2017

Project 4

KIMBERLY ROETEN & EVAN WALSH

CIS 445-01: DATA MINING

For this project, the nodes that were used were a file import with the real estate data, a data partition node that only uses training and validation, as the data set is too small to use test data, a statexplore node to produce a statistical summary of the input data, then I used a filter node for the variable selection and transformation nodes leading up to interactive binning, and the 3 models we want in this diagram, neural network, linear regression, and memory base reasoning with 10 neighbors. These three models were also set up without the filtering nodes, to see the difference. So essentially there are two paths in this diagram, one that filters outliers and one that doesn't.

When calling to perform the data cleaning and variable transformation, it appears that the total area variable is likely to be a linear combination of the variables first floor, second floor, and upper area. Assuming the total area variable is a summation of these other variables, using all four in the model is not necessary, unless you want to find out the most significant variable leading up to that point, that makes the total area variable so significant.

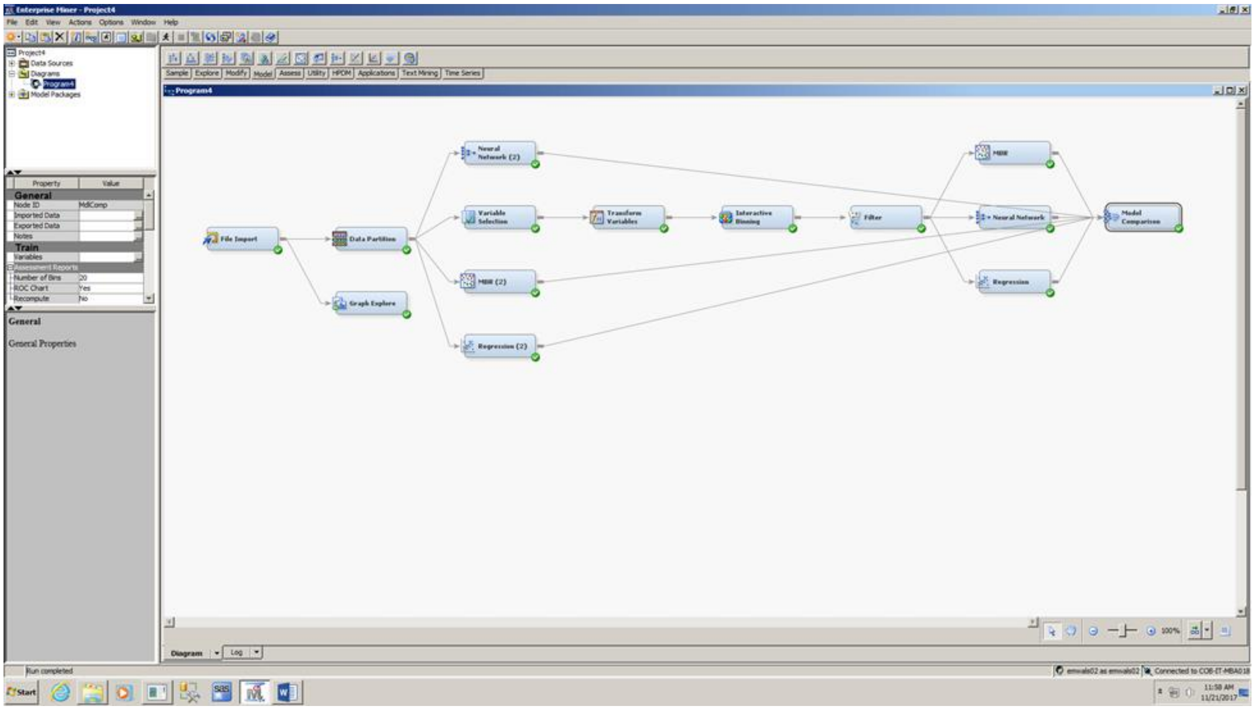
The most significant variable that is determined by the linear regression node not connected to the filter node includes garage type. The most significant variable that is determined by the linear regression node that is connected to the filtering nodes includes is total area.

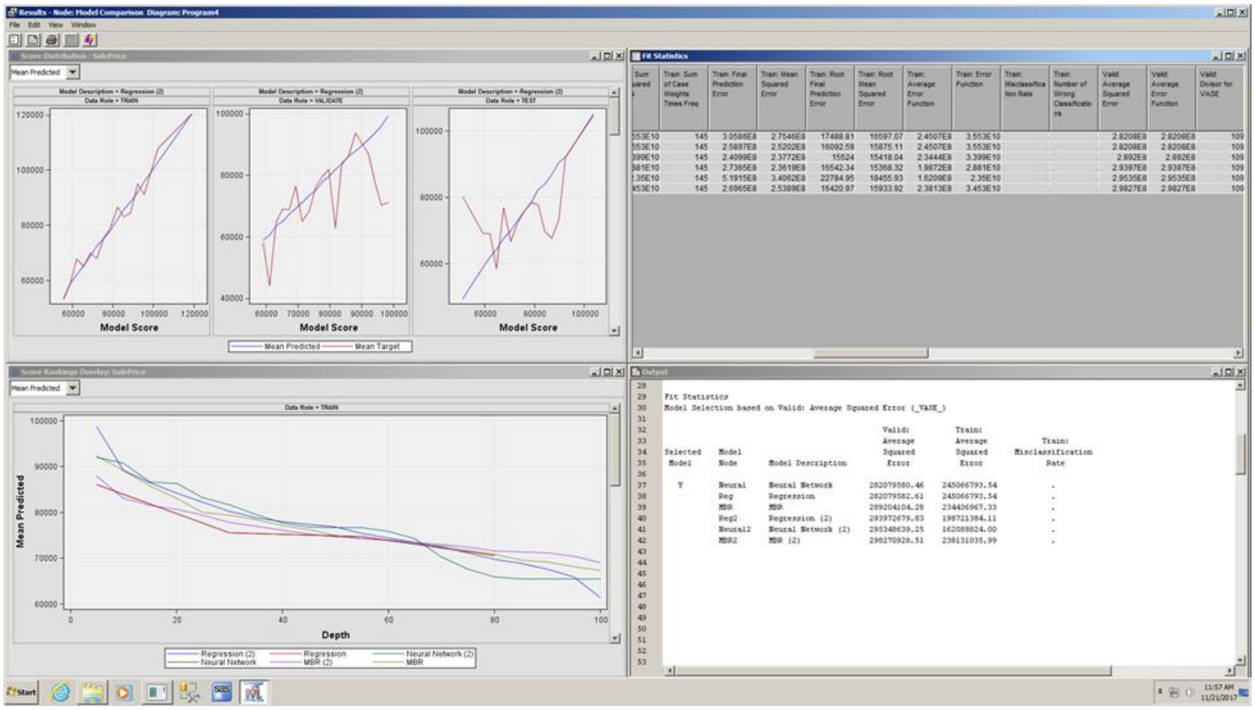
After examining the effects and p values, there is no need to use the impute or replacement node because after variable selection, transformation, and interactive binning are used, their purpose is to replace values, so there should be no missing values in the data set after these stages in the workflow diagram. The most significant f value after analyzing the effects is the age variable, which makes sense considering it was first of all assigned to the interval level, and these types have the most significant distance clearly defined between the values/levels this variable takes. It is such a broad variable that in terms of results produced, and because age had negative values before it underwent the transformation and filtering node, the variables change based on their performance of the data cleaning.

In terms of errors in the assessment score distributions for all six models, I believe the best model to predict the sale prices more accurately would be the linear regression that does not undergo filtering, because the mean predicted has a much steadier flow than any of the other models, and proves that it will have the most significant results when it comes to sale price, the target variable. Not even removing the outliers after filtering and variable transformation will produce more accurate results with less errors, its only purpose is to improve data preparation and not have as many issues.

In terms of low RMSE, MAE, and high R^2, obtaining great results was hard as the older houses in low-end neighborhoods are difficult to appraise accurately due to their lack of homogeneity. In future projects, perhaps using a clustering node to group similar properties together would be the key to creating the models that could possibly yield better prediction results, or smaller errors so to speak.

In terms of how well this specific project got students like me more acquainted with the software, the instructions on how to complete it was a 5 on a scale of 1 to 10, because even though it gave us an idea of what nodes to use, the instructions were pretty vague compared to previous projects and tutorials. And not having used filter and MBR nodes before was challenging, because until we figure out how to set up the diagram and properly connect all the nodes, we were experiencing run errors, but hopefully our results will do. Evan’s contribution to this project was the flow diagram, and Kimberly’s contribution was the report summary.





Sum uared s	Train: Sum of Case Weights Times Freq	Train: Final Prediction Error	Train: Mean Squared Error	Train: Root Final Prediction Error	Train: Root Mean Squared Error	Train: Average Error Function	Train: Error Function	Train: Misclassification Rate	Train: Number of Wrong Classifications	Valid: Average Squared Error	Valid: Average Error Function	Valid: Divisor for VASE
553E10	145	3.0586E8	2.7546E8	17488.81	16597.07	2.4507E8	3.553E10			2.8208E8	2.8208E8	109
553E10	145	2.5897E8	2.5202E8	16092.59	15875.11	2.4507E8	3.553E10			2.8208E8	2.8208E8	109
399E10	145	2.4099E8	2.3772E8	15524	15418.04	2.3444E8	3.399E10			2.892E8	2.892E8	109
881E10	145	2.7365E8	2.3619E8	16542.34	15368.32	1.9872E8	2.881E10			2.9397E8	2.9397E8	109
.35E10	145	5.1915E8	3.4062E8	22784.95	18455.93	1.6209E8	2.35E10			2.9535E8	2.9535E8	109
453E10	145	2.6965E8	2.5389E8	16420.97	15933.92	2.3813E8	3.453E10			2.9827E8	2.9827E8	109

```

Fit Statistics
Model Selection based on Valid: Average Squared Error (_VASE_)

                                                 Valid:          Train:
                                                 Average         Average          Train:
Selected    Model                                Squared         Squared      Misclassification
 Model      Node       Model Description          Error           Error            Rate

   Y        Neural     Neural Network          282079580.46    245066793.54          .
            Reg        Regression              282079582.61    245066793.54          .
            MBR        MBR                     289204104.28    234436967.33          .
            Reg2       Regression (2)          293972679.83    198721384.11          .
            Neural2    Neural Network (2)      295348639.25    162088824.00          .
            MBR2       MBR (2)                 298270928.51    238131035.99          .
```