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 [383]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C10_air\madrid_2010.csv")
a
```

Out[383]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	
0	2010-03-01 01:00:00	NaN	0.29	NaN	NaN	NaN	25.090000	29.219999	NaN	68.930000	
1	2010-03-01 01:00:00	NaN	0.27	NaN	NaN	NaN	24.879999	30.040001	NaN	NaN	
2	2010-03-01 01:00:00	NaN	0.28	NaN	NaN	NaN	17.410000	20.540001	NaN	72.120003	
3	2010-03-01 01:00:00	0.38	0.24	1.74	NaN	0.05	15.610000	21.080000	NaN	72.970001	19
4	2010-03-01 01:00:00	0.79	NaN	1.32	NaN	NaN	21.430000	26.070000	NaN	NaN	24
...	...	...	...	...	...	...	...	...	...	...	
209443	2010-08-01 00:00:00	NaN	0.55	NaN	NaN	NaN	125.000000	219.899994	NaN	25.379999	
209444	2010-08-01 00:00:00	NaN	0.27	NaN	NaN	NaN	45.709999	47.410000	NaN	NaN	51
209445	2010-08-01 00:00:00	NaN	NaN	NaN	NaN	0.24	46.560001	49.040001	NaN	46.250000	
209446	2010-08-01 00:00:00	NaN	NaN	NaN	NaN	NaN	46.770000	50.119999	NaN	77.709999	
209447	2010-08-01 00:00:00	0.92	0.43	0.71	NaN	0.25	76.330002	88.190002	NaN	52.259998	47

209448 rows × 17 columns

In [384]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209448 entries, 0 to 209447
Data columns (total 17 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     209448 non-null  object 
 1   BEN      60268 non-null   float64
 2   CO       94982 non-null   float64
 3   EBE      60253 non-null   float64
 4   MXY      6750 non-null    float64
 5   NMHC     51727 non-null   float64
 6   NO_2     208219 non-null  float64
 7   NOx      208210 non-null  float64
 8   OXY      6750 non-null    float64
 9   O_3      126684 non-null  float64
 10  PM10     106186 non-null  float64
 11  PM25     55514 non-null   float64
 12  PXY      6740 non-null    float64
 13  SO_2     93184 non-null   float64
 14  TCH      51730 non-null   float64
 15  TOL      60171 non-null   float64
 16  station  209448 non-null  int64  
dtypes: float64(15), int64(1), object(1)
memory usage: 27.2+ MB
```

In [385]:

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

Out[385]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	
0	2010-03-01 01:00:00	104.00	0.29	104.00	104.0	104.00	25.090000	29.219999	104.0	68.93
1	2010-03-01 01:00:00	104.00	0.27	104.00	104.0	104.00	24.879999	30.040001	104.0	104.00
2	2010-03-01 01:00:00	104.00	0.28	104.00	104.0	104.00	17.410000	20.540001	104.0	72.12
3	2010-03-01 01:00:00	0.38	0.24	1.74	104.0	0.05	15.610000	21.080000	104.0	72.97
4	2010-03-01 01:00:00	0.79	104.00	1.32	104.0	104.00	21.430000	26.070000	104.0	104.00
...	...	...	...	...	...	...	...	...	...	...
209443	2010-08-01 00:00:00	104.00	0.55	104.00	104.0	104.00	125.000000	219.899994	104.0	25.37
209444	2010-08-01 00:00:00	104.00	0.27	104.00	104.0	104.00	45.709999	47.410000	104.0	104.00
209445	2010-08-01 00:00:00	104.00	104.00	104.00	104.0	0.24	46.560001	49.040001	104.0	46.25
209446	2010-08-01 00:00:00	104.00	104.00	104.00	104.0	104.00	46.770000	50.119999	104.0	77.70
209447	2010-08-01 00:00:00	0.92	0.43	0.71	104.0	0.25	76.330002	88.190002	104.0	52.25

209448 rows × 17 columns

In [386]:

```
b.columns
```

Out[386]:

```
Index(['date', 'BEN', 'CO', 'EBE', 'MXY', 'NMHC', 'NO_2', 'NOx', 'OXY', 'O
_3',
       'PM10', 'PM25', 'PXY', 'SO_2', 'TCH', 'TOL', 'station'],
      dtype='object')
```

In [387]:

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

Out[387]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	
0	2010-03-01 01:00:00	104.00	0.29	104.00	104.0	104.00	25.090000	29.219999	104.0	68.930000	1
1	2010-03-01 01:00:00	104.00	0.27	104.00	104.0	104.00	24.879999	30.040001	104.0	104.000000	1
2	2010-03-01 01:00:00	104.00	0.28	104.00	104.0	104.00	17.410000	20.540001	104.0	72.120003	1
3	2010-03-01 01:00:00	0.38	0.24	1.74	104.0	0.05	15.610000	21.080000	104.0	72.970001	
4	2010-03-01 01:00:00	0.79	104.00	1.32	104.0	104.00	21.430000	26.070000	104.0	104.000000	
5	2010-03-01 01:00:00	0.56	104.00	0.58	104.0	104.00	21.370001	25.870001	104.0	104.000000	1
6	2010-03-01 01:00:00	104.00	104.00	104.00	104.0	104.00	16.660000	25.230000	104.0	104.000000	
7	2010-03-01 01:00:00	104.00	0.23	104.00	104.0	104.00	17.799999	21.639999	104.0	55.880001	1
8	2010-03-01 01:00:00	104.00	104.00	104.00	104.0	104.00	12.050000	14.870000	104.0	57.369999	1
9	2010-03-01 01:00:00	1.48	0.18	0.51	104.0	104.00	16.780001	21.680000	104.0	78.660004	

In [388]:

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

Out[388]:

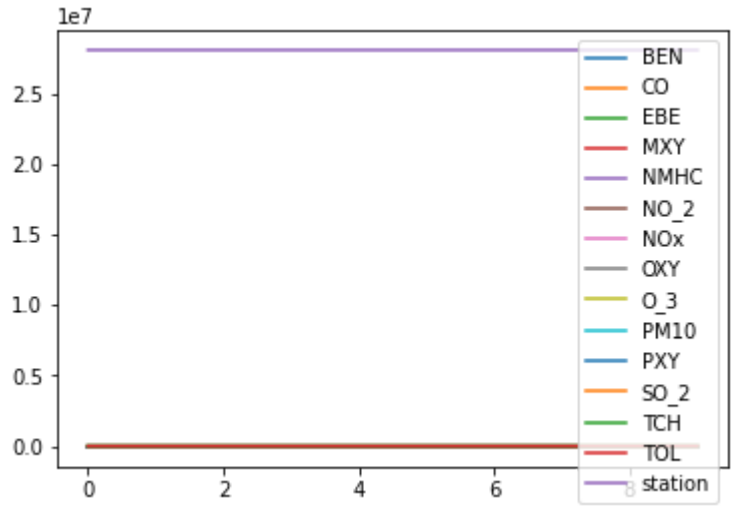
	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	PM10
0	104.00	0.29	104.00	104.0	104.00	25.090000	29.219999	104.0	68.930000	104.000000
1	104.00	0.27	104.00	104.0	104.00	24.879999	30.040001	104.0	104.000000	104.000000
2	104.00	0.28	104.00	104.0	104.00	17.410000	20.540001	104.0	72.120003	104.000000
3	0.38	0.24	1.74	104.0	0.05	15.610000	21.080000	104.0	72.970001	19.410000
4	0.79	104.00	1.32	104.0	104.00	21.430000	26.070000	104.0	104.000000	24.670000
5	0.56	104.00	0.58	104.0	104.00	21.370001	25.870001	104.0	104.000000	104.000000
6	104.00	104.00	104.00	104.0	104.00	16.660000	25.230000	104.0	104.000000	39.799999
7	104.00	0.23	104.00	104.0	104.00	17.799999	21.639999	104.0	55.880001	104.000000
8	104.00	104.00	104.00	104.0	104.00	12.050000	14.870000	104.0	57.369999	104.000000
9	1.48	0.18	0.51	104.0	104.00	16.780001	21.680000	104.0	78.660004	21.969999

In [389]:

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

Out[389]:

```
<AxesSubplot:>
```

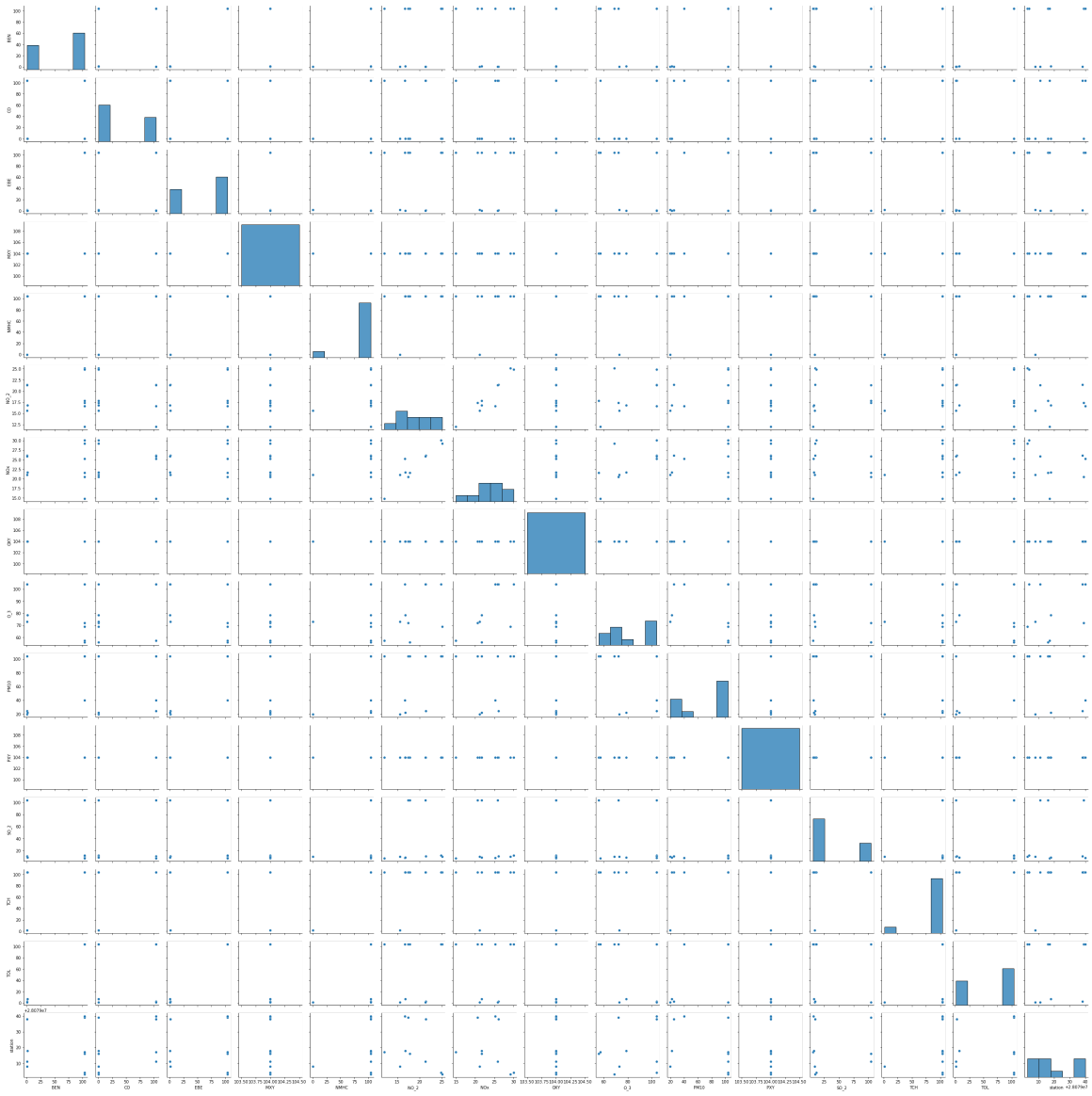


In [390]:

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

Out[390]:

```
<seaborn.axisgrid.PairGrid at 0x1184283c0d0>
```



In [391]:

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

In [392]:

```
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 [393]:

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

Out[393]:

```
LinearRegression()
```

In [394]:

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

```
1.4707070508895583
```

In [395]:

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

Out[395]:

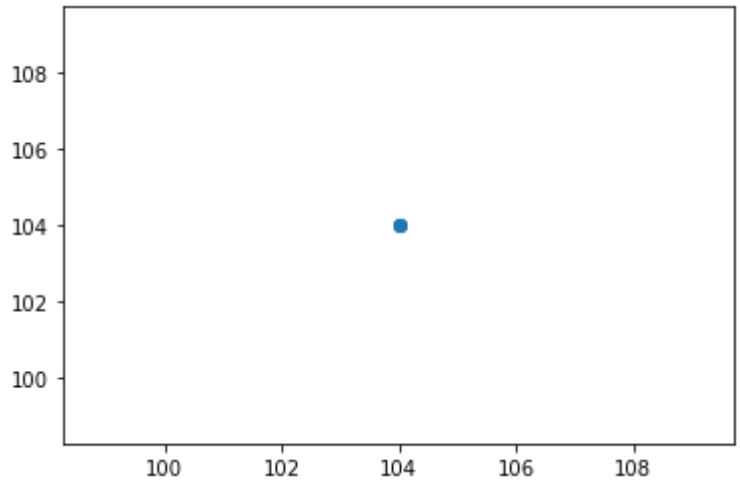
	Co-efficient
BEN	1.319435e-14
CO	1.680341e-16
EBE	-1.325155e-14
MXY	2.220446e-16
NMHC	9.858586e-01
NO_2	-1.759636e-15
NOx	1.965964e-15
OXY	0.000000e+00

In [396]:

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

Out[396]:

```
<matplotlib.collections.PathCollection at 0x11855bb41f0>
```



In [397]:

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

```
0.0
```

In [398]:

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

In [399]:

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

Out[399]:

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

In [400]:

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

Out[400]:

```
0.0
```

In [401]:

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

Out[401]:

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

In [402]:

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

Out[402]:

```
0.0
```

In [403]:

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

Out[403]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O
0	2010-03-01 01:00:00	104.00	0.29	104.00	104.00	104.00	25.090000	29.219999	104.00	68.9300
1	2010-03-01 01:00:00	104.00	0.27	104.00	104.00	104.00	24.879999	30.040001	104.00	104.0000
2	2010-03-01 01:00:00	104.00	0.28	104.00	104.00	104.00	17.410000	20.540001	104.00	72.1200
3	2010-03-01 01:00:00	0.38	0.24	1.74	104.00	0.05	15.610000	21.080000	104.00	72.9700
4	2010-03-01 01:00:00	0.79	104.00	1.32	104.00	104.00	21.430000	26.070000	104.00	104.0000
...	...	...	...	...	...	...	...	...	...	
6995	2010-03-13 06:00:00	0.69	0.26	0.47	0.53	0.23	40.490002	42.220001	0.84	22.1700
6996	2010-03-13 06:00:00	104.00	104.00	104.00	104.00	0.09	52.590000	66.339996	104.00	23.8500
6997	2010-03-13 06:00:00	104.00	104.00	104.00	104.00	104.00	41.950001	44.310001	104.00	104.0000
6998	2010-03-13 06:00:00	104.00	104.00	104.00	104.00	104.00	27.459999	30.540001	104.00	47.3699
6999	2010-03-13 06:00:00	104.00	104.00	104.00	104.00	104.00	36.830002	42.049999	104.00	104.0000

7000 rows × 17 columns

In [404]:

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

In [405]:

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

In [406]:

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

In [407]:

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

Out[407]:

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

In [408]:

```
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 [409]:

```
i=[[10,20,30,40,50,60,11,22,33,44,55,54,21,78]]
```

In [410]:

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

```
[28079004]
```

In [411]:

```
logr.classes_
```

Out[411]:

```
array([28079003, 28079004, 28079008, 28079011, 28079016, 28079017,
       28079018, 28079024, 28079027, 28079036, 28079038, 28079039,
       28079040, 28079047, 28079049, 28079050, 28079054, 28079055,
       28079056, 28079057, 28079058, 28079059, 28079060, 28079099],
      dtype=int64)
```

In [412]:

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

Out[412]:

```
4.637892697435168e-221
```

In [413]:

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

Out[413]:

```
0.9999999999996778
```

In [414]:

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

Out[414]:

```
0.8485714285714285
```

In [415]:

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

Out[415]:

```
ElasticNet()
```

In [416]:

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

```
[9.38132934e-05 1.16034006e-05 0.00000000e+00 0.00000000e+00
 9.85040131e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00]
```

In [417]:

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

```
1.5375459078231017
```

In [418]:

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

```
0.0
```

In [419]:

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

Out[419]:

```
RandomForestClassifier()
```

In [420]:

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

In [421]:

```
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[421]:

```
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 [422]:

```
grid_search.best_score_
```

Out[422]:

```
0.8351020408163266
```

In [423]:

```
rfc_best=grid_search.best_estimator_
```

In [424]:

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

```
e = [205, 151, 189, 200, 229, 231, 193, 221, 172, 226\n214, 209, 212, 22
7, 211, 213, 188, 206, 210, 212\n198, 196, 210, 177]'),
 Text(1116.0, 2038.5, 'X[0] <= -0.421\ngini = 0.908\nsamples = 1407\nvalu
e = [205, 136, 189, 0, 0, 231, 193, 211, 0, 226, 214\n0, 212, 0, 0, 0, 0,
0, 0, 212, 0, 0, 0, 177]'),
 Text(558.0, 1585.5, 'X[12] <= -0.572\ngini = 0.799\nsamples = 653\nvalue
= [0, 0, 189, 0, 0, 0, 193, 211, 0, 0, 214, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 177]'),
 Text(279.0, 1132.5, 'X[10] <= -1.521\ngini = 0.665\nsamples = 395\nvalue
= [0, 0, 189, 0, 0, 0, 0, 210, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0,
0, 177]'),
 Text(139.5, 679.5, 'X[0] <= -1.559\ngini = 0.495\nsamples = 258\nvalue =
[0, 0, 0, 0, 0, 0, 0, 210, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 1
71]'),
 Text(69.75, 226.5, 'gini = 0.374\nsamples = 155\nvalue = [0, 0, 0, 0, 0,
0, 0, 163, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 54]'),
 Text(209.25, 226.5, 'gini = 0.409\nsamples = 103\nvalue = [0, 0, 0, 0,
0, 0, 0, 47, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 117]'),
 Text(418.5, 679.5, 'X[2] <= -1.56\ngini = 0.06\nsamples = 137\nvalue =
[0, 0, 189, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
```

In [ ]: