IMPORTING THE NECESSARY LIBRARIES

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

importing the datase

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

```
df.head()
```

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

5 rows × 35 columns

```
df.shape
```

```
(1470, 35)
```

```
df.corr()
```

```
<ipython-input-58-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in D
  df.corr()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount
Age	1.000000	0.010661	-0.001686	0.208034	NaN
DailyRate	0.010661	1.000000	-0.004985	-0.016806	NaN
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042	NaN
Education	0.208034	-0.016806	0.021042	1.000000	NaN
EmployeeCount	NaN	NaN	NaN	NaN	NaN
EmployeeNumber	-0.010145	-0.050990	0.032916	0.042070	NaN
EnvironmentSatisfaction	0.010146	0.018355	-0.016075	-0.027128	NaN

```
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()
```

.on	EmployeeCount	EmployeeNu
00	1470.0	1470.00
25	1.0	1024.86
65	0.0	602.02
00	1.0	1.00
00	1.0	491.25
00	1.0	1020.50
00	1.0	1555.75
00	1.0	2068.00

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
```

```
df.describe()
```

	Age	DistanceFromHome	EnvironmentSatisfaction	JobInvolvement	JobSatisf
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.
mean	36.923810	9.192517	2.721769	2.729932	2.
std	9.135373	8.106864	1.093082	0.711561	1.
min	18.000000	1.000000	1.000000	1.000000	1.
25%	30.000000	2.000000	2.000000	2.000000	2.
50%	36.000000	7.000000	3.000000	3.000000	3.
75%	43.000000	14.000000	4.000000	3.000000	4.
max	60.000000	29.000000	4.000000	4.000000	4.

dropping all the unnecessary columns

Attrition is the dependent column

```
df.drop(['BusinessTravel','DailyRate','Department','Education','EducationField','EmployeeNumber','StandardHours','NumCompaniesWorked','JobLev
```

```
df.describe()
```

	Age	DistanceFromHome	EnvironmentSatisfaction	JobInvolvement	JobSatisf
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.
mean	36.923810	9.192517	2.721769	2.729932	2.
std	9.135373	8.106864	1.093082	0.711561	1.
min	18.000000	1.000000	1.000000	1.000000	1.
25%	30.000000	2.000000	2.000000	2.000000	2.
50%	36.000000	7.000000	3.000000	3.000000	3.
75%	43.000000	14.000000	4.000000	3.000000	4.
max	60.000000	29.000000	4.000000	4.000000	4.

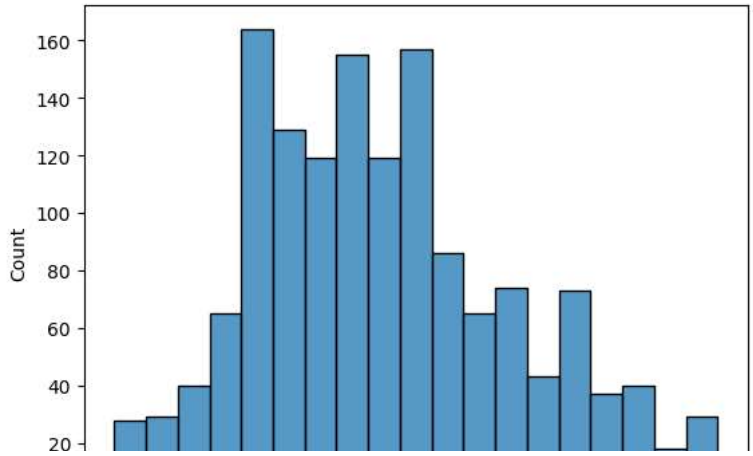
```
df.isnull().any()
```

```
Age                       False
Attrition                 False
DistanceFromHome          False
EnvironmentSatisfaction   False
Gender                    False
JobInvolvement            False
JobSatisfaction           False
MaritalStatus             False
MonthlyIncome             False
OverTime                  False
PercentSalaryHike         False
TotalWorkingYears         False
WorkLifeBalance           False
YearsInCurrentRole        False
YearsSinceLastPromotion   False
dtype: bool
```

DATA VISUALIZATION

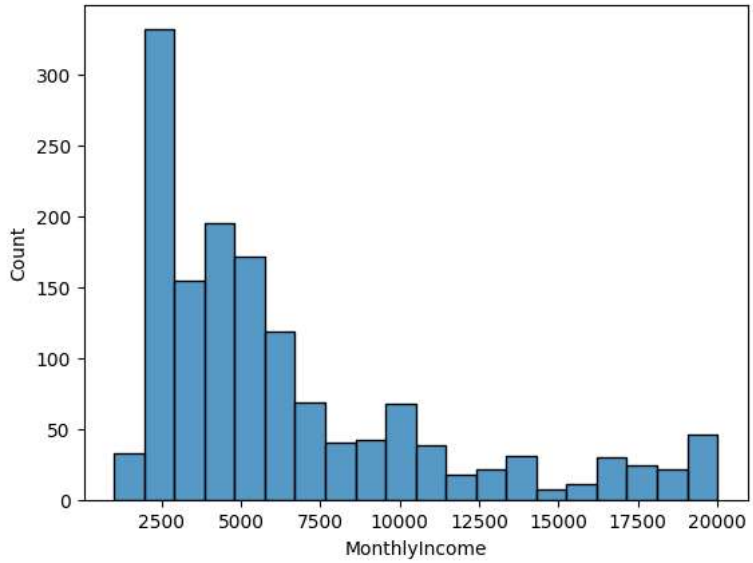
```
# Univariate Analysis
sns.histplot(df["Age"])
```

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



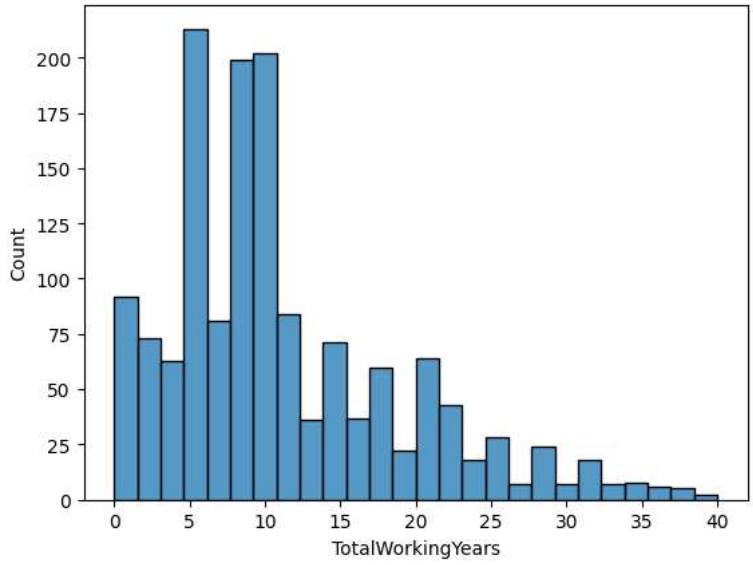

```
sns.histplot(df.MonthlyIncome)
```

```
<Axes: xlabel='MonthlyIncome', ylabel='Count'>
```



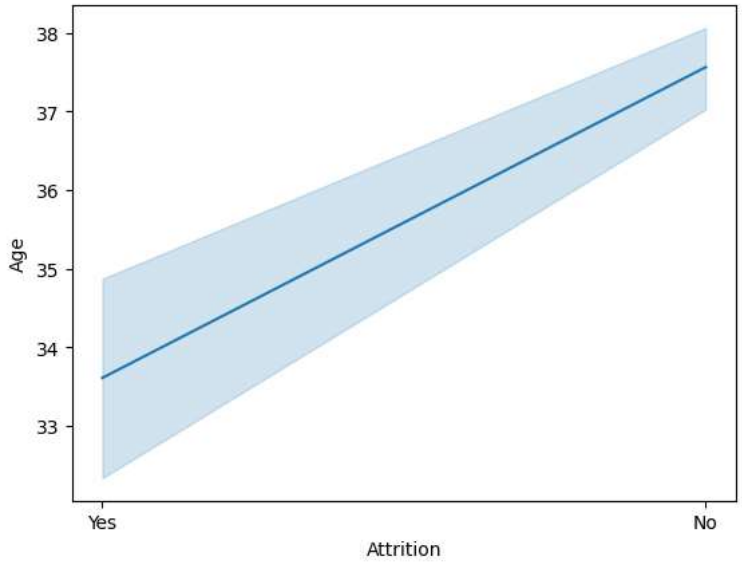

```
sns.histplot(df.TotalWorkingYears)
```

```
<Axes: xlabel='TotalWorkingYears', ylabel='Count'>
```



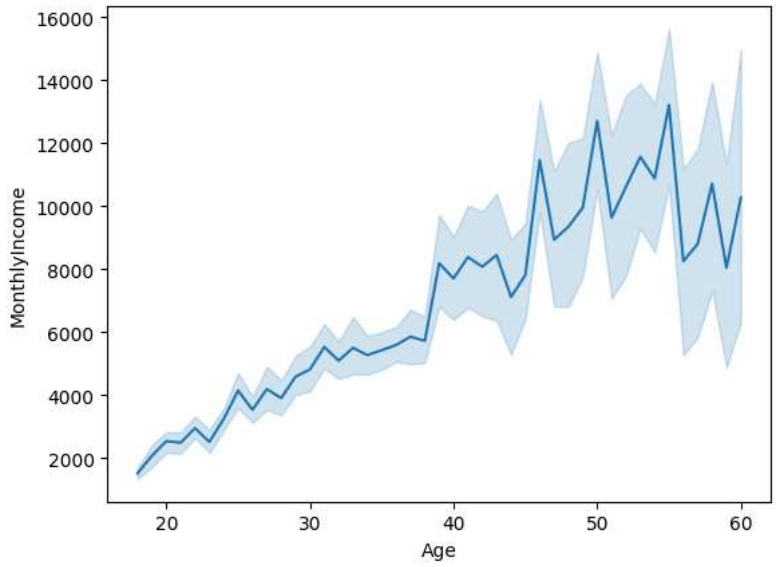

```
# Bivariate Analysis
sns.lineplot(x = "Attrition", y = "Age", data = df)
```

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




```
sns.lineplot(x = "Age", y = "MonthlyIncome", data = df)
```

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




```
sns.lineplot(x = "TotalWorkingYears", y = "MonthlyIncome", data = df)
```

```
<Axes: xlabel='TotalWorkingYears', ylabel='MonthlyIncome'>
```

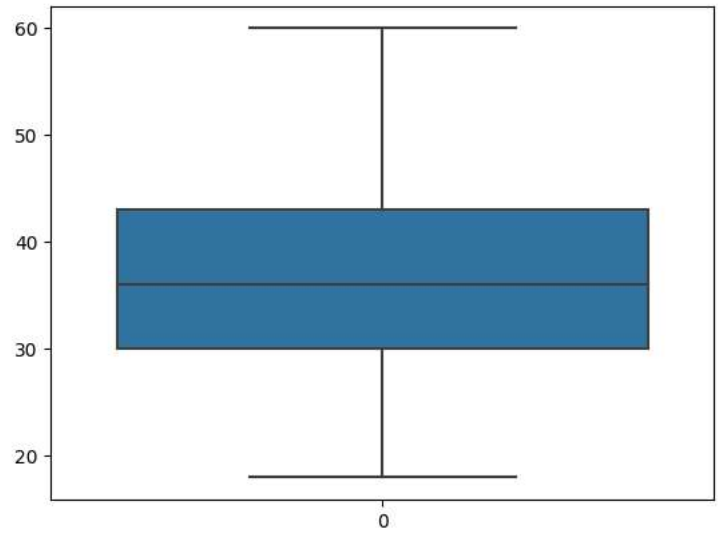
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 15 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   Age                      1470 non-null   int64 
 1   Attrition                1470 non-null   object
 2   DistanceFromHome         1470 non-null   int64 
 3   EnvironmentSatisfaction  1470 non-null   int64 
 4   Gender                   1470 non-null   object
 5   JobInvolvement           1470 non-null   int64 
 6   JobSatisfaction          1470 non-null   int64 
 7   MaritalStatus            1470 non-null   object
 8   MonthlyIncome            1470 non-null   int64 
 9   OverTime                 1470 non-null   object
 10  PercentSalaryHike        1470 non-null   int64 
 11  TotalWorkingYears        1470 non-null   int64 
 12  WorkLifeBalance          1470 non-null   int64 
 13  YearsInCurrentRole       1470 non-null   int64 
 14  YearsSinceLastPromotion  1470 non-null   int64 
dtypes: int64(11), object(4)
memory usage: 172.4+ KB
```

OUTLIER DETECTION

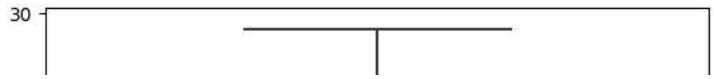
```
sns.boxplot(df.Age)
```

```
<Axes: >
```



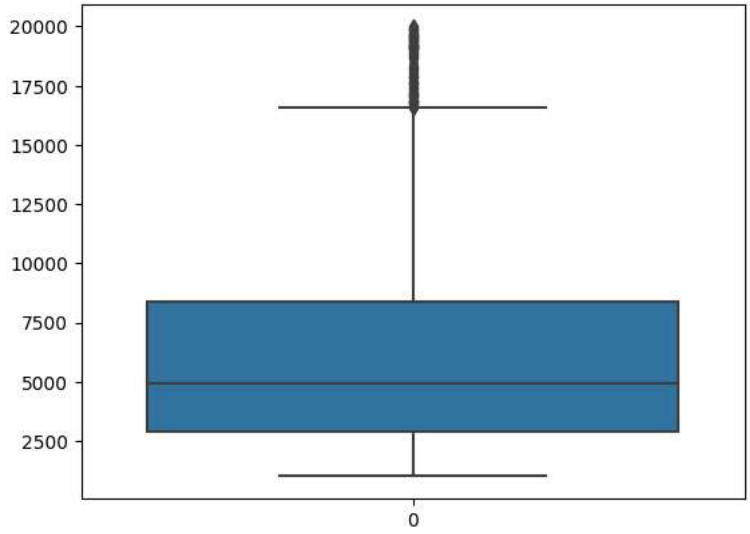

```
sns.boxplot(df.DistanceFromHome)
```

```
<Axes: >
```




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

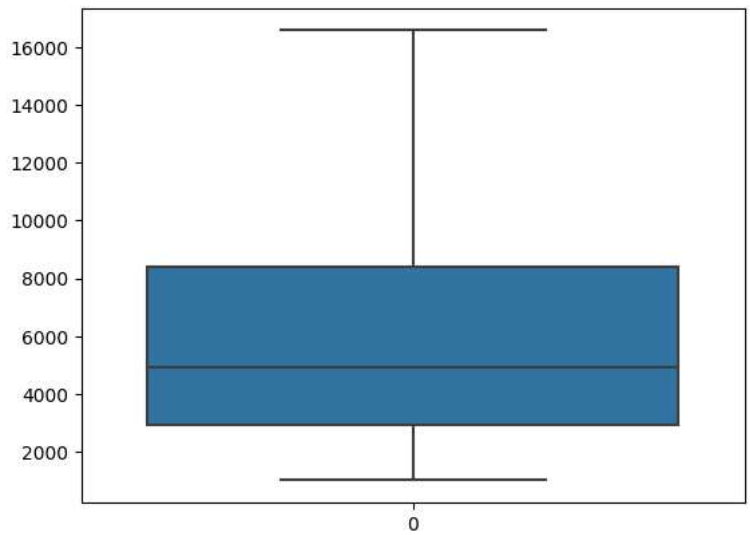
```
<Axes: >
```




```
Q1 = df['MonthlyIncome'].quantile(0.25)
Q3 = df['MonthlyIncome'].quantile(0.75)
IQR = Q3 - Q1
d = 1.5
lower_lim = Q1 -(d*IQR)
upper_lim = Q3 + (d*IQR)
df['MonthlyIncome']=np.where(df['MonthlyIncome']>upper_lim,upper_lim,np.where(df['MonthlyIncome']<lower_lim,lower_lim,df['MonthlyIncome']))
```

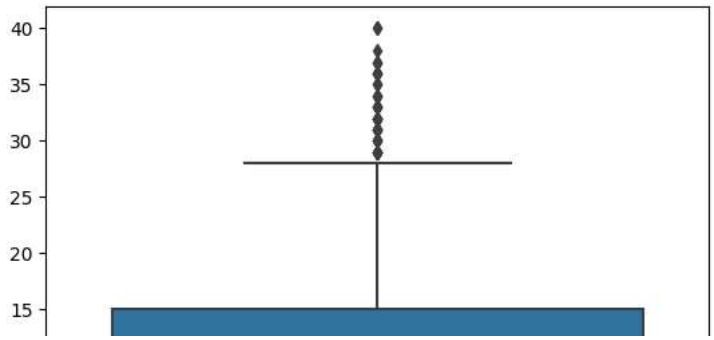
```
sns.boxplot(df.MonthlyIncome)
```

```
<Axes: >
```




```
sns.boxplot(df.TotalWorkingYears)
```

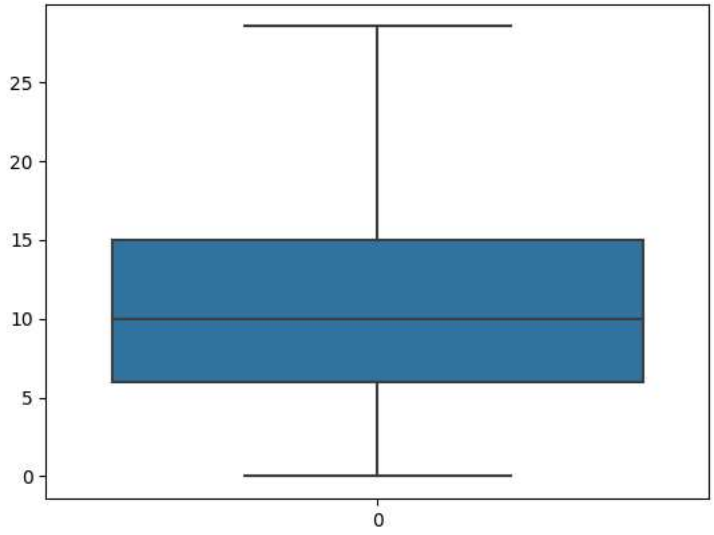
```
<Axes: >
```




```
Q1 = df['TotalWorkingYears'].quantile(0.25)
Q3 = df['TotalWorkingYears'].quantile(0.75)
IQR = Q3 - Q1
d = 1.5
lower_lim = Q1 -(d*IQR)
upper_lim = Q3 + (d*IQR)
df['TotalWorkingYears']=np.where(df['TotalWorkingYears']>upper_lim,upper_lim,np.where(df['TotalWorkingYears']<lower_lim,lower_lim,df['TotalWo
```

```
sns.boxplot(df.TotalWorkingYears)
```

```
<Axes: >
```



Label Encoding

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
columns = ['Attrition', 'Gender','MaritalStatus','OverTime']
df[columns] = df[columns].apply(le.fit_transform)
```

Splitting the dependent and independent variable

```
x = df.drop(columns = "Attrition")
y = df["Attrition"]
```

```
x.head()
```

```
   Age  DistanceFromHome  EnvironmentSatisfaction  Gender  JobInvolvement  JobSatisfacti
```

```
y.head()
```

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

feature scaling for logistic regression

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

Splitting into train and test data

```
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)
```

Model Building --- Logistic Regression

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

```
df.Attrition.value_counts().plot(kind = "pie", autopct = "%1.1f%%")
```

```
<Axes: ylabel='Attrition'>
```

Evaluation of the classification

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

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

```
0.8707482993197279
```

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

```
array([[241,   4],
       [ 34,  15]])
```

```
pd.crosstab(y_test, pred) # Visualized confusion matrix
```

col_0	0	1
Attrition		
0	241	4
1	34	15

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

```
                  precision    recall  f1-score   support

               0       0.88      0.98      0.93       245
               1       0.79      0.31      0.44        49

        accuracy                           0.87       294
       macro avg       0.83      0.64      0.68       294
    weighted avg       0.86      0.87      0.85       294
```

```
probability=model.predict_proba(x_test)[:,1] # probability of all the yes values
probability
```

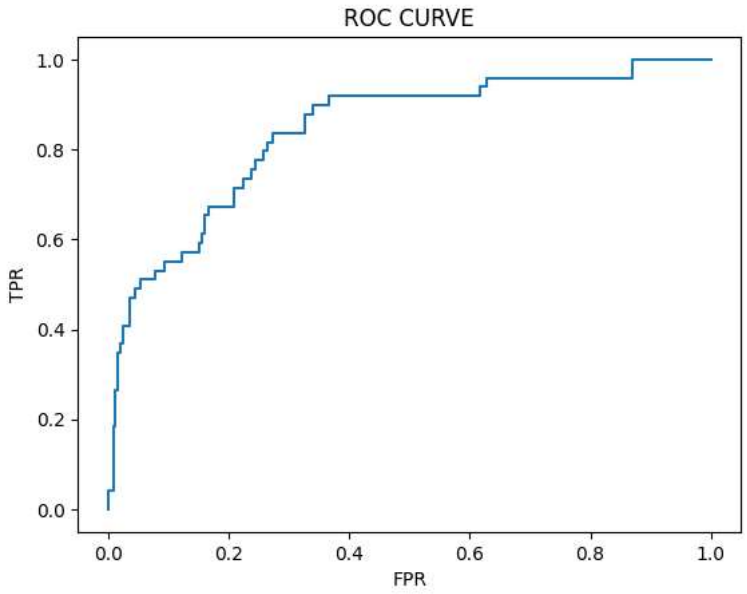
```
            0.04599096, 0.61773815, 0.06023599, 0.00959877, 0.27968035,
            0.0842562 , 0.30259009, 0.02538328, 0.69789422, 0.20782178,
            0.0515761 , 0.1464348 , 0.24578231, 0.02447109, 0.22122421,
            0.31186016, 0.02428186, 0.06830356, 0.05988681, 0.42919823,
            0.3299274 , 0.11381061, 0.0466628 , 0.71363351, 0.04911082,
            0.02187439, 0.16762865, 0.05888231, 0.13255516, 0.10796323,
            0.07749958, 0.07697303, 0.10776168, 0.06121553, 0.0612211 ,
            0.06532578, 0.02067614, 0.01888257, 0.02843847, 0.04773017,
            0.42121334, 0.2174884 , 0.01128639, 0.7261964 , 0.44541614,
            0.14534399, 0.57328758, 0.09805092, 0.19111059, 0.5744568 ,
            0.16928395, 0.01950284, 0.16877633, 0.08187363, 0.09909516,
            0.0599772 , 0.19181103, 0.09649451, 0.04659456, 0.37580431,
            0.03868532, 0.18202413, 0.23016278, 0.14492632, 0.15522114,
            0.08285658, 0.32017969, 0.04660827, 0.09075653, 0.05204234,
            0.11079437, 0.10724618, 0.09165732, 0.12238473, 0.04740504,
            0.0132515 , 0.03403514, 0.20647038, 0.02810751, 0.03385499,
            0.0944743 , 0.00750029, 0.0802044 , 0.01560663, 0.31123987,
            0.45351263, 0.12244881, 0.19410278, 0.24514104, 0.02099799,
            0.1780486 , 0.32882673, 0.26789763, 0.119946  , 0.07207794,
            0.20272913, 0.51473141, 0.41198323, 0.02070214, 0.06195954,
            0.04716694, 0.0654499 , 0.24905305, 0.03763038, 0.26472447,
            0.19621147, 0.1002986 , 0.02248279, 0.18463981, 0.10535546,
            0.04151099, 0.08150019, 0.09715958, 0.01831639, 0.02532645,
            0.13620907, 0.07633671, 0.08716973, 0.69723351, 0.02840483,
            0.01610639, 0.01600085, 0.09841706, 0.08542302, 0.06429813,
            0.02756709, 0.19868174, 0.45278515, 0.18086062, 0.04916228,
            0.44738387, 0.49569043, 0.33383642, 0.15580932, 0.25027566,
            0.1022055 , 0.08073811, 0.11339104, 0.18469605, 0.2565929 ,
            0.02326258, 0.13941762, 0.00760761, 0.12854839, 0.12618728,
            0.1393614 , 0.34844187, 0.04589769, 0.18069311, 0.03729283,
            0.04973812, 0.07705865, 0.05207065, 0.05253495, 0.02416478,
            0.46462668, 0.02072513, 0.10699699, 0.77590472, 0.11508102,
```

```
0.03544475, 0.05941833, 0.11161862, 0.05491243, 0.0231996 ,
0.20827328, 0.11700163, 0.13628483, 0.06954993, 0.06988645,
0.02941673, 0.0837539 , 0.0109935 , 0.67691392, 0.03085246,
0.08857201, 0.26988523, 0.03197149, 0.57008109, 0.12436864,
0.43905377, 0.52585167, 0.28197513, 0.05744372, 0.12118214,
0.10946615, 0.03848744, 0.01368207, 0.37489003, 0.08839392,
0.16090726, 0.11710773, 0.41477659, 0.10060868, 0.11785707,
0.03996752, 0.58684823, 0.00428713, 0.06358941, 0.04534226,
0.05625275, 0.2142873 , 0.06301476, 0.11662493, 0.1874911 ,
0.03163527, 0.01841097, 0.11747885, 0.02751903, 0.2651188 ,
0.07446389, 0.13519759, 0.70186102, 0.08672511, 0.38814187,
0.02949061, 0.06394048, 0.15147126, 0.24924537, 0.04506879,
0.03425287, 0.26362348, 0.09956386, 0.01954104, 0.14287163,
0.28912133, 0.06960402, 0.01740465, 0.09153147, 0.00937296,
0.23841396, 0.18150312, 0.01121302, 0.30189644, 0.052956  ,
0.03935999, 0.52911812, 0.40327884, 0.06879774, 0.21014479,
0.17300474, 0.30658594, 0.67374806, 0.05044605, 0.21725974,
0.05178539, 0.02154401, 0.64069311, 0.15948953, 0.16032123,
0.4344616 , 0.04451498, 0.23354616, 0.04093299, 0.05339782,
0.0866061 , 0.01431701, 0.40213906, 0.41223774, 0.1501187 ,
0.11879022, 0.01374278, 0.11379516, 0.02103421, 0.02919475,
0.04201696, 0.10468593, 0.30030105, 0.15293258, 0.23814858,
0.24051204, 0.04890163, 0.14968683, 0.08236902, 0.06229793,
0.32440277, 0.01595591, 0.4437252 , 0.00410957, 0.05043896,
0.28780428, 0.5354455 , 0.03799566, 0.30691028])
```

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()
```



Model Building --- Decision Tree

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

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

DecisionTreeClassifier
DecisionTreeClassifier()

```
y_pred1 = dtc.predict(x_test)
```

```
accuracy_score(y_test, y_pred1)
```

```
0.7857142857142857
```

```
confusion_matrix(y_test, y_pred1)
```

```
array([[213,  32],
       [ 31,  18]])
```

```
pd.crosstab(y_test, y_pred1)
```

col_0	0	1
Attrition		
0	213	32
1	31	18

```
print(classification_report(y_test, y_pred1))
```

```
                   precision    recall  f1-score   support

               0       0.87      0.87      0.87       245
               1       0.36      0.37      0.36        49

        accuracy                           0.79       294
       macro avg       0.62      0.62      0.62       294
    weighted avg       0.79      0.79      0.79       294
```

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

```
 Text(0.06619151840782256, 0.38333333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.06566347469220246, 0.6388888888888888, 'gini = 0.0\nsamples = 20\nvalue =
[20, 0]'),
 Text(0.06566347469220246, 0.8055555555555556, 'x[9] <= 0.679\ngini = 0.375\nsamples
= 8\nvalue = [2, 6]'),
 Text(0.060191518467852256, 0.75, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.07113543091655267, 0.75, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.12038303693570451, 0.8611111111111112, 'x[12] <= 0.028\ngini =
0.426\nsamples = 39\nvalue = [12, 27]'),
 Text(0.11491108071135431, 0.8055555555555556, 'x[0] <= 0.369\ngini = 0.394\nsamples
= 37\nvalue = [10, 27]'),
 Text(0.1094391244870041, 0.75, 'x[7] <= 0.119\ngini = 0.353\nsamples = 35\nvalue =
[8, 27]'),
 Text(0.09302325581395349, 0.6944444444444444, 'x[10] <= 0.018\ngini =
0.278\nsamples = 30\nvalue = [5, 25]'),
 Text(0.07660738714090287, 0.6388888888888888, 'x[7] <= 0.02\ngini = 0.494\nsamples
= 9\nvalue = [4, 5]'),
 Text(0.07113543091655267, 0.5833333333333334, 'gini = 0.0\nsamples = 3\nvalue = [3,
0]'),
 Text(0.08207934336525308, 0.5833333333333334, 'x[1] <= 0.036\ngini = 0.278\nsamples
= 6\nvalue = [1, 5]'),
 Text(0.07660738714090287, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.08755129958960328, 0.5277777777777778, 'gini = 0.0\nsamples = 5\nvalue = [0,
5]'),
 Text(0.1094391244870041, 0.6388888888888888, 'x[0] <= 0.036\ngini = 0.091\nsamples
= 21\nvalue = [1, 20]'),
 Text(0.1039671682626539, 0.5833333333333334, 'x[4] <= 0.5\ngini = 0.32\nsamples =
5\nvalue = [1, 4]'),
 Text(0.09849521203830369, 0.5277777777777778, 'gini = 0.0\nsamples = 4\nvalue = [0,
4]'),
 Text(0.1094391244870041, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.11491108071135431, 0.5833333333333334, 'gini = 0.0\nsamples = 16\nvalue =
[0, 16]'),
 Text(0.12585499316005472, 0.6944444444444444, 'x[9] <= 0.393\ngini = 0.48\nsamples
= 5\nvalue = [3, 2]'),
 Text(0.12038303693570451, 0.6388888888888888, 'gini = 0.0\nsamples = 3\nvalue = [3,
0]'),
 Text(0.13132694938440492, 0.6388888888888888, 'gini = 0.0\nsamples = 2\nvalue = [0,
2]'),
 Text(0.12038303693570451, 0.75, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.12585499316005472, 0.8055555555555556, 'gini = 0.0\nsamples = 2\nvalue = [2,
0]'),
 Text(0.5524164244186046, 0.9166666666666666, 'x[8] <= 0.5\ngini = 0.235\nsamples =
1098\nvalue = [949, 149]'),
 Text(0.2756497948016416, 0.8611111111111112, 'x[11] <= 0.167\ngini = 0.162\nsamples
= 798\nvalue = [727, 71]'),
 Text(0.16552667578659372, 0.8055555555555556, 'x[7] <= 0.107\ngini = 0.38\nsamples
= 47\nvalue = [35, 12]'),
 Text(0.14227086183310533, 0.75, 'x[2] <= 0.167\ngini = 0.444\nsamples = 6\nvalue =
```

```
⌊2, 4⌋'),
 Text(0.13679890560875513, 0.6944444444444444, 'gini = 0.0\nsamples = 3\nvalue = [0,
3]'),
 Text(0.14774281805745554, 0.6944444444444444, 'x[13] <= 0.333\ngini =
0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.14227086183310533, 0.6388888888888888, 'gini = 0.0\nsamples = 2\nvalue = [2,
0]'),
 Text(0.15321477428180574, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.18878248974008208, 0.75, 'x[13] <= 0.4\ngini = 0.314\nsamples = 41\nvalue =
[33, 8]'),
 Text(0.17510259917920656, 0.6944444444444444, 'x[1] <= 0.304\ngini = 0.239\nsamples
= 36\nvalue = [31, 5]'),
 Text(0.16415868673050615, 0.6388888888888888, 'x[12] <= 0.056\ngini = 0.08\nsamples
= 24\nvalue = [23, 1]'),
 Text(0.15868673050615595, 0.5833333333333334, 'x[7] <= 0.207\ngini = 0.444\nsamples
= 3\nvalue = [2, 1]'),
 Text(0.15321477428180574, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.16415868673050615, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue = [2,
0]'),
 Text(0.16963064295485636, 0.5833333333333334, 'gini = 0.0\nsamples = 21\nvalue =
[21, 0]'),
 Text(0.18604651162790697, 0.6388888888888888, 'x[1] <= 0.482\ngini = 0.444\nsamples
= 12\nvalue = [8, 4]'),
 Text(0.18057455540355677, 0.5833333333333334, 'gini = 0.0\nsamples = 3\nvalue = [0,
3]'),
 Text(0.19151846785225718, 0.5833333333333334, 'x[5] <= 0.167\ngini = 0.198\nsamples
= 9\nvalue = [8, 1]'),
 Text(0.18604651162790697, 0.5277777777777778, 'x[0] <= 0.333\ngini = 0.5\nsamples =
2\nvalue = [1, 1]'),
 Text(0.18057455540355677, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.19151846785225718, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.19699042407660738, 0.5277777777777778, 'gini = 0.0\nsamples = 7\nvalue = [7,
0]'),
 Text(0.2024623803009576, 0.6944444444444444, 'x[5] <= 0.5\ngini = 0.48\nsamples =
5\nvalue = [2, 3]'),
 Text(0.19699042407660738, 0.6388888888888888, 'gini = 0.0\nsamples = 2\nvalue = [2,
0]'),
 Text(0.2079343365253078, 0.6388888888888888, 'gini = 0.0\nsamples = 3\nvalue = [0,
3]'),
 Text(0.3857729138166895, 0.8055555555555556, 'x[2] <= 0.167\ngini = 0.145\nsamples
= 751\nvalue = [692, 59]'),
 Text(0.2503419972640219, 0.75, 'x[4] <= 0.167\ngini = 0.237\nsamples = 153\nvalue =
[132, 21]'),
 Text(0.2243502051983584, 0.6944444444444444, 'x[9] <= 0.429\ngini = 0.5\nsamples =
10\nvalue = [5, 5]'),
 Text(0.2188782489740082, 0.6388888888888888, 'x[7] <= 0.247\ngini = 0.408\nsamples
= 7\nvalue = [2, 5]'),
 Text(0.213406292749658, 0.5833333333333334, 'x[12] <= 0.139\ngini = 0.444\nsamples
= 3\nvalue = [2, 1]'),
 Text(0.2079343365253078, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue = [2,
0]'),
 Text(0.2188782489740082, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.2243502051983584, 0.5833333333333334, 'gini = 0.0\nsamples = 4\nvalue = [0,
4]'),
 Text(0.22982216142270862, 0.6388888888888888, 'gini = 0.0\nsamples = 3\nvalue = [3,
0]'),
 Text(0.2763337893296854, 0.6944444444444444, 'x[5] <= 0.167\ngini = 0.199\nsamples
= 143\nvalue = [127, 16]'),
 Text(0.24623803009575923, 0.6388888888888888, 'x[7] <= 0.214\ngini = 0.375\nsamples
= 24\nvalue = [18, 6]'),
 Text(0.23529411764705882, 0.5833333333333334, 'x[0] <= 0.548\ngini = 0.494\nsamples
= 9\nvalue = [4, 5]'),
 Text(0.22982216142270862, 0.5277777777777778, 'x[4] <= 0.5\ngini = 0.444\nsamples =
6\nvalue = [4, 2]'),
 Text(0.2243502051983584, 0.4722222222222222, 'gini = 0.0\nsamples = 3\nvalue = [3,
0]'),
 Text(0.23529411764705882, 0.4722222222222222, 'x[4] <= 0.833\ngini = 0.444\nsamples
= 3\nvalue = [1, 2]'),
 Text(0.22982216142270862, 0.4166666666666667, 'gini = 0.0\nsamples = 2\nvalue = [0,
2]'),
 Text(0.24076607387140903, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.24076607387140903, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue = [0,
3]'),
 Text(0.25718194254445964, 0.5833333333333334, 'x[0] <= 0.393\ngini = 0.124\nsamples
= 15\nvalue = [14, 1]'),
 Text(0.25170998632010944, 0.5277777777777778, 'x[1] <= 0.268\ngini = 0.5\nsamples =
2\nvalue = [1, 1]'),
 Text(0.24623803009575923, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
```

```
 Text(0.25718194254445964, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.26265389876880985, 0.5277777777777778, 'gini = 0.0\nsamples = 13\nvalue =
[13, 0]'),
 Text(0.3064295485636115, 0.6388888888888888, 'x[7] <= 0.444\ngini = 0.154\nsamples
= 119\nvalue = [109, 10]'),
 Text(0.28454172366621067, 0.5833333333333334, 'x[0] <= 0.893\ngini = 0.07\nsamples
= 82\nvalue = [79, 3]'),
 Text(0.27359781121751026, 0.5277777777777778, 'x[7] <= 0.228\ngini = 0.05\nsamples
= 78\nvalue = [76, 2]'),
 Text(0.26812585499316005, 0.4722222222222222, 'gini = 0.0\nsamples = 48\nvalue =
[48, 0]'),
 Text(0.27906976744186046, 0.4722222222222222, 'x[7] <= 0.231\ngini = 0.124\nsamples
= 30\nvalue = [28, 2]'),
 Text(0.27359781121751026, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.28454172366621067, 0.4166666666666667, 'x[7] <= 0.248\ngini = 0.067\nsamples
= 29\nvalue = [28, 1]'),
 Text(0.27906976744186046, 0.3611111111111111, 'x[3] <= 0.5\ngini = 0.375\nsamples =
4\nvalue = [3, 1]'),
 Text(0.27359781121751026, 0.3055555555555556, 'gini = 0.0\nsamples = 3\nvalue = [3,
0]'),
 Text(0.28454172366621067, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.29001367989056087, 0.3611111111111111, 'gini = 0.0\nsamples = 25\nvalue =
[25, 0]'),
 Text(0.2954856361149111, 0.5277777777777778, 'x[1] <= 0.804\ngini = 0.375\nsamples
= 4\nvalue = [3, 1]'),
 Text(0.29001367989056087, 0.4722222222222222, 'gini = 0.0\nsamples = 3\nvalue = [3,
0]'),
 Text(0.3009575923392613, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.3283173734610123, 0.5833333333333334, 'x[7] <= 0.588\ngini = 0.307\nsamples
= 37\nvalue = [30, 7]'),
 Text(0.3173734610123119, 0.5277777777777778, 'x[4] <= 0.5\ngini = 0.5\nsamples =
12\nvalue = [6, 6]'),
 Text(0.3119015047879617, 0.4722222222222222, 'gini = 0.0\nsamples = 4\nvalue = [4,
0]'),
 Text(0.3228454172366621, 0.4722222222222222, 'x[9] <= 0.179\ngini = 0.375\nsamples
= 8\nvalue = [2, 6]'),
 Text(0.3173734610123119, 0.4166666666666667, 'gini = 0.0\nsamples = 5\nvalue = [0,
5]'),
 Text(0.3283173734610123, 0.4166666666666667, 'x[5] <= 0.5\ngini = 0.444\nsamples =
3\nvalue = [2, 1]'),
 Text(0.3228454172366621, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.3337893296853625, 0.3611111111111111, 'gini = 0.0\nsamples = 2\nvalue = [2,
0]'),
 Text(0.3392612859097127, 0.5277777777777778, 'x[13] <= 0.7\ngini = 0.077\nsamples =
25\nvalue = [24, 1]'),
 Text(0.3337893296853625, 0.4722222222222222, 'gini = 0.0\nsamples = 21\nvalue =
[21, 0]'),
 Text(0.3447332421340629, 0.4722222222222222, 'x[12] <= 0.417\ngini = 0.375\nsamples
= 4\nvalue = [3, 1]'),
 Text(0.3392612859097127, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.35020519835841313, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue = [3,
0]'),
 Text(0.521203830369357, 0.75, 'x[0] <= 0.345\ngini = 0.119\nsamples = 598\nvalue =
[560, 38]'),
 Text(0.4100547195622435, 0.6944444444444444, 'x[7] <= 0.067\ngini = 0.205\nsamples
= 198\nvalue = [175, 23]'),
 Text(0.4045827633378933, 0.6388888888888888, 'gini = 0.0\nsamples = 2\nvalue = [0,
2]'),
 Text(0.4155266757865937, 0.6388888888888888, 'x[9] <= 0.036\ngini = 0.191\nsamples
= 196\nvalue = [175, 21]'),
 Text(0.37756497948016415, 0.5833333333333334, 'x[4] <= 0.5\ngini = 0.384\nsamples =
27\nvalue = [20, 7]'),
 Text(0.36114911080711354, 0.5277777777777778, 'x[7] <= 0.199\ngini = 0.494\nsamples
= 9\nvalue = [4, 5]'),
 Text(0.35567715458276333, 0.4722222222222222, 'gini = 0.0\nsamples = 3\nvalue = [0,
3]'),
 Text(0.36662106703146374, 0.4722222222222222, 'x[7] <= 0.476\ngini = 0.444\nsamples
= 6\nvalue = [4, 2]'),
 Text(0.36114911080711354, 0.4166666666666667, 'gini = 0.0\nsamples = 4\nvalue = [4,
0]'),
 Text(0.37209302325581395, 0.4166666666666667, 'gini = 0.0\nsamples = 2\nvalue = [0,
2]'),
 Text(0.39398084815321477, 0.5277777777777778, 'x[1] <= 0.071\ngini = 0.198\nsamples
= 18\nvalue = [16, 2]'),
 Text(0.38850889192886456, 0.4722222222222222, 'x[0] <= 0.274\ngini = 0.444\nsamples
= 6\nvalue = [4, 2]'),
 Text(0.38303693570451436, 0.4166666666666667, 'x[6] <= 0.25\ngini = 0.32\nsamples =
5\nvalue = [4, 1]'),
 Text(0.37756497948016415, 0.3611111111111111, 'x[5] <= 0.167\ngini = 0.5\nsamples =
```

```
2\nvalue = [1, 1]'),
 Text(0.37209302325581395, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.38303693570451436, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.38850889192886456, 0.3611111111111111, 'gini = 0.0\nsamples = 3\nvalue = [3,
0]'),
 Text(0.39398084815321477, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.399452804377565, 0.4722222222222222, 'gini = 0.0\nsamples = 12\nvalue = [12,
0]'),
 Text(0.45348837209302323, 0.5833333333333334, 'x[10] <= 0.088\ngini =
0.152\nsamples = 169\nvalue = [155, 14]'),
 Text(0.4158686730506156, 0.5277777777777778, 'x[9] <= 0.179\ngini = 0.42\nsamples =
10\nvalue = [7, 3]'),
 Text(0.4103967168262654, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue = [0,
2]'),
 Text(0.4213406292749658, 0.4722222222222222, 'x[7] <= 0.119\ngini = 0.219\nsamples
= 8\nvalue = [7, 1]'),
 Text(0.4158686730506156, 0.4166666666666667, 'gini = 0.0\nsamples = 7\nvalue = [7,
0]'),
 Text(0.426812585499316, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.4911080711354309, 0.5277777777777778, 'x[11] <= 0.5\ngini = 0.129\nsamples =
159\nvalue = [148, 11]'),
 Text(0.45690834473324216, 0.4722222222222222, 'x[1] <= 0.554\ngini = 0.26\nsamples
= 39\nvalue = [33, 6]'),
 Text(0.4377564979480164, 0.4166666666666667, 'x[6] <= 0.75\ngini = 0.133\nsamples =
28\nvalue = [26, 2]'),
 Text(0.4322845417236662, 0.3611111111111111, 'gini = 0.0\nsamples = 20\nvalue =
[20, 0]'),
 Text(0.4432284541723666, 0.3611111111111111, 'x[9] <= 0.321\ngini = 0.375\nsamples
= 8\nvalue = [6, 2]'),
 Text(0.4377564979480164, 0.3055555555555556, 'gini = 0.0\nsamples = 5\nvalue = [5,
0]'),
 Text(0.4487004103967168, 0.3055555555555556, 'x[0] <= 0.25\ngini = 0.444\nsamples =
3\nvalue = [1, 2]'),
 Text(0.4432284541723666, 0.25, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.454172366621067, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.47606019151846785, 0.4166666666666667, 'x[7] <= 0.349\ngini = 0.463\nsamples
= 11\nvalue = [7, 4]'),
 Text(0.46511627906976744, 0.3611111111111111, 'x[1] <= 0.696\ngini = 0.278\nsamples
= 6\nvalue = [5, 1]'),
 Text(0.45964432284541723, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.47058823529411764, 0.3055555555555556, 'gini = 0.0\nsamples = 5\nvalue = [5,
0]'),
 Text(0.48700410396716826, 0.3611111111111111, 'x[1] <= 0.75\ngini = 0.48\nsamples =
5\nvalue = [2, 3]'),
 Text(0.48153214774281805, 0.3055555555555556, 'gini = 0.0\nsamples = 2\nvalue = [2,
0]'),
 Text(0.49247606019151846, 0.3055555555555556, 'gini = 0.0\nsamples = 3\nvalue = [0,
3]'),
 Text(0.5253077975376197, 0.4722222222222222, 'x[12] <= 0.25\ngini = 0.08\nsamples =
120\nvalue = [115, 5]'),
 Text(0.5198358413132695, 0.4166666666666667, 'x[12] <= 0.194\ngini = 0.12\nsamples
= 78\nvalue = [73, 5]'),
 Text(0.5088919288645691, 0.3611111111111111, 'x[0] <= 0.25\ngini = 0.083\nsamples =
69\nvalue = [66, 3]'),
 Text(0.5034199726402189, 0.3055555555555556, 'gini = 0.0\nsamples = 30\nvalue =
[30, 0]'),
 Text(0.5143638850889193, 0.3055555555555556, 'x[10] <= 0.228\ngini = 0.142\nsamples
= 39\nvalue = [36, 3]'),
 Text(0.5088919288645691, 0.25, 'x[7] <= 0.236\ngini = 0.278\nsamples = 18\nvalue =
[15, 3]'),
 Text(0.5034199726402189, 0.19444444444444445, 'x[9] <= 0.321\ngini = 0.208\nsamples
= 17\nvalue = [15, 2]'),
 Text(0.49794801641586867, 0.1388888888888889, 'gini = 0.0\nsamples = 11\nvalue =
[11, 0]'),
 Text(0.5088919288645691, 0.1388888888888889, 'x[5] <= 0.667\ngini = 0.444\nsamples
= 6\nvalue = [4, 2]'),
 Text(0.5034199726402189, 0.08333333333333333, 'x[12] <= 0.056\ngini =
0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.49794801641586867, 0.027777777777777776, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.5088919288645691, 0.027777777777777776, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.5143638850889193, 0.08333333333333333, 'gini = 0.0\nsamples = 3\nvalue = [3,
0]'),
 Text(0.5143638850889193, 0.19444444444444445, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.5198358413132695, 0.25, 'gini = 0.0\nsamples = 21\nvalue = [21, 0]'),
 Text(0.53077975376197, 0.3611111111111111, 'x[0] <= 0.214\ngini = 0.346\nsamples =
9\nvalue = [7, 2]'),
```

```
 Text(0.5253077975376197, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.5362517099863201, 0.3055555555555556, 'x[1] <= 0.196\ngini = 0.219\nsamples
= 8\nvalue = [7, 1]'),
 Text(0.53077975376197, 0.25, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(0.5417236662106704, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.53077975376197, 0.4166666666666667, 'gini = 0.0\nsamples = 42\nvalue = [42,
0]'),
 Text(0.6323529411764706, 0.6944444444444444, 'x[13] <= 0.967\ngini = 0.072\nsamples
= 400\nvalue = [385, 15]'),
 Text(0.6025991792065664, 0.6388888888888888, 'x[9] <= 0.321\ngini = 0.064\nsamples
= 394\nvalue = [381, 13]'),
 Text(0.5595075239398085, 0.5833333333333334, 'x[7] <= 0.069\ngini = 0.031\nsamples
= 253\nvalue = [249, 4]'),
 Text(0.5417236662106704, 0.5277777777777778, 'x[7] <= 0.068\ngini = 0.375\nsamples
= 4\nvalue = [3, 1]'),
 Text(0.5362517099863201, 0.4722222222222222, 'gini = 0.0\nsamples = 3\nvalue = [3,
0]'),
 Text(0.5471956224350205, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.5772913816689467, 0.5277777777777778, 'x[5] <= 0.167\ngini = 0.024\nsamples
= 249\nvalue = [246, 3]'),
 Text(0.5581395348837209, 0.4722222222222222, 'x[7] <= 0.128\ngini = 0.089\nsamples
= 43\nvalue = [41, 2]'),
 Text(0.5471956224350205, 0.4166666666666667, 'x[7] <= 0.112\ngini = 0.444\nsamples
= 3\nvalue = [2, 1]'),
 Text(0.5417236662106704, 0.3611111111111111, 'gini = 0.0\nsamples = 2\nvalue = [2,
0]'),
 Text(0.5526675786593708, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.5690834473324213, 0.4166666666666667, 'x[1] <= 0.018\ngini = 0.049\nsamples
= 40\nvalue = [39, 1]'),
 Text(0.5636114911080712, 0.3611111111111111, 'x[10] <= 0.719\ngini = 0.278\nsamples
= 6\nvalue = [5, 1]'),
 Text(0.5581395348837209, 0.3055555555555556, 'gini = 0.0\nsamples = 5\nvalue = [5,
0]'),
 Text(0.5690834473324213, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.5745554035567716, 0.3611111111111111, 'gini = 0.0\nsamples = 34\nvalue =
[34, 0]'),
 Text(0.5964432284541724, 0.4722222222222222, 'x[9] <= 0.036\ngini = 0.01\nsamples =
206\nvalue = [205, 1]'),
 Text(0.5909712722298222, 0.4166666666666667, 'x[10] <= 0.263\ngini = 0.043\nsamples
= 46\nvalue = [45, 1]'),
 Text(0.585499316005472, 0.3611111111111111, 'x[5] <= 0.5\ngini = 0.198\nsamples =
9\nvalue = [8, 1]'),
 Text(0.5800273597811217, 0.3055555555555556, 'x[7] <= 0.173\ngini = 0.5\nsamples =
2\nvalue = [1, 1]'),
 Text(0.5745554035567716, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.585499316005472, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5909712722298222, 0.3055555555555556, 'gini = 0.0\nsamples = 7\nvalue = [7,
0]'),
```

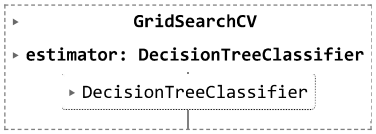
```
from sklearn.model_selection import GridSearchCV
parameter={
    'criterion':['gini','entropy'],
    'splitter':['best','random'],
    'max_depth':[1,2,3,4,5],
    'max_features':['auto', 'sqrt', 'log2']
    }
```

```
 = 103\nvalue = [94, 9]'),
```

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

```
 Text(0.612859097127223, 0.4166666666666667, 'x[0] <= 0.393\ngini = 0.444\nsamples =
```

```
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')
```




```
 Text(0.6347469220246238, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [0,
```

```
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 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
  warnings.warn(
```

```
dtc_cv=DecisionTreeClassifier(criterion= 'entropy',
 max_depth=5,
 max_features='auto',
 splitter='random')
dtc_cv.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 1.1 an
  warnings.warn(
```

```
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=5, max_features='auto',
                       splitter='random')
```

```
  warnings.warn(
```

```
pred1 = dtc_cv.predict(x_test)
pred1
```

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

```
probability=dtc.predict_proba(x_test)[:,1] # probability of all the yes values
probability
```

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

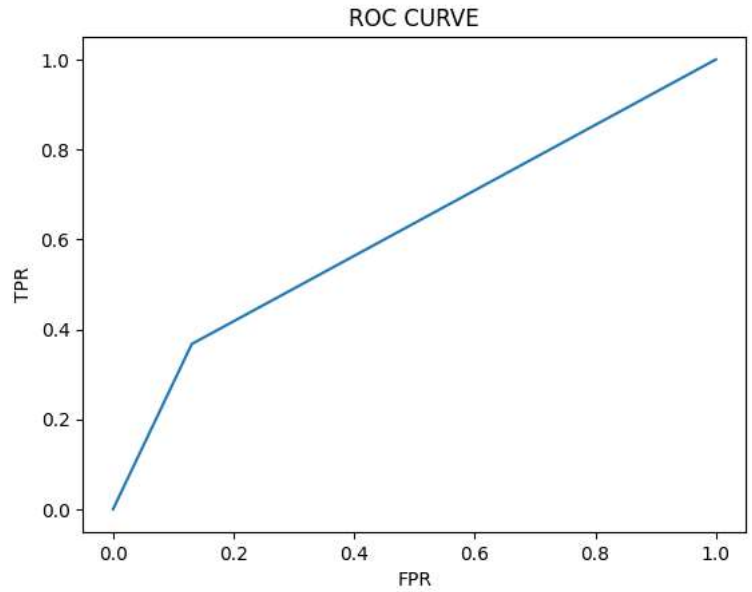
```
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in 1.1 ar
```

```
print(classification_report(y_test, pred1))
```

```
              precision    recall  f1-score   support

           0       0.86      0.98      0.92       245
           1       0.67      0.20      0.31        49

    accuracy                           0.85       294
   macro avg       0.76      0.59      0.61       294
weighted avg       0.83      0.85      0.82       294
```

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()
```

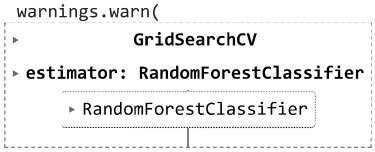


Model_Building --- Random Forest Classifier

```
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, 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(
sklearn.utils._param_validation.InvalidParameterError: The 'max_features' parameter of RandomForestClassifier must be an int in the rang

  warnings.warn(some_fits_failed_message, FitFailedWarning)
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_search.py:952: UserWarning: One or more of the test scores are non-fini
 0.86136462 0.86392148 0.86051717 0.8622483  0.85287556 0.85370129
 0.85287556 0.85882949        nan 0.85543242 0.85628712 0.85794582
 0.85624366 0.85538896 0.85882225 0.86220484 0.85710561 0.86137911
 0.86053165 0.85627264 0.8656164  0.86222657        nan 0.85033319
 0.85795306 0.85794582 0.85880776 0.86050992 0.85627264 0.85626539
 0.86476894 0.86137187 0.86393597 0.86050268 0.85883674 0.85374475
        nan 0.85373751 0.85966971 0.85880776 0.86049544 0.85882225
 0.86137187 0.85625815 0.86221208 0.86223381 0.85880776 0.8520281
 0.85541793 0.86308127        nan 0.85373751 0.85882225 0.86051717
 0.8621976  0.85966247 0.85882949 0.85371578 0.85796755 0.85798204
 0.85882949 0.85712009 0.85884398 0.85880776]
  warnings.warn(
```




```
pred3 = rfc_cv.predict(x_test)


print(classification_report(y_test, pred3))
```

```
                  precision    recall  f1-score   support

               0       0.87      0.96      0.91       245
               1       0.54      0.27      0.36        49

        accuracy                           0.84       294
       macro avg       0.70      0.61      0.63       294
    weighted avg       0.81      0.84      0.82       294
```

```
probability=rfc_cv.predict_proba(x_test)[:,1] # probability of all the yes values
probability
```

```
array([1.44844457e-02, 7.18997774e-02, 1.63456920e-01, 1.28918317e-01,
       8.40000000e-01, 1.23878736e-01, 6.20384615e-01, 3.18984916e-02,
       2.63016545e-03, 1.13679194e-01, 1.99574428e-03, 1.62936581e-01,
       9.86635353e-02, 7.03888889e-01, 2.32222222e-01, 3.32416721e-02,
       9.70126549e-02, 4.67050000e-01, 1.36930551e-02, 1.41492588e-01,
       6.65663781e-01, 1.60486101e-02, 7.16864623e-02, 2.66358004e-02,
       3.11074830e-01, 7.38142402e-02, 5.61164804e-02, 3.15762543e-02,
       6.94477124e-01, 4.57037414e-02, 6.37677132e-02, 2.25724863e-02,
       2.63760358e-01, 1.24089365e-01, 9.56206544e-03, 7.58315547e-02,
       6.86789995e-02, 1.24320652e-01, 1.24154089e-01, 7.01970692e-02,
       7.51464221e-02, 2.02172371e-02, 1.33426007e-01, 1.09248210e-01,
       4.95588777e-02, 5.82911111e-01, 1.48914581e-01, 7.74037177e-02,
       7.30000000e-01, 5.40588235e-01, 3.13474026e-01, 6.16821446e-01,
       1.24582454e-01, 3.01066611e-02, 4.73750000e-01, 1.24002788e-01,
       6.43441018e-03, 2.43915285e-02, 7.20310794e-02, 1.07224391e-01,
       5.00692848e-04, 1.74999172e-01, 1.23677449e-01, 1.52577317e-01,
```

```
4.01685610e-01, 7.30314147e-02, 2.00111111e-01, 1.60928686e-01,
1.03542702e-01, 9.09038900e-02, 2.06187491e-01, 6.95000000e-01,
8.01908978e-02, 6.43922113e-03, 2.03245393e-02, 5.59955845e-02,
8.13606929e-02, 1.90508046e-01, 1.18808662e-01, 3.30913689e-03,
3.72477777e-03, 6.39074762e-03, 8.68536899e-02, 1.64147662e-01,
4.28307597e-02, 9.52109758e-02, 5.02057353e-02, 4.86000000e-01,
6.52498355e-02, 2.03708961e-01, 4.92222222e-01, 2.08576354e-03,
1.09746524e-01, 1.79154457e-01, 1.01603388e-01, 2.94050254e-02,
9.22986003e-02, 1.73766759e-01, 6.45540791e-02, 2.92723386e-02,
2.63758307e-01, 3.67567254e-01, 4.80000000e-01, 1.02498772e-02,
2.46758317e-02, 6.18550337e-02, 1.62275488e-02, 5.37935401e-01,
1.54466458e-01, 1.26318587e-01, 2.32792182e-01, 4.71307113e-01,
5.62904338e-02, 5.30755828e-02, 9.54022754e-02, 3.52410686e-02,
1.51221755e-01, 1.19914927e-01, 1.29421728e-02, 8.83275382e-03,
3.71417665e-02, 1.04258879e-01, 1.87029472e-02, 8.92222222e-01,
1.05323030e-01, 1.58076205e-01, 3.75798615e-02, 1.12875845e-01,
5.15371606e-02, 2.17453780e-02, 1.84680314e-01, 4.10270270e-01,
2.54463065e-01, 3.60092427e-01, 2.60420638e-01, 2.70000000e-01,
3.14768803e-01, 5.71115936e-02, 1.61895110e-01, 6.21797486e-02,
9.93960761e-02, 1.30133333e-01, 1.53503224e-01, 5.24528242e-02,
4.56666667e-01, 3.19640356e-02, 1.00270270e-01, 1.32919365e-01,
1.53739935e-02, 1.49171114e-01, 1.29164571e-01, 2.82795788e-01,
1.03522265e-01, 2.70507113e-01, 1.37421215e-01, 6.57593026e-02,
2.72411380e-01, 1.09921230e-02, 1.02431060e-01, 4.72988296e-02,
4.40896154e-01, 3.44616473e-03, 3.33557717e-01, 6.30000000e-01,
1.93333333e-01, 1.66289545e-01, 7.15969326e-02, 9.51442013e-02,
1.74575472e-02, 6.11530259e-02, 2.64650948e-03, 3.89281837e-02,
1.21267745e-01, 4.63602731e-02, 2.64386612e-01, 5.77369697e-01,
2.39916573e-02, 4.80998916e-01, 6.65107862e-02, 8.08120476e-02,
3.71555365e-02, 2.57822808e-02, 6.07937722e-02, 5.68656417e-01,
4.98246304e-02, 7.87796770e-03, 2.76901742e-01, 2.07702066e-01,
3.99414532e-01, 1.14980227e-01, 5.67967427e-01, 9.00000000e-01,
2.51491803e-01, 4.69629287e-03, 1.25038336e-01, 4.20245874e-02,
1.56563046e-02, 2.45885407e-02, 4.25909091e-01, 1.27805374e-01,
1.70535847e-01, 1.12292449e-01, 3.50270112e-01, 4.35831058e-02,
9.43722741e-02, 9.09906510e-02, 5.55357143e-01, 3.20782743e-02,
8.24113071e-02, 9.72946829e-03, 1.41188883e-01, 8.97116215e-02,
4.06638808e-02, 9.76430134e-02, 3.74546339e-02, 7.52108967e-02,
1.20393606e-02, 1.94894141e-01, 4.33815006e-02, 6.20000000e-01,
1.16536183e-01, 2.03037097e-01, 7.30666667e-01, 6.10442260e-02,
1.56135470e-01, 7.63398245e-02, 2.28315651e-02, 2.62212681e-01,
4.30270270e-01, 2.11434646e-02, 1.12828269e-02, 1.96858776e-01
```

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()
```