IBM Data Science Capstone Project Airbnb Analysis for Beijing Chaoyang District

Shuren Qu

10/18/2020

Introduction

There are more than 11,000 Airbnb rooms listed in data set published by "public.opendatasoft.com". For potential investors who want to start an Airbnb business in Beijing, it would beneficial to conduct an analysis based on location and see if there is any correlation between location features and Airbnb monetization. And for both investors and travelers, they will be benefited from a visualization color coded each Airbnb asset with its segmentation associated with its location features.

Data Source

I acquired Airbnb listing data from "public.opendatasoft.com"， which contains a list of Airbnb assets and its location, price, number of rent information. In addition, I used API provided by "Foursquare" to find common venues around Airbnb assets to portrait its location features.

Prepare Data

There are many fields which may not useful to our analysis, so we will drop them, but remaining: "Coordinates", to find location features through Foursquare API calls; "Room price", as indication ability of monetization; And "Room type" as we want to only use the most common room type for analysis, as room type itself is an independent variable which price may depend on.

	Room ID	Host ID	Neighbourhood	Room type	Room Price	Minimum nights	Number of reviews	Date last review	Number of reviews per month	Rooms rent by the host	Availibility	Updated Date	City	Country	Coordinates
0	23863938	147652234	Chaoyang	Entire home/apt	398	1	6	2/19/2019	0.35	5	364	9/23/2019	Beijing	China	39.8952155567, 116.46591907
1	23914071	19772308	Chaoyang	Entire home/apt	418	1	1	4/1/2018	0.06	1	0	9/23/2019	Beijing	China	39.9577003398, 116.443189661
2	23915836	158663144	Chaoyang	Entire home/apt	397	20	49	6/12/2019	2.74	9	3	9/23/2019	Beijing	China	39.891989506, 116.44585669
3	24186440	94142508	Chaoyang	Entire home/apt	518	1	0	NaN	NaN	7	365	9/23/2019	Beijing	China	39.9265754348, 116.615418345
4	24274046	29488633	Chaoyang	Private room	171	1	15	8/30/2019	0.85	27	358	9/23/2019	Beijing	China	39.9975167474, 116.464205076

3 key columns are chosen, which are price, latitude and longitude. See table below of data description: price range is big enough for analysis

	Room type	Room Price	Coordinates
0	Entire home/apt	398	39.8952155567, 116.46591907
1	Entire home/apt	418	39.9577003398, 116.443189661
2	Entire home/apt	397	39.891989506, 116.44585669
3	Entire home/apt	518	39.9265754348, 116.615418345
4	Private room	171	39.9975167474, 116.464205076
...	...	...	...
11825	Private room	697	39.9859502087, 116.435503117
11826	Entire home/apt	525	39.9001321372, 116.470426807
11827	Entire home/apt	647	39.9324756528, 116.467495116
11828	Private room	199	39.8875319023, 116.466949285
11829	Entire home/apt	801	39.9133870212, 116.47546383

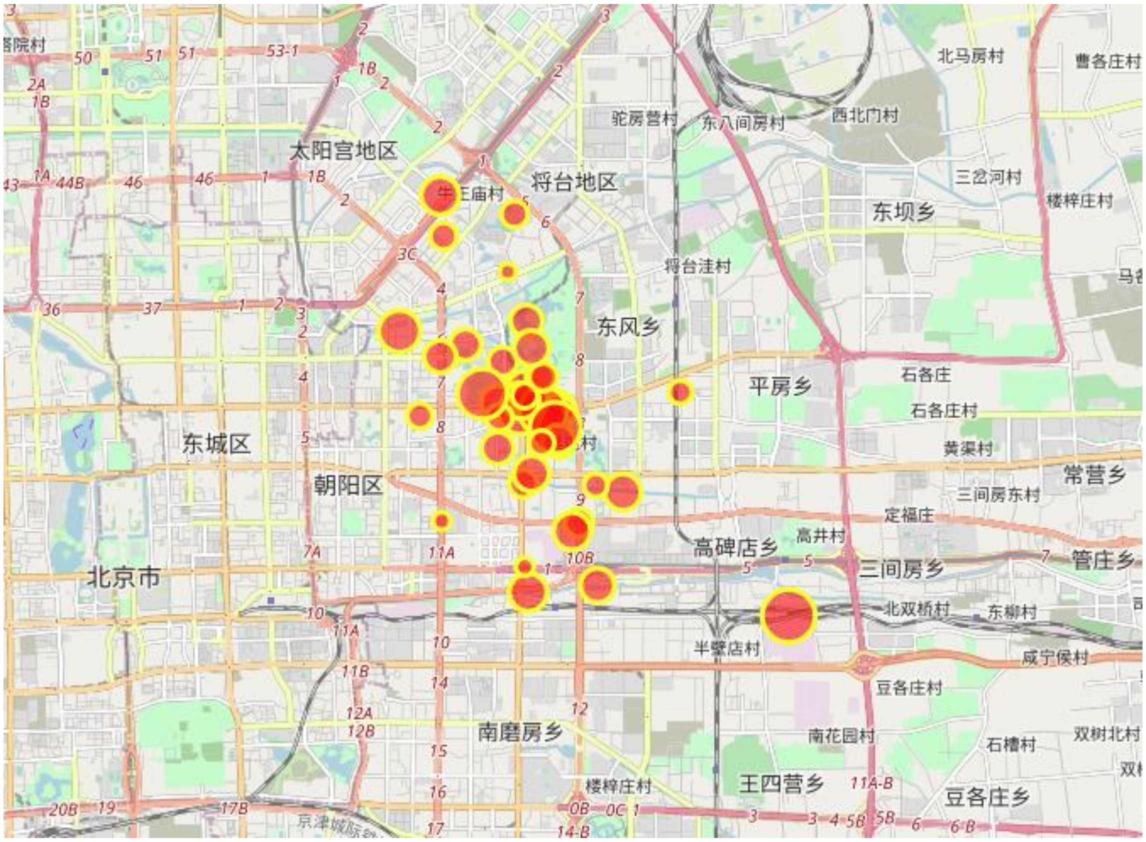
11830 rows × 3 columns

DBSCAN (10,000+ Rooms to 40 Neighborhoods)

There are more than 10,000 records, which is hard to show on map. And for rooms too close with each other, there won't be much difference in terms of location features. So, I did DBSCAN (Density Based Scanning) on coordinates to further group rooms into neighborhoods

- epsilon = 0.003 (0.003 change in latitude and longitude can draw a reasonable size of area on map)
- Minimum Samples = 7

Below is results of 40 neighborhoods on map, created by folium:



Add Location Features and Segment based on features

Next step, I added most common venues using Foursquare API, process the data to easily show top 10 most common venues for each neighborhood:

	Neighborhood	Antique Shop	Asian Restaurant	Athletics & Sports	BBQ Joint	Bagel Shop	Bakery	Bar	Beijing Restaurant	Bookstore	...	Supermarket	Sushi Restaurant	Szechuan Restaurant	Taiwanese Restaurant	Tennis Court	Thai Restaurant	Vietnamese Restaurant	Xinjiang Restaurant	Yoga Studio	Yunnan Restaurant
0	-1.0	0.0	0.00	0.0	0.0	0.100	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.00	0.0	0.0	0.100	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
2	1.0	0.0	0.00	0.0	0.0	0.000	0.1	0.1	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.1	0.0	0.0	0.0
3	2.0	0.0	0.20	0.0	0.0	0.000	0.0	0.0	0.0	0.1	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.1	0.0
4	3.0	0.0	0.00	0.0	0.0	0.125	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
5	4.0	0.0	0.00	0.0	0.0	0.100	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
6	5.0	0.0	0.00	0.0	0.0	0.000	0.1	0.1	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.1	0.0	0.0	0.0
7	6.0	0.0	0.00	0.0	0.0	0.100	0.0	0.0	0.0	0.1	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
8	7.0	0.0	0.00	0.0	0.0	0.100	0.0	0.1	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
9	8.0	0.0	0.00	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
10	9.0	0.0	0.00	0.0	0.0	0.100	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
11	10.0	0.0	0.00	0.0	0.0	0.100	0.1	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.1	0.0	0.0	0.0
12	11.0	1.0	0.00	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
13	12.0	0.0	0.00	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
14	13.0	0.0	0.00	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.400000	0.0	0.0	0.0	0.0	0.0
15	14.0	0.0	0.00	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
16	15.0	0.0	0.00	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
17	16.0	0.0	0.00	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
18	17.0	0.0	0.00	0.0	0.0	0.100	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
19	18.0	0.0	0.00	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.10	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
20	19.0	0.0	0.00	0.0	0.0	0.100	0.1	0.1	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.1	0.0	0.0	0.0
21	20.0	0.0	0.00	0.1	0.0	0.000	0.1	0.1	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.1	0.0	0.0	0.0
22	21.0	0.0	0.20	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.1	0.0	0.0
23	22.0	0.0	0.20	0.0	0.0	0.000	0.0	0.0	0.0	0.1	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
24	23.0	0.0	0.00	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.25	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
25	24.0	0.0	0.00	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
26	25.0	0.0	0.00	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.333333	0.0	0.0	0.0	0.0	0.0
27	26.0	0.0	0.30	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.1	0.0	0.0
28	27.0	0.0	0.00	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
29	28.0	0.0	0.00	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.00	0.25	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
30	29.0	0.0	0.00	0.0	0.0	0.000	0.1	0.1	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.1	0.0	0.0	0.0
31	30.0	0.0	0.00	0.0	0.0	0.000	0.1	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.1	0.0	0.0	0.0	0.1
32	31.0	0.0	0.00	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.00	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
33	32.0	0.0	0.00	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.20	0.00	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
34	33.0	0.0	0.00	0.0	0.0	0.000	0.0	0.0	0.0	0.0	...	0.00	0.10	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0

And then, I ran a K-Nearest algorithm to cluster neighborhoods into Clusters: 5 Clusters found based on K Nearest Algorithm:

Based on the frequency of most common venue, I named each cluster based on their characteristics:

Cluster 0: Coffee Shop Area
Asian Restaurant 3
Coffee Shop 7
Convenience Store 1
Hotel 2
Japanese Restaurant 2

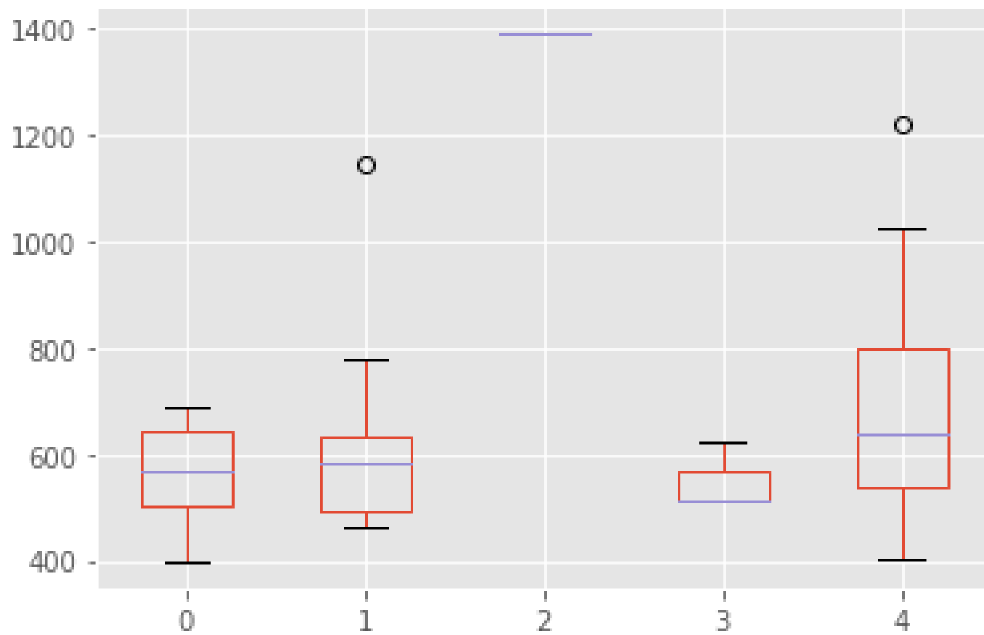
Cluster 1: Foreign Restaurants Area
Chinese Restaurant 1
Cocktail Bar 1
Grocery Store 2
Hotpot Restaurant 1
Italian Restaurant 4
Mexican Restaurant 1
Noodle House 1
Yunnan Restaurant 1

Cluster 2:
Antique Shop 1

Cluster 3: Park Area
Park 2
Tennis Court 1

Cluster 4: Modern Lifestyle Chinese Restaurant 2
Coffee Shop 4
Coworking Space 1
Gym 1
Hotel 3

To evaluate if different venue has different price performance, I drew a box plot. Although it's not very significant, the chart shows Antique shop area and Modern Lifestyle area outperforms in price:



Visualization

Finally, as one of the objectives, I did a visualization to show different cluster and their price performance. This could be beneficial for both investors and customers: