Capstone Project: NYC Airbnb data clustering analysis and recommendation

Applied Data Science Capstone by IBM/Coursera

Table of contents

1. Introduction

1.1 Background

I have been living in the New York City for 3 years and it has been a dream tour place for a lot of people. The prices of the hotels in the NYC is notoriusly high and as a result, while some people are rich enough, most people prefer to find alternative residences. Some of them decided to live far a way from the city and take a bus or train there for tour, while others chose to live in an Airbnb.

1.2 Problem

It is always hard to find a good Airbnb for travelers new to the New York City. There are a lot of factors to considers, such as food, neighbourhood, locations, house reviews, availabilities, and etc. Customers usually have to do intensive research on the locations to mine the information about the factors they care, and this could be time consuming. Often people relies on some online reviews or recommendations but his is also inefficient. In this case, a clustering of the house choices could be both commercially and economically valuable.

1.3 Methods

To create the model the recommendation system, I used the KMeans unsupervised learning to cluster the houses first, and then grouped their centroids, and choose centroids with the criterion from the customers, and generate recommendatin from the centroids for the customers

2. Data Descriptions

2.1 Data Acquisition

The data comes from Kaggle.com, which is a platform for data science projects data as well as competitions. This datasets is originally from Airbnb website and is therefore an official data. It describes the 2019 Airbnb house providers' listing informations. (Link: https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data) The data are the following:

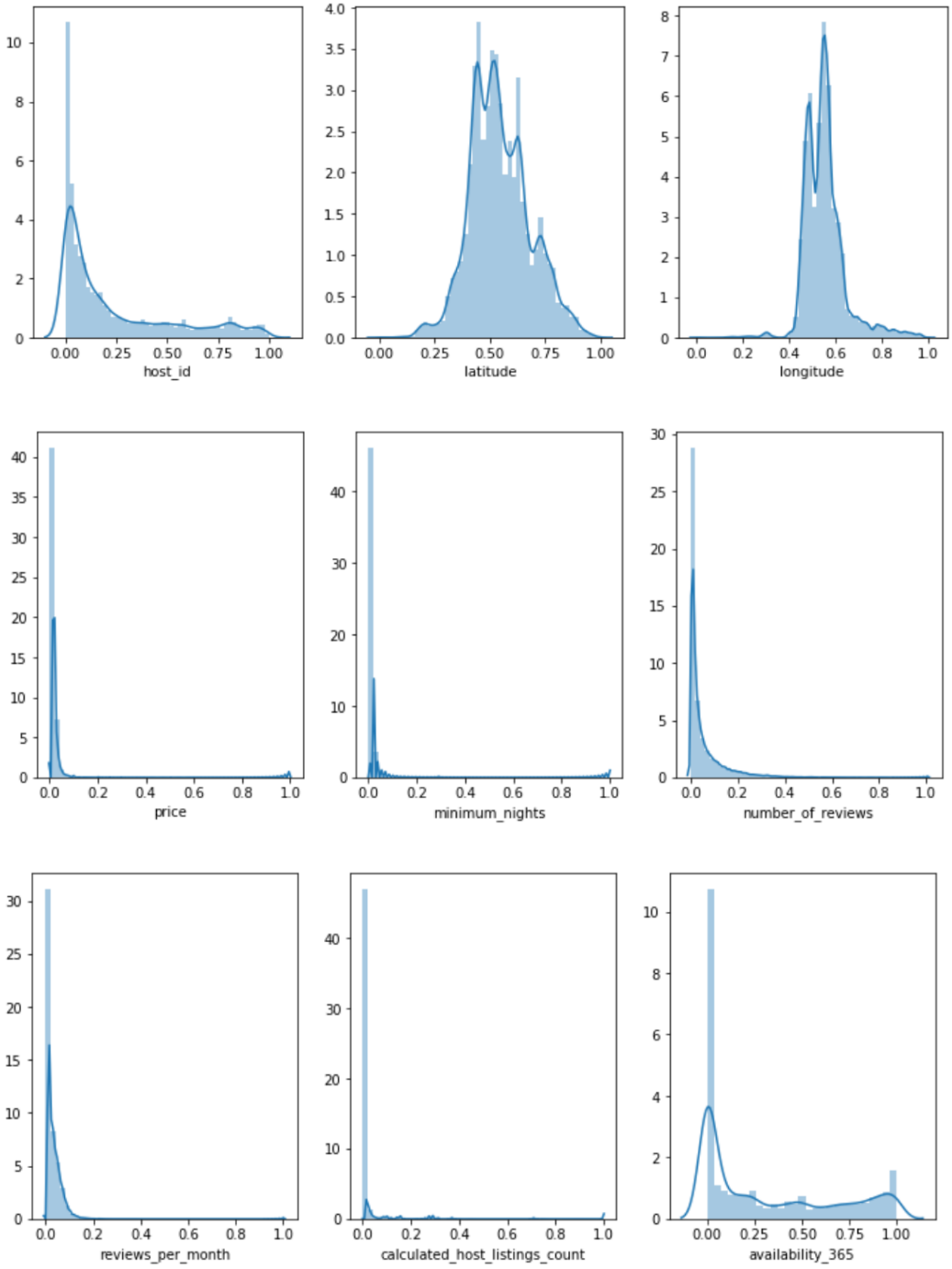
1. location data
2. review data
3. price data
4. neighborhood data
5. availability

2.2 Data Cleaning

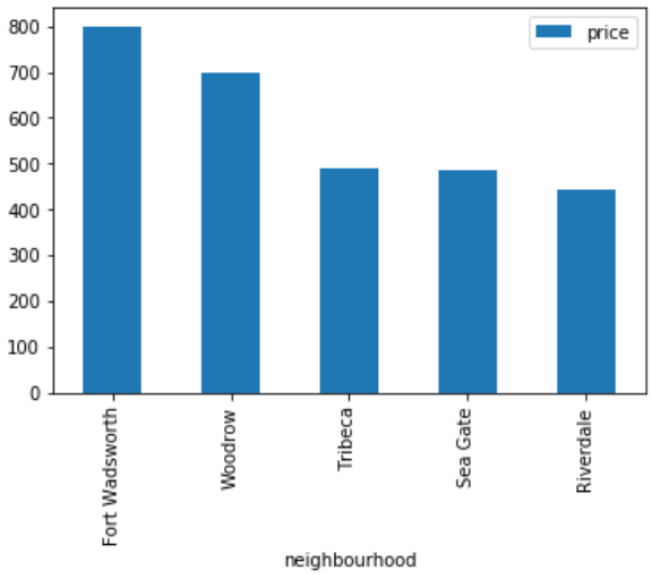
The Dataset from the the Kaggle about Airbnb data was separated into two tables, numerical data and categorical data. The two tables were normalized and encoded and then combined back into one tables for modeling. The main package of the whole cleaning process is Scikit-Learn

2.3 Exploratory Data Analysis

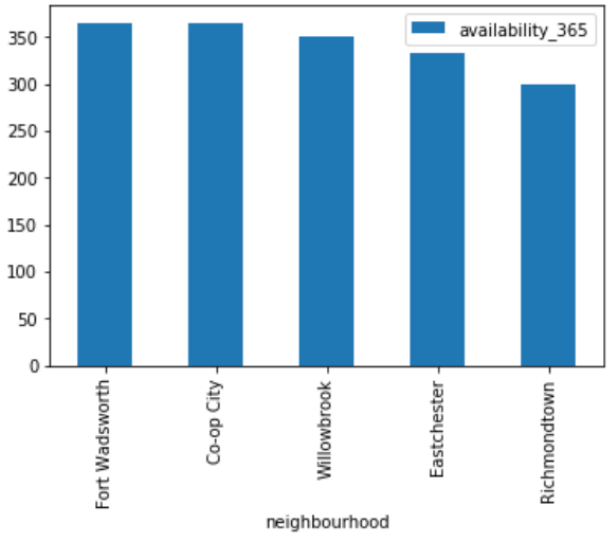
a) Normalized data distribution:


host_id
latitude
longitude
price
minimum_nights
number_of_reviews
reviews_per_month
calculated_host_listings_count
availability_365

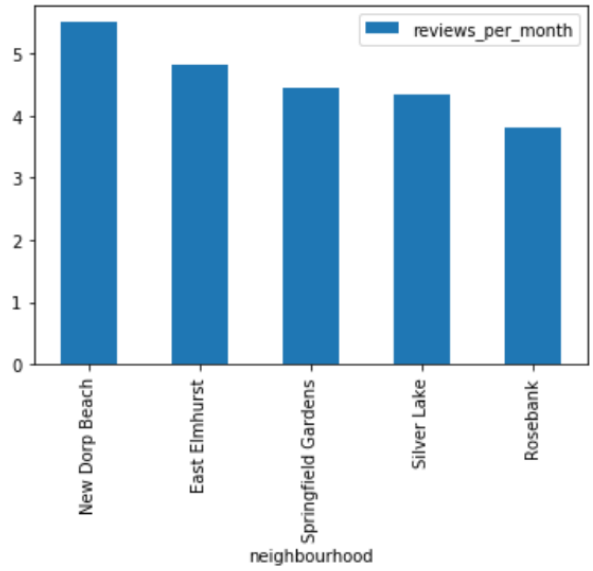
b) Rank:



average price by neighborhood top 5



average availability by neighborhood top 5



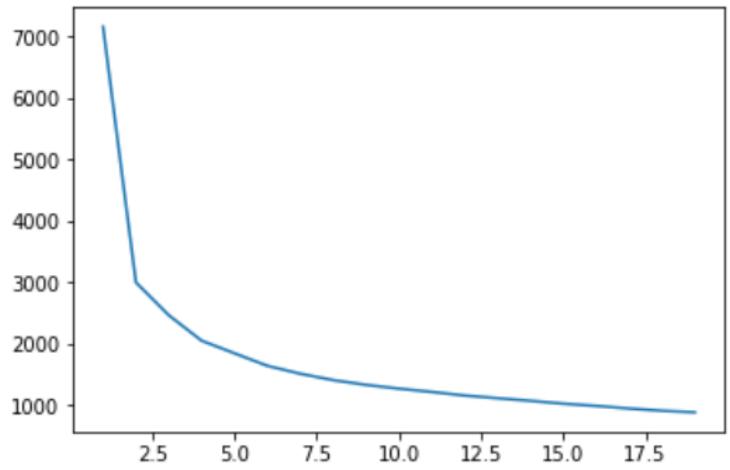
average review per month by neighborhood top 5

3. Modeling and Analysis

3.1 Clustering

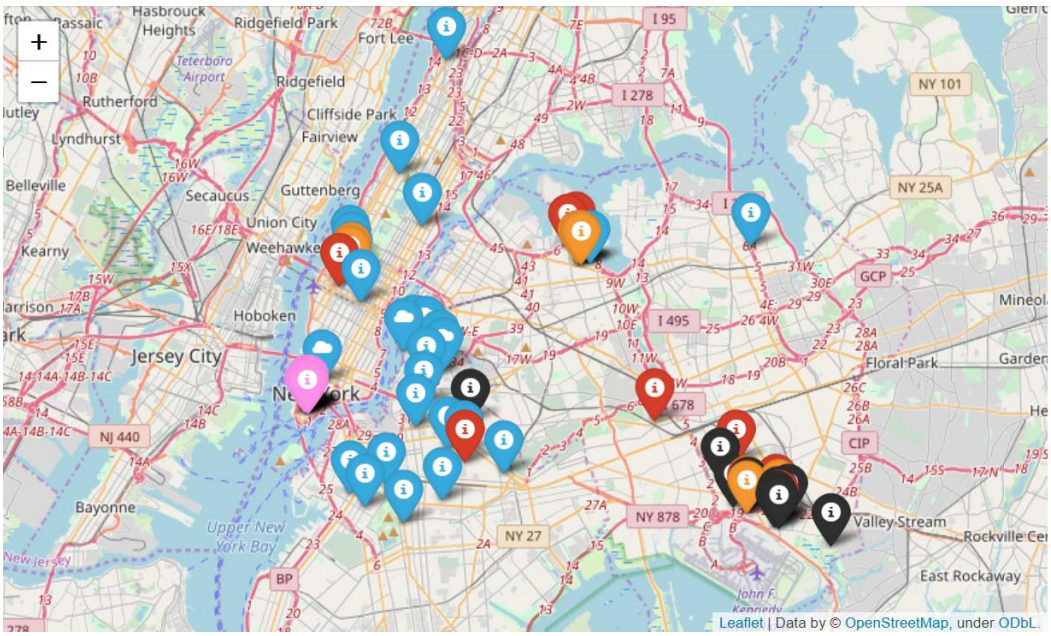
Using the K-Means algorithm to cluster the housing data and generate the report, and then based on the report to make recommendation.

To find out the best K value for the K-Means algorithm, elbow method, which is an empirical method, is used. It finds the proper K value by finding out the point where the marginal decrease in the total distance drops.



In this case, when k equals to roughly 2, the marginal decrease drops the most, but 2 clusters would be meaningless. Instead I chose k equal to 6, after which the marginal decrease becomes really low.

By clustering the data we get the following result:



The different color represents different groups, and from the map we could tell the cluster was sort of successful with similar housing options in the same cluster.

3.2 Recommendation system

Note this is not a collaborative filtering or content based filtering! This is a simple ranking method which only allow us to choose the one most important factor!

We first generate a rank chart like this:

	name	price rank	minimum_nights rank	number_of_reviews rank	reviews_per_month rank	calculated_host_listings_count rank	availability_365 rank	last_reviev
0	Cluster Group 0	3	2	3	5	5	5	
1	Cluster Group 1	0	3	5	1	1	1	
2	Cluster Group 2	2	0	1	2	4	4	
3	Cluster Group 3	5	4	2	3	0	0	
4	Cluster Group 4	4	1	4	0	2	3	
5	Cluster Group 5	1	5	0	4	3	2	

And then based on customer needs, calculate the rank with weights and find out the cluster with the highest score and make the recommendations

3.3 Example: a customer caring only price and review per month

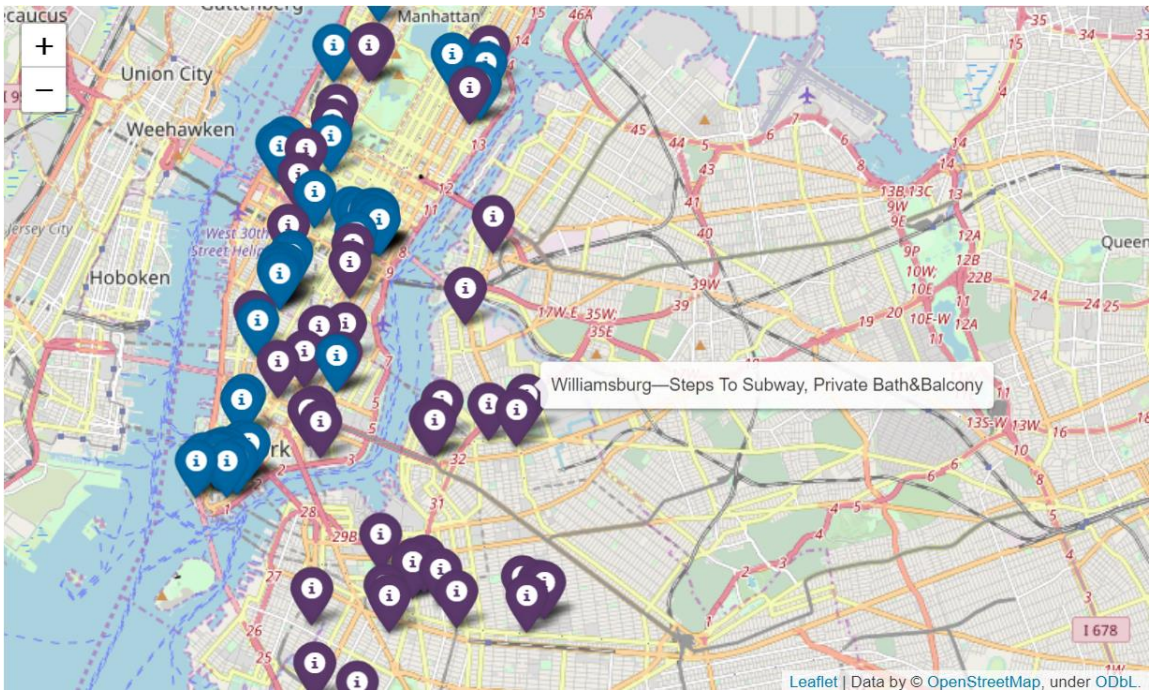
Suppose that we have a customer who are sensitive to price and the reviews per month equally, what should he or she choose?

First we find out the cluster with the highest score:

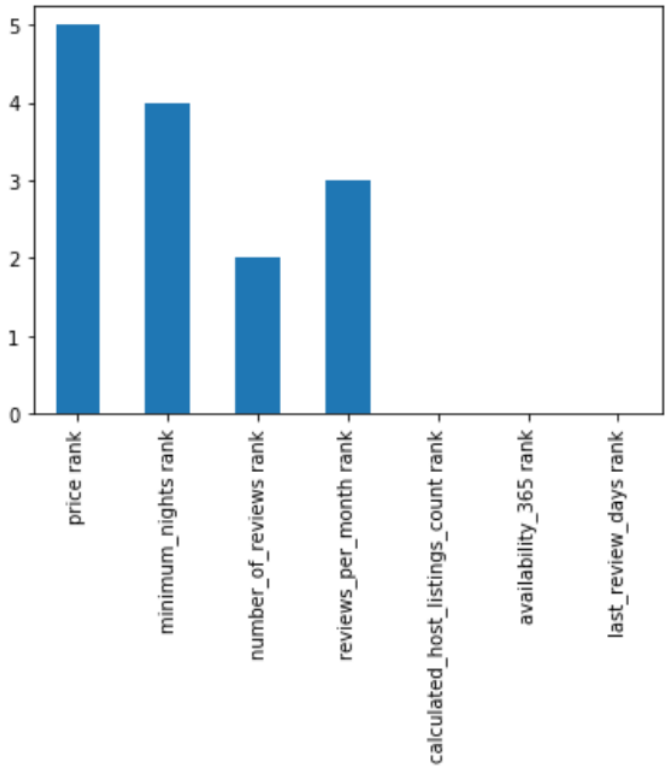
```
number of clusters recommended:  2
the clusters recommended:  [3 0]
```

It looks like with this data, we have two cluster with the same score!

Let's make recommendations:



Let’s take a look at cluster 3 that we recommended:



The cluster 3 has a relatively well reviews per month score and a really good price sccore.

Note that the in the chart the Y-axis is the score value of the rank, and 5 means the cluster is the best in this particular aspect.

As the map suggests, if the customer cares about both price and the review, there are a variety of choices. On manhattan, the majority of the recommendations lies on Financial District, West Village, and East Village, where the housing price is relatively better. At the same time, per the personal experience, there are a lot of housing available in East Village as well. The community is cozy and quiet, and is very nice to live in. It's no wonder why it has the best score in the number of review, which suggests about the popularity of a place.

In Brooklyn, the maojority lies in Williamsburg and Downtown Brooklyn. Both are very popular place nowadays even for New Yorkers to live, and the price is much lower than on the manhattan.

4. Conclusion

The purpose of this project is to build a model that could provide insights for people looking for airbnb in NYC through clustering analysis. The data comes mainly from Kaggle and the model is built mainly on Scikit-Learn.

The recommendation system is based on clustering analysis. We use this version because we don't have access to enough data to make a content based filtering or collaborative filtering. As a result, we applied Clustering method of KMeans to group the choices, and then use weighted score depending on customer needs to choose the cluster. This isn't the most accurate way but it could provide insights for customers and save their time.Through this clustering analysis, customer could add their preference on the community by focusing on the area that they are interested in. The final decision would be made based on customer's other consideration

The model could be further improved by mining more data and features. For example, how long does it take to go from the house to the main tourist sites? How many good restaurant are there within reach? What's the safety issue there? More data could improve the clusters.