31-07-2023

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

In [511]:
```
a=pd.read_csv(r"C:\Users\user\Downloads\23_Vande Bharat.csv")
a
```

Out[511]:

	Sr. No.	Train Name	Train Number	Originating City	Originating Station	Terminal City	Terminal Stat
0	1	New Delhi - Varanasi Vande Bharat Express	22435/22436	Delhi	New Delhi	Varanasi	Varanasi Junct
1	2	New Delhi - Shri Mata Vaishno Devi Katra Vande...	22439/22440	Delhi	New Delhi	Katra	Shri Mata Vaisl Devi Ka
2	3	Mumbai Central - Gandhinagar Capital Vande Bha...	20901/20902	Mumbai	Mumbai Central	Gandhinagar	Gandhinagar Cap
3	4	New Delhi - Amb Andaura Vande Bharat Express	22447/22448	Delhi	New Delhi	Andaura	Amb Anda
4	5	MGR Chennai Central - Mysuru Vande Bharat Express	20607/20608	Chennai	Chennai Central	Mysuru	Mysore Junct
5	6	Bilaspur - Nagpur Vande Bharat Express	20825/20826	Bilaspur, Chhattisgarh	Bilaspur Junction	Nagpur	Nagpur Junct
6	7	Howrah - New Jalpaiguri Vande Bharat Express	22301/22302	Kolkata	Howrah Junction	Siliguri	New Jalpaig Junct
7	8	Visakhapatnam - Secunderabad Vande Bharat Express	20833/20834	Visakhapatnam	Visakhapatnam Junction	Hyderabad	Secunderal Junct
8	9	Mumbai CSMT - Solapur Vande Bharat Express	22225/22226	Mumbai	Chhatrapati Shivaji Terminus	Solapur	Sola
9	10	Mumbai CSMT - Sainagar Shirdi Vande Bharat Exp...	22223/22224	Mumbai	Chhatrapati Shivaji Terminus	Shirdi	Sainagar Sh
10	11	Rani Kamalapati (Habibganj) - Hazrat Nizamuddi...	20171/20172	Bhopal	Habibganj (Rani Kamalapati)	Delhi	Hazrat Nizamuc
11	12	Secunderabad - Tirupati Vande Bharat Express	20701/20702	Hyderabad	Secunderabad Junction	Tirupati	Tirup
12	13	MGR Chennai Central - Coimbatore Vande Bharat ...	20643/20644	Chennai	Chennai Central	Coimbatore	Coimbatore Junct
13	14	Delhi Cantonment - Ajmer Vande Bharat Express	20977/20978	Delhi	Delhi Cantonment	Ajmer	Ajmer Junct
14	15	Kasaragod - Thiruvananthapuram Vande Bharat Ex...	20633/20634	Kasaragod	Kasaragod	Thiruvananthapuram	Thiruvananthapur Cen
15	16	Howrah - Puri Vande Bharat Express	22895/22896	Kolkata	Howrah Junction	Puri	P
16	17	Anand Vihar Terminal - Dehradun Vande Bharat E...	22457/22458	Delhi	Anand Vihar Terminal	Dehradun	Dehradun Term

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

In [512]:
```
a=a.head(10)
a
```

Out[512]:

Train Number	Originating City	Originating Station	Terminal City	Terminal Station	Operator	No. of Cars	Frequency	Distance	T
435/22436	Delhi	New Delhi	Varanasi	Varanasi Junction	NR	16	Except Thursdays	759 km (472 mi)	
439/22440	Delhi	New Delhi	Katra	Shri Mata Vaishno Devi Katra	NR	16	Except Tuesdays	655 km (407 mi)	
901/20902	Mumbai	Mumbai Central	Gandhinagar	Gandhinagar Capital	WR	16	Except Wednesdays	522 km (324 mi)	
447/22448	Delhi	New Delhi	Andaura	Amb Andaura	NR	16	Except Fridays	412 km (256 mi)	
607/20608	Chennai	Chennai Central	Mysuru	Mysore Junction	SR	16	Except Wednesdays	496 km (308 mi)	
825/20826	Bilaspur, Chhattisgarh	Bilaspur Junction	Nagpur	Nagpur Junction	SECR	8	Except Saturdays	412 km (256 mi)	
301/22302	Kolkata	Howrah Junction	Siliguri	New Jalpaiguri Junction	ER	16	Except Wednesdays	565 km (351 mi)	
833/20834	Visakhapatnam	Visakhapatnam Junction	Hyderabad	Secunderabad Junction	ECoR	16	Except Sundays	698 km (434 mi)	
225/22226	Mumbai	Chhatrapati Shivaji Terminus	Solapur	Solapur	CR	16	Except Wednesdays (22225) , Except Thursdays (...	452 km (281 mi)	
223/22224	Mumbai	Chhatrapati Shivaji Terminus	Shirdi	Sainagar Shirdi	CR	16	Except Tuesdays	339 km (211 mi)	

In [513]:
```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 16 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   Sr. No.              10 non-null     int64 
 1   Train Name           10 non-null     object
 2   Train Number         10 non-null     object
 3   Originating City     10 non-null     object
 4   Originating Station  10 non-null     object
 5   Terminal City        10 non-null     object
 6   Terminal Station     10 non-null     object
 7   Operator             10 non-null     object
 8   No. of Cars          10 non-null     int64 
 9   Frequency            10 non-null     object
 10  Distance             10 non-null     object
 11  Travel Time          10 non-null     object
 12  Speed                10 non-null     object
 13  Average Speed        10 non-null     object
 14  Inauguration         10 non-null     object
 15  Average occupancy    10 non-null     object
dtypes: int64(2), object(14)
memory usage: 1.4+ KB
```

In [514]:
```
a.columns
```

Out[514]:
```
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 [515]:
```
d=a[['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']]
d
```

Out[515]:

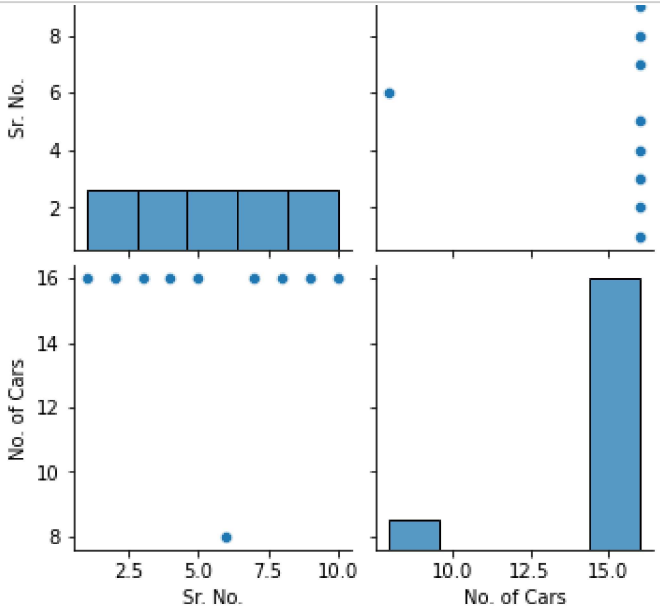
Train Number	Originating City	Originating Station	Terminal City	Terminal Station	Operator	No. of Cars	Frequency	Distance	T
435/22436	Delhi	New Delhi	Varanasi	Varanasi Junction	NR	16	Except Thursdays	759 km (472 mi)	
439/22440	Delhi	New Delhi	Katra	Shri Mata Vaishno Devi Katra	NR	16	Except Tuesdays	655 km (407 mi)	
901/20902	Mumbai	Mumbai Central	Gandhinagar	Gandhinagar Capital	WR	16	Except Wednesdays	522 km (324 mi)	
447/22448	Delhi	New Delhi	Andaura	Amb Andaura	NR	16	Except Fridays	412 km (256 mi)	
607/20608	Chennai	Chennai Central	Mysuru	Mysore Junction	SR	16	Except Wednesdays	496 km (308 mi)	
825/20826	Bilaspur, Chhattisgarh	Bilaspur Junction	Nagpur	Nagpur Junction	SECR	8	Except Saturdays	412 km (256 mi)	
301/22302	Kolkata	Howrah Junction	Siliguri	New Jalpaiguri Junction	ER	16	Except Wednesdays	565 km (351 mi)	
833/20834	Visakhapatnam	Visakhapatnam Junction	Hyderabad	Secunderabad Junction	ECoR	16	Except Sundays	698 km (434 mi)	
225/22226	Mumbai	Chhatrapati Shivaji Terminus	Solapur	Solapur	CR	16	Except Wednesdays (22225) , Except Thursdays (...	452 km (281 mi)	
223/22224	Mumbai	Chhatrapati Shivaji Terminus	Shirdi	Sainagar Shirdi	CR	16	Except Tuesdays	339 km (211 mi)	

In [516]:
```
d.describe()
```

Out[516]:

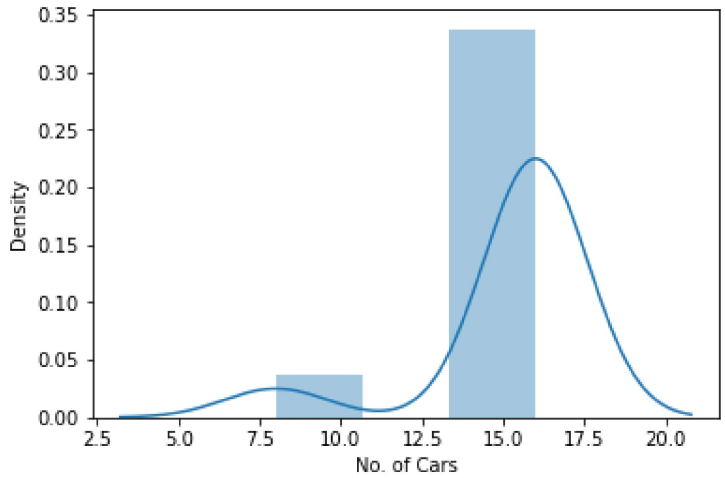
	Sr. No.	No. of Cars
count	10.00000	10.000000
mean	5.50000	15.200000
std	3.02765	2.529822
min	1.00000	8.000000
25%	3.25000	16.000000
50%	5.50000	16.000000
75%	7.75000	16.000000
max	10.00000	16.000000

In [517]:
```
sns.pairplot(d)
```



In [519]:
```
sns.distplot(a['No. of Cars'])
```

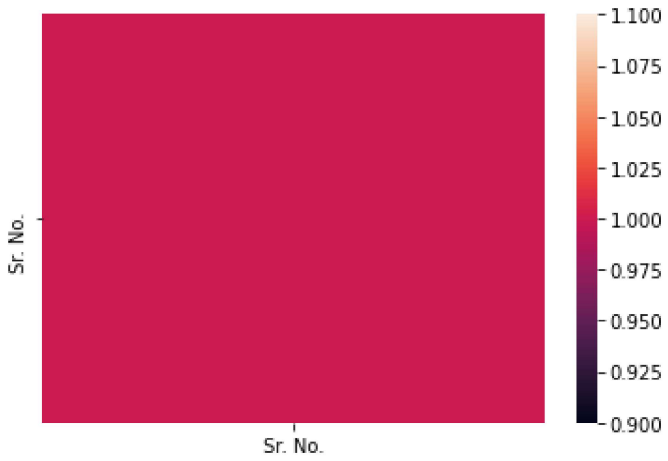
Out[519]: <AxesSubplot:xlabel='No. of Cars', ylabel='Density'>



In [521]:
```
x1=a[['Sr. No.']]
```

In [522]:
```
sns.heatmap(x1.corr())
```

Out[522]: <AxesSubplot:>



In [523]:
```
x=a[['Sr. No.']]
y=a['No. of Cars']
```

In [524]:
```
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 [525]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[525]: LinearRegression()

In [526]:
```
print(lr.intercept_)
```

```
16.131147540983605
```

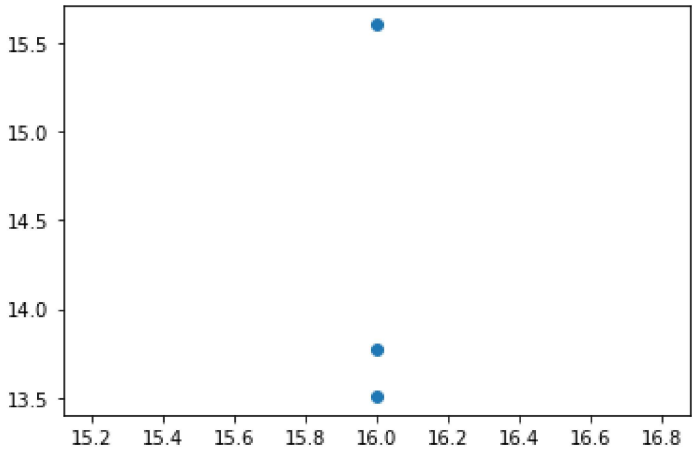
In [527]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[527]:

	Co-efficient
Sr. No.	-0.262295

In [528]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[528]:
```
<matplotlib.collections.PathCollection at 0x190c7da9df0>
```



In [529]:
```
print(lr.score(x_test,y_test))
```

```
0.0
```

In [530]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [531]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[531]:
```
Ridge(alpha=10)
```

In [532]:
```
rr.score(x_test,y_test)
```

Out[532]:
```
0.0
```

In [533]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[533]:
```
Lasso(alpha=10)
```

In [534]:
```
la.score(x_test,y_test)
```

Out[534]:
```
0.0
```

In [535]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[535]:
```
ElasticNet()
```

In [536]:
```
print(en.coef_)
```

```
[-0.14711359]
```

In [537]:
```
print(en.intercept_)
```

```
15.57169459962756
```

In [538]:
```
print(en.predict(x_test))
```

```
[15.27746741 14.10055866 14.24767225]
```

In [539]:
```
en.score(x_test,y_test)
```

Out[539]:
```
0.0
```

In [540]:
```
from sklearn import metrics
```

In [541]:
```
print("Mean Absolute Error",metrics.mean_absolute_error(y_test,prediction))
```

```
Mean Absolute Error 1.7049180327868865
```

In [542]:
```
print("Mean Squared Error",metrics.mean_squared_error(y_test,prediction))
```

```
Mean Squared Error 3.778195825494942
```

In [543]:
```
print(" Root Mean Squared Error",np.sqrt(metrics.mean_squared_error(y_test,prediction))
```

```
 Root Mean Squared Error 1.9437581705281504
```

In [544]:
```
import pickle
```

In [545]:
```
filename="prediction"
pickle.dump(lr,open(filename,'wb'))
```

In [546]:
```
import pandas as pd
import pickle
```

In [547]:
```
filename="prediction"
model=pickle.load(open(filename,"rb"))
```

In [548]:
```
real=[[10],[15]]
result=model.predict(real)
```

In [549]:
```
result
```

Out[549]:
```
array([13.50819672, 12.19672131])
```