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 [341]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C10_air\madrid_2009.csv")
a
```

Out[341]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	
0	2009-10-01 01:00:00	NaN	0.27	NaN	NaN	NaN	39.889999	48.150002	NaN	50.680000	18.2
1	2009-10-01 01:00:00	NaN	0.22	NaN	NaN	NaN	21.230000	24.260000	NaN	55.880001	10.5
2	2009-10-01 01:00:00	NaN	0.18	NaN	NaN	NaN	31.230000	34.880001	NaN	49.060001	25.1
3	2009-10-01 01:00:00	0.95	0.33	1.43	2.68	0.25	55.180000	81.360001	1.57	36.669998	26.5
4	2009-10-01 01:00:00	NaN	0.41	NaN	NaN	0.12	61.349998	76.260002	NaN	38.090000	23.7
...	...	...	...	...	...	...	...	...	...	...	
215683	2009-06-01 00:00:00	0.50	0.22	0.39	0.75	0.09	22.000000	24.510000	1.00	82.239998	10.8
215684	2009-06-01 00:00:00	NaN	0.31	NaN	NaN	NaN	76.110001	101.099998	NaN	41.220001	9.9
215685	2009-06-01 00:00:00	0.13	NaN	0.86	NaN	0.23	81.050003	99.849998	NaN	24.830000	12.4
215686	2009-06-01 00:00:00	0.21	NaN	2.96	NaN	0.10	72.419998	82.959999	NaN	NaN	13.0
215687	2009-06-01 00:00:00	0.37	0.32	0.99	1.36	0.14	54.290001	64.480003	1.06	56.919998	15.3

215688 rows × 17 columns

In [342]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215688 entries, 0 to 215687
Data columns (total 17 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     215688 non-null  object 
 1   BEN      60082 non-null   float64
 2   CO       190801 non-null  float64
 3   EBE      60081 non-null   float64
 4   MXY      24846 non-null   float64
 5   NMHC     74748 non-null   float64
 6   NO_2     214562 non-null  float64
 7   NOx      214565 non-null  float64
 8   OXY      24854 non-null   float64
 9   O_3      204482 non-null  float64
 10  PM10     196331 non-null  float64
 11  PM25     55822 non-null   float64
 12  PXY      24854 non-null   float64
 13  SO_2     212671 non-null  float64
 14  TCH      75213 non-null   float64
 15  TOL      59920 non-null   float64
 16  station  215688 non-null  int64  
dtypes: float64(15), int64(1), object(1)
memory usage: 28.0+ MB
```

In [343]:

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

Out[343]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	
0	2009-10-01 01:00:00	104.00	0.27	104.00	104.00	104.00	39.889999	48.150002	104.00	50.6
1	2009-10-01 01:00:00	104.00	0.22	104.00	104.00	104.00	21.230000	24.260000	104.00	55.8
2	2009-10-01 01:00:00	104.00	0.18	104.00	104.00	104.00	31.230000	34.880001	104.00	49.0
3	2009-10-01 01:00:00	0.95	0.33	1.43	2.68	0.25	55.180000	81.360001	1.57	36.6
4	2009-10-01 01:00:00	104.00	0.41	104.00	104.00	0.12	61.349998	76.260002	104.00	38.0
...	...	...	...	...	...	...	...	...	...	
215683	2009-06-01 00:00:00	0.50	0.22	0.39	0.75	0.09	22.000000	24.510000	1.00	82.2
215684	2009-06-01 00:00:00	104.00	0.31	104.00	104.00	104.00	76.110001	101.099998	104.00	41.2
215685	2009-06-01 00:00:00	0.13	104.00	0.86	104.00	0.23	81.050003	99.849998	104.00	24.8
215686	2009-06-01 00:00:00	0.21	104.00	2.96	104.00	0.10	72.419998	82.959999	104.00	104.0
215687	2009-06-01 00:00:00	0.37	0.32	0.99	1.36	0.14	54.290001	64.480003	1.06	56.9

215688 rows × 17 columns

In [344]:

```
b.columns
```

Out[344]:

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

In [345]:

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

Out[345]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	
0	2009-10-01 01:00:00	104.00	0.27	104.00	104.00	104.00	39.889999	48.150002	104.00	50.680000	18
1	2009-10-01 01:00:00	104.00	0.22	104.00	104.00	104.00	21.230000	24.260000	104.00	55.880001	10
2	2009-10-01 01:00:00	104.00	0.18	104.00	104.00	104.00	31.230000	34.880001	104.00	49.060001	25
3	2009-10-01 01:00:00	0.95	0.33	1.43	2.68	0.25	55.180000	81.360001	1.57	36.669998	26
4	2009-10-01 01:00:00	104.00	0.41	104.00	104.00	0.12	61.349998	76.260002	104.00	38.090000	23
5	2009-10-01 01:00:00	104.00	0.29	104.00	104.00	104.00	43.200001	50.080002	104.00	35.840000	21
6	2009-10-01 01:00:00	104.00	0.20	104.00	104.00	104.00	35.430000	38.520000	104.00	33.549999	17
7	2009-10-01 01:00:00	104.00	0.15	104.00	104.00	104.00	27.309999	33.150002	104.00	53.549999	16
8	2009-10-01 01:00:00	104.00	0.21	104.00	104.00	0.39	33.889999	40.799999	104.00	58.549999	16
9	2009-10-01 01:00:00	104.00	0.32	104.00	104.00	104.00	46.349998	60.540001	104.00	45.340000	15

In [346]:

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

Out[346]:

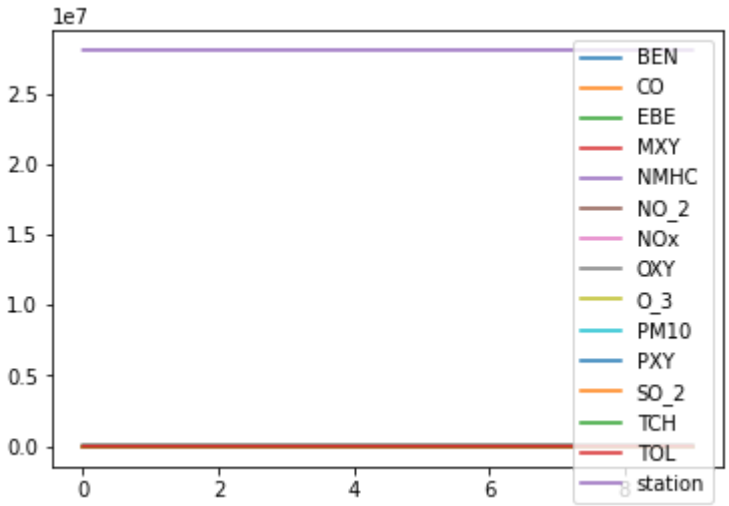
	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	PM10	
0	104.00	0.27	104.00	104.00	104.00	39.889999	48.150002	104.00	50.680000	18.260000	1
1	104.00	0.22	104.00	104.00	104.00	21.230000	24.260000	104.00	55.880001	10.580000	1
2	104.00	0.18	104.00	104.00	104.00	31.230000	34.880001	104.00	49.060001	25.190001	1
3	0.95	0.33	1.43	2.68	0.25	55.180000	81.360001	1.57	36.669998	26.530001	
4	104.00	0.41	104.00	104.00	0.12	61.349998	76.260002	104.00	38.090000	23.760000	1
5	104.00	0.29	104.00	104.00	104.00	43.200001	50.080002	104.00	35.840000	21.870001	1
6	104.00	0.20	104.00	104.00	104.00	35.430000	38.520000	104.00	33.549999	17.350000	1
7	104.00	0.15	104.00	104.00	104.00	27.309999	33.150002	104.00	53.549999	16.520000	1
8	104.00	0.21	104.00	104.00	0.39	33.889999	40.799999	104.00	58.549999	16.650000	1
9	104.00	0.32	104.00	104.00	104.00	46.349998	60.540001	104.00	45.340000	15.160000	1

In [347]:

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

Out[347]:

```
<AxesSubplot:>
```

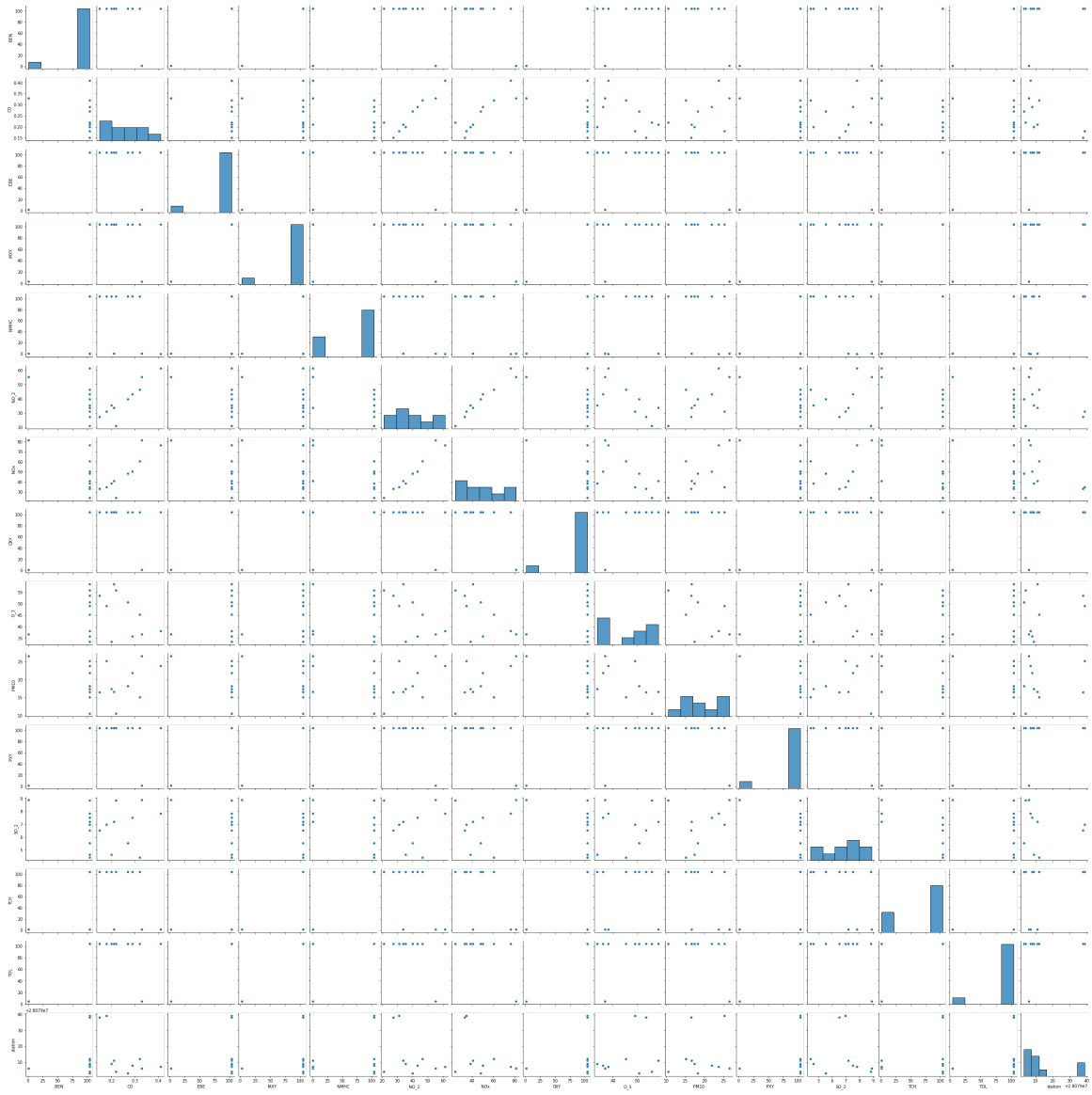


In [348]:

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

Out[348]:

```
<seaborn.axisgrid.PairGrid at 0x1182c767190>
```



In [349]:

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

In [350]:

```
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 [351]:

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

Out[351]:

```
LinearRegression()
```

In [352]:

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

```
0.01035384802160877
```

In [353]:

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

Out[353]:

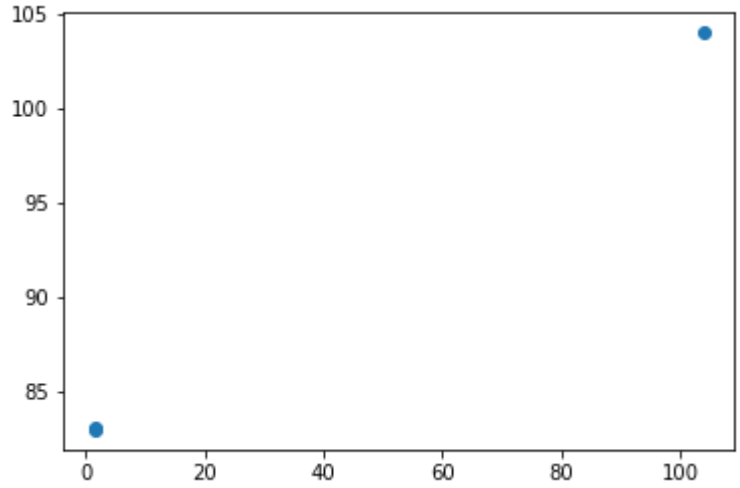
	Co-efficient
BEN	2.008102e-01
CO	-2.374448e-14
EBE	1.998749e-01
MXY	1.974390e-01
NMHC	2.021743e-01
NO_2	-1.447804e-16
NOx	1.958487e-16
OXY	1.996020e-01

In [354]:

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

Out[354]:

```
<matplotlib.collections.PathCollection at 0x118408a8400>
```



In [355]:

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

```
-0.8987927239332554
```

In [356]:

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

In [357]:

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

Out[357]:

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

In [358]:

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

Out[358]:

```
-0.8991111446183997
```

In [359]:

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

Out[359]:

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

In [360]:

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

Out[360]:

```
-1.9928047100494264
```

In [361]:

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

Out[361]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O
0	2009-10-01 01:00:00	104.00	0.27	104.00	104.00	104.00	39.889999	48.150002	104.00	50.6800
1	2009-10-01 01:00:00	104.00	0.22	104.00	104.00	104.00	21.230000	24.260000	104.00	55.8800
2	2009-10-01 01:00:00	104.00	0.18	104.00	104.00	104.00	31.230000	34.880001	104.00	49.0600
3	2009-10-01 01:00:00	0.95	0.33	1.43	2.68	0.25	55.180000	81.360001	1.57	36.6699
4	2009-10-01 01:00:00	104.00	0.41	104.00	104.00	0.12	61.349998	76.260002	104.00	38.0900
...	...	...	...	...	...	...	...	...	...	
6995	2009-10-12 16:00:00	0.42	0.74	0.43	1.08	0.49	11.680000	15.810000	0.67	84.3899
6996	2009-10-12 16:00:00	104.00	0.23	104.00	104.00	104.00	33.090000	54.380001	104.00	57.4800
6997	2009-10-12 16:00:00	0.13	104.00	0.31	104.00	0.19	27.670000	36.860001	104.00	56.2400
6998	2009-10-12 16:00:00	0.20	104.00	1.00	104.00	0.13	16.459999	30.200001	104.00	104.0000
6999	2009-10-12 16:00:00	0.23	0.25	0.63	1.08	0.18	22.760000	32.700001	0.67	64.7399

7000 rows × 17 columns

In [362]:

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

In [363]:

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

In [364]:

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

In [365]:

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

Out[365]:

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

In [366]:

```
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 [367]:

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

In [368]:

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

```
[28079021]
```

In [369]:

```
logr.classes_
```

Out[369]:

```
array([28079003, 28079004, 28079006, 28079007, 28079008, 28079009,
       28079011, 28079012, 28079014, 28079016, 28079017, 28079018,
       28079019, 28079021, 28079022, 28079023, 28079024, 28079025,
       28079026, 28079027, 28079036, 28079038, 28079039, 28079040,
       28079099], dtype=int64)
```

In [370]:

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

Out[370]:

```
6.149368503537759e-164
```

In [371]:

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

Out[371]:

```
3.3876813073808526e-49
```

In [372]:

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

Out[372]:

```
0.5666666666666667
```

In [373]:

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

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_coordinat
e_descent.py:530: ConvergenceWarning: Objective did not converge. You migh
t want to increase the number of iterations. Duality gap: 1.83598498142228
28, tolerance: 0.9026455200838853
  model = cd_fast.enet_coordinate_descent(
```

Out[373]:

```
ElasticNet()
```

In [374]:

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

```
[ 6.77447772e-01 -0.00000000e+00  2.59564600e-01  4.69069600e-02
  1.34238184e-02 -0.00000000e+00 -0.00000000e+00  2.60652611e-06]
```

In [375]:

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

```
0.2703463186058599
```

In [376]:

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

```
-1.9187824253342924
```

In [377]:

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

Out[377]:

```
RandomForestClassifier()
```

In [378]:

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

In [379]:

```
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[379]:

```
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 [380]:

```
grid_search.best_score_
```

Out[380]:

```
0.5724489795918368
```

In [381]:

```
rfc_best=grid_search.best_estimator_
```

In [382]:

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

```
092\nvalue = [193, 191, 215, 193, 198, 204, 218, 187, 198, 188\n189, 185,
206, 172, 189, 211, 213, 179, 184, 171\n190, 214, 215, 203, 194]'),
 Text(1785.6, 2038.5, 'X[10] <= -1.068\ngini = 0.958\nsamples = 2902\nval
ue = [188, 185, 197, 193, 197, 203, 212, 185, 196, 187\n183, 167, 204, 16
4, 189, 209, 199, 174, 182, 0\n186, 205, 211, 195, 191]'),
 Text(892.8, 1585.5, 'X[7] <= -2.815\ngini = 0.666\nsamples = 354\nvalue
= [0, 0, 173, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 199, 0, 0, 0, 0, 0,
0, 0, 191]'),
 Text(446.4, 1132.5, 'X[0] <= -1.664\ngini = 0.409\nsamples = 156\nvalue
= [0, 0, 16, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 191, 0, 0, 0, 0, 0,
0, 0, 51]'),
 Text(223.2, 679.5, 'X[4] <= -1.345\ngini = 0.245\nsamples = 21\nvalue =
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 5, 0, 0, 0, 0, 0, 0, 0,
30]'),
 Text(111.6, 226.5, 'gini = 0.0\nsamples = 10\nvalue = [0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 17]'),
 Text(334.79999999999995, 226.5, 'gini = 0.401\nsamples = 11\nvalue = [0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 1
3]'),
 Text(669.5999999999999, 679.5, 'X[13] <= -1.749\ngini = 0.29\nsamples =
```

In [ ]: