Creating a Machine Learning AdsBot

Daniel Rondon

August 2020

1 Introduction

This project aims at obtaining the IBM Data Science Certification. For this we need to solve a case based on real life data, using Python and Jupyter Notebook. Along with this report, a presentation and a LinkedIn article were also produced.

1.1 Background

One way to monetize a web or mobile application is to find direct sponsors or ads. There are for instance a number of agencies dedicated to finding sponsors and publicity space into popular applications. Many AdsBots are also found in the market. Creating an in-house AdsBot allows avoiding to pay classical Ads services and commercial AdsBots and having as well a fit-to-purpose high-customized bot.
For instance, Airbnb could have the need to create this kind of bot to monetize its website. Showing Venues Ads relevant to the kind of rental the customers were looking for.

1.2 Problem

Collaborative filtering has for instance been one of the usual methods to create this kind of bot. However, we can show here how to reduce this problem into a simple classification problem. This project aims at creating a cluster classification based on features of Airbnb rentals.
That could be compared with the common venues around a given cluster. Venue data compilation would be done around each rental employed to build the predictive model.
Once the model is built, it would be possible to pass through the different features of the rental. It would be therefore able to assign rentals to a cluster class and it would give the type of venue for advertising, in terms of the predicted cluster.
After that it would be then possible to get the venue details in order to: 1) enable the stakeholder to sell Ads space to small businesses for instance and 2) enable the bot to pick a venue according to the venue type related to the predicted cluster. This last point is out of the scope of this project.

1.3 Interest

The main interest to have an AdsBot integrated to a mobile or web application is to boost monetization. Actually, when advertisers invest in Ads on a particular application they expect an effective increase in revenues and an AdsBot allows targeting the right audience with the right product.
In our case we built a very simple AdsBot to target visitors of Airbnb web pages with local businesses or venues around a given rental.

2 Data acquisition and cleaning

This was a fundamental stage of the project in order to set the base for success.

2.1 Data sources

We had two main data sources: Airbnb data and venues data from Foursquare API.

2.1.1 Getting Airbnb data

Airbnb's data in Manhattan from 2019 was used to create our AdsBot. The data has been originally scrapped in 2019 and published on Kaggle website. The data can be found on this site:
https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data.
Or it could be directly downloaded from the IBM storage dedicated to this project:
https://cloud-object-storage-tz-cos-standard-oa8.s3.eu-de.cloud-object-storage.appdomain.cloud/AB_NYC_2019.csv
Dataset has been already clean and ready to use. It describes the listing activity and metrics in NYC, NY for 2019. This data file included the needed information to find out more about hosts, geographical availability, necessary metrics to make predictions.

2.1.2 Getting Foursquare data

Venue data was scraped from Foursquare API sending geographical information, Latitude and Longitude, by Neighborhood. To perform this task we first converted addresses information into geographical information. This worked sending the neighborhood names through a geocoder API and then receiving the coordinates.
The final results of executing this are shown in the following table (see Table 1):

	Neighborhood	Latitude	Longitude
0	Midtown	40,76	-73,98
1	Harlem	40,81	-73,95
2	East Harlem	40,79	-73,94
3	Murray Hill	40,76	-73,81
4	Hell's Kitchen	40,76	-73,99

Table 1. Neighborhood coordinates collected from Geocoder API.

Through Foursquare API, in addition to coordinates, it was also sent a radius of investigation equal to 500 m around the coordinates with a limit of 100 Venues per API result. The free version of Foursquare API in any case is limited to 100 Venues per API call.
A dataset with the venue contact information was generated as a subproduct. For instance, it could be utilized to contact the small business and other venues owners in order to advertise his/her business if it matches with his/her type of business.

2.2 Features selection

In this section, the features selected to build the prediction model are discussed.

2.2.1 Position

Basically the most important features were those related to the geographical location. This could be translated into the Latitudes and Longitudes of each rental.

2.2.2 Price

The Price is expressed in dollars. It was the main driver of all trends and without surprise it was well correlated with geographical location of the rentals. For more details please refer to the section: Data Preprocessing and Cleaning.

2.2.3 Popularity

Along with this report, this feature is called “Popularity”. However, as it can be observed this feature was in reality related to the reviews received by the hosts. Popularity should be more related to “stars” given to the host of the rental nevertheless we did not have this data at the moment.
It can also be expected that a rental receiving reviews, good or bad, are visited enough to be included into the investigation. However, one knows that this is not entirely true. There is a fact that people do not give reviews to Airbnb rentals deserving a negative one. So people give no feedback instead of sending a negative review.

2.2.4 Dwelling

This feature could be helpful to explore the data. However, it was not included into Machine Learning algorithms. This feature depends on prices therefore it might overweight the clustering classification with the Price feature. Also, the selected algorithm that we are using for clustering classification accepts not discrete values, only continuous values.

For practical reason, Dwelling string names were replaced for integer discrete values as follows:

- Entire home/apt = 1;
- Private room = 2;
- Shared room = 3.

2.2.5 Features left out

All others not used columns were left out. Those features giving little information about the classification of the rentals were dropped out. Those dropped fields were: 'id', 'name', 'host_id', 'host_name', 'last_review', 'availability_365', ‘calculated_host_listings_count', 'reviews_per_month', 'minimum_nights'.
When looking for improvements in the Machine Learning classification stage, this list could be helpful in future works.

2.3 Data preprocessing and cleaning

Predictive models rely on clean data. In the next section the process of cleaning the data is described.

2.3.1 Basic cleaning

As mentioned before, conventional data cleaning of data was performed, for instance changing column names, keeping only rentals located in Manhattan, and filtering the data, etc. In the sections below, data cleaning related to Data Analysis is discussed.

2.3.2 Before cleaning

As it can be observed in Figure 1, a first Data Analysis was performed on the initial Data Distribution of Price and Popularity Fields. Distributions of Price and Popularity as histogram were plotted in Figure 1.a and Figure 1.b, and as Box plot in Figure 1.c and 1.d.

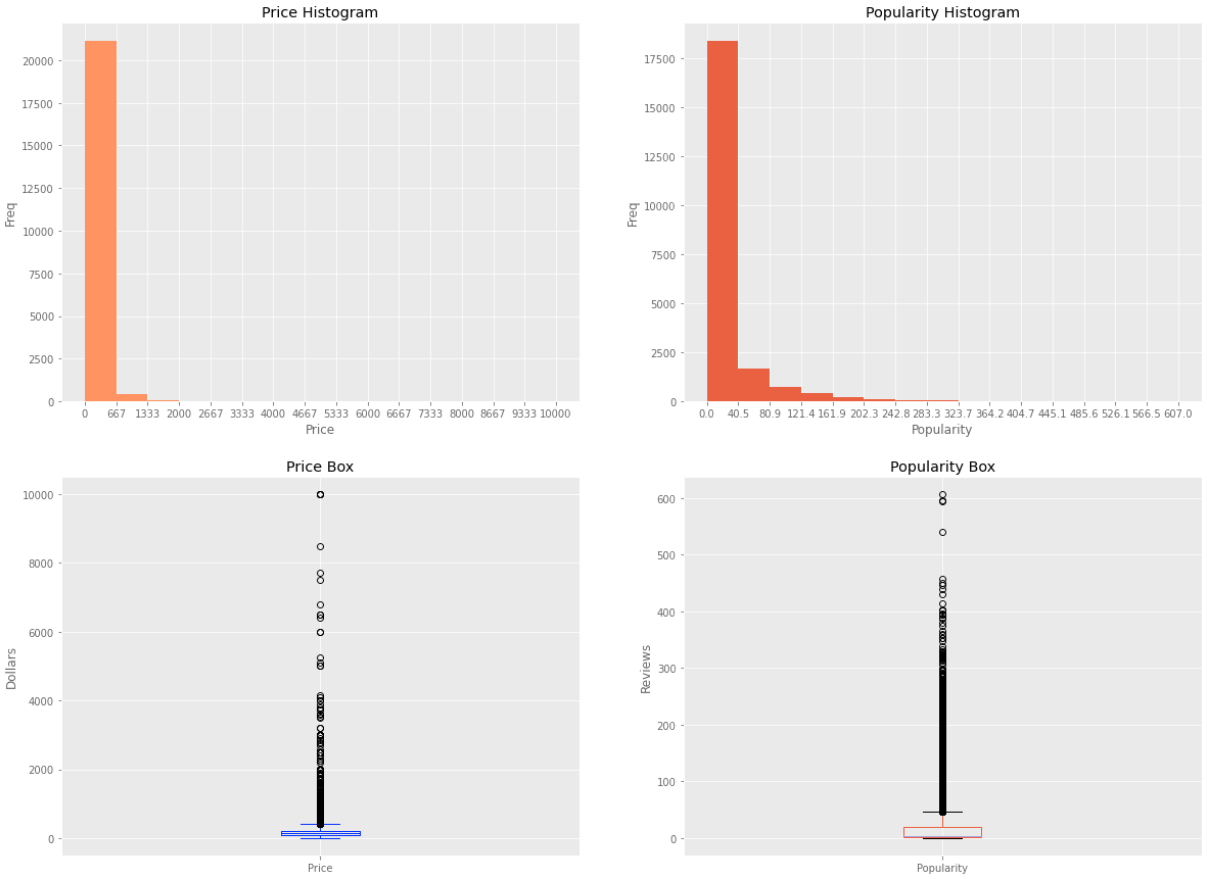

Figure 1. Distributions before cleaning the data: a) Prices Histogram, b) Popularity Histogram, c) Box plot distribution of Prices and d) Box plot distribution of Popularity.

We can take a closer look at the statistics of these two features, and placing them into the following table:

	Price	Popularity
count	21661.00	21661.00
mean	196.87	20.98
std	291.38	42.57
min	0.00	0.00
25 %	95.00	1.00
50 %	150.00	4.00
75 %	220.00	19.00
max	10000.00	607.00

Table 2. Statistics of the feature Price and Popularity before data cleaning

As it can be observed above in Figure 1.a and 1.b, there are few host rentals with prices that are abnormally high. It was not recommended cleaning the values all over

220 dollars (see Table 2) to eliminate abnormal values of Price between 220 and 10000 dollars. This is because the maximum values can vary by Neighborhoods.

Price and Popularity data were therefore organized by Neighborhood in order to understand where to exactly find those outsiders values. As confirmed in Figure 2a. and Figure 2.b, this is because the rage of Prices was different depending on the Neighborhood. This information was useful to clean the data.

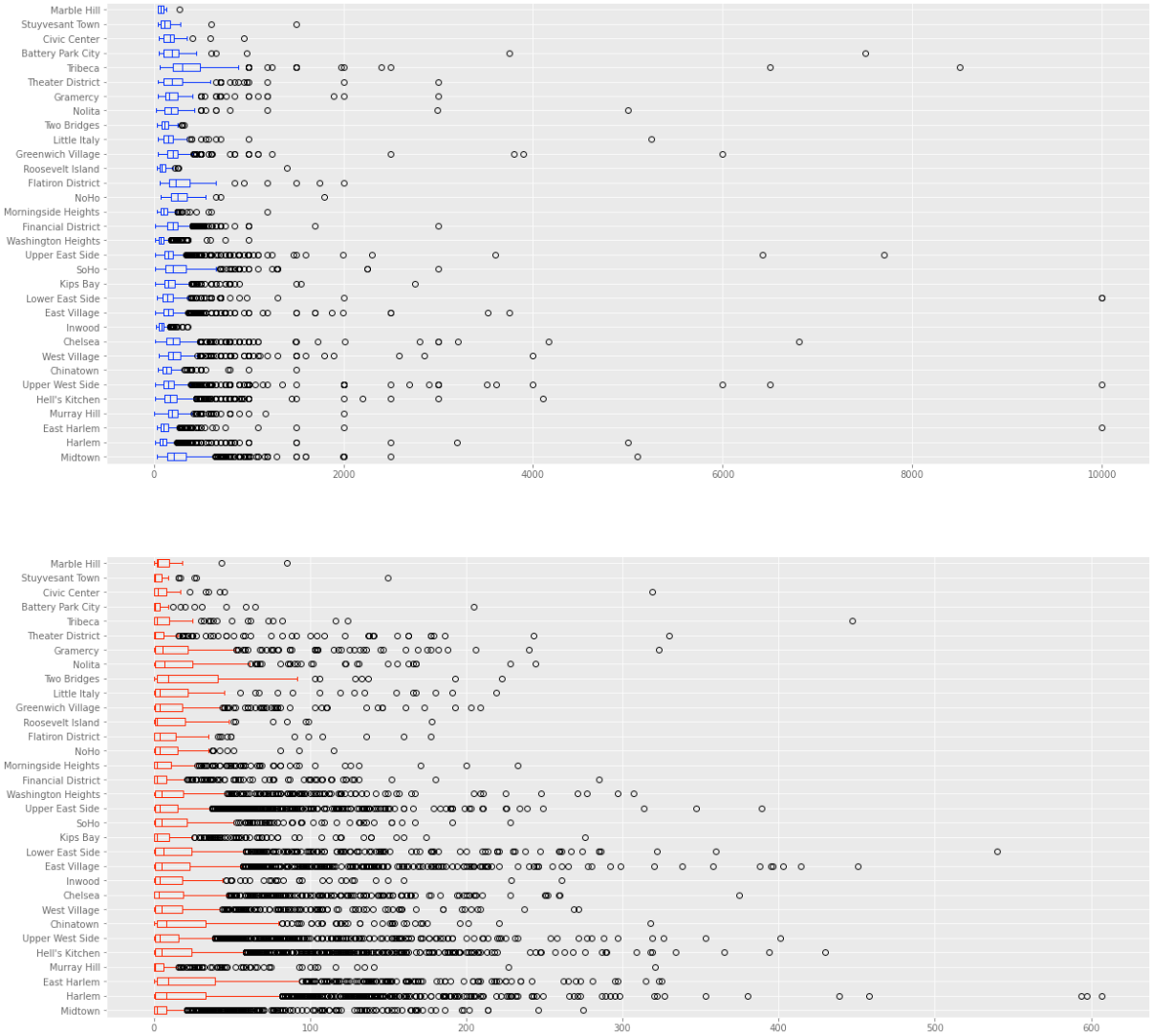

Figure 2. Box plot distribution of data before cleaning organized by Neighborhood: a) Price distribution and b) Popularity distribution.

In Figure 2.a it could be inferred that there are few outsiders above Quantile .75, and the average of Price could also be grouped. This was good news for classification because the data was not extremely sparse. However, in Figure 2.b we can observe an extensive amount of outsiders values and Reviews equal or close to zero. It means that Popularity would not be very easy to classify because of the sparse data.

2.3.3 Cleaning rentals with zero reviews

In order to clear the distribution of the reviews, the rentals with zero reviews were excluded from the study. As mentioned before, it was assumed that rentals with no review were most likely, or in most of the case, rentals with negative impressions. We did not want to include them into the bot predictions.

Dwelling	Neighborhood	Latitude	Longitude	Price	Popularity
1	Battery Park City	40,70	-74,02	10,00	1,00
2	Battery Park City	40,70	-74,02	10,00	1,00
3	Battery Park City	40,70	-74,02	10,00	1,00

Table 3. Minimum values of features by kind of Dwelling.

As it can be observed above in Table 3, rentals with 0 reviews were left out.

2.3.4 Cleaning Prices Out of Range

In order to clean the prices out of range, or outsiders values, the procedure consisted of filtering the Airbnb data organized by Neighborhood as shown in Figure 2. Therefore, the values above the Quantile .75 value were phased out for one each of the Neighborhood.

Dwelling	Neighborhood	Latitude	Longitude	Price	Popularity
1	West Village	40,87	-73,92	320,00	229,00
2	West Village	40,88	-73,91	330,00	607,00
3	West Village	40,88	-73,91	485,00	403,00

Table 4. Maximum value of the features by dwelling.

As observed above (see Table 4), the max prices were set in an adequate range of values for the three kinds of Dwelling.

2.3.5 After cleaning

After the cleaning process the final count of rentals was reduced to 12 959. Only these will be used for classification.

As shown in Figures 3.a and 3.c, the distribution of Price was globally improved.

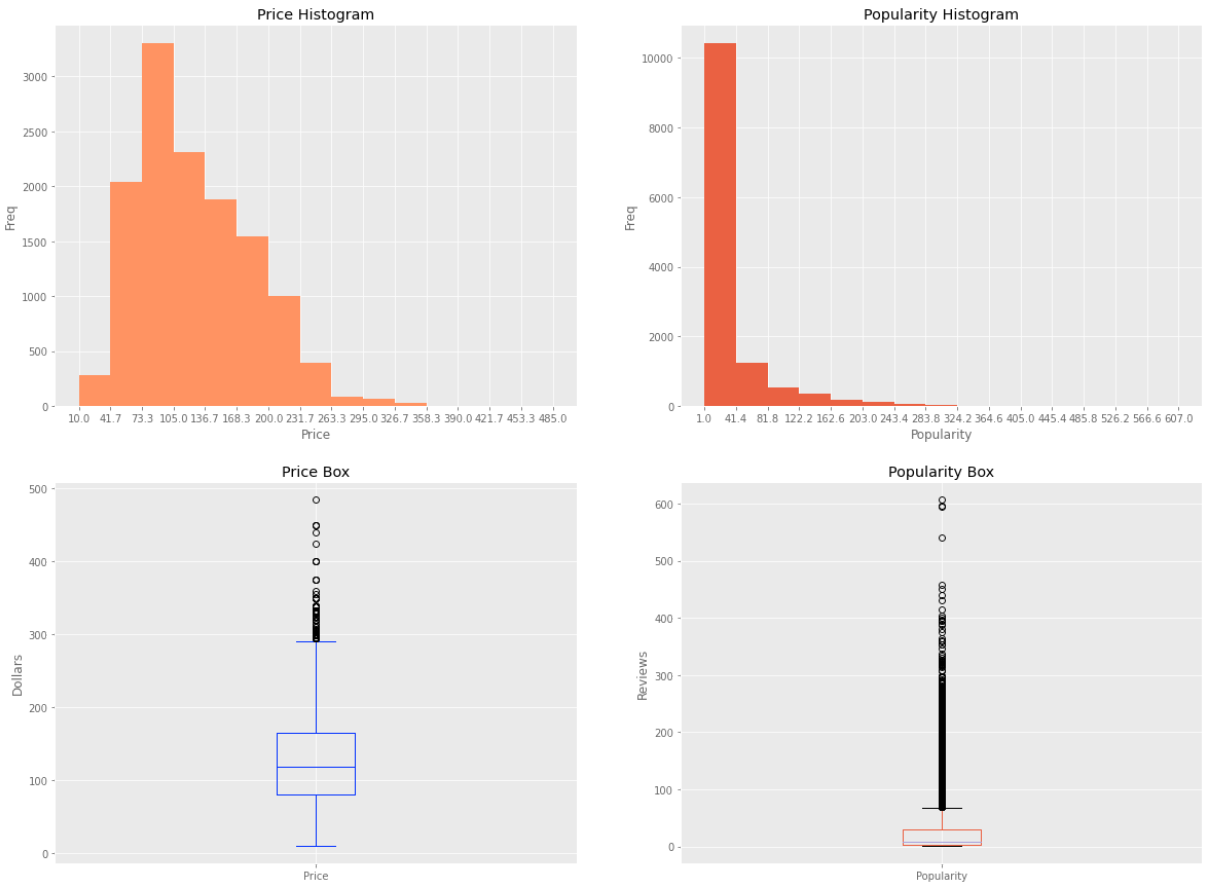

Figure 3. Distributions after cleaning the data: a) Prices Histogram, b) Popularity Histogram, c) Box plot distribution of Prices and d) Box plot distribution of Popularity.

Otherwise, in terms of Popularity, a lot of outside values were found in the range from 100 to 600 reviews. However, most rentals have between 1 and 40 reviews: the median should be somewhere between 15 and 20 reviews per rental.
For visualization purposes data was organized by neighborhood.

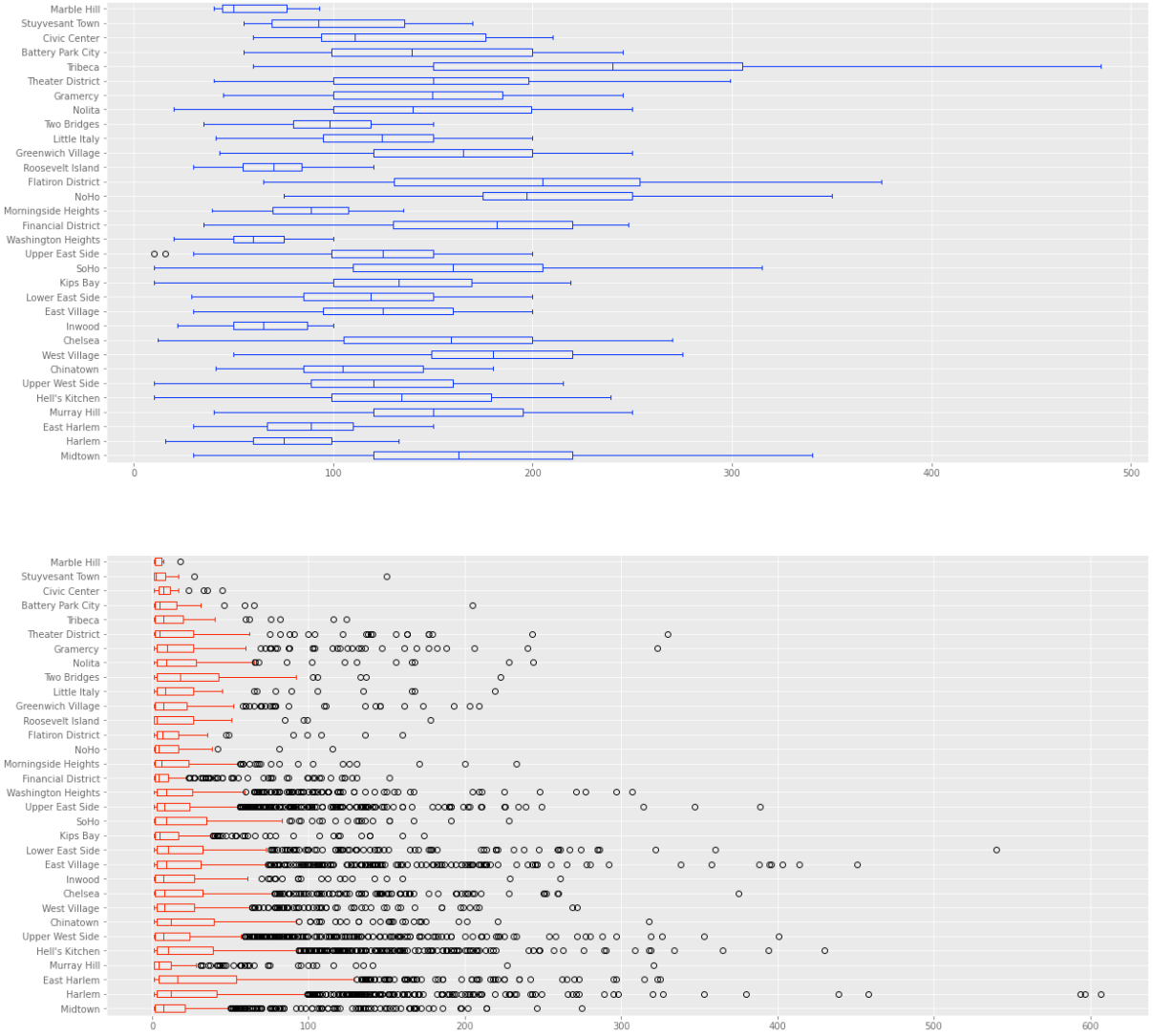

Figure 4. Box plot distribution before data cleaning organized by Neighborhood: a) Price distribution and b) Popularity distribution.

The distribution of prices in Figure 4 were significantly improved compared to the Prices before cleaning. However, for reviews distribution was not the case because there was a significant number of rentals with only 1 review. Taking a closer look at the Bar plot of Popularity, Reviews of most of the Neighborhoods are around 15 (see medians on the Box plot in Figure 4).

We can also take a closer look at the statistics of these two features, and placing them into the following table:

Stats.	Price	Popularity
count	12959,00	12959,00
mean	126,49	28,14
std	57,14	48,66
min	10,00	1,00
25 %	80,00	3,00
50 %	119,00	8,00
75 %	165,00	29,00
max	485,00	607,00

Table 5. Statistics of the feature Price and Popularity after data cleaning

As it can be observed in Table 5, aberrant Prices and rentals with zero reviews were eventually excluded.

3 Exploratory Data Analysis

In this section, the data was analyzed more in depth after to perform the cleaning.

3.1 Airbnb Data

The exploratory data analysis is basically based on maps generated using the Folium. This is a Python library to visualize spatial data.

The Price Location Correlation follows as expected. For example, prices decrease from South to North and increase in the Eastern side of Lower Manhattan.

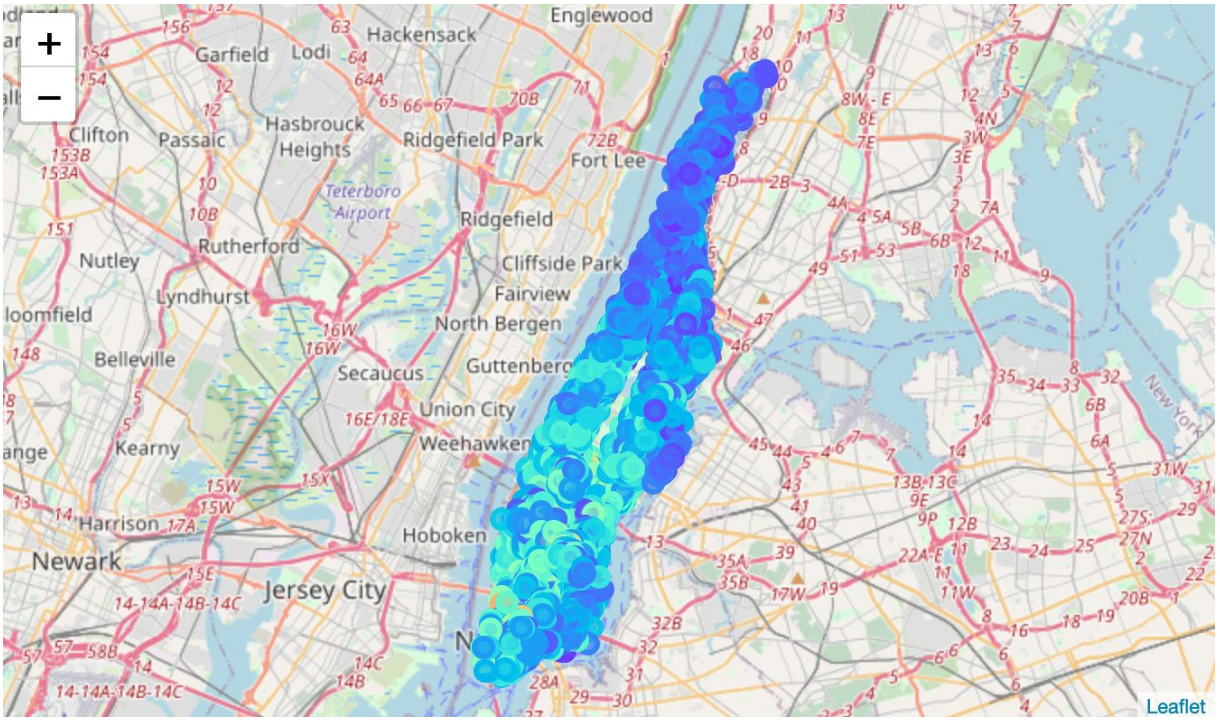

Figure 5. Distribution of rentals with prices as color scale in Manhattan.

There isn't an obvious trend. However, we can observe some clusters of average reviews (highlighted on purple in Figure 6) and around those in their border we have rentals with a higher number of reviews (green points in Figure 6) and by far! (red points). Nevertheless, there is not an easily observable correlation between Popularity and Price. In the paragraphs below we take a closer look at this.

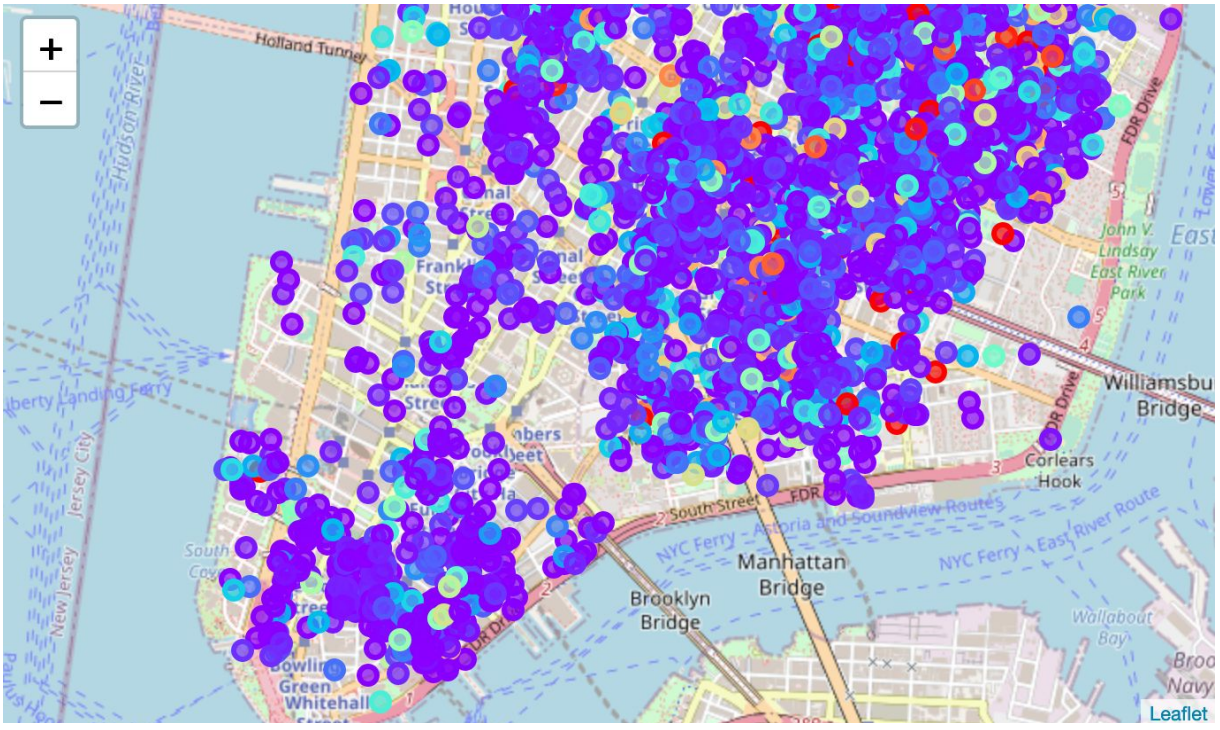

Figure 6. Distribution of rentals with prices as color scale in Lower Manhattan.

Then, looking at the Pearson correlations between features, it could be concluded that, as expected, Price had a good correlation with location (Latitude and Longitude) and dwelling. However, Popularity once more seemed to have a very poor correlation compared to the others features.

	Latitude	Longitude	Dwelling	Price	Popularity
Latitude	1,00	0,86	-0,24	-0,47	0,03
Longitude	0,86	1,00	-0,23	-0,50	0,03
Dwelling	-0,24	-0,23	1,00	0,60	-0,10
Price	-0,47	-0,50	0,60	1,00	-0,06
Popularity	0,03	0,03	-0,10	-0,06	1,00

Table 6. Correlation table between features after data cleaning.

Fortunately, when looking at the cross-plots in Figure 7, we could observe that there were some trends showing from Popularity data. That meant that we could find some tendencies helping the clustering classification to characterize the rentals.

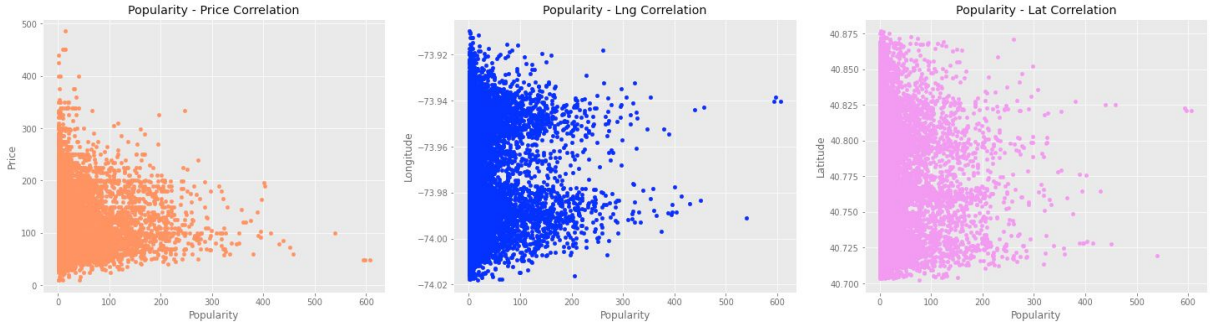

Figure 7. Cross-plots of a) Prices, b) Latitude and c) Longitude, depending on Popularity.

Analyzing Figure 7.a, reviews per rental increase when prices approach the median of all dwelling together. However, when prices are too high or two low, the amount of reviews per rental decreases close to zero (rentals with zero reviews were cleaned before).

Regarding the Figures 7.b and 7.c, some trends are also shown. For instance, reviews tend to increase in-land and decrease toward the coast, the east side has more reviews than the west side. As a funny fact, we can see the cave created by Central Park in the middle. Eventually, the number of reviews softly decreases toward Upper Manhattan (see Figure 7.c).

3.2 Venues Data

After sending the neighborhoods coordinates through the Foursquare API, 2597 venues were returned and 293 unique categories of venues were collected.

Checking the number of Venues returned we could see that the limit of venues per API call was reached in 17 neighborhoods from 32 in total. It can be therefore inferred that, in those neighborhoods, we probably have a sub-representation of the real number of Venues. This can be resolved using the paid version of the API or with a workaround discussed in the Way Forward section (see below).

4 Clustering Classification

In order to understand how rentals were distributed, a clustering classification was performed.

4.1 Normalizing the Data

A number of four features were utilized to perform the cluster classification: Latitude, Longitude, Popularity and Price. These features were first standardized using a method available within the Python library to perform the K-mean clustering classification (Scikit-Learn Library).

4.2 Modeling

Since beforehand it was unknown how rentals were defined, an unsupervised algorithm was opted to. Furthermore, the challenge was to build it as simple as possible. In this project a simple unsupervised algorithm was applied: the K-means algorithm.

When using K-means clustering classification, there is a parameter that should be previously considered: the number of clusters. This parameter was defined at four (4) clusters. Therefore, four (4) clusters were investigated to classify the Airbnb rentals.

In the section below the clusters results were analyzed. After analyzing them, it could be noticed that four (4) clusters fitted well to our case. When it would not be the case, the number of clusters could be easily modified, either increased or decreased.

4.3 Clusters Analysis

In this section, each cluster was analyzed by comparing their statistics. The objective was to understand each one of those clusters.

The four features were organized by cluster (see Table 7). Thanks to the statistics included in the table it could be possible to analyze how the clusters were organized by the algorithm.

Clusters	Latitude		Longitude		Popularity			Price		
Stats.	Min	Max	Min	Max	Mean	Min	Max	Mean	Min	Max
0	40,**0**70	40,80	-74,02	-73,94	16,26	1,00	197,00	214,82	177,00	485,00
1	40,70	40,88	-74,02	-73,91	17,49	1,00	102,00	76,42	10,00	110,00
2	40,70	40,87	-74,02	-73,92	162,65	88,00	607,00	115,47	30,00	333,00
3	40,70	40,83	-74,02	-73,93	14,73	1,00	94,00	140,07	108,00	177,00

Table 7. Statistical information of features organized by cluster classification.

In summary, and based on Table 7, the rentals were arranged as follows:

- Cluster_0: Expensive Dwellings with average Popularity (Red).
- Cluster_1: Cheap Dwellings with average Popularity (Orange).
- Cluster_2: Rentals with a high level of reviews (Light green).
- Cluster_3: Middle price Dwellings with average Popularity (Blue).

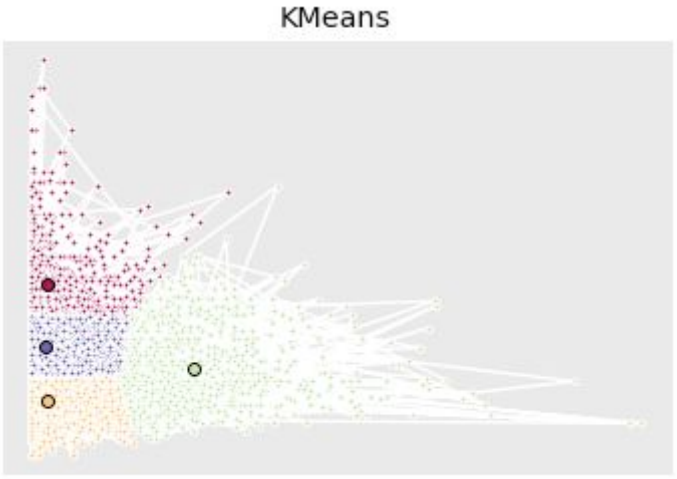

Figure 8. Cross-plot of Price vs Popularity of the rentals colored by the cluster classification.

As shown above in Figure 8, the clusters were heavily assigned by the price except for Cluster_2 (in light green). This cluster basically included the most popular rentals.

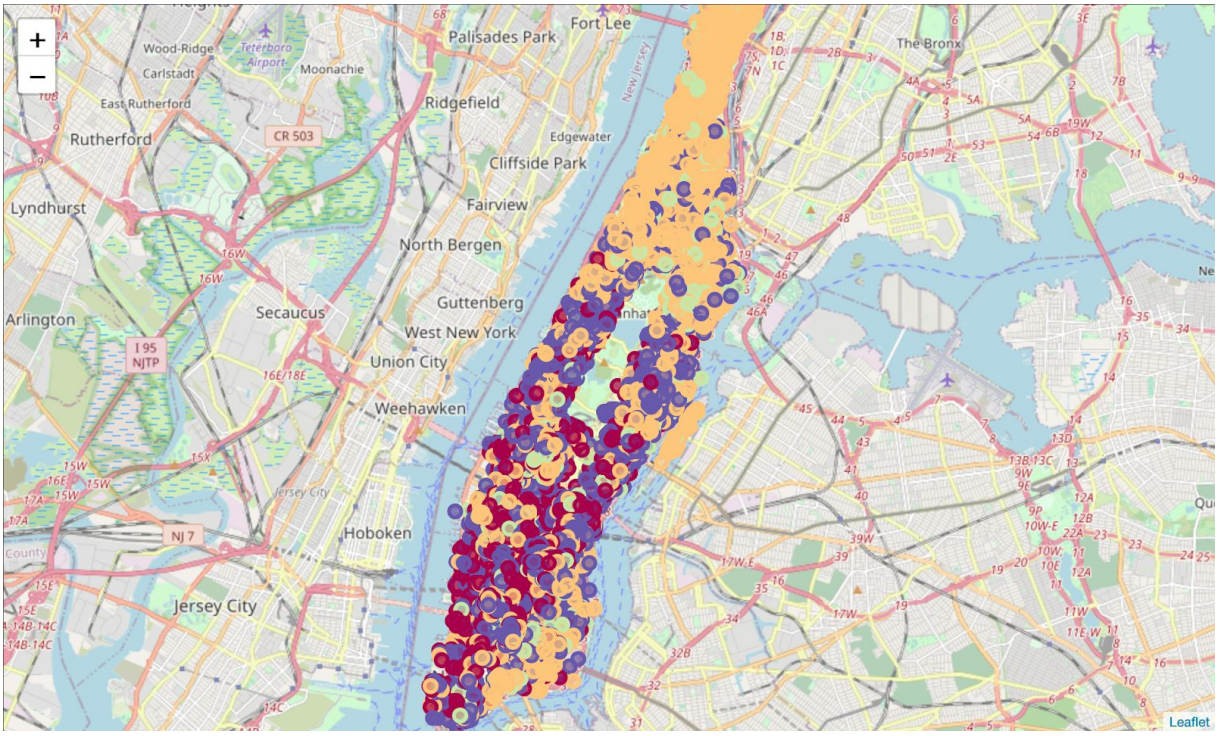

Figure 9. Distribution of rentals with Cluster classification as color scale in Manhattan.

Concerning the cluster distributions on a map (see Figure 9), those were mainly aligned with the Price distribution. We can see that Cluster_1 (in orange) is more widespread in Upper Manhattan, East Village, Lower East Side and Roosevelt Island.

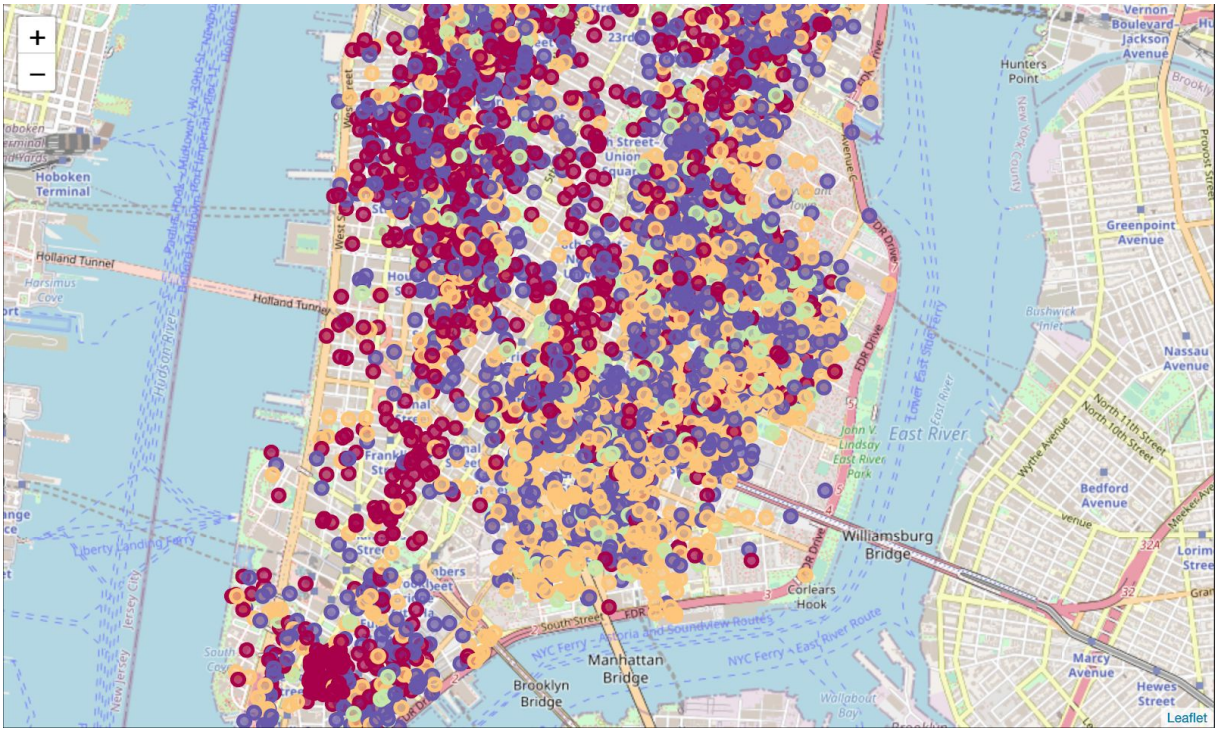

Figure 10. Distribution of rentals with Cluster classification as color scale in Lower Manhattan.

Zooming the map in Lower Manhattan (see Figure 10), it was found that the Cluster_0 (Red) and Cluster_1 (Orange) gathered in high density areas and Cluster_2 (light green) was always found in the surroundings of these areas. That confirmed what was seen in Figure 6. That said, Airbnb customers wrote more reviews for the rentals away from the areas of high concentration of both cheap and expensive rentals.

	Neighborhood	Cluster_0	Cluster_1	Cluster_2	Cluster_3
1	Battery Park City	37 %	33 %	3 %	27 %
2	Chelsea	38 %	22 %	10 %	31 %
3	Chinatown	3 %	47 %	11 %	40 %
4	Civic Center	25 %	50 %	0 %	25 %
5	East Harlem	0 %	62 %	14 %	24 %

Table 8. A sample of Neighborhoods showing the proportion of the clusters.

Taking a look at the percentage distribution of the clusters by Neighborhood (see Table 8), it could be noticed that the Neighborhoods have a significant influence on the cluster's incidence. For instance, in East Harlem it could not be found Cluster_0 rentals (expensive rentals) however it also has 14% of rentals with large amounts of reviews. Battery Park has the same proportion of expensive, middle and cheap rentals however there are only a few rentals with high levels of reviews. cheap rentals however there are only a few rentals with high levels of reviews. All this relevant information could give an idea how Ads will be distributed among the Neighborhoods.

5 Predictive Modeling

After building our clustering classification model, the next step is to perform predictions with it.

5.1 Preprocessing

Information about the Clusters was first injected into the data of Venues collected from Foursquare API. It was done by looking for venues around each rental and then tagging the venues with the cluster assigned to the investigated rental. The radius of investigation was 50 m.

Actually, generating this dataset was highly demanding on CPU resources and the coding may be improved.

5.2 Group and Rank Venues

The dataset generated in the previous section was eventually organized by Cluster. This was in order to get the frequency of Venues Category by Cluster and creating a ranking table (see Table 9).

Cluster	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
0	Italian Restaurant	Pizza Place	Coffee Shop	Cocktail Bar	Sandwich Place	Ice Cream Shop	American Restaurant	Thai Restaurant	Mediterranean Restaurant	Café
1	Italian Restaurant	Coffee Shop	Bakery	Ice Cream Shop	Pizza Place	Sandwich Place	Cocktail Bar	Chinese Restaurant	Café	Mexican Restaurant
2	Pizza Place	Coffee Shop	Italian Restaurant	Cocktail Bar	Bakery	Sandwich Place	Wine Shop	Ice Cream Shop	American Restaurant	Wine Bar
3	Italian Restaurant	Pizza Place	Coffee Shop	Sandwich Place	Ice Cream Shop	Bakery	Chinese Restaurant	Cocktail Bar	Café	Wine Bar

Table 9. Most common venues by Cluster Classification (from 1st to 10th most common venue).

In this way the top 10 venues for each neighborhood are organized in the most common venues. For prediction purposes, the 8th most common venues were selected.

On one hand, the most common venue was selected from the tail because we assumed that they needed more Ads to reach their new customers. In this way, Ads will give them a real added value, effectively increasing their revenues.

On the other hand, it could be selected as the 10th most common venue or even the 30th most common value. The final choice on this point could be made by stakeholders for instance.

5.3 Testing Predictions

The Dataset to test the model was collected directly from Airbnb website in 2020 (see Table 10).

Address	Dwelling	Neighborhood	Price	Popularity
W 49th st & 11th Ave	Entire Apartment	Hell's Kitchen	54,00	176,00
W 51st st & 9 Ave	Private Room	Hell's Kitchen	36,00	7,00
90 Bleecker st	Shared Room	Noho	170,00	11,00
W 48th st & 8 Ave	Entire Apartment	Theater	422,00	47,00
209 W 147th st	Private Room	Harlem	33,00	12,00

Table 10. List of rentals scraped from Airbnb website (August 2020).

After gathering the location coordinates for each rental (see Table 11), this small dataset of five (5) rentals were sent through the prediction model.

	Latitude	Longitude	Price	Popularity
Test_1	40,77	-74,00	54,00	176,00
Test_2	40,76	-73,98	36,00	7,00
Test_3	40,69	-73,92	170,00	11,00
Test_4	40,76	-73,98	422,00	47,00
Test_5	40,82	-73,94	33,00	12,00

Table 11. Final format to be used as an input test into the prediction model.

Once the clusters were assigned to the collected rentals, those were compared with the 8th most common venues (see table 8). Eventually the bot performed a series of suggestions on the venues (see Figure 11).

```
Rental is categorized in Cluster # 2
Advertising business category: Ice Cream Shop is suggested
Rental is categorized in Cluster # 1
Advertising business category: Chinese Restaurant is suggested
Rental is categorized in Cluster # 3
Advertising business category: Cocktail Bar is suggested
Rental is categorized in Cluster # 0
Advertising business category: Thai Restaurant is suggested
Rental is categorized in Cluster # 1
Advertising business category: Chinese Restaurant is suggested
```

Figure 11. Output from the prediction model: suggested venues according to the Cluster Classification of the rental.

The bot outcomes cloud be explained as follows:

- Test_1: This Hell's Kitchen apartment was assigned to Cluster_2. It means that it is a popular rental with an average price. Therefore, an Ice Cream Shop must be suggested for advertising.

- Test_2: This other Hell's Kitchen private room was considered as Cluster_1. That means a cheap rental with average number of reviews and concluding that Chinese Restaurant should be advertised.
- Test_3: The Noho shared apartment was set within Cluster_3 middle-price rentals and a Cocktail Bar was therefore suggested for advertising.
- Test_4: For this apartment located in the Theater district in Midtown. This rental was thus assigned to Cluster_0, suggesting Thai Restaurant for advertising.
- Test_5: This private room in Harlem was also assigned to Cluster_1 and Chinese Restaurant was suggested for advertising.

6 Conclusions

Although all improvements that could be done in the future, it was possible to build a simple AdsBot from Airbnb and Foursquare data using the K-means Machine Learning Algorithm. Our AdsBot was capable of suggesting a least common category of venue (usually small businesses) located close to characterized rentals.

Those rentals were characterized by Machine Learning based on Location, Price and Popularity. Therefore, they were indirectly related to the visitors profile and type of hunted rental. Thus, whether the visitor's behavior changes or data becomes larger, the bot will be able to adapt the suggestions without significant changes into the code.

This capability to adapt to the data represents an added value provided by Machine Learning. For instance, performing the same classification by hand would imply to change min and max ranges per cluster each time that data evolves.

7 Way forward

For future work there are many improvements that can be done to this project. A few improvements could be mentioned:

7.1 Optimizing the code:

There are two or three blocks of code that repeat a couple of times. These code blocks could be converted into functions. It would help the code to be more readable and specially more efficient.

7.2 Increasing number of Venues

The numbers of venues are limited to 100 in the free version of Foursquare. In the case that it would be needed to have more than 100 per neighborhood, the best way to have this limit increase is to get the paid version.

However, it is also possible to apply a workaround: create a dummy Neighborhood using a mid-point between each pair of Neighborhoods. This mid-point can receive another 100 venues. The duplicated venues only have to be eventually cleaned. It allows covering a major number of venues though it will never be a guarantee to have them all.

7.3 Popularity feature

There are many ways the Popularity feature could be improved. One of them is to obtain the text data for each review, and not the sole number of reviews. For this project we unfortunately did not have this information.

However, once this information would be collected it would be possible to apply Machine Learning algorithms for sentiment analysis. This will allow us to perform a classification of emotions (positive, negative and neutral) within this text data.

7.4 Using Seaborn Library

In this project we basically utilized Matplolib to draw plots. However, the plots can be improved using the Seaborn library instead of Matplotlib, for example.

7.5 Suggesting venues

Because it was out of the scope of this project, the system given capabilities of suggesting the name and address of the venue was not built. However, it can be easily done by making a call to the Foursquare API and then selecting the venue based on location, venue category and whether they subscribed to Ads or not.