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 [622]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C10_air\madrid_2015.csv")
a
```

Out[622]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL
0	2015-10-01 01:00:00	NaN	0.8	NaN	NaN	90.0	82.0	NaN	NaN	NaN	10.0	NaN	NaN
1	2015-10-01 01:00:00	2.0	0.8	1.6	0.33	40.0	95.0	4.0	37.0	24.0	12.0	1.83	8.3
2	2015-10-01 01:00:00	3.1	NaN	1.8	NaN	29.0	97.0	NaN	NaN	NaN	NaN	NaN	7.1
3	2015-10-01 01:00:00	NaN	0.6	NaN	NaN	30.0	103.0	2.0	NaN	NaN	NaN	NaN	NaN
4	2015-10-01 01:00:00	NaN	NaN	NaN	NaN	95.0	96.0	2.0	NaN	NaN	9.0	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
210091	2015-08-01 00:00:00	NaN	0.2	NaN	NaN	11.0	33.0	53.0	NaN	NaN	NaN	NaN	NaN
210092	2015-08-01 00:00:00	NaN	0.2	NaN	NaN	1.0	5.0	NaN	26.0	NaN	10.0	NaN	NaN
210093	2015-08-01 00:00:00	NaN	NaN	NaN	NaN	1.0	7.0	74.0	NaN	NaN	NaN	NaN	NaN
210094	2015-08-01 00:00:00	NaN	NaN	NaN	NaN	3.0	7.0	65.0	NaN	NaN	NaN	NaN	NaN
210095	2015-08-01 00:00:00	NaN	NaN	NaN	NaN	1.0	9.0	54.0	29.0	NaN	NaN	NaN	NaN

210096 rows × 14 columns

In [623]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210096 entries, 0 to 210095
Data columns (total 14 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     210096 non-null  object 
 1   BEN      51039 non-null   float64
 2   CO       86827 non-null   float64
 3   EBE      50962 non-null   float64
 4   NMHC     25756 non-null   float64
 5   NO       208805 non-null  float64
 6   NO_2     208805 non-null  float64
 7   O_3      121574 non-null  float64
 8   PM10     102745 non-null  float64
 9   PM25     48798 non-null   float64
 10  SO_2     86898 non-null   float64
 11  TCH      25756 non-null   float64
 12  TOL      50626 non-null   float64
 13  station  210096 non-null  int64  
dtypes: float64(12), int64(1), object(1)
memory usage: 22.4+ MB
```

In [624]:

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

Out[624]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH
0	2015-10-01 01:00:00	104.0	0.8	104.0	104.00	90.0	82.0	104.0	104.0	104.0	10.0	104.00
1	2015-10-01 01:00:00	2.0	0.8	1.6	0.33	40.0	95.0	4.0	37.0	24.0	12.0	1.83
2	2015-10-01 01:00:00	3.1	104.0	1.8	104.00	29.0	97.0	104.0	104.0	104.0	104.0	104.00
3	2015-10-01 01:00:00	104.0	0.6	104.0	104.00	30.0	103.0	2.0	104.0	104.0	104.0	104.00
4	2015-10-01 01:00:00	104.0	104.0	104.0	104.00	95.0	96.0	2.0	104.0	104.0	9.0	104.00
...	...	...	...	...	...	...	...	...	...	...	...	...
210091	2015-08-01 00:00:00	104.0	0.2	104.0	104.00	11.0	33.0	53.0	104.0	104.0	104.0	104.00
210092	2015-08-01 00:00:00	104.0	0.2	104.0	104.00	1.0	5.0	104.0	26.0	104.0	10.0	104.00
210093	2015-08-01 00:00:00	104.0	104.0	104.0	104.00	1.0	7.0	74.0	104.0	104.0	104.0	104.00
210094	2015-08-01 00:00:00	104.0	104.0	104.0	104.00	3.0	7.0	65.0	104.0	104.0	104.0	104.00
210095	2015-08-01 00:00:00	104.0	104.0	104.0	104.00	1.0	9.0	54.0	29.0	104.0	104.0	104.00

210096 rows × 14 columns

In [625]:

```
b.columns
```

Out[625]:

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

In [626]:

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

Out[626]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL
0	2015-10-01 01:00:00	104.0	0.8	104.0	104.00	90.0	82.0	104.0	104.0	104.0	10.0	104.00	104.
1	2015-10-01 01:00:00	2.0	0.8	1.6	0.33	40.0	95.0	4.0	37.0	24.0	12.0	1.83	8.
2	2015-10-01 01:00:00	3.1	104.0	1.8	104.00	29.0	97.0	104.0	104.0	104.0	104.0	104.00	7.
3	2015-10-01 01:00:00	104.0	0.6	104.0	104.00	30.0	103.0	2.0	104.0	104.0	104.0	104.00	104.
4	2015-10-01 01:00:00	104.0	104.0	104.0	104.00	95.0	96.0	2.0	104.0	104.0	9.0	104.00	104.
5	2015-10-01 01:00:00	0.7	0.4	0.3	104.00	35.0	104.0	1.0	26.0	104.0	3.0	104.00	3.
6	2015-10-01 01:00:00	0.5	0.3	0.3	0.12	6.0	83.0	1.0	19.0	12.0	3.0	1.29	4.
7	2015-10-01 01:00:00	104.0	104.0	104.0	104.00	54.0	94.0	1.0	104.0	104.0	104.0	104.00	104.
8	2015-10-01 01:00:00	104.0	0.5	104.0	104.00	38.0	114.0	16.0	104.0	104.0	104.0	104.00	104.
9	2015-10-01 01:00:00	104.0	0.7	104.0	104.00	64.0	97.0	104.0	34.0	104.0	6.0	104.00	104.

In [627]:

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

Out[627]:

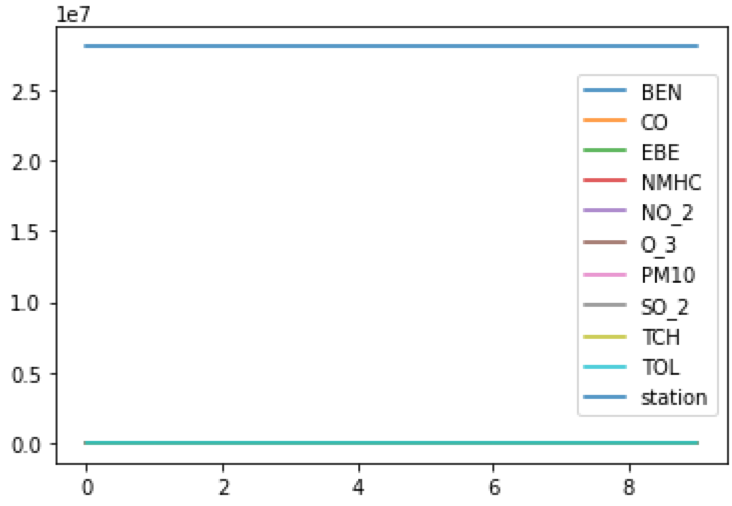
	BEN	CO	EBE	NMHC	NO_2	O_3	PM10	SO_2	TCH	TOL	station
0	104.0	0.8	104.0	104.00	82.0	104.0	104.0	10.0	104.00	104.0	28079004
1	2.0	0.8	1.6	0.33	95.0	4.0	37.0	12.0	1.83	8.3	28079008
2	3.1	104.0	1.8	104.00	97.0	104.0	104.0	104.0	104.00	7.1	28079011
3	104.0	0.6	104.0	104.00	103.0	2.0	104.0	104.0	104.00	104.0	28079016
4	104.0	104.0	104.0	104.00	96.0	2.0	104.0	9.0	104.00	104.0	28079017
5	0.7	0.4	0.3	104.00	104.0	1.0	26.0	3.0	104.00	3.3	28079018
6	0.5	0.3	0.3	0.12	83.0	1.0	19.0	3.0	1.29	4.8	28079024
7	104.0	104.0	104.0	104.00	94.0	1.0	104.0	104.0	104.00	104.0	28079027
8	104.0	0.5	104.0	104.00	114.0	16.0	104.0	104.0	104.00	104.0	28079035
9	104.0	0.7	104.0	104.00	97.0	104.0	34.0	6.0	104.00	104.0	28079036

In [628]:

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

Out[628]:

```
<AxesSubplot:>
```

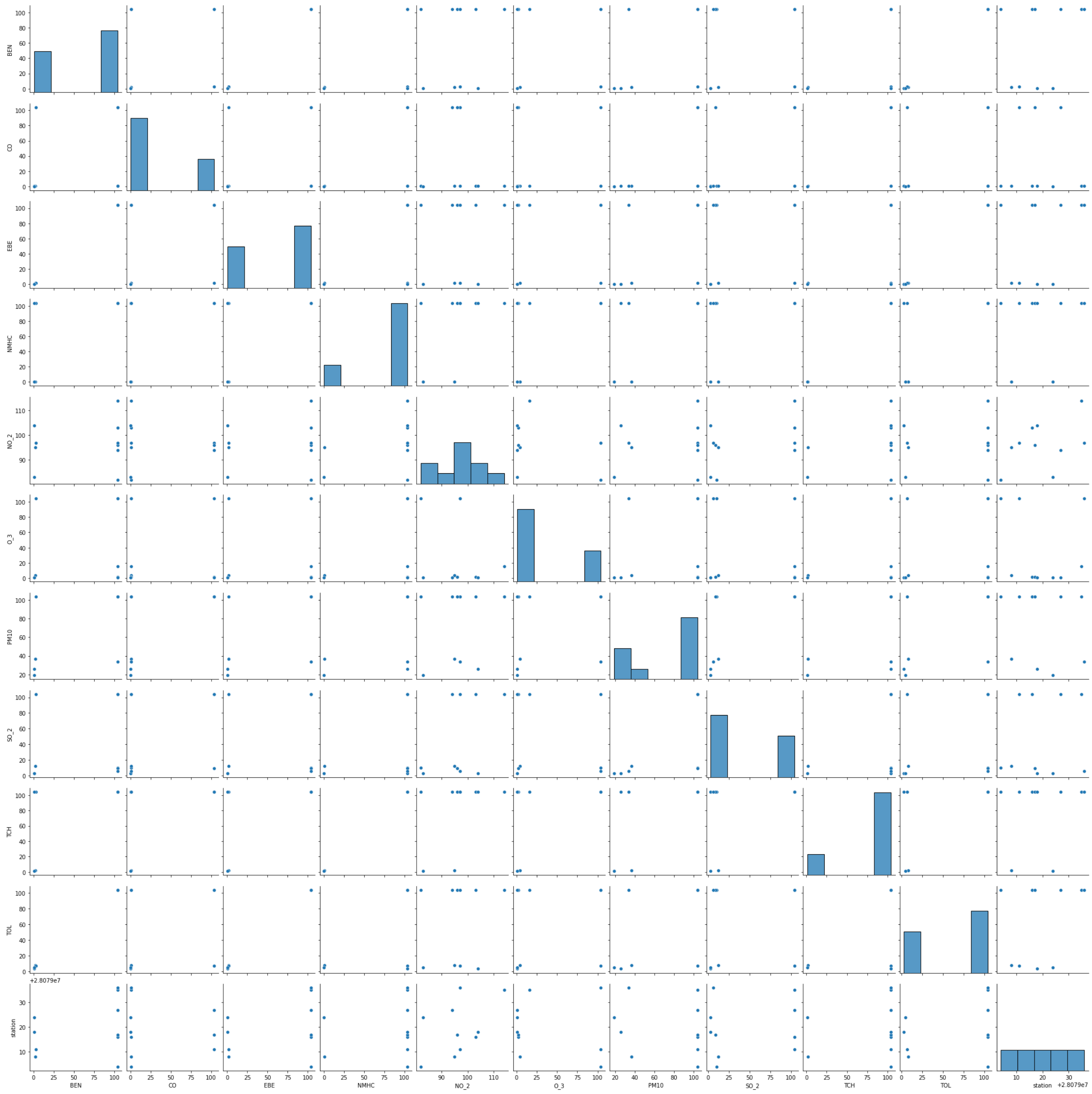


In [629]:

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

Out[629]:

```
<seaborn.axisgrid.PairGrid at 0x118836ffb80>
```



In [630]:

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

In [631]:

```
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 [632]:

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

Out[632]:

```
LinearRegression()
```

In [633]:

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

```
0.8449704632442376
```

In [634]:

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

Out[634]:

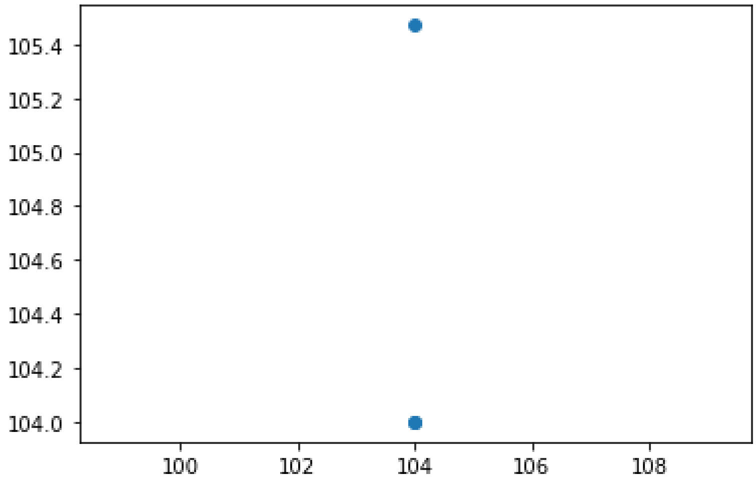
	Co-efficient
BEN	1.624383e+00
CO	2.893869e-16
EBE	-1.618118e+00
NMHC	9.856096e-01
NO_2	3.207197e-16

In [635]:

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

Out[635]:

```
<matplotlib.collections.PathCollection at 0x1188f77f340>
```



In [636]:

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

```
0.0
```

In [637]:

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

In [638]:

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

Out[638]:

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

In [639]:

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

Out[639]:

```
0.0
```

In [640]:

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

Out[640]:

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

In [641]:

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

Out[641]:

```
0.0
```

In [642]:

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

Out[642]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	
0	2015-10-01 01:00:00	104.0	0.8	104.0	104.00	90.0	82.0	104.0	104.0	104.0	10.0	104.00	1
1	2015-10-01 01:00:00	2.0	0.8	1.6	0.33	40.0	95.0	4.0	37.0	24.0	12.0	1.83	
2	2015-10-01 01:00:00	3.1	104.0	1.8	104.00	29.0	97.0	104.0	104.0	104.0	104.0	104.00	
3	2015-10-01 01:00:00	104.0	0.6	104.0	104.00	30.0	103.0	2.0	104.0	104.0	104.0	104.00	1
4	2015-10-01 01:00:00	104.0	104.0	104.0	104.00	95.0	96.0	2.0	104.0	104.0	9.0	104.00	1
...	...	...	...	...	...	...	...	...	...	...	...	...	
6995	2015-10-13 04:00:00	104.0	0.3	104.0	104.00	32.0	44.0	7.0	104.0	104.0	104.0	104.00	1
6996	2015-10-13 04:00:00	104.0	104.0	104.0	104.00	9.0	44.0	104.0	11.0	104.0	4.0	104.00	1
6997	2015-10-13 04:00:00	104.0	104.0	104.0	104.00	30.0	27.0	104.0	28.0	23.0	104.0	104.00	1
6998	2015-10-13 04:00:00	104.0	104.0	104.0	104.00	32.0	46.0	104.0	14.0	11.0	104.0	104.00	1
6999	2015-10-13 04:00:00	104.0	104.0	104.0	104.00	44.0	37.0	2.0	104.0	104.0	104.0	104.00	1

7000 rows × 14 columns

In [643]:

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

In [644]:

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

In [645]:

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

In [646]:

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

Out[646]:

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

In [647]:

```
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 [648]:

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

In [649]:

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

```
[28079059]
```

In [650]:

```
logr.classes_
```

Out[650]:

```
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 [651]:

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

Out[651]:

```
0.0
```

In [652]:

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

Out[652]:

```
0.0
```

In [653]:

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

Out[653]:

```
0.9395238095238095
```

In [654]:

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

Out[654]:

```
ElasticNet()
```

In [655]:

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

```
[0.00000000e+00 0.00000000e+00 3.55198081e-04 9.86404264e-01
 0.00000000e+00]
```

In [656]:

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

```
1.371050983454026
```

In [657]:

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

```
0.0
```

In [658]:

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

Out[658]:

```
RandomForestClassifier()
```

In [659]:

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

In [660]:

```
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[660]:

```
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 [661]:

```
grid_search.best_score_
```

Out[661]:

```
0.9920408163265306
```

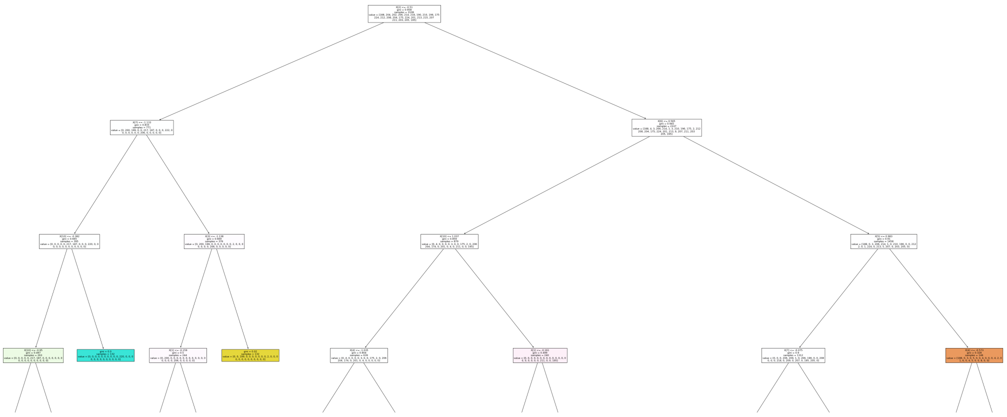
In [662]:

```
rfc_best=grid_search.best_estimator_
```

In [663]:

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

```
[164, 0, 3, 0, 1, 0, 0, 3, 0, 0, 0, 4, 2, 0\n1, 6, 0, 0, 2, 0, 0, 1, 0,
0]'),
 Text(4032.0, 226.5, 'gini = 0.0\nsamples = 91\nvalue = [142, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(4320.0, 226.5, 'gini = 0.721\nsamples = 30\nvalue = [22, 0, 3, 0,
1, 0, 0, 3, 0, 0, 0, 4, 2, 0\n1, 6, 0, 0, 2, 0, 0, 1, 0, 0]')]
```



In [ ]: