NAME: P.NAGA BHAVANI

SLOT: MORNING SLOT(10-12)

CAMPUS: VIT AP

REGISTRATION NO.: 21BCE9043

ASSIGNMENT NO.: 4

email-id: bhavani.21bce9043@vitapstudent.ac.in

Assignment 22 sep

1.Download the Employee Attrition Dataset https://www.kaggle.com/datasets/patelprashant/employee-attrition 2.Perfrom Data Preprocessing 3.Model Building using Logistic Regression and Decision Tree 4.Calculate Performance metrics

IMPORT LIBRARIES

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

IMPORT DATASET

In [2]:
```
df=pd.read_csv("Employee-Attrition.csv")
```

In [3]:
```
df
```

Out[3]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromH
0	41	Yes	Travel_Rarely	1102	Sales	
1	49	No	Travel_Frequently	279	Research & Development	
2	37	Yes	Travel_Rarely	1373	Research & Development	
3	33	No	Travel_Frequently	1392	Research & Development	
4	27	No	Travel_Rarely	591	Research & Development	
...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research & Development	
1466	39	No	Travel_Rarely	613	Research & Development	
1467	27	No	Travel_Rarely	155	Research & Development	
1468	49	No	Travel_Frequently	1023	Sales	
1469	34	No	Travel_Rarely	628	Research & Development	

1470 rows × 35 columns

In [4]:
```
df.head()
```

Out[4]:

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

5 rows × 35 columns

In [5]:
```
df.tail()
```

Out[5]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromH
1465	36	No	Travel_Frequently	884	Research & Development	
1466	39	No	Travel_Rarely	613	Research & Development	
1467	27	No	Travel_Rarely	155	Research & Development	
1468	49	No	Travel_Frequently	1023	Sales	
1469	34	No	Travel_Rarely	628	Research & Development	

5 rows × 35 columns

```
In [6]: df.shape
```

```
Out[6]: (1470, 35)
```

```
In [7]: df.info()
```

Loading [MathJax]/extensions/Safe.js

```
<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 [8]:

```
df.describe()
```

Out[8]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeC
count	1470.000000	1470.000000	1470.000000	1470.000000	14
mean	36.923810	802.485714	9.192517	2.912925	
std	9.135373	403.509100	8.106864	1.024165	
min	18.000000	102.000000	1.000000	1.000000	
25%	30.000000	465.000000	2.000000	2.000000	
50%	36.000000	802.000000	7.000000	3.000000	
75%	43.000000	1157.000000	14.000000	4.000000	
max	60.000000	1499.000000	29.000000	5.000000	

8 rows × 26 columns

In [9]:

```
#Checking for Null Values.
df.isnull().any()
```

```
Out[9]: Age                         False
        Attrition                   False
        BusinessTravel              False
        DailyRate                   False
        Department                  False
        DistanceFromHome            False
        Education                   False
        EducationField              False
        EmployeeCount               False
        EmployeeNumber              False
        EnvironmentSatisfaction     False
        Gender                      False
        HourlyRate                  False
        JobInvolvement              False
        JobLevel                    False
        JobRole                     False
        JobSatisfaction             False
        MaritalStatus               False
        MonthlyIncome               False
        MonthlyRate                 False
        NumCompaniesWorked          False
        Over18                      False
        OverTime                    False
        PercentSalaryHike           False
        PerformanceRating           False
        RelationshipSatisfaction    False
        StandardHours               False
        StockOptionLevel            False
        TotalWorkingYears           False
        TrainingTimesLastYear       False
        WorkLifeBalance             False
        YearsAtCompany              False
        YearsInCurrentRole          False
        YearsSinceLastPromotion     False
        YearsWithCurrManager        False
        dtype: bool
```

```
In [10]: df.isnull().sum()
```

Out[10]:

```
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 [11]:

```
corr=df.corr()
corr
```

```
C:\Users\NAGA BHAVANI\AppData\Local\Temp\ipykernel_17540\3182140910.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  corr=df.corr()
```

Loading [MathJax]/extensions/Safe.js

Out[11]:

	Age	DailyRate	DistanceFromHome	Education
Age	1.000000	0.010661	-0.001686	0.208034
DailyRate	0.010661	1.000000	-0.004985	-0.016806
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042
Education	0.208034	-0.016806	0.021042	1.000000
EmployeeCount	NaN	NaN	NaN	NaN
EmployeeNumber	-0.010145	-0.050990	0.032916	0.042070
EnvironmentSatisfaction	0.010146	0.018355	-0.016075	-0.027128
HourlyRate	0.024287	0.023381	0.031131	0.016775
JobInvolvement	0.029820	0.046135	0.008783	0.042438
JobLevel	0.509604	0.002966	0.005303	0.101589
JobSatisfaction	-0.004892	0.030571	-0.003669	-0.011296
MonthlyIncome	0.497855	0.007707	-0.017014	0.094961
MonthlyRate	0.028051	-0.032182	0.027473	-0.026084
NumCompaniesWorked	0.299635	0.038153	-0.029251	0.126317
PercentSalaryHike	0.003634	0.022704	0.040235	-0.011111
PerformanceRating	0.001904	0.000473	0.027110	-0.024539
RelationshipSatisfaction	0.053535	0.007846	0.006557	-0.009118
StandardHours	NaN	NaN	NaN	NaN
StockOptionLevel	0.037510	0.042143	0.044872	0.018422
TotalWorkingYears	0.680381	0.014515	0.004628	0.148280
TrainingTimesLastYear	-0.019621	0.002453	-0.036942	-0.025100
WorkLifeBalance	-0.021490	-0.037848	-0.026556	0.009819
YearsAtCompany	0.311309	-0.034055	0.009508	0.069114
YearsInCurrentRole	0.212901	0.009932	0.018845	0.060236
YearsSinceLastPromotion	0.216513	-0.033229	0.010029	0.054254
YearsWithCurrManager	0.202089	-0.026363	0.014406	0.069065

26 rows × 26 columns

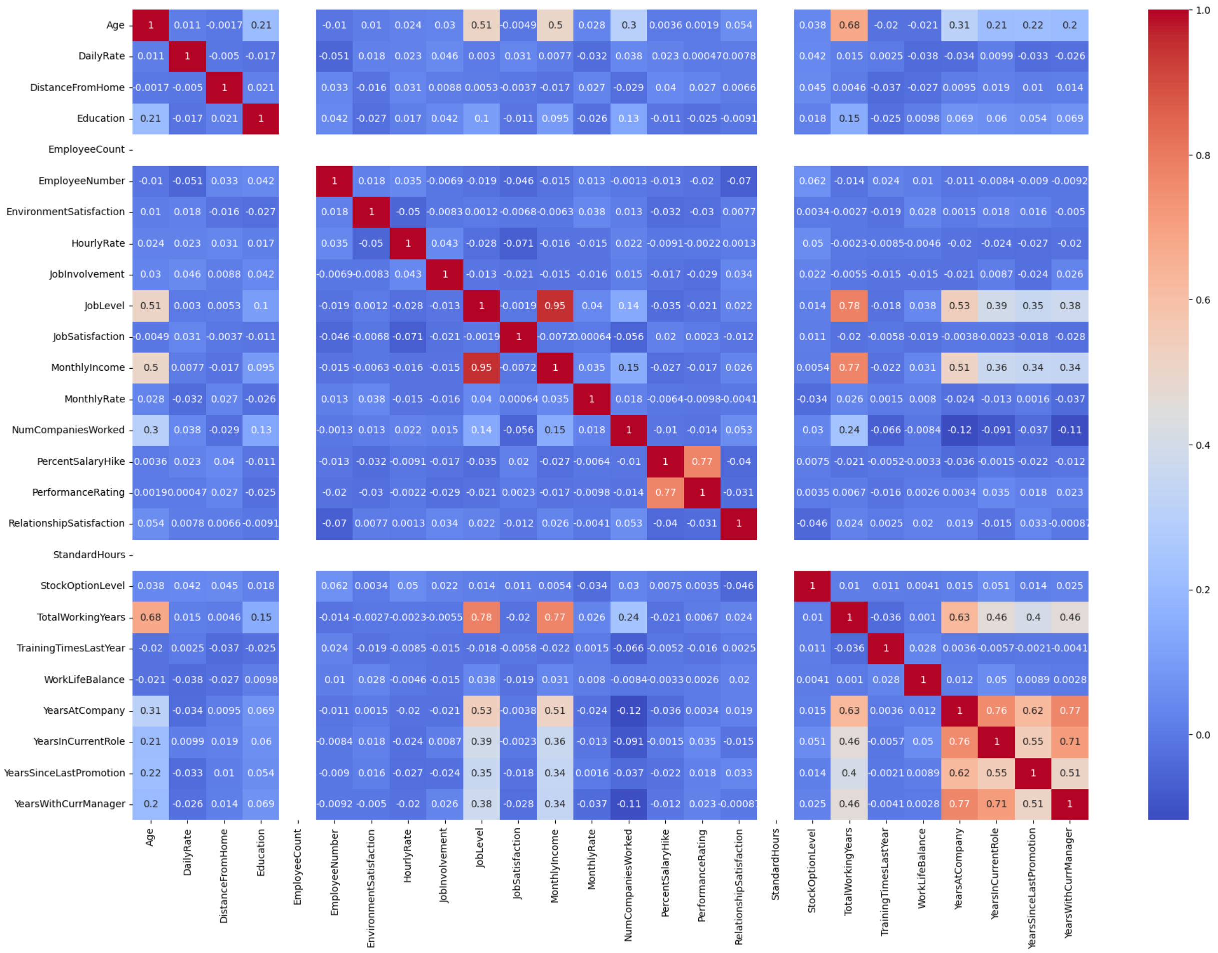
In [12]:

```
plt.subplots(figsize=(22,15))
sns.heatmap(corr,annot=True,cmap="coolwarm")
```

Out[12]:

```
<Axes: >
```

Loading [MathJax]/extensions/Safe.js



In [13]:
```
df.Attrition.value_counts()
```

Out[13]:
```
No      1233
Yes      237
Name: Attrition, dtype: int64
```

In [14]:
```
df.isnull().any()
```

Loading [MathJax]/extensions/Safe.js

Out[14]:

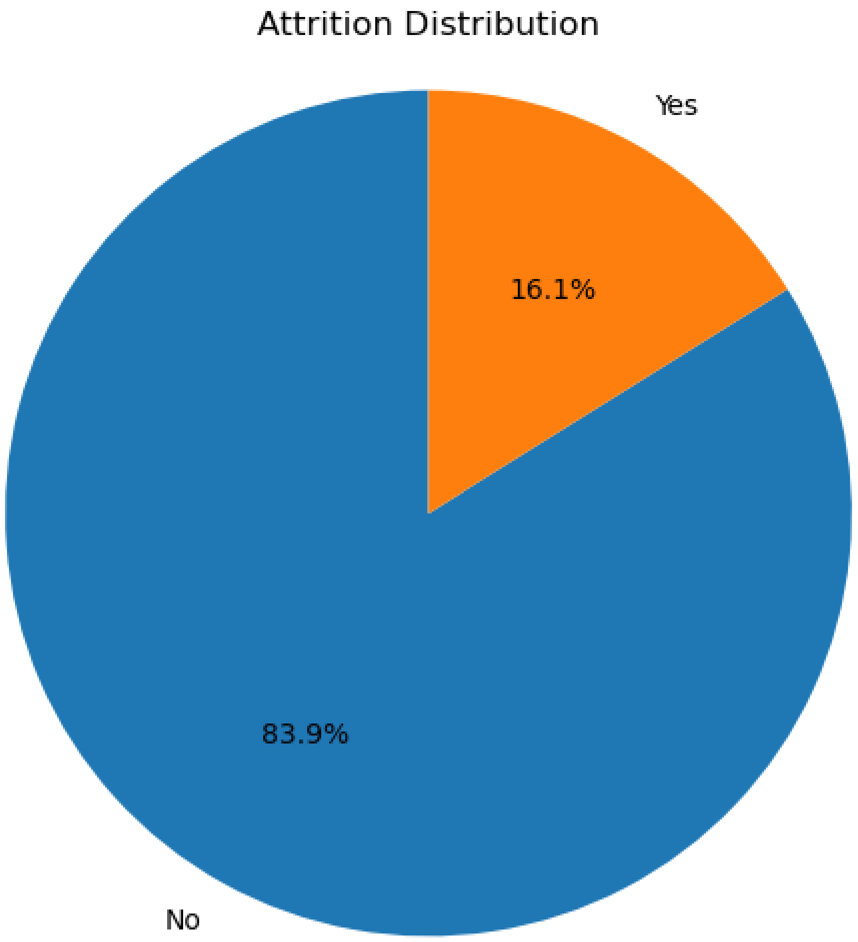
```
Age                         False
Attrition                   False
BusinessTravel              False
DailyRate                   False
Department                  False
DistanceFromHome            False
Education                   False
EducationField              False
EmployeeCount           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

DATA VISUALIZATION

In [15]:

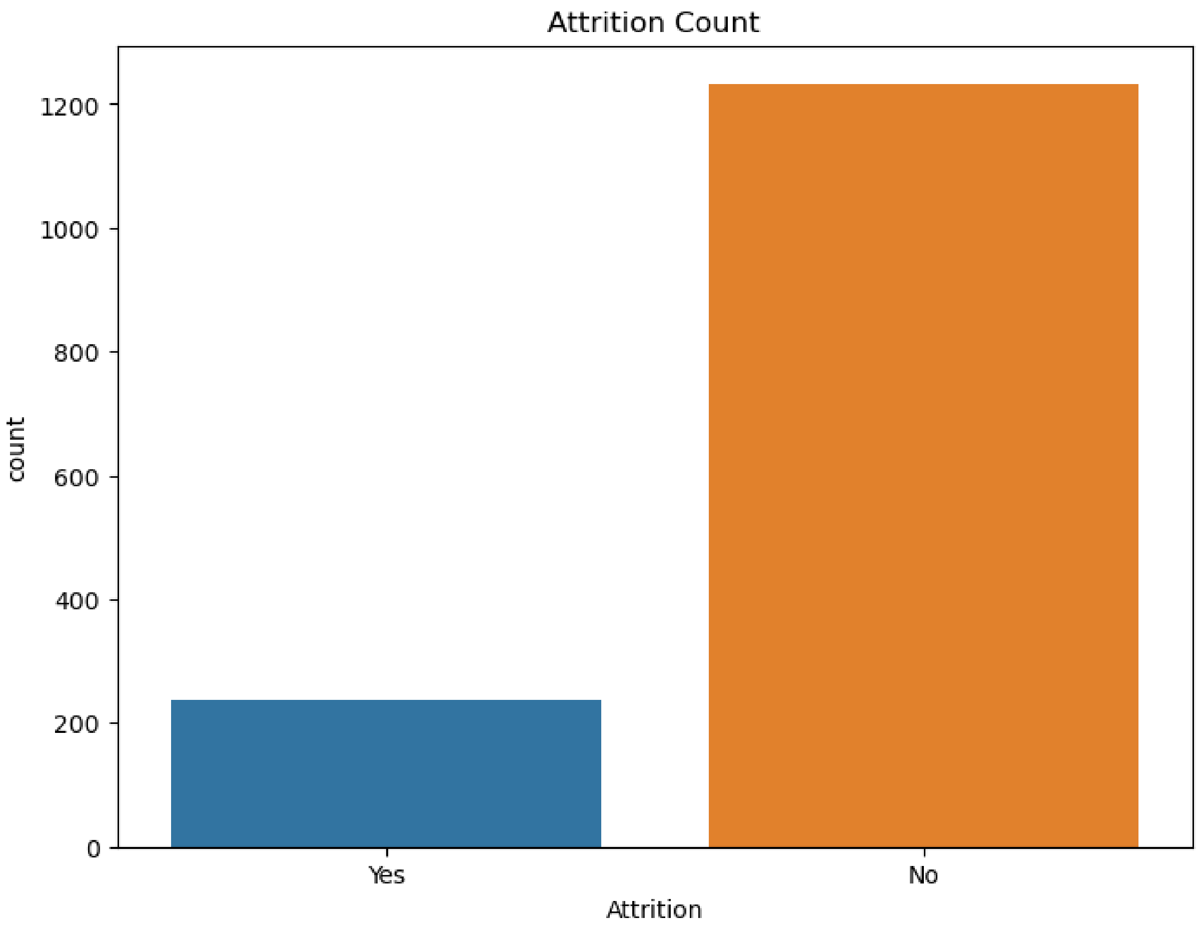
```
attrition_counts = df['Attrition'].value_counts()
plt.figure(figsize=(6, 6))
plt.pie(attrition_counts, labels=attrition_counts.index, autopct='%1.1f%%',
plt.title('Attrition Distribution')
plt.axis('equal')

plt.show()
```



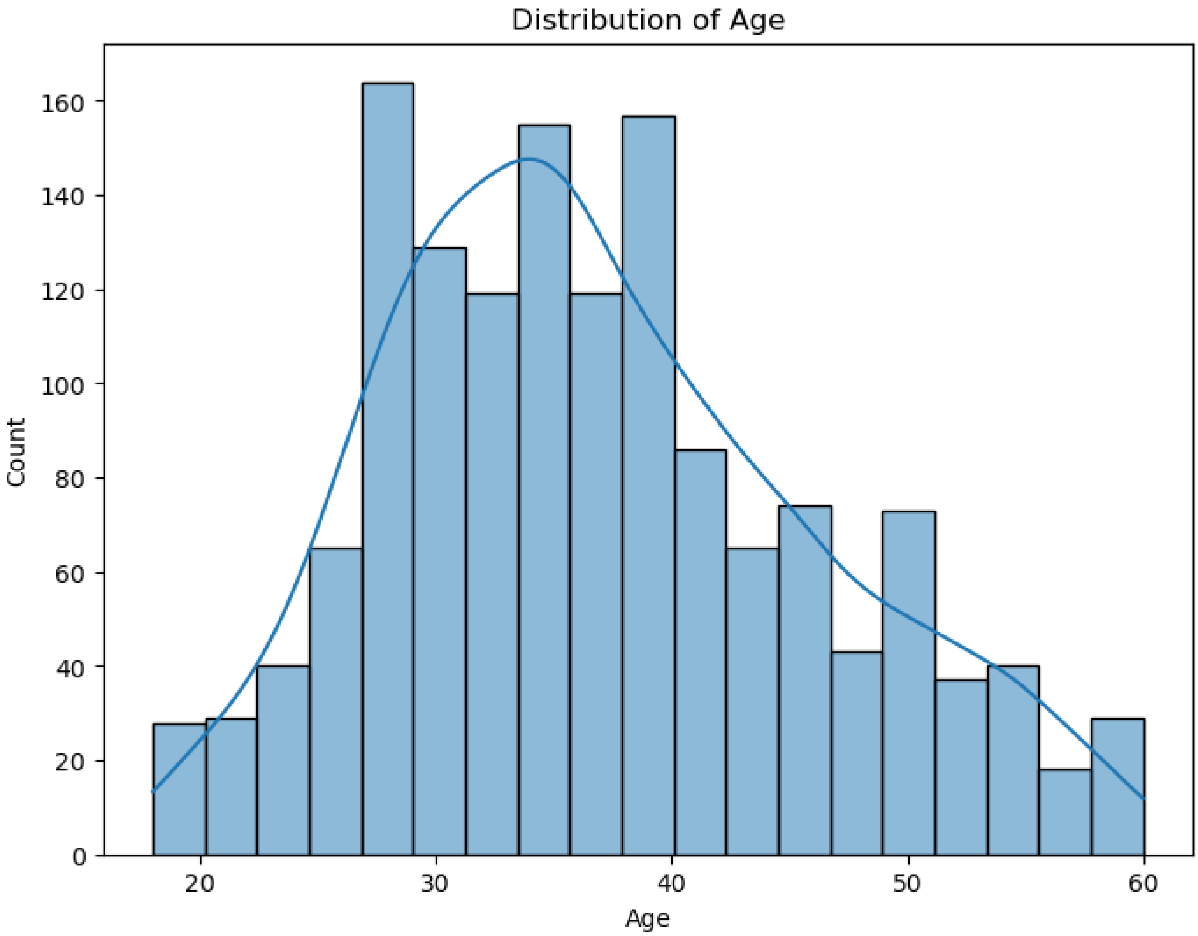
In [16]:

```
plt.figure(figsize=(8, 6))
sns.countplot(x="Attrition", data=df)
plt.title("Attrition Count")
plt.show()
```



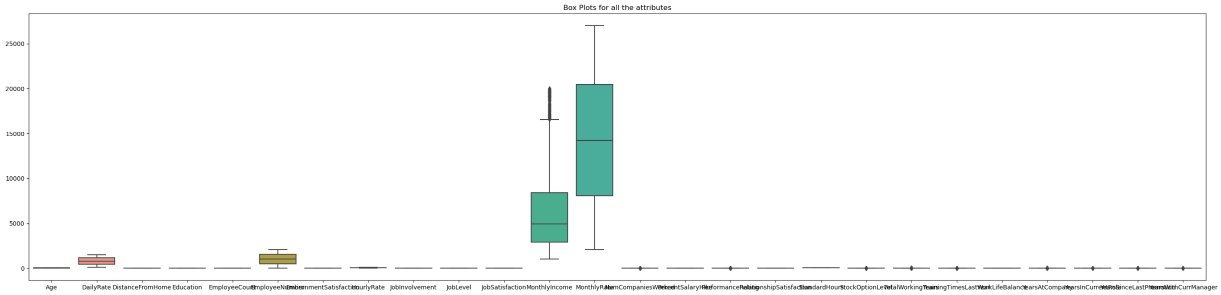
In [17]:

```
plt.figure(figsize=(8, 6))
sns.histplot(data=df, x="Age", kde=True)
plt.title("Distribution of Age")
plt.show()
```



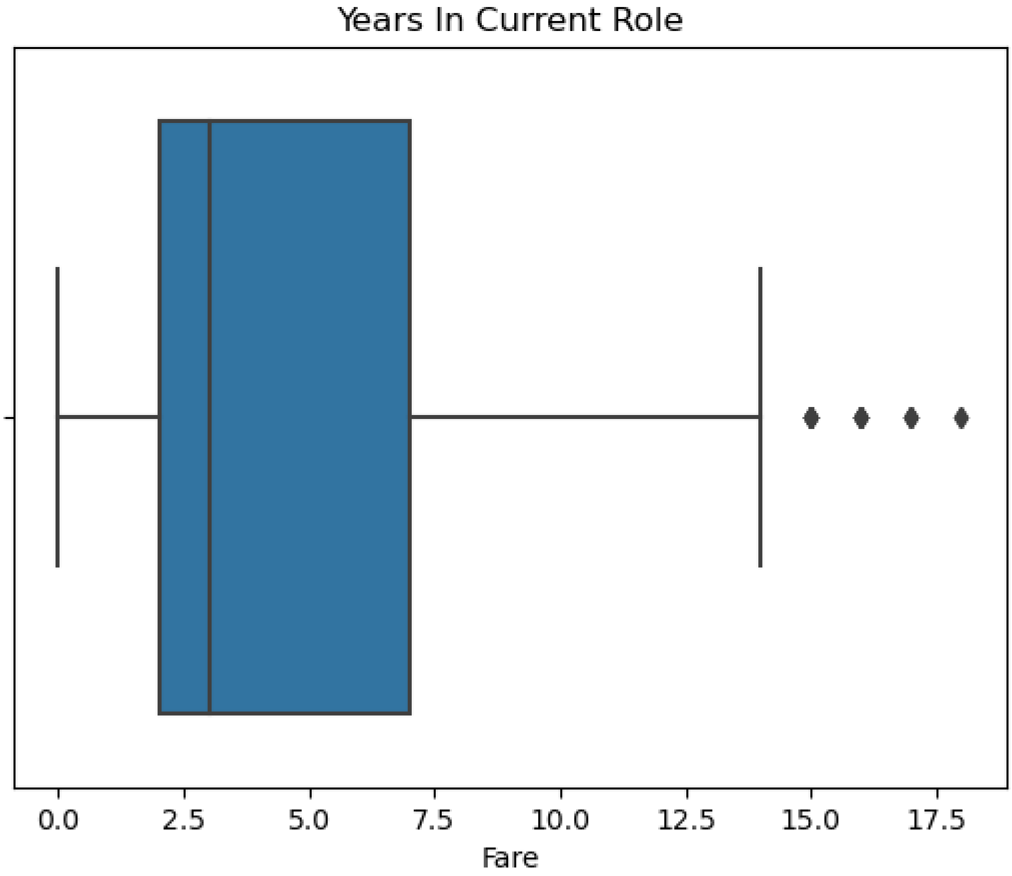
OUTLIER DETECTION

In [18]:
```
plt.figure(figsize=(35, 8))
sns.boxplot(data=df)
plt.title('Box Plots for all the attributes')
plt.show()
```



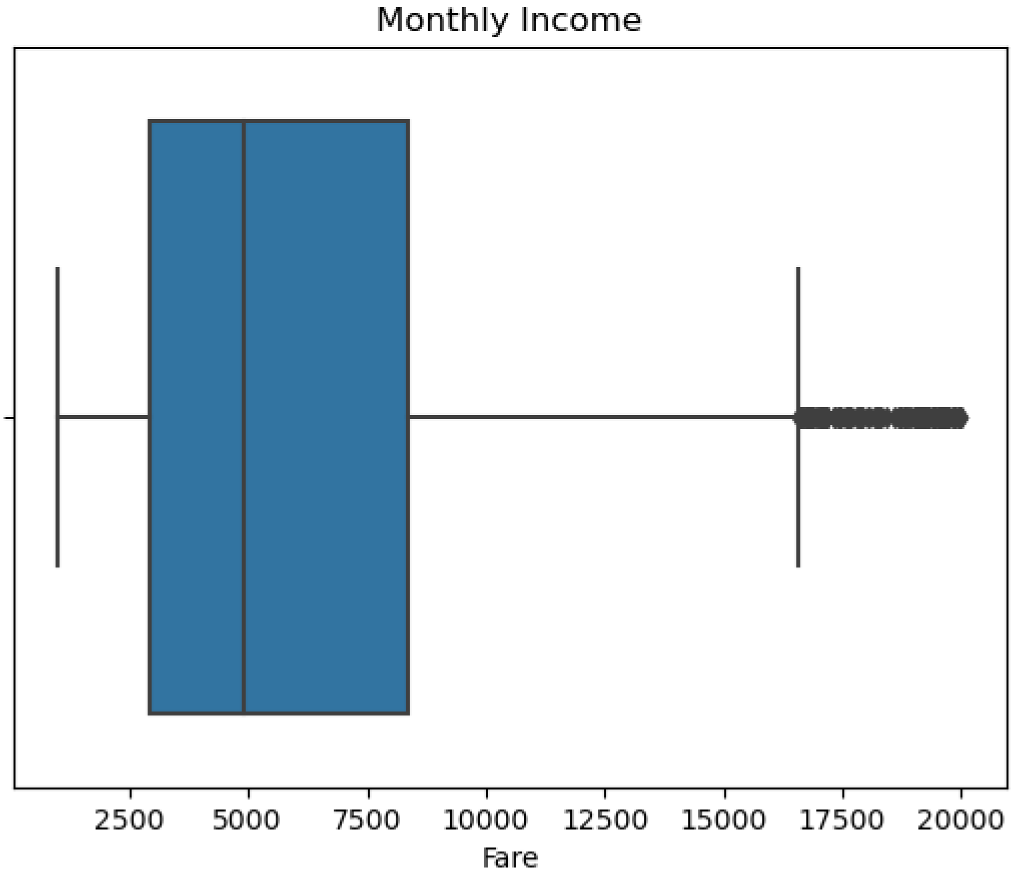
In [19]:
```
sns.boxplot(data=df, x='YearsInCurrentRole')
plt.title('Years In Current Role')
plt.xlabel('Fare')
plt.show()
```

Loading [MathJax]/extensions/Safe.js



In [20]:

```
sns.boxplot(data=df, x='MonthlyIncome')
plt.title('Monthly Income')
plt.xlabel('Fare')
plt.show()
```



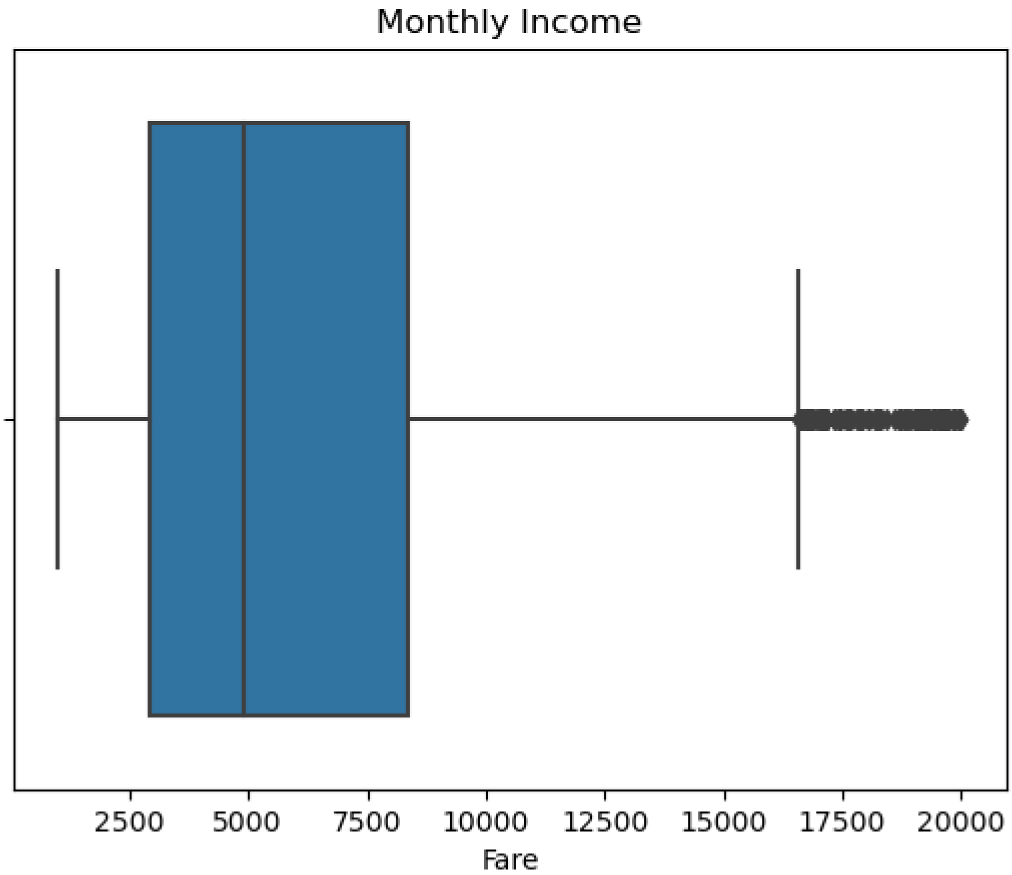
In [21]:

```
from scipy import stats

z_scores = stats.zscore(df['MonthlyIncome'])
z_score_threshold = 3
df_cleaned = df[(np.abs(z_scores) <= z_score_threshold)]
```

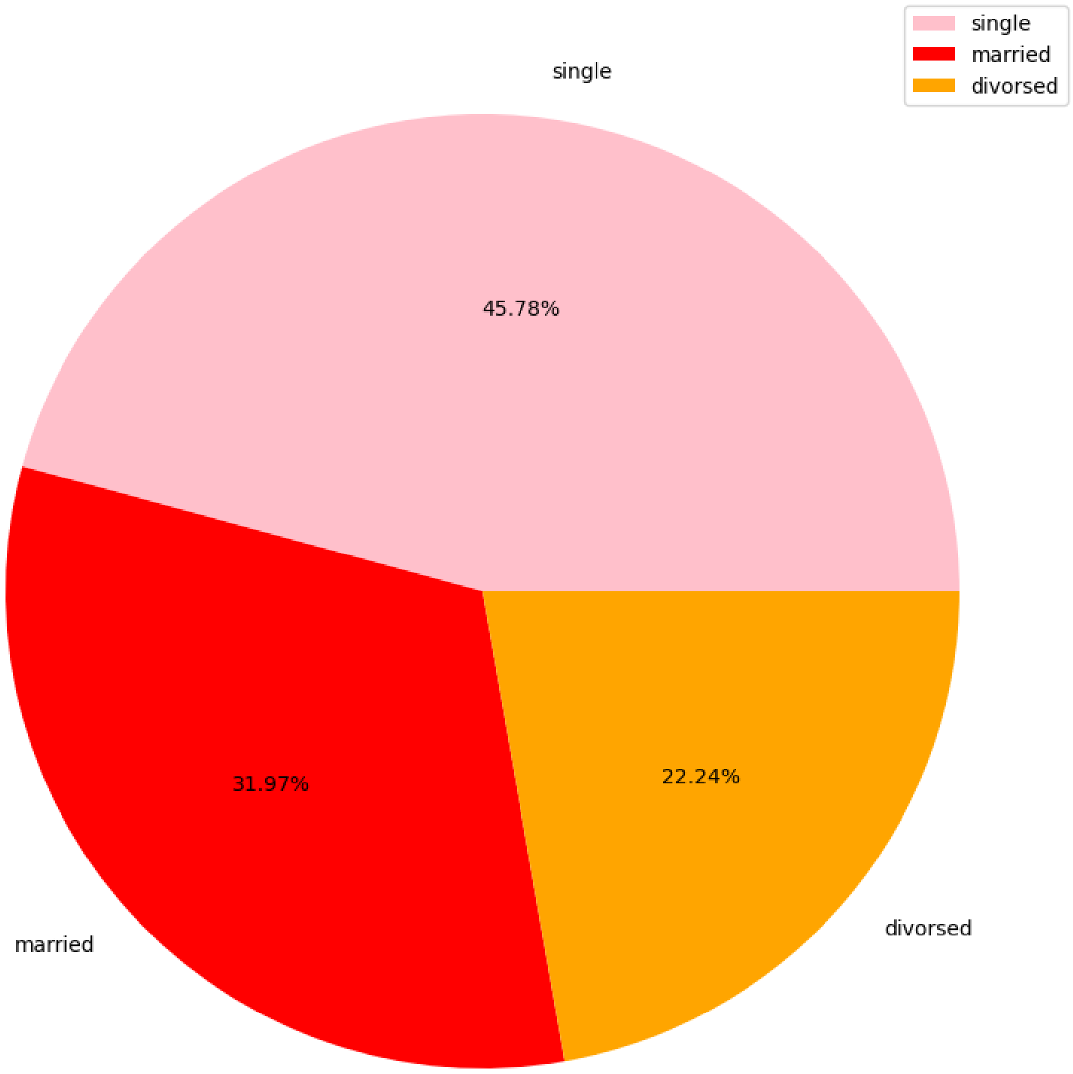
In [22]:

```
sns.boxplot(data=df_cleaned, x='MonthlyIncome')
plt.title('Monthly Income')
plt.xlabel('Fare')
plt.show()
```



```
In [79]: labels=["single","married","divorsed"]
         numbers=df.iloc[:,17].value_counts()
         fig=plt.figure(figsize=(10,10))
         axes1=fig.add_axes([0.1,0.1,0.8,0.8])
         axes1.pie(numbers,labels=labels,autopct='%0.2f%%',colors=['pink','red','orar
         axes1.legend()
```

```
Out[79]: <matplotlib.legend.Legend at 0x198cfc1c950>
```



So the outliers are in large quantity, and they are inside the threshold, so let us not remove the outliers

SPLITTING INDEPENDENT AND DEPENDENT VARIABLES

In [23]:
```
x= df.drop(columns=["Attrition"])
y = df["Attrition"]
```

In [24]:
```
x.head()
```

Out[24]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Educatio
0	41	Travel_Rarely	1102	Sales	1	
1	49	Travel_Frequently	279	Research & Development	8	
2	37	Travel_Rarely	1373	Research & Development	2	
3	33	Travel_Frequently	1392	Research & Development	3	
4	27	Travel_Rarely	591	Research & Development	2	

5 rows × 34 columns

In [25]:

```
y.head()
```

Out[25]:

```
0    Yes
1     No
2    Yes
3     No
4     No
Name: Attrition, dtype: object
```

ENCODING

In [26]:

```
categorical_features = x.select_dtypes(include=['object']).columns.tolist()
x_encoded = pd.get_dummies(x, columns=categorical_features, drop_first=True)
```

In [27]:

```
x_encoded.head()
```

Out[27]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeI
0	41	1102	1	2	1	
1	49	279	8	1	1	
2	37	1373	2	2	1	
3	33	1392	3	4	1	
4	27	591	2	1	1	

5 rows × 47 columns

FEATURE SCALING

In [28]:

```
from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()
x_scaled = pd.DataFrame(scaler.fit_transform(x_encoded), columns=x_encoded.c
```

In [29]:

```
x_scaled.head()
```

Out[29]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	Empl
0	0.446350	0.742527	-1.010909	-0.891688	0.0	
1	1.322365	-1.297775	-0.147150	-1.868426	0.0	
2	0.008343	1.414363	-0.887515	-0.891688	0.0	
3	-0.429664	1.461466	-0.764121	1.061787	0.0	
4	-1.086676	-0.524295	-0.887515	-1.868426	0.0	

5 rows × 47 columns

In [30]:

```
x=x_scaled
```

TRAIN AND TEST SPLIT

In [31]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, rar
```

MODEL BUILDING

LOGISTIC REGRESSION,DECISION TREE

In [32]:

```
# Import the necessary libraries
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, confusior
from joblib import dump
```

In [33]:

```
logreg_model = LogisticRegression(random_state=42)
dt_model = DecisionTreeClassifier(random_state=42)
```

In [34]:

```
logreg_model.fit(x_train, y_train)
dt_model.fit(x_train, y_train)
```

Out[34]:

```
▼        DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)
```

In [35]:

```
logreg_predictions = logreg_model.predict(x_test)
```

```
dt_predictions = dt_model.predict(x_test)

logreg_accuracy = accuracy_score(y_test, logreg_predictions)
print("Logistic Regression Accuracy:", logreg_accuracy)

dt_accuracy = accuracy_score(y_test, dt_predictions)
print("Decision Tree Accuracy:", dt_accuracy)

logreg_report = classification_report(y_test, logreg_predictions)
print("Classification Report for Logistic Regression:\n", logreg_report)

dt_report = classification_report(y_test, dt_predictions)
print("Classification Report for Decision Tree Classifier:\n", dt_report)

logreg_conf_matrix = confusion_matrix(y_test, logreg_predictions)
print("Confusion Matrix for Logistic Regression:\n", logreg_conf_matrix)

dt_conf_matrix = confusion_matrix(y_test, dt_predictions)
print("Confusion Matrix for Decision Tree Classifier:\n", dt_conf_matrix)
```

```
Logistic Regression Accuracy: 0.8809523809523809
Decision Tree Accuracy: 0.7721088435374149
Classification Report for Logistic Regression:
               precision    recall  f1-score   support

          No       0.92      0.95      0.93       255
         Yes       0.56      0.46      0.51        39

    accuracy                           0.88       294
   macro avg       0.74      0.70      0.72       294
weighted avg       0.87      0.88      0.88       294

Classification Report for Decision Tree Classifier:
               precision    recall  f1-score   support

          No       0.87      0.86      0.87       255
         Yes       0.17      0.18      0.17        39

    accuracy                           0.77       294
   macro avg       0.52      0.52      0.52       294
weighted avg       0.78      0.77      0.78       294

Confusion Matrix for Logistic Regression:
 [[241  14]
 [ 21  18]]
Confusion Matrix for Decision Tree Classifier:
 [[220  35]
 [ 32   7]]
```

Model Building

DECISION TREE

```
In [37]: from sklearn.tree import DecisionTreeClassifier
         dtc=DecisionTreeClassifier()
```

```
In [38]: dtc.fit(x_train,y_train)
```

```
Out[38]: ▼ DecisionTreeClassifier

         DecisionTreeClassifier()
```

```
In [39]: pred=dtc.predict(x_test)
```

```
In [40]: pred
```

```
Out[40]: array(['No', 'No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'No', 'No', 'No',
                'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No',
                'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'No', 'Yes', 'No', 'Yes', 'No', 'No', 'Yes',
                'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'Yes', 'No', 'No',
                'No', 'Yes', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'No', 'Yes', 'No', 'Yes', 'No', 'No', 'No', 'No',
                'Yes', 'No', 'No', 'No', 'No', 'No', 'Yes', 'Yes', 'No', 'No',
                'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
                'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No',
                'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No',
                'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
                'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No',
                'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes',
                'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No',
                'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No',
                'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No',
                'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No',
                'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'Yes', 'No', 'No',
                'No'], dtype=object)
```

```
In [41]: y_test
```

Out[41]:

```
1041     No
184      No
1222    Yes
67       No
220      No
       ... 
567      No
560      No
945      No
522      No
651      No
Name: Attrition, Length: 294, dtype: object
```

In [42]:

```
df
```

Out[42]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromH
0	41	Yes	Travel_Rarely	1102	Sales	
1	49	No	Travel_Frequently	279	Research & Development	
2	37	Yes	Travel_Rarely	1373	Research & Development	
3	33	No	Travel_Frequently	1392	Research & Development	
4	27	No	Travel_Rarely	591	Research & Development	
...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research & Development	
1466	39	No	Travel_Rarely	613	Research & Development	
1467	27	No	Travel_Rarely	155	Research & Development	
1468	49	No	Travel_Frequently	1023	Sales	
1469	34	No	Travel_Rarely	628	Research & Development	

1470 rows × 35 columns

In [78]:

```
# Evaluation metrics
# Accuracy score
accuracy = accuracy_score(y_test, pred)
print("Accuracy of Decision tree model: ",accuracy)
```

```
Accuracy of Decision tree model:  0.8741496598639455
```

Evaluation of classification model

In [43]:
```
#Accuracy score
from sklearn.metrics import accuracy_score,confusion_matrix,classification_r
```

In [44]:
```
accuracy_score(y_test,pred)
```

Out[44]:
```
0.782312925170068
```

In [45]:
```
confusion_matrix(y_test,pred)
```

Out[45]:
```
array([[223,  32],
       [ 32,   7]], dtype=int64)
```

In [46]:
```
pd.crosstab(y_test,pred)
```

Out[46]:

col_0	No	Yes
Attrition		
No	223	32
Yes	32	7

Roc-AUC curve

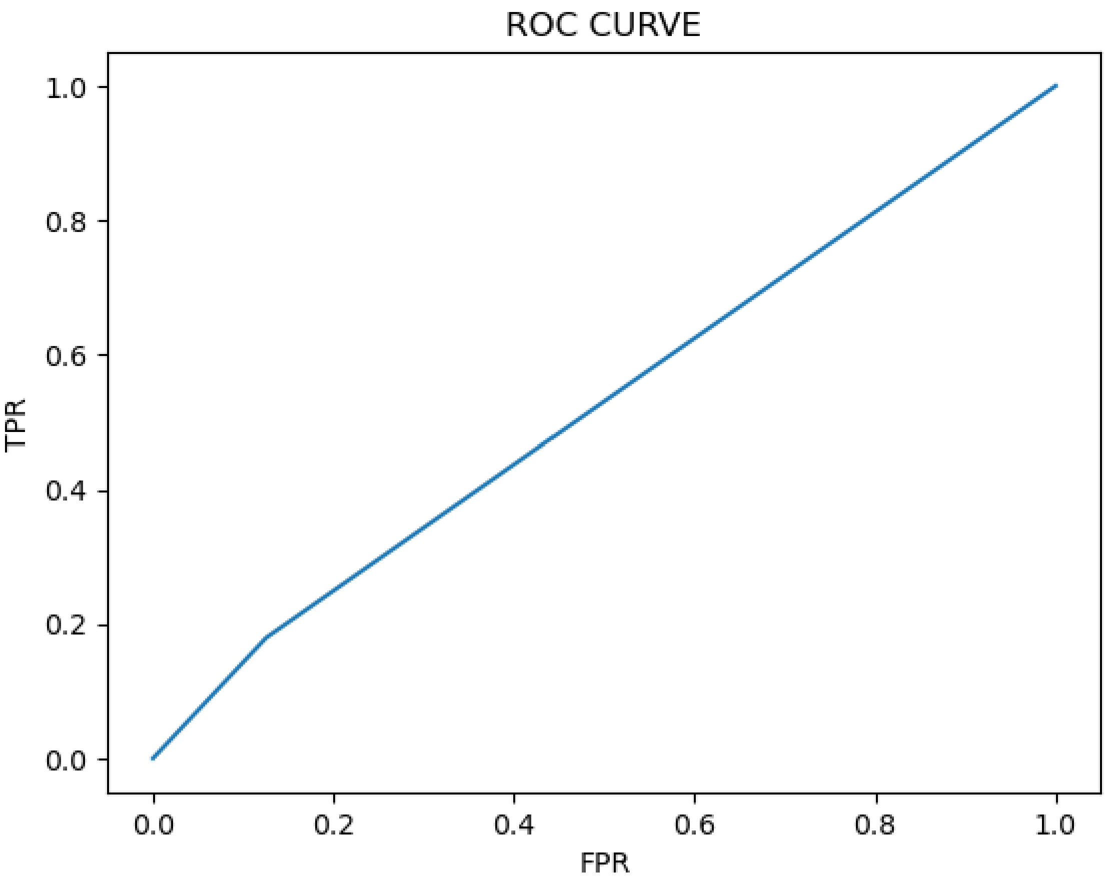
In [47]:
```
probability=dtc.predict_proba(x_test)[:,1]
```

In [48]:
```
probability
```

Out[48]:
```
array([0., 0., 1., 0., 0., 1., 1., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0.,
       0., 1., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0.,
       0., 1., 0., 0., 0., 0., 1., 0., 0., 0., 1., 0., 0., 0., 1., 0., 0.,
       0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 0., 0.,
       0., 1., 0., 0., 0., 0., 0., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0.,
       0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0.,
       0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0.,
       0., 0., 1., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 1., 0.,
       0., 0., 0., 0., 0., 1., 0., 0., 0., 1., 0., 1., 0., 1., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0.,
       0., 0., 0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0.,
       0., 1., 0., 0., 0.])
```

In [49]:
```
fpr, tpr, thresholds = roc_curve(y_test, probability, pos_label='Yes')
```

In [50]:
```
plt.plot(fpr,tpr)
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('ROC CURVE')
plt.show()
```



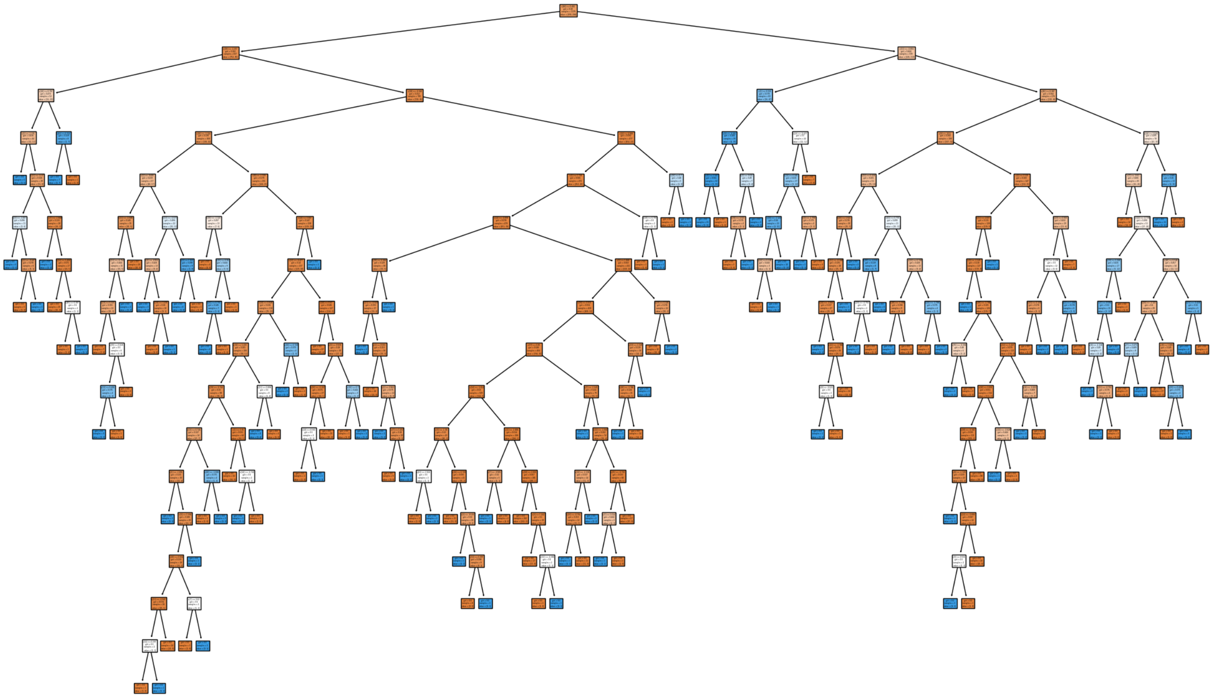
HYPER PARAMETER TUNING

In [51]:
```
from sklearn import tree
plt.figure(figsize=(25,15))
tree.plot_tree(dtc,filled=True)
```

Loading [MathJax]/extensions/Safe.js

```
Out[51]: [Text(0.46866438356164386, 0.9705882352941176, 'x[46] <= 0.482\ngini = 0.28
         \nsamples = 1176\nvalue = [978, 198]'),
          Text(0.18904109589041096, 0.9117647058823529, 'x[19] <= -1.257\ngini = 0.1
         82\nsamples = 837\nvalue = [752, 85]'),
          Text(0.0365296803652968, 0.8529411764705882, 'x[26] <= 0.797\ngini = 0.473
         \nsamples = 52\nvalue = [32, 20]'),
          Text(0.021917808219178082, 0.7941176470588235, 'x[1] <= -1.371\ngini = 0.4
         16\nsamples = 44\nvalue = [31, 13]'),
          Text(0.014611872146118721, 0.7352941176470589, 'gini = 0.0\nsamples = 5\nv
         alue = [0, 5]'),
          Text(0.029223744292237442, 0.7352941176470589, 'x[21] <= -0.37\ngini = 0.3
         26\nsamples = 39\nvalue = [31, 8]'),
          Text(0.014611872146118721, 0.6764705882352942, 'x[0] <= -1.087\ngini = 0.4
         96\nsamples = 11\nvalue = [5, 6]'),
          Text(0.0073059360730593605, 0.6176470588235294, 'gini = 0.0\nsamples = 5\n
         value = [0, 5]'),
          Text(0.021917808219178082, 0.6176470588235294, 'x[2] <= -0.147\ngini = 0.2
         78\nsamples = 6\nvalue = [5, 1]'),
          Text(0.014611872146118721, 0.5588235294117647, 'gini = 0.0\nsamples = 5\nv
         alue = [5, 0]'),
          Text(0.029223744292237442, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nv
         alue = [0, 1]'),
          Text(0.043835616438356165, 0.6764705882352942, 'x[7] <= -1.668\ngini = 0.1
         33\nsamples = 28\nvalue = [26, 2]'),
          Text(0.0365296803652968, 0.6176470588235294, 'gini = 0.0\nsamples = 1\nval
         ue = [0, 1]'),
          Text(0.05114155251141553, 0.6176470588235294, 'x[12] <= 1.406\ngini = 0.07
         1\nsamples = 27\nvalue = [26, 1]'),
          Text(0.043835616438356165, 0.5588235294117647, 'gini = 0.0\nsamples = 25\n
         value = [25, 0]'),
          Text(0.058447488584474884, 0.5588235294117647, 'x[45] <= 0.387\ngini = 0.5
         \nsamples = 2\nvalue = [1, 1]'),
          Text(0.05114155251141553, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
          Text(0.06575342465753424, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
          Text(0.05114155251141553, 0.7941176470588235, 'x[41] <= 0.755\ngini = 0.21
         9\nsamples = 8\nvalue = [1, 7]'),
          Text(0.043835616438356165, 0.7352941176470589, 'gini = 0.0\nsamples = 7\nv
         alue = [0, 7]'),
          Text(0.058447488584474884, 0.7352941176470589, 'gini = 0.0\nsamples = 1\nv
         alue = [1, 0]'),
          Text(0.3415525114155251, 0.8529411764705882, 'x[18] <= -0.345\ngini = 0.15
         2\nsamples = 785\nvalue = [720, 65]'),
          Text(0.16678082191780821, 0.7941176470588235, 'x[10] <= -1.114\ngini = 0.2
         36\nsamples = 308\nvalue = [266, 42]'),
          Text(0.12054794520547946, 0.7352941176470589, 'x[2] <= -0.085\ngini = 0.41
         9\nsamples = 57\nvalue = [40, 17]'),
          Text(0.10228310502283106, 0.6764705882352942, 'x[11] <= -0.688\ngini = 0.2
         45\nsamples = 35\nvalue = [30, 5]'),
          Text(0.09497716894977169, 0.6176470588235294, 'x[11] <= -0.754\ngini = 0.4
         44\nsamples = 15\nvalue = [10, 5]'),
          Text(0.08767123287671233, 0.5588235294117647, 'x[13] <= 0.323\ngini = 0.35
         5\nsamples = 13\nvalue = [10, 3]'),
          Text(0.08036529680365297, 0.5, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
          Text(0.09497716894977169, 0.5, 'x[19] <= 0.028\ngini = 0.5\nsamples = 6\nv
         alue = [3, 3]'),
  Loading [MathJax]/extensions/Safe.js 8767123287671233, 0.4411764705882353, 'x[12] <= 0.406\ngini = 0.37
```

```
5\nsamples = 4\nvalue = [1, 3]'),
 Text(0.08036529680365297, 0.38235294117647056, 'gini = 0.0\nsamples = 3\nv
alue = [0, 3]'),
 Text(0.09497716894977169, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nv
alue = [1, 0]'),
 Text(0.10228310502283106, 0.4411764705882353, 'gini = 0.0\nsamples = 2\nva
lue = [2, 0]'),
 Text(0.10228310502283106, 0.5588235294117647, 'gini = 0.0\nsamples = 2\nva
lue = [0, 2]'),
 Text(0.1095890410958904, 0.6176470588235294, 'gini = 0.0\nsamples = 20\nva
lue = [20, 0]'),
 Text(0.13881278538812786, 0.6764705882352942, 'x[13] <= -0.077\ngini = 0.4
96\nsamples = 22\nvalue = [10, 12]'),
 Text(0.12420091324200913, 0.6176470588235294, 'x[1] <= -0.951\ngini = 0.42
6\nsamples = 13\nvalue = [9, 4]'),
 Text(0.11689497716894977, 0.5588235294117647, 'gini = 0.0\nsamples = 3\nva
lue = [0, 3]'),
 Text(0.13150684931506848, 0.5588235294117647, 'x[34] <= 1.435\ngini = 0.18
\nsamples = 10\nvalue = [9, 1]'),
 Text(0.12420091324200913, 0.5, 'gini = 0.0\nsamples = 9\nvalue = [9, 0]'),
 Text(0.13881278538812786, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.15342465753424658, 0.6176470588235294, 'x[0] <= 1.925\ngini = 0.198
\nsamples = 9\nvalue = [1, 8]'),
 Text(0.1461187214611872, 0.5588235294117647, 'gini = 0.0\nsamples = 8\nval
ue = [0, 8]'),
 Text(0.16073059360730593, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nva
lue = [1, 0]'),
 Text(0.213013698630137, 0.7352941176470589, 'x[8] <= -1.729\ngini = 0.179
\nsamples = 251\nvalue = [226, 25]'),
 Text(0.17534246575342466, 0.6764705882352942, 'x[16] <= -1.122\ngini = 0.4
97\nsamples = 13\nvalue = [7, 6]'),
 Text(0.1680365296803653, 0.6176470588235294, 'gini = 0.0\nsamples = 4\nval
ue = [4, 0]'),
 Text(0.182648401826484, 0.6176470588235294, 'x[15] <= 0.96\ngini = 0.444\n
samples = 9\nvalue = [3, 6]'),
 Text(0.17534246575342466, 0.5588235294117647, 'x[41] <= 0.755\ngini = 0.24
5\nsamples = 7\nvalue = [1, 6]'),
 Text(0.1680365296803653, 0.5, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.182648401826484, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.18995433789954339, 0.5588235294117647, 'gini = 0.0\nsamples = 2\nva
lue = [2, 0]'),
 Text(0.25068493150684934, 0.6764705882352942, 'x[24] <= 3.667\ngini = 0.14
7\nsamples = 238\nvalue = [219, 19]'),
 Text(0.24337899543378996, 0.6176470588235294, 'x[23] <= -0.477\ngini = 0.1
4\nsamples = 237\nvalue = [219, 18]'),
 Text(0.2182648401826484, 0.5588235294117647, 'x[24] <= 1.028\ngini = 0.236
\nsamples = 110\nvalue = [95, 15]'),
 Text(0.19726027397260273, 0.5, 'x[34] <= 1.435\ngini = 0.201\nsamples = 10
6\nvalue = [94, 12]'),
 Text(0.1771689497716895, 0.4411764705882353, 'x[19] <= -0.743\ngini = 0.15
\nsamples = 98\nvalue = [90, 8]'),
 Text(0.1589041095890411, 0.38235294117647056, 'x[13] <= 0.724\ngini = 0.32
7\nsamples = 34\nvalue = [27, 7]'),
 Text(0.14429223744292238, 0.3235294117647059, 'x[1] <= -1.691\ngini = 0.23
1\nsamples = 30\nvalue = [26, 4]'),
 Text(0.136986301369863, 0.2647058823529412, 'gini = 0.0\nsamples = 1\nvalu
e = [0, 1]'),
 Text(0.15159817351598173, 0.2647058823529412, 'x[0] <= 1.596\ngini = 0.185
\nsamples = 29\nvalue = [26, 3]'),
 Text(0.14429223744292238, 0.20588235294117646, 'x[20] <= 0.932\ngini = 0.1
33\nsamples = 28\nvalue = [26, 2]'),
 Text(0.12968036529680366, 0.14705882352941177, 'x[11] <= -0.947\ngini = 0.
074\nsamples = 26\nvalue = [25, 1]'),
 Text(0.1223744292237443, 0.08823529411764706, 'x[32] <= 0.397\ngini = 0.5
\nsamples = 2\nvalue = [1, 1]'),
 Text(0.11506849315068493, 0.029411764705882353, 'gini = 0.0\nsamples = 1\n
value = [1, 0]'),
 Text(0.12968036529680366, 0.029411764705882353, 'gini = 0.0\nsamples = 1\n
value = [0, 1]'),
 Text(0.136986301369863, 0.08823529411764706, 'gini = 0.0\nsamples = 24\nva
lue = [24, 0]'),
 Text(0.1589041095890411, 0.14705882352941177, 'x[7] <= 0.62\ngini = 0.5\ns
amples = 2\nvalue = [1, 1]'),
 Text(0.15159817351598173, 0.08823529411764706, 'gini = 0.0\nsamples = 1\nv
alue = [1, 0]'),
 Text(0.16621004566210046, 0.08823529411764706, 'gini = 0.0\nsamples = 1\nv
alue = [0, 1]'),
 Text(0.1589041095890411, 0.20588235294117646, 'gini = 0.0\nsamples = 1\nva
lue = [0, 1]'),
 Text(0.1735159817351598, 0.3235294117647059, 'x[20] <= -1.008\ngini = 0.37
5\nsamples = 4\nvalue = [1, 3]'),
 Text(0.16621004566210046, 0.2647058823529412, 'gini = 0.0\nsamples = 1\nva
lue = [1, 0]'),
 Text(0.18082191780821918, 0.2647058823529412, 'gini = 0.0\nsamples = 3\nva
lue = [0, 3]'),
 Text(0.1954337899543379, 0.38235294117647056, 'x[1] <= 1.65\ngini = 0.031
\nsamples = 64\nvalue = [63, 1]'),
 Text(0.18812785388127853, 0.3235294117647059, 'gini = 0.0\nsamples = 62\nv
alue = [62, 0]'),
 Text(0.20273972602739726, 0.3235294117647059, 'x[14] <= -0.741\ngini = 0.5
\nsamples = 2\nvalue = [1, 1]'),
 Text(0.1954337899543379, 0.2647058823529412, 'gini = 0.0\nsamples = 1\nval
ue = [0, 1]'),
 Text(0.2100456621004566, 0.2647058823529412, 'gini = 0.0\nsamples = 1\nval
ue = [1, 0]'),
 Text(0.217351598173516, 0.4411764705882353, 'x[12] <= 0.898\ngini = 0.5\ns
amples = 8\nvalue = [4, 4]'),
 Text(0.2100456621004566, 0.38235294117647056, 'gini = 0.0\nsamples = 4\nva
lue = [0, 4]'),
 Text(0.22465753424657534, 0.38235294117647056, 'gini = 0.0\nsamples = 4\nv
alue = [4, 0]'),
 Text(0.23926940639269406, 0.5, 'x[16] <= 0.729\ngini = 0.375\nsamples = 4
\nvalue = [1, 3]'),
 Text(0.2319634703196347, 0.4411764705882353, 'gini = 0.0\nsamples = 3\nval
ue = [0, 3]'),
 Text(0.2465753424657534, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nval
ue = [1, 0]'),
 Text(0.2684931506849315, 0.5588235294117647, 'x[42] <= 0.67\ngini = 0.046
\nsamples = 127\nvalue = [124, 3]'),
 Text(0.26118721461187216, 0.5, 'gini = 0.0\nsamples = 97\nvalue = [97,
0]'),
 Text(0.27579908675799086, 0.5, 'x[0] <= 0.939\ngini = 0.18\nsamples = 30\n
value = [27, 3]'),
 Text(0.26118721461187216, 0.4411764705882353, 'x[10] <= -0.207\ngini = 0.0
71\nsamples = 27\nvalue = [26, 1]'),
 Text(0.2538812785388l276, 0.38235294117647056, 'x[24] <= 0.563\ngini = 0.5
\nsamples = 2\nvalue = [1, 1]'),
 Text(0.2465753424657534, 0.3235294117647059, 'gini = 0.0\nsamples = 1\nval
ue = [1, 0]'),
 Text(0.26118721461187216, 0.3235294117647059, 'gini = 0.0\nsamples = 1\nva
lue = [0, 1]'),
 Text(0.2684931506849315, 0.38235294117647056, 'gini = 0.0\nsamples = 25\nv
alue = [25, 0]'),
 Text(0.29041095890410956, 0.4411764705882353, 'x[10] <= 0.7\ngini = 0.444
\nsamples = 3\nvalue = [1, 2]'),
 Text(0.2831050228310502, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nva
lue = [1, 0]'),
 Text(0.29771689497716897, 0.38235294117647056, 'gini = 0.0\nsamples = 2\nv
alue = [0, 2]'),
 Text(0.2579908675799087, 0.6176470588235294, 'gini = 0.0\nsamples = 1\nval
ue = [0, 1]'),
 Text(0.516324200913242, 0.7941176470588235, 'x[22] <= 3.999\ngini = 0.092
\nsamples = 477\nvalue = [454, 23]'),
 Text(0.47465753424657536, 0.7352941176470589, 'x[1] <= 1.718\ngini = 0.081
\nsamples = 472\nvalue = [452, 20]'),
 Text(0.4132420091324201, 0.6764705882352942, 'x[22] <= -0.899\ngini = 0.07
8\nsamples = 470\nvalue = [451, 19]'),
 Text(0.31232876712328766, 0.6176470588235294, 'x[39] <= 1.346\ngini = 0.22
9\nsamples = 38\nvalue = [33, 5]'),
 Text(0.3050228310502283, 0.5588235294117647, 'x[11] <= -0.929\ngini = 0.19
3\nsamples = 37\nvalue = [33, 4]'),
 Text(0.29771689497716897, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.31232876712328766, 0.5, 'x[2] <= 0.593\ngini = 0.153\nsamples = 36
\nvalue = [33, 3]'),
 Text(0.3050228310502283, 0.4411764705882353, 'gini = 0.0\nsamples = 24\nva
lue = [24, 0]'),
 Text(0.319634703196347, 0.4411764705882353, 'x[2] <= 0.963\ngini = 0.375\n
samples = 12\nvalue = [9, 3]'),
 Text(0.31232876712328766, 0.38235294117647056, 'gini = 0.0\nsamples = 2\nv
alue = [0, 2]'),
 Text(0.3269406392694064, 0.38235294117647056, 'x[7] <= 1.457\ngini = 0.18
\nsamples = 10\nvalue = [9, 1]'),
 Text(0.319634703196347, 0.3235294117647059, 'gini = 0.0\nsamples = 9\nvalu
e = [9, 0]'),
 Text(0.33424657534246577, 0.3235294117647059, 'gini = 0.0\nsamples = 1\nva
lue = [0, 1]'),
 Text(0.319634703196347, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nvalu
e = [0, 1]'),
 Text(0.5141552511415525, 0.6176470588235294, 'x[5] <= 1.668\ngini = 0.063
\nsamples = 432\nvalue = [418, 14]'),
 Text(0.4821917808219178, 0.5588235294117647, 'x[36] <= 2.515\ngini = 0.056
\nsamples = 420\nvalue = [408, 12]'),
 Text(0.44018264840182647, 0.5, 'x[37] <= 0.85\ngini = 0.048\nsamples = 404
\nvalue = [394, 10]'),
 Text(0.3926940639269406, 0.4411764705882353, 'x[14] <= -1.014\ngini = 0.03
\nsamples = 329\nvalue = [324, 5]'),
 Text(0.3634703196347032, 0.38235294117647056, 'x[3] <= -1.38\ngini = 0.13
\nsamples = 43\nvalue = [40, 3]'),
 Text(0.34885844748858447, 0.3235294117647059, 'x[43] <= 1.922\ngini = 0.5
\nsamples = 2\nvalue = [1, 1]'),
 Text(0.3415525114155251, 0.2647058823529412, 'gini = 0.0\nsamples = 1\nval
ue = [1, 0]'),
 Text(0.3561643835616438, 0.2647058823529412, 'gini = 0.0\nsamples = 1\nval
ue = [0, 1]'),
 Text(0.3780821917808219, 0.3235294117647059, 'x[16] <= 0.729\ngini = 0.093
\nsamples = 41\nvalue = [39, 2]'),
 Text(0.37077625570776257, 0.2647058823529412, 'gini = 0.0\nsamples = 31\nv
alue = [31, 0]'),
 Text(0.38538812785388127, 0.2647058823529412, 'x[0] <= -0.539\ngini = 0.32
\nsamples = 10\nvalue = [8, 2]'),
 Text(0.3780821917808219, 0.20588235294117646, 'gini = 0.0\nsamples = 1\nva
lue = [0, 1]'),
 Text(0.3926940639269406, 0.20588235294117646, 'x[32] <= 0.397\ngini = 0.19
8\nsamples = 9\nvalue = [8, 1]'),
 Text(0.38538812785388127, 0.14705882352941177, 'gini = 0.0\nsamples = 8\nv
alue = [8, 0]'),
 Text(0.4, 0.14705882352941177, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.42191780821917807, 0.38235294117647056, 'x[1] <= -1.686\ngini = 0.0
14\nsamples = 286\nvalue = [284, 2]'),
 Text(0.40730593607305937, 0.3235294117647059, 'x[7] <= -0.905\ngini = 0.32
\nsamples = 5\nvalue = [4, 1]'),
 Text(0.4, 0.2647058823529412, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.4146118721461187, 0.2647058823529412, 'gini = 0.0\nsamples = 4\nval
ue = [4, 0]'),
 Text(0.4365296803652968, 0.3235294117647059, 'x[18] <= 2.003\ngini = 0.007
\nsamples = 281\nvalue = [280, 1]'),
 Text(0.4292237442922374, 0.2647058823529412, 'gini = 0.0\nsamples = 262\nv
alue = [262, 0]'),
 Text(0.4438356164383562, 0.2647058823529412, 'x[5] <= 1.242\ngini = 0.1\ns
amples = 19\nvalue = [18, 1]'),
 Text(0.4365296803652968, 0.20588235294117646, 'gini = 0.0\nsamples = 17\nv
alue = [17, 0]'),
 Text(0.4511415525114155, 0.20588235294117646, 'x[12] <= -0.381\ngini = 0.5
\nsamples = 2\nvalue = [1, 1]'),
 Text(0.4438356164383562, 0.14705882352941177, 'gini = 0.0\nsamples = 1\nva
lue = [1, 0]'),
 Text(0.45844748858447487, 0.14705882352941177, 'gini = 0.0\nsamples = 1\nv
alue = [0, 1]'),
 Text(0.4876712328767123, 0.4411764705882353, 'x[7] <= -1.545\ngini = 0.124
\nsamples = 75\nvalue = [70, 5]'),
 Text(0.480365296803653, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nval
ue = [0, 1]'),
 Text(0.4949771689497717, 0.38235294117647056, 'x[12] <= -1.065\ngini = 0.1
02\nsamples = 74\nvalue = [70, 4]'),
 Text(0.480365296803653, 0.3235294117647059, 'x[12] <= -1.136\ngini = 0.337
\nsamples = 14\nvalue = [11, 3]'),
 Text(0.4730593607305936, 0.2647058823529412, 'x[23] <= -0.754\ngini = 0.15
3\nsamples = 12\nvalue = [11, 1]'),
 Text(0.4657534246575342, 0.20588235294117646, 'gini = 0.0\nsamples = 1\nva
lue = [0, 1]'),
 Text(0.480365296803653, 0.20588235294117646, 'gini = 0.0\nsamples = 11\nva
lue = [11, 0]'),
 Text(0.4876712328767123, 0.2647058823529412, 'gini = 0.0\nsamples = 2\nval
ue = [0, 2]'),
 Text(0.5095890410958904, 0.3235294117647059, 'x[25] <= -0.735\ngini = 0.03
3\nsamples = 60\nvalue = [59, 1]'),
 Text(0.502283105022831, 0.2647058823529412, 'x[7] <= 1.014\ngini = 0.444\n
samples = 3\nvalue = [2, 1]'),
 Text(0.4949771689497717, 0.20588235294117646, 'gini = 0.0\nsamples = 1\nva
lue = [0, 1]'),
 Text(0.5095890410958904, 0.20588235294117646, 'gini = 0.0\nsamples = 2\nva
lue = [2, 0]'),
 Text(0.5168949771689497, 0.2647058823529412, 'gini = 0.0\nsamples = 57\nva
lue = [57, 0]'),
 Text(0.5242009132420091, 0.5, 'x[13] <= 2.325\ngini = 0.219\nsamples = 16
\nvalue = [14, 2]'),
 Text(0.5168949771689497, 0.4411764705882353, 'x[20] <= -1.784\ngini = 0.12
4\nsamples = 15\nvalue = [14, 1]'),
 Text(0.5095890410958904, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nva
lue = [0, 1]'),
 Text(0.5242009132420091, 0.38235294117647056, 'gini = 0.0\nsamples = 14\nv
alue = [14, 0]'),
 Text(0.5315068493150685, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nval
ue = [0, 1]'),
 Text(0.5461187214611872, 0.5588235294117647, 'x[13] <= 1.124\ngini = 0.278
\nsamples = 12\nvalue = [10, 2]'),
 Text(0.5388127853881278, 0.5, 'gini = 0.0\nsamples = 10\nvalue = [10,
0]'),
 Text(0.5534246575342465, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.5360730593607306, 0.6764705882352942, 'x[26] <= 0.797\ngini = 0.5\n
samples = 2\nvalue = [1, 1]'),
 Text(0.5287671232876713, 0.6176470588235294, 'gini = 0.0\nsamples = 1\nval
ue = [1, 0]'),
 Text(0.54337899543379, 0.6176470588235294, 'gini = 0.0\nsamples = 1\nvalue
= [0, 1]'),
 Text(0.5579908675799087, 0.7352941176470589, 'x[10] <= 0.7\ngini = 0.48\ns
amples = 5\nvalue = [2, 3]'),
 Text(0.5506849315068493, 0.6764705882352942, 'gini = 0.0\nsamples = 2\nval
ue = [2, 0]'),
 Text(0.5652968036529681, 0.6764705882352942, 'gini = 0.0\nsamples = 3\nval
ue = [0, 3]'),
 Text(0.7482876712328768, 0.9117647058823529, 'x[11] <= -0.856\ngini = 0.44
4\nsamples = 339\nvalue = [226, 113]'),
 Text(0.6310502283105023, 0.8529411764705882, 'x[1] <= 0.319\ngini = 0.379
\nsamples = 63\nvalue = [16, 47]'),
 Text(0.6018264840182649, 0.7941176470588235, 'x[0] <= -0.101\ngini = 0.184
\nsamples = 39\nvalue = [4, 35]'),
 Text(0.5872146118721461, 0.7352941176470589, 'x[5] <= 1.587\ngini = 0.061
\nsamples = 32\nvalue = [1, 31]'),
 Text(0.5799086757990868, 0.6764705882352942, 'gini = 0.0\nsamples = 31\nva
lue = [0, 31]'),
 Text(0.5945205479452055, 0.6764705882352942, 'gini = 0.0\nsamples = 1\nval
ue = [1, 0]'),
 Text(0.6164383561643836, 0.7352941176470589, 'x[0] <= 0.611\ngini = 0.49\n
samples = 7\nvalue = [3, 4]'),
 Text(0.6091324200913242, 0.6764705882352942, 'x[36] <= 2.515\ngini = 0.375
\nsamples = 4\nvalue = [3, 1]'),
 Text(0.6018264840182649, 0.6176470588235294, 'gini = 0.0\nsamples = 3\nval
ue = [3, 0]'),
 Text(0.6164383561643836, 0.6176470588235294, 'gini = 0.0\nsamples = 1\nval
ue = [0, 1]'),
 Text(0.6237442922374429, 0.6764705882352942, 'gini = 0.0\nsamples = 3\nval
ue = [0, 3]'),
 Text(0.6602739726027397, 0.7941176470588235, 'x[23] <= -0.477\ngini = 0.5
\nsamples = 24\nvalue = [12, 12]'),
 Text(0.6529680365296804, 0.7352941176470589, 'x[12] <= 0.345\ngini = 0.415
\nsamples = 17\nvalue = [5, 12]'),
 Text(0.6383561643835617, 0.6764705882352942, 'x[16] <= -1.122\ngini = 0.26
\nsamples = 13\nvalue = [2, 11]'),
 Text(0.6310502283105023, 0.6176470588235294, 'x[21] <= 0.338\ngini = 0.444
\nsamples = 3\nvalue = [2, 1]'),
 Text(0.6237442922374429, 0.5588235294117647, 'gini = 0.0\nsamples = 2\nval
ue = [2, 0]'),
 Text(0.6383561643835617, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nval
ue = [0, 1]'),
 Text(0.645662100456621, 0.6176470588235294, 'gini = 0.0\nsamples = 10\nval
ue = [0, 10]'),
 Text(0.6675799086757991, 0.6764705882352942, 'x[0] <= -1.798\ngini = 0.375
\nsamples = 4\nvalue = [3, 1]'),
 Text(0.6602739726027397, 0.6176470588235294, 'gini = 0.0\nsamples = 1\nval
ue = [0, 1]'),
 Text(0.6748858447488585, 0.6176470588235294, 'gini = 0.0\nsamples = 3\nval
ue = [3, 0]'),
 Text(0.6675799086757991, 0.7352941176470589, 'gini = 0.0\nsamples = 7\nval
ue = [7, 0]'),
 Text(0.8655251141552511, 0.8529411764705882, 'x[45] <= 0.387\ngini = 0.364
\nsamples = 276\nvalue = [210, 66]'),
 Text(0.780365296803653, 0.7941176470588235, 'x[9] <= -0.51\ngini = 0.264\n
samples = 198\nvalue = [167, 31]'),
 Text(0.7168949771689498, 0.7352941176470589, 'x[12] <= 0.328\ngini = 0.431
\nsamples = 51\nvalue = [35, 16]'),
 Text(0.6968036529680365, 0.6764705882352942, 'x[43] <= 1.922\ngini = 0.278
\nsamples = 30\nvalue = [25, 5]'),
 Text(0.6894977168949772, 0.6176470588235294, 'x[11] <= -0.348\ngini = 0.19
1\nsamples = 28\nvalue = [25, 3]'),
 Text(0.6821917808219178, 0.5588235294117647, 'x[22] <= -1.063\ngini = 0.13
7\nsamples = 27\nvalue = [25, 2]'),
 Text(0.6748858447488585, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6894977168949772, 0.5, 'x[21] <= -1.786\ngini = 0.074\nsamples = 26
\nvalue = [25, 1]'),
 Text(0.6821917808219178, 0.4411764705882353, 'x[8] <= -0.323\ngini = 0.5\n
samples = 2\nvalue = [1, 1]'),
 Text(0.6748858447488585, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nva
lue = [0, 1]'),
 Text(0.6894977168949772, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nva
lue = [1, 0]'),
 Text(0.6968036529680365, 0.4411764705882353, 'gini = 0.0\nsamples = 24\nva
lue = [24, 0]'),
 Text(0.6968036529680365, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nval
ue = [0, 1]'),
 Text(0.7041095890410959, 0.6176470588235294, 'gini = 0.0\nsamples = 2\nval
ue = [0, 2]'),
 Text(0.736986301369863, 0.6764705882352942, 'x[41] <= 0.755\ngini = 0.499
\nsamples = 21\nvalue = [10, 11]'),
 Text(0.7187214611872146, 0.6176470588235294, 'x[19] <= -0.936\ngini = 0.21
9\nsamples = 8\nvalue = [1, 7]'),
 Text(0.7114155251141553, 0.5588235294117647, 'x[23] <= -0.754\ngini = 0.5
\nsamples = 2\nvalue = [1, 1]'),
 Text(0.7041095890410959, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.7187214611872146, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.726027397260274, 0.5588235294117647, 'gini = 0.0\nsamples = 6\nvalu
e = [0, 6]'),
 Text(0.7552511415525114, 0.6176470588235294, 'x[13] <= 0.724\ngini = 0.426
\nsamples = 13\nvalue = [9, 4]'),
 Text(0.7406392694063927, 0.5588235294117647, 'x[20] <= -1.396\ngini = 0.19
8\nsamples = 9\nvalue = [8, 1]'),
 Text(0.7333333333333333, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.7479452054794521, 0.5, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(0.7698630136986301, 0.5588235294117647, 'x[21] <= -1.786\ngini = 0.37
5\nsamples = 4\nvalue = [1, 3]'),
 Text(0.7625570776255708, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.7771689497716895, 0.5, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.8438356164383561, 0.7352941176470589, 'x[2] <= 1.519\ngini = 0.183
\nsamples = 147\nvalue = [132, 15]'),
 Text(0.8118721461187215, 0.6764705882352942, 'x[11] <= 2.837\ngini = 0.131
\nsamples = 128\nvalue = [119, 9]'),
 Text(0.8045662100456621, 0.6176470588235294, 'x[0] <= -1.525\ngini = 0.118
\nsamples = 127\nvalue = [119, 8]'),
 Text(0.7972602739726027, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nval
ue = [0, 1]'),
 Text(0.8118721461187215, 0.5588235294117647, 'x[20] <= -1.784\ngini = 0.10
5\nsamples = 126\nvalue = [119, 7]'),
 Text(0.7917808219178082, 0.5, 'x[35] <= -0.204\ngini = 0.48\nsamples = 5\n
value = [3, 2]'),
 Text(0.7844748858447489, 0.4411764705882353, 'gini = 0.0\nsamples = 2\nval
ue = [0, 2]'),
 Text(0.7990867579908676, 0.4411764705882353, 'gini = 0.0\nsamples = 3\nval
ue = [3, 0]'),
 Text(0.8319634703196347, 0.5, 'x[12] <= 1.447\ngini = 0.079\nsamples = 121
\nvalue = [116, 5]'),
 Text(0.8136986301369863, 0.4411764705882353, 'x[20] <= 2.095\ngini = 0.051
\nsamples = 114\nvalue = [111, 3]'),
 Text(0.7990867579908676, 0.38235294117647056, 'x[18] <= -0.345\ngini = 0.0
35\nsamples = 111\nvalue = [109, 2]'),
 Text(0.7917808219178082, 0.3235294117647059, 'x[8] <= -1.729\ngini = 0.159
\nsamples = 23\nvalue = [21, 2]'),
 Text(0.7844748858447489, 0.2647058823529412, 'gini = 0.0\nsamples = 1\nval
ue = [0, 1]'),
 Text(0.7990867579908676, 0.2647058823529412, 'x[22] <= -0.899\ngini = 0.08
7\nsamples = 22\nvalue = [21, 1]'),
 Text(0.7917808219178082, 0.20588235294117646, 'x[0] <= 0.556\ngini = 0.5\n
samples = 2\nvalue = [1, 1]'),
 Text(0.7844748858447489, 0.14705882352941177, 'gini = 0.0\nsamples = 1\nva
lue = [0, 1]'),
 Text(0.7990867579908676, 0.14705882352941177, 'gini = 0.0\nsamples = 1\nva
lue = [1, 0]'),
 Text(0.806392694063927, 0.20588235294117646, 'gini = 0.0\nsamples = 20\nva
lue = [20, 0]'),
 Text(0.806392694063927, 0.3235294117647059, 'gini = 0.0\nsamples = 88\nval
ue = [88, 0]'),
 Text(0.828310502283105, 0.38235294117647056, 'x[12] <= -0.626\ngini = 0.44
4\nsamples = 3\nvalue = [2, 1]'),
 Text(0.8210045662100457, 0.3235294117647059, 'gini = 0.0\nsamples = 1\nval
ue = [0, 1]'),
 Text(0.8356164383561644, 0.3235294117647059, 'gini = 0.0\nsamples = 2\nval
ue = [2, 0]'),
 Text(0.8502283105022831, 0.4411764705882353, 'x[6] <= -0.203\ngini = 0.408
\nsamples = 7\nvalue = [5, 2]'),
 Text(0.8429223744292238, 0.38235294117647056, 'gini = 0.0\nsamples = 2\nva
lue = [0, 2]'),
 Text(0.8575342465753425, 0.38235294117647056, 'gini = 0.0\nsamples = 5\nva
lue = [5, 0]'),
 Text(0.8191780821917808, 0.6176470588235294, 'gini = 0.0\nsamples = 1\nval
ue = [0, 1]'),
 Text(0.8757990867579909, 0.6764705882352942, 'x[23] <= 0.903\ngini = 0.432
\nsamples = 19\nvalue = [13, 6]'),
 Text(0.8684931506849315, 0.6176470588235294, 'x[35] <= -0.204\ngini = 0.5
\nsamples = 12\nvalue = [6, 6]'),
 Text(0.8538812785388128, 0.5588235294117647, 'x[13] <= 2.125\ngini = 0.278
\nsamples = 6\nvalue = [5, 1]'),
 Text(0.8465753424657534, 0.5, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.8611872146118722, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8831050228310502, 0.5588235294117647, 'x[1] <= 1.268\ngini = 0.278
\nsamples = 6\nvalue = [1, 5]'),
 Text(0.8757990867579909, 0.5, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(0.8904109589041096, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.8831050228310502, 0.6176470588235294, 'gini = 0.0\nsamples = 7\nval
ue = [7, 0]'),
 Text(0.9506849315068493, 0.7941176470588235, 'x[24] <= 1.028\ngini = 0.495
\nsamples = 78\nvalue = [43, 35]'),
 Text(0.9360730593607306, 0.7352941176470589, 'x[7] <= -0.61\ngini = 0.466
\nsamples = 65\nvalue = [41, 24]'),
 Text(0.9287671232876712, 0.6764705882352942, 'gini = 0.0\nsamples = 14\nva
lue = [14, 0]'),
 Text(0.94337899543379, 0.6764705882352942, 'x[22] <= -0.736\ngini = 0.498
\nsamples = 51\nvalue = [27, 24]'),
 Text(0.919634703196347, 0.6176470588235294, 'x[39] <= 1.346\ngini = 0.415
\nsamples = 17\nvalue = [5, 12]'),
 Text(0.9123287671232877, 0.5588235294117647, 'x[7] <= 0.596\ngini = 0.32\n
samples = 15\nvalue = [3, 12]'),
 Text(0.9050228310502283, 0.5, 'x[6] <= -1.118\ngini = 0.49\nsamples = 7\nv
alue = [3, 4]'),
 Text(0.897716894977169, 0.4411764705882353, 'gini = 0.0\nsamples = 3\nvalu
e = [0, 3]'),
 Text(0.9123287671232877, 0.4411764705882353, 'x[11] <= -0.753\ngini = 0.37
5\nsamples = 4\nvalue = [3, 1]'),
 Text(0.9050228310502283, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nva
lue = [0, 1]'),
 Text(0.919634703196347, 0.38235294117647056, 'gini = 0.0\nsamples = 3\nval
ue = [3, 0]'),
 Text(0.919634703196347, 0.5, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
 Text(0.9269406392694064, 0.5588235294117647, 'gini = 0.0\nsamples = 2\nval
ue = [2, 0]'),
 Text(0.9671232876712329, 0.6176470588235294, 'x[20] <= 0.544\ngini = 0.457
\nsamples = 34\nvalue = [22, 12]'),
 Text(0.9488584474885845, 0.5588235294117647, 'x[16] <= -1.122\ngini = 0.4
\nsamples = 29\nvalue = [21, 8]'),
 Text(0.9342465753424658, 0.5, 'x[7] <= 0.473\ngini = 0.469\nsamples = 8\nv
alue = [3, 5]'),
 Text(0.9269406392694064, 0.4411764705882353, 'gini = 0.0\nsamples = 3\nval
ue = [3, 0]'),
 Text(0.9415525114155251, 0.4411764705882353, 'gini = 0.0\nsamples = 5\nval
ue = [0, 5]'),
 Text(0.9634703196347032, 0.5, 'x[8] <= 1.083\ngini = 0.245\nsamples = 21\n
value = [18, 3]'),
 Text(0.9561643835616438, 0.4411764705882353, 'gini = 0.0\nsamples = 17\nva
lue = [17, 0]'),
 Text(0.9707762557077626, 0.4411764705882353, 'x[29] <= 0.428\ngini = 0.375
\nsamples = 4\nvalue = [1, 3]'),
 Text(0.9634703196347032, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nva
lue = [1, 0]'),
 Text(0.9780821917808219, 0.38235294117647056, 'gini = 0.0\nsamples = 3\nva
lue = [0, 3]'),
 Text(0.9853881278538813, 0.5588235294117647, 'x[11] <= -0.428\ngini = 0.32
\nsamples = 5\nvalue = [1, 4]'),
 Text(0.9780821917808219, 0.5, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.9926940639269406, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.965296803652968, 0.7352941176470589, 'x[7] <= 0.989\ngini = 0.26\ns
amples = 13\nvalue = [2, 11]'),
 Text(0.9579908675799087, 0.6764705882352942, 'gini = 0.0\nsamples = 11\nva
lue = [0, 11]'),
 Text(0.9726027397260274, 0.6764705882352942, 'gini = 0.0\nsamples = 2\nval
ue = [2, 0]')]
```



In [52]:

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

}
```

Loading [MathJax]/extensions/Safe.js

In [53]:
```
grid_search=GridSearchCV(estimator=dtc,param_grid=parameter,cv=5,scoring="ac
```

In [54]:
```
grid_search.fit(x_train,y_train)
```

```
C:\Users\NAGA BHAVANI\anaconda3\Lib\site-packages\sklearn\model_selection\_v
alidation.py:425: FitFailedWarning:
100 fits failed out of a total of 300.
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 err
or_score='raise'.

Below are more details about the failures:
----------------------------------------------------------------------------
----
100 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\NAGA BHAVANI\anaconda3\Lib\site-packages\sklearn\model_sele
ction\_validation.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\NAGA BHAVANI\anaconda3\Lib\site-packages\sklearn\base.py",
line 1144, in wrapper
    estimator._validate_params()
  File "C:\Users\NAGA BHAVANI\anaconda3\Lib\site-packages\sklearn\base.py",
line 637, in _validate_params
    validate_parameter_constraints(
  File "C:\Users\NAGA BHAVANI\anaconda3\Lib\site-packages\sklearn\utils\_par
am_validation.py", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'max_features' pa
rameter of DecisionTreeClassifier must be an int in the range [1, inf), a fl
oat in the range (0.0, 1.0], a str among {'log2', 'sqrt'} or None. Got 'aut
o' instead.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\Users\NAGA BHAVANI\anaconda3\Lib\site-packages\sklearn\model_selection\_s
earch.py:976: UserWarning: One or more of the test scores are non-finite: [
nan        nan 0.83163361 0.83163361 0.83163361 0.83163361
        nan        nan 0.83248107 0.82737829 0.82568338 0.83588893
        nan        nan 0.82567977 0.82567616 0.83505229 0.8273855
        nan        nan 0.83503065 0.82652362 0.83671475 0.82060224
        nan        nan 0.82228273 0.83335016 0.81635052 0.82822575
        nan        nan 0.83163361 0.83163361 0.82993148 0.83163361
        nan        nan 0.82822935 0.83503787 0.83248467 0.83078255
        nan        nan 0.83843851 0.82654165 0.82485034 0.82993148
        nan        nan 0.8384313  0.83163361 0.83162279 0.8248251
        nan        nan 0.81379373 0.83759466 0.83247746 0.82991345]
  warnings.warn(
```

Out[54]:

▸ **GridSearchCV**

▸ **estimator: DecisionTreeClassifier**

▸ DecisionTreeClassifier

Loading [MathJax]/extensions/Safe.js

In [55]:
```
grid_search.best_params_
```

Out[55]:
```
{'criterion': 'entropy',
 'max_depth': 3,
 'max_features': 'sqrt',
 'splitter': 'best'}
```

In [56]:
```
dtc_cv=DecisionTreeClassifier(criterion= 'entropy',
 max_depth=3,
 max_features='sqrt',
 splitter='best')
dtc_cv.fit(x_train,y_train)
```

Out[56]:
```
▼                     DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=3, max_featur
es='sqrt')
```

In [57]:
```
pred=dtc_cv.predict(x_test)
```

In [58]:
```
print(classification_report(y_test,pred))
```

```
              precision    recall  f1-score   support

          No       0.87      1.00      0.93       255
         Yes       0.00      0.00      0.00        39

    accuracy                           0.87       294
   macro avg       0.43      0.50      0.46       294
weighted avg       0.75      0.87      0.81       294
```

```
C:\Users\NAGA BHAVANI\anaconda3\Lib\site-packages\sklearn\metrics\_classific
ation.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined
and being set to 0.0 in labels with no predicted samples. Use `zero_division
` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\NAGA BHAVANI\anaconda3\Lib\site-packages\sklearn\metrics\_classific
ation.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined
and being set to 0.0 in labels with no predicted samples. Use `zero_division
` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\NAGA BHAVANI\anaconda3\Lib\site-packages\sklearn\metrics\_classific
ation.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defined
and being set to 0.0 in labels with no predicted samples. Use `zero_division
` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

RANDOM FOREST

In [59]:
```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 1000, criterion = 'entrop
classifier.fit(x_train, y_train)
```

Loading [MathJax]/extensions/Safe.js

Out[59]:

```
▼                  RandomForestClassifier
RandomForestClassifier(criterion='entropy', n_estimators=1000, rand
om_state=0)
```

In [60]:

```
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)
```

```
[[255   0]
 [ 35   4]]
```

Out[60]:

```
0.8809523809523809
```

In [61]:

```
from sklearn.ensemble import RandomForestClassifier
```

In [62]:

```
rfc=RandomForestClassifier()
```

In [63]:

```
forest_params = [{'max_depth': list(range(10, 15)), 'max_features': list(rar
```

In [64]:

```
rfc_cv=GridSearchCV(rfc,param_grid=forest_params,cv=10,scoring="accuracy")
```

In [72]:

```
rfc_cv.fit(x_train,y_train)
```

```
C:\Users\NAGA BHAVANI\anaconda3\Lib\site-packages\sklearn\model_selection\_v
alidation.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 err
or_score='raise'.

Below are more details about the failures:
----------------------------------------------------------------------------
----
50 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\NAGA BHAVANI\anaconda3\Lib\site-packages\sklearn\model_sele
ction\_validation.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\NAGA BHAVANI\anaconda3\Lib\site-packages\sklearn\base.py",
line 1144, in wrapper
    estimator._validate_params()
  File "C:\Users\NAGA BHAVANI\anaconda3\Lib\site-packages\sklearn\base.py",
line 637, in _validate_params
    validate_parameter_constraints(
  File "C:\Users\NAGA BHAVANI\anaconda3\Lib\site-packages\sklearn\utils\_par
am_validation.py", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'max_features' pa
rameter of RandomForestClassifier must be an int in the range [1, inf), a fl
oat in the range (0.0, 1.0], a str among {'log2', 'sqrt'} or None. Got 0 ins
tead.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\Users\NAGA BHAVANI\anaconda3\Lib\site-packages\sklearn\model_selection\_s
earch.py:976: UserWarning: One or more of the test scores are non-finite: [
nan 0.84014921 0.8478053  0.84610314 0.84950022 0.85034767
 0.84863827 0.85714907 0.85799652 0.85627988 0.85799652 0.85120238
 0.85457772 0.85971317        nan 0.83759959 0.84355353 0.85120962
 0.84694336 0.84610314 0.85545415 0.85120238 0.85203535 0.85800377
 0.85799652 0.85967695 0.85969144 0.85884398        nan 0.8427278
 0.8452412  0.84865276 0.84440099 0.8478053  0.85203535 0.85205708
 0.85289729 0.85204259 0.85884398 0.85800377 0.86137187 0.85630885
        nan 0.83844705 0.84610314 0.84866725 0.84952919 0.85461394
 0.8528828  0.85204259 0.85034043 0.85798928 0.8622483  0.85798204
 0.85712734 0.85118065        nan 0.83928727 0.84269883 0.84099667
 0.84693611 0.85204983 0.85546864 0.85801101 0.85629436 0.85712734
 0.85460669 0.85373026 0.85457048 0.85885122]
  warnings.warn(
```

Out[72]:

```
▸ GridSearchCV
▸ estimator: RandomForestClassifier
    ▸ RandomForestClassifier
```

In [73]:

```
pred=rfc_cv.predict(x_test)
```

Loading [MathJax]/extensions/Safe.js

```
In [68]: print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

          No       0.88      1.00      0.94       255
         Yes       1.00      0.10      0.19        39

    accuracy                           0.88       294
   macro avg       0.94      0.55      0.56       294
weighted avg       0.90      0.88      0.84       294
```

```
In [74]: rfc_cv.best_params_
```

```
Out[74]: {'max_depth': 13, 'max_features': 10}
```

```
In [ ]:
```

Loading [MathJax]/extensions/Safe.js