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,Rid
from sklearn.model_selection import train_test_split
```

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

Out[2]:

	date	BEN	CO	EBE	MXY	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	105.0
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	100.5
2	2001-08-01 01:00:00	NaN	0.28	NaN	NaN	NaN	50.660000	61.380001	NaN	46.310001	100.0
3	2001-08-01 01:00:00	NaN	0.47	NaN	NaN	NaN	69.790001	73.449997	NaN	40.650002	69.7
4	2001-08-01 01:00:00	NaN	0.39	NaN	NaN	NaN	22.830000	24.799999	NaN	66.309998	75.1
...	...	...	...	...	...	...	...	...	...	...	
217867	2001-04-01 00:00:00	10.45	1.81	NaN	NaN	NaN	73.000000	264.399994	NaN	5.200000	47.8
217868	2001-04-01 00:00:00	5.20	0.69	4.56	NaN	0.13	71.080002	129.300003	NaN	13.460000	26.8
217869	2001-04-01 00:00:00	0.49	1.09	NaN	1.00	0.19	76.279999	128.399994	0.35	5.020000	40.7
217870	2001-04-01 00:00:00	5.62	1.01	5.04	11.38	NaN	80.019997	197.000000	2.58	5.840000	37.8
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	35.3

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]:

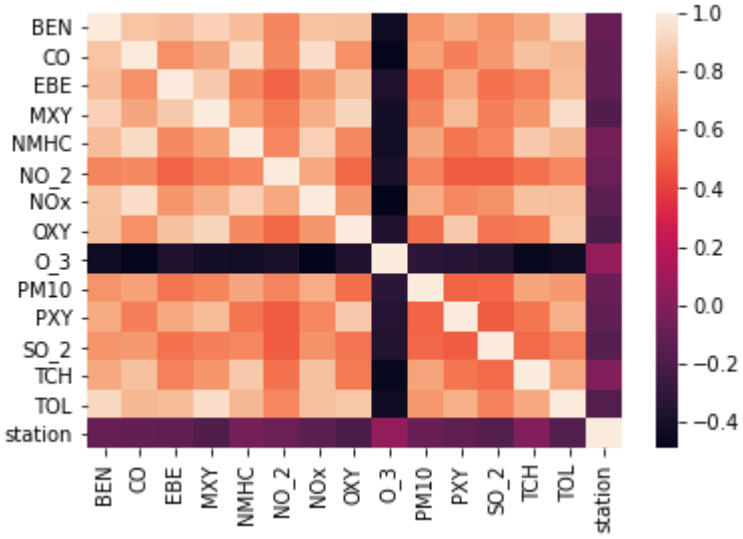
	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3
1	2001-08-01 01:00:00	1.50	0.34	1.49	4.100000	0.07	56.250000	75.169998	2.11	42.160000
5	2001-08-01 01:00:00	2.11	0.63	2.48	5.940000	0.05	66.260002	118.099998	3.15	33.500000
21	2001-08-01 01:00:00	0.80	0.43	0.71	1.200000	0.10	27.190001	29.700001	0.76	56.990002
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)
```

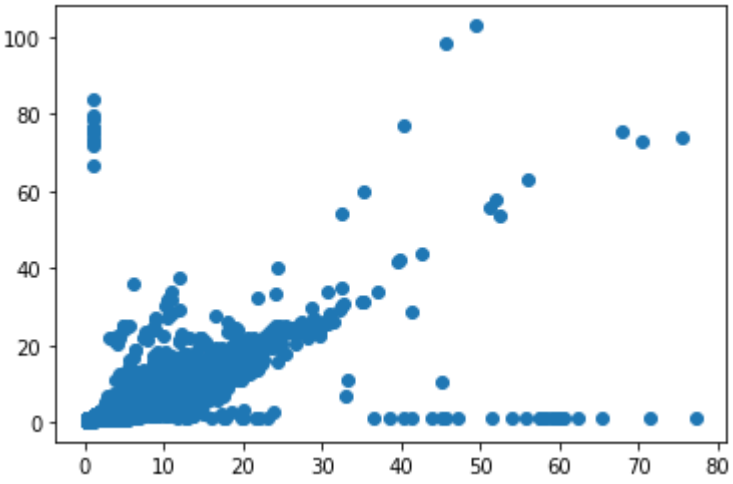
In [6]:
```
sns.heatmap(df1.corr())
```

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)
```

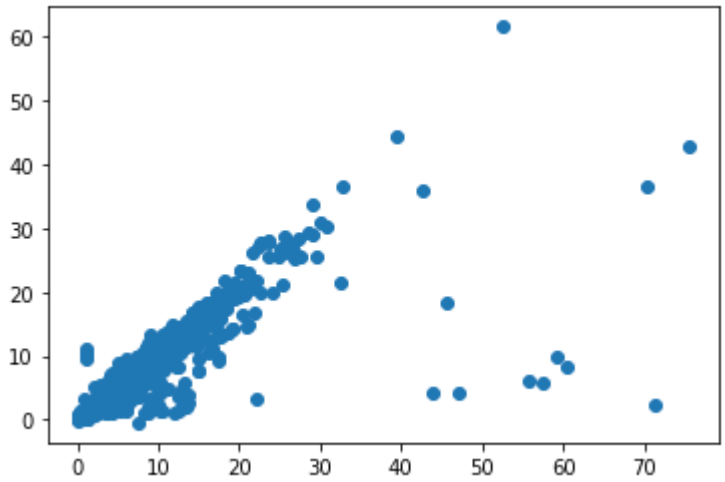
Linear

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
```

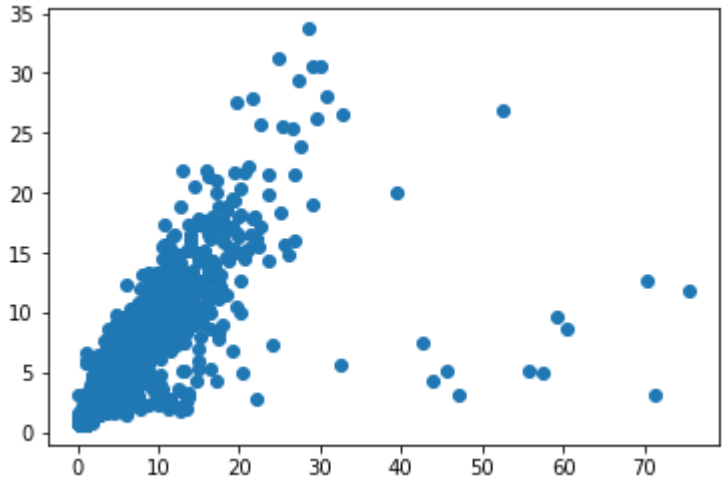
Lasso

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)
```

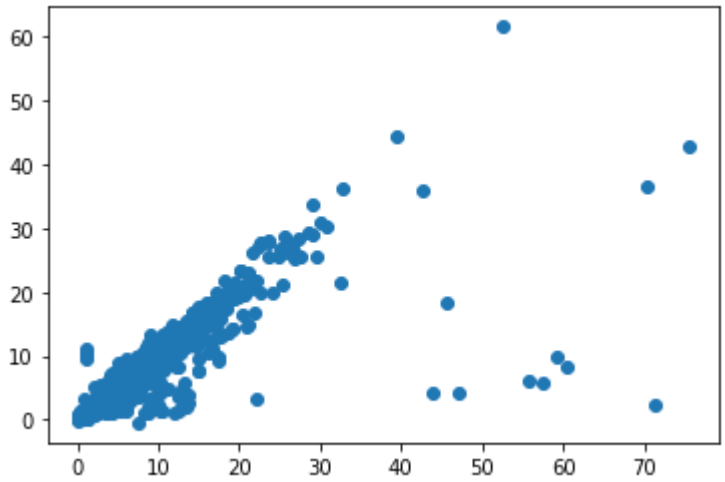
Ridge

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)
```

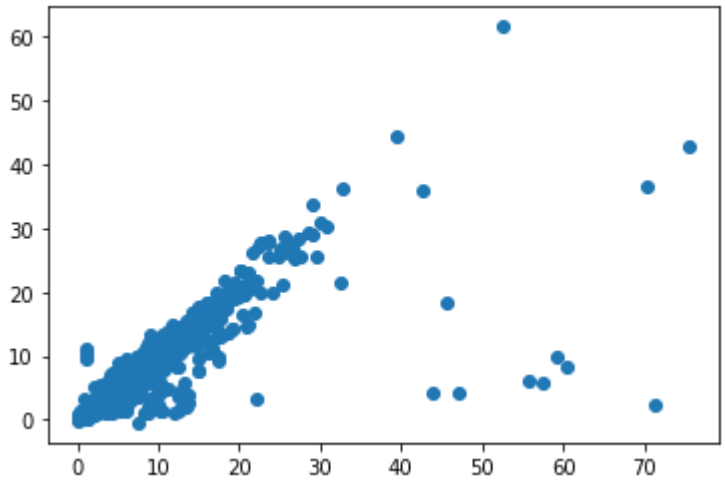
ElasticNet

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="accu
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',"
```

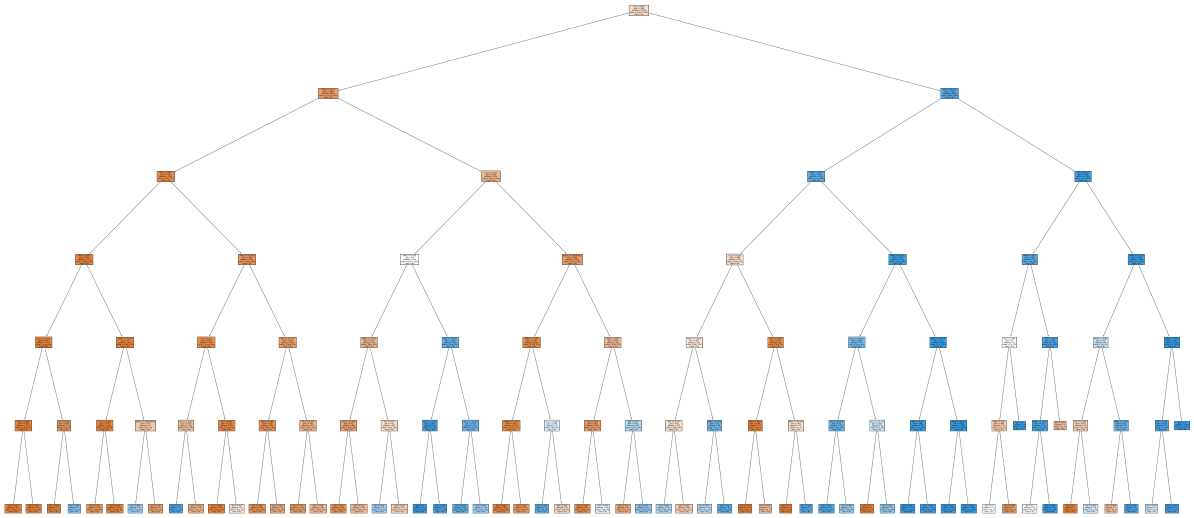
Out[38]:
```
[Text(2387.0, 2019.0857142857144, 'CO <= 1.025\ngini = 0.488\nsamples = 13145
\nvalue = [11977, 8791]\nclass = Yes'),
 Text(1220.923076923077, 1708.457142857143, 'PM10 <= 25.05\ngini = 0.292\nsam
ples = 8587\nvalue = [11173, 2407]\nclass = Yes'),
 Text(610.4615384615385, 1397.8285714285716, 'MXY <= 3.655\ngini = 0.169\nsam
ples = 5035\nvalue = [7134, 732]\nclass = Yes'),
 Text(305.2307692307692, 1087.2, 'MXY <= 1.875\ngini = 0.074\nsamples = 2723
\nvalue = [4067, 163]\nclass = Yes'),
 Text(152.6153846153846, 776.5714285714287, 'SO_2 <= 19.11\ngini = 0.044\nsam
ples = 1310\nvalue = [1978, 46]\nclass = Yes'),
 Text(76.3076923076923, 465.9428571428573, 'EBE <= 0.605\ngini = 0.039\nsampl
es = 1283\nvalue = [1946, 40]\nclass = Yes'),
 Text(38.15384615384615, 155.3142857142857, 'gini = 0.021\nsamples = 651\nval
ue = [1014, 11]\nclass = Yes'),
 Text(114.46153846153845, 155.3142857142857, 'gini = 0.059\nsamples = 632\nva
lue = [932, 29]\nclass = Yes'),
 Text(228.9230769230769, 465.9428571428573, 'CO <= 0.81\ngini = 0.266\nsample
s = 27\nvalue = [32, 6]\nclass = Yes'),
 Text(190.76923076923077, 155.3142857142857, 'gini = 0.062\nsamples = 22\nval
ue = [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\nsampl
es = 1413\nvalue = [2089, 117]\nclass = Yes'),
 Text(381.53846153846155, 465.9428571428573, 'O_3 <= 18.605\ngini = 0.076\nsa
mples = 1346\nvalue = [2028, 83]\nclass = Yes'),
 Text(343.38461538461536, 155.3142857142857, 'gini = 0.258\nsamples = 200\nva
lue = [268, 48]\nclass = Yes'),
 Text(419.6923076923077, 155.3142857142857, 'gini = 0.038\nsamples = 1146\nva
lue = [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\nvalu
e = [51, 14]\nclass = Yes'),
 Text(915.6923076923076, 1087.2, 'PM10 <= 14.945\ngini = 0.264\nsamples = 231
2\nvalue = [3067, 569]\nclass = Yes'),
 Text(763.0769230769231, 776.5714285714287, 'O_3 <= 10.83\ngini = 0.173\nsamp
les = 1047\nvalue = [1479, 156]\nclass = Yes'),
 Text(686.7692307692307, 465.9428571428573, 'NO_2 <= 40.42\ngini = 0.456\nsam
ples = 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\nval
ue = [125, 51]\nclass = Yes'),
 Text(839.3846153846154, 465.9428571428573, 'CO <= 0.965\ngini = 0.114\nsampl
es = 928\nvalue = [1352, 87]\nclass = Yes'),
 Text(801.2307692307692, 155.3142857142857, 'gini = 0.094\nsamples = 904\nval
ue = [1330, 69]\nclass = Yes'),
 Text(877.5384615384615, 155.3142857142857, 'gini = 0.495\nsamples = 24\nvalu
e = [22, 18]\nclass = Yes'),
 Text(1068.3076923076924, 776.5714285714287, 'NOx <= 107.05\ngini = 0.328\nsa
mples = 1265\nvalue = [1588, 413]\nclass = Yes'),
 Text(992.0, 465.9428571428573, 'BEN <= 2.995\ngini = 0.235\nsamples = 644\nv
alue = [891, 140]\nclass = Yes'),
 Text(953.8461538461538, 155.3142857142857, 'gini = 0.206\nsamples = 563\nval
```

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

```
 Text(2289.230769230769, 776.5714285714287, 'BEN <= 2.805\ngini = 0.436\nsamp
les = 825\nvalue = [898, 425]\nclass = Yes'),
 Text(2212.9230769230767, 465.9428571428573, 'PXY <= 2.515\ngini = 0.305\nsam
ples = 569\nvalue = [741, 171]\nclass = Yes'),
 Text(2174.769230769231, 155.3142857142857, 'gini = 0.184\nsamples = 461\nval
ue = [656, 75]\nclass = Yes'),
 Text(2251.076923076923, 155.3142857142857, 'gini = 0.498\nsamples = 108\nval
ue = [85, 96]\nclass = No'),
 Text(2365.5384615384614, 465.9428571428573, 'EBE <= 2.93\ngini = 0.472\nsamp
les = 256\nvalue = [157, 254]\nclass = No'),
 Text(2327.3846153846152, 155.3142857142857, 'gini = 0.375\nsamples = 46\nval
ue = [51, 17]\nclass = Yes'),
 Text(2403.6923076923076, 155.3142857142857, 'gini = 0.427\nsamples = 210\nva
lue = [106, 237]\nclass = No'),
 Text(3553.076923076923, 1708.457142857143, 'MXY <= 14.675\ngini = 0.199\nsam
ples = 4558\nvalue = [804, 6384]\nclass = No'),
 Text(3052.3076923076924, 1397.8285714285716, 'NMHC <= 0.205\ngini = 0.263\ns
amples = 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\ns
amples = 583\nvalue = [481, 416]\nclass = Yes'),
 Text(2518.153846153846, 465.9428571428573, 'PXY <= 2.175\ngini = 0.49\nsampl
es = 525\nvalue = [460, 344]\nclass = Yes'),
 Text(2480.0, 155.3142857142857, 'gini = 0.469\nsamples = 121\nvalue = [68, 1
13]\nclass = No'),
 Text(2556.3076923076924, 155.3142857142857, 'gini = 0.467\nsamples = 404\nva
lue = [392, 231]\nclass = Yes'),
 Text(2670.769230769231, 465.9428571428573, 'NMHC <= 0.165\ngini = 0.35\nsamp
les = 58\nvalue = [21, 72]\nclass = No'),
 Text(2632.6153846153848, 155.3142857142857, 'gini = 0.477\nsamples = 18\nval
ue = [11, 17]\nclass = No'),
 Text(2708.9230769230767, 155.3142857142857, 'gini = 0.26\nsamples = 40\nvalu
e = [10, 55]\nclass = No'),
 Text(2899.6923076923076, 776.5714285714287, 'NOx <= 202.6\ngini = 0.173\nsam
ples = 68\nvalue = [104, 11]\nclass = Yes'),
 Text(2823.3846153846152, 465.9428571428573, 'PM10 <= 46.23\ngini = 0.078\nsa
mples = 57\nvalue = [95, 4]\nclass = Yes'),
 Text(2785.230769230769, 155.3142857142857, 'gini = 0.022\nsamples = 52\nvalu
e = [89, 1]\nclass = Yes'),
 Text(2861.5384615384614, 155.3142857142857, 'gini = 0.444\nsamples = 5\nvalu
e = [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 = 227
5\nvalue = [134, 3462]\nclass = No'),
 Text(3204.9230769230767, 776.5714285714287, 'NO_2 <= 102.45\ngini = 0.365\ns
amples = 275\nvalue = [102, 322]\nclass = No'),
 Text(3128.6153846153848, 465.9428571428573, 'TOL <= 21.035\ngini = 0.31\nsam
ples = 220\nvalue = [65, 274]\nclass = No'),
 Text(3090.4615384615386, 155.3142857142857, 'gini = 0.201\nsamples = 147\nva
lue = [25, 195]\nclass = No'),
 Text(3166.769230769231, 155.3142857142857, 'gini = 0.446\nsamples = 73\nvalu
```

```
e = [40, 79]\nclass = No'),
 Text(3281.230769230769, 465.9428571428573, 'SO_2 <= 11.575\ngini = 0.492\nsa
mples = 55\nvalue = [37, 48]\nclass = No'),
 Text(3243.076923076923, 155.3142857142857, 'gini = 0.0\nsamples = 8\nvalue =
[15, 0]\nclass = Yes'),
 Text(3319.3846153846152, 155.3142857142857, 'gini = 0.431\nsamples = 47\nval
ue = [22, 48]\nclass = No'),
 Text(3510.153846153846, 776.5714285714287, 'PM10 <= 35.275\ngini = 0.02\nsam
ples = 2000\nvalue = [32, 3140]\nclass = No'),
 Text(3433.846153846154, 465.9428571428573, 'PXY <= 3.215\ngini = 0.06\nsampl
es = 437\nvalue = [22, 689]\nclass = No'),
 Text(3395.6923076923076, 155.3142857142857, 'gini = 0.007\nsamples = 176\nva
lue = [1, 283]\nclass = No'),
 Text(3472.0, 155.3142857142857, 'gini = 0.094\nsamples = 261\nvalue = [21, 4
06]\nclass = No'),
 Text(3586.4615384615386, 465.9428571428573, 'PM10 <= 46.215\ngini = 0.008\ns
amples = 1563\nvalue = [10, 2451]\nclass = No'),
 Text(3548.3076923076924, 155.3142857142857, 'gini = 0.022\nsamples = 482\nva
lue = [8, 725]\nclass = No'),
 Text(3624.6153846153848, 155.3142857142857, 'gini = 0.002\nsamples = 1081\nv
alue = [2, 1726]\nclass = No'),
 Text(4053.846153846154, 1397.8285714285716, 'TOL <= 32.51\ngini = 0.064\nsam
ples = 1632\nvalue = [85, 2495]\nclass = No'),
 Text(3853.5384615384614, 1087.2, 'PM10 <= 26.4\ngini = 0.245\nsamples = 224
\nvalue = [48, 288]\nclass = No'),
 Text(3777.230769230769, 776.5714285714287, 'OXY <= 7.595\ngini = 0.5\nsample
s = 25\nvalue = [19, 19]\nclass = Yes'),
 Text(3739.076923076923, 465.9428571428573, 'BEN <= 5.37\ngini = 0.464\nsampl
es = 20\nvalue = [19, 11]\nclass = Yes'),
 Text(3700.9230769230767, 155.3142857142857, 'gini = 0.5\nsamples = 13\nvalue
= [9, 9]\nclass = Yes'),
 Text(3777.230769230769, 155.3142857142857, 'gini = 0.278\nsamples = 7\nvalue
= [10, 2]\nclass = Yes'),
 Text(3815.3846153846152, 465.9428571428573, 'gini = 0.0\nsamples = 5\nvalue
= [0, 8]\nclass = No'),
 Text(3929.846153846154, 776.5714285714287, 'PXY <= 25.0\ngini = 0.176\nsampl
es = 199\nvalue = [29, 269]\nclass = No'),
 Text(3891.6923076923076, 465.9428571428573, 'NMHC <= 0.215\ngini = 0.157\nsa
mples = 194\nvalue = [25, 267]\nclass = No'),
 Text(3853.5384615384614, 155.3142857142857, 'gini = 0.5\nsamples = 35\nvalue
= [24, 25]\nclass = No'),
 Text(3929.846153846154, 155.3142857142857, 'gini = 0.008\nsamples = 159\nval
ue = [1, 242]\nclass = No'),
 Text(3968.0, 465.9428571428573, 'gini = 0.444\nsamples = 5\nvalue = [4, 2]\n
class = Yes'),
 Text(4254.153846153846, 1087.2, 'NMHC <= 0.215\ngini = 0.032\nsamples = 1408
\nvalue = [37, 2207]\nclass = No'),
 Text(4120.615384615385, 776.5714285714287, 'PM10 <= 40.89\ngini = 0.49\nsamp
les = 46\nvalue = [30, 40]\nclass = No'),
 Text(4044.3076923076924, 465.9428571428573, 'CO <= 1.155\ngini = 0.473\nsamp
les = 25\nvalue = [24, 15]\nclass = Yes'),
 Text(4006.153846153846, 155.3142857142857, 'gini = 0.0\nsamples = 7\nvalue =
[12, 0]\nclass = Yes'),
 Text(4082.4615384615386, 155.3142857142857, 'gini = 0.494\nsamples = 18\nval
ue = [12, 15]\nclass = No'),
 Text(4196.923076923077, 465.9428571428573, 'NMHC <= 0.175\ngini = 0.312\nsam
ples = 21\nvalue = [6, 25]\nclass = No'),
```

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

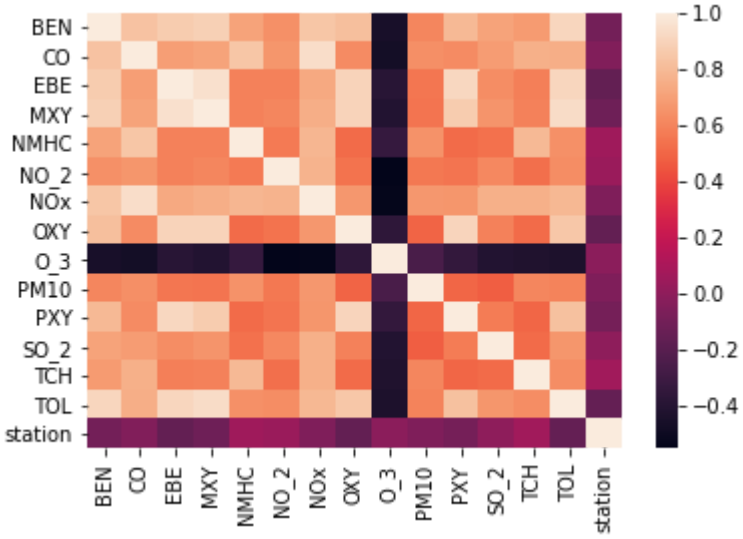
	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	PM10
1	2002-04-01 01:00:00	1.93	0.71	2.33	6.20	0.15	98.150002	153.399994	2.67	6.85	20.980000
5	2002-04-01 01:00:00	3.19	0.72	3.23	7.65	0.11	113.699997	187.000000	3.53	12.37	27.450001
22	2002-04-01 01:00:00	2.02	0.80	1.57	3.66	0.15	93.860001	101.300003	1.77	6.99	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

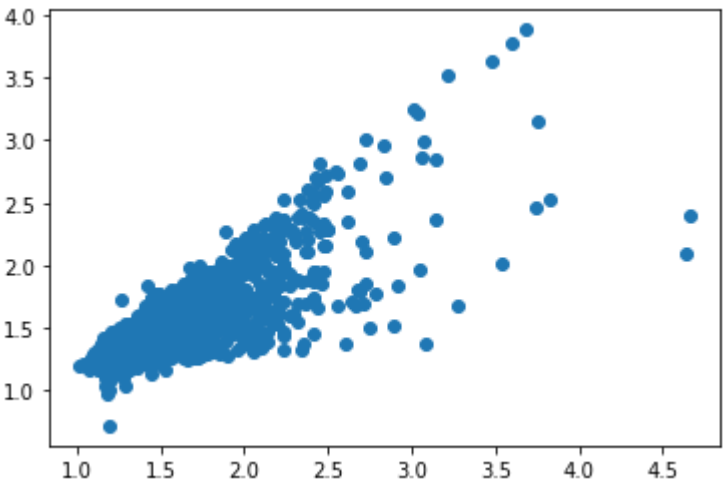
In [55]:
```
li=LinearRegression()
li.fit(x_train,y_train)
```

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 [ ]:
```

```

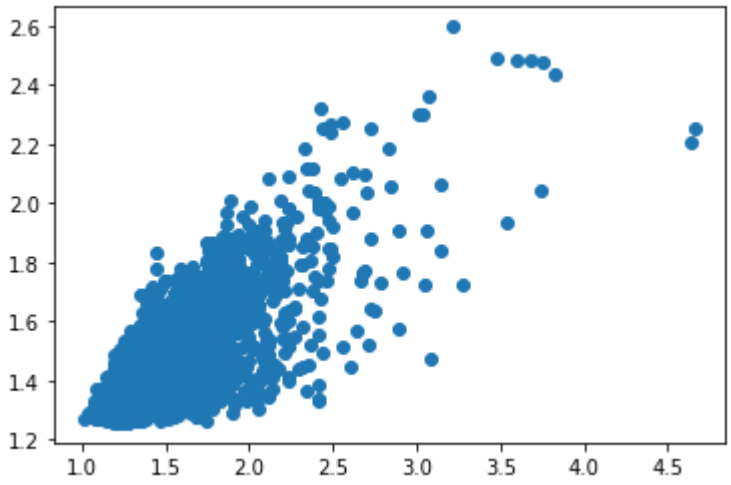
Lasso

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)
```

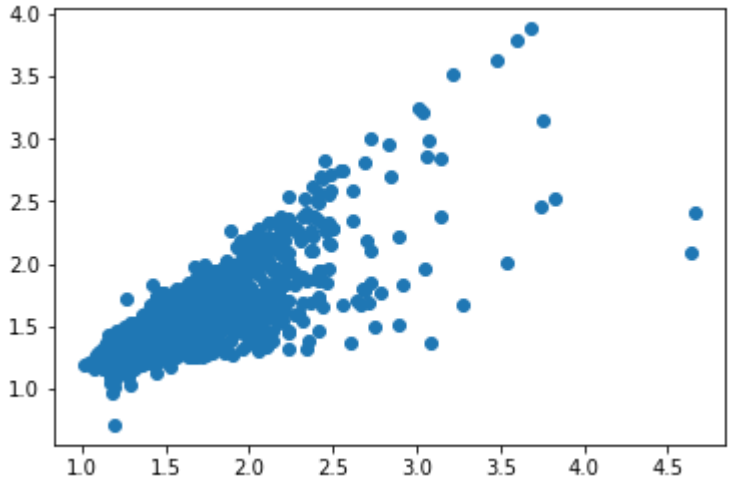
Ridge

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)
```

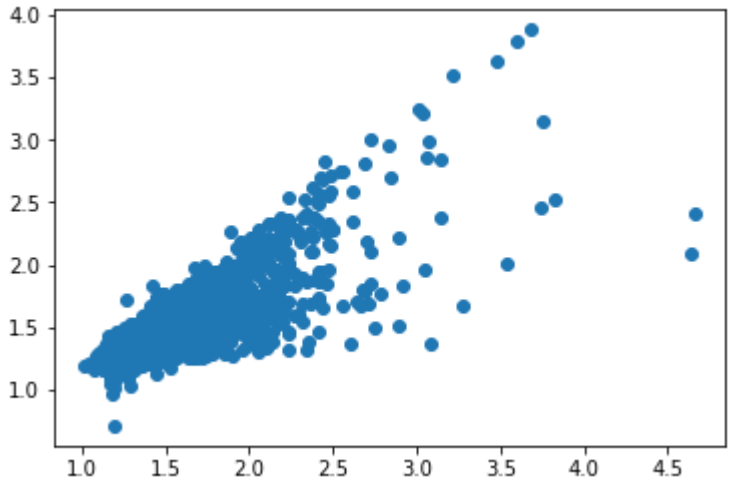
ElasticNet

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="accu
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',"
```

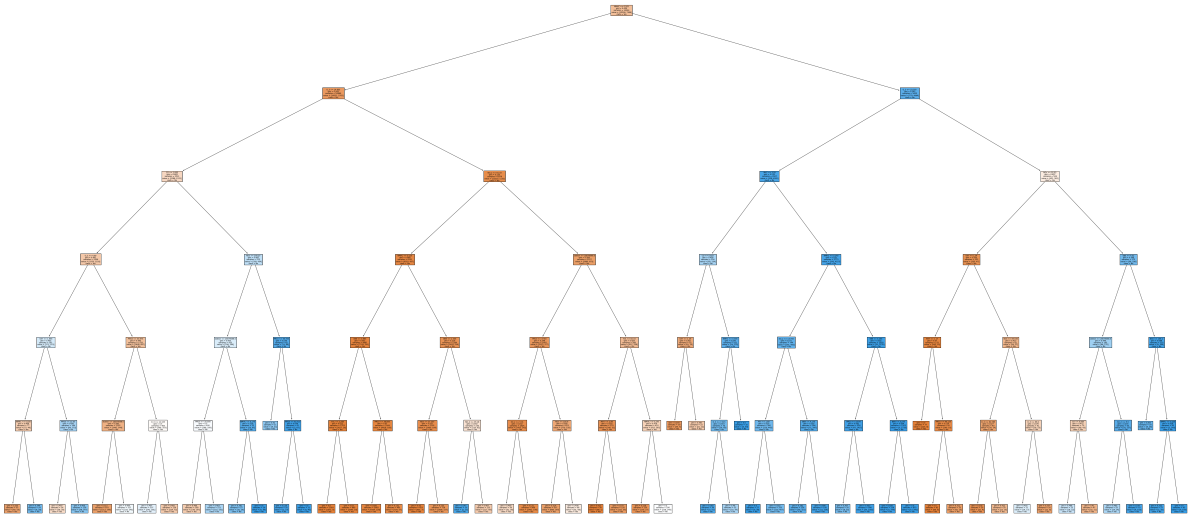
Out[89]:
```
[Text(2329.3867924528304, 2019.0857142857144, 'NMHC <= 0.205\ngini = 0.439\ns
amples = 14344\nvalue = [15300, 7366]\nclass = Yes'),
 Text(1247.6037735849056, 1708.457142857143, 'O_3 <= 14.805\ngini = 0.265\nsa
mples = 10889\nvalue = [14521, 2702]\nclass = Yes'),
 Text(642.2264150943397, 1397.8285714285716, 'CO <= 1.085\ngini = 0.483\nsamp
les = 2511\nvalue = [2288, 1570]\nclass = Yes'),
 Text(336.9056603773585, 1087.2, 'O_3 <= 4.69\ngini = 0.461\nsamples = 1996\n
value = [1979, 1116]\nclass = Yes'),
 Text(168.45283018867926, 776.5714285714287, 'OXY <= 1.665\ngini = 0.492\nsam
ples = 241\nvalue = [171, 219]\nclass = No'),
 Text(84.22641509433963, 465.9428571428573, 'NMHC <= 0.165\ngini = 0.468\nsam
ples = 77\nvalue = [77, 46]\nclass = Yes'),
 Text(42.113207547169814, 155.3142857142857, 'gini = 0.328\nsamples = 53\nval
ue = [69, 18]\nclass = Yes'),
 Text(126.33962264150944, 155.3142857142857, 'gini = 0.346\nsamples = 24\nval
ue = [8, 28]\nclass = No'),
 Text(252.67924528301887, 465.9428571428573, 'PM10 <= 15.03\ngini = 0.456\nsa
mples = 164\nvalue = [94, 173]\nclass = No'),
 Text(210.56603773584908, 155.3142857142857, 'gini = 0.487\nsamples = 36\nval
ue = [36, 26]\nclass = Yes'),
 Text(294.7924528301887, 155.3142857142857, 'gini = 0.406\nsamples = 128\nval
ue = [58, 147]\nclass = No'),
 Text(505.35849056603774, 776.5714285714287, 'PM10 <= 33.975\ngini = 0.443\ns
amples = 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\nval
ue = [1213, 299]\nclass = Yes'),
 Text(463.24528301886795, 155.3142857142857, 'gini = 0.499\nsamples = 317\nva
lue = [229, 245]\nclass = No'),
 Text(589.5849056603774, 465.9428571428573, 'O_3 <= 12.105\ngini = 0.5\nsampl
es = 463\nvalue = [366, 353]\nclass = Yes'),
 Text(547.4716981132076, 155.3142857142857, 'gini = 0.497\nsamples = 337\nval
ue = [242, 281]\nclass = No'),
 Text(631.6981132075472, 155.3142857142857, 'gini = 0.465\nsamples = 126\nval
ue = [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\nsamp
les = 400\nvalue = [288, 299]\nclass = No'),
 Text(715.9245283018869, 155.3142857142857, 'gini = 0.469\nsamples = 189\nval
ue = [178, 107]\nclass = Yes'),
 Text(800.1509433962265, 155.3142857142857, 'gini = 0.463\nsamples = 211\nval
ue = [110, 192]\nclass = No'),
 Text(926.4905660377359, 465.9428571428573, 'NMHC <= 0.175\ngini = 0.226\nsam
ples = 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\ns
amples = 62\nvalue = [11, 88]\nclass = No'),
 Text(1010.7169811320755, 465.9428571428573, 'gini = 0.42\nsamples = 21\nvalu
e = [9, 21]\nclass = No'),
 Text(1094.9433962264152, 465.9428571428573, 'PM10 <= 63.715\ngini = 0.056\ns
```

amples = 41\nvalue = [2, 67]\nclass = No'),
 Text(1052.8301886792453, 155.3142857142857, 'gini = 0.128\nsamples = 20\nval
ue = [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\ns
amples = 8378\nvalue = [12233, 1132]\nclass = Yes'),
 Text(1516.0754716981132, 1087.2, 'NMHC <= 0.125\ngini = 0.089\nsamples = 615
1\nvalue = [9353, 457]\nclass = Yes'),
 Text(1347.622641509434, 776.5714285714287, 'EBE <= 0.885\ngini = 0.051\nsamp
les = 4867\nvalue = [7535, 201]\nclass = Yes'),
 Text(1263.3962264150944, 465.9428571428573, 'TOL <= 2.285\ngini = 0.012\nsam
ples = 1703\nvalue = [2617, 16]\nclass = Yes'),
 Text(1221.2830188679245, 155.3142857142857, 'gini = 0.0\nsamples = 1018\nval
ue = [1572, 0]\nclass = Yes'),
 Text(1305.5094339622642, 155.3142857142857, 'gini = 0.03\nsamples = 685\nval
ue = [1045, 16]\nclass = Yes'),
 Text(1431.8490566037738, 465.9428571428573, 'PM10 <= 38.695\ngini = 0.07\nsa
mples = 3164\nvalue = [4918, 185]\nclass = Yes'),
 Text(1389.735849056604, 155.3142857142857, 'gini = 0.058\nsamples = 2683\nva
lue = [4164, 128]\nclass = Yes'),
 Text(1473.9622641509436, 155.3142857142857, 'gini = 0.131\nsamples = 481\nva
lue = [754, 57]\nclass = Yes'),
 Text(1684.5283018867926, 776.5714285714287, 'BEN <= 2.305\ngini = 0.216\nsam
ples = 1284\nvalue = [1818, 256]\nclass = Yes'),
 Text(1600.301886792453, 465.9428571428573, 'NOx <= 37.255\ngini = 0.161\nsam
ples = 1141\nvalue = [1699, 164]\nclass = Yes'),
 Text(1558.188679245283, 155.3142857142857, 'gini = 0.02\nsamples = 423\nvalu
e = [691, 7]\nclass = Yes'),
 Text(1642.4150943396228, 155.3142857142857, 'gini = 0.233\nsamples = 718\nva
lue = [1008, 157]\nclass = Yes'),
 Text(1768.754716981132, 465.9428571428573, 'NO_2 <= 35.18\ngini = 0.492\nsam
ples = 143\nvalue = [119, 92]\nclass = Yes'),
 Text(1726.6415094339625, 155.3142857142857, 'gini = 0.147\nsamples = 20\nval
ue = [2, 23]\nclass = No'),
 Text(1810.867924528302, 155.3142857142857, 'gini = 0.467\nsamples = 123\nval
ue = [117, 69]\nclass = Yes'),
 Text(2189.8867924528304, 1087.2, 'station <= 28079068.0\ngini = 0.308\nsampl
es = 2227\nvalue = [2880, 675]\nclass = Yes'),
 Text(2021.433962264151, 776.5714285714287, 'PM10 <= 37.83\ngini = 0.248\nsam
ples = 1647\nvalue = [2223, 377]\nclass = Yes'),
 Text(1937.2075471698115, 465.9428571428573, 'SO_2 <= 5.01\ngini = 0.162\nsam
ples = 1024\nvalue = [1453, 142]\nclass = Yes'),
 Text(1895.0943396226417, 155.3142857142857, 'gini = 0.485\nsamples = 56\nval
ue = [51, 36]\nclass = Yes'),
 Text(1979.3207547169814, 155.3142857142857, 'gini = 0.131\nsamples = 968\nva
lue = [1402, 106]\nclass = Yes'),
 Text(2105.6603773584907, 465.9428571428573, 'NOx <= 195.6\ngini = 0.358\nsam
ples = 623\nvalue = [770, 235]\nclass = Yes'),
 Text(2063.547169811321, 155.3142857142857, 'gini = 0.314\nsamples = 537\nval
ue = [689, 167]\nclass = Yes'),
 Text(2147.7735849056608, 155.3142857142857, 'gini = 0.496\nsamples = 86\nval
ue = [81, 68]\nclass = Yes'),
 Text(2358.33962264151, 776.5714285714287, 'TOL <= 8.115\ngini = 0.429\nsampl
es = 580\nvalue = [657, 298]\nclass = Yes'),
 Text(2274.11320754717, 465.9428571428573, 'MXY <= 2.98\ngini = 0.203\nsample
s = 170\nvalue = [240, 31]\nclass = Yes'),

```
 Text(2232.0, 155.3142857142857, 'gini = 0.022\nsamples = 50\nvalue = [88, 1]
\nclass = Yes'),
 Text(2316.2264150943397, 155.3142857142857, 'gini = 0.275\nsamples = 120\nva
lue = [152, 30]\nclass = Yes'),
 Text(2442.566037735849, 465.9428571428573, 'NMHC <= 0.145\ngini = 0.476\nsam
ples = 410\nvalue = [417, 267]\nclass = Yes'),
 Text(2400.4528301886794, 155.3142857142857, 'gini = 0.228\nsamples = 120\nva
lue = [179, 27]\nclass = Yes'),
 Text(2484.679245283019, 155.3142857142857, 'gini = 0.5\nsamples = 290\nvalue
= [238, 240]\nclass = No'),
 Text(3411.169811320755, 1708.457142857143, 'O_3 <= 33.295\ngini = 0.245\nsam
ples = 3455\nvalue = [779, 4664]\nclass = No'),
 Text(2884.7547169811323, 1397.8285714285716, 'MXY <= 3.59\ngini = 0.132\nsam
ples = 2927\nvalue = [328, 4281]\nclass = No'),
 Text(2653.1320754716985, 1087.2, 'CO <= 0.625\ngini = 0.458\nsamples = 154\n
value = [85, 154]\nclass = No'),
 Text(2568.905660377359, 776.5714285714287, 'OXY <= 1.29\ngini = 0.362\nsampl
es = 52\nvalue = [61, 19]\nclass = Yes'),
 Text(2526.7924528301887, 465.9428571428573, 'gini = 0.254\nsamples = 30\nval
ue = [40, 7]\nclass = Yes'),
 Text(2611.0188679245284, 465.9428571428573, 'gini = 0.463\nsamples = 22\nval
ue = [21, 12]\nclass = Yes'),
 Text(2737.3584905660377, 776.5714285714287, 'NMHC <= 0.305\ngini = 0.256\nsa
mples = 102\nvalue = [24, 135]\nclass = No'),
 Text(2695.245283018868, 465.9428571428573, 'O_3 <= 7.315\ngini = 0.343\nsamp
les = 70\nvalue = [24, 85]\nclass = No'),
 Text(2653.1320754716985, 155.3142857142857, 'gini = 0.163\nsamples = 38\nval
ue = [5, 51]\nclass = No'),
 Text(2737.3584905660377, 155.3142857142857, 'gini = 0.46\nsamples = 32\nvalu
e = [19, 34]\nclass = No'),
 Text(2779.471698113208, 465.9428571428573, 'gini = 0.0\nsamples = 32\nvalue
= [0, 50]\nclass = No'),
 Text(3116.377358490566, 1087.2, 'NMHC <= 0.245\ngini = 0.105\nsamples = 2773
\nvalue = [243, 4127]\nclass = No'),
 Text(2947.924528301887, 776.5714285714287, 'SO_2 <= 13.705\ngini = 0.271\nsa
mples = 744\nvalue = [191, 992]\nclass = No'),
 Text(2863.6981132075475, 465.9428571428573, 'PXY <= 1.69\ngini = 0.335\nsamp
les = 360\nvalue = [120, 445]\nclass = No'),
 Text(2821.5849056603774, 155.3142857142857, 'gini = 0.0\nsamples = 27\nvalue
= [0, 38]\nclass = No'),
 Text(2905.811320754717, 155.3142857142857, 'gini = 0.352\nsamples = 333\nval
ue = [120, 407]\nclass = No'),
 Text(3032.1509433962265, 465.9428571428573, 'BEN <= 4.175\ngini = 0.203\nsam
ples = 384\nvalue = [71, 547]\nclass = No'),
 Text(2990.037735849057, 155.3142857142857, 'gini = 0.262\nsamples = 226\nval
ue = [59, 321]\nclass = No'),
 Text(3074.2641509433965, 155.3142857142857, 'gini = 0.096\nsamples = 158\nva
lue = [12, 226]\nclass = No'),
 Text(3284.8301886792456, 776.5714285714287, 'EBE <= 5.265\ngini = 0.032\nsam
ples = 2029\nvalue = [52, 3135]\nclass = No'),
 Text(3200.603773584906, 465.9428571428573, 'PM10 <= 18.44\ngini = 0.062\nsam
ples = 926\nvalue = [47, 1417]\nclass = No'),
 Text(3158.4905660377362, 155.3142857142857, 'gini = 0.367\nsamples = 20\nval
ue = [8, 25]\nclass = No'),
 Text(3242.7169811320755, 155.3142857142857, 'gini = 0.053\nsamples = 906\nva
lue = [39, 1392]\nclass = 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\nval
ue = [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\nsa
mples = 528\nvalue = [451, 383]\nclass = Yes'),
 Text(3642.7924528301887, 1087.2, 'OXY <= 1.34\ngini = 0.206\nsamples = 253\n
value = [355, 47]\nclass = Yes'),
 Text(3495.3962264150946, 776.5714285714287, 'MXY <= 1.17\ngini = 0.07\nsampl
es = 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\nsa
mples = 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\nval
ue = [42, 10]\nclass = Yes'),
 Text(3790.1886792452833, 776.5714285714287, 'NOx <= 59.535\ngini = 0.413\nsa
mples = 81\nvalue = [90, 37]\nclass = Yes'),
 Text(3705.9622641509436, 465.9428571428573, 'O_3 <= 62.585\ngini = 0.327\nsa
mples = 40\nvalue = [54, 14]\nclass = Yes'),
 Text(3663.849056603774, 155.3142857142857, 'gini = 0.202\nsamples = 20\nvalu
e = [31, 4]\nclass = Yes'),
 Text(3748.0754716981137, 155.3142857142857, 'gini = 0.422\nsamples = 20\nval
ue = [23, 10]\nclass = Yes'),
 Text(3874.415094339623, 465.9428571428573, 'O_3 <= 66.0\ngini = 0.476\nsampl
es = 41\nvalue = [36, 23]\nclass = Yes'),
 Text(3832.301886792453, 155.3142857142857, 'gini = 0.495\nsamples = 21\nvalu
e = [14, 17]\nclass = No'),
 Text(3916.5283018867926, 155.3142857142857, 'gini = 0.337\nsamples = 20\nval
ue = [22, 6]\nclass = Yes'),
 Text(4232.377358490567, 1087.2, 'EBE <= 4.52\ngini = 0.346\nsamples = 275\nv
alue = [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\nsampl
es = 70\nvalue = [68, 43]\nclass = Yes'),
 Text(4000.7547169811323, 155.3142857142857, 'gini = 0.488\nsamples = 23\nval
ue = [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\nsampl
es = 95\nvalue = [20, 127]\nclass = No'),
 Text(4169.207547169812, 155.3142857142857, 'gini = 0.465\nsamples = 25\nvalu
e = [14, 24]\nclass = No'),
 Text(4253.433962264151, 155.3142857142857, 'gini = 0.104\nsamples = 70\nvalu
e = [6, 103]\nclass = No'),
 Text(4337.660377358491, 776.5714285714287, 'NOx <= 146.1\ngini = 0.088\nsamp
les = 110\nvalue = [8, 166]\nclass = No'),
 Text(4295.5471698113215, 465.9428571428573, 'gini = 0.305\nsamples = 20\nval
ue = [6, 26]\nclass = No'),
 Text(4379.773584905661, 465.9428571428573, 'NMHC <= 0.225\ngini = 0.028\nsam
ples = 90\nvalue = [2, 140]\nclass = No'),
 Text(4337.660377358491, 155.3142857142857, 'gini = 0.102\nsamples = 27\nvalu
e = [2, 35]\nclass = No'),
```

```
 Text(4421.88679245283, 155.3142857142857, 'gini = 0.0\nsamples = 63\nvalue =
[0, 105]\nclass = 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

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,Rid
from sklearn.model_selection import train_test_split
```

In [2]:
```
df=pd.read_csv("madrid_2003.csv")
df
```

Out[2]:

	date	BEN	CO	EBE	MXY	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	55.209
1	2003-03-01 01:00:00	NaN	1.45	NaN	NaN	0.26	72.110001	250.000000	0.73	6.720000	52.389
2	2003-03-01 01:00:00	NaN	1.57	NaN	NaN	NaN	80.559998	224.199997	NaN	21.049999	63.240
3	2003-03-01 01:00:00	NaN	2.45	NaN	NaN	NaN	78.370003	450.399994	NaN	4.220000	67.839
4	2003-03-01 01:00:00	NaN	3.26	NaN	NaN	NaN	96.250000	479.100006	NaN	8.460000	95.779
...	...	...	...	...	...	...	...	...	...	...	
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	7.380
243980	2003-10-01 00:00:00	0.32	0.08	0.36	0.72	NaN	10.450000	14.760000	1.00	34.610001	7.400
243981	2003-10-01 00:00:00	NaN	NaN	NaN	NaN	0.07	34.639999	50.810001	NaN	32.160000	16.830
243982	2003-10-01 00:00:00	NaN	NaN	NaN	NaN	0.07	32.580002	41.020000	NaN	NaN	13.570
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	12.350

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]:

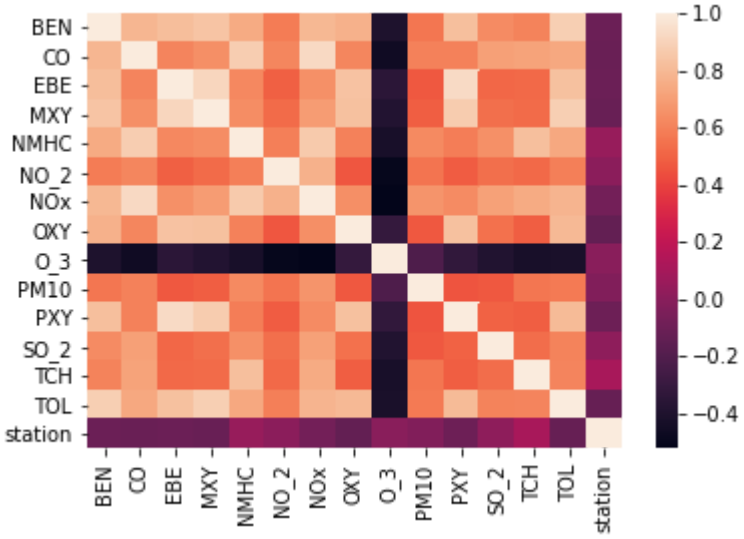
	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	
5	2003-03-01 01:00:00	8.41	1.94	9.83	21.49	0.45	90.300003	384.899994	9.48	9.950000	95.15
23	2003-03-01 01:00:00	3.46	1.27	3.43	7.08	0.18	54.250000	173.300003	3.37	6.540000	53.00
27	2003-03-01 01:00:00	6.39	1.79	5.75	10.88	0.33	75.459999	281.100006	3.68	6.690000	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	58.88
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	25.68
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	7.38
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)
```

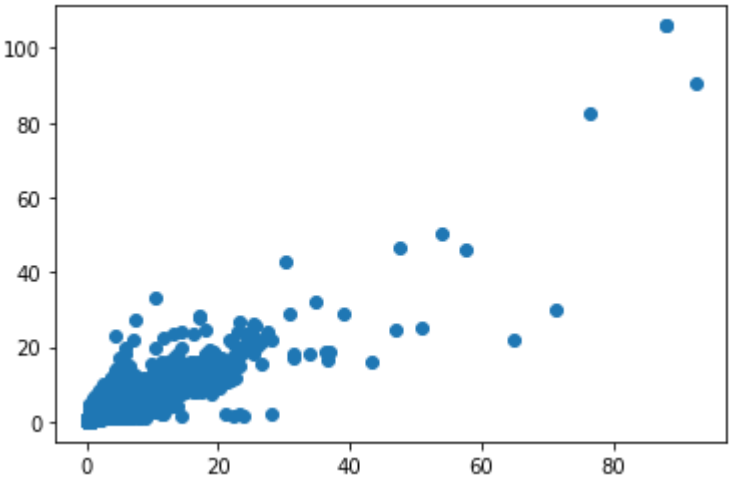
In [6]:
```
sns.heatmap(df1.corr())
```

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)
```

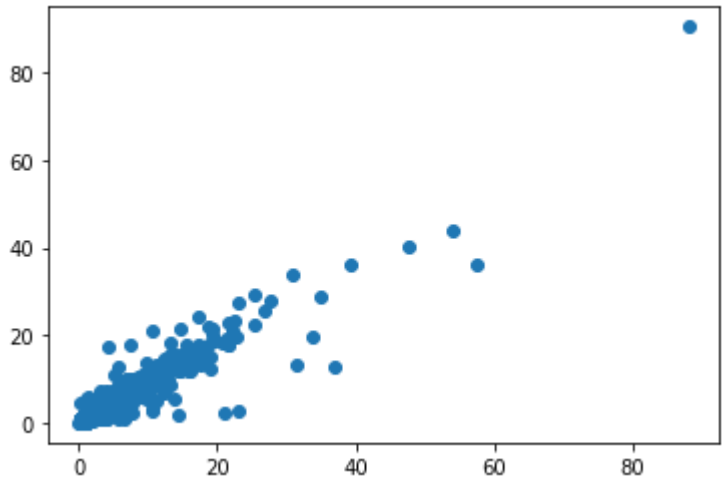
Linear

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
```

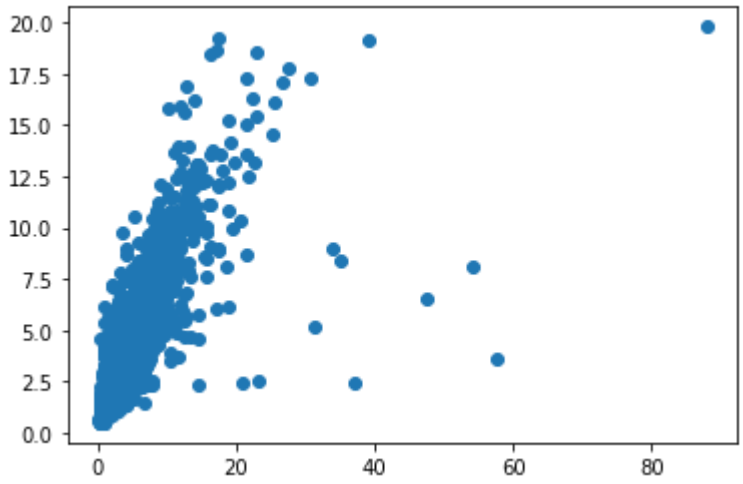
Lasso

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)
```

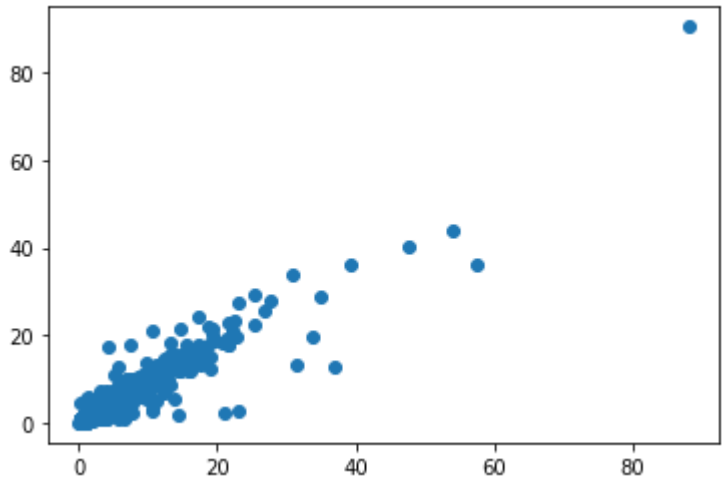
Ridge

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)
```

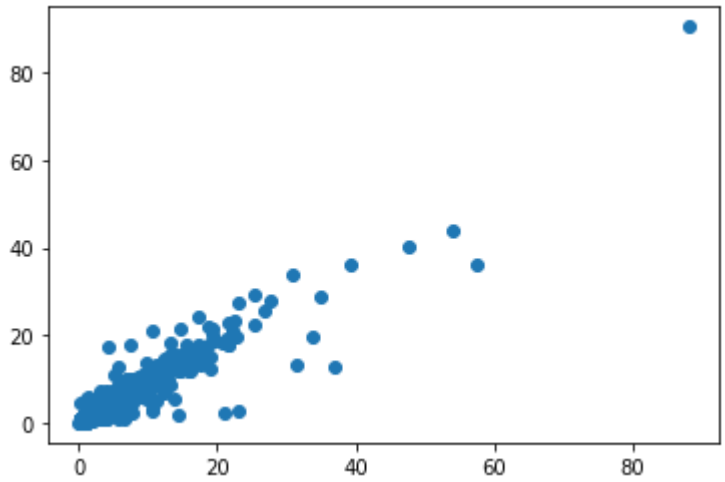
ElasticNet

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="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',"
```

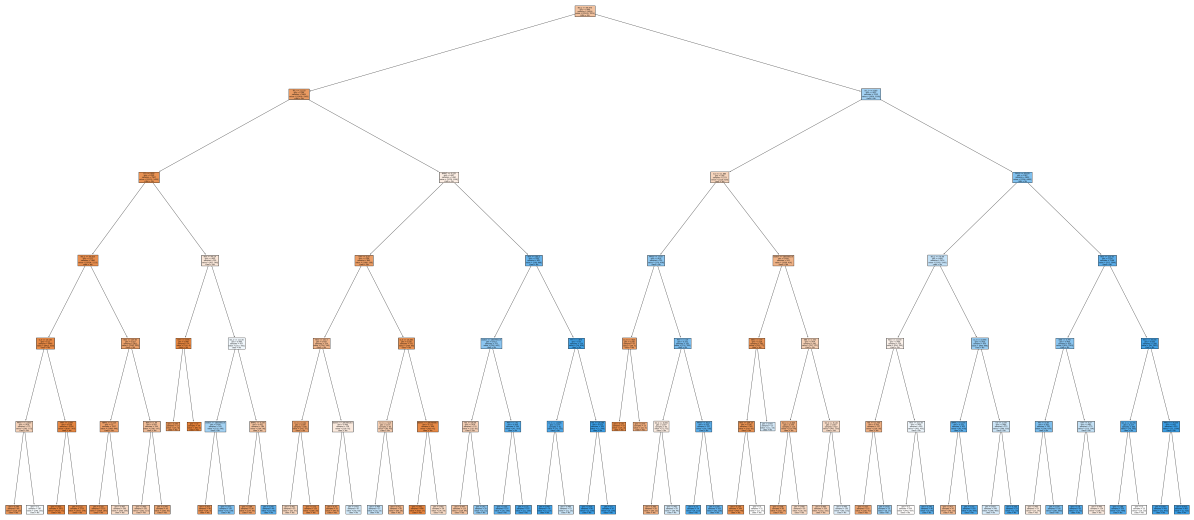
Out[39]:
```
[Text(2184.6696428571427, 2019.0857142857144, 'NO_2 <= 66.475\ngini = 0.449\n
samples = 14618\nvalue = [15233, 7874]\nclass = Yes'),
 Text(1111.017857142857, 1708.457142857143, 'TOL <= 11.125\ngini = 0.284\nsam
ples = 9483\nvalue = [12409, 2565]\nclass = Yes'),
 Text(548.0357142857142, 1397.8285714285716,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

```
e = [10, 60]\nclass = No'),
 Text(1673.9999999999998, 1397.8285714285716, 'NMHC <= 0.155\ngini = 0.497\ns
amples = 1496\nvalue = [1278, 1106]\nclass = Yes'),
 Text(1355.142857142857, 1087.2, 'OXY <= 3.16\ngini = 0.296\nsamples = 845\nv
alue = [1080, 238]\nclass = Yes'),
 Text(1195.7142857142856, 776.5714285714287, 'NOx <= 101.4\ngini = 0.418\nsam
ples = 314\nvalue = [350, 148]\nclass = Yes'),
 Text(1116.0, 465.9428571428573, 'O_3 <= 13.42\ngini = 0.309\nsamples = 186\n
value = [241, 57]\nclass = Yes'),
 Text(1076.142857142857, 155.3142857142857, 'gini = 0.481\nsamples = 53\nvalu
e = [55, 37]\nclass = Yes'),
 Text(1155.8571428571427, 155.3142857142857, 'gini = 0.175\nsamples = 133\nva
lue = [186, 20]\nclass = Yes'),
 Text(1275.4285714285713, 465.9428571428573, 'station <= 28079015.0\ngini =
0.496\nsamples = 128\nvalue = [109, 91]\nclass = Yes'),
 Text(1235.5714285714284, 155.3142857142857, 'gini = 0.26\nsamples = 31\nvalu
e = [44, 8]\nclass = Yes'),
 Text(1315.2857142857142, 155.3142857142857, 'gini = 0.493\nsamples = 97\nval
ue = [65, 83]\nclass = No'),
 Text(1514.5714285714284, 776.5714285714287, 'O_3 <= 13.285\ngini = 0.195\nsa
mples = 531\nvalue = [730, 90]\nclass = Yes'),
 Text(1434.8571428571427, 465.9428571428573, 'O_3 <= 7.215\ngini = 0.471\nsam
ples = 85\nvalue = [80, 49]\nclass = Yes'),
 Text(1395.0, 155.3142857142857, 'gini = 0.484\nsamples = 27\nvalue = [16, 2
3]\nclass = No'),
 Text(1474.7142857142856, 155.3142857142857, 'gini = 0.411\nsamples = 58\nval
ue = [64, 26]\nclass = Yes'),
 Text(1594.2857142857142, 465.9428571428573, 'station <= 28079068.0\ngini =
0.112\nsamples = 446\nvalue = [650, 41]\nclass = Yes'),
 Text(1554.4285714285713, 155.3142857142857, 'gini = 0.052\nsamples = 408\nva
lue = [616, 17]\nclass = Yes'),
 Text(1634.142857142857, 155.3142857142857, 'gini = 0.485\nsamples = 38\nvalu
e = [34, 24]\nclass = Yes'),
 Text(1992.8571428571427, 1087.2, 'NOx <= 185.2\ngini = 0.302\nsamples = 651
\nvalue = [198, 868]\nclass = No'),
 Text(1833.4285714285713, 776.5714285714287, 'station <= 28079015.0\ngini =
0.413\nsamples = 377\nvalue = [182, 442]\nclass = No'),
 Text(1753.7142857142856, 465.9428571428573, 'TOL <= 28.06\ngini = 0.474\nsam
ples = 133\nvalue = [135, 85]\nclass = Yes'),
 Text(1713.8571428571427, 155.3142857142857, 'gini = 0.447\nsamples = 113\nva
lue = [126, 64]\nclass = Yes'),
 Text(1793.5714285714284, 155.3142857142857, 'gini = 0.42\nsamples = 20\nvalu
e = [9, 21]\nclass = No'),
 Text(1913.1428571428569, 465.9428571428573, 'MXY <= 7.41\ngini = 0.206\nsamp
les = 244\nvalue = [47, 357]\nclass = No'),
 Text(1873.2857142857142, 155.3142857142857, 'gini = 0.255\nsamples = 137\nva
lue = [35, 198]\nclass = No'),
 Text(1952.9999999999998, 155.3142857142857, 'gini = 0.131\nsamples = 107\nva
lue = [12, 159]\nclass = No'),
 Text(2152.285714285714, 776.5714285714287, 'PXY <= 3.875\ngini = 0.07\nsampl
es = 274\nvalue = [16, 426]\nclass = No'),
 Text(2072.5714285714284, 465.9428571428573, 'O_3 <= 13.895\ngini = 0.173\nsa
mples = 91\nvalue = [14, 132]\nclass = No'),
 Text(2032.7142857142856, 155.3142857142857, 'gini = 0.049\nsamples = 76\nval
ue = [3, 117]\nclass = No'),
 Text(2112.428571428571, 155.3142857142857, 'gini = 0.488\nsamples = 15\nvalu
e = [11, 15]\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\nvalu
e = [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\nsa
mples = 5135\nvalue = [2824, 5309]\nclass = No'),
 Text(2690.3571428571427, 1397.8285714285716, 'O_3 <= 11.885\ngini = 0.48\nsa
mples = 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\nsam
ples = 31\nvalue = [39, 7]\nclass = Yes'),
 Text(2311.7142857142853, 465.9428571428573, 'gini = 0.142\nsamples = 16\nval
ue = [24, 2]\nclass = Yes'),
 Text(2391.428571428571, 465.9428571428573, 'gini = 0.375\nsamples = 15\nvalu
e = [15, 5]\nclass = Yes'),
 Text(2550.8571428571427, 776.5714285714287, 'MXY <= 3.29\ngini = 0.372\nsamp
les = 320\nvalue = [131, 399]\nclass = No'),
 Text(2471.142857142857, 465.9428571428573, 'SO_2 <= 6.895\ngini = 0.497\nsam
ples = 83\nvalue = [72, 61]\nclass = Yes'),
 Text(2431.285714285714, 155.3142857142857, 'gini = 0.386\nsamples = 28\nvalu
e = [34, 12]\nclass = Yes'),
 Text(2511.0, 155.3142857142857, 'gini = 0.492\nsamples = 55\nvalue = [38, 4
9]\nclass = No'),
 Text(2630.5714285714284, 465.9428571428573, 'PM10 <= 36.21\ngini = 0.253\nsa
mples = 237\nvalue = [59, 338]\nclass = No'),
 Text(2590.7142857142853, 155.3142857142857, 'gini = 0.14\nsamples = 74\nvalu
e = [9, 110]\nclass = No'),
 Text(2670.428571428571, 155.3142857142857, 'gini = 0.295\nsamples = 163\nval
ue = [50, 228]\nclass = No'),
 Text(2929.5, 1087.2, 'station <= 28079015.0\ngini = 0.404\nsamples = 922\nva
lue = [1058, 414]\nclass = Yes'),
 Text(2829.8571428571427, 776.5714285714287, 'BEN <= 5.59\ngini = 0.224\nsamp
les = 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\nval
ue = [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\nval
ue = [11, 14]\nclass = No'),
 Text(3029.142857142857, 776.5714285714287, 'EBE <= 1.435\ngini = 0.474\nsamp
les = 542\nvalue = [537, 337]\nclass = Yes'),
 Text(2949.428571428571, 465.9428571428573, 'SO_2 <= 8.805\ngini = 0.378\nsam
ples = 115\nvalue = [130, 44]\nclass = Yes'),
 Text(2909.5714285714284, 155.3142857142857, 'gini = 0.308\nsamples = 85\nval
ue = [102, 24]\nclass = Yes'),
 Text(2989.285714285714, 155.3142857142857, 'gini = 0.486\nsamples = 30\nvalu
e = [28, 20]\nclass = Yes'),
 Text(3108.8571428571427, 465.9428571428573, 'SO_2 <= 9.145\ngini = 0.487\nsa
mples = 427\nvalue = [407, 293]\nclass = Yes'),
 Text(3068.9999999999995, 155.3142857142857, 'gini = 0.468\nsamples = 324\nva
lue = [322, 192]\nclass = Yes'),
 Text(3148.7142857142853, 155.3142857142857, 'gini = 0.496\nsamples = 103\nva
```

```
lue = [85, 101]\nclass = No'),
 Text(3826.2857142857138, 1397.8285714285716, 'PM10 <= 43.635\ngini = 0.387\n
samples = 3862\nvalue = [1596, 4489]\nclass = No'),
 Text(3507.428571428571, 1087.2, 'NO_2 <= 76.99\ngini = 0.484\nsamples = 1674
\nvalue = [1079, 1545]\nclass = No'),
 Text(3347.9999999999995, 776.5714285714287, 'NOx <= 111.55\ngini = 0.499\nsa
mples = 739\nvalue = [610, 549]\nclass = Yes'),
 Text(3268.285714285714, 465.9428571428573, 'TOL <= 11.755\ngini = 0.393\nsam
ples = 137\nvalue = [158, 58]\nclass = Yes'),
 Text(3228.428571428571, 155.3142857142857, 'gini = 0.337\nsamples = 122\nval
ue = [150, 41]\nclass = Yes'),
 Text(3308.142857142857, 155.3142857142857, 'gini = 0.435\nsamples = 15\nvalu
e = [8, 17]\nclass = No'),
 Text(3427.7142857142853, 465.9428571428573, 'NOx <= 211.65\ngini = 0.499\nsa
mples = 602\nvalue = [452, 491]\nclass = No'),
 Text(3387.8571428571427, 155.3142857142857, 'gini = 0.5\nsamples = 554\nvalu
e = [443, 437]\nclass = Yes'),
 Text(3467.5714285714284, 155.3142857142857, 'gini = 0.245\nsamples = 48\nval
ue = [9, 54]\nclass = No'),
 Text(3666.8571428571427, 776.5714285714287, 'O_3 <= 10.695\ngini = 0.435\nsa
mples = 935\nvalue = [469, 996]\nclass = No'),
 Text(3587.142857142857, 465.9428571428573, 'NMHC <= 0.145\ngini = 0.252\nsam
ples = 363\nvalue = [83, 478]\nclass = No'),
 Text(3547.285714285714, 155.3142857142857, 'gini = 0.417\nsamples = 47\nvalu
e = [45, 19]\nclass = Yes'),
 Text(3626.9999999999995, 155.3142857142857, 'gini = 0.141\nsamples = 316\nva
lue = [38, 459]\nclass = No'),
 Text(3746.5714285714284, 465.9428571428573, 'CO <= 1.575\ngini = 0.489\nsamp
les = 572\nvalue = [386, 518]\nclass = No'),
 Text(3706.7142857142853, 155.3142857142857, 'gini = 0.493\nsamples = 549\nva
lue = [384, 485]\nclass = No'),
 Text(3786.428571428571, 155.3142857142857, 'gini = 0.108\nsamples = 23\nvalu
e = [2, 33]\nclass = No'),
 Text(4145.142857142857, 1087.2, 'NOx <= 215.95\ngini = 0.254\nsamples = 2188
\nvalue = [517, 2944]\nclass = No'),
 Text(3985.7142857142853, 776.5714285714287, 'EBE <= 4.795\ngini = 0.409\nsam
ples = 920\nvalue = [419, 1043]\nclass = No'),
 Text(3905.9999999999995, 465.9428571428573, 'PXY <= 0.97\ngini = 0.382\nsamp
les = 745\nvalue = [304, 879]\nclass = No'),
 Text(3866.142857142857, 155.3142857142857, 'gini = 0.495\nsamples = 19\nvalu
e = [17, 14]\nclass = Yes'),
 Text(3945.8571428571427, 155.3142857142857, 'gini = 0.374\nsamples = 726\nva
lue = [287, 865]\nclass = No'),
 Text(4065.428571428571, 465.9428571428573, 'OXY <= 5.27\ngini = 0.485\nsampl
es = 175\nvalue = [115, 164]\nclass = No'),
 Text(4025.5714285714284, 155.3142857142857, 'gini = 0.16\nsamples = 51\nvalu
e = [7, 73]\nclass = No'),
 Text(4105.285714285714, 155.3142857142857, 'gini = 0.496\nsamples = 124\nval
ue = [108, 91]\nclass = Yes'),
 Text(4304.571428571428, 776.5714285714287, 'NOx <= 261.85\ngini = 0.093\nsam
ples = 1268\nvalue = [98, 1901]\nclass = No'),
 Text(4224.857142857142, 465.9428571428573, 'SO_2 <= 32.705\ngini = 0.227\nsa
mples = 358\nvalue = [76, 505]\nclass = No'),
 Text(4185.0, 155.3142857142857, 'gini = 0.188\nsamples = 335\nvalue = [57, 4
86]\nclass = No'),
 Text(4264.714285714285, 155.3142857142857, 'gini = 0.5\nsamples = 23\nvalue
= [19, 19]\nclass = Yes'),
```

```
 Text(4384.285714285714, 465.9428571428573, 'PM10 <= 76.335\ngini = 0.031\nsa
mples = 910\nvalue = [22, 1396]\nclass = No'),
 Text(4344.428571428571, 155.3142857142857, 'gini = 0.059\nsamples = 443\nval
ue = [21, 673]\nclass = No'),
 Text(4424.142857142857, 155.3142857142857, 'gini = 0.003\nsamples = 467\nval
ue = [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	MXY	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	39.99(
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	22.95(
2	2004-08-01 01:00:00	NaN	1.02	NaN	NaN	NaN	93.389999	138.600006	NaN	20.860001	49.48(
3	2004-08-01 01:00:00	NaN	0.53	NaN	NaN	NaN	87.290001	105.000000	NaN	36.730000	31.07(
4	2004-08-01 01:00:00	NaN	0.17	NaN	NaN	NaN	34.910000	35.349998	NaN	86.269997	54.08(
...	...	...	...	...	...	...	...	...	...	...	
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	30.90(
245492	2004-06-01 00:00:00	2.49	0.75	2.44	4.57	NaN	97.139999	146.899994	2.34	7.740000	37.68
245493	2004-06-01 00:00:00	NaN	NaN	NaN	NaN	0.13	102.699997	132.600006	NaN	17.809999	22.84(
245494	2004-06-01 00:00:00	NaN	NaN	NaN	NaN	0.09	82.599998	102.599998	NaN	NaN	45.63(
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	24.38

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]:

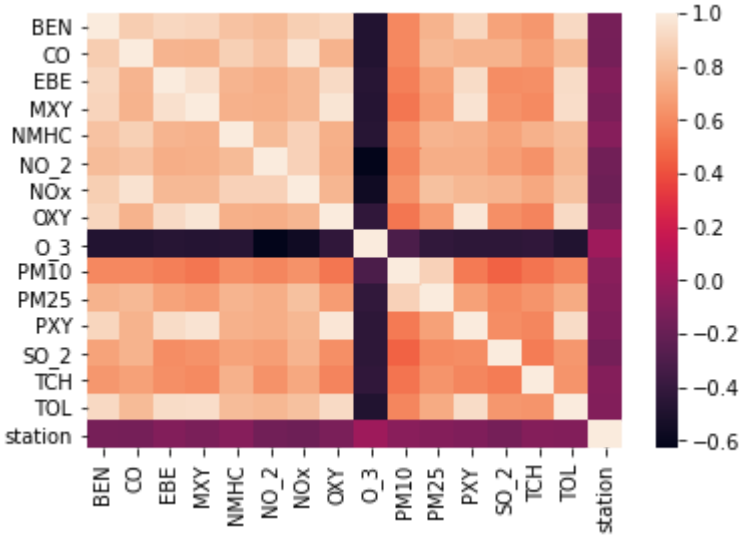
	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	P
5	2004-08-01 01:00:00	3.24	0.63	5.55	9.72	0.06	103.800003	144.800003	5.04	32.480000	59.11(
22	2004-08-01 01:00:00	0.55	0.36	0.54	0.86	0.07	31.980000	32.799999	0.50	79.040001	43.54(
26	2004-08-01 01:00:00	1.80	0.46	2.28	4.62	0.21	62.259998	75.470001	2.47	54.419998	46.63(
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	86.93(
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	48.95(
...	...	...	...	...	...	...	...	...	...	...	
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	19.29(
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	31.80(
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	21.21(
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	30.90(
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	24.38(

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

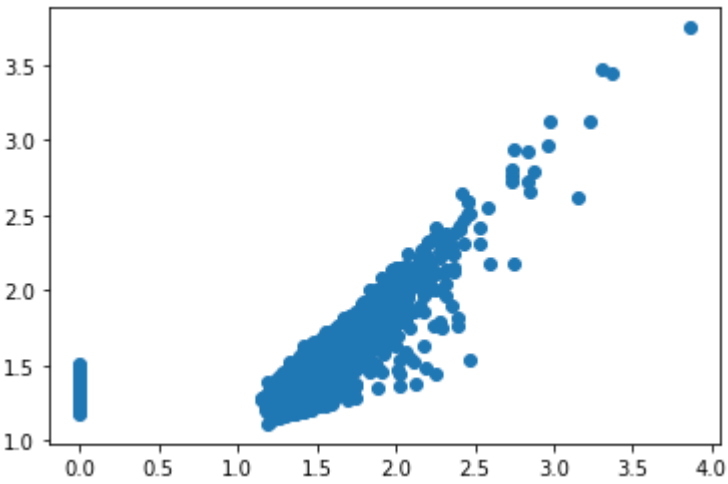
In [47]:
```
li=LinearRegression()
li.fit(x_train,y_train)
```

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 [ ]:
```

```

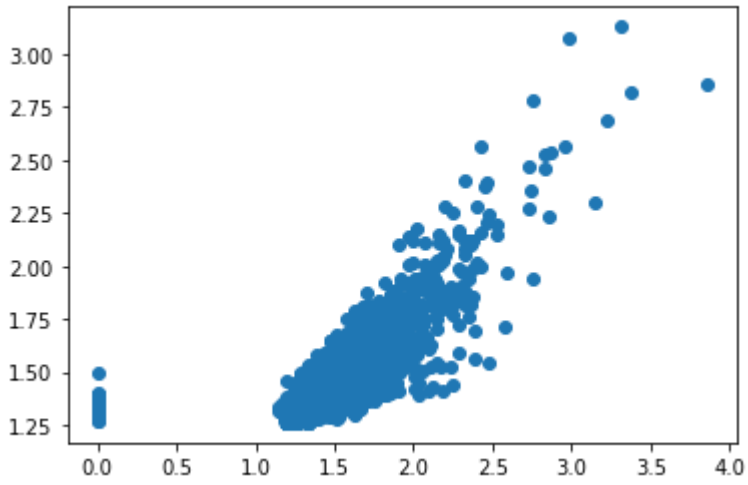
Lasso

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)
```

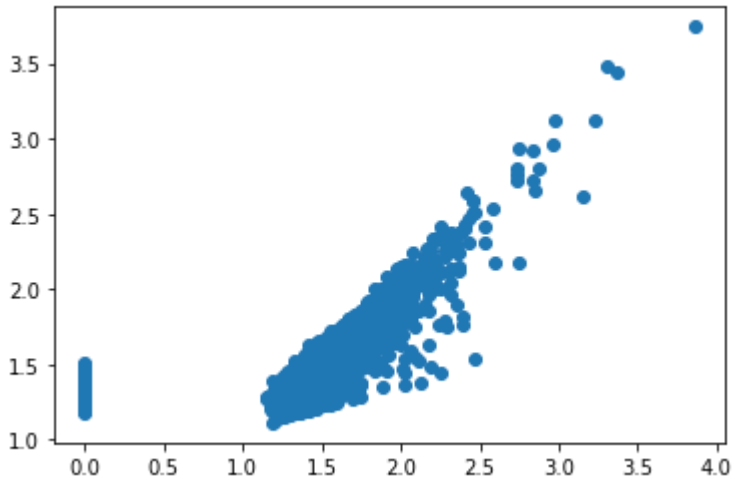
Ridge

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)
```

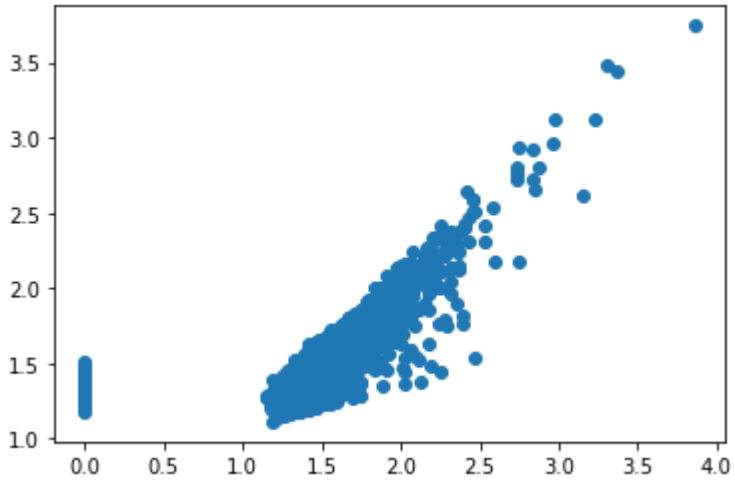
ElasticNet

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="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',"
```

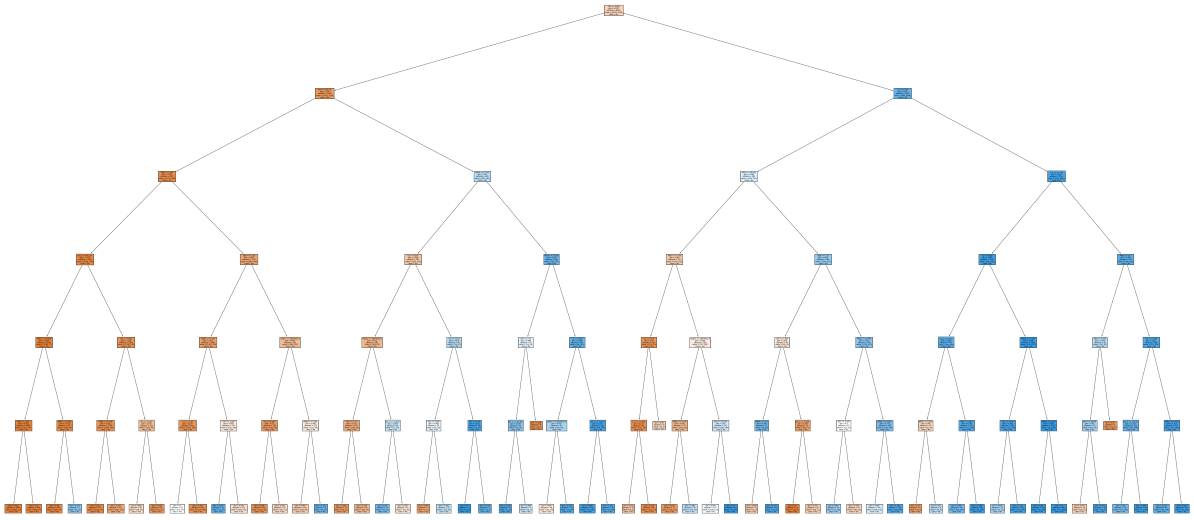
Out[75]:
```
[Text(2292.129310344828, 2019.0857142857144, 'TOL <= 9.985\ngini = 0.476\nsam
ples = 8605\nvalue = [8279, 5298]\nclass = Yes'),
 Text(1207.3965517241381, 1708.457142857143, 'NOx <= 142.25\ngini = 0.25\nsam
ples = 5526\nvalue = [7419, 1270]\nclass = Yes'),
 Text(615.7241379310345, 1397.8285714285716,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

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

```
 Text(2039.5862068965519, 155.3142857142857, 'gini = 0.496\nsamples = 48\nval
ue = [42, 35]\nclass = Yes'),
 Text(2116.551724137931, 155.3142857142857, 'gini = 0.222\nsamples = 43\nvalu
e = [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\nva
lue = [10, 182]\nclass = No'),
 Text(2270.4827586206898, 155.3142857142857, 'gini = 0.241\nsamples = 30\nval
ue = [8, 49]\nclass = No'),
 Text(3376.8620689655177, 1708.457142857143, 'CO <= 0.755\ngini = 0.29\nsampl
es = 3079\nvalue = [860, 4028]\nclass = No'),
 Text(2799.6206896551726, 1397.8285714285716, 'PM10 <= 30.095\ngini = 0.496\n
samples = 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\nsampl
es = 104\nvalue = [140, 24]\nclass = Yes'),
 Text(2385.9310344827586, 465.9428571428573, 'O_3 <= 14.655\ngini = 0.212\nsa
mples = 95\nvalue = [131, 18]\nclass = Yes'),
 Text(2347.4482758620693, 155.3142857142857, 'gini = 0.459\nsamples = 10\nval
ue = [9, 5]\nclass = Yes'),
 Text(2424.4137931034484, 155.3142857142857, 'gini = 0.174\nsamples = 85\nval
ue = [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\nsa
mples = 117\nvalue = [125, 55]\nclass = Yes'),
 Text(2501.379310344828, 155.3142857142857, 'gini = 0.276\nsamples = 77\nvalu
e = [101, 20]\nclass = Yes'),
 Text(2578.344827586207, 155.3142857142857, 'gini = 0.483\nsamples = 40\nvalu
e = [24, 35]\nclass = No'),
 Text(2693.7931034482763, 465.9428571428573, 'NOx <= 137.05\ngini = 0.49\nsam
ples = 151\nvalue = [106, 140]\nclass = No'),
 Text(2655.3103448275865, 155.3142857142857, 'gini = 0.499\nsamples = 137\nva
lue = [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\n
value = [243, 517]\nclass = No'),
 Text(2924.689655172414, 776.5714285714287, 'OXY <= 2.485\ngini = 0.495\nsamp
les = 49\nvalue = [48, 39]\nclass = Yes'),
 Text(2847.724137931035, 465.9428571428573, 'NMHC <= 0.11\ngini = 0.346\nsamp
les = 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\nvalu
e = [3, 25]\nclass = No'),
 Text(3001.6551724137935, 465.9428571428573, 'SO_2 <= 10.045\ngini = 0.338\ns
amples = 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\nvalu
e = [18, 11]\nclass = Yes'),
 Text(3232.551724137931, 776.5714285714287, 'PM25 <= 18.465\ngini = 0.412\nsa
```

```
mples = 435\nvalue = [195, 478]\nclass = No'),
 Text(3155.586206896552, 465.9428571428573, 'OXY <= 3.155\ngini = 0.5\nsample
s = 67\nvalue = [53, 52]\nclass = Yes'),
 Text(3117.1034482758623, 155.3142857142857, 'gini = 0.444\nsamples = 25\nval
ue = [13, 26]\nclass = No'),
 Text(3194.068965517242, 155.3142857142857, 'gini = 0.478\nsamples = 42\nvalu
e = [40, 26]\nclass = Yes'),
 Text(3309.5172413793107, 465.9428571428573, 'NMHC <= 0.165\ngini = 0.375\nsa
mples = 368\nvalue = [142, 426]\nclass = No'),
 Text(3271.034482758621, 155.3142857142857, 'gini = 0.494\nsamples = 160\nval
ue = [106, 132]\nclass = No'),
 Text(3348.0000000000005, 155.3142857142857, 'gini = 0.194\nsamples = 208\nva
lue = [36, 294]\nclass = No'),
 Text(3954.1034482758623, 1397.8285714285716, 'O_3 <= 15.745\ngini = 0.129\ns
amples = 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\nsa
mples = 176\nvalue = [44, 254]\nclass = No'),
 Text(3463.4482758620693, 465.9428571428573, 'NMHC <= 0.155\ngini = 0.48\nsam
ples = 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\nsamp
les = 162\nvalue = [32, 246]\nclass = No'),
 Text(3578.896551724138, 155.3142857142857, 'gini = 0.349\nsamples = 76\nvalu
e = [29, 100]\nclass = No'),
 Text(3655.8620689655177, 155.3142857142857, 'gini = 0.039\nsamples = 86\nval
ue = [3, 146]\nclass = No'),
 Text(3848.275862068966, 776.5714285714287, 'NMHC <= 0.235\ngini = 0.034\nsam
ples = 1369\nvalue = [38, 2137]\nclass = No'),
 Text(3771.3103448275865, 465.9428571428573, 'TOL <= 10.47\ngini = 0.145\nsam
ples = 246\nvalue = [32, 374]\nclass = No'),
 Text(3732.8275862068967, 155.3142857142857, 'gini = 0.455\nsamples = 15\nval
ue = [7, 13]\nclass = No'),
 Text(3809.7931034482763, 155.3142857142857, 'gini = 0.121\nsamples = 231\nva
lue = [25, 361]\nclass = No'),
 Text(3925.241379310345, 465.9428571428573, 'NMHC <= 0.275\ngini = 0.007\nsam
ples = 1123\nvalue = [6, 1763]\nclass = No'),
 Text(3886.7586206896553, 155.3142857142857, 'gini = 0.041\nsamples = 181\nva
lue = [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\n
value = [164, 901]\nclass = No'),
 Text(4117.6551724137935, 776.5714285714287, 'BEN <= 4.675\ngini = 0.476\nsam
ples = 80\nvalue = [52, 81]\nclass = No'),
 Text(4079.1724137931037, 465.9428571428573, 'NOx <= 168.05\ngini = 0.45\nsam
ples = 75\nvalue = [41, 79]\nclass = No'),
 Text(4040.689655172414, 155.3142857142857, 'gini = 0.474\nsamples = 39\nvalu
e = [35, 22]\nclass = Yes'),
 Text(4117.6551724137935, 155.3142857142857, 'gini = 0.172\nsamples = 36\nval
ue = [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\nsamp
les = 598\nvalue = [112, 820]\nclass = No'),
 Text(4233.103448275862, 465.9428571428573, 'NO_2 <= 103.45\ngini = 0.313\nsa
mples = 275\nvalue = [81, 336]\nclass = No'),
 Text(4194.620689655173, 155.3142857142857, 'gini = 0.36\nsamples = 215\nvalu
e = [77, 250]\nclass = No'),
 Text(4271.586206896552, 155.3142857142857, 'gini = 0.085\nsamples = 60\nvalu
e = [4, 86]\nclass = No'),
 Text(4387.034482758621, 465.9428571428573, 'NMHC <= 0.225\ngini = 0.113\nsam
ples = 323\nvalue = [31, 484]\nclass = No'),
 Text(4348.551724137931, 155.3142857142857, 'gini = 0.279\nsamples = 92\nvalu
e = [24, 119]\nclass = No'),
 Text(4425.517241379311, 155.3142857142857, 'gini = 0.037\nsamples = 231\nval
ue = [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

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]:

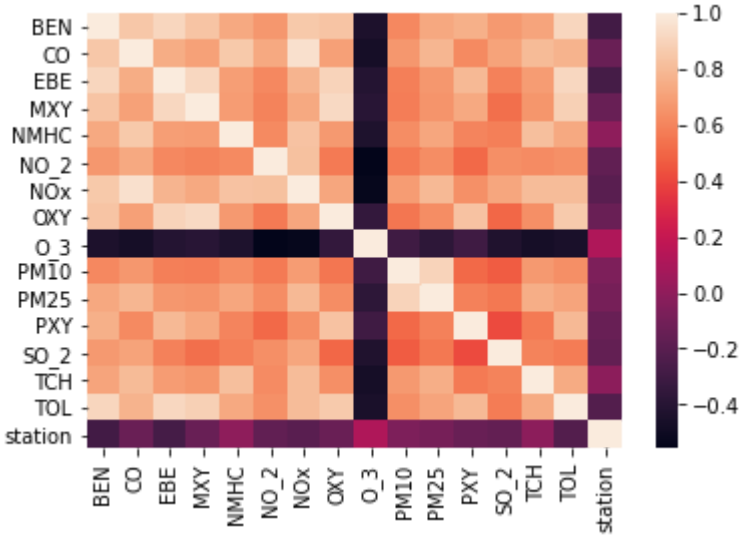
	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	PM10
5	2005-11-01 01:00:00	1.92	0.88	2.44	5.14	0.22	90.309998	207.699997	2.78	13.760000	18.07
22	2005-11-01 01:00:00	0.30	0.22	0.25	0.59	0.11	18.540001	19.020000	0.67	46.799999	9.88
25	2005-11-01 01:00:00	0.67	0.49	0.94	3.44	0.17	48.740002	74.349998	1.57	23.430000	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)
```

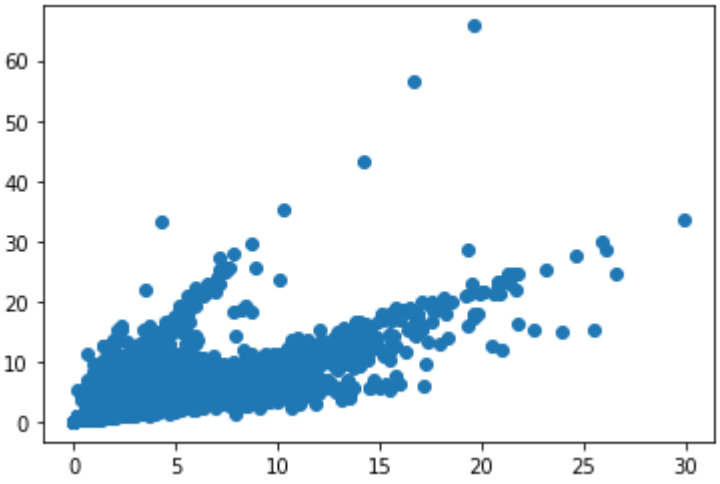
In [6]:
```
sns.heatmap(df1.corr())
```

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)
```

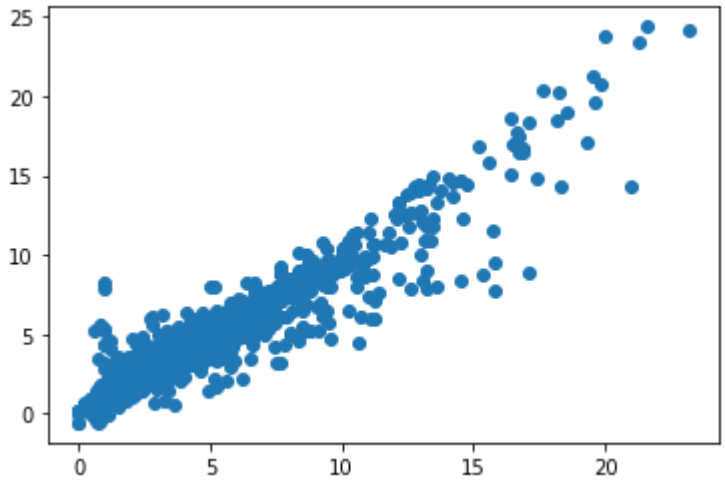
Linear

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
```

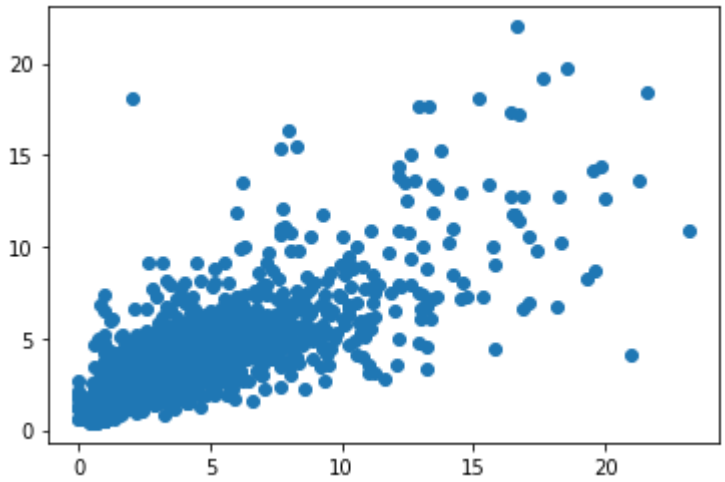
Lasso

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)
```

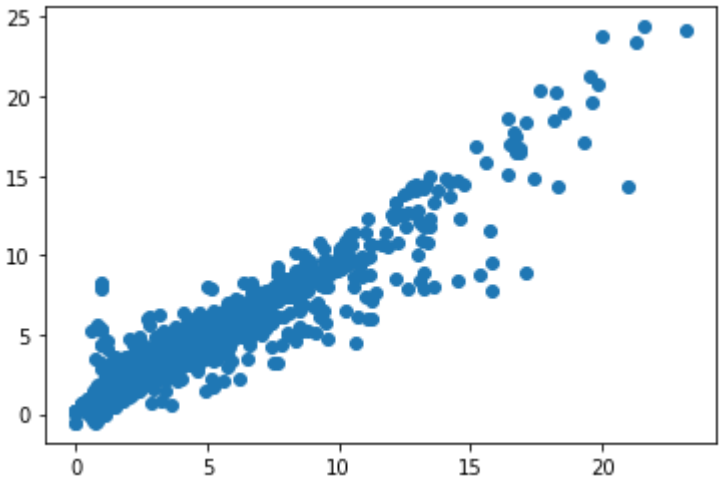
Ridge

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)
```

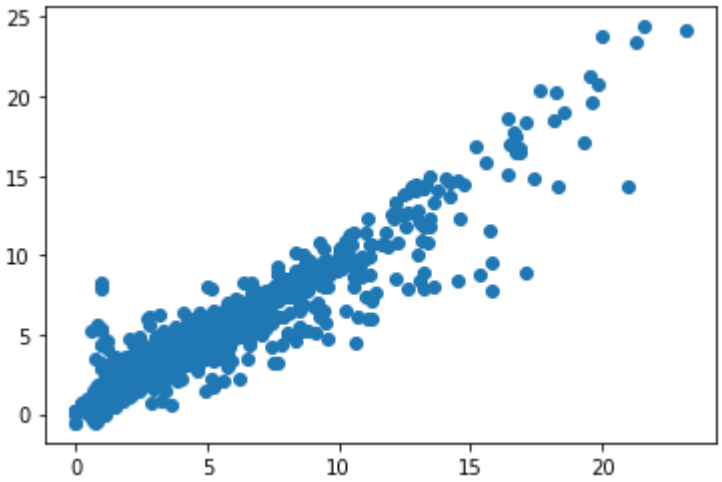
ElasticNet

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_
```

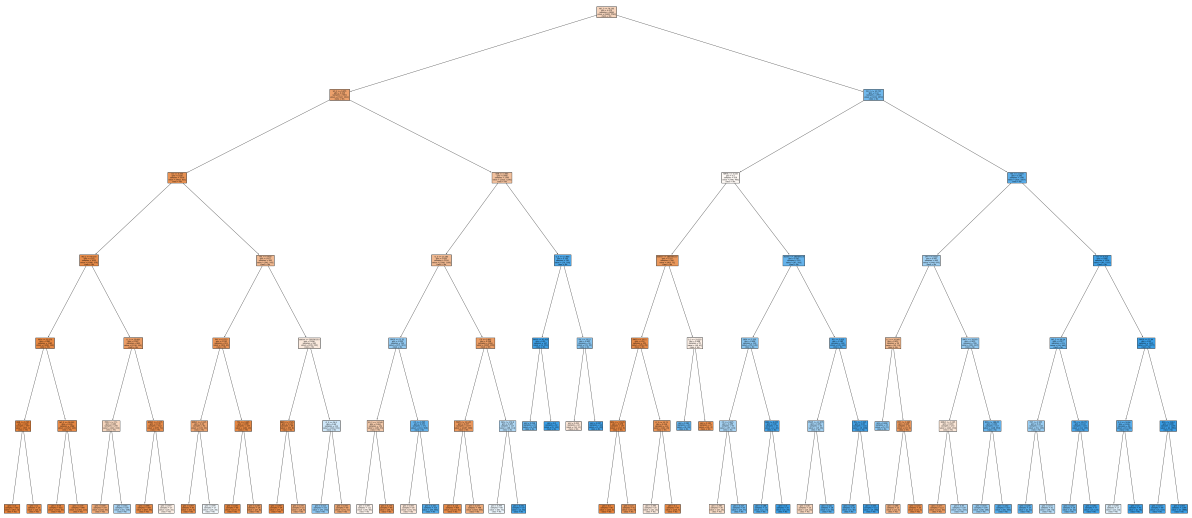
In [39]:
```
from sklearn.tree import plot_tree

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\nsamples = 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\nsamples = 3536\nvalue = [4922, 621]\nclass = Yes'),
 Text(330.6666666666667, 1087.2, 'NO_2 <= 45.175\ngini = 0.147\nsamples = 3053\nvalue = [4380, 379]\nclass = Yes'),
 Text(165.33333333333334, 776.5714285714287, 'NOx <= 22.075\ngini = 0.081\nsamples = 2185\nvalue = [3260, 144]\nclass = Yes'),
 Text(82.66666666666667, 465.9428571428573, 'BEN <= 0.765\ngini = 0.002\nsamples = 597\nvalue = [935, 1]\nclass = Yes'),
 Text(41.333333333333336, 155.3142857142857, 'gini = 0.0\nsamples = 582\nvalue = [917, 0]\nclass = Yes'),
 Text(124.0, 155.3142857142857, 'gini = 0.1\nsamples = 15\nvalue = [18, 1]\nclass = 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\nvalue = [933, 30]\nclass = Yes'),
 Text(289.33333333333337, 155.3142857142857, 'gini = 0.139\nsamples = 966\nvalue = [1392, 113]\nclass = Yes'),
 Text(496.0, 776.5714285714287, 'O_3 <= 25.665\ngini = 0.287\nsamples = 868\nvalue = [1120, 235]\nclass = Yes'),
 Text(413.33333333333337, 465.9428571428573, 'PM10 <= 27.1\ngini = 0.481\nsamples = 256\nvalue = [244, 164]\nclass = Yes'),
 Text(372.0, 155.3142857142857, 'gini = 0.359\nsamples = 147\nvalue = [183, 56]\nclass = Yes'),
 Text(454.6666666666667, 155.3142857142857, 'gini = 0.461\nsamples = 109\nvalue = [61, 108]\nclass = No'),
 Text(578.6666666666667, 465.9428571428573, 'NMHC <= 0.175\ngini = 0.139\nsamples = 612\nvalue = [876, 71]\nclass = Yes'),
 Text(537.3333333333334, 155.3142857142857, 'gini = 0.098\nsamples = 585\nvalue = [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 = [542, 242]\nclass = Yes'),
 Text(826.6666666666667, 776.5714285714287, 'PXY <= 0.715\ngini = 0.218\nsamples = 201\nvalue = [303, 43]\nclass = Yes'),
 Text(744.0, 465.9428571428573, 'NMHC <= 0.135\ngini = 0.34\nsamples = 92\nvalue = [119, 33]\nclass = Yes'),
 Text(702.6666666666667, 155.3142857142857, 'gini = 0.097\nsamples = 65\nvalue = [93, 5]\nclass = Yes'),
 Text(785.3333333333334, 155.3142857142857, 'gini = 0.499\nsamples = 27\nvalue = [26, 28]\nclass = No'),
 Text(909.3333333333334, 465.9428571428573, 'TOL <= 3.485\ngini = 0.098\nsamples = 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\nvalue = [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\nsamples = 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]:

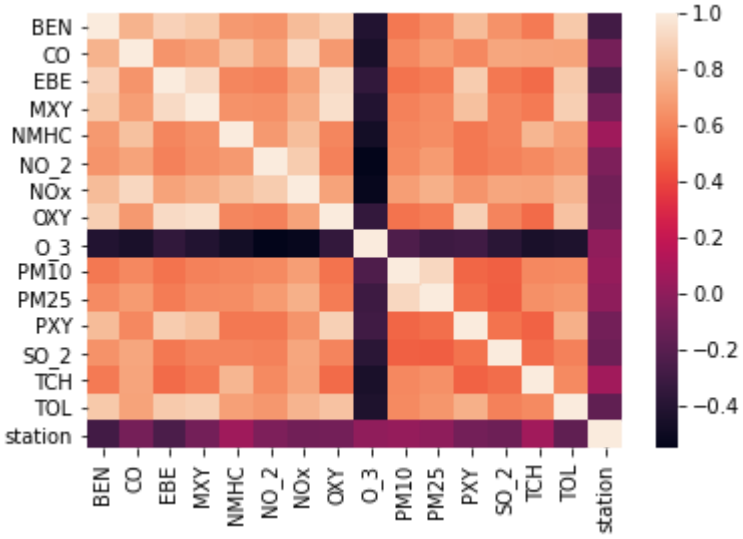
	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3
5	2006-02-01 01:00:00	9.41	1.69	9.98	19.959999	0.44	142.199997	453.500000	11.31	5.990000
22	2006-02-01 01:00:00	1.69	0.79	1.24	2.670000	0.17	59.910000	120.199997	1.11	2.450000
25	2006-02-01 01:00:00	2.35	1.47	2.64	9.660000	0.40	117.699997	346.399994	5.15	4.780000
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

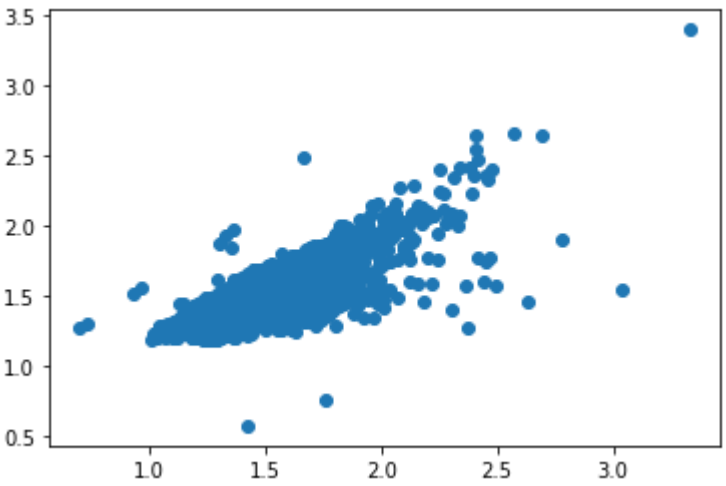
In [47]:
```
li=LinearRegression()
li.fit(x_train,y_train)
```

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
```

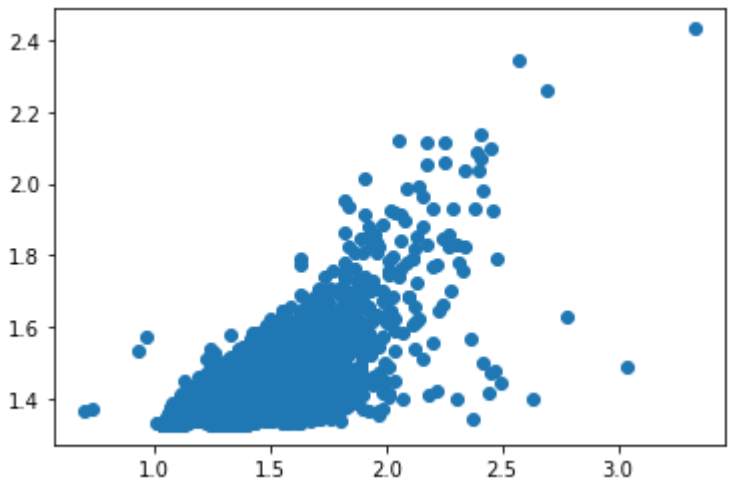
Lasso

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)
```

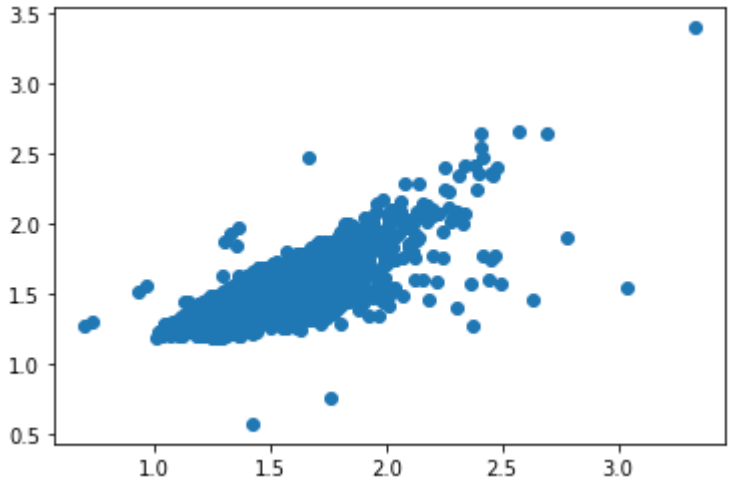
Ridge

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)
```

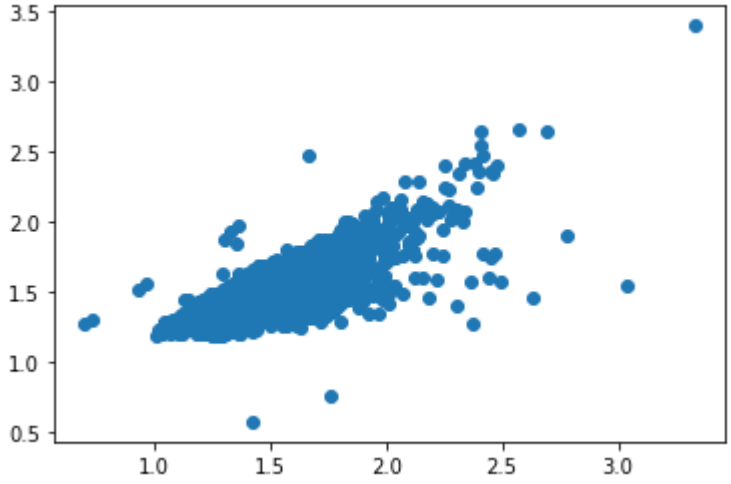
ElasticNet

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',"
```

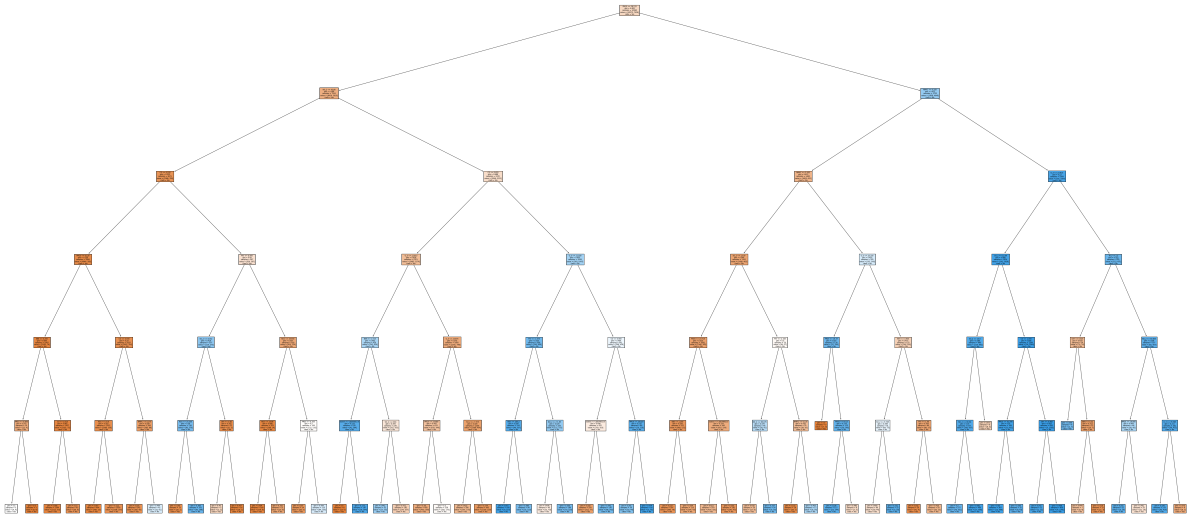
Out[75]:
```
[Text(2359.474137931035, 2019.0857142857144, 'PM10 <= 38.115\ngini = 0.483\ns
amples = 10996\nvalue = [10272, 7058]\nclass = Yes'),
 Text(1231.448275862069, 1708.457142857143, 'NO_2 <= 43.88\ngini = 0.388\nsam
ples = 7241\nvalue = [8414, 3010]\nclass = Yes'),
 Text(615.7241379310345, 1397.8285714285716, '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]:
```
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

2007

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_2007.csv")
df
```

Out[2]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	
0	2007-12-01 01:00:00	NaN	2.86	NaN	NaN	NaN	282.200012	1054.000000	NaN	4.030000	156.1
1	2007-12-01 01:00:00	NaN	1.82	NaN	NaN	NaN	86.419998	354.600006	NaN	3.260000	80.8
2	2007-12-01 01:00:00	NaN	1.47	NaN	NaN	NaN	94.639999	319.000000	NaN	5.310000	53.0
3	2007-12-01 01:00:00	NaN	1.64	NaN	NaN	NaN	127.900002	476.700012	NaN	4.500000	105.3
4	2007-12-01 01:00:00	4.64	1.86	4.26	7.98	0.57	145.100006	573.900024	3.49	52.689999	106.5
...	...	...	...	...	...	...	...	...	...	...	
225115	2007-03-01 00:00:00	0.30	0.45	1.00	0.30	0.26	8.690000	11.690000	1.00	42.209999	6.7
225116	2007-03-01 00:00:00	NaN	0.16	NaN	NaN	NaN	46.820000	51.480000	NaN	22.150000	5.7
225117	2007-03-01 00:00:00	0.24	NaN	0.20	NaN	0.09	51.259998	66.809998	NaN	18.540001	13.0
225118	2007-03-01 00:00:00	0.11	NaN	1.00	NaN	0.05	24.240000	36.930000	NaN	NaN	6.6
225119	2007-03-01 00:00:00	0.53	0.40	1.00	1.70	0.12	32.360001	47.860001	1.37	24.150000	10.2

225120 rows × 17 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 225120 entries, 0 to 225119
Data columns (total 17 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     225120 non-null  object 
 1   BEN      68885 non-null   float64
 2   CO       206748 non-null  float64
 3   EBE      68883 non-null   float64
 4   MXY      26061 non-null   float64
 5   NMHC     86883 non-null   float64
 6   NO_2     223985 non-null  float64
 7   NOx      223972 non-null  float64
 8   OXY      26062 non-null   float64
 9   O_3      211850 non-null  float64
 10  PM10     222588 non-null  float64
 11  PM25     68870 non-null   float64
 12  PXY      26062 non-null   float64
 13  SO_2     224372 non-null  float64
 14  TCH      87026 non-null   float64
 15  TOL      68845 non-null   float64
 16  station  225120 non-null  int64  
dtypes: float64(15), int64(1), object(1)
memory usage: 29.2+ MB
```

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

Out[4]:

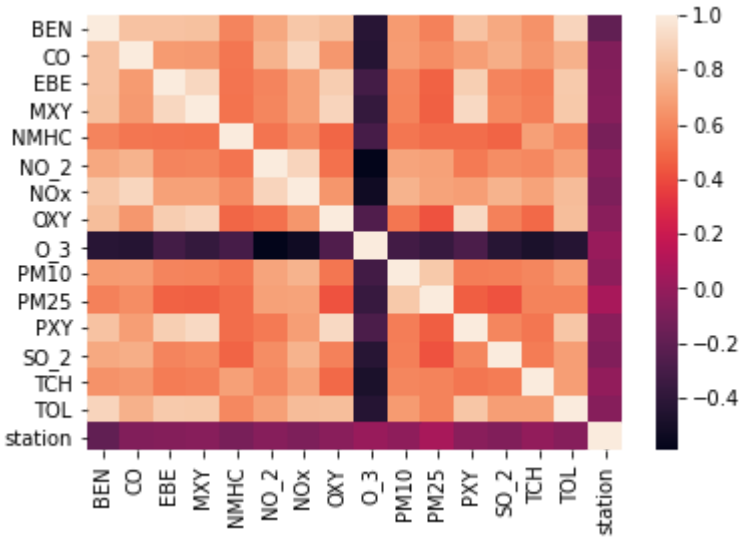
	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	
4	2007-12-01 01:00:00	4.64	1.86	4.26	7.98	0.57	145.100006	573.900024	3.49	52.689999	106.5(
21	2007-12-01 01:00:00	1.98	0.31	2.56	6.06	0.35	76.059998	208.899994	1.70	1.000000	37.79
25	2007-12-01 01:00:00	2.82	1.42	3.15	7.02	0.49	123.099998	402.399994	2.60	7.160000	70.8(
30	2007-12-01 02:00:00	4.65	1.89	4.41	8.21	0.65	151.000000	622.700012	3.55	58.080002	117.09
47	2007-12-01 02:00:00	1.97	0.30	2.15	5.08	0.33	78.760002	189.800003	1.62	1.000000	34.74
...	...	...	...	...	...	...	...	...	...	...	
225073	2007-02-28 23:00:00	2.12	0.47	2.51	4.99	0.05	43.560001	83.889999	2.57	13.090000	21.86
225094	2007-02-28 23:00:00	0.87	0.45	1.19	2.66	0.13	40.000000	61.959999	1.79	20.440001	15.07
225098	2007-03-01 00:00:00	0.95	0.41	1.55	3.11	0.05	36.090000	63.349998	1.74	17.160000	9.21
225115	2007-03-01 00:00:00	0.30	0.45	1.00	0.30	0.26	8.690000	11.690000	1.00	42.209999	6.76
225119	2007-03-01 00:00:00	0.53	0.40	1.00	1.70	0.12	32.360001	47.860001	1.37	24.150000	10.26

25443 rows × 17 columns

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

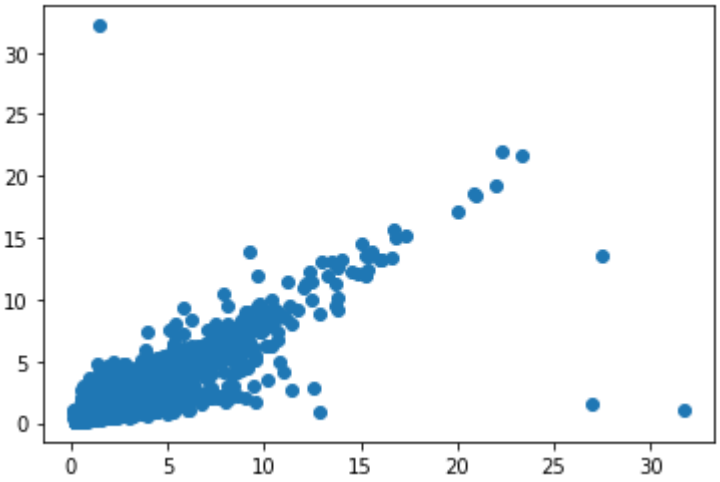
In [6]:
```
sns.heatmap(df1.corr())
```

Out[6]: <AxesSubplot:>



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

Out[7]: [<matplotlib.lines.Line2D at 0x2118cb47160>]



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)
```

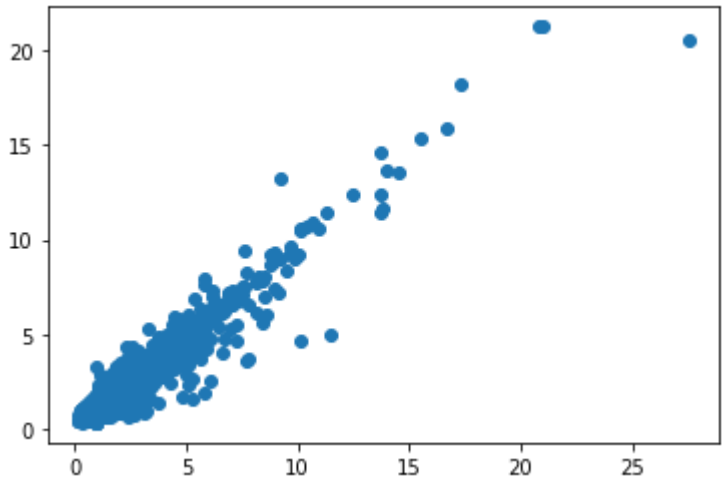
Linear

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 0x2118d9e5bb0>
```



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

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

Out[14]:
```
1.34    1130
1.33    1067
1.35    1037
1.36    1002
1.32     991
        ...
4.07       1
2.71       1
0.40       1
0.38       1
3.32       1
Name: TCH, Length: 250, 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    14025
2.0    11418
Name: TCH, dtype: int64
```

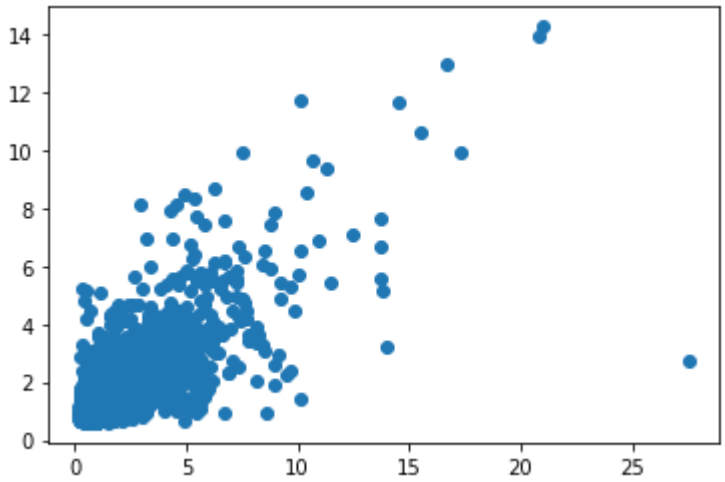
Lasso

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 0x2118da53880>
```



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

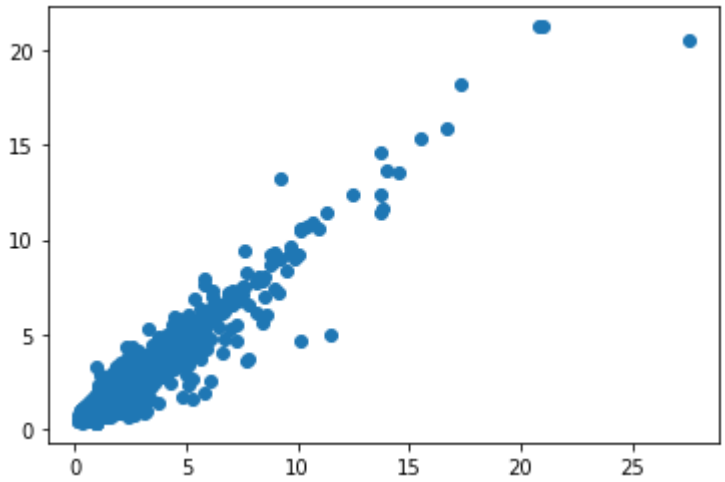
Ridge

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 0x2118cb1d6a0>
```



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

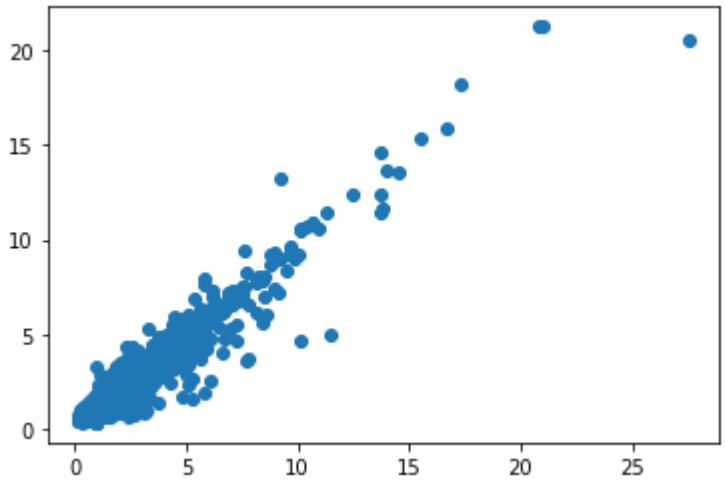
ElasticNet

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 0x2118dadef40>
```



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

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

```
0.9130065705546464
```

Out[25]:
```
0.8598350807252741
```

Logistic

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

Out[26]:
```
Low     14025
High    11418
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 0x2118db28d00>
```

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',"
```

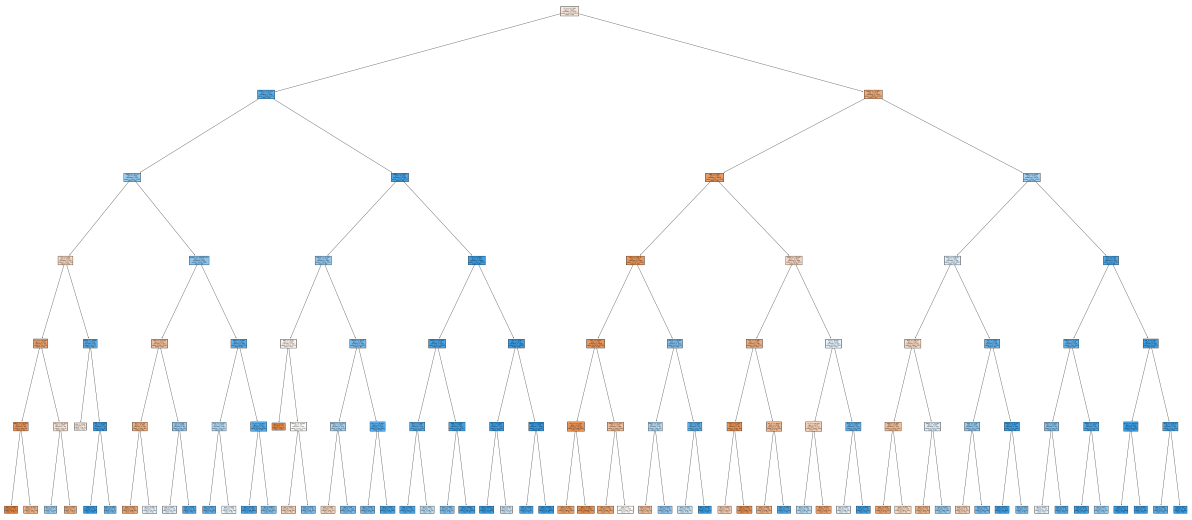
Out[39]:
```
[Text(2134.3500000000004, 2019.0857142857144, 'O_3 <= 17.425\ngini = 0.495\ns
amples = 11234\nvalue = [9753, 8057]\nclass = Yes'),
 Text(995.1, 1708.457142857143, 'PM10 <= 25.535\ngini = 0.251\nsamples = 3760
\nvalue = [888, 5137]\nclass = No'),
 Text(492.90000000000003, 1397.8285714285716, 'PM10 <= 11.12\ngini = 0.438\ns
amples = 974\nvalue = [509, 1060]\nclass = No'),
 Text(241.8, 1087.2, 'CO <= 0.45\ngini = 0.483\nsamples = 103\nvalue = [105,
72]\nclass = Yes'),
 Text(148.8, 776.5714285714287, 'O_3 <= 14.835\ngini = 0.326\nsamples = 74\nv
alue = [97, 25]\nclass = Yes'),
 Text(74.4, 465.9428571428573, 'PM10 <= 10.265\ngini = 0.265\nsamples = 60\nv
alue = [86, 16]\nclass = Yes'),
 Text(37.2, 155.3142857142857, 'gini = 0.098\nsamples = 33\nvalue = [55, 3]\n
class = Yes'),
 Text(111.60000000000001, 155.3142857142857, 'gini = 0.416\nsamples = 27\nval
ue = [31, 13]\nclass = Yes'),
 Text(223.20000000000002, 465.9428571428573, 'NOx <= 60.305\ngini = 0.495\nsa
mples = 14\nvalue = [11, 9]\nclass = Yes'),
 Text(186.0, 155.3142857142857, 'gini = 0.444\nsamples = 7\nvalue = [3, 6]\nc
lass = No'),
 Text(260.40000000000003, 155.3142857142857, 'gini = 0.397\nsamples = 7\nvalu
e = [8, 3]\nclass = Yes'),
 Text(334.8, 776.5714285714287, 'PM25 <= 4.895\ngini = 0.249\nsamples = 29\nv
alue = [8, 47]\nclass = No'),
 Text(297.6, 465.9428571428573, 'gini = 0.496\nsamples = 6\nvalue = [6, 5]\nc
lass = Yes'),
 Text(372.0, 465.9428571428573, 'TOL <= 5.025\ngini = 0.087\nsamples = 23\nva
lue = [2, 42]\nclass = No'),
 Text(334.8, 155.3142857142857, 'gini = 0.0\nsamples = 18\nvalue = [0, 35]\nc
lass = No'),
 Text(409.20000000000005, 155.3142857142857, 'gini = 0.346\nsamples = 5\nvalu
e = [2, 7]\nclass = No'),
 Text(744.0, 1087.2, 'station <= 28079015.0\ngini = 0.412\nsamples = 871\nval
ue = [404, 988]\nclass = No'),
 Text(595.2, 776.5714285714287, 'NOx <= 175.55\ngini = 0.437\nsamples = 227\n
value = [244, 116]\nclass = Yes'),
 Text(520.8000000000001, 465.9428571428573, 'TOL <= 8.14\ngini = 0.398\nsampl
es = 197\nvalue = [230, 87]\nclass = Yes'),
 Text(483.6, 155.3142857142857, 'gini = 0.354\nsamples = 166\nvalue = [208, 6
2]\nclass = Yes'),
 Text(558.0, 155.3142857142857, 'gini = 0.498\nsamples = 31\nvalue = [22, 25]
\nclass = No'),
 Text(669.6, 465.9428571428573, 'CO <= 0.82\ngini = 0.439\nsamples = 30\nvalu
e = [14, 29]\nclass = No'),
 Text(632.4000000000001, 155.3142857142857, 'gini = 0.497\nsamples = 18\nvalu
e = [12, 14]\nclass = No'),
 Text(706.8000000000001, 155.3142857142857, 'gini = 0.208\nsamples = 12\nvalu
e = [2, 15]\nclass = No'),
 Text(892.8000000000001, 776.5714285714287, 'BEN <= 0.455\ngini = 0.262\nsamp
les = 644\nvalue = [160, 872]\nclass = No'),
 Text(818.4000000000001, 465.9428571428573, 'PXY <= 0.625\ngini = 0.471\nsamp
les = 65\nvalue = [41, 67]\nclass = No'),
 Text(781.2, 155.3142857142857, 'gini = 0.291\nsamples = 21\nvalue = [6, 28]
\nclass = No'),
 Text(855.6, 155.3142857142857, 'gini = 0.499\nsamples = 44\nvalue = [35, 39]
\nclass = No'),
 Text(967.2, 465.9428571428573, 'O_3 <= 10.815\ngini = 0.224\nsamples = 579\n
```

```
value = [119, 805]\nclass = No'),
 Text(930.0000000000001, 155.3142857142857, 'gini = 0.139\nsamples = 405\nval
ue = [49, 604]\nclass = No'),
 Text(1004.4000000000001, 155.3142857142857, 'gini = 0.383\nsamples = 174\nva
lue = [70, 201]\nclass = No'),
 Text(1497.3000000000002, 1397.8285714285716, 'NMHC <= 0.225\ngini = 0.156\ns
amples = 2786\nvalue = [379, 4077]\nclass = No'),
 Text(1209.0, 1087.2, 'NMHC <= 0.155\ngini = 0.436\nsamples = 500\nvalue = [2
56, 542]\nclass = No'),
 Text(1078.8000000000002, 776.5714285714287, 'MXY <= 0.63\ngini = 0.498\nsamp
les = 152\nvalue = [124, 111]\nclass = Yes'),
 Text(1041.6000000000001, 465.9428571428573, 'gini = 0.133\nsamples = 9\nvalu
e = [13, 1]\nclass = Yes'),
 Text(1116.0, 465.9428571428573, 'PM10 <= 44.565\ngini = 0.5\nsamples = 143\n
value = [111, 110]\nclass = Yes'),
 Text(1078.8000000000002, 155.3142857142857, 'gini = 0.483\nsamples = 107\nva
lue = [96, 66]\nclass = Yes'),
 Text(1153.2, 155.3142857142857, 'gini = 0.379\nsamples = 36\nvalue = [15, 4
4]\nclass = No'),
 Text(1339.2, 776.5714285714287, 'PM10 <= 40.12\ngini = 0.359\nsamples = 348
\nvalue = [132, 431]\nclass = No'),
 Text(1264.8000000000002, 465.9428571428573, 'PM25 <= 14.615\ngini = 0.461\ns
amples = 173\nvalue = [96, 170]\nclass = No'),
 Text(1227.6000000000001, 155.3142857142857, 'gini = 0.481\nsamples = 55\nval
ue = [55, 37]\nclass = Yes'),
 Text(1302.0, 155.3142857142857, 'gini = 0.36\nsamples = 118\nvalue = [41, 13
3]\nclass = No'),
 Text(1413.6000000000001, 465.9428571428573, 'SO_2 <= 9.425\ngini = 0.213\nsa
mples = 175\nvalue = [36, 261]\nclass = No'),
 Text(1376.4, 155.3142857142857, 'gini = 0.348\nsamples = 52\nvalue = [20, 6
9]\nclass = No'),
 Text(1450.8000000000002, 155.3142857142857, 'gini = 0.142\nsamples = 123\nva
lue = [16, 192]\nclass = No'),
 Text(1785.6000000000001, 1087.2, 'CO <= 0.815\ngini = 0.065\nsamples = 2286
\nvalue = [123, 3535]\nclass = No'),
 Text(1636.8000000000002, 776.5714285714287, 'PM25 <= 15.605\ngini = 0.119\ns
amples = 1103\nvalue = [112, 1656]\nclass = No'),
 Text(1562.4, 465.9428571428573, 'OXY <= 2.695\ngini = 0.235\nsamples = 176\n
value = [37, 235]\nclass = No'),
 Text(1525.2, 155.3142857142857, 'gini = 0.165\nsamples = 150\nvalue = [21, 2
11]\nclass = No'),
 Text(1599.6000000000001, 155.3142857142857, 'gini = 0.48\nsamples = 26\nvalu
e = [16, 24]\nclass = No'),
 Text(1711.2, 465.9428571428573, 'EBE <= 0.635\ngini = 0.095\nsamples = 927\n
value = [75, 1421]\nclass = No'),
 Text(1674.0000000000002, 155.3142857142857, 'gini = 0.394\nsamples = 28\nval
ue = [10, 27]\nclass = No'),
 Text(1748.4, 155.3142857142857, 'gini = 0.085\nsamples = 899\nvalue = [65, 1
394]\nclass = No'),
 Text(1934.4, 776.5714285714287, 'TOL <= 6.18\ngini = 0.012\nsamples = 1183\n
value = [11, 1879]\nclass = No'),
 Text(1860.0000000000002, 465.9428571428573, 'TOL <= 6.025\ngini = 0.055\nsam
ples = 147\nvalue = [6, 208]\nclass = No'),
 Text(1822.8000000000002, 155.3142857142857, 'gini = 0.029\nsamples = 142\nva
lue = [3, 204]\nclass = No'),
 Text(1897.2, 155.3142857142857, 'gini = 0.49\nsamples = 5\nvalue = [3, 4]\nc
lass = No'),
```

```
 Text(2008.8000000000002, 465.9428571428573, 'PM10 <= 29.675\ngini = 0.006\ns
amples = 1036\nvalue = [5, 1671]\nclass = No'),
 Text(1971.6000000000001, 155.3142857142857, 'gini = 0.159\nsamples = 15\nval
ue = [2, 21]\nclass = No'),
 Text(2046.0000000000002, 155.3142857142857, 'gini = 0.004\nsamples = 1021\nv
alue = [3, 1650]\nclass = No'),
 Text(3273.6000000000004, 1708.457142857143, 'NMHC <= 0.285\ngini = 0.373\nsa
mples = 7474\nvalue = [8865, 2920]\nclass = Yes'),
 Text(2678.4, 1397.8285714285716, 'EBE <= 1.625\ngini = 0.29\nsamples = 6348
\nvalue = [8257, 1759]\nclass = Yes'),
 Text(2380.8, 1087.2, 'NOx <= 181.8\ngini = 0.255\nsamples = 5740\nvalue = [7
677, 1354]\nclass = Yes'),
 Text(2232.0, 776.5714285714287, 'TOL <= 4.525\ngini = 0.24\nsamples = 5638\n
value = [7638, 1240]\nclass = Yes'),
 Text(2157.6000000000004, 465.9428571428573, 'SO_2 <= 5.125\ngini = 0.214\nsa
mples = 5152\nvalue = [7129, 991]\nclass = Yes'),
 Text(2120.4, 155.3142857142857, 'gini = 0.324\nsamples = 688\nvalue = [863,
220]\nclass = Yes'),
 Text(2194.8, 155.3142857142857, 'gini = 0.195\nsamples = 4464\nvalue = [626
6, 771]\nclass = Yes'),
 Text(2306.4, 465.9428571428573, 'NMHC <= 0.225\ngini = 0.441\nsamples = 486
\nvalue = [509, 249]\nclass = Yes'),
 Text(2269.2000000000003, 155.3142857142857, 'gini = 0.351\nsamples = 301\nva
lue = [364, 107]\nclass = Yes'),
 Text(2343.6000000000004, 155.3142857142857, 'gini = 0.5\nsamples = 185\nvalu
e = [145, 142]\nclass = Yes'),
 Text(2529.6000000000004, 776.5714285714287, 'PM10 <= 38.89\ngini = 0.38\nsam
ples = 102\nvalue = [39, 114]\nclass = No'),
 Text(2455.2000000000003, 465.9428571428573, 'PM25 <= 10.23\ngini = 0.472\nsa
mples = 49\nvalue = [26, 42]\nclass = No'),
 Text(2418.0, 155.3142857142857, 'gini = 0.444\nsamples = 9\nvalue = [10, 5]
\nclass = Yes'),
 Text(2492.4, 155.3142857142857, 'gini = 0.422\nsamples = 40\nvalue = [16, 3
7]\nclass = No'),
 Text(2604.0, 465.9428571428573, 'CO <= 0.705\ngini = 0.259\nsamples = 53\nva
lue = [13, 72]\nclass = No'),
 Text(2566.8, 155.3142857142857, 'gini = 0.485\nsamples = 16\nvalue = [12, 1
7]\nclass = No'),
 Text(2641.2000000000003, 155.3142857142857, 'gini = 0.035\nsamples = 37\nval
ue = [1, 55]\nclass = No'),
 Text(2976.0, 1087.2, 'PM25 <= 15.645\ngini = 0.484\nsamples = 608\nvalue =
[580, 405]\nclass = Yes'),
 Text(2827.2000000000003, 776.5714285714287, 'PXY <= 1.285\ngini = 0.382\nsam
ples = 281\nvalue = [326, 113]\nclass = Yes'),
 Text(2752.8, 465.9428571428573, 'PXY <= 0.755\ngini = 0.233\nsamples = 92\nv
alue = [116, 18]\nclass = Yes'),
 Text(2715.6000000000004, 155.3142857142857, 'gini = 0.469\nsamples = 13\nval
ue = [10, 6]\nclass = Yes'),
 Text(2790.0, 155.3142857142857, 'gini = 0.183\nsamples = 79\nvalue = [106, 1
2]\nclass = Yes'),
 Text(2901.6000000000004, 465.9428571428573, 'NO_2 <= 98.25\ngini = 0.429\nsa
mples = 189\nvalue = [210, 95]\nclass = Yes'),
 Text(2864.4, 155.3142857142857, 'gini = 0.399\nsamples = 181\nvalue = [208,
79]\nclass = Yes'),
 Text(2938.8, 155.3142857142857, 'gini = 0.198\nsamples = 8\nvalue = [2, 16]
\nclass = No'),
 Text(3124.8, 776.5714285714287, 'CO <= 0.675\ngini = 0.498\nsamples = 327\nv
```

```
alue = [254, 292]\nclass = No'),
 Text(3050.4, 465.9428571428573, 'O_3 <= 34.815\ngini = 0.477\nsamples = 213
\nvalue = [220, 142]\nclass = Yes'),
 Text(3013.2000000000003, 155.3142857142857, 'gini = 0.476\nsamples = 96\nval
ue = [65, 101]\nclass = No'),
 Text(3087.6000000000004, 155.3142857142857, 'gini = 0.331\nsamples = 117\nva
lue = [155, 41]\nclass = Yes'),
 Text(3199.2000000000003, 465.9428571428573, 'NOx <= 174.65\ngini = 0.301\nsa
mples = 114\nvalue = [34, 150]\nclass = No'),
 Text(3162.0000000000005, 155.3142857142857, 'gini = 0.497\nsamples = 41\nval
ue = [28, 33]\nclass = No'),
 Text(3236.4, 155.3142857142857, 'gini = 0.093\nsamples = 73\nvalue = [6, 11
7]\nclass = No'),
 Text(3868.8, 1397.8285714285716, 'NMHC <= 0.455\ngini = 0.451\nsamples = 112
6\nvalue = [608, 1161]\nclass = No'),
 Text(3571.2000000000003, 1087.2, 'NOx <= 118.95\ngini = 0.495\nsamples = 795
\nvalue = [563, 685]\nclass = No'),
 Text(3422.4, 776.5714285714287, 'NOx <= 48.73\ngini = 0.484\nsamples = 533\n
value = [490, 342]\nclass = Yes'),
 Text(3348.0000000000005, 465.9428571428573, 'BEN <= 0.365\ngini = 0.439\nsam
ples = 317\nvalue = [337, 162]\nclass = Yes'),
 Text(3310.8, 155.3142857142857, 'gini = 0.339\nsamples = 138\nvalue = [170,
47]\nclass = Yes'),
 Text(3385.2000000000003, 155.3142857142857, 'gini = 0.483\nsamples = 179\nva
lue = [167, 115]\nclass = Yes'),
 Text(3496.8, 465.9428571428573, 'PM25 <= 12.065\ngini = 0.497\nsamples = 216
\nvalue = [153, 180]\nclass = No'),
 Text(3459.6000000000004, 155.3142857142857, 'gini = 0.413\nsamples = 76\nval
ue = [85, 35]\nclass = Yes'),
 Text(3534.0000000000005, 155.3142857142857, 'gini = 0.435\nsamples = 140\nva
lue = [68, 145]\nclass = No'),
 Text(3720.0000000000005, 776.5714285714287, 'CO <= 0.655\ngini = 0.289\nsamp
les = 262\nvalue = [73, 343]\nclass = No'),
 Text(3645.6000000000004, 465.9428571428573, 'MXY <= 1.86\ngini = 0.431\nsamp
les = 120\nvalue = [62, 135]\nclass = No'),
 Text(3608.4, 155.3142857142857, 'gini = 0.474\nsamples = 26\nvalue = [27, 1
7]\nclass = Yes'),
 Text(3682.8, 155.3142857142857, 'gini = 0.353\nsamples = 94\nvalue = [35, 11
8]\nclass = No'),
 Text(3794.4, 465.9428571428573, 'PM25 <= 37.79\ngini = 0.095\nsamples = 142
\nvalue = [11, 208]\nclass = No'),
 Text(3757.2000000000003, 155.3142857142857, 'gini = 0.074\nsamples = 134\nva
lue = [8, 199]\nclass = No'),
 Text(3831.6000000000004, 155.3142857142857, 'gini = 0.375\nsamples = 8\nvalu
e = [3, 9]\nclass = No'),
 Text(4166.400000000001, 1087.2, 'TOL <= 2.01\ngini = 0.158\nsamples = 331\nv
alue = [45, 476]\nclass = No'),
 Text(4017.6000000000004, 776.5714285714287, 'NMHC <= 0.475\ngini = 0.268\nsa
mples = 129\nvalue = [32, 169]\nclass = No'),
 Text(3943.2000000000003, 465.9428571428573, 'PXY <= 0.625\ngini = 0.404\nsam
ples = 44\nvalue = [18, 46]\nclass = No'),
 Text(3906.0000000000005, 155.3142857142857, 'gini = 0.497\nsamples = 19\nval
ue = [13, 15]\nclass = No'),
 Text(3980.4, 155.3142857142857, 'gini = 0.239\nsamples = 25\nvalue = [5, 31]
\nclass = No'),
 Text(4092.0000000000005, 465.9428571428573, 'PM25 <= 12.15\ngini = 0.183\nsa
mples = 85\nvalue = [14, 123]\nclass = No'),
```

```
 Text(4054.8, 155.3142857142857, 'gini = 0.073\nsamples = 64\nvalue = [4, 10
1]\nclass = No'),
 Text(4129.200000000001, 155.3142857142857, 'gini = 0.43\nsamples = 21\nvalue
= [10, 22]\nclass = No'),
 Text(4315.200000000001, 776.5714285714287, 'SO_2 <= 9.17\ngini = 0.078\nsamp
les = 202\nvalue = [13, 307]\nclass = No'),
 Text(4240.8, 465.9428571428573, 'NO_2 <= 54.075\ngini = 0.024\nsamples = 98
\nvalue = [2, 164]\nclass = No'),
 Text(4203.6, 155.3142857142857, 'gini = 0.0\nsamples = 61\nvalue = [0, 106]
\nclass = No'),
 Text(4278.0, 155.3142857142857, 'gini = 0.064\nsamples = 37\nvalue = [2, 58]
\nclass = No'),
 Text(4389.6, 465.9428571428573, 'NOx <= 156.35\ngini = 0.133\nsamples = 104
\nvalue = [11, 143]\nclass = No'),
 Text(4352.400000000001, 155.3142857142857, 'gini = 0.265\nsamples = 48\nvalu
e = [11, 59]\nclass = No'),
 Text(4426.8, 155.3142857142857, 'gini = 0.0\nsamples = 56\nvalue = [0, 84]\n
class = No')]
```



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

```
Linear: 0.9130054463649523
Lasso: 0.5416068633176341
Ridge: 0.9130065705546464
ElasticNet: 0.8500784330712808
Logistic: 0.5553517620856806
Random Forest: 0.8690061763054464
```

Best Model is Ridge Regression

2008

In [41]:
```
df2=pd.read_csv("madrid_2008.csv")
df2
```

Out[41]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	P
0	2008-06-01 01:00:00	NaN	0.47	NaN	NaN	NaN	83.089996	120.699997	NaN	16.990000	16.88
1	2008-06-01 01:00:00	NaN	0.59	NaN	NaN	NaN	94.820000	130.399994	NaN	17.469999	19.04
2	2008-06-01 01:00:00	NaN	0.55	NaN	NaN	NaN	75.919998	104.599998	NaN	13.470000	20.27
3	2008-06-01 01:00:00	NaN	0.36	NaN	NaN	NaN	61.029999	66.559998	NaN	23.110001	10.85
4	2008-06-01 01:00:00	1.68	0.80	1.70	3.01	0.30	105.199997	214.899994	1.61	12.120000	37.16
...	...	...	...	...	...	...	...	...	...	...	
226387	2008-11-01 00:00:00	0.48	0.30	0.57	1.00	0.31	13.050000	14.160000	0.91	57.400002	5.45
226388	2008-11-01 00:00:00	NaN	0.30	NaN	NaN	NaN	41.880001	48.500000	NaN	35.830002	15.02
226389	2008-11-01 00:00:00	0.25	NaN	0.56	NaN	0.11	83.610001	102.199997	NaN	14.130000	17.54
226390	2008-11-01 00:00:00	0.54	NaN	2.70	NaN	0.18	70.639999	81.860001	NaN	NaN	11.91
226391	2008-11-01 00:00:00	0.75	0.36	1.20	2.75	0.16	58.240002	74.239998	1.64	31.910000	12.69

226392 rows × 17 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 226392 entries, 0 to 226391
Data columns (total 17 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     226392 non-null  object 
 1   BEN      67047 non-null   float64
 2   CO       208109 non-null  float64
 3   EBE      67044 non-null   float64
 4   MXY      25867 non-null   float64
 5   NMHC     85079 non-null   float64
 6   NO_2     225315 non-null  float64
 7   NOx      225311 non-null  float64
 8   OXY      25878 non-null   float64
 9   O_3      215716 non-null  float64
 10  PM10     220179 non-null  float64
 11  PM25     67833 non-null   float64
 12  PXY      25877 non-null   float64
 13  SO_2     225405 non-null  float64
 14  TCH      85107 non-null   float64
 15  TOL      66940 non-null   float64
 16  station  226392 non-null  int64  
dtypes: float64(15), int64(1), object(1)
memory usage: 29.4+ MB
```

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

Out[43]:

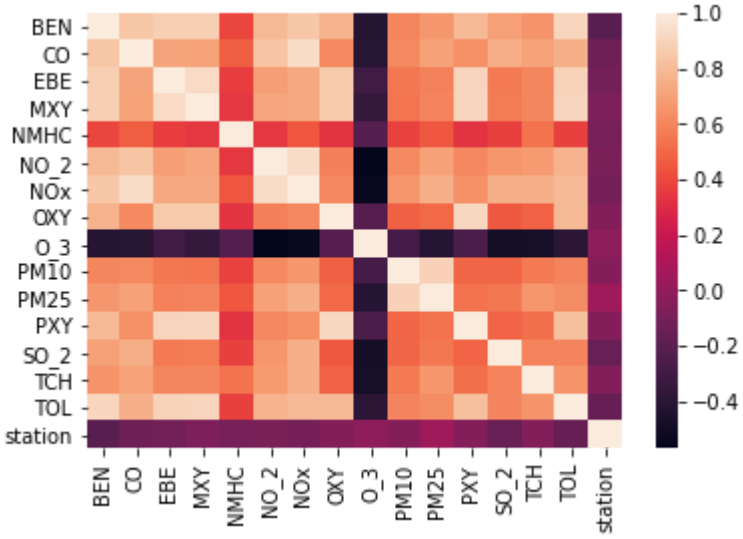
	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	P
4	2008-06-01 01:00:00	1.68	0.80	1.70	3.01	0.30	105.199997	214.899994	1.61	12.120000	37.16(
21	2008-06-01 01:00:00	0.32	0.37	1.00	0.39	0.33	21.580000	22.180000	1.00	35.770000	7.90(
25	2008-06-01 01:00:00	0.73	0.39	1.04	1.70	0.18	64.839996	86.709999	1.31	23.379999	14.76(
30	2008-06-01 02:00:00	1.95	0.51	1.98	3.77	0.24	79.750000	143.399994	2.03	18.090000	31.13(
47	2008-06-01 02:00:00	0.36	0.39	0.39	0.50	0.34	26.790001	27.389999	1.00	33.029999	7.62(
...	...	...	...	...	...	...	...	...	...	...	
226362	2008-10-31 23:00:00	0.47	0.35	0.65	1.00	0.33	22.480000	25.020000	1.00	33.509998	10.20(
226366	2008-10-31 23:00:00	0.92	0.46	1.21	2.75	0.19	78.440002	106.199997	1.70	18.320000	14.14(
226371	2008-11-01 00:00:00	1.83	0.53	2.22	4.51	0.17	93.260002	158.399994	2.38	18.770000	20.75(
226387	2008-11-01 00:00:00	0.48	0.30	0.57	1.00	0.31	13.050000	14.160000	0.91	57.400002	5.45(
226391	2008-11-01 00:00:00	0.75	0.36	1.20	2.75	0.16	58.240002	74.239998	1.64	31.910000	12.69(

25631 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

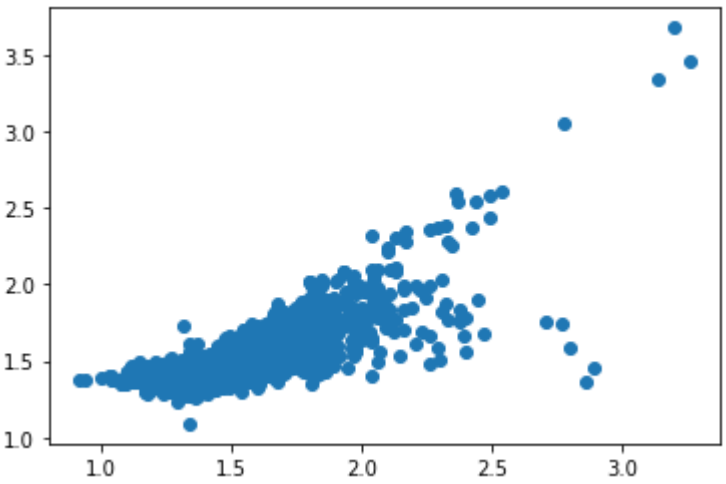
In [47]:
```
li=LinearRegression()
li.fit(x_train,y_train)
```

Out[47]: LinearRegression()

In [ ]:
```

```

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

Out[48]: <matplotlib.collections.PathCollection at 0x2118db65550>



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

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

Out[50]:
```
1.38    1274
1.37    1246
1.36    1243
1.39    1242
1.35    1209
        ...
2.41       1
2.95       1
0.98       1
2.64       1
2.61       1
Name: TCH, Length: 177, 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]:
```
2.0    12904
1.0    12727
Name: TCH, dtype: int64
```

In [ ]:
```

```

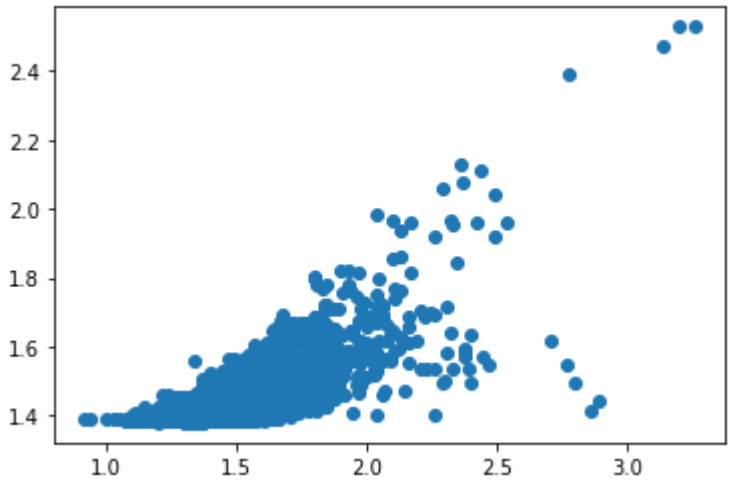
Lasso

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 0x2118dbbc100>



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

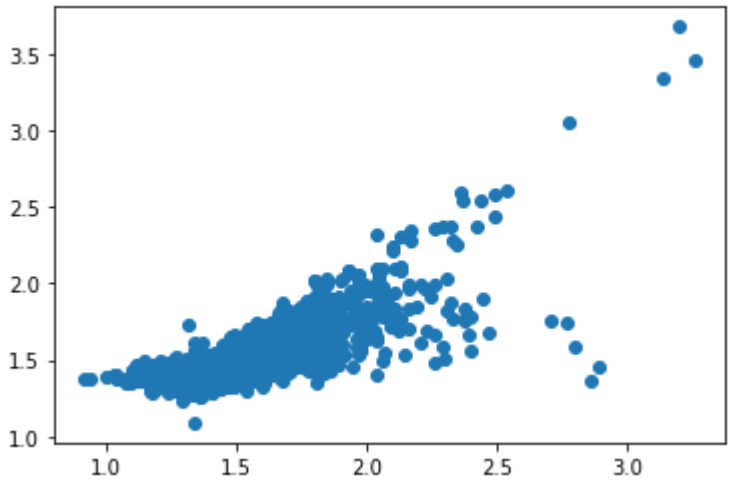
Ridge

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 0x2118dc096d0>



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

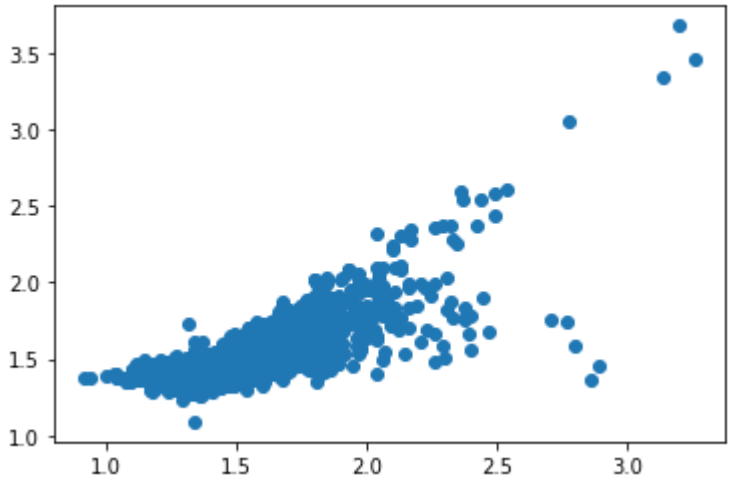
ElasticNet

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 0x2118dc5cca0>
```



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

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

```
0.6614787517671646
```

Out[61]:
```
0.6579137146941019
```

Logistic

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

Out[62]:
```
High    12904
Low     12727
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 0x2118d472880>
```

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(2217.053571428571, 2019.0857142857144, 'NOx <= 95.465\ngini = 0.5\nsa
mples = 11390\nvalue = [8921, 9020]\nclass = No'),
 Text(1101.0535714285713, 1708.457142857143, 'NO_2 <= 45.97\ngini = 0.411\n
samples = 7185\nvalue = [8055, 3272]\nclass = Yes'),
 Text(548.0357142857142, 1397.8285714285716, 'station <= 28079015.0\ngini =
0.364\nsamples = 5401\nvalue = [6447, 2024]\nclass = Yes'),
 Text(298.9285714285714, 1087.2, 'NMHC <= 0.255\ngini = 0.162\nsamples = 73
5\nvalue = [1047, 102]\nclass = Yes'),
 Text(159.42857142857142, 776.5714285714287, 'MXY <= 1.275\ngini = 0.107\ns
amples = 703\nvalue = [1043, 63]\nclass = Yes'),
 Text(79.71428571428571, 465.9428571428573, 'CO <= 0.425\ngini = 0.035\nsam
ples = 432\nvalue = [652, 12]\nclass = Yes'),
 Text(39.857142857142854, 155.3142857142857, 'gini = 0.024\nsamples = 426\n
value = [648, 8]\nclass = Yes'),
 Text(119.57142857142856, 155.3142857142857, 'gini = 0.5\nsamples = 6\nvalu
e = [4, 4]\nclass = Yes'),
 Text(239.1428571428571, 465.9428571428573, 'NO_2 <= 36.62\ngini = 0.204\ns
amples = 271\nvalue = [391, 51]\nclass = Yes'),
 Text(199.28571428571428, 155.3142857142857, 'gini = 0.105\nsamples = 130\n
```

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

```
Linear: 0.6614554468063345
Lasso: 0.4603230758526199
Ridge: 0.6614787517671646
ElasticNet: 0.5801002117823061
Logistic: 0.5009102730819246
Random Forest: 0.8317820819146347
```

Best model is Random Forest

2009 ¶

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_2009.csv")
df
```

Out[2]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	PM
0	2009-10-01 01:00:00	NaN	0.27	NaN	NaN	NaN	39.889999	48.150002	NaN	50.680000	18.260(
1	2009-10-01 01:00:00	NaN	0.22	NaN	NaN	NaN	21.230000	24.260000	NaN	55.880001	10.580(
2	2009-10-01 01:00:00	NaN	0.18	NaN	NaN	NaN	31.230000	34.880001	NaN	49.060001	25.190(
3	2009-10-01 01:00:00	0.95	0.33	1.43	2.68	0.25	55.180000	81.360001	1.57	36.669998	26.530(
4	2009-10-01 01:00:00	NaN	0.41	NaN	NaN	0.12	61.349998	76.260002	NaN	38.090000	23.760(
...	...	...	...	...	...	...	...	...	...	...	
215683	2009-06-01 00:00:00	0.50	0.22	0.39	0.75	0.09	22.000000	24.510000	1.00	82.239998	10.830(
215684	2009-06-01 00:00:00	NaN	0.31	NaN	NaN	NaN	76.110001	101.099998	NaN	41.220001	9.920(
215685	2009-06-01 00:00:00	0.13	NaN	0.86	NaN	0.23	81.050003	99.849998	NaN	24.830000	12.460(
215686	2009-06-01 00:00:00	0.21	NaN	2.96	NaN	0.10	72.419998	82.959999	NaN	NaN	13.030(
215687	2009-06-01 00:00:00	0.37	0.32	0.99	1.36	0.14	54.290001	64.480003	1.06	56.919998	15.360(

215688 rows × 17 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215688 entries, 0 to 215687
Data columns (total 17 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     215688 non-null  object 
 1   BEN      60082 non-null   float64
 2   CO       190801 non-null  float64
 3   EBE      60081 non-null   float64
 4   MXY      24846 non-null   float64
 5   NMHC     74748 non-null   float64
 6   NO_2     214562 non-null  float64
 7   NOx      214565 non-null  float64
 8   OXY      24854 non-null   float64
 9   O_3      204482 non-null  float64
 10  PM10     196331 non-null  float64
 11  PM25     55822 non-null   float64
 12  PXY      24854 non-null   float64
 13  SO_2     212671 non-null  float64
 14  TCH      75213 non-null   float64
 15  TOL      59920 non-null   float64
 16  station  215688 non-null  int64  
dtypes: float64(15), int64(1), object(1)
memory usage: 28.0+ MB
```

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

Out[4]:

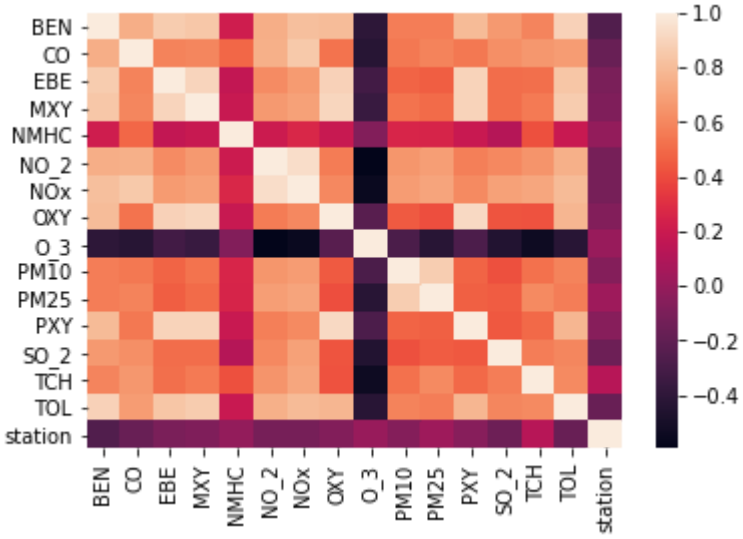
	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	PM1
3	2009-10-01 01:00:00	0.95	0.33	1.43	2.68	0.25	55.180000	81.360001	1.57	36.669998	26.5300(
20	2009-10-01 01:00:00	0.38	0.32	0.32	0.89	0.01	17.969999	19.240000	1.00	65.870003	10.5200(
24	2009-10-01 01:00:00	0.55	0.24	0.65	1.79	0.18	36.619999	43.919998	1.28	48.070000	19.1500(
28	2009-10-01 02:00:00	0.65	0.21	1.20	2.04	0.18	37.169998	48.869999	1.21	26.950001	32.2000(
45	2009-10-01 02:00:00	0.38	0.30	0.50	1.15	0.00	17.889999	19.299999	1.00	60.009998	12.2600(
...	...	...	...	...	...	...	...	...	...	...	
215659	2009-05-31 23:00:00	0.54	0.27	1.00	0.69	0.09	28.280001	29.490000	0.86	78.750000	15.1700(
215663	2009-05-31 23:00:00	0.74	0.35	1.13	1.65	0.15	56.410000	69.870003	1.26	56.799999	11.8000(
215667	2009-06-01 00:00:00	0.78	0.29	0.99	1.96	0.04	64.870003	82.629997	1.13	58.000000	12.6700(
215683	2009-06-01 00:00:00	0.50	0.22	0.39	0.75	0.09	22.000000	24.510000	1.00	82.239998	10.8300(
215687	2009-06-01 00:00:00	0.37	0.32	0.99	1.36	0.14	54.290001	64.480003	1.06	56.919998	15.3600(

24717 rows × 17 columns

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

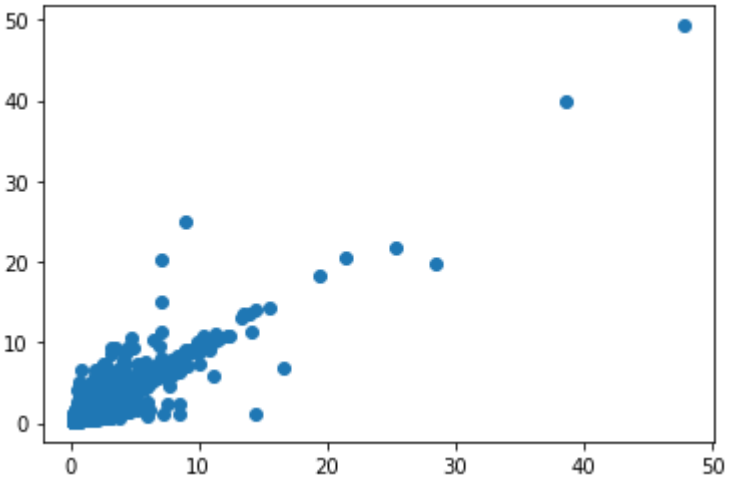
In [6]:
```
sns.heatmap(df1.corr())
```

Out[6]: <AxesSubplot:>



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

Out[7]: [<matplotlib.lines.Line2D at 0x23c87847160>]



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)
```

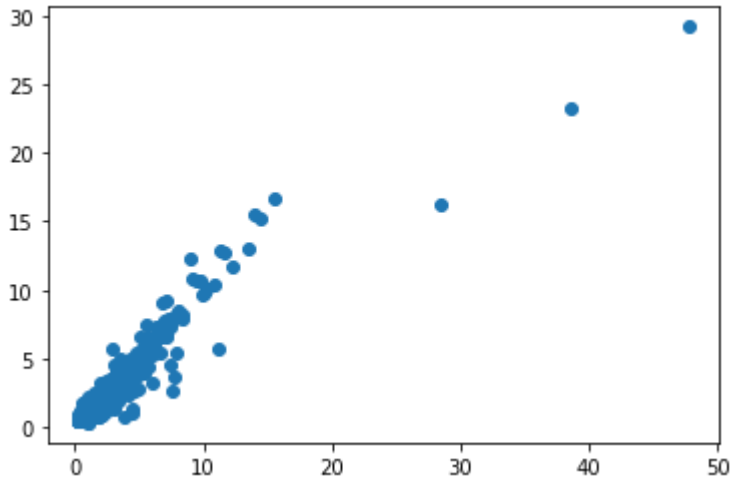
Linear

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 0x23c8790c130>
```



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

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

Out[14]:
```
1.39    1091
1.36    1056
1.38    1046
1.40    1018
1.37    1017
        ...
2.52       1
1.16       1
2.41       1
1.13       1
2.79       1
Name: TCH, Length: 169, 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    12963
2.0    11754
Name: TCH, dtype: int64
```

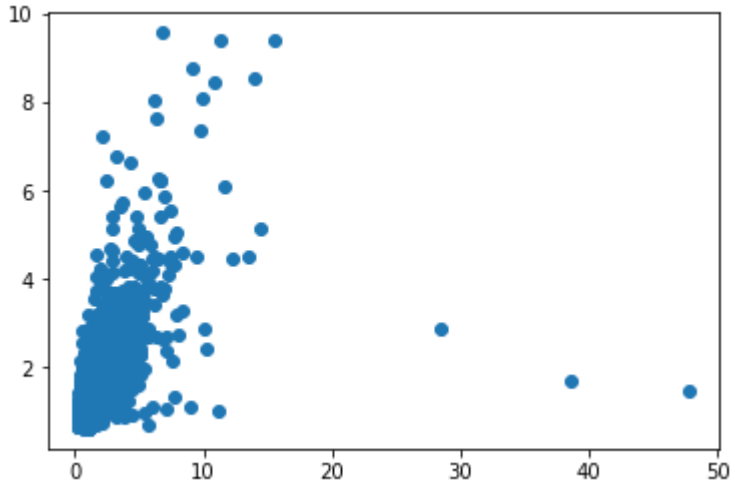
Lasso

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 0x23c88533880>
```



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

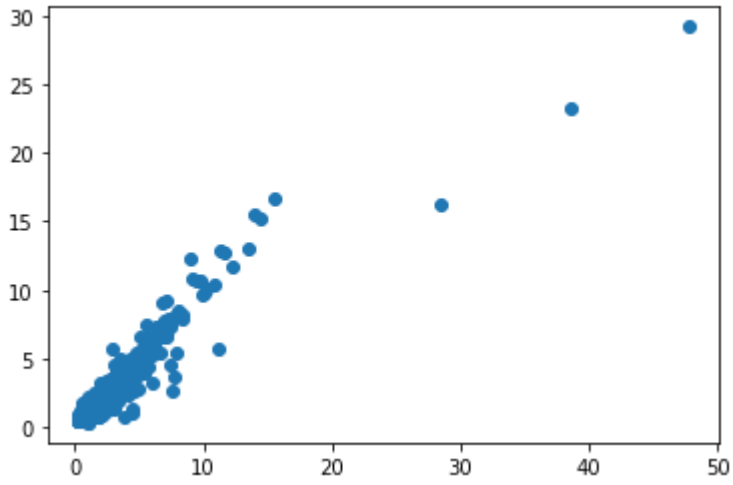
Ridge

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 0x23c871da250>



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

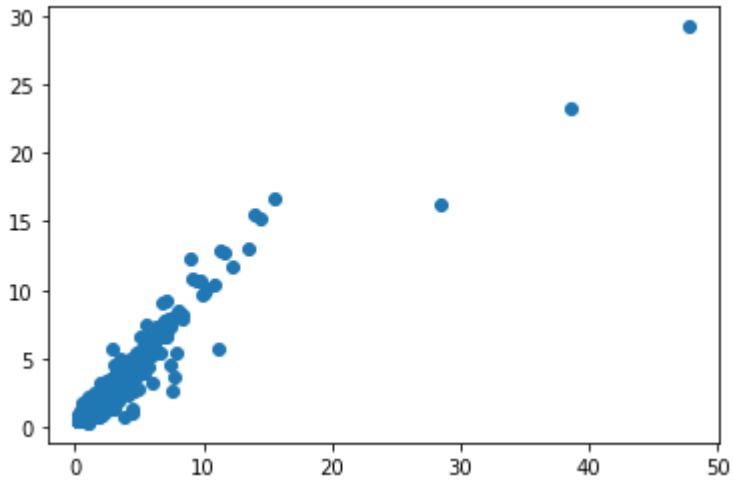
ElasticNet

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 0x23c885beca0>



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

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

```
0.8783497693150732
```

Out[25]:
```
0.8864182797143654
```

Logistic

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

Out[26]:
```
Low     12963
High    11754
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 0x23c88602520>
```

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',"
```

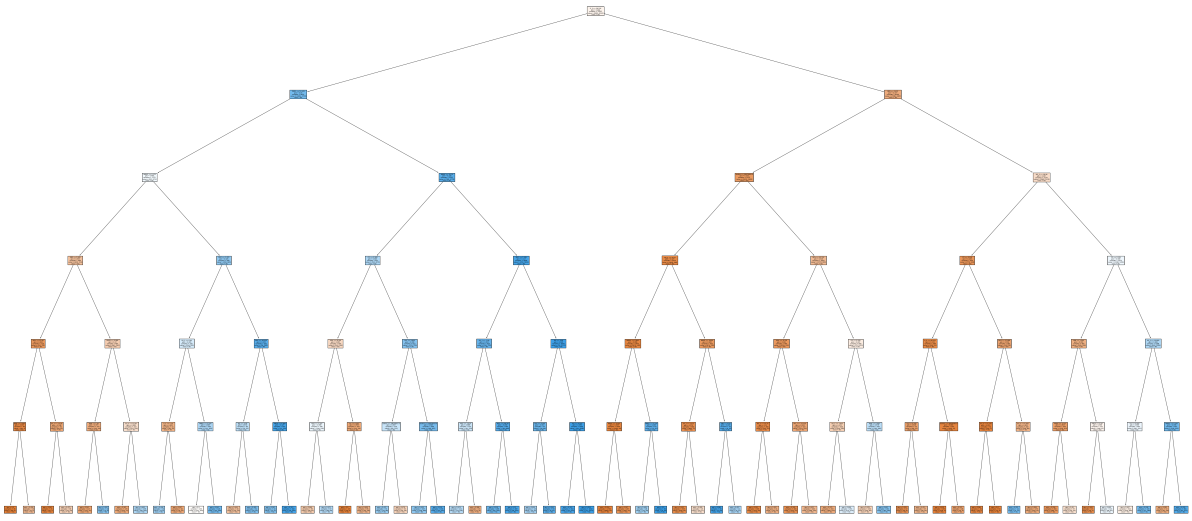
Out[39]:
```
[Text(2232.0, 2019.0857142857144, 'O_3 <= 36.105\ngini = 0.499\nsamples = 109
31\nvalue = [9061, 8240]\nclass = Yes'),
 Text(1116.0, 1708.457142857143, 'PM25 <= 10.105\ngini = 0.35\nsamples = 4675
\nvalue = [1681, 5753]\nclass = No'),
 Text(558.0, 1397.8285714285716, 'NMHC <= 0.175\ngini = 0.499\nsamples = 1175
\nvalue = [912, 974]\nclass = No'),
 Text(279.0, 1087.2, 'EBE <= 0.585\ngini = 0.445\nsamples = 580\nvalue = [62
0, 311]\nclass = Yes'),
 Text(139.5, 776.5714285714287, 'NOx <= 37.82\ngini = 0.262\nsamples = 116\nv
alue = [153, 28]\nclass = Yes'),
 Text(69.75, 465.9428571428573, 'BEN <= 0.565\ngini = 0.104\nsamples = 46\nva
lue = [69, 4]\nclass = Yes'),
 Text(34.875, 155.3142857142857, 'gini = 0.0\nsamples = 36\nvalue = [60, 0]\n
class = Yes'),
 Text(104.625, 155.3142857142857, 'gini = 0.426\nsamples = 10\nvalue = [9, 4]
\nclass = Yes'),
 Text(209.25, 465.9428571428573, 'SO_2 <= 10.58\ngini = 0.346\nsamples = 70\n
value = [84, 24]\nclass = Yes'),
 Text(174.375, 155.3142857142857, 'gini = 0.041\nsamples = 31\nvalue = [47,
1]\nclass = Yes'),
 Text(244.125, 155.3142857142857, 'gini = 0.473\nsamples = 39\nvalue = [37, 2
3]\nclass = Yes'),
 Text(418.5, 776.5714285714287, 'PM25 <= 5.985\ngini = 0.47\nsamples = 464\nv
alue = [467, 283]\nclass = Yes'),
 Text(348.75, 465.9428571428573, 'BEN <= 1.65\ngini = 0.371\nsamples = 127\nv
alue = [150, 49]\nclass = Yes'),
 Text(313.875, 155.3142857142857, 'gini = 0.341\nsamples = 117\nvalue = [147,
41]\nclass = Yes'),
 Text(383.625, 155.3142857142857, 'gini = 0.397\nsamples = 10\nvalue = [3, 8]
\nclass = No'),
 Text(488.25, 465.9428571428573, 'SO_2 <= 10.84\ngini = 0.489\nsamples = 337
\nvalue = [317, 234]\nclass = Yes'),
 Text(453.375, 155.3142857142857, 'gini = 0.367\nsamples = 179\nvalue = [222,
71]\nclass = Yes'),
 Text(523.125, 155.3142857142857, 'gini = 0.465\nsamples = 158\nvalue = [95,
163]\nclass = No'),
 Text(837.0, 1087.2, 'MXY <= 1.475\ngini = 0.425\nsamples = 595\nvalue = [29
2, 663]\nclass = No'),
 Text(697.5, 776.5714285714287, 'SO_2 <= 6.63\ngini = 0.493\nsamples = 334\nv
alue = [234, 295]\nclass = No'),
 Text(627.75, 465.9428571428573, 'O_3 <= 5.19\ngini = 0.402\nsamples = 72\nva
lue = [83, 32]\nclass = Yes'),
 Text(592.875, 155.3142857142857, 'gini = 0.444\nsamples = 10\nvalue = [4, 8]
\nclass = No'),
 Text(662.625, 155.3142857142857, 'gini = 0.357\nsamples = 62\nvalue = [79, 2
4]\nclass = Yes'),
 Text(767.25, 465.9428571428573, 'NMHC <= 0.225\ngini = 0.463\nsamples = 262
\nvalue = [151, 263]\nclass = No'),
 Text(732.375, 155.3142857142857, 'gini = 0.5\nsamples = 136\nvalue = [107, 1
05]\nclass = Yes'),
 Text(802.125, 155.3142857142857, 'gini = 0.341\nsamples = 126\nvalue = [44,
158]\nclass = No'),
 Text(976.5, 776.5714285714287, 'NOx <= 64.325\ngini = 0.235\nsamples = 261\n
value = [58, 368]\nclass = No'),
 Text(906.75, 465.9428571428573, 'TOL <= 2.44\ngini = 0.479\nsamples = 59\nva
lue = [35, 53]\nclass = No'),
 Text(871.875, 155.3142857142857, 'gini = 0.444\nsamples = 18\nvalue = [20, 1
```

```
0]\nclass = Yes'),
 Text(941.625, 155.3142857142857, 'gini = 0.383\nsamples = 41\nvalue = [15, 4
3]\nclass = No'),
 Text(1046.25, 465.9428571428573, 'PM10 <= 6.485\ngini = 0.127\nsamples = 202
\nvalue = [23, 315]\nclass = No'),
 Text(1011.375, 155.3142857142857, 'gini = 0.391\nsamples = 10\nvalue = [4, 1
1]\nclass = No'),
 Text(1081.125, 155.3142857142857, 'gini = 0.111\nsamples = 192\nvalue = [19,
304]\nclass = No'),
 Text(1674.0, 1397.8285714285716, 'NMHC <= 0.175\ngini = 0.239\nsamples = 350
0\nvalue = [769, 4779]\nclass = No'),
 Text(1395.0, 1087.2, 'CO <= 0.385\ngini = 0.465\nsamples = 871\nvalue = [51
0, 875]\nclass = No'),
 Text(1255.5, 776.5714285714287, 'NOx <= 81.99\ngini = 0.483\nsamples = 259\n
value = [240, 166]\nclass = Yes'),
 Text(1185.75, 465.9428571428573, 'MXY <= 1.325\ngini = 0.499\nsamples = 156
\nvalue = [116, 128]\nclass = No'),
 Text(1150.875, 155.3142857142857, 'gini = 0.466\nsamples = 59\nvalue = [58,
34]\nclass = Yes'),
 Text(1220.625, 155.3142857142857, 'gini = 0.472\nsamples = 97\nvalue = [58,
94]\nclass = No'),
 Text(1325.25, 465.9428571428573, 'CO <= 0.305\ngini = 0.359\nsamples = 103\n
value = [124, 38]\nclass = Yes'),
 Text(1290.375, 155.3142857142857, 'gini = 0.0\nsamples = 18\nvalue = [29, 0]
\nclass = Yes'),
 Text(1360.125, 155.3142857142857, 'gini = 0.408\nsamples = 85\nvalue = [95,
38]\nclass = Yes'),
 Text(1534.5, 776.5714285714287, 'PXY <= 0.805\ngini = 0.399\nsamples = 612\n
value = [270, 709]\nclass = No'),
 Text(1464.75, 465.9428571428573, 'station <= 28079015.0\ngini = 0.487\nsampl
es = 116\nvalue = [81, 112]\nclass = No'),
 Text(1429.875, 155.3142857142857, 'gini = 0.437\nsamples = 73\nvalue = [41,
86]\nclass = No'),
 Text(1499.625, 155.3142857142857, 'gini = 0.478\nsamples = 43\nvalue = [40,
26]\nclass = Yes'),
 Text(1604.25, 465.9428571428573, 'station <= 28079015.0\ngini = 0.365\nsampl
es = 496\nvalue = [189, 597]\nclass = No'),
 Text(1569.375, 155.3142857142857, 'gini = 0.419\nsamples = 337\nvalue = [15
6, 366]\nclass = No'),
 Text(1639.125, 155.3142857142857, 'gini = 0.219\nsamples = 159\nvalue = [33,
231]\nclass = No'),
 Text(1953.0, 1087.2, 'MXY <= 1.595\ngini = 0.117\nsamples = 2629\nvalue = [2
59, 3904]\nclass = No'),
 Text(1813.5, 776.5714285714287, 'SO_2 <= 7.28\ngini = 0.303\nsamples = 620\n
value = [181, 792]\nclass = No'),
 Text(1743.75, 465.9428571428573, 'MXY <= 1.515\ngini = 0.488\nsamples = 125
\nvalue = [91, 124]\nclass = No'),
 Text(1708.875, 155.3142857142857, 'gini = 0.474\nsamples = 111\nvalue = [74,
118]\nclass = No'),
 Text(1778.625, 155.3142857142857, 'gini = 0.386\nsamples = 14\nvalue = [17,
6]\nclass = Yes'),
 Text(1883.25, 465.9428571428573, 'NOx <= 106.45\ngini = 0.209\nsamples = 495
\nvalue = [90, 668]\nclass = No'),
 Text(1848.375, 155.3142857142857, 'gini = 0.319\nsamples = 208\nvalue = [61,
245]\nclass = No'),
 Text(1918.125, 155.3142857142857, 'gini = 0.12\nsamples = 287\nvalue = [29,
423]\nclass = No'),
```

```
 Text(2092.5, 776.5714285714287, 'TOL <= 2.82\ngini = 0.048\nsamples = 2009\n
value = [78, 3112]\nclass = No'),
 Text(2022.75, 465.9428571428573, 'OXY <= 0.765\ngini = 0.314\nsamples = 57\n
value = [16, 66]\nclass = No'),
 Text(1987.875, 155.3142857142857, 'gini = 0.454\nsamples = 14\nvalue = [8, 1
5]\nclass = No'),
 Text(2057.625, 155.3142857142857, 'gini = 0.234\nsamples = 43\nvalue = [8, 5
1]\nclass = No'),
 Text(2162.25, 465.9428571428573, 'TOL <= 4.515\ngini = 0.039\nsamples = 1952
\nvalue = [62, 3046]\nclass = No'),
 Text(2127.375, 155.3142857142857, 'gini = 0.115\nsamples = 310\nvalue = [31,
475]\nclass = No'),
 Text(2197.125, 155.3142857142857, 'gini = 0.024\nsamples = 1642\nvalue = [3
1, 2571]\nclass = No'),
 Text(3348.0, 1708.457142857143, 'BEN <= 0.635\ngini = 0.377\nsamples = 6256
\nvalue = [7380, 2487]\nclass = Yes'),
 Text(2790.0, 1397.8285714285716, 'station <= 28079062.0\ngini = 0.275\nsampl
es = 4122\nvalue = [5420, 1069]\nclass = Yes'),
 Text(2511.0, 1087.2, 'NO_2 <= 29.57\ngini = 0.127\nsamples = 2470\nvalue =
[3648, 266]\nclass = Yes'),
 Text(2371.5, 776.5714285714287, 'NMHC <= 0.655\ngini = 0.086\nsamples = 1970
\nvalue = [2974, 140]\nclass = Yes'),
 Text(2301.75, 465.9428571428573, 'NMHC <= 0.455\ngini = 0.038\nsamples = 190
7\nvalue = [2959, 59]\nclass = Yes'),
 Text(2266.875, 155.3142857142857, 'gini = 0.016\nsamples = 1700\nvalue = [26
62, 22]\nclass = Yes'),
 Text(2336.625, 155.3142857142857, 'gini = 0.197\nsamples = 207\nvalue = [29
7, 37]\nclass = Yes'),
 Text(2441.25, 465.9428571428573, 'NMHC <= 0.725\ngini = 0.264\nsamples = 63
\nvalue = [15, 81]\nclass = No'),
 Text(2406.375, 155.3142857142857, 'gini = 0.449\nsamples = 28\nvalue = [15,
29]\nclass = No'),
 Text(2476.125, 155.3142857142857, 'gini = 0.0\nsamples = 35\nvalue = [0, 52]
\nclass = No'),
 Text(2650.5, 776.5714285714287, 'NMHC <= 0.625\ngini = 0.265\nsamples = 500
\nvalue = [674, 126]\nclass = Yes'),
 Text(2580.75, 465.9428571428573, 'SO_2 <= 13.51\ngini = 0.174\nsamples = 459
\nvalue = [666, 71]\nclass = Yes'),
 Text(2545.875, 155.3142857142857, 'gini = 0.143\nsamples = 434\nvalue = [64
3, 54]\nclass = Yes'),
 Text(2615.625, 155.3142857142857, 'gini = 0.489\nsamples = 25\nvalue = [23,
17]\nclass = Yes'),
 Text(2720.25, 465.9428571428573, 'EBE <= 0.83\ngini = 0.222\nsamples = 41\nv
alue = [8, 55]\nclass = No'),
 Text(2685.375, 155.3142857142857, 'gini = 0.083\nsamples = 30\nvalue = [2, 4
4]\nclass = No'),
 Text(2755.125, 155.3142857142857, 'gini = 0.457\nsamples = 11\nvalue = [6, 1
1]\nclass = No'),
 Text(3069.0, 1087.2, 'NOx <= 45.155\ngini = 0.429\nsamples = 1652\nvalue =
[1772, 803]\nclass = Yes'),
 Text(2929.5, 776.5714285714287, 'NOx <= 31.29\ngini = 0.261\nsamples = 796\n
value = [1055, 193]\nclass = Yes'),
 Text(2859.75, 465.9428571428573, 'NO_2 <= 23.41\ngini = 0.13\nsamples = 369
\nvalue = [544, 41]\nclass = Yes'),
 Text(2824.875, 155.3142857142857, 'gini = 0.101\nsamples = 338\nvalue = [51
2, 29]\nclass = Yes'),
 Text(2894.625, 155.3142857142857, 'gini = 0.397\nsamples = 31\nvalue = [32,
```

```
12]\nclass = Yes'),
 Text(2999.25, 465.9428571428573, 'NO_2 <= 27.525\ngini = 0.353\nsamples = 42
7\nvalue = [511, 152]\nclass = Yes'),
 Text(2964.375, 155.3142857142857, 'gini = 0.193\nsamples = 164\nvalue = [22
2, 27]\nclass = Yes'),
 Text(3034.125, 155.3142857142857, 'gini = 0.422\nsamples = 263\nvalue = [28
9, 125]\nclass = Yes'),
 Text(3208.5, 776.5714285714287, 'CO <= 0.335\ngini = 0.497\nsamples = 856\nv
alue = [717, 610]\nclass = Yes'),
 Text(3138.75, 465.9428571428573, 'SO_2 <= 7.635\ngini = 0.472\nsamples = 559
\nvalue = [534, 330]\nclass = Yes'),
 Text(3103.875, 155.3142857142857, 'gini = 0.375\nsamples = 319\nvalue = [36
0, 120]\nclass = Yes'),
 Text(3173.625, 155.3142857142857, 'gini = 0.496\nsamples = 240\nvalue = [17
4, 210]\nclass = No'),
 Text(3278.25, 465.9428571428573, 'BEN <= 0.485\ngini = 0.478\nsamples = 297
\nvalue = [183, 280]\nclass = No'),
 Text(3243.375, 155.3142857142857, 'gini = 0.476\nsamples = 106\nvalue = [10
0, 64]\nclass = Yes'),
 Text(3313.125, 155.3142857142857, 'gini = 0.401\nsamples = 191\nvalue = [83,
216]\nclass = No'),
 Text(3906.0, 1397.8285714285716, 'NO_2 <= 39.96\ngini = 0.487\nsamples = 213
4\nvalue = [1960, 1418]\nclass = Yes'),
 Text(3627.0, 1087.2, 'CO <= 0.305\ngini = 0.239\nsamples = 559\nvalue = [77
0, 124]\nclass = Yes'),
 Text(3487.5, 776.5714285714287, 'O_3 <= 49.575\ngini = 0.134\nsamples = 299
\nvalue = [438, 34]\nclass = Yes'),
 Text(3417.75, 465.9428571428573, 'CO <= 0.245\ngini = 0.318\nsamples = 74\nv
alue = [89, 22]\nclass = Yes'),
 Text(3382.875, 155.3142857142857, 'gini = 0.191\nsamples = 37\nvalue = [50,
6]\nclass = Yes'),
 Text(3452.625, 155.3142857142857, 'gini = 0.413\nsamples = 37\nvalue = [39,
16]\nclass = Yes'),
 Text(3557.25, 465.9428571428573, 'station <= 28079062.0\ngini = 0.064\nsampl
es = 225\nvalue = [349, 12]\nclass = Yes'),
 Text(3522.375, 155.3142857142857, 'gini = 0.019\nsamples = 197\nvalue = [31
6, 3]\nclass = Yes'),
 Text(3592.125, 155.3142857142857, 'gini = 0.337\nsamples = 28\nvalue = [33,
9]\nclass = Yes'),
 Text(3766.5, 776.5714285714287, 'NOx <= 27.985\ngini = 0.336\nsamples = 260
\nvalue = [332, 90]\nclass = Yes'),
 Text(3696.75, 465.9428571428573, 'SO_2 <= 6.775\ngini = 0.019\nsamples = 73
\nvalue = [104, 1]\nclass = Yes'),
 Text(3661.875, 155.3142857142857, 'gini = 0.064\nsamples = 20\nvalue = [29,
1]\nclass = Yes'),
 Text(3731.625, 155.3142857142857, 'gini = 0.0\nsamples = 53\nvalue = [75, 0]
\nclass = Yes'),
 Text(3836.25, 465.9428571428573, 'SO_2 <= 6.305\ngini = 0.404\nsamples = 187
\nvalue = [228, 89]\nclass = Yes'),
 Text(3801.375, 155.3142857142857, 'gini = 0.351\nsamples = 12\nvalue = [5, 1
7]\nclass = No'),
 Text(3871.125, 155.3142857142857, 'gini = 0.369\nsamples = 175\nvalue = [22
3, 72]\nclass = Yes'),
 Text(4185.0, 1087.2, 'CO <= 0.355\ngini = 0.499\nsamples = 1575\nvalue = [11
90, 1294]\nclass = No'),
 Text(4045.5, 776.5714285714287, 'EBE <= 1.295\ngini = 0.391\nsamples = 487\n
value = [545, 198]\nclass = Yes'),
```

```
 Text(3975.75, 465.9428571428573, 'PM25 <= 17.23\ngini = 0.304\nsamples = 350
\nvalue = [439, 101]\nclass = Yes'),
 Text(3940.875, 155.3142857142857, 'gini = 0.256\nsamples = 303\nvalue = [39
9, 71]\nclass = Yes'),
 Text(4010.625, 155.3142857142857, 'gini = 0.49\nsamples = 47\nvalue = [40, 3
0]\nclass = Yes'),
 Text(4115.25, 465.9428571428573, 'PM25 <= 5.83\ngini = 0.499\nsamples = 137
\nvalue = [106, 97]\nclass = Yes'),
 Text(4080.375, 155.3142857142857, 'gini = 0.165\nsamples = 17\nvalue = [20,
2]\nclass = Yes'),
 Text(4150.125, 155.3142857142857, 'gini = 0.499\nsamples = 120\nvalue = [86,
95]\nclass = No'),
 Text(4324.5, 776.5714285714287, 'SO_2 <= 10.805\ngini = 0.466\nsamples = 108
8\nvalue = [645, 1096]\nclass = No'),
 Text(4254.75, 465.9428571428573, 'CO <= 0.605\ngini = 0.499\nsamples = 655\n
value = [497, 551]\nclass = No'),
 Text(4219.875, 155.3142857142857, 'gini = 0.497\nsamples = 522\nvalue = [45
0, 385]\nclass = Yes'),
 Text(4289.625, 155.3142857142857, 'gini = 0.344\nsamples = 133\nvalue = [47,
166]\nclass = No'),
 Text(4394.25, 465.9428571428573, 'NMHC <= 0.105\ngini = 0.336\nsamples = 433
\nvalue = [148, 545]\nclass = No'),
 Text(4359.375, 155.3142857142857, 'gini = 0.382\nsamples = 72\nvalue = [81,
28]\nclass = Yes'),
 Text(4429.125, 155.3142857142857, 'gini = 0.203\nsamples = 361\nvalue = [67,
517]\nclass = No')]
```



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

```
Linear: 0.8783680625661898
Lasso: 0.37328599597971324
Ridge: 0.8783497693150732
ElasticNet: 0.575045121851463
Logistic: 0.5225188781014024
Random Forest: 0.8625513973793
```

Best Model is Linear Regression

2010

In [41]:
```
df2=pd.read_csv("madrid_2010.csv")
df2
```

Out[41]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	P
0	2010-03-01 01:00:00	NaN	0.29	NaN	NaN	NaN	25.090000	29.219999	NaN	68.930000	
1	2010-03-01 01:00:00	NaN	0.27	NaN	NaN	NaN	24.879999	30.040001	NaN	NaN	
2	2010-03-01 01:00:00	NaN	0.28	NaN	NaN	NaN	17.410000	20.540001	NaN	72.120003	
3	2010-03-01 01:00:00	0.38	0.24	1.74	NaN	0.05	15.610000	21.080000	NaN	72.970001	19.41(
4	2010-03-01 01:00:00	0.79	NaN	1.32	NaN	NaN	21.430000	26.070000	NaN	NaN	24.67(
...	...	...	...	...	...	...	...	...	...	...	
209443	2010-08-01 00:00:00	NaN	0.55	NaN	NaN	NaN	125.000000	219.899994	NaN	25.379999	
209444	2010-08-01 00:00:00	NaN	0.27	NaN	NaN	NaN	45.709999	47.410000	NaN	NaN	51.25(
209445	2010-08-01 00:00:00	NaN	NaN	NaN	NaN	0.24	46.560001	49.040001	NaN	46.250000	
209446	2010-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	46.770000	50.119999	NaN	77.709999	
209447	2010-08-01 00:00:00	0.92	0.43	0.71	NaN	0.25	76.330002	88.190002	NaN	52.259998	47.15(

209448 rows × 17 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209448 entries, 0 to 209447
Data columns (total 17 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     209448 non-null  object 
 1   BEN      60268 non-null   float64
 2   CO       94982 non-null   float64
 3   EBE      60253 non-null   float64
 4   MXY      6750 non-null    float64
 5   NMHC     51727 non-null   float64
 6   NO_2     208219 non-null  float64
 7   NOx      208210 non-null  float64
 8   OXY      6750 non-null    float64
 9   O_3      126684 non-null  float64
 10  PM10     106186 non-null  float64
 11  PM25     55514 non-null   float64
 12  PXY      6740 non-null    float64
 13  SO_2     93184 non-null   float64
 14  TCH      51730 non-null   float64
 15  TOL      60171 non-null   float64
 16  station  209448 non-null  int64  
dtypes: float64(15), int64(1), object(1)
memory usage: 27.2+ MB
```

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

Out[43]:

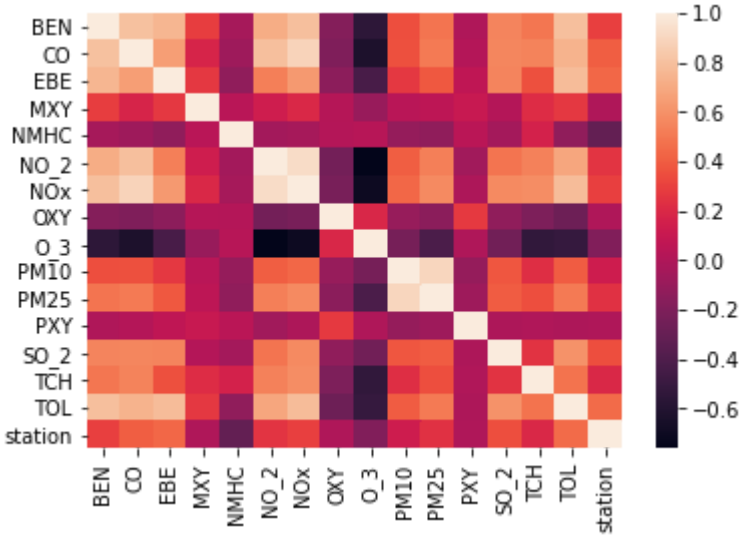
	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	PM1
11	2010-03-01 01:00:00	0.78	0.18	0.84	0.73	0.28	10.420000	11.900000	1.0	90.309998	18.3700
23	2010-03-01 01:00:00	0.70	0.23	1.00	0.73	0.18	17.820000	22.290001	1.0	70.550003	23.6399
35	2010-03-01 02:00:00	0.58	0.17	0.84	0.73	0.28	3.500000	4.950000	1.0	68.849998	5.6000
47	2010-03-01 02:00:00	0.33	0.21	0.84	0.73	0.17	10.810000	14.900000	1.0	74.750000	7.8900
59	2010-03-01 03:00:00	0.38	0.16	0.64	1.00	0.26	2.750000	4.200000	1.0	93.629997	5.1300
...	...	...	...	...	...	...	...	...	...	...	
191879	2010-05-31 22:00:00	0.60	0.26	0.82	0.13	0.16	33.360001	43.779999	1.0	38.459999	20.3400
191891	2010-05-31 23:00:00	0.41	0.16	0.71	0.19	0.10	24.299999	26.059999	1.0	50.290001	14.3800
191903	2010-05-31 23:00:00	0.57	0.28	0.64	0.19	0.18	35.540001	44.590000	1.0	34.020000	22.8400
191915	2010-06-01 00:00:00	0.34	0.16	0.69	0.22	0.10	23.559999	25.209999	1.0	45.930000	10.7700
191927	2010-06-01 00:00:00	0.43	0.25	0.79	0.22	0.18	34.910000	42.369999	1.0	29.540001	15.3500

6666 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

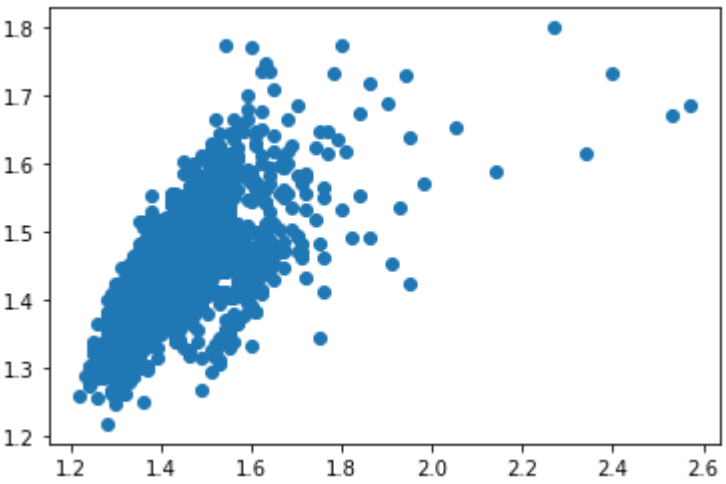
In [47]:
```
li=LinearRegression()
li.fit(x_train,y_train)
```

Out[47]: LinearRegression()

In [ ]:
```

```

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

Out[48]: <matplotlib.collections.PathCollection at 0x23c8864f190>



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

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

Out[50]:
```
1.36    364
1.38    351
1.39    324
1.35    323
1.37    321
       ...
2.07      1
2.17      1
2.53      1
2.12      1
2.05      1
Name: TCH, Length: 100, 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    3340
2.0    3326
Name: TCH, dtype: int64
```

In [ ]:
```

```

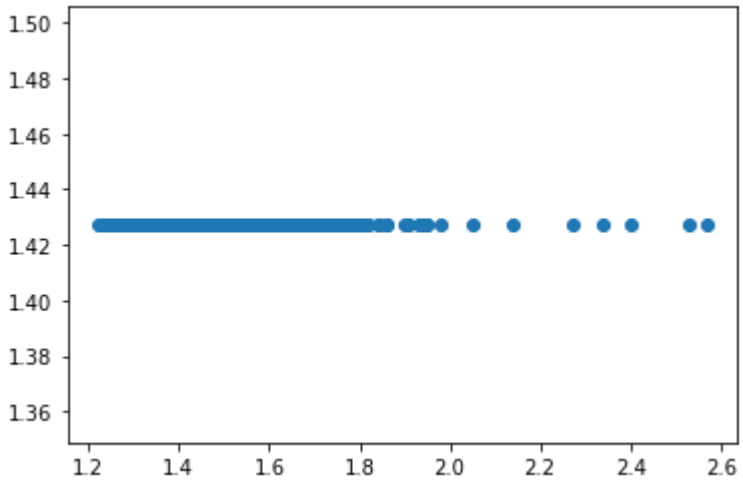
Lasso

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 0x23c886ac280>



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

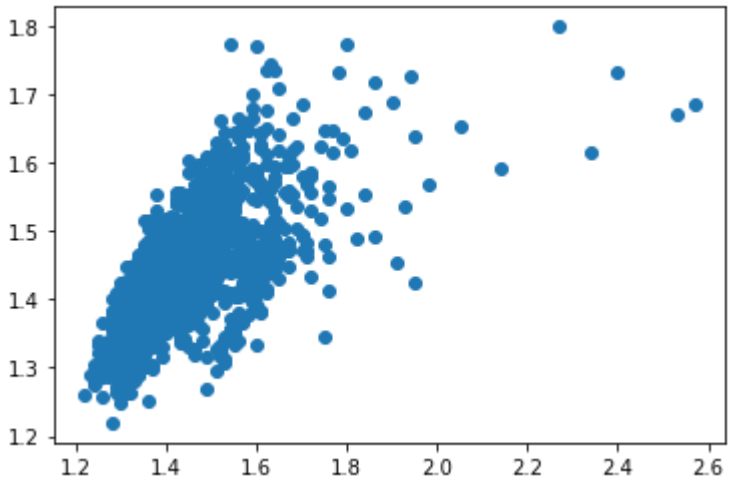
Ridge

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 0x23c8870a310>



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

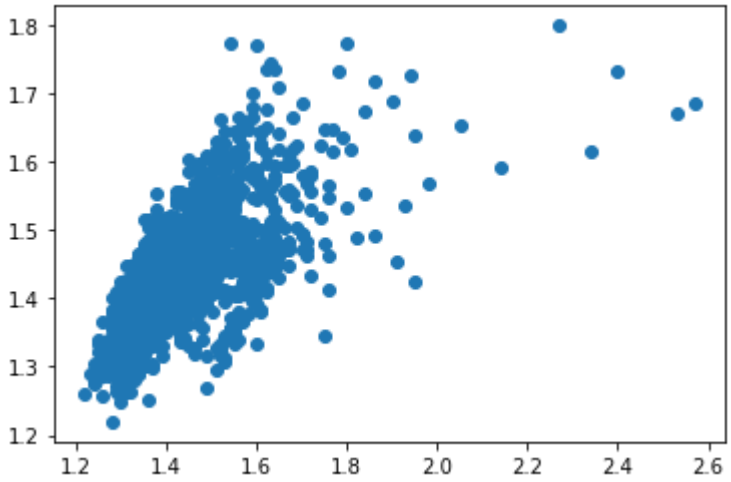
ElasticNet

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 0x23c88762e20>
```



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

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

```
0.4590393479321705
```

Out[61]:
```
0.44781912704161386
```

Logistic

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

Out[62]:
```
Low     3340
High    3326
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 0x23c8813aac0>
```

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_
```

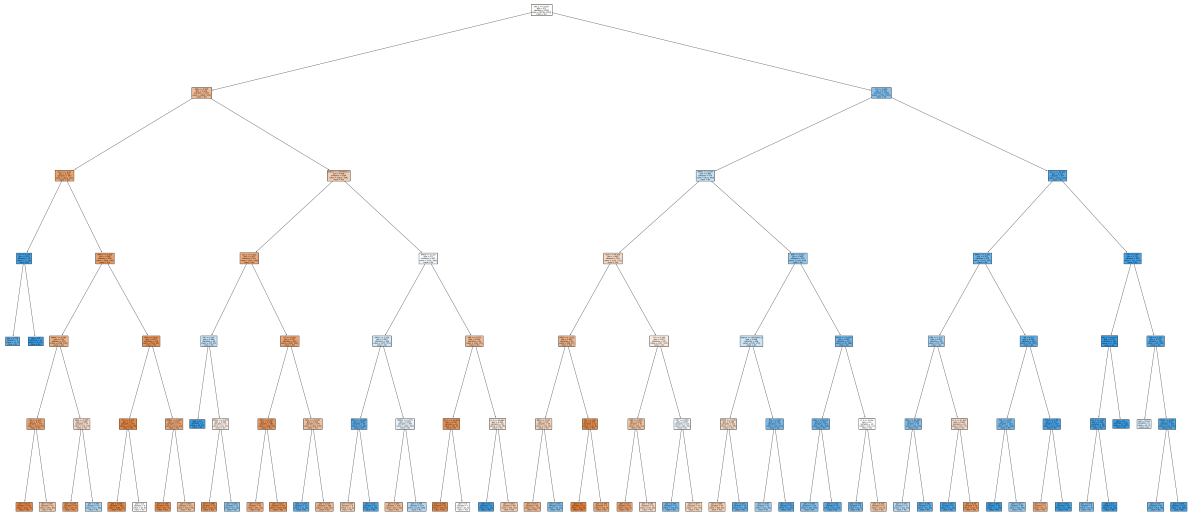
In [75]:
```
from sklearn.tree import plot_tree

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

Out[75]:
```
[Text(2028.8446601941748, 2019.0857142857144, 'NO_2 <= 32.59\ngini = 0.5\nsam
ples = 2951\nvalue = [2356, 2310]\nclass = Yes'),
 Text(753.0291262135922, 1708.457142857143, 'EBE <= 0.505\ngini = 0.428\nsamp
les = 1655\nvalue = [1802, 810]\nclass = Yes'),
 Text(238.36893203883497, 1397.8285714285716, 'SO_2 <= 6.42\ngini = 0.333\nsa
mples = 647\nvalue = [801, 214]\nclass = Yes'),
 Text(86.67961165048544, 1087.2, 'CO <= 0.175\ngini = 0.071\nsamples = 18\nva
lue = [1, 26]\nclass = No'),
 Text(43.33980582524272, 776.5714285714287, 'gini = 0.278\nsamples = 5\nvalue
= [1, 5]\nclass = No'),
 Text(130.01941747572818, 776.5714285714287, 'gini = 0.0\nsamples = 13\nvalue
= [0, 21]\nclass = No'),
 Text(390.05825242718447, 1087.2, 'PM10 <= 9.235\ngini = 0.308\nsamples = 629
\nvalue = [800, 188]\nclass = Yes'),
 Text(216.6990291262136, 776.5714285714287, 'NMHC <= 0.385\ngini = 0.39\nsamp
les = 251\nvalue = [298, 108]\nclass = Yes'),
 Text(130.01941747572818, 465.9428571428573, 'CO <= 0.215\ngini = 0.317\nsamp
les = 174\nvalue = [224, 55]\nclass = Yes'),
 Text(86.67961165048544, 155.3142857142857, 'gini = 0.113\nsamples = 101\nval
ue = [141, 9]\nclass = Yes'),
 Text(173.35922330097088, 155.3142857142857, 'gini = 0.459\nsamples = 73\nval
ue = [83, 46]\nclass = Yes'),
 Text(303.37864077669906, 465.9428571428573, 'BEN <= 0.265\ngini = 0.486\nsam
ples = 77\nvalue = [74, 53]\nclass = Yes'),
 Text(260.03883495145635, 155.3142857142857, 'gini = 0.226\nsamples = 34\nval
ue = [47, 7]\nclass = Yes'),
 Text(346.71844660194176, 155.3142857142857, 'gini = 0.466\nsamples = 43\nval
ue = [27, 46]\nclass = No'),
 Text(563.4174757281554, 776.5714285714287, 'NOx <= 19.02\ngini = 0.237\nsamp
les = 378\nvalue = [502, 80]\nclass = Yes'),
 Text(476.73786407766994, 465.9428571428573, 'CO <= 0.225\ngini = 0.11\nsampl
es = 201\nvalue = [291, 18]\nclass = Yes'),
 Text(433.3980582524272, 155.3142857142857, 'gini = 0.078\nsamples = 192\nval
ue = [285, 12]\nclass = Yes'),
 Text(520.0776699029127, 155.3142857142857, 'gini = 0.5\nsamples = 9\nvalue =
[6, 6]\nclass = Yes'),
 Text(650.0970873786408, 465.9428571428573, 'NMHC <= 0.185\ngini = 0.351\nsam
ples = 177\nvalue = [211, 62]\nclass = Yes'),
 Text(606.7572815533981, 155.3142857142857, 'gini = 0.155\nsamples = 65\nvalu
e = [97, 9]\nclass = Yes'),
 Text(693.4368932038835, 155.3142857142857, 'gini = 0.433\nsamples = 112\nval
ue = [114, 53]\nclass = Yes'),
 Text(1267.6893203883496, 1397.8285714285716, 'station <= 28079062.0\ngini =
0.468\nsamples = 1008\nvalue = [1001, 596]\nclass = Yes'),
 Text(931.8058252427185, 1087.2, 'SO_2 <= 7.225\ngini = 0.367\nsamples = 521
\nvalue = [625, 200]\nclass = Yes'),
 Text(780.1165048543689, 776.5714285714287, 'PXY <= 0.47\ngini = 0.484\nsampl
es = 38\nvalue = [25, 36]\nclass = No'),
 Text(736.7766990291262, 465.9428571428573, 'gini = 0.0\nsamples = 7\nvalue =
[0, 14]\nclass = No'),
 Text(823.4563106796116, 465.9428571428573, 'NMHC <= 0.305\ngini = 0.498\nsam
ples = 31\nvalue = [25, 22]\nclass = Yes'),
 Text(780.1165048543689, 155.3142857142857, 'gini = 0.198\nsamples = 14\nvalu
e = [16, 2]\nclass = Yes'),
 Text(866.7961165048544, 155.3142857142857, 'gini = 0.428\nsamples = 17\nvalu
e = [9, 20]\nclass = No'),
 Text(1083.495145631068, 776.5714285714287, 'CO <= 0.195\ngini = 0.337\nsampl
```

```
es = 483\nvalue = [600, 164]\nclass = Yes'),
 Text(996.8155339805826, 465.9428571428573, 'MXY <= 0.325\ngini = 0.238\nsamp
les = 284\nvalue = [400, 64]\nclass = Yes'),
 Text(953.4757281553399, 155.3142857142857, 'gini = 0.354\nsamples = 37\nvalu
e = [47, 14]\nclass = Yes'),
 Text(1040.1553398058254, 155.3142857142857, 'gini = 0.217\nsamples = 247\nva
lue = [353, 50]\nclass = Yes'),
 Text(1170.1747572815534, 465.9428571428573, 'NOx <= 10.055\ngini = 0.444\nsa
mples = 199\nvalue = [200, 100]\nclass = Yes'),
 Text(1126.8349514563108, 155.3142857142857, 'gini = 0.245\nsamples = 14\nval
ue = [3, 18]\nclass = No'),
 Text(1213.5145631067962, 155.3142857142857, 'gini = 0.415\nsamples = 185\nva
lue = [197, 82]\nclass = Yes'),
 Text(1603.5728155339807, 1087.2, 'PM25 <= 11.25\ngini = 0.5\nsamples = 487\n
value = [376, 396]\nclass = No'),
 Text(1430.2135922330099, 776.5714285714287, 'PM10 <= 5.105\ngini = 0.493\nsa
mples = 389\nvalue = [275, 351]\nclass = No'),
 Text(1343.5339805825242, 465.9428571428573, 'NOx <= 14.835\ngini = 0.266\nsa
mples = 30\nvalue = [9, 48]\nclass = No'),
 Text(1300.1941747572816, 155.3142857142857, 'gini = 0.469\nsamples = 5\nvalu
e = [5, 3]\nclass = Yes'),
 Text(1386.873786407767, 155.3142857142857, 'gini = 0.15\nsamples = 25\nvalue
= [4, 45]\nclass = No'),
 Text(1516.8932038834953, 465.9428571428573, 'NMHC <= 0.175\ngini = 0.498\nsa
mples = 359\nvalue = [266, 303]\nclass = No'),
 Text(1473.5533980582525, 155.3142857142857, 'gini = 0.435\nsamples = 105\nva
lue = [117, 55]\nclass = Yes'),
 Text(1560.2330097087379, 155.3142857142857, 'gini = 0.469\nsamples = 254\nva
lue = [149, 248]\nclass = No'),
 Text(1776.9320388349515, 776.5714285714287, 'EBE <= 0.775\ngini = 0.426\nsam
ples = 98\nvalue = [101, 45]\nclass = Yes'),
 Text(1690.2524271844661, 465.9428571428573, 'SO_2 <= 10.265\ngini = 0.236\ns
amples = 42\nvalue = [57, 9]\nclass = Yes'),
 Text(1646.9126213592233, 155.3142857142857, 'gini = 0.158\nsamples = 35\nval
ue = [53, 5]\nclass = Yes'),
 Text(1733.5922330097087, 155.3142857142857, 'gini = 0.5\nsamples = 7\nvalue
= [4, 4]\nclass = Yes'),
 Text(1863.611650485437, 465.9428571428573, 'O_3 <= 49.865\ngini = 0.495\nsam
ples = 56\nvalue = [44, 36]\nclass = Yes'),
 Text(1820.2718446601943, 155.3142857142857, 'gini = 0.0\nsamples = 8\nvalue
= [0, 11]\nclass = No'),
 Text(1906.9514563106798, 155.3142857142857, 'gini = 0.462\nsamples = 48\nval
ue = [44, 25]\nclass = Yes'),
 Text(3304.6601941747576, 1708.457142857143, 'CO <= 0.345\ngini = 0.394\nsamp
les = 1296\nvalue = [554, 1500]\nclass = No'),
 Text(2643.728155339806, 1397.8285714285716, 'NMHC <= 0.195\ngini = 0.486\nsa
mples = 619\nvalue = [419, 590]\nclass = No'),
 Text(2297.009708737864, 1087.2, 'NOx <= 44.935\ngini = 0.485\nsamples = 226
\nvalue = [215, 152]\nclass = Yes'),
 Text(2123.650485436893, 776.5714285714287, 'PM25 <= 11.55\ngini = 0.407\nsam
ples = 59\nvalue = [68, 27]\nclass = Yes'),
 Text(2036.9708737864078, 465.9428571428573, 'BEN <= 0.615\ngini = 0.48\nsamp
les = 38\nvalue = [36, 24]\nclass = Yes'),
 Text(1993.6310679611652, 155.3142857142857, 'gini = 0.411\nsamples = 28\nval
ue = [32, 13]\nclass = Yes'),
 Text(2080.310679611651, 155.3142857142857, 'gini = 0.391\nsamples = 10\nvalu
e = [4, 11]\nclass = No'),
```

```
 Text(2210.330097087379, 465.9428571428573, 'TOL <= 1.49\ngini = 0.157\nsampl
es = 21\nvalue = [32, 3]\nclass = Yes'),
 Text(2166.990291262136, 155.3142857142857, 'gini = 0.0\nsamples = 12\nvalue
= [22, 0]\nclass = Yes'),
 Text(2253.6699029126216, 155.3142857142857, 'gini = 0.355\nsamples = 9\nvalu
e = [10, 3]\nclass = Yes'),
 Text(2470.3689320388353, 776.5714285714287, 'CO <= 0.295\ngini = 0.497\nsamp
les = 167\nvalue = [147, 125]\nclass = Yes'),
 Text(2383.6893203883496, 465.9428571428573, 'EBE <= 0.495\ngini = 0.449\nsam
ples = 57\nvalue = [60, 31]\nclass = Yes'),
 Text(2340.349514563107, 155.3142857142857, 'gini = 0.32\nsamples = 16\nvalue
= [24, 6]\nclass = Yes'),
 Text(2427.0291262135925, 155.3142857142857, 'gini = 0.484\nsamples = 41\nval
ue = [36, 25]\nclass = Yes'),
 Text(2557.0485436893205, 465.9428571428573, 'PXY <= 0.565\ngini = 0.499\nsam
ples = 110\nvalue = [87, 94]\nclass = No'),
 Text(2513.7087378640776, 155.3142857142857, 'gini = 0.354\nsamples = 37\nval
ue = [14, 47]\nclass = No'),
 Text(2600.3883495145633, 155.3142857142857, 'gini = 0.477\nsamples = 73\nval
ue = [73, 47]\nclass = Yes'),
 Text(2990.4466019417478, 1087.2, 'BEN <= 0.615\ngini = 0.434\nsamples = 393
\nvalue = [204, 438]\nclass = No'),
 Text(2817.087378640777, 776.5714285714287, 'station <= 28079062.0\ngini = 0.
489\nsamples = 186\nvalue = [126, 170]\nclass = No'),
 Text(2730.4077669902913, 465.9428571428573, 'PM25 <= 11.65\ngini = 0.486\nsa
mples = 111\nvalue = [97, 69]\nclass = Yes'),
 Text(2687.0679611650485, 155.3142857142857, 'gini = 0.466\nsamples = 97\nval
ue = [94, 55]\nclass = Yes'),
 Text(2773.747572815534, 155.3142857142857, 'gini = 0.291\nsamples = 14\nvalu
e = [3, 14]\nclass = No'),
 Text(2903.766990291262, 465.9428571428573, 'TOL <= 2.09\ngini = 0.347\nsampl
es = 75\nvalue = [29, 101]\nclass = No'),
 Text(2860.4271844660198, 155.3142857142857, 'gini = 0.448\nsamples = 35\nval
ue = [20, 39]\nclass = No'),
 Text(2947.106796116505, 155.3142857142857, 'gini = 0.221\nsamples = 40\nvalu
e = [9, 62]\nclass = No'),
 Text(3163.8058252427186, 776.5714285714287, 'PM25 <= 16.365\ngini = 0.349\ns
amples = 207\nvalue = [78, 268]\nclass = No'),
 Text(3077.126213592233, 465.9428571428573, 'NOx <= 59.515\ngini = 0.326\nsam
ples = 191\nvalue = [66, 256]\nclass = No'),
 Text(3033.7864077669906, 155.3142857142857, 'gini = 0.385\nsamples = 113\nva
lue = [50, 142]\nclass = No'),
 Text(3120.4660194174758, 155.3142857142857, 'gini = 0.216\nsamples = 78\nval
ue = [16, 114]\nclass = No'),
 Text(3250.485436893204, 465.9428571428573, 'O_3 <= 33.54\ngini = 0.5\nsample
s = 16\nvalue = [12, 12]\nclass = Yes'),
 Text(3207.1456310679614, 155.3142857142857, 'gini = 0.346\nsamples = 6\nvalu
e = [2, 7]\nclass = No'),
 Text(3293.8252427184466, 155.3142857142857, 'gini = 0.444\nsamples = 10\nval
ue = [10, 5]\nclass = Yes'),
 Text(3965.5922330097087, 1397.8285714285716, 'NO_2 <= 58.595\ngini = 0.225\n
samples = 677\nvalue = [135, 910]\nclass = No'),
 Text(3683.883495145631, 1087.2, 'NMHC <= 0.185\ngini = 0.327\nsamples = 318
\nvalue = [103, 397]\nclass = No'),
 Text(3510.5242718446602, 776.5714285714287, 'EBE <= 1.375\ngini = 0.453\nsam
ples = 98\nvalue = [52, 98]\nclass = No'),
 Text(3423.844660194175, 465.9428571428573, 'NOx <= 64.465\ngini = 0.388\nsam
```

```
ples = 74\nvalue = [29, 81]\nclass = No'),
 Text(3380.5048543689322, 155.3142857142857, 'gini = 0.473\nsamples = 37\nval
ue = [20, 32]\nclass = No'),
 Text(3467.1844660194174, 155.3142857142857, 'gini = 0.262\nsamples = 37\nval
ue = [9, 49]\nclass = No'),
 Text(3597.203883495146, 465.9428571428573, 'O_3 <= 21.545\ngini = 0.489\nsam
ples = 24\nvalue = [23, 17]\nclass = Yes'),
 Text(3553.864077669903, 155.3142857142857, 'gini = 0.133\nsamples = 8\nvalue
= [1, 13]\nclass = No'),
 Text(3640.5436893203887, 155.3142857142857, 'gini = 0.26\nsamples = 16\nvalu
e = [22, 4]\nclass = Yes'),
 Text(3857.2427184466023, 776.5714285714287, 'TOL <= 3.5\ngini = 0.249\nsampl
es = 220\nvalue = [51, 299]\nclass = No'),
 Text(3770.5631067961167, 465.9428571428573, 'TOL <= 2.165\ngini = 0.34\nsamp
les = 120\nvalue = [41, 148]\nclass = No'),
 Text(3727.223300970874, 155.3142857142857, 'gini = 0.038\nsamples = 34\nvalu
e = [1, 50]\nclass = No'),
 Text(3813.9029126213595, 155.3142857142857, 'gini = 0.412\nsamples = 86\nval
ue = [40, 98]\nclass = No'),
 Text(3943.9223300970875, 465.9428571428573, 'PM25 <= 5.56\ngini = 0.117\nsam
ples = 100\nvalue = [10, 151]\nclass = No'),
 Text(3900.5825242718447, 155.3142857142857, 'gini = 0.375\nsamples = 5\nvalu
e = [6, 2]\nclass = Yes'),
 Text(3987.2621359223303, 155.3142857142857, 'gini = 0.051\nsamples = 95\nval
ue = [4, 149]\nclass = No'),
 Text(4247.300970873786, 1087.2, 'OXY <= 0.99\ngini = 0.111\nsamples = 359\nv
alue = [32, 513]\nclass = No'),
 Text(4160.621359223302, 776.5714285714287, 'NO_2 <= 63.34\ngini = 0.017\nsam
ples = 149\nvalue = [2, 233]\nclass = No'),
 Text(4117.281553398058, 465.9428571428573, 'SO_2 <= 9.97\ngini = 0.124\nsamp
les = 20\nvalue = [2, 28]\nclass = No'),
 Text(4073.9417475728155, 155.3142857142857, 'gini = 0.375\nsamples = 5\nvalu
e = [2, 6]\nclass = No'),
 Text(4160.621359223302, 155.3142857142857, 'gini = 0.0\nsamples = 15\nvalue
= [0, 22]\nclass = No'),
 Text(4203.961165048544, 465.9428571428573, 'gini = 0.0\nsamples = 129\nvalue
= [0, 205]\nclass = No'),
 Text(4333.980582524272, 776.5714285714287, 'EBE <= 0.375\ngini = 0.175\nsamp
les = 210\nvalue = [30, 280]\nclass = No'),
 Text(4290.64077669903, 465.9428571428573, 'gini = 0.494\nsamples = 5\nvalue
= [4, 5]\nclass = No'),
 Text(4377.320388349514, 465.9428571428573, 'PM10 <= 20.64\ngini = 0.158\nsam
ples = 205\nvalue = [26, 275]\nclass = No'),
 Text(4333.980582524272, 155.3142857142857, 'gini = 0.31\nsamples = 69\nvalue
= [18, 76]\nclass = No'),
 Text(4420.660194174758, 155.3142857142857, 'gini = 0.074\nsamples = 136\nval
ue = [8, 199]\nclass = No')]
```



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

```
Linear: 0.45865884208093644
Lasso: -0.0002593585325185721
Ridge: 0.4590393479321705
ElasticNet: 0.35747558851830485
Logistic: 0.4895
Random Forest: 0.7788255465066438
```

Best model is Random Forest

2011

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_2011.csv")
df
```

Out[2]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	
0	2011-11-01 01:00:00	NaN	1.0	NaN	NaN	154.0	84.0	NaN	NaN	NaN	6.0	NaN	NaN	2
1	2011-11-01 01:00:00	2.5	0.4	3.5	0.26	68.0	92.0	3.0	40.0	24.0	9.0	1.54	8.7	2
2	2011-11-01 01:00:00	2.9	NaN	3.8	NaN	96.0	99.0	NaN	NaN	NaN	NaN	NaN	7.2	2
3	2011-11-01 01:00:00	NaN	0.6	NaN	NaN	60.0	83.0	2.0	NaN	NaN	NaN	NaN	NaN	2
4	2011-11-01 01:00:00	NaN	NaN	NaN	NaN	44.0	62.0	3.0	NaN	NaN	3.0	NaN	NaN	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
209923	2011-09-01 00:00:00	NaN	0.2	NaN	NaN	5.0	19.0	44.0	NaN	NaN	NaN	NaN	NaN	2
209924	2011-09-01 00:00:00	NaN	0.1	NaN	NaN	6.0	29.0	NaN	11.0	NaN	7.0	NaN	NaN	2
209925	2011-09-01 00:00:00	NaN	NaN	NaN	0.23	1.0	21.0	28.0	NaN	NaN	NaN	1.44	NaN	2
209926	2011-09-01 00:00:00	NaN	NaN	NaN	NaN	3.0	15.0	48.0	NaN	NaN	NaN	NaN	NaN	2
209927	2011-09-01 00:00:00	NaN	NaN	NaN	NaN	4.0	33.0	38.0	13.0	NaN	NaN	NaN	NaN	2

209928 rows × 14 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209928 entries, 0 to 209927
Data columns (total 14 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     209928 non-null  object 
 1   BEN      51393 non-null   float64
 2   CO       87127 non-null   float64
 3   EBE      51350 non-null   float64
 4   NMHC     43517 non-null   float64
 5   NO       208954 non-null  float64
 6   NO_2     208973 non-null  float64
 7   O_3      122049 non-null  float64
 8   PM10     103743 non-null  float64
 9   PM25     51079 non-null   float64
 10  SO_2     87131 non-null   float64
 11  TCH      43519 non-null   float64
 12  TOL      51175 non-null   float64
 13  station  209928 non-null  int64  
dtypes: float64(12), int64(1), object(1)
memory usage: 22.4+ MB
```

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

Out[4]:

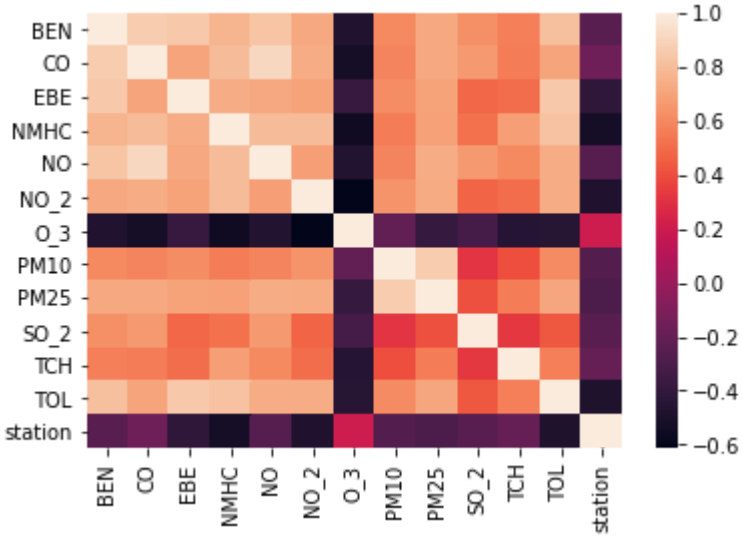
	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	s
1	2011-11-01 01:00:00	2.5	0.4	3.5	0.26	68.0	92.0	3.0	40.0	24.0	9.0	1.54	8.7	280
6	2011-11-01 01:00:00	0.7	0.3	1.1	0.16	17.0	66.0	7.0	22.0	16.0	2.0	1.36	1.7	280
25	2011-11-01 02:00:00	1.8	0.3	2.8	0.20	34.0	76.0	3.0	34.0	21.0	8.0	1.71	7.4	280
30	2011-11-01 02:00:00	1.0	0.4	1.3	0.18	31.0	67.0	5.0	25.0	18.0	3.0	1.40	2.9	280
49	2011-11-01 03:00:00	1.3	0.2	2.4	0.22	29.0	72.0	3.0	33.0	20.0	8.0	1.75	6.2	280
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
209862	2011-08-31 22:00:00	0.4	0.1	1.0	0.06	1.0	13.0	33.0	21.0	6.0	5.0	1.26	0.7	280
209881	2011-08-31 23:00:00	0.9	0.1	1.8	0.16	11.0	45.0	30.0	32.0	17.0	3.0	1.34	4.9	280
209886	2011-08-31 23:00:00	0.6	0.1	1.1	0.05	1.0	12.0	48.0	19.0	7.0	5.0	1.26	0.9	280
209905	2011-09-01 00:00:00	0.6	0.1	1.3	0.15	6.0	35.0	34.0	21.0	12.0	3.0	1.32	3.8	280
209910	2011-09-01 00:00:00	0.7	0.1	1.1	0.04	1.0	12.0	46.0	8.0	5.0	5.0	1.25	0.9	280

16460 rows × 14 columns

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

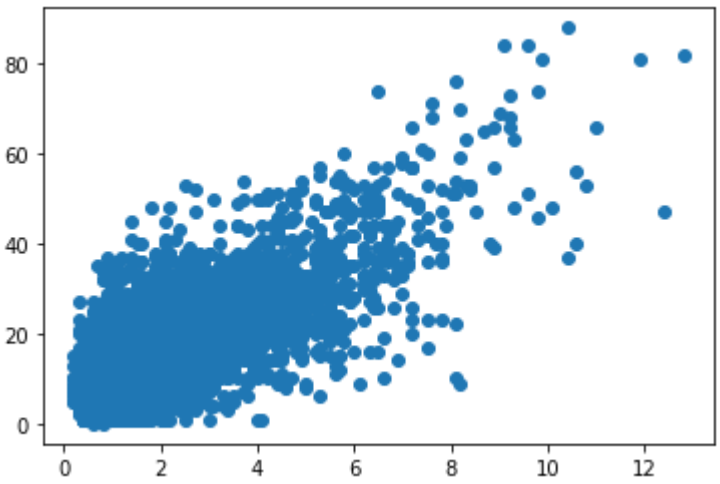
In [6]:
```
sns.heatmap(df1.corr())
```

Out[6]: <AxesSubplot:>



In [7]:
```
plt.plot(df1["EBE"],df1["PM25"],"o")
```

Out[7]: [<matplotlib.lines.Line2D at 0x1aa70adb160>]



In [8]:
```
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)
```

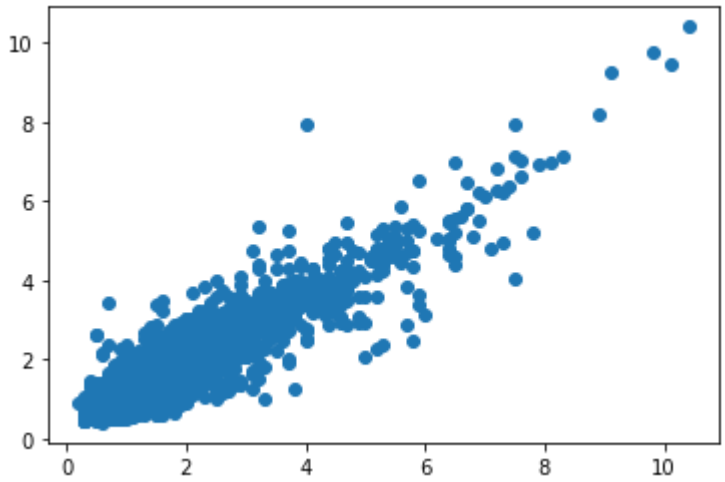
Linear

In [9]:
```
li=LinearRegression()
li.fit(x_train,y_train)
```

Out[9]: LinearRegression()

In [10]:
```
prediction=li.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[10]:
```
<matplotlib.collections.PathCollection at 0x1aa70cb4cd0>
```



In [11]:
```
lis=li.score(x_test,y_test)
```

In [12]:
```
df1["TCH"].value_counts()
```

Out[12]:
```
1.30    897
1.29    878
1.28    856
1.31    827
1.27    820
       ...
3.41      1
2.88      1
2.41      1
2.80      1
2.49      1
Name: TCH, Length: 171, dtype: int64
```

In [13]:
```
df1.loc[df1["TCH"]<1.40,"TCH"]=1
df1.loc[df1["TCH"]>1.40,"TCH"]=2
df1["TCH"].value_counts()
```

Out[13]:
```
1.0    12828
2.0     3632
Name: TCH, dtype: int64
```

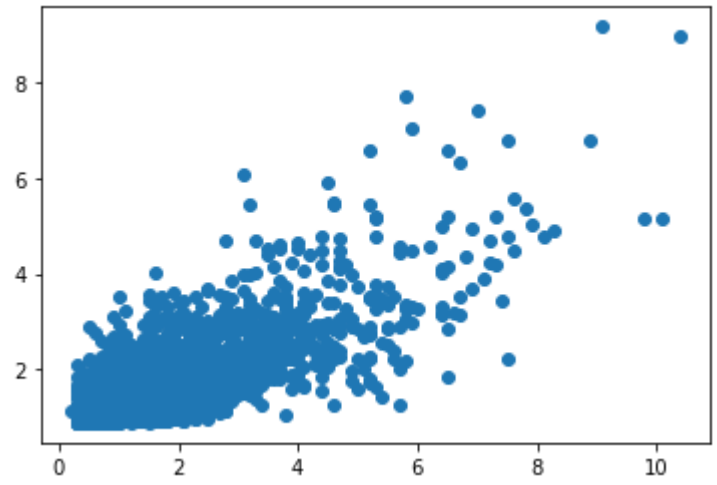
Lasso

In [14]:
```
la=Lasso(alpha=5)
la.fit(x_train,y_train)
```

Out[14]:
```
Lasso(alpha=5)
```

In [15]:
```
prediction1=la.predict(x_test)
plt.scatter(y_test,prediction1)
```

Out[15]: <matplotlib.collections.PathCollection at 0x1aa70d1ee80>



In [16]:
```
las=la.score(x_test,y_test)
```

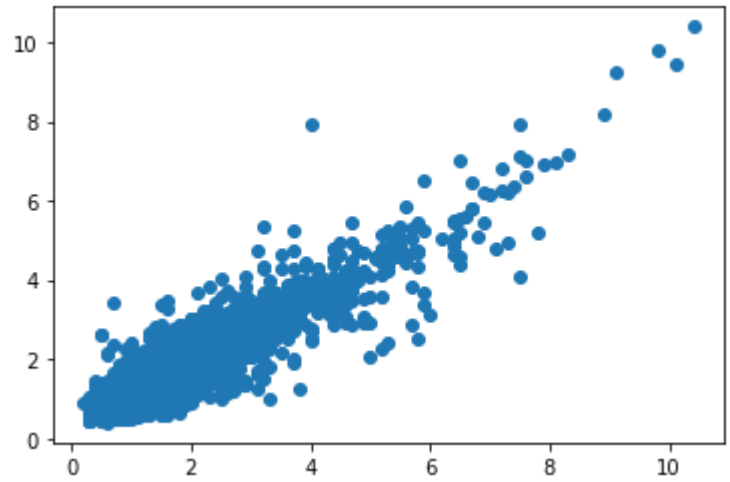
Ridge

In [17]:
```
rr=Ridge(alpha=1)
rr.fit(x_train,y_train)
```

Out[17]: Ridge(alpha=1)

In [18]:
```
prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[18]: <matplotlib.collections.PathCollection at 0x1aa70b271f0>



In [19]:
```
rrs=rr.score(x_test,y_test)
```

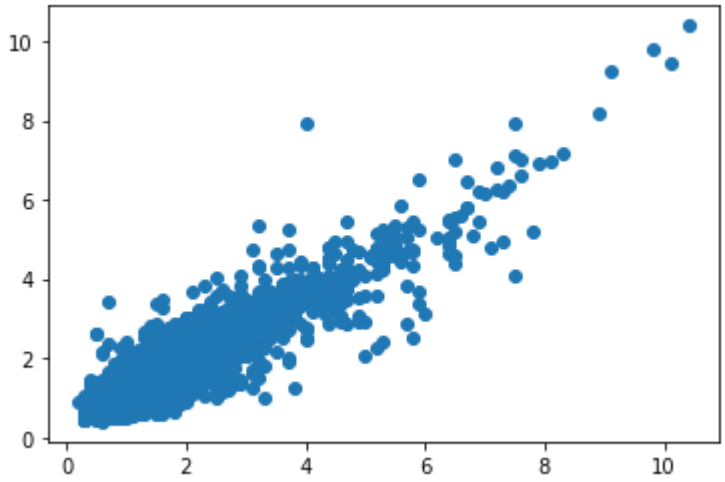
ElasticNet

In [20]:
```
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[20]:
```
ElasticNet()
```

In [21]:
```
prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[21]:
```
<matplotlib.collections.PathCollection at 0x1aa70d9ca30>
```



In [22]:
```
ens=en.score(x_test,y_test)
```

In [23]:
```
print(rr.score(x_test,y_test))
rr.score(x_train,y_train)
```

```
0.8305645679872002
```

Out[23]:
```
0.8132155812013233
```

Logistic

In [24]:
```
g={"TCH":{1.0:"Low",2.0:"High"}}
df1=df1.replace(g)
df1["TCH"].value_counts()
```

Out[24]:
```
Low     12828
High     3632
Name: TCH, dtype: int64
```

In [25]:
```
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 [26]:
```
lo=LogisticRegression()
lo.fit(x_train,y_train)
```

Out[26]:
```
LogisticRegression()
```

In [27]:
```
prediction3=lo.predict(x_test)
plt.scatter(y_test,prediction3)
```

Out[27]:
```
<matplotlib.collections.PathCollection at 0x1aa70b57cd0>
```

In [28]:
```
los=lo.score(x_test,y_test)
```

Random Forest

In [29]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
```

In [30]:
```
g1={"TCH":{"Low":1.0,"High":2.0}}
df1=df1.replace(g1)
```

In [31]:
```
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 [32]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[32]:
```
RandomForestClassifier()
```

In [33]:
```
parameter={
    'max_depth':[1,2,4,5,6],
    'min_samples_leaf':[5,10,15,20,25],
    'n_estimators':[10,20,30,40,50]
}
```

In [34]:
```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accu
grid_search.fit(x_train,y_train)
```

Out[34]:
```
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 [35]:
```
rfcs=grid_search.best_score_
```

In [36]:
```
rfc_best=grid_search.best_estimator_
```

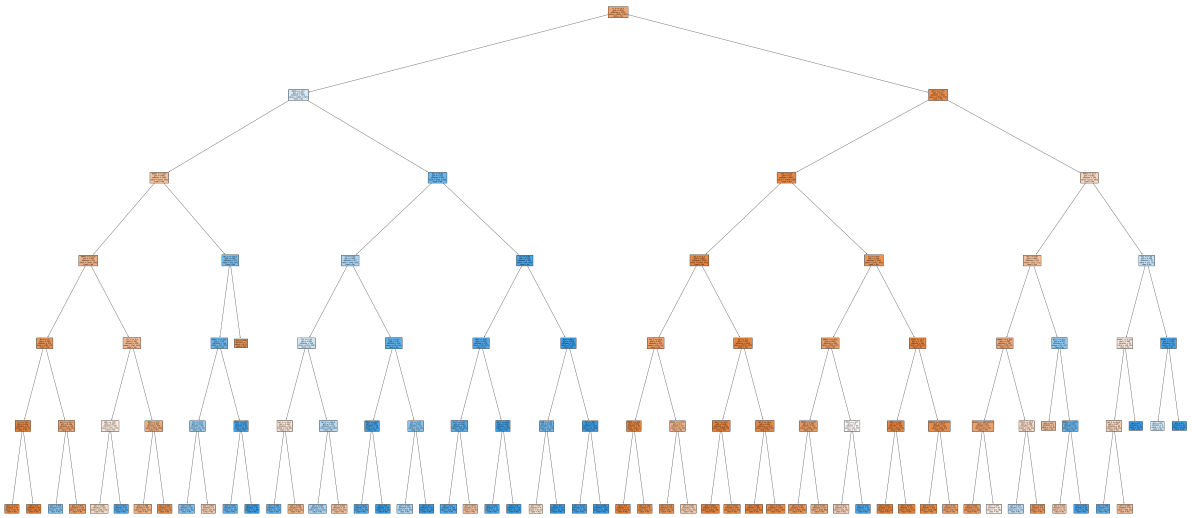
In [37]:
```
from sklearn.tree import plot_tree

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

Out[37]:
[Text(2316.467889908257, 2019.0857142857144, 'O_3 <= 19.5\ngini = 0.338\nsamples = 7259\nvalue = [9040, 2482]\nclass = Yes'),
 Text(1116.0, 1708.457142857143, 'PM25 <= 16.5\ngini = 0.494\nsamples = 1972\nvalue = [1394, 1745]\nclass = No'),
 Text(593.8348623853211, 1397.8285714285716, 'NMHC <= 0.235\ngini = 0.446\nsamples = 1040\nvalue = [1092, 553]\nclass = Yes'),
 Text(327.6330275229358, 1087.2, 'NMHC <= 0.155\ngini = 0.399\nsamples = 915\nvalue = [1043, 396]\nclass = Yes'),
 Text(163.8165137614679, 776.5714285714287, 'NO <= 5.5\ngini = 0.245\nsamples = 219\nvalue = [282, 47]\nclass = Yes'),
 Text(81.90825688073394, 465.9428571428573, 'O_3 <= 12.5\ngini = 0.044\nsamples = 91\nvalue = [130, 3]\nclass = Yes'),
 Text(40.95412844036697, 155.3142857142857, 'gini = 0.117\nsamples = 35\nvalue = [45, 3]\nclass = Yes'),
 Text(122.86238532110092, 155.3142857142857, 'gini = 0.0\nsamples = 56\nvalue = [85, 0]\nclass = Yes'),
 Text(245.72477064220183, 465.9428571428573, 'NO_2 <= 23.5\ngini = 0.348\nsamples = 128\nvalue = [152, 44]\nclass = Yes'),
 Text(204.77064220183485, 155.3142857142857, 'gini = 0.408\nsamples = 12\nvalue = [4, 10]\nclass = No'),
 Text(286.6788990825688, 155.3142857142857, 'gini = 0.304\nsamples = 116\nvalue = [148, 34]\nclass = Yes'),
 Text(491.44954128440367, 776.5714285714287, 'SO_2 <= 3.5\ngini = 0.431\nsamples = 696\nvalue = [761, 349]\nclass = Yes'),
 Text(409.5412844036697, 465.9428571428573, 'EBE <= 1.85\ngini = 0.495\nsamples = 198\nvalue = [171, 141]\nclass = Yes'),
 Text(368.58715596330273, 155.3142857142857, 'gini = 0.485\nsamples = 181\nvalue = [168, 119]\nclass = Yes'),
 Text(450.4954128440367, 155.3142857142857, 'gini = 0.211\nsamples = 17\nvalue = [3, 22]\nclass = No'),
 Text(573.3577981651376, 465.9428571428573, 'BEN <= 1.75\ngini = 0.385\nsamples = 498\nvalue = [590, 208]\nclass = Yes'),
 Text(532.4036697247707, 155.3142857142857, 'gini = 0.396\nsamples = 467\nvalue = [537, 201]\nclass = Yes'),
 Text(614.3119266055046, 155.3142857142857, 'gini = 0.206\nsamples = 31\nvalue = [53, 7]\nclass = Yes'),
 Text(860.0366972477065, 1087.2, 'NO_2 <= 102.0\ngini = 0.363\nsamples = 125\nvalue = [49, 157]\nclass = No'),
 Text(819.0825688073394, 776.5714285714287, 'NMHC <= 0.255\ngini = 0.33\nsamples = 118\nvalue = [41, 156]\nclass = No'),
 Text(737.1743119266055, 465.9428571428573, 'CO <= 0.55\ngini = 0.434\nsamples = 65\nvalue = [35, 75]\nclass = No'),
 Text(696.2201834862385, 155.3142857142857, 'gini = 0.39\nsamples = 56\nvalue = [25, 69]\nclass = No'),
 Text(778.1284403669724, 155.3142857142857, 'gini = 0.469\nsamples = 9\nvalue = [10, 6]\nclass = Yes'),
 Text(900.9908256880734, 465.9428571428573, 'PM10 <= 22.5\ngini = 0.128\nsamples = 53\nvalue = [6, 81]\nclass = No'),
 Text(860.0366972477065, 155.3142857142857, 'gini = 0.302\nsamples = 16\nvalue = [5, 22]\nclass = No'),
 Text(941.9449541284404, 155.3142857142857, 'gini = 0.033\nsamples = 37\nvalue = [1, 59]\nclass = No'),
 Text(900.9908256880734, 776.5714285714287, 'gini = 0.198\nsamples = 7\nvalue = [8, 1]\nclass = Yes'),
 Text(1638.1651376146788, 1397.8285714285716, 'TOL <= 6.85\ngini = 0.323\nsamples = 932\nvalue = [302, 1192]\nclass = No'),
 Text(1310.532110091743, 1087.2, 'CO <= 0.55\ngini = 0.464\nsamples = 407\nva

```
lue = [248, 429]\nclass = No'),
 Text(1146.7155963302753, 776.5714285714287, 'TOL <= 3.35\ngini = 0.491\nsamp
les = 298\nvalue = [219, 286]\nclass = No'),
 Text(1064.8073394495414, 465.9428571428573, 'NO_2 <= 55.5\ngini = 0.493\nsam
ples = 83\nvalue = [74, 58]\nclass = Yes'),
 Text(1023.8532110091743, 155.3142857142857, 'gini = 0.295\nsamples = 27\nval
ue = [7, 32]\nclass = No'),
 Text(1105.7614678899083, 155.3142857142857, 'gini = 0.403\nsamples = 56\nval
ue = [67, 26]\nclass = Yes'),
 Text(1228.6238532110092, 465.9428571428573, 'NO_2 <= 106.5\ngini = 0.475\nsa
mples = 215\nvalue = [145, 228]\nclass = No'),
 Text(1187.6697247706422, 155.3142857142857, 'gini = 0.454\nsamples = 191\nva
lue = [113, 212]\nclass = No'),
 Text(1269.5779816513761, 155.3142857142857, 'gini = 0.444\nsamples = 24\nval
ue = [32, 16]\nclass = Yes'),
 Text(1474.348623853211, 776.5714285714287, 'SO_2 <= 9.5\ngini = 0.28\nsample
s = 109\nvalue = [29, 143]\nclass = No'),
 Text(1392.440366972477, 465.9428571428573, 'PM10 <= 28.5\ngini = 0.092\nsamp
les = 48\nvalue = [4, 79]\nclass = No'),
 Text(1351.48623853211, 155.3142857142857, 'gini = 0.213\nsamples = 19\nvalue
= [4, 29]\nclass = No'),
 Text(1433.394495412844, 155.3142857142857, 'gini = 0.0\nsamples = 29\nvalue
= [0, 50]\nclass = No'),
 Text(1556.2568807339449, 465.9428571428573, 'CO <= 0.75\ngini = 0.404\nsampl
es = 61\nvalue = [25, 64]\nclass = No'),
 Text(1515.302752293578, 155.3142857142857, 'gini = 0.462\nsamples = 45\nvalu
e = [25, 44]\nclass = No'),
 Text(1597.2110091743118, 155.3142857142857, 'gini = 0.0\nsamples = 16\nvalue
= [0, 20]\nclass = No'),
 Text(1965.7981651376147, 1087.2, 'CO <= 0.65\ngini = 0.123\nsamples = 525\nv
alue = [54, 763]\nclass = No'),
 Text(1801.9816513761468, 776.5714285714287, 'EBE <= 3.85\ngini = 0.209\nsamp
les = 255\nvalue = [48, 356]\nclass = No'),
 Text(1720.073394495413, 465.9428571428573, 'SO_2 <= 16.5\ngini = 0.248\nsamp
les = 195\nvalue = [46, 271]\nclass = No'),
 Text(1679.119266055046, 155.3142857142857, 'gini = 0.21\nsamples = 185\nvalu
e = [36, 266]\nclass = No'),
 Text(1761.0275229357799, 155.3142857142857, 'gini = 0.444\nsamples = 10\nval
ue = [10, 5]\nclass = Yes'),
 Text(1883.8899082568807, 465.9428571428573, 'NO_2 <= 82.0\ngini = 0.045\nsam
ples = 60\nvalue = [2, 85]\nclass = No'),
 Text(1842.9357798165138, 155.3142857142857, 'gini = 0.153\nsamples = 17\nval
ue = [2, 22]\nclass = No'),
 Text(1924.8440366972477, 155.3142857142857, 'gini = 0.0\nsamples = 43\nvalue
= [0, 63]\nclass = No'),
 Text(2129.6146788990827, 776.5714285714287, 'NO <= 89.0\ngini = 0.029\nsampl
es = 270\nvalue = [6, 407]\nclass = No'),
 Text(2047.7064220183486, 465.9428571428573, 'BEN <= 2.55\ngini = 0.346\nsamp
les = 16\nvalue = [4, 14]\nclass = No'),
 Text(2006.7522935779816, 155.3142857142857, 'gini = 0.49\nsamples = 6\nvalue
= [4, 3]\nclass = Yes'),
 Text(2088.6605504587155, 155.3142857142857, 'gini = 0.0\nsamples = 10\nvalue
= [0, 11]\nclass = No'),
 Text(2211.5229357798166, 465.9428571428573, 'NO <= 98.5\ngini = 0.01\nsample
s = 254\nvalue = [2, 393]\nclass = No'),
 Text(2170.5688073394494, 155.3142857142857, 'gini = 0.142\nsamples = 11\nval
ue = [1, 12]\nclass = No'),
```

```
 Text(2252.4770642201834, 155.3142857142857, 'gini = 0.005\nsamples = 243\nva
lue = [1, 381]\nclass = No'),
 Text(3516.935779816514, 1708.457142857143, 'NO_2 <= 77.5\ngini = 0.16\nsampl
es = 5287\nvalue = [7646, 737]\nclass = Yes'),
 Text(2948.697247706422, 1397.8285714285716, 'TOL <= 3.15\ngini = 0.13\nsampl
es = 4997\nvalue = [7374, 553]\nclass = Yes'),
 Text(2621.064220183486, 1087.2, 'SO_2 <= 1.5\ngini = 0.083\nsamples = 3554\n
value = [5402, 245]\nclass = Yes'),
 Text(2457.2477064220184, 776.5714285714287, 'NO_2 <= 23.5\ngini = 0.302\nsam
ples = 110\nvalue = [132, 30]\nclass = Yes'),
 Text(2375.3394495412845, 465.9428571428573, 'PM25 <= 10.0\ngini = 0.034\nsam
ples = 40\nvalue = [56, 1]\nclass = Yes'),
 Text(2334.3853211009173, 155.3142857142857, 'gini = 0.0\nsamples = 31\nvalue
= [45, 0]\nclass = Yes'),
 Text(2416.293577981651, 155.3142857142857, 'gini = 0.153\nsamples = 9\nvalue
= [11, 1]\nclass = Yes'),
 Text(2539.1559633027523, 465.9428571428573, 'NMHC <= 0.155\ngini = 0.4\nsamp
les = 70\nvalue = [76, 29]\nclass = Yes'),
 Text(2498.201834862385, 155.3142857142857, 'gini = 0.219\nsamples = 28\nvalu
e = [35, 5]\nclass = Yes'),
 Text(2580.110091743119, 155.3142857142857, 'gini = 0.466\nsamples = 42\nvalu
e = [41, 24]\nclass = Yes'),
 Text(2784.880733944954, 776.5714285714287, 'NO <= 1.5\ngini = 0.075\nsamples
= 3444\nvalue = [5270, 215]\nclass = Yes'),
 Text(2702.97247706422, 465.9428571428573, 'NO_2 <= 32.5\ngini = 0.036\nsampl
es = 1323\nvalue = [2086, 39]\nclass = Yes'),
 Text(2662.0183486238534, 155.3142857142857, 'gini = 0.021\nsamples = 1174\nv
alue = [1871, 20]\nclass = Yes'),
 Text(2743.9266055045873, 155.3142857142857, 'gini = 0.149\nsamples = 149\nva
lue = [215, 19]\nclass = Yes'),
 Text(2866.788990825688, 465.9428571428573, 'NO_2 <= 19.5\ngini = 0.099\nsamp
les = 2121\nvalue = [3184, 176]\nclass = Yes'),
 Text(2825.8348623853212, 155.3142857142857, 'gini = 0.048\nsamples = 822\nva
lue = [1277, 32]\nclass = Yes'),
 Text(2907.743119266055, 155.3142857142857, 'gini = 0.131\nsamples = 1299\nva
lue = [1907, 144]\nclass = Yes'),
 Text(3276.3302752293575, 1087.2, 'O_3 <= 47.5\ngini = 0.234\nsamples = 1443
\nvalue = [1972, 308]\nclass = Yes'),
 Text(3112.5137614678897, 776.5714285714287, 'NMHC <= 0.225\ngini = 0.318\nsa
mples = 780\nvalue = [998, 247]\nclass = Yes'),
 Text(3030.605504587156, 465.9428571428573, 'NMHC <= 0.195\ngini = 0.285\nsam
ples = 714\nvalue = [946, 197]\nclass = Yes'),
 Text(2989.651376146789, 155.3142857142857, 'gini = 0.236\nsamples = 501\nval
ue = [684, 108]\nclass = Yes'),
 Text(3071.559633027523, 155.3142857142857, 'gini = 0.379\nsamples = 213\nval
ue = [262, 89]\nclass = Yes'),
 Text(3194.4220183486236, 465.9428571428573, 'PM25 <= 18.5\ngini = 0.5\nsampl
es = 66\nvalue = [52, 50]\nclass = Yes'),
 Text(3153.467889908257, 155.3142857142857, 'gini = 0.472\nsamples = 53\nvalu
e = [50, 31]\nclass = Yes'),
 Text(3235.376146788991, 155.3142857142857, 'gini = 0.172\nsamples = 13\nvalu
e = [2, 19]\nclass = No'),
 Text(3440.146788990826, 776.5714285714287, 'PM25 <= 12.5\ngini = 0.111\nsamp
les = 663\nvalue = [974, 61]\nclass = Yes'),
 Text(3358.238532110092, 465.9428571428573, 'BEN <= 0.45\ngini = 0.055\nsampl
es = 339\nvalue = [520, 15]\nclass = Yes'),
 Text(3317.2844036697247, 155.3142857142857, 'gini = 0.021\nsamples = 127\nva
```

```
lue = [191, 2]\nclass = Yes'),
 Text(3399.1926605504586, 155.3142857142857, 'gini = 0.073\nsamples = 212\nva
lue = [329, 13]\nclass = Yes'),
 Text(3522.0550458715597, 465.9428571428573, 'station <= 28079016.0\ngini =
0.167\nsamples = 324\nvalue = [454, 46]\nclass = Yes'),
 Text(3481.1009174311926, 155.3142857142857, 'gini = 0.147\nsamples = 293\nva
lue = [416, 36]\nclass = Yes'),
 Text(3563.0091743119265, 155.3142857142857, 'gini = 0.33\nsamples = 31\nvalu
e = [38, 10]\nclass = Yes'),
 Text(4085.1743119266052, 1397.8285714285716, 'PM25 <= 21.5\ngini = 0.481\nsa
mples = 290\nvalue = [272, 184]\nclass = Yes'),
 Text(3870.1651376146788, 1087.2, 'TOL <= 6.65\ngini = 0.445\nsamples = 214\n
value = [221, 111]\nclass = Yes'),
 Text(3767.7798165137615, 776.5714285714287, 'NMHC <= 0.235\ngini = 0.367\nsa
mples = 167\nvalue = [200, 64]\nclass = Yes'),
 Text(3685.8715596330276, 465.9428571428573, 'station <= 28079016.0\ngini =
0.324\nsamples = 137\nvalue = [172, 44]\nclass = Yes'),
 Text(3644.9174311926604, 155.3142857142857, 'gini = 0.251\nsamples = 111\nva
lue = [151, 26]\nclass = Yes'),
 Text(3726.8256880733943, 155.3142857142857, 'gini = 0.497\nsamples = 26\nval
ue = [21, 18]\nclass = Yes'),
 Text(3849.6880733944954, 465.9428571428573, 'PM10 <= 35.0\ngini = 0.486\nsam
ples = 30\nvalue = [28, 20]\nclass = Yes'),
 Text(3808.733944954128, 155.3142857142857, 'gini = 0.477\nsamples = 21\nvalu
e = [11, 17]\nclass = No'),
 Text(3890.642201834862, 155.3142857142857, 'gini = 0.255\nsamples = 9\nvalue
= [17, 3]\nclass = Yes'),
 Text(3972.5504587155965, 776.5714285714287, 'NO_2 <= 81.5\ngini = 0.427\nsam
ples = 47\nvalue = [21, 47]\nclass = No'),
 Text(3931.5963302752293, 465.9428571428573, 'gini = 0.43\nsamples = 9\nvalue
= [11, 5]\nclass = Yes'),
 Text(4013.5045871559632, 465.9428571428573, 'NMHC <= 0.235\ngini = 0.311\nsa
mples = 38\nvalue = [10, 42]\nclass = No'),
 Text(3972.5504587155965, 155.3142857142857, 'gini = 0.463\nsamples = 10\nval
ue = [7, 4]\nclass = Yes'),
 Text(4054.4587155963304, 155.3142857142857, 'gini = 0.136\nsamples = 28\nval
ue = [3, 38]\nclass = No'),
 Text(4300.183486238532, 1087.2, 'TOL <= 7.05\ngini = 0.484\nsamples = 76\nva
lue = [51, 73]\nclass = No'),
 Text(4218.275229357798, 776.5714285714287, 'NMHC <= 0.255\ngini = 0.496\nsam
ples = 54\nvalue = [48, 40]\nclass = Yes'),
 Text(4177.321100917431, 465.9428571428573, 'NO_2 <= 80.5\ngini = 0.477\nsamp
les = 48\nvalue = [48, 31]\nclass = Yes'),
 Text(4136.366972477064, 155.3142857142857, 'gini = 0.298\nsamples = 5\nvalue
= [2, 9]\nclass = No'),
 Text(4218.275229357798, 155.3142857142857, 'gini = 0.438\nsamples = 43\nvalu
e = [46, 22]\nclass = Yes'),
 Text(4259.229357798165, 465.9428571428573, 'gini = 0.0\nsamples = 6\nvalue =
[0, 9]\nclass = No'),
 Text(4382.091743119266, 776.5714285714287, 'NMHC <= 0.215\ngini = 0.153\nsam
ples = 22\nvalue = [3, 33]\nclass = No'),
 Text(4341.137614678899, 465.9428571428573, 'gini = 0.49\nsamples = 5\nvalue
= [3, 4]\nclass = No'),
 Text(4423.045871559633, 465.9428571428573, 'gini = 0.0\nsamples = 17\nvalue
= [0, 29]\nclass = No')]
```



In [38]:
```
print("Linear:",lis)
print("Lasso:",las)
print("Ridge:",rrs)
print("ElasticNet:",ens)
print("Logistic:",los)
print("Random Forest:",rfcs)
```

```
Linear: 0.8305073316874275
Lasso: 0.5846802065548421
Ridge: 0.8305645679872002
ElasticNet: 0.7133987542437089
Logistic: 0.7778452814904819
Random Forest: 0.8899496615170978
```

Best Model is Random Forest

2012

In [39]:
```
df2=pd.read_csv("madrid_2012.csv")
df2
```

Out[39]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	
0	2012-09-01 01:00:00	NaN	0.2	NaN	NaN	7.0	18.0	NaN	NaN	NaN	2.0	NaN	NaN	28
1	2012-09-01 01:00:00	0.3	0.3	0.7	NaN	3.0	18.0	55.0	10.0	9.0	1.0	NaN	2.4	28
2	2012-09-01 01:00:00	0.4	NaN	0.7	NaN	2.0	10.0	NaN	NaN	NaN	NaN	NaN	1.5	28
3	2012-09-01 01:00:00	NaN	0.2	NaN	NaN	1.0	6.0	50.0	NaN	NaN	NaN	NaN	NaN	28
4	2012-09-01 01:00:00	NaN	NaN	NaN	NaN	1.0	13.0	54.0	NaN	NaN	3.0	NaN	NaN	28
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
210715	2012-03-01 00:00:00	NaN	0.6	NaN	NaN	37.0	84.0	14.0	NaN	NaN	NaN	NaN	NaN	28
210716	2012-03-01 00:00:00	NaN	0.4	NaN	NaN	5.0	76.0	NaN	17.0	NaN	7.0	NaN	NaN	28
210717	2012-03-01 00:00:00	NaN	NaN	NaN	0.34	3.0	41.0	24.0	NaN	NaN	NaN	1.34	NaN	28
210718	2012-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	44.0	36.0	NaN	NaN	NaN	NaN	NaN	28
210719	2012-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	56.0	40.0	18.0	NaN	NaN	NaN	NaN	28

210720 rows × 14 columns

In [40]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210720 entries, 0 to 210719
Data columns (total 14 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     210720 non-null  object 
 1   BEN      51511 non-null   float64
 2   CO       87097 non-null   float64
 3   EBE      51482 non-null   float64
 4   NMHC     30736 non-null   float64
 5   NO       209871 non-null  float64
 6   NO_2     209872 non-null  float64
 7   O_3      122339 non-null  float64
 8   PM10     104838 non-null  float64
 9   PM25     52164 non-null   float64
 10  SO_2     87333 non-null   float64
 11  TCH      30736 non-null   float64
 12  TOL      51373 non-null   float64
 13  station  210720 non-null  int64  
dtypes: float64(12), int64(1), object(1)
memory usage: 22.5+ MB
```

In [41]:
```
df3=df2.dropna()
df3
```

Out[41]:

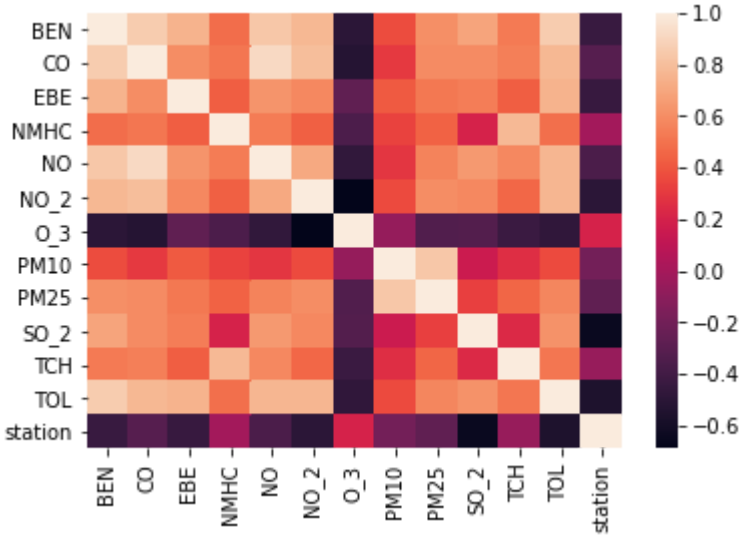
	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	s
6	2012-09-01 01:00:00	0.4	0.2	0.8	0.24	1.0	7.0	57.0	11.0	7.0	2.0	1.33	0.6	280
30	2012-09-01 02:00:00	0.4	0.2	0.7	0.24	1.0	5.0	55.0	5.0	5.0	2.0	1.33	0.5	280
54	2012-09-01 03:00:00	0.4	0.2	0.7	0.24	1.0	4.0	56.0	6.0	4.0	2.0	1.33	0.5	280
78	2012-09-01 04:00:00	0.3	0.2	0.7	0.25	1.0	5.0	54.0	6.0	5.0	2.0	1.34	0.4	280
102	2012-09-01 05:00:00	0.4	0.2	0.7	0.24	1.0	3.0	53.0	8.0	5.0	2.0	1.33	0.5	280
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
210654	2012-02-29 22:00:00	0.6	0.3	0.5	0.09	1.0	35.0	57.0	25.0	21.0	3.0	1.12	2.3	280
210673	2012-02-29 23:00:00	2.0	0.4	2.4	0.21	16.0	79.0	20.0	37.0	25.0	12.0	1.33	6.2	280
210678	2012-02-29 23:00:00	0.7	0.3	0.6	0.09	1.0	27.0	63.0	22.0	18.0	3.0	1.11	1.9	280
210697	2012-03-01 00:00:00	1.5	0.4	1.7	0.21	16.0	79.0	17.0	28.0	21.0	11.0	1.34	4.9	280
210702	2012-03-01 00:00:00	0.6	0.3	0.5	0.09	1.0	23.0	61.0	18.0	16.0	3.0	1.11	1.2	280

10916 rows × 14 columns

In [42]:
```
df3=df3.drop(["date"],axis=1)
```

In [43]:
```
sns.heatmap(df3.corr())
```

Out[43]: <AxesSubplot:>



In [44]:
```
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

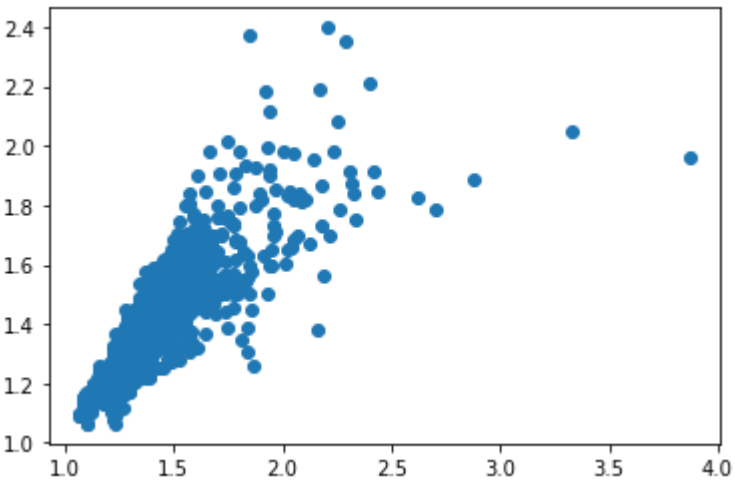
In [45]:
```
li=LinearRegression()
li.fit(x_train,y_train)
```

Out[45]: LinearRegression()

In [ ]:
```

```

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

Out[46]: <matplotlib.collections.PathCollection at 0x1aa72e176a0>



In [47]:
```
lis=li.score(x_test,y_test)
```

In [48]:
```
df3["TCH"].value_counts()
```

Out[48]:
```
1.30    737
1.31    676
1.32    644
1.33    552
1.29    529
       ...
3.03      1
3.01      1
2.47      1
2.33      1
2.07      1
Name: TCH, Length: 167, dtype: int64
```

In [49]:
```
df3.loc[df3["TCH"]<1.40,"TCH"]=1
df3.loc[df3["TCH"]>1.40,"TCH"]=2
df3["TCH"].value_counts()
```

Out[49]:
```
1.0    8772
2.0    2144
Name: TCH, dtype: int64
```

In [ ]:
```

```

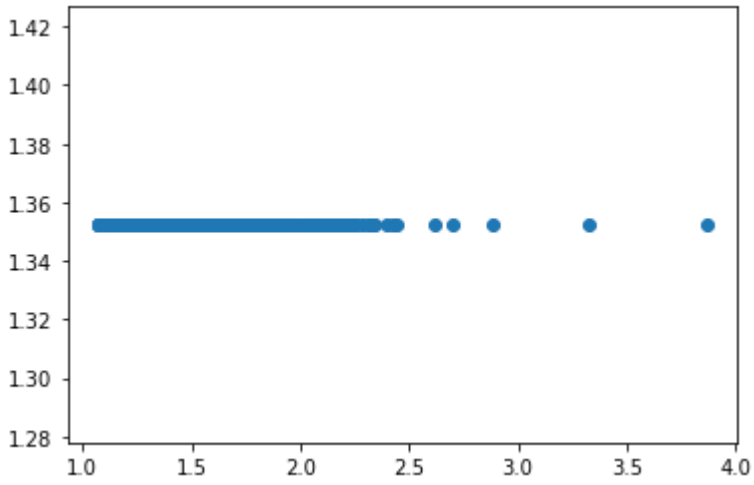
Lasso

In [50]:
```
la=Lasso(alpha=5)
la.fit(x_train,y_train)
```

Out[50]:
```
Lasso(alpha=5)
```

In [51]:
```
prediction1=la.predict(x_test)
plt.scatter(y_test,prediction1)
```

Out[51]: <matplotlib.collections.PathCollection at 0x1aa71741790>



In [52]:
```
las=la.score(x_test,y_test)
```

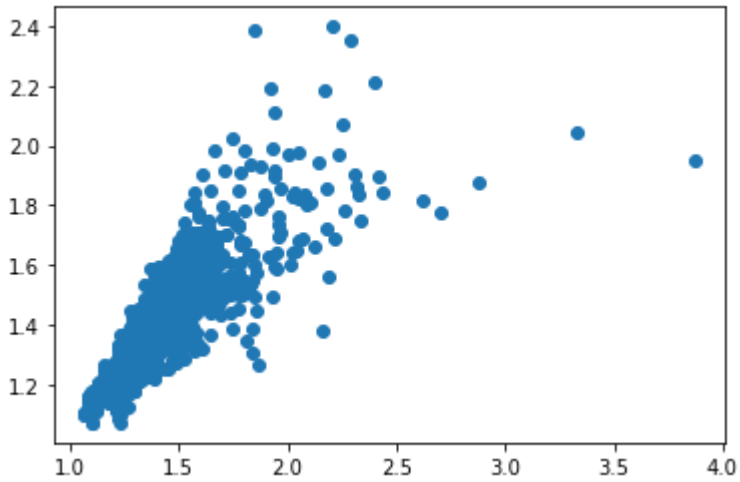
Ridge

In [53]:
```
rr=Ridge(alpha=1)
rr.fit(x_train,y_train)
```

Out[53]: Ridge(alpha=1)

In [54]:
```
prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[54]: <matplotlib.collections.PathCollection at 0x1aa717a3310>



In [55]:
```
rrs=rr.score(x_test,y_test)
```

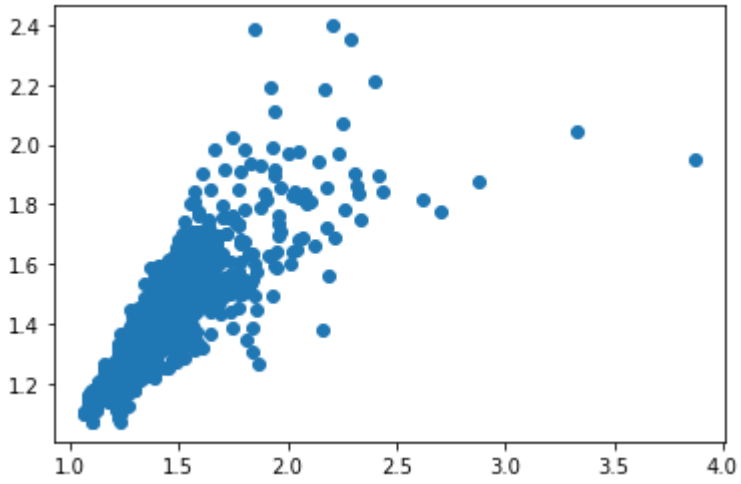
ElasticNet

In [56]:
```
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[56]:
```
ElasticNet()
```

In [57]:
```
prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[57]:
```
<matplotlib.collections.PathCollection at 0x1aa717f88e0>
```



In [58]:
```
ens=en.score(x_test,y_test)
```

In [59]:
```
print(rr.score(x_test,y_test))
rr.score(x_train,y_train)
```

```
0.6854937601485451
```

Out[59]:
```
0.6889077427743225
```

Logistic

In [60]:
```
g={"TCH":{1.0:"Low",2.0:"High"}}
df3=df3.replace(g)
df3["TCH"].value_counts()
```

Out[60]:
```
Low     8772
High    2144
Name: TCH, dtype: int64
```

In [61]:
```
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 [62]:
```
lo=LogisticRegression()
lo.fit(x_train,y_train)
```

Out[62]:
```
LogisticRegression()
```

In [63]:
```
prediction3=lo.predict(x_test)
plt.scatter(y_test,prediction3)
```

Out[63]:
```
<matplotlib.collections.PathCollection at 0x1aa716bc580>
```

In [64]:
```
los=lo.score(x_test,y_test)
```

Random Forest

In [65]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
```

In [66]:
```
g1={"TCH":{"Low":1.0,"High":2.0}}
df3=df3.replace(g1)
```

In [67]:
```
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 [68]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[68]:
```
RandomForestClassifier()
```

In [69]:
```
parameter={
    'max_depth':[1,2,4,5,6],
    'min_samples_leaf':[5,10,15,20,25],
    'n_estimators':[10,20,30,40,50]
}
```

In [70]:
```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accu
grid_search.fit(x_train,y_train)
```

Out[70]:
```
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 [71]:
```
rfcs=grid_search.best_score_
```

In [72]:
```
rfc_best=grid_search.best_estimator_
```

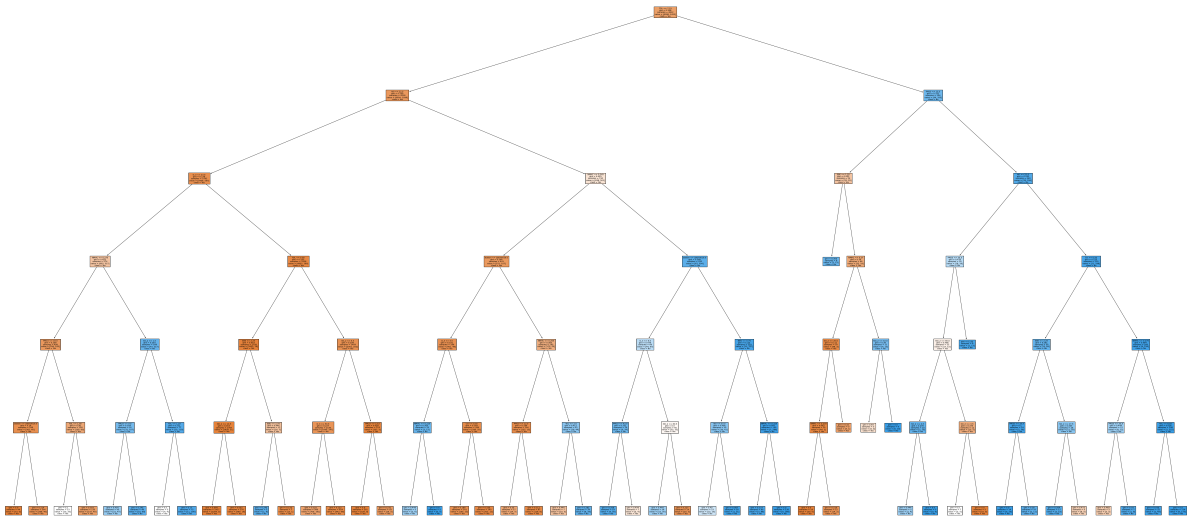
In [73]:
```
from sklearn.tree import plot_tree

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

Out[73]:
```
[Text(2493.5625, 2019.0857142857144, 'TOL <= 7.15\ngini = 0.308\nsamples = 4812\nvalue = [6188, 1453]\nclass = Yes'),
 Text(1488.0, 1708.457142857143, 'NO <= 22.5\ngini = 0.259\nsamples = 4518\nvalue = [6094, 1098]\nclass = Yes'),
 Text(744.0, 1397.8285714285716, 'O_3 <= 23.5\ngini = 0.178\nsamples = 3782\nvalue = [5456, 597]\nclass = Yes'),
 Text(372.0, 1087.2, 'NMHC <= 0.275\ngini = 0.452\nsamples = 574\nvalue = [603, 317]\nclass = Yes'),
 Text(186.0, 776.5714285714287, 'PM10 <= 16.5\ngini = 0.258\nsamples = 406\nvalue = [556, 100]\nclass = Yes'),
 Text(93.0, 465.9428571428573, 'station <= 28079016.0\ngini = 0.18\nsamples = 248\nvalue = [361, 40]\nclass = Yes'),
 Text(46.5, 155.3142857142857, 'gini = 0.0\nsamples = 44\nvalue = [73, 0]\nclass = Yes'),
 Text(139.5, 155.3142857142857, 'gini = 0.214\nsamples = 204\nvalue = [288, 40]\nclass = Yes'),
 Text(279.0, 465.9428571428573, 'TOL <= 1.05\ngini = 0.36\nsamples = 158\nvalue = [195, 60]\nclass = Yes'),
 Text(232.5, 155.3142857142857, 'gini = 0.5\nsamples = 23\nvalue = [20, 20]\nclass = Yes'),
 Text(325.5, 155.3142857142857, 'gini = 0.303\nsamples = 135\nvalue = [175, 40]\nclass = Yes'),
 Text(558.0, 776.5714285714287, 'SO_2 <= 1.5\ngini = 0.293\nsamples = 168\nvalue = [47, 217]\nclass = No'),
 Text(465.0, 465.9428571428573, 'PM25 <= 9.5\ngini = 0.354\nsamples = 91\nvalue = [32, 107]\nclass = No'),
 Text(418.5, 155.3142857142857, 'gini = 0.434\nsamples = 45\nvalue = [22, 47]\nclass = No'),
 Text(511.5, 155.3142857142857, 'gini = 0.245\nsamples = 46\nvalue = [10, 60]\nclass = No'),
 Text(651.0, 465.9428571428573, 'TOL <= 1.25\ngini = 0.211\nsamples = 77\nvalue = [15, 110]\nclass = No'),
 Text(604.5, 155.3142857142857, 'gini = 0.5\nsamples = 5\nvalue = [4, 4]\nclass = Yes'),
 Text(697.5, 155.3142857142857, 'gini = 0.17\nsamples = 72\nvalue = [11, 106]\nclass = No'),
 Text(1116.0, 1087.2, 'TOL <= 0.95\ngini = 0.103\nsamples = 3208\nvalue = [4853, 280]\nclass = Yes'),
 Text(930.0, 776.5714285714287, 'EBE <= 2.25\ngini = 0.019\nsamples = 1799\nvalue = [2852, 28]\nclass = Yes'),
 Text(837.0, 465.9428571428573, 'NO_2 <= 13.5\ngini = 0.015\nsamples = 1787\nvalue = [2838, 21]\nclass = Yes'),
 Text(790.5, 155.3142857142857, 'gini = 0.004\nsamples = 1417\nvalue = [2269, 5]\nclass = Yes'),
 Text(883.5, 155.3142857142857, 'gini = 0.053\nsamples = 370\nvalue = [569, 16]\nclass = Yes'),
 Text(1023.0, 465.9428571428573, 'EBE <= 2.75\ngini = 0.444\nsamples = 12\nvalue = [14, 7]\nclass = Yes'),
 Text(976.5, 155.3142857142857, 'gini = 0.245\nsamples = 5\nvalue = [1, 6]\nclass = No'),
 Text(1069.5, 155.3142857142857, 'gini = 0.133\nsamples = 7\nvalue = [13, 1]\nclass = Yes'),
 Text(1302.0, 776.5714285714287, 'SO_2 <= 5.5\ngini = 0.199\nsamples = 1409\nvalue = [2001, 252]\nclass = Yes'),
 Text(1209.0, 465.9428571428573, 'O_3 <= 50.5\ngini = 0.227\nsamples = 1190\nvalue = [1649, 247]\nclass = Yes'),
 Text(1162.5, 155.3142857142857, 'gini = 0.325\nsamples = 566\nvalue = [706,
```

```
181]\nclass = Yes'),
 Text(1255.5, 155.3142857142857, 'gini = 0.122\nsamples = 624\nvalue = [943,
66]\nclass = Yes'),
 Text(1395.0, 465.9428571428573, 'PM25 <= 13.5\ngini = 0.028\nsamples = 219\n
value = [352, 5]\nclass = Yes'),
 Text(1348.5, 155.3142857142857, 'gini = 0.0\nsamples = 201\nvalue = [326, 0]
\nclass = Yes'),
 Text(1441.5, 155.3142857142857, 'gini = 0.271\nsamples = 18\nvalue = [26, 5]
\nclass = Yes'),
 Text(2232.0, 1397.8285714285716, 'NMHC <= 0.245\ngini = 0.493\nsamples = 736
\nvalue = [638, 501]\nclass = Yes'),
 Text(1860.0, 1087.2, 'station <= 28079016.0\ngini = 0.266\nsamples = 441\nva
lue = [571, 107]\nclass = Yes'),
 Text(1674.0, 776.5714285714287, 'O_3 <= 4.5\ngini = 0.191\nsamples = 296\nva
lue = [401, 48]\nclass = Yes'),
 Text(1581.0, 465.9428571428573, 'NMHC <= 0.215\ngini = 0.305\nsamples = 10\n
value = [3, 13]\nclass = No'),
 Text(1534.5, 155.3142857142857, 'gini = 0.444\nsamples = 5\nvalue = [3, 6]\n
class = No'),
 Text(1627.5, 155.3142857142857, 'gini = 0.0\nsamples = 5\nvalue = [0, 7]\ncl
ass = No'),
 Text(1767.0, 465.9428571428573, 'CO <= 0.35\ngini = 0.149\nsamples = 286\nva
lue = [398, 35]\nclass = Yes'),
 Text(1720.5, 155.3142857142857, 'gini = 0.189\nsamples = 141\nvalue = [195,
23]\nclass = Yes'),
 Text(1813.5, 155.3142857142857, 'gini = 0.105\nsamples = 145\nvalue = [203,
12]\nclass = Yes'),
 Text(2046.0, 776.5714285714287, 'NMHC <= 0.205\ngini = 0.383\nsamples = 145
\nvalue = [170, 59]\nclass = Yes'),
 Text(1953.0, 465.9428571428573, 'PM10 <= 12.5\ngini = 0.164\nsamples = 104\n
value = [152, 15]\nclass = Yes'),
 Text(1906.5, 155.3142857142857, 'gini = 0.35\nsamples = 17\nvalue = [24, 7]
\nclass = Yes'),
 Text(1999.5, 155.3142857142857, 'gini = 0.111\nsamples = 87\nvalue = [128,
8]\nclass = Yes'),
 Text(2139.0, 465.9428571428573, 'TOL <= 2.0\ngini = 0.412\nsamples = 41\nval
ue = [18, 44]\nclass = No'),
 Text(2092.5, 155.3142857142857, 'gini = 0.444\nsamples = 7\nvalue = [10, 5]
\nclass = Yes'),
 Text(2185.5, 155.3142857142857, 'gini = 0.282\nsamples = 34\nvalue = [8, 39]
\nclass = No'),
 Text(2604.0, 1087.2, 'station <= 28079016.0\ngini = 0.248\nsamples = 295\nva
lue = [67, 394]\nclass = No'),
 Text(2418.0, 776.5714285714287, 'O_3 <= 6.5\ngini = 0.479\nsamples = 66\nval
ue = [44, 67]\nclass = No'),
 Text(2325.0, 465.9428571428573, 'PM10 <= 43.0\ngini = 0.331\nsamples = 26\nv
alue = [9, 34]\nclass = No'),
 Text(2278.5, 155.3142857142857, 'gini = 0.208\nsamples = 19\nvalue = [4, 30]
\nclass = No'),
 Text(2371.5, 155.3142857142857, 'gini = 0.494\nsamples = 7\nvalue = [5, 4]\n
class = Yes'),
 Text(2511.0, 465.9428571428573, 'NO_2 <= 80.5\ngini = 0.5\nsamples = 40\nval
ue = [35, 33]\nclass = Yes'),
 Text(2464.5, 155.3142857142857, 'gini = 0.444\nsamples = 26\nvalue = [15, 3
0]\nclass = No'),
 Text(2557.5, 155.3142857142857, 'gini = 0.227\nsamples = 14\nvalue = [20, 3]
\nclass = Yes'),
```

```
 Text(2790.0, 776.5714285714287, 'BEN <= 0.75\ngini = 0.123\nsamples = 229\nv
alue = [23, 327]\nclass = No'),
 Text(2697.0, 465.9428571428573, 'NO <= 37.5\ngini = 0.404\nsamples = 39\nval
ue = [16, 41]\nclass = No'),
 Text(2650.5, 155.3142857142857, 'gini = 0.49\nsamples = 23\nvalue = [15, 20]
\nclass = No'),
 Text(2743.5, 155.3142857142857, 'gini = 0.087\nsamples = 16\nvalue = [1, 21]
\nclass = No'),
 Text(2883.0, 465.9428571428573, 'NMHC <= 0.315\ngini = 0.047\nsamples = 190
\nvalue = [7, 286]\nclass = No'),
 Text(2836.5, 155.3142857142857, 'gini = 0.137\nsamples = 60\nvalue = [7, 88]
\nclass = No'),
 Text(2929.5, 155.3142857142857, 'gini = 0.0\nsamples = 130\nvalue = [0, 198]
\nclass = No'),
 Text(3499.125, 1708.457142857143, 'PM25 <= 15.5\ngini = 0.331\nsamples = 294
\nvalue = [94, 355]\nclass = No'),
 Text(3162.0, 1397.8285714285716, 'EBE <= 1.35\ngini = 0.452\nsamples = 54\nv
alue = [55, 29]\nclass = Yes'),
 Text(3115.5, 1087.2, 'gini = 0.305\nsamples = 8\nvalue = [3, 13]\nclass = N
o'),
 Text(3208.5, 1087.2, 'NMHC <= 0.27\ngini = 0.36\nsamples = 46\nvalue = [52,
16]\nclass = Yes'),
 Text(3115.5, 776.5714285714287, 'SO_2 <= 20.0\ngini = 0.111\nsamples = 35\nv
alue = [48, 3]\nclass = Yes'),
 Text(3069.0, 465.9428571428573, 'NMHC <= 0.235\ngini = 0.083\nsamples = 30\n
value = [44, 2]\nclass = Yes'),
 Text(3022.5, 155.3142857142857, 'gini = 0.0\nsamples = 18\nvalue = [30, 0]\n
class = Yes'),
 Text(3115.5, 155.3142857142857, 'gini = 0.219\nsamples = 12\nvalue = [14, 2]
\nclass = Yes'),
 Text(3162.0, 465.9428571428573, 'gini = 0.32\nsamples = 5\nvalue = [4, 1]\nc
lass = Yes'),
 Text(3301.5, 776.5714285714287, 'NO_2 <= 52.0\ngini = 0.36\nsamples = 11\nva
lue = [4, 13]\nclass = No'),
 Text(3255.0, 465.9428571428573, 'gini = 0.49\nsamples = 5\nvalue = [4, 3]\nc
lass = Yes'),
 Text(3348.0, 465.9428571428573, 'gini = 0.0\nsamples = 6\nvalue = [0, 10]\nc
lass = No'),
 Text(3836.25, 1397.8285714285716, 'NO <= 51.0\ngini = 0.191\nsamples = 240\n
value = [39, 326]\nclass = No'),
 Text(3580.5, 1087.2, 'PM25 <= 35.5\ngini = 0.476\nsamples = 30\nvalue = [18,
28]\nclass = No'),
 Text(3534.0, 776.5714285714287, 'NO_2 <= 60.0\ngini = 0.498\nsamples = 22\nv
alue = [17, 15]\nclass = Yes'),
 Text(3441.0, 465.9428571428573, 'SO_2 <= 4.5\ngini = 0.278\nsamples = 10\nva
lue = [2, 10]\nclass = No'),
 Text(3394.5, 155.3142857142857, 'gini = 0.444\nsamples = 5\nvalue = [2, 4]\n
class = No'),
 Text(3487.5, 155.3142857142857, 'gini = 0.0\nsamples = 5\nvalue = [0, 6]\ncl
ass = No'),
 Text(3627.0, 465.9428571428573, 'SO_2 <= 3.5\ngini = 0.375\nsamples = 12\nva
lue = [15, 5]\nclass = Yes'),
 Text(3580.5, 155.3142857142857, 'gini = 0.5\nsamples = 7\nvalue = [5, 5]\ncl
ass = Yes'),
 Text(3673.5, 155.3142857142857, 'gini = 0.0\nsamples = 5\nvalue = [10, 0]\nc
lass = Yes'),
 Text(3627.0, 776.5714285714287, 'gini = 0.133\nsamples = 8\nvalue = [1, 13]
```

```
\nclass = No'),
 Text(4092.0, 1087.2, 'CO <= 0.55\ngini = 0.123\nsamples = 210\nvalue = [21,
298]\nclass = No'),
 Text(3906.0, 776.5714285714287, 'EBE <= 2.25\ngini = 0.281\nsamples = 50\nva
lue = [13, 64]\nclass = No'),
 Text(3813.0, 465.9428571428573, 'PM10 <= 33.5\ngini = 0.1\nsamples = 24\nval
ue = [2, 36]\nclass = No'),
 Text(3766.5, 155.3142857142857, 'gini = 0.0\nsamples = 14\nvalue = [0, 25]\n
class = No'),
 Text(3859.5, 155.3142857142857, 'gini = 0.26\nsamples = 10\nvalue = [2, 11]
\nclass = No'),
 Text(3999.0, 465.9428571428573, 'SO_2 <= 17.5\ngini = 0.405\nsamples = 26\nv
alue = [11, 28]\nclass = No'),
 Text(3952.5, 155.3142857142857, 'gini = 0.204\nsamples = 17\nvalue = [3, 23]
\nclass = No'),
 Text(4045.5, 155.3142857142857, 'gini = 0.473\nsamples = 9\nvalue = [8, 5]\n
class = Yes'),
 Text(4278.0, 776.5714285714287, 'NMHC <= 0.275\ngini = 0.064\nsamples = 160
\nvalue = [8, 234]\nclass = No'),
 Text(4185.0, 465.9428571428573, 'PM10 <= 33.5\ngini = 0.444\nsamples = 12\nv
alue = [6, 12]\nclass = No'),
 Text(4138.5, 155.3142857142857, 'gini = 0.469\nsamples = 5\nvalue = [5, 3]\n
class = Yes'),
 Text(4231.5, 155.3142857142857, 'gini = 0.18\nsamples = 7\nvalue = [1, 9]\nc
lass = No'),
 Text(4371.0, 465.9428571428573, 'TOL <= 8.55\ngini = 0.018\nsamples = 148\nv
alue = [2, 222]\nclass = No'),
 Text(4324.5, 155.3142857142857, 'gini = 0.075\nsamples = 32\nvalue = [2, 49]
\nclass = No'),
 Text(4417.5, 155.3142857142857, 'gini = 0.0\nsamples = 116\nvalue = [0, 173]
\nclass = No')]
```



In [74]:
```
print("Linear:",lis)
print("Lasso:",las)
print("Ridge:",rrs)
print("ElasticNet:",ens)
print("Logistic:",los)
print("Random Forest:",rfcs)
```

```
Linear: 0.6851032563985955
Lasso: -0.0014194828338582877
Ridge: 0.6854937601485451
ElasticNet: 0.34195759108298374
Logistic: 0.8006106870229007
Random Forest: 0.9331240896615699
```

Best model is Random Forest

2013

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_2013.csv")
df
```

Out[2]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	
0	2013-11-01 01:00:00	NaN	0.6	NaN	NaN	135.0	74.0	NaN	NaN	NaN	7.0	NaN	NaN	2
1	2013-11-01 01:00:00	1.5	0.5	1.3	NaN	71.0	83.0	2.0	23.0	16.0	12.0	NaN	8.3	2
2	2013-11-01 01:00:00	3.9	NaN	2.8	NaN	49.0	70.0	NaN	NaN	NaN	NaN	NaN	9.0	2
3	2013-11-01 01:00:00	NaN	0.5	NaN	NaN	82.0	87.0	3.0	NaN	NaN	NaN	NaN	NaN	2
4	2013-11-01 01:00:00	NaN	NaN	NaN	NaN	242.0	111.0	2.0	NaN	NaN	12.0	NaN	NaN	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
209875	2013-03-01 00:00:00	NaN	0.4	NaN	NaN	8.0	39.0	52.0	NaN	NaN	NaN	NaN	NaN	2
209876	2013-03-01 00:00:00	NaN	0.4	NaN	NaN	1.0	11.0	NaN	6.0	NaN	2.0	NaN	NaN	2
209877	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	4.0	75.0	NaN	NaN	NaN	NaN	NaN	2
209878	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	11.0	52.0	NaN	NaN	NaN	NaN	NaN	2
209879	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	1.0	10.0	75.0	3.0	NaN	NaN	NaN	NaN	2

209880 rows × 14 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209880 entries, 0 to 209879
Data columns (total 14 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     209880 non-null  object 
 1   BEN      50462 non-null   float64
 2   CO       87018 non-null   float64
 3   EBE      50463 non-null   float64
 4   NMHC     25935 non-null   float64
 5   NO       209108 non-null  float64
 6   NO_2     209108 non-null  float64
 7   O_3      121858 non-null  float64
 8   PM10     104339 non-null  float64
 9   PM25     51980 non-null   float64
 10  SO_2     86970 non-null   float64
 11  TCH      25935 non-null   float64
 12  TOL      50317 non-null   float64
 13  station  209880 non-null  int64  
dtypes: float64(12), int64(1), object(1)
memory usage: 22.4+ MB
```

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

Out[4]:

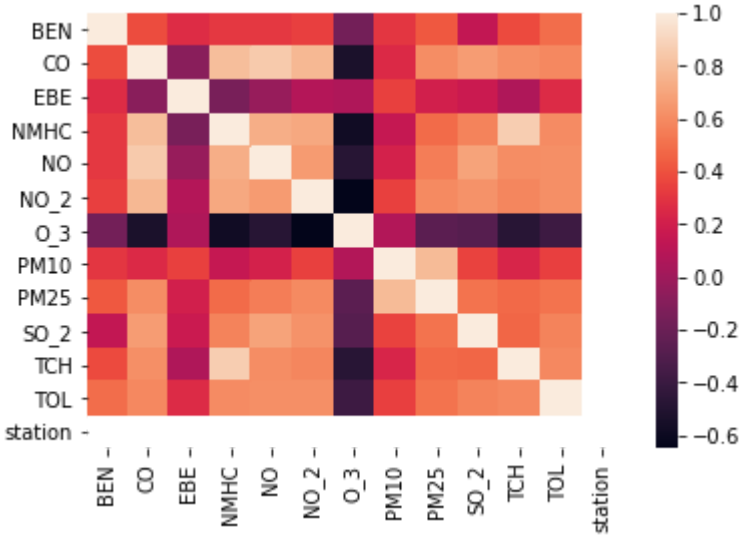
	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	st
17286	2013-08-01 01:00:00	0.4	0.2	0.8	0.28	1.0	24.0	79.0	35.0	8.0	3.0	1.49	1.3	2807
17310	2013-08-01 02:00:00	0.5	0.2	0.9	0.28	1.0	16.0	93.0	60.0	18.0	3.0	1.61	4.0	2807
17334	2013-08-01 03:00:00	0.5	0.2	1.1	0.29	1.0	14.0	90.0	38.0	12.0	3.0	1.71	2.8	2807
17358	2013-08-01 04:00:00	0.6	0.2	1.2	0.26	1.0	12.0	84.0	30.0	8.0	3.0	1.44	2.8	2807
17382	2013-08-01 05:00:00	0.3	0.2	0.8	0.25	1.0	15.0	72.0	25.0	7.0	3.0	1.40	1.7	2807
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
209622	2013-02-28 14:00:00	1.1	0.3	0.3	0.27	3.0	17.0	64.0	5.0	5.0	2.0	1.41	0.9	2807
209646	2013-02-28 15:00:00	1.3	0.4	0.3	0.27	2.0	16.0	66.0	6.0	5.0	1.0	1.40	0.9	2807
209670	2013-02-28 16:00:00	1.1	0.3	0.3	0.27	1.0	17.0	65.0	5.0	4.0	1.0	1.40	0.7	2807
209694	2013-02-28 17:00:00	1.0	0.3	0.4	0.27	1.0	18.0	64.0	5.0	5.0	1.0	1.39	0.7	2807
209718	2013-02-28 18:00:00	1.0	0.3	0.4	0.27	1.0	22.0	62.0	6.0	6.0	1.0	1.39	0.7	2807

7315 rows × 14 columns

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

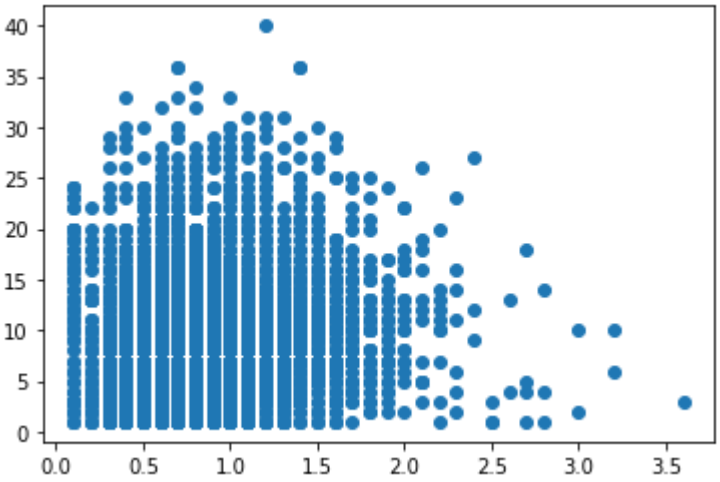
In [6]:
```
sns.heatmap(df1.corr())
```

Out[6]: <AxesSubplot:>



In [7]:
```
plt.plot(df1["EBE"],df1["PM25"],"o")
```

Out[7]: [<matplotlib.lines.Line2D at 0x25a0674f250>]



In [8]:
```
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)
```

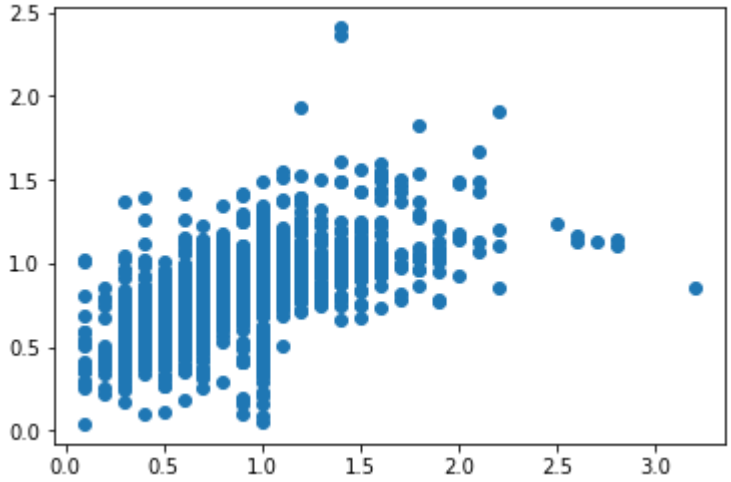
Linear

In [9]:
```
li=LinearRegression()
li.fit(x_train,y_train)
```

Out[9]: LinearRegression()

In [10]:
```
prediction=li.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[10]:
```
<matplotlib.collections.PathCollection at 0x25a06816790>
```



In [11]:
```
lis=li.score(x_test,y_test)
```

In [12]:
```
df1["TCH"].value_counts()
```

Out[12]:
```
1.32    888
1.33    843
1.34    729
1.31    719
1.35    556
       ...
1.23      1
2.09      1
1.84      1
2.25      1
2.29      1
Name: TCH, Length: 114, dtype: int64
```

In [13]:
```
df1.loc[df1["TCH"]<1.40,"TCH"]=1
df1.loc[df1["TCH"]>1.40,"TCH"]=2
df1["TCH"].value_counts()
```

Out[13]:
```
1.0    5718
2.0    1597
Name: TCH, dtype: int64
```

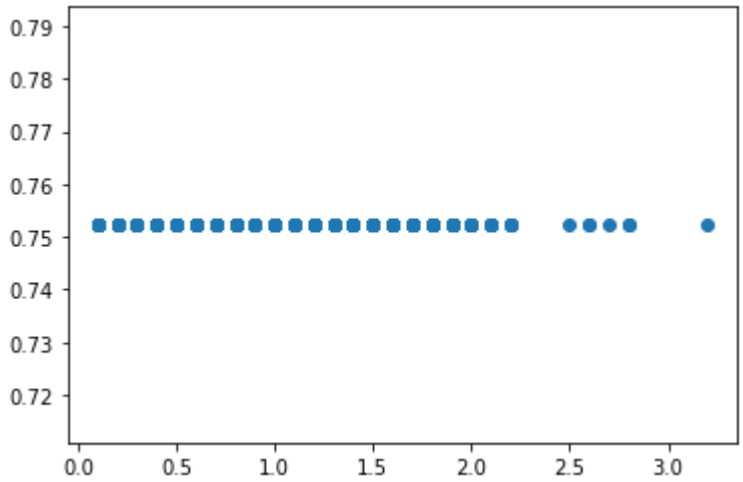
Lasso

In [14]:
```
la=Lasso(alpha=5)
la.fit(x_train,y_train)
```

Out[14]:
```
Lasso(alpha=5)
```

In [15]:
```
prediction1=la.predict(x_test)
plt.scatter(y_test,prediction1)
```

Out[15]: <matplotlib.collections.PathCollection at 0x25a0687ed90>



In [16]:
```
las=la.score(x_test,y_test)
```

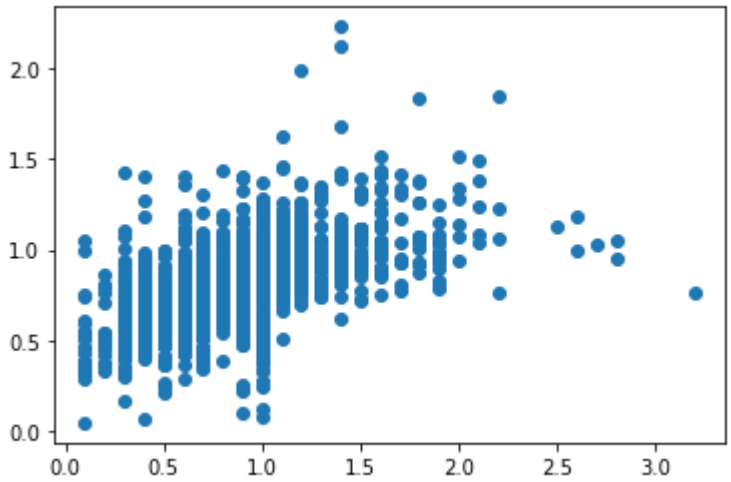
Ridge

In [17]:
```
rr=Ridge(alpha=1)
rr.fit(x_train,y_train)
```

Out[17]: Ridge(alpha=1)

In [18]:
```
prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[18]: <matplotlib.collections.PathCollection at 0x25a066009d0>



In [19]:
```
rrs=rr.score(x_test,y_test)
```

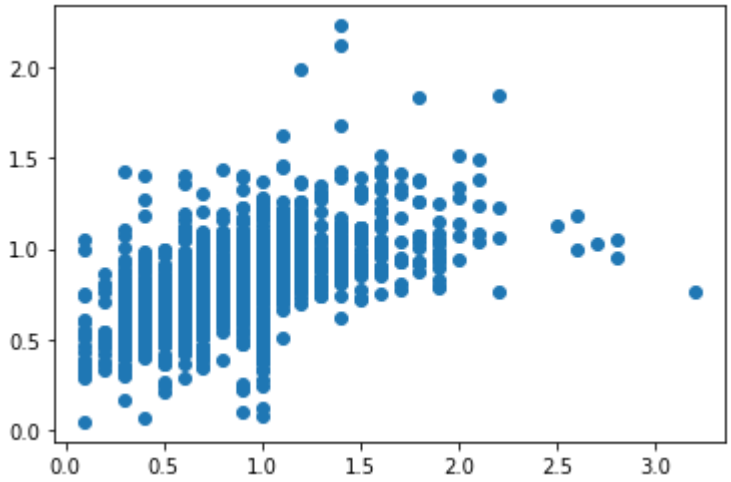
ElasticNet

In [20]:
```
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[20]:
```
ElasticNet()
```

In [21]:
```
prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[21]:
```
<matplotlib.collections.PathCollection at 0x25a07126cd0>
```



In [22]:
```
ens=en.score(x_test,y_test)
```

In [23]:
```
print(rr.score(x_test,y_test))
rr.score(x_train,y_train)
```

```
0.38794864124666883
```

Out[23]:
```
0.391803939842712
```

Logistic

In [24]:
```
g={"TCH":{1.0:"Low",2.0:"High"}}
df1=df1.replace(g)
df1["TCH"].value_counts()
```

Out[24]:
```
Low     5718
High    1597
Name: TCH, dtype: int64
```

In [25]:
```
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 [26]:
```
lo=LogisticRegression()
lo.fit(x_train,y_train)
```

Out[26]:
```
LogisticRegression()
```

In [27]:
```
prediction3=lo.predict(x_test)
plt.scatter(y_test,prediction3)
```

Out[27]:
```
<matplotlib.collections.PathCollection at 0x25a07187280>
```

In [28]:
```
los=lo.score(x_test,y_test)
```

Random Forest

In [29]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
```

In [30]:
```
g1={"TCH":{"Low":1.0,"High":2.0}}
df1=df1.replace(g1)
```

In [31]:
```
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 [32]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[32]:
```
RandomForestClassifier()
```

In [33]:
```
parameter={
    'max_depth':[1,2,4,5,6],
    'min_samples_leaf':[5,10,15,20,25],
    'n_estimators':[10,20,30,40,50]
}
```

In [34]:
```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accu
grid_search.fit(x_train,y_train)
```

Out[34]:
```
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 [35]:
```
rfcs=grid_search.best_score_
```

In [36]:
```
rfc_best=grid_search.best_estimator_
```

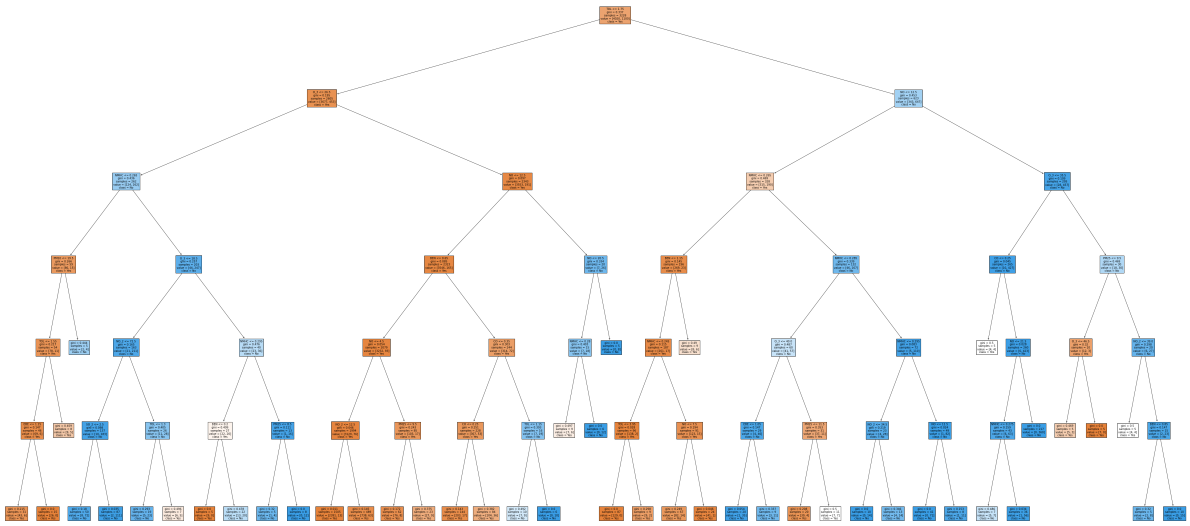
In [37]:
```
from sklearn.tree import plot_tree

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

Out[37]:
```
[Text(2305.4210526315787, 2019.0857142857144, 'TOL <= 1.75\ngini = 0.337\nsam
ples = 3228\nvalue = [4020, 1100]\nclass = Yes'),
 Text(1204.1052631578948, 1708.457142857143, 'O_3 <= 26.5\ngini = 0.195\nsamp
les = 2605\nvalue = [3677, 453]\nclass = Yes'),
 Text(469.89473684210526, 1397.8285714285716, 'NMHC <= 0.265\ngini = 0.436\ns
amples = 262\nvalue = [124, 262]\nclass = No'),
 Text(234.94736842105263, 1087.2, 'PM10 <= 19.5\ngini = 0.266\nsamples = 59\n
value = [80, 15]\nclass = Yes'),
 Text(176.21052631578948, 776.5714285714287, 'TOL <= 1.55\ngini = 0.217\nsamp
les = 54\nvalue = [78, 11]\nclass = Yes'),
 Text(117.47368421052632, 465.9428571428573, 'EBE <= 1.15\ngini = 0.147\nsamp
les = 46\nvalue = [69, 6]\nclass = Yes'),
 Text(58.73684210526316, 155.3142857142857, 'gini = 0.215\nsamples = 31\nvalu
e = [43, 6]\nclass = Yes'),
 Text(176.21052631578948, 155.3142857142857, 'gini = 0.0\nsamples = 15\nvalue
= [26, 0]\nclass = Yes'),
 Text(234.94736842105263, 465.9428571428573, 'gini = 0.459\nsamples = 8\nvalu
e = [9, 5]\nclass = Yes'),
 Text(293.6842105263158, 776.5714285714287, 'gini = 0.444\nsamples = 5\nvalue
= [2, 4]\nclass = No'),
 Text(704.8421052631579, 1087.2, 'O_3 <= 18.5\ngini = 0.257\nsamples = 203\nv
alue = [44, 247]\nclass = No'),
 Text(469.89473684210526, 776.5714285714287, 'NO_2 <= 72.5\ngini = 0.165\nsam
ples = 163\nvalue = [21, 211]\nclass = No'),
 Text(352.42105263157896, 465.9428571428573, 'SO_2 <= 2.5\ngini = 0.098\nsamp
les = 137\nvalue = [10, 183]\nclass = No'),
 Text(293.6842105263158, 155.3142857142857, 'gini = 0.18\nsamples = 50\nvalue
= [8, 72]\nclass = No'),
 Text(411.1578947368421, 155.3142857142857, 'gini = 0.035\nsamples = 87\nvalu
e = [2, 111]\nclass = No'),
 Text(587.3684210526316, 465.9428571428573, 'TOL <= 1.3\ngini = 0.405\nsample
s = 26\nvalue = [11, 28]\nclass = No'),
 Text(528.6315789473684, 155.3142857142857, 'gini = 0.293\nsamples = 19\nvalu
e = [5, 23]\nclass = No'),
 Text(646.1052631578947, 155.3142857142857, 'gini = 0.496\nsamples = 7\nvalue
= [6, 5]\nclass = Yes'),
 Text(939.7894736842105, 776.5714285714287, 'NMHC <= 0.295\ngini = 0.476\nsam
ples = 40\nvalue = [23, 36]\nclass = No'),
 Text(822.3157894736842, 465.9428571428573, 'BEN <= 0.2\ngini = 0.499\nsample
s = 27\nvalue = [22, 20]\nclass = Yes'),
 Text(763.578947368421, 155.3142857142857, 'gini = 0.0\nsamples = 5\nvalue =
[9, 0]\nclass = Yes'),
 Text(881.0526315789474, 155.3142857142857, 'gini = 0.478\nsamples = 22\nvalu
e = [13, 20]\nclass = No'),
 Text(1057.2631578947369, 465.9428571428573, 'PM25 <= 8.5\ngini = 0.111\nsamp
les = 13\nvalue = [1, 16]\nclass = No'),
 Text(998.5263157894736, 155.3142857142857, 'gini = 0.32\nsamples = 5\nvalue
= [1, 4]\nclass = No'),
 Text(1116.0, 155.3142857142857, 'gini = 0.0\nsamples = 8\nvalue = [0, 12]\nc
lass = No'),
 Text(1938.3157894736842, 1397.8285714285716, 'NO <= 12.5\ngini = 0.097\nsamp
les = 2343\nvalue = [3553, 191]\nclass = Yes'),
 Text(1644.6315789473683, 1087.2, 'BEN <= 0.65\ngini = 0.085\nsamples = 2323
\nvalue = [3546, 165]\nclass = Yes'),
 Text(1409.6842105263158, 776.5714285714287, 'NO <= 4.5\ngini = 0.054\nsample
s = 2079\nvalue = [3232, 93]\nclass = Yes'),
 Text(1292.2105263157894, 465.9428571428573, 'NO_2 <= 12.5\ngini = 0.046\nsam
```

```
ples = 1994\nvalue = [3129, 76]\nclass = Yes'),
 Text(1233.4736842105262, 155.3142857142857, 'gini = 0.011\nsamples = 1505\nv
alue = [2391, 13]\nclass = Yes'),
 Text(1350.9473684210527, 155.3142857142857, 'gini = 0.145\nsamples = 489\nva
lue = [738, 63]\nclass = Yes'),
 Text(1527.157894736842, 465.9428571428573, 'PM25 <= 9.5\ngini = 0.243\nsampl
es = 85\nvalue = [103, 17]\nclass = Yes'),
 Text(1468.421052631579, 155.3142857142857, 'gini = 0.172\nsamples = 62\nvalu
e = [76, 8]\nclass = Yes'),
 Text(1585.8947368421052, 155.3142857142857, 'gini = 0.375\nsamples = 23\nval
ue = [27, 9]\nclass = Yes'),
 Text(1879.578947368421, 776.5714285714287, 'CO <= 0.35\ngini = 0.303\nsample
s = 244\nvalue = [314, 72]\nclass = Yes'),
 Text(1762.1052631578948, 465.9428571428573, 'CO <= 0.25\ngini = 0.251\nsampl
es = 228\nvalue = [307, 53]\nclass = Yes'),
 Text(1703.3684210526317, 155.3142857142857, 'gini = 0.143\nsamples = 144\nva
lue = [203, 17]\nclass = Yes'),
 Text(1820.842105263158, 155.3142857142857, 'gini = 0.382\nsamples = 84\nvalu
e = [104, 36]\nclass = Yes'),
 Text(1997.0526315789473, 465.9428571428573, 'TOL <= 1.15\ngini = 0.393\nsamp
les = 16\nvalue = [7, 19]\nclass = No'),
 Text(1938.3157894736842, 155.3142857142857, 'gini = 0.492\nsamples = 10\nval
ue = [7, 9]\nclass = No'),
 Text(2055.7894736842104, 155.3142857142857, 'gini = 0.0\nsamples = 6\nvalue
= [0, 10]\nclass = No'),
 Text(2232.0, 1087.2, 'NO <= 20.5\ngini = 0.334\nsamples = 20\nvalue = [7, 2
6]\nclass = No'),
 Text(2173.2631578947367, 776.5714285714287, 'NMHC <= 0.28\ngini = 0.403\nsam
ples = 15\nvalue = [7, 18]\nclass = No'),
 Text(2114.5263157894738, 465.9428571428573, 'gini = 0.497\nsamples = 9\nvalu
e = [7, 6]\nclass = Yes'),
 Text(2232.0, 465.9428571428573, 'gini = 0.0\nsamples = 6\nvalue = [0, 12]\nc
lass = No'),
 Text(2290.7368421052633, 776.5714285714287, 'gini = 0.0\nsamples = 5\nvalue
= [0, 8]\nclass = No'),
 Text(3406.7368421052633, 1708.457142857143, 'NO <= 12.5\ngini = 0.453\nsampl
es = 623\nvalue = [343, 647]\nclass = No'),
 Text(2848.7368421052633, 1397.8285714285716, 'NMHC <= 0.265\ngini = 0.469\ns
amples = 328\nvalue = [315, 190]\nclass = Yes'),
 Text(2525.684210526316, 1087.2, 'BEN <= 1.15\ngini = 0.145\nsamples = 196\nv
alue = [269, 23]\nclass = Yes'),
 Text(2466.9473684210525, 776.5714285714287, 'NMHC <= 0.245\ngini = 0.115\nsa
mples = 187\nvalue = [261, 17]\nclass = Yes'),
 Text(2349.4736842105262, 465.9428571428573, 'TOL <= 3.95\ngini = 0.028\nsamp
les = 96\nvalue = [138, 2]\nclass = Yes'),
 Text(2290.7368421052633, 155.3142857142857, 'gini = 0.0\nsamples = 87\nvalue
= [129, 0]\nclass = Yes'),
 Text(2408.2105263157896, 155.3142857142857, 'gini = 0.298\nsamples = 9\nvalu
e = [9, 2]\nclass = Yes'),
 Text(2584.4210526315787, 465.9428571428573, 'NO <= 7.5\ngini = 0.194\nsample
s = 91\nvalue = [123, 15]\nclass = Yes'),
 Text(2525.684210526316, 155.3142857142857, 'gini = 0.249\nsamples = 67\nvalu
e = [82, 14]\nclass = Yes'),
 Text(2643.157894736842, 155.3142857142857, 'gini = 0.046\nsamples = 24\nvalu
e = [41, 1]\nclass = Yes'),
 Text(2584.4210526315787, 776.5714285714287, 'gini = 0.49\nsamples = 9\nvalue
= [8, 6]\nclass = Yes'),
```

```
 Text(3171.7894736842104, 1087.2, 'NMHC <= 0.285\ngini = 0.339\nsamples = 132
\nvalue = [46, 167]\nclass = No'),
 Text(2936.842105263158, 776.5714285714287, 'O_3 <= 40.0\ngini = 0.487\nsampl
es = 60\nvalue = [41, 57]\nclass = No'),
 Text(2819.3684210526317, 465.9428571428573, 'EBE <= 1.05\ngini = 0.147\nsamp
les = 29\nvalue = [4, 46]\nclass = No'),
 Text(2760.6315789473683, 155.3142857142857, 'gini = 0.054\nsamples = 20\nval
ue = [1, 35]\nclass = No'),
 Text(2878.1052631578946, 155.3142857142857, 'gini = 0.337\nsamples = 9\nvalu
e = [3, 11]\nclass = No'),
 Text(3054.315789473684, 465.9428571428573, 'PM25 <= 11.5\ngini = 0.353\nsamp
les = 31\nvalue = [37, 11]\nclass = Yes'),
 Text(2995.578947368421, 155.3142857142857, 'gini = 0.208\nsamples = 20\nvalu
e = [30, 4]\nclass = Yes'),
 Text(3113.0526315789475, 155.3142857142857, 'gini = 0.5\nsamples = 11\nvalue
= [7, 7]\nclass = Yes'),
 Text(3406.7368421052633, 776.5714285714287, 'NMHC <= 0.295\ngini = 0.083\nsa
mples = 72\nvalue = [5, 110]\nclass = No'),
 Text(3289.2631578947367, 465.9428571428573, 'NO_2 <= 34.5\ngini = 0.219\nsam
ples = 23\nvalue = [4, 28]\nclass = No'),
 Text(3230.5263157894738, 155.3142857142857, 'gini = 0.0\nsamples = 10\nvalue
= [0, 14]\nclass = No'),
 Text(3348.0, 155.3142857142857, 'gini = 0.346\nsamples = 13\nvalue = [4, 14]
\nclass = No'),
 Text(3524.2105263157896, 465.9428571428573, 'NO <= 11.5\ngini = 0.024\nsampl
es = 49\nvalue = [1, 82]\nclass = No'),
 Text(3465.4736842105262, 155.3142857142857, 'gini = 0.0\nsamples = 41\nvalue
= [0, 71]\nclass = No'),
 Text(3582.9473684210525, 155.3142857142857, 'gini = 0.153\nsamples = 8\nvalu
e = [1, 11]\nclass = No'),
 Text(3964.7368421052633, 1397.8285714285716, 'O_3 <= 35.5\ngini = 0.109\nsam
ples = 295\nvalue = [28, 457]\nclass = No'),
 Text(3759.157894736842, 1087.2, 'CO <= 0.25\ngini = 0.045\nsamples = 265\nva
lue = [10, 427]\nclass = No'),
 Text(3700.4210526315787, 776.5714285714287, 'gini = 0.5\nsamples = 5\nvalue
= [4, 4]\nclass = Yes'),
 Text(3817.8947368421054, 776.5714285714287, 'NO <= 21.5\ngini = 0.028\nsampl
es = 260\nvalue = [6, 423]\nclass = No'),
 Text(3759.157894736842, 465.9428571428573, 'NMHC <= 0.275\ngini = 0.159\nsam
ples = 43\nvalue = [6, 63]\nclass = No'),
 Text(3700.4210526315787, 155.3142857142857, 'gini = 0.486\nsamples = 7\nvalu
e = [5, 7]\nclass = No'),
 Text(3817.8947368421054, 155.3142857142857, 'gini = 0.034\nsamples = 36\nval
ue = [1, 56]\nclass = No'),
 Text(3876.6315789473683, 465.9428571428573, 'gini = 0.0\nsamples = 217\nvalu
e = [0, 360]\nclass = No'),
 Text(4170.315789473684, 1087.2, 'PM25 <= 9.5\ngini = 0.469\nsamples = 30\nva
lue = [18, 30]\nclass = No'),
 Text(4052.842105263158, 776.5714285714287, 'O_3 <= 46.5\ngini = 0.32\nsample
s = 10\nvalue = [12, 3]\nclass = Yes'),
 Text(3994.1052631578946, 465.9428571428573, 'gini = 0.469\nsamples = 5\nvalu
e = [5, 3]\nclass = Yes'),
 Text(4111.578947368421, 465.9428571428573, 'gini = 0.0\nsamples = 5\nvalue =
[7, 0]\nclass = Yes'),
 Text(4287.789473684211, 776.5714285714287, 'NO_2 <= 39.0\ngini = 0.298\nsamp
les = 20\nvalue = [6, 27]\nclass = No'),
 Text(4229.0526315789475, 465.9428571428573, 'gini = 0.5\nsamples = 5\nvalue
```

```
= [4, 4]\nclass = Yes'),
 Text(4346.526315789473, 465.9428571428573, 'BEN <= 0.85\ngini = 0.147\nsampl
es = 15\nvalue = [2, 23]\nclass = No'),
 Text(4287.789473684211, 155.3142857142857, 'gini = 0.32\nsamples = 5\nvalue
= [2, 8]\nclass = No'),
 Text(4405.263157894737, 155.3142857142857, 'gini = 0.0\nsamples = 10\nvalue
= [0, 15]\nclass = No')]
```



In [38]:
```
print("Linear:",lis)
print("Lasso:",las)
print("Ridge:",rrs)
print("ElasticNet:",ens)
print("Logistic:",los)
print("Random Forest:",rfcs)
```

```
Linear: 0.4114978071131482
Lasso: -6.234443005292967e-05
Ridge: 0.38794864124666883
ElasticNet: 0.09906134120330623
Logistic: 0.7831435079726652
Random Forest: 0.9505859375000001
```

Best Model is Random Forest

2014

In [39]:
```
df2=pd.read_csv("madrid_2014.csv")
df2
```

Out[39]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	
0	2014-06-01 01:00:00	NaN	0.2	NaN	NaN	3.0	10.0	NaN	NaN	NaN	3.0	NaN	NaN	28
1	2014-06-01 01:00:00	0.2	0.2	0.1	0.11	3.0	17.0	68.0	10.0	5.0	5.0	1.36	1.3	28
2	2014-06-01 01:00:00	0.3	NaN	0.1	NaN	2.0	6.0	NaN	NaN	NaN	NaN	NaN	1.1	28
3	2014-06-01 01:00:00	NaN	0.2	NaN	NaN	1.0	6.0	79.0	NaN	NaN	NaN	NaN	NaN	28
4	2014-06-01 01:00:00	NaN	NaN	NaN	NaN	1.0	6.0	75.0	NaN	NaN	4.0	NaN	NaN	28
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
210019	2014-09-01 00:00:00	NaN	0.5	NaN	NaN	20.0	84.0	29.0	NaN	NaN	NaN	NaN	NaN	28
210020	2014-09-01 00:00:00	NaN	0.3	NaN	NaN	1.0	22.0	NaN	15.0	NaN	6.0	NaN	NaN	28
210021	2014-09-01 00:00:00	NaN	NaN	NaN	NaN	1.0	13.0	70.0	NaN	NaN	NaN	NaN	NaN	28
210022	2014-09-01 00:00:00	NaN	NaN	NaN	NaN	3.0	38.0	42.0	NaN	NaN	NaN	NaN	NaN	28
210023	2014-09-01 00:00:00	NaN	NaN	NaN	NaN	1.0	26.0	65.0	11.0	NaN	NaN	NaN	NaN	28

210024 rows × 14 columns

In [40]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210024 entries, 0 to 210023
Data columns (total 14 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     210024 non-null  object 
 1   BEN      46703 non-null   float64
 2   CO       87023 non-null   float64
 3   EBE      46722 non-null   float64
 4   NMHC     25021 non-null   float64
 5   NO       209154 non-null  float64
 6   NO_2     209154 non-null  float64
 7   O_3      121681 non-null  float64
 8   PM10     104311 non-null  float64
 9   PM25     51954 non-null   float64
 10  SO_2     87141 non-null   float64
 11  TCH      25021 non-null   float64
 12  TOL      46570 non-null   float64
 13  station  210024 non-null  int64  
dtypes: float64(12), int64(1), object(1)
memory usage: 22.4+ MB
```

In [41]:
```
df3=df2.dropna()
df3
```

Out[41]:

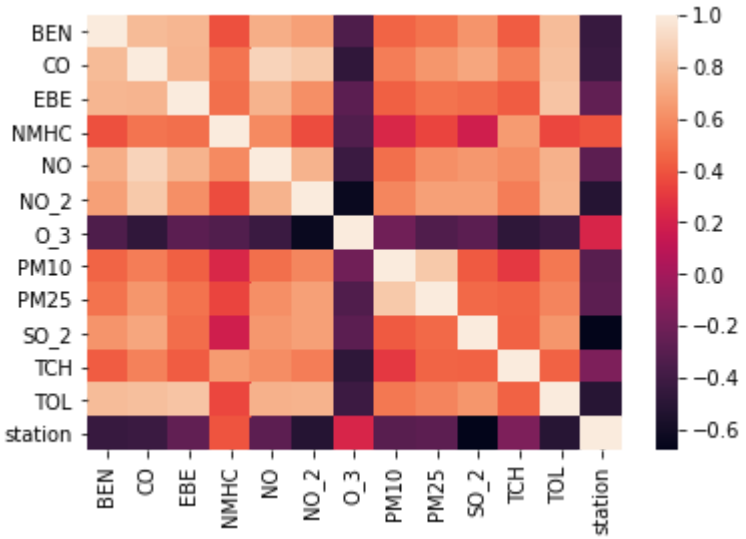
	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	s
1	2014-06-01 01:00:00	0.2	0.2	0.1	0.11	3.0	17.0	68.0	10.0	5.0	5.0	1.36	1.3	280
6	2014-06-01 01:00:00	0.1	0.2	0.1	0.23	1.0	5.0	80.0	4.0	3.0	2.0	1.21	0.1	280
25	2014-06-01 02:00:00	0.2	0.2	0.1	0.11	4.0	21.0	63.0	9.0	6.0	5.0	1.36	0.8	280
30	2014-06-01 02:00:00	0.2	0.2	0.1	0.23	1.0	4.0	88.0	7.0	5.0	2.0	1.21	0.1	280
49	2014-06-01 03:00:00	0.1	0.2	0.1	0.11	4.0	18.0	66.0	9.0	7.0	6.0	1.36	0.9	280
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
209958	2014-08-31 22:00:00	0.2	0.2	0.1	0.22	1.0	28.0	96.0	61.0	15.0	3.0	1.28	0.1	280
209977	2014-08-31 23:00:00	1.1	0.7	0.7	0.19	36.0	118.0	23.0	60.0	25.0	9.0	1.27	6.5	280
209982	2014-08-31 23:00:00	0.2	0.2	0.1	0.21	1.0	17.0	90.0	28.0	14.0	3.0	1.27	0.2	280
210001	2014-09-01 00:00:00	0.6	0.4	0.4	0.12	6.0	63.0	41.0	26.0	15.0	8.0	1.19	4.1	280
210006	2014-09-01 00:00:00	0.2	0.2	0.1	0.23	1.0	30.0	69.0	18.0	13.0	3.0	1.30	0.1	280

13946 rows × 14 columns

In [42]:
```
df3=df3.drop(["date"],axis=1)
```

In [43]:
```
sns.heatmap(df3.corr())
```

Out[43]: <AxesSubplot:>



In [44]:
```
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

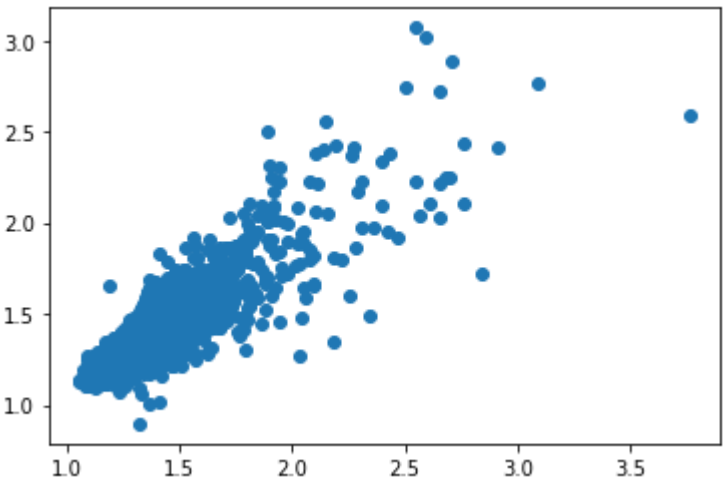
In [45]:
```
li=LinearRegression()
li.fit(x_train,y_train)
```

Out[45]: LinearRegression()

In [ ]:
```

```

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

Out[46]: <matplotlib.collections.PathCollection at 0x25a0b165d60>



In [47]:
```
lis=li.score(x_test,y_test)
```

In [48]:
```
df3["TCH"].value_counts()
```

Out[48]:
```
1.37    601
1.36    598
1.34    529
1.35    528
1.38    515
       ...
2.50      1
2.86      1
2.70      1
3.04      1
4.37      1
Name: TCH, Length: 184, dtype: int64
```

In [49]:
```
df3.loc[df3["TCH"]<1.40,"TCH"]=1
df3.loc[df3["TCH"]>1.40,"TCH"]=2
df3["TCH"].value_counts()
```

Out[49]:
```
1.0    9997
2.0    3949
Name: TCH, dtype: int64
```

In [ ]:
```

```

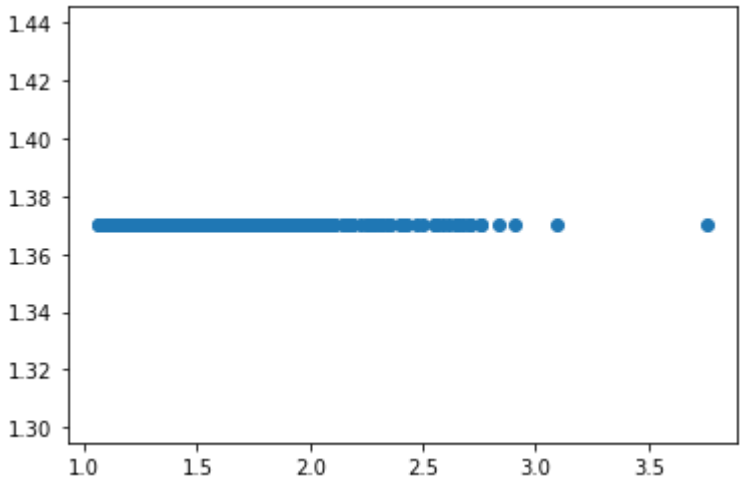
Lasso

In [50]:
```
la=Lasso(alpha=5)
la.fit(x_train,y_train)
```

Out[50]:
```
Lasso(alpha=5)
```

In [51]:
```
prediction1=la.predict(x_test)
plt.scatter(y_test,prediction1)
```

Out[51]: <matplotlib.collections.PathCollection at 0x25a0b1bca90>



In [52]:
```
las=la.score(x_test,y_test)
```

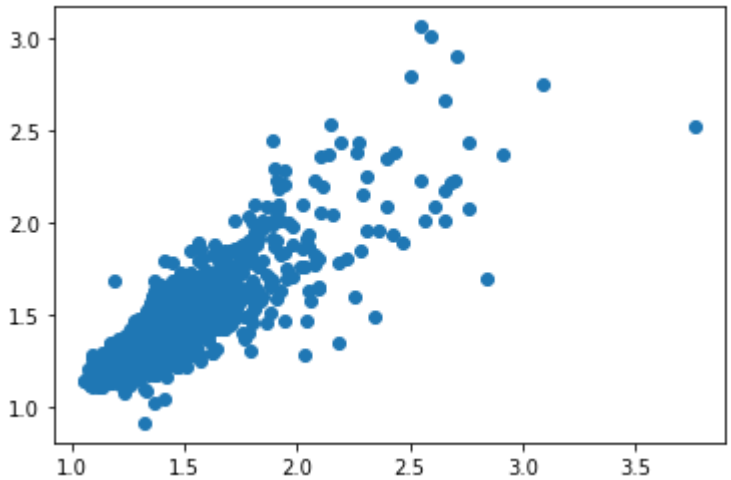
Ridge

In [53]:
```
rr=Ridge(alpha=1)
rr.fit(x_train,y_train)
```

Out[53]: Ridge(alpha=1)

In [54]:
```
prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[54]: <matplotlib.collections.PathCollection at 0x25a0b21f100>



In [55]:
```
rrs=rr.score(x_test,y_test)
```

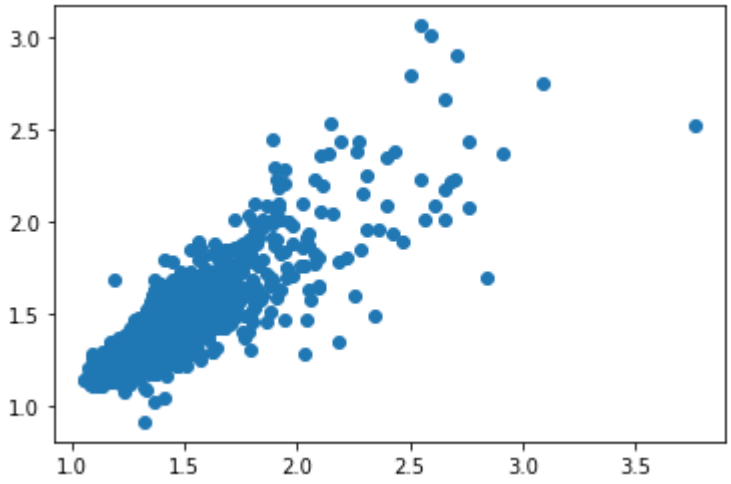
ElasticNet

In [56]:
```
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[56]:
```
ElasticNet()
```

In [57]:
```
prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[57]:
```
<matplotlib.collections.PathCollection at 0x25a0b26e6a0>
```



In [58]:
```
ens=en.score(x_test,y_test)
```

In [59]:
```
print(rr.score(x_test,y_test))
rr.score(x_train,y_train)
```

```
0.7037488127556343
```

Out[59]:
```
0.7081013643502528
```

Logistic

In [60]:
```
g={"TCH":{1.0:"Low",2.0:"High"}}
df3=df3.replace(g)
df3["TCH"].value_counts()
```

Out[60]:
```
Low     9997
High    3949
Name: TCH, dtype: int64
```

In [61]:
```
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 [62]:
```
lo=LogisticRegression()
lo.fit(x_train,y_train)
```

Out[62]:
```
LogisticRegression()
```

In [63]:
```
prediction3=lo.predict(x_test)
plt.scatter(y_test,prediction3)
```

Out[63]:
```
<matplotlib.collections.PathCollection at 0x25a0b2a06a0>
```

In [64]:
```
los=lo.score(x_test,y_test)
```

Random Forest

In [65]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
```

In [66]:
```
g1={"TCH":{"Low":1.0,"High":2.0}}
df3=df3.replace(g1)
```

In [67]:
```
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 [68]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[68]:
```
RandomForestClassifier()
```

In [69]:
```
parameter={
    'max_depth':[1,2,4,5,6],
    'min_samples_leaf':[5,10,15,20,25],
    'n_estimators':[10,20,30,40,50]
}
```

In [70]:
```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accu
grid_search.fit(x_train,y_train)
```

Out[70]:
```
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 [71]:
```
rfcs=grid_search.best_score_
```

In [72]:
```
rfc_best=grid_search.best_estimator_
```

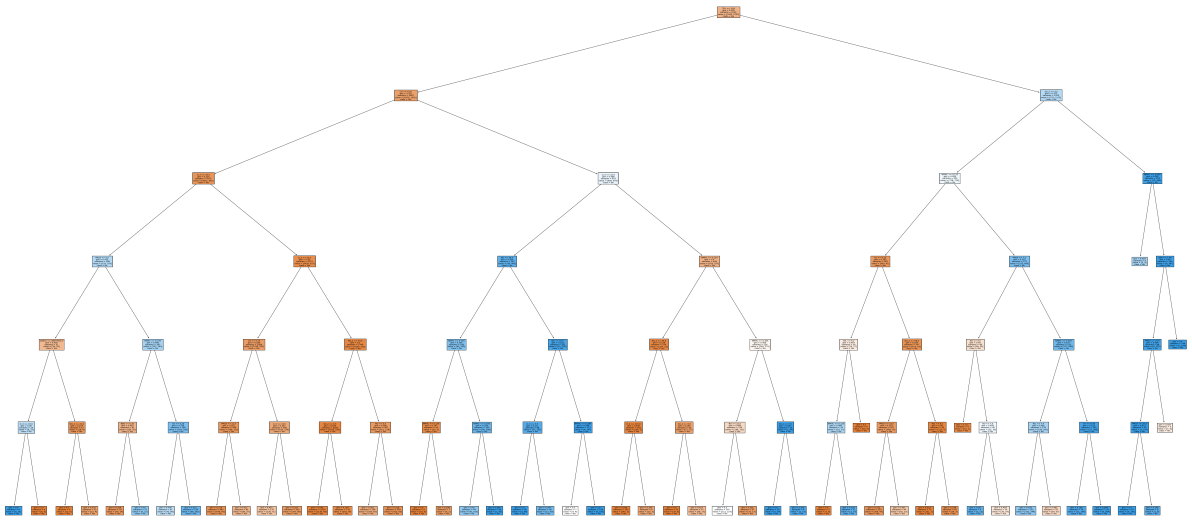
In [73]:
```
from sklearn.tree import plot_tree

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

Out[73]:
```
[Text(2730.6382978723404, 2019.0857142857144, 'TOL <= 3.65\ngini = 0.405\nsam
ples = 6192\nvalue = [7009, 2753]\nclass = Yes'),
 Text(1519.659574468085, 1708.457142857143, 'CO <= 0.35\ngini = 0.321\nsample
s = 4987\nvalue = [6278, 1581]\nclass = Yes'),
 Text(759.8297872340426, 1397.8285714285716, 'O_3 <= 24.5\ngini = 0.237\nsamp
les = 4172\nvalue = [5672, 902]\nclass = Yes'),
 Text(379.9148936170213, 1087.2, 'PM25 <= 5.5\ngini = 0.472\nsamples = 398\nv
alue = [233, 379]\nclass = No'),
 Text(189.95744680851064, 776.5714285714287, 'station <= 28079016.0\ngini =
0.426\nsamples = 31\nvalue = [36, 16]\nclass = Yes'),
 Text(94.97872340425532, 465.9428571428573, 'O_3 <= 19.0\ngini = 0.48\nsample
s = 14\nvalue = [8, 12]\nclass = No'),
 Text(47.48936170212766, 155.3142857142857, 'gini = 0.0\nsamples = 8\nvalue =
[0, 12]\nclass = No'),
 Text(142.46808510638297, 155.3142857142857, 'gini = 0.0\nsamples = 6\nvalue
= [8, 0]\nclass = Yes'),
 Text(284.93617021276594, 465.9428571428573, 'NO_2 <= 21.0\ngini = 0.219\nsam
ples = 17\nvalue = [28, 4]\nclass = Yes'),
 Text(237.4468085106383, 155.3142857142857, 'gini = 0.0\nsamples = 5\nvalue =
[13, 0]\nclass = Yes'),
 Text(332.4255319148936, 155.3142857142857, 'gini = 0.332\nsamples = 12\nvalu
e = [15, 4]\nclass = Yes'),
 Text(569.8723404255319, 776.5714285714287, 'NMHC <= 0.235\ngini = 0.456\nsam
ples = 367\nvalue = [197, 363]\nclass = No'),
 Text(474.8936170212766, 465.9428571428573, 'BEN <= 0.35\ngini = 0.407\nsampl
es = 81\nvalue = [88, 35]\nclass = Yes'),
 Text(427.40425531914894, 155.3142857142857, 'gini = 0.256\nsamples = 59\nval
ue = [79, 14]\nclass = Yes'),
 Text(522.3829787234042, 155.3142857142857, 'gini = 0.42\nsamples = 22\nvalue
= [9, 21]\nclass = No'),
 Text(664.8510638297872, 465.9428571428573, 'CO <= 0.25\ngini = 0.374\nsample
s = 286\nvalue = [109, 328]\nclass = No'),
 Text(617.3617021276596, 155.3142857142857, 'gini = 0.478\nsamples = 67\nvalu
e = [43, 66]\nclass = No'),
 Text(712.3404255319149, 155.3142857142857, 'gini = 0.321\nsamples = 219\nval
ue = [66, 262]\nclass = No'),
 Text(1139.7446808510638, 1087.2, 'O_3 <= 52.5\ngini = 0.16\nsamples = 3774\n
value = [5439, 523]\nclass = Yes'),
 Text(949.7872340425532, 776.5714285714287, 'CO <= 0.25\ngini = 0.299\nsample
s = 1008\nvalue = [1299, 291]\nclass = Yes'),
 Text(854.8085106382979, 465.9428571428573, 'PM25 <= 11.5\ngini = 0.251\nsamp
les = 568\nvalue = [766, 132]\nclass = Yes'),
 Text(807.3191489361702, 155.3142857142857, 'gini = 0.216\nsamples = 483\nval
ue = [678, 95]\nclass = Yes'),
 Text(902.2978723404256, 155.3142857142857, 'gini = 0.417\nsamples = 85\nvalu
e = [88, 37]\nclass = Yes'),
 Text(1044.7659574468084, 465.9428571428573, 'O_3 <= 33.5\ngini = 0.354\nsamp
les = 440\nvalue = [533, 159]\nclass = Yes'),
 Text(997.2765957446809, 155.3142857142857, 'gini = 0.451\nsamples = 137\nval
ue = [136, 71]\nclass = Yes'),
 Text(1092.2553191489362, 155.3142857142857, 'gini = 0.297\nsamples = 303\nva
lue = [397, 88]\nclass = Yes'),
 Text(1329.7021276595744, 776.5714285714287, 'NO_2 <= 24.5\ngini = 0.1\nsampl
es = 2766\nvalue = [4140, 232]\nclass = Yes'),
 Text(1234.723404255319, 465.9428571428573, 'SO_2 <= 5.5\ngini = 0.056\nsampl
es = 2299\nvalue = [3529, 104]\nclass = Yes'),
 Text(1187.2340425531916, 155.3142857142857, 'gini = 0.042\nsamples = 2029\nv
```

```
alue = [3136, 68]\nclass = Yes'),
 Text(1282.212765957447, 155.3142857142857, 'gini = 0.154\nsamples = 270\nval
ue = [393, 36]\nclass = Yes'),
 Text(1424.6808510638298, 465.9428571428573, 'NO <= 5.5\ngini = 0.286\nsample
s = 467\nvalue = [611, 128]\nclass = Yes'),
 Text(1377.1914893617022, 155.3142857142857, 'gini = 0.381\nsamples = 158\nva
lue = [189, 65]\nclass = Yes'),
 Text(1472.1702127659576, 155.3142857142857, 'gini = 0.226\nsamples = 309\nva
lue = [422, 63]\nclass = Yes'),
 Text(2279.4893617021276, 1397.8285714285716, 'O_3 <= 22.5\ngini = 0.498\nsam
ples = 815\nvalue = [606, 679]\nclass = No'),
 Text(1899.5744680851064, 1087.2, 'NO <= 35.5\ngini = 0.265\nsamples = 367\nv
alue = [90, 482]\nclass = No'),
 Text(1709.6170212765958, 776.5714285714287, 'NMHC <= 0.155\ngini = 0.391\nsa
mples = 166\nvalue = [69, 190]\nclass = No'),
 Text(1614.6382978723404, 465.9428571428573, 'NMHC <= 0.135\ngini = 0.147\nsa
mples = 16\nvalue = [23, 2]\nclass = Yes'),
 Text(1567.1489361702127, 155.3142857142857, 'gini = 0.0\nsamples = 9\nvalue
= [14, 0]\nclass = Yes'),
 Text(1662.127659574468, 155.3142857142857, 'gini = 0.298\nsamples = 7\nvalue
= [9, 2]\nclass = Yes'),
 Text(1804.595744680851, 465.9428571428573, 'NMHC <= 0.295\ngini = 0.316\nsam
ples = 150\nvalue = [46, 188]\nclass = No'),
 Text(1757.1063829787233, 155.3142857142857, 'gini = 0.42\nsamples = 94\nvalu
e = [45, 105]\nclass = No'),
 Text(1852.0851063829787, 155.3142857142857, 'gini = 0.024\nsamples = 56\nval
ue = [1, 83]\nclass = No'),
 Text(2089.531914893617, 776.5714285714287, 'NO <= 54.5\ngini = 0.125\nsample
s = 201\nvalue = [21, 292]\nclass = No'),
 Text(1994.5531914893618, 465.9428571428573, 'O_3 <= 5.5\ngini = 0.211\nsampl
es = 88\nvalue = [17, 125]\nclass = No'),
 Text(1947.063829787234, 155.3142857142857, 'gini = 0.0\nsamples = 38\nvalue
= [0, 66]\nclass = No'),
 Text(2042.0425531914893, 155.3142857142857, 'gini = 0.347\nsamples = 50\nval
ue = [17, 59]\nclass = No'),
 Text(2184.5106382978724, 465.9428571428573, 'NMHC <= 0.285\ngini = 0.046\nsa
mples = 113\nvalue = [4, 167]\nclass = No'),
 Text(2137.021276595745, 155.3142857142857, 'gini = 0.5\nsamples = 7\nvalue =
[4, 4]\nclass = Yes'),
 Text(2232.0, 155.3142857142857, 'gini = 0.0\nsamples = 106\nvalue = [0, 163]
\nclass = No'),
 Text(2659.404255319149, 1087.2, 'NMHC <= 0.155\ngini = 0.4\nsamples = 448\nv
alue = [516, 197]\nclass = Yes'),
 Text(2469.446808510638, 776.5714285714287, 'NO_2 <= 56.5\ngini = 0.118\nsamp
les = 224\nvalue = [327, 22]\nclass = Yes'),
 Text(2374.468085106383, 465.9428571428573, 'O_3 <= 115.0\ngini = 0.038\nsamp
les = 165\nvalue = [250, 5]\nclass = Yes'),
 Text(2326.978723404255, 155.3142857142857, 'gini = 0.016\nsamples = 159\nval
ue = [240, 2]\nclass = Yes'),
 Text(2421.9574468085107, 155.3142857142857, 'gini = 0.355\nsamples = 6\nvalu
e = [10, 3]\nclass = Yes'),
 Text(2564.425531914894, 465.9428571428573, 'SO_2 <= 5.5\ngini = 0.296\nsampl
es = 59\nvalue = [77, 17]\nclass = Yes'),
 Text(2516.936170212766, 155.3142857142857, 'gini = 0.0\nsamples = 22\nvalue
= [39, 0]\nclass = Yes'),
 Text(2611.9148936170213, 155.3142857142857, 'gini = 0.427\nsamples = 37\nval
ue = [38, 17]\nclass = Yes'),
```

```
 Text(2849.3617021276596, 776.5714285714287, 'NMHC <= 0.305\ngini = 0.499\nsa
mples = 224\nvalue = [189, 175]\nclass = Yes'),
 Text(2754.3829787234044, 465.9428571428573, 'NO <= 22.5\ngini = 0.484\nsampl
es = 199\nvalue = [188, 131]\nclass = Yes'),
 Text(2706.8936170212764, 155.3142857142857, 'gini = 0.5\nsamples = 142\nvalu
e = [113, 114]\nclass = No'),
 Text(2801.872340425532, 155.3142857142857, 'gini = 0.301\nsamples = 57\nvalu
e = [75, 17]\nclass = Yes'),
 Text(2944.340425531915, 465.9428571428573, 'SO_2 <= 5.5\ngini = 0.043\nsampl
es = 25\nvalue = [1, 44]\nclass = No'),
 Text(2896.851063829787, 155.3142857142857, 'gini = 0.0\nsamples = 16\nvalue
= [0, 34]\nclass = No'),
 Text(2991.8297872340427, 155.3142857142857, 'gini = 0.165\nsamples = 9\nvalu
e = [1, 10]\nclass = No'),
 Text(3941.6170212765956, 1708.457142857143, 'SO_2 <= 9.5\ngini = 0.473\nsamp
les = 1205\nvalue = [731, 1172]\nclass = No'),
 Text(3561.7021276595747, 1397.8285714285716, 'NMHC <= 0.175\ngini = 0.499\ns
amples = 952\nvalue = [718, 776]\nclass = No'),
 Text(3300.5106382978724, 1087.2, 'CO <= 0.25\ngini = 0.263\nsamples = 381\nv
alue = [493, 91]\nclass = Yes'),
 Text(3181.7872340425533, 776.5714285714287, 'TOL <= 6.2\ngini = 0.497\nsampl
es = 24\nvalue = [20, 17]\nclass = Yes'),
 Text(3134.2978723404253, 465.9428571428573, 'NMHC <= 0.125\ngini = 0.466\nsa
mples = 16\nvalue = [10, 17]\nclass = No'),
 Text(3086.808510638298, 155.3142857142857, 'gini = 0.0\nsamples = 5\nvalue =
[7, 0]\nclass = Yes'),
 Text(3181.7872340425533, 155.3142857142857, 'gini = 0.255\nsamples = 11\nval
ue = [3, 17]\nclass = No'),
 Text(3229.276595744681, 465.9428571428573, 'gini = 0.0\nsamples = 8\nvalue =
[10, 0]\nclass = Yes'),
 Text(3419.2340425531916, 776.5714285714287, 'NO_2 <= 68.5\ngini = 0.234\nsam
ples = 357\nvalue = [473, 74]\nclass = Yes'),
 Text(3324.255319148936, 465.9428571428573, 'PM25 <= 15.5\ngini = 0.279\nsamp
les = 267\nvalue = [348, 70]\nclass = Yes'),
 Text(3276.7659574468084, 155.3142857142857, 'gini = 0.231\nsamples = 223\nva
lue = [306, 47]\nclass = Yes'),
 Text(3371.744680851064, 155.3142857142857, 'gini = 0.457\nsamples = 44\nvalu
e = [42, 23]\nclass = Yes'),
 Text(3514.2127659574467, 465.9428571428573, 'TOL <= 8.1\ngini = 0.06\nsample
s = 90\nvalue = [125, 4]\nclass = Yes'),
 Text(3466.723404255319, 155.3142857142857, 'gini = 0.019\nsamples = 72\nvalu
e = [101, 1]\nclass = Yes'),
 Text(3561.7021276595747, 155.3142857142857, 'gini = 0.198\nsamples = 18\nval
ue = [24, 3]\nclass = Yes'),
 Text(3822.8936170212764, 1087.2, 'PM25 <= 7.5\ngini = 0.372\nsamples = 571\n
value = [225, 685]\nclass = No'),
 Text(3656.68085106383, 776.5714285714287, 'NO <= 5.5\ngini = 0.49\nsamples =
48\nvalue = [48, 36]\nclass = Yes'),
 Text(3609.191489361702, 465.9428571428573, 'gini = 0.111\nsamples = 8\nvalue
= [16, 1]\nclass = Yes'),
 Text(3704.1702127659573, 465.9428571428573, 'NO <= 7.5\ngini = 0.499\nsample
s = 40\nvalue = [32, 35]\nclass = No'),
 Text(3656.68085106383, 155.3142857142857, 'gini = 0.26\nsamples = 5\nvalue =
[2, 11]\nclass = No'),
 Text(3751.6595744680853, 155.3142857142857, 'gini = 0.494\nsamples = 35\nval
ue = [30, 24]\nclass = Yes'),
 Text(3989.1063829787236, 776.5714285714287, 'NMHC <= 0.255\ngini = 0.337\nsa
```

```
mples = 523\nvalue = [177, 649]\nclass = No'),
 Text(3894.127659574468, 465.9428571428573, 'TOL <= 7.4\ngini = 0.459\nsample
s = 278\nvalue = [156, 281]\nclass = No'),
 Text(3846.6382978723404, 155.3142857142857, 'gini = 0.416\nsamples = 223\nva
lue = [102, 244]\nclass = No'),
 Text(3941.6170212765956, 155.3142857142857, 'gini = 0.483\nsamples = 55\nval
ue = [54, 37]\nclass = Yes'),
 Text(4084.0851063829787, 465.9428571428573, 'CO <= 0.45\ngini = 0.102\nsampl
es = 245\nvalue = [21, 368]\nclass = No'),
 Text(4036.595744680851, 155.3142857142857, 'gini = 0.185\nsamples = 74\nvalu
e = [12, 104]\nclass = No'),
 Text(4131.574468085107, 155.3142857142857, 'gini = 0.064\nsamples = 171\nval
ue = [9, 264]\nclass = No'),
 Text(4321.531914893617, 1397.8285714285716, 'NMHC <= 0.245\ngini = 0.062\nsa
mples = 253\nvalue = [13, 396]\nclass = No'),
 Text(4274.04255319149, 1087.2, 'gini = 0.459\nsamples = 9\nvalue = [5, 9]\nc
lass = No'),
 Text(4369.021276595745, 1087.2, 'BEN <= 1.45\ngini = 0.04\nsamples = 244\nva
lue = [8, 387]\nclass = No'),
 Text(4321.531914893617, 776.5714285714287, 'PM25 <= 24.5\ngini = 0.094\nsamp
les = 98\nvalue = [8, 153]\nclass = No'),
 Text(4274.04255319149, 465.9428571428573, 'PM25 <= 20.5\ngini = 0.013\nsampl
es = 93\nvalue = [1, 147]\nclass = No'),
 Text(4226.553191489362, 155.3142857142857, 'gini = 0.0\nsamples = 87\nvalue
= [0, 141]\nclass = No'),
 Text(4321.531914893617, 155.3142857142857, 'gini = 0.245\nsamples = 6\nvalue
= [1, 6]\nclass = No'),
 Text(4369.021276595745, 465.9428571428573, 'gini = 0.497\nsamples = 5\nvalue
= [7, 6]\nclass = Yes'),
 Text(4416.510638297872, 776.5714285714287, 'gini = 0.0\nsamples = 146\nvalue
= [0, 234]\nclass = No')]
```



In [74]:
```
print("Linear:",lis)
print("Lasso:",las)
print("Ridge:",rrs)
print("ElasticNet:",ens)
print("Logistic:",los)
print("Random Forest:",rfcs)
```

```
Linear: 0.6984961782299648
Lasso: -0.00010435529624808204
Ridge: 0.7037488127556343
ElasticNet: 0.45599972417723134
Logistic: 0.7194072657743786
Random Forest: 0.8882401147305881
```

Best model is Random Forest

2015

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_2015.csv")
df
```

Out[2]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	
0	2015-10-01 01:00:00	NaN	0.8	NaN	NaN	90.0	82.0	NaN	NaN	NaN	10.0	NaN	NaN	28
1	2015-10-01 01:00:00	2.0	0.8	1.6	0.33	40.0	95.0	4.0	37.0	24.0	12.0	1.83	8.3	28
2	2015-10-01 01:00:00	3.1	NaN	1.8	NaN	29.0	97.0	NaN	NaN	NaN	NaN	NaN	7.1	28
3	2015-10-01 01:00:00	NaN	0.6	NaN	NaN	30.0	103.0	2.0	NaN	NaN	NaN	NaN	NaN	28
4	2015-10-01 01:00:00	NaN	NaN	NaN	NaN	95.0	96.0	2.0	NaN	NaN	9.0	NaN	NaN	28
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
210091	2015-08-01 00:00:00	NaN	0.2	NaN	NaN	11.0	33.0	53.0	NaN	NaN	NaN	NaN	NaN	28
210092	2015-08-01 00:00:00	NaN	0.2	NaN	NaN	1.0	5.0	NaN	26.0	NaN	10.0	NaN	NaN	28
210093	2015-08-01 00:00:00	NaN	NaN	NaN	NaN	1.0	7.0	74.0	NaN	NaN	NaN	NaN	NaN	28
210094	2015-08-01 00:00:00	NaN	NaN	NaN	NaN	3.0	7.0	65.0	NaN	NaN	NaN	NaN	NaN	28
210095	2015-08-01 00:00:00	NaN	NaN	NaN	NaN	1.0	9.0	54.0	29.0	NaN	NaN	NaN	NaN	28

210096 rows × 14 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210096 entries, 0 to 210095
Data columns (total 14 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     210096 non-null  object 
 1   BEN      51039 non-null   float64
 2   CO       86827 non-null   float64
 3   EBE      50962 non-null   float64
 4   NMHC     25756 non-null   float64
 5   NO       208805 non-null  float64
 6   NO_2     208805 non-null  float64
 7   O_3      121574 non-null  float64
 8   PM10     102745 non-null  float64
 9   PM25     48798 non-null   float64
 10  SO_2     86898 non-null   float64
 11  TCH      25756 non-null   float64
 12  TOL      50626 non-null   float64
 13  station  210096 non-null  int64  
dtypes: float64(12), int64(1), object(1)
memory usage: 22.4+ MB
```

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

Out[4]:

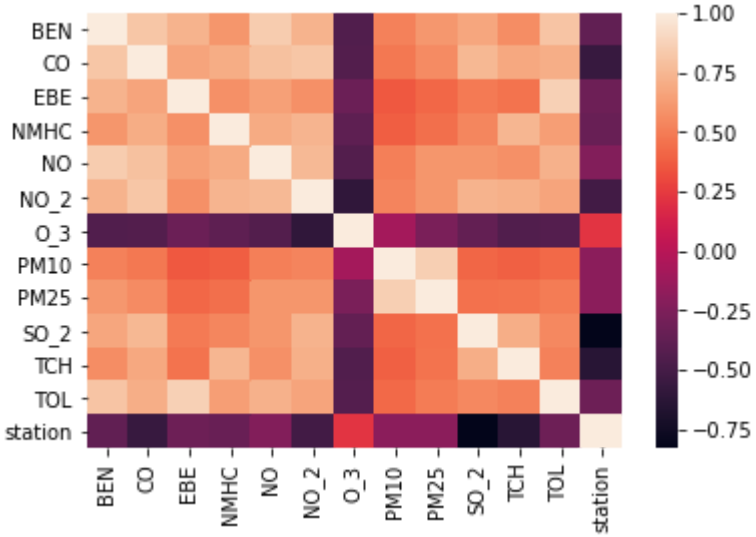
	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	
1	2015-10-01 01:00:00	2.0	0.8	1.6	0.33	40.0	95.0	4.0	37.0	24.0	12.0	1.83	8.3	28
6	2015-10-01 01:00:00	0.5	0.3	0.3	0.12	6.0	83.0	1.0	19.0	12.0	3.0	1.29	4.8	28
25	2015-10-01 02:00:00	1.6	0.7	1.3	0.38	81.0	105.0	4.0	36.0	19.0	13.0	1.93	6.9	28
30	2015-10-01 02:00:00	0.4	0.3	0.3	0.11	5.0	72.0	2.0	16.0	10.0	2.0	1.27	7.8	28
49	2015-10-01 03:00:00	2.2	0.8	1.8	0.41	111.0	104.0	4.0	35.0	20.0	14.0	2.05	13.9	28
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
210030	2015-07-31 22:00:00	0.1	0.1	0.1	0.06	1.0	10.0	69.0	10.0	3.0	2.0	1.18	0.2	28
210049	2015-07-31 23:00:00	0.4	0.3	0.1	0.12	3.0	28.0	56.0	15.0	7.0	12.0	1.45	1.2	28
210054	2015-07-31 23:00:00	0.1	0.1	0.1	0.06	1.0	10.0	63.0	5.0	1.0	2.0	1.18	0.2	28
210073	2015-08-01 00:00:00	0.1	0.3	0.1	0.11	2.0	23.0	59.0	5.0	2.0	11.0	1.44	0.6	28
210078	2015-08-01 00:00:00	0.1	0.1	0.1	0.06	1.0	8.0	65.0	7.0	1.0	2.0	1.18	0.4	28

16026 rows × 14 columns

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

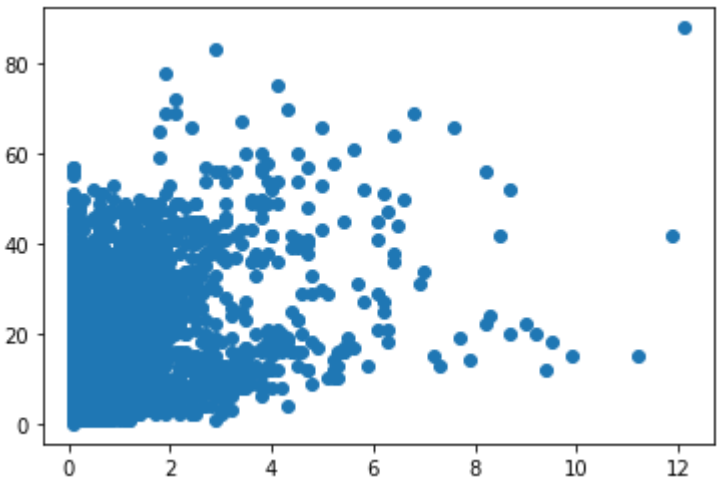
In [6]:
```
sns.heatmap(df1.corr())
```

Out[6]: <AxesSubplot:>



In [7]:
```
plt.plot(df1["EBE"],df1["PM25"],"o")
```

Out[7]: [<matplotlib.lines.Line2D at 0x23a1638ebb0>]



In [8]:
```
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)
```

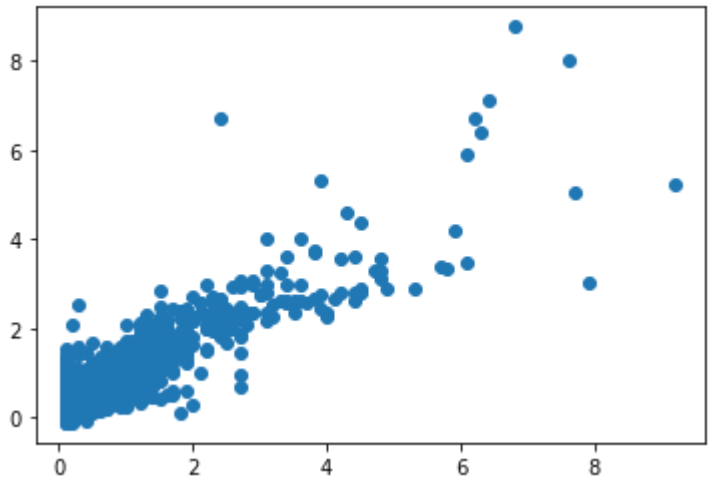
Linear

In [9]:
```
li=LinearRegression()
li.fit(x_train,y_train)
```

Out[9]: LinearRegression()

In [10]:
```
prediction=li.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[10]:
```
<matplotlib.collections.PathCollection at 0x23a16575700>
```



In [11]:
```
lis=li.score(x_test,y_test)
```

In [12]:
```
df1["TCH"].value_counts()
```

Out[12]:
```
1.20    905
1.19    873
1.21    793
1.22    638
1.18    465
       ...
2.79      1
4.46      1
2.48      1
3.43      1
2.63      1
Name: TCH, Length: 184, dtype: int64
```

In [13]:
```
df1.loc[df1["TCH"]<1.40,"TCH"]=1
df1.loc[df1["TCH"]>1.40,"TCH"]=2
df1["TCH"].value_counts()
```

Out[13]:
```
2.0    8290
1.0    7736
Name: TCH, dtype: int64
```

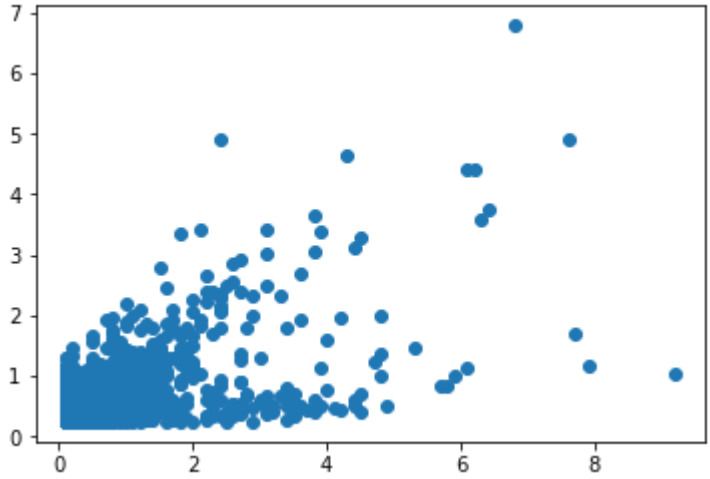
Lasso

In [14]:
```
la=Lasso(alpha=5)
la.fit(x_train,y_train)
```

Out[14]:
```
Lasso(alpha=5)
```

In [15]:
```
prediction1=la.predict(x_test)
plt.scatter(y_test,prediction1)
```

Out[15]: <matplotlib.collections.PathCollection at 0x23a165d8e50>



In [16]:
```
las=la.score(x_test,y_test)
```

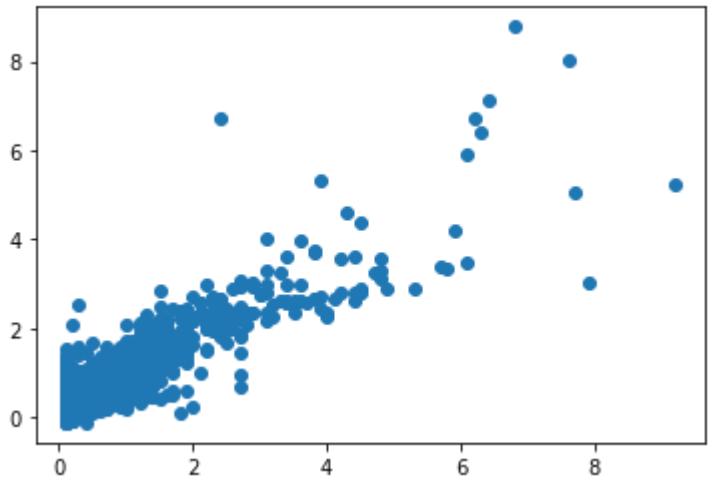
Ridge

In [17]:
```
rr=Ridge(alpha=1)
rr.fit(x_train,y_train)
```

Out[17]: Ridge(alpha=1)

In [18]:
```
prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[18]: <matplotlib.collections.PathCollection at 0x23a163df9a0>



In [19]:
```
rrs=rr.score(x_test,y_test)
```

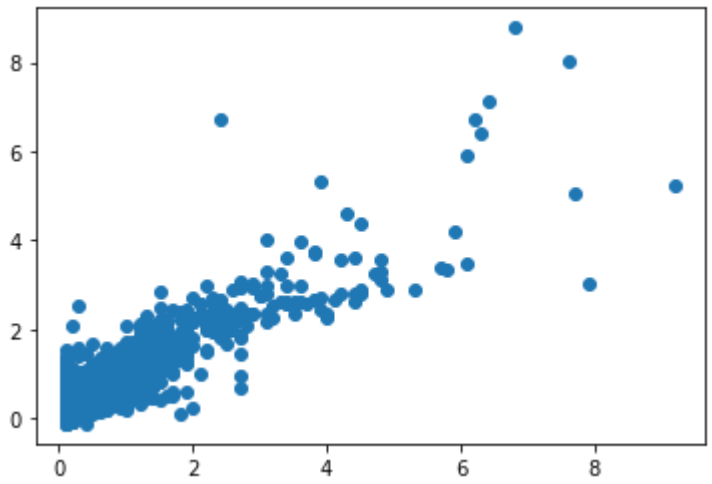
ElasticNet

In [20]:
```
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[20]:
```
ElasticNet()
```

In [21]:
```
prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[21]:
```
<matplotlib.collections.PathCollection at 0x23a16657ee0>
```



In [22]:
```
ens=en.score(x_test,y_test)
```

In [23]:
```
print(rr.score(x_test,y_test))
rr.score(x_train,y_train)
```

```
0.8030115171965686
```

Out[23]:
```
0.7576572353945755
```

Logistic

In [24]:
```
g={"TCH":{1.0:"Low",2.0:"High"}}
df1=df1.replace(g)
df1["TCH"].value_counts()
```

Out[24]:
```
High    8290
Low     7736
Name: TCH, dtype: int64
```

In [25]:
```
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 [26]:
```
lo=LogisticRegression()
lo.fit(x_train,y_train)
```

Out[26]:
```
LogisticRegression()
```

In [27]:
```
prediction3=lo.predict(x_test)
plt.scatter(y_test,prediction3)
```

Out[27]:
```
<matplotlib.collections.PathCollection at 0x23a16414a60>
```

In [28]:
```
los=lo.score(x_test,y_test)
```

Random Forest

In [29]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
```

In [30]:
```
g1={"TCH":{"Low":1.0,"High":2.0}}
df1=df1.replace(g1)
```

In [31]:
```
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 [32]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[32]:
```
RandomForestClassifier()
```

In [33]:
```
parameter={
    'max_depth':[1,2,4,5,6],
    'min_samples_leaf':[5,10,15,20,25],
    'n_estimators':[10,20,30,40,50]
}
```

In [34]:
```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accu
grid_search.fit(x_train,y_train)
```

Out[34]:
```
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 [35]:
```
rfcs=grid_search.best_score_
```

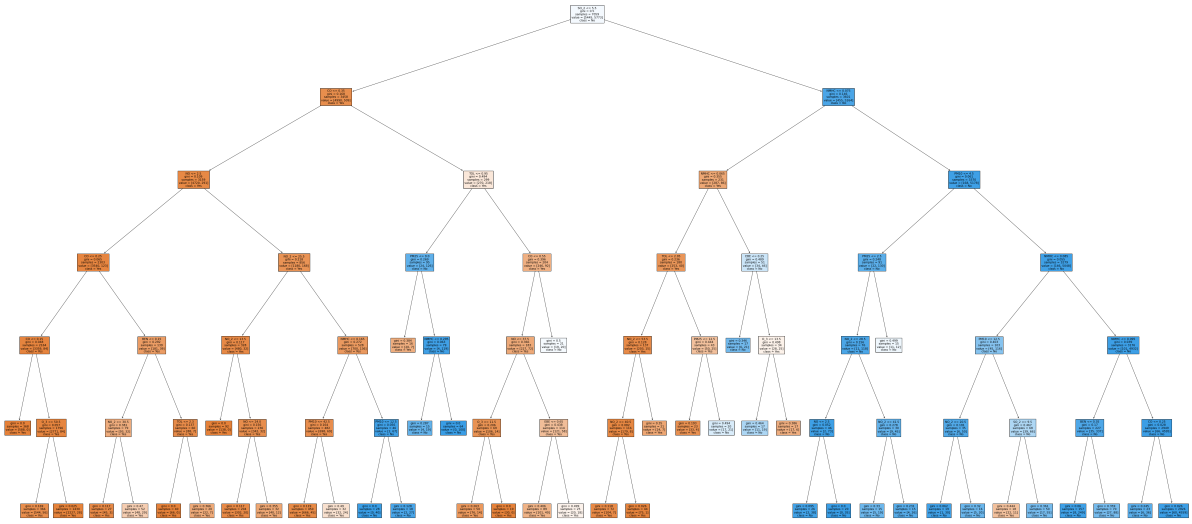
In [36]:
```
rfc_best=grid_search.best_estimator_
```

In [37]:
```
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[37]:
```
[Text(2200.56338028169, 2019.0857142857144, 'SO_2 <= 5.5\ngini = 0.5\nsamples
= 7059\nvalue = [5445, 5773]\nclass = No'),
 Text(1257.4647887323945, 1708.457142857143, 'CO <= 0.35\ngini = 0.168\nsampl
es = 3458\nvalue = [4990, 509]\nclass = Yes'),
 Text(723.0422535211268, 1397.8285714285716, 'NO <= 2.5\ngini = 0.109\nsample
s = 3159\nvalue = [4720, 291]\nclass = Yes'),
 Text(345.80281690140845, 1087.2, 'CO <= 0.25\ngini = 0.065\nsamples = 2303\n
value = [3540, 123]\nclass = Yes'),
 Text(125.74647887323944, 776.5714285714287, 'CO <= 0.15\ngini = 0.048\nsampl
es = 2164\nvalue = [3359, 84]\nclass = Yes'),
 Text(62.87323943661972, 465.9428571428573, 'gini = 0.0\nsamples = 368\nvalue
= [588, 0]\nclass = Yes'),
 Text(188.61971830985917, 465.9428571428573, 'O_3 <= 50.5\ngini = 0.057\nsamp
les = 1796\nvalue = [2771, 84]\nclass = Yes'),
 Text(125.74647887323944, 155.3142857142857, 'gini = 0.169\nsamples = 366\nva
lue = [544, 56]\nclass = Yes'),
 Text(251.49295774647888, 155.3142857142857, 'gini = 0.025\nsamples = 1430\nv
alue = [2227, 28]\nclass = Yes'),
 Text(565.8591549295775, 776.5714285714287, 'BEN <= 0.15\ngini = 0.292\nsampl
es = 139\nvalue = [181, 39]\nclass = Yes'),
 Text(440.11267605633805, 465.9428571428573, 'NO_2 <= 30.5\ngini = 0.381\nsam
ples = 79\nvalue = [93, 32]\nclass = Yes'),
 Text(377.23943661971833, 155.3142857142857, 'gini = 0.117\nsamples = 27\nval
ue = [45, 3]\nclass = Yes'),
 Text(502.98591549295776, 155.3142857142857, 'gini = 0.47\nsamples = 52\nvalu
e = [48, 29]\nclass = Yes'),
 Text(691.6056338028169, 465.9428571428573, 'TOL <= 2.3\ngini = 0.137\nsample
s = 60\nvalue = [88, 7]\nclass = Yes'),
 Text(628.7323943661972, 155.3142857142857, 'gini = 0.0\nsamples = 40\nvalue
= [66, 0]\nclass = Yes'),
 Text(754.4788732394367, 155.3142857142857, 'gini = 0.366\nsamples = 20\nvalu
e = [22, 7]\nclass = Yes'),
 Text(1100.281690140845, 1087.2, 'NO_2 <= 25.5\ngini = 0.218\nsamples = 856\n
value = [1180, 168]\nclass = Yes'),
 Text(880.2253521126761, 776.5714285714287, 'NO_2 <= 13.5\ngini = 0.117\nsamp
les = 328\nvalue = [480, 32]\nclass = Yes'),
 Text(817.3521126760563, 465.9428571428573, 'gini = 0.0\nsamples = 92\nvalue
= [138, 0]\nclass = Yes'),
 Text(943.0985915492957, 465.9428571428573, 'NO <= 24.0\ngini = 0.156\nsample
s = 236\nvalue = [342, 32]\nclass = Yes'),
 Text(880.2253521126761, 155.3142857142857, 'gini = 0.117\nsamples = 204\nval
ue = [302, 20]\nclass = Yes'),
 Text(1005.9718309859155, 155.3142857142857, 'gini = 0.355\nsamples = 32\nval
ue = [40, 12]\nclass = Yes'),
 Text(1320.338028169014, 776.5714285714287, 'NMHC <= 0.165\ngini = 0.272\nsam
ples = 528\nvalue = [700, 136]\nclass = Yes'),
 Text(1194.5915492957747, 465.9428571428573, 'PM10 <= 42.5\ngini = 0.164\nsam
ples = 482\nvalue = [698, 69]\nclass = Yes'),
 Text(1131.718309859155, 155.3142857142857, 'gini = 0.119\nsamples = 450\nval
ue = [666, 45]\nclass = Yes'),
 Text(1257.4647887323945, 155.3142857142857, 'gini = 0.49\nsamples = 32\nvalu
e = [32, 24]\nclass = Yes'),
 Text(1446.0845070422536, 465.9428571428573, 'PM10 <= 12.5\ngini = 0.056\nsam
ples = 46\nvalue = [2, 67]\nclass = No'),
 Text(1383.2112676056338, 155.3142857142857, 'gini = 0.0\nsamples = 28\nvalue
= [0, 40]\nclass = No'),
 Text(1508.9577464788733, 155.3142857142857, 'gini = 0.128\nsamples = 18\nval
```

```
ue = [2, 27]\nclass = No'),
 Text(1791.887323943662, 1397.8285714285716, 'TOL <= 0.95\ngini = 0.494\nsamp
les = 299\nvalue = [270, 218]\nclass = Yes'),
 Text(1571.830985915493, 1087.2, 'PM25 <= 9.0\ngini = 0.269\nsamples = 95\nva
lue = [24, 126]\nclass = No'),
 Text(1508.9577464788733, 776.5714285714287, 'gini = 0.384\nsamples = 16\nval
ue = [20, 7]\nclass = Yes'),
 Text(1634.7042253521126, 776.5714285714287, 'NMHC <= 0.295\ngini = 0.063\nsa
mples = 79\nvalue = [4, 119]\nclass = No'),
 Text(1571.830985915493, 465.9428571428573, 'gini = 0.287\nsamples = 15\nvalu
e = [4, 19]\nclass = No'),
 Text(1697.5774647887324, 465.9428571428573, 'gini = 0.0\nsamples = 64\nvalue
= [0, 100]\nclass = No'),
 Text(2011.943661971831, 1087.2, 'CO <= 0.55\ngini = 0.396\nsamples = 204\nva
lue = [246, 92]\nclass = Yes'),
 Text(1949.0704225352113, 776.5714285714287, 'NO <= 37.5\ngini = 0.366\nsampl
es = 183\nvalue = [227, 72]\nclass = Yes'),
 Text(1823.323943661972, 465.9428571428573, 'O_3 <= 11.5\ngini = 0.206\nsampl
es = 69\nvalue = [106, 14]\nclass = Yes'),
 Text(1760.4507042253522, 155.3142857142857, 'gini = 0.263\nsamples = 50\nval
ue = [76, 14]\nclass = Yes'),
 Text(1886.1971830985915, 155.3142857142857, 'gini = 0.0\nsamples = 19\nvalue
= [30, 0]\nclass = Yes'),
 Text(2074.8169014084506, 465.9428571428573, 'EBE <= 0.65\ngini = 0.438\nsamp
les = 114\nvalue = [121, 58]\nclass = Yes'),
 Text(2011.943661971831, 155.3142857142857, 'gini = 0.406\nsamples = 89\nvalu
e = [101, 40]\nclass = Yes'),
 Text(2137.6901408450703, 155.3142857142857, 'gini = 0.499\nsamples = 25\nval
ue = [20, 18]\nclass = Yes'),
 Text(2074.8169014084506, 776.5714285714287, 'gini = 0.5\nsamples = 21\nvalue
= [19, 20]\nclass = No'),
 Text(3143.661971830986, 1708.457142857143, 'NMHC <= 0.075\ngini = 0.146\nsam
ples = 3601\nvalue = [455, 5264]\nclass = No'),
 Text(2672.112676056338, 1397.8285714285716, 'NMHC <= 0.065\ngini = 0.355\nsa
mples = 231\nvalue = [287, 86]\nclass = Yes'),
 Text(2514.929577464789, 1087.2, 'TOL <= 2.05\ngini = 0.236\nsamples = 180\nv
alue = [253, 40]\nclass = Yes'),
 Text(2389.1830985915494, 776.5714285714287, 'NO_2 <= 53.5\ngini = 0.128\nsam
ples = 137\nvalue = [203, 15]\nclass = Yes'),
 Text(2326.3098591549297, 465.9428571428573, 'NO_2 <= 40.5\ngini = 0.082\nsam
ples = 116\nvalue = [179, 8]\nclass = Yes'),
 Text(2263.43661971831, 155.3142857142857, 'gini = 0.118\nsamples = 72\nvalue
= [104, 7]\nclass = Yes'),
 Text(2389.1830985915494, 155.3142857142857, 'gini = 0.026\nsamples = 44\nval
ue = [75, 1]\nclass = Yes'),
 Text(2452.056338028169, 465.9428571428573, 'gini = 0.35\nsamples = 21\nvalue
= [24, 7]\nclass = Yes'),
 Text(2640.676056338028, 776.5714285714287, 'PM25 <= 12.5\ngini = 0.444\nsamp
les = 43\nvalue = [50, 25]\nclass = Yes'),
 Text(2577.8028169014087, 465.9428571428573, 'gini = 0.193\nsamples = 23\nval
ue = [33, 4]\nclass = Yes'),
 Text(2703.549295774648, 465.9428571428573, 'gini = 0.494\nsamples = 20\nvalu
e = [17, 21]\nclass = No'),
 Text(2829.2957746478874, 1087.2, 'EBE <= 0.25\ngini = 0.489\nsamples = 51\nv
alue = [34, 46]\nclass = No'),
 Text(2766.4225352112676, 776.5714285714287, 'gini = 0.346\nsamples = 17\nval
ue = [6, 21]\nclass = No'),
```

```
 Text(2892.169014084507, 776.5714285714287, 'O_3 <= 13.5\ngini = 0.498\nsampl
es = 34\nvalue = [28, 25]\nclass = Yes'),
 Text(2829.2957746478874, 465.9428571428573, 'gini = 0.464\nsamples = 17\nval
ue = [11, 19]\nclass = No'),
 Text(2955.042253521127, 465.9428571428573, 'gini = 0.386\nsamples = 17\nvalu
e = [17, 6]\nclass = Yes'),
 Text(3615.211267605634, 1397.8285714285716, 'PM10 <= 4.5\ngini = 0.061\nsamp
les = 3370\nvalue = [168, 5178]\nclass = No'),
 Text(3269.4084507042253, 1087.2, 'PM25 <= 2.5\ngini = 0.248\nsamples = 91\nv
alue = [22, 130]\nclass = No'),
 Text(3206.5352112676055, 776.5714285714287, 'NO_2 <= 28.5\ngini = 0.156\nsam
ples = 76\nvalue = [11, 118]\nclass = No'),
 Text(3080.7887323943664, 465.9428571428573, 'NO <= 2.5\ngini = 0.052\nsample
s = 46\nvalue = [2, 73]\nclass = No'),
 Text(3017.9154929577467, 155.3142857142857, 'gini = 0.095\nsamples = 26\nval
ue = [2, 38]\nclass = No'),
 Text(3143.661971830986, 155.3142857142857, 'gini = 0.0\nsamples = 20\nvalue
= [0, 35]\nclass = No'),
 Text(3332.281690140845, 465.9428571428573, 'NO_2 <= 41.5\ngini = 0.278\nsamp
les = 30\nvalue = [9, 45]\nclass = No'),
 Text(3269.4084507042253, 155.3142857142857, 'gini = 0.33\nsamples = 15\nvalu
e = [5, 19]\nclass = No'),
 Text(3395.154929577465, 155.3142857142857, 'gini = 0.231\nsamples = 15\nvalu
e = [4, 26]\nclass = No'),
 Text(3332.281690140845, 776.5714285714287, 'gini = 0.499\nsamples = 15\nvalu
e = [11, 12]\nclass = No'),
 Text(3961.0140845070423, 1087.2, 'NMHC <= 0.085\ngini = 0.055\nsamples = 327
9\nvalue = [146, 5048]\nclass = No'),
 Text(3709.5211267605637, 776.5714285714287, 'PM10 <= 12.5\ngini = 0.403\nsam
ples = 103\nvalue = [45, 116]\nclass = No'),
 Text(3583.774647887324, 465.9428571428573, 'NO_2 <= 20.5\ngini = 0.191\nsamp
les = 35\nvalue = [6, 50]\nclass = No'),
 Text(3520.9014084507044, 155.3142857142857, 'gini = 0.062\nsamples = 19\nval
ue = [1, 30]\nclass = No'),
 Text(3646.647887323944, 155.3142857142857, 'gini = 0.32\nsamples = 16\nvalue
= [5, 20]\nclass = No'),
 Text(3835.2676056338028, 465.9428571428573, 'SO_2 <= 9.5\ngini = 0.467\nsamp
les = 68\nvalue = [39, 66]\nclass = No'),
 Text(3772.394366197183, 155.3142857142857, 'gini = 0.444\nsamples = 18\nvalu
e = [22, 11]\nclass = Yes'),
 Text(3898.1408450704225, 155.3142857142857, 'gini = 0.361\nsamples = 50\nval
ue = [17, 55]\nclass = No'),
 Text(4212.507042253521, 776.5714285714287, 'NMHC <= 0.095\ngini = 0.039\nsam
ples = 3176\nvalue = [101, 4932]\nclass = No'),
 Text(4086.760563380282, 465.9428571428573, 'BEN <= 0.35\ngini = 0.17\nsample
s = 227\nvalue = [35, 337]\nclass = No'),
 Text(4023.887323943662, 155.3142857142857, 'gini = 0.06\nsamples = 157\nvalu
e = [8, 249]\nclass = No'),
 Text(4149.633802816901, 155.3142857142857, 'gini = 0.359\nsamples = 70\nvalu
e = [27, 88]\nclass = No'),
 Text(4338.2535211267605, 465.9428571428573, 'CO <= 0.25\ngini = 0.028\nsampl
es = 2949\nvalue = [66, 4595]\nclass = No'),
 Text(4275.380281690141, 155.3142857142857, 'gini = 0.245\nsamples = 23\nvalu
e = [6, 36]\nclass = No'),
 Text(4401.12676056338, 155.3142857142857, 'gini = 0.026\nsamples = 2926\nval
ue = [60, 4559]\nclass = No')]
```



In [38]:
```
print("Linear:",lis)
print("Lasso:",las)
print("Ridge:",rrs)
print("ElasticNet:",ens)
print("Logistic:",los)
print("Random Forest:",rfcs)
```

```
Linear: 0.8029944802406465
Lasso: 0.4077775855828065
Ridge: 0.8030115171965686
ElasticNet: 0.7001824428916614
Logistic: 0.5187188019966722
Random Forest: 0.9561419147798181
```

Best Model is Random Forest

2016

In [39]:
```
df2=pd.read_csv("madrid_2016.csv")
df2
```

Out[39]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	
0	2016-11-01 01:00:00	NaN	0.7	NaN	NaN	153.0	77.0	NaN	NaN	NaN	7.0	NaN	NaN	2
1	2016-11-01 01:00:00	3.1	1.1	2.0	0.53	260.0	144.0	4.0	46.0	24.0	18.0	2.44	14.4	2
2	2016-11-01 01:00:00	5.9	NaN	7.5	NaN	297.0	139.0	NaN	NaN	NaN	NaN	NaN	26.0	2
3	2016-11-01 01:00:00	NaN	1.0	NaN	NaN	154.0	113.0	2.0	NaN	NaN	NaN	NaN	NaN	2
4	2016-11-01 01:00:00	NaN	NaN	NaN	NaN	275.0	127.0	2.0	NaN	NaN	18.0	NaN	NaN	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
209491	2016-07-01 00:00:00	NaN	0.2	NaN	NaN	2.0	29.0	73.0	NaN	NaN	NaN	NaN	NaN	2
209492	2016-07-01 00:00:00	NaN	0.3	NaN	NaN	1.0	29.0	NaN	36.0	NaN	5.0	NaN	NaN	2
209493	2016-07-01 00:00:00	NaN	NaN	NaN	NaN	1.0	19.0	71.0	NaN	NaN	NaN	NaN	NaN	2
209494	2016-07-01 00:00:00	NaN	NaN	NaN	NaN	6.0	17.0	85.0	NaN	NaN	NaN	NaN	NaN	2
209495	2016-07-01 00:00:00	NaN	NaN	NaN	NaN	2.0	46.0	61.0	34.0	NaN	NaN	NaN	NaN	2

209496 rows × 14 columns

In [40]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209496 entries, 0 to 209495
Data columns (total 14 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     209496 non-null  object 
 1   BEN      50755 non-null   float64
 2   CO       85999 non-null   float64
 3   EBE      50335 non-null   float64
 4   NMHC     25970 non-null   float64
 5   NO       208614 non-null  float64
 6   NO_2     208614 non-null  float64
 7   O_3      121197 non-null  float64
 8   PM10     102892 non-null  float64
 9   PM25     52165 non-null   float64
 10  SO_2     86023 non-null   float64
 11  TCH      25970 non-null   float64
 12  TOL      50662 non-null   float64
 13  station  209496 non-null  int64  
dtypes: float64(12), int64(1), object(1)
memory usage: 22.4+ MB
```

In [41]:
```
df3=df2.dropna()
df3
```

Out[41]:

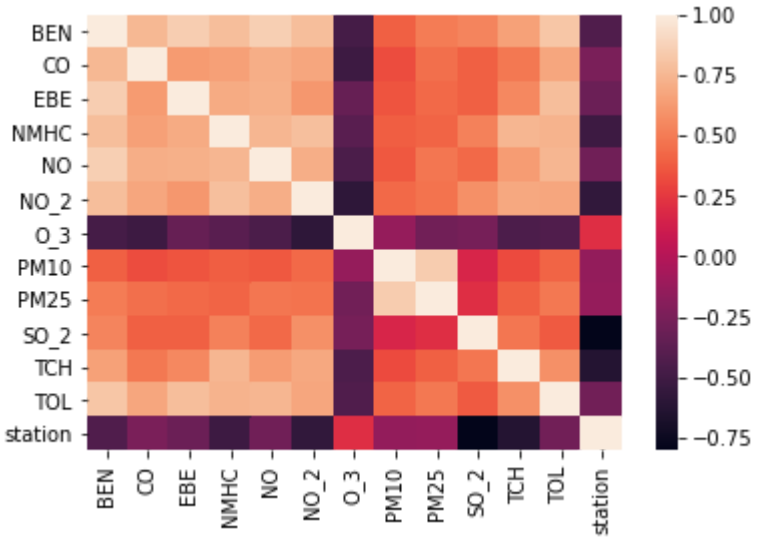
	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	
1	2016-11-01 01:00:00	3.1	1.1	2.0	0.53	260.0	144.0	4.0	46.0	24.0	18.0	2.44	14.4	28
6	2016-11-01 01:00:00	0.7	0.8	0.4	0.13	57.0	66.0	3.0	23.0	15.0	4.0	1.35	5.0	28
25	2016-11-01 02:00:00	2.7	1.0	2.1	0.40	139.0	114.0	4.0	37.0	21.0	14.0	2.30	15.0	28
30	2016-11-01 02:00:00	0.7	0.7	0.4	0.13	48.0	59.0	3.0	23.0	15.0	3.0	1.35	5.0	28
49	2016-11-01 03:00:00	1.7	0.8	1.4	0.25	53.0	90.0	4.0	31.0	19.0	10.0	1.95	10.7	28
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
209430	2016-06-30 22:00:00	0.1	0.2	0.1	0.02	1.0	5.0	97.0	19.0	12.0	2.0	1.15	0.2	28
209449	2016-06-30 23:00:00	0.6	0.4	0.3	0.15	14.0	63.0	54.0	29.0	13.0	16.0	1.48	1.9	28
209454	2016-06-30 23:00:00	0.1	0.2	0.1	0.02	1.0	7.0	91.0	16.0	9.0	2.0	1.15	0.3	28
209473	2016-07-01 00:00:00	0.6	0.4	0.3	0.16	11.0	68.0	45.0	24.0	14.0	16.0	1.50	1.9	28
209478	2016-07-01 00:00:00	0.1	0.2	0.1	0.02	1.0	6.0	89.0	16.0	9.0	2.0	1.15	0.2	28

16932 rows × 14 columns

In [42]:
```
df3=df3.drop(["date"],axis=1)
```

In [43]:
```
sns.heatmap(df3.corr())
```

Out[43]: <AxesSubplot:>



In [44]:
```
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

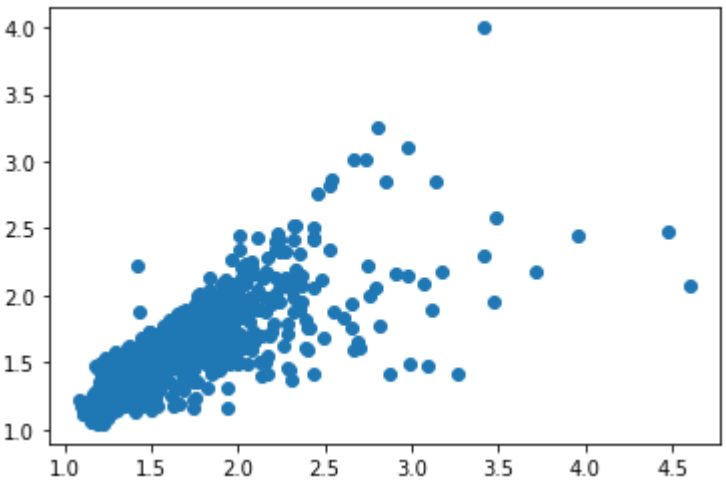
In [45]:
```
li=LinearRegression()
li.fit(x_train,y_train)
```

Out[45]: LinearRegression()

In [ ]:
```

```

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

Out[46]: <matplotlib.collections.PathCollection at 0x23a1724cd60>



In [47]:
```
lis=li.score(x_test,y_test)
```

In [48]:
```
df3["TCH"].value_counts()
```

Out[48]:
```
1.16    757
1.18    701
1.17    683
1.19    618
1.15    577
       ...
4.82      1
2.78      1
3.59      1
3.10      1
4.07      1
Name: TCH, Length: 217, dtype: int64
```

In [49]:
```
df3.loc[df3["TCH"]<1.40,"TCH"]=1
df3.loc[df3["TCH"]>1.40,"TCH"]=2
df3["TCH"].value_counts()
```

Out[49]:
```
1.0    10002
2.0     6930
Name: TCH, dtype: int64
```

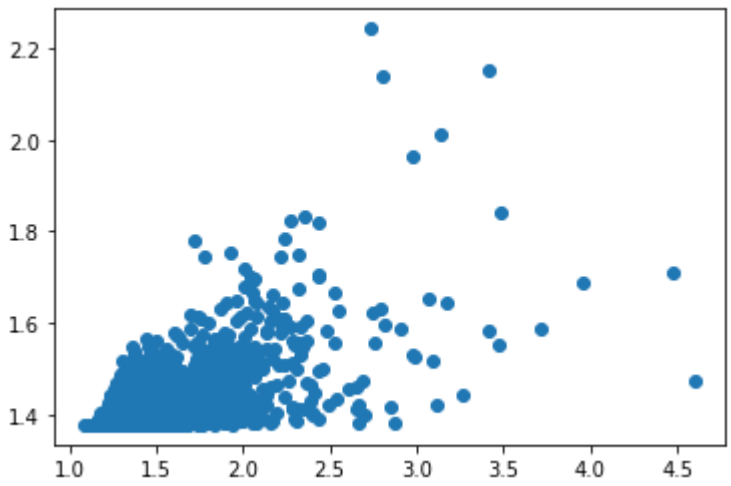
Lasso

In [50]:
```
la=Lasso(alpha=5)
la.fit(x_train,y_train)
```

Out[50]:
```
Lasso(alpha=5)
```

In [51]:
```
prediction1=la.predict(x_test)
plt.scatter(y_test,prediction1)
```

Out[51]:
```
<matplotlib.collections.PathCollection at 0x23a16fa9e80>
```



In [52]:
```
las=la.score(x_test,y_test)
```

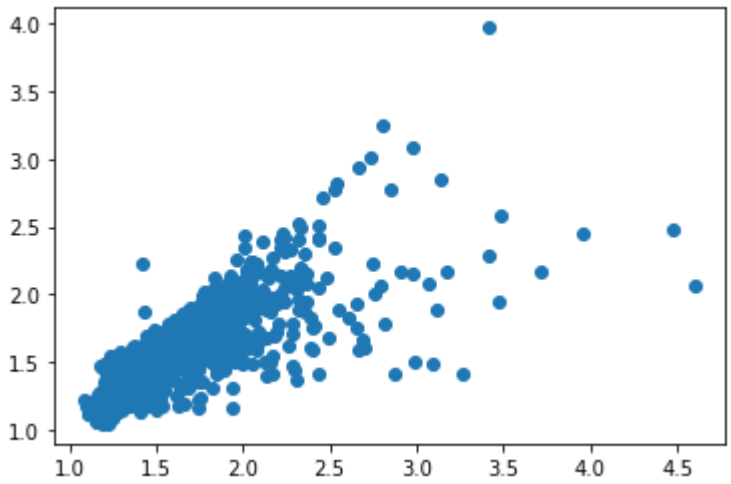
Ridge

In [53]:
```
rr=Ridge(alpha=1)
rr.fit(x_train,y_train)
```

Out[53]:
```
Ridge(alpha=1)
```

In [54]:
```
prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[54]:
```
<matplotlib.collections.PathCollection at 0x23a16ffef40>
```



In [55]:
```
rrs=rr.score(x_test,y_test)
```

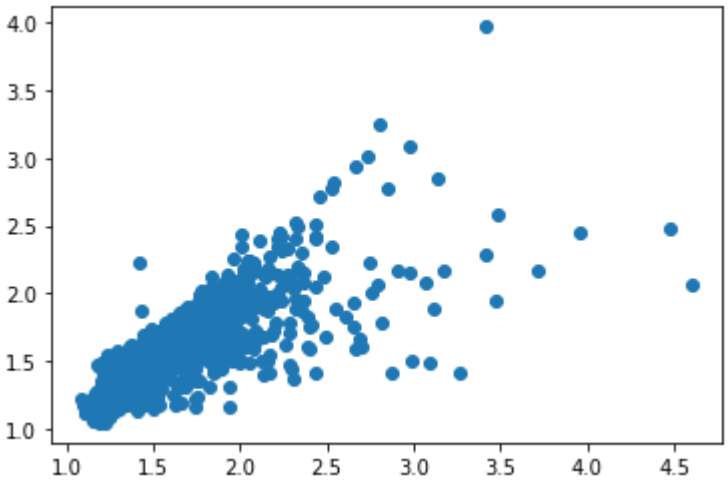
ElasticNet

In [56]:
```
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[56]:
```
ElasticNet()
```

In [57]:
```
prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[57]:
```
<matplotlib.collections.PathCollection at 0x23a17022f40>
```



In [58]:
```
ens=en.score(x_test,y_test)
```

In [59]:
```
print(rr.score(x_test,y_test))
rr.score(x_train,y_train)
```

```
0.7500531919104193
```

Out[59]:
```
0.7540860619941899
```

Logistic

In [60]:
```
g={"TCH":{1.0:"Low",2.0:"High"}}
df3=df3.replace(g)
df3["TCH"].value_counts()
```

Out[60]:
```
Low     10002
High     6930
Name: TCH, dtype: int64
```

In [61]:
```
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 [62]:
```
lo=LogisticRegression()
lo.fit(x_train,y_train)
```

Out[62]:
```
LogisticRegression()
```

In [63]:
```
prediction3=lo.predict(x_test)
plt.scatter(y_test,prediction3)
```

Out[63]:
```
<matplotlib.collections.PathCollection at 0x23a170c0ca0>
```

In [64]:
```
los=lo.score(x_test,y_test)
```

Random Forest

In [65]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
```

In [66]:
```
g1={"TCH":{"Low":1.0,"High":2.0}}
df3=df3.replace(g1)
```

In [67]:
```
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 [68]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[68]:
```
RandomForestClassifier()
```

In [69]:
```
parameter={
    'max_depth':[1,2,4,5,6],
    'min_samples_leaf':[5,10,15,20,25],
    'n_estimators':[10,20,30,40,50]
}
```

In [70]:
```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accu
grid_search.fit(x_train,y_train)
```

Out[70]:
```
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 [71]:
```
rfcs=grid_search.best_score_
```

In [72]:
```
rfc_best=grid_search.best_estimator_
```

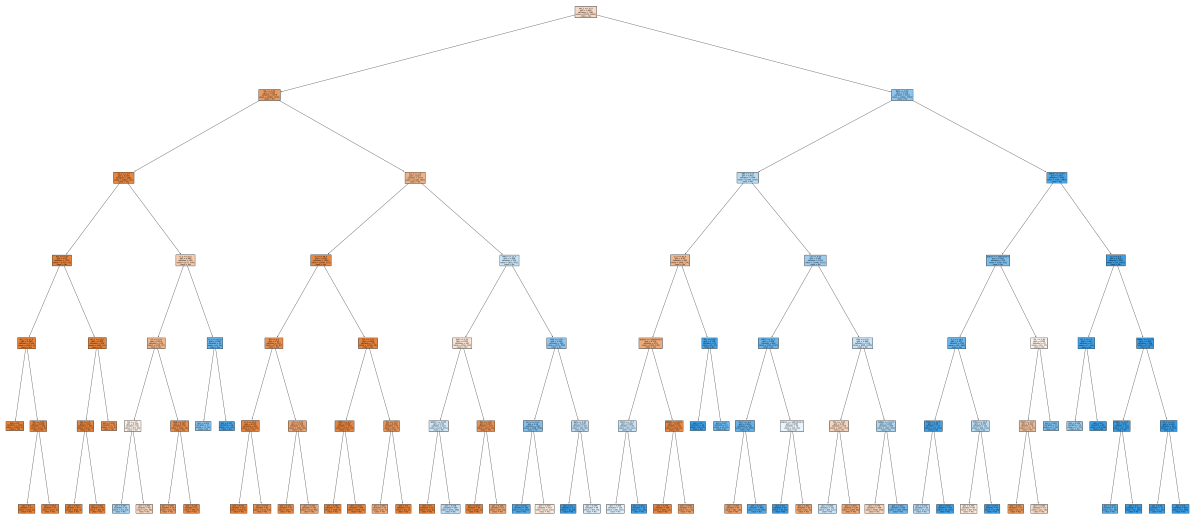
In [73]:
```
from sklearn.tree import plot_tree

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

Out[73]:
```
[Text(2187.8019801980195, 2019.0857142857144, 'NO_2 <= 37.5\ngini = 0.484\nsa
mples = 7481\nvalue = [6970, 4882]\nclass = Yes'),
 Text(999.980198019802, 1708.457142857143, 'TOL <= 0.55\ngini = 0.283\nsample
s = 4079\nvalue = [5368, 1102]\nclass = Yes'),
 Text(453.02970297029697, 1397.8285714285716, 'SO_2 <= 3.5\ngini = 0.075\nsam
ples = 1882\nvalue = [2887, 117]\nclass = Yes'),
 Text(220.99009900990097, 1087.2, 'TOL <= 0.35\ngini = 0.01\nsamples = 1702\n
value = [2710, 13]\nclass = Yes'),
 Text(88.39603960396039, 776.5714285714287, 'NO_2 <= 11.5\ngini = 0.003\nsamp
les = 1243\nvalue = [2015, 3]\nclass = Yes'),
 Text(44.198019801980195, 465.9428571428573, 'gini = 0.0\nsamples = 1100\nval
ue = [1803, 0]\nclass = Yes'),
 Text(132.59405940594058, 465.9428571428573, 'TOL <= 0.25\ngini = 0.028\nsamp
les = 143\nvalue = [212, 3]\nclass = Yes'),
 Text(88.39603960396039, 155.3142857142857, 'gini = 0.0\nsamples = 67\nvalue
= [105, 0]\nclass = Yes'),
 Text(176.79207920792078, 155.3142857142857, 'gini = 0.053\nsamples = 76\nval
ue = [107, 3]\nclass = Yes'),
 Text(353.58415841584156, 776.5714285714287, 'NO_2 <= 29.5\ngini = 0.028\nsam
ples = 459\nvalue = [695, 10]\nclass = Yes'),
 Text(309.38613861386136, 465.9428571428573, 'NO <= 1.5\ngini = 0.02\nsamples
= 443\nvalue = [671, 7]\nclass = Yes'),
 Text(265.18811881188117, 155.3142857142857, 'gini = 0.014\nsamples = 377\nva
lue = [582, 4]\nclass = Yes'),
 Text(353.58415841584156, 155.3142857142857, 'gini = 0.063\nsamples = 66\nval
ue = [89, 3]\nclass = Yes'),
 Text(397.78217821782175, 465.9428571428573, 'gini = 0.198\nsamples = 16\nval
ue = [24, 3]\nclass = Yes'),
 Text(685.0693069306931, 1087.2, 'O_3 <= 91.5\ngini = 0.466\nsamples = 180\nv
alue = [177, 104]\nclass = Yes'),
 Text(574.5742574257425, 776.5714285714287, 'CO <= 0.25\ngini = 0.414\nsample
s = 149\nvalue = [169, 70]\nclass = Yes'),
 Text(486.17821782178214, 465.9428571428573, 'NO <= 2.5\ngini = 0.498\nsample
s = 82\nvalue = [67, 60]\nclass = Yes'),
 Text(441.98019801980195, 155.3142857142857, 'gini = 0.457\nsamples = 22\nval
ue = [12, 22]\nclass = No'),
 Text(530.3762376237623, 155.3142857142857, 'gini = 0.483\nsamples = 60\nvalu
e = [55, 38]\nclass = Yes'),
 Text(662.9702970297029, 465.9428571428573, 'BEN <= 0.25\ngini = 0.163\nsampl
es = 67\nvalue = [102, 10]\nclass = Yes'),
 Text(618.7722772277227, 155.3142857142857, 'gini = 0.298\nsamples = 16\nvalu
e = [18, 4]\nclass = Yes'),
 Text(707.1683168316831, 155.3142857142857, 'gini = 0.124\nsamples = 51\nvalu
e = [84, 6]\nclass = Yes'),
 Text(795.5643564356435, 776.5714285714287, 'O_3 <= 107.5\ngini = 0.308\nsamp
les = 31\nvalue = [8, 34]\nclass = No'),
 Text(751.3663366336633, 465.9428571428573, 'gini = 0.403\nsamples = 17\nvalu
e = [7, 18]\nclass = No'),
 Text(839.7623762376237, 465.9428571428573, 'gini = 0.111\nsamples = 14\nvalu
e = [1, 16]\nclass = No'),
 Text(1546.930693069307, 1397.8285714285716, 'SO_2 <= 3.5\ngini = 0.407\nsamp
les = 2197\nvalue = [2481, 985]\nclass = Yes'),
 Text(1193.3465346534654, 1087.2, 'O_3 <= 38.5\ngini = 0.108\nsamples = 1269
\nvalue = [1856, 113]\nclass = Yes'),
 Text(1016.5544554455445, 776.5714285714287, 'NO <= 5.5\ngini = 0.19\nsamples
= 512\nvalue = [732, 87]\nclass = Yes'),
 Text(928.1584158415841, 465.9428571428573, 'O_3 <= 32.5\ngini = 0.075\nsampl
```

```
es = 302\nvalue = [471, 19]\nclass = Yes'),
 Text(883.9603960396039, 155.3142857142857, 'gini = 0.027\nsamples = 220\nval
ue = [356, 5]\nclass = Yes'),
 Text(972.3564356435643, 155.3142857142857, 'gini = 0.193\nsamples = 82\nvalu
e = [115, 14]\nclass = Yes'),
 Text(1104.9504950495048, 465.9428571428573, 'CO <= 0.15\ngini = 0.328\nsampl
es = 210\nvalue = [261, 68]\nclass = Yes'),
 Text(1060.7524752475247, 155.3142857142857, 'gini = 0.121\nsamples = 17\nval
ue = [29, 2]\nclass = Yes'),
 Text(1149.148514851485, 155.3142857142857, 'gini = 0.345\nsamples = 193\nval
ue = [232, 66]\nclass = Yes'),
 Text(1370.1386138613861, 776.5714285714287, 'TOL <= 3.35\ngini = 0.044\nsamp
les = 757\nvalue = [1124, 26]\nclass = Yes'),
 Text(1281.7425742574255, 465.9428571428573, 'PM25 <= 4.5\ngini = 0.034\nsamp
les = 724\nvalue = [1077, 19]\nclass = Yes'),
 Text(1237.5445544554455, 155.3142857142857, 'gini = 0.083\nsamples = 145\nva
lue = [198, 9]\nclass = Yes'),
 Text(1325.9405940594058, 155.3142857142857, 'gini = 0.022\nsamples = 579\nva
lue = [879, 10]\nclass = Yes'),
 Text(1458.5346534653463, 465.9428571428573, 'TOL <= 3.75\ngini = 0.226\nsamp
les = 33\nvalue = [47, 7]\nclass = Yes'),
 Text(1414.3366336633662, 155.3142857142857, 'gini = 0.384\nsamples = 15\nval
ue = [20, 7]\nclass = Yes'),
 Text(1502.7326732673266, 155.3142857142857, 'gini = 0.0\nsamples = 18\nvalue
= [27, 0]\nclass = Yes'),
 Text(1900.5148514851485, 1087.2, 'PM10 <= 11.5\ngini = 0.486\nsamples = 928
\nvalue = [625, 872]\nclass = No'),
 Text(1723.7227722772277, 776.5714285714287, 'EBE <= 0.25\ngini = 0.493\nsamp
les = 435\nvalue = [394, 312]\nclass = Yes'),
 Text(1635.326732673267, 465.9428571428573, 'NMHC <= 0.055\ngini = 0.49\nsamp
les = 320\nvalue = [219, 291]\nclass = No'),
 Text(1591.128712871287, 155.3142857142857, 'gini = 0.0\nsamples = 20\nvalue
= [28, 0]\nclass = Yes'),
 Text(1679.5247524752474, 155.3142857142857, 'gini = 0.478\nsamples = 300\nva
lue = [191, 291]\nclass = No'),
 Text(1812.1188118811879, 465.9428571428573, 'TOL <= 1.05\ngini = 0.191\nsamp
les = 115\nvalue = [175, 21]\nclass = Yes'),
 Text(1767.9207920792078, 155.3142857142857, 'gini = 0.019\nsamples = 58\nval
ue = [106, 1]\nclass = Yes'),
 Text(1856.3168316831682, 155.3142857142857, 'gini = 0.348\nsamples = 57\nval
ue = [69, 20]\nclass = Yes'),
 Text(2077.3069306930693, 776.5714285714287, 'EBE <= 0.25\ngini = 0.414\nsamp
les = 493\nvalue = [231, 560]\nclass = No'),
 Text(1988.9108910891086, 465.9428571428573, 'SO_2 <= 12.5\ngini = 0.39\nsamp
les = 386\nvalue = [168, 464]\nclass = No'),
 Text(1944.7128712871286, 155.3142857142857, 'gini = 0.193\nsamples = 243\nva
lue = [44, 362]\nclass = No'),
 Text(2033.108910891089, 155.3142857142857, 'gini = 0.495\nsamples = 143\nval
ue = [124, 102]\nclass = Yes'),
 Text(2165.7029702970294, 465.9428571428573, 'BEN <= 0.35\ngini = 0.478\nsamp
les = 107\nvalue = [63, 96]\nclass = No'),
 Text(2121.5049504950493, 155.3142857142857, 'gini = 0.0\nsamples = 13\nvalue
= [0, 20]\nclass = No'),
 Text(2209.9009900990095, 155.3142857142857, 'gini = 0.496\nsamples = 94\nval
ue = [63, 76]\nclass = No'),
 Text(3375.623762376237, 1708.457142857143, 'BEN <= 0.95\ngini = 0.418\nsampl
es = 3402\nvalue = [1602, 3780]\nclass = No'),
```

```
 Text(2795.524752475247, 1397.8285714285716, 'EBE <= 0.15\ngini = 0.475\nsamp
les = 2394\nvalue = [1456, 2297]\nclass = No'),
 Text(2541.386138613861, 1087.2, 'O_3 <= 81.0\ngini = 0.42\nsamples = 380\nva
lue = [408, 175]\nclass = Yes'),
 Text(2430.891089108911, 776.5714285714287, 'station <= 28079016.0\ngini = 0.
379\nsamples = 355\nvalue = [405, 138]\nclass = Yes'),
 Text(2342.49504950495, 465.9428571428573, 'PM10 <= 17.5\ngini = 0.486\nsampl
es = 134\nvalue = [87, 122]\nclass = No'),
 Text(2298.29702970297, 155.3142857142857, 'gini = 0.496\nsamples = 120\nvalu
e = [86, 102]\nclass = No'),
 Text(2386.6930693069307, 155.3142857142857, 'gini = 0.091\nsamples = 14\nval
ue = [1, 20]\nclass = No'),
 Text(2519.287128712871, 465.9428571428573, 'NMHC <= 0.115\ngini = 0.091\nsam
ples = 221\nvalue = [318, 16]\nclass = Yes'),
 Text(2475.089108910891, 155.3142857142857, 'gini = 0.074\nsamples = 206\nval
ue = [301, 12]\nclass = Yes'),
 Text(2563.485148514851, 155.3142857142857, 'gini = 0.308\nsamples = 15\nvalu
e = [17, 4]\nclass = Yes'),
 Text(2651.8811881188117, 776.5714285714287, 'TOL <= 0.95\ngini = 0.139\nsamp
les = 25\nvalue = [3, 37]\nclass = No'),
 Text(2607.6831683168316, 465.9428571428573, 'gini = 0.0\nsamples = 14\nvalue
= [0, 25]\nclass = No'),
 Text(2696.0792079207918, 465.9428571428573, 'gini = 0.32\nsamples = 11\nvalu
e = [3, 12]\nclass = No'),
 Text(3049.6633663366333, 1087.2, 'CO <= 0.35\ngini = 0.443\nsamples = 2014\n
value = [1048, 2122]\nclass = No'),
 Text(2872.8712871287125, 776.5714285714287, 'PM25 <= 21.5\ngini = 0.309\nsam
ples = 800\nvalue = [242, 1027]\nclass = No'),
 Text(2784.4752475247524, 465.9428571428573, 'SO_2 <= 3.5\ngini = 0.293\nsamp
les = 766\nvalue = [217, 999]\nclass = No'),
 Text(2740.2772277227723, 155.3142857142857, 'gini = 0.305\nsamples = 67\nval
ue = [82, 19]\nclass = Yes'),
 Text(2828.6732673267325, 155.3142857142857, 'gini = 0.213\nsamples = 699\nva
lue = [135, 980]\nclass = No'),
 Text(2961.267326732673, 465.9428571428573, 'station <= 28079016.0\ngini = 0.
498\nsamples = 34\nvalue = [25, 28]\nclass = No'),
 Text(2917.0693069306926, 155.3142857142857, 'gini = 0.0\nsamples = 14\nvalue
= [0, 21]\nclass = No'),
 Text(3005.4653465346532, 155.3142857142857, 'gini = 0.342\nsamples = 20\nval
ue = [25, 7]\nclass = Yes'),
 Text(3226.455445544554, 776.5714285714287, 'EBE <= 0.25\ngini = 0.488\nsampl
es = 1214\nvalue = [806, 1095]\nclass = No'),
 Text(3138.059405940594, 465.9428571428573, 'TOL <= 1.85\ngini = 0.48\nsample
s = 270\nvalue = [257, 171]\nclass = Yes'),
 Text(3093.861386138614, 155.3142857142857, 'gini = 0.488\nsamples = 131\nval
ue = [89, 121]\nclass = No'),
 Text(3182.257425742574, 155.3142857142857, 'gini = 0.354\nsamples = 139\nval
ue = [168, 50]\nclass = Yes'),
 Text(3314.8514851485147, 465.9428571428573, 'NMHC <= 0.125\ngini = 0.468\nsa
mples = 944\nvalue = [549, 924]\nclass = No'),
 Text(3270.653465346534, 155.3142857142857, 'gini = 0.476\nsamples = 396\nval
ue = [366, 235]\nclass = Yes'),
 Text(3359.049504950495, 155.3142857142857, 'gini = 0.332\nsamples = 548\nval
ue = [183, 689]\nclass = No'),
 Text(3955.7227722772273, 1397.8285714285716, 'NMHC <= 0.155\ngini = 0.163\ns
amples = 1008\nvalue = [146, 1483]\nclass = No'),
 Text(3734.7326732673264, 1087.2, 'station <= 28079016.0\ngini = 0.369\nsampl
```

```
es = 345\nvalue = [136, 422]\nclass = No'),
 Text(3580.0396039603957, 776.5714285714287, 'O_3 <= 18.5\ngini = 0.317\nsamp
les = 298\nvalue = [95, 385]\nclass = No'),
 Text(3491.6435643564355, 465.9428571428573, 'PM10 <= 11.5\ngini = 0.08\nsamp
les = 171\nvalue = [11, 252]\nclass = No'),
 Text(3447.4455445544554, 155.3142857142857, 'gini = 0.475\nsamples = 10\nval
ue = [7, 11]\nclass = No'),
 Text(3535.8415841584156, 155.3142857142857, 'gini = 0.032\nsamples = 161\nva
lue = [4, 241]\nclass = No'),
 Text(3668.4356435643563, 465.9428571428573, 'PM10 <= 26.5\ngini = 0.475\nsam
ples = 127\nvalue = [84, 133]\nclass = No'),
 Text(3624.2376237623757, 155.3142857142857, 'gini = 0.495\nsamples = 86\nval
ue = [82, 67]\nclass = Yes'),
 Text(3712.6336633663364, 155.3142857142857, 'gini = 0.057\nsamples = 41\nval
ue = [2, 66]\nclass = No'),
 Text(3889.425742574257, 776.5714285714287, 'TOL <= 5.85\ngini = 0.499\nsampl
es = 47\nvalue = [41, 37]\nclass = Yes'),
 Text(3845.227722772277, 465.9428571428573, 'PM25 <= 19.5\ngini = 0.444\nsamp
les = 30\nvalue = [34, 17]\nclass = Yes'),
 Text(3801.029702970297, 155.3142857142857, 'gini = 0.231\nsamples = 11\nvalu
e = [13, 2]\nclass = Yes'),
 Text(3889.425742574257, 155.3142857142857, 'gini = 0.486\nsamples = 19\nvalu
e = [21, 15]\nclass = Yes'),
 Text(3933.623762376237, 465.9428571428573, 'gini = 0.384\nsamples = 17\nvalu
e = [7, 20]\nclass = No'),
 Text(4176.712871287128, 1087.2, 'O_3 <= 4.5\ngini = 0.018\nsamples = 663\nva
lue = [10, 1061]\nclass = No'),
 Text(4066.217821782178, 776.5714285714287, 'SO_2 <= 7.5\ngini = 0.09\nsample
s = 97\nvalue = [7, 141]\nclass = No'),
 Text(4022.019801980198, 465.9428571428573, 'gini = 0.434\nsamples = 14\nvalu
e = [7, 15]\nclass = No'),
 Text(4110.415841584158, 465.9428571428573, 'gini = 0.0\nsamples = 83\nvalue
= [0, 126]\nclass = No'),
 Text(4287.207920792079, 776.5714285714287, 'NMHC <= 0.175\ngini = 0.006\nsam
ples = 566\nvalue = [3, 920]\nclass = No'),
 Text(4198.811881188119, 465.9428571428573, 'PM10 <= 12.5\ngini = 0.02\nsampl
es = 125\nvalue = [2, 197]\nclass = No'),
 Text(4154.6138613861385, 155.3142857142857, 'gini = 0.278\nsamples = 10\nval
ue = [2, 10]\nclass = No'),
 Text(4243.009900990099, 155.3142857142857, 'gini = 0.0\nsamples = 115\nvalue
= [0, 187]\nclass = No'),
 Text(4375.603960396039, 465.9428571428573, 'SO_2 <= 5.5\ngini = 0.003\nsampl
es = 441\nvalue = [1, 723]\nclass = No'),
 Text(4331.405940594059, 155.3142857142857, 'gini = 0.08\nsamples = 17\nvalue
= [1, 23]\nclass = No'),
 Text(4419.801980198019, 155.3142857142857, 'gini = 0.0\nsamples = 424\nvalue
= [0, 700]\nclass = No')]
```



In [74]:
```
print("Linear:",lis)
print("Lasso:",las)
print("Ridge:",rrs)
print("ElasticNet:",ens)
print("Logistic:",los)
print("Random Forest:",rfcs)
```

```
Linear: 0.7499272637946903
Lasso: 0.19751618918213187
Ridge: 0.7500531919104193
ElasticNet: 0.5703555451049473
Logistic: 0.5938976377952756
Random Forest: 0.9190010124873439
```

Best model is Random Forest

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,Elasti
from sklearn.model_selection import train_test_split
```

In [2]:
```
df=pd.read_csv("C:/Users/user/Downloads/FP1_air/csvs_per_year/csvs_per_year/madrid_2017
df
```

Out[2]:

	date	BEN	CH4	CO	EBE	NMHC	NO	NO_2	NOx	O_3	PM10	PM25	SO_2	TCH	TOL	
0	2017-06-01 01:00:00	NaN	NaN	0.3	NaN	NaN	4.0	38.0	NaN	NaN	NaN	NaN	5.0	NaN	NaN	28
1	2017-06-01 01:00:00	0.6	NaN	0.3	0.4	0.08	3.0	39.0	NaN	71.0	22.0	9.0	7.0	1.4	2.9	28
2	2017-06-01 01:00:00	0.2	NaN	NaN	0.1	NaN	1.0	14.0	NaN	NaN	NaN	NaN	NaN	NaN	0.9	28
3	2017-06-01 01:00:00	NaN	NaN	0.2	NaN	NaN	1.0	9.0	NaN	91.0	NaN	NaN	NaN	NaN	NaN	28
4	2017-06-01 01:00:00	NaN	NaN	NaN	NaN	NaN	1.0	19.0	NaN	69.0	NaN	NaN	2.0	NaN	NaN	28
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
210115	2017-08-01 00:00:00	NaN	NaN	0.2	NaN	NaN	1.0	27.0	NaN	65.0	NaN	NaN	NaN	NaN	NaN	28
210116	2017-08-01 00:00:00	NaN	NaN	0.2	NaN	NaN	1.0	14.0	NaN	NaN	73.0	NaN	7.0	NaN	NaN	28
210117	2017-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	1.0	4.0	NaN	83.0	NaN	NaN	NaN	NaN	NaN	28
210118	2017-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	1.0	11.0	NaN	78.0	NaN	NaN	NaN	NaN	NaN	28
210119	2017-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	1.0	14.0	NaN	77.0	60.0	NaN	NaN	NaN	NaN	28

210120 rows × 16 columns

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210120 entries, 0 to 210119
Data columns (total 16 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     210120 non-null  object 
 1   BEN      50201 non-null   float64
 2   CH4      6410 non-null    float64
 3   CO       87001 non-null   float64
 4   EBE      49973 non-null   float64
 5   NMHC     25472 non-null   float64
 6   NO       209065 non-null  float64
 7   NO_2     209065 non-null  float64
 8   NOx      52818 non-null   float64
 9   O_3      121398 non-null  float64
 10  PM10     104141 non-null  float64
 11  PM25     52023 non-null   float64
 12  SO_2     86803 non-null   float64
 13  TCH      25472 non-null   float64
 14  TOL      50117 non-null   float64
 15  station  210120 non-null  int64  
dtypes: float64(14), int64(1), object(1)
memory usage: 25.6+ MB
```

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

Out[4]:

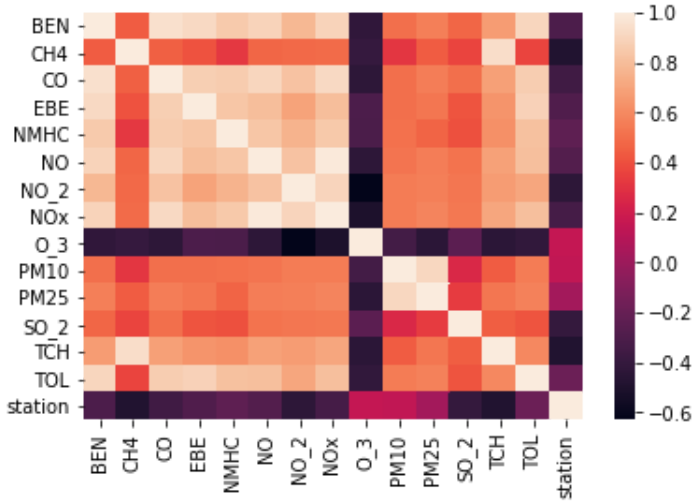
	date	BEN	CH4	CO	EBE	NMHC	NO	NO_2	NOx	O_3	PM10	PM25	SO_2	TCH	TOL	s
87457	2017-10-01 01:00:00	0.6	1.22	0.3	0.4	0.09	4.0	54.0	60.0	43.0	12.0	9.0	13.0	1.31	2.3	280
87462	2017-10-01 01:00:00	0.2	1.18	0.2	0.1	0.09	1.0	26.0	28.0	42.0	14.0	6.0	3.0	1.27	1.1	280
87481	2017-10-01 02:00:00	0.4	1.22	0.2	0.2	0.06	2.0	32.0	36.0	53.0	14.0	10.0	13.0	1.28	1.3	280
87486	2017-10-01 02:00:00	0.2	1.19	0.2	0.1	0.07	1.0	15.0	17.0	51.0	18.0	8.0	3.0	1.26	0.8	280
87505	2017-10-01 03:00:00	0.3	1.23	0.2	0.2	0.06	2.0	27.0	29.0	57.0	15.0	10.0	13.0	1.29	1.0	280
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
158238	2017-12-31 22:00:00	0.3	1.11	0.2	0.1	0.03	1.0	8.0	9.0	73.0	3.0	1.0	3.0	1.14	0.2	280
158257	2017-12-31 23:00:00	0.6	1.38	0.3	0.1	0.03	6.0	42.0	51.0	47.0	7.0	4.0	3.0	1.41	0.9	280
158262	2017-12-31 23:00:00	0.3	1.11	0.2	0.1	0.03	1.0	6.0	8.0	72.0	6.0	3.0	3.0	1.14	0.2	280
158281	2018-01-01 00:00:00	0.5	1.38	0.2	0.1	0.02	2.0	20.0	23.0	69.0	4.0	2.0	3.0	1.39	0.6	280
158286	2018-01-01 00:00:00	0.3	1.11	0.2	0.1	0.03	1.0	1.0	3.0	83.0	8.0	5.0	3.0	1.14	0.2	280

4127 rows × 16 columns

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

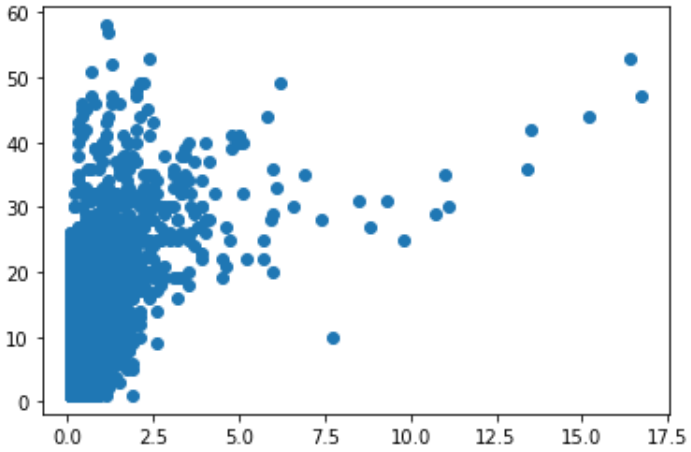
In [6]:
```
sns.heatmap(df1.corr())
```

Out[6]: <AxesSubplot:>



In [7]:
```
plt.plot(df1["EBE"],df1["PM25"],"o")
```

Out[7]: [<matplotlib.lines.Line2D at 0x25f5f6d7fd0>]



In [8]:
```
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)
```

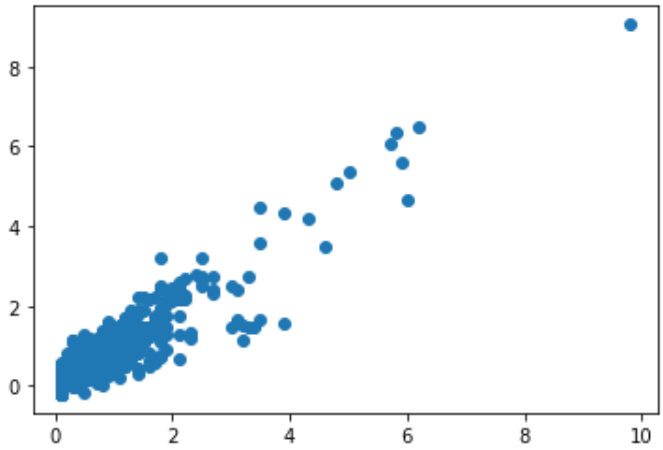
Linear

In [9]:
```
li=LinearRegression()
li.fit(x_train,y_train)
```

Out[9]: LinearRegression()

In [10]:
```
prediction=li.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[10]:
```
<matplotlib.collections.PathCollection at 0x25f5f7a3c70>
```



In [11]:
```
lis=li.score(x_test,y_test)
```

In [12]:
```
df1["TCH"].value_counts()
```

Out[12]:
```
1.24    124
1.36    118
1.26    112
1.25    110
1.41    107
       ...
3.23      1
2.47      1
2.35      1
2.61      1
2.94      1
Name: TCH, Length: 164, dtype: int64
```

In [13]:
```
df1.loc[df1["TCH"]<1.40,"TCH"]=1
df1.loc[df1["TCH"]>1.40,"TCH"]=2
df1["TCH"].value_counts()
```

Out[13]:
```
1.0    2428
2.0    1699
Name: TCH, dtype: int64
```

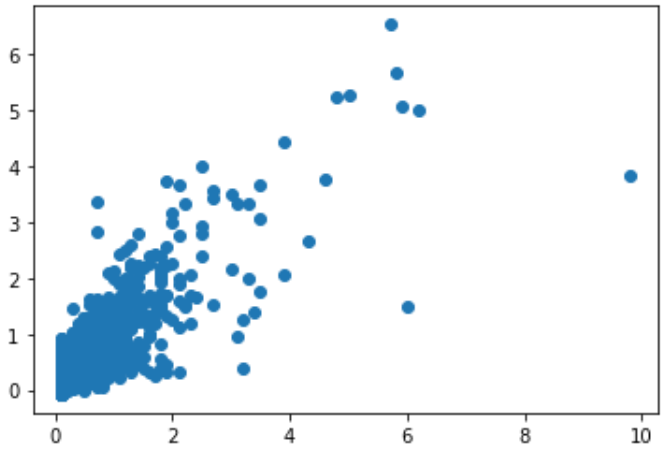
Lasso

In [14]:
```
la=Lasso(alpha=5)
la.fit(x_train,y_train)
```

Out[14]:
```
Lasso(alpha=5)
```

In [15]:
```
prediction1=la.predict(x_test)
plt.scatter(y_test,prediction1)
```

Out[15]:
```
<matplotlib.collections.PathCollection at 0x25f5f80f820>
```



In [16]:
```
las=la.score(x_test,y_test)
```

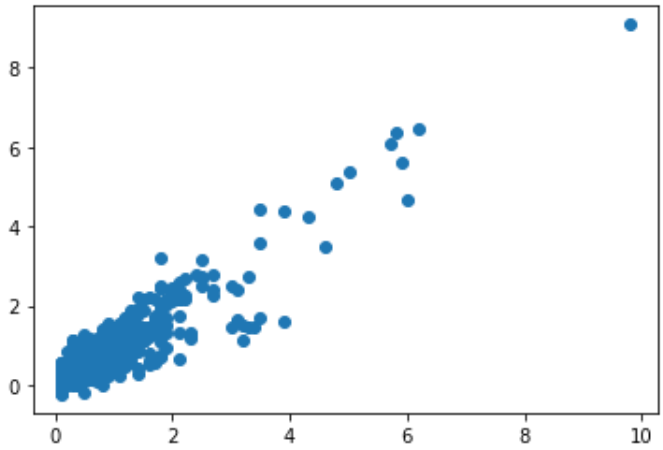
Ridge

In [17]:
```
rr=Ridge(alpha=1)
rr.fit(x_train,y_train)
```

Out[17]:
```
Ridge(alpha=1)
```

In [18]:
```
prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[18]:
```
<matplotlib.collections.PathCollection at 0x25f5f64b2e0>
```



In [19]:
```
rrs=rr.score(x_test,y_test)
```

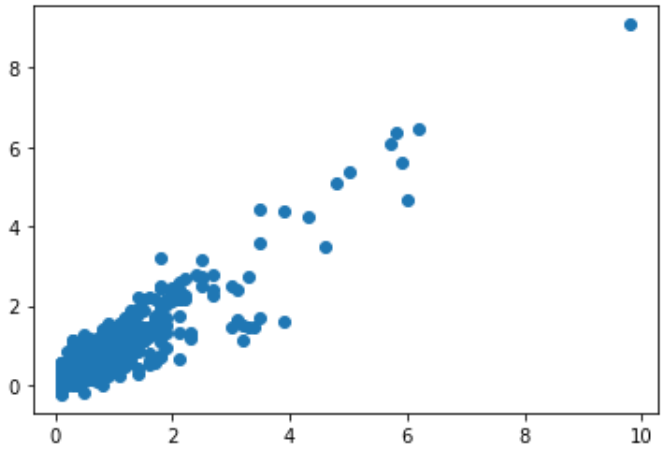
ElasticNet

In [20]:
```
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[20]:
```
ElasticNet()
```

In [21]:
```
prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[21]:
```
<matplotlib.collections.PathCollection at 0x25f60140640>
```



In [22]:
```
ens=en.score(x_test,y_test)
```

In [23]:
```
print(rr.score(x_test,y_test))
rr.score(x_train,y_train)
```

```
0.8446472469016151
```

Out[23]:
```
0.8976558488919304
```

Logistic

In [24]:
```
g={"TCH":{1.0:"Low",2.0:"High"}}
df1=df1.replace(g)
df1["TCH"].value_counts()
```

Out[24]:
```
Low     2428
High    1699
Name: TCH, dtype: int64
```

In [25]:
```
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 [26]:
```
lo=LogisticRegression()
lo.fit(x_train,y_train)
```

Out[26]:
```
LogisticRegression()
```

In [27]:
```
prediction3=lo.predict(x_test)
plt.scatter(y_test,prediction3)
```

Out[27]:
```
<matplotlib.collections.PathCollection at 0x25f601ad700>
```

In [28]:
```
los=lo.score(x_test,y_test)
```

Random Forest

In [29]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
```

In [30]:
```
g1={"TCH":{"Low":1.0,"High":2.0}}
df1=df1.replace(g1)
```

In [31]:
```
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 [32]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[32]:
```
RandomForestClassifier()
```

In [33]:
```
parameter={
    'max_depth':[1,2,4,5,6],
    'min_samples_leaf':[5,10,15,20,25],
    'n_estimators':[10,20,30,40,50]
}
```

In [34]:
```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[34]:
```
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 [35]:
```
rfcs=grid_search.best_score_
```

In [36]:
```
rfc_best=grid_search.best_estimator_
```

In [37]:
```
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"],fill
```

Out[37]:
```
[Text(1962.6206896551726, 2019.0857142857144, 'BEN <= 0.65\ngini = 0.477\nsamples =
1820\nvalue = [1757, 1131]\nclass = Yes'),
 Text(837.0000000000001, 1708.457142857143, 'NO_2 <= 24.5\ngini = 0.239\nsamples = 9
40\nvalue = [1312, 211]\nclass = Yes'),
 Text(384.82758620689657, 1397.8285714285716, 'CH4 <= 1.315\ngini = 0.069\nsamples =
362\nvalue = [567, 21]\nclass = Yes'),
 Text(230.89655172413796, 1087.2, 'CO <= 0.15\ngini = 0.004\nsamples = 342\nvalue =
[555, 1]\nclass = Yes'),
 Text(153.93103448275863, 776.5714285714287, 'CH4 <= 1.215\ngini = 0.021\nsamples =
60\nvalue = [93, 1]\nclass = Yes'),
 Text(76.96551724137932, 465.9428571428573, 'gini = 0.0\nsamples = 55\nvalue = [89,
0]\nclass = Yes'),
 Text(230.89655172413796, 465.9428571428573, 'gini = 0.32\nsamples = 5\nvalue = [4,
1]\nclass = Yes'),
 Text(307.86206896551727, 776.5714285714287, 'gini = 0.0\nsamples = 282\nvalue = [46
2, 0]\nclass = Yes'),
 Text(538.7586206896552, 1087.2, 'NMHC <= 0.025\ngini = 0.469\nsamples = 20\nvalue =
[12, 20]\nclass = No'),
 Text(461.79310344827593, 776.5714285714287, 'TOL <= 0.75\ngini = 0.245\nsamples = 1
```

In [38]:
```
print("Linear:",lis)
print("Lasso:",las)
print("Ridge:",rrs)
print("ElasticNet:",ens)
print("Logistic:",los)
print("Random Forest:",rfcs)
```

```
Linear: 0.8441268043977549
Lasso: 0.6393010463991731
Ridge: 0.8446472469016151
ElasticNet: 0.7899757472729623
Logistic: 0.579499596448749
Random Forest: 0.9681440443213296
```

Best Model is Random Forest

In [39]:
```
df2=pd.read_csv("C:/Users/user/Downloads/FP1_air/csvs_per_year/csvs_per_year/madrid_201
df2
```

Out[39]:

	date	BEN	CH4	CO	EBE	NMHC	NO	NO_2	NOx	O_3	PM10	PM25	SO_2	TCH	TOL	
0	2018-03-01 01:00:00	NaN	NaN	0.3	NaN	NaN	1.0	29.0	31.0	NaN	NaN	NaN	2.0	NaN	NaN	2
1	2018-03-01 01:00:00	0.5	1.39	0.3	0.2	0.02	6.0	40.0	49.0	52.0	5.0	4.0	3.0	1.41	0.8	2
2	2018-03-01 01:00:00	0.4	NaN	NaN	0.2	NaN	4.0	41.0	47.0	NaN	NaN	NaN	NaN	NaN	1.1	1
3	2018-03-01 01:00:00	NaN	NaN	0.3	NaN	NaN	1.0	35.0	37.0	54.0	NaN	NaN	NaN	NaN	NaN	2
4	2018-03-01 01:00:00	NaN	NaN	NaN	NaN	NaN	1.0	27.0	29.0	49.0	NaN	NaN	3.0	NaN	NaN	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
69091	2018-02-01 00:00:00	NaN	NaN	0.5	NaN	NaN	66.0	91.0	192.0	1.0	35.0	22.0	NaN	NaN	NaN	2
69092	2018-02-01 00:00:00	NaN	NaN	0.7	NaN	NaN	87.0	107.0	241.0	NaN	29.0	NaN	15.0	NaN	NaN	2
69093	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	28.0	48.0	91.0	2.0	NaN	NaN	NaN	NaN	NaN	2
69094	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	141.0	103.0	320.0	2.0	NaN	NaN	NaN	NaN	NaN	2
69095	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	69.0	96.0	202.0	3.0	26.0	NaN	NaN	NaN	NaN	2

69096 rows × 16 columns

In [40]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 69096 entries, 0 to 69095
Data columns (total 16 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   date     69096 non-null  object 
 1   BEN      16950 non-null  float64
 2   CH4      8440 non-null   float64
 3   CO       28598 non-null  float64
 4   EBE      16949 non-null  float64
 5   NMHC     8440 non-null   float64
 6   NO       68826 non-null  float64
 7   NO_2     68826 non-null  float64
 8   NOx      68826 non-null  float64
 9   O_3      40049 non-null  float64
 10  PM10     36911 non-null  float64
 11  PM25     18912 non-null  float64
 12  SO_2     28586 non-null  float64
 13  TCH      8440 non-null   float64
 14  TOL      16950 non-null  float64
 15  station  69096 non-null  int64  
dtypes: float64(14), int64(1), object(1)
memory usage: 8.4+ MB
```

In [41]:
```
df3=df2.dropna()
df3
```

Out[41]:

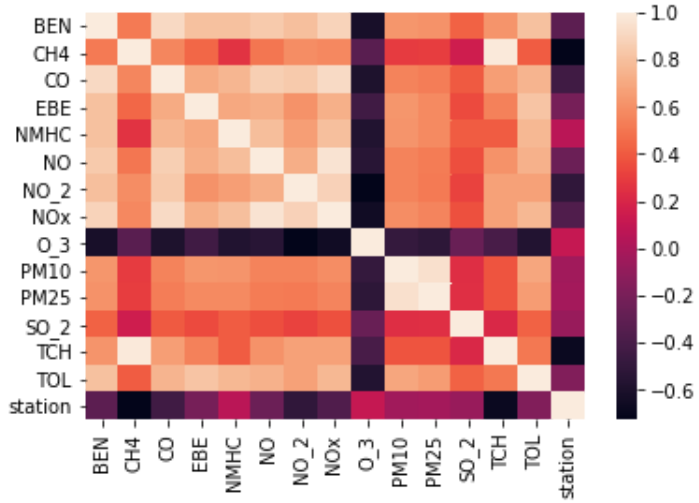
	date	BEN	CH4	CO	EBE	NMHC	NO	NO_2	NOx	O_3	PM10	PM25	SO_2	TCH	TOL	
1	2018-03-01 01:00:00	0.5	1.39	0.3	0.2	0.02	6.0	40.0	49.0	52.0	5.0	4.0	3.0	1.41	0.8	28
6	2018-03-01 01:00:00	0.4	1.11	0.2	0.1	0.06	1.0	25.0	27.0	55.0	5.0	4.0	4.0	1.16	1.4	28
25	2018-03-01 02:00:00	0.4	1.42	0.2	0.1	0.01	4.0	26.0	32.0	64.0	4.0	4.0	3.0	1.44	0.7	28
30	2018-03-01 02:00:00	0.3	1.10	0.2	0.1	0.05	1.0	12.0	13.0	69.0	5.0	4.0	4.0	1.14	0.8	28
49	2018-03-01 03:00:00	0.3	1.41	0.2	0.1	0.01	3.0	16.0	20.0	68.0	3.0	2.0	3.0	1.42	0.4	28
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
69030	2018-01-31 22:00:00	1.8	1.21	0.7	1.7	0.19	151.0	129.0	361.0	1.0	45.0	26.0	11.0	1.40	11.9	28
69049	2018-01-31 23:00:00	3.1	1.87	1.2	2.0	0.35	296.0	162.0	615.0	3.0	39.0	23.0	8.0	2.22	12.5	28
69054	2018-01-31 23:00:00	1.6	1.17	0.6	1.4	0.15	127.0	106.0	301.0	1.0	43.0	25.0	8.0	1.32	10.3	28
69073	2018-02-01 00:00:00	3.2	1.53	1.0	2.1	0.19	125.0	117.0	309.0	3.0	37.0	24.0	6.0	1.72	13.0	28
69078	2018-02-01 00:00:00	1.3	1.14	0.4	0.8	0.10	54.0	73.0	155.0	1.0	27.0	16.0	5.0	1.24	6.8	28

4562 rows × 16 columns

In [42]:
```
df3=df3.drop(["date"],axis=1)
```

In [43]:
```
sns.heatmap(df3.corr())
```

Out[43]: <AxesSubplot:>



In [44]:
```
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

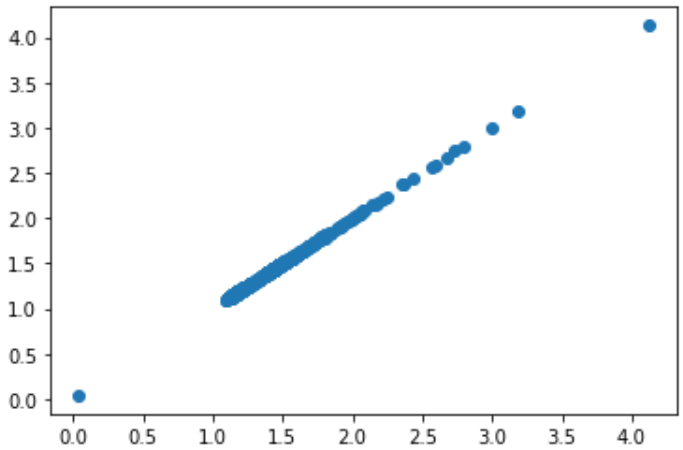
In [45]:
```
li=LinearRegression()
li.fit(x_train,y_train)
```

Out[45]: LinearRegression()

In [ ]:
```

```

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

Out[46]: <matplotlib.collections.PathCollection at 0x25f605d5070>



In [47]:
```
lis=li.score(x_test,y_test)
```

In [48]:
```
df3["TCH"].value_counts()
```

Out[48]:
```
1.15    246
1.43    232
1.44    223
1.14    210
1.13    201
       ...
2.35      1
2.58      1
2.73      1
2.12      1
1.96      1
Name: TCH, Length: 143, dtype: int64
```

In [49]:
```
df3.loc[df3["TCH"]<1.40,"TCH"]=1
df3.loc[df3["TCH"]>1.40,"TCH"]=2
df3["TCH"].value_counts()
```

Out[49]:
```
2.0    2477
1.0    2085
Name: TCH, dtype: int64
```

In [ ]:
```

```

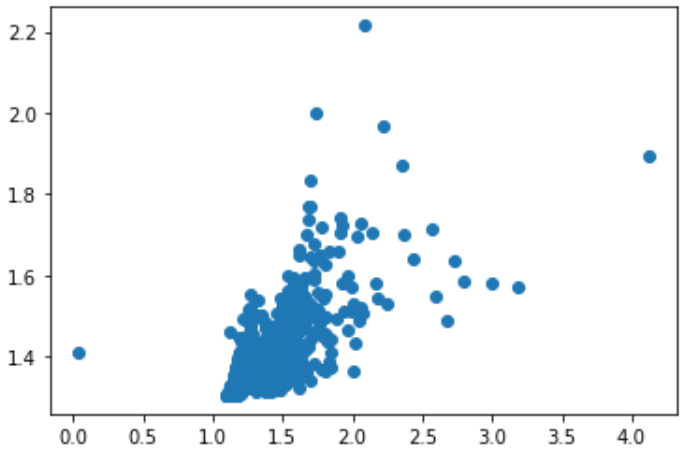
Lasso

In [50]:
```
la=Lasso(alpha=5)
la.fit(x_train,y_train)
```

Out[50]:
```
Lasso(alpha=5)
```

In [51]:
```
prediction1=la.predict(x_test)
plt.scatter(y_test,prediction1)
```

Out[51]:
```
<matplotlib.collections.PathCollection at 0x25f60639190>
```



In [52]:
```
las=la.score(x_test,y_test)
```

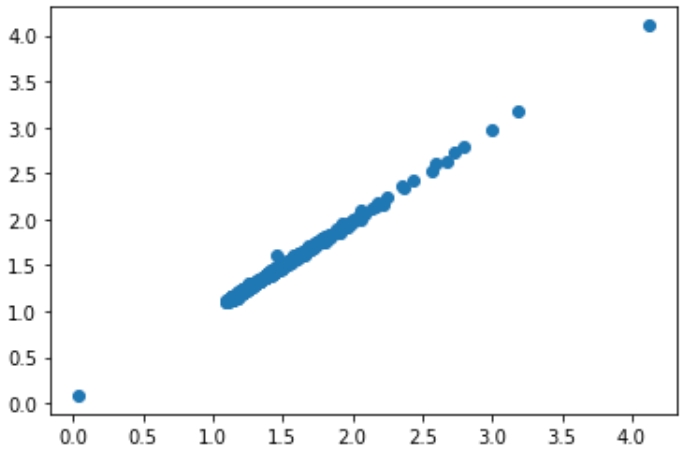
Ridge

In [53]:
```
rr=Ridge(alpha=1)
rr.fit(x_train,y_train)
```

Out[53]:
```
Ridge(alpha=1)
```

In [54]:
```
prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[54]:
```
<matplotlib.collections.PathCollection at 0x25f6068d760>
```



In [55]:
```
rrs=rr.score(x_test,y_test)
```

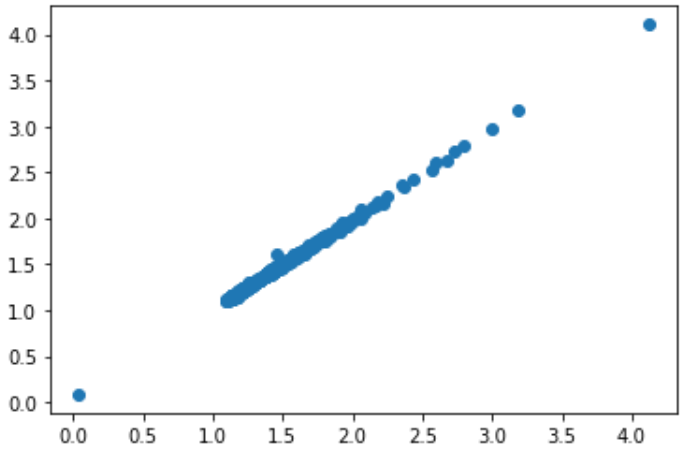
ElasticNet

In [56]:
```
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[56]:
```
ElasticNet()
```

In [57]:
```
prediction2=rr.predict(x_test)
plt.scatter(y_test,prediction2)
```

Out[57]:
```
<matplotlib.collections.PathCollection at 0x25f606f32b0>
```



In [58]:
```
ens=en.score(x_test,y_test)
```

In [59]:
```
print(rr.score(x_test,y_test))
rr.score(x_train,y_train)
```

```
0.9979283454674264
```

Out[59]:
```
0.998113461617783
```

Logistic

In [60]:
```
g={"TCH":{1.0:"Low",2.0:"High"}}
df3=df3.replace(g)
df3["TCH"].value_counts()
```

Out[60]:
```
High    2477
Low     2085
Name: TCH, dtype: int64
```

In [61]:
```
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 [62]:
```
lo=LogisticRegression()
lo.fit(x_train,y_train)
```

Out[62]:
```
LogisticRegression()
```

In [63]:
```
prediction3=lo.predict(x_test)
plt.scatter(y_test,prediction3)
```

Out[63]:
```
<matplotlib.collections.PathCollection at 0x25f6075c7f0>
```

In [64]:
```
los=lo.score(x_test,y_test)
```

Random Forest

In [65]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
```

In [66]:
```
g1={"TCH":{"Low":1.0,"High":2.0}}
df3=df3.replace(g1)
```

In [67]:
```
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 [68]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[68]:
```
RandomForestClassifier()
```

In [69]:
```
parameter={
    'max_depth':[1,2,4,5,6],
    'min_samples_leaf':[5,10,15,20,25],
    'n_estimators':[10,20,30,40,50]
}
```

In [70]:
```
grid_search=GridSearchCV(estimator=rfc,param_grid=parameter,cv=2,scoring="accuracy")
grid_search.fit(x_train,y_train)
```

Out[70]:
```
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')
```

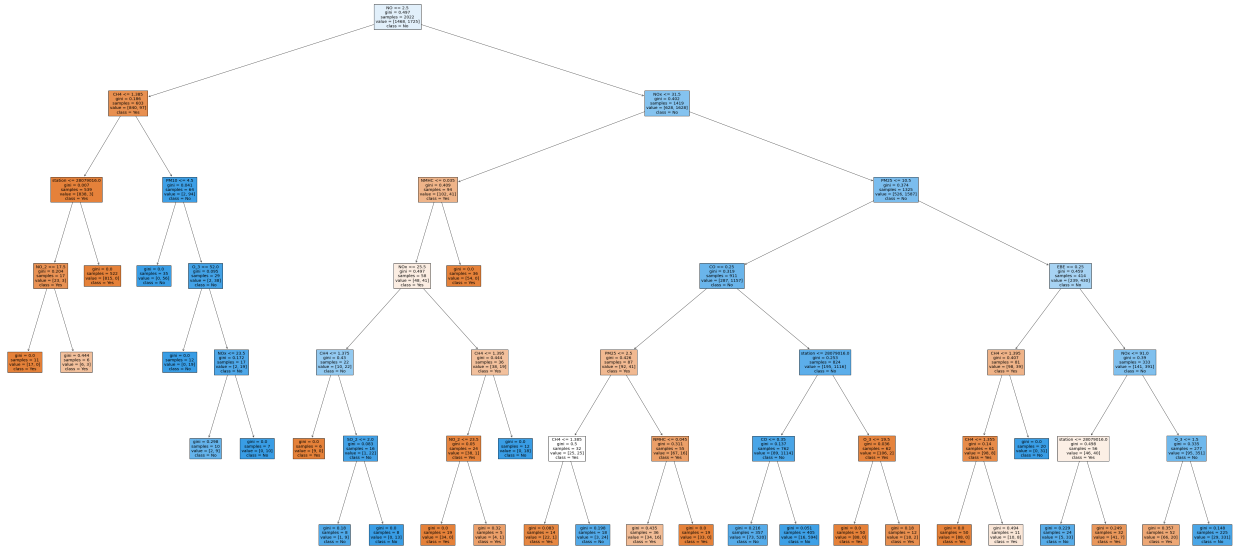
In [71]:
```
rfcs=grid_search.best_score_
```

In [72]:
```
rfc_best=grid_search.best_estimator_
```

In [73]:
```
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"],fill
```

Out[73]:
```
[Text(1435.6875, 2019.0857142857144, 'NO <= 2.5\ngini = 0.497\nsamples = 2022\nvalue =
[1468, 1725]\nclass = No'),
 Text(465.0, 1708.457142857143, 'CH4 <= 1.385\ngini = 0.186\nsamples = 603\nvalue = [8
40, 97]\nclass = Yes'),
 Text(279.0, 1397.8285714285716, 'station <= 28079016.0\ngini = 0.007\nsamples = 539\n
value = [838, 3]\nclass = Yes'),
 Text(186.0, 1087.2, 'NO_2 <= 17.5\ngini = 0.204\nsamples = 17\nvalue = [23, 3]\nclass
= Yes'),
 Text(93.0, 776.5714285714287, 'gini = 0.0\nsamples = 11\nvalue = [17, 0]\nclass = Ye
s'),
 Text(279.0, 776.5714285714287, 'gini = 0.444\nsamples = 6\nvalue = [6, 3]\nclass = Ye
s'),
 Text(372.0, 1087.2, 'gini = 0.0\nsamples = 522\nvalue = [815, 0]\nclass = Yes'),
 Text(651.0, 1397.8285714285716, 'PM10 <= 4.5\ngini = 0.041\nsamples = 64\nvalue = [2,
94]\nclass = No'),
 Text(558.0, 1087.2, 'gini = 0.0\nsamples = 35\nvalue = [0, 56]\nclass = No'),
 Text(744.0, 1087.2, 'O_3 <= 52.0\ngini = 0.095\nsamples = 29\nvalue = [2, 38]\nclass
= No'),
 Text(651.0, 776.5714285714287, 'gini = 0.0\nsamples = 12\nvalue = [0, 19]\nclass = N
o'),
 Text(837.0, 776.5714285714287, 'NOx <= 23.5\ngini = 0.172\nsamples = 17\nvalue = [2,
19]\nclass = No'),
 Text(744.0, 465.9428571428573, 'gini = 0.298\nsamples = 10\nvalue = [2, 9]\nclass = N
o'),
 Text(930.0, 465.9428571428573, 'gini = 0.0\nsamples = 7\nvalue = [0, 10]\nclass = N
o'),
 Text(2406.375, 1708.457142857143, 'NOx <= 31.5\ngini = 0.402\nsamples = 1419\nvalue =
[628, 1628]\nclass = No'),
 Text(1581.0, 1397.8285714285716, 'NMHC <= 0.035\ngini = 0.409\nsamples = 94\nvalue =
[102, 41]\nclass = Yes'),
 Text(1488.0, 1087.2, 'NOx <= 25.5\ngini = 0.497\nsamples = 58\nvalue = [48, 41]\nclas
s = Yes'),
 Text(1209.0, 776.5714285714287, 'CH4 <= 1.375\ngini = 0.43\nsamples = 22\nvalue = [1
0, 22]\nclass = No'),
 Text(1116.0, 465.9428571428573, 'gini = 0.0\nsamples = 6\nvalue = [9, 0]\nclass = Ye
s'),
 Text(1302.0, 465.9428571428573, 'SO_2 <= 2.0\ngini = 0.083\nsamples = 16\nvalue = [1,
22]\nclass = No'),
 Text(1209.0, 155.3142857142857, 'gini = 0.18\nsamples = 8\nvalue = [1, 9]\nclass = N
o'),
 Text(1395.0, 155.3142857142857, 'gini = 0.0\nsamples = 8\nvalue = [0, 13]\nclass = N
o'),
 Text(1767.0, 776.5714285714287, 'CH4 <= 1.395\ngini = 0.444\nsamples = 36\nvalue = [3
8, 19]\nclass = Yes'),
 Text(1674.0, 465.9428571428573, 'NO_2 <= 23.5\ngini = 0.05\nsamples = 24\nvalue = [3
8, 1]\nclass = Yes'),
 Text(1581.0, 155.3142857142857, 'gini = 0.0\nsamples = 19\nvalue = [34, 0]\nclass = Y
es'),
 Text(1767.0, 155.3142857142857, 'gini = 0.32\nsamples = 5\nvalue = [4, 1]\nclass = Ye
s'),
 Text(1860.0, 465.9428571428573, 'gini = 0.0\nsamples = 12\nvalue = [0, 18]\nclass = N
o'),
 Text(1674.0, 1087.2, 'gini = 0.0\nsamples = 36\nvalue = [54, 0]\nclass = Yes'),
 Text(3231.75, 1397.8285714285716, 'PM25 <= 10.5\ngini = 0.374\nsamples = 1325\nvalue
= [526, 1587]\nclass = No'),
 Text(2604.0, 1087.2, 'CO <= 0.25\ngini = 0.319\nsamples = 911\nvalue = [287, 1157]\nc
lass = No'),
 Text(2232.0, 776.5714285714287, 'PM25 <= 2.5\ngini = 0.426\nsamples = 87\nvalue = [9
2, 41]\nclass = Yes'),
 Text(2046.0, 465.9428571428573, 'CH4 <= 1.385\ngini = 0.5\nsamples = 32\nvalue = [25,
25]\nclass = Yes'),
```

```
 Text(1953.0, 155.3142857142857, 'gini = 0.083\nsamples = 14\nvalue = [22, 1]\nclass =
Yes'),
 Text(2139.0, 155.3142857142857, 'gini = 0.198\nsamples = 18\nvalue = [3, 24]\nclass =
No'),
 Text(2418.0, 465.9428571428573, 'NMHC <= 0.045\ngini = 0.311\nsamples = 55\nvalue =
[67, 16]\nclass = Yes'),
 Text(2325.0, 155.3142857142857, 'gini = 0.435\nsamples = 36\nvalue = [34, 16]\nclass
= Yes'),
 Text(2511.0, 155.3142857142857, 'gini = 0.0\nsamples = 19\nvalue = [33, 0]\nclass = Y
es'),
 Text(2976.0, 776.5714285714287, 'station <= 28079016.0\ngini = 0.253\nsamples = 824\n
value = [195, 1116]\nclass = No'),
 Text(2790.0, 465.9428571428573, 'CO <= 0.35\ngini = 0.137\nsamples = 762\nvalue = [8
9, 1114]\nclass = No'),
 Text(2697.0, 155.3142857142857, 'gini = 0.216\nsamples = 357\nvalue = [73, 520]\nclas
s = No'),
 Text(2883.0, 155.3142857142857, 'gini = 0.051\nsamples = 405\nvalue = [16, 594]\nclas
s = No'),
 Text(3162.0, 465.9428571428573, 'O_3 <= 19.5\ngini = 0.036\nsamples = 62\nvalue = [10
6, 2]\nclass = Yes'),
 Text(3069.0, 155.3142857142857, 'gini = 0.0\nsamples = 50\nvalue = [88, 0]\nclass = Y
es'),
 Text(3255.0, 155.3142857142857, 'gini = 0.18\nsamples = 12\nvalue = [18, 2]\nclass =
Yes'),
 Text(3859.5, 1087.2, 'EBE <= 0.25\ngini = 0.459\nsamples = 414\nvalue = [239, 430]\nc
lass = No'),
 Text(3627.0, 776.5714285714287, 'CH4 <= 1.395\ngini = 0.407\nsamples = 81\nvalue = [9
8, 39]\nclass = Yes'),
 Text(3534.0, 465.9428571428573, 'CH4 <= 1.355\ngini = 0.14\nsamples = 61\nvalue = [9
8, 8]\nclass = Yes'),
 Text(3441.0, 155.3142857142857, 'gini = 0.0\nsamples = 50\nvalue = [88, 0]\nclass = Y
es'),
 Text(3627.0, 155.3142857142857, 'gini = 0.494\nsamples = 11\nvalue = [10, 8]\nclass =
Yes'),
 Text(3720.0, 465.9428571428573, 'gini = 0.0\nsamples = 20\nvalue = [0, 31]\nclass = N
o'),
 Text(4092.0, 776.5714285714287, 'NOx <= 91.0\ngini = 0.39\nsamples = 333\nvalue = [14
1, 391]\nclass = No'),
 Text(3906.0, 465.9428571428573, 'station <= 28079016.0\ngini = 0.498\nsamples = 56\nv
alue = [46, 40]\nclass = Yes'),
 Text(3813.0, 155.3142857142857, 'gini = 0.229\nsamples = 24\nvalue = [5, 33]\nclass =
No'),
 Text(3999.0, 155.3142857142857, 'gini = 0.249\nsamples = 32\nvalue = [41, 7]\nclass =
Yes'),
 Text(4278.0, 465.9428571428573, 'O_3 <= 1.5\ngini = 0.335\nsamples = 277\nvalue = [9
5, 351]\nclass = No'),
 Text(4185.0, 155.3142857142857, 'gini = 0.357\nsamples = 52\nvalue = [66, 20]\nclass
= Yes'),
 Text(4371.0, 155.3142857142857, 'gini = 0.148\nsamples = 225\nvalue = [29, 331]\nclas
s = No')]
```



In [74]:
```
print("Linear:",lis)
print("Lasso:",las)
print("Ridge:",rrs)
print("ElasticNet:",ens)
print("Logistic:",los)
print("Random Forest:",rfcs)
```

```
Linear: 0.9996222720077558
Lasso: 0.3728582371674869
Ridge: 0.9979283454674264
ElasticNet: 0.5889752318704515
Logistic: 0.5536888239590942
Random Forest: 0.9802696314989101
```

Best model is Linear Regression

In [ ]: