MANIPAL
ACADEMY of HIGHER EDUCATION
(Deemed to be University under Section 3 of the UGC Act, 1956)

MANIPAL SCHOOL OF INFORMATION SCIENCES

(A Constituent unit of MAHE, Manipal)

Uber Data Analysis

Reg. Number	Name	Branch
201046017	Divya G B	BDA
201046018	Bhanuprakash S	BDA

Under the guidance of

Sudhakara Upadya P

Assistant Professor - Selection Grade,

Manipal School of Information Sciences,

MAHE, MANIPAL

10/06/2021

MANIPAL SCHOOL OF INFORMATION SCIENCES
MANIPAL
(A constituent unit of MAHE, Manipal)

Contents

LIST OF FIGURES

1. Objectives

Uber is the fastest growing start up in the world, with over 8 million users, 1 billion Uber trips, and 160,000+ people driving for Uber in 449 cities across 66 countries.

Uber has "eaten the world" in less than 5 years, tackling issues such as poor transportation infrastructure in some cities, unsatisfactory customer experience, late cars, drivers refusing to accept credit cards, and more.

When we look up what uber drivers care about on the internet, we discover that they are often unsure where to go when they don't have passengers, which, of course, wastes a lot of time and money.

So, in this project, we analyse uber data to determine where uber drivers can go to find potential business when they don't have passengers for a certain period of time and visualize data, allowing the company to avail the benefit of understanding the complex data and gain insights that would help them to craft decisions. It will be a huge progress for uber drivers and, of course, uber if they can figure out how to deal with the demand. If the drivers are always aware of what they should do and where they should go, they will generate a lot of value.

2. Project design status

Figure 2.1: Uber Data Analysis Method.

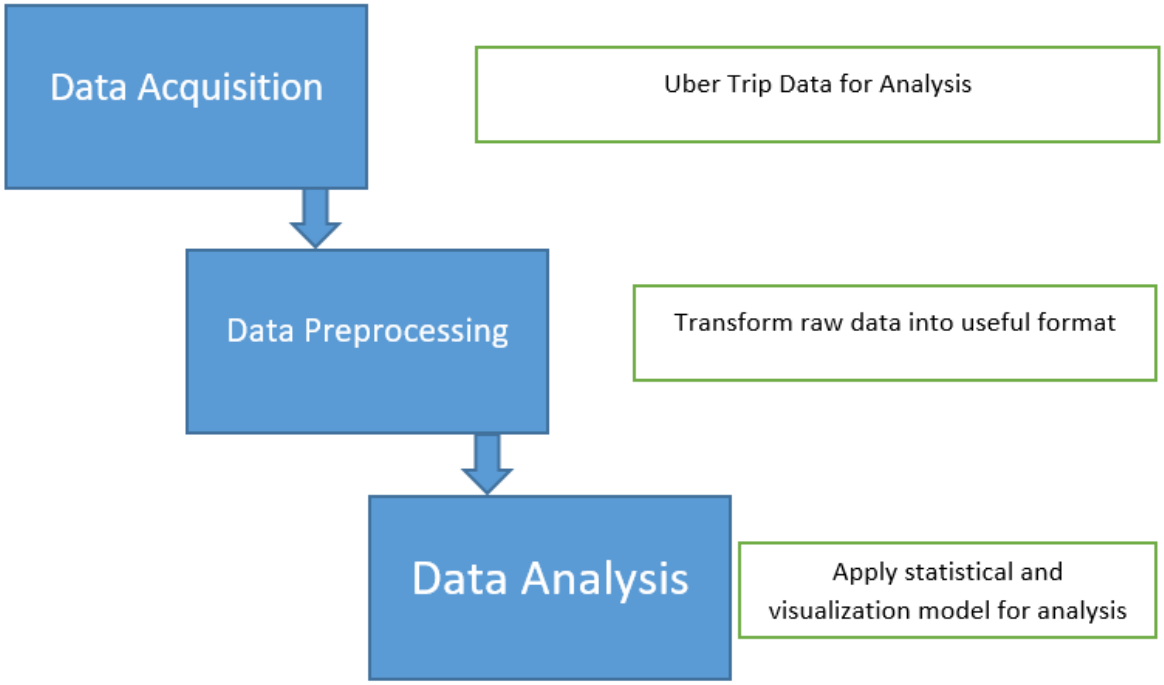

Figure 2.2: Flow Diagram of the Uber data analysis

3. Test environment creation

Programming Language: Python 3

Libraries: Python Standard libraries, Folium, Geopy

Dataset: Uber Data Set, Weather Data Set

IDE: Jupyter Notebook

Operating System: Windows 10 3.2

4. Datasets

Uber Dataset:

The yellow and green taxi trip records include fields capturing pick-up and drop-off dates/times, pick-up and drop-off locations, trip distances, itemized fares, rate types, payment types, and driver-reported passenger counts. The data used in the attached datasets were collected and provided to the NYC Taxi and Limousine Commission (TLC) by technology providers authorized under the Taxicab & Livery Passenger Enhancement Programs.

```
1 uber_data.describe() #yellow_tripdata_2015-07.csv
```

	passenger_count	trip_distance	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	fare_amount	tip_amount	total_amount
count	1.140211e+07	1.140211e+07	1.140211e+07	1.140211e+07	1.140211e+07	1.140211e+07	1.140211e+07	1.140211e+07	1.140211e+07
mean	1.698129e+00	7.344546e+00	-7.397389e+01	4.074993e+01	-7.384298e+01	4.067890e+01	1.300737e+01	1.679025e+00	1.612449e+01
std	1.350153e+00	5.213063e+03	4.009635e-02	3.826368e-02	3.115968e+00	1.716123e+00	7.225877e+01	2.578918e+00	7.297751e+01
min	0.000000e+00	0.000000e+00	-7.987411e+01	9.370500e-02	-7.818333e+02	0.000000e+00	-2.500000e+02	-1.110000e+02	-2.503000e+02
25%	1.000000e+00	1.040000e+00	-7.399232e+01	4.073685e+01	-7.399156e+01	4.073532e+01	6.500000e+00	0.000000e+00	8.760000e+00
50%	1.000000e+00	1.790000e+00	-7.398218e+01	4.075278e+01	-7.398049e+01	4.075337e+01	9.500000e+00	1.150000e+00	1.180000e+01
75%	2.000000e+00	3.290000e+00	-7.396855e+01	4.076661e+01	-7.396443e+01	4.076779e+01	1.500000e+01	2.260000e+00	1.780000e+01
max	9.000000e+00	1.008336e+07	-7.009658e+01	5.311364e+01	0.000000e+00	4.052833e+02	1.861312e+05	8.500000e+02	1.861325e+05

Fig 4.1: Description of 2015 Uber Dataset.

Weather Dataset:

Weather data includes any facts or numbers about the state of the atmosphere, including temperature, wind speed, rain or snow, humidity and pressure. (Dataset: central_park_weather.csv).

```
1 weather_dataset.describe()
```

	DATE	AWND	PRCP	SNOW	SNWD	TMAX	TMIN
count	4.383000e+03	4158.000000	4383.000000	4381.000000	4382.000000	4383.000000	4383.000000
mean	2.014568e+07	5.428793	0.138382	0.094910	0.447239	63.245950	49.120922
std	3.452705e+04	2.470983	0.374848	0.829364	2.103354	18.173531	16.664035
min	2.009010e+07	0.000000	0.000000	0.000000	0.000000	13.000000	1.000000
25%	2.012010e+07	3.580000	0.000000	0.000000	0.000000	48.500000	36.000000
50%	2.015010e+07	4.920000	0.000000	0.000000	0.000000	65.000000	50.000000
75%	2.017567e+07	6.710000	0.060000	0.000000	0.000000	79.000000	64.000000
max	2.020123e+07	22.820000	5.810000	27.300000	23.000000	104.000000	84.000000

Fig 4.2: Description of Central Park Weather Dataset.

The Weather Dataset contains the following fields:

PRCP = Precipitation (tenths of mm)

AWND = Wind Speed (mph)

SNOW = Snowfall (mm)

SNWD = Snow depth (mm)

TMAX = Maximum temperature (tenths of degrees C)

TMIN = Minimum temperature (tenths of degrees C)

5. Result Obtained

- **Loading and Visualising the Uber dataset**

Data Collection/Processing

```
#IMPORTING LIBRARIES
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
#Loading dataset
fields = ['tpep_pickup_datetime', 'tpep_dropoff_datetime', 'passenger_count', 'trip_distance', 'pickup_longitude',
          'pickup_latitude', 'dropoff_longitude', 'dropoff_latitude', 'fare_amount', 'tip_amount', 'total_amount']
uber_data = pd.read_csv('yellow_tripdata_2015-07.csv', usecols = fields)
uber_data = uber_data[(uber_data['pickup_longitude'] < -70) & (uber_data['pickup_longitude'] > -80)]
```

Fig 5.1: Loading the uber dataset(csv file) using pandas library.

```
sns.pairplot(uber_data[0:1000], diag_kind='kde')
```

```
<seaborn.axisgrid.PairGrid at 0x20fb7ab02e8>
```

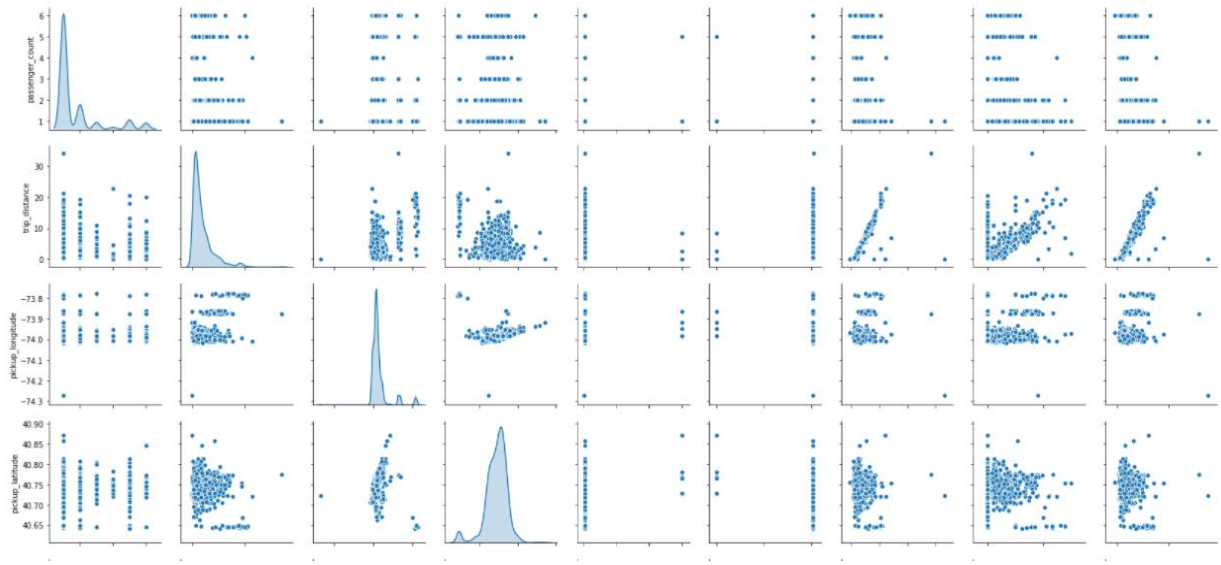

Fig 5.2: Visualising the uber dataset by plotting pair plots.

- **Exploring the Uber Dataset**

The spatial distribution of pickup locations for 5000 Uber trips is plotted in the image below.

```
import matplotlib.pyplot as plt

raw_data = pd.read_csv('uber-raw-data-apr14.csv')

lat = raw_data["Lat"].head(5000)
lon = raw_data["Lon"].head(5000)

plt.title ("Pick-up Location Scatter Plot")
plt.scatter(lon, lat)
plt.xlabel("longitude")
plt.ylabel("latitude")
plt.show()
```

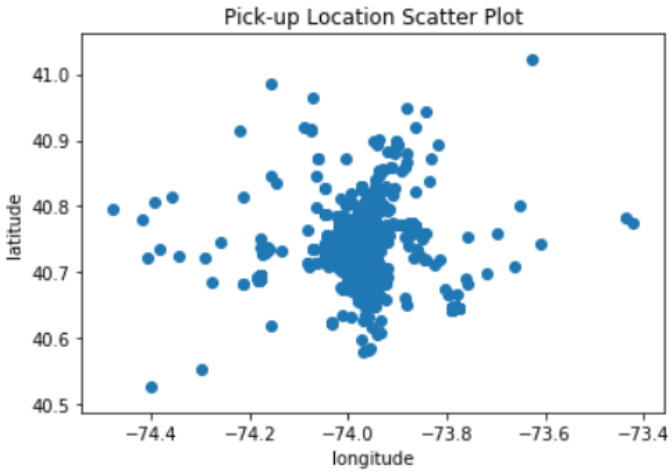

Fig 5.3: Spatial distribution of the pickup locations.

In this project we are working on "yellow_trip_data - July2015" dataset. In this dataset, we have three important attributes: date, pickup location, and drop off location. We collect data on rides related to pickups and then plot a scatter plot of pick-up and drop-off locations on a Google map to get a clear picture of the real situation of the related drives with such a large number of potential customers.

- **Finding the nearby pickups in the dataset in specific time.**

```
import numpy as np
from geopy.geocoders import Nominatim
geolocator = Nominatim(user_agent="http")

location = geolocator.geocode("Broadway Theatre")
print((location.latitude, location.longitude))

pickups = uber_data[(uber_data['pickup_longitude'] < location.longitude + 0.002) & (uber_data['pickup_longitude'] > location
                    & (uber_data['pickup_latitude'] < location.latitude + 0.001) & (uber_data['pickup_latitude'] > location.
                    & uber_data['tpep_pickup_datetime'].str.contains(" 22:")]

print(len(pickups.index), "pick-ups near", location.address)
print("between 22:00 to 23:00")

print("average passenger number of the selected pick-ups is", np.mean(pickups['passenger_count'].tolist()))
print("average trip distance of the selected pick-ups is", np.mean(pickups['trip_distance'].tolist()))
print("average tip amount earned in the selected pick-ups is", np.mean(pickups['tip_amount'].tolist()))
print("average total earnings in the selected pick-ups is", np.mean(pickups['total_amount'].tolist()))
```

```
(40.76339815, -73.98335504506701)
3309 pick-ups near Broadway Theatre, 1681, Broadway, Theater District, Manhattan, New York County, New York, 10019, United Stat
es
between 22:00 to 23:00
average passenger number of the selected pick-ups is 1.733756421879722
average trip distance of the selected pick-ups is 2.617878513145966
average tip amount earned in the selected pick-ups is 1.5461589604110002
average total earnings in the selected pick-ups is 15.038646116651558
```

Fig 5.4: Finding the nearby pickups in the dataset.

- **Displaying pickup and drop off location on map.**

```
pickup_location = pickups[['pickup_latitude', 'pickup_longitude']]
dropoff_location = pickups[['dropoff_latitude', 'dropoff_longitude']]

import folium
import pandas as pd

map_kenya = folium.Map(location=[40.765650, -73.984340], zoom_start=15)

for index, loc2 in dropoff_location.iterrows():
    location2 = [loc2['dropoff_latitude'], loc2['dropoff_longitude']]
    folium.Circle(location2, radius=0.5, color="green", fill=False,).add_to(map_kenya)
for index, loc1 in pickup_location.iterrows():
    location1 = [loc1['pickup_latitude'], loc1['pickup_longitude']]
    folium.Circle(location1, radius=0.5, color="red", fill=False,).add_to(map_kenya)


map_kenya
```

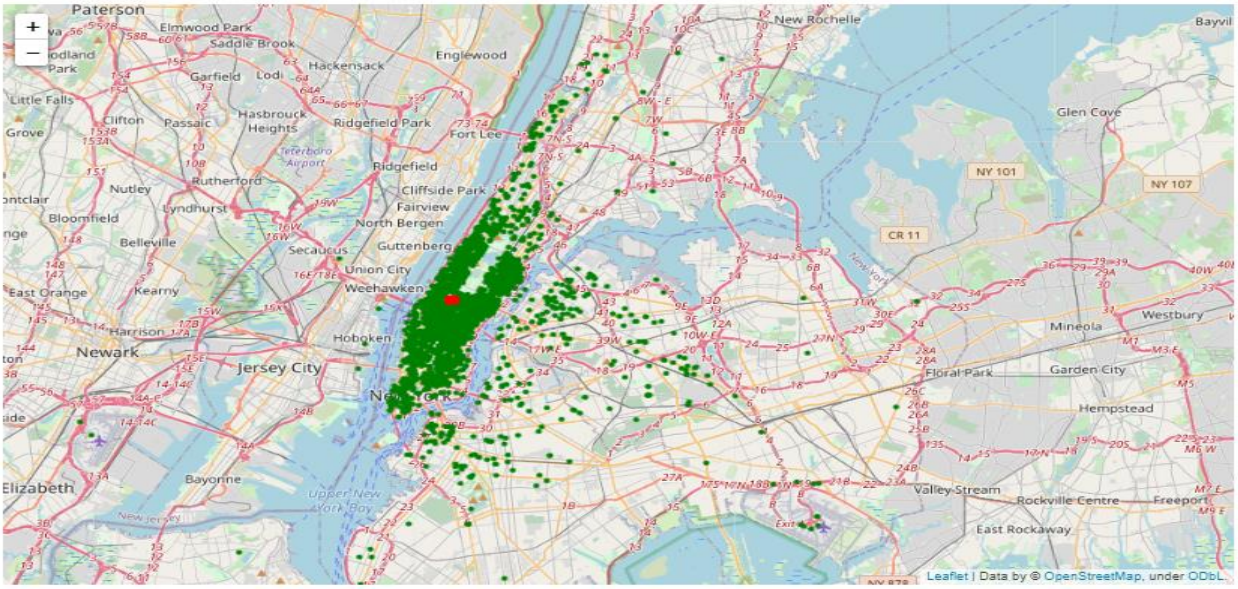

Fig 5.5: Displaying pickup and drop off location on map.

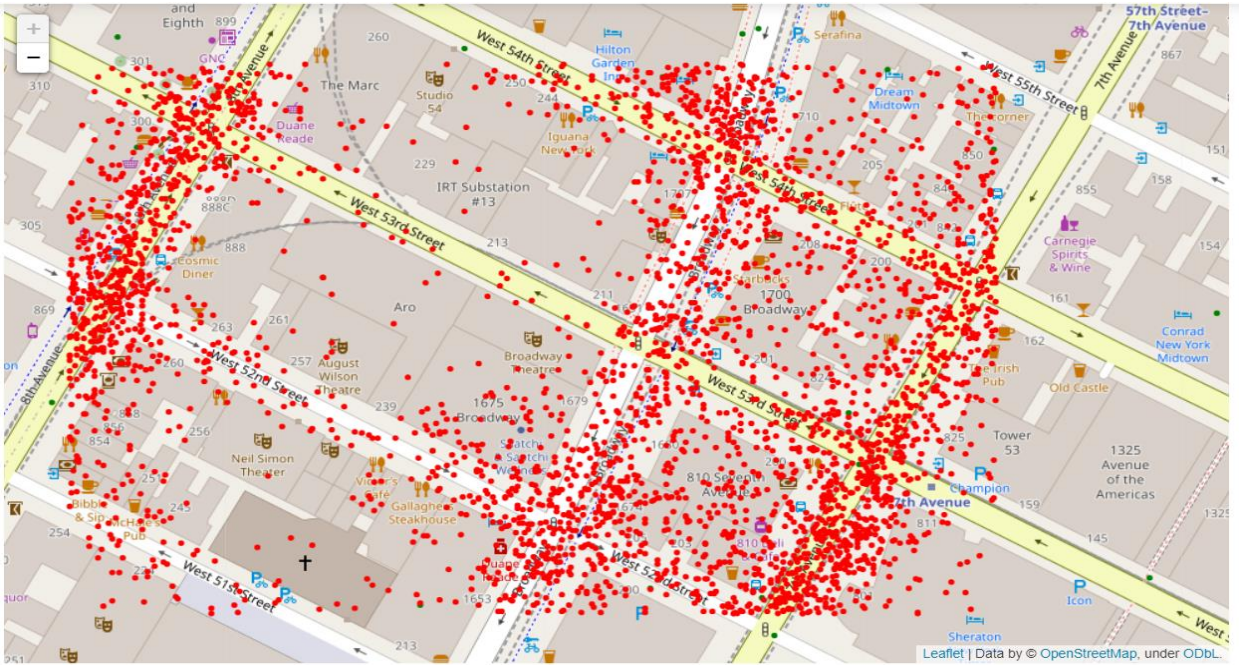

Fig 5.6: Pickup location near Broadway Theatre.

- **Loading and Visualising Weather Dataset**

```
weather_data = pd.read_csv('central_park_weather_new.csv')
```

```
weather_data
```

	STATION	NAME	DATE	AWND	PRCP	SNOW	SNWD	TMAX	TMIN
0	USW00094728	NY CITY CENTRAL PARK, NY US	20090101	11.18	0.00	0.0	0.0	26	15
1	USW00094728	NY CITY CENTRAL PARK, NY US	20090102	6.26	0.00	0.0	0.0	34	23
2	USW00094728	NY CITY CENTRAL PARK, NY US	20090103	10.07	0.00	0.0	0.0	38	29
3	USW00094728	NY CITY CENTRAL PARK, NY US	20090104	7.61	0.00	0.0	0.0	42	25
4	USW00094728	NY CITY CENTRAL PARK, NY US	20090105	6.93	0.00	0.0	0.0	43	38
...	...	...	...	...	...	...	...	...	...
4378	USW00094728	NY CITY CENTRAL PARK, NY US	20201227	4.03	0.00	0.0	0.0	37	24
4379	USW00094728	NY CITY CENTRAL PARK, NY US	20201228	5.37	0.00	0.0	0.0	50	35
4380	USW00094728	NY CITY CENTRAL PARK, NY US	20201229	8.95	0.00	0.0	0.0	43	30
4381	USW00094728	NY CITY CENTRAL PARK, NY US	20201230	5.14	0.00	NaN	NaN	43	28
4382	USW00094728	NY CITY CENTRAL PARK, NY US	20201231	5.14	0.48	0.0	0.0	48	36

4383 rows × 9 columns

Fig 5.7: Loading Weather Dataset.

Visualising weather data

```
sns.pairplot(weather_data, diag_kind='kde', plot_kws={'alpha': 0.2})
```

```
<seaborn.axisgrid.PairGrid at 0x17927e05e48>
```

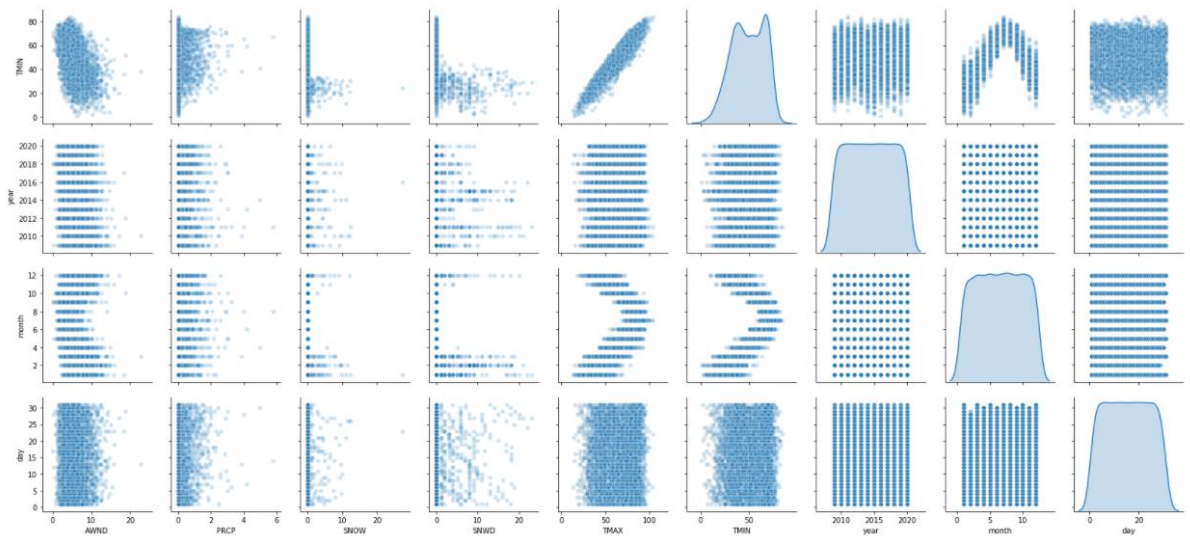

Fig 5.8: Visualising the weather dataset by plotting pair plots.

- **Analysing the Central Park Precipitation using 'DATE' and 'PREC' from the data set.**

```
weather_data = pd.read_csv('central_park_weather_new.csv')
weather_data = weather_data[(weather_data['DATE'] > 20150700) & (weather_data['DATE'] < 20150800)]
weather_data.head()

date = range(0, len(weather_data.index))
precipitation = weather_data["PRCP"]
tmax = weather_data["TMAX"]
tmin = weather_data["TMIN"]

plt.title ("Central Park Precipitation")
plt.plot(date, precipitation, color = "blue")
plt.xlabel("Date")
plt.ylabel("PREC")
plt.show()

plt.title ("Central Park Temperature")
plt.plot(date, tmax, color = "red")
plt.plot(date, tmin, color = "green")
plt.xlabel("Date")
plt.ylabel("Temp")
plt.show()
```

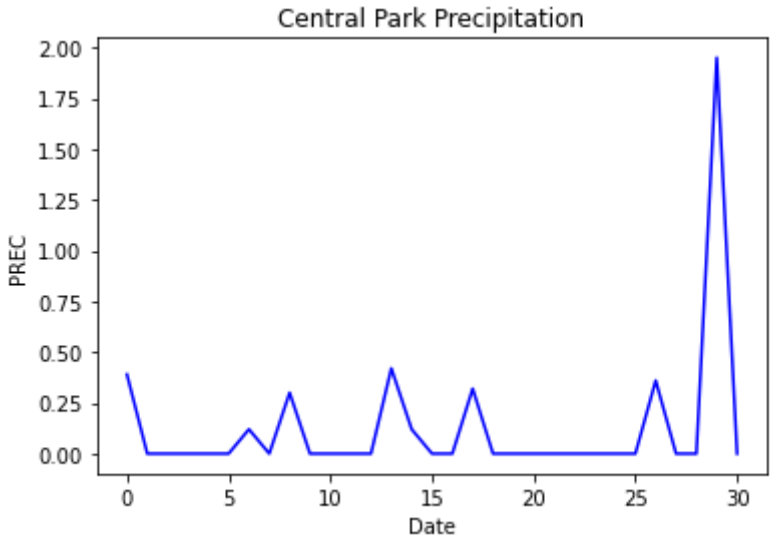

Fig 5.9: Date and Precepitation from 1st July 2015 to 31st July 2015.

In the graph plotted above, we can say that Percipitation in high between 28th July to 30th July.

- **Analysing the Central Park Temperature.**

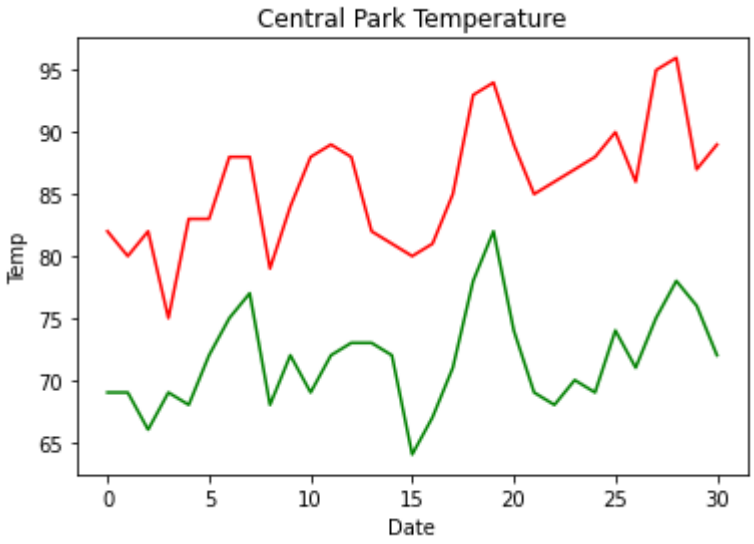

Fig 5.10: Maximum and Minimum Temperature between 1st July to 30th July 2015.

In the graph plotted above, we can visualize the change in Maximum and Minimum Temperatures and analyze the increase in temperature from July 1st to July 30th 2015. We can use these data to compare it with the Uber Dataset to analyze the demand for uber by plotting it on maps.

- **Analysing the Uber Demands.**

```
uber_0720 = uber_data[uber_data['tpep_pickup_datetime'].str.contains("2015-07-20")]
uber_0721 = uber_data[uber_data['tpep_pickup_datetime'].str.contains("2015-07-21")]
uber_0722 = uber_data[uber_data['tpep_pickup_datetime'].str.contains("2015-07-22")]
uber_0723 = uber_data[uber_data['tpep_pickup_datetime'].str.contains("2015-07-23")]
uber_0724 = uber_data[uber_data['tpep_pickup_datetime'].str.contains("2015-07-24")]
uber_0725 = uber_data[uber_data['tpep_pickup_datetime'].str.contains("2015-07-25")]
uber_0726 = uber_data[uber_data['tpep_pickup_datetime'].str.contains("2015-07-26")]
uber_0727 = uber_data[uber_data['tpep_pickup_datetime'].str.contains("2015-07-27")]
uber_0728 = uber_data[uber_data['tpep_pickup_datetime'].str.contains("2015-07-28")]
uber_0729 = uber_data[uber_data['tpep_pickup_datetime'].str.contains("2015-07-29")]

uber_amount = [len(uber_0720.index), len(uber_0721.index), len(uber_0722.index),
               len(uber_0723.index), len(uber_0724.index), len(uber_0725.index),
               len(uber_0726.index), len(uber_0727.index), len(uber_0728.index),
               len(uber_0729.index)]
x = range(20,30)
plt.title("uber demands from 7.20 to 7.29")
plt.plot(x, uber_amount)
plt.xlabel("date")
plt.ylabel("uber demands")
plt.show()
```

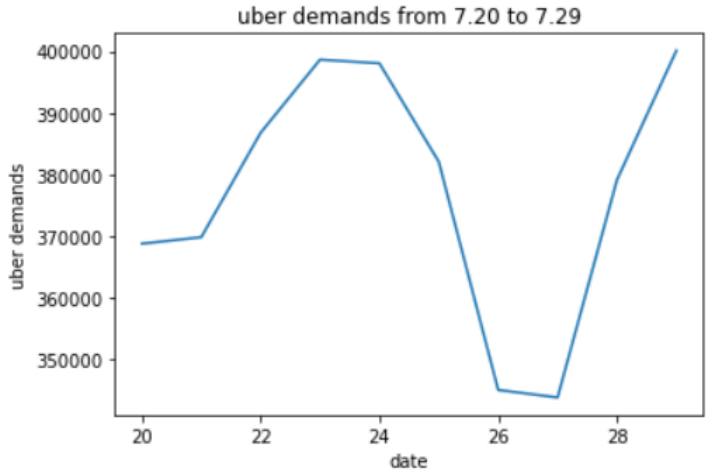

Fig 5.11: Uber Demand from 20th July to 29th July 2015.

The above graph shows the change in Uber Demand (from 20th July to 29th July 2015). By observing the plotted graph, we can analyze that demand for Uber is high between 23rd to 25th of July and 28th to 29th of July.

- **Comparision Uber Demand and Weather Precipitation.**

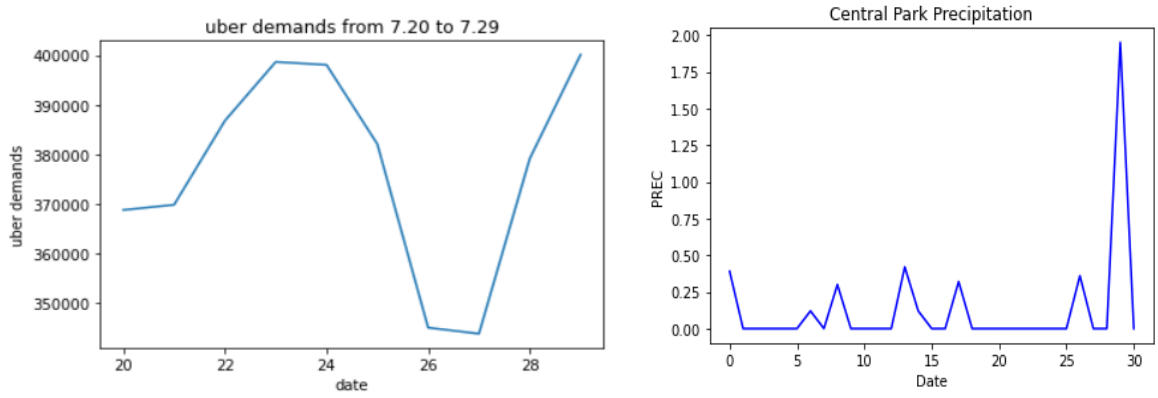

Fig 5.12: Comparision with Uber Demand and Weather Precipitation.

We can see from the above two graphs that precipitation is high from July 28th to July 30th, and uber demand is also high, but Uber Demand is high from July 23rd to July 25th, and whether precipitation is low. We can conclude from this analysis that weather has no impact on Uber Demands.

- **Displaying the Pickup location of 29th July 2015.**

PickUp Location on 29th July 2015

```
import folium
import pandas as pd
m = folium.Map([40.765650, -73.984340], zoom_start=11)

for index, row in pickup.iterrows():
    folium.Circle([row['pickup_latitude'], row['pickup_longitude']], radius=10, fill=True).add_to(m)
m
```

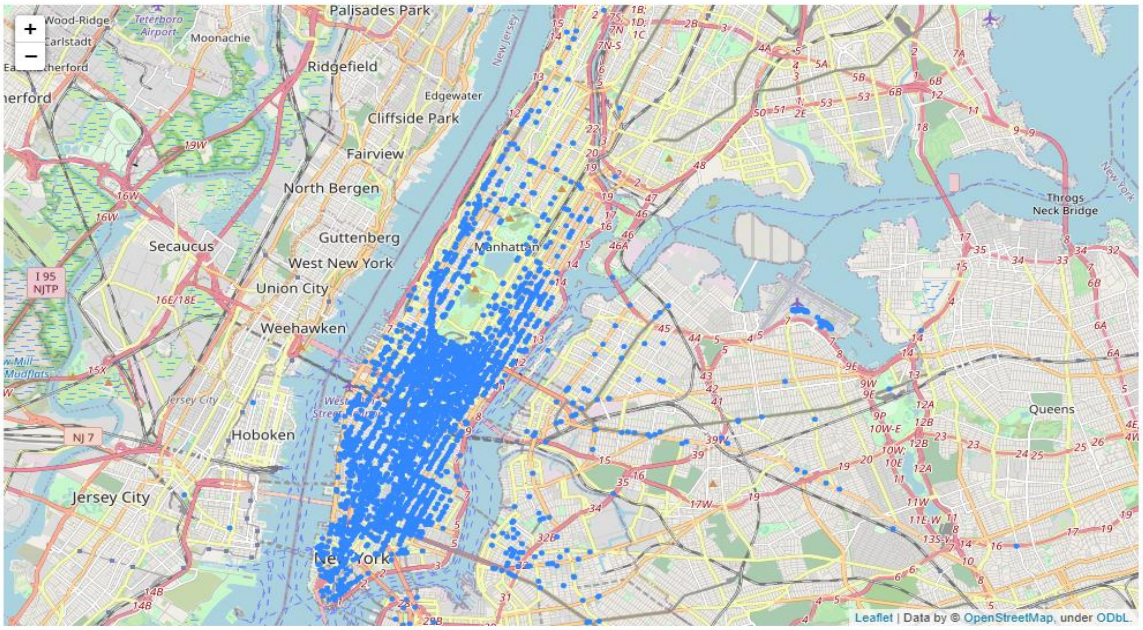

Fig 5.13: Uber Pickup locations of 29th July 2015.

- **Visualising the pickup location by plotting Heatmap.**

Heat Map

```
from folium.plugins import HeatMap
HeatMap(pickup).add_to(m)
m
```

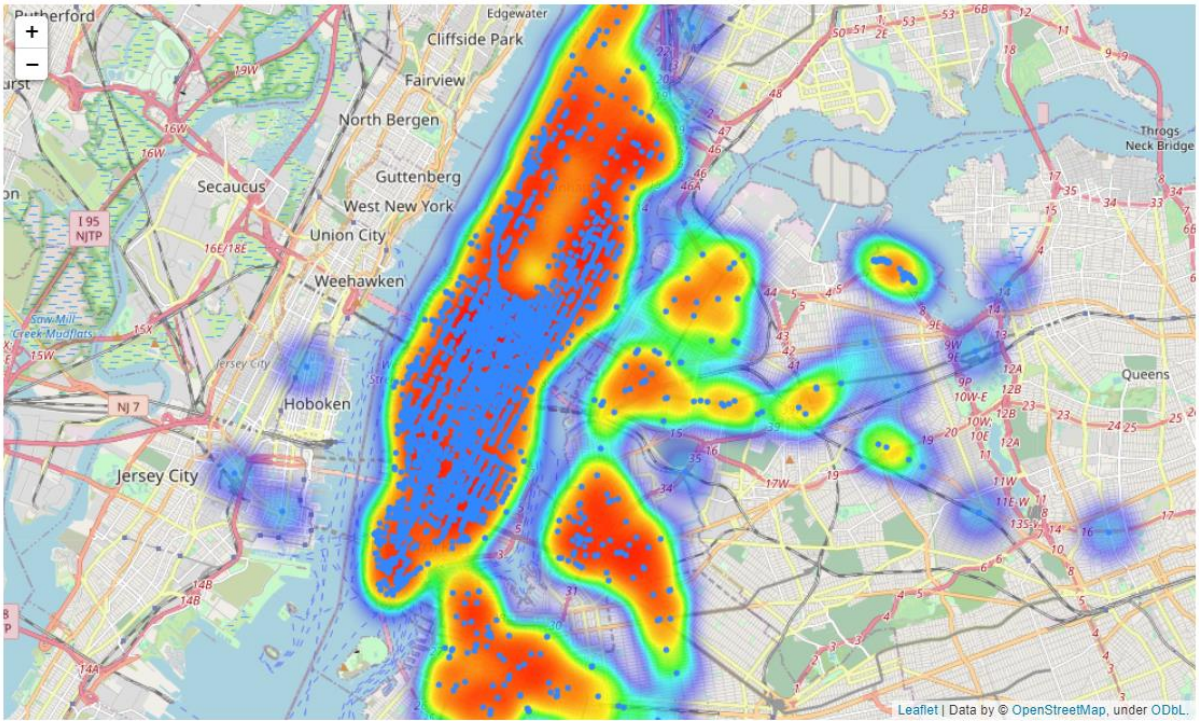

Fig 5.14: Heat Map of Uber Pickup locations.

Heatmap on google map helps us understand the pick-up locations easily. From the above analysis, it seems that different places have generally different types of uber demands. This help drivers determine where to go for potential customers and weather seems to have no much effect on uber demands.

6. Conclusion

This analytics project is very component to understand the use of data analytics. Uber Data Analysis project enables us to understand the complex data visualization, which makes it easier to understand the core values of the databases.

Data science is very interesting and this is one of the projects which prove it. In this project, we created a geo plot using the folium maps of New York, which provided us with the details of how various users travelled from different locations. This helps the driver to be aware of what they should do and where they should go, as well as the average fare, trip distance, tip amount earned, total earnings and time spent on rides. We also compared weather data analysis to Uber demand analysis to see if the weather has an impact on Uber rides/demands.

7. References

- https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page
- https://www.weather.gov/okx/CentralParkHistorical
- https://towardsdatascience.com/exploratory-data-analysis-eda-a-pratical-approach-using-your-uber-rides-dataset-5e9f0e892149
- https://www.kaggle.com/theoddwaffle/uber-data-analysis
- https://www.uber.com/in/en/careers/teams/data-science/