In [87]:

```
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 [88]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\C10_air\madrid_2002.csv")
a
```

Out[88]:

| | date | BEN | CO | EBE | MXY | NMHC | NO_2 | NOx | OXY | O_3 | PI |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2002-04-01 01:00:00 | NaN | 1.39 | NaN | NaN | NaN | 145.100006 | 352.100006 | NaN | 6.54 | 41.990 |
| **1** | 2002-04-01 01:00:00 | 1.93 | 0.71 | 2.33 | 6.20 | 0.15 | 98.150002 | 153.399994 | 2.67 | 6.85 | 20.980 |
| **2** | 2002-04-01 01:00:00 | NaN | 0.80 | NaN | NaN | NaN | 103.699997 | 134.000000 | NaN | 13.01 | 28.440 |
| **3** | 2002-04-01 01:00:00 | NaN | 1.61 | NaN | NaN | NaN | 97.599998 | 268.000000 | NaN | 5.12 | 42.180 |
| **4** | 2002-04-01 01:00:00 | NaN | 1.90 | NaN | NaN | NaN | 92.089996 | 237.199997 | NaN | 7.28 | 76.330 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **217291** | 2002-11-01 00:00:00 | 4.16 | 1.14 | NaN | NaN | NaN | 81.080002 | 265.700012 | NaN | 7.21 | 36.750 |
| **217292** | 2002-11-01 00:00:00 | 3.67 | 1.73 | 2.89 | NaN | 0.38 | 113.900002 | 373.100006 | NaN | 5.66 | 63.389 |
| **217293** | 2002-11-01 00:00:00 | 1.37 | 0.58 | 1.17 | 2.37 | 0.15 | 65.389999 | 107.699997 | 1.30 | 9.11 | 9.640 |
| **217294** | 2002-11-01 00:00:00 | 4.51 | 0.91 | 4.83 | 10.99 | NaN | 149.800003 | 202.199997 | 1.00 | 5.75 | N |
| **217295** | 2002-11-01 00:00:00 | 3.11 | 1.17 | 3.00 | 7.77 | 0.26 | 80.110001 | 180.300003 | 2.25 | 7.38 | 29.240 |

217296 rows × 16 columns

In [89]:

```
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 217296 entries, 0 to 217295
Data columns (total 16 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     217296 non-null  object 
 1   BEN      66747 non-null   float64
 2   CO       216637 non-null  float64
 3   EBE      58547 non-null   float64
 4   MXY      41255 non-null   float64
 5   NMHC     87045 non-null   float64
 6   NO_2     216439 non-null  float64
 7   NOx      216439 non-null  float64
 8   OXY      41314 non-null   float64
 9   O_3      216726 non-null  float64
 10  PM10     209113 non-null  float64
 11  PXY      41256 non-null   float64
 12  SO_2     216507 non-null  float64
 13  TCH      87115 non-null   float64
 14  TOL      66619 non-null   float64
 15  station  217296 non-null  int64  
dtypes: float64(14), int64(1), object(1)
memory usage: 26.5+ MB
```

In [90]:

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

Out[90]:

| | date | BEN | CO | EBE | MXY | NMHC | NO_2 | NOx | OXY | O_3 |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2002-04-01 01:00:00 | 104.00 | 1.39 | 104.00 | 104.00 | 104.00 | 145.100006 | 352.100006 | 104.00 | 6.54 |
| **1** | 2002-04-01 01:00:00 | 1.93 | 0.71 | 2.33 | 6.20 | 0.15 | 98.150002 | 153.399994 | 2.67 | 6.85 |
| **2** | 2002-04-01 01:00:00 | 104.00 | 0.80 | 104.00 | 104.00 | 104.00 | 103.699997 | 134.000000 | 104.00 | 13.01 |
| **3** | 2002-04-01 01:00:00 | 104.00 | 1.61 | 104.00 | 104.00 | 104.00 | 97.599998 | 268.000000 | 104.00 | 5.12 |
| **4** | 2002-04-01 01:00:00 | 104.00 | 1.90 | 104.00 | 104.00 | 104.00 | 92.089996 | 237.199997 | 104.00 | 7.28 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **217291** | 2002-11-01 00:00:00 | 4.16 | 1.14 | 104.00 | 104.00 | 104.00 | 81.080002 | 265.700012 | 104.00 | 7.21 |
| **217292** | 2002-11-01 00:00:00 | 3.67 | 1.73 | 2.89 | 104.00 | 0.38 | 113.900002 | 373.100006 | 104.00 | 5.66 |
| **217293** | 2002-11-01 00:00:00 | 1.37 | 0.58 | 1.17 | 2.37 | 0.15 | 65.389999 | 107.699997 | 1.30 | 9.11 |
| **217294** | 2002-11-01 00:00:00 | 4.51 | 0.91 | 4.83 | 10.99 | 104.00 | 149.800003 | 202.199997 | 1.00 | 5.75 |
| **217295** | 2002-11-01 00:00:00 | 3.11 | 1.17 | 3.00 | 7.77 | 0.26 | 80.110001 | 180.300003 | 2.25 | 7.38 |

217296 rows × 16 columns

In [91]:

```
b.columns
```

Out[91]:

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

In [92]:

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

Out[92]:

| | date | BEN | CO | EBE | MXY | NMHC | NO_2 | NOx | OXY | O_3 |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2002-04-01 01:00:00 | 104.00 | 1.39 | 104.00 | 104.00 | 104.00 | 145.100006 | 352.100006 | 104.00 | 6.540000 |
| **1** | 2002-04-01 01:00:00 | 1.93 | 0.71 | 2.33 | 6.20 | 0.15 | 98.150002 | 153.399994 | 2.67 | 6.850000 |
| **2** | 2002-04-01 01:00:00 | 104.00 | 0.80 | 104.00 | 104.00 | 104.00 | 103.699997 | 134.000000 | 104.00 | 13.010000 |
| **3** | 2002-04-01 01:00:00 | 104.00 | 1.61 | 104.00 | 104.00 | 104.00 | 97.599998 | 268.000000 | 104.00 | 5.120000 |
| **4** | 2002-04-01 01:00:00 | 104.00 | 1.90 | 104.00 | 104.00 | 104.00 | 92.089996 | 237.199997 | 104.00 | 7.280000 |
| **5** | 2002-04-01 01:00:00 | 3.19 | 0.72 | 3.23 | 7.65 | 0.11 | 113.699997 | 187.000000 | 3.53 | 12.370000 |
| **6** | 2002-04-01 01:00:00 | 104.00 | 0.78 | 104.00 | 104.00 | 0.09 | 101.000000 | 119.300003 | 104.00 | 20.549999 |
| **7** | 2002-04-01 01:00:00 | 104.00 | 1.06 | 104.00 | 104.00 | 104.00 | 127.300003 | 204.100006 | 104.00 | 3.150000 |
| **8** | 2002-04-01 01:00:00 | 104.00 | 1.21 | 104.00 | 104.00 | 104.00 | 106.300003 | 126.599998 | 104.00 | 22.389999 |
| **9** | 2002-04-01 01:00:00 | 104.00 | 0.61 | 104.00 | 104.00 | 0.14 | 95.540001 | 110.699997 | 104.00 | 27.770000 |

In [93]:

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

Out[93]:

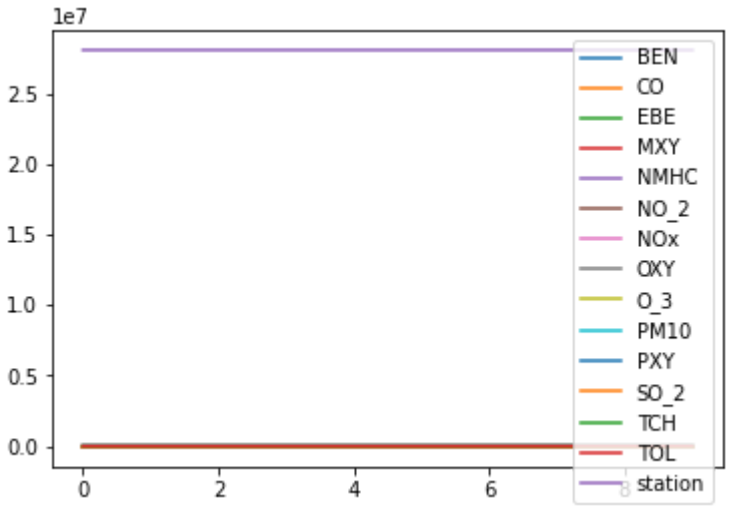
| | BEN | CO | EBE | MXY | NMHC | NO_2 | NOx | OXY | O_3 | PM10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 104.00 | 1.39 | 104.00 | 104.00 | 104.00 | 145.100006 | 352.100006 | 104.00 | 6.540000 | 41.990002 |
| **1** | 1.93 | 0.71 | 2.33 | 6.20 | 0.15 | 98.150002 | 153.399994 | 2.67 | 6.850000 | 20.980000 |
| **2** | 104.00 | 0.80 | 104.00 | 104.00 | 104.00 | 103.699997 | 134.000000 | 104.00 | 13.010000 | 28.440001 |
| **3** | 104.00 | 1.61 | 104.00 | 104.00 | 104.00 | 97.599998 | 268.000000 | 104.00 | 5.120000 | 42.180000 |
| **4** | 104.00 | 1.90 | 104.00 | 104.00 | 104.00 | 92.089996 | 237.199997 | 104.00 | 7.280000 | 76.330002 |
| **5** | 3.19 | 0.72 | 3.23 | 7.65 | 0.11 | 113.699997 | 187.000000 | 3.53 | 12.370000 | 27.450001 |
| **6** | 104.00 | 0.78 | 104.00 | 104.00 | 0.09 | 101.000000 | 119.300003 | 104.00 | 20.549999 | 23.950001 |
| **7** | 104.00 | 1.06 | 104.00 | 104.00 | 104.00 | 127.300003 | 204.100006 | 104.00 | 3.150000 | 49.639999 |
| **8** | 104.00 | 1.21 | 104.00 | 104.00 | 104.00 | 106.300003 | 126.599998 | 104.00 | 22.389999 | 32.090000 |
| **9** | 104.00 | 0.61 | 104.00 | 104.00 | 0.14 | 95.540001 | 110.699997 | 104.00 | 27.770000 | 24.610001 |

In [94]:

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

Out[94]:

```
<AxesSubplot:>
```

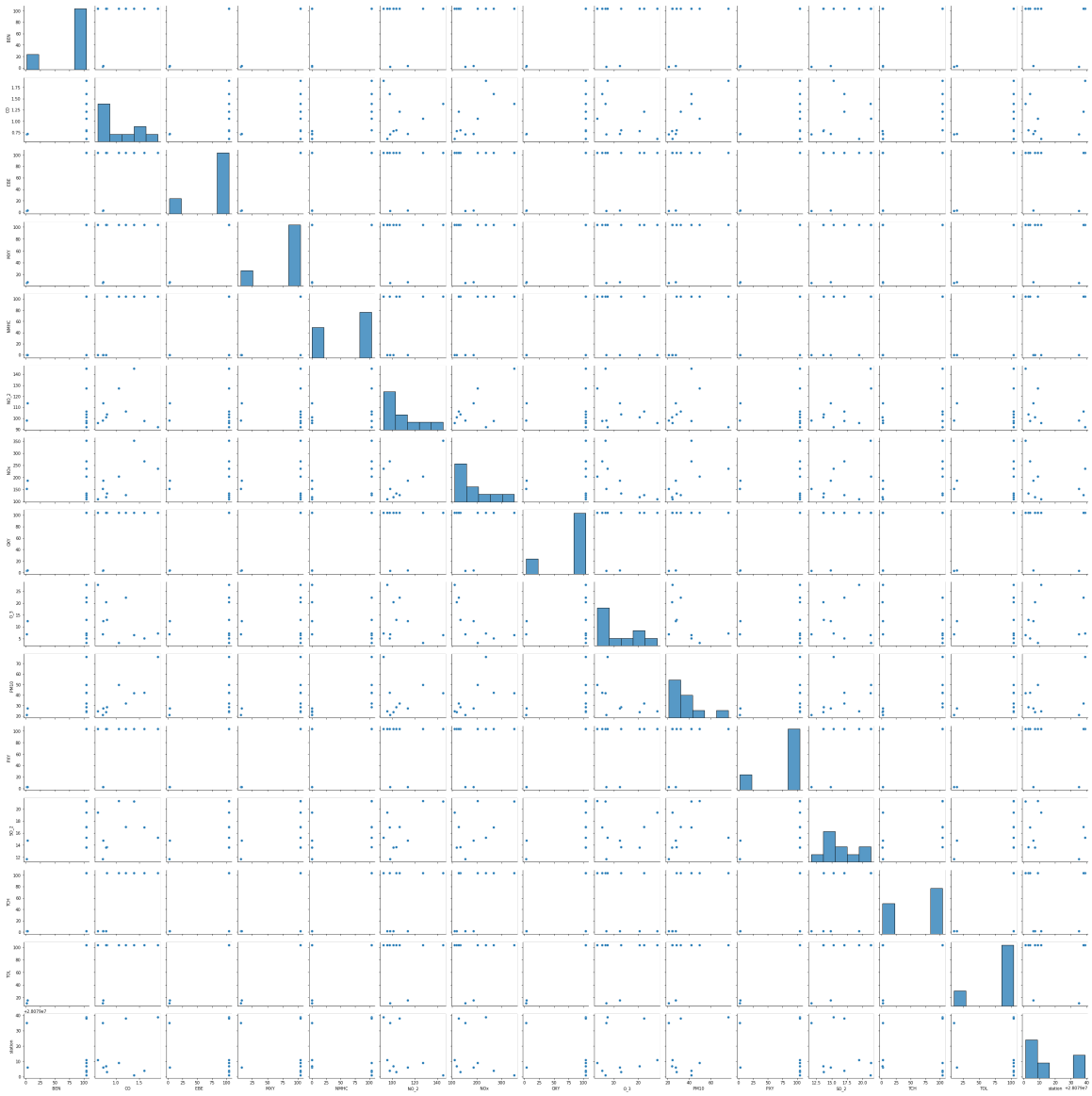


In [95]:

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

Out[95]:

```
<seaborn.axisgrid.PairGrid at 0x117cacc1af0>
```



In [96]:

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

In [97]:

```
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 [98]:

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

Out[98]:

```
LinearRegression()
```

In [99]:

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

```
1.7271590735801041
```

In [100]:

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

Out[100]:

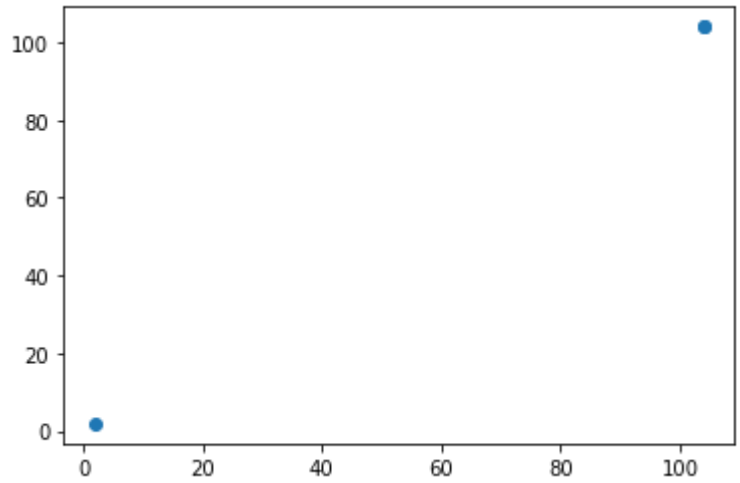
| | Co-efficient |
|---|---|
| **BEN** | -0.000077 |
| **CO** | -0.001505 |
| **EBE** | -0.000077 |
| **MXY** | -0.000073 |
| **NMHC** | 0.983742 |
| **NO_2** | -0.000031 |
| **NOx** | 0.000003 |
| **OXY** | -0.000076 |

In [101]:

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

Out[101]:

```
<matplotlib.collections.PathCollection at 0x117ca8e7790>
```




In [102]:

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

```
0.9999996394512323
```

In [103]:

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

In [104]:

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

Out[104]:

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

In [105]:

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

Out[105]:

```
0.999991844283914
```

In [106]:

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

Out[106]:

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

In [107]:

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

Out[107]:

```
0.999982587410832
```

In [108]:

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

Out[108]:

| | date | BEN | CO | EBE | MXY | NMHC | NO_2 | NOx | OXY | O_ |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2002-04-01 01:00:00 | 104.00 | 1.39 | 104.00 | 104.00 | 104.00 | 145.100006 | 352.100006 | 104.00 | 6.54000 |
| **1** | 2002-04-01 01:00:00 | 1.93 | 0.71 | 2.33 | 6.20 | 0.15 | 98.150002 | 153.399994 | 2.67 | 6.85000 |
| **2** | 2002-04-01 01:00:00 | 104.00 | 0.80 | 104.00 | 104.00 | 104.00 | 103.699997 | 134.000000 | 104.00 | 13.01000 |
| **3** | 2002-04-01 01:00:00 | 104.00 | 1.61 | 104.00 | 104.00 | 104.00 | 97.599998 | 268.000000 | 104.00 | 5.12000 |
| **4** | 2002-04-01 01:00:00 | 104.00 | 1.90 | 104.00 | 104.00 | 104.00 | 92.089996 | 237.199997 | 104.00 | 7.28000 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | . |
| **6995** | 2002-04-12 16:00:00 | 2.58 | 0.79 | 104.00 | 104.00 | 104.00 | 81.639999 | 146.500000 | 104.00 | 25.57000 |
| **6996** | 2002-04-12 16:00:00 | 1.73 | 0.71 | 1.31 | 104.00 | 0.13 | 59.470001 | 108.500000 | 104.00 | 32.32000 |
| **6997** | 2002-04-12 16:00:00 | 0.81 | 0.57 | 0.66 | 1.27 | 0.12 | 31.200001 | 36.630001 | 0.69 | 45.16000 |
| **6998** | 2002-04-12 16:00:00 | 1.62 | 0.58 | 1.62 | 3.69 | 104.00 | 64.000000 | 150.100006 | 1.13 | 26.11000 |
| **6999** | 2002-04-12 16:00:00 | 2.19 | 0.95 | 2.21 | 6.75 | 0.15 | 65.169998 | 117.099998 | 2.96 | 33.27999 |

7000 rows × 16 columns

In [109]:

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

In [110]:

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

In [111]:

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

In [112]:

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

Out[112]:

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

In [113]:

```
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 [114]:

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

In [115]:

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

```
[28079003]
```

In [116]:

```
logr.classes_
```

Out[116]:

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

In [117]:

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

Out[117]:

```
1.4508531683134893e-58
```

In [118]:

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

Out[118]:

```
0.9410614351603439
```

In [119]:

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

Out[119]:

```
0.5671428571428572
```

In [120]:

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

Out[120]:

```
ElasticNet()
```

In [121]:

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

```
[0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 9.83178447e-01 0.00000000e+00 1.58156823e-04 0.00000000e+00]
```

In [122]:

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

```
1.7007615561073877
```

In [123]:

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

```
0.9999994884228944
```

In [124]:

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

Out[124]:

```
RandomForestClassifier()
```

In [125]:

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

In [126]:

```
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[126]:

```
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 [127]:

```
grid_search.best_score_
```

Out[127]:

```
0.5681632653061224
```

In [128]:

```
rfc_best=grid_search.best_estimator_
```

In [129]:

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

Out[129]:

```
[Text(2143.2272727272725, 2491.5, 'X[10] <= -0.631\ngini = 0.96\nsamples
= 3104\nvalue = [179, 188, 179, 212, 174, 202, 221, 215, 198, 182\n168, 1
93, 207, 206, 197, 208, 190, 180, 207, 185\n215, 204, 192, 184, 214]'),
 Text(887.7272727272727, 2038.5, 'X[7] <= -2.041\ngini = 0.799\nsamples =
607\nvalue = [0, 0, 0, 210, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 174, 2
07, 185, 0, 0, 0, 0, 214]'),
 Text(507.27272727272725, 1585.5, 'X[12] <= -0.205\ngini = 0.673\nsamples
= 277\nvalue = [0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 155, 1
94, 63, 0, 0, 0, 0, 34]'),
 Text(405.8181818181818, 1132.5, 'X[12] <= -1.235\ngini = 0.562\nsamples
= 165\nvalue = [0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 155,
0, 63, 0, 0, 0, 0, 34]'),
 Text(202.9090909090909, 679.5, 'X[11] <= -0.451\ngini = 0.216\nsamples =
36\nvalue = [0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 52,
0, 0, 0, 0, 3]'),
 Text(101.45454545454545, 226.5, 'gini = 0.455\nsamples = 15\nvalue = [0,
0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 17, 0, 0, 0, 0,
3]'),
```

In [ ]: