In [130]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
import re
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
```

In [496]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C10_air\madrid_2012.csv")
a
```

Out[496]:

	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
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
2	2012-09-01 01:00:00	0.4	NaN	0.7	NaN	2.0	10.0	NaN	NaN	NaN	NaN	NaN	1.5
3	2012-09-01 01:00:00	NaN	0.2	NaN	NaN	1.0	6.0	50.0	NaN	NaN	NaN	NaN	NaN
4	2012-09-01 01:00:00	NaN	NaN	NaN	NaN	1.0	13.0	54.0	NaN	NaN	3.0	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
210715	2012-03-01 00:00:00	NaN	0.6	NaN	NaN	37.0	84.0	14.0	NaN	NaN	NaN	NaN	NaN
210716	2012-03-01 00:00:00	NaN	0.4	NaN	NaN	5.0	76.0	NaN	17.0	NaN	7.0	NaN	NaN
210717	2012-03-01 00:00:00	NaN	NaN	NaN	0.34	3.0	41.0	24.0	NaN	NaN	NaN	1.34	NaN
210718	2012-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	44.0	36.0	NaN	NaN	NaN	NaN	NaN
210719	2012-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	56.0	40.0	18.0	NaN	NaN	NaN	NaN

210720 rows × 14 columns

In [497]:

```
a.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 [498]:

```
b=a.fillna(value=104)
b
```

Out[498]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH
0	2012-09-01 01:00:00	104.0	0.2	104.0	104.00	7.0	18.0	104.0	104.0	104.0	2.0	104.00
1	2012-09-01 01:00:00	0.3	0.3	0.7	104.00	3.0	18.0	55.0	10.0	9.0	1.0	104.00
2	2012-09-01 01:00:00	0.4	104.0	0.7	104.00	2.0	10.0	104.0	104.0	104.0	104.0	104.00
3	2012-09-01 01:00:00	104.0	0.2	104.0	104.00	1.0	6.0	50.0	104.0	104.0	104.0	104.00
4	2012-09-01 01:00:00	104.0	104.0	104.0	104.00	1.0	13.0	54.0	104.0	104.0	3.0	104.00
...	...	...	...	...	...	...	...	...	...	...	...	...
210715	2012-03-01 00:00:00	104.0	0.6	104.0	104.00	37.0	84.0	14.0	104.0	104.0	104.0	104.00
210716	2012-03-01 00:00:00	104.0	0.4	104.0	104.00	5.0	76.0	104.0	17.0	104.0	7.0	104.00
210717	2012-03-01 00:00:00	104.0	104.0	104.0	0.34	3.0	41.0	24.0	104.0	104.0	104.0	1.34
210718	2012-03-01 00:00:00	104.0	104.0	104.0	104.00	2.0	44.0	36.0	104.0	104.0	104.0	104.00
210719	2012-03-01 00:00:00	104.0	104.0	104.0	104.00	2.0	56.0	40.0	18.0	104.0	104.0	104.00

210720 rows × 14 columns

In [499]:

```
b.columns
```

Out[499]:

```
Index(['date', 'BEN', 'CO', 'EBE', 'NMHC', 'NO', 'NO_2', 'O_3', 'PM10', 'P
M25',
       'SO_2', 'TCH', 'TOL', 'station'],
      dtype='object')
```

In [500]:

```
c=b.head(10)
c
```

Out[500]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL
0	2012-09-01 01:00:00	104.0	0.2	104.0	104.00	7.0	18.0	104.0	104.0	104.0	2.0	104.00	104.0
1	2012-09-01 01:00:00	0.3	0.3	0.7	104.00	3.0	18.0	55.0	10.0	9.0	1.0	104.00	2.4
2	2012-09-01 01:00:00	0.4	104.0	0.7	104.00	2.0	10.0	104.0	104.0	104.0	104.0	104.00	1.5
3	2012-09-01 01:00:00	104.0	0.2	104.0	104.00	1.0	6.0	50.0	104.0	104.0	104.0	104.00	104.0
4	2012-09-01 01:00:00	104.0	104.0	104.0	104.00	1.0	13.0	54.0	104.0	104.0	3.0	104.00	104.0
5	2012-09-01 01:00:00	0.2	0.2	1.0	104.00	1.0	9.0	57.0	14.0	104.0	1.0	104.00	0.2
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
7	2012-09-01 01:00:00	104.0	104.0	104.0	0.11	1.0	2.0	65.0	104.0	104.0	104.0	1.18	104.0
8	2012-09-01 01:00:00	104.0	0.2	104.0	104.00	6.0	14.0	57.0	104.0	104.0	2.0	104.00	104.0
9	2012-09-01 01:00:00	104.0	0.2	104.0	104.00	1.0	7.0	104.0	13.0	104.0	1.0	104.00	104.0

In [501]:

```
d=c[['BEN', 'CO', 'EBE', 'NMHC', 'NO_2', 'O_3',
 'PM10', 'SO_2', 'TCH', 'TOL', 'station']]
d
```

Out[501]:

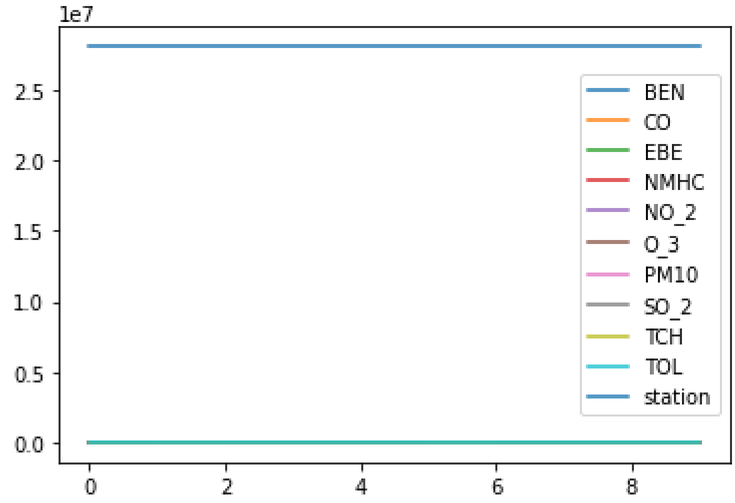
	BEN	CO	EBE	NMHC	NO_2	O_3	PM10	SO_2	TCH	TOL	station
0	104.0	0.2	104.0	104.00	18.0	104.0	104.0	2.0	104.00	104.0	28079004
1	0.3	0.3	0.7	104.00	18.0	55.0	10.0	1.0	104.00	2.4	28079008
2	0.4	104.0	0.7	104.00	10.0	104.0	104.0	104.0	104.00	1.5	28079011
3	104.0	0.2	104.0	104.00	6.0	50.0	104.0	104.0	104.00	104.0	28079016
4	104.0	104.0	104.0	104.00	13.0	54.0	104.0	3.0	104.00	104.0	28079017
5	0.2	0.2	1.0	104.00	9.0	57.0	14.0	1.0	104.00	0.2	28079018
6	0.4	0.2	0.8	0.24	7.0	57.0	11.0	2.0	1.33	0.6	28079024
7	104.0	104.0	104.0	0.11	2.0	65.0	104.0	104.0	1.18	104.0	28079027
8	104.0	0.2	104.0	104.00	14.0	57.0	104.0	2.0	104.00	104.0	28079035
9	104.0	0.2	104.0	104.00	7.0	104.0	13.0	1.0	104.00	104.0	28079036

In [502]:

```
d.plot.line()
```

Out[502]:

```
<AxesSubplot:>
```

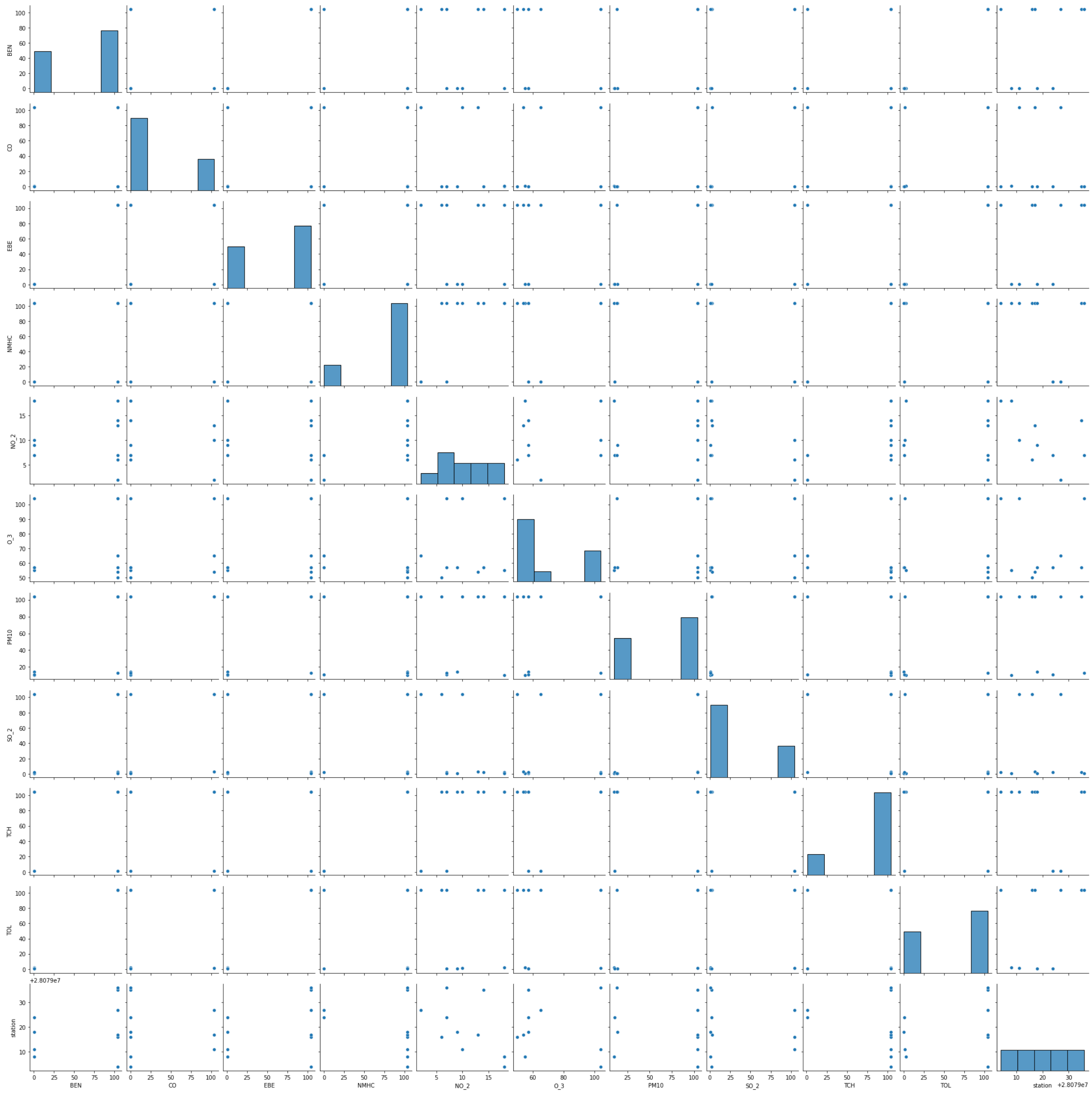


In [503]:

```
sns.pairplot(d)
```

Out[503]:

```
<seaborn.axisgrid.PairGrid at 0x11861bc7640>
```



In [504]:

```
x=d[['BEN', 'CO', 'EBE', 'NMHC', 'NO_2']]
y=d['TCH']
```

In [505]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3)
```

In [506]:

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[506]:

```
LinearRegression()
```

In [507]:

```
print(lr.intercept_)
```

```
1.079995522033144
```

In [508]:

```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[508]:

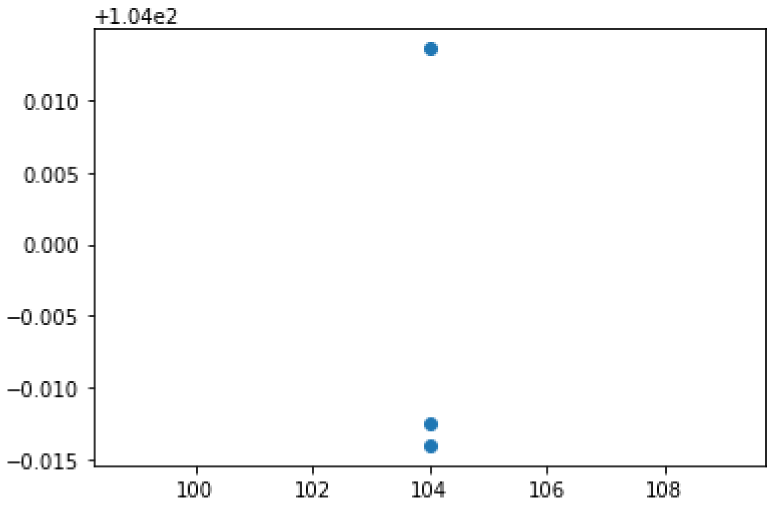
	Co-efficient
BEN	-4.486966e-04
CO	-5.579009e-07
EBE	3.554396e-04
NMHC	9.894886e-01
NO_2	1.481931e-03

In [509]:

```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[509]:

```
<matplotlib.collections.PathCollection at 0x1186acf8580>
```



In [510]:

```
print(lr.score(x_test,y_test))
```

```
0.0
```

In [511]:

```
from sklearn.linear_model import Ridge,Lasso
```

In [512]:

```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[512]:

```
Ridge(alpha=10)
```

In [513]:

```
rr.score(x_test,y_test)
```

Out[513]:

```
0.0
```

In [514]:

```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[514]:

```
Lasso(alpha=10)
```

In [515]:

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

Out[515]:

```
0.0
```

In [516]:

```
a1=b.head(7000)
a1
```

Out[516]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TC
0	2012-09-01 01:00:00	104.0	0.2	104.0	104.0	7.0	18.0	104.0	104.0	104.0	2.0	104.0	104
1	2012-09-01 01:00:00	0.3	0.3	0.7	104.0	3.0	18.0	55.0	10.0	9.0	1.0	104.0	2
2	2012-09-01 01:00:00	0.4	104.0	0.7	104.0	2.0	10.0	104.0	104.0	104.0	104.0	104.0	1
3	2012-09-01 01:00:00	104.0	0.2	104.0	104.0	1.0	6.0	50.0	104.0	104.0	104.0	104.0	104
4	2012-09-01 01:00:00	104.0	104.0	104.0	104.0	1.0	13.0	54.0	104.0	104.0	3.0	104.0	104
...	...	...	...	...	...	...	...	...	...	...	...	...	
6995	2012-09-13 04:00:00	104.0	0.1	104.0	104.0	1.0	5.0	51.0	104.0	104.0	104.0	104.0	104
6996	2012-09-13 04:00:00	104.0	104.0	104.0	104.0	1.0	6.0	104.0	5.0	104.0	2.0	104.0	104
6997	2012-09-13 04:00:00	104.0	104.0	104.0	104.0	1.0	6.0	104.0	7.0	6.0	104.0	104.0	104
6998	2012-09-13 04:00:00	104.0	104.0	104.0	104.0	1.0	9.0	104.0	5.0	1.0	104.0	104.0	104
6999	2012-09-13 04:00:00	104.0	104.0	104.0	104.0	1.0	5.0	43.0	104.0	104.0	104.0	104.0	104

7000 rows × 14 columns

In [517]:

```
e=a1[['BEN', 'CO', 'EBE', 'NMHC', 'NO_2', 'O_3',
 'PM10', 'SO_2', 'TCH', 'TOL', 'station']]
```

In [518]:

```
f=e.iloc[:,0:14]
g=e.iloc[:,-1]
```

In [519]:

```
h=StandardScaler().fit_transform(f)
```

In [520]:

```
logr=LogisticRegression(max_iter=10000)
logr.fit(h,g)
```

Out[520]:

```
LogisticRegression(max_iter=10000)
```

In [521]:

```
from sklearn.model_selection import train_test_split
h_train,h_test,g_train,g_test=train_test_split(h,g,test_size=0.3)
```

In [522]:

```
i=[[10,20,30,40,50,60,11,22,33,44,55]]
```

In [523]:

```
prediction=logr.predict(i)
print(prediction)
```

```
[28079059]
```

In [524]:

```
logr.classes_
```

Out[524]:

```
array([28079004, 28079008, 28079011, 28079016, 28079017, 28079018,
       28079024, 28079027, 28079035, 28079036, 28079038, 28079039,
       28079040, 28079047, 28079048, 28079049, 28079050, 28079054,
       28079055, 28079056, 28079057, 28079058, 28079059, 28079060],
      dtype=int64)
```

In [525]:

```
logr.predict_proba(i)[0][0]
```

Out[525]:

```
0.0
```

In [526]:

```
logr.predict_proba(i)[0][1]
```

Out[526]:

```
0.0
```

In [527]:

```
logr.score(h_test,g_test)
```

Out[527]:

```
0.959047619047619
```

In [528]:

```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[528]:

```
ElasticNet()
```

In [529]:

```
print(en.coef_)
```

```
[-0.         -0.         -0.          0.98914588  0.        ]
```

In [530]:

```
print(en.intercept_)
```

```
1.1154203887232939
```

In [531]:

```
prediction=en.predict(x_test)
print(en.score(x_test,y_test))
```

```
0.0
```

In [532]:

```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(h_train,g_train)
```

Out[532]:

```
RandomForestClassifier()
```

In [533]:

```
parameters={'max_depth':[1,2,3,4,5],
 'min_samples_leaf':[5,10,15,20,25],
 'n_estimators':[10,20,30,40,50]
 }
```

In [534]:

```
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
grid_search.fit(h_train,g_train)
```

Out[534]:

```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [1, 2, 3, 4, 5],
                         'min_samples_leaf': [5, 10, 15, 20, 25],
                         'n_estimators': [10, 20, 30, 40, 50]},
             scoring='accuracy')
```

In [535]:

```
grid_search.best_score_
```

Out[535]:

```
0.9957142857142858
```

In [536]:

```
rfc_best=grid_search.best_estimator_
```

In [537]:

```
from sklearn.tree import plot_tree
plt.figure(figsize=(80,50))
plot_tree(rfc_best.estimators_[2],filled=True)
```

```
3096\nvalue = [191, 192, 202, 190, 170, 214, 204, 216, 221, 210\n221, 20
3, 204, 177, 221, 231, 195, 228, 194, 201\n204, 225, 191, 195]'),
 Text(1083.1764705882351, 2038.5, 'X[9] <= -0.309\ngini = 0.899\nsamples
= 1274\nvalue = [191, 191, 0, 0, 170, 213, 202, 0, 220, 210, 221\n0, 203,
0, 0, 0, 0, 0, 0, 0, 204, 0, 0, 0]'),
 Text(393.88235294117646, 1585.5, 'X[8] <= -1.127\ngini = 0.749\nsamples
= 511\nvalue = [0, 191, 0, 0, 0, 210, 202, 0, 0, 0, 221, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(262.5882352941176, 1132.5, 'gini = 0.0\nsamples = 131\nvalue = [0,
0, 0, 0, 0, 0, 200, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(525.1764705882352, 1132.5, 'X[0] <= -1.748\ngini = 0.668\nsamples =
380\nvalue = [0, 191, 0, 0, 0, 210, 2, 0, 0, 0, 221, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0]'),
 Text(262.5882352941176, 679.5, 'X[9] <= -1.777\ngini = 0.555\nsamples =
203\nvalue = [0, 31, 0, 0, 0, 189, 0, 0, 0, 0, 114, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0]'),
 Text(131.2941176470588, 226.5, 'gini = 0.458\nsamples = 143\nvalue = [0,
14, 0, 0, 0, 172, 0, 0, 0, 0, 64, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0]'),
 Text(393.88235294117646, 226.5, 'gini = 0.564\nsamples = 60\nvalue = [0,
```

In [ ]:

```
```

In [ ]:

```
```