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]:

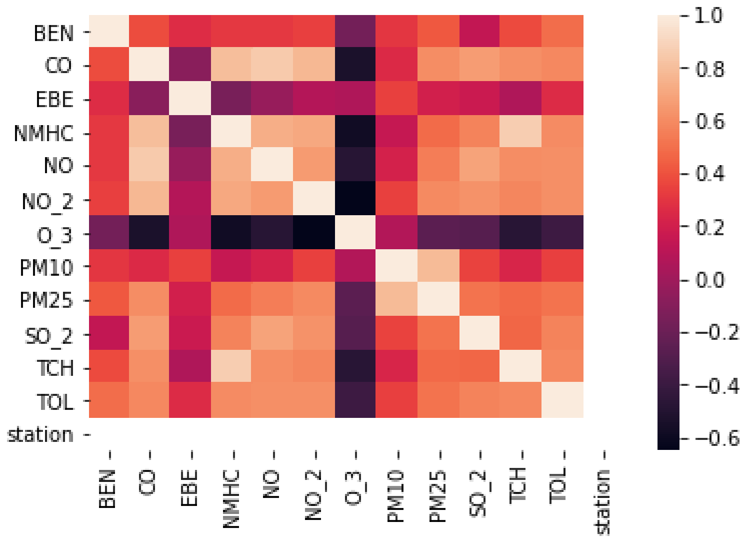
	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	st
17286	2013-08-01 01:00:00	0.4	0.2	0.8	0.28	1.0	24.0	79.0	35.0	8.0	3.0	1.49	1.3	2807
17310	2013-08-01 02:00:00	0.5	0.2	0.9	0.28	1.0	16.0	93.0	60.0	18.0	3.0	1.61	4.0	2807
17334	2013-08-01 03:00:00	0.5	0.2	1.1	0.29	1.0	14.0	90.0	38.0	12.0	3.0	1.71	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)
```

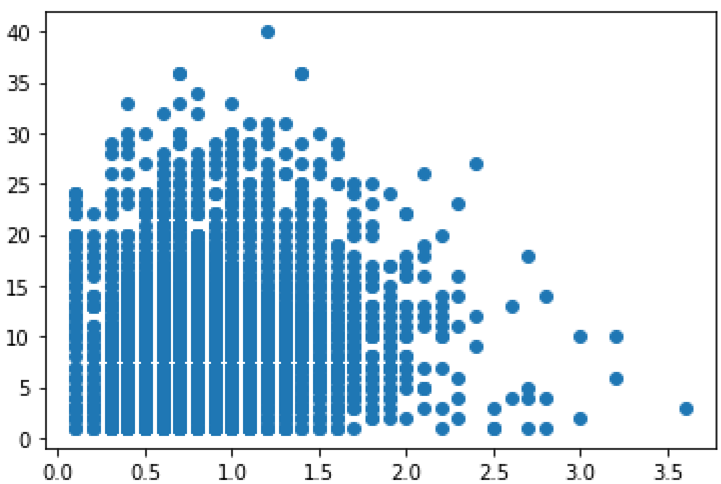
In [6]:
```
sns.heatmap(df1.corr())
```

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)
```

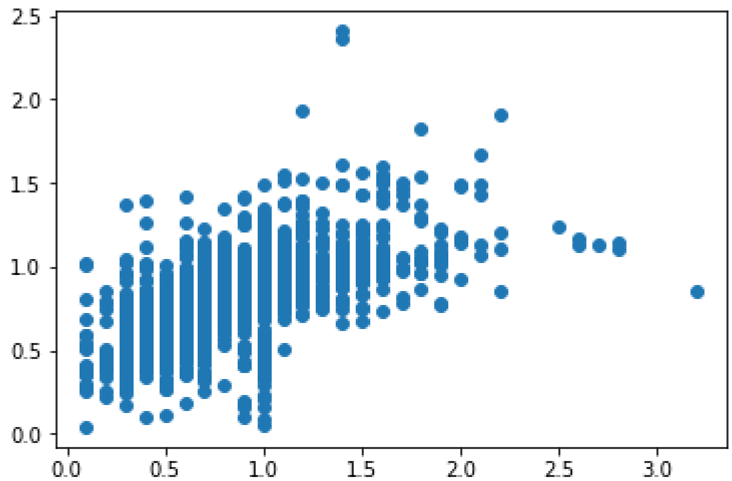
Linear

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
```

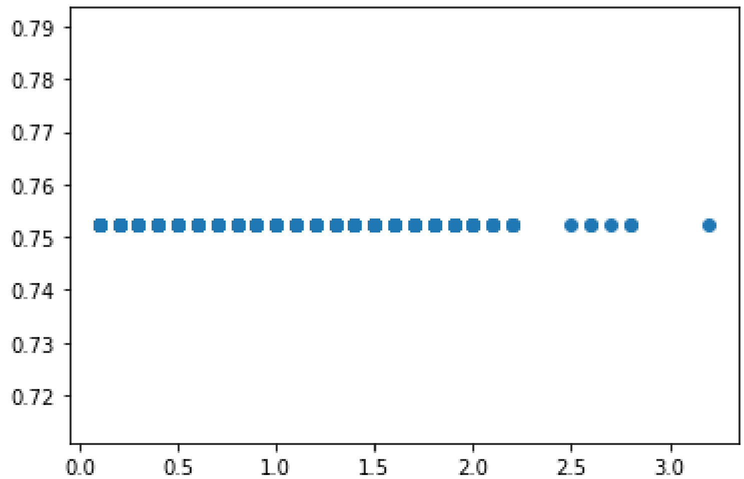
Lasso

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)
```

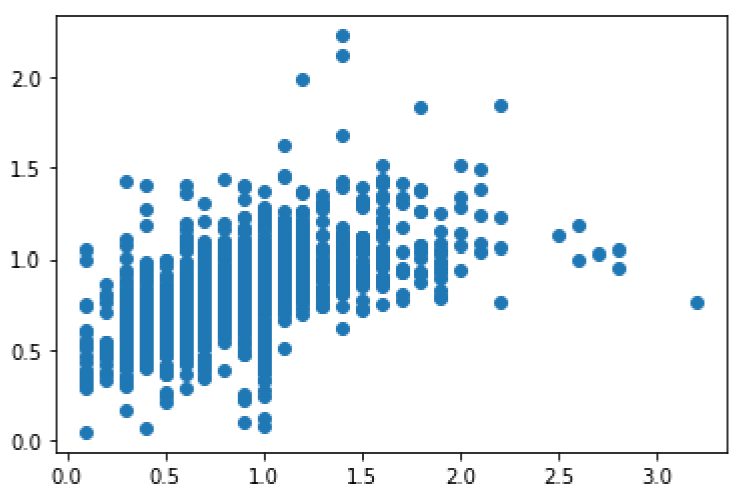
Ridge

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)
```

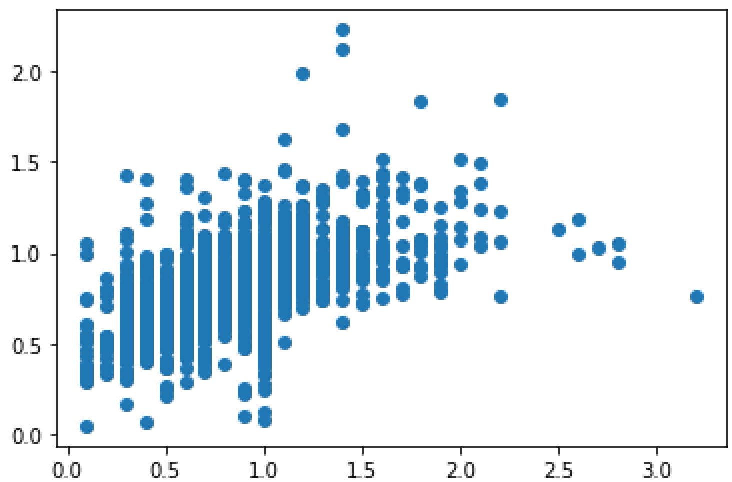
ElasticNet

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_
```

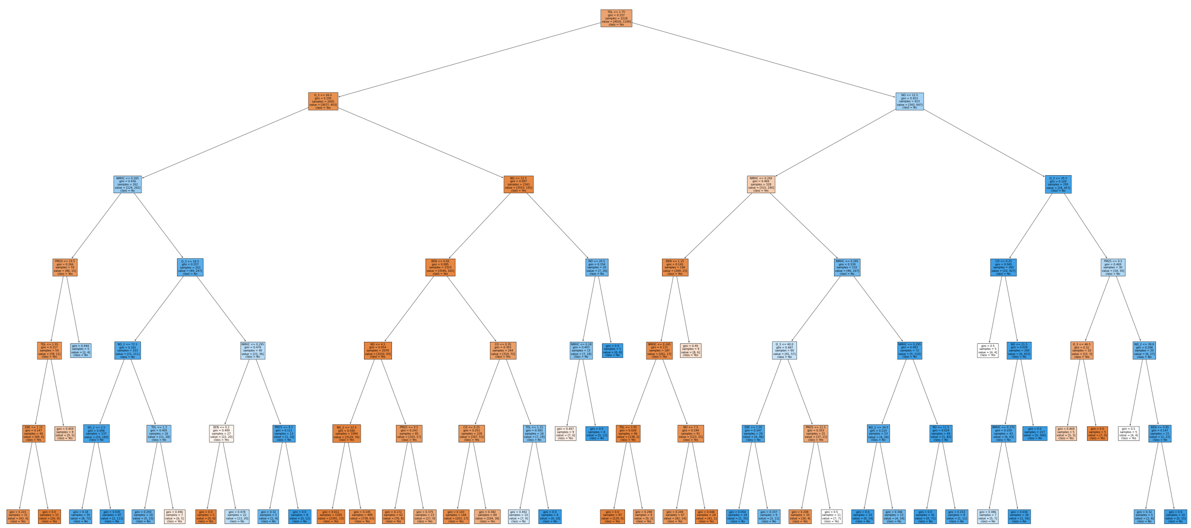
In [37]:
```
from sklearn.tree import plot_tree

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]:

```python
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]:

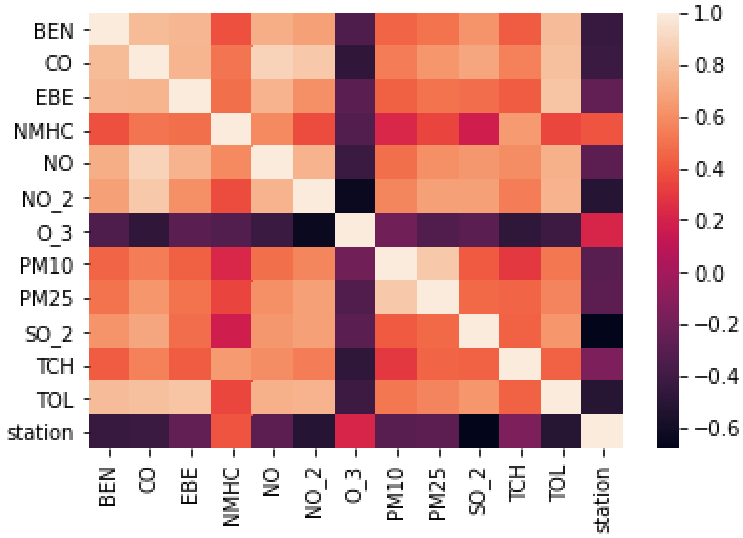
	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	s
1	2014-06-01 01:00:00	0.2	0.2	0.1	0.11	3.0	17.0	68.0	10.0	5.0	5.0	1.36	1.3	280
6	2014-06-01 01:00:00	0.1	0.2	0.1	0.23	1.0	5.0	80.0	4.0	3.0	2.0	1.21	0.1	280
25	2014-06-01 02:00:00	0.2	0.2	0.1	0.11	4.0	21.0	63.0	9.0	6.0	5.0	1.36	0.8	280
30	2014-06-01 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

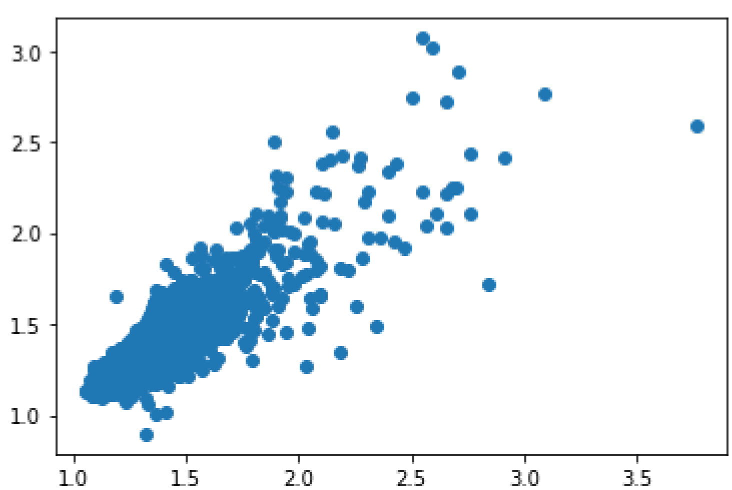
In [45]:
```
li=LinearRegression()
li.fit(x_train,y_train)
```

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 [ ]:

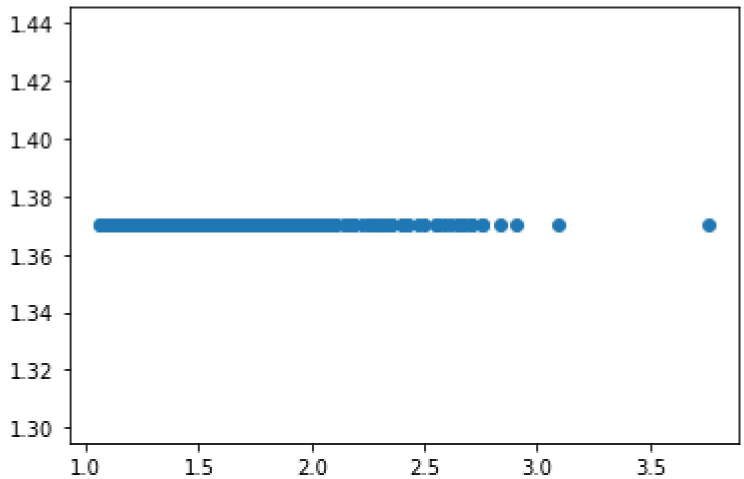
Lasso

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)
```

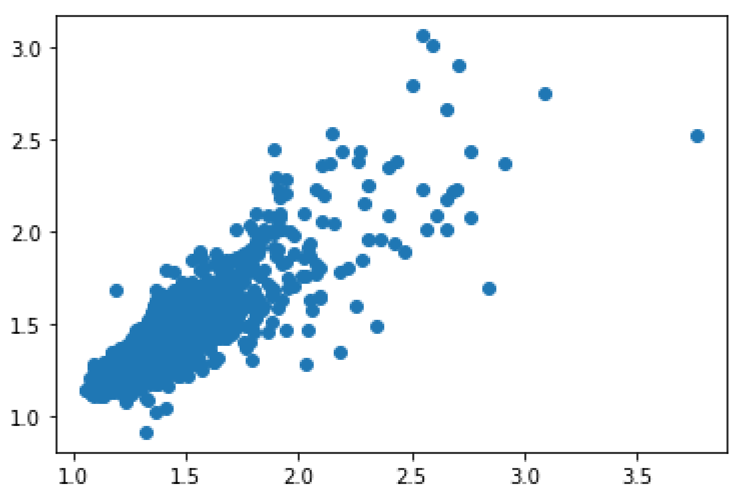
Ridge

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)
```

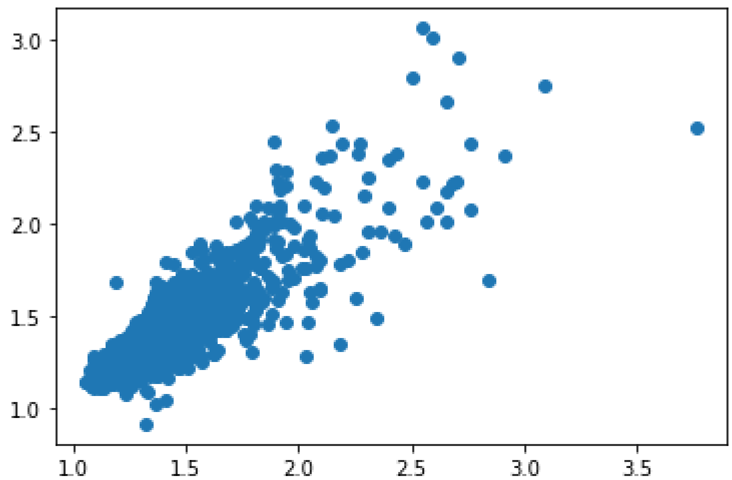
ElasticNet

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_
```

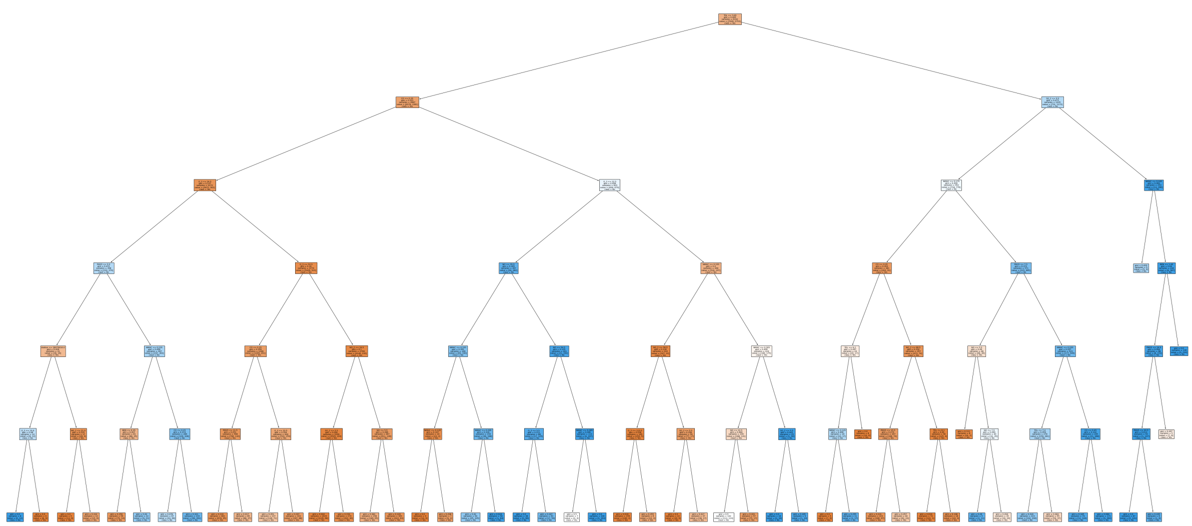
In [73]:
```
from sklearn.tree import plot_tree

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