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 [299]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C10_air\madrid_2008.csv")
a
```

Out[299]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	
0	2008-06-01 01:00:00	NaN	0.47	NaN	NaN	NaN	83.089996	120.699997	NaN	16.990000	16
1	2008-06-01 01:00:00	NaN	0.59	NaN	NaN	NaN	94.820000	130.399994	NaN	17.469999	19
2	2008-06-01 01:00:00	NaN	0.55	NaN	NaN	NaN	75.919998	104.599998	NaN	13.470000	20
3	2008-06-01 01:00:00	NaN	0.36	NaN	NaN	NaN	61.029999	66.559998	NaN	23.110001	10
4	2008-06-01 01:00:00	1.68	0.80	1.70	3.01	0.30	105.199997	214.899994	1.61	12.120000	37
...	...	...	...	...	...	...	...	...	...	...	
226387	2008-11-01 00:00:00	0.48	0.30	0.57	1.00	0.31	13.050000	14.160000	0.91	57.400002	5
226388	2008-11-01 00:00:00	NaN	0.30	NaN	NaN	NaN	41.880001	48.500000	NaN	35.830002	15
226389	2008-11-01 00:00:00	0.25	NaN	0.56	NaN	0.11	83.610001	102.199997	NaN	14.130000	17
226390	2008-11-01 00:00:00	0.54	NaN	2.70	NaN	0.18	70.639999	81.860001	NaN	NaN	11
226391	2008-11-01 00:00:00	0.75	0.36	1.20	2.75	0.16	58.240002	74.239998	1.64	31.910000	12

226392 rows × 17 columns

In [300]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 226392 entries, 0 to 226391
Data columns (total 17 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     226392 non-null  object 
 1   BEN      67047 non-null   float64
 2   CO       208109 non-null  float64
 3   EBE      67044 non-null   float64
 4   MXY      25867 non-null   float64
 5   NMHC     85079 non-null   float64
 6   NO_2     225315 non-null  float64
 7   NOx      225311 non-null  float64
 8   OXY      25878 non-null   float64
 9   O_3      215716 non-null  float64
 10  PM10     220179 non-null  float64
 11  PM25     67833 non-null   float64
 12  PXY      25877 non-null   float64
 13  SO_2     225405 non-null  float64
 14  TCH      85107 non-null   float64
 15  TOL      66940 non-null   float64
 16  station  226392 non-null  int64  
dtypes: float64(15), int64(1), object(1)
memory usage: 29.4+ MB
```

In [301]:

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

Out[301]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	
0	2008-06-01 01:00:00	104.00	0.47	104.00	104.00	104.00	83.089996	120.699997	104.00	16.
1	2008-06-01 01:00:00	104.00	0.59	104.00	104.00	104.00	94.820000	130.399994	104.00	17.
2	2008-06-01 01:00:00	104.00	0.55	104.00	104.00	104.00	75.919998	104.599998	104.00	13.
3	2008-06-01 01:00:00	104.00	0.36	104.00	104.00	104.00	61.029999	66.559998	104.00	23.
4	2008-06-01 01:00:00	1.68	0.80	1.70	3.01	0.30	105.199997	214.899994	1.61	12.
...	...	...	...	...	...	...	...	...	...	
226387	2008-11-01 00:00:00	0.48	0.30	0.57	1.00	0.31	13.050000	14.160000	0.91	57.
226388	2008-11-01 00:00:00	104.00	0.30	104.00	104.00	104.00	41.880001	48.500000	104.00	35.
226389	2008-11-01 00:00:00	0.25	104.00	0.56	104.00	0.11	83.610001	102.199997	104.00	14.
226390	2008-11-01 00:00:00	0.54	104.00	2.70	104.00	0.18	70.639999	81.860001	104.00	104.
226391	2008-11-01 00:00:00	0.75	0.36	1.20	2.75	0.16	58.240002	74.239998	1.64	31.

226392 rows × 17 columns

In [302]:

```
b.columns
```

Out[302]:

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

In [303]:

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

Out[303]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3
0	2008-06-01 01:00:00	104.00	0.47	104.00	104.00	104.00	83.089996	120.699997	104.00	16.990000
1	2008-06-01 01:00:00	104.00	0.59	104.00	104.00	104.00	94.820000	130.399994	104.00	17.469999
2	2008-06-01 01:00:00	104.00	0.55	104.00	104.00	104.00	75.919998	104.599998	104.00	13.470000
3	2008-06-01 01:00:00	104.00	0.36	104.00	104.00	104.00	61.029999	66.559998	104.00	23.110001
4	2008-06-01 01:00:00	1.68	0.80	1.70	3.01	0.30	105.199997	214.899994	1.61	12.120000
5	2008-06-01 01:00:00	104.00	0.47	104.00	104.00	0.22	67.820000	101.099998	104.00	20.610001
6	2008-06-01 01:00:00	0.17	0.40	0.44	104.00	0.15	72.639999	91.220001	104.00	17.040001
7	2008-06-01 01:00:00	104.00	0.51	104.00	104.00	104.00	80.440002	141.500000	104.00	10.310000
8	2008-06-01 01:00:00	104.00	0.36	104.00	104.00	104.00	68.150002	85.639999	104.00	23.580000
9	2008-06-01 01:00:00	104.00	0.18	104.00	104.00	0.16	58.330002	64.769997	104.00	35.060001

In [304]:

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

Out[304]:

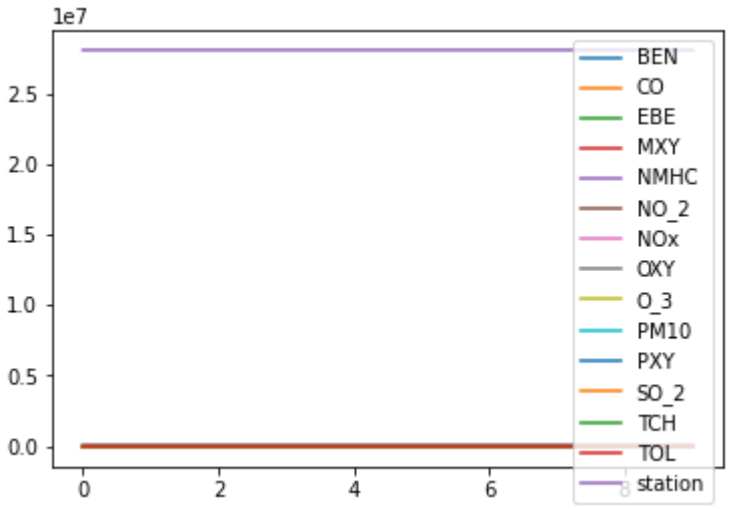
	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	PM10
0	104.00	0.47	104.00	104.00	104.00	83.089996	120.699997	104.00	16.990000	16.889999
1	104.00	0.59	104.00	104.00	104.00	94.820000	130.399994	104.00	17.469999	19.040001
2	104.00	0.55	104.00	104.00	104.00	75.919998	104.599998	104.00	13.470000	20.270000
3	104.00	0.36	104.00	104.00	104.00	61.029999	66.559998	104.00	23.110001	10.850000
4	1.68	0.80	1.70	3.01	0.30	105.199997	214.899994	1.61	12.120000	37.160000
5	104.00	0.47	104.00	104.00	0.22	67.820000	101.099998	104.00	20.610001	23.389999
6	0.17	0.40	0.44	104.00	0.15	72.639999	91.220001	104.00	17.040001	19.940001
7	104.00	0.51	104.00	104.00	104.00	80.440002	141.500000	104.00	10.310000	37.259998
8	104.00	0.36	104.00	104.00	104.00	68.150002	85.639999	104.00	23.580000	15.060000
9	104.00	0.18	104.00	104.00	0.16	58.330002	64.769997	104.00	35.060001	7.400000

In [305]:

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

Out[305]:

```
<AxesSubplot:>
```

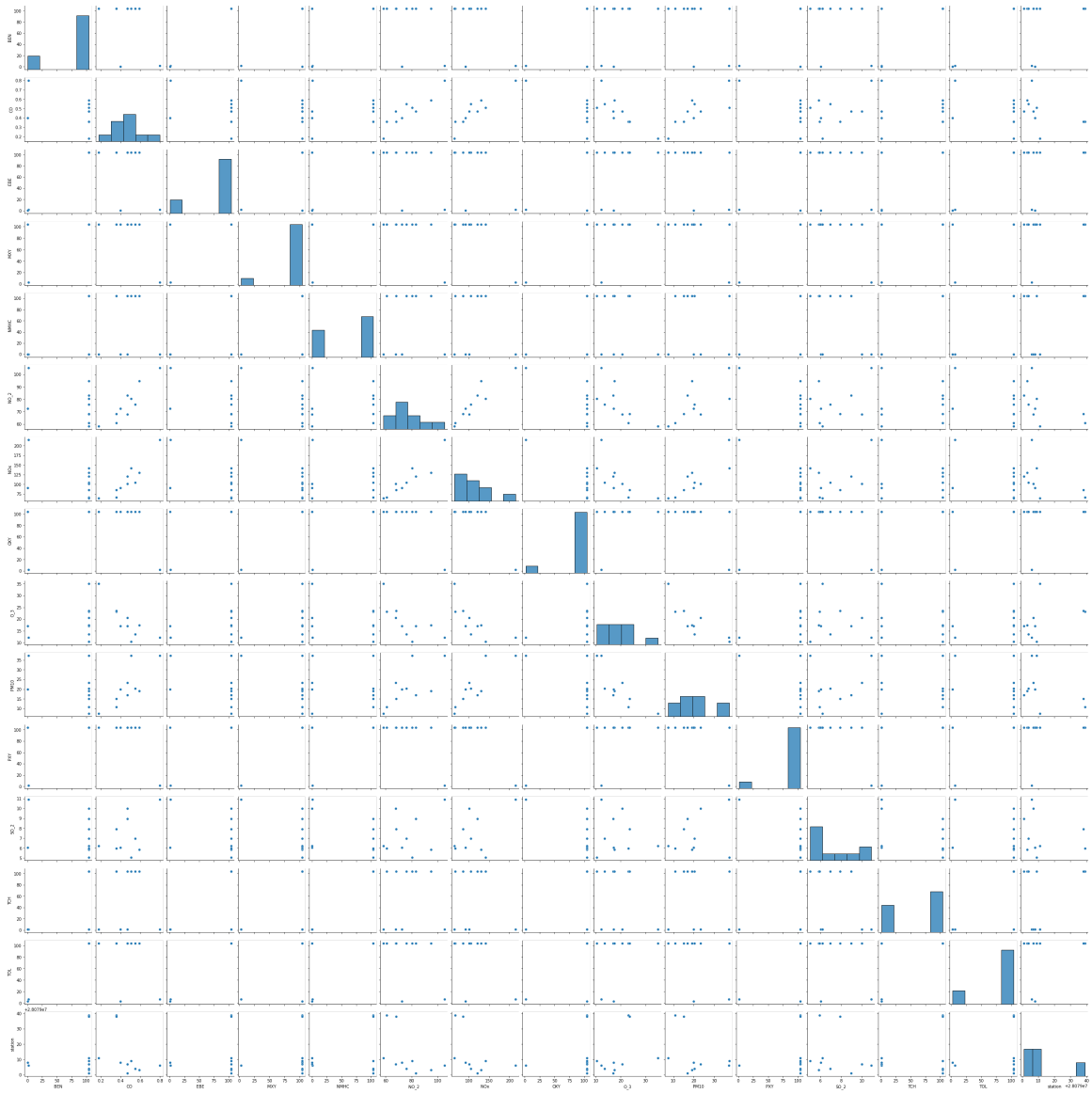


In [306]:

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

Out[306]:

```
<seaborn.axisgrid.PairGrid at 0x1181d8910d0>
```



In [307]:

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

In [308]:

```
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 [309]:

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

Out[309]:

```
LinearRegression()
```

In [310]:

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

```
1.1918373514437803
```

In [311]:

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

Out[311]:

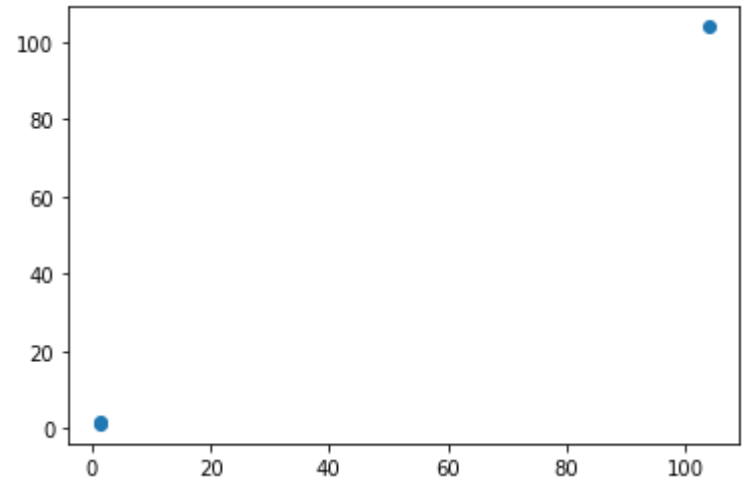
	Co-efficient
BEN	-3.826333e-04
CO	5.114989e-14
EBE	-3.816383e-04
MXY	0.000000e+00
NMHC	9.893043e-01
NO_2	4.091510e-16
NOx	3.302534e-16
OXY	0.000000e+00

In [312]:

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

Out[312]:

```
<matplotlib.collections.PathCollection at 0x1182c724070>
```



In [313]:

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

```
0.999999217096281
```

In [314]:

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

In [315]:

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

Out[315]:

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

In [316]:

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

Out[316]:

```
0.9999943889213244
```

In [317]:

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

Out[317]:

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

In [318]:

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

Out[318]:

```
0.9999608606266869
```

In [319]:

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

Out[319]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3
0	2008-06-01 01:00:00	104.00	0.47	104.0	104.00	104.00	83.089996	120.699997	104.00	16.990000
1	2008-06-01 01:00:00	104.00	0.59	104.0	104.00	104.00	94.820000	130.399994	104.00	17.469999
2	2008-06-01 01:00:00	104.00	0.55	104.0	104.00	104.00	75.919998	104.599998	104.00	13.470000
3	2008-06-01 01:00:00	104.00	0.36	104.0	104.00	104.00	61.029999	66.559998	104.00	23.110001
4	2008-06-01 01:00:00	1.68	0.80	1.7	3.01	0.30	105.199997	214.899994	1.61	12.120000
...	...	...	...	...	...	...	...	...	...	...
6995	2008-06-12 06:00:00	104.00	0.32	104.0	104.00	104.00	65.290001	86.440002	104.00	18.590000
6996	2008-06-12 06:00:00	104.00	0.12	104.0	104.00	104.00	27.959999	31.129999	104.00	59.799999
6997	2008-06-12 06:00:00	104.00	0.14	104.0	104.00	0.13	15.480000	18.360001	104.00	73.620003
6998	2008-06-12 06:00:00	104.00	0.29	104.0	104.00	104.00	15.630000	18.490000	104.00	78.970001
6999	2008-06-12 06:00:00	104.00	0.16	104.0	104.00	104.00	11.860000	14.410000	104.00	88.370003

7000 rows × 17 columns

In [320]:

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

In [321]:

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

In [322]:

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

In [323]:

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

Out[323]:

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

In [324]:

```
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 [325]:

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

In [326]:

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

```
[28079039]
```

In [327]:

```
logr.classes_
```

Out[327]:

```
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 [328]:

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

Out[328]:

```
8.385635135144448e-68
```

In [329]:

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

Out[329]:

```
1.6358718737218 69e-98
```

In [330]:

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

Out[330]:

```
0.58
```

In [331]:

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

Out[331]:

```
ElasticNet()
```

In [332]:

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

```
[-0.         -0.         -0.          0.          0.98847039  0.
  0.          0.        ]
```

In [333]:

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

```
1.1856658265911193
```

In [334]:

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

```
0.9999970988081355
```

In [335]:

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

Out[335]:

```
RandomForestClassifier()
```

In [336]:

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

In [337]:

```
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[337]:

```
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 [338]:

```
grid_search.best_score_
```

Out[338]:

```
0.656530612244898
```

In [339]:

```
rfc_best=grid_search.best_estimator_
```

In [340]:

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

```
= 3127\nvalue = [199, 200, 162, 190, 173, 173, 189, 211, 187, 183\n195, 1
85, 196, 186, 207, 192, 171, 171, 188, 189\n181, 230, 204, 189, 189, 16
0]'),
 Text(648.8372093023256, 2038.5, 'X[6] <= -0.72\ngini = 0.665\nsamples =
336\nvalue = [0, 0, 0, 190, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 171,
0, 0, 0, 0, 0, 0, 0, 160]'),
 Text(311.4418604651163, 1585.5, 'X[1] <= -0.304\ngini = 0.355\nsamples =
121\nvalue = [0, 0, 0, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 156, 0,
0, 0, 0, 0, 0, 0, 33]'),
 Text(207.62790697674419, 1132.5, 'X[6] <= -0.923\ngini = 0.559\nsamples
= 30\nvalue = [0, 0, 0, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 12, 0,
0, 0, 0, 0, 0, 0, 33]'),
 Text(103.81395348837209, 679.5, 'gini = 0.54\nsamples = 11\nvalue = [0,
0, 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 12, 0, 0, 0, 0, 0, 0, 0,
2]'),
 Text(311.4418604651163, 679.5, 'gini = 0.202\nsamples = 19\nvalue = [0,
0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3
1]'),
 Text(415.25581395348837, 1132.5, 'gini = 0.0\nsamples = 91\nvalue = [0,
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0\n0. 0. 0. 144. 0. 0. 0. 0. 0. 0. 0.
```

In [ ]: