**Exploring the Relationship of Uber Pickups and Venues of New York City**

**1. Introduction**

I explored New York City, segmented and clustered its neighborhoods based on their popular venue categories following the instructions of the course lab. The biggest clustered dataset showed that there were many coffee shops, cafes, restaurants of various cuisines, and bars and clubs. It also had health-related venues such as gyms, yoga studios, cycle studios, etc. The data clearly showed that the city is very diverse.

In a big city like New York City, it is very popular to take taxes or use ride sharing services like Uber and Lyft. The Uber Pickups in New York City is available on kaggle.com. I would like to further explore New York City combining the Foursquare’s location data with the Uber Pickups data to see if venue categories are related to Uber usage count. This may discover popular pickup locations and this finding may be helpful for the city to better understand how much traffic increase in certain areas by Uber.

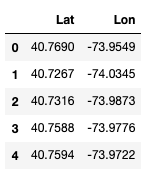
**2. Data Acquisition and Cleaning**

2.1 Data Sources

The Uber Pickups in New York City dataset can be found on Kaggle.com [[link](https://www.kaggle.com/fivethirtyeight/uber-pickups-in-new-york-city)]. This dataset contains 6 months of Uber pick up geo-locations (latitudes and longitudes) of year 2014. Location information along with venue details will be dynamically obtained by utilizing the Foursquare location APIs.

2.2 Data Cleaning

The Uber Pickups data is a zipped file (Uber-dataset.zip) and contains 6 CSV files in it. And each CSV file contains pick up data for each month from Apr through Sep 2014. The individual file has 4 attributes (columns): Date/Time, Lat, Lon, and Base. Latitude and longitude data is required to query nearest venues with the Foursquare APIs, but the other columns, Date/Time and Base, are not needed for the analysis. So I dropped these two unnecessary columns from the dataset and combined all the data from the 6 CSV files and created a table as shown in <Figure 1>. In order to read CSV files in sequence with for loop, I renamed the file names with index number at the end: uber-raw-data-0.csv uber-raw-data-1.csv, …, uber-raw-data-5.

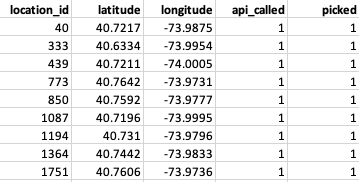


<Figure 1>

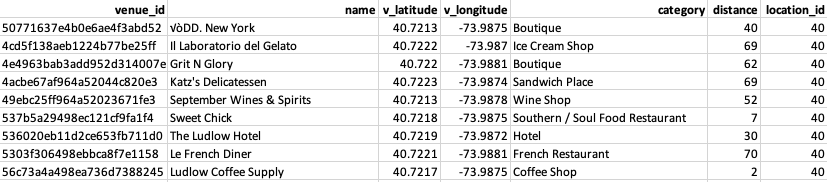
I created a table with all the location data and there was 4,534,327 rows. I then inserted the entire data into the Locations table in the SQLite database as shown in <Figure 2>. It was not possible to call the Foursquare API for all the locations in the table due to the daily limit, I randomly chose 8,000 locations among them. Utilizing the SQLite database, I was able to keep track of locations that had been used to get nearby venues by calling the search API (<https://api.foursquare.com/v2/venues/search>).

I collected venues within a radius of 75 meters from each coordinate (latitude and longitude) in the Locations table. I selected 75 meters after testing several shorter distances. The API did not return any nearby venues for many pick-up locations for shorter radiuses. 75 meters was big enough to find more than one nearby venues. Since the API also provides a distance between the pick-up location and the venue, I can later use only the first n venues listed in the order of distance.

I created two more Foursquare developer accounts in order to speed up collecting nearby venues. However, it still took several days to call the APIs for all those 8,000 randomly chosen locations. I put all the collected venues’ information such as venue\_id, name, latitude, longitude, category and a distance from the pick-up location into another table, Venues, in the database as shown in <Figure 3>.



<Figure 2>



<Figure 3>

As the next step, I picked only two closely located venues from each pick-up location in order to reduce duplicated venues that may be located in the overlapped areas within radiuses of multiple pick-up locations as shown in <Figure 4>.

75m

Location B

Location A

Picked for Location A

Picked for Location B

Veneus

<Figure 4>

This was achieved through a few steps of data processing. First, I got all the venue data from the Venues table in the database and created a Pandas dataframe. As the second step, I sorted the dataset with these two columns: location\_id and distance, and got the results as shown in <Figure 5>. Lastly, I grouped the dataset by the location\_id column and picked only the top two rows by using head(2) method. The results are shown in <Figure 6>.



<Figure 5>



<Figure 6>

2.3 Feature Selection

[To be continued]