

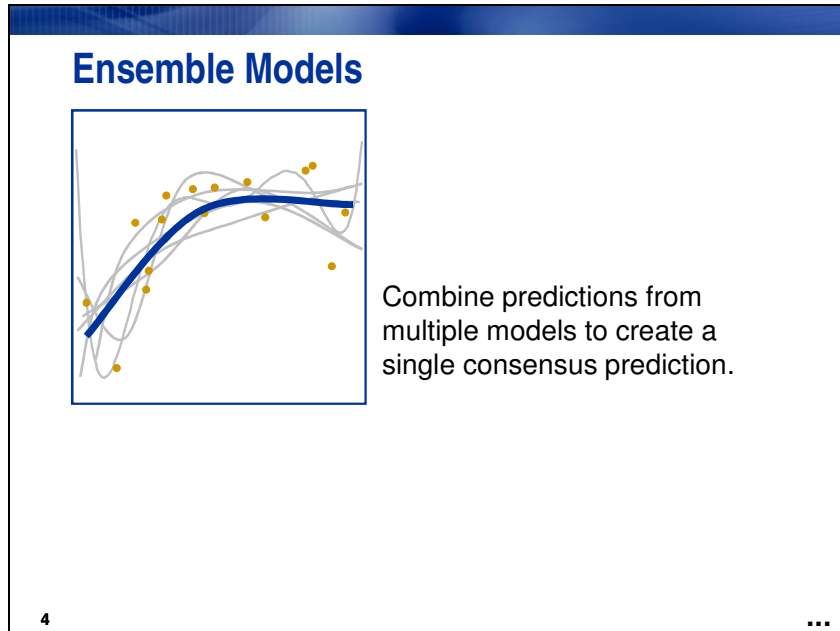
Chapter 9 Special Topics

0.1	Introduction	Error! Bookmark not defined.
0.2	A Section Title	Error! Bookmark not defined.
	Demonstration: <Type title of demo here.>	Error! Bookmark not defined.
	Exercises.....	Error! Bookmark not defined.
0.3	Chapter Summary	Error! Bookmark not defined.
0.4	Solutions	Error! Bookmark not defined.
	Solutions to Exercises.....	Error! Bookmark not defined.
	Solutions to Student Activities (Polls/Quizzes)	Error! Bookmark not defined.

9.1 Introduction

This chapter contains a selection of optional special topics related to predictive modeling. Unlike other chapters, each section is independent of the preceding section. However, demonstrations in this chapter depend on the demonstrations found in Chapters 2 through 8.

9.2 Ensemble Models



The Ensemble node creates a new model by combining the predictions from multiple models. For prediction estimates and rankings, this is usually done by averaging. When the predictions are decisions, this is done by voting. The commonly observed advantage of ensemble models is that the combined model is better than the individual models that compose it. It is important to note that the ensemble model can be more accurate than the individual models only if the individual models disagree with one another. You should always compare the model performance of the ensemble model with the individual models.



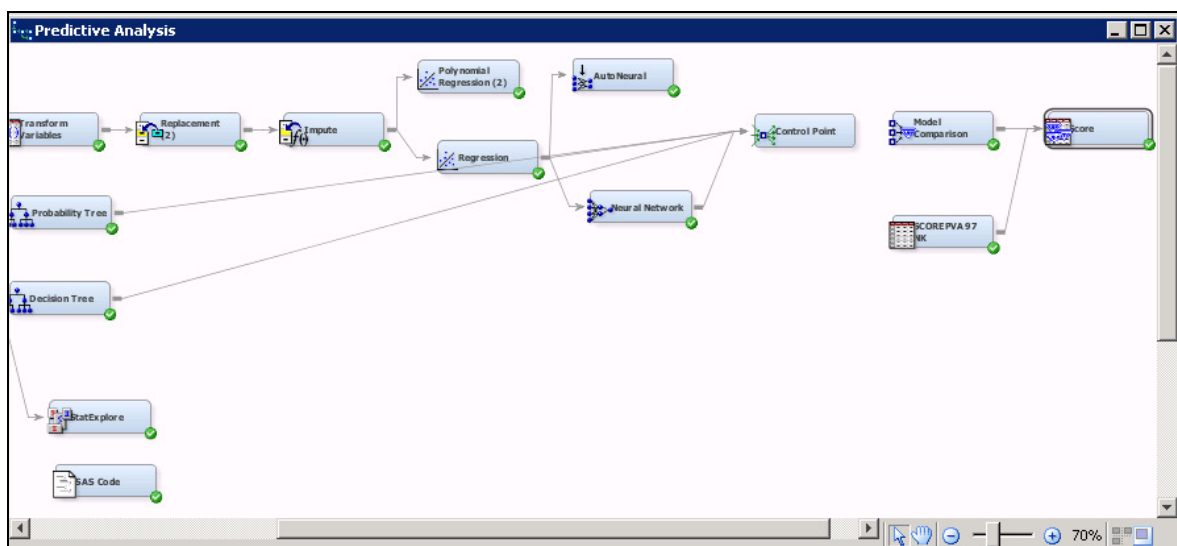
Creating Ensemble Models

This demonstration continues from the **Predictive Analysis** diagram. This demonstration proceeds in two parts. First, a control point is added to the process flow to reduce clutter. Then, the Ensemble tool is introduced as potential prediction model.

Diagram Control Point

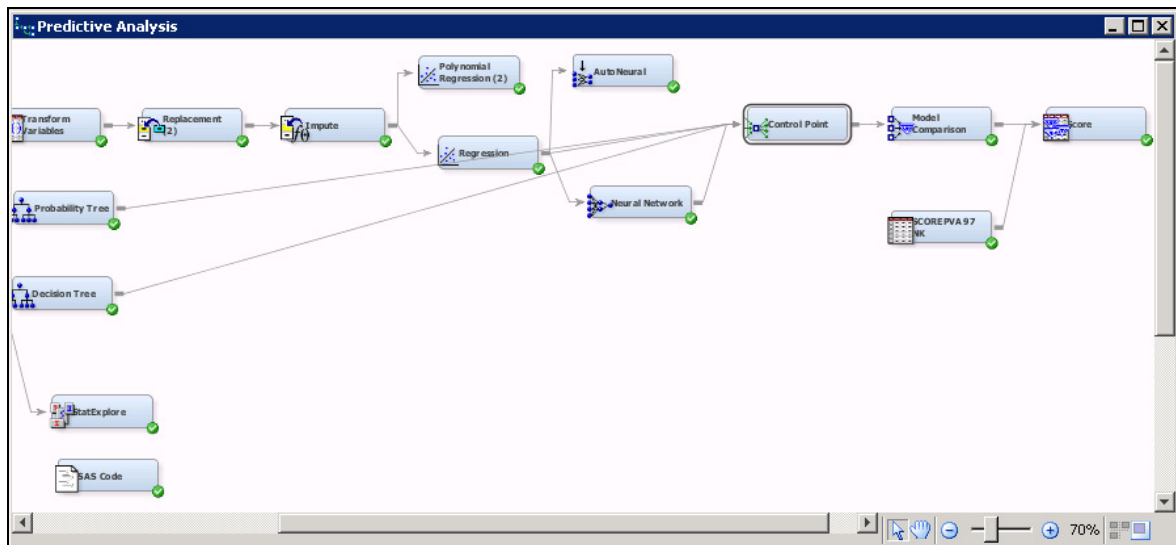
Follow these steps to add a control point to your analysis diagram:

1. Select the **Utility** tab.
2. Drag a **Control Point** tool into the diagram workspace.
3. Delete all connections to the Model Comparison node.



4. Connect model nodes of interest to the **Control Point** node.
5. Connect the **Control Point** node to the **Model Comparison** node.

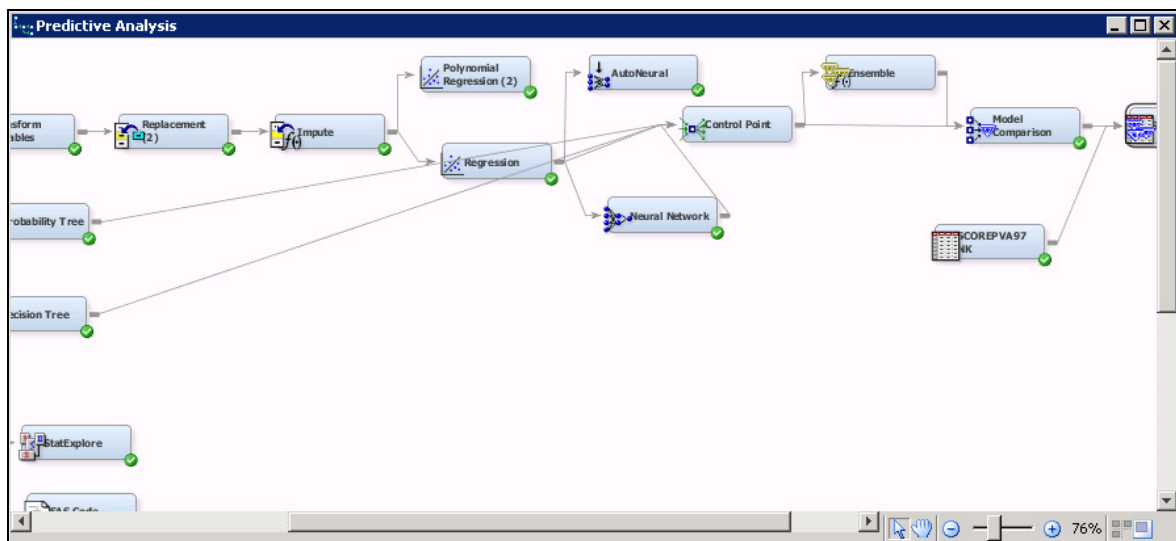
The completed process flow should appear as shown.



The Control Point node serves as a junction in the diagram. A single connection out of the Control Point node is equivalent to all the connections into the node.

6. Select the **Model** tab.
7. Drag an **Ensemble** tool into the diagram.
8. Connect the **Control Point** node to the **Ensemble** node.
9. Connect the **Ensemble** node to the **Model Comparison** node.

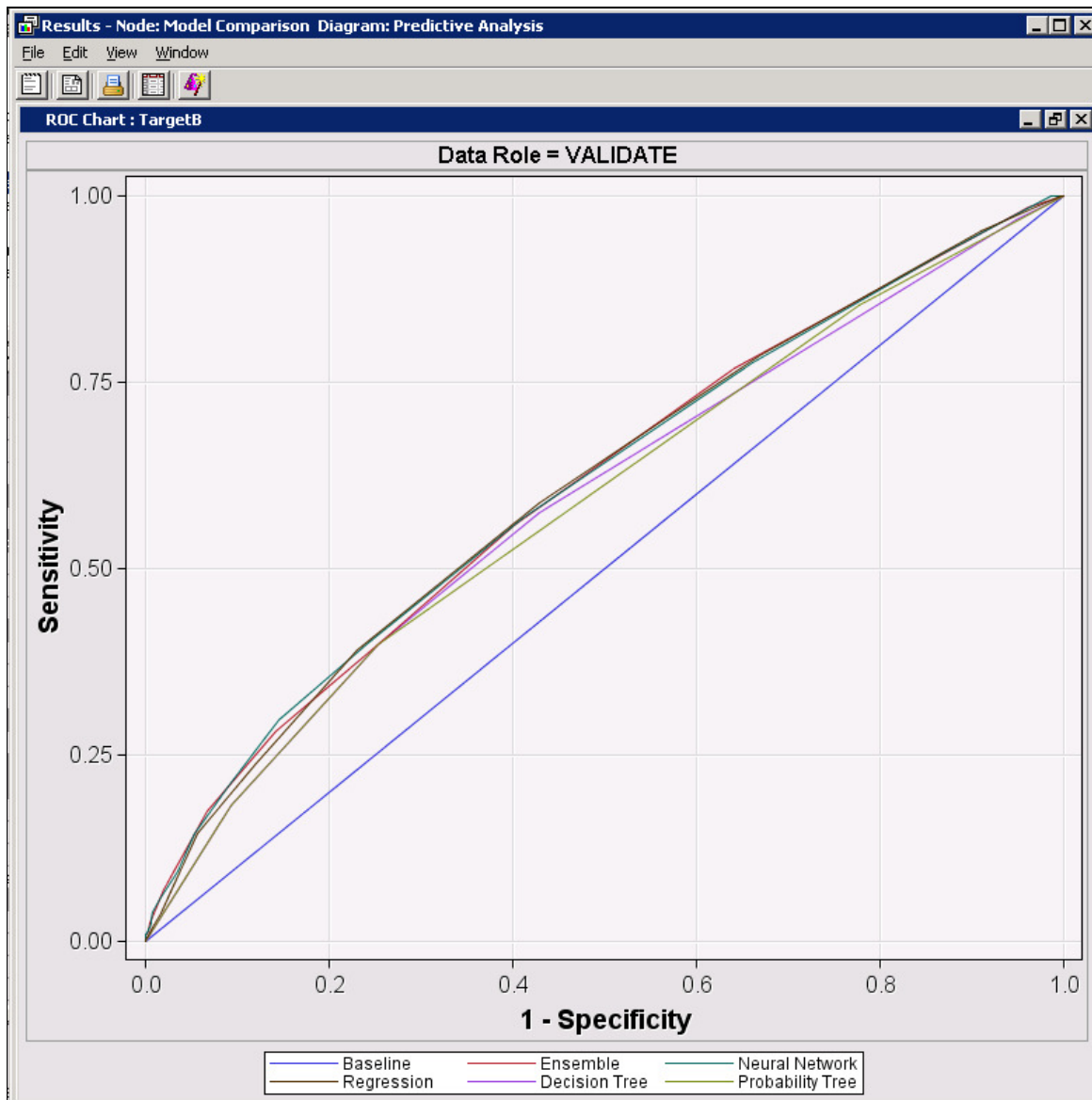
The completed process should appear as shown.

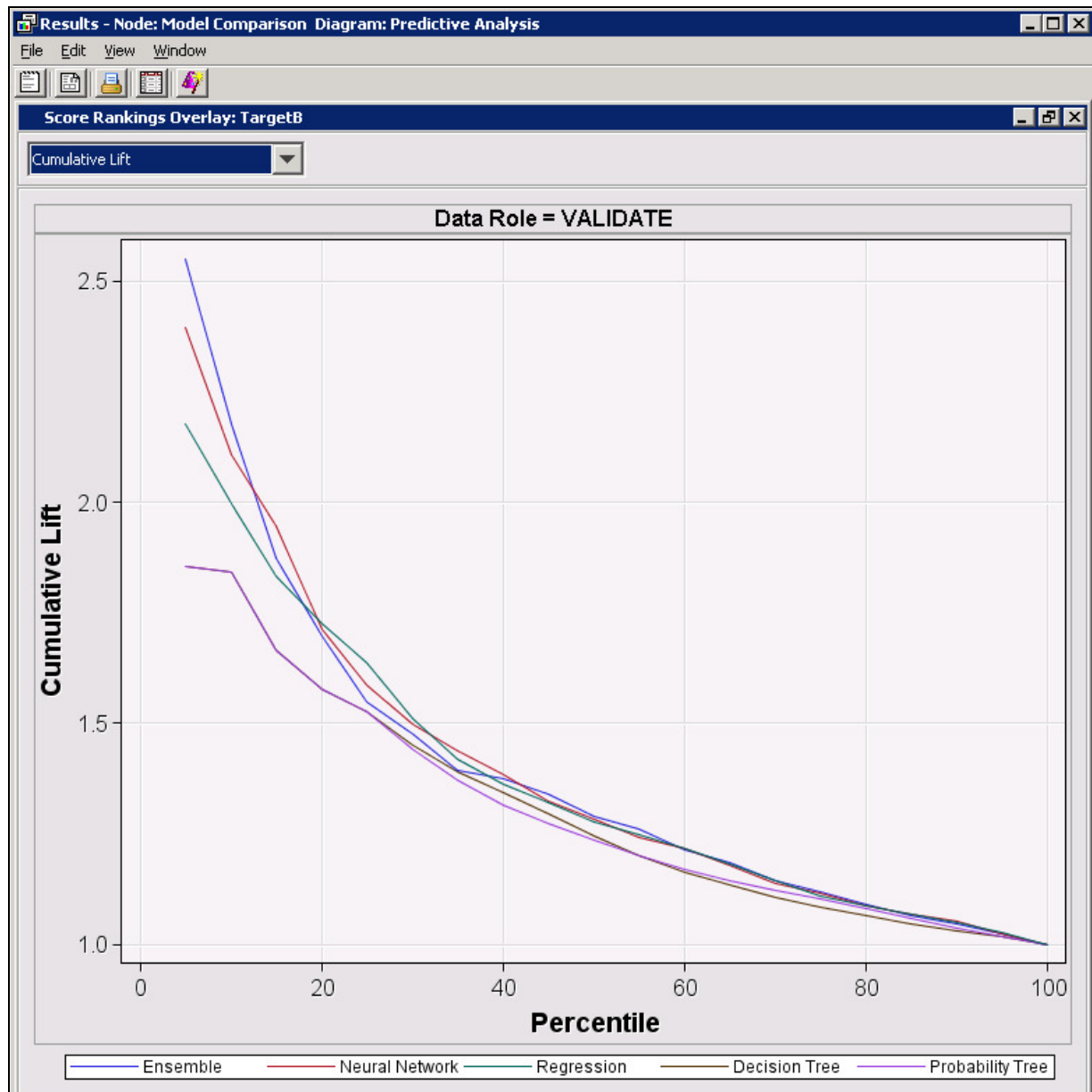


Ensemble Tool

The Ensemble node is not itself a model. It merely combines model predictions. For a categorical target variable, you can choose to combine models using the following functions:

- *Average* takes the average of the prediction estimates from the different models as the prediction from the Ensemble node. This is the default method.
- *Maximum* takes the maximum of the prediction estimates from the different models as the prediction from the Ensemble node.
- *Voting* uses one of two methods for prediction. The average method averages the prediction estimates from the models that decide the primary outcome and ignores any model that decides the secondary outcome. The proportion method ignores the prediction estimates and instead returns the proportion of models deciding the primary outcome.
- Run the Model Comparison node and view the results. This enables you to see how the ensemble model compares to the individual models.

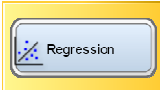
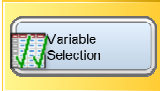
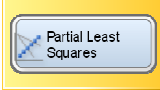
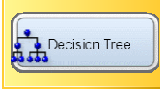




The ROC chart and Score Rankings plots show that the ensemble model is similar to the other models.

9.3 Variable Selection

Input Selection Alternatives

	Sequential selection
	Univariate + forward selection (R square) Tree-like selection (Chi-square)
	Variable Importance in the Projection (VIP)
	Split search selection

7

Removing redundant or irrelevant inputs from a training data set often reduces overfitting and improves prediction performance. Some prediction models (for example, neural networks) do not include methods for selecting inputs. For these models, input selection is done with a separate SAS Enterprise Miner tool.

Chapter 4 introduced sequential selection to find useful inputs for regression models. Later in Chapter 5, these inputs were used to compensate for the Neural Networks tool's lack of input selection capabilities. SAS Enterprise Miner features several additional methods to perform input selection for modeling tools inherently missing this capability. The following demonstrations illustrate the use of the Variable Selection, Partial Least Squares, and Decision Tree tools for selection of potentially useful inputs.



Using the Variable Selection Node

The Variable Selection tool provides a selection based on one of two criteria.

When you use the **R-squared variable selection criterion**, a two-step process is followed:

1. SAS Enterprise Miner computes the squared correlation for each variable and then assigns the Rejected role to those variables that have a value less than the squared correlation criterion. (The default is 0.005.)
2. SAS Enterprise Miner evaluates the remaining (not rejected) variables using a forward stepwise R-squared regression. Variables that have a stepwise R-squared improvement less than the threshold criterion (default = 0.0005) are assigned the Rejected role.

When you use the **chi-squared selection criterion**, variable selection is performed using binary splits for maximizing the chi-squared value of a 2x2 frequency table. The rows of the 2x2 table are defined by the (binary) target variable. The columns of the table are formed by a partition of the training data using a selected input.

Several partitions are considered for each input. For an L -level class input (binary, ordinal, or nominal), partitions are formed by comparing each input level separately to the remaining $L-1$ input levels, creating a collection of L possible data partitions. The partition with the highest chi-squared value is chosen as the input's best partition. For interval inputs, partitions are formed by dividing the input range into (a maximum of) 50 equal-length bins and splitting the data into two subsets at 1 of the 49 bin boundaries. The partition with the highest chi-squared statistic is chosen as the interval input's best partition. The partition/variable combination with the highest chi-squared statistic is used to split the data and the process is repeated within both subsets. The partitioning stops when no input has a chi-squared statistic in excess of a user-specified threshold. All variables not used in at least one partition are rejected.

The Variable Selection node's chi-squared approach is quite similar to a decision tree algorithm with its ability to detect nonlinear and nonadditive relationships between the inputs and the target. However, the method for handling categorical inputs makes it sensitive to spurious input/target correlations. Instead of the Variable Selection node's chi-squared setting, you might want to try the Decision Tree node, properly configured for input selection.

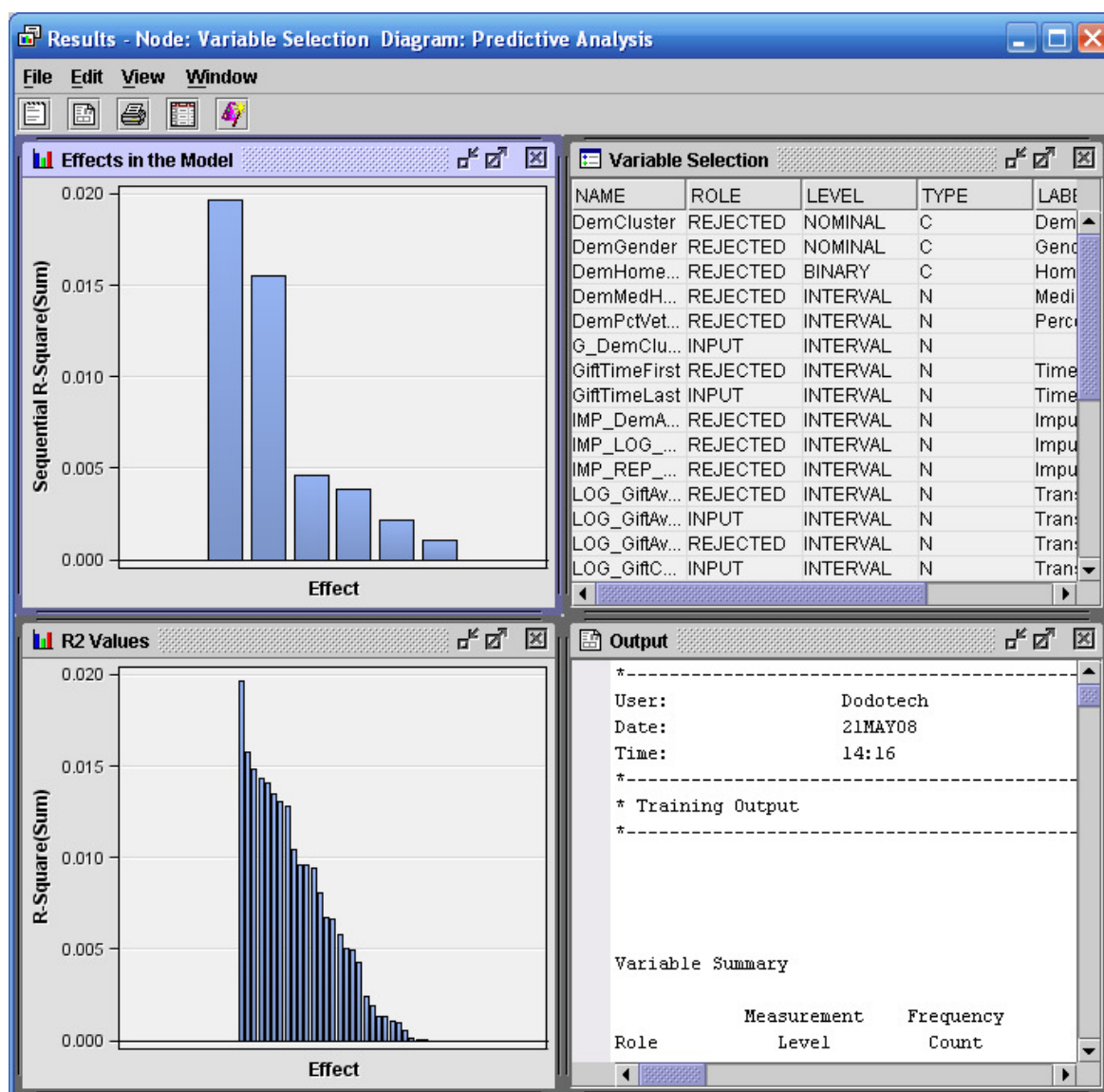
The following steps show how to use the Variable Selection tool with the R-squared setting.

1. Select the **Explore** tab.
2. Drag a **Variable Selection** tool into the diagram workspace.

Additional available options include the following:

- Use AOV16 Variables – When selected, this option requests SAS Enterprise Miner to bin interval variables into 16 equally spaced groups (AOV16). The AOV16 variables are created to help identify nonlinear relationships with the target. Bins with zero observations are eliminated, which means that an AOV16 variable can have fewer than 16 bins.
- Use Group Variables – When set to **Yes**, this option enables the number of levels of a class variable to be reduced, based on the relationship of the levels to the target variable.
- Use Interactions – When this option is selected, SAS Enterprise Miner evaluates two-way interactions for categorical inputs.

5. Run the Variable Selection node and view the results. The Results window opens.



6. Maximize the Variable Selection window.

7. Select the **ROLE** column heading to sort the variables by their assigned roles.
8. Select the **Reasons for Rejection** column heading.

Variable Selection					
Variable Name	ROLE	LEVEL	TYPE	Variable Label	Reasons for Rejection ▲
G_DemClu...	INPUT	NOMINAL	N		
GiftTimeLast	INPUT	INTERVAL	N	Time Since ...	
LOG_GiftAv...	INPUT	INTERVAL	N	Transforme...	
LOG_GiftC...	INPUT	INTERVAL	N	Transforme...	
LOG_GiftC...	INPUT	INTERVAL	N	Transforme...	
REP_Statu...	INPUT	NOMINAL	C	Replace:St...	
DemGender	REJECTED	NOMINAL	C	Gender	Varsel:Sma...
DemHome...	REJECTED	BINARY	C	Home Owner	Varsel:Sma...
DemMedH...	REJECTED	INTERVAL	N	Median Ho...	Varsel:Sma...
DemPctVet...	REJECTED	INTERVAL	N	Percent Vet...	Varsel:Sma...
GiftTimeFirst	REJECTED	INTERVAL	N	Time Since ...	Varsel:Sma...
IMP_DemA...	REJECTED	INTERVAL	N	Imputed: Age	Varsel:Sma...
IMP_LOG_...	REJECTED	INTERVAL	N	Imputed: Tr...	Varsel:Sma...
IMP_REP_...	REJECTED	INTERVAL	N	Imputed: R...	Varsel:Sma...
LOG_GiftAv...	REJECTED	INTERVAL	N	Transforme...	Varsel:Sma...
LOG_GiftAv...	REJECTED	INTERVAL	N	Transforme...	Varsel:Sma...
LOG_GiftC...	REJECTED	INTERVAL	N	Transforme...	Varsel:Sma...
LOG_GiftC...	REJECTED	INTERVAL	N	Transforme...	Varsel:Sma...
M_DemAge	REJECTED	BINARY	N	Imputation I...	Varsel:Sma...
M_LOG_Gif...	REJECTED	BINARY	N	Imputation I...	Varsel:Sma...
M_REP_De...	REJECTED	BINARY	N	Imputation I...	Varsel:Sma...
PromCnt12	REJECTED	INTERVAL	N	Promotion ...	Varsel:Sma...
PromCnt36	REJECTED	INTERVAL	N	Promotion ...	Varsel:Sma...
PromCntAll	REJECTED	INTERVAL	N	Promotion ...	Varsel:Sma...
PromCntCa...	REJECTED	INTERVAL	N	Promotion ...	Varsel:Sma...
PromCntCa...	REJECTED	INTERVAL	N	Promotion ...	Varsel:Sma...
PromCntCa...	REJECTED	INTERVAL	N	Promotion ...	Varsel:Sma...
StatusCatSt...	REJECTED	BINARY	N	Status Cate...	Varsel:Sma...
DemCluster	REJECTED	NOMINAL	C	Demograp...	Varsel:Sma...

The Variable Selection node finds that most inputs have insufficient target correlation to justify keeping them. You can try these inputs in a subsequent model or adjust the R-squared settings to be less severe. Notice the input **G_DemCluster** is a grouping of the original **DemCluster** input.



Binary targets generate notoriously low R-squared statistics. A more appropriate association measure might be the likelihood chi-squared statistic found in the Regression node.



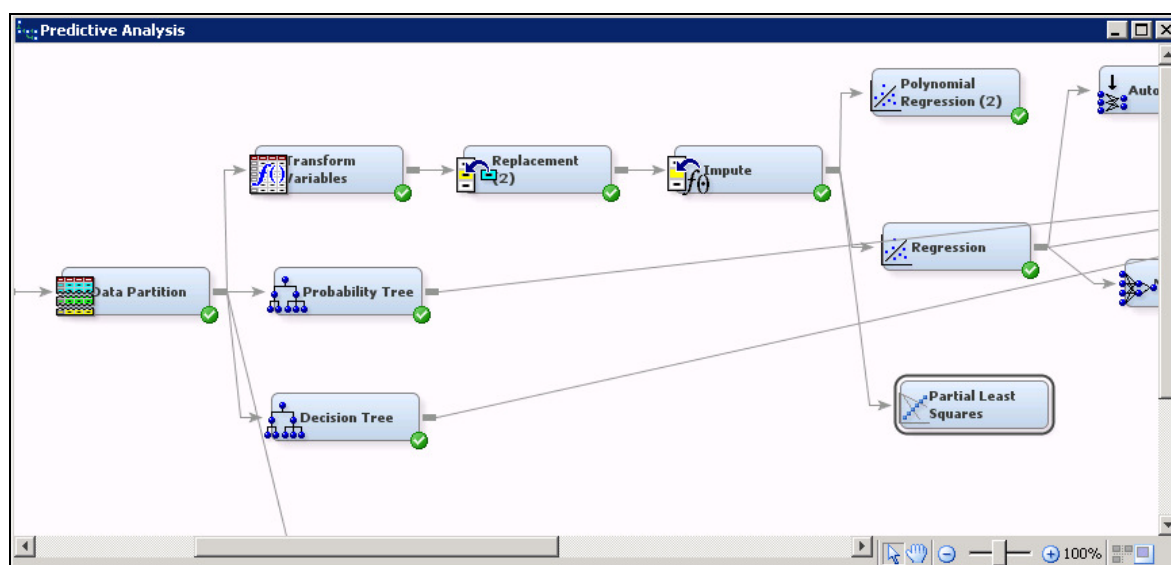
Using Partial Least Squares for Input Selection

Partial least squares (PLS) regression can be thought of as a merging of multiple and principal components regression. In multiple regression, the goal is to find linear combinations of the inputs that account for as much (linear) variation in the target as possible. In principal component regression, the goal is to find linear combinations of the inputs that account for as much (linear) variation in the input space as possible, and then use these linear combinations (called *principal component vectors*) as the inputs to a multiple regression model. In PLS regression, the goal is to have linear combinations of the inputs (called *latent vectors*) that account for variation in **both** the inputs and the target. The technique can extract a small number of latent vectors from a set of correlated inputs that correlate with the target.


A useful feature of the PLS procedure is the inclusion of an input importance metric named *variable importance in the projection* (VIP). VIP quantifies the relative importance of the original input variables to the latent vectors. A sufficiently small VIP for an input (less than, 0.8, by default) plus a small parameter estimate (less than 0.1, by default) permits removal of the input from the model.

The following steps demonstrate the use of the PLS tool for variable selection:

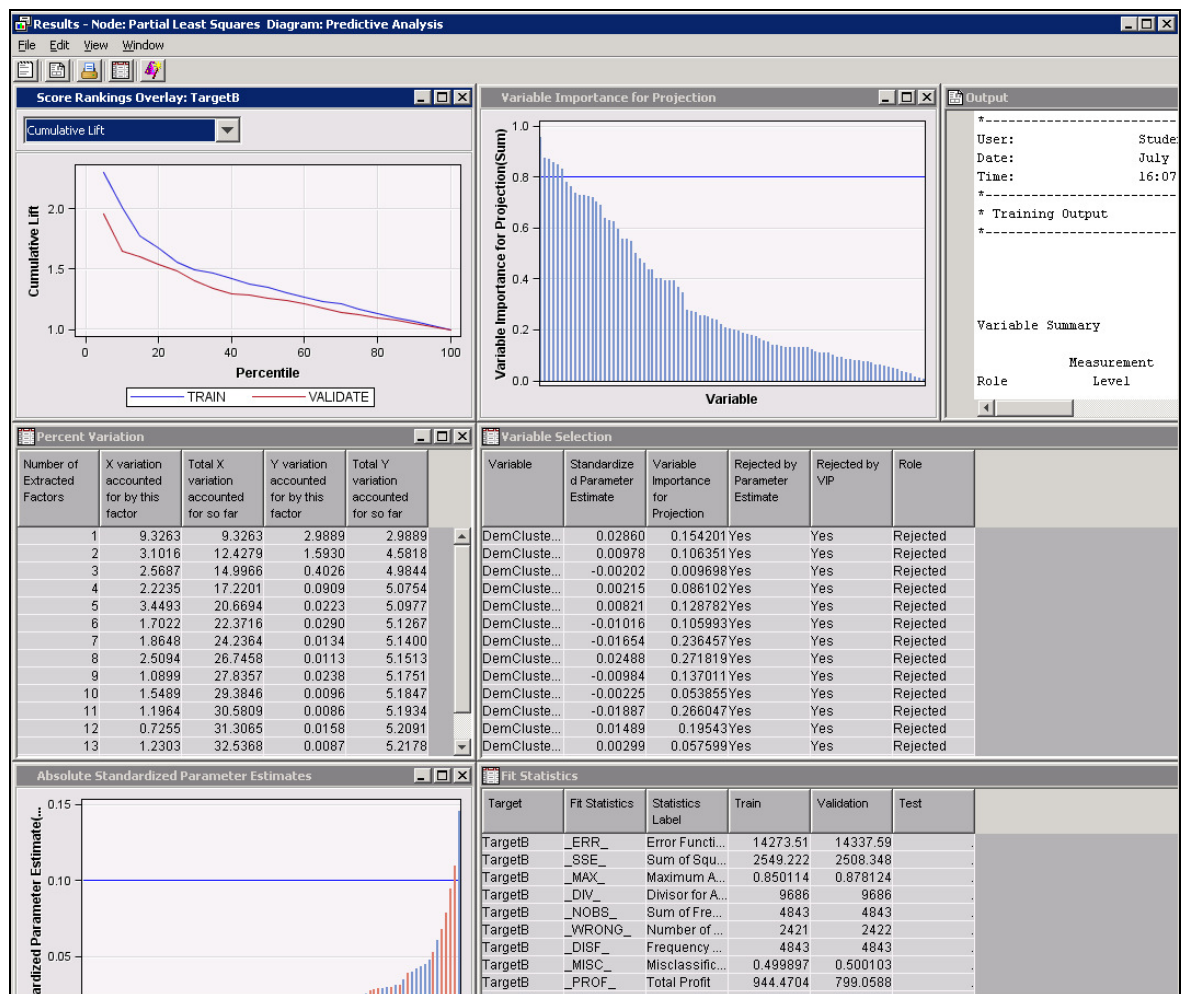
1. Select the **Model** tab.
2. Drag a **Partial Least Squares** tool into the diagram workspace.
3. Connect the **Impute** node to the **Partial Least Squares** node.



4. Select **Export Selected Variables** ⇒ **Yes**.

Train	
Variables	
<input checked="" type="checkbox"/> Modeling Techniques	
Regression Model	PLS
PLS Algorithm	NIPALS
Maximum Iteration	200
Epsilon	1.0E-12
<input checked="" type="checkbox"/> Number of Factors	
Default	Yes
Number of Factors	15
<input checked="" type="checkbox"/> Cross Validation	
CV Method	None
CV N Parameter	7
<input checked="" type="checkbox"/> Random CV Options	
Number of Iterations	10
Default No. of Test Obs.	Yes
No. of Test Obs.	100
Default Random Seed	Yes
Random Seed	1234
Score	
<input checked="" type="checkbox"/> Variable Selection	
Variable Selection Criterion	Both
Para. Est. Cutoff	0.1
VIP Cutoff	0.8
Export Selected Variables	Yes
Hide Rejected Variables	No

5. Run the Partial Least Squares node and view the results.

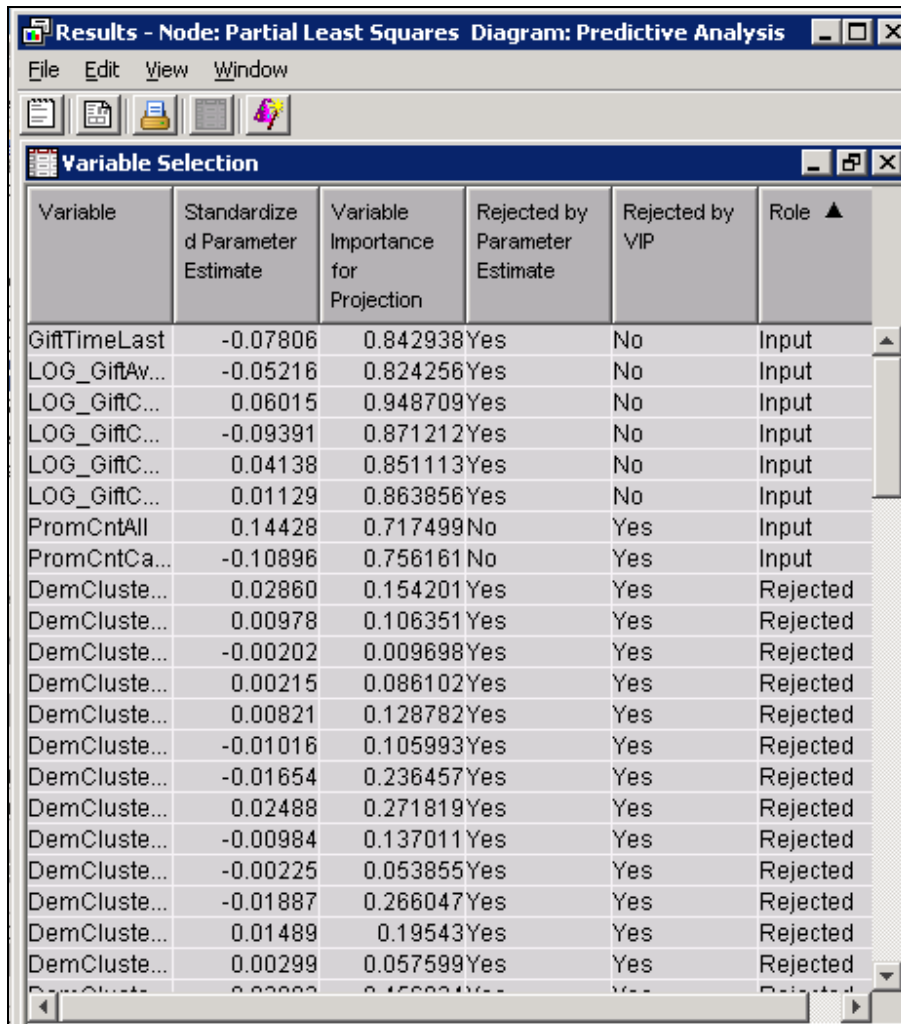


6. Maximize the Percent Variation window.

Percent Variation				
Number of Extracted Factors	X variation accounted for by this factor	Total X variation accounted for so far	Y variation accounted for by this factor	Total Y variation accounted for so far
1	9.3263	9.3263	2.9889	2.9889
2	3.1016	12.4279	1.5930	4.5818
3	2.5687	14.9966	0.4026	4.9844
4	2.2235	17.2201	0.0909	5.0754
5	3.4493	20.6694	0.0223	5.0977
6	1.7022	22.3716	0.0290	5.1267
7	1.8648	24.2364	0.0134	5.1400
8	2.5094	26.7458	0.0113	5.1513
9	1.0899	27.8357	0.0238	5.1751
10	1.5489	29.3846	0.0096	5.1847
11	1.1964	30.5809	0.0086	5.1934
12	0.7255	31.3065	0.0158	5.2091
13	1.2303	32.5368	0.0087	5.2178
14	1.3137	33.8505	0.0065	5.2243
15	0.8789	34.7294	0.0089	5.2332

By default, the Partial Least Squares tool extracts 15 latent vectors or factors from the training data set. These factors account for 35% and 5.2% of the variation in the inputs and target, respectively.

7. Maximize the Variable Selection window.
8. Select the **Role** column heading to sort the table by variable role.



Variable	Standardized Parameter Estimate	Variable Importance for Projection	Rejected by Parameter Estimate	Rejected by VIP	Role ▲
GiftTimeLast	-0.07806	0.842938	Yes	No	Input
LOG_GiftAv...	-0.05216	0.824256	Yes	No	Input
LOG_GiftC...	0.06015	0.948709	Yes	No	Input
LOG_GiftC...	-0.09391	0.871212	Yes	No	Input
LOG_GiftC...	0.04138	0.851113	Yes	No	Input
LOG_GiftC...	0.01129	0.863856	Yes	No	Input
PromCntAll	0.14428	0.717499	No	Yes	Input
PromCntCa...	-0.10896	0.756161	No	Yes	Input
DemCluste...	0.02860	0.154201	Yes	Yes	Rejected
DemCluste...	0.00978	0.106351	Yes	Yes	Rejected
DemCluste...	-0.00202	0.009698	Yes	Yes	Rejected
DemCluste...	0.00215	0.086102	Yes	Yes	Rejected
DemCluste...	0.00821	0.128782	Yes	Yes	Rejected
DemCluste...	-0.01016	0.105993	Yes	Yes	Rejected
DemCluste...	-0.01654	0.236457	Yes	Yes	Rejected
DemCluste...	0.02488	0.271819	Yes	Yes	Rejected
DemCluste...	-0.00984	0.137011	Yes	Yes	Rejected
DemCluste...	-0.00225	0.053855	Yes	Yes	Rejected
DemCluste...	-0.01887	0.266047	Yes	Yes	Rejected
DemCluste...	0.01489	0.19543	Yes	Yes	Rejected
DemCluste...	0.00299	0.057599	Yes	Yes	Rejected
DemCluste...	0.00000	0.00000	Yes	Yes	Rejected

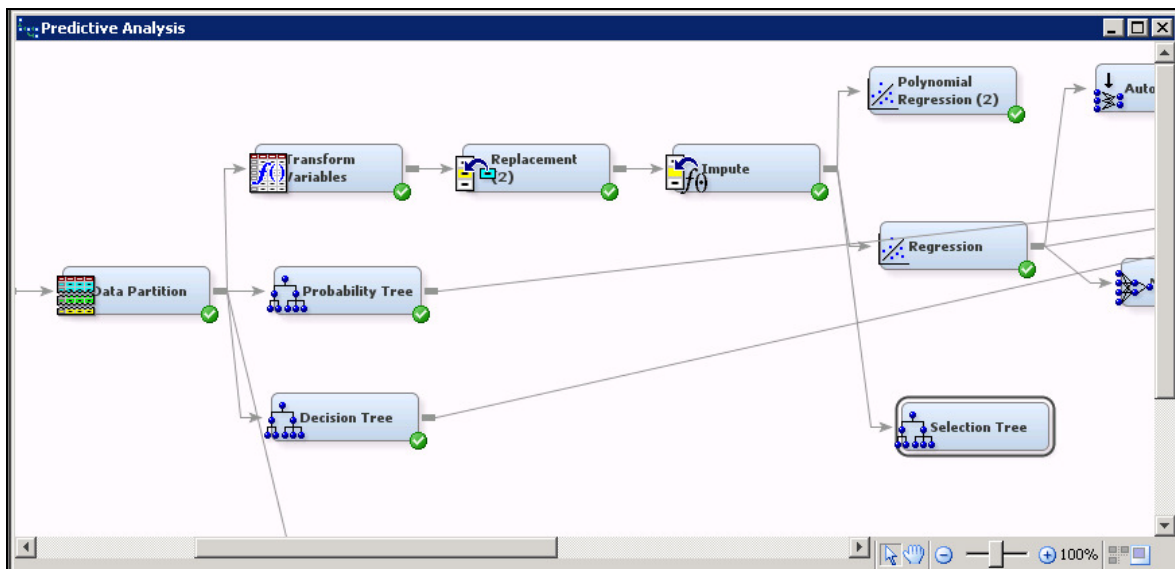
The majority of the selected inputs relate to donation count, promotion count, time since donation, and amount of donation.



Using the Decision Tree Node for Input Selection

Decision trees can be used to select inputs for flexible predictive models. They have an advantage over using a standard regression model or the Variable Selection tool's R-squared method if the inputs' relationships to the target are nonlinear or nonadditive.

1. Connect a **Decision Tree** node to the **Impute** node. Rename the Decision Tree node **Selection Tree**.



You can use the Tree node with default settings to select inputs. However, this tends to select too few inputs for a subsequent model. Two changes to the Tree defaults result in more inputs being selected. Generally, when you use trees to select inputs for flexible models, it is better to err on the side of too many inputs rather than too few. The model's complexity optimization method can usually compensate for the extra inputs.



The following changes to the defaults act independently. You can experiment to discover which method generalizes best with your data.

2. Type **1** as the Number of Surrogate Rules value.
3. Select **Subtree** ⇒ **Method** ⇒ **Largest**.

Train	
Variables	...
Interactive	...
Splitting Rule	
Criterion	Default
Significance Level	0.2
Missing Values	Use in search
Use Input Once	No
Maximum Branch	2
Maximum Depth	6
Minimum Categorical Size	5
Node	
Leaf Size	5
Number of Rules	5
Number of Surrogate Rules	1
Split Size	
Split Search	
Exhaustive	5000
Node Sample	20000
Subtree	
Method	Largest
Number of Leaves	1
Assessment Measure	Decision
Assessment Fraction	0.25
Cross Validation	

Changing the number of surrogates enables inclusion of surrogate splits in the variable selection process. By definition, surrogate inputs are typically correlated with the selected split input. While it is usually a bad practice to include redundant inputs in predictive models, many flexible models tolerate some degree of input redundancy. The advantage of including surrogates in the variable selection is to enable inclusion of inputs that do not appear in the tree explicitly but are still important predictors of the target.

Changing the Subtree method causes the tree algorithm to not prune the tree. As with adding surrogate splits to the variable selection process, it tends to add (possibly irrelevant) inputs to the selection list.

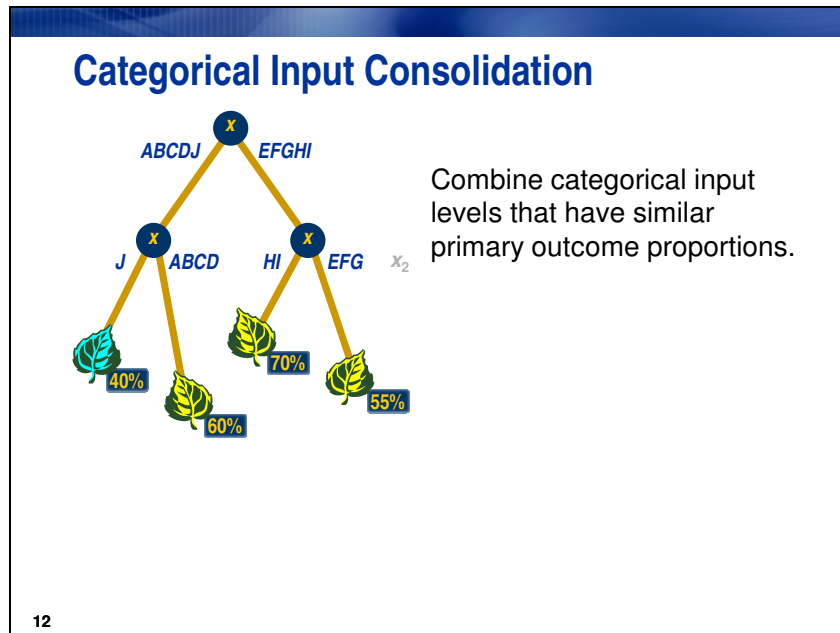
4. Run the Selection Tree node and view the results.

5. View lines 60 through 73 of the Output window.

Obs	NAME	LABEL	NRULES	NSURROGATES	IMPORTANCE	VIMPORTANCE	RATIO
1	LOG_GiftCnt36	Transformed: Gift Count 36 Months	1	0	1.00000	1.00000	1.00000
2	LOG_GiftCntCard36	Transformed: Gift Count Card 36 Months	0	1	0.89945	0.89945	1.00000
3	PromCnt12	Promotion Count 12 Months	2	1	0.66372	0.36177	0.54506
4	DemCluster	Demographic Cluster	3	2	0.65908	0.09777	0.14834
5	DemMedHomeValue	Median Home Value Region	2	1	0.64347	0.10861	0.16879
6	LOG_GiftAvg36	Transformed: Gift Amount Average 36 Months	0	3	0.62730	0.70944	1.13095
7	LOG_GiftAvgLast	Transformed: Gift Amount Last	1	0	0.52413	0.67188	1.28189
8	GiftTimeLast	Time Since Last Gift	1	0	0.48092	0.44530	0.92595
9	IMP_LOG_GiftAvgCard36	Imputed: Transformed: Gift Amount Average Card 36 Months	2	0	0.40002	0.32098	0.80240
10	LOG_GiftAvgAll	Transformed: Gift Amount Average All Months	1	0	0.39360	0.00000	0.00000
11	PromCnt36	Promotion Count 36 Months	0	1	0.37819	0.00000	0.00000
12	PromCntCard12	Promotion Count Card 12 Months	0	1	0.36544	0.00000	0.00000
13	DemHomeOwner	Home Owner	1	0	0.30767	0.00000	0.00000
14	StatusCatStarAll	Status Category Star All Months	1	0	0.29456	0.06599	0.22403
15	IMP_REP_DemMedIncome	Imputed: Replacement: DemMedIncome	0	1	0.26645	0.00000	0.00000
16	LOG_GiftCntCardAll	Transformed: Gift Count Card All Months	0	1	0.26255	0.05882	0.22403
17	DemPctVeterans	Percent Veterans Region	0	1	0.02873	0.00000	0.00000

The IMPORTANCE column quantifies approximately how much of the overall variability in the target each input explains. The values are normalized by the amount of variability explained by the input with the highest importance. The variable importance definition not only considers inputs selected as split variables, but it also accounts for surrogate inputs (if a positive number of surrogate rules is selected in the Properties panel). For example, the second most important input (**LOG_GiftCntCard36**) accounts for almost the same variability as the most important input (**LOG_GiftCnt36**) even though it does not explicitly appear in the tree.

9.4 Categorical Input Consolidation



Categorical inputs pose a major problem for parametric predictive models such as regressions and neural networks. Because each categorical level must be coded by an indicator variable, a single input can account for more model parameters than all other inputs combined.

Decision trees, on the other hand, thrive on categorical inputs. They can easily group the distinct levels of the categorical variable together and produce good predictions.

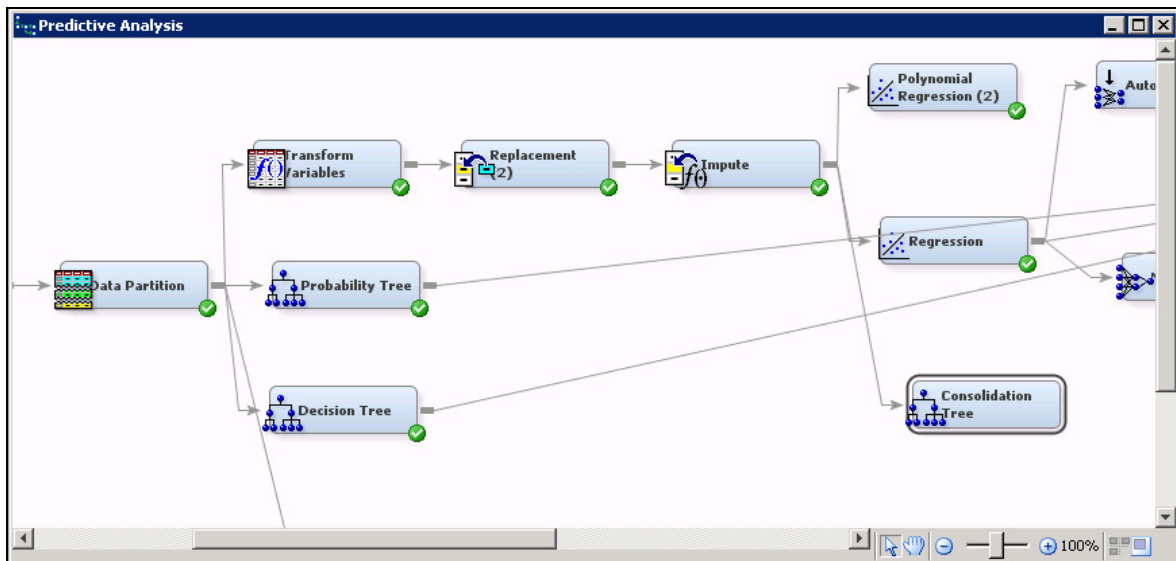
It is possible to use a decision tree to consolidate the levels of a categorical input. You simply build a decision tree using the categorical variable of interest as the sole modeling input. The split search algorithm then groups input levels with similar primary outcome proportions. The IDs for each leaf replace the original levels of the input.



Consolidating Categorical Inputs

Follow these steps to use a tree model to group categorical input levels and create useful inputs for regression and neural network models.

1. Connect a **Decision Tree** node to the **Impute** node, and rename the Decision Tree node **Consolidation Tree**.

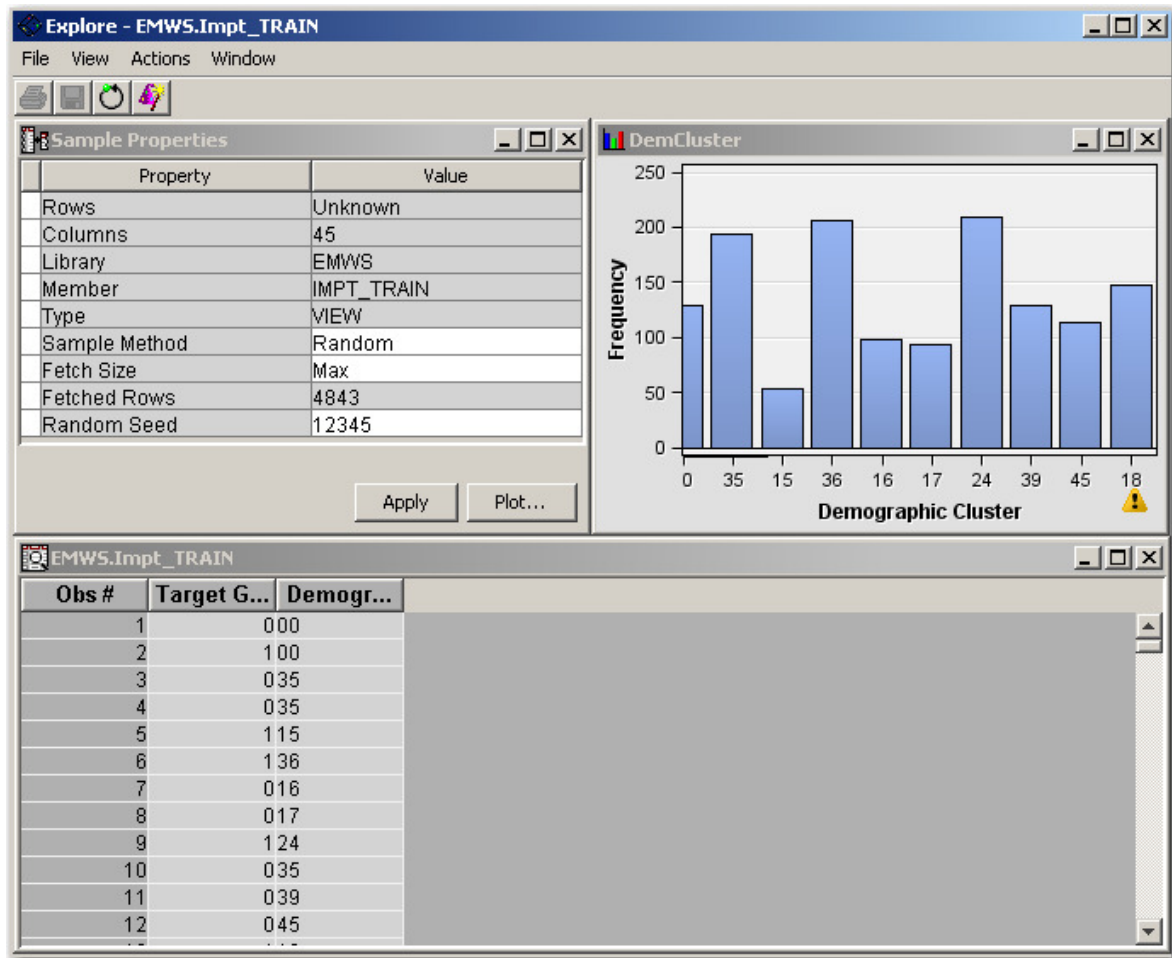


The Consolidation Tree node is used to group the levels of **DemCluster**, a categorical input with more than 50 levels.

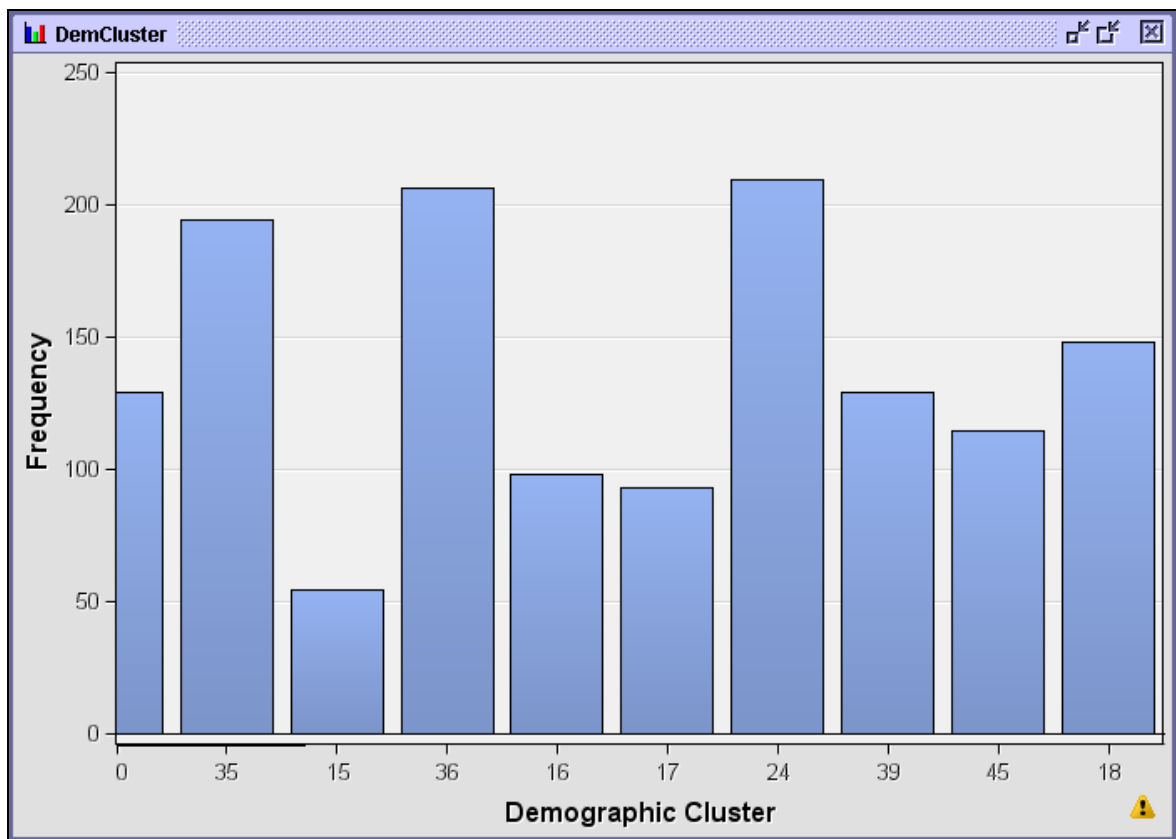
You use a tree model to group these levels based on their associations with **TargetB**. From this grouping, a new modeling input is created. You can use this input in place of **DemCluster** in a regression or other model. In this way, the predictive prowess of **DemCluster** is incorporated into a model without the plethora of parameters needed to encode the original.

The grouping can be done autonomously by simply running the Decision Tree node, or interactively by using the node's interactive training features. You use the automatic method here.

2. Select **Variables...** from the Consolidation Tree Properties panel.
3. Select **DemCluster** ⇒ **Explore...**. The Explore window opens.



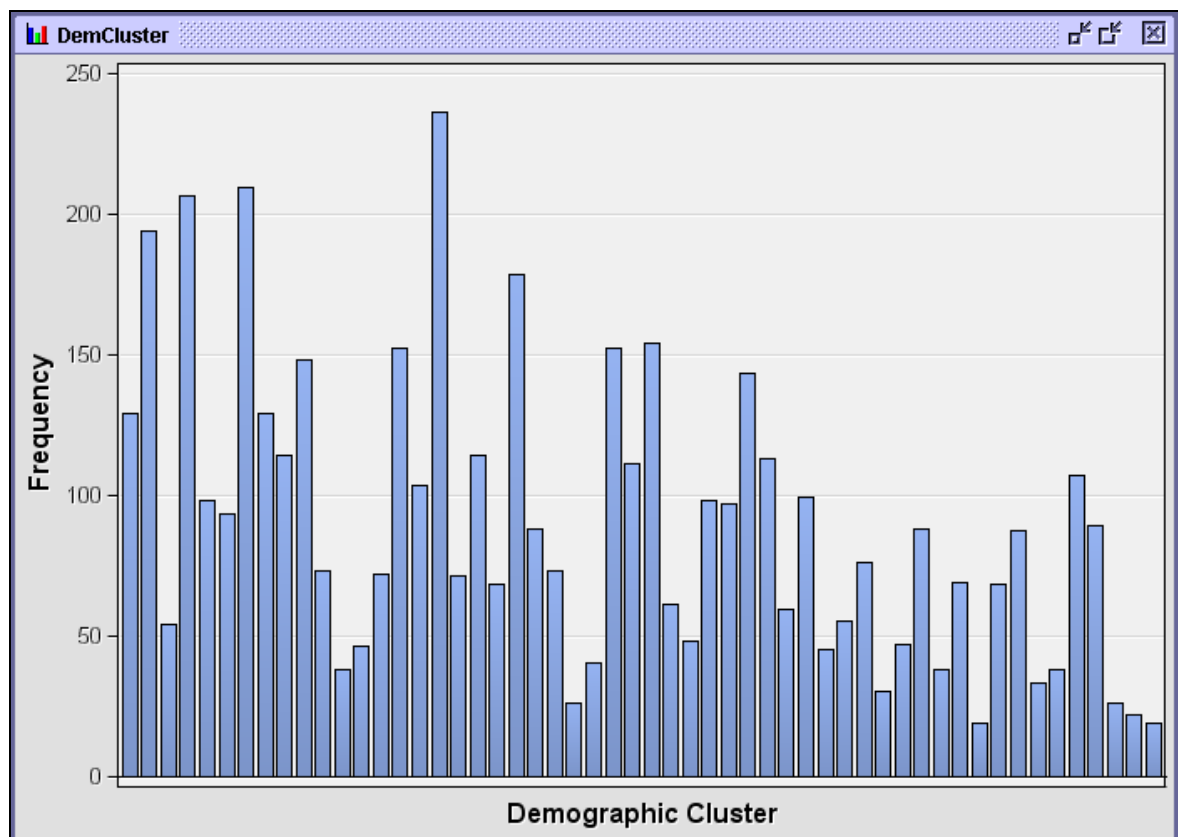
4. Maximize the DemCluster histogram.



The **DemCluster** input has more than 50 levels, but you can see the distribution of only a few of these levels.

5. Click  in the lower right corner of the DemCluster window.

The histogram expands to show the relative frequencies of each level of **DemCluster**.



The histogram reveals input levels with low case counts. This can detrimentally affect the performance of most models.

6. Close the Explore window.

8. Select Use \Rightarrow Yes for **DemCluster** and **TargetB**.

Variables - Tree4

(none) ☐ not Equal to ☐ ☐

Columns: ☐ Label ☐ Mining ☒ Basic ☐ Statistics

Name	Use	Report	Role	Level	Type	Format	Informat	Length
DemCluster	Yes	No	Input	Nominal	Character			2
TargetB	Yes	No	Target	Binary	Numeric			8
DemGender	No	No	Input	Nominal	Character			3
DemHomeOwn	No	No	Input	Binary	Character			3
DemMedHom	No	No	Input	Interval	Numeric	DOLLAR11.0		8
DemMedIncor	No	No	Rejected	Interval	Numeric	DOLLAR11.0		8
DemPctVeteran	No	No	Input	Interval	Numeric			8
GiftTimeFirst	No	No	Input	Interval	Numeric			8
GiftTimeLast	No	No	Input	Interval	Numeric			8
IMP_DemAge	No	No	Input	Interval	Numeric			8
IMP_LOG_Gift	No	No	Input	Interval	Numeric			8
IMP_REP_Dem	No	No	Input	Interval	Numeric			8
LOG_GiftAvg3	No	No	Input	Interval	Numeric			8
LOG_GiftAvgA	No	No	Input	Interval	Numeric			8
LOG_GiftAvgL	No	No	Input	Interval	Numeric			8
LOG_GiftCnt3	No	No	Input	Interval	Numeric			8
LOG_GiftCntA	No	No	Input	Interval	Numeric			8
LOG_GiftCntC	No	No	Input	Interval	Numeric			8
LOG_GiftCntC	No	No	Input	Interval	Numeric			8
M_DemAge	No	No	Input	Binary	Numeric			8
M_LOG_GiftAv	No	No	Input	Binary	Numeric			8
M_REP_Dem	No	No	Input	Binary	Numeric			8
PromCnt12	No	No	Input	Interval	Numeric			8
PromCnt36	No	No	Input	Interval	Numeric			8
PromCntAll	No	No	Input	Interval	Numeric			8
PromCntCard	No	No	Input	Interval	Numeric			8
PromCntCard	No	No	Input	Interval	Numeric			8
PromCntCard	No	No	Input	Interval	Numeric			8
REP_StatusC	No	No	Input	Nominal	Character			5
StatusCat96N	No	No	Rejected	Nominal	Character			5

Explore... Update Path OK Cancel

After sorting on the column **Use**, the Variables window should appear as shown.

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

10. Make these changes in the **Train** property group.

- a. Select **Assessment Measure** \Rightarrow **Average Squared Error**. This optimizes the tree for prediction estimates.
- b. Select **Bonferroni Adjustment** \Rightarrow **No**.

When you evaluate a potential split, the Decision Tree tool applies, by default, a Bonferroni adjustment to the splits logworth. The adjustment penalizes the logworth of potential **DemCluster** splits. The penalty is calculated as the log of the number of partitions of **DemCluster** levels split into two groups, or $\log_{10}(2^{L-1} - 1)$. With 54 distinct levels, the penalty is quite large. It is also, in this case, quite unnecessary. The penalty avoids favoring inputs with many possible splits. Here you are building a tree with only one input. It is impossible to favor this input over others because there are no other inputs.

11. Make these changes in the Score property group.

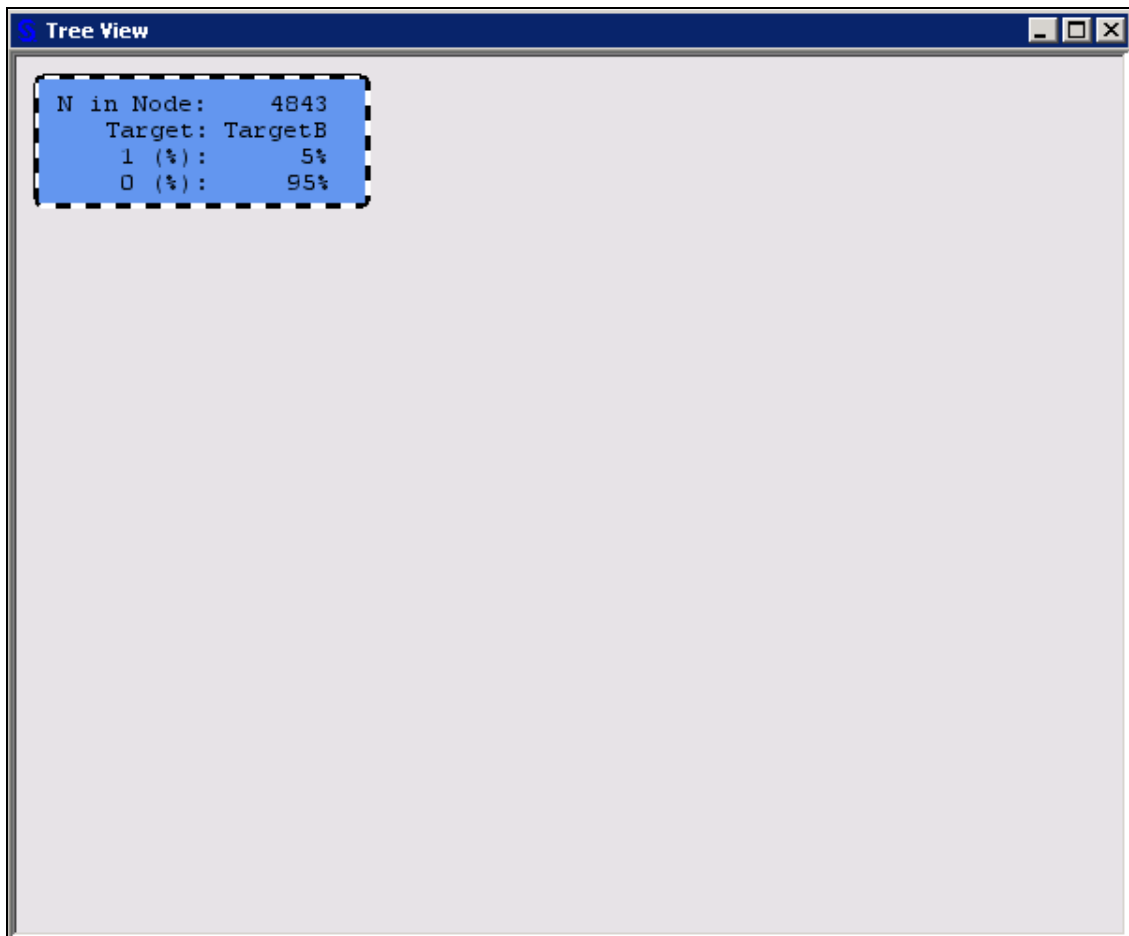
- a. Select **Variable Selection** \Rightarrow **No**. This prevents the decision tree from rejecting inputs in subsequent nodes.
- b. Select **Leaf Role** \Rightarrow **Input**. This adds a new input (**_NODE_**) to the training data.

12. Now use the Interactive Tree tool to cluster **DemCluster** values into related groups.

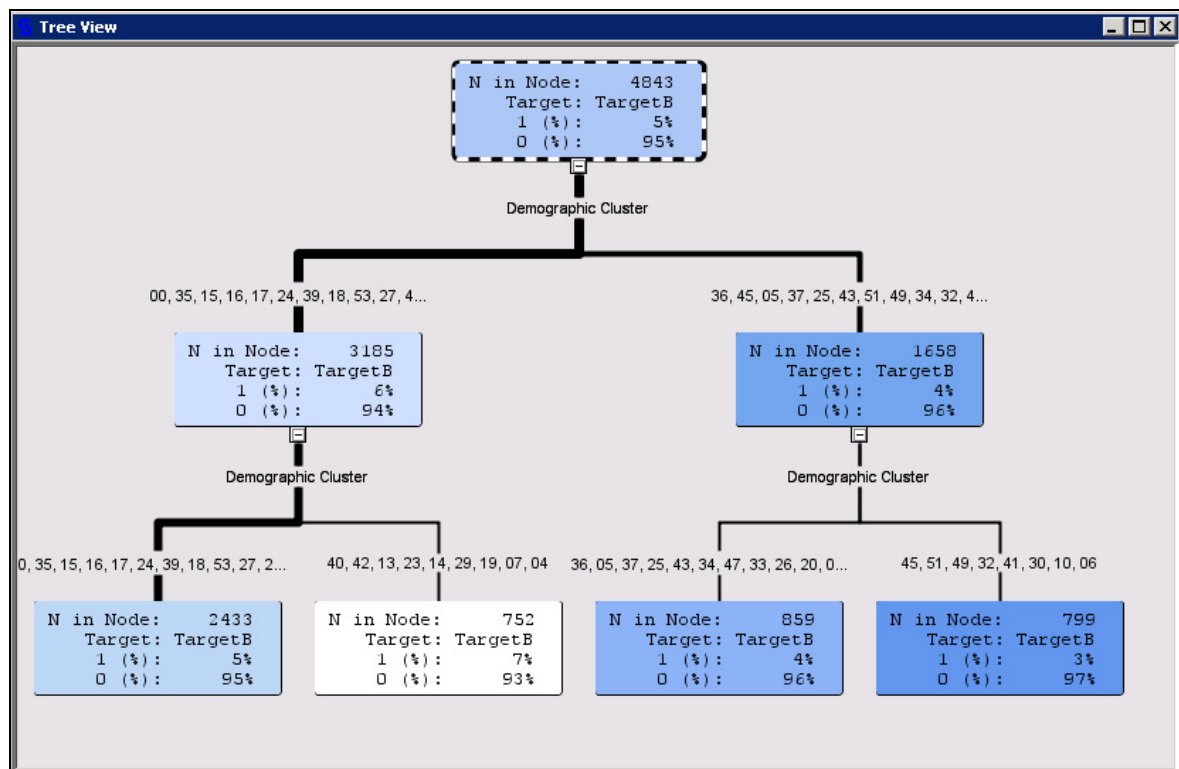
- a. Select **Interactive...** from the Decision Tree Properties panel.

Property	Value
Node ID	Tree4
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Interactive	...
Use Frozen Tree	No
Use Multiple Targets	No

The SAS Enterprise Miner Tree Desktop Application opens.



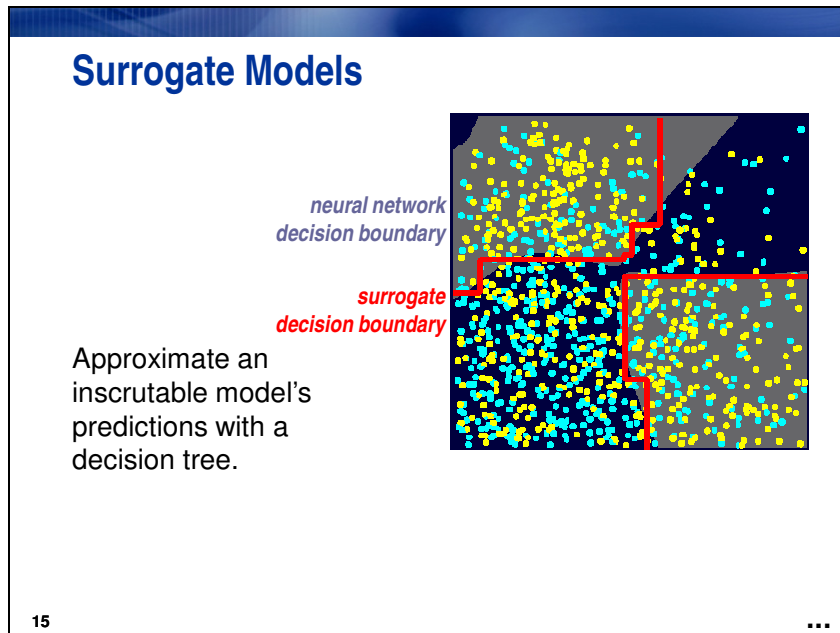
- b. Right-click on the root node and select **Train Node** from the option menu.



The levels of **DemCluster** are partitioned into four groups corresponding to the four leaves of the tree.

An input named `_NODE_` is added to the training data. You can use the Transform Variables tool to rename `_NODE_` to a more descriptive value. You can use the Replacement tool to change the level names.

9.5 Surrogate Models



The usual criticism of neural networks and similar flexible models is the difficulty in understanding the predictions.

This criticism stems from the complex parameterizations found in the model. While it is true that little insight can be gained by analyzing the actual parameters of the model, much can be gained by analyzing the resulting prediction decisions.

A profit matrix or other mechanism for generating a decision threshold defines regions in the input space corresponding to the primary decision. The idea behind a surrogate model is building an easy-to-understand model that describes this region. In this way, characteristics used by the neural network to make the primary decision can be understood, even if the neural network model itself cannot be.



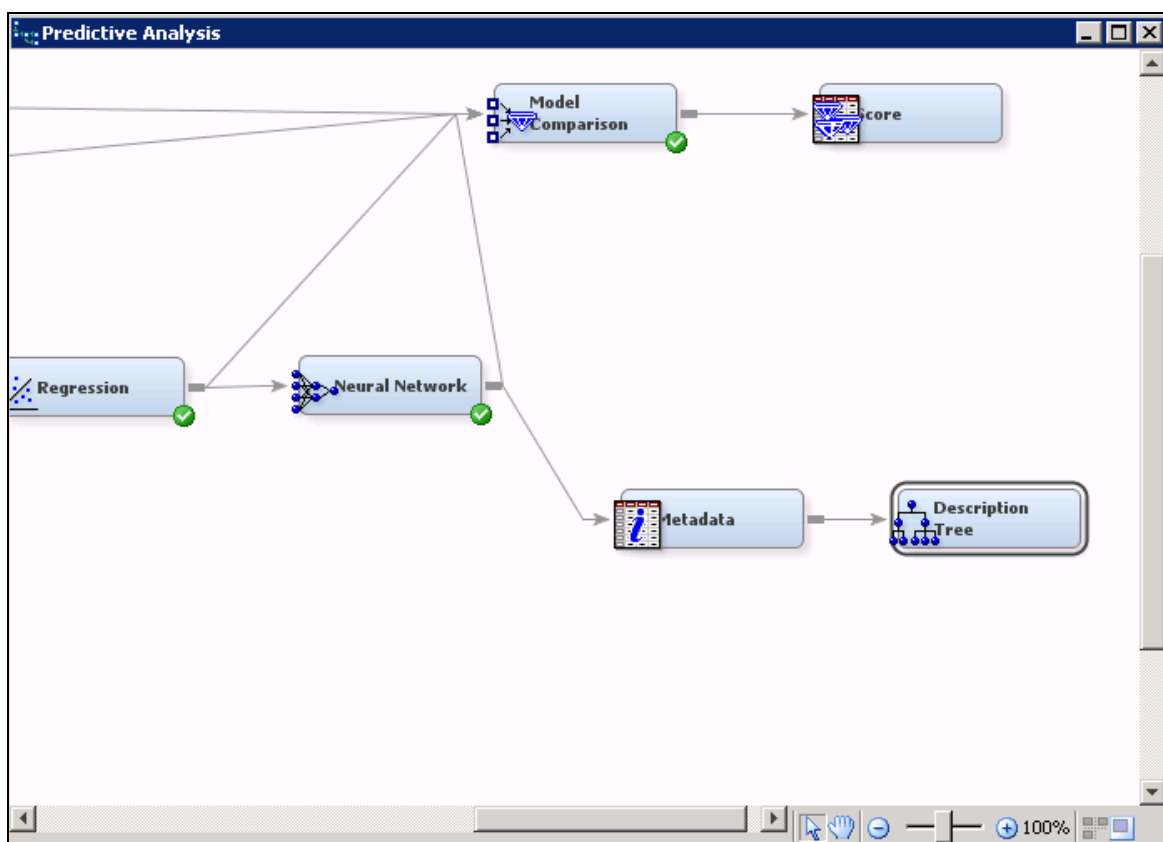
Describing Decision Segments with Surrogate Models

In this demonstration, a decision tree is used to isolate cases found solicitation-worthy by a neural network model. Using the decision tree, characteristics of solicit-decision donors can be understood even if the model making the decision is a mystery.

Setting Up the Diagram

You need to attach two nodes to the modeling node that you want to study.

1. Select the **Utilities** tab.
2. Drag a **Metadata** tool into the diagram workspace.
3. Connect the **Neural Network** node to the **Metadata** node.
4. Select the **Model** tab.
5. Drag a **Decision Tree** tool into the diagram workspace.
6. Connect the **Decision Tree** tool to the **Metadata** node.
7. Rename the Decision Tree node **Description Tree**.



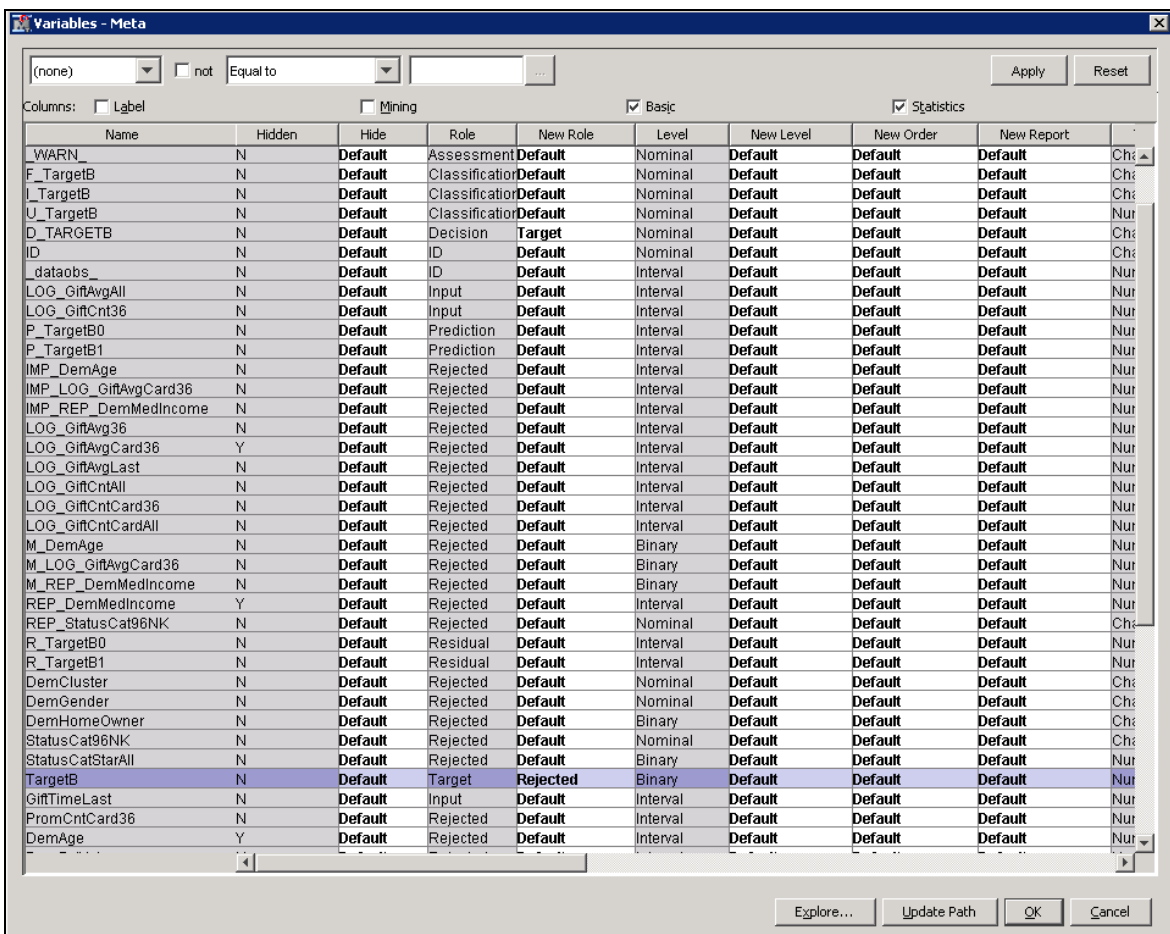
Changing the Metadata

You must change the focus of the analysis from **TargetB** to the variable describing the neural network decisions. SAS Enterprise Miner automatically creates this variable in the training and validation data sets exported from a modeling node. The variable's name is **D_target**, where **target** is the name of the original target variable.

 **D_TARGETB** is automatically created on the definition of decision data only.

The following steps change the target variable from **TargetB** to **D_TARGETB**:

1. Scroll down to the Variables section in the Metadata node properties. Select **Train...**. The Variables window opens.



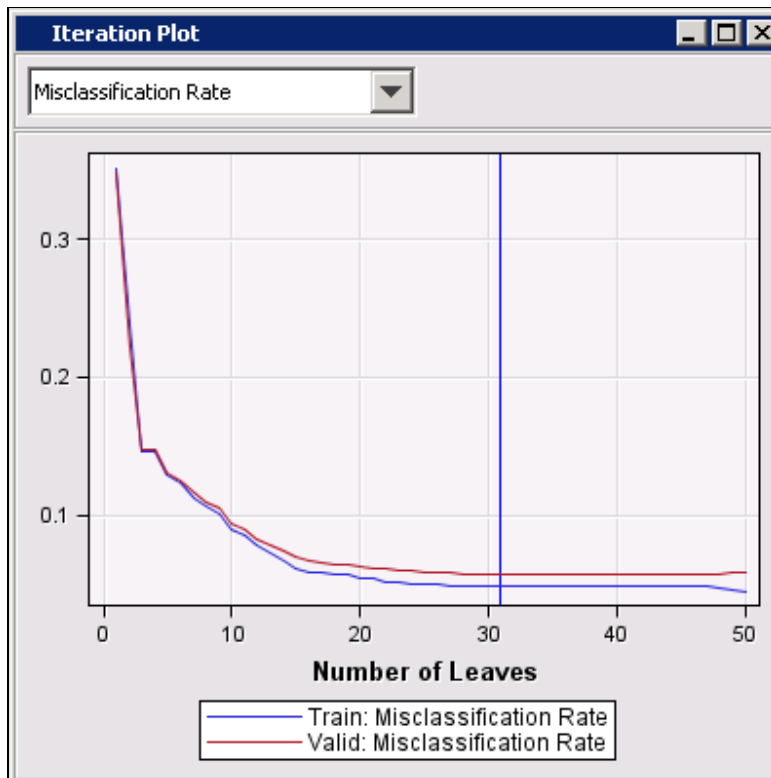
Name	Hidden	Hide	Role	New Role	Level	New Level	New Order	New Report	
WARN	N	Default	Assessment	Default	Nominal	Default	Default	Default	Chi
F_TargetB	N	Default	Classification	Default	Nominal	Default	Default	Default	Chi
I_TargetB	N	Default	Classification	Default	Nominal	Default	Default	Default	Chi
U_TargetB	N	Default	Classification	Default	Nominal	Default	Default	Default	Nur
D_TARGETB	N	Default	Decision	Target	Nominal	Default	Default	Default	Chi
ID	N	Default	ID	Default	Nominal	Default	Default	Default	Chi
_dataobs	N	Default	ID	Default	Interval	Default	Default	Default	Nur
LOG_GiftAvgAll	N	Default	Input	Default	Interval	Default	Default	Default	Nur
LOG_GiftCnt36	N	Default	Input	Default	Interval	Default	Default	Default	Nur
P_TargetB0	N	Default	Prediction	Default	Interval	Default	Default	Default	Nur
P_TargetB1	N	Default	Prediction	Default	Interval	Default	Default	Default	Nur
IMP_DemAge	N	Default	Rejected	Default	Interval	Default	Default	Default	Nur
IMP_LOG_GiftAvgCard36	N	Default	Rejected	Default	Interval	Default	Default	Default	Nur
IMP_REP_DemMedIncome	N	Default	Rejected	Default	Interval	Default	Default	Default	Nur
LOG_GiftAvg36	N	Default	Rejected	Default	Interval	Default	Default	Default	Nur
LOG_GiftAvgCard36	Y	Default	Rejected	Default	Interval	Default	Default	Default	Nur
LOG_GiftAvgLast	N	Default	Rejected	Default	Interval	Default	Default	Default	Nur
LOG_GiftCntAll	N	Default	Rejected	Default	Interval	Default	Default	Default	Nur
LOG_GiftCntCard36	N	Default	Rejected	Default	Interval	Default	Default	Default	Nur
LOG_GiftCntCardAll	N	Default	Rejected	Default	Interval	Default	Default	Default	Nur
M_DemAge	N	Default	Rejected	Default	Binary	Default	Default	Default	Nur
M_LOG_GiftAvgCard36	N	Default	Rejected	Default	Binary	Default	Default	Default	Nur
M_REP_DemMedIncome	N	Default	Rejected	Default	Binary	Default	Default	Default	Nur
REP_DemMedIncome	Y	Default	Rejected	Default	Interval	Default	Default	Default	Nur
REP_StatusCat96NK	N	Default	Rejected	Default	Nominal	Default	Default	Default	Chi
R_TargetB0	N	Default	Residual	Default	Interval	Default	Default	Default	Nur
R_TargetB1	N	Default	Residual	Default	Interval	Default	Default	Default	Nur
DemCluster	N	Default	Rejected	Default	Nominal	Default	Default	Default	Chi
DemGender	N	Default	Rejected	Default	Nominal	Default	Default	Default	Chi
DemHomeOwner	N	Default	Rejected	Default	Binary	Default	Default	Default	Chi
StatusCat96NK	N	Default	Rejected	Default	Nominal	Default	Default	Default	Chi
StatusCatStarAll	N	Default	Rejected	Default	Binary	Default	Default	Default	Nur
TargetB	N	Default	Target	Rejected	Binary	Default	Default	Default	Nur
GiftTimeLast	N	Default	Input	Default	Interval	Default	Default	Default	Nur
PromCntCard36	N	Default	Rejected	Default	Interval	Default	Default	Default	Nur
DemAge	Y	Default	Rejected	Default	Interval	Default	Default	Default	Nur

2. Select **New Role** ⇒ **Target** for **D_TARGETB**.
3. Select **New Role** ⇒ **Rejected** for **TargetB**.
4. Select **OK** to close the Variables dialog box.
5. Run the Metadata node. Do not view the results.

Exploring the Description Tree

Follow these steps to explore the description tree.

1. Run the Description Tree node. View the results.
2. Select **View** ⇒ **Model** ⇒ **Iteration Plot** from the Description Tree results.
3. Change the basis of the plot to **Misclassification Rate**.



The Assessment plot shows the trade-off between tree complexity and agreement with the original neural network model. The autonomously fit description agrees with the neural network about 95% of the time, and as the tree becomes more complicated, the agreement with the neural network improves. You can scrutinize this tree for the input values resulting in a solicit decision, or you can simplify the description with some accuracy loss. Surprisingly, 85% of the decisions made by the neural network model can be summarized by a three-leaf tree!

The following steps generate the three-leaf description tree:

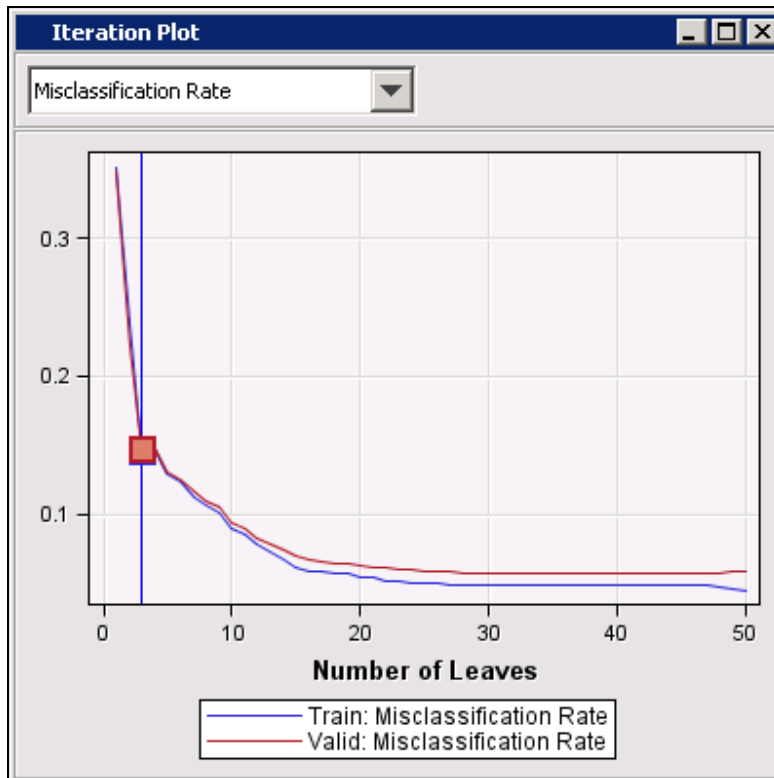
4. Close the Description Tree results window.

5. Under the Subtree properties of the description tree, change the Method to **N** and the Number of Leaves to **3**.

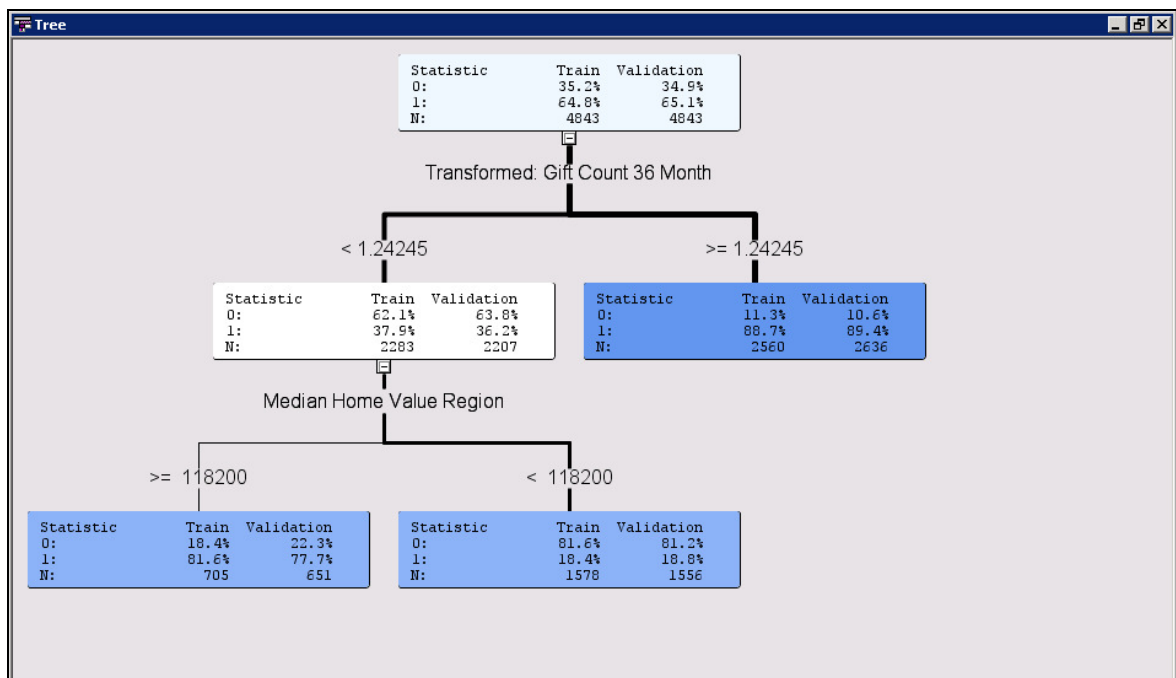
Property	Value
Leaf Size	5
Number of Rules	5
Number of Surrogate Rules	0
Split Size	
<input checked="" type="checkbox"/> Split Search	
Exhaustive	5000
Node Sample	20000
<input checked="" type="checkbox"/> Subtree	
Method	N
Number of Leaves	3
Assessment Measure	Decision
Assessment Fraction	0.25
<input checked="" type="checkbox"/> Cross Validation	
Perform Cross Validation	No
Number of Subsets	10
Number of Repeats	1
Seed	12345
<input checked="" type="checkbox"/> Observation Based Impurity	
Observation Based Impurity	No
Number Single Var Impurity	5
<input checked="" type="checkbox"/> P-Value Adjustment	
Bonferroni Adjustment	Yes
Time of Kass Adjustment	Before
Inputs	No
Number of Inputs	1
Split Adjustment	Yes
<input checked="" type="checkbox"/> Output Variables	
Leaf Variable	Yes

6. Run the Description Tree node and open the Results window.

7. Select **View** ⇒ **Model** ⇒ **Iteration Plot** from the Description Tree results.
8. Change the basis of the plot to **Misclassification Rate**.



The accuracy dropped to 85%, but the tree has only two rules.



You can conclude from this tree that the neural network is soliciting donors who gave relatively many times in the past or donors who gave fewer times, but live in more expensive neighborhoods.

You should experiment with accuracy versus tree size trade-offs (and other tree options) to achieve a description of the neural network model that is both understandable and accurate.

It should be noted that the inputs used to describe the tree are the same as the inputs used to build the neural network. You can also consider other inputs to describe the tree. For example, some of the monetary inputs were log-transformed. To improve understandability, you can substitute the original inputs for these without loss of model accuracy. You can also consider inputs that were **not** used to build the model to describe the decisions. For example, if a charity's Marketing Department was interested in soliciting new individuals similar to those selected by the neural network, but without a previous donation history, it could attempt to describe the donors using demographic inputs only.



It should be noted that separate sampling distorts the results shown in this section. To accurately describe the solicit and ignore decisions, you must down-weight the primary cases and up-weight the secondary cases. This can be done by defining a frequency variable defined as shown.

```
if TargetB=1 then WEIGHT = 0.05/0.50;  
else           WEIGHT = 0.95/0.50;
```

You can define this variable using a Transform Variables tool's Expression Builder. You then set the model role of this variable to Frequency.

Another approach is to build the model on the **Score** data set. Because the **Score** data set is not oversampled, you get accurate agreement percentages. This is possible because the description tree does not depend on the value of the target variables, only on the model predictions.