

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [3]: df=pd.read_csv(r"C:\Users\Admin\Downloads\23_Vande Bharat.csv")
df
```

Out[3]:

	Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City	Tr
0	1	New Delhi - Varanasi Vande Bharat Express	22435/22436	Delhi	New Delhi	Varanasi	Va
1	2	New Delhi - Shri Mata Vaishno Devi Katra Vande...	22439/22440	Delhi	New Delhi	Katra	St
2	3	Mumbai Central - Gandhinagar Capital Vande Bha...	20901/20902	Mumbai	Mumbai Central	Gandhinagar	Gan
3	4	New Delhi - Amb Andaura Vande Bharat Express	22447/22448	Delhi	New Delhi	Andaura	
4	5	MGR Chennai Central - Mysuru Vande Bharat Express	20607/20608	Chennai	Chennai Central	Mysuru	I
5	6	Bilaspur - Nagpur Vande Bharat Express	20825/20826	Bilaspur, Chhattisgarh	Bilaspur Junction	Nagpur	I
6	7	Howrah - New Jalpaiguri Vande Bharat Express	22301/22302	Kolkata	Howrah Junction	Siliguri	
7	8	Visakhapatnam - Secunderabad Vande Bharat Express	20833/20834	Visakhapatnam	Visakhapatnam Junction	Hyderabad	
8	9	Mumbai CSMT - Solapur Vande Bharat Express	22225/22226	Mumbai	Chhatrapati Shivaji Terminus	Solapur	
9	10	Mumbai CSMT - Sainagar Shirdi Vande Bharat Exp...	22223/22224	Mumbai	Chhatrapati Shivaji Terminus	Shirdi	
10	11	Rani Kamalapati (Habibganj) - Hazrat Nizamuddi...	20171/20172	Bhopal	Habibganj (Rani Kamalapati)	Delhi	Ha
11	12	Secunderabad - Tirupati Vande Bharat Express	20701/20702	Hyderabad	Secunderabad Junction	Tirupati	
12	13	MGR Chennai Central - Coimbatore Vande Bharat ...	20643/20644	Chennai	Chennai Central	Coimbatore	Coim
13	14	Delhi Cantonment - Ajmer Vande Bharat Express	20977/20978	Delhi	Delhi Cantonment	Ajmer	
14	15	Kasaragod - Thiruvananthapuram Vande Bharat Ex...	20633/20634	Kasaragod	Kasaragod	Thiruvananthapuram	Thiru
15	16	Howrah - Puri Vande Bharat Express	22895/22896	Kolkata	Howrah Junction	Puri	

Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City	Terminal Station
16	Anand Vihar Terminal - Dehradun Vande Bharat Express	22457/22458	Delhi	Anand Vihar Terminal	Dehradun	Dehradun
17	New Jalpaiguri - Guwahati Vande Bharat Express	22227/22228	Siliguri	New Jalpaiguri Junction	Guwahati	Guwahati
18	Mumbai CSMT - Madgaon Vande Bharat Express	22229/22230	Mumbai	Chhatrapati Shivaji Terminus	Madgaon	Madgaon
19	Mumbai CSMT - Madgaon Vande Bharat Express	22229/22230	Mumbai	Chhatrapati Shivaji Terminus	Madgaon	Madgaon
20	Patna - Ranchi Vande Bharat Express	22349/22350	Patna	Patna Junction	Ranchi	Ranchi
21	KSR Bengaluru - Dharwad Vande Bharat Express	20661/20662	Bangalore	Bangalore City	Hubballi - Dharwad	Hubballi - Dharwad
22	Rani Kamalapati (Habibganj) - Jabalpur Vande Bharat Express	20173/20174	Bhopal	Habibganj (Rani Kamalapati)	Jabalpur	Jabalpur
23	Indore - Bhopal Vande Bharat Express	20911/20912	Indore	Indore Junction	Bhopal	Bhopal
24	Jodhpur - Sabarmati (Ahmedabad) Vande Bharat Express	12461/12462	Jodhpur	Jodhpur Junction	Ahmedabad	Ahmedabad
25	Gorakhpur - Lucknow Charbagh Vande Bharat Express	22549/22550	Gorakhpur	Gorakhpur Junction	Charbagh	Charbagh

In [13]: `df.columns`

Out[13]: `Index(['Sr. No.', 'Train Name', 'Train Number', 'Originating City', 'Originating Station', 'Terminal City', 'Terminal Station', 'Operator', 'No. of Cars', 'Frequency', 'Distance', 'Travel Time', 'Speed', 'Average Speed', 'Inauguration', 'Average occupancy'], dtype='object')`

```
In [14]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26 entries, 0 to 25
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Sr. No.                26 non-null    int64
1   Train Name            26 non-null    object
2   Train Number          26 non-null    object
3   Originating City      26 non-null    object
4   Originating Station   26 non-null    object
5   Terminal City         26 non-null    object
6   Terminal Station      26 non-null    object
7   Operator              26 non-null    object
8   No. of Cars           26 non-null    int64
9   Frequency             26 non-null    object
10  Distance              26 non-null    object
11  Travel Time           26 non-null    object
12  Speed                26 non-null    object
13  Average Speed         26 non-null    object
14  Inauguration          26 non-null    object
15  Average occupancy     26 non-null    object
dtypes: int64(2), object(14)
memory usage: 3.4+ KB
```

```
In [16]: x=df[['No. of Cars']]
        y=df['Sr. No.']
```

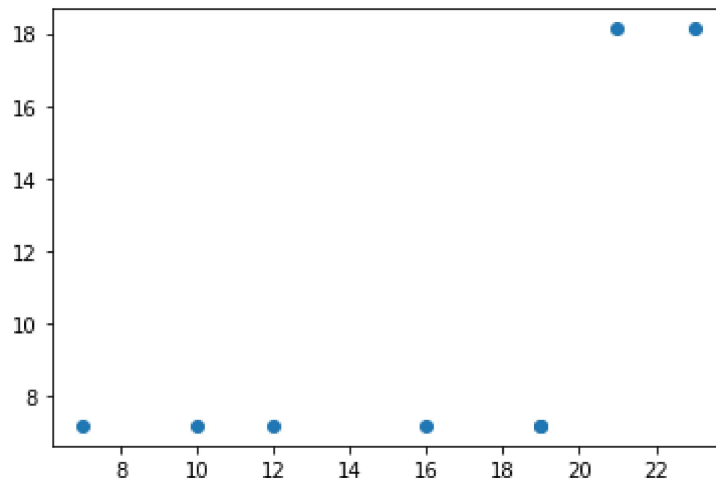
```
In [17]: from sklearn.model_selection import train_test_split
        x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

```
In [18]: from sklearn.linear_model import LinearRegression
        lr= LinearRegression()
        lr.fit(x_train,y_train)
```

```
Out[18]: LinearRegression()
```

```
In [19]: prediction= lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[19]: <matplotlib.collections.PathCollection at 0x270c75077c0>



```
In [20]: print('Linear Regression(score):',lr.score(x_test,y_test))
print('Linear Regression(train score)',lr.score(x_train,y_train))
```

Linear Regression(score): -0.8626848249027228
Linear Regression(train score) 0.5155472166729659

```
In [21]: from sklearn.linear_model import Ridge,Lasso
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
print('Ridge(test score):',rr.score(x_test,y_test))
```

Ridge(test score): -0.81962401658609

```
In [22]: la=Lasso(alpha=10)
la.fit(x_train,y_train)
print('Lasso (test score)',la.score(x_test,y_test))
```

Lasso (test score) -0.46552668148971765

```
In [23]: from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
print(en.score(x_test,y_test))
```

-0.796296769970479

```
In [24]: import pickle
file="predict"
pickle.dump(lr,open(file,'wb'))
```

In []:

