In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data = pd.read_csv('WA_Fn-UseC_-HR-Employee-Attrition.csv')
data.head()
```

Out[2]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Educa
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life
4	27	No	Travel_Rarely	591	Research & Development	2	1	

5 rows × 35 columns

In [3]:
```
data.tail()
```

Out[3]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Ed
1465	36	No	Travel_Frequently	884	Research & Development	23	2	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	
1468	49	No	Travel_Frequently	1023	Sales	2	3	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	

5 rows × 35 columns

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1470 non-null   int64
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64
 6   Education                 1470 non-null   int64
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64
 9   EmployeeNumber            1470 non-null   int64
 10  EnvironmentSatisfaction   1470 non-null   int64
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64
 13  JobInvolvement            1470 non-null   int64
 14  JobLevel                  1470 non-null   int64
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64
 19  MonthlyRate               1470 non-null   int64
 20  NumCompaniesWorked        1470 non-null   int64
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64
 24  PerformanceRating         1470 non-null   int64
 25  RelationshipSatisfaction  1470 non-null   int64
 26  StandardHours             1470 non-null   int64
 27  StockOptionLevel          1470 non-null   int64
 28  TotalWorkingYears         1470 non-null   int64
 29  TrainingTimesLastYear     1470 non-null   int64
 30  WorkLifeBalance           1470 non-null   int64
 31  YearsAtCompany            1470 non-null   int64
 32  YearsInCurrentRole        1470 non-null   int64
 33  YearsSinceLastPromotion   1470 non-null   int64
 34  YearsWithCurrManager      1470 non-null   int64
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

In [5]:
```
data.describe()
```

Out[5]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNum
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000

8 rows × 26 columns

In [6]:
```
data.isnull().sum()
```

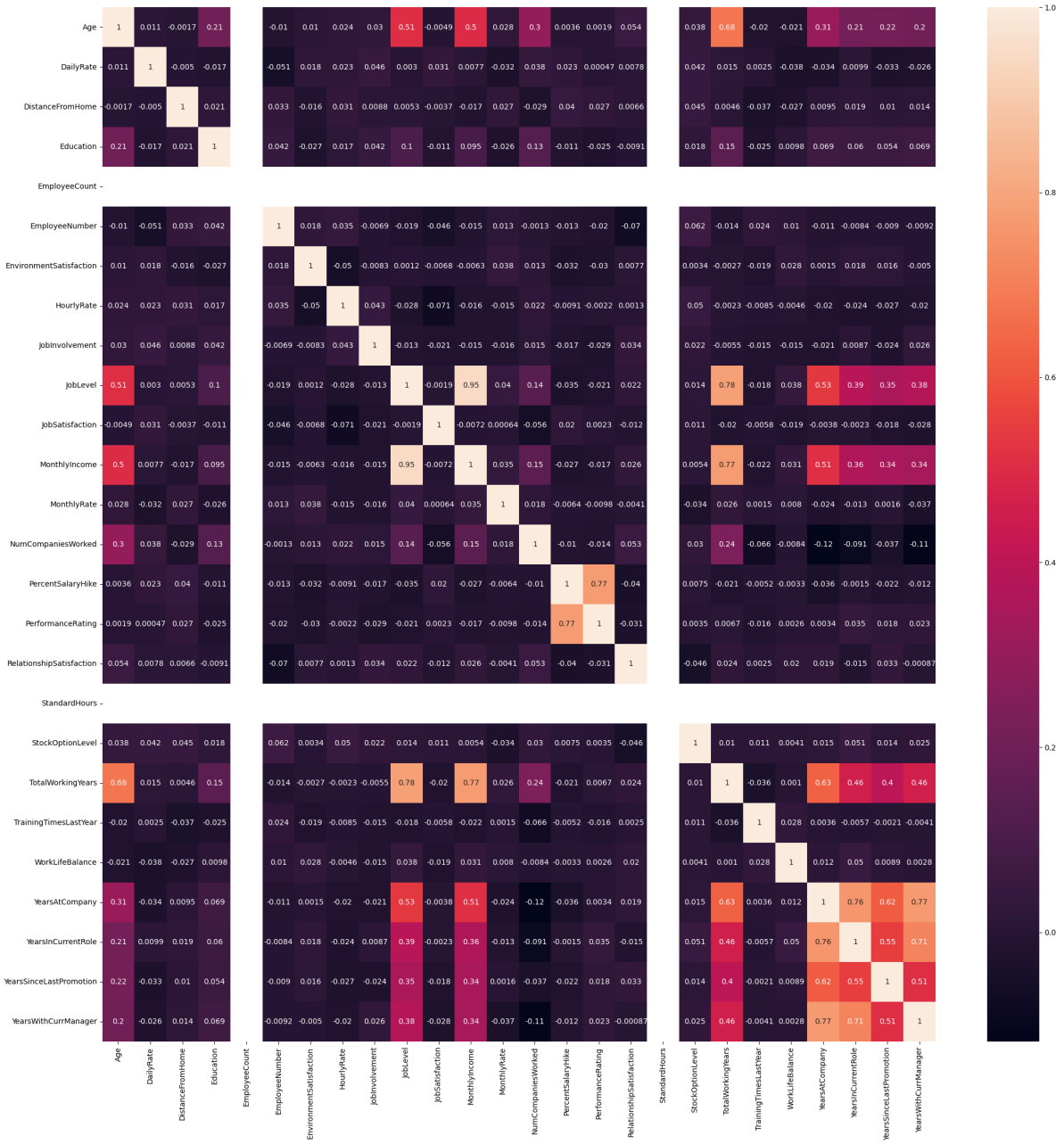
Out[6]:
```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
MonthlyRate                 0
NumCompaniesWorked          0
Over18                      0
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StandardHours               0
StockOptionLevel            0
TotalWorkingYears           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

In [7]:
```
cor = data.corr()
```

```
C:\Users\karth\AppData\Local\Temp\ipykernel_37264\2763113851.py:1: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future ver
sion, it will default to False. Select only valid columns or specify the value of
numeric_only to silence this warning.
  cor = data.corr()
```

In [8]:
```
fig, ax = plt.subplots(figsize=(25,25))
sns.heatmap(cor, annot=True)
```

Out[8]:
```
<Axes: >
```



In [9]:
```
sns.pairplot(data)
```

Out[9]:
```
<seaborn.axisgrid.PairGrid at 0x2402822bb10>
```

In [10]:
```
from sklearn.preprocessing import LabelEncoder
```

In [11]:
```
le=LabelEncoder()
```

In [12]:
```
data["BusinessTravel"]=le.fit_transform(data["BusinessTravel"])
```

In [13]:
```
data["Department"]=le.fit_transform(data["Department"])
```

In [14]:
```
data["EducationField"]=le.fit_transform(data["EducationField"])
```

In [15]:
```
data["Gender"]=le.fit_transform(data["Gender"])
```

In [16]:
```
data["JobRole"]=le.fit_transform(data["JobRole"])
```

In [17]:
```
data["MaritalStatus"]=le.fit_transform(data["MaritalStatus"])
```

In [18]:
```
data["Over18"]=le.fit_transform(data["Over18"])
```

In [19]:
```
data["OverTime"]=le.fit_transform(data["OverTime"])
```

In [20]:
```
data.head()
```

Out[20]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Educatio
0	41	Yes	2	1102	2	1	2	
1	49	No	1	279	1	8	1	
2	37	Yes	2	1373	1	2	2	
3	33	No	1	1392	1	3	4	
4	27	No	2	591	1	2	1	

5 rows × 35 columns

In [21]:
```
data.tail()
```

Out[21]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Edu
1465	36	No	1	884	1	23	2	
1466	39	No	2	613	1	6	1	
1467	27	No	2	155	1	4	3	
1468	49	No	1	1023	2	2	3	
1469	34	No	2	628	1	8	3	

5 rows × 35 columns

In [22]:
```
X=data.drop(columns=["EmployeeNumber","EmployeeCount","StandardHours","Attrition",
```

In [23]:
```
y=data["Attrition"]
```

In [24]:
```
from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
```

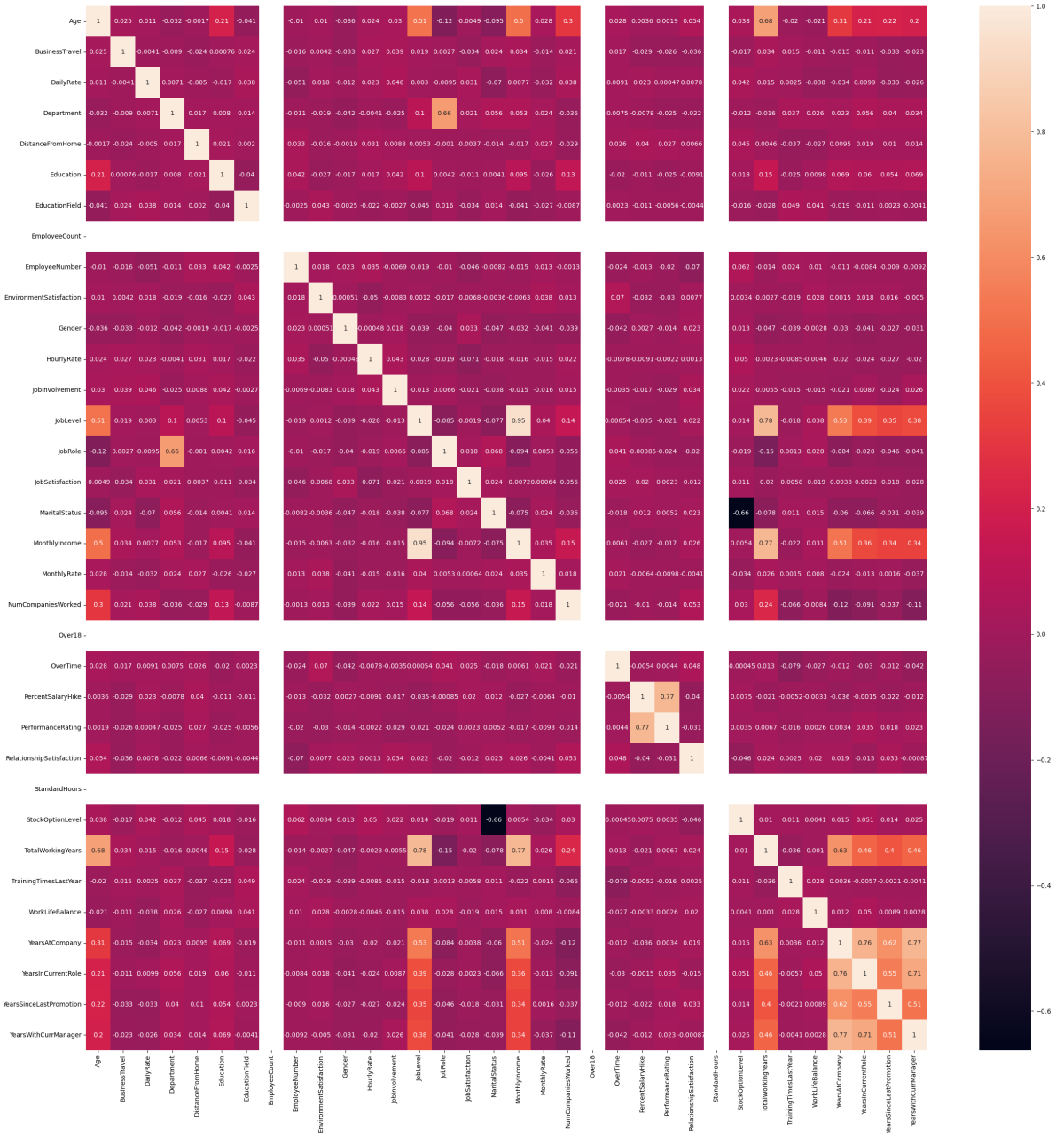
In [25]:
```
X_Scaled=ms.fit_transform(X)
```

In [26]:
```
cor=data.corr()
```

```
C:\Users\karth\AppData\Local\Temp\ipykernel_37264\1426905697.py:1: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future ver
sion, it will default to False. Select only valid columns or specify the value of
numeric_only to silence this warning.
  cor=data.corr()
```

In [27]:
```
fig, ax = plt.subplots(figsize=(30,30))
sns.heatmap(cor, annot=True)
```

Out[27]:
```
<Axes: >
```



In [28]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(X_Scaled,y,test_size =0.2,random_
```

In [29]:
```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state=0)
classifier.fit(x_train,y_train)
```

Out[29]:
```
LogisticRegression(random_state=0)
```

In [30]:
```
from sklearn.metrics import accuracy_score,confusion_matrix
y_pred = classifier.predict(x_test)
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)*100
```

Out[30]:
```
[[242   3]
 [ 32  17]]
88.09523809523809
```

In [31]:
```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,
```

In [32]:
```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

          No       0.88      0.99      0.93       245
         Yes       0.85      0.35      0.49        49

    accuracy                           0.88       294
   macro avg       0.87      0.67      0.71       294
weighted avg       0.88      0.88      0.86       294
```

In [33]:
```
from sklearn.tree import DecisionTreeClassifier
dtc=DecisionTreeClassifier()
```

In [34]:
```
dtc.fit(x_train,y_train)
```

Out[34]:
```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

In [35]:
```
from sklearn.metrics import accuracy_score,confusion_matrix
y_pred = dtc.predict(x_test)
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)*100
```

Out[35]:
```
[[202  43]
 [ 35  14]]
73.46938775510205
```

In [36]:
```
from sklearn import tree
plt.figure(figsize=(25,15))
tree.plot_tree(dtc,filled=True)
```

Out[36]:
```
[Text(0.32216075922818793, 0.9722222222222222, 'x[23] <= 0.038\ngini = 0.269\nsamples = 1176\nvalue = [988, 188]'),
 Text(0.0738255033557047, 0.9166666666666666, 'x[14] <= 0.75\ngini = 0.5\nsamples = 78\nvalue = [39, 39]'),
 Text(0.0436241610738255, 0.8611111111111112, 'x[4] <= 0.554\ngini = 0.426\nsamples = 39\nvalue = [27, 12]'),
 Text(0.026845637583892617, 0.8055555555555556, 'x[13] <= 0.167\ngini = 0.312\nsamples = 31\nvalue = [25, 6]'),
 Text(0.013422818791946308, 0.75, 'x[15] <= 0.057\ngini = 0.49\nsamples = 7\nvalue = [3, 4]'),
 Text(0.006711409395973154, 0.6944444444444444, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.020134228187919462, 0.6944444444444444, 'x[4] <= 0.018\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),
 Text(0.013422818791946308, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.026845637583892617, 0.6388888888888888, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.040268456375838924, 0.75, 'x[17] <= 0.056\ngini = 0.153\nsamples = 24\nvalue = [22, 2]'),
 Text(0.03355704697986577, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.04697986577181208, 0.6944444444444444, 'x[7] <= 0.167\ngini = 0.083\nsamples = 23\nvalue = [22, 1]'),
 Text(0.040268456375838924, 0.6388888888888888, 'x[4] <= 0.036\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.03355704697986577, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.04697986577181208, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.053691275167785234, 0.6388888888888888, 'gini = 0.0\nsamples = 21\nvalue = [21, 0]'),
 Text(0.06040268456375839, 0.8055555555555556, 'x[19] <= 0.679\ngini = 0.375\nsamples = 8\nvalue = [2, 6]'),
 Text(0.053691275167785234, 0.75, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.06711409395973154, 0.75, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.1040268456375839, 0.8611111111111112, 'x[9] <= 0.364\ngini = 0.426\nsamples = 39\nvalue = [12, 27]'),
 Text(0.087248322147651, 0.8055555555555556, 'x[0] <= 0.369\ngini = 0.133\nsamples = 14\nvalue = [1, 13]'),
 Text(0.08053691275167785, 0.75, 'gini = 0.0\nsamples = 13\nvalue = [0, 13]'),
 Text(0.09395973154362416, 0.75, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.12080536912751678, 0.8055555555555556, 'x[18] <= 0.5\ngini = 0.493\nsamples = 25\nvalue = [11, 14]'),
 Text(0.10738255033557047, 0.75, 'x[2] <= 0.106\ngini = 0.484\nsamples = 17\nvalue = [10, 7]'),
 Text(0.10067114093959731, 0.6944444444444444, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.11409395973154363, 0.6944444444444444, 'x[21] <= 0.167\ngini = 0.408\nsamples = 14\nvalue = [10, 4]'),
 Text(0.10067114093959731, 0.6388888888888888, 'x[23] <= 0.013\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.09395973154362416, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.10738255033557047, 0.5833333333333334, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.12751677852348994, 0.6388888888888888, 'x[3] <= 0.75\ngini = 0.18\nsamples = 10\nvalue = [9, 1]'),
 Text(0.12080536912751678, 0.5833333333333334, 'gini = 0.0\nsamples = 9\nvalue = [9, 0]'),
 Text(0.1342281879194631, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.1342281879194631, 0.75, 'x[29] <= 0.029\ngini = 0.219\nsamples = 8\nvalue = [1, 7]'),
```

```
 Text(0.12751677852348994, 0.6944444444444444, 'gini = 0.0\nsamples = 7\nvalue =
[0, 7]'),
 Text(0.14093959731543623, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.5704960151006712, 0.9166666666666666, 'x[18] <= 0.5\ngini = 0.235\nsamples
= 1098\nvalue = [949, 149]'),
 Text(0.3188443791946309, 0.8611111111111112, 'x[25] <= 0.167\ngini = 0.162\nsampl
es = 798\nvalue = [727, 71]'),
 Text(0.18456375838926176, 0.8055555555555556, 'x[2] <= 0.747\ngini = 0.38\nsample
s = 47\nvalue = [35, 12]'),
 Text(0.17785234899328858, 0.75, 'x[10] <= 0.5\ngini = 0.463\nsamples = 33\nvalue
= [21, 12]'),
 Text(0.15436241610738255, 0.6944444444444444, 'x[4] <= 0.446\ngini = 0.42\nsample
s = 10\nvalue = [3, 7]'),
 Text(0.1476510067114094, 0.6388888888888888, 'gini = 0.0\nsamples = 6\nvalue =
[0, 6]'),
 Text(0.1610738255033557, 0.6388888888888888, 'x[29] <= 0.382\ngini = 0.375\nsampl
es = 4\nvalue = [3, 1]'),
 Text(0.15436241610738255, 0.5833333333333334, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
 Text(0.16778523489932887, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.20134228187919462, 0.6944444444444444, 'x[24] <= 0.583\ngini = 0.34\nsampl
es = 23\nvalue = [18, 5]'),
 Text(0.18791946308724833, 0.6388888888888888, 'x[28] <= 0.333\ngini = 0.117\nsamp
les = 16\nvalue = [15, 1]'),
 Text(0.18120805369127516, 0.5833333333333334, 'gini = 0.0\nsamples = 15\nvalue =
[15, 0]'),
 Text(0.19463087248322147, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.21476510067114093, 0.6388888888888888, 'x[29] <= 0.529\ngini = 0.49\nsampl
es = 7\nvalue = [3, 4]'),
 Text(0.2080536912751678, 0.5833333333333334, 'gini = 0.0\nsamples = 4\nvalue =
[0, 4]'),
 Text(0.2214765100671141, 0.5833333333333334, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
 Text(0.1912751677852349, 0.75, 'gini = 0.0\nsamples = 14\nvalue = [14, 0]'),
 Text(0.453125, 0.8055555555555556, 'x[26] <= 0.963\ngini = 0.145\nsamples = 751\n
value = [692, 59]'),
 Text(0.44641359060402686, 0.75, 'x[26] <= 0.113\ngini = 0.143\nsamples = 750\nval
ue = [692, 58]'),
 Text(0.3104026845637584, 0.6944444444444444, 'x[7] <= 0.167\ngini = 0.218\nsample
s = 257\nvalue = [225, 32]'),
 Text(0.25671140939597314, 0.6388888888888888, 'x[29] <= 0.147\ngini = 0.355\nsamp
les = 65\nvalue = [50, 15]'),
 Text(0.2348993288590604, 0.5833333333333334, 'x[29] <= 0.029\ngini = 0.303\nsampl
es = 59\nvalue = [48, 11]'),
 Text(0.21140939597315436, 0.5277777777777778, 'x[10] <= 0.5\ngini = 0.463\nsample
s = 22\nvalue = [14, 8]'),
 Text(0.19798657718120805, 0.4722222222222222, 'x[9] <= 0.179\ngini = 0.198\nsampl
es = 9\nvalue = [8, 1]'),
 Text(0.1912751677852349, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.20469798657718122, 0.4166666666666667, 'gini = 0.0\nsamples = 8\nvalue =
[8, 0]'),
 Text(0.22483221476510068, 0.4722222222222222, 'x[9] <= 0.4\ngini = 0.497\nsamples
= 13\nvalue = [6, 7]'),
 Text(0.2181208053691275, 0.4166666666666667, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0]'),
 Text(0.23154362416107382, 0.4166666666666667, 'x[4] <= 0.286\ngini = 0.346\nsampl
es = 9\nvalue = [2, 7]'),
 Text(0.22483221476510068, 0.3611111111111111, 'x[16] <= 0.591\ngini = 0.444\nsamp
les = 3\nvalue = [2, 1]'),
 Text(0.2181208053691275, 0.3055555555555556, 'gini = 0.0\nsamples = 2\nvalue =
```

```
[2, 0]'),
 Text(0.23154362416107382, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.23825503355704697, 0.3611111111111111, 'gini = 0.0\nsamples = 6\nvalue =
[0, 6]'),
 Text(0.25838926174496646, 0.5277777777777778, 'x[13] <= 0.167\ngini = 0.149\nsamp
les = 37\nvalue = [34, 3]'),
 Text(0.2516778523489933, 0.4722222222222222, 'x[26] <= 0.088\ngini = 0.5\nsamples
= 6\nvalue = [3, 3]'),
 Text(0.24496644295302014, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.25838926174496646, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
 Text(0.2651006711409396, 0.4722222222222222, 'gini = 0.0\nsamples = 31\nvalue =
[31, 0]'),
 Text(0.2785234899328859, 0.5833333333333334, 'x[21] <= 0.667\ngini = 0.444\nsampl
es = 6\nvalue = [2, 4]'),
 Text(0.27181208053691275, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.28523489932885904, 0.5277777777777778, 'x[16] <= 0.916\ngini = 0.444\nsamp
les = 3\nvalue = [2, 1]'),
 Text(0.2785234899328859, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.29194630872483224, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.3640939597315436, 0.6388888888888888, 'x[0] <= 0.321\ngini = 0.161\nsample
s = 192\nvalue = [175, 17]'),
 Text(0.31543624161073824, 0.5833333333333334, 'x[6] <= 0.1\ngini = 0.294\nsamples
= 67\nvalue = [55, 12]'),
 Text(0.3087248322147651, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.3221476510067114, 0.5277777777777778, 'x[25] <= 0.5\ngini = 0.26\nsamples
= 65\nvalue = [55, 10]'),
 Text(0.3053691275167785, 0.4722222222222222, 'x[9] <= 0.679\ngini = 0.469\nsample
s = 16\nvalue = [10, 6]'),
 Text(0.2986577181208054, 0.4166666666666667, 'x[4] <= 0.018\ngini = 0.444\nsample
s = 9\nvalue = [3, 6]'),
 Text(0.29194630872483224, 0.3611111111111111, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.3053691275167785, 0.3611111111111111, 'x[6] <= 0.4\ngini = 0.245\nsamples
= 7\nvalue = [1, 6]'),
 Text(0.2986577181208054, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.31208053691275167, 0.3055555555555556, 'gini = 0.0\nsamples = 6\nvalue =
[0, 6]'),
 Text(0.31208053691275167, 0.4166666666666667, 'gini = 0.0\nsamples = 7\nvalue =
[7, 0]'),
 Text(0.3389261744966443, 0.4722222222222222, 'x[2] <= 0.037\ngini = 0.15\nsamples
= 49\nvalue = [45, 4]'),
 Text(0.33221476510067116, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.34563758389261745, 0.4166666666666667, 'x[2] <= 0.938\ngini = 0.117\nsampl
es = 48\nvalue = [45, 3]'),
 Text(0.3389261744966443, 0.3611111111111111, 'x[5] <= 0.875\ngini = 0.081\nsample
s = 47\nvalue = [45, 2]'),
 Text(0.32550335570469796, 0.3055555555555556, 'x[10] <= 0.167\ngini = 0.043\nsamp
les = 45\nvalue = [44, 1]'),
 Text(0.3187919463087248, 0.25, 'x[8] <= 0.5\ngini = 0.444\nsamples = 3\nvalue =
[2, 1]'),
 Text(0.31208053691275167, 0.19444444444444445, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.32550335570469796, 0.19444444444444445, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.33221476510067116, 0.25, 'gini = 0.0\nsamples = 42\nvalue = [42, 0]'),
```

```
 Text(0.3523489932885906, 0.3055555555555556, 'x[9] <= 0.293\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.34563758389261745, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.35906040268456374, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3523489932885906, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.412751677852349, 0.5833333333333334, 'x[6] <= 0.9\ngini = 0.077\nsamples =
125\nvalue = [120, 5]'),
 Text(0.39932885906040266, 0.5277777777777778, 'x[0] <= 0.393\ngini = 0.05\nsample
s = 118\nvalue = [115, 3]'),
 Text(0.3926174496644295, 0.4722222222222222, 'x[2] <= 0.956\ngini = 0.185\nsample
s = 29\nvalue = [26, 3]'),
 Text(0.3859060402684564, 0.4166666666666667, 'x[10] <= 0.167\ngini = 0.133\nsampl
es = 28\nvalue = [26, 2]'),
 Text(0.3724832214765101, 0.3611111111111111, 'x[12] <= 0.938\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.36577181208053694, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.37919463087248323, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.39932885906040266, 0.3611111111111111, 'x[29] <= 0.147\ngini = 0.074\nsamp
les = 26\nvalue = [25, 1]'),
 Text(0.3926174496644295, 0.3055555555555556, 'gini = 0.0\nsamples = 24\nvalue =
[24, 0]'),
 Text(0.40604026845637586, 0.3055555555555556, 'x[17] <= 0.111\ngini = 0.5\nsample
s = 2\nvalue = [1, 1]'),
 Text(0.39932885906040266, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.412751677852349, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.39932885906040266, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.40604026845637586, 0.4722222222222222, 'gini = 0.0\nsamples = 89\nvalue =
[89, 0]'),
 Text(0.4261744966442953, 0.5277777777777778, 'x[9] <= 0.514\ngini = 0.408\nsample
s = 7\nvalue = [5, 2]'),
 Text(0.41946308724832215, 0.4722222222222222, 'x[27] <= 0.056\ngini = 0.444\nsamp
les = 3\nvalue = [1, 2]'),
 Text(0.412751677852349, 0.4166666666666667, 'gini = 0.0\nsamples = 2\nvalue = [0,
2]'),
 Text(0.4261744966442953, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.43288590604026844, 0.4722222222222222, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0]'),
 Text(0.5824244966442953, 0.6944444444444444, 'x[26] <= 0.787\ngini = 0.1\nsamples
= 493\nvalue = [467, 26]'),
 Text(0.5473993288590604, 0.6388888888888888, 'x[13] <= 0.5\ngini = 0.094\nsamples
= 486\nvalue = [462, 24]'),
 Text(0.4974832214765101, 0.5833333333333334, 'x[12] <= 0.938\ngini = 0.154\nsampl
es = 191\nvalue = [175, 16]'),
 Text(0.49077181208053694, 0.5277777777777778, 'x[16] <= 0.481\ngini = 0.145\nsamp
les = 190\nvalue = [175, 15]'),
 Text(0.4714765100671141, 0.4722222222222222, 'x[16] <= 0.47\ngini = 0.221\nsample
s = 95\nvalue = [83, 12]'),
 Text(0.46476510067114096, 0.4166666666666667, 'x[29] <= 0.794\ngini = 0.207\nsamp
les = 94\nvalue = [83, 11]'),
 Text(0.45805369127516776, 0.3611111111111111, 'x[5] <= 0.375\ngini = 0.192\nsampl
es = 93\nvalue = [83, 10]'),
 Text(0.43288590604026844, 0.3055555555555556, 'x[6] <= 0.9\ngini = 0.363\nsamples
= 21\nvalue = [16, 5]'),
 Text(0.4261744966442953, 0.25, 'x[15] <= 0.413\ngini = 0.266\nsamples = 19\nvalue
= [16, 3]'),
 Text(0.412751677852349, 0.19444444444444445, 'x[15] <= 0.141\ngini = 0.117\nsampl
es = 16\nvalue = [15, 1]'),
 Text(0.40604026845637586, 0.1388888888888889, 'x[13] <= 0.167\ngini = 0.5\nsample
s = 2\nvalue = [1, 1]'),
```

```
 Text(0.39932885906040266, 0.08333333333333333, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.412751677852349, 0.08333333333333333, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.41946308724832215, 0.1388888888888889, 'gini = 0.0\nsamples = 14\nvalue =
[14, 0]'),
 Text(0.4395973154362416, 0.19444444444444445, 'x[17] <= 0.278\ngini = 0.444\nsamp
les = 3\nvalue = [1, 2]'),
 Text(0.43288590604026844, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.4463087248322148, 0.1388888888888889, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.4395973154362416, 0.25, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.48322147651006714, 0.3055555555555556, 'x[27] <= 0.139\ngini = 0.129\nsamp
les = 72\nvalue = [67, 5]'),
 Text(0.4597315436241611, 0.25, 'x[29] <= 0.206\ngini = 0.444\nsamples = 6\nvalue
= [4, 2]'),
 Text(0.45302013422818793, 0.19444444444444445, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
 Text(0.4664429530201342, 0.19444444444444445, 'x[8] <= 0.5\ngini = 0.444\nsamples
= 3\nvalue = [1, 2]'),
 Text(0.4597315436241611, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.47315436241610737, 0.1388888888888889, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.5067114093959731, 0.25, 'x[2] <= 0.958\ngini = 0.087\nsamples = 66\nvalue
= [63, 3]'),
 Text(0.49328859060402686, 0.19444444444444445, 'x[24] <= 0.583\ngini = 0.061\nsam
ples = 64\nvalue = [62, 2]'),
 Text(0.4865771812080537, 0.1388888888888889, 'gini = 0.0\nsamples = 52\nvalue =
[52, 0]'),
 Text(0.5, 0.1388888888888889, 'x[3] <= 0.75\ngini = 0.278\nsamples = 12\nvalue =
[10, 2]'),
 Text(0.49328859060402686, 0.08333333333333333, 'gini = 0.0\nsamples = 9\nvalue =
[9, 0]'),
 Text(0.5067114093959731, 0.08333333333333333, 'x[15] <= 0.429\ngini = 0.444\nsamp
les = 3\nvalue = [1, 2]'),
 Text(0.5, 0.027777777777777776, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5134228187919463, 0.027777777777777776, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.5201342281879194, 0.19444444444444445, 'x[19] <= 0.286\ngini = 0.5\nsample
s = 2\nvalue = [1, 1]'),
 Text(0.5134228187919463, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.5268456375838926, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.4714765100671141, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.47818791946308725, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.5100671140939598, 0.4722222222222222, 'x[17] <= 0.5\ngini = 0.061\nsamples
= 95\nvalue = [92, 3]'),
 Text(0.5033557046979866, 0.4166666666666667, 'gini = 0.0\nsamples = 76\nvalue =
[76, 0]'),
 Text(0.5167785234899329, 0.4166666666666667, 'x[29] <= 0.088\ngini = 0.266\nsampl
es = 19\nvalue = [16, 3]'),
 Text(0.5033557046979866, 0.3611111111111111, 'x[6] <= 0.3\ngini = 0.444\nsamples
= 3\nvalue = [1, 2]'),
 Text(0.4966442953020134, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.5100671140939598, 0.3055555555555556, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.5302013422818792, 0.3611111111111111, 'x[15] <= 0.108\ngini = 0.117\nsampl
es = 16\nvalue = [15, 1]'),
```

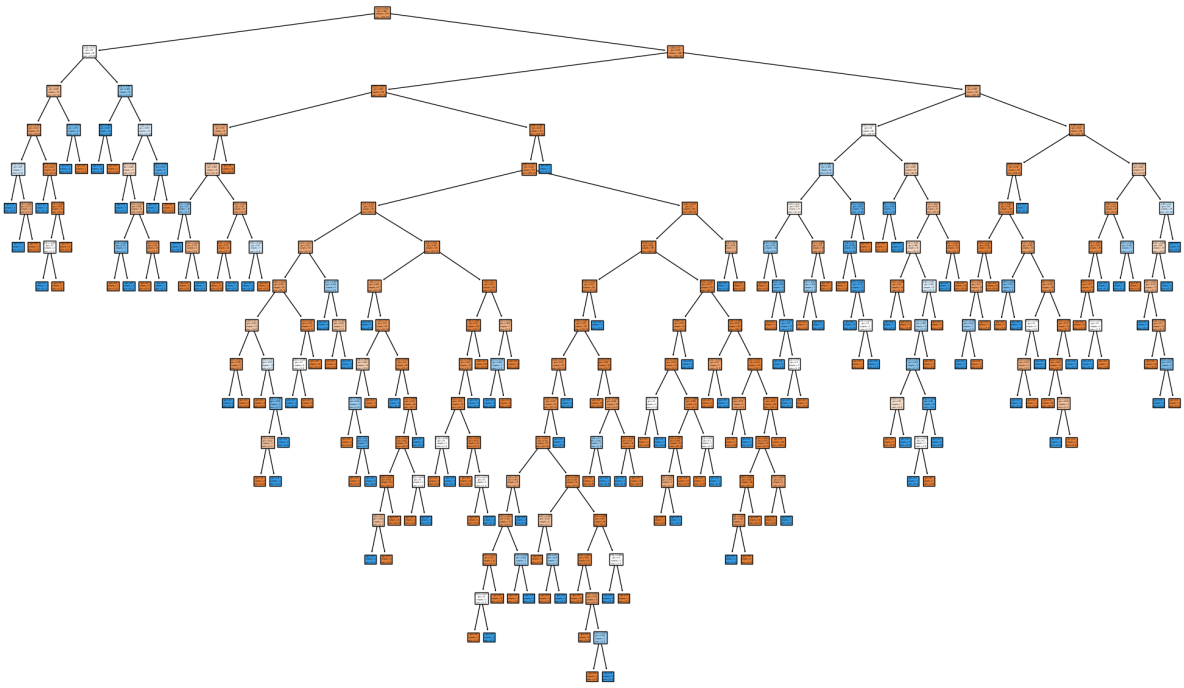
```
 Text(0.5234899328859061, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.5369127516778524, 0.3055555555555556, 'gini = 0.0\nsamples = 15\nvalue =
[15, 0]'),
 Text(0.5041946308724832, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.5973154362416108, 0.5833333333333334, 'x[19] <= 0.036\ngini = 0.053\nsampl
es = 295\nvalue = [287, 8]'),
 Text(0.5738255033557047, 0.5277777777777778, 'x[28] <= 0.7\ngini = 0.159\nsamples
= 46\nvalue = [42, 4]'),
 Text(0.5671140939597316, 0.4722222222222222, 'x[9] <= 0.071\ngini = 0.124\nsample
s = 45\nvalue = [42, 3]'),
 Text(0.5503355704697986, 0.4166666666666667, 'x[7] <= 0.5\ngini = 0.5\nsamples =
2\nvalue = [1, 1]'),
 Text(0.5436241610738255, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.5570469798657718, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.5838926174496645, 0.4166666666666667, 'x[23] <= 0.688\ngini = 0.089\nsampl
es = 43\nvalue = [41, 2]'),
 Text(0.5704697986577181, 0.3611111111111111, 'x[12] <= 0.062\ngini = 0.048\nsampl
es = 41\nvalue = [40, 1]'),
 Text(0.5637583892617449, 0.3055555555555556, 'x[0] <= 0.595\ngini = 0.375\nsample
s = 4\nvalue = [3, 1]'),
 Text(0.5570469798657718, 0.25, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.5704697986577181, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5771812080536913, 0.3055555555555556, 'gini = 0.0\nsamples = 37\nvalue =
[37, 0]'),
 Text(0.5973154362416108, 0.3611111111111111, 'x[3] <= 0.75\ngini = 0.5\nsamples =
2\nvalue = [1, 1]'),
 Text(0.5906040268456376, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.6040268456375839, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.5805369127516778, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.6208053691275168, 0.5277777777777778, 'x[15] <= 0.056\ngini = 0.032\nsampl
es = 249\nvalue = [245, 4]'),
 Text(0.6040268456375839, 0.4722222222222222, 'x[15] <= 0.054\ngini = 0.32\nsample
s = 5\nvalue = [4, 1]'),
 Text(0.5973154362416108, 0.4166666666666667, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0]'),
 Text(0.610738255033557, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.6375838926174496, 0.4722222222222222, 'x[2] <= 0.015\ngini = 0.024\nsample
s = 244\nvalue = [241, 3]'),
 Text(0.6241610738255033, 0.4166666666666667, 'x[19] <= 0.714\ngini = 0.278\nsampl
es = 6\nvalue = [5, 1]'),
 Text(0.6174496644295302, 0.3611111111111111, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]'),
 Text(0.6308724832214765, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.6510067114093959, 0.4166666666666667, 'x[21] <= 0.167\ngini = 0.017\nsampl
es = 238\nvalue = [236, 2]'),
 Text(0.6442953020134228, 0.3611111111111111, 'x[25] <= 0.833\ngini = 0.073\nsampl
es = 53\nvalue = [51, 2]'),
 Text(0.6308724832214765, 0.3055555555555556, 'x[29] <= 0.088\ngini = 0.041\nsampl
es = 48\nvalue = [47, 1]'),
 Text(0.6241610738255033, 0.25, 'x[0] <= 0.345\ngini = 0.245\nsamples = 7\nvalue =
[6, 1]'),
 Text(0.6174496644295302, 0.19444444444444445, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.6308724832214765, 0.19444444444444445, 'gini = 0.0\nsamples = 6\nvalue =
[6, 0]'),
```

```
 Text(0.6375838926174496, 0.25, 'gini = 0.0\nsamples = 41\nvalue = [41, 0]'),
 Text(0.6577181208053692, 0.3055555555555556, 'x[27] <= 0.417\ngini = 0.32\nsample
s = 5\nvalue = [4, 1]'),
 Text(0.6510067114093959, 0.25, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.6644295302013423, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6577181208053692, 0.3611111111111111, 'gini = 0.0\nsamples = 185\nvalue =
[185, 0]'),
 Text(0.6174496644295302, 0.6388888888888888, 'x[8] <= 0.5\ngini = 0.408\nsamples
= 7\nvalue = [5, 2]'),
 Text(0.610738255033557, 0.5833333333333334, 'gini = 0.0\nsamples = 2\nvalue = [0,
2]'),
 Text(0.6241610738255033, 0.5833333333333334, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]'),
 Text(0.45983640939597314, 0.75, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8221476510067114, 0.8611111111111112, 'x[15] <= 0.157\ngini = 0.385\nsampl
es = 300\nvalue = [222, 78]'),
 Text(0.7340604026845637, 0.8055555555555556, 'x[22] <= 0.167\ngini = 0.5\nsamples
= 96\nvalue = [49, 47]'),
 Text(0.697986577181208, 0.75, 'x[4] <= 0.161\ngini = 0.459\nsamples = 42\nvalue =
[15, 27]'),
 Text(0.6711409395973155, 0.6944444444444444, 'x[16] <= 0.41\ngini = 0.499\nsample
s = 23\nvalue = [12, 11]'),
 Text(0.6510067114093959, 0.6388888888888888, 'x[15] <= 0.061\ngini = 0.426\nsampl
es = 13\nvalue = [4, 9]'),
 Text(0.6442953020134228, 0.5833333333333334, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.6577181208053692, 0.5833333333333334, 'x[24] <= 0.25\ngini = 0.298\nsample
s = 11\nvalue = [2, 9]'),
 Text(0.6510067114093959, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.6644295302013423, 0.5277777777777778, 'x[19] <= 0.429\ngini = 0.18\nsample
s = 10\nvalue = [1, 9]'),
 Text(0.6577181208053692, 0.4722222222222222, 'gini = 0.0\nsamples = 8\nvalue =
[0, 8]'),
 Text(0.6711409395973155, 0.4722222222222222, 'x[1] <= 0.75\ngini = 0.5\nsamples =
2\nvalue = [1, 1]'),
 Text(0.6644295302013423, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.6778523489932886, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.6912751677852349, 0.6388888888888888, 'x[7] <= 0.167\ngini = 0.32\nsamples
= 10\nvalue = [8, 2]'),
 Text(0.6845637583892618, 0.5833333333333334, 'x[2] <= 0.544\ngini = 0.444\nsample
s = 3\nvalue = [1, 2]'),
 Text(0.6778523489932886, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.6912751677852349, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.697986577181208, 0.5833333333333334, 'gini = 0.0\nsamples = 7\nvalue = [7,
0]'),
 Text(0.7248322147651006, 0.6944444444444444, 'x[11] <= 0.125\ngini = 0.266\nsampl
es = 19\nvalue = [3, 16]'),
 Text(0.7181208053691275, 0.6388888888888888, 'x[9] <= 0.2\ngini = 0.198\nsamples
= 18\nvalue = [2, 16]'),
 Text(0.7114093959731543, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.7248322147651006, 0.5833333333333334, 'x[27] <= 0.306\ngini = 0.111\nsampl
es = 17\nvalue = [1, 16]'),
 Text(0.7181208053691275, 0.5277777777777778, 'gini = 0.0\nsamples = 15\nvalue =
[0, 15]'),
 Text(0.7315436241610739, 0.5277777777777778, 'x[29] <= 0.235\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.7248322147651006, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
```

```
 Text(0.738255033557047, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.7315436241610739, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.7701342281879194, 0.75, 'x[0] <= 0.202\ngini = 0.466\nsamples = 54\nvalue
= [34, 20]'),
 Text(0.7516778523489933, 0.6944444444444444, 'x[0] <= 0.107\ngini = 0.245\nsample
s = 7\nvalue = [1, 6]'),
 Text(0.7449664429530202, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.7583892617449665, 0.6388888888888888, 'gini = 0.0\nsamples = 6\nvalue =
[0, 6]'),
 Text(0.7885906040268457, 0.6944444444444444, 'x[2] <= 0.622\ngini = 0.418\nsample
s = 47\nvalue = [33, 14]'),
 Text(0.7718120805369127, 0.6388888888888888, 'x[2] <= 0.145\ngini = 0.482\nsample
s = 32\nvalue = [19, 13]'),
 Text(0.7583892617449665, 0.5833333333333334, 'x[15] <= 0.068\ngini = 0.18\nsample
s = 10\nvalue = [9, 1]'),
 Text(0.7516778523489933, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.7651006711409396, 0.5277777777777778, 'gini = 0.0\nsamples = 9\nvalue =
[9, 0]'),
 Text(0.785234899328859, 0.5833333333333334, 'x[16] <= 0.87\ngini = 0.496\nsamples
= 22\nvalue = [10, 12]'),
 Text(0.7785234899328859, 0.5277777777777778, 'x[25] <= 0.833\ngini = 0.465\nsampl
es = 19\nvalue = [7, 12]'),
 Text(0.7718120805369127, 0.4722222222222222, 'x[17] <= 0.167\ngini = 0.415\nsampl
es = 17\nvalue = [5, 12]'),
 Text(0.7583892617449665, 0.4166666666666667, 'x[19] <= 0.321\ngini = 0.49\nsample
s = 7\nvalue = [4, 3]'),
 Text(0.7516778523489933, 0.3611111111111111, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0]'),
 Text(0.7651006711409396, 0.3611111111111111, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.785234899328859, 0.4166666666666667, 'x[16] <= 0.235\ngini = 0.18\nsamples
= 10\nvalue = [1, 9]'),
 Text(0.7785234899328859, 0.3611111111111111, 'x[15] <= 0.092\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.7718120805369127, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.785234899328859, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.7919463087248322, 0.3611111111111111, 'gini = 0.0\nsamples = 8\nvalue =
[0, 8]'),
 Text(0.785234899328859, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue = [2,
0]'),
 Text(0.7919463087248322, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
 Text(0.8053691275167785, 0.6388888888888888, 'x[9] <= 0.064\ngini = 0.124\nsample
s = 15\nvalue = [14, 1]'),
 Text(0.7986577181208053, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.8120805369127517, 0.5833333333333334, 'gini = 0.0\nsamples = 14\nvalue =
[14, 0]'),
 Text(0.910234899328859, 0.8055555555555556, 'x[14] <= 0.75\ngini = 0.258\nsamples
= 204\nvalue = [173, 31]'),
 Text(0.8573825503355704, 0.75, 'x[15] <= 0.992\ngini = 0.138\nsamples = 147\nvalu
e = [136, 11]'),
 Text(0.8506711409395973, 0.6944444444444444, 'x[4] <= 0.482\ngini = 0.128\nsample
s = 146\nvalue = [136, 10]'),
 Text(0.8322147651006712, 0.6388888888888888, 'x[26] <= 0.063\ngini = 0.038\nsampl
es = 104\nvalue = [102, 2]'),
 Text(0.825503355704698, 0.5833333333333334, 'x[9] <= 0.193\ngini = 0.32\nsamples
= 10\nvalue = [8, 2]'),
```

```
 Text(0.8187919463087249, 0.5277777777777778, 'x[11] <= 0.625\ngini = 0.444\nsampl
es = 3\nvalue = [1, 2]'),
 Text(0.8120805369127517, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.825503355704698, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.8322147651006712, 0.5277777777777778, 'gini = 0.0\nsamples = 7\nvalue =
[7, 0]'),
 Text(0.8389261744966443, 0.5833333333333334, 'gini = 0.0\nsamples = 94\nvalue =
[94, 0]'),
 Text(0.8691275167785235, 0.6388888888888888, 'x[7] <= 0.167\ngini = 0.308\nsample
s = 42\nvalue = [34, 8]'),
 Text(0.8523489932885906, 0.5833333333333334, 'x[16] <= 0.194\ngini = 0.375\nsampl
es = 4\nvalue = [1, 3]'),
 Text(0.8456375838926175, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.8590604026845637, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.8859060402684564, 0.5833333333333334, 'x[0] <= 0.393\ngini = 0.229\nsample
s = 38\nvalue = [33, 5]'),
 Text(0.87248322147651, 0.5277777777777778, 'x[11] <= 0.375\ngini = 0.5\nsamples =
6\nvalue = [3, 3]'),
 Text(0.8657718120805369, 0.4722222222222222, 'x[10] <= 0.333\ngini = 0.375\nsampl
es = 4\nvalue = [3, 1]'),
 Text(0.8590604026845637, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.87248322147651, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue = [3,
0]'),
 Text(0.8791946308724832, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.8993288590604027, 0.5277777777777778, 'x[24] <= 0.917\ngini = 0.117\nsampl
es = 32\nvalue = [30, 2]'),
 Text(0.8926174496644296, 0.4722222222222222, 'x[12] <= 0.812\ngini = 0.062\nsampl
es = 31\nvalue = [30, 1]'),
 Text(0.8859060402684564, 0.4166666666666667, 'gini = 0.0\nsamples = 28\nvalue =
[28, 0]'),
 Text(0.8993288590604027, 0.4166666666666667, 'x[2] <= 0.561\ngini = 0.444\nsample
s = 3\nvalue = [2, 1]'),
 Text(0.8926174496644296, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.9060402684563759, 0.3611111111111111, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.9060402684563759, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.8640939597315436, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.9630872483221476, 0.75, 'x[12] <= 0.812\ngini = 0.456\nsamples = 57\nvalue
= [37, 20]'),
 Text(0.9395973154362416, 0.6944444444444444, 'x[28] <= 0.4\ngini = 0.238\nsamples
= 29\nvalue = [25, 4]'),
 Text(0.9261744966442953, 0.6388888888888888, 'x[9] <= 0.964\ngini = 0.142\nsample
s = 26\nvalue = [24, 2]'),
 Text(0.9194630872483222, 0.5833333333333334, 'x[19] <= 0.75\ngini = 0.077\nsample
s = 25\nvalue = [24, 1]'),
 Text(0.912751677852349, 0.5277777777777778, 'gini = 0.0\nsamples = 23\nvalue = [2
3, 0]'),
 Text(0.9261744966442953, 0.5277777777777778, 'x[17] <= 0.333\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.9194630872483222, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.9328859060402684, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.9328859060402684, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
```

```
 Text(0.9530201342281879, 0.6388888888888888, 'x[25] <= 0.833\ngini = 0.444\nsampl
es = 3\nvalue = [1, 2]'),
 Text(0.9463087248322147, 0.5833333333333334, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.959731543624161, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.9865771812080537, 0.6944444444444444, 'x[28] <= 0.1\ngini = 0.49\nsamples
= 28\nvalue = [12, 16]'),
 Text(0.9798657718120806, 0.6388888888888888, 'x[10] <= 0.833\ngini = 0.48\nsample
s = 20\nvalue = [12, 8]'),
 Text(0.9731543624161074, 0.5833333333333334, 'x[4] <= 0.018\ngini = 0.415\nsample
s = 17\nvalue = [12, 5]'),
 Text(0.9664429530201343, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.9798657718120806, 0.5277777777777778, 'x[15] <= 0.365\ngini = 0.32\nsample
s = 15\nvalue = [12, 3]'),
 Text(0.9731543624161074, 0.4722222222222222, 'gini = 0.0\nsamples = 11\nvalue =
[11, 0]'),
 Text(0.9865771812080537, 0.4722222222222222, 'x[23] <= 0.325\ngini = 0.375\nsampl
es = 4\nvalue = [1, 3]'),
 Text(0.9798657718120806, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.9932885906040269, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.9865771812080537, 0.5833333333333334, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.9932885906040269, 0.6388888888888888, 'gini = 0.0\nsamples = 8\nvalue =
[0, 8]')]
```



In [37]:
```
from sklearn.model_selection import GridSearchCV
parameter={
 'criterion':['gini','entropy'],
  'splitter':['best','random'],
  'max_depth':[1,2,3,4,5,6,7,8,9,10],
  'max_features':['auto', 'sqrt', 'log2']

}
```

In [38]:
```
grid_search=GridSearchCV(estimator=dtc,param_grid=parameter,cv=5,scoring="accuracy
```

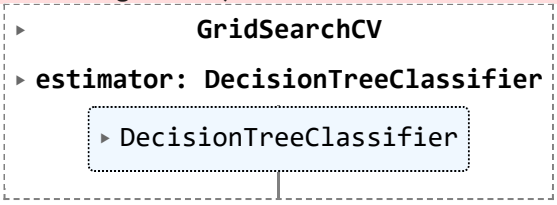
In [39]:
```
grid_search.fit(x_train,y_train)
```

```
C:\Users\karth\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py:
425: FitFailedWarning:
200 fits failed out of a total of 600.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_sco
re='raise'.

Below are more details about the failures:
--------------------------------------------------------------------------------
200 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\karth\anaconda3\Lib\site-packages\sklearn\model_selection\_valida
tion.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\karth\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in
wrapper
    estimator._validate_params()
  File "C:\Users\karth\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in
_validate_params
    validate_parameter_constraints(
  File "C:\Users\karth\anaconda3\Lib\site-packages\sklearn\utils\_param_validatio
n.py", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'max_features' paramete
r of DecisionTreeClassifier must be an int in the range [1, inf), a float in the r
ange (0.0, 1.0], a str among {'log2', 'sqrt'} or None. Got 'auto' instead.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\Users\karth\anaconda3\Lib\site-packages\sklearn\model_selection\_search.py:976:
UserWarning: One or more of the test scores are non-finite: [       nan        nan
0.83843491 0.84013704 0.83588172 0.84013704
        nan        nan 0.84013704 0.84013704 0.84522178 0.83928597
        nan        nan 0.84948071 0.83758384 0.8409881  0.83504147
        nan        nan 0.83588172 0.83757303 0.83503426 0.84863325
        nan        nan 0.82225748 0.82993509 0.83501623 0.83587811
        nan        nan 0.82056978 0.83332853 0.81799495 0.83503426
        nan        nan 0.8112189  0.83249189 0.80100613 0.83419762
        nan        nan 0.80531554 0.83417959 0.81717274 0.83161197
        nan        nan 0.79508114 0.82056257 0.8069744  0.81295709
        nan        nan 0.76698161 0.79507393 0.77379012 0.79935088
        nan        nan 0.84013704 0.84013704 0.84013704 0.84013704
        nan        nan 0.83417959 0.83843491 0.83758384 0.84013704
        nan        nan 0.82736747 0.84013704 0.84268301 0.84269023
        nan        nan 0.83246664 0.8409881  0.83419401 0.83927515
        nan        nan 0.82991345 0.82994591 0.82990624 0.83673999
        nan        nan 0.81887126 0.82992788 0.84440317 0.82822935
        nan        nan 0.82143166 0.83417959 0.8205842  0.83928237
        nan        nan 0.8205842  0.82231158 0.82479986 0.8248251
        nan        nan 0.7976163  0.81892896 0.80102056 0.82568698
        nan        nan 0.79674721 0.80101334 0.80105301 0.80695276]
  warnings.warn(
```

Out[39]:



In [40]:
```
grid_search.best_params_
```

Out[40]:
```
{'criterion': 'gini',
 'max_depth': 3,
 'max_features': 'sqrt',
 'splitter': 'best'}
```

In [41]:
```
dtc_cv=DecisionTreeClassifier(criterion= 'entropy',
 max_depth= 4,
 max_features= 'sqrt',
 splitter= 'best')
dtc_cv.fit(x_train,y_train)
```

Out[41]:

DecisionTreeClassifier

```
DecisionTreeClassifier(criterion='entropy', max_depth=4, max_features='sq
rt')
```

In [42]:
```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

          No       0.85      0.82      0.84       245
         Yes       0.25      0.29      0.26        49

    accuracy                           0.73       294
   macro avg       0.55      0.56      0.55       294
weighted avg       0.75      0.73      0.74       294
```

In [43]:
```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 1000, criterion = 'entropy', ra
classifier.fit(x_train, y_train)
```

Out[43]:

RandomForestClassifier

```
RandomForestClassifier(criterion='entropy', n_estimators=1000, random_sta
te=0)
```

In [44]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
y_pred = classifier.predict(x_test)
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

Out[44]:
```
[[243   2]
 [ 41   8]]
0.8537414965986394
```

In [45]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [46]:
```
rfc=RandomForestClassifier()
```

In [47]:
```
forest_params = [{'max_depth': list(range(10, 15)), 'max_features': list(range(0,1
```

In [48]:
```
rfc_cv=GridSearchCV(rfc,param_grid=forest_params,cv=10,scoring="accuracy")
```

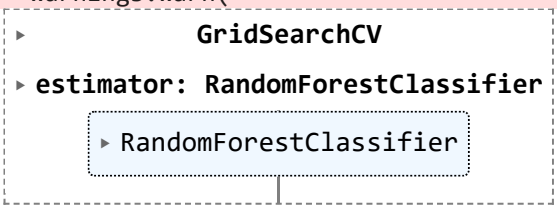
In [49]:
```
rfc_cv.fit(x_train,y_train)
```

```
C:\Users\karth\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py:
425: FitFailedWarning:
50 fits failed out of a total of 700.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_sco
re='raise'.

Below are more details about the failures:
--------------------------------------------------------------------------------
50 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\karth\anaconda3\Lib\site-packages\sklearn\model_selection\_valida
tion.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\karth\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in
wrapper
    estimator._validate_params()
  File "C:\Users\karth\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in
_validate_params
    validate_parameter_constraints(
  File "C:\Users\karth\anaconda3\Lib\site-packages\sklearn\utils\_param_validatio
n.py", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'max_features' paramete
r of RandomForestClassifier must be an int in the range [1, inf), a float in the r
ange (0.0, 1.0], a str among {'log2', 'sqrt'} or None. Got 0 instead.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\Users\karth\anaconda3\Lib\site-packages\sklearn\model_selection\_search.py:976:
UserWarning: One or more of the test scores are non-finite: [       nan 0.84863827
0.85800377 0.85967695 0.86054614 0.86224105
 0.86479067 0.85714907 0.85713458 0.8613936  0.85881501 0.86221932
 0.85797479 0.86221932        nan 0.85118789 0.85630161 0.85714182
 0.85713458 0.85966971 0.86308127 0.85882225 0.8596842  0.86222657
 0.86050992 0.85881501 0.85882225 0.85541793        nan 0.84779806
 0.85713458 0.85544691 0.8622483  0.86055338 0.85966247 0.85712734
 0.85713458 0.8605389  0.86053165 0.86137911 0.86138635 0.85457048
        nan 0.84864552 0.85205708 0.85713458 0.86055338 0.85967695
 0.85969868 0.86137911 0.85625815 0.86223381 0.85966247 0.86051717
 0.85966971 0.8681805         nan 0.85204983 0.85543242 0.85967695
 0.86055338 0.85970592 0.85969144 0.85966971 0.85627264 0.85544691
 0.8613936  0.86053165 0.85966247 0.85797479]
  warnings.warn(
```

Out[49]:



In [50]:

```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

          No       0.86      0.99      0.92       245
         Yes       0.80      0.16      0.27        49

    accuracy                           0.85       294
   macro avg       0.83      0.58      0.59       294
weighted avg       0.85      0.85      0.81       294
```