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]:

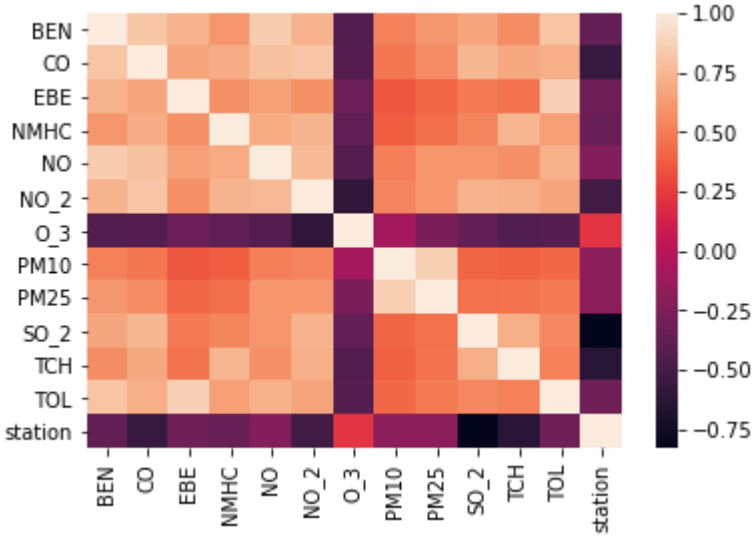
	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	
1	2015-10-01 01:00:00	2.0	0.8	1.6	0.33	40.0	95.0	4.0	37.0	24.0	12.0	1.83	8.3	28
6	2015-10-01 01:00:00	0.5	0.3	0.3	0.12	6.0	83.0	1.0	19.0	12.0	3.0	1.29	4.8	28
25	2015-10-01 02:00:00	1.6	0.7	1.3	0.38	81.0	105.0	4.0	36.0	19.0	13.0	1.93	6.9	28
30	2015-10-01 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)
```

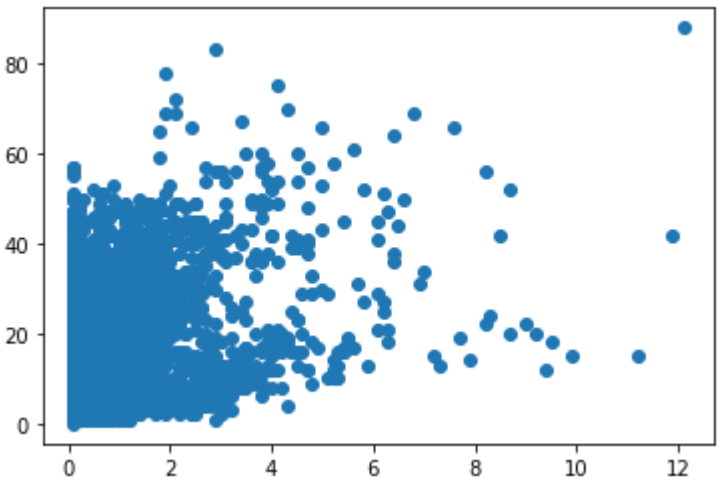
In [6]:
```
sns.heatmap(df1.corr())
```

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)
```

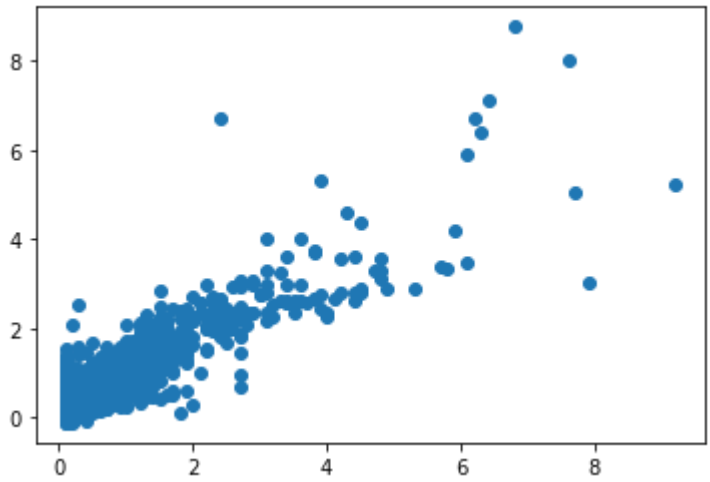
Linear

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
```

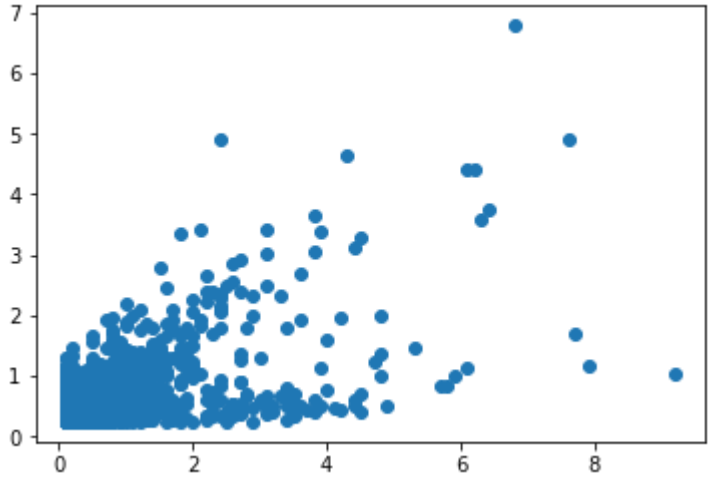
Lasso

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)
```

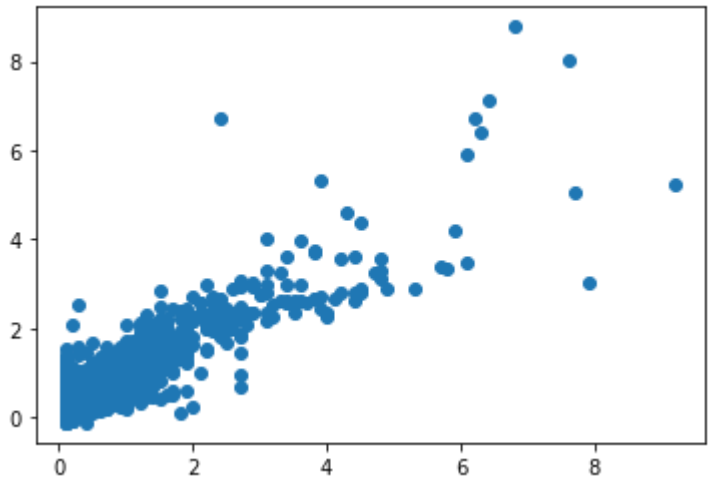
Ridge

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)
```

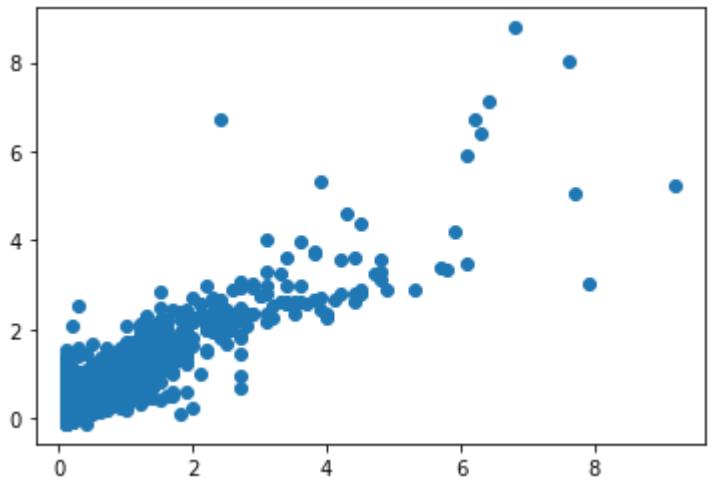
ElasticNet

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_
```

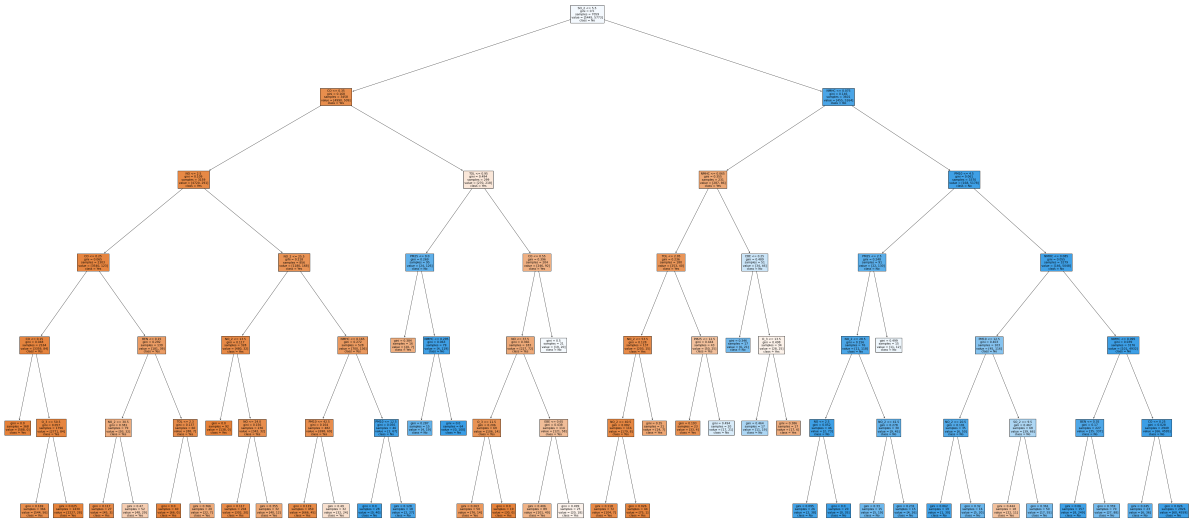
In [36]:
```
rfc_best=grid_search.best_estimator_
```

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]:

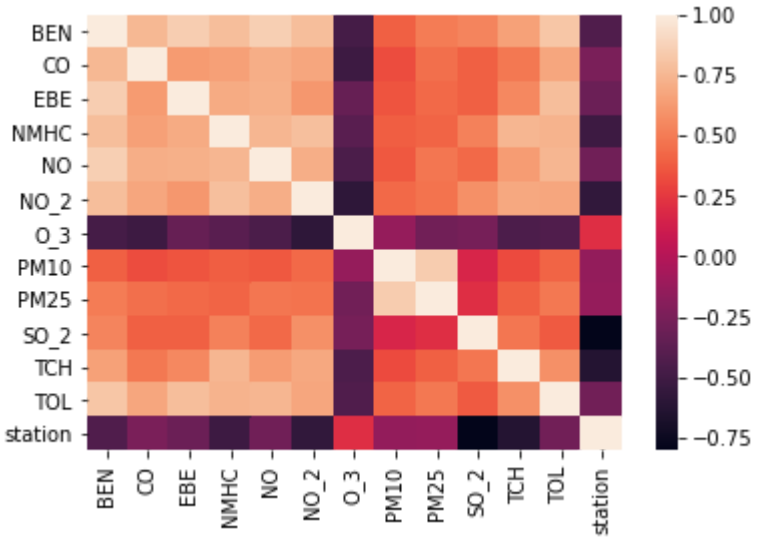
	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	
1	2016-11-01 01:00:00	3.1	1.1	2.0	0.53	260.0	144.0	4.0	46.0	24.0	18.0	2.44	14.4	28
6	2016-11-01 01:00:00	0.7	0.8	0.4	0.13	57.0	66.0	3.0	23.0	15.0	4.0	1.35	5.0	28
25	2016-11-01 02:00:00	2.7	1.0	2.1	0.40	139.0	114.0	4.0	37.0	21.0	14.0	2.30	15.0	28
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

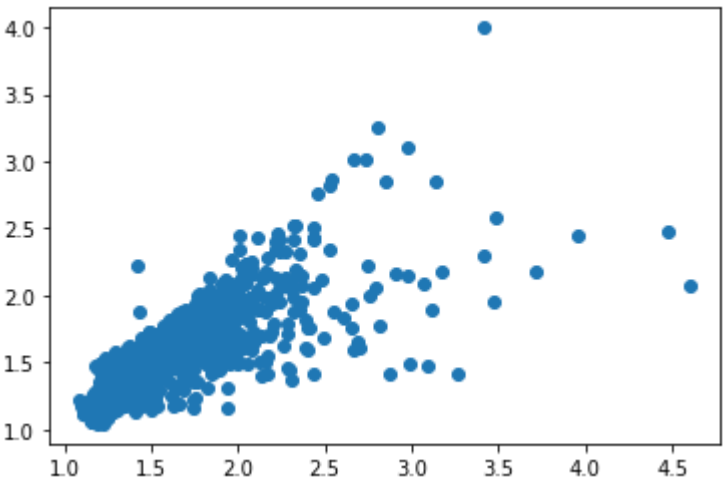
In [45]:
```
li=LinearRegression()
li.fit(x_train,y_train)
```

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
```

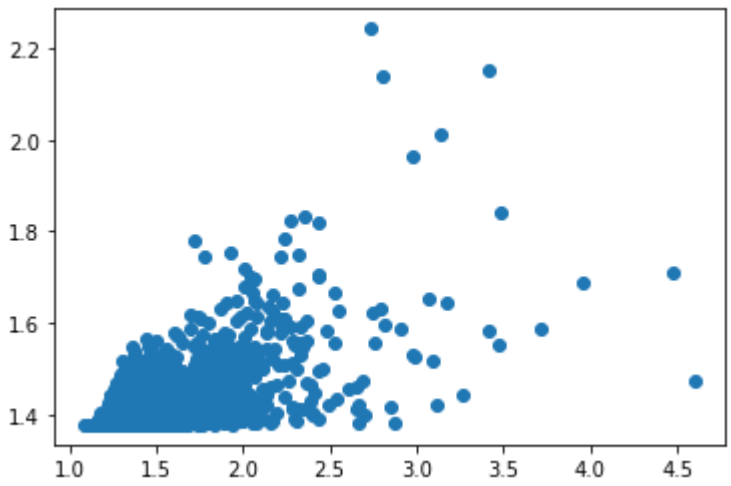
Lasso

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)
```

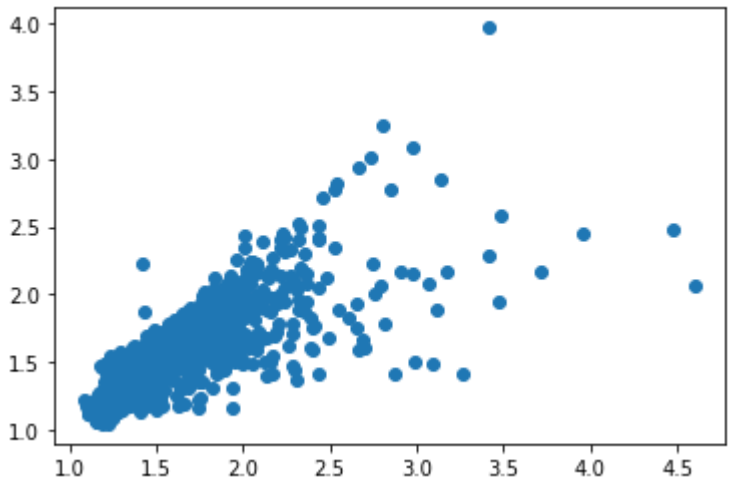
Ridge

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)
```

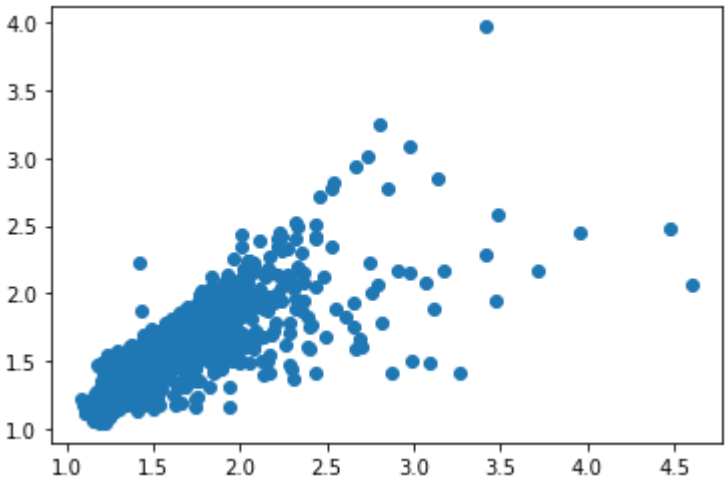
ElasticNet

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_
```

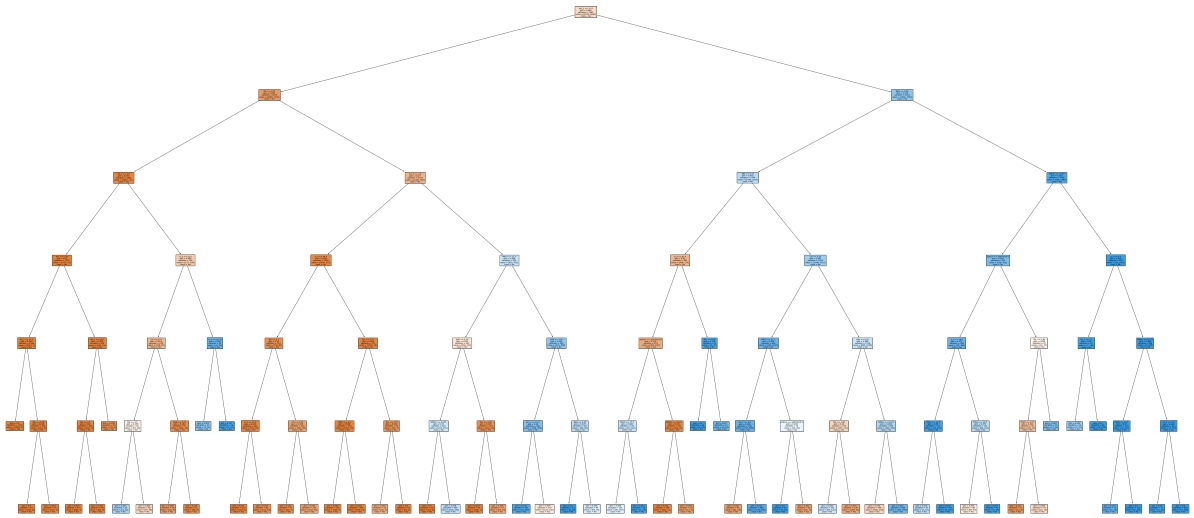
In [73]:
```
from sklearn.tree import plot_tree

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