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 [257]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C10_air\madrid_2007.csv")
a
```

Out[257]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	
0	2007-12-01 01:00:00	NaN	2.86	NaN	NaN	NaN	282.200012	1054.000000	NaN	4.030000	1
1	2007-12-01 01:00:00	NaN	1.82	NaN	NaN	NaN	86.419998	354.600006	NaN	3.260000	
2	2007-12-01 01:00:00	NaN	1.47	NaN	NaN	NaN	94.639999	319.000000	NaN	5.310000	
3	2007-12-01 01:00:00	NaN	1.64	NaN	NaN	NaN	127.900002	476.700012	NaN	4.500000	1
4	2007-12-01 01:00:00	4.64	1.86	4.26	7.98	0.57	145.100006	573.900024	3.49	52.689999	1
...	...	...	...	...	...	...	...	...	...	...	
225115	2007-03-01 00:00:00	0.30	0.45	1.00	0.30	0.26	8.690000	11.690000	1.00	42.209999	
225116	2007-03-01 00:00:00	NaN	0.16	NaN	NaN	NaN	46.820000	51.480000	NaN	22.150000	
225117	2007-03-01 00:00:00	0.24	NaN	0.20	NaN	0.09	51.259998	66.809998	NaN	18.540001	
225118	2007-03-01 00:00:00	0.11	NaN	1.00	NaN	0.05	24.240000	36.930000	NaN	NaN	
225119	2007-03-01 00:00:00	0.53	0.40	1.00	1.70	0.12	32.360001	47.860001	1.37	24.150000	

225120 rows × 17 columns

In [258]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 225120 entries, 0 to 225119
Data columns (total 17 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     225120 non-null  object 
 1   BEN      68885 non-null   float64
 2   CO       206748 non-null  float64
 3   EBE      68883 non-null   float64
 4   MXY      26061 non-null   float64
 5   NMHC     86883 non-null   float64
 6   NO_2     223985 non-null  float64
 7   NOx      223972 non-null  float64
 8   OXY      26062 non-null   float64
 9   O_3      211850 non-null  float64
 10  PM10     222588 non-null  float64
 11  PM25     68870 non-null   float64
 12  PXY      26062 non-null   float64
 13  SO_2     224372 non-null  float64
 14  TCH      87026 non-null   float64
 15  TOL      68845 non-null   float64
 16  station  225120 non-null  int64  
dtypes: float64(15), int64(1), object(1)
memory usage: 29.2+ MB
```

In [259]:

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

Out[259]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	
0	2007-12-01 01:00:00	104.00	2.86	104.00	104.00	104.00	282.200012	1054.000000	104.00	4
1	2007-12-01 01:00:00	104.00	1.82	104.00	104.00	104.00	86.419998	354.600006	104.00	3
2	2007-12-01 01:00:00	104.00	1.47	104.00	104.00	104.00	94.639999	319.000000	104.00	5
3	2007-12-01 01:00:00	104.00	1.64	104.00	104.00	104.00	127.900002	476.700012	104.00	4
4	2007-12-01 01:00:00	4.64	1.86	4.26	7.98	0.57	145.100006	573.900024	3.49	52
...	...	...	...	...	...	...	...	...	...	
225115	2007-03-01 00:00:00	0.30	0.45	1.00	0.30	0.26	8.690000	11.690000	1.00	42
225116	2007-03-01 00:00:00	104.00	0.16	104.00	104.00	104.00	46.820000	51.480000	104.00	22
225117	2007-03-01 00:00:00	0.24	104.00	0.20	104.00	0.09	51.259998	66.809998	104.00	18
225118	2007-03-01 00:00:00	0.11	104.00	1.00	104.00	0.05	24.240000	36.930000	104.00	104
225119	2007-03-01 00:00:00	0.53	0.40	1.00	1.70	0.12	32.360001	47.860001	1.37	24

225120 rows × 17 columns

In [260]:

```
b.columns
```

Out[260]:

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

In [261]:

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

Out[261]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3
0	2007-12-01 01:00:00	104.00	2.86	104.00	104.00	104.00	282.200012	1054.000000	104.00	4.030000
1	2007-12-01 01:00:00	104.00	1.82	104.00	104.00	104.00	86.419998	354.600006	104.00	3.260000
2	2007-12-01 01:00:00	104.00	1.47	104.00	104.00	104.00	94.639999	319.000000	104.00	5.310000
3	2007-12-01 01:00:00	104.00	1.64	104.00	104.00	104.00	127.900002	476.700012	104.00	4.500000
4	2007-12-01 01:00:00	4.64	1.86	4.26	7.98	0.57	145.100006	573.900024	3.49	52.689999
5	2007-12-01 01:00:00	104.00	1.35	104.00	104.00	0.56	115.300003	319.600006	104.00	9.880000
6	2007-12-01 01:00:00	5.54	1.87	4.65	104.00	0.75	165.100006	520.000000	104.00	4.780000
7	2007-12-01 01:00:00	104.00	1.57	104.00	104.00	104.00	97.830002	369.000000	104.00	4.870000
8	2007-12-01 01:00:00	104.00	0.70	104.00	104.00	104.00	107.699997	188.500000	104.00	4.560000
9	2007-12-01 01:00:00	104.00	1.48	104.00	104.00	0.69	152.500000	485.200012	104.00	8.230000

In [262]:

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

Out[262]:

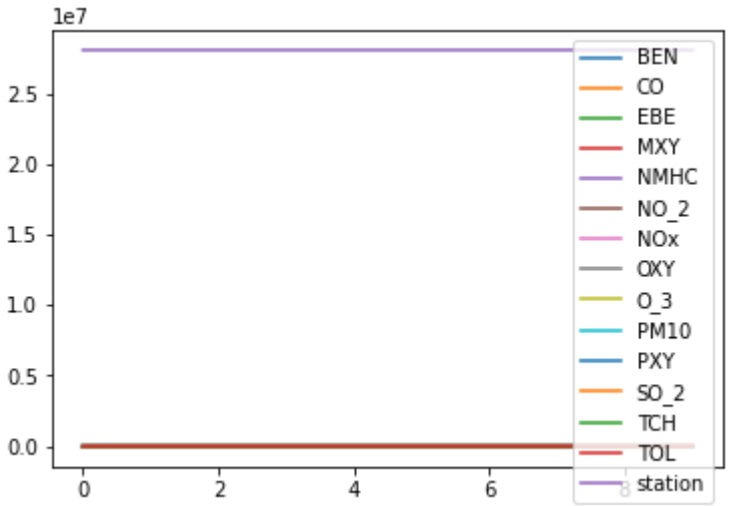
	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	PM
0	104.00	2.86	104.00	104.00	104.00	282.200012	1054.000000	104.00	4.030000	156.1999
1	104.00	1.82	104.00	104.00	104.00	86.419998	354.600006	104.00	3.260000	80.8099
2	104.00	1.47	104.00	104.00	104.00	94.639999	319.000000	104.00	5.310000	53.0999
3	104.00	1.64	104.00	104.00	104.00	127.900002	476.700012	104.00	4.500000	105.3000
4	4.64	1.86	4.26	7.98	0.57	145.100006	573.900024	3.49	52.689999	106.5000
5	104.00	1.35	104.00	104.00	0.56	115.300003	319.600006	104.00	9.880000	57.5000
6	5.54	1.87	4.65	104.00	0.75	165.100006	520.000000	104.00	4.780000	75.9899
7	104.00	1.57	104.00	104.00	104.00	97.830002	369.000000	104.00	4.870000	59.5900
8	104.00	0.70	104.00	104.00	104.00	107.699997	188.500000	104.00	4.560000	43.3400
9	104.00	1.48	104.00	104.00	0.69	152.500000	485.200012	104.00	8.230000	80.8300

In [263]:

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

Out[263]:

```
<AxesSubplot:>
```

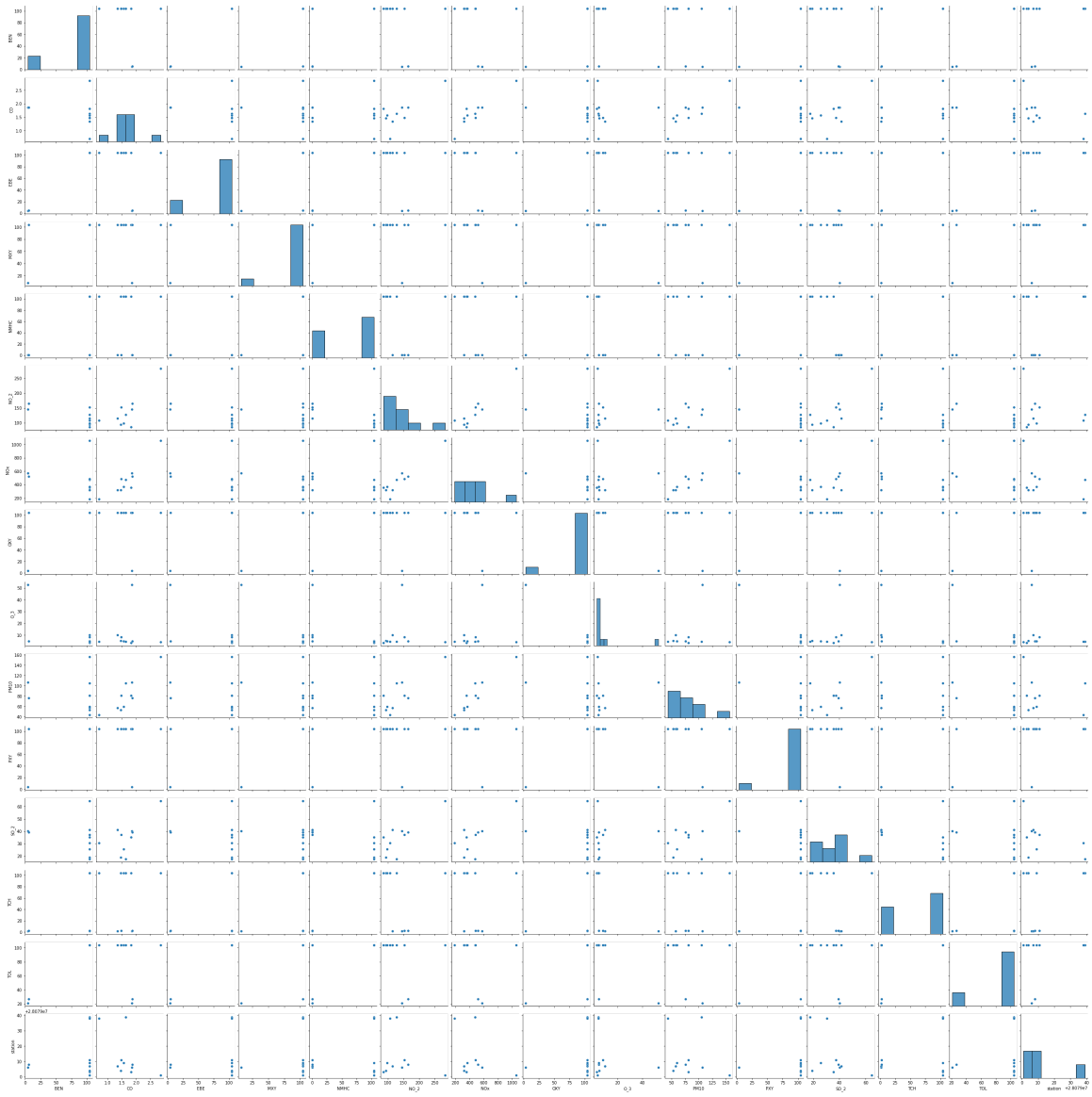


In [264]:

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

Out[264]:

```
<seaborn.axisgrid.PairGrid at 0x11797bd79a0>
```



In [265]:

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

In [266]:

```
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 [267]:

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

Out[267]:

```
LinearRegression()
```

In [268]:

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

```
-1.207616367852438
```

In [269]:

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

Out[269]:

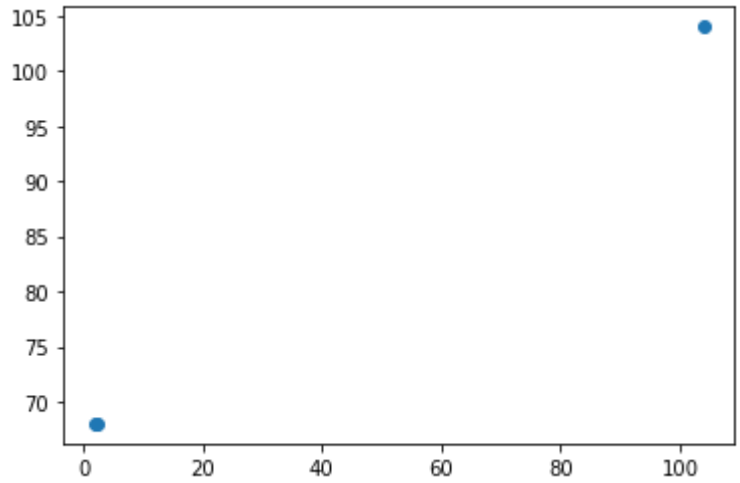
	Co-efficient
BEN	3.317732e-01
CO	2.168144e-13
EBE	3.347797e-01
MXY	-1.400320e-03
NMHC	3.479249e-01
NO_2	3.482874e-16
NOx	-4.170717e-16
OXY	-1.465801e-03

In [270]:

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

Out[270]:

```
<matplotlib.collections.PathCollection at 0x117976d7640>
```



In [271]:

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

```
-0.25792078095703275
```

In [272]:

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

In [273]:

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

Out[273]:

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

In [274]:

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

Out[274]:

```
-0.2549180477755906
```

In [275]:

```
la=Lasso(alpha=10)
la.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: 2.65920401852099
44, tolerance: 1.4849767777783587
  model = cd_fast.enet_coordinate_descent(
```

Out[275]:

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

In [276]:

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

Out[276]:

```
-1.0011645139932854
```

In [277]:

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

Out[277]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	
0	2007-12-01 01:00:00	104.00	2.86	104.00	104.00	104.00	282.200012	1054.000000	104.00	4.030
1	2007-12-01 01:00:00	104.00	1.82	104.00	104.00	104.00	86.419998	354.600006	104.00	3.260
2	2007-12-01 01:00:00	104.00	1.47	104.00	104.00	104.00	94.639999	319.000000	104.00	5.310
3	2007-12-01 01:00:00	104.00	1.64	104.00	104.00	104.00	127.900002	476.700012	104.00	4.500
4	2007-12-01 01:00:00	4.64	1.86	4.26	7.98	0.57	145.100006	573.900024	3.49	52.689
...	...	...	...	...	...	...	...	...	...	
6995	2007-12-12 06:00:00	104.00	0.63	104.00	104.00	104.00	43.520000	99.480003	104.00	3.070
6996	2007-12-12 06:00:00	104.00	0.52	104.00	104.00	104.00	37.279999	73.059998	104.00	1.000
6997	2007-12-12 06:00:00	104.00	0.29	104.00	104.00	104.00	55.619999	118.900002	104.00	104.000
6998	2007-12-12 06:00:00	0.62	0.34	0.49	0.68	0.21	59.540001	101.300003	0.33	17.809
6999	2007-12-12 06:00:00	104.00	0.26	104.00	104.00	0.29	39.490002	43.330002	104.00	20.870

7000 rows × 17 columns

In [278]:

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

In [279]:

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

In [280]:

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

In [281]:

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

Out[281]:

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

In [282]:

```
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 [283]:

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

In [284]:

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

```
[28079038]
```

In [285]:

```
logr.classes_
```

Out[285]:

```
array([28079001, 28079003, 28079004, 28079006, 28079007, 28079008,
       28079009, 28079011, 28079012, 28079014, 28079015, 28079016,
       28079018, 28079019, 28079021, 28079022, 28079023, 28079024,
       28079025, 28079026, 28079027, 28079036, 28079038, 28079039,
       28079040, 28079099], dtype=int64)
```

In [286]:

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

Out[286]:

```
1.0701549751974171e-63
```

In [287]:

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

Out[287]:

```
1.1266207040001885e-104
```

In [288]:

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

Out[288]:

```
0.4895238095238095
```

In [289]:

```
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: 2.43332004441908
06, tolerance: 1.4849767777783587
  model = cd_fast.enet_coordinate_descent(
```

Out[289]:

```
ElasticNet()
```

In [290]:

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

```
[ 0.79518134 -0.          0.20228184 -0.          0.03276292 -0.
  0.         -0.00483021]
```

In [291]:

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

```
-2.6481699737983604
```

In [292]:

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

```
-1.8037473558047155
```

In [293]:

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

Out[293]:

```
RandomForestClassifier()
```

In [294]:

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

In [295]:

```
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[295]:

```
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 [296]:

```
grid_search.best_score_
```

Out[296]:

```
0.5495918367346939
```

In [297]:

```
rfc_best=grid_search.best_estimator_
```

In [298]:

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

```
= 3089\nvalue = [184, 203, 183, 172, 213, 206, 160, 188, 165, 163\n176, 1
68, 172, 229, 214, 221, 202, 176, 181, 177\n178, 179, 199, 212, 185, 19
4]'),
 Text(1310.086956521739, 2038.5, 'X[7] <= -2.752\ngini = 0.666\nsamples =
340\nvalue = [0, 0, 0, 172, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 176,
0, 0, 0, 0, 0, 0, 0, 194]'),
 Text(776.3478260869565, 1585.5, 'X[13] <= -1.683\ngini = 0.656\nsamples
= 260\nvalue = [0, 0, 0, 110, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 171,
0, 0, 0, 0, 0, 0, 0, 138]'),
 Text(388.17391304347825, 1132.5, 'X[11] <= -0.689\ngini = 0.503\nsamples
= 60\nvalue = [0, 0, 0, 20, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 62, 0,
0, 0, 0, 0, 0, 0, 12]'),
 Text(194.08695652173913, 679.5, 'X[3] <= -2.863\ngini = 0.333\nsamples =
49\nvalue = [0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 62, 0, 0,
0, 0, 0, 0, 0, 8]'),
 Text(97.04347826086956, 226.5, 'gini = 0.531\nsamples = 25\nvalue = [0,
0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 26, 0, 0, 0, 0, 0, 0, 0,
8]'),
 Text(291.1304347826087, 226.5, 'gini = 0.0\nsamples = 24\nvalue = [0, 0,
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0\n0. 0. 0. 36. 0. 0. 0. 0. 0. 0. 0.
```

In [ ]: