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]:

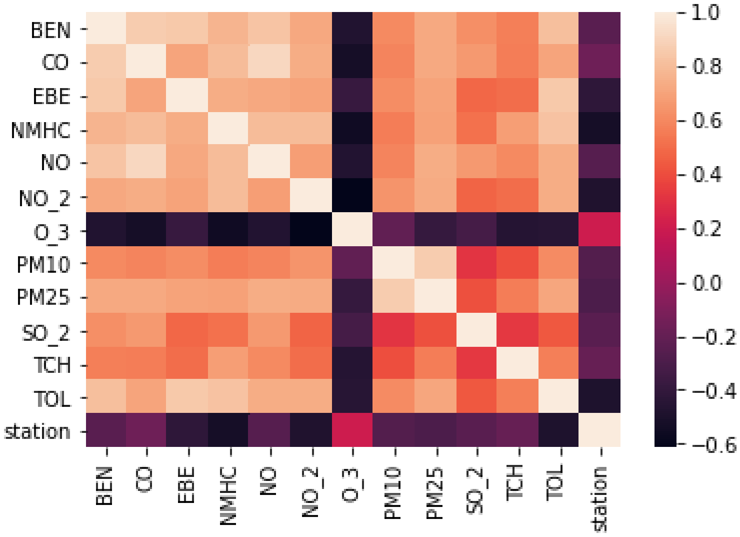
	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	s
1	2011-11-01 01:00:00	2.5	0.4	3.5	0.26	68.0	92.0	3.0	40.0	24.0	9.0	1.54	8.7	280
6	2011-11-01 01:00:00	0.7	0.3	1.1	0.16	17.0	66.0	7.0	22.0	16.0	2.0	1.36	1.7	280
25	2011-11-01 02:00:00	1.8	0.3	2.8	0.20	34.0	76.0	3.0	34.0	21.0	8.0	1.71	7.4	280
30	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)
```

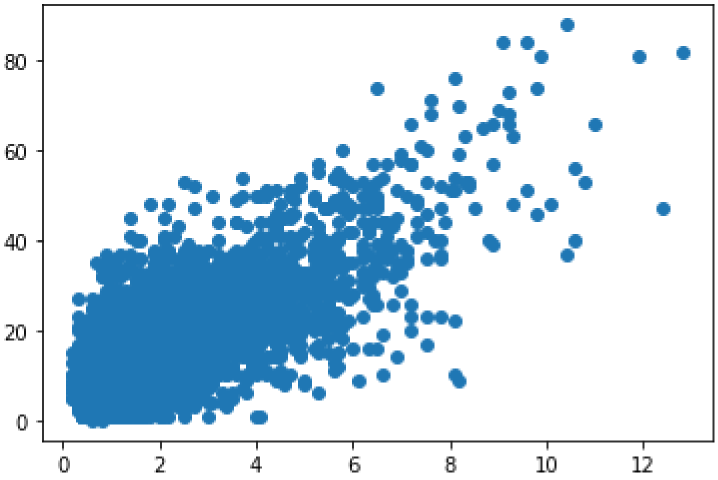
```
In [6]: sns.heatmap(df1.corr())
```

```
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)
```

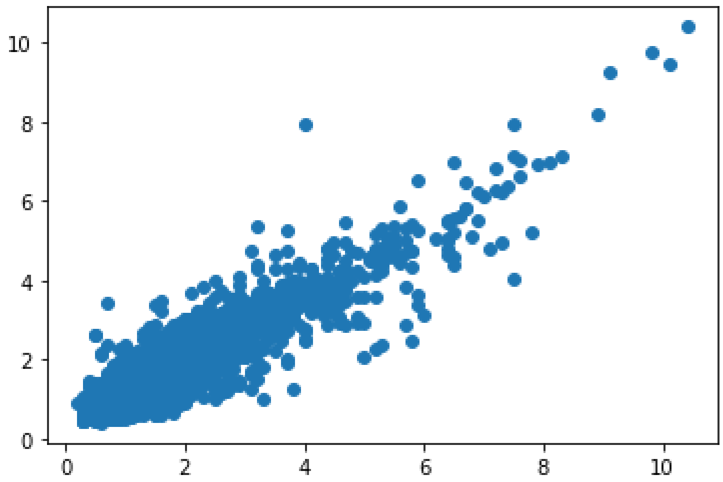
```
Out[9]: LinearRegression()
```

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]:

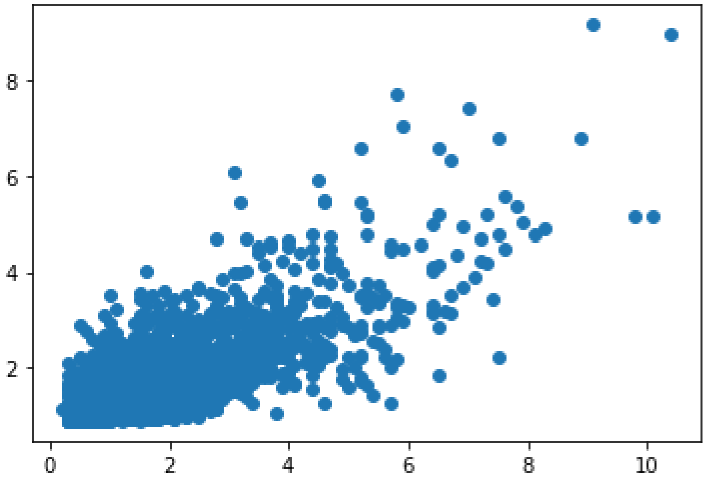
```
la=Lasso(alpha=5)
la.fit(x_train,y_train)
```

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)
```

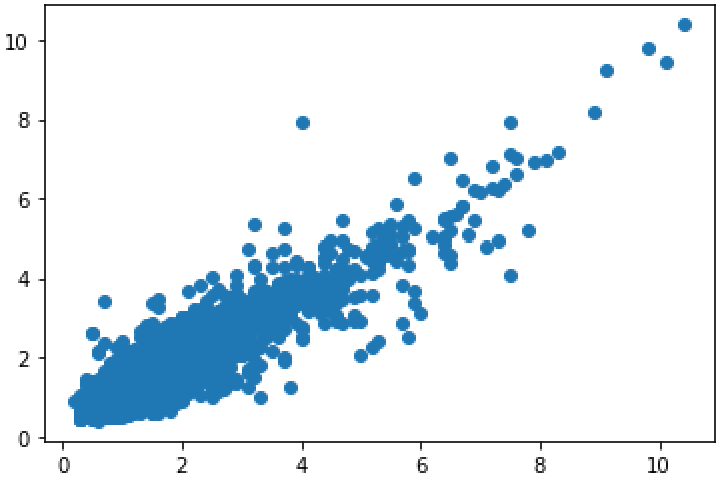
Ridge

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)
```

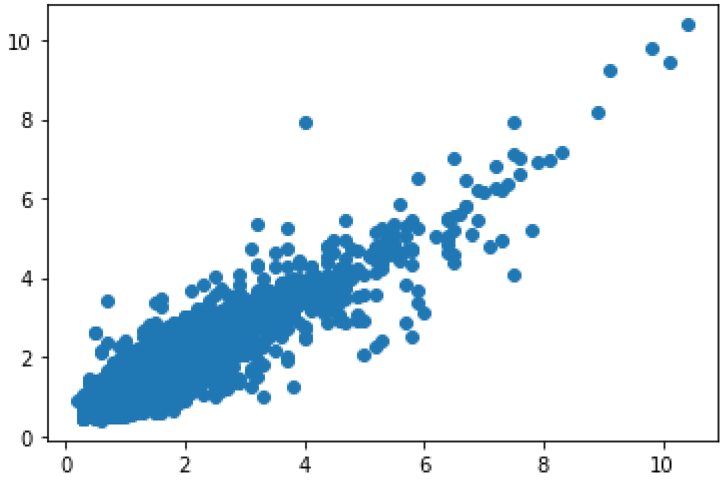
ElasticNet

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
```

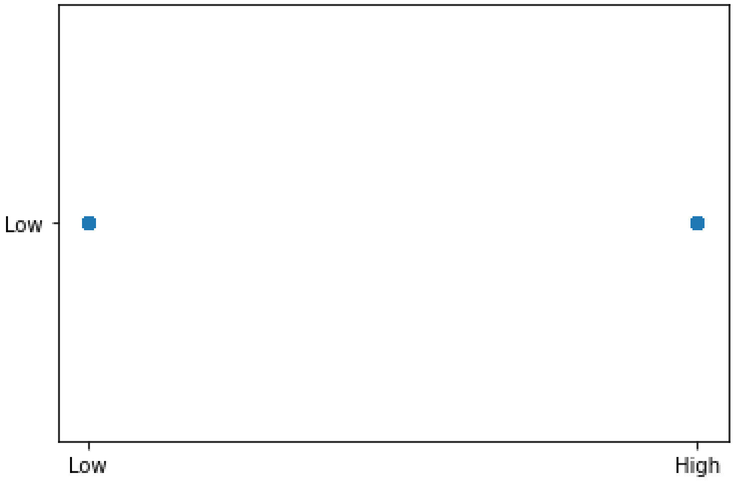
In [25]:
```
x=df1.drop(["TCH"],axis=1)
y=df1["TCH"]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

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]:

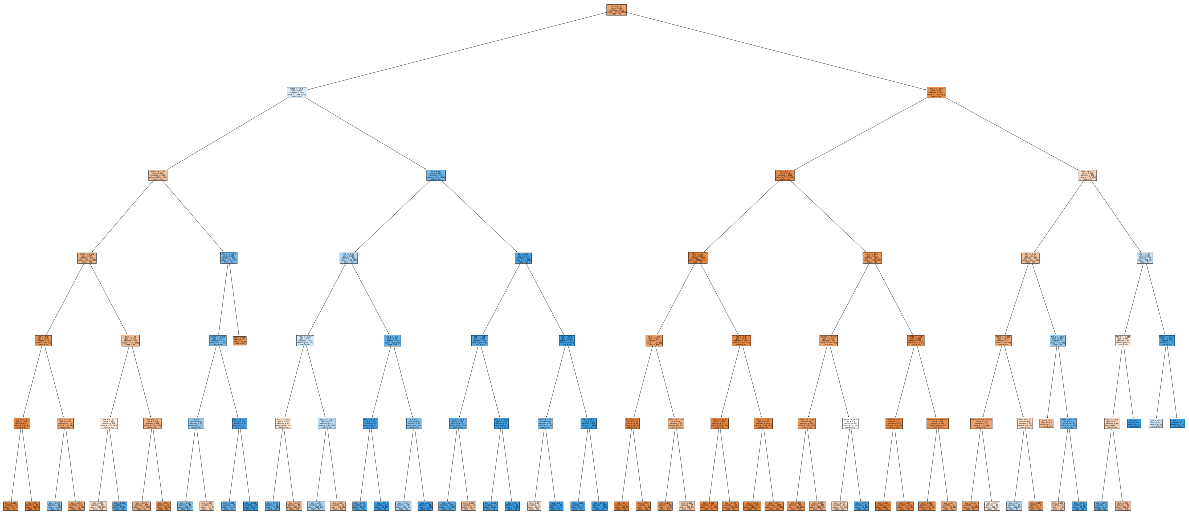
```
rfc_best=grid_search.best_estimator_
```

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\nsamp
les = 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\nsa
mples = 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\nsampl
es = 91\nvalue = [130, 3]\nclass = Yes'),
 Text(40.95412844036697, 155.3142857142857, 'gini = 0.117\nsamples = 35\nvalu
e = [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\nsam
ples = 128\nvalue = [152, 44]\nclass = Yes'),
 Text(204.77064220183485, 155.3142857142857, 'gini = 0.408\nsamples = 12\nval
ue = [4, 10]\nclass = No'),
 Text(286.6788990825688, 155.3142857142857, 'gini = 0.304\nsamples = 116\nval
ue = [148, 34]\nclass = Yes'),
 Text(491.44954128440367, 776.5714285714287, 'SO_2 <= 3.5\ngini = 0.431\nsamp
les = 696\nvalue = [761, 349]\nclass = Yes'),
 Text(409.5412844036697, 465.9428571428573, 'EBE <= 1.85\ngini = 0.495\nsampl
es = 198\nvalue = [171, 141]\nclass = Yes'),
 Text(368.58715596330273, 155.3142857142857, 'gini = 0.485\nsamples = 181\nva
lue = [168, 119]\nclass = Yes'),
 Text(450.4954128440367, 155.3142857142857, 'gini = 0.211\nsamples = 17\nvalu
e = [3, 22]\nclass = No'),
 Text(573.3577981651376, 465.9428571428573, 'BEN <= 1.75\ngini = 0.385\nsampl
es = 498\nvalue = [590, 208]\nclass = Yes'),
 Text(532.4036697247707, 155.3142857142857, 'gini = 0.396\nsamples = 467\nval
ue = [537, 201]\nclass = Yes'),
 Text(614.3119266055046, 155.3142857142857, 'gini = 0.206\nsamples = 31\nvalu
e = [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\nsamp
les = 118\nvalue = [41, 156]\nclass = No'),
 Text(737.1743119266055, 465.9428571428573, 'CO <= 0.55\ngini = 0.434\nsample
s = 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\nsamp
les = 53\nvalue = [6, 81]\nclass = No'),
 Text(860.0366972477065, 155.3142857142857, 'gini = 0.302\nsamples = 16\nvalu
e = [5, 22]\nclass = No'),
 Text(941.9449541284404, 155.3142857142857, 'gini = 0.033\nsamples = 37\nvalu
e = [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\nsam
ples = 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]:

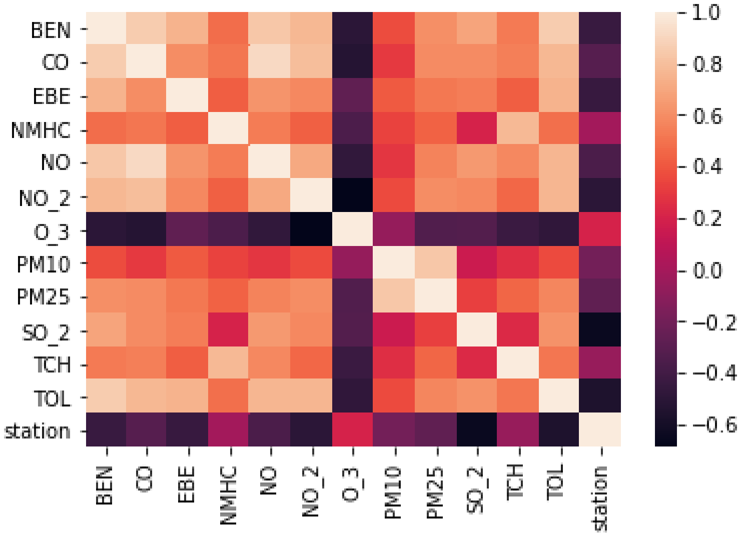
	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL	s
6	2012-09-01 01:00:00	0.4	0.2	0.8	0.24	1.0	7.0	57.0	11.0	7.0	2.0	1.33	0.6	280
30	2012-09-01 02:00:00	0.4	0.2	0.7	0.24	1.0	5.0	55.0	5.0	5.0	2.0	1.33	0.5	280
54	2012-09-01 03:00:00	0.4	0.2	0.7	0.24	1.0	4.0	56.0	6.0	4.0	2.0	1.33	0.5	280
78	2012-09-01 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

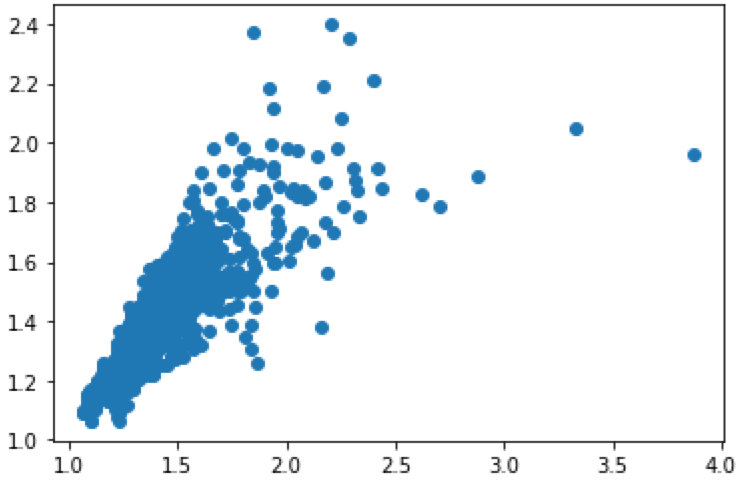
In [45]:
```
li=LinearRegression()
li.fit(x_train,y_train)
```

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 [ ]:
```

```

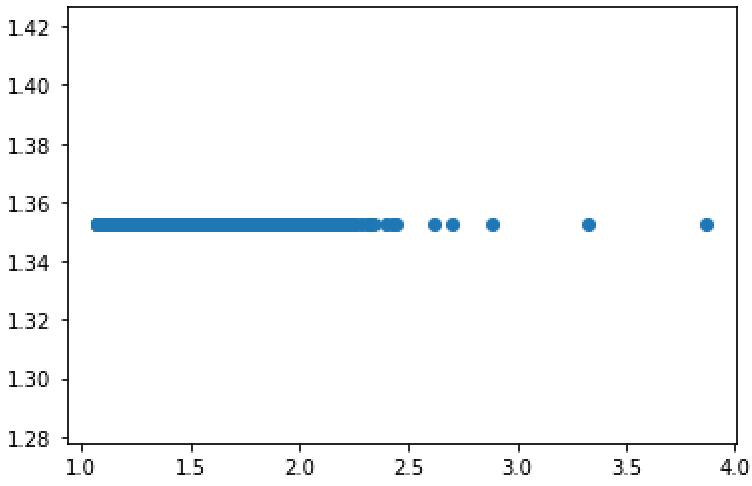
Lasso

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)
```

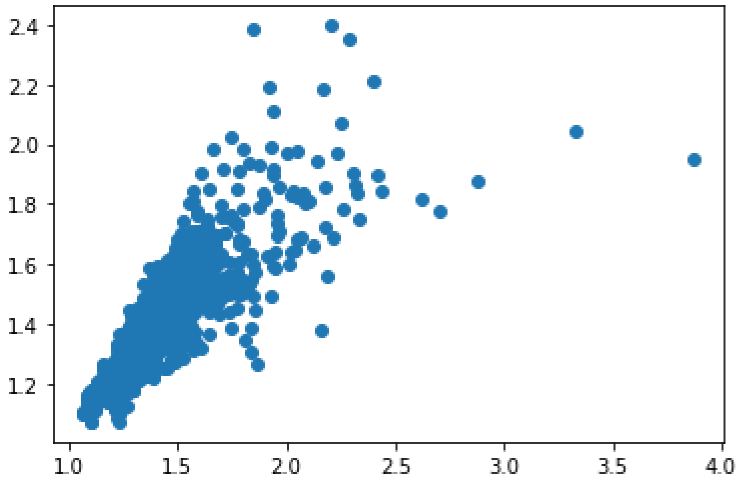
Ridge

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)
```

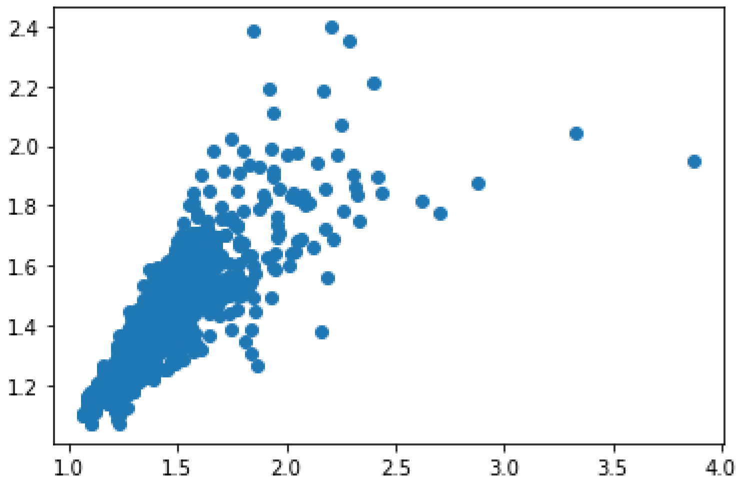
ElasticNet

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
```

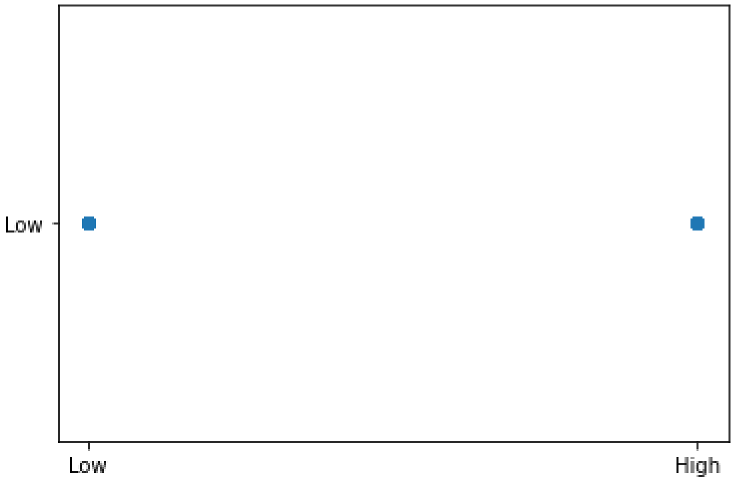
```
In [61]: x=df3.drop(["TCH"],axis=1)
         y=df3["TCH"]
         x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

```
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_
```

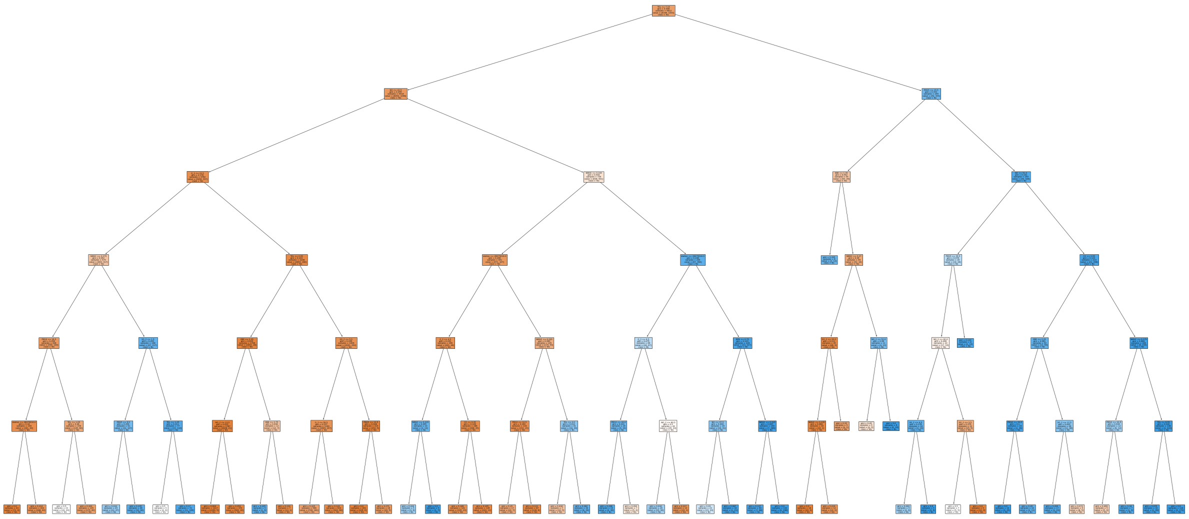
In [73]:

```
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 = 48
12\nvalue = [6188, 1453]\nclass = Yes'),
 Text(1488.0, 1708.457142857143, 'NO <= 22.5\ngini = 0.259\nsamples = 4518\nv
alue = [6094, 1098]\nclass = Yes'),
 Text(744.0, 1397.8285714285716, 'O_3 <= 23.5\ngini = 0.178\nsamples = 3782\n
value = [5456, 597]\nclass = Yes'),
 Text(372.0, 1087.2, 'NMHC <= 0.275\ngini = 0.452\nsamples = 574\nvalue = [60
3, 317]\nclass = Yes'),
 Text(186.0, 776.5714285714287, 'PM10 <= 16.5\ngini = 0.258\nsamples = 406\nv
alue = [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]\ncl
ass = Yes'),
 Text(139.5, 155.3142857142857, 'gini = 0.214\nsamples = 204\nvalue = [288, 4
0]\nclass = Yes'),
 Text(279.0, 465.9428571428573, 'TOL <= 1.05\ngini = 0.36\nsamples = 158\nval
ue = [195, 60]\nclass = Yes'),
 Text(232.5, 155.3142857142857, 'gini = 0.5\nsamples = 23\nvalue = [20, 20]\n
class = Yes'),
 Text(325.5, 155.3142857142857, 'gini = 0.303\nsamples = 135\nvalue = [175, 4
0]\nclass = Yes'),
 Text(558.0, 776.5714285714287, 'SO_2 <= 1.5\ngini = 0.293\nsamples = 168\nva
lue = [47, 217]\nclass = No'),
 Text(465.0, 465.9428571428573, 'PM25 <= 9.5\ngini = 0.354\nsamples = 91\nval
ue = [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\nval
ue = [15, 110]\nclass = No'),
 Text(604.5, 155.3142857142857, 'gini = 0.5\nsamples = 5\nvalue = [4, 4]\ncla
ss = 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 = [48
53, 280]\nclass = Yes'),
 Text(930.0, 776.5714285714287, 'EBE <= 2.25\ngini = 0.019\nsamples = 1799\nv
alue = [2852, 28]\nclass = Yes'),
 Text(837.0, 465.9428571428573, 'NO_2 <= 13.5\ngini = 0.015\nsamples = 1787\n
value = [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, 1
6]\nclass = Yes'),
 Text(1023.0, 465.9428571428573, 'EBE <= 2.75\ngini = 0.444\nsamples = 12\nva
lue = [14, 7]\nclass = Yes'),
 Text(976.5, 155.3142857142857, 'gini = 0.245\nsamples = 5\nvalue = [1, 6]\nc
lass = 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\n
value = [2001, 252]\nclass = Yes'),
 Text(1209.0, 465.9428571428573, 'O_3 <= 50.5\ngini = 0.227\nsamples = 1190\n
value = [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