In [1]:
```python
import numpy as np
import pandas as pd
```

In [2]:
```python
df = pd.read_csv('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')
```

Out[2]:

	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) %
0	Kolkata	1343	1223	9100	3145	1635	5100
1	Haldia	4450	3101	7000	6340	5070	7900
2	Paradeep	7640	5425	7100	10640	7650	7100
3	Visakhapatnam	8220	6742	8200	10810	7293	6700
4	Ennore	4700	1496	3200	6420	3100	4800

In [3]:
```
# Renaming the columns
df.rename(columns = {'Traffic in Eleventh Plan (MT) (2011-12)Proj.':'Traffic_Proj
df
```

Out[3]:

	Port	Traffic_Projected	Traffic_Achieved	Traffic in Eleventh Plan (MT) (2011-12) %	Total_Capacity_Projected	Total_Cap
0	Kolkata	1343	1223	9100	3145	
1	Haldia	4450	3101	7000	6340	
2	Paradeep	7640	5425	7100	10640	
3	Visakhapatnam	8220	6742	8200	10810	
4	Ennore	4700	1496	3200	6420	
5	Chennai	5750	5571	9700	7230	
6	Tuticorin	3172	2810	8900	6398	
7	Cochin	3817	2010	5300	5475	
8	NMPT	4881	3294	6800	6050	
9	Mormugao	4455	3900	8800	6690	
10	Mumbai	7105	5618	7900	9191	
11	JNPT	6604	6575	10000	9560	
12	Kandla	8672	8250	9500	12220	

In [4]:
```
# Perparing the Calculations:

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

Out[4]:
```
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
```

In [5]:
```
Total_Percent = round( (df.Total_Capacity_Achieved/df.Total_Capacity_Projected)*1
Total_Percent
```

Out[5]:
```
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
```

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

Out[6]:

	Port	Traffic_Projected	Traffic_Achieved	Traffic_Percent%	Total_Capacity_Projected	T
0	Kolkata	1343	1223	91.06	3145	
1	Haldia	4450	3101	69.69	6340	
2	Paradeep	7640	5425	71.01	10640	
3	Visakhapatnam	8220	6742	82.02	10810	
4	Ennore	4700	1496	31.83	6420	
5	Chennai	5750	5571	96.89	7230	
6	Tuticorin	3172	2810	88.59	6398	
7	Cochin	3817	2010	52.66	5475	
8	NMPT	4881	3294	67.49	6050	
9	Mormugao	4455	3900	87.54	6690	
10	Mumbai	7105	5618	79.07	9191	
11	JNPT	6604	6575	99.56	9560	
12	Kandla	8672	8250	95.13	12220	

In [7]:
```
df.shape
```

Out[7]:
```
(13, 7)
```

In [8]:
```
# Checking for null values
df.isnull().sum()
```

Out[8]:
```
Port                        0
Traffic_Projected           0
Traffic_Achieved            0
Traffic_Percent%            0
Total_Capacity_Projected    0
Total_Capacity_Achieved     0
Total_Percent%              0
dtype: int64
```

In [9]:
```
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
```

In [10]:
```
df.describe()
```

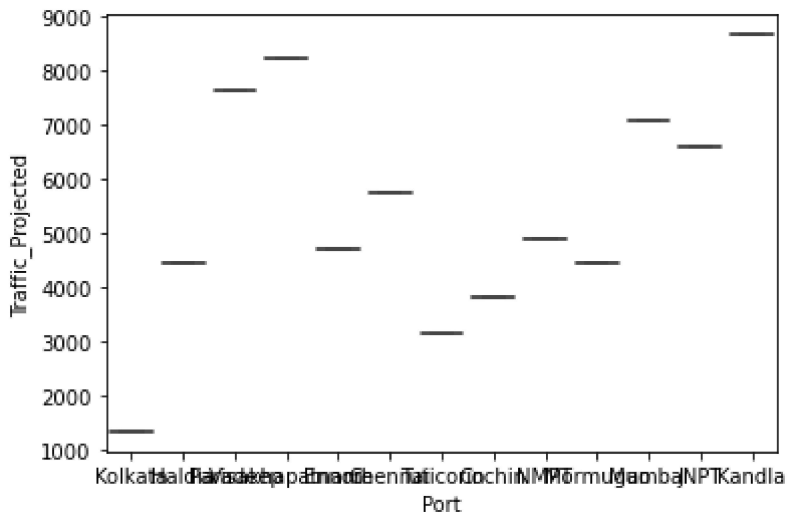
Out[10]:

	Traffic_Projected	Traffic_Achieved	Traffic_Percent%	Total_Capacity_Projected	Total_Capacity
count	13.000000	13.000000	13.000000	13.000000	
mean	5446.846154	4308.846154	77.887692	7705.307692	5
std	2133.280019	2212.894855	19.382398	2570.242673	2
min	1343.000000	1223.000000	31.830000	3145.000000	1
25%	4450.000000	2810.000000	69.690000	6340.000000	4
50%	4881.000000	3900.000000	82.020000	6690.000000	5
75%	7105.000000	5618.000000	91.060000	9560.000000	7
max	8672.000000	8250.000000	99.560000	12220.000000	8

In [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
```



In [13]:
```
# 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())
```

```
Visakhapatnam    1
Kolkata          1
Kandla           1
Mormugao         1
Paradeep         1
Chennai          1
Ennore           1
Tuticorin        1
NMPT             1
Cochin           1
Mumbai           1
Haldia           1
JNPT             1
Name: Port, dtype: int64
0     1
1     1
2     1
3     1
4     1
5     1
6     1
7     1
8     1
9     1
10    1
11    1
12    1
Name: Port, dtype: int64
```

In [14]:
```
# Classification
#y = df.Traffic_Percent
#print(y)
#df.drop(['Traffic_Percent'],axis=1)
df.head()
```

Out[14]:

	Port	Traffic_Projected	Traffic_Achieved	Traffic_Percent%	Total_Capacity_Projected	Total_Capac
0	6	1343	1223	91.06	3145	
1	3	4450	3101	69.69	6340	
2	10	7640	5425	71.01	10640	
3	12	8220	6742	82.02	10810	
4	2	4700	1496	31.83	6420	

In [16]:
```
ddf = df.drop(['Traffic_Percent%'],axis=1)
ddf
```

Out[16]:

	Port	Traffic_Projected	Traffic_Achieved	Total_Capacity_Projected	Total_Capacity_Achieved	Tot
0	6	1343	1223	3145	1635	
1	3	4450	3101	6340	5070	
2	10	7640	5425	10640	7650	
3	12	8220	6742	10810	7293	
4	2	4700	1496	6420	3100	
5	0	5750	5571	7230	7972	
6	11	3172	2810	6398	3334	
7	1	3817	2010	5475	4098	
8	9	4881	3294	6050	5097	
9	7	4455	3900	6690	4190	
10	8	7105	5618	9191	4453	
11	4	6604	6575	9560	6400	
12	5	8672	8250	12220	8691	

In [17]:
```
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
```

In [18]:
```
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
```

In [19]:
```
#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)
```

In [20]:
```
from sklearn.linear_model import LinearRegression
mlr=LinearRegression()
mlr.fit(x_train,y_train)

LinearRegression()

x_test[0:5]
```

Out[20]:

	Traffic_Projected	Traffic_Achieved	Total_Capacity_Projected	Total_Capacity_Achieved	Total_Per
6	3172	2810	6398	3334	
11	6604	6575	9560	6400	
4	4700	1496	6420	3100	

In [21]:
```
y_test[0:5]
```

Out[21]:

	Traffic_Achieved
6	2810
11	6575
4	1496

In [22]:
```
mlr.predict(x_test[0:5])
```

Out[22]:
```
array([[2810.],
       [6575.],
       [1496.]])
```

In [23]:
```
from sklearn.metrics import r2_score
r2_score(mlr.predict(x_test),y_test)
```

Out[23]: 1.0

In [25]:
```
from sklearn.metrics import mean_squared_error
a = mlr.predict(x_test)
mean_squared_error(a,y_test)
```

Out[25]:
```
3.394887097353051e-24
```

In [ ]: