**Capstone 2 Milestone Report**

**Proposal**

The US housing market is worth approximately 33 Trillion dollars and is one of the biggest industries in the country. My goal in this project is to use data science tools to predict the expected listing price of a home in the US. And I will examine the model for insights.

Based on the data provided, I’m hoping to be able to provide an accurate estimate of the price for a home in the US, based on variables like the State, number of bedrooms, bath rooms, and using NLP to convert the text in the Title and Description of the listings to usable data.

Columns to exclude include Agent Information (all empty), and Crawl Date (when data was scraped by bot). I’ve also excluded the Price column as this is our target variable.

Our primary client would be the home Buyers and Sellers, whether they are first time sellers or house flippers. The goal of this project is to help them determine what is a realistic listing price for their property based on our analysis. This will help minimize the listing time for these properties and ensure that they are competitive in the market. It's not uncommon for some listings to go on for months if not years. Oftentimes Sellers may be pricing themselves out of a sale, due to overestimating their property value, or they may be losing money from under valuing their property. Alternatively if you are a Buyer you want to understand all options available to you to make an informed decision. Another target are real estate investors.

**Data Wrangling**

The dataset was obtained through Kraggle, via web scraping services off the Trulia website. This dataset was aggregated by the poster to help others perform analysis on the market data. Running through the data set there are roughly 30000 observations vs 68 columns. After processing the data sheet further it is estimated that there will be over 1000 possible variables. Variables range from NLP data, geographic data, time series, categorical variables, continuous and image data among others. Note that there are currently 17 states missing from this data set. After reviewing further, these states were omitted because this dataset is actually a sample of a larger one that was aggregated. The missing states are Arkansas, Connecticut, Delaware, Hawaii, Maine, Mississippi, Missouri, Montana, New Hampshire, North Dakota, Rhode Island, South Carolina, South Dakota, Utah, West Virginia, Wyoming. As such these states will be omitted in our analysis and model. My findings would also not apply to these states.

To start off with I imported numpy, pandas, and pyplot to read in the CSV dataset, which contained 30006 rows and 68 columns. Upon a quick visual inspection, I found that there were about 25 columns of image data, that linked back to the Trulia site, and a bunch of Nans in the Broker field. In order to get a better idea of what we were dealing with, I ran the code “df.isnull().sum()” and it revealed that there were 25 image data columns, 14 columns related to the Agents/Brokers which were all empty, a Property Type column which was promising at first but this code showed that out of 30006 values, 29942 were Nans. As a result I dropped these columns, as they were not useful or could not be used. In addition I also removed the ID columns which are internal tools to Trulia, as well as the Style column, which was a combination of the Beds/Bathroom columns in that it simply aggregated the two. To ensure that I had a comprehensive dataset for the US, I ran the code df.groupby([‘State’]).count(), and it showed that out of the 50 states, we only had data for 33 plus Washington DC. As a result these states will be omitted in our analysis and model. My findings would also not apply to these states.

The missing states are Arkansas, Connecticut, Delaware, Hawaii, Maine, Mississippi, Missouri, Montana, New Hampshire, North Dakota, Rhode Island, South Carolina, South Dakota, Utah, West Virginia, Wyoming.

After dropping these values, I focused on cleaning my dataset to remove any remaining Nans. The variables left that still contained Nans, were Description, Facts, Beds, Bath, Sqr\_Ft, Lot\_Size, Year\_Built, Price\_Sqr\_Ft, Last\_Sold\_Year, Last\_Sold\_For, Last\_Tax\_Assessemnt, Last\_Tax\_Year, and Days\_On\_Trulia. To start off with, I replaced the Nans in Description and Facts with empty strings, because I plan on using NLP to process the text there, and Nans will need to be replaced with these empty strings in order to do so. The next challenge, that I faced was the fact that for the columns Sqr\_Ft, Lot\_Size, Price\_Sqr\_Ft, Last\_Sold\_For, and Last\_Tax\_Assessment, the values should’ve been integers, but instead they were strings (for example, instead of saying 18000, it would show 18000ft). To remedy this, I used a regex to replace these special characters (.str.replace(r’\D+’, ‘’)), and then ‘pd.to\_numeric()’ to convert them to integers. Once this issue was resolved, I moved to replace the Nans with 0 using fillna, so I could process it further. Next I moved to replace these 0s with either the mean of that column (if the figure was simply a count or measurement) or the median (if the value was a year count), with the condition that it be by the given state (ie replace the nan of that listing, with the mean of that listings column by its state). To do this I used df.groupby(‘State’).Days\_On\_Trulia.transform(‘mean’). After completing this process, I checked the dataframe visually again, but to my surprise it showed that there was in fact an large number of listings that still had 0s. Upon further review, it looks like this was due to the fact that certain states had no data at all for certain variables. To remedy this, I filled them in with the averages for the respective columns.

**Initial Findings**

To review our variables for potential patterns to our target variable price, I plotted them in scatter graphs. The first two variables that were tried were the Beds and Bath variables. To ensure that I didn’t miss anything, I plotted the actual values, the residual plots, and transformed them via Log and Square Root. For both the Beds and Bath variables, the original, Log and Square Root graphs, did not show a particularly strong pattern, but the residuals graphs for both show a very slight negative linear regression relationship. To clarify these variables were plotted with Price on the Y axis, and were plotted separately.

The next variable I checked was the Sqr Ft variable. Plotting the regular, residual and square root graphs did not show any new patterns, but when plotting the log graph, it does show a parabola pattern of sorts.

Moving onto the Lot Size variable, plotting it against the Price variable did not show any particular pattern for the regular graph, square root graph, or residual graph. Checking the graph with the log transformation, does show a graph with a very faint bimodal distribution.

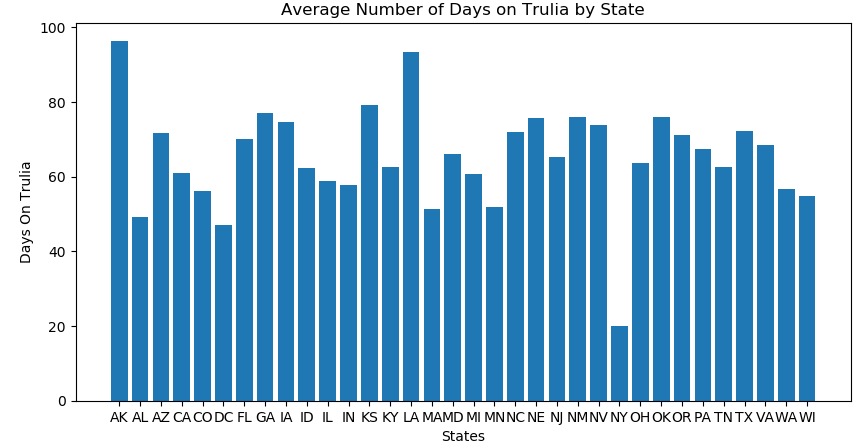
For the next variable Year\_Built, I changed it to Building Age, by subtracting the values of the column from the Max value. Plotting these variables against the Price variable, all 4 graphs, shows a very slight negative linear relationship between how old the house is and the price (ie, older the building the lower the value).

Moving on the variable Last Sold Year, I converted it to Time Since LS (Last Sale) by subtracting the values of the column from the max of the column. Plotting these values against the Price variable does not show any discernible patterns, in any of our graphs.

The next two variables I tested were Last Sold For and Last Tax assessment. For these two variables when I plotted them against the Price, the regular scatter plot, square root, and residual does not show a discernible pattern. However, when log transformation is applied to both of them, it shows a strong linear regression pattern.

And last but not least, checking the Days on Trulia variable and comparing it to the Price variable did not show any partners. Even after transforming our data there were no new patterns or anything that may hint at a pattern.

After wrangling and cleaning our dataset, I was able to find some very interesting information. Comparing the average number of Days on Trulia variable with the States variable, we can see on average how long a home can expect to be listed on Trulia.



**Statistical Analysis**

As part of my project on predicting housing prices in the US, I went through the data and explored a vast number of variables and potential connections. Something that did peak my curiosity is the value of houses in certain regions of the country. The general consensus from most Americans is that the major states (in our case New York, Texas, and California, the top three states in terms of GDP representing 31.4 % in the first quarter of 2020), have a higher cost of living and thus the housing prices are bigger.

For our statistical review, we will set the null hypothesis (Ho) as there is no difference between the price for a house in New York, Texas and California vs the rest of the country, whereas our alternative hypothesis (H1) is that there is a price difference when buying a home in New York, California and Texas vs the rest of the country.

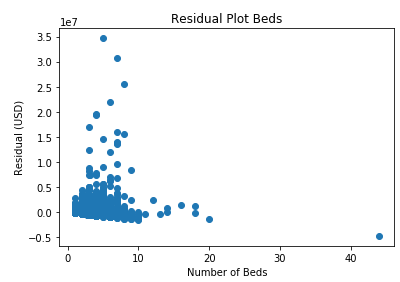
To complete the analysis, i pulled data from the ‘Price’ column of the data set, and separated it by the States NY, CA and TX and the rest of the country removing these three states , using “big\_states = df.Price[df.State.isin(['CA', 'TX', 'NY'])]” and “small\_states = df.Price[df.State != 'CA'][df.State != 'TX'][df.State != 'NY']”. For my analysis I know that we only have a sample for the complete Trulia Dataset, and thus I will use the T score instead of the Z score. To complete this process I used the function “stats.ttest\_ind”.

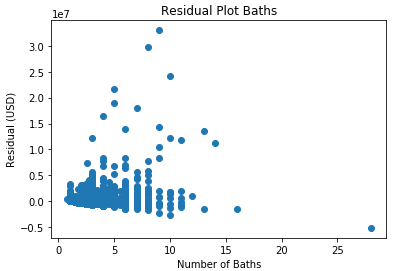
After running our formula, the results we received was tstat = 24.33266350950908

, with a p value of 1.596 e^-129. Interpreting our results, due to the small p value, we will reject the null hypothesis here that the cost is the same in the large states of New York, California, and Texas compared to the rest of the country. In terms of our analysis we can conclusively say that there is a difference in cost between buying a home in New York, California and Texas compared to the rest of the country.

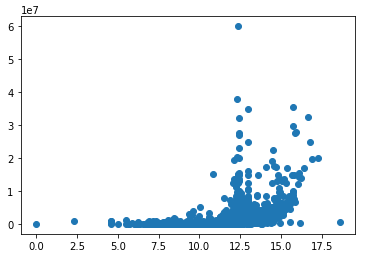
**Results and In-Depth Analysis**

In order to obtain as much data as possible for our dataset, first I ran df.get\_dummies, on our States variable (all 37) to obtain an additional 37 dummy columns for the dataset. Next I employed the isnull().astype(int) function on all of the columns to create new columns, with a 1 for having an null and 0 if no nulls. Once this was settled, I ran through each of the variables and tested them using linear regression to find how strongly they would relate to our target variable “Price”. Running through all of my variables (Beds, Bath, Building Age, Last Sold For, Last Tax Assessment, Years Since LT (Last Tax), Days on Trulia, Sqr Ft, Longitude, Latitude), I found that the variable with the highest R2 score when plotted in LInear Regression against the Price, was the Bath variable with a test score of 0.1899, and the lowest was Building Age at 0.0015. Performing EDA on these 7 variables, the Beds and Bath variable showed a possible negative (slight) linear relationship in the residual graphs, but no such pattern in the regular one, as shown below.





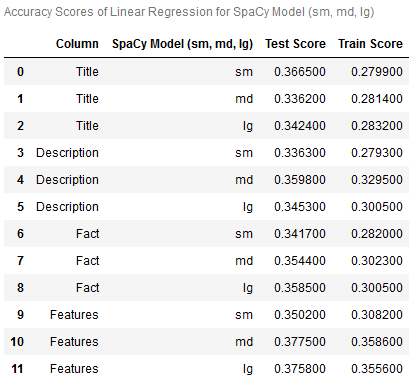
We also see a slight positive Linear Regression relationship in the Log transformations for the Last Tax Assessment and Last Sold For variables. I did suspect that there might be strong correlation here, considering that a homes last sold price, and last tax assessment can be a very strong indicator of a homes value. Demographics might change, and housing markets do evolve, but to my surprise there isn’t an overly strong relationship when I plotted them out. I did find a strong linear relationship when I applied the Log transformation as shown in the graph below.



After completing this process, the next step I took was to run an Multivariate Linear Regression model with the selected variables. This is the most basic Machine Learning model and it will give us some valuable insights. Without processing any of the text that we were given, this model gave an R2 score of 0.331 for the Test Set and 0.265 for the Training Set. This shows that the model as it is currently constructed is not ideal for predicting price, but we may be able to improve upon it.

The next step that I took was to process the text data in the dataset. There were 4 main columns/variables that contained potentially useful NLP data, (Title, Description, Facts, Features). To be clear, in terms of text data 3 out of these variables contained normal text data, with the exception of the Features variable, which seems as it contained categorical data. My initial thought was to drop this variable in favor of the other three, but I figured it would be best to perform some tests before making a final choice. To help me perform this analysis I employed the SpaCy NLP model.

In the SpaCy library, it is important to choose the size that we are using. SpaCy offers three different sizes (sm, md, lg) and each required different computation powers and provides different results. In order to find the best results for my model, I concatenated each variable separately to a copy of my original data set, and then ran it with my Linear Regression Model. The size that performed best were then combined with my original dataset to create one final dataset, that I would use moving forward.



As you can see from the table above, from the Title variable the sm size, performed best with an 0.367 R2 score, from the Description variable the md size performed best with a 0.359 R2 score, from the Fact variable lg size performed best with 0.359 and from the Features variable the md size performed best with a score of 0.378. From these tests, I can conclude that the Features variable does in fact contain useful NLP data due to the improvement in my models Linear Regression Test Score.

After concating all 4 variables into my base dataset, the new dataset has 30006 observations, with 1257 variables. Running a Multivariate Regression with this new dataset, gave an Test R2 score of 0.392 and a Train R2 score of 0.424. The R2 score did increase by a modest 0.06 but I had expected a bigger increase. After running our new model, it is evident that the Linear Regression model will not allow us to adequately predict the price.

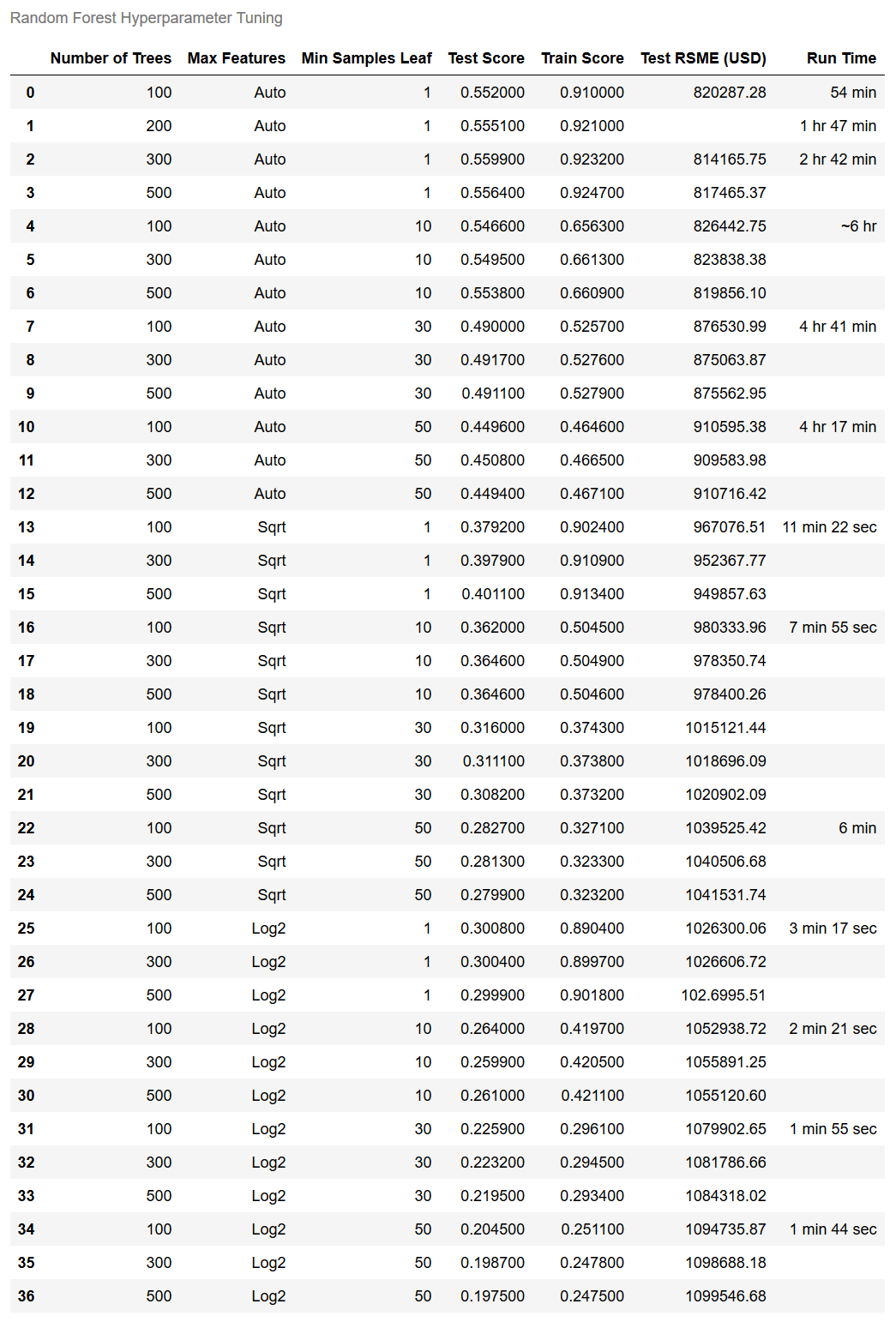
Moving on my next choice was to run a Random Tree Regression model on the data. To begin, I assigned my X and y values, and ran a train\_test\_split with a test size of 0.2 and a random state of 0.

When I began the model my initial plan was to run it at 100, 500 and 1000 trees, max features at auto, sqrt, log2, and min sample leafs at 10, 30, 50. The reasoning for this is that in a prior model I had run these trees and max features with no issues, and from what I had found they were good parameters to tune to measure any changes in R2 score . Min Samples Leaf were chosen, as from my research on the Random Forest Regressor documentation, it seems as if this variable can smooth out regression and potentially reduce the “noise” in the model.

However, I was forced to adjust the number of trees chosen, as I found that for every 100 trees it took the model approximately 54 minutes to run it in the base Random Forest Regression model (no tuning). This was most likely due to the size of the dataset, as the new dataframe with all the new data contained 30006 observations and 1257 variables. As a result, I readjusted my models to run at 100, 300 and 500 trees. In the hyperparameter tuning table below, you will see that I added the run time as one of the columns.

After running all 36 Random Forest Regression models, I found that while changing the Max Features and increasing the Min Samples Leaf, made my model run significantly faster, (base model running 100, 300, 500 trees took 6 hours compared to 1 min 44 seconds running 100, 300, 500 trees at Max Features = Log2, and Min Samples Leaf =50) it actually hampered the R2 score of my model. You can actually see a gradual decline in my Hyperparameter Tuning table below, as the further down it goes (more tuning and faster runtime) the less accurate it gets.

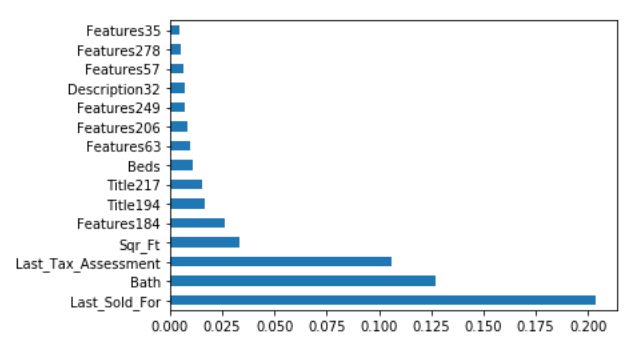
From the data I have the Random Forest Regression model performed best with 300 trees, Max Features being set to Auto, and Min Samples Leaf set to 30 with an R2 score of 0.4917 on the test set and 0.5276 on the training set. This is a very respectable score and it shows that our model needed all of the data we generated to perform stronger. By tuning the number of trees and the Min Samples Leaf, I was able to find the ideal parameters that maximized the R2 score and resolve any overfitting problems.



**Conclusion**

After completing our analysis, I can conclude that this dataset was able to produce a very respectacle R2 score ( 0.4917 on the test set and 0.5276 on the training set), but is not ideal for this purpose. By using the .coef\_ and .feature\_importance\_ features on the Linear Regression and Random Forest Regression models I was able to find some very interesting insights comparing our variables to the target variable Price.

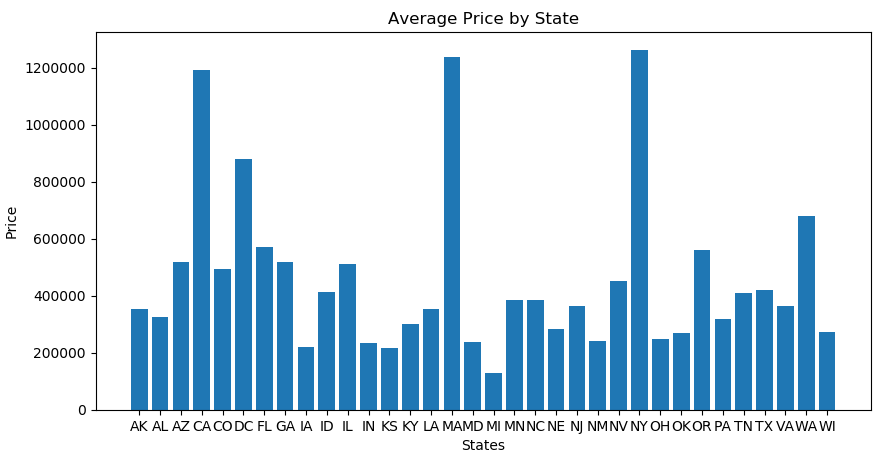
Looking at the graph below, you can see that for our Random Forest Regression, the variables Last\_Sold\_For, Bath, Last\_Tax\_Assesment and Sqr\_Ft, had a big hand in predicting the target variable price. So when deciding to buy or sell a home, homeowners and buyers want to emphasize these variables in their decisions.



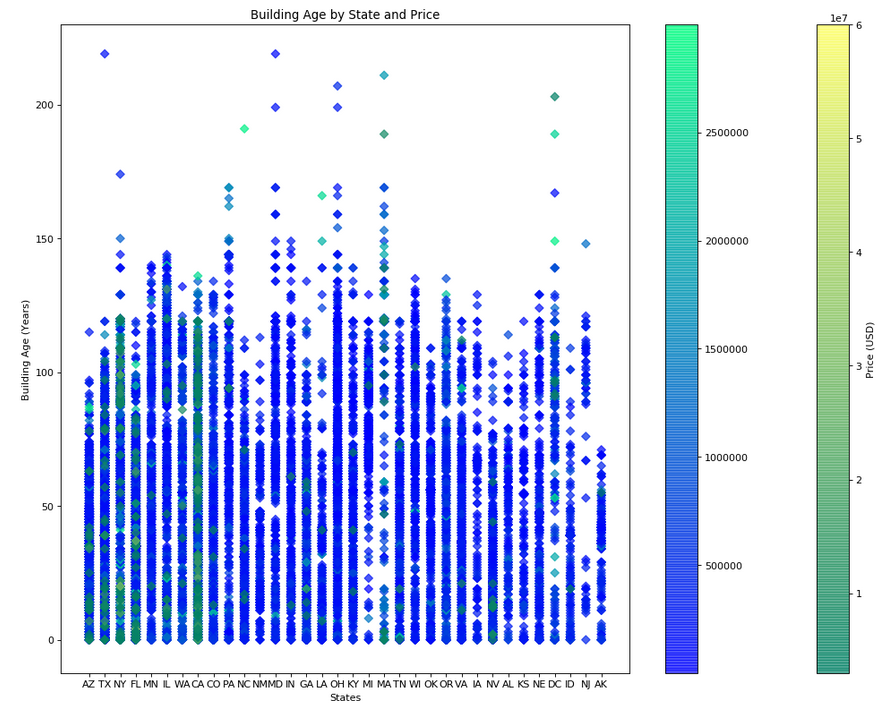
Reviewing the information for the .coef\_ variable, there is a wealth of valuable information. Looking at some of the variables I mentioned above, we can see that the Bath variable had a .coef\_ of 206,276.46 USD, meaning that for every bathroom a listing has, you can expect an 206,276.46 USD increase in cost for that home.

For the Sqr\_Ft variable, you can see that for every square foot of the unit there is an increase of 0.08 USD. So generally the bigger the unit the more expensive it will be.

While I was performing EDA on the dataset I came across some very interesting insights. By plotting the States variable by the average price of each individual state, you can see that the most expensive homes reside in the states California, Massachusetts, New York, Washington state, and Washington DC. This was a bit surprising, because during our statistical analysis we found that the three states with the highest GDP were Texas, New York, and California so naturally it was assumed that they would have the higher prices.



However, the most interesting insights that I found are in the graph below where I plotted the States by the Building Age, and mapped it with the Price variable. When you look at the individual states, you will find that for some states like North Carolina, Louisiana and Pennsylvania will have their home values on a linear pattern, with the older homes generally retaining a higher value than newer ones. Whereas, for other states like California, New York, and Florida, regardless of when the home was built, it will have virtually no effect on the price of the listing. It could be a newly built home, or one hundreds of years old, there is simply no correlation between the Price and Building Age



In conclusion, after employing different regression models, and hypertuning several parameters, I have determined that our dataset is able to predict price in a fairly decent capability and I was able to find some very insightful data regarding Trulia home listings in the US, and I do hope to be able to apply this information, should I choose to buy or sell a home in the near future.