Submitted by-
Name- DIPTO GHOSH
ASTU Roll number- 172010007013
College- Barak Valley Engineering College
Semester- 6th semester
Session- 2017-2021

PROJECT REPORT

***Objective*:**

The main objective of the given project was to analyse a given dataset of Uber rides using machine learning in order to:

- Find traveling time and calculate the average speed of the trip.
- Visualize the data in terms of trips per hour of the day, per day of the week, and per month of the year.
- From the above step find out in which month highest trips are made.

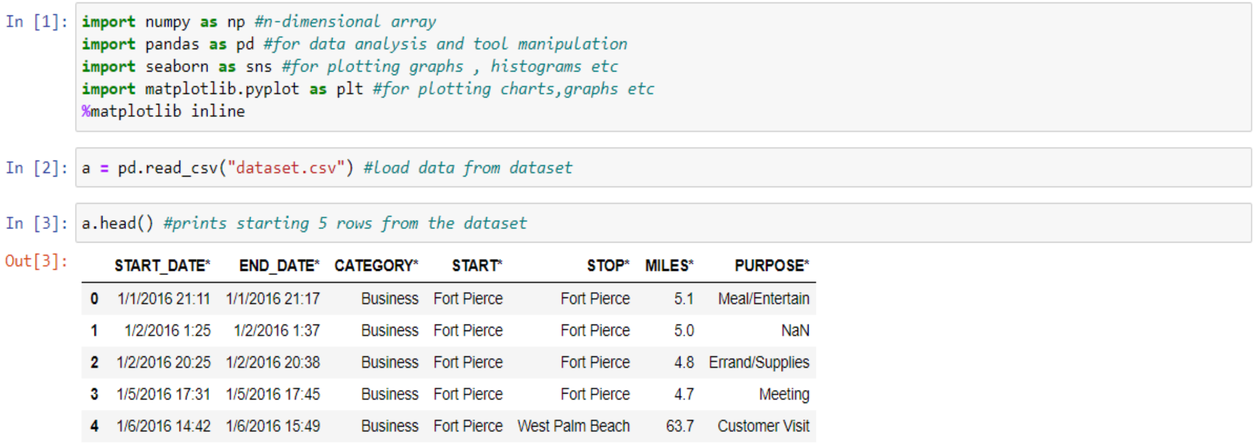
***Strategies implemented*:**

Initially, it was required to search the loaded dataset for columns with null values and drop them for a meaningful analysis. Then, conveniently, the columns with timestamp were converted to find out travelling time, speed and rides based on hourly, weekly, monthly basis resp.

Finally the database was updated and the outcomes were plotted into bar graph.

Execution of the model:

1. Importing the libraries:



```
In [1]: import numpy as np #n-dimensional array
        import pandas as pd #for data analysis and tool manipulation
        import seaborn as sns #for plotting graphs , histograms etc
        import matplotlib.pyplot as plt #for plotting charts,graphs etc
        %matplotlib inline
```

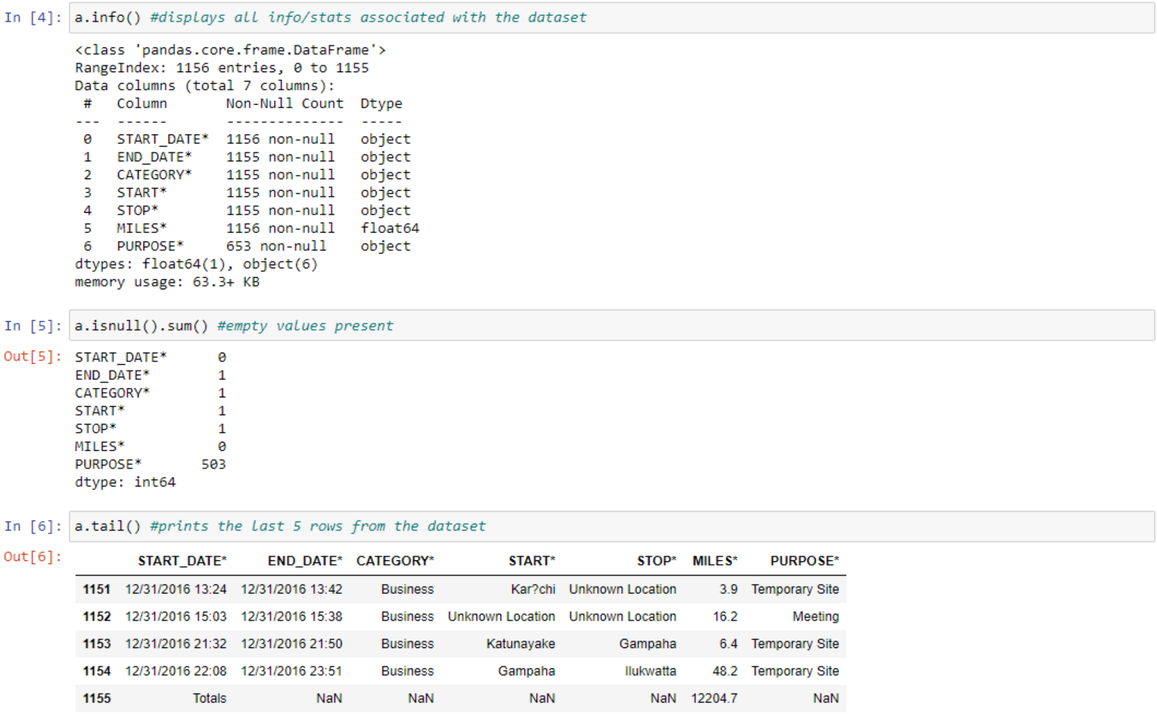
```
In [2]: a = pd.read_csv("dataset.csv") #load data from dataset
```

```
In [3]: a.head() #prints starting 5 rows from the dataset
```

Out[3]:

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
0	1/1/2016 21:11	1/1/2016 21:17	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain
1	1/2/2016 1:25	1/2/2016 1:37	Business	Fort Pierce	Fort Pierce	5.0	NaN
2	1/2/2016 20:25	1/2/2016 20:38	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies
3	1/5/2016 17:31	1/5/2016 17:45	Business	Fort Pierce	Fort Pierce	4.7	Meeting
4	1/6/2016 14:42	1/6/2016 15:49	Business	Fort Pierce	West Palm Beach	63.7	Customer Visit

2. Display the information associated with the dataset:



```
In [4]: a.info() #displays all info/stats associated with the dataset

        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 1156 entries, 0 to 1155
        Data columns (total 7 columns):
         #   Column       Non-Null Count  Dtype
        ---  ------       --------------  -----
         0   START_DATE*  1156 non-null   object
         1   END_DATE*    1155 non-null   object
         2   CATEGORY*    1155 non-null   object
         3   START*       1155 non-null   object
         4   STOP*        1155 non-null   object
         5   MILES*       1156 non-null   float64
         6   PURPOSE*     653 non-null    object
        dtypes: float64(1), object(6)
        memory usage: 63.3+ KB
```

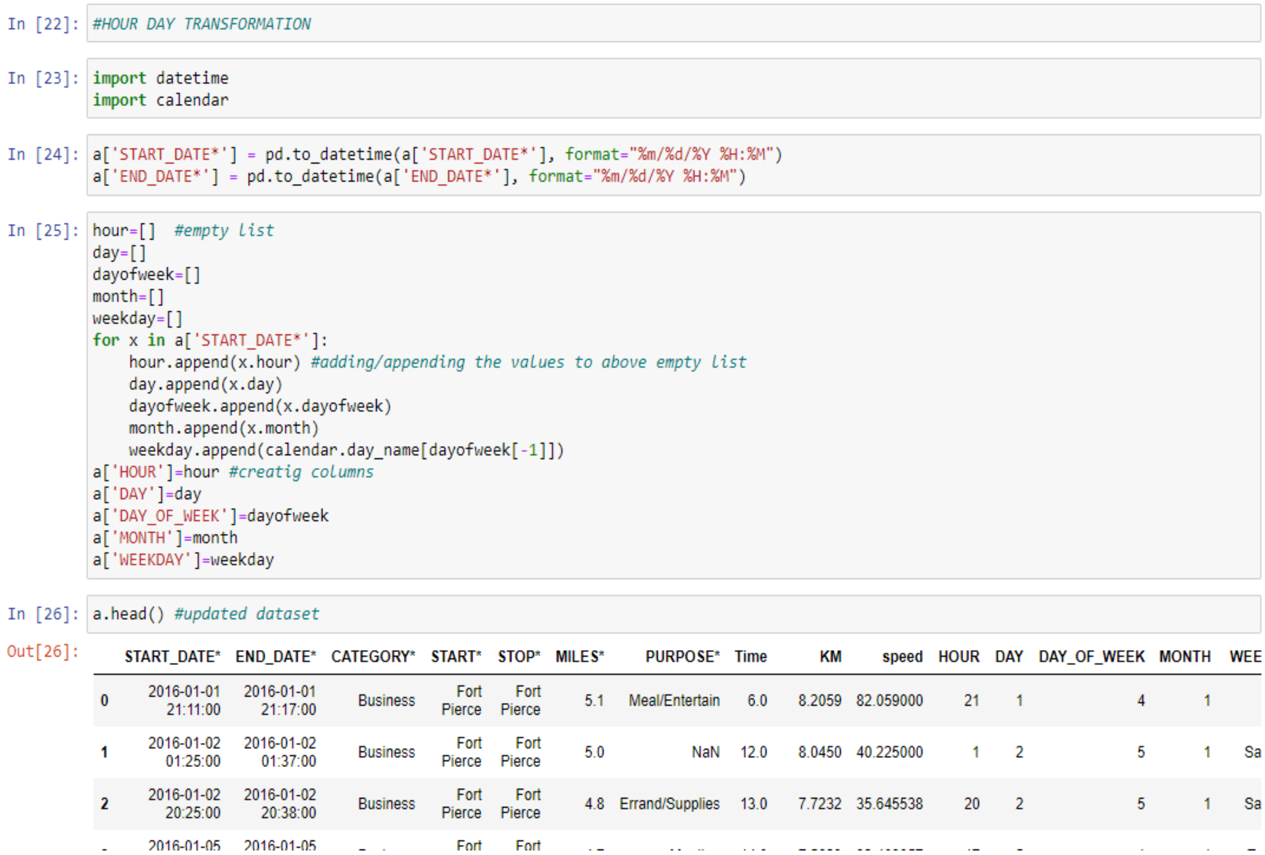
```
In [5]: a.isnull().sum() #empty values present

Out[5]: START_DATE*      0
        END_DATE*        1
        CATEGORY*        1
        START*           1
        STOP*            1
        MILES*           0
        PURPOSE*       503
        dtype: int64
```

```
In [6]: a.tail() #prints the last 5 rows from the dataset
```

Out[6]:

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*
1151	12/31/2016 13:24	12/31/2016 13:42	Business	Kar?chi	Unknown Location	3.9	Temporary Site
1152	12/31/2016 15:03	12/31/2016 15:38	Business	Unknown Location	Unknown Location	16.2	Meeting
1153	12/31/2016 21:32	12/31/2016 21:50	Business	Katunayake	Gampaha	6.4	Temporary Site
1154	12/31/2016 22:08	12/31/2016 23:51	Business	Gampaha	Ilukwatta	48.2	Temporary Site
1155	Totals	NaN	NaN	NaN	NaN	12204.7	NaN

3. Hour-day transformation:



```
In [22]: #HOUR DAY TRANSFORMATION

In [23]: import datetime
         import calendar

In [24]: a['START_DATE*'] = pd.to_datetime(a['START_DATE*'], format="%m/%d/%Y %H:%M")
         a['END_DATE*'] = pd.to_datetime(a['END_DATE*'], format="%m/%d/%Y %H:%M")

In [25]: hour=[]  #empty list
         day=[]
         dayofweek=[]
         month=[]
         weekday=[]
         for x in a['START_DATE*']:
             hour.append(x.hour) #adding/appending the values to above empty list
             day.append(x.day)
             dayofweek.append(x.dayofweek)
             month.append(x.month)
             weekday.append(calendar.day_name[dayofweek[-1]])
         a['HOUR']=hour #creatig columns
         a['DAY']=day
         a['DAY_OF_WEEK']=dayofweek
         a['MONTH']=month
         a['WEEKDAY']=weekday

In [26]: a.head() #updated dataset
```

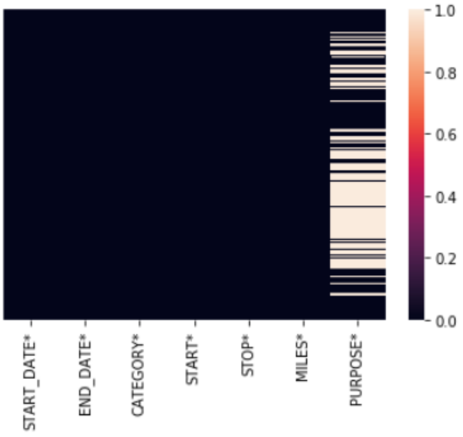
Out[26]:

	START_DATE*	END_DATE*	CATEGORY*	START*	STOP*	MILES*	PURPOSE*	Time	KM	speed	HOUR	DAY	DAY_OF_WEEK	MONTH	WEE
0	2016-01-01 21:11:00	2016-01-01 21:17:00	Business	Fort Pierce	Fort Pierce	5.1	Meal/Entertain	6.0	8.2059	82.059000	21	1	4	1	
1	2016-01-02 01:25:00	2016-01-02 01:37:00	Business	Fort Pierce	Fort Pierce	5.0	NaN	12.0	8.0450	40.225000	1	2	5	1	Sa
2	2016-01-02 20:25:00	2016-01-02 20:38:00	Business	Fort Pierce	Fort Pierce	4.8	Errand/Supplies	13.0	7.7232	35.645538	20	2	5	1	Sa
	2016-01-05	2016-01-05		Fort	Fort										

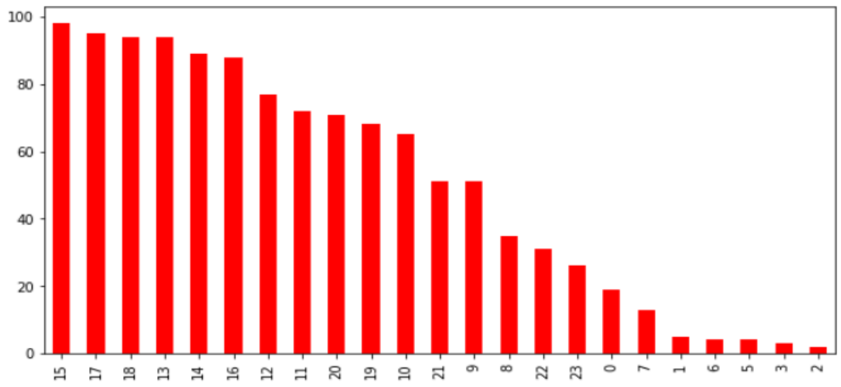
Output of plotted graphs:

Heatmap:

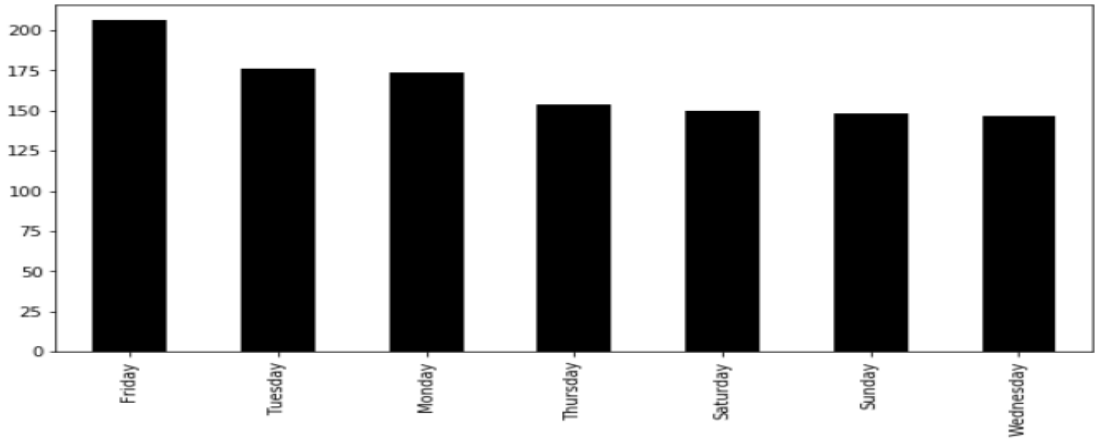
```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x170503e1780>
```



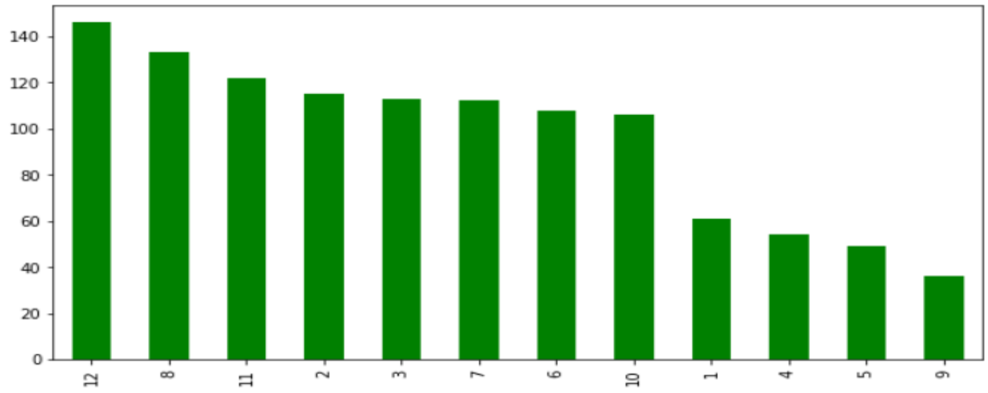
Hour graph:



Weekday graph:



Month of trip graph:



From the above graph we find the highest number of trips take place in December.