CIS-445 DATA MINING

Project 3

Due: See Blackboard

Worth 100 points

Objectives

Learn about **Commercial Data Mining Tool: SAS Enterprise Miner (EM) 14.1**

Task: Classification/Prediction of the Binary Outcome
Model: Neural Network
Nodes: Input Data, Data Partition, Neural Network, StatExplore or Graph Explore, Transform Variables, Replacement, Impute, Variable Selection, Model Comparison, Control Point, Reporter
Software: SAS 9.4, SAS Enterprise Miner 14.1

Important Notes and Directions/Clarifications – Read them before you start the Project

As you become more experienced with SAS EM 14.1 and the data mining issues, my directions will be less explicit and I will only provide them when I feel that they are absolutely necessary. The project builds on knowledge you have gained in the two previous Tutorials and Project 2.

In this project you will use a HMEQ data set containing 13 variables and 5960 observations. The SAS data set resides in the SAS library named SAMPSIO. This is the same data set that you have used in Tutorial 2. You will find this data set when you follow the procedure for reading data sets in the Input Data node. This historical data set contains information about the customers whom the bank has extended a loan in the past. Out of 5960 applicants, about 20% have defaulted. The data set contains 12 independent variables and one dependent/target variable named BAD that will be used as the binary target (dependent/output) variable. The variable BAD has two class labels: 0 and 1 for those who repaid and defaulted on a loan, respectively. The following table describes the variables.

Variable number	Name of variable	Description
1	BAD	1=client defaulted on loan, 0=loan repaid – **Target, Binary**
2	LOAN	Amount of the loan requested
3	MORTDUE	Amount due on existing mortgage
4	VALUE	Value of current property
5	REASON	DebtCon=debt consolidation, HomeImp=home improvement
6	JOB	Six occupational categories
7	YOJ	Years at present job
8	DEROG	Number of derogatory reports
9	DELINQ	Number of delinquent payments
10	CLAGE	Age of oldest trade lines in months
11	NINQ	Number of recent credit inquires
12	CLNO	Number of trade lines
13	DEBTINC	Debt-to-income ratio

Using the data set you will build (train, validate/test) several neural network (NN) models and compare their classification accuracy. You can then choose the best model and "implement" it in a real world environment. This model would allow a bank to predict whether a new loan applicant, described by variables 2 through 12, will default on a loan or not. For new loan applicants the class labels (0 or 1) for the target variable BAD are obviously unknown. Simply, there is no target variable. We would expect that the model would generate the appropriate class label (0 or 1) and the associated probability, and tell us how likely it is that each new

applicant would default upon a loan or repay a loan. This would have been achieved by scoring a data set containing new loan applicants by the Score node. Recall Tutorials 1 and 2 as well as Project 2 about Widget Buyers. We will not use the Score node in this project.

All variables in the data set seem to be relevant and banks indeed use these (and other) variables to evaluate the creditworthiness of their customers. The number of cases/samples in the data set (which is divided into training, validation, and test cases) seems to be sufficient as well. This is true especially for class 0 (loan repaid). The current data set, which is dominated by applicants who repaid a loan (80%), will likely create the response model that will tend to classify the majority of "bad" applicants as good prospects for repaying a loan. It is obviously the false positive error that banks would like to avoid. Applicants who defaulted upon a loan are underrepresented in the data set. In a real world project, however, it would be necessary to balance the data set better so that it contains approximately the same representation of applicants who repaid and defaulted on a loan. We will try to balance the data set better in the next project.

In the Input Data node explore the distribution of input variables and the target variable. Also explore the variables using the Graph Explore node or the StatExplore node for this purpose.

Use the following setting in the Data Partition node: 50%, 50%, 0% for the training, validation, and test sets, respectively. So you will look at the performance of the models on the validation set.

Use the Replacement/Impute node to eliminate missing values. This is an important step as unlike decision trees, neural networks do not tolerate missing values well. They simply reject all observations which have one or more missing values. This would substantially decrease the amount of observations in the data set.

Create two paths in the workflow, with and without the Transform Variables node. Use the Transform Variables node to create new variables INDELINQ, INDEROG, log (YOJ) and use the Interactive Binning node for the variable NINQ exactly as you did in Tutorial 2. Each path with and without the Transform Variables node should have three Neural Network (NN) nodes.

Like in Tutorial 2 use the Variable Selection node as neural networks may work better with fewer variables. The variables which have too little prediction power will be rejected.

To avoid excessive number of connections between the nodes, you may use the Control Point node.

In each of the two paths with three NN nodes use a different number of neurons in the hidden layer, say, 1, 3 (the default), and 5, for example.

Pass all 6 models to the single Model Comparison node to compare the classification results from the 6 models.

At the end attach the Reporter node, run it, and scan the pdf file it created for each node in your workflow. To summarize, you may use a similar workflow as in Tutorial 2, but use only neural networks as the models.

We will be interested in the performance of the models generated on the validation set only as, similarly to the test set, it provides a good and unbiased indication on the future performance of the built models. This is the approach that SAS uses.

By default, the Neural Network node uses 3 neurons the hidden layer, and the number of neurons in the output layer is equal to the number of classes in the target variable. Three neurons in the hidden layer not necessarily produce the best classification rates. Neural network application developers usually try several neural network architectures by changing the number of neurons in the hidden layer, training parameters, and finally trying different learning

algorithms. Then they choose the one that produces the best classification rates for the validation and test sets.

The SAS EM 14.1 runs on top of SAS 9.4, extremely powerful and huge software. Again, do not worry that in the nodes you use, including the Neural Network node, there will be many items that you do not fully understand. As I mentioned in the previous tutorials, to be a real expert in data mining or to obtain an M.S. degree in the field of data analytics, one would have to take about 10 other classes. Nevertheless, the project will give you a good exposure to the data mining issues.

Use on-line Help to obtain additional information or consult with me. Select Help from the menu and then select the topic on which you need help or search for the topic using keyword(s).

Turn in:

1. Write a 2-page single-spaced report to briefly describe the project and to summarize your results. Use Time-Roman font, point 12 and 1" for the top, bottom, right, and left margins. In the report:

    A. Describe briefly the nodes in the diagram and each path of the workflow.
    B. Discuss the confusion matrices and include the confusion matrix for the model which you deem is the best and generated the best classification results on the validation set.
    C. Include and discuss the ROC charts which show you the global performance of all 6 models for the continuum of cutoff points from within the range [0, 1]. (The standard cut-off point is 0.5.) Discuss the lift charts. Each of the two charts should have 6 models.
    D. For this application, which NN model (with how many neurons in the hidden layer) and in which of the 2 paths would you choose as the best model to classify future customers?
    E. Is it better to transform variables or not?
    F. Does variable selection help in achieving better classification accuracy rates?
    G. Is it easy to interpret the weights of your best NN model or any NN model?
    H. Analyze your best NN model with respect to the overall correct classification accuracy rates, correct classification accuracy of bad and good loans, as well as the false positive and false negative classification errors.
    I. Does your best NN model tend to classify more bad loans as good ones?

2. It is essential that you discuss your results adequately well! (The confusion matrices and the two charts do not count toward a 2-page length of the report.)

3. Attach to your report:

    A. The final workflow diagram with all the nodes you have used. I need to see the checkmark on each node and the last node (the Reporter node) that ran successfully.
    B. The confusion matrices representing the correct classification accuracy rates for the standard 0.5 cut-off for the validation data set only for all 6 NN models, with and without transforming variables.
    C. The lift chart and the ROC chart from the Model Comparison node. On each chart you should have 6 curves, one for each NN model.

4. Merge all items from points 1 and 3.A-3.C into a single file, name it as *Project3_YourLastname_YourFirstName.pdf*, save it in the Project3 folder, and e-mail to jozef.zurada@louisville.edu.