

Modeling Credit Risks: A Introduction to Data Mining using SAS Enterprise Miner¹

Note:

In this exercise you'll be using SAS Enterprise Miner software. SAS Enterprise Miner is only available in the Computer Lab in O'Neil. You will not be using Visual Interdev for this assignment. Learning data mining can be challenging. Enterprise Miner is a complex software package that requires significant hands-on experience to master. *It's important to note that most organizations employ teams with three distinct skill sets on data mining projects – IS, marketing/business knowledge, and statistics expertise.* This isn't a statistics class, so this exercise is not meant to provide a formal stats instruction. Statistics jargon is avoided where possible and concepts are deliberately simplified. In some areas, however, we just can't avoid certain terms. Unlike other exercises where you gained & applied a skill (e.g. developing a website, completing database queries), this exercise is designed simply to be a 'test drive' so you can get an idea how one popular software package, SAS Enterprise Miner, can be used to create models that can solve real-world business problems. I do expect, however, that you retain knowledge regarding issues such as the conditions under which various models can be appropriately applied. This is an important experience since business unit personnel are intimately involved with the data mining process. Your experience with this tool sets you apart from folks in other programs, most of whom can't 'talk the talk', let alone 'walk the walk' with this technology. This exercise may require several hours to complete. Those who want to take a break & return to their work later can save work via myFiles for later access, however it is **STRONGLY** advised that you try to complete the assignment in one sitting in case you have problems re-loading your saved work. An estimate of completion time is 3 hours, realizing that people work at different paces. It is recommended that you read this document prior to arriving at the lab so you can get an idea of the steps that you'll take. While the software is very powerful, at times it may not seem user-friendly. **Start early** and feel free to ask your instructor or fellow classmates for assistance if you're stuck. Be patient – you are welcome to re-read this text or run the exercise over again. Performing this exercise several times may clarify concepts beyond simply repeating the commands to get Enterprise Miner to run. Good luck!

Further Information

Students interested in working beyond this exercise to learn more about the SAS Enterprise Miner data mining software might want to purchase the book "Getting Started with Enterprise Miner(TM) Version 4.0". It's available for less than \$20 from most large Internet bookstores. The software and data sets for this exercise will remain in the O'Neil computer lab, so you are welcome to do extra learning on your own. SAS likely has additional titles related to data mining as well.

A little information on SAS.

¹ © Copyright John M. Gallagher, Ph.D, Last Updated November 28, 2005. This lesson has been adapted from the book *Getting Started with Enterprise Miner Version 4.0* from the SAS Institute.

The SAS Corporation is located in Cary, NC (a geek hot-spot of the Southeastern U.S.). Enterprise Miner is a \$90,000+ package that is an add-on to the firm's flagship SAS statistics package. Enterprise Miner is the most popular data mining tool on the market. SAS has been listed as the largest privately held software company in the United States and regularly ranks among the best U.S. firms to work for.





Introduction

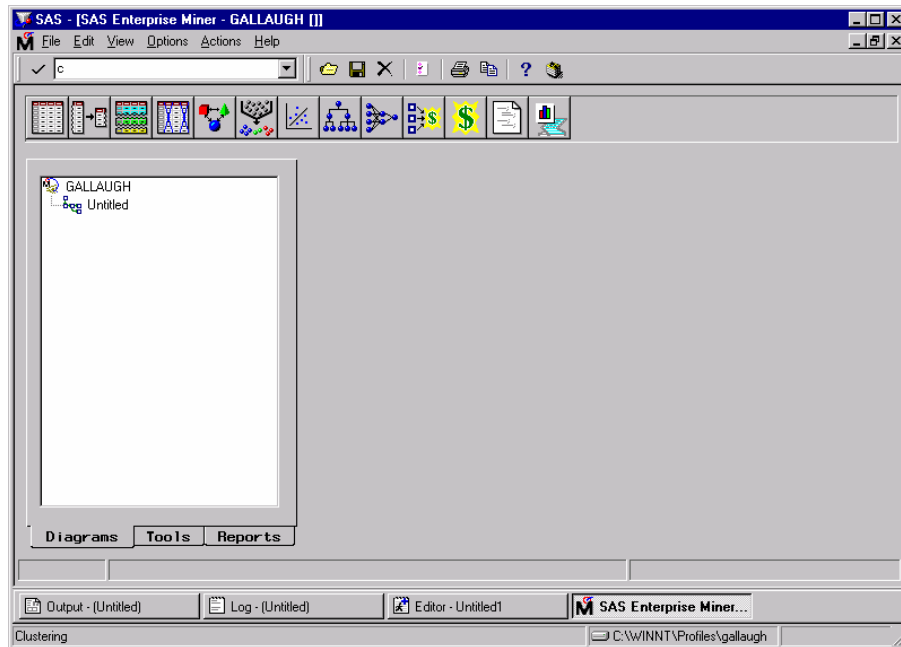
You've got a problem – you don't reliably know if a given customer will pay back or default on the loans that your firm grants them. On average, one of every nine loans you grant is defaulted on, but you're not sure how to tell the potentially good applicants from bad ones. You've got historical data on customer information prior to being granted a loan, as well as information on whether these past loans were good (paid off) or bad (defaulted on). You need to build a predictive model to help you make better loan approval decisions and SAS Enterprise Miner is here to help!

Launch SAS and Start Enterprise Miner

- Log on to a computer in the O'Neill Computer Lab and start SAS. SAS is launched by selecting: Start -> All Programs -> Applications -> SAS -> SAS 9.1.(English)
 - If you are asked if you want to view the 'Start Guides', click the 'don't show this dialog box again' check box and click 'Close'.
- Using the 'Solutions' menu in SAS, select 'Analysis' -> 'Enterprise Miner'. SAS Enterprise Miner will initialize and the SAS Enterprise Miner window will appear.
 - If you are asked if you want to start the tutorial, click the 'don't show this dialog box again' check box and click 'Close'.

Before we start, let's perform a bit of housekeeping to maximize your working space.

- If the main SAS window (the larger window behind the one labeled SAS Enterprise Miner) is not maximized so that it is taking up the entire screen, click the maximize button  in the upper right-hand corner of this window. The SAS window should now occupy the entire screen.
- If the main SAS window shows a long, thin window to the left labeled 'Explorer', close this window by clicking the close box  of the window labeled 'Explorer'.
- If the main SAS window shows a long, thin window to the left labeled 'Results', close this window by clicking the close box  of the window labeled 'Results'.
- Maximize the SAS Enterprise Miner window by clicking the maximize button  in the upper right-hand corner of this window. Your screen should look like the one below.



Create a new local project

- Use the File menu to select New -> Project.
- Type a name for your project 'E-Miner Project' would be a good one.
 - Note: Your project is being saved to the "D:\:" temp drive & will be deleted after you shut down from this PC. If you want to backup your work, you can upload it to myFiles or save it to a disk/CD. Also note: to return to SAS and access any saved files after you've shut down, drag your saved files back to the temp folder, launch SAS and start Enterprise Miner as described in the previous section, then select "File -> Open", navigate to the "D" drive where you saved your work, and select your saved file (which will be named 'Credit Risk Diagram.dmd' if you follow the steps below).
- Click the 'Create' button. A project will be created and you will see the project name in the upper left-hand corner in the Diagrams tab of a window that we will refer to as the Project Navigator. A default diagram labeled 'Untitled' is created underneath it.

Rename the Untitled diagram

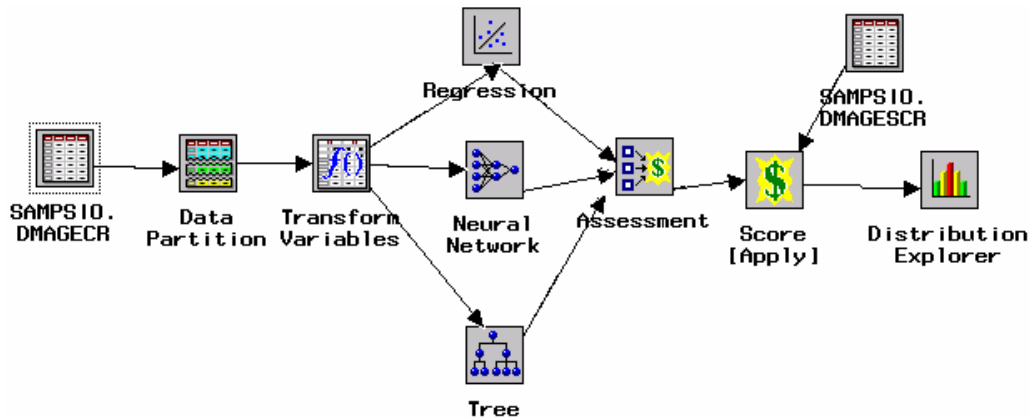
- Right click on the diagram icon to the left of the word 'Untitled' and select 'Rename'.
- Type the new name as 'Credit Risk Diagram' and press enter.

We're ready to begin!

Data Mining Diagrams

SAS is a very robust but complicated statistics package. Those of you who have done advanced research work may be familiar with this tool. If so, you probably know that using SAS typically requires a lot of programming. Fortunately, the Enterprise Miner tool sits 'on top' of SAS and allows us to set up a data mining session using relatively easy-to-use dialog boxes rather than coding. We'll be building the steps necessary to

perform our data mining exercise by grabbing what are called 'nodes', small icons that contain dialogs that let us set up steps in the data mining process. The screen shot below shows how our node diagram will look when we have completed this exercise.



Each node has a specific function. Read the diagram from left to right as follows. *SAMP SIO . DMAGECR* is a data input source that reads in a file with that name. This file contains historical loan data from Germany with information on customer background and whether the loan was good (it was paid off) or bad (it was in default). This past data will be used to create models to help us better evaluate new customers to determine who is an appropriate credit risk.

Data Partition is used to partition our data into a training set for building predictive models and a validation set to further refine and examine the accuracy of these models. We do this in part to make sure that results of the trained and validated models are similar and aren't the result of a quirk in the sample used to build the models.

Transform variables will perform transformations on the data, to 'get it into shape' so that it's ready to be used by the model building methods we're going to employ.

Regression, *Neural Network*, and *Tree* are the three statistical techniques that we will be using to build competing models – *Regression* will be used to create a linear logistical regression model, *Neural Network* will create a neural network model, and *Tree* will create a Decision Tree model. We don't know which model will be the best at predicting good vs. bad loans, so we're going to try out these three techniques and compare the results to choose a best model.

Assessment will allow us to compare the three models so that we can try to determine which one is best for our use (i.e. which one does the best job at predicting good vs. bad loans).

Score is the end-game in our data mining exercise. It allows us to take the predictive model that we built from historical data and apply it to a new data set in order to 'score' the data. In our case that means to apply the model to each record in the new file in order

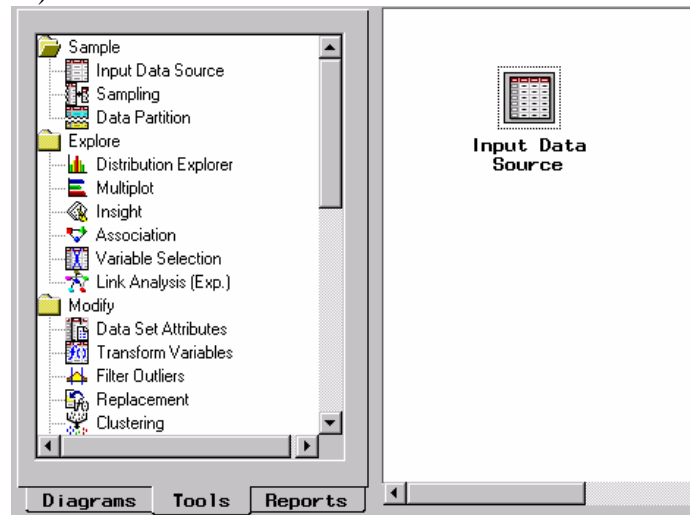
to predict if each record will yield a good or bad loan. This data will be read in from the input data node labeled 'SAMPSIO.DMAGESCR'.

Distribution Explorer will allow us to explore the distribution of predicted good vs. bad loans in the file we scored.

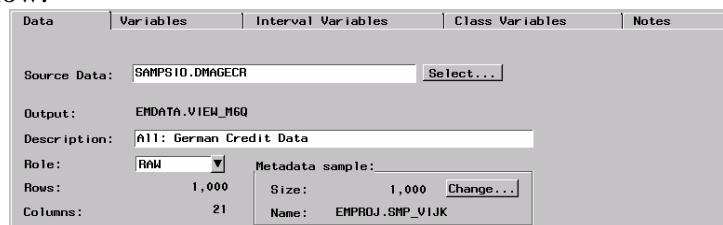
Defining an Input Data Set

Add an Input Data Source Icon.

- Click the 'Tools' tab at the bottom of the Project Navigator to display the selection of nodes that we can use.
- Under the 'Sample' folder in the Tools palette, locate the icon labeled 'Input Data Source'. Drag and drop 'Input Data Source' into the Diagram Workspace (the white space on the left).



- Open the Input Data Source node icon that you just dropped in your diagram workspace by double-clicking on the icon. The input data source window will appear.
- Click the 'Select' button to the right of the Source Data field.
- Click the down arrow next to the Library field in this dialog box & select SAMPSIO from the drop down. A list of available tables will be displayed.
- In the "Tables" list, click to select the table DMAGECR (All: German Credit Data).
- Click "OK" at the bottom of the page to continue.
- After a second or two, the output and description information should be filled in as depicted below.



You are using a data set containing actual German Credit Data (so we'll assume this is data from your bank and that you're working in Germany). There are 1,000 observations in the sample. In practice, many firms mine data sets that contain many thousand records – you're working with a relatively small sample here in order to minimize processing time.

NOTE: Throughout the exercise, you may use the “What’s This?” question mark tool to get additional information regarding the specifics of the screen you are on. While in some programs this tool is relatively useless, SAS has a fairly robust help screen that can provide additional insight into options the software provides.

Setting the Target Variable and Target Profile

We first need to specify the target or dependent variable that we're trying to model. In this case, the target is the variable GOOD_BAD. This variable contains an indicator as to whether the loan was a good one or a bad one. We'll be using the historical data to build a model based on past results of this GOOD_BAD variable. Once the model is built, we'll apply it to new data where we know the input variables (e.g. information on the customer), but don't know the likely outcome of granting the customer a loan (i.e. are they likely to pay it off or default?). We set the target as follows:

- Select the Variables tab at the top of the window. You'll see many variables listed that describe various attributes of past loan consumers (for example, whether or not they were employed; whether they were married, etc.).
- Scroll (if you need to) to the bottom of the variable list so that you can see the variable GOOD_BAD (it should be the last variable in the list).
- Locate the 'Model Role' cell to the right of the GOOD_BAD variable, right-click in the cell, and select 'Set Model Role' from the pop-up menu. Another pop-up menu opens.
- Select 'target'. You should see the 'Model Role' cell for the GOOD_BAD variable change from input to target. To relate this to concepts in statistics – we've just set the GOOD_BAD variable to be our dependent variable – that is the variable that is 'calculated' by the result of the equations we develop in our models.

Now let's set up the Target Profile. The Target Profile will allow us to perform a profit calculation to determine the economic impact of four possible outcomes depicted in the grid below.

Decisions		
Target Values	Accept	Reject
Good	-\$1	\$0
Bad	\$5	0

In our example, we'll assume that if we reject a loan (good or bad) there is no loss but no profit from that loan. But if we accept a bad loan, the economic impact is five times worse than accepting a good loan. The values in the table above reflect the loss related to

a decision. Therefore, accepting a good loan gives a 'loss' of negative one dollar, meaning a negative loss, which is a double-negative or a one dollar profit, as opposed to a five to one loss for accepting a bad loan. Sorry that this notation is a bit confusing, but statistics packages aren't known for being particularly user-friendly.

There are a bunch of steps we need to go through to tell SAS that this is the set of profit/loss calculations that the model needs to consider. First, we need to tell Enterprise Miner to consider this target profile when evaluating our models.

- Still within the 'Variables' tab, open the Target Profiler by right-clicking in any cell of the GOOD_BAD target variable row, and select 'Edit target profile...'.
– You may see a dialog box stating "No target profile found. Processing time to create a new target profile could be excessive. Do you want to continue?". If you see this dialog box, click 'Yes'. The 'Target Profiles for GOOD_BAD' window opens.
– Click the 'Assessment information' tab.

Create a copy of the Default Loss matrix

- Click the 'Default Loss' list item to highlight it.
– Right-click the item labeled 'Default Loss', then select 'Copy'. You should see a new item at the bottom of the list labeled 'Profit Matrix'.


Set this new Profit Matrix so that Enterprise Miner uses it when evaluating models. Do this as follows:


- Click the 'Profit Matrix' entry to highlight this item.
– Right-click on the entry 'Profit Matrix' and select the 'Set to use' menu item. An asterisk appears besides the matrix name indicating that it is now the active decision matrix that will be used by SAS.

Let's rename the 'Profit Matrix' to something more appropriate like 'Realistic Loss'.

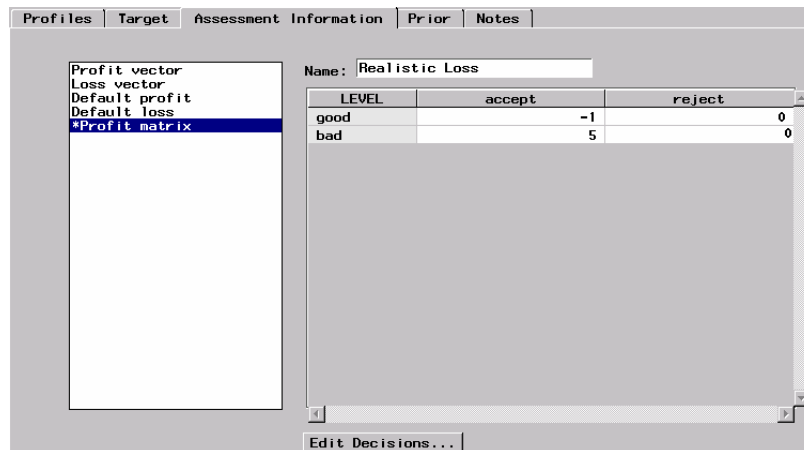
- In the name field near the top of this window, backspace over the phrase 'Profit Matrix' and type in 'Realistic Loss'.

Let's rename the decision columns to 'accept' and 'reject' instead of 'good' and 'bad'.

- Click the 'Edit Decisions...' button at the bottom of the window. The 'Editing Decisions and Utilities' window will appear.
– Type 'accept' in place of the decision named 'good'.
– Type 'reject' in place of the decision named 'bad'.
– Close the 'Editing Decisions and Utilities' window by clicking the close window icon  (the second one in the upper right-hand corner of the screen. It is on the same gray line as the menu items). When prompted to save changes, select 'Yes'. This will return you to the 'Target Profiles for GOOD_BAD' window.

Note: unless specified otherwise, whenever asked to close a window in this exercise, you should always click the second (not the top) close window icon  in the top left-hand corner of your screen. Clicking the top close window icon will close your SAS session and you don't want to do this.

- Type the values into the loss matrix to reflect the five to one target profile table we showed earlier. The window should look like the one below:

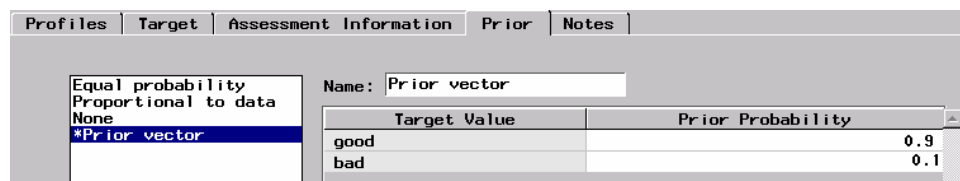


We also need to tell SAS our best guess as to the probabilities of good vs. bad loans being received in the data set that we're going to score after creating the models. This will be used in part to provide a baseline to compare having no prediction to any new models we come up with. Do this as follows:

- Click the 'Prior' tab.
- Add a new prior vector by right-clicking in an open white-space area of the list box and selecting the 'Add' pop-up menu item. A new Prior vector is added.

Set the Prior vector that you just added so that it is the one that SAS will use:

- Click to highlight the line labeled 'Prior vector' that you just created.
- Right-click this item and select 'Set to use'. An asterisk will appear to the left of this item indicating that it is set to be used by SAS.
- Type .9 (that's point nine) into the Prior Probability cell for 'good'.
- Type .1 (that's point one) into the Prior Probability cell for 'bad'. Your window should look like the one below.



- Close the window by clicking the second (not the top) close window icon in the top right-hand corner of your screen, and follow the prompts to save your changes to the target profile. You will be returned to the Variables tab of the Input Data Source node.
- Close the Input Data Source window and save all changes when prompted. You will be returned to the screen where you create your node diagram.

Let's summarize what you've just accomplished. You've:

- (1) chosen an input data source with data describing previous loan information;
- (2) selected the variable you ultimately want to model (i.e. GOOD_BAD or whether the loan is a good risk or not);

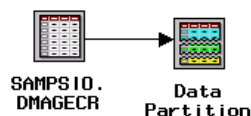
- (3) set the target variable's profile to specify four possible economic outcomes; and
- (4) entered a best guess for the probability of the target being a good or bad loan.

Setting Up the Data Partition

In this data mining example, we'll use a portion of the data (which we'll refer to as the training data) to build or fit the models. We'll hold a portion of the data for later use to validate the data so that SAS can verify and fine-tune the models that it develops. In practice, sometimes folks use a third data set for an additional testing & comparing of models against one another, but since our sample only has 1,000 observations (a relatively small amount when compared to the multi-gigabyte files common in industrial examples), we'll skip this last testing phase. To create the training and validation data sets, follow these steps:

Add a Data Partition node to the Diagram Worksapce.

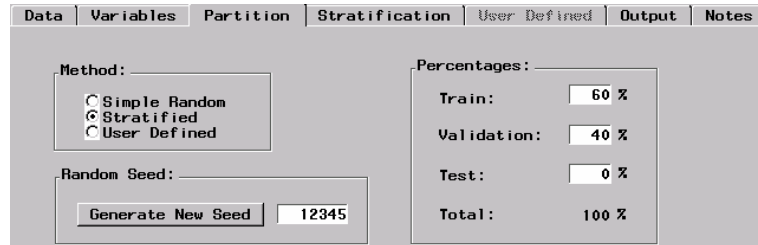
- Drag and drop the line 'Data Partition' from the Tools Palette of the Project Navigator on the left into the Diagram Workspace on the right. You may want to position it to the right of the node you'd previously set up, but don't worry about being overly neat in lining things up. 'Data Partition' is near the top of the list, the third item under 'Sample' (you may have to scroll to find it).
- Connect the Input Data source node (labeled SAMPSIO.DMAGECR) to the Data Partition Node by drawing an arrow between the nodes. This can be tricky. Do this by holding the left mouse button down on the outer-right edge of the Input Data Source node, then dragging your mouse pointer to the Data Partition node. A line will appear as you drag. Release the mouse when you've drawn the line between the two nodes. The line will probably be covered in a 'box', so you won't initially be able to see the arrowhead.
Some students that have had trouble with this step mistakenly selected the SAMPSIO icon first, so that it has a grey box around it. That WON'T work. If you have selected the SAMPSIO icon so that there is a grey box around it, de-select the icon by clicking in the whitespace surrounding the icons. The little grey box should go away & you should now be able to draw your arrows.
- Click in the white space to the side of one of the nodes in order to 'deselect' the arrow that you've just drawn in order to see that it has shown up as pointing from the Input Data Source node to the Data Partition node. Your diagram should look roughly like the one below.



- Double click on the 'Data Partition' icon to open this node. The SAS Data Partition window will open.

Tell SAS to use 60% of the input data source for training (building) models and 40% for validating models.

- Type 60 as your training percentage
- Type 40 as your validation percentage
- Type 0 as your test percentage
- Be sure the random seed is set to 12345. If not, type this number into the Random seed box next to the 'Generate New Seed' button (but do not click this button).
- Select the 'Stratified' radio button from the 'Method' area in the left of this window. Your screen should look like the one below.



By selecting 'Stratified' above, you are telling SAS to preserve the ratio of good and bad loans used in the build and validation phases. This can be important when a particular event is a rare occurrence (you don't want to under or over-consider an event's likelihood when building or testing the model). Now we need to tell SAS to stratify on the GOOD_BAD variable – that is, to create the same ratio of outcomes for this variable when choosing samples for test & validation.

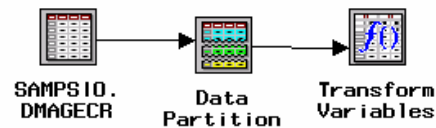
- Select the 'Stratification' tab.
- Right click the 'Status' cell of the GOOD_BAD variable (it should be the last variable in the list) and select the 'Set Status' pop-up menu item. Another pop-up menu appears.
- Select the 'use' pop-up menu item.
- Close the 'Data Partition' node (use the close window icon as mentioned before). Save node settings when prompted. You will be returned to your node diagram.

Creating Variable Transformations

Those of you who have taken statistics realize that sometimes the data that we want to examine may not be in the best form for building a reliable model. If this is the case, we may transform the data, modifying the variables so that they work better with the techniques that we are employing. Transforming variables can be used to stabilize variance, remove nonlinearity, improve additivity, and correct nonnormality. In this example we're going to look at some of the data in our sample. We'll find that one of the variables is skewed, so we'll tell SAS to transform the variable so that its distribution is more 'normal' (i.e. so that the distribution looks more like a bell curve than one that is skewed to one side). We'll also set up ranges for an AGE variable so that we consider age ranges rather than the exact value of an applicant's age.

First, add a 'Transform Variables' node to the Diagram Workspace

- Drag and drop the line 'Transform Variables' from the Tools Palette of the Project Navigator on the left into the Diagram Workspace on the right. You may want to position this node to the right of the nodes that you already set up. 'Transform Variables' is the second sub-item under the 'Modify' heading in the list (you may have to scroll to find it).
- Using the technique described earlier, draw an arrow connecting the Data Partition node to the Transform Variables node. Your diagram should look like the one listed below.



- Open the Transform Variables icon by double-clicking it. A list of variables will be displayed.

Someone familiar with statistics might notice that the Skew figure for the AMOUNT variable looks a bit high. It's around 1.95, whereas if the distribution was more like a bell-curve, this number would be near zero. Let's view the distribution of this variable to see how skewed it looks.

- Right click in any cell of the AMOUNT variable row and select the 'View Distribution of AMOUNT' pop-up menu item. A histogram (bar chart) of AMOUNT is displayed in the Variable Histogram window.

The histogram that appears should show that the right-tail of the distribution is really long compared to the left tail. Let's tell SAS to create a new variable that corrects this skewness.

- Close the window to return to the Variables tab of the Transform Variables window.
- Right-click in any cell of the row that contains the variable AMOUNT and select the 'Transform...' pop-up menu item. Another pop-up menu opens.
- Select the 'Maximize Normality' power transformation to create the transformation and return to the Variables tab of the Transform Variables window. You should notice that SAS has created a line which represents a new variable with a name beginning with AMOU_XXX (where XXX are three random characters).

A couple of things happened here – first, the formula is listed as $\text{Log}(\text{AMOUNT})$, stating that SAS has decided that taking the Log of AMOUNT would be a good way to reduce skewness. Those of you who have taken statistics will recall that a log transformation can make skewed data appear more like a bell-curve distribution, which is necessary for certain techniques. Also, by looking at the Skew line for this new variable, we can see that the value now approaches zero (skew is probably somewhere around 0.13), a much greater improvement. Finally, SAS has set the 'Keep' cell for AMOUNT to the value of 'no', but has set the 'Keep' cell for the new variable to 'yes' – this is so that we use the new variable and not the old in our model building process.

Let's take a look at the distribution of this new variable to see the improvement in the distribution. We do this because it may be more appropriate to consider an individual in a group of age ranges than to consider the applicant's precise range.

- Right click on the new variable (AMOU_XXX) and select 'View Distribution'. A histogram will be displayed which looks much more like a bell curve (high in the middle, small on the ends) – the skewness (shift to one side) has for the most part been eliminated.
- Close the histogram window to return to the Variables tab of the Transform Variables window.

Now let's create a set of ranges (stats jocks would call this an ordinal grouping) from the variable AGE.

- Right-click in any cell of the AGE variable row and select the 'Transform' pop-up menu item. Another pop-up menu opens.
- Select 'Bucket'. This selection opens the Input Number dialog box, which is used to define the number of buckets (groups) that you want to create for the AGE variable.

By default, the node creates two buckets. It's tough to read because SAS has a crummy user interface that doesn't move the cursor, but the number in this dialog box really is a '2'.

- Change the number to 4 and click "Close" to accept this change. You will be shown a histogram of responses along with a divider line that indicates the break where the four distinct categories are created. This data is sort of skewed, too, but we'll ignore it since we're simplifying this variable into just four categories and because we want to keep the relatively evenly spaced age ranges that SAS came up with. Keeping this data slightly skewed won't impact the accuracy of our model much.
- Close the Select Values window to accept the defaults and return to the Variables tab. A new AGE_XXXX variable is added and set 'Keep' to 'Yes', while AGE has its 'Keep' cell set to 'No'.
- Close the Transform Variables node, saving changes when prompted. You should be returned to your node diagram.

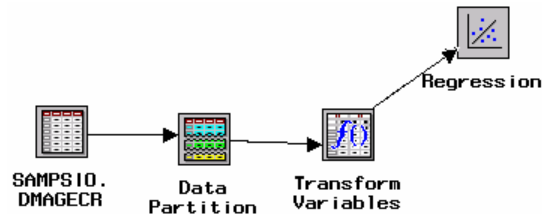
Now that you've 'cleaned up your data' by setting up a new variable to correct the skewness in the amount variable, and set up a variable to indicate age ranges for the age variable, you're now ready to start building models!

Creating a Stepwise Logistic Regression Model

Now let's set up the model building portion of Enterprise Miner. We'll start with the regression model. Many credit organizations use logistic regression to model a binary target such as GOOD_BAD (a binary target is where there are only two answers like 'Good' and 'Bad'). For this reason, we'll consider such a model in our analysis. By using a feature called 'Stepwise', we'll have SAS try various variable combinations for its regression model and select only those variables that it deems most significant in making a prediction of good vs. bad loans.

First let's add a regression node.

- Drag and drop the line 'Regression' from the Tools Palette of the Project Navigator on the left into the Diagram Workspace on the right. You may want to position drop this node to the right and top of the nodes that you already set up. 'Regression' is the first sub-item under the 'Model' heading in the list (you may have to scroll to find it).
- Using the technique described earlier, draw an arrow connecting the Transform Variables node to the Regression node. Your diagram should look like the one below.



Set up the Regression node

- Double click the Regression node's icon to open it. A list of variables will be displayed and the Variables tab will be selected.


Those of you who have taken statistics know that a regression model is a linear model – the variables used will attempt to roughly plot a line. While we could specify the variables to use in our Regression model, we're not sure what the right combination of variables is. Some of these variables may be significant in determining the likelihood of a good or bad loan, some might not be. Since we don't know which variables might be significant, we're going to let SAS identify an appropriate model using a technique called stepwise regression. Stepwise regression systematically adds and deletes variables from the model based on the Entry and Stay significance levels (for the curious stat jocks, the default values for significance is 0.05, but this can be adjusted). Let's configure the stepwise technique.

- Select the 'Selection Method' tab.
- Click the 'Method' drop-down arrow and select the 'Stepwise' item from the menu that appears. The options in this dialog box will become enabled (no longer grayed out). There are a number of options in this and other tabs, but we will be using the default settings for this exercise.
- Save the model using the 'File' menu to select 'Save Model As...'. A 'Save Model As' window will appear.
- Type 'Regression' as the model name and 'Logistic Regression' as the model description.
- Click 'OK' to close the 'Save As...' window and return to the 'Linear and Logistic Regression' window.
- Close the 'Linear and Logistic Regression' window to return to the node diagram.

Now let's train (build) the regression model. SAS will run the stepwise regression and select the variables that go best with a linear model of the form

$$\text{GOOD_BAD} = \text{constant} + \text{coefficient1}(\text{Var1}) + \text{coefficient2}(\text{Var2}) + \dots + \text{coefficientN}(\text{VarN})$$

That form should look familiar from your stats class. If it doesn't, don't worry – SAS is taking care of creating the model for you. To train the model:

- Right click on the 'Regression' node icon in the Diagram Workspace and select the 'Run' menu item. You'll see a border hop from node to node as the diagram you created goes to work reading in data, partitioning it, transforming it, then building an optimal regression model using the stepwise technique. This will take a few seconds.
- When completed successfully, SAS will notify you and ask if you want to see results. Click 'Yes'.
- A list of the variables that Stepwise has suggested for your model is displayed in graphic form. The variables are listed as X_n because there simply isn't enough space to write out each variable name. If you are interested in exploring the variable names and statistical values for each item, click the 'View Info' tool  in the toolbar, then click and hold the mouse down on the respective bar. The details on this item (the variable label & its t-score) are displayed. The variables are presented from left to right based on descending absolute value of the variable's t-score (an indicator of the strength of significance of each variable in the model). The color of the bars indicates whether the t-score is actually positive (red) or negative (yellow). More statistics are available in the 'Statistics' and 'Log' tab, but you don't have to examine these unless you're curious. On this graphics page, “effect” refers to a variable.
- Close the window and return to the nodes diagram.

NOTE: If at any time you want to refer back to this results diagram, just right-click on the regression node from the nodes diagram and click “results.” This holds true for later results for the neural networks and tree diagram nodes, as well.

Creating a Neural Network Model

IMPORTANT LEARNING NOTES: In some situations, neural networks can be superior to regression models. Regression models are basically linear. If there is a non-linear relationship in your data, the Regression model wouldn't be appropriate to use. Folks with statistics backgrounds know there are ways to finesse regression models using polynomial and interaction terms, but we won't get into this now (again, this isn't a statistics course). And even these techniques aren't guaranteed to work

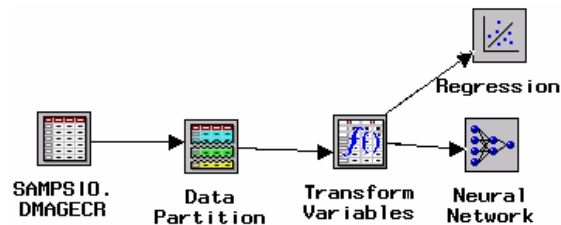
Another way around the problem of modeling data that contain non-linear relationships is to use a neural network. The type of neural network we're creating is called a multilayer perceptron (MLP) model (my, that sounds geeky!). Don't be intimidated – it's just the default technique that SAS Enterprise Miner uses, so we'll stick with the defaults for this exercise. Neural networks are flexible classification methods that, when carefully tuned, often provide optimal performance in classification problems such as this one.

Unfortunately, neural networks are notoriously difficult to interpret – particularly when it comes to assessing the importance of individual inputs on the classification. To put this another way, we might see that a neural network model comes up with superior results, but due to the complexity of the model, we might not be able to tell why it makes the recommendations it does. For this reason, neural network models have come to be known as "black box" predictive modeling tools.

The fact that neural networks are difficult to interpret creates a problem with their use. The Equal Credit Opportunity Act mandates that any models used to determine an applicant's credit worthiness be easily interpreted, in part, to ensure that discriminatory practices haven't been used in the loan process. Both standard regression models and decision tree models are considered easy to interpret and are therefore accepted methods for determining the creditworthiness of a mortgage applicant. Even though neural networks may provide a more predictive model, their use for certain applications may be limited by U.S. law, because they are not transparently interpretable. Since we're working with German data in this exercise (and since it's just an exercise), we'll ignore U.S. Law for now. It should also be noted that neural networks are extremely useful in many areas where interpretability may not be necessary. These include models to determine customer preferences or fraud prevention.

Let's create a neural network model!

- Drag and drop the line 'Neural Network' from the Tools Palette of the Project Navigator on the left into the Diagram Workspace on the right. You may want to position this node to the right of the Transform Variables node. 'Neural Network' is the third sub-item under the 'Model' heading in the list (you may have to scroll to find it).
- Using the technique described earlier, draw an arrow connecting the 'Transform Variables' node to the 'Neural Network' node. Your diagram should look like the one listed below.



- Double click the 'Neural Network' icon to open the Neural Network node. The Neural Network window will open and a list of variables will be displayed. These variables are the same ones as displayed when you opened the previous regression node. It's simply the variables in our data set.
- Save the model using the 'File' menu to select 'Save New Model' (**NOT** Save As...). Enter 'Neural' as the name and 'Neural Network Model' as the description.
- Select 'OK' to close the 'Save Model As' window.
- Close the 'Neural Network' window.

Now let's run the Neural Network node to build our neural network model.

- Right click on the Neural Network icon and select 'Run' from the menu.
 - You may receive a Windows Security Alert mentioning that SAS is blocked from accessing the Internet. If so, click 'OK'.
- The 'Neural Network Monitor' window will appear (it may flash by super fast, so you may miss all of the stuff mentioned in this bullet point, particularly if the Windows Security Alert showed). If the 'Train' and 'Valid' lines do not begin snaking across the

'Monitor' graph, click the 'Continue' button at the top of the screen & the neural network training should begin. You should see (but it may flash by too quickly) the two lines travel through 100 iterations. What's happening is that SAS is trying various combinations to build an optimal neural network. It will stop when the lines reach 100. You will be returned to the node diagram page and see a dialog box stating that the Neural Network has completed successfully.

- Click 'Yes' to see results.

The results probably won't mean much to you if you don't have a strong knowledge of statistics. The important thing to note is that SAS used the historical data we had to build a predictive neural network model that can be used to predict if we should accept or reject a loan applicant. For the curious, the Plot tab of the Results window shows the average squared error for each iteration of the training and validation sets. The vertical line shows where the optimal average error was achieved. Anything beyond this line was an 'overtraining' of the validation data set. It's also worth noting that your results may differ slightly each time you run the neural network (so your results are likely different from your classmates'). This is because the neural network technique used uses a random 'start' seed as the beginning point to begin its exploration for an optimal model.

- Close the 'Results' window for the neural network run and return to the node diagram.

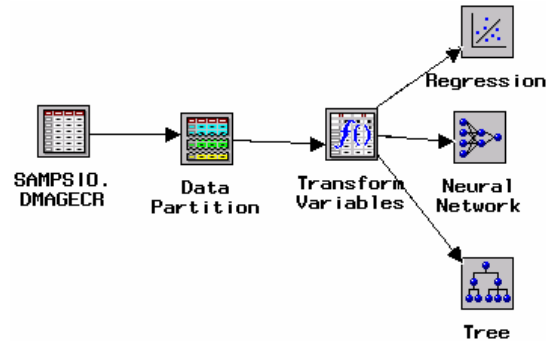
Creating a Tree Model

IMPORTANT LEARNING NOTE: Decision Tree models create a set of simple rules that break data up according to the value of a particular variable. Each fork or 'breakpoint' in the tree either leads to another rule acting on another variable and another breakpoint, or to a decision (in our case either to accept or reject a loan). You can consider decision trees as following a sort of if-then logic similar to expert systems (examples of rules might be things like 'if the applicant has a high checking balance and has been on the job for more than 1 year then approve the loan', or 'if the applicant has a low checking balance and has been on the job less than 1 year, reject the loan', etc.). Unlike an expert system, the decision tree doesn't get its expertise from an expert. Instead, it performs a statistical examination of the data and creates a tree (in our case, a set of if/then rules, with each decision path terminating in a decision to accept or reject the loan). Decision trees are also strong in modeling non-linear relationships. They are also particularly good when data contain missing values (a common occurrence when organizations are gathering data from disparate sources).

Let's create a decision tree model.

- Drag and drop the line 'Tree' from the Tools Palette of the Project Navigator on the left into the Diagram Workspace on the right. You may want to position this node to the lower right of the Transform Variables node. 'Tree' is the second sub-item under the 'Model' heading in the list (you may have to scroll to find it).

- Using the technique described earlier, draw an arrow connecting the 'Transform Variables' node to the 'Tree' node. Your diagram should look like the one listed below.



- Double click the 'Tree' icon to open this node. The list of variables from our data input file will be displayed in a window labeled 'Tree'.

Stats jocks could do a lot of customizing of the tree algorithm. We'll just accept the defaults that SAS prompts us with, so let's save this model.

- Select 'Save Model As...' under the File menu.
- Enter 'Tree' as the model name, 'Decision Tree' as its description, and click 'OK'.
- Close the 'Tree' window and return to the node diagram.

Now let's run the Tree node to create the Decision Tree model.

- Right-click on the 'Tree' icon and select 'Run'.
- After a few seconds, you will be informed that the Decision Tree ran successfully. Click 'Yes' when asked if you want to view results.

The information in this window provides detailed information on how the optimal tree model was arrived at. You'll see a summary table in the top left portion of the window, a tree ring navigator in the top right (a graphical display of possible data segments from which to form a tree), and assessment tools in the lower half of the window. We won't get into all the stats in this exercise – you're just 'test driving' Enterprise Miner. However, let's take a look at the graphical representation of the tree so that we can get an idea what this looks like.

- Select 'Tree' from the 'View' menu.

You'll see the tree diagram starts with a single node (called the root) at the top. Each break point lists a variable name. As you scroll down the tree may be re-drawn (not the nicest user interface, but the results remain the same).

For the curious, each row of a node contains the following statistics:

- The first row lists the percentage of good values in the training and validation data sets.
- The second row lists the percentage of bad values in the training and validation data sets.
- The third row lists the number of good values in the training and validation data sets.

- The fourth row lists the number of bad values in the training and validation data sets.
- The fifth row lists the total number of observations in the training and validation data sets.
- The sixth row lists the decision alternative assigned to the node (accept or reject).
- The seventh row lists the expected loss for the accept decision in the training and validation data sets.
- The eighth row lists the expected loss for the reject in the training and validation data sets (all reject decisions show no loss, this is consistent with how we configured the target profile earlier).

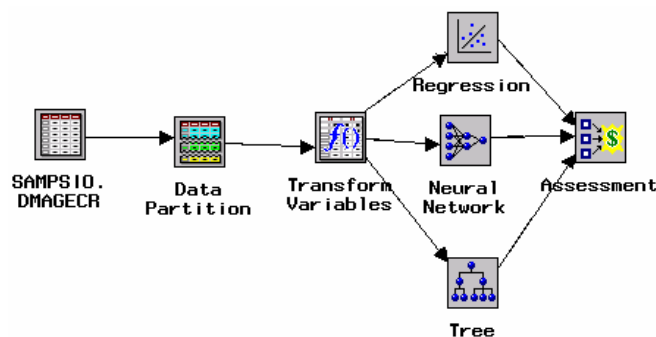
You'll also see that under some variables the nodes branch off based on a range (e.g. something like $DURATION < 22.5$, $DURATION \geq 22.5$). For others, the branching occurs if a value is in a particular group (e.g. take one path if the value contains a 1 or 2 – listed as .2, take the other path if it contains a 3 or 4 – listed as .4). Each path ends up in a final accept or reject decision. If this were a statistics course we'd interrogate the tree much more to understand how it was constructed, but we're simply taking a gentle test-drive, so let's return to the node diagram so that we can compare our three models.

- Close the Tree Diagram and Tree Results windows so that you're back at the node diagram.

Assessing the models

The assessment node will allow you to compare the three models you created based on characteristics such as predictive power. Let's set up an assessment node.

- Drag and drop the line 'Assessment' from the Tools Palette of the Project Navigator on the left into the Diagram Workspace on the right. You may want to position this node to the right of the 'Neural Network' node. 'Assessment' is the first sub-item under the 'Assess' heading in the list (you may have to scroll to find it). It's OK if you have to scroll the diagram workspace to make room on the right for the new nodes.
- Using the technique described earlier, draw three arrows connecting the 'Regression', 'Neural Network', and 'Tree' nodes to the 'Assessment' node. Your diagram should look like the one listed below.



- Double click on the 'Assessment' icon to open this node. The 'Models' tab of the 'Assessment Tool' window is displayed.
- Select all three models by clicking on the first row, holding down your mouse, and dragging your mouse pointer across each model row entry to the bottom row. The three rows should be highlighted as depicted below.

Models									
Options									
Reports									
Output									
Tool	Name	Description	Target	Target Event	By Group ID	By Group Description	Root ASE	Valid Root AS	
Tree	Tree	Decision Tree	GOOD_BAD	good			0.4074465608	*****	
Neural Network	Neural	Neural Network Model	GOOD_BAD	good			0.3929272612	*****	
Regression	Regression	Logistic Regression	GOOD_BAD	good			0.396539024	*****	

Create a lift chart to compare the three models.

- Use the 'Tools' menu to select the 'Lift Chart' item. A lift chart will be displayed

For this chart, the customer cases are sorted from left to right by individuals most likely to have good credit as predicted by each model. The sorted group is then lumped into ten deciles (groups of 10%) along the X axis. The left-most decile is the 10% of the customers most likely to have good credit. The vertical axis represents the actual cumulative response rate in each decile.

The lift chart displays the cumulative % response values for a baseline model and for the three predictive models. (The baseline model is essentially no prediction – earlier we stated that 9 out of 10 customers should have good credit, so this is why the line is flat at 90%). By looking at which line is consistently 'higher', we can identify which model seems to perform better. Unfortunately, in some circumstances there may not be a clear 'winner'. Your comparison probably shows one model doing better in certain deciles, but perhaps underperforming in others.

Print out a copy of this chart to hand in

- Select 'Print' from the 'File' menu. Collect your printout & write your name and section number on it to be handed in. If your printout does not clearly indicate which line represents which item in the key, please compare your printout with what's on your screen & indicate each line on your printout by writing an "N", "T", and "R" respectively next to the appropriate line.

There does not seem to be a clear-cut winner or 'champion' model to use for subsequent scoring to determine if new applicants should be accepted or rejected. However, each model performs better than the baseline of using no model at all. The German credit data set that we're using is highly random, which makes it difficult to develop a really good model. This is typical of data mining problems. While we would have liked to have identified a clear 'champion' model, let's decide to use the neural network model to score a file named SAMPSIO.DMAGESCR. This file contains records of new customers that have yet to be approved or rejected. In practice, one might build a model like this and use it as part of a decision support system in the loan approval process. Loan evaluators might run data through the system, examine the accept/reject decision, and consider these results along with criteria that are not captured in the model.

Let's select the 'Neural Network' model as the one to use to accept or reject records in the new data file.

- Close the lift chart window to return to the 'Assessment Tool' window listing the three models.
- Select the 'Output' tab.
- Click the 'Neural Network' entry in the 'Highlight Model for Output:' list box.
- Close the Assessment window to return to the node diagram. The 'Neural Network' model will remain selected as our 'preferred' model for subsequent use.

Scoring New Applicant Data

The purpose of predictive modeling is to apply the model to new data in order to improve our decision-making accuracy (to achieve some objective like minimizing loss or maximizing profit). For this exercise, we'll assume that the file named SAMPSIO.DMAGESCR contains new applicants that haven't received an accept or reject decision.

First, set up the Input Data Source for this new file.

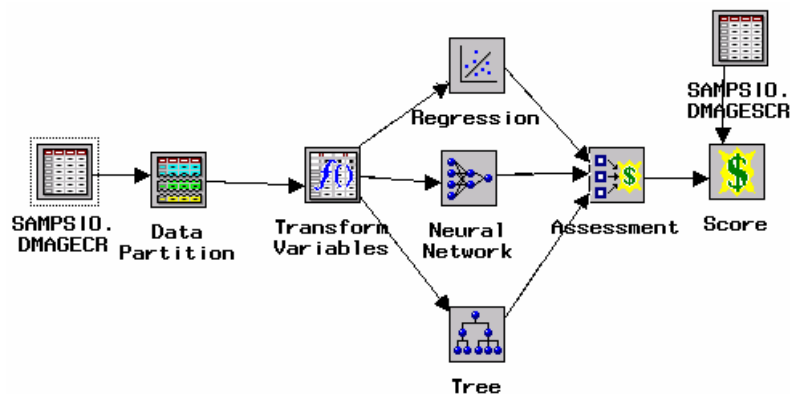
- Drag and drop the line 'Input Data Source' from the Tools Palette of the Project Navigator on the left into the Diagram Workspace on the right. You may want to position this node to the upper right of the 'Assessment' node. 'Input Data Source' is the first sub-item under the 'Sample' heading near the top of the list (you may have to scroll to find it). It's OK if you have to scroll the diagram workspace to make room on the right for the new nodes.
- DO NOT draw an arrow between nodes yet. Instead, double click the 'Input Data Source' icon to open the node.
- Click the down arrow beside the text field named 'Role' and select 'Score' from the drop down menu (this item was likely previously set to 'Raw').
- Type SAMPSIO.DMAGESCR in the 'Source Data' field and press the Select button. You should see a screen that reads “Library: SAMPSIO” at the top and a list of tables below. In the “Tables” list, DMAGESCR should be highlighted. Click “OK” at the bottom of the page to continue. After a few seconds, a description and sample information will be loaded into the dialog box. It will show that the size of the sample is 75, indicating that you're going to 'score' 75 items to determine if they should be accepted or rejected. Your dialog box should look like the one below.

Data	Variables	Interval Variables	Class Variables
Source Data:	SAMPSIO.DMAGESCR <input type="button" value="Select..."/>		
Output:	EMDATA.VIEW_249		
Description:	SAMPSIO.DMAGESCR		
Role:	SCORE <input type="button" value="v"/>	Metadata sample:	
Rows:	75	Size:	75 <input type="button" value="Change..."/>
Columns:	21	Name:	EMPROJ.SMP_V12Q

- Close the Input Data Source window and save changes when prompted.

Now set up the scoring node so that you can score this new data set using the neural network model.

- Drag and drop the line 'Score' from the Tools Palette of the Project Navigator on the left into the Diagram Workspace on the right. 'Score' is the first sub-item under the 'Scoring' heading near the bottom of the list (you may have to scroll to find it). It's OK if you have to scroll the diagram workspace to make room on the right for the new nodes.
- Using the technique described earlier, draw an arrow connecting the 'Assessment' node to the 'Score' node.
- Draw an arrow connecting the Input Data Source node you just added (now labeled 'SAMP5IO.DMAGESCR') to the 'Score' node. Your diagram should look like the one below.



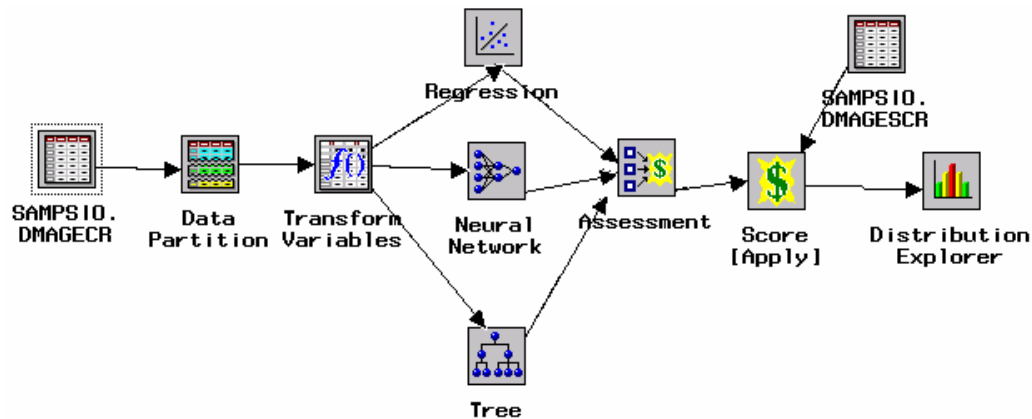
- Double-click the 'Score' node to open it. The settings tab is displayed.
- Select the radio button labeled 'Apply training data score code to score data set'.
- Close the 'Score' window, clicking 'Yes' to save changes when prompted.

Now score the new data set.

- Right-click the 'Score' node and select 'Run'. The neural network model will automatically be applied to the new data & each record will be given an accept or reject recommendation. After a second or two, SAS will inform you that the run for the score has been completed successfully. You can actually select 'No' when asked if you want to view results, because we're going to be using the 'Distribution Explorer' node to explore the results of the scoring.

Add the Distribution Explorer node.

- Drag and drop the line 'Distribution Explorer' from the Tools Palette of the Project Navigator on the left into the Diagram Workspace on the right. 'Distribution Explorer' is the first sub-item under the 'Explore' heading, which is toward the top of the list (you may have to scroll to find it). It's OK if you have to scroll the diagram workspace to make room on the right for the new node.
- Draw an arrow connecting the 'Score' node to the 'Distribution Explorer' node. Your final node diagram should look something like the one below.



- Open the Distribution Explorer node by double-clicking on it. The 'Distribution Explorer' window will open, displaying a list of variables.
- Select the 'Data' tab at the top of this window.

SAS exports the results of the scored data into a file that has a suffix containing .SD_. We need to select this file so that we can explore the results.

- Click the 'Select...' button. An 'Imports Map' window will appear that shows the name 'Predecessors' with 'Score (Apply)' under it.
- Click the plus '+' sign next to the 'Score (Apply)' item to expand this list.
- An item named 'SAS_DATA_SETS' will be displayed. Click the plus '+' sign next to this item to expand the list.
- The expanded list might be drawn strangely, but when you click to scroll the list down, the list will be re-drawn when you get to the bottom (Amazing what passes for a \$90,000 software program, eh?).
- Find and select the score data. This data contains the suffix ".SD_". When you click on it, you'll notice that the name of this file appears in the 'Selected Data' field just below this list.
- Click 'OK' to accept this selection.

We're going to explore the scored data by looking at plots of the results. In order to do this, we need to tell SAS which variables to use for the X and Y axis. The two variables that we care about that SAS created are:

- EL_GOOD_BAD_ - contains the expected loss values for making the good decision.
- D_GOOD_BAD - assigns either the accept or reject decision status to an applicant in the score data set.

SAS has actually created many more variables that we'd get into if this were a stats class, but we'll ignore these for purposes of this exercise.

Assign EL_GOOD_BAD_ as the x-axis variable.

- Click the 'Variables' tab. A list of variables will be displayed.
- Scroll to the bottom of the list to find the EL_GOOD_BAD_ variable. Right-click in the Axis cell for the variable EL_GOOD_BAD and select 'Set Axis' from the popup

menu. Another menu will appear. Select 'X'. The letter 'X' will appear in the Axis cell for this variable, indicating that you set things up correctly.


Now assign D_GOOD_BAD_ as the y-axis variable.

- Find the D_GOOD_BAD_ variable (probably second from the bottom). Right-click in the Axis cell for the variable D_GOOD_BAD and select 'Set Axis' from the popup menu. Another menu will appear. Select 'Y'. The letter 'Y' will appear in the Axis cell for this variable, indicating that you set things up correctly.

Now, view a bar chart of the accepted and rejected customers in the score data set.

- Select the 'Y Axis' tab.

The chart looks nice (probably showing a roughly 60/40 accept/reject split). It's a good thing we ran these models, since the data we received likely contained many more bad loans that are typical in a larger sample (which we were told was a 9/1 good/bad ratio). We can actually determine the percentage of accepts and rejects by using an icon in the icon bar that SAS refers to as the 'Probe' tool.

- Click the 'Probe' tool  icon in the icon bar.
- Now move the mouse within the 'Accept' bar and click and hold down the mouse. As long as you hold the mouse, a little information balloon will display the percentage of loans in the data set that your model determined to accept. Try the same with the 'Reject' bar.
- Print out this graph – you worked hard to produce it. Write your name and section number at the top and hand it in with other print out you made of the lift chart comparing the three models.

You can now Exit SAS Enterprise Miner, Exit SAS, and log off your machine.

That seemed like a lot of work to go through just to view the graph. If we were running through this exercise in a 'real' situation, we would next write a program to create a separate file listing each applicant, the 'score' assigned to that applicant, and the model's accept or reject decision. This would involve a little bit of coding. We'll skip the coding here. Once this coding was done, the data mining tool would be automated to score any new data entered in a data file and it could be easily run any time you want.