**Consolidated Report Capstone 1 - Jun Chen**

**Proposal**

AirBNB NYC - For this AirBNB data set for the state of New York, we will be examining a data set of 16 columns by 48896 rows. The data in question originates from insideairbnb.com which is an “independent, non-commercial set of tools and data that allows you to explore how Airbnb is really being used in cities around the world.” I had discovered this data while browsing Kaggle, as I personally have a passion for traveling and this peaked my interest. The data published by this site is compiled directly from AirBnB.

As one of the core attractions of the US, New York is visited by millions of tourists every year. My audience would be travelers who are interested in visiting New York state, and trying to find housing during their visit. From this data set I hope to be able to accurately predict the price for an AirBnB in the state. My main deliverable would most likely be a notebook, consisting of the code used for the project, as well as slides and graphs explaining the project itself.

Categorical variables in the data set include Neighbourhood(City), Neighbourhood, room\_type.

Continuous variables include, host\_name (length of name), latitude and longitude, price, min nights, min reviews, reviews per month, number of listings of host listings, availability out of 365 days.

Based on the data provided, I’m hoping to be able to provide an accurate estimate of the average price for an AirBNB in New York, based on variables like the city, it’s given neighborhood, the room type, the minimum number of stays and the possibly its availability out of 365 days a year. Other possible target values could be the average availability of a certain room type, given a price range, and its neighborhood

I will try to obtain this target variable through using the tools taught to me through this course, using data wrangling, and analysis.

Columns to exclude in this dataset would include those that are unrelated to our analysis, which are mainly the listing ID and the Host ID. These two columns mainly serves as ids in internal Airbnb data and is usually not public

**Data Wrangling**

For this portion of the course, I’m running some data wrangling to prepare and clean the data for further analysis.

To start off with, I imported numpy, pandas and matplotlib.pyplot and read the csv dataset, which contained 48896 rows and 16 columns (note: found on Kraggle, but sourced form an AirBnB dataset/project site), into the Jupyter Notebook environment, and inspected it visually. Something I did notice is that there were a large number of listings that had NaN, mainly in the last\_review and reviews\_per\_month columns. To remedy this, we replace the NaN in reviews per month with 0 to enter a valid value, and for the last\_review column I employed dropna(method=’ffill’) to replace the NaT with the date of the previous row.

Next, using the pandas function drop\_duplicates, I checked to remove any listings that may have been duplicated in the dataset. After running the function, there were 0 rows removed.

In order to check the dataset for outliers, I ran the .describe() pandas function, and set up a boxplot to visually inspect the data. From the information gathered, there were a few substantial outliers, in the Brooklyn, Manhattan and Queens area. This may very well be caused by the fact that these cities are bigger, and their demographics may be larger. AirBnB allows a wide variety of listings, including but not limited to penthouses, retreats, and boats among other things. I chose not to remove these outliers because they are valid data, although this is something to keep in mind when running further analysis down the line.

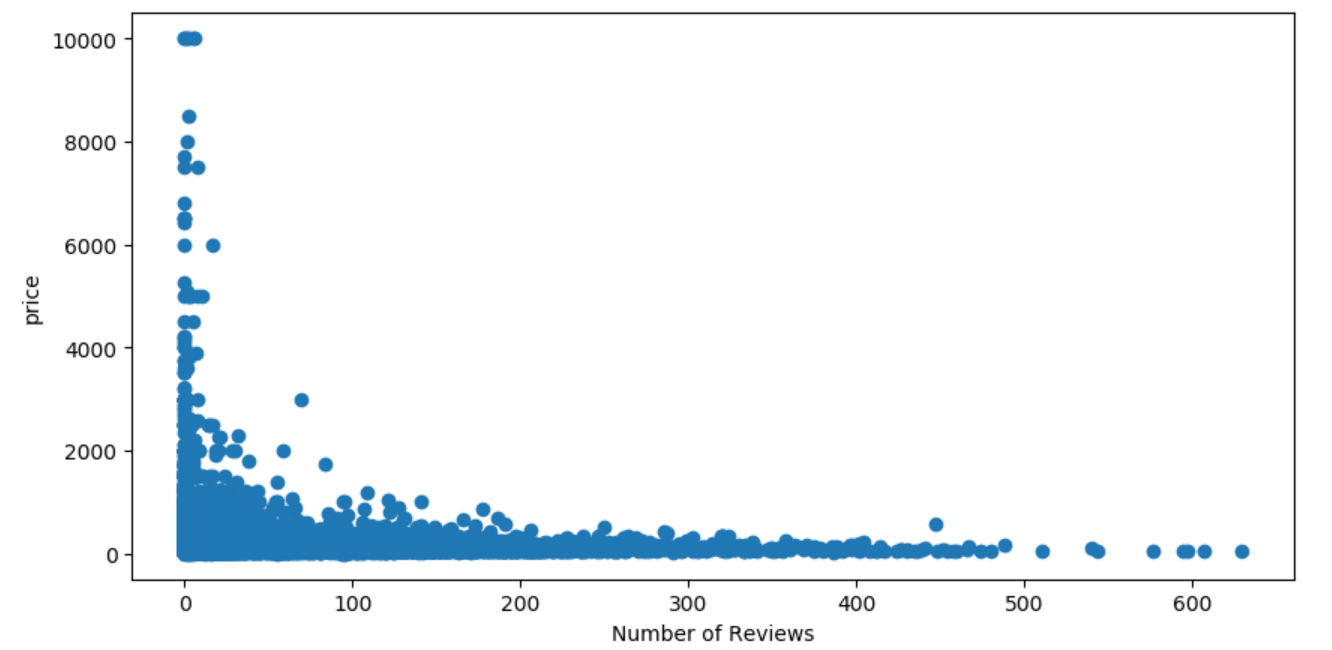
Moving on, using the .info function, I inspected my dataset, and most of the data was consistent (0 NaN after the prior cleaning) with the exception of the name and host\_name columns, which had had values inconsistent with the rest of the data. Non-Nulls respectively. Based on my knowledge of AirBnB’s setup and the inspecting the data earlier, the inconsistency with these two columns could've been caused by AirBnB users with multiple listings, and using the same template descriptions from the AirBnB site. However to check this, the function dropna was used to remove any columns that had a NaN value. After this was done, approximately, 22 rows were removed. Running .info again, shows that by removing the NaNs this fixed the inconsistency, and all columns now show 48821 Non-Nulls. As a last step, I ran the function “assert df.notnull().all().all()” to ensure that all data in my dataset is non null.

**Initial Findings**

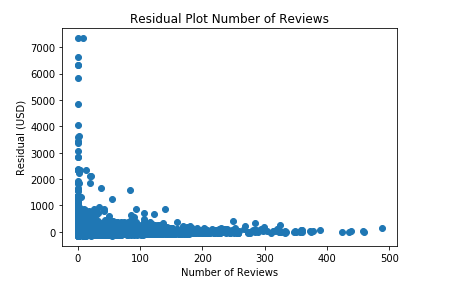
As part of my first capstone project, I’m working on exploring my dataset and finding interesting results. My dataset contains information from AirBnB about listings in New York City and contains 48858 rows and 18 columns. The goal of this project is to accurately predict the Price for a listing based on the featured variables on the dataset..

To start off with, I imported my dataset from the ‘Wranged\_df.csv” and I used “df.info()” to obtain a breakdown of my data set, and their datatype (Dtype, cell 5). Based, on the summary provided, I decided to focus on the featured variables, “neighbourhood\_group”, “neighbourhood”, “latitude”, “longitude”, “minimum\_nights”, “number of reviews”, “reviews\_per\_month”, “calculated\_host\_listings\_count”, and “availability\_365”.

The first variable we looked at was “number\_of\_reviews”. I plotted it against the “price” variable, using a scatter and bar plot (cells 6-8), and from the shape of the graph, there is a very steep but apparently negative relationship, in the form of a curve.

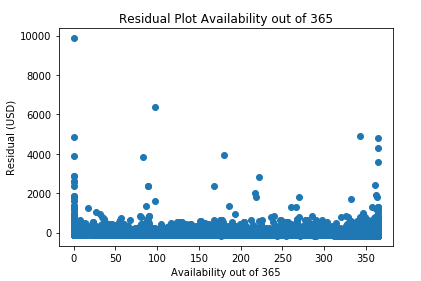


In order to take a more indepth look into the “Number of Reviews” I plotted the residual plot for this variable. Looking at it below, its very similar to the original reviews vs price chart, and shows a possible linear relationship.

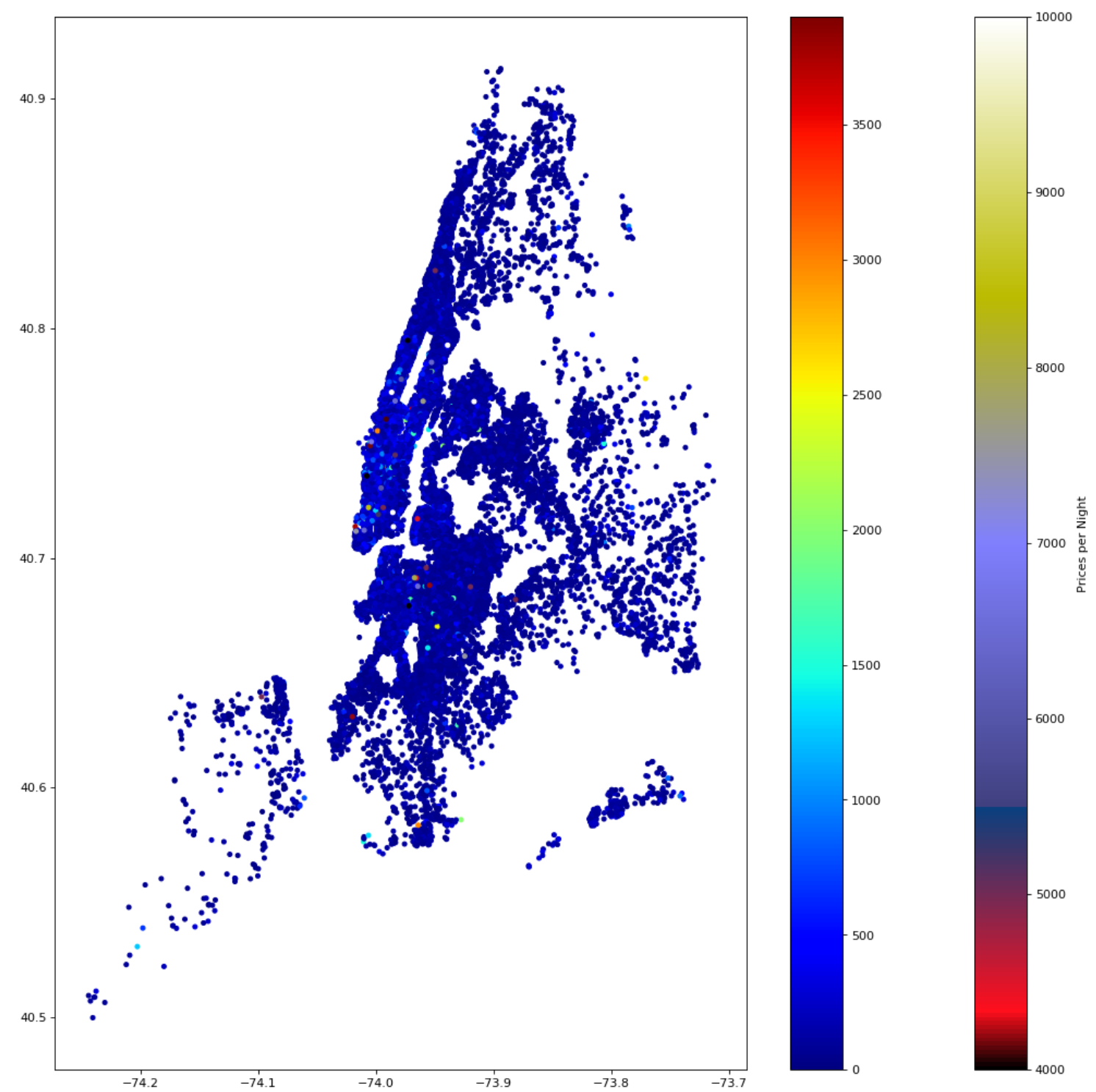


Moving on, the next variable I looked at was, the “calculated\_host\_listings\_count”, which is a measure how how many listings that given host had. Plotting a scatter plot between this and ‘price’ (cell 9) does not show a clear pattern.

The next variable we used was the ‘availability\_365’ column. This variable is a breakdown of how often an AirBnB was available out of the year (total number of days out of possible 365). Plotting this data against our ‘price’ variable (cell 11), proved inconclusive, as there were not clear patterns or relationships between the two. Plotting the residual plot for this variable, did not show any new patterns as shown below.



Geographically, New York City is broken down into 5 boroughs, or in this case “neighbourhood\_groups”. Using the data provided, I plotted three separate scatter plots, one for ‘latitude”, ‘longitude’ and ‘neighbourhood\_group’ (cells 12-14). Based on the graphs that were generated, there wasn’t a clear linear relationship between these variables and the price. However, what I did discover was that the closer you get to central NYC, (mainly Manhattan and Brooklyn) the higher the prices tend to trend. While this may not help our Linear Regression model, it is an interesting data fact, and relevant to potential visitors. I’ve included the heatmap of NYC’s boroughs for reference below.



Moving on, the next variable that I plotted and compared to ‘price’ is our ‘minimum\_nights’ variable, which is an indicator of the minimum nights, you must stay at the unit to book it. Upon, plotting this scatter graph (cell 16), it became clear that there is an issue with the variable in this data. As we all know, on the planet Earth, there are only 365 days in a year, but in our data for “minimum\_nights”, there were figures going past 1200 nights. This may be an issue with how AirBnB organizes this data, but from what we can gather there are no relevant patterns, and the data is suspect.

And last but not least, we’ve arrived at the last variable for our current exploration “reviews\_per\_month”. Plotting this data vs ‘price’, shows a negative linear relationship, with a curve, similar to our results from ‘number\_of\_reviews’. This result merits further review and exploration.

Running the “.describe(include=’all’)” function on my data has also revealed some very interesting insights to the data. Looking at our summary, on average for an AirBnb in NYC the cost is 152.74 USD, with a Standard Deviation of 240.23 USD. This shows that while the vast majority of the listings are in the lower price range between 69 and 175 USD (the 25th, and 75th quartile ranges), we can expect to see a significant shift in price between different neighbourhoods.

Taking a look at our ‘neighbourhood’ column, we can see that there are 221 individual neighbourhoods shared between the 5 boroughs. And comparing the names of our host, it looks like there are 11450 different names over 48858 listing, with the name “Michael” being most popular with a count of 417, so more likely than not the next time you rent an AirBnB in NYC, your host will be named Michael.

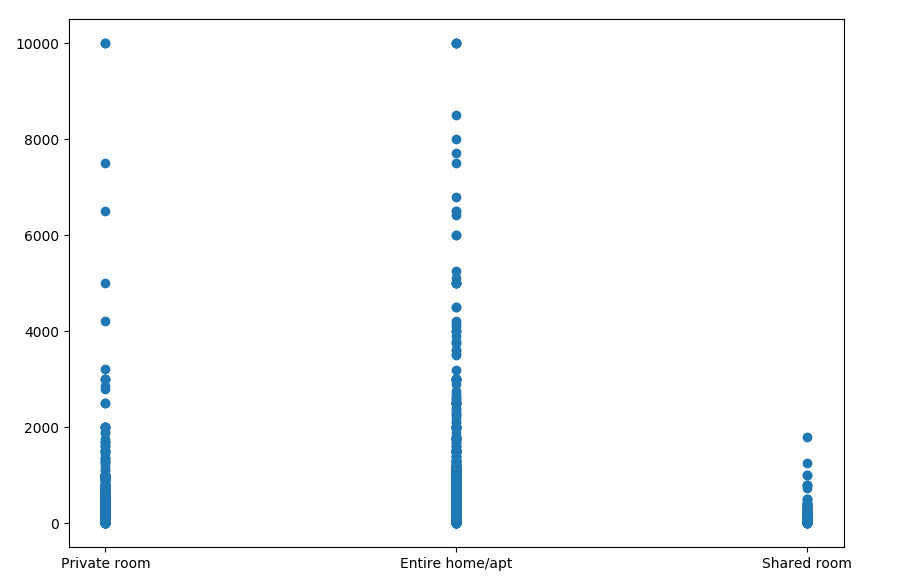
Performing EDA, between the “neighbourhood\_groups” and “price” provided some very interesting insights, when I applied a colorbar with our availability out of a 365 days dataset. Per the graph below, Manhattan and Queens, have the most AirBnB listings, that have high availability (150+). So if by chance, a last minute business trip needs to be made to NYC for example, it would serve you best to look in Manhattan and Queens for an AirBnB. Alternatively, if you are a prospective AirBnB unit owner, you can consider taking advantage of the fact that Brooklyn, Staten Island and the Bronx, have lower availability throughout the year. If you have an existing unit in those neighbourhoods, you may want to increase your own availability or consider purchasing a unit in these neighbourhoods.

Over the course of our data exploration, we have learned a number of interesting facts from the data. This includes, that prices will trend higher, the closer you are to the heart of NYC, and the main variables we should be focusing on are ‘number\_of\_reviews’ and ‘reviews\_per\_month’.

**Statistical Analysis**

As part of my project on predicting AirBNB prices in NYC, I went through the data and explored a vast number of variables and potential connections. One of the more interesting statistics that stood out to me during this process was the price in NYC of renting a room vs a whole apt. Based on our everyday assumptions, renting out an apartment should be more expensive, due to the size of the unit, and the privacy it provides, whereas renting a room you will have less space and you would need to share a public space with co tenants. Of course this isn’t as simple for a city like NYC, where particular boroughs are more expensive (please see attached notebook for reference).

For our statistical review, we will set the null hypothesis (Ho) as there is no difference between the price for a whole apartment and a room in NYC, whereas our alternative hypothesis (H1) is that renting a room in NYC is less expensive. Note, in our data set there is also a 3rd type of listing where you would rent to share a room with more tenants. For the purpose of our statistical analysis, I will omit this because there were only 1159 shared room listings, and from our EDA none of these listings had a price of more than 2000 USD, compared to our apartment listings (25393 listings), and room listings (22306 listings) which were more spread out in their price. Please see the graph below for reference.



To complete the analysis, i pulled data from the ‘Price’ column of the data set, and separated it by ‘Private room’ and ‘Entire home/apt’ , using “l = df.price[df.room\_type == 'Private room']

m = df.price[df.room\_type == 'Entire home/apt']”. As we have all the data from Airbnb, circa 2017 - 2019 for the NYC area, it is assumed that we have a full population, and thus for my analysis, I will use the Z score instead of the T score. For the next step, I had to do some research and find a python function that will allow me to compare the two data sets. It ultimately came down to using ‘stats.ztest()’ or ‘stats.ComepareMeans.ztest\_ind()’. Both of these functions allow me to compare the means of the two data sets, but ztest assumes that the standard deviation is the same, whereas ztest\_ind, does not, so it was chosen as the method used.

After running our formula, the results we received was tstat = 58.631293966982135, with a p value of 0.0. Interpreting our results, due to the small p value, we will reject the null hypothesis here that the cost is the same in NYC for whole apartments and single rooms. In terms of our analysis we can conclusively say that there is a difference in cost between renting a whole apartment and renting a single room.

**Approach**

For this project the goal is to predict the price for an AirBnB stay in NYC, and I will attempt to do so using Linear Regression and Random Forest Regression model. The main reason for this choice is that during our EDA, it was evident that there may be a relationship between some of the continuous variables and our target variables (Price). Linear Regression is a simple but very powerful model that will allow us to examine these relationships and determine if there is in fact a real connection. The plan is to use the Random Forest Regression model to check the results of our Linear Regression model and ensure that no mistakes occurred.

With regards to the hyperparameter tuning, plan on using a mixed approach between using the GridSearchCV function, and manual iterative tuning. GridSearchCV allows us to obtain the best parameter for most models, but depending on the resources available to us, this isn’t always feasible, which is where the manual tuning comes in.

**Results and In-Depth Analysis**

My capstone 1 project revolves around using machine learning techniques to predict the price for an overnight stay for an AirBnB in New York City. Our dataset in question was found on Kaggle, but it originated from insideairbnb.com which is an independent, non-commercial set of tools and data that allows for exploration of AirBnB’s data, by analyzing and aggregating publicly available information from AirBnB’s website.

To start off, steps were taken to clean up the data set, removing any “NaNs”, and filling in any missing data. After this was completed we imported the dataset from the ‘Wranged\_df.csv” and I used “df.info()” to obtain a breakdown of my data set, and their datatype. Based, on the summary provided, and after performing EDA on these continuous variables, I decided to focus on the continuous variables, “neighbourhood\_group”, “neighbourhood”, “latitude”, “longitude”, “minimum\_nights”, “number of reviews”, “reviews\_per\_month”, “calculated\_host\_listings\_count”, last\_review, and “availability\_365”. For the variable “last\_review”, it was a time series, so to use it as part of my model, it was converted to show days since the last review was posted.

To clarify, my criteria for these continuous variables were that the EDA had to show a clear possible pattern between them, and my target variable (price).

Based on my EDA for each of the variables, it was determined that the best model to try first, would be Linear Regression. In order to test how accurate our predictions were for each variable, Linear Regression was ran on each of the independent continuous variables, and were compared with the actual price using the .score function. After running this for all variables, the highest score we got back for each individual variable was ~ 0.0022, which indicated that our continuous variables were not a strong predictor of price. The next step was to perform an Multivariate Linear Regression, by using all of our continuous variables. Using this method, my .score increased significantly, with an accuracy as high as ~ 0.054 (note: I ran this 10 separate times, with no set random seed, to see how significant our results could vary). This is a noticeable improvement over the independent variables, but still not statistically significant.

In my bid to find more connections between the continuous variables, and our target variables, I decided to apply the np.log function to transform the continuous variables and run the Multivariate Linear Regression again. Our graphs did show a change, but the accuracy of our model actually dropped to ~0.014.

To improve my model, the next step was to look at alternative regression models. As such, I trained and fitted, my data using Ridge, Lasso, and Elastic Net models. Comparing the results I obtained (I ran Ridge, Lasso, Elastic Net, along with Multivariate Linear Regression on the same train\_test\_split data set), Ridge regression was the only model that showed a possible improvement. Without optimizing our Ridge model, after 10 trials of both Ridge Regression and Multivariate Linear Regression (using the same Train, Test, Splitted data), there was only an 1.74 e -7 improvement between Ridge and Multivariate Linear Regression. Using GridSearchCV to fully hypertune our Ridge model, resulted in an improvement of 6.4 e -6, which is insignificant. Please see the table below for the parameter settings.

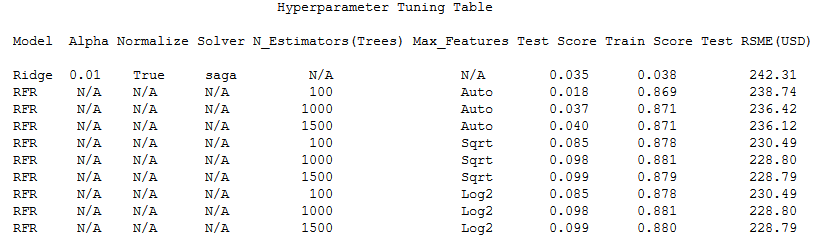
After exhausting the possibilities of our Linear Regression model, it became evident that Linear Regression was not the right model to find a connection between our continuous variables, and our target variable, price. To ensure that the Linear Regression model was not the cause for the poor prediction, it was decided that the next step would be to try a Random Forest predictor on our data.

To set up the model, I read in our cleaned dataset, and assigned our target variable and independent variables. Based on my earlier results, I knew that combining our independent continuous variable allowed us to get the best results, so they were assigned to the “collection” variable.

After, I applied “train\_test\_split” to the data with “test\_size” = . 0.2 and an “random\_state” of 0. To test the Random Forest Regressor model, I decided to focus on 100, 1000 and 1500 trees as the main parameters, with the reasoning being that 100 trees is a good entry level test, with an increase to 1000, and eventually 1500 trees to see if we can find an noticeable improvement. Running the Random Forest Regressor (no tuning) with 100 trees, netted an accuracy of 0.0185. Escalating at 1000 and 1500 trees did show a noticeable improvement to a score of 0.0374 and 0.0398 respectively. To further tune this model, I explored other parameters of the Random Forest Regressor that I could tune. Based on my research I decided to focus on the Max Features parameters of Random Forest Regressor. The Max Features parameter focuses on how many features the tree considers before employing its next split, with the default being “auto”. As seen on my Hyperparameter Tuning Table below, changing Max Features to “Sqrt” and “Log2” resulted in a significant jump in the accuracy score for all of the trees.

Due to the computing power necessary to increase the number of trees in the model (my computer almost crashed at 1500 trees), and the minimum improvement in the score between 1000 trees, and 1500 trees (0.001), I concluded that this adequately represented the accuracy of the Random Forest Regression model. And on another note, I did also consider changing the “criterion” parameter from ‘mse” (default), to ‘mae’, which would’ve changed the criteria on how the quality of a split in the tree is determined. But due to the resources needed using ‘mae’ compared to ‘mse”, (it took 15 minutes to run my three ‘mse’ tree models), compared to the ‘mae’ models (I waited over 2 hours and it did not even finish the first model I had set up), it was not possible to continue further.

There were noticeable improvements, comparing my tuned Random Forest Regression model to the Linear Regression model, but the accuracy overall is very low for any prediction model. As such I can conclude that our data set is not fit for predicting the Price of an AirBnB in NYC.



**Insights from Ridge Regression**

Although this data set was not a great match for price prediction, there were some very interesting observations. By applying the “.coef\_” function on our Ridge Regression model, you can see how our continuous variables were weighted, and what effect they had on the target variable, price. From this we can draw some insight.

The continuous variable “number of reviews” had a coefficient of approximately -0.18, meaning that for every review the listing has, you can expect the price of the listing to drop by approximately 18 cents. This matches some of the initial finds we found earlier, as listings with more visitors, tends to have lower prices. Prospective guests can filter out by this variable to find cheaper listings.

Another interesting continuous variable is “availability\_365”. This variable counts how many days an AirBnB listing is available out of 365 days (a full year). Based on the coefficients, we can see that for every day an AirBnB is available to be rented out, out of 365 days, there is an increase of approximately 0.196 USD. This is very useful information to AirBnB hosts as this can help them determine how to choose the best price for their listings.

In conclusion, after employing different regression models, and hypertuning several parameters, I have determined that our dataset is not fit to predict the price of an AirBnB listing in NYC. I was able to find some very insightful data regarding AirBnB’s in NYC, and I do hope to be able to apply this information, should I choose to travel to New York City in the near future.