

2001

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

```
In [2]: df=pd.read_csv("madrid_2001.csv")
df
```

```
Out[2]:
```

	date	BEN	CO	EBE	MXV	NMHC	NO_2	NOx	OXY	O_3	
0	2001-08-01 01:00:00	NaN	0.37	NaN	NaN	NaN	58.400002	87.150002	NaN	34.529999	105.00
1	2001-08-01 01:00:00	1.50	0.34	1.49	4.10	0.07	56.250000	75.169998	2.11	42.160000	100.50
2	2001-08-01 01:00:00	NaN	0.28	NaN	NaN	NaN	50.660000	61.380001	NaN	46.310001	100.00
3	2001-08-01 01:00:00	NaN	0.47	NaN	NaN	NaN	69.790001	73.449997	NaN	40.650002	69.70
4	2001-08-01 01:00:00	NaN	0.39	NaN	NaN	NaN	22.830000	24.799999	NaN	66.309998	75.10
...
217867	2001-04-01 00:00:00	10.45	1.81	NaN	NaN	NaN	73.000000	264.399994	NaN	5.200000	47.80
217868	2001-04-01 00:00:00	5.20	0.69	4.56	NaN	0.13	71.080002	129.300003	NaN	13.460000	26.80
217869	2001-04-01 00:00:00	0.49	1.09	NaN	1.00	0.19	76.279999	128.399994	0.35	5.020000	40.70
217870	2001-04-01 00:00:00	5.62	1.01	5.04	11.38	NaN	80.019997	197.000000	2.58	5.840000	37.80
217871	2001-04-01 00:00:00	8.09	1.62	6.66	13.04	0.18	76.809998	206.300003	5.20	8.340000	35.30

217872 rows × 16 columns



```
In [3]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 217872 entries, 0 to 217871
Data columns (total 16 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   date        217872 non-null  object
1   BEN         70389 non-null   float64
2   CO          216341 non-null  float64
3   EBE         57752 non-null   float64
4   MXY         42753 non-null   float64
5   NMHC        85719 non-null   float64
6   NO_2        216331 non-null  float64
7   NOx         216318 non-null  float64
8   OXY         42856 non-null   float64
9   O_3         216514 non-null  float64
10  PM10        207776 non-null  float64
11  PXY         42845 non-null   float64
12  SO_2        216403 non-null  float64
13  TCH         85797 non-null   float64
14  TOL         70196 non-null   float64
15  station     217872 non-null  int64
dtypes: float64(14), int64(1), object(1)
memory usage: 26.6+ MB
```

```
In [4]: df1=df.dropna()
df1
```

```
Out[4]:
```

	date	BEN	CO	EBE	MXV	NMHC	NO_2	NOx	OXY	O_3
1	2001-08-01 01:00:00	1.50	0.34	1.49	4.100000	0.07	56.250000	75.169998	2.11	42.160000
5	2001-08-01 01:00:00	2.11	0.63	2.48	5.940000	0.05	66.260002	118.099998	3.15	33.500000
21	2001-08-01 01:00:00	0.80	0.43	0.71	1.200000	0.10	27.190001	29.700001	0.76	56.990002
23	2001-08-01 01:00:00	1.29	0.34	1.41	3.090000	0.07	40.750000	51.570000	1.70	51.580002
25	2001-08-01 02:00:00	0.87	0.06	0.88	2.410000	0.01	29.709999	31.440001	1.20	56.520000
...
217829	2001-03-31 23:00:00	11.76	4.48	7.71	17.219999	0.89	103.900002	548.500000	7.62	9.680000
217847	2001-03-31 23:00:00	9.79	2.65	7.59	9.730000	0.46	91.320000	315.899994	3.75	6.660000
217849	2001-04-01 00:00:00	5.86	1.22	5.66	13.710000	0.25	64.370003	218.300003	6.46	7.480000
217853	2001-04-01 00:00:00	14.47	1.83	11.39	26.059999	0.33	84.230003	259.200012	11.39	5.440000
217871	2001-04-01 00:00:00	8.09	1.62	6.66	13.040000	0.18	76.809998	206.300003	5.20	8.340000

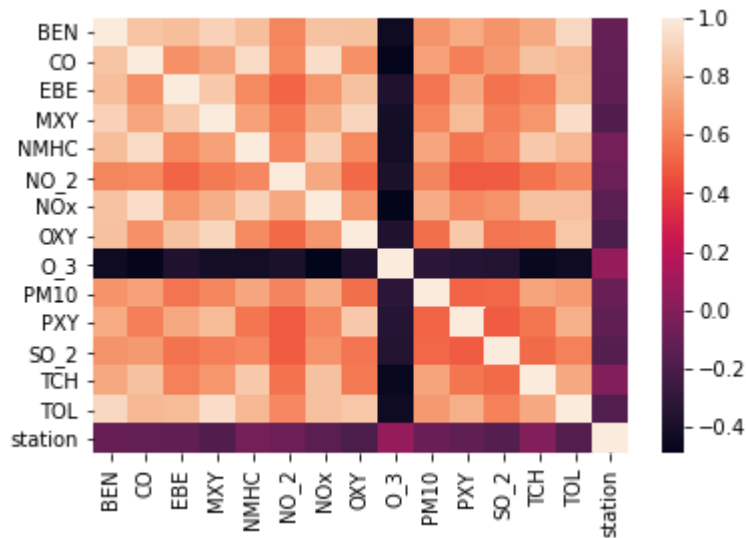
29669 rows × 16 columns



```
In [5]: df1=df1.drop(["date"],axis=1)
```

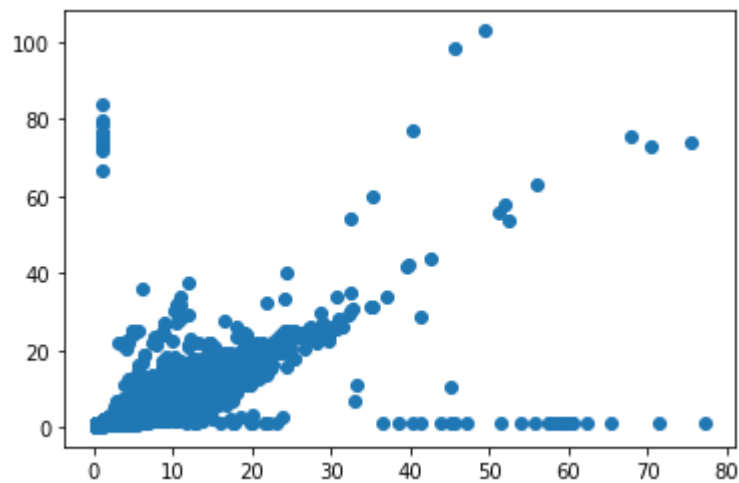
```
In [6]: sns.heatmap(df1.corr())
```

```
Out[6]: <AxesSubplot:>
```



```
In [7]: plt.plot(df1["EBE"],df1["PXY"],"o")
```

```
Out[7]: [<matplotlib.lines.Line2D at 0x23d88ab14f0>]
```



```
In [8]: data=df[["EBE","PXY"]]
```

```
In [9]: # sns.stripplot(x=df["EBE"],y=df["PXY"],jitter=True,marker='o',color='blue')
```

```
In [41]: x=df1.drop(["EBE"],axis=1)
y=df1["EBE"]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

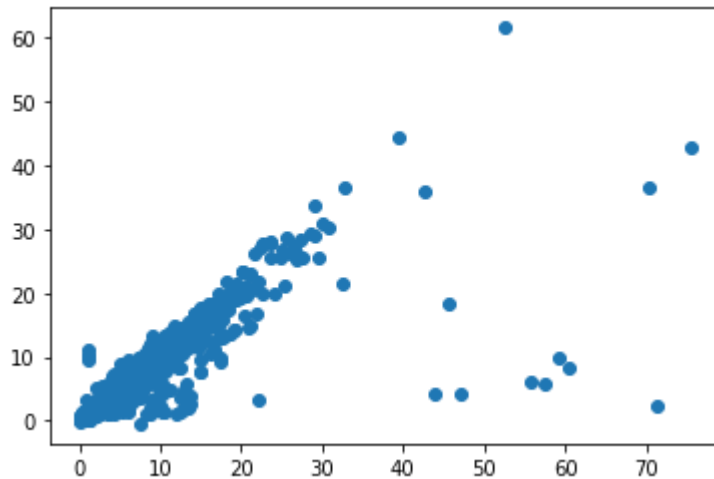
Linear

```
In [11]: li=LinearRegression()
li.fit(x_train,y_train)
```

```
Out[11]: LinearRegression()
```

```
In [12]: prediction=li.predict(x_test)
plt.scatter(y_test,prediction)
```

```
Out[12]: <matplotlib.collections.PathCollection at 0x23d88b7d460>
```



```
In [13]: lis=li.score(x_test,y_test)
```

```
In [14]: df1["TCH"].value_counts()
```

```
Out[14]: 1.28    988
1.32    938
1.33    908
1.29    908
1.27    905
...
4.39     1
3.57     1
4.37     1
3.59     1
4.21     1
Name: TCH, Length: 269, dtype: int64
```

```
In [15]: df1.loc[df1["TCH"]<1.40,"TCH"]=1
df1.loc[df1["TCH"]>1.40,"TCH"]=2
df1["TCH"].value_counts()
```

```
Out[15]: 1.0    17204
2.0    12465
Name: TCH, dtype: int64
```

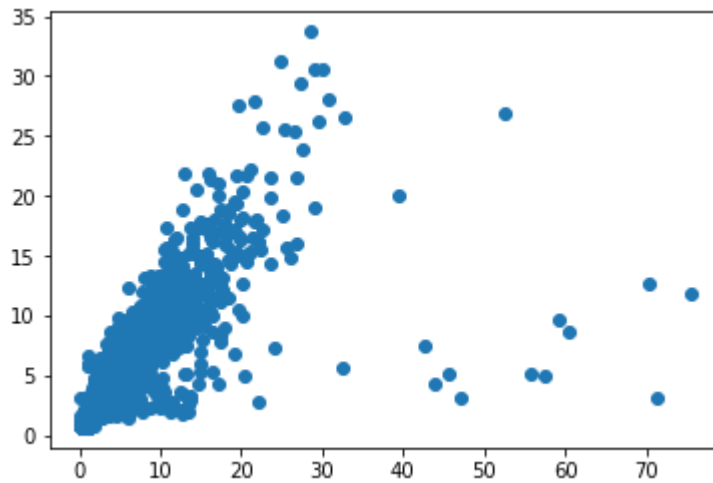
Lasso

```
In [16]: la=Lasso(alpha=5)
la.fit(x_train,y_train)
```

Out[16]: Lasso(alpha=5)

```
In [17]: prediction1=la.predict(x_test)
plt.scatter(y_test,prediction1)
```

Out[17]: <matplotlib.collections.PathCollection at 0x23d88bec670>



```
In [18]: las=la.score(x_test,y_test)
```

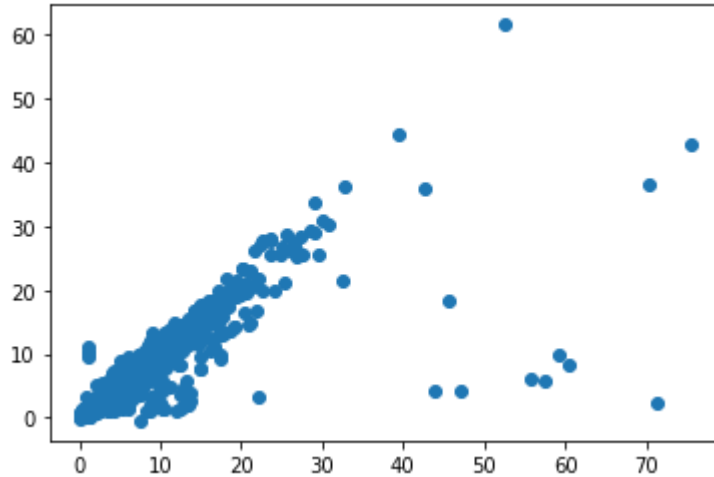
Ridge

```
In [19]: rr=Ridge(alpha=1)
rr.fit(x_train,y_train)
```

Out[19]: Ridge(alpha=1)

```
In [20]: prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[20]: <matplotlib.collections.PathCollection at 0x23d88ac8ee0>



```
In [21]: rrs=rr.score(x_test,y_test)
```

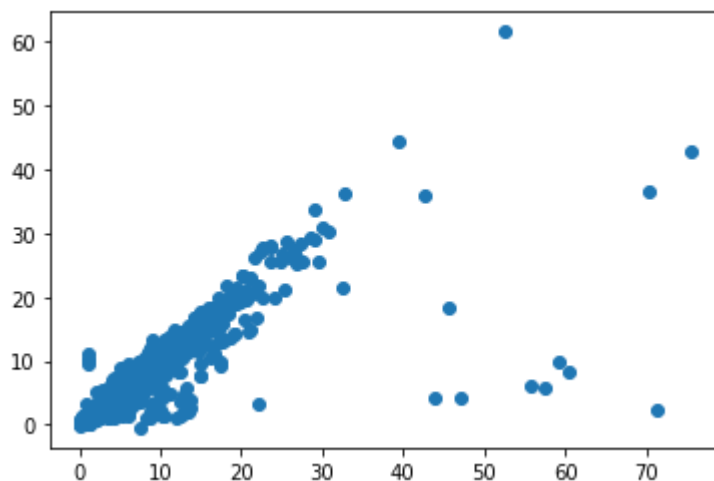
ElasticNet

```
In [22]: en=ElasticNet()
en.fit(x_train,y_train)
```

Out[22]: ElasticNet()

```
In [23]: prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[23]: <matplotlib.collections.PathCollection at 0x23d89e36cd0>



```
In [24]: ens=en.score(x_test,y_test)
```

```
In [25]: print(rr.score(x_test,y_test))
rr.score(x_train,y_train)
```

```
0.7877953739336674
```

```
Out[25]: 0.759499092953202
```

Logistic

```
In [43]: g={"TCH":{1.0:"Low",2.0:"High"}}
df1=df1.replace(g)
df1["TCH"].value_counts()
```

```
Out[43]: Low      17204
High      12465
Name: TCH, dtype: int64
```

```
In [44]: x=df1.drop(["TCH"],axis=1)
y=df1["TCH"]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

```
In [45]: lo=LogisticRegression()
lo.fit(x_train,y_train)
```

```
Out[45]: LogisticRegression()
```

```
In [46]: prediction3=lo.predict(x_test)
plt.scatter(y_test,prediction3)
```

```
Out[46]: <matplotlib.collections.PathCollection at 0x23d896d8490>
```



```
In [47]: los=lo.score(x_test,y_test)
```


Random Forest

```
In [30]: from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
```

```
In [31]: g1={"TCH":{"Low":1.0,"High":2.0}}
df1=df1.replace(g1)
```

```
In [32]: x=df1.drop(["TCH"],axis=1)
y=df1["TCH"]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

```
In [33]: rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

```
Out[33]: RandomForestClassifier()
```

```
In [34]: parameter={
    'max_depth':[1,2,4,5,6],
    'min_samples_leaf':[5,10,15,20,25],
    'n_estimators':[10,20,30,40,50]
}
```

```
In [35]: grid_search=GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

```
Out[35]: GridSearchCV(cv=2, estimator=RandomForestClassifier(),
    param_grid={'max_depth': [1, 2, 4, 5, 6],
    'min_samples_leaf': [5, 10, 15, 20, 25],
    'n_estimators': [10, 20, 30, 40, 50]},
    scoring='accuracy')
```

```
In [36]: rfcs=grid_search.best_score_
```

```
In [37]: rfc_best=grid_search.best_estimator_
```

```
In [38]: from sklearn.tree import plot_tree

plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5], feature_names=x.columns, class_names=['Yes', "I
```

```

Out[38]: [Text(2387.0, 2019.0857142857144, 'CO <= 1.025\ngini = 0.488\nsamples = 13145\nvalue = [11977, 8791]\nclass = Yes'),
Text(1220.923076923077, 1708.457142857143, 'PM10 <= 25.05\ngini = 0.292\nsamples = 8587\nvalue = [11173, 2407]\nclass = Yes'),
Text(610.4615384615385, 1397.8285714285716, 'MXV <= 3.655\ngini = 0.169\nsamples = 5035\nvalue = [7134, 732]\nclass = Yes'),
Text(305.2307692307692, 1087.2, 'MXV <= 1.875\ngini = 0.074\nsamples = 2723\nvalue = [4067, 163]\nclass = Yes'),
Text(152.6153846153846, 776.5714285714287, 'SO_2 <= 19.11\ngini = 0.044\nsamples = 1310\nvalue = [1978, 46]\nclass = Yes'),
Text(76.3076923076923, 465.9428571428573, 'EBE <= 0.605\ngini = 0.039\nsamples = 1283\nvalue = [1946, 40]\nclass = Yes'),
Text(38.15384615384615, 155.3142857142857, 'gini = 0.021\nsamples = 651\nvalue = [1014, 11]\nclass = Yes'),
Text(114.46153846153845, 155.3142857142857, 'gini = 0.059\nsamples = 632\nvalue = [932, 29]\nclass = Yes'),
Text(228.9230769230769, 465.9428571428573, 'CO <= 0.81\ngini = 0.266\nsamples = 27\nvalue = [32, 6]\nclass = Yes'),
Text(190.76923076923077, 155.3142857142857, 'gini = 0.062\nsamples = 22\nvalue = [30, 1]\nclass = Yes'),
Text(267.0769230769231, 155.3142857142857, 'gini = 0.408\nsamples = 5\nvalue = [2, 5]\nclass = No'),
Text(457.8461538461538, 776.5714285714287, 'NMHC <= 0.145\ngini = 0.1\nsamples = 1413\nvalue = [2089, 117]\nclass = Yes'),
Text(381.53846153846155, 465.9428571428573, 'O_3 <= 18.605\ngini = 0.076\nsamples = 1346\nvalue = [2028, 83]\nclass = Yes'),
Text(343.38461538461536, 155.3142857142857, 'gini = 0.258\nsamples = 200\nvalue = [268, 48]\nclass = Yes'),
Text(419.6923076923077, 155.3142857142857, 'gini = 0.038\nsamples = 1146\nvalue = [1760, 35]\nclass = Yes'),
Text(534.1538461538462, 465.9428571428573, 'station <= 28079030.0\ngini = 0.46\nsamples = 67\nvalue = [61, 34]\nclass = Yes'),
Text(496.0, 155.3142857142857, 'gini = 0.444\nsamples = 20\nvalue = [10, 20]\nclass = No'),
Text(572.3076923076923, 155.3142857142857, 'gini = 0.338\nsamples = 47\nvalue = [51, 14]\nclass = Yes'),
Text(915.6923076923076, 1087.2, 'PM10 <= 14.945\ngini = 0.264\nsamples = 2312\nvalue = [3067, 569]\nclass = Yes'),
Text(763.0769230769231, 776.5714285714287, 'O_3 <= 10.83\ngini = 0.173\nsamples = 1047\nvalue = [1479, 156]\nclass = Yes'),
Text(686.7692307692307, 465.9428571428573, 'NO_2 <= 40.42\ngini = 0.456\nsamples = 119\nvalue = [127, 69]\nclass = Yes'),
Text(648.6153846153846, 155.3142857142857, 'gini = 0.18\nsamples = 12\nvalue = [2, 18]\nclass = No'),
Text(724.9230769230769, 155.3142857142857, 'gini = 0.412\nsamples = 107\nvalue = [125, 51]\nclass = Yes'),
Text(839.3846153846154, 465.9428571428573, 'CO <= 0.965\ngini = 0.114\nsamples = 928\nvalue = [1352, 87]\nclass = Yes'),
Text(801.2307692307692, 155.3142857142857, 'gini = 0.094\nsamples = 904\nvalue = [1330, 69]\nclass = Yes'),
Text(877.5384615384615, 155.3142857142857, 'gini = 0.495\nsamples = 24\nvalue = [22, 18]\nclass = Yes'),
Text(1068.3076923076924, 776.5714285714287, 'NOx <= 107.05\ngini = 0.328\nsamples = 1265\nvalue = [1588, 413]\nclass = Yes'),
Text(992.0, 465.9428571428573, 'BEN <= 2.995\ngini = 0.235\nsamples = 644\nvalue = [891, 140]\nclass = Yes'),
Text(953.8461538461538, 155.3142857142857, 'gini = 0.206\nsamples = 563\nvalue = [1014, 11]\nclass = Yes')

```

```

ue = [796, 105]\n\nclass = Yes'),
  Text(1030.1538461538462, 155.3142857142857, 'gini = 0.393\n\ nsamples = 81\n\nvalue = [95, 35]\n\nclass = Yes'),
  Text(1144.6153846153845, 465.9428571428573, 'TOL <= 11.075\n\ ngini = 0.404\n\nsamples = 621\n\nvalue = [697, 273]\n\nclass = Yes'),
  Text(1106.4615384615383, 155.3142857142857, 'gini = 0.317\n\ nsamples = 159\n\nvalue = [203, 50]\n\nclass = Yes'),
  Text(1182.7692307692307, 155.3142857142857, 'gini = 0.429\n\ nsamples = 462\n\nvalue = [494, 223]\n\nclass = Yes'),
  Text(1831.3846153846152, 1397.8285714285716, 'O_3 <= 23.205\n\ ngini = 0.414\n\nsamples = 3552\n\nvalue = [4039, 1675]\n\nclass = Yes'),
  Text(1526.1538461538462, 1087.2, 'NMHC <= 0.165\n\ ngini = 0.5\n\nsamples = 1258\n\nvalue = [1012, 1017]\n\nclass = No'),
  Text(1373.5384615384614, 776.5714285714287, 'NMHC <= 0.135\n\ ngini = 0.417\n\nsamples = 768\n\nvalue = [866, 364]\n\nclass = Yes'),
  Text(1297.2307692307693, 465.9428571428573, 'TOL <= 11.955\n\ ngini = 0.35\n\nsamples = 488\n\nvalue = [624, 182]\n\nclass = Yes'),
  Text(1259.076923076923, 155.3142857142857, 'gini = 0.231\n\ nsamples = 254\n\nvalue = [358, 55]\n\nclass = Yes'),
  Text(1335.3846153846155, 155.3142857142857, 'gini = 0.437\n\ nsamples = 234\n\nvalue = [266, 127]\n\nclass = Yes'),
  Text(1449.8461538461538, 465.9428571428573, 'TOL <= 6.73\n\ ngini = 0.49\n\nsamples = 280\n\nvalue = [242, 182]\n\nclass = Yes'),
  Text(1411.6923076923076, 155.3142857142857, 'gini = 0.456\n\ nsamples = 33\n\nvalue = [19, 35]\n\nclass = No'),
  Text(1488.0, 155.3142857142857, 'gini = 0.479\n\ nsamples = 247\n\nvalue = [223, 147]\n\nclass = Yes'),
  Text(1678.7692307692307, 776.5714285714287, 'NOx <= 114.45\n\ ngini = 0.299\n\nsamples = 490\n\nvalue = [146, 653]\n\nclass = No'),
  Text(1602.4615384615383, 465.9428571428573, 'NO_2 <= 54.96\n\ ngini = 0.091\n\nsamples = 96\n\nvalue = [7, 139]\n\nclass = No'),
  Text(1564.3076923076924, 155.3142857142857, 'gini = 0.0\n\ nsamples = 39\n\nvalue = [0, 64]\n\nclass = No'),
  Text(1640.6153846153845, 155.3142857142857, 'gini = 0.156\n\ nsamples = 57\n\nvalue = [7, 75]\n\nclass = No'),
  Text(1755.076923076923, 465.9428571428573, 'NO_2 <= 90.295\n\ ngini = 0.335\n\nsamples = 394\n\nvalue = [139, 514]\n\nclass = No'),
  Text(1716.923076923077, 155.3142857142857, 'gini = 0.276\n\ nsamples = 278\n\nvalue = [78, 394]\n\nclass = No'),
  Text(1793.2307692307693, 155.3142857142857, 'gini = 0.447\n\ nsamples = 116\n\nvalue = [61, 120]\n\nclass = No'),
  Text(2136.6153846153848, 1087.2, 'station <= 28079068.0\n\ ngini = 0.293\n\nsamples = 2294\n\nvalue = [3027, 658]\n\nclass = Yes'),
  Text(1984.0, 776.5714285714287, 'NMHC <= 0.195\n\ ngini = 0.178\n\nsamples = 1469\n\nvalue = [2129, 233]\n\nclass = Yes'),
  Text(1907.6923076923076, 465.9428571428573, 'NMHC <= 0.135\n\ ngini = 0.141\n\nsamples = 1397\n\nvalue = [2075, 171]\n\nclass = Yes'),
  Text(1869.5384615384614, 155.3142857142857, 'gini = 0.089\n\ nsamples = 1065\n\nvalue = [1629, 80]\n\nclass = Yes'),
  Text(1945.8461538461538, 155.3142857142857, 'gini = 0.281\n\ nsamples = 332\n\nvalue = [446, 91]\n\nclass = Yes'),
  Text(2060.3076923076924, 465.9428571428573, 'O_3 <= 33.825\n\ ngini = 0.498\n\nsamples = 72\n\nvalue = [54, 62]\n\nclass = No'),
  Text(2022.1538461538462, 155.3142857142857, 'gini = 0.284\n\ nsamples = 22\n\nvalue = [6, 29]\n\nclass = No'),
  Text(2098.4615384615386, 155.3142857142857, 'gini = 0.483\n\ nsamples = 50\n\nvalue = [48, 33]\n\nclass = Yes'),

```

```
Text(2289.230769230769, 776.5714285714287, 'BEN <= 2.805\ngini = 0.436\nsamples = 825\nvalue = [898, 425]\nclass = Yes'),
Text(2212.9230769230767, 465.9428571428573, 'PXY <= 2.515\ngini = 0.305\nsamples = 569\nvalue = [741, 171]\nclass = Yes'),
Text(2174.769230769231, 155.3142857142857, 'gini = 0.184\nsamples = 461\nvalue = [656, 75]\nclass = Yes'),
Text(2251.076923076923, 155.3142857142857, 'gini = 0.498\nsamples = 108\nvalue = [85, 96]\nclass = No'),
Text(2365.5384615384614, 465.9428571428573, 'EBE <= 2.93\ngini = 0.472\nsamples = 256\nvalue = [157, 254]\nclass = No'),
Text(2327.3846153846152, 155.3142857142857, 'gini = 0.375\nsamples = 46\nvalue = [51, 17]\nclass = Yes'),
Text(2403.6923076923076, 155.3142857142857, 'gini = 0.427\nsamples = 210\nvalue = [106, 237]\nclass = No'),
Text(3553.076923076923, 1708.457142857143, 'MXY <= 14.675\ngini = 0.199\nsamples = 4558\nvalue = [804, 6384]\nclass = No'),
Text(3052.3076923076924, 1397.8285714285716, 'NMHC <= 0.205\ngini = 0.263\nsamples = 2926\nvalue = [719, 3889]\nclass = No'),
Text(2747.076923076923, 1087.2, 'SO_2 <= 42.85\ngini = 0.488\nsamples = 651\nvalue = [585, 427]\nclass = Yes'),
Text(2594.4615384615386, 776.5714285714287, 'PM10 <= 53.345\ngini = 0.497\nsamples = 583\nvalue = [481, 416]\nclass = Yes'),
Text(2518.153846153846, 465.9428571428573, 'PXY <= 2.175\ngini = 0.49\nsamples = 525\nvalue = [460, 344]\nclass = Yes'),
Text(2480.0, 155.3142857142857, 'gini = 0.469\nsamples = 121\nvalue = [68, 113]\nclass = No'),
Text(2556.3076923076924, 155.3142857142857, 'gini = 0.467\nsamples = 404\nvalue = [392, 231]\nclass = Yes'),
Text(2670.769230769231, 465.9428571428573, 'NMHC <= 0.165\ngini = 0.35\nsamples = 58\nvalue = [21, 72]\nclass = No'),
Text(2632.6153846153848, 155.3142857142857, 'gini = 0.477\nsamples = 18\nvalue = [11, 17]\nclass = No'),
Text(2708.9230769230767, 155.3142857142857, 'gini = 0.26\nsamples = 40\nvalue = [10, 55]\nclass = No'),
Text(2899.6923076923076, 776.5714285714287, 'NOx <= 202.6\ngini = 0.173\nsamples = 68\nvalue = [104, 11]\nclass = Yes'),
Text(2823.3846153846152, 465.9428571428573, 'PM10 <= 46.23\ngini = 0.078\nsamples = 57\nvalue = [95, 4]\nclass = Yes'),
Text(2785.230769230769, 155.3142857142857, 'gini = 0.022\nsamples = 52\nvalue = [89, 1]\nclass = Yes'),
Text(2861.5384615384614, 155.3142857142857, 'gini = 0.444\nsamples = 5\nvalue = [6, 3]\nclass = Yes'),
Text(2976.0, 465.9428571428573, 'PM10 <= 36.795\ngini = 0.492\nsamples = 11\nvalue = [9, 7]\nclass = Yes'),
Text(2937.846153846154, 155.3142857142857, 'gini = 0.0\nsamples = 5\nvalue = [8, 0]\nclass = Yes'),
Text(3014.153846153846, 155.3142857142857, 'gini = 0.219\nsamples = 6\nvalue = [1, 7]\nclass = No'),
Text(3357.5384615384614, 1087.2, 'NMHC <= 0.225\ngini = 0.072\nsamples = 2275\nvalue = [134, 3462]\nclass = No'),
Text(3204.9230769230767, 776.5714285714287, 'NO_2 <= 102.45\ngini = 0.365\nsamples = 275\nvalue = [102, 322]\nclass = No'),
Text(3128.6153846153848, 465.9428571428573, 'TOL <= 21.035\ngini = 0.31\nsamples = 220\nvalue = [65, 274]\nclass = No'),
Text(3090.4615384615386, 155.3142857142857, 'gini = 0.201\nsamples = 147\nvalue = [25, 195]\nclass = No'),
Text(3166.769230769231, 155.3142857142857, 'gini = 0.446\nsamples = 73\nvalue = [10, 10]\nclass = No')
```

```

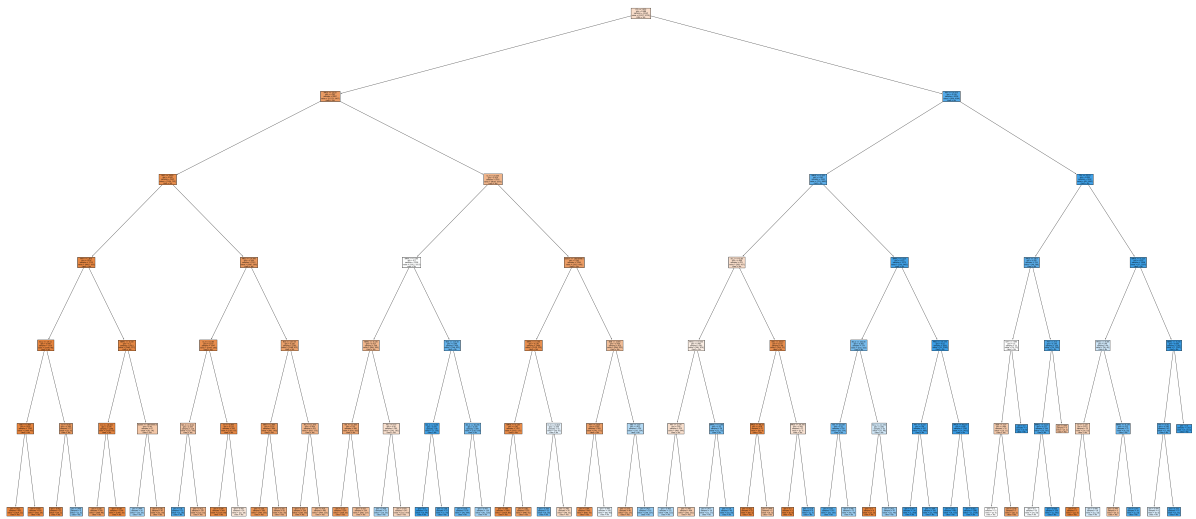
e = [40, 79]\n\nclass = No'),
  Text(3281.230769230769, 465.9428571428573, 'SO_2 <= 11.575\ngini = 0.492\nsamples = 55\n\nclass = No'),
  Text(3243.076923076923, 155.3142857142857, 'gini = 0.0\nsamples = 8\n\nclass = Yes'),
  Text(3319.3846153846152, 155.3142857142857, 'gini = 0.431\nsamples = 47\n\nclass = No'),
  Text(3510.153846153846, 776.5714285714287, 'PM10 <= 35.275\ngini = 0.02\nsamples = 2000\n\nclass = No'),
  Text(3433.846153846154, 465.9428571428573, 'PXY <= 3.215\ngini = 0.06\nsamples = 437\n\nclass = No'),
  Text(3395.6923076923076, 155.3142857142857, 'gini = 0.007\nsamples = 176\n\nclass = No'),
  Text(3472.0, 155.3142857142857, 'gini = 0.094\nsamples = 261\n\nclass = No'),
  Text(3586.4615384615386, 465.9428571428573, 'PM10 <= 46.215\ngini = 0.008\nsamples = 1563\n\nclass = No'),
  Text(3548.3076923076924, 155.3142857142857, 'gini = 0.022\nsamples = 482\n\nclass = No'),
  Text(3624.6153846153848, 155.3142857142857, 'gini = 0.002\nsamples = 1081\n\nclass = No'),
  Text(4053.846153846154, 1397.8285714285716, 'TOL <= 32.51\ngini = 0.064\nsamples = 1632\n\nclass = No'),
  Text(3853.5384615384614, 1087.2, 'PM10 <= 26.4\ngini = 0.245\nsamples = 224\n\nclass = No'),
  Text(3777.230769230769, 776.5714285714287, 'OXY <= 7.595\ngini = 0.5\nsamples = 25\n\nclass = Yes'),
  Text(3739.076923076923, 465.9428571428573, 'BEN <= 5.37\ngini = 0.464\nsamples = 20\n\nclass = Yes'),
  Text(3700.9230769230767, 155.3142857142857, 'gini = 0.5\nsamples = 13\n\nclass = Yes'),
  Text(3777.230769230769, 155.3142857142857, 'gini = 0.278\nsamples = 7\n\nclass = Yes'),
  Text(3815.3846153846152, 465.9428571428573, 'gini = 0.0\nsamples = 5\n\nclass = No'),
  Text(3929.846153846154, 776.5714285714287, 'PXY <= 25.0\ngini = 0.176\nsamples = 199\n\nclass = No'),
  Text(3891.6923076923076, 465.9428571428573, 'NMHC <= 0.215\ngini = 0.157\nsamples = 194\n\nclass = No'),
  Text(3853.5384615384614, 155.3142857142857, 'gini = 0.5\nsamples = 35\n\nclass = No'),
  Text(3929.846153846154, 155.3142857142857, 'gini = 0.008\nsamples = 159\n\nclass = No'),
  Text(3968.0, 465.9428571428573, 'gini = 0.444\nsamples = 5\n\nclass = Yes'),
  Text(4254.153846153846, 1087.2, 'NMHC <= 0.215\ngini = 0.032\nsamples = 1408\n\nclass = No'),
  Text(4120.615384615385, 776.5714285714287, 'PM10 <= 40.89\ngini = 0.49\nsamples = 46\n\nclass = No'),
  Text(4044.3076923076924, 465.9428571428573, 'CO <= 1.155\ngini = 0.473\nsamples = 25\n\nclass = Yes'),
  Text(4006.153846153846, 155.3142857142857, 'gini = 0.0\nsamples = 7\n\nclass = Yes'),
  Text(4082.4615384615386, 155.3142857142857, 'gini = 0.494\nsamples = 18\n\nclass = No'),
  Text(4196.923076923077, 465.9428571428573, 'NMHC <= 0.175\ngini = 0.312\nsamples = 21\n\nclass = No'),

```

```

Text(4158.7692307692305, 155.3142857142857, 'gini = 0.444\nsamples = 5\nvalue = [4, 2]\nnclass = Yes'),
Text(4235.076923076923, 155.3142857142857, 'gini = 0.147\nsamples = 16\nvalue = [2, 23]\nnclass = No'),
Text(4387.692307692308, 776.5714285714287, 'NMHC <= 0.245\ngini = 0.006\nsamples = 1362\nvalue = [7, 2167]\nnclass = No'),
Text(4349.538461538462, 465.9428571428573, 'PXY <= 5.81\ngini = 0.198\nsamples = 43\nvalue = [7, 56]\nnclass = No'),
Text(4311.384615384615, 155.3142857142857, 'gini = 0.494\nsamples = 6\nvalue = [4, 5]\nnclass = No'),
Text(4387.692307692308, 155.3142857142857, 'gini = 0.105\nsamples = 37\nvalue = [3, 51]\nnclass = No'),
Text(4425.846153846153, 465.9428571428573, 'gini = 0.0\nsamples = 1319\nvalue = [0, 2111]\nnclass = No')]

```



```

In [48]: print("Linear:",lis)
          print("Lasso:",las)
          print("Ridge:",rrs)
          print("ElasticNet:",ens)
          print("Logistic:",los)
          print("Random Forest:",rfcs)

```

```

Linear: 0.7877914082809474
Lasso: 0.6608359189741109
Ridge: 0.7877953739336674
ElasticNet: 0.7744562684843173
Logistic: 0.5802718795640939
Random Forest: 0.9163617103235747

```

Best Model is Random Forest

2002

```
In [49]: df2=pd.read_csv("madrid_2002.csv")
df2
```

```
Out[49]:
```

	date	BEN	CO	EBE	MXV	NMHC	NO_2	NOx	OXY	O_3	PM10
0	2002-04-01 01:00:00	NaN	1.39	NaN	NaN	NaN	145.100006	352.100006	NaN	6.54	41.990002
1	2002-04-01 01:00:00	1.93	0.71	2.33	6.20	0.15	98.150002	153.399994	2.67	6.85	20.980000
2	2002-04-01 01:00:00	NaN	0.80	NaN	NaN	NaN	103.699997	134.000000	NaN	13.01	28.440001
3	2002-04-01 01:00:00	NaN	1.61	NaN	NaN	NaN	97.599998	268.000000	NaN	5.12	42.180000
4	2002-04-01 01:00:00	NaN	1.90	NaN	NaN	NaN	92.089996	237.199997	NaN	7.28	76.330002
...
217291	2002-11-01 00:00:00	4.16	1.14	NaN	NaN	NaN	81.080002	265.700012	NaN	7.21	36.750000
217292	2002-11-01 00:00:00	3.67	1.73	2.89	NaN	0.38	113.900002	373.100006	NaN	5.66	63.389999
217293	2002-11-01 00:00:00	1.37	0.58	1.17	2.37	0.15	65.389999	107.699997	1.30	9.11	9.640000
217294	2002-11-01 00:00:00	4.51	0.91	4.83	10.99	NaN	149.800003	202.199997	1.00	5.75	NaN
217295	2002-11-01 00:00:00	3.11	1.17	3.00	7.77	0.26	80.110001	180.300003	2.25	7.38	29.240000

217296 rows × 16 columns




```
In [50]: df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 217296 entries, 0 to 217295
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   date        217296 non-null object
 1   BEN         66747 non-null float64
 2   CO          216637 non-null float64
 3   EBE         58547 non-null float64
 4   MXY         41255 non-null float64
 5   NMHC        87045 non-null float64
 6   NO_2        216439 non-null float64
 7   NOx         216439 non-null float64
 8   OXY         41314 non-null float64
 9   O_3         216726 non-null float64
10   PM10        209113 non-null float64
11   PXY         41256 non-null float64
12   SO_2        216507 non-null float64
13   TCH         87115 non-null float64
14   TOL         66619 non-null float64
15   station     217296 non-null int64
dtypes: float64(14), int64(1), object(1)
memory usage: 26.5+ MB
```

```
In [51]: df3=df2.dropna()  
df3
```

```
Out[51]:
```

	date	BEN	CO	EBE	MXV	NMHC	NO_2	NOx	OXY	O_3	PM10
1	2002-04-01 01:00:00	1.93	0.71	2.33	6.20	0.15	98.150002	153.399994	2.67	6.85	20.980000
5	2002-04-01 01:00:00	3.19	0.72	3.23	7.65	0.11	113.699997	187.000000	3.53	12.37	27.450001
22	2002-04-01 01:00:00	2.02	0.80	1.57	3.66	0.15	93.860001	101.300003	1.77	6.99	33.000000
24	2002-04-01 01:00:00	3.02	1.04	2.43	5.38	0.21	103.699997	195.399994	2.15	14.04	37.310001
26	2002-04-01 02:00:00	2.02	0.53	2.24	5.97	0.12	91.599998	136.199997	2.55	6.76	19.980000
...
217269	2002-10-31 23:00:00	1.24	0.28	1.26	2.64	0.11	60.080002	64.160004	1.23	15.64	13.910000
217271	2002-10-31 23:00:00	3.13	1.30	2.93	7.90	0.28	84.779999	184.000000	2.23	7.94	32.529999
217273	2002-11-01 00:00:00	2.50	0.97	3.63	9.95	0.19	61.759998	132.100006	4.46	5.45	29.500000
217293	2002-11-01 00:00:00	1.37	0.58	1.17	2.37	0.15	65.389999	107.699997	1.30	9.11	9.640000
217295	2002-11-01 00:00:00	3.11	1.17	3.00	7.77	0.26	80.110001	180.300003	2.25	7.38	29.240000

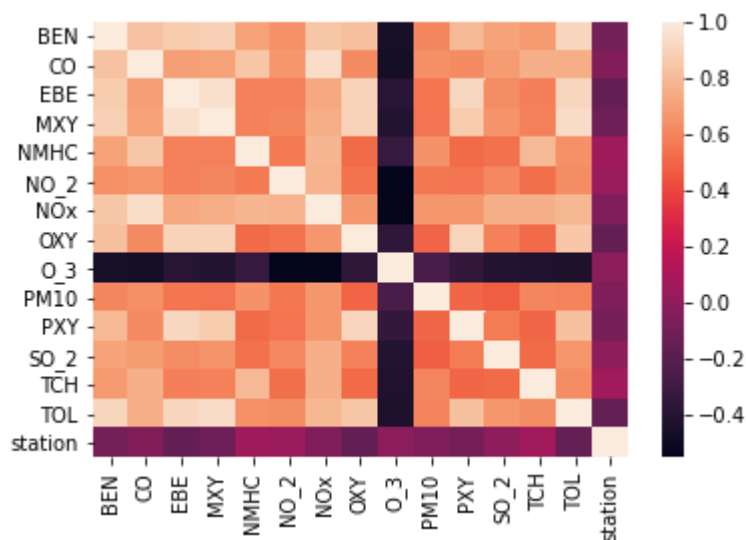
32381 rows × 16 columns



```
In [52]: df3=df3.drop(["date"],axis=1)
```

```
In [53]: sns.heatmap(df3.corr())
```

```
Out[53]: <AxesSubplot:>
```



```
In [54]: x=df3.drop(["TCH"],axis=1)
y=df3["TCH"]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

Linear

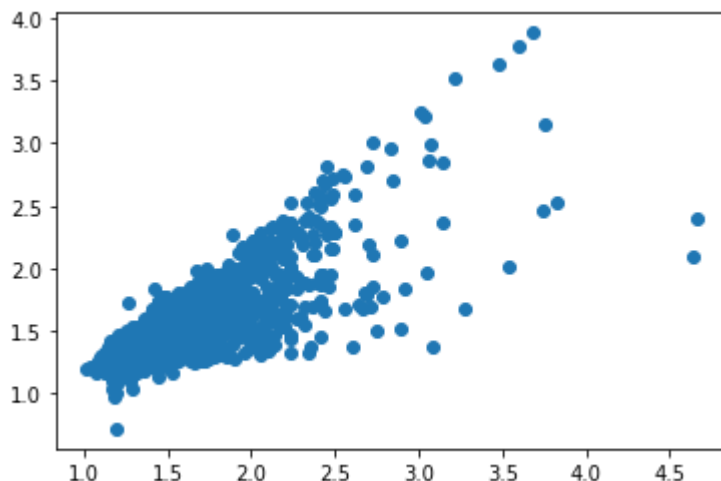
```
In [55]: li=LinearRegression()
li.fit(x_train,y_train)
```

```
Out[55]: LinearRegression()
```

```
In [ ]:
```

```
In [56]: prediction=li.predict(x_test)
plt.scatter(y_test,prediction)
```

```
Out[56]: <matplotlib.collections.PathCollection at 0x23d89e9d4f0>
```



```
In [57]: lis=li.score(x_test,y_test)
```

```
In [58]: df3["TCH"].value_counts()
```

```
Out[58]: 1.29    1318
         1.30    1253
         1.27    1244
         1.28    1232
         1.31    1187
         ...
         2.51     1
         4.66     1
         2.63     1
         3.19     1
         3.34     1
         Name: TCH, Length: 232, dtype: int64
```

```
In [59]: df3.loc[df3["TCH"]<1.40,"TCH"]=1
         df3.loc[df3["TCH"]>1.40,"TCH"]=2
         df3["TCH"].value_counts()
```

```
Out[59]: 1.0    21925
         2.0    10456
         Name: TCH, dtype: int64
```

```
In [ ]:
```

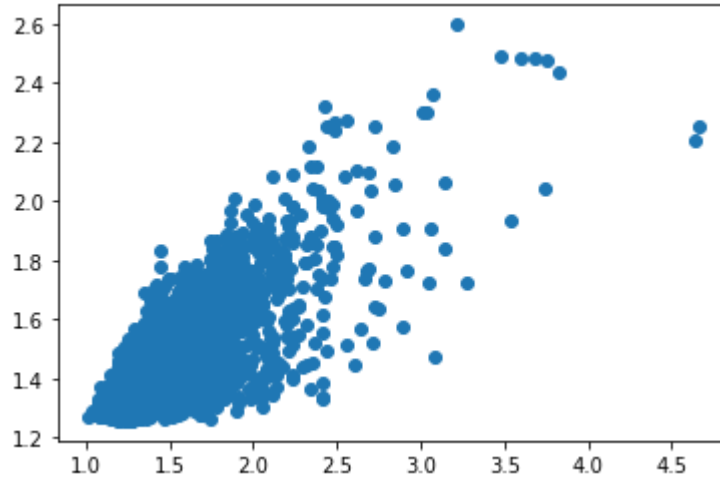
Lasso

```
In [60]: la=Lasso(alpha=5)
         la.fit(x_train,y_train)
```

```
Out[60]: Lasso(alpha=5)
```

```
In [61]: prediction1=la.predict(x_test)
plt.scatter(y_test,prediction1)
```

Out[61]: <matplotlib.collections.PathCollection at 0x23d89ef3e20>



```
In [62]: las=la.score(x_test,y_test)
```

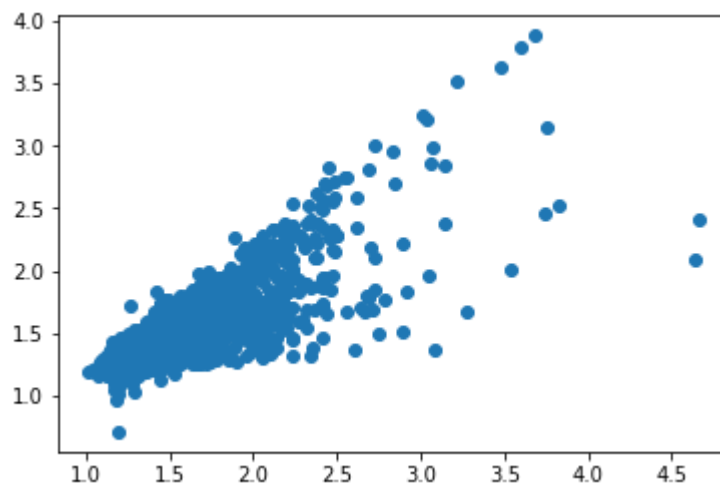
Ridge

```
In [63]: rr=Ridge(alpha=1)
rr.fit(x_train,y_train)
```

Out[63]: Ridge(alpha=1)

```
In [64]: prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[64]: <matplotlib.collections.PathCollection at 0x23d89f59250>



```
In [65]: rrs=rr.score(x_test,y_test)
```

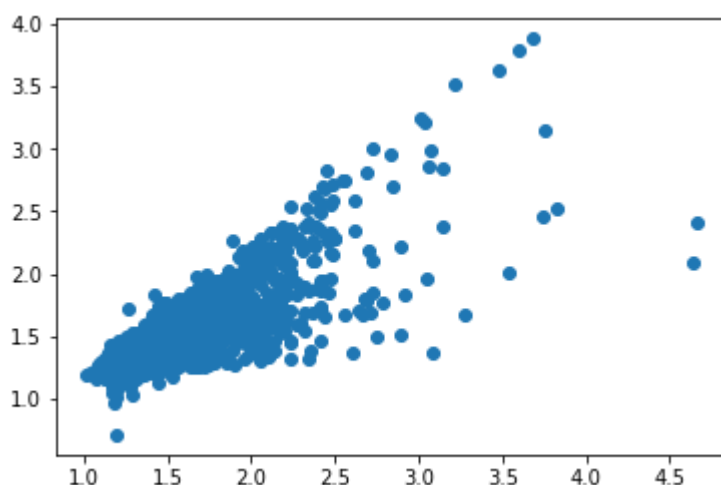
ElasticNet

```
In [66]: en=ElasticNet()
en.fit(x_train,y_train)
```

```
Out[66]: ElasticNet()
```

```
In [67]: prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

```
Out[67]: <matplotlib.collections.PathCollection at 0x23d88c175b0>
```



```
In [68]: ens=en.score(x_test,y_test)
```

```
In [69]: print(rr.score(x_test,y_test))
rr.score(x_train,y_train)
```

```
0.7234772839066235
```

```
Out[69]: 0.7035718884047415
```

Logistic

```
In [75]: g={"TCH":{1.0:"Low",2.0:"High"}}
df3=df3.replace(g)
df3["TCH"].value_counts()
```

```
Out[75]: Low      21925
High      10456
Name: TCH, dtype: int64
```

```
In [76]: x=df3.drop(["TCH"],axis=1)
y=df3["TCH"]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

```
In [77]: lo=LogisticRegression()
lo.fit(x_train,y_train)
```

```
Out[77]: LogisticRegression()
```

```
In [78]: prediction3=lo.predict(x_test)
plt.scatter(y_test,prediction3)
```

```
Out[78]: <matplotlib.collections.PathCollection at 0x23d89413ca0>
```



```
In [80]: los=lo.score(x_test,y_test)
```

Random Forest

```
In [81]: from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
```

```
In [82]: g1={"TCH":{"Low":1.0,"High":2.0}}
df3=df3.replace(g1)
```

```
In [83]: x=df3.drop(["TCH"],axis=1)
y=df3["TCH"]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

```
In [84]: rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

```
Out[84]: RandomForestClassifier()
```

```
In [85]: parameter={
    'max_depth':[1,2,4,5,6],
    'min_samples_leaf':[5,10,15,20,25],
    'n_estimators':[10,20,30,40,50]
}
```

```
In [86]: grid_search=GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

```
Out[86]: GridSearchCV(cv=2, estimator=RandomForestClassifier(),
    param_grid={'max_depth': [1, 2, 4, 5, 6],
    'min_samples_leaf': [5, 10, 15, 20, 25],
    'n_estimators': [10, 20, 30, 40, 50]},
    scoring='accuracy')
```

```
In [87]: rfcs=grid_search.best_score_
```

```
In [88]: rfc_best=grid_search.best_estimator_
```



```
In [89]: from sklearn.tree import plot_tree

plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['Yes','No'])
```

```

Out[89]: [Text(2329.3867924528304, 2019.0857142857144, 'NMHC <= 0.205\ngini = 0.439\nsamples = 14344\nvalue = [15300, 7366]\nclass = Yes'),
Text(1247.6037735849056, 1708.457142857143, 'O_3 <= 14.805\ngini = 0.265\nsamples = 10889\nvalue = [14521, 2702]\nclass = Yes'),
Text(642.2264150943397, 1397.8285714285716, 'CO <= 1.085\ngini = 0.483\nsamples = 2511\nvalue = [2288, 1570]\nclass = Yes'),
Text(336.9056603773585, 1087.2, 'O_3 <= 4.69\ngini = 0.461\nsamples = 1996\nvalue = [1979, 1116]\nclass = Yes'),
Text(168.45283018867926, 776.5714285714287, 'OXY <= 1.665\ngini = 0.492\nsamples = 241\nvalue = [171, 219]\nclass = No'),
Text(84.22641509433963, 465.9428571428573, 'NMHC <= 0.165\ngini = 0.468\nsamples = 77\nvalue = [77, 46]\nclass = Yes'),
Text(42.113207547169814, 155.3142857142857, 'gini = 0.328\nsamples = 53\nvalue = [69, 18]\nclass = Yes'),
Text(126.33962264150944, 155.3142857142857, 'gini = 0.346\nsamples = 24\nvalue = [8, 28]\nclass = No'),
Text(252.67924528301887, 465.9428571428573, 'PM10 <= 15.03\ngini = 0.456\nsamples = 164\nvalue = [94, 173]\nclass = No'),
Text(210.56603773584908, 155.3142857142857, 'gini = 0.487\nsamples = 36\nvalue = [36, 26]\nclass = Yes'),
Text(294.7924528301887, 155.3142857142857, 'gini = 0.406\nsamples = 128\nvalue = [58, 147]\nclass = No'),
Text(505.35849056603774, 776.5714285714287, 'PM10 <= 33.975\ngini = 0.443\nsamples = 1755\nvalue = [1808, 897]\nclass = Yes'),
Text(421.13207547169816, 465.9428571428573, 'station <= 28079068.0\ngini = 0.398\nsamples = 1292\nvalue = [1442, 544]\nclass = Yes'),
Text(379.0188679245283, 155.3142857142857, 'gini = 0.317\nsamples = 975\nvalue = [1213, 299]\nclass = Yes'),
Text(463.24528301886795, 155.3142857142857, 'gini = 0.499\nsamples = 317\nvalue = [229, 245]\nclass = No'),
Text(589.5849056603774, 465.9428571428573, 'O_3 <= 12.105\ngini = 0.5\nsamples = 463\nvalue = [366, 353]\nclass = Yes'),
Text(547.4716981132076, 155.3142857142857, 'gini = 0.497\nsamples = 337\nvalue = [242, 281]\nclass = No'),
Text(631.6981132075472, 155.3142857142857, 'gini = 0.465\nsamples = 126\nvalue = [124, 72]\nclass = Yes'),
Text(947.5471698113208, 1087.2, 'NOx <= 309.25\ngini = 0.482\nsamples = 515\nvalue = [309, 454]\nclass = No'),
Text(842.2641509433963, 776.5714285714287, 'station <= 28079068.0\ngini = 0.495\nsamples = 453\nvalue = [298, 366]\nclass = No'),
Text(758.0377358490566, 465.9428571428573, 'PM10 <= 35.595\ngini = 0.5\nsamples = 400\nvalue = [288, 299]\nclass = No'),
Text(715.9245283018869, 155.3142857142857, 'gini = 0.469\nsamples = 189\nvalue = [178, 107]\nclass = Yes'),
Text(800.1509433962265, 155.3142857142857, 'gini = 0.463\nsamples = 211\nvalue = [110, 192]\nclass = No'),
Text(926.4905660377359, 465.9428571428573, 'NMHC <= 0.175\ngini = 0.226\nsamples = 53\nvalue = [10, 67]\nclass = No'),
Text(884.377358490566, 155.3142857142857, 'gini = 0.381\nsamples = 27\nvalue = [10, 29]\nclass = No'),
Text(968.6037735849058, 155.3142857142857, 'gini = 0.0\nsamples = 26\nvalue = [0, 38]\nclass = No'),
Text(1052.8301886792453, 776.5714285714287, 'PM10 <= 46.755\ngini = 0.198\nsamples = 62\nvalue = [11, 88]\nclass = No'),
Text(1010.7169811320755, 465.9428571428573, 'gini = 0.42\nsamples = 21\nvalue = [9, 21]\nclass = No'),
Text(1094.9433962264152, 465.9428571428573, 'PM10 <= 63.715\ngini = 0.056\nsamples = 10\nvalue = [10, 0]\nclass = No')

```

```

amples = 41\nvalue = [2, 67]\nclass = No'),
  Text(1052.8301886792453, 155.3142857142857, 'gini = 0.128\nsamples = 20\nvalue = [2, 27]\nclass = No'),
  Text(1137.056603773585, 155.3142857142857, 'gini = 0.0\nsamples = 21\nvalue = [0, 40]\nclass = No'),
  Text(1852.9811320754718, 1397.8285714285716, 'NO_2 <= 61.33\ngini = 0.155\nsamples = 8378\nvalue = [12233, 1132]\nclass = Yes'),
  Text(1516.0754716981132, 1087.2, 'NMHC <= 0.125\ngini = 0.089\nsamples = 6151\nvalue = [9353, 457]\nclass = Yes'),
  Text(1347.622641509434, 776.5714285714287, 'EBE <= 0.885\ngini = 0.051\nsamples = 4867\nvalue = [7535, 201]\nclass = Yes'),
  Text(1263.3962264150944, 465.9428571428573, 'TOL <= 2.285\ngini = 0.012\nsamples = 1703\nvalue = [2617, 16]\nclass = Yes'),
  Text(1221.2830188679245, 155.3142857142857, 'gini = 0.0\nsamples = 1018\nvalue = [1572, 0]\nclass = Yes'),
  Text(1305.5094339622642, 155.3142857142857, 'gini = 0.03\nsamples = 685\nvalue = [1045, 16]\nclass = Yes'),
  Text(1431.8490566037738, 465.9428571428573, 'PM10 <= 38.695\ngini = 0.07\nsamples = 3164\nvalue = [4918, 185]\nclass = Yes'),
  Text(1389.735849056604, 155.3142857142857, 'gini = 0.058\nsamples = 2683\nvalue = [4164, 128]\nclass = Yes'),
  Text(1473.9622641509436, 155.3142857142857, 'gini = 0.131\nsamples = 481\nvalue = [754, 57]\nclass = Yes'),
  Text(1684.5283018867926, 776.5714285714287, 'BEN <= 2.305\ngini = 0.216\nsamples = 1284\nvalue = [1818, 256]\nclass = Yes'),
  Text(1600.301886792453, 465.9428571428573, 'NOx <= 37.255\ngini = 0.161\nsamples = 1141\nvalue = [1699, 164]\nclass = Yes'),
  Text(1558.188679245283, 155.3142857142857, 'gini = 0.02\nsamples = 423\nvalue = [691, 7]\nclass = Yes'),
  Text(1642.4150943396228, 155.3142857142857, 'gini = 0.233\nsamples = 718\nvalue = [1008, 157]\nclass = Yes'),
  Text(1768.754716981132, 465.9428571428573, 'NO_2 <= 35.18\ngini = 0.492\nsamples = 143\nvalue = [119, 92]\nclass = Yes'),
  Text(1726.6415094339625, 155.3142857142857, 'gini = 0.147\nsamples = 20\nvalue = [2, 23]\nclass = No'),
  Text(1810.867924528302, 155.3142857142857, 'gini = 0.467\nsamples = 123\nvalue = [117, 69]\nclass = Yes'),
  Text(2189.8867924528304, 1087.2, 'station <= 28079068.0\ngini = 0.308\nsamples = 2227\nvalue = [2880, 675]\nclass = Yes'),
  Text(2021.433962264151, 776.5714285714287, 'PM10 <= 37.83\ngini = 0.248\nsamples = 1647\nvalue = [2223, 377]\nclass = Yes'),
  Text(1937.2075471698115, 465.9428571428573, 'SO_2 <= 5.01\ngini = 0.162\nsamples = 1024\nvalue = [1453, 142]\nclass = Yes'),
  Text(1895.0943396226417, 155.3142857142857, 'gini = 0.485\nsamples = 56\nvalue = [51, 36]\nclass = Yes'),
  Text(1979.3207547169814, 155.3142857142857, 'gini = 0.131\nsamples = 968\nvalue = [1402, 106]\nclass = Yes'),
  Text(2105.6603773584907, 465.9428571428573, 'NOx <= 195.6\ngini = 0.358\nsamples = 623\nvalue = [770, 235]\nclass = Yes'),
  Text(2063.547169811321, 155.3142857142857, 'gini = 0.314\nsamples = 537\nvalue = [689, 167]\nclass = Yes'),
  Text(2147.7735849056608, 155.3142857142857, 'gini = 0.496\nsamples = 86\nvalue = [81, 68]\nclass = Yes'),
  Text(2358.33962264151, 776.5714285714287, 'TOL <= 8.115\ngini = 0.429\nsamples = 580\nvalue = [657, 298]\nclass = Yes'),
  Text(2274.11320754717, 465.9428571428573, 'MXV <= 2.98\ngini = 0.203\nsamples = 170\nvalue = [240, 31]\nclass = Yes'),

```

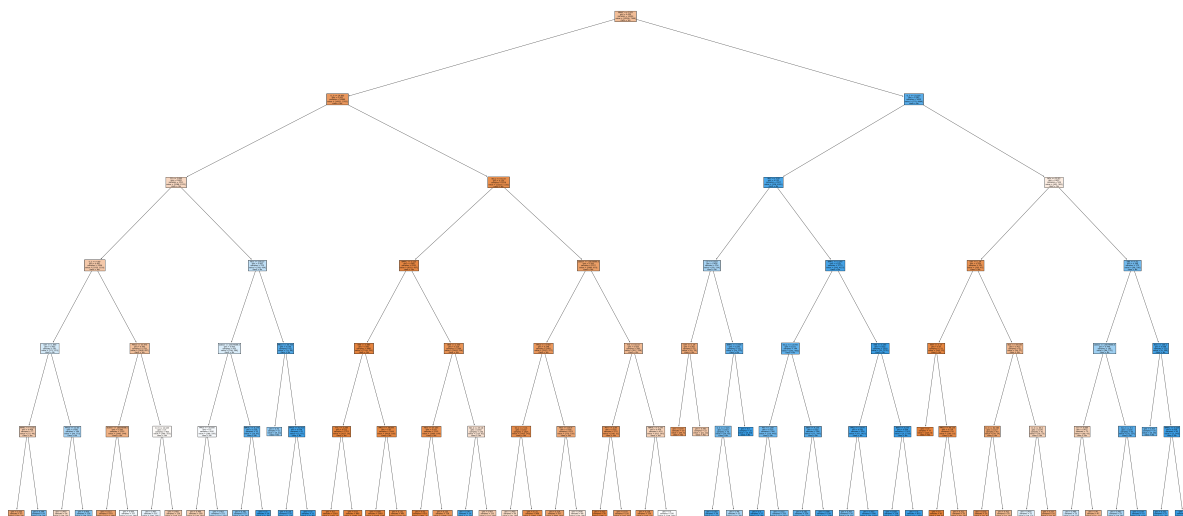
```
Text(2232.0, 155.3142857142857, 'gini = 0.022\nsamples = 50\nvalue = [88, 1]\nnclass = Yes'),
Text(2316.2264150943397, 155.3142857142857, 'gini = 0.275\nsamples = 120\nvalue = [152, 30]\nnclass = Yes'),
Text(2442.566037735849, 465.9428571428573, 'NMHC <= 0.145\ngini = 0.476\nsamples = 410\nvalue = [417, 267]\nnclass = Yes'),
Text(2400.4528301886794, 155.3142857142857, 'gini = 0.228\nsamples = 120\nvalue = [179, 27]\nnclass = Yes'),
Text(2484.679245283019, 155.3142857142857, 'gini = 0.5\nsamples = 290\nvalue = [238, 240]\nnclass = No'),
Text(3411.169811320755, 1708.457142857143, 'O_3 <= 33.295\ngini = 0.245\nsamples = 3455\nvalue = [779, 4664]\nnclass = No'),
Text(2884.7547169811323, 1397.8285714285716, 'MXY <= 3.59\ngini = 0.132\nsamples = 2927\nvalue = [328, 4281]\nnclass = No'),
Text(2653.1320754716985, 1087.2, 'CO <= 0.625\ngini = 0.458\nsamples = 154\nvalue = [85, 154]\nnclass = No'),
Text(2568.905660377359, 776.5714285714287, 'OXY <= 1.29\ngini = 0.362\nsamples = 52\nvalue = [61, 19]\nnclass = Yes'),
Text(2526.7924528301887, 465.9428571428573, 'gini = 0.254\nsamples = 30\nvalue = [40, 7]\nnclass = Yes'),
Text(2611.0188679245284, 465.9428571428573, 'gini = 0.463\nsamples = 22\nvalue = [21, 12]\nnclass = Yes'),
Text(2737.3584905660377, 776.5714285714287, 'NMHC <= 0.305\ngini = 0.256\nsamples = 102\nvalue = [24, 135]\nnclass = No'),
Text(2695.245283018868, 465.9428571428573, 'O_3 <= 7.315\ngini = 0.343\nsamples = 70\nvalue = [24, 85]\nnclass = No'),
Text(2653.1320754716985, 155.3142857142857, 'gini = 0.163\nsamples = 38\nvalue = [5, 51]\nnclass = No'),
Text(2737.3584905660377, 155.3142857142857, 'gini = 0.46\nsamples = 32\nvalue = [19, 34]\nnclass = No'),
Text(2779.471698113208, 465.9428571428573, 'gini = 0.0\nsamples = 32\nvalue = [0, 50]\nnclass = No'),
Text(3116.377358490566, 1087.2, 'NMHC <= 0.245\ngini = 0.105\nsamples = 2773\nvalue = [243, 4127]\nnclass = No'),
Text(2947.924528301887, 776.5714285714287, 'SO_2 <= 13.705\ngini = 0.271\nsamples = 744\nvalue = [191, 992]\nnclass = No'),
Text(2863.6981132075475, 465.9428571428573, 'PXY <= 1.69\ngini = 0.335\nsamples = 360\nvalue = [120, 445]\nnclass = No'),
Text(2821.5849056603774, 155.3142857142857, 'gini = 0.0\nsamples = 27\nvalue = [0, 38]\nnclass = No'),
Text(2905.811320754717, 155.3142857142857, 'gini = 0.352\nsamples = 333\nvalue = [120, 407]\nnclass = No'),
Text(3032.1509433962265, 465.9428571428573, 'BEN <= 4.175\ngini = 0.203\nsamples = 384\nvalue = [71, 547]\nnclass = No'),
Text(2990.037735849057, 155.3142857142857, 'gini = 0.262\nsamples = 226\nvalue = [59, 321]\nnclass = No'),
Text(3074.2641509433965, 155.3142857142857, 'gini = 0.096\nsamples = 158\nvalue = [12, 226]\nnclass = No'),
Text(3284.8301886792456, 776.5714285714287, 'EBE <= 5.265\ngini = 0.032\nsamples = 2029\nvalue = [52, 3135]\nnclass = No'),
Text(3200.603773584906, 465.9428571428573, 'PM10 <= 18.44\ngini = 0.062\nsamples = 926\nvalue = [47, 1417]\nnclass = No'),
Text(3158.4905660377362, 155.3142857142857, 'gini = 0.367\nsamples = 20\nvalue = [8, 25]\nnclass = No'),
Text(3242.7169811320755, 155.3142857142857, 'gini = 0.053\nsamples = 906\nvalue = [39, 1392]\nnclass = No'),
Text(3369.0566037735853, 465.9428571428573, 'BEN <= 5.275\ngini = 0.006\nsam
```

```

ples = 1103\nvalue = [5, 1718]\nclass = No'),
  Text(3326.943396226415, 155.3142857142857, 'gini = 0.026\nsamples = 228\nvalue = [5, 376]\nclass = No'),
  Text(3411.169811320755, 155.3142857142857, 'gini = 0.0\nsamples = 875\nvalue = [0, 1342]\nclass = No'),
  Text(3937.5849056603774, 1397.8285714285716, 'NOx <= 92.93\ngini = 0.497\nsamples = 528\nvalue = [451, 383]\nclass = Yes'),
  Text(3642.7924528301887, 1087.2, 'OXY <= 1.34\ngini = 0.206\nsamples = 253\nvalue = [355, 47]\nclass = Yes'),
  Text(3495.3962264150946, 776.5714285714287, 'MXV <= 1.17\ngini = 0.07\nsamples = 172\nvalue = [265, 10]\nclass = Yes'),
  Text(3453.283018867925, 465.9428571428573, 'gini = 0.0\nsamples = 77\nvalue = [125, 0]\nclass = Yes'),
  Text(3537.509433962264, 465.9428571428573, 'PM10 <= 46.125\ngini = 0.124\nsamples = 95\nvalue = [140, 10]\nclass = Yes'),
  Text(3495.3962264150946, 155.3142857142857, 'gini = 0.0\nsamples = 62\nvalue = [98, 0]\nclass = Yes'),
  Text(3579.6226415094343, 155.3142857142857, 'gini = 0.311\nsamples = 33\nvalue = [42, 10]\nclass = Yes'),
  Text(3790.1886792452833, 776.5714285714287, 'NOx <= 59.535\ngini = 0.413\nsamples = 81\nvalue = [90, 37]\nclass = Yes'),
  Text(3705.9622641509436, 465.9428571428573, 'O_3 <= 62.585\ngini = 0.327\nsamples = 40\nvalue = [54, 14]\nclass = Yes'),
  Text(3663.849056603774, 155.3142857142857, 'gini = 0.202\nsamples = 20\nvalue = [31, 4]\nclass = Yes'),
  Text(3748.0754716981137, 155.3142857142857, 'gini = 0.422\nsamples = 20\nvalue = [23, 10]\nclass = Yes'),
  Text(3874.415094339623, 465.9428571428573, 'O_3 <= 66.0\ngini = 0.476\nsamples = 41\nvalue = [36, 23]\nclass = Yes'),
  Text(3832.301886792453, 155.3142857142857, 'gini = 0.495\nsamples = 21\nvalue = [14, 17]\nclass = No'),
  Text(3916.5283018867926, 155.3142857142857, 'gini = 0.337\nsamples = 20\nvalue = [22, 6]\nclass = Yes'),
  Text(4232.377358490567, 1087.2, 'EBE <= 4.52\ngini = 0.346\nsamples = 275\nvalue = [96, 336]\nclass = No'),
  Text(4127.094339622642, 776.5714285714287, 'station <= 28079068.0\ngini = 0.449\nsamples = 165\nvalue = [88, 170]\nclass = No'),
  Text(4042.867924528302, 465.9428571428573, 'CO <= 0.985\ngini = 0.475\nsamples = 70\nvalue = [68, 43]\nclass = Yes'),
  Text(4000.7547169811323, 155.3142857142857, 'gini = 0.488\nsamples = 23\nvalue = [16, 22]\nclass = No'),
  Text(4084.981132075472, 155.3142857142857, 'gini = 0.41\nsamples = 47\nvalue = [52, 21]\nclass = Yes'),
  Text(4211.320754716981, 465.9428571428573, 'SO_2 <= 9.5\ngini = 0.235\nsamples = 95\nvalue = [20, 127]\nclass = No'),
  Text(4169.207547169812, 155.3142857142857, 'gini = 0.465\nsamples = 25\nvalue = [14, 24]\nclass = No'),
  Text(4253.433962264151, 155.3142857142857, 'gini = 0.104\nsamples = 70\nvalue = [6, 103]\nclass = No'),
  Text(4337.660377358491, 776.5714285714287, 'NOx <= 146.1\ngini = 0.088\nsamples = 110\nvalue = [8, 166]\nclass = No'),
  Text(4295.5471698113215, 465.9428571428573, 'gini = 0.305\nsamples = 20\nvalue = [6, 26]\nclass = No'),
  Text(4379.773584905661, 465.9428571428573, 'NMHC <= 0.225\ngini = 0.028\nsamples = 90\nvalue = [2, 140]\nclass = No'),
  Text(4337.660377358491, 155.3142857142857, 'gini = 0.102\nsamples = 27\nvalue = [2, 35]\nclass = No'),

```

```
Text(4421.88679245283, 155.3142857142857, 'gini = 0.0\nsamples = 63\nvalue = [0, 105]\nnclass = No')]
```



```
In [90]: print("Linear:",lis)
print("Lasso:",las)
print("Ridge:",rrs)
print("ElasticNet:",ens)
print("Logistic:",los)
print("Random Forest:",rfcs)
```

```
Linear: 0.7234015760409874
Lasso: 0.5477773816966718
Ridge: 0.7234772839066235
ElasticNet: 0.6100678624001161
Logistic: 0.687596500257334
Random Forest: 0.8931439159975294
```

Best model is Random Forest