Data Preprocessing

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

In [2]:
```
data = pd.read_csv('c:\\Users\\Vennela Baratam\\Downloads\\Employee  Attrition.csv
data
```

Out[2]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Ec
0	41	Yes	Travel_Rarely	1102	Sales	1	2	
1	49	No	Travel_Frequently	279	Research & Development	8	1	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	
4	27	No	Travel_Rarely	591	Research & Development	2	1	
...	...	...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research & Development	23	2	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	
1468	49	No	Travel_Frequently	1023	Sales	2	3	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	

1470 rows × 35 columns

Data Preprocessing

In [3]:
```
data
```

Out[3]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Ec
0	41	Yes	Travel_Rarely	1102	Sales	1	2	
1	49	No	Travel_Frequently	279	Research & Development	8	1	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	
4	27	No	Travel_Rarely	591	Research & Development	2	1	
...	...	...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research & Development	23	2	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	
1468	49	No	Travel_Frequently	1023	Sales	2	3	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	

1470 rows × 35 columns

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

Out[4]:

	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 [5]:
```
data.tail()
```

Out[5]:

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

5 rows × 35 columns

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

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

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

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

In [8]:
```
data.describe()
```

Out[8]:

	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 [ ]:

Check null values

In [9]:
```
data.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]:
```
data.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]:
```
data["Education"].value_counts()
```

Out[11]:
```
3    572
4    398
2    282
1    170
5     48
Name: Education, dtype: int64
```

In [12]:
```
data["Department"].value_counts()
```

Out[12]:
```
Research & Development    961
Sales                     446
Human Resources            63
Name: Department, dtype: int64
```

Data Visulization

In [13]:
```
corr = data.corr()
corr
```

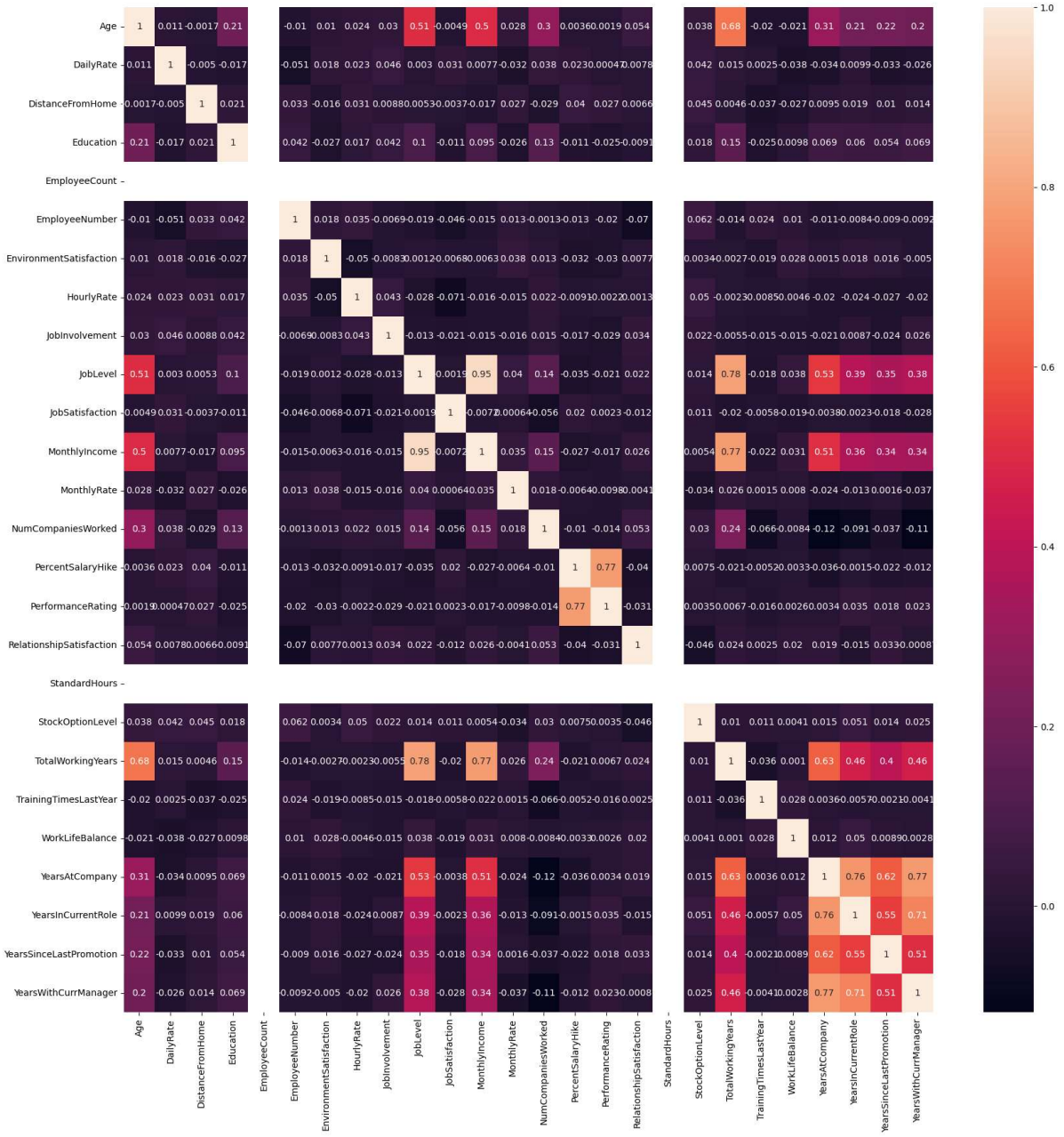
Out[13]:

	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 [14]:
```
plt.subplots(figsize=(20,20))
sns.heatmap(corr,annot = True)
```

Out[14]:
```
<AxesSubplot:>
```



In [15]:
```
corr.isnull()
```

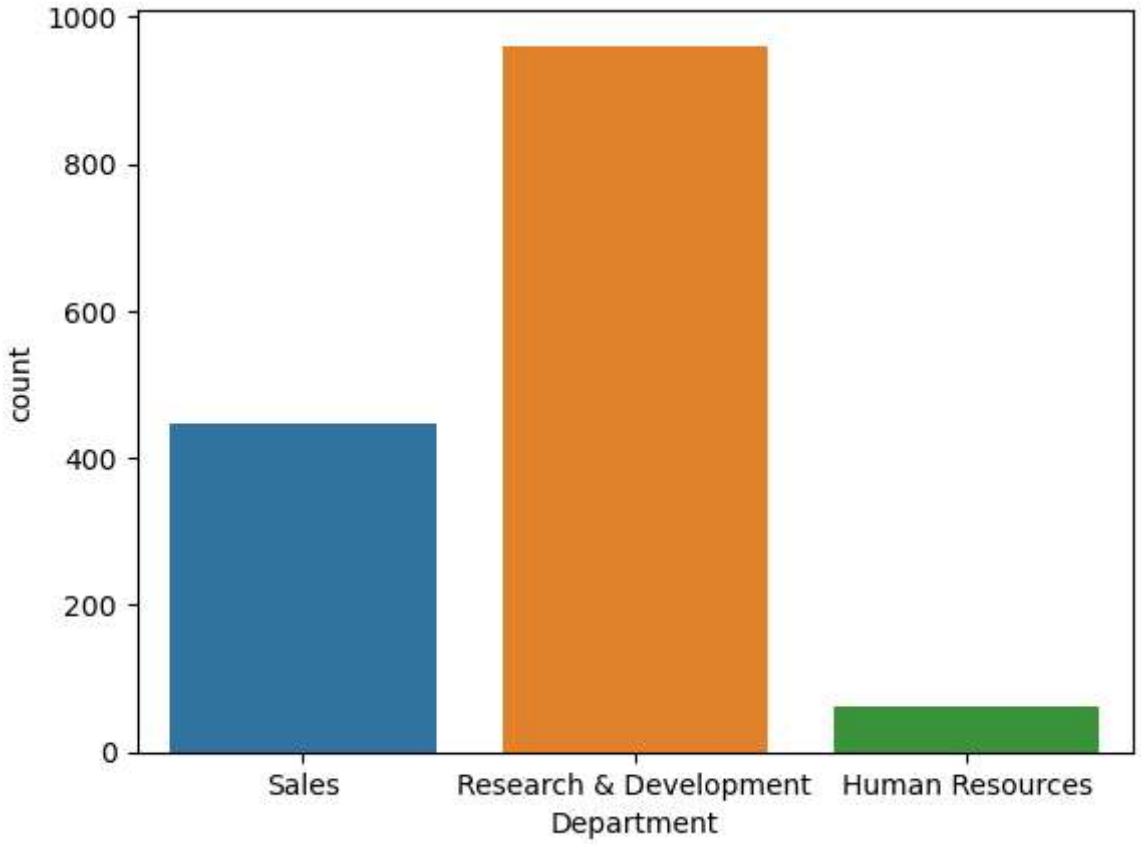
Out[15]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	Emplo
Age	False	False	False	False	True	
DailyRate	False	False	False	False	True	
DistanceFromHome	False	False	False	False	True	
Education	False	False	False	False	True	
EmployeeCount	True	True	True	True	True	
EmployeeNumber	False	False	False	False	True	
EnvironmentSatisfaction	False	False	False	False	True	
HourlyRate	False	False	False	False	True	
JobInvolvement	False	False	False	False	True	
JobLevel	False	False	False	False	True	
JobSatisfaction	False	False	False	False	True	
MonthlyIncome	False	False	False	False	True	
MonthlyRate	False	False	False	False	True	
NumCompaniesWorked	False	False	False	False	True	
PercentSalaryHike	False	False	False	False	True	
PerformanceRating	False	False	False	False	True	
RelationshipSatisfaction	False	False	False	False	True	
StandardHours	True	True	True	True	True	
StockOptionLevel	False	False	False	False	True	
TotalWorkingYears	False	False	False	False	True	
TrainingTimesLastYear	False	False	False	False	True	
WorkLifeBalance	False	False	False	False	True	
YearsAtCompany	False	False	False	False	True	
YearsInCurrentRole	False	False	False	False	True	
YearsSinceLastPromotion	False	False	False	False	True	
YearsWithCurrManager	False	False	False	False	True	

26 rows × 26 columns

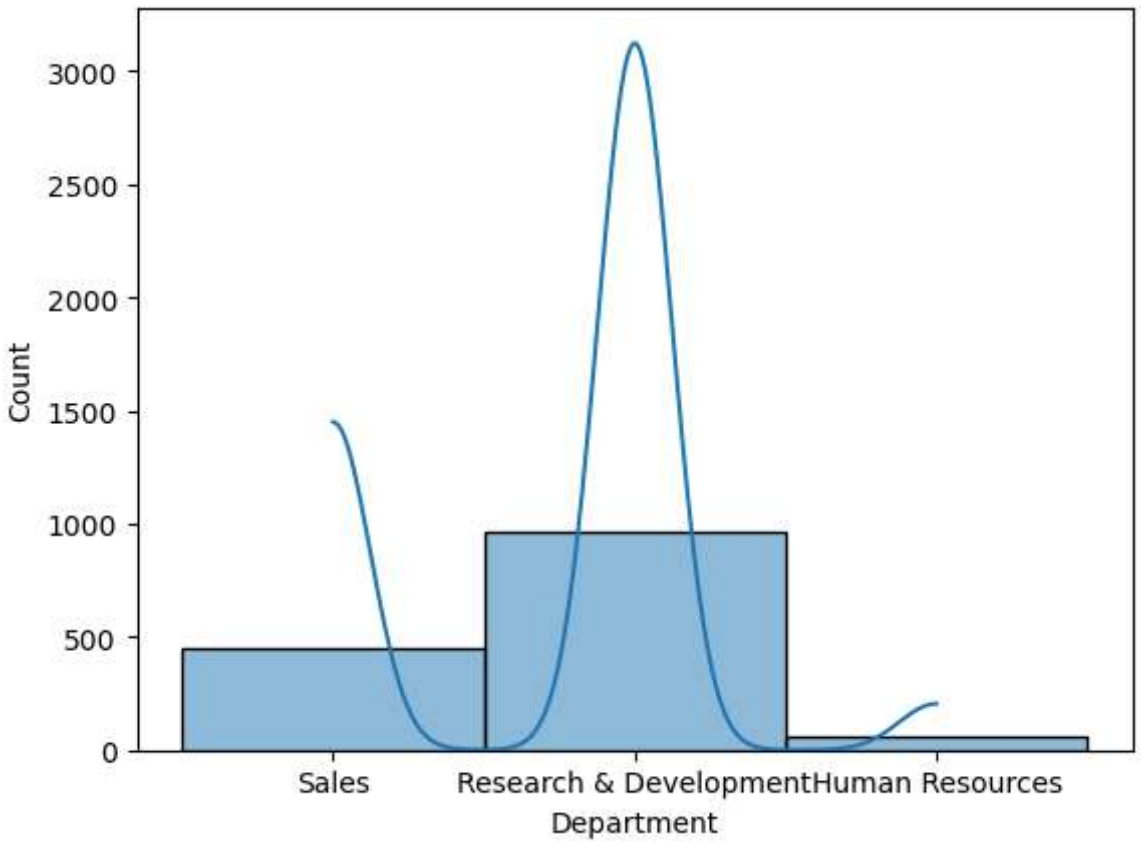
In [16]:
```
sns.countplot(x="Department",data=data)
```

Out[16]:
```
<AxesSubplot:xlabel='Department', ylabel='count'>
```



In [17]:
```
sns.histplot(x="Department",data= data,kde=True)
```

Out[17]:
```
<AxesSubplot:xlabel='Department', ylabel='Count'>
```



Outlier Detection

In [18]:
```
data.head()
```

Out[18]:

	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 [19]:
```
data.tail()
```

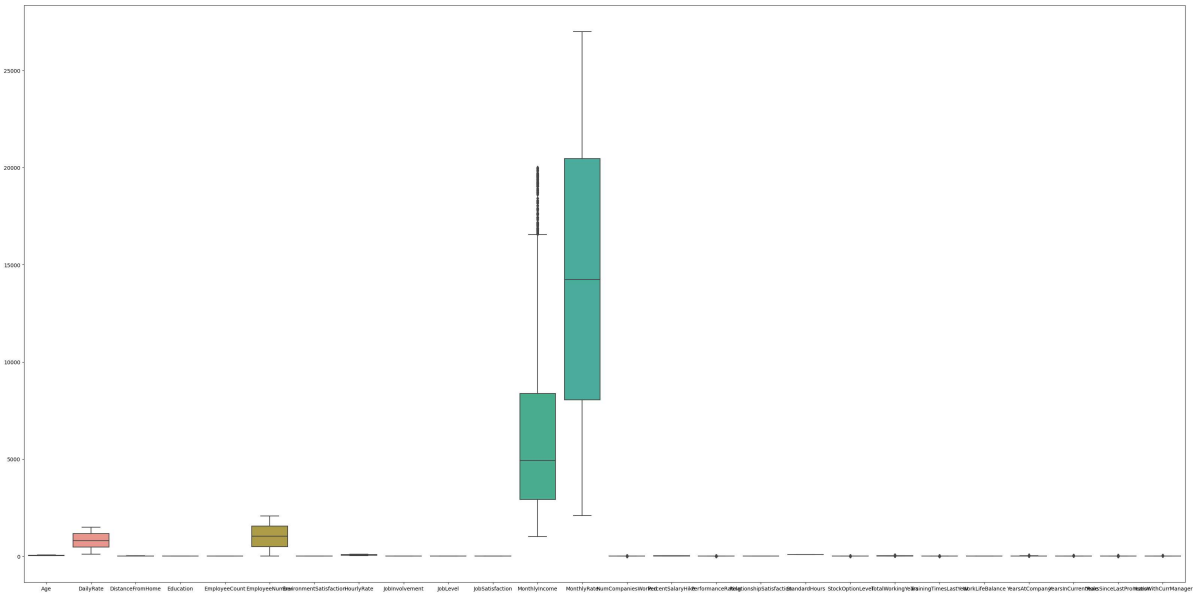
Out[19]:

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

5 rows × 35 columns

In [20]:
```
plt.figure(figsize=(40,20))
sns.boxplot(data=data)
```

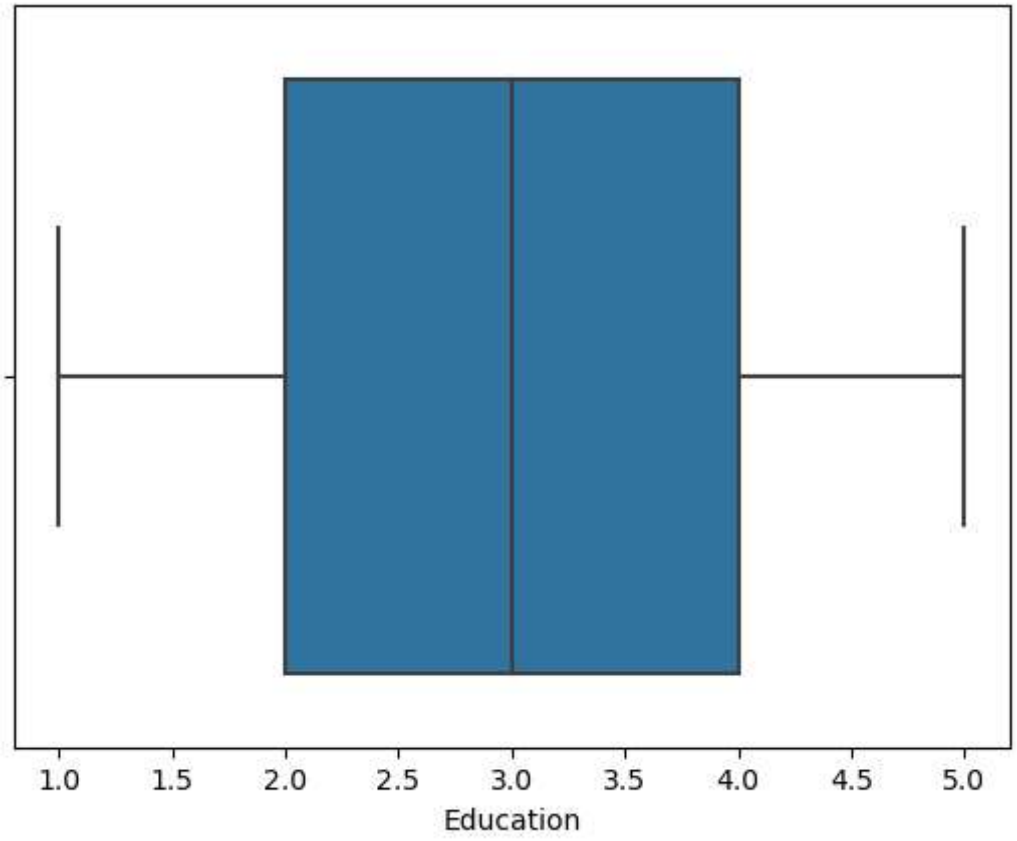
Out[20]:
```
<AxesSubplot:>
```



In [21]:
```
sns.boxplot(data["Education"])
```

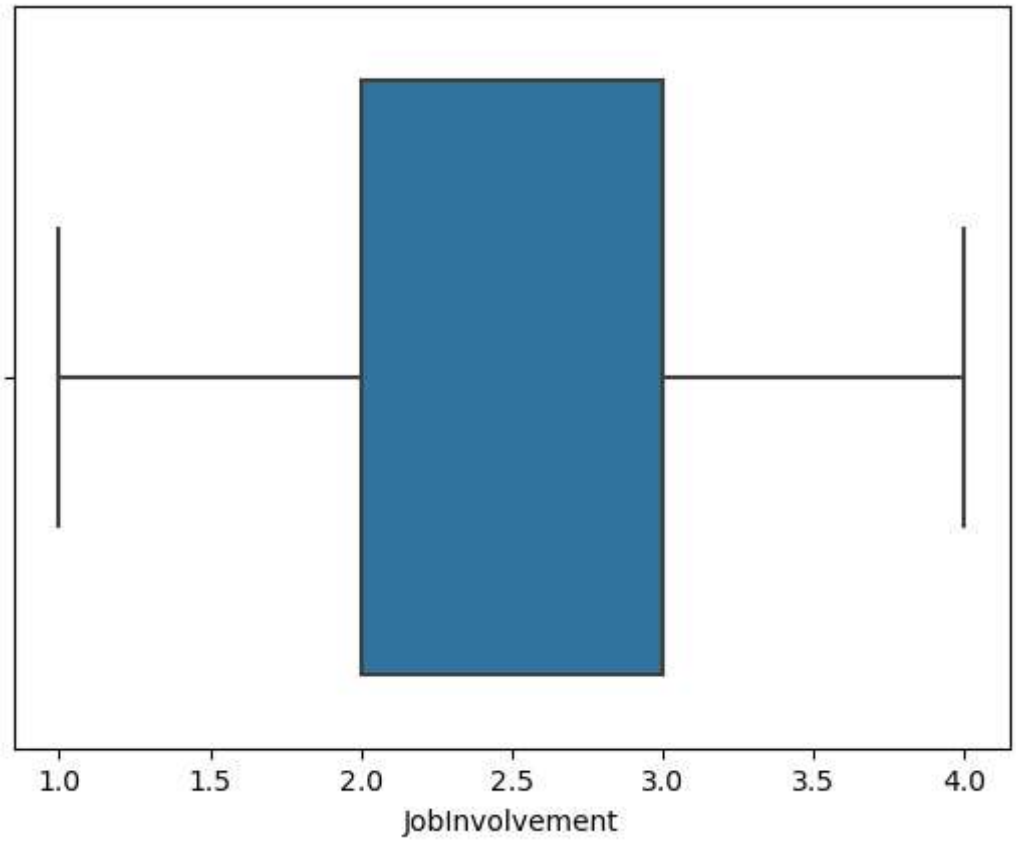
```
C:\Users\Vennela Baratam\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variable as a keyword arg: x. From version 0.12, t
he only valid positional argument will be `data`, and passing other arguments with
out an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[21]:
```
<AxesSubplot:xlabel='Education'>
```



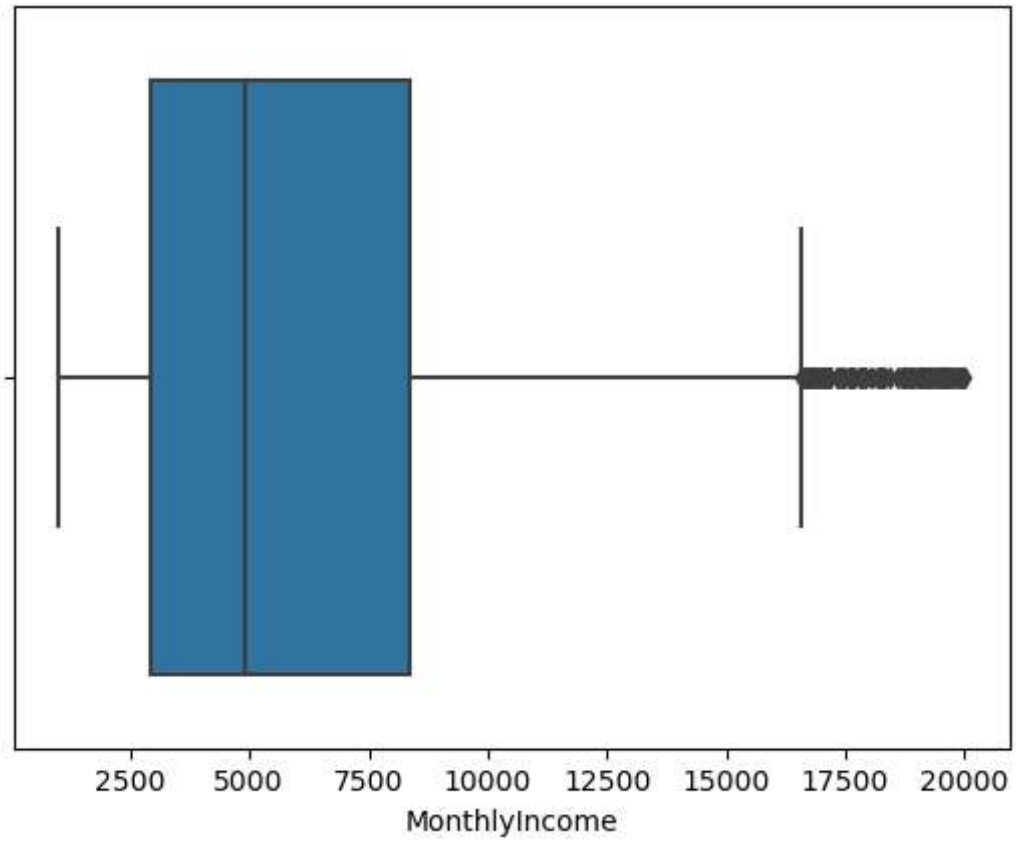
In [22]:
```
sns.boxplot(data=data,x ="JobInvolvement")
```

Out[22]:
```
<AxesSubplot:xlabel='JobInvolvement'>
```



In [23]:
```
sns.boxplot(data=data,x="MonthlyIncome")
```

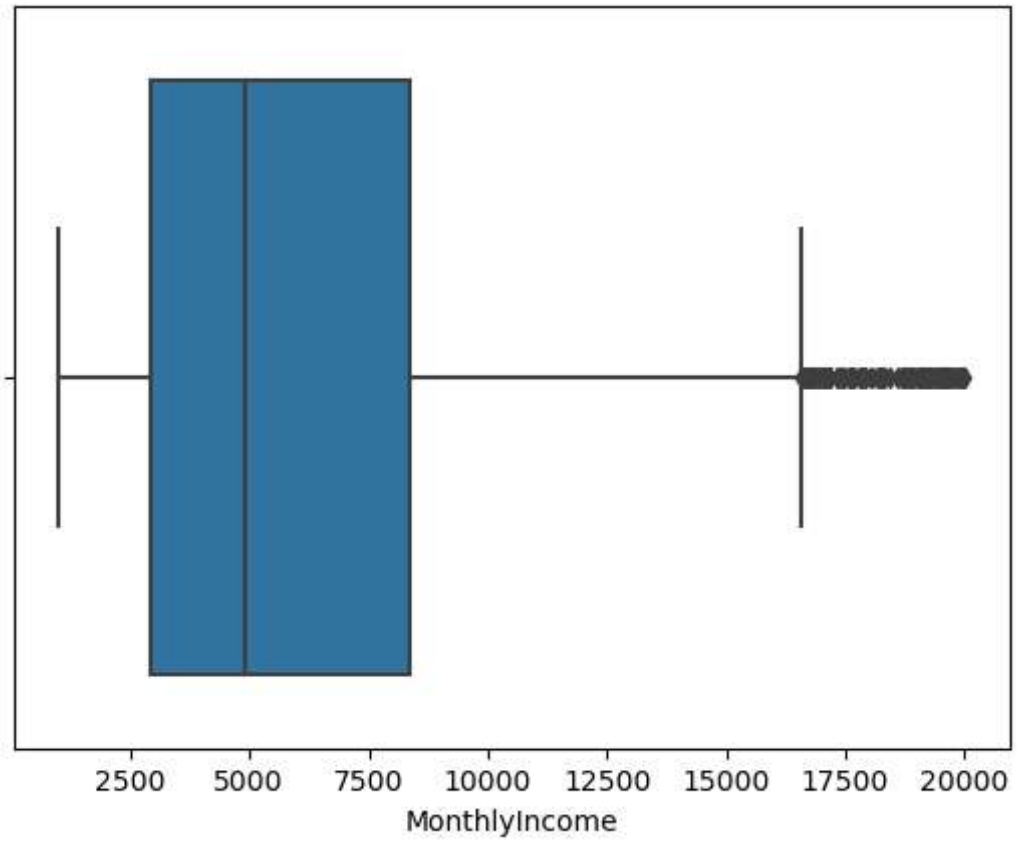
Out[23]:
```
<AxesSubplot:xlabel='MonthlyIncome'>
```



In [24]:
```
from scipy import stats
z_scores = stats.zscore(data['MonthlyIncome'])
data_cleaned = data[(np.abs(z_scores)<=3)]
```

In [25]:
```
sns.boxplot(data=data_cleaned,x="MonthlyIncome")
```

Out[25]:
```
<AxesSubplot:xlabel='MonthlyIncome'>
```



Outliers are large in quantity and inside in thresholds , so let us not remove the outliers

Splitting Dependent an Independent variables

In [26]:
```
x = data.drop(columns=["Attrition"],axis=1)
```

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

Out[27]:

	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 [28]:
```
type(x)
```

Out[28]:
```
pandas.core.frame.DataFrame
```

In [29]:
```
y = data["Attrition"]
```

In [30]:
```
y.head()
```

Out[30]:
```
0    Yes
1     No
2    Yes
3     No
4     No
Name: Attrition, dtype: object
```

In [31]:
```
type(y)
```

Out[31]:
```
pandas.core.series.Series
```

Encoding

In [32]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

In [33]:
```
categorical_features = x.select_dtypes(include=['object']).columns.tolist()
x_encoded = pd.get_dummies(x, columns = categorical_features,drop_first = True)
```

In [34]:
```
x_encoded.head()
```

Out[34]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	Environmer
0	41	1102	1	2	1	1	
1	49	279	8	1	1	2	
2	37	1373	2	2	1	4	
3	33	1392	3	4	1	5	
4	27	591	2	1	1	7	

5 rows × 47 columns

Feature Scaling

In [35]:
```
from sklearn.preprocessing import StandardScaler
s = StandardScaler()
x_scaled = pd.DataFrame(s.fit_transform(x_encoded),columns=x_encoded.columns)
```

In [36]:
```
x_scaled.head()
```

Out[36]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	Enviro
0	0.446350	0.742527	-1.010909	-0.891688	0.0	-1.701283	
1	1.322365	-1.297775	-0.147150	-1.868426	0.0	-1.699621	
2	0.008343	1.414363	-0.887515	-0.891688	0.0	-1.696298	
3	-0.429664	1.461466	-0.764121	1.061787	0.0	-1.694636	
4	-1.086676	-0.524295	-0.887515	-1.868426	0.0	-1.691313	

5 rows × 47 columns

Splitting Data into Train and Test

In [37]:
```
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,ra
```

In [38]:
```
x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

Out[38]:
```
((1176, 47), (294, 47), (1176,), (294,))
```

Logistic Regression

In [39]:
```
from sklearn.linear_model import LogisticRegression
```

In [40]:
```
lr=LogisticRegression()
```

Training and testing the model

In [41]:
```
lr.fit(x_train,y_train)
```

Out[41]:
```
LogisticRegression()
```

In [42]:
```
y_pred=lr.predict(x_test)
y_pred
```

Out[42]:
```
array(['No', 'No', 'Yes', 'No', 'Yes', 'No', 'No', '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', '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', 'Yes', 'No',
       'No', 'No', 'Yes', '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', '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', 'No', 'Yes', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'Yes',
       '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', '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', 'No', 'No', 'No', 'Yes', 'Yes', 'No', 'No', 'No', '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', 'No', 'Yes', 'No', 'No'], dtype=object)
```

In [43]:
```
y_test
```

Out[43]:
```
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 [44]:
```
data.head()
```

Out[44]:

	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

Evalution of model

In [45]:
```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,r
```

In [46]:
```
accuracy_score(y_test,y_pred)
```

Out[46]:
```
0.8775510204081632
```

In [47]:
```
confusion_matrix(y_test,y_pred)
```

Out[47]:
```
array([[238,   7],
       [ 29,  20]], dtype=int64)
```

In [48]:
```
pd.crosstab(y_test,y_pred)
```

Out[48]:

col_0	No	Yes
Attrition		
No	238	7
Yes	29	20

In [49]:
```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

          No       0.89      0.97      0.93       245
         Yes       0.74      0.41      0.53        49

    accuracy                           0.88       294
   macro avg       0.82      0.69      0.73       294
weighted avg       0.87      0.88      0.86       294
```

In [50]:
```
probability=lr.predict_proba(x_test)[:,1]
probability
```

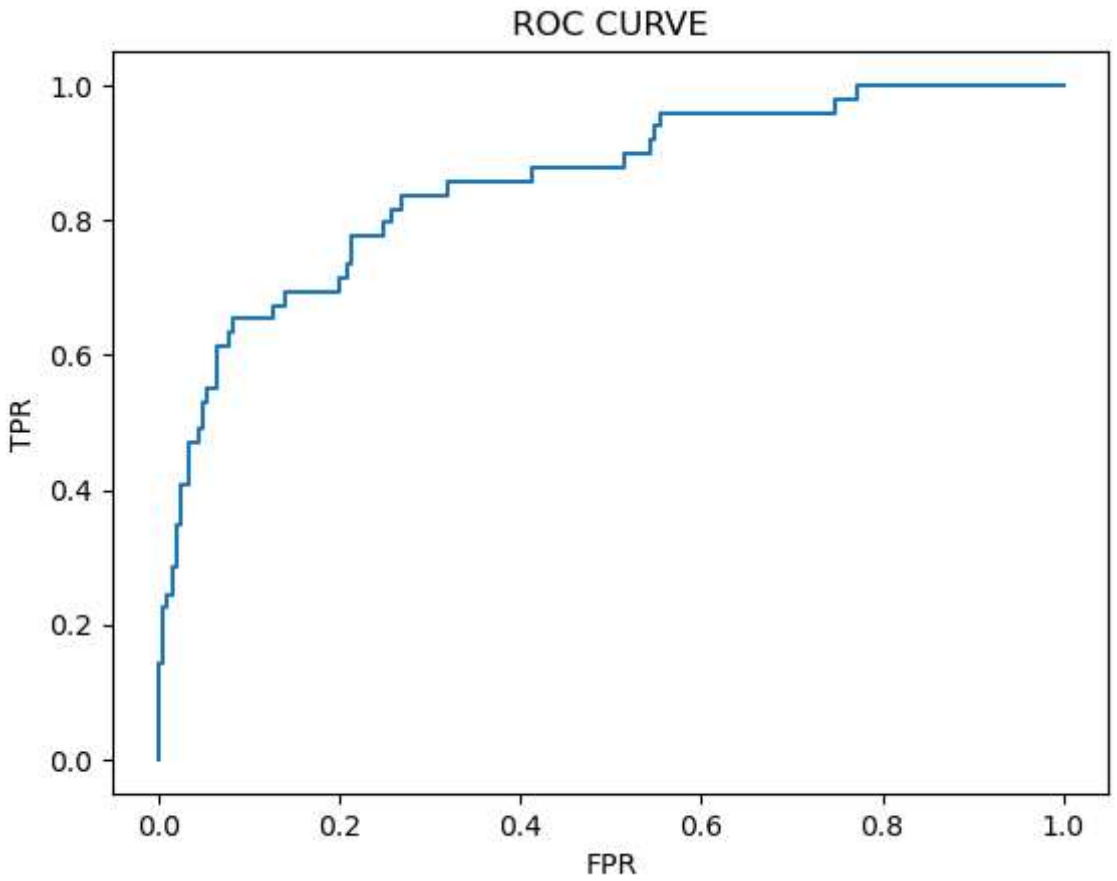
Out[50]:
```
array([6.61367680e-02, 8.06198087e-02, 5.95640954e-01, 1.42633628e-01,
       7.84184536e-01, 4.42409704e-02, 4.90141163e-01, 3.64319554e-02,
       8.44925839e-04, 4.30164151e-01, 3.70048030e-02, 1.96283567e-01,
       1.51964558e-02, 5.39471444e-01, 5.88816107e-02, 1.18804547e-02,
       1.13489514e-01, 3.82574810e-02, 2.32011787e-02, 2.66149260e-01,
       1.31568762e-01, 9.99913247e-03, 1.30286201e-02, 3.88975456e-02,
       7.51582081e-01, 4.58920880e-01, 6.43869830e-02, 2.97280577e-02,
       6.90209855e-01, 3.77607239e-02, 5.59306642e-03, 3.05902665e-01,
       3.65028773e-02, 3.90288488e-02, 2.27899092e-02, 5.90358760e-03,
       2.32731579e-01, 5.29386349e-02, 1.38241033e-02, 1.13353905e-01,
       3.42777217e-02, 8.33820691e-03, 1.07151981e-03, 5.52578825e-03,
       8.80282261e-03, 5.59254552e-01, 4.21978392e-01, 5.99712843e-04,
       4.00447399e-01, 3.68527002e-01, 3.55627577e-02, 8.16716675e-01,
       1.58976261e-02, 3.55101528e-01, 4.94410470e-01, 2.54453615e-01,
       1.03880530e-02, 4.22173742e-01, 2.22235299e-02, 2.95631863e-01,
       1.26998325e-02, 1.53660325e-01, 1.65718568e-01, 2.37981819e-02,
       1.13922041e-01, 2.33156457e-02, 3.48112915e-01, 1.78088100e-01,
       1.21220858e-01, 1.82213934e-01, 3.38118048e-02, 1.58551253e-01,
       9.58027931e-02, 3.08386596e-02, 7.44773642e-02, 4.13402445e-02,
       1.92788840e-02, 2.80415152e-02, 4.93838215e-01, 1.65502835e-02,
       3.08277540e-03, 1.74647710e-02, 2.61191408e-01, 1.97665118e-02,
       1.95892874e-02, 7.95183218e-02, 6.72183840e-04, 1.32195797e-02,
       1.24777389e-02, 6.32250258e-02, 5.79625808e-02, 7.28350500e-02,
       3.24892323e-01, 1.73612641e-01, 8.28152031e-04, 7.99639374e-02,
       4.97002372e-01, 6.41780395e-01, 5.52704989e-02, 7.62263342e-02,
       1.32740233e-01, 5.87050953e-01, 5.91342152e-01, 5.08443347e-03,
       1.06560968e-01, 6.69002781e-03, 6.09068154e-02, 2.11328359e-01,
       3.68701854e-02, 3.96643551e-01, 2.94309518e-02, 5.18292700e-02,
       3.44618460e-03, 2.96577876e-01, 2.24618906e-02, 2.45843670e-02,
       9.84944236e-03, 2.82977886e-02, 2.36629032e-03, 4.89083971e-03,
       1.06593614e-01, 3.22277227e-02, 8.43908260e-02, 8.89755562e-01,
       1.01655842e-02, 2.07510840e-03, 3.14944475e-03, 9.91294127e-02,
       1.39271996e-01, 1.05763678e-02, 4.62491869e-03, 3.29517891e-01,
       7.00654301e-01, 1.96667235e-01, 9.97786091e-03, 3.85631608e-01,
       6.96550561e-01, 2.28516613e-01, 1.82557187e-01, 4.88667496e-01,
       2.73199468e-02, 3.53431498e-02, 3.17068584e-02, 3.13677515e-01,
       4.71792593e-01, 3.75527841e-02, 2.67239299e-01, 3.83012181e-03,
       7.37038139e-02, 1.24740256e-01, 4.07667992e-03, 1.71768746e-01,
       7.20048951e-02, 1.75320014e-01, 7.13178095e-03, 2.39429405e-02,
       1.98658569e-01, 3.20547275e-01, 3.96927966e-03, 9.49330798e-03,
       7.45555587e-01, 3.12330244e-03, 3.32408074e-02, 9.19392780e-01,
       1.48852250e-02, 2.57190744e-01, 1.73014525e-01, 1.51785386e-01,
       1.86931379e-02, 8.28851842e-04, 2.54342547e-01, 4.56483830e-02,
       3.99204597e-02, 1.25490112e-01, 1.70919482e-02, 8.25299901e-02,
       4.38992386e-02, 4.31506741e-02, 3.39152219e-02, 5.12964217e-02,
       2.12219051e-02, 1.57474151e-01, 2.44952859e-03, 8.22895146e-01,
       5.98438753e-02, 8.69783952e-02, 5.40701108e-01, 8.54091278e-03,
       5.57078600e-01, 2.31232882e-01, 2.11246138e-01, 2.75461703e-01,
       1.33154014e-01, 1.82199815e-02, 3.32865074e-02, 1.09429647e-01,
       1.86460970e-02, 7.81476677e-03, 2.65574308e-01, 1.69272765e-02,
       3.86686532e-01, 2.20988923e-01, 8.08612471e-01, 2.03966645e-02,
       1.28650474e-01, 1.77680007e-02, 3.57289398e-01, 5.30474583e-04,
       1.67105418e-01, 1.00764993e-02, 8.58290380e-02, 3.05739812e-01,
       6.81826517e-02, 3.81711531e-01, 1.10859745e-01, 4.71842280e-03,
       1.86894274e-02, 5.45048658e-02, 4.72421338e-02, 9.34489342e-02,
       8.33653672e-02, 4.44029727e-01, 5.18663398e-01, 1.79271982e-01,
       2.87090882e-01, 2.67710024e-03, 1.09509162e-01, 2.65241921e-01,
       7.65032998e-01, 4.64180665e-02, 1.97677279e-01, 2.06495230e-01,
       3.00475823e-02, 4.28680929e-02, 9.01257179e-02, 1.16204269e-01,
       2.66591825e-01, 9.07332777e-04, 7.47052094e-02, 1.49451887e-03,
       1.41520152e-01, 2.92363790e-01, 5.07029822e-03, 1.26501783e-01,
       3.79539275e-02, 1.77150306e-02, 1.54456455e-01, 3.43384106e-01,
       4.27481928e-02, 3.44423944e-02, 2.93165700e-01, 7.16660273e-02,
       5.38151329e-01, 1.63749013e-02, 1.23951727e-01, 6.52278702e-02,
```

```
2.45783680e-03, 6.25002778e-01, 6.25608255e-01, 3.74616723e-01,
1.41256456e-01, 2.84604841e-02, 2.91843045e-01, 5.95069780e-02,
3.21045151e-02, 6.99486994e-02, 1.15155181e-03, 4.47317458e-01,
5.39084197e-01, 2.55697184e-02, 2.85616158e-02, 8.43094503e-03,
6.55595901e-02, 2.48427612e-02, 9.60265542e-03, 5.93422188e-03,
8.58521685e-02, 3.64680251e-01, 1.20449475e-01, 1.84226608e-01,
6.91761349e-01, 3.24455148e-03, 4.85788438e-02, 7.34675868e-02,
5.35355633e-02, 7.58287245e-02, 2.30602991e-04, 1.53563842e-01,
1.43056931e-03, 9.15060156e-03, 1.08956580e-01, 6.50047900e-01,
2.36548949e-02, 7.52606168e-02])
```

In [51]:
```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
y_test = label_encoder.fit_transform(y_test)
```

In [52]:
```
fpr,tpr,threshsholds = roc_curve(y_test,probability)
```

In [53]:
```
plt.plot(fpr,tpr)
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('ROC CURVE')
plt.show()
```



Decision Tree

In [54]:
```
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier()
```

In [113…
```
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,ra
```

In [67]:
```
dtc.fit(x_train,y_train)
```

Out[67]:
```
DecisionTreeClassifier()
```

In [68]:
```
y_pred1 = dtc.predict(x_test)
y_pred1
```

Out[68]:
```
array(['No', 'No', 'Yes', 'No', 'No', 'Yes', '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', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No',
       'No', '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', 'No', 'No', 'No', 'Yes', 'No', 'No',
       'Yes', 'No', 'No', 'No', 'Yes', 'No', 'No', 'Yes', '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', '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', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'Yes', 'No', 'No', 'No',
       'No', 'Yes', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'Yes',
       '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', 'Yes', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No',
       'No', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No',
       'No', 'No', 'No', 'No', 'Yes', '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', 'Yes', 'No',
       'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes',
       'No', 'Yes', 'No', 'No', 'Yes', 'No', 'No', 'Yes', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No'], dtype=object)
```

In [69]:
```
y_test
```

Out[69]:
```
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 [70]:
```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,r
```

In [71]:
```
accuracy_score(y_test,y_pred1)
```

Out[71]:
```
0.7551020408163265
```

In [73]:
```
confusion_matrix(y_test,y_pred1)
```

Out[73]:
```
array([[208,  37],
       [ 35,  14]], dtype=int64)
```

In [74]:
```
pd.crosstab(y_test,y_pred1)
```

Out[74]:

col_0	No	Yes
Attrition		
No	208	37
Yes	35	14

In [75]:
```
print(classification_report(y_test,y_pred1))
```

```
              precision    recall  f1-score   support

          No       0.86      0.85      0.85       245
         Yes       0.27      0.29      0.28        49

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

In [77]:
```
probability=dtc.predict_proba(x_test)[:,1]
probability
```

Out[77]:
```
array([0., 0., 1., 0., 0., 1., 1., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0.,
       0., 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.,
       1., 0., 0., 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., 0., 0., 0., 1., 0.,
       0., 1., 0., 0., 0., 1., 0., 0., 1., 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., 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., 1., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 1., 1., 0., 0., 0., 0., 1., 0., 0., 1.,
       0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0.,
       0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0.,
       0., 1., 0., 0., 0., 0., 1., 0., 1., 0., 1., 0., 1., 0., 0., 0., 0.,
       0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 1., 0., 1., 0., 0., 1., 0., 0., 1., 0., 0., 0.,
       0., 0., 0., 0., 0.])
```

Hyper parameter tuning

In [79]:
```
from sklearn import tree
plt.figure(figsize=(25,15))
tree.plot_tree(dtc,filled=True)
```

Out[79]:
```
[Text(0.3329547827903091, 0.96875, 'X[19] <= -1.257\ngini = 0.269\nsamples = 1176
\nvalue = [988, 188]'),
 Text(0.08688387635756056, 0.90625, 'X[45] <= 0.387\ngini = 0.5\nsamples = 78\nval
ue = [39, 39]'),
 Text(0.05012531328320802, 0.84375, 'X[2] <= 0.902\ngini = 0.426\nsamples = 39\nva
lue = [27, 12]'),
 Text(0.03341687552213868, 0.78125, 'X[26] <= 0.797\ngini = 0.312\nsamples = 31\nv
alue = [25, 6]'),
 Text(0.020050125313283207, 0.71875, 'X[10] <= -1.114\ngini = 0.198\nsamples = 27
\nvalue = [24, 3]'),
 Text(0.013366750208855471, 0.65625, 'X[46] <= 0.482\ngini = 0.5\nsamples = 6\nval
ue = [3, 3]'),
 Text(0.006683375104427736, 0.59375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.020050125313283207, 0.59375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.026733500417710943, 0.65625, 'gini = 0.0\nsamples = 21\nvalue = [21, 0]'),
 Text(0.04678362573099415, 0.71875, 'X[7] <= -1.102\ngini = 0.375\nsamples = 4\nva
lue = [1, 3]'),
 Text(0.040100250626566414, 0.65625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.053467000835421885, 0.65625, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.06683375104427736, 0.78125, 'X[14] <= 1.446\ngini = 0.375\nsamples = 8\nva
lue = [2, 6]'),
 Text(0.06015037593984962, 0.71875, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.07351712614870509, 0.71875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.12364243943191311, 0.84375, 'X[41] <= 0.755\ngini = 0.426\nsamples = 39\nv
alue = [12, 27]'),
 Text(0.10025062656641603, 0.78125, 'X[32] <= 0.397\ngini = 0.26\nsamples = 26\nva
lue = [4, 22]'),
 Text(0.08688387635756056, 0.71875, 'X[7] <= 1.482\ngini = 0.095\nsamples = 20\nva
lue = [1, 19]'),
 Text(0.08020050125313283, 0.65625, 'gini = 0.0\nsamples = 18\nvalue = [0, 18]'),
 Text(0.0935672514619883, 0.65625, 'X[13] <= -0.878\ngini = 0.5\nsamples = 2\nvalu
e = [1, 1]'),
 Text(0.08688387635756056, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.10025062656641603, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.1136173767752715, 0.71875, 'X[5] <= -0.401\ngini = 0.5\nsamples = 6\nvalue
= [3, 3]'),
 Text(0.10693400167084377, 0.65625, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.12030075187969924, 0.65625, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.14703425229741018, 0.78125, 'X[1] <= 0.712\ngini = 0.473\nsamples = 13\nva
lue = [8, 5]'),
 Text(0.14035087719298245, 0.71875, 'X[12] <= 1.103\ngini = 0.32\nsamples = 10\nva
lue = [8, 2]'),
 Text(0.1336675020885547, 0.65625, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(0.14703425229741018, 0.65625, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.15371762740183792, 0.71875, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.5790256892230576, 0.90625, 'X[46] <= 0.482\ngini = 0.235\nsamples = 1098\n
value = [949, 149]'),
 Text(0.33014828738512947, 0.84375, 'X[21] <= -1.786\ngini = 0.162\nsamples = 798
\nvalue = [727, 71]'),
 Text(0.18045112781954886, 0.78125, 'X[5] <= -0.173\ngini = 0.38\nsamples = 47\nva
lue = [35, 12]'),
 Text(0.1670843776106934, 0.71875, 'X[45] <= 0.387\ngini = 0.1\nsamples = 19\nvalu
e = [18, 1]'),
 Text(0.16040100250626566, 0.65625, 'gini = 0.0\nsamples = 18\nvalue = [18, 0]'),
 Text(0.17376775271512113, 0.65625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.19381787802840433, 0.71875, 'X[11] <= -0.789\ngini = 0.477\nsamples = 28\n
value = [17, 11]'),
 Text(0.1871345029239766, 0.65625, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.20050125313283207, 0.65625, 'X[5] <= 0.099\ngini = 0.413\nsamples = 24\nva
lue = [17, 7]'),
 Text(0.19381787802840433, 0.59375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.2071846282372598, 0.59375, 'X[25] <= 0.386\ngini = 0.351\nsamples = 22\nva
lue = [17, 5]'),
 Text(0.19381787802840433, 0.53125, 'X[1] <= -1.649\ngini = 0.133\nsamples = 14\nv
```

```
alue = [13, 1]'),
 Text(0.1871345029239766, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.20050125313283207, 0.46875, 'gini = 0.0\nsamples = 13\nvalue = [13, 0]'),
 Text(0.22055137844611528, 0.53125, 'X[1] <= -0.596\ngini = 0.5\nsamples = 8\nvalu
e = [4, 4]'),
 Text(0.21386800334168754, 0.46875, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.227234753550543, 0.46875, 'X[24] <= 2.58\ngini = 0.32\nsamples = 5\nvalue
= [4, 1]'),
 Text(0.22055137844611528, 0.40625, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.23391812865497075, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.47984544695071013, 0.78125, 'X[19] <= 3.564\ngini = 0.145\nsamples = 751\n
value = [692, 59]'),
 Text(0.4731620718462824, 0.71875, 'X[22] <= -0.41\ngini = 0.143\nsamples = 750\nv
alue = [692, 58]'),
 Text(0.3366750208855472, 0.65625, 'X[6] <= -1.118\ngini = 0.218\nsamples = 257\nv
alue = [225, 32]'),
 Text(0.3057644110275689, 0.59375, 'X[25] <= -0.455\ngini = 0.355\nsamples = 65\nv
alue = [50, 15]'),
 Text(0.28404344193817876, 0.53125, 'X[25] <= -1.016\ngini = 0.303\nsamples = 59\n
value = [48, 11]'),
 Text(0.2606516290726817, 0.46875, 'X[8] <= -0.323\ngini = 0.463\nsamples = 22\nva
lue = [14, 8]'),
 Text(0.24728487886382622, 0.40625, 'X[7] <= -1.151\ngini = 0.198\nsamples = 9\nva
lue = [8, 1]'),
 Text(0.24060150375939848, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.25396825396825395, 0.34375, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(0.27401837928153716, 0.40625, 'X[7] <= -0.388\ngini = 0.497\nsamples = 13\nv
alue = [6, 7]'),
 Text(0.2673350041771094, 0.34375, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.2807017543859649, 0.34375, 'X[2] <= -0.024\ngini = 0.346\nsamples = 9\nval
ue = [2, 7]'),
 Text(0.27401837928153716, 0.28125, 'X[5] <= 0.083\ngini = 0.444\nsamples = 3\nval
ue = [2, 1]'),
 Text(0.2673350041771094, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.2807017543859649, 0.21875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.28738512949039263, 0.28125, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.30743525480367584, 0.46875, 'X[10] <= -1.114\ngini = 0.149\nsamples = 37\n
value = [34, 3]'),
 Text(0.3007518796992481, 0.40625, 'X[21] <= -0.37\ngini = 0.5\nsamples = 6\nvalue
= [3, 3]'),
 Text(0.29406850459482037, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.30743525480367584, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.3141186299081036, 0.40625, 'gini = 0.0\nsamples = 31\nvalue = [31, 0]'),
 Text(0.32748538011695905, 0.53125, 'X[5] <= -1.479\ngini = 0.444\nsamples = 6\nva
lue = [2, 4]'),
 Text(0.3208020050125313, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.3341687552213868, 0.46875, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.36758563074352546, 0.59375, 'X[30] <= 0.178\ngini = 0.161\nsamples = 192\n
value = [175, 17]'),
 Text(0.3609022556390977, 0.53125, 'X[21] <= -0.37\ngini = 0.24\nsamples = 122\nva
lue = [105, 17]'),
 Text(0.34753550543024225, 0.46875, 'X[7] <= 0.399\ngini = 0.463\nsamples = 22\nva
lue = [14, 8]'),
 Text(0.3408521303258145, 0.40625, 'X[0] <= -0.156\ngini = 0.444\nsamples = 12\nva
lue = [4, 8]'),
 Text(0.3341687552213868, 0.34375, 'X[14] <= 0.626\ngini = 0.198\nsamples = 9\nval
ue = [1, 8]'),
 Text(0.32748538011695905, 0.28125, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
 Text(0.3408521303258145, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.34753550543024225, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.35421888053467, 0.40625, 'gini = 0.0\nsamples = 10\nvalue = [10, 0]'),
 Text(0.37426900584795321, 0.46875, 'X[1] <= -1.711\ngini = 0.164\nsamples = 100\nv
alue = [91, 9]'),
 Text(0.36758563074352546, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
```

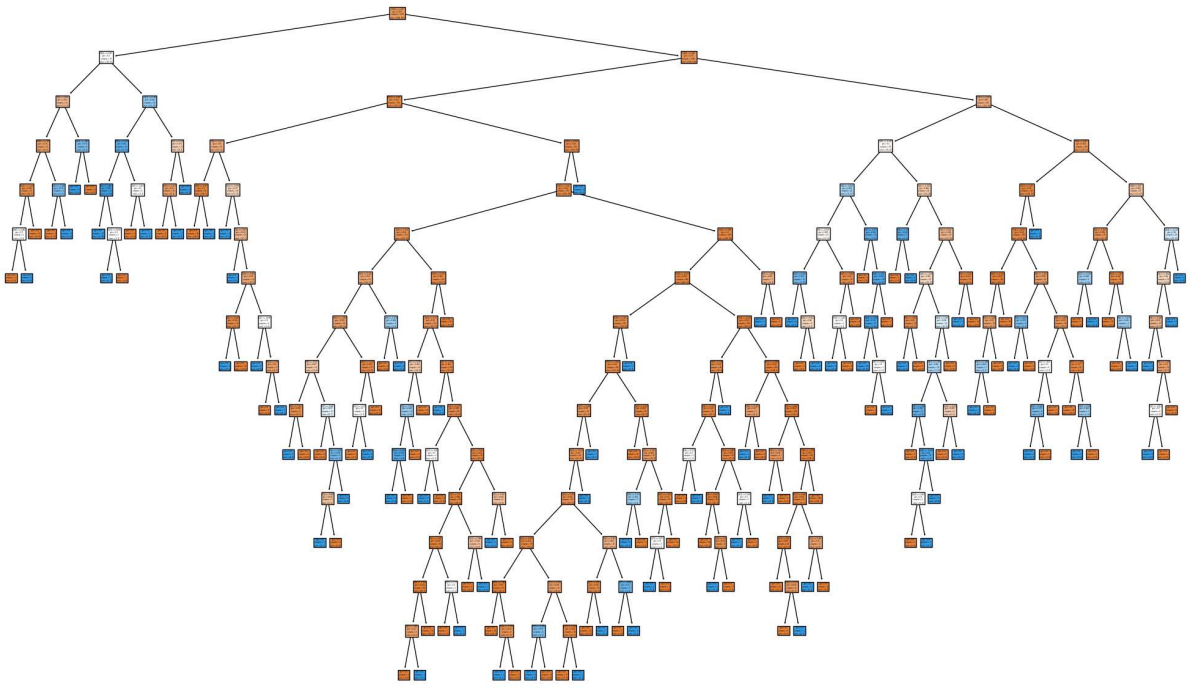
```
 Text(0.38095238095238093, 0.40625, 'X[5] <= -1.627\ngini = 0.149\nsamples = 99\nv
alue = [91, 8]'),
 Text(0.3617376775271512, 0.34375, 'X[37] <= 0.85\ngini = 0.5\nsamples = 4\nvalue
= [2, 2]'),
 Text(0.35505430242272346, 0.28125, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.3684210526315789, 0.28125, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.40016708437761067, 0.34375, 'X[1] <= 1.528\ngini = 0.118\nsamples = 95\nva
lue = [89, 6]'),
 Text(0.3817878028404344, 0.28125, 'X[3] <= 1.55\ngini = 0.086\nsamples = 89\nvalu
e = [85, 4]'),
 Text(0.36507936507936506, 0.21875, 'X[0] <= 2.144\ngini = 0.047\nsamples = 83\nva
lue = [81, 2]'),
 Text(0.3517126148705096, 0.15625, 'X[8] <= -1.729\ngini = 0.024\nsamples = 81\nva
lue = [80, 1]'),
 Text(0.34502923976608185, 0.09375, 'X[5] <= 1.183\ngini = 0.32\nsamples = 5\nvalu
e = [4, 1]'),
 Text(0.3383458646616541, 0.03125, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.3517126148705096, 0.03125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3583959899749373, 0.09375, 'gini = 0.0\nsamples = 76\nvalue = [76, 0]'),
 Text(0.37844611528822053, 0.15625, 'X[12] <= 1.27\ngini = 0.5\nsamples = 2\nvalue
= [1, 1]'),
 Text(0.3717627401837928, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.38512949039264827, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.39849624060150374, 0.21875, 'X[12] <= 0.348\ngini = 0.444\nsamples = 6\nva
lue = [4, 2]'),
 Text(0.391812865497076, 0.15625, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.4051796157059315, 0.15625, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.41854636591478694, 0.28125, 'X[0] <= -0.156\ngini = 0.444\nsamples = 6\nva
lue = [4, 2]'),
 Text(0.4118629908103592, 0.21875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.4252297410192147, 0.21875, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.3742690058479532, 0.53125, 'gini = 0.0\nsamples = 70\nvalue = [70, 0]'),
 Text(0.6096491228070176, 0.65625, 'X[22] <= 3.999\ngini = 0.1\nsamples = 493\nval
ue = [467, 26]'),
 Text(0.5735171261487051, 0.59375, 'X[10] <= -0.207\ngini = 0.094\nsamples = 486\n
value = [462, 24]'),
 Text(0.5213032581453634, 0.53125, 'X[43] <= 1.922\ngini = 0.154\nsamples = 191\nv
alue = [175, 16]'),
 Text(0.5146198830409356, 0.46875, 'X[12] <= -0.035\ngini = 0.145\nsamples = 190\n
value = [175, 15]'),
 Text(0.4903926482873851, 0.40625, 'X[25] <= 2.629\ngini = 0.221\nsamples = 95\nva
lue = [83, 12]'),
 Text(0.48370927318295737, 0.34375, 'X[12] <= -0.073\ngini = 0.207\nsamples = 94\n
value = [83, 11]'),
 Text(0.47702589807852963, 0.28125, 'X[26] <= 0.797\ngini = 0.192\nsamples = 93\nv
alue = [83, 10]'),
 Text(0.441938178780284, 0.21875, 'X[11] <= 0.439\ngini = 0.124\nsamples = 75\nval
ue = [70, 5]'),
 Text(0.41854636591478694, 0.15625, 'X[39] <= 1.346\ngini = 0.037\nsamples = 53\nv
alue = [52, 1]'),
 Text(0.4118629908103592, 0.09375, 'gini = 0.0\nsamples = 45\nvalue = [45, 0]'),
 Text(0.4252297410192147, 0.09375, 'X[16] <= -1.122\ngini = 0.219\nsamples = 8\nva
lue = [7, 1]'),
 Text(0.41854636591478694, 0.03125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.4319131161236424, 0.03125, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(0.4653299916457811, 0.15625, 'X[11] <= 0.705\ngini = 0.298\nsamples = 22\nva
lue = [18, 4]'),
 Text(0.4519632414369256, 0.09375, 'X[19] <= 0.285\ngini = 0.375\nsamples = 4\nval
ue = [1, 3]'),
 Text(0.4452798663324979, 0.03125, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.45864661654135336, 0.03125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.47869674185463656, 0.09375, 'X[13] <= 2.325\ngini = 0.105\nsamples = 18\nv
alue = [17, 1]'),
 Text(0.47201336675020883, 0.03125, 'gini = 0.0\nsamples = 17\nvalue = [17, 0]'),
```

```
 Text(0.4853801169590643, 0.03125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5121136173767753, 0.21875, 'X[45] <= 0.387\ngini = 0.401\nsamples = 18\nva
lue = [13, 5]'),
 Text(0.49874686716791977, 0.15625, 'X[37] <= 0.85\ngini = 0.142\nsamples = 13\nva
lue = [12, 1]'),
 Text(0.49206349206349204, 0.09375, 'gini = 0.0\nsamples = 12\nvalue = [12, 0]'),
 Text(0.5054302422723476, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5254803675856308, 0.15625, 'X[7] <= -0.585\ngini = 0.32\nsamples = 5\nvalu
e = [1, 4]'),
 Text(0.518796992481203, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5321637426900585, 0.09375, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.4903926482873851, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.49707602339181284, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5388471177944862, 0.40625, 'X[13] <= 0.724\ngini = 0.061\nsamples = 95\nva
lue = [92, 3]'),
 Text(0.5321637426900585, 0.34375, 'gini = 0.0\nsamples = 76\nvalue = [76, 0]'),
 Text(0.545530492898914, 0.34375, 'X[5] <= -1.149\ngini = 0.266\nsamples = 19\nval
ue = [16, 3]'),
 Text(0.5321637426900585, 0.28125, 'X[16] <= 0.729\ngini = 0.444\nsamples = 3\nval
ue = [1, 2]'),
 Text(0.5254803675856308, 0.21875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.5388471177944862, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5588972431077694, 0.28125, 'X[25] <= -0.876\ngini = 0.117\nsamples = 16\nv
alue = [15, 1]'),
 Text(0.5522138680033417, 0.21875, 'X[19] <= 0.671\ngini = 0.5\nsamples = 2\nvalue
= [1, 1]'),
 Text(0.545530492898914, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5588972431077694, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5655806182121972, 0.21875, 'gini = 0.0\nsamples = 14\nvalue = [14, 0]'),
 Text(0.5279866332497911, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6257309941520468, 0.53125, 'X[14] <= -1.014\ngini = 0.053\nsamples = 295\n
value = [287, 8]'),
 Text(0.6023391812865497, 0.46875, 'X[24] <= 2.58\ngini = 0.159\nsamples = 46\nval
ue = [42, 4]'),
 Text(0.595655806182122, 0.40625, 'X[8] <= -1.729\ngini = 0.124\nsamples = 45\nval
ue = [42, 3]'),
 Text(0.5789473684210527, 0.34375, 'X[6] <= 0.712\ngini = 0.5\nsamples = 2\nvalue
= [1, 1]'),
 Text(0.5722639933166249, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5856307435254804, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6123642439431913, 0.34375, 'X[19] <= 2.085\ngini = 0.089\nsamples = 43\nva
lue = [41, 2]'),
 Text(0.5989974937343359, 0.28125, 'X[0] <= 0.665\ngini = 0.048\nsamples = 41\nval
ue = [40, 1]'),
 Text(0.5923141186299081, 0.21875, 'gini = 0.0\nsamples = 33\nvalue = [33, 0]'),
 Text(0.6056808688387636, 0.21875, 'X[19] <= -0.1\ngini = 0.219\nsamples = 8\nvalu
e = [7, 1]'),
 Text(0.5989974937343359, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6123642439431913, 0.15625, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(0.6257309941520468, 0.28125, 'X[22] <= 0.244\ngini = 0.5\nsamples = 2\nvalue
= [1, 1]'),
 Text(0.6190476190476191, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6324143692564745, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.6090225563909775, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6491228070175439, 0.46875, 'X[11] <= -0.943\ngini = 0.032\nsamples = 249\n
value = [245, 4]'),
 Text(0.6324143692564745, 0.40625, 'X[18] <= -0.345\ngini = 0.32\nsamples = 5\nval
ue = [4, 1]'),
 Text(0.6257309941520468, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6390977443609023, 0.34375, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.6658312447786132, 0.40625, 'X[1] <= -1.685\ngini = 0.024\nsamples = 244\nv
alue = [241, 3]'),
 Text(0.6524644945697577, 0.34375, 'X[35] <= -0.204\ngini = 0.278\nsamples = 6\nva
lue = [5, 1]'),
```

```
 Text(0.64578111946533, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6591478696741855, 0.28125, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.6791979949874687, 0.34375, 'X[16] <= -1.122\ngini = 0.017\nsamples = 238\n
value = [236, 2]'),
 Text(0.672514619883041, 0.28125, 'X[21] <= 1.046\ngini = 0.073\nsamples = 53\nval
ue = [51, 2]'),
 Text(0.6591478696741855, 0.21875, 'X[37] <= 0.85\ngini = 0.041\nsamples = 48\nval
ue = [47, 1]'),
 Text(0.6524644945697577, 0.15625, 'gini = 0.0\nsamples = 42\nvalue = [42, 0]'),
 Text(0.6658312447786132, 0.15625, 'X[14] <= 0.079\ngini = 0.278\nsamples = 6\nval
ue = [5, 1]'),
 Text(0.6591478696741855, 0.09375, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.672514619883041, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6858813700918964, 0.21875, 'X[11] <= 0.365\ngini = 0.32\nsamples = 5\nvalu
e = [4, 1]'),
 Text(0.6791979949874687, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6925647451963242, 0.15625, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.6858813700918964, 0.28125, 'gini = 0.0\nsamples = 185\nvalue = [185, 0]'),
 Text(0.64578111946533, 0.59375, 'X[1] <= -0.467\ngini = 0.408\nsamples = 7\nvalue
= [5, 2]'),
 Text(0.6390977443609023, 0.53125, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.6524644945697577, 0.53125, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.48652882205513787, 0.71875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8279030910609858, 0.84375, 'X[11] <= -0.533\ngini = 0.385\nsamples = 300\n
value = [222, 78]'),
 Text(0.7451963241436925, 0.78125, 'X[18] <= -0.345\ngini = 0.5\nsamples = 96\nval
ue = [49, 47]'),
 Text(0.7126148705096074, 0.71875, 'X[2] <= -0.456\ngini = 0.459\nsamples = 42\nva
lue = [15, 27]'),
 Text(0.6925647451963242, 0.65625, 'X[5] <= -0.275\ngini = 0.499\nsamples = 23\nva
lue = [12, 11]'),
 Text(0.672514619883041, 0.59375, 'X[12] <= 0.245\ngini = 0.355\nsamples = 13\nval
ue = [3, 10]'),
 Text(0.6658312447786132, 0.53125, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
 Text(0.6791979949874687, 0.53125, 'X[20] <= 0.544\ngini = 0.48\nsamples = 5\nvalu
e = [3, 2]'),
 Text(0.672514619883041, 0.46875, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.6858813700918964, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.7126148705096074, 0.59375, 'X[19] <= -0.936\ngini = 0.18\nsamples = 10\nva
lue = [9, 1]'),
 Text(0.7059314954051796, 0.53125, 'X[34] <= 1.435\ngini = 0.5\nsamples = 2\nvalue
= [1, 1]'),
 Text(0.6992481203007519, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.7126148705096074, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.7192982456140351, 0.53125, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(0.7326649958228906, 0.65625, 'X[7] <= -1.077\ngini = 0.266\nsamples = 19\nva
lue = [3, 16]'),
 Text(0.7259816207184628, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.7393483709273183, 0.59375, 'X[9] <= -0.51\ngini = 0.198\nsamples = 18\nval
ue = [2, 16]'),
 Text(0.7326649958228906, 0.53125, 'X[24] <= 1.339\ngini = 0.111\nsamples = 17\nva
lue = [1, 16]'),
 Text(0.7259816207184628, 0.46875, 'gini = 0.0\nsamples = 15\nvalue = [0, 15]'),
 Text(0.7393483709273183, 0.46875, 'X[3] <= 0.085\ngini = 0.5\nsamples = 2\nvalue
= [1, 1]'),
 Text(0.7326649958228906, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.746031746031746, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.746031746031746, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.7777777777777778, 0.71875, 'X[0] <= -1.141\ngini = 0.466\nsamples = 54\nva
lue = [34, 20]'),
 Text(0.7593984962406015, 0.65625, 'X[5] <= -1.14\ngini = 0.245\nsamples = 7\nvalu
e = [1, 6]'),
 Text(0.7527151211361738, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.7660818713450293, 0.59375, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
```

```
 Text(0.7961570593149541, 0.65625, 'X[1] <= 0.419\ngini = 0.418\nsamples = 47\nval
ue = [33, 14]'),
 Text(0.7794486215538847, 0.59375, 'X[1] <= -1.236\ngini = 0.482\nsamples = 32\nva
lue = [19, 13]'),
 Text(0.7660818713450293, 0.53125, 'X[1] <= -1.655\ngini = 0.18\nsamples = 10\nval
ue = [9, 1]'),
 Text(0.7593984962406015, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.772765246449457, 0.46875, 'gini = 0.0\nsamples = 9\nvalue = [9, 0]'),
 Text(0.7928153717627402, 0.53125, 'X[12] <= 1.329\ngini = 0.496\nsamples = 22\nva
lue = [10, 12]'),
 Text(0.7861319966583125, 0.46875, 'X[41] <= 0.755\ngini = 0.465\nsamples = 19\nva
lue = [7, 12]'),
 Text(0.772765246449457, 0.40625, 'X[5] <= -1.199\ngini = 0.298\nsamples = 11\nval
ue = [2, 9]'),
 Text(0.7660818713450293, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.7794486215538847, 0.34375, 'X[16] <= -0.659\ngini = 0.18\nsamples = 10\nva
lue = [1, 9]'),
 Text(0.772765246449457, 0.28125, 'X[34] <= 1.435\ngini = 0.5\nsamples = 2\nvalue
= [1, 1]'),
 Text(0.7660818713450293, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.7794486215538847, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.7861319966583125, 0.28125, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
 Text(0.7994987468671679, 0.40625, 'X[14] <= -0.057\ngini = 0.469\nsamples = 8\nva
lue = [5, 3]'),
 Text(0.7928153717627402, 0.34375, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.8061821219715957, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.7994987468671679, 0.46875, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.8128654970760234, 0.59375, 'X[7] <= -1.545\ngini = 0.124\nsamples = 15\nva
lue = [14, 1]'),
 Text(0.8061821219715957, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8195488721804511, 0.53125, 'gini = 0.0\nsamples = 14\nvalue = [14, 0]'),
 Text(0.910609857978279, 0.78125, 'X[45] <= 0.387\ngini = 0.258\nsamples = 204\nva
lue = [173, 31]'),
 Text(0.8646616541353384, 0.71875, 'X[11] <= 2.837\ngini = 0.138\nsamples = 147\nv
alue = [136, 11]'),
 Text(0.8579782790309106, 0.65625, 'X[2] <= 0.655\ngini = 0.128\nsamples = 146\nva
lue = [136, 10]'),
 Text(0.8395989974937343, 0.59375, 'X[22] <= -0.736\ngini = 0.038\nsamples = 104\n
value = [102, 2]'),
 Text(0.8329156223893066, 0.53125, 'X[7] <= -1.102\ngini = 0.32\nsamples = 10\nval
ue = [8, 2]'),
 Text(0.8262322472848789, 0.46875, 'X[11] <= 1.248\ngini = 0.444\nsamples = 3\nval
ue = [1, 2]'),
 Text(0.8195488721804511, 0.40625, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.8329156223893066, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.8395989974937343, 0.46875, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(0.8462823725981621, 0.53125, 'gini = 0.0\nsamples = 94\nvalue = [94, 0]'),
 Text(0.8763575605680869, 0.59375, 'X[6] <= -1.118\ngini = 0.308\nsamples = 42\nva
lue = [34, 8]'),
 Text(0.8596491228070176, 0.53125, 'X[1] <= 0.813\ngini = 0.375\nsamples = 4\nvalu
e = [1, 3]'),
 Text(0.8529657477025898, 0.46875, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.8663324979114453, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.8930659983291562, 0.53125, 'X[0] <= -0.265\ngini = 0.229\nsamples = 38\nva
lue = [33, 5]'),
 Text(0.8796992481203008, 0.46875, 'X[2] <= 1.827\ngini = 0.5\nsamples = 6\nvalue
= [3, 3]'),
 Text(0.873015873015873, 0.40625, 'X[5] <= 0.184\ngini = 0.375\nsamples = 4\nvalue
= [1, 3]'),
 Text(0.8663324979114453, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.8796992481203008, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.8863826232247285, 0.40625, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.9064327485380117, 0.46875, 'X[31] <= 1.262\ngini = 0.117\nsamples = 32\nva
lue = [30, 2]'),
```

```
 Text(0.899749373433584, 0.40625, 'gini = 0.0\nsamples = 29\nvalue = [29, 0]'),
 Text(0.9131161236424394, 0.40625, 'X[15] <= 0.96\ngini = 0.444\nsamples = 3\nvalu
e = [1, 2]'),
 Text(0.9064327485380117, 0.34375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.9197994987468672, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.8713450292397661, 0.65625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.9565580618212197, 0.71875, 'X[42] <= 0.67\ngini = 0.456\nsamples = 57\nval
ue = [37, 20]'),
 Text(0.9264828738512949, 0.65625, 'X[5] <= -1.458\ngini = 0.238\nsamples = 29\nva
lue = [25, 4]'),
 Text(0.9131161236424394, 0.59375, 'X[11] <= -0.365\ngini = 0.444\nsamples = 3\nva
lue = [1, 2]'),
 Text(0.9064327485380117, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.9197994987468672, 0.53125, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.9398496240601504, 0.59375, 'X[24] <= 1.183\ngini = 0.142\nsamples = 26\nva
lue = [24, 2]'),
 Text(0.9331662489557226, 0.53125, 'gini = 0.0\nsamples = 23\nvalue = [23, 0]'),
 Text(0.9465329991645781, 0.53125, 'X[8] <= -0.323\ngini = 0.444\nsamples = 3\nval
ue = [1, 2]'),
 Text(0.9398496240601504, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.9532163742690059, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.9866332497911445, 0.65625, 'X[24] <= -0.214\ngini = 0.49\nsamples = 28\nva
lue = [12, 16]'),
 Text(0.9799498746867168, 0.59375, 'X[2] <= 1.765\ngini = 0.48\nsamples = 20\nvalu
e = [12, 8]'),
 Text(0.9732664995822891, 0.53125, 'X[2] <= -0.949\ngini = 0.415\nsamples = 17\nva
lue = [12, 5]'),
 Text(0.9665831244778613, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.9799498746867168, 0.46875, 'X[6] <= -1.118\ngini = 0.32\nsamples = 15\nval
ue = [12, 3]'),
 Text(0.9732664995822891, 0.40625, 'X[0] <= -0.211\ngini = 0.5\nsamples = 6\nvalue
= [3, 3]'),
 Text(0.9665831244778613, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.9799498746867168, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.9866332497911445, 0.40625, 'gini = 0.0\nsamples = 9\nvalue = [9, 0]'),
 Text(0.9866332497911445, 0.53125, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.9933166248955723, 0.59375, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]')]
```



In [81]:

```
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 [85]:
```
grid_search=GridSearchCV(estimator=dtc,param_grid=parameter,cv=5,scoring="accuracy"
```

In [87]:
```
grid_search.fit(x_train,y_train)
```

Out[87]:
```
GridSearchCV(cv=5, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [1, 2, 3, 4, 5],
                         'max_features': ['auto', 'sqrt', 'log2'],
                         'splitter': ['best', 'random']},
             scoring='accuracy')
```

In [88]:
```
grid_search.best_params_
```

Out[88]:
```
{'criterion': 'entropy',
 'max_depth': 2,
 'max_features': 'sqrt',
 'splitter': 'best'}
```

In [89]:
```
dtc_cv = DecisionTreeClassifier(criterion ='entropy',max_depth=3,max_features='sqr
dtc_cv.fit(x_train,y_train)
```

Out[89]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=3, max_features='sqrt')
```

In [90]:
```
y_pred = dtc_cv.predict(x_test)
```

In [91]:
```
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

          No       0.85      0.96      0.90       245
         Yes       0.41      0.14      0.21        49

    accuracy                           0.82       294
   macro avg       0.63      0.55      0.56       294
weighted avg       0.78      0.82      0.79       294
```

Random Forest

In [92]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
```

In [93]:
```
rfc.fit(x_train,y_train)
```

Out[93]:
```
RandomForestClassifier()
```

In [95]:
```
y_pred2= dtc.predict(x_test)
y_pred2
```

Out[95]:
```
array(['No', 'No', 'Yes', 'No', 'No', 'Yes', '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', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No',
       'No', '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', 'No', 'No', 'No', 'Yes', 'No', 'No',
       'Yes', 'No', 'No', 'No', 'Yes', 'No', 'No', 'Yes', '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', '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', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'Yes', 'No', 'No', 'No',
       'No', 'Yes', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'Yes',
       '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', 'Yes', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No',
       'No', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No',
       'No', 'No', 'No', 'No', 'Yes', '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', 'Yes', 'No',
       'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes',
       'No', 'Yes', 'No', 'No', 'Yes', 'No', 'No', 'Yes', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No'], dtype=object)
```

In [96]:
```
y_test
```

Out[96]:
```
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 [97]:
```
forest_params=[{'max_depth':list(range(10,15)),'max_features':list(range(10,15))}]
```

In [100…
```
rfc_cv=GridSearchCV(rfc,param_grid=forest_params,cv=10,scoring='accuracy')
```

In [102…
```
rfc_cv.fit(x_train,y_train)
```

Out[102]:
```
GridSearchCV(cv=10, estimator=RandomForestClassifier(),
             param_grid=[{'max_depth': [10, 11, 12, 13, 14],
                          'max_features': [10, 11, 12, 13, 14]}],
             scoring='accuracy')
```

In [103…
```
y_pred = rfc_cv.predict(x_test)
```

In [104…
```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,
```

In [105…
```
accuracy_score(y_test,y_pred2)
```

Out[105]:
```
0.7551020408163265
```

In [106…
```
confusion_matrix(y_test,y_pred2)
```

Out[106]:
```
array([[208,  37],
       [ 35,  14]], dtype=int64)
```

In [107…
```
pd.crosstab(y_test,y_pred2)
```

Out[107]:

Attrition	No	Yes
No	208	37
Yes	35	14

In [108…
```
print(classification_report(y_test,y_pred2))
```

```
              precision    recall  f1-score   support

          No       0.86      0.85      0.85       245
         Yes       0.27      0.29      0.28        49

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

In [109…
```
rfc_cv.best_params_
```

Out[109]:
```
{'max_depth': 14, 'max_features': 13}
```

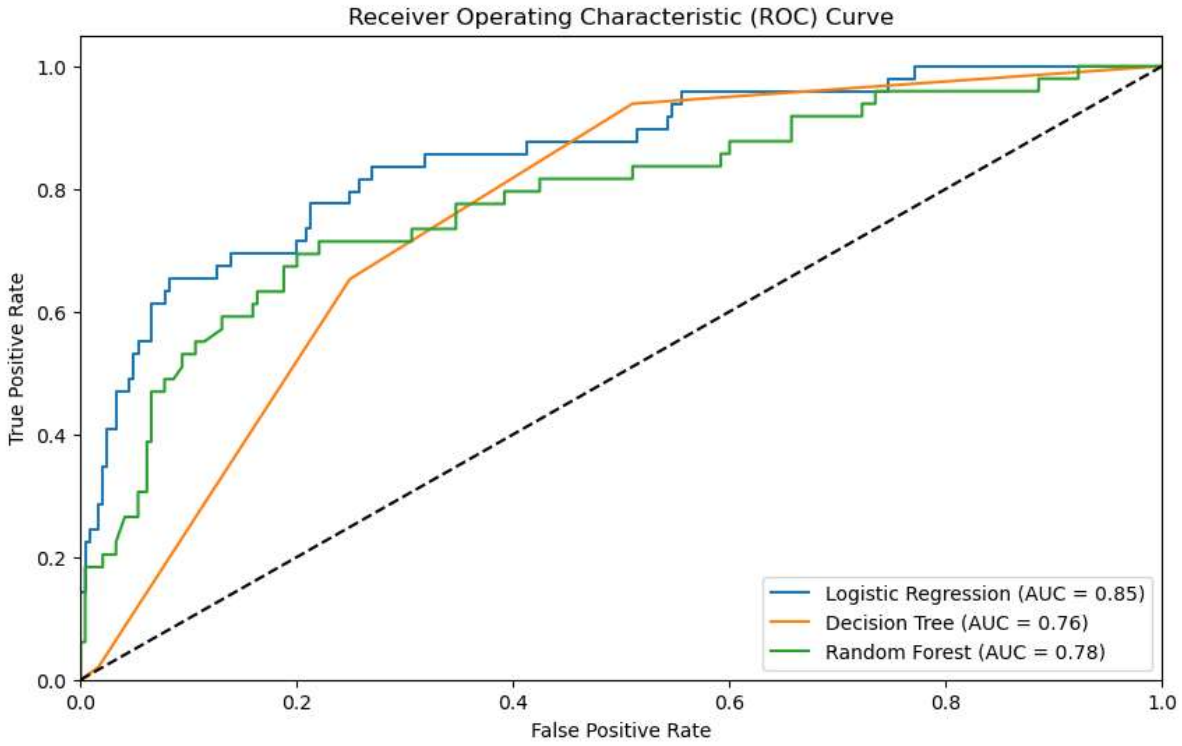
In [110…
```
probability=rfc.predict_proba(x_test)[:,1]
```

In [111…
```
probability
```

Out[111]:
```
array([0.06, 0.08, 0.22, 0.14, 0.73, 0.43, 0.35, 0.11, 0.1 , 0.2 , 0.06,
       0.08, 0.06, 0.49, 0.04, 0.03, 0.1 , 0.16, 0.1 , 0.21, 0.47, 0.05,
       0.03, 0.1 , 0.39, 0.15, 0.08, 0.05, 0.6 , 0.06, 0.05, 0.15, 0.18,
       0.1 , 0.15, 0.03, 0.22, 0.11, 0.07, 0.25, 0.23, 0.08, 0.08, 0.14,
       0.11, 0.39, 0.3 , 0.03, 0.65, 0.38, 0.16, 0.45, 0.17, 0.16, 0.48,
       0.07, 0.06, 0.07, 0.04, 0.37, 0.08, 0.21, 0.07, 0.16, 0.27, 0.1 ,
       0.23, 0.13, 0.08, 0.21, 0.18, 0.3 , 0.06, 0.07, 0.15, 0.14, 0.11,
       0.11, 0.38, 0.07, 0.04, 0.1 , 0.2 , 0.11, 0.19, 0.08, 0.12, 0.18,
       0.07, 0.05, 0.54, 0.07, 0.08, 0.25, 0.09, 0.03, 0.13, 0.33, 0.18,
       0.2 , 0.15, 0.25, 0.29, 0.08, 0.1 , 0.07, 0.08, 0.38, 0.27, 0.18,
       0.16, 0.2 , 0.05, 0.06, 0.15, 0.14, 0.04, 0.13, 0.05, 0.02, 0.12,
       0.02, 0.03, 0.52, 0.22, 0.13, 0.03, 0.1 , 0.09, 0.09, 0.06, 0.38,
       0.37, 0.21, 0.21, 0.15, 0.35, 0.15, 0.18, 0.19, 0.11, 0.15, 0.06,
       0.18, 0.21, 0.12, 0.08, 0.03, 0.14, 0.16, 0.15, 0.22, 0.14, 0.2 ,
       0.15, 0.14, 0.17, 0.2 , 0.16, 0.07, 0.28, 0.09, 0.19, 0.56, 0.09,
       0.18, 0.13, 0.1 , 0.14, 0.03, 0.12, 0.06, 0.09, 0.19, 0.11, 0.34,
       0.11, 0.18, 0.15, 0.14, 0.09, 0.04, 0.08, 0.54, 0.04, 0.04, 0.36,
       0.08, 0.21, 0.24, 0.27, 0.4 , 0.15, 0.02, 0.16, 0.18, 0.12, 0.03,
       0.35, 0.07, 0.28, 0.12, 0.25, 0.03, 0.07, 0.11, 0.37, 0.07, 0.16,
       0.03, 0.07, 0.21, 0.12, 0.18, 0.06, 0.07, 0.09, 0.25, 0.11, 0.34,
       0.09, 0.28, 0.36, 0.26, 0.13, 0.07, 0.12, 0.29, 0.57, 0.18, 0.02,
       0.31, 0.07, 0.09, 0.07, 0.23, 0.23, 0.03, 0.08, 0.08, 0.22, 0.13,
       0.06, 0.15, 0.12, 0.11, 0.08, 0.33, 0.09, 0.14, 0.17, 0.14, 0.35,
       0.06, 0.15, 0.2 , 0.04, 0.62, 0.2 , 0.42, 0.17, 0.12, 0.19, 0.01,
       0.09, 0.22, 0.04, 0.25, 0.17, 0.03, 0.1 , 0.06, 0.18, 0.09, 0.07,
       0.12, 0.11, 0.29, 0.1 , 0.31, 0.24, 0.15, 0.12, 0.19, 0.14, 0.15,
       0.01, 0.07, 0.06, 0.1 , 0.1 , 0.27, 0.09, 0.1 ])
```

In [115…
```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
y_test = label_encoder.fit_transform(y_test)
```

In [116…
```
# Import the necessary libraries
from sklearn.metrics import roc_curve, roc_auc_score
import matplotlib.pyplot as plt

logreg_roc_auc = roc_auc_score(y_test, lr.predict_proba(x_test)[:, 1])

# Calculate the ROC AUC score for Decision Tree Classifier
dt_roc_auc = roc_auc_score(y_test, grid_search.predict_proba(x_test)[:, 1])

# Compute ROC curve for Logistic Regression
logreg_fpr, logreg_tpr, _ = roc_curve(y_test, lr.predict_proba(x_test)[:, 1])

# Compute ROC curve for Decision Tree Classifier
dt_fpr, dt_tpr, _ = roc_curve(y_test, grid_search.predict_proba(x_test)[:, 1])

# Calculate the ROC AUC score for Random Forest
rf_roc_auc = roc_auc_score(y_test, rfc_cv.predict_proba(x_test)[:, 1])

# Compute ROC curve for Random Forest
rf_fpr, rf_tpr, _ = roc_curve(y_test, rfc_cv.predict_proba(x_test)[:, 1])

# Plot ROC AUC curves
plt.figure(figsize=(10, 6))
plt.plot(logreg_fpr, logreg_tpr, label='Logistic Regression (AUC = %0.2f)' % logreg
plt.plot(dt_fpr, dt_tpr, label='Decision Tree (AUC = %0.2f)' % dt_roc_auc)
plt.plot(rf_fpr, rf_tpr, label='Random Forest (AUC = %0.2f)' % rf_roc_auc)
plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc='lower right')
plt.show()
```



In [ ]: