DS6372 – Group Project 1 – Kaggle housing Study Linear model

The Kaggle study in question involves developing a model based on 81 variables to correlate home features with sale price. The data for this study is complicated. 43 of the data points are strictly categorical, and an additional 4 are integers which could be considered categorical (such as month of the year and mssubclass) and 6965 cells within the imported data frame have N/A values. In order to perform analysis on the data, dummy values will have to be placed for these abnormalities.

By plotting some of our continuous data, a couple of observations become obvious. Most data seems to be clustered near a centroid with a large scattered tail on either the high end, the low end, or both. Additionally, most of the continuous data seems to be used to describe features that do not seem to be present in all houses. In the chart be low of sales price vs wooddecksf, garagearea, and 2ndflrsf you can observe a large vertical line at zero. Since this value is effectively a n/a value, we will treat the presence of a feature, such as a garage or a 2nd floor, as categorical, and insert a dummy value to tell our model the feature is there. Another place we can use this same methodology, is with the yearremodadd variable, In that case, we will create an additional variable called ‘has\_remodadd’ where the variable will have a value of TRUE whenever the year built is not equal to the year remodeled.

|  |  |
| --- | --- |
|  |  |
|  |  |

The strategy we will be using to fill null values is to iterate over all the columns. If the column is numeric, we will add another category called “\_isNA” which will fill in for all null\_values. For continuous variables, we will substitute the median value for the N/A value. We are choosing the median value because most columns seem to have a long tail, and using the median should better protect our model from extreme data points. After filling all null-values, we will create the has\_ variables for continuous values which represent the presence of that feature (such as a porch or basement), and convert the following variables to log scale: (saleprice, grlivingarea, lotarea, 2stflrsf, 2ndflrsf, bsmntunfsf, bsmntfinsf1, totalbsmntsf, garagearea, wooddecksf and openporchsf). Overall this gives us a much more linear model then we had prior to log-transformations as shown in the figures below: Keep in mind values at 0 have been omitted as they are now supplanted by dummy variables.

|  |  |
| --- | --- |
|  |  |
|  |  |

With all our variables transformed into a more palatable form, it is time to begin modeling. Our first model will be a simple model based on 15 primarily continuous varaibles which were found to be statistically significant at a .95 confidence level by running individual models verses saleprice. Those variables are represented by the block indented SAS code below

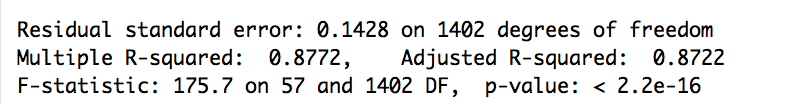
lm(formula = saleprice ~ landcontour + lotarea + neighborhood +

bldgtype + overallqual + foundation + bsmtqual + heatingqc +

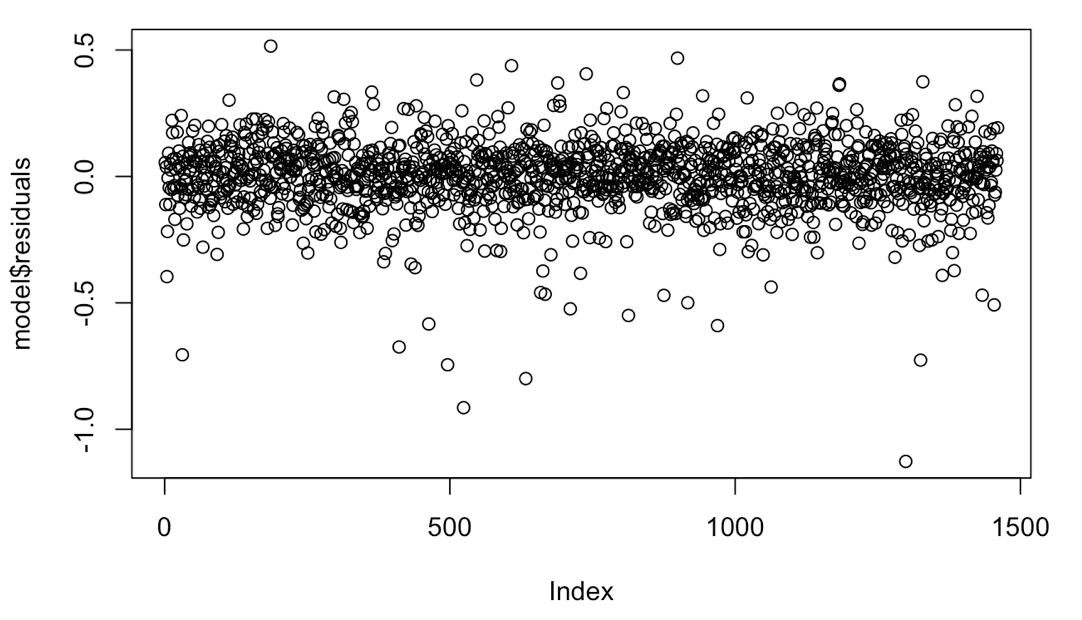
x1stflrsf + x2ndflrsf + fullbath + bedroomabvgr + fireplacequ +

yearremodadd + garagecars, data = sub\_frame)

Even though there were only 15 variables included, the final model has 57 coefficients due to all the categorical variables (which are automatically handled by the lm command within R). The summary overview is shown in the below screenshot taken from R.



In addition to our statistics, we also took a look at the residuals (below).



The residuals seem to be normally and uniformly distributed; so this seems like a linear model should be valid for our analysis.

Overall, an adjusted R-squared of .8722 seems very strong, but there is the possibility we are overfitting this model to the test set, but this model was primarily intended to give us some insights into the primary drivers of sale price, and when we output those values, our strongest drivers are listed in the table below:

|  |  |  |
| --- | --- | --- |
| **Category** | **Coefficient (linear)** | **p-value** |
| x1stflrsf | 3.97246 | < 2e-16 |
| bsmtqualEx | 3.708347 | 5.12e-09 |
| bsmtqualGd | 3.349694 | 8.38e-06 |
| neighborhoodStoneBr | 3.279167 | 0.000267 |
| fireplacequEx | 3.01289 | 0.002673 |
| foundationPConc | 2.957821 | 8.06e-06 |
| lotarea | 2.955896 | 8.71e-10 |

An interesting observation about the factors listed above is that they are almost all categorical (with the exception of 1st floor square footage. Overall it appears as if this is a trend throughout the coefficients. The data with the most predictive power is categorical. Lot area is the 2nd most predictive continuous value in our model, and it is the 16th largest coefficient. To try to better understand some of these categorical values, we will plot them against saleprice. These plots are shown below.

|  |  |
| --- | --- |
|  |  |
|  |  |

Looking at some of our quality predictors above. The 1st floor square footage remains a strong predictor with normally distributed residuals. The data is nicely distributed throughout the y and x axis, and the .95 confidence intervals seem to accurately capture 95% of the data points. This is less true for the lotarea data, our 2nd best continuous predictor, lot area still; seems strong in confidence interval and normal distribution, but most of the data points are clumped around 8,000 sqft and the model is likely not predicting outside of it’s narrow range of influence.

Looking at the collinearity of some of our continuous variables, we find there is some strong covariance within the dataset.

|  |  |
| --- | --- |
| **Column1: column2:etc…** | **Interaction p-value** |
| x1stflrsf:yearremodadd:x2ndflrsf:garageyrblt:overallqual | 0.0002192613 |
| x1stflrsf:yearremodadd | 0.0015975620 |
| x1stflrsf:garageyrblt | 0.0016296621 |
| x1stflrsf:lotarea | 0.0034951131 |
| lotarea:yearremodadd:x2ndflrsf:overallqual | 0.0052057902 |
| x1stflrsf:lotarea:yearremodadd | 0.0056872233 |
| lotarea:yearremodadd:x2ndflrsf:garageyrblt:overallqual | 0.0063110126 |
| x1stflrsf:lotarea:garageyrblt | 0.0068042593 |
| lotarea:yearremodadd:garagearea:garageyrblt | 0.0069827554 |
| x1stflrsf:lotarea:yearremodadd:garageyrblt | 0.0092279787 |
| lotarea:garagearea:garageyrblt | 0.0279669105 |
| x1stflrsf:x2ndflrsf:garageyrblt | 0.0282345040 |
| lotarea:yearremodadd:x2ndflrsf:garagearea | 0.0291629938 |
| lotarea:yearremodadd:garagearea | 0.0327535216 |
| x1stflrsf:lotarea:yearremodadd:garagearea | 0.0372774281 |

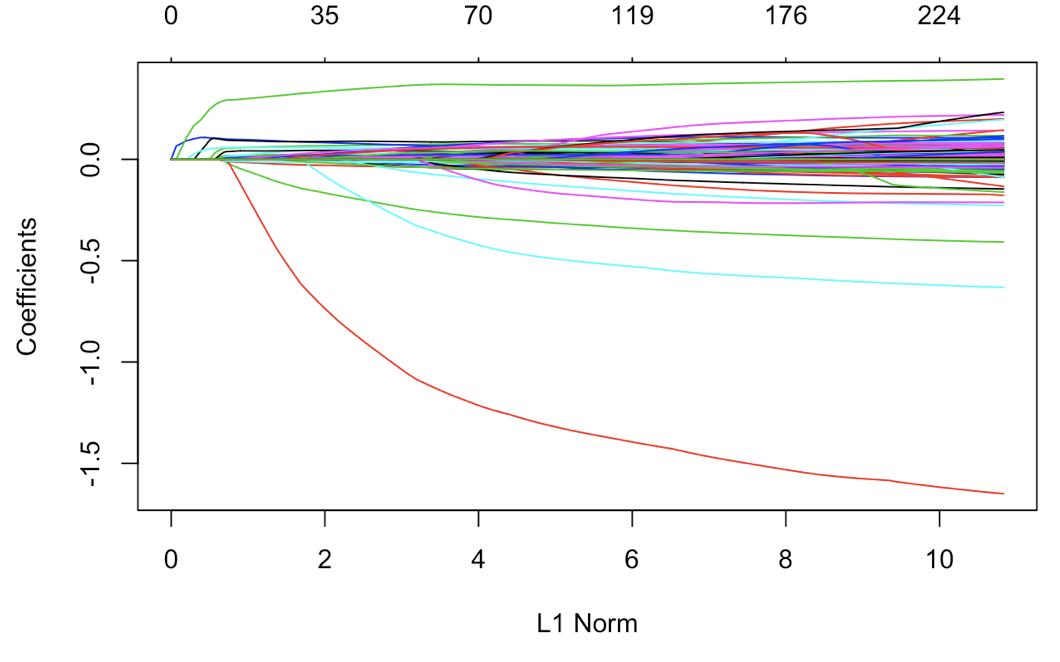
1st floor square footage is highly correlated with just about every other continuous variable. 2nd floor square footage also seems to be highly correlated with garage area, and year remolded. Overall it looks like just about every continuous variable is statistically significant with other variables. We suspect this is also true with categorical data, but it would take a very long time to manually go through all of those datasets and determine which points to keep and which to eliminate. A more efficient approach would be to utilize a Ridge regression model to reduce the negative impact of redundant predictor variables.

2. Our highest performing Kaggle model was a lasso regression on all the variables in the data set. We utilized the glmnet library with a min lambda ratio of .00005 and 2500 different lambda iterations. While applying max in sample accuracy as our stopping criteria, the model performance

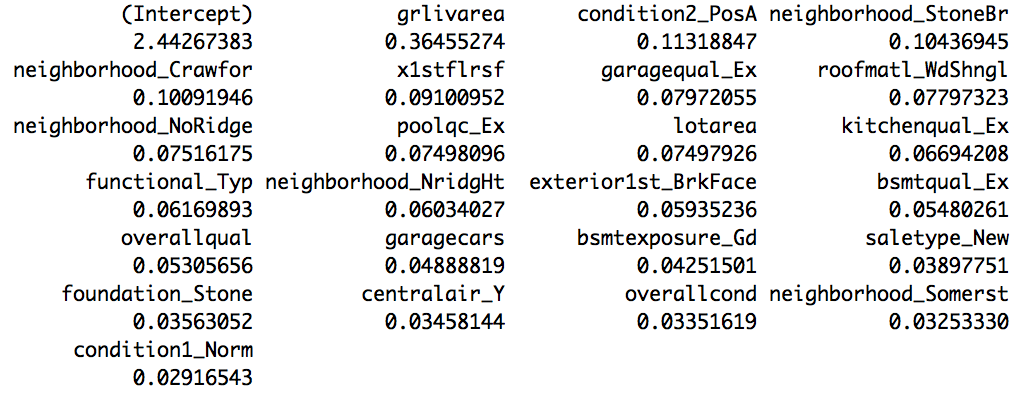
Without cross validating, the model had a training set average MSE of .00085 on the log scale data. On the 1500th lambda iteration. The top 10 coefficients used in that model are shown below.



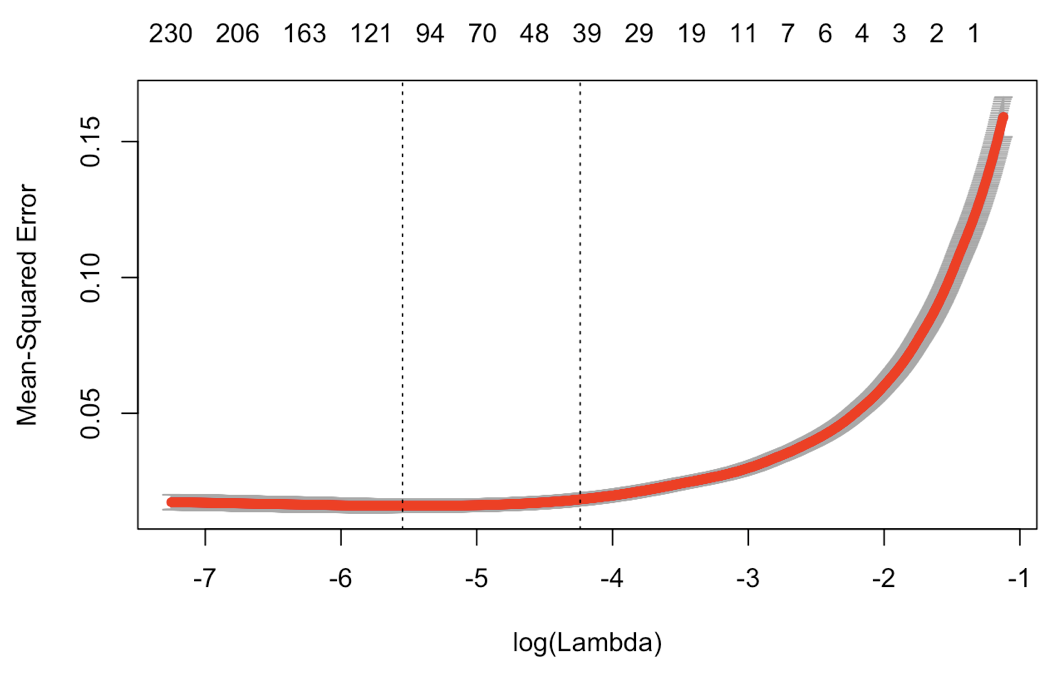
And the model of coefficient fallout as lambda is increased is also shown below.

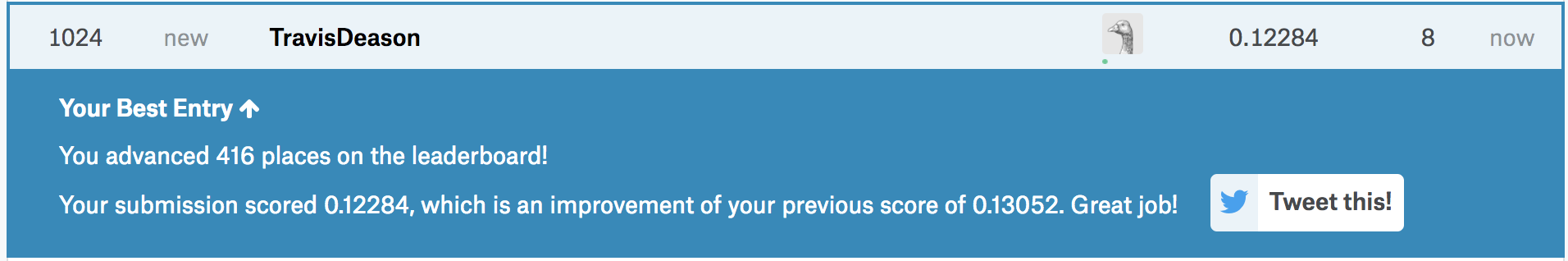


Using a 10 fold cross validated version of the same glmnet model as shown (where cross validation was used as the optimization criteria), we obtained a MSE on the test set of .0371. Overall our coefficient set used was similar in the cross-validated model.



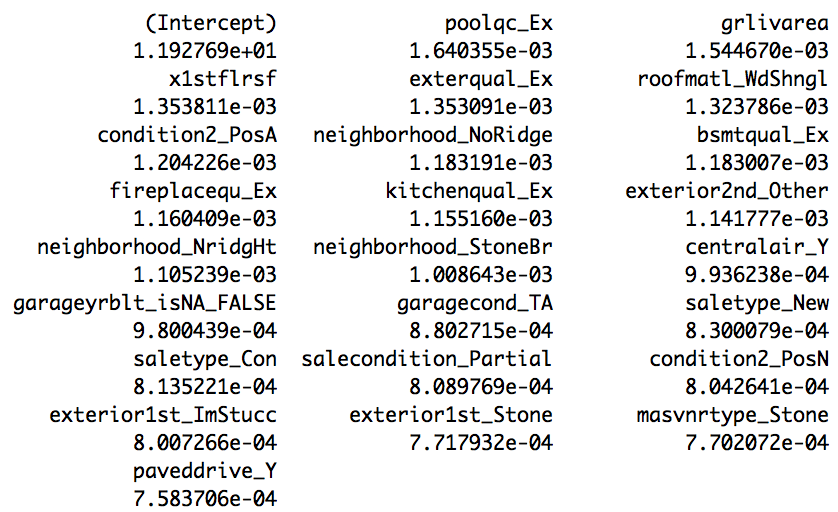
And the MSE/Lambda curve shows that 2500 iterations may have been slightly overkill with the curve starting to rebound at ln(lambda) = -6



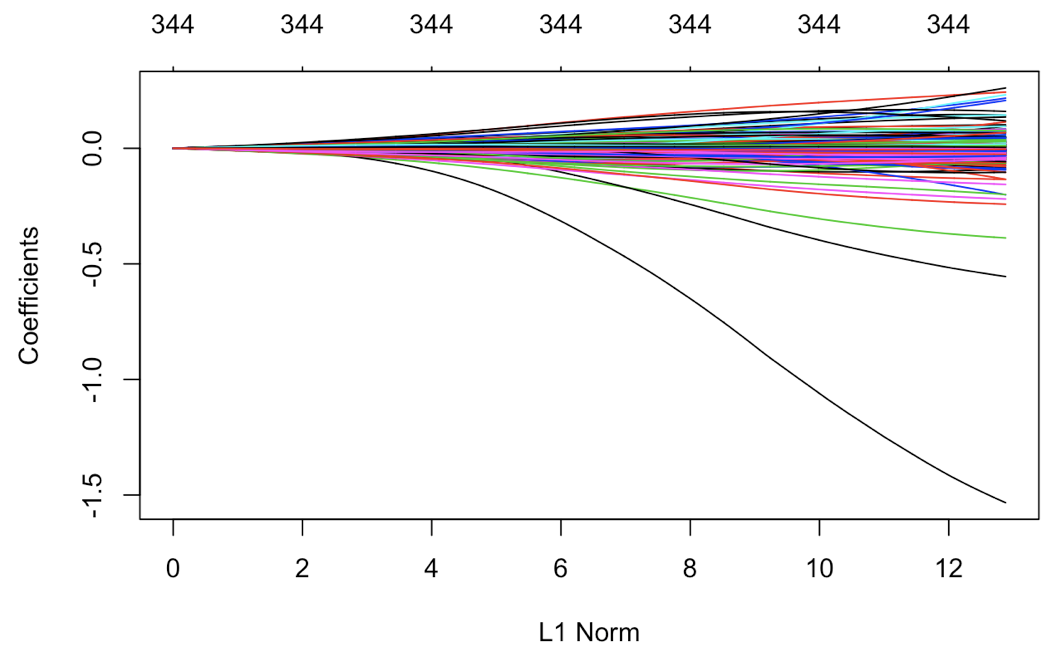


Our next best Kaggle model was a Ridge regression on all the variables in the data set. We utilized the glmnet library with a min lambda ratio of .00005 and 2500 different lambda iterations. While applying max in sample accuracy as our stopping criteria, the model performance

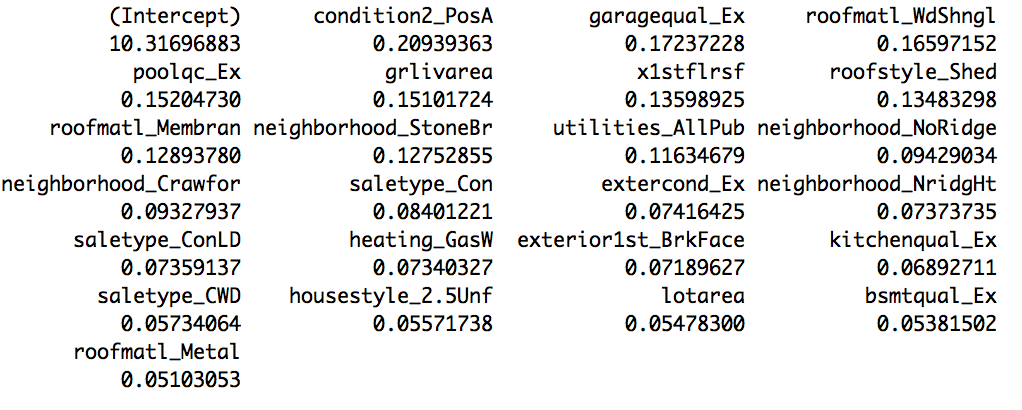
Without cross validating, the model had a training set average MSE of .036 on the log scale data. On the 2300th lambda iteration. The top 10 coefficients used in that model are shown below.



And the model of coefficient fallout as lambda is increased is also shown below.



Using a 10 fold cross validated version of the same glmnet model as shown (where cross validation was used as the optimization criteria). The coefficents utilized in the cross validated model were notably diffirent then the standard Ridge model.



And the MSE/Lambda curve shows that Ridge regression required much more iterations to converge on a more reliable model then LASSO, however our min MSE with a cross validated LASSO was .079 (significantly higher then with the LASSO model)

