In [1]:
```
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
```

In [2]:
```
a=pd.read_csv(r"C:\Users\user\Downloads\C10_air\csvs_per_year\csvs_per_year\ma
```

Out[2]:

	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 [3]:

```
<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 [4]:

```
b=a.fillna(value=257)
```

Out[4]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH
0	2015-10-01 01:00:00	257.0	0.8	257.0	257.00	90.0	82.0	257.0	257.0	257.0	10.0	257.00
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	257.0	1.8	257.00	29.0	97.0	257.0	257.0	257.0	257.0	257.00
3	2015-10-01 01:00:00	257.0	0.6	257.0	257.00	30.0	103.0	2.0	257.0	257.0	257.0	257.00
4	2015-10-01 01:00:00	257.0	257.0	257.0	257.00	95.0	96.0	2.0	257.0	257.0	9.0	257.00
...	...	...	...	...	...	...	...	...	...	...	...	...
210091	2015-08-01 00:00:00	257.0	0.2	257.0	257.00	11.0	33.0	53.0	257.0	257.0	257.0	257.00
210092	2015-08-01 00:00:00	257.0	0.2	257.0	257.00	1.0	5.0	257.0	26.0	257.0	10.0	257.00
210093	2015-08-01 00:00:00	257.0	257.0	257.0	257.00	1.0	7.0	74.0	257.0	257.0	257.0	257.00
210094	2015-08-01 00:00:00	257.0	257.0	257.0	257.00	3.0	7.0	65.0	257.0	257.0	257.0	257.00
210095	2015-08-01 00:00:00	257.0	257.0	257.0	257.00	1.0	9.0	54.0	29.0	257.0	257.0	257.00

210096 rows × 14 columns

In [5]:

Out[5]:

```
Index(['date', 'BEN', 'CO', 'EBE', 'NMHC', 'NO', 'NO_2', 'O_3', 'PM10', 'PM25
',
       'SO_2', 'TCH', 'TOL', 'station'],
      dtype='object')
```

In [6]:
```
c=b.head(11)
```

Out[6]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TO
0	2015-10-01 01:00:00	257.0	0.8	257.0	257.00	90.0	82.0	257.0	257.0	257.0	10.0	257.00	257.
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	257.0	1.8	257.00	29.0	97.0	257.0	257.0	257.0	257.0	257.00	7.
3	2015-10-01 01:00:00	257.0	0.6	257.0	257.00	30.0	103.0	2.0	257.0	257.0	257.0	257.00	257.
4	2015-10-01 01:00:00	257.0	257.0	257.0	257.00	95.0	96.0	2.0	257.0	257.0	9.0	257.00	257.
5	2015-10-01 01:00:00	0.7	0.4	0.3	257.00	35.0	104.0	1.0	26.0	257.0	3.0	257.00	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	257.0	257.0	257.0	257.00	54.0	94.0	1.0	257.0	257.0	257.0	257.00	257.
8	2015-10-01 01:00:00	257.0	0.5	257.0	257.00	38.0	114.0	16.0	257.0	257.0	257.0	257.00	257.
9	2015-10-01 01:00:00	257.0	0.7	257.0	257.00	64.0	97.0	257.0	34.0	257.0	6.0	257.00	257.
10	2015-10-01 01:00:00	0.3	257.0	0.4	257.00	16.0	69.0	257.0	18.0	12.0	3.0	257.00	3.

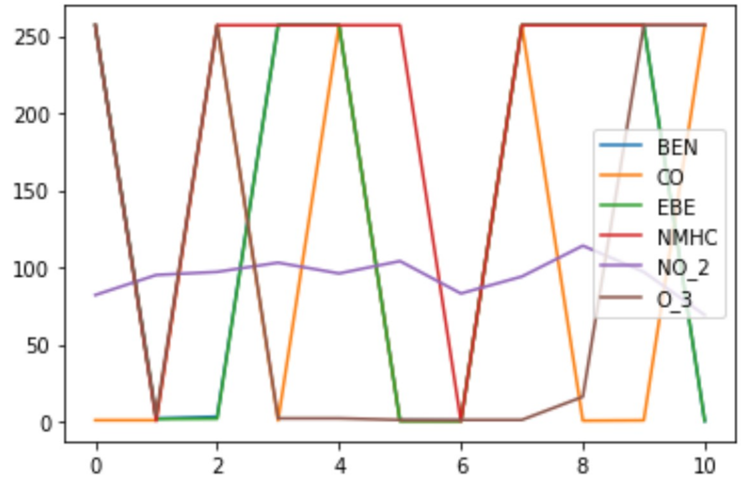
In [7]:
```
d=c[['BEN','CO','EBE','NMHC','NO_2','O_3']]
```

Out[7]:

	BEN	CO	EBE	NMHC	NO_2	O_3
0	257.0	0.8	257.0	257.00	82.0	257.0
1	2.0	0.8	1.6	0.33	95.0	4.0
2	3.1	257.0	1.8	257.00	97.0	257.0
3	257.0	0.6	257.0	257.00	103.0	2.0
4	257.0	257.0	257.0	257.00	96.0	2.0
5	0.7	0.4	0.3	257.00	104.0	1.0
6	0.5	0.3	0.3	0.12	83.0	1.0
7	257.0	257.0	257.0	257.00	94.0	1.0
8	257.0	0.5	257.0	257.00	114.0	16.0
9	257.0	0.7	257.0	257.00	97.0	257.0
10	0.3	257.0	0.4	257.00	69.0	257.0

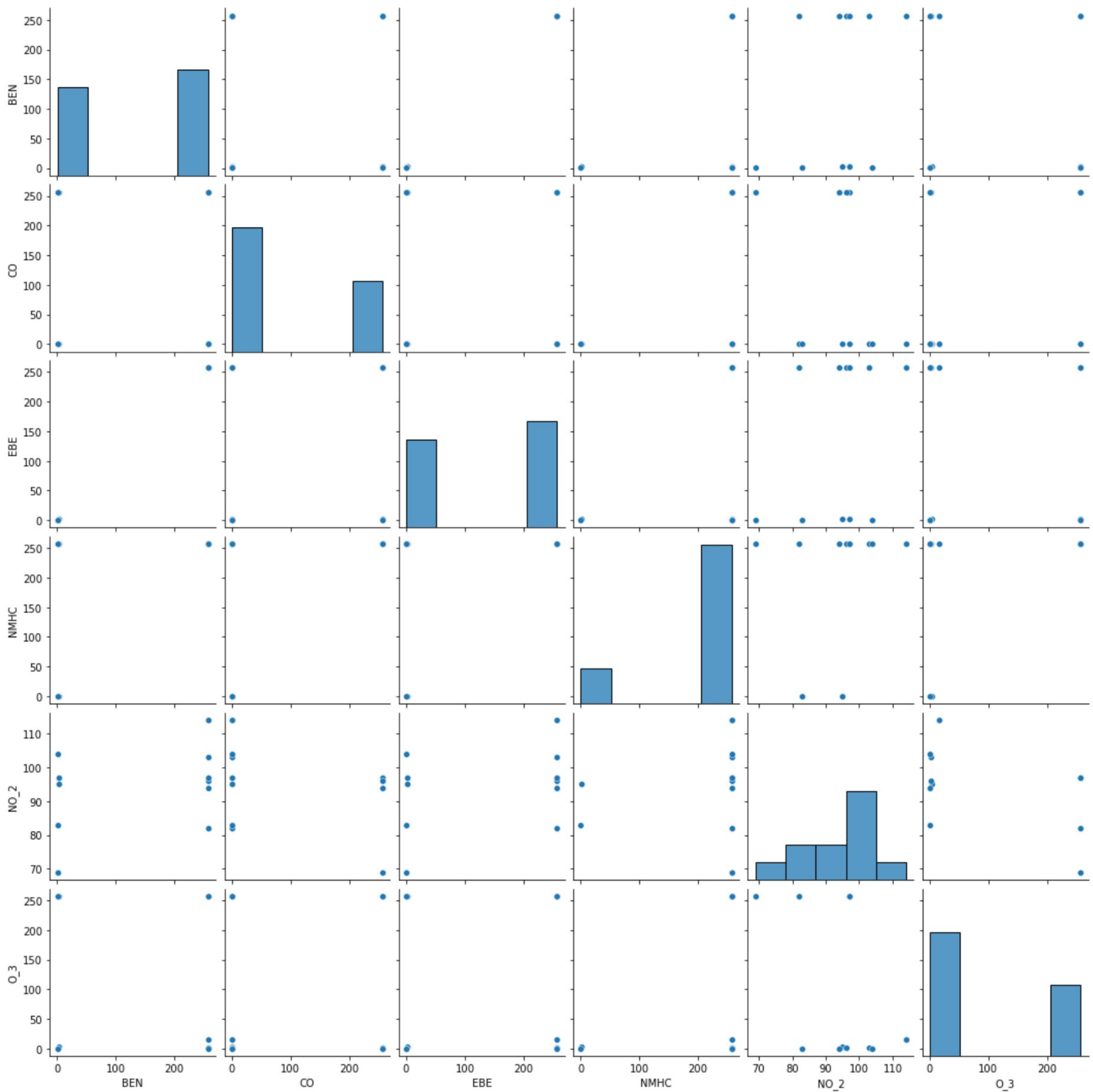
In [8]:

Out[8]: <AxesSubplot:>



In [9]:

Out[9]: <seaborn.axisgrid.PairGrid at 0x22ce5daadc0>



In [10]:
```
x=d[['BEN', 'CO', 'EBE', 'NMHC', 'NO_2']]
```

In [11]:
```
from sklearn.model_selection import train_test_split
```

In [12]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
```

Out[12]: LinearRegression()

In [13]:

```
-5.400124791776761e-13
```

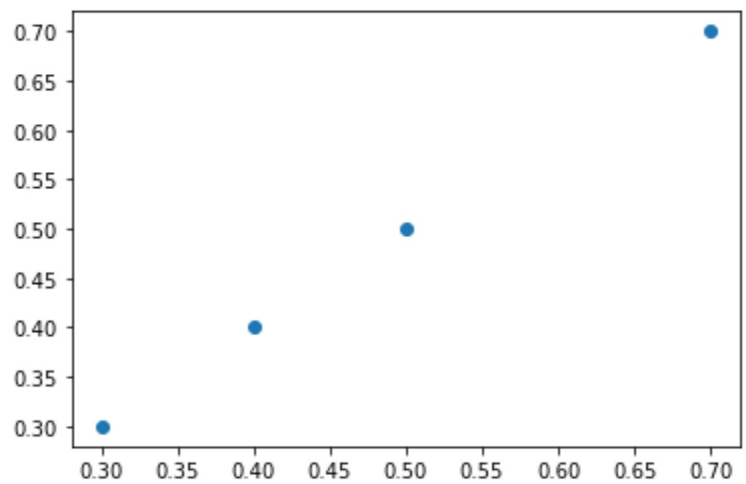
In [14]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
```

Out[14]:

	Co-efficient
BEN	-6.722470e-14
CO	1.000000e+00
EBE	6.672053e-14
NMHC	1.198311e-15
NO_2	2.334646e-15

In [15]:
```
prediction=lr.predict(x_test)
```

Out[15]: <matplotlib.collections.PathCollection at 0x22ce6f7f670>



In [16]:

```
1.0
```

In [17]:

In [18]:
```
rr=Ridge(alpha=10)
```

Out[18]: Ridge(alpha=10)

In [19]:

Out[19]: 0.9680380043101725

In [20]:
```
la=Lasso(alpha=10)
```

Out[20]: Lasso(alpha=10)

In [21]:

Out[21]: 0.619677082719316

In [22]:
```
a1=b.head(6000)
```
Out[22]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	
0	2015-10-01 01:00:00	257.0	0.8	257.0	257.00	90.0	82.0	257.0	257.0	257.0	10.0	257.00	2
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	257.0	1.8	257.00	29.0	97.0	257.0	257.0	257.0	257.0	257.00	
3	2015-10-01 01:00:00	257.0	0.6	257.0	257.00	30.0	103.0	2.0	257.0	257.0	257.0	257.00	2
4	2015-10-01 01:00:00	257.0	257.0	257.0	257.00	95.0	96.0	2.0	257.0	257.0	9.0	257.00	2
...	...	...	...	...	...	...	...	...	...	...	...	...	
5995	2015-10-11 10:00:00	257.0	0.2	257.0	257.00	15.0	38.0	33.0	257.0	257.0	257.0	257.00	2
5996	2015-10-11 10:00:00	257.0	0.2	257.0	257.00	3.0	17.0	257.0	17.0	257.0	12.0	257.00	2
5997	2015-10-11 10:00:00	257.0	257.0	257.0	257.00	1.0	10.0	50.0	257.0	257.0	257.0	257.00	2
5998	2015-10-11 10:00:00	257.0	257.0	257.0	257.00	6.0	12.0	40.0	257.0	257.0	257.0	257.00	2
5999	2015-10-11 10:00:00	257.0	257.0	257.0	257.00	2.0	23.0	28.0	21.0	257.0	257.0	257.00	2

6000 rows × 14 columns

In [23]:
```
e=a1[['BEN', 'CO', 'EBE','NMHC', 'NO_2','O_3',
```
In [24]:
```
f=e.iloc[:,0:14]
```
In [25]:

In [26]:
```
logr=LogisticRegression(max_iter=10000)
```
Out[26]: LogisticRegression(max_iter=10000)

In [27]:
```
from sklearn.model_selection import train_test_split
```
In [28]:

In [29]:
```
prediction=logr.predict(i)
```
```
[28079050]
```

In [30]:

Out[30]:
```
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 [31]:

Out[31]:
```
0.0
```

In [32]:

Out[32]:
```
0.0
```

In [33]:

Out[33]:
```
0.9427777777777778
```

In [34]:
```python
from sklearn.linear_model import ElasticNet
en=ElasticNet()
```

Out[34]:
```
ElasticNet()
```

In [35]:

```
[-1.67116145e-01  9.99910493e-01  1.66647893e-01  1.77658477e-04
  0.00000000e+00]
```

In [36]:

```
0.0812472450688233
```

In [37]:
```python
prediction=en.predict(x_test)
```

```
0.9320694755487613
```

In [38]:
```python
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
```

Out[38]:
```
RandomForestClassifier()
```

In [39]:
```python
parameters={'max_depth':[1,2,3,4,5],
 'min_samples_leaf':[5,10,15,20,25],
 'n_estimators':[10,20,30,40,50]
```

In [40]:
```python
from sklearn.model_selection import GridSearchCV
grid_search=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="acc
```

Out[40]:
```
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 [41]:

Out[41]:
```
0.99
```

In [42]:

In [43]:
```python
from sklearn.tree import plot_tree
plt.figure(figsize=(80,50))
```

Out[43]:
```
[Text(1918.125, 2491.5, 'X[7] <= -0.126\ngini = 0.958\nsamples = 2654\nvalue
= [187, 184, 177, 160, 208, 178, 158, 180, 185, 166\n166, 179, 179, 188, 178,
151, 187, 182, 165, 194\n151, 170, 180, 147]'),
 Text(976.5, 2038.5, 'X[10] <= -1.803\ngini = 0.899\nsamples = 1092\nvalue =
[186, 184, 0, 0, 208, 176, 158, 0, 171, 164, 166\n0, 179, 0, 0, 0, 0, 0, 0,
0, 151, 0, 0, 0]'),
 Text(837.0, 1585.5, 'gini = 0.0\nsamples = 113\nvalue = [186, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(1116.0, 1585.5, 'X[0] <= -0.577\ngini = 0.888\nsamples = 979\nvalue =
[0, 184, 0, 0, 208, 176, 158, 0, 171, 164, 166, 0\n179, 0, 0, 0, 0, 0, 0, 0,
151, 0, 0, 0]'),
 Text(558.0, 1132.5, 'X[3] <= -1.148\ngini = 0.749\nsamples = 415\nvalue =
[0, 182, 0, 0, 0, 171, 158, 0, 0, 0, 164, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0]'),
 Text(279.0, 679.5, 'X[1] <= -1.187\ngini = 0.498\nsamples = 208\nvalue = [0,
182, 0, 0, 0, 0, 158, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(139.5, 226.5, 'gini = 0.153\nsamples = 87\nvalue = [0, 12, 0, 0, 0, 0,
132, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(418.5, 226.5, 'gini = 0.23\nsamples = 121\nvalue = [0, 170, 0, 0, 0, 0,
```

From this observation I had observe that the RIDGE is a highest accuracy of 0.9680380043101725

In [ ]: