# Data Preprocessing

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

#importing dataset
df=pd.read_csv("Employee-Attrition.csv")

df.head()
```

```
   Age Attrition     BusinessTravel  DailyRate                Department
\
0   41       Yes      Travel_Rarely       1102                     Sales

1   49        No  Travel_Frequently        279  Research & Development

2   37       Yes      Travel_Rarely       1373  Research & Development

3   33        No  Travel_Frequently       1392  Research & Development

4   27        No      Travel_Rarely        591  Research & Development


   DistanceFromHome  Education EducationField  EmployeeCount
EmployeeNumber  \
0                 1          2  Life Sciences              1
1
1                 8          1  Life Sciences              1
2
2                 2          2          Other              1
4
3                 3          4  Life Sciences              1
5
4                 2          1        Medical              1
7

   ...  RelationshipSatisfaction StandardHours  StockOptionLevel  \
0  ...                         1            80                 0
1  ...                         4            80                 1
2  ...                         2            80                 0
3  ...                         3            80                 0
4  ...                         4            80                 1

   TotalWorkingYears  TrainingTimesLastYear WorkLifeBalance
YearsAtCompany  \
0                  8                      0               1
```

```
6
1                  10                         3                3
10
2                   7                         3                3
0
3                   8                         3                3
8
4                   6                         3                3
2

  YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0                  4                        0                     5
1                  7                        1                     7
2                  0                        0                     0
3                  7                        3                     0
4                  2                        2                     2

[5 rows x 35 columns]
```

```
df.shape
```

```
(1470, 35)
```

```
#checking NULL Values
df.isnull().any()
```

```
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
```

```
df.isnull().sum()
```

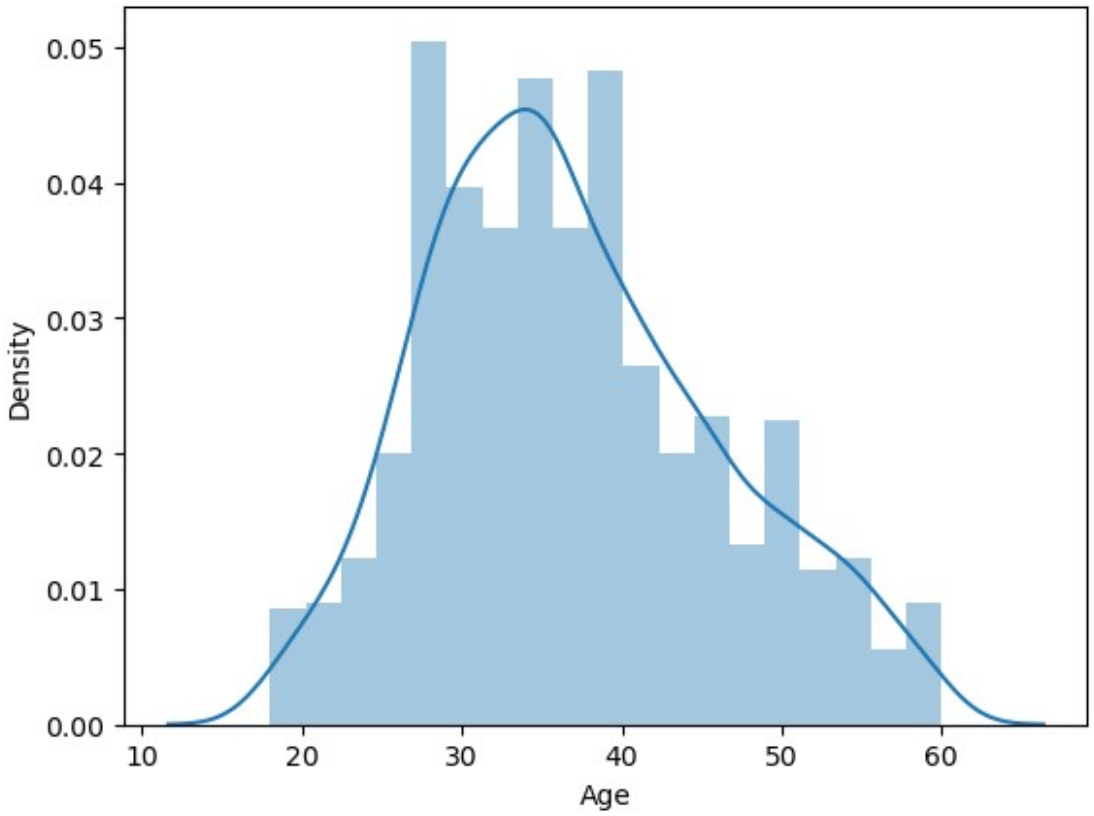
```
Age                          0
Attrition                    0
BusinessTravel               0
DailyRate                    0
Department                   0
DistanceFromHome             0
Education                    0
EducationField               0
EmployeeCount                0
EmployeeNumber               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
```

```
#Data Visualization.
sns.distplot(df["Age"])
```

```
/var/folders/75/l1625v4n7qnbfp296v2f82ch0000gn/T/
ipykernel_4203/2400079689.py:2: UserWarning:

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

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

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

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

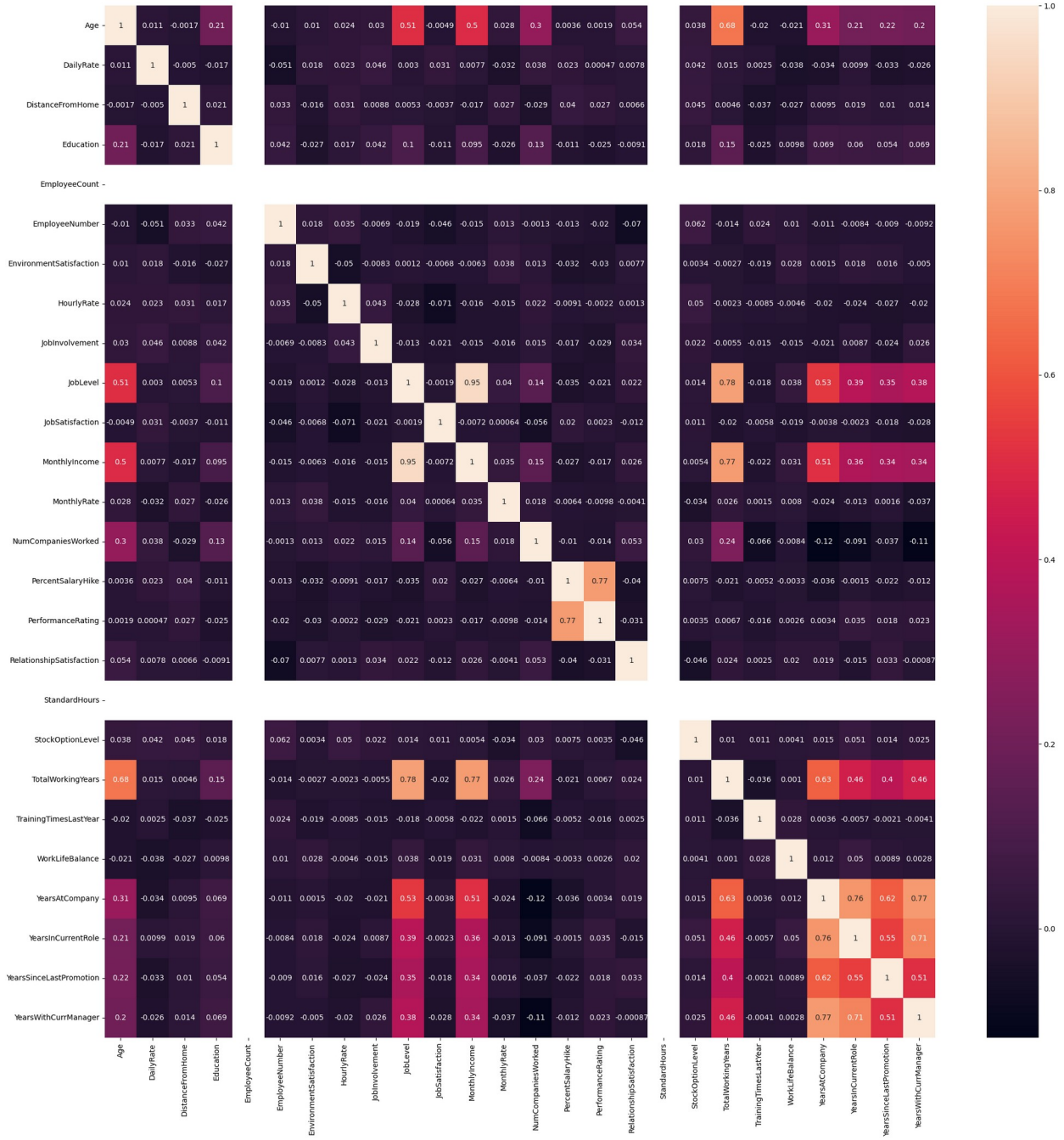
```
<Axes: xlabel='Age', ylabel='Density'>
```



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

```
/var/folders/75/l1625v4n7qnbfp296v2f82ch0000gn/T/
ipykernel_4203/2164201515.py:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it
will default to False. Select only valid columns or specify the value
of numeric_only to silence this warning.
  corr=df.corr()
```

```
<Axes: >
```

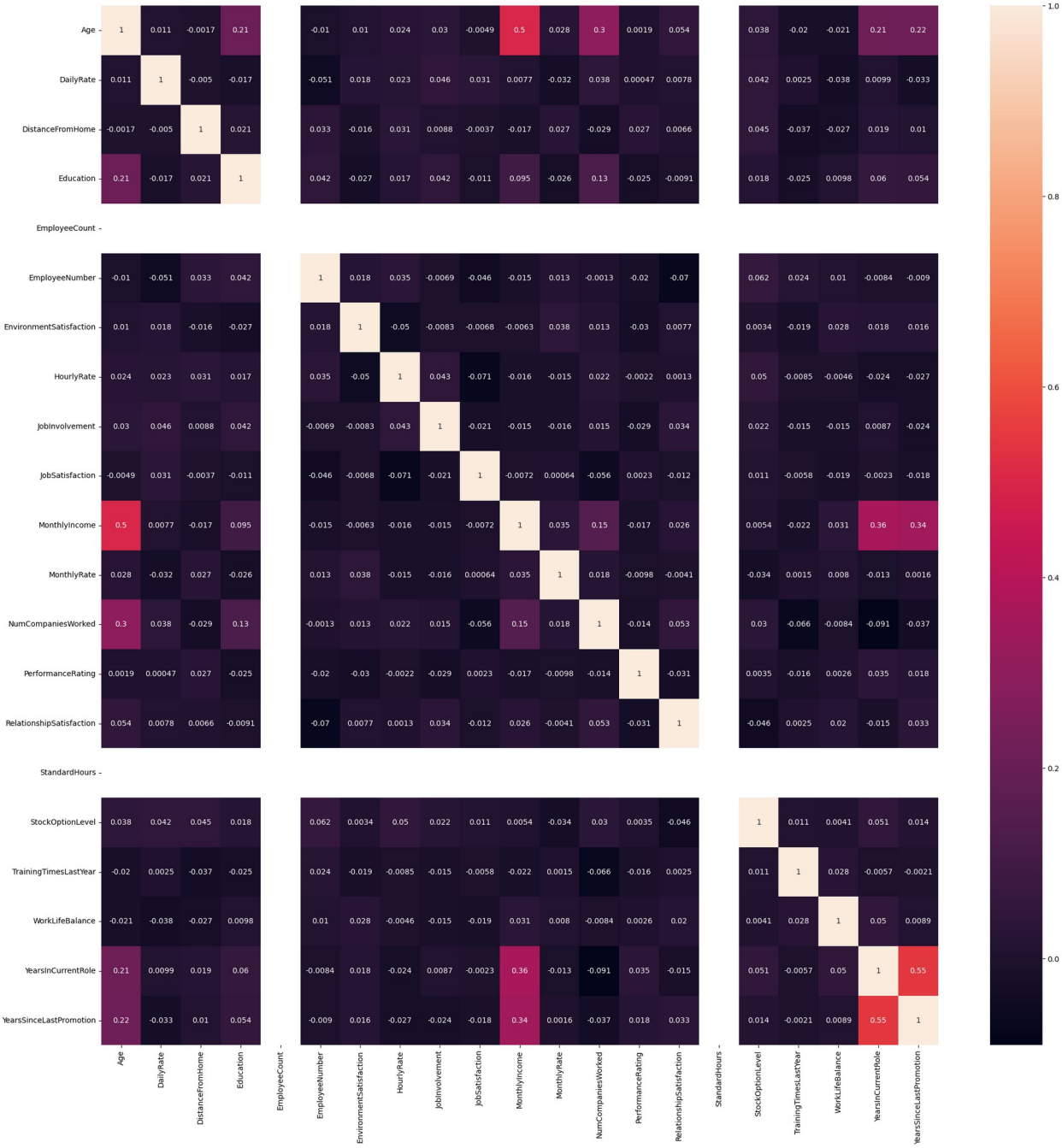



```
#removing one of two highly correlated columns
df.drop(['JobLevel', 'PercentSalaryHike', 'TotalWorkingYears',
'YearsAtCompany','YearsWithCurrManager'],axis=1,inplace=True)

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

```
/var/folders/75/l1625v4n7qnbfp296v2f82ch0000gn/T/
ipykernel_4203/2164201515.py:1: FutureWarning: The default value of
```

```
numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  corr=df.corr()
```

```
<Axes: >
```




```
df["EmployeeCount"].nunique()
```

```
1
```

```
df["Over18"].nunique()
```

```
1
```

```
df["PerformanceRating"].unique()
```

```
array([3, 4])
```

```
df["StandardHours"].nunique()
```

```
1
```

```
df["WorkLifeBalance"].unique()
```

```
array([1, 3, 2, 4])
```

```
df["YearsInCurrentRole"].unique()
```

```
array([ 4,  7,  0,  2,  5,  9,  8,  3,  6, 13,  1, 15, 14, 16, 11, 10,
12,
       18, 17])
```

```
df["YearsSinceLastPromotion"].unique()
```

```
array([ 0,  1,  3,  2,  7,  4,  8,  6,  5, 15,  9, 13, 12, 10, 11,
14])
```

```
df.drop(['EmployeeCount'],axis=1,inplace=True)
```

```
df.drop(['Over18'],axis=1,inplace=True)
```

```
df.drop(['StandardHours'],axis=1,inplace=True)
```

```
#outlier detection and removal
df.iloc[0:5,0:30]
```

```
   Age Attrition     BusinessTravel  DailyRate              Department
\
0   41       Yes      Travel_Rarely       1102                   Sales

1   49        No  Travel_Frequently        279  Research & Development

2   37       Yes      Travel_Rarely       1373  Research & Development

3   33        No  Travel_Frequently       1392  Research & Development

4   27        No      Travel_Rarely        591  Research & Development


   DistanceFromHome  Education EducationField  EmployeeNumber  \
0                 1          2  Life Sciences               1
1                 8          1  Life Sciences               2
```

```
2                   2           2          Other                 4
3                   3           4  Life Sciences                 5
4                   2           1        Medical                 7

   EnvironmentSatisfaction  ... MonthlyRate  NumCompaniesWorked
OverTime  \
0                        2  ...       19479                   8
Yes
1                        3  ...       24907                   1
No
2                        4  ...        2396                   6
Yes
3                        4  ...       23159                   1
Yes
4                        1  ...       16632                   2
No

  PerformanceRating  RelationshipSatisfaction StockOptionLevel  \
0                 3                         1                 0
1                 4                         4                 1
2                 3                         2                 0
3                 3                         3                 0
4                 3                         4                 1

   TrainingTimesLastYear  WorkLifeBalance  YearsInCurrentRole  \
0                      0                1                   4
1                      3                3                   7
2                      3                3                   0
3                      3                3                   7
4                      3                3                   2

  YearsSinceLastPromotion
0                       0
1                       1
2                       0
3                       3
4                       2

[5 rows x 27 columns]
```

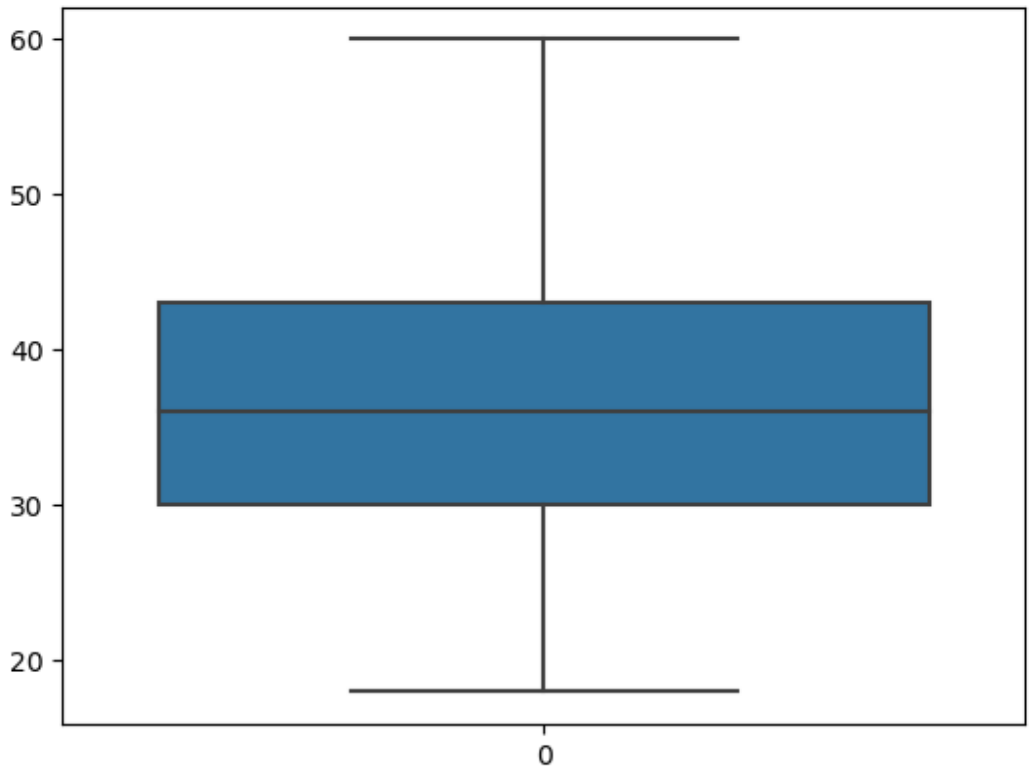
```
df.median()
```

```
/var/folders/75/l1625v4n7qnbfp296v2f82ch0000gn/T/
ipykernel_4203/530051474.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()
```

```
Age                          36.0
DailyRate                   802.0
DistanceFromHome              7.0
Education                     3.0
EmployeeNumber             1020.5
EnvironmentSatisfaction       3.0
HourlyRate                   66.0
JobInvolvement                3.0
JobSatisfaction               3.0
MonthlyIncome              4919.0
MonthlyRate               14235.5
NumCompaniesWorked            2.0
PerformanceRating             3.0
RelationshipSatisfaction      3.0
StockOptionLevel              1.0
TrainingTimesLastYear         3.0
WorkLifeBalance               3.0
YearsInCurrentRole            3.0
YearsSinceLastPromotion       1.0
dtype: float64
```

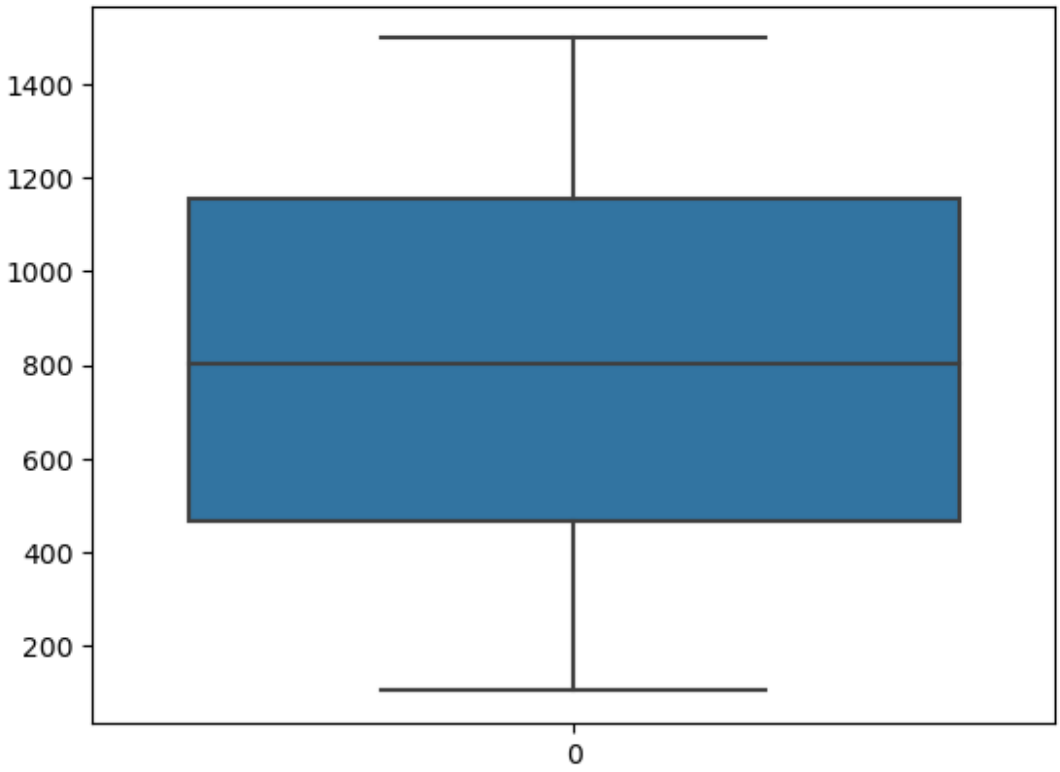
```
sns.boxplot(df.Age)
```

```
<Axes: >
```



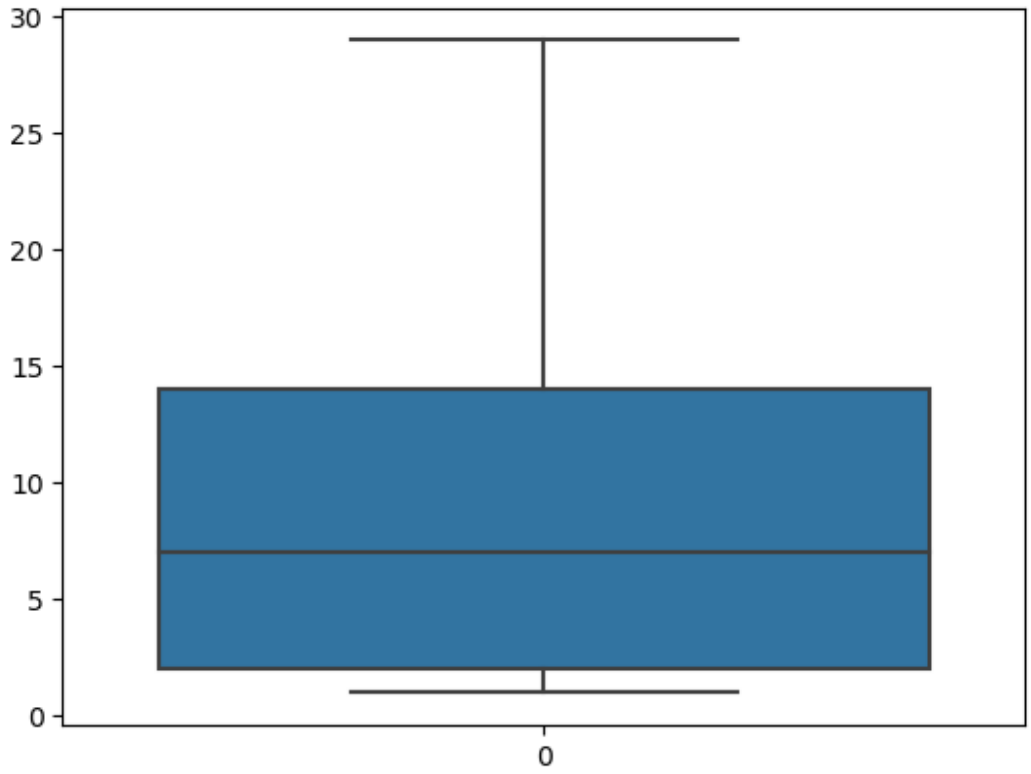
```
sns.boxplot(df.DailyRate)
```

```
<Axes: >
```



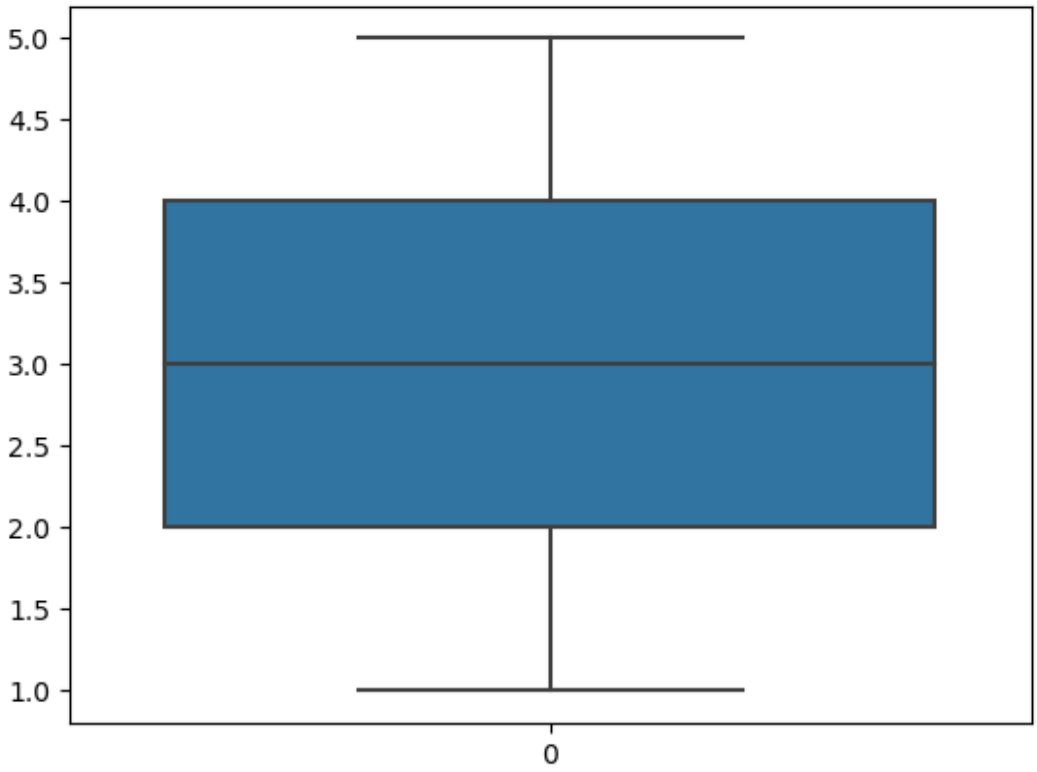

```
sns.boxplot(df.DistanceFromHome)
```

```
<Axes: >
```



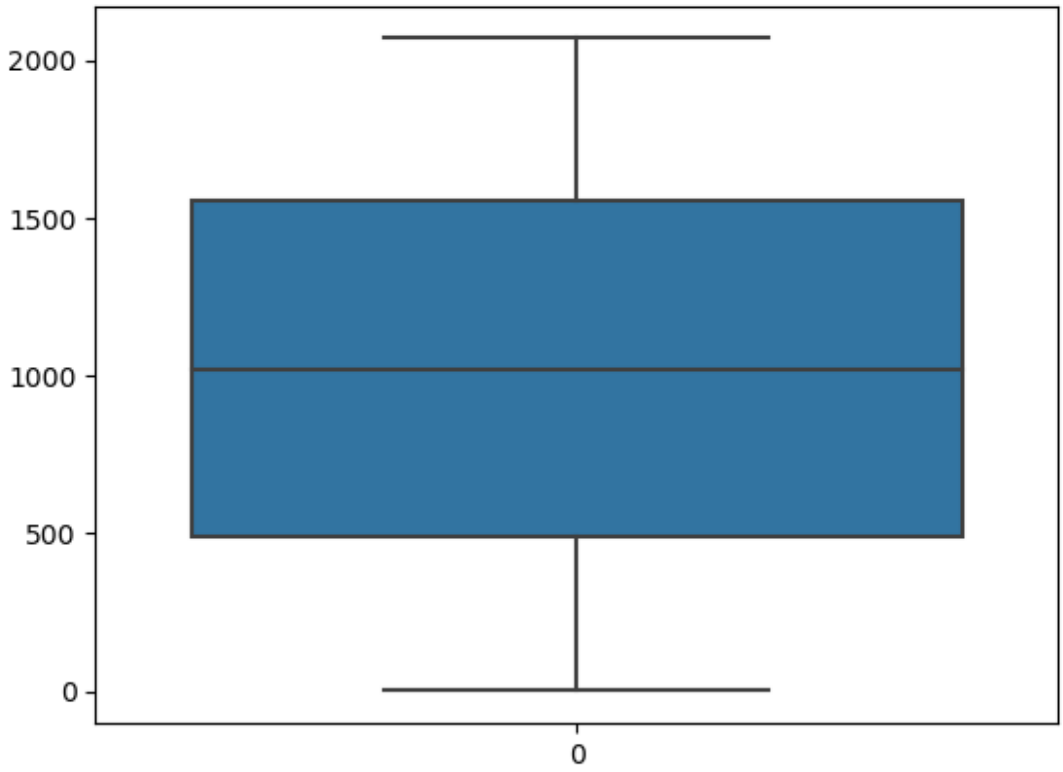

```
sns.boxplot(df.Education)
```

```
<Axes: >
```



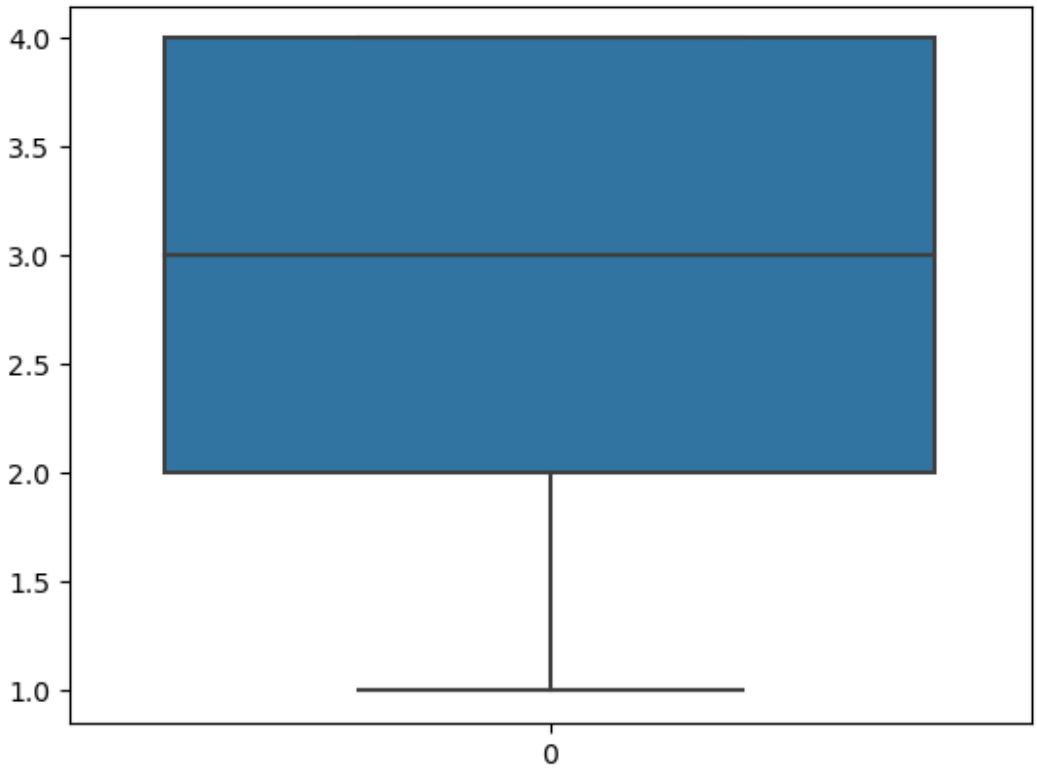

```
sns.boxplot(df.EmployeeNumber)
```

```
<Axes: >
```



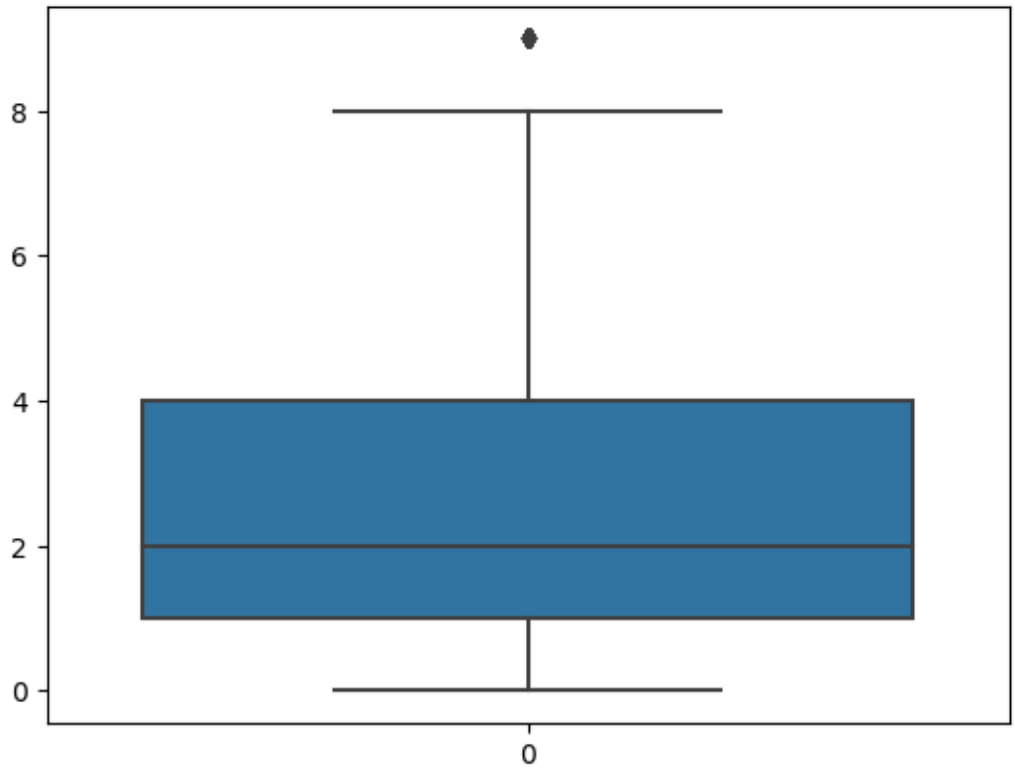

```
sns.boxplot(df.EnvironmentSatisfaction)
```

```
<Axes: >
```



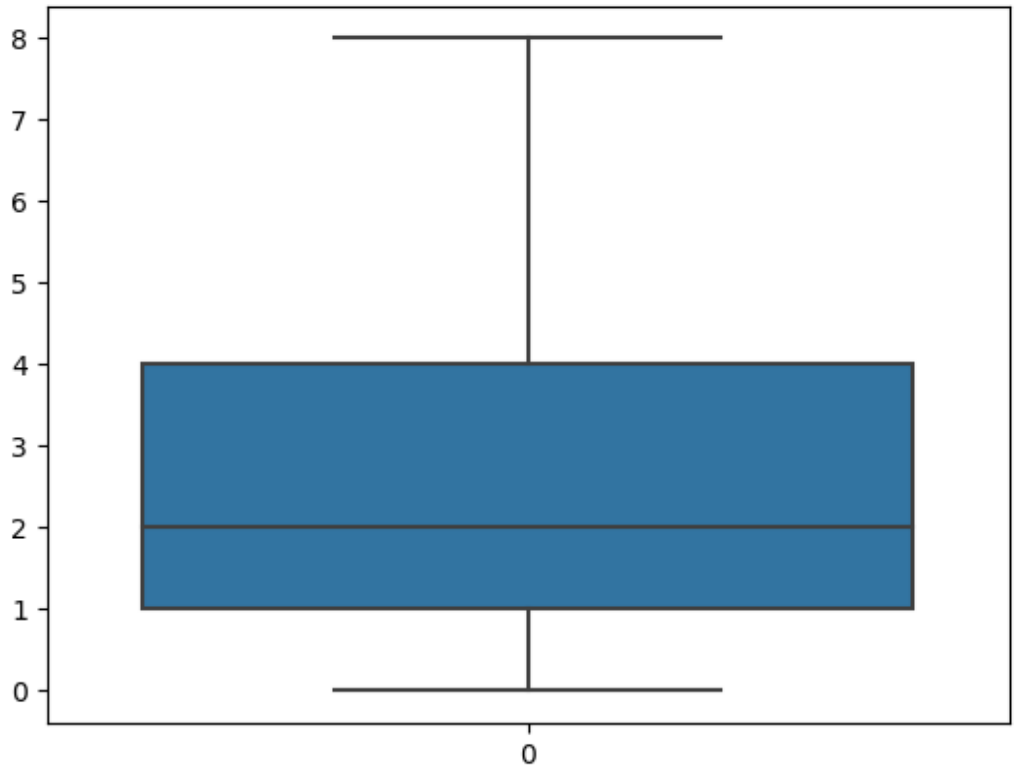

```
sns.boxplot(df.NumCompaniesWorked)
```

```
<Axes: >
```




```
q1=df.NumCompaniesWorked.quantile(0.25)
q3=df.NumCompaniesWorked.quantile(0.75)
iqr=q3-q1

upper_limit=q3+1.5*iqr
lower_limit=q1-1.5*iqr

df['NumCompaniesWorked']=np.where(df['NumCompaniesWorked']>upper_limit
,2,df['NumCompaniesWorked'])

sns.boxplot(df.NumCompaniesWorked)
```

```
<Axes: >
```




```
pd.set_option('display.max_columns', None)
df.head()
```

```
   Age Attrition     BusinessTravel  DailyRate                Department
\
0   41       Yes      Travel_Rarely       1102                     Sales

1   49        No  Travel_Frequently        279  Research & Development

2   37       Yes      Travel_Rarely       1373  Research & Development

3   33        No  Travel_Frequently       1392  Research & Development

4   27        No      Travel_Rarely        591  Research & Development


   DistanceFromHome  Education EducationField  EmployeeNumber  \
0                 1          2  Life Sciences               1
1                 8          1  Life Sciences               2
2                 2          2         Other               4
3                 3          4  Life Sciences               5
4                 2          1        Medical               7

   EnvironmentSatisfaction  Gender  HourlyRate  JobInvolvement  \
0                        2  Female          94               3
1                        3    Male          61               2
```

```
2                        4    Male          92                2
3                        4  Female          56                3
4                        1    Male          40                3

                  JobRole  JobSatisfaction MaritalStatus  MonthlyIncome
\
0        Sales Executive                4        Single           5993

1     Research Scientist                2       Married           5130

2  Laboratory Technician                3        Single           2090

3     Research Scientist                3       Married           2909

4  Laboratory Technician                2       Married           3468


   MonthlyRate  NumCompaniesWorked OverTime  PerformanceRating  \
0        19479                   8      Yes                  3
1        24907                   1       No                  4
2         2396                   6      Yes                  3
3        23159                   1      Yes                  3
4        16632                   2       No                  3

   RelationshipSatisfaction  StockOptionLevel
TrainingTimesLastYear  \
0                         1                 0                       0

1                         4                 1                       3

2                         2                 0                       3

3                         3                 0                       3

4                         4                 1                       3


   WorkLifeBalance  YearsInCurrentRole  YearsSinceLastPromotion
0                1                   4                        0
1                3                   7                        1
2                3                   0                        0
3                3                   7                        3
4                3                   2                        2
```

```
#splitting into dependent and independent variables
x=df.drop(['Attrition'],axis=1)
y=df.Attrition

x.head()
```

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

   DistanceFromHome  Education EducationField  EmployeeNumber  \
0                 1          2  Life Sciences               1
1                 8          1  Life Sciences               2
2                 2          2          Other               4
3                 3          4  Life Sciences               5
4                 2          1        Medical               7

   EnvironmentSatisfaction  Gender  HourlyRate  JobInvolvement  \
0                        2  Female          94               3
1                        3    Male          61               2
2                        4    Male          92               2
3                        4  Female          56               3
4                        1    Male          40               3

                JobRole  JobSatisfaction MaritalStatus  MonthlyIncome
\
0        Sales Executive                4        Single           5993

1     Research Scientist                2       Married           5130

2  Laboratory Technician                3        Single           2090

3     Research Scientist                3       Married           2909

4  Laboratory Technician                2       Married           3468


   MonthlyRate  NumCompaniesWorked OverTime  PerformanceRating  \
0        19479                   8      Yes                  3
1        24907                   1       No                  4
2         2396                   6      Yes                  3
3        23159                   1      Yes                  3
4        16632                   2       No                  3

   RelationshipSatisfaction  StockOptionLevel
TrainingTimesLastYear  \
0                         1                 0                      0

1                         4                 1                      3

2                         2                 0                      3

3                         3                 0                      3
```

```
4                          4                  1                          3


   WorkLifeBalance  YearsInCurrentRole  YearsSinceLastPromotion
0                1                   4                        0
1                3                   7                        1
2                3                   0                        0
3                3                   7                        3
4                3                   2                        2
```

```
y.head()
```

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

```
#label Encoding
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
x.BusinessTravel=le.fit_transform(x.BusinessTravel)
x.Department=le.fit_transform(x.Department)
x.EducationField=le.fit_transform(x.EducationField)
x.Gender=le.fit_transform(x.Gender)
x.JobRole=le.fit_transform(x.JobRole)
x.MaritalStatus=le.fit_transform(x.MaritalStatus)
x.OverTime=le.fit_transform(x.OverTime)
x.head()
```

```
   Age  BusinessTravel  DailyRate  Department  DistanceFromHome
Education  \
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

   EducationField  EmployeeNumber  EnvironmentSatisfaction  Gender  \
0               1               1                        2       0
1               1               2                        3       1
2               4               4                        4       1
3               1               5                        4       0
4               3               7                        1       1
```

```
   HourlyRate  JobInvolvement  JobRole  JobSatisfaction  MaritalStatus
\
0          94               3        7                4              2

1          61               2        6                2              1

2          92               2        2                3              2

3          56               3        6                3              1

4          40               3        2                2              1


   MonthlyIncome  MonthlyRate  NumCompaniesWorked  OverTime  \
0           5993        19479                   8         1
1           5130        24907                   1         0
2           2090         2396                   6         1
3           2909        23159                   1         1
4           3468        16632                   2         0

   PerformanceRating  RelationshipSatisfaction  StockOptionLevel  \
0                  3                         1                 0
1                  4                         4                 1
2                  3                         2                 0
3                  3                         3                 0
4                  3                         4                 1

   TrainingTimesLastYear  WorkLifeBalance  YearsInCurrentRole  \
0                      0                1                   4
1                      3                3                   7
2                      3                3                   0
3                      3                3                   7
4                      3                3                   2

   YearsSinceLastPromotion
0                        0
1                        1
2                        0
3                        3
4                        2
```

```
y=y.to_frame()
```

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

```
y=y.squeeze()
```

```
y.head()
```

```
0    1
1    0
```

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

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

```
x_scaled
```

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

      Education  EducationField  EmployeeNumber
EnvironmentSatisfaction  \
0          0.25             0.2       0.000000
0.333333
1          0.00             0.2       0.000484
0.666667
2          0.25             0.8       0.001451
1.000000
3          0.75             0.2       0.001935
1.000000
4          0.00             0.6       0.002903
0.000000
...         ...             ...            ...                      .
```

```
..
1465       0.25              0.6        0.996613
0.666667
1466       0.00              0.6        0.997097
1.000000
1467       0.50              0.2        0.998065
0.333333
1468       0.50              0.6        0.998549
1.000000
1469       0.50              0.6        1.000000
0.333333

      Gender  HourlyRate  JobInvolvement  JobRole  JobSatisfaction  \
0        0.0    0.914286        0.666667    0.875         1.000000
1        1.0    0.442857        0.333333    0.750         0.333333
2        1.0    0.885714        0.333333    0.250         0.666667
3        0.0    0.371429        0.666667    0.750         0.666667
4        1.0    0.142857        0.666667    0.250         0.333333
...      ...         ...             ...      ...              ...
1465     1.0    0.157143        1.000000    0.250         1.000000
1466     1.0    0.171429        0.333333    0.000         0.000000
1467     1.0    0.814286        1.000000    0.500         0.333333
1468     1.0    0.471429        0.333333    0.875         0.333333
1469     1.0    0.742857        1.000000    0.250         0.666667

      MaritalStatus  MonthlyIncome  MonthlyRate  NumCompaniesWorked
OverTime  \
0               1.0       0.262454     0.698053               1.000
1.0
1               0.5       0.217009     0.916001               0.125
0.0
2               1.0       0.056925     0.012126               0.750
1.0
3               0.5       0.100053     0.845814               0.125
1.0
4               0.5       0.129489     0.583738               0.250
0.0
...             ...            ...          ...                 ...
...
1465            0.5       0.082254     0.409396               0.500
0.0
1466            0.5       0.472986     0.777474               0.500
0.0
1467            0.5       0.270300     0.123670               0.125
1.0
1468            0.5       0.230700     0.447661               0.250
0.0
1469            0.5       0.178778     0.326601               0.250
0.0
```

```
      PerformanceRating  RelationshipSatisfaction  StockOptionLevel  \
0                   0.0                  0.000000          0.000000
1                   1.0                  1.000000          0.333333
2                   0.0                  0.333333          0.000000
3                   0.0                  0.666667          0.000000
4                   0.0                  1.000000          0.333333
...                 ...                       ...               ...
1465                0.0                  0.666667          0.333333
1466                0.0                  0.000000          0.333333
1467                1.0                  0.333333          0.333333
1468                0.0                  1.000000          0.000000
1469                0.0                  0.000000          0.000000

      TrainingTimesLastYear  WorkLifeBalance  YearsInCurrentRole  \
0                  0.000000         0.000000            0.222222
1                  0.500000         0.666667            0.388889
2                  0.500000         0.666667            0.000000
3                  0.500000         0.666667            0.388889
4                  0.500000         0.666667            0.111111
...                     ...              ...                 ...
1465               0.500000         0.666667            0.111111
1466               0.833333         0.666667            0.388889
1467               0.000000         0.666667            0.111111
1468               0.500000         0.333333            0.333333
1469               0.500000         1.000000            0.166667

      YearsSinceLastPromotion
0                    0.000000
1                    0.066667
2                    0.000000
3                    0.200000
4                    0.133333
...                       ...
1465                 0.000000
1466                 0.066667
1467                 0.000000
1468                 0.000000
1469                 0.066667

[1470 rows x 26 columns]
```

```
#Splitting Data into Train and Test.
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.
2,random_state=0)

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

```
((1176, 26), (294, 26), (1176,), (294,))
```

```
x_train.head()
```

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

      Education  EducationField  EmployeeNumber
EnvironmentSatisfaction  \
1374       0.50             0.2        0.937107
1.000000
1092       0.50             1.0        0.747460
1.000000
768        0.50             0.4        0.515239
0.666667
569        0.75             0.2        0.381229
0.000000
911        0.00             0.2        0.615385
0.666667

      Gender  HourlyRate  JobInvolvement  JobRole  JobSatisfaction  \
1374     0.0    0.600000        0.666667    0.375              1.0
1092     1.0    0.957143        0.666667    0.750              1.0
768      1.0    0.628571        0.666667    0.875              0.0
569      1.0    0.657143        0.333333    0.875              0.0
911      1.0    0.614286        0.000000    1.000              1.0

      MaritalStatus  MonthlyIncome  MonthlyRate  NumCompaniesWorked
OverTime  \
1374            0.5       0.888152     0.388155               0.500
1.0
1092            0.5       0.059136     0.100020               0.500
0.0
768             0.5       0.388994     0.807990               0.125
0.0
569             1.0       0.346393     0.487252               0.125
0.0
911             1.0       0.005740     0.238747               0.125
1.0

      PerformanceRating  RelationshipSatisfaction  StockOptionLevel  \
1374                0.0                  0.666667          0.333333
```

```
1092                 1.0                   1.000000          0.333333
768                  0.0                   0.333333          0.333333
569                  0.0                   0.333333          0.000000
911                  0.0                   1.000000          0.000000

      TrainingTimesLastYear  WorkLifeBalance  YearsInCurrentRole  \
1374               0.333333         0.333333            0.000000
1092               0.500000         0.666667            0.222222
768                0.500000         0.333333            0.388889
569                0.166667         0.666667            0.388889
911                0.666667         0.666667            0.000000

      YearsSinceLastPromotion
1374                 0.000000
1092                 0.000000
768                  0.466667
569                  0.000000
911                  0.066667
```

# Logistic Regression Model Building

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

model.fit(x_train,y_train)

LogisticRegression()

pred=model.predict(x_test)

pred

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, 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, 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, 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])
```

```
y_test
```

```
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
```

# Evaluation of Logistic Regression Model

```
#Accuracy score
from sklearn.metrics import
accuracy_score,confusion_matrix,classification_report,roc_auc_score,ro
c_curve

p=accuracy_score(y_test,pred)
accuracy_score(y_test,pred)
```

```
0.8843537414965986
```

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

```
array([[242,   3],
       [ 31,  18]])
```

```
pd.crosstab(y_test,pred)
```

```
col_0        0   1
Attrition
0          242   3
1           31  18
```

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

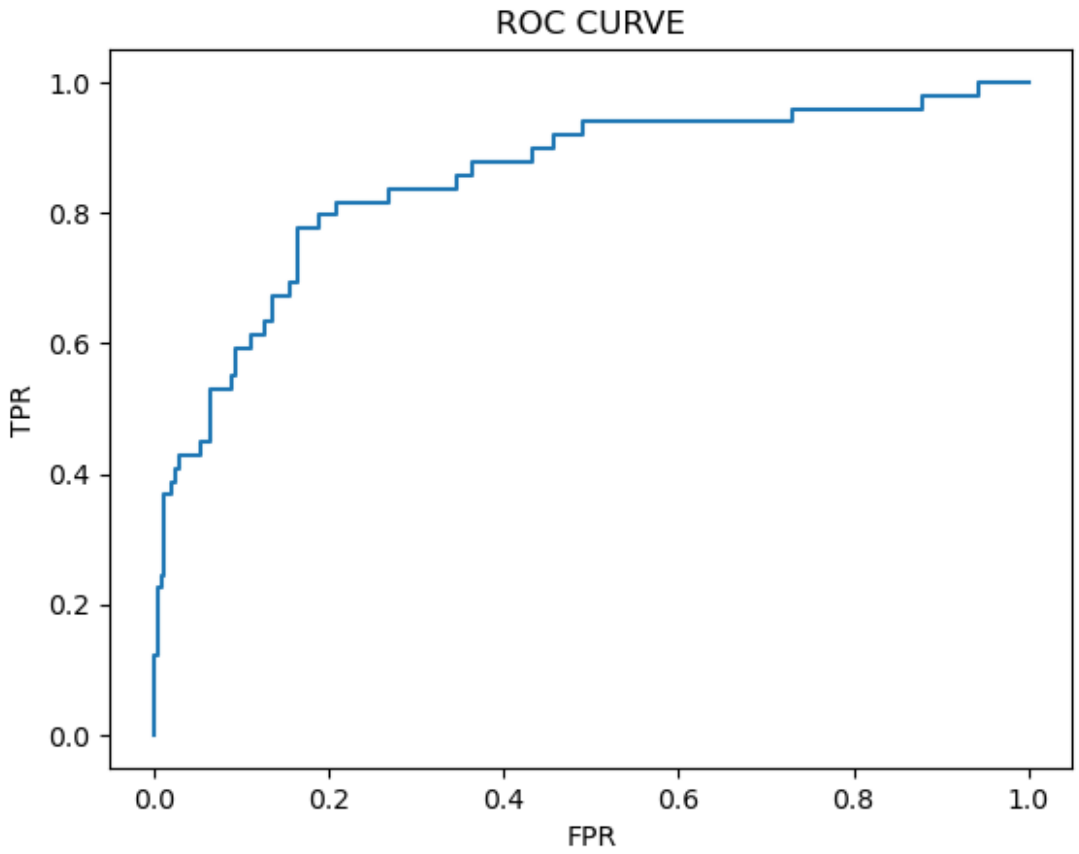
```
              precision    recall  f1-score   support

           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
```

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

```
probability
```

```
array([0.18532596, 0.21074855, 0.36371946, 0.05046911, 0.63269845,
       0.05414756, 0.69495093, 0.08924896, 0.01405847, 0.18691855,
       0.08151551, 0.30709245, 0.02326356, 0.6927869 , 0.31550294,
       0.04531878, 0.13603538, 0.2545832 , 0.0564401 , 0.19915012,
       0.2566306 , 0.02323744, 0.05120053, 0.05625222, 0.61338058,
       0.41873684, 0.06416652, 0.03784882, 0.64125079, 0.04786785,
       0.01946755, 0.06509844, 0.07778437, 0.24587936, 0.08168993,
       0.07838116, 0.07499929, 0.07932737, 0.04376017, 0.05126789,
       0.10274525, 0.0230451 , 0.01914007, 0.02072721, 0.03150459,
       0.52971447, 0.19671177, 0.00522148, 0.76213205, 0.53709074,
       0.11684521, 0.43891975, 0.0909896 , 0.24821133, 0.66785067,
       0.32706794, 0.02323478, 0.31484279, 0.03320729, 0.14291994,
       0.02628869, 0.18323607, 0.16285239, 0.0359007 , 0.43104435,
       0.0264165 , 0.2119949 , 0.21613901, 0.11438524, 0.08878625,
       0.10179045, 0.28321471, 0.06740378, 0.08599529, 0.05413943,
       0.05139055, 0.0455572 , 0.10901957, 0.16443483, 0.04456544,
       0.01732033, 0.02650078, 0.15815977, 0.02973569, 0.04029292,
       0.09646624, 0.00813451, 0.02764671, 0.03680343, 0.17324225,
       0.28417093, 0.17223339, 0.27194985, 0.24759707, 0.0251624 ,
       0.17389093, 0.36553838, 0.26265936, 0.07774703, 0.05126206,
       0.30432329, 0.7749655 , 0.40658177, 0.02170859, 0.06901483,
       0.04396064, 0.05302501, 0.11923936, 0.04552359, 0.12488685,
       0.16716875, 0.05528105, 0.02266012, 0.19169842, 0.07696218,
       0.04637376, 0.07539872, 0.13606043, 0.0147093 , 0.01580461,
       0.13557355, 0.05470912, 0.07263096, 0.82769235, 0.03745154,
       0.03965809, 0.01360083, 0.13707492, 0.15955433, 0.07354848,
       0.01835709, 0.26473266, 0.53085648, 0.42093428, 0.07084086,
       0.48825034, 0.59917   , 0.16876821, 0.08337742, 0.31000067,
       0.11493549, 0.07692831, 0.10160066, 0.14422955, 0.20774523,
       0.02179043, 0.17638724, 0.00436062, 0.11919021, 0.16626063,
       0.07177462, 0.28141277, 0.06277678, 0.19430839, 0.04603545,
       0.03557427, 0.05394644, 0.08760897, 0.0304259 , 0.01222833,
       0.47975826, 0.01558192, 0.16853502, 0.80142614, 0.11091125,
       0.2786657 , 0.19207034, 0.13475844, 0.03733218, 0.01098748,
```

```
       0.05694537, 0.07998275, 0.14019159, 0.08495555, 0.01392303,
       0.14118124, 0.10643604, 0.10708076, 0.05120921, 0.12764331,
       0.03137797, 0.10488951, 0.00783618, 0.72084162, 0.04257806,
       0.04189839, 0.31616575, 0.06243518, 0.71911779, 0.1243246 ,
       0.36908101, 0.40272961, 0.25077157, 0.06693704, 0.06581206,
       0.15804026, 0.04629401, 0.01647846, 0.21140125, 0.07278495,
       0.15035364, 0.15524342, 0.60224146, 0.06411758, 0.21684531,
       0.02969623, 0.43238943, 0.00648632, 0.11866829, 0.03739583,
       0.11743849, 0.15798224, 0.05022259, 0.09933937, 0.21376722,
       0.03348568, 0.02126876, 0.08127425, 0.03865786, 0.13749949,
       0.07799974, 0.22544841, 0.7346826 , 0.10473773, 0.49927999,
       0.01351087, 0.11280109, 0.24612507, 0.35631869, 0.04624246,
       0.03354115, 0.26638873, 0.05790247, 0.01564704, 0.17794329,
       0.40684088, 0.16231711, 0.01389607, 0.07791657, 0.02278518,
       0.1253285 , 0.30319368, 0.01703636, 0.19259385, 0.04219834,
       0.04707036, 0.45990441, 0.33941586, 0.05196281, 0.19278017,
       0.32993252, 0.38577097, 0.80740361, 0.05992691, 0.2012654 ,
       0.05749433, 0.01058121, 0.61291506, 0.17318323, 0.35183829,
       0.44134431, 0.05126491, 0.29939162, 0.04954516, 0.08939216,
       0.1038666 , 0.01193164, 0.28233072, 0.44218353, 0.09846265,
       0.10884983, 0.01708318, 0.16644139, 0.04980655, 0.02830771,
       0.04359696, 0.08816435, 0.2513607 , 0.11988478, 0.19428493,
       0.25599145, 0.01700871, 0.17614081, 0.09088042, 0.05086016,
       0.18176732, 0.01146869, 0.36334556, 0.00280144, 0.03707845,
       0.20885587, 0.78338154, 0.03837995, 0.32531172])

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

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




# Decesion Tree

```
from sklearn.tree import DecisionTreeClassifier
dtc=DecisionTreeClassifier()

dtc.fit(x_train,y_train)

DecisionTreeClassifier()

pred2=dtc.predict(x_test)

pred2

array([0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0,
0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1,
       0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1,
0,
       1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0,
0,
```

```
       0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0,
1,
       0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,
1,
       0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
0,
       0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       1, 0, 1, 0, 0, 0, 0, 0, 1, 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, 1, 0, 0, 0, 1, 1,
0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 1, 0, 0, 0])
```

```
y_test
```

```
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
```

# Evaluation

```
p2=accuracy_score(y_test,pred2)
accuracy_score(y_test,pred2)
```

```
0.7619047619047619
```

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

```
array([[210,  35],
       [ 29,  20]])
```

```
pd.crosstab(y_test,pred)
```

```
col_0        0   1
Attrition
```

```
0          242   3
1           31  18
```

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

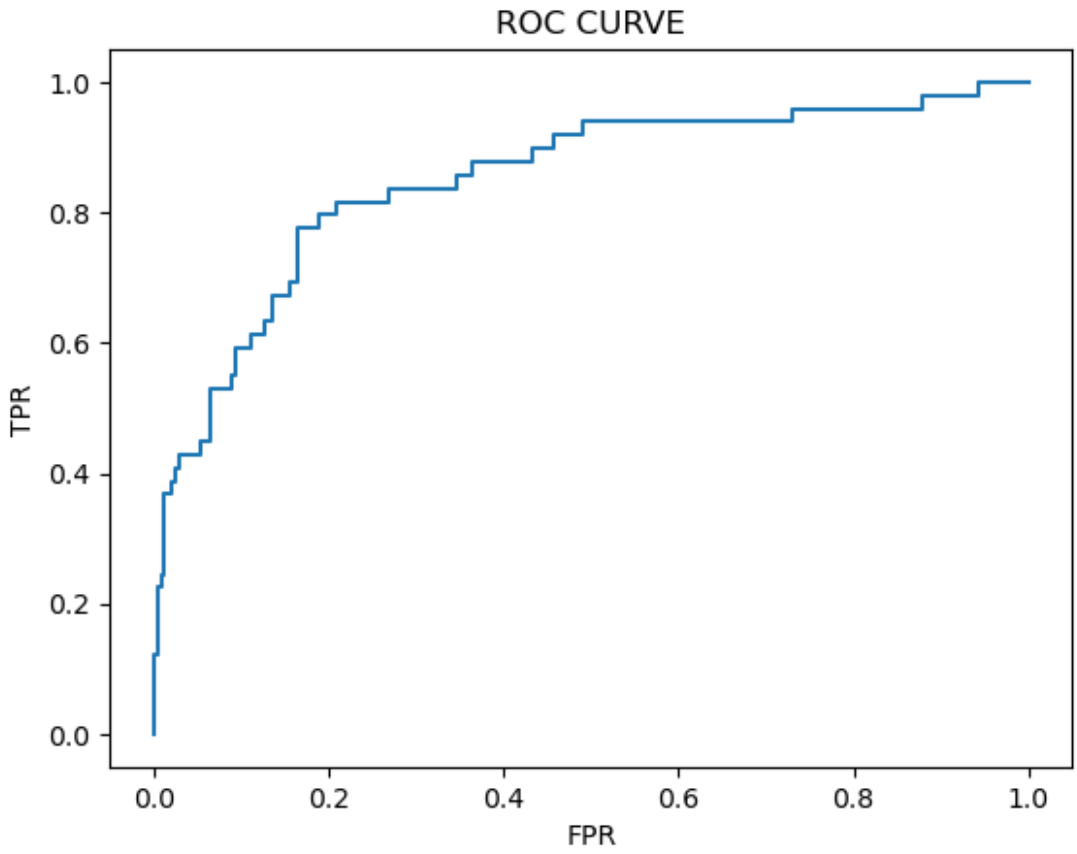
```
              precision    recall  f1-score   support

           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
```

```
probability2=dtc.predict_proba(x_test)[:,1]
```

```
probability2
```

```
array([0., 0., 0., 0., 1., 0., 1., 0., 0., 0., 0., 0., 1., 1., 0., 0.,
1.,
       0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0.,
0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 0., 1., 0.,
0.,
       0., 1., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 1., 1., 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., 1., 0., 0., 1.,
0.,
       1., 0., 0., 0., 0., 1., 0., 0., 0., 1., 0., 0., 0., 1., 1., 1.,
0.,
       0., 0., 0., 0., 1., 1., 1., 0., 0., 0., 0., 0., 1., 0., 0., 0.,
0.,
       1., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0.,
0.,
       1., 0., 1., 0., 0., 0., 1., 1., 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., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1.,
0.,
       0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
0.,
       0., 0., 0., 0., 0., 0., 0., 1., 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., 1., 0., 1., 0., 1., 0., 0., 0., 0., 0., 1., 0., 0., 0.,
1.,
```

```
       0., 0., 0., 1., 0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 1.])

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

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




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

[Text(0.3889727184934814, 0.9705882352941176, 'x[15] <= 0.093\ngini =
0.269\nsamples = 1176\nvalue = [988, 188]'),
 Text(0.13275591501690004, 0.9117647058823529, 'x[18] <= 0.5\ngini =
0.431\nsamples = 271\nvalue = [186, 85]'),
 Text(0.07188556253017865, 0.8529411764705882, 'x[0] <= 0.083\ngini =
0.327\nsamples = 194\nvalue = [154, 40]'),
 Text(0.0386286817962337, 0.7941176470588235, 'x[2] <= 0.377\ngini =
```

```
0.488\nsamples = 19\nvalue = [8, 11]'),
 Text(0.02317720907774022, 0.7352941176470589, 'x[15] <= 0.017\ngini =
0.298\nsamples = 11\nvalue = [2, 9]'),
 Text(0.01545147271849348, 0.6764705882352942, 'x[5] <= 0.375\ngini =
0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.00772573635924674, 0.6176470588235294, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.02317720907774022, 0.6176470588235294, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.03090294543698696, 0.6764705882352942, 'gini = 0.0\nsamples =
8\nvalue = [0, 8]'),
 Text(0.05408015451472718, 0.7352941176470589, 'x[2] <= 0.827\ngini =
0.375\nsamples = 8\nvalue = [6, 2]'),
 Text(0.04635441815548044, 0.6764705882352942, 'gini = 0.0\nsamples =
6\nvalue = [6, 0]'),
 Text(0.06180589087397392, 0.6764705882352942, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.10514244326412361, 0.7941176470588235, 'x[15] <= 0.033\ngini =
0.277\nsamples = 175\nvalue = [146, 29]'),
 Text(0.08498309995171414, 0.7352941176470589, 'x[14] <= 0.75\ngini =
0.48\nsamples = 10\nvalue = [4, 6]'),
 Text(0.0772573635924674, 0.6764705882352942, 'x[7] <= 0.773\ngini =
0.32\nsamples = 5\nvalue = [4, 1]'),
 Text(0.06953162723322066, 0.6176470588235294, 'gini = 0.0\nsamples =
4\nvalue = [4, 0]'),
 Text(0.08498309995171414, 0.6176470588235294, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.09270883631096088, 0.6764705882352942, 'gini = 0.0\nsamples =
5\nvalue = [0, 5]'),
 Text(0.12530178657653307, 0.7352941176470589, 'x[17] <= 0.562\ngini =
0.24\nsamples = 165\nvalue = [142, 23]'),
 Text(0.10816030902945437, 0.6764705882352942, 'x[15] <= 0.092\ngini =
0.187\nsamples = 144\nvalue = [129, 15]'),
 Text(0.10043457267020763, 0.6176470588235294, 'x[24] <= 0.694\ngini =
0.177\nsamples = 143\nvalue = [129, 14]'),
 Text(0.09270883631096088, 0.5588235294117647, 'x[2] <= 0.006\ngini =
0.166\nsamples = 142\nvalue = [129, 13]'),
 Text(0.08498309995171414, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.10043457267020763, 0.5, 'x[0] <= 0.869\ngini = 0.156\nsamples
= 141\nvalue = [129, 12]'),
 Text(0.07387735393529696, 0.4411764705882353, 'x[16] <= 0.063\ngini =
0.135\nsamples = 137\nvalue = [127, 10]'),
 Text(0.04394012554321584, 0.38235294117647056, 'x[24] <= 0.056\ngini
= 0.444\nsamples = 6\nvalue = [4, 2]'),
 Text(0.036214389183969097, 0.3235294117647059, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.05166586190246258, 0.3235294117647059, 'gini = 0.0\nsamples =
4\nvalue = [4, 0]'),
```

```
 Text(0.10381458232737807, 0.38235294117647056, 'x[8] <= 0.167\ngini =
0.115\nsamples = 131\nvalue = [123, 8]'),
 Text(0.06711733462095607, 0.3235294117647059, 'x[10] <= 0.271\ngini =
0.32\nsamples = 20\nvalue = [16, 4]'),
 Text(0.05166586190246258, 0.2647058823529412, 'x[0] <= 0.143\ngini =
0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.04394012554321584, 0.20588235294117646, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.05939159826170932, 0.20588235294117646, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.08256880733944955, 0.2647058823529412, 'x[23] <= 0.833\ngini =
0.117\nsamples = 16\nvalue = [15, 1]'),
 Text(0.0748430709802028, 0.20588235294117646, 'gini = 0.0\nsamples =
15\nvalue = [15, 0]'),
 Text(0.09029454369869629, 0.20588235294117646, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.1405118300338001, 0.3235294117647059, 'x[15] <= 0.055\ngini =
0.069\nsamples = 111\nvalue = [107, 4]'),
 Text(0.11347175277643651, 0.2647058823529412, 'x[13] <= 0.167\ngini =
0.32\nsamples = 10\nvalue = [8, 2]'),
 Text(0.10574601641718977, 0.20588235294117646, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.12119748913568325, 0.20588235294117646, 'x[17] <= 0.062\ngini
= 0.198\nsamples = 9\nvalue = [8, 1]'),
 Text(0.11347175277643651, 0.14705882352941177, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.12892322549493, 0.14705882352941177, 'gini = 0.0\nsamples = 8\
nvalue = [8, 0]'),
 Text(0.16755190729116368, 0.2647058823529412, 'x[10] <= 0.057\ngini =
0.039\nsamples = 101\nvalue = [99, 2]'),
 Text(0.1521004345726702, 0.20588235294117646, 'x[16] <= 0.464\ngini =
0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.14437469821342347, 0.14705882352941177, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.15982617093191695, 0.14705882352941177, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.18300338000965716, 0.20588235294117646, 'x[22] <= 0.75\ngini =
0.02\nsamples = 98\nvalue = [97, 1]'),
 Text(0.17527764365041043, 0.14705882352941177, 'gini = 0.0\nsamples =
86\nvalue = [86, 0]'),
 Text(0.1907291163689039, 0.14705882352941177, 'x[16] <= 0.196\ngini =
0.153\nsamples = 12\nvalue = [11, 1]'),
 Text(0.18300338000965716, 0.08823529411764706, 'x[23] <= 0.5\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.17527764365041043, 0.029411764705882353, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
 Text(0.1907291163689039, 0.029411764705882353, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.19845485272815064, 0.08823529411764706, 'gini = 0.0\nsamples =
```

```
10\nvalue = [10, 0]'),
 Text(0.1269917914051183, 0.4411764705882353, 'x[21] <= 0.167\ngini =
0.5\nsamples = 4\nvalue = [2, 2]'),
 Text(0.11926605504587157, 0.38235294117647056, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.13471752776436505, 0.38235294117647056, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.10816030902945437, 0.5588235294117647, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.1158860453887011, 0.6176470588235294, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.14244326412361177, 0.6764705882352942, 'x[13] <= 0.167\ngini =
0.472\nsamples = 21\nvalue = [13, 8]'),
 Text(0.13471752776436505, 0.6176470588235294, 'gini = 0.0\nsamples =
4\nvalue = [0, 4]'),
 Text(0.15016900048285853, 0.6176470588235294, 'x[4] <= 0.196\ngini =
0.36\nsamples = 17\nvalue = [13, 4]'),
 Text(0.14244326412361177, 0.5588235294117647, 'gini = 0.0\nsamples =
9\nvalue = [9, 0]'),
 Text(0.15789473684210525, 0.5588235294117647, 'x[6] <= 0.8\ngini =
0.5\nsamples = 8\nvalue = [4, 4]'),
 Text(0.15016900048285853, 0.5, 'x[4] <= 0.268\ngini = 0.32\nsamples =
5\nvalue = [4, 1]'),
 Text(0.14244326412361177, 0.4411764705882353, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.15789473684210525, 0.4411764705882353, 'gini = 0.0\nsamples =
4\nvalue = [4, 0]'),
 Text(0.165620473201352, 0.5, 'gini = 0.0\nsamples = 3\nvalue = [0,
3]'),
 Text(0.19362626750362144, 0.8529411764705882, 'x[0] <= 0.202\ngini =
0.486\nsamples = 77\nvalue = [32, 45]'),
 Text(0.16948334138097537, 0.7941176470588235, 'x[25] <= 0.1\ngini =
0.172\nsamples = 21\nvalue = [2, 19]'),
 Text(0.16175760502172865, 0.7352941176470589, 'gini = 0.0\nsamples =
19\nvalue = [0, 19]'),
 Text(0.17720907774022213, 0.7352941176470589, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.2177691936262675, 0.7941176470588235, 'x[2] <= 0.691\ngini =
0.497\nsamples = 56\nvalue = [30, 26]'),
 Text(0.1926605504587156, 0.7352941176470589, 'x[17] <= 0.062\ngini =
0.478\nsamples = 38\nvalue = [15, 23]'),
 Text(0.17334620956059874, 0.6764705882352942, 'x[1] <= 0.25\ngini =
0.219\nsamples = 8\nvalue = [7, 1]'),
 Text(0.165620473201352, 0.6176470588235294, 'gini = 0.0\nsamples = 1\
nvalue = [0, 1]'),
 Text(0.1810719459198455, 0.6176470588235294, 'gini = 0.0\nsamples =
7\nvalue = [7, 0]'),
 Text(0.21197489135683245, 0.6764705882352942, 'x[0] <= 0.429\ngini =
0.391\nsamples = 30\nvalue = [8, 22]'),
```

```
 Text(0.19652341863833897, 0.6176470588235294, 'x[10] <= 0.136\ngini =
0.198\nsamples = 18\nvalue = [2, 16]'),
 Text(0.18879768227909222, 0.5588235294117647, 'x[13] <= 0.333\ngini =
0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.1810719459198455, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.19652341863833897, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [2,
0]'),
 Text(0.2042491549975857, 0.5588235294117647, 'gini = 0.0\nsamples =
15\nvalue = [0, 15]'),
 Text(0.22742636407532593, 0.6176470588235294, 'x[2] <= 0.417\ngini =
0.5\nsamples = 12\nvalue = [6, 6]'),
 Text(0.21970062771607918, 0.5588235294117647, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
 Text(0.23515210043457266, 0.5588235294117647, 'x[2] <= 0.556\ngini =
0.444\nsamples = 9\nvalue = [3, 6]'),
 Text(0.22742636407532593, 0.5, 'gini = 0.0\nsamples = 5\nvalue = [0,
5]'),
 Text(0.2428778367938194, 0.5, 'x[13] <= 0.833\ngini = 0.375\nsamples
= 4\nvalue = [3, 1]'),
 Text(0.23515210043457266, 0.4411764705882353, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.25060357315306614, 0.4411764705882353, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
 Text(0.2428778367938194, 0.7352941176470589, 'x[23] <= 0.167\ngini =
0.278\nsamples = 18\nvalue = [15, 3]'),
 Text(0.23515210043457266, 0.6764705882352942, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.25060357315306614, 0.6764705882352942, 'x[10] <= 0.114\ngini =
0.117\nsamples = 16\nvalue = [15, 1]'),
 Text(0.2428778367938194, 0.6176470588235294, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.25832930951231287, 0.6176470588235294, 'gini = 0.0\nsamples =
15\nvalue = [15, 0]'),
 Text(0.6451895219700627, 0.9117647058823529, 'x[18] <= 0.5\ngini =
0.202\nsamples = 905\nvalue = [802, 103]'),
 Text(0.4270280057943023, 0.8529411764705882, 'x[8] <= 0.167\ngini =
0.148\nsamples = 659\nvalue = [606, 53]'),
 Text(0.30746016417189764, 0.7941176470588235, 'x[11] <= 0.167\ngini =
0.278\nsamples = 144\nvalue = [120, 24]'),
 Text(0.2737807822308064, 0.7352941176470589, 'x[16] <= 0.168\ngini =
0.48\nsamples = 10\nvalue = [4, 6]'),
 Text(0.26605504587155965, 0.6764705882352942, 'gini = 0.0\nsamples =
4\nvalue = [4, 0]'),
 Text(0.2815065185900531, 0.6764705882352942, 'gini = 0.0\nsamples =
6\nvalue = [0, 6]'),
 Text(0.3411395461129889, 0.7352941176470589, 'x[2] <= 0.04\ngini =
0.233\nsamples = 134\nvalue = [116, 18]'),
 Text(0.2969579913085466, 0.6764705882352942, 'x[16] <= 0.235\ngini =
```

```
0.49\nsamples = 7\nvalue = [3, 4]'),
 Text(0.2892322549492999, 0.6176470588235294, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
 Text(0.30468372766779334, 0.6176470588235294, 'gini = 0.0\nsamples =
4\nvalue = [0, 4]'),
 Text(0.3853211009174312, 0.6764705882352942, 'x[24] <= 0.083\ngini =
0.196\nsamples = 127\nvalue = [113, 14]'),
 Text(0.32013520038628684, 0.6176470588235294, 'x[1] <= 0.75\ngini =
0.426\nsamples = 26\nvalue = [18, 8]'),
 Text(0.30468372766779334, 0.5588235294117647, 'x[14] <= 0.75\ngini =
0.32\nsamples = 5\nvalue = [1, 4]'),
 Text(0.2969579913085466, 0.5, 'gini = 0.0\nsamples = 4\nvalue = [0,
4]'),
 Text(0.31240946402704006, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.3355866731047803, 0.5588235294117647, 'x[0] <= 0.464\ngini =
0.308\nsamples = 21\nvalue = [17, 4]'),
 Text(0.32786093674553357, 0.5, 'x[23] <= 0.5\ngini = 0.48\nsamples =
10\nvalue = [6, 4]'),
 Text(0.31240946402704006, 0.4411764705882353, 'x[2] <= 0.288\ngini =
0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.30468372766779334, 0.38235294117647056, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.32013520038628684, 0.38235294117647056, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.343312409464027, 0.4411764705882353, 'x[7] <= 0.83\ngini =
0.278\nsamples = 6\nvalue = [5, 1]'),
 Text(0.3355866731047803, 0.38235294117647056, 'gini = 0.0\nsamples =
5\nvalue = [5, 0]'),
 Text(0.3510381458232738, 0.38235294117647056, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.343312409464027, 0.5, 'gini = 0.0\nsamples = 11\nvalue = [11,
0]'),
 Text(0.4505070014485756, 0.6176470588235294, 'x[21] <= 0.833\ngini =
0.112\nsamples = 101\nvalue = [95, 6]'),
 Text(0.4263640753259295, 0.5588235294117647, 'x[2] <= 0.978\ngini =
0.096\nsamples = 99\nvalue = [94, 5]'),
 Text(0.4012554321583776, 0.5, 'x[0] <= 0.179\ngini = 0.079\nsamples =
97\nvalue = [93, 4]'),
 Text(0.374215354901014, 0.4411764705882353, 'x[15] <= 0.159\ngini =
0.298\nsamples = 11\nvalue = [9, 2]'),
 Text(0.36648961854176726, 0.38235294117647056, 'gini = 0.0\nsamples =
7\nvalue = [7, 0]'),
 Text(0.38194109126026077, 0.38235294117647056, 'x[10] <= 0.479\ngini
= 0.5\nsamples = 4\nvalue = [2, 2]'),
 Text(0.374215354901014, 0.3235294117647059, 'gini = 0.0\nsamples = 2\
nvalue = [2, 0]'),
 Text(0.3896668276195075, 0.3235294117647059, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
```

```
 Text(0.4282955094157412, 0.4411764705882353, 'x[12] <= 0.062\ngini =
0.045\nsamples = 86\nvalue = [84, 2]'),
 Text(0.41284403669724773, 0.38235294117647056, 'x[17] <= 0.188\ngini
= 0.278\nsamples = 6\nvalue = [5, 1]'),
 Text(0.40511830033800095, 0.3235294117647059, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.42056977305649446, 0.3235294117647059, 'gini = 0.0\nsamples =
5\nvalue = [5, 0]'),
 Text(0.4437469821342347, 0.38235294117647056, 'x[23] <= 0.833\ngini =
0.025\nsamples = 80\nvalue = [79, 1]'),
 Text(0.4360212457749879, 0.3235294117647059, 'gini = 0.0\nsamples =
69\nvalue = [69, 0]'),
 Text(0.4514727184934814, 0.3235294117647059, 'x[20] <= 0.167\ngini =
0.165\nsamples = 11\nvalue = [10, 1]'),
 Text(0.4437469821342347, 0.2647058823529412, 'x[4] <= 0.25\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.4360212457749879, 0.20588235294117646, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.4514727184934814, 0.20588235294117646, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.45919845485272814, 0.2647058823529412, 'gini = 0.0\nsamples =
9\nvalue = [9, 0]'),
 Text(0.4514727184934814, 0.5, 'x[6] <= 0.6\ngini = 0.5\nsamples = 2\
nvalue = [1, 1]'),
 Text(0.4437469821342347, 0.4411764705882353, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.45919845485272814, 0.4411764705882353, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.47464992757122165, 0.5588235294117647, 'x[9] <= 0.5\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.46692419121197487, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.4823756639304684, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.5465958474167069, 0.7941176470588235, 'x[7] <= 0.015\ngini =
0.106\nsamples = 515\nvalue = [486, 29]'),
 Text(0.5074843070980203, 0.7352941176470589, 'x[12] <= 0.625\ngini =
0.5\nsamples = 4\nvalue = [2, 2]'),
 Text(0.49975857073877356, 0.6764705882352942, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.5152100434572671, 0.6764705882352942, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.5857073877353935, 0.7352941176470589, 'x[0] <= 0.345\ngini =
0.1\nsamples = 511\nvalue = [484, 27]'),
 Text(0.5306615161757605, 0.6764705882352942, 'x[6] <= 0.1\ngini =
0.189\nsamples = 151\nvalue = [135, 16]'),
 Text(0.5229357798165137, 0.6176470588235294, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.5383872525350072, 0.6176470588235294, 'x[2] <= 0.937\ngini =
```

```
0.18\nsamples = 150\nvalue = [135, 15]'),
 Text(0.5171414775470787, 0.5588235294117647, 'x[23] <= 0.5\ngini =
0.154\nsamples = 143\nvalue = [131, 12]'),
 Text(0.49782713664896183, 0.5, 'x[7] <= 0.166\ngini = 0.314\nsamples
= 41\nvalue = [33, 8]'),
 Text(0.4901014002897151, 0.4411764705882353, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.5055528730082086, 0.4411764705882353, 'x[2] <= 0.902\ngini =
0.229\nsamples = 38\nvalue = [33, 5]'),
 Text(0.49782713664896183, 0.38235294117647056, 'x[10] <= 0.264\ngini
= 0.193\nsamples = 37\nvalue = [33, 4]'),
 Text(0.4901014002897151, 0.3235294117647059, 'x[12] <= 0.625\ngini =
0.391\nsamples = 15\nvalue = [11, 4]'),
 Text(0.4823756639304684, 0.2647058823529412, 'gini = 0.0\nsamples =
7\nvalue = [7, 0]'),
 Text(0.49782713664896183, 0.2647058823529412, 'x[16] <= 0.651\ngini =
0.5\nsamples = 8\nvalue = [4, 4]'),
 Text(0.4901014002897151, 0.20588235294117646, 'x[13] <= 0.833\ngini =
0.32\nsamples = 5\nvalue = [1, 4]'),
 Text(0.4823756639304684, 0.14705882352941177, 'gini = 0.0\nsamples =
4\nvalue = [0, 4]'),
 Text(0.49782713664896183, 0.14705882352941177, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.5055528730082086, 0.20588235294117646, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
 Text(0.5055528730082086, 0.3235294117647059, 'gini = 0.0\nsamples =
22\nvalue = [22, 0]'),
 Text(0.5132786093674553, 0.38235294117647056, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.5364558184451955, 0.5, 'x[10] <= 0.479\ngini = 0.075\nsamples
= 102\nvalue = [98, 4]'),
 Text(0.5287300820859488, 0.4411764705882353, 'gini = 0.0\nsamples =
55\nvalue = [55, 0]'),
 Text(0.5441815548044423, 0.4411764705882353, 'x[10] <= 0.5\ngini =
0.156\nsamples = 47\nvalue = [43, 4]'),
 Text(0.5287300820859488, 0.38235294117647056, 'x[25] <= 0.233\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.5210043457267021, 0.3235294117647059, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.5364558184451955, 0.3235294117647059, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.5596330275229358, 0.38235294117647056, 'x[5] <= 0.375\ngini =
0.124\nsamples = 45\nvalue = [42, 3]'),
 Text(0.5519072911636891, 0.3235294117647059, 'x[24] <= 0.194\ngini =
0.278\nsamples = 18\nvalue = [15, 3]'),
 Text(0.5441815548044423, 0.2647058823529412, 'gini = 0.0\nsamples =
13\nvalue = [13, 0]'),
 Text(0.5596330275229358, 0.2647058823529412, 'x[16] <= 0.169\ngini =
0.48\nsamples = 5\nvalue = [2, 3]'),
```

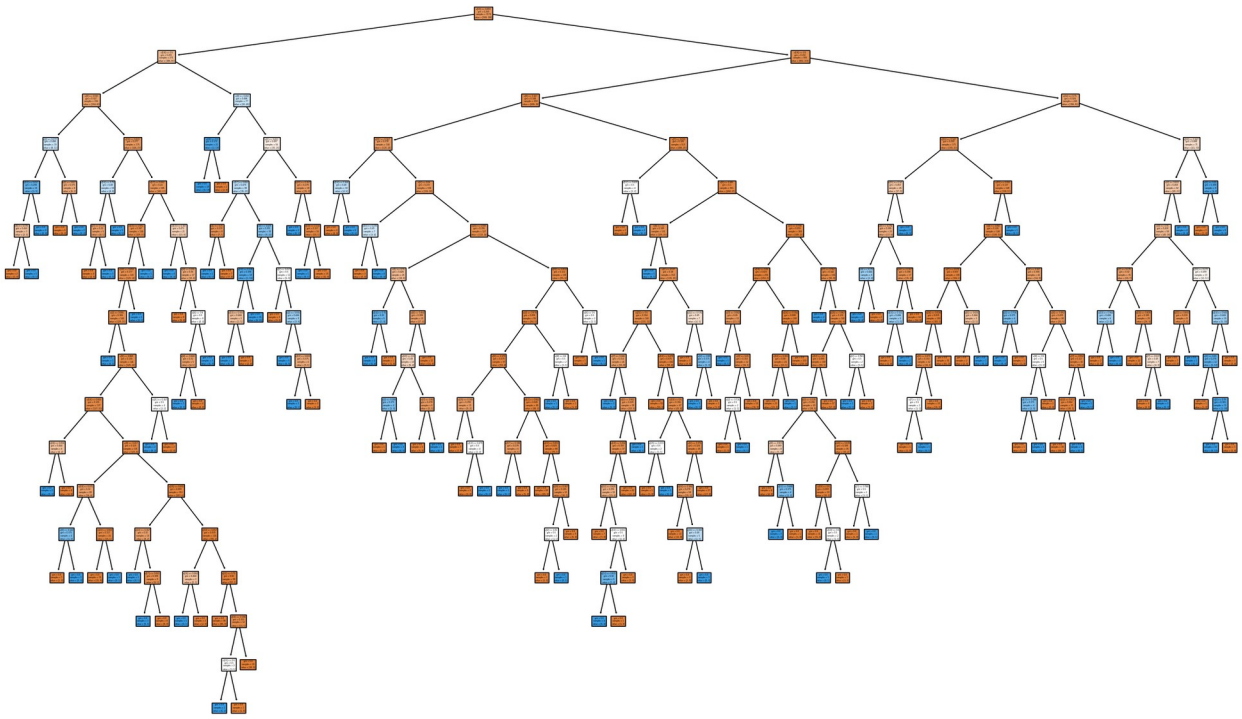
```
 Text(0.5519072911636891, 0.20588235294117646, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.5673587638821825, 0.20588235294117646, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.5673587638821825, 0.3235294117647059, 'gini = 0.0\nsamples =
27\nvalue = [27, 0]'),
 Text(0.5596330275229358, 0.5588235294117647, 'x[12] <= 0.375\ngini =
0.49\nsamples = 7\nvalue = [4, 3]'),
 Text(0.5519072911636891, 0.5, 'gini = 0.0\nsamples = 3\nvalue = [3,
0]'),
 Text(0.5673587638821825, 0.5, 'x[2] <= 0.99\ngini = 0.375\nsamples =
4\nvalue = [1, 3]'),
 Text(0.5596330275229358, 0.4411764705882353, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.5750845002414292, 0.4411764705882353, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.6407532592950266, 0.6764705882352942, 'x[7] <= 0.758\ngini =
0.059\nsamples = 360\nvalue = [349, 11]'),
 Text(0.613713182037663, 0.6176470588235294, 'x[16] <= 0.06\ngini =
0.023\nsamples = 256\nvalue = [253, 3]'),
 Text(0.5905359729599228, 0.5588235294117647, 'x[7] <= 0.082\ngini =
0.26\nsamples = 13\nvalue = [11, 2]'),
 Text(0.582810236600676, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.5982617093191694, 0.5, 'x[3] <= 0.25\ngini = 0.153\nsamples =
12\nvalue = [11, 1]'),
 Text(0.5905359729599228, 0.4411764705882353, 'x[2] <= 0.716\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.582810236600676, 0.38235294117647056, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.5982617093191694, 0.38235294117647056, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.6059874456784162, 0.4411764705882353, 'gini = 0.0\nsamples =
10\nvalue = [10, 0]'),
 Text(0.6368903911154031, 0.5588235294117647, 'x[16] <= 0.154\ngini =
0.008\nsamples = 243\nvalue = [242, 1]'),
 Text(0.6291646547561565, 0.5, 'x[16] <= 0.151\ngini = 0.08\nsamples =
24\nvalue = [23, 1]'),
 Text(0.6214389183969097, 0.4411764705882353, 'gini = 0.0\nsamples =
23\nvalue = [23, 0]'),
 Text(0.6368903911154031, 0.4411764705882353, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.6446161274746499, 0.5, 'gini = 0.0\nsamples = 219\nvalue =
[219, 0]'),
 Text(0.6677933365523901, 0.6176470588235294, 'x[7] <= 0.762\ngini =
0.142\nsamples = 104\nvalue = [96, 8]'),
 Text(0.6600676001931434, 0.5588235294117647, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.6755190729116369, 0.5588235294117647, 'x[25] <= 0.867\ngini =
```

```
0.111\nsamples = 102\nvalue = [96, 6]'),
 Text(0.6600676001931434, 0.5, 'x[13] <= 0.5\ngini = 0.095\nsamples =
100\nvalue = [95, 5]'),
 Text(0.6523418638338967, 0.4411764705882353, 'x[2] <= 0.219\ngini =
0.206\nsamples = 43\nvalue = [38, 5]'),
 Text(0.6253017865765331, 0.38235294117647056, 'x[8] <= 0.833\ngini =
0.469\nsamples = 8\nvalue = [5, 3]'),
 Text(0.6175760502172863, 0.3235294117647059, 'gini = 0.0\nsamples =
4\nvalue = [4, 0]'),
 Text(0.6330275229357798, 0.3235294117647059, 'x[12] <= 0.688\ngini =
0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.6253017865765331, 0.2647058823529412, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.6407532592950266, 0.2647058823529412, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.6793819410912603, 0.38235294117647056, 'x[24] <= 0.639\ngini =
0.108\nsamples = 35\nvalue = [33, 2]'),
 Text(0.6639304683727668, 0.3235294117647059, 'x[19] <= 0.5\ngini =
0.059\nsamples = 33\nvalue = [32, 1]'),
 Text(0.65620473201352, 0.2647058823529412, 'gini = 0.0\nsamples = 31\
nvalue = [31, 0]'),
 Text(0.6716562047320135, 0.2647058823529412, 'x[10] <= 0.5\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.6639304683727668, 0.20588235294117646, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.6793819410912603, 0.20588235294117646, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.6948334138097537, 0.3235294117647059, 'x[7] <= 0.788\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.687107677450507, 0.2647058823529412, 'gini = 0.0\nsamples = 1\
nvalue = [1, 0]'),
 Text(0.7025591501690005, 0.2647058823529412, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.6677933365523901, 0.4411764705882353, 'gini = 0.0\nsamples =
57\nvalue = [57, 0]'),
 Text(0.6909705456301304, 0.5, 'x[2] <= 0.761\ngini = 0.5\nsamples =
2\nvalue = [1, 1]'),
 Text(0.6832448092708836, 0.4411764705882353, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.6986962819893772, 0.4411764705882353, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.8633510381458233, 0.8529411764705882, 'x[14] <= 0.75\ngini =
0.324\nsamples = 246\nvalue = [196, 50]'),
 Text(0.7653307580878802, 0.7941176470588235, 'x[15] <= 0.147\ngini =
0.202\nsamples = 175\nvalue = [155, 20]'),
 Text(0.7218734910671173, 0.7352941176470589, 'x[15] <= 0.14\ngini =
0.453\nsamples = 26\nvalue = [17, 9]'),
 Text(0.7141477547078706, 0.6764705882352942, 'x[0] <= 0.274\ngini =
0.386\nsamples = 23\nvalue = [17, 6]'),
```

```
 Text(0.6986962819893772, 0.6176470588235294, 'x[4] <= 0.375\ngini =
0.444\nsamples = 6\nvalue = [2, 4]'),
 Text(0.6909705456301304, 0.5588235294117647, 'gini = 0.0\nsamples =
4\nvalue = [0, 4]'),
 Text(0.7064220183486238, 0.5588235294117647, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.7295992274263641, 0.6176470588235294, 'x[22] <= 0.167\ngini =
0.208\nsamples = 17\nvalue = [15, 2]'),
 Text(0.7218734910671173, 0.5588235294117647, 'x[15] <= 0.111\ngini =
0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.7141477547078706, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.7295992274263641, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [0,
2]'),
 Text(0.7373249637856109, 0.5588235294117647, 'gini = 0.0\nsamples =
14\nvalue = [14, 0]'),
 Text(0.7295992274263641, 0.6764705882352942, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.8087880251086431, 0.7352941176470589, 'x[15] <= 0.992\ngini =
0.137\nsamples = 149\nvalue = [138, 11]'),
 Text(0.8010622887493964, 0.6764705882352942, 'x[4] <= 0.482\ngini =
0.126\nsamples = 148\nvalue = [138, 10]'),
 Text(0.7682279092225978, 0.6176470588235294, 'x[7] <= 0.978\ngini =
0.037\nsamples = 106\nvalue = [104, 2]'),
 Text(0.7527764365041043, 0.5588235294117647, 'x[24] <= 0.028\ngini =
0.019\nsamples = 103\nvalue = [102, 1]'),
 Text(0.7450507001448575, 0.5, 'x[5] <= 0.375\ngini = 0.153\nsamples =
12\nvalue = [11, 1]'),
 Text(0.7373249637856109, 0.4411764705882353, 'x[7] <= 0.2\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.7295992274263641, 0.38235294117647056, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.7450507001448575, 0.38235294117647056, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.7527764365041043, 0.4411764705882353, 'gini = 0.0\nsamples =
10\nvalue = [10, 0]'),
 Text(0.7605021728633511, 0.5, 'gini = 0.0\nsamples = 91\nvalue = [91,
0]'),
 Text(0.7836793819410912, 0.5588235294117647, 'x[5] <= 0.625\ngini =
0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.7759536455818445, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [2,
0]'),
 Text(0.791405118300338, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.8338966682761951, 0.6176470588235294, 'x[8] <= 0.167\ngini =
0.308\nsamples = 42\nvalue = [34, 8]'),
 Text(0.8145823273780782, 0.5588235294117647, 'x[2] <= 0.736\ngini =
0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.8068565910188314, 0.5, 'gini = 0.0\nsamples = 3\nvalue = [0,
3]'),
```

```
 Text(0.822308063737325, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.8532110091743119, 0.5588235294117647, 'x[0] <= 0.393\ngini =
0.229\nsamples = 38\nvalue = [33, 5]'),
 Text(0.8377595364558185, 0.5, 'x[8] <= 0.5\ngini = 0.5\nsamples = 6\
nvalue = [3, 3]'),
 Text(0.8300338000965717, 0.4411764705882353, 'x[17] <= 0.188\ngini =
0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.822308063737325, 0.38235294117647056, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.8377595364558185, 0.38235294117647056, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.8454852728150651, 0.4411764705882353, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.8686624818928054, 0.5, 'x[7] <= 0.992\ngini = 0.117\nsamples =
32\nvalue = [30, 2]'),
 Text(0.8609367455335587, 0.4411764705882353, 'x[22] <= 0.917\ngini =
0.062\nsamples = 31\nvalue = [30, 1]'),
 Text(0.8532110091743119, 0.38235294117647056, 'gini = 0.0\nsamples =
30\nvalue = [30, 0]'),
 Text(0.8686624818928054, 0.38235294117647056, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.8763882182520522, 0.4411764705882353, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.8165137614678899, 0.6764705882352942, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.9613713182037663, 0.7941176470588235, 'x[25] <= 0.367\ngini =
0.488\nsamples = 71\nvalue = [41, 30]'),
 Text(0.9459198454852729, 0.7352941176470589, 'x[2] <= 0.933\ngini =
0.458\nsamples = 62\nvalue = [40, 22]'),
 Text(0.9381941091260261, 0.6764705882352942, 'x[3] <= 0.75\ngini =
0.428\nsamples = 58\nvalue = [40, 18]'),
 Text(0.9072911636890391, 0.6176470588235294, 'x[7] <= 0.175\ngini =
0.32\nsamples = 35\nvalue = [28, 7]'),
 Text(0.8918396909705456, 0.5588235294117647, 'x[10] <= 0.364\ngini =
0.444\nsamples = 6\nvalue = [2, 4]'),
 Text(0.8841139546112989, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [2,
0]'),
 Text(0.8995654273297924, 0.5, 'gini = 0.0\nsamples = 4\nvalue = [0,
4]'),
 Text(0.9227426364075326, 0.5588235294117647, 'x[10] <= 0.829\ngini =
0.185\nsamples = 29\nvalue = [26, 3]'),
 Text(0.9150169000482858, 0.5, 'gini = 0.0\nsamples = 22\nvalue = [22,
0]'),
 Text(0.9304683727667793, 0.5, 'x[7] <= 0.517\ngini = 0.49\nsamples =
7\nvalue = [4, 3]'),
 Text(0.9227426364075326, 0.4411764705882353, 'gini = 0.0\nsamples =
4\nvalue = [4, 0]'),
 Text(0.9381941091260261, 0.4411764705882353, 'gini = 0.0\nsamples =
```

```
3\nvalue = [0, 3]'),
 Text(0.969097054563013, 0.6176470588235294, 'x[16] <= 0.272\ngini =
0.499\nsamples = 23\nvalue = [12, 11]'),
 Text(0.9536455818445195, 0.5588235294117647, 'x[11] <= 0.833\ngini =
0.219\nsamples = 8\nvalue = [7, 1]'),
 Text(0.9459198454852729, 0.5, 'gini = 0.0\nsamples = 7\nvalue = [7,
0]'),
 Text(0.9613713182037663, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.9845485272815065, 0.5588235294117647, 'x[10] <= 0.943\ngini =
0.444\nsamples = 15\nvalue = [5, 10]'),
 Text(0.9768227909222598, 0.5, 'x[1] <= 0.25\ngini = 0.278\nsamples =
12\nvalue = [2, 10]'),
 Text(0.969097054563013, 0.4411764705882353, 'gini = 0.0\nsamples = 1\
nvalue = [1, 0]'),
 Text(0.9845485272815065, 0.4411764705882353, 'x[2] <= 0.79\ngini =
0.165\nsamples = 11\nvalue = [1, 10]'),
 Text(0.9768227909222598, 0.38235294117647056, 'gini = 0.0\nsamples =
10\nvalue = [0, 10]'),
 Text(0.9922742636407532, 0.38235294117647056, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.9922742636407532, 0.5, 'gini = 0.0\nsamples = 3\nvalue = [3,
0]'),
 Text(0.9536455818445195, 0.6764705882352942, 'gini = 0.0\nsamples =
4\nvalue = [0, 4]'),
 Text(0.9768227909222598, 0.7352941176470589, 'x[1] <= 0.25\ngini =
0.198\nsamples = 9\nvalue = [1, 8]'),
 Text(0.969097054563013, 0.6764705882352942, 'gini = 0.0\nsamples = 1\
nvalue = [1, 0]'),
 Text(0.9845485272815065, 0.6764705882352942, 'gini = 0.0\nsamples =
8\nvalue = [0, 8]')]
```



```
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']

}

grid_search=GridSearchCV(estimator=dtc,param_grid=parameter,cv=5,scori
ng="accuracy")

grid_search.fit(x_train,y_train)
```

```
/Users/jewel/anaconda3/lib/python3.11/site-packages/sklearn/
model_selection/_validation.py:425: FitFailedWarning:
100 fits failed out of a total of 300.
The score on these train-test partitions for these parameters will be
set to nan.
If these failures are not expected, you can try to debug them by
setting error_score='raise'.

Below are more details about the failures:
--------------------------------------------------------------------
----------
100 fits failed with the following error:
Traceback (most recent call last):
```

```
  File
"/Users/jewel/anaconda3/lib/python3.11/site-packages/sklearn/model_sel
ection/_validation.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File
"/Users/jewel/anaconda3/lib/python3.11/site-packages/sklearn/base.py",
line 1144, in wrapper
    estimator._validate_params()
  File
"/Users/jewel/anaconda3/lib/python3.11/site-packages/sklearn/base.py",
line 637, in _validate_params
    validate_parameter_constraints(
  File
"/Users/jewel/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 DecisionTreeClassifier 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 'auto' instead.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
/Users/jewel/anaconda3/lib/python3.11/site-packages/sklearn/model_sele
ction/_search.py:976: UserWarning: One or more of the test scores are
non-finite: [       nan        nan 0.84013704 0.84269023 0.84013704
0.84013704
        nan        nan 0.8409881  0.83928597 0.83673278 0.83673278
        nan        nan 0.83333574 0.84439235 0.83673999 0.83928237
        nan        nan 0.84268662 0.84099531 0.83248107 0.84183916
        nan        nan 0.83420123 0.83248107 0.83503065 0.84012982
        nan        nan 0.84013704 0.84013704 0.84013704 0.84013704
        nan        nan 0.83333574 0.84013704 0.83503065 0.84013704
        nan        nan 0.84268301 0.84012982 0.84097367 0.83503065
        nan        nan 0.83928237 0.83928597 0.83503426 0.83927876
        nan        nan 0.84777497 0.83588532 0.83502344 0.84182834]
  warnings.warn(

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

grid_search.best_params_

{'criterion': 'entropy',
 'max_depth': 5,
 'max_features': 'sqrt',
 'splitter': 'best'}
```

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

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

```
pred3=dtc_cv.predict(x_test)
```

```
p3=accuracy_score(y_test,pred3)
accuracy_score(y_test,pred3)
```

```
0.8163265306122449
```

# Random Forest

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

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

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

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

```
/Users/jewel/anaconda3/lib/python3.11/site-packages/sklearn/
model_selection/_validation.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='raise'.

Below are more details about the failures:
--------------------------------------------------------------------
----------
50 fits failed with the following error:
Traceback (most recent call last):
  File
"/Users/jewel/anaconda3/lib/python3.11/site-packages/sklearn/model_sel
ection/_validation.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File
"/Users/jewel/anaconda3/lib/python3.11/site-packages/sklearn/base.py",
line 1144, in wrapper
```

```
    estimator._validate_params()
  File
"/Users/jewel/anaconda3/lib/python3.11/site-packages/sklearn/base.py",
line 637, in _validate_params
    validate_parameter_constraints(
  File
"/Users/jewel/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 RandomForestClassifier 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/jewel/anaconda3/lib/python3.11/site-packages/sklearn/model_sele
ction/_search.py:976: UserWarning: One or more of the test scores are
non-finite: [       nan 0.84439374 0.85034043 0.85289729 0.8503187
0.85796755
 0.85627264 0.85966247 0.85286832 0.85626539 0.86050268 0.85455599
 0.85796031 0.85454875        nan 0.84524844 0.85119513 0.85204259
 0.85796031 0.85201362 0.85624366 0.85538896 0.85880052 0.85796031
 0.85541069 0.85370129 0.85627264 0.85797479        nan 0.84440099
 0.84948573 0.85203535 0.85370853 0.85457048 0.85882225 0.86051717
 0.86051717 0.85540345 0.85878604 0.85457048 0.85795306 0.85712009
        nan 0.84524844 0.84777633 0.85202086 0.85031146 0.86051717
 0.85795306 0.85541069 0.86134289 0.85456323 0.85963349 0.85457772
 0.85202086 0.85708388        nan 0.84440099 0.84864552 0.85287556
 0.85543966 0.85883674 0.85625815 0.85709836 0.85541069 0.85540345
 0.85541069 0.85625815 0.8647617  0.85199189]
  warnings.warn(

GridSearchCV(cv=10, estimator=RandomForestClassifier(),
             param_grid=[{'max_depth': [10, 11, 12, 13, 14],
                          'max_features': [0, 1, 2, 3, 4, 5, 6, 7, 8,
9, 10, 11,
                                           12, 13]}],
             scoring='accuracy')

pred4=rfc_cv.predict(x_test)

p4=accuracy_score(y_test,pred4)
accuracy_score(y_test,pred4)

0.8571428571428571

#Evaluating the best model with the help of accuracy from each model
print("Logistic Regression: ",p)
print("Decesion Tree: ",p2)
```

```
print("Decesion Tree with grid search CV: ",p3)
print("Random Forest: ",p4)
```

```
Logistic Regression:  0.8843537414965986
Decesion Tree:  0.7619047619047619
Decesion Tree with grid search CV:  0.8163265306122449
Random Forest:  0.8571428571428571
```

From the above we can see that Logistic Regression model has the best accurancy for this dataset followed by random forest