

# Importing libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

# Importing dataset

In [2]:

```
data=pd.read_csv(r"C:\Users\user\Downloads\vande bharat.csv")
data
```

Out[2]:

Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City	Termin...
0 1	New Delhi - Varanasi Vande Bharat Express	22435/22436	Delhi	New Delhi	Varanasi	Varanas
1 2	New Delhi - Shri Mata Vaishno Devi Katra Vande...	22439/22440	Delhi	New Delhi	Katra	Shri Mat...
2 3	Mumbai Central - Gandhinagar Capital Vande Bha...	20901/20902	Mumbai	Mumbai Central	Gandhinagar	Gandhinag...
3 4	New Delhi - Amb Andaura Vande Bharat Express	22447/22448	Delhi	New Delhi	Andaura	Amk...
4 5	MGR Chennai Central - Mysuru Vande Bharat Express	20607/20608	Chennai	Chennai Central	Mysuru	Mysore
5 6	Bilaspur - Nagpur Vande Bharat Express	20825/20826	Bilaspur, Chhattisgarh	Bilaspur Junction	Nagpur	Nagpu...
6 7	Howrah - New Jalpaiguri Vande Bharat Express	22301/22302	Kolkata	Howrah Junction	Siliguri	New...
7 8	Visakhapatnam - Secunderabad Vande Bharat Express	20833/20834	Visakhapatnam	Visakhapatnam Junction	Hyderabad	Secu...
8 9	Mumbai CSMT - Solapur Vande Bharat Express	22225/22226	Mumbai	Chhatrapati Shivaji Terminus	Solapur	
9 10	Mumbai CSMT - Sainagar Shirdi	22223/22224	Mumbai	Chhatrapati Shivaji	Shirdi	Saina...

Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City	Termination
	Vande Bharat Express			Terminus		
10 11	Rani Kamalapati (Habibganj) - Hazrat Nizamuddin	20171/20172	Bhopal	Habibganj (Rani Kamalapati)	Delhi	Hazrat Nizamuddin
11 12	Secunderabad - Tirupati Vande Bharat Express	20701/20702	Hyderabad	Secunderabad Junction	Tirupati	
12 13	MGR Chennai Central - Coimbatore Vande Bharat Express	20643/20644	Chennai	Chennai Central	Coimbatore	Coimbatore
13 14	Delhi Cantonment - Ajmer Vande Bharat Express	20977/20978	Delhi	Delhi Cantonment	Ajmer	Ajmer
14 15	Kasaragod - Thiruvananthapuram Vande Bharat Express	20633/20634	Kasaragod	Kasaragod	Thiruvananthapuram	Thiruvananthapuram
15 16	Howrah - Puri Vande Bharat Express	22895/22896	Kolkata	Howrah Junction	Puri	
16 17	Anand Vihar Terminal - Dehradun Vande Bharat Express	22457/22458	Delhi	Anand Vihar Terminal	Dehradun	Dehradur
17 18	New Jalpaiguri - Guwahati Vande Bharat Express	22227/22228	Siliguri	New Jalpaiguri Junction	Guwahati	
18 19	Mumbai CSMT - Madgaon Vande Bharat Express	22229/22230	Mumbai	Chhatrapati Shivaji Terminus	Madgaon	Madgaon
19 19	Mumbai CSMT - Madgaon Vande Bharat Express	22229/22230	Mumbai	Chhatrapati Shivaji Terminus	Madgaon	Madgaon
20 20	Patna - Ranchi Vande Bharat Express	22349/22350	Patna	Patna Junction	Ranchi	Ranchi
21 21	KSR Bengaluru - Dharwad Vande Bharat Express	20661/20662	Bangalore	Bangalore City	Hubballi - Dharwad	
22 22	Rani Kamalapati (Habibganj) - Jabalpur Vande Bharat Express	20173/20174	Bhopal	Habibganj (Rani Kamalapati)	Jabalpur	Jabalpur
23 23	Indore - Bhopal Vande Bharat Express	20911/20912	Indore	Indore Junction	Bhopal	Bhopal

Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City	Termin...
24 24	Jodhpur - Sabarmati (Ahmedabad) Vande Bharat E...	12461/12462	Jodhpur	Jodhpur Junction	Ahmedabad	Sabarmat
25 25	Gorakhpur - Lucknow Charbagh Vande Bharat Express	22549/22550	Gorakhpur	Gorakhpur Junction	Charbagh	Lucknow

## info

In [3]:

```
# to identify missing values
data.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
```

## describe

In [4]:

```
# to display summary of the dataset
data.describe()
```

Out[4]:

	Sr. No.	No. of Cars
count	26.000000	26.000000
mean	13.230769	12.923077
std	7.306478	3.969112
min	1.000000	8.000000

Sr. No.	No. of Cars
25%	7.250000
50%	13.500000
75%	19.000000
max	25.000000

## columns

```
In [5]: # to display headings of the dataset
data.columns
```

```
Out[5]: 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 [6]: a=data.dropna(axis=1)
a
```

Out[6]:

Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City	Terminal Station
0 1	New Delhi - Varanasi Vande Bharat Express	22435/22436	Delhi	New Delhi	Varanasi	Varanas
1 2	New Delhi - Shri Mata Vaishno Devi Katra Vande...	22439/22440	Delhi	New Delhi	Katra	Shri Mat
2 3	Mumbai Central - Gandhinagar Capital Vande Bha...	20901/20902	Mumbai	Mumbai Central	Gandhinagar	Gandhinag
3 4	New Delhi - Amb Andaura Vande Bharat Express	22447/22448	Delhi	New Delhi	Andaura	Amk
4 5	MGR Chennai Central - Mysuru Vande Bharat Express	20607/20608	Chennai	Chennai Central	Mysuru	Mysore
5 6	Bilaspur - Nagpur Vande Bharat Express	20825/20826	Bilaspur, Chhattisgarh	Bilaspur Junction	Nagpur	Nagpu
6 7	Howrah - New Jalpaiguri Vande Bharat Express	22301/22302	Kolkata	Howrah Junction	Siliguri	New
7 8	Visakhapatnam - Secunderabad	20833/20834	Visakhapatnam	Visakhapatnam Junction	Hyderabad	Secu

Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City	Termin...
Vande Bharat Express						
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	Saina
10 11	Rani Kamalapati (Habibganj) - Hazrat Nizamuddi...	20171/20172	Bhopal	Habibganj (Rani Kamalapati)	Delhi	Hazrat Ni...
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	Coimbatore
13 14	Delhi Cantonment - Ajmer Vande Bharat Express	20977/20978	Delhi	Delhi Cantonment	Ajmer	Ajme
14 15	Kasaragod - Thiruvananthapuram Vande Bharat Ex...	20633/20634	Kasaragod	Kasaragod	Thiruvananthapuram	Thiruvanan...
15 16	Howrah - Puri Vande Bharat Express	22895/22896	Kolkata	Howrah Junction	Puri	
16 17	Anand Vihar Terminal - Dehradun Vande Bharat E...	22457/22458	Delhi	Anand Vihar Terminal	Dehradun	Dehradur
17 18	New Jalpaiguri - Guwahati Vande Bharat Express	22227/22228	Siliguri	New Jalpaiguri Junction	Guwahati	
18 19	Mumbai CSMT - Madgaon Vande Bharat Express	22229/22230	Mumbai	Chhatrapati Shivaji Terminus	Madgaon	Madgaor
19 19	Mumbai CSMT - Madgaon Vande Bharat Express	22229/22230	Mumbai	Chhatrapati Shivaji Terminus	Madgaon	Madgaor
20 20	Patna - Ranchi Vande Bharat Express	22349/22350	Patna	Patna Junction	Ranchi	Ranch
21 21	KSR Bengaluru - Dharwad Vande	20661/20662	Bangalore	Bangalore City	Hubballi - Dharwad	

Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City	Termin...
Bharat Express						
22 22	Rani Kamalapati (Habibganj) - Jabalpur Vande B...	20173/20174	Bhopal	Habibganj (Rani Kamalapati)	Jabalpur	Jabalpu...
23 23	Indore - Bhopal Vande Bharat Express	20911/20912	Indore	Indore Junction	Bhopal	Bhopa...
24 24	Jodhpur - Sabarmati (Ahmedabad) Vande Bharat E...	12461/12462	Jodhpur	Jodhpur Junction	Ahmedabad	Sabarmat...
25 25	Gorakhpur - Lucknow Charbagh Vande Bharat Express	22549/22550	Gorakhpur	Gorakhpur Junction	Charbagh	Lucknow

In [7]: `a.columns`

Out[7]: `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')`

## To train the model-Model Building

In [8]: `x=a[['Sr. No.']]  
y=a['No. of Cars']`

In [9]: `# to split my dataset into training and test data  
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)`

## Linear regression

In [10]: `from sklearn.linear_model import LinearRegression  
lr= LinearRegression()  
lr.fit(x_train,y_train)`

Out[10]: `LinearRegression()`

In [11]: `print(lr.intercept_)`

16.76155268022181

In [12]:

```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

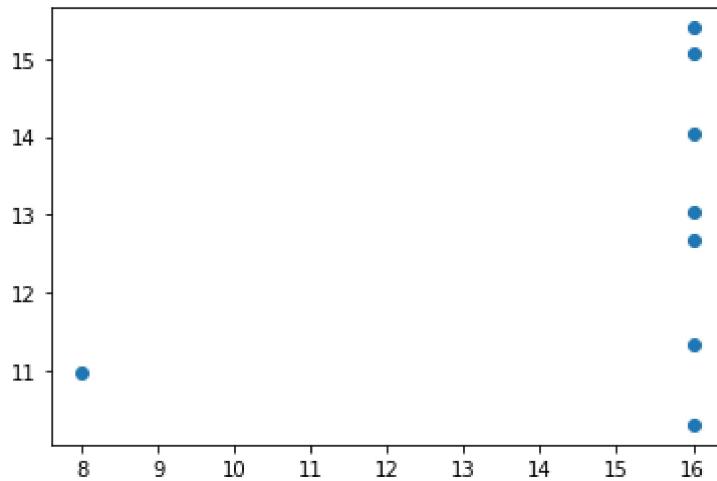
Out[12]:

<b>Co-efficient</b>	
<b>Sr. No.</b>	-0.340111

In [13]:

```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[13]: <matplotlib.collections.PathCollection at 0x19dce4da5e0>



In [14]:

```
print(lr.score(x_test,y_test))
```

-0.5760347565145274

In [15]:

```
lr.score(x_train,y_train)
```

Out[15]: 0.4345861573218319

## Ridge regression

In [16]:

```
from sklearn.linear_model import Ridge,Lasso
```

In [17]:

```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
rr.score(x_test,y_test)
```

Out[17]: -0.5775105759407175

In [18]:

```
rr.score(x_train,y_train)
```

```
Out[18]: 0.4345497129379914
```

## Lasso regression

```
In [19]: la=Lasso(alpha=10)
la.fit(x_train,y_train)
la.score(x_train,y_train)
```

```
Out[19]: 0.33061203532552885
```

```
In [20]: la.score(x_test,y_test)
```

```
Out[20]: -0.7848286310742023
```

## Elastic net regression

```
In [21]: from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

```
Out[21]: ElasticNet()
```

```
In [22]: print(en.coef_)
```

```
[-0.32905591]
```

```
In [23]: print(en.intercept_)
```

```
16.606782768102658
```

```
In [24]: predict=en.predict(x_test)
```

```
In [25]: print(en.score(x_test,y_test))
```

```
-0.5816931944161561
```

```
In [26]: from sklearn import metrics
```

```
In [27]: print("Mean Absolute error:",metrics.mean_absolute_error(y_test,predict))
```

```
Mean Absolute error: 2.930568285976168
```

```
In [28]: print("Mean Squared error:",metrics.mean_squared_error(y_test,predict))
```

```
Mean Squared error: 11.071852360913093
```

```
In [29]: print("Root squared error:",np.sqrt(metrics.mean_squared_error(y_test,predict)))
```

```
Root squared error: 3.3274393098767545
```

## Model saving

```
In [30]: import pickle  
filename="prediction"  
pickle.dump(lr,open(filename,'wb'))  
filename='prediction'  
model=pickle.load(open(filename,'rb'))
```

```
In [31]: real=[[10],[7]]  
result=model.predict(real)  
result
```

```
Out[31]: array([13.36044362, 14.38077634])
```

```
In [ ]:
```