2017

Project 2

KIMBERLY ROETEN

CIS 445-01: DATA MINING

Confusion matrices in this flow process diagram that was built for this project exist for all of the model nodes, which excludes the SAS Code node in this version because it was not required for the project. But unfortunately, since the SAS Code node is not included, a confusion matrix cannot be made using this node to see the distribution of both the flow that uses a “cutoff node” to change the probability cutoff and the other flow that drops the cutoff node and keeps probability cutoff as 0.5. But each of the 3 model nodes generated in their own ways the probability that a customer was a widget buyer or not. The confusion matrices in this case are classified as the event classification tables in the output results, where the operating cut-off points affect the correct and incorrect classification rates and the number of false positive and false negative errors. So in other words, the confusion matrices show the number of correct and incorrect predictions made by the classification model compared to the actual outcomes, in this case target values, in the data given. What all 3 confusion matrices share in common is that they display matrices for 2 main classes, positive and negative. The positive value is the proportion of positive cases that were correctly identified, in contrast to the negative value that includes the proportion of negative cases that were correctly identified.

So starting off with the decision tree confusion matrix, the data role was train in this case and the false negative proportion was the only value that did not produce any results. According to the data, there were more positive cases that were correctly identified than there were negative cases.

False Negative	True Negative	False Positive	True Positive
0	6	3	11

Second comes the neural network confusion matrix, the data role was also classified as train and no false positive results were produced in this classification table. Instead, true negative went up by 3 and true positive stayed the same.

False Negative	True Negative	False Positive	True Positive
0	9	0	11

Lastly is where the regression confusion matrix comes in, and as you can see, the regression model produced the same results as the neural network model. This is because the neural network and logistic regression perform equally well for all cut-off points, so therefore on the model comparison chart the two curves are overlapping themselves, the only difference is they have a few different data points that are used to construct them. The data role was train in this case, just like the previous two models, so from here I think it is safe to conclude that the decision tree confusion matrix produced the best results, which will be further discussed.

False Negative	True Negative	False Positive	True Positive
0	9	0	11

In terms of the ROC and lift charts of all three models, they measure the effectiveness of the classification models calculated as the ratios between the results obtained with and without the models. So in contrast to confusion matrices which evaluate models on the whole population, these charts evaluate model performance in portions of the population. They show how much more likely we are to receive positive responses than if we were to contact a random sample of responses from the population as a whole.

So according to the lift charts, by contacting anywhere from about 5-10% of the widget buyers based on the predictive model, we will reach close to 2 times as many respondents, as if we use no model.

The rules generated by decision tree show that there are more cases where income is high or missing rather than low. And those with a high/missing income that are younger than 30.5 years of age are less than those who are at least 30.5 years of age. Out of those that are younger than 30.5, 60% of them are widget buyers and the other 40% of them are not. Out of those that are at least 30.5 years of age, 100% of them are non-widget buyers. Out of those whose income is low, about 89% of them are widget buyers and the other 11% are not. Splitting rules occurred more frequently with those who have high or missing income because they produced more results, so it made more sense to elaborate.

Importance of variables in these results are calculated using the decision tree methodology, where it evaluates the overall value or importance in this case of the variables over the fitted tree. They are impacted by both the number of observations and the purity of the splitting results, and in this case, the variable income is the purest of them all because of its 1.000 entropy. The other variables including x2, x4, x5 and residence are the most impure because of their 0.000 entropies.

Effects generated for logistic regression coefficients are no doubt the way they resulted because all values for regression were left as default. The positive value predictions have more of an effect on the absolute coefficient, and that is not surprising at all if you take into consideration the confusion matrices.

After thoroughly examining each predictive model used for this project, the variable that has the most predictive power is income, because it was rated the most important, therefore age was the second important, this is why the decision tree split up the variables how they were.

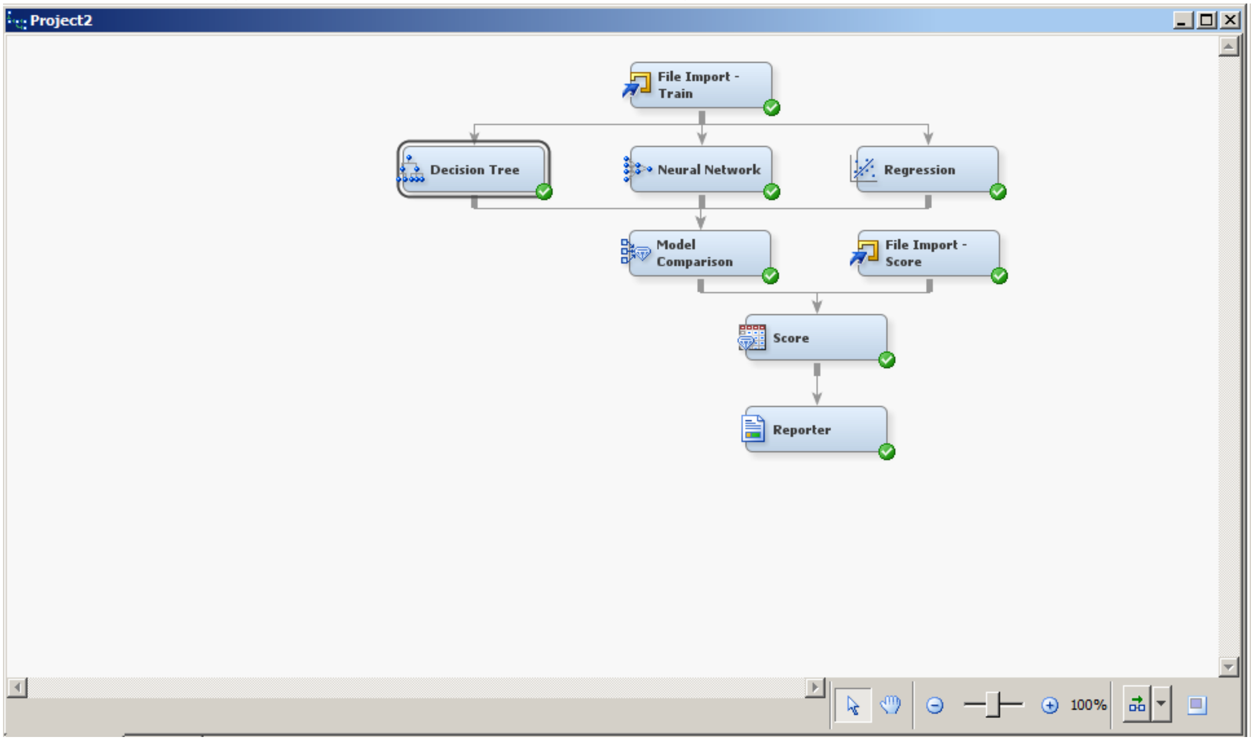
The logistic regression model however did not identify the same variables as the decision tree in terms of their predictive power, and I think it was due to what they look for. For instance, a logistic regression model is searching for a single linear decision based on features you have.

And a decision tree essentially partitions into half spaces so to speak, so it then because more of a non-linear decision boundary.

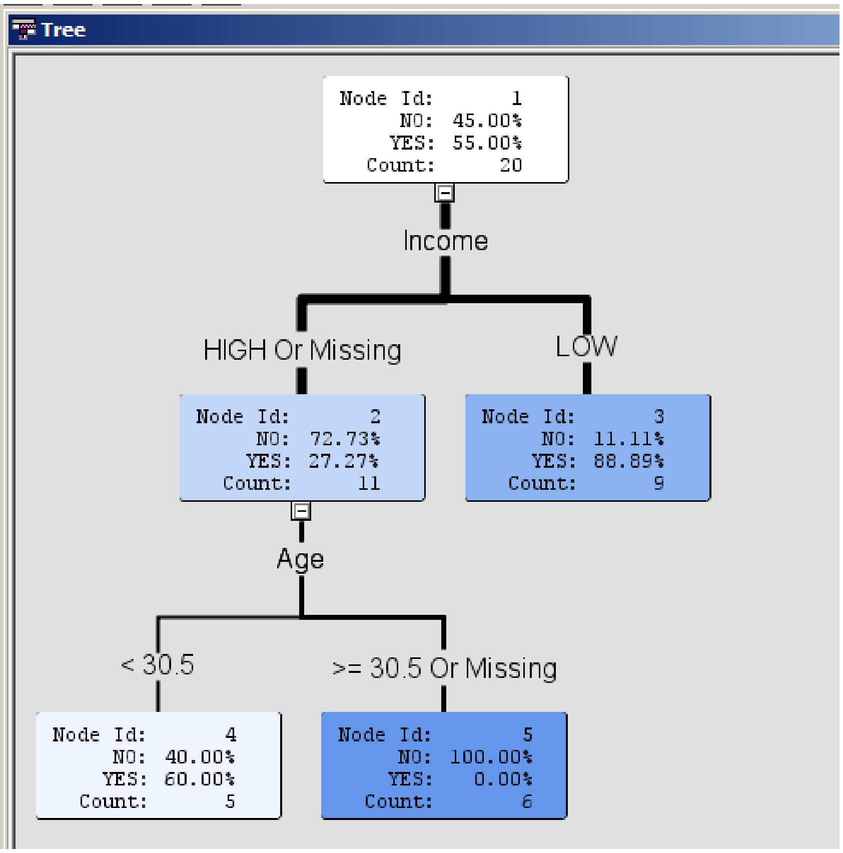
After thoroughly examining the final weights of the neural network, it is safe to conclude that according to the neural network residence in terms of chi-square was ranked the variable with the most weight, which makes sense considering it was a variable that was not ranked at all in the decision tree model.

Out of the nine cases from the WidgBuyScore.xlsx data set, around 70% of the results, which is about 6 cases were classified as non-widget buyers, while the other 3 (approximately 30%) were classified as widget buyers.

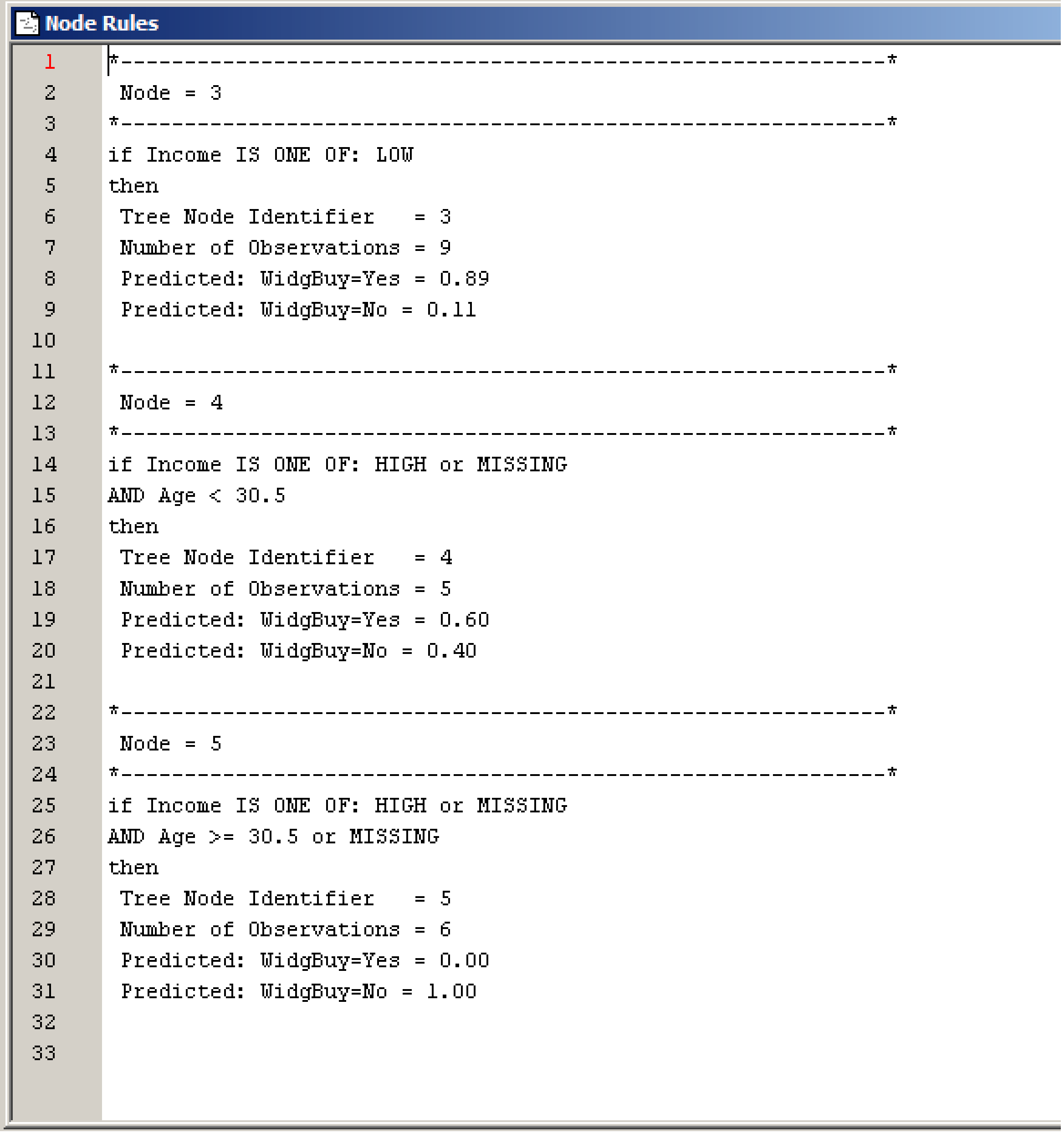
1a. Workflow/diagram with all nodes used (excluding the SAS Code)



1b. Tree diagram generated by decision tree which shows variables, branches and nodes

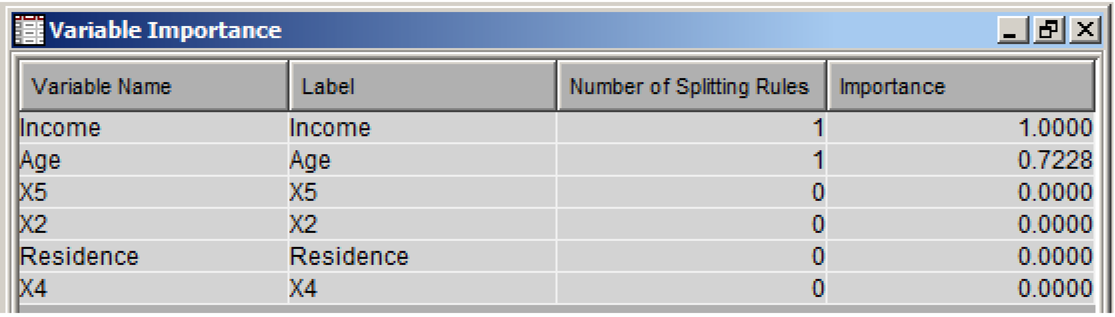


1c. Window with rules generated by decision tree



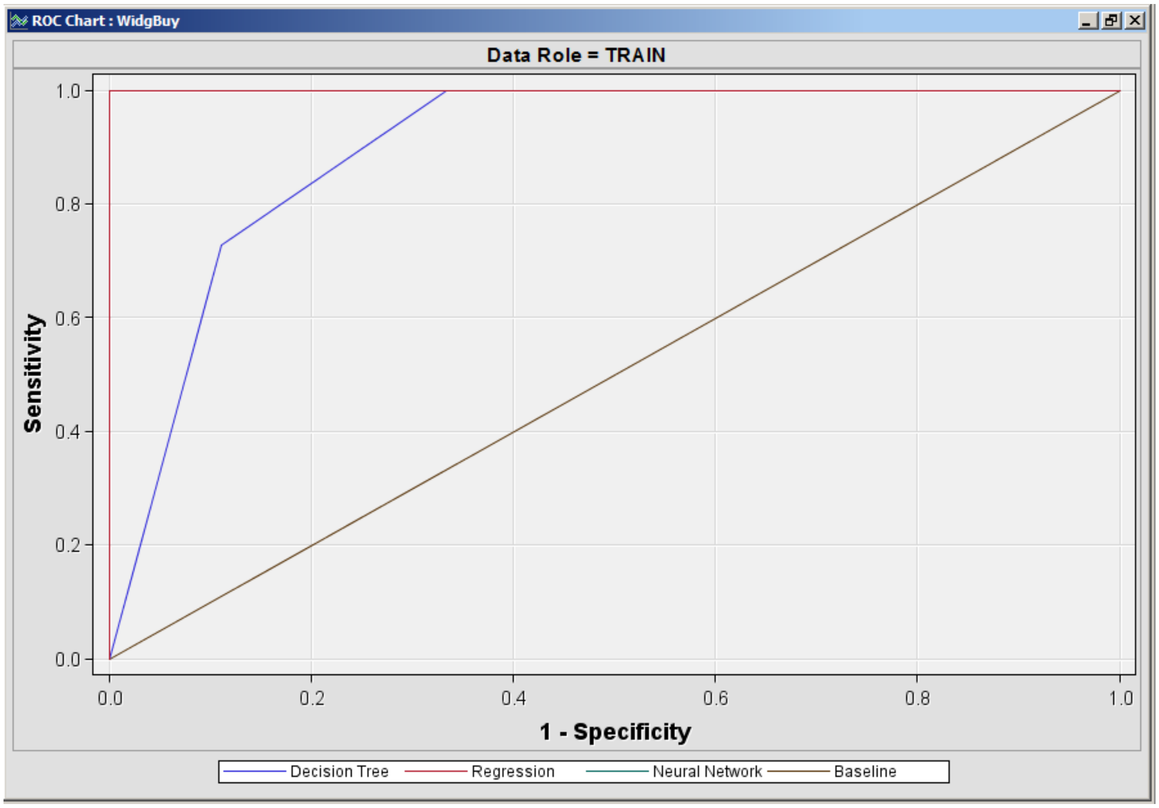
```
*------------------------------------------------------------*
 Node = 3
*------------------------------------------------------------*
if Income IS ONE OF: LOW
then
 Tree Node Identifier   = 3
 Number of Observations = 9
 Predicted: WidgBuy=Yes = 0.89
 Predicted: WidgBuy=No = 0.11

*------------------------------------------------------------*
 Node = 4
*------------------------------------------------------------*
if Income IS ONE OF: HIGH or MISSING
AND Age < 30.5
then
 Tree Node Identifier   = 4
 Number of Observations = 5
 Predicted: WidgBuy=Yes = 0.60
 Predicted: WidgBuy=No = 0.40

*------------------------------------------------------------*
 Node = 5
*------------------------------------------------------------*
if Income IS ONE OF: HIGH or MISSING
AND Age >= 30.5 or MISSING
then
 Tree Node Identifier   = 5
 Number of Observations = 6
 Predicted: WidgBuy=Yes = 0.00
 Predicted: WidgBuy=No = 1.00
```

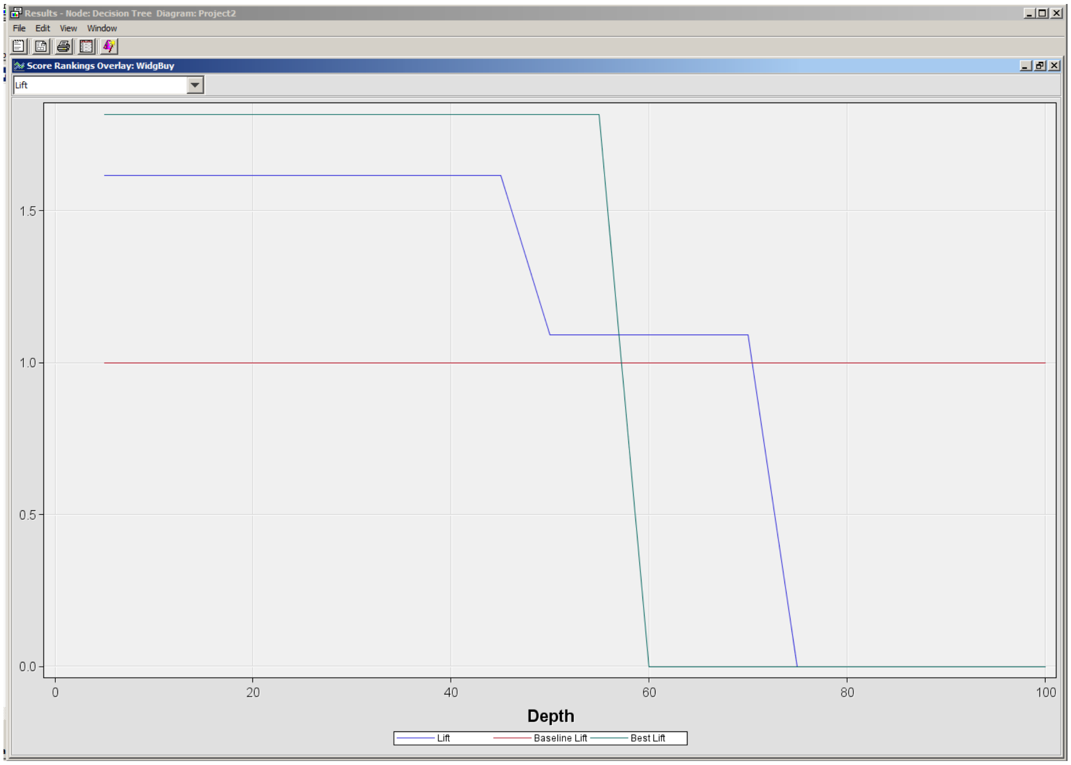
1d. Table with relative importance of variables used in decision tree

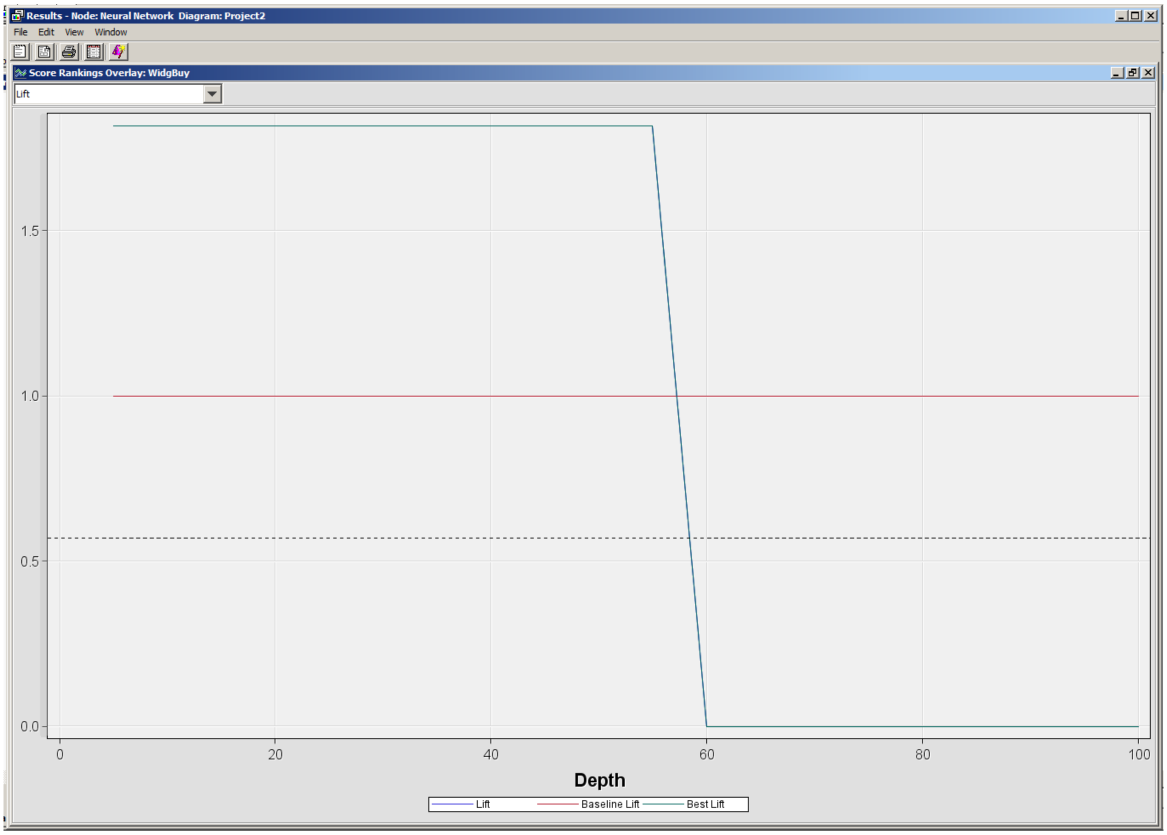


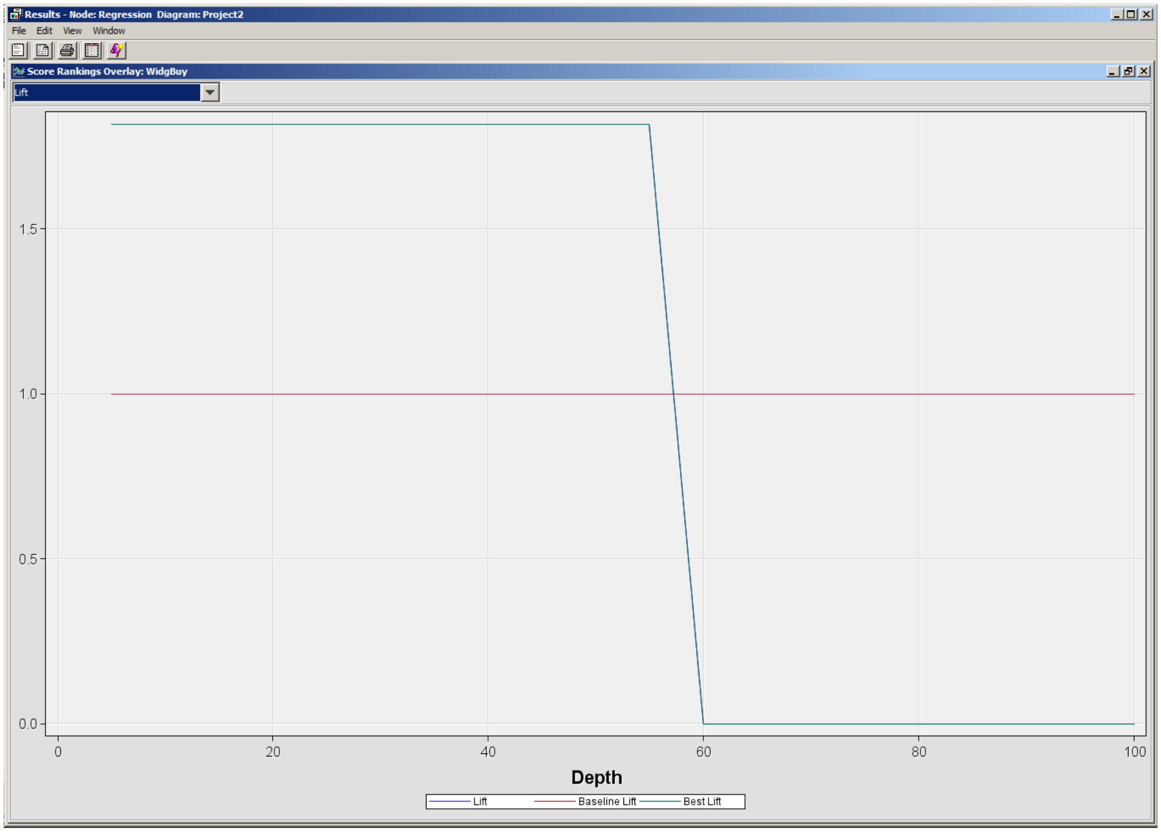
Variable Name	Label	Number of Splitting Rules	Importance
Income	Income	1	1.0000
Age	Age	1	0.7228
X5	X5	0	0.0000
X2	X2	0	0.0000
Residence	Residence	0	0.0000
X4	X4	0	0.0000

1e. ROC/Lift Charts for 3 models (Decision Tree, Neural Network, Regression respectively)

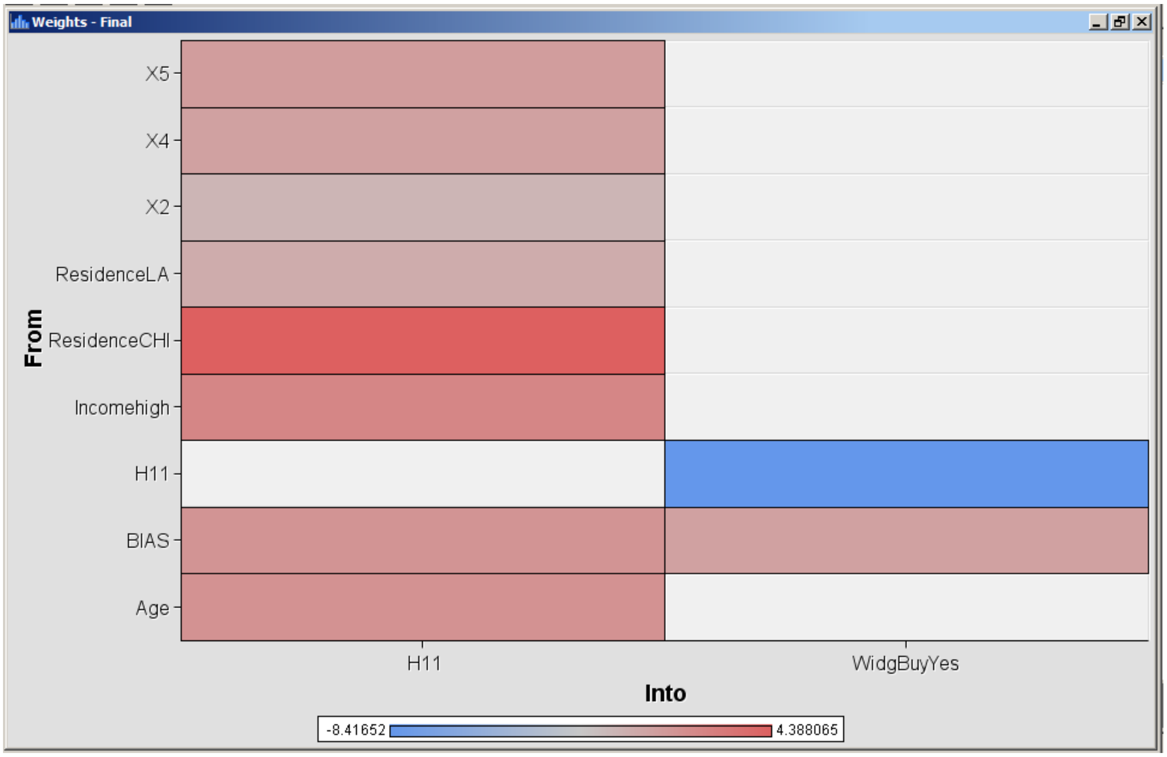




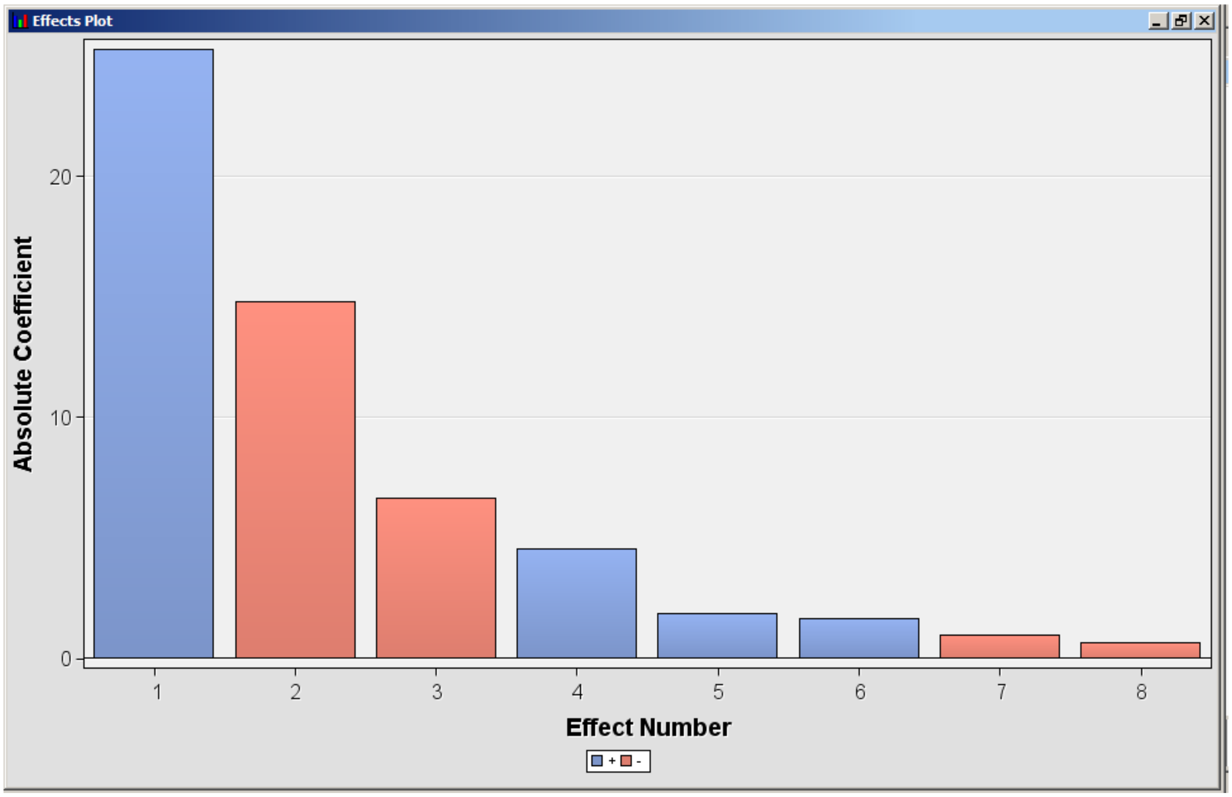

Results - Node: Neural Network Diagram: Project2
File
Edit
View
Window
Score Rankings Overlay: WidgBuy
Lift
1.5
1.0
0.5
0.0
0
20
40
60
80
100
Depth
Lift
Baseline Lift
Best Lift


Results - Node: Regression Diagram: Project2
File
Edit
View
Window
Score Rankings Overlay: WidgBuy
Lift
1.5
1.0
0.5
0.0
0
20
40
60
80
100
Depth
Lift
Baseline Lift
Best Lift

1f. Window with final weights for neural network



1g. Chart with effects for regression model


Effects Plot
Absolute Coefficient
Effect Number