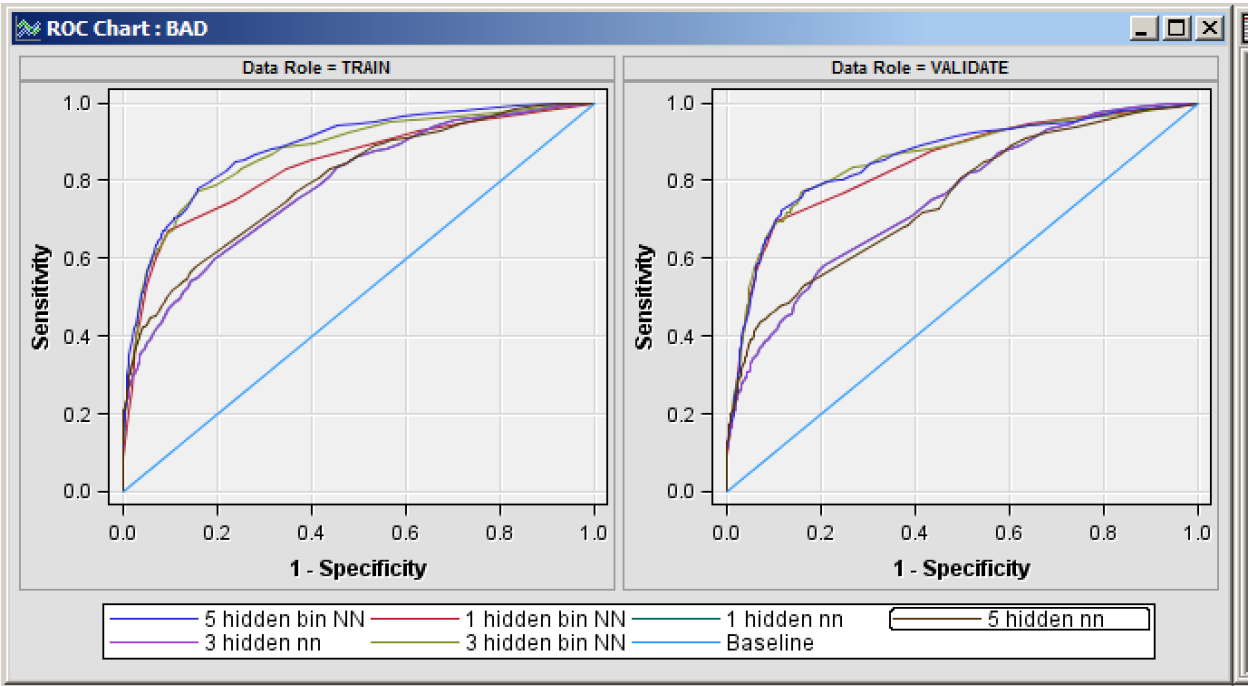
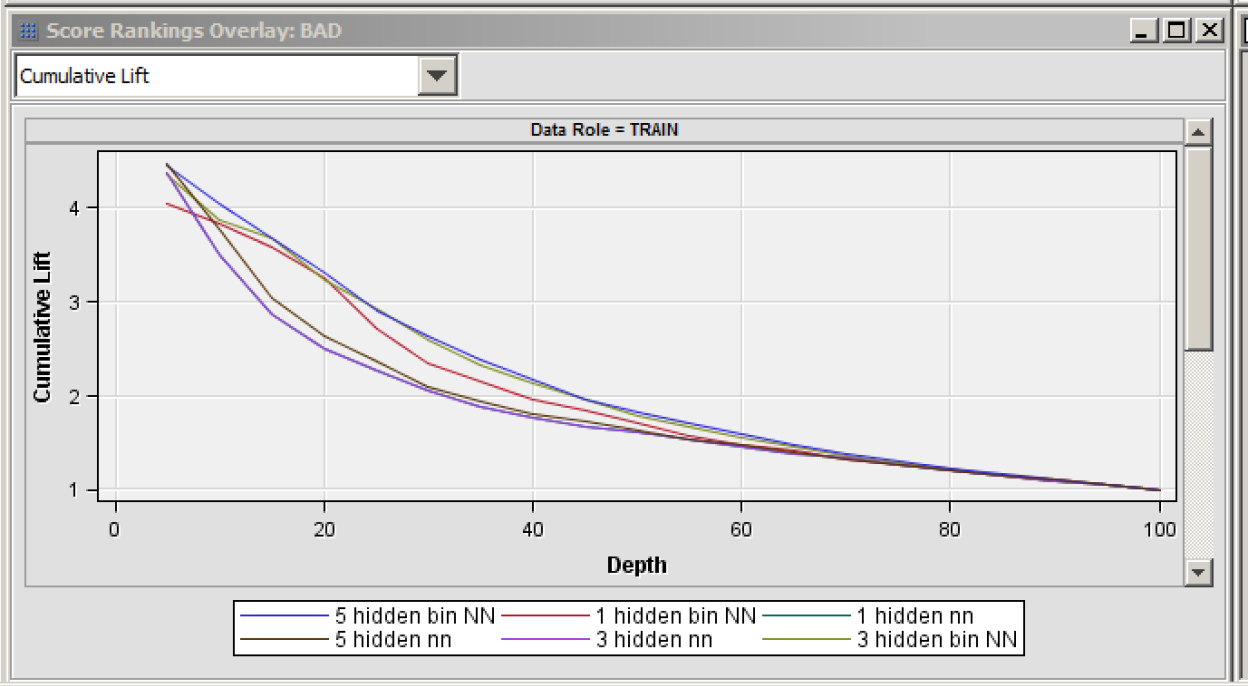
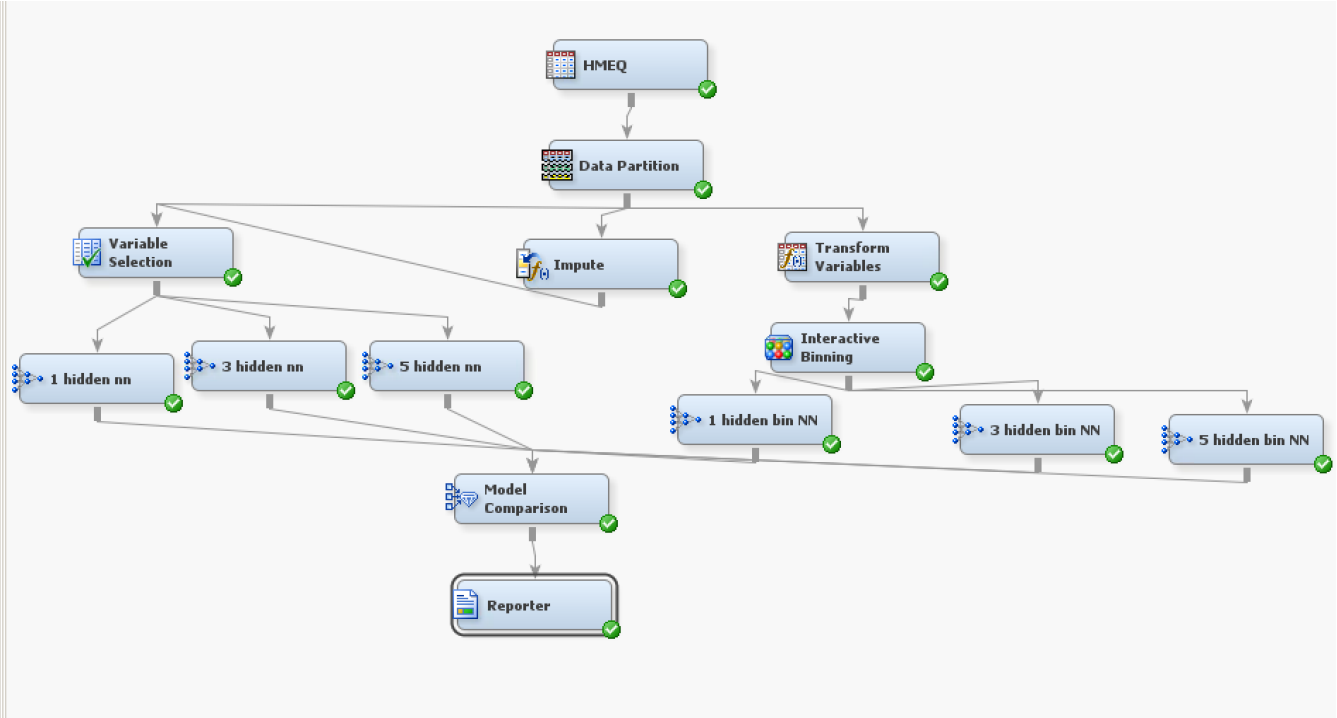

ROC Chart : BAD
Data Role = TRAIN
Data Role = VALIDATE
Sensitivity
1 - Specificity
0.0
0.2
0.4
0.6
0.8
1.0
5 hidden bin NN
1 hidden bin NN
1 hidden nn
5 hidden nn
3 hidden nn
3 hidden bin NN
Baseline


Score Rankings Overlay: BAD
Cumulative Lift
Data Role = TRAIN
Cumulative Lift
Depth
1
2
3
4
0
20
40
60
80
100
5 hidden bin NN
1 hidden bin NN
1 hidden nn
5 hidden nn
3 hidden nn
3 hidden bin NN



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
Neural2	3 hidden nn	TRAIN	BAD		413	2318	67	181
Neural2	3 hidden nn	VALIDATE	BAD		419	2285	101	176
Neural	3 hidden bin NN	TRAIN	BAD		253	2254	131	341
Neural	3 hidden bin NN	VALIDATE	BAD		249	2236	150	346
Neural3	5 hidden nn	TRAIN	BAD		394	2327	58	200
Neural3	5 hidden nn	VALIDATE	BAD		401	2300	86	194
Neural4	1 hidden nn	TRAIN	BAD		413	2318	67	181
Neural4	1 hidden nn	VALIDATE	BAD		419	2285	101	176
Neural5	1 hidden bin NN	TRAIN	BAD		262	2247	138	332
Neural5	1 hidden bin NN	VALIDATE	BAD		260	2232	154	335
Neural6	5 hidden bin NN	TRAIN	BAD		245	2246	139	349
Neural6	5 hidden bin NN	VALIDATE	BAD		247	2221	165	348

--
* Score Output
--

--
* Report Output
--

Damon Quire

CIS 445

Project 3

Basically this diagram uses the nodes to replace missing values first from the data set because Neural Networks pretty much hate missing values. Then when the missing values have been adjusted, and everything is ready for testing, the diagram splits into two different paths, one that transforms the variables and uses binning to smooth the variables, and then the other that simply goes straight into doing the neural network modeling. Each path has a neural network that contains 1 neuron in the hidden layer, 3 neurons in the hidden layer, and 5 neurons in the hidden layer. By doing this, we can really see which way of going about manipulating the given data set and which type of neural network best suits this certain data set for making the correct outcome guess.

The confusion matrices show the number of false positive, false negative, true positive, and true negative outcomes. This is showing how each model is accurately, or inaccurately assigning the outcome. By looking at this companies would be able to get a general idea for which model to go with without needing charts, by just comparing the numbers. They want high true occurrences, low false negatives and VERY low false positives. In cases like this one where they're determining whether or not to grant someone a loan, a false negative isn't sought after, but it isn't risky because they simply didn't give that person a loan. HOWEVER, when it grants a false negative that means that a loan company would lend money to someone who is high risk for not being able to pay off said loan. This creates very high risk loans that would typically be accompanied by a very large interest rate to account for the risk they're taking on, however if given a false positive, they wouldn't assign a higher interest rate or deny the loan in general because the model told them to grant the loan. This is very risky and could lead to a lot of money lost down the road.

For the ROC and Lift charts, the easiest way I've heard you explain it is you're looking for the largest area under the curve to show the best model. These models each have 6 curves that are assigned to each of the neural network nodes, by looking at these curves, a person can establish which of the models is the best way of modeling this data to most accurately guess the outcome. This adds a visual layer to deciding the best model instead of having to stud the numbers themselves, they are plotted to give you and easier understanding of exactly which model would be the prime model to choose.

While browsing through the two different charts that I have attached to this file, it is easy to tell that the neural networks with 5 neurons in the hidden layer seem to preform really well on both of the paths of the work flow. However, it is clear that the one with 5 neurons in the hidden layer that also goes through variable transformation and interactive binning is the best model to suit this data set(the dark blue curve). That means, it accurately predicts the outcome of the data set more frequently than the other models, making it the least risky model to use. By using this

model, a loan company wouldn't have to worry as much about correctly classifying a specific case as a good low risk loan or a loan to deny. So clearly it is helpful in this case at least to transform the variables due to the fact that it makes the model a better fit for the data that we're given.

Interpreting weights for me is still a little fuzzy to be honest. For what I can tell it seems that weights mean that if you have a factor that heavily effects the outcome, it is weighted higher to give it higher priority in the decision making process. These weights we then adjust until we are given the most accurate predictions possible. By doing this, the factors in determining the outcome are not all the same as far as how much they can effect the decision or not. For instance, income is a heavily weighted factor in a loan decision scenario, whereas, city of residence probably doesn't play nearly as big of a role in determining the predicted outcome. This is why we adjusted the weights in our last homework until we had no error, this steadies the factors involved in making the decisions and doesn't allow any of the determining factors to overrule their own magnitude to cause an error in the prediction and it also doesn't allow for factors to be under accounted for either. Hopefully I have a decent grasp of the weight concept and am on the right track.

My chosen model doesn't necessarily have the least false positives like I discussed earlier that are very risky, it has about 80 false positives which is close to the lowest among the models but isn't the best, however when looking at everything else as a whole, it has very numerous true outcomes and not a ton of false negatives. So overall when looking at the confusion matrix I would say it supports the hypothesis that I have stated while looking at the graphs that the model with five neurons in the hidden layer that goes through variable transformation is the best model when it comes to accurately predicting the outcome of the given data set.

Overall I think this project did a good job of showing how manipulating certain paths or the nodes themselves can drastically change the performance of the model itself. The model doesn't do everything correctly every time without some human intervention or trial and error. By giving us an opportunity to do this decision making ourselves and choosing paths ourselves, it helps our understanding more than following an exact set of instructions like the tutorials. Both are effective in their own right but I think this project helped me learn more about the models themselves and how to manipulate them themselves. I just wonder if this hypothesis would remain constant among other data sets or if this was just the best practice for this set specifically. I would think the same node would be the best model with all similar decision processes, but with ones that heavily differ, or have a vastly larger number of factors that go into the decision process would this result remain somewhat constant?

PS SAS ran the slowest it has ever ran for me. In fact the first 3-4 times I tried opening it it said "could not find the URL that identifies the SAS environment". Then when it finally worked it took a solid 5-10 minutes to run my full flow.