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 [131]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C10_air\madrid_2004.csv")
a
```

Out[131]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	
0	2004-08-01 01:00:00	NaN	0.66	NaN	NaN	NaN	89.550003	118.900002	NaN	40.020000	39
1	2004-08-01 01:00:00	2.66	0.54	2.99	6.08	0.18	51.799999	53.860001	3.28	51.689999	22
2	2004-08-01 01:00:00	NaN	1.02	NaN	NaN	NaN	93.389999	138.600006	NaN	20.860001	49
3	2004-08-01 01:00:00	NaN	0.53	NaN	NaN	NaN	87.290001	105.000000	NaN	36.730000	31
4	2004-08-01 01:00:00	NaN	0.17	NaN	NaN	NaN	34.910000	35.349998	NaN	86.269997	54
...	...	...	...	...	...	...	...	...	...	...	
245491	2004-06-01 00:00:00	0.75	0.21	0.85	1.55	0.07	59.580002	64.389999	0.66	33.029999	30
245492	2004-06-01 00:00:00	2.49	0.75	2.44	4.57	NaN	97.139999	146.899994	2.34	7.740000	37
245493	2004-06-01 00:00:00	NaN	NaN	NaN	NaN	0.13	102.699997	132.600006	NaN	17.809999	22
245494	2004-06-01 00:00:00	NaN	NaN	NaN	NaN	0.09	82.599998	102.599998	NaN	NaN	45
245495	2004-06-01 00:00:00	3.01	0.67	2.78	5.12	0.20	92.550003	141.000000	2.60	11.460000	24

245496 rows × 17 columns

In [132]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 245496 entries, 0 to 245495
Data columns (total 17 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     245496 non-null  object 
 1   BEN      65158 non-null   float64
 2   CO       226043 non-null  float64
 3   EBE      56781 non-null   float64
 4   MXY      39867 non-null   float64
 5   NMHC     107630 non-null  float64
 6   NO_2     243280 non-null  float64
 7   NOx      243283 non-null  float64
 8   OXY      39882 non-null   float64
 9   O_3      233811 non-null  float64
 10  PM10     234655 non-null  float64
 11  PM25     58145 non-null   float64
 12  PXY      39891 non-null   float64
 13  SO_2     243402 non-null  float64
 14  TCH      107650 non-null  float64
 15  TOL      64914 non-null   float64
 16  station  245496 non-null  int64  
dtypes: float64(15), int64(1), object(1)
memory usage: 31.8+ MB
```

In [133]:

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

Out[133]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	
0	2004-08-01 01:00:00	104.00	0.66	104.00	104.00	104.00	89.550003	118.900002	104.00	40.
1	2004-08-01 01:00:00	2.66	0.54	2.99	6.08	0.18	51.799999	53.860001	3.28	51.
2	2004-08-01 01:00:00	104.00	1.02	104.00	104.00	104.00	93.389999	138.600006	104.00	20.
3	2004-08-01 01:00:00	104.00	0.53	104.00	104.00	104.00	87.290001	105.000000	104.00	36.
4	2004-08-01 01:00:00	104.00	0.17	104.00	104.00	104.00	34.910000	35.349998	104.00	86.
...	...	...	...	...	...	...	...	...	...	
245491	2004-06-01 00:00:00	0.75	0.21	0.85	1.55	0.07	59.580002	64.389999	0.66	33.
245492	2004-06-01 00:00:00	2.49	0.75	2.44	4.57	104.00	97.139999	146.899994	2.34	7.
245493	2004-06-01 00:00:00	104.00	104.00	104.00	104.00	0.13	102.699997	132.600006	104.00	17.
245494	2004-06-01 00:00:00	104.00	104.00	104.00	104.00	0.09	82.599998	102.599998	104.00	104.
245495	2004-06-01 00:00:00	3.01	0.67	2.78	5.12	0.20	92.550003	141.000000	2.60	11.

245496 rows × 17 columns

In [134]:

```
b.columns
```

Out[134]:

```
Index(['date', 'BEN', 'CO', 'EBE', 'MXY', 'NMHC', 'NO_2', 'NOx', 'OXY', 'O
_3',
       'PM10', 'PM25', 'PXY', 'SO_2', 'TCH', 'TOL', 'station'],
      dtype='object')
```

In [135]:

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

Out[135]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3
0	2004-08-01 01:00:00	104.00	0.66	104.00	104.00	104.00	89.550003	118.900002	104.00	40.020000
1	2004-08-01 01:00:00	2.66	0.54	2.99	6.08	0.18	51.799999	53.860001	3.28	51.689999
2	2004-08-01 01:00:00	104.00	1.02	104.00	104.00	104.00	93.389999	138.600006	104.00	20.860001
3	2004-08-01 01:00:00	104.00	0.53	104.00	104.00	104.00	87.290001	105.000000	104.00	36.730000
4	2004-08-01 01:00:00	104.00	0.17	104.00	104.00	104.00	34.910000	35.349998	104.00	86.269997
5	2004-08-01 01:00:00	3.24	0.63	5.55	9.72	0.06	103.800003	144.800003	5.04	32.480000
6	2004-08-01 01:00:00	104.00	0.43	104.00	104.00	0.17	54.270000	64.279999	104.00	66.589996
7	2004-08-01 01:00:00	1.41	0.47	2.35	104.00	0.02	71.730003	87.519997	104.00	53.270000
8	2004-08-01 01:00:00	104.00	1.28	104.00	104.00	104.00	147.699997	202.500000	104.00	10.280000
9	2004-08-01 01:00:00	104.00	0.43	104.00	104.00	0.27	54.290001	68.099998	104.00	66.709999

In [136]:

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

Out[136]:

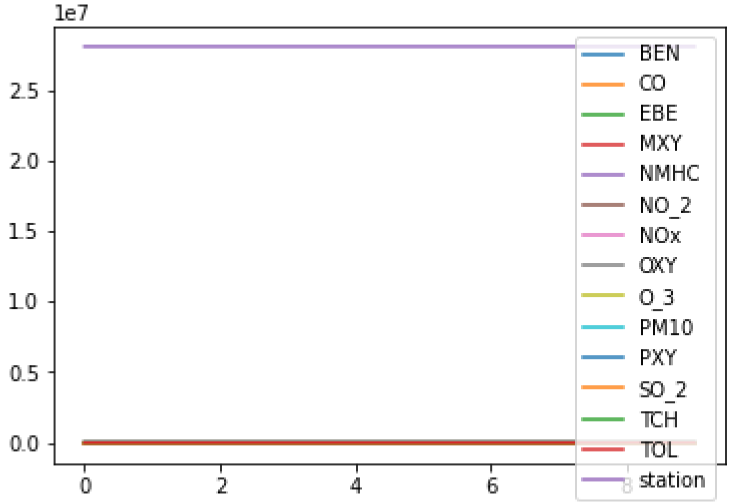
	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3	PM10
0	104.00	0.66	104.00	104.00	104.00	89.550003	118.900002	104.00	40.020000	39.990002
1	2.66	0.54	2.99	6.08	0.18	51.799999	53.860001	3.28	51.689999	22.950001
2	104.00	1.02	104.00	104.00	104.00	93.389999	138.600006	104.00	20.860001	49.480000
3	104.00	0.53	104.00	104.00	104.00	87.290001	105.000000	104.00	36.730000	31.070000
4	104.00	0.17	104.00	104.00	104.00	34.910000	35.349998	104.00	86.269997	54.080002
5	3.24	0.63	5.55	9.72	0.06	103.800003	144.800003	5.04	32.480000	59.110001
6	104.00	0.43	104.00	104.00	0.17	54.270000	64.279999	104.00	66.589996	54.270000
7	1.41	0.47	2.35	104.00	0.02	71.730003	87.519997	104.00	53.270000	45.180000
8	104.00	1.28	104.00	104.00	104.00	147.699997	202.500000	104.00	10.280000	52.430000
9	104.00	0.43	104.00	104.00	0.27	54.290001	68.099998	104.00	66.709999	54.700001

In [137]:

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

Out[137]:

```
<AxesSubplot:>
```

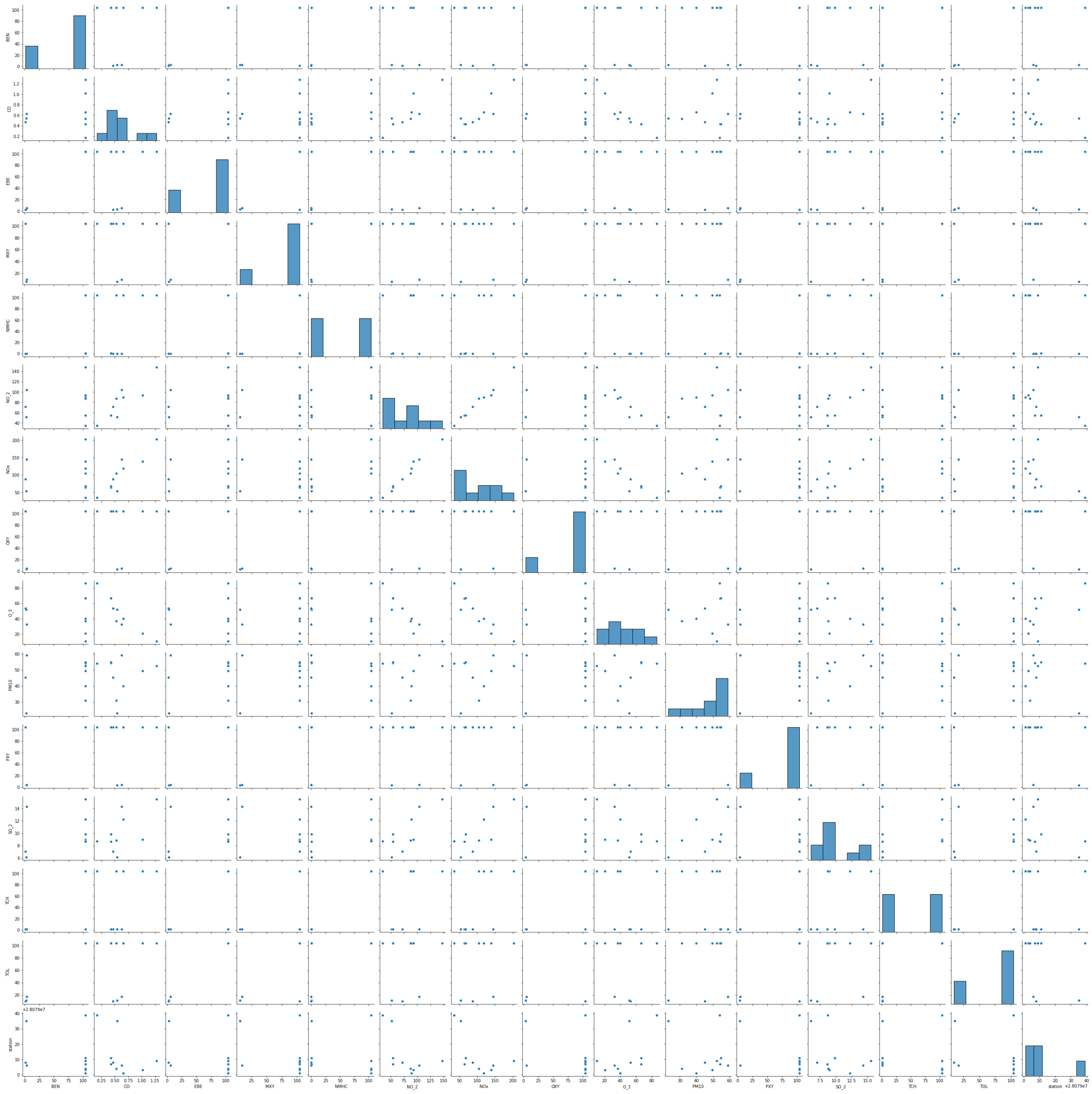


In [138]:

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

Out[138]:

```
<seaborn.axisgrid.PairGrid at 0x117dff22280>
```



In [139]:

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

In [140]:

```
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 [141]:

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

Out[141]:

```
LinearRegression()
```

In [142]:

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

```
1.1722703752459154
```

In [143]:

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

Out[143]:

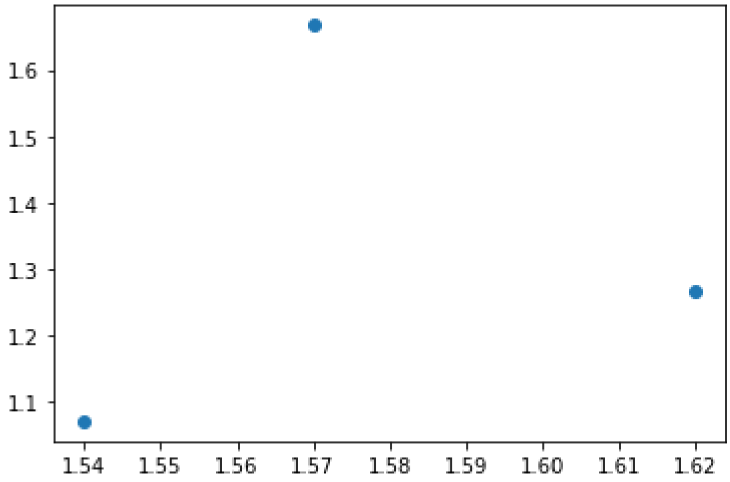
	Co-efficient
BEN	1.010303e-14
CO	7.075394e-01
EBE	1.623701e-15
MXY	0.000000e+00
NMHC	9.880562e-01
NO_2	1.220572e-02
NOx	-1.293821e-02
OXY	0.000000e+00

In [144]:

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

Out[144]:

```
<matplotlib.collections.PathCollection at 0x117e8545850>
```



In [145]:

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

```
-107.29274323648137
```

In [146]:

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

In [147]:

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

Out[147]:

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

In [148]:

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

Out[148]:

```
-24.966205643790936
```

In [149]:

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

Out[149]:

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

In [150]:

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

Out[150]:

```
-21.039777091364407
```

In [151]:

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

Out[151]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3
0	2004-08-01 01:00:00	104.00	0.66	104.00	104.00	104.00	89.550003	118.900002	104.00	40.020000
1	2004-08-01 01:00:00	2.66	0.54	2.99	6.08	0.18	51.799999	53.860001	3.28	51.689999
2	2004-08-01 01:00:00	104.00	1.02	104.00	104.00	104.00	93.389999	138.600006	104.00	20.860001
3	2004-08-01 01:00:00	104.00	0.53	104.00	104.00	104.00	87.290001	105.000000	104.00	36.730000
4	2004-08-01 01:00:00	104.00	0.17	104.00	104.00	104.00	34.910000	35.349998	104.00	86.269997
...	...	...	...	...	...	...	...	...	...	...
6995	2004-08-11 11:00:00	104.00	0.35	104.00	104.00	104.00	38.959999	60.660000	104.00	28.830000
6996	2004-08-11 11:00:00	104.00	0.44	104.00	104.00	104.00	48.400002	99.690002	104.00	24.700001
6997	2004-08-11 11:00:00	0.20	0.20	104.00	104.00	104.00	32.580002	50.669998	104.00	6.940000
6998	2004-08-11 11:00:00	104.00	0.38	104.00	104.00	0.10	54.660000	83.279999	104.00	23.760000
6999	2004-08-11 11:00:00	0.66	0.20	1.03	1.88	0.02	16.709999	22.690001	1.05	50.040001

7000 rows × 17 columns

In [152]:

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

In [153]:

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

In [154]:

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

In [155]:

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

Out[155]:

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

In [156]:

```
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 [157]:

```
i=[[10,20,30,40,50,60,11,22,33,44,55,54,21,78]]
```

In [158]:

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

```
[28079004]
```

In [159]:

```
logr.classes_
```

Out[159]:

```
array([28079001, 28079003, 28079004, 28079006, 28079007, 28079008,
       28079009, 28079011, 28079012, 28079014, 28079015, 28079016,
       28079017, 28079018, 28079019, 28079021, 28079022, 28079023,
       28079024, 28079025, 28079026, 28079027, 28079035, 28079036,
       28079038, 28079039, 28079040, 28079099], dtype=int64)
```

In [160]:

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

Out[160]:

```
2.456148385298682e-93
```

In [161]:

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

Out[161]:

```
4.916459799150848e-280
```

In [162]:

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

Out[162]:

```
0.5738095238095238
```

In [163]:

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

Out[163]:

```
ElasticNet()
```

In [164]:

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

```
[0.        0.        0.        0.        0.9872135 0.        0.
 0.       ]
```

In [165]:

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

```
1.3163720572500779
```

In [166]:

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

```
-33.70843896726327
```

In [167]:

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

Out[167]:

```
RandomForestClassifier()
```

In [168]:

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

In [169]:

```
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[169]:

```
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 [170]:

```
grid_search.best_score_
```

Out[170]:

```
0.6318367346938776
```

In [171]:

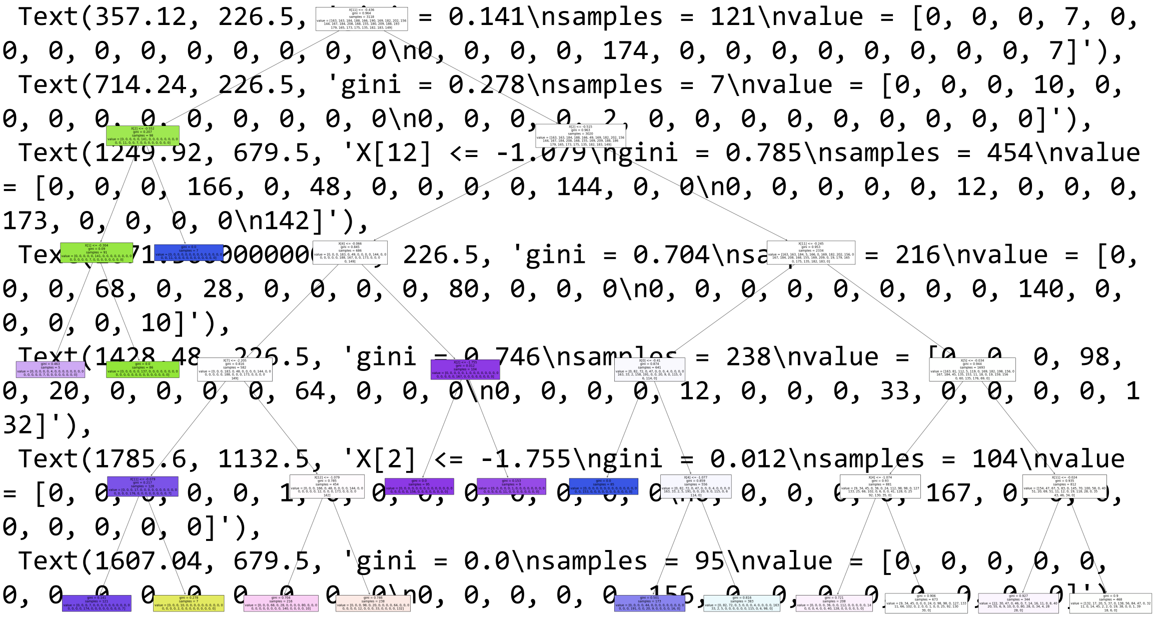
```
rfc_best=grid_search.best_estimator_
```

In [172]:

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

```
Out[172]:
```

```
[Text(1383.84, 2491.5, 'X[11] <= -0.436\ngini = 0.964\nsamples = 3118\nval
ue = [163, 163, 184, 188, 166, 190, 169, 182, 202, 156\n144, 167, 184, 20
8, 168, 155, 180, 209, 188, 193\n179, 165, 173, 175, 135, 182, 183, 14
9]'),
 Text(535.6800000000001, 2038.5, 'X[2] <= -0.552\ngini = 0.207\nsamples =
98\nvalue = [0, 0, 0, 0, 0, 141, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 11, 0, 0,
7, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(357.12, 1585.5, 'X[1] <= -0.304\ngini = 0.09\nsamples = 91\nvalue =
[0, 0, 0, 0, 0, 141, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 7, 0, 0, 0, 0,
0, 0, 0, 0]'),
 Text(178.56, 1132.5, 'gini = 0.463\nsamples = 5\nvalue = [0, 0, 0, 0, 0,
4, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(535.6800000000001, 1132.5, 'gini = 0.0\nsamples = 86\nvalue = [0, 0,
0, 0, 0, 137, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0]'),
 Text(714.24, 1585.5, 'gini = 0.0\nsamples = 7\nvalue = [0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 11, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(2232.0, 2038.5, 'X[2] <= -0.515\ngini = 0.963\nsamples = 3020\nvalue
= [163, 163, 184, 188, 166, 49, 169, 182, 202, 156\n144, 167, 184, 208, 16
8, 155, 169, 209, 188, 186\n179, 165, 173, 175, 135, 182, 183, 149]'),
 Text(1339.2, 1585.5, 'X[4] <= -0.066\ngini = 0.845\nsamples = 686\nvalue
= [0, 0, 0, 183, 0, 49, 0, 0, 0, 0, 144, 0, 0\n0, 0, 0, 0, 0, 188, 167, 0,
0, 173, 0, 0, 0\n0, 149]'),
 Text(892.8, 1132.5, 'X[7] <= -2.205\ngini = 0.816\nsamples = 582\nvalue =
[0, 0, 0, 183, 0, 48, 0, 0, 0, 0, 144, 0, 0\n0, 0, 0, 0, 0, 188, 0, 0, 0,
173, 0, 0, 0, 0\n149]'),
 Text(535.6800000000001, 679.5, 'X[11] <= -0.079\ngini = 0.217\nsamples =
128\nvalue = [0, 0, 0, 17, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 176,
0, 0, 0, 0, 0, 0, 0, 0, 7]'),
 Text(357.12, 226.5,          = 0.141\nsamples = 121\nvalue = [0, 0, 0, 7, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 174, 0, 0, 0, 0, 0, 0, 0, 0, 7]'),
 Text(714.24, 226.5, 'gini = 0.278\nsamples = 7\nvalue = [0, 0, 0, 10, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(1249.92, 679.5, 'X[12] <= -1.079\ngini = 0.785\nsamples = 454\nvalue
= [0, 0, 0, 166, 0, 48, 0, 0, 0, 0, 144, 0, 0\n0, 0, 0, 0, 0, 12, 0, 0, 0,
173, 0, 0, 0, 0\n142]'),
 Tex   71.36000000000       226.5, 'gini = 0.704\nsa           = 216\nvalue = [0,
0, 0, 68, 0, 28, 0, 0, 0, 0, 80, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0, 0, 140, 0,
0, 0, 0, 10]'),
 Text(1428.48, 226.5, 'gini = 0.746\nsamples = 238\nvalue = [0, 0, 0, 98,
0, 20, 0, 0, 0, 0, 64, 0, 0, 0\n0, 0, 0, 0, 12, 0, 0, 0, 33, 0, 0, 0, 0, 1
32]'),
 Text(1785.6, 1132.5, 'X[2] <= -1.755\ngini = 0.012\nsamples = 104\nvalue
= [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 167, 0, 0, 0, 0,
0, 0, 0, 0, 0]'),
 Text(1607.04, 679.5, 'gini = 0.0\nsamples = 95\nvalue = [0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 156, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(1964.16, 679.5, 'gini = 0.153\nsamples = 9\nvalue = [0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 11, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(3124.8, 1585.5, 'X[11] <= -0.245\ngini = 0.953\nsamples = 2334\nvalu
e = [163, 163, 184, 5, 166, 0, 169, 182, 202, 156, 0\n167, 184, 208, 168,
155, 169, 209, 0, 19, 179, 165\n0, 175, 135, 182, 183, 0]'),
 Text(2499.84, 1132.5, 'X[0] <= -0.41\ngini = 0.874\nsamples = 641\nvalue
= [0, 82, 72, 0, 47, 0, 0, 0, 4, 0, 0, 0, 0\n163, 33, 2, 158, 191, 0, 0, 2
0, 9, 0, 115, 0\n6, 114, 0]'),
 Text(2321.28, 679.5, 'gini = 0.0\nsamples = 85\nvalue = [0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 153, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]'),
 Text(2678.4, 679.5, 'X[4] <= -1.077\ngini = 0.859\nsamples = 556\nvalue =
[0, 82, 72, 0, 47, 0, 0, 0, 4, 0, 0, 0, 0\n163, 33, 2, 5, 191, 0, 0, 20,
9, 0, 115, 0, 6\n114, 0]'),
 Text(2499.84, 226.5, 'gini = 0.501\nsamples = 173\nvalue = [0, 0, 0, 0, 4
```



In [ ]: