NAME: K S H V SAI HARI KRISHNA

REG NO: 21BCE8069

1.Download the Employee Attrition Dataset

https://www.kaggle.com/datasets/patelprashant/employee-attrition

2.Perfrom Data Preprocessing

3.Model Building using Logistic Regression and Decision Tree and Random Forest

4.Calculate Performance metrics

```
#Import the Libraries.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#Importing the dataset.
df=pd.read_csv("Employee-Attrition.csv")
```

```
df.head()
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumbe
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7

5 rows × 35 columns

```
df.shape
```

```
(1470, 35)
```

```
df.Age.value_counts()
```

```
35    78
34    77
36    69
31    69
29    68
32    61
30    60
33    58
38    58
40    57
37    50
27    48
28    48
42    46
39    42
45    41
41    40
26    39
44    33
46    33
43    32
50    30
25    26
24    26
49    24
47    24
55    22
51    19
53    19
48    19
54    18
52    18
```

```
22    16
56    14
23    14
58    14
21    13
20    11
59    10
19     9
18     8
60     5
57     4
Name: Age, dtype: int64
```

```
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
```

```
df.describe()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	J
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	1470.000000	1470.000000	
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	2.721769	65.891156	
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	1.093082	20.329428	
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	1.000000	30.000000	
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	2.000000	48.000000	
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	3.000000	66.000000	
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	4.000000	83.750000	
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	4.000000	100.000000	

8 rows × 26 columns

```
#Checking for 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["YearsWithCurrManager"])
```

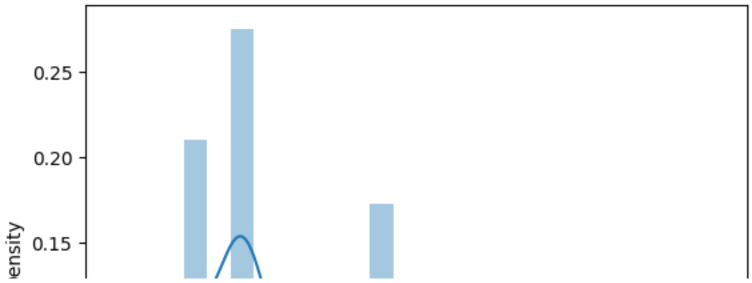
```
<ipython-input-12-71e8291be26b>:2: UserWarning:

`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["YearsWithCurrManager"])
<Axes: xlabel='YearsWithCurrManager', ylabel='Density'>
```



```
df.corr()
```

```
<ipython-input-13-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ver
  df.corr()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	Hc
Age	1.000000	0.010661	-0.001686	0.208034	NaN	-0.010145	0.010146	
DailyRate	0.010661	1.000000	-0.004985	-0.016806	NaN	-0.050990	0.018355	
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042	NaN	0.032916	-0.016075	
Education	0.208034	-0.016806	0.021042	1.000000	NaN	0.042070	-0.027128	
EmployeeCount	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
EmployeeNumber	-0.010145	-0.050990	0.032916	0.042070	NaN	1.000000	0.017621	
EnvironmentSatisfaction	0.010146	0.018355	-0.016075	-0.027128	NaN	0.017621	1.000000	
HourlyRate	0.024287	0.023381	0.031131	0.016775	NaN	0.035179	-0.049857	
JobInvolvement	0.029820	0.046135	0.008783	0.042438	NaN	-0.006888	-0.008278	
JobLevel	0.509604	0.002966	0.005303	0.101589	NaN	-0.018519	0.001212	
JobSatisfaction	-0.004892	0.030571	-0.003669	-0.011296	NaN	-0.046247	-0.006784	
MonthlyIncome	0.497855	0.007707	-0.017014	0.094961	NaN	-0.014829	-0.006259	
MonthlyRate	0.028051	-0.032182	0.027473	-0.026084	NaN	0.012648	0.037600	
NumCompaniesWorked	0.299635	0.038153	-0.029251	0.126317	NaN	-0.001251	0.012594	
PercentSalaryHike	0.003634	0.022704	0.040235	-0.011111	NaN	-0.012944	-0.031701	
PerformanceRating	0.001904	0.000473	0.027110	-0.024539	NaN	-0.020359	-0.029548	
RelationshipSatisfaction	0.053535	0.007846	0.006557	-0.009118	NaN	-0.069861	0.007665	
StandardHours	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
StockOptionLevel	0.037510	0.042143	0.044872	0.018422	NaN	0.062227	0.003432	
TotalWorkingYears	0.680381	0.014515	0.004628	0.148280	NaN	-0.014365	-0.002693	
TrainingTimesLastYear	-0.019621	0.002453	-0.036942	-0.025100	NaN	0.023603	-0.019359	
WorkLifeBalance	-0.021490	-0.037848	-0.026556	0.009819	NaN	0.010309	0.027627	
YearsAtCompany	0.311309	-0.034055	0.009508	0.069114	NaN	-0.011240	0.001458	
YearsInCurrentRole	0.212901	0.009932	0.018845	0.060236	NaN	-0.008416	0.018007	
YearsSinceLastPromotion	0.216513	-0.033229	0.010029	0.054254	NaN	-0.009019	0.016194	
YearsWithCurrManager	0.202089	-0.026363	0.014406	0.069065	NaN	-0.009197	-0.004999	

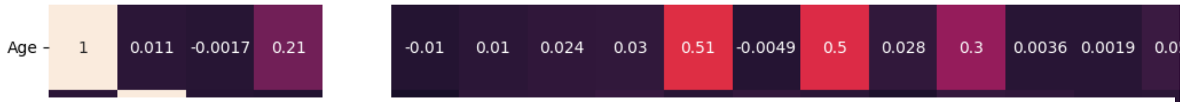
26 rows × 26 columns

```
df.head()
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5
					Research &					7

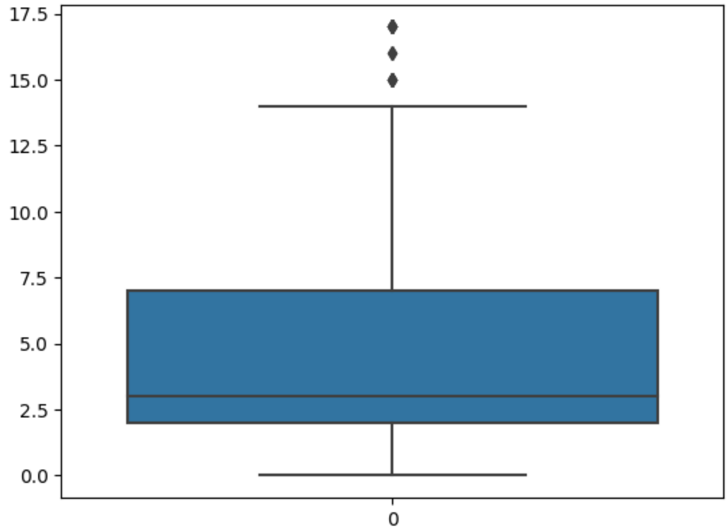
```
plt.subplots(figsize = (25,25))
sns.heatmap(df.corr(),annot=True)
```

```
<ipython-input-15-9329d5e70af4>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ver
  sns.heatmap(df.corr(),annot=True)
<Axes: >
```



```
sns.boxplot(df.YearsWithCurrManager)
```

```
<Axes: >
```



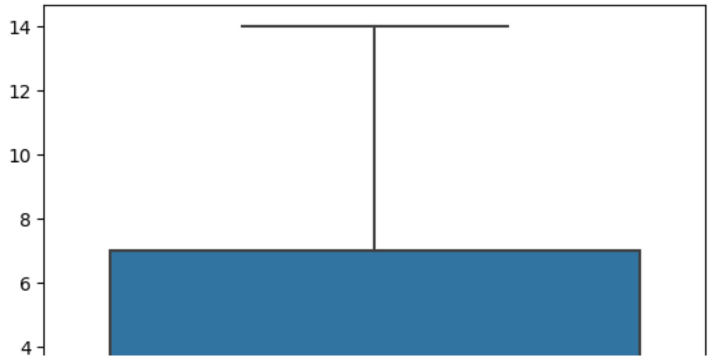
```
from scipy import stats
z_scores = np.abs(stats.zscore(df['YearsWithCurrManager']))
max_threshold=3
outliers = df['YearsWithCurrManager'][z_scores > max_threshold]

# Print and visualize the outliers
print("Outliers detected using Z-Score:")
print(outliers)
```

```
Outliers detected using Z-Score:
28      17
123     15
153     15
187     15
231     15
386     17
561     16
616     17
635     15
686     17
875     17
926     17
1078    17
1348    16
Name: YearsWithCurrManager, dtype: int64
```

```
q1 = df.YearsWithCurrManager.quantile(0.25)
q3 = df.YearsWithCurrManager.quantile(0.75)
print(q1)
print(q3)
upperlimit = q3+1.5*(q3-q1)
upperlimit
lowerlimit = q1-1.5*(q3-q1)
lowerlimit
df.median()
df["YearsWithCurrManager"]=np.where(df["YearsWithCurrManager"]>upperlimit,14,df['YearsWithCurrManager'])
sns.boxplot(df.YearsWithCurrManager)
```

```
2.0
7.0
<ipython-input-18-3a17581b0650>:9: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future v
  df.median()
<Axes: >
```



```
from scipy import stats
z_scores = np.abs(stats.zscore(df['YearsWithCurrManager']))
max_threshold=3
outliers = df['YearsWithCurrManager'][z_scores > max_threshold]

# Print and visualize the outliers
print("Outliers detected using Z-Score:")
print(outliers)
```

```
Outliers detected using Z-Score:
Series([], Name: YearsWithCurrManager, dtype: int64)
```

```
df.head()
```

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

5 rows × 35 columns

```
x=df.drop('Attrition',axis=1)
x.head()
```

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	Environm
0	41	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	
1	49	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	
2	37	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	
3	33	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	
4	27	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	

5 rows × 34 columns

```
y=df.Attrition
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.head()
x.Department    =le.fit_transform(x.Department  )
x.head()
x.EducationField    =le.fit_transform(x.EducationField  )
x.head()
x.Gender=le.fit_transform(x.Gender)
x.head()
x.JobRole   =le.fit_transform(x.JobRole )
x.head()
x.MaritalStatus =le.fit_transform(x.MaritalStatus   )
x.head()
x.Over18    =le.fit_transform(x.Over18  )
x.head()
x.OverTime  =le.fit_transform(x.OverTime    )
x.head()
```

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

5 rows × 34 columns

```
df.columns
```

```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate',
       'JobInvolvement', 'JobLevel', 'JobRole', 'JobSatisfaction',
       'MaritalStatus', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked',
       'Over18', 'OverTime', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

```
#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	Education	EducationField	EmployeeCount	EmployeeNumber	En
0	0.547619	1.0	0.715820	1.0	0.000000	0.25	0.2	0.0	0.000000	
1	0.738095	0.5	0.126700	0.5	0.250000	0.00	0.2	0.0	0.000484	
2	0.452381	1.0	0.909807	0.5	0.035714	0.25	0.8	0.0	0.001451	
3	0.357143	0.5	0.923407	0.5	0.071429	0.75	0.2	0.0	0.001935	
4	0.214286	1.0	0.350036	0.5	0.035714	0.00	0.6	0.0	0.002903	
...	...	...	...	...	...	...	...	...	...	
1465	0.428571	0.5	0.559771	0.5	0.785714	0.25	0.6	0.0	0.996613	
1466	0.500000	1.0	0.365784	0.5	0.178571	0.00	0.6	0.0	0.997097	
1467	0.214286	1.0	0.037938	0.5	0.107143	0.50	0.2	0.0	0.998065	
1468	0.738095	0.5	0.659270	1.0	0.035714	0.50	0.6	0.0	0.998549	
1469	0.380952	1.0	0.376521	0.5	0.250000	0.50	0.6	0.0	1.000000	

1470 rows × 34 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, 34), (294, 34), (1176,), (294,))
```

```
x_train.head()
```

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	En
1374	0.952381	1.0	0.360057	1.0	0.714286	0.50	0.2	0.0	0.937107	
1092	0.642857	1.0	0.607015	0.5	0.964286	0.50	1.0	0.0	0.747460	
768	0.523810	1.0	0.141732	1.0	0.892857	0.50	0.4	0.0	0.515239	
569	0.428571	0.0	0.953472	1.0	0.250000	0.75	0.2	0.0	0.381229	
911	0.166667	0.5	0.355762	1.0	0.821429	0.00	0.2	0.0	0.615385	

5 rows × 34 columns

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

```
model.fit(x_train,y_train)
pred=model.predict(x_test)
pred
```

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

```
#label encoding
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
y=le.fit_transform(y)
```

```
y_test
```

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

```
df
```

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

1470 rows × 35 columns

Evaluation of classification model

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

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

```
    0.8843537414965986
```

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

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

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

col_0	No	Yes
Attrition		
No	242	3
Yes	31	18

Roc-AUC curve

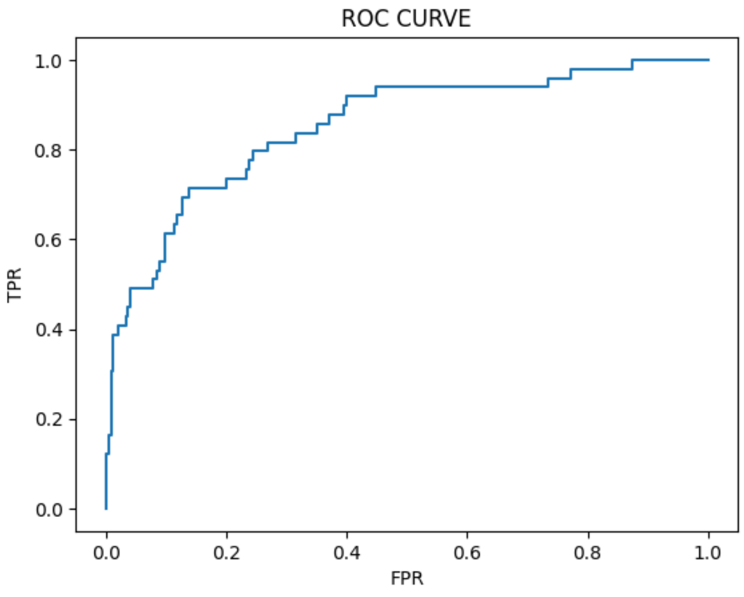
```
probability=model.predict_proba(x_test)[:,1]
probability
```

```
           0.06205401, 0.61414184, 0.07466397, 0.00797252, 0.39157785,
           0.05281564, 0.33160211, 0.02022395, 0.6671328 , 0.19419683,
           0.0335299 , 0.10954936, 0.17130578, 0.043804  , 0.2241511 ,
           0.23531373, 0.01475346, 0.06562592, 0.05019163, 0.59115162,
           0.44667993, 0.07401303, 0.0449937 , 0.67637047, 0.05859033,
           0.01545736, 0.03386798, 0.07021403, 0.1707141 , 0.07767295,
           0.04154894, 0.08312937, 0.06997437, 0.03567429, 0.05269126,
           0.05742727, 0.02144976, 0.01779053, 0.01301572, 0.02825292,
           0.50162054, 0.41541766, 0.00299378, 0.74315718, 0.51799699,
           0.09708281, 0.48942319, 0.07941138, 0.25720931, 0.66861063,
           0.26482373, 0.01970983, 0.30281497, 0.02858501, 0.16213966,
           0.02040161, 0.2173984 , 0.13768821, 0.03568054, 0.37558052,
           0.03010741, 0.29718154, 0.15832399, 0.10264349, 0.08700774,
           0.0815183 , 0.30943969, 0.08708969, 0.07442596, 0.12300414,
           0.00857215, 0.02394842, 0.13606932, 0.02587787, 0.03217004,
           0.0821409 , 0.00518749, 0.035308  , 0.03813342, 0.14270872,
           0.26418695, 0.16461435, 0.27401734, 0.24146954, 0.02119787,
           0.17774284, 0.34102562, 0.28338745, 0.06906981, 0.04948532,
           0.24465264, 0.74929682, 0.35691434, 0.01878265, 0.08772637,
           0.03239915, 0.05413857, 0.15215059, 0.07127406, 0.13828798,
           0.09342465, 0.04693869, 0.02494493, 0.15041914, 0.07133392,
           0.03025642, 0.05306455, 0.1165452 , 0.00872431, 0.01229042,
           0.17575238, 0.05005249, 0.09018395, 0.82857166, 0.03066995,
           0.0228189 , 0.00874605, 0.13496234, 0.16593413, 0.05060052,
           0.01520085, 0.29791945, 0.54919611, 0.33581407, 0.0469494 ,
           0.38773566, 0.61348127, 0.14171081, 0.07455884, 0.2409655 ,
           0.09528764, 0.06730943, 0.09797576, 0.20026612, 0.20053142,
           0.03046036, 0.14877431, 0.0036571 , 0.11146887, 0.15912883,
           0.06017571, 0.17964687, 0.06063618, 0.1199213 , 0.03284092,
           0.02688355, 0.06536903, 0.08335812, 0.01464284, 0.01536292,
           0.37701597, 0.01262506, 0.15004068, 0.80530948, 0.11655522,
           0.28461049, 0.17042029, 0.15392139, 0.02756879, 0.00599553,
           0.04142216, 0.09958411, 0.11567269, 0.10448555, 0.01830036,
           0.1444171 , 0.1048541 , 0.10079777, 0.05099176, 0.09183576,
           0.02893646, 0.09754427, 0.00516687, 0.75206394, 0.04227453,
           0.04018918, 0.37563319, 0.04457964, 0.72551665, 0.10583031,
           0.36656526, 0.38293703, 0.32923777, 0.05248015, 0.08216713,
           0.13748888, 0.04309097, 0.01429957, 0.2656631 , 0.06297408,
           0.16075744, 0.15388494, 0.67190498, 0.05834473, 0.28467369,
           0.04694404, 0.46237195, 0.00339026, 0.13927388, 0.02695884,
           0.12707414, 0.17395277, 0.0750947 , 0.10135673, 0.16496216,
           0.02583798, 0.01790826, 0.08850395, 0.02838351, 0.13795992,
           0.08655223, 0.22164621, 0.73379009, 0.17294814, 0.40907888,
           0.01503347, 0.11411826, 0.21412683, 0.32566668, 0.03366086,
           0.04472831, 0.32127248, 0.05442236, 0.0242917 , 0.16228044,
           0.32858438, 0.22879119, 0.00852736, 0.0798162 , 0.01140248,
           0.14102568, 0.29116266, 0.01282151, 0.17118076, 0.04051376,
           0.04165738, 0.42684273, 0.35009936, 0.0366853 , 0.11692325,
           0.37940034, 0.31562415, 0.79587005, 0.05488792, 0.21568794,
           0.06397987, 0.00569145, 0.66085682, 0.35796045, 0.37592133,
           0.3650533 , 0.03568965, 0.21192376, 0.05892118, 0.06428028,
           0.10143977, 0.00796354, 0.2678938 , 0.4288445 , 0.0652538 ,
           0.09309022, 0.01226927, 0.14314823, 0.04989664, 0.02304292,
           0.02508766, 0.06618985, 0.24272596, 0.26663754, 0.1979951 ,
           0.26504226, 0.01648205, 0.15826843, 0.08519882, 0.02669729,
           0.18757572, 0.00768502, 0.27928747, 0.0027473 , 0.02506718,
           0.22608608, 0.72428674, 0.07739605, 0.26575953])
```

```
#label encoding
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
y_test=le.fit_transform(y_test)
```

```
 # 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()
```



DecisionTreeClassifier

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

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

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

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

```
pred
```

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

```
y_test
```

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

```
df
```

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

Evaluation of classification model

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

```
#label encoding
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
y=le.fit_transform(y)
#label encoding
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
pred=le.fit_transform(pred)
```

```
y_test
```

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

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

```
0.7482993197278912
```

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

```
array([[203,  42],
       [ 32,  17]])
```

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

col_0	0	1
row_0		
0	203	42
1	32	17

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

```
              precision    recall  f1-score   support

           0       0.86      0.83      0.85       245
           1       0.29      0.35      0.31        49

    accuracy                           0.75       294
   macro avg       0.58      0.59      0.58       294
weighted avg       0.77      0.75      0.76       294
```

Roc-AUC curve

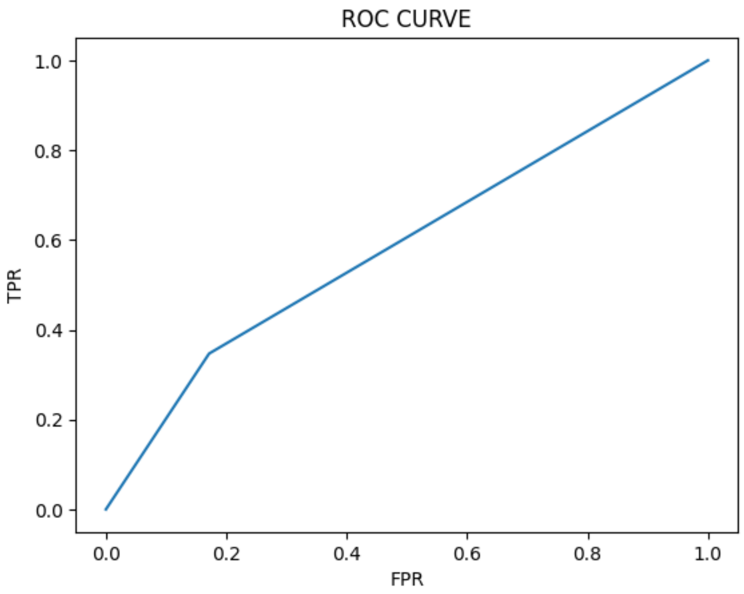
```
probability=dtc.predict_proba(x_test)[:,1]
```

```
probability
```

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

```
fpr,tpr,thresholds =  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(0.3291164340101523, 0.9722222222222222, 'x[27] <= 0.038\ngini =
0.269\nsamples = 1176\nvalue = [988, 188]'),
 Text(0.08121827411167512, 0.9166666666666666, 'x[16] <= 0.75\ngini =
0.5\nsamples = 78\nvalue = [39, 39]'),
 Text(0.050761421319796954, 0.8611111111111112, 'x[4] <= 0.554\ngini =
0.426\nsamples = 39\nvalue = [27, 12]'),
 Text(0.0338409475465313, 0.8055555555555556, 'x[15] <= 0.167\ngini =
0.312\nsamples = 31\nvalue = [25, 6]'),
 Text(0.02030456852791878, 0.75, 'x[21] <= 0.5\ngini = 0.49\nsamples = 7\nvalue =
[3, 4]'),
 Text(0.01353637901861252, 0.6944444444444444, 'x[22] <= 0.321\ngini =
0.375\nsamples = 4\nvalue = [3, 1]'),
 Text(0.00676818950930626, 0.6388888888888888, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
 Text(0.02030456852791878, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.02707275803722504, 0.6944444444444444, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.047377326565143825, 0.75, 'x[19] <= 0.056\ngini = 0.153\nsamples =
24\nvalue = [22, 2]'),
 Text(0.04060913705583756, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.05414551607445008, 0.6944444444444444, 'x[9] <= 0.167\ngini =
0.083\nsamples = 23\nvalue = [22, 1]'),
 Text(0.047377326565143825, 0.6388888888888888, 'x[0] <= 0.214\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.04060913705583756, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.05414551607445008, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.06091370558375635, 0.6388888888888888, 'gini = 0.0\nsamples = 21\nvalue =
[21, 0]'),
 Text(0.0676818950930626, 0.8055555555555556, 'x[8] <= 0.385\ngini =
0.375\nsamples = 8\nvalue = [2, 6]'),
 Text(0.06091370558375635, 0.75, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.07445008460236886, 0.75, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.1116751269035533, 0.8611111111111112, 'x[11] <= 0.364\ngini =
0.426\nsamples = 39\nvalue = [12, 27]'),
 Text(0.09475465313028765, 0.8055555555555556, 'x[29] <= 0.167\ngini =
0.133\nsamples = 14\nvalue = [1, 13]'),
 Text(0.08798646362098139, 0.75, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.10152284263959391, 0.75, 'gini = 0.0\nsamples = 13\nvalue = [0, 13]'),
 Text(0.12859560067681894, 0.8055555555555556, 'x[8] <= 0.105\ngini =
0.493\nsamples = 25\nvalue = [11, 14]'),
 Text(0.11505922165820642, 0.75, 'x[22] <= 0.464\ngini = 0.278\nsamples =
6\nvalue = [5, 1]'),
 Text(0.10829103214890017, 0.6944444444444444, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]'),
 Text(0.1218274111675127, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.14213197969543148, 0.75, 'x[15] <= 0.5\ngini = 0.432\nsamples = 19\nvalue
= [6, 13]'),
 Text(0.1353637901861252, 0.6944444444444444, 'gini = 0.0\nsamples = 7\nvalue =
[0, 7]'),
 Text(0.14890016920473773, 0.6944444444444444, 'x[6] <= 0.4\ngini = 0.5\nsamples
= 12\nvalue = [6, 6]'),
 Text(0.1353637901861252, 0.6388888888888888, 'x[3] <= 0.75\ngini =
0.278\nsamples = 6\nvalue = [5, 1]'),
 Text(0.12859560067681894, 0.5833333333333334, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]'),
 Text(0.14213197969543148, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.16243654822335024, 0.6388888888888888, 'x[8] <= 0.249\ngini =
0.278\nsamples = 6\nvalue = [1, 5]'),
 Text(0.155668358714044, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.1692047377326565, 0.5833333333333334, 'gini = 0.0\nsamples = 5\nvalue =
[0, 5]'),
 Text(0.5770145939086294, 0.9166666666666666, 'x[21] <= 0.5\ngini =
0.235\nsamples = 1098\nvalue = [949, 149]'),
 Text(0.3325401861252115, 0.8611111111111112, 'x[29] <= 0.167\ngini =
0.162\nsamples = 798\nvalue = [727, 71]'),
 Text(0.18274111675126903, 0.8055555555555556, 'x[8] <= 0.445\ngini =
0.38\nsamples = 47\nvalue = [35, 12]'),
 Text(0.1692047377326565, 0.75, 'x[16] <= 0.75\ngini = 0.1\nsamples = 19\nvalue =
[18, 1]'),
 Text(0.16243654822335024, 0.6944444444444444, 'gini = 0.0\nsamples = 18\nvalue =
[18, 0]'),
 Text(0.17597292724196278, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.19627749576988154, 0.75, 'x[17] <= 0.094\ngini = 0.477\nsamples =
28\nvalue = [17, 11]'),
 Text(0.1895093062605753, 0.6944444444444444, 'gini = 0.0\nsamples = 4\nvalue =
[0, 4]'),
 Text(0.20304568527918782, 0.6944444444444444, 'x[8] <= 0.524\ngini =
0.413\nsamples = 24\nvalue = [17, 7]'),
 Text(0.19627749576988154, 0.6388888888888888, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.2098138747884941, 0.6388888888888888, 'x[33] <= 0.393\ngini =
0.351\nsamples = 22\nvalue = [17, 5]'),
 Text(0.19627749576988154, 0.5833333333333334, 'x[2] <= 0.025\ngini =
0.133\nsamples = 14\nvalue = [13, 1]'),
 Text(0.1895093062605753, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.20304568527918782, 0.5277777777777778, 'gini = 0.0\nsamples = 13\nvalue =
[13, 0]'),
 Text(0.2233502538071066, 0.5833333333333334, 'x[2] <= 0.329\ngini = 0.5\nsamples
= 8\nvalue = [4, 4]'),
 Text(0.21658206429780033, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.23011844331641285, 0.5277777777777778, 'x[12] <= 0.333\ngini =
0.32\nsamples = 5\nvalue = [4, 1]'),
 Text(0.2233502538071066, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.23688663282571912, 0.4722222222222222, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0]'),
 Text(0.48233925549915396, 0.8055555555555556, 'x[30] <= 0.963\ngini =
0.145\nsamples = 751\nvalue = [692, 59]'),
 Text(0.4755710659898477, 0.75, 'x[30] <= 0.113\ngini = 0.143\nsamples =
750\nvalue = [692, 58]'),
 Text(0.35152284263959394, 0.6944444444444444, 'x[9] <= 0.167\ngini =
0.218\nsamples = 257\nvalue = [225, 32]'),
 Text(0.3096446700507614, 0.6388888888888888, 'x[33] <= 0.179\ngini =
0.355\nsamples = 65\nvalue = [50, 15]'),
 Text(0.2876480541455161, 0.5833333333333334, 'x[33] <= 0.036\ngini =
0.303\nsamples = 59\nvalue = [48, 11]'),
 Text(0.2639593908629442, 0.5277777777777778, 'x[12] <= 0.5\ngini =
0.463\nsamples = 22\nvalue = [14, 8]'),
 Text(0.25042301184433163, 0.4722222222222222, 'x[11] <= 0.179\ngini =
0.198\nsamples = 9\nvalue = [8, 1]'),
 Text(0.2436548223350254, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.2571912013536379, 0.4166666666666667, 'gini = 0.0\nsamples = 8\nvalue =
[8, 0]'),
 Text(0.27749576988155666, 0.4722222222222222, 'x[11] <= 0.4\ngini =
0.497\nsamples = 13\nvalue = [6, 7]'),
 Text(0.2707275803722504, 0.4166666666666667, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0]'),
 Text(0.28426395939086296, 0.4166666666666667, 'x[4] <= 0.286\ngini =
0.346\nsamples = 9\nvalue = [2, 7]'),
 Text(0.27749576988155666, 0.3611111111111111, 'x[0] <= 0.226\ngini =
0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.2707275803722504, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.28426395939086296, 0.3055555555555556, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.2910321489001692, 0.3611111111111111, 'gini = 0.0\nsamples = 6\nvalue =
[0, 6]'),
 Text(0.311336717428088, 0.5277777777777778, 'x[15] <= 0.167\ngini =
0.149\nsamples = 37\nvalue = [34, 3]'),
 Text(0.30456852791878175, 0.4722222222222222, 'x[29] <= 0.5\ngini = 0.5\nsamples
= 6\nvalue = [3, 3]'),
 Text(0.29780033840947545, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
 Text(0.311336717428088, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.31810490693739424, 0.4722222222222222, 'gini = 0.0\nsamples = 31\nvalue =
[31, 0]'),
 Text(0.3316412859560068, 0.5833333333333334, 'x[8] <= 0.065\ngini =
0.444\nsamples = 6\nvalue = [2, 4]'),
 Text(0.3248730964467005, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.338409475465313, 0.5277777777777778, 'gini = 0.0\nsamples = 4\nvalue =
[0, 4]'),
 Text(0.3934010152284264, 0.6388888888888888, 'x[0] <= 0.321\ngini =
0.161\nsamples = 192\nvalue = [175, 17]'),
 Text(0.3587140439932318, 0.5833333333333334, 'x[6] <= 0.1\ngini = 0.294\nsamples
= 67\nvalue = [55, 12]'),
 Text(0.35194585448392557, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.36548223350253806, 0.5277777777777778, 'x[29] <= 0.5\ngini =
0.26\nsamples = 65\nvalue = [55, 10]'),
 Text(0.34856175972927245, 0.4722222222222222, 'x[11] <= 0.679\ngini =
0.469\nsamples = 16\nvalue = [10, 6]'),
 Text(0.34179357021996615, 0.4166666666666667, 'x[6] <= 0.4\ngini =
0.444\nsamples = 9\nvalue = [3, 6]'),
 Text(0.3350253807106599, 0.3611111111111111, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.34856175972927245, 0.3611111111111111, 'x[2] <= 0.126\ngini =
0.245\nsamples = 7\nvalue = [1, 6]'),
 Text(0.34179357021996615, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.3553299492385787, 0.3055555555555556, 'gini = 0.0\nsamples = 6\nvalue =
[0, 6]'),
 Text(0.3553299492385787, 0.4166666666666667, 'gini = 0.0\nsamples = 7\nvalue =
[7, 0]'),
 Text(0.3824027072758037, 0.4722222222222222, 'x[2] <= 0.037\ngini =
0.15\nsamples = 49\nvalue = [45, 4]'),
 Text(0.3756345177664975, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.38917089678510997, 0.4166666666666667, 'x[2] <= 0.938\ngini =
 Text(0.38917089678510997, 0.4166666666666667, 'x[2] <= 0.938\ngini =
0.117\nsamples = 48\nvalue = [45, 3]'),
 Text(0.3824027072758037, 0.3611111111111111, 'x[5] <= 0.875\ngini =
0.081\nsamples = 47\nvalue = [45, 2]'),
 Text(0.3688663282571912, 0.3055555555555556, 'x[12] <= 0.167\ngini =
0.043\nsamples = 45\nvalue = [44, 1]'),
 Text(0.36209813874788493, 0.25, 'x[3] <= 0.75\ngini = 0.444\nsamples = 3\nvalue
= [2, 1]'),
 Text(0.3553299492385787, 0.19444444444444445, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.3688663282571912, 0.19444444444444445, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.3756345177664975, 0.25, 'gini = 0.0\nsamples = 42\nvalue = [42, 0]'),
 Text(0.39593908629441626, 0.3055555555555556, 'x[32] <= 0.1\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.38917089678510997, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.4027072758037225, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.39593908629441626, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.428087986463621, 0.5833333333333334, 'x[8] <= 0.022\ngini =
0.077\nsamples = 125\nvalue = [120, 5]'),
 Text(0.40947546531302875, 0.5277777777777778, 'x[2] <= 0.578\ngini =
0.5\nsamples = 4\nvalue = [2, 2]'),
 Text(0.4027072758037225, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.41624365482233505, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.4467005076142132, 0.5277777777777778, 'x[18] <= 0.968\ngini =
0.048\nsamples = 121\nvalue = [118, 3]'),
 Text(0.42978003384094754, 0.4722222222222222, 'x[2] <= 0.98\ngini =
0.033\nsamples = 118\nvalue = [116, 2]'),
 Text(0.41624365482233505, 0.4166666666666667, 'x[14] <= 0.938\ngini =
0.017\nsamples = 114\nvalue = [113, 1]'),
 Text(0.40947546531302875, 0.3611111111111111, 'gini = 0.0\nsamples = 107\nvalue
= [107, 0]'),
 Text(0.4230118443316413, 0.3611111111111111, 'x[16] <= 0.25\ngini =
0.245\nsamples = 7\nvalue = [6, 1]'),
 Text(0.41624365482233505, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.42978003384094754, 0.3055555555555556, 'gini = 0.0\nsamples = 6\nvalue =
[6, 0]'),
 Text(0.4433164128595601, 0.4166666666666667, 'x[1] <= 0.25\ngini =
0.375\nsamples = 4\nvalue = [3, 1]'),
 Text(0.4365482233502538, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.4500846023688663, 0.3611111111111111, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
 Text(0.46362098138747887, 0.4722222222222222, 'x[19] <= 0.278\ngini =
0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.45685279187817257, 0.4166666666666667, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.4703891708967851, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.5996192893401016, 0.6944444444444444, 'x[30] <= 0.787\ngini =
0.1\nsamples = 493\nvalue = [467, 26]'),
 Text(0.5647208121827412, 0.6388888888888888, 'x[15] <= 0.5\ngini =
0.094\nsamples = 486\nvalue = [462, 24]'),
 Text(0.5152284263959391, 0.5833333333333334, 'x[14] <= 0.938\ngini =
0.154\nsamples = 191\nvalue = [175, 16]'),
 Text(0.5084602368866328, 0.5277777777777778, 'x[18] <= 0.481\ngini =
0.145\nsamples = 190\nvalue = [175, 15]'),
 Text(0.4906937394247039, 0.4722222222222222, 'x[33] <= 0.964\ngini =
0.221\nsamples = 95\nvalue = [83, 12]'),
 Text(0.48392554991539766, 0.4166666666666667, 'x[18] <= 0.47\ngini =
0.207\nsamples = 94\nvalue = [83, 11]'),
 Text(0.47715736040609136, 0.3611111111111111, 'x[5] <= 0.375\ngini =
0.192\nsamples = 93\nvalue = [83, 10]'),
 Text(0.45516074450084604, 0.3055555555555556, 'x[6] <= 0.9\ngini =
0.363\nsamples = 21\nvalue = [16, 5]'),
 Text(0.44839255499153974, 0.25, 'x[17] <= 0.413\ngini = 0.266\nsamples =
19\nvalue = [16, 3]'),
 Text(0.43485617597292725, 0.19444444444444445, 'x[8] <= 0.215\ngini =
0.117\nsamples = 16\nvalue = [15, 1]'),
 Text(0.428087986463621, 0.1388888888888889, 'x[31] <= 0.417\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.4213197969543147, 0.08333333333333333, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.43485617597292725, 0.08333333333333333, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.4416243654822335, 0.1388888888888889, 'gini = 0.0\nsamples = 14\nvalue =
[14, 0]'),
 Text(0.4619289340101523, 0.19444444444444445, 'x[31] <= 0.556\ngini =
0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.45516074450084604, 0.1388888888888889, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.4686971235194585, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.4619289340101523, 0.25, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.49915397631133673, 0.3055555555555556, 'x[31] <= 0.139\ngini =
0.129\nsamples = 72\nvalue = [67, 5]'),
 Text(0.48223350253807107, 0.25, 'x[8] <= 0.68\ngini = 0.444\nsamples = 6\nvalue
= [4, 2]'),
 Text(0.4754653130287648, 0.19444444444444445, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0]'),
 Text(0.4890016920473773, 0.19444444444444445, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.5160744500846024, 0.25, 'x[11] <= 0.993\ngini = 0.087\nsamples =
66\nvalue = [63, 3]'),
 Text(0.5025380710659898, 0.19444444444444445, 'x[28] <= 0.583\ngini =
0.061\nsamples = 64\nvalue = [62, 2]'),
 Text(0.4957698815566836, 0.1388888888888889, 'gini = 0.0\nsamples = 51\nvalue =
[51, 0]'),
 Text(0.5093062605752962, 0.1388888888888889, 'x[9] <= 0.5\ngini = 0.26\nsamples
= 13\nvalue = [11, 2]'),
 Text(0.5025380710659898, 0.08333333333333333, 'x[17] <= 0.335\ngini =
0.5\nsamples = 4\nvalue = [2, 2]'),
 Text(0.4957698815566836, 0.027777777777777776, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.5093062605752962, 0.027777777777777776, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.5160744500846024, 0.08333333333333333, 'gini = 0.0\nsamples = 9\nvalue =
[9, 0]'),
 Text(0.5296108291032149, 0.19444444444444445, 'x[24] <= 0.5\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.5228426395939086, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.5363790186125211, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.4906937394247039, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.49746192893401014, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.5262267343485617, 0.4722222222222222, 'x[19] <= 0.5\ngini =
0.061\nsamples = 95\nvalue = [92, 3]'),
 Text(0.5194585448392555, 0.4166666666666667, 'gini = 0.0\nsamples = 76\nvalue =
[76, 0]'),
 Text(0.5329949238578681, 0.4166666666666667, 'x[8] <= 0.161\ngini =
0.266\nsamples = 19\nvalue = [16, 3]'),
 Text(0.5194585448392555, 0.3611111111111111, 'x[24] <= 0.833\ngini =
0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.5126903553299492, 0.3055555555555556, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.5262267343485617, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.5465313028764806, 0.3611111111111111, 'x[31] <= 0.639\ngini =
0.117\nsamples = 16\nvalue = [15, 1]'),
 Text(0.5397631133671743, 0.3055555555555556, 'gini = 0.0\nsamples = 14\nvalue =
[14, 0]'),
 Text(0.5532994923857868, 0.3055555555555556, 'x[19] <= 0.944\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
```

```
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']

}
```

```
      Text(0.5668358714043993, 0.4166666666666667, 'x[26] <= 0.5\ngini = 0.5\nsamples
```

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

```
     [0, 1]'),
```

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

```
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
```

```
    /usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
      warnings.warn(
    /usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
      warnings.warn(
    /usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
```

```
grid_search.best_params_
```

```
    {'criterion': 'gini',
     'max_depth': 3,
     'max_features': 'auto',
     'splitter': 'random'}

    /usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.
```

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

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

     ▸ estimator: DecisionTreeClassifier
```

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

```
#label encoding
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
y=le.fit_transform(y)
#label encoding
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
pred=le.fit_transform(pred)
```

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

```
                  precision    recall  f1-score   support

               0       0.85      0.97      0.90       245
               1       0.43      0.12      0.19        49

        accuracy                           0.83       294
       macro avg       0.64      0.54      0.55       294
    weighted avg       0.78      0.83      0.78       294
```

RandomForestClassifier

```
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)
```

```
    /usr/local/lib/python3.10/dist-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 "/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_validation.py", line 686, in _fit_and_score
        estimator.fit(X_train, y_train, **fit_params)
      File "/usr/local/lib/python3.10/dist-packages/sklearn/ensemble/_forest.py", line 340, in fit
        self._validate_params()
      File "/usr/local/lib/python3.10/dist-packages/sklearn/base.py", line 600, in _validate_params
        validate_parameter_constraints(
      File "/usr/local/lib/python3.10/dist-packages/sklearn/utils/_param_validation.py", line 97, in validate_parameter_constraints
        raise InvalidParameterError(
```

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

```
    /usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_search.py:952: UserWarning: One or more of the test scores are non-
```

```
#label encoding
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
y=le.fit_transform(y)
#label encoding
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
pred=le.fit_transform(pred)
```

```
     0.85458496 0.85712009 0.86307403 0.85884398 0.85970592 0.85798928
```

```
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
```

```
rfc_cv.best_params_
```

```
    {'max_depth': 12, 'max_features': 9}
```