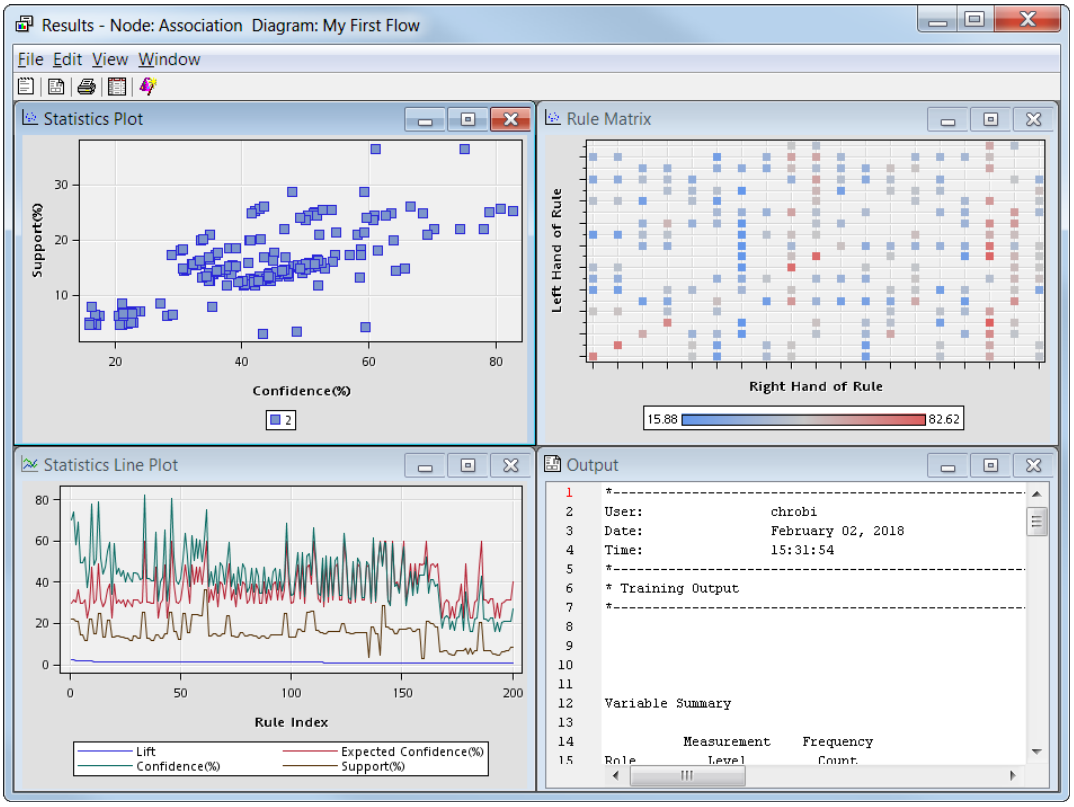
To perform sequence discovery, the input data set must have a separate observation for each product purchased by each customer at each visit. In addition to assignment of **ID** and **Target** roles, your input data must contain a **Sequence** variable. The sequence variable is used for timing comparisons. 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.



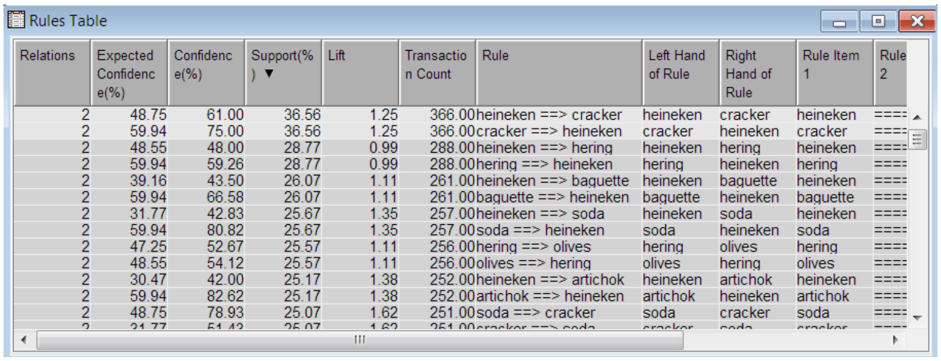
```
*------------------------------------------------------------
User:                  chrobi
Date:                  February 02, 2018
Time:                  15:31:54
*------------------------------------------------------------
* Training Output
*------------------------------------------------------------




Variable Summary

          Measurement     Frequency
Role         Level          Count
```

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 as a customer chosen at random to purchase B.

Sort the Rules Table by clicking the **Support (%)** column heading. 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.

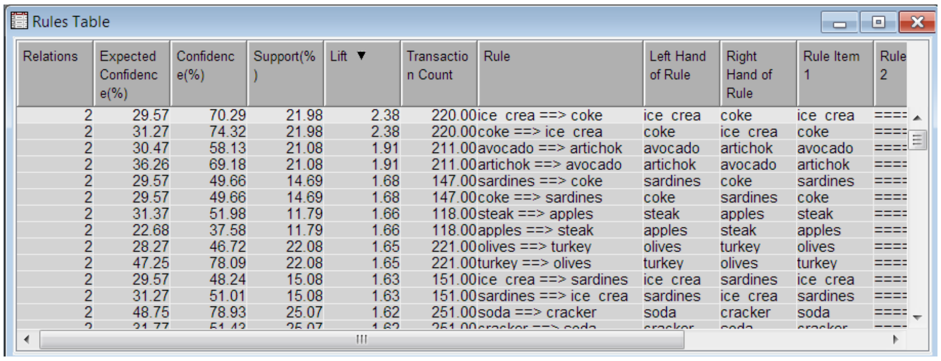

Rules Table

Relations	Expected Confidence(%)	Confidence(%)	Support(%) ▼	Lift	Transaction Count	Rule	Left Hand of Rule	Right Hand of Rule	Rule Item 1	Rule 2
2	48.75	61.00	36.56	1.25	366.00	heineken ==> cracker	heineken	cracker	heineken	====
2	59.94	75.00	36.56	1.25	366.00	cracker ==> heineken	cracker	heineken	cracker	====
2	48.55	48.00	28.77	0.99	288.00	heineken ==> hering	heineken	hering	heineken	====
2	59.94	59.26	28.77	0.99	288.00	hering ==> heineken	hering	heineken	hering	====
2	39.16	43.50	26.07	1.11	261.00	heineken ==> baguette	heineken	baguette	heineken	====
2	59.94	66.58	26.07	1.11	261.00	baguette ==> heineken	baguette	heineken	baguette	====
2	31.77	42.83	25.67	1.35	257.00	heineken ==> soda	heineken	soda	heineken	====
2	59.94	80.82	25.67	1.35	257.00	soda ==> heineken	soda	heineken	soda	====
2	47.25	52.67	25.57	1.11	256.00	hering ==> olives	hering	olives	hering	====
2	48.55	54.12	25.57	1.11	256.00	olives ==> hering	olives	hering	olives	====
2	30.47	42.00	25.17	1.38	252.00	heineken ==> artichok	heineken	artichok	heineken	====
2	59.94	82.62	25.17	1.38	252.00	artichok ==> heineken	artichok	heineken	artichok	====
2	48.75	78.93	25.07	1.62	251.00	soda ==> cracker	soda	cracker	soda	====
2	31.77	51.43	25.07	1.62	251.00	cracker ==> soda	cracker	soda	cracker	====

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**.


Rules Table

Relations	Expected Confidence(%)	Confidence(%)	Support(%)	Lift ▼	Transaction Count	Rule	Left Hand of Rule	Right Hand of Rule	Rule Item 1	Rule 2
2	29.57	70.29	21.98	2.38	220.00	ice crea ==> coke	ice crea	coke	ice crea	====
2	31.27	74.32	21.98	2.38	220.00	coke ==> ice crea	coke	ice crea	coke	====
2	30.47	58.13	21.08	1.91	211.00	avocado ==> artichok	avocado	artichok	avocado	====
2	36.26	69.18	21.08	1.91	211.00	artichok ==> avocado	artichok	avocado	artichok	====
2	29.57	49.66	14.69	1.68	147.00	sardines ==> coke	sardines	coke	sardines	====
2	29.57	49.66	14.69	1.68	147.00	coke ==> sardines	coke	sardines	coke	====
2	31.37	51.98	11.79	1.66	118.00	steak ==> apples	steak	apples	steak	====
2	22.68	37.58	11.79	1.66	118.00	apples ==> steak	apples	steak	apples	====
2	28.27	46.72	22.08	1.65	221.00	olives ==> turkey	olives	turkey	olives	====
2	47.25	78.09	22.08	1.65	221.00	turkey ==> olives	turkey	olives	turkey	====
2	29.57	48.24	15.08	1.63	151.00	ice crea ==> sardines	ice crea	sardines	ice crea	====
2	31.27	51.01	15.08	1.63	151.00	sardines ==> ice crea	sardines	ice crea	sardines	====
2	48.75	78.93	25.07	1.62	251.00	soda ==> cracker	soda	cracker	soda	====
2	31.77	51.43	25.07	1.62	251.00	cracker ==> soda	cracker	soda	cracker	====

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.

Chapter 6
Link Analysis

Problem Formulation **91**

Examining Web Log Data **92**
Building the Process Flow Diagram 92
Running the Link Analysis Node 93

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

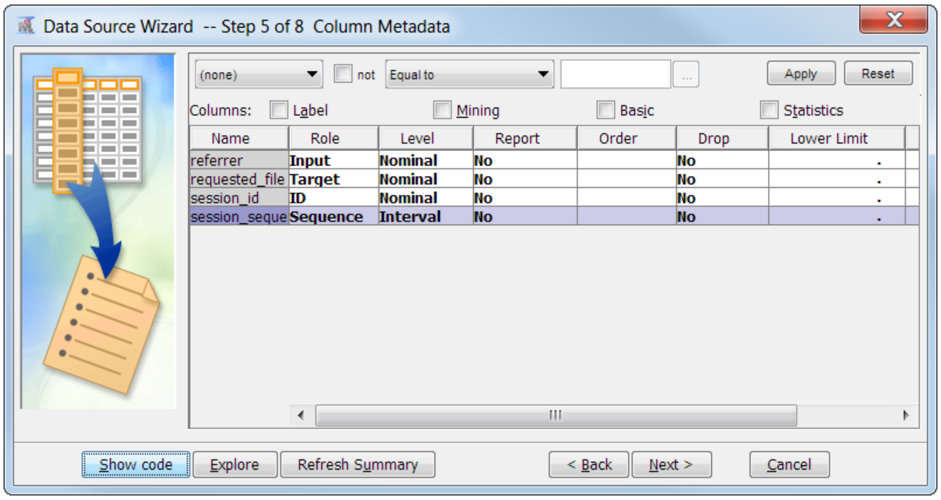
Variable Name	Model Role	Measurement Level	Description
REFERRER	Rejected	Nominal	Referring URL
REQUESTED_FILE	Target	Nominal	File that the visitor clicked
SESSION_IDENTIFIER	ID	Nominal	Unique ID code for the session
SESSION_SEQUENCE	Sequence	Interval	Order in which files were requested within a session

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.

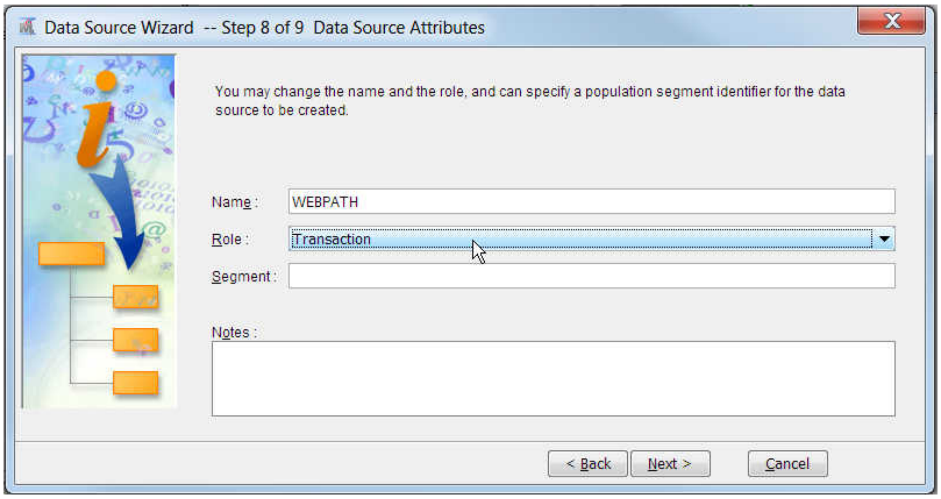
1. In the Project Panel, right-click **Data Sources** and click **Create Data Source**.
2. In the Data Source Wizard — Metadata Source window, click **Next**.
3. In the Data Source Wizard — Select a SAS Table and enter **`SAMPSIO.WEBPATH`** in the **Table** field. Click **Next**.
4. In the Data Source 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**.



Name	Role	Level	Report	Order	Drop	Lower Limit
referrer	Input	Nominal	No		No	.
requested_file	Target	Nominal	No		No	.
session_id	ID	Nominal	No		No	.
session_seque	Sequence	Interval	No		No	.

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**.



Data Source Wizard -- Step 8 of 9 Data Source Attributes

You may change the name and the role, and can specify a population segment identifier for the data source to be created.

Name : WEBPATH

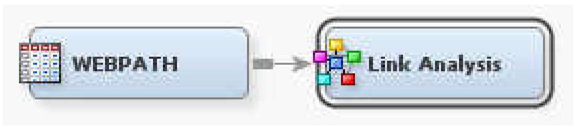
Role : Transaction

Segment :

Notes :

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

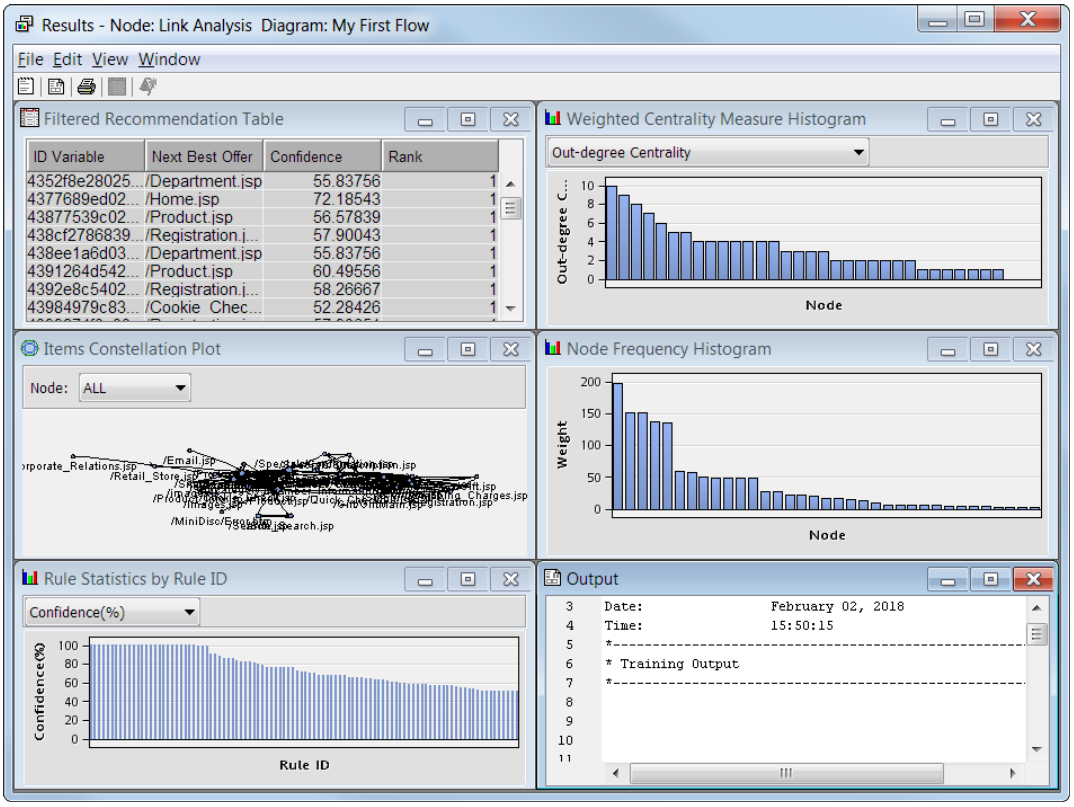
In the Project Panel, drag the **WEBPATH** data source to your diagram workspace. From the **Explore** tab, drag a **Link Analysis** node to your diagram workspace. Connect the **WEBPATH** data source to the **Link Analysis** node.



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

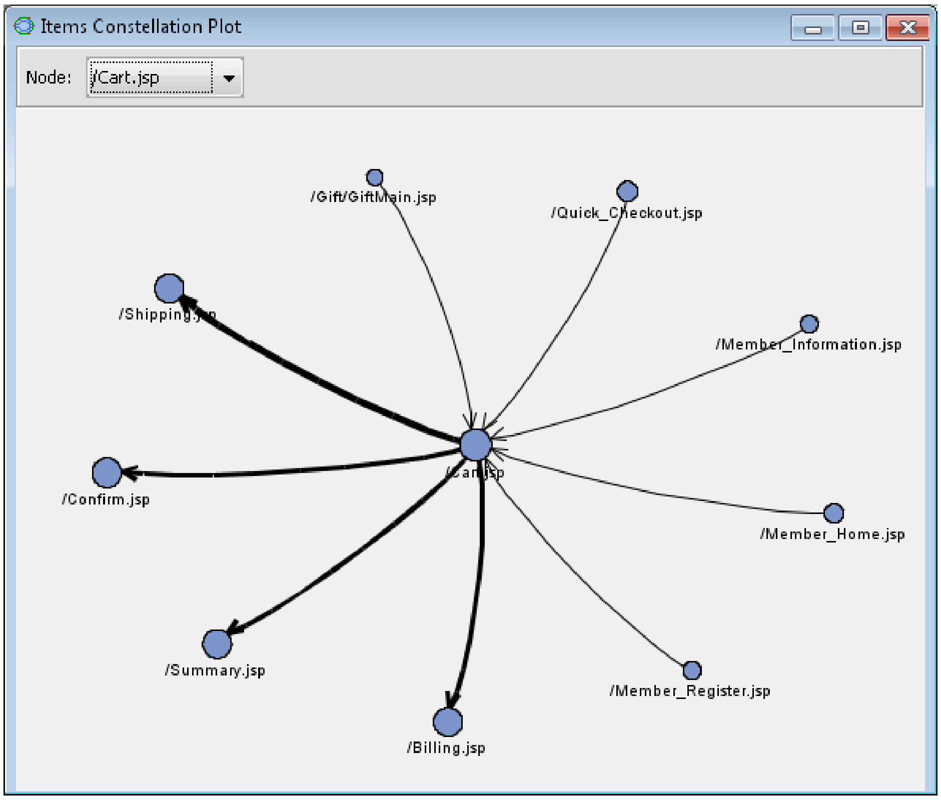
In your diagram workspace, right-click the **Link Analysis** node and click **Run**. In the Confirmation window, click **Yes**. Click **Results** in the Run Status window.



ID Variable	Next Best Offer	Confidence	Rank
4352f8e28025...	/Department.jsp	55.83756	1
4377689ed02...	/Home.jsp	72.18543	1
43877539c02...	/Product.jsp	56.57839	1
438cf2786839...	/Registration.j...	57.90043	1
438ee1a6d03...	/Department.jsp	55.83756	1
4391264d542...	/Product.jsp	60.49556	1
4392e8c5402...	/Registration.j...	58.26667	1
43984979c83...	/Cookie_Chec...	52.28426	1

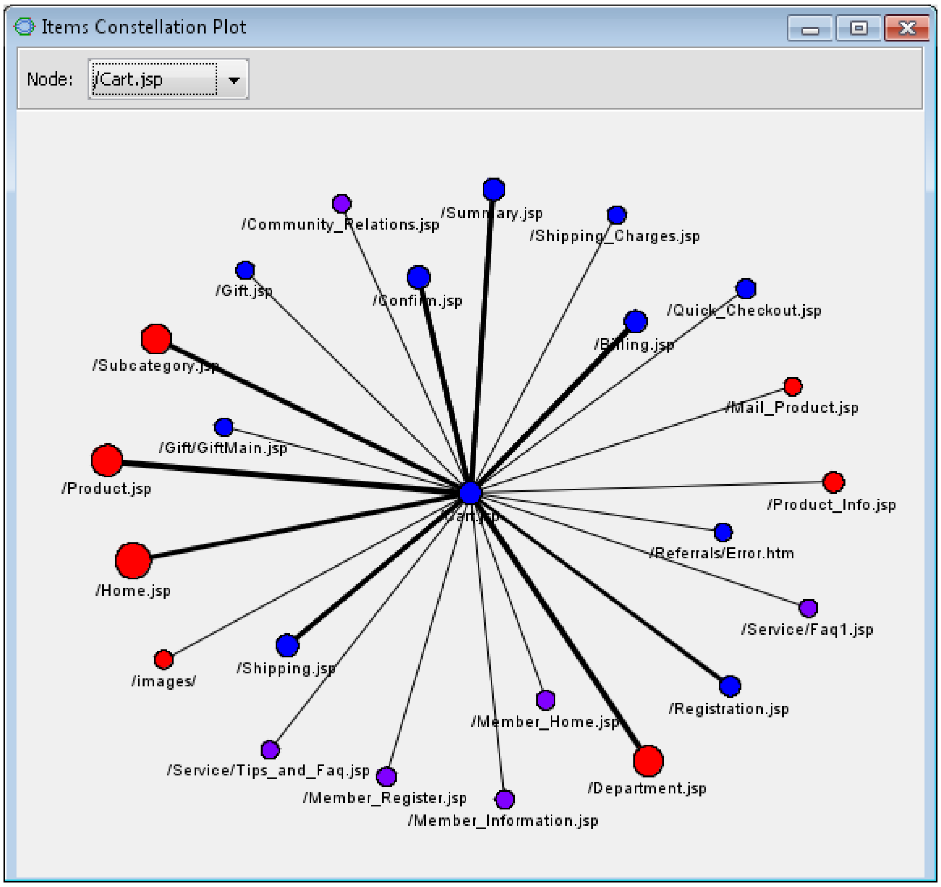
```
Date:                February 02, 2018
Time:                15:50:15
*------------------------------------------------------------
* Training Output
*------------------------------------------------------------
```

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**.



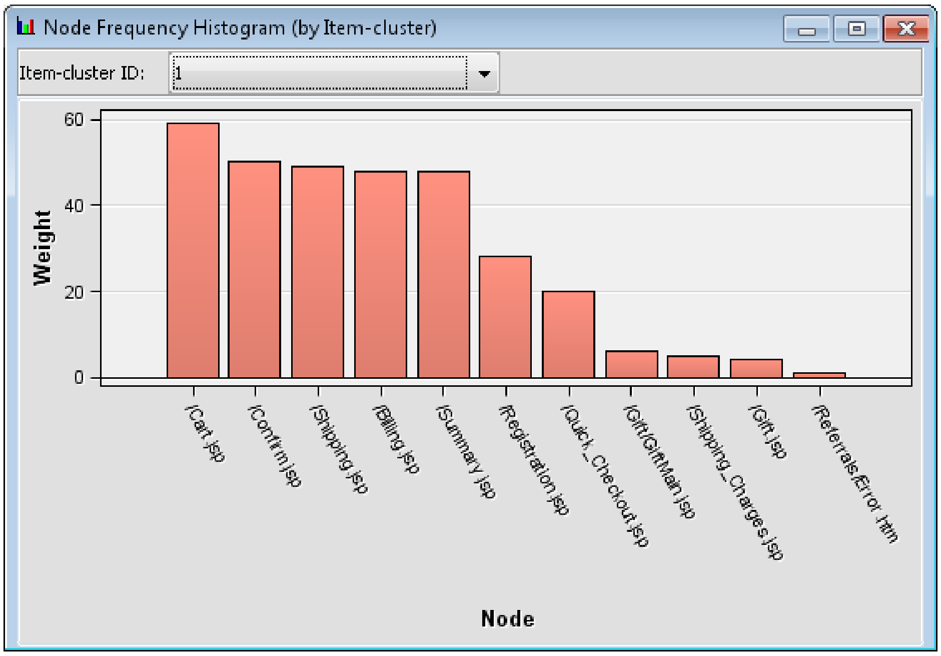
The thickness of each arrow represents the relative frequency of that link. In this example, the most frequent links appear to be from **/Cart.jsp**to **/Confirm.jsp**, **/ Summary.jsp**, **/Billing.jsp**, and **/Shipping.jsp**.

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.

Recommended Reading

Here is the recommended reading list for this title:

- The online Help for SAS Enterprise Miner
- *Getting Started with SAS Enterprise Miner 14.3.*
- "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"