In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as py
import seaborn as sns
from sklearn.linear_model import LogisticRegression
```

In [2]:
```python
df =pd.read_csv(r"D:\datasets\madrid_2009.csv")
df
```

Out[2]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	
0	2009-10-01 01:00:00	NaN	0.27	NaN	NaN	NaN	39.889999	48.150002	NaN	5
1	2009-10-01 01:00:00	NaN	0.22	NaN	NaN	NaN	21.230000	24.260000	NaN	5
2	2009-10-01 01:00:00	NaN	0.18	NaN	NaN	NaN	31.230000	34.880001	NaN	4
3	2009-10-01 01:00:00	0.95	0.33	1.43	2.68	0.25	55.180000	81.360001	1.57	3
4	2009-10-01 01:00:00	NaN	0.41	NaN	NaN	0.12	61.349998	76.260002	NaN	3
...	...	...	...	...	...	...	...	...	...	
215683	2009-06-01 00:00:00	0.50	0.22	0.39	0.75	0.09	22.000000	24.510000	1.00	8
215684	2009-06-01 00:00:00	NaN	0.31	NaN	NaN	NaN	76.110001	101.099998	NaN	4
215685	2009-06-01 00:00:00	0.13	NaN	0.86	NaN	0.23	81.050003	99.849998	NaN	2
215686	2009-06-01 00:00:00	0.21	NaN	2.96	NaN	0.10	72.419998	82.959999	NaN	
215687	2009-06-01 00:00:00	0.37	0.32	0.99	1.36	0.14	54.290001	64.480003	1.06	5

215688 rows × 17 columns

In [3]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215688 entries, 0 to 215687
Data columns (total 17 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     215688 non-null  object 
 1   BEN      60082 non-null   float64
 2   CO       190801 non-null  float64
 3   EBE      60081 non-null   float64
 4   MXY      24846 non-null   float64
 5   NMHC     74748 non-null   float64
 6   NO_2     214562 non-null  float64
 7   NOx      214565 non-null  float64
 8   OXY      24854 non-null   float64
 9   O_3      204482 non-null  float64
 10  PM10     196331 non-null  float64
 11  PM25     55822 non-null   float64
 12  PXY      24854 non-null   float64
 13  SO_2     212671 non-null  float64
 14  TCH      75213 non-null   float64
 15  TOL      59920 non-null   float64
 16  station  215688 non-null  int64  
dtypes: float64(15), int64(1), object(1)
memory usage: 28.0+ MB
```

In [4]:
```
df1 =df.fillna(value=0)
df1
```

Out[4]:

	date	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	
0	2009-10-01 01:00:00	0.00	0.27	0.00	0.00	0.00	39.889999	48.150002	0.00	5
1	2009-10-01 01:00:00	0.00	0.22	0.00	0.00	0.00	21.230000	24.260000	0.00	5
2	2009-10-01 01:00:00	0.00	0.18	0.00	0.00	0.00	31.230000	34.880001	0.00	4
3	2009-10-01 01:00:00	0.95	0.33	1.43	2.68	0.25	55.180000	81.360001	1.57	3
4	2009-10-01 01:00:00	0.00	0.41	0.00	0.00	0.12	61.349998	76.260002	0.00	3
...	...	...	...	...	...	...	...	...	...	
215683	2009-06-01 00:00:00	0.50	0.22	0.39	0.75	0.09	22.000000	24.510000	1.00	8
215684	2009-06-01 00:00:00	0.00	0.31	0.00	0.00	0.00	76.110001	101.099998	0.00	4
215685	2009-06-01 00:00:00	0.13	0.00	0.86	0.00	0.23	81.050003	99.849998	0.00	2
215686	2009-06-01 00:00:00	0.21	0.00	2.96	0.00	0.10	72.419998	82.959999	0.00	
215687	2009-06-01 00:00:00	0.37	0.32	0.99	1.36	0.14	54.290001	64.480003	1.06	5

215688 rows × 17 columns

In [5]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215688 entries, 0 to 215687
Data columns (total 17 columns):
 #   Column   Non-Null Count   Dtype  
---  ------   --------------   -----  
 0   date     215688 non-null  object 
 1   BEN      215688 non-null  float64
 2   CO       215688 non-null  float64
 3   EBE      215688 non-null  float64
 4   MXY      215688 non-null  float64
 5   NMHC     215688 non-null  float64
 6   NO_2     215688 non-null  float64
 7   NOx      215688 non-null  float64
 8   OXY      215688 non-null  float64
 9   O_3      215688 non-null  float64
 10  PM10     215688 non-null  float64
 11  PM25     215688 non-null  float64
 12  PXY      215688 non-null  float64
 13  SO_2     215688 non-null  float64
 14  TCH      215688 non-null  float64
 15  TOL      215688 non-null  float64
 16  station  215688 non-null  int64  
dtypes: float64(15), int64(1), object(1)
memory usage: 28.0+ MB
```

In [6]:
```
df1.columns
```

Out[6]:
```
Index(['date', 'BEN', 'CO', 'EBE', 'MXY', 'NMHC', 'NO_2', 'NOx', 'OXY', 'O_
3',
       'PM10', 'PM25', 'PXY', 'SO_2', 'TCH', 'TOL', 'station'],
      dtype='object')
```

In [7]:
```
df2=df1[['BEN', 'CO', 'EBE', 'MXY', 'NMHC', 'NO_2', 'NOx', 'OXY', 'O_3',
       'PM10', 'PM25', 'PXY', 'SO_2', 'TCH', 'TOL', 'station']]
df2
```

Out[7]:

	BEN	CO	EBE	MXY	NMHC	NO_2	NOx	OXY	O_3
0	0.00	0.27	0.00	0.00	0.00	39.889999	48.150002	0.00	50.680000
1	0.00	0.22	0.00	0.00	0.00	21.230000	24.260000	0.00	55.880001
2	0.00	0.18	0.00	0.00	0.00	31.230000	34.880001	0.00	49.060001
3	0.95	0.33	1.43	2.68	0.25	55.180000	81.360001	1.57	36.669998
4	0.00	0.41	0.00	0.00	0.12	61.349998	76.260002	0.00	38.090000
...	...	...	...	...	...	...	...	...	...
215683	0.50	0.22	0.39	0.75	0.09	22.000000	24.510000	1.00	82.239998
215684	0.00	0.31	0.00	0.00	0.00	76.110001	101.099998	0.00	41.220001
215685	0.13	0.00	0.86	0.00	0.23	81.050003	99.849998	0.00	24.830000
215686	0.21	0.00	2.96	0.00	0.10	72.419998	82.959999	0.00	0.000000
215687	0.37	0.32	0.99	1.36	0.14	54.290001	64.480003	1.06	56.919998

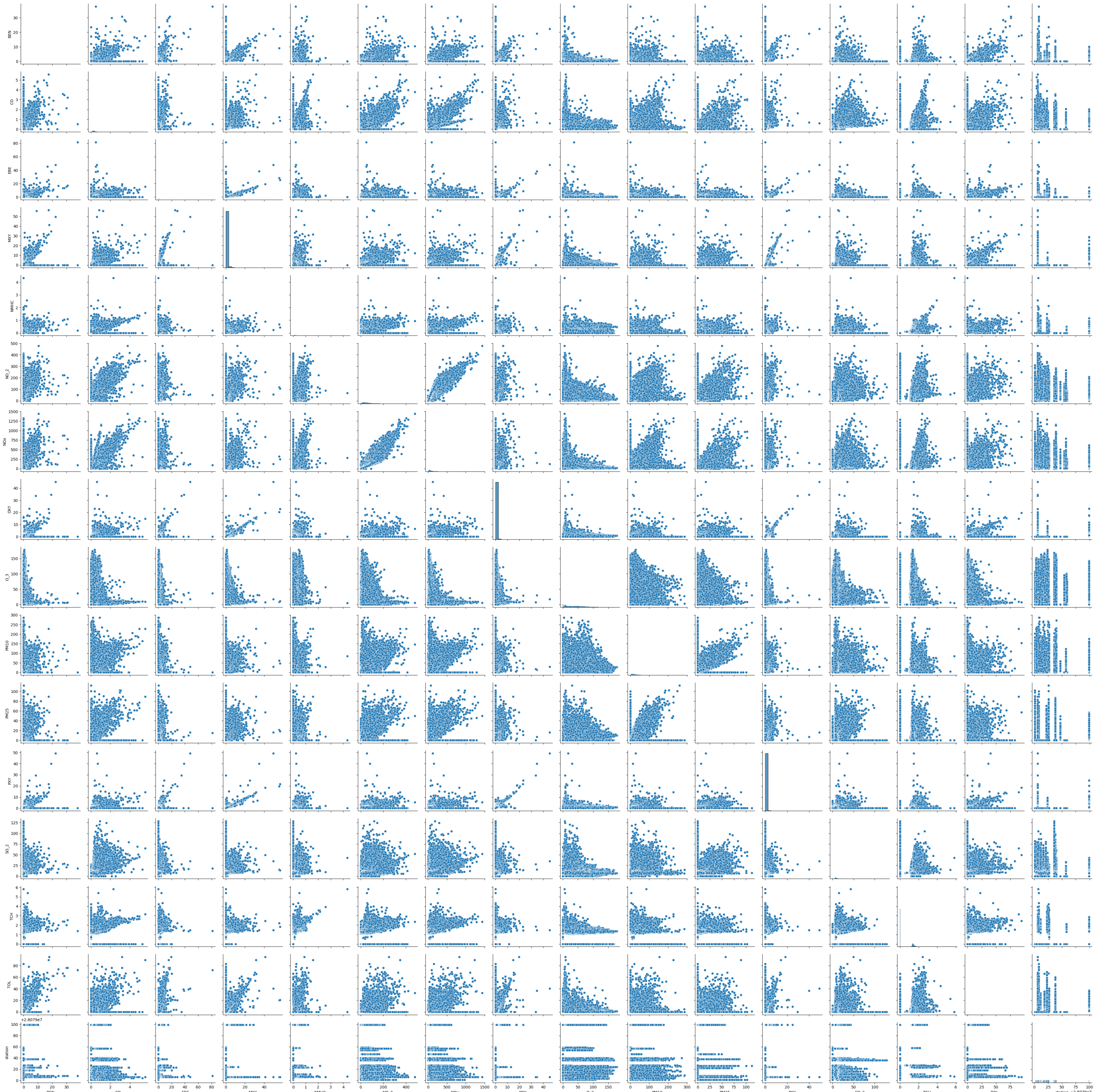
215688 rows × 16 columns

In [8]:

```
sns.pairplot(df2)
```

```
C:\Users\HP\AppData\Local\Programs\Python\Python311\Lib\site-packages\seabor
n\axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

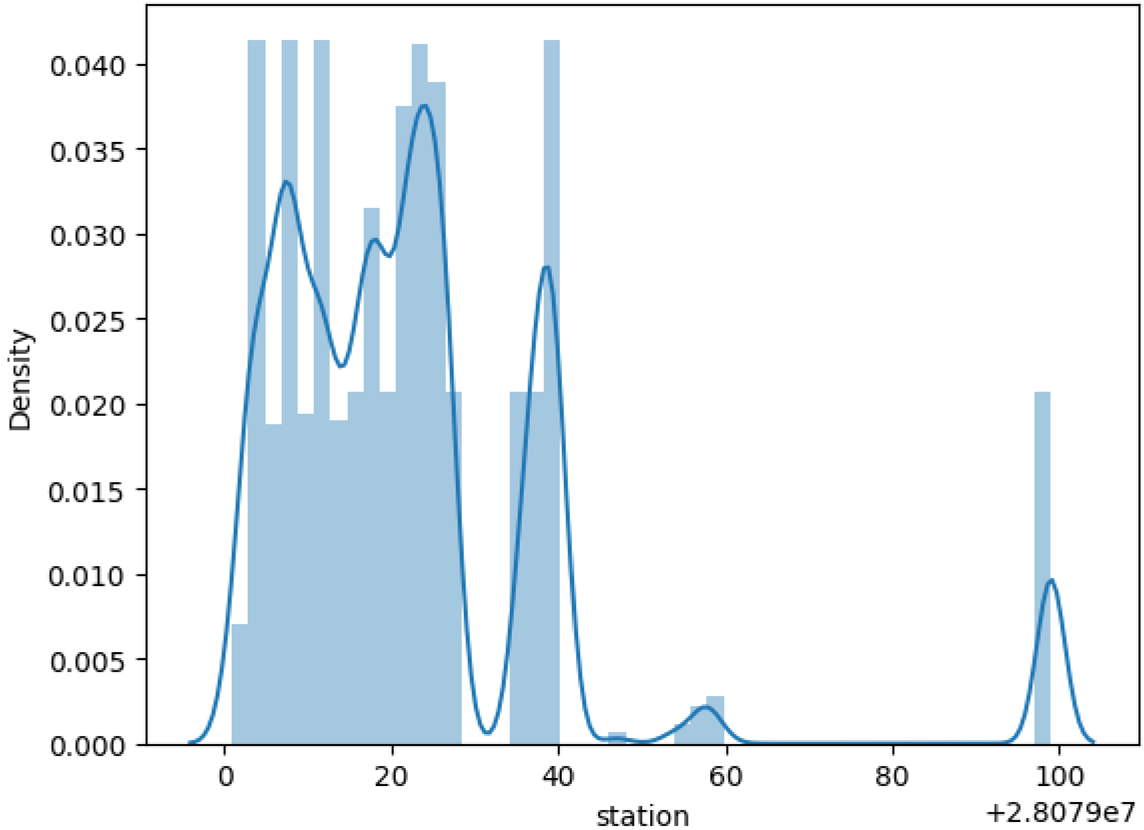
Out[8]: <seaborn.axisgrid.PairGrid at 0x17d8ba9ebd0>



In [9]:
```python
sns.distplot(df2['station'])
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_10528\1070072814.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df2['station'])
```

Out[9]: <Axes: xlabel='station', ylabel='Density'>



In [10]:
```
x=df2[['BEN', 'CO', 'EBE', 'MXY', 'NMHC', 'NO_2', 'NOx', 'OXY', 'O_3',
       'PM10', 'PM25', 'PXY', 'SO_2', 'TCH', 'TOL']]
y=df2['station']
```

In [11]:
```
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)
```

linear

In [12]:
```
from sklearn.linear_model import LinearRegression
```

In [13]:
```
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[13]:
```
▼ LinearRegression
LinearRegression()
```

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

Out[14]:

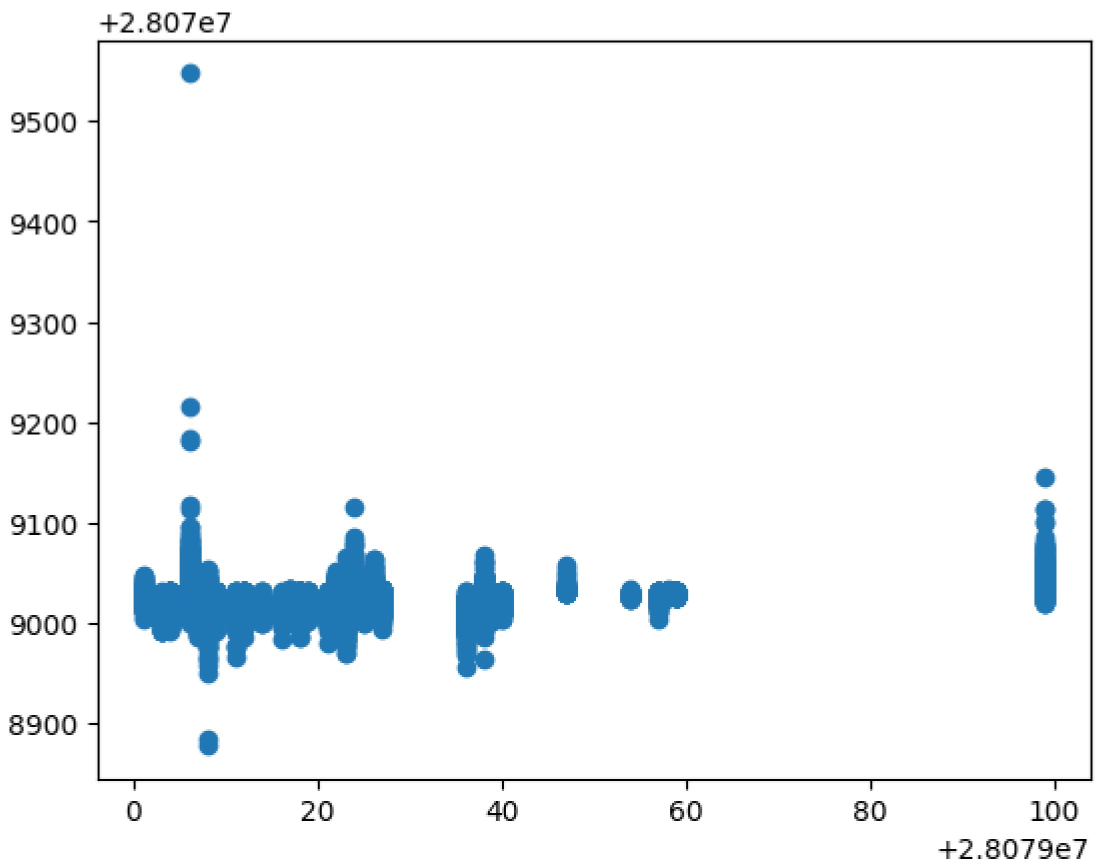
	Co-efficient
BEN	-4.615807
CO	-4.992281
EBE	1.758629
MXY	-3.154330
NMHC	5.333756
NO_2	-0.115052
NOx	0.038630
OXY	9.468954
O_3	-0.048493
PM10	-0.014585
PM25	0.540350
PXY	6.623647
SO_2	-0.482322
TCH	0.235949
TOL	-0.331949

In [15]:
```
print(lr.intercept_)
```

```
28079031.57755233
```

In [16]:
```
prediction =lr.predict(x_test)
py.scatter(y_test,prediction)
```

Out[16]: <matplotlib.collections.PathCollection at 0x17da7023fd0>



In [17]:
```
print(lr.score(x_test,y_test))
```

```
0.16164022524589827
```

In [18]:
```
print(lr.score(x_train,y_train))
```

```
0.16574919725540282
```

Ridge

In [19]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [20]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[20]:
```
▼ Ridge
Ridge(alpha=10)
```

In [21]:
```
rr.score(x_test,y_test)
```

Out[21]: 0.16164686275461426

Lasso

In [22]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[22]:
```
▼     Lasso
Lasso(alpha=10)
```

In [23]:
```
la.score(x_test,y_test)
```

Out[23]:
```
0.08330465988481084
```

elasticnet

In [24]:
```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[24]:
```
▼ ElasticNet
ElasticNet()
```

In [25]:
```
print(en.coef_)
```

```
[-0.         -0.          0.25594261  1.44132117  0.         -0.10674645
  0.0164265   1.08447441 -0.04738088 -0.00726588  0.62195308  0.90028299
 -0.50143208  0.28762132 -0.20115409]
```

In [26]:
```
print(en.intercept_)
```

```
28079031.76020447
```

In [27]:
```
print(en.predict(x_test))
```

```
[28079022.11968691 28079017.58153    28079024.58582972 ...
 28079033.68694673 28079021.0396411  28079021.17881544]
```

In [28]:
```
print(en.score(x_test,y_test))
```

```
0.13279398323860492
```

logistic

In [29]:
```
feature_matrix =df2.iloc[:,0:15]
target_vector=df2.iloc[:,-1]
```

In [30]:
```
feature_matrix=df2[['BEN', 'CO', 'EBE', 'MXY', 'NMHC', 'NO_2', 'NOx', 'OXY',
       'PM10', 'PM25', 'PXY', 'SO_2', 'TCH', 'TOL']]
y=df2['station']
```

In [31]:
```
feature_matrix.shape
```

Out[31]:
```
(215688, 15)
```

In [32]:
```
target_vector.shape
```

Out[32]:
```
(215688,)
```

In [33]:
```
from sklearn.preprocessing import StandardScaler
```

In [34]:
```
fs=StandardScaler().fit_transform(feature_matrix)
```

In [35]:
```
logr = LogisticRegression()
logr.fit(fs,target_vector)
```

```
C:\Users\HP\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklear
n\linear_model\_logistic.py:460: ConvergenceWarning: lbfgs failed to converg
e (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regre
ssion
  n_iter_i = _check_optimize_result(
```

Out[35]:
```
▼ LogisticRegression
LogisticRegression()
```

In [36]:
```
observation=[[1,2,3,4,5,6,7,8,9,11,12,13,14,15,16]]
```

In [37]:
```
prediction =logr.predict(observation)
print(prediction)
```

```
[28079099]
```

In [38]:
```
logr.classes_
```

Out[38]:
```
array([28079001, 28079003, 28079004, 28079006, 28079007, 28079008,
       28079009, 28079011, 28079012, 28079014, 28079016, 28079017,
       28079018, 28079019, 28079021, 28079022, 28079023, 28079024,
       28079025, 28079026, 28079027, 28079036, 28079038, 28079039,
       28079040, 28079047, 28079054, 28079057, 28079058, 28079059,
       28079099], dtype=int64)
```

In [39]:
```
logr.score(fs,target_vector)
```

Out[39]: 0.5425522050369052

In [40]:
```
logr.predict_proba(observation)[0][0]
```

Out[40]: 3.7014022758991974e-133

In [41]:
```
logr.predict_proba(observation)[0][1]
```

Out[41]: 1.9803684011074334e-191

Random forest

In [42]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import plot_tree
```

In [43]:
```
x=df2.drop('station',axis=1)
y=df2['station']
```

In [44]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.70)
```

In [45]:
```
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
```

Out[45]:
```
▼ RandomForestClassifier
RandomForestClassifier()
```

In [46]:
```
parameters={'max_depth':[1,2,3,4,5],
            'min_samples_leaf' :[6,7,8,9,10],
            'n_estimators':[11,12,13,14,15]}
```

In [47]:
```
from sklearn.model_selection import GridSearchCV
```

In [48]:
```
grid_search =GridSearchCV(estimator =rfc,param_grid=parameters,cv=2,scoring=
grid_search.fit(x_train,y_train)
```

Out[48]:
```
▶ GridSearchCV
▶ estimator: RandomForestClassifier
  ▶ RandomForestClassifier
```

In [49]:
```
grid_search.best_score_
```

Out[49]: 0.5005718171421506

In [50]:
```
rfc_best=grid_search.best_estimator_
```

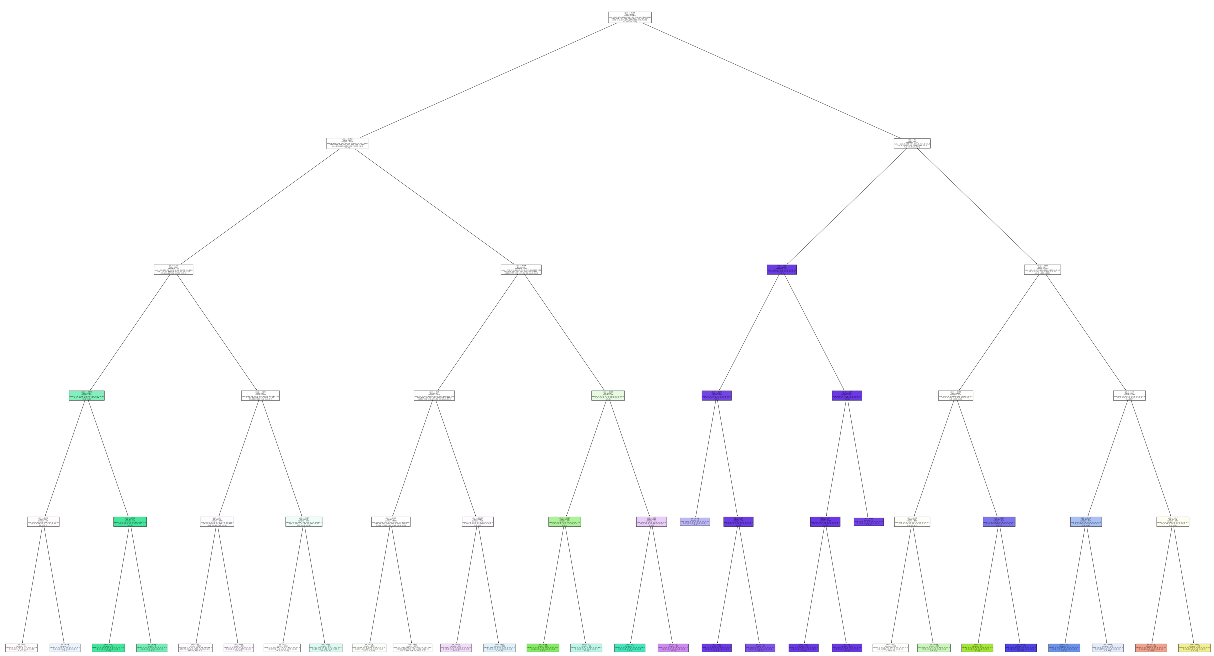
In [51]:
```
py.figure(figsize=(80,50))
plot_tree(rfc_best.estimators_[5],filled=True)
```

```
Out[51]: [Text(0.5178571428571429, 0.9166666666666666, 'x[13] <= 0.305\ngini = 0.961
\nsamples = 40817\nvalue = [869, 2706, 2665, 2334, 2727, 2672, 2475, 2510,
2625\n2385, 2659, 1347, 2609, 2604, 2401, 2301, 2545, 2574\n2370, 2643, 264
6, 2741, 2672, 2598, 2600, 76, 178\n292, 210, 122, 2550]'),
 Text(0.2857142857142857, 0.75, 'x[5] <= 30.735\ngini = 0.943\nsamples = 26
685\nvalue = [869, 2706, 2665, 38, 273, 9, 2475, 476, 2625, 2385\n2659, 134
7, 2609, 2604, 2401, 2301, 224, 23, 2370\n11, 4, 2741, 2672, 2598, 2600, 7
6, 178, 292, 0\n122, 0]'),
 Text(0.14285714285714285, 0.5833333333333334, 'x[9] <= 0.375\ngini = 0.935
\nsamples = 7722\nvalue = [98, 530, 616, 29, 50, 8, 428, 63, 375, 404, 1196
\n738, 1122, 562, 449, 654, 33, 23, 682, 10, 4, 732\n895, 1098, 1173, 16, 5
8, 104, 0, 53, 0]'),
 Text(0.07142857142857142, 0.4166666666666667, 'x[12] <= 6.545\ngini = 0.56
7\nsamples = 745\nvalue = [3, 2, 12, 28, 16, 8, 28, 1, 20, 18, 3, 738, 31\n
5, 6, 7, 33, 14, 9, 9, 2, 1, 3, 6, 4, 0, 58\n15, 0, 53, 0]'),
 Text(0.03571428571428571, 0.25, 'x[1] <= 0.075\ngini = 0.92\nsamples = 228
\nvalue = [3, 0, 10, 28, 9, 8, 28, 1, 20, 5, 3, 5, 19\n5, 6, 7, 26, 14, 9,
9, 2, 1, 3, 6, 4, 0, 58\n15, 0, 53, 0]'),
 Text(0.017857142857142856, 0.08333333333333333, 'gini = 0.91\nsamples = 19
9\nvalue = [3, 0, 10, 28, 9, 6, 15, 1, 20, 5, 2, 5, 16\n5, 6, 7, 10, 14, 7,
9, 2, 1, 3, 1, 4, 0, 58\n15, 0, 53, 0]'),
 Text(0.05357142857142857, 0.08333333333333333, 'gini = 0.735\nsamples = 29
\nvalue = [0, 0, 0, 0, 0, 2, 13, 0, 0, 0, 1, 0, 3, 0\n0, 0, 16, 0, 2, 0, 0,
0, 0, 5, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.10714285714285714, 0.25, 'x[6] <= 47.36\ngini = 0.107\nsamples = 51
7\nvalue = [0, 2, 2, 0, 7, 0, 0, 0, 0, 13, 0, 733, 12, 0\n0, 0, 7, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.08928571428571429, 0.08333333333333333, 'gini = 0.101\nsamples = 50
8\nvalue = [0, 2, 2, 0, 7, 0, 0, 0, 0, 13, 0, 723, 12, 0\n0, 0, 4, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.125, 0.08333333333333333, 'gini = 0.355\nsamples = 9\nvalue = [0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 10, 0, 0\n0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0\n0, 0, 0]'),
 Text(0.21428571428571427, 0.4166666666666667, 'x[12] <= 8.145\ngini = 0.92
7\nsamples = 6977\nvalue = [95, 528, 604, 1, 34, 0, 400, 62, 355, 386, 1193
\n0, 1091, 557, 443, 647, 0, 9, 673, 1, 2, 731\n892, 1092, 1169, 16, 0, 89,
0, 0, 0]'),
 Text(0.17857142857142858, 0.25, 'x[10] <= 0.3\ngini = 0.911\nsamples = 468
9\nvalue = [0, 93, 0, 0, 0, 0, 376, 5, 296, 367, 966, 0\n239, 510, 266, 63
5, 0, 8, 589, 1, 2, 385, 657\n1048, 892, 16, 0, 89, 0, 0, 0]'),
 Text(0.16071428571428573, 0.08333333333333333, 'gini = 0.891\nsamples = 38
44\nvalue = [0, 93, 0, 0, 0, 0, 376, 5, 296, 367, 966, 0\n239, 510, 266, 2,
0, 0, 589, 0, 2, 385, 0, 1048\n892, 0, 0, 89, 0, 0, 0]'),
 Text(0.19642857142857142, 0.08333333333333333, 'gini = 0.518\nsamples = 84
5\nvalue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 633, 0, 8, 0, 1,
0, 0, 657, 0, 0, 16, 0\n0, 0, 0, 0]'),
 Text(0.25, 0.25, 'x[8] <= 70.87\ngini = 0.875\nsamples = 2288\nvalue = [9
5, 435, 604, 1, 34, 0, 24, 57, 59, 19, 227, 0\n852, 47, 177, 12, 0, 1, 84,
0, 0, 346, 235, 44\n277, 0, 0, 0, 0, 0, 0]'),
 Text(0.23214285714285715, 0.08333333333333333, 'gini = 0.9\nsamples = 1445
\nvalue = [74, 215, 284, 1, 34, 0, 24, 40, 45, 15, 205, 0\n341, 37, 141, 1
1, 0, 1, 32, 0, 0, 340, 188, 31\n196, 0, 0, 0, 0, 0, 0]'),
 Text(0.26785714285714285, 0.08333333333333333, 'gini = 0.774\nsamples = 84
3\nvalue = [21, 220, 320, 0, 0, 0, 0, 17, 14, 4, 22, 0\n511, 10, 36, 1, 0,
0, 52, 0, 0, 6, 47, 13, 81\n0, 0, 0, 0, 0, 0]'),
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js , 0.5833333333333334, 'x[2] <= 0.065\ngini = 0.942
```

```
\nsamples = 18963\nvalue = [771, 2176, 2049, 9, 223, 1, 2047, 413, 2250, 19
81\n1463, 609, 1487, 2042, 1952, 1647, 191, 0, 1688, 1\n0, 2009, 1777, 150
0, 1427, 60, 120, 188, 0, 69, 0]'),
 Text(0.35714285714285715, 0.4166666666666667, 'x[10] <= 0.36\ngini = 0.941
\nsamples = 18675\nvalue = [771, 2176, 2049, 0, 223, 0, 2047, 223, 2250, 19
81\n1463, 609, 1347, 2042, 1952, 1647, 176, 0, 1688, 0\n0, 2009, 1671, 150
0, 1427, 60, 120, 188, 0, 69, 0]'),
 Text(0.32142857142857145, 0.25, 'x[12] <= 6.405\ngini = 0.93\nsamples = 16
046\nvalue = [1, 2176, 2049, 0, 223, 0, 2047, 223, 2250, 1981\n1463, 609, 1
347, 2042, 1952, 20, 176, 0, 1688, 0\n0, 2009, 2, 1500, 1427, 0, 120, 188,
0, 69, 0]'),
 Text(0.30357142857142855, 0.08333333333333333, 'gini = 0.818\nsamples = 23
23\nvalue = [0, 84, 0, 0, 0, 0, 945, 0, 848, 1, 54, 0, 91\n1, 167, 4, 15,
0, 867, 0, 0, 57, 0, 7, 413, 0\n120, 37, 0, 69, 0]'),
 Text(0.3392857142857143, 0.08333333333333333, 'gini = 0.927\nsamples = 137
23\nvalue = [1, 2092, 2049, 0, 223, 0, 1102, 223, 1402, 1980\n1409, 609, 12
56, 2041, 1785, 16, 161, 0, 821, 0, 0\n1952, 2, 1493, 1014, 0, 0, 151, 0,
0, 0]'),
 Text(0.39285714285714285, 0.25, 'x[5] <= 63.695\ngini = 0.646\nsamples = 2
629\nvalue = [770, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 1627, 0, 0, 0,
0, 0, 0, 1669, 0, 0, 60, 0\n0, 0, 0, 0]'),
 Text(0.375, 0.08333333333333333, 'gini = 0.622\nsamples = 1426\nvalue = [3
14, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 816, 0, 0, 0, 0, 0, 0, 1065,
0, 0, 48, 0\n0, 0, 0, 0]'),
 Text(0.4107142857142857, 0.08333333333333333, 'gini = 0.653\nsamples = 120
3\nvalue = [456, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 811, 0, 0, 0, 0,
0, 0, 604, 0, 0, 12, 0\n0, 0, 0, 0]'),
 Text(0.5, 0.4166666666666667, 'x[2] <= 1.035\ngini = 0.685\nsamples = 288
\nvalue = [0, 0, 0, 9, 0, 1, 0, 190, 0, 0, 0, 0, 140, 0\n0, 0, 15, 0, 0, 1,
0, 0, 106, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.4642857142857143, 0.25, 'x[14] <= 3.385\ngini = 0.524\nsamples = 17
4\nvalue = [0, 0, 0, 4, 0, 1, 0, 186, 0, 0, 0, 0, 73, 0\n0, 0, 13, 0, 0, 1,
0, 0, 13, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.44642857142857145, 0.08333333333333333, 'gini = 0.359\nsamples = 11
4\nvalue = [0, 0, 0, 4, 0, 0, 0, 146, 0, 0, 0, 0, 10, 0\n0, 0, 11, 0, 0, 1,
0, 0, 12, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.48214285714285715, 0.08333333333333333, 'gini = 0.513\nsamples = 60
\nvalue = [0, 0, 0, 0, 0, 1, 0, 40, 0, 0, 0, 0, 63, 0\n0, 0, 2, 0, 0, 0, 0,
0, 1, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.5357142857142857, 0.25, 'x[8] <= 6.36\ngini = 0.549\nsamples = 114
\nvalue = [0, 0, 0, 5, 0, 0, 0, 4, 0, 0, 0, 0, 67, 0\n0, 0, 2, 0, 0, 0, 0,
0, 93, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.5178571428571429, 0.08333333333333333, 'gini = 0.1\nsamples = 29\nv
alue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 36, 0\n0, 0, 0, 0, 0, 0, 0, 0,
2, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.5535714285714286, 0.08333333333333333, 'gini = 0.475\nsamples = 85
\nvalue = [0, 0, 0, 5, 0, 0, 0, 4, 0, 0, 0, 0, 31, 0\n0, 0, 2, 0, 0, 0, 0,
0, 91, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.75, 0.75, 'x[8] <= 0.3\ngini = 0.89\nsamples = 14132\nvalue = [0,
0, 0, 2296, 2454, 2663, 0, 2034, 0, 0, 0, 0\n0, 0, 0, 0, 2321, 2551, 0, 263
2, 2642, 0, 0, 0\n0, 0, 0, 0, 210, 0, 2550]'),
 Text(0.6428571428571429, 0.5833333333333334, 'x[2] <= 0.565\ngini = 0.019
\nsamples = 1679\nvalue = [0, 0, 0, 1, 0, 2, 0, 3, 0, 0, 0, 0, 0, 0\n0, 0,
0, 0, 0, 12, 2618, 0, 0, 0, 0, 0, 0, 0\n7, 0, 0]'),
 Text(0.5892857142857143, 0.4166666666666667, 'x[12] <= 3.245\ngini = 0.112
\nsamples = 257\nvalue = [0, 0, 0, 0, 0, 2, 0, 3, 0, 0, 0, 0, 0, 0\n0, 0,
```

```
0, 0, 0, 11, 371, 0, 0, 0, 0, 0, 0, 0\n7, 0, 0]'),
 Text(0.5714285714285714, 0.25, 'gini = 0.475\nsamples = 9\nvalue = [0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 11, 0, 0, 0, 0, 0, 0, 0,
0\n7, 0, 0]'),
 Text(0.6071428571428571, 0.25, 'x[4] <= 0.305\ngini = 0.026\nsamples = 248
\nvalue = [0, 0, 0, 0, 0, 2, 0, 3, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 371,
0, 0, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.5892857142857143, 0.08333333333333333, 'gini = 0.011\nsamples = 235
\nvalue = [0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 349,
0, 0, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.625, 0.08333333333333333, 'gini = 0.211\nsamples = 13\nvalue = [0,
0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 22, 0, 0, 0, 0, 0,
0, 0\n0, 0, 0]'),
 Text(0.6964285714285714, 0.4166666666666667, 'x[14] <= 13.7\ngini = 0.002
\nsamples = 1422\nvalue = [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0,
0, 0, 0, 1, 2247, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.6785714285714286, 0.25, 'x[14] <= 0.565\ngini = 0.001\nsamples = 14
08\nvalue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 1, 22
28, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.6607142857142857, 0.08333333333333333, 'gini = 0.005\nsamples = 238
\nvalue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 1, 384,
0, 0, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.6964285714285714, 0.08333333333333333, 'gini = 0.0\nsamples = 1170
\nvalue = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 184
4, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.7142857142857143, 0.25, 'gini = 0.095\nsamples = 14\nvalue = [0, 0,
0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 19, 0, 0, 0, 0, 0, 0,
0\n0, 0, 0]'),
 Text(0.8571428571428571, 0.5833333333333334, 'x[11] <= 0.105\ngini = 0.877
\nsamples = 12453\nvalue = [0, 0, 0, 2295, 2454, 2661, 0, 2031, 0, 0, 0, 0
\n0, 0, 0, 0, 2321, 2551, 0, 2620, 24, 0, 0, 0\n0, 0, 0, 0, 203, 0, 255
0]'),
 Text(0.7857142857142857, 0.4166666666666667, 'x[10] <= 0.365\ngini = 0.809
\nsamples = 7872\nvalue = [0, 0, 0, 88, 2454, 2661, 0, 2031, 0, 0, 0, 0\n0,
0, 0, 0, 2321, 22, 0, 2620, 24, 0, 0, 0, 0\n0, 0, 203, 0, 23]'),
 Text(0.75, 0.25, 'x[4] <= 0.315\ngini = 0.76\nsamples = 5735\nvalue = [0,
0, 0, 0, 2454, 1967, 0, 2031, 0, 0, 0, 0, 0\n0, 0, 0, 2321, 0, 0, 10, 24,
0, 0, 0, 0, 0\n0, 0, 203, 0, 0]'),
 Text(0.7321428571428571, 0.08333333333333333, 'gini = 0.758\nsamples = 500
2\nvalue = [0, 0, 0, 0, 2202, 1853, 0, 1477, 0, 0, 0, 0, 0\n0, 0, 0, 2122,
0, 0, 7, 24, 0, 0, 0, 0, 0, 0\n0, 195, 0, 0]'),
 Text(0.7678571428571429, 0.08333333333333333, 'gini = 0.669\nsamples = 733
\nvalue = [0, 0, 0, 0, 252, 114, 0, 554, 0, 0, 0, 0, 0\n0, 0, 0, 199, 0, 0,
3, 0, 0, 0, 0, 0, 0, 0\n0, 8, 0, 0]'),
 Text(0.8214285714285714, 0.25, 'x[9] <= 0.715\ngini = 0.382\nsamples = 213
7\nvalue = [0, 0, 0, 88, 0, 694, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 22, 0, 26
10, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 23]'),
 Text(0.8035714285714286, 0.08333333333333333, 'gini = 0.0\nsamples = 422\n
value = [0, 0, 0, 0, 0, 679, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0\n0, 0, 0]'),
 Text(0.8392857142857143, 0.08333333333333333, 'gini = 0.103\nsamples = 171
5\nvalue = [0, 0, 0, 88, 0, 15, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 22, 0, 261
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 23]'),
 Text(0.9285714285714286, 0.4166666666666667, 'x[2] <= 1.005\ngini = 0.665
\nsamples = 4581\nvalue = [0, 0, 0, 2207, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0,
0, 0, 2529, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 2527]'),
```

```
 Text(0.8928571428571429, 0.25, 'x[13] <= 1.335\ngini = 0.567\nsamples = 25
26\nvalue = [0, 0, 0, 611, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 2323, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 1051]'),
 Text(0.875, 0.08333333333333333, 'gini = 0.357\nsamples = 917\nvalue = [0,
0, 0, 234, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 1146, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0\n0, 0, 82]'),
 Text(0.9107142857142857, 0.08333333333333333, 'gini = 0.613\nsamples = 160
9\nvalue = [0, 0, 0, 377, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 1177, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 969]'),
 Text(0.9642857142857143, 0.25, 'x[6] <= 112.7\ngini = 0.556\nsamples = 205
5\nvalue = [0, 0, 0, 1596, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 206, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 1476]'),
 Text(0.9464285714285714, 0.08333333333333333, 'gini = 0.479\nsamples = 882
\nvalue = [0, 0, 0, 337, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 0, 121, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 957]'),
 Text(0.9821428571428571, 0.08333333333333333, 'gini = 0.464\nsamples = 117
3\nvalue = [0, 0, 0, 1259, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 85, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0\n0, 0, 519]')]
```



concusion

The bestfit model is logistic Regression with score of 0.5425522050369052

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js