Final Report

Vu Minh Tuan - Data Scientist

Introduction/Business Problem:

The crux of the problem that I am solving lies in the fact that we travel due to social or official reasons all the time throughout the world in different cities in different countries. While we do that, we usually reserve the hotels for our temporary stay. Over the years many systems have been developed to bring the cornucopia of hotels to our grip where we can choose from a uncountable options. But what about the **loan rangers** or the adventure seeking **solo travellers** or the **group of backpackers** who travel just for fun or in a whim. There is no instant solution for those. We can also imagine a scenario where there are some troubles with your hotel reservations or you missed the reservation and you need a place to stay. All these problem can be solved by this application. ***It tracks your current location and gives you a list of available hotels in close proximity and gives you the direction(of course, we don't want you get lost).*** What if we could leave the responsibility to our machine and algorithm. That could save us a lot of time. The user that we can approach through this system is immeasurable. The hospitality industry is multi-billion dollar industry and one of the biggest in the world and by implementing our platform we can simplify a humungous task that will surely be appreciated by a huge number of users.

Data:

Our data has 2 main components.

1. New York location data in JSON format. The dataset contains all the information about the boroughs and the neighbourhoods present in New York. Apart from the names, the co-ordinates of the neighbourhoods are the primary data that we shall use from this dataset.
2. The other part of the data comes from the foursquare API. We use two types of queries for fetching the data from the foursquare API.
    - First type of query is ‘explore’ that is used to fetch the venues present in a 1.5 km radius of the neighbourhood of our target.
    - Second type of query that I am using is “venues”. This query is used to check the details about the venues that are hotels and get those details

about them since we are interested in only those venues that are hotels. We check those venues by using their venue id.

In the ultimate stage of execution of the idea there will be an app that will track your location and tell you the list of hotels near you. The app will show the rating of the hotels and the distance of the hotel from your current location. The idea is inspired by the mobile app called Mr. Jitters. The app looks for your current location and when it finds you an icon appears on the screen and when you tap the icon it tells you the location of the cafés near you in a walking distance with their ratings and the distances. The app is also open sourced on GitHub.

User Interface :

So we shall not take the headache of developing an app, as it is out of the scope of the course. Instead we shall establish that the backend of the idea works perfectly. So we shall feed fixed location through queries in Foursquare API and then explore the location for venues and then we shall look for venues that has category Hotel. We shall then look for the ratings of those hotels and the distance from our current co-ordinate and put it in a data-frame. This data will be available for the user.

Our Data :

We shall keep track of the searches by the users and since the topic of this capstone is **"Battle of Neighbourhoods"** we shall compare the hotels of one neighbourhood to other neighbourhoods.

Machine Learning :

For our own interest we shall use the ratings, number of tips, number of likes of the hotels to cluster the hotels in a certain neighbourhood. We shall then try to identify those clusters and after creating clusters in different neighbourhoods. We shall be able to compare hotels in different neighbourhoods.

Methodology

- First we read the New York location data from the json file and created the neighbourhood dataframe for Newyork. This dataframe is then used to project the neighbourhoods on the map of New York.
- We chose 2 boroughs from the neighbourhoods i.e. **Manhattan and Staten Island**. So we filter the main neighbourhood dataframe and create 2 separate

dataframes for Manhattan and Staten Island. The dataframe contains 4 columns : **Name of the Borough, Name of the Neighbourhood, Latitude, Longitude**

- Our next task is to use the Foursquare data to explore the neighbourhoods in **Manhattan and Staten Island borough**one by one and make a venue dataframe of the for Manhattan and other one for Staten Island. The dataframe consists of **Neighbourhood Name, Neighbourhood Latitude, Neighbourhood Longitude, Venue Id, Venue Name, Venue Latitude, Venue Longitude, Venue Distance and Venue Category**. We do this for both the boroughs. **Our explore URL has a radius of 1500m.**
- To get the list of the hotels our next job is to filter the venue dataframes and get the venues which are hotels. After this our approach has 2 aspects.

1. User Interface
2. Backend Statistical Analysis

User Interface :

Using the venue id from the **explore** feature of the Foursquare API we write another URL that uses the **venue id** to get the information about the venues. Using the **venues** feature of the Foursquare API we get the **Ratings** of the hotels that is necessary for the User Interface. Using columns excluding ***'Neighborhood Latitude','Neighborhood Longitude','Venue Id','Venue Latitude','Venue Longitude','Venue Category'*** from the **manhattan_hotels** dataframe and ***'Rating'*** column from the **manhattan_hotels_data** dataframe we create the user interface for the Manhattan Hotel data called user_sees. Like below :
|Neighbourhood|Hotel Name |Distance |Rating| |----------:|:---------:|:-------:|:----| |Chinatown |Hotel 50 Bowery NYC| 214m |8.9| |Chinatown |Crosby Street Hotel| 866m |9.3| |Manhattanville| Aloft Harlem | 1003m|8.1|

We do the same for the Staten Island borough. Here we use the columns ***Neighborhood, Venue Name, Venue Distance***from the **staten_hotels** dataframe and rename them as ***'Neighborhood','Hotel Name','Distance'***. We also include the column ***Rating*** from the **staten_hotel_data** dataframe and we create the dataframe for the user inteface.
|Neighbourhood|Hotel Name |Distance |Rating| |----------:|:---------:|:-------:|:----| |Rosebank |Staten Island Motor Lodge| 954m| Not Rated yet| |Travis |Staten Island New York Hotel | 55m| Not Rated yet| |Travis |Comfort Inn| 46m | 4.8 |

Backend Statistical Analysis

Here we shall cluster the hotels into categories of similar feature. **manhattan_hotels_data** has many columns. The ones we shall use for clustering are ***Like Counter, Tip Counter, Rating*** and create the dataframe **manhattan_hotels_cluster**. We do the same for Staten Island hotels and create the **staten_hotel_cluster** dataframe.

Result :

The result from the clustering shows that we can categorise the hotels of Manhattan and Staten Island of similar category. For Manhattan we can say that the hotels with category label:

- 0 -> The number of **'*Likes*'** and **'*Tips*'** are high but not as high as those in Cluster 2. The ratings of the hotels are mixed but they are mostly high.
- 1 -> The number of **'*Likes*'** and **'*Tips*'** are comparatively much lower than the hotels in the other clusters. The ratings are moderate and not as good as the hotels in the other clusters.
- 2 -> These hotels have a very high number of **'*Likes*'** and **'*Tips*'**. Although the ratings of any of the hotels are not out of the charts they are really high.

For Staten Island we can say that the hotels with category label:

- 0 -> These hotels have a higher number of **'*Likes*'** and **'*Tips*'** than other clusters. Although the ratings of any of the hotels are not out of the charts they are really high.
- 1 -> The number of **'*Likes*'** and **'*Tips*'** are comparatively much lower than the hotels in the other clusters. The ratings are moderate and not as good as the hotels in the other clusters.
- 2 -> The number of **'*Likes*'** and **'*Tips*'** are high but not as high as those in Cluster 0. The ratings of the hotels are mixed but they are mostly high.

Discussion :

We can check the hotel dataframes for Manhattan and Staten Island and see that there are more options of good hotels in Manhattan borough than that in Staten Island. One

observation that we can make on the Staten Island dataframe is that some of the hotels do not have Ratings and they are shown in the dataframe as **Not rated yet**. We had to remove those hotels from the dataframe for clustering since Rating is one of our clustering parameters. By checking the Rating and other features like, **Number of Likes** and **Number of Tips** we can straightaway tell that Manhattan has many more good hotels than Staten Island has. Not only that, the number of hotels in Staten Island is lesser than that in Manhattan. Also since some of the hotels are not rated in Staten Island we can make few guesses about them :

1. They might not be good enough for being rated.
2. The residents might be too ignorant to rate the hotels.
3. The ratings might not be registered in Foursquare and it might be in some other website. Since the rating of other hotels in the Staten Island borough is not that good we can guess that the 1st option is highly probable. Also it can be said from the number of hotels present in the boroughs that Staten Island is fairly a smaller borough than Manhattan. On the other hand there are many options in Manhattan and the hotels are rated highly. The number of likes in most of the hotels are more than the hotels in Staten Island.

Conclusion :

We have gone through a process of accessing data from the Foursquare API to create our dataframe so that we can show the necessary data to the users of our app. All they need to know is the **Name of the Neighbourhood, Name of the hotel, Rating, Distance from your current location**. Other information that we have collected are used for our own analysis. We have used the venue id for using the **Venues** feature of the Foursquare API. Then we got the Rating, Number of Likes, Dislike Flag, Number of tips etc. Since none of the hotels were disliked that column was not used for clustering. Other three columns were used for clustering. Since clustering was used for grouping similarly featured hotels distinguished them. We also learnt from the foursquare data that Manhattan has more and better options than there is in Staten Island.