Damon Quire

CIS 445

Project 4

In this project I have split the diagram into two separate paths, one being a path that uses the transform variables node along with the filter node to help account for outliers and make them unable to skew the results. The other doesn't use these two nodes, allowing outliers to effect the outcome of the model comparison node. By doing this, and assigning the same models to each oath we can tell which path benefits the data set the most. This allows us to possibly run further nodes or tests with the path and node that is the best to even more accurately predict the outcome variable or target variable of the data set. This way we save time (especially when using the virtual environment at UofL) by not having to run the same nodes for every single model and path. We instead can pinpoint pretty quickly which model and preceding nodes work best for a given data set.

As far as the graph is concerned, all models and paths are pretty hard to visually interpret which route was the best for this given data set. So, I turn my eye to the chart with the given errors. Here, to my surprise a node from the path that doesn't deal with outliers has the lowest combined squared error and mean squared error. Meaning it's predicted values varied less from the actual values than any other predicting model that I have given the diagram. This node was the Memory based reasoning node which I don't believe we've used before. This node on the other path was actually the second best model in terms of these squared error statistics. From how I understand it, this node relies on remembering similar cases that it was given and then using it's outcome or output to drive it's decision making. In easy to understand terms I see it as this: say you're looking at a statistical table for tickets given to drivers over certain speed limits in certain areas. If ten previous times, when a driver was doing 10 over the speed limit in a 35 MPH zone, they were given a ticket, the model would assume if given the same inputs, the output would remain consistent. This model seems like it could be VERY useful for a lot of different data sets because it's literally using actual examples from memory and applying the same outcome. They say the definition of insanity is doing something over and over again expecting a different result, this model is basically doing the opposite of that. It's assigning the outcome of other similar situation's outcomes, giving it the same result. If it's way of predicting outcomes is the exact opposite of the definition of insanity, it has to be good for something right?

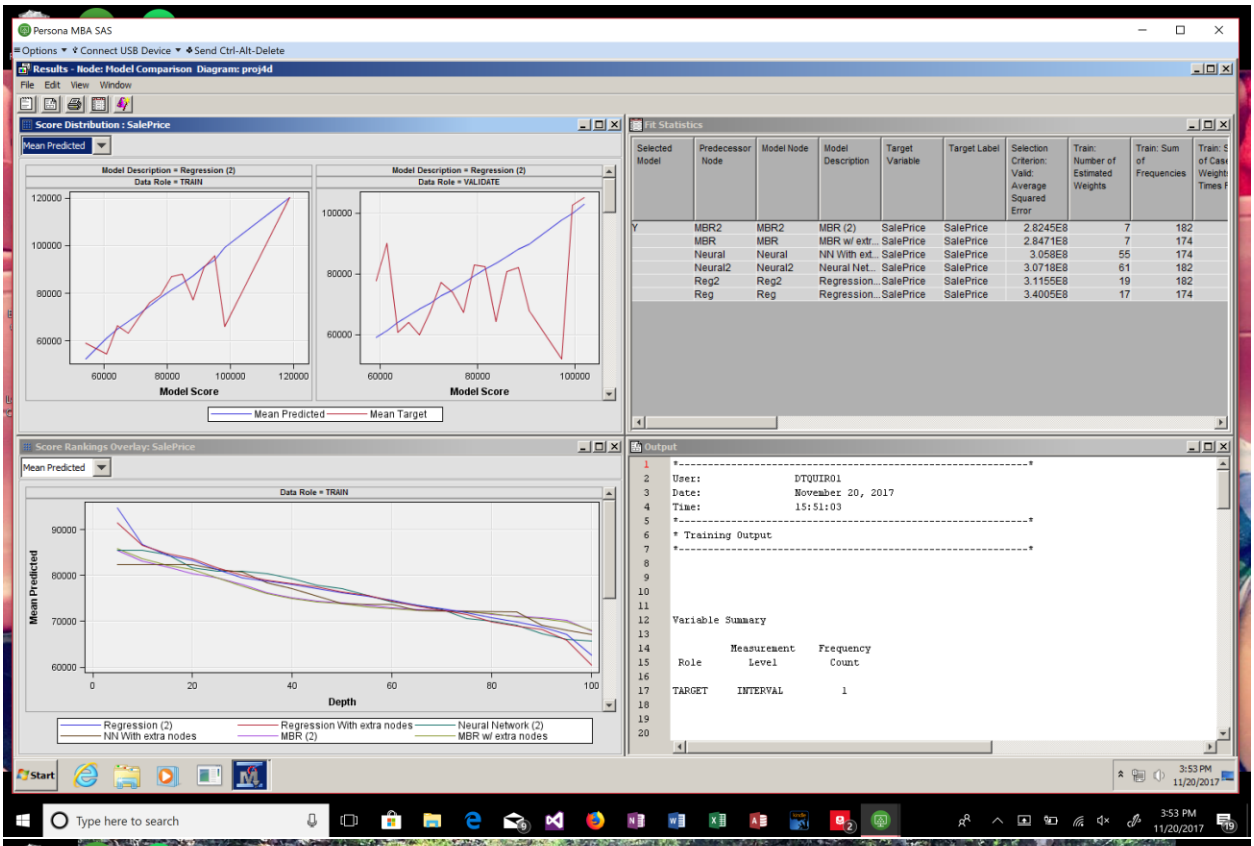
The next best models were the neural networks, the better being the one that did have the extra nodes to help eliminate the skewness of the results due to the outliers. To me, this means that MBR isn't affected as much by statistical outliers as neural networks is (maybe I'm wrong). This would make sense to me because neural networks would be heavily effected when you calculate "S" if they had a very large or small input. This would skew the results most definitely and make the weights hard to calculate if you have a few inputs that vastly differ from the mean or average of the given data set.

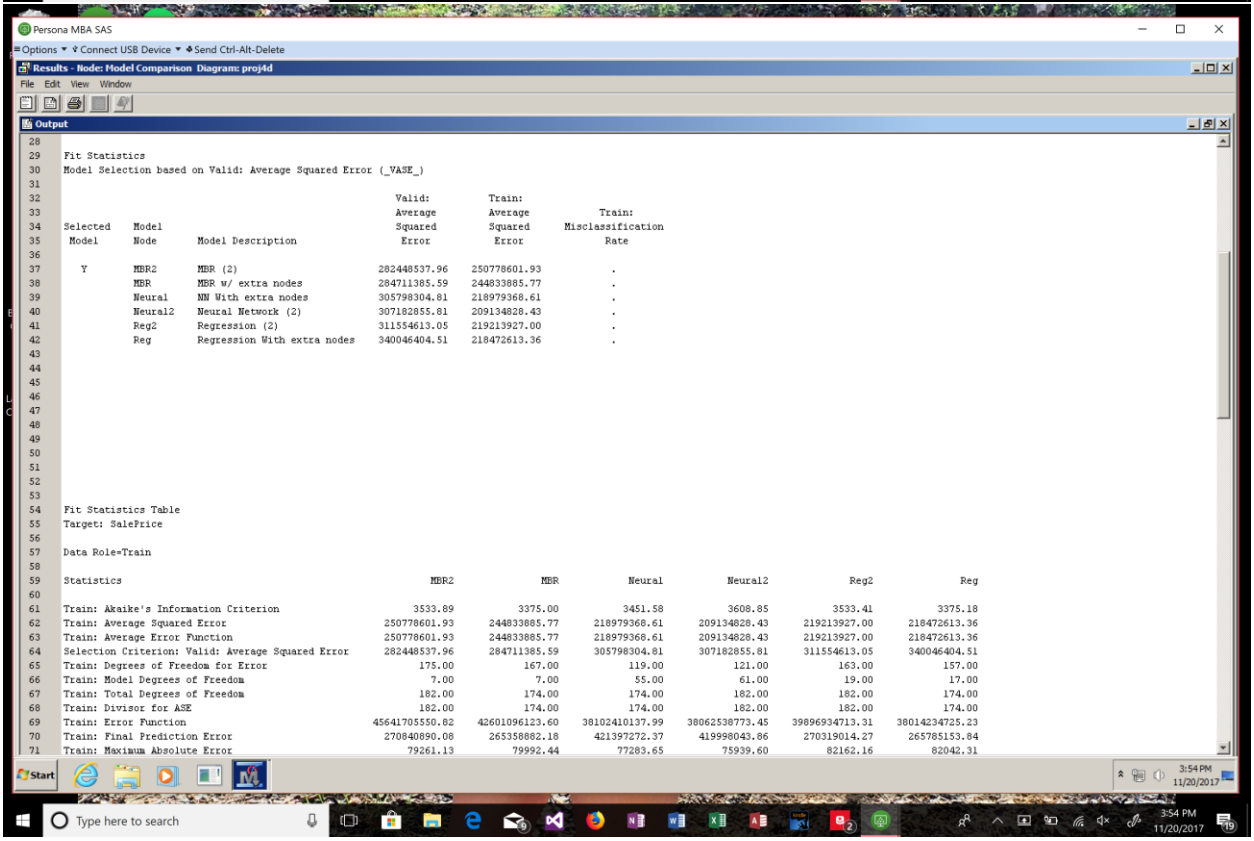
The next best models, in dead last were the regression models. Again, just like the MBR models, the one without the extra nodes to take care of outliers actually performed better. This is interesting to me because I would assume the opposite, maybe the way I accounted for the outliers could use some adjustment. However, it doesn't appear that the regression models would have performed well either way given they performed worse than any other model on any other path. I believe on a previous homework or project, the regression models prevailed as the best predictor for that given data set. It's interesting to see how different data sets are modeled vastly different from other data sets.

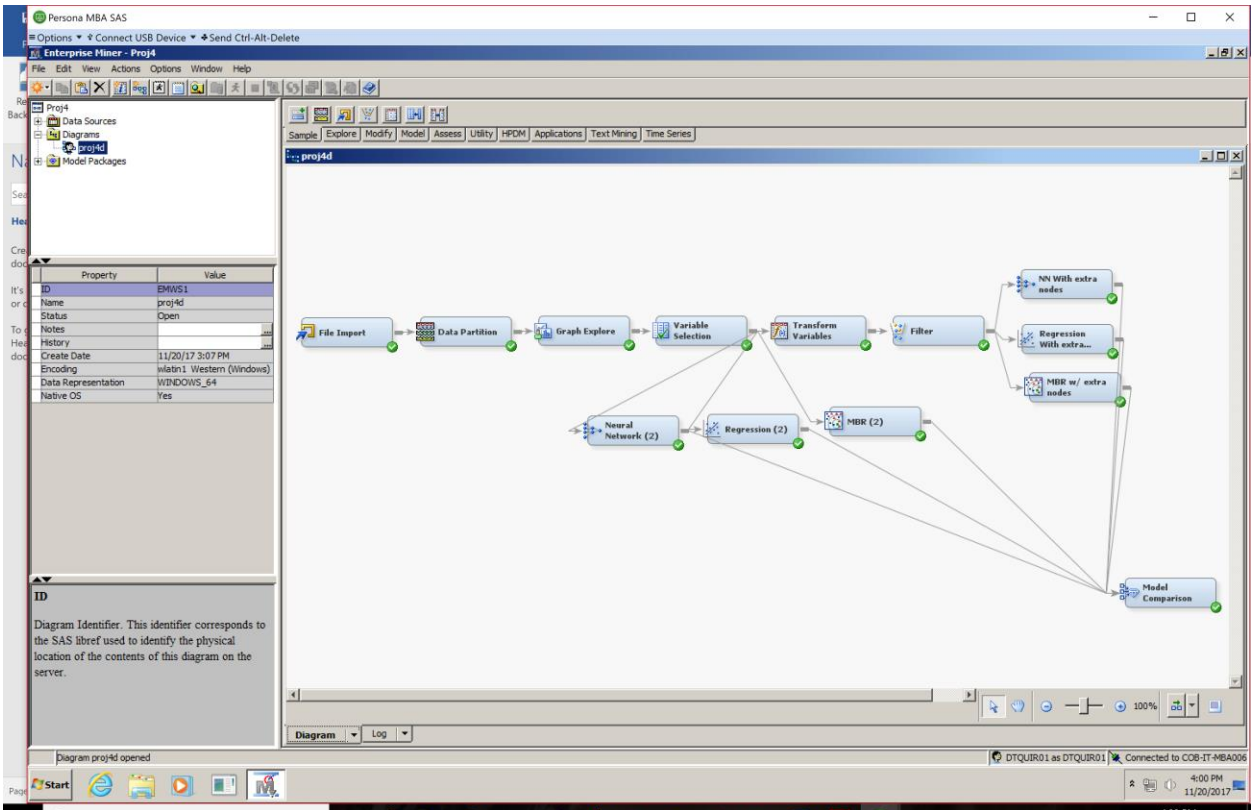
To me removing outliers always seems like a good idea in my head because it allows for a given sample to be closer to the overall population and disallows large variations to make a difference in the predicting of outcomes. However, I could see how in memory based reasoning, outliers could actually play as a benefit, by having past examples to remember and base your reasoning on, it could be beneficial to have an outlier thrown in there with it's actual outcome to help predict future outlier's outcomes. This way when it happens again (which it will), the model will have a basis for assigning the outcome because it's experienced it before. Whereas models that deal heavily with calculations more than patters, outliers very often skew the overall results. Which is why I think the neural networks, in my diagram did better when the outliers were dealt with prior to the modeling, and the MBR's did better when they were not dealt with. That's the only way it makes sense to me in my head.

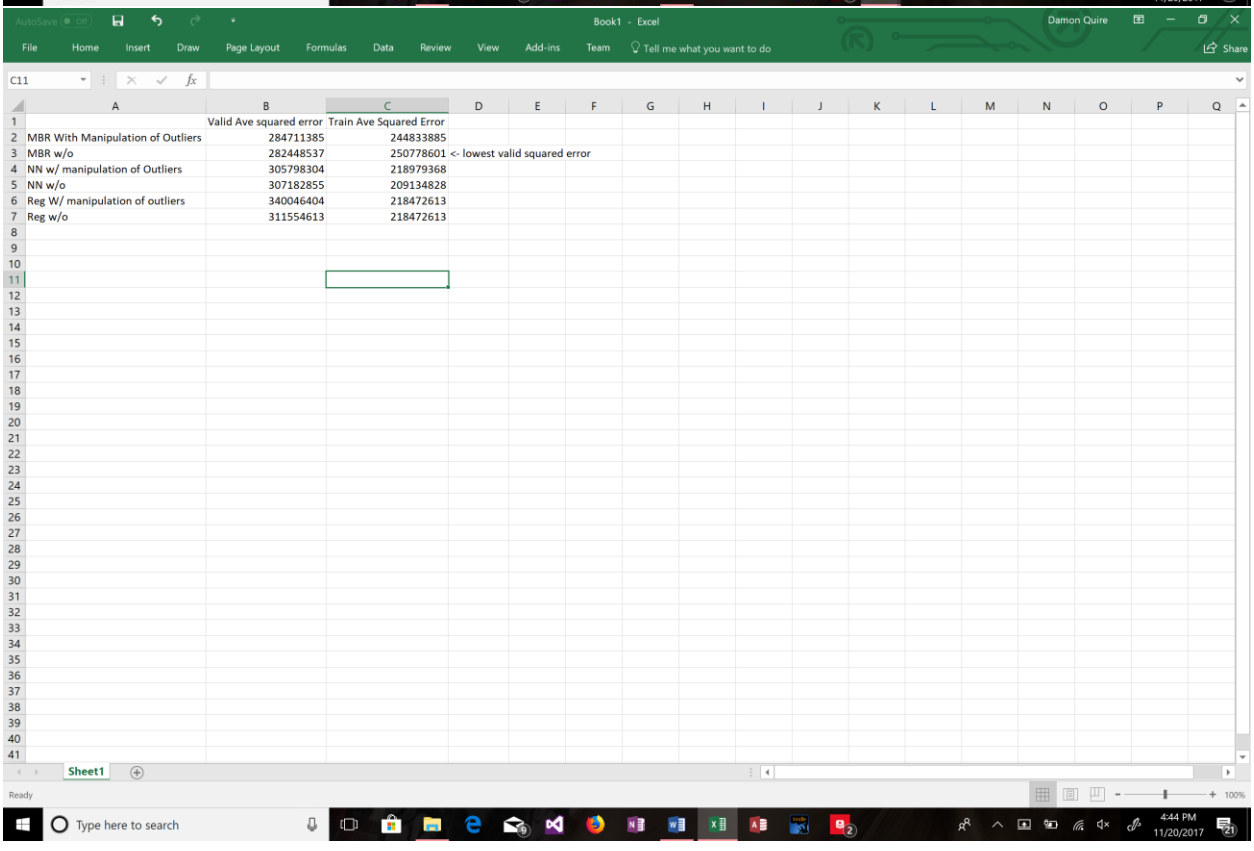
This assignment was somewhat difficult because I didn't have your instructions to hold onto as a backing for progress because a lot of the instructions were open ended and allowed us to handle the manipulation of the data set ourselves the way we see fit. This forces us to really know what we're doing and the order that the nodes need to be in (hopefully I know what I'm doing somewhat). This I think overall is a good thing, because I don't think I learned that much from doing the tutorials, but this time I was forced to click on properties that I didn't before just to get a description on what I'd be accomplishing by manipulating the property. I think I have benefited from this project more than the previous ones.

I don't 100% know what you mean for the creation of the table but I will throw in the statistics like the errors that I used to pick the best model. Obviously based on what I've mentioned in the report, the best model that I would pick to determine sale price would be the MBR model without the manipulation of outliers. This model provided the least overall error which in my mind is what I look for when determining something like price. Had it been something like a loan decision I wouldn't have used this to determine the best model because something like that isn't all about the error because errors like false positives are far more risky and dangerous that false negatives, so simply showing the total error would not tell the whole story on which model is doing the best job. That is when you could use the confusion matrices to show how many of each type of misclassifications were present in each model that was used in the diagram.


Persona MBA SAS
Options
Connect USB Device
Send Ctrl-Alt-Delete
Results - Node: Model Comparison Diagram: proj4d
File Edit View Window
Score Distribution : SalePrice
Mean Predicted
Model Description = Regression (2)
Data Role = TRAIN
Model Description = Regression (2)
Data Role = VALIDATE
Model Score
Mean Predicted
Mean Target
Fit Statistics
Selected Model
Predecessor Node
Model Node
Model Description
Target Variable
Target Label
Selection Criterion: Valid: Average Squared Error
Train: Number of Estimated Weights
Train: Sum of Frequencies
Y MBR2 MBR2 MBR (2) SalePrice SalePrice 2.8245E8 7 182
MBR MBR MBR w/ extr... SalePrice SalePrice 2.8471E8 7 174
Neural Neural NN With ext... SalePrice SalePrice 3.058E8 55 174
Neural2 Neural2 Neural Net... SalePrice SalePrice 3.0718E8 61 182
Reg2 Reg2 Regression... SalePrice SalePrice 3.1155E8 19 182
Reg Reg Regression... SalePrice SalePrice 3.4005E8 17 174
Score Rankings Overlay: SalePrice
Mean Predicted
Data Role = TRAIN
Mean Predicted
Depth
Regression (2)
NN With extra nodes
Regression With extra nodes
MBR (2)
Neural Network (2)
MBR w/ extra nodes
Output
* Training Output
User: DTQUIR01
Date: November 20, 2017
Time: 15:51:03
Variable Summary
Role Measurement Level Frequency Count
TARGET INTERVAL 1
Start
Type here to search
3:53 PM
11/20/2017


Persona MBA SAS
Options
Connect USB Device
Send Ctrl-Alt-Delete
Results - Node: Model Comparison Diagram: proj4d
File Edit View Window
Output
Fit Statistics
Model Selection based on Valid: Average Squared Error (_VASE_)
Selected Model
Model Node
Model Description
Valid: Average Squared Error
Train: Average Squared Error
Train: Misclassification Rate
Y MBR2 MBR (2) 282448537.96 250778601.93 .
MBR MBR w/ extra nodes 284711385.59 244833885.77 .
Neural NN With extra nodes 305798304.81 218979368.61 .
Neural2 Neural Network (2) 307182855.81 209134828.43 .
Reg2 Regression (2) 311554613.05 219213927.00 .
Reg Regression With extra nodes 340046404.51 218472613.36 .
Fit Statistics Table
Target: SalePrice
Data Role=Train
Statistics MBR2 MBR Neural Neural2 Reg2 Reg
Train: Akaike's Information Criterion 3533.89 3375.00 3451.58 3608.85 3533.41 3375.18
Train: Average Squared Error 250778601.93 244833885.77 218979368.61 209134828.43 219213927.00 218472613.36
Train: Average Error Function 250778601.93 244833885.77 218979368.61 209134828.43 219213927.00 218472613.36
Selection Criterion: Valid: Average Squared Error 282448537.96 284711385.59 305798304.81 307182855.81 311554613.05 340046404.51
Train: Degrees of Freedom for Error 175.00 167.00 119.00 121.00 163.00 157.00
Train: Model Degrees of Freedom 7.00 7.00 55.00 61.00 19.00 17.00
Train: Total Degrees of Freedom 182.00 174.00 174.00 182.00 182.00 174.00
Train: Divisor for ASE 182.00 174.00 174.00 182.00 182.00 174.00
Train: Error Function 45641705550.82 42601096123.60 38102410137.99 38062538773.45 39896934713.31 38014234725.23
Train: Final Prediction Error 270840890.08 265358882.18 421397272.37 419998043.86 270319014.27 265785153.84
Train: Maximum Absolute Error 79261.13 79992.44 77283.65 75939.60 82162.16 82042.31
Start
Type here to search
3:54 PM
11/20/2017





	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1		Valid Ave squared error	Train Ave Squared Error														
2	MBR With Manipulation of Outliers	284711385	244833885														
3	MBR w/o	282448537	250778601	<- lowest valid squared error													
4	NN w/ manipulation of Outliers	305798304	218979368														
5	NN w/o	307182855	209134828														
6	Reg W/ manipulation of outliers	340046404	218472613														
7	Reg w/o	311554613	218472613														