Chapter 3  Predictive Modeling Using Regression

3.1  Introduction to Regression............................................................................................3-3

3.2  Regression in Enterprise Miner ....................................................................................3-8
3.1 Introduction to Regression

**Objectives**

- Describe linear and logistic regression.
- Explore data issues associated with regression.
- Discuss variable selection methods.
### Linear versus Logistic Regression

<table>
<thead>
<tr>
<th>Linear Regression</th>
<th>Logistic Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target is an interval variable.</td>
<td>Target is a discrete (binary or ordinal) variable.</td>
</tr>
<tr>
<td>Input variables have any measurement level.</td>
<td>Input variables have any measurement level.</td>
</tr>
<tr>
<td>Predicted values are the mean of the target variable at the given values of the input variables.</td>
<td>Predicted values are the probability of a particular level(s) of the target variable at the given values of the input variables.</td>
</tr>
</tbody>
</table>

The Regression node in Enterprise Miner does either linear or logistic regression depending upon the measurement level of the target variable.

Linear regression is done if the target variable is an interval variable. In linear regression the model predicts the mean of the target variable at the given values of the input variables.

Logistic regression is done if the target variable is a discrete variable. In logistic regression the model predicts the probability of a particular level(s) of the target variable at the given values of the input variables. Because the predictions are probabilities, which are bounded by 0 and 1 and are not linear in this space, the probabilities must be transformed in order to be adequately modeled. The most common transformation for a binary target is the logit transformation. Probit and complementary log-log transformations are also available in the regression node.
Recall that one assumption of logistic regression is that the logit transformation of the probabilities of the target variable results in a linear relationship with the input variables.
Regression uses only full cases in the model. This means that any case, or observation, that has a missing value will be excluded from consideration when building the model. As discussed earlier, when there are many potential input variables to be considered, this could result in an unacceptably high loss of data. Therefore, when possible, missing values should be imputed prior to running a regression model.

Other reasons for imputing missing values include the following:

- Decision trees handle missing values directly, whereas regression and neural network models ignore all observations with missing values on any of the input variables. It is more appropriate to compare models built on the same set of observations. Therefore, before doing a regression or building a neural network model, you should perform data replacement, particularly if you plan to compare the results to results obtained from a decision tree model.

- If the missing values are in some way related to each other or to the target variable, the models created without those observations may be biased.

- If missing values are not imputed during the modeling process, observations with missing values cannot be scored with the score code built from the models.
There are three variable selection methods available in the Regression node of Enterprise Miner.

**Forward** first selects the best one-variable model. Then it selects the best two variables among those that contain the first selected variable. This process continues until it reaches the point where no additional variables have a \( p \)-value less than the specified entry \( p \)-value.

**Backward** starts with the full model. Next, the variable that is least significant, given the other variables, is removed from the model. This process continues until all of the remaining variables have a \( p \)-value less than the specified stay \( p \)-value.

**Stepwise** is a modification of the forward selection method. The difference is that variables already in the model do not necessarily stay there. After each variable is entered into the model, this method looks at all the variables already included in the model and deletes any variable that is not significant at the specified level. The process ends when none of the variables outside the model has a \( p \)-value less than the specified entry value and every variable in the model is significant at the specified stay value.

The specified \( p \)-values are also known as *significance levels*. 
3.2 Regression in Enterprise Miner

Objectives
- Conduct missing value imputation.
- Examine transformations of data.
- Generate a regression model.
Imputation, Transformation, and Regression

The data for this example is from a nonprofit organization that relies on fundraising campaigns to support their efforts. After analyzing the data, a subset of 19 predictor variables was selected to model the response to a mailing. Two response variables were stored in the data set. One response variable related to whether or not someone responded to the mailing (TARGET_B), and the other response variable measured how much the person actually donated in U.S. dollars (TARGET_D).

<table>
<thead>
<tr>
<th>Name</th>
<th>Model Role</th>
<th>Measurement Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGE</td>
<td>Input</td>
<td>Interval</td>
<td>Donor's age</td>
</tr>
<tr>
<td>AVG GIFT</td>
<td>Input</td>
<td>Interval</td>
<td>Donor's average gift</td>
</tr>
<tr>
<td>CARD GIFT</td>
<td>Input</td>
<td>Interval</td>
<td>Donor's gifts to card promotions</td>
</tr>
<tr>
<td>CARD PROM</td>
<td>Input</td>
<td>Interval</td>
<td>Number of card promotions</td>
</tr>
<tr>
<td>FED GOV</td>
<td>Input</td>
<td>Interval</td>
<td>% of household in federal government</td>
</tr>
<tr>
<td>FIRST TT</td>
<td>Input</td>
<td>Interval</td>
<td>Elapsed time since first donation</td>
</tr>
<tr>
<td>GENDER</td>
<td>Input</td>
<td>Binary</td>
<td>F=female, M=Male</td>
</tr>
<tr>
<td>HOME OWN R</td>
<td>Input</td>
<td>Binary</td>
<td>H=homeowner, U=unknown</td>
</tr>
<tr>
<td>ID CODE</td>
<td>ID</td>
<td>Nominal</td>
<td>ID code, unique for each donor</td>
</tr>
<tr>
<td>INCOME</td>
<td>Input</td>
<td>Ordinal</td>
<td>Income level (integer values 0-9)</td>
</tr>
<tr>
<td>LAST TT</td>
<td>Input</td>
<td>Interval</td>
<td>Elapsed time since last donation</td>
</tr>
<tr>
<td>LOCAL GOV</td>
<td>Input</td>
<td>Interval</td>
<td>% of household in local government</td>
</tr>
<tr>
<td>MALE MIL I</td>
<td>Input</td>
<td>Interval</td>
<td>% of household males active in the military</td>
</tr>
<tr>
<td>MALE VET</td>
<td>Input</td>
<td>Interval</td>
<td>% of household male veterans</td>
</tr>
<tr>
<td>NUM PROM</td>
<td>Input</td>
<td>Interval</td>
<td>Total number of promotions</td>
</tr>
<tr>
<td>PC OWNERS</td>
<td>Input</td>
<td>Binary</td>
<td>Y=donor owns computer (missing otherwise)</td>
</tr>
<tr>
<td>PETS</td>
<td>Input</td>
<td>Binary</td>
<td>Y=donor owns pets (missing otherwise)</td>
</tr>
<tr>
<td>STATE GOV</td>
<td>Input</td>
<td>Interval</td>
<td>% of household in state government</td>
</tr>
<tr>
<td>TARGET B</td>
<td>Target</td>
<td>Binary</td>
<td>1=donor to campaign, 0=did not contribute</td>
</tr>
<tr>
<td>TARGET_D</td>
<td>Target</td>
<td>Interval</td>
<td>Dollar amount of contribution to campaign</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>----------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>TIMELAG</td>
<td>Input</td>
<td>Interval</td>
<td>Time between first and second donation</td>
</tr>
</tbody>
</table>

The variable TARGET_D is not considered in this chapter, so its model role will be set to **Rejected**.

A card promotion is one where the charitable organization sends potential donors an assortment of greeting cards and requests a donation for them.

The MYRAW data set in the CRSSAMP library contains 6,974 observations for building and comparing competing models. This data set will be split equally into training and validation data sets for analysis.

**Building the Initial Flow and Identifying the Input Data**

1. Open a new diagram by selecting **File ⇒ New ⇒ Diagram**.
2. On the Diagrams subtab, name the new diagram by right-clicking on **Untitled** and selecting **Rename**.
3. Name the new diagram **Non-Profit**.
4. Add an **Input Data Source** node to the diagram workspace by dragging the node from the toolbar or from the Tools tab.
5. Add a **Data Partition** node to the diagram and connect it to the Input Data Source node.
6. To specify the input data, double-click on the **Input Data Source** node.
7. Click on **Select...** in order to choose the data set.
8. Click on the **CRSSAMP** and select **CRSSAMP** from the list of defined libraries.
9. Select the **MYRAW** data set from the list of data sets in the CRSSAMP library and then select **OK**.
Observe that this data set has 6,974 observations (rows) and 21 variables (columns). Evaluate (and update, if necessary) the assignments that were made using the metadata sample.

1. Click on the **Variables** tab to see all of the variables and their respective assignments.

2. Click on the **Name** column heading to sort the variables by their name. A portion of the table showing the first 10 variables is shown below.

The first several variables (AGE through FIRSTT) have the measurement level **interval** because they are numeric in the data set and have more than 10 distinct levels in the metadata sample. The model role for all **interval** variables is set to **input** by default. The variables GENDER and HOMEOWNR have the measurement level **binary** because they have only two different nonmissing levels in the metadata sample. The model role for all **binary** variables is set to **input** by default.
The variable IDCODE is listed as a nominal variable because it is a character variable with more than two nonmissing levels in the metadata sample. Furthermore, because it is nominal and the number of distinct values is at least 2000 or greater than 90% of the sample size, the IDCODE variable has the model role **id**. If the ID value had been stored as a number, it would have been assigned an **interval** measurement level and an **input** model role.

The variable INCOME is listed as an ordinal variable because it is a numeric variable with more than two but no more than ten distinct levels in the metadata sample. All ordinal variables are set to have the **input** model role.

Scroll down to see the rest of the variables.

The variables PCOWNERS and PETS both are identified as **unary** for their measurement level. This is because there is only one nonmissing level in the metadata sample. It does not matter in this case whether the variable was character or numeric, the measurement level is set to **unary** and the model role is set to **rejected**.

These variables do have useful information, however, and it is the way in which they are coded that makes them seem useless. Both variables contain the value `Y` for a person if the person has that condition (pet owner for PETS, computer owner for PCOWNERS) and a missing value otherwise. Decision trees handle missing values directly, so no data modification needs to be done for fitting a decision tree; however, neural networks and regression models ignore any observation with a missing value, so you will need to recode these variables to get at the desired information. For example, you can recode the missing values as a `U`, for unknown. You do this later using the Replacement node.

**Identifying Target Variables**

Note that the variables TARGET_B and TARGET_D are the response variables for this analysis. TARGET_B is binary even though it is a numeric variable since there are only two non-missing levels in the metadata sample. TARGET_D has the interval measurement level. Both variables are set to have the **input** model role (just like...
any other binary or interval variable). This analysis will focus on TARGET_B, so you need to change the model role for TARGET_B to target and the model role TARGET_D to rejected because you should not use a response variable as a predictor.

1. Right-click in the Model Role column of the row for TARGET_B.
2. Select Set Model Role ⇒ target from the pop-up menu.
3. Right-click in the Model Role column of the row for TARGET_D.
4. Select Set Model Role ⇒ rejected from the pop-up menu.

**Inspecting Distributions**

You can inspect the distribution of values in the metadata sample for each of the variables. To view the distribution of TARGET_B:

1. Right-click in the name column of the row for TARGET_B.
2. Select View distribution of TARGET_B.

![Distribution of TARGET_B](image)

Investigate the distribution of the unary variables, PETS and PCOWNERS. What percentage of the observations have pets? What percentage of the observations own personal computers? Recall that these distributions depend on the metadata sample. The numbers may be slightly different if you refresh your metadata sample; however, these distributions are only being used for a quick overview of the data.

Evaluate the distribution of other variables as desired. For example, consider the distribution of INCOME. Some analysts would assign the interval measurement level to this variable. If this were done and the distribution was highly skewed, a transformation of this variable may lead to better results.
Modifying Variable Information

Earlier you changed the model role for TARGET_B to \texttt{target}. Now modify the model role and measurement level for PCOWNERS and PETS.

1. Click and drag to select the rows for PCOWNERS and PETS.

2. Right-click in the Model Role column for one of these variables and select \texttt{Set Model Role} $\Rightarrow$ \texttt{input} from the pop-up menu.

3. Right-click in the measurement column for one of these variables and select \texttt{Set Measurement} $\Rightarrow$ \texttt{binary} from the pop-up menu.

Understanding the Target Profiler for a Binary Target

When building predictive models, the "best" model often varies according to the criteria used for evaluation. One criterion might suggest that the best model is the one that most accurately predicts the response. Another criterion might suggest that the best model is the one that generates the highest expected profit. These criteria can lead to quite different results.

In this analysis, you are analyzing a binary variable. The accuracy criteria would choose the model that best predicts whether someone actually responded; however, there are different profits and losses associated with different types of errors. Specifically, it costs less than a dollar to send someone a mailing, but you receive a median of $13.00 from those that respond. Therefore, to send a mailing to someone that would not respond costs less than a dollar, but failing to mail to someone that would have responded costs over $12.00 in lost revenue.

In the example shown here, the median is used as the measure of central tendency. In computing expected profit, it is theoretically more appropriate to use the mean.

In addition to considering the ramifications of different types of errors, it is important to consider whether or not the sample is representative of the population. In your sample, almost 50% of the observations represent responders. In the population, however, the response rate was much closer to 5% than 50%. In order to obtain appropriate predicted values, you must adjust these predicted probabilities based on the prior probabilities. In this situation, accuracy would yield a very poor model because you would be correct approximately 95% of the time in concluding that nobody will respond. Unfortunately, this does not satisfactorily solve your problem of trying to identify the "best" subset of a population for your mailing.

In the case of rare target events, it is not uncommon to oversample. This is because you tend to get better models when they are built on a data set that is more balanced with respect to the levels of the target variable.

Using the Target Profiler

When building predictive models, the choice of the "best" model depends on the criteria you use to compare competing models. Enterprise Miner allows you to specify information about the target that can be used to compare competing models. To generate a target profile for a variable, you must have already set the model role
for the variable to target. This analysis focuses on the variable TARGET_B. To set up the target profile for this TARGET_B, proceed as follows:

1. Right-click over the row for TARGET_B and select **Edit target profile**…

2. When the message stating that no target profile was found appears, select **Yes** to create the profile.

The target profiler opens with the Profiles tab active. You can use the default profile or you can create your own.

3. Select **Edit -> Create New Profile** to create a new profile.

4. Type **My Profile** as the description for this new profile (currently named Profile1).

5. To set the newly created profile for use, position your cursor in the row corresponding to your new profile in the Use column and right-click.

6. Select **Set to use**.
The values stored in the remaining tabs of the target profiler may vary according to which profile is selected. Make sure that your new profile is selected before examining the remainder of the tabs.

7. Select the **Target** tab.

This tab shows that TARGET_B is a binary target variable that uses the BEST12 format. It also shows that the two levels are sorted in descending order, and that the first listed level and modeled event is level 1 (the value next to Event).

8. To see the levels and associated frequencies for the target, select **Levels...**. Close the Levels window when you are done.
9. To incorporate profit and cost information into this profile, select the **Assessment Information** tab.

![GrabScreen.png](https://example.com/GrabScreen.png)

By default, the target profiler assumes you are trying to maximize profit using the default profit vector. This profit vector assigns a profit of 1 for each responder you correctly identify and a profit of 0 for every nonresponder you predict to respond. In other words, the best model maximizes accuracy. You can also build your model based on loss, or you can build it to minimize misclassification.

10. For this problem, create a new profit matrix by right-clicking in the open area where the vectors and matrices are listed and selecting **Add**.

![GrabScreen.png](https://example.com/GrabScreen.png)

A new matrix is formed. The new matrix is the same as the default profit matrix, but you can edit the fields and change the values, if desired. You can also change the name of the matrix.

11. Type **My Matrix** in the name field and press the Enter key.
For this problem, responders gave a median of $13.00, and it costs approximately 68 cents to mail to each person; therefore, the net profit for

- mailing to a responder is $13.00 – 0.68 = 12.32
- mailing to a nonresponder is $0.00 – 0.68 = -0.68

12. Enter the profits associated with the vector for action (LEVEL=1). Your matrix should appear as shown below. You may need to maximize your window to see all of the cells simultaneously. Do not forget to change the bottom right cell of the matrix to 0.

13. To make the newly created matrix active, click on My Matrix to highlight it.

14. Right-click on My Matrix and select Set to use.

15. To examine the decision criteria, select Edit Decisions...
By default, you attempt to maximize profit. Because your costs have already been built into your matrix, do not specify them here. Optionally, you could specify profits of 13 and 0 (rather than 12.32 and -0.68) and then use a fixed cost of 0.68 for Decision=1 and 0 for Decision=0, but that is not done in this example. If the cost is not constant for each person, Enterprise Miner allows you to specify a cost variable. The radio buttons enable you to choose one of three ways to use the matrix or vector that is activated. You can choose to

- maximize profit (default) - use the active matrix on the previous page as a profit matrix, but do not use any information regarding a fixed cost or cost variable.
- maximize profit with costs - use the active matrix on the previous page as a profit matrix in conjunction with the cost information.
- minimize loss - consider the matrix or vector on the previous page as a loss matrix.

16. Close the Editing Decisions and Utilities window without modifying the table.
17. As discussed earlier, the proportions in the population are not represented in the sample. To adjust for this, select the Prior tab.
By default, there are three predefined prior vectors in the Prior tab:

- Equal Probability - contains equal probability prior values for each level of the target.
- Proportional to data - contains prior probabilities that are proportional to the probabilities in the data.
- None - (default) does not apply prior class probabilities.

18. To add a new prior vector, right-click in the open area where the prior profiles are activated and select Add. A new prior profile is added to the list, named Prior vector.

19. To highlight the new prior profile, select Prior vector.

20. Modify the prior vector to represent the true proportions in the population.

21. To make the prior vector the active vector, select Prior vector in the prior profiles list to highlight it.

22. Right-click on Prior vector and select Set to use.

23. Close the target profiler. Select Yes to save changes when prompted.
Investigating Descriptive Statistics

The metadata is used to compute descriptive statistics for every variable.

1. Select the **Interval Variables** tab.

Investigate the descriptive statistics provided for the interval variables. Inspecting the minimum and maximum values indicates no unusual values (such as AGE=0 or TARGET_D<0). AGE has a high percentage of missing values (26%). TIMELAG has a somewhat smaller percentage (9%).

2. Select the **Class Variables** tab.

Investigate the number of levels, percentage of missing values, and the sort order of each variable. Observe that the sort order for TARGET_B is descending whereas the sort order for all the others is ascending. This occurs because you have a binary target event. It is common to code a binary target with a 1 when the event occurs and a 0 otherwise. Sorting in descending order makes the 1 the first level, and this identifies the target event for a binary variable. It is useful to sort other similarly coded binary variables in descending order as well for interpreting results of a regression model.
If the maximum number of distinct values is greater than or equal to 128, the Class Variables tab will indicate 128 values.

Close the Input Data Source node and save the changes when prompted.

**The Data Partition Node**

1. Open the Data Partition node.

2. The right side enables you to specify the percentage of the data to allocate to training, validation, and testing data. Enter 50 for the values of training and validation.

   Observe that when you enter the 50 for training, the total percentage (110) turns red, indicating an inconsistency in the values. The number changes color again when the total percentage is 100. If the total is not 100%, the data partition node will not close.

3. Close the Data Partition node. Select Yes to save changes when prompted.

**Preliminary Investigation**

1. Add an Insight node to the workspace and connect it to the Data Partition node as illustrated below.

2. To run the flow from the Insight node, right-click on the node and select Run.
3. Select **Yes** when prompted to see the results. A portion of the output is shown below.

Observe that the upper-left corner has the numbers 2000 and 21, which indicate there are 2000 rows (observations) and 21 columns (variables). This represents a sample from either the training data set or the validation data set, but how would you know which one?

1. Close the Insight data set to return to the workspace.

2. To open the Insight node, right-click on the node in the workspace and select **Open...**. The Data tab is initially active and is displayed below.

Observe that the selected data set is the training data set. The name of the data set is composed of key letters (in this case, TRN) and some random alphanumeric characters (in this case, 00DG0). The TRN00DG0 data set is stored in the EMDATA library. The bottom of the tab indicates that Insight, by default, is generating a random sample of 2000 observations from the training data based on the random seed 12345.
3. To change which data set Insight is using, choose **Select...**

You can see the predecessor nodes listed in a table. The Data Partition node is the only predecessor.

4. Click on the **next to Data Partition and then click on the **next to SAS_DATA_SETS. Two data sets are shown that represent the training and validation data sets.

5. Leave the training data set as the selected data and select **OK** to return to the Data tab.

6. Select **Properties...** The Information tab is active. This tab provides information about when the data set was constructed as well as the number of rows and columns.
7. Select the **Table View** tab.

This tab enables you to view the data for the currently selected data set in tabular form. The check box enables you to see the column headings using the variable labels. Unchecking the box would cause the table to use the SAS variable names for column headings. If no label is associated with the variable, the column heading cell displays the SAS variable name.

8. Close the Data set details window when you are finished to return to the main Insight dialog.

9. Select the radio button next to **Entire data set** to run Insight using the entire data set.

You can run Insight with the new settings by proceeding as follows:

1. Close the Insight Settings window and select **Yes** when prompted to save changes.

2. Run the diagram from the Insight node.

3. Select **Yes** when prompted to see the results.
You can also run Insight without closing the main dialog by selecting the run icon ( ) from the toolbar and selecting Yes when prompted to see the results.

Use Insight to look at the distribution of each of the variables:

1. Select Analyze ➔ Distribution (Y).
2. Highlight all of the variables except IDCODE in the variable list (IDCODE is the last variable in the list).
3. Select Y.
4. Select IDCODE ➔ Label.
5. Select OK.

Charts for numeric variables include histograms, box and whisker plots, and assorted descriptive statistics.

The distribution of AGE is not overly skewed, so no transformation seems necessary.
Charts for character variables include mosaic plots and histograms.

The variable HOMEOWNR has the value H when the person is a homeowner and a value of U when the ownership status is unknown. The bar at the far left represents a missing value for HOMEOWNR. These missing values indicate that the value for HOMEOWNR is unknown, so recoding these missing values into the level U would remove the redundancy in this set of categories. You do this later in the Replacement node.

Some general comments about other distributions appear below.

- INCOME is treated like a continuous variable because it is a numeric variable.
- There are more females than males in the training data set, and the observations with missing values for GENDER should be recoded to M or F for regression and neural network models. Alternatively, the missing values could be recoded to U for unknown.
- The variable MALEMILI is a numeric variable, but the information may be better represented if the values are binned into a new variable.
- The variable MALEVET does not seem to need a transformation, but there is a spike in the graph near MALEVET=0.
- The variables LOCALGOV, STATEGOV, and FEDGOV are skewed to the right, so they may benefit from a log transformation.
- The variables PETS and PCOWNERS only contain the values Y and missing. Recoding the missing values to U for unknown would make these variable more useful for regression and neural network models.
• The distributions of CARDPROM and NUMPROM do not need any transformation.
• The variables CARDGIFT and TIMELAG may benefit from a log transformation.
• The variable AVGGIFT may yield better results if its values are binned.

You can use Insight to see how responders are distributed.

1. Scroll to the distribution of TARGET_B.
2. Select the bar corresponding to TARGET_B=1
3. Scroll to the other distributions and inspect the highlighting pattern.

Examples of the highlighting pattern for TIMELAG and PCOWNERS are shown. These graphs do not show any clear relationships.
When you are finished, return to the main process flow diagram by closing the Insight windows.

**Understanding Data Replacement**

1. Add a Replacement node to the diagram. Your new diagram should appear as follows:
2. Open the Replacement node.

3. The Defaults tab is displayed first. Check the box for **Create imputed indicator variables** and use the arrow to change the Role field to **input**.

This requests the creation of new variables, each having a prefix `M_` followed by the original variable name. These new variables have a value of 1 when an observation has a missing value for the associated variable and 0 otherwise. If the “missingness” of a variable is related to the response variable, the regression and the neural network model can use these newly created indicator variables to identify observations that had missing values originally.

The Replacement node allows you to replace certain values before imputing. Perhaps a data set has coded all missing values as 999. In this situation, select the Replace before imputation check box and then have the value replaced before imputing.

When the class variables in the score data set contain values that are not in the training data set, these unknown values can be imputed by the most frequent values or missing values. To do this, select the Replace unknown level with check box and then use the drop-down list to choose either most frequent value (count) or missing value.

**Using Data Replacement**

1. Select the **Data** tab. Most nodes have a Data tab that enables you to see the names of the data sets being processed as well as a view of the data in each one. The radio button next to Training is selected.
2. Select the **Training** subtab under the Data tab.

By default, the imputation is based on a random sample of the training data. The seed is used to initialize the randomization process. Generating a new seed creates a different sample.

3. To use the entire training data set, select the button next to **Entire data set**. The subtab information now appears as pictured below.

4. Return to the **Defaults** tab and select the **Imputation Methods** subtab.

This shows that the default imputation method for Interval Variables is the mean (of the random sample from the training data set or the entire training data set, depending on the settings in the Data tab). By default, imputation for class variables is done
using the most frequently occurring level (or mode) in the same sample. If the most commonly occurring value is missing, it uses the second most frequently occurring level in the sample.

Click on the arrow next to the method for interval variables. Enterprise Miner provides the following methods for imputing missing values for interval variables:

- Mean – uses the arithmetic average. This is the default.
- Median – uses the 50th percentile.
- Midrange – uses the maximum plus the minimum divided by two.
- Distribution-based – calculates replacement values based on the random percentiles of the variable’s distribution.
- Tree imputation – estimates replacement values with a decision tree using the remaining input and rejected variables that have a status of use as the predictors.
- Tree imputation with surrogates – is the same as above except that surrogate variables are used for splitting whenever a split variable has a missing values. This prevents forcing everyone with a missing value for a variable into the same node.
- Mid-min spacing – uses the mid-minimum spacing statistic. To calculate this statistic, the data is trimmed using $N$ percent of the data as specified in the Proportion for mid-minimum spacing entry field. By default, 90% of the data is used to trim the original data. In other words, 5% of the data is dropped from each end of the distribution. The mid-range is calculated from this trimmed data.
- Tukey’s biweight, Huber’s, and Andrew’s wave – are robust M-estimators of location. This class of estimators minimize functions of the deviations of the observations from the estimate that are more general than the sum of squared deviations or the sum of absolute deviations. M-estimators generalize the idea of the maximum-likelihood estimator of the location parameter in a specified distribution.
- Default constant – enables you to set a default value to be imputed for some or all variables.
- None – turns off the imputation for interval variables.

Click on the arrow next to the method for class variables. Enterprise Miner provides several of the same methods for imputing missing values for class variables including distribution-based, tree imputation, tree imputation with surrogates, default constant, and none.
5. Select **Tree imputation** as the imputation method for both types of variables.

When using the tree imputation for imputing missing values, use the entire training data set for more consistent results.

Regardless of the values set in this section, you can select any imputation method for any variable. This tab merely controls the default settings.

6. Select the **Constant values** subtab. This subtab enables you to replace certain values (before imputing, if desired, using the check box on the Defaults tab). It also enables you to specify constants for imputing missing values.

7. Enter U in the field for character variables.
8. Select the **Tree Imputation** tab. This tab enables you to set the variables that will be used when using tree imputation. Observe that target variables are not available, and rejected variables are not used by default. To use a rejected variable, you can set the Status to **use**, but that would be inappropriate here because the rejected variable TARGET_D is related to the target variable TARGET_B.

![Replacement](image)

Suppose you want to change the imputation method for AGE to mean and CARDPROM to 20.

1. Select the **Interval Variables** tab.

2. To specify the imputation method for AGE, position the tip of your cursor on the row for AGE in the Imputation Method column and right-click.

3. Select **Select Method…** ⇒ **mean**.

4. To specify the imputation method for CARDPROM, position the tip of your cursor on the row for CARDPROM in the Imputation Method column and right-click.

5. Select **Select Method…** ⇒ **set value**.

6. Type **20** for the new value.

7. Select **OK**.

8. Specify **none** as the imputation method for TARGET_D in like manner.
Inspect the resulting window. A portion of the window appears below.

Recall that the variables HOMEOWNR, PCOWNERS, and PETS should have the missing values set to \( U \).

1. Select the **Class Variables** tab.
2. Control-click to select the rows for **HOMEOWNR**, **PCOWNERS**, and **PETS**.
3. Right-click on one of the selected rows in the Imputation Method column.
4. Select **Select Method... ⇒ default constant**.
5. To change the imputation for TARGET_B to none, right-click on the row for TARGET_B in the Imputation Method column.
6. Choose **Select method... ⇒ none**.
7. Select the **Output** tab. While the Data tab shows the input data, the Output tab shows the output data set information.

![Replacement node](image)

8. Close the Replacement node saving the changes when prompted.

**Performing Variable Transformations**

Some input variables have highly skewed distributions. In highly skewed distributions, a small percentage of the points may have a great deal of influence. On occasion, performing a transformation on an input variable may yield a better fitting model. This section demonstrates how to perform some common transformations.

Add a Transform Variables node to the flow as shown below.

![Transform Variables node](image)

Open the Transform Variables node by right-clicking on it and selecting **Open...**. The Variables tab is shown by default. It displays statistics for the interval-level variables including the mean, standard deviation, skewness, and kurtosis (calculated from the metadata sample). The Transform Variables node enables you to rapidly transform
interval-valued variables using standard transformations. You can also create new variables whose values are calculated from existing variables in the data set. Observe that the only non-greyed column in this dialog is the Keep column.

You can view the distribution of each of the variables just as you did in the Input Data Source node. Begin by viewing the distribution of AGE. The distribution of AGE is not highly skewed, so no transformation is performed. Close the distribution of AGE.
Investigate the distribution of AVGGIFT.

This variable has the majority of its observations near zero, and very few observations appear to be higher than 30. Consider creating a new grouping variable that creates bins for the values of AVGGIFT. You can create just such a grouping variable in several different ways.

- Bucket - creates cutoffs at approximately equally spaced intervals.
- Quantile - creates bins with approximately equal frequencies.
- Optimal Binning for Relationship to Target - creates cutoffs that yield optimal relationship to target (for binary targets).

The Optimal Binning for Relationship to Target transformation uses the DMSPLIT procedure to optimally split a variable into \( n \) groups with regard to a binary target. This binning transformation is useful when there is a nonlinear relationship between the input variable and the binary target. An ordinal measurement level is assigned to the transformed variable.

To create the \( n \) optimal groups, the node applies a recursive process of splitting the variable into groups that maximize the association with the target values. To determine the optimum groups and to speed processing, the node uses the metadata as input.

Close the distribution of AVGGIFT.

Create bins for AVGGIFT. Suppose your earlier analysis suggested binning the values into the intervals 0-10, 10-20, and 20+.

1. Right-click on the row for AVGGIFT and select Transform... ⇒ Bucket.
2. The default number of buckets is 4. Change this value to 3 using the arrows.
3. Select Close.
4. Enter 10 in the Value field for Bin 1. Press the Enter key.

5. Use the ▼ to change from Bin 1 to Bin 2.

6. Enter 20 in the Value field for Bin 2 and press the Enter key. The result appears as pictured below:

7. Close the plot and select Yes to save the changes when prompted.
A new variable is added to the table. The new variable has the truncated name of the original variable followed by a random string of digits. Note that the Enterprise Miner set the value of Keep to No for the original variable. If you wanted to use both the binned variable and the original variable in the analysis, you would need to modify this attribute for AVGGIFT and set the value of Keep to Yes, but that is not done here.

<table>
<thead>
<tr>
<th>Name</th>
<th>Keep</th>
<th>Role</th>
<th>Formula</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGE</td>
<td>Yes</td>
<td>input</td>
<td>62.112945839</td>
<td></td>
</tr>
<tr>
<td>AVGGIFT</td>
<td>No</td>
<td>input</td>
<td>12.81512315</td>
<td></td>
</tr>
<tr>
<td>AVGG_NDA</td>
<td>Yes</td>
<td>input</td>
<td>AVGGIFT</td>
<td>12.81512315</td>
</tr>
<tr>
<td>CARDGIFT</td>
<td>Yes</td>
<td>input</td>
<td>5.3205</td>
<td></td>
</tr>
</tbody>
</table>

Examine the distribution of the new variable.

The View Info tool reveals that there is over 40% of the data in each of the two lowest categories and there is approximately 10% of the data in the highest category.

Recall that the distributions of LOCALGOV, STATEGOV, FEDGOV, CARDGIFT, and TIMELAG were highly skewed to the right. A log transformation of these variables may provide more stable results.

Begin by transforming CARDGIFT.

1. Position the tip of the cursor on the row for CARDGIFT and right-click.
2. Select Transform… ⇒ log.

Inspect the resulting table.

<table>
<thead>
<tr>
<th>Name</th>
<th>Keep</th>
<th>Role</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGE</td>
<td>Yes</td>
<td>input</td>
<td></td>
</tr>
<tr>
<td>AVGGIFT</td>
<td>No</td>
<td>input</td>
<td></td>
</tr>
<tr>
<td>AVGG_NDA</td>
<td>Yes</td>
<td>input</td>
<td>AVGGIFT</td>
</tr>
<tr>
<td>CARDGIFT</td>
<td>No</td>
<td>input</td>
<td></td>
</tr>
<tr>
<td>CARD_8EF</td>
<td>Yes</td>
<td>input</td>
<td>log((CARDGIFT + 1))</td>
</tr>
</tbody>
</table>

The formula shows that Enterprise Miner has performed the log transformation after adding 1 to the value of CARDGIFT. Why has this occurred? Recall that CARDGIFT has a minimum value of zero. The logarithm of zero is undefined, and the logarithm of something close to zero is extremely negative. The Enterprise Miner takes this
information into account and actually uses the transformation log(CARDGIFT+1) to create a new variable with values greater than or equal to zero (because the log(1)=0).

Inspect the distribution of the transformed variable. It is much less skewed than before.

Perform log transformations on the other variables (FEDGOV, LOCALGOV, STATEGOV, and TIMELAG).

1. Press and hold the Ctrl key on the keyboard.
2. While holding the Ctrl key, select each of the variables.
3. When all have been selected, release the Ctrl key.
4. Right-click on one of the selected rows and select Transform... → log.
5. View the distributions of these newly created variables.

It may be appropriate at times to keep the original variable and the created variable although it is not done here. It is also not commonly done when the original variable and the transformed variable have the same measurement level.

Close the node when you are finished, saving changes when prompted.
Fitting a Regression Model

1. Connect a Regression node to the diagram as shown.

2. Open the Regression node.

3. Find the Tools menu on the top of the session window and select Tools ⇒ Interaction Builder… This tool enables you to easily add interactions and higher-order terms to the model, although you do not do so now.

The input variables are shown on the left, and the terms in the model are shown on the right. The Regression node fits a model containing all main effects by default.
4. Select **Cancel** to close the Interaction Builder window when you are finished inspecting it.

5. Select the **Selection Method** tab. This tab enables you to perform different types of variable selection using various criteria. You can choose backward, forward, or stepwise selection. The default in Enterprise Miner is to construct a model with all input variables that have a status of use.

6. Select **Stepwise** using the arrow next to the Method field.

If you choose the **Forward**, **Backward**, or **Stepwise** effect selection method, then you can use the Criteria field to specify a selection criterion to be used to select the final model. The node first performs the effect selection process, which generates a set of candidate models corresponding to each step in the process. Effect selection is done based on the **Entry** or **Stay Significance Levels** found in the Criteria subtab of the Model Selection tab. Once the effect selection process terminates, the candidate model that optimizes the selection criterion on the validation data set is chosen as the final model.

Inspect the Effect Hierarchy options in the lower-left corner of the window. Model hierarchy refers to the requirement that for any effect in the model, all effects that it contains must also be in the model. For example, in order for the interaction A*B to be in the model, the main effects A and B must also be in the model. The Effect Hierarchy options enable you to control how a set of effects is entered into or removed from the model during the variable selection process.
7. Select the **Criteria** subtab.

![Image of Model Ordering window]

The list of candidate effects can be seen by selecting from the main menu **Tools ⇒ Model Ordering**... This opens the Model Ordering window.

The Start value, \( n \), selects the first \( n \) effects from the beginning of this list as the first model. For the Forward and Stepwise selection methods, the default Start value is 0. For the Backward selection method, the default is the total number of candidate effects.

The Force value, \( n \), is used to specify the effects that must be included in the final model regardless of the selection method. The first \( n \) variables in the Model Ordering list will be included in every model.

The Stop value for the Forward method is the maximum number of effects to appear in the final model. For the Backward method, the Stop value is the minimum number
of effects to appear in the final model. For the Backward selection method, the default Stop value is 0. For the Forward selection method, the default is the total number of input variables. The Stop option is not used in the Stepwise selection method.

The Stepwise stopping criteria field enables you to set the maximum number of steps before the Stepwise method stops. The default is set to twice the number of effects in the model.

The Stepwise method uses cutoffs for the variables entering the model and for the variables leaving the model. The default significance levels are 0.05 for both entry and stay.

Changing these values may impact the final variables included in the model.

8. Close the Regression node saving the changes when prompted.

9. Because you have changed the default settings for the node, it prompts you to change the default model name. Enter **StepReg** for the model name.

![](image)

10. Select **OK**.

11. Run the diagram from the regression node.

12. Select **Yes** to view the results when prompted.

The results open with the Estimates tab active and provide a graph of the effect T-scores.

![](image)

The graph shows the relative significance of the model parameter estimates. There are several ways to determine what variables the bars represent.
1. You can expand the legend. Select the Move and Resize Legend tool, point at the dark part of the legend’s top border until your cursor becomes a double-headed arrow. Drag the legend border up to make the legend area bigger.

<table>
<thead>
<tr>
<th>Effect Label</th>
<th>Effect T-scores</th>
<th>Parameter Estimate</th>
<th>T-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1 \log(CARDGIFT)$</td>
<td>0.5923138353</td>
<td>6.9532138353</td>
<td></td>
</tr>
<tr>
<td>$X_2 \text{LASTT}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_3 \text{CARDPROM}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$X_4 \text{intercept}$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. As an alternative to making the legend bigger, you can use the View Info tool, Select the View Info tool and point and click on any of the bars in the graph. Information about that bar will be displayed.

3. Select the Table radio button in the Estimates tab. Parameter estimates and effect T-scores will only be shown for variables in the selected model.

Regardless of the method chosen to determine the variables in the model and their relative importance, for the model to predict TARGET_B, the important variables are:

- $\log(CARDGIFT)$ – the natural log of the donor’s gifts to previous card promotions
- \text{LASTT} – the elapsed time since the last donation
- \text{CARDPROM} – the number of card promotions previously received.
4. Select the **Statistics** tab.

![Results - Regression](image)

The Statistics tab lists fit statistics, in alphabetical order, for the training data, validation data, and test data analyzed with the regression model. In this example, you only have training and validation data sets.

5. Close the regression results.

**Fitting a Default Decision Tree**

1. Add a default Tree node to the workspace. Connect the Data Partition to the Tree.

2. Add an Assessment node to the workspace and then connect the Tree node and the Regression node to the Assessment node. The flow should now appear like the one pictured below.

![Decision Tree Flow](image)
A decision tree handles missing values directly, so it does not need data replacement. Monotonic transformations of interval variables will probably not improve the tree fit because the tree bins numeric variables. The tree may perform worse if you connect it after binning a variable in the Transform Variables node, because binning reduces the splits the tree can consider (unless you include the original variable and the binned variable in the model).

3. Run the flow from the Assessment node and select **Yes** when you are prompted to view the results. The Assessment node opens with two models displayed.

You can change the name for the model by editing the Name column. This feature is especially useful when you fit several competing models of the same type.

4. Enter the name **DefTree** in the Name column for the Tree tool to indicate that you have fit a default tree.

5. To generate a lift chart, highlight both rows in the Assessment node. You can do this by selecting the row for one model and then Ctrl-clicking on the row for the other model. You can also drag through both rows to highlight them simultaneously.

6. Select **Tools ⇧ Lift Chart** to compare how the models perform on the validation data set.
Observe that the regression model outperforms the default tree throughout the graph.

7. Close the Assessment node when you have finished inspecting the various lift charts.