PROJECT REPORT

MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

TOPIC: Analyze UBER Data in Python Using Machine Learning.

SUBMITTED BY:-

NAME:ARUNESH DHAR

SEM:7TH

ASTU ROLL :172010007007

COLLEGE: BARAK VALLEY ENGINEERING COLLEGE

CONTENTS

ABSTRACT

In machine learning, a computer first learns to perform a task by studying a training set of examples. The computer then performs the same task with data it hasn’t encountered before. This project presents a brief overview of machine learning technologies, with concrete case from of code analysis.

INTRODUCTION

This project outlines the behaviour of a ordinary uber customer. The project inputs the datasets and undergo exploratory data analysis to calculate the total time taken and the average speed of the trips performed by the uber drivers. It also visualize the data in terms of trips per hour of the day, per day of the week, and per month of the year.

CODE ANALYSIS

1.IMPORTING THE FILES:

```
import numpy as np # linear algebra

import pandas as pd   # data analysis and manipulation of tools

import matplotlib.pyplot as plt   #visualization of datasets

import seaborn as sns   # plotting the histogram ,graphs etc

%matplotlib inline
```

EXPLANATION: The python program begins by importing all the important libraries:

1.NUMPY LIBRARY: We have imported numpy for performing mathematical operations on arrays. Specially here for undergoing linear algebra

2.PANDAS LIBRARY: The pandas library is imported for data analysis via its data frame datastructures.

3.MATPLOTLIB.PYPLOT: This library is used for visualizing the datasets in the form of graph ,histogram etc.

4.SEABORN: This library is imported for plotting histograms , graphs, etc.

5.%MATPLOTLIB: It is a magic function in python. It sets the backend of matplotlib to the inline backend.

2. READING OF DATASETS:

```
datass=pd.read_csv('dataset.csv')
```

Here we have given a csv ie comma separated value files. The datasets is being stored in an variable name datass.

3. EXPLORATORY DATA ANALYSIS :

a. datass.head() : Here we are printing the first few rows of the datasets.

b. uber_xp.tail() : Here we print the last few

c. datass.info(): It gives the information of the datasets.

d.uber_xp.isnull(): it shows the null or the empty values if present. Our datasets consist of empty values mainly in the last row.

e.n_col = datass.select_dtypes(include= np.number).columns

print("Numerical columns are:",n_col):

It prints all the columns having numerical values in the datasets.

f.c_col = datass.select_dtypes(exclude= np.number).columns

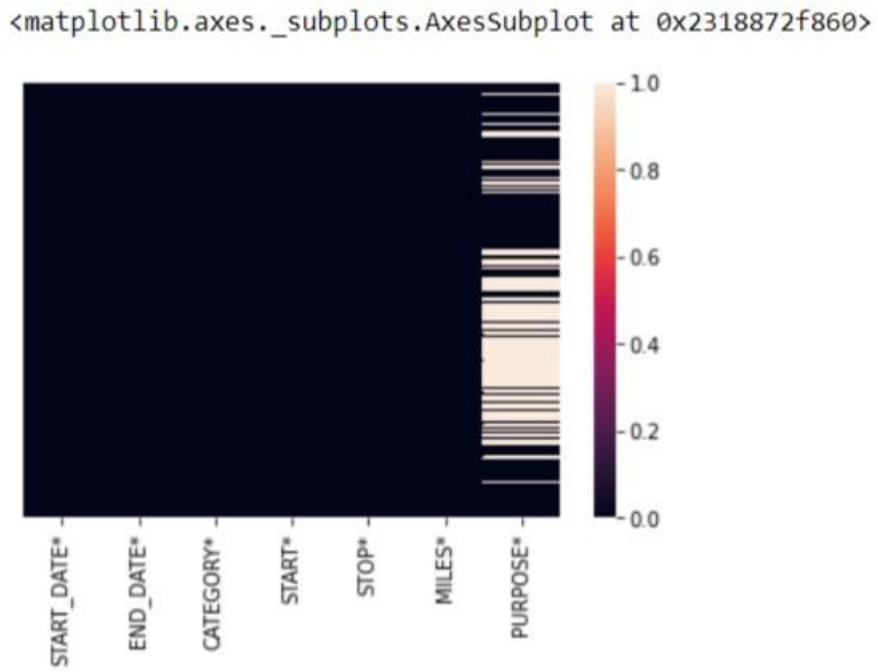
print("categorical column:",c_col): It prints all the column having categorical values.

CLEANING THE DATA SETS

datass .drop(dpp.tail(1).index, inplace=True) #dropping the last row since it h as all null values and is not required

sns.heatmap(datass.isnull(),yticklabels=False) : visualizing the present datasets after cleaning using a seaborn heatmap.

HEATMAP:



TIME REQUIRED FOR THE TRIPS

1. datass= datass ['END_DATE*']- datass ['START_DATE']: we find the time required for the trip in seconds by subtracting the start date from the end date.

2. datass.head(5): printing the first five rows of the new datasets.

3. datass ['time'] datass ['time'].dt.total_seconds()/60.0

datass.head(): coverting into minutes.

4. datass ['speed']=(datass ['MILES*']/ datass ['time'])*60

datass.head(): calculating the speed from time and distance and printing the first few rows of the new datasets.

4. datass.tail(): printing the last few rows.

CALCULATING THE AVERAGE SPEED

1. avg_trp = datass.groupby('speed').mean(): calculate the average speed of each uber driver.

2. print(avg_trp): printing the average value.

TRANSFORMATION

1. datass ['START_DATE*'] = pd.to_datetime(datass ['START_DATE*'], format="%m/%d/%Y %H:%M")

2. datass ['END_DATE*'] = pd.to_datetime(datass['END_DATE*'], format="%m/%d/%Y %H:%M"): it transform to hour,month,year,minutes

TAKING EMPTY SET

3. hour=[]

4. day=[]

5. dayofweek=[]

6. month=[]

7. weekday=[]

appending the values

8. for x in datass['START_DATE*']:

9. hour.append(x.hour)

10. day.append(x.day)

11. dayofweek.append(x.dayofweek)

12. month.append(x.month)

13. weekday.append(calendar.day_name[dayofweek[-1]])

creating columns

14. datass**['HOUR']=hour**

15. datass['DAY']=day

16. datass['DAY_OF_WEEK']=dayofweek

17. datass['MONTH']=month

18. datass['WEEKDAY']=weekday

1. datass.head(): Printing the first few rows .

TRIPS PER HOUR

1. datass ['HOUR'].value_counts().plot(kind='bar',figsize=(10,5),color='green')

plt.xlabel("hours")

plt.ylabel("frequency")

plt.title("trips per hour a day"):

These codes gives a scatter plot of hour vs frequency .

Trips per day of a week

datass ['WEEKDAY'].value_counts().plot(kind='bar',figsize=(10,5),color='yellow')

plt.xlabel('day')

plt.ylabel('frequency')

plt.title('no.of trips per day of the week'):

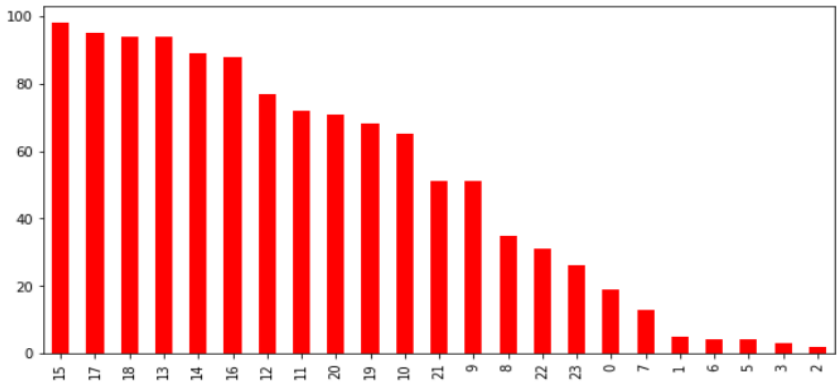
These codes gives the scatter plot of weekday vs frequency. The x axis is plotted with day and y axis is plotted with frequency.

no of trips per month of a year

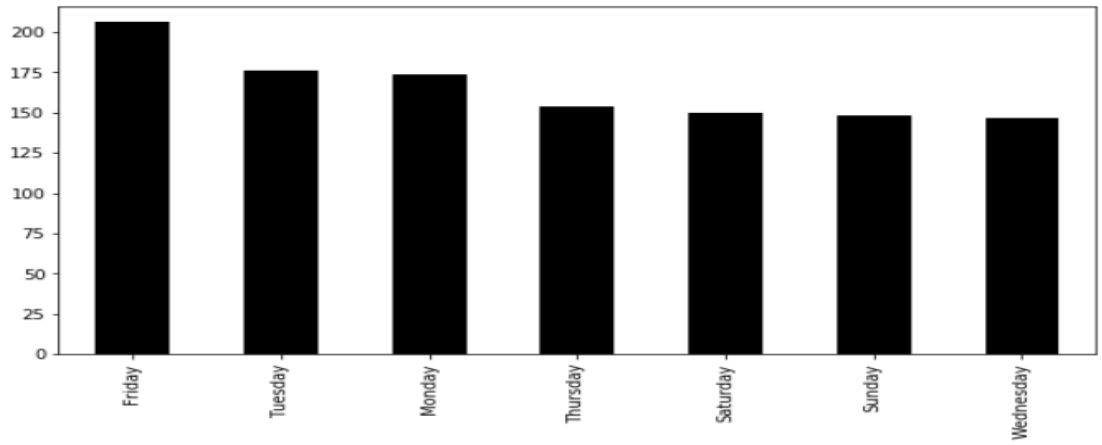
```
datass ['MONTH'].value_counts().plot(kind='bar',figsize=(10,5),color='blue')plt.xlabel('month')plt.ylabel('frequency')plt.title('no. of trips per month of the year')
```

: these codes gives the scatter plot of month vs frequency. The x axis is plotted with month and y axis is plotted with frequency.

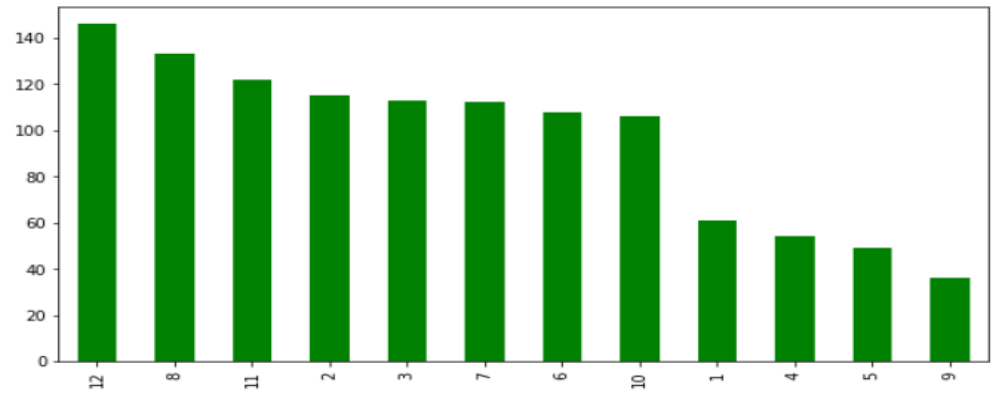
HOUR GRAPH



WEEKLY GRAPH



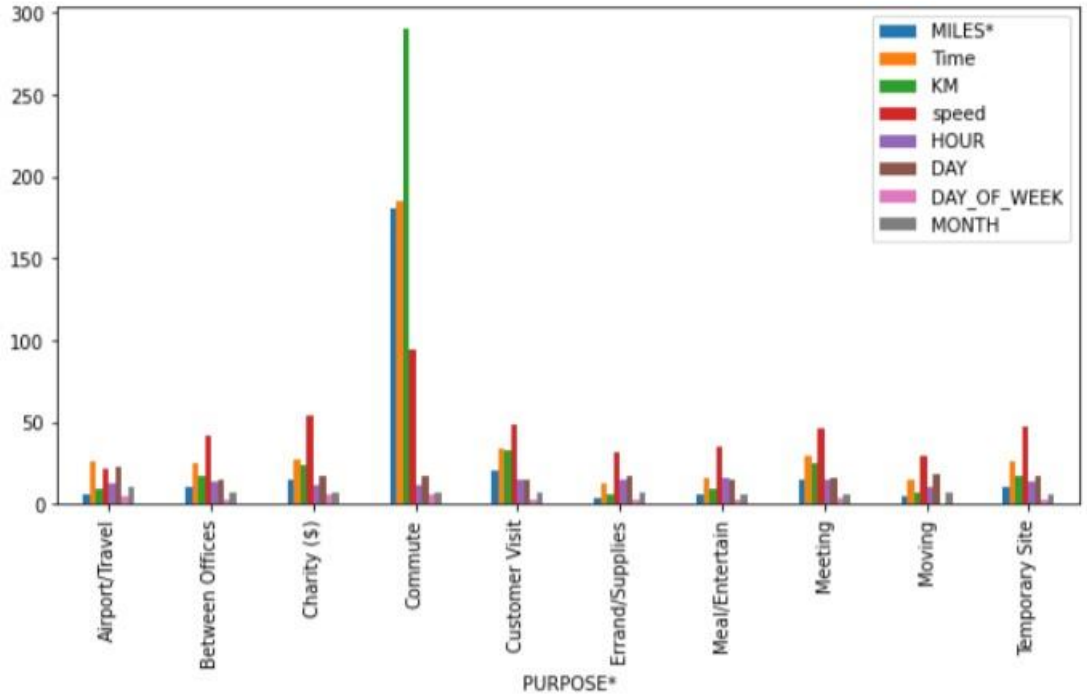
MONTHLY OF TRIP GRAPH



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

```
datass .groupby('PURPOSE*').mean().plot(kind='bar',figsize=(10,5)) #ov
erall analysis
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x23189037ba8>
```



CONCLUSION

These project project help us to understand the notion of working of uber using machine learning. It enable the authority of the company to analyse the trips made by their drivers by calculating the total time taken by them and the frequency of trip per hour of a day. These information helps for the future prediction.

REFERENCES

1. kaggle.com
2. Numpy.org