SENG 474 Project

Data Mining: New York City Airbnb Open Data

Bay Shi	V00822560
Xu Wang	V00905790
Guangming Yang	V00821898

Table of Contents

1 Introduction

1.1 Background

Airbnb, also known as the most popular website for travellers to find a place to live. Airbnb has become the largest online marketplace for arranging or offering lodging or homestays in the world. With the impressive development in the lodging market, there is massive data from thousands of listing in different cities. Every owner can post his/her own place for visitors to stay, so in general, it has a huge dataset for owners, houses, and tenants. Price can be determined by the size of the place, the location of the house, different periods of time etc. Our goal is to use techniques and methodologies in data mining to predict the price.

As more data is obtained from more homeowners, different prices are listed on Airbnb. For example, Airbnb prices are different in different neighbourhoods, and prices are different due to different types of houses. We are going to analyze the price in different areas with techniques and methodologies and predict properties price with massive data.

1.2 Data Collection

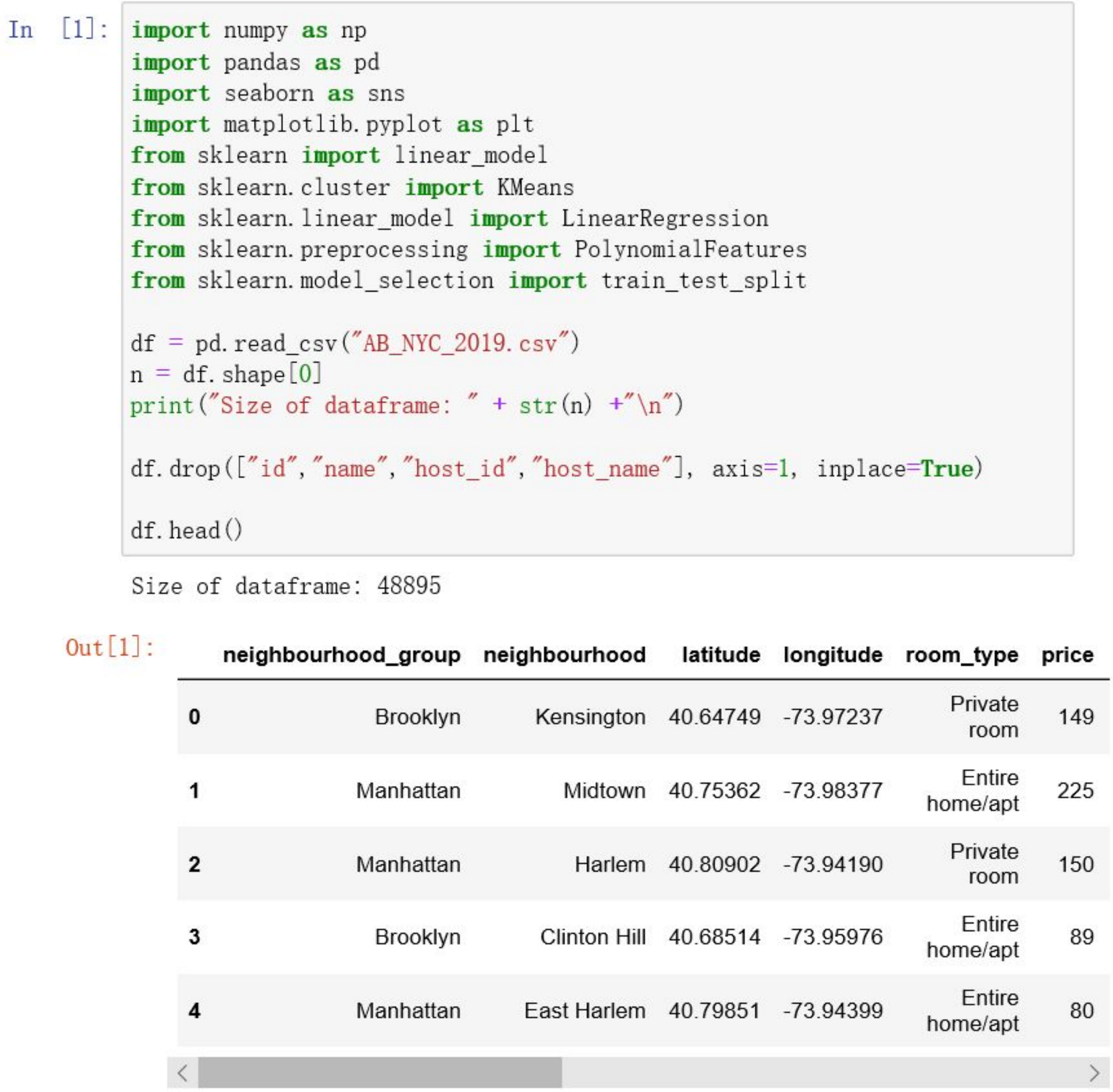
The data we are using is “New York City Airbnb Open Data” [1]. The data is the Airbnb price in different neighbourhoods in New York in 2019. In our data, there are different attributes that can influence the price of housing. The dataset is listed clearly with the following attributes:

- ID – the listing id from different owners on the website
- Name – Airbnb name
- Host_id – id of owners
- Host_name – the name of the host
- Neighborhood_group – location
- Neighbourhood – area
- Latitude – latitude coordinates
- Longitude – longitude coordinates
- Room_type – listing space type
- Price – price in dollars
- Minimum_nights – the amount of nights minimum
- Number_of_reviews – number of reviews
- Last_review – latest review
- Reviews_per_month – number of reviews per month
- Calculated_host_list – the amount of listing per host
- Availability_365 – number of days when the listing is available for booking

Features, such as “Neighborhood_group”, “Neighborhood”, “Latitude”, “Longitude”, “Price”, “Number_of_reviews” are categorical, so we drop the remaining data.

2 Data Preprocessing and Visualization

At first, we dropped some attributes that will not be used, so the attributes we are using is “Neighborhood_group”, “Neighborhood”, “Latitude”, “Longitude”, “Price”, “Number_of_reviews”.



```
In  [1]: import numpy as np
         import pandas as pd
         import seaborn as sns
         import matplotlib.pyplot as plt
         from sklearn import linear_model
         from sklearn.cluster import KMeans
         from sklearn.linear_model import LinearRegression
         from sklearn.preprocessing import PolynomialFeatures
         from sklearn.model_selection import train_test_split

         df = pd.read_csv("AB_NYC_2019.csv")
         n = df.shape[0]
         print("Size of dataframe: " + str(n) +"\n")

         df.drop(["id","name","host_id","host_name"], axis=1, inplace=True)

         df.head()

         Size of dataframe: 48895
```

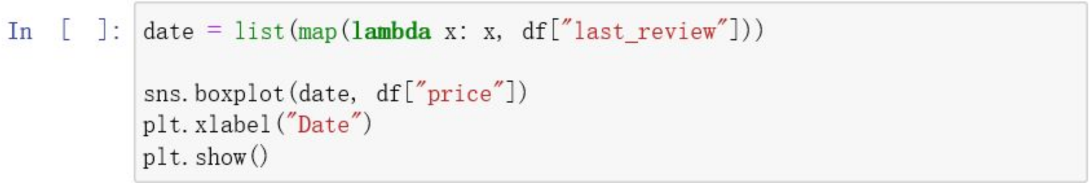
Out[1]:

	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price
0	Brooklyn	Kensington	40.64749	-73.97237	Private room	149
1	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225
2	Manhattan	Harlem	40.80902	-73.94190	Private room	150
3	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89
4	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80

2.1 Last Review Date

By observing the graph, there is no linear relationship between date and price. So the last review date will not be a significant attribute to affect the price.

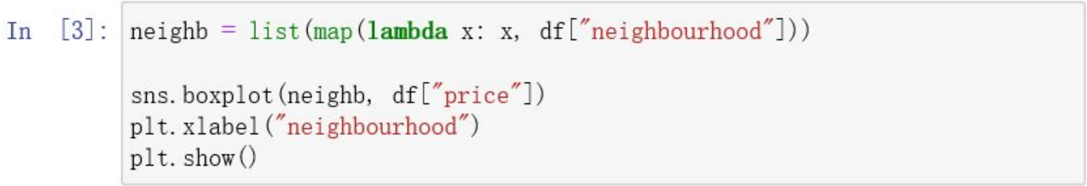
We grouped the dates for visualization:



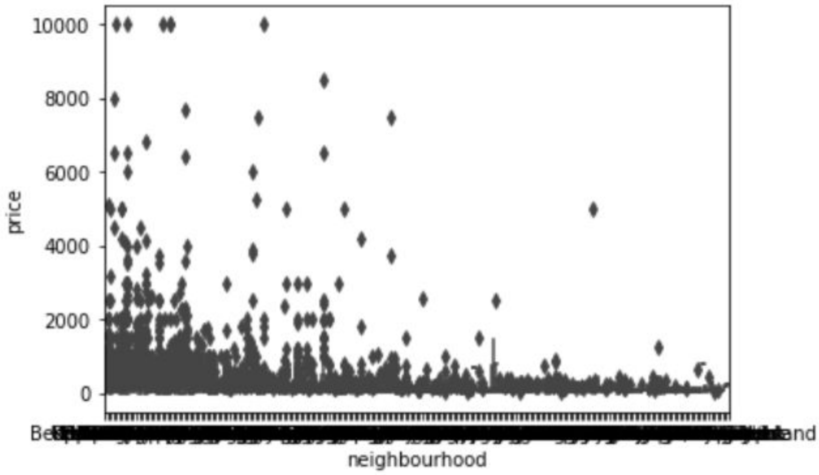
```
In [ ]: date = list(map(lambda x: x, df["last_review"]))

        sns.boxplot(date, df["price"])
        plt.xlabel("Date")
        plt.show()
```

2.2 Neighborhood

From the graph, there is a weak relationship between neighbourhood and price. As we can see from the graph, the closer the Airbnb is to downtown, the price is higher. And the quantities are much more than in other regions.

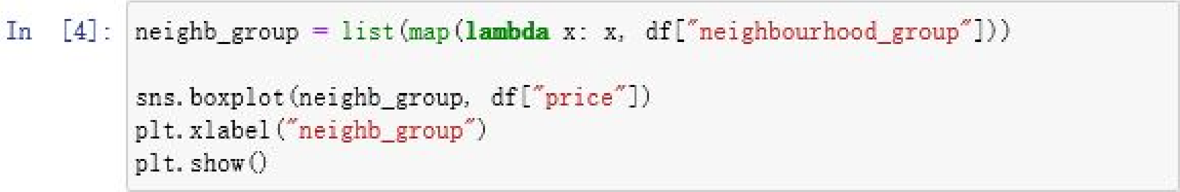


```
In [3]: neighb = list(map(lambda x: x, df["neighbourhood"]))

        sns.boxplot(neighb, df["price"])
        plt.xlabel("neighbourhood")
        plt.show()
```

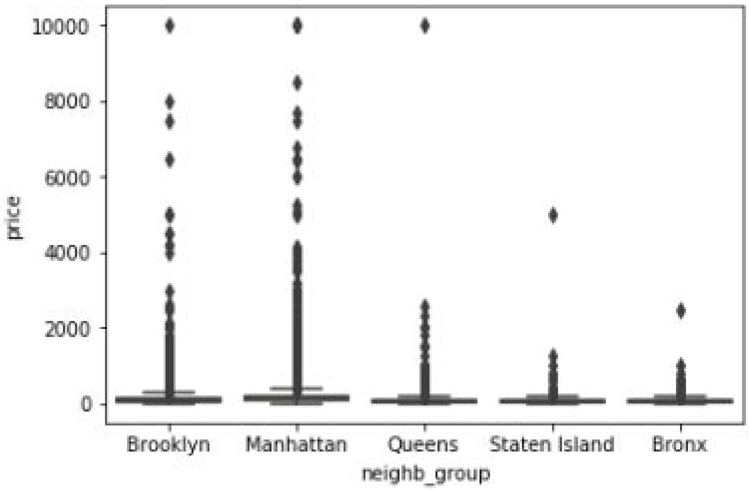


2.3 Neighborhood Group

As we can see from the graph, the Airbnb with the highest prices are in Manhattan, Brooklyn, and Queens. The most quantities of Airbnb we have in our dataset are in Manhattan. Moreover, the price in Brooklyn, Manhattan, and Queens are the highest among all neighbour groups.



```
In [4]: neighb_group = list(map(lambda x: x, df["neighbourhood_group"]))

        sns.boxplot(neighb_group, df["price"])
        plt.xlabel("neighb_group")
        plt.show()
```

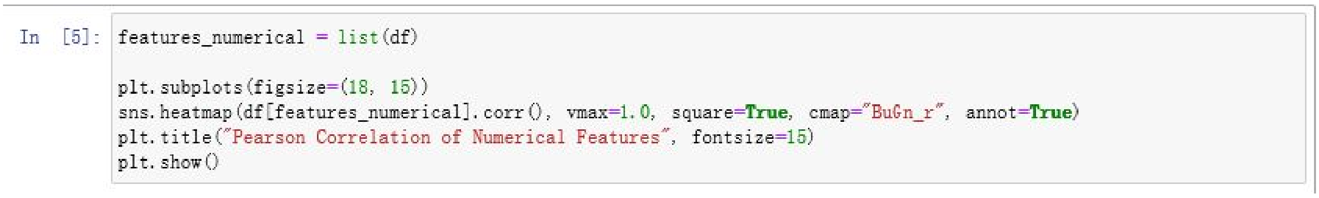


2.4 Pearson Correlation of Numerical Features

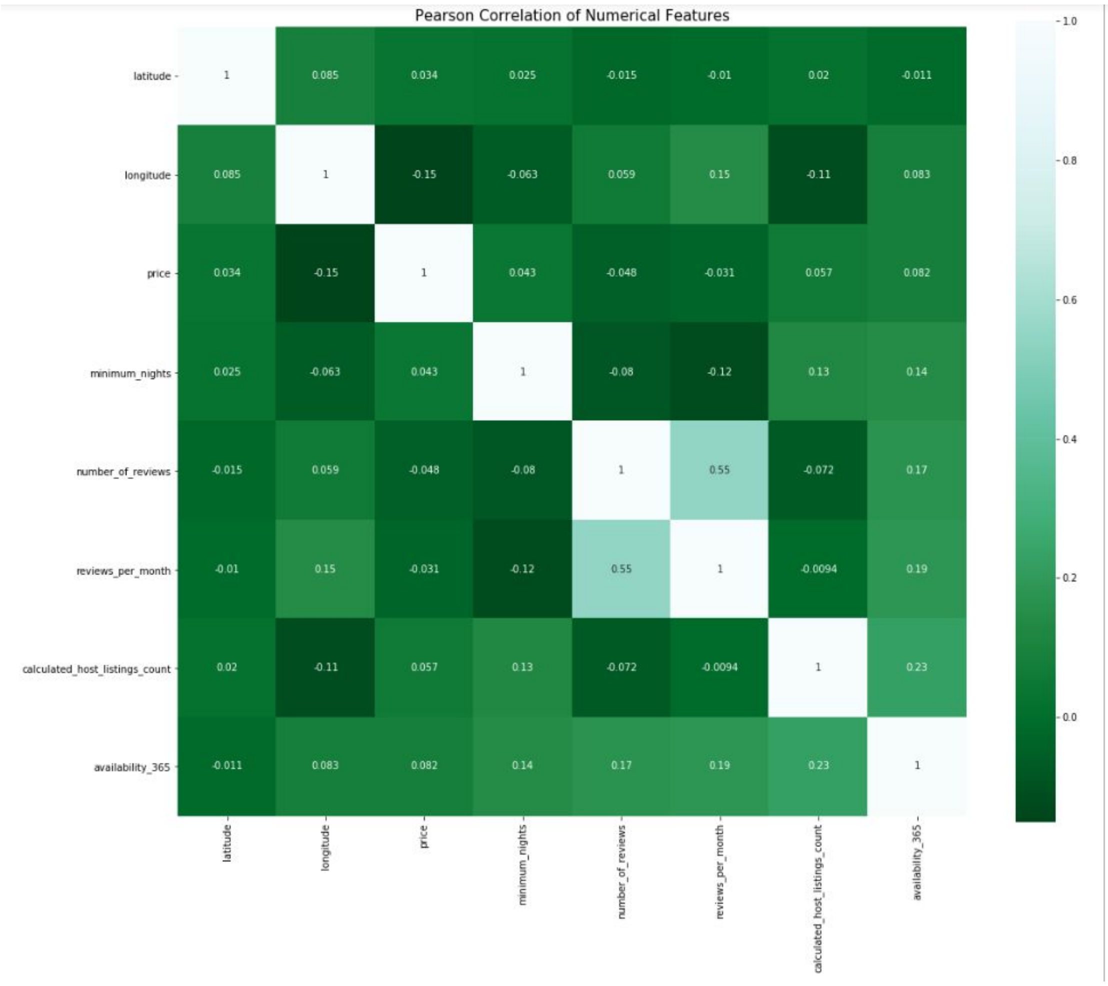
Pearson correlation is an intuitive method to perceive the linear relationship between two quantitative variables. This resulting value lies in [-1, 1], in which -1 means perfect negative correlation, +1 means perfect positive correlation and 0 means that there is no linear relation between two variables [2]. The formula of calculation of the Pearson Correlation Coefficient is [3]:

Where:

- μX is the mean of X
- μY is the mean of Y
- E is the expectation



```
In  [5]: features_numerical = list(df)

         plt.subplots(figsize=(18, 15))
         sns.heatmap(df[features_numerical].corr(), vmax=1.0, square=True, cmap="BuGn_r", annot=True)
         plt.title("Pearson Correlation of Numerical Features", fontsize=15)
         plt.show()
```



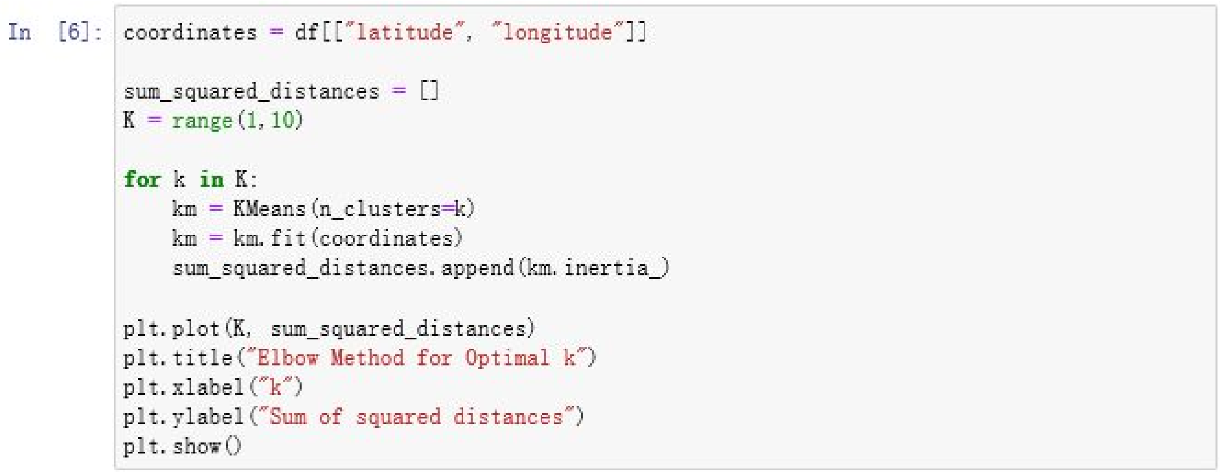
In order to reveal the Correlation Coefficient, we decided to visualize the correlation between two variables by using the heatmap, on which both x-axis and y-axis are the quantitative features of the data, and each cell is filled with a numerical value between -1 to 1, which represents how much two variables are linearly related to each other. In addition, a white-to-dark green is applied to encode different values in which white green indicates the perfect positive linear relationship, and dark green means the perfect negative. According to the definition of Correlation Coefficient, we imported the Seaborn library in python to generate the heatmap, as shown above, where all numerical features are included, and categorical features are exclusive. It is clear to perceive that all features have the perfect linear relation with itself, indicated by 1 in white green presented on the diagonal of the heatmap. For our purpose, we want to to find out what is the linear relationship between price and other features. “If the coefficient value lies between ±0.5 and ±1, then it is said to be a strong correlation. If the value lies between ±0.3 to ±0.49, it is said to be a medium correlation” [4]. We can eventually conclude from the heatmap that the number of host listing and the availability in one year are the two highest degrees of linear relation to the price. However, they are around a medium correlation. So the linear relationship is weak in our prediction.

3 Data Mining

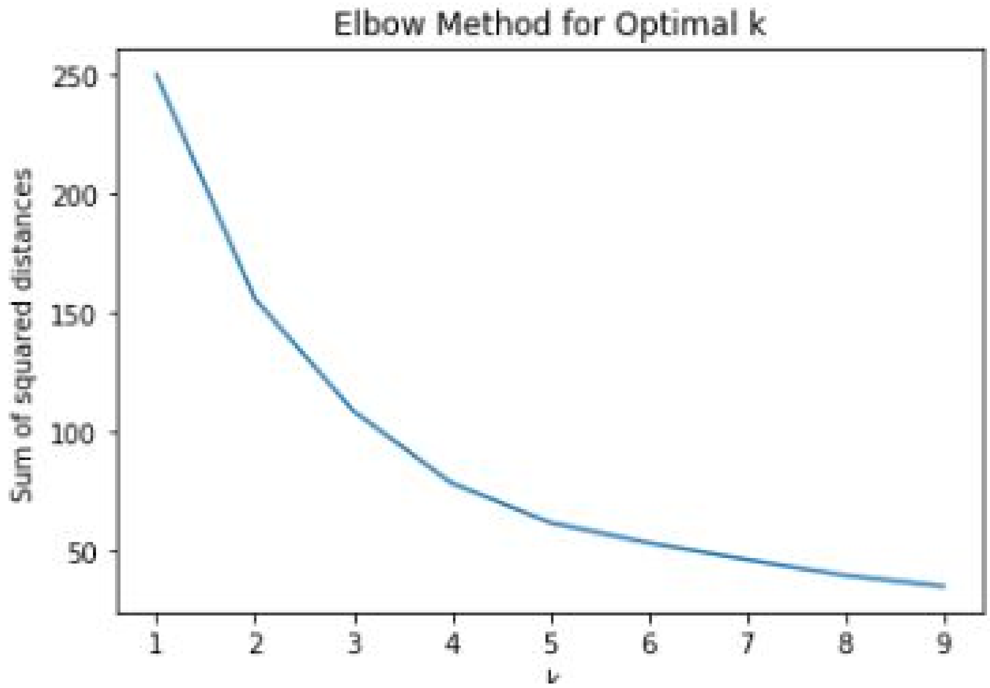
Data mining is the process to discover further information based on the large datasets and the implementing algorithm. In this project, the data set is the Airbnb price in KC and the implementing algorithm is linear regression.

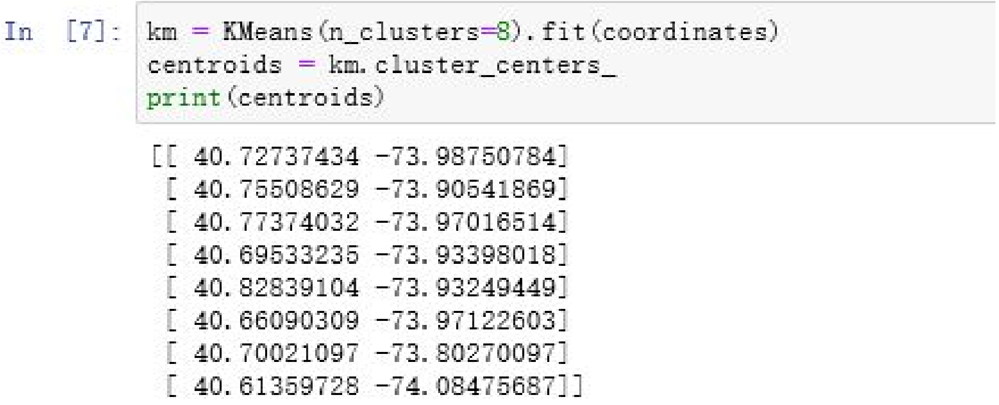
Taking lower computational costs of over 20,000 data into consideration, we took advantage of the holdout method to split data prior to the evaluation. The holdout method has simply split the dataset randomly into two sets, called the training set and the test set [5]. The size of each set is arbitrary, but the ratio between the training set and testing set typically is 8:2, 7:3 or 6:4. Based on the Pareto Principle [6], we end up separating 80% of the dataset into the training set and 20% are test set [7].

The ratio would affect the evaluation result, this will be explicitly discussed in 4.2. In addition, the dataset is randomly separated each time. Therefore, the resulting accuracy is slightly different.



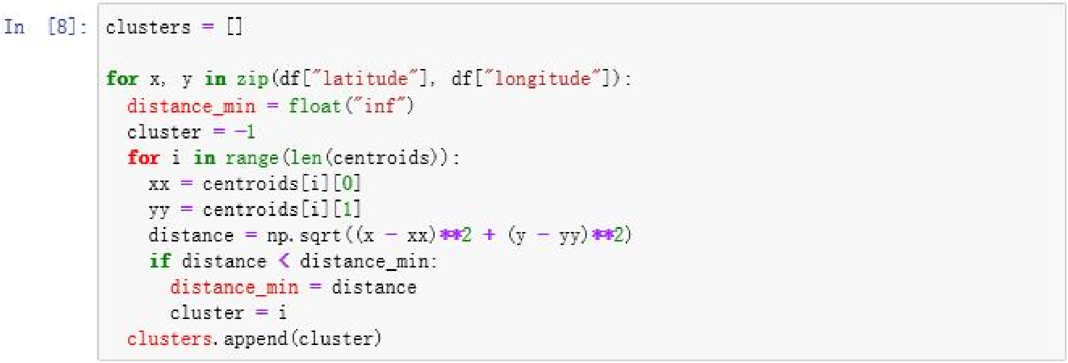
```
In  [6]: coordinates = df[["latitude", "longitude"]]

         sum_squared_distances = []
         K = range(1, 10)

         for k in K:
             km = KMeans(n_clusters=k)
             km = km.fit(coordinates)
             sum_squared_distances.append(km.inertia_)

         plt.plot(K, sum_squared_distances)
         plt.title("Elbow Method for Optimal k")
         plt.xlabel("k")
         plt.ylabel("Sum of squared distances")
         plt.show()
```



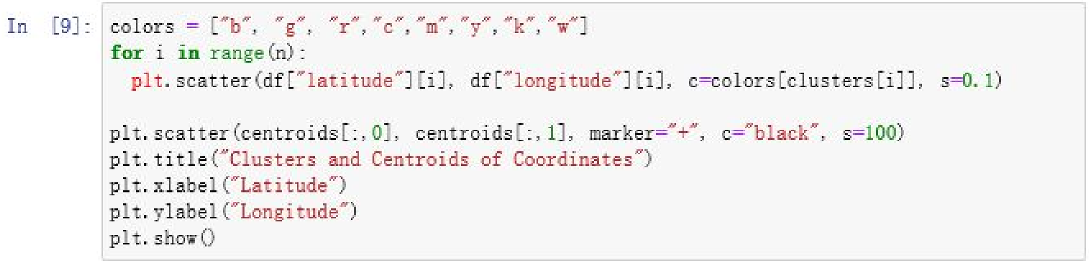


```
In  [7]: km = KMeans(n_clusters=8).fit(coordinates)
         centroids = km.cluster_centers_
         print(centroids)
```

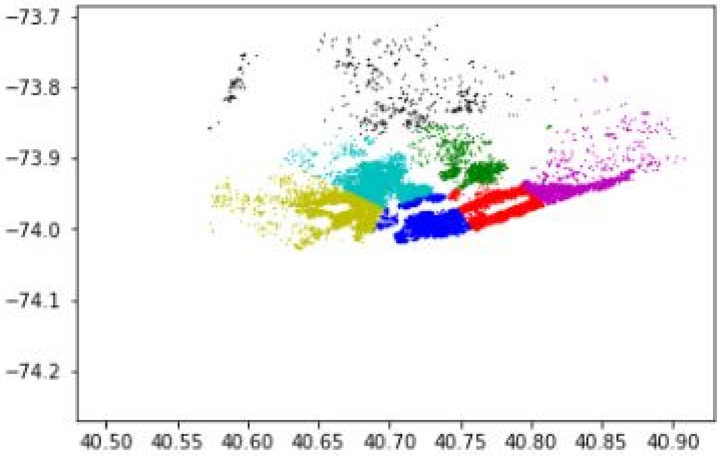
```
[[ 40.72737434 -73.98750784]
 [ 40.75508629 -73.90541869]
 [ 40.77374032 -73.97016514]
 [ 40.69533235 -73.93398018]
 [ 40.82839104 -73.93249449]
 [ 40.66090309 -73.97122603]
 [ 40.70021097 -73.80270097]
 [ 40.61359728 -74.08475687]]
```

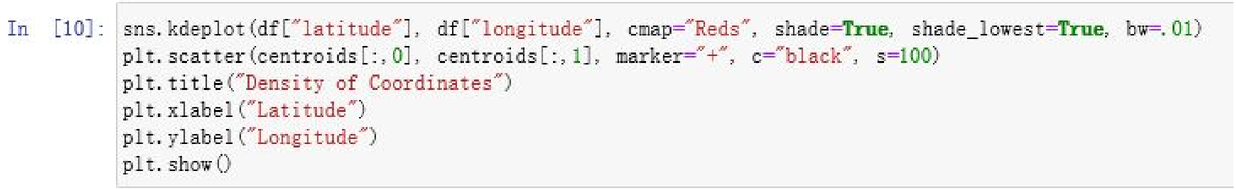


```
In  [8]: clusters = []

         for x, y in zip(df["latitude"], df["longitude"]):
           distance_min = float("inf")
           cluster = -1
           for i in range(len(centroids)):
             xx = centroids[i][0]
             yy = centroids[i][1]
             distance = np.sqrt((x - xx)**2 + (y - yy)**2)
             if distance < distance_min:
               distance_min = distance
               cluster = i
           clusters.append(cluster)
```

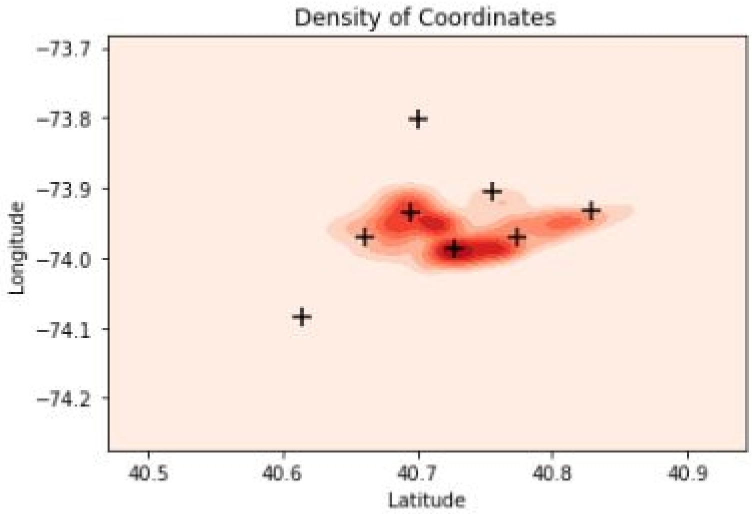


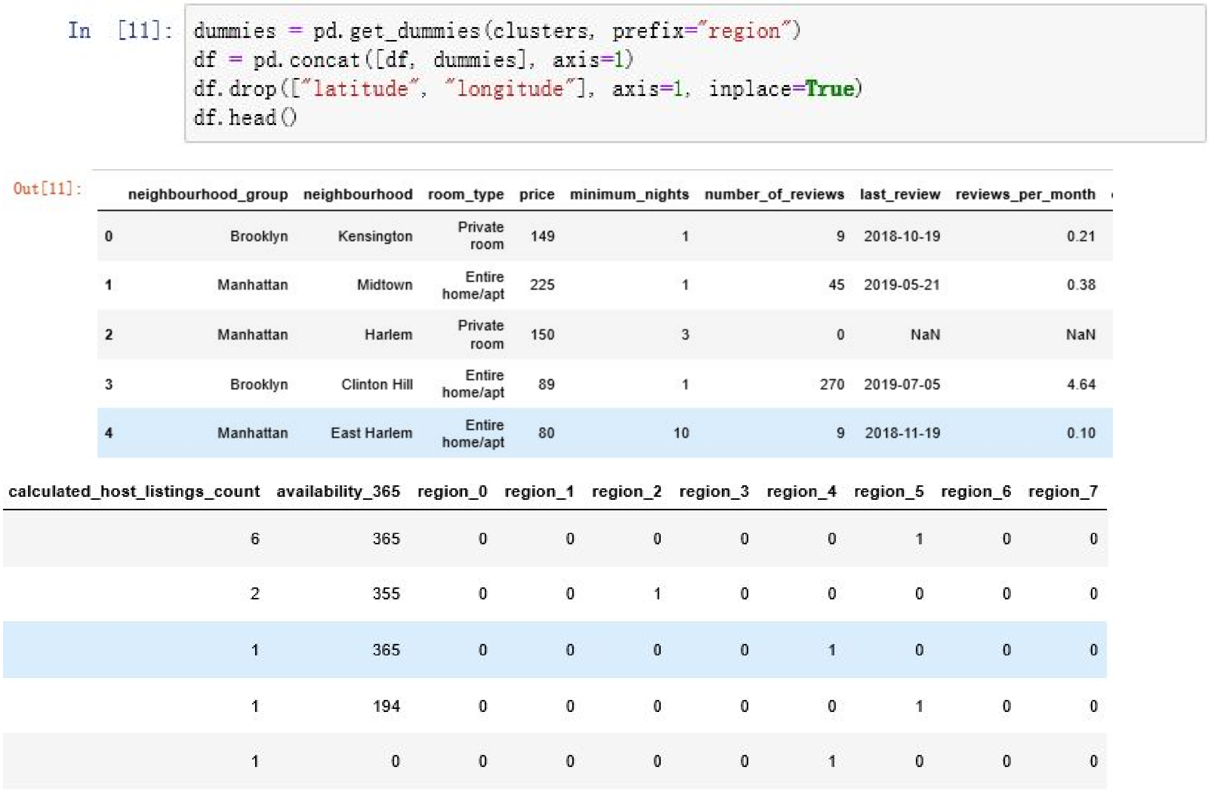
```
In [9]: colors = ["b", "g", "r","c","m","y","k","w"]
        for i in range(n):
          plt.scatter(df["latitude"][i], df["longitude"][i], c=colors[clusters[i]], s=0.1)

        plt.scatter(centroids[:,0], centroids[:,1], marker="+", c="black", s=100)
        plt.title("Clusters and Centroids of Coordinates")
        plt.xlabel("Latitude")
        plt.ylabel("Longitude")
        plt.show()
```





```
In [10]: sns.kdeplot(df["latitude"], df["longitude"], cmap="Reds", shade=True, shade_lowest=True, bw=.01)
         plt.scatter(centroids[:,0], centroids[:,1], marker="+", c="black", s=100)
         plt.title("Density of Coordinates")
         plt.xlabel("Latitude")
         plt.ylabel("Longitude")
         plt.show()
```

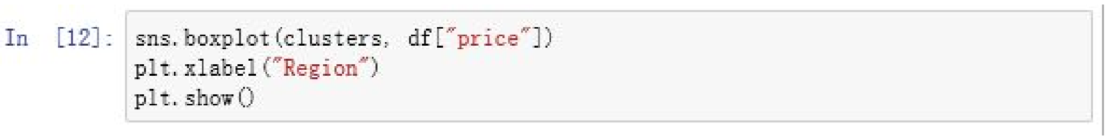




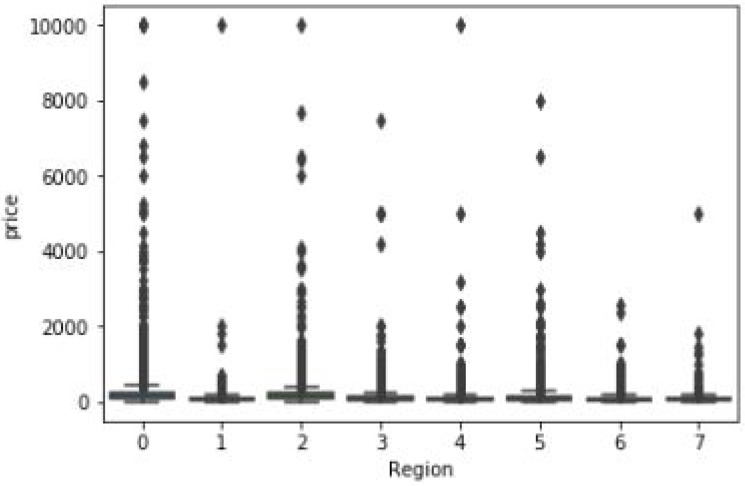
In [11]:
```python
dummies = pd.get_dummies(clusters, prefix="region")
df = pd.concat([df, dummies], axis=1)
df.drop(["latitude", "longitude"], axis=1, inplace=True)
df.head()
```

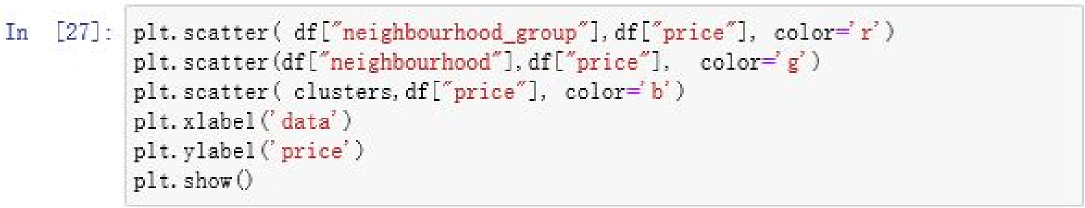
Out[11]:

	neighbourhood_group	neighbourhood	room_type	price	minimum_nights	number_of_reviews	last_review	reviews_per_month	calculated_host_listings_count	availability_365	region_0	region_1	region_2	region_3	region_4	region_5	region_6	region_7
0	Brooklyn	Kensington	Private room	149	1	9	2018-10-19	0.21	6	365	0	0	0	0	0	1	0	0
1	Manhattan	Midtown	Entire home/apt	225	1	45	2019-05-21	0.38	2	355	0	0	1	0	0	0	0	0
2	Manhattan	Harlem	Private room	150	3	0	NaN	NaN	1	365	0	0	0	0	1	0	0	0
3	Brooklyn	Clinton Hill	Entire home/apt	89	1	270	2019-07-05	4.64	1	194	0	0	0	0	0	1	0	0
4	Manhattan	East Harlem	Entire home/apt	80	10	9	2018-11-19	0.10	1	0	0	0	0	0	1	0	0	0

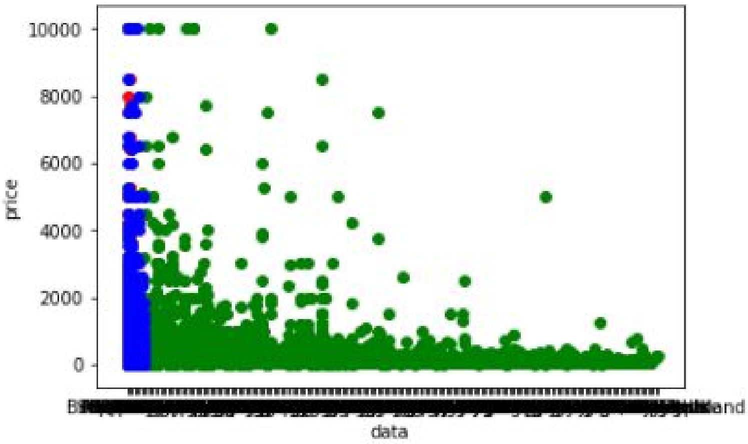


In [12]:
```python
sns.boxplot(clusters, df["price"])
plt.xlabel("Region")
plt.show()
```





```
In  [27]: plt.scatter( df["neighbourhood_group"],df["price"], color='r')
          plt.scatter(df["neighbourhood"],df["price"],  color='g')
          plt.scatter( clusters,df["price"], color='b')
          plt.xlabel('data')
          plt.ylabel('price')
          plt.show()
```



4 Evaluation

Since we have already visualized and processed the data, we can start to build our model with algorithms, linear regression and polynomial regression. The linear regression is polynomial regression with degree equals to 1.

4.1 Linear Regression

Linear regression is a method to model the relationship between two or more variables. We fed the training dataset to the regression, then run the test set on the trained algorithm. Our primary goal is to evaluate the model, hence Relative Squared Error [8], which was encapsulated in sklearn module, was used to calculate the accuracy:

$$RSE = \frac{\sum_{i=1}^{n}(p_i - a_i)^2}{\sum_{i=1}^{n}(\bar{a} - a_i)^2}$$

Where:

- p = predicted target
- a = actual target

However, there was a variety of features, in order to avoid both underfitting and overfitting, we needed to analyze what degree of polynomial features results in the optimal accuracy.

Degree	Accuracy(near dt)	Accuracy(far from dt)
1	73.4%	72.8%
2	81.7%	80.6%
3	Negative	Negative

4.2 Influence of testset ratio

The ratio of testset also made an influence on accuracy. Here, we compared the result of 3 different ratios (0.5, 0.2, 0.1), with the degree of polynomial features of 2.

Ratio of testset size = 0.5

Time	Accuracy(near dt)	Accuracy(far from dt)
1	83.4%	79.5%
2	82.3%	79.3%
3	81.9.%	79.8%
4	80.7%	78.4%
5	78.5%	80.4%

Ratio of testset size = 0.2

Time	Accuracy(near dt)	Accuracy(far from dt)
1	85.6%	80.3%
2	83.4%	77.6%
3	81.2%	79.8%
4	80.9%	78.3%
5	84.4%	79.2%

Ratio of testset size = 0.1

Time	Accuracy(near dt)	Accuracy(far from dt)
1	81.6%	77.4%
2	78,4%	80.5%
3	80.2%	75.4%
4	81.4%	78.5%
5	76.8%	81.2%

The average accuracy of the 3 ratios of testset

Ratio	Mean Accuracy(near dt)	Mean Accuracy(far from dt)
0.5	80.6%	78.4%
0.2	83,4%	80.5%
0.1	80.2%	75.4%

4.3 The range of accuracy

The models randomly split the dataset to training set and testset, so the accuracy is not constant, and has a range of changing, which is discussed below.

Degree = 2
Ratio of testset size = 0.2

Time	Accuracy	Accuracy
1	83.6%	80.4%
2	81,4%	82.5%
3	84.2%	75.4%
4	85.4%	83.5%
5	84.8%	81.2%

So from the table above, the accuracy for Airbnbs near dt are from around 81% to 85%, while Airbnbs far from dt is around 75% to 83%.

5 Conclusion

After analyzing the graphs and tables generated, the linear correlation among variables is observed using Pearson Correlation. This correlation reveals the linear relationship between the numeric feature and our target value. What's more, the observation from Pearson Heatmap shows that the numeric values and the diverging green colours encode the close linearly relationship between the feature and the price. This observation suggests a further feature engineering.

The graphs demonstrate a linear relationship between the feature and the price. With this relationship, the feature is highly related to the price when a significant linear relation is observed. For example, when the neighbourhood is close to the Manhattan area, the price is observed to be higher. On the other hand, some features only has partially linear relationship with the target value.

In terms of accuracy, the right-fit degree has a positive impact on the accuracy in terms of polynomial regression. Therefore, the test of the right-fit degree appears to be the most significant part.

Lastly, the testset size also has an impact on the accuracy of the results. However, since the training sets and the test sets are randomly generated, the accuracy never stays constant. A range of accuracy could be found by conducting more experiments.

6 References

[1] Kaggle.com. (2019). New York City Airbnb Open Data. [online] Available at: https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data [Accessed 4 Dec. 2019].

[2] Blog.datadive.net. (2019). Feature selection – Part I: univariate selection | Diving into data. [online] Available at: https://blog.datadive.net/selecting-good-features-part-i-univariate-selection/ [Accessed 4 Dec. 2019].

[3] En.wikipedia.org. (2019). Pearson correlation coefficient. [online] Available at: https://en.wikipedia.org/wiki/Pearson_correlation_coefficient [Accessed 4 Dec. 2019].

[4] Statistics Solutions. (2019). Pearson's Correlation Coefficient - Statistics Solutions. [online] Available at: https://www.statisticssolutions.com/pearsons-correlation-coefficient/[Accessed 4 Dec. 2019].

[5] Cs.cmu.edu. (2019). Cross Validation. [online] Available at: https://www.cs.cmu.edu/~schneide/tut5/node42.html[Accessed 4 Dec. 2019].

[6] En.wikipedia.org. (2019). Pareto principle. [online] Available at: https://en.wikipedia.org/wiki/Pareto_principle[Accessed 4 Dec. 2019].

[7] Anon, (2019). [online] Available at: https://stackoverflow.com/questions/31771627/splitting-training-and-test-data] [https://en.wikipedia.org/wiki/Pareto_principle[Accessed 4 Dec. 2019].

[8] Saedsayad.com. (2019). Model Evaluation. [online] Available at: https://www.saedsayad.com/model_evaluation_r.htm[Accessed 4 Dec. 2019].