Springboard Data Science Track
Capstone Project-1
Final Report

1. Problem Statement:

1.1. Purpose:

This dataset is going to be used to create a model that explores pricing metrics in NYC and predicts rental prices for the upcoming period.

1.2. Targeted stakeholders:

-NYC Housing Authority

-Tourism Tour Organizations while accounting their tour costs

-Tourists planning to visit NYC

-Airbnb App users

-Airbnb Hosts And Potential Hosts while balancing their rent prices

-Airbnb Application Improvers

-Other Hosting Application Member Users(such as Booking.com)

-NewBie Hosting Application Startup Investors.

1.3. Business Outcome:

This project is going to be useful for these beneficiaries because:

-Airbnb users as bookers can see the forecast price in terms of location and seasons over the year before reserving their rental place.

-Airbnb users as hosts are able to see appropriate listing prices by checking this model, so they will not list the place for under or over price.

-Airbnb application creators can check this model to improve their recommendation system for bookers.

-Airbnb investors will also have the advance derived from mutualist benefits between hosts and bookers.

-Other hosting app users can check this model for balancing their rental prices.

2. Dataset:

This open-source data set includes 16 columns and 48895 rows providing Airbnb listing information and metrics for hosts, places and geographical features in NYC, NY area between the years 2008 and 2019.

The data is sourced from the Inside Airbnb website http://insideairbnb.com/get-the-data.html which hosts publicly available data from the Airbnb website.

2.1. Cleaning Steps

I have dropped 'host_name' column that could be unethical to use for future data exploration and predictions.

```
data.drop(['host_name'], axis=1, inplace=True)
```

2.2. Missing Values

I checked the columns containing null items

```
data.isnull().sum()
```

I filled them with'0'

```
data.fillna({'name':0}, inplace=True)**unlogical
data.fillna({'last_review':0}, inplace=True)**date time
data.fillna({'reviews_per_month':0}, inplace=True)
```

I controlled and made sure there is no missing items anymore.

```
data.isnull().sum()
```

2.3. Outliers

Because of the fact that Min price per night: $0, Max price per night: $10000, there were quite observable outliers.

First, I have detected and dropped listings whose price is 0.

Because the first step was not enough, I have detected other anomalies by using mean() and std() and also sns.boxplot.

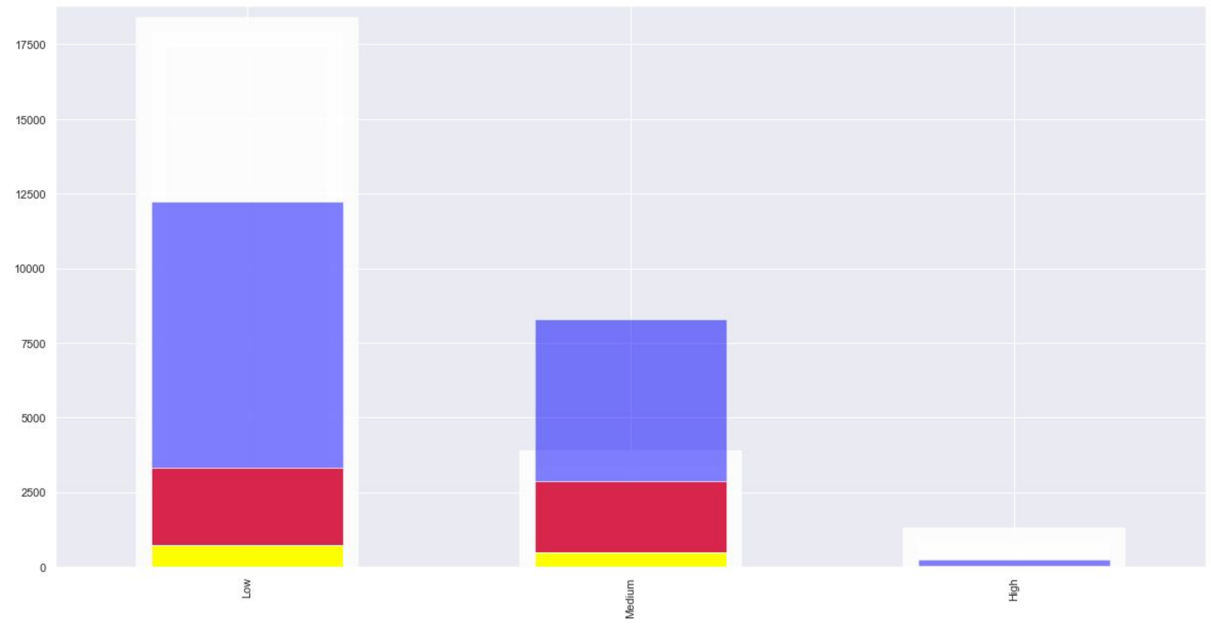
When extra expensive listings matter while comparing the boroughs, I used logged prices instead of considering those as an outlier and removing them. Because those listings are real and important data to analyze.

In all other cases, I ignored them and continue to exploratory analysis at reasonable prices.

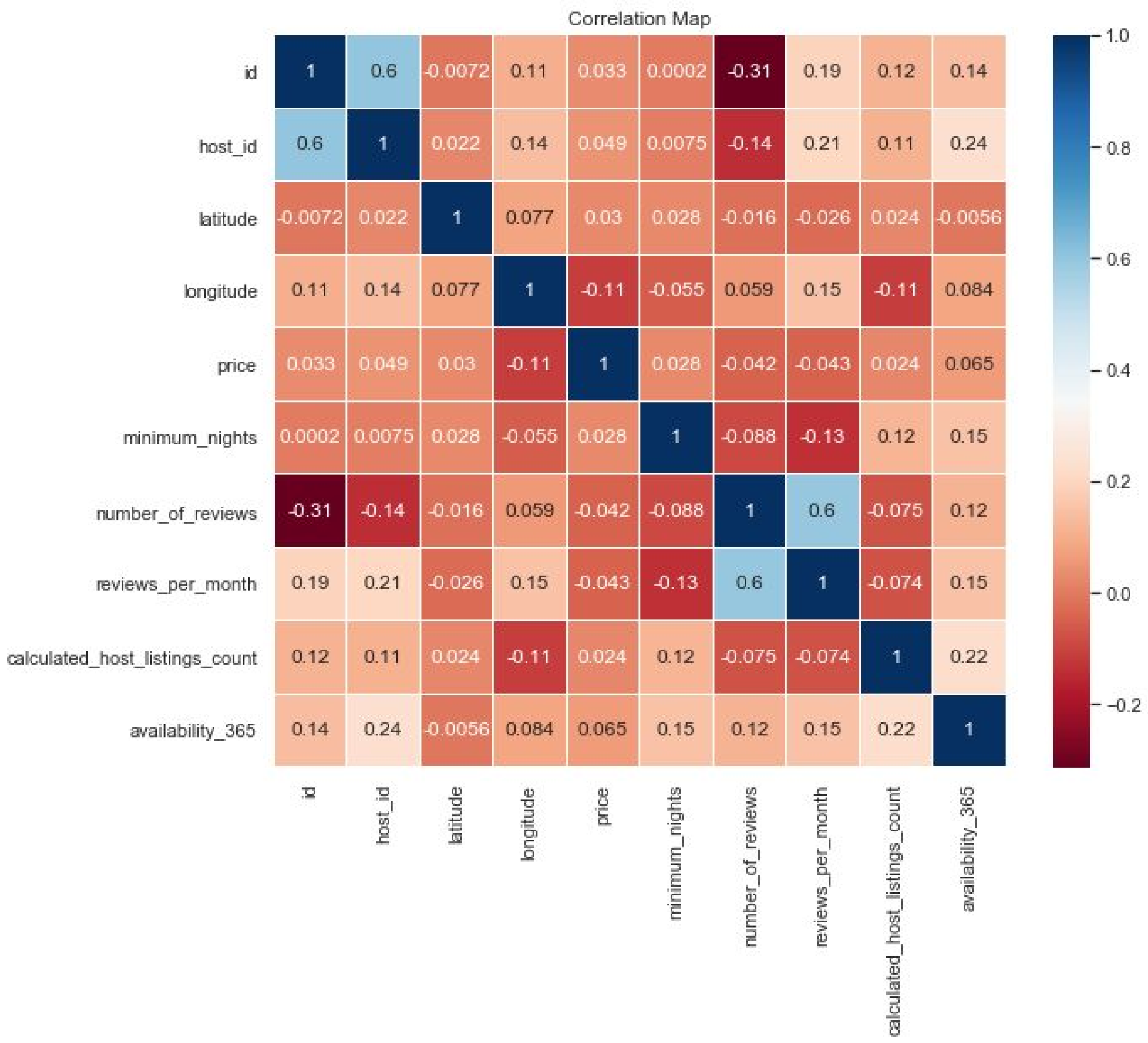
3. Data Exploring

I categorize listing prices as High, Medium, or Low and according to results around 325000 listing is in the medium category while around 15000 listings is priced lower and 1000 listing is higher.

This plot shows those categories according to the distribution of boroughs.



I have created a correlation map: overall, all correlation between column categories are weak.



I have created a WordMap and According to this WordMap, 'Private Room', 'Apartment', and 'NYC' are the top 3 words used in listing title on AirBNB.

To answer the question What is the average price per night in NYC?

First, I simply calculated the mean of all listing prices.

Secondly, I calculated the average price according to top reviewed listings to eliminate unpopular/inactivated listings.

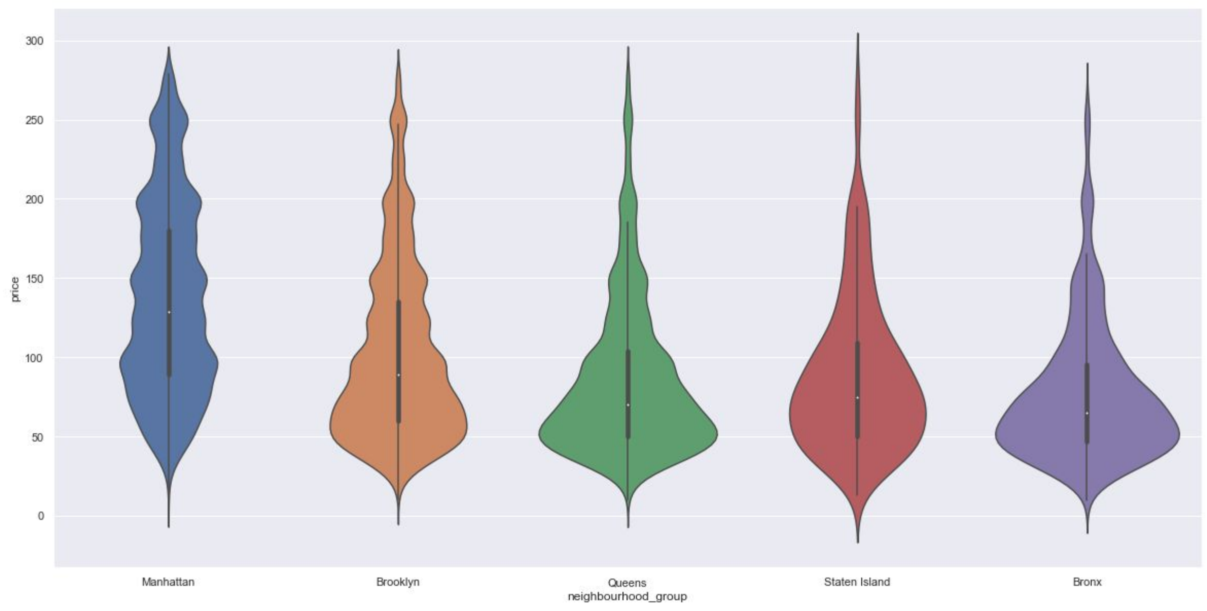
Thirdly, I calculated the average price according to cleaned data from outliers.

The result: Average price per night is $ 113.54639354600214

To calculate the price disperse according to 5 boroughs: 'Manhattan', 'Brooklyn', 'Queens', 'Staten Island', 'Bronx'.

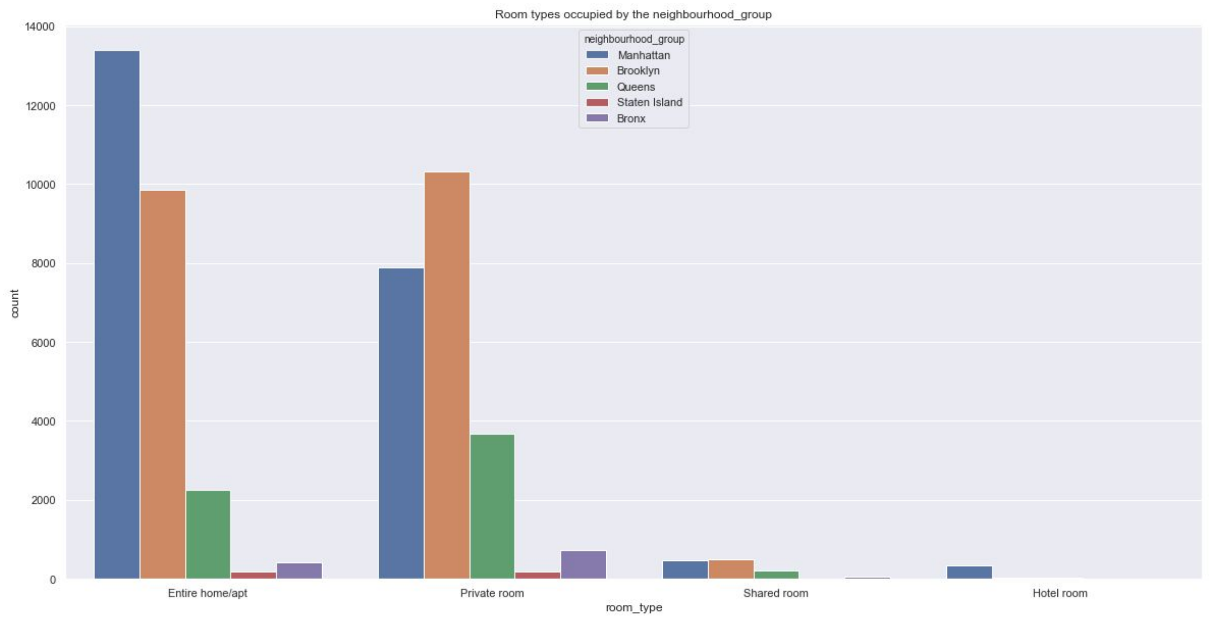
1) To do this, first I divided the data into 5 sub data. Then I used sns.stripplot.

2) Because of these sub-data groups have outliers, I used the price_new data to see closely the disperse via sns.violinplot.



I found the disperse of room types related to boroughs:

Entire home/apt	Private room	Shared room	Hotel room	
Manhattan	13381.0	7877.0	470.0	337.0
Brooklyn	9863.0	10326.0	497.0	43.0
Queens	2264.0	3679.0	216.0	37.0
Staten Island	194.0	179.0	5.0	NaN
Bronx	431.0	718.0	66.0	NaN



Hotels are listed on Airbnb mostly for Manhattan which has also the most amount of Entire home-Apt listings. Brooklyn's listings are generally based on Private Room. While shared rooms and hotel rooms are the least popular listings, the entire room and private rooms listings significantly have higher demand.
As a last part of exploration, I have created maps that show the distribution of room types according to longitude and latitude and also examined them borough by borough.

4. Statistics

To do an analysis of variance, I am going to use ANOVA instead of t-test because there are more than 2(5) boroughs to compare their means of prices

The P-value obtained from ANOVA analysis is significant (P<0.05), and therefore, we conclude that there are significant differences among treatments.

From ANOVA analysis, we know that treatment differences are statistically significant, but ANOVA does not tell which treatments are significantly different from each other. To know the pairs of significant different treatments, we will perform multiple pairwise comparison (Post-hoc comparison) analysis using Tukey HSD test.

Multiple Comparison of Means - Tukey HSD, FWER=0.05

```
=====================================================
group1 group2 meandiff p-adj    lower    upper   reject
-----------------------------------------------------
  bron   broo  68.2857  0.001   45.6607  90.9107   True
  bron   manh  95.0265  0.001   72.4015 117.6514   True
  bron   quee  16.4841 0.2716   -6.1409  39.1091  False
  bron   stat  21.5132 0.0716   -1.1118  44.1382  False
  broo   manh  26.7407 0.0111    4.1158  49.3657   True
  broo   quee -51.8016  0.001  -74.4266 -29.1766   True
  broo   stat -46.7725  0.001  -69.3975 -24.1475   True
  manh   quee -78.5423  0.001 -101.1673 -55.9173   True
  manh   stat -73.5132  0.001  -96.1382 -50.8882   True
  quee   stat   5.0291    0.9  -17.5959  27.6541  False
-----------------------------------------------------
```

Above results from Tukey HSD suggests that except 3 pairs:

1) Bronx-Queens

2) Bronx-StatenIslands

3) Queens-StatenIslands

all other pairwise comparisons for treatments rejects the null hypothesis and indicates statistically significant differences.

5. Machine Learning

After analysis, I have decided to drop these columns as they will not be useful in prediction:

```
data.drop(["name","last_review","host_id", "id", "neighbourhood", "host_name"], axis=1, inplace=True)
```

I switched strings to numerical data:

```
numeric_data = pd.get_dummies(data, columns=["neighbourhood_group","room_type"], prefix = ['ng',"rt"],drop_first=True)
```

I have created training and test data:

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
Dimensions of the training feature matrix: (40479, 14)
Dimensions of the training target vector: (40479,)
Dimensions of the test feature matrix: (10120, 14)
Dimensions of the test target vector: (10120,)
```

For standardization, I have used RobustScaler to remove outliers.

6. Linear Regression

I found the Least- Squared Error (Minimize Sum of Squared Errors-SSE)

7. Ridge Regression (L2)

To avoid overfit, I also minimize Lamda*Slopper in Ridge Regression in addition to min squared error.

```
Ridge Coef : [-2.36475507e+01 -2.48338550e+01  1.82693598e-01 -5.67334391e+00
 -7.76563025e+00 -4.00620748e-01  4.63332544e+01 -5.02492761e+01
  3.68553010e+01 -1.05588159e+01 -1.84550182e+02 -2.90899460e+01
 -9.03236038e+01 -1.28429040e+02]
Ridge Best Estimator: Ridge(alpha=0.31622776601683794, copy_X=True, fit_intercept=True,
    max_iter=10000, normalize=False, random_state=42, solver='auto',
    tol=0.001)
Ridge MSE:  112016.44348145518
```

8. Lasso (L1)

Difference from Ridge is to minimize lamda|slope| not lamda(slope)**2

The other difference is unsignificant coefs are accepted as 0.(No effect to other dependents. In other words If there is a high correlated feature, Lasso use just one of those and accept others as 0. Lasso prevents over fitting similar to Ridge.

```
Lasso Coef : [ -22.92006578  -24.64355751    0.18726548   -5.67301192   -7.72947761

   -0.39811332   46.18188973  -46.34864321   39.88195601   -7.30649982

 -176.12225314  -25.54594366  -90.2194084  -127.22224085]

Lasso Best Estimator: Lasso(alpha=0.02592943797404667, copy_X=True, fit_intercept=True,

      max_iter=10000, normalize=False, positive=False, precompute=False,

      random_state=42, selection='cyclic', tol=0.0001, warm_start=False)
Lasso MSE:  112015.2263969845
```

9. Elastic Net

uses the strong features of both Lasso and Ridge. very useful while high correlated features' implemention.

```
ElasticNet Coef : [ -23.09698239  -24.50958453    0.18582122   -5.67506973   -7.76192893

   -0.39864104   46.24383615  -46.89682006   39.68287688   -8.09268536

 -175.09851674  -27.75203768  -90.25648055 -127.23378519]

ElasticNet Best Estimator: : ElasticNet(alpha=0.0037065129109221566, copy_X=True, fit_intercept=True,

           l1_ratio=0.9500000000000001, max_iter=10000, normalize=False,

           positive=False, precompute=False, random_state=42,

           selection='cyclic', tol=0.0001, warm_start=False)
ElasticNet MSE:  112019.65551282265
```

10. XGBoost

```
XGBRegressor MSE:  84716.541772843
```