

2003

```
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_2003.csv")
df
```

```
Out[2]:
```

	date	BEN	CO	EBE	MXV	NMHC	NO_2	NOx	OXY	O_3	PM
0	2003-03-01 01:00:00	NaN	1.72	NaN	NaN	NaN	73.900002	316.299988	NaN	10.550000	55.2099
1	2003-03-01 01:00:00	NaN	1.45	NaN	NaN	0.26	72.110001	250.000000	0.73	6.720000	52.3899
2	2003-03-01 01:00:00	NaN	1.57	NaN	NaN	NaN	80.559998	224.199997	NaN	21.049999	63.2400
3	2003-03-01 01:00:00	NaN	2.45	NaN	NaN	NaN	78.370003	450.399994	NaN	4.220000	67.8399
4	2003-03-01 01:00:00	NaN	3.26	NaN	NaN	NaN	96.250000	479.100006	NaN	8.460000	95.7799
...
243979	2003-10-01 00:00:00	0.20	0.16	2.01	3.17	0.02	31.799999	32.299999	1.68	34.049999	7.3800
243980	2003-10-01 00:00:00	0.32	0.08	0.36	0.72	NaN	10.450000	14.760000	1.00	34.610001	7.4000
243981	2003-10-01 00:00:00	NaN	NaN	NaN	NaN	0.07	34.639999	50.810001	NaN	32.160000	16.8300
243982	2003-10-01 00:00:00	NaN	NaN	NaN	NaN	0.07	32.580002	41.020000	NaN	NaN	13.5700
243983	2003-10-01 00:00:00	1.00	0.29	2.15	6.41	0.07	37.150002	56.849998	2.28	21.480000	12.3500

243984 rows × 16 columns



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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 243984 entries, 0 to 243983
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   date        243984 non-null object
 1   BEN         69745 non-null float64
 2   CO          225340 non-null float64
 3   EBE         61244 non-null float64
 4   MXY         42045 non-null float64
 5   NMHC        111951 non-null float64
 6   NO_2        242625 non-null float64
 7   NOx         242629 non-null float64
 8   OXY         42072 non-null float64
 9   O_3         234131 non-null float64
10   PM10        240896 non-null float64
11   PXY         42063 non-null float64
12   SO_2        242729 non-null float64
13   TCH         111991 non-null float64
14   TOL         69439 non-null float64
15   station     243984 non-null int64
dtypes: float64(14), int64(1), object(1)
memory usage: 29.8+ MB
```

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

```
Out[4]:
```

	date	BEN	CO	EBE	MXV	NMHC	NO_2	NOx	OXY	O_3	I
5	2003-03-01 01:00:00	8.41	1.94	9.83	21.49	0.45	90.300003	384.899994	9.48	9.950000	95.15
23	2003-03-01 01:00:00	3.46	1.27	3.43	7.08	0.18	54.250000	173.300003	3.37	6.540000	53.00
27	2003-03-01 01:00:00	6.39	1.79	5.75	10.88	0.33	75.459999	281.100006	3.68	6.690000	63.84
33	2003-03-01 02:00:00	7.42	1.47	10.63	24.73	0.35	83.309998	277.200012	11.00	9.900000	58.86
51	2003-03-01 02:00:00	3.62	1.29	3.20	7.08	0.19	42.209999	166.300003	3.41	6.380000	47.59
...
243955	2003-09-30 23:00:00	1.75	0.41	3.07	9.38	0.09	46.290001	77.709999	3.11	18.280001	7.52
243957	2003-10-01 00:00:00	2.35	0.60	3.88	10.86	0.11	61.240002	133.100006	0.89	10.900000	10.24
243961	2003-10-01 00:00:00	2.97	0.82	4.53	10.88	0.05	36.529999	131.300003	5.52	12.940000	25.66
243979	2003-10-01 00:00:00	0.20	0.16	2.01	3.17	0.02	31.799999	32.299999	1.68	34.049999	7.36
243983	2003-10-01 00:00:00	1.00	0.29	2.15	6.41	0.07	37.150002	56.849998	2.28	21.480000	12.35

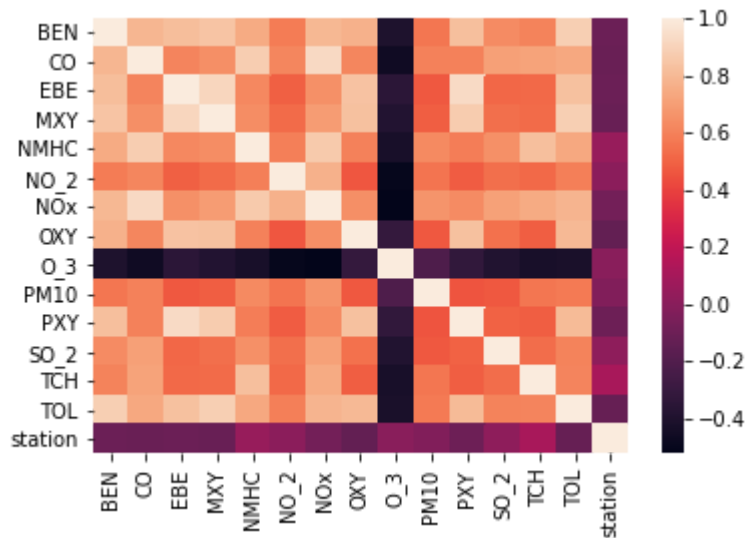
33010 rows × 16 columns



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

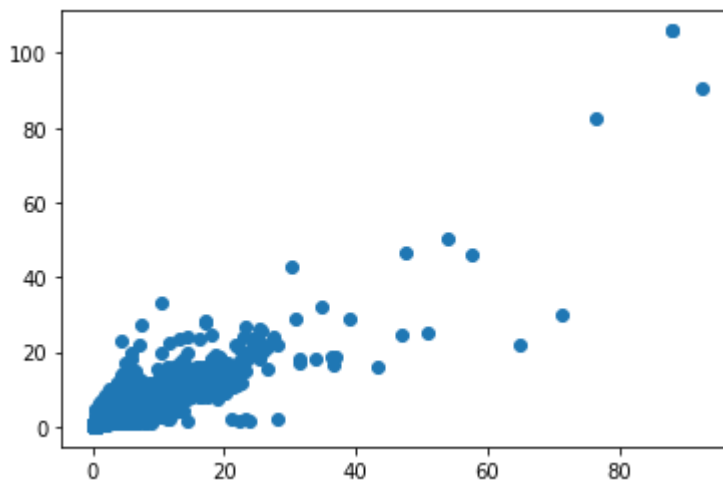
```
In [6]: sns.heatmap(df1.corr())
```

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



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

```
Out[7]: [<matplotlib.lines.Line2D at 0x255505b0490>]
```



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

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

```
In [10]: 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)
```

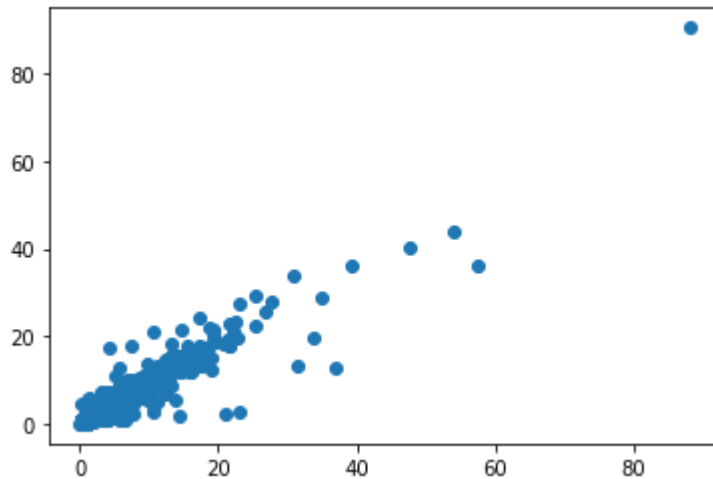
Linear

```
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 0x25550a8dee0>
```



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

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

```
Out[14]: 1.30    1344
1.31    1342
1.32    1281
1.27    1279
1.29    1262
...
3.50      1
3.87      1
3.21      1
3.14      1
1.01      1
Name: TCH, Length: 243, 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    21614
2.0    11396
Name: TCH, dtype: int64
```

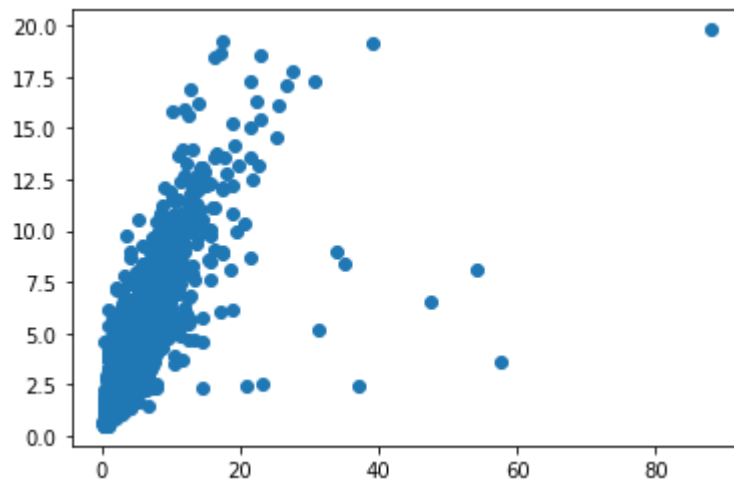
Lasso

```
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 0x25550acbc70>



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

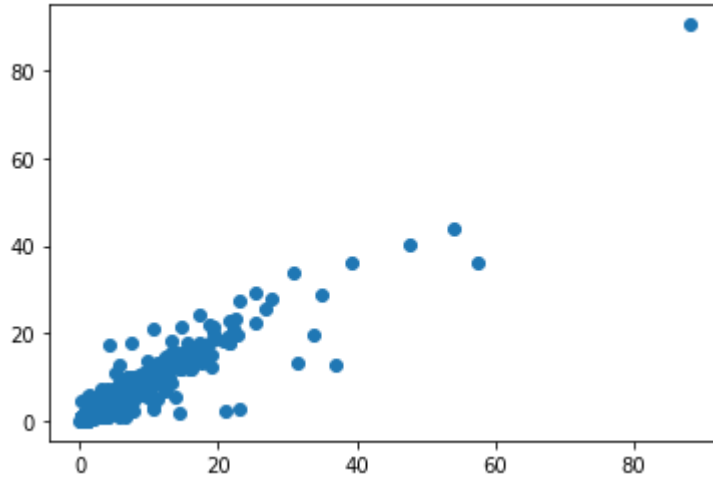
Ridge

```
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 0x255504ca8e0>



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

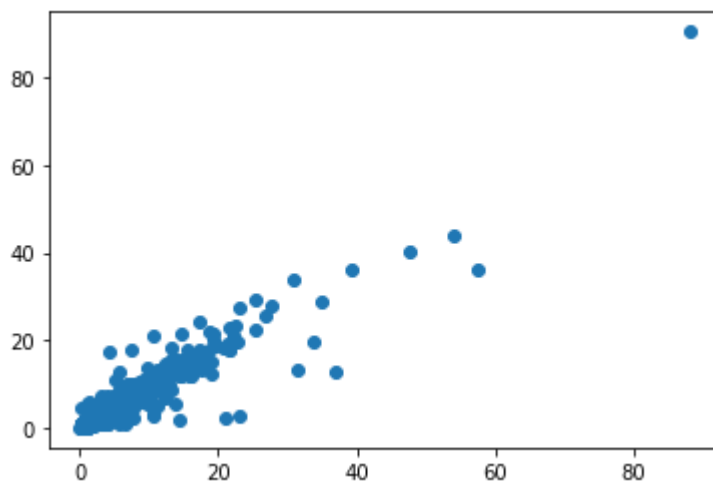
ElasticNet

```
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 0x25551ce3cd0>



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

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

0.9195176980660054

Out[25]: 0.9131593670000937

Logistic

```
In [26]: g={"TCH":{1.0:"Low",2.0:"High"}}  
df1=df1.replace(g)  
df1["TCH"].value_counts()
```

Out[26]: Low 21614
High 11396
Name: TCH, dtype: int64

```
In [27]: 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 [28]: lo=LogisticRegression()  
lo.fit(x_train,y_train)
```

Out[28]: LogisticRegression()

```
In [29]: prediction3=lo.predict(x_test)  
plt.scatter(y_test,prediction3)
```

Out[29]: <matplotlib.collections.PathCollection at 0x25551986520>



```
In [30]: los=lo.score(x_test,y_test)
```


Random Forest

```
In [31]: from sklearn.ensemble import RandomForestClassifier
        from sklearn.model_selection import GridSearchCV
```

```
In [32]: g1={"TCH":{"Low":1.0,"High":2.0}}
        df1=df1.replace(g1)
```

```
In [33]: 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 [34]: rfc=RandomForestClassifier()
        rfc.fit(x_train,y_train)
```

```
Out[34]: RandomForestClassifier()
```

```
In [35]: parameter={
        'max_depth':[1,2,4,5,6],
        'min_samples_leaf':[5,10,15,20,25],
        'n_estimators':[10,20,30,40,50]
        }
```

```
In [36]: grid_search=GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accuracy")
        grid_search.fit(x_train,y_train)
```

```
Out[36]: 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 [37]: rfcs=grid_search.best_score_
```

```
In [38]: rfc_best=grid_search.best_estimator_
```

```
In [39]: 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[39]: [Text(2184.6696428571427, 2019.0857142857144, 'NO_2 <= 66.475\ngini = 0.449\n
samples = 14618\nvalue = [15233, 7874]\nnclass = Yes'),
Text(1111.017857142857, 1708.457142857143, 'TOL <= 11.125\ngini = 0.284\nsam
ples = 9483\nvalue = [12409, 2565]\nnclass = Yes'),
Text(548.0357142857142, 1397.8285714285716, 'CO <= 0.805\ngini = 0.205\nsamp
les = 7987\nvalue = [11131, 1459]\nnclass = Yes'),
Text(318.85714285714283, 1087.2, 'NO_2 <= 50.625\ngini = 0.171\nsamples = 74
86\nvalue = [10704, 1113]\nnclass = Yes'),
Text(159.42857142857142, 776.5714285714287, 'O_3 <= 13.135\ngini = 0.11\nsam
ples = 5665\nvalue = [8412, 520]\nnclass = Yes'),
Text(79.71428571428571, 465.9428571428573, 'PM10 <= 13.925\ngini = 0.478\nsa
mples = 271\nvalue = [265, 173]\nnclass = Yes'),
Text(39.857142857142854, 155.3142857142857, 'gini = 0.246\nsamples = 84\nval
ue = [119, 20]\nnclass = Yes'),
Text(119.57142857142856, 155.3142857142857, 'gini = 0.5\nsamples = 187\nvalu
e = [146, 153]\nnclass = No'),
Text(239.1428571428571, 465.9428571428573, 'MXV <= 1.645\ngini = 0.078\nsamp
les = 5394\nvalue = [8147, 347]\nnclass = Yes'),
Text(199.28571428571428, 155.3142857142857, 'gini = 0.033\nsamples = 2671\nv
alue = [4175, 71]\nnclass = Yes'),
Text(279.0, 155.3142857142857, 'gini = 0.122\nsamples = 2723\nvalue = [3972,
276]\nnclass = Yes'),
Text(478.2857142857142, 776.5714285714287, 'NOx <= 107.65\ngini = 0.327\nsam
ples = 1821\nvalue = [2292, 593]\nnclass = Yes'),
Text(398.57142857142856, 465.9428571428573, 'NMHC <= 0.125\ngini = 0.3\nsamp
les = 1527\nvalue = [1984, 446]\nnclass = Yes'),
Text(358.71428571428567, 155.3142857142857, 'gini = 0.185\nsamples = 1132\nv
alue = [1626, 187]\nnclass = Yes'),
Text(438.4285714285714, 155.3142857142857, 'gini = 0.487\nsamples = 395\nval
ue = [358, 259]\nnclass = Yes'),
Text(558.0, 465.9428571428573, 'BEN <= 1.94\ngini = 0.437\nsamples = 294\nva
lue = [308, 147]\nnclass = Yes'),
Text(518.1428571428571, 155.3142857142857, 'gini = 0.465\nsamples = 201\nval
ue = [202, 117]\nnclass = Yes'),
Text(597.8571428571428, 155.3142857142857, 'gini = 0.344\nsamples = 93\nvalu
e = [106, 30]\nnclass = Yes'),
Text(777.2142857142857, 1087.2, 'NOx <= 48.14\ngini = 0.495\nsamples = 501\n
value = [427, 346]\nnclass = Yes'),
Text(677.5714285714286, 776.5714285714287, 'OXY <= 0.545\ngini = 0.068\nsamp
les = 74\nvalue = [110, 4]\nnclass = Yes'),
Text(637.7142857142857, 465.9428571428573, 'gini = 0.245\nsamples = 15\nvalu
e = [24, 4]\nnclass = Yes'),
Text(717.4285714285713, 465.9428571428573, 'gini = 0.0\nsamples = 59\nvalue
= [86, 0]\nnclass = Yes'),
Text(876.8571428571428, 776.5714285714287, 'SO_2 <= 11.135\ngini = 0.499\nsa
mples = 427\nvalue = [317, 342]\nnclass = No'),
Text(797.1428571428571, 465.9428571428573, 'station <= 28079015.0\ngini = 0.
456\nsamples = 243\nvalue = [133, 246]\nnclass = No'),
Text(757.2857142857142, 155.3142857142857, 'gini = 0.203\nsamples = 46\nvalu
e = [54, 7]\nnclass = Yes'),
Text(836.9999999999999, 155.3142857142857, 'gini = 0.373\nsamples = 197\nval
ue = [79, 239]\nnclass = No'),
Text(956.5714285714284, 465.9428571428573, 'NMHC <= 0.185\ngini = 0.451\nsam
ples = 184\nvalue = [184, 96]\nnclass = Yes'),
Text(916.7142857142857, 155.3142857142857, 'gini = 0.284\nsamples = 138\nval
ue = [174, 36]\nnclass = Yes'),
Text(996.4285714285713, 155.3142857142857, 'gini = 0.245\nsamples = 46\nvalu

```

```

e = [10, 60]\n\nclass = No'),
  Text(1673.9999999999998, 1397.8285714285716, 'NMHC <= 0.155\n\ngini = 0.497\n\nsamples = 1496\n\nvalue = [1278, 1106]\n\nclass = Yes'),
  Text(1355.142857142857, 1087.2, 'OXY <= 3.16\n\ngini = 0.296\n\nsamples = 845\n\nvalue = [1080, 238]\n\nclass = Yes'),
  Text(1195.7142857142856, 776.5714285714287, 'NOx <= 101.4\n\ngini = 0.418\n\nsamples = 314\n\nvalue = [350, 148]\n\nclass = Yes'),
  Text(1116.0, 465.9428571428573, 'O_3 <= 13.42\n\ngini = 0.309\n\nsamples = 186\n\nvalue = [241, 57]\n\nclass = Yes'),
  Text(1076.142857142857, 155.3142857142857, 'gini = 0.481\n\nsamples = 53\n\nvalue = [55, 37]\n\nclass = Yes'),
  Text(1155.8571428571427, 155.3142857142857, 'gini = 0.175\n\nsamples = 133\n\nvalue = [186, 20]\n\nclass = Yes'),
  Text(1275.4285714285713, 465.9428571428573, 'station <= 28079015.0\n\ngini = 0.496\n\nsamples = 128\n\nvalue = [109, 91]\n\nclass = Yes'),
  Text(1235.5714285714284, 155.3142857142857, 'gini = 0.26\n\nsamples = 31\n\nvalue = [44, 8]\n\nclass = Yes'),
  Text(1315.2857142857142, 155.3142857142857, 'gini = 0.493\n\nsamples = 97\n\nvalue = [65, 83]\n\nclass = No'),
  Text(1514.5714285714284, 776.5714285714287, 'O_3 <= 13.285\n\ngini = 0.195\n\nsamples = 531\n\nvalue = [730, 90]\n\nclass = Yes'),
  Text(1434.8571428571427, 465.9428571428573, 'O_3 <= 7.215\n\ngini = 0.471\n\nsamples = 85\n\nvalue = [80, 49]\n\nclass = Yes'),
  Text(1395.0, 155.3142857142857, 'gini = 0.484\n\nsamples = 27\n\nvalue = [16, 23]\n\nclass = No'),
  Text(1474.7142857142856, 155.3142857142857, 'gini = 0.411\n\nsamples = 58\n\nvalue = [64, 26]\n\nclass = Yes'),
  Text(1594.2857142857142, 465.9428571428573, 'station <= 28079068.0\n\ngini = 0.112\n\nsamples = 446\n\nvalue = [650, 41]\n\nclass = Yes'),
  Text(1554.4285714285713, 155.3142857142857, 'gini = 0.052\n\nsamples = 408\n\nvalue = [616, 17]\n\nclass = Yes'),
  Text(1634.142857142857, 155.3142857142857, 'gini = 0.485\n\nsamples = 38\n\nvalue = [34, 24]\n\nclass = Yes'),
  Text(1992.8571428571427, 1087.2, 'NOx <= 185.2\n\ngini = 0.302\n\nsamples = 651\n\nvalue = [198, 868]\n\nclass = No'),
  Text(1833.4285714285713, 776.5714285714287, 'station <= 28079015.0\n\ngini = 0.413\n\nsamples = 377\n\nvalue = [182, 442]\n\nclass = No'),
  Text(1753.7142857142856, 465.9428571428573, 'TOL <= 28.06\n\ngini = 0.474\n\nsamples = 133\n\nvalue = [135, 85]\n\nclass = Yes'),
  Text(1713.8571428571427, 155.3142857142857, 'gini = 0.447\n\nsamples = 113\n\nvalue = [126, 64]\n\nclass = Yes'),
  Text(1793.5714285714284, 155.3142857142857, 'gini = 0.42\n\nsamples = 20\n\nvalue = [9, 21]\n\nclass = No'),
  Text(1913.1428571428569, 465.9428571428573, 'MXV <= 7.41\n\ngini = 0.206\n\nsamples = 244\n\nvalue = [47, 357]\n\nclass = No'),
  Text(1873.2857142857142, 155.3142857142857, 'gini = 0.255\n\nsamples = 137\n\nvalue = [35, 198]\n\nclass = No'),
  Text(1952.9999999999998, 155.3142857142857, 'gini = 0.131\n\nsamples = 107\n\nvalue = [12, 159]\n\nclass = No'),
  Text(2152.285714285714, 776.5714285714287, 'PXY <= 3.875\n\ngini = 0.07\n\nsamples = 274\n\nvalue = [16, 426]\n\nclass = No'),
  Text(2072.5714285714284, 465.9428571428573, 'O_3 <= 13.895\n\ngini = 0.173\n\nsamples = 91\n\nvalue = [14, 132]\n\nclass = No'),
  Text(2032.7142857142856, 155.3142857142857, 'gini = 0.049\n\nsamples = 76\n\nvalue = [3, 117]\n\nclass = No'),
  Text(2112.428571428571, 155.3142857142857, 'gini = 0.488\n\nsamples = 15\n\nvalue = [11, 15]\n\nclass = No'),

```

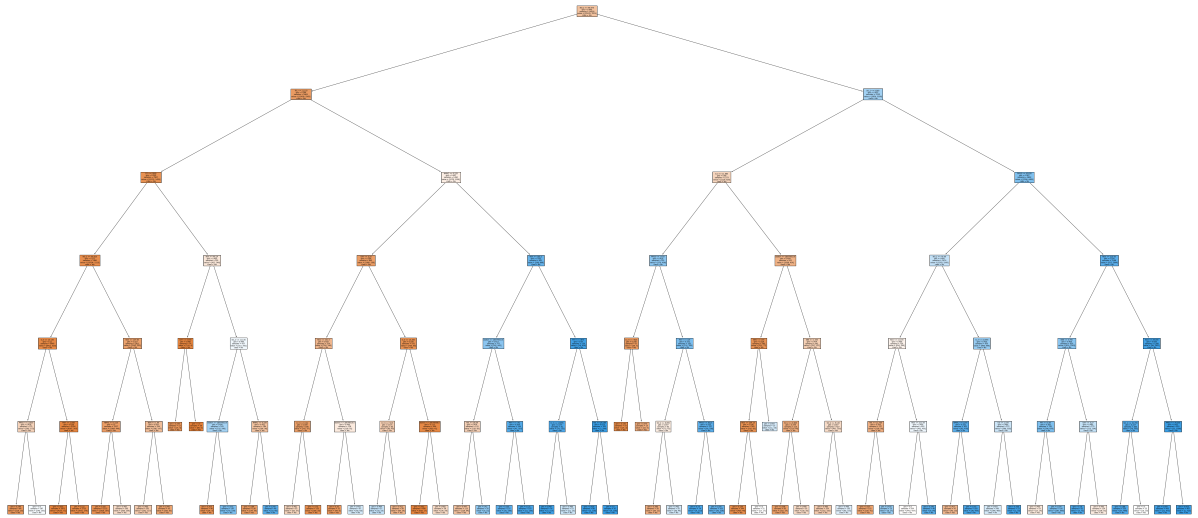
```
Text(2232.0, 465.9428571428573, 'TOL <= 26.165\ngini = 0.013\nsamples = 183\nvalue = [2, 294]\nclass = No'),
Text(2192.142857142857, 155.3142857142857, 'gini = 0.029\nsamples = 86\nvalue = [2, 136]\nclass = No'),
Text(2271.8571428571427, 155.3142857142857, 'gini = 0.0\nsamples = 97\nvalue = [0, 158]\nclass = No'),
Text(3258.3214285714284, 1708.457142857143, 'SO_2 <= 9.585\ngini = 0.453\nsamples = 5135\nvalue = [2824, 5309]\nclass = No'),
Text(2690.3571428571427, 1397.8285714285716, 'O_3 <= 11.885\ngini = 0.48\nsamples = 1273\nvalue = [1228, 820]\nclass = Yes'),
Text(2451.2142857142853, 1087.2, 'PM10 <= 17.93\ngini = 0.416\nsamples = 351\nvalue = [170, 406]\nclass = No'),
Text(2351.5714285714284, 776.5714285714287, 'O_3 <= 7.625\ngini = 0.258\nsamples = 31\nvalue = [39, 7]\nclass = Yes'),
Text(2311.7142857142853, 465.9428571428573, 'gini = 0.142\nsamples = 16\nvalue = [24, 2]\nclass = Yes'),
Text(2391.428571428571, 465.9428571428573, 'gini = 0.375\nsamples = 15\nvalue = [15, 5]\nclass = Yes'),
Text(2550.8571428571427, 776.5714285714287, 'MXV <= 3.29\ngini = 0.372\nsamples = 320\nvalue = [131, 399]\nclass = No'),
Text(2471.142857142857, 465.9428571428573, 'SO_2 <= 6.895\ngini = 0.497\nsamples = 83\nvalue = [72, 61]\nclass = Yes'),
Text(2431.285714285714, 155.3142857142857, 'gini = 0.386\nsamples = 28\nvalue = [34, 12]\nclass = Yes'),
Text(2511.0, 155.3142857142857, 'gini = 0.492\nsamples = 55\nvalue = [38, 49]\nclass = No'),
Text(2630.5714285714284, 465.9428571428573, 'PM10 <= 36.21\ngini = 0.253\nsamples = 237\nvalue = [59, 338]\nclass = No'),
Text(2590.7142857142853, 155.3142857142857, 'gini = 0.14\nsamples = 74\nvalue = [9, 110]\nclass = No'),
Text(2670.428571428571, 155.3142857142857, 'gini = 0.295\nsamples = 163\nvalue = [50, 228]\nclass = No'),
Text(2929.5, 1087.2, 'station <= 28079015.0\ngini = 0.404\nsamples = 922\nvalue = [1058, 414]\nclass = Yes'),
Text(2829.8571428571427, 776.5714285714287, 'BEN <= 5.59\ngini = 0.224\nsamples = 380\nvalue = [521, 77]\nclass = Yes'),
Text(2790.0, 465.9428571428573, 'NOx <= 254.15\ngini = 0.196\nsamples = 365\nvalue = [510, 63]\nclass = Yes'),
Text(2750.142857142857, 155.3142857142857, 'gini = 0.156\nsamples = 344\nvalue = [493, 46]\nclass = Yes'),
Text(2829.8571428571427, 155.3142857142857, 'gini = 0.5\nsamples = 21\nvalue = [17, 17]\nclass = Yes'),
Text(2869.7142857142853, 465.9428571428573, 'gini = 0.493\nsamples = 15\nvalue = [11, 14]\nclass = No'),
Text(3029.142857142857, 776.5714285714287, 'EBE <= 1.435\ngini = 0.474\nsamples = 542\nvalue = [537, 337]\nclass = Yes'),
Text(2949.428571428571, 465.9428571428573, 'SO_2 <= 8.805\ngini = 0.378\nsamples = 115\nvalue = [130, 44]\nclass = Yes'),
Text(2909.5714285714284, 155.3142857142857, 'gini = 0.308\nsamples = 85\nvalue = [102, 24]\nclass = Yes'),
Text(2989.285714285714, 155.3142857142857, 'gini = 0.486\nsamples = 30\nvalue = [28, 20]\nclass = Yes'),
Text(3108.8571428571427, 465.9428571428573, 'SO_2 <= 9.145\ngini = 0.487\nsamples = 427\nvalue = [407, 293]\nclass = Yes'),
Text(3068.9999999999995, 155.3142857142857, 'gini = 0.468\nsamples = 324\nvalue = [322, 192]\nclass = Yes'),
Text(3148.7142857142853, 155.3142857142857, 'gini = 0.496\nsamples = 103\nvalue = [103, 0]\nclass = Yes')
```

```

lue = [85, 101]\n\nclass = No'),
Text(3826.2857142857138, 1397.8285714285716, 'PM10 <= 43.635\ngini = 0.387\n
samples = 3862\n\nclass = No'),
Text(3507.428571428571, 1087.2, 'NO_2 <= 76.99\ngini = 0.484\n\samples = 1674
\n\nclass = No'),
Text(3347.9999999999995, 776.5714285714287, 'NOx <= 111.55\ngini = 0.499\n\sam
ples = 739\n\nclass = Yes'),
Text(3268.285714285714, 465.9428571428573, 'TOL <= 11.755\ngini = 0.393\n\sam
ples = 137\n\nclass = Yes'),
Text(3228.428571428571, 155.3142857142857, 'gini = 0.337\n\samples = 122\n\nclass = Yes'),
Text(3308.142857142857, 155.3142857142857, 'gini = 0.435\n\samples = 15\n\nclass = No'),
Text(3427.7142857142853, 465.9428571428573, 'NOx <= 211.65\ngini = 0.499\n\sam
ples = 602\n\nclass = No'),
Text(3387.8571428571427, 155.3142857142857, 'gini = 0.5\n\samples = 554\n\nclass = Yes'),
Text(3467.5714285714284, 155.3142857142857, 'gini = 0.245\n\samples = 48\n\nclass = No'),
Text(3666.8571428571427, 776.5714285714287, 'O_3 <= 10.695\ngini = 0.435\n\sam
ples = 935\n\nclass = No'),
Text(3587.142857142857, 465.9428571428573, 'NMHC <= 0.145\ngini = 0.252\n\sam
ples = 363\n\nclass = No'),
Text(3547.285714285714, 155.3142857142857, 'gini = 0.417\n\samples = 47\n\nclass = Yes'),
Text(3626.9999999999995, 155.3142857142857, 'gini = 0.141\n\samples = 316\n\nclass = No'),
Text(3746.5714285714284, 465.9428571428573, 'CO <= 1.575\ngini = 0.489\n\sam
ples = 572\n\nclass = No'),
Text(3706.7142857142853, 155.3142857142857, 'gini = 0.493\n\samples = 549\n\nclass = No'),
Text(3786.428571428571, 155.3142857142857, 'gini = 0.108\n\samples = 23\n\nclass = No'),
Text(4145.142857142857, 1087.2, 'NOx <= 215.95\ngini = 0.254\n\samples = 2188
\n\nclass = No'),
Text(3985.7142857142853, 776.5714285714287, 'EBE <= 4.795\ngini = 0.409\n\sam
ples = 920\n\nclass = No'),
Text(3905.9999999999995, 465.9428571428573, 'PXY <= 0.97\ngini = 0.382\n\sam
ples = 745\n\nclass = No'),
Text(3866.142857142857, 155.3142857142857, 'gini = 0.495\n\samples = 19\n\nclass = Yes'),
Text(3945.8571428571427, 155.3142857142857, 'gini = 0.374\n\samples = 726\n\nclass = No'),
Text(4065.428571428571, 465.9428571428573, 'OXY <= 5.27\ngini = 0.485\n\sam
ples = 175\n\nclass = No'),
Text(4025.5714285714284, 155.3142857142857, 'gini = 0.16\n\samples = 51\n\nclass = No'),
Text(4105.285714285714, 155.3142857142857, 'gini = 0.496\n\samples = 124\n\nclass = Yes'),
Text(4304.571428571428, 776.5714285714287, 'NOx <= 261.85\ngini = 0.093\n\sam
ples = 1268\n\nclass = No'),
Text(4224.857142857142, 465.9428571428573, 'SO_2 <= 32.705\ngini = 0.227\n\sam
ples = 358\n\nclass = No'),
Text(4185.0, 155.3142857142857, 'gini = 0.188\n\samples = 335\n\nclass = No'),
Text(4264.714285714285, 155.3142857142857, 'gini = 0.5\n\samples = 23\n\nclass = Yes'),

```

```
Text(4384.285714285714, 465.9428571428573, 'PM10 <= 76.335\ngini = 0.031\nsamples = 910\nvalue = [22, 1396]\nclass = No'),
Text(4344.428571428571, 155.3142857142857, 'gini = 0.059\nsamples = 443\nvalue = [21, 673]\nclass = No'),
Text(4424.142857142857, 155.3142857142857, 'gini = 0.003\nsamples = 467\nvalue = [1, 723]\nclass = No')]
```



```
In [40]: print("Linear:", lis)
print("Lasso:", las)
print("Ridge:", rrs)
print("ElasticNet:", ens)
print("Logistic:", los)
print("Random Forest:", rfcs)
```

```
Linear: 0.9195103306695078
Lasso: 0.6724055165668984
Ridge: 0.9195176980660054
ElasticNet: 0.8935055393696031
Logistic: 0.6567706755528627
Random Forest: 0.8821569724921972
```

Best Model is Ridge Regression

2004

```
In [41]: df2=pd.read_csv("madrid_2004.csv")
df2
```

```
Out[41]:
```

	date	BEN	CO	EBE	MXV	NMHC	NO_2	NOx	OXY	O_3	P
0	2004-08-01 01:00:00	NaN	0.66	NaN	NaN	NaN	89.550003	118.900002	NaN	40.020000	39.990000
1	2004-08-01 01:00:00	2.66	0.54	2.99	6.08	0.18	51.799999	53.860001	3.28	51.689999	22.950000
2	2004-08-01 01:00:00	NaN	1.02	NaN	NaN	NaN	93.389999	138.600006	NaN	20.860001	49.480000
3	2004-08-01 01:00:00	NaN	0.53	NaN	NaN	NaN	87.290001	105.000000	NaN	36.730000	31.070000
4	2004-08-01 01:00:00	NaN	0.17	NaN	NaN	NaN	34.910000	35.349998	NaN	86.269997	54.080000
...
245491	2004-06-01 00:00:00	0.75	0.21	0.85	1.55	0.07	59.580002	64.389999	0.66	33.029999	30.900000
245492	2004-06-01 00:00:00	2.49	0.75	2.44	4.57	NaN	97.139999	146.899994	2.34	7.740000	37.680000
245493	2004-06-01 00:00:00	NaN	NaN	NaN	NaN	0.13	102.699997	132.600006	NaN	17.809999	22.840000
245494	2004-06-01 00:00:00	NaN	NaN	NaN	NaN	0.09	82.599998	102.599998	NaN	NaN	45.630000
245495	2004-06-01 00:00:00	3.01	0.67	2.78	5.12	0.20	92.550003	141.000000	2.60	11.460000	24.380000

245496 rows × 17 columns




```
In [42]: df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 245496 entries, 0 to 245495
Data columns (total 17 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   date        245496 non-null object
 1   BEN         65158 non-null float64
 2   CO          226043 non-null float64
 3   EBE         56781 non-null float64
 4   MXY         39867 non-null float64
 5   NMHC        107630 non-null float64
 6   NO_2        243280 non-null float64
 7   NOx         243283 non-null float64
 8   OXY         39882 non-null float64
 9   O_3         233811 non-null float64
10   PM10        234655 non-null float64
11   PM25        58145 non-null float64
12   PXY         39891 non-null float64
13   SO_2        243402 non-null float64
14   TCH         107650 non-null float64
15   TOL         64914 non-null float64
16   station     245496 non-null int64
dtypes: float64(15), int64(1), object(1)
memory usage: 31.8+ MB
```

```
In [43]: df3=df2.dropna()  
df3
```

```
Out[43]:
```

	date	BEN	CO	EBE	MXV	NMHC	NO_2	NOx	OXY	O_3	P
5	2004-08-01 01:00:00	3.24	0.63	5.55	9.72	0.06	103.800003	144.800003	5.04	32.480000	59.110
22	2004-08-01 01:00:00	0.55	0.36	0.54	0.86	0.07	31.980000	32.799999	0.50	79.040001	43.549
26	2004-08-01 01:00:00	1.80	0.46	2.28	4.62	0.21	62.259998	75.470001	2.47	54.419998	46.630
32	2004-08-01 02:00:00	1.94	0.67	3.14	4.91	0.06	113.500000	165.800003	2.56	26.980000	86.930
49	2004-08-01 02:00:00	0.29	0.30	0.47	0.76	0.07	33.919998	34.840000	0.46	75.570000	48.959
...
245463	2004-05-31 23:00:00	0.62	0.08	0.54	0.70	0.04	44.360001	45.450001	0.42	43.419998	19.290
245467	2004-05-31 23:00:00	2.39	0.67	2.49	3.92	0.20	89.809998	132.800003	2.09	14.740000	31.809
245473	2004-06-01 00:00:00	3.72	1.12	4.33	8.79	0.24	113.900002	253.600006	4.51	9.380000	21.219
245491	2004-06-01 00:00:00	0.75	0.21	0.85	1.55	0.07	59.580002	64.389999	0.66	33.029999	30.900
245495	2004-06-01 00:00:00	3.01	0.67	2.78	5.12	0.20	92.550003	141.000000	2.60	11.460000	24.389

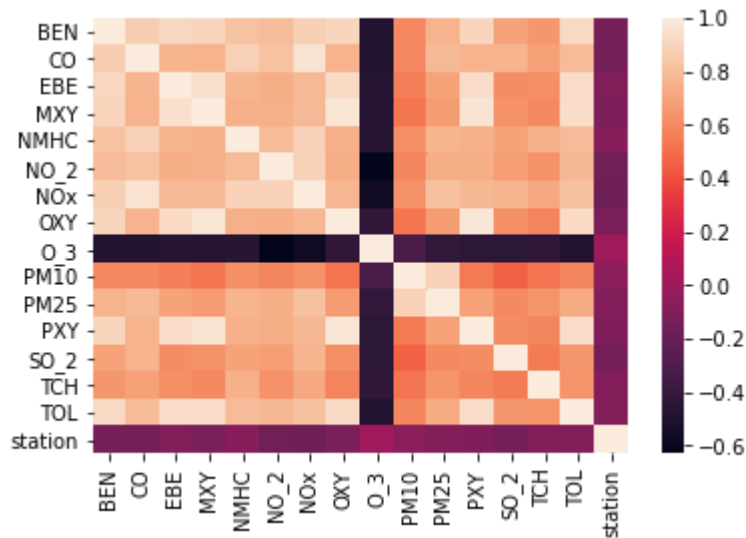
19397 rows × 17 columns



```
In [44]: df3=df3.drop(["date"],axis=1)
```

```
In [45]: sns.heatmap(df3.corr())
```

```
Out[45]: <AxesSubplot:>
```



```
In [46]: 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

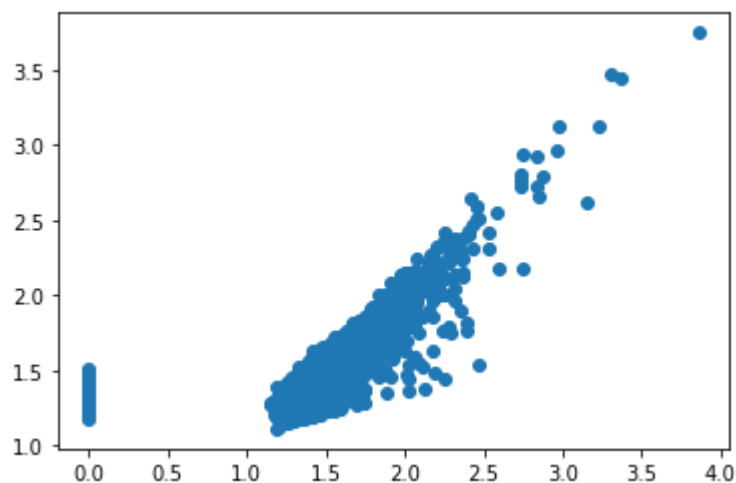
```
In [47]: li=LinearRegression()
li.fit(x_train,y_train)
```

```
Out[47]: LinearRegression()
```

```
In [ ]:
```

```
In [48]: prediction=li.predict(x_test)
plt.scatter(y_test,prediction)
```

```
Out[48]: <matplotlib.collections.PathCollection at 0x25551c2ebb0>
```



```
In [49]: lis=li.score(x_test,y_test)
```

```
In [50]: df3["TCH"].value_counts()
```

```
Out[50]: 1.34    740
         1.33    714
         1.35    708
         1.37    688
         1.36    679
         ...
         2.95     1
         3.65     1
         3.59     1
         2.58     1
         3.86     1
         Name: TCH, Length: 191, dtype: int64
```

```
In [51]: df3.loc[df3["TCH"]<1.40,"TCH"]=1
         df3.loc[df3["TCH"]>1.40,"TCH"]=2
         df3["TCH"].value_counts()
```

```
Out[51]: 1.0    11861
         2.0     7536
         Name: TCH, dtype: int64
```

```
In [ ]:
```

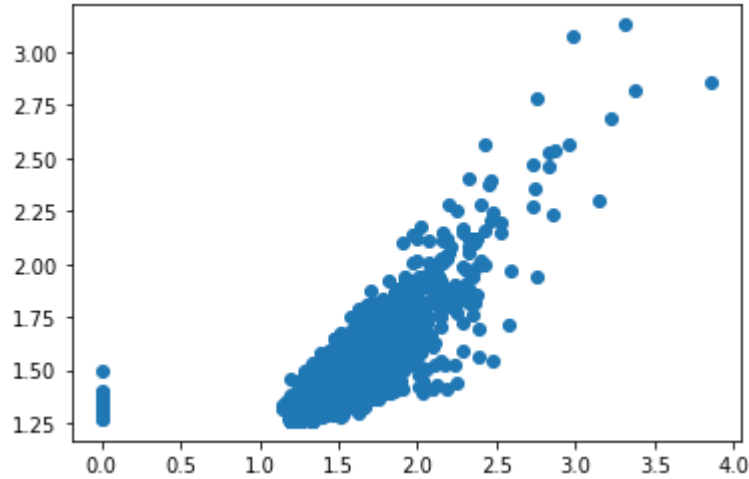
Lasso

```
In [52]: la=Lasso(alpha=5)
         la.fit(x_train,y_train)
```

```
Out[52]: Lasso(alpha=5)
```

```
In [53]: prediction1=la.predict(x_test)
plt.scatter(y_test,prediction1)
```

Out[53]: <matplotlib.collections.PathCollection at 0x25551d39cd0>



```
In [54]: las=la.score(x_test,y_test)
```

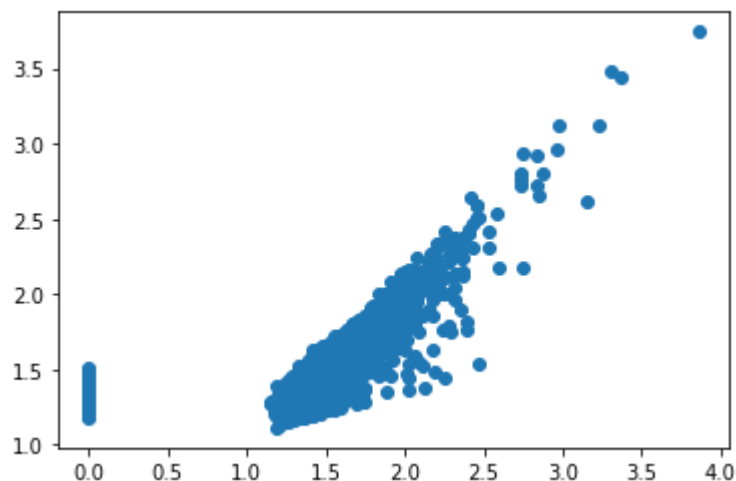
Ridge

```
In [55]: rr=Ridge(alpha=1)
rr.fit(x_train,y_train)
```

Out[55]: Ridge(alpha=1)

```
In [56]: prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[56]: <matplotlib.collections.PathCollection at 0x25551da22e0>



```
In [57]: rrs=rr.score(x_test,y_test)
```

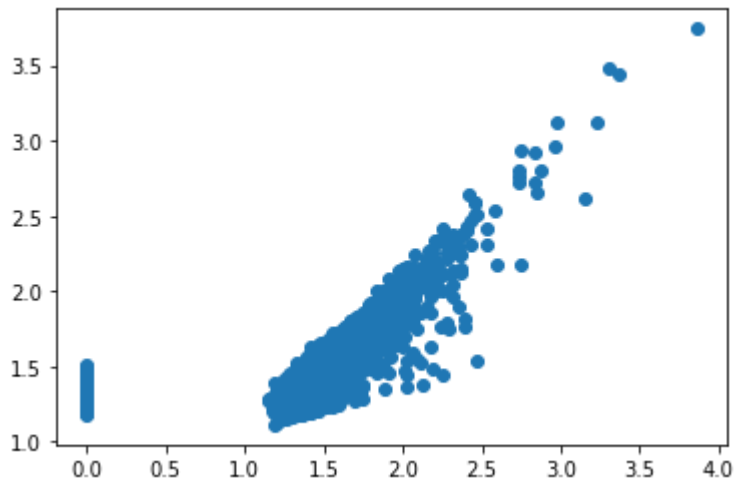
ElasticNet

```
In [58]: en=ElasticNet()
en.fit(x_train,y_train)
```

Out[58]: ElasticNet()

```
In [59]: prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[59]: <matplotlib.collections.PathCollection at 0x25551df9df0>



```
In [60]: ens=en.score(x_test,y_test)
```

```
In [61]: print(rr.score(x_test,y_test))
rr.score(x_train,y_train)
```

0.5759339018894761

Out[61]: 0.5960803646994859

Logistic

```
In [62]: g={"TCH":{1.0:"Low",2.0:"High"}}
df3=df3.replace(g)
df3["TCH"].value_counts()
```

Out[62]: Low 11861
High 7536
Name: TCH, dtype: int64

```
In [63]: 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 [64]: lo=LogisticRegression()
lo.fit(x_train,y_train)
```

```
Out[64]: LogisticRegression()
```

```
In [65]: prediction3=lo.predict(x_test)
plt.scatter(y_test,prediction3)
```

```
Out[65]: <matplotlib.collections.PathCollection at 0x25551710670>
```



```
In [66]: los=lo.score(x_test,y_test)
```

Random Forest

```
In [67]: from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
```

```
In [68]: g1={"TCH":{"Low":1.0,"High":2.0}}
df3=df3.replace(g1)
```

```
In [69]: 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 [70]: rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

```
Out[70]: RandomForestClassifier()
```

```
In [71]: parameter={
    'max_depth':[1,2,4,5,6],
    'min_samples_leaf':[5,10,15,20,25],
    'n_estimators':[10,20,30,40,50]
}
```

```
In [72]: grid_search=GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

```
Out[72]: 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 [73]: rfcs=grid_search.best_score_
```

```
In [74]: rfc_best=grid_search.best_estimator_
```



```
In [75]: 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[75]: [Text(2292.129310344828, 2019.0857142857144, 'TOL <= 9.985\ngini = 0.476\nsam
ples = 8605\nvalue = [8279, 5298]\n\nclass = Yes'),
Text(1207.3965517241381, 1708.457142857143, 'NOx <= 142.25\ngini = 0.25\nsam
ples = 5526\nvalue = [7419, 1270]\n\nclass = Yes'),
Text(615.7241379310345, 1397.8285714285716, 'BEN <= 1.445\ngini = 0.183\nsam
ples = 5027\nvalue = [7112, 809]\n\nclass = Yes'),
Text(307.86206896551727, 1087.2, 'NO_2 <= 40.515\ngini = 0.112\nsamples = 37
00\nvalue = [5505, 347]\n\nclass = Yes'),
Text(153.93103448275863, 776.5714285714287, 'PM10 <= 36.955\ngini = 0.039\ns
amples = 2435\nvalue = [3796, 77]\n\nclass = Yes'),
Text(76.96551724137932, 465.9428571428573, 'EBE <= 1.125\ngini = 0.025\nsamps
les = 2088\nvalue = [3237, 41]\n\nclass = Yes'),
Text(38.48275862068966, 155.3142857142857, 'gini = 0.013\nsamples = 1663\nv
alue = [2601, 17]\n\nclass = Yes'),
Text(115.44827586206898, 155.3142857142857, 'gini = 0.07\nsamples = 425\nv
alue = [636, 24]\n\nclass = Yes'),
Text(230.89655172413796, 465.9428571428573, 'NMHC <= 0.165\ngini = 0.114\ns
amples = 347\nvalue = [559, 36]\n\nclass = Yes'),
Text(192.41379310344828, 155.3142857142857, 'gini = 0.08\nsamples = 340\nv
alue = [554, 24]\n\nclass = Yes'),
Text(269.3793103448276, 155.3142857142857, 'gini = 0.415\nsamples = 7\nv
alue = [5, 12]\n\nclass = No'),
Text(461.79310344827593, 776.5714285714287, 'TOL <= 7.055\ngini = 0.236\nsam
ples = 1265\nvalue = [1709, 270]\n\nclass = Yes'),
Text(384.82758620689657, 465.9428571428573, 'PM25 <= 15.78\ngini = 0.202\ns
amples = 1128\nvalue = [1559, 201]\n\nclass = Yes'),
Text(346.3448275862069, 155.3142857142857, 'gini = 0.111\nsamples = 777\nv
alue = [1129, 71]\n\nclass = Yes'),
Text(423.3103448275862, 155.3142857142857, 'gini = 0.357\nsamples = 351\nv
alue = [430, 130]\n\nclass = Yes'),
Text(538.7586206896552, 465.9428571428573, 'SO_2 <= 8.61\ngini = 0.432\nsamps
les = 137\nvalue = [150, 69]\n\nclass = Yes'),
Text(500.2758620689656, 155.3142857142857, 'gini = 0.488\nsamples = 83\nv
alue = [78, 57]\n\nclass = Yes'),
Text(577.2413793103449, 155.3142857142857, 'gini = 0.245\nsamples = 54\nv
alue = [72, 12]\n\nclass = Yes'),
Text(923.5862068965519, 1087.2, 'BEN <= 2.135\ngini = 0.347\nsamples = 1327
\nvalue = [1607, 462]\n\nclass = Yes'),
Text(769.6551724137931, 776.5714285714287, 'NMHC <= 0.145\ngini = 0.322\nsam
ples = 1086\nvalue = [1343, 339]\n\nclass = Yes'),
Text(692.6896551724138, 465.9428571428573, 'O_3 <= 7.96\ngini = 0.237\nsamps
les = 872\nvalue = [1151, 183]\n\nclass = Yes'),
Text(654.2068965517242, 155.3142857142857, 'gini = 0.5\nsamples = 53\nv
alue = [37, 37]\n\nclass = Yes'),
Text(731.1724137931035, 155.3142857142857, 'gini = 0.205\nsamples = 819\nv
alue = [1114, 146]\n\nclass = Yes'),
Text(846.6206896551724, 465.9428571428573, 'OXY <= 1.285\ngini = 0.495\nsamps
les = 214\nvalue = [192, 156]\n\nclass = Yes'),
Text(808.1379310344828, 155.3142857142857, 'gini = 0.278\nsamples = 13\nv
alue = [3, 15]\n\nclass = No'),
Text(885.1034482758621, 155.3142857142857, 'gini = 0.489\nsamples = 201\nv
alue = [189, 141]\n\nclass = Yes'),
Text(1077.5172413793105, 776.5714285714287, 'station <= 28079015.0\ngini =
0.434\nsamples = 241\nvalue = [264, 123]\n\nclass = Yes'),
Text(1000.5517241379312, 465.9428571428573, 'CO <= 0.765\ngini = 0.27\nsamps
les = 118\nvalue = [151, 29]\n\nclass = Yes'),
Text(962.0689655172415, 155.3142857142857, 'gini = 0.208\nsamples = 99\nv
alue = [151, 29]\n\nclass = Yes')

```

```

e = [135, 18]\n\nclass = Yes'),
  Text(1039.0344827586207, 155.3142857142857, 'gini = 0.483\n\ nsamples = 19\n\ nvalue = [16, 11]\n\nclass = Yes'),
  Text(1154.4827586206898, 465.9428571428573, 'NMHC <= 0.165\n\ ngini = 0.496\n\ nsamples = 123\n\ nvalue = [113, 94]\n\nclass = Yes'),
  Text(1116.0, 155.3142857142857, 'gini = 0.469\n\ nsamples = 100\n\ nvalue = [105, 63]\n\nclass = Yes'),
  Text(1192.9655172413793, 155.3142857142857, 'gini = 0.326\n\ nsamples = 23\n\ nvalue = [8, 31]\n\nclass = No'),
  Text(1799.0689655172416, 1397.8285714285716, 'NMHC <= 0.155\n\ ngini = 0.48\n\ nsamples = 499\n\ nvalue = [307, 461]\n\nclass = No'),
  Text(1539.3103448275863, 1087.2, 'CO <= 0.875\n\ ngini = 0.458\n\ nsamples = 233\n\ nvalue = [225, 124]\n\nclass = Yes'),
  Text(1385.3793103448277, 776.5714285714287, 'station <= 28079015.0\n\ ngini = 0.415\n\ nsamples = 187\n\ nvalue = [199, 83]\n\nclass = Yes'),
  Text(1308.4137931034484, 465.9428571428573, 'PXY <= 1.77\n\ ngini = 0.373\n\ nsamples = 159\n\ nvalue = [182, 60]\n\nclass = Yes'),
  Text(1269.9310344827588, 155.3142857142857, 'gini = 0.301\n\ nsamples = 76\n\ nvalue = [93, 21]\n\nclass = Yes'),
  Text(1346.8965517241381, 155.3142857142857, 'gini = 0.424\n\ nsamples = 83\n\ nvalue = [89, 39]\n\nclass = Yes'),
  Text(1462.344827586207, 465.9428571428573, 'SO_2 <= 10.935\n\ ngini = 0.489\n\ nsamples = 28\n\ nvalue = [17, 23]\n\nclass = No'),
  Text(1423.8620689655174, 155.3142857142857, 'gini = 0.36\n\ nsamples = 12\n\ nvalue = [4, 13]\n\nclass = No'),
  Text(1500.8275862068967, 155.3142857142857, 'gini = 0.491\n\ nsamples = 16\n\ nvalue = [13, 10]\n\nclass = Yes'),
  Text(1693.2413793103449, 776.5714285714287, 'CO <= 1.015\n\ ngini = 0.475\n\ nsamples = 46\n\ nvalue = [26, 41]\n\nclass = No'),
  Text(1616.2758620689656, 465.9428571428573, 'TOL <= 5.605\n\ ngini = 0.498\n\ nsamples = 36\n\ nvalue = [25, 28]\n\nclass = No'),
  Text(1577.793103448276, 155.3142857142857, 'gini = 0.346\n\ nsamples = 7\n\ nvalue = [7, 2]\n\nclass = Yes'),
  Text(1654.7586206896553, 155.3142857142857, 'gini = 0.483\n\ nsamples = 29\n\ nvalue = [18, 26]\n\nclass = No'),
  Text(1770.2068965517242, 465.9428571428573, 'PXY <= 1.45\n\ ngini = 0.133\n\ nsamples = 10\n\ nvalue = [1, 13]\n\nclass = No'),
  Text(1731.7241379310346, 155.3142857142857, 'gini = 0.0\n\ nsamples = 5\n\ nvalue = [0, 8]\n\nclass = No'),
  Text(1808.689655172414, 155.3142857142857, 'gini = 0.278\n\ nsamples = 5\n\ nvalue = [1, 5]\n\nclass = No'),
  Text(2058.8275862068967, 1087.2, 'PM10 <= 17.795\n\ ngini = 0.315\n\ nsamples = 26\n\ nvalue = [82, 337]\n\nclass = No'),
  Text(1962.6206896551726, 776.5714285714287, 'OXY <= 2.31\n\ ngini = 0.498\n\ nsamples = 19\n\ nvalue = [14, 16]\n\nclass = No'),
  Text(1924.137931034483, 465.9428571428573, 'NO_2 <= 85.385\n\ ngini = 0.408\n\ nsamples = 13\n\ nvalue = [6, 15]\n\nclass = No'),
  Text(1885.6551724137933, 155.3142857142857, 'gini = 0.198\n\ nsamples = 6\n\ nvalue = [1, 8]\n\nclass = No'),
  Text(1962.6206896551726, 155.3142857142857, 'gini = 0.486\n\ nsamples = 7\n\ nvalue = [5, 7]\n\nclass = No'),
  Text(2001.1034482758623, 465.9428571428573, 'gini = 0.198\n\ nsamples = 6\n\ nvalue = [8, 1]\n\nclass = Yes'),
  Text(2155.034482758621, 776.5714285714287, 'CO <= 0.875\n\ ngini = 0.288\n\ nsamples = 247\n\ nvalue = [68, 321]\n\nclass = No'),
  Text(2078.0689655172414, 465.9428571428573, 'station <= 28079015.0\n\ ngini = 0.459\n\ nsamples = 91\n\ nvalue = [50, 90]\n\nclass = No'),

```

```
Text(2039.5862068965519, 155.3142857142857, 'gini = 0.496\nsamples = 48\nvalue = [42, 35]\nclass = Yes'),
Text(2116.551724137931, 155.3142857142857, 'gini = 0.222\nsamples = 43\nvalue = [8, 55]\nclass = No'),
Text(2232.0, 465.9428571428573, 'NO_2 <= 104.15\ngini = 0.134\nsamples = 156\nvalue = [18, 231]\nclass = No'),
Text(2193.5172413793107, 155.3142857142857, 'gini = 0.099\nsamples = 126\nvalue = [10, 182]\nclass = No'),
Text(2270.4827586206898, 155.3142857142857, 'gini = 0.241\nsamples = 30\nvalue = [8, 49]\nclass = No'),
Text(3376.8620689655177, 1708.457142857143, 'CO <= 0.755\ngini = 0.29\nsamples = 3079\nvalue = [860, 4028]\nclass = No'),
Text(2799.6206896551726, 1397.8285714285716, 'PM10 <= 30.095\ngini = 0.496\nsamples = 856\nvalue = [614, 736]\nclass = No'),
Text(2520.6206896551726, 1087.2, 'PM25 <= 10.46\ngini = 0.467\nsamples = 372\nvalue = [371, 219]\nclass = Yes'),
Text(2424.4137931034484, 776.5714285714287, 'CO <= 0.715\ngini = 0.25\nsamples = 104\nvalue = [140, 24]\nclass = Yes'),
Text(2385.9310344827586, 465.9428571428573, 'O_3 <= 14.655\ngini = 0.212\nsamples = 95\nvalue = [131, 18]\nclass = Yes'),
Text(2347.4482758620693, 155.3142857142857, 'gini = 0.459\nsamples = 10\nvalue = [9, 5]\nclass = Yes'),
Text(2424.4137931034484, 155.3142857142857, 'gini = 0.174\nsamples = 85\nvalue = [122, 13]\nclass = Yes'),
Text(2462.896551724138, 465.9428571428573, 'gini = 0.48\nsamples = 9\nvalue = [9, 6]\nclass = Yes'),
Text(2616.8275862068967, 776.5714285714287, 'station <= 28079015.0\ngini = 0.496\nsamples = 268\nvalue = [231, 195]\nclass = Yes'),
Text(2539.8620689655177, 465.9428571428573, 'NMHC <= 0.145\ngini = 0.424\nsamples = 117\nvalue = [125, 55]\nclass = Yes'),
Text(2501.379310344828, 155.3142857142857, 'gini = 0.276\nsamples = 77\nvalue = [101, 20]\nclass = Yes'),
Text(2578.344827586207, 155.3142857142857, 'gini = 0.483\nsamples = 40\nvalue = [24, 35]\nclass = No'),
Text(2693.7931034482763, 465.9428571428573, 'NOx <= 137.05\ngini = 0.49\nsamples = 151\nvalue = [106, 140]\nclass = No'),
Text(2655.3103448275865, 155.3142857142857, 'gini = 0.499\nsamples = 137\nvalue = [106, 113]\nclass = No'),
Text(2732.2758620689656, 155.3142857142857, 'gini = 0.0\nsamples = 14\nvalue = [0, 27]\nclass = No'),
Text(3078.6206896551726, 1087.2, 'BEN <= 1.75\ngini = 0.435\nsamples = 484\nvalue = [243, 517]\nclass = No'),
Text(2924.689655172414, 776.5714285714287, 'OXY <= 2.485\ngini = 0.495\nsamples = 49\nvalue = [48, 39]\nclass = Yes'),
Text(2847.724137931035, 465.9428571428573, 'NMHC <= 0.11\ngini = 0.346\nsamples = 22\nvalue = [8, 28]\nclass = No'),
Text(2809.241379310345, 155.3142857142857, 'gini = 0.469\nsamples = 6\nvalue = [5, 3]\nclass = Yes'),
Text(2886.206896551724, 155.3142857142857, 'gini = 0.191\nsamples = 16\nvalue = [3, 25]\nclass = No'),
Text(3001.6551724137935, 465.9428571428573, 'SO_2 <= 10.045\ngini = 0.338\nsamples = 27\nvalue = [40, 11]\nclass = Yes'),
Text(2963.1724137931037, 155.3142857142857, 'gini = 0.0\nsamples = 13\nvalue = [22, 0]\nclass = Yes'),
Text(3040.137931034483, 155.3142857142857, 'gini = 0.471\nsamples = 14\nvalue = [18, 11]\nclass = Yes'),
Text(3232.551724137931, 776.5714285714287, 'PM25 <= 18.465\ngini = 0.412\nsamples = 14\nvalue = [18, 11]\nclass = Yes')
```

```

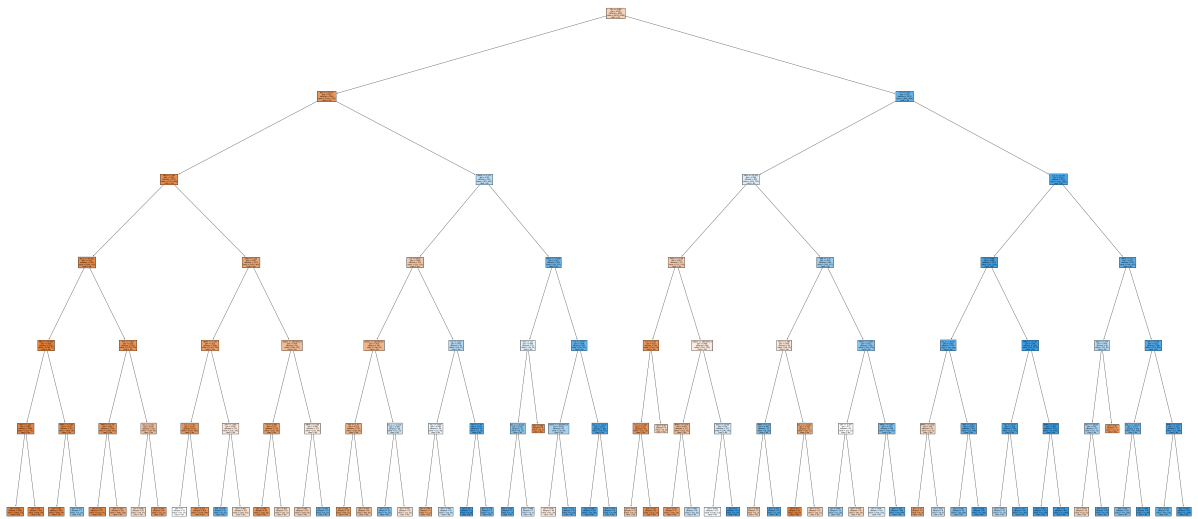
mples = 435\nvalue = [195, 478]\nclass = No'),
  Text(3155.586206896552, 465.9428571428573, 'OXY <= 3.155\ngini = 0.5\nsamples = 67\nvalue = [53, 52]\nclass = Yes'),
  Text(3117.1034482758623, 155.3142857142857, 'gini = 0.444\nsamples = 25\nvalue = [13, 26]\nclass = No'),
  Text(3194.068965517242, 155.3142857142857, 'gini = 0.478\nsamples = 42\nvalue = [40, 26]\nclass = Yes'),
  Text(3309.5172413793107, 465.9428571428573, 'NMHC <= 0.165\ngini = 0.375\nsamples = 368\nvalue = [142, 426]\nclass = No'),
  Text(3271.034482758621, 155.3142857142857, 'gini = 0.494\nsamples = 160\nvalue = [106, 132]\nclass = No'),
  Text(3348.0000000000005, 155.3142857142857, 'gini = 0.194\nsamples = 208\nvalue = [36, 294]\nclass = No'),
  Text(3954.1034482758623, 1397.8285714285716, 'O_3 <= 15.745\ngini = 0.129\nsamples = 2223\nvalue = [246, 3292]\nclass = No'),
  Text(3694.3448275862074, 1087.2, 'CO <= 0.885\ngini = 0.064\nsamples = 1545\nvalue = [82, 2391]\nclass = No'),
  Text(3540.4137931034484, 776.5714285714287, 'SO_2 <= 8.105\ngini = 0.252\nsamples = 176\nvalue = [44, 254]\nclass = No'),
  Text(3463.4482758620693, 465.9428571428573, 'NMHC <= 0.155\ngini = 0.48\nsamples = 14\nvalue = [12, 8]\nclass = Yes'),
  Text(3424.9655172413795, 155.3142857142857, 'gini = 0.32\nsamples = 6\nvalue = [8, 2]\nclass = Yes'),
  Text(3501.931034482759, 155.3142857142857, 'gini = 0.48\nsamples = 8\nvalue = [4, 6]\nclass = No'),
  Text(3617.379310344828, 465.9428571428573, 'NOx <= 175.0\ngini = 0.204\nsamples = 162\nvalue = [32, 246]\nclass = No'),
  Text(3578.896551724138, 155.3142857142857, 'gini = 0.349\nsamples = 76\nvalue = [29, 100]\nclass = No'),
  Text(3655.8620689655177, 155.3142857142857, 'gini = 0.039\nsamples = 86\nvalue = [3, 146]\nclass = No'),
  Text(3848.275862068966, 776.5714285714287, 'NMHC <= 0.235\ngini = 0.034\nsamples = 1369\nvalue = [38, 2137]\nclass = No'),
  Text(3771.3103448275865, 465.9428571428573, 'TOL <= 10.47\ngini = 0.145\nsamples = 246\nvalue = [32, 374]\nclass = No'),
  Text(3732.8275862068967, 155.3142857142857, 'gini = 0.455\nsamples = 15\nvalue = [7, 13]\nclass = No'),
  Text(3809.7931034482763, 155.3142857142857, 'gini = 0.121\nsamples = 231\nvalue = [25, 361]\nclass = No'),
  Text(3925.241379310345, 465.9428571428573, 'NMHC <= 0.275\ngini = 0.007\nsamples = 1123\nvalue = [6, 1763]\nclass = No'),
  Text(3886.7586206896553, 155.3142857142857, 'gini = 0.041\nsamples = 181\nvalue = [6, 278]\nclass = No'),
  Text(3963.724137931035, 155.3142857142857, 'gini = 0.0\nsamples = 942\nvalue = [0, 1485]\nclass = No'),
  Text(4213.862068965517, 1087.2, 'PM25 <= 12.1\ngini = 0.261\nsamples = 678\nvalue = [164, 901]\nclass = No'),
  Text(4117.6551724137935, 776.5714285714287, 'BEN <= 4.675\ngini = 0.476\nsamples = 80\nvalue = [52, 81]\nclass = No'),
  Text(4079.1724137931037, 465.9428571428573, 'NOx <= 168.05\ngini = 0.45\nsamples = 75\nvalue = [41, 79]\nclass = No'),
  Text(4040.689655172414, 155.3142857142857, 'gini = 0.474\nsamples = 39\nvalue = [35, 22]\nclass = Yes'),
  Text(4117.6551724137935, 155.3142857142857, 'gini = 0.172\nsamples = 36\nvalue = [6, 57]\nclass = No'),
  Text(4156.137931034483, 465.9428571428573, 'gini = 0.26\nsamples = 5\nvalue = [11, 2]\nclass = Yes'),

```

```

Text(4310.068965517242, 776.5714285714287, 'TOL <= 15.88\ngini = 0.211\nsamples = 598\nvalue = [112, 820]\nclass = No'),
Text(4233.103448275862, 465.9428571428573, 'NO_2 <= 103.45\ngini = 0.313\nsamples = 275\nvalue = [81, 336]\nclass = No'),
Text(4194.620689655173, 155.3142857142857, 'gini = 0.36\nsamples = 215\nvalue = [77, 250]\nclass = No'),
Text(4271.586206896552, 155.3142857142857, 'gini = 0.085\nsamples = 60\nvalue = [4, 86]\nclass = No'),
Text(4387.034482758621, 465.9428571428573, 'NMHC <= 0.225\ngini = 0.113\nsamples = 323\nvalue = [31, 484]\nclass = No'),
Text(4348.551724137931, 155.3142857142857, 'gini = 0.279\nsamples = 92\nvalue = [24, 119]\nclass = No'),
Text(4425.517241379311, 155.3142857142857, 'gini = 0.037\nsamples = 231\nvalue = [7, 365]\nclass = No')]

```



```

In [76]: print("Linear:",lis)
print("Lasso:",las)
print("Ridge:",rrs)
print("ElasticNet:",ens)
print("Logistic:",los)
print("Random Forest:",rfcs)

```

```

Linear: 0.5758911744718893
Lasso: 0.49886828067074285
Ridge: 0.5759339018894761
ElasticNet: 0.5250372976356534
Logistic: 0.6163230240549828
Random Forest: 0.8953377842749368

```

Best model is Random Forest

