ASSIGNMENT 3

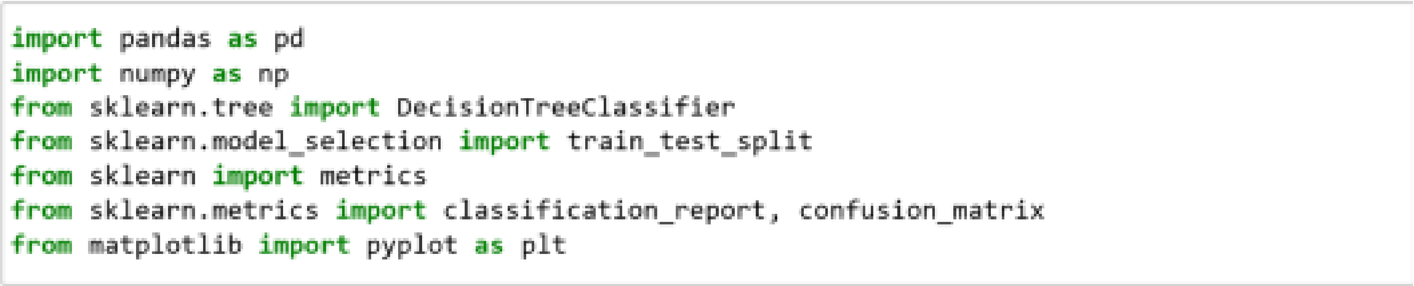
Name : Abhishek Badgujar

Class : BE - A

Roll no : 49

Problem Statement :Trip History Analysis: Use trip history dataset that is from a bike sharing service in the United States. The data is provided quarter-wise from 2010 (Q4) onwards. Each file has 7 columns. Predict the class of user.

In [104]:



```
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import classification_report, confusion_matrix
from matplotlib import pyplot as plt
```

In [105]:

```
df = pd.read_csv(
"202102-capitalbikeshare-tripdata.csv" )
df.head()
```

```
c:\users\admin\appdata\local\programs\python\python38-32\lib\site-packages\I
Python\core\interactiveshell.py:3172: DtypeWarning: Columns (5,7) have mixed
types.Specify dtype option on import or set low_memory=False. has_raised =
await self.run_ast_nodes(code_ast.body, cell_name, Out[105]:
```

ride_id rideable_type started_at ended_at start_station_name start_station_id

0 0F961E4450F8544E classic_bike 2021-02- 2021-02- 21st St & 20 20 Pennsylvania Ave 14:03:25 14:14:17 NW

2021-02- 2021-02- Hains Point/Buckeye 31252

15 15

& Ohio Dr SW 09:54:23 11:21:02

1 DFD528B4F2B3CA6A

2021-02- 2021-02- Hains Point/Buckeye 15 15 & Ohio Dr SW

classic_bike **2** 2398431BB0EB78BE 09:53:12 09:53:34

2021-02- 2021-02- Hains Point/Buckeye 24 24

& Ohio Dr SW 14:50:17 15:29:01

2021-02- 2021-02- 05 05 Hains Point/Buckeye

classic_bike **3** 6E32C58697957443

31273 31273 31273

classic_bike

16:39:10 16:39:13 & Ohio Dr SW 31273

4 2DCACE8B26B0A50A classic_bike

In
[106]:

```
df.dtypes
```

Out[106]:

```
ride_id object
rideable_type object
started_at object
ended_at object
start_station_name object
start_station_id object
end_station_name object
end_station_id object
start_lat float64
start_lng float64
end_lat float64
end_lng float64
member_casual object
dtype: object In [107]:
```

```
print(df.isnull().sum())
```

```
ride_id 0
rideable_type 0
started_at 0 ended_at
0 start_station_name 8295
start_station_id 8295
end_station_name 9312
end_station_id 9312
start_lat 2
start_lng 2

end_lat 106
end_lng 106
member_casual 0
```

Out[109]:

ride_id started_at ended_at start_station_name start_station_id end_station_n

In

New York A

0 0F961E4450F8544E 20

15th S

Hains Point/Buckeye
2021-02- 2021-02- 21st St &

20 Pennsylvania Ave 31252

14:03:25 14:14:17 NW

2021-02- 2021-02- H

1 DFD528B4F2B3CA6A & Ohio Dr SW

15 15 31273 Point/Bucke

09:54:23 11:21:02 Ohio D

Hains Point/Buckeye

2 2398431BB0EB78BE & Ohio Dr SW

Hains Point/Buckeye

3 6E32C58697957443 24

& Ohio Dr SW

Hains Point/Buckeye

4
2DCACE8B26B0A50
A

& Ohio Dr SW

In [110]:
2021-02- 2021-02- H 15 15 31273 Point/Bucke

09:53:12 09:53:34 Ohio D 2021-02- 2021-02- H

24 31273 Point/Bucke

14:50:17 15:29:01 Ohio D 2021-02- 2021-02- H

05 05 31273 Point/Bucke

16:39:10 16:39:13 Ohio D

```
df["start_station_name"].fillna("Not known", inplace = True)
df["end_station_name"].fillna("Not known", inplace = True)
df["start_station_id"].fillna("0", inplace = True)
df["end_station_id"].fillna("0", inplace = True)
print(df.isnull().sum())
```

```
ride_id 0
started_at 0
ended_at 0
start_station_name 0
start_station_id 0
end_station_name 0
In
end_station_id 0
member_casual 0
```

```
ride_id started_at ended_at start_station_name start_station_id end_stati

     0 0F961E4450F8544E 2021-02- 2021-02- 21st St & 31252 New Y
                                                                   20 20 Pennsylvania Ave 15
                                          14:03:25 14:14:17 NW

                               2021-02- 2021-02-
                                          Hains Point/Buckeye
     1 DFD528B4F2B3CA6A 15 15 31273 Point/B
                                            & Ohio Dr SW
                               09:54:23 11:21:02 Oh

                                    2021-02- 2021-02- Hains Point/Buckeye
     2 2398431BB0EB78BE 15 15 & Ohio Dr SW 31273 Point/B
                               09:53:12 09:53:34 Oh

                               2021-02- 2021-02-
                                          Hains Point/Buckeye
     3 6E32C58697957443 24 24 31273 Point/B
                                            & Ohio Dr SW
                               14:50:17 15:29:01 Oh

                                    2021-02- 2021-02- Hains Point/Buckeye
     4 2DCACE8B26B0A50A 05 05 & Ohio Dr SW 31273 Point/B
                                                        16:39:10 16:39:13 Oh ... ... ... ... ... ...

                               2021-02- 2021-02-
 77500 009F4F7752A11024 11 11 1st & K St NE 31662 20th &
                               12:25:21 12:48:50

                               2021-02-
                                            2021-02- Maryland Ave &
 77501 7A87D690A552427D 09 09 17th St NE 31656 20th &
                               12:39:15 13:05:19

                               2021-02- 2021-02-
                                            Maryland Ave &
 77502 D157EF3275190210 09 09 31656 20th &
                                       12:39:24 13:05:03 17th St NE

                                                                             Sm
                               2021-02- 2021-02- Natio
 77503 D72FC8BD078FDE51 04 04 4th & M St SW 31108 Jeffe
                               14:42:32 14:57:21

                               2021-02- 2021-02-
                                         09 09 M St & Pennsylvania
 77504 726098DBA147C32B 14:59:11 15:08:43 Ave NW 31246 Wisconsi
 77505 rows × 8 columns
```

In

```
ride_id object
started_at object
ended_at object
start_station_name object
start_station_id object
end_station_name object
end_station_id object
member_casual int64 dtype:
object
```

In [113]:

```
from sklearn.preprocessing import LabelEncoder #Create
a list with categorical predictors
cat_var
=['start_station_name','end_station_name','member_casual','ride_id','started_at','e
#Initiate LabelEncoder le
= LabelEncoder()
#A for loop to transform the categorical values to numerical values
for n in cat_var: df[n] = le.fit_transform(df[n]) df.dtypes
```

Out[113]:

```
ride_id int32
started_at int32
ended_at int32
start_station_name int32
start_station_id object
end_station_name int32
end_station_id object
member_casual int32
```

	ride_id	started_at	ended_at	start_station_name	end_station_name
0	4669	47455	47350	110	444
1	67851	36604	36762	338	337
2	10678	36603	36539	338	337
3	33157	58511	58561	338	337

In

4	13755	10278	10159	338	337
...	...	...	...	...	...
77500	201	28421	28426	90	101
77501	36935	23150	23190	395	101
77502	63492	23151	23189	395	101
77503	65292	6190	6168	154	520
77504	34458	23684	23643	387	130

77505 rows × 5 columns

⚠

```
LogisticRegression()
```

⚠

```
In [ ]:
```