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	2014-06-01 01:00:00	NaN	0.2	NaN	NaN	3.0	10.0	NaN	NaN	NaN	3.0	NaN	NaN
1	2014-06-01 01:00:00	0.2	0.2	0.1	0.11	3.0	17.0	68.0	10.0	5.0	5.0	1.36	1.3
2	2014-06-01 01:00:00	0.3	NaN	0.1	NaN	2.0	6.0	NaN	NaN	NaN	NaN	NaN	1.1
3	2014-06-01 01:00:00	NaN	0.2	NaN	NaN	1.0	6.0	79.0	NaN	NaN	NaN	NaN	NaN
4	2014-06-01 01:00:00	NaN	NaN	NaN	NaN	1.0	6.0	75.0	NaN	NaN	4.0	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
210019	2014-09-01 00:00:00	NaN	0.5	NaN	NaN	20.0	84.0	29.0	NaN	NaN	NaN	NaN	NaN
210020	2014-09-01 00:00:00	NaN	0.3	NaN	NaN	1.0	22.0	NaN	15.0	NaN	6.0	NaN	NaN
210021	2014-09-01 00:00:00	NaN	NaN	NaN	NaN	1.0	13.0	70.0	NaN	NaN	NaN	NaN	NaN
210022	2014-09-01 00:00:00	NaN	NaN	NaN	NaN	3.0	38.0	42.0	NaN	NaN	NaN	NaN	NaN
210023	2014-09-01 00:00:00	NaN	NaN	NaN	NaN	1.0	26.0	65.0	11.0	NaN	NaN	NaN	NaN

210024 rows × 14 columns

In [3]:

```
.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210024 entries, 0 to 210023
Data columns (total 14 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     210024 non-null  object 
 1   BEN      46703 non-null   float64
 2   CO       87023 non-null   float64
 3   EBE      46722 non-null   float64
 4   NMHC     25021 non-null   float64
 5   NO       209154 non-null  float64
 6   NO_2     209154 non-null  float64
 7   O_3      121681 non-null  float64
 8   PM10     104311 non-null  float64
 9   PM25     51954 non-null   float64
 10  SO_2     87141 non-null   float64
 11  TCH      25021 non-null   float64
 12  TOL      46570 non-null   float64
 13  station  210024 non-null  int64  
dtypes: float64(12), int64(1), object(1)
memory usage: 22.4+ MB
```

In [4]:
```
b=a.fillna(value=222)
```

Out[4]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH
0	2014-06-01 01:00:00	222.0	0.2	222.0	222.00	3.0	10.0	222.0	222.0	222.0	3.0	222.00
1	2014-06-01 01:00:00	0.2	0.2	0.1	0.11	3.0	17.0	68.0	10.0	5.0	5.0	1.36
2	2014-06-01 01:00:00	0.3	222.0	0.1	222.00	2.0	6.0	222.0	222.0	222.0	222.0	222.00
3	2014-06-01 01:00:00	222.0	0.2	222.0	222.00	1.0	6.0	79.0	222.0	222.0	222.0	222.00
4	2014-06-01 01:00:00	222.0	222.0	222.0	222.00	1.0	6.0	75.0	222.0	222.0	4.0	222.00
...	...	...	...	...	...	...	...	...	...	...	...	...
210019	2014-09-01 00:00:00	222.0	0.5	222.0	222.00	20.0	84.0	29.0	222.0	222.0	222.0	222.00
210020	2014-09-01 00:00:00	222.0	0.3	222.0	222.00	1.0	22.0	222.0	15.0	222.0	6.0	222.00
210021	2014-09-01 00:00:00	222.0	222.0	222.0	222.00	1.0	13.0	70.0	222.0	222.0	222.0	222.00
210022	2014-09-01 00:00:00	222.0	222.0	222.0	222.00	3.0	38.0	42.0	222.0	222.0	222.0	222.00
210023	2014-09-01 00:00:00	222.0	222.0	222.0	222.00	1.0	26.0	65.0	11.0	222.0	222.0	222.00

210024 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	TOL
0	2014-06-01 01:00:00	222.0	0.2	222.0	222.00	3.0	10.0	222.0	222.0	222.0	3.0	222.00	222.(
1	2014-06-01 01:00:00	0.2	0.2	0.1	0.11	3.0	17.0	68.0	10.0	5.0	5.0	1.36	1.3
2	2014-06-01 01:00:00	0.3	222.0	0.1	222.00	2.0	6.0	222.0	222.0	222.0	222.0	222.00	1.1
3	2014-06-01 01:00:00	222.0	0.2	222.0	222.00	1.0	6.0	79.0	222.0	222.0	222.0	222.00	222.(
4	2014-06-01 01:00:00	222.0	222.0	222.0	222.00	1.0	6.0	75.0	222.0	222.0	4.0	222.00	222.(
5	2014-06-01 01:00:00	0.1	0.4	0.1	222.00	1.0	10.0	83.0	7.0	222.0	2.0	222.00	0.2
6	2014-06-01 01:00:00	0.1	0.2	0.1	0.23	1.0	5.0	80.0	4.0	3.0	2.0	1.21	0.1
7	2014-06-01 01:00:00	222.0	222.0	222.0	222.00	1.0	1.0	86.0	222.0	222.0	222.0	222.00	222.(
8	2014-06-01 01:00:00	222.0	0.3	222.0	222.00	5.0	22.0	68.0	222.0	222.0	4.0	222.00	222.(
9	2014-06-01 01:00:00	222.0	0.2	222.0	222.00	1.0	4.0	222.0	14.0	222.0	1.0	222.00	222.(
10	2014-06-01 01:00:00	0.1	222.0	0.1	222.00	6.0	18.0	222.0	8.0	5.0	2.0	222.00	0.7

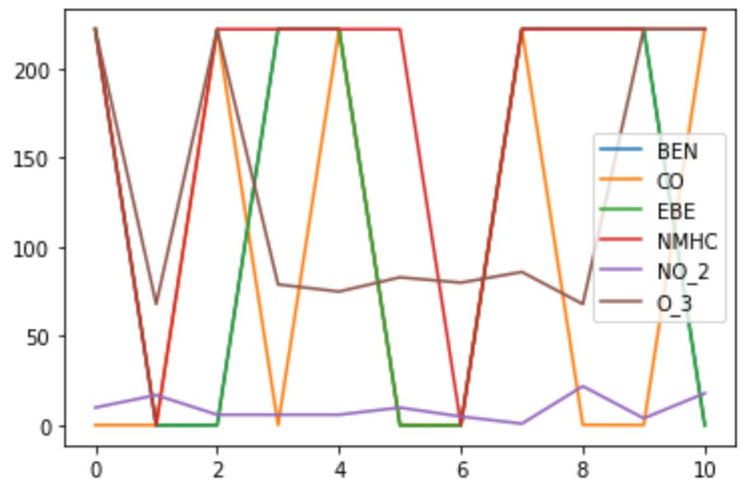
In [7]:
```
d=c[['BEN','CO','EBE','NMHC','NO_2','O_3']]
```

Out[7]:

	BEN	CO	EBE	NMHC	NO_2	O_3
0	222.0	0.2	222.0	222.00	10.0	222.0
1	0.2	0.2	0.1	0.11	17.0	68.0
2	0.3	222.0	0.1	222.00	6.0	222.0
3	222.0	0.2	222.0	222.00	6.0	79.0
4	222.0	222.0	222.0	222.00	6.0	75.0
5	0.1	0.4	0.1	222.00	10.0	83.0
6	0.1	0.2	0.1	0.23	5.0	80.0
7	222.0	222.0	222.0	222.00	1.0	86.0
8	222.0	0.3	222.0	222.00	22.0	68.0
9	222.0	0.2	222.0	222.00	4.0	222.0
10	0.1	222.0	0.1	222.00	18.0	222.0

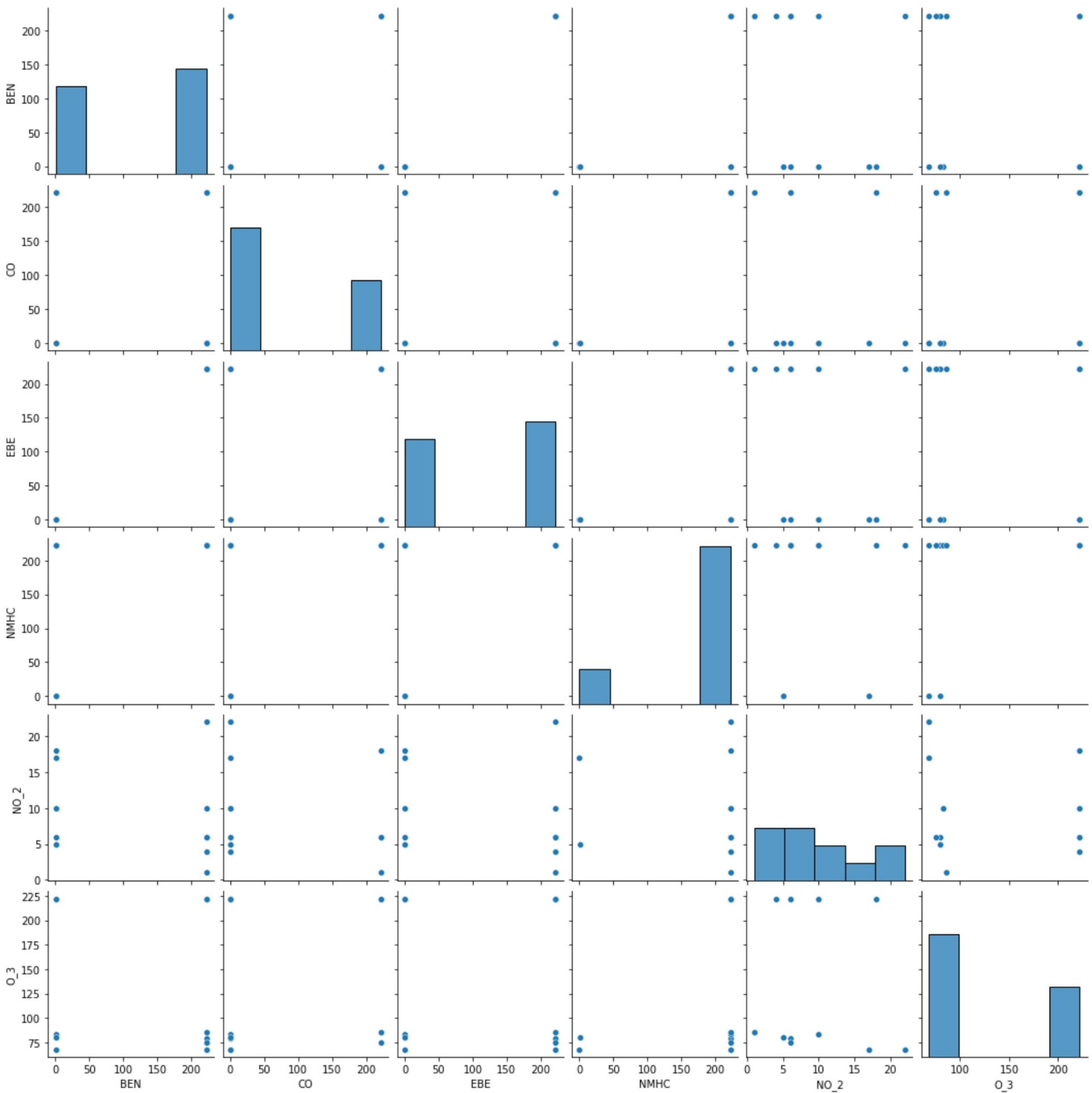
In [8]:

Out[8]: <AxesSubplot:>



In [9]:

Out[9]: <seaborn.axisgrid.PairGrid at 0x27702e9c340>



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]:

```
0.0
```

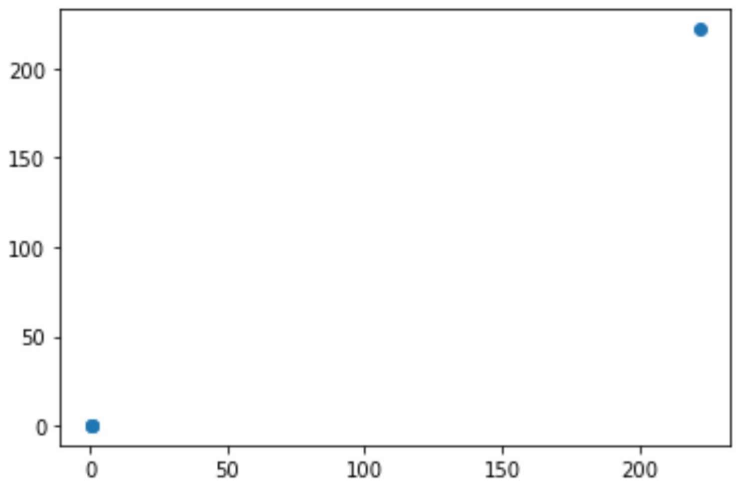
In [14]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
```

Out[14]:

	Co-efficient
BEN	1.888206e-13
CO	1.000000e+00
EBE	-1.892682e-13
NMHC	4.802573e-16
NO_2	-1.703788e-15

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

Out[15]: <matplotlib.collections.PathCollection at 0x2770578e4c0>



In [16]:

```
1.0
```

In [17]:

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

Out[18]: Ridge(alpha=10)

In [19]:

Out[19]: 0.9999999543984697

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

Out[20]: Lasso(alpha=10)

In [21]:

Out[21]: 0.9999991931063271

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

Out[22]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	
0	2014-06-01 01:00:00	222.0	0.2	222.0	222.00	3.0	10.0	222.0	222.0	222.0	3.0	222.00	2
1	2014-06-01 01:00:00	0.2	0.2	0.1	0.11	3.0	17.0	68.0	10.0	5.0	5.0	1.36	
2	2014-06-01 01:00:00	0.3	222.0	0.1	222.00	2.0	6.0	222.0	222.0	222.0	222.0	222.00	
3	2014-06-01 01:00:00	222.0	0.2	222.0	222.00	1.0	6.0	79.0	222.0	222.0	222.0	222.00	2
4	2014-06-01 01:00:00	222.0	222.0	222.0	222.00	1.0	6.0	75.0	222.0	222.0	4.0	222.00	2
...	...	...	...	...	...	...	...	...	...	...	...	...	
5995	2014-06-11 10:00:00	222.0	0.4	222.0	222.00	31.0	73.0	45.0	222.0	222.0	222.0	222.00	2
5996	2014-06-11 10:00:00	222.0	0.3	222.0	222.00	8.0	24.0	222.0	24.0	222.0	6.0	222.00	2
5997	2014-06-11 10:00:00	222.0	222.0	222.0	222.00	1.0	10.0	88.0	222.0	222.0	222.0	222.00	2
5998	2014-06-11 10:00:00	222.0	222.0	222.0	222.00	7.0	17.0	70.0	222.0	222.0	222.0	222.00	2
5999	2014-06-11 10:00:00	222.0	222.0	222.0	222.00	5.0	27.0	222.0	30.0	222.0	222.0	222.00	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]: 0.95

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

Out[34]: ElasticNet()

In [35]:

```
[-1.65763063e-01  9.99965212e-01  1.65669519e-01  6.09045377e-06
 -0.00000000e+00]
```

In [36]:

```
0.015797123874477847
```

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

```
0.9999999957832388
```

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]:
```
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]:

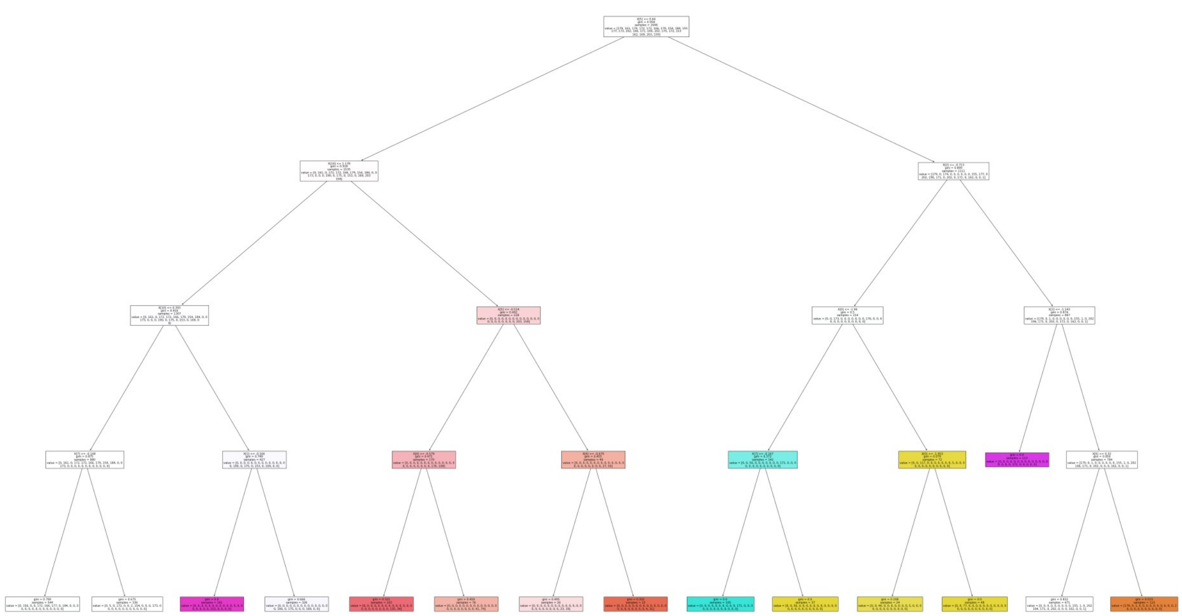
Out[41]:
```
0.9983333333333333
```

In [42]:

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

Out[43]:
```
[Text(2431.285714285714, 2446.2, 'X[5] <= 0.64\ngini = 0.958\nsamples = 2646\
nvalue = [179, 161, 174, 172, 172, 166, 179, 154, 184, 155\n177, 173, 202, 19
6, 171, 190, 202, 175, 172, 153\n162, 169, 203, 159]'),
 Text(1275.4285714285713, 1902.6, 'X[10] <= 1.178\ngini = 0.928\nsamples = 15
35\nvalue = [0, 161, 0, 172, 172, 166, 179, 154, 184, 0, 0\n173, 0, 0, 0, 19
0, 0, 175, 0, 153, 0, 169, 203\n158]'),
 Text(637.7142857142857, 1359.0, 'X[10] <= 0.355\ngini = 0.916\nsamples = 130
7\nvalue = [0, 161, 0, 172, 172, 166, 179, 154, 184, 0, 0\n173, 0, 0, 0, 190,
0, 175, 0, 153, 0, 169, 0\n0]'),
 Text(318.85714285714283, 815.3999999999999, 'X[7] <= -0.148\ngini = 0.875\ns
amples = 880\nvalue = [0, 161, 0, 172, 172, 166, 179, 154, 184, 0, 0\n173, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(159.42857142857142, 271.7999999999997, 'gini = 0.799\nsamples = 544\nva
lue = [0, 156, 0, 0, 172, 166, 177, 0, 184, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0]'),
 Text(478.2857142857142, 271.7999999999997, 'gini = 0.675\nsamples = 336\nval
ue = [0, 5, 0, 172, 0, 0, 2, 154, 0, 0, 0, 173, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0]'),
 Text(956.5714285714284, 815.3999999999999, 'X[1] <= -0.166\ngini = 0.749\nsa
mples = 427\nvalue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 190, 0, 17
5, 0, 153, 0, 169, 0, 0]'),
 Text(797.1428571428571, 271.7999999999997, 'gini = 0.0\nsamples = 101\nvalue
= [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 153, 0, 0, 0,
0]'),
 Text(1116.0, 271.7999999999997, 'gini = 0.666\nsamples = 326\nvalue = [0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 190, 0, 175, 0, 0, 0, 169, 0, 0]'),
 Text(1913.1428571428569, 1359.0, 'X[5] <= -0.514\ngini = 0.492\nsamples = 22
8\nvalue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0,
203, 158]'),
 Text(1594.2857142857142, 815.3999999999999, 'X[4] <= -0.579\ngini = 0.471\ns
amples = 179\nvalue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 176, 108]'),
 Text(1434.8571428571427, 271.7999999999997, 'gini = 0.321\nsamples = 103\nva
lue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 135,
34]'),
 Text(1753.7142857142856, 271.7999999999997, 'gini = 0.459\nsamples = 76\nval
ue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 41, 7
4]'),
 Text(2232.0, 815.3999999999999, 'X[4] <= -0.679\ngini = 0.455\nsamples = 49\
nvalue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 2
7, 50]'),
 Text(2072.5714285714284, 271.7999999999997, 'gini = 0.495\nsamples = 26\nval
ue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 23, 1
9]'),
 Text(2391.428571428571, 271.7999999999997, 'gini = 0.202\nsamples = 23\nvalu
e = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 4, 3
1]'),
 Text(3587.142857142857, 1902.6, 'X[2] <= -0.713\ngini = 0.899\nsamples = 111
1\nvalue = [179, 0, 174, 0, 0, 0, 0, 0, 0, 155, 177, 0\n202, 196, 171, 0, 20
2, 0, 172, 0, 162, 0, 0, 1]'),
 Text(3188.5714285714284, 1359.0, 'X[0] <= -1.96\ngini = 0.5\nsamples = 214\n
value = [0, 0, 173, 0, 0, 0, 0, 0, 0, 0, 176, 0, 0, 0\n0, 0, 0, 0, 0, 0,
0, 0, 0]'),
```

```
 Text(2869.7142857142853, 815.3999999999999, 'X[7] <= -0.167\ngini = 0.372\ns
amples = 142\nvalue = [0, 0, 56, 0, 0, 0, 0, 0, 0, 0, 171, 0, 0, 0\n0, 0, 0,
0, 0, 0, 0, 0, 0, 0]'),
 Text(2710.285714285714, 271.7999999999997, 'gini = 0.0\nsamples = 105\nvalue
= [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 171, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0]'),
 Text(3029.142857142857, 271.7999999999997, 'gini = 0.0\nsamples = 37\nvalue
= [0, 0, 56, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0]'),
 Text(3507.428571428571, 815.3999999999999, 'X[0] <= -1.953\ngini = 0.079\nsa
mples = 72\nvalue = [0, 0, 117, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0\n0, 0, 0, 0,
0, 0, 0, 0, 0, 0]'),
 Text(3347.9999999999995, 271.7999999999997, 'gini = 0.198\nsamples = 24\nval
ue = [0, 0, 40, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0]'),
 Text(3666.8571428571427, 271.7999999999997, 'gini = 0.0\nsamples = 48\nvalue
= [0, 0, 77, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0]'),
 Text(3985.7142857142853, 1359.0, 'X[3] <= -1.143\ngini = 0.874\nsamples = 89
7\nvalue = [179, 0, 1, 0, 0, 0, 0, 0, 0, 155, 1, 0, 202\n196, 171, 0, 202, 0,
172, 0, 162, 0, 0, 1]'),
 Text(3826.2857142857138, 815.3999999999999, 'gini = 0.0\nsamples = 113\nvalu
e = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 172, 0, 0, 0, 0,
0]'),
 Text(4145.142857142857, 815.3999999999999, 'X[6] <= 0.32\ngini = 0.856\nsamp
les = 784\nvalue = [179, 0, 1, 0, 0, 0, 0, 0, 0, 155, 1, 0, 202\n196, 171, 0,
202, 0, 0, 0, 162, 0, 0, 1]'),
 Text(3985.7142857142853, 271.7999999999997, 'gini = 0.832\nsamples = 671\nva
lue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 155, 1, 0, 202\n194, 171, 0, 202, 0, 0, 0,
162, 0, 0, 1]'),
 Text(4304.571428571428, 271.7999999999997, 'gini = 0.033\nsamples = 113\nval
ue = [179, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2\n0, 0, 0, 0, 0, 0, 0, 0, 0,
0]')]
```



From this observation I had observe that the ELASTICNET is a highest accuracy of 0.9999999957832388

In [ ]: