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	CH4	CO	EBE	NMHC	NO	NO_2	NOx	O_3	PM10	PM25	SO_2
0	2018-03-01 01:00:00	NaN	NaN	0.3	NaN	NaN	1.0	29.0	31.0	NaN	NaN	NaN	2.0
1	2018-03-01 01:00:00	0.5	1.39	0.3	0.2	0.02	6.0	40.0	49.0	52.0	5.0	4.0	3.0
2	2018-03-01 01:00:00	0.4	NaN	NaN	0.2	NaN	4.0	41.0	47.0	NaN	NaN	NaN	NaN
3	2018-03-01 01:00:00	NaN	NaN	0.3	NaN	NaN	1.0	35.0	37.0	54.0	NaN	NaN	NaN
4	2018-03-01 01:00:00	NaN	NaN	NaN	NaN	NaN	1.0	27.0	29.0	49.0	NaN	NaN	3.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
69091	2018-02-01 00:00:00	NaN	NaN	0.5	NaN	NaN	66.0	91.0	192.0	1.0	35.0	22.0	NaN
69092	2018-02-01 00:00:00	NaN	NaN	0.7	NaN	NaN	87.0	107.0	241.0	NaN	29.0	NaN	15.0
69093	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	28.0	48.0	91.0	2.0	NaN	NaN	NaN
69094	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	141.0	103.0	320.0	2.0	NaN	NaN	NaN
69095	2018-02-01 00:00:00	NaN	NaN	NaN	NaN	NaN	69.0	96.0	202.0	3.0	26.0	NaN	NaN

69096 rows × 16 columns

In [3]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 69096 entries, 0 to 69095
Data columns (total 16 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   date     69096 non-null  object 
 1   BEN      16950 non-null  float64
 2   CH4      8440 non-null   float64
 3   CO       28598 non-null  float64
 4   EBE      16949 non-null  float64
 5   NMHC     8440 non-null   float64
 6   NO       68826 non-null  float64
 7   NO_2     68826 non-null  float64
 8   NOx      68826 non-null  float64
 9   O_3      40049 non-null  float64
 10  PM10     36911 non-null  float64
 11  PM25     18912 non-null  float64
 12  SO_2     28586 non-null  float64
 13  TCH      8440 non-null   float64
 14  TOL      16950 non-null  float64
 15  station  69096 non-null  int64  
dtypes: float64(14), int64(1), object(1)
memory usage: 8.4+ MB
```

In [4]:
```
b=a.fillna(value=333)
```

Out[4]:

	date	BEN	CH4	CO	EBE	NMHC	NO	NO_2	NOx	O_3	PM10	PM25
0	2018-03-01 01:00:00	333.0	333.00	0.3	333.0	333.00	1.0	29.0	31.0	333.0	333.0	333.0
1	2018-03-01 01:00:00	0.5	1.39	0.3	0.2	0.02	6.0	40.0	49.0	52.0	5.0	4.0
2	2018-03-01 01:00:00	0.4	333.00	333.0	0.2	333.00	4.0	41.0	47.0	333.0	333.0	333.0
3	2018-03-01 01:00:00	333.0	333.00	0.3	333.0	333.00	1.0	35.0	37.0	54.0	333.0	333.0
4	2018-03-01 01:00:00	333.0	333.00	333.0	333.0	333.00	1.0	27.0	29.0	49.0	333.0	333.0
...	...	...	...	...	...	...	...	...	...	...	...	...
69091	2018-02-01 00:00:00	333.0	333.00	0.5	333.0	333.00	66.0	91.0	192.0	1.0	35.0	22.0
69092	2018-02-01 00:00:00	333.0	333.00	0.7	333.0	333.00	87.0	107.0	241.0	333.0	29.0	333.0
69093	2018-02-01 00:00:00	333.0	333.00	333.0	333.0	333.00	28.0	48.0	91.0	2.0	333.0	333.0
69094	2018-02-01 00:00:00	333.0	333.00	333.0	333.0	333.00	141.0	103.0	320.0	2.0	333.0	333.0
69095	2018-02-01 00:00:00	333.0	333.00	333.0	333.0	333.00	69.0	96.0	202.0	3.0	26.0	333.0

69096 rows × 16 columns

In [5]:
```

```

Out[5]:
```
Index(['date', 'BEN', 'CH4', 'CO', 'EBE', 'NMHC', 'NO', 'NO_2', 'NOx', 'O_3',
       'PM10', 'PM25', 'SO_2', 'TCH', 'TOL', 'station'],
      dtype='object')
```

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

Out[6]:

	date	BEN	CH4	CO	EBE	NMHC	NO	NO_2	NOx	O_3	PM10	PM25	SO_2
0	2018-03-01 01:00:00	333.0	333.00	0.3	333.0	333.00	1.0	29.0	31.0	333.0	333.0	333.0	2.0
1	2018-03-01 01:00:00	0.5	1.39	0.3	0.2	0.02	6.0	40.0	49.0	52.0	5.0	4.0	3.0
2	2018-03-01 01:00:00	0.4	333.00	333.0	0.2	333.00	4.0	41.0	47.0	333.0	333.0	333.0	333.0
3	2018-03-01 01:00:00	333.0	333.00	0.3	333.0	333.00	1.0	35.0	37.0	54.0	333.0	333.0	333.0
4	2018-03-01 01:00:00	333.0	333.00	333.0	333.0	333.00	1.0	27.0	29.0	49.0	333.0	333.0	3.0
5	2018-03-01 01:00:00	0.3	333.00	0.3	0.2	333.00	1.0	27.0	29.0	57.0	8.0	333.0	6.0
6	2018-03-01 01:00:00	0.4	1.11	0.2	0.1	0.06	1.0	25.0	27.0	55.0	5.0	4.0	4.0
7	2018-03-01 01:00:00	333.0	333.00	333.0	333.0	333.00	1.0	37.0	39.0	54.0	333.0	333.0	333.0
8	2018-03-01 01:00:00	333.0	333.00	0.5	333.0	333.00	3.0	43.0	47.0	29.0	333.0	333.0	5.0
9	2018-03-01 01:00:00	333.0	333.00	0.2	333.0	333.00	2.0	26.0	29.0	333.0	4.0	333.0	6.0
10	2018-03-01 01:00:00	0.4	333.00	333.0	0.3	333.00	2.0	30.0	34.0	333.0	2.0	2.0	3.0

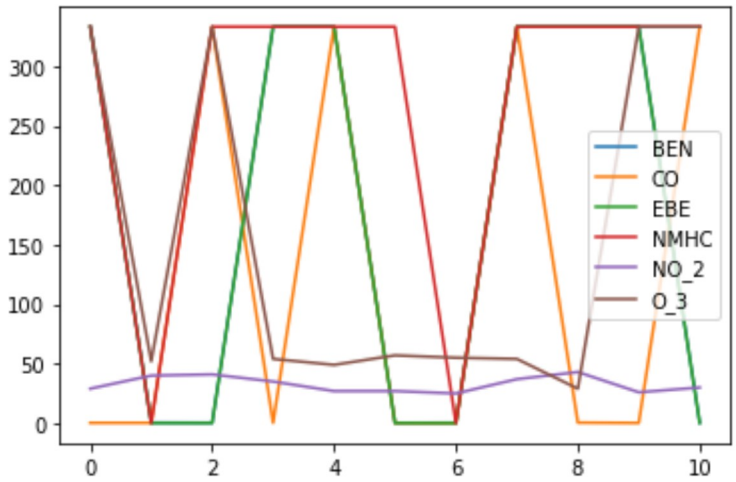
In [7]:
```
d=c[['BEN','CO','EBE','NMHC','NO_2','O_3']]
```

Out[7]:

	BEN	CO	EBE	NMHC	NO_2	O_3
0	333.0	0.3	333.0	333.00	29.0	333.0
1	0.5	0.3	0.2	0.02	40.0	52.0
2	0.4	333.0	0.2	333.00	41.0	333.0
3	333.0	0.3	333.0	333.00	35.0	54.0
4	333.0	333.0	333.0	333.00	27.0	49.0
5	0.3	0.3	0.2	333.00	27.0	57.0
6	0.4	0.2	0.1	0.06	25.0	55.0
7	333.0	333.0	333.0	333.00	37.0	54.0
8	333.0	0.5	333.0	333.00	43.0	29.0
9	333.0	0.2	333.0	333.00	26.0	333.0
10	0.4	333.0	0.3	333.00	30.0	333.0

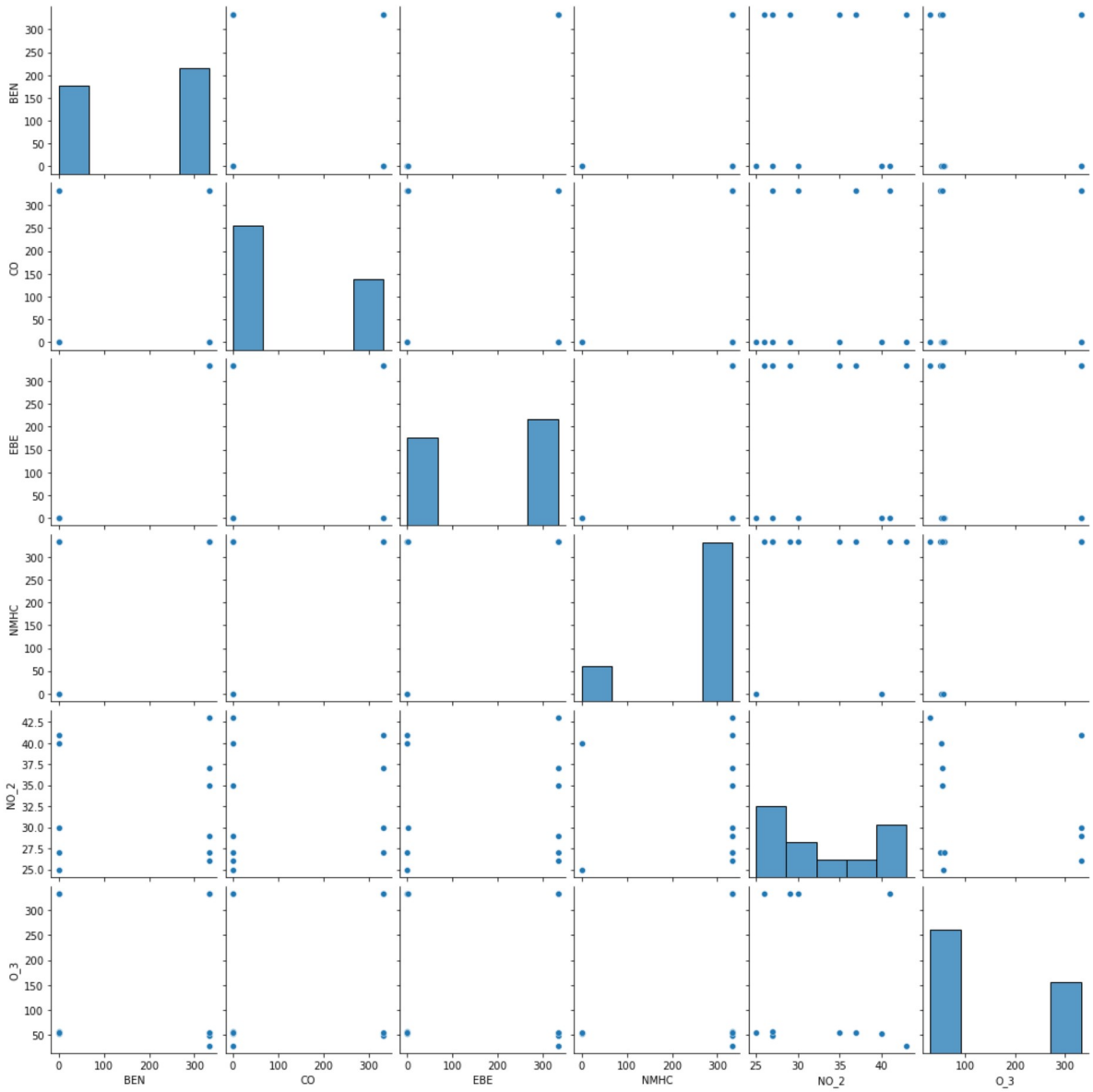
In [8]:

Out[8]: <AxesSubplot:>



In [9]:

Out[9]: <seaborn.axisgrid.PairGrid at 0x20f49017130>



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]:

```
-2.842170943040401e-14
```

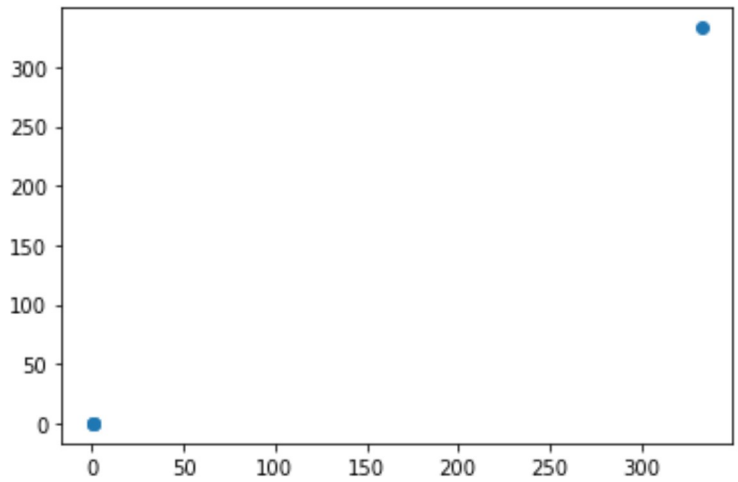
In [14]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
```

Out[14]:

	Co-efficient
BEN	-3.104025e-13
CO	1.000000e+00
EBE	3.104587e-13
NMHC	1.459823e-16
NO_2	-1.066815e-15

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

Out[15]: <matplotlib.collections.PathCollection at 0x20f4b5f85b0>



In [16]:

```
1.0
```

In [17]:

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

Out[18]:
```
Ridge(alpha=10)
```

In [19]:

Out[19]:
```
0.9999999680867979
```

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

Out[20]:
```
Lasso(alpha=10)
```

In [21]:

Out[21]:
```
0.9999998406945124
```

In [22]:
```
a1=b.head(6500)
```

Out[22]:

	date	BEN	CH4	CO	EBE	NMHC	NO	NO_2	NOx	O_3	PM10	PM25	SC
0	2018-03-01 01:00:00	333.0	333.00	0.3	333.0	333.00	1.0	29.0	31.0	333.0	333.0	333.0	
1	2018-03-01 01:00:00	0.5	1.39	0.3	0.2	0.02	6.0	40.0	49.0	52.0	5.0	4.0	
2	2018-03-01 01:00:00	0.4	333.00	333.0	0.2	333.00	4.0	41.0	47.0	333.0	333.0	333.0	33
3	2018-03-01 01:00:00	333.0	333.00	0.3	333.0	333.00	1.0	35.0	37.0	54.0	333.0	333.0	33
4	2018-03-01 01:00:00	333.0	333.00	333.0	333.0	333.00	1.0	27.0	29.0	49.0	333.0	333.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
6495	2018-03-12 07:00:00	333.0	333.00	333.0	333.0	333.00	1.0	12.0	14.0	66.0	333.0	333.0	33
6496	2018-03-12 07:00:00	333.0	333.00	333.0	333.0	333.00	22.0	19.0	52.0	333.0	1.0	1.0	33
6497	2018-03-12 07:00:00	333.0	333.00	333.0	333.0	333.00	1.0	16.0	17.0	65.0	333.0	333.0	33
6498	2018-03-12 07:00:00	0.4	1.24	333.0	0.1	0.04	1.0	14.0	16.0	333.0	6.0	333.0	33
6499	2018-03-12 07:00:00	333.0	333.00	0.2	333.0	333.00	4.0	18.0	24.0	69.0	8.0	3.0	33

6500 rows × 16 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]:
```
0.9492307692307692
```

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

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

In [35]:

```
[-0.41602473  0.99985562  0.41568003 -0.          0.        ]
```

In [36]:

```
0.12191672042280288
```

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

```
0.9999999016312336
```

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

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

In [39]:
```
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]:
```
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]:

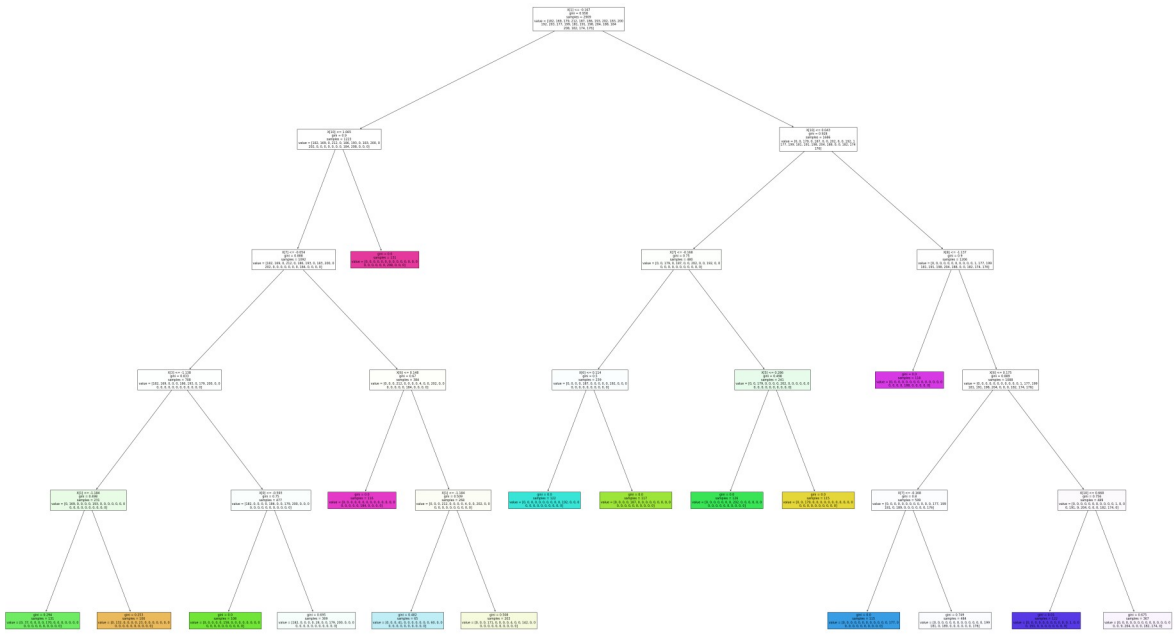
Out[41]:
```
0.9980219780219781
```

In [42]:

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

Out[43]:
```
[Text(2189.076923076923, 2491.5, 'X[1] <= -0.167\ngini = 0.958\nsamples = 290
9\nvalue = [182, 169, 179, 212, 187, 186, 193, 202, 183, 200\n192, 203, 177,
199, 181, 191, 198, 204, 188, 184\n208, 182, 174, 176]'),
 Text(1287.6923076923076, 2038.5, 'X[10] <= 1.065\ngini = 0.9\nsamples = 122
3\nvalue = [182, 169, 0, 212, 0, 186, 193, 0, 183, 200, 0\n202, 0, 0, 0, 0,
0, 0, 0, 184, 208, 0, 0, 0]'),
 Text(1116.0, 1585.5, 'X[7] <= -0.054\ngini = 0.888\nsamples = 1092\nvalue =
[182, 169, 0, 212, 0, 186, 193, 0, 183, 200, 0\n202, 0, 0, 0, 0, 0, 0, 0, 18
4, 0, 0, 0, 0]'),
 Text(686.7692307692307, 1132.5, 'X[3] <= -1.138\ngini = 0.833\nsamples = 70
8\nvalue = [182, 169, 0, 0, 0, 186, 193, 0, 179, 200, 0, 0\n0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0]'),
 Text(343.38461538461536, 679.5, 'X[1] <= -1.184\ngini = 0.498\nsamples = 23
1\nvalue = [0, 169, 0, 0, 0, 0, 193, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0,
0, 0, 0, 0]'),
 Text(171.69230769230768, 226.5, 'gini = 0.294\nsamples = 131\nvalue = [0, 3
7, 0, 0, 0, 0, 170, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(515.0769230769231, 226.5, 'gini = 0.253\nsamples = 100\nvalue = [0, 13
2, 0, 0, 0, 0, 23, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(1030.1538461538462, 679.5, 'X[0] <= -0.593\ngini = 0.75\nsamples = 477\
nvalue = [182, 0, 0, 0, 0, 186, 0, 0, 179, 200, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0]'),
 Text(858.4615384615383, 226.5, 'gini = 0.0\nsamples = 108\nvalue = [0, 0, 0,
0, 0, 158, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(1201.8461538461538, 226.5, 'gini = 0.695\nsamples = 369\nvalue = [182,
0, 0, 0, 0, 28, 0, 0, 179, 200, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(1545.230769230769, 1132.5, 'X[6] <= 0.148\ngini = 0.67\nsamples = 384\n
value = [0, 0, 0, 212, 0, 0, 0, 0, 4, 0, 0, 202, 0, 0\n0, 0, 0, 0, 0, 184, 0,
0, 0, 0]'),
 Text(1373.5384615384614, 679.5, 'gini = 0.0\nsamples = 116\nvalue = [0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 184, 0, 0, 0, 0]'),
 Text(1716.9230769230767, 679.5, 'X[1] <= -1.184\ngini = 0.509\nsamples = 26
8\nvalue = [0, 0, 0, 212, 0, 0, 0, 0, 4, 0, 0, 202, 0, 0\n0, 0, 0, 0, 0, 0,
0, 0, 0, 0]'),
 Text(1545.230769230769, 226.5, 'gini = 0.482\nsamples = 65\nvalue = [0, 0,
0, 41, 0, 0, 0, 0, 0, 0, 0, 60, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(1888.6153846153845, 226.5, 'gini = 0.508\nsamples = 203\nvalue = [0, 0,
0, 171, 0, 0, 0, 0, 4, 0, 0, 142, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(1459.3846153846152, 1585.5, 'gini = 0.0\nsamples = 131\nvalue = [0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 208, 0, 0, 0]'),
 Text(3090.461538461538, 2038.5, 'X[10] <= 0.043\ngini = 0.928\nsamples = 168
6\nvalue = [0, 0, 179, 0, 187, 0, 0, 202, 0, 0, 192, 1\n177, 199, 181, 191, 1
98, 204, 188, 0, 0, 182, 174\n176]'),
 Text(2575.3846153846152, 1585.5, 'X[7] <= -0.168\ngini = 0.75\nsamples = 48
0\nvalue = [0, 0, 179, 0, 187, 0, 0, 202, 0, 0, 192, 0, 0\n0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0]'),
 Text(2232.0, 1132.5, 'X[6] <= 0.114\ngini = 0.5\nsamples = 239\nvalue = [0,
0, 0, 0, 187, 0, 0, 0, 0, 0, 192, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(2060.3076923076924, 679.5, 'gini = 0.0\nsamples = 122\nvalue = [0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 192, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(2403.6923076923076, 679.5, 'gini = 0.0\nsamples = 117\nvalue = [0, 0,
0, 0, 187, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(2918.7692307692305, 1132.5, 'X[5] <= 0.286\ngini = 0.498\nsamples = 24
```

```
1\nvalue = [0, 0, 179, 0, 0, 0, 0, 202, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0,
0, 0, 0, 0]'),
 Text(2747.076923076923, 679.5, 'gini = 0.0\nsamples = 126\nvalue = [0, 0, 0,
0, 0, 0, 0, 202, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(3090.461538461538, 679.5, 'gini = 0.0\nsamples = 115\nvalue = [0, 0, 17
9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(3605.5384615384614, 1585.5, 'X[8] <= -1.137\ngini = 0.9\nsamples = 120
6\nvalue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 177, 199\n181, 191, 198, 204,
188, 0, 0, 182, 174, 176]'),
 Text(3433.8461538461534, 1132.5, 'gini = 0.0\nsamples = 118\nvalue = [0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 188, 0, 0, 0, 0, 0]'),
 Text(3777.230769230769, 1132.5, 'X[6] <= 0.175\ngini = 0.889\nsamples = 108
8\nvalue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 177, 199\n181, 191, 198, 204,
0, 0, 0, 182, 174, 176]'),
 Text(3433.8461538461534, 679.5, 'X[7] <= -0.168\ngini = 0.8\nsamples = 599\n
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 177, 199\n181, 0, 189, 0, 0, 0,
0, 0, 0, 176]'),
 Text(3262.1538461538457, 226.5, 'gini = 0.0\nsamples = 115\nvalue = [0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 177, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(3605.5384615384614, 226.5, 'gini = 0.749\nsamples = 484\nvalue = [0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 199\n181, 0, 189, 0, 0, 0, 0, 0, 0, 176]'),
 Text(4120.615384615385, 679.5, 'X[10] <= 0.668\ngini = 0.756\nsamples = 489\
nvalue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0\n0, 191, 9, 204, 0, 0, 0,
182, 174, 0]'),
 Text(3948.9230769230767, 226.5, 'gini = 0.01\nsamples = 122\nvalue = [0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0\n0, 191, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(4292.307692307692, 226.5, 'gini = 0.675\nsamples = 367\nvalue = [0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 9, 204, 0, 0, 0, 182, 174, 0]')]
```



From this observation I had observe that the RIDGE is a highest accuracy of 0.9999999680867979

In [ ]: