In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Data Preprocessing

In [2]:

```
df = pd.read_csv("Employee-Attrition.csv")
```

In [3]:

```
df.head()
```

Out[3]:

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

5 rows × 35 columns

In [4]:

```
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
```

```
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64
 24  PerformanceRating         1470 non-null   int64
 25  RelationshipSatisfaction  1470 non-null   int64
 26  StandardHours             1470 non-null   int64
 27  StockOptionLevel          1470 non-null   int64
 28  TotalWorkingYears         1470 non-null   int64
 29  TrainingTimesLastYear     1470 non-null   int64
 30  WorkLifeBalance           1470 non-null   int64
 31  YearsAtCompany            1470 non-null   int64
 32  YearsInCurrentRole        1470 non-null   int64
 33  YearsSinceLastPromotion   1470 non-null   int64
 34  YearsWithCurrManager      1470 non-null   int64
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

In [5]:

```
df.shape
```

Out[5]:

```
(1470, 35)
```

In [6]:

```
df.describe()
```

Out[6]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfactio
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	1470.00000
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	2.72176
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	1.09308
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	1.00000
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	2.00000
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	3.00000
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	4.00000
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	4.00000

8 rows × 26 columns

In [7]:

```
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
```

```
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
```

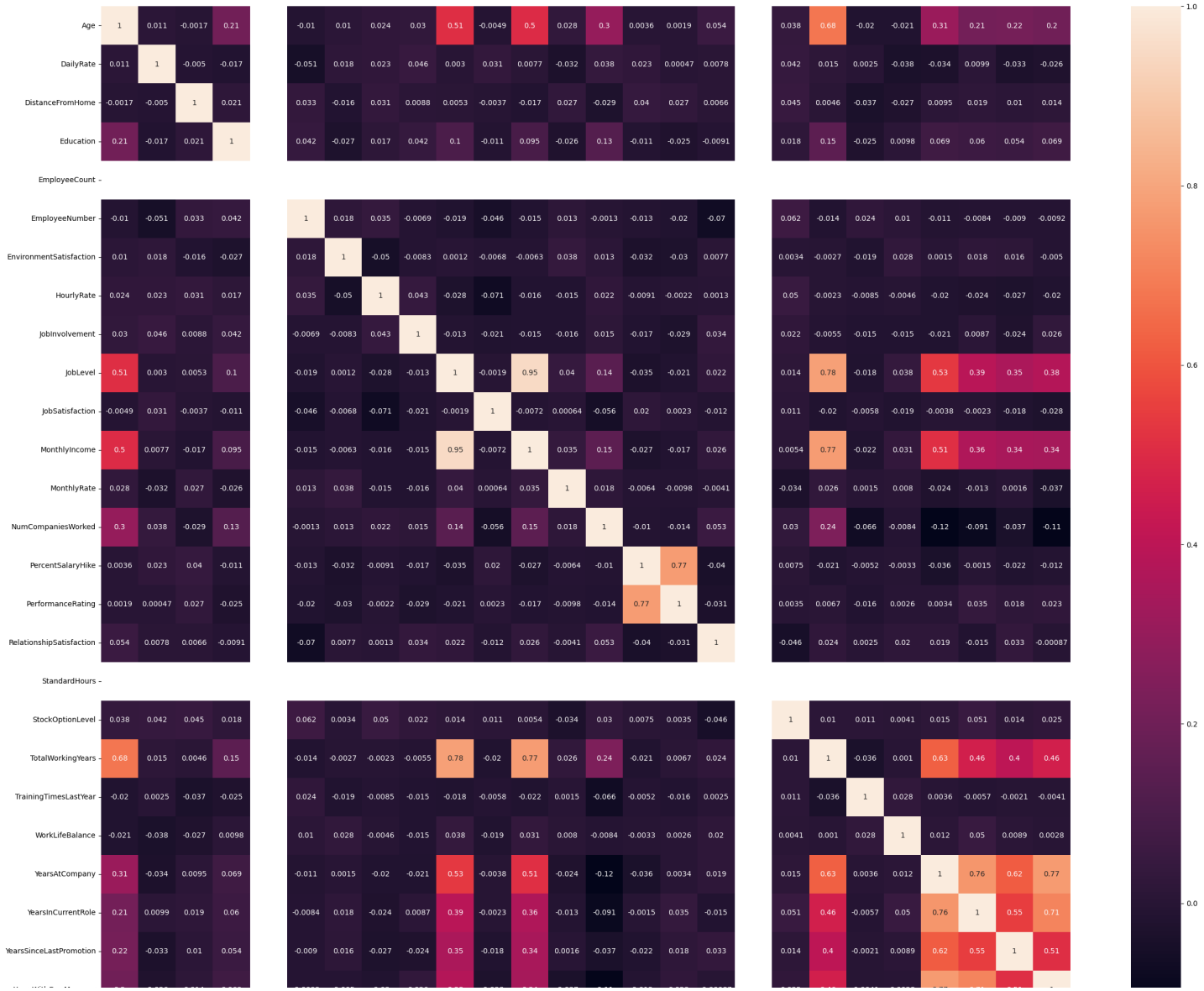
In [8]:

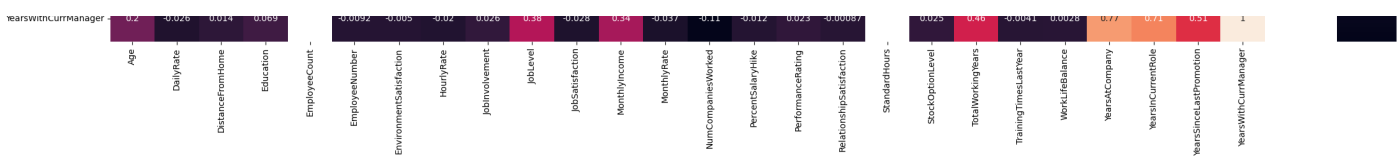
```
plt.subplots(figsize=(30,25))
sns.heatmap(df.corr(),annot=True)
```

```
/var/folders/m8/dg41v9m11bdcfq4q15h80_140000gn/T/ipykernel_13814/974398346.py:2: FutureWa
rning: 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[8]:

```
<Axes: >
```



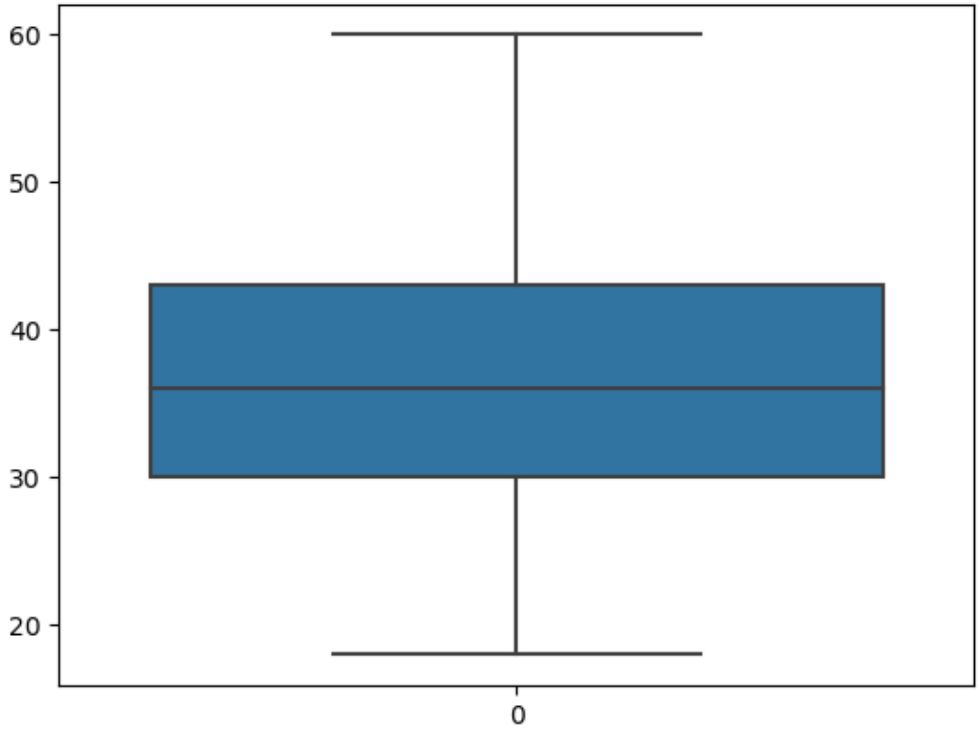


In [9]:

```
sns.boxplot(df['Age'])
```

Out[9]:

<Axes: >

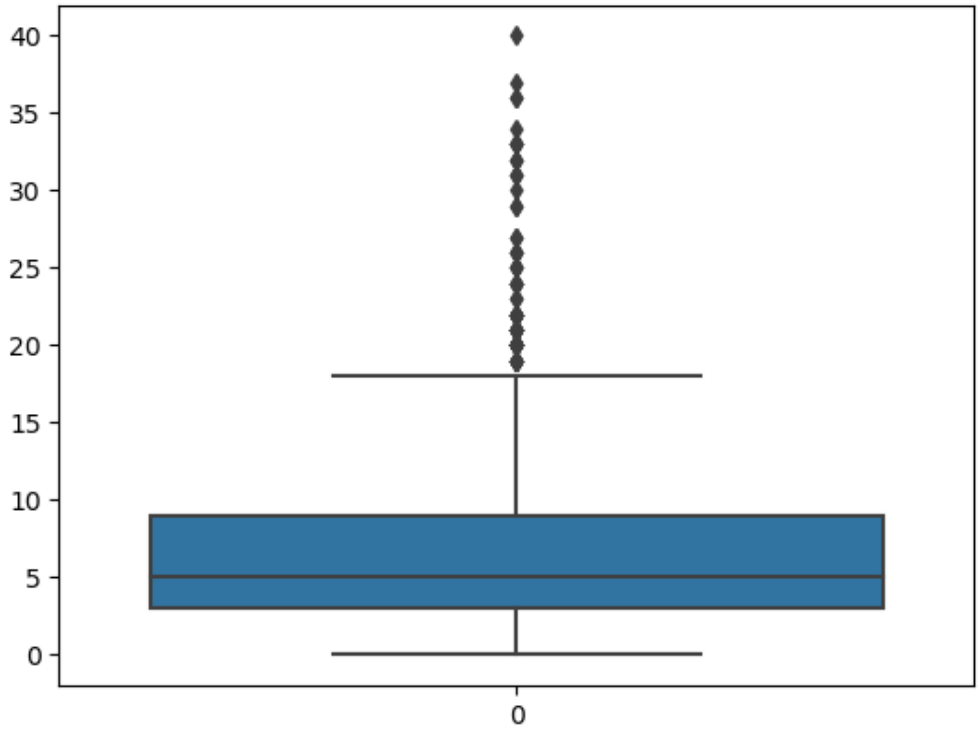


In [10]:

```
sns.boxplot(df.YearsAtCompany)
```

Out[10]:

<Axes: >



In [11]:

```
q1 = df.YearsAtCompany.quantile(0.25)
q3 = df.YearsAtCompany.quantile(0.75)
IQR = q3-q1
upper_limit = q3 + 1.5*IQR
```

In [12]:

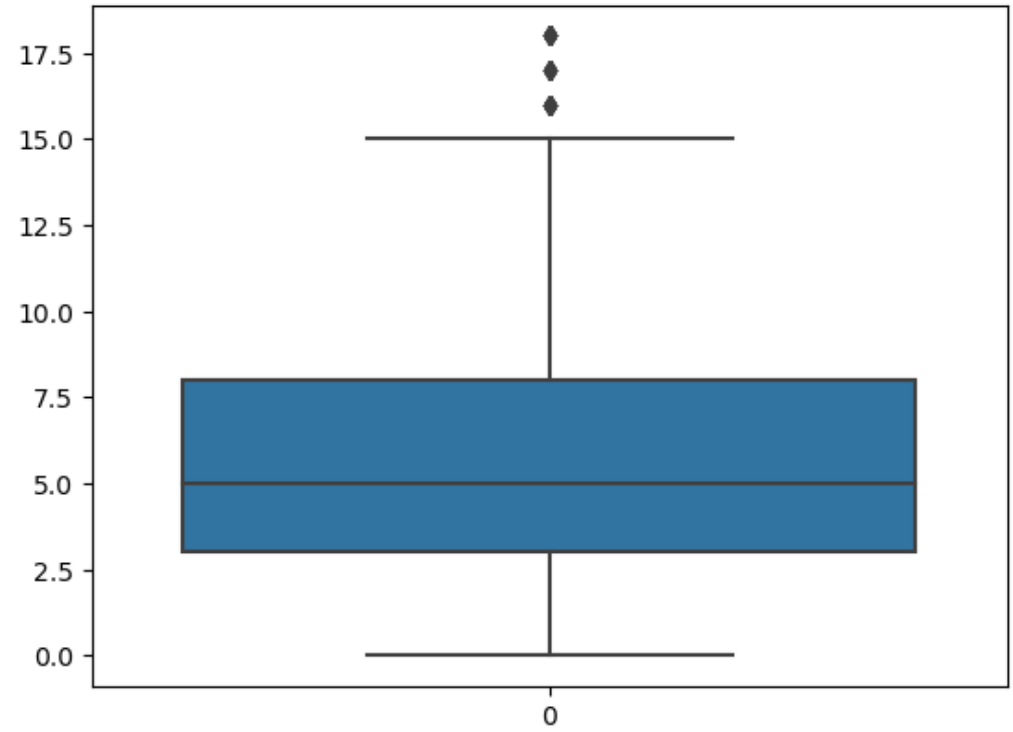
```
df['YearsAtCompany']= np.where(df['YearsAtCompany']>upper_limit,df.YearsAtCompany.median
(),df['YearsAtCompany'])
```

In [13]:

```
sns.boxplot(df.YearsAtCompany)
```

Out[13]:

<Axes: >

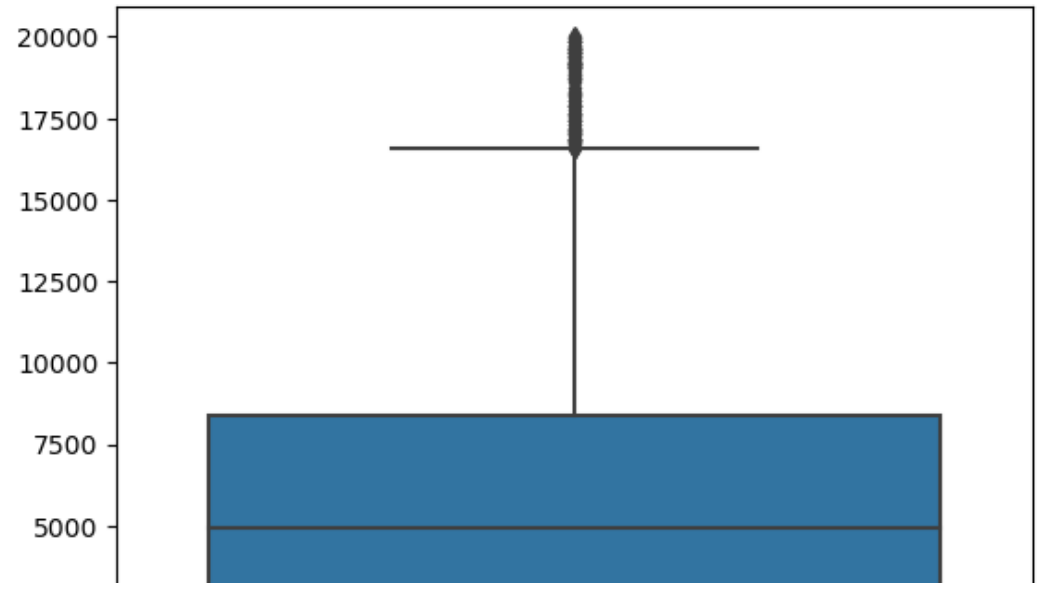


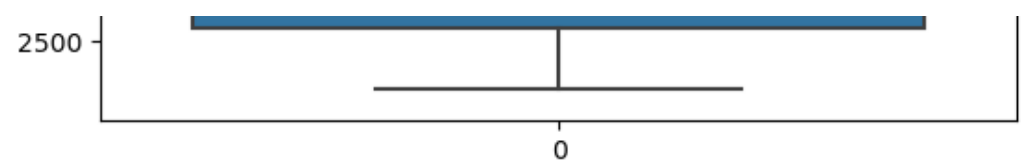
In [14]:

```
sns.boxplot(df.MonthlyIncome)
```

Out[14]:

<Axes: >





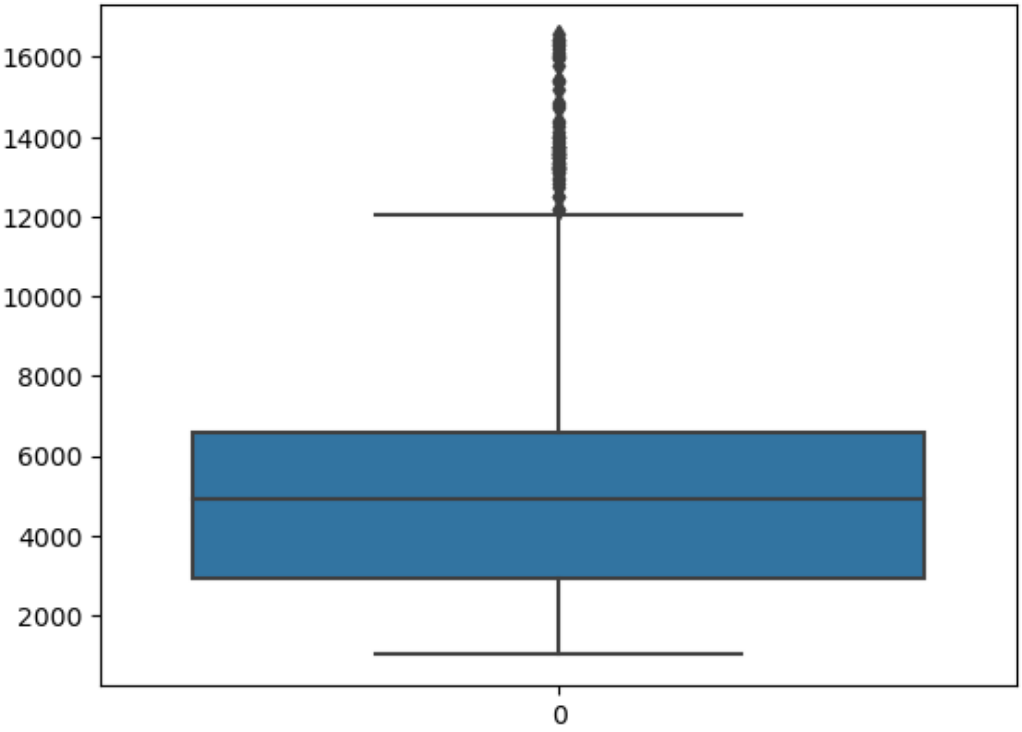
In [15]:

```
q1 = df.MonthlyIncome.quantile(0.25)
q3 = df.MonthlyIncome.quantile(0.75)
IQR = q3-q1
upper_limit = q3 + 1.5*IQR

df["MonthlyIncome"]= np.where(df['MonthlyIncome']>upper_limit,df.MonthlyIncome.median(),
df['MonthlyIncome'])
sns.boxplot(df.MonthlyIncome)
```

Out[15]:

<Axes: >

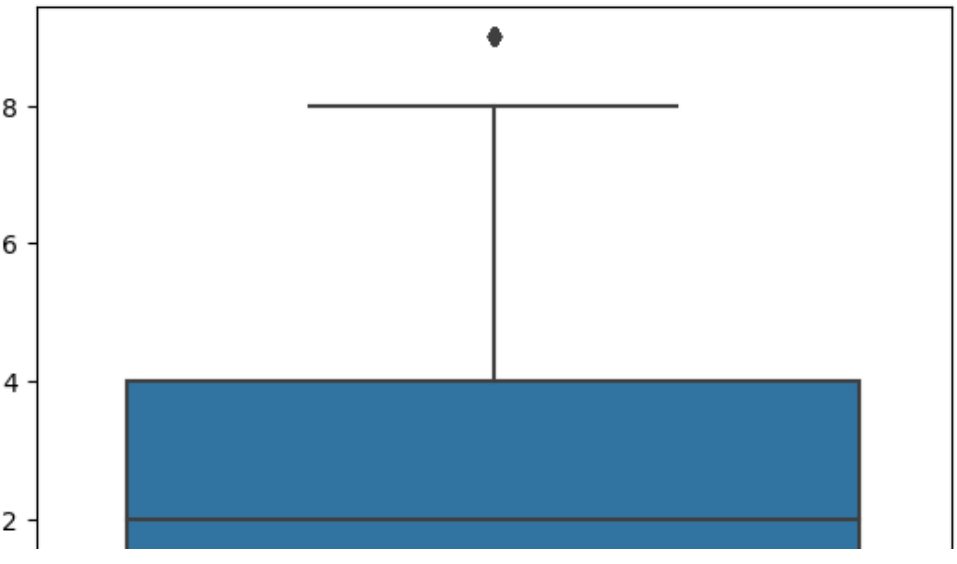


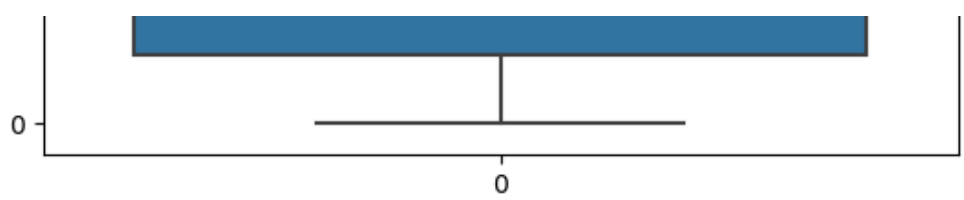
In [16]:

```
sns.boxplot(df.NumCompaniesWorked)
```

Out[16]:

<Axes: >





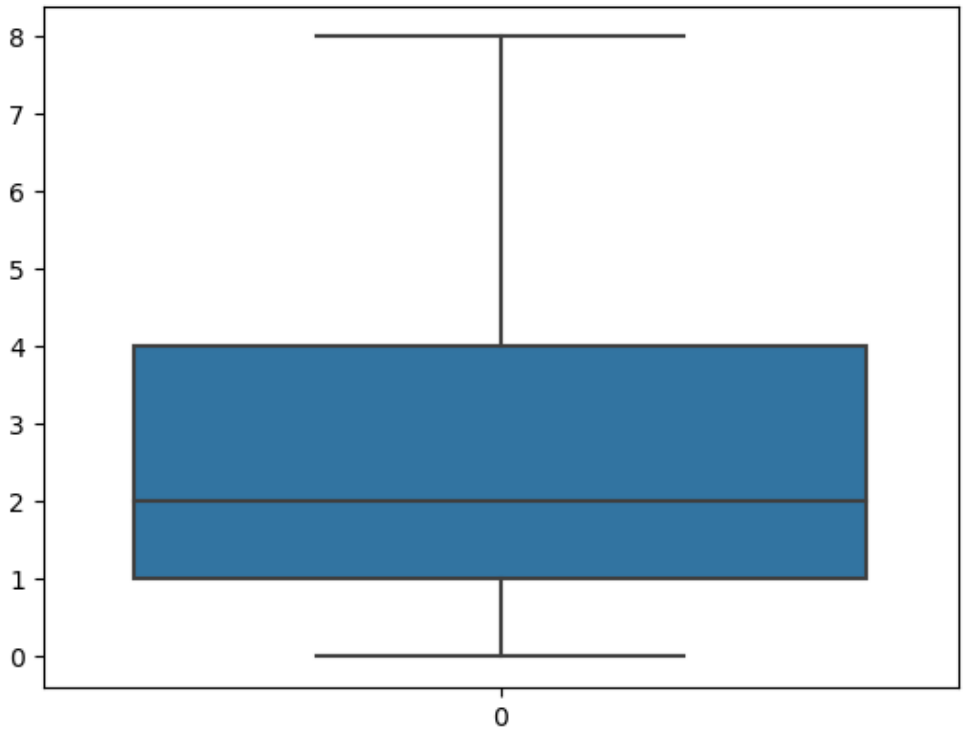
In [17]:

```
q1 = df.NumCompaniesWorked.quantile(0.25)
q3 = df.NumCompaniesWorked.quantile(0.75)
IQR = q3-q1
upper_limit = q3 + 1.5*IQR

df["NumCompaniesWorked"]= np.where(df['NumCompaniesWorked']>upper_limit,df.NumCompaniesW
orked.median(),df['NumCompaniesWorked'])
sns.boxplot(df.NumCompaniesWorked)
```

Out[17]:

<Axes: >



In [18]:

```
df.head()
```

Out[18]:

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

5 rows × 35 columns

In [19]:

```
from sklearn.preprocessing import LabelEncoder
```

In [20]:

```
le = LabelEncoder()
```

In [21]:

```
df.Attrition = le.fit_transform(df.Attrition)
```

In [22]:

```
df.BusinessTravel = le.fit_transform(df.BusinessTravel)
```

In [23]:

```
df.Department = le.fit_transform(df.Department)
```

In [24]:

```
df.Gender = le.fit_transform(df.Gender)
```

In [25]:

```
df.MaritalStatus = le.fit_transform(df.MaritalStatus)
```

In [26]:

```
df.EducationField = le.fit_transform(df.EducationField)
```

In [27]:

```
df.JobRole = le.fit_transform(df.JobRole)
```

In [28]:

```
df.Over18 = le.fit_transform(df.Over18)
```

In [29]:

```
df.OverTime = le.fit_transform(df.OverTime)
```

In [30]:

```
df.head()
```

Out[30]:

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

5 rows × 35 columns

In [31]:

```
y = df['Attrition']
x = df.drop(['Attrition','EmployeeNumber','EmployeeCount','StockOptionLevel','Over18','StandardHours'], axis = 1)
```

In [32]:

```
x.head()
```

Out[32]:

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

5 rows × 29 columns

In [33]:

```
y.head()
```

Out[33]:

```
0    1
1    0
2    1
3    0
4    0
Name: Attrition, dtype: int64
```

In [34]:

```
from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
x_scaled=pd.DataFrame(ms.fit_transform(x),columns=x.columns)
```

In [35]:

```
x_scaled
```

Out[35]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfactio
0	0.547619	1.0	0.715820	1.0	0.000000	0.25	0.2	0.33333
1	0.738095	0.5	0.126700	0.5	0.250000	0.00	0.2	0.66666
2	0.452381	1.0	0.909807	0.5	0.035714	0.25	0.8	1.00000
3	0.357143	0.5	0.923407	0.5	0.071429	0.75	0.2	1.00000
4	0.214286	1.0	0.350036	0.5	0.035714	0.00	0.6	0.00000
...	...	...	...	...	...	...	...	...
1465	0.428571	0.5	0.559771	0.5	0.785714	0.25	0.6	0.66666
1466	0.500000	1.0	0.365784	0.5	0.178571	0.00	0.6	1.00000
1467	0.214286	1.0	0.037938	0.5	0.107143	0.50	0.2	0.33333
1468	0.738095	0.5	0.659270	1.0	0.035714	0.50	0.6	1.00000
1469	0.380952	1.0	0.376521	0.5	0.250000	0.50	0.6	0.33333

1470 rows × 29 columns

In [36]:

```
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_state=0)
```

In [37]:

```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[37]:

```
((1176, 29), (294, 29), (1176,), (294,))
```

In [38]:

```
x_train.head()
```

Out[38]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfactio
1374	0.952381	1.0	0.360057	1.0	0.714286	0.50	0.2	1.00000
1092	0.642857	1.0	0.607015	0.5	0.964286	0.50	1.0	1.00000
768	0.523810	1.0	0.141732	1.0	0.892857	0.50	0.4	0.66666
569	0.428571	0.0	0.953472	1.0	0.250000	0.75	0.2	0.00000
911	0.166667	0.5	0.355762	1.0	0.821429	0.00	0.2	0.66666

5 rows × 29 columns

Logistic Regression

In [39]:

```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
```

In [40]:

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

Out[40]:

```
▾ LogisticRegression
LogisticRegression()
```

In [41]:

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

In [42]:

```
pred
```

Out[42]:

```
array([0, 0, 0, 0, 1, 0, 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, 1, 0, 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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 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,
       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, 0, 0, 0, 1, 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, 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, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 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, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0])
```

In [43]:

```
y_test
```

Out[43]:

```
442     0
1091    0
981     1
785     0
1332    1
       ..
1439    0
481     0
124     1
198     0
1229    0
Name: Attrition, Length: 294, dtype: int64
```

In [44]:

```
x.head()
```

Out[44]:

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

5 rows × 29 columns

In [45]:

```
x.iloc[1].values
```

Out[45]:

```
array([4.9000e+01, 1.0000e+00, 2.7900e+02, 1.0000e+00, 8.0000e+00,
       1.0000e+00, 1.0000e+00, 3.0000e+00, 1.0000e+00, 6.1000e+01,
       2.0000e+00, 2.0000e+00, 6.0000e+00, 2.0000e+00, 1.0000e+00,
       5.1300e+03, 2.4907e+04, 1.0000e+00, 0.0000e+00, 2.3000e+01,
       4.0000e+00, 4.0000e+00, 1.0000e+01, 3.0000e+00, 3.0000e+00,
       1.0000e+01, 7.0000e+00, 1.0000e+00, 7.0000e+00])
```

In [46]:

```
model.predict(ms.transform([x.iloc[1].values]))
```

```
/Users/samarthgayakhe/anaconda3/lib/python3.11/site-packages/sklearn/base.py:464: UserWar
ning: X does not have valid feature names, but MinMaxScaler was fitted with feature names
  warnings.warn(
/Users/samarthgayakhe/anaconda3/lib/python3.11/site-packages/sklearn/base.py:464: UserWar
ning: X does not have valid feature names, but LogisticRegression was fitted with feature
names
  warnings.warn(
```

Out[46]:

```
array([0])
```

In [47]:

```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,roc_auc
```

```
_score,roc_curve
```

In [48]:

```
accuracy_score(y_test,pred)
```

Out[48]:

```
0.8809523809523809
```

In [49]:

```
confusion_matrix(y_test,pred)
```

Out[49]:

```
array([[241,   4],
       [ 31,  18]])
```

In [50]:

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

In [51]:

```
probability
```

Out[51]:

```
array([0.13705292, 0.20413369, 0.34312874, 0.07768427, 0.66643466,
       0.06220359, 0.5657188 , 0.06097656, 0.00589668, 0.29760658,
       0.05218131, 0.33150501, 0.01930216, 0.66046911, 0.20259666,
       0.03005587, 0.11097207, 0.18535186, 0.04135716, 0.20903188,
       0.25383482, 0.01786367, 0.06561924, 0.05861195, 0.55095385,
       0.37187294, 0.07564371, 0.03247636, 0.71615233, 0.05904737,
       0.01545625, 0.03041236, 0.06922794, 0.16299079, 0.06728342,
       0.02858353, 0.10391573, 0.07447643, 0.03357148, 0.06734202,
       0.08106547, 0.01783665, 0.01638939, 0.01172776, 0.02186822,
       0.52454019, 0.24045964, 0.00391247, 0.7614898 , 0.49747348,
       0.11845236, 0.4877299 , 0.06633316, 0.23092007, 0.71015397,
       0.27329606, 0.02679241, 0.31896298, 0.02702889, 0.17983703,
       0.02435044, 0.19792797, 0.15744753, 0.0325198 , 0.39140371,
       0.03924587, 0.2467184 , 0.12877425, 0.09677819, 0.09966012,
       0.0710539 , 0.35568251, 0.06245966, 0.11664995, 0.05931661,
       0.10012556, 0.07016361, 0.08541958, 0.1980308 , 0.03484952,
       0.0117672 , 0.03290184, 0.16023984, 0.02577893, 0.02908716,
       0.0895246 , 0.01071049, 0.04192113, 0.03542206, 0.17725704,
       0.32063684, 0.18447531, 0.33528566, 0.26673461, 0.02292565,
       0.19098532, 0.32407706, 0.25671874, 0.08291004, 0.05087582,
       0.18591817, 0.68001623, 0.37611187, 0.02103873, 0.08855535,
       0.02567966, 0.05607746, 0.15674119, 0.07000229, 0.148906  ,
       0.08218393, 0.06032917, 0.02365155, 0.14958914, 0.06513179,
       0.03256922, 0.04471741, 0.12649693, 0.01097451, 0.0120382 ,
       0.15097344, 0.0578532 , 0.07719942, 0.79309637, 0.03441276,
       0.02112975, 0.01063343, 0.13517988, 0.16270342, 0.04795822,
       0.01637024, 0.28084147, 0.52210744, 0.3112341 , 0.04024336,
       0.42218968, 0.57519886, 0.15057287, 0.11524213, 0.32642558,
       0.12822559, 0.07806723, 0.1213583 , 0.1699777 , 0.20398287,
       0.03043194, 0.19576413, 0.0081222 , 0.07026465, 0.16829778,
       0.06494135, 0.21496007, 0.07008288, 0.20581403, 0.04865849,
       0.02403687, 0.10192445, 0.07753706, 0.02357914, 0.01133108,
       0.48738476, 0.0140777 , 0.13177434, 0.82052436, 0.10430838,
       0.28765256, 0.14207827, 0.14405031, 0.02907779, 0.01065587,
       0.03986393, 0.07977763, 0.11917718, 0.10459444, 0.03259684,
       0.15499041, 0.10599853, 0.09778676, 0.06617238, 0.1567461 ,
       0.02632536, 0.089076  , 0.01164187, 0.78349927, 0.03929854,
       0.04849599, 0.37606476, 0.03990634, 0.68495648, 0.10669824,
       0.40939502, 0.37482294, 0.3030796 , 0.05776824, 0.10576604,
       0.16225437, 0.04360157, 0.01400713, 0.26482602, 0.05701699,
       0.12907133, 0.17305038, 0.64186631, 0.08458952, 0.23212631,
       0.03991026, 0.52365163, 0.00434289, 0.13438826, 0.02738351,
       0.12268298, 0.22089373, 0.06384565, 0.11454498, 0.12594275,
       0.03132331, 0.02995546, 0.07095559, 0.04704303, 0.14444555,
```

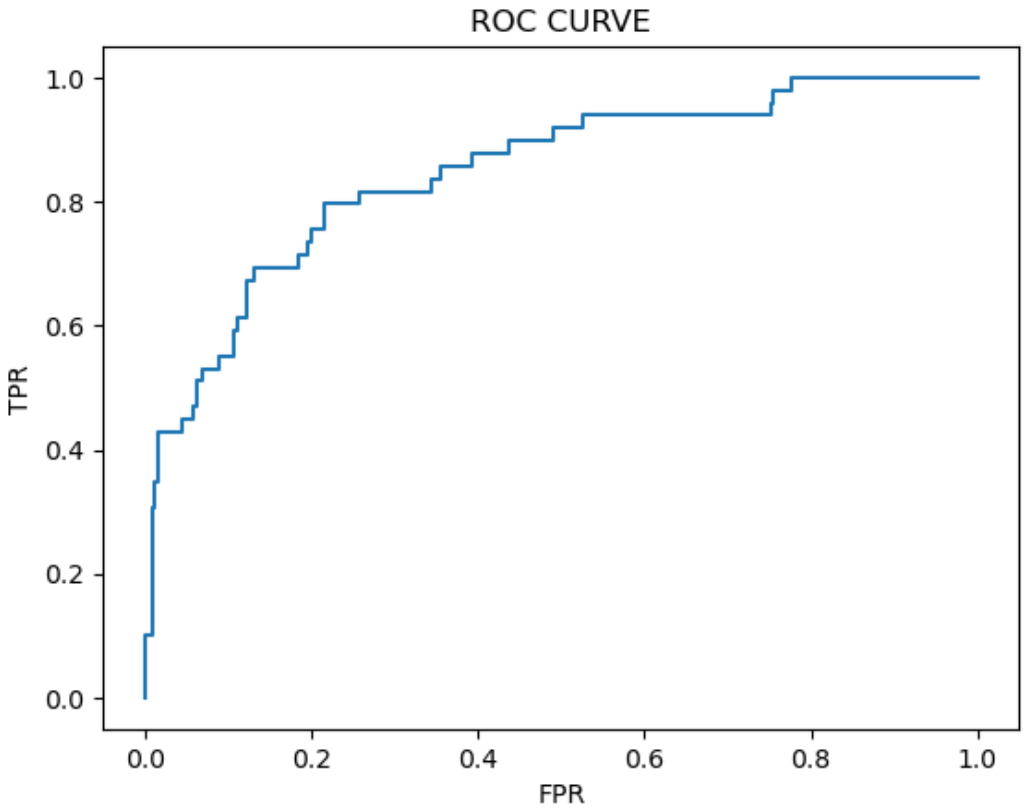
```
0.11151227, 0.21993793, 0.71471845, 0.126911  , 0.33878135,
0.01720784, 0.1154823 , 0.21502212, 0.36410483, 0.04430082,
0.06309539, 0.30896393, 0.06042337, 0.01611364, 0.18042562,
0.37449008, 0.23488451, 0.00726084, 0.10123849, 0.01011296,
0.19766521, 0.22715129, 0.01142139, 0.19065821, 0.05422612,
0.03861509, 0.40582042, 0.36715018, 0.04890088, 0.11701719,
0.32173631, 0.32338326, 0.77410265, 0.06838264, 0.17169693,
0.09668869, 0.00819949, 0.67071277, 0.21762394, 0.35101999,
0.3832691 , 0.04171714, 0.16694179, 0.06872956, 0.06361792,
0.09241628, 0.01017511, 0.25901001, 0.54023218, 0.07632533,
0.09318032, 0.01157029, 0.11949442, 0.04581067, 0.02004951,
0.03026311, 0.06824873, 0.26601795, 0.14457213, 0.21911835,
0.25847947, 0.01911555, 0.16507675, 0.13117057, 0.04600318,
0.18693986, 0.00866713, 0.29095561, 0.0044326 , 0.03188011,
0.23017337, 0.67589739, 0.02953884, 0.31361769])
```

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 [55]:

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

Out[55]:

```
▾ DecisionTreeClassifier
```

```
DecisionTreeClassifier()
```

In [56]:

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

In [57]:

```
pred
```

Out[57]:

```
array([0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 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, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0])
```

In [58]:

```
y_test
```

Out[58]:

```
442     0
1091    0
981     1
785     0
1332    1
       ..
1439    0
481     0
124     1
198     0
1229    0
Name: Attrition, Length: 294, dtype: int64
```

In [59]:

```
dtc.predict(ms.transform([x.iloc[1].values]))
```

```
/Users/samarthgayakhe/anaconda3/lib/python3.11/site-packages/sklearn/base.py:464: UserWar
ning: X does not have valid feature names, but MinMaxScaler was fitted with feature names
  warnings.warn(
/Users/samarthgayakhe/anaconda3/lib/python3.11/site-packages/sklearn/base.py:464: UserWar
ning: X does not have valid feature names, but DecisionTreeClassifier was fitted with fea
ture names
  warnings.warn(
```

Out[59]:

```
array([0])
```

In [60]:

```
accuracy_score(y_test,pred)
```

Out[60]:

```
0.7687074829931972
```

In [61]:

```
confusion_matrix(y_test,pred)
```

Out[61]:

```
array([[206,  39],
       [ 29,  20]])
```

In [62]:

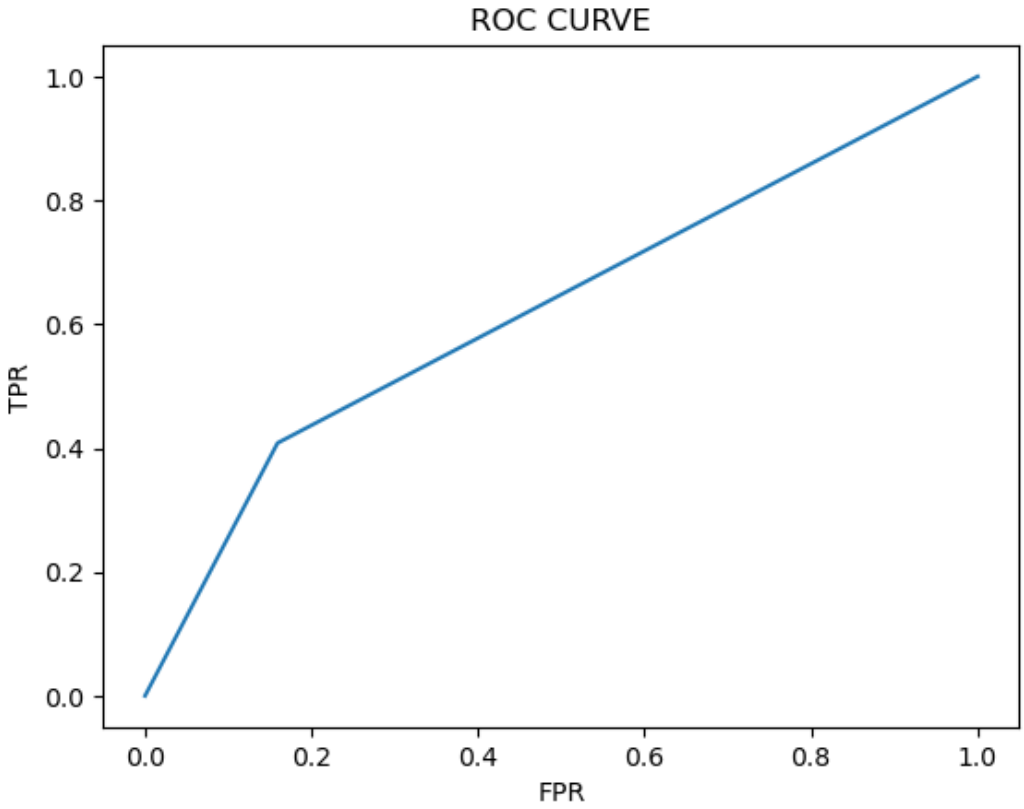
```
probability=dtc.predict_proba(x_test)[:,1]
```

In [63]:

```
fpr,tpr,threshsholds = roc_curve(y_test,probability)
```

In [64]:

```
plt.plot(fpr,tpr)
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('ROC CURVE')
plt.show()
```



In [65]:

```
from sklearn import tree
plt.figure(figsize=(25,15))
tree.plot_tree(dtc,filled=True)
```

Out[65]:

```
[Text(0.31961171096345514, 0.96875, 'x[22] <= 0.038\ngini = 0.269\nsamples = 1176\nvalue
= [988, 188]'),
 Text(0.07308970099667775, 0.90625, 'x[14] <= 0.75\ngini = 0.5\nsamples = 78\nvalue = [39
, 39]'),
 Text(0.04318936877076412, 0.84375, 'x[4] <= 0.554\ngini = 0.426\nsamples = 39\nvalue = [
27, 12]'),
 Text(0.026578073089700997, 0.78125, 'x[13] <= 0.167\ngini = 0.312\nsamples = 31\nvalue =
[25, 6]'),
 Text(0.013289036544850499, 0.71875, 'x[15] <= 0.07\ngini = 0.49\nsamples = 7\nvalue = [3
, 4]'),
 Text(0.006644518272425249, 0.65625, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.019933554817275746, 0.65625, 'x[7] <= 0.5\ngini = 0.375\nsamples = 4\nvalue = [3,
1]'),
 Text(0.013289036544850499, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
```

Text(0.015289056544850499, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.026578073089700997, 0.59375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
Text(0.03986710963455149, 0.71875, 'x[17] <= 0.062\ngini = 0.153\nsamples = 24\nvalue = [22, 2]'),
Text(0.03322259136212625, 0.65625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.046511627906976744, 0.65625, 'x[7] <= 0.167\ngini = 0.083\nsamples = 23\nvalue = [22, 1]'),
Text(0.03986710963455149, 0.59375, 'x[12] <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.03322259136212625, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.046511627906976744, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.053156146179401995, 0.59375, 'gini = 0.0\nsamples = 21\nvalue = [21, 0]'),
Text(0.059800664451827246, 0.78125, 'x[19] <= 0.679\ngini = 0.375\nsamples = 8\nvalue = [2, 6]'),
Text(0.053156146179401995, 0.71875, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
Text(0.0664451827242525, 0.71875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
Text(0.10299003322259136, 0.84375, 'x[9] <= 0.364\ngini = 0.426\nsamples = 39\nvalue = [12, 27]'),
Text(0.08637873754152824, 0.78125, 'x[15] <= 0.122\ngini = 0.133\nsamples = 14\nvalue = [1, 13]'),
Text(0.07973421926910298, 0.71875, 'gini = 0.0\nsamples = 13\nvalue = [0, 13]'),
Text(0.09302325581395349, 0.71875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.11960132890365449, 0.78125, 'x[18] <= 0.5\ngini = 0.493\nsamples = 25\nvalue = [11, 14]'),
Text(0.10631229235880399, 0.71875, 'x[6] <= 0.7\ngini = 0.484\nsamples = 17\nvalue = [10, 7]'),
Text(0.09966777408637874, 0.65625, 'x[2] <= 0.106\ngini = 0.408\nsamples = 14\nvalue = [10, 4]'),
Text(0.09302325581395349, 0.59375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.10631229235880399, 0.59375, 'x[24] <= 0.5\ngini = 0.278\nsamples = 12\nvalue = [10, 2]'),
Text(0.09966777408637874, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.11295681063122924, 0.53125, 'x[2] <= 0.8\ngini = 0.165\nsamples = 11\nvalue = [10, 1]'),
Text(0.10631229235880399, 0.46875, 'gini = 0.0\nsamples = 10\nvalue = [10, 0]'),
Text(0.11960132890365449, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.11295681063122924, 0.65625, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
Text(0.132890365448505, 0.71875, 'x[9] <= 0.45\ngini = 0.219\nsamples = 8\nvalue = [1, 7]'),
Text(0.12624584717607973, 0.65625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.13953488372093023, 0.65625, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]'),
Text(0.5661337209302325, 0.90625, 'x[18] <= 0.5\ngini = 0.235\nsamples = 1098\nvalue = [949, 149]'),
Text(0.31634136212624586, 0.84375, 'x[24] <= 0.167\ngini = 0.162\nsamples = 798\nvalue = [727, 71]'),
Text(0.18272425249169436, 0.78125, 'x[2] <= 0.747\ngini = 0.38\nsamples = 47\nvalue = [35, 12]'),
Text(0.1760797342192691, 0.71875, 'x[10] <= 0.5\ngini = 0.463\nsamples = 33\nvalue = [21, 12]'),
Text(0.15282392026578073, 0.65625, 'x[4] <= 0.446\ngini = 0.42\nsamples = 10\nvalue = [3, 7]'),
Text(0.1461794019933555, 0.59375, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
Text(0.15946843853820597, 0.59375, 'x[4] <= 0.714\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),
Text(0.15282392026578073, 0.53125, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
Text(0.16611295681063123, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.19933554817275748, 0.65625, 'x[23] <= 0.583\ngini = 0.34\nsamples = 23\nvalue = [18, 5]'),
Text(0.18604651162790697, 0.59375, 'x[9] <= 0.107\ngini = 0.117\nsamples = 16\nvalue = [15, 1]'),
Text(0.17940199335548174, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.19269102990033224, 0.53125, 'gini = 0.0\nsamples = 15\nvalue = [15, 0]'),
Text(0.21262458471760798, 0.59375, 'x[28] <= 0.529\ngini = 0.49\nsamples = 7\nvalue = [3, 4]'),
Text(0.2059800664451827, 0.53125, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
Text(0.21926910299003322, 0.53125, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
Text(0.18936877076411196, 0.71875, 'gini = 0.0\nsamples = 14\nvalue = [14, 0]'),
Text(0.44995847176079734, 0.78125, 'x[22] <= 0.975\ngini = 0.145\nsamples = 751\nvalue = [692, 59]'),
Text(0.4433139534883721, 0.71875, 'x[25] <= 0.25\ngini = 0.143\nsamples = 750\nvalue = [692, 58]'),
Text(0.3122923588039867, 0.65625, 'x[7] <= 0.167\ngini = 0.218\nsamples = 257\nvalue = [225, 32]')

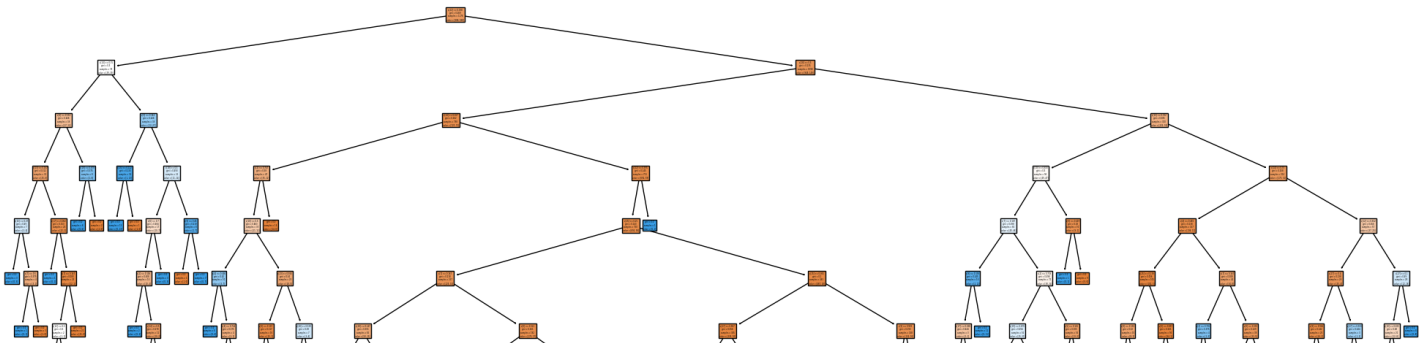
```
225, 32]'),
 Text(0.2541528239202658, 0.59375, 'x[28] <= 0.147\ngini = 0.355\nsamples = 65\nvalue = [
50, 15]'),
 Text(0.23255813953488372, 0.53125, 'x[28] <= 0.029\ngini = 0.303\nsamples = 59\nvalue =
[48, 11]'),
 Text(0.20930232558139536, 0.46875, 'x[10] <= 0.5\ngini = 0.463\nsamples = 22\nvalue = [1
4, 8]'),
 Text(0.19601328903654486, 0.40625, 'x[9] <= 0.179\ngini = 0.198\nsamples = 9\nvalue = [8
, 1]'),
 Text(0.1893687707641196, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.2026578073089701, 0.34375, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(0.22259136212624583, 0.40625, 'x[9] <= 0.4\ngini = 0.497\nsamples = 13\nvalue = [6,
7]'),
 Text(0.2159468438538206, 0.34375, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.2292358803986711, 0.34375, 'x[4] <= 0.286\ngini = 0.346\nsamples = 9\nvalue = [2,
7]'),
 Text(0.22259136212624583, 0.28125, 'x[9] <= 0.629\ngini = 0.444\nsamples = 3\nvalue = [2
, 1]'),
 Text(0.2159468438538206, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.2292358803986711, 0.21875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.23588039867109634, 0.28125, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.2558139534883721, 0.46875, 'x[13] <= 0.167\ngini = 0.149\nsamples = 37\nvalue = [
34, 3]'),
 Text(0.24916943521594684, 0.40625, 'x[24] <= 0.5\ngini = 0.5\nsamples = 6\nvalue = [3, 3
]'),
 Text(0.2425249169435216, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.2558139534883721, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.26245847176079734, 0.40625, 'gini = 0.0\nsamples = 31\nvalue = [31, 0]'),
 Text(0.2757475083056478, 0.53125, 'x[21] <= 0.667\ngini = 0.444\nsamples = 6\nvalue = [2
, 4]'),
 Text(0.2691029900332226, 0.46875, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.2823920265780731, 0.46875, 'x[24] <= 0.5\ngini = 0.444\nsamples = 3\nvalue = [2,
1]'),
 Text(0.2757475083056478, 0.40625, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.28903654485049834, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3704318936877076, 0.59375, 'x[0] <= 0.321\ngini = 0.161\nsamples = 192\nvalue = [
175, 17]'),
 Text(0.31893687707641194, 0.53125, 'x[6] <= 0.1\ngini = 0.294\nsamples = 67\nvalue = [55
, 12]'),
 Text(0.3122923588039867, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.32558139534883723, 0.46875, 'x[24] <= 0.5\ngini = 0.26\nsamples = 65\nvalue = [55
, 10]'),
 Text(0.30232558139534885, 0.40625, 'x[6] <= 0.5\ngini = 0.469\nsamples = 16\nvalue = [10,
6]'),
 Text(0.2956810631229236, 0.34375, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(0.3089700996677741, 0.34375, 'x[7] <= 0.833\ngini = 0.444\nsamples = 9\nvalue = [3,
6]'),
 Text(0.30232558139534885, 0.28125, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(0.31561461794019935, 0.28125, 'x[17] <= 0.5\ngini = 0.375\nsamples = 4\nvalue = [3,
1]'),
 Text(0.3089700996677741, 0.21875, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.3222591362126246, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3488372093023256, 0.40625, 'x[2] <= 0.037\ngini = 0.15\nsamples = 49\nvalue = [45
, 4]'),
 Text(0.34219269102990035, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3554817275747508, 0.34375, 'x[2] <= 0.938\ngini = 0.117\nsamples = 48\nvalue = [4
5, 3]'),
 Text(0.3488372093023256, 0.28125, 'x[5] <= 0.875\ngini = 0.081\nsamples = 47\nvalue = [4
5, 2]'),
 Text(0.33554817275747506, 0.21875, 'x[10] <= 0.167\ngini = 0.043\nsamples = 45\nvalue =
[44, 1]'),
 Text(0.3289036544850498, 0.15625, 'x[27] <= 0.067\ngini = 0.444\nsamples = 3\nvalue = [2
, 1]'),
 Text(0.3222591362126246, 0.09375, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.33554817275747506, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.34219269102990035, 0.15625, 'gini = 0.0\nsamples = 42\nvalue = [42, 0]'),
 Text(0.36212624584717606, 0.21875, 'x[8] <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]
'),
 Text(0.3554817275747508, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3687707641196013, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.36212624584717606, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.4219269102990033, 0.53125, 'x[6] <= 0.9\ngini = 0.077\nsamples = 125\nvalue = [12
0, 5]')
```

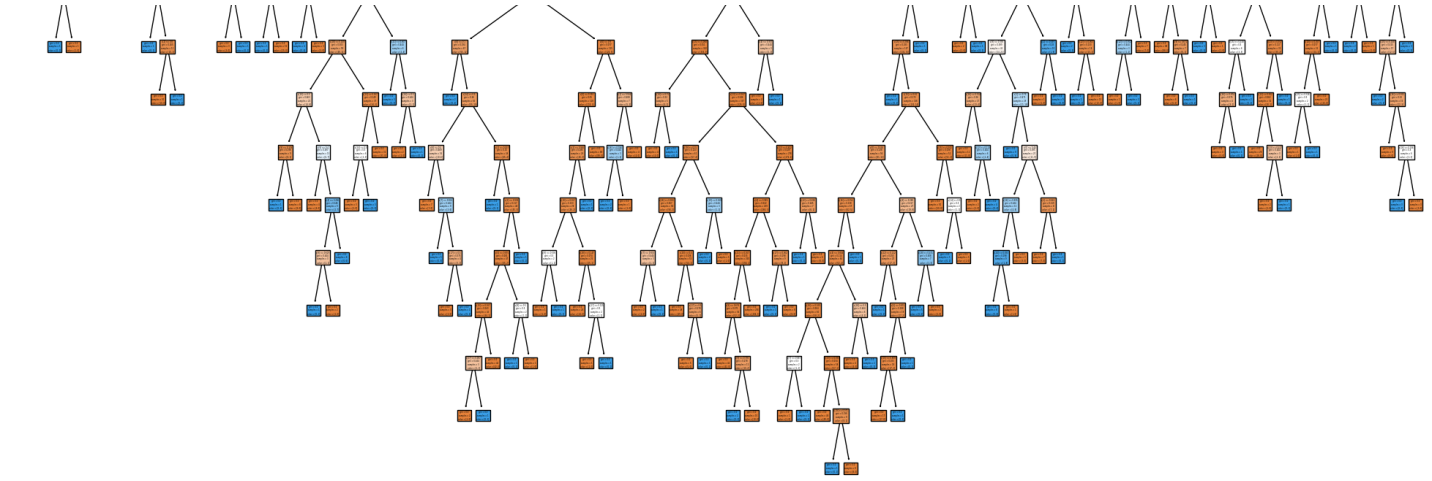
0, 3]'),
 Text(0.40863787375415284, 0.46875, 'x[0] <= 0.393\ngini = 0.05\nsamples = 118\nvalue = [
115, 3]'),
 Text(0.4019933554817276, 0.40625, 'x[2] <= 0.956\ngini = 0.185\nsamples = 29\nvalue = [2
6, 3]'),
 Text(0.3953488372093023, 0.34375, 'x[10] <= 0.167\ngini = 0.133\nsamples = 28\nvalue = [
26, 2]'),
 Text(0.38205980066445183, 0.28125, 'x[9] <= 0.036\ngini = 0.5\nsamples = 2\nvalue = [1,
1]'),
 Text(0.3754152823920266, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.38870431893687707, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.40863787375415284, 0.28125, 'x[28] <= 0.147\ngini = 0.074\nsamples = 26\nvalue =
[25, 1]'),
 Text(0.4019933554817276, 0.21875, 'gini = 0.0\nsamples = 24\nvalue = [24, 0]'),
 Text(0.4152823920265781, 0.21875, 'x[17] <= 0.125\ngini = 0.5\nsamples = 2\nvalue = [1,
1]'),
 Text(0.40863787375415284, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.4219269102990033, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.40863787375415284, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.4152823920265781, 0.40625, 'gini = 0.0\nsamples = 89\nvalue = [89, 0]'),
 Text(0.43521594684385384, 0.46875, 'x[2] <= 0.594\ngini = 0.408\nsamples = 7\nvalue = [5
, 2]'),
 Text(0.42857142857142855, 0.40625, 'x[26] <= 0.056\ngini = 0.444\nsamples = 3\nvalue = [
1, 2]'),
 Text(0.4219269102990033, 0.34375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.43521594684385384, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.4418604651162791, 0.40625, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.5743355481727574, 0.65625, 'x[15] <= 0.365\ngini = 0.1\nsamples = 493\nvalue = [4
67, 26]'),
 Text(0.5116279069767442, 0.59375, 'x[27] <= 0.967\ngini = 0.056\nsamples = 345\nvalue =
[335, 10]'),
 Text(0.4883720930232558, 0.53125, 'x[15] <= 0.068\ngini = 0.051\nsamples = 342\nvalue =
[333, 9]'),
 Text(0.46179401993355484, 0.46875, 'x[14] <= 0.75\ngini = 0.32\nsamples = 5\nvalue = [4,
1]'),
 Text(0.45514950166112955, 0.40625, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.4684385382059801, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5149501661129569, 0.46875, 'x[13] <= 0.167\ngini = 0.046\nsamples = 337\nvalue =
[329, 8]'),
 Text(0.48172757475083056, 0.40625, 'x[19] <= 0.75\ngini = 0.123\nsamples = 61\nvalue = [
57, 4]'),
 Text(0.46511627906976744, 0.34375, 'x[26] <= 0.083\ngini = 0.067\nsamples = 58\nvalue =
[56, 2]'),
 Text(0.45182724252491696, 0.28125, 'x[7] <= 0.833\ngini = 0.444\nsamples = 3\nvalue = [2
, 1]'),
 Text(0.44518272425249167, 0.21875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.4584717607973422, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.47840531561461797, 0.28125, 'x[20] <= 0.5\ngini = 0.036\nsamples = 55\nvalue = [5
4, 1]'),
 Text(0.4717607973421927, 0.21875, 'gini = 0.0\nsamples = 49\nvalue = [49, 0]'),
 Text(0.4850498338870432, 0.21875, 'x[17] <= 0.5\ngini = 0.278\nsamples = 6\nvalue = [5,
1]'),
 Text(0.47840531561461797, 0.15625, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.49169435215946844, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.4983388704318937, 0.34375, 'x[19] <= 0.929\ngini = 0.444\nsamples = 3\nvalue = [1
, 2]'),
 Text(0.49169435215946844, 0.28125, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.5049833887043189, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5481727574750831, 0.40625, 'x[22] <= 0.812\ngini = 0.029\nsamples = 276\nvalue =
[272, 4]'),
 Text(0.53156146179402, 0.34375, 'x[4] <= 0.982\ngini = 0.022\nsamples = 269\nvalue = [26
6, 3]'),
 Text(0.5182724252491694, 0.28125, 'x[21] <= 0.167\ngini = 0.015\nsamples = 260\nvalue =
[258, 2]'),
 Text(0.5116279069767442, 0.21875, 'x[9] <= 0.679\ngini = 0.071\nsamples = 54\nvalue = [5
2, 2]'),
 Text(0.5049833887043189, 0.15625, 'gini = 0.0\nsamples = 42\nvalue = [42, 0]'),
 Text(0.5182724252491694, 0.15625, 'x[9] <= 0.736\ngini = 0.278\nsamples = 12\nvalue = [1
0, 2]'),
 Text(0.5116279069767442, 0.09375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.5249169435215947, 0.09375, 'gini = 0.0\nsamples = 10\nvalue = [10, 0]'),
 Text(0.5249169435215947, 0.21875, 'gini = 0.0\nsamples = 206\nvalue = [206, 0]'),
 Text(0.5448504983388704, 0.28125, 'x[17] <= 0.062\ngini = 0.198\nsamples = 9\nvalue = [8

```
 Text(0.5446504983388704, 0.28125, 'x[17] <= 0.062\ngini = 0.198\nsamples = 9\nvalue = [8
, 1]'),
 Text(0.5382059800664452, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5514950166112956, 0.21875, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(0.5647840531561462, 0.34375, 'x[22] <= 0.838\ngini = 0.245\nsamples = 7\nvalue = [6
, 1]'),
 Text(0.5581395348837209, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5714285714285714, 0.28125, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.5348837209302325, 0.53125, 'x[4] <= 0.357\ngini = 0.444\nsamples = 3\nvalue = [2,
1]'),
 Text(0.5282392026578073, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.5415282392026578, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6370431893687708, 0.59375, 'x[9] <= 0.993\ngini = 0.193\nsamples = 148\nvalue = [
132, 16]'),
 Text(0.6303986710963455, 0.53125, 'x[15] <= 0.366\ngini = 0.183\nsamples = 147\nvalue =
[132, 15]'),
 Text(0.6237541528239202, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6370431893687708, 0.46875, 'x[15] <= 0.61\ngini = 0.173\nsamples = 146\nvalue = [
132, 14]'),
 Text(0.6129568106312292, 0.40625, 'x[0] <= 0.536\ngini = 0.238\nsamples = 94\nvalue = [8
1, 13]'),
 Text(0.5913621262458472, 0.34375, 'x[16] <= 0.952\ngini = 0.138\nsamples = 67\nvalue = [
62, 5]'),
 Text(0.584717607973422, 0.28125, 'x[4] <= 0.804\ngini = 0.114\nsamples = 66\nvalue = [62
, 4]'),
 Text(0.5681063122923588, 0.21875, 'x[28] <= 0.029\ngini = 0.064\nsamples = 60\nvalue = [
58, 2]'),
 Text(0.5548172757475083, 0.15625, 'x[7] <= 0.667\ngini = 0.5\nsamples = 2\nvalue = [1, 1
]'),
 Text(0.5481727574750831, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5614617940199336, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5813953488372093, 0.15625, 'x[23] <= 0.583\ngini = 0.034\nsamples = 58\nvalue = [
57, 1]'),
 Text(0.574750830564784, 0.09375, 'gini = 0.0\nsamples = 49\nvalue = [49, 0]'),
 Text(0.5880398671096345, 0.09375, 'x[19] <= 0.071\ngini = 0.198\nsamples = 9\nvalue = [8
, 1]'),
 Text(0.5813953488372093, 0.03125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5946843853820598, 0.03125, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(0.6013289036544851, 0.21875, 'x[28] <= 0.5\ngini = 0.444\nsamples = 6\nvalue = [4,
2]'),
 Text(0.5946843853820598, 0.15625, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.6079734219269103, 0.15625, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.5980066445182725, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6345514950166113, 0.34375, 'x[4] <= 0.268\ngini = 0.417\nsamples = 27\nvalue = [1
9, 8]'),
 Text(0.6212624584717608, 0.28125, 'x[22] <= 0.212\ngini = 0.255\nsamples = 20\nvalue = [
17, 3]'),
 Text(0.6146179401993356, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.627906976744186, 0.21875, 'x[5] <= 0.875\ngini = 0.188\nsamples = 19\nvalue = [17
, 2]'),
 Text(0.6212624584717608, 0.15625, 'x[15] <= 0.609\ngini = 0.105\nsamples = 18\nvalue = [
17, 1]'),
 Text(0.6146179401993356, 0.09375, 'gini = 0.0\nsamples = 17\nvalue = [17, 0]'),
 Text(0.627906976744186, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6345514950166113, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6478405315614618, 0.28125, 'x[28] <= 0.471\ngini = 0.408\nsamples = 7\nvalue = [2
, 5]'),
 Text(0.6411960132890365, 0.21875, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(0.654485049833887, 0.21875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.6611295681063123, 0.40625, 'x[25] <= 0.972\ngini = 0.038\nsamples = 52\nvalue = [
51, 1]'),
 Text(0.654485049833887, 0.34375, 'gini = 0.0\nsamples = 50\nvalue = [50, 0]'),
 Text(0.6677740863787376, 0.34375, 'x[14] <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]
'),
 Text(0.6611295681063123, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6744186046511628, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.6436877076411196, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.4566029900332226, 0.71875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8159260797342193, 0.84375, 'x[15] <= 0.192\ngini = 0.385\nsamples = 300\nvalue =
[222, 78]'),
 Text(0.7323504983388704, 0.78125, 'x[10] <= 0.833\ngini = 0.5\nsamples = 96\nvalue = [49
, 47]'),
 Text(0.709717607973422, 0.71875, 'x[7] <= 0.167\ngini = 0.498\nsamples = 86\nvalue = [40
```

Text(0.7097176079734220, 0.71875, 'x[7] <= 0.167\ngini = 0.498\nsamples = 86\nvalue = [40
, 46]'),
Text(0.6843853820598007, 0.65625, 'x[9] <= 0.193\ngini = 0.245\nsamples = 14\nvalue = [2
, 12]'),
Text(0.6777408637873754, 0.59375, 'x[17] <= 0.188\ngini = 0.444\nsamples = 3\nvalue = [2
, 1]'),
Text(0.6710963455149501, 0.53125, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
Text(0.6843853820598007, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.6910299003322259, 0.59375, 'gini = 0.0\nsamples = 11\nvalue = [0, 11]'),
Text(0.7350498338870431, 0.65625, 'x[2] <= 0.739\ngini = 0.498\nsamples = 72\nvalue = [3
8, 34]'),
Text(0.7159468438538206, 0.59375, 'x[19] <= 0.679\ngini = 0.494\nsamples = 56\nvalue = [
25, 31]'),
Text(0.6976744186046512, 0.53125, 'x[9] <= 0.336\ngini = 0.499\nsamples = 46\nvalue = [2
4, 22]'),
Text(0.6810631229235881, 0.46875, 'x[16] <= 0.514\ngini = 0.36\nsamples = 17\nvalue = [1
3, 4]'),
Text(0.6744186046511628, 0.40625, 'gini = 0.0\nsamples = 11\nvalue = [11, 0]'),
Text(0.6877076411960132, 0.40625, 'x[22] <= 0.113\ngini = 0.444\nsamples = 6\nvalue = [2
, 4]'),
Text(0.6810631229235881, 0.34375, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
Text(0.6943521594684385, 0.34375, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
Text(0.7142857142857143, 0.46875, 'x[1] <= 0.75\ngini = 0.471\nsamples = 29\nvalue = [11
, 18]'),
Text(0.7076411960132890, 0.40625, 'gini = 0.0\nsamples = 9\nvalue = [0, 9]'),
Text(0.7209302325581395, 0.40625, 'x[0] <= 0.44\ngini = 0.495\nsamples = 20\nvalue = [11
, 9]'),
Text(0.7076411960132890, 0.34375, 'x[16] <= 0.815\ngini = 0.444\nsamples = 12\nvalue = [4
, 8]'),
Text(0.7009966777408638, 0.28125, 'x[24] <= 0.833\ngini = 0.198\nsamples = 9\nvalue = [1
, 8]'),
Text(0.6943521594684385, 0.21875, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
Text(0.7076411960132890, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.7142857142857143, 0.28125, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
Text(0.7342192691029901, 0.34375, 'x[2] <= 0.644\ngini = 0.219\nsamples = 8\nvalue = [7,
1]'),
Text(0.7275747508305648, 0.28125, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
Text(0.7408637873754153, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.7342192691029901, 0.53125, 'x[2] <= 0.097\ngini = 0.18\nsamples = 10\nvalue = [1,
9]'),
Text(0.7275747508305648, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.7408637873754153, 0.46875, 'gini = 0.0\nsamples = 9\nvalue = [0, 9]'),
Text(0.7541528239202658, 0.59375, 'x[5] <= 0.125\ngini = 0.305\nsamples = 16\nvalue = [1
3, 3]'),
Text(0.7475083056478405, 0.53125, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.7607973421926910, 0.53125, 'x[22] <= 0.063\ngini = 0.133\nsamples = 14\nvalue = [1
3, 1]'),
Text(0.7541528239202658, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.7674418604651163, 0.46875, 'gini = 0.0\nsamples = 13\nvalue = [13, 0]'),
Text(0.7549833887043189, 0.71875, 'x[9] <= 0.407\ngini = 0.18\nsamples = 10\nvalue = [9,
1]'),
Text(0.7483388704318937, 0.65625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.7616279069767442, 0.65625, 'gini = 0.0\nsamples = 9\nvalue = [9, 0]'),
Text(0.8995016611295681, 0.78125, 'x[14] <= 0.75\ngini = 0.258\nsamples = 204\nvalue = [
173, 31]'),
Text(0.8355481727574751, 0.71875, 'x[4] <= 0.482\ngini = 0.138\nsamples = 147\nvalue = [
136, 11]'),
Text(0.8073089700996677, 0.65625, 'x[25] <= 0.139\ngini = 0.056\nsamples = 105\nvalue =
[102, 3]'),
Text(0.7940199335548173, 0.59375, 'x[9] <= 0.193\ngini = 0.32\nsamples = 10\nvalue = [8,
2]'),
Text(0.7873754152823921, 0.53125, 'x[12] <= 0.562\ngini = 0.444\nsamples = 3\nvalue = [1
, 2]'),
Text(0.7807308970099668, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.7940199335548173, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.8006644518272426, 0.53125, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
Text(0.8205980066445183, 0.59375, 'x[0] <= 0.857\ngini = 0.021\nsamples = 95\nvalue = [9
4, 1]'),
Text(0.8139534883720930, 0.53125, 'gini = 0.0\nsamples = 88\nvalue = [88, 0]'),
Text(0.8272425249169435, 0.53125, 'x[9] <= 0.671\ngini = 0.245\nsamples = 7\nvalue = [6,
1]'),
Text(0.8205980066445183, 0.46875, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
Text(0.8338870431893688, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),

```
Text(0.8353870431893688, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8637873754152824, 0.65625, 'x[7] <= 0.167\ngini = 0.308\nsamples = 42\nvalue = [34, 8]'),
 Text(0.8471760797342193, 0.59375, 'x[2] <= 0.736\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.840531561461794, 0.53125, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.8538205980066446, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.8803986710963455, 0.59375, 'x[0] <= 0.393\ngini = 0.229\nsamples = 38\nvalue = [33, 5]'),
 Text(0.867109634551495, 0.53125, 'x[15] <= 0.428\ngini = 0.5\nsamples = 6\nvalue = [3, 3]'),
 Text(0.8604651162790697, 0.46875, 'x[16] <= 0.874\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),
 Text(0.8538205980066446, 0.40625, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.867109634551495, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8737541528239202, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.893687707641196, 0.53125, 'x[23] <= 0.917\ngini = 0.117\nsamples = 32\nvalue = [30, 2]'),
 Text(0.8870431893687708, 0.46875, 'x[12] <= 0.812\ngini = 0.062\nsamples = 31\nvalue = [30, 1]'),
 Text(0.8803986710963455, 0.40625, 'gini = 0.0\nsamples = 28\nvalue = [28, 0]'),
 Text(0.893687707641196, 0.40625, 'x[23] <= 0.417\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.8870431893687708, 0.34375, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.9003322259136213, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.9003322259136213, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.9634551495016611, 0.71875, 'x[12] <= 0.812\ngini = 0.456\nsamples = 57\nvalue = [37, 20]'),
 Text(0.9401993355481728, 0.65625, 'x[27] <= 0.4\ngini = 0.238\nsamples = 29\nvalue = [25, 4]'),
 Text(0.9269102990033222, 0.59375, 'x[9] <= 0.964\ngini = 0.142\nsamples = 26\nvalue = [24, 2]'),
 Text(0.920265780730897, 0.53125, 'x[10] <= 0.167\ngini = 0.077\nsamples = 25\nvalue = [24, 1]'),
 Text(0.9136212624584718, 0.46875, 'x[15] <= 0.316\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.9069767441860465, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.920265780730897, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.9269102990033222, 0.46875, 'gini = 0.0\nsamples = 23\nvalue = [23, 0]'),
 Text(0.9335548172757475, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.9534883720930233, 0.59375, 'x[27] <= 0.933\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.946843853820598, 0.53125, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.9601328903654485, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.9867109634551495, 0.65625, 'x[27] <= 0.1\ngini = 0.49\nsamples = 28\nvalue = [12, 16]'),
 Text(0.9800664451827242, 0.59375, 'x[10] <= 0.833\ngini = 0.48\nsamples = 20\nvalue = [12, 8]'),
 Text(0.973421926910299, 0.53125, 'x[25] <= 0.028\ngini = 0.415\nsamples = 17\nvalue = [12, 5]'),
 Text(0.9667774086378738, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.9800664451827242, 0.46875, 'x[16] <= 0.505\ngini = 0.32\nsamples = 15\nvalue = [12, 3]'),
 Text(0.973421926910299, 0.40625, 'gini = 0.0\nsamples = 9\nvalue = [9, 0]'),
 Text(0.9867109634551495, 0.40625, 'x[16] <= 0.706\ngini = 0.5\nsamples = 6\nvalue = [3, 3]'),
 Text(0.9800664451827242, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.9933554817275747, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.9867109634551495, 0.53125, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.9933554817275747, 0.59375, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]')]
```



In [66]:

```
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 [67]:

```
grid_search=GridSearchCV(estimator=dtc,param_grid=parameter,cv=5,scoring="accuracy")
```

grid_search.fit(x_train,y_train)

In [69]:

```
grid_search.best_params_
```

Out[69]:

```
{'criterion': 'entropy',
 'max_depth': 5,
 'max_features': 'log2',
 'splitter': 'random'}
```

In [70]:

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

Out[70]:

```
▾ DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=3, max_features='sqrt')
```

In [71]:

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

In [72]:

```
print(classification_report(y_test,pred))
```

```
              precision    recall  f1-score   support

           0       0.84      0.98      0.91       245
           1       0.43      0.06      0.11        49
```

```
    accuracy                           0.83       294
   macro avg       0.63      0.52      0.51       294
weighted avg       0.77      0.83      0.77       294
```

Random forest

In [73]:

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
```

In [74]:

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

In [75]:

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

In [76]:

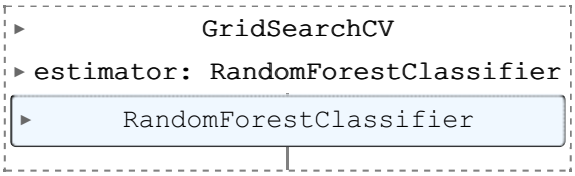
```
rfc_cv.fit(x_train,y_train)
```

```
/Users/samarthgayakhe/anaconda3/lib/python3.11/site-packages/sklearn/model_selection/_val
idation.py:425: FitFailedWarning:
50 fits failed out of a total of 700.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score='rai
se'.

Below are more details about the failures:
--------------------------------------------------------------------------------
50 fits failed with the following error:
Traceback (most recent call last):
  File "/Users/samarthgayakhe/anaconda3/lib/python3.11/site-packages/sklearn/model_select
ion/_validation.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "/Users/samarthgayakhe/anaconda3/lib/python3.11/site-packages/sklearn/base.py", li
ne 1144, in wrapper
    estimator._validate_params()
  File "/Users/samarthgayakhe/anaconda3/lib/python3.11/site-packages/sklearn/base.py", li
ne 637, in _validate_params
    validate_parameter_constraints(
  File "/Users/samarthgayakhe/anaconda3/lib/python3.11/site-packages/sklearn/utils/_param
_validation.py", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'max_features' parameter of Ra
ndomForestClassifier must be an int in the range [1, inf), a float in the range (0.0, 1.0
], a str among {'sqrt', 'log2'} or None. Got 0 instead.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
/Users/samarthgayakhe/anaconda3/lib/python3.11/site-packages/sklearn/model_selection/_sea
rch.py:976: UserWarning: One or more of the test scores are non-finite: [       nan 0.847
79082 0.85459221 0.85801101 0.863103   0.86053165
 0.85713458 0.86223381 0.86222657 0.8596842  0.85287556 0.85456323
 0.85882225 0.86222657        nan 0.84949297 0.85373751 0.85713458
 0.8596842  0.85969144 0.85881501 0.85797479 0.85627988 0.86391424
 0.85880052 0.85967695 0.85796755 0.85798928        nan 0.85204259
 0.85373751 0.8596842  0.85628712 0.86137911 0.85456323 0.86052441
 0.85967695 0.8596842  0.86052441 0.85711285 0.85799652 0.86223381
        nan 0.85120238 0.85885122 0.85545415 0.86224105 0.85711285
 0.86306678 0.85882949 0.86136462 0.86054614 0.85286107 0.85882949
 0.85455599 0.85796755        nan 0.85120238 0.85885122 0.85884398
 0.85883674 0.85882949 0.85967695 0.86053165 0.86053165 0.85796755
 0.85965522 0.85373751 0.85966247 0.85455599]
  warnings.warn(
```

Out[76]:



In [77]:

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

In [78]:

```
print(classification_report(y_test,pred))
```

```
              precision    recall  f1-score   support

           0       0.85      0.98      0.91       245
           1       0.67      0.16      0.26        49

    accuracy                           0.85       294
   macro avg       0.76      0.57      0.59       294
weighted avg       0.82      0.85      0.81       294
```

In [79]:

```
rfc_cv.best_params_
```

Out[79]:

```
{'max_depth': 11, 'max_features': 9}
```