Capstone Project - Predicting the Price of a House in the LA Region

By: Hoddy Mahon

Mentor: Amir Ziai

Springboard Data Science Intensive

July 3, 2018

Updated: 7/24/18

**Table of Contents**

**1 Introduction 3**

**2 Data Acquisition and Cleaning 4**

**3 Data Cleaning Wrangling 5**

**4 Data Exploration 8**

**5 Modeling 13**

**6 Using Model and Recommendations 14**

**7 Assumptions and Limitations 15**

**8 Conclusions 16**

**1 Introduction**

For the large majority of Americans, a home will be the largest purchase one will make in their lifetime. Yet, very few people know how a house is priced. Sure, one could look at similar houses in the area that were just bought. But what if a similar house hasn’t been bought in a while in that area? Shouldn’t an individual have some level of certainty that they are paying fair value on their home before making such a large purchase?

This project aims to build a supervised machine learning model that will predict housing prices in a similar manner to the Zillow ‘Zestimate’. With such a model at an individual’s disposal, transparency can exist in the market allowing them to have a trusted guide in this large transaction.

**2 Data Acquisition**

The data is acquired from the Kaggle Competition: Zillow’s Home Value Prediction (Zestimate). Now the competition itself is focused on a different problem than the one this project tackles. The competition wants the user to predict the log-error between Zillow’s Zestimate and the actual sale price, given all the features of a home.

In this project, we will attempt to build a new model to calculate the value of the home. We will use the data supplied by Zillow which consists of 2017 housing data from three southern California counties.

The dataset consists of close to 3 million records and 58 features. Having a dataset this large presented some difficulties due to limited computational resources. 100k records of the 3 million records were used in many situations to save computational time and allowed for the focus to be on implementing the techniques described in this course.

An important point to note is that the actual price a home sold for is unavailable in this round of the Kaggle competition. A user had to advance past the first round to get to that point. Therefore, this data is not available to use in this project. To compensate, the feature ‘taxvaluedollarcnt’ is used as the dependent variable that we are aiming to solve via this model. According to the data dictionary, ‘taxvaluedollarcnt’ is the total tax assessed value of the parcel.

**3 Data Cleaning and Wrangling**

The biggest challenges in this dataset were dealing with missing values as well as dealing with highly correlated features that were essentially the same thing. An example of the latter being, ‘bathroomcnt', 'calculatedbathnbr', and 'fullbathcnt’. All three of these features are describing how many bathrooms the home has and were seen as being highly correlated on a heatmap.

The following are the steps that were used in cleaning this dataset:

1. First, I created a bar chart to observe the null values per feature. This was used as the basis for many of the feature engineering decisions to come.
2. Next, I created a heat map that shows the correlation amongst the dataset’s features. This also was used for many of the feature engineering decisions to come.
3. Originally ‘hashottuborspa’ feature had values of ‘True’ or ‘N/A’. I wanted to make this contain numerical values, so True values were converted to 1. Also, N/A’s were assumed to mean that the home did not have a hot tub or a spa so those were converted to 0.
4. Investigated the 'pooltypeid10' and 'hashottuborspa' fields. Based on the data dictionary, these two fields should contain the same data. They don’t, so we will remove the ‘pooltypeid10’ field as that has more null values (since we did Step 3).
5. I assumed for 3 pool related features, if the value was null that meant that a pool did not exist. The NA’s were converted to 0.
6. Analyzed the values for ‘poolcnt’ and ‘poolsizesum’ and made transformations. The logic here is that if ‘poolcnt’ equals 1, thus saying that a pool exists, ‘poolsizesum’ should be populated. Therefore, whenever ‘poolcnt’ equaled 1, and ‘poolsizesum’ was N/A, the median value of ‘poolsizesum’ was filled in. Additionally, when ‘poolcnt’ was 0, and thus saying that a pool does not exist, ‘poolsizesum’ was converted from N/A to 0.
7. I then observed the number of null values in ‘threequarterbathnbr’ and its value counts. Again, there were a large number of null values. In investigating the value counts, almost all of the feature’s values equal 1. Therefore, I assumed that the null values could be changed to 1 as well.
8. Observed the number of null values in ‘airconditioningtypeid’ and its value counts. There were a large number of null values. However, in investigating the value counts, a large majority of the values equal id 1. Looking in the data dictionary, that value corresponds to central ac. This makes sense logically that most homes have central ac. Therefore, I assumed that the null values could be changed to this id 1 value as well.
9. Following a similar process of that in Step 8, I observed the number of null values in ‘heatingorsystemtypeid’ and its value counts. Again, there were a large number of null values. In investigating the value counts, the majority of the values equal id 2. Looking in the data dictionary, that value corresponds to central heat. This makes sense logically that most homes have central heat. Therefore, I assumed that the null values could be changed to this id 2 value as well.
10. I then compared the difference in the number of null values between 'garagecarcnt' and 'garagetotalsqft'. There were the same number of null values in the two features which is what we’re expecting. Therefore, the next step was to convert the NA’s to 0 for both features. The assumptions being that if null in the ‘garagecarcnt’ field, it means there are no garages. Likewise, no garage means the size is 0 by default.
11. Next, I analyzed the values for ‘fireplacecnt’ and ‘fireplaceflag and made transformations. Based on the data dictionary, these two fields should be related. However, there are significantly more null values in ‘fireplaceflag’ than ‘fireplacecnt’. In reality, those null value counts should be equal. Because of the large discrepancy, I made the ‘fireplaceflag’ from scratch using ‘fireplacecnt’. Whenever, ‘fireplacecnt’ was greater than 0, ‘fireplaceflag’ was set to 1. Whenever ‘fireplacecnt’ was null, ‘fireplaceflag’ was set to 0. Lastly, whenever ‘fireplacecnt’ was null, that was converted to 0.
12. Using the correlation heat map, I observed that ‘calculatedfinishedsquarefeet’ and the 4 ‘finishedsquarefeet…’ columns were very highly correlated. I then looked at the amount of null values per column. The ‘calculatedfinishedsquarefeet’ column had the least amount of null values by a lot. Thus, the that column was kept in the dataset and the 4 ‘finishedsquarefeet…’ columns were removed.
13. Again, looking at the feature correlations heatmap, 'bathroomcnt', 'calculatedbathnbr', and 'fullbathcnt' are also highly correlated. Additionally, based on the data dictionary, they seem to be giving very similar information. Using the same strategy as in Step 12, I compared the amount of null values per feature. The ‘bathroomcnt’ feature had the fewest null values. Therefore, that feature was kept in the dataset and 'calculatedbathnbr' and 'fullbathcnt' were removed.
14. Remove fields that are missing more than 92% of the data after this feature engineering has taken place. The reason being that there is not sufficient data in the respective feature to be useful in the model creation.
15. Removed features that are of object data type as they will not work with the regression techniques I was testing.
16. For the null values in our target variable ‘taxvaluedollarcnt’, ‘taxamount’ will be used to fill them in. The reason being is that this feature is strongly correlated with ‘taxvaluedollarcnt’.
17. For all of the remaining null values, the median value is used to fill them in.
18. As mentioned in the ‘Data Acquisition’ section, because we are using ‘taxvaluedollarcnt’ as our target variable, the following fields were not used in building the model due to being too strongly correlated with the target variable: ‘structuretaxvaluedollarcnt’, ‘landtaxvaluedollarcnt’, and ‘taxamount’. The thought being that incorporating these features into the model would defeat the purpose of the problem we are trying to solve. Additionally, 'assessmentyear' was also removed from the dataset because its relevance is based off of the other tax related features. Since the other tax related features were removed, it seemed appropriate to remove this feature as well.
19. Lastly, I took the log of the target variable ‘taxvaluedollarcnt’. The reason here was the variance in this variable is very large, and I wanted the target variable to accurately reflect the changes in the model. Because we are trying to model something, and the mechanism acts via a relative change, using the log-scale for the dependent variable can be an important way of capturing the behavior seen in the data.

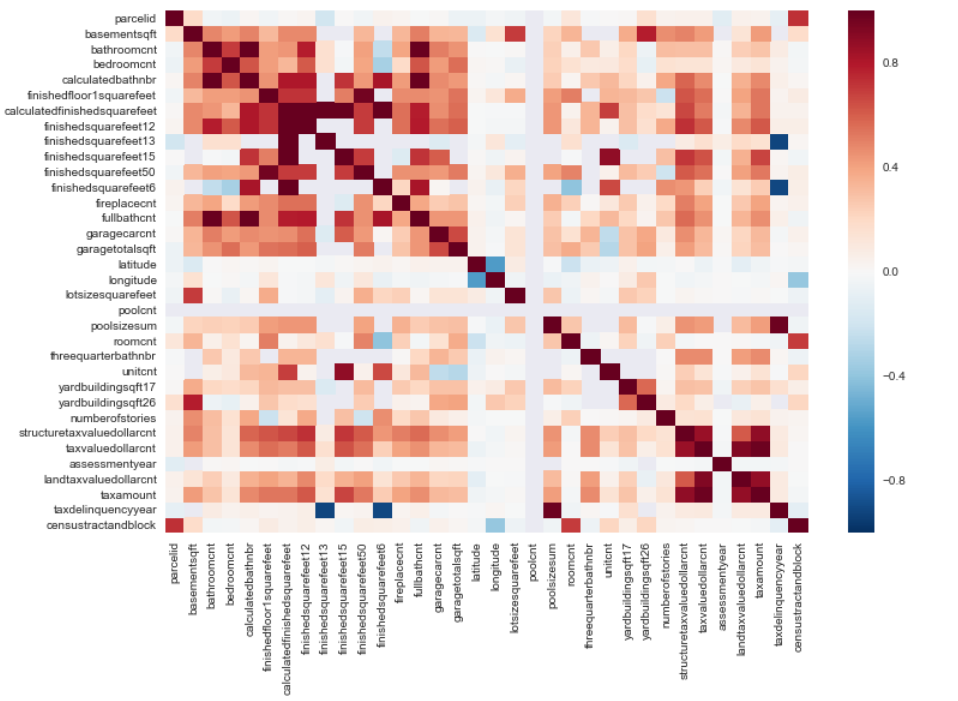
**4 Data Exploration** (code for this section can be found in the Data Exploration notebook)

The two most important data exploration activities were covered in the ‘Data Cleaning and Wrangling’ where I discussed the feature correlation heatmap and null value bar chart that were created. Those two plots were vital in the data cleaning and wrangling process as they gave me insight into what needed to be done from a feature engineering standpoint. Besides those two, I looked at how some features relate to the target variable.

Below are the type of plots that was created in the data exploration process and some takeaways for each.

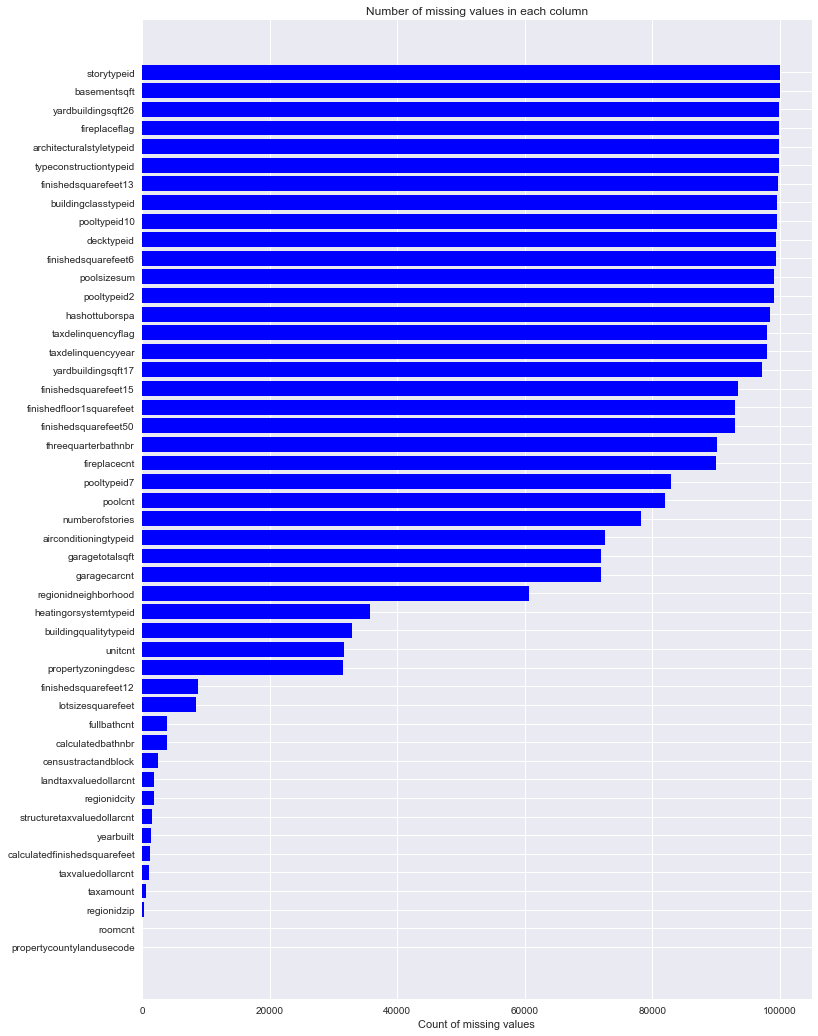
**Feature Correlation Heatmap**

As seen by the legend on the right side, dark red means a strong positive correlation amongst features while dark blue indicates a strong negative correlation amongst features. Observations from this were discussed in Section 3, ‘Data Cleaning and Wrangling’.



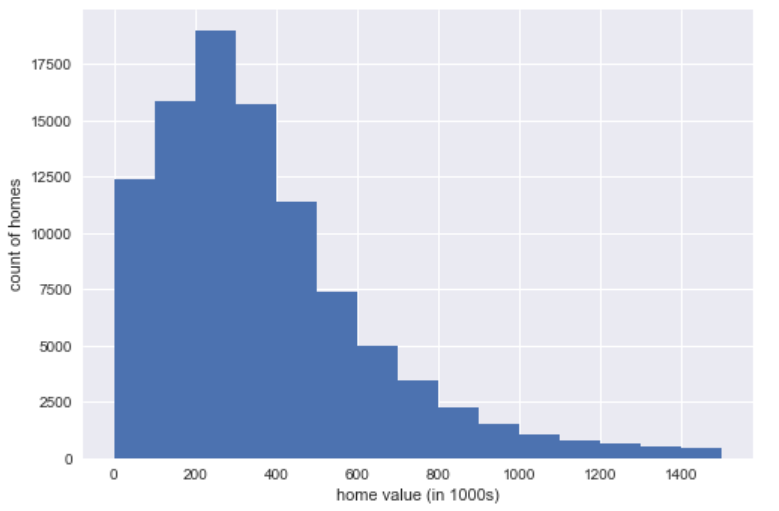
**Null Value Bar Chart**

As seen by this bar chart, there are many features with 90% or greater null values. Dealing with all of these null values, led to the feature engineering process to be a lengthy process.



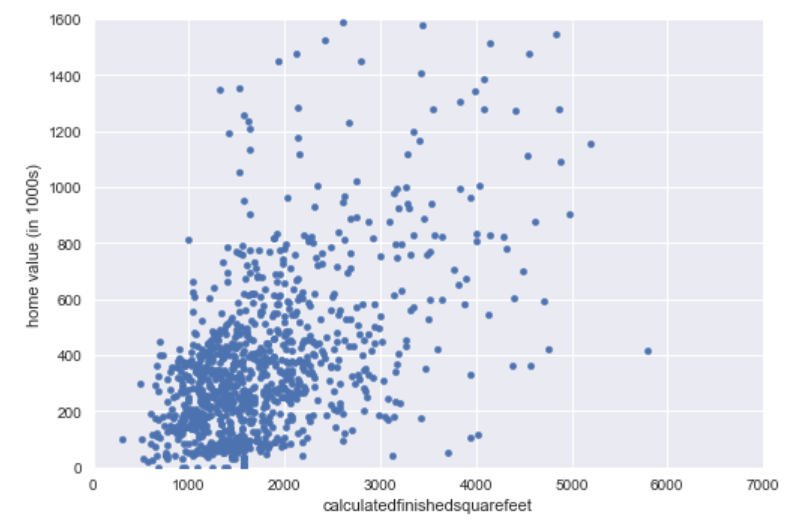
**Histogram of housing prices**

The housing prices are skewed right. This makes sense logically as most of the houses are around the 100 – 500k valuation. However, there are a good amount of more expensive homes and mansions that skew the dataset to the right.



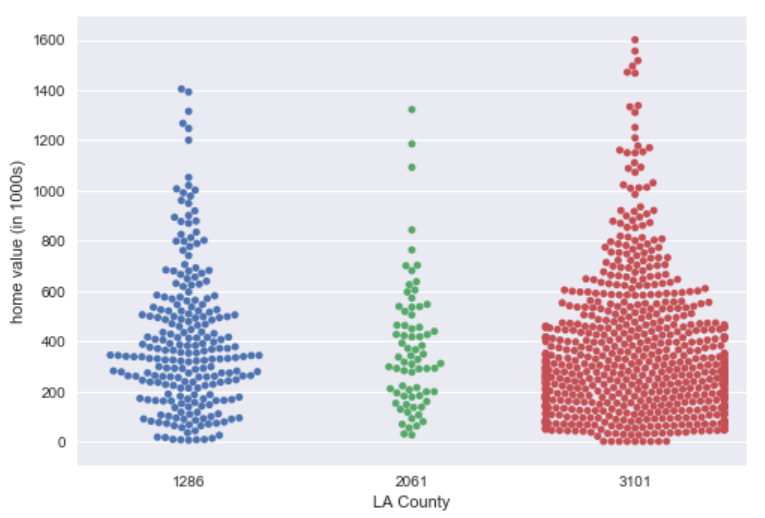
**Scatter plot showing relationship between the square footage of the home and the home value**

It does seem like there’s a fairly strong positive correlation between the ‘calculatedfinishedsquarefeet’ feature and ‘taxvaluedollarcnt’. This can be substantiated by the heatmap that was created above.



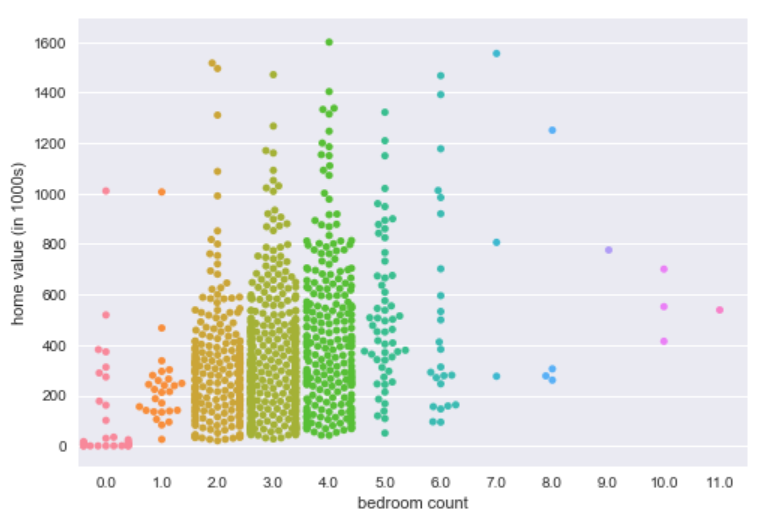
**Swarm plot showing home values per the 3 LA counties found in the dataset**

This swarm plot showing home values per the three LA Counties that are found in this dataset had a few interesting takeaways. First, the third county, 3101, has the largest volume of records by a large amount. It also appears to have both the poorest and the wealthiest homes. The first county, 1286, probably has the highest average looking at the swarm plot. While the second county, 2061, has the fewest data points by a significant amount and does not possess many homes over $800k.



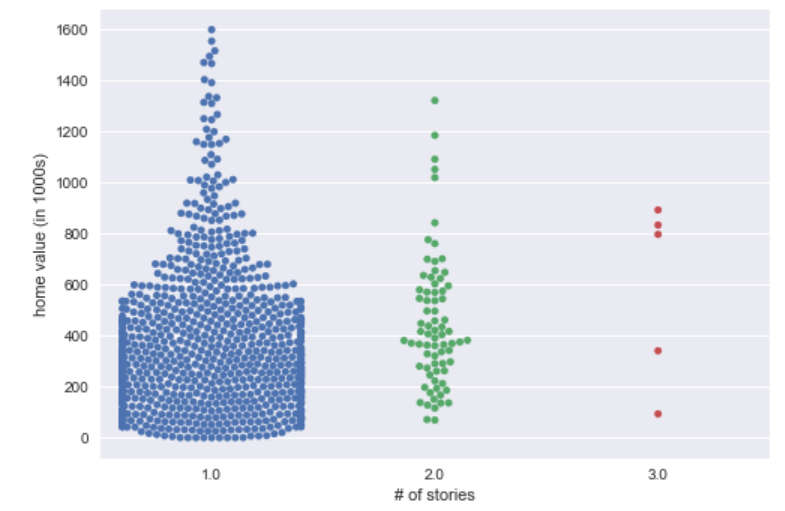
**Swarm plot showing home values per the number of bedrooms**

Most homes seem to have between 2-4 bedrooms. 4 bedroom houses appear to have the most amount of higher priced homes.



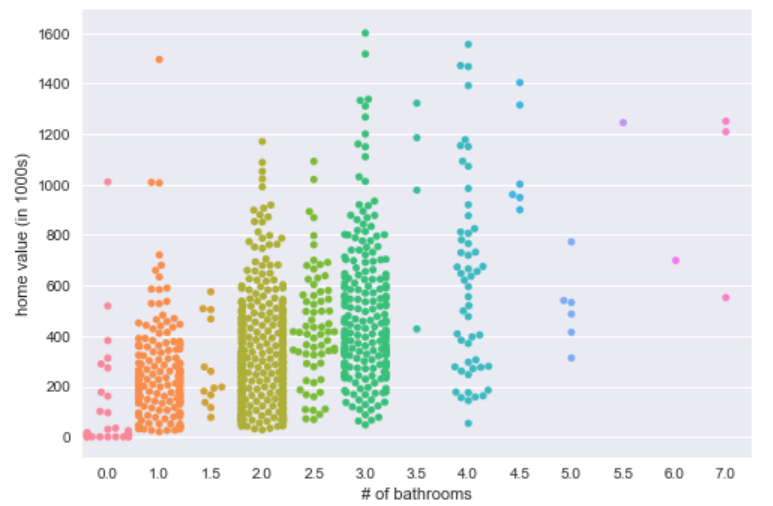
**Swarm plot showing home values per the number of stories a home has**

It doesn’t seem like three story houses are more expensive than one story houses, which I hypothesized. If anything, one story houses may be more expensive than three storied ones. It is a bit tough to tell considering the large volume of one story houses compared to three story houses however. Also, it appears that two story houses would have the highest average home value amongst the three values.



**Swarm plot showing home values per the number of bathrooms in a home**

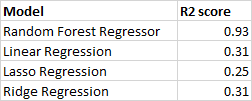
If the home has over 3.5 bathrooms, there is a very high chance the home is above the mean house price. Meanwhile, if the home only has 1 bathroom, odds are it is below the mean house price.



**5 Modeling**

I split 70% of my data into the training set and 30% of the data into the test set. Then I ran basic baseline models on this training data. Because the goal of this model is to predict the valuation of a home, a continuous value, the models that were run were: Random Forest Regressor, Linear Regression, Lasso Regression, and Ridge Regression.

Even though I did not tune any of the hyperparameters for these models, it was clear which model was performing the best on the training data. Here are the results of my model exploration on the training data:



Therefore, the model that was ultimately used was the Random Forest Regressor model.

When determining which model to pick, I did not use any hyperparameter tuning and simply stuck to the default model settings. However, after I chose a Random Forest Regressor, I used a Randomized search on the hyper parameters in order to tune them. Utilizing the results of this hyperparameter tuning, the model’s accuracy increased by .19%. The end result was that the model received an R2 score of .81 on the training data.

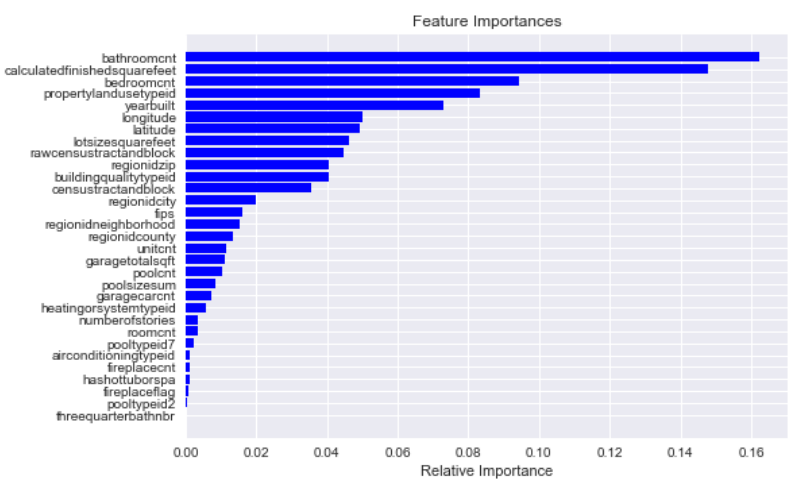
This model was then applied to the test data and received an R2 score of .66 and a mean squared error of .43. The R2 indicates the percentage of the variance in the dependent variable that the independent variables explain collectively; the mean squared error tells us the average squared difference between the estimated values and what is estimated.

**6 Using Model and Recommendations**

Because the model has a predictive accuracy in the mid 60% range, having this model at one’s disposal could serve as an aid in the house buying process but should not be the only resource one uses.

Looking at the feature importance graph below, the number of bathrooms, the square footage of a home, and the number of bedrooms are the three most important features in the model’s home valuations. This more or less confirms traditional thought in regards to home valuations.

An interesting point to note is that the first location-based feature that is shown on the below graph is ‘longitude’ and that is the 6th most important feature. This would seem to contradict the saying “LOCATION, LOCATION, LOCATION” when it comes to home valuations. However, this data was only taken from three southern California counties. If the assumption is that most of the areas in those counties are generally similar, one could see how a specific location in one of those counties does not have as much impact compared to if the dataset consisted of homes throughout the whole United States.



**7 Assumptions and Limitations**

As mentioned at the beginning of this paper, having a dependent variable that was the actual price a home sold for was not available in this round of the Kaggle competition. Therefore, ‘taxvaluedollarcnt’, which is the total tax assessed value of the parcel, was used as the target variable in its place. The assumption is that this tax assessed value would correlate very strongly to the actual price the home sold for. However, it would have been interesting to have that data to compare if that hypothesis is true.

The last major assumption was listed in Section 3, Step 16. For the null values in our target variable ‘taxvaluedollarcnt’, ‘taxamount’ was used to fill them in. The reason being was that this feature is strongly correlated with ‘taxvaluedollarcnt’. This impacted about 1k records which didn’t seem like a lot. At first, I wanted to delete these null records in the dependent variable. However, doing so and not filling them in with ‘taxamount’ drastically reduced the model’s predictive accuracy. Therefore, I made the decision that this transformation must occur in order for the model to have some predictive significance.

**8 Conclusions**

I created the model using a Random Forest Regressor and fitted it on the training data with 31 features. The end result was an R2 score of .66. Data cleaning, data exploration, experimenting with different model types, and cross validation were some of the data science components that were utilized in this final project.

The mantra that 80% of your time will be spent on gathering the data and preparing it and only 20% of your time will be spent on building the model hit home for me when working on this project. This was my first data science project, and I did not fully understand the end to end process until this point. There were a lot of feature engineering steps that needed to be done that I did not fully understand at the beginning of this course. Yet, these steps had to occur if my model was to have any sort of significance.

Overall, I learned how to approach a data science problem, how to explore data, how to make sense of the data, and how to tune a model. I know there’s still a lot to learn. However, I feel that I have a solid basis to build upon. Now, I know what I don’t know, which is something I could not say at the start of this course. I look forward to using what Springboard taught me to continue my data science journey and build upon this experience.