Kamryn Parker

DATA-LA 485

Professor McGinnis-Brown

Projet 4

Predicting House Prices

Background

Housing prices have long been something that can be mysterious and hard to understand. Recently the economy has been recovering from COVID-19 and housing prices have skyrocketed since. Being an informed buyer can be one of the best advantages possible. This project focuses on attempting to determine house prices based on numerous factors that go into purchasing a home. Creating a model that can be accurate in determining a home price can be monumental in helping home buyers decrease buyers remorse and know they are getting the best home possible.

Methods

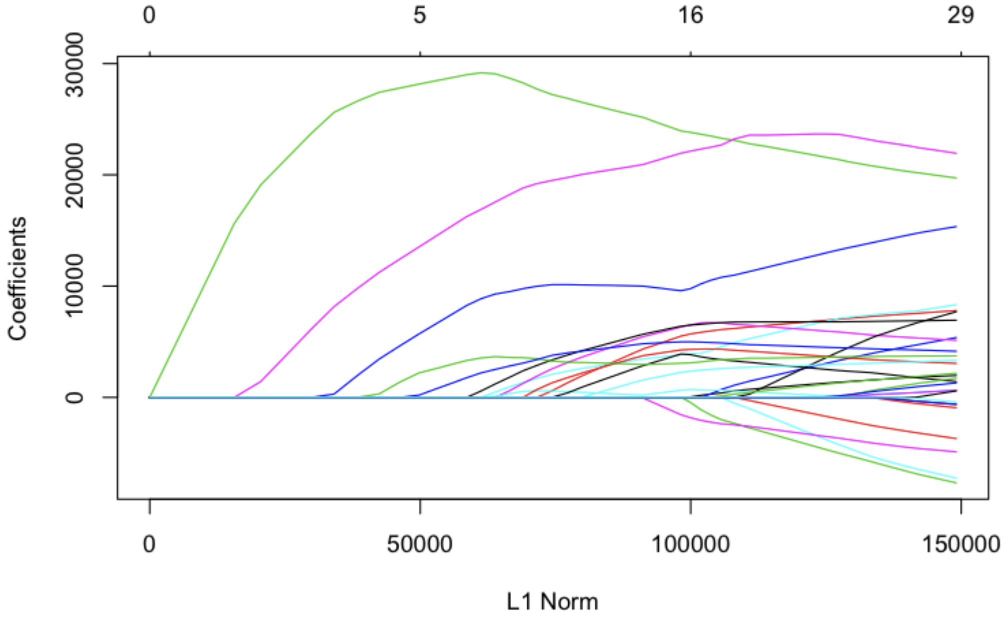
The data was collected by Dean De Cock in 2011 and was compiled for Kaggle (an online competition site). It describes the prices and features of over 1,400 homes that sold in Ames, Iowa since 2004. This dataset was pretty tricky to choose a model for because the target variable was continuous data instead of categorical. However, I was up to the challenge and did some research on the different algorithms. The one that I landed on was a Lasso Regression.

Lasso regression is a model where values are essentially “shrunk” down towards a central point. However, with this model I could only enter numeric data and not categorical. That was a sacrifice I decided to make with this specific model because lasso regression can be helpful with continuous target variables. I included all years in the data from 2004 to 2010 because I did not have another reason not to.

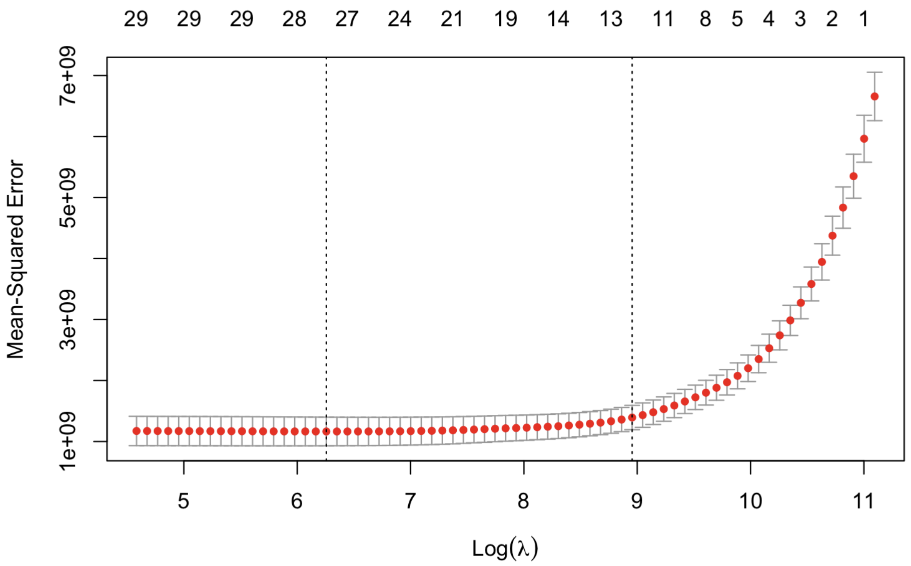
To set up the data I first split the given train data into a train and test split (just to evaluate it). Then I did a select() and selected only the numeric data columns. I then used the scaled() function to scale all of the data because the lasso regression works best with scaled data. I had to transform the data back into a dataframe after this function. I left out the target variable from this set so that it would not scale with the other features. So I had to use cbind() to combine the target column with the rest of the features. Finally, I dropped all the NAs in this set to limit missing variables. I decided to do it after scaling and subsetting the data because if I dropped NAs before all of that it would pretty much eliminate the entire dataset. In order to get my dataset to cooperate with the glmnet model function I actually had to transform my data into a matrix. After all of this I was able to run the function.

Overview of Data

After running the model I plotted this simple plot of the coefficients.



This plot shows the sum of magnitudes of the vector coefficients. We can see that most of the coefficients stay near zero or at least under 1000. From this I used cross validation to find the model that gave the best predictions or in the lasso regression model, the best lambda. I fit the lasso model on the training data again and drew a plot of the Mean Squared Error as a function of Lambda.



This plot shows us how the model grows as the lambdas get larger. So, there is a smaller mean squared error as the log(lambda) increases. After using cross validation the best lambda value found was 689.6322. This matches up with the plot because the log of this lambda is approximately 2.83861. As we can see in the plot the top values are around that area. I then used this in the prediction of the model fit to predict the test variables. I found I had an accuracy between 78 and 79% on various runs of the data.

Some of the limitations this data could have is how the predictions are not exactly a whole number like the target variables are originally formatted. However, when testing rounding these values there wasn't much of a difference in accuracy. It is worth noting however so that people don't think my model is inaccurate but rather it isn't rounded like the original data is. There are also general errors in the data potentially because I did not include categorical data so it might not be perfectly correct based on that exclusion alone.

Implications

Overall, I was pretty impressed with obtaining an initial accuracy of 78-79%. I expected a much lower accuracy considering this was my first time working with a non-categorical target variable and new model. With a model score of this high, doing some more feature engineering could improve such a score at least by a little bit but probably not an extraneous amount. Some more exploration that could be done is calculating how far off the original prediction was to the actual prediction. This could help with seeing how "generally" accurate the model can be.

	Pred	actual	difference
1	220570.08	208500	12070.08101
2	187040.99	181500	5540.98597
3	222104.01	223500	1395.99453
4	181156.16	140000	41156.16435
5	289019.22	250000	39019.22166
7	270430.01	307000	36569.99159
9	166485.38	129900	36585.37559
10	105169.86	118000	12830.14252

The table above details the predictions from the actuals and the differences between them. To continue my point from above, looking at the differences in the predictions vs. the actual can potentially help understand if your model was at least in the ball-park of estimation. Looking at just the first 10 results it can be seen there is a pretty large spread of whether a price was less than a $10,000 difference or much larger. So, it can be concluded that my model still has some improvements to be made.

In the future a model like this could potentially be used to help housing appraisers accurately appraise different houses because they would be able to input almost every feature thought of and get a roughly accurate result. It can also help home buyers have a transparent look at fair home prices.

I submitted the predictions on Kaggle and got a RMSE of 7.89476. This is not great of course but I think there is some more tuning that can be done like I have already explained before. One of the reasons I think there was such a high RMSE is because I had about 230 NA predictions so I had to input those to 0. I am not sure at the moment how to fix those issues however considering I had to have the exact amount of rows originally for submission. So, when I ran the predictions with my model I couldn’t drop the NAs like before.

Continuing work on this model would look like being able to use both the numeric and non-numeric features in the data to hopefully make an even better model. For now though, I am happy with the results I was able to find utilizing the various modeling techniques I used from the course.