

# **Implementation in Enterprise Miner: Decision Tree with Binary Response**

## **Outline**

- 8.1 Example**
- 8.2 The Options in Tree Node**
- 8.3 Tree Results**
- 8.4 Example Continued**

## **Appendix A: Tree and Missing Values**

## 8.1 Example

1. Select **STA5703.GENERAL** in the **Input Data Source** node.

This data set is from the 10<sup>th</sup> Gvu (graphical Visualization and Usability Center) WWW User Survey. [Copyright 1994-1998 Georgia Tech Research Corporation. @All rights reserved.] The data set was created from the general demographics portion of the survey. There are 5,022 cases (respondents).

**Source:** Gvu's WWW User Survey - [http://www.gvu.gatech.edu/user\\_surveys](http://www.gvu.gatech.edu/user_surveys)

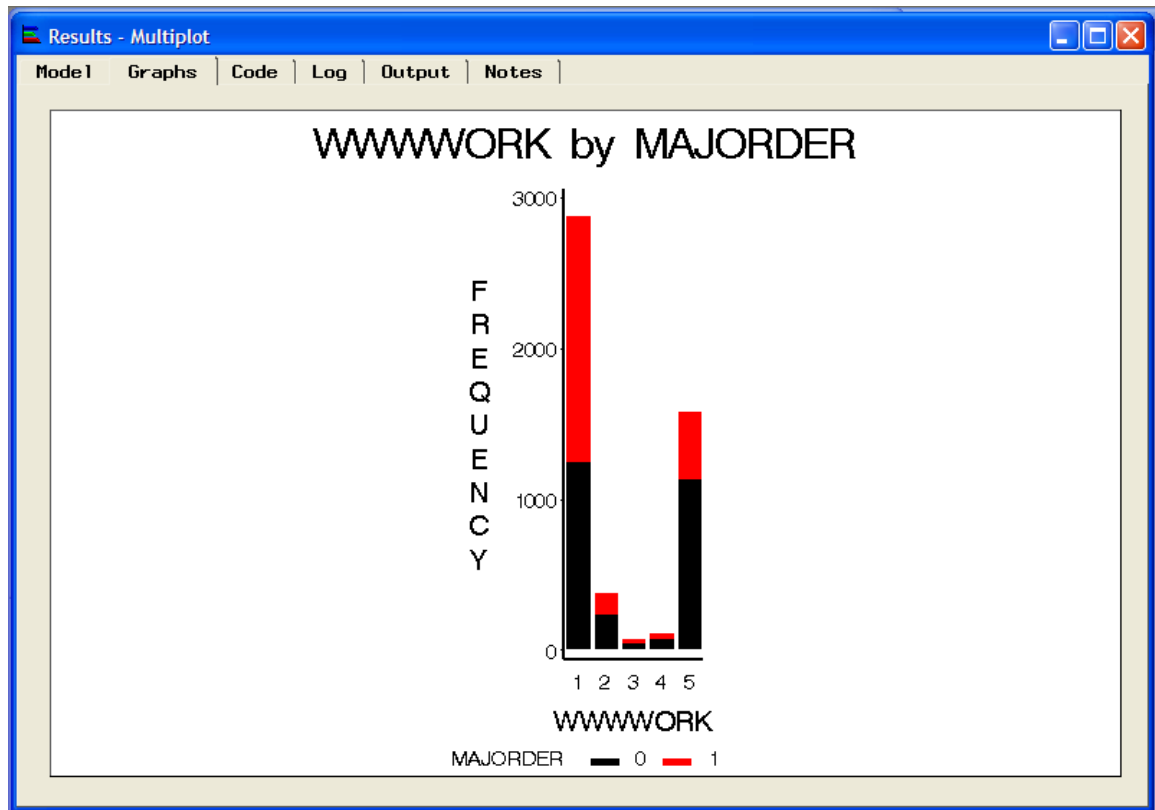
- Set the model roles of MAJORDER and COOKIE to target. Two binary targets were created from the survey:
  - MAJORDER – whether or not a respondent made a purchase online of more than \$100.
  - COOKIE – whether or not a respondent has ever changed their “cookie” preferences.
- Associated with each case are 38 categorical input variables representing demographics. Twelve of the inputs are nominal, nine are ordinal, and 17 are binary. They are all coded as integer values.
- Explore the data within the **Input Data Source** node or add an **Insight** node.

2. Data Exploration

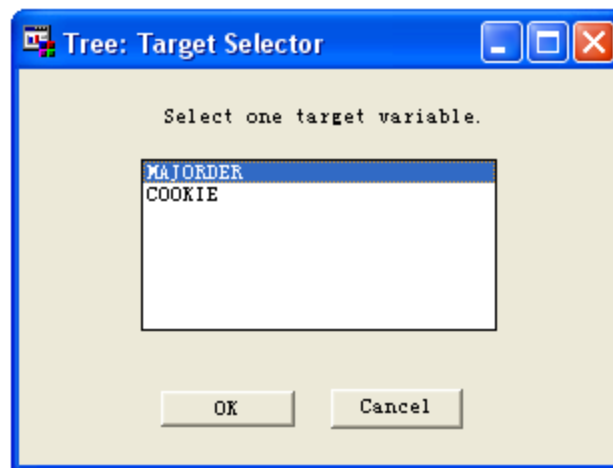
Many tools available in EM enable you to explore your data further. In particular, for data with binary response, the Multiplot node creates a series of histograms and bar charts that enable you to examine the relationships between the input variables and the binary target.

- Right-click on the Multiplot node and select **Run**.
- When prompted, select **Yes** to view the results.

By using the Page Down button on your keyboard, you can view the histograms generated from this data set.



3. Partition the data into 67% for training and 33% for validation in the **Data Partition** node.
4. Add a **Tree** node and open it. The Target Selector window first comes up.



- Select MAJORDER in the Target Selector window.
- The other response COOKIE is set as “don’t use” by default.

## 8.2 The Options in the Tree Node.

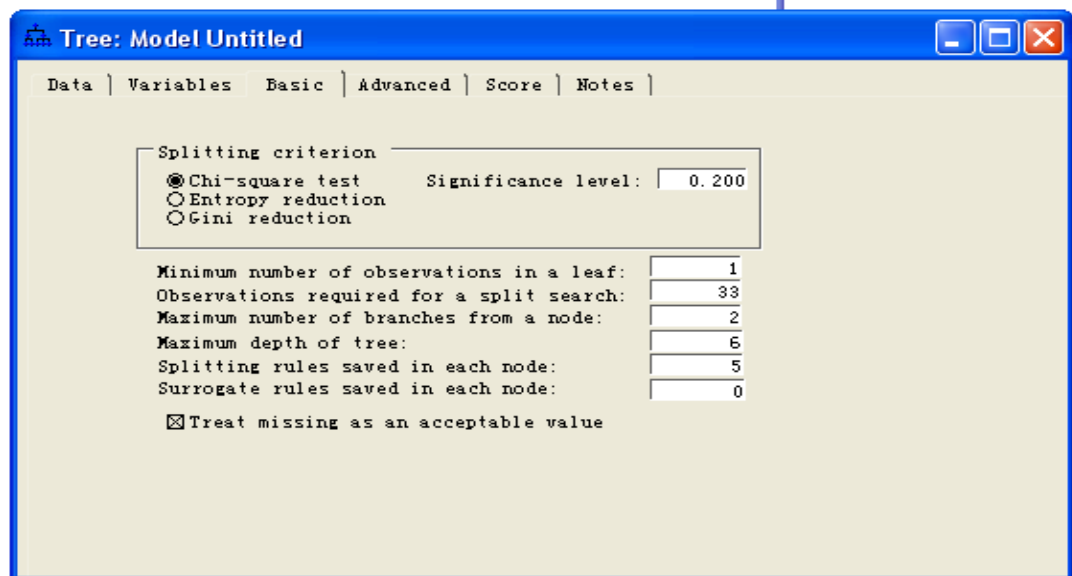
The interface to the **Tree** node is a window with the following tabs: Data Tab, Variables Tab, Basic Tab, Advanced Tab, Score Tab, Notes Tab

- **Data Tab**

The Data tab displays the name and a description of the predecessor data sets that have been assigned to one of these roles:

- Training** -- used to fit the initial tree.
- Validation** -- used by default for assessment of the tree. Because decision trees have the capacity for overtraining, the node also uses the validation data set to retreat to a simpler fit (if necessary) than the fit based solely on the training data set.
- Test** -- additional "hold out" data set that you can use for model assessment.
- Score** -- used for predicting target values for a new data set that may not contain the target. To score the score data set, specify the desired score data source in the Data tab, click the **Score** radio button in the Score tab, and then run the node. You can also choose to score a model with a successor **Score** node instead of scoring with the **Tree** node.

- **Basic Tab**



Use the Basic tab to specify the tree splitting criterion and values related to the size of the tree. The options available depending on the type of responses: categorical or continuous. For nominal or binary targets, you have a choice of three splitting criteria:

- **Chi-Square test** (default) -- the Pearson Chi-Square measure of the target vs. the branch node, with a default **Significance level** of 0.20.
- **Entropy Reduction** -- the reduction in the entropy measure of node impurity.
- **Gini Reduction** -- the reduction in the Gini measure of node impurity.

For ordinal targets, the splitting criteria choices are:

- **Entropy Reduction** (default) -- the reduction in the entropy measure of node impurity.
- **Gini Reduction** -- the reduction in the Gini measure of node impurity.

You can specify the following positive integer values in the Basic tab:

- **Minimum number of observations in a leaf** (default = the larger value of 5 and the total number of observations divided by 10)
- **Observations required for a split search** This option prevents the splitting of nodes with too few observations. That is, nodes with fewer observations than the value specified in **Observations required for a split search** will not be split.
- **Maximum number of branches from a node** (default = 2)
- **Maximum depth of tree** (default = 6)
- **Splitting rules saved in each node** (default = 5)
- **Surrogate rules saved in each node** (default = 0)

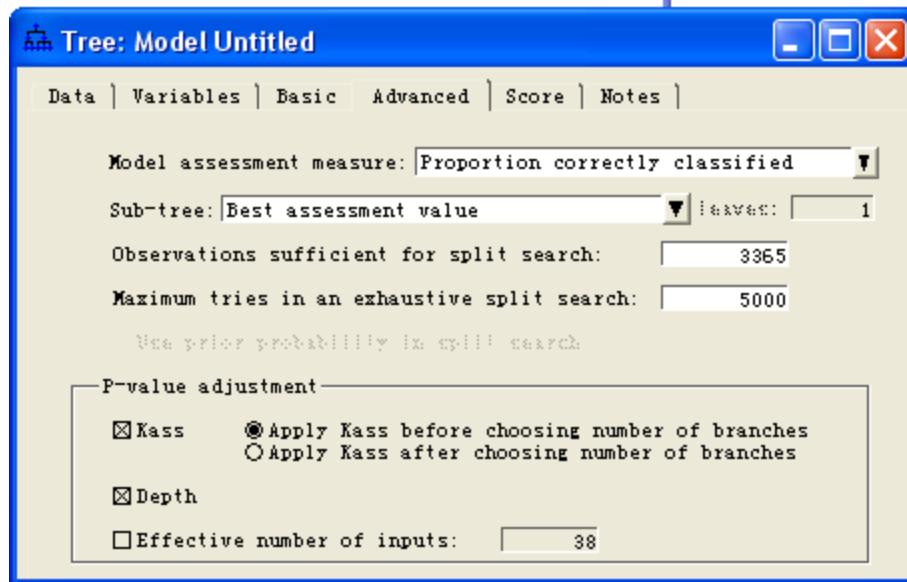
The **Splitting rules saved in each node** option controls the number of splitting rules that are saved for display in the Competing Splits window of the Results Browser. For each node, a number of splitting rules are evaluated as the tree is constructed. A statistic that measures the strength of the rule is computed. If the Chi-square is selected, then the computed statistic is the LOGWORTH. If the Entropy or Gini reduction option is selected, then the computed statistic is WORTH, which measures the reduction in variance for the split. For both LOGWORTH and WORTH, larger values are better. The splitting rules are sorted by LOGWORTH or WORTH depending on the splitting criteria selected. Rules that have a value of WORTH  $\geq 0.05$  are automatically saved.

The **Surrogate rules saved in each node** option specifies the maximum number of surrogate splitting rules to create. A surrogate rule is a back-up to the main splitting rule. The first surrogate rule is invoked when the main splitting rule relies on an input with missing values. If the **first** surrogate also relies on an input with missing values, the **next** surrogate is invoked. If missing values prevent the main rule and **all** of the surrogates from applying to an observation, then the main rule assigns the observation to the branch designated for receiving missing values.

The **Treat missing as an acceptable value option** allows observations with missing values to be used when calculating the logworth/worth of a particular split. If the check box is not selected, only observations with nonmissing values will be used when

calculating the logworth/worth values. For more information about missing values and surrogate rules, read **Appendix A - Tree and Missing Values**.

- **Advanced Tab**



In the Advanced tab, you can set the following options:

- Choose the Model Assessment Measure
- Specify the Subtree Method
- Set the Split Search Criteria
- Choose the P-value Adjustment Method
- Incorporating Profit or Loss in Tree Construction

### **Advanced Tab: Choosing the Model Assessment Measure**

The model assessment measure is used by the node to select the best tree based on the results obtained from the validation data. If validation is not available, the node uses the training data to select the best tree based on the selected assessment measure. The list of available model assessment measures depends on the target measurement scale, and whether you defined a profit, profit with costs (revenue) or loss matrix for the target.

For categorical targets, you can choose the from the following model assessment measures:

- No Profit or Loss Matrix Defined:
  - **Proportion misclassified** -- (default) chooses the tree with the smallest misclassification rate.

- **Ordinal proportion correctly classified** -- (for ordinal targets only) chooses the tree with the best classification rate when weighted for the ordinal distances. Let  $\text{Order}(Y)$  denote the rank order of target value  $Y$ . In this case,  $\text{ORDER}(Y)$  takes on the values 1, 2, 3, ...,  $n$  target levels. The classification rate weighted for ordinal distances is equal to:  
  
$$\frac{\text{Sum over obs. of } (n \text{ target levels} - 1 - |\text{ORDER}(Y) - \text{ORDER}(\text{Predicted}(Y))|)}{n \text{ observations.}}$$
- **Proportion of event in top 10, 25 or 50%** -- chooses the tree that has the highest proportion of the target event in the top  $n\%$  of the data. You use this model assessment criterion when your overall goal is to create the tree with the best lift value.
- **Total leaf impurity (Gini index)** -- chooses the tree with the greatest reduction in leaf impurity (smallest Gini index).
- Defined Profit or Loss Matrix:
  - **Proportion misclassified** -- evaluates the tree based on the misclassification rate.
  - **Average profit/loss** -- (default) chooses the tree that provides the maximum profit or minimal loss. The node automatically recognizes the type of decision matrix that you specified in the target profile. If the decision matrix contains less than two decisions, the **Proportion misclassified** model assessment criterion is actually used to select the best tree.
  - **Average Profit/Loss in top 10, 25 or 50%** -- chooses the tree that provides the maximum average profit or minimum average loss in the top  $n\%$  of the data. You use this model assessment criterion when your overall goal is to create the tree with the best lift value.
  - **Total leaf impurity (Gini index)** -- chooses the tree with the greatest reduction in leaf impurity (smallest Gini index).

## Advanced Tab: Specify the Subtree Method

Use the Advanced tab to select the subtree method used to create the predicted values. Trees tend to grow too large. A large tree that appears to fit the training data well is likely to be fitting random variations in the data, and will probably generate poor predictions on new data. Expert opinion is divided as to how to judge when a tree is large enough.

The **Tree** node evaluates all possible subtrees of the constructed tree, and reports the results for the best subtree for each possible number of leaves. The node supports the following subtree methods:

<b>Best assessment value</b>	The subtree which produces the best results according to the selected <b>Model assessment measure</b> chosen. Validation data is used if it is available.
<b>The most leaves</b>	<b>The most leaves</b> option selects the full tree. This is appropriate when the tree will be constructed interactively, or when other options are relied on for stopping the training.
<b>At most indicated number of leaves</b>	If you select the <b>At most indicated number of leaves</b> subtree method, then the node uses the largest subtree with at most $n$ leaves. To set the number of leaves, type a value in the <b>leaves</b> entry field (the default is 1).

### Advanced Tab: Set the Split Search Criteria

The **Observations sufficient for split search** option sets an upper limit on the number of observations used in the sample to determine a split. By default, a split search uses a within-node sample of 5000 observations. If there are less than 5000 observations in the data set then all observations in the data set are used. More memory and longer CPU times are required when you increase the **Observations sufficient for split search** number. The within-node sample is limited to **32,767** observations (ignoring any frequency variables).

The **Observations sufficient for split search** option in the Advanced tab has a different purpose than the **Observations required for split search** option in the Basic tab. The Basic tab option is used to prevent the splitting of a node that contains too few observations, whereas the Advanced tab option specifies a sample size for determining splits, but does not prevent splitting. The **Observations sufficient for split search** option can be used to reduce memory and CPU usage requirements.

### Advanced Tab: Choose the P-value Adjustment Method

If you selected **Chi-square test** (or **F test** for interval-valued responses) in the Basic tab, you can choose a method to adjust the **p**-values in the Advanced tab. The methods are:

- **Kass** - (checked by default) multiplies the p-value by a Bonferroni factor that depends on the number of branches, target values (Chi-Square), and sometimes on the number of distinct input values. Kass's original CHAID algorithm applies this factor after the split is selected. By default, the **Tree** node applies the factor before the split is selected, and therefore selects the split based on adjusted p-values (the **Apply Kass before choosing number of branches** check box is checked). The KASS adjustment may cause the p-value to become less significant than an alternative method called Gabriel's adjustment. If so, the Gabriel's p-value is used. Click the **Apply Kass after choosing number of branches** radio button for the CHAID way of applying the factor after selecting the split.

- **Depth** - (checked by default) adjusts the final **p**-value for a partition to simultaneously accept all previous partitions used to create the current subset. The CHAID algorithm has multiplicity adjustments within each node, but does not provide a multiplicity adjustment for the number of leaves. If a tree has grown to, say, a thousand leaves, and you do a significance test in each leaf at  $\alpha=0.05$ , a CHAID algorithm will obtain about 50 false rejections. Hence, the tree is likely to grow too big. DEPTH does a Bonferroni adjustment for the number of leaves to correct excessive rejections.
- **Effective number of inputs** - adjusts the **p**-value for the effective number of inputs. Using more inputs increases the likelihood of a random input winning over a truly predictive input. The input adjustment multiplies the **p**-value by the number declared in the **Effective number of inputs** field. The default **Effective number of inputs** equals the number of inputs with status set to **use** in the Variables tab.

By default, the Kass and Depth **p**-value adjustments methods are selected. You may de-select all methods of adjusting the **p**-values, or select only specific methods.

### **Advanced Tab: Incorporating Profit or Loss in Tree Construction**

Like other models, a profit or loss function may be used to evaluate a tree. In some situations, *the profit or loss function may also be used in the construction of the original tree*. This is possible when all of the following is true:

- the target is nominal or ordinal.
- the splitting criterion is set to Gini or entropy reduction.
- the decision alternatives of the loss or profit matrix are the target values.
- the matrix is of type profit or loss (not a profit with costs matrix).

The profit (or loss) matrix modifies the splitting criterion, affecting which splitting rule is chosen.

Explicitly, if the target has values  $x$ ,  $y$ , and  $z$ , and if the predicted value in the node is  $z$ , then the term representing target value  $v$  in the Gini (or entropy) impurity measure is multiplied by a factor approximating the loss of misclassifying  $v$  as  $z$ . Here,  $v$  equals  $x$ ,  $y$ , or  $z$  as appropriate for the term in the Gini or entropy impurity measure.

The approximating loss matrix equals the original loss matrix, in which a constant is added to all of the elements to force the smallest element to be 1. If a profit matrix is specified, then the elements are first multiplied by minus 1.

Therefore, the term representing target  $v$  in the impurity measure is either left unchanged or made larger. The factor increase is proportional to the misclassification cost magnitude. Larger misclassification costs amplify the impurity more than smaller costs.

If the requirements listed above are true, then a **Use profit/loss matrix during split search** check box appears at the bottom of the Advanced tab. To include the profit or loss matrix in the split search, check the box. The default for nominal targets is to not include the matrix. For ordinal targets, the matrix must be included. If you defined a prior vector in the target profile, you can request its inclusion in the split search by checking the **Use prior probability split search** box.

- **Score Tab:**

- **Data subtab**

Use the Data subtab to configure whether or not to score the predecessor data sets listed on the node's Data main tab. By default, training, validation, and test predecessor data sets are not scored. To score them, click the **Training, Validation, and Test** check box, then run the node.

- **Variables subtab**

Use the Variables subtab to select the output variables for subsequent modeling with another node. The variables are output to the data set listed besides the **Data set** field on the Data main tab. The following output choices are available:

1. **Input variable selection**- accepted variables.
2. **Dummy variables** - dummy variables for the leaves (terminal nodes) in the tree. If any observations are assigned to a specific leaf then its associated dummy variable contains a value of 1, otherwise it contains a value of 0. These dummy variables can be viewed as the important "interactions" between variables that can be used for subsequent modeling.  
To create the dummy variables in the scored data sets,
  1. Select the **Process or Score: Training, Validation, and Test** check box in the Data subtab of the Score tab if you want to create the dummy variables in the scored training, validation, or test data sets. Otherwise, they are created only for the score data set.
  2. Select the **Dummy variables** check box in the Variables subtab of the Score tab and run (or re-run) the **Tree** node. If the **Tree** has been run, you can select the **Dummy variables** check box in the New Variables subtab of the Score tab in the Results Browser without re-running the node.
3. **Leaf identification variable** - these variables can be used for group processing. For example, you can pass the leaf identification variable to a successor **Group Processing** node, then use a modeling node (such as **Regression**) to create separate models for each leaf node.
4. **Prediction variables**

### 8.3. Tree Result

Run the tree node using the default settings and view the results of the Tree Node

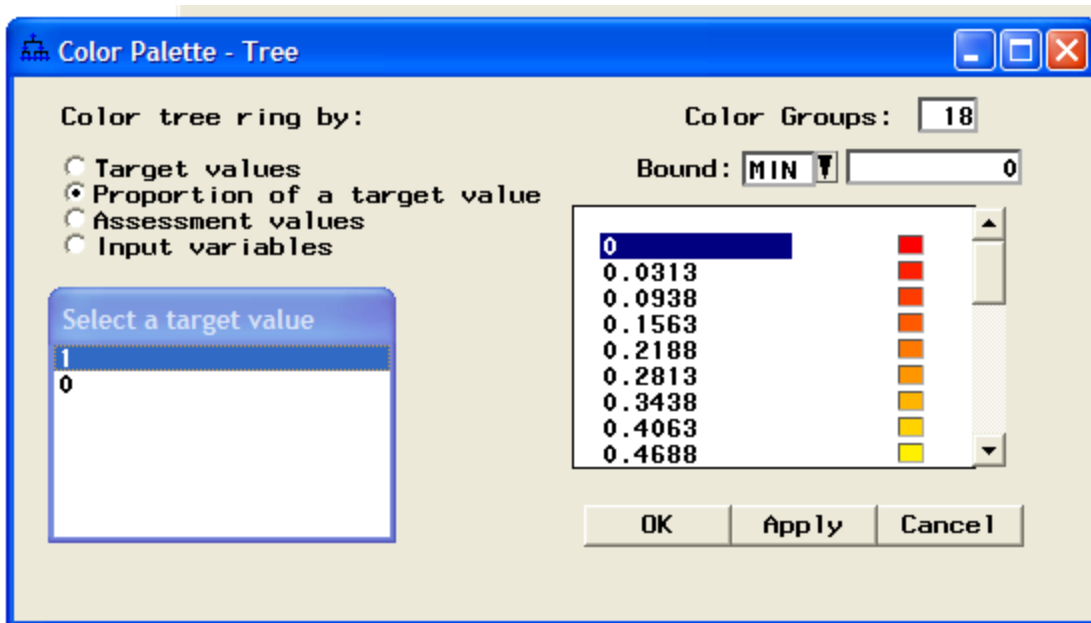
#### (i) The Model Tab

You may view target profile information and the Basic and Advanced tab settings that were used to create the tree in browse mode.

#### (ii) Tree Results Brower: All Tab

The Results Browser opens to the All tab, which contains the following information: **Summary Table**, **Tree Ring Navigator**, **Assessment Table**, and **Assessment Graph**. The Summary Table provides summary statistics for the current tree. The Tree Ring is a graphical display of possible data segments from which to form a tree. To help differentiate the data segments (tree rings), a color legend is displayed in a separate window. By default, the segment color hues in the legend (Assessment Colors window) correspond to the assessment values.

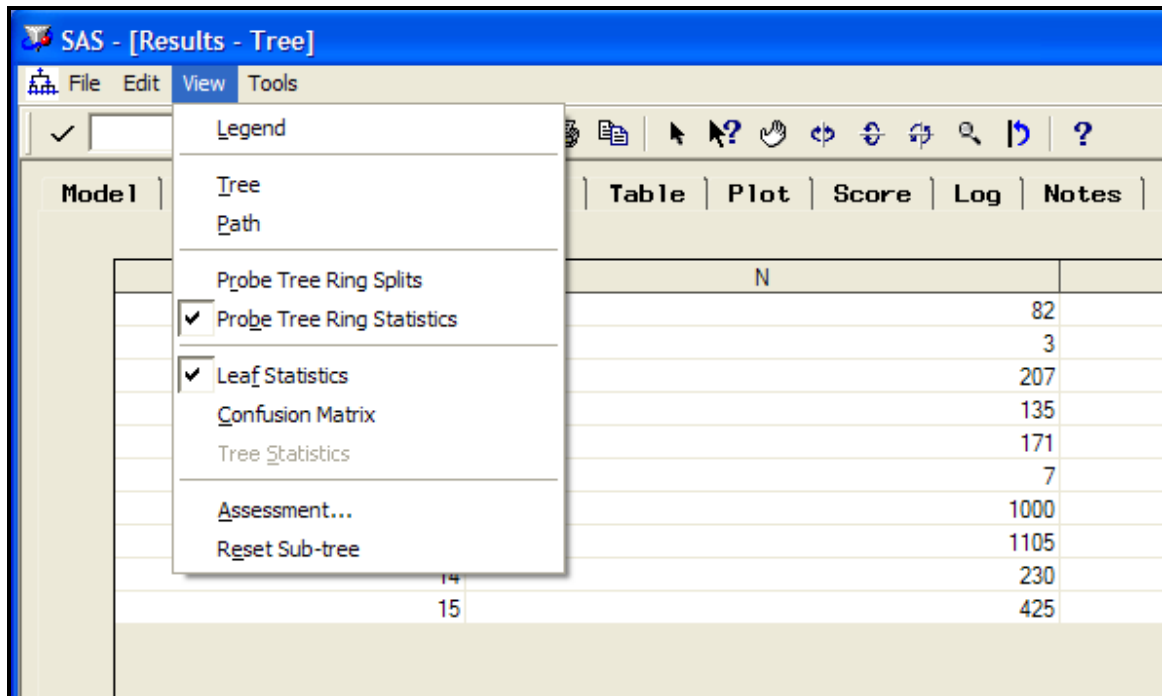
- Tools => Define colors. Color the tree results according to the proportion of the target event (1).



#### (iii) The Summary Tab

For categorical targets, the Summary tab contains a *Confusion Matrix* (default) or Leaf Statistics. Select the type of table you want to view using the **View** pull-down menu. The confusion matrix is basically a contingency comparing the predicted values with the observed responses.

- View => Leaf Statistics



#### (iv) Tree Ring Navigator Tab

The Tree Ring Navigator is a graphical display of tree complexity, split balance, and discriminatory power. Select the Tree Ring tab to see an enlarged version of the Tree Ring.

The center region represents the entire data set (the root node of the tree). The ring surrounding the center represents the initial split. Successive rings represent successive stages of splits. The sizes of displayed segments in one ring are proportional to the number of training observations in the segments.

For nominal targets, the color hues in the Tree Ring segments correspond to the assessment values. The default Tree Ring is segmented by different shades of orange/yellow. The legend window displays the assessment values for each hue. Nodes that have similar assessment values have a similar color.

You can use the tool box to control the Tree Ring. The Tree Ring Navigator also enables you to view specific segments in the Tree Diagram.

**(v) Tree Results Browser: Table Tab**

The screenshot shows the SAS Results - Tree window with the Table tab selected. The window title is "SAS - [Results - Tree]". The menu bar includes File, Edit, View, and Tools. Below the menu bar is a toolbar with various icons. The main content area is titled "Proportion Correctly Classified" and contains a table with three columns: Leaves, Training, and Validation. The table lists 47 rows of data, with the Training and Validation columns containing decimal values representing the proportion of correctly classified instances.

Leaves	Training	Validation
1	0.5456	0.5299
2	0.6312	0.6500
3	0.6312	0.6500
4	0.6455	0.6566
5	0.6455	0.6566
6	0.6568	0.6596
7	0.6660	0.6614
8	0.6710	0.6663
9	0.6802	0.6681
10	0.6752	0.6723
11	0.6844	0.6741
12	0.6877	0.6759
13	0.6877	0.6759
14	0.6895	0.6765
15	0.6895	0.6765
16	0.6895	0.6765
17	0.6895	0.6765
18	0.6895	0.6765
19	0.6895	0.6765
20	0.6895	0.6765
21	0.6895	0.6765
22	0.6895	0.6765
23	0.6895	0.6765
24	0.6895	0.6765
25	0.6895	0.6765
26	0.6895	0.6765
27	0.6895	0.6765
28	0.6895	0.6765
29	0.6895	0.6765
30	0.6895	0.6765
31	0.6895	0.6765
32	0.6895	0.6765
33	0.6895	0.6765
34	0.6895	0.6765
35	0.6895	0.6765
36	0.6895	0.6765
37	0.6895	0.6765
38	0.6895	0.6765
39	0.6895	0.6765
40	0.6895	0.6765
41	0.6921	0.6747
42	0.6921	0.6747
43	0.6921	0.6747
44	0.6921	0.6747
45	0.6921	0.6747
46	0.6948	0.6741
47	0.6978	0.6717

The Table tab provides a measure of how well the tree describes the data. Assessment statistics for each subtree are listed for the training and validation data sets. Obviously, validation statistics are not listed if you have no validation data set.

The default assessment values displayed in the table depend on the **Model assessment measure** configured in the Advanced tab. The table displays the assessment for several candidate partitions of the data. If a validation data set is used, the assessment based on the validation data will be more reliable than that based on the training data.

The Table lists the assessment value for the training and validation data for each subtree. By default, the tree with the highest assessment value for validation and the fewest number of segments is highlighted in the Assessment Table. The vertical reference line in the Assessment Graph corresponds to this tree. When you select a different tree in the Assessment Table, the reference line in the Assessment Graph automatically moves to that number of segments, and the Tree Ring Navigator and Summary Table are updated to reflect the new tree.

**(vi) Tree Results Browser: Plot Tab**

The Plot tab displays a plot of the assessment values on the vertical axis for the different subtrees. The **Tree** node automatically chooses the subtree that optimizes the model assessment measure that you chose in the Advanced tab. The vertical reference line identifies this subtree.

To set the vertical axis statistic for the assessment plot, right-click the plot, and choose one of the following pop-up menu items:

- For categorical targets:
  - No Profit or Loss Matrix Defined:
    - **Proportion classified correctly**
    - **Ordinal proportion correctly classified** (ordinal targets only)
    - **Proportion of event in either the top 10, 25, or 50%**
    - **Total leaf impurity (Gini index)**
  - Defined Profit or Loss Matrix:
    - **Proportion classified correctly**
    - **Average profit or loss**
    - **Average profit or loss in either the top 10, 25, or 50%**
    - **Total leaf impurity (Gini index)**

**(vii) Score Tab:**

- **Variables Selection subtab**

The Variable Selection subtab lists a table of input variables, measures of importance, variable roles, variable labels and number of splitting rules using the variable. The total variable importance of an input  $X$  is equal to the square root of the sum over nodes of agreement multiplied by the sum of square errors. For a categorical target, the sum of square errors is equal to the Gini index.

Input variables that are assigned the input role can be used as model input by the successor modeling node. The **Tree** node automatically assigns the **input** model role to those variables that have a value of importance greater than or equal to 0.05. All other variables are assigned the **rejected** role. To manually change the model role for a variable, right-click on the desired cell of the Role column, select the Set Role column, and then select either **input** or **rejected**.

By default, **Exported role as indicated in this table** check box is selected and variables are exported with roles as shown in this table. If you deselect this check box, variables are exported to successor nodes with the same roles as they came into the **Tree** node.

- **New Variables subtab**

By default, **Leaf identification variables** and **Prediction variables** are exported to the data sets listed in the Data subtab. You can also include the **Dummy variables** by selecting the corresponding check box.



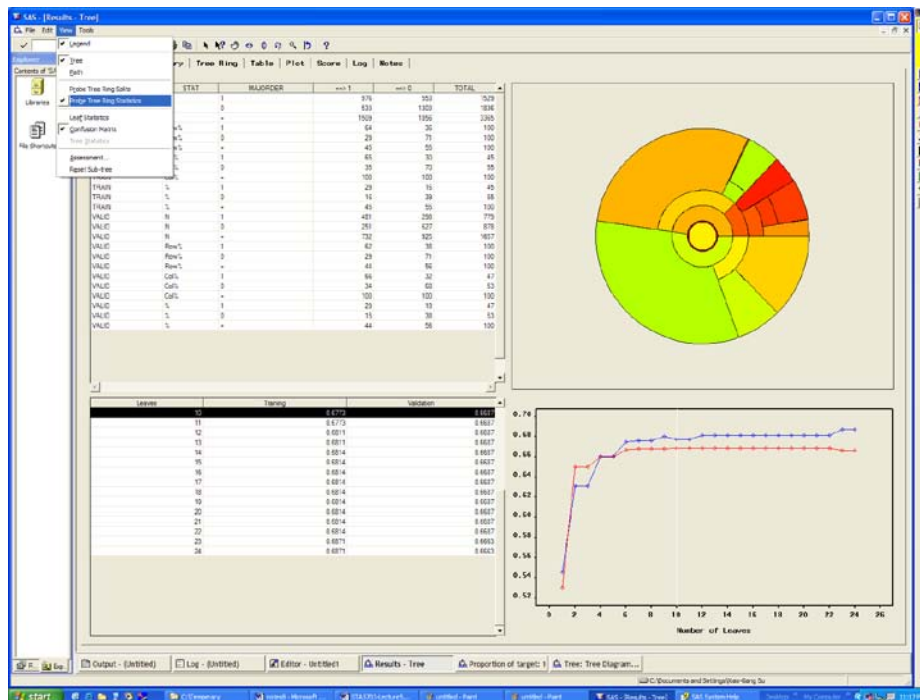
- a. View => Select “Diagram-node types” from the pop-up menu. Select the button for leaves, variables, and values.
- b. View => Statistics. Set the select value to NO for all rows except MAJORDER values: 1, Count per class, and N in node.
- c. Right click to access the pop-up menu item => View Competing Splits.

A table view of **Variable, Logworth, Groups, and Label** is displayed. The measure of worth indicates how well a variable divides the data into each class. Good splitting variables have large values of Logworth.

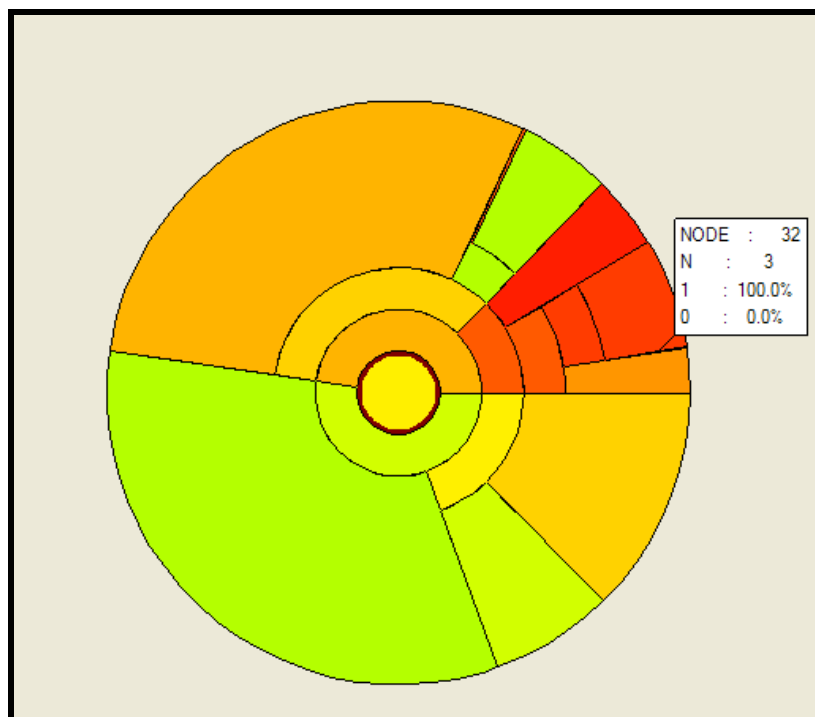
- d. Right click => View Surrogate Splits.

If you select the **View surrogate** splits pop-up menu item in the tree diagram, a table view of **Variable, Logworth, Groups, and Label** is displayed. Before you can view surrogate splits in the tree diagram, you must specify that surrogate rules should be saved in each node in the Basic tab. The measure of agreement is the proportion of training observations that the main splitting rule and the surrogate rule assign to the same branch. Observations with a missing value in the main splitting variable are excluded from the evaluation of agreement. If the value of an observation is not missing on the main splitting variable but is missing on the surrogate variable, then that observation counts against agreement. If several surrogate rules exist, they are applied in order of decreasing agreement with the main rule. If none of the rules can be applied, the observation will be assigned to the branch that is designated for missing values.

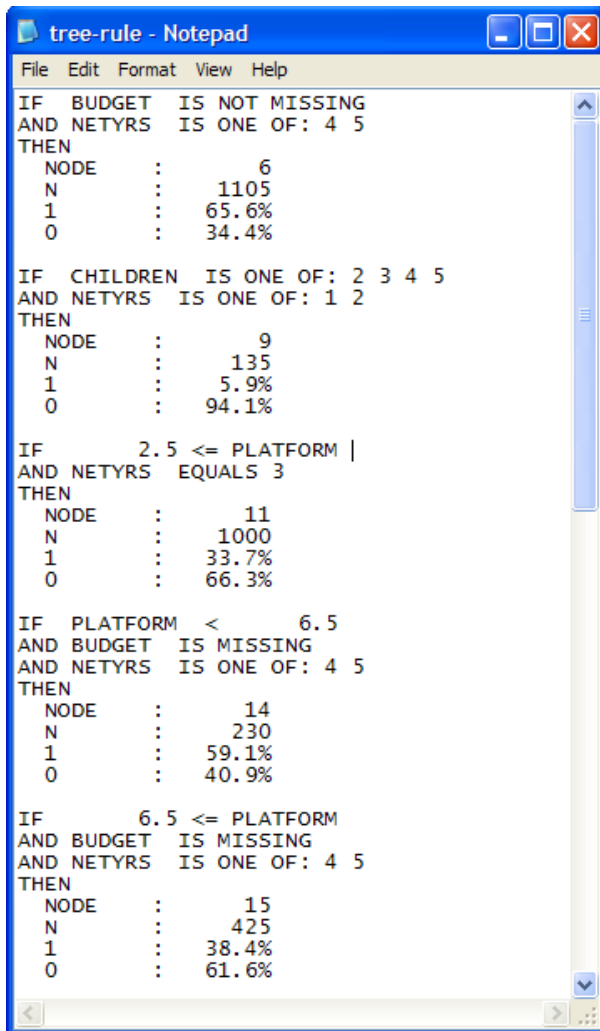
- Go back to Results – Tree and View => Probe Tree Ring Statistics



Point your mouse to any of the terminal nodes. You may find one of them has 3 observations only, which results from an imbalanced split.



- **File => Save Rules.** Save the decision tree rules as a text file and open the saved file using Notepad.



```
tree-rule - Notepad
File Edit Format View Help
IF BUDGET IS NOT MISSING
AND NETYRS IS ONE OF: 4 5
THEN
  NODE      :      6
  N         :    1105
  1         :    65.6%
  0         :    34.4%

IF CHILDREN IS ONE OF: 2 3 4 5
AND NETYRS IS ONE OF: 1 2
THEN
  NODE      :      9
  N         :    135
  1         :     5.9%
  0         :    94.1%

IF      2.5 <= PLATFORM |
AND NETYRS EQUALS 3
THEN
  NODE      :     11
  N         :   1000
  1         :    33.7%
  0         :    66.3%

IF PLATFORM < 6.5
AND BUDGET IS MISSING
AND NETYRS IS ONE OF: 4 5
THEN
  NODE      :     14
  N         :    230
  1         :    59.1%
  0         :    40.9%

IF      6.5 <= PLATFORM
AND BUDGET IS MISSING
AND NETYRS IS ONE OF: 4 5
THEN
  NODE      :     15
  N         :    425
  1         :    38.4%
  0         :    61.6%
```

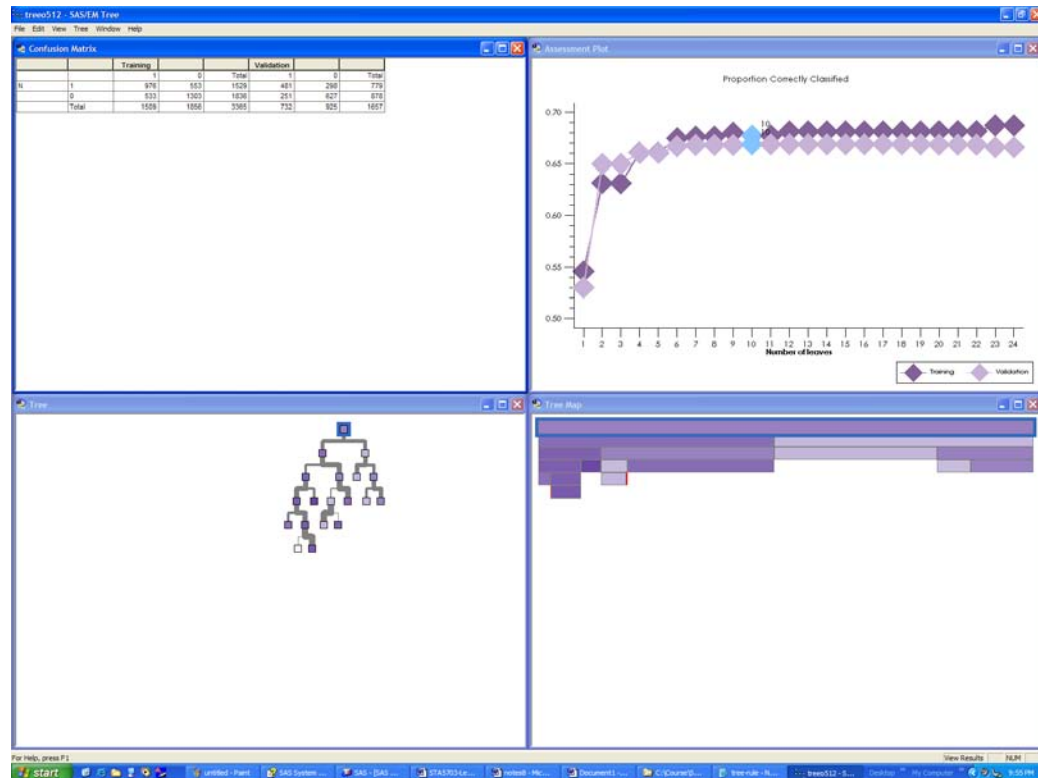
- **New Tree Viewer**

A new tree viewer will be available in a future version of EM. To obtain access to this new viewer,

- a. In the command bar, type the statement **%let emv4tree=1.**



- b. Press the return key.
- c. Return to the Enterprise Miner window.
- d. Right-click on the Tree node and select **New View...**



- Click on the label for the tree in the diagram, and change the label to **Default**.
- Add another **Tree** node to the workspace and connect the Data Partition node to the Tree node.
- Change the option settings in the Tree node.
- Add more Tree nodes and experiment different options in the Basic and Advanced tabs in Tree node.
- **Comparing Tree Models**
  - Add an Assessment node
  - Connect all Tree nodes to the Assessment node.
  - Right-click on the Assessment node and select **Run**.
  - When prompted, select **Yes** to view the results.
  - In the Assessment Tool window, click and drag to select both of the models.
  - Select **Tools** => **Lift Chart**.

## **Appendix A: Tree and Missing Values**

The search for a split on an input uses observations whose values are missing on the input. These observations are assigned to the same branch. The branch may or may not contain other observations. The branch chosen is the one that makes the split worth the most.

For splits on a categorical variable, missing values are treated as a separate category. For numerical variables, missing values are treated as having the same unknown non-missing value.

One advantage of using missing data during the search is that the worth of split is computed with the same number of observations for each input. Another advantage is that an association of the missing values with the target values can contribute to the predictive ability of the split.

When a split is applied to an observation where the required input value is missing, surrogate splitting rules are considered before assigning the observation to the branch for missing values.

A surrogate splitting rule is a back-up to the main splitting rule. For example, the main splitting rule might use COUNTY as input and the surrogate might use REGION. If the COUNTY is unknown and the REGION is known the surrogate is used.

If several surrogate rules exist, each surrogate is considered in sequence until one can be applied to the observation. If none can be applied, the main rule assigns the observation to the branch designated for missing values.

The surrogates are considered in the order of their agreement with the main splitting rule. The agreement is measured as the proportion of training observations it and the main rule assign to the same branch. The measure excludes the observations that the main rule cannot be applied to. Among the remaining observations, those where the surrogate rule cannot be applied count as observations not assigned to the same branch. Thus, an observation with a missing value on the input used in the surrogate rule (but not the input used in the primary rule) counts against the surrogate.

The **Surrogate rules saved in each node** option in the Basic tab determines the number of surrogates sought. A surrogate is discarded if its agreement is  $\leq 1/B$ , where B is the number of branches. As a consequence, a node might have less surrogates than the number specified in the **Surrogate rules saved in each node** option.

When you are training or validating the tree, observations with missing target values are ignored. There are two aspects to the missing value problem. First, some cases in the training sample may have missing values on the measurement vector. Second, we need to have complete set of measurements to score a new case. Enterprise Miner treats the missing value as a new nominal level to solve both types of missing value imputation problem in decision tree modeling.

- **Nominal Variable:** A nominal input variable with L levels becomes to L+1 levels and the extra level comes from the missing value. If a new case has a missing value on a splitting variable, then this case is send to whatever branch contains the missing values.
- **Ordinal Variable:** For an ordinal input variable, it also treats the missing value as a separate level. However, we can not place any order on the missing value. This leads to the necessity to modify the split search process and to treat the missing value as a nominal level. Obviously, this will increase the number of possible partitions significantly.

Treat missing value as a new nominal level can solve both types of missing value imputation problem. There are other solutions available such as "Surrogate Splits" as well. "Surrogate splits" method was introduced by BFOS (1980). The idea is "Define a measure of similarity between any two splits  $s_1$  and  $s_2$  of a node t. Assume that the best split on node d is  $s_1$  that is on variable  $x_m$ . Assume that the split  $s_2$  is most similar to split  $s_1$  on the variable other than  $x_m$ . We call  $s_2$  the surrogate for  $s_1$ . Similarly, we can define the second best surrogate and so on. If a case has  $x_m$  missing in its measurement, we can use the best surrogate split. If it has missing value on the variable used in the best surrogate split, we can use the second best surrogate and so on. We can specify surrogate splits options in the Basic tab of tree node in Enterprise Miner as well.

Treat missing value as an extra nominal level increases the number of possible partitions significantly. Use surrogate split increases the complexity of the score code significantly. Consequently, it increases the computation time to score a new case dramatically as well. Currently, it is still not clear which method is better.

### Example 5.10

Suppose that A is a nominal input variable with L levels. How many more splits are necessary if some cases have missing values on A?

<Solutions>

$$\text{Differences} = B_{L+1} - B_L = \sum_{i=0}^{L+1} S(L+1, i) - \sum_{i=0}^L S(L, i)$$

When L = 8, the difference = 21146 - 4139 = 17007 and when L = 3, the difference is 10 (= 14 - 4).

### Example 5.11

List all possible partitions for an ordinal variable with 3 levels and one missing level.

<Solutions>

Two-way splits: 1-23?, 12-3?, 123-?, 1?-23, 12?-3,  
 Three-way splits: 1-2-3?, 1-2?-3, 1?-2-3, 12-3-?, 1-23-?  
 Four-way split: 1-2-3-?

Thus, there are 11 possible splits. There are  $2^{L-2}(L+3)-1$  possible partitions for an ordinal variable with L levels and one missing level.

**Example 5.12**

Prove that the number for the possible partitions for an ordinal variable with L levels and one missing level is  $2^{L-2}(L+3)-1$ .

<Solutions>

The number of possible B-way partitions for an ordinal variable of L levels is  $\binom{L-1}{B-1}$

and the number of possible (B-1)-way partitions is  $\binom{L-1}{B-2}$ . Suppose we add one

nominal level to represent the missing value

(1) Treat the missing as a single new class to each of the original (B-1)-way partitions:

this adds  $\binom{L-1}{B-2}$  B-way partitions

(2) Add the missing value to any one class of the B-way partitions: this adds  $B \binom{L-1}{B-1}$ .

Thus, we have  $B \binom{L-1}{B-1} + \binom{L-1}{B-2}$  B-way partitions.

The total number of partitions is  $\sum_{i=2}^L \left( i \binom{L-1}{i-1} + \binom{L-1}{i-2} \right) = 2^{L-2}(L+3)-1$ .

**Appendix B: Normality Test in HW 3**

Add a SAS Code node and connect it to the Clustering node. And then run the following SAS program in the SAS Code node.

```
PROC SORT;
BY _SEGMNT_;
RUN;

PROC UNIVARIATE NORMAL PLOT;
VAR MORTDUE;
BY _SEGMNT_;
RUN;
```

Last time, I wrongly specified **NORMAL** as **NORM**, which caused the failure of the execution.