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 [538]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C10_air\madrid_2013.csv")
a
```

Out[538]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TOL
0	2013-11-01 01:00:00	NaN	0.6	NaN	NaN	135.0	74.0	NaN	NaN	NaN	7.0	NaN	NaN
1	2013-11-01 01:00:00	1.5	0.5	1.3	NaN	71.0	83.0	2.0	23.0	16.0	12.0	NaN	8.
2	2013-11-01 01:00:00	3.9	NaN	2.8	NaN	49.0	70.0	NaN	NaN	NaN	NaN	NaN	9.
3	2013-11-01 01:00:00	NaN	0.5	NaN	NaN	82.0	87.0	3.0	NaN	NaN	NaN	NaN	NaN
4	2013-11-01 01:00:00	NaN	NaN	NaN	NaN	242.0	111.0	2.0	NaN	NaN	12.0	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	..
209875	2013-03-01 00:00:00	NaN	0.4	NaN	NaN	8.0	39.0	52.0	NaN	NaN	NaN	NaN	NaN
209876	2013-03-01 00:00:00	NaN	0.4	NaN	NaN	1.0	11.0	NaN	6.0	NaN	2.0	NaN	NaN
209877	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	4.0	75.0	NaN	NaN	NaN	NaN	NaN
209878	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	2.0	11.0	52.0	NaN	NaN	NaN	NaN	NaN
209879	2013-03-01 00:00:00	NaN	NaN	NaN	NaN	1.0	10.0	75.0	3.0	NaN	NaN	NaN	NaN

209880 rows × 14 columns

In [539]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209880 entries, 0 to 209879
Data columns (total 14 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     209880 non-null  object 
 1   BEN      50462 non-null   float64
 2   CO       87018 non-null   float64
 3   EBE      50463 non-null   float64
 4   NMHC     25935 non-null   float64
 5   NO       209108 non-null  float64
 6   NO_2     209108 non-null  float64
 7   O_3      121858 non-null  float64
 8   PM10     104339 non-null  float64
 9   PM25     51980 non-null   float64
 10  SO_2     86970 non-null   float64
 11  TCH      25935 non-null   float64
 12  TOL      50317 non-null   float64
 13  station  209880 non-null  int64  
dtypes: float64(12), int64(1), object(1)
memory usage: 22.4+ MB
```

In [540]:

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

Out[540]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH
0	2013-11-01 01:00:00	104.0	0.6	104.0	104.0	135.0	74.0	104.0	104.0	104.0	7.0	104.0
1	2013-11-01 01:00:00	1.5	0.5	1.3	104.0	71.0	83.0	2.0	23.0	16.0	12.0	104.0
2	2013-11-01 01:00:00	3.9	104.0	2.8	104.0	49.0	70.0	104.0	104.0	104.0	104.0	104.0
3	2013-11-01 01:00:00	104.0	0.5	104.0	104.0	82.0	87.0	3.0	104.0	104.0	104.0	104.0
4	2013-11-01 01:00:00	104.0	104.0	104.0	104.0	242.0	111.0	2.0	104.0	104.0	12.0	104.0
...	...	...	...	...	...	...	...	...	...	...	...	...
209875	2013-03-01 00:00:00	104.0	0.4	104.0	104.0	8.0	39.0	52.0	104.0	104.0	104.0	104.0
209876	2013-03-01 00:00:00	104.0	0.4	104.0	104.0	1.0	11.0	104.0	6.0	104.0	2.0	104.0
209877	2013-03-01 00:00:00	104.0	104.0	104.0	104.0	2.0	4.0	75.0	104.0	104.0	104.0	104.0
209878	2013-03-01 00:00:00	104.0	104.0	104.0	104.0	2.0	11.0	52.0	104.0	104.0	104.0	104.0
209879	2013-03-01 00:00:00	104.0	104.0	104.0	104.0	1.0	10.0	75.0	3.0	104.0	104.0	104.0

209880 rows × 14 columns

In [541]:

```
b.columns
```

Out[541]:

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

In [542]:

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

Out[542]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	TC
0	2013-11-01 01:00:00	104.0	0.6	104.0	104.00	135.0	74.0	104.0	104.0	104.0	7.0	104.00	104
1	2013-11-01 01:00:00	1.5	0.5	1.3	104.00	71.0	83.0	2.0	23.0	16.0	12.0	104.00	8
2	2013-11-01 01:00:00	3.9	104.0	2.8	104.00	49.0	70.0	104.0	104.0	104.0	104.0	104.00	9
3	2013-11-01 01:00:00	104.0	0.5	104.0	104.00	82.0	87.0	3.0	104.0	104.0	104.0	104.00	104
4	2013-11-01 01:00:00	104.0	104.0	104.0	104.00	242.0	111.0	2.0	104.0	104.0	12.0	104.00	104
5	2013-11-01 01:00:00	1.0	0.6	0.8	104.00	70.0	70.0	2.0	24.0	104.0	6.0	104.00	5
6	2013-11-01 01:00:00	104.0	0.4	104.0	0.29	51.0	80.0	5.0	23.0	14.0	4.0	1.44	104
7	2013-11-01 01:00:00	104.0	104.0	104.0	0.23	29.0	60.0	4.0	104.0	104.0	104.0	1.51	104
8	2013-11-01 01:00:00	104.0	1.0	104.0	104.00	165.0	107.0	2.0	104.0	104.0	11.0	104.00	104
9	2013-11-01 01:00:00	104.0	0.6	104.0	104.00	63.0	93.0	104.0	11.0	104.0	8.0	104.00	104

In [543]:

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

Out[543]:

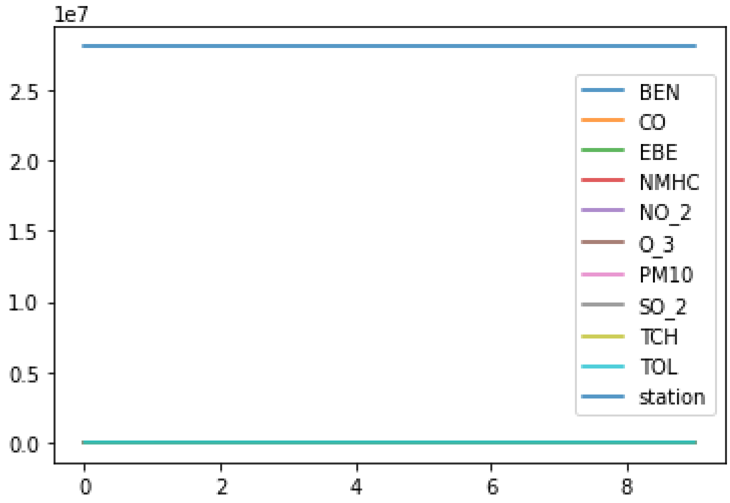
	BEN	CO	EBE	NMHC	NO_2	O_3	PM10	SO_2	TCH	TOL	station
0	104.0	0.6	104.0	104.00	74.0	104.0	104.0	7.0	104.00	104.0	28079004
1	1.5	0.5	1.3	104.00	83.0	2.0	23.0	12.0	104.00	8.3	28079008
2	3.9	104.0	2.8	104.00	70.0	104.0	104.0	104.0	104.00	9.0	28079011
3	104.0	0.5	104.0	104.00	87.0	3.0	104.0	104.0	104.00	104.0	28079016
4	104.0	104.0	104.0	104.00	111.0	2.0	104.0	12.0	104.00	104.0	28079017
5	1.0	0.6	0.8	104.00	70.0	2.0	24.0	6.0	104.00	5.2	28079018
6	104.0	0.4	104.0	0.29	80.0	5.0	23.0	4.0	1.44	104.0	28079024
7	104.0	104.0	104.0	0.23	60.0	4.0	104.0	104.0	1.51	104.0	28079027
8	104.0	1.0	104.0	104.00	107.0	2.0	104.0	11.0	104.00	104.0	28079035
9	104.0	0.6	104.0	104.00	93.0	104.0	11.0	8.0	104.00	104.0	28079036

In [544]:

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

Out[544]:

```
<AxesSubplot:>
```

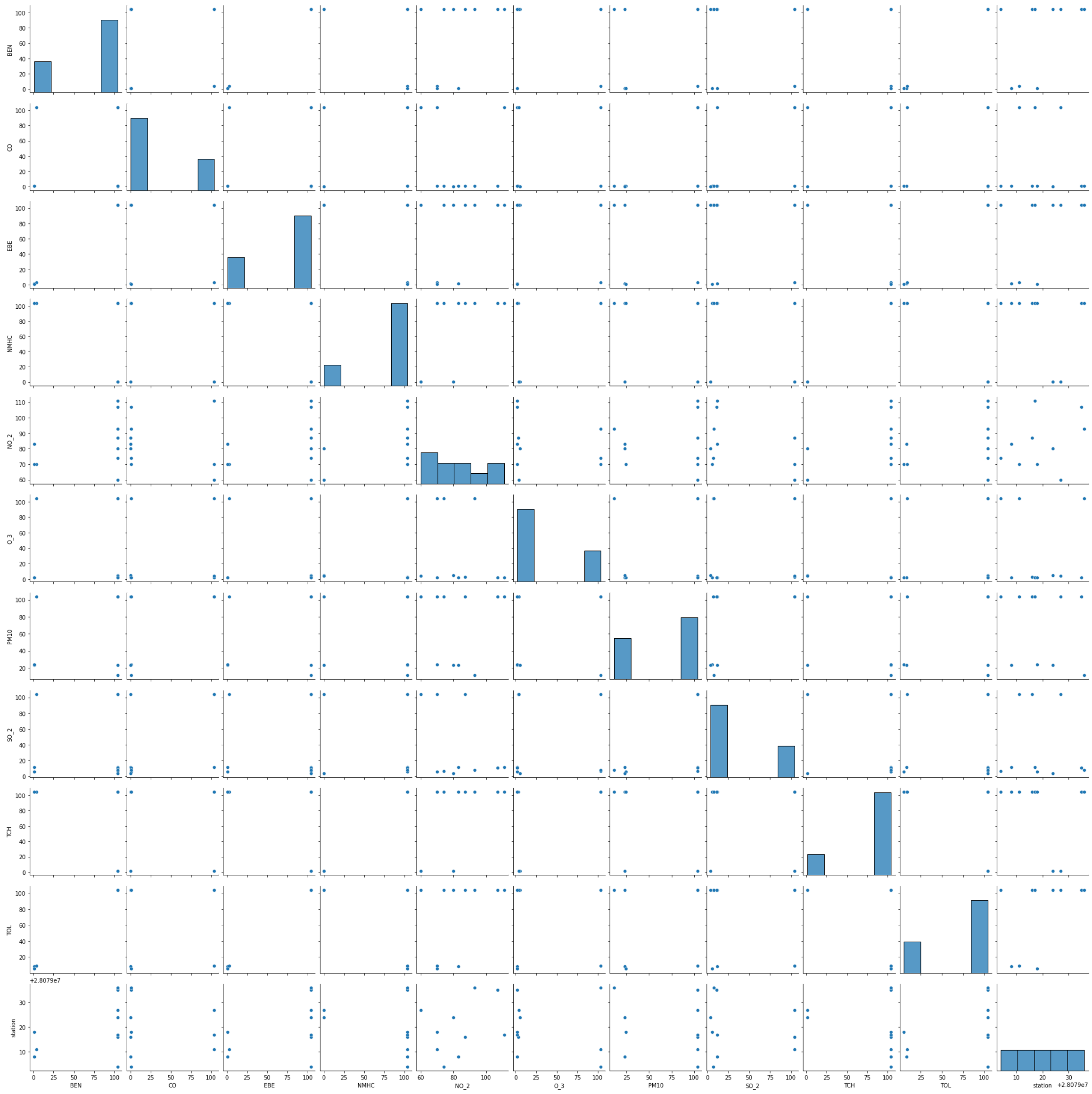


In [545]:

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

Out[545]:

```
<seaborn.axisgrid.PairGrid at 0x1186e23bfa0>
```



In [546]:

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

In [547]:

```
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 [548]:

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

Out[548]:

```
LinearRegression()
```

In [549]:

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

```
1.1532157632747015
```

In [550]:

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

Out[550]:

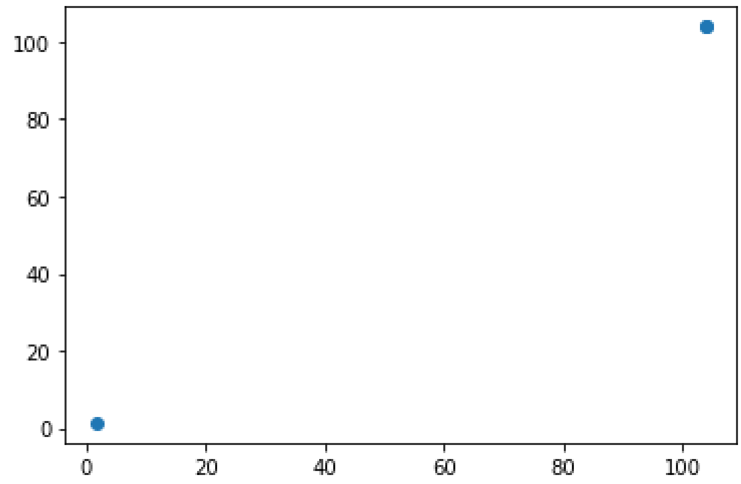
	Co-efficient
BEN	-3.339620e-12
CO	2.854507e-14
EBE	3.332956e-12
NMHC	9.889114e-01
NO_2	-1.138840e-16

In [551]:

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

Out[551]:

```
<matplotlib.collections.PathCollection at 0x118772352e0>
```



In [552]:

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

```
0.999997611321236
```

In [553]:

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

In [554]:

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

Out[554]:

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

In [555]:

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

Out[555]:

```
0.9999995125979597
```

In [556]:

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

Out[556]:

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

In [557]:

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

Out[557]:

```
0.9999538023684293
```

In [558]:

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

Out[558]:

	date	BEN	CO	EBE	NMHC	NO	NO_2	O_3	PM10	PM25	SO_2	TCH	
0	2013-11-01 01:00:00	104.0	0.6	104.0	104.0	135.0	74.0	104.0	104.0	104.0	7.0	104.0	1
1	2013-11-01 01:00:00	1.5	0.5	1.3	104.0	71.0	83.0	2.0	23.0	16.0	12.0	104.0	
2	2013-11-01 01:00:00	3.9	104.0	2.8	104.0	49.0	70.0	104.0	104.0	104.0	104.0	104.0	
3	2013-11-01 01:00:00	104.0	0.5	104.0	104.0	82.0	87.0	3.0	104.0	104.0	104.0	104.0	1
4	2013-11-01 01:00:00	104.0	104.0	104.0	104.0	242.0	111.0	2.0	104.0	104.0	12.0	104.0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	
6995	2013-11-13 04:00:00	104.0	0.2	104.0	104.0	1.0	8.0	40.0	104.0	104.0	104.0	104.0	1
6996	2013-11-13 04:00:00	104.0	104.0	104.0	104.0	1.0	5.0	104.0	3.0	104.0	1.0	104.0	1
6997	2013-11-13 04:00:00	104.0	104.0	104.0	104.0	1.0	6.0	104.0	3.0	2.0	104.0	104.0	1
6998	2013-11-13 04:00:00	104.0	104.0	104.0	104.0	1.0	9.0	104.0	5.0	1.0	104.0	104.0	1
6999	2013-11-13 04:00:00	104.0	104.0	104.0	104.0	1.0	9.0	43.0	104.0	104.0	104.0	104.0	1

7000 rows × 14 columns

In [559]:

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

In [560]:

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

In [561]:

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

In [562]:

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

Out[562]:

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

In [563]:

```
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 [564]:

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

In [565]:

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

```
[28079050]
```

In [566]:

```
logr.classes_
```

Out[566]:

```
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 [567]:

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

Out[567]:

```
0.0
```

In [568]:

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

Out[568]:

```
0.0
```

In [569]:

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

Out[569]:

```
0.9552380952380952
```

In [570]:

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

Out[570]:

```
ElasticNet()
```

In [571]:

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

```
[-0.         0.        -0.         0.9881566  0.       ]
```

In [572]:

```
print(en.intercept_)

```

```
1.2205308629257416
```

In [573]:

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

```
0.9999994119369457
```

In [574]:

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

Out[574]:

```
RandomForestClassifier()
```

In [575]:

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

In [576]:

```
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[576]:

```
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 [577]:

```
grid_search.best_score_
```

Out[577]:

```
0.9979591836734694
```

In [578]:

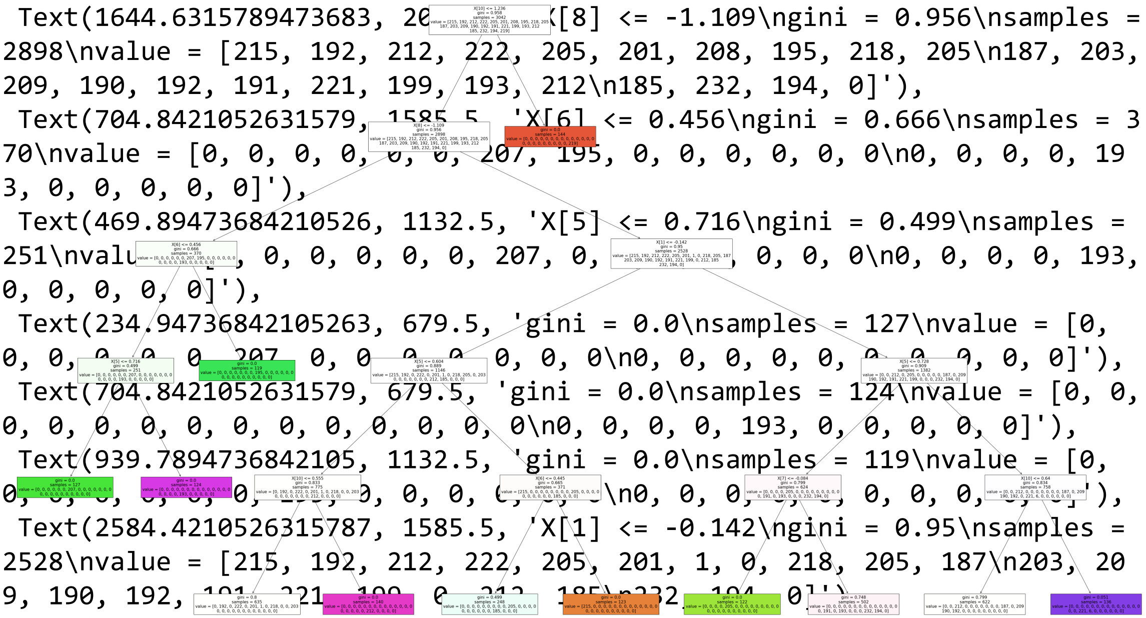
```
rfc_best=grid_search.best_estimator_
```

In [579]:

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

```
Out[579]:
```

```
[Text(1879.578947368421, 2491.5, 'X[10] <= 1.236\ngini = 0.958\nsamples =
3042\nvalue = [215, 192, 212, 222, 205, 201, 208, 195, 218, 205\n187, 203,
209, 190, 192, 191, 221, 199, 193, 212\n185, 232, 194, 219]'),
 Text(1644.6315789473683, 2038.5, 'X[8] <= -1.109\ngini = 0.956\nsamples =
2898\nvalue = [215, 192, 212, 222, 205, 201, 208, 195, 218, 205\n187, 203,
209, 190, 192, 191, 221, 199, 193, 212\n185, 232, 194, 0]'),
 Text(704.8421052631579, 1585.5, 'X[6] <= 0.456\ngini = 0.666\nsamples = 3
70\nvalue = [0, 0, 0, 0, 0, 0, 207, 195, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 19
3, 0, 0, 0, 0, 0]'),
 Text(469.89473684210526, 1132.5, 'X[5] <= 0.716\ngini = 0.499\nsamples =
251\nvalue = [0, 0, 0, 0, 0, 0, 207, 0, 0, 0, 0, 0\n0, 0, 0, 0, 193,
0, 0, 0, 0, 0]'),
 Text(234.94736842105263, 679.5, 'gini = 0.0\nsamples = 127\nvalue = [0,
0, 0, 0, 0, 0, 207, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(704.8421052631579, 679.5, 'gini = 0.0\nsamples = 124\nvalue = [0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 193, 0, 0, 0, 0, 0]'),
 Text(939.7894736842105, 1132.5, 'gini = 0.0\nsamples = 119\nvalue = [0,
0, 0, 0, 0, 0, 0, 195, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(2584.4210526315787, 1585.5, 'X[1] <= -0.142\ngini = 0.95\nsamples =
2528\nvalue = [215, 192, 212, 222, 205, 201, 1, 0, 218, 205, 187\n203, 20
9, 190, 192, 191, 221, 199, 0, 212, 185\n232, 194, 0]'),
 Text(1644.6315789473683, 1132.5, 'X[5] <= 0.604\ngini = 0.889\nsamples =
1146\nvalue = [215, 192, 0, 222, 0, 201, 1, 0, 218, 205, 0, 203\n0, 0, 0,
0, 0, 0, 0, 212, 185, 0, 0, 0]'),
 Text(1174.7368421052631, 679.5, 'X[10] <= 0.555\ngini = 0.833\nsamples =
775\nvalue = [0, 192, 0, 222, 0, 201, 1, 0, 218, 0, 0, 203\n0, 0, 0, 0, 0,
0, 0, 212, 0, 0, 0, 0]'),
 Text(939.7894736842105, 226.5, 'gini = 0.8\nsamples = 635\nvalue = [0, 19
2, 0, 222, 0, 201, 1, 0, 218, 0, 0, 203\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0]'),
 Text(1409.6842105263158, 226.5, 'gini = 0.0\nsamples = 140\nvalue = [0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 212, 0, 0, 0, 0]'),
 Text(2114.5263157894738, 679.5, 'X[6] <= 0.445\ngini = 0.665\nsamples = 3
71\nvalue = [215, 0, 0, 0, 0, 0, 0, 0, 0, 205, 0, 0, 0, 0\n0, 0, 0, 0, 0,
0, 185, 0, 0, 0]')
```



In [ ]