Jenet Baribeau
CIS 445-01
Project 3

1.

A. Describe briefly each path of the workflow.

After importing the data file, we explored what the status of the data would look like without manipulating it. This demonstrates the need to contend with the missing values. The second work flow element is partitioning the data to create the training and validation sets. The data partition also provides model weights for estimation prior to to assessment. Since this diagram uses a Neural Network its necessary to impute the data and replace missing values. I chose Impute over replacement because the replacement seemed to be too general and the impute node favors the modal qualities of the elements. I chose impute because we used it once before in a previous project or tutorial.

There are 2 channels to analyze which channel would be the best – Variable Selection or Transform Variables into Bins. At the split, the transformation node followed by the interactive binning which is used to model nonlinear functions of continuous distributions. New variables were used to maximize node and create new variables minus the NINQ. The NINQ variable was binned. The final analysis was the Neural Networks after being binned.

The secondary channel, Variable Selection, which evaluated the importance of the input variables in prediction. I did not use the Control Point as I didn't know really how it contributed to the final outcome. The Neural Networks followed just like previously selected a total of 6 NN in each channel there was a node with 1 hidden layer, 3 hidden layers, and 5 hidden layers.

The Model Comparison that provided a common framework to compare all 6 NN models. The recommendation was the NN 6 which contained 5 layers without the variable transformation.

Lastly, the Report Node which provided a comprehensive review of the nodes demonstrating which nodes are under consideration for either channel.

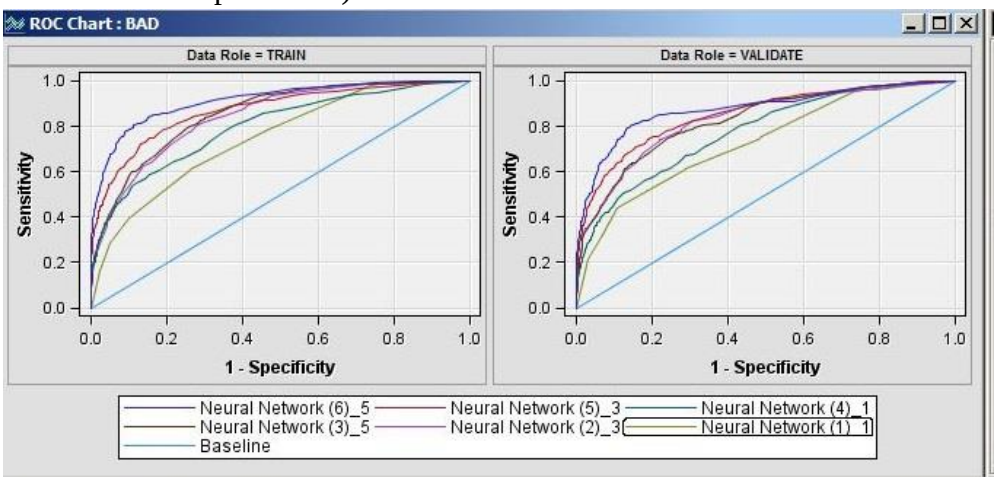
B. Discuss the confusion matrices and include the confusion matrix for the model which you deem is the best and generated the best classification results on the validation set.

The confusion matrix showed that the NN 1 with only 1 hidden layer provided the least amount of false positives but the most false negatives. Giving the best choice as the NN 6 with 5 hidden layers and no variable transformation. With a 3% Type II error, the Neural Network 6 with 5 hidden layers has less dangerous error.

Event Classification Table
Model Selection based on Valid: Misclassification Rate (_VMISC_)

Model Node	Model Description	Data Role	Target	Target Label	False Negative	True Negative	False Positive	True Positive
Neural4	Neural Network (4)_1	TRAIN	BAD		399	2313	72	195
Neural4	Neural Network (4)_1	VALIDATE	BAD		409	2289	97	186
Neural5	Neural Network (5)_3	TRAIN	BAD		327	2313	72	267
Neural5	Neural Network (5)_3	VALIDATE	BAD		321	2298	88	274
Neural6	Neural Network (6)_5	TRAIN	BAD		208	2271	114	386
Neural6	Neural Network (6)_5	VALIDATE	BAD		216	2230	156	379
Neural	Neural Network (1)_1	TRAIN	BAD		495	2323	62	99
Neural	Neural Network (1)_1	VALIDATE	BAD		467	2311	75	128
Neural2	Neural Network (2)_3	TRAIN	BAD		371	2275	110	223
Neural2	Neural Network (2)_3	VALIDATE	BAD		335	2234	152	260
Neural3	Neural Network (3)_5	TRAIN	BAD		389	2310	75	205
Neural3	Neural Network (3)_5	VALIDATE	BAD		367	2281	105	228

C. Include and discuss the lift and 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 .5)



This ROC chart demonstrates the tradeoff between sensitive and specificity by the blue colored curve (Neural Network (6)_5) proving to be the most accurate in predictions. The yellow colored curve being the lowest and least accurate as it's closest to the baseline representing an area of .5 being worthless which was also the cutoff. Sensitivity + Specificity = 1 so the closer to .5 the line, the less accurate.

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?
The neuralnet with 5 hidden layers and only the variable selection proved to be the best method to classify future customers. Not only did it have the least percentage of errors, but also there was the strongest correlation.

E. Is it better to transform variables or not?
The evidence in this example shows that it was better to not transform the variables but rather choose the variable selection to review the best elements and utilize hidden networks to transform the variables usefully.

F. Does variable selection help?
Variable selection helped greatly with the outcome. The transformation variables supplied 1 non-useful line with the 1 hidden layer didn't provide any specificity. Variable selection is a means to supply elements to construct a model that can predict the relationships in the data. Variable selection alone is not an effective determiner for determining predictions.

G. Is it easy to interpret the weights of your best NN model or any NN model?
The NN model finds a linear relationship between several elements that don't have an obvious relationship. Minimizing the risk by determining the bias with high variance. Easy to determine the weights for the program to determine but would lengthy to

determine by hand. As there are several other elements that contribute to the determination. The iterative process of determining the weights is lengthy but easily processed through by the SAS process.

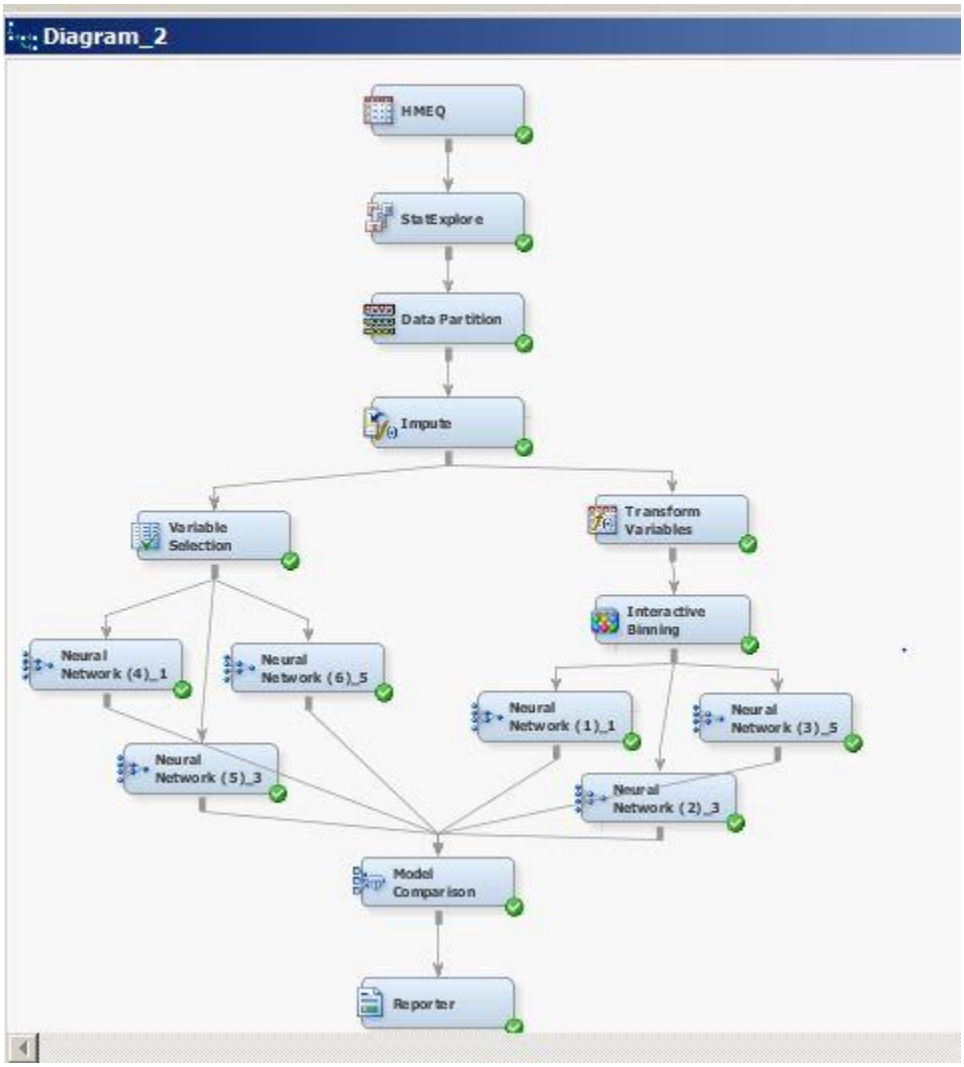
H. Analyze your best NN model with respect to the overall correct classification accuracy rates, correct classification accuracy of bad and good loans, as well as the false positive and false negative classification errors.
The smallest classification error was in the range of 3% and up to 8% with the Type II errors being the more dangerous of the errors by predicting a false positive. The false positive would represent that they would grant a loan to someone who didn't really qualify. The NN (6) option had the least highest Type II error and less than 2% of the false negative rate but it still had the highest Type 1 error of the other neural networks analyzed. Here it would be better perhaps that a few good loans are denied then several loans are issued to bad credit with the possibility of defaulting.

I. Does your best NN model tend to classify more bad loans as good ones?
Neural Network 6 classified many more bad loans than good ones in relation to True positives. However, the true negatives were in tight relation to the other neural networks. The True positives showed much more than the other neural networks by more than 100 approvals over the other neural networks.

2. In reviewing the charts and the data, I think that the Neural Network 6 is the most accurate. The multilayer implementation allows for the testing of the performance by choosing the data to be reviewed. The elements of the data the credit history (DELINQ), the urgency of the requirement (NINQ) and the number reports(DEROG) changing these elements to allow for more specificity created more bad than good outcomes. The LOG(YOJ) and some correlation but not as strong as the DELINQ did.
The number of hidden neurons was more than I thought necessary and could potentially lead to overfitting of the bad loan capacity. As 5 hidden layers seems potentially high with all of the error types falling within less than 1% of each other. Not thoroughly understanding the potentiality of over-fitting and underfitting is difficult. Comparatively the analysis presented that classifying loans can be further improved by coupling the analysis with other known personal information that perhaps cannot be accounted for in an algorithm. Sometimes good faith in people can affect behavior for the better.

3.

A. Workflow Diagram

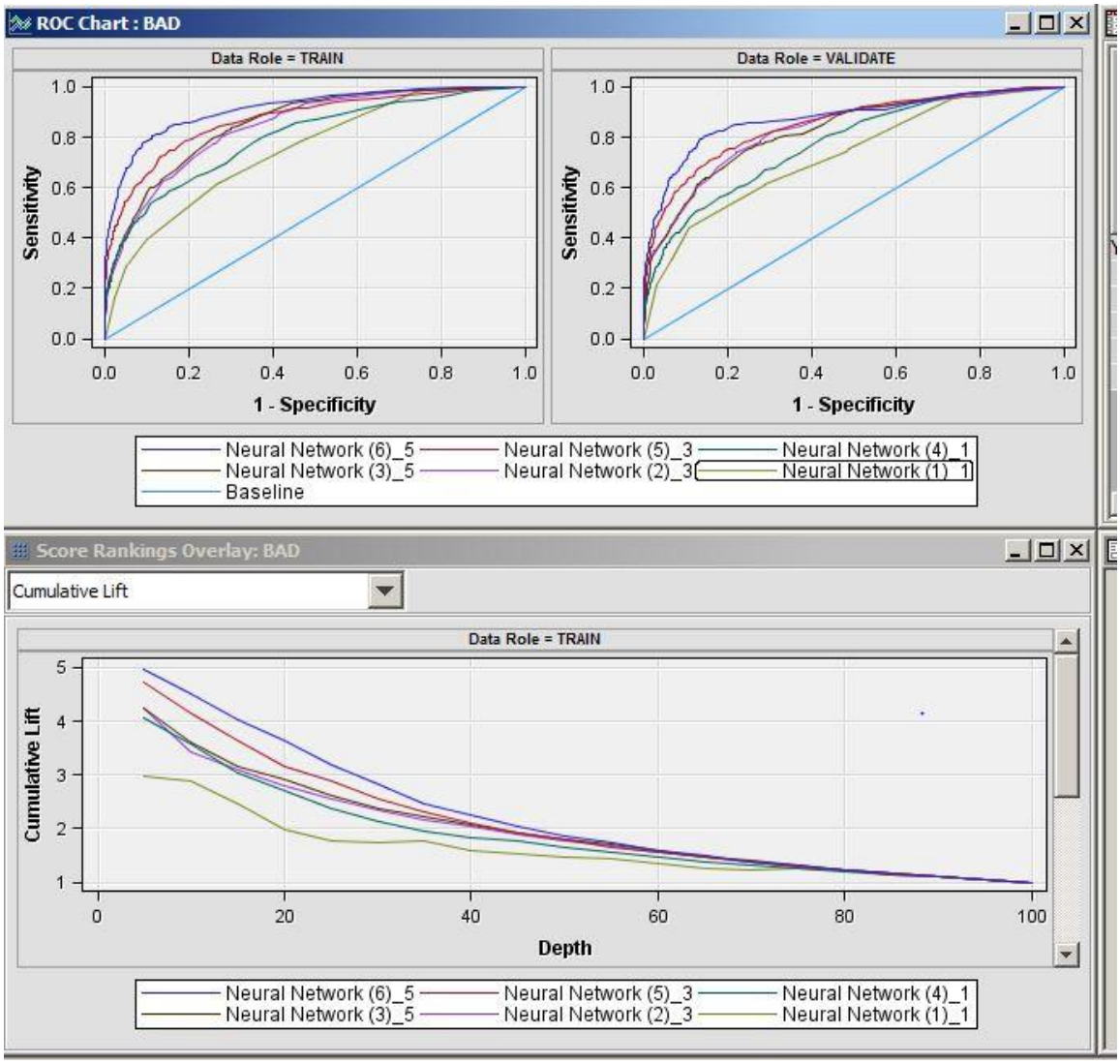


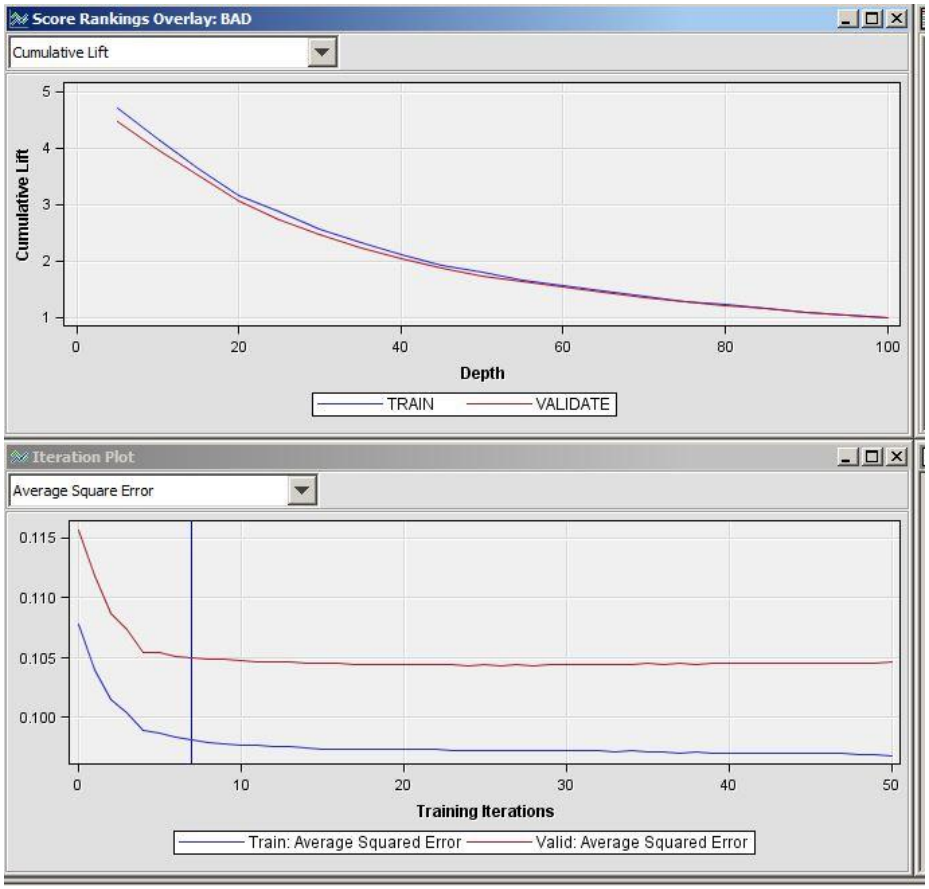
B. Confusion matrices

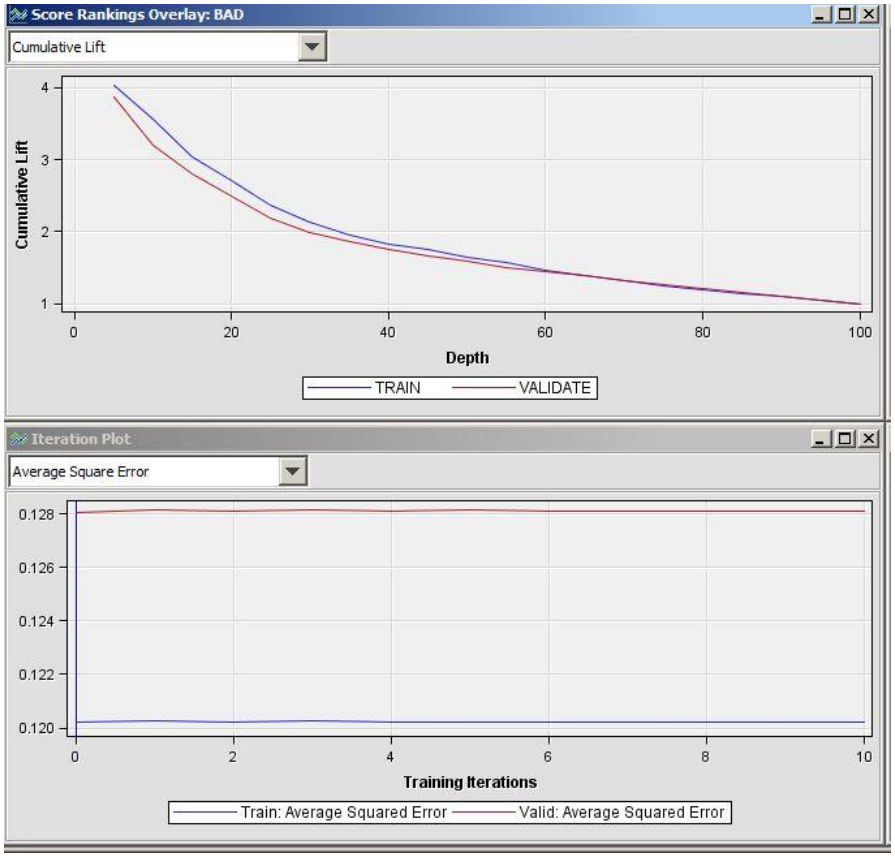
Event Classification Table
Model Selection based on Valid: Misclassification Rate (_VMISC_)

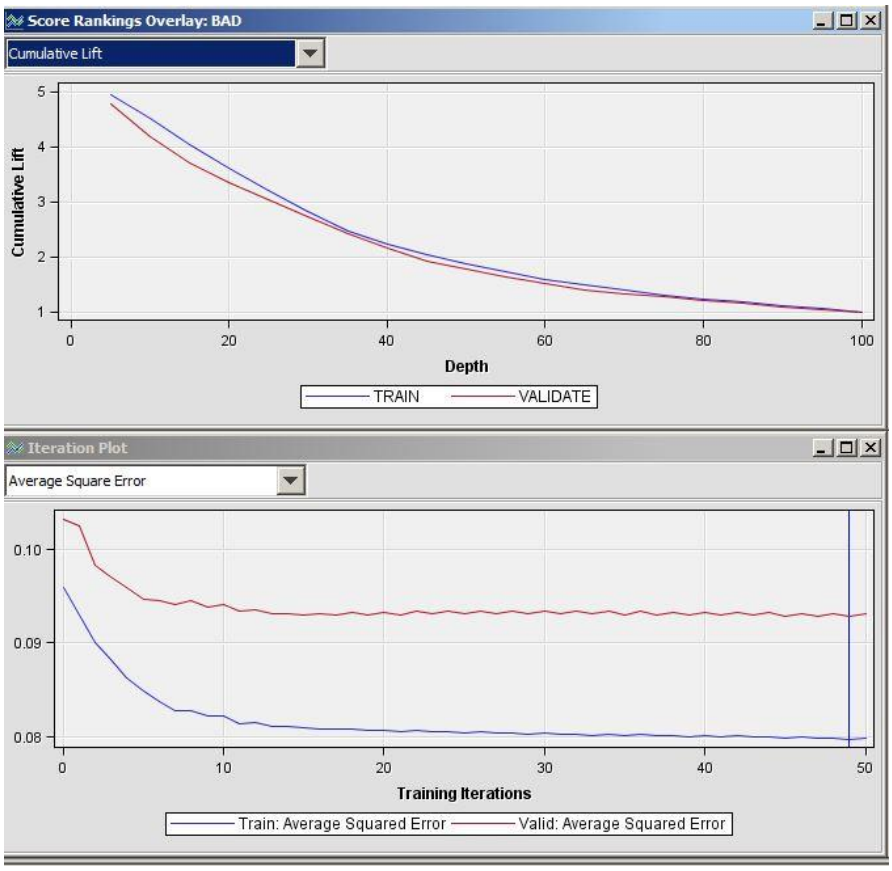
Model Node	Model Description	Data Role	Target	Target Label	False Negative	True Negative	False Positive	True Positive
Neural4	Neural Network (4)_1	TRAIN	BAD		399	2313	72	195
Neural4	Neural Network (4)_1	VALIDATE	BAD		409	2289	97	186
Neural5	Neural Network (5)_3	TRAIN	BAD		327	2313	72	267
Neural5	Neural Network (5)_3	VALIDATE	BAD		321	2298	88	274
Neural6	Neural Network (6)_5	TRAIN	BAD		208	2271	114	386
Neural6	Neural Network (6)_5	VALIDATE	BAD		216	2230	156	379
Neural	Neural Network (1)_1	TRAIN	BAD		495	2323	62	99
Neural	Neural Network (1)_1	VALIDATE	BAD		467	2311	75	128
Neural2	Neural Network (2)_3	TRAIN	BAD		371	2275	110	223
Neural2	Neural Network (2)_3	VALIDATE	BAD		335	2234	152	260
Neural3	Neural Network (3)_5	TRAIN	BAD		389	2310	75	205
Neural3	Neural Network (3)_5	VALIDATE	BAD		367	2281	105	228

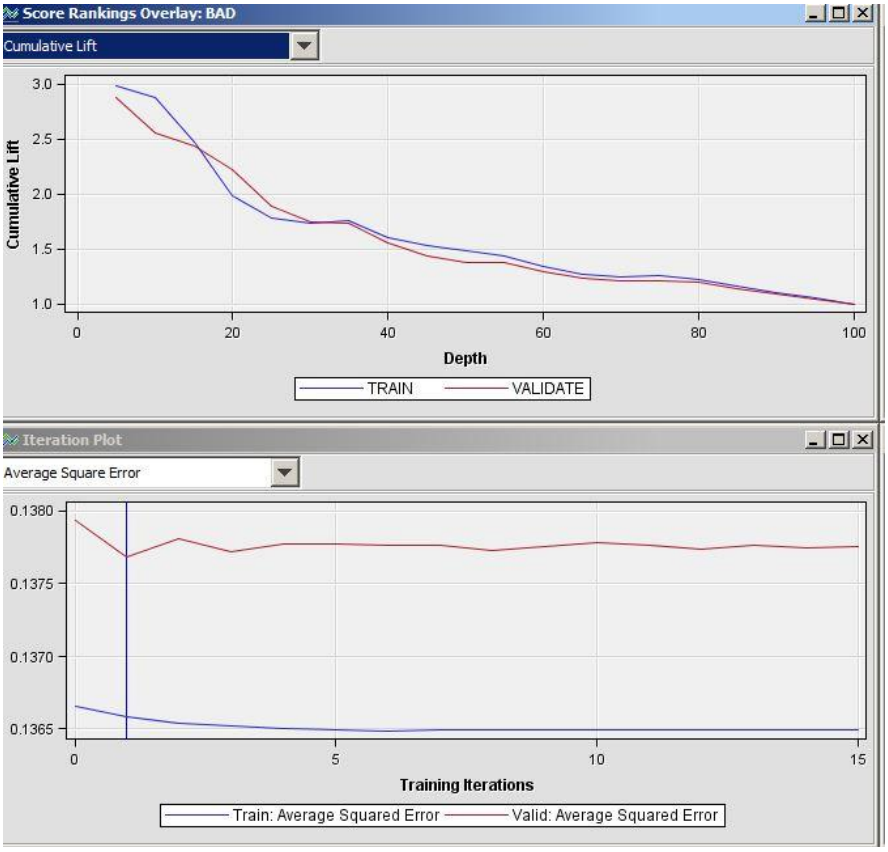
C. Lift Charts and ROC Chart

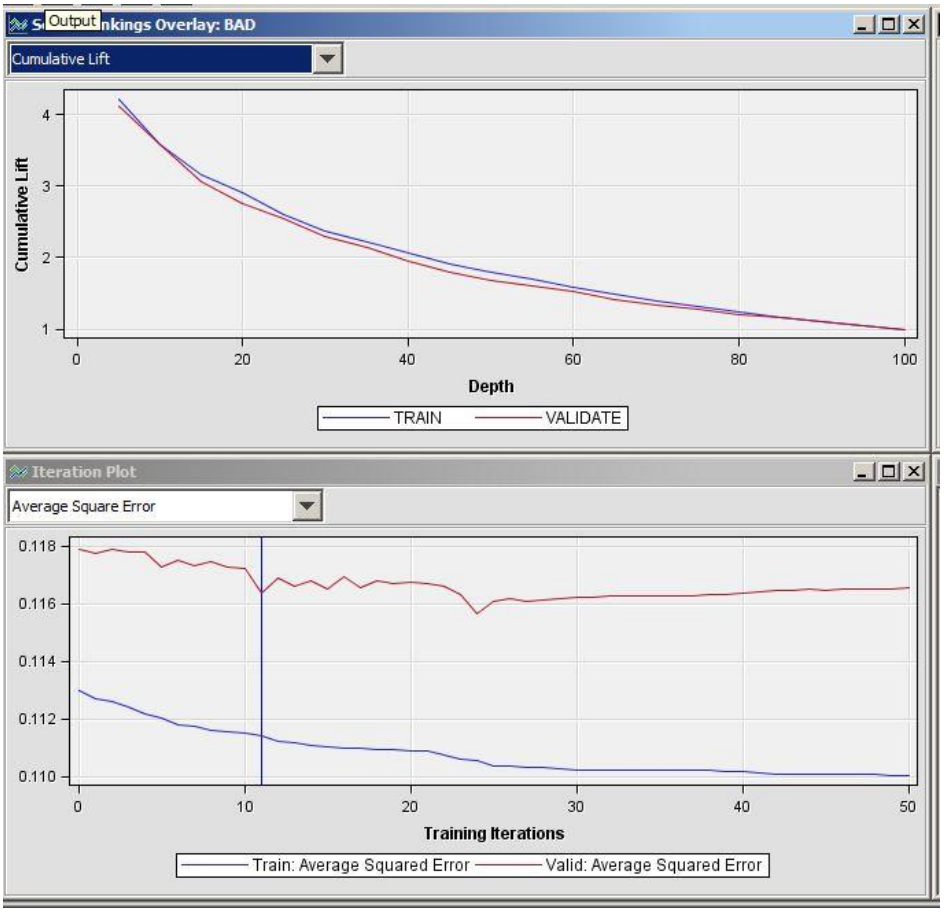


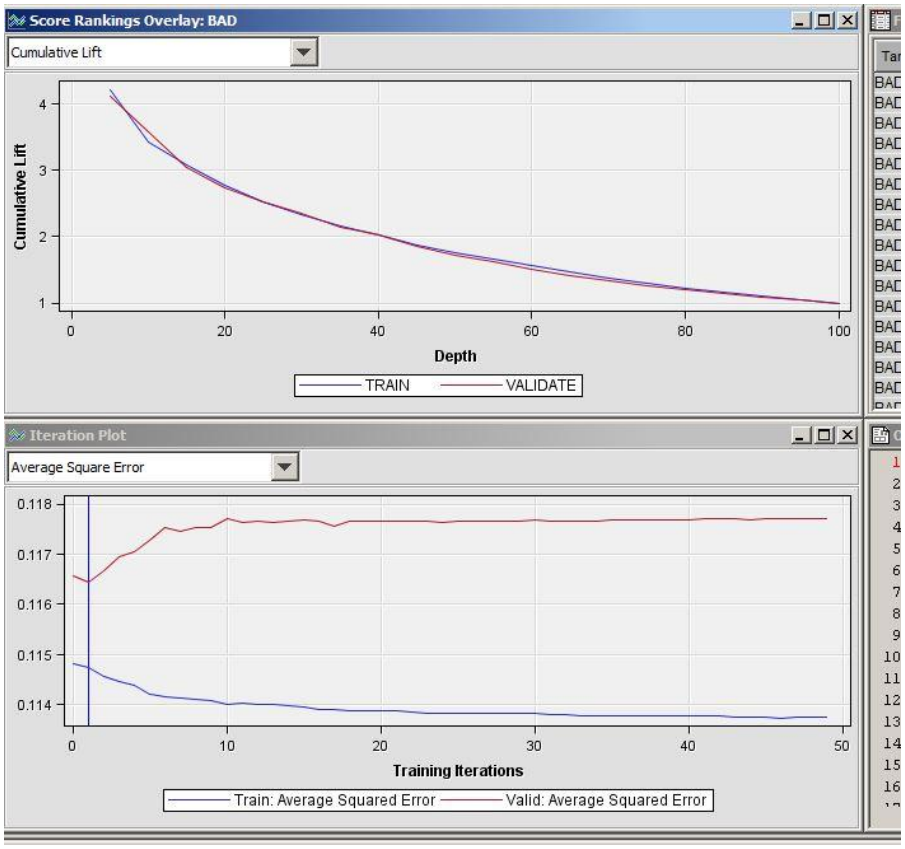

Score Rankings Overlay: BAD
Cumulative Lift
Cumulative Lift
5
4
3
2
1
0
20
40
60
80
100
Depth
TRAIN
VALIDATE
Iteration Plot
Average Square Error
0.115
0.110
0.105
0.100
0
10
20
30
40
50
Training Iterations
Train: Average Squared Error
Valid: Average Squared Error


Score Rankings Overlay: BAD
Cumulative Lift
Cumulative Lift
4
3
2
1
0
20
40
60
80
100
Depth
TRAIN
VALIDATE
Iteration Plot
Average Square Error
0.128
0.126
0.124
0.122
0.120
0
2
4
6
8
10
Training Iterations
Train: Average Squared Error
Valid: Average Squared Error


Score Rankings Overlay: BAD
Cumulative Lift
Cumulative Lift
5
4
3
2
1
0
20
40
60
80
100
Depth
TRAIN
VALIDATE
Iteration Plot
Average Square Error
0.10
0.09
0.08
0
10
20
30
40
50
Training Iterations
Train: Average Squared Error
Valid: Average Squared Error


Score Rankings Overlay: BAD
Cumulative Lift
Cumulative Lift
3.0
2.5
2.0
1.5
1.0
0
20
40
60
80
100
Depth
TRAIN
VALIDATE
Iteration Plot
Average Square Error
0.1380
0.1375
0.1370
0.1365
0
5
10
15
Training Iterations
Train: Average Squared Error
Valid: Average Squared Error


Output
nkings Overlay: BAD
Cumulative Lift
Cumulative Lift
Depth
TRAIN
VALIDATE
Iteration Plot
Average Square Error
Training Iterations
Train: Average Squared Error
Valid: Average Squared Error


Score Rankings Overlay: BAD
Cumulative Lift
Cumulative Lift
Depth
TRAIN
VALIDATE
Iteration Plot
Average Square Error
Training Iterations
Train: Average Squared Error
Valid: Average Squared Error