

Assignment 4. Dhana Rahul Sai

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sn
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
```

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

Out [2]:

| | Age | Attrition | BusinessTravel | DailyRate | Department | DistanceFromHome | Education |
|---|-----|-----------|-------------------|-----------|------------------------|------------------|-----------|
| 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 |

5 rows × 35 columns

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

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

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

Out [4]:

| | Age | DailyRate | DistanceFromHome | Education | EmployeeCount | Employ |
|--------------|-------------|-------------|------------------|-------------|---------------|--------|
| count | 1470.000000 | 1470.000000 | 1470.000000 | 1470.000000 | 1470.0 | 1470.0 |
| mean | 36.923810 | 802.485714 | 9.192517 | 2.912925 | 1.0 | 1.0 |
| std | 9.135373 | 403.509100 | 8.106864 | 1.024165 | 0.0 | 0.0 |
| min | 18.000000 | 102.000000 | 1.000000 | 1.000000 | 1.0 | 1.0 |
| 25% | 30.000000 | 465.000000 | 2.000000 | 2.000000 | 1.0 | 1.0 |
| 50% | 36.000000 | 802.000000 | 7.000000 | 3.000000 | 1.0 | 1.0 |
| 75% | 43.000000 | 1157.000000 | 14.000000 | 4.000000 | 1.0 | 1.0 |
| max | 60.000000 | 1499.000000 | 29.000000 | 5.000000 | 1.0 | 2.0 |

8 rows × 26 columns

```
In [5]: df.isnull().any()
```

```
Out[5]: 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 [6]: df.isnull().sum()
```

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

```
In [7]: corr = df.corr()
corr
```

```
/var/folders/ks/ljk00dm1703810nybztmtjgw0000gn/T/ipykernel_52257/2
438084875.py:1: FutureWarning: The default value of numeric_only i
n DataFrame.corr is deprecated. In a future version, it will defau
lt to False. Select only valid columns or specify the value of num
eric_only to silence this warning.
  corr = df.corr()
```

```
Out[7]:
```

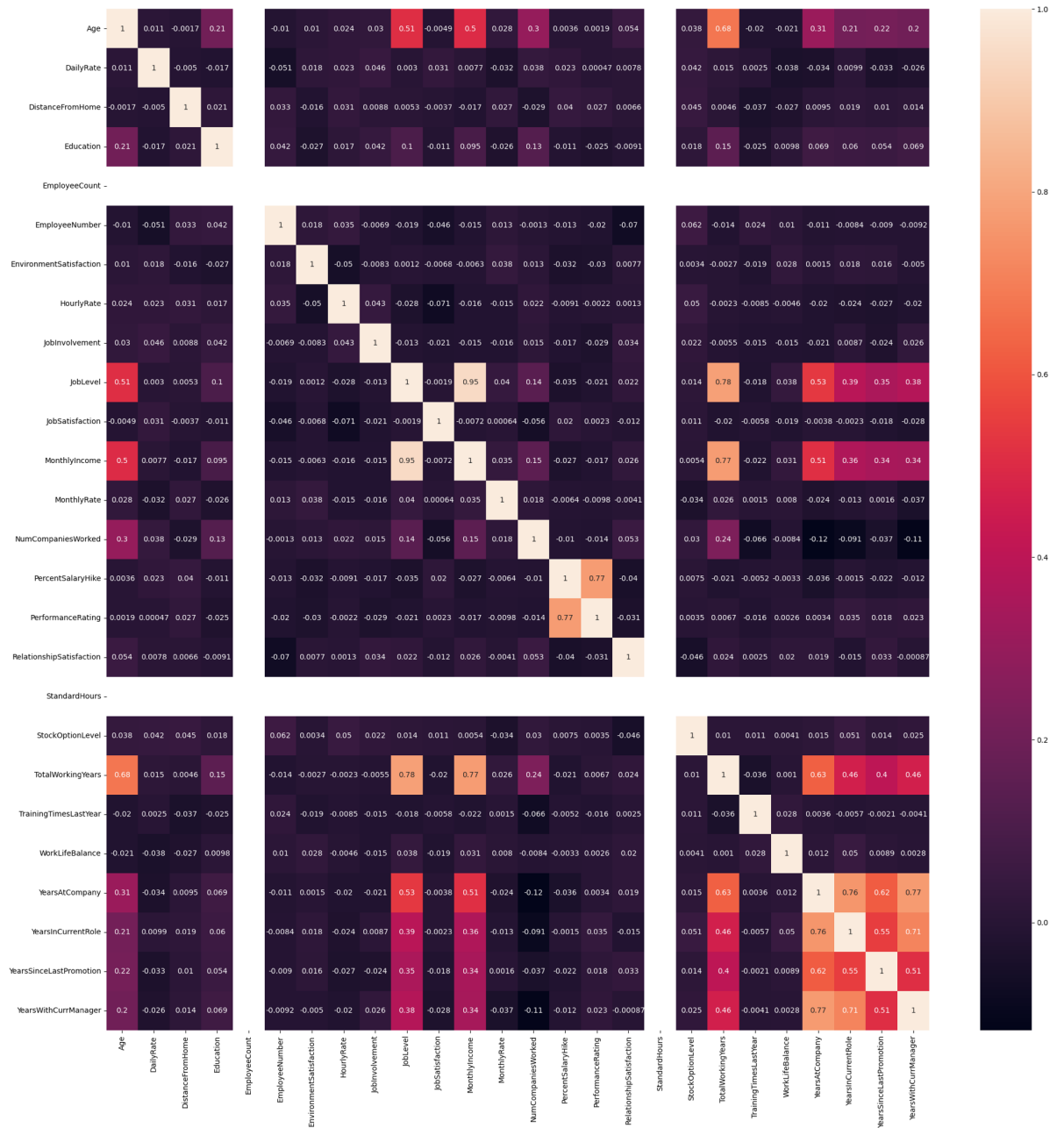
| | Age | DailyRate | DistanceFromHome | Education | EmployeeCc |
|-----|----------|-----------|------------------|-----------|------------|
| Age | 1.000000 | 0.010661 | -0.001686 | 0.208034 | ↑ |

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

26 rows × 26 columns

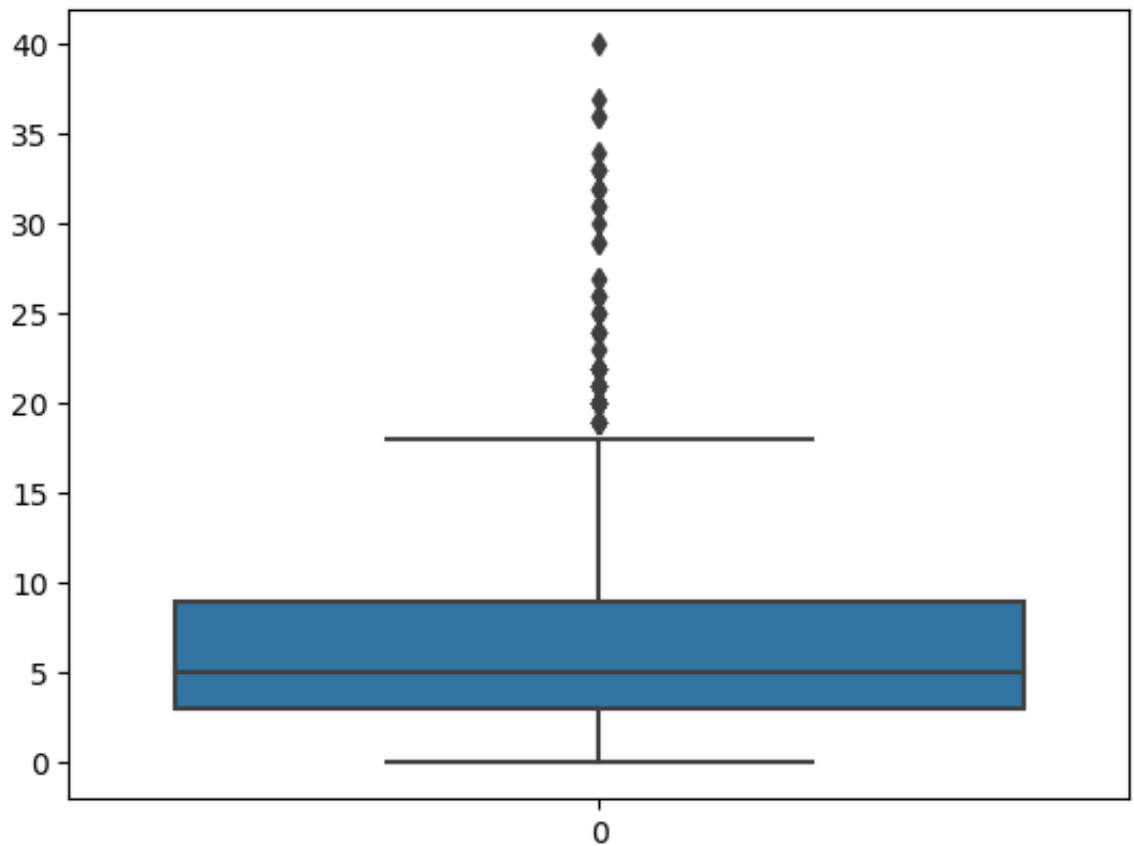
```
In [8]: plt.subplots(figsize = (25,25))
sn.heatmap(corr,annot = True)
```

Out[8]: <Axes: >



```
In [9]: sn.boxplot(df.YearsAtCompany)
```

```
Out[9]: <Axes: >
```



```
In [59]: q1 = df.YearsAtCompany.quantile(0.25)
q3 = df.YearsAtCompany.quantile(0.75)
print(q1)
print(q3)
```

```
3.0
8.0
```

```
In [60]: q3-q1
```

```
Out[60]: 5.0
```

```
In [61]: upperlimit = q3+1.5*(q3-q1)
upperlimit
```

```
Out[61]: 15.5
```

```
In [62]: lowerlimit = q1-1.5*(q3-q1)
lowerlimit
```

```
Out[62]: -4.5
```


In [63]: `df.median()`

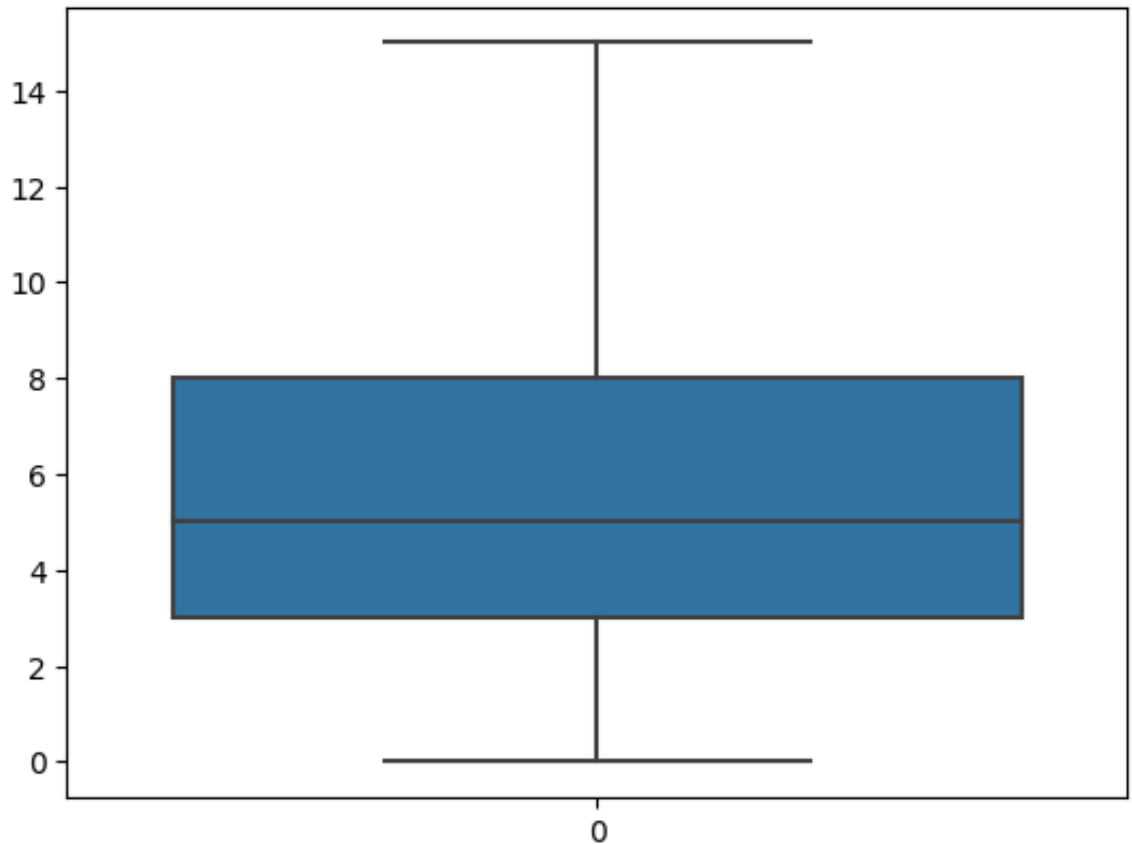
```
/var/folders/ks/ljk00dm1703810nybztmtjgw0000gn/T/ipykernel_52257/530051474.py:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
```

```
df.median()
```

```
Out[63]: Age                36.0
DailyRate                802.0
DistanceFromHome          7.0
Education                 3.0
EmployeeCount             1.0
EmployeeNumber           1020.5
EnvironmentSatisfaction   3.0
HourlyRate                66.0
JobInvolvement            3.0
JobLevel                  2.0
JobSatisfaction           3.0
MonthlyIncome            4919.0
MonthlyRate              14235.5
NumCompaniesWorked        2.0
PercentSalaryHike         14.0
PerformanceRating         3.0
RelationshipSatisfaction   3.0
StandardHours             80.0
StockOptionLevel          1.0
TotalWorkingYears         10.0
TrainingTimesLastYear     3.0
WorkLifeBalance           3.0
YearsAtCompany            5.0
YearsInCurrentRole        3.0
YearsSinceLastPromotion    1.0
YearsWithCurrManager       3.0
dtype: float64
```

```
In [64]: df["YearsAtCompany"] = np.where(df["YearsAtCompany"] > upperlimit, 5, df["YearsAtCompany"])
sn.boxplot(df.YearsAtCompany)
```

Out [64]: <Axes: >



```
In [65]: x = df.drop('Attrition', axis=1)
y = df['Attrition']
```

```
In [66]: x.head()
```

Out [66]:

| | Age | BusinessTravel | DailyRate | Department | DistanceFromHome | Education | Education |
|---|-----|-------------------|-----------|------------------------|------------------|-----------|-----------|
| 0 | 41 | Travel_Rarely | 1102 | Sales | 1 | 2 | Life Sci |
| 1 | 49 | Travel_Frequently | 279 | Research & Development | 8 | 1 | Life Sci |
| 2 | 37 | Travel_Rarely | 1373 | Research & Development | 2 | 2 | |
| 3 | 33 | Travel_Frequently | 1392 | Research & Development | 3 | 4 | Life Sci |
| 4 | 27 | Travel_Rarely | 591 | Research & Development | 2 | 1 | Ma |

5 rows × 34 columns

In [67]: `type(x)`

Out[67]: `pandas.core.frame.DataFrame`

In [68]: `type(y)`

Out[68]: `pandas.core.series.Series`

In [69]: `le = LabelEncoder()
x.BusinessTravel = le.fit_transform(x.BusinessTravel)
x.head()`

Out[69]:

| | Age | BusinessTravel | DailyRate | Department | DistanceFromHome | Education | EducationF |
|---|-----|----------------|-----------|------------------------|------------------|-----------|------------|
| 0 | 41 | 2 | 1102 | Sales | 1 | 2 | Life Scier |
| 1 | 49 | 1 | 279 | Research & Development | 8 | 1 | Life Scier |
| 2 | 37 | 2 | 1373 | Research & Development | 2 | 2 | C |
| 3 | 33 | 1 | 1392 | Research & Development | 3 | 4 | Life Scier |
| 4 | 27 | 2 | 591 | Research & Development | 2 | 1 | Mec |

5 rows × 34 columns

In [70]: `y = le.fit_transform(y)
y`

Out[70]: `array([1, 0, 1, ..., 0, 0, 0])`

In [71]: `x.Department = le.fit_transform(x.Department)
x.head()`

Out[71]:

| | Age | BusinessTravel | DailyRate | Department | DistanceFromHome | Education | EducationF |
|---|-----|----------------|-----------|------------|------------------|-----------|------------|
| 0 | 41 | 2 | 1102 | 2 | 1 | 2 | Life Scien |
| 1 | 49 | 1 | 279 | 1 | 8 | 1 | Life Scien |
| 2 | 37 | 2 | 1373 | 1 | 2 | 2 | Of |
| 3 | 33 | 1 | 1392 | 1 | 3 | 4 | Life Scien |
| 4 | 27 | 2 | 591 | 1 | 2 | 1 | Med |

5 rows × 34 columns

```
In [72]: x.EducationField = le.fit_transform(x.EducationField)
x.head()
```

Out [72]:

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

5 rows × 34 columns

```
In [73]: x.Gender = le.fit_transform(x.Gender)
x.head()
```

Out [73]:

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

5 rows × 34 columns

```
In [74]: x.MaritalStatus = le.fit_transform(x.MaritalStatus)
x.Over18 = le.fit_transform(x.Over18)
x.OverTime = le.fit_transform(x.OverTime)
```

```
In [75]: x.JobRole = le.fit_transform(x.JobRole)
x.head()
```

Out [75]:

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

5 rows × 34 columns

```
In [76]: ms = MinMaxScaler()
```

```
In [77]: x_scaled= pd.DataFrame(ms.fit_transform(x),columns=x.columns)
x_scaled.head()
```

Out [77]:

| | Age | BusinessTravel | DailyRate | Department | DistanceFromHome | Education | Educational |
|---|----------|----------------|-----------|------------|------------------|-----------|-------------|
| 0 | 0.547619 | 1.0 | 0.715820 | 1.0 | 0.000000 | 0.25 | |
| 1 | 0.738095 | 0.5 | 0.126700 | 0.5 | 0.250000 | 0.00 | |
| 2 | 0.452381 | 1.0 | 0.909807 | 0.5 | 0.035714 | 0.25 | |
| 3 | 0.357143 | 0.5 | 0.923407 | 0.5 | 0.071429 | 0.75 | |
| 4 | 0.214286 | 1.0 | 0.350036 | 0.5 | 0.035714 | 0.00 | |

5 rows × 34 columns

In [78]: `x.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 34 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Age                                   1470 non-null   int64
1   BusinessTravel                       1470 non-null   int64
2   DailyRate                           1470 non-null   int64
3   Department                           1470 non-null   int64
4   DistanceFromHome                    1470 non-null   int64
5   Education                           1470 non-null   int64
6   EducationField                       1470 non-null   int64
7   EmployeeCount                       1470 non-null   int64
8   EmployeeNumber                      1470 non-null   int64
9   EnvironmentSatisfaction              1470 non-null   int64
10  Gender                               1470 non-null   int64
11  HourlyRate                           1470 non-null   int64
12  JobInvolvement                       1470 non-null   int64
13  JobLevel                             1470 non-null   int64
14  JobRole                              1470 non-null   int64
15  JobSatisfaction                      1470 non-null   int64
16  MaritalStatus                       1470 non-null   int64
17  MonthlyIncome                       1470 non-null   int64
18  MonthlyRate                         1470 non-null   int64
19  NumCompaniesWorked                  1470 non-null   int64
20  Over18                              1470 non-null   int64
21  OverTime                            1470 non-null   int64
22  PercentSalaryHike                   1470 non-null   int64
23  PerformanceRating                   1470 non-null   int64
24  RelationshipSatisfaction             1470 non-null   int64
25  StandardHours                       1470 non-null   int64
26  StockOptionLevel                    1470 non-null   int64
27  TotalWorkingYears                   1470 non-null   int64
28  TrainingTimesLastYear               1470 non-null   int64
29  WorkLifeBalance                     1470 non-null   int64
30  YearsAtCompany                      1470 non-null   int64
31  YearsInCurrentRole                  1470 non-null   int64
32  YearsSinceLastPromotion              1470 non-null   int64
33  YearsWithCurrManager                 1470 non-null   int64
dtypes: int64(34)
memory usage: 390.6 KB
```

In [79]: `x_train,x_test,y_train,y_test = train_test_split(x_scaled,y,test_si`

In [80]: `x_train.shape,x_test.shape,y_train.shape,y_test.shape`

Out[80]: `((1176, 34), (294, 34), (1176,), (294,))`

In [81]: `model = LogisticRegression()`

```
In [82]: model.fit(x_train,y_train)
pred = model.predict(x_test)
pred
```

```
Out[82]: 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, 0, 0, 0, 1, 1, 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, 1, 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, 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, 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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,
0, 0, 0, 0, 0, 1, 0, 0])
```

In [83]: `y_test`

Out[83]: `array([0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 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, 1, 0, 0, 0,
0, 0,
0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
0, 0,
1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 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, 1,
1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0,
0, 1,
0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 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,
1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0,
0, 0,
0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0, 0,
0, 1, 0, 0, 0, 1, 0, 0])`

In [84]: `from sklearn.metrics import accuracy_score, confusion_matrix, classif`

In [85]: `accuracy_score(y_test, pred)`

Out[85]: `0.8843537414965986`

In [86]: `confusion_matrix(y_test, pred)`

Out[86]: `array([[242, 3],
[31, 18]])`


```
In [87]: print(classification_report(y_test,pred))
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.89 | 0.99 | 0.93 | 245 |
| 1 | 0.86 | 0.37 | 0.51 | 49 |
| accuracy | | | 0.88 | 294 |
| macro avg | 0.87 | 0.68 | 0.72 | 294 |
| weighted avg | 0.88 | 0.88 | 0.86 | 294 |

```
In [88]: probability = model.predict_proba(x_test)[: ,1]
```

```
In [89]: probability
```

```
Out[89]: array([0.13129344, 0.19063335, 0.31606743, 0.06912079, 0.66111675,
0.06664352, 0.53360486, 0.05535516, 0.01115506, 0.37890312,
0.06554933, 0.31683836, 0.0174328 , 0.67487634, 0.21223868,
0.03272593, 0.09716532, 0.17867399, 0.0490532 , 0.19100706,
0.25062208, 0.02612901, 0.05602917, 0.05738692, 0.59302378,
0.40852515, 0.05769446, 0.03164683, 0.70222873, 0.06687497,
0.01246585, 0.024766 , 0.08406986, 0.17287548, 0.07911948,
0.02971553, 0.09775613, 0.06744791, 0.03396524, 0.05444227,
0.08937865, 0.01682485, 0.03031769, 0.02127854, 0.02039929,
0.49851904, 0.43037902, 0.00316471, 0.76468481, 0.52075908,
0.12103073, 0.48568596, 0.07290815, 0.26651569, 0.68813509,
0.29060313, 0.01866981, 0.3218195 , 0.02646703, 0.17563982,
0.01970086, 0.22970281, 0.16896101, 0.03516105, 0.41102226,
0.0367185 , 0.24892659, 0.11974632, 0.09210885, 0.10160491,
0.13712867, 0.30428858, 0.06266796, 0.07335326, 0.12528893,
0.06432958, 0.0498183 , 0.08054668, 0.2016799 , 0.02873493,
0.00651569, 0.02199604, 0.15106584, 0.02511468, 0.02952711,
0.07366313, 0.00859023, 0.03365362, 0.03772954, 0.13836696,
0.30494001, 0.17837697, 0.28723058, 0.2521727 , 0.01741985,
0.1888494 , 0.35399373, 0.29018696, 0.07659449, 0.05055041,
0.25500554, 0.75967666, 0.36520339, 0.0203079 , 0.0959336 ,
0.02959457, 0.04998732, 0.16196842, 0.05265155, 0.11869794,
0.08143997, 0.05082647, 0.02712705, 0.13795176, 0.06262179,
0.03052752, 0.04435616, 0.12439291, 0.00669959, 0.01090308,
0.16429022, 0.04520733, 0.06724621, 0.82510359, 0.03426937,
0.04052945, 0.00944367, 0.13553925, 0.1735486 , 0.05026756,
0.01841438, 0.26713763, 0.57966781, 0.3303566 , 0.04363467,
0.4170305 , 0.58246893, 0.13521646, 0.08095531, 0.25749714,
0.11008118, 0.0706616 , 0.10296284, 0.13399714, 0.20637223,
0.02396339, 0.18217252, 0.00718892, 0.12566219, 0.17079061,
0.06299132, 0.14645391, 0.06119035, 0.15316349, 0.03211461,
0.02190932, 0.07523043, 0.07882732, 0.01241429, 0.01017215,
0.50437288, 0.00884962, 0.16485885, 0.82504984, 0.10939135,
0.29008563, 0.15605828, 0.14065239, 0.0313591 , 0.00618827,
0.04130253, 0.07886936, 0.13111324, 0.10819328, 0.02745423,
0.14792273, 0.10253452, 0.08750792, 0.05099305, 0.10389624,
```

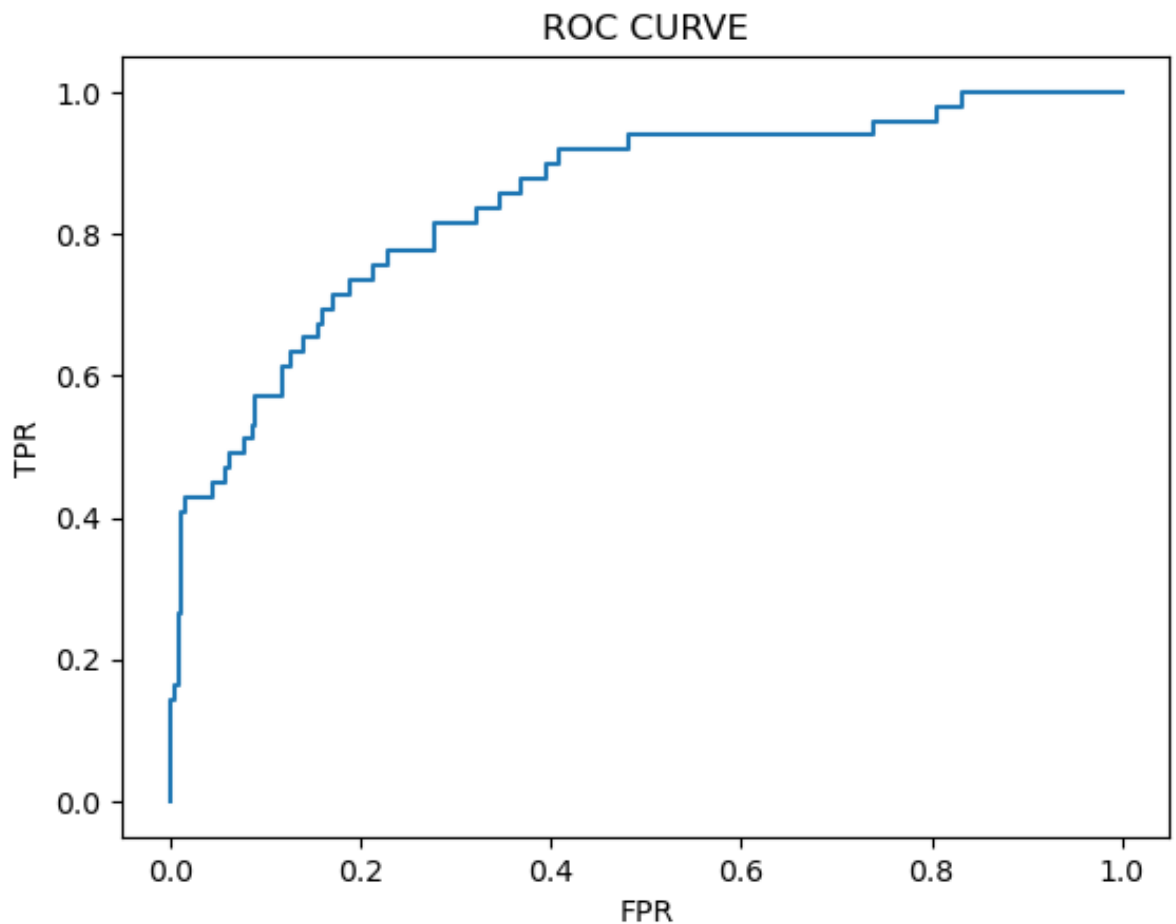
```

0.03043292, 0.09934079, 0.00654389, 0.78755642, 0.03984101,
0.04018459, 0.40481172, 0.04423958, 0.75269826, 0.11009143,
0.37930431, 0.39525297, 0.26918214, 0.05260852, 0.08744177,
0.15176058, 0.04212169, 0.01255515, 0.28326466, 0.05069983,
0.14448888, 0.16065742, 0.68330218, 0.06140364, 0.23366946,
0.03252966, 0.4811088 , 0.00304084, 0.14248158, 0.02473565,
0.12173117, 0.16736195, 0.05674375, 0.10454042, 0.15180532,
0.02767503, 0.02195401, 0.07535491, 0.03231883, 0.14763257,
0.08409273, 0.23106975, 0.74360528, 0.13342724, 0.38878379,
0.01519813, 0.12253089, 0.22980165, 0.34257431, 0.03828766,
0.04509894, 0.32718126, 0.05519182, 0.01657527, 0.17729295,
0.37212058, 0.24963294, 0.00754558, 0.08581666, 0.00916159,
0.13673904, 0.27136552, 0.01306282, 0.16965985, 0.03761936,
0.03450295, 0.4196152 , 0.37337648, 0.03552566, 0.11274966,
0.39679707, 0.3198251 , 0.81227789, 0.04812145, 0.18963906,
0.07416573, 0.00608465, 0.68769554, 0.38276649, 0.37284333,
0.36195972, 0.03399947, 0.16796271, 0.06025104, 0.06681017,
0.10803038, 0.00687341, 0.2312921 , 0.4714917 , 0.07107663,
0.09116817, 0.01311644, 0.15572877, 0.05018955, 0.01958606,
0.02465726, 0.06610706, 0.25452864, 0.26035366, 0.20232153,
0.28205369, 0.01569937, 0.16370365, 0.09700501, 0.0321814 ,
0.1961292 , 0.00848977, 0.22133319, 0.00435181, 0.02071151,
0.20833583, 0.74305185, 0.08042248, 0.287313  ])

```

```
In [90]: fpr,tpr,threshholds = roc_curve(y_test,probability)
```

```
In [91]: plt.plot(fpr, tpr)
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('ROC CURVE')
plt.show()
```



Decision Tree

```
In [92]: from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier()
```

```
In [93]: y_train
```

```
Out[93]: array([0, 0, 0, ..., 0, 0, 0])
```

```
In [94]: dt.fit(x_train, y_train)
```

```
Out[94]: ▼ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
In [95]: pred = dt.predict(x_test)
```

```
In [96]: pred
```

```
Out[96]: array([0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0, 0,
0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0,
0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
0, 1,
1, 0, 0, 1, 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, 1, 0, 0, 0, 0, 0, 0,
0, 1,
1, 1, 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, 0, 0, 0, 0, 0,
0, 1,
0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 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, 0,
1, 1,
0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0,
0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0,
0, 1,
0, 0, 0, 0, 0, 0, 0, 0, 0])
```

In [97]: `y_test`

Out[97]: `array([0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 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, 1, 0, 0, 0,
0, 0,
0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
0, 0,
1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 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, 1,
1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0,
0, 1,
0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 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,
1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0,
0, 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, 1, 0, 0])`

In [98]: `accuracy_score(y_test, pred)`

Out[98]: `0.7619047619047619`

In [99]: `confusion_matrix(y_test, pred)`

Out[99]: `array([[206, 39],
[31, 18]])`

In [100]: `print(classification_report(y_test, pred))`

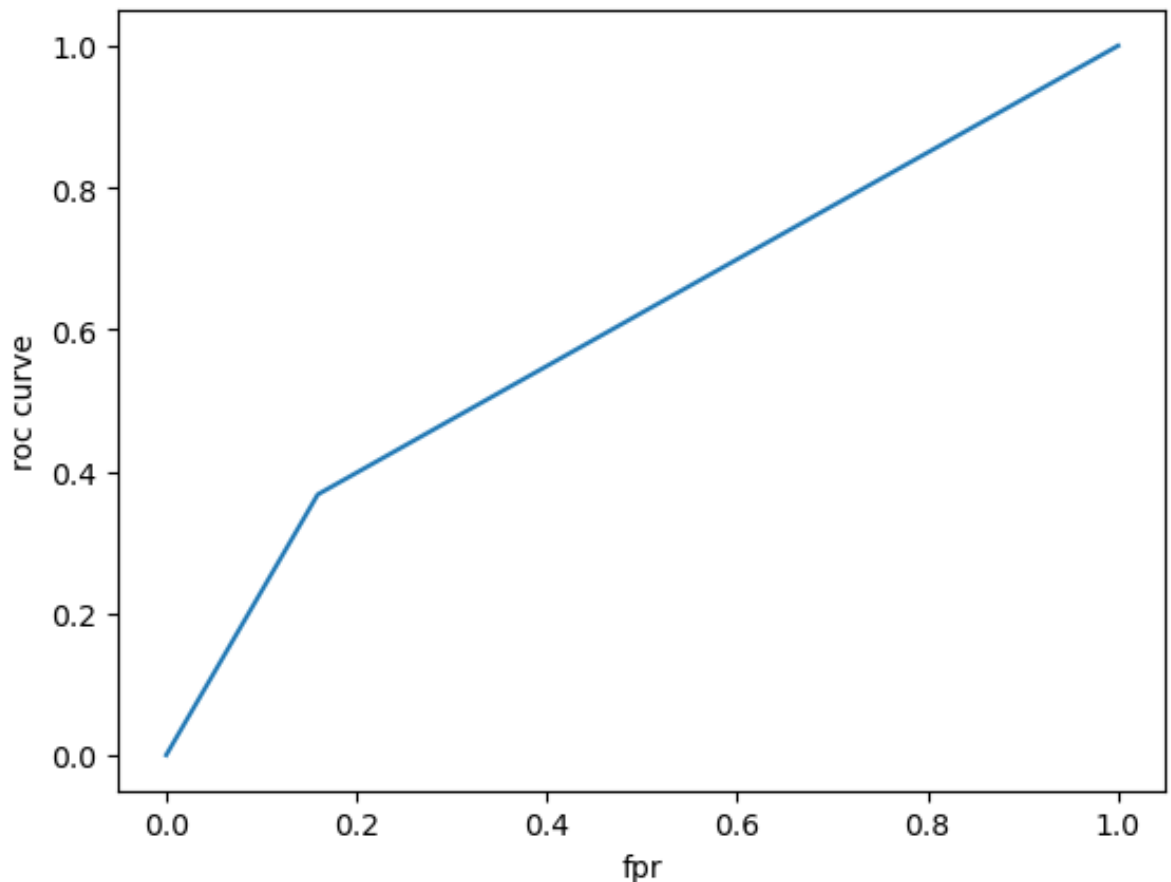
| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.87 | 0.84 | 0.85 | 245 |
| 1 | 0.32 | 0.37 | 0.34 | 49 |
| accuracy | | | 0.76 | 294 |
| macro avg | 0.59 | 0.60 | 0.60 | 294 |
| weighted avg | 0.78 | 0.76 | 0.77 | 294 |

```
In [101]: probab = dt.predict_proba(x_test)[: ,1]
          probab
```

```
Out[101]: array([0., 0., 0., 1., 0., 0., 1., 0., 0., 0., 0., 0., 0., 1., 0.,
0., 0.,
           0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
0., 0.,
           0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1.,
1., 1.,
           1., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0.,
0., 0.,
           0., 0., 1., 1., 0., 1., 0., 0., 1., 0., 0., 0., 0., 0., 0.,
1., 0.,
           0., 0., 1., 1., 0., 0., 1., 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., 1., 0., 0., 0., 0., 0., 0., 0., 1., 1., 1.,
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., 0.,
           0., 0., 0., 0., 0., 1., 0., 1., 1., 0., 0., 0., 0., 1., 0.,
0., 1.,
           0., 0., 1., 0., 1., 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., 0., 0.,
1.,
           0., 1., 0., 0., 1., 0., 1., 1., 0., 0., 1., 0., 0., 0., 0.,
0., 0.,
           0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1.,
0., 0.,
           0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 0.,
0., 0.,
           0., 0., 0., 1., 0., 1., 0., 1., 0., 1., 1., 0., 0., 1., 0.,
0., 0.,
           0., 0., 0., 0., 0., 0.] )
```

```
In [102]: fpr, tpr, threshholds = roc_curve(y_test, probab)
```

```
In [103]: plt.plot(fpr,tpr)
plt.xlabel('fpr')
plt.ylabel('roc curve')
plt.show()
```



```
In [104]: from sklearn import tree
plt.figure(figsize = (25,25))
tree.plot_tree(dt,filled= True)
```

```
Out[104]: [Text(0.3142744479495268, 0.9705882352941176, 'x[27] <= 0.038\nGIN
i = 0.269\nsamples = 1176\nvalue = [988, 188]'),
Text(0.0694006309148265, 0.9117647058823529, 'x[16] <= 0.75\nGIN
i = 0.5\nsamples = 78\nvalue = [39, 39]'),
Text(0.04100946372239748, 0.8529411764705882, 'x[4] <= 0.554\nGIN
i = 0.426\nsamples = 39\nvalue = [27, 12]'),
Text(0.025236593059936908, 0.7941176470588235, 'x[15] <= 0.167\nGIN
i = 0.312\nsamples = 31\nvalue = [25, 6]'),
Text(0.012618296529968454, 0.7352941176470589, 'x[9] <= 0.5\nGIN
i = 0.49\nsamples = 7\nvalue = [3, 4]'),
Text(0.006309148264984227, 0.6764705882352942, 'gini = 0.0\nsampl
es = 3\nvalue = [0, 3]'),
Text(0.01892744479495268, 0.6764705882352942, 'x[16] <= 0.25\nGIN
i = 0.375\nsamples = 4\nvalue = [3, 1]'),
Text(0.012618296529968454, 0.6176470588235294, 'gini = 0.0\nsampl
es = 3\nvalue = [3, 0]'),
Text(0.025236593059936908, 0.6176470588235294, 'gini = 0.0\nsampl
es = 1\nvalue = [0, 1]'),
```

```

Text(0.03785488958990536, 0.7352941176470589, 'x[19] <= 0.056\ngini = 0.153\nsamples = 24\nvalue = [22, 2]'),
Text(0.031545741324921134, 0.6764705882352942, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.04416403785488959, 0.6764705882352942, 'x[9] <= 0.167\ngini = 0.083\nsamples = 23\nvalue = [22, 1]'),
Text(0.03785488958990536, 0.6176470588235294, 'x[10] <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.031545741324921134, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.04416403785488959, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.050473186119873815, 0.6176470588235294, 'gini = 0.0\nsamples = 21\nvalue = [21, 0]'),
Text(0.056782334384858045, 0.7941176470588235, 'x[22] <= 0.679\ngini = 0.375\nsamples = 8\nvalue = [2, 6]'),
Text(0.050473186119873815, 0.7352941176470589, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
Text(0.06309148264984227, 0.7352941176470589, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
Text(0.09779179810725552, 0.8529411764705882, 'x[11] <= 0.364\ngini = 0.426\nsamples = 39\nvalue = [12, 27]'),
Text(0.08201892744479496, 0.7941176470588235, 'x[29] <= 0.167\ngini = 0.133\nsamples = 14\nvalue = [1, 13]'),
Text(0.07570977917981073, 0.7352941176470589, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.08832807570977919, 0.7352941176470589, 'gini = 0.0\nsamples = 13\nvalue = [0, 13]'),
Text(0.11356466876971609, 0.7941176470588235, 'x[8] <= 0.105\ngini = 0.493\nsamples = 25\nvalue = [11, 14]'),
Text(0.10094637223974763, 0.7352941176470589, 'x[1] <= 0.75\ngini = 0.278\nsamples = 6\nvalue = [5, 1]'),
Text(0.0946372239747634, 0.6764705882352942, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.10725552050473186, 0.6764705882352942, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
Text(0.12618296529968454, 0.7352941176470589, 'x[15] <= 0.5\ngini = 0.432\nsamples = 19\nvalue = [6, 13]'),
Text(0.11987381703470032, 0.6764705882352942, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]'),
Text(0.13249211356466878, 0.6764705882352942, 'x[6] <= 0.4\ngini = 0.5\nsamples = 12\nvalue = [6, 6]'),
Text(0.11987381703470032, 0.6176470588235294, 'x[14] <= 0.875\ngini = 0.278\nsamples = 6\nvalue = [5, 1]'),
Text(0.11356466876971609, 0.5588235294117647, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
Text(0.12618296529968454, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.14511041009463724, 0.6176470588235294, 'x[8] <= 0.249\ngini = 0.278\nsamples = 6\nvalue = [1, 5]'),
Text(0.138801261829653, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.15141955835962145, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]')

```



```

s = 5\nvalue = [0, 5]'),
Text(0.5591482649842271, 0.9117647058823529, 'x[21] <= 0.5\ngini
= 0.235\nsamples = 1098\nvalue = [949, 149]'),
Text(0.2854889589905363, 0.8529411764705882, 'x[29] <= 0.167\ngini
= 0.162\nsamples = 798\nvalue = [727, 71]'),
Text(0.1640378548895899, 0.7941176470588235, 'x[8] <= 0.445\ngini
= 0.38\nsamples = 47\nvalue = [35, 12]'),
Text(0.15141955835962145, 0.7352941176470589, 'x[16] <= 0.75\ngini
= 0.1\nsamples = 19\nvalue = [18, 1]'),
Text(0.14511041009463724, 0.6764705882352942, 'gini = 0.0\nsamples
= 18\nvalue = [18, 0]'),
Text(0.15772870662460567, 0.6764705882352942, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.17665615141955837, 0.7352941176470589, 'x[17] <= 0.094\ngini
= 0.477\nsamples = 28\nvalue = [17, 11]'),
Text(0.17034700315457413, 0.6764705882352942, 'gini = 0.0\nsamples
= 4\nvalue = [0, 4]'),
Text(0.1829652996845426, 0.6764705882352942, 'x[32] <= 0.6\ngini
= 0.413\nsamples = 24\nvalue = [17, 7]'),
Text(0.17665615141955837, 0.6176470588235294, 'x[11] <= 0.486\ngini
= 0.351\nsamples = 22\nvalue = [17, 5]'),
Text(0.17034700315457413, 0.5588235294117647, 'x[24] <= 0.5\ngini
= 0.496\nsamples = 11\nvalue = [6, 5]'),
Text(0.1640378548895899, 0.5, 'x[27] <= 0.525\ngini = 0.408\nsamples
= 7\nvalue = [2, 5]'),
Text(0.15772870662460567, 0.4411764705882353, 'gini = 0.0\nsamples
= 5\nvalue = [0, 5]'),
Text(0.17034700315457413, 0.4411764705882353, 'gini = 0.0\nsamples
= 2\nvalue = [2, 0]'),
Text(0.17665615141955837, 0.5, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0]'),
Text(0.1829652996845426, 0.5588235294117647, 'gini = 0.0\nsamples
= 11\nvalue = [11, 0]'),
Text(0.1892744479495268, 0.6176470588235294, 'gini = 0.0\nsamples
= 2\nvalue = [0, 2]'),
Text(0.4069400630914827, 0.7941176470588235, 'x[27] <= 0.975\ngini
= 0.145\nsamples = 751\nvalue = [692, 59]'),
Text(0.40063091482649843, 0.7352941176470589, 'x[30] <= 0.3\ngini
= 0.143\nsamples = 750\nvalue = [692, 58]'),
Text(0.2787854889589905, 0.6764705882352942, 'x[9] <= 0.167\ngini
= 0.218\nsamples = 257\nvalue = [225, 32]'),
Text(0.23817034700315456, 0.6176470588235294, 'x[33] <= 0.147\ngini
= 0.355\nsamples = 65\nvalue = [50, 15]'),
Text(0.21766561514195584, 0.5588235294117647, 'x[33] <= 0.029\ngini
= 0.303\nsamples = 59\nvalue = [48, 11]'),
Text(0.19558359621451105, 0.5, 'x[12] <= 0.5\ngini = 0.463\nsamples
= 22\nvalue = [14, 8]'),
Text(0.1829652996845426, 0.4411764705882353, 'x[11] <= 0.179\ngini
= 0.198\nsamples = 9\nvalue = [8, 1]'),
Text(0.17665615141955837, 0.38235294117647056, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.1892744479495268, 0.38235294117647056, 'gini = 0.0\nsamples
= 8\nvalue = [8, 0]'),

```

```

Text(0.2082018927444795, 0.4411764705882353, 'x[11] <= 0.4\nngini
= 0.497\nsamples = 13\nvalue = [6, 7]'),
Text(0.20189274447949526, 0.38235294117647056, 'gini = 0.0\nsampl
es = 4\nvalue = [4, 0]'),
Text(0.21451104100946372, 0.38235294117647056, 'x[4] <= 0.286\nngi
ni = 0.346\nsamples = 9\nvalue = [2, 7]'),
Text(0.2082018927444795, 0.3235294117647059, 'x[2] <= 0.369\nngini
= 0.444\nsamples = 3\nvalue = [2, 1]'),
Text(0.20189274447949526, 0.2647058823529412, 'gini = 0.0\nsampl
es = 2\nvalue = [2, 0]'),
Text(0.21451104100946372, 0.2647058823529412, 'gini = 0.0\nsampl
es = 1\nvalue = [0, 1]'),
Text(0.22082018927444794, 0.3235294117647059, 'gini = 0.0\nsampl
es = 6\nvalue = [0, 6]'),
Text(0.23974763406940064, 0.5, 'x[15] <= 0.167\nngini = 0.149\nsam
ples = 37\nvalue = [34, 3]'),
Text(0.2334384858044164, 0.4411764705882353, 'x[29] <= 0.5\nngini
= 0.5\nsamples = 6\nvalue = [3, 3]'),
Text(0.22712933753943218, 0.38235294117647056, 'gini = 0.0\nsampl
es = 3\nvalue = [3, 0]'),
Text(0.23974763406940064, 0.38235294117647056, 'gini = 0.0\nsampl
es = 3\nvalue = [0, 3]'),
Text(0.24605678233438485, 0.4411764705882353, 'gini = 0.0\nsampl
es = 31\nvalue = [31, 0]'),
Text(0.2586750788643533, 0.5588235294117647, 'x[8] <= 0.065\nngini
= 0.444\nsamples = 6\nvalue = [2, 4]'),
Text(0.25236593059936907, 0.5, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
Text(0.26498422712933756, 0.5, 'gini = 0.0\nsamples = 4\nvalue =
[0, 4]'),
Text(0.3194006309148265, 0.6176470588235294, 'x[0] <= 0.321\nngini
= 0.161\nsamples = 192\nvalue = [175, 17]'),
Text(0.28391167192429023, 0.5588235294117647, 'x[6] <= 0.1\nngini
= 0.294\nsamples = 67\nvalue = [55, 12]'),
Text(0.277602523659306, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [0
, 2]'),
Text(0.2902208201892745, 0.5, 'x[29] <= 0.5\nngini = 0.26\nsamples
= 65\nvalue = [55, 10]'),
Text(0.26813880126182965, 0.4411764705882353, 'x[6] <= 0.5\nngini
= 0.469\nsamples = 16\nvalue = [10, 6]'),
Text(0.2618296529968454, 0.38235294117647056, 'gini = 0.0\nsampl
es = 7\nvalue = [7, 0]'),
Text(0.2744479495268139, 0.38235294117647056, 'x[9] <= 0.833\nngin
i = 0.444\nsamples = 9\nvalue = [3, 6]'),
Text(0.26813880126182965, 0.3235294117647059, 'gini = 0.0\nsampl
es = 5\nvalue = [0, 5]'),
Text(0.2807570977917981, 0.3235294117647059, 'x[17] <= 0.155\nngin
i = 0.375\nsamples = 4\nvalue = [3, 1]'),
Text(0.2744479495268139, 0.2647058823529412, 'gini = 0.0\nsamples
= 3\nvalue = [3, 0]'),
Text(0.2870662460567823, 0.2647058823529412, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.31230283911671924, 0.4411764705882353, 'x[2] <= 0.037\nngin

```

```

i = 0.15\nsamples = 49\nvalue = [45, 4]'),
Text(0.305993690851735, 0.38235294117647056, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.3186119873817035, 0.38235294117647056, 'x[2] <= 0.938\ngini
i = 0.117\nsamples = 48\nvalue = [45, 3]'),
Text(0.31230283911671924, 0.3235294117647059, 'x[5] <= 0.875\ngini
i = 0.081\nsamples = 47\nvalue = [45, 2]'),
Text(0.2996845425867508, 0.2647058823529412, 'x[12] <= 0.167\ngini
i = 0.043\nsamples = 45\nvalue = [44, 1]'),
Text(0.29337539432176657, 0.20588235294117646, 'x[15] <= 0.5\ngini
i = 0.444\nsamples = 3\nvalue = [2, 1]'),
Text(0.2870662460567823, 0.14705882352941177, 'gini = 0.0\nsample
s = 1\nvalue = [0, 1]'),
Text(0.2996845425867508, 0.14705882352941177, 'gini = 0.0\nsample
s = 2\nvalue = [2, 0]'),
Text(0.305993690851735, 0.20588235294117646, 'gini = 0.0\nsamples
= 42\nvalue = [42, 0]'),
Text(0.3249211356466877, 0.2647058823529412, 'x[24] <= 0.5\ngini
= 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.3186119873817035, 0.20588235294117646, 'gini = 0.0\nsample
s = 1\nvalue = [0, 1]'),
Text(0.3312302839116719, 0.20588235294117646, 'gini = 0.0\nsample
s = 1\nvalue = [1, 0]'),
Text(0.3249211356466877, 0.3235294117647059, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.3548895899053628, 0.5588235294117647, 'x[8] <= 0.022\ngini
= 0.077\nsamples = 125\nvalue = [120, 5]'),
Text(0.33753943217665616, 0.5, 'x[14] <= 0.5\ngini = 0.5\nsamples
= 4\nvalue = [2, 2]'),
Text(0.3312302839116719, 0.4411764705882353, 'gini = 0.0\nsamples
= 2\nvalue = [2, 0]'),
Text(0.3438485804416404, 0.4411764705882353, 'gini = 0.0\nsamples
= 2\nvalue = [0, 2]'),
Text(0.3722397476340694, 0.5, 'x[18] <= 0.968\ngini = 0.048\nsamp
les = 121\nvalue = [118, 3]'),
Text(0.35646687697160884, 0.4411764705882353, 'x[2] <= 0.98\ngini
= 0.033\nsamples = 118\nvalue = [116, 2]'),
Text(0.3438485804416404, 0.38235294117647056, 'x[14] <= 0.938\ngi
ni = 0.017\nsamples = 114\nvalue = [113, 1]'),
Text(0.33753943217665616, 0.3235294117647059, 'gini = 0.0\nsample
s = 107\nvalue = [107, 0]'),
Text(0.3501577287066246, 0.3235294117647059, 'x[11] <= 0.193\ngini
i = 0.245\nsamples = 7\nvalue = [6, 1]'),
Text(0.3438485804416404, 0.2647058823529412, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.35646687697160884, 0.2647058823529412, 'gini = 0.0\nsample
s = 6\nvalue = [6, 0]'),
Text(0.36908517350157727, 0.38235294117647056, 'x[0] <= 0.405\ngi
ni = 0.375\nsamples = 4\nvalue = [3, 1]'),
Text(0.3627760252365931, 0.3235294117647059, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.3753943217665615, 0.3235294117647059, 'gini = 0.0\nsamples
= 3\nvalue = [3, 0]'),

```

```

Text(0.38801261829652994, 0.4411764705882353, 'x[1] <= 0.75\ngini
= 0.444\nsamples = 3\nvalue = [2, 1]'),
Text(0.38170347003154576, 0.38235294117647056, 'gini = 0.0\nsampl
es = 2\nvalue = [2, 0]'),
Text(0.3943217665615142, 0.38235294117647056, 'gini = 0.0\nsampl
es = 1\nvalue = [0, 1]'),
Text(0.5224763406940063, 0.6764705882352942, 'x[17] <= 0.299\ngin
i = 0.1\nsamples = 493\nvalue = [467, 26]'),
Text(0.43296529968454256, 0.6176470588235294, 'x[32] <= 0.9\ngini
= 0.06\nsamples = 291\nvalue = [282, 9]'),
Text(0.42665615141955837, 0.5588235294117647, 'x[8] <= 0.016\ngin
i = 0.054\nsamples = 290\nvalue = [282, 8]'),
Text(0.4069400630914827, 0.5, 'x[24] <= 0.5\ngini = 0.444\nsampl
es = 3\nvalue = [2, 1]'),
Text(0.40063091482649843, 0.4411764705882353, 'gini = 0.0\nsampl
es = 1\nvalue = [0, 1]'),
Text(0.41324921135646686, 0.4411764705882353, 'gini = 0.0\nsampl
es = 2\nvalue = [2, 0]'),
Text(0.44637223974763407, 0.5, 'x[17] <= 0.056\ngini = 0.048\nsam
ples = 287\nvalue = [280, 7]'),
Text(0.42586750788643535, 0.4411764705882353, 'x[17] <= 0.054\ngi
ni = 0.32\nsamples = 5\nvalue = [4, 1]'),
Text(0.4195583596214511, 0.38235294117647056, 'gini = 0.0\nsampl
es = 4\nvalue = [4, 0]'),
Text(0.43217665615141954, 0.38235294117647056, 'gini = 0.0\nsampl
es = 1\nvalue = [0, 1]'),
Text(0.4668769716088328, 0.4411764705882353, 'x[15] <= 0.5\ngini
= 0.042\nsamples = 282\nvalue = [276, 6]'),
Text(0.444794952681388, 0.38235294117647056, 'x[2] <= 0.34\ngini
= 0.092\nsamples = 104\nvalue = [99, 5]'),
Text(0.43217665615141954, 0.3235294117647059, 'x[22] <= 0.75\ngin
i = 0.219\nsamples = 32\nvalue = [28, 4]'),
Text(0.42586750788643535, 0.2647058823529412, 'x[19] <= 0.722\ngi
ni = 0.175\nsamples = 31\nvalue = [28, 3]'),
Text(0.4195583596214511, 0.20588235294117646, 'x[31] <= 0.083\ngi
ni = 0.124\nsamples = 30\nvalue = [28, 2]'),
Text(0.4069400630914827, 0.14705882352941177, 'x[11] <= 0.45\ngin
i = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.40063091482649843, 0.08823529411764706, 'gini = 0.0\nsampl
es = 1\nvalue = [1, 0]'),
Text(0.41324921135646686, 0.08823529411764706, 'gini = 0.0\nsampl
es = 1\nvalue = [0, 1]'),
Text(0.43217665615141954, 0.14705882352941177, 'x[4] <= 0.946\ngi
ni = 0.069\nsamples = 28\nvalue = [27, 1]'),
Text(0.42586750788643535, 0.08823529411764706, 'gini = 0.0\nsampl
es = 25\nvalue = [25, 0]'),
Text(0.4384858044164038, 0.08823529411764706, 'x[19] <= 0.056\ngi
ni = 0.444\nsamples = 3\nvalue = [2, 1]'),
Text(0.43217665615141954, 0.029411764705882353, 'gini = 0.0\nsamp
les = 1\nvalue = [0, 1]'),
Text(0.444794952681388, 0.029411764705882353, 'gini = 0.0\nsampl
es = 2\nvalue = [2, 0]'),
Text(0.43217665615141954, 0.20588235294117646, 'gini = 0.0\nsampl

```

```

es = 1\nvalue = [0, 1]'),
Text(0.4384858044164038, 0.2647058823529412, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.45741324921135645, 0.3235294117647059, 'x[22] <= 0.036\ngi
ni = 0.027\nsamples = 72\nvalue = [71, 1]'),
Text(0.45110410094637227, 0.2647058823529412, 'x[0] <= 0.44\ngini
= 0.198\nsamples = 9\nvalue = [8, 1]'),
Text(0.444794952681388, 0.20588235294117646, 'gini = 0.0\nsamples
= 7\nvalue = [7, 0]'),
Text(0.45741324921135645, 0.20588235294117646, 'x[0] <= 0.5\ngini
= 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.45110410094637227, 0.14705882352941177, 'gini = 0.0\nsampl
es = 1\nvalue = [0, 1]'),
Text(0.4637223974763407, 0.14705882352941177, 'gini = 0.0\nsampl
es = 1\nvalue = [1, 0]'),
Text(0.4637223974763407, 0.2647058823529412, 'gini = 0.0\nsamples
= 63\nvalue = [63, 0]'),
Text(0.4889589905362776, 0.38235294117647056, 'x[33] <= 0.088\ngi
ni = 0.011\nsamples = 178\nvalue = [177, 1]'),
Text(0.48264984227129337, 0.3235294117647059, 'x[30] <= 0.633\ngi
ni = 0.133\nsamples = 14\nvalue = [13, 1]'),
Text(0.47634069400630913, 0.2647058823529412, 'gini = 0.0\nsampl
es = 12\nvalue = [12, 0]'),
Text(0.4889589905362776, 0.2647058823529412, 'x[8] <= 0.611\ngini
= 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.48264984227129337, 0.20588235294117646, 'gini = 0.0\nsampl
es = 1\nvalue = [1, 0]'),
Text(0.4952681388012618, 0.20588235294117646, 'gini = 0.0\nsampl
es = 1\nvalue = [0, 1]'),
Text(0.4952681388012618, 0.3235294117647059, 'gini = 0.0\nsamples
= 164\nvalue = [164, 0]'),
Text(0.4392744479495268, 0.5588235294117647, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.61198738170347, 0.6176470588235294, 'x[17] <= 0.5\ngini =
0.154\nsamples = 202\nvalue = [185, 17]'),
Text(0.5772870662460567, 0.5588235294117647, 'x[0] <= 0.536\ngini
= 0.264\nsamples = 96\nvalue = [81, 15]'),
Text(0.5520504731861199, 0.5, 'x[4] <= 0.804\ngini = 0.182\nsampl
es = 69\nvalue = [62, 7]'),
Text(0.5394321766561514, 0.4411764705882353, 'x[18] <= 0.952\ngin
i = 0.121\nsamples = 62\nvalue = [58, 4]'),
Text(0.5331230283911672, 0.38235294117647056, 'x[11] <= 0.971\ngi
ni = 0.094\nsamples = 61\nvalue = [58, 3]'),
Text(0.526813880126183, 0.3235294117647059, 'x[33] <= 0.029\ngini
= 0.064\nsamples = 60\nvalue = [58, 2]'),
Text(0.5141955835962145, 0.2647058823529412, 'x[26] <= 0.333\ngin
i = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.5078864353312302, 0.20588235294117646, 'gini = 0.0\nsampl
es = 1\nvalue = [0, 1]'),
Text(0.5205047318611987, 0.20588235294117646, 'gini = 0.0\nsampl
es = 1\nvalue = [1, 0]'),
Text(0.5394321766561514, 0.2647058823529412, 'x[28] <= 0.583\ngin
i = 0.034\nsamples = 58\nvalue = [57, 1]'),

```

```

Text(0.5331230283911672, 0.20588235294117646, 'gini = 0.0\nsample
s = 49\nvalue = [49, 0]'),
Text(0.5457413249211357, 0.20588235294117646, 'x[32] <= 0.3\ngini
= 0.198\nsamples = 9\nvalue = [8, 1]'),
Text(0.5394321766561514, 0.14705882352941177, 'gini = 0.0\nsample
s = 8\nvalue = [8, 0]'),
Text(0.5520504731861199, 0.14705882352941177, 'gini = 0.0\nsample
s = 1\nvalue = [0, 1]'),
Text(0.5394321766561514, 0.3235294117647059, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.5457413249211357, 0.38235294117647056, 'gini = 0.0\nsample
s = 1\nvalue = [0, 1]'),
Text(0.5646687697160884, 0.4411764705882353, 'x[0] <= 0.298\ngini
= 0.49\nsamples = 7\nvalue = [4, 3]'),
Text(0.5583596214511041, 0.38235294117647056, 'gini = 0.0\nsample
s = 3\nvalue = [0, 3]'),
Text(0.5709779179810726, 0.38235294117647056, 'gini = 0.0\nsample
s = 4\nvalue = [4, 0]'),
Text(0.6025236593059937, 0.5, 'x[4] <= 0.268\ngini = 0.417\nsampl
es = 27\nvalue = [19, 8]'),
Text(0.5899053627760252, 0.4411764705882353, 'x[27] <= 0.212\ngin
i = 0.255\nsamples = 20\nvalue = [17, 3]'),
Text(0.583596214511041, 0.38235294117647056, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.5962145110410094, 0.38235294117647056, 'x[5] <= 0.875\ngin
i = 0.188\nsamples = 19\nvalue = [17, 2]'),
Text(0.5899053627760252, 0.3235294117647059, 'x[19] <= 0.889\ngin
i = 0.105\nsamples = 18\nvalue = [17, 1]'),
Text(0.583596214511041, 0.2647058823529412, 'gini = 0.0\nsamples
= 17\nvalue = [17, 0]'),
Text(0.5962145110410094, 0.2647058823529412, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.6025236593059937, 0.3235294117647059, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.6151419558359621, 0.4411764705882353, 'x[18] <= 0.091\ngin
i = 0.408\nsamples = 7\nvalue = [2, 5]'),
Text(0.6088328075709779, 0.38235294117647056, 'gini = 0.0\nsample
s = 2\nvalue = [2, 0]'),
Text(0.6214511041009464, 0.38235294117647056, 'gini = 0.0\nsample
s = 5\nvalue = [0, 5]'),
Text(0.6466876971608833, 0.5588235294117647, 'x[12] <= 0.167\ngin
i = 0.037\nsamples = 106\nvalue = [104, 2]'),
Text(0.6340694006309149, 0.5, 'x[9] <= 0.167\ngini = 0.444\nsampl
es = 3\nvalue = [2, 1]'),
Text(0.6277602523659306, 0.4411764705882353, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.6403785488958991, 0.4411764705882353, 'gini = 0.0\nsamples
= 2\nvalue = [2, 0]'),
Text(0.6593059936908517, 0.5, 'x[26] <= 0.833\ngini = 0.019\nsamp
les = 103\nvalue = [102, 1]'),
Text(0.6529968454258676, 0.4411764705882353, 'gini = 0.0\nsamples
= 99\nvalue = [99, 0]'),
Text(0.6656151419558359, 0.4411764705882353, 'x[2] <= 0.152\ngini

```

```

= 0.375\nsamples = 4\nvalue = [3, 1]'),
Text(0.6593059936908517, 0.38235294117647056, 'gini = 0.0\nsample
s = 1\nvalue = [0, 1]'),
Text(0.6719242902208202, 0.38235294117647056, 'gini = 0.0\nsample
s = 3\nvalue = [3, 0]'),
Text(0.41324921135646686, 0.7352941176470589, 'gini = 0.0\nsample
s = 1\nvalue = [0, 1]'),
Text(0.832807570977918, 0.8529411764705882, 'x[17] <= 0.157\ngini
= 0.385\nsamples = 300\nvalue = [222, 78]'),
Text(0.75, 0.7941176470588235, 'x[26] <= 0.167\ngini = 0.5\nsampl
es = 96\nvalue = [49, 47]'),
Text(0.7160883280757098, 0.7352941176470589, 'x[4] <= 0.161\ngini
= 0.459\nsamples = 42\nvalue = [15, 27]'),
Text(0.6908517350157729, 0.6764705882352942, 'x[8] <= 0.415\ngini
= 0.499\nsamples = 23\nvalue = [12, 11]'),
Text(0.6719242902208202, 0.6176470588235294, 'x[18] <= 0.561\ngin
i = 0.355\nsamples = 13\nvalue = [3, 10]'),
Text(0.6656151419558359, 0.5588235294117647, 'gini = 0.0\nsamples
= 8\nvalue = [0, 8]'),
Text(0.6782334384858044, 0.5588235294117647, 'x[28] <= 0.583\ngin
i = 0.48\nsamples = 5\nvalue = [3, 2]'),
Text(0.6719242902208202, 0.5, 'gini = 0.0\nsamples = 3\nvalue = [
3, 0]'),
Text(0.6845425867507886, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [
0, 2]'),
Text(0.7097791798107256, 0.6176470588235294, 'x[24] <= 0.167\ngin
i = 0.18\nsamples = 10\nvalue = [9, 1]'),
Text(0.7034700315457413, 0.5588235294117647, 'x[19] <= 0.167\ngin
i = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.6971608832807571, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [
0, 1]'),
Text(0.7097791798107256, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [
1, 0]'),
Text(0.7160883280757098, 0.5588235294117647, 'gini = 0.0\nsamples
= 8\nvalue = [8, 0]'),
Text(0.7413249211356467, 0.6764705882352942, 'x[13] <= 0.125\ngin
i = 0.266\nsamples = 19\nvalue = [3, 16]'),
Text(0.7350157728706624, 0.6176470588235294, 'x[11] <= 0.2\ngini
= 0.198\nsamples = 18\nvalue = [2, 16]'),
Text(0.7287066246056783, 0.5588235294117647, 'gini = 0.0\nsamples
= 1\nvalue = [1, 0]'),
Text(0.7413249211356467, 0.5588235294117647, 'x[0] <= 0.595\ngini
= 0.111\nsamples = 17\nvalue = [1, 16]'),
Text(0.7350157728706624, 0.5, 'gini = 0.0\nsamples = 15\nvalue =
[0, 15]'),
Text(0.7476340694006309, 0.5, 'x[24] <= 0.833\ngini = 0.5\nsampl
es = 2\nvalue = [1, 1]'),
Text(0.7413249211356467, 0.4411764705882353, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.7539432176656151, 0.4411764705882353, 'gini = 0.0\nsamples
= 1\nvalue = [1, 0]'),
Text(0.7476340694006309, 0.6176470588235294, 'gini = 0.0\nsamples
= 1\nvalue = [1, 0]'),

```

```

Text(0.7839116719242902, 0.7352941176470589, 'x[0] <= 0.202\ngini
= 0.466\nsamples = 54\nvalue = [34, 20]'),
Text(0.7665615141955836, 0.6764705882352942, 'x[0] <= 0.107\ngini
= 0.245\nsamples = 7\nvalue = [1, 6]'),
Text(0.7602523659305994, 0.6176470588235294, 'gini = 0.0\nsamples
= 1\nvalue = [1, 0]'),
Text(0.7728706624605678, 0.6176470588235294, 'gini = 0.0\nsamples
= 6\nvalue = [0, 6]'),
Text(0.8012618296529969, 0.6764705882352942, 'x[2] <= 0.622\ngini
= 0.418\nsamples = 47\nvalue = [33, 14]'),
Text(0.7854889589905363, 0.6176470588235294, 'x[2] <= 0.145\ngini
= 0.482\nsamples = 32\nvalue = [19, 13]'),
Text(0.7728706624605678, 0.5588235294117647, 'x[30] <= 0.633\ngin
i = 0.18\nsamples = 10\nvalue = [9, 1]'),
Text(0.7665615141955836, 0.5, 'gini = 0.0\nsamples = 9\nvalue = [
9, 0]'),
Text(0.7791798107255521, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [
0, 1]'),
Text(0.7981072555205048, 0.5588235294117647, 'x[18] <= 0.87\ngini
= 0.496\nsamples = 22\nvalue = [10, 12]'),
Text(0.7917981072555205, 0.5, 'x[8] <= 0.41\ngini = 0.465\nsampl
e s = 19\nvalue = [7, 12]'),
Text(0.7791798107255521, 0.4411764705882353, 'x[18] <= 0.715\ngin
i = 0.469\nsamples = 8\nvalue = [5, 3]'),
Text(0.7728706624605678, 0.38235294117647056, 'gini = 0.0\nsampl
e s = 5\nvalue = [5, 0]'),
Text(0.7854889589905363, 0.38235294117647056, 'gini = 0.0\nsampl
e s = 3\nvalue = [0, 3]'),
Text(0.804416403785489, 0.4411764705882353, 'x[0] <= 0.25\ngini =
0.298\nsamples = 11\nvalue = [2, 9]'),
Text(0.7981072555205048, 0.38235294117647056, 'gini = 0.0\nsampl
e s = 1\nvalue = [1, 0]'),
Text(0.8107255520504731, 0.38235294117647056, 'x[3] <= 0.25\ngini
= 0.18\nsamples = 10\nvalue = [1, 9]'),
Text(0.804416403785489, 0.3235294117647059, 'x[6] <= 0.1\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.7981072555205048, 0.2647058823529412, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.8107255520504731, 0.2647058823529412, 'gini = 0.0\nsamples
= 1\nvalue = [1, 0]'),
Text(0.8170347003154574, 0.3235294117647059, 'gini = 0.0\nsamples
= 8\nvalue = [0, 8]'),
Text(0.804416403785489, 0.5, 'gini = 0.0\nsamples = 3\nvalue = [3
, 0]'),
Text(0.8170347003154574, 0.6176470588235294, 'x[19] <= 0.944\ngin
i = 0.124\nsamples = 15\nvalue = [14, 1]'),
Text(0.8107255520504731, 0.5588235294117647, 'gini = 0.0\nsamples
= 14\nvalue = [14, 0]'),
Text(0.8233438485804416, 0.5588235294117647, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.9156151419558359, 0.7941176470588235, 'x[16] <= 0.75\ngini
= 0.258\nsamples = 204\nvalue = [173, 31]'),
Text(0.8659305993690851, 0.7352941176470589, 'x[17] <= 0.992\ngin

```



```

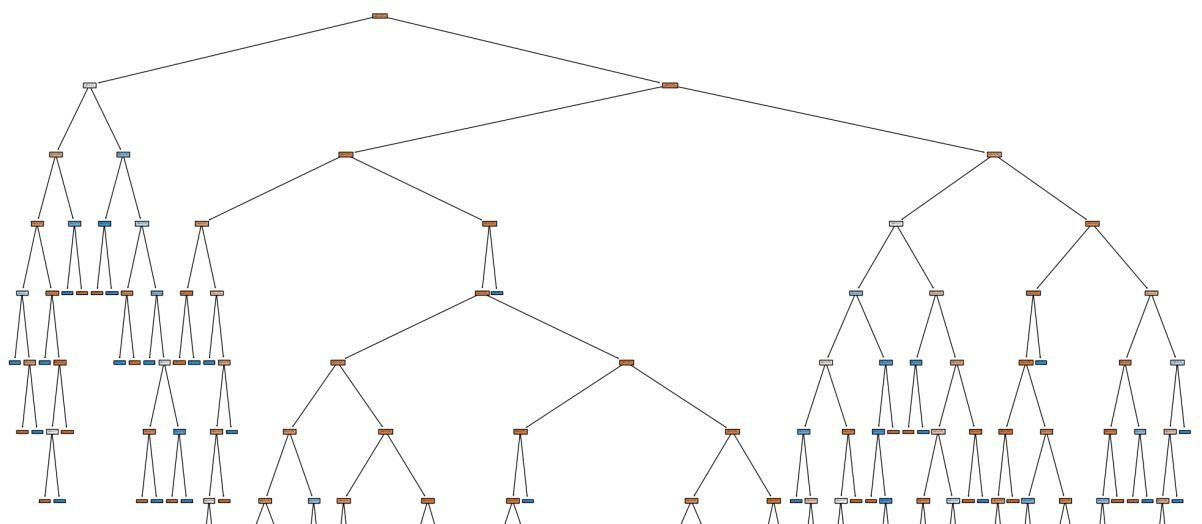
i = 0.138\nsamples = 147\nvalue = [136, 11]'),
Text(0.8596214511041009, 0.6764705882352942, 'x[4] <= 0.482\ngini
= 0.128\nsamples = 146\nvalue = [136, 10]'),
Text(0.8422712933753943, 0.6176470588235294, 'x[30] <= 0.167\ngini
i = 0.038\nsamples = 104\nvalue = [102, 2]'),
Text(0.8359621451104101, 0.5588235294117647, 'x[11] <= 0.193\ngini
i = 0.32\nsamples = 10\nvalue = [8, 2]'),
Text(0.8296529968454258, 0.5, 'x[27] <= 0.475\ngini = 0.444\nsamples
= 3\nvalue = [1, 2]'),
Text(0.8233438485804416, 0.4411764705882353, 'gini = 0.0\nsamples
= 2\nvalue = [0, 2]'),
Text(0.8359621451104101, 0.4411764705882353, 'gini = 0.0\nsamples
= 1\nvalue = [1, 0]'),
Text(0.8422712933753943, 0.5, 'gini = 0.0\nsamples = 7\nvalue = [
7, 0]'),
Text(0.8485804416403786, 0.5588235294117647, 'gini = 0.0\nsamples
= 94\nvalue = [94, 0]'),
Text(0.8769716088328076, 0.6176470588235294, 'x[9] <= 0.167\ngini
= 0.308\nsamples = 42\nvalue = [34, 8]'),
Text(0.861198738170347, 0.5588235294117647, 'x[29] <= 0.833\ngini
= 0.375\nsamples = 4\nvalue = [1, 3]'),
Text(0.8548895899053628, 0.5, 'gini = 0.0\nsamples = 3\nvalue = [
0, 3]'),
Text(0.8675078864353313, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [
1, 0]'),
Text(0.8927444794952681, 0.5588235294117647, 'x[0] <= 0.393\ngini
= 0.229\nsamples = 38\nvalue = [33, 5]'),
Text(0.8801261829652997, 0.5, 'x[9] <= 0.5\ngini = 0.5\nsamples =
6\nvalue = [3, 3]'),
Text(0.8738170347003155, 0.4411764705882353, 'x[11] <= 0.643\ngini
i = 0.375\nsamples = 4\nvalue = [1, 3]'),
Text(0.8675078864353313, 0.38235294117647056, 'gini = 0.0\nsamples
= 3\nvalue = [0, 3]'),
Text(0.8801261829652997, 0.38235294117647056, 'gini = 0.0\nsamples
= 1\nvalue = [1, 0]'),
Text(0.886435331230284, 0.4411764705882353, 'gini = 0.0\nsamples
= 2\nvalue = [2, 0]'),
Text(0.9053627760252366, 0.5, 'x[28] <= 0.917\ngini = 0.117\nsamples
= 32\nvalue = [30, 2]'),
Text(0.8990536277602523, 0.4411764705882353, 'x[8] <= 0.992\ngini
= 0.062\nsamples = 31\nvalue = [30, 1]'),
Text(0.8927444794952681, 0.38235294117647056, 'gini = 0.0\nsamples
= 30\nvalue = [30, 0]'),
Text(0.9053627760252366, 0.38235294117647056, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.9116719242902208, 0.4411764705882353, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.8722397476340694, 0.6764705882352942, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
Text(0.9652996845425867, 0.7352941176470589, 'x[14] <= 0.812\ngini
i = 0.456\nsamples = 57\nvalue = [37, 20]'),
Text(0.943217665615142, 0.6764705882352942, 'x[32] <= 0.4\ngini =
0.238\nsamples = 29\nvalue = [25, 4]'),

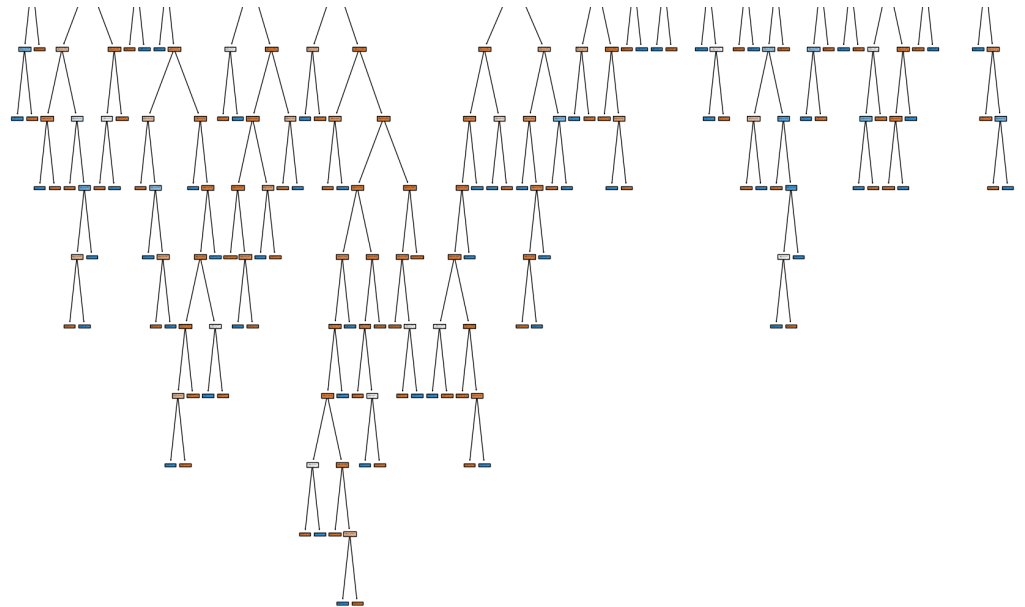
```

```

Text(0.9305993690851735, 0.6176470588235294, 'x[8] <= 0.071\ngini
= 0.142\nsamples = 26\nvalue = [24, 2]'),
Text(0.9242902208201893, 0.5588235294117647, 'x[19] <= 0.056\ngini
i = 0.444\nsamples = 3\nvalue = [1, 2]'),
Text(0.917981072555205, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1
, 0]'),
Text(0.9305993690851735, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [
0, 2]'),
Text(0.9369085173501577, 0.5588235294117647, 'gini = 0.0\nsamples
= 23\nvalue = [23, 0]'),
Text(0.9558359621451105, 0.6176470588235294, 'x[1] <= 0.25\ngini
= 0.444\nsamples = 3\nvalue = [1, 2]'),
Text(0.9495268138801262, 0.5588235294117647, 'gini = 0.0\nsamples
= 1\nvalue = [1, 0]'),
Text(0.9621451104100947, 0.5588235294117647, 'gini = 0.0\nsamples
= 2\nvalue = [0, 2]'),
Text(0.9873817034700315, 0.6764705882352942, 'x[32] <= 0.1\ngini
= 0.49\nsamples = 28\nvalue = [12, 16]'),
Text(0.9810725552050473, 0.6176470588235294, 'x[12] <= 0.833\ngini
i = 0.48\nsamples = 20\nvalue = [12, 8]'),
Text(0.9747634069400631, 0.5588235294117647, 'x[4] <= 0.018\ngini
= 0.415\nsamples = 17\nvalue = [12, 5]'),
Text(0.9684542586750788, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [
0, 2]'),
Text(0.9810725552050473, 0.5, 'x[17] <= 0.365\ngini = 0.32\nsampl
es = 15\nvalue = [12, 3]'),
Text(0.9747634069400631, 0.4411764705882353, 'gini = 0.0\nsamples
= 11\nvalue = [11, 0]'),
Text(0.9873817034700315, 0.4411764705882353, 'x[4] <= 0.179\ngini
= 0.375\nsamples = 4\nvalue = [1, 3]'),
Text(0.9810725552050473, 0.38235294117647056, 'gini = 0.0\nsampl
es = 1\nvalue = [1, 0]'),
Text(0.9936908517350158, 0.38235294117647056, 'gini = 0.0\nsampl
es = 3\nvalue = [0, 3]'),
Text(0.9873817034700315, 0.5588235294117647, 'gini = 0.0\nsamples
= 3\nvalue = [0, 3]'),
Text(0.9936908517350158, 0.6176470588235294, 'gini = 0.0\nsamples
= 8\nvalue = [0, 8]')]

```





```
In [105]: from sklearn.model_selection import GridSearchCV
```

```
In [108]: parameter = {
    'criterion': ['gini', 'entropy'],
    'splitter' : ['best', 'random'],
    'max_depth': [1, 2, 3, 4, 5],
    'max_features': ['auto', 'sqrt', 'log2']
}
```

```
In [109]: grid_search = GridSearchCV(estimator=dt,param_grid=parameter,cv=5,s
    grid_search.fit(x_train,y_train)
```

/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.

warnings.warn(

/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.

warnings.warn(

/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.

warnings.warn(

/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.

warnings.warn(

/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.

```

o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(

```

Page 37 of 49


```

es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(

```

Page 40 of 49


```

o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o`` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
warnings.warn(

```

```

/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o'` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o'` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o'` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o'` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o'` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o'` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o'` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o'` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o'` has been deprecated in 1.1 and will be removed in 1.3. To keep
the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packag
es/sklearn/tree/_classes.py:269: FutureWarning: `max_features='aut
o'` has been deprecated in 1.1 and will be removed in 1.3. To keep

```

Page 43 of 49

Page 44 of 49

```
warnings.warn(
```

Out[109]:

```
GridSearchCV
estimator: DecisionTreeClassifier
DecisionTreeClassifier
```

In [110]: `grid_search.best_params_`

```
Out[110]: {'criterion': 'entropy',
           'max_depth': 3,
           'max_features': 'log2',
           'splitter': 'random'}
```

```
In [111]: dt_cv = DecisionTreeClassifier(criterion='entropy',max_depth=3,max_
dt_cv.fit(x_train,y_train)
```

```
Out[111]: DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=3, max_feat
ures='sqrt')
```

In [112]: `pred = dt_cv.predict(x_test)`In [113]: `print(classification_report(y_test,pred))`

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.83 | 0.98 | 0.90 | 245 |
| 1 | 0.20 | 0.02 | 0.04 | 49 |
| accuracy | | | 0.82 | 294 |
| macro avg | 0.52 | 0.50 | 0.47 | 294 |
| weighted avg | 0.73 | 0.82 | 0.76 | 294 |

random forest

```
In [114]: from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
```

```
In [115]: forest_params = [{'max_depth':list(range(10,15)),'max_features': li
```

In [117]:

```
rf_cv = GridSearchCV(rf,param_grid = forest_params,cv = 10,scoring
rf_cv.fit(x_train,y_train)
```

/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/model_selection/_validation.py:378: 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='raise'.

Below are more details about the failures:

50 fits failed with the following error:

Traceback (most recent call last):

File "/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/model_selection/_validation.py", line 686, in _fit_and_score

estimator.fit(X_train, y_train, **fit_params)

File "/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/ensemble/_forest.py", line 340, in fit

self._validate_params()

File "/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/base.py", line 600, in _validate_params

validate_parameter_constraints(

File "/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/utils/_param_validation.py", line 97, in validate_parameter_constraints

raise InvalidParameterError(

sklearn.utils._param_validation.InvalidParameterError: The 'max_features' parameter of RandomForestClassifier must be an int in the range [1, inf), a float in the range (0.0, 1.0], a str among {'log 2', 'sqrt', 'auto' (deprecated)} or None. Got 0 instead.

warnings.warn(some_fits_failed_message, FitFailedWarning)

/Users/dhanarahulsainadiminti/anaconda3/lib/python3.11/site-packages/sklearn/model_selection/_search.py:952: UserWarning: One or more of the test scores are non-finite: [nan 0.84610314 0.85373026 0.85713458 0.8596842 0.86308851

0.85714182 0.86136462 0.86137187 0.85627264 0.85969144 0.86393597

0.85796755 0.85883674 nan 0.85205708 0.85375924 0.85629436

0.85969144 0.85713458 0.86138635 0.8605389 0.8622483 0.86052441

0.86221932 0.86221932 0.86053165 0.85625815 nan 0.84779806

0.85118065 0.85969868 0.85713458 0.85798928 0.86053165 0.86223381

0.8605389 0.8596842 0.86137911 0.85881501 0.85712009 0.85965522

nan 0.85036216 0.85885847 0.85714182 0.85798204 0.85798928

0.85627264 0.85712734 0.85969868 0.86051717 0.86138635 0.85542518

0.86307403 0.85714182 nan 0.85119513 0.85373751 0.85459221

0.86224105 0.85882225 0.86394321 0.85798928 0.85542518 0.86051717

0.86053165 0.8596842 0.86221208 0.86136462]

warnings.warn(

Out[117]:

GridSearchCV

```

> estimator: RandomForestClassifier
  ▶ RandomForestClassifier

```

```
In [118]: pred = rf_cv.predict(x_test)
```

```
In [119]: accuracy_score(y_test, pred)
```

```
Out[119]: 0.8469387755102041
```

```
In [120]: 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 [121]: rf_cv.best_params_
```

```
Out[121]: {'max_depth': 14, 'max_features': 6}
```

```
In [123]: rf.fit(x_train, y_train)
```

```
Out[123]:
  ▼ RandomForestClassifier
  RandomForestClassifier()
```

```
In [125]: probabo = rf.predict_proba(x_test)[: , 1]
probabo
```

```
Out[125]: array([0.08, 0.08, 0.18, 0.19, 0.7 , 0.36, 0.38, 0.16, 0.08, 0.22,
0.05,
0.12, 0.1 , 0.58, 0.07, 0. , 0.06, 0.21, 0.09, 0.18, 0.5 ,
0.04,
0.01, 0.07, 0.39, 0.2 , 0.04, 0.08, 0.65, 0.04, 0.07, 0.07,
0.28,
0.1 , 0.13, 0.05, 0.18, 0.11, 0.04, 0.14, 0.14, 0.14, 0.08,
0.12,
0.1 , 0.33, 0.41, 0.06, 0.7 , 0.4 , 0.15, 0.47, 0.19, 0.11,
0.5 ,
0.18, 0.07, 0.04, 0.07, 0.34, 0.1 , 0.14, 0.1 , 0.15, 0.37,
0.1 ,
0.27, 0.11, 0.15, 0.21, 0.09, 0.39, 0.12, 0.02, 0.07, 0.14,
0.14,
```

```

0.12, 0.45, 0.05, 0.04, 0.07, 0.11, 0.14, 0.14, 0.06, 0.09,
0.28,
0.07, 0.06, 0.66, 0.11, 0.07, 0.26, 0.08, 0.1 , 0.09, 0.34,
0.09,
0.15, 0.27, 0.35, 0.22, 0.09, 0.08, 0.07, 0.09, 0.55, 0.25,
0.27,
0.2 , 0.12, 0.12, 0.05, 0.13, 0.1 , 0.04, 0.09, 0.05, 0.03,
0.12,
0.06, 0.05, 0.59, 0.23, 0.14, 0.04, 0.07, 0.21, 0.09, 0.05,
0.3 ,
0.47, 0.18, 0.24, 0.09, 0.27, 0.15, 0.17, 0.19, 0.07, 0.19,
0.07,
0.17, 0.2 , 0.12, 0.12, 0.07, 0.09, 0.07, 0.13, 0.16, 0.07,
0.24,
0.16, 0.15, 0.06, 0.14, 0.16, 0.09, 0.26, 0.02, 0.17, 0.56,
0.1 ,
0.16, 0.16, 0.09, 0.04, 0.04, 0.07, 0.1 , 0.09, 0.28, 0.11,
0.45,
0.11, 0.21, 0.19, 0.13, 0.02, 0.03, 0.03, 0.46, 0.04, 0.17,
0.32,
0.11, 0.18, 0.28, 0.3 , 0.57, 0.17, 0.03, 0.07, 0.09, 0.15,
0.05,
0.38, 0.09, 0.2 , 0.1 , 0.25, 0.04, 0.16, 0.16, 0.41, 0.08,
0.16,
0.01, 0.1 , 0.19, 0.06, 0.16, 0.05, 0.08, 0.06, 0.24, 0.17,
0.4 ,
0.13, 0.3 , 0.46, 0.18, 0.18, 0.05, 0.09, 0.31, 0.55, 0.15,
0. ,
0.35, 0.04, 0.11, 0.01, 0.3 , 0.3 , 0.02, 0.16, 0.11, 0.24,
0.07,
0.06, 0.09, 0.14, 0.1 , 0.09, 0.23, 0.07, 0.18, 0.22, 0.16,
0.35,
0.12, 0.2 , 0.1 , 0.03, 0.64, 0.27, 0.36, 0.13, 0.1 , 0.16,
0.04,
0.13, 0.22, 0.04, 0.29, 0.2 , 0.04, 0.06, 0.06, 0.13, 0.09,
0.04,
0.11, 0.11, 0.39, 0.15, 0.3 , 0.3 , 0.12, 0.16, 0.18, 0.13,
0.17,
0.02, 0.13, 0.12, 0.01, 0.18, 0.25, 0.12, 0.1 ] )

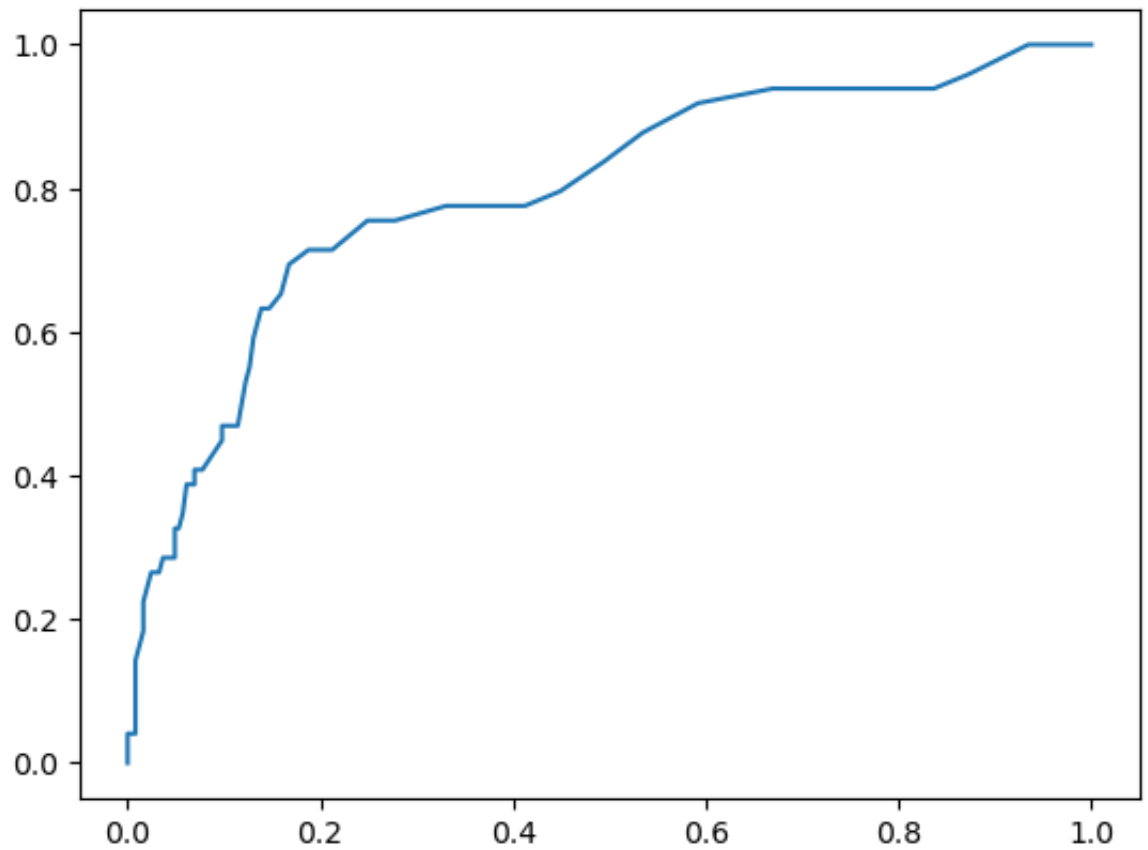
```

```
In [126]: fpr, tpr, threshholds = roc_curve(y_test, probabo)
```



```
In [127]: plt.plot(fpr,tpr)
```

```
Out[127]: [<matplotlib.lines.Line2D at 0x138d25410>]
```



```
In [ ]:
```