Airbnb Customer Segmentation

Carlos Gustavo Salas Flores (cs582@duke.edu)

1 Introduction

Airbnb has quickly become the preferred choice for travelers around the world because of it's simplicity and flexibility. However, it's not only the customers the ones who are being benefited, but also the hosts. Further, there are times when hosts take advantage of this app and start running illegal businesses like hotels without paying the required taxes. That's why some people have attempted to find out who exactly could be running their businesses in this way.

In this project, I analyze Airbnb data from New York that I retrieved from http://insideairbnb.com/ and analyze the data to segment the customers in different groups according to their characteristics. For this project, I make use of Unsupervised Machine Learning algorithms, namely, K-Means and OPTICS Clustering

Finally, I perform data visualization in order to compare the characteristics of different groups and find if the availability of these listings is greater than the rest and if their prices are more accessible than the median.

2 Overview of K-Means

The K-Means algorithm is an unsupervised Machine Learning approach to cluster data into different groups.

In a subspace of N-dimensions, a K-Means algorithm will cluster the data in K groups. Each group will have a centroid and every data-point will be assigned to its closest centroid.

The K-Means algorithm consists of 3 main steps:

2.1 Initialization

The centroids will have a starting position, for example, the centroid C_1 would be initially located at position $(x_1^{C_1}, x_2^{C_1}, ..., x_k^{C_1})$.

Such locations can be completely random and they will be adjusted on every iteration.

2.2 Assignment

Each data-point will be assigned to a certain centroid. The distance used in this case is the Euclidean distance.

$$d_{euclidean} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_m - y_m)^2}$$

2.3 Update

Once these groups have been established, the centroids will be updated to be in the weighted middle of their groups.

$$x_j^{C_i} = \sum_{X_i} x_h$$

The algorithm should repeat steps 2.2 and 2.3 until it converges.

3 Overview of OPTICS Clustering

Similar to K-Means, OPTICS is also an unsupervised Machine Learning algorithm. But different from it, OPTICS does not take an M number of clusters/groups, but rather asks for a minimum number of data-points in order to consider each group.

This algorithm is similar to DBSCAN but is slightly more complex.

3.1 Core Point and Core Distance

For a given data-point that has not been mapped before it will be considered a Core Point if there are enough data-points in its radius. In this case I use the default distance used in Scikit Learn which is the minkowski distance.

$$d_{minkowski} = \left(\sum_{i=1}^{k} |x_i - y_i|^p \right)^{1/p}$$

Where p is also set by default to 2 so it ends up being equivalent to the euclidean distance.

For a given Core Point the algorithm checks if it can reach a given data-point from there. If so, those two points will be clustered in the same group. Once it has finished to find all the points that are close to that Core Point, it will check if any of the other points is also a Core Point.

3.2 Min Number of Points

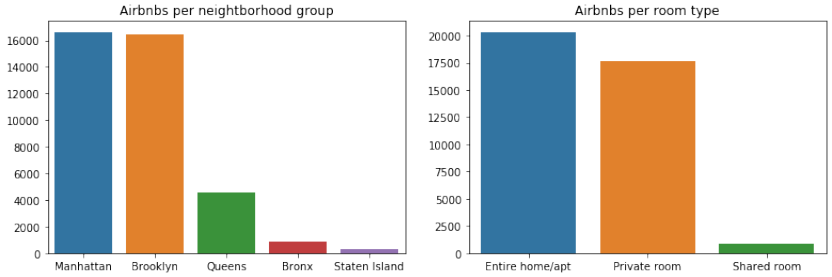
Once it has finished to check all the Core Points in that group. It will check if the number of data-points in that group is larger than or equal to the minimum required, which in this case was 85.

Steps 3.1 and 3.2 are repeated until all the points have been checked.

4 Methodology

The data was retrieved on May of 2019.

It consists of 38843 data-points and contains data from users in Manhattan, Brooklyn, Queens, Bronx, and Staten Island. The dominant regions are Manhattan and Brooklyn, while Staten Island has only a few hundred users.



(a) Users in each region of New York (b) Room Types

Figure 1: Database's Bar Chart

To perform the data analysis, I choose 5 features to perform the data analysis, namely:

- Room Type - T_x
- Price - P_x
- Reviews per month - R_x
- Calculated Host Listings - L_x
- Availability - A_x

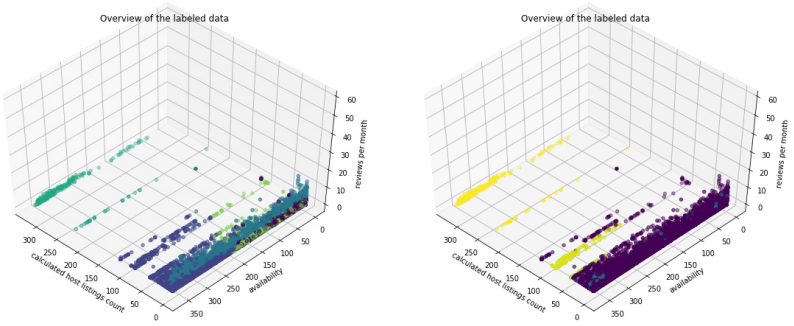
I decided not to use geographical information such as location and neighborhood since those cold produce biased results. In order to find the right K for K-Means, I used the Elbow method, which turned out to give the value $K = 6$.

Since the room type T_x was originally a non-numerical variable, I mapped it to turn it numerical. Afterwards, all these variables were re-scaled using scikit learn's StandardScaler. And then this new matrix A of features is passed to the K-Means and OPTICS algorithm. So finally each data-point is mapped to a group A and a group B.

5 Results

The resulting groups are shown in Tables 1 and 2. Those groups A were produced by K-Means and groups B were produced by OPTICS.

By looking at Figure 1(a), one can observe that some elements could actually belong to the same group, but due to the limitations of K-Means, they were



(a) groups produced by K-Means (b) groups produced by OPTICS

Figure 2: 3D plots of the data-points

clustered in different groups. Figure 1(b) shows the groups given by OPTICS, however most of the elements don't belong to any group, but Figure 2 shows only those groups produced by OPTICS that belong to some group.

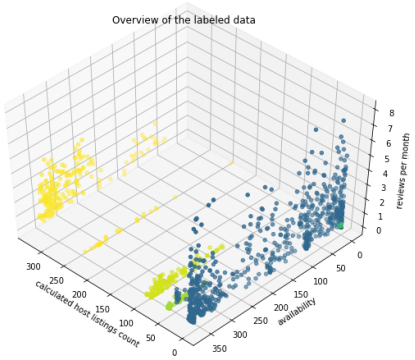


Figure 3: 3D plot of the data-points that map to any of the groups produced by OPTICS

It can be observed that both methods agree on one group, Group A6 and Group B16 are the exact same. Which turns out to be a very suspicious group since their Avg. number of host listings is significantly higher than the rest of the groups.

Table 1: Groups A				
Group ID	Avg. Price	Avg. Reviews per Month	Avg. Host listings	Avg. Availability
1	$127.04	83.47	2.08	145.45
2	$162.10	33.85	7.94	308.21
3	$182.10	16.62	1.68	34.36
4	$271.96	5.57	315.68	286.34
5	$77.39	17.83	2.00	41.42
6	$6439.50	4.66	2.88	185.61

Table 2: Groups B				
Group ID	Avg. Price	Avg. Reviews per Month	Avg. Host listings	Avg. Availability
1	$49.64	0.04	1.00	0.00
2	$39.55	0.03	1.00	0.00
3	$60.09	0.04	1.00	0.00
4	$79.83	0.03	1.00	0.00
5	$99.17	0.04	1.00	0.00
6	$57.16	1.42	4.77	165.87
7	$99.26	0.03	1.00	0.00
8	$122.45	0.04	1.00	0.00
9	$149.88	0.02	1.00	0.00
10	$176.44	0.05	1.00	0.02
11	$199.16	0.08	1.00	0.11
12	$224.97	0.06	1.04	0.05
13	$249.77	0.07	1.00	0.00
14	$141.40	0.11	51.31	343.32
15	$181.95	0.20	92.55	314.19
16	$271.96	1.71	315.68	286.34

5.1 Which Airbnbs are suspicious?

As previously discussed, the most suspicious group is Group A4 or B16 (is the exact same in both tables) and the reason for that is the Avg. number of listings, which is the number of rooms/apartments/etc... that they offer, is significantly higher than the rest of the groups, furthermore, the rooms are available during most of the year. This is very striking because those are characteristics of a Hotel, which suggests that this could potentially be a group of customers running Hotels via Airbnb.

There are other two groups that look highly suspicious, those are Groups B14 and B15, since they have more than 50 listings on average and they are also open during more of the year, almost all of it.

Another suspicious group is Group A2, this is since the Avg. number of host listings is about 8, which is also higher than what is found in the rest of the groups of Table 1.

Finally, another slightly suspicious group can be found in Table 2, Group B6 has an average of 4.77 host listings which is still higher than what can be found in the rest of the groups in Table 2.

5.2 Which Airbnbs are more accessible?

After looking at the different groups generated by the algorithms, I was trying to find if those suspicious users offer more accessible prices since they run their listings as a business.

After an examination, it was clear that Group A4/B16 was not more accessible than the rest. By looking at the groups in Table 1, such group was the second most expensive group of all, and by comparing it to those in Table 2 it could also be observed that such group is one of the most expensive ones.

5.3 Which Airbnbs are open most of the time?

Finally, I attempted to find if the relationship of the different groups with their availability.

By looking at the average number of available days in a year, it can be observed that users in Group A4/B16 are available during most of the year (286.34 days), similarly are Groups B14 and B15 (343.32 and 314.19 days) as well as those in Group A2 (308.21 days) and Group B6 (165.87 days).

In general there's a correlation between the Avg. number of host listings and the Avg. availability since a high/low number of host listings corresponds to a high/low availability.

6 Conclusion

This data analysis of Airbnb users in New York gives insightful information about how these are using the app, and what are some of the characteristics of the groups of users that use this app.

I identified 3 highly suspicious groups in the dataset and 2 others that are slightly suspicious. Nonetheless, there could be overlapping users in some of these groups, so it would be better to just stick with one single Machine Learning method for customer segmentation.

Finally, there are some limitations of this study, one is that the K-Means algorithm does not perform as well in every kind of data, so it should be better to stick with OPTICS or use another alternative like DBSCAN or manually label data and use supervised Machine Learning algorithms. Another one is that this study only applies to NY city, therefore, suspicious groups in other regions could look very different. And lastly, it could be possible to make some

sort of sentiment analysis to analyze the comments and find more insightful information.