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 [664]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C10_air\madrid_2016.csv")
a
```

Out[664]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOl
0	2016-11-01 01:00:00	NaN	0.7	NaN	NaN	153.0	77.0	NaN	NaN	NaN	7.0	NaN	NaN
1	2016-11-01 01:00:00	3.1	1.1	2.0	0.53	260.0	144.0	4.0	46.0	24.0	18.0	2.44	14.4
2	2016-11-01 01:00:00	5.9	NaN	7.5	NaN	297.0	139.0	NaN	NaN	NaN	NaN	NaN	26.(
3	2016-11-01 01:00:00	NaN	1.0	NaN	NaN	154.0	113.0	2.0	NaN	NaN	NaN	NaN	NaN
4	2016-11-01 01:00:00	NaN	NaN	NaN	NaN	275.0	127.0	2.0	NaN	NaN	18.0	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
209491	2016-07-01 00:00:00	NaN	0.2	NaN	NaN	2.0	29.0	73.0	NaN	NaN	NaN	NaN	NaN
209492	2016-07-01 00:00:00	NaN	0.3	NaN	NaN	1.0	29.0	NaN	36.0	NaN	5.0	NaN	NaN
209493	2016-07-01 00:00:00	NaN	NaN	NaN	NaN	1.0	19.0	71.0	NaN	NaN	NaN	NaN	NaN
209494	2016-07-01 00:00:00	NaN	NaN	NaN	NaN	6.0	17.0	85.0	NaN	NaN	NaN	NaN	NaN
209495	2016-07-01 00:00:00	NaN	NaN	NaN	NaN	2.0	46.0	61.0	34.0	NaN	NaN	NaN	NaN

209496 rows × 14 columns

In [665]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209496 entries, 0 to 209495
Data columns (total 14 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     209496 non-null  object 
 1   BEN      50755 non-null   float64
 2   CO       85999 non-null   float64
 3   EBE      50335 non-null   float64
 4   NMHC     25970 non-null   float64
 5   NO       208614 non-null  float64
 6   NO_2     208614 non-null  float64
 7   O_3      121197 non-null  float64
 8   PM10     102892 non-null  float64
 9   PM25     52165 non-null   float64
 10  SO_2     86023 non-null   float64
 11  TCH      25970 non-null   float64
 12  TOL      50662 non-null   float64
 13  station  209496 non-null  int64  
dtypes: float64(12), int64(1), object(1)
memory usage: 22.4+ MB
```

In [666]:

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

Out[666]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH
0	2016-11-01 01:00:00	104.0	0.7	104.0	104.00	153.0	77.0	104.0	104.0	104.0	7.0	104.0(
1	2016-11-01 01:00:00	3.1	1.1	2.0	0.53	260.0	144.0	4.0	46.0	24.0	18.0	2.44
2	2016-11-01 01:00:00	5.9	104.0	7.5	104.00	297.0	139.0	104.0	104.0	104.0	104.0	104.0(
3	2016-11-01 01:00:00	104.0	1.0	104.0	104.00	154.0	113.0	2.0	104.0	104.0	104.0	104.0(
4	2016-11-01 01:00:00	104.0	104.0	104.0	104.00	275.0	127.0	2.0	104.0	104.0	18.0	104.0(
...	...	...	...	...	...	...	...	...	...	...	...	...
209491	2016-07-01 00:00:00	104.0	0.2	104.0	104.00	2.0	29.0	73.0	104.0	104.0	104.0	104.0(
209492	2016-07-01 00:00:00	104.0	0.3	104.0	104.00	1.0	29.0	104.0	36.0	104.0	5.0	104.0(
209493	2016-07-01 00:00:00	104.0	104.0	104.0	104.00	1.0	19.0	71.0	104.0	104.0	104.0	104.0(
209494	2016-07-01 00:00:00	104.0	104.0	104.0	104.00	6.0	17.0	85.0	104.0	104.0	104.0	104.0(
209495	2016-07-01 00:00:00	104.0	104.0	104.0	104.00	2.0	46.0	61.0	34.0	104.0	104.0	104.0(

209496 rows × 14 columns

In [667]:

```
b.columns
```

Out[667]:

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

In [668]:

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

Out[668]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TC
0	2016-11-01 01:00:00	104.0	0.7	104.0	104.00	153.0	77.0	104.0	104.0	104.0	7.0	104.00	104
1	2016-11-01 01:00:00	3.1	1.1	2.0	0.53	260.0	144.0	4.0	46.0	24.0	18.0	2.44	14
2	2016-11-01 01:00:00	5.9	104.0	7.5	104.00	297.0	139.0	104.0	104.0	104.0	104.0	104.00	26
3	2016-11-01 01:00:00	104.0	1.0	104.0	104.00	154.0	113.0	2.0	104.0	104.0	104.0	104.00	104
4	2016-11-01 01:00:00	104.0	104.0	104.0	104.00	275.0	127.0	2.0	104.0	104.0	18.0	104.00	104
5	2016-11-01 01:00:00	0.9	0.5	0.5	104.00	66.0	82.0	1.0	27.0	104.0	8.0	104.00	6
6	2016-11-01 01:00:00	0.7	0.8	0.4	0.13	57.0	66.0	3.0	23.0	15.0	4.0	1.35	5
7	2016-11-01 01:00:00	104.0	104.0	104.0	104.00	52.0	78.0	1.0	104.0	104.0	104.0	104.00	104
8	2016-11-01 01:00:00	104.0	1.2	104.0	104.00	205.0	85.0	6.0	104.0	104.0	104.0	104.00	104
9	2016-11-01 01:00:00	104.0	0.7	104.0	104.00	114.0	91.0	104.0	37.0	104.0	6.0	104.00	104

In [669]:

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

Out[669]:

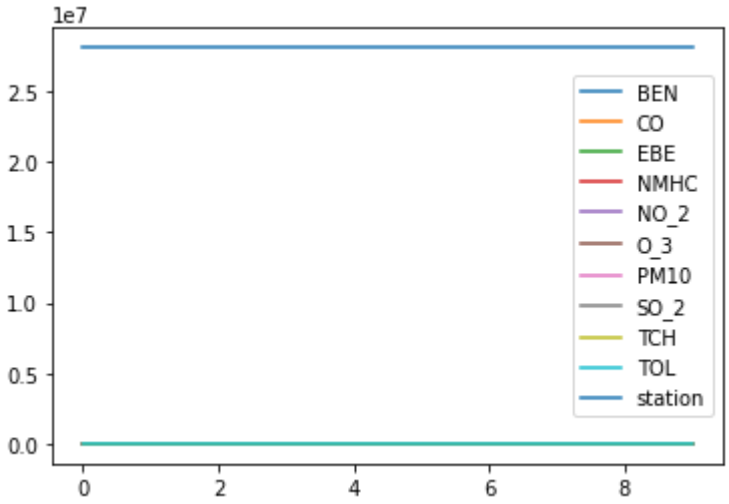
	BEN	CO	EBE	NMHC	NO_2	O_3	PM10	SO_2	TCH	TOL	station
0	104.0	0.7	104.0	104.00	77.0	104.0	104.0	7.0	104.00	104.0	28079004
1	3.1	1.1	2.0	0.53	144.0	4.0	46.0	18.0	2.44	14.4	28079008
2	5.9	104.0	7.5	104.00	139.0	104.0	104.0	104.0	104.00	26.0	28079011
3	104.0	1.0	104.0	104.00	113.0	2.0	104.0	104.0	104.00	104.0	28079016
4	104.0	104.0	104.0	104.00	127.0	2.0	104.0	18.0	104.00	104.0	28079017
5	0.9	0.5	0.5	104.00	82.0	1.0	27.0	8.0	104.00	6.0	28079018
6	0.7	0.8	0.4	0.13	66.0	3.0	23.0	4.0	1.35	5.0	28079024
7	104.0	104.0	104.0	104.00	78.0	1.0	104.0	104.0	104.00	104.0	28079027
8	104.0	1.2	104.0	104.00	85.0	6.0	104.0	104.0	104.00	104.0	28079035
9	104.0	0.7	104.0	104.00	91.0	104.0	37.0	6.0	104.00	104.0	28079036

In [670]:

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

Out[670]:

```
<AxesSubplot:>
```

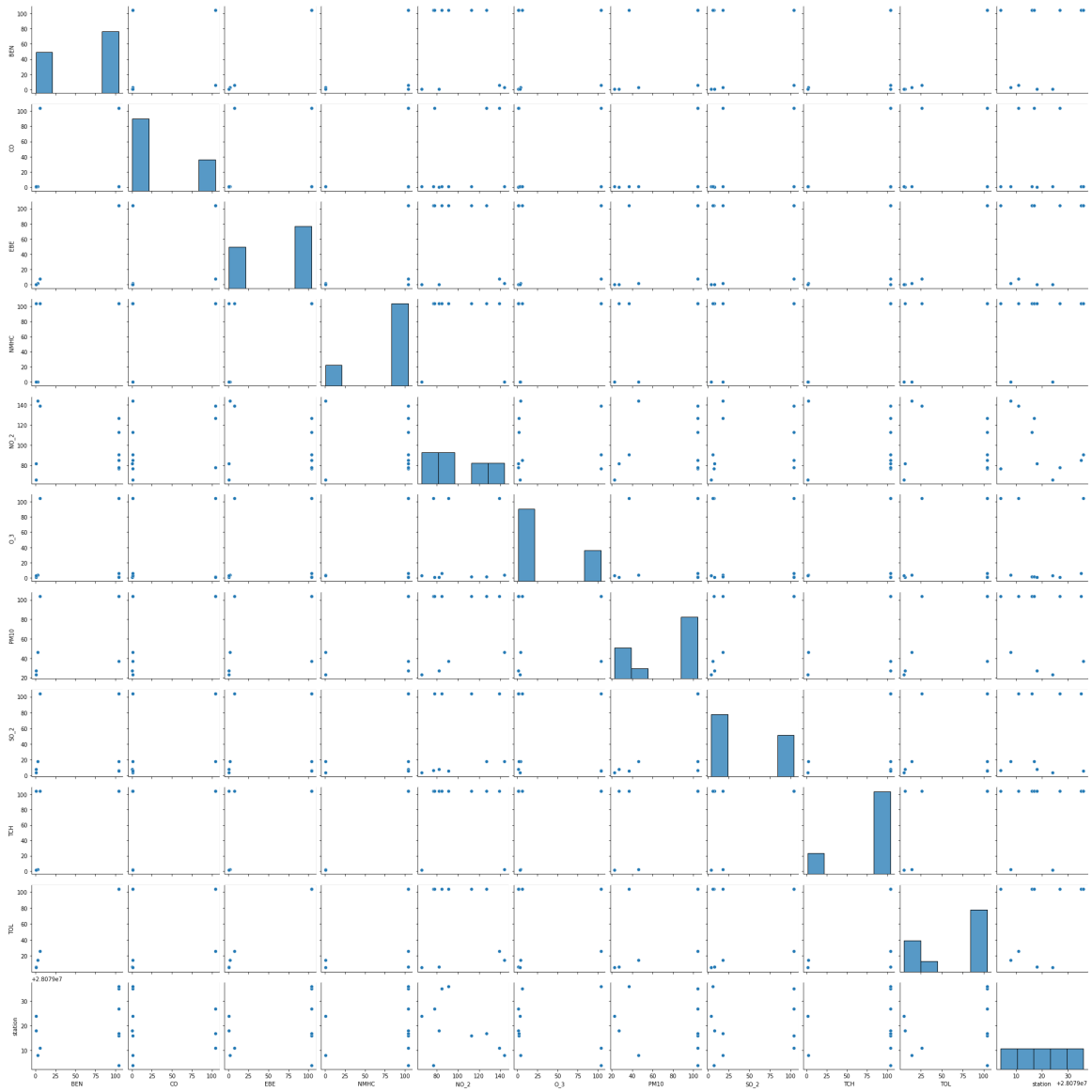


In [671]:

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

Out[671]:

```
<seaborn.axisgrid.PairGrid at 0x1188f7f0460>
```



In [672]:

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

In [673]:

```
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 [674]:

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

Out[674]:

```
LinearRegression()
```

In [675]:

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

```
0.9442409819765203
```

In [676]:

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

Out[676]:

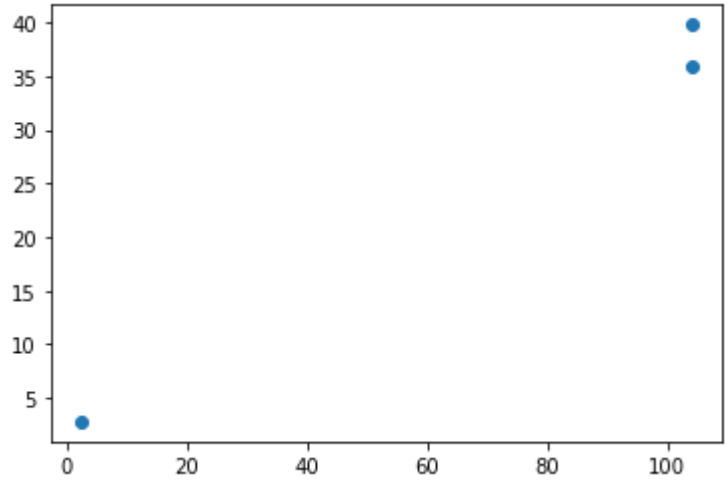
	Co-efficient
BEN	3.293822e-01
CO	-8.443878e-17
EBE	3.303388e-01
NMHC	3.311997e-01
NO_2	-1.875982e-16

In [677]:

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

Out[677]:

```
<matplotlib.collections.PathCollection at 0x1189b963f40>
```



In [678]:

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

```
-0.2746461947031644
```

In [679]:

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

In [680]:

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

Out[680]:

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

In [681]:

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

Out[681]:

```
-0.27334724629485607
```

In [682]:

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

Out[682]:

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

In [683]:

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

Out[683]:

```
-1.8708432060948388
```

In [684]:

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

Out[684]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH
0	2016-11-01 01:00:00	104.0	0.7	104.0	104.00	153.0	77.0	104.0	104.0	104.0	7.0	104.00
1	2016-11-01 01:00:00	3.1	1.1	2.0	0.53	260.0	144.0	4.0	46.0	24.0	18.0	2.44
2	2016-11-01 01:00:00	5.9	104.0	7.5	104.00	297.0	139.0	104.0	104.0	104.0	104.0	104.00
3	2016-11-01 01:00:00	104.0	1.0	104.0	104.00	154.0	113.0	2.0	104.0	104.0	104.0	104.00
4	2016-11-01 01:00:00	104.0	104.0	104.0	104.00	275.0	127.0	2.0	104.0	104.0	18.0	104.00
...	...	...	...	...	...	...	...	...	...	...	...	...
6995	2016-11-13 04:00:00	104.0	0.7	104.0	104.00	96.0	71.0	5.0	104.0	104.0	104.0	104.00
6996	2016-11-13 04:00:00	104.0	104.0	104.0	104.00	45.0	70.0	104.0	26.0	104.0	9.0	104.00
6997	2016-11-13 04:00:00	104.0	104.0	104.0	104.00	87.0	70.0	104.0	28.0	23.0	104.0	104.00
6998	2016-11-13 04:00:00	104.0	104.0	104.0	104.00	66.0	59.0	104.0	33.0	26.0	104.0	104.00
6999	2016-11-13 04:00:00	104.0	104.0	104.0	104.00	98.0	53.0	1.0	104.0	104.0	104.0	104.00

7000 rows × 14 columns

In [685]:

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

In [686]:

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

In [687]:

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

In [688]:

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

Out[688]:

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

In [689]:

```
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 [690]:

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

In [691]:

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

```
[28079059]
```

In [692]:

```
logr.classes_
```

Out[692]:

```
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 [693]:

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

Out[693]:

```
0.0
```

In [694]:

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

Out[694]:

```
0.0
```

In [695]:

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

Out[695]:

```
0.950952380952381
```

In [696]:

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

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_coordinat
e_descent.py:530: ConvergenceWarning: Objective did not converge. You migh
t want to increase the number of iterations. Duality gap: 1.71616200636469
98, tolerance: 0.9031733567233087
  model = cd_fast.enet_coordinate_descent(
```

Out[696]:

```
ElasticNet()
```

In [697]:

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

```
[0.68219062 0.         0.2599273  0.05015844 0.        ]
```

In [698]:

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

```
0.7973611846029485
```

In [699]:

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

```
-1.6065303183198552
```

In [700]:

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

Out[700]:

```
RandomForestClassifier()
```

In [701]:

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

In [702]:

```
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[702]:

```
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 [703]:

```
grid_search.best_score_
```

Out[703]:

```
0.9910204081632653
```

In [704]:

```
rfc_best=grid_search.best_estimator_
```

In [705]:

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

Out[705]:

```
[Text(1916.6086956521738, 2491.5, 'X[10] <= -1.376\ngini = 0.958\nsamples
= 3098\nvalue = [194, 200, 243, 202, 203, 174, 243, 221, 218, 212\n207, 18
4, 204, 199, 187, 193, 193, 203, 198, 209\n199, 221, 212, 181]'),
 Text(970.4347826086956, 2038.5, 'X[7] <= -0.0\ngini = 0.663\nsamples = 38
3\nvalue = [194, 200, 243, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0]'),
 Text(388.17391304347825, 1585.5, 'X[7] <= -1.205\ngini = 0.5\nsamples = 2
40\nvalue = [194, 197, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0]'),
 Text(194.08695652173913, 1132.5, 'gini = 0.0\nsamples = 105\nvalue = [17
1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(582.2608695652174, 1132.5, 'X[6] <= -0.429\ngini = 0.187\nsamples =
135\nvalue = [23, 197, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0]'),
 Text(388.17391304347825, 679.5, 'gini = 0.0\nsamples = 110\nvalue = [0, 1
76, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(776.3478260869565, 679.5, 'gini = 0.499\nsamples = 25\nvalue = [23,
21, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(1552.695652173913, 1585.5, 'X[2] <= -1.744\ngini = 0.024\nsamples =
143\nvalue = [0, 3, 243, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0]'),
 Text(1358.608695652174, 1132.5, 'X[4] <= -0.584\ngini = 0.059\nsamples =
55\nvalue = [0, 3, 95, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0]'),
 Tex[illegible]4.5217391304348[illegible].5, 'gini = 0.0\nsamples = [illegible]value = [0, 0,
51, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(1552.695652173913, 679.5, 'gini = 0.12\nsamples = 25\nvalue = [0, 3,
44, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(1746.782608695652, 1132.5, 'gini = 0.0\nsamples = 88\nvalue = [0, 0,
148, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(2862.782608695652, 2038.5, 'X[3] <= -1.132\ngini = 0.952\nsamples =
2715\nvalue = [0, 0, 0, 202, 203, 174, [illegible]21, 218, [illegible]07\n184, 204,
199, 187, 193, 193, 203, 198, 209, 199\n221, 212, 181]'),
 Text(2329.0434782608695, 1585.5, 'X[1] <= -0.166\ngini = 0.495\nsamples =
274\nvalue = [0, 0, 0, 0, 0, 0, 243, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 198,
0, 0, 0, 0, 0]'),
 Text(2134.9565217391305, 1132.5, 'gini = 0.0\nsamples = 141\nvalue = [0,
0, 0, 0, 0, 0, 243, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(2523.1304347826085, 1132.5, 'gini = 0.0\nsamples = 133\nvalue = [0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 198, 0, 0, 0, 0, 0]'),
 Text(3396.5217391304345, 1585.5, 'X[10] <= 1.123\ngini = 0.947\nsamples =
2441\nvalue = [0, 0, 0, 202, 203, 174, 0, 221, 218, 212, 207\n184, 204, 19
9, 187, 193, 193, 203, 0, 209, 199\n221, 212, 181]'),
 Text(2911.304347826087, 1132.5, 'X[10] <= 0.981\ngini = 0.937\nsamples =
2056\nvalue = [0, 0, 0, 202, 203, 174, 0, 221, 218, 212, 207\n184, 204, 19
9, 187, 193, 193, 203, 0, 209, 199, 0\n0, 0]'),
 Text(2523.1304347826085, 679.5, 'X[7] <= 0.01\ngini = 0.928\nsamples = 17
97\nvalue = [0, 0, 0, 202, 203, 174, 0, 221, 218, 212, 207\n184, 204, 199,
187, 193, 193, 203, 0, 0, 0, 0, 0\n0]')
```