IBM Advanced Data Science Capstone Project

Airbnb rental price prediction in New York City

TABLE OF CONTENTS

1

Data Acquisition

The process of collecting the data from the source

2

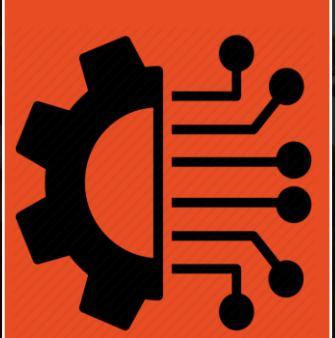
Data Exploration

Data exploration is an approach which uses visual exploration to understand the dataset

3

Data Wrangling

Data wrangling is the process of cleaning and unifying messy and complex data sets for easy access



4

ETL & Feature Engineering

Feature engineering refers to a process of selecting and transforming variables when creating a predictive model

5

Data Partitioning

The process of splitting the data into train validation and test for building a model

6

Data Modelling

Modeling is the statistical technique that is chosen to find trends, patterns and relation-ships within existing data.

7

Model Selection

Based on various metrics the best model will be chosen for implementation

What is Airbnb?

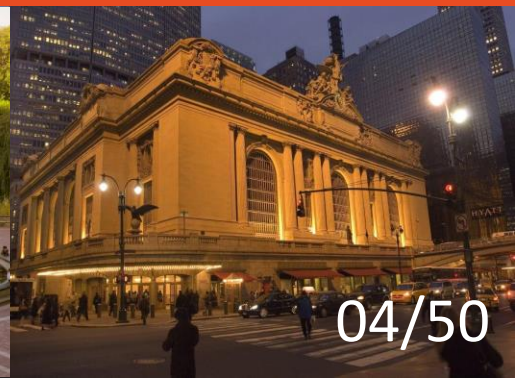
Airbnb is an online marketplace for arranging or offering lodging, primarily homestays, or tourism experiences. The company does not own any of the real estate listings, nor does it host events; it acts as a broker, receiving commissions from each booking. The company is based in San Francisco, California, United States.

It is a platform that connects people who want to rent out their homes with people who are looking for accommodations in that locale. It currently covers more than 81,000 cities and 191 countries worldwide. The company's name comes from "air mattress Bed & Breakfast."

New York City

- New York City is the most populous city in the United States.
- With an estimated population of 8.400.000 distributed over about 784 square kilometers, New York City is also the most densely populated major city in the United States.
- New York City received a record of almost 63 millions tourists in 2017, and Manhattan hosts three of the world’s 10 most-visited tourist attractions: Times Square, Central Park and Grand Central Terminal.



The aim of our project

Airbnb is a platform where millions of hosts and travelers list their space and book unique accommodations anywhere in the world. Airbnb affects the city's housing supply and affordability. In a modern, busy cosmopolitan like New York City, there are so many competitions for housing and huge demands for accommodations, Airbnb pricing is important to get right. If the price is too high the hosts might risk losing customers while if the price is too low they might be missing out on potential benefits. The goal of our project is to come up with the appropriate prediction of Airbnb price using Machine Learning, so the hosts can achieve optimal profits. One of the challenges we faced is that since the raw data we used consists of both categorical and numerical values and missing values which makes classification more difficult.

PROJECT DESCRIPTION

CONTEXT

This data file includes all needed information to find out more about hosts, geographical availability, necessary metrics to make predictions and draw conclusions.

ACKNOWLEDGEMENT

The dataset is obtained directly from the Airbnb website. Specifically, this dataset describes the listing activity and metrics in New York City as was in June 8, 2020.

PLANNING, EXECUTION & IMPLEMENTATION

Here we will be using various machine learning algorithms to predict the best model for rental prices & segmentation then we will select the best suited one based on the metrics for implementation to achieve the goal of the project.

CONTENT

Since 2008, guests and hosts have used Airbnb to expand on traveling possibilities and present more unique, personalized way of experiencing the world. This dataset describes the listing activity and metrics in New York City.

OBJECTIVE

The objective is to create a best model to predict the rental prices of the Airbnb listings throughout New York State & also create clusters based on all the variables to create a data structure and segment the data.

1. Data Acquisition

Data Source

- The data set is extracted from the Airbnb website and information is update as of June 8, 2020.
- The data set is in a comma separated values format (CSV).
- The dimensions of the New York City data consists of 16 variables with 49530 observations.
- The target variable for developing the predictive model will be the prices of the Airbnb properties in USD.

Data Dictionary:

1.	id:	listing ID
2.	name:	name of the listing
3.	host_id:	host ID
4.	host_name:	name of the host
5.	neighbourhood_group:	location
6.	neighbourhood:	area
7.	latitude:	latitude coordinates
8.	longitude:	longitude coordinates
9.	room_type:	listing space type
10.	price:	price in dollars
11.	minimum_nights:	amount of nights minimum
12.	number_of_reviews:	number of reviews
13.	last_review:	latest review date
14.	reviews_per_month:	number of reviews per month
15.	calculated_host_listings_count:	amount of listing per host
16.	availability_365:	number of days when listing is available for booking

2. Data Exploration

Pandas - profiling

- After acquiring the dataset from the source we can explore it to understand the various features present in the dataset and its characteristics.
- In Python, there is a special package to do this all exploration of data in to a single step known as pandas-profiling.
- The pandas-profiling Python package is a great tool to create HTML profiling reports. For a given dataset, it computes the following statistics:
  - ✓Essentials: type, unique values, missing values.
  - ✓Quantile statistics like minimum value, Q1, median, Q3, maximum, range, interquartile range.
  - ✓Descriptive statistics like mean, mode, standard deviation, sum, median absolute deviation, coefficient of variation, kurtosis, skewness.
  - ✓Most frequent values.
  - ✓Histograms.
  - ✓Correlations highlighting of highly correlated variables, Spearman and Pearson matrices.

2. Data Exploration

New York City encompasses five county-level administrative divisions called boroughs:

- Manhattan
- Brooklyn
- Queens
- Bronx
- Staten Island

- The number of unique neighbourhood_group in dataset is :

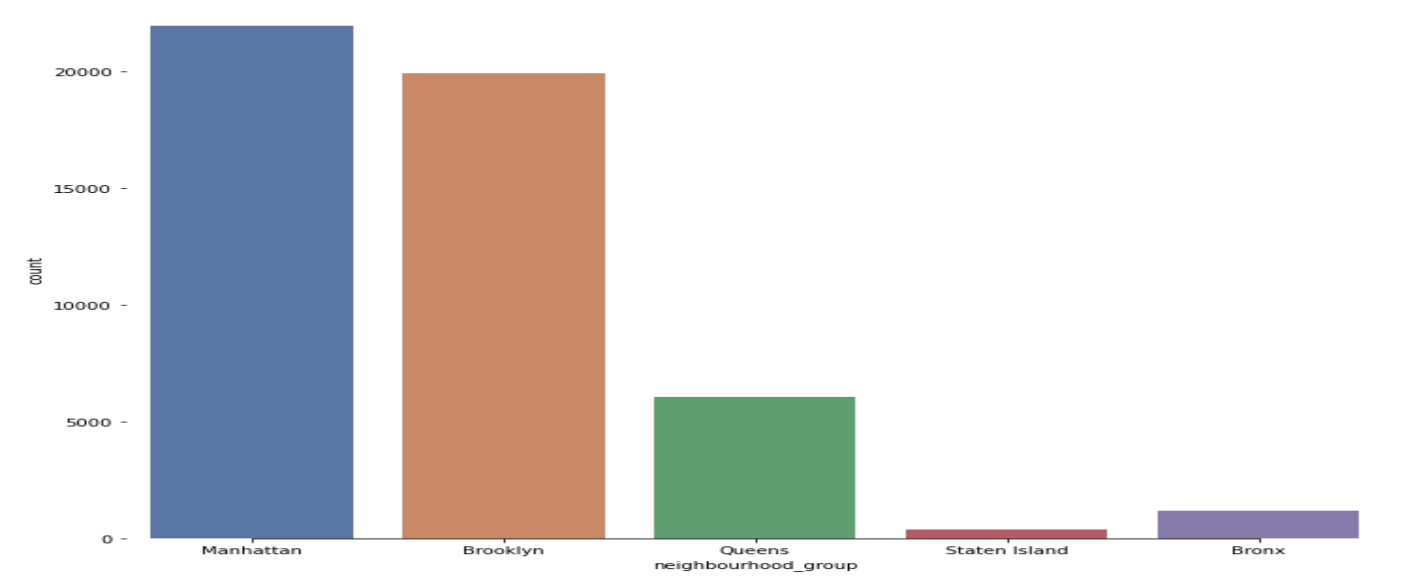
```
 Manhattan        21963
Brooklyn          19931
Queens             6068
Bronx              1198
Staten Island       370
Name: neighbourhood_group, dtype: int64
```

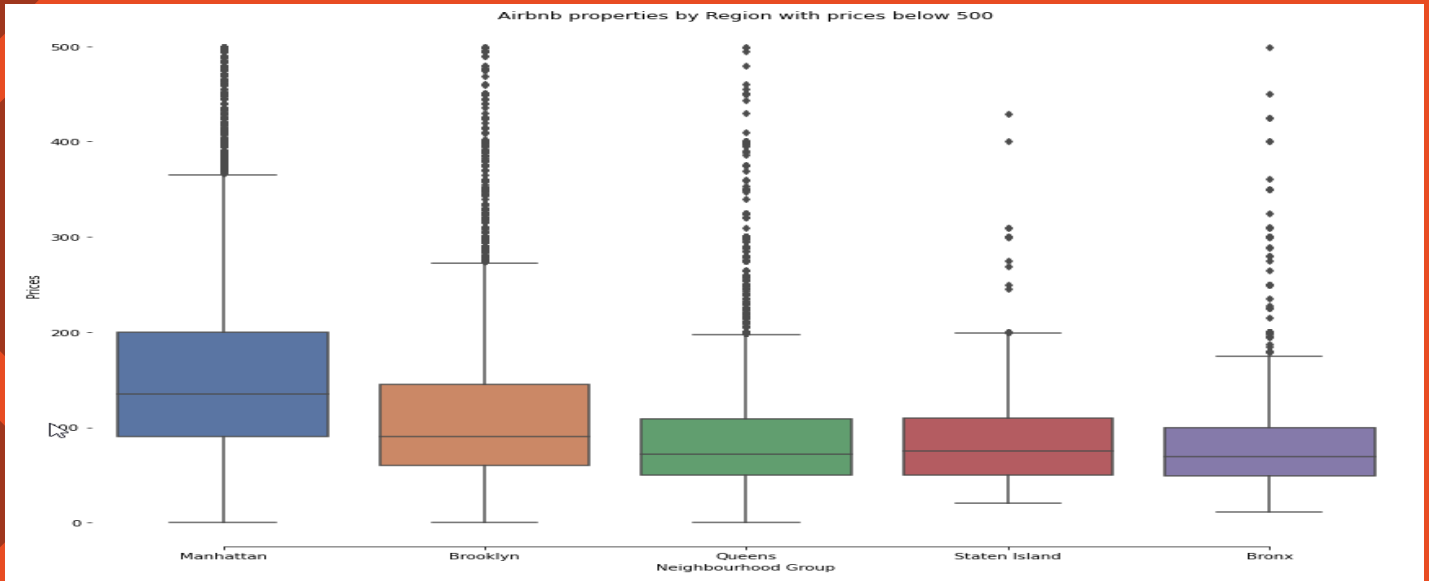
2. Data Exploration

The New York City is composed of five boroughs:

Manhattan	Brooklyn	Queens	Bronx	Staten Island
				
Airbnb listings 21.963	**Airbnb listings 19.931**	**Airbnb listings 6.068**	**Airbnb listings 1.198**	**Airbnb listings 370**
Average price ***$ 218,86***	*Average price* ***$ 125,06***	*Average price* ***$ 99,75***	*Average price* ***$ 90,18***	*Average price* ***$ 116,91***
32 unique neighborhood	*48 unique neighborhood*	*51 unique neighborhood*	*48 unique neighborhood*	*43 unique neighborhood*

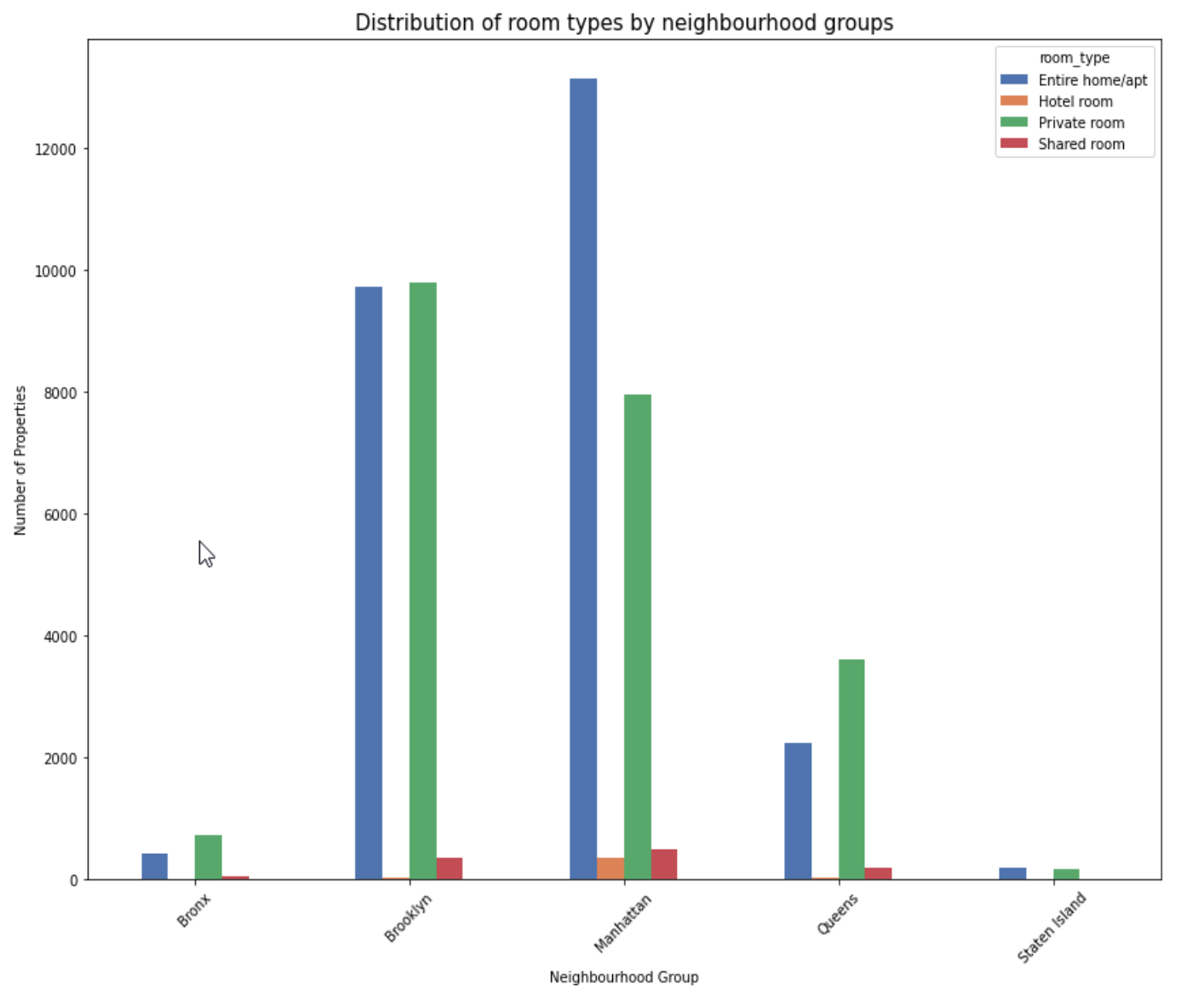
2. Data Exploration and Insights





- From the counterplot, we can see that most of the Airbnb properties are located in Manhattan.
- From the boxplot, we confirm that the Manhattan region also consist of a high variance in term of prices as compared to the other regions.
- We also notice some outliers in the prices for other regions i.e. prices above $ 1000.

2. Data Exploration and Insights



- From the bar chart, we observe that prices for Entire Home/Apt is the most listed by the hosts.
- The demand for private rooms and Entire home/Apt rentals is greater as compared to the other room types.
- As most of the hotels are located in Manhattan, the other boroughs consist mostly of private or shared rooms.

2. Data Exploration

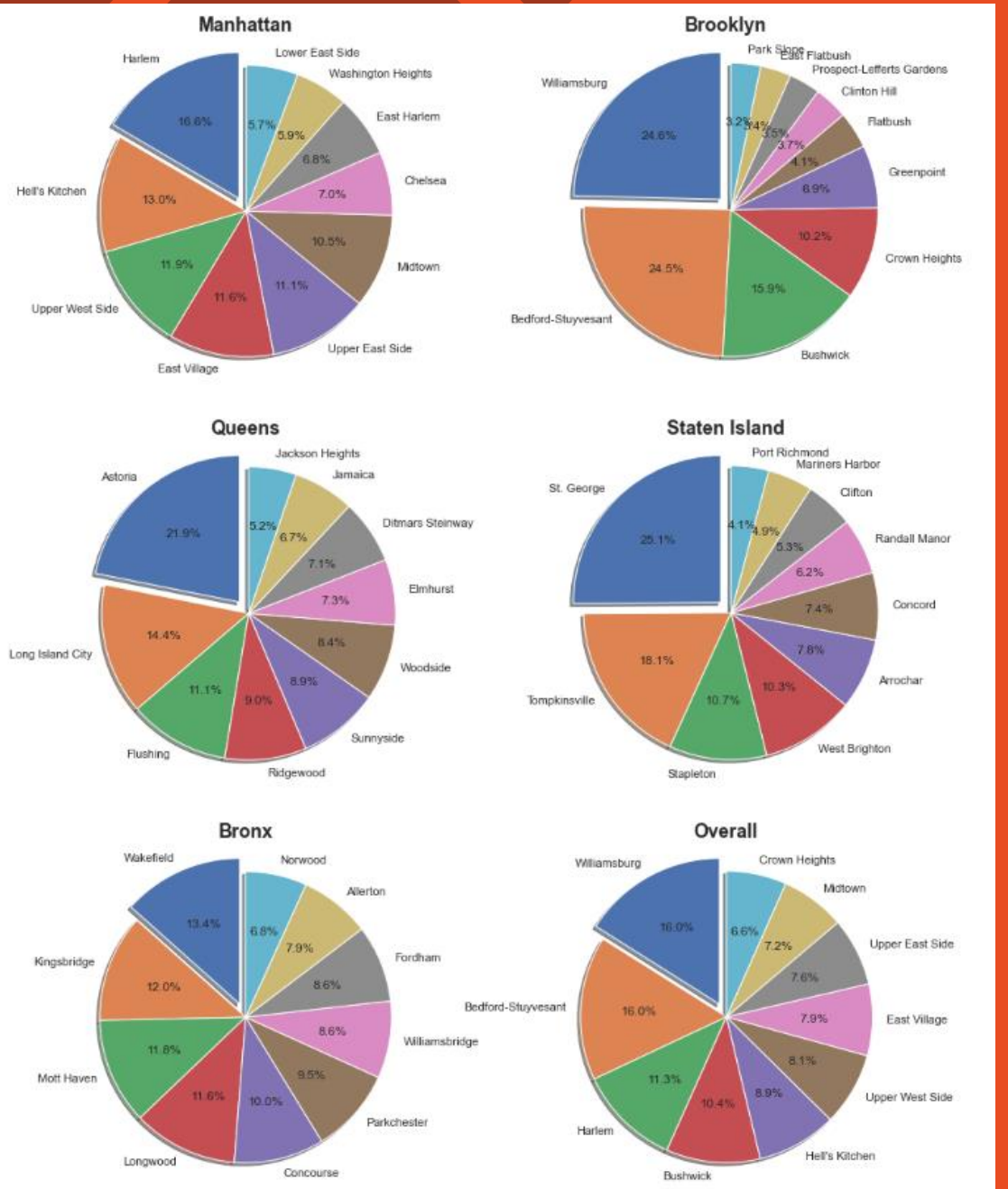
2.3. Most popular Neighborhood

```
n_count=df_listings.neighbourhood.value_counts()
n_count.head(10)
```

```
Williamsburg          3752
Bedford-Stuyvesant    3736
Harlem                2649
Bushwick              2431
Hell's Kitchen        2076
Upper West Side       1899
East Village          1860
Upper East Side       1769
Midtown               1673
Crown Heights         1553
Name: neighbourhood, dtype: int64
```

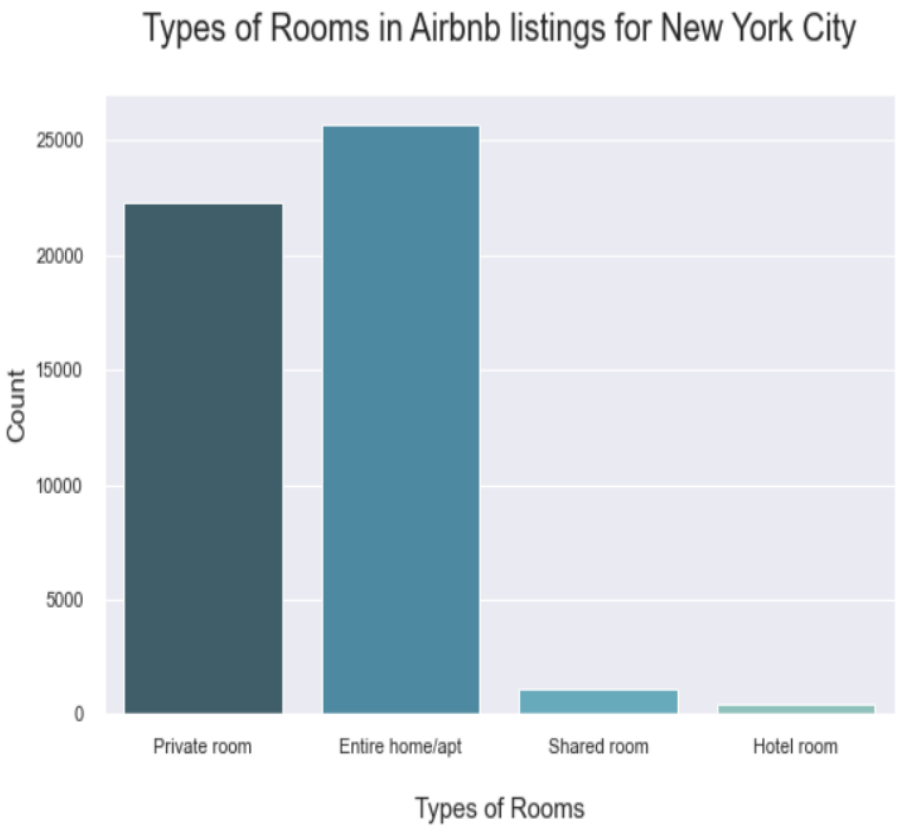
```
region=df_listings.groupby(['neighbourhood_group','neighbourhood']).count()[['id']]
region=region['id'].groupby(level=0, group_keys=False)
region.nlargest()
```

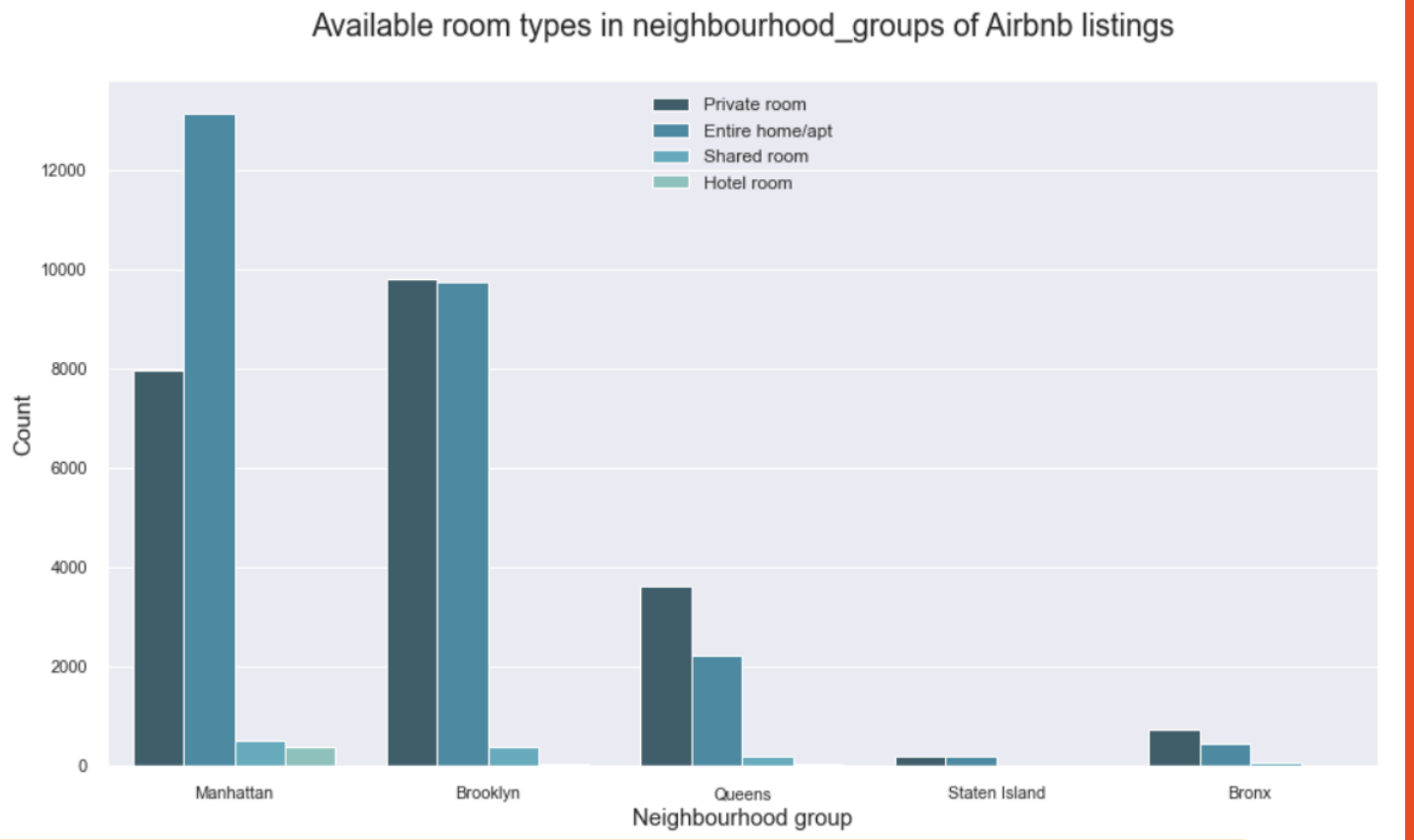
```
neighbourhood_group  neighbourhood
Bronx                Wakefield              75
                     Kingsbridge            67
                     Mott Haven             66
                     Longwood               65
                     Concourse              56
Brooklyn             Williamsburg         3752
                     Bedford-Stuyvesant   3736
                     Bushwick             2431
                     Crown Heights        1553
                     Greenpoint           1049
Manhattan            Harlem               2649
                     Hell's Kitchen       2076
                     Upper West Side      1899
                     East Village         1860
                     Upper East Side      1769
Queens               Astoria               892
                     Long Island City      589
                     Flushing              453
                     Ridgewood             366
                     Sunnyside             364
Staten Island        St. George             61
                     Tompkinsville          44
                     Stapleton              26
                     West Brighton          25
                     Arrochar               19
Name: id, dtype: int64
```



2. Data Exploration

2.3. Room types and their availability



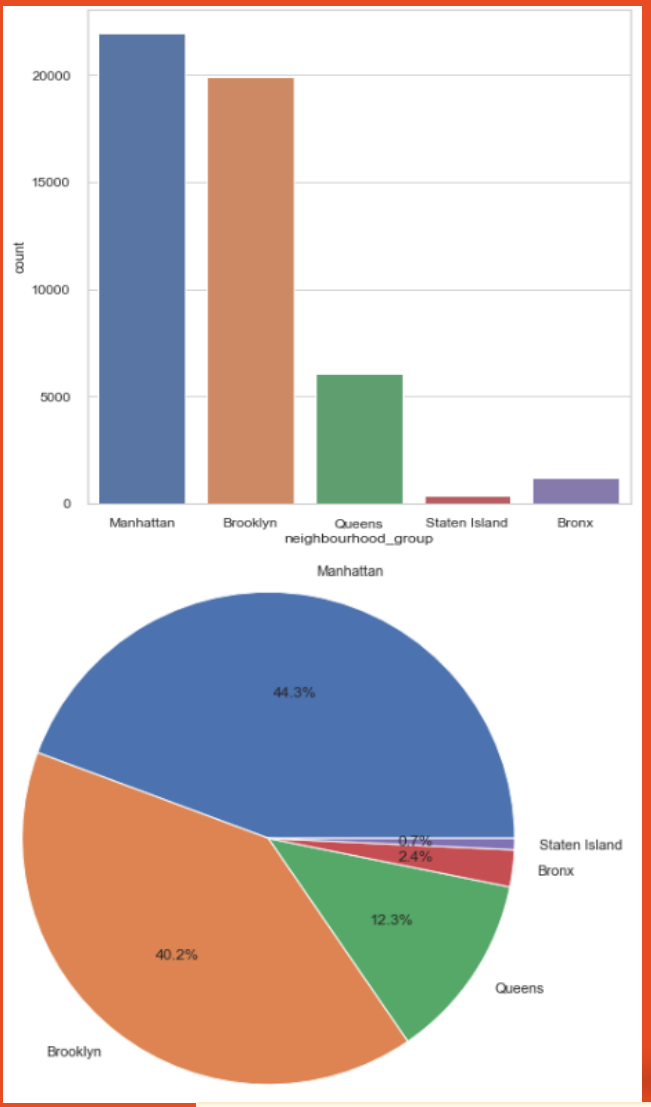


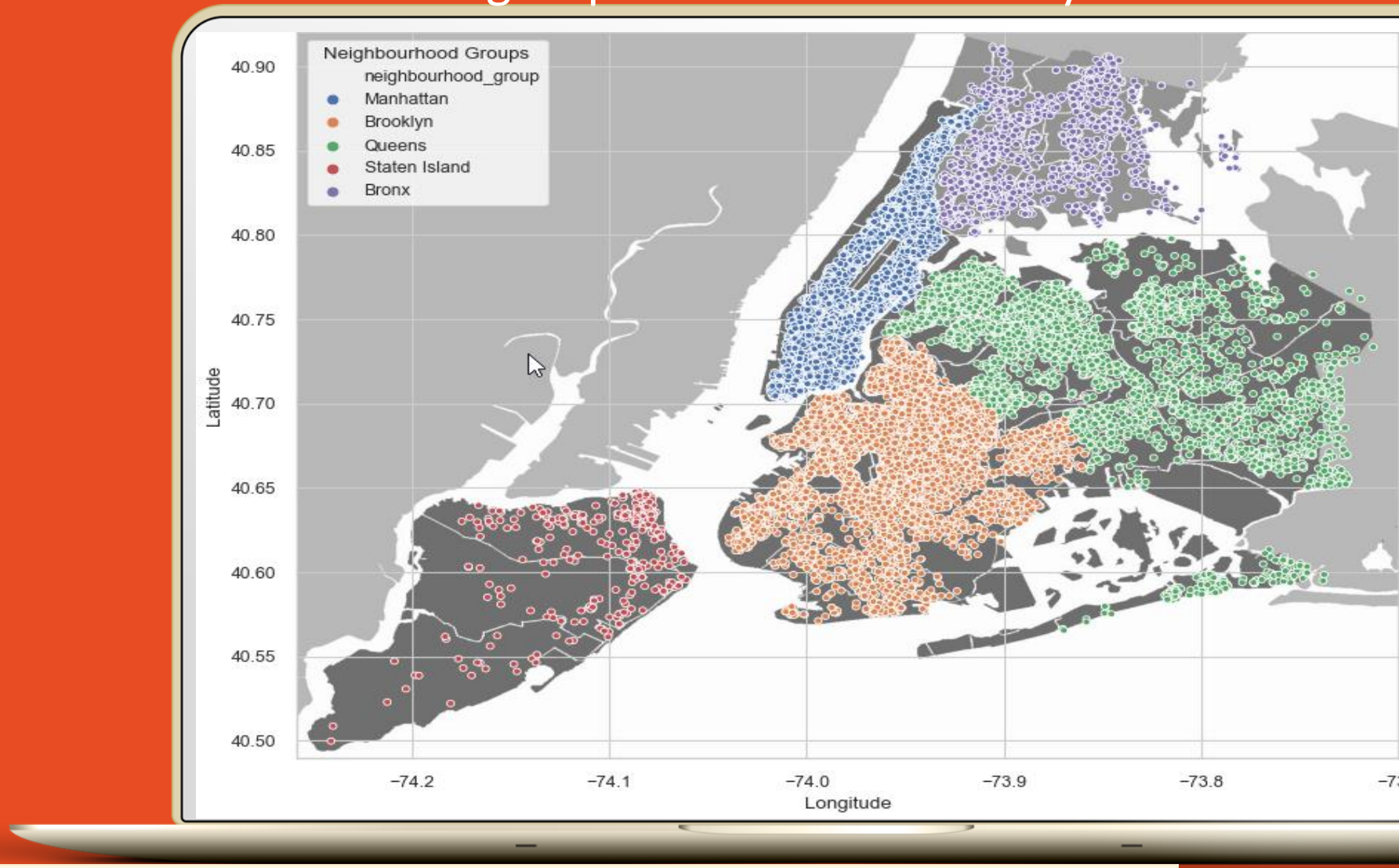
From the above, we can see that we have few number of `Shared rooms` as usually the travelers and tourists prefer to have a space for them without any disturbances, therefore `Entire home/apt` & `Private rooms` are more popular and have more availability.

We don't have much Airbnb listings in boroughs of `Staten Island` and `Bronx`, hence we will be having less numbers of rooms in all categories. Likewise we have more number of Airbnb listings in both `Manhattan` and `Brooklyn`, but the ratio of the rooms in each borough is almost the same.

2. Data Exploration

2.5. Distribution of Airbnb listings by Neighborhood group across New York City





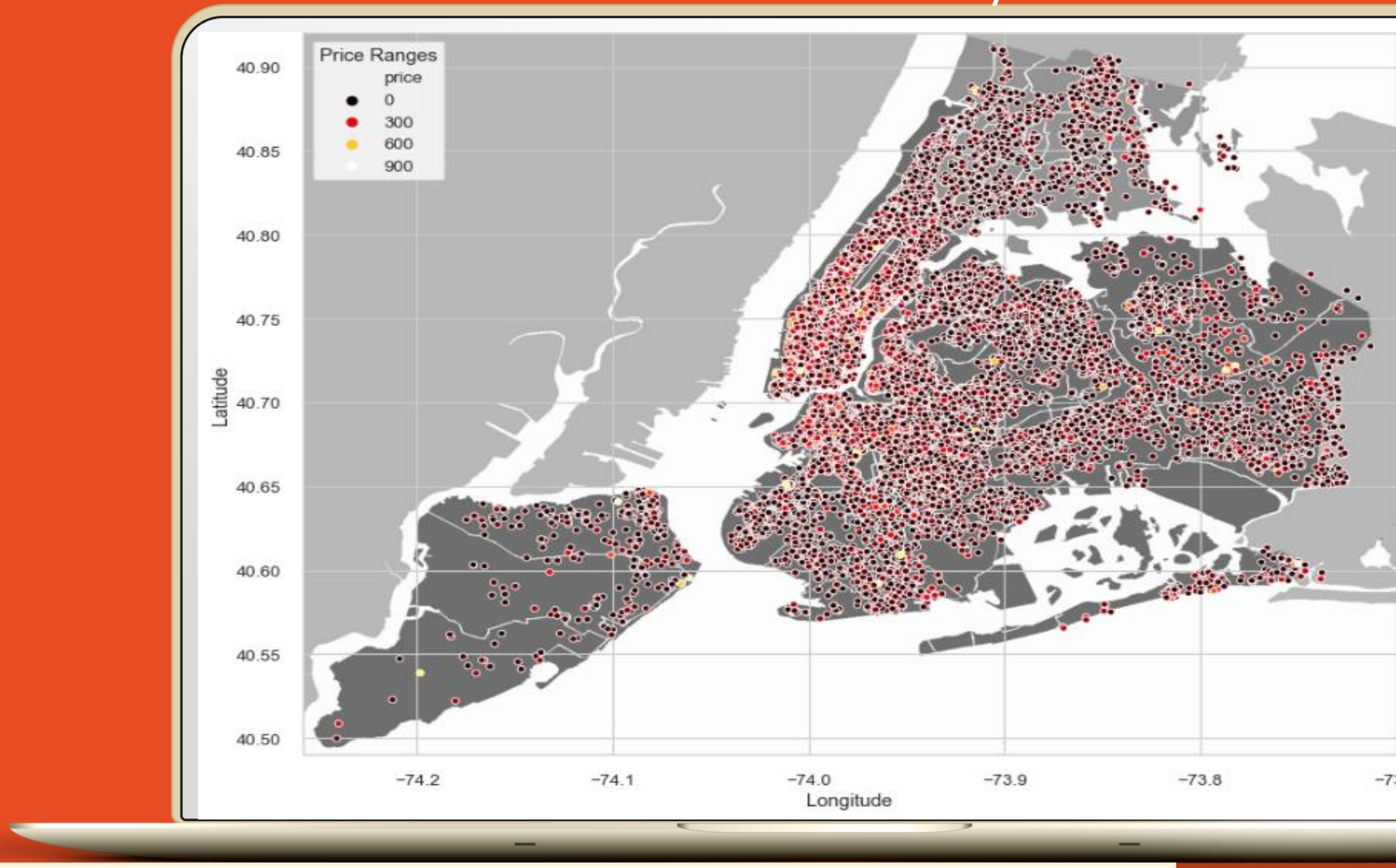
From the above spatial analysis, it is evident that there are a lot of Airbnb listings in both `Manhattan` & `Brooklyn` and very few Airbnb rental spaces in `Staten Island` as our Data Analysis indicate.

2. Data Exploration

2.6. Price Distribution of Airbnb listings across New York City

The Average price of each Boroughs :

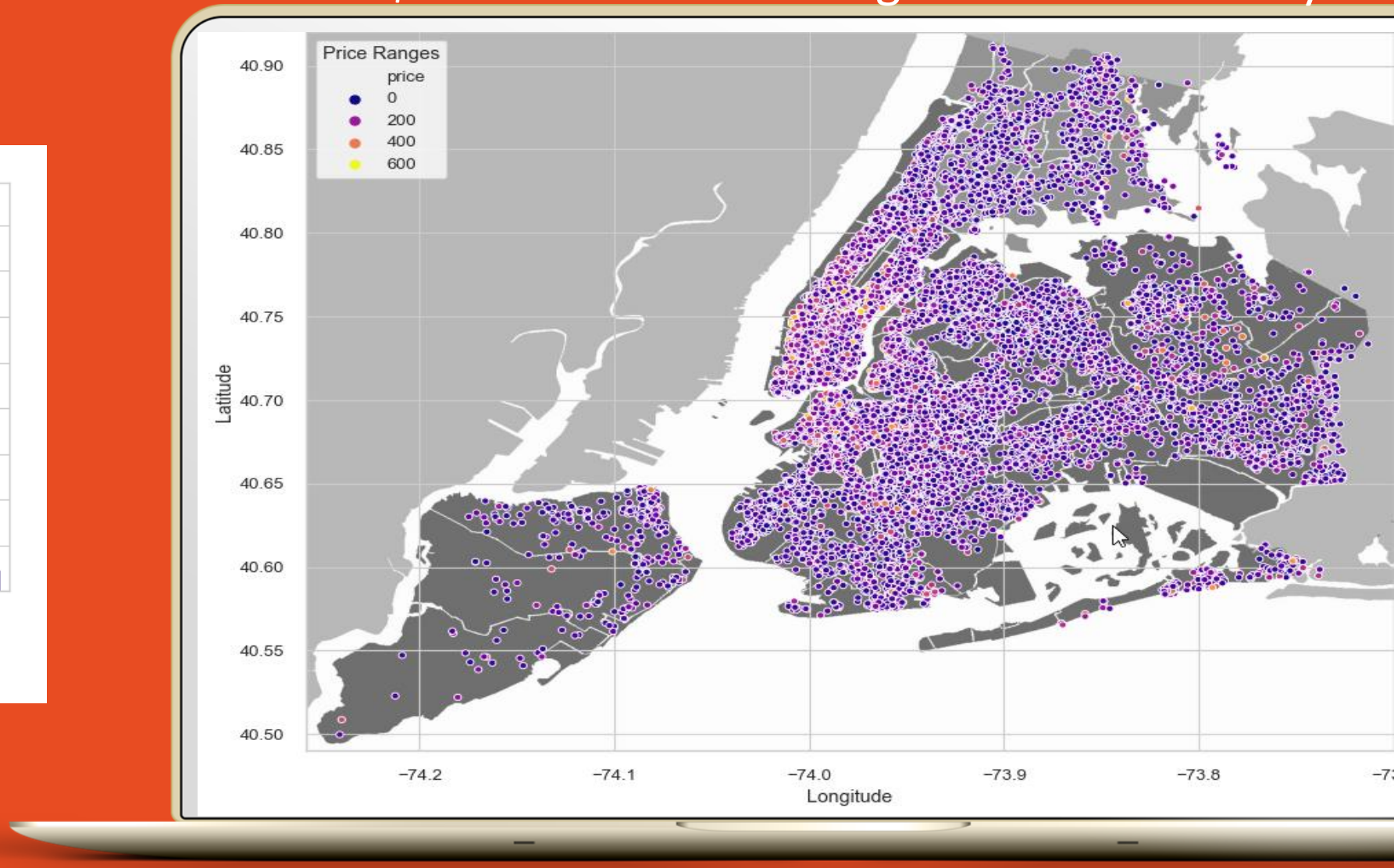
neighbourhood_group	Average-Price
Bronx	90.176127
Brooklyn	125.056194
Manhattan	218.855166
Queens	99.745056
Staten Island	116.908108



From the above, we can see that the average price of rent is the same for the boroughs of `Bronx`, `Queens` and `Staten Island`. `Brooklyn` is more expensive than the three neighborhoud_group mentioned above. Finally, we must notice that the rent price is highly overpriced in region of `Manhattan`.

2. Data Exploration

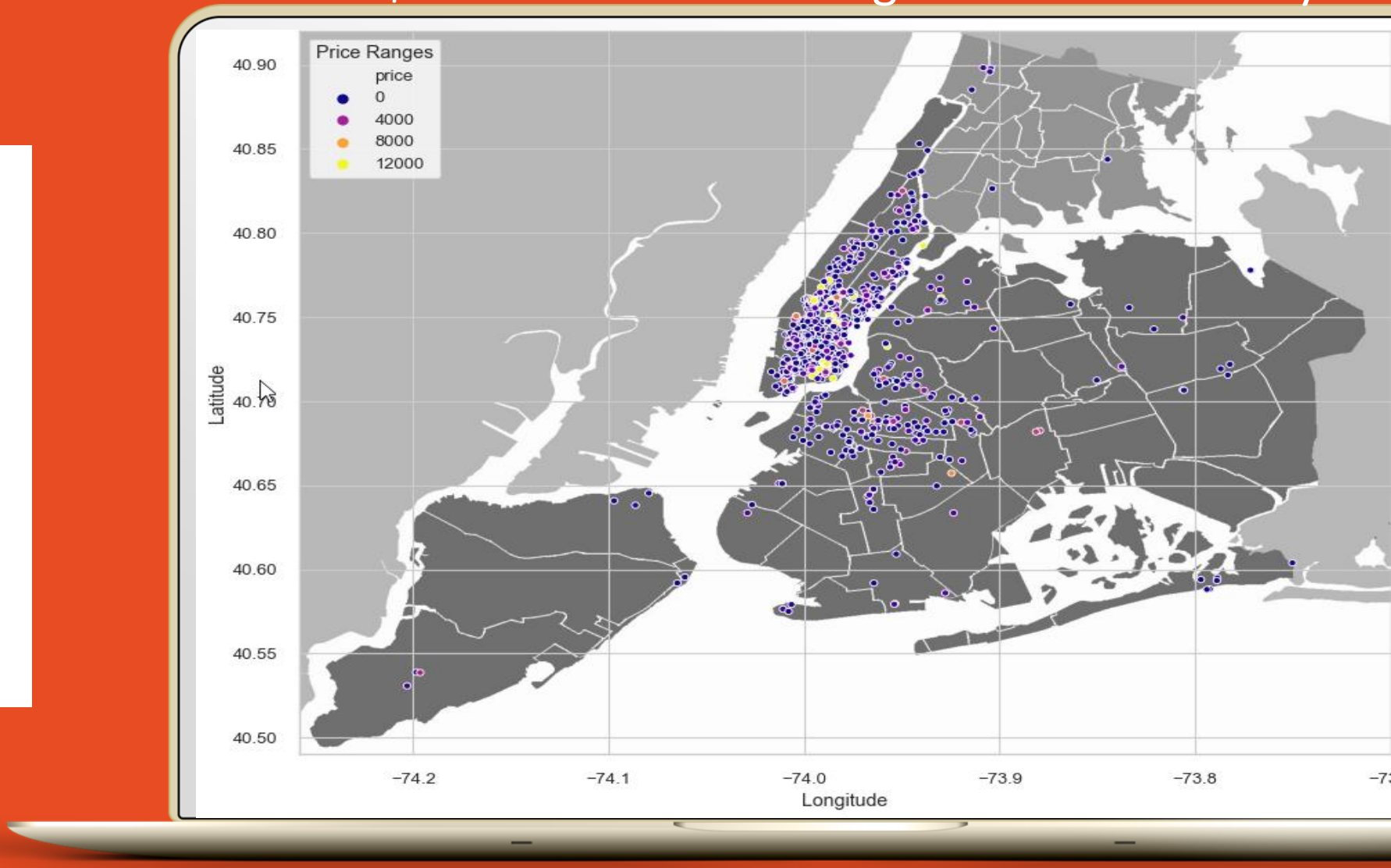
2.7. Price Distribution of Airbnb listings with price below $600 across the boroughs of New York City



More than 80% of Airbnb listings in New York City which price is below $600 are located in boroughs of `Manhattan` and `Brooklyn`.

2. Data Exploration

2.8. Price Distribution of Airbnb listings with price above $600 across the boroughs of New York City

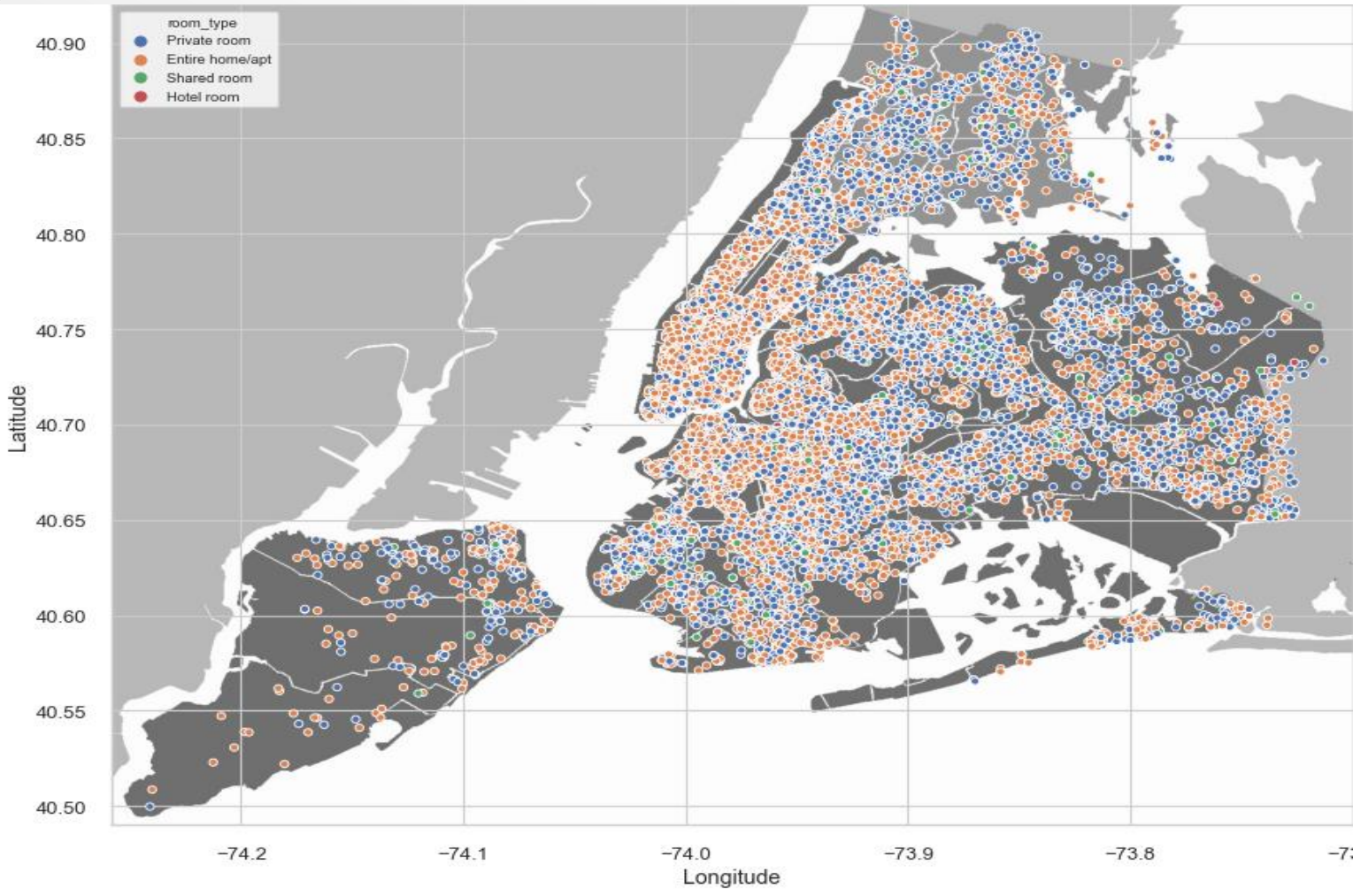


Almost three-quarters of Airbnb listings in New York City which price is above $600 are located in borough of `Manhattan`.

2. Data Exploration

2.9. Distribution of Room types across the boroughs of New York City

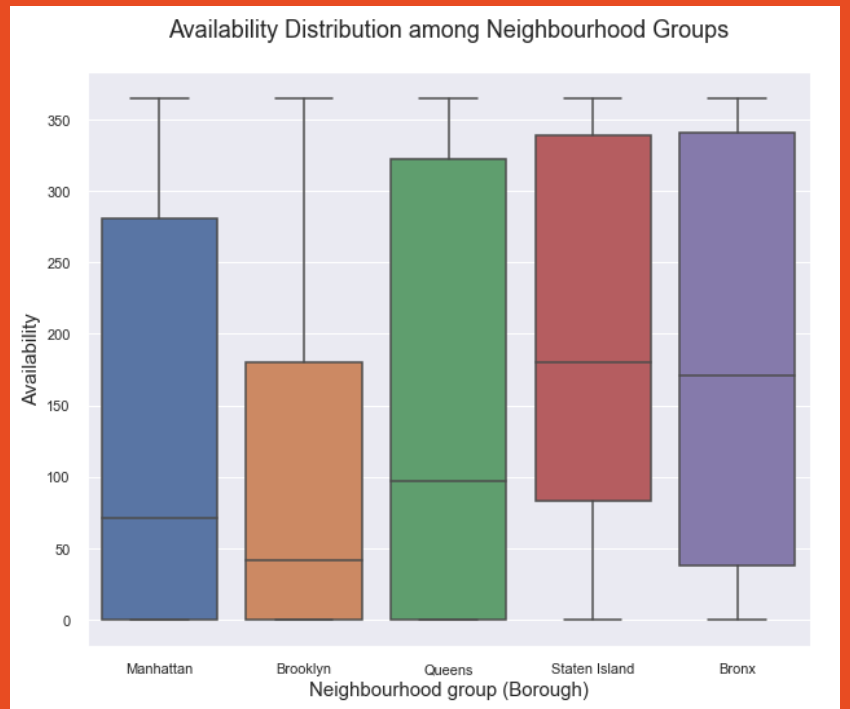
neighbourhood_group	room_type	id
Bronx	Entire home/apt	428
	Private room	726
	Shared room	44
Brooklyn	Entire home/apt	9728
	Hotel room	34
	Private room	9804
	Shared room	365
Manhattan	Entire home/apt	13149
	Hotel room	364
	Private room	7957
	Shared room	493
Queens	Entire home/apt	2229
	Hotel room	33
	Private room	3611
	Shared room	195
Staten Island	Entire home/apt	182
	Private room	181
	Shared room	7



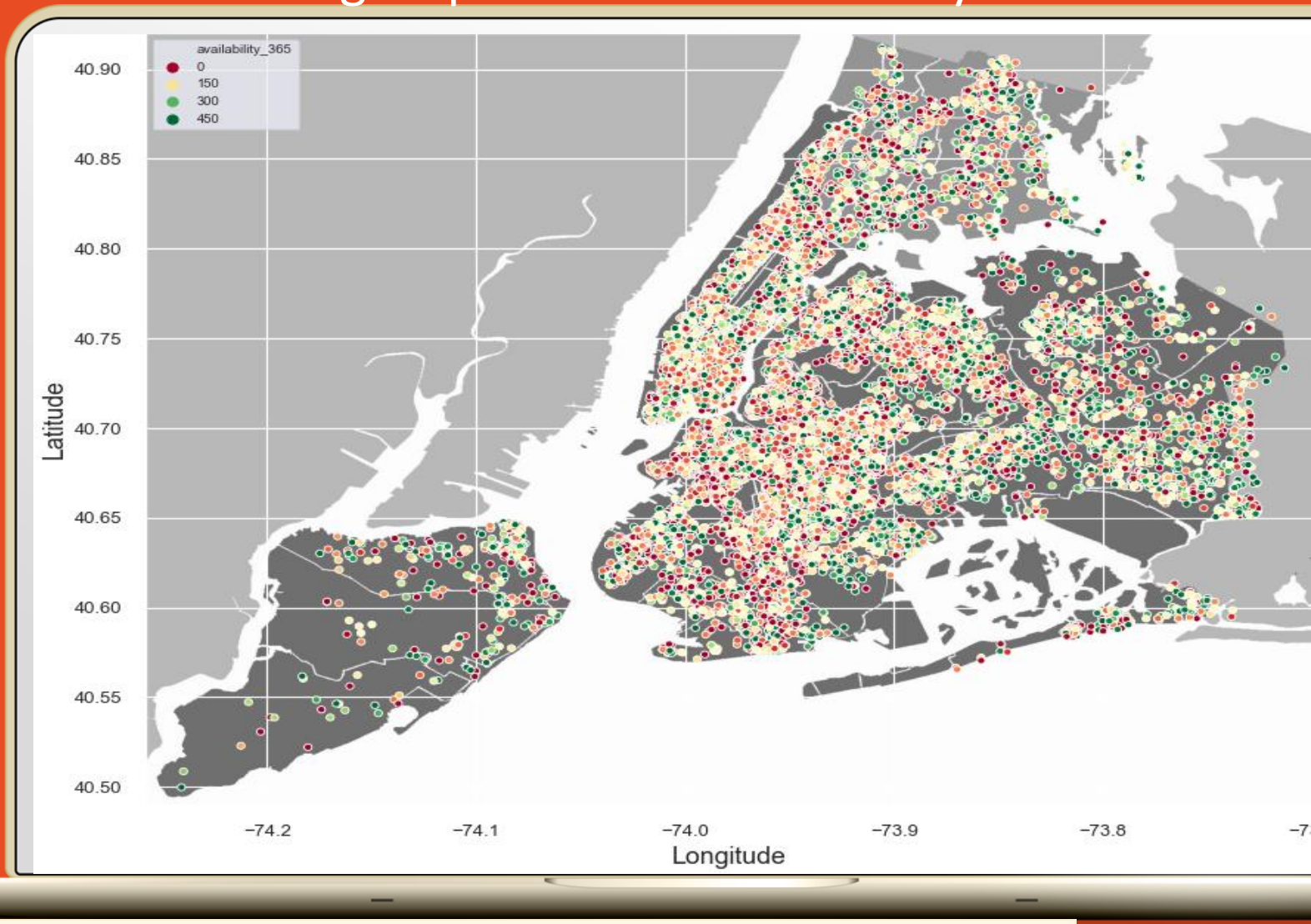
From the above spatial analysis we can see that the `Manhattan` has more number of `Entire home/apt` (orange points) compared with `Private rooms` (blue points) and `Brooklyn` has more number of `Private Rooms` than `Entire home/apt`.

2. Data Exploration

2.10. Distribution of Airbnb listings by Neighborhood group across New York City




```
Percentage of Airbnb listings not available in New York      :  40.65 %
Percentage of Airbnb listings not available in Manhattan     :  41.68 %
Percentage of Airbnb listings not available in Brooklyn      :  45.1 %
Percentage of Airbnb listings not available in Queens        :  27.87 %
Percentage of Airbnb listings not available in Staten Island :  14.05 %
Percentage of Airbnb listings not available in Bronx         :  20.7 %
```

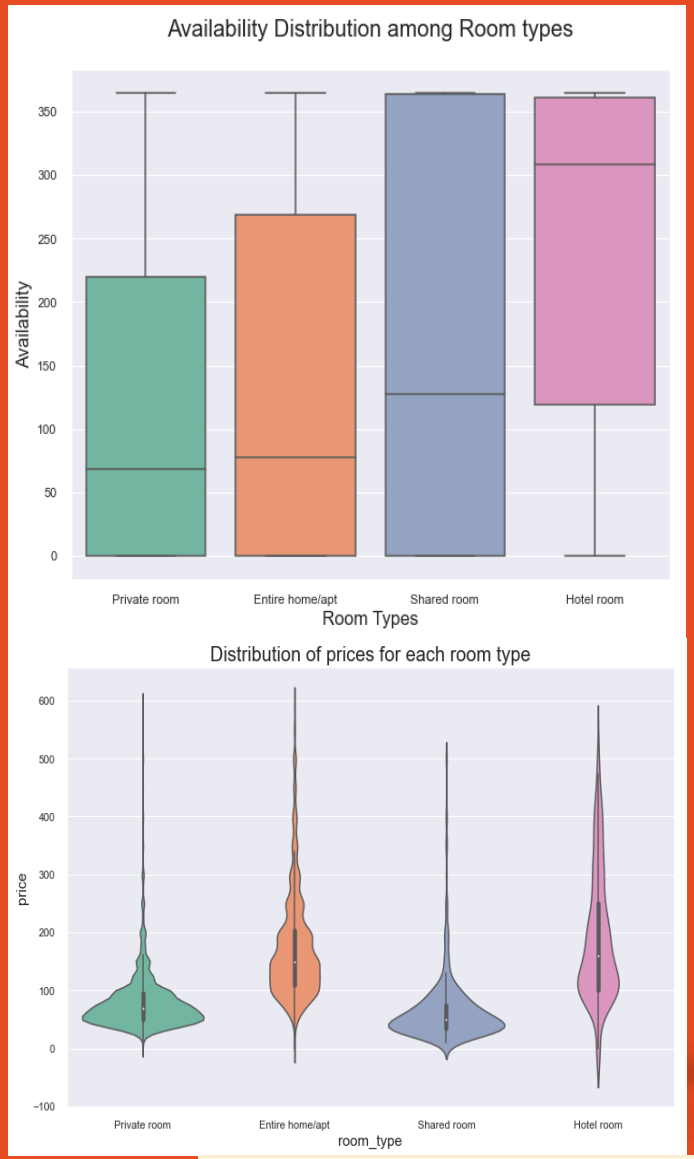


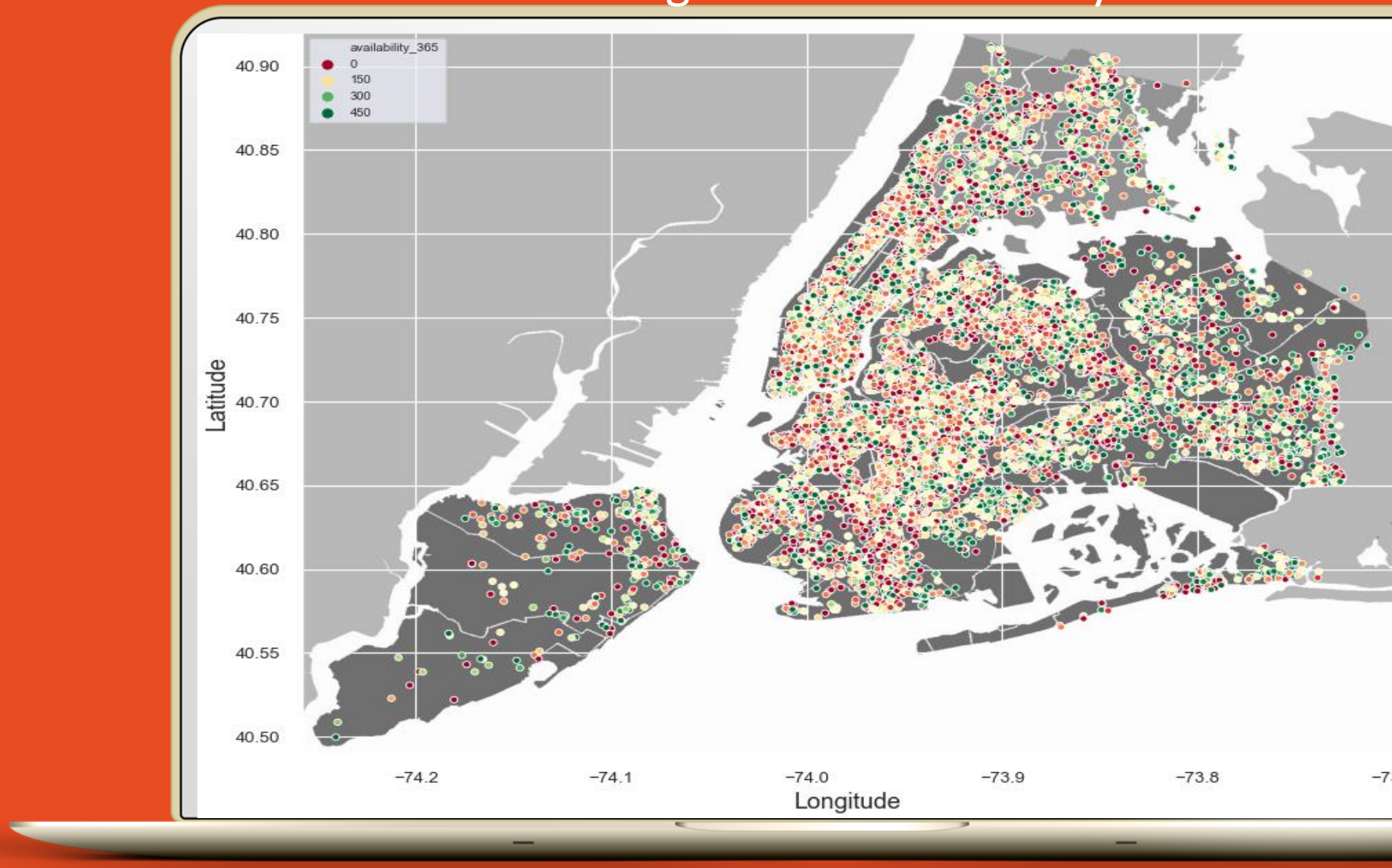
In the above plot, Red dots represents less availability of rooms (throughout a year) and Green dots represents more availability.

We can notice that the boroughs like `Brooklyn` and `Manhattan` are having less availability of rooms due to its popularity and on the other hand the boroughs `Staten Island` and `Bronx` have more availability of rooms as because its less popular than the above two boroughs.

2. Data Exploration

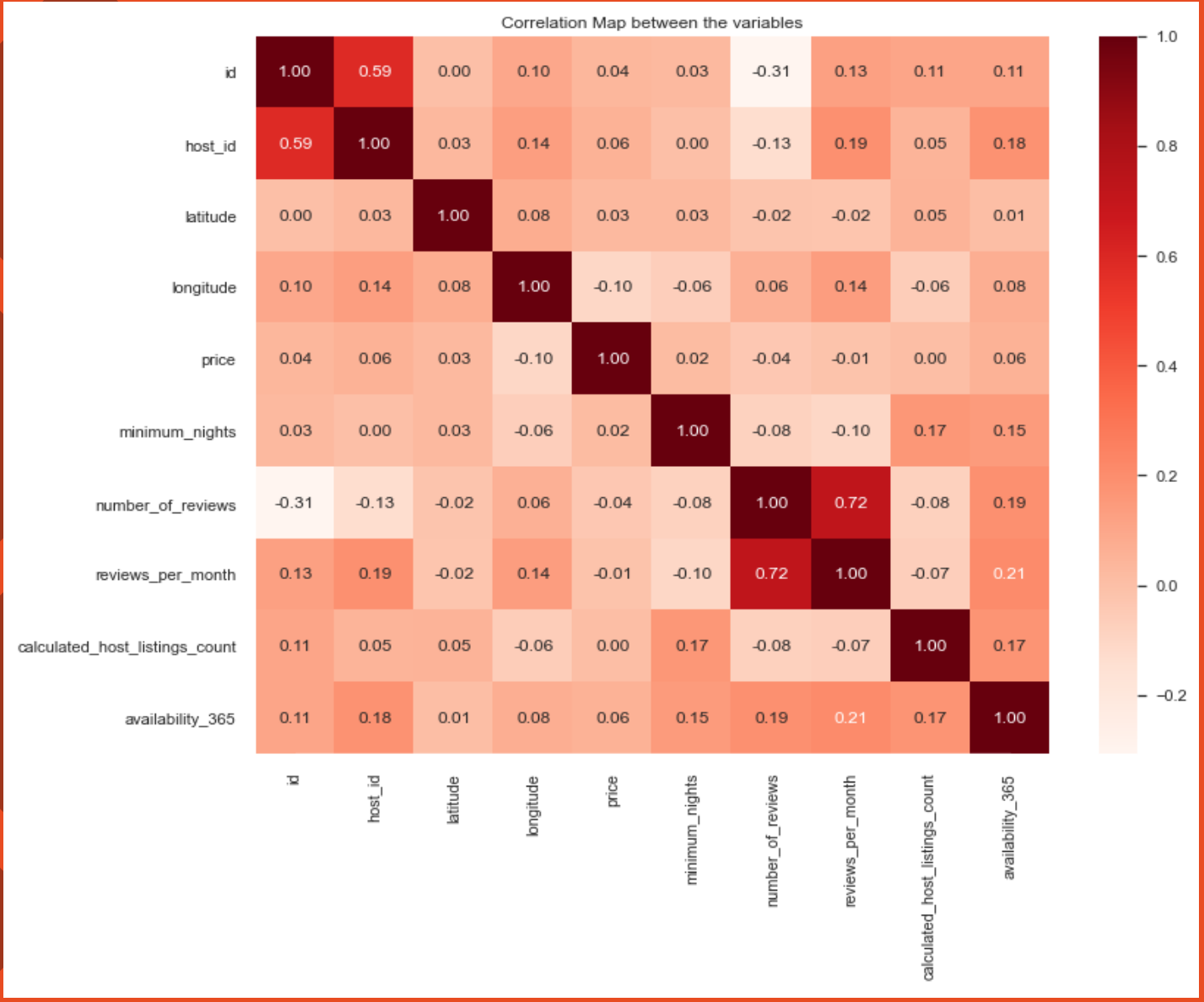
2.11. Availability of Airbnb listings across the boroughs of New York City





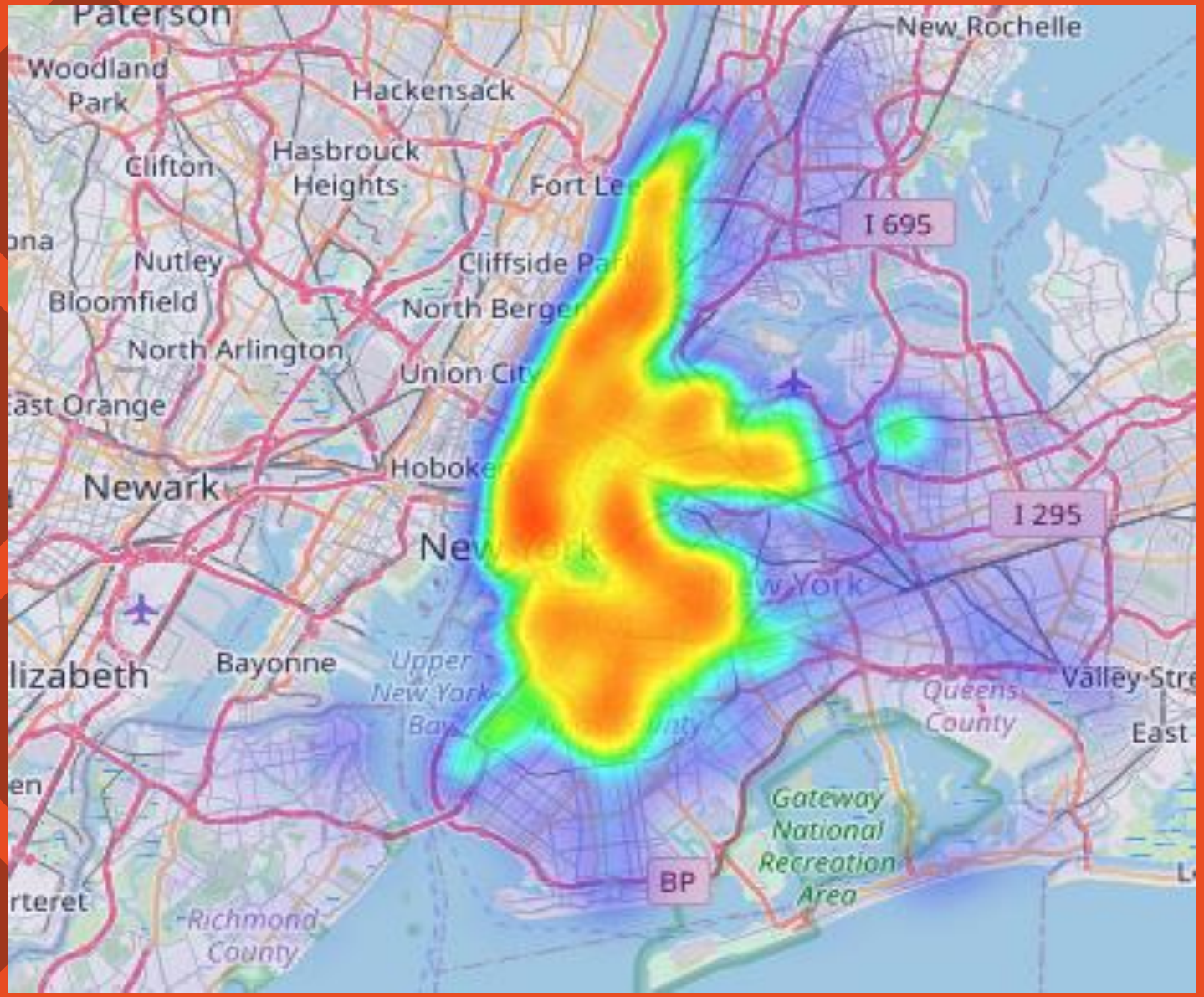
From above we can see that there is less availability in `Private rooms` and more availability in `hotel rooms`.

2. Data Exploration - Heatmap



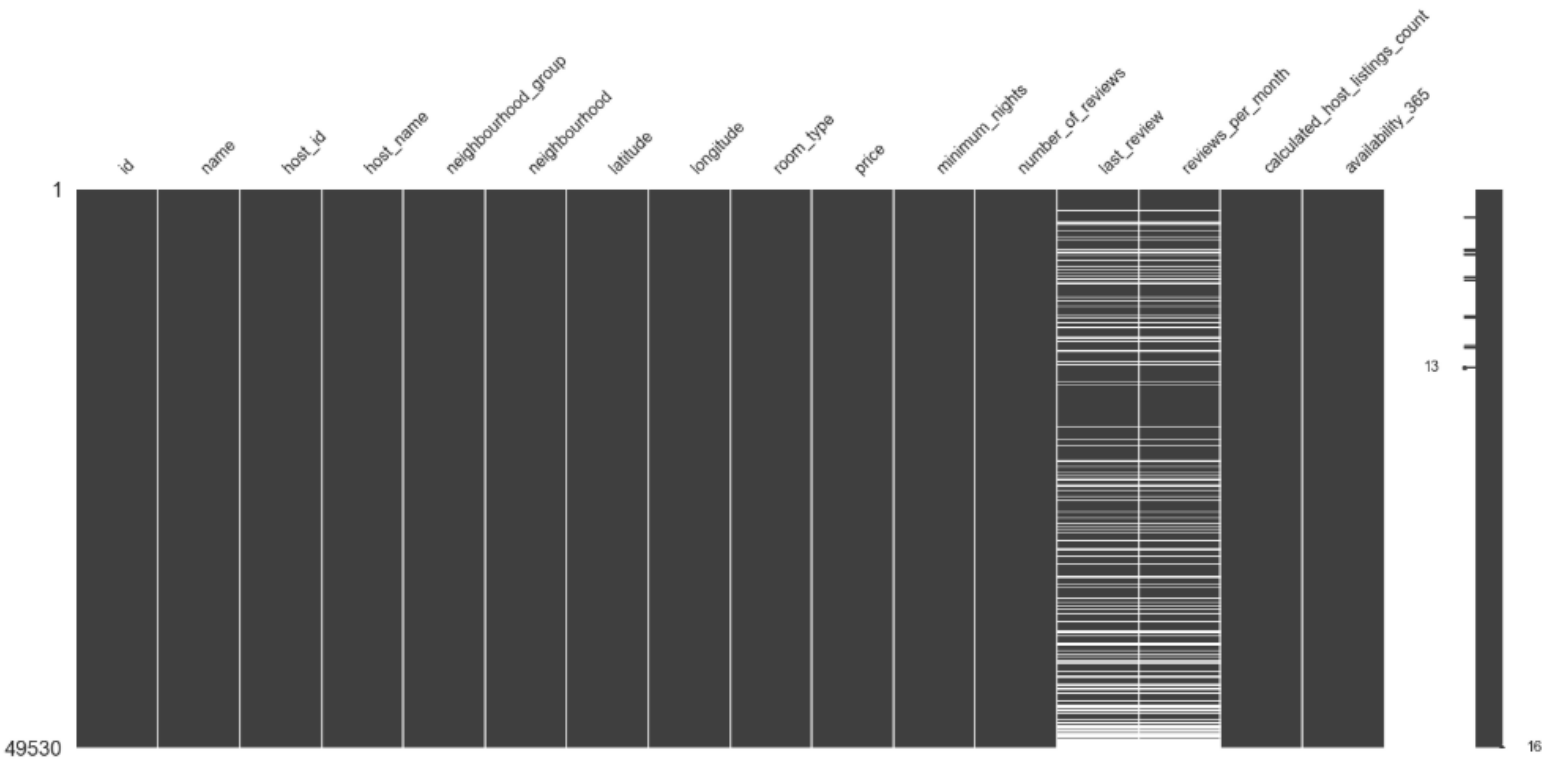
- “latitude” is the most correlated variable with “price”, which emphasizes one of the most important concepts in real estate *‘Location matters’.*
- “longitude” seems to be negatively correlated with “price”.
- “price” is also positively correlated with “availability_365”.
- “price” is negatively correlated with “number_of_reviews” and “reviews_per_month”.

2. Data Exploration - Conclusions



- The closer you are to Midtown and the lower Manhattan area, the more expensive it gets.
- The average price for an Airbnb rented space in New York City is $162.7 per night.
- The percentage of Airbnb listings not available in NYC is 40.6%.
- The average minimum number of nights you need to get an Airbnb listing in New York City is 8.2.

3. Data Wrangling



	Total	Percent %
reviews_per_month	11319	22.852816
last_review	11319	22.852816
name	18	0.036342
host_name	6	0.012114
availability_365	0	0.000000
calculated_host_listings_count	0	0.000000
number_of_reviews	0	0.000000
minimum_nights	0	0.000000
price	0	0.000000
room_type	0	0.000000
longitude	0	0.000000
latitude	0	0.000000
neighbourhood	0	0.000000
neighbourhood_group	0	0.000000
host_id	0	0.000000
id	0	0.000000

From the above we can see that the features `last_review` and `reviews_per_month` are missing about 23% of the data, hence we will be replacing null values in their columns with '0' in the dataset.

We will also be dropping the features `id` , `name` , `host_id` and `host_name` as they possess insignificant relationship to the target variable `price` and also these are textual data. Hence we will be dropping these features while modelling but for Exploratory Data Analysis (EDA) we will make use of it to understand the dataset better.

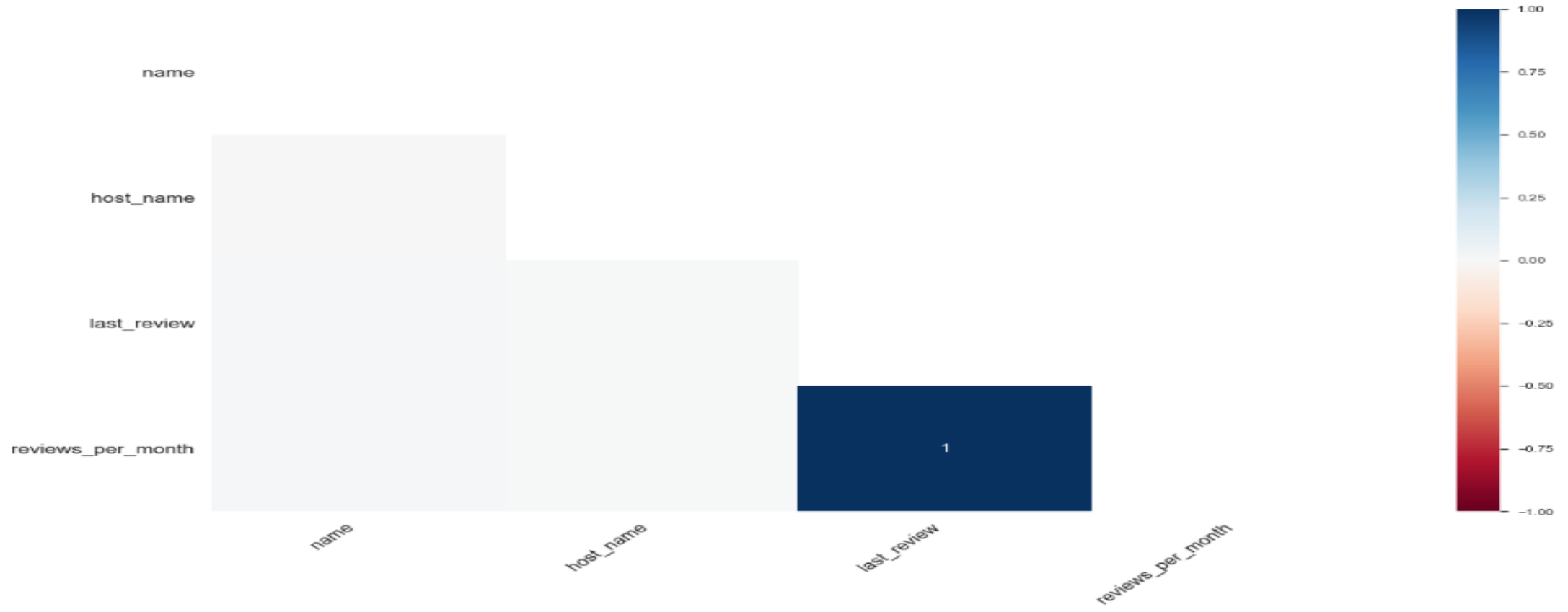
Finally, we will examine if we have data with `price` equals to '0' and we will be dropping the zero prices from the dataset.

3. Data Wrangling

We use a library to better visualize the missing values and the correlation between them to better know if they have type MCAR, MCR, MNAR.

```
ms.heatmap(df_listings)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2060b9be248>
```



There is a correlation between two columns "last_review" and "reviews_per_months" which is normal.

3. Data Wrangling

Data Cleaning

- First, we will check the insignificant features with respect to the target variable based on P-values by using as user-defined functions.

	Pearson-Coeff	P-Value	P-Ind	C-Ind
latitude	0.027656	7.539657e-10	Strong	Positive
longitude	-0.100169	1.419651e-110	Strong	Negative
minimum_nights	0.015939	3.904913e-04	Strong	Positive
number_of_reviews	-0.035733	1.825600e-15	Strong	Negative
reviews_per_month	-0.027681	7.281707e-10	Strong	Negative
calculated_host_listings_count	0.004707	2.949542e-01	Insignificant	Positive
availability_365	0.055743	2.250904e-35	Strong	Positive

From the above results we can see that the *"calculated_host_listings_counts"* has an insignificant relationship with the target variable *"price", so we will remove it.*

3. Data Wrangling

Data Cleaning

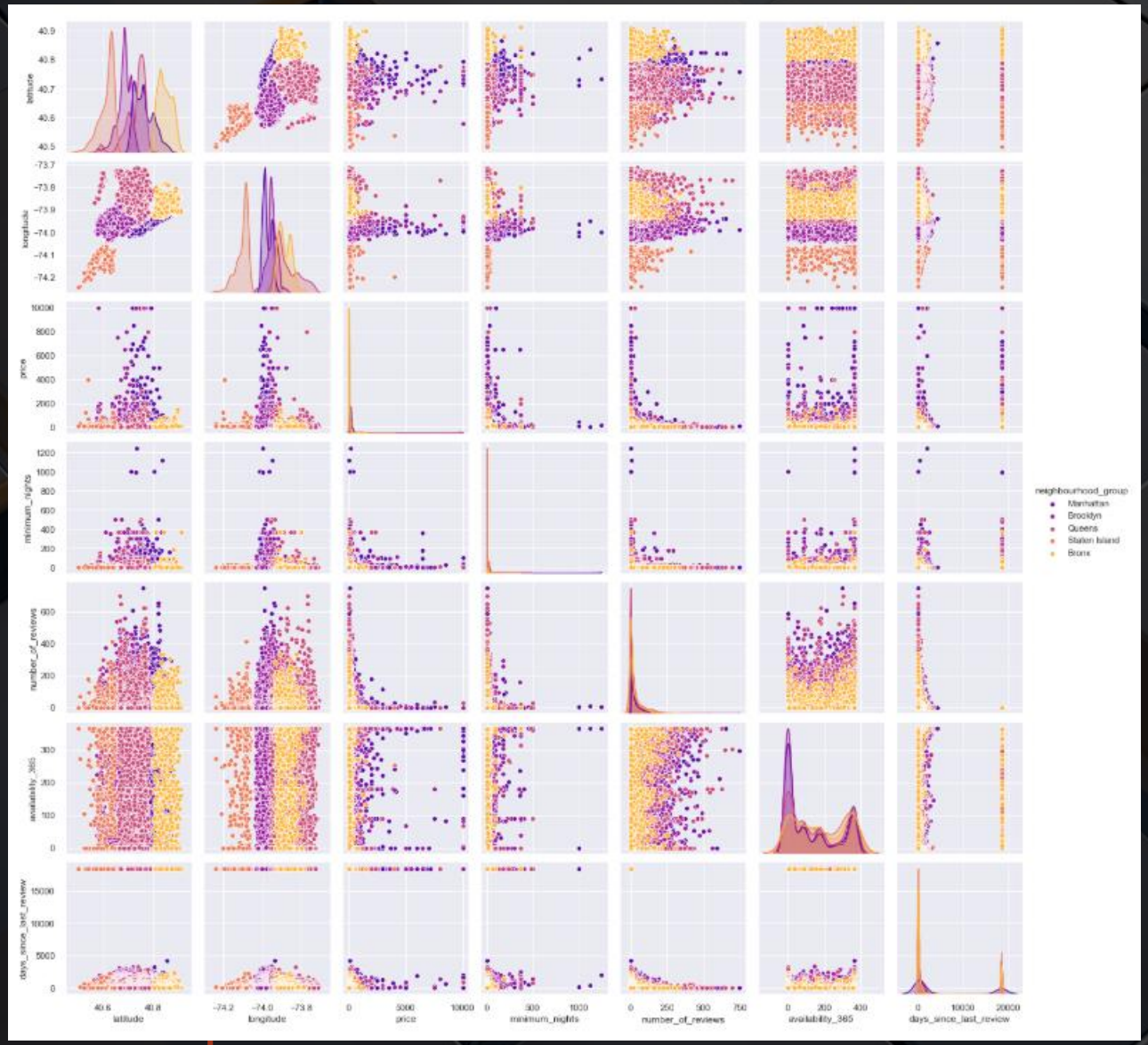
- Then, we check whether any multi-correlation is present between the independent variables in the dataset by user-defined functions.

```
    Correlation with more than : 0.5


Corr Value        Fearture1                                  Feature 2
0.7467            reviews_per_month                          number_of_reviews


 The above mentioned correlations only are present in the given Dataset
```

The correlation between *“reviews_per_month” and “number_of reviews” is about 0.75, hence we will remove one of them (“reviews_per_month”).*

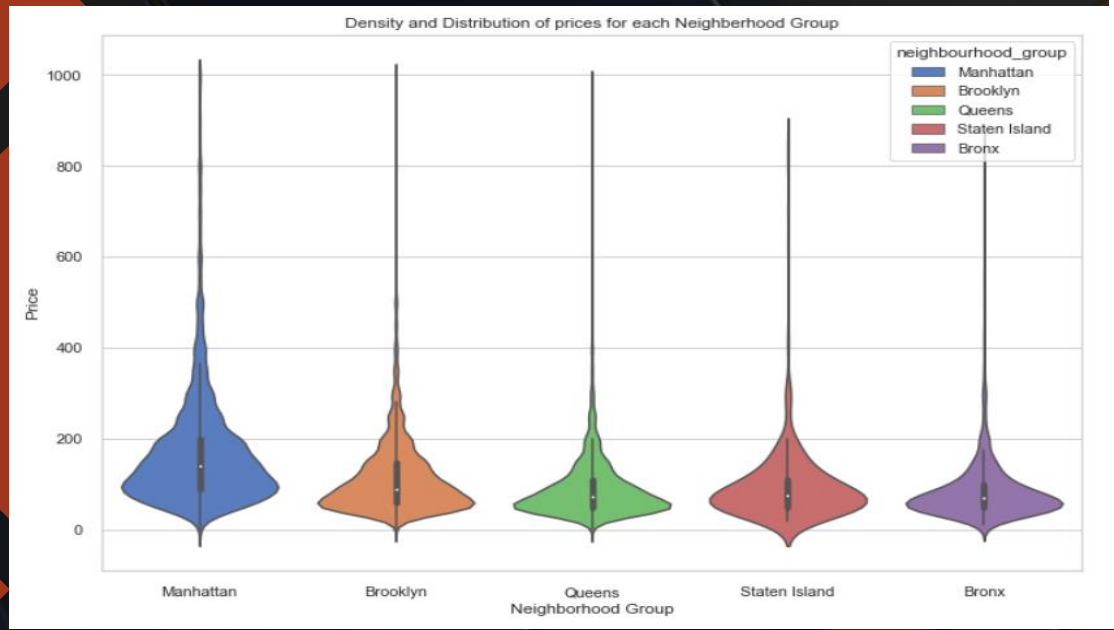
3. Data Wrangling

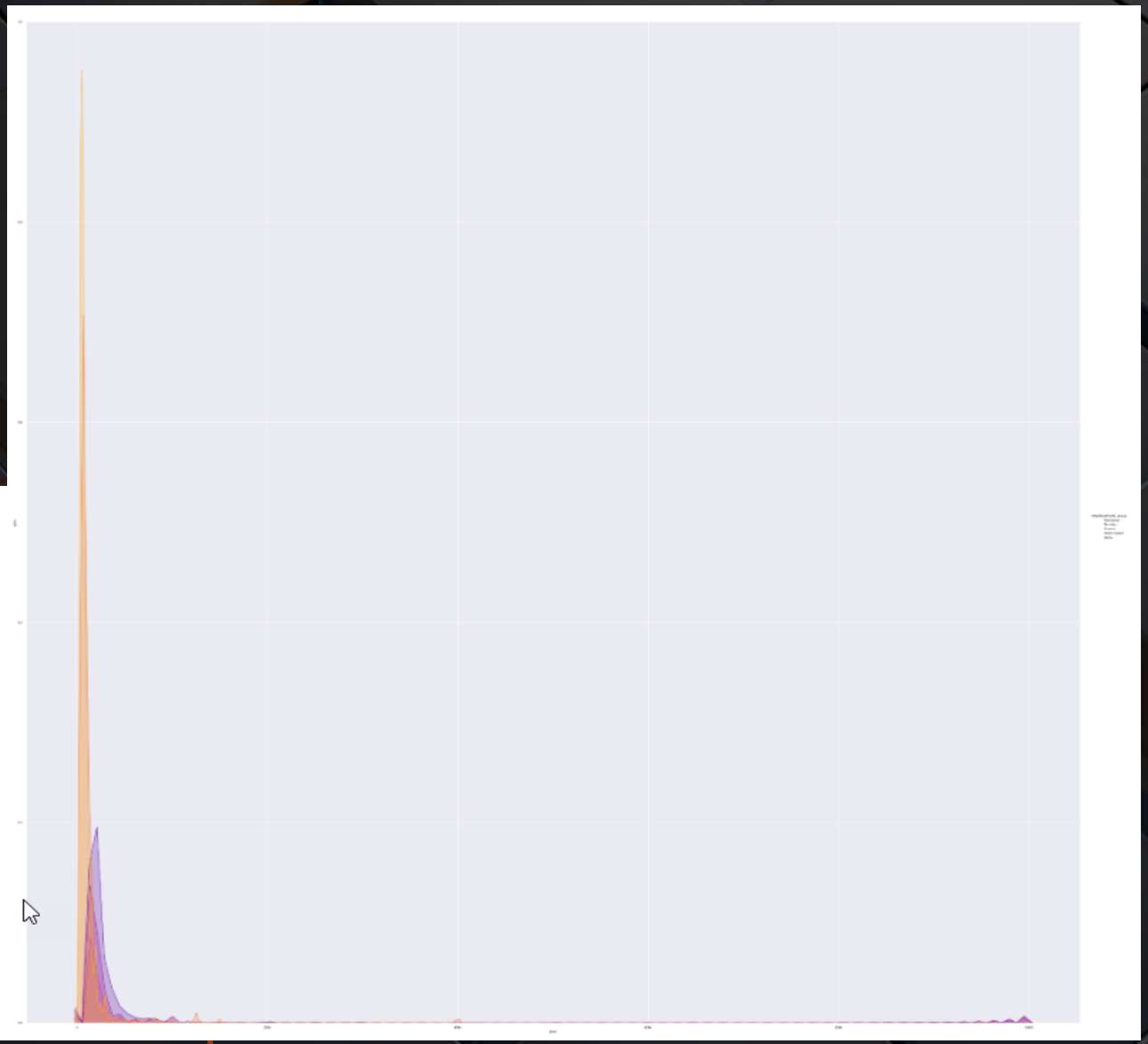
- Pairplot will be used to find whether any variable can be able to differentiate the price distribution. In our project, we will be using the “neighbourhood_group” feature.



3. Data Wrangling

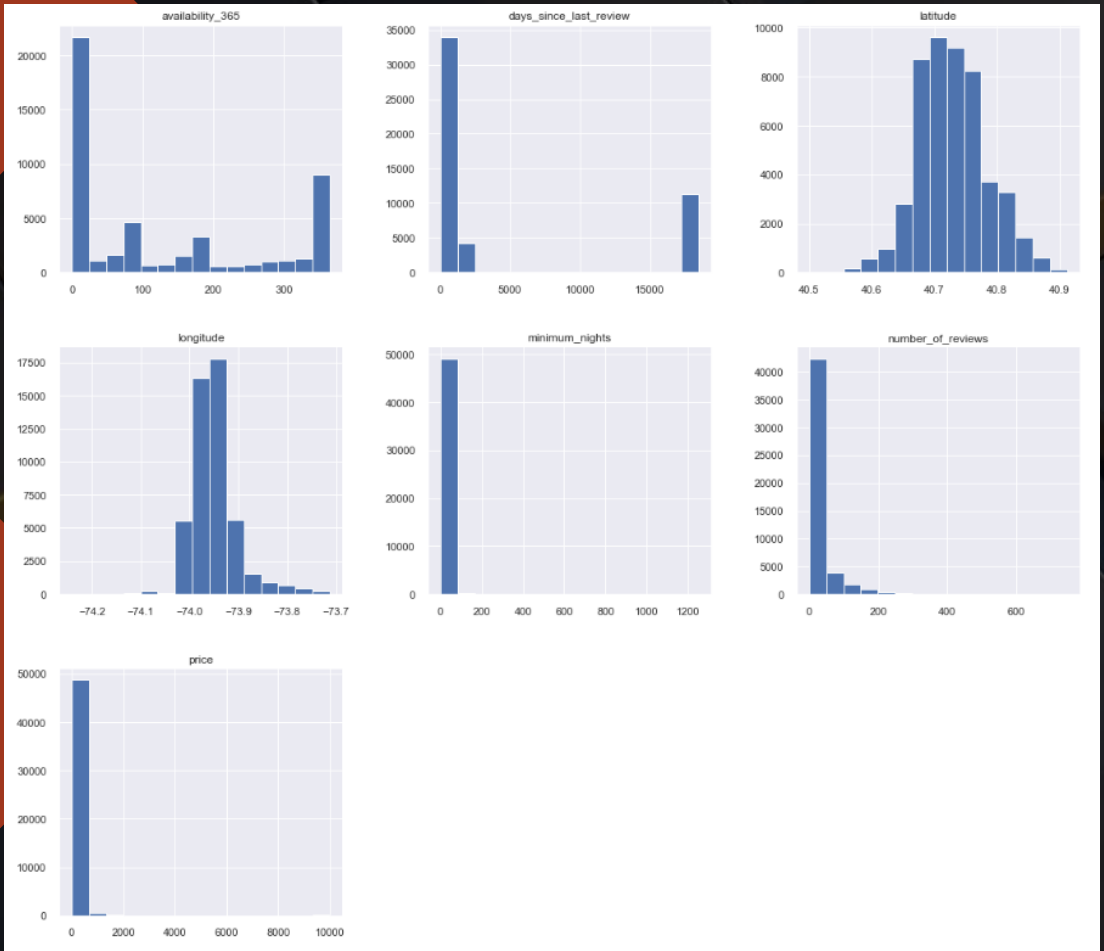
From the above plots, we can see that the price distribution (target variable) is heavily skewed towards right (positive) and the neighborhood group variable is not separating the price distribution well as it overlaps on each other. We observe the same through violin plot when we have tried to get know about the average price of rent based on neighborhood_group.





3. Data Wrangling

Statistical distribution of data



We can see in the histograms above that the variables “minimum_nights”, “number_of_reviews” and “price” have outliers.
To better understand these data, we examine the Statistical summary of the variables using the command describe().

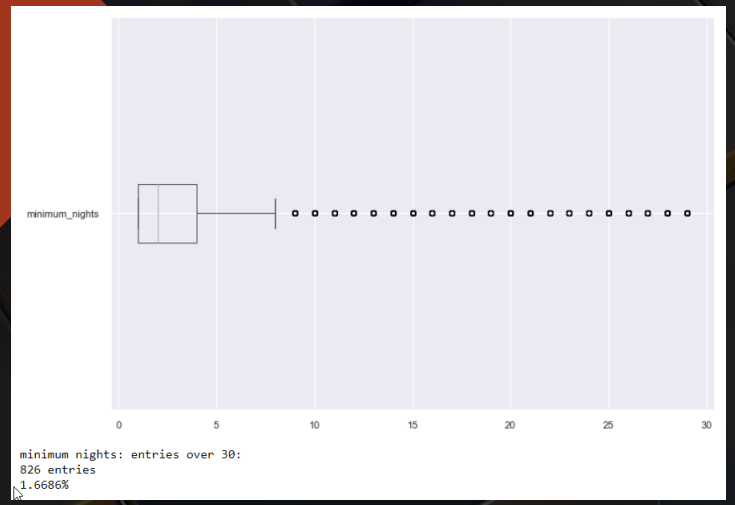
	availability_365	days_since_last_review	latitude	longitude	minimum_nights	number_of_reviews	price
count	49504.000000	49504.000000	49504.000000	49504.000000	49504.000000	49504.000000	49504.000000
mean	126.689621	4563.217700	40.729238	-73.951033	8.191257	23.870314	162.729295
std	142.380712	7569.010477	0.054684	0.047553	21.966500	48.246649	419.405845
min	0.000000	34.000000	40.499790	-74.240840	1.000000	0.000000	10.000000
25%	0.000000	147.000000	40.689810	-73.983360	2.000000	1.000000	68.000000
50%	79.000000	338.000000	40.723840	-73.955350	3.000000	5.000000	101.000000
75%	267.000000	1731.000000	40.762790	-73.934280	6.000000	23.000000	175.000000
max	365.000000	18456.000000	40.911690	-73.712990	1250.000000	746.000000	10000.000000

Looking at this summary, we can come to some conclusions:

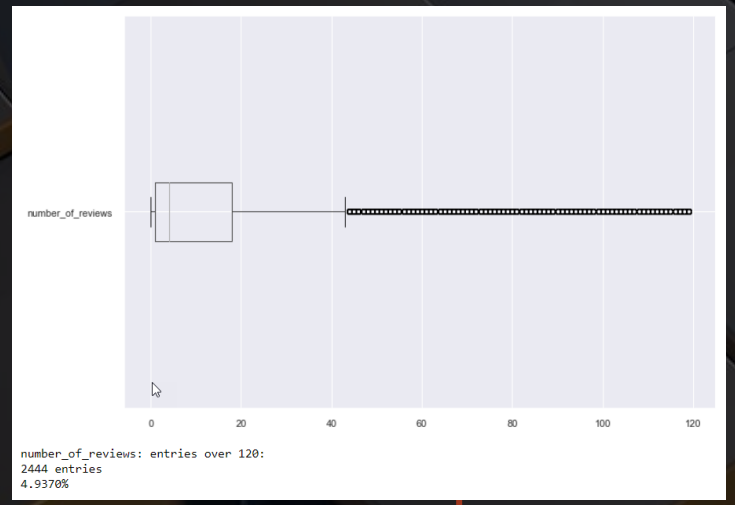
- The variable `minimum_nights` at its max is way over the real limit (365 days).
- The 75% values of all of the analyzed variables are a lot lower than the max for each variable.

3. Data Wrangling

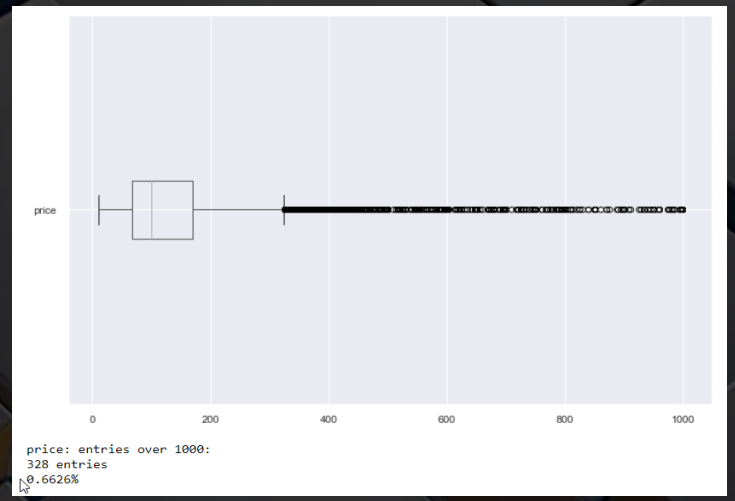
minimum_nights



number_of_reviews



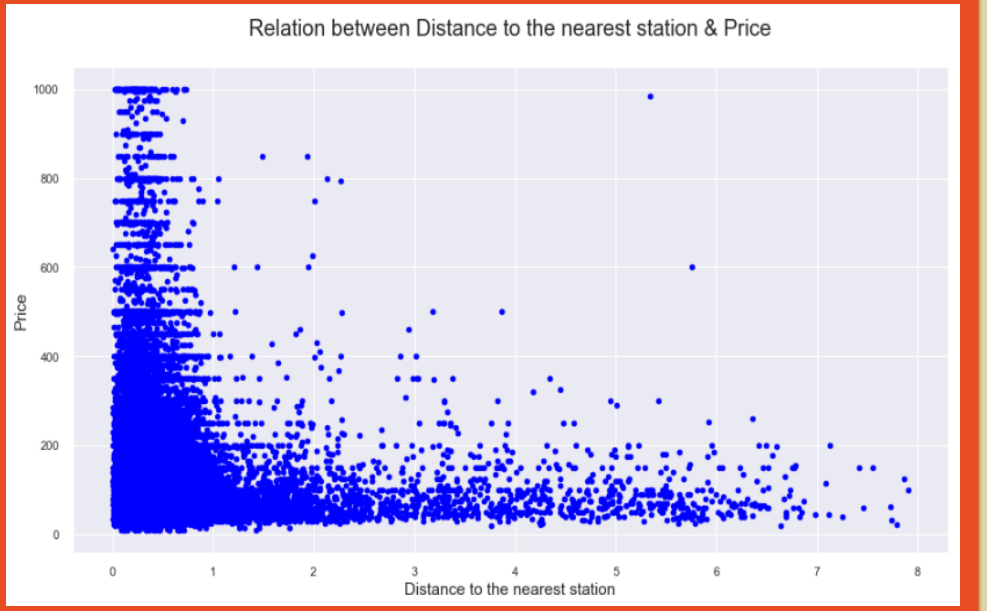
price

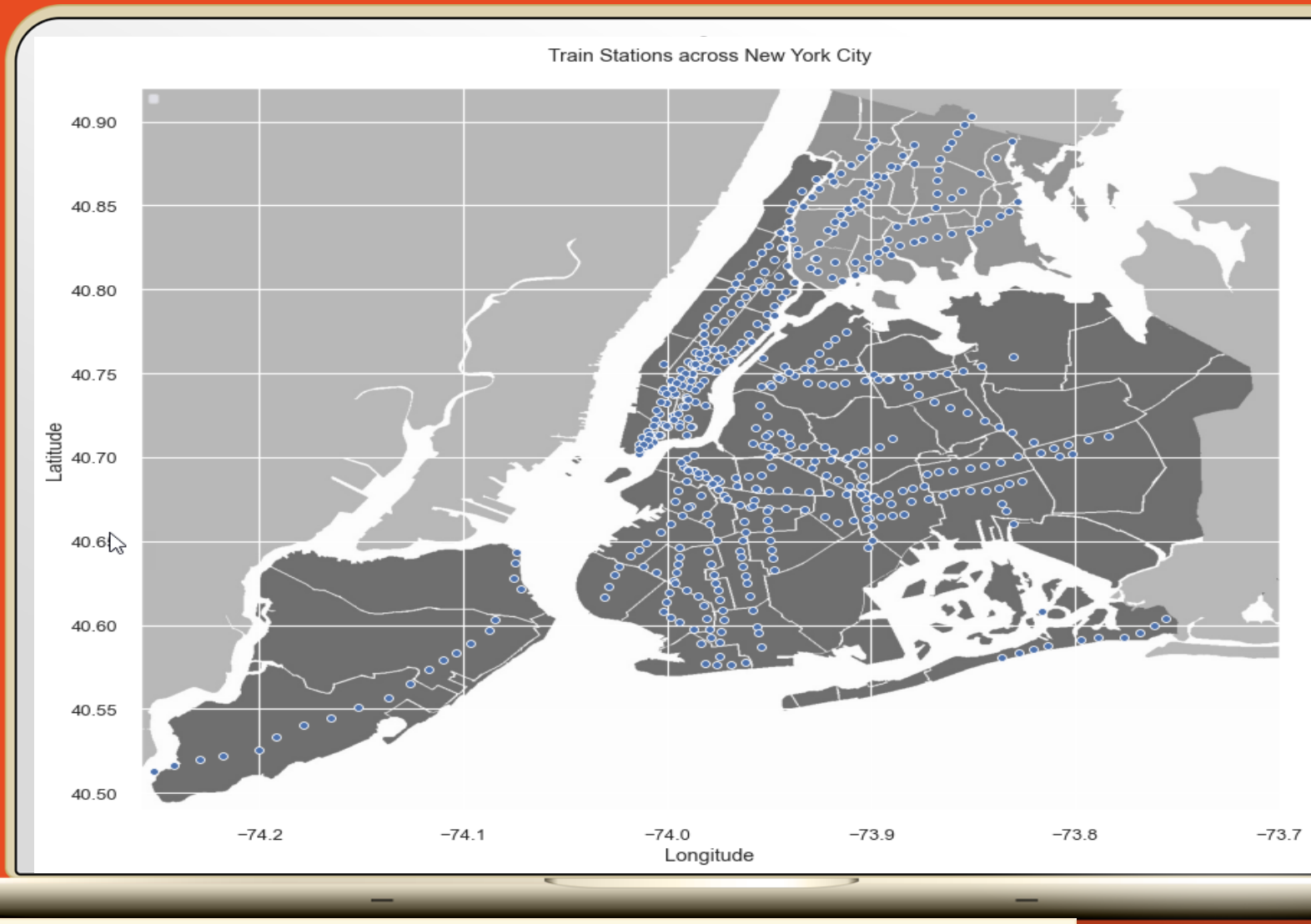


We decide to drop the following records:

- Listings whose “minimum_nights” is more than 30 days (826 entries or 1.7% of the whole dataset)
- Listings whose “number_of_reviews” is over 120 (2.444 entries or 4.9% of the whole dataset)
- Listings whose “price” exceeds $1000 (328 entries or 0.7% of the whole dataset)

4. ETL & Feature Engineering





For each Airbnb listing ID file, we go through each subway stop and calculate its distance from each subway stop. Then, we store the minimum distance value in a 'distance to station' array, so that we can keep track of the minimum distance to the subway for each Airbnb listing ID. Because this exercise is computationally expensive, we choose to export our data table to a new .csv file.

4. ETL & Feature Engineering

Skewness

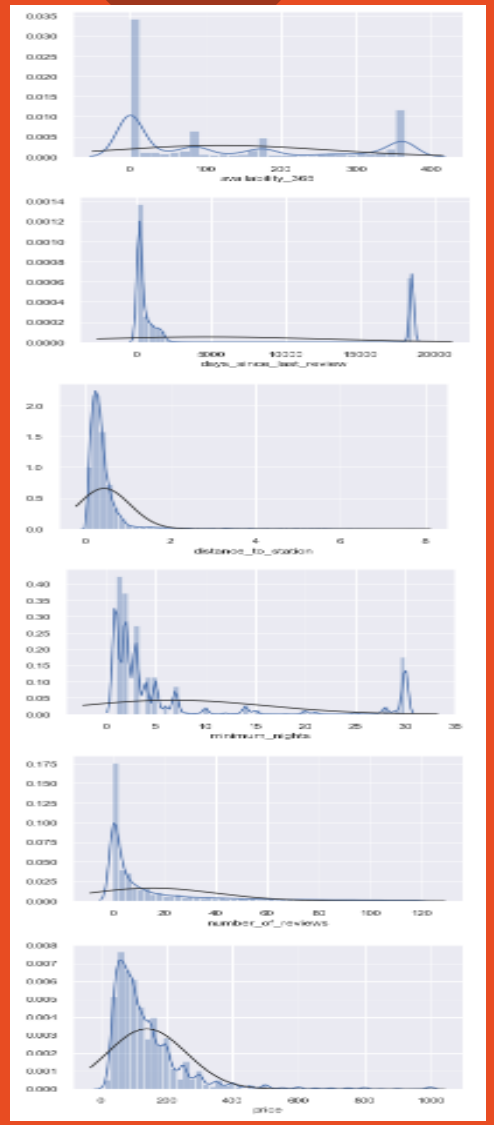
- We examine If an independent/dependent variable has a right skew (mean > median). Then, taking the log would make the distribution of our transformed variable appear more symmetric (more normal). There are no regression assumptions that require our independent or dependent variables to be normal. However, because we have outliers in our dependent and independent variables, a log transformation could reduce the influence of these observations.
- The variance of our regression residuals are increasing with our regression predictions. Taking the log of our dependent and independent variables may eliminate the heteroscedasticity of residuals and thereby improves the fitness of the model.

```
availability_365          0.741381
days_since_last_review    1.239384
distance_to_station       5.992169
minimum_nights            1.869446
number_of_reviews         2.189368
price                     3.165180
dtype: float64
```

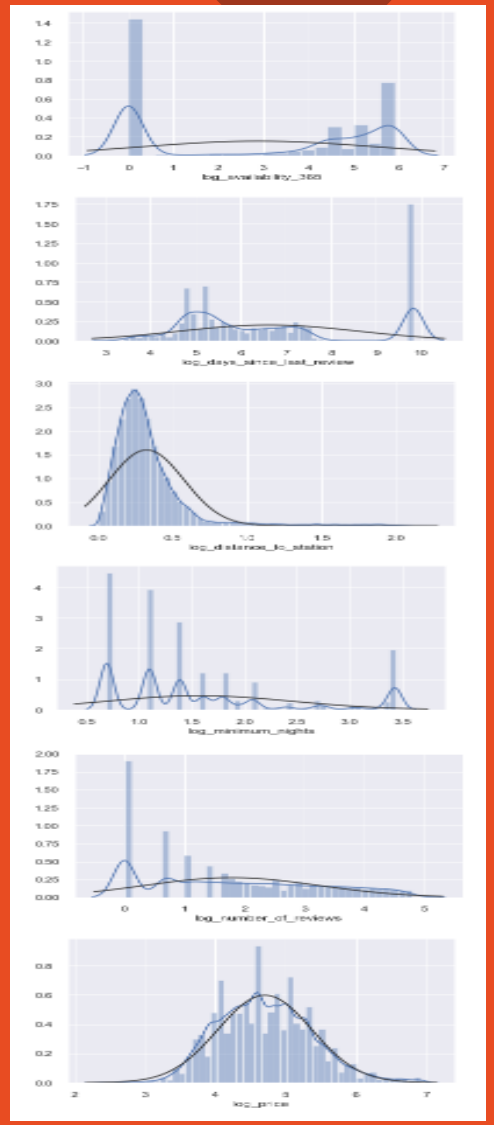
4. ETL & Feature Engineering

- availability_365
- days_since_last_review
- distance_to_station
- minimum_nights
- number_of_reviews
- price

Before Log Transformation



After Log Transformation



4. ETL & Feature Engineering

Data Preprocessing and Feature Engineering

- Remove columns not required for training of model (id, name, host_id and host_name).
- The columns “reviews_per_month” was also dropped as it had a strong correlation with the column “number_of_reviews”.
- An added feature “distance_to_station” was created from the dataset of Train Station locations.
- Numerical variables are normalized using logarithmic scaling.
- Categorical variables such as “neighborhood”, “neighborhood_group” and “room_type” are converting into dummy variables.

5. Data Partitioning

- We split/partition the data in to train and test split with 75% and 25% respectively.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=10)
```

```
print(X_train.shape,y_train.shape)
```

```
(34469, 237) (34469,)
```

```
print(X_test.shape,y_test.shape)
```

```
(11490, 237) (11490,)
```

- Then, we will using RobustScaler to scale the data to nullify the effect of outliers even after logarithmic transformation.

```
from sklearn.preprocessing import RobustScaler
scale= RobustScaler()
X_train_sc=scale.fit_transform(X_train)
X_test_sc=scale.transform(X_test)
```

```
print(X_train_sc.shape,X_test_sc.shape)
```

```
(34469, 237) (11490, 237)
```

6. Data Modeling

Baseline Models

1. **Linear Regression:** To capture the linear relationship between the dependent and the independent variable.

2. **Ridge Regression:** Regularized linear model which penalized the higher coefficients by using alpha parameter to eliminate the overfitting issues.

3. **Lasso Regression:** Least Absolute Shrinkage Selective Operator which penalizes the higher coefficients by using the parameter alpha to eliminate the overfitting issues and helps the feature selection.

6. Data Modeling

Ensemble methods

- An ensemble method is a Machine Learning technique that combines several base models in order to produce one optimal predictive model.

- When considering ensemble learning, there are two primary methods: bagging and boosting.
  - Bagging involves the training of many independent models and combines their predictions through some form of aggregation (averaging). An exemple of a bagging ensemble is a Random Forest.
  - Boosting instead trains models sequentially, where each model learns from the errors of the previous model. Starting with a weak base model, models are trained iteratively, each adding to the prediction of the previous model to produce a strong overall prediction.

6. Data Modeling

Ensemble models

4. **Decision Tree:** It uses a decision tree as a predictive model to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.
5. **KNN Regressor:** The KNN algorithm uses "features similarity" to predict the values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set.
6. **Random Forest:** Utilizes bagging (bootstrapping and aggregating) strategy with base eliminator as decision tree and creates a number of trees based on which the model predicts. Here all the trees are independent to each other.
7. **Gradient Boosting:** Gradient Boosting Regressor in which all the weak learners will be combined to form a strong learner. Here, decision tree is the base estimator and all the estimators are dependent on each other (learns by residuals of the previous decision tree).

6. Data Modeling

Ensemble models

9. **XGBoost Regressor:** Extreme Gradient Regressor provides an efficient implementation of the Gradient Boosting algorithm. The main benefit of the XGBoost implementation is computational efficiency and often better performance.
10. **Light GBM Regressor:** Light Gradient Boosting Machine is a gradient boosting framework that uses tree based algorithm. It splits the tree leaf wise with the best fit. So when growing on the same leaf, the leaf-wise algorithm can reduce more loss than the level-wise algorithm (like XGBoost) and hence results in much better accuracy.
11. **Artificial Neural Networks:** Artificial Neural Networks (ANN) are computational algorithms. It intended to simulate the behavior of biological systems composed of “neurons” and inspired by a central nervous system. It is capable of Machine Learning as well as pattern recognition.
12. **K Means Clustering:** K-Means Clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster.

6. Data Modeling

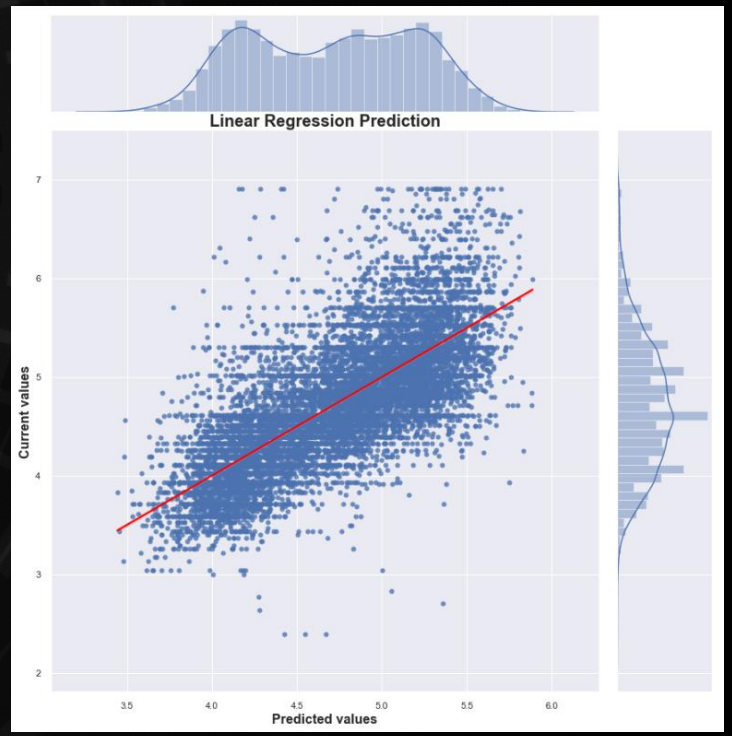
Model evaluation strategy

- **Mean Absolute Error (MAE):** MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It's the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.
- **Root mean squared error (RMSE):** RMSE is a quadratic scoring rule that also measures the average magnitude of the error. It's the square root of the average of squared differences between prediction and actual observation.
- **R^2 score:** The R^2 score also known as the coefficient of determination measures the amount of variance of the predictive model.
- **Training Error:** Training error is the error that we get when we run the trained model back on the training data. Remember that this data has already been used to train the model and this necessarily doesn't mean that the model once trained will accurately perform when applied back on the training data itself.
- **Test Error:** Test error is the error that we get when we run the trained model on a set of data that it has previously never been exposed to. This data is often used to measure the accuracy of the model before it is shipped to production.

6. Data Modeling

1 Linear Regression

	model	MAE	RMSE	Training_R2_score	Training_error	Test_R2_score	Test_error
0	Linear Regression	33.12%	44.19%	54.91%	20.17%	55.39%	19.53%



2 Ridge Regression

	model	MAE	RMSE	Training_R2_score	Training_error	Test_R2_score	Test_error
0	Ridge Regression	33.12%	44.17%	54.69%	20.27%	55.44%	19.51%

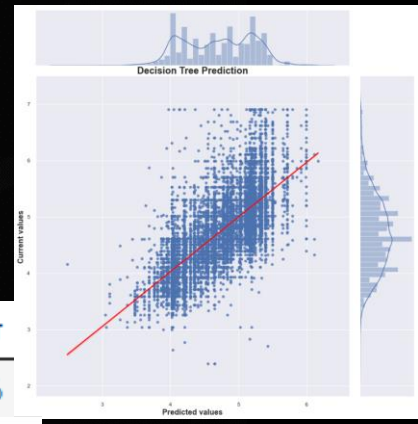
3 Lasso Regression

	model	MAE	RMSE	Training_R2_score	Training_error	Test_R2_score	Test_error
0	Lasso Regression	33.11%	44.15%	54.82%	20.21%	55.47%	19.49%

6. Data Modeling

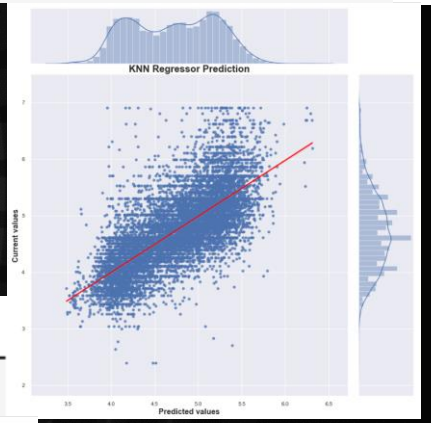
4 Decision Tree Regression

	model	MAE	RMSE	Training R2 Score	Training Error	Test R2 Score	Test Error
0	Decision Tree Regression	33.07%	44.42%	57.82%	18.87%	54.92%	19.73%



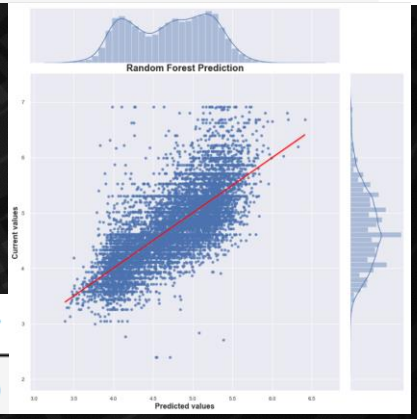
5 KNN Regression

	model	MAE	RMSE	Training R2 Score	Training Error	Test R2 Score	Test Error
0	KNN Regression	33.35%	44.50%	57.67%	18.93%	54.75%	19.81%



6 Random Forest Regression

	model	MAE	RMSE	Training R2 Score	Training Error	Test R2 Score	Test Error
0	Random Forest Regression	31.64%	42.61%	71.66%	12.65%	58.68%	18.15%



6. Data Modeling

The most important features

Most Important Variables

- `Entire home/apt` : This feature indicates that the type of the listing room type (entire home/apartment, Private room, Shared room and Hotel room). This is appealing to guests that prefer privacy and are willing to pay a premium for this feature. It makes sense that this would be one of the most important variables because two listings can have the same features and asthetics, but differ on price mainly due to the fact that an Entire home/apartment features one party as opposed to other room types that require the guest to share the space with others.
- `longitude` & `latitude` : This captures the location of the Airbnb listings. Certain neighborhoods and locations, due to their proximity to key attractions will have naturally more expensive properties than others, especially in Manhattan which is the most expensive borough in New York City.
- `distance_to_station` : As anticipated proximity to station to subway train and other transit hubs plays a big impact in pricing. The ease of transportation associated with living closer to a subway station is attached with a price premium.
- `days_since_last_review` & `number_of_reviews` : Reviews provide valuable insight as to how past guests have assessed the space that a prospective guest is staying in. The most recent and the more reviews there are, the more information there is for a prospective guest to make an informed decision.
- `minimum_nights` : The duration of the lease of the Airbnb property plays a decisive role in its rental price. The lower it is the duration, the higher will be its daily rental price.
- `availability_365` : This feature more broadly represents availability as a whole. This is potentially a feature that captures the supply aspect of the Airbnb listing. If there are a limited number of days for which the listing is available this supply factor may affect its rental price.

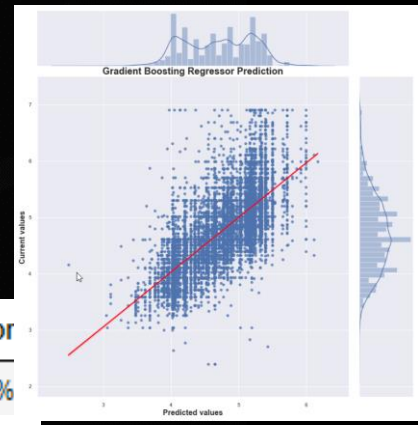
Top fifteen features by importance:

	Feature	Feature Importance
0	Entire home/apt	36.85%
1	longitude	15.88%
2	latitude	12.54%
3	log_distance_to_station	7.40%
4	log_days_since_last_review	5.82%
5	log_availability_365	5.65%
6	log_minimum_nights	4.40%
7	log_number_of_reviews	3.85%
8	Midtown	0.91%
9	Manhattan	0.68%
10	Hotel room	0.55%
11	Shared room	0.44%
12	Private room	0.30%
13	Upper East Side	0.14%
14	SoHo	0.13%

6. Data Modeling

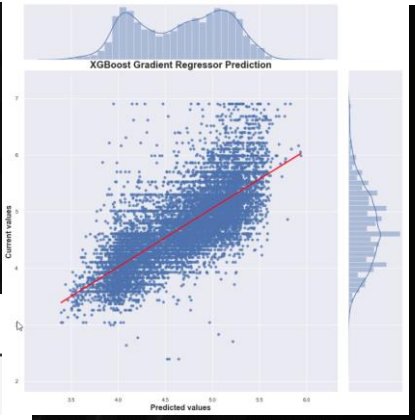
7 Gradient Boosting Regression

	model	MAE	RMSE	Training R2 Score	Training Error	Test R2 Score	Test Error
0	Gradient Booster Regression	31.70%	42.39%	61.25%	17.33%	58.94%	17.97%



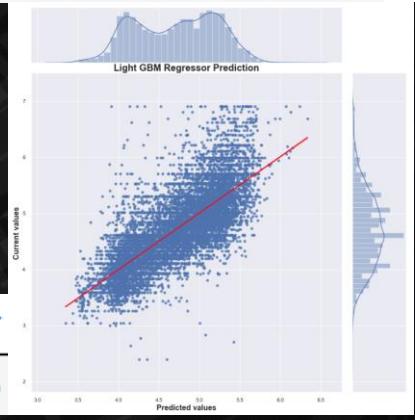
8 XGBoost Regression

	model	MAE	RMSE	Training R2 Score	Training Error	Test R2 Score	Test Error
0	XGBoost Gradient Regression	31.58%	42.75%	59.04%	18.32%	58.25%	18.28%



9 Light GBM Regression

	model	MAE	RMSE	Training R2 Score	Training Error	Test R2 Score	Test Error
0	Light GBM Regression	31.41%	42.05%	63.77%	16.21%	59.61%	17.68%



6. Data Modeling

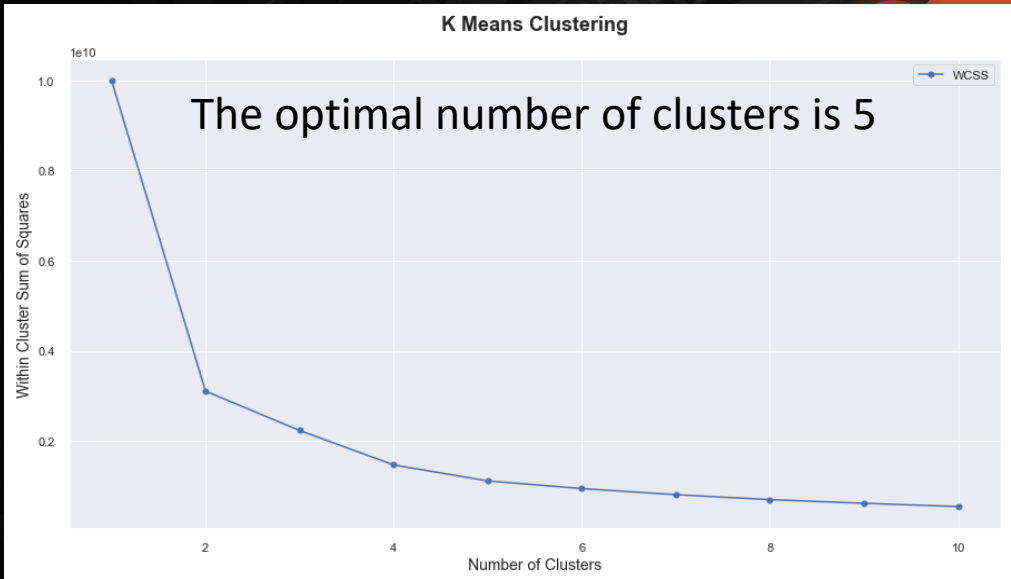
10. Artificial Neural Network

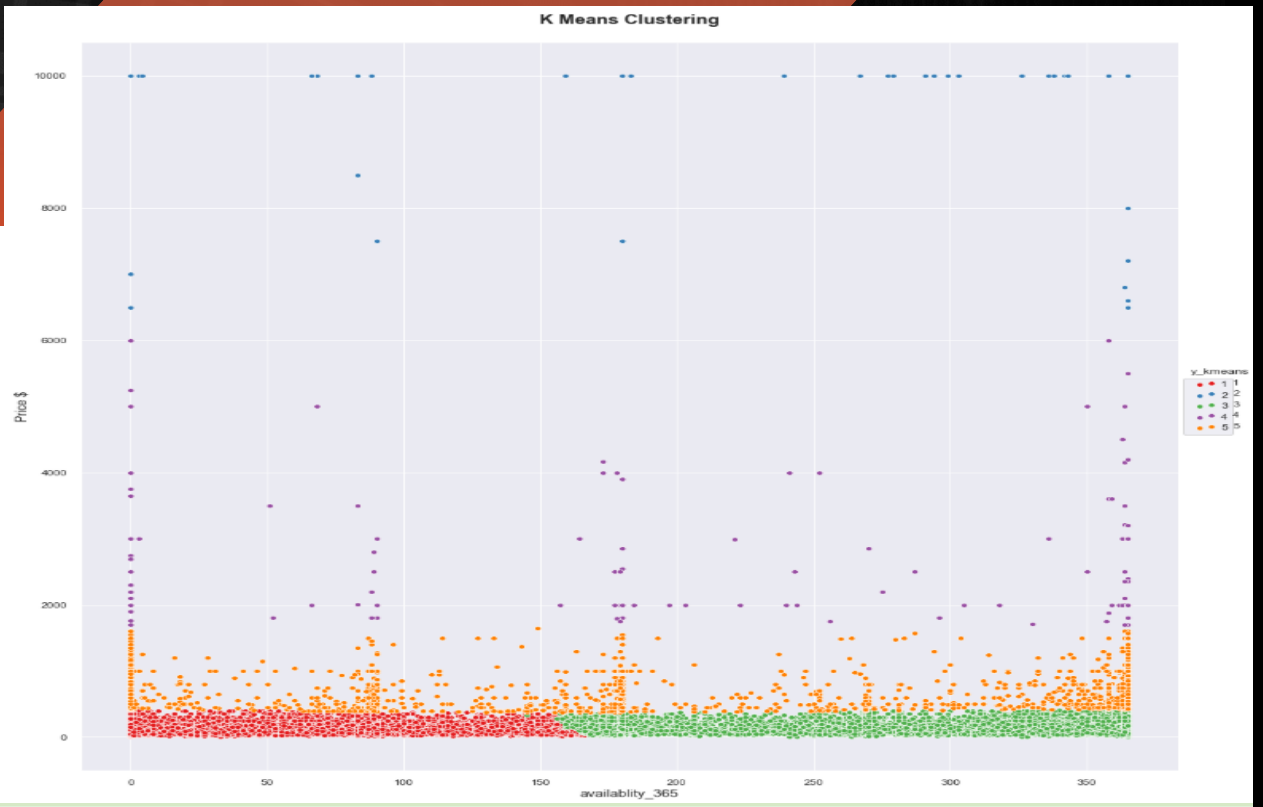
- An artificial Neural Network (ANN) was also trained on the data using Keras library on Tensorflow backend.
- The model was build using 2 hidden layers of 128 neurons on the first hidden layer and 64 neurons on the second layer.
- The “sigmoid” activation was applied on each layer except the output layer.
- A maximum of 50 epochs was trained using the “Adadelta” optimizer to measure the Mean Squared Error (MSE) and the Mean Absolute Error (MAE) of the ANN model.

	model	MAE	RMSE	Training R2 Score	Training Error	Test R2 Score	Test Error
0	Artificial Neural Network	34.33%	44.55%	54.14%	20.51%	54.65%	19.85%

6. Data Modeling

11. K-Means Clustering





Examining the plot we can see that the :

- Cluster 1 (RED) includes 30717 Airbnb listings having availability less than 160 days and price ranges from 0 to 400 dollars.
- Cluster 2 (BLUE) includes 78 Airbnb listings having all the availability and price ranges from 6000 to 10000 dollars.
- Cluster 3 (GREEN) includes 16891 Airbnb listings having availability more than 160 days and price ranges from 0 to 400 dollars.
- Cluster 4 (PURPLE) includes 127 Airbnb listings having all the availability and price ranges from 1600 to 5000 dollars.
- Cluster 5 (ORANGE) includes 1717 Airbnb listings having all the availability and price ranges from 400 to 1600 dollars.

7. Model Selection

Comparison of baseline models

	model	MAE	RMSE	Training R2 Score	Training Error	Test R2 Score	Test Error
0	Linear Regression	33.12%	44.19%	54.91%	20.17%	55.39%	19.53%
1	Ridge Regression	33.12%	44.17%	54.69%	20.27%	55.44%	19.51%
2	Lasso Regression	33.11%	44.15%	54.82%	20.21%	55.47%	19.49%

The results from the Lasso regression model are similar to the ones from Linear Regression model and Ridge Regression model. From the comparison table, we can see that Lasso Regression model based on selected features could explain approximately 55.5% of the variation of price. We must notice that the difference between the baseline models are barely visible.

7. Model Selection

Comparison of ensemble models

	model	MAE	RMSE	Training R2 Score	Training Error	Test R2 Score	Test Error
0	Decision Tree Regression	33.07%	44.42%	57.82%	18.87%	54.92%	19.73%
1	KNN Regression	33.35%	44.50%	57.67%	18.93%	54.75%	19.81%
2	Random Forest Regression	31.64%	42.61%	71.66%	12.65%	58.68%	18.15%
3	Gradient Booster Regression	31.70%	42.39%	61.25%	17.33%	58.94%	17.97%
4	XGBoost Gradient Regression	31.58%	42.75%	59.04%	18.32%	58.25%	18.28%
5	Light GBM Regression	31.41%	42.05%	63.77%	16.21%	59.61%	17.68%
6	Artificial Neural Network	34.33%	44.55%	54.14%	20.51%	54.65%	19.85%

From the result data frame, models adopting ensemble techniques performs better than linear models. Furthermore, ensemble models adopting boosting method have slightly better performance. Light GBM Regression model has the best performance of 59.61% prediction accuracy score. Gradient Boosting Regressor gives a close second score of 58.94%. In addition, there could be less concern of over-fitting in the Boosting Models due to the relatively lower differences between training and testing errors, even though there is no certain indication of over-fitting issue here. Overall, the three ensemble models have very similar performances. The only defect in boosting method is time-consuming due to serial-computing.


The end