Jonathan Malone

MATH 5373

November 7, 2023

Abstract

The King County Housing data set is a robust data set, containing variables for 21,613 homes sold over a one-year period from May 2014 to May 2015. Although there are 18 independent variables, not counting the unique IDs, they all play a role in determining the price of the house. However, by taking out similar variables (latitude/longitude compared to zip code; square footage above/living area footage 15/lot area 15 compared to living area/lot area), we fit a linear regression model consisting of fourteen of the available eighteen variables. After eliminating outliers and influencers, the percentage error was 18.66% with a p-value of 2.2e-16.

Even though all the variables played a role in coming up with the best model, the top three most significant variables are zip code, the square footage of the living area, and the “grade” given to the home by King County. The AIC scores for these three variables were significantly higher than any of the other variables, and each of these variables showed a high positive correlation to price.

Resources

Regression Modeling--PART A.pptx

Regression Modeling--PART B.pptx

Regression Modeling--PART C.pptx

Regression Modeling--PART D.pptx

github.com/NikhilKumarMutyala/Linear-Regression-from-scartch-on-KC-House-Dataset/blob/master/linear-regression-from-scratch.ipynb

The first thing I did before starting model creation was to clean and format the data to make it easier to work with. I removed the time component from the date variable since incorporated a character, so I could convert it from an object to a number. This made the date a usable variable to potentially incorporate into the model. The second thing I did was to update the year built if there had been a renovation of the house. By updating the year built to the renovation year, I was able to more accurately compare houses with similar features by assuming that houses that had been renovated had been brought up to code and used similar interior features as houses built that same year. This also removed a variable from the model, making it able to generalize more easily. Last, I removed similar variables to again make the model more generalizable. Since zip code and latitude/longitude both describe the location of the house, I removed the latitude and longitude variables. The variables sqft_living15 and sqft_lot15 correlated to renovations. Since I incorporated the renovation year into the year built, I removed these variables as well.

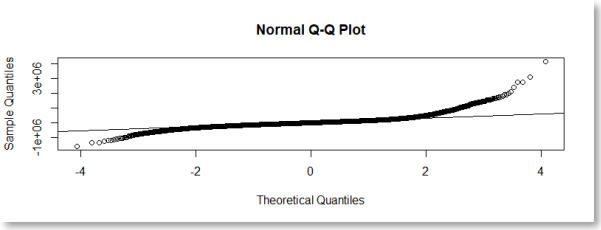
To build the model, I fed the remaining fifteen variables into the Akaike Information Criterion (AIC) metric to determine which variables to include in the model. While doing this, I converted the zip code, waterfront, grade, and condition variables into factors. After performing the metric forward, backward, and both directions, I was able to develop a model using the results to then predict future housing prices.

```
Call:
lm(formula = price ~ factor(zipcode) + sqft_living + factor(grade) +
    view + factor(condition) + factor(waterfront) + sqft_lot +
    date + sqft_above + floors + bathrooms + yr_built + bedrooms,
    data = filtered_data)
```

To evaluate the selected model, I printed a model summary. It seemed to show the model was well fitted, with a median residual only 9140 less than the actual data. The p-value for the model was very statistically significant at 2.2e-16.

```
Residual standard error: 145800 on 20746 degrees of freedom
Multiple R-squared:  0.5787,    Adjusted R-squared:  0.5782
F-statistic:  1187 on 24 and 20746 DF,  p-value: < 2.2e-16
```

To continue evaluating the model, I ran a Q-Q plot of the residuals to check for normal distribution and found there to be huge discrepancies at the left and right tails. The percentage error for the model was quite high at 28.12%, and there were multiple data points with a leverage above 0.8, with one even having a perfect 1.0 leverage score.



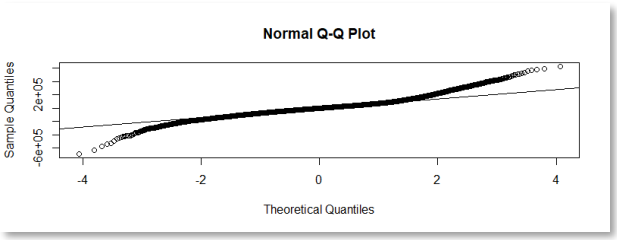

```
> sigma(selected.lm)*100/mean(housing_data$price)
[1] 28.12427
```

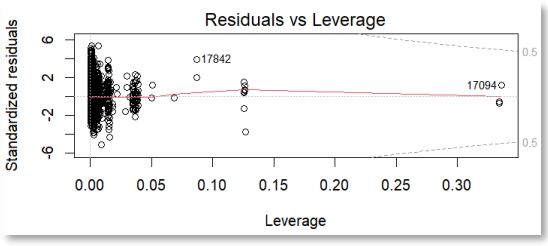
To combat this, I sought to remove the outliers that were over-influencing the model results. I used R code to do this. I selected any outlier with a z-score greater than 2 from the data set and created a new data set with the filtered results.

```
> cooks_distance <- cooks.distance(selected.lm)
> z_scores <- abs(scale(housing_data$price))
> outlier_indices <- which(z_scores > 2)
> filtered_data <- housing_data[-outlier_indices, ]
> selected.lm <- lm(price ~ factor(zipcode) + bedrooms + bathrooms + sqft_living
+ sqft_lot + floors + factor(waterfront) + view + factor(condition) + factor(grad
e) + sqft_above + sqft_basement + yr_built + date, data = filtered_data)
```

After, I rebuilt the model again using AIC metrics to choose the variables. Finally, I reran the model to test the results. The new percentage error was reduced from 28.12% to only 18.66%. The tails of the Q-Q plot were much more in line with the normal distribution. Lastly, there were no longer any data points with leverage scores above 0.5.

```
> sigma(selected.lm)*100/mean(filtered_data$price)
[1] 18.65862
```





In building the model, I found most variables had some degree of effect, so excluding one seemed to be the wrong choice. However, to build an easier model to work with, I wanted to choose the least number of variables that gave almost as good of a percentage error. There were three variables according to AIC that had the biggest impact – zip code, square footage, and grade. After these three, there was minimal difference in the AIC when adding the remaining variables. These three variables also had the biggest correlation to price. So, I built a new model taking only these three variables into account and compared it to the more robust model. It performed almost just as well, with a percentage error of only 20.41%, compared to the bigger model's 18.66%. This also made the model more generalizable, having fewer variables.

```
> selected.lm <- lm(formula = price ~ factor(zipcode) + sqft_living + factor(grad
e), data = filtered_data)
> sigma(selected.lm)*100/mean(filtered_data$price)
[1] 20.41351
```

In conclusion, I believe the more robust model did very well and can be a very accurate predictor of future prices once the outliers have been removed. I do believe that it could be harder to work with; you must clean more data, and with so many variables included, you take the chance of not having all the needed information to include in the model. Using the abbreviated model gives almost as good of an indicator of price, with lower upkeep and being more user-friendly. I would use the larger model in a business setting where data is more available, and the enhanced smaller model in a customer-centric way for clients to be able to do front-end research on their own.