Taxi Company App Analysis

Himanshu Hooda
Lambton college
Toronto, Canada
c0814556@mylambton.ca

Hitesh Pathania
Lambton college
Toronto, Canada
c0823824@mylambton.ca

Jayesh Patil
Lambton college
Toronto, Canada
c0817392@mylambton.ca

Rithik Uppal
Lambton college
Toronto, Canada
c0823009@mylambton.ca

I. INTRODUCTION

This project is based on the data analysis and visualization of the taxi trip. Using the various attributes of the dataset we have plotted graph n the basis of which some insights can be found and used to make the services better.

II. DATASET

We have used 2016 NYC Yellow Cab trip record data. Shape of the dataset was 1458644 rows and 11 columns. The dataset consists of following attributes:

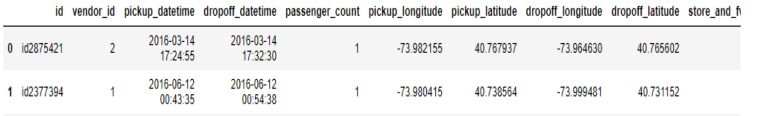
- id - unique trip id
- vendor_id - a code identifying the service provider linked with the trip record
- pickup_datetime - date and time of pick up
- dropoff_datetime - date and time of drop off
- passenger_count - the number of passengers in the vehicle.
- pickup_longitude - the longitude of pickup
- pickup_latitude - the latitude of pickup
- dropoff_longitude - the longitude dropoff
- dropoff_latitude - the latitude dropoff
- store_and_fwd_flag - Y=store and forward; N=not a store and forward trip
- trip_duration – trip duration (in seconds)

III. DATA PREPROCESSING

Importing Libraries: The first step for this will be importing the required libraries that can be used for cleaning, analysis and visualization of the data. Following libraries are used in the project:

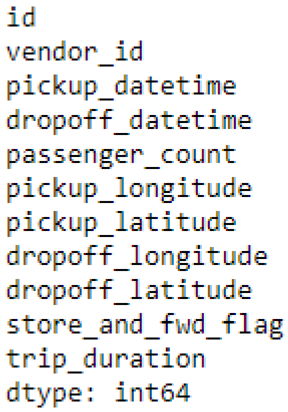
- Pandas
- Numpy
- Seaborn
- Matplotlib
- Datetime
- Folium
- Math

Read the Data: After importing the library we first need to read the csv file on jupyter notebook.



	id	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	store_and_f
0	id2875421	2	2016-03-14 17:24:55	2016-03-14 17:32:30	1	-73.982155	40.767937	-73.964630	40.765602	
1	id2377394	1	2016-06-12 00:43:35	2016-06-12 00:54:38	1	-73.980415	40.738564	-73.999481	40.731152	

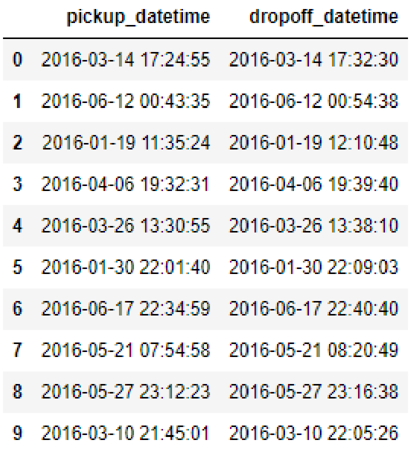
Handling missing values: The null values can be checked by using isnull() with the sum() this will give us the count of null values present in each columns**.**



```
id                    0
vendor_id             0
pickup_datetime       0
dropoff_datetime      0
passenger_count       0
pickup_longitude      0
pickup_latitude       0
dropoff_longitude     0
dropoff_latitude      0
store_and_fwd_flag    0
trip_duration         0
dtype: int64
```

As there was no missing value in the data we continue with the preprocessing steps.

The pickup and drop off date and time column was further divided into date, time, hour and day of the week using the datetime library.



	pickup_datetime	dropoff_datetime
0	2016-03-14 17:24:55	2016-03-14 17:32:30
1	2016-06-12 00:43:35	2016-06-12 00:54:38
2	2016-01-19 11:35:24	2016-01-19 12:10:48
3	2016-04-06 19:32:31	2016-04-06 19:39:40
4	2016-03-26 13:30:55	2016-03-26 13:38:10
5	2016-01-30 22:01:40	2016-01-30 22:09:03
6	2016-06-17 22:34:59	2016-06-17 22:40:40
7	2016-05-21 07:54:58	2016-05-21 08:20:49
8	2016-05-27 23:12:23	2016-05-27 23:16:38
9	2016-03-10 21:45:01	2016-03-10 22:05:26

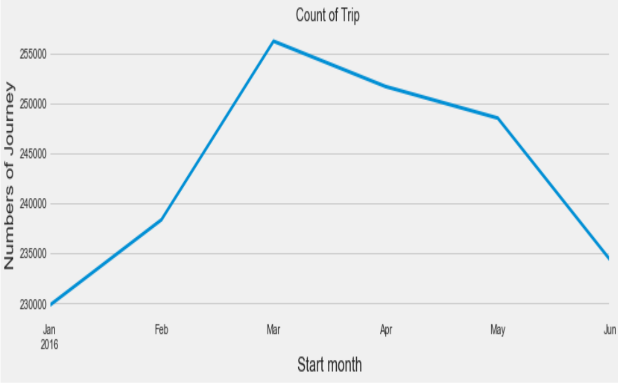
This helped us in further analysis of the data for plotting different visualization of differen attributes present in the dataset.

IV. Data Visualization

To Visualize the dataset, we used matplotlib and seaborn library and used the following plots:

- Bar chart
- Density plot
- Heat Maps
- Line chart
- Scatter plot

Month with highest trip volume: To get the highest number of trips we used pickup_datetime in monthly time period which was set using “.to_period(‘M’)”.



From the above visualization we can find out the that mon of the march has the highest volume of trip.

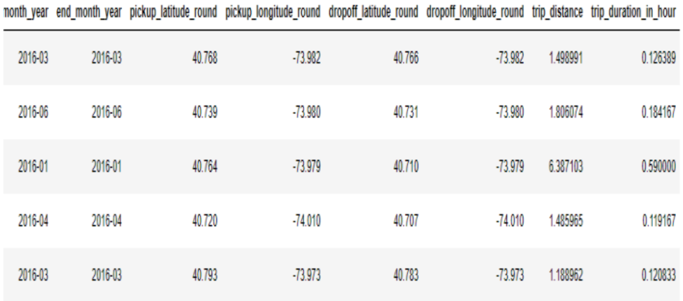
After this we calculated the trip distance which will be helpful for further analysis. Following formulas was used to do so:

*a = sin(d_lat / 2)** 2 + cos(pickup_lat) * cos(dropoff_lat) * sin(d_lon / 2)**2*

*c = 2 * atan2(sqrt(a), sqrt(1 - a))*

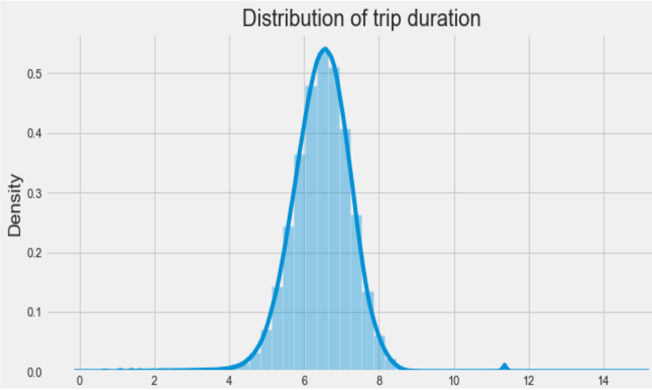
*distance = R * c, where R is approximate radius of earth in km*

Then we converted the trip duration in hours by using the lambda function.

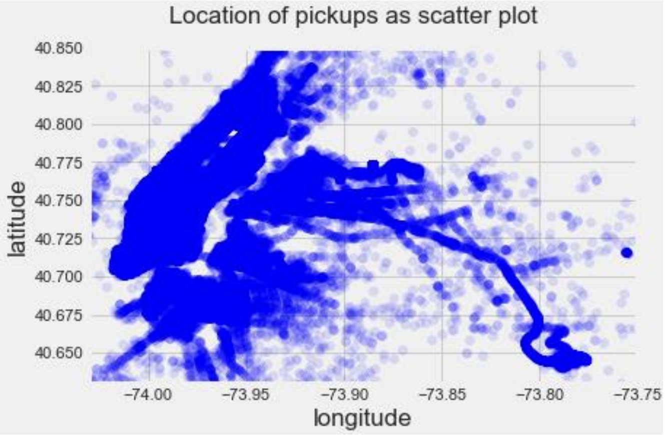


nonth_year	end_month_year	pickup_latitude_round	pickup_longitude_round	dropoff_latitude_round	dropoff_longitude_round	trip_distance	trip_duration_in_hour
2016-03	2016-03	40.768	-73.982	40.766	-73.982	1.498991	0.126389
2016-06	2016-06	40.739	-73.980	40.731	-73.980	1.806074	0.184167
2016-01	2016-01	40.764	-73.979	40.710	-73.979	6.387103	0.590000
2016-04	2016-04	40.720	-74.010	40.707	-74.010	1.485965	0.119167
2016-03	2016-03	40.793	-73.973	40.783	-73.973	1.188962	0.120833

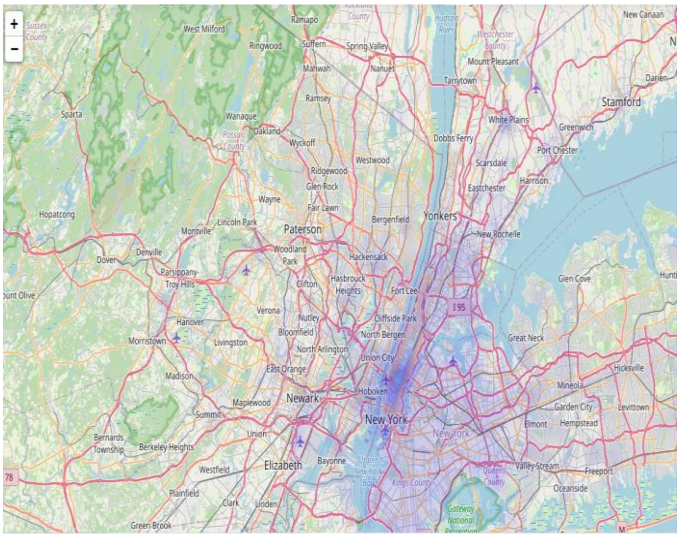
Distribution of trip duration: This was plotted using the tirp_duration attribute. A distplot shows the histogram with a line in combination.



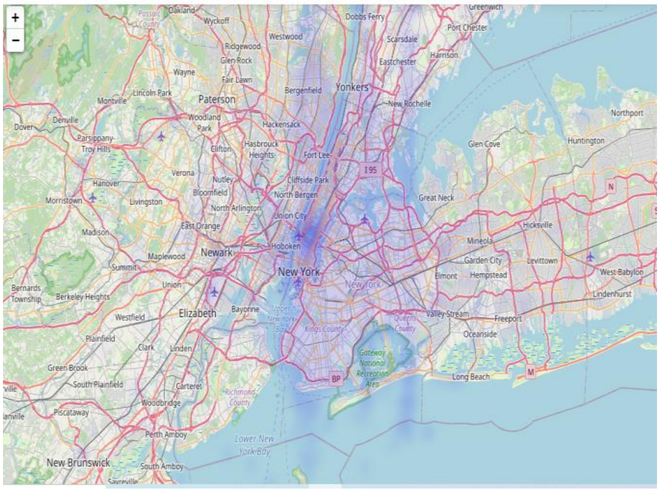
Scatter plot of pickup location: The location of pickups was plotted using the latitude and longitude from the dataset.



Heat map of pickup and dropoff location: These are plotted using longitude and latitude of pickup and dropoff location. The plot of the pickup and drop f is shown below:

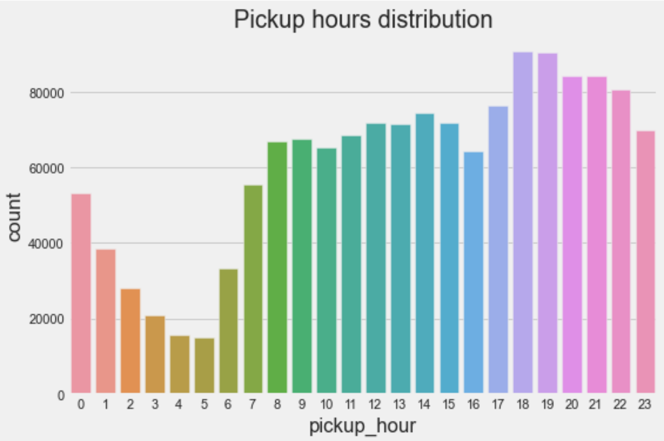


The bluish region represents the most the common pickup location.

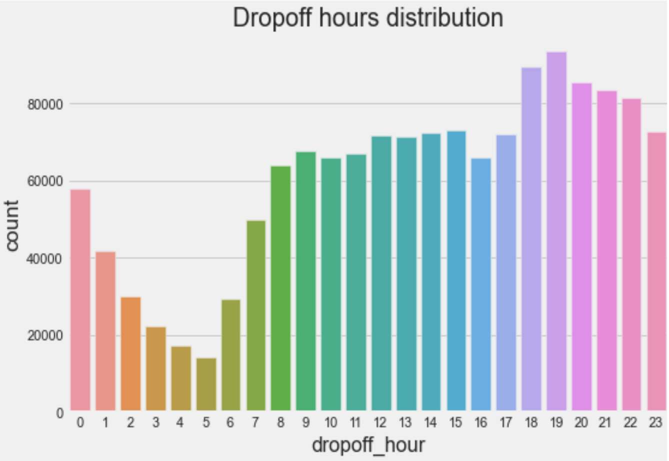


Again in this region the bluish region represents the common dropoff location the darker the region is the ost common spot it is for dropoff and pickup.

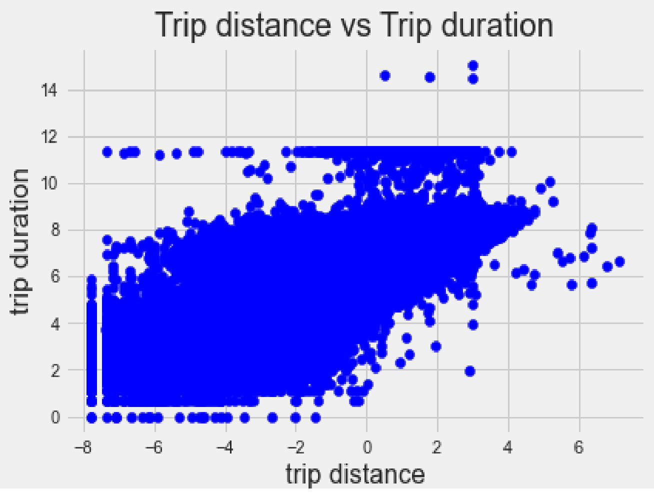
Distribution of Pickup and Dropoff hours:



This shows that most pickup occur at 6 and 7 pm on average.

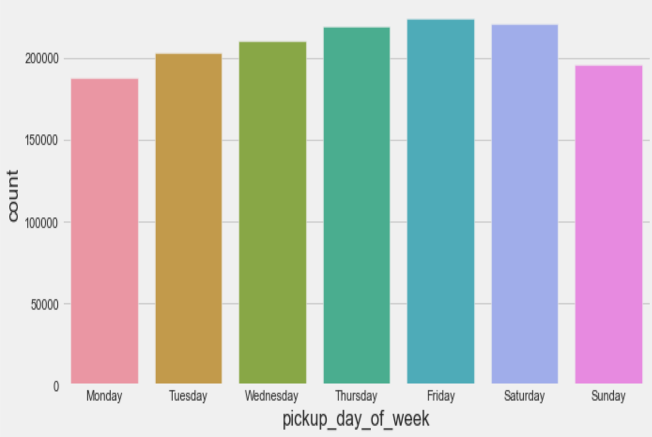


This shows that most dropoff also occur around the same time as pickup.

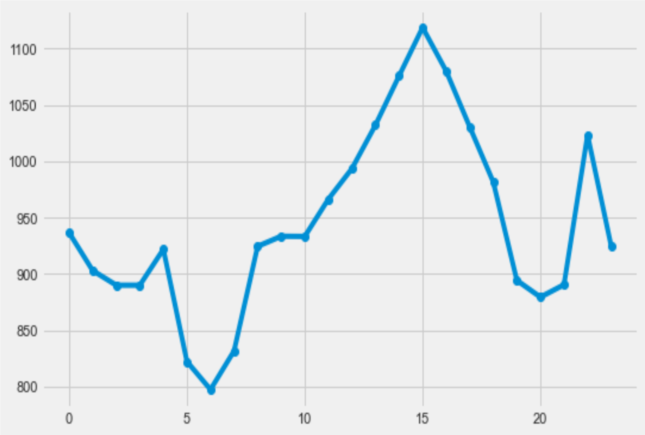


The above graph represents the distribution between trip distance and duration.

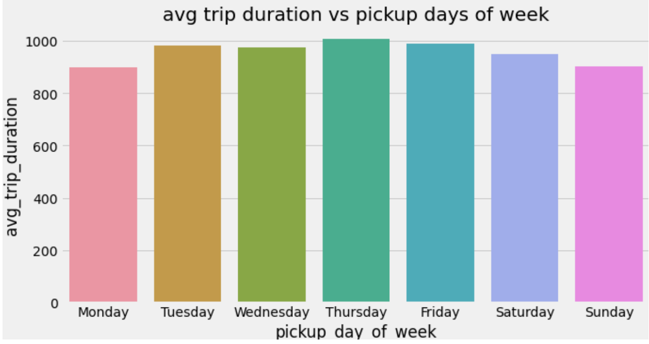
Count of pickup on different day of week: A countplot was made for the pickup days of the week which we acquired from the pickup datetime column.



Monday has the least amount of pickup ut of all the days in a week.

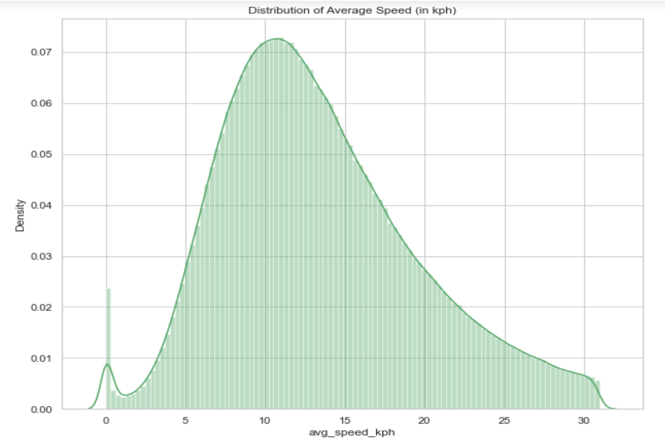


Above plot represents the average trip duration for pickuphours. The average trip duration can also be used to find the different durations during a day of week.

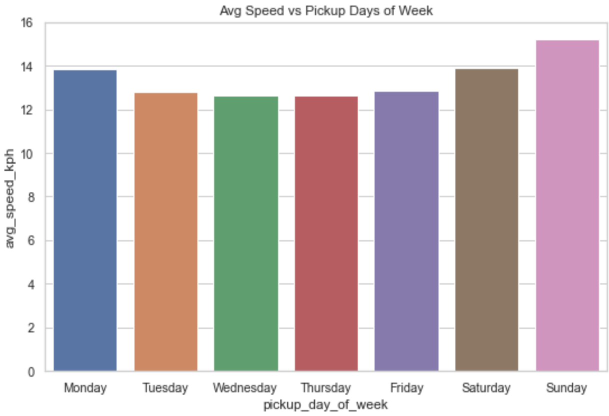


Average speed : The average speed(in kph) is calculated using this formula: *train['avg_speed_kph']=train['trip_distance']/train['trip_duration_in_hour']*

Distribution plot of the average speed after removing the outliers is shown below:



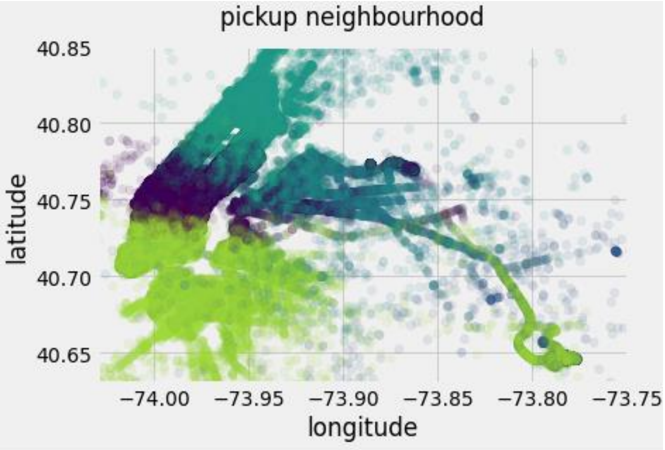
After this we plotted a graph to find average speed on each day of a week. From which we found out that the average speed is higher on Sunday than any other week of day.



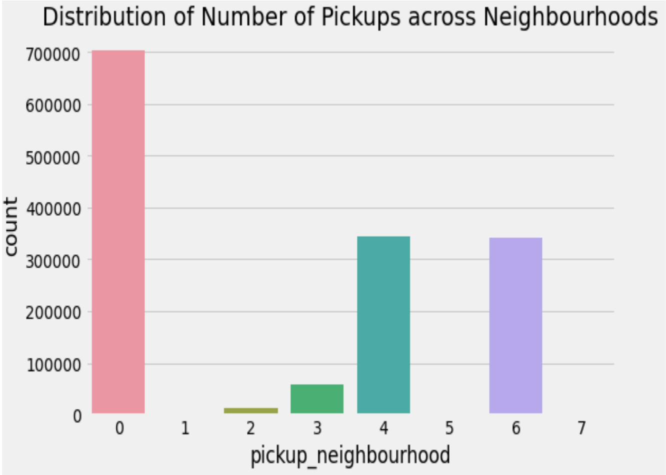
V. Data Modelling

In this project we used Kmeans clustering algorithm to make clusters of neigbourhood on the scatter plot using the pickup and dropoff locations as the attributes.

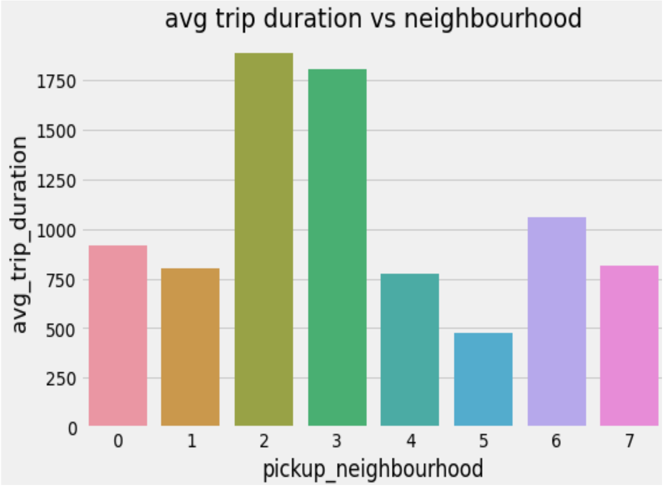
We made 8 clusters using the clustering algorithm which indicates different neighborhood for pickup. The clustering is shown below:



Using the pickup neigborhood we then plot a graph for count of pickups from each neighborhood. In which we found that neighborhood/cluster 0 had the most number of pickups.



Also we found out that average trip duration by passengers from neigborhood/cluster 2 and 3 is higher than any other neighborhood.



VI. Conclusion

With the help of the above data, we can conclude

1. March month is busy month as compared to whole year.
2. Busiest location is the nearby area of the air-port
3. 6pm – 8pm has most pickup and drop-off in whole day and busiest days are Thursday and Friday.
4. Average speed of the taxis remains between 12-13 kmph on busy day/weeldays and upto15 on weekends
5. Most of the pickup has been done from near by area of airport

Refrances

sklearn.ensemble.RandomForestRegressor. (2021). Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

Matplotlib — Visualization with Python. (2021). Matplotlib. https://matplotlib.org

folium. (2021, November 19). PyPI. https://pypi.org/project/folium/