

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
In [2]: df = pd.read_csv("Bharat.csv")
df
```

Out[2]:

	Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City	Termini
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 Vande Bharat Exp...	22223/22224	Mumbai	Chhatrapati Shivaji Terminus	Shirdi	Saina
10	11	Rani Kamalapati (Habibganj) - Hazrat	20171/20172	Bhopal	Habibganj (Rani	Delhi	Hazrat Ni:

Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City	Termini
	Nizamuddi...			Kamalapati)		
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 Dehradun
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 Ranch
21	21	KSR Bengaluru - Dharwad Vande Bharat Express	20661/20662	Bangalore	Bangalore City	Hubbali - Dharwad
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

Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City	Terminal Station
25	Gorakhpur - Lucknow Charbagh Vande Bharat Express	22549/22550	Gorakhpur	Gorakhpur Junction	Charbagh	Lucknow

In [3]:

```
df.head()
```

Out[3]:

Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City	Terminal Station	Operator	No. of Cars
0	New Delhi - Varanasi Vande Bharat Express	22435/22436	Delhi	New Delhi	Varanasi	Varanasi Junction	NR	16
1	New Delhi - Shri Mata Vaishno Devi Katra Vande...	22439/22440	Delhi	New Delhi	Katra	Shri Mata Vaishno Devi Katra	NR	16
2	Mumbai Central - Gandhinagar Capital Vande Bha...	20901/20902	Mumbai	Mumbai Central	Gandhinagar	Gandhinagar Capital	WR	16
3	New Delhi - Amb Andaura Vande Bharat Express	22447/22448	Delhi	New Delhi	Andaura	Amb Andaura	NR	16
4	MGR Chennai Central - Mysuru Vande Bharat Express	20607/20608	Chennai	Chennai Central	Mysuru	Mysore Junction	SR	16

## Data cleaning and pre processing

In [4]:

```
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 [5]: `df.describe()`

Out[5]:

	Sr. No.	No. of Cars
<b>count</b>	26.000000	26.000000
<b>mean</b>	13.230769	12.923077
<b>std</b>	7.306478	3.969112
<b>min</b>	1.000000	8.000000
<b>25%</b>	7.250000	8.000000
<b>50%</b>	13.500000	16.000000
<b>75%</b>	19.000000	16.000000
<b>max</b>	25.000000	16.000000

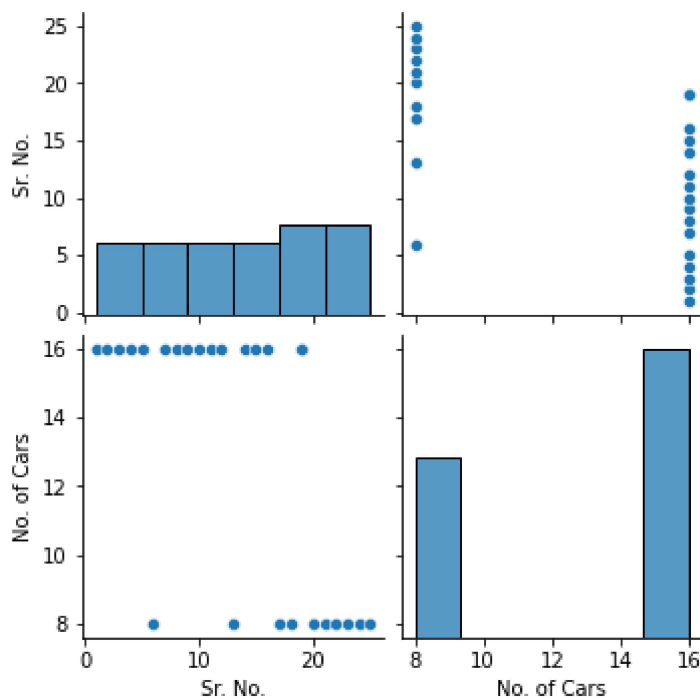
In [6]: `df.columns`

Out[6]: 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')

## EDA and VISUALIZATION

In [7]: `sns.pairplot(df)`

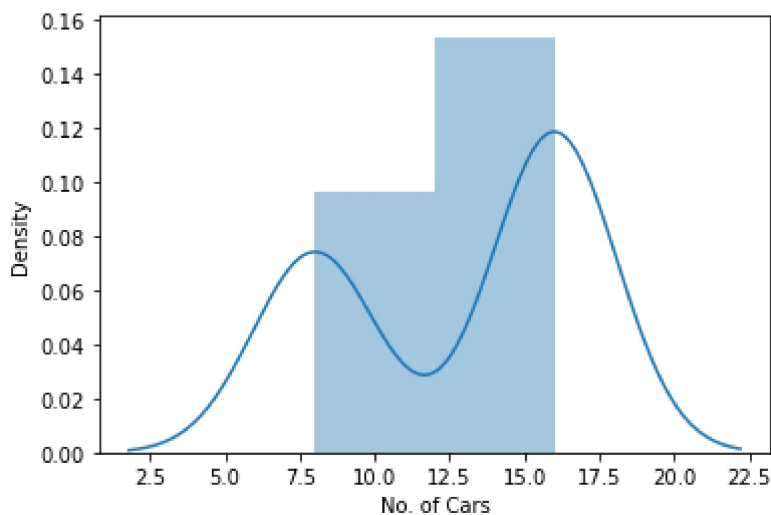
Out[7]: <seaborn.axisgrid.PairGrid at 0x2190deec70>



In [8]: `sns.distplot(df['No. of Cars'])`

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)

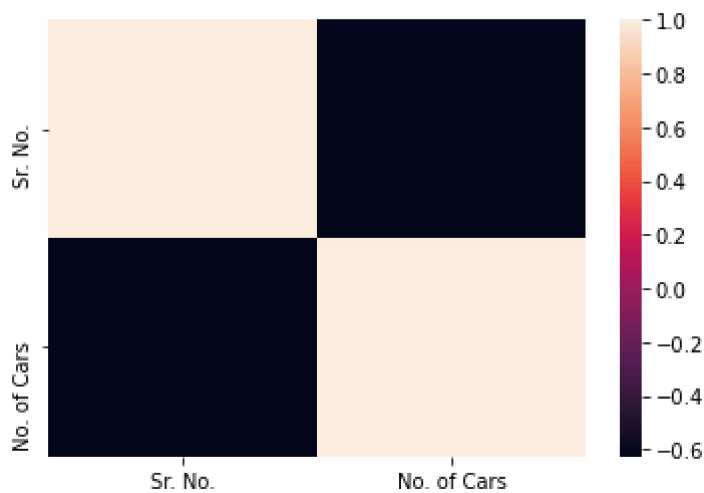
Out[8]: `<AxesSubplot:xlabel='No. of Cars', ylabel='Density'>`



In [9]: `df1 = df[['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']]`

In [10]: `sns.heatmap(df1.corr())`

Out[10]: <AxesSubplot:>



```
In [11]: x = df1[['Sr. No.', 'No. of Cars']]
          y = df1['No. of Cars']
```

**split the data into training and test data**

```
In [12]: x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3)
```

```
In [13]: lr = LinearRegression()
          lr.fit(x_train, y_train)
```

Out[13]: LinearRegression()

```
In [14]: lr.intercept_
```

Out[14]: 1.7763568394002505e-15

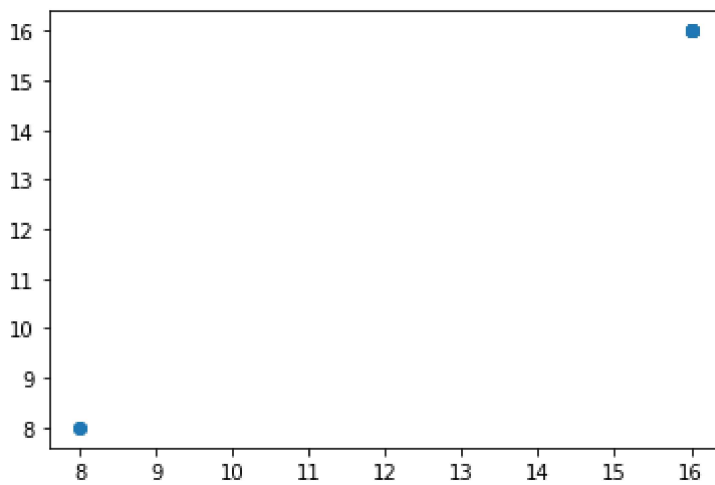
```
In [15]: coeff = pd.DataFrame(lr.coef_, x.columns, columns=['Co-efficient'])
          coeff
```

```
Out[15]:
```

	Co-efficient
Sr. No.	-5.683231e-17
No. of Cars	1.000000e+00

```
In [16]: prediction = lr.predict(x_test)
          plt.scatter(y_test, prediction)
```

Out[16]: <matplotlib.collections.PathCollection at 0x2190e950460>



```
In [17]: lr.score(x_test,y_test)
```

```
Out[17]: 1.0
```

```
In [18]: from sklearn.linear_model import Ridge,Lasso
```

```
In [19]: rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
rr.score(x_test,y_test)
rr.score(x_train,y_train)
```

```
Out[19]: 0.9982383408308338
```

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

```
Out[20]: 0.9979129387960406
```

```
In [21]: la = Lasso(alpha=10)
la.fit(x_train,y_train)
```

```
Out[21]: Lasso(alpha=10)
```

```
In [22]: la.score(x_test,y_test)
```

```
Out[22]: 0.5122591863421235
```

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

```
Out[23]: ElasticNet()
```

```
In [24]: print(en.coef_)
```

```
[-0.01741441  0.9194055 ]
```

```
In [25]: print(en.intercept_)
```

```
1.2303085175699309
```

```
In [26]: print(en.predict(x_test))
```

```
[15.88855333 15.80148129  8.18502116 15.78406688 15.60992281  8.21984998  
15.76665247 15.66216603]
```

```
In [27]: print(en.score(x_test,y_test))
```

```
0.9947732596733252
```

## Evaluation Metrics

```
In [28]: from sklearn import metrics
```

```
In [29]: print("Mean Absolute Error:",metrics.mean_absolute_error(y_test,prediction))
```

```
Mean Absolute Error: 4.440892098500626e-16
```

```
In [30]: print("Mean Squared Error:",metrics.mean_squared_error(y_test,prediction))
```

```
Mean Squared Error: 7.888609052210118e-31
```

```
In [31]: print("Root Mean Squared Error:",np.sqrt(metrics.mean_squared_error(y_test,prediction))
```

```
Root Mean Squared Error: 8.881784197001252e-16
```

```
In [32]: import pickle
```

```
In [33]: filename='prediction'  
pickle.dump(lr,open(filename,'wb'))
```

```
In [34]: import pandas as pd  
import pickle
```

```
In [35]: filename='prediction'  
model = pickle.load(open(filename,'rb'))
```

```
In [36]: real = [[10,20],[11,45]]  
result = model.predict(real)
```

```
In [37]: result
```



```
Out[37]: array([20., 45.])
```