2005

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,Rid
from sklearn.model_selection import train_test_split
```

In [2]:

```
df=pd.read_csv("madrid_2005.csv")
df
```

Out[2]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	PM10
0	2005-11-01 01:00:00	NaN	0.77	NaN	NaN	NaN	57.130001	128.699997	NaN	14.720000	14.91
1	2005-11-01 01:00:00	1.52	0.65	1.49	4.57	0.25	86.559998	181.699997	1.27	11.680000	30.93
2	2005-11-01 01:00:00	NaN	0.40	NaN	NaN	NaN	46.119999	53.000000	NaN	30.469999	14.60
3	2005-11-01 01:00:00	NaN	0.42	NaN	NaN	NaN	37.220001	52.009998	NaN	21.379999	15.16
4	2005-11-01 01:00:00	NaN	0.57	NaN	NaN	NaN	32.160000	36.680000	NaN	33.410000	5.00
...	...	...	...	...	...	...	...	...	...	...	...
236995	2006-01-01 00:00:00	1.08	0.36	1.01	NaN	0.11	21.990000	23.610001	NaN	43.349998	5.00
236996	2006-01-01 00:00:00	0.39	0.54	1.00	1.00	0.11	2.200000	4.220000	1.00	69.639999	4.95
236997	2006-01-01 00:00:00	0.19	NaN	0.26	NaN	0.08	26.730000	30.809999	NaN	43.840000	4.31
236998	2006-01-01 00:00:00	0.14	NaN	1.00	NaN	0.06	13.770000	17.770000	NaN	NaN	5.00
236999	2006-01-01 00:00:00	0.50	0.40	0.73	1.84	0.13	20.940001	26.950001	1.49	48.259998	5.67

237000 rows × 17 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 237000 entries, 0 to 236999
Data columns (total 17 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     237000 non-null  object 
 1   BEN      70370 non-null   float64
 2   CO       217656 non-null  float64
 3   EBE      68955 non-null   float64
 4   MXY      32549 non-null   float64
 5   NMHC     92854 non-null   float64
 6   NO_2     235022 non-null  float64
 7   NOx      235049 non-null  float64
 8   OXY      32555 non-null   float64
 9   O_3      223162 non-null  float64
 10  PM10     232142 non-null  float64
 11  PM25     69407 non-null   float64
 12  PXY      32549 non-null   float64
 13  SO_2     235277 non-null  float64
 14  TCH      93076 non-null   float64
 15  TOL      70255 non-null   float64
 16  station  237000 non-null  int64  
dtypes: float64(15), int64(1), object(1)
memory usage: 30.7+ MB
```

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

Out[4]:

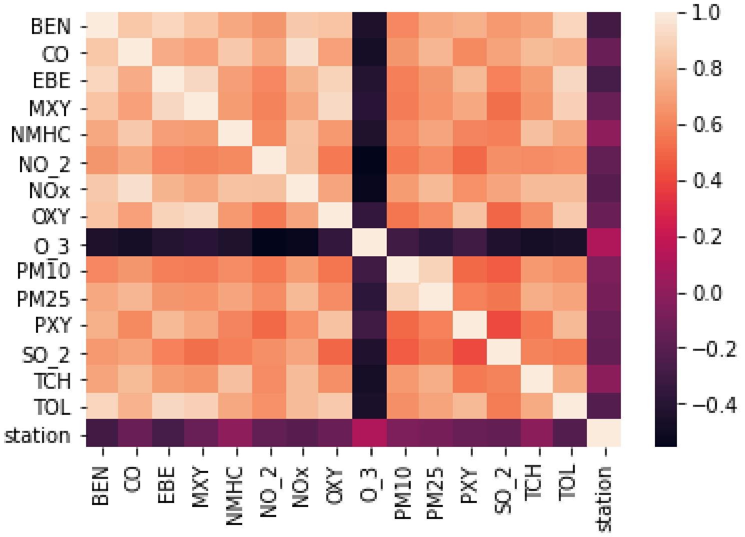
	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	PM10
5	2005-11-01 01:00:00	1.92	0.88	2.44	5.14	0.22	90.309998	207.699997	2.78	13.760000	18.07
22	2005-11-01 01:00:00	0.30	0.22	0.25	0.59	0.11	18.540001	19.020000	0.67	46.799999	9.88
25	2005-11-01 01:00:00	0.67	0.49	0.94	3.44	0.17	48.740002	74.349998	1.57	23.430000	13.88
31	2005-11-01 02:00:00	3.10	0.84	3.21	6.82	0.22	89.919998	224.199997	3.72	12.390000	28.74
48	2005-11-01 02:00:00	0.39	0.20	0.29	0.68	0.11	16.639999	17.080000	0.40	47.689999	8.78
...	...	...	...	...	...	...	...	...	...	...	...
236970	2005-12-31 23:00:00	0.37	0.39	1.00	1.00	0.10	4.500000	5.550000	1.00	57.779999	8.26
236973	2005-12-31 23:00:00	0.92	0.45	1.26	3.42	0.14	37.250000	49.060001	2.57	31.889999	19.73
236979	2006-01-01 00:00:00	1.00	0.38	1.11	2.35	0.04	35.919998	59.480000	1.39	35.810001	4.22
236996	2006-01-01 00:00:00	0.39	0.54	1.00	1.00	0.11	2.200000	4.220000	1.00	69.639999	4.95
236999	2006-01-01 00:00:00	0.50	0.40	0.73	1.84	0.13	20.940001	26.950001	1.49	48.259998	5.67

20070 rows × 17 columns

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

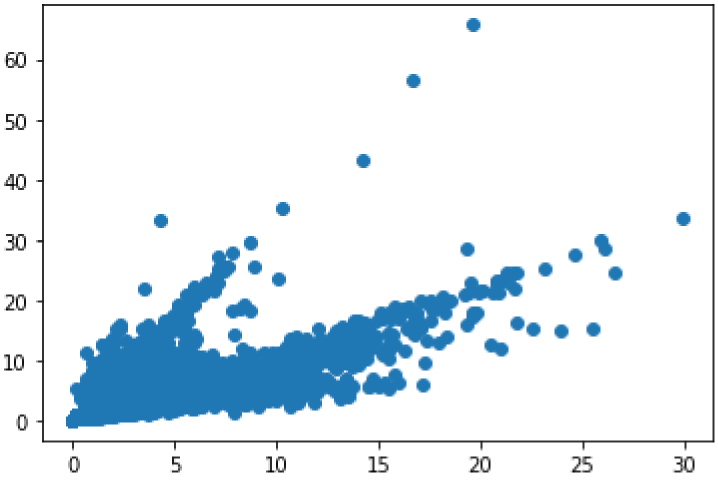
In [6]:
```
sns.heatmap(df1.corr())
```

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



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

Out[7]:
```
[<matplotlib.lines.Line2D at 0x27bd222a160>]
```



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)
```

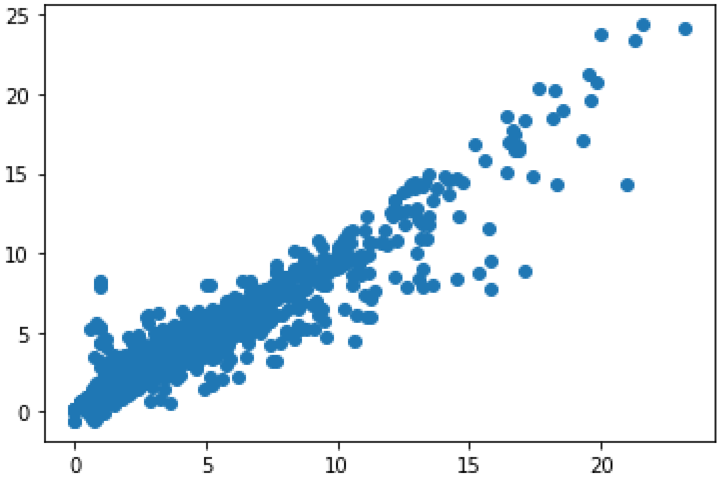
Linear

```
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 0x27bd22e7bb0>
```



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

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

Out[14]: 1.31    845
         1.33    820
         1.28    812
         1.30    806
         1.34    794
                ...
         3.04      1
         3.22      1
         2.79      1
         2.68      1
         3.37      1
         Name: TCH, Length: 198, 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    12093
         2.0     7977
         Name: TCH, dtype: int64
```

Lasso

In [16]:

```
la=Lasso(alpha=5)
la.fit(x_train,y_train)
```

Out[16]:

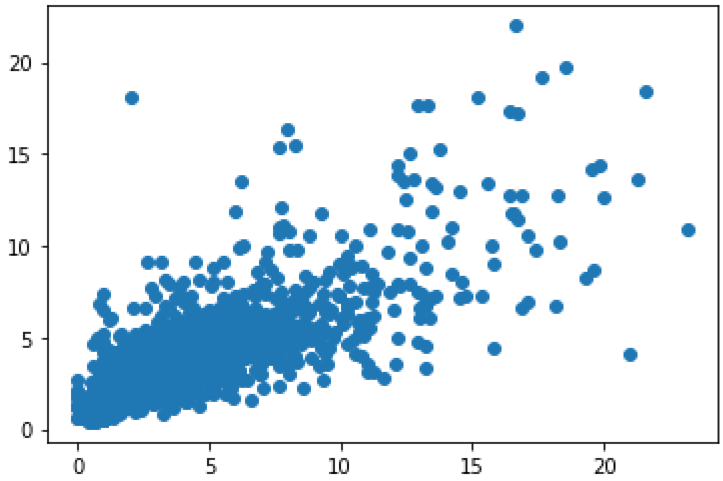
```
Lasso(alpha=5)
```

In [17]:

```
prediction1=la.predict(x_test)
plt.scatter(y_test,prediction1)
```

Out[17]:

```
<matplotlib.collections.PathCollection at 0x27bd2ed5880>
```



In [18]:

```
las=la.score(x_test,y_test)
```

Ridge

In [19]:

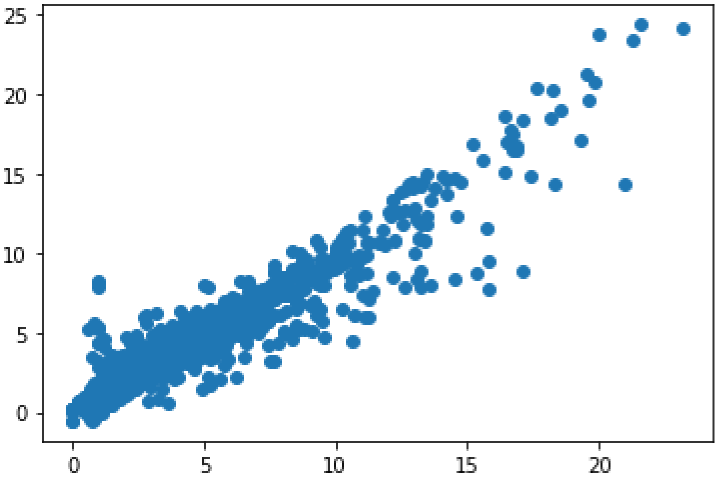
```
rr=Ridge(alpha=1)
rr.fit(x_train,y_train)
```

Out[19]:

```
Ridge(alpha=1)
```

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

Out[20]:
```
<matplotlib.collections.PathCollection at 0x27bd22002e0>
```



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

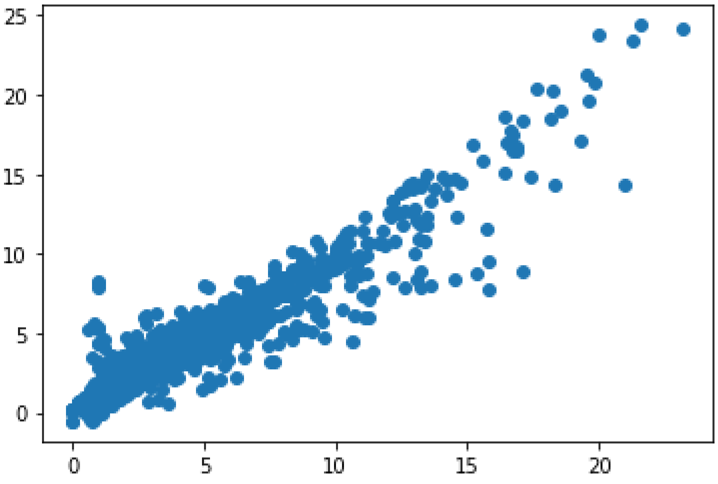
ElasticNet

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 0x27bd2f5acd0>
```



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

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

```
0.9254069105731428
```

Out[25]:
```
0.9242640908733394
```

Logistic

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

Out[26]:
```
Low     12093
High     7977
Name: TCH, dtype: int64
```

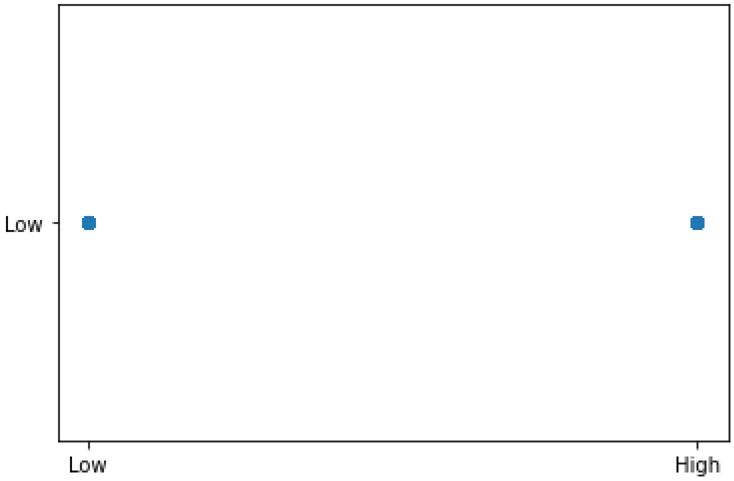
In [27]:
```
x=df1.drop(["TCH"],axis=1)
y=df1["TCH"]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

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 0x27bd2fbaa60>
```



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="accu
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_
```

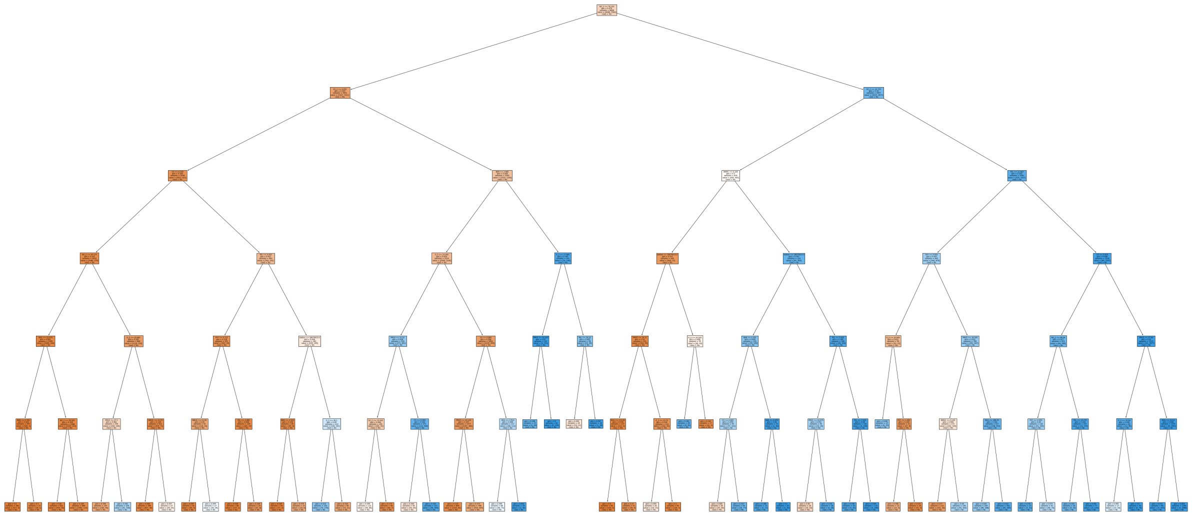
In [38]:
```
rfc_best=grid_search.best_estimator_
```

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',"
```

Out[39]:
```
[Text(2273.3333333333335, 2019.0857142857144, 'NO_2 <= 78.195\ngini = 0.479\n
samples = 8898\nvalue = [8466, 5583]\nclass = Yes'),
 Text(1271.0, 1708.457142857143, 'SO_2 <= 9.055\ngini = 0.325\nsamples = 5901
\nvalue = [7447, 1911]\nclass = Yes'),
 Text(661.3333333333334, 1397.8285714285716, 'CO <= 0.555\ngini = 0.199\nsamp
les = 3536\nvalue = [4922, 621]\nclass = Yes'),
 Text(330.6666666666667, 1087.2, 'NO_2 <= 45.175\ngini = 0.147\nsamples = 305
3\nvalue = [4380, 379]\nclass = Yes'),
 Text(165.33333333333334, 776.5714285714287, 'NOx <= 22.075\ngini = 0.081\nsa
mples = 2185\nvalue = [3260, 144]\nclass = Yes'),
 Text(82.66666666666667, 465.9428571428573, 'BEN <= 0.765\ngini = 0.002\nsamp
les = 597\nvalue = [935, 1]\nclass = Yes'),
 Text(41.333333333333336, 155.3142857142857, 'gini = 0.0\nsamples = 582\nvalu
e = [917, 0]\nclass = Yes'),
 Text(124.0, 155.3142857142857, 'gini = 0.1\nsamples = 15\nvalue = [18, 1]\nc
lass = Yes'),
 Text(248.0, 465.9428571428573, 'NO_2 <= 28.395\ngini = 0.109\nsamples = 1588
\nvalue = [2325, 143]\nclass = Yes'),
 Text(206.66666666666669, 155.3142857142857, 'gini = 0.06\nsamples = 622\nval
ue = [933, 30]\nclass = Yes'),
 Text(289.33333333333337, 155.3142857142857, 'gini = 0.139\nsamples = 966\nva
lue = [1392, 113]\nclass = Yes'),
 Text(496.0, 776.5714285714287, 'O_3 <= 25.665\ngini = 0.287\nsamples = 868\n
value = [1120, 235]\nclass = Yes'),
 Text(413.33333333333337, 465.9428571428573, 'PM10 <= 27.1\ngini = 0.481\nsam
ples = 256\nvalue = [244, 164]\nclass = Yes'),
 Text(372.0, 155.3142857142857, 'gini = 0.359\nsamples = 147\nvalue = [183, 5
6]\nclass = Yes'),
 Text(454.6666666666667, 155.3142857142857, 'gini = 0.461\nsamples = 109\nval
ue = [61, 108]\nclass = No'),
 Text(578.6666666666667, 465.9428571428573, 'NMHC <= 0.175\ngini = 0.139\nsam
ples = 612\nvalue = [876, 71]\nclass = Yes'),
 Text(537.3333333333334, 155.3142857142857, 'gini = 0.098\nsamples = 585\nval
ue = [847, 46]\nclass = Yes'),
 Text(620.0, 155.3142857142857, 'gini = 0.497\nsamples = 27\nvalue = [29, 25]
\nclass = Yes'),
 Text(992.0, 1087.2, 'TOL <= 4.435\ngini = 0.427\nsamples = 483\nvalue = [54
2, 242]\nclass = Yes'),
 Text(826.6666666666667, 776.5714285714287, 'PXY <= 0.715\ngini = 0.218\nsamp
les = 201\nvalue = [303, 43]\nclass = Yes'),
 Text(744.0, 465.9428571428573, 'NMHC <= 0.135\ngini = 0.34\nsamples = 92\nva
lue = [119, 33]\nclass = Yes'),
 Text(702.6666666666667, 155.3142857142857, 'gini = 0.097\nsamples = 65\nvalu
e = [93, 5]\nclass = Yes'),
 Text(785.3333333333334, 155.3142857142857, 'gini = 0.499\nsamples = 27\nvalu
e = [26, 28]\nclass = No'),
 Text(909.3333333333334, 465.9428571428573, 'TOL <= 3.485\ngini = 0.098\nsamp
les = 109\nvalue = [184, 10]\nclass = Yes'),
 Text(868.0, 155.3142857142857, 'gini = 0.028\nsamples = 80\nvalue = [137, 2]
\nclass = Yes'),
 Text(950.6666666666667, 155.3142857142857, 'gini = 0.249\nsamples = 29\nvalu
e = [47, 8]\nclass = Yes'),
 Text(1157.3333333333335, 776.5714285714287, 'station <= 28079015.0\ngini =
0.496\nsamples = 282\nvalue = [239, 199]\nclass = Yes'),
 Text(1074.6666666666667, 465.9428571428573, 'BEN <= 2.505\ngini = 0.143\nsam
ples = 69\nvalue = [95, 8]\nclass = Yes'),
 Text(1033.3333333333335, 155.3142857142857, 'gini = 0.049\nsamples = 54\nval
ue = [77, 2]\nclass = Yes'),
 Text(1116.0, 155.3142857142857, 'gini = 0.375\nsamples = 15\nvalue = [18, 6]
\nclass = Yes'),
 Text(1240.0, 465.9428571428573, 'O_3 <= 36.1\ngini = 0.49\nsamples = 213\nva
lue = [144, 191]\nclass = No'),
 Text(1198.6666666666667, 155.3142857142857, 'gini = 0.413\nsamples = 153\nva
lue = [70, 170]\nclass = No'),
 Text(1281.3333333333335, 155.3142857142857, 'gini = 0.344\nsamples = 60\nval
ue = [74, 21]\nclass = Yes'),
 Text(1880.6666666666667, 1397.8285714285716, 'BEN <= 3.685\ngini = 0.448\nsa
mples = 2365\nvalue = [2525, 1290]\nclass = Yes'),
 Text(1653.3333333333335, 1087.2, 'O_3 <= 13.265\ngini = 0.424\nsamples = 222
4\nvalue = [2506, 1100]\nclass = Yes'),
 Text(1488.0, 776.5714285714287, 'PM10 <= 21.52\ngini = 0.428\nsamples = 530
\nvalue = [271, 601]\nclass = No'),
 Text(1405.3333333333335, 465.9428571428573, 'SO_2 <= 13.545\ngini = 0.464\ns
amples = 159\nvalue = [161, 93]\nclass = Yes'),
 Text(1364.0, 155.3142857142857, 'gini = 0.499\nsamples = 111\nvalue = [95, 8
6]\nclass = Yes'),
 Text(1446.6666666666667, 155.3142857142857, 'gini = 0.173\nsamples = 48\nval
ue = [66, 7]\nclass = Yes'),
 Text(1570.6666666666667, 465.9428571428573, 'CO <= 0.565\ngini = 0.293\nsamp
les = 371\nvalue = [110, 508]\nclass = No'),
 Text(1529.3333333333335, 155.3142857142857, 'gini = 0.494\nsamples = 56\nval
ue = [54, 43]\nclass = Yes'),
 Text(1612.0, 155.3142857142857, 'gini = 0.192\nsamples = 315\nvalue = [56, 4
65]\nclass = No'),
 Text(1818.6666666666667, 776.5714285714287, 'CO <= 0.805\ngini = 0.298\nsamp
les = 1694\nvalue = [2235, 499]\nclass = Yes'),
 Text(1736.0, 465.9428571428573, 'PM25 <= 12.145\ngini = 0.257\nsamples = 157
7\nvalue = [2167, 386]\nclass = Yes'),
 Text(1694.6666666666667, 155.3142857142857, 'gini = 0.115\nsamples = 809\nva
lue = [1243, 81]\nclass = Yes'),
 Text(1777.3333333333335, 155.3142857142857, 'gini = 0.373\nsamples = 768\nva
lue = [924, 305]\nclass = Yes'),
 Text(1901.3333333333335, 465.9428571428573, 'NOx <= 178.8\ngini = 0.469\nsam
ples = 117\nvalue = [68, 113]\nclass = No'),
 Text(1860.0, 155.3142857142857, 'gini = 0.499\nsamples = 88\nvalue = [66, 7
3]\nclass = No'),
 Text(1942.6666666666667, 155.3142857142857, 'gini = 0.091\nsamples = 29\nval
ue = [2, 40]\nclass = No'),
 Text(2108.0, 1087.2, 'O_3 <= 17.865\ngini = 0.165\nsamples = 141\nvalue = [1
9, 190]\nclass = No'),
 Text(2025.3333333333335, 776.5714285714287, 'PM10 <= 42.425\ngini = 0.05\nsa
mples = 107\nvalue = [4, 151]\nclass = No'),
 Text(1984.0, 465.9428571428573, 'gini = 0.298\nsamples = 15\nvalue = [4, 18]
\nclass = No'),
 Text(2066.666666666667, 465.9428571428573, 'gini = 0.0\nsamples = 92\nvalue
= [0, 133]\nclass = No'),
 Text(2190.666666666667, 776.5714285714287, 'TOL <= 19.12\ngini = 0.401\nsamp
les = 34\nvalue = [15, 39]\nclass = No'),
 Text(2149.3333333333335, 465.9428571428573, 'gini = 0.493\nsamples = 16\nval
ue = [14, 11]\nclass = Yes'),
 Text(2232.0, 465.9428571428573, 'gini = 0.067\nsamples = 18\nvalue = [1, 28]
\nclass = No'),
 Text(3275.666666666667, 1708.457142857143, 'SO_2 <= 10.735\ngini = 0.34\nsam
ples = 2997\nvalue = [1019, 3672]\nclass = No'),
 Text(2738.3333333333335, 1397.8285714285716, 'NMHC <= 0.165\ngini = 0.5\nsam
ples = 616\nvalue = [490, 465]\nclass = Yes'),
 Text(2500.666666666667, 1087.2, 'station <= 28079015.0\ngini = 0.257\nsample
s = 306\nvalue = [403, 72]\nclass = Yes'),
 Text(2397.3333333333335, 776.5714285714287, 'PM25 <= 19.11\ngini = 0.14\nsam
ples = 251\nvalue = [354, 29]\nclass = Yes'),
 Text(2314.666666666667, 465.9428571428573, 'NOx <= 173.65\ngini = 0.078\nsam
ples = 137\nvalue = [212, 9]\nclass = Yes'),
 Text(2273.3333333333335, 155.3142857142857, 'gini = 0.0\nsamples = 96\nvalue
= [153, 0]\nclass = Yes'),
 Text(2356.0, 155.3142857142857, 'gini = 0.23\nsamples = 41\nvalue = [59, 9]
\nclass = Yes'),
 Text(2480.0, 465.9428571428573, 'O_3 <= 13.73\ngini = 0.216\nsamples = 114\n
value = [142, 20]\nclass = Yes'),
 Text(2438.666666666667, 155.3142857142857, 'gini = 0.495\nsamples = 20\nvalu
e = [16, 13]\nclass = Yes'),
 Text(2521.3333333333335, 155.3142857142857, 'gini = 0.1\nsamples = 94\nvalue
= [126, 7]\nclass = Yes'),
 Text(2604.0, 776.5714285714287, 'O_3 <= 19.525\ngini = 0.498\nsamples = 55\n
value = [49, 43]\nclass = Yes'),
 Text(2562.666666666667, 465.9428571428573, 'gini = 0.282\nsamples = 28\nvalu
e = [8, 39]\nclass = No'),
 Text(2645.3333333333335, 465.9428571428573, 'gini = 0.162\nsamples = 27\nval
ue = [41, 4]\nclass = Yes'),
 Text(2976.0, 1087.2, 'station <= 28079015.0\ngini = 0.297\nsamples = 310\nva
lue = [87, 393]\nclass = No'),
 Text(2810.666666666667, 776.5714285714287, 'BEN <= 3.335\ngini = 0.406\nsamp
les = 130\nvalue = [59, 149]\nclass = No'),
 Text(2728.0, 465.9428571428573, 'CO <= 0.805\ngini = 0.459\nsamples = 95\nva
lue = [57, 103]\nclass = No'),
 Text(2686.666666666667, 155.3142857142857, 'gini = 0.485\nsamples = 36\nvalu
e = [37, 26]\nclass = Yes'),
 Text(2769.3333333333335, 155.3142857142857, 'gini = 0.327\nsamples = 59\nval
ue = [20, 77]\nclass = No'),
 Text(2893.3333333333335, 465.9428571428573, 'EBE <= 4.055\ngini = 0.08\nsamp
les = 35\nvalue = [2, 46]\nclass = No'),
 Text(2852.0, 155.3142857142857, 'gini = 0.188\nsamples = 15\nvalue = [2, 17]
\nclass = No'),
 Text(2934.666666666667, 155.3142857142857, 'gini = 0.0\nsamples = 20\nvalue
= [0, 29]\nclass = No'),
 Text(3141.3333333333335, 776.5714285714287, 'TOL <= 5.625\ngini = 0.185\nsam
ples = 180\nvalue = [28, 244]\nclass = No'),
 Text(3058.666666666667, 465.9428571428573, 'PM25 <= 22.675\ngini = 0.452\nsa
mples = 36\nvalue = [19, 36]\nclass = No'),
 Text(3017.3333333333335, 155.3142857142857, 'gini = 0.48\nsamples = 18\nvalu
e = [15, 10]\nclass = Yes'),
 Text(3100.0, 155.3142857142857, 'gini = 0.231\nsamples = 18\nvalue = [4, 26]
\nclass = No'),
 Text(3224.0, 465.9428571428573, 'TOL <= 7.035\ngini = 0.08\nsamples = 144\nv
alue = [9, 208]\nclass = No'),
 Text(3182.666666666667, 155.3142857142857, 'gini = 0.198\nsamples = 29\nvalu
e = [5, 40]\nclass = No'),
 Text(3265.3333333333335, 155.3142857142857, 'gini = 0.045\nsamples = 115\nva
lue = [4, 168]\nclass = No'),
 Text(3813.0, 1397.8285714285716, 'TOL <= 11.825\ngini = 0.243\nsamples = 238
1\nvalue = [529, 3207]\nclass = No'),
 Text(3492.666666666667, 1087.2, 'OXY <= 0.825\ngini = 0.447\nsamples = 845\n
value = [444, 870]\nclass = No'),
 Text(3348.0, 776.5714285714287, 'O_3 <= 10.945\ngini = 0.422\nsamples = 56\n
value = [53, 23]\nclass = Yes'),
 Text(3306.666666666667, 465.9428571428573, 'gini = 0.408\nsamples = 15\nvalu
e = [6, 15]\nclass = No'),
 Text(3389.3333333333335, 465.9428571428573, 'PXY <= 1.385\ngini = 0.249\nsam
ples = 41\nvalue = [47, 8]\nclass = Yes'),
 Text(3348.0, 155.3142857142857, 'gini = 0.388\nsamples = 15\nvalue = [14, 5]
\nclass = Yes'),
 Text(3430.666666666667, 155.3142857142857, 'gini = 0.153\nsamples = 26\nvalu
e = [33, 3]\nclass = Yes'),
 Text(3637.3333333333335, 776.5714285714287, 'PM25 <= 18.505\ngini = 0.432\ns
amples = 789\nvalue = [391, 847]\nclass = No'),
 Text(3554.666666666667, 465.9428571428573, 'NMHC <= 0.165\ngini = 0.494\nsam
ples = 272\nvalue = [243, 194]\nclass = Yes'),
 Text(3513.3333333333335, 155.3142857142857, 'gini = 0.277\nsamples = 113\nva
lue = [151, 30]\nclass = Yes'),
 Text(3596.0, 155.3142857142857, 'gini = 0.46\nsamples = 159\nvalue = [92, 16
4]\nclass = No'),
 Text(3720.0, 465.9428571428573, 'NOx <= 188.65\ngini = 0.301\nsamples = 517
\nvalue = [148, 653]\nclass = No'),
 Text(3678.666666666667, 155.3142857142857, 'gini = 0.434\nsamples = 164\nval
ue = [88, 188]\nclass = No'),
 Text(3761.3333333333335, 155.3142857142857, 'gini = 0.202\nsamples = 353\nva
lue = [60, 465]\nclass = No'),
 Text(4133.333333333334, 1087.2, 'CO <= 0.915\ngini = 0.068\nsamples = 1536\n
value = [85, 2337]\nclass = No'),
 Text(3968.0, 776.5714285714287, 'NO_2 <= 98.26\ngini = 0.317\nsamples = 221
\nvalue = [64, 260]\nclass = No'),
 Text(3885.3333333333335, 465.9428571428573, 'PXY <= 2.955\ngini = 0.44\nsamp
les = 118\nvalue = [51, 105]\nclass = No'),
 Text(3844.0, 155.3142857142857, 'gini = 0.239\nsamples = 27\nvalue = [5, 31]
\nclass = No'),
 Text(3926.666666666667, 155.3142857142857, 'gini = 0.473\nsamples = 91\nvalu
e = [46, 74]\nclass = No'),
 Text(4050.666666666667, 465.9428571428573, 'TOL <= 13.275\ngini = 0.143\nsam
ples = 103\nvalue = [13, 155]\nclass = No'),
 Text(4009.3333333333335, 155.3142857142857, 'gini = 0.343\nsamples = 26\nval
ue = [9, 32]\nclass = No'),
 Text(4092.0000000000005, 155.3142857142857, 'gini = 0.061\nsamples = 77\nval
ue = [4, 123]\nclass = No'),
 Text(4298.666666666667, 776.5714285714287, 'PM10 <= 27.64\ngini = 0.02\nsamp
les = 1315\nvalue = [21, 2077]\nclass = No'),
 Text(4216.0, 465.9428571428573, 'O_3 <= 10.94\ngini = 0.228\nsamples = 60\nv
alue = [13, 86]\nclass = No'),
 Text(4174.666666666667, 155.3142857142857, 'gini = 0.495\nsamples = 17\nvalu
e = [13, 16]\nclass = No'),
 Text(4257.333333333334, 155.3142857142857, 'gini = 0.0\nsamples = 43\nvalue
= [0, 70]\nclass = No'),
 Text(4381.333333333334, 465.9428571428573, 'NOx <= 214.0\ngini = 0.008\nsamp
les = 1255\nvalue = [8, 1991]\nclass = No'),
 Text(4340.0, 155.3142857142857, 'gini = 0.072\nsamples = 69\nvalue = [4, 10
3]\nclass = No'),
 Text(4422.666666666667, 155.3142857142857, 'gini = 0.004\nsamples = 1186\nva
lue = [4, 1888]\nclass = No')]
```



In [40]:

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

```
Linear: 0.9254053895586418
Lasso: 0.7070521306882476
Ridge: 0.9254069105731428
ElasticNet: 0.9042630713701663
Logistic: 0.6027238000332171
Random Forest: 0.9084634785463566
```

Best Model is RidgeRegression

2006

In [41]:
```
df2=pd.read_csv("madrid_2006.csv")
df2
```

Out[41]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	P
0	2006-02-01 01:00:00	NaN	1.84	NaN	NaN	NaN	155.100006	490.100006	NaN	4.880000	97.57
1	2006-02-01 01:00:00	1.68	1.01	2.38	6.36	0.32	94.339996	229.699997	3.04	7.100000	25.82
2	2006-02-01 01:00:00	NaN	1.25	NaN	NaN	NaN	66.800003	192.000000	NaN	4.430000	34.41
3	2006-02-01 01:00:00	NaN	1.68	NaN	NaN	NaN	103.000000	407.799988	NaN	4.830000	28.26
4	2006-02-01 01:00:00	NaN	1.31	NaN	NaN	NaN	105.400002	269.200012	NaN	6.990000	54.18
...	...	...	...	...	...	...	...	...	...	...	
230563	2006-05-01 00:00:00	5.88	0.83	6.23	NaN	0.20	112.500000	218.000000	NaN	24.389999	93.12
230564	2006-05-01 00:00:00	0.76	0.32	0.48	1.09	0.08	51.900002	54.820000	0.61	48.410000	29.46
230565	2006-05-01 00:00:00	0.96	NaN	0.69	NaN	0.19	135.100006	179.199997	NaN	11.460000	64.68
230566	2006-05-01 00:00:00	0.50	NaN	0.67	NaN	0.10	82.599998	105.599998	NaN	NaN	94.36
230567	2006-05-01 00:00:00	1.95	0.74	1.99	4.00	0.24	107.300003	160.199997	2.01	17.730000	52.49

230568 rows × 17 columns

In [42]:

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 230568 entries, 0 to 230567
Data columns (total 17 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     230568 non-null  object 
 1   BEN      73979 non-null   float64
 2   CO       211665 non-null  float64
 3   EBE      73948 non-null   float64
 4   MXY      33422 non-null   float64
 5   NMHC     90829 non-null   float64
 6   NO_2     228855 non-null  float64
 7   NOx      228855 non-null  float64
 8   OXY      33472 non-null   float64
 9   O_3      216511 non-null  float64
 10  PM10     227469 non-null  float64
 11  PM25     61758 non-null   float64
 12  PXY      33447 non-null   float64
 13  SO_2     229125 non-null  float64
 14  TCH      90887 non-null   float64
 15  TOL      73840 non-null   float64
 16  station  230568 non-null  int64  
dtypes: float64(15), int64(1), object(1)
memory usage: 29.9+ MB
```

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

Out[43]:

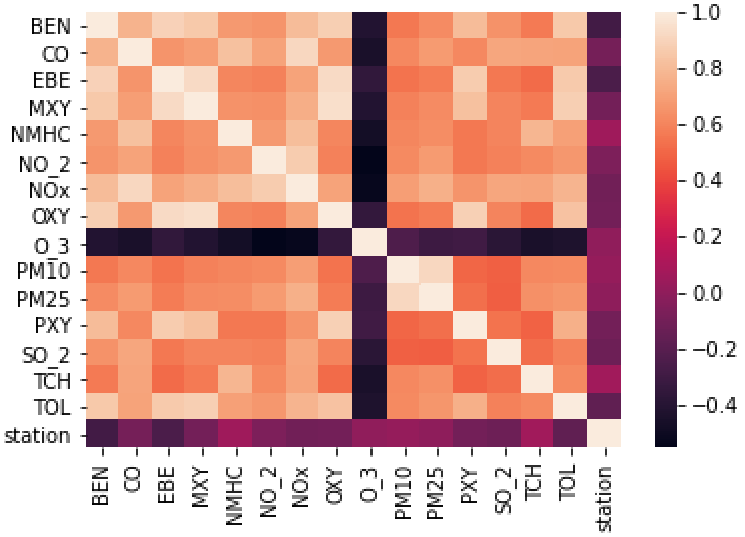
	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3
5	2006-02-01 01:00:00	9.41	1.69	9.98	19.959999	0.44	142.199997	453.500000	11.31	5.990000
22	2006-02-01 01:00:00	1.69	0.79	1.24	2.670000	0.17	59.910000	120.199997	1.11	2.450000
25	2006-02-01 01:00:00	2.35	1.47	2.64	9.660000	0.40	117.699997	346.399994	5.15	4.780000
31	2006-02-01 02:00:00	4.39	0.85	7.92	17.139999	0.25	92.059998	237.000000	9.24	5.920000
48	2006-02-01 02:00:00	1.93	0.79	1.24	2.740000	0.16	60.189999	125.099998	1.11	2.280000
...	...	...	...	...	...	...	...	...	...	...
230538	2006-04-30 23:00:00	0.42	0.40	0.37	0.430000	0.10	49.259998	51.689999	1.00	64.599998
230541	2006-04-30 23:00:00	1.63	0.94	1.53	2.200000	0.33	63.220001	211.399994	1.35	17.670000
230547	2006-05-01 00:00:00	3.99	1.06	3.71	7.960000	0.26	202.399994	343.500000	3.92	11.130000
230564	2006-05-01 00:00:00	0.76	0.32	0.48	1.090000	0.08	51.900002	54.820000	0.61	48.410000
230567	2006-05-01 00:00:00	1.95	0.74	1.99	4.000000	0.24	107.300003	160.199997	2.01	17.730000

24758 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

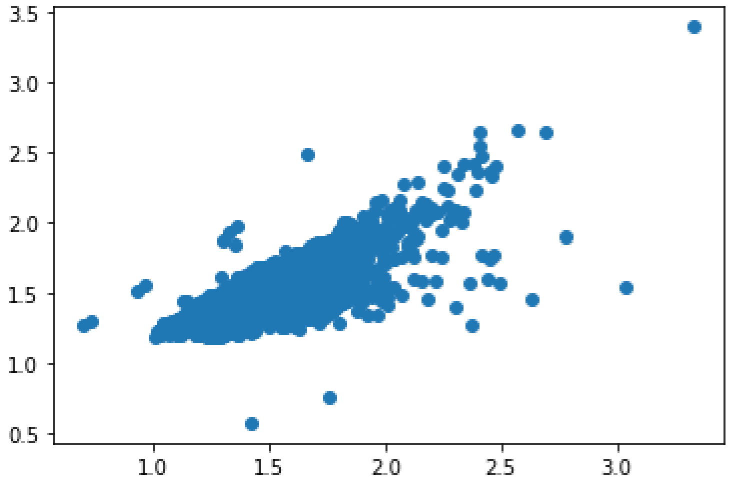
In [47]:
```
li=LinearRegression()
li.fit(x_train,y_train)
```

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

In [ ]:

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

Out[48]:
```
<matplotlib.collections.PathCollection at 0x27bd2fd2610>
```



In [49]:

```
lis=li.score(x_test,y_test)
```

In [50]:

```
df3["TCH"].value_counts()
```

Out[50]:

```
1.35    921
1.30    916
1.36    914
1.33    909
1.31    908
       ... 
0.94      1
0.81      1
0.72      1
3.33      1
2.91      1
Name: TCH, Length: 188, 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    14706
2.0    10052
Name: TCH, dtype: int64
```

Lasso

In [52]:

```
la=Lasso(alpha=5)
la.fit(x_train,y_train)
```

Out[52]:

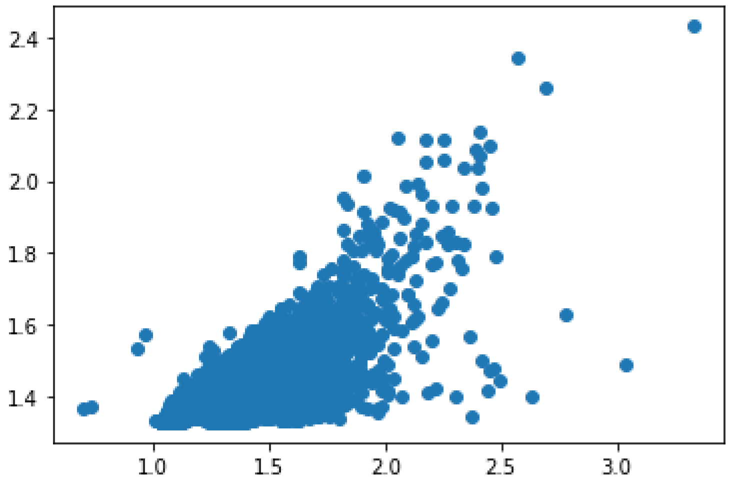
```
Lasso(alpha=5)
```

In [53]:

```
prediction1=la.predict(x_test)
plt.scatter(y_test,prediction1)
```

Out[53]:

```
<matplotlib.collections.PathCollection at 0x27bd3026760>
```



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

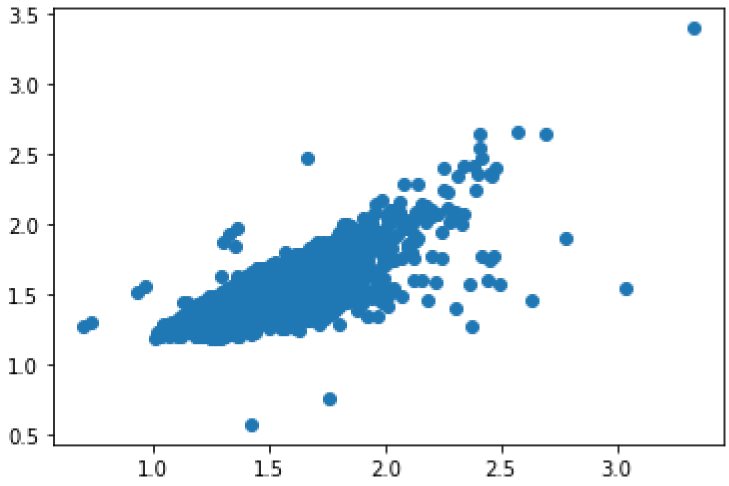
Ridge

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 0x27bd3075d30>
```



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

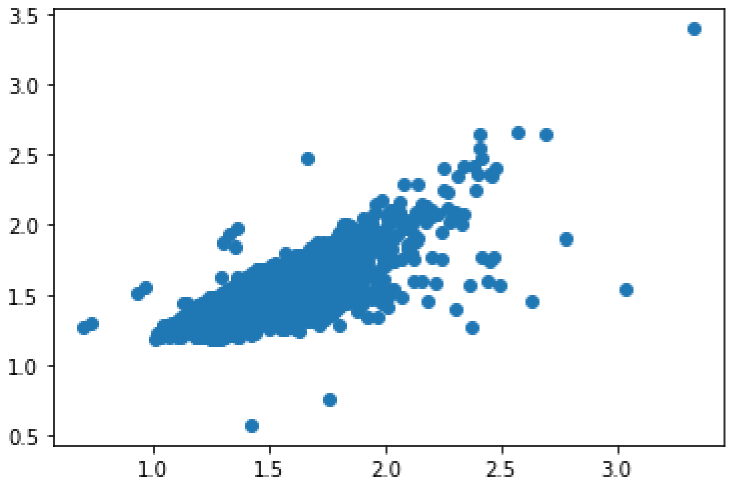
ElasticNet

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 0x27bd30d1880>
```



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

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

         0.6661494572681268

Out[61]: 0.6777209747604127
```

Logistic

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

```
Out[62]: Low     14706
         High    10052
         Name: TCH, dtype: int64
```

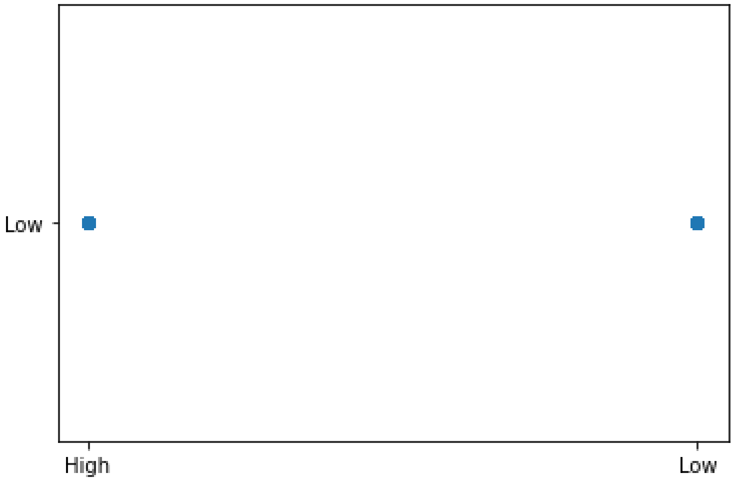
```
In [63]: x=df3.drop(["TCH"],axis=1)
         y=df3["TCH"]
         x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

```
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 0x27bd2b2d280>
```



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="accu
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_
```

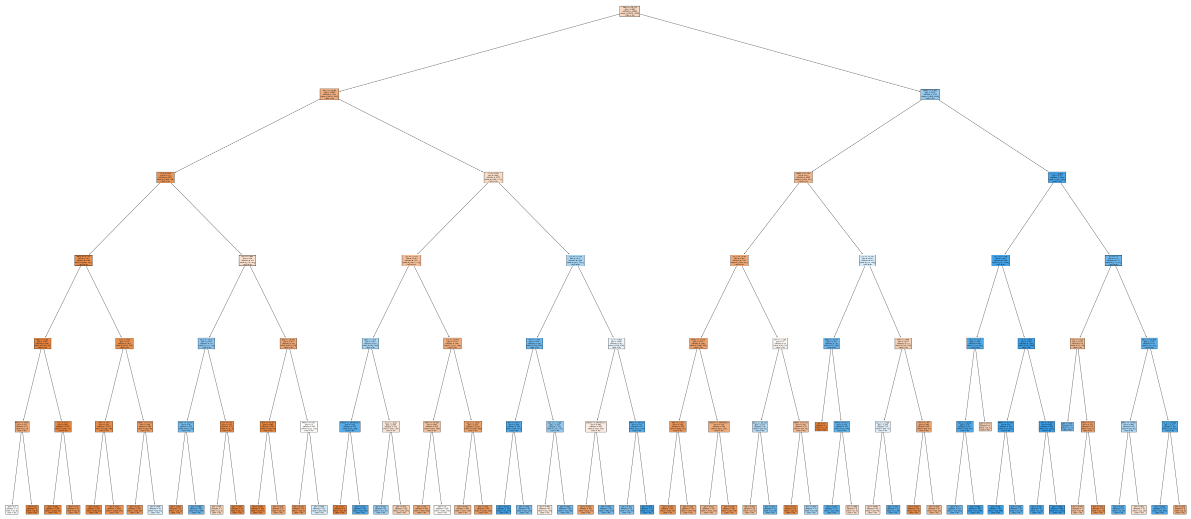
In [74]:
```
rfc_best=grid_search.best_estimator_
```

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',"
```

Out[75]:

```
[Text(2359.474137931035, 2019.0857142857144, 'PM10 <= 38.115\ngini = 0.483\ns
amples = 10996\nvalue = [10272, 7058]\nclass = Yes'),
 Text(1231.448275862069, 1708.457142857143, 'NO_2 <= 43.88\ngini = 0.388\nsam
ples = 7241\nvalue = [8414, 3010]\nclass = Yes'),
 Text(615.7241379310345, 1397.8285714285716, 'CO <= 0.535\ngini = 0.223\nsamp
les = 3871\nvalue = [5368, 789]\nclass = Yes'),
 Text(307.86206896551727, 1087.2, 'PM25 <= 8.135\ngini = 0.139\nsamples = 328
4\nvalue = [4844, 392]\nclass = Yes'),
 Text(153.93103448275863, 776.5714285714287, 'EBE <= 0.255\ngini = 0.065\nsam
ples = 1447\nvalue = [2193, 76]\nclass = Yes'),
 Text(76.96551724137932, 465.9428571428573, 'PM10 <= 9.865\ngini = 0.375\nsam
ples = 10\nvalue = [12, 4]\nclass = Yes'),
 Text(38.48275862068966, 155.3142857142857, 'gini = 0.5\nsamples = 5\nvalue =
[4, 4]\nclass = Yes'),
 Text(115.44827586206898, 155.3142857142857, 'gini = 0.0\nsamples = 5\nvalue
= [8, 0]\nclass = Yes'),
 Text(230.89655172413796, 465.9428571428573, 'CO <= 0.515\ngini = 0.062\nsamp
les = 1437\nvalue = [2181, 72]\nclass = Yes'),
 Text(192.41379310344828, 155.3142857142857, 'gini = 0.056\nsamples = 1394\nv
alue = [2122, 63]\nclass = Yes'),
 Text(269.3793103448276, 155.3142857142857, 'gini = 0.23\nsamples = 43\nvalue
= [59, 9]\nclass = Yes'),
 Text(461.79310344827593, 776.5714285714287, 'CO <= 0.445\ngini = 0.19\nsampl
es = 1837\nvalue = [2651, 316]\nclass = Yes'),
 Text(384.82758620689657, 465.9428571428573, 'NOx <= 19.9\ngini = 0.167\nsamp
les = 1553\nvalue = [2268, 230]\nclass = Yes'),
 Text(346.3448275862069, 155.3142857142857, 'gini = 0.057\nsamples = 494\nval
ue = [760, 23]\nclass = Yes'),
 Text(423.3103448275862, 155.3142857142857, 'gini = 0.212\nsamples = 1059\nva
lue = [1508, 207]\nclass = Yes'),
 Text(538.7586206896552, 465.9428571428573, 'NMHC <= 0.175\ngini = 0.299\nsam
ples = 284\nvalue = [383, 86]\nclass = Yes'),
 Text(500.2758620689656, 155.3142857142857, 'gini = 0.202\nsamples = 242\nval
ue = [350, 45]\nclass = Yes'),
 Text(577.2413793103449, 155.3142857142857, 'gini = 0.494\nsamples = 42\nvalu
e = [33, 41]\nclass = No'),
 Text(923.5862068965519, 1087.2, 'PXY <= 0.995\ngini = 0.49\nsamples = 587\nv
alue = [524, 397]\nclass = Yes'),
 Text(769.6551724137931, 776.5714285714287, 'SO_2 <= 10.52\ngini = 0.416\nsam
ples = 237\nvalue = [114, 273]\nclass = No'),
 Text(692.6896551724138, 465.9428571428573, 'NOx <= 14.355\ngini = 0.36\nsamp
les = 216\nvalue = [83, 270]\nclass = No'),
 Text(654.2068965517242, 155.3142857142857, 'gini = 0.165\nsamples = 15\nvalu
e = [20, 2]\nclass = Yes'),
 Text(731.1724137931035, 155.3142857142857, 'gini = 0.308\nsamples = 201\nval
ue = [63, 268]\nclass = No'),
 Text(846.6206896551724, 465.9428571428573, 'OXY <= 0.82\ngini = 0.161\nsampl
es = 21\nvalue = [31, 3]\nclass = Yes'),
 Text(808.1379310344828, 155.3142857142857, 'gini = 0.48\nsamples = 5\nvalue
= [3, 2]\nclass = Yes'),
 Text(885.1034482758621, 155.3142857142857, 'gini = 0.067\nsamples = 16\nvalu
e = [28, 1]\nclass = Yes'),
 Text(1077.5172413793105, 776.5714285714287, 'NOx <= 19.545\ngini = 0.357\nsa
mples = 350\nvalue = [410, 124]\nclass = Yes'),
 Text(1000.5517241379312, 465.9428571428573, 'NOx <= 14.96\ngini = 0.086\nsam
ples = 201\nvalue = [298, 14]\nclass = Yes'),
 Text(962.0689655172415, 155.3142857142857, 'gini = 0.054\nsamples = 180\nval
ue = [278, 8]\nclass = Yes'),
 Text(1039.0344827586207, 155.3142857142857, 'gini = 0.355\nsamples = 21\nval
ue = [20, 6]\nclass = Yes'),
 Text(1154.4827586206898, 465.9428571428573, 'PM25 <= 6.355\ngini = 0.5\nsamp
les = 149\nvalue = [112, 110]\nclass = Yes'),
 Text(1116.0, 155.3142857142857, 'gini = 0.257\nsamples = 24\nvalue = [28, 5]
\nclass = Yes'),
 Text(1192.9655172413793, 155.3142857142857, 'gini = 0.494\nsamples = 125\nva
lue = [84, 105]\nclass = No'),
 Text(1847.1724137931037, 1397.8285714285716, 'CO <= 0.655\ngini = 0.488\nsam
ples = 3370\nvalue = [3046, 2221]\nclass = Yes'),
 Text(1539.3103448275863, 1087.2, 'O_3 <= 12.815\ngini = 0.436\nsamples = 232
1\nvalue = [2483, 1174]\nclass = Yes'),
 Text(1385.3793103448277, 776.5714285714287, 'BEN <= 1.015\ngini = 0.462\nsam
ples = 429\nvalue = [242, 426]\nclass = No'),
 Text(1308.4137931034484, 465.9428571428573, 'station <= 28079015.0\ngini =
0.233\nsamples = 195\nvalue = [41, 264]\nclass = No'),
 Text(1269.9310344827588, 155.3142857142857, 'gini = 0.0\nsamples = 8\nvalue
= [13, 0]\nclass = Yes'),
 Text(1346.8965517241381, 155.3142857142857, 'gini = 0.173\nsamples = 187\nva
lue = [28, 264]\nclass = No'),
 Text(1462.344827586207, 465.9428571428573, 'SO_2 <= 8.84\ngini = 0.494\nsamp
les = 234\nvalue = [201, 162]\nclass = Yes'),
 Text(1423.8620689655174, 155.3142857142857, 'gini = 0.442\nsamples = 52\nval
ue = [27, 55]\nclass = No'),
 Text(1500.8275862068967, 155.3142857142857, 'gini = 0.472\nsamples = 182\nva
lue = [174, 107]\nclass = Yes'),
 Text(1693.2413793103449, 776.5714285714287, 'O_3 <= 23.875\ngini = 0.375\nsa
mples = 1892\nvalue = [2241, 748]\nclass = Yes'),
 Text(1616.2758620689656, 465.9428571428573, 'PM10 <= 23.885\ngini = 0.447\ns
amples = 633\nvalue = [675, 344]\nclass = Yes'),
 Text(1577.793103448276, 155.3142857142857, 'gini = 0.38\nsamples = 405\nvalu
e = [493, 169]\nclass = Yes'),
 Text(1654.7586206896553, 155.3142857142857, 'gini = 0.5\nsamples = 228\nvalu
e = [182, 175]\nclass = Yes'),
 Text(1770.2068965517242, 465.9428571428573, 'SO_2 <= 7.975\ngini = 0.326\nsa
mples = 1259\nvalue = [1566, 404]\nclass = Yes'),
 Text(1731.7241379310346, 155.3142857142857, 'gini = 0.425\nsamples = 356\nva
lue = [397, 175]\nclass = Yes'),
 Text(1808.689655172414, 155.3142857142857, 'gini = 0.274\nsamples = 903\nval
ue = [1169, 229]\nclass = Yes'),
 Text(2155.034482758621, 1087.2, 'O_3 <= 11.035\ngini = 0.455\nsamples = 1049
\nvalue = [563, 1047]\nclass = No'),
 Text(2001.1034482758623, 776.5714285714287, 'PXY <= 1.675\ngini = 0.335\nsam
ples = 508\nvalue = [165, 609]\nclass = No'),
 Text(1924.137931034483, 465.9428571428573, 'NOx <= 165.2\ngini = 0.204\nsamp
les = 254\nvalue = [44, 337]\nclass = No'),
 Text(1885.6551724137933, 155.3142857142857, 'gini = 0.073\nsamples = 160\nva
lue = [9, 227]\nclass = No'),
 Text(1962.6206896551726, 155.3142857142857, 'gini = 0.366\nsamples = 94\nval
ue = [35, 110]\nclass = No'),
 Text(2078.0689655172414, 465.9428571428573, 'NO_2 <= 71.89\ngini = 0.426\nsa
mples = 254\nvalue = [121, 272]\nclass = No'),
 Text(2039.5862068965519, 155.3142857142857, 'gini = 0.496\nsamples = 65\nval
ue = [51, 43]\nclass = Yes'),
 Text(2116.551724137931, 155.3142857142857, 'gini = 0.359\nsamples = 189\nval
ue = [70, 229]\nclass = No'),
 Text(2308.9655172413795, 776.5714285714287, 'CO <= 0.905\ngini = 0.499\nsamp
les = 541\nvalue = [398, 438]\nclass = No'),
 Text(2232.0, 465.9428571428573, 'station <= 28079015.0\ngini = 0.496\nsample
s = 452\nvalue = [379, 319]\nclass = Yes'),
 Text(2193.5172413793107, 155.3142857142857, 'gini = 0.318\nsamples = 255\nva
lue = [311, 77]\nclass = Yes'),
 Text(2270.4827586206898, 155.3142857142857, 'gini = 0.342\nsamples = 197\nva
lue = [68, 242]\nclass = No'),
 Text(2385.9310344827586, 465.9428571428573, 'PM10 <= 29.625\ngini = 0.237\ns
amples = 89\nvalue = [19, 119]\nclass = No'),
 Text(2347.4482758620693, 155.3142857142857, 'gini = 0.391\nsamples = 39\nval
ue = [16, 44]\nclass = No'),
 Text(2424.4137931034484, 155.3142857142857, 'gini = 0.074\nsamples = 50\nval
ue = [3, 75]\nclass = No'),
 Text(3487.5000000000005, 1708.457142857143, 'NMHC <= 0.185\ngini = 0.431\nsa
mples = 3755\nvalue = [1858, 4048]\nclass = No'),
 Text(3011.2758620689656, 1397.8285714285716, 'NMHC <= 0.165\ngini = 0.411\ns
amples = 1466\nvalue = [1603, 651]\nclass = Yes'),
 Text(2770.7586206896553, 1087.2, 'SO_2 <= 13.39\ngini = 0.346\nsamples = 116
1\nvalue = [1406, 402]\nclass = Yes'),
 Text(2616.8275862068967, 776.5714285714287, 'PM25 <= 23.715\ngini = 0.325\ns
amples = 1082\nvalue = [1346, 345]\nclass = Yes'),
 Text(2539.8620689655177, 465.9428571428573, 'EBE <= 1.305\ngini = 0.254\nsam
ples = 522\nvalue = [689, 121]\nclass = Yes'),
 Text(2501.379310344828, 155.3142857142857, 'gini = 0.219\nsamples = 397\nval
ue = [545, 78]\nclass = Yes'),
 Text(2578.344827586207, 155.3142857142857, 'gini = 0.354\nsamples = 125\nval
ue = [144, 43]\nclass = Yes'),
 Text(2693.7931034482763, 465.9428571428573, 'station <= 28079062.0\ngini =
0.379\nsamples = 560\nvalue = [657, 224]\nclass = Yes'),
 Text(2655.3103448275865, 155.3142857142857, 'gini = 0.423\nsamples = 379\nva
lue = [414, 181]\nclass = Yes'),
 Text(2732.2758620689656, 155.3142857142857, 'gini = 0.255\nsamples = 181\nva
lue = [243, 43]\nclass = Yes'),
 Text(2924.689655172414, 776.5714285714287, 'PXY <= 2.165\ngini = 0.5\nsample
s = 79\nvalue = [60, 57]\nclass = Yes'),
 Text(2847.724137931035, 465.9428571428573, 'NO_2 <= 60.475\ngini = 0.475\nsa
mples = 44\nvalue = [26, 41]\nclass = No'),
 Text(2809.241379310345, 155.3142857142857, 'gini = 0.435\nsamples = 20\nvalu
e = [17, 8]\nclass = Yes'),
 Text(2886.206896551724, 155.3142857142857, 'gini = 0.337\nsamples = 24\nvalu
e = [9, 33]\nclass = No'),
 Text(3001.6551724137935, 465.9428571428573, 'NMHC <= 0.135\ngini = 0.435\nsa
mples = 35\nvalue = [34, 16]\nclass = Yes'),
 Text(2963.1724137931037, 155.3142857142857, 'gini = 0.133\nsamples = 18\nval
ue = [26, 2]\nclass = Yes'),
 Text(3040.137931034483, 155.3142857142857, 'gini = 0.463\nsamples = 17\nvalu
e = [8, 14]\nclass = No'),
 Text(3251.7931034482763, 1087.2, 'O_3 <= 25.335\ngini = 0.493\nsamples = 305
\nvalue = [197, 249]\nclass = No'),
 Text(3117.1034482758623, 776.5714285714287, 'PM25 <= 16.37\ngini = 0.319\nsa
mples = 130\nvalue = [37, 149]\nclass = No'),
 Text(3078.6206896551726, 465.9428571428573, 'gini = 0.0\nsamples = 5\nvalue
= [7, 0]\nclass = Yes'),
 Text(3155.586206896552, 465.9428571428573, 'PM10 <= 87.13\ngini = 0.279\nsam
ples = 125\nvalue = [30, 149]\nclass = No'),
 Text(3117.1034482758623, 155.3142857142857, 'gini = 0.238\nsamples = 117\nva
lue = [23, 144]\nclass = No'),
 Text(3194.068965517242, 155.3142857142857, 'gini = 0.486\nsamples = 8\nvalue
= [7, 5]\nclass = Yes'),
 Text(3386.4827586206898, 776.5714285714287, 'MXY <= 3.905\ngini = 0.473\nsam
ples = 175\nvalue = [160, 100]\nclass = Yes'),
 Text(3309.5172413793107, 465.9428571428573, 'SO_2 <= 9.765\ngini = 0.498\nsa
mples = 91\nvalue = [64, 73]\nclass = No'),
 Text(3271.034482758621, 155.3142857142857, 'gini = 0.489\nsamples = 69\nvalu
e = [59, 44]\nclass = Yes'),
 Text(3348.0000000000005, 155.3142857142857, 'gini = 0.251\nsamples = 22\nval
ue = [5, 29]\nclass = No'),
 Text(3463.4482758620693, 465.9428571428573, 'PXY <= 1.99\ngini = 0.343\nsamp
les = 84\nvalue = [96, 27]\nclass = Yes'),
 Text(3424.9655172413795, 155.3142857142857, 'gini = 0.162\nsamples = 29\nval
ue = [41, 4]\nclass = Yes'),
 Text(3501.931034482759, 155.3142857142857, 'gini = 0.416\nsamples = 55\nvalu
e = [55, 23]\nclass = Yes'),
 Text(3963.724137931035, 1397.8285714285716, 'O_3 <= 23.815\ngini = 0.13\nsam
ples = 2289\nvalue = [255, 3397]\nclass = No'),
 Text(3752.068965517242, 1087.2, 'SO_2 <= 13.09\ngini = 0.088\nsamples = 1934
\nvalue = [142, 2926]\nclass = No'),
 Text(3655.8620689655177, 776.5714285714287, 'SO_2 <= 13.0\ngini = 0.189\nsam
ples = 685\nvalue = [114, 962]\nclass = No'),
 Text(3617.379310344828, 465.9428571428573, 'NO_2 <= 80.64\ngini = 0.18\nsamp
les = 677\nvalue = [106, 957]\nclass = No'),
 Text(3578.896551724138, 155.3142857142857, 'gini = 0.3\nsamples = 277\nvalue
= [79, 351]\nclass = No'),
 Text(3655.8620689655177, 155.3142857142857, 'gini = 0.082\nsamples = 400\nva
lue = [27, 606]\nclass = No'),
 Text(3694.3448275862074, 465.9428571428573, 'gini = 0.473\nsamples = 8\nvalu
e = [8, 5]\nclass = Yes'),
 Text(3848.275862068966, 776.5714285714287, 'CO <= 1.055\ngini = 0.028\nsampl
es = 1249\nvalue = [28, 1964]\nclass = No'),
 Text(3771.3103448275865, 465.9428571428573, 'PM25 <= 42.795\ngini = 0.07\nsa
mples = 399\nvalue = [23, 611]\nclass = No'),
 Text(3732.8275862068967, 155.3142857142857, 'gini = 0.05\nsamples = 367\nval
ue = [15, 568]\nclass = No'),
 Text(3809.7931034482763, 155.3142857142857, 'gini = 0.265\nsamples = 32\nval
ue = [8, 43]\nclass = No'),
 Text(3925.241379310345, 465.9428571428573, 'OXY <= 0.965\ngini = 0.007\nsamp
les = 850\nvalue = [5, 1353]\nclass = No'),
 Text(3886.7586206896553, 155.3142857142857, 'gini = 0.142\nsamples = 11\nval
ue = [1, 12]\nclass = No'),
 Text(3963.724137931035, 155.3142857142857, 'gini = 0.006\nsamples = 839\nval
ue = [4, 1341]\nclass = No'),
 Text(4175.379310344828, 1087.2, 'TOL <= 2.3\ngini = 0.312\nsamples = 355\nva
lue = [113, 471]\nclass = No'),
 Text(4040.689655172414, 776.5714285714287, 'OXY <= 0.525\ngini = 0.429\nsamp
les = 28\nvalue = [31, 14]\nclass = Yes'),
 Text(4002.2068965517246, 465.9428571428573, 'gini = 0.346\nsamples = 6\nvalu
e = [2, 7]\nclass = No'),
 Text(4079.1724137931037, 465.9428571428573, 'BEN <= 0.32\ngini = 0.313\nsamp
les = 22\nvalue = [29, 7]\nclass = Yes'),
 Text(4040.689655172414, 155.3142857142857, 'gini = 0.444\nsamples = 7\nvalue
= [8, 4]\nclass = Yes'),
 Text(4117.6551724137935, 155.3142857142857, 'gini = 0.219\nsamples = 15\nval
ue = [21, 3]\nclass = Yes'),
 Text(4310.068965517242, 776.5714285714287, 'NO_2 <= 60.845\ngini = 0.258\nsa
mples = 327\nvalue = [82, 457]\nclass = No'),
 Text(4233.103448275862, 465.9428571428573, 'PM25 <= 19.895\ngini = 0.464\nsa
mples = 46\nvalue = [26, 45]\nclass = No'),
 Text(4194.620689655173, 155.3142857142857, 'gini = 0.251\nsamples = 21\nvalu
e = [5, 29]\nclass = No'),
 Text(4271.586206896552, 155.3142857142857, 'gini = 0.491\nsamples = 25\nvalu
e = [21, 16]\nclass = Yes'),
 Text(4387.034482758621, 465.9428571428573, 'O_3 <= 76.585\ngini = 0.211\nsam
ples = 281\nvalue = [56, 412]\nclass = No'),
 Text(4348.551724137931, 155.3142857142857, 'gini = 0.194\nsamples = 275\nval
ue = [50, 410]\nclass = No'),
 Text(4425.517241379311, 155.3142857142857, 'gini = 0.375\nsamples = 6\nvalue
= [6, 2]\nclass = Yes')]
```



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

```
Linear: 0.665987858680082
Lasso: 0.46111308223729897
Ridge: 0.6661494572681268
ElasticNet: 0.5322012509285768
Logistic: 0.598546042003231
Random Forest: 0.863531448355453
```

Best model is Random Forest