Data Preprocessing

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

In [2]:
```
df=pd.read_csv("D:\AI intern\WA_Fn-UseC_-HR-Employee-Attrition.csv")
```

In [3]:
```
df.head()
```

Out[3]:

	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 [4]:
```
df.shape
```

Out[4]:
```
(1470, 35)
```

In [5]:
```
df.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 [6]:
```
df.describe()
```

Out[6]:

	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

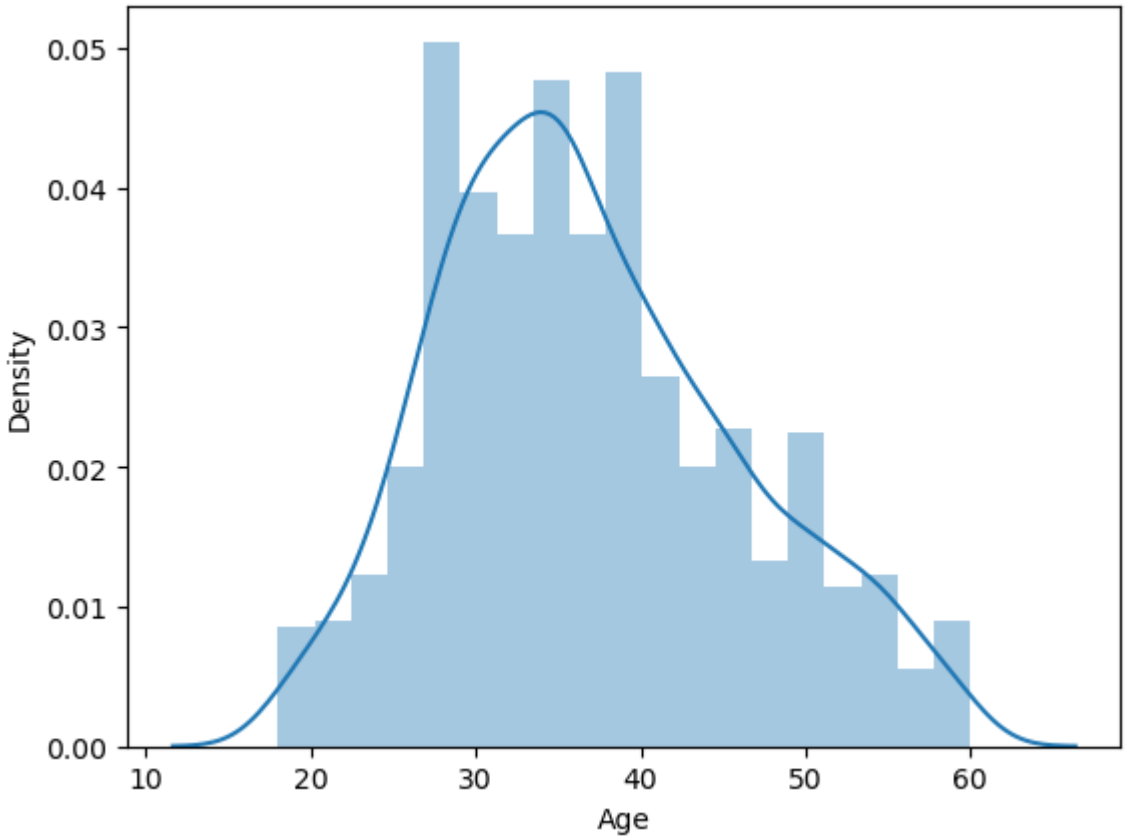
In [7]:
```
#Checking for Null Values.
df.isnull().any()
```

Out[7]:
```
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 [8]:
```
#Data Visualization.
sns.distplot(df["Age"])
```

```
C:\Users\saisa\AppData\Local\Temp\ipykernel_15996\2400079689.py:2: UserWarning:

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

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

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

  sns.distplot(df["Age"])
```

Out[8]: <Axes: xlabel='Age', ylabel='Density'>



In [9]:
```
df.corr()
```

```
C:\Users\saisa\AppData\Local\Temp\ipykernel_15996\1134722465.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.
  df.corr()
```

Out[9]:

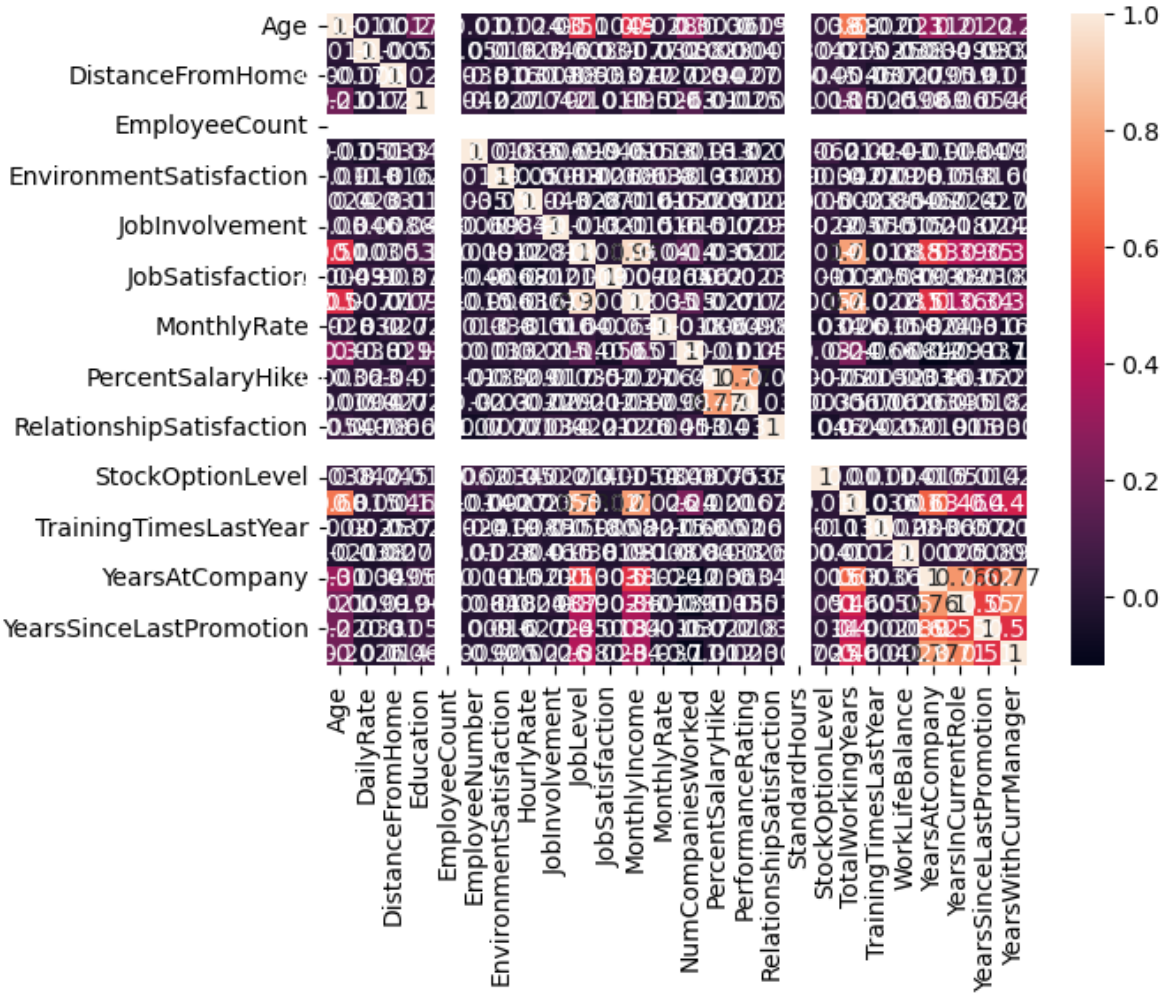
	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	En
Age	1.000000	0.010661	-0.001686	0.208034	NaN	
DailyRate	0.010661	1.000000	-0.004985	-0.016806	NaN	
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042	NaN	
Education	0.208034	-0.016806	0.021042	1.000000	NaN	
EmployeeCount	NaN	NaN	NaN	NaN	NaN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

26 rows × 26 columns

In [10]:
```
sns.heatmap(df.corr(),annot=True)
```

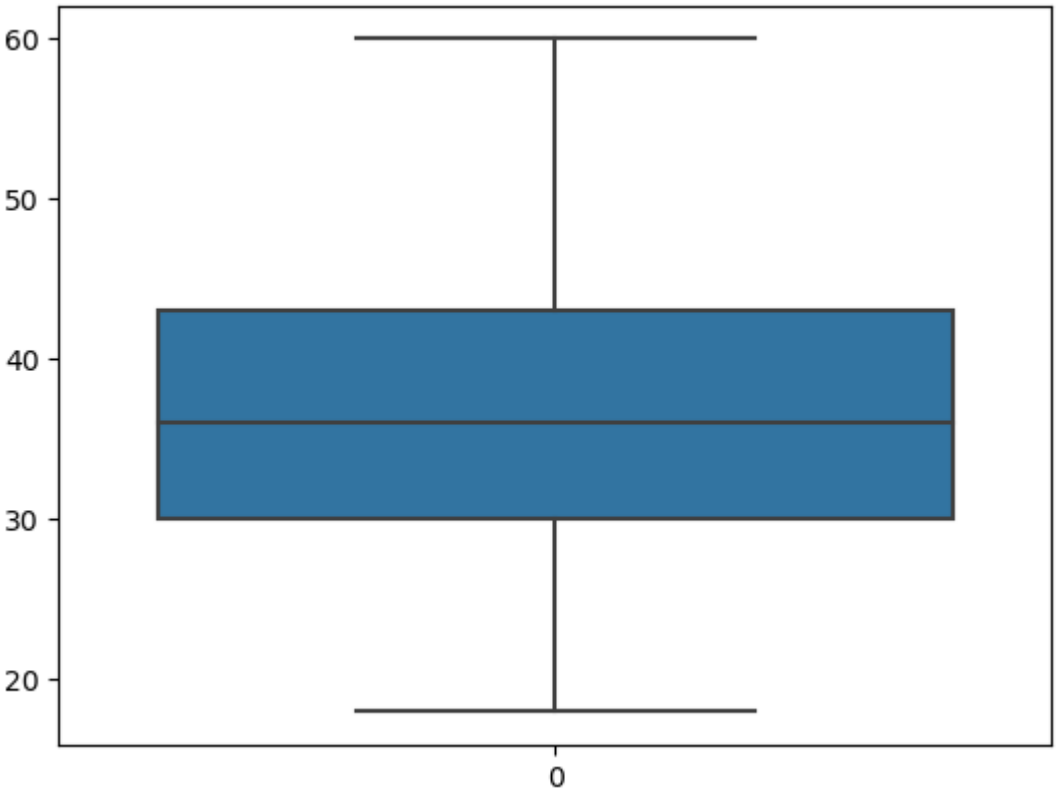
```
C:\Users\saisa\AppData\Local\Temp\ipykernel_15996\4277794465.py:1: FutureWarning:
The default value of numeric_only in DataFrame.corr is deprecated. In a future ver
sion, it will default to False. Select only valid columns or specify the value of
numeric_only to silence this warning.
  sns.heatmap(df.corr(),annot=True)
```

Out[10]:
```
<Axes: >
```



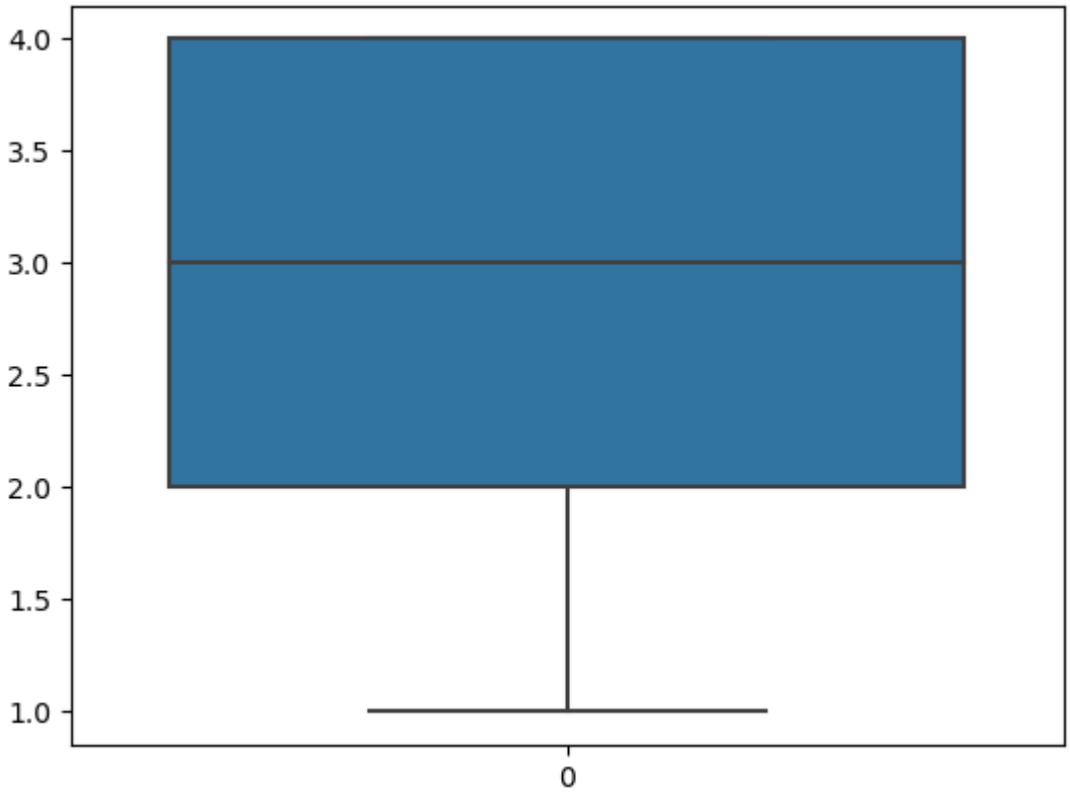
In [11]:
```
sns.boxplot(df.Age)
```

Out[11]: <Axes: >



In [12]:
```
sns.boxplot(df.JobSatisfaction)
```

Out[12]: <Axes: >



In [13]:
```
#Splitting Dependent and Independent variables
x = df.iloc[:, [0] + list(range(2, df.shape[1]))]
x.head()
```

Out[13]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField
0	41	Travel_Rarely	1102	Sales	1	2	Life Sciences
1	49	Travel_Frequently	279	Research & Development	8	1	Life Sciences
2	37	Travel_Rarely	1373	Research & Development	2	2	Other
3	33	Travel_Frequently	1392	Research & Development	3	4	Life Sciences
4	27	Travel_Rarely	591	Research & Development	2	1	Medical

5 rows × 34 columns

In [14]:
```
y=df["Attrition"]
y.head()
```

Out[14]:
```
0    Yes
1     No
2    Yes
3     No
4     No
Name: Attrition, dtype: object
```

In [15]:
```
#Label encoding
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
x.BusinessTravel=le.fit_transform(x.BusinessTravel)
x.head()
```

```
C:\Users\saisa\AppData\Local\Temp\ipykernel_15996\2771569609.py:4: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  x.BusinessTravel=le.fit_transform(x.BusinessTravel)
```

Out[15]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	E
0	41	2	1102	Sales	1	2	Life Sciences	
1	49	1	279	Research & Development	8	1	Life Sciences	
2	37	2	1373	Research & Development	2	2	Other	
3	33	1	1392	Research & Development	3	4	Life Sciences	
4	27	2	591	Research & Development	2	1	Medical	

5 rows × 34 columns

In [16]:

```
x.Department=le.fit_transform(x.Department)
x.head()
```

```
C:\Users\saisa\AppData\Local\Temp\ipykernel_15996\82449547.py:1: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  x.Department=le.fit_transform(x.Department)
```

Out[16]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	Er
0	41	2	1102	2	1	2	Life Sciences	
1	49	1	279	1	8	1	Life Sciences	
2	37	2	1373	1	2	2	Other	
3	33	1	1392	1	3	4	Life Sciences	
4	27	2	591	1	2	1	Medical	

5 rows × 34 columns

In [17]:

```
x.EducationField=le.fit_transform(x.EducationField)
x.head()
```

```
C:\Users\saisa\AppData\Local\Temp\ipykernel_15996\2936879973.py:1: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  x.EducationField=le.fit_transform(x.EducationField)
```

Out[17]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	Er
0	41	2	1102	2	1	2	1	
1	49	1	279	1	8	1	1	
2	37	2	1373	1	2	2	4	
3	33	1	1392	1	3	4	1	
4	27	2	591	1	2	1	3	

5 rows × 34 columns

In [18]:
```
x.Gender=le.fit_transform(x.Gender)
x.head()
```

```
C:\Users\saisa\AppData\Local\Temp\ipykernel_15996\30094682.py:1: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  x.Gender=le.fit_transform(x.Gender)
```

Out[18]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	Er
0	41	2	1102	2	1	2	1	
1	49	1	279	1	8	1	1	
2	37	2	1373	1	2	2	4	
3	33	1	1392	1	3	4	1	
4	27	2	591	1	2	1	3	

5 rows × 34 columns

In [19]:
```
x.JobRole=le.fit_transform(x.JobRole)
x.head()
```

```
C:\Users\saisa\AppData\Local\Temp\ipykernel_15996\1892396863.py:1: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  x.JobRole=le.fit_transform(x.JobRole)
```

Out[19]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	Er
0	41	2	1102	2	1	2	1	
1	49	1	279	1	8	1	1	
2	37	2	1373	1	2	2	4	
3	33	1	1392	1	3	4	1	
4	27	2	591	1	2	1	3	

5 rows × 34 columns

In [20]:
```
x.MaritalStatus=le.fit_transform(x.MaritalStatus)
x.head()
```

```
C:\Users\saisa\AppData\Local\Temp\ipykernel_15996\1557157179.py:1: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  x.MaritalStatus=le.fit_transform(x.MaritalStatus)
```

Out[20]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	Er
0	41	2	1102	2	1	2	1	
1	49	1	279	1	8	1	1	
2	37	2	1373	1	2	2	4	
3	33	1	1392	1	3	4	1	
4	27	2	591	1	2	1	3	

5 rows × 34 columns

In [21]:
```
x.Over18=le.fit_transform(x.Over18)
x.head()
```

```
C:\Users\saisa\AppData\Local\Temp\ipykernel_15996\749637236.py:1: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  x.Over18=le.fit_transform(x.Over18)
```

Out[21]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	Er
0	41	2	1102	2	1	2	1	
1	49	1	279	1	8	1	1	
2	37	2	1373	1	2	2	4	
3	33	1	1392	1	3	4	1	
4	27	2	591	1	2	1	3	

5 rows × 34 columns

In [22]:
```
x.OverTime=le.fit_transform(x.OverTime)
x.head()
```

```
C:\Users\saisa\AppData\Local\Temp\ipykernel_15996\2729453452.py:1: SettingWithCopy
Warning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  x.OverTime=le.fit_transform(x.OverTime)
```

Out[22]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	Er
0	41	2	1102	2	1	2	1	
1	49	1	279	1	8	1	1	
2	37	2	1373	1	2	2	4	
3	33	1	1392	1	3	4	1	
4	27	2	591	1	2	1	3	

5 rows × 34 columns

In [23]:
```
#feature scaling
from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
x_scaled=pd.DataFrame(ms.fit_transform(x),columns=x.columns)
```

In [24]:
```
x_scaled
```

Out[24]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationF
0	0.547619	1.0	0.715820	1.0	0.000000	0.25	
1	0.738095	0.5	0.126700	0.5	0.250000	0.00	
2	0.452381	1.0	0.909807	0.5	0.035714	0.25	
3	0.357143	0.5	0.923407	0.5	0.071429	0.75	
4	0.214286	1.0	0.350036	0.5	0.035714	0.00	
...	...	...	...	...	...	...	
1465	0.428571	0.5	0.559771	0.5	0.785714	0.25	
1466	0.500000	1.0	0.365784	0.5	0.178571	0.00	
1467	0.214286	1.0	0.037938	0.5	0.107143	0.50	
1468	0.738095	0.5	0.659270	1.0	0.035714	0.50	
1469	0.380952	1.0	0.376521	0.5	0.250000	0.50	

1470 rows × 34 columns

In [25]:
```
#Splitting Data into Train and Test.
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_stat
```

In [26]:
```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[26]:
```
((1176, 34), (294, 34), (1176,), (294,))
```

In [27]:
```
x_train.head()
```

Out[27]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationF
1374	0.952381	1.0	0.360057	1.0	0.714286	0.50	
1092	0.642857	1.0	0.607015	0.5	0.964286	0.50	
768	0.523810	1.0	0.141732	1.0	0.892857	0.50	
569	0.428571	0.0	0.953472	1.0	0.250000	0.75	
911	0.166667	0.5	0.355762	1.0	0.821429	0.00	

5 rows × 34 columns

Model Builiding(Logistic Regression)

In [28]:
```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
```

In [29]:
```
model.fit(x_train,y_train)
```

Out[29]:
```
▾ LogisticRegression

LogisticRegression()
```

In [30]:
```
pred=model.predict(x_test)
```

In [31]:
```
pred
```

Out[31]:
```
array(['No', 'No', 'No', 'No', 'Yes', 'No', 'Yes', 'No', 'No', 'No', 'No',
       'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'Yes', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'No', 'No', 'No', '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', 'Yes', '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', 'No', 'No',
       'No', 'Yes', 'No', 'No', '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', 'Yes',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No',
       'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', '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', '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',
       'Yes', 'No', 'No', 'No', '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', 'Yes', 'No', 'No'],
      dtype=object)
```

In [32]:
```
y_test
```

Out[32]:
```
442      No
1091     No
981     Yes
785      No
1332    Yes
       ...
1439     No
481      No
124     Yes
198      No
1229     No
Name: Attrition, Length: 294, dtype: object
```

Evaluation of Classification model

In [33]:
```
#Accuracy score
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,r
```

In [34]:
```
accuracy_score(y_test,pred)
```

Out[34]:
```
0.8843537414965986
```

In [35]:
```
confusion_matrix(y_test,pred)
```

Out[35]:
```
array([[242,   3],
       [ 31,  18]], dtype=int64)
```

In [36]:
```
pd.crosstab(y_test,pred)
```

Out[36]:

col_0	No	Yes
Attrition		
No	242	3
Yes	31	18

In [37]:
```
print(classification_report(y_test,pred))
```

```
              precision    recall  f1-score   support

          No       0.89      0.99      0.93       245
         Yes       0.86      0.37      0.51        49

    accuracy                           0.88       294
   macro avg       0.87      0.68      0.72       294
weighted avg       0.88      0.88      0.86       294
```

In [38]:
```
probability=model.predict_proba(x_test)[:,1]
```

In [39]:
```
probability
```

Out[39]:
```
array([0.16000127, 0.20600667, 0.31532384, 0.09242886, 0.63667551,
       0.06153061, 0.61819432, 0.0757087 , 0.00841372, 0.3912069 ,
       0.05398439, 0.33293123, 0.02020698, 0.67215483, 0.19786547,
       0.03454902, 0.11043981, 0.17101703, 0.04477777, 0.22783614,
       0.2335018 , 0.01553905, 0.06464492, 0.05029956, 0.58792413,
       0.44849464, 0.07412714, 0.04460935, 0.67666632, 0.0584383 ,
       0.01599026, 0.03521098, 0.06963085, 0.17397462, 0.07830857,
       0.04288032, 0.08150424, 0.07106342, 0.03622137, 0.05223965,
       0.04862098, 0.02091497, 0.01819361, 0.01362467, 0.02873997,
       0.50236969, 0.41553218, 0.00306874, 0.73976412, 0.51382382,
       0.09637213, 0.48845516, 0.08036228, 0.25757243, 0.66516772,
       0.26308027, 0.01964858, 0.30198497, 0.02919946, 0.16038964,
       0.02102747, 0.21670232, 0.13981568, 0.0358316 , 0.37208403,
       0.03002317, 0.29091186, 0.16041142, 0.10437497, 0.08695177,
       0.08217589, 0.30984518, 0.08531362, 0.07420689, 0.12268651,
       0.06192552, 0.04640904, 0.07624712, 0.19738483, 0.03236316,
       0.00884439, 0.0244108 , 0.13635803, 0.0260104 , 0.03341008,
       0.08186888, 0.00499397, 0.03474852, 0.03858027, 0.14602694,
       0.26167665, 0.16667357, 0.27400109, 0.24159565, 0.02160421,
       0.17748606, 0.34076078, 0.28022482, 0.06914126, 0.05003806,
       0.24437761, 0.74698271, 0.35438567, 0.01920627, 0.08778845,
       0.03255847, 0.05461351, 0.15123251, 0.06843702, 0.13752637,
       0.09584388, 0.04669882, 0.02493091, 0.15383171, 0.07081259,
       0.03089296, 0.0537667 , 0.11554316, 0.00881616, 0.01263271,
       0.17552253, 0.05045234, 0.08823238, 0.82995757, 0.03017756,
       0.0236819 , 0.0087012 , 0.1349589 , 0.16474801, 0.05202613,
       0.01524549, 0.29278083, 0.54767448, 0.34275448, 0.04629541,
       0.38966344, 0.61333366, 0.14552367, 0.07402366, 0.24143471,
       0.09418418, 0.0689069 , 0.10061956, 0.19346327, 0.20026293,
       0.03004939, 0.14900424, 0.00348846, 0.11225149, 0.15843155,
       0.06047573, 0.18601882, 0.06085869, 0.12221317, 0.03280184,
       0.02738799, 0.06356425, 0.08302382, 0.01541716, 0.014665  ,
       0.38517822, 0.01264231, 0.14961974, 0.80508787, 0.11598661,
       0.2842811 , 0.17020143, 0.1530583 , 0.02764153, 0.00613226,
       0.04191632, 0.09782393, 0.11551417, 0.10377982, 0.01779313,
       0.14371315, 0.10615435, 0.10298963, 0.05132621, 0.09061081,
       0.02897383, 0.09924087, 0.00512032, 0.75108423, 0.04296968,
       0.04062134, 0.37518972, 0.04563128, 0.7251816 , 0.10671665,
       0.36949086, 0.38146941, 0.32095493, 0.05266802, 0.08172004,
       0.13947833, 0.04334317, 0.01469593, 0.26413988, 0.06330966,
       0.1614747 , 0.15380517, 0.67152357, 0.05840793, 0.27891823,
       0.04512564, 0.46033865, 0.00348431, 0.14068967, 0.02747401,
       0.12714133, 0.17284246, 0.07341066, 0.10099827, 0.16870885,
       0.02560842, 0.01824031, 0.08670796, 0.02834237, 0.13710215,
       0.08778935, 0.2200061 , 0.73401148, 0.15938978, 0.4095449 ,
       0.01513845, 0.11306309, 0.21497506, 0.32337575, 0.03409266,
       0.04256318, 0.32157531, 0.05454465, 0.02348479, 0.16423352,
       0.32696147, 0.22892063, 0.00877159, 0.08198819, 0.01156361,
       0.1408691 , 0.29235147, 0.01270305, 0.17329916, 0.04081391,
       0.04094165, 0.42771425, 0.34958286, 0.03766772, 0.12025286,
       0.37698923, 0.3192629 , 0.79559338, 0.05385659, 0.21597037,
       0.06383728, 0.00570991, 0.66018187, 0.35855286, 0.37783606,
       0.36781398, 0.03554512, 0.21718203, 0.05943622, 0.06554485,
       0.10081475, 0.00818713, 0.26591316, 0.42809675, 0.06542835,
       0.09296803, 0.01259826, 0.14226651, 0.05072662, 0.02372258,
       0.02586923, 0.06760427, 0.24315648, 0.26961432, 0.19831733,
       0.2652296 , 0.0165923 , 0.15784236, 0.08398982, 0.02711775,
       0.18750547, 0.00783535, 0.2844239 , 0.00270742, 0.02484969,
       0.22585745, 0.72775605, 0.07691547, 0.26304359])
```

In [40]:
```
# roc_curve
fpr,tpr,threshsholds = roc_curve(y_test,probability)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[40], line 2
      1 # roc_curve
----> 2 fpr,tpr,threshsholds = roc_curve(y_test,probability)

File ~\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.py:211, in validate_params.<locals>.decorator.<locals>.wrapper(*args, **kwargs)
    205 try:
    206     with config_context(
    207         skip_parameter_validation=(
    208             prefer_skip_nested_validation or global_skip_validation
    209         )
    210     ):
--> 211         return func(*args, **kwargs)
    212 except InvalidParameterError as e:
    213     # When the function is just a wrapper around an estimator, we allow
    214     # the function to delegate validation to the estimator, but we replace
    215     # the name of the estimator by the name of the function in the error
    216     # message to avoid confusion.
    217     msg = re.sub(
    218         r"parameter of \w+ must be",
    219         f"parameter of {func.__qualname__} must be",
    220         str(e),
    221     )

File ~\anaconda3\Lib\site-packages\sklearn\metrics\_ranking.py:1094, in roc_curve(y_true, y_score, pos_label, sample_weight, drop_intermediate)
    992 @validate_params(
    993     {
    994         "y_true": ["array-like"],
   (...)
   1003     y_true, y_score, *, pos_label=None, sample_weight=None, drop_intermediate=True
   1004 ):
   1005     """Compute Receiver operating characteristic (ROC).
   1006
   1007     Note: this implementation is restricted to the binary classification task.
   (...)
   1092     array([ inf, 0.8 , 0.4 , 0.35, 0.1 ])
   1093     """
-> 1094     fps, tps, thresholds = _binary_clf_curve(
   1095         y_true, y_score, pos_label=pos_label, sample_weight=sample_weight
   1096     )
   1098     # Attempt to drop thresholds corresponding to points in between and
   1099     # collinear with other points. These are always suboptimal and do not
   1100     # appear on a plotted ROC curve (and thus do not affect the AUC).
   (...)
   1105     # but does not drop more complicated cases like fps = [1, 3, 7],
   1106     # tps = [1, 2, 4]; there is no harm in keeping too many thresholds.
   1107     if drop_intermediate and len(fps) > 2:

File ~\anaconda3\Lib\site-packages\sklearn\metrics\_ranking.py:820, in _binary_clf_curve(y_true, y_score, pos_label, sample_weight)
    817     y_score = y_score[nonzero_weight_mask]
    818     sample_weight = sample_weight[nonzero_weight_mask]
--> 820 pos_label = _check_pos_label_consistency(pos_label, y_true)
    822 # make y_true a boolean vector
    823 y_true = y_true == pos_label

File ~\anaconda3\Lib\site-packages\sklearn\utils\validation.py:2246, in _check_pos_label_consistency(pos_label, y_true)
   2235 if pos_label is None and (
```

```
   2236     classes.dtype.kind in "OUS"
   2237     or not (
   (...)
   2243     )
   2244 ):
   2245     classes_repr = ", ".join(repr(c) for c in classes)
-> 2246     raise ValueError(
   2247         f"y_true takes value in {{{classes_repr}}} and pos_label is not "
   2248         "specified: either make y_true take value in {0, 1} or "
   2249         "{-1, 1} or pass pos_label explicitly."
   2250     )
   2251 elif pos_label is None:
   2252     pos_label = 1

ValueError: y_true takes value in {'No', 'Yes'} and pos_label is not specified: ei
ther make y_true take value in {0, 1} or {-1, 1} or pass pos_label explicitly.
```

In [41]:
```
plt.plot(fpr,tpr)
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('ROC CURVE')
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[41], line 1
----> 1 plt.plot(fpr,tpr)
      2 plt.xlabel('FPR')
      3 plt.ylabel('TPR')

NameError: name 'fpr' is not defined
```

Model Building(Decision Tree)

In [42]:
```
from sklearn.tree import DecisionTreeClassifier
dtc=DecisionTreeClassifier()
```

In [43]:
```
dtc.fit(x_train,y_train)
```

Out[43]:
```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

In [44]:
```
pred=dtc.predict(x_test)
```

In [45]:
```
pred
```

Out[45]:
```
array(['No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No',
       'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No',
       'No', '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', 'Yes', 'No', 'Yes', 'Yes', 'No', 'No',
       'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No',
       'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No',
       'Yes', 'Yes', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No',
       'No', 'Yes', 'No', 'No', 'No', 'Yes', 'No', 'No', 'Yes', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'Yes', 'Yes', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'Yes', 'No', 'No', 'No',
       'No', 'No', 'No', 'Yes', 'Yes', 'Yes', 'Yes', 'No', 'No', 'No',
       'Yes', 'No', 'No', 'No', 'Yes', 'No', 'Yes', 'No', 'No', 'Yes',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'Yes', 'Yes', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No',
       'Yes', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'Yes',
       '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', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'Yes', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'No', 'Yes',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No'], dtype=object)
```

In [46]:
```
y_test
```

Out[46]:
```
442      No
1091     No
981     Yes
785      No
1332    Yes
       ...
1439     No
481      No
124     Yes
198      No
1229     No
Name: Attrition, Length: 294, dtype: object
```

In [47]:
```
#Accuracy score
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,r
```

In [48]:
```
accuracy_score(y_test,pred)
```

Out[48]:
```
0.7517006802721088
```

In [49]:
```
confusion_matrix(y_test,pred)
```

Out[49]:
```
array([[207,  38],
       [ 35,  14]], dtype=int64)
```

In [50]:
```
pd.crosstab(y_test,pred)
```

Out[50]:

col_0	No	Yes
Attrition		
No	207	38
Yes	35	14

In [51]:
```
print(classification_report(y_test,pred))
```

```
              precision    recall  f1-score   support

          No       0.86      0.84      0.85       245
         Yes       0.27      0.29      0.28        49

    accuracy                           0.75       294
   macro avg       0.56      0.57      0.56       294
weighted avg       0.76      0.75      0.75       294
```

In [52]:
```
probability=dtc.predict_proba(x_test)[:,1]
```

In [53]:
```
probability
```

Out[53]:
```
array([0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0.,
       0., 0., 0., 1., 0., 0., 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., 1., 0., 1.,
       1., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 0., 0.,
       0., 0., 0., 1., 0., 0., 0., 1., 1., 0., 0., 0., 1., 0., 0., 0., 0.,
       0., 1., 0., 0., 0., 1., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0.,
       1., 0., 0., 0., 0., 0., 0., 0., 1., 1., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 1., 1., 0., 0., 0., 0., 0., 0., 1., 1., 1., 1., 0.,
       0., 0., 1., 0., 0., 0., 1., 0., 1., 0., 0., 1., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 1., 0., 1., 1., 0., 0., 0., 0., 1., 0., 0., 1.,
       0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0.,
       1., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0.,
       0., 1., 0., 0., 0., 0., 1., 0., 0., 0., 1., 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., 1., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 1., 1., 0., 1., 0., 0., 0.,
       0., 0., 0., 0., 0.])
```

In [54]:
```
# roc_curve
fpr,tpr,threshsholds = roc_curve(y_test,probability)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[54], line 2
      1 # roc_curve
----> 2 fpr,tpr,threshsholds = roc_curve(y_test,probability)

File ~\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.py:211, in validate_params.<locals>.decorator.<locals>.wrapper(*args, **kwargs)
    205 try:
    206     with config_context(
    207         skip_parameter_validation=(
    208             prefer_skip_nested_validation or global_skip_validation
    209         )
    210     ):
--> 211         return func(*args, **kwargs)
    212 except InvalidParameterError as e:
    213     # When the function is just a wrapper around an estimator, we allow
    214     # the function to delegate validation to the estimator, but we replace
    215     # the name of the estimator by the name of the function in the error
    216     # message to avoid confusion.
    217     msg = re.sub(
    218         r"parameter of \w+ must be",
    219         f"parameter of {func.__qualname__} must be",
    220         str(e),
    221     )

File ~\anaconda3\Lib\site-packages\sklearn\metrics\_ranking.py:1094, in roc_curve(y_true, y_score, pos_label, sample_weight, drop_intermediate)
    992 @validate_params(
    993     {
    994         "y_true": ["array-like"],
   (...)
   1003     y_true, y_score, *, pos_label=None, sample_weight=None, drop_intermediate=True
   1004 ):
   1005     """Compute Receiver operating characteristic (ROC).
   1006
   1007     Note: this implementation is restricted to the binary classification task.
   (...)
   1092     array([ inf, 0.8 , 0.4 , 0.35, 0.1 ])
   1093     """
-> 1094     fps, tps, thresholds = _binary_clf_curve(
   1095         y_true, y_score, pos_label=pos_label, sample_weight=sample_weight
   1096     )
   1098     # Attempt to drop thresholds corresponding to points in between and
   1099     # collinear with other points. These are always suboptimal and do not
   1100     # appear on a plotted ROC curve (and thus do not affect the AUC).
   (...)
   1105     # but does not drop more complicated cases like fps = [1, 3, 7],
   1106     # tps = [1, 2, 4]; there is no harm in keeping too many thresholds.
   1107     if drop_intermediate and len(fps) > 2:

File ~\anaconda3\Lib\site-packages\sklearn\metrics\_ranking.py:820, in _binary_clf_curve(y_true, y_score, pos_label, sample_weight)
    817     y_score = y_score[nonzero_weight_mask]
    818     sample_weight = sample_weight[nonzero_weight_mask]
--> 820 pos_label = _check_pos_label_consistency(pos_label, y_true)
    822 # make y_true a boolean vector
    823 y_true = y_true == pos_label

File ~\anaconda3\Lib\site-packages\sklearn\utils\validation.py:2246, in _check_pos_label_consistency(pos_label, y_true)
   2235 if pos_label is None and (
```

```
   2236     classes.dtype.kind in "OUS"
   2237     or not (
   (...)
   2243     )
   2244 ):
   2245     classes_repr = ", ".join(repr(c) for c in classes)
-> 2246     raise ValueError(
   2247         f"y_true takes value in {{{classes_repr}}} and pos_label is not "
   2248         "specified: either make y_true take value in {0, 1} or "
   2249         "{-1, 1} or pass pos_label explicitly."
   2250     )
   2251 elif pos_label is None:
   2252     pos_label = 1

ValueError: y_true takes value in {'No', 'Yes'} and pos_label is not specified: ei
ther make y_true take value in {0, 1} or {-1, 1} or pass pos_label explicitly.
```

In [55]:
```
plt.plot(fpr,tpr)
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('ROC CURVE')
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[55], line 1
----> 1 plt.plot(fpr,tpr)
      2 plt.xlabel('FPR')
      3 plt.ylabel('TPR')

NameError: name 'fpr' is not defined
```

In [56]:
```
from sklearn import tree
plt.figure(figsize=(25,15))
tree.plot_tree(dtc,filled=True)
```

Out[56]:
```
[Text(0.3245448253833049, 0.9722222222222222, 'x[27] <= 0.038\ngini = 0.269\nsampl
es = 1176\nvalue = [988, 188]'),
 Text(0.07495741056218058, 0.9166666666666666, 'x[16] <= 0.75\ngini = 0.5\nsamples
= 78\nvalue = [39, 39]'),
 Text(0.044293015332197615, 0.8611111111111112, 'x[4] <= 0.554\ngini = 0.426\nsamp
les = 39\nvalue = [27, 12]'),
 Text(0.027257240204429302, 0.8055555555555556, 'x[15] <= 0.167\ngini = 0.312\nsam
ples = 31\nvalue = [25, 6]'),
 Text(0.013628620102214651, 0.75, 'x[9] <= 0.5\ngini = 0.49\nsamples = 7\nvalue =
[3, 4]'),
 Text(0.006814310051107325, 0.6944444444444444, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.020442930153321975, 0.6944444444444444, 'x[23] <= 0.5\ngini = 0.375\nsampl
es = 4\nvalue = [3, 1]'),
 Text(0.013628620102214651, 0.6388888888888888, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
 Text(0.027257240204429302, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.04088586030664395, 0.75, 'x[19] <= 0.056\ngini = 0.153\nsamples = 24\nvalu
e = [22, 2]'),
 Text(0.034071550255536626, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.04770017035775128, 0.6944444444444444, 'x[9] <= 0.167\ngini = 0.083\nsampl
es = 23\nvalue = [22, 1]'),
 Text(0.04088586030664395, 0.6388888888888888, 'x[18] <= 0.283\ngini = 0.5\nsample
s = 2\nvalue = [1, 1]'),
 Text(0.034071550255536626, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.04770017035775128, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.054514480408858604, 0.6388888888888888, 'gini = 0.0\nsamples = 21\nvalue =
[21, 0]'),
 Text(0.06132879045996593, 0.8055555555555556, 'x[8] <= 0.385\ngini = 0.375\nsampl
es = 8\nvalue = [2, 6]'),
 Text(0.054514480408858604, 0.75, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.06814310051107325, 0.75, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.10562180579216354, 0.8611111111111112, 'x[11] <= 0.364\ngini = 0.426\nsamp
les = 39\nvalue = [12, 27]'),
 Text(0.08858603066439523, 0.8055555555555556, 'x[17] <= 0.1\ngini = 0.133\nsample
s = 14\nvalue = [1, 13]'),
 Text(0.0817717206132879, 0.75, 'gini = 0.0\nsamples = 13\nvalue = [0, 13]'),
 Text(0.09540034071550256, 0.75, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.12265758091993186, 0.8055555555555556, 'x[8] <= 0.105\ngini = 0.493\nsampl
es = 25\nvalue = [11, 14]'),
 Text(0.10902896081771721, 0.75, 'x[12] <= 0.5\ngini = 0.278\nsamples = 6\nvalue =
[5, 1]'),
 Text(0.10221465076660988, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.11584327086882454, 0.6944444444444444, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]'),
 Text(0.1362862010221465, 0.75, 'x[15] <= 0.5\ngini = 0.432\nsamples = 19\nvalue =
[6, 13]'),
 Text(0.12947189097103917, 0.6944444444444444, 'gini = 0.0\nsamples = 7\nvalue =
[0, 7]'),
 Text(0.14310051107325383, 0.6944444444444444, 'x[6] <= 0.4\ngini = 0.5\nsamples =
12\nvalue = [6, 6]'),
 Text(0.12947189097103917, 0.6388888888888888, 'x[12] <= 0.167\ngini = 0.278\nsamp
les = 6\nvalue = [5, 1]'),
 Text(0.12265758091993186, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.1362862010221465, 0.5833333333333334, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]'),
 Text(0.1567291311754685, 0.6388888888888888, 'x[8] <= 0.249\ngini = 0.278\nsample
s = 6\nvalue = [1, 5]'),
```

```
 Text(0.14991482112436116, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.1635434412265758, 0.5833333333333334, 'gini = 0.0\nsamples = 5\nvalue =
[0, 5]'),
 Text(0.5741322402044293, 0.9166666666666666, 'x[21] <= 0.5\ngini = 0.235\nsamples
= 1098\nvalue = [949, 149]'),
 Text(0.3279919080068143, 0.8611111111111112, 'x[29] <= 0.167\ngini = 0.162\nsampl
es = 798\nvalue = [727, 71]'),
 Text(0.17717206132879046, 0.8055555555555556, 'x[8] <= 0.445\ngini = 0.38\nsample
s = 47\nvalue = [35, 12]'),
 Text(0.1635434412265758, 0.75, 'x[16] <= 0.75\ngini = 0.1\nsamples = 19\nvalue =
[18, 1]'),
 Text(0.1567291311754685, 0.6944444444444444, 'gini = 0.0\nsamples = 18\nvalue =
[18, 0]'),
 Text(0.17035775127768313, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.19080068143100512, 0.75, 'x[17] <= 0.094\ngini = 0.477\nsamples = 28\nvalu
e = [17, 11]'),
 Text(0.1839863713798978, 0.6944444444444444, 'gini = 0.0\nsamples = 4\nvalue =
[0, 4]'),
 Text(0.19761499148211242, 0.6944444444444444, 'x[8] <= 0.524\ngini = 0.413\nsampl
es = 24\nvalue = [17, 7]'),
 Text(0.19080068143100512, 0.6388888888888888, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.20442930153321975, 0.6388888888888888, 'x[33] <= 0.324\ngini = 0.351\nsamp
les = 22\nvalue = [17, 5]'),
 Text(0.19080068143100512, 0.5833333333333334, 'x[2] <= 0.025\ngini = 0.133\nsampl
es = 14\nvalue = [13, 1]'),
 Text(0.1839863713798978, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.19761499148211242, 0.5277777777777778, 'gini = 0.0\nsamples = 13\nvalue =
[13, 0]'),
 Text(0.21805792163543442, 0.5833333333333334, 'x[2] <= 0.329\ngini = 0.5\nsamples
= 8\nvalue = [4, 4]'),
 Text(0.21124361158432708, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.22487223168654175, 0.5277777777777778, 'x[18] <= 0.747\ngini = 0.32\nsampl
es = 5\nvalue = [4, 1]'),
 Text(0.21805792163543442, 0.4722222222222222, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0]'),
 Text(0.23168654173764908, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.47881175468483816, 0.8055555555555556, 'x[27] <= 0.975\ngini = 0.145\nsamp
les = 751\nvalue = [692, 59]'),
 Text(0.47199744463373083, 0.75, 'x[30] <= 0.113\ngini = 0.143\nsamples = 750\nval
ue = [692, 58]'),
 Text(0.3471039182282794, 0.6944444444444444, 'x[9] <= 0.167\ngini = 0.218\nsample
s = 257\nvalue = [225, 32]'),
 Text(0.3049403747870528, 0.6388888888888888, 'x[33] <= 0.147\ngini = 0.355\nsampl
es = 65\nvalue = [50, 15]'),
 Text(0.282793867120954, 0.5833333333333334, 'x[33] <= 0.029\ngini = 0.303\nsample
s = 59\nvalue = [48, 11]'),
 Text(0.25894378194207834, 0.5277777777777778, 'x[12] <= 0.5\ngini = 0.463\nsample
s = 22\nvalue = [14, 8]'),
 Text(0.2453151618398637, 0.4722222222222222, 'x[11] <= 0.179\ngini = 0.198\nsampl
es = 9\nvalue = [8, 1]'),
 Text(0.23850085178875638, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.252129471890971, 0.4166666666666667, 'gini = 0.0\nsamples = 8\nvalue = [8,
0]'),
 Text(0.272572402044293, 0.4722222222222222, 'x[11] <= 0.4\ngini = 0.497\nsamples
= 13\nvalue = [6, 7]'),
 Text(0.2657580919931857, 0.4166666666666667, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0]'),
```

```
 Text(0.27938671209540034, 0.4166666666666667, 'x[4] <= 0.286\ngini = 0.346\nsampl
es = 9\nvalue = [2, 7]'),
 Text(0.272572402044293, 0.3611111111111111, 'x[11] <= 0.629\ngini = 0.444\nsample
s = 3\nvalue = [2, 1]'),
 Text(0.2657580919931857, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.27938671209540034, 0.3055555555555556, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.28620102214650767, 0.3611111111111111, 'gini = 0.0\nsamples = 6\nvalue =
[0, 6]'),
 Text(0.30664395229982966, 0.5277777777777778, 'x[15] <= 0.167\ngini = 0.149\nsamp
les = 37\nvalue = [34, 3]'),
 Text(0.29982964224872233, 0.4722222222222222, 'x[29] <= 0.5\ngini = 0.5\nsamples
= 6\nvalue = [3, 3]'),
 Text(0.293015332197615, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue = [3,
0]'),
 Text(0.30664395229982966, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.313458262350937, 0.4722222222222222, 'gini = 0.0\nsamples = 31\nvalue = [3
1, 0]'),
 Text(0.3270868824531516, 0.5833333333333334, 'x[8] <= 0.065\ngini = 0.444\nsample
s = 6\nvalue = [2, 4]'),
 Text(0.3202725724020443, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.3339011925042589, 0.5277777777777778, 'gini = 0.0\nsamples = 4\nvalue =
[0, 4]'),
 Text(0.389267461669506, 0.6388888888888888, 'x[0] <= 0.321\ngini = 0.161\nsamples
= 192\nvalue = [175, 17]'),
 Text(0.3543441226575809, 0.5833333333333334, 'x[6] <= 0.1\ngini = 0.294\nsamples
= 67\nvalue = [55, 12]'),
 Text(0.3475298126064736, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.36115843270868825, 0.5277777777777778, 'x[29] <= 0.5\ngini = 0.26\nsamples
= 65\nvalue = [55, 10]'),
 Text(0.3441226575809199, 0.4722222222222222, 'x[11] <= 0.679\ngini = 0.469\nsampl
es = 16\nvalue = [10, 6]'),
 Text(0.3373083475298126, 0.4166666666666667, 'x[4] <= 0.018\ngini = 0.444\nsample
s = 9\nvalue = [3, 6]'),
 Text(0.33049403747870526, 0.3611111111111111, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.3441226575809199, 0.3611111111111111, 'x[1] <= 0.25\ngini = 0.245\nsamples
= 7\nvalue = [1, 6]'),
 Text(0.3373083475298126, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.35093696763202725, 0.3055555555555556, 'gini = 0.0\nsamples = 6\nvalue =
[0, 6]'),
 Text(0.35093696763202725, 0.4166666666666667, 'gini = 0.0\nsamples = 7\nvalue =
[7, 0]'),
 Text(0.3781942078364566, 0.4722222222222222, 'x[2] <= 0.037\ngini = 0.15\nsamples
= 49\nvalue = [45, 4]'),
 Text(0.37137989778534924, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.3850085178875639, 0.4166666666666667, 'x[2] <= 0.938\ngini = 0.117\nsample
s = 48\nvalue = [45, 3]'),
 Text(0.3781942078364566, 0.3611111111111111, 'x[5] <= 0.875\ngini = 0.081\nsample
s = 47\nvalue = [45, 2]'),
 Text(0.3645655877342419, 0.3055555555555556, 'x[12] <= 0.167\ngini = 0.043\nsampl
es = 45\nvalue = [44, 1]'),
 Text(0.3577512776831346, 0.25, 'x[14] <= 0.625\ngini = 0.444\nsamples = 3\nvalue
= [2, 1]'),
 Text(0.35093696763202725, 0.19444444444444445, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.3645655877342419, 0.19444444444444445, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
```

```
 Text(0.37137989778534924, 0.25, 'gini = 0.0\nsamples = 42\nvalue = [42, 0]'),
 Text(0.39182282793867124, 0.3055555555555556, 'x[27] <= 0.125\ngini = 0.5\nsample
s = 2\nvalue = [1, 1]'),
 Text(0.3850085178875639, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.3986371379897785, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.39182282793867124, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.424190800681431, 0.5833333333333334, 'x[8] <= 0.022\ngini = 0.077\nsamples
= 125\nvalue = [120, 5]'),
 Text(0.40545144804088584, 0.5277777777777778, 'x[2] <= 0.578\ngini = 0.5\nsamples
= 4\nvalue = [2, 2]'),
 Text(0.3986371379897785, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.4122657580919932, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.44293015332197616, 0.5277777777777778, 'x[18] <= 0.968\ngini = 0.048\nsamp
les = 121\nvalue = [118, 3]'),
 Text(0.42589437819420783, 0.4722222222222222, 'x[2] <= 0.98\ngini = 0.033\nsample
s = 118\nvalue = [116, 2]'),
 Text(0.4122657580919932, 0.4166666666666667, 'x[14] <= 0.938\ngini = 0.017\nsampl
es = 114\nvalue = [113, 1]'),
 Text(0.40545144804088584, 0.3611111111111111, 'gini = 0.0\nsamples = 107\nvalue =
[107, 0]'),
 Text(0.4190800681431005, 0.3611111111111111, 'x[16] <= 0.25\ngini = 0.245\nsample
s = 7\nvalue = [6, 1]'),
 Text(0.4122657580919932, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.42589437819420783, 0.3055555555555556, 'gini = 0.0\nsamples = 6\nvalue =
[6, 0]'),
 Text(0.4395229982964225, 0.4166666666666667, 'x[12] <= 0.833\ngini = 0.375\nsampl
es = 4\nvalue = [3, 1]'),
 Text(0.43270868824531517, 0.3611111111111111, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
 Text(0.4463373083475298, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.4599659284497445, 0.4722222222222222, 'x[30] <= 0.038\ngini = 0.444\nsampl
es = 3\nvalue = [2, 1]'),
 Text(0.45315161839863716, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.46678023850085176, 0.4166666666666667, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.5968909710391823, 0.6944444444444444, 'x[30] <= 0.787\ngini = 0.1\nsamples
= 493\nvalue = [467, 26]'),
 Text(0.5617546848381602, 0.6388888888888888, 'x[15] <= 0.5\ngini = 0.094\nsamples
= 486\nvalue = [462, 24]'),
 Text(0.5119250425894378, 0.5833333333333334, 'x[14] <= 0.938\ngini = 0.154\nsampl
es = 191\nvalue = [175, 16]'),
 Text(0.5051107325383305, 0.5277777777777778, 'x[18] <= 0.481\ngini = 0.145\nsampl
es = 190\nvalue = [175, 15]'),
 Text(0.48722316865417375, 0.4722222222222222, 'x[18] <= 0.47\ngini = 0.221\nsampl
es = 95\nvalue = [83, 12]'),
 Text(0.4804088586030664, 0.4166666666666667, 'x[33] <= 0.794\ngini = 0.207\nsampl
es = 94\nvalue = [83, 11]'),
 Text(0.4735945485519591, 0.3611111111111111, 'x[5] <= 0.375\ngini = 0.192\nsample
s = 93\nvalue = [83, 10]'),
 Text(0.4514480408858603, 0.3055555555555556, 'x[6] <= 0.9\ngini = 0.363\nsamples
= 21\nvalue = [16, 5]'),
 Text(0.444633730834753, 0.25, 'x[17] <= 0.413\ngini = 0.266\nsamples = 19\nvalue
= [16, 3]'),
 Text(0.43100511073253833, 0.19444444444444445, 'x[19] <= 0.056\ngini = 0.117\nsam
ples = 16\nvalue = [15, 1]'),
 Text(0.424190800681431, 0.1388888888888889, 'x[6] <= 0.4\ngini = 0.5\nsamples = 2
\nvalue = [1, 1]'),
 Text(0.41737649063032367, 0.08333333333333333, 'gini = 0.0\nsamples = 1\nvalue =
```

```
[1, 0]'),
 Text(0.43100511073253833, 0.08333333333333333, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.43781942078364566, 0.1388888888888889, 'gini = 0.0\nsamples = 14\nvalue =
[14, 0]'),
 Text(0.45826235093696766, 0.19444444444444445, 'x[26] <= 0.667\ngini = 0.444\nsam
ples = 3\nvalue = [1, 2]'),
 Text(0.4514480408858603, 0.1388888888888889, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.46507666098807493, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.45826235093696766, 0.25, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.4957410562180579, 0.3055555555555556, 'x[31] <= 0.139\ngini = 0.129\nsampl
es = 72\nvalue = [67, 5]'),
 Text(0.4787052810902896, 0.25, 'x[8] <= 0.68\ngini = 0.444\nsamples = 6\nvalue =
[4, 2]'),
 Text(0.47189097103918226, 0.19444444444444445, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0]'),
 Text(0.4855195911413969, 0.19444444444444445, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.5127768313458262, 0.25, 'x[11] <= 0.993\ngini = 0.087\nsamples = 66\nvalue
= [63, 3]'),
 Text(0.4991482112436116, 0.19444444444444445, 'x[28] <= 0.583\ngini = 0.061\nsamp
les = 64\nvalue = [62, 2]'),
 Text(0.49233390119250425, 0.1388888888888889, 'gini = 0.0\nsamples = 51\nvalue =
[51, 0]'),
 Text(0.5059625212947189, 0.1388888888888889, 'x[3] <= 0.75\ngini = 0.26\nsamples
= 13\nvalue = [11, 2]'),
 Text(0.4991482112436116, 0.08333333333333333, 'gini = 0.0\nsamples = 9\nvalue =
[9, 0]'),
 Text(0.5127768313458262, 0.08333333333333333, 'x[27] <= 0.3\ngini = 0.5\nsamples
= 4\nvalue = [2, 2]'),
 Text(0.5059625212947189, 0.027777777777777776, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.5195911413969335, 0.027777777777777776, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.5264054514480409, 0.19444444444444445, 'x[9] <= 0.333\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.5195911413969335, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.5332197614991482, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.48722316865417375, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.4940374787052811, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.5229982964224872, 0.4722222222222222, 'x[19] <= 0.5\ngini = 0.061\nsamples
= 95\nvalue = [92, 3]'),
 Text(0.5161839863713799, 0.4166666666666667, 'gini = 0.0\nsamples = 76\nvalue =
[76, 0]'),
 Text(0.5298126064735945, 0.4166666666666667, 'x[8] <= 0.161\ngini = 0.266\nsample
s = 19\nvalue = [16, 3]'),
 Text(0.5161839863713799, 0.3611111111111111, 'x[22] <= 0.143\ngini = 0.444\nsampl
es = 3\nvalue = [1, 2]'),
 Text(0.5093696763202725, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.5229982964224872, 0.3055555555555556, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.5434412265758092, 0.3611111111111111, 'x[33] <= 0.059\ngini = 0.117\nsampl
es = 16\nvalue = [15, 1]'),
 Text(0.5366269165247018, 0.3055555555555556, 'x[30] <= 0.262\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.5298126064735945, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5434412265758092, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
```

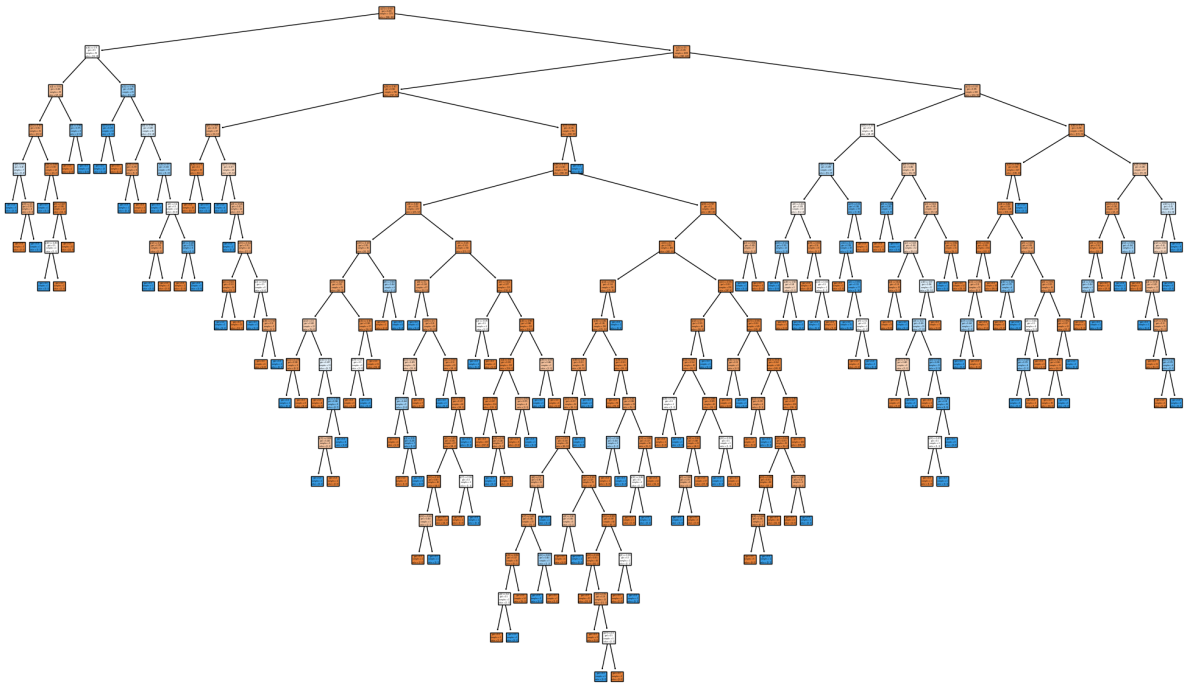
```
 Text(0.5502555366269165, 0.3055555555555556, 'gini = 0.0\nsamples = 14\nvalue =
[14, 0]'),
 Text(0.5187393526405452, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.6115843270868825, 0.5833333333333334, 'x[22] <= 0.036\ngini = 0.053\nsampl
es = 295\nvalue = [287, 8]'),
 Text(0.5877342419080068, 0.5277777777777778, 'x[32] <= 0.7\ngini = 0.159\nsamples
= 46\nvalue = [42, 4]'),
 Text(0.5809199318568995, 0.4722222222222222, 'x[11] <= 0.071\ngini = 0.124\nsampl
es = 45\nvalue = [42, 3]'),
 Text(0.5638841567291312, 0.4166666666666667, 'x[18] <= 0.702\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.5570698466780238, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.5706984667802385, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.5979557069846678, 0.4166666666666667, 'x[27] <= 0.688\ngini = 0.089\nsampl
es = 43\nvalue = [41, 2]'),
 Text(0.5843270868824532, 0.3611111111111111, 'x[14] <= 0.062\ngini = 0.048\nsampl
es = 41\nvalue = [40, 1]'),
 Text(0.5775127768313458, 0.3055555555555556, 'x[9] <= 0.167\ngini = 0.375\nsample
s = 4\nvalue = [3, 1]'),
 Text(0.5706984667802385, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5843270868824532, 0.25, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.5911413969335605, 0.3055555555555556, 'gini = 0.0\nsamples = 37\nvalue =
[37, 0]'),
 Text(0.6115843270868825, 0.3611111111111111, 'x[30] <= 0.212\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.6047700170357752, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.6183986371379898, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.5945485519591142, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.6354344122657581, 0.5277777777777778, 'x[17] <= 0.056\ngini = 0.032\nsampl
es = 249\nvalue = [245, 4]'),
 Text(0.6183986371379898, 0.4722222222222222, 'x[16] <= 0.75\ngini = 0.32\nsamples
= 5\nvalue = [4, 1]'),
 Text(0.6115843270868825, 0.4166666666666667, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0]'),
 Text(0.6252129471890971, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.6524701873935264, 0.4722222222222222, 'x[2] <= 0.015\ngini = 0.024\nsample
s = 244\nvalue = [241, 3]'),
 Text(0.6388415672913118, 0.4166666666666667, 'x[22] <= 0.714\ngini = 0.278\nsampl
es = 6\nvalue = [5, 1]'),
 Text(0.6320272572402045, 0.3611111111111111, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]'),
 Text(0.645655877342419, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.666098807495741, 0.4166666666666667, 'x[24] <= 0.167\ngini = 0.017\nsample
s = 238\nvalue = [236, 2]'),
 Text(0.6592844974446337, 0.3611111111111111, 'x[29] <= 0.833\ngini = 0.073\nsampl
es = 53\nvalue = [51, 2]'),
 Text(0.645655877342419, 0.3055555555555556, 'x[33] <= 0.088\ngini = 0.041\nsample
s = 48\nvalue = [47, 1]'),
 Text(0.6388415672913118, 0.25, 'x[18] <= 0.824\ngini = 0.245\nsamples = 7\nvalue
= [6, 1]'),
 Text(0.6320272572402045, 0.19444444444444445, 'gini = 0.0\nsamples = 6\nvalue =
[6, 0]'),
 Text(0.645655877342419, 0.19444444444444445, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.6524701873935264, 0.25, 'gini = 0.0\nsamples = 41\nvalue = [41, 0]'),
 Text(0.6729131175468483, 0.3055555555555556, 'x[31] <= 0.417\ngini = 0.32\nsample
```

```
s = 5\nvalue = [4, 1]'),
 Text(0.666098807495741, 0.25, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.6797274275979557, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6729131175468483, 0.3611111111111111, 'gini = 0.0\nsamples = 185\nvalue =
[185, 0]'),
 Text(0.6320272572402045, 0.6388888888888888, 'x[2] <= 0.366\ngini = 0.408\nsample
s = 7\nvalue = [5, 2]'),
 Text(0.6252129471890971, 0.5833333333333334, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.6388415672913118, 0.5833333333333334, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]'),
 Text(0.4856260647359455, 0.75, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8202725724020443, 0.8611111111111112, 'x[17] <= 0.157\ngini = 0.385\nsampl
es = 300\nvalue = [222, 78]'),
 Text(0.731686541737649, 0.8055555555555556, 'x[26] <= 0.167\ngini = 0.5\nsamples
= 96\nvalue = [49, 47]'),
 Text(0.696763202725724, 0.75, 'x[4] <= 0.161\ngini = 0.459\nsamples = 42\nvalue =
[15, 27]'),
 Text(0.6729131175468483, 0.6944444444444444, 'x[8] <= 0.415\ngini = 0.499\nsample
s = 23\nvalue = [12, 11]'),
 Text(0.6592844974446337, 0.6388888888888888, 'x[18] <= 0.561\ngini = 0.355\nsampl
es = 13\nvalue = [3, 10]'),
 Text(0.6524701873935264, 0.5833333333333334, 'gini = 0.0\nsamples = 8\nvalue =
[0, 8]'),
 Text(0.666098807495741, 0.5833333333333334, 'x[28] <= 0.583\ngini = 0.48\nsamples
= 5\nvalue = [3, 2]'),
 Text(0.6592844974446337, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
 Text(0.6729131175468483, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.686541737649063, 0.6388888888888888, 'x[29] <= 0.833\ngini = 0.18\nsamples
= 10\nvalue = [9, 1]'),
 Text(0.6797274275979557, 0.5833333333333334, 'gini = 0.0\nsamples = 8\nvalue =
[8, 0]'),
 Text(0.6933560477001703, 0.5833333333333334, 'x[24] <= 0.333\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.686541737649063, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.7001703577512777, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.7206132879045997, 0.6944444444444444, 'x[27] <= 0.35\ngini = 0.266\nsample
s = 19\nvalue = [3, 16]'),
 Text(0.7137989778534923, 0.6388888888888888, 'x[11] <= 0.2\ngini = 0.198\nsamples
= 18\nvalue = [2, 16]'),
 Text(0.706984667802385, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.7206132879045997, 0.5833333333333334, 'x[0] <= 0.595\ngini = 0.111\nsample
s = 17\nvalue = [1, 16]'),
 Text(0.7137989778534923, 0.5277777777777778, 'gini = 0.0\nsamples = 15\nvalue =
[0, 15]'),
 Text(0.727427597955707, 0.5277777777777778, 'x[9] <= 0.5\ngini = 0.5\nsamples = 2
\nvalue = [1, 1]'),
 Text(0.7206132879045997, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.7342419080068143, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.727427597955707, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.7666098807495741, 0.75, 'x[0] <= 0.202\ngini = 0.466\nsamples = 54\nvalue
= [34, 20]'),
 Text(0.747870528109029, 0.6944444444444444, 'x[8] <= 0.164\ngini = 0.245\nsamples
= 7\nvalue = [1, 6]'),
 Text(0.7410562180579217, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
```

```
 Text(0.7546848381601363, 0.6388888888888888, 'gini = 0.0\nsamples = 6\nvalue =
[0, 6]'),
 Text(0.7853492333901193, 0.6944444444444444, 'x[2] <= 0.622\ngini = 0.418\nsample
s = 47\nvalue = [33, 14]'),
 Text(0.768313458262351, 0.6388888888888888, 'x[2] <= 0.145\ngini = 0.482\nsamples
= 32\nvalue = [19, 13]'),
 Text(0.7546848381601363, 0.5833333333333334, 'x[4] <= 0.821\ngini = 0.18\nsamples
= 10\nvalue = [9, 1]'),
 Text(0.747870528109029, 0.5277777777777778, 'gini = 0.0\nsamples = 9\nvalue = [9,
0]'),
 Text(0.7614991482112436, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.7819420783645656, 0.5833333333333334, 'x[18] <= 0.87\ngini = 0.496\nsample
s = 22\nvalue = [10, 12]'),
 Text(0.7751277683134583, 0.5277777777777778, 'x[8] <= 0.41\ngini = 0.465\nsamples
= 19\nvalue = [7, 12]'),
 Text(0.7614991482112436, 0.4722222222222222, 'x[18] <= 0.715\ngini = 0.469\nsampl
es = 8\nvalue = [5, 3]'),
 Text(0.7546848381601363, 0.4166666666666667, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]'),
 Text(0.768313458262351, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue = [0,
3]'),
 Text(0.7887563884156729, 0.4722222222222222, 'x[0] <= 0.25\ngini = 0.298\nsamples
= 11\nvalue = [2, 9]'),
 Text(0.7819420783645656, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.7955706984667802, 0.4166666666666667, 'x[4] <= 0.018\ngini = 0.18\nsamples
= 10\nvalue = [1, 9]'),
 Text(0.7887563884156729, 0.3611111111111111, 'x[28] <= 0.417\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.7819420783645656, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.7955706984667802, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.8023850085178875, 0.3611111111111111, 'gini = 0.0\nsamples = 8\nvalue =
[0, 8]'),
 Text(0.7887563884156729, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
 Text(0.8023850085178875, 0.6388888888888888, 'x[11] <= 0.064\ngini = 0.124\nsampl
es = 15\nvalue = [14, 1]'),
 Text(0.7955706984667802, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.8091993185689949, 0.5833333333333334, 'gini = 0.0\nsamples = 14\nvalue =
[14, 0]'),
 Text(0.9088586030664395, 0.8055555555555556, 'x[16] <= 0.75\ngini = 0.258\nsample
s = 204\nvalue = [173, 31]'),
 Text(0.8551959114139693, 0.75, 'x[17] <= 0.992\ngini = 0.138\nsamples = 147\nvalu
e = [136, 11]'),
 Text(0.848381601362862, 0.6944444444444444, 'x[4] <= 0.482\ngini = 0.128\nsamples
= 146\nvalue = [136, 10]'),
 Text(0.8296422487223168, 0.6388888888888888, 'x[30] <= 0.063\ngini = 0.038\nsampl
es = 104\nvalue = [102, 2]'),
 Text(0.8228279386712095, 0.5833333333333334, 'x[11] <= 0.193\ngini = 0.32\nsample
s = 10\nvalue = [8, 2]'),
 Text(0.8160136286201022, 0.5277777777777778, 'x[27] <= 0.475\ngini = 0.444\nsampl
es = 3\nvalue = [1, 2]'),
 Text(0.8091993185689949, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.8228279386712095, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.8296422487223168, 0.5277777777777778, 'gini = 0.0\nsamples = 7\nvalue =
[7, 0]'),
 Text(0.8364565587734242, 0.5833333333333334, 'gini = 0.0\nsamples = 94\nvalue =
[94, 0]'),
```

```
 Text(0.8671209540034072, 0.6388888888888888, 'x[9] <= 0.167\ngini = 0.308\nsample
s = 42\nvalue = [34, 8]'),
 Text(0.8500851788756388, 0.5833333333333334, 'x[18] <= 0.194\ngini = 0.375\nsampl
es = 4\nvalue = [1, 3]'),
 Text(0.8432708688245315, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.8568994889267462, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.8841567291311755, 0.5833333333333334, 'x[0] <= 0.393\ngini = 0.229\nsample
s = 38\nvalue = [33, 5]'),
 Text(0.8705281090289608, 0.5277777777777778, 'x[4] <= 0.821\ngini = 0.5\nsamples
= 6\nvalue = [3, 3]'),
 Text(0.8637137989778535, 0.4722222222222222, 'x[8] <= 0.549\ngini = 0.375\nsample
s = 4\nvalue = [1, 3]'),
 Text(0.8568994889267462, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.8705281090289608, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.8773424190800682, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.8977853492333902, 0.5277777777777778, 'x[8] <= 0.992\ngini = 0.117\nsample
s = 32\nvalue = [30, 2]'),
 Text(0.8909710391822828, 0.4722222222222222, 'x[28] <= 0.917\ngini = 0.062\nsampl
es = 31\nvalue = [30, 1]'),
 Text(0.8841567291311755, 0.4166666666666667, 'gini = 0.0\nsamples = 30\nvalue =
[30, 0]'),
 Text(0.8977853492333902, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.9045996592844975, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.8620102214650767, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.9625212947189097, 0.75, 'x[14] <= 0.812\ngini = 0.456\nsamples = 57\nvalue
= [37, 20]'),
 Text(0.938671209540034, 0.6944444444444444, 'x[32] <= 0.4\ngini = 0.238\nsamples
= 29\nvalue = [25, 4]'),
 Text(0.9250425894378195, 0.6388888888888888, 'x[8] <= 0.071\ngini = 0.142\nsample
s = 26\nvalue = [24, 2]'),
 Text(0.9182282793867121, 0.5833333333333334, 'x[2] <= 0.206\ngini = 0.444\nsample
s = 3\nvalue = [1, 2]'),
 Text(0.9114139693356048, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.9250425894378195, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.9318568994889267, 0.5833333333333334, 'gini = 0.0\nsamples = 23\nvalue =
[23, 0]'),
 Text(0.9522998296422487, 0.6388888888888888, 'x[32] <= 0.933\ngini = 0.444\nsampl
es = 3\nvalue = [1, 2]'),
 Text(0.9454855195911414, 0.5833333333333334, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.959114139693356, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.9863713798977853, 0.6944444444444444, 'x[32] <= 0.1\ngini = 0.49\nsamples
= 28\nvalue = [12, 16]'),
 Text(0.979557069846678, 0.6388888888888888, 'x[12] <= 0.833\ngini = 0.48\nsamples
= 20\nvalue = [12, 8]'),
 Text(0.9727427597955707, 0.5833333333333334, 'x[4] <= 0.018\ngini = 0.415\nsample
s = 17\nvalue = [12, 5]'),
 Text(0.9659284497444633, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.979557069846678, 0.5277777777777778, 'x[17] <= 0.365\ngini = 0.32\nsamples
= 15\nvalue = [12, 3]'),
 Text(0.9727427597955707, 0.4722222222222222, 'gini = 0.0\nsamples = 11\nvalue =
[11, 0]'),
```

```
 Text(0.9863713798977853, 0.4722222222222222, 'x[4] <= 0.179\ngini = 0.375\nsample
s = 4\nvalue = [1, 3]'),
 Text(0.979557069846678, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.9931856899488927, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.9863713798977853, 0.5833333333333334, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.9931856899488927, 0.6388888888888888, 'gini = 0.0\nsamples = 8\nvalue =
[0, 8]')]
```



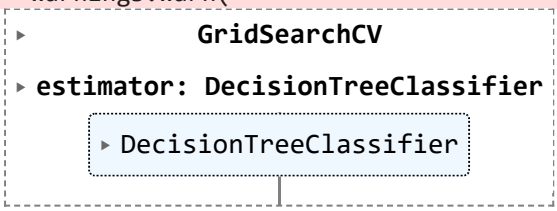
In [57]:
```
from sklearn.model_selection import GridSearchCV
parameter={
 'criterion':['gini','entropy'],
  'splitter':['best','random'],
  'max_depth':[1,2,3,4,5],
  'max_features':['auto', 'sqrt', 'log2']

}
```

In [58]:
```
grid_search=GridSearchCV(estimator=dtc,param_grid=parameter,cv=5,scoring="accuracy"
```

In [59]:
```
grid_search.fit(x_train,y_train)
```

```
C:\Users\saisa\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.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 error_sco
re='raise'.

Below are more details about the failures:
--------------------------------------------------------------------------------
100 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\saisa\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\saisa\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in
wrapper
    estimator._validate_params()
  File "C:\Users\saisa\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in
_validate_params
    validate_parameter_constraints(
  File "C:\Users\saisa\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 {'sqrt', 'log2'} or None. Got 'auto' instead.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\Users\saisa\anaconda3\Lib\site-packages\sklearn\model_selection\_search.py:976:
UserWarning: One or more of the test scores are non-finite: [       nan        nan
0.84013704 0.84013704 0.84013704 0.84013704
        nan        nan 0.8341832  0.84013704 0.82993148 0.84354129
        nan        nan 0.8384313  0.83843491 0.83672196 0.84013704
        nan        nan 0.82824739 0.83589614 0.84353047 0.83589254
        nan        nan 0.84694915 0.83843491 0.82399567 0.83845655
        nan        nan 0.84013704 0.84013704 0.84013704 0.84013704
        nan        nan 0.83673278 0.83844212 0.83334295 0.84013704
        nan        nan 0.84182113 0.83928597 0.84014064 0.83843491
        nan        nan 0.82653805 0.84184277 0.8350595  0.84269383
        nan        nan 0.83589614 0.84695636 0.83247386 0.84438154]
  warnings.warn(
```

Out[59]:



In [60]:
```
grid_search.best_params_
```

Out[60]:
```
{'criterion': 'entropy',
 'max_depth': 5,
 'max_features': 'sqrt',
 'splitter': 'random'}
```

In [61]:
```
dtc_cv=DecisionTreeClassifier(criterion= 'entropy',
 max_depth=3,
 max_features='sqrt',
 splitter='best')
dtc_cv.fit(x_train,y_train)
```

Out[61]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=3, max_features='sq
rt')
```

In [62]:
```
pred=dtc_cv.predict(x_test)
```

In [63]:
```
print(classification_report(y_test,pred))
```

```
              precision    recall  f1-score   support

          No       0.83      0.98      0.90       245
         Yes       0.14      0.02      0.04        49

    accuracy                           0.82       294
   macro avg       0.49      0.50      0.47       294
weighted avg       0.72      0.82      0.75       294
```

In [64]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
```

In [65]:
```
forest_params = [{'max_depth': list(range(10, 15)), 'max_features': list(range(0,14
```

In [66]:
```
rfc_cv= GridSearchCV(rfc,param_grid=forest_params,cv=10,scoring="accuracy")
```

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

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

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

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

In [ ]:
```

```