```
import numpy as np
import pandas as pd

# Loading the dataset
df = pd.read_csv('D:/ibm/datafile_02.csv')
print(df.columns)
df.head()
```

```
Index(['Port', 'Traffic in Eleventh Plan (MT) (2011-12)Proj.',
       'Traffic in Eleventh Plan (MT) (2011-12) Ach.',
       'Traffic in Eleventh Plan (MT) (2011-12) %',
       'Total Capacity  in Eleventh Plan (MT) (2011-12) Proj.',
       'Total Capacity  in Eleventh Plan (MT) (2011-12) Ach.',
       'Total Capacity  in Eleventh Plan (MT) (2011-12) %'],
      dtype='object')
```

	Port	Traffic in Eleventh Plan (MT) (2011-12)Proj.	Traffic in Eleventh Plan (MT) (2011-12) Ach.	Traffic in Eleventh Plan (MT) (2011-12) %	Total Capacity in Eleventh Plan (MT) (2011-12) Proj.	Total Capacity in Eleventh Plan (MT) (2011-12) Ach.
0	Kolkata	1343	1223	9100	3145	1635
1	Haldia	4450	3101	7000	6340	5070
2	Paradeep	7640	5425	7100	10640	7650
3	Visakhapatnam	8220	6742	8200	10810	7293
4	Ennore	4700	1496	3200	6420	3100

```
   Total Capacity  in Eleventh Plan (MT) (2011-12) %
0                                                5100
1                                                7900
2                                                7100
3                                                6700
4                                                4800
```

```
# Preprocessing the dataset

# Renaming the columns
df.rename(columns = {'Traffic in Eleventh Plan (MT) (2011-
12)Proj.':'Traffic_Projected','Traffic in Eleventh Plan (MT) (2011-12)
Ach.':'Traffic_Achieved', 'Total Capacity  in Eleventh Plan (MT)
(2011-12) Proj.':'Total_Capacity_Projected', 'Total Capacity  in
Eleventh Plan (MT) (2011-12) Ach.':'Total_Capacity_Achieved'}, inplace
= True)
df
```

```
             Port  Traffic_Projected  Traffic_Achieved  \
0         Kolkata               1343              1223
1          Haldia               4450              3101
2        Paradeep               7640              5425
3   Visakhapatnam               8220              6742
4          Ennore               4700              1496
5         Chennai               5750              5571
6       Tuticorin               3172              2810
7          Cochin               3817              2010
8            NMPT               4881              3294
9        Mormugao               4455              3900
10         Mumbai               7105              5618
11           JNPT               6604              6575
12         Kandla               8672              8250

    Traffic in Eleventh Plan (MT) (2011-12) %
Total_Capacity_Projected  \
0                                          9100
3145
1                                          7000
6340
2                                          7100
10640
3                                          8200
10810
4                                          3200
6420
5                                          9700
7230
```

```
6                                          8900
6398
7                                          5300
5475
8                                          6800
6050
9                                          8800
6690
10                                         7900
9191
11                                        10000
9560
12                                         9500
12220

    Total_Capacity_Achieved  Total Capacity  in Eleventh Plan (MT)
(2011-12) %
0                        1635
5100
1                        5070
7900
2                        7650
7100
3                        7293
6700
4                        3100
4800
5                        7972
11000
6                        3334
5200
7                        4098
7400
8                        5097
8400
9                        4190
6200
10                       4453
4800
11                       6400
6600
12                       8691
7100
```

```
# Perparing the Calculations:

Traffic_Percent =
round((df.Traffic_Achieved/df.Traffic_Projected)*100,2)

Traffic_Percent
```

```
0     91.06
1     69.69
2     71.01
3     82.02
4     31.83
5     96.89
6     88.59
7     52.66
8     67.49
9     87.54
10    79.07
11    99.56
12    95.13
dtype: float64
```

```
Total_Percent =
round( (df.Total_Capacity_Achieved/df.Total_Capacity_Projected)*100,2)
Total_Percent
```

```
0      51.99
1      79.97
2      71.90
3      67.47
4      48.29
5     110.26
6      52.11
7      74.85
8      84.25
9      62.63
10     48.45
11     66.95
12     71.12
dtype: float64
```

```
# Replacing the existing columns with newly created columns
df.rename(columns = {'Traffic in Eleventh Plan (MT) (2011-12)
%':'Traffic_Percent','Total Capacity  in Eleventh Plan (MT) (2011-12)
%':'Total_Percent'}, inplace = True)
df.iloc[:,3:4] = Traffic_Percent
df.iloc[:,6:] = Total_Percent
df
```

```
             Port  Traffic_Projected  Traffic_Achieved
Traffic_Percent  \
0         Kolkata               1343              1223
91.06
1          Haldia               4450              3101
69.69
2        Paradeep               7640              5425
71.01
3   Visakhapatnam               8220              6742
```

```
82.02
4          Ennore               4700               1496
31.83
5         Chennai               5750               5571
96.89
6       Tuticorin               3172               2810
88.59
7          Cochin               3817               2010
52.66
8            NMPT               4881               3294
67.49
9        Mormugao               4455               3900
87.54
10         Mumbai               7105               5618
79.07
11           JNPT               6604               6575
99.56
12         Kandla               8672               8250
95.13
```

	Total_Capacity_Projected	Total_Capacity_Achieved	Total_Percent
0	3145	1635	51.99
1	6340	5070	79.97
2	10640	7650	71.90
3	10810	7293	67.47
4	6420	3100	48.29
5	7230	7972	110.26
6	6398	3334	52.11
7	5475	4098	74.85
8	6050	5097	84.25
9	6690	4190	62.63
10	9191	4453	48.45
11	9560	6400	66.95
12	12220	8691	71.12

```
df.shape
```

```
(13, 7)
```

```
# Checking for null values
```

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

```
Port                        0
Traffic_Projected           0
Traffic_Achieved            0
Traffic_Percent             0
Total_Capacity_Projected    0
Total_Capacity_Achieved     0
Total_Percent               0
dtype: int64
```

```
# Summary of Dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13 entries, 0 to 12
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Port                      13 non-null     object
 1   Traffic_Projected         13 non-null     int64
 2   Traffic_Achieved          13 non-null     int64
 3   Traffic_Percent           13 non-null     float64
 4   Total_Capacity_Projected  13 non-null     int64
 5   Total_Capacity_Achieved   13 non-null     int64
 6   Total_Percent             13 non-null     float64
dtypes: float64(2), int64(4), object(1)
memory usage: 856.0+ bytes
```

```
df.describe()
```

```
       Traffic_Projected  Traffic_Achieved  Traffic_Percent  \
count          13.000000         13.000000        13.000000
mean         5446.846154       4308.846154        77.887692
std          2133.280019       2212.894855        19.382398
min          1343.000000       1223.000000        31.830000
25%          4450.000000       2810.000000        69.690000
50%          4881.000000       3900.000000        82.020000
75%          7105.000000       5618.000000        91.060000
max          8672.000000       8250.000000        99.560000

       Total_Capacity_Projected  Total_Capacity_Achieved
Total_Percent
count                 13.000000                13.000000
13.000000
mean                7705.307692              5306.384615
68.480000
std                 2570.242673              2140.254796
17.252637
min                 3145.000000              1635.000000
48.290000
25%                 6340.000000              4098.000000
52.110000
50%                 6690.000000              5070.000000
67.470000
75%                 9560.000000              7293.000000
74.850000
max                12220.000000              8691.000000
110.260000
```

```
cor = df.corr
cor
```

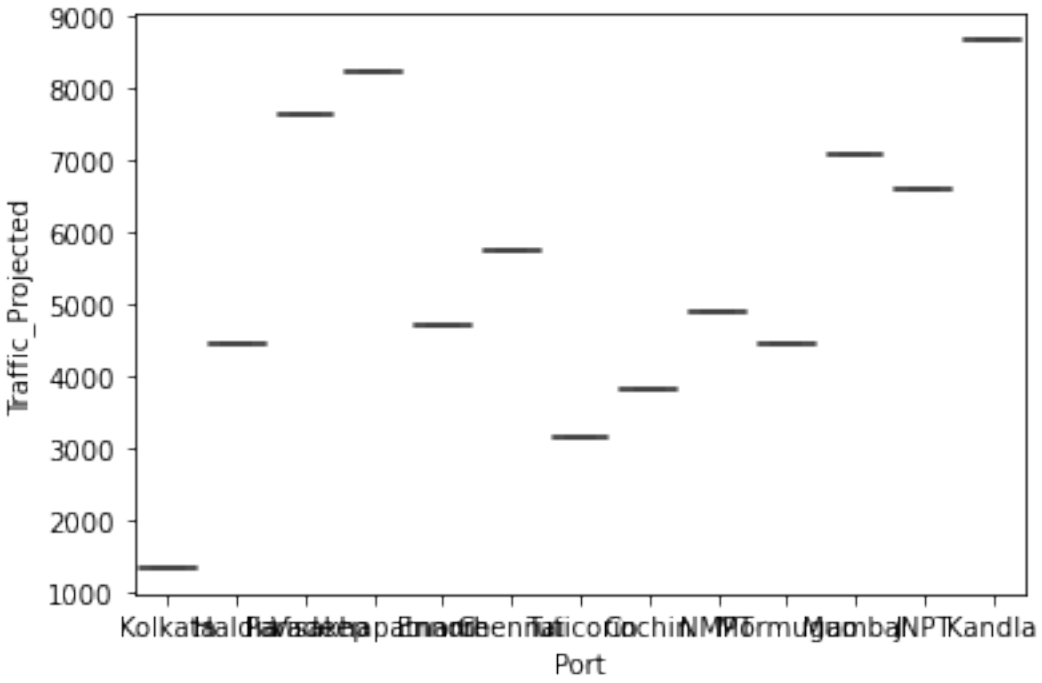
```
<bound method DataFrame.corr of             Port  Traffic_Projected
Traffic_Achieved  Traffic_Percent  \
0         Kolkata               1343              1223
91.06
1          Haldia               4450              3101
69.69
2        Paradeep               7640              5425
71.01
3   Visakhapatnam               8220              6742
82.02
4          Ennore               4700              1496
31.83
5         Chennai               5750              5571
96.89
6       Tuticorin               3172              2810
88.59
7          Cochin               3817              2010
52.66
8            NMPT               4881              3294
67.49
9        Mormugao               4455              3900
87.54
10         Mumbai               7105              5618
79.07
11           JNPT               6604              6575
99.56
12         Kandla               8672              8250
95.13

    Total_Capacity_Projected  Total_Capacity_Achieved  Total_Percent
0                       3145                     1635          51.99
1                       6340                     5070          79.97
2                      10640                     7650          71.90
3                      10810                     7293          67.47
4                       6420                     3100          48.29
5                       7230                     7972         110.26
6                       6398                     3334          52.11
7                       5475                     4098          74.85
8                       6050                     5097          84.25
9                       6690                     4190          62.63
10                      9191                     4453          48.45
11                      9560                     6400          66.95
12                     12220                     8691          71.12
>
```

```
#Finding Outliers anr replacing the outliers
import matplotlib.pyplot as plt
import seaborn as sns

sns.boxplot(x='Port',y='Traffic_Projected',data=df)
```

```
plt.rcParams["figure.figsize"] = [17.50, 3.50]
plt.rcParams["figure.autolayout"] = True
```




```
# Check For Categorical Columns and do encoding


from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
print(df.Port.value_counts())

df.Port = le.fit_transform(df.Port)
print(df.Port.value_counts())
```

```
Kolkata          1
Haldia           1
Paradeep         1
Visakhapatnam    1
Ennore           1
Chennai          1
Tuticorin        1
Cochin           1
NMPT             1
Mormugao         1
Mumbai           1
JNPT             1
Kandla           1
Name: Port, dtype: int64
6     1
```

```
3     1
10    1
12    1
2     1
0     1
11    1
1     1
9     1
7     1
8     1
4     1
5     1
Name: Port, dtype: int64
```

```
# Classification

#y = df.Traffic_Percent
#print(y)

#df.drop(['Traffic_Percent'],axis=1)
```

```
df.head()
```

```
   Port  Traffic_Projected  Traffic_Achieved  Traffic_Percent  \
0     6               1343              1223            91.06
1     3               4450              3101            69.69
2    10               7640              5425            71.01
3    12               8220              6742            82.02
4     2               4700              1496            31.83

   Total_Capacity_Projected  Total_Capacity_Achieved  Total_Percent
0                      3145                     1635          51.99
1                      6340                     5070          79.97
2                     10640                     7650          71.90
3                     10810                     7293          67.47
4                      6420                     3100          48.29
```

```
ddf = df.drop(['Traffic_Percent'],axis=1)
ddf
```

```
    Port  Traffic_Projected  Traffic_Achieved
Total_Capacity_Projected  \
0      6               1343              1223
3145
1      3               4450              3101
6340
2     10               7640              5425
10640
3     12               8220              6742
```

```
10810
4      2               4700              1496
6420
5      0               5750              5571
7230
6     11               3172              2810
6398
7      1               3817              2010
5475
8      9               4881              3294
6050
9      7               4455              3900
6690
10     8               7105              5618
9191
11     4               6604              6575
9560
12     5               8672              8250
12220
```

```
    Total_Capacity_Achieved  Total_Percent
0                      1635          51.99
1                      5070          79.97
2                      7650          71.90
3                      7293          67.47
4                      3100          48.29
5                      7972         110.26
6                      3334          52.11
7                      4098          74.85
8                      5097          84.25
9                      4190          62.63
10                     4453          48.45
11                     6400          66.95
12                     8691          71.12
```

```
x = ddf.iloc[:,1:]
print(x)
```

```
    Traffic_Projected  Traffic_Achieved  Total_Capacity_Projected  \
0                1343              1223                      3145
1                4450              3101                      6340
2                7640              5425                     10640
3                8220              6742                     10810
4                4700              1496                      6420
5                5750              5571                      7230
6                3172              2810                      6398
7                3817              2010                      5475
8                4881              3294                      6050
9                4455              3900                      6690
10               7105              5618                      9191
```

```
11                6604              6575                        9560
12                8672              8250                       12220
```

```
    Total_Capacity_Achieved  Total_Percent
0                      1635          51.99
1                      5070          79.97
2                      7650          71.90
3                      7293          67.47
4                      3100          48.29
5                      7972         110.26
6                      3334          52.11
7                      4098          74.85
8                      5097          84.25
9                      4190          62.63
10                     4453          48.45
11                     6400          66.95
12                     8691          71.12
```

```
y = df.iloc[:,2:3]
print(y)
```

```
    Traffic_Achieved
0               1223
1               3101
2               5425
3               6742
4               1496
5               5571
6               2810
7               2010
8               3294
9               3900
10              5618
11              6575
12              8250
```

```
#1. Logistic Regression
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test =
train_test_split(x,y,test_size=0.2,random_state=0)
print(x_train.shape
     )
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(10, 5)
(3, 5)
(10, 1)
(3, 1)
```

```
from sklearn.linear_model import LinearRegression
mlr=LinearRegression()
mlr.fit(x_train,y_train)
```

```
LinearRegression()
```

```
x_test[0:5]
```

```
    Traffic_Projected  Traffic_Achieved  Total_Capacity_Projected  \
6                3172              2810                      6398
11               6604              6575                      9560
4                4700              1496                      6420

    Total_Capacity_Achieved  Total_Percent
6                      3334          52.11
11                     6400          66.95
4                      3100          48.29
```

```
y_test[0:5]
```

```
    Traffic_Achieved
6               2810
11              6575
4               1496
```

```
mlr.predict(x_test[0:5])
```

```
array([[2810.],
       [6575.],
       [1496.]])
```

```
from sklearn.metrics import r2_score
r2_score(mlr.predict(x_test),y_test)
```

```
1.0
```

```
from sklearn.metrics import mean_squared_error
a = mlr.predict(x_test)
mean_squared_error(a,y_test)
```

```
6.376183888429589e-25
```