**In-Class Exercise: Decision Trees (Introduction to Predictive Modeling)**

(adapted from Applied Analytics using SAS Enterprise Miner, SAS Institute, Cary, NC. 2010)



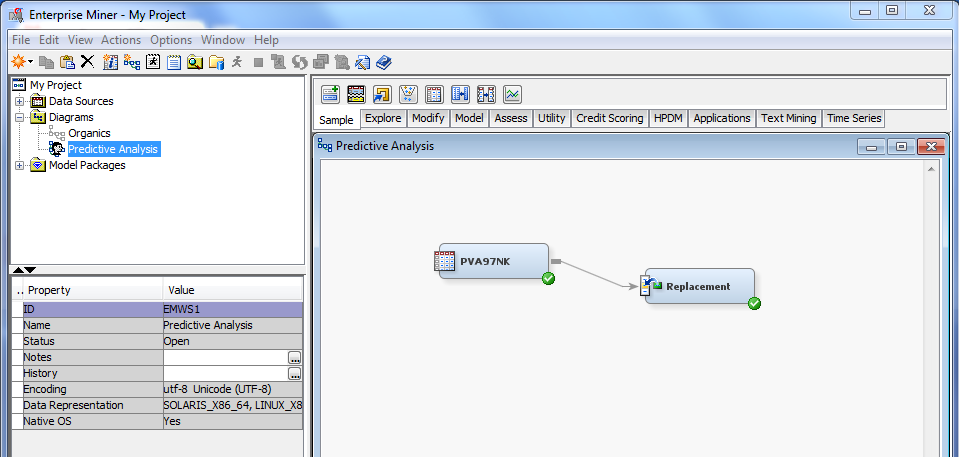
**Creating Training and Validation Data**

A critical step in prediction is choosing among competing models. Given a set of training data, you can easily generate models that very accurately predict a target value (an outcome) from a set of input values. Input values might be customer age and income, and the target value might be whether that customer buys a car this year. Unfortunately, these predictions might be accurate only for the training data itself. Attempts to generalize the predictions beyond the original data set can result in substantial accuracy reductions.

To avoid this pitfall, SAS Enterprise Miner uses a *validation* *data set* to assess model performance and make sure we’re getting accurate predictions. Typically, the validation data set is created by partitioning the original data set. Part of the data are selected for training and used to build the model, and part of the data are selected for validation and used to tune and compare models.

**🖉** SAS Enterprise Miner also enables the creation of a third partition, named the *test* *data set*.   
The test data set is meant for obtaining unbiased estimates of model performance from a single selected model. We’re not going to worry about that in this exercise.

This demonstration assumes that you completed the demonstrations in last week. Open the SAS Enterprise Miner workspace you used for the last in-class exercise. Open up the diagram associated with your workspace and you should see something like this:

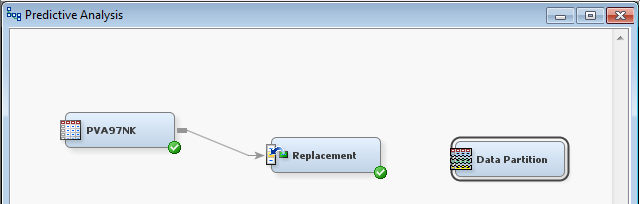


Follow these steps to partition the raw data into training and validation sets.

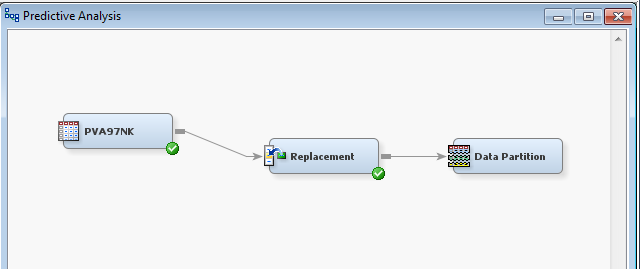
1. Select the **Sample** tool tab. The Data Partition tool is the second from the left.



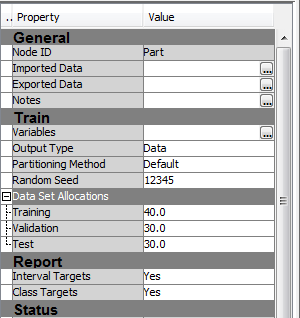
1. Drag a **Data Partition** tool into the diagram workspace, next to the Replacement node.



1. Connect the **Replacement** node to the **Data Partition** node.



1. Select the **Data Partition** node and examine its Properties panel.

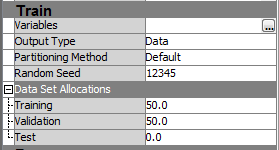


Use the Properties panel to select the fraction of data devoted to the Training, Validation, and Test partitions. By default, less than half the available data is used for generating the predictive models.

**🖉** There is a trade-off in various partitioning strategies. More data allocated for training results in more stable (better) predictive models, but poorer model assessments (figuring out how good the model is). Also, the Test partition is used only for calculating fit statistics after the modeling and model selection is complete (like a “double-check” on the quality of the model). Many analysts regard this as a waste of data; that’s why we’re not doing it.

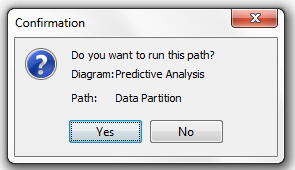
A typical partitioning strategy skips the Test partition and places an equal number of cases in Training and Validation.

1. Type **50** as the Training value in the Data Partition node.
2. Type **50** as the Validation value in the Data Partition node.
3. Type **0** as the Test value in the Data Partition node.

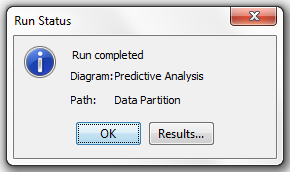


**🖉** With smaller raw data sets, model stability can become an important issue because you may not have enough data to really capture what’s going on. In that case, increase the number of cases devoted to the Training partition.

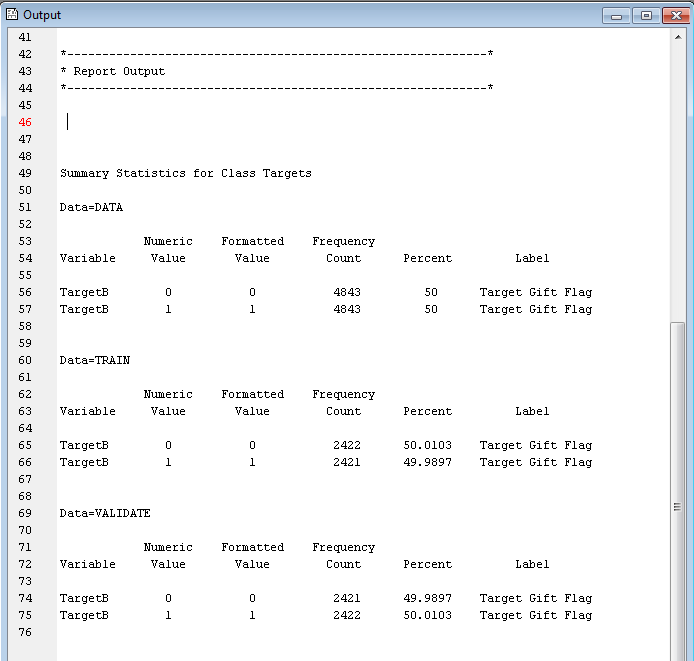
1. Right-click the **Data Partition** node and select **Run**.
2. Select **Yes** in the Confirmation window. SAS Enterprise Miner runs the Data Partition node.



When the Data Partition process is complete, a Run Status window opens.



1. Select **Results…** in the Run Status window to view the results.

The Results - Node: Data Partition window provides a basic metadata summary of the raw data that feeds the node and, more importantly, a frequency table that shows the distribution of the target variable, **TargetB**, in the raw, training, and validation data sets.  
  


**🖉** The Partition node attempts to maintain the proportion of zeros and ones in the training and validation partitions. The proportions are not exactly the same due to an odd number of zeros' and ones' cases in the raw data.

The analysis data was selected and partitioned. You are ready to build predictive models.

Constructing a Decision Tree Predictive Model

This demonstration illustrates the interactive construction of a decision tree model and builds   
on the process flows of previous demonstrations.

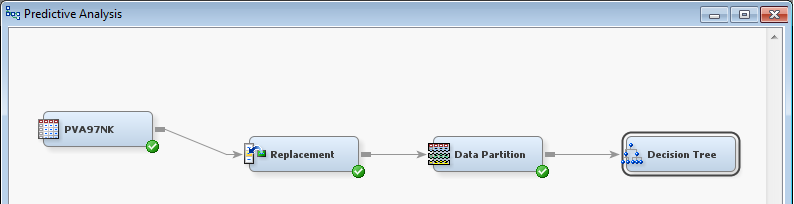
#### Preparing for Interactive Tree Construction

Follow these steps to prepare the Decision Tree tool for interactive tree construction.

1. Select the **Model** tab. The Decision Tree tool is second from the left.

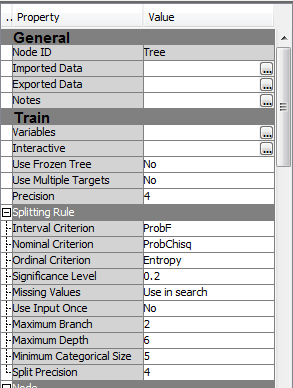


1. Drag a **Decision Tree** tool into the diagram workspace. Place the node next to the **Data Partition** node.
2. Connect the **Data Partition** node to the **Decision Tree** node.

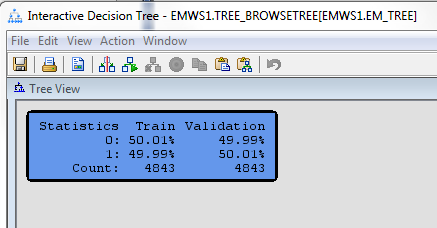


The Decision Tree tool can build predictive models autonomously (automatically without your input) or interactively. To build a model autonomously, simply run the Decision Tree node. Building models interactively, however, gives you more control and can be more informative and helpful when you first learn about creating decision trees (and even when you are more experienced!).

1. Select **Interactive** ⇨  from the Decision Tree node's Properties panel.



The SAS Enterprise Miner Interactive Decision Tree application opens.

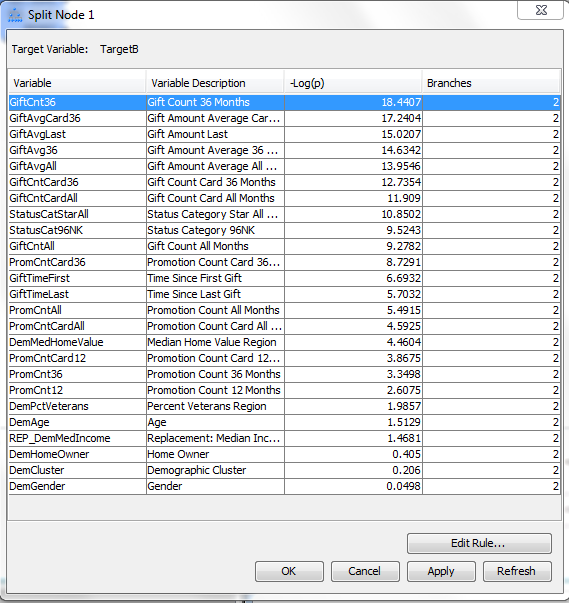


#### Creating a Splitting Rule

Decision tree models involve recursive partitioning of the training data in an attempt to isolate groups of cases (observations or data points) with identical target values. The blue box in the Tree window represents the un-partitioned training data. The statistics show the distribution of **TargetB**.

Use the following steps to create an initial splitting rule:

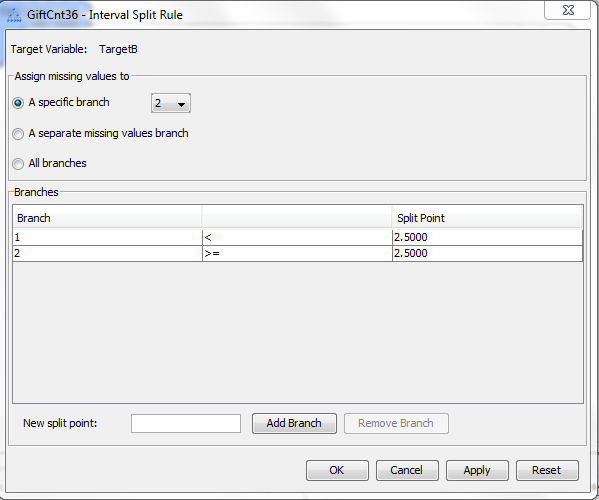
1. Right-click the blue box and select **Split Node…** from the menu. TheSplit Node 1 dialog box opens.

s

The Split Node dialog box shows the -Log(p-value), or *logworth,* of using each variable to partitioning the training set. The variable with the highest logworth is the best predictor at this point in the analysis. As the logworth increases, the p-value gets smaller and the Chi-squared value for the model is bigger. That means that using that variable as a predictor best isolates “pure” outcomes – we ideally want each leaf node to only contain one outcome (buy/not buy; default on loan/not default).

We see that **Gift Count 36 Months** has the highest logworth, followed by **Gift Amount Average Card 36 Months** and **Gift Amount Last**. You can choose to split the data on the selected input or edit the rule for more information.

1. Select **Edit Rule…**. The GiftCnt36 - Interval Splitting Rule dialog box opens.



This dialog box allows you to partition using the input variable **Gift Count 36 Months**. Two branches are created. The first branch contains cases with a 36-month gift count less than 2.5, and the second branch contains cases with a 36-month gift count greater than or equal to 2.5. In other words, with cases that have a 36-month gift count of 0, 1, or 2 branch left and cases with 3 or more branch right. Any cases with a missing or unknown 36-month gift count are placed in the first branch.

**🖉** The interactive tree assigns any cases with missing values to the branch that maximizes the logworth (or “purity”) of the split. For **GiftCnt36**, this rule assigns cases with missing values to branch 2.

1. Select **Apply**. The GiftCnt36 - Interval Splitting Rule dialog box remains open, and the Tree View window is updated. You may have to click “OK” to see the tree.



The training data is partitioned into two subsets. The first subset, cases with a 36-month gift count less than 2.5, has a higher than average concentration of **TargetB**=0 cases (the person **does not** give a gift). The second subset, cases with a 36-month gift count greater than 2.5, has a higher than average concentration of **TargetB**=1 cases (the person **does** give a gift). The second branch also has slightly more cases, indicated by the Count.

The partition of the data also represents your first predictive model! This model assigns to all cases in the left branch a predicted **TargetB** value equal to 0.4297 (about 43%) and to all cases in the right branch a predicted **TargetB** value equal to 0.5625 (about 56%).

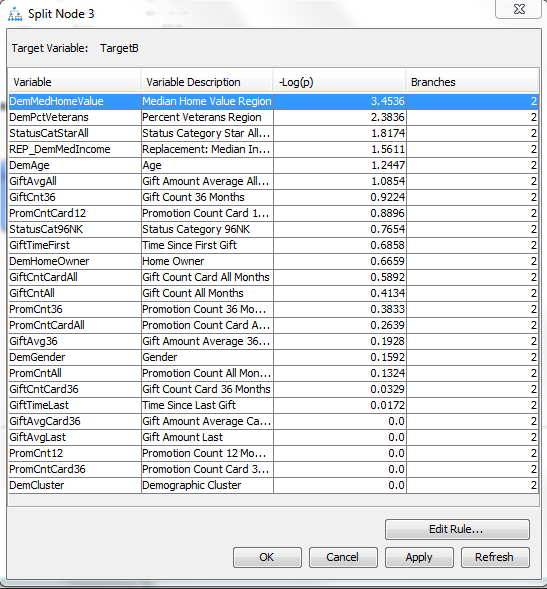
**This means – people with a 36-month gift count less than 2.5 give an additional gift 43% of the time, while people with a 36-month gift count greater than 2.5 give an additional gift 56% of the time.   
  
In most cases, “1” means the outcome happens and “0” means it doesn’t happen.**

Note: on the right of each node, you see the breakdown of how the predictive model actually performs against the validation data set. For now, the performance is almost identical, which is good. Eventually, when you start adding enough classifier variables, you will see that the fit to the training data will be much higher than the out of sample predictive capacity for your model.

#### Adding More Splits

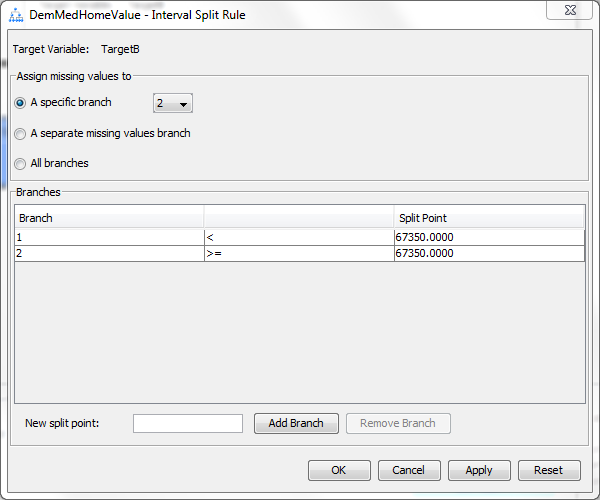
Use the following steps to interactively add more splits to the decision tree:

1. Select the lower left partition (the one associated with the branch “<2.5”). The Split Node dialog box is updated to show the possible partition variables and their logworths.



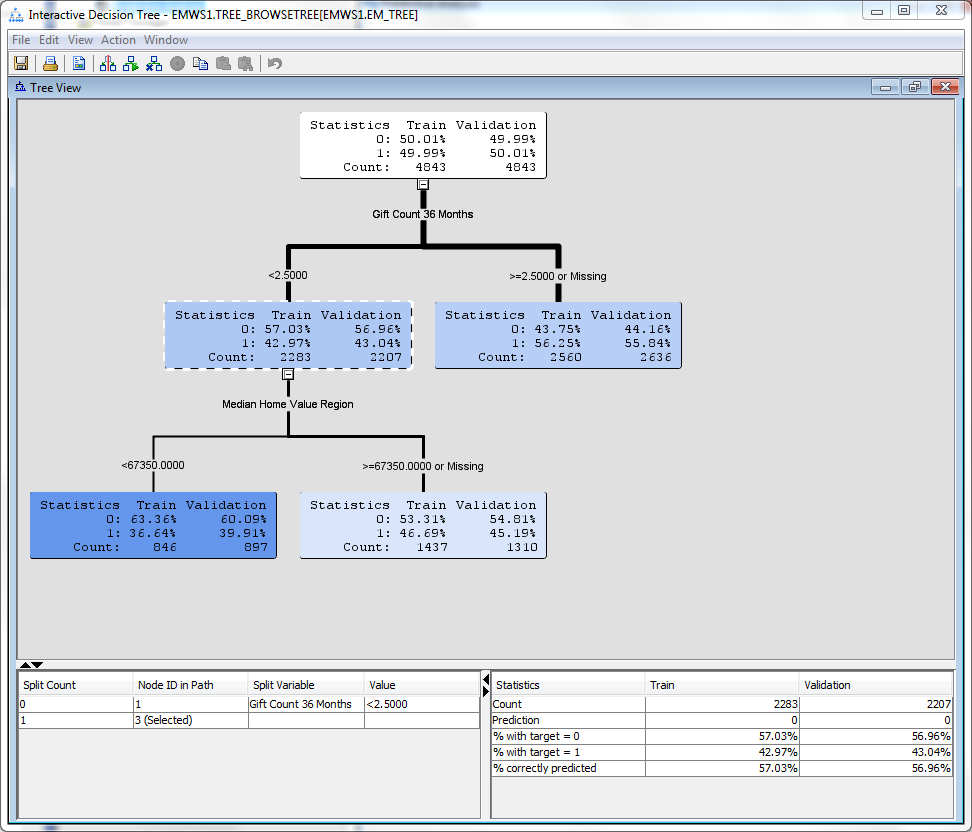
The input with the highest logworth is **Median Home Value Region**.

1. Select **Edit Rule…**. The DemMedHomeValue - Interval Splitting Rule dialog box opens.



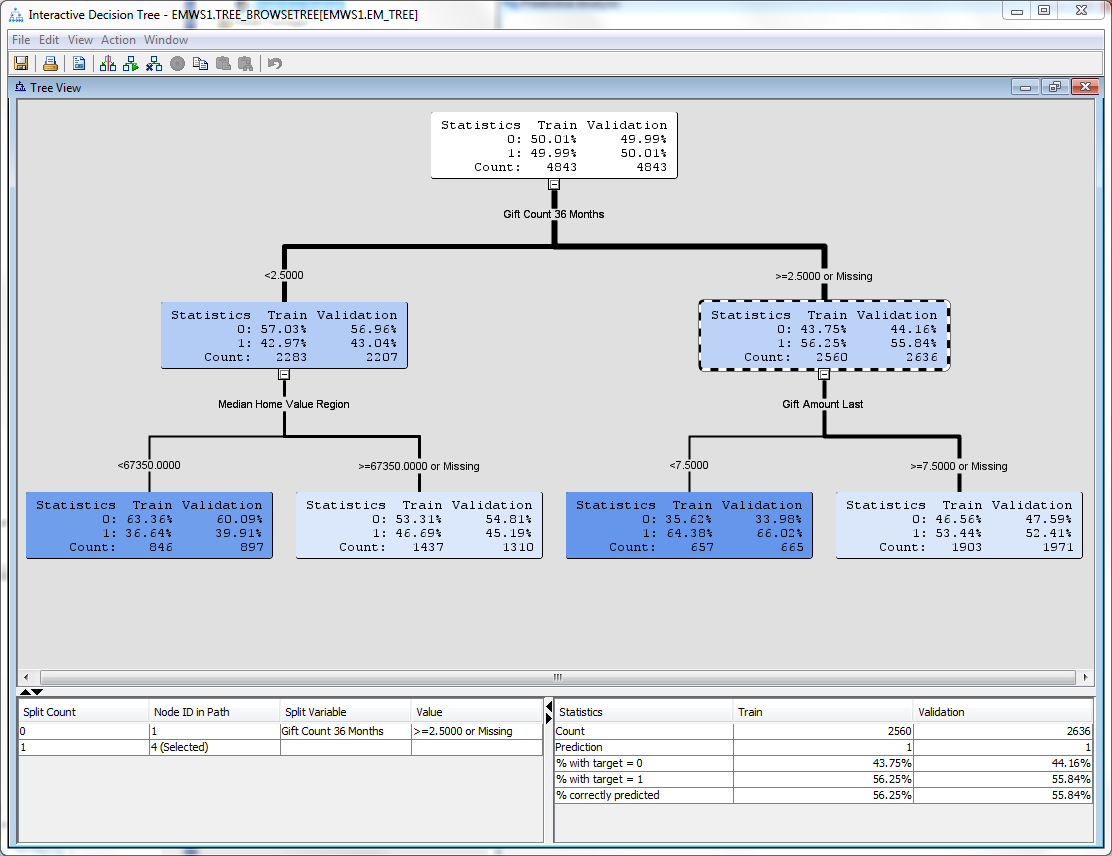
Branch 1 contains all cases with a median home value less than 67350. Branch 2 contains all cases that are equal to or greater than 67350.

1. Select **Apply**. The Tree View window is updated to show the additional split.



Both left and right leaves contain a lower-than-average proportion of cases with **TargetB**=1 (The percentage of cases in the training set that equals 1 is less than 50% for both nodes)

1. Repeat the process for the branch that corresponds to cases with **Gift Count 36 Month** greater than 2.5. Make sure you use the split with the highest logworth value (GiftAvgLast).



The right-branch cases are split using the **Gift Amount Last** input. This time, both branches have a higher-than-average concentration of **TargetB**=1 cases.

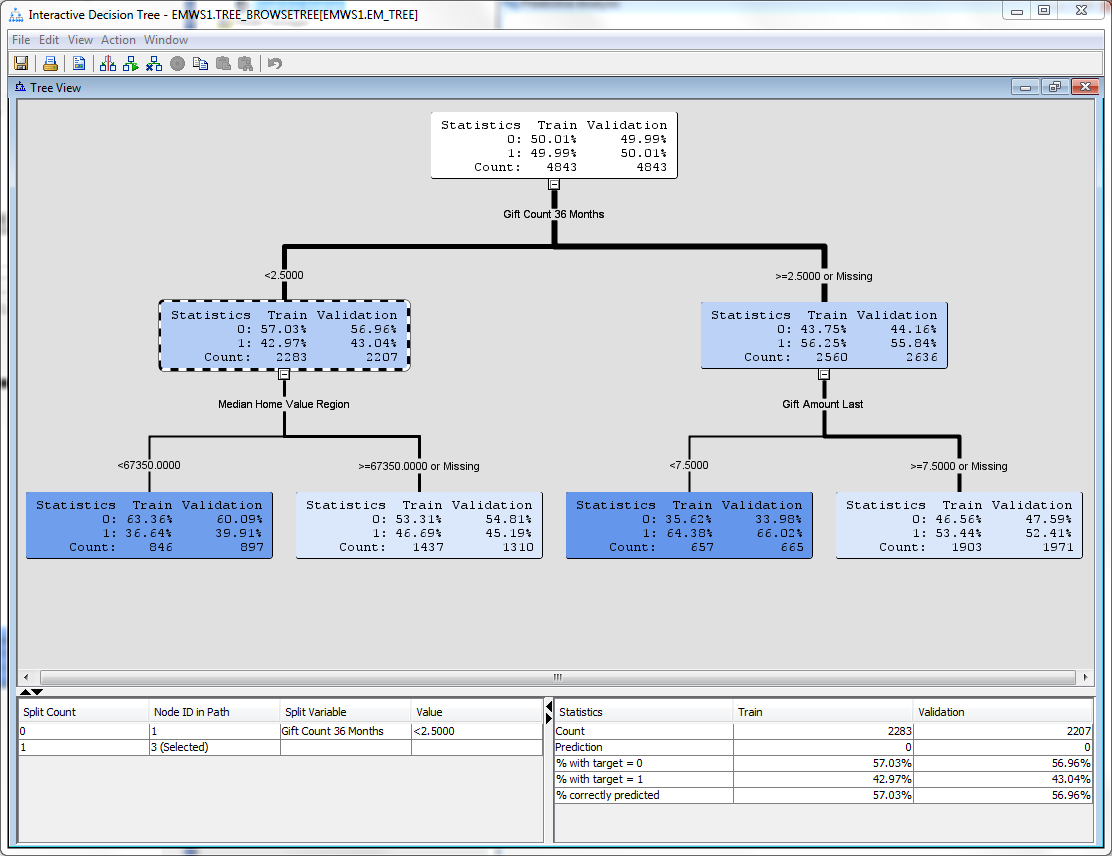
**🖉** Remember that the main focus is selecting useful input variables for the predictive model using the **training data**. The value of each additional variable diminishes with each new split, which prompts the question: Where do you stop splitting? The validation data provides the necessary information to answer this question. Eventually, the model’s performance (prediction accuracy) in the validation data will begin to grow worse – that’s where you stop! More on this later…

#### Changing a Splitting Rule XYZZY

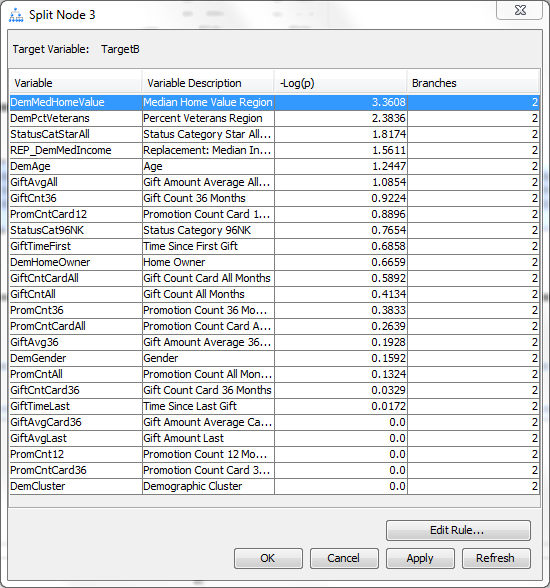
The bottom left split on **Median Home Value** at 67350 was found to be optimal in a statistical sense, but it might seem strange to someone looking at the tree. Why not split at the more logical, “round” value of 70000? You can use the Splitting Rule window to edit where a split occurs.

Use the following steps to define your own splitting rule for an input.

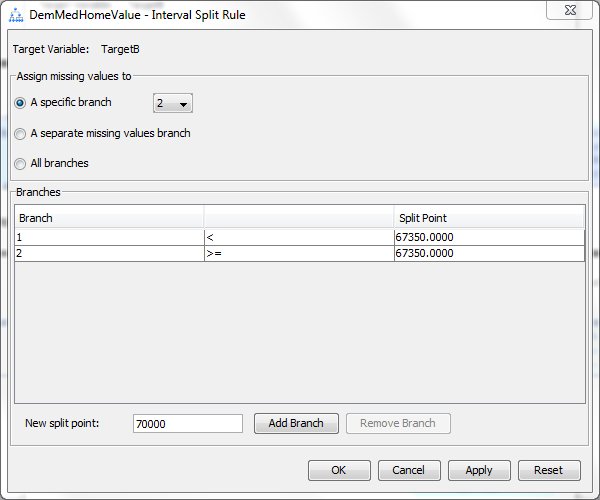
1. Select the node above the label **Median Home Value Region**.



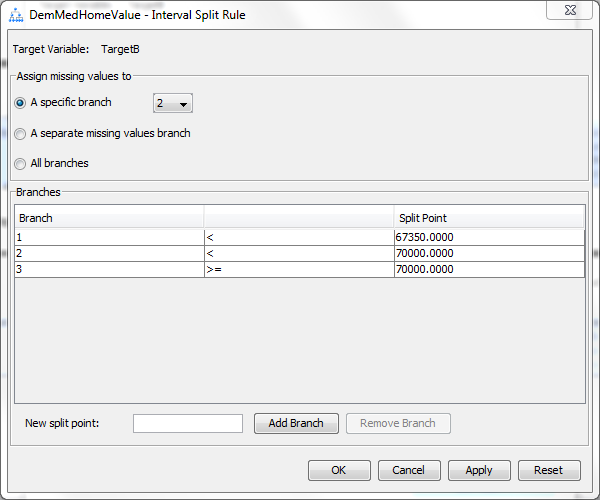
1. Right-click this node and select **Split Node**… from the menu. The Split Node window opens.



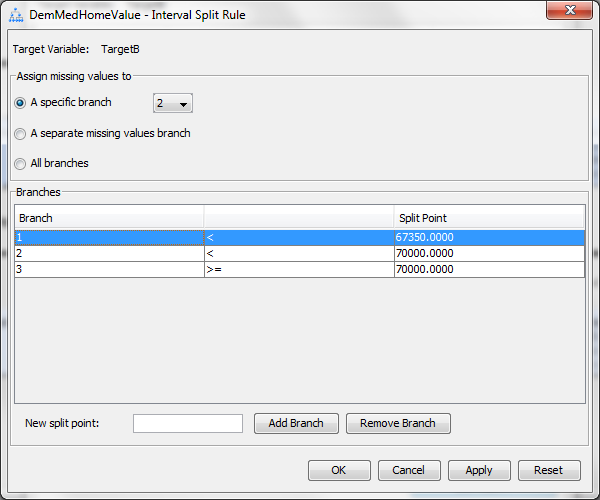
1. Select **Edit Rule**…. The DemMedHomeValue - Interval Splitting Rule dialog box opens.
2. Type **70000** in the New split point field.



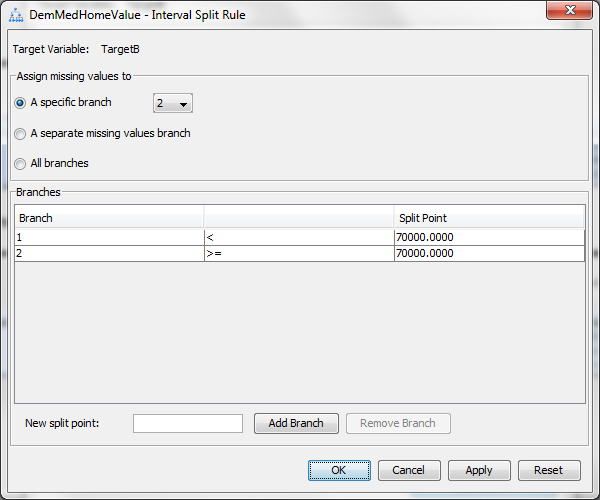
1. Select **Add Branch**. The Interval Splitting Rule dialog box now shows three branches.



1. Select **Branch 1** by highlighting the row. Then select **Remove Branch**.

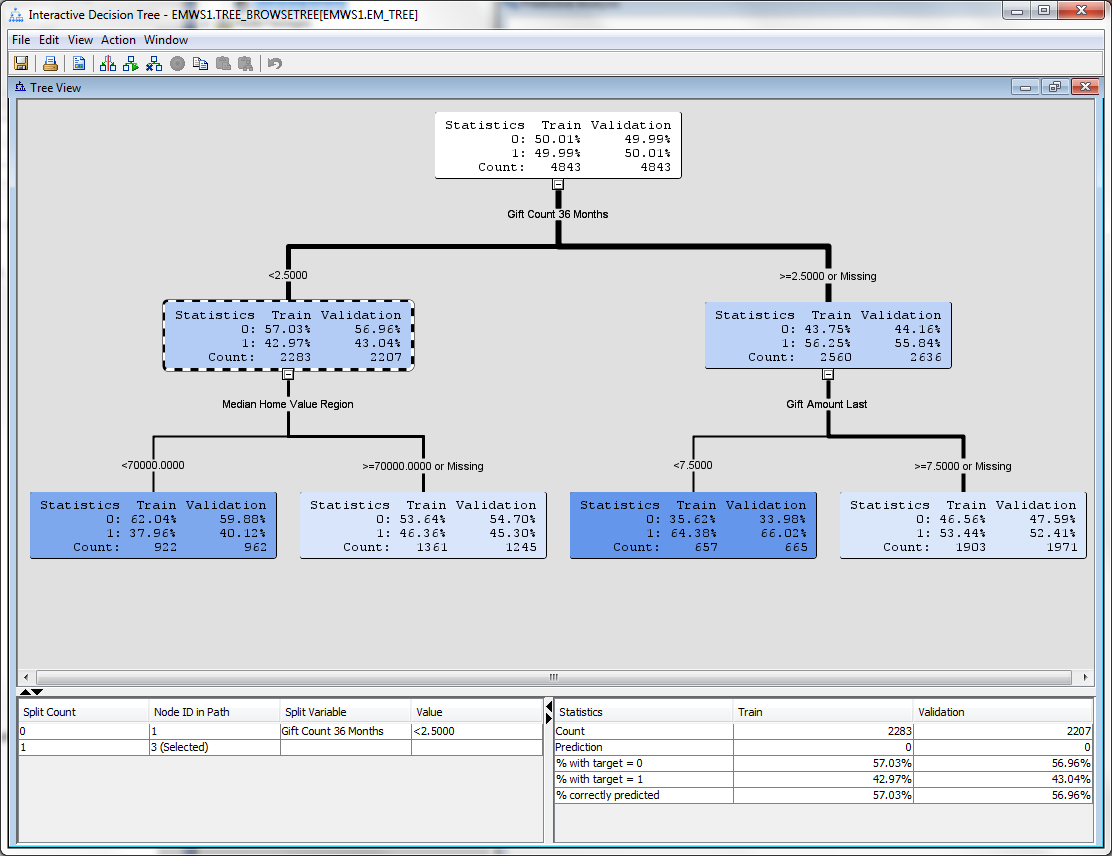


The split point for the partition is moved to 70000.



**🖉** In general, to change a Decision Tree split point, add the new split point first and then remove the unwanted split point.

1. Select **OK** to close the Interval Splitting Rule dialog box and to update the Tree View window.



Overriding the statistically optimal split point slightly changed the distribution of cases in both nodes.

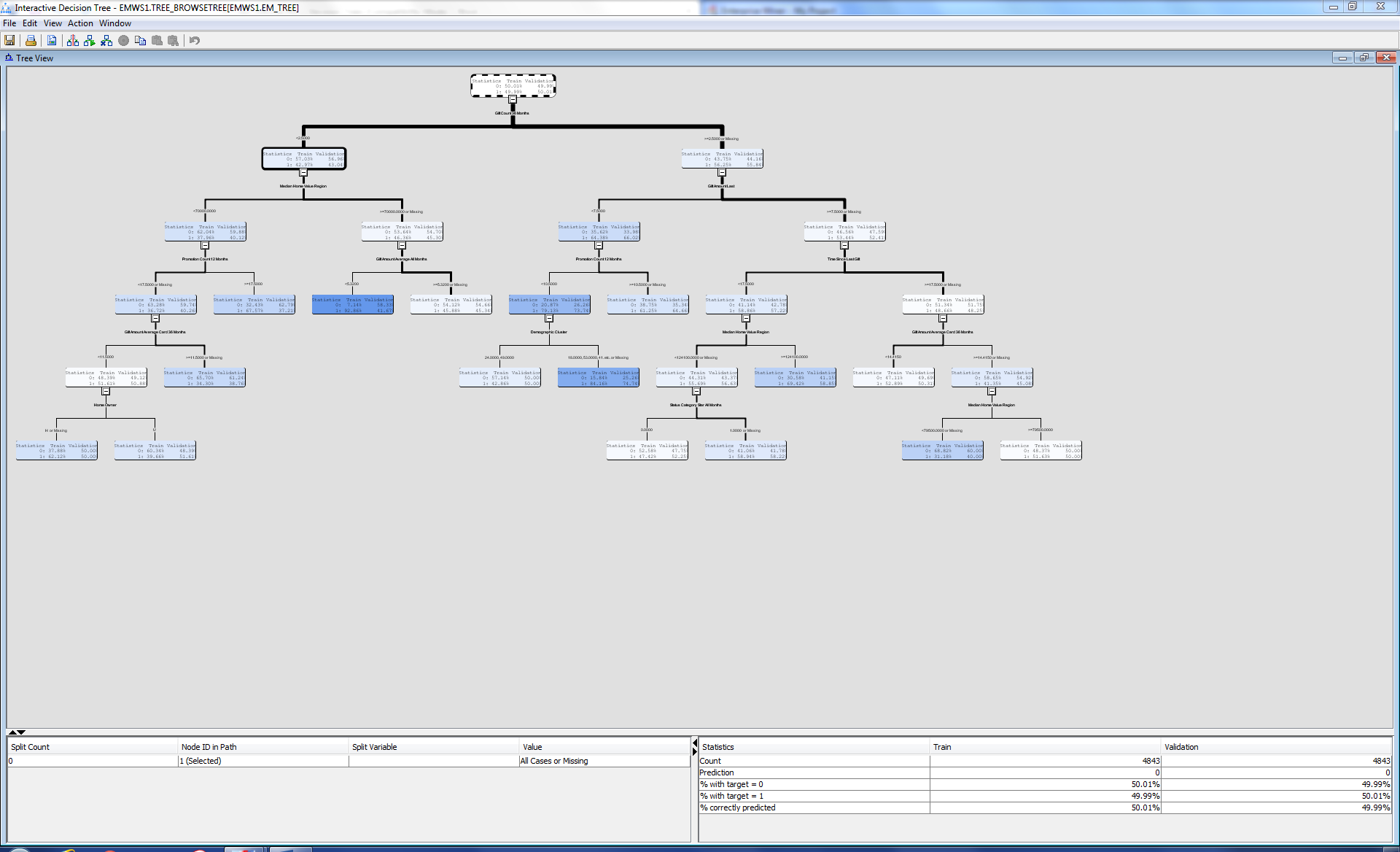
**Creating the Maximal Tree**

As demonstrated thus far, the process of building the decision tree model is one of carefully considering and selecting each split sequentially. There are other, more automated ways to “grow” a tree.

Follow these steps to automatically grow a tree with the SAS Enterprise Miner Interactive Tree tool:

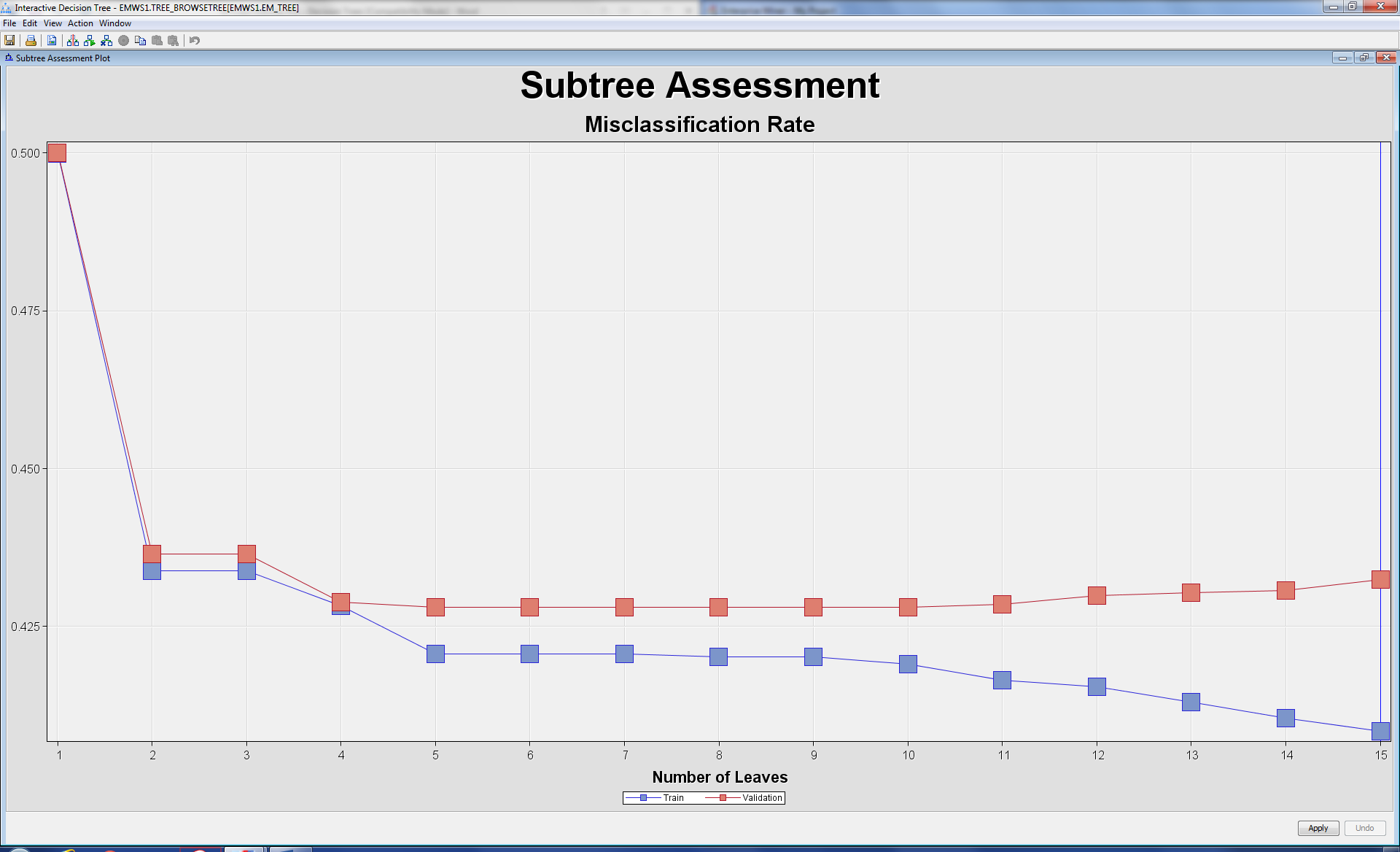
1. Select the root node of the tree.
2. Right-click and select **Train Node** from the menu. The tree is grown until stopping rules prohibit further growth.
3. It is difficult to see the entire tree without scrolling. However, you can zoom into the Tree View window to examine the basic shape of the tree.

Right-click inside the Tree View window and select **View** ⇨ **50%** . The tree view is scaled to provide the general shape of the maximal tree.



Plots and tables within the Interactive Tree tool provide a preliminary assessment of the maximal tree.

1. Select **View** ⇨ **Subtree Assessment Plot**.



This is a plot of the misclassification rate of trees automatically generated with 2 nodes, 3 nodes, 4 nodes, etc. up to 15 nodes (where it stopped because that includes every possible variable). Looking only at the “Train” result (the blue line) the majority of the improvement in fit occurs over the first few splits, it appears that the maximal (every predictor included) tree generates the lowest misclassification rate.

The model performance in terms of “misclassification” for the training data seems to indicate that the maximal tree is the most preferred for assigning predictions to cases. However, considering model performance based solely on the blue series plot is misleading.

Remember that the model’s usefulness is how well it performs with **new, as yet unseen data**. That’s the validation set. Using only the training data creates an “overfitted” models that does well making predictions based on that single set of data but may not apply to all data – every sample is different. We want to make sure our decision tree works with any data we throw at it.

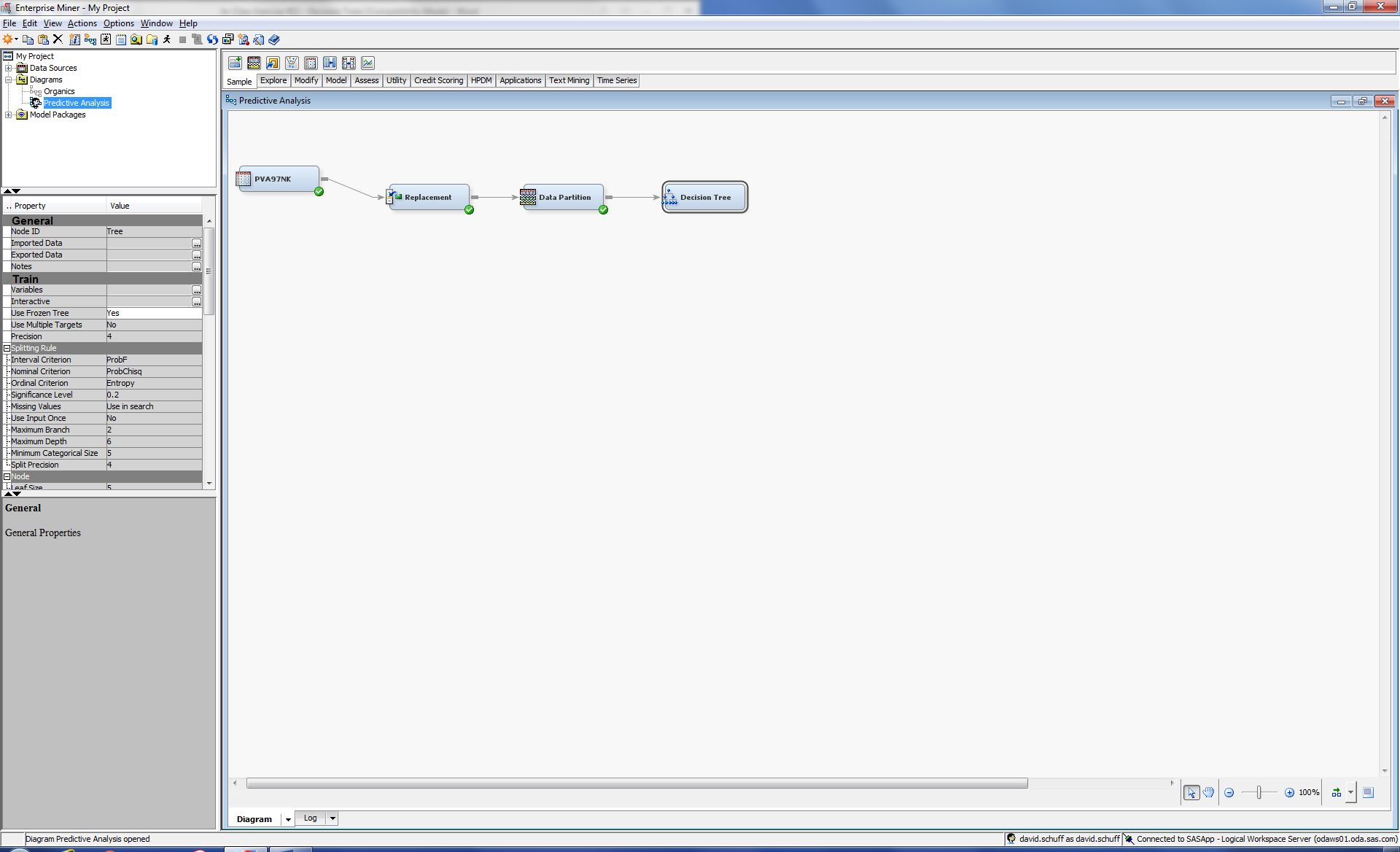
Your next task is to further investigate how well this model generalizes to another similar, but independent, sample of data (i.e., the validation data). While we have some preliminary indications from the Subtree Assessment plot of misclassification, as explained in the next section, we can do more to figure this out.

Select **File** ⇨ **Exit** to close the Interactive Tree application. A dialog box opens. You may be asked whether you want to save the maximal tree to the default location. If so, select **Yes**. The Interactive Tree results are now available from the Decision Tree node in the SAS Enterprise Miner flow.

Assessing a Decision Tree

Use the following steps to perform an assessment of the maximal tree on the validation sample.

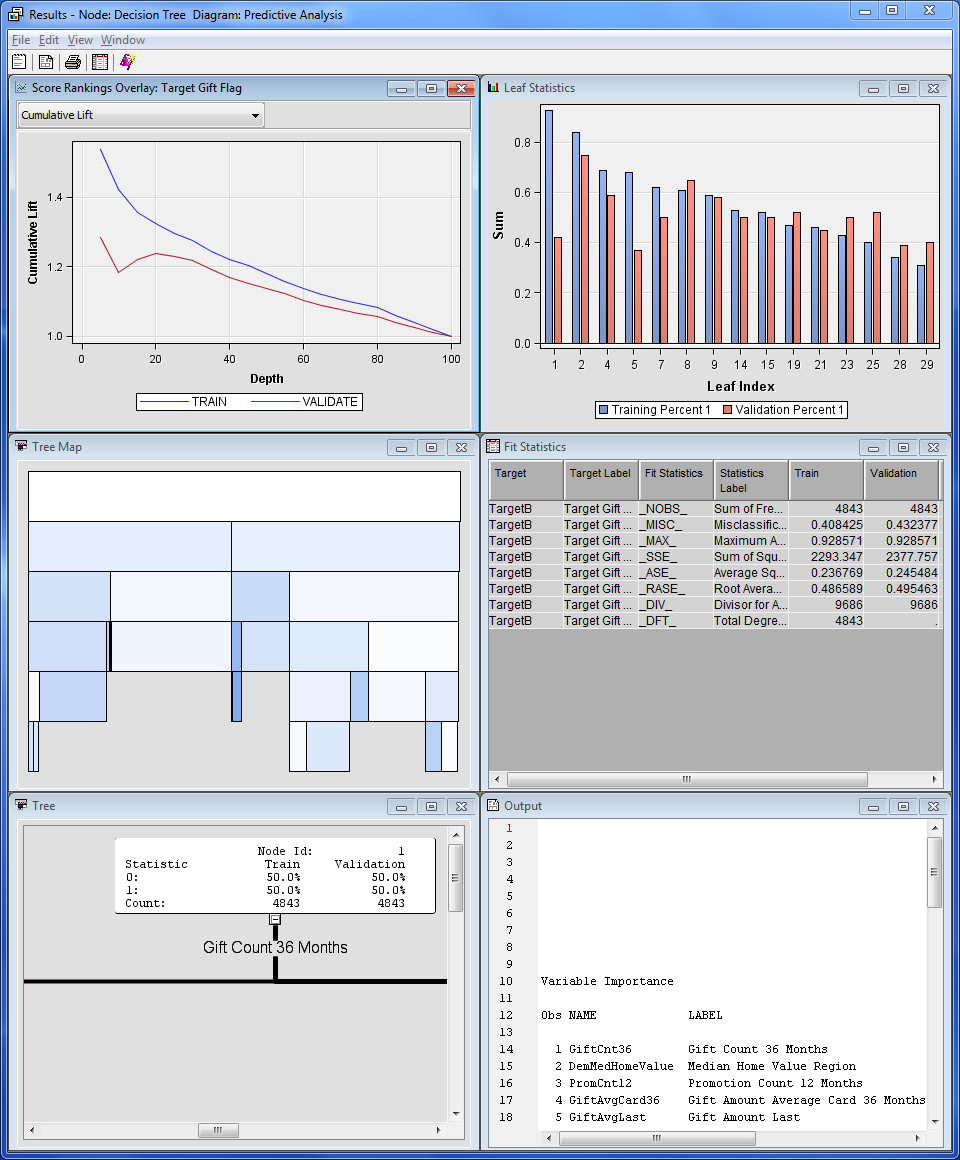
1. Select the **Decision Tree** node. In the Decision Tree node's properties, change **Use Frozen Tree** from No to **Yes**.



The Frozen Tree property prevents the maximal tree from being changed by other property settings when the flow is run.

1. Right-click the **Decision Tree** node and run it. When it’s done, select **Results**.

The Results window opens.

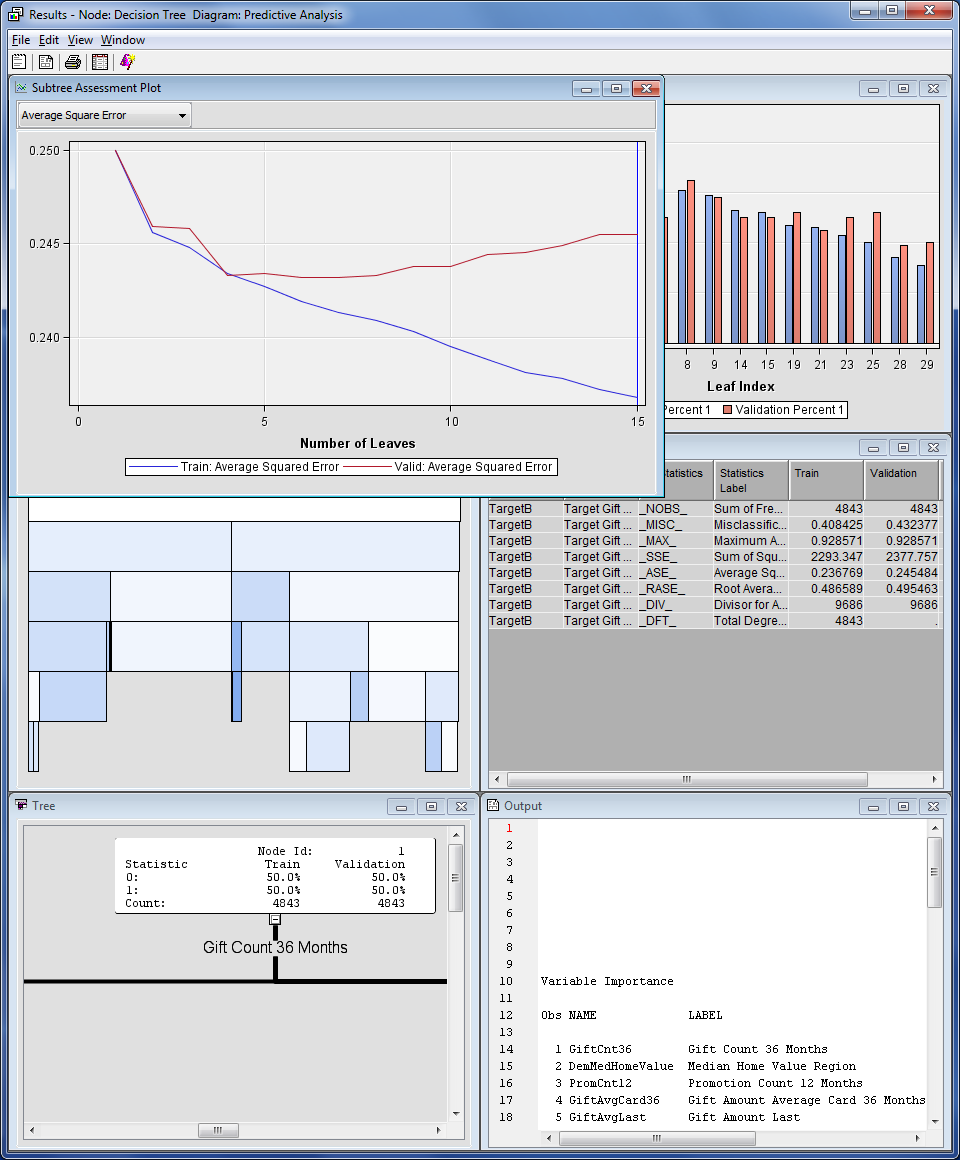


**🖉** The Results window contains a variety of diagnostic plots and tables, including a cumulative lift chart, a tree map, and a table of fit statistics. The diagnostic tools shown in the results vary with the measurement level of the target variable.

The saved tree diagram is in the bottom left corner of the Results window. You can verify this by maximizing the tree plot.

1. The main focus of this part of the demonstration is on assessing the performance of a 15-leaf tree, created with the Interactive Decision Tree tool, on the validation sample. Notice that several of the diagnostics shown in the results of the Decision Tree node use the validation data as a basis.

Select **View** ⇨ **Model** ⇨ **Subtree Assessment Plot**.



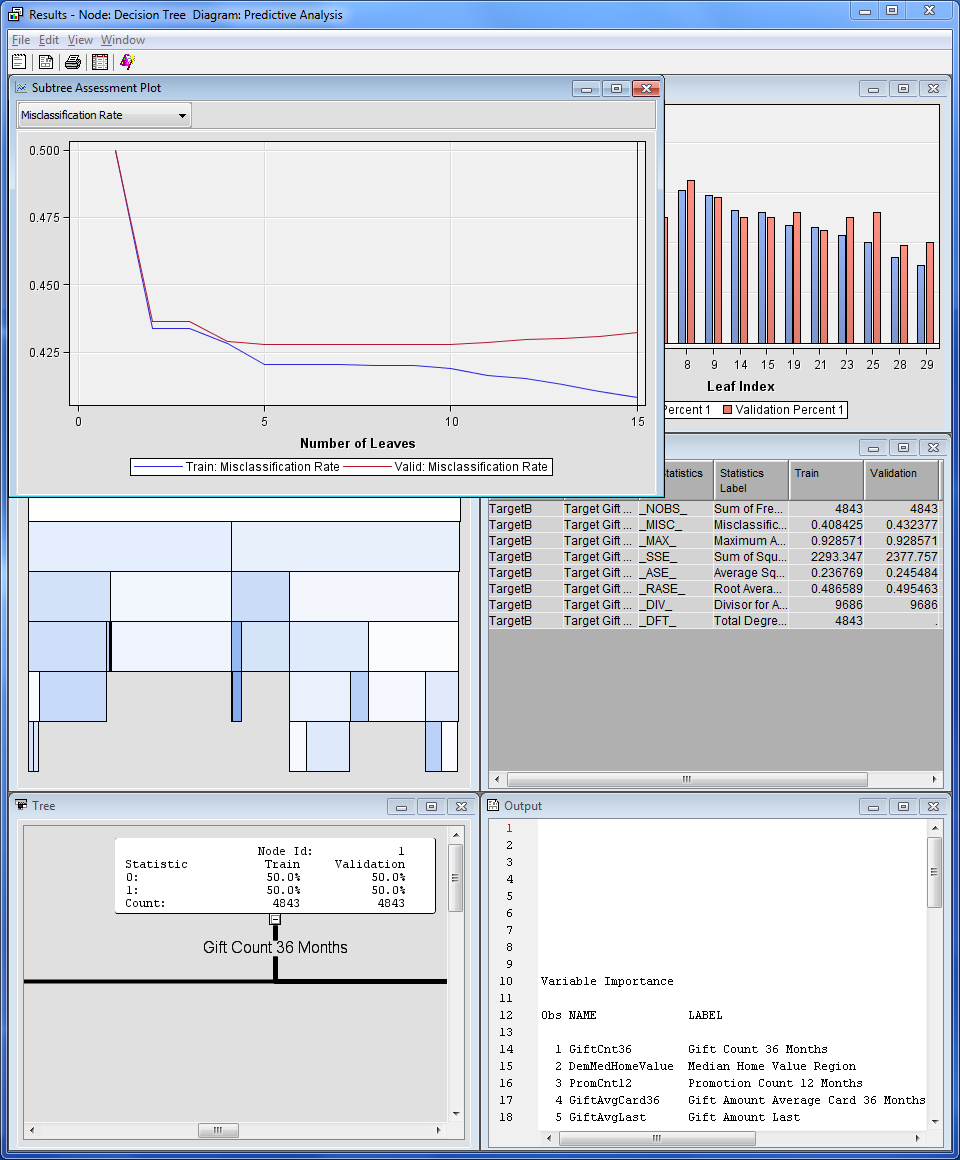
The default plot shows the Average Square Error corresponding to each subtree. The deviation between the predicted and actual outcome for each case is squared and averaged across each outcome category (leaf node). Lower numbers are better – lower average squared error means the model performs better as a predictor because it is “wrong” less often.

This plot is similar to the one generated with the Interactive Decision Tree tool, and it confirms the optimality of the 15-leaf tree. Trees can have more leaves either by having more splits, more input variables, or both.

The performance on the training sample becomes better and better as the tree becomes more complex. However, the performance on the validation sample only improves up to a tree of, approximately, four or five leaves, and then diminishes as model complexity increases.

**🖉** The validation performance shows evidence of model overfitting. Over the range of one to approximately four leaves (1-2 predictors, producing four outcome groups), the precision of the model improves with the increase in complexity. However, precision decreases as complexity increases past this range; the additional complexity (additional variables) helps with this training sample, but this makes the model custom-fit to this particular set of data. It will work less well when used with the validation data (or any other data set).

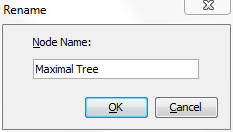
1. To further explore validation performance, select the arrow in the upper left corner of the Assessment Plot, and switch the assessment statistic to **Misclassification Rate** (the stat we looked at previously in the Interactive Decision Tree tool).



The validation performance under Misclassification Rate is similar to the performance under Average Square Error. The optimal tree appears to have, roughly, four or five leaves.

*Misclassification Rate*:  
The name “*decision tree model*” comes from the decision that is made at the leaf or terminal node. In SAS Enterprise Miner, this decision, by default, is a classification based on the *predicted target value*. A predicted target value in excess of 0.5 results in a primary outcome classification (a “1”), and a predicted target value less than 0.5 results in a secondary outcome classification (a “0”). Within a leaf, the predicted target value decides the classification for all cases regardless of each case's actual outcome (majority rules - if the prediction is a 60% probability of Target B equaling one at a given leaf node, then it is assumed that ALL elements falling in that leaf node will be equal to 1).   
  
Of course, this classification will be inaccurate for the fraction of cases where this is not true. The fraction of cases for which the wrong decision (classification) was made is the proportion misclassified is calculated for both the training and validation data sets, and is referred to as the *misclassification rate.*

1. Close the Decision Tree Results window.
2. Right-click on the **Decision Tree** node and select **Rename**. Name the node **Maximal Tree**.



The current process flow should look similar to the following:



The next step is to generate the optimal tree using the default Decision Tree properties in   
SAS Enterprise Miner.

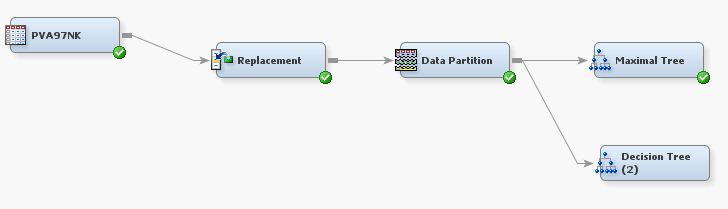
**Pruning a Decision Tree**

So we know we don’t want the maximal tree – it’s clearly not the best one for the validation set. So we need to “prune” it – find the number of leaves that give us the best result.

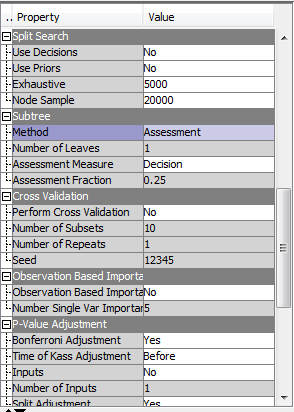
The optimal tree in the sequence can be identified using tools contained in the Decision Tree node. The relevant properties are listed under the Subtree heading.

1. Drag another **Decision Tree** node from the Model tab into the diagram.

Connect it to the **Data Partition** node so that your diagram looks similar to what is shown below.



1. Select the new **Decision Tree** node and scroll down in the Decision Tree properties to the **Subtree** section. The main tree pruning properties are listed under the Subtree heading (you may have to scroll down to find it).



The default method used to prune the maximal tree is Assessment. This means that algorithms in SAS Enterprise Miner choose the best tree in the sequence based on some optimality measure.

The Assessment Measureproperty specifies the optimality measure used to select the best tree in the sequence. The default measure is Decision. The default measure varies with the measurement level of the target variable, and other settings in SAS Enterprise Miner.

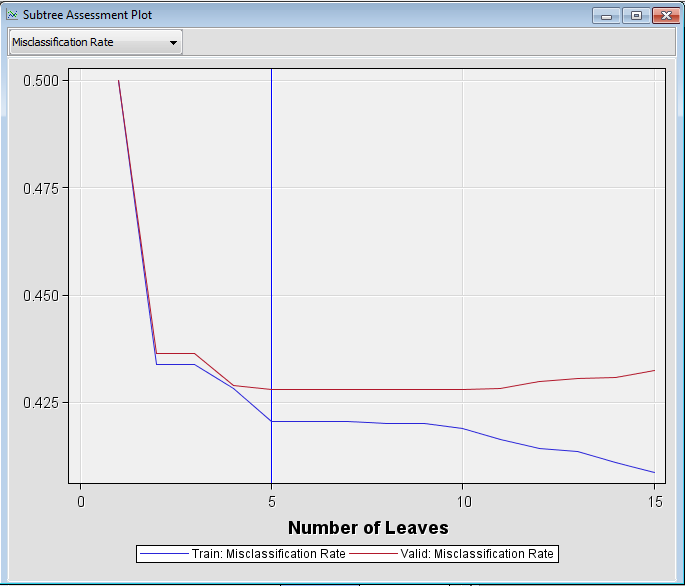
**🖉** Metadata plays an important role in how SAS Enterprise Miner functions. Recall that a binary (value = 0 or 1) target variable was selected for the decision tree model (i.e.., TargetB). Based on this, SAS Enterprise Miner assumes that you want a tree that is optimized for making the best **decisions** (i.e., predicting 1 or 0 values, as opposed to ranking outcomes or generating probability estimates). That is, using the Decision as the assessment measure, SAS Enterprise Miner identifies the tree with the **lowest misclassification rate on the validation sample**.

1. Right-click on the new **Decision Tree** node and select **Run**.
2. After the Decision Tree node runs, select the **Results…**.button.
3. In the Result window, select **View** ⇨ **Model** ⇨ **Subtree Assessment Plot**.

The Plot shows model performance under Average Square Error by default. However, this is not the criterion used to select the optimal tree.

1. Change the basis of the plot to **Misclassification Rate**.

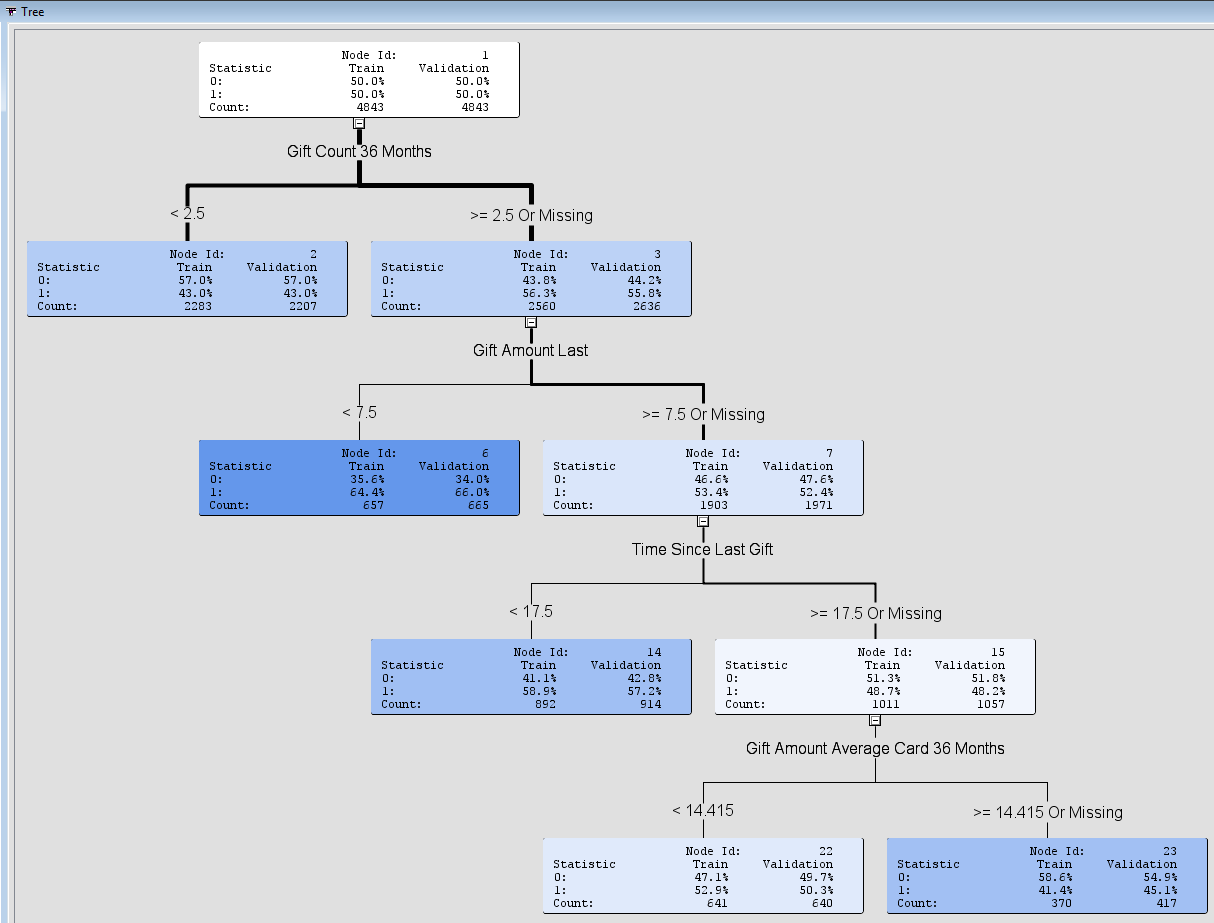
The five-leaf tree has the lowest **misclassification** **rate** on the **validation sample**.



SAS starts with the maximal tree (15-leaves) and then finds the best combination of 15 leaves, recalculating the misclassification rate each time. Then it prunes it back and finds the best 14-leaf tree (the one with the lowest misclassification rate), then the best 13-leaf tree, and so on. This means there can be more than one tree with the same number of leaves – this makes sense because I can split the tree different ways and use different combinations of variables.

The blue line represents the “best of the best” – the best 5-leaf tree has a lower misclassification rate for the validation set than any of those best trees with less or more than 5 leaves. You can see this visually – it’s the lowest point on the red line. Remember, we don’t care at this point that the blue line is lower because that’s the training set.

The optimal tree generated by the Decision Tree node is found in the bottom left corner of the Results window. Maximize that window to see the whole tree:



**IMPORTANT: This is a 5-leaf tree because there are 5 nodes with nothing below them. If there are subnodes (like the node >= 2.5 Or Missing), then it isn’t considered a leaf node. A leaf is the endpoint of the tree and that’s what you use to predict outcomes.**

**So, there is a 57.2% chance that someone will give an additional gift when: their 36-month gift count is more than 2.5, their last gift amount was greater than 7.5, and the time since their last gift was less than 17.5 (check out the highlighted node above).**

1. Close the Results window of the Decision Tree node.

**Now answer the following questions**

What is the probability that a person will make a gift?  
(HINT: Use the results from decision tree for validation data)

1. if they made 1 gift in the last 36 months
2. if they made 5 gifts in the last 36 months and last gift amount is 5
3. if they made 5 gifts in the last 36 months, last gift amount is 10, and time since last gift is 5
4. if they made 5 gifts in the last 36 months, last gift amount is 10, time since last gift is 20, and average gift amount for last 36 months is 15