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 [580]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C10_air\madrid_2014.csv")
a
```

Out[580]:

	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
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
2	2014-06-01 01:00:00	0.3	NaN	0.1	NaN	2.0	6.0	NaN	NaN	NaN	NaN	NaN	1.1
3	2014-06-01 01:00:00	NaN	0.2	NaN	NaN	1.0	6.0	79.0	NaN	NaN	NaN	NaN	NaN
4	2014-06-01 01:00:00	NaN	NaN	NaN	NaN	1.0	6.0	75.0	NaN	NaN	4.0	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
210019	2014-09-01 00:00:00	NaN	0.5	NaN	NaN	20.0	84.0	29.0	NaN	NaN	NaN	NaN	NaN
210020	2014-09-01 00:00:00	NaN	0.3	NaN	NaN	1.0	22.0	NaN	15.0	NaN	6.0	NaN	NaN
210021	2014-09-01 00:00:00	NaN	NaN	NaN	NaN	1.0	13.0	70.0	NaN	NaN	NaN	NaN	NaN
210022	2014-09-01 00:00:00	NaN	NaN	NaN	NaN	3.0	38.0	42.0	NaN	NaN	NaN	NaN	NaN
210023	2014-09-01 00:00:00	NaN	NaN	NaN	NaN	1.0	26.0	65.0	11.0	NaN	NaN	NaN	NaN

210024 rows × 14 columns

In [581]:

```
a.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 [582]:

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

Out[582]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH
0	2014-06-01 01:00:00	104.0	0.2	104.0	104.00	3.0	10.0	104.0	104.0	104.0	3.0	104.00
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
2	2014-06-01 01:00:00	0.3	104.0	0.1	104.00	2.0	6.0	104.0	104.0	104.0	104.0	104.00
3	2014-06-01 01:00:00	104.0	0.2	104.0	104.00	1.0	6.0	79.0	104.0	104.0	104.0	104.00
4	2014-06-01 01:00:00	104.0	104.0	104.0	104.00	1.0	6.0	75.0	104.0	104.0	4.0	104.00
...	...	...	...	...	...	...	...	...	...	...	...	...
210019	2014-09-01 00:00:00	104.0	0.5	104.0	104.00	20.0	84.0	29.0	104.0	104.0	104.0	104.00
210020	2014-09-01 00:00:00	104.0	0.3	104.0	104.00	1.0	22.0	104.0	15.0	104.0	6.0	104.00
210021	2014-09-01 00:00:00	104.0	104.0	104.0	104.00	1.0	13.0	70.0	104.0	104.0	104.0	104.00
210022	2014-09-01 00:00:00	104.0	104.0	104.0	104.00	3.0	38.0	42.0	104.0	104.0	104.0	104.00
210023	2014-09-01 00:00:00	104.0	104.0	104.0	104.00	1.0	26.0	65.0	11.0	104.0	104.0	104.00

210024 rows × 14 columns

In [583]:

```
b.columns
```

Out[583]:

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

In [584]:

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

Out[584]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL
0	2014-06-01 01:00:00	104.0	0.2	104.0	104.00	3.0	10.0	104.0	104.0	104.0	3.0	104.00	104.0
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
2	2014-06-01 01:00:00	0.3	104.0	0.1	104.00	2.0	6.0	104.0	104.0	104.0	104.0	104.00	1.1
3	2014-06-01 01:00:00	104.0	0.2	104.0	104.00	1.0	6.0	79.0	104.0	104.0	104.0	104.00	104.0
4	2014-06-01 01:00:00	104.0	104.0	104.0	104.00	1.0	6.0	75.0	104.0	104.0	4.0	104.00	104.0
5	2014-06-01 01:00:00	0.1	0.4	0.1	104.00	1.0	10.0	83.0	7.0	104.0	2.0	104.00	0.2
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
7	2014-06-01 01:00:00	104.0	104.0	104.0	104.00	1.0	1.0	86.0	104.0	104.0	104.0	104.00	104.0
8	2014-06-01 01:00:00	104.0	0.3	104.0	104.00	5.0	22.0	68.0	104.0	104.0	4.0	104.00	104.0
9	2014-06-01 01:00:00	104.0	0.2	104.0	104.00	1.0	4.0	104.0	14.0	104.0	1.0	104.00	104.0

In [585]:

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

Out[585]:

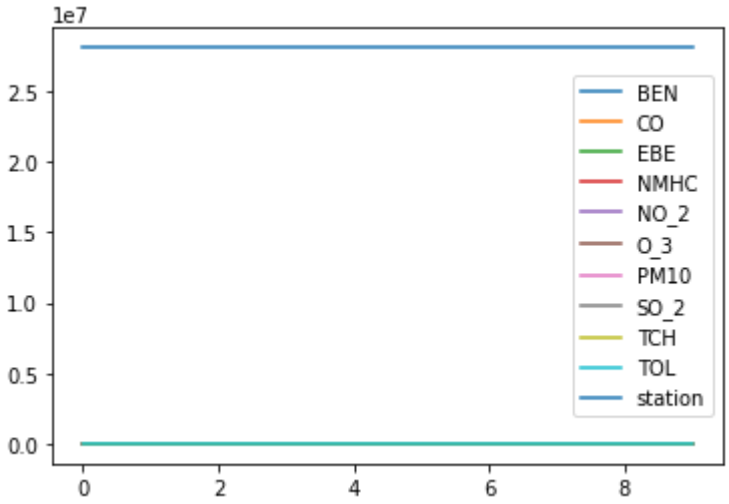
	BEN	CO	EBE	NMHC	NO_2	O_3	PM10	SO_2	TCH	TOL	station
0	104.0	0.2	104.0	104.00	10.0	104.0	104.0	3.0	104.00	104.0	28079004
1	0.2	0.2	0.1	0.11	17.0	68.0	10.0	5.0	1.36	1.3	28079008
2	0.3	104.0	0.1	104.00	6.0	104.0	104.0	104.0	104.00	1.1	28079011
3	104.0	0.2	104.0	104.00	6.0	79.0	104.0	104.0	104.00	104.0	28079016
4	104.0	104.0	104.0	104.00	6.0	75.0	104.0	4.0	104.00	104.0	28079017
5	0.1	0.4	0.1	104.00	10.0	83.0	7.0	2.0	104.00	0.2	28079018
6	0.1	0.2	0.1	0.23	5.0	80.0	4.0	2.0	1.21	0.1	28079024
7	104.0	104.0	104.0	104.00	1.0	86.0	104.0	104.0	104.00	104.0	28079027
8	104.0	0.3	104.0	104.00	22.0	68.0	104.0	4.0	104.00	104.0	28079035
9	104.0	0.2	104.0	104.00	4.0	104.0	14.0	1.0	104.00	104.0	28079036

In [586]:

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

Out[586]:

```
<AxesSubplot:>
```

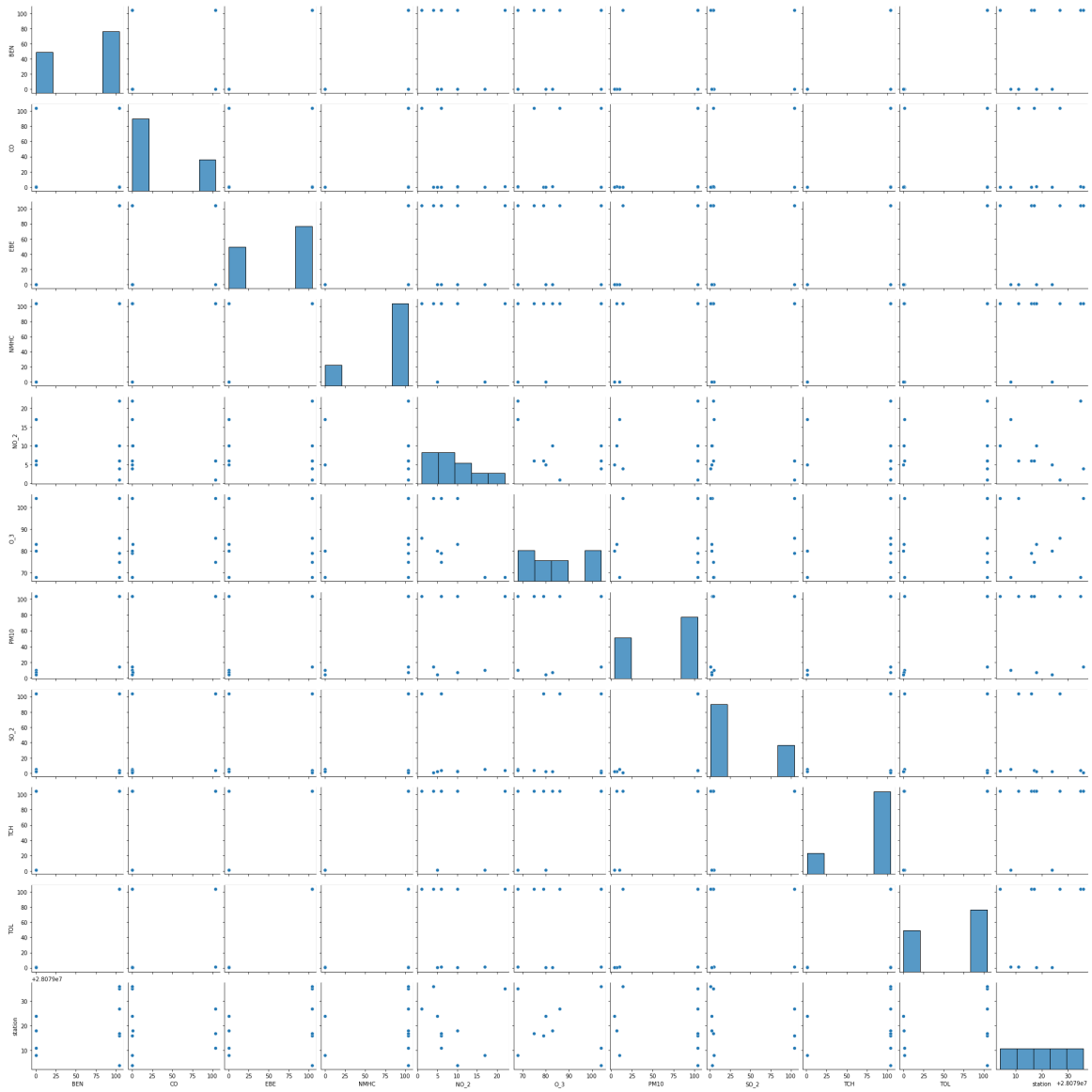


In [587]:

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

Out[587]:

```
<seaborn.axisgrid.PairGrid at 0x11877417580>
```



In [588]:

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

In [589]:

```
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 [590]:

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

Out[590]:

```
LinearRegression()
```

In [591]:

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

```
1.2513711292464365
```

In [592]:

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

Out[592]:

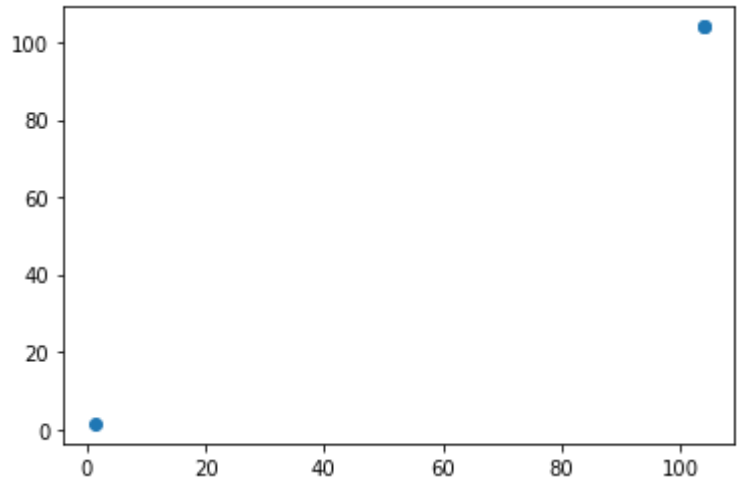
	Co-efficient
BEN	-4.754873e-04
CO	1.093141e-16
EBE	4.754873e-04
NMHC	9.879676e-01
NO_2	-1.890190e-17

In [593]:

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

Out[593]:

```
<matplotlib.collections.PathCollection at 0x118834f6340>
```



In [594]:

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

```
0.9999897573255686
```

In [595]:

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

In [596]:

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

Out[596]:

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

In [597]:

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

Out[597]:

```
0.9999757641054597
```

In [598]:

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

Out[598]:

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

In [599]:

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

Out[599]:

```
0.9998705926381068
```

In [600]:

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

Out[600]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	T
0	2014-06-01 01:00:00	104.0	0.2	104.0	104.00	3.0	10.0	104.0	104.0	104.0	3.0	104.00	10
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	
2	2014-06-01 01:00:00	0.3	104.0	0.1	104.00	2.0	6.0	104.0	104.0	104.0	104.0	104.00	
3	2014-06-01 01:00:00	104.0	0.2	104.0	104.00	1.0	6.0	79.0	104.0	104.0	104.0	104.00	10
4	2014-06-01 01:00:00	104.0	104.0	104.0	104.00	1.0	6.0	75.0	104.0	104.0	4.0	104.00	10
...	...	...	...	...	...	...	...	...	...	...	...	...	
6995	2014-06-13 04:00:00	104.0	0.2	104.0	104.00	1.0	16.0	63.0	104.0	104.0	104.0	104.00	10
6996	2014-06-13 04:00:00	104.0	104.0	104.0	104.00	3.0	18.0	104.0	22.0	104.0	4.0	104.00	10
6997	2014-06-13 04:00:00	104.0	104.0	104.0	104.00	2.0	17.0	104.0	22.0	15.0	104.0	104.00	10
6998	2014-06-13 04:00:00	104.0	104.0	104.0	104.00	1.0	14.0	104.0	29.0	14.0	104.0	104.00	10
6999	2014-06-13 04:00:00	104.0	104.0	104.0	104.00	3.0	14.0	59.0	104.0	104.0	104.0	104.00	10

7000 rows × 14 columns

In [601]:

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

In [602]:

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

In [603]:

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

In [604]:

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

Out[604]:

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

In [605]:

```
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 [606]:

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

In [607]:

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

```
[28079060]
```

In [608]:

```
logr.classes_
```

Out[608]:

```
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 [609]:

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

Out[609]:

```
0.0
```

In [610]:

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

Out[610]:

```
0.0
```

In [611]:

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

Out[611]:

```
0.9476190476190476
```

In [612]:

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

Out[612]:

```
ElasticNet()
```

In [613]:

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

```
[ 0.00000000e+00  0.00000000e+00  2.84958761e-04  9.86978827e-01
 -0.00000000e+00]
```

In [614]:

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

```
1.318344130042405
```

In [615]:

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

```
0.9999838439802928
```

In [616]:

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

Out[616]:

```
RandomForestClassifier()
```

In [617]:

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

In [618]:

```
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[618]:

```
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 [619]:

```
grid_search.best_score_
```

Out[619]:

```
0.9938775510204082
```

In [620]:

```
rfc_best=grid_search.best_estimator_
```

In [621]:

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

```
alue = [204, 201, 230, 170, 206, 172, 164, 205, 213, 201\n211, 211, 217,
218, 199, 197, 243, 206, 209, 203\n190, 204, 218, 208]'),
 Text(1116.0, 2038.5, 'X[6] <= 0.927\ngini = 0.797\nsamples = 628\nvalue
= [0, 198, 229, 0, 0, 172, 162, 0, 0, 0, 210, 0\n0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0]'),
 Text(697.5, 1585.5, 'X[3] <= -1.141\ngini = 0.747\nsamples = 480\nvalue
= [0, 197, 0, 0, 0, 172, 160, 0, 0, 0, 210, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0]'),
 Text(279.0, 1132.5, 'X[9] <= -2.0\ngini = 0.493\nsamples = 230\nvalue =
[0, 197, 0, 0, 0, 0, 155, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0]'),
 Text(139.5, 679.5, 'gini = 0.0\nsamples = 107\nvalue = [0, 0, 0, 0, 0,
0, 151, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(418.5, 679.5, 'X[3] <= -2.661\ngini = 0.039\nsamples = 123\nvalue =
[0, 197, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0]'),
 Text(279.0, 226.5, 'gini = 0.0\nsamples = 117\nvalue = [0, 193, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(558.0, 226.5, 'gini = 0.5\nsamples = 6\nvalue = [0, 4, 0, 0, 0, 0,
4, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
```

In [ ]: