Assignment-4

Name-Rishima Chowdhury

Reg no-21BCE1097

Email- rishima.chowdhury2021@vitstudent.ac.in

Campus-VIT Chennai

Slot- Morning

Data Preprocessing

1. import the necessary libraries

2. import the dataset

3. Handling null values

4. Data visualization

5. outlier detection

6. Seperate Dependent and Independent variables.

7. Encoding

8. Splitting into training and testing set

9. Perform feature scaling

1. Import the necessary libraries.

In [155…
```python
import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
```
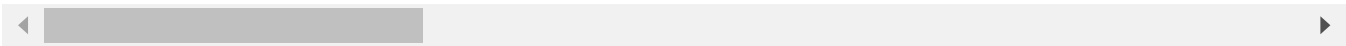
2. import the dataset

In [156…
```python
dataset=pd.read_csv("Employee-Attrition.csv")
dataset.head()
```

Out[156]:

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

5 rows × 35 columns

In [157… `dataset.tail()`

Out[157]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Ed
1465	36	No	Travel_Frequently	884	Research & Development	23	2	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	
1468	49	No	Travel_Frequently	1023	Sales	2	3	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	

5 rows × 35 columns
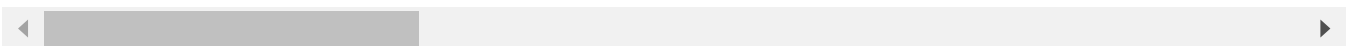
In [158… `dataset.shape`

Out[158]: `(1470, 35)`

In [159… `dataset.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
```

Each row in the dataset represents an employee of a fictional company ABC, and contains the following features:

Age: Age of employees

Department: Department of work

Distance from home

Education: 1-Below College; 2-College; 3-Bachelor; 4-Master; 5-Doctor;

Education Field

Environment Satisfaction: 1-Low; 2-Medium; 3-High; 4-Very High

Job Satisfaction: 1-Low; 2-Medium; 3-High; 4-Very High

Marital Status

Monthly Income

Num Companies Worked: Number of companies worked prior to ABC

Work Life Balance: 1-Bad; 2-Good; 3-Better; 4-Best

Years At Company: Current years of service

Attrition: Employee attrition status

In [160…   `dataset.describe()`

Out[160]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNum
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.0000
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865:
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024:
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000(
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250(
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500(
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750(
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000(

8 rows × 26 columns
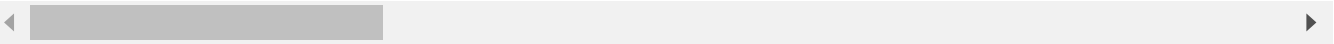
In [161…   `dataset.corr()`

```
<ipython-input-161-c187c74d1e71>:1: FutureWarning: The default value of numeric_on
ly in DataFrame.corr is deprecated. In a future version, it will default to False.
Select only valid columns or specify the value of numeric_only to silence this war
ning.
  dataset.corr()
```

Out[161]:

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

26 rows × 26 columns

3. Checking for Null Values

In [162…

```python
dataset.isnull().any()
```

```
Out[162]:    Age                            False
             Attrition                      False
             BusinessTravel                 False
             DailyRate                      False
             Department                     False
             DistanceFromHome               False
             Education                      False
             EducationField                 False
             EmployeeCount                  False
             EmployeeNumber                 False
             EnvironmentSatisfaction        False
             Gender                         False
             HourlyRate                     False
             JobInvolvement                 False
             JobLevel                       False
             JobRole                        False
             JobSatisfaction                False
             MaritalStatus                  False
             MonthlyIncome                  False
             MonthlyRate                    False
             NumCompaniesWorked             False
             Over18                         False
             OverTime                       False
             PercentSalaryHike              False
             PerformanceRating              False
             RelationshipSatisfaction       False
             StandardHours                  False
             StockOptionLevel               False
             TotalWorkingYears              False
             TrainingTimesLastYear          False
             WorkLifeBalance                False
             YearsAtCompany                 False
             YearsInCurrentRole             False
             YearsSinceLastPromotion        False
             YearsWithCurrManager           False
             dtype: bool
```

In [163…
```python
dataset.isnull().sum()
```

```
Out[163]:   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
```

4. Data Visualization

Visualizing Distribution of features

```python
In [164…   plt.figure(figsize = (15, 7))
           plt.style.use('seaborn-white')
           plt.subplot(331)
           sns.distplot(dataset['Age'])
           fig = plt.gcf()
           fig.set_size_inches(10,10)

           plt.subplot(332)
           sns.distplot(dataset['DistanceFromHome'])
           fig = plt.gcf()
           fig.set_size_inches(10,10)

           plt.subplot(333)
           sns.distplot(dataset['DailyRate'])
           fig = plt.gcf()
           fig.set_size_inches(10,10)

           plt.subplot(334)
           sns.distplot(dataset['YearsAtCompany'])
           fig = plt.gcf()
           fig.set_size_inches(10,10)
```

```python
plt.subplot(335)
sns.distplot(dataset['TotalWorkingYears'])
fig = plt.gcf()
fig.set_size_inches(10,10)

plt.subplot(336)
sns.distplot(dataset['NumCompaniesWorked'])
fig = plt.gcf()
fig.set_size_inches(10,10)

plt.subplot(337)
sns.distplot(dataset['MonthlyRate'])
fig = plt.gcf()
fig.set_size_inches(10,10)

plt.subplot(338)
sns.distplot(dataset['MonthlyIncome'])
fig = plt.gcf()
fig.set_size_inches(10,10)

plt.subplot(339)
sns.distplot(dataset['PercentSalaryHike'])
fig = plt.gcf()
fig.set_size_inches(10,10)
```

```
<ipython-input-164-2e0bc05fbe5a>:2: MatplotlibDeprecationWarning: The seaborn styl
es shipped by Matplotlib are deprecated since 3.6, as they no longer correspond to
the styles shipped by seaborn. However, they will remain available as 'seaborn-v0_
8-<style>'. Alternatively, directly use the seaborn API instead.
  plt.style.use('seaborn-white')
<ipython-input-164-2e0bc05fbe5a>:4: 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(dataset['Age'])
<ipython-input-164-2e0bc05fbe5a>:9: 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(dataset['DistanceFromHome'])
<ipython-input-164-2e0bc05fbe5a>:14: 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(dataset['DailyRate'])
<ipython-input-164-2e0bc05fbe5a>:19: 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(dataset['YearsAtCompany'])
<ipython-input-164-2e0bc05fbe5a>:24: 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(dataset['TotalWorkingYears'])
<ipython-input-164-2e0bc05fbe5a>:29: 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(dataset['NumCompaniesWorked'])
```
<ipython-input-164-2e0bc05fbe5a>:34: 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(dataset['MonthlyRate'])
```
<ipython-input-164-2e0bc05fbe5a>:39: 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(dataset['MonthlyIncome'])
```
<ipython-input-164-2e0bc05fbe5a>:44: 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(dataset['PercentSalaryHike'])
```
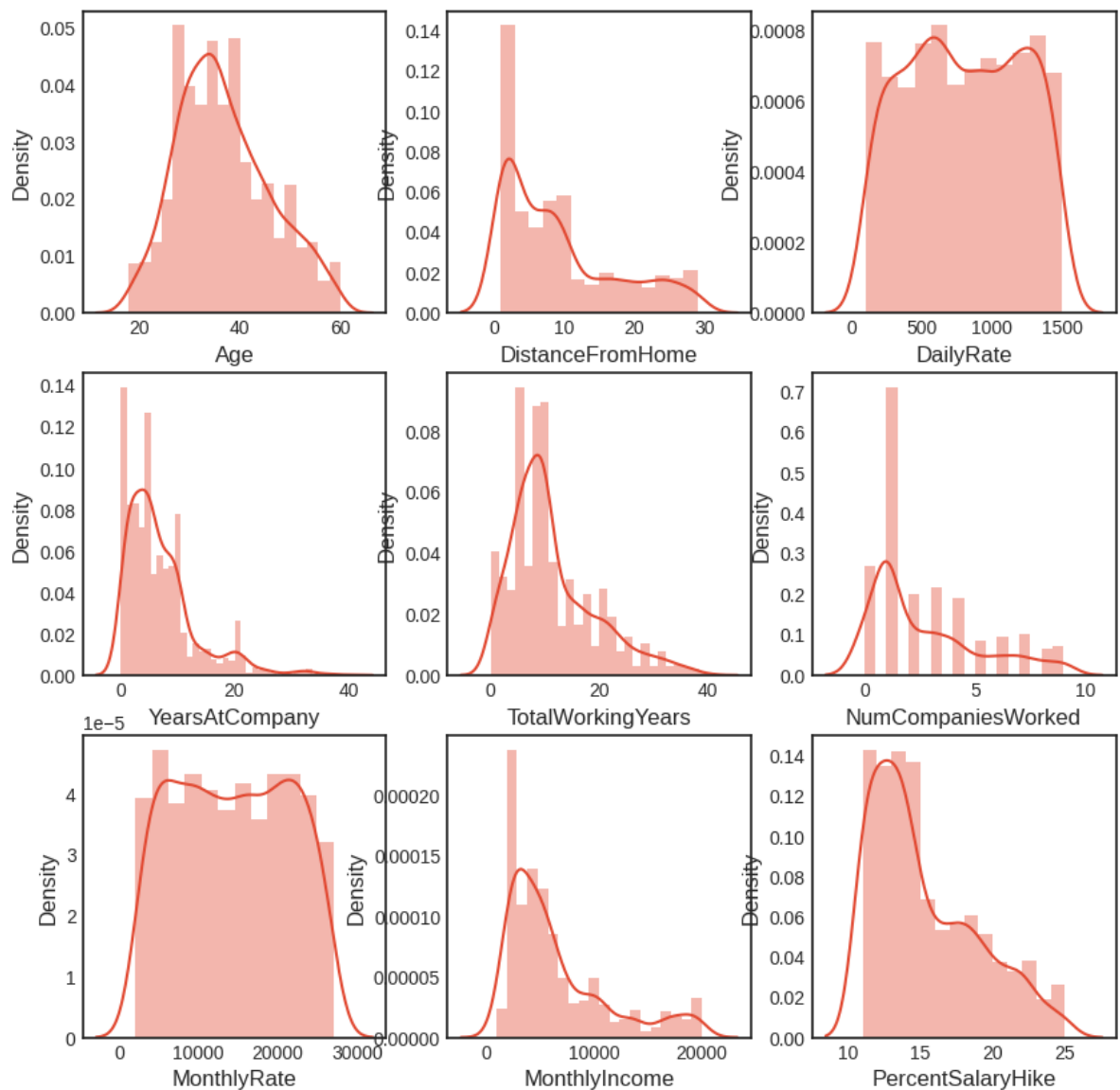
Inference:

Monthly and Daily Rates has proportional normal distribution

Most of the features except age parameter are right skewed

Attrition frecuency plotting

```
colors = ["#C0EDD2","#FF0018"]
df_class = dataset["Attrition"].value_counts().reset_index()
plt.figure(figsize=(6,3))
ax = sns.barplot(x="index",y="Attrition",data=df_class,palette=colors)
sns.despine()
ax.text(-0.5,1450,"Attrition frecuency",
        fontsize=15,
         fontweight='bold')
for num,text in enumerate(df_class["Attrition"]):
        ax.text(num,text-100,text)
plt.xlabel("Attrition")
plt.ylabel("Frecuency")
plt.show(block=False)
```

Attrition frecuency
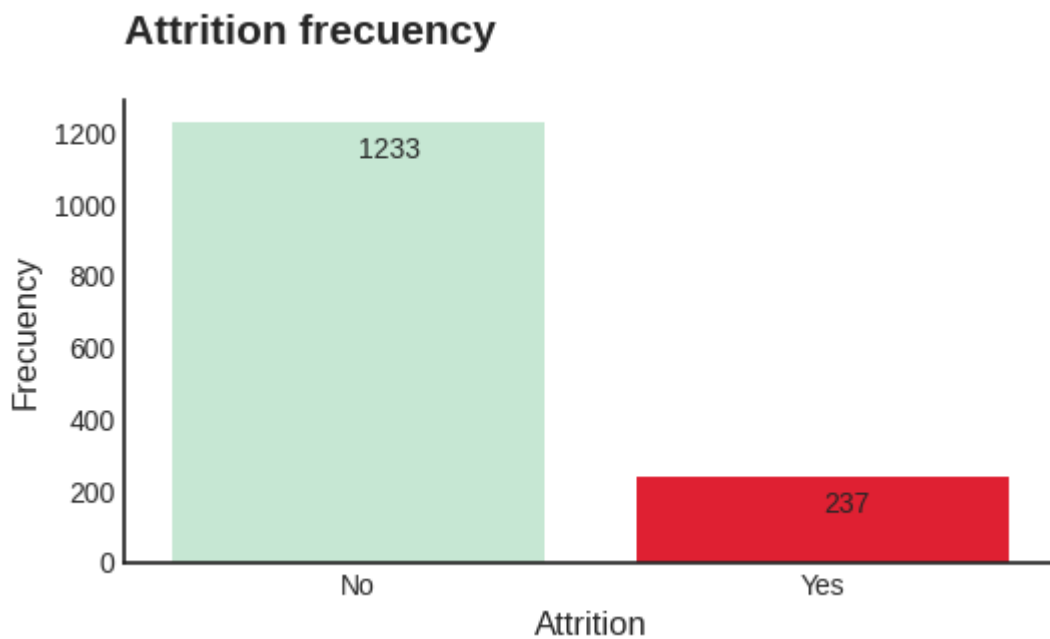
Inference

Attriction frecuency is plotted, which clearly shows no's frquency more than yes'.

Numeric and Categoric data heatmap plotting

```
In [166...   # numeric and categoric
            num_col = dataset.describe().columns.tolist()
            cat_col = dataset.describe(include="object").columns.tolist()
            corr = dataset[num_col].corr()
            plt.subplots(figsize=(20,15))
            ax = sns.heatmap(
                corr,annot=True)
            ax.text(0,-2.2,"HeatMap",
                    fontsize=15,
                     fontweight='bold')
            plt.show(block=False)
```

HeatMap
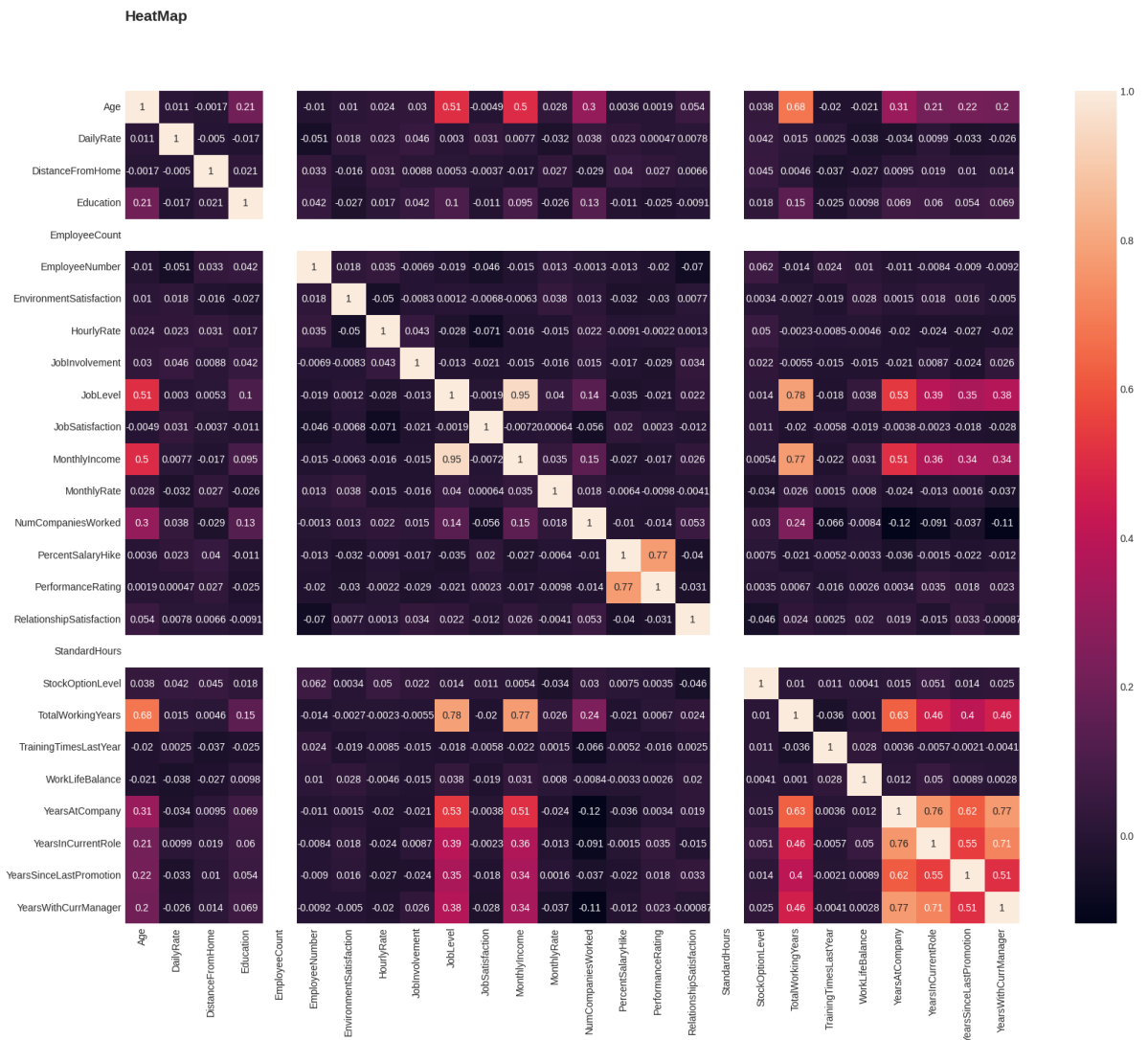
Inference- The band which has lighter colour has high correlation

Boxes in lower right corner have high correlation,but those are relationship between their working period which obviously correlates and has nothing to aid for our objective, so we neglect that

Job level and Monthly income has perfect positive correlation which means higher the job level, better the income

Also Job level has high correlation with Working years, when the employyee's year of working increases he/she is being promoted based on seniority

There is also a high correlation between Monthly Income and Total working years which depicts that employees earn higher income owing to their seniority with the firm

Age Distribution-Histogram

```
In [167…   plt.figure(figsize=(20,5))
           plt.hist(dataset.Age,bins=20)
           plt.xlabel("Age")
           plt.ylabel("Counts")
           plt.title("Age Counts")
           plt.show()
```
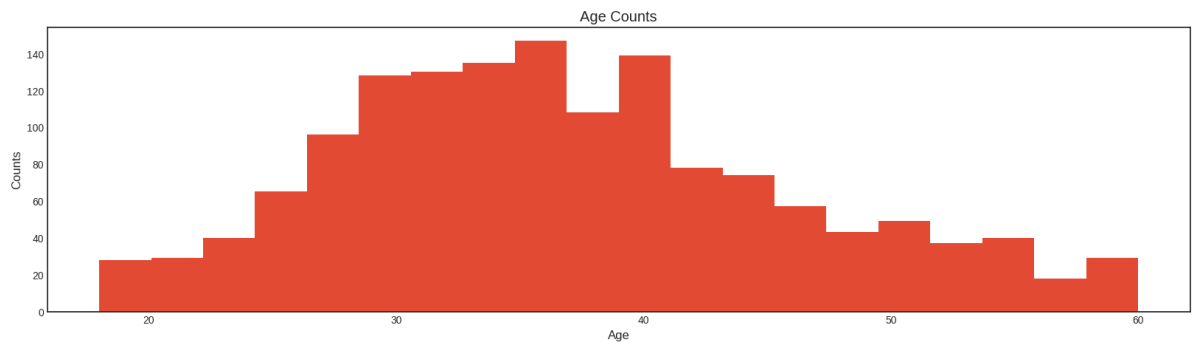
Inference

From the age distribution histogram, we can conclude that employees aged between 30-40 are working in IBM

Department Distribution-Pie Chart

In [168…
```python
labels = ['R&D', 'Sales', 'HR']
sizes = dataset['Department'].value_counts()
colors = plt.cm.copper(np.linspace(0, 1, 5))
explode = [0.1, 0.1, 0.2]

plt.rcParams['figure.figsize'] = (9, 9)
plt.pie(sizes, labels = labels, colors = colors, explode = explode, shadow = True)
plt.title('Department Distribution', fontsize = 20)
plt.legend()
plt.show()
```

Department Distribution
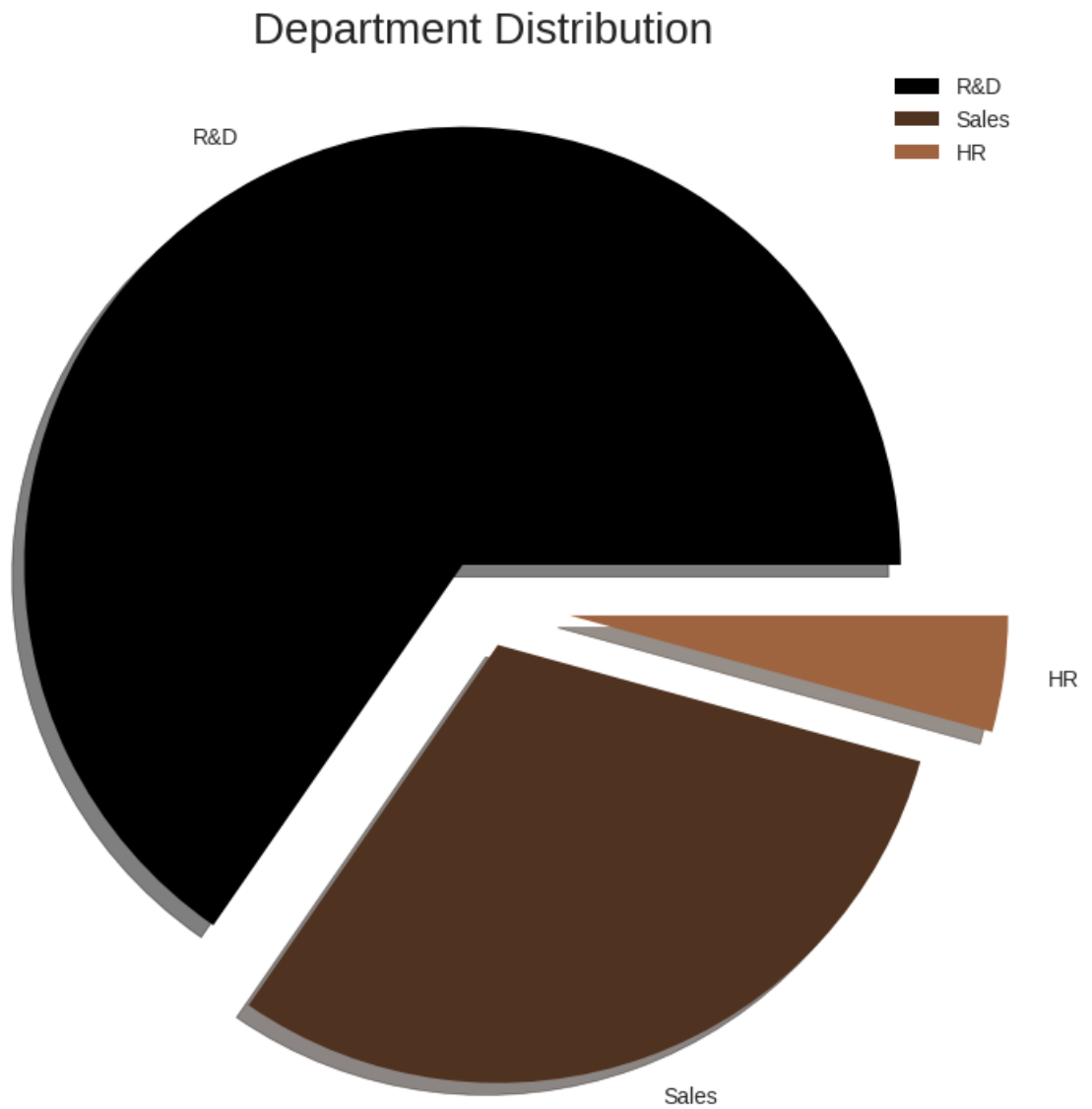
Inference:

The given dataset has majority of employees from Research and Development followed by Sales and HR department

Distribution of Salary across Job Roles-Boxplot

```
In [169…
plt.rcParams['figure.figsize'] = (16, 8)
ax = sns.boxplot(x = dataset['JobRole'], y =dataset['MonthlyIncome'], data = dataset
ax.set_xlabel(xlabel = 'Names of Job Roles', fontsize = 20)
ax.set_ylabel(ylabel = 'Monthly Income', fontsize = 20)
ax.set_title(label = 'Distribution of Salary across Job Roles', fontsize = 30)
plt.xticks(rotation = 90)
plt.show()
```

Distribution of Salary across Job Roles
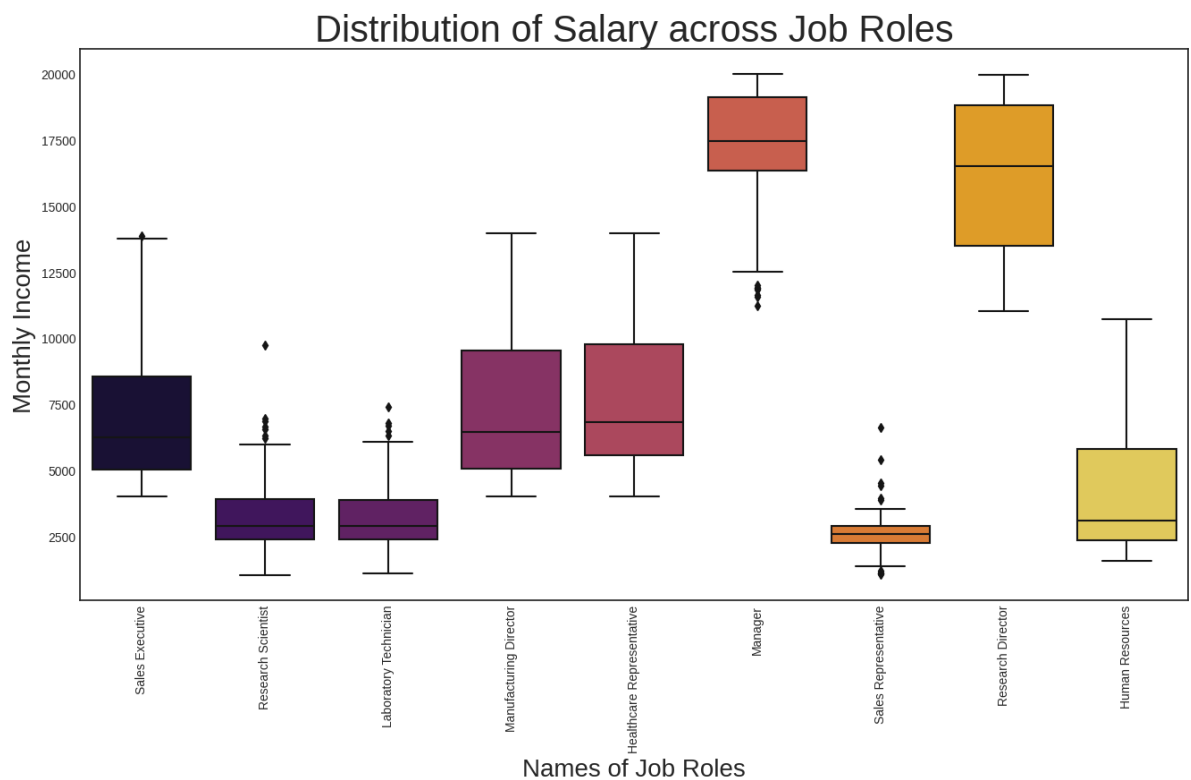
Inference

Highest average monthly income(17500) is given to Managers

Lowest average monthly income(2500) is given to Sales Representative, But there are many outliers in Sales Representative income

Research Scientist, Laboratory Technician and HR gets almost same average income

Relationship with working years and income- Joint & lm Plot

```
In [170…    plt.figure(figsize=(10,5))
            plt.style.use('ggplot')
            sns.jointplot(x='TotalWorkingYears', y='MonthlyIncome', data=dataset)
```

Out[170]:   `<seaborn.axisgrid.JointGrid at 0x7a874f3eada0>`

             `<Figure size 1000x500 with 0 Axes>`

```
In [171...  sns.lmplot(x = 'TotalWorkingYears', y = 'MonthlyIncome', data = dataset, col = 'Att
            plt.show()
```
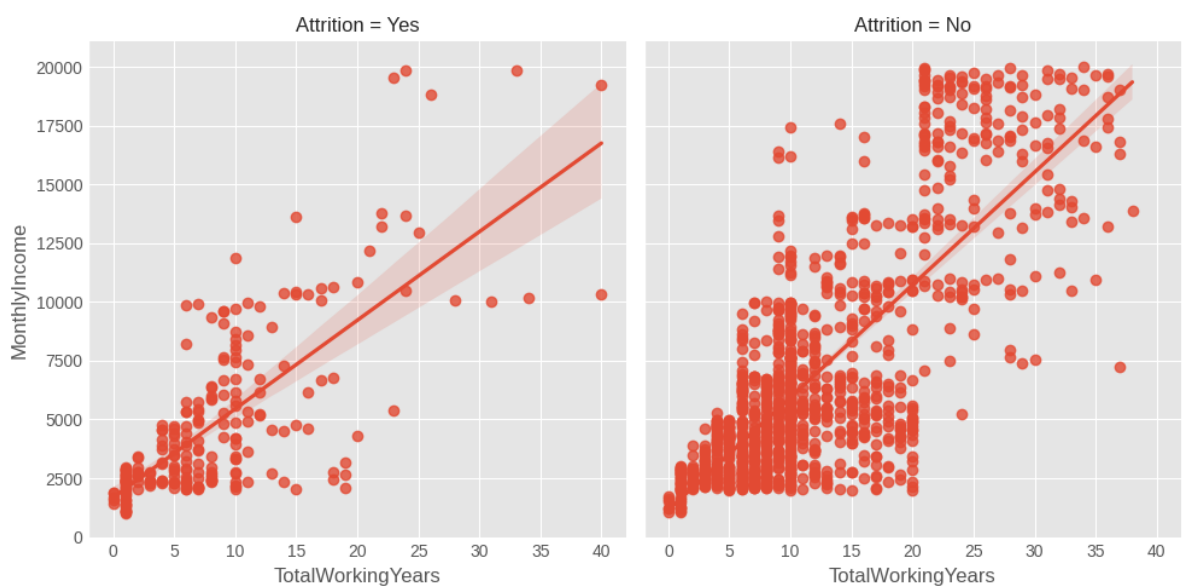

Inference

Employees who had less working years were receiving comparitively lower salary

There are many fresher employees working in IBM

Most of the employees who got laid off had less working years(fresher and junior level) and they were receiving less salary

Very few people from senior category and receiving higher income got laid off

Gender vs Monthly Income-Violin Plot

```
In [172…   fig,ax = plt.subplots(figsize=(10,7))
           sns.violinplot(x='Gender', y='MonthlyIncome',hue='Attrition',split=True,data=datase
```

Out[172]:   <Axes: xlabel='Gender', ylabel='MonthlyIncome'>
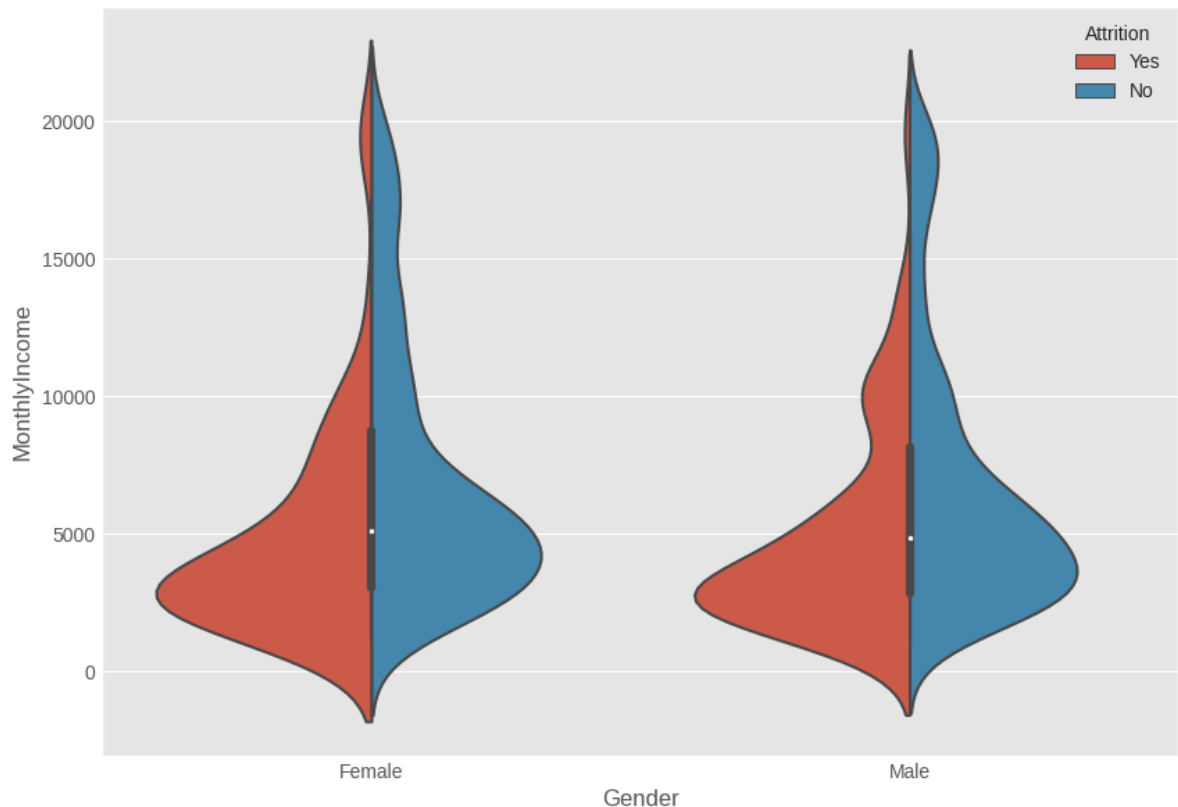
Inference

Both the genders weren't discriminated in income and the attrition is at the same rate in both the cases

Relationship with Age and Monthly income-Pair plot

```
In [173…   plt.style.use('ggplot')
           g = sns.pairplot(dataset, vars=["MonthlyIncome", "Age"],hue="Attrition",size=5)
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:2095: UserWarning: The
`size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```

Inference

The Income band widens for elder employees between 40-60 years of age and the income band restricts to 10000 for 20-30 year old employees

The employee who were laid off falll under low income(5000-1000) and young age(20-30) category

```
In [174…   dataset.describe(include="object").T   #quick summary of categorical variables
```

Out[174]:

	count	unique	top	freq
Attrition	1470	2	No	1233
BusinessTravel	1470	3	Travel_Rarely	1043
Department	1470	3	Research & Development	961
EducationField	1470	6	Life Sciences	606
Gender	1470	2	Male	882
JobRole	1470	9	Sales Executive	326
MaritalStatus	1470	3	Married	673
Over18	1470	1	Y	1470
OverTime	1470	2	No	1054

Gender count-Bar Plot

In [175…

```
dataset['Gender'].value_counts().plot(kind='bar',color=['salmon','lightblue'],title
```

Out[175]:

```
<Axes: title={'center': 'Count of different gender'}>
```
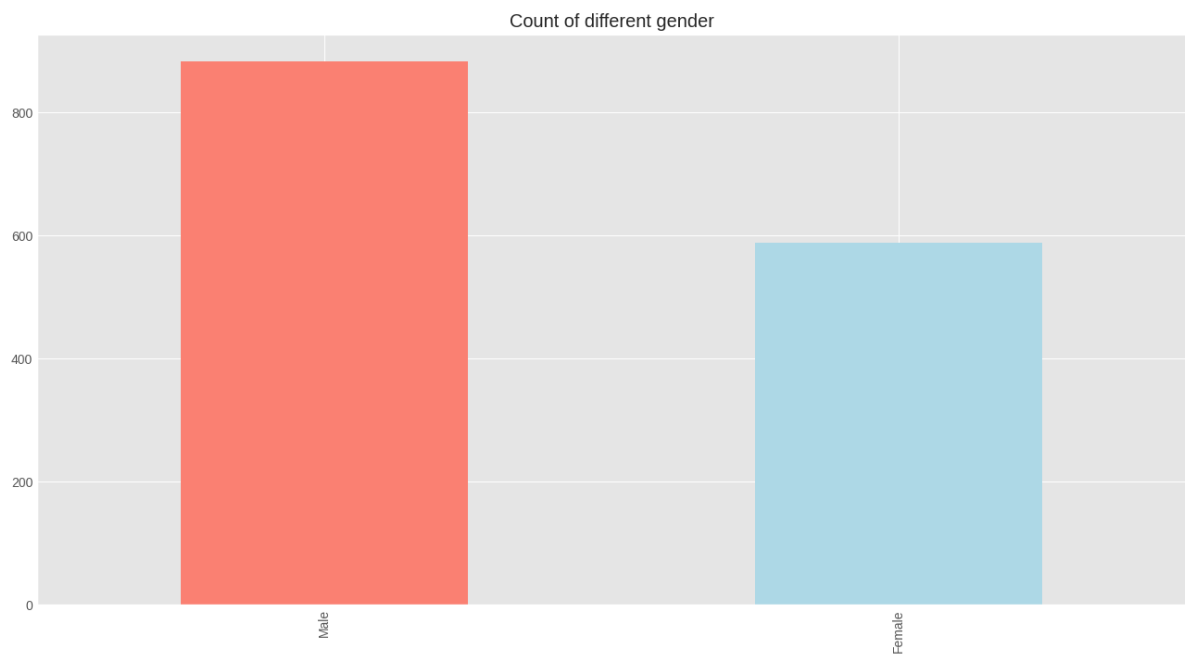

Inference

Shows the gender count.

5. Outlier Detector

In [176…

```
dataset.head()
```

Out[176]:

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

5 rows × 35 columns

In [177…
```python
sns.boxplot(dataset["TotalWorkingYears"])
```

Out[177]: `<Axes: >`

In [178…
```python
q1=dataset.TotalWorkingYears.quantile(0.25)
q3=dataset.TotalWorkingYears.quantile(0.75)
print(q1)
print(q3)
IQR=q3-q1
IQR
upper_limit = q3+1.5*IQR
upper_limit
lower_limit = q1-1.5*IQR
lower_limit
dataset.median()
```

```
6.0
15.0
```

<ipython-input-178-7c0c8ed094ab>:11: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
  dataset.median()

```
Out[178]:  Age                          36.0
           DailyRate                   802.0
           DistanceFromHome              7.0
           Education                     3.0
           EmployeeCount                 1.0
           EmployeeNumber             1020.5
           EnvironmentSatisfaction       3.0
           HourlyRate                   66.0
           JobInvolvement                3.0
           JobLevel                      2.0
           JobSatisfaction               3.0
           MonthlyIncome              4919.0
           MonthlyRate               14235.5
           NumCompaniesWorked            2.0
           PercentSalaryHike            14.0
           PerformanceRating             3.0
           RelationshipSatisfaction      3.0
           StandardHours                80.0
           StockOptionLevel              1.0
           TotalWorkingYears            10.0
           TrainingTimesLastYear         3.0
           WorkLifeBalance               3.0
           YearsAtCompany                5.0
           YearsInCurrentRole            3.0
           YearsSinceLastPromotion       1.0
           YearsWithCurrManager          3.0
           dtype: float64
```

In [179… `dataset['TotalWorkingYears']= np.where(dataset['TotalWorkingYears']>upper_limit,30,`

In [180… `sns.boxplot(dataset["TotalWorkingYears"])`

Out[180]: `<Axes: >`
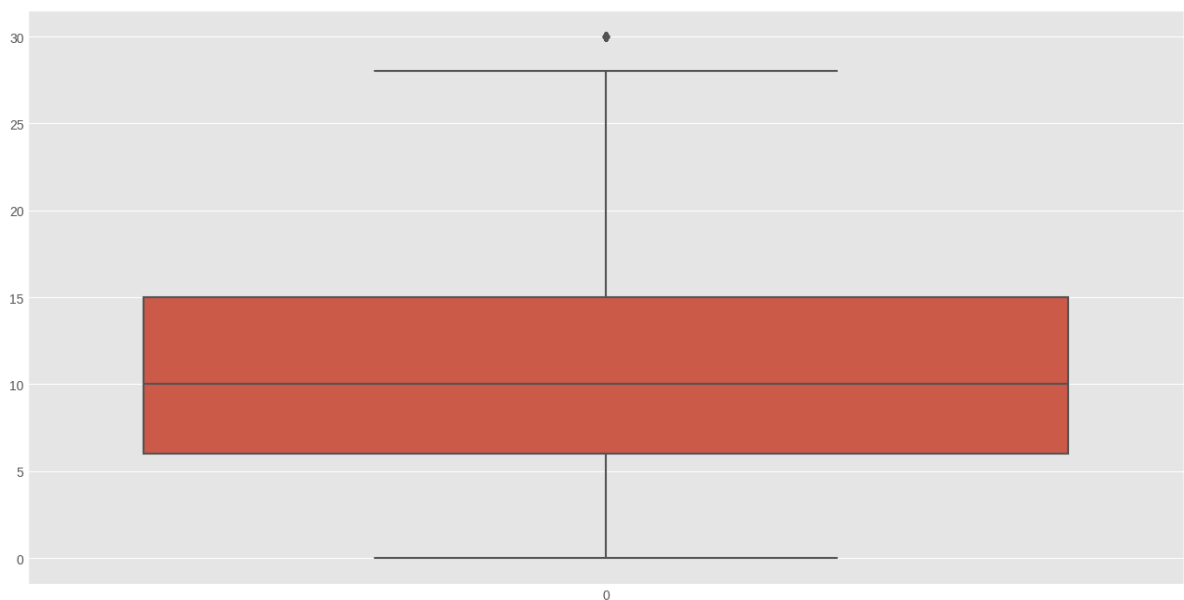
In [181… `sns.boxplot(dataset["YearsInCurrentRole"])`

Out[181]: `<Axes: >`
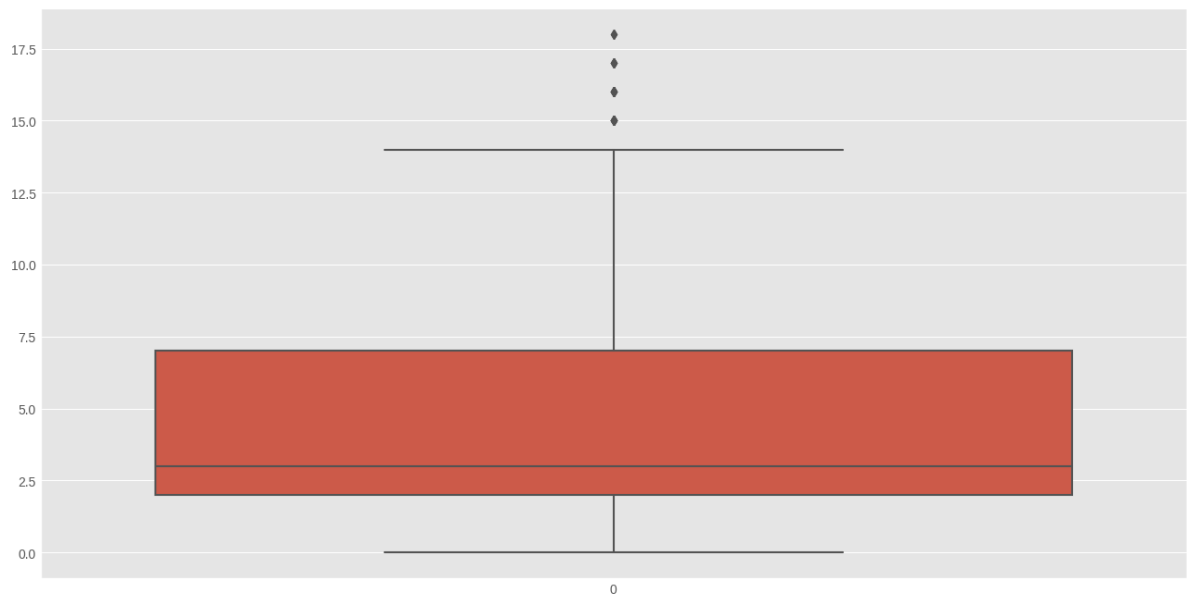
```
In [182…   q1=dataset.YearsInCurrentRole.quantile(0.25)
           q3=dataset.YearsInCurrentRole.quantile(0.75)
           print(q1)
           print(q3)
           IQR=q3-q1
           IQR
           upper_limit = q3+1.5*IQR
           upper_limit
           lower_limit = q1-1.5*IQR
           lower_limit
           dataset.median()
```

```
2.0
7.0
```

```
<ipython-input-182-830ef09e96cd>:11: FutureWarning: The default value of numeric_o
nly in DataFrame.median is deprecated. In a future version, it will default to Fal
se. In addition, specifying 'numeric_only=None' is deprecated. Select only valid c
olumns or specify the value of numeric_only to silence this warning.
  dataset.median()
```

Out[182]:
```
Age                          36.0
DailyRate                   802.0
DistanceFromHome              7.0
Education                     3.0
EmployeeCount                 1.0
EmployeeNumber             1020.5
EnvironmentSatisfaction       3.0
HourlyRate                   66.0
JobInvolvement                3.0
JobLevel                      2.0
JobSatisfaction               3.0
MonthlyIncome              4919.0
MonthlyRate               14235.5
NumCompaniesWorked            2.0
PercentSalaryHike            14.0
PerformanceRating             3.0
RelationshipSatisfaction      3.0
StandardHours                80.0
StockOptionLevel              1.0
TotalWorkingYears            10.0
TrainingTimesLastYear         3.0
WorkLifeBalance               3.0
YearsAtCompany                5.0
YearsInCurrentRole            3.0
YearsSinceLastPromotion       1.0
YearsWithCurrManager          3.0
dtype: float64
```

In [183...
```python
dataset['YearsInCurrentRole']= np.where(dataset['YearsInCurrentRole']>upper_limit,
```

In [184...
```python
sns.boxplot(dataset["YearsInCurrentRole"])
```

Out[184]:   <Axes: >
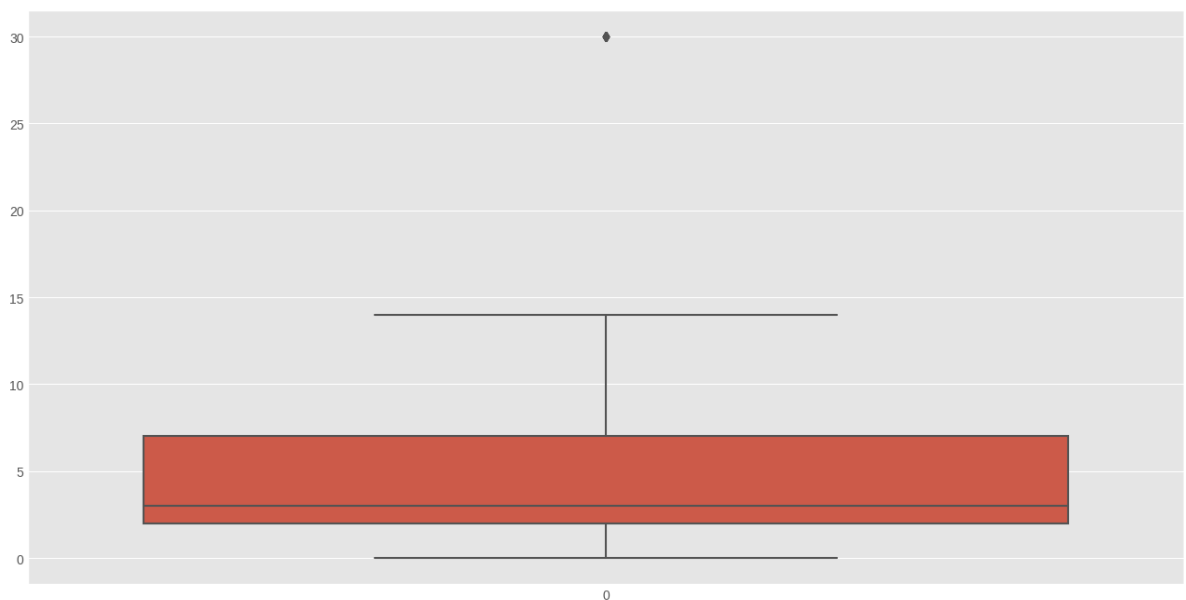
6. Seperate Dependent and Independent variables.

In [185...
```python
dataset.head()
```

Out[185]:

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

5 rows × 35 columns

In [186…
```python
y=dataset['Attrition']
y
```

Out[186]:
```
0       Yes
1        No
2       Yes
3        No
4        No
       ...
1465     No
1466     No
1467     No
1468     No
1469     No
Name: Attrition, Length: 1470, dtype: object
```

In [187…
```python
x=dataset.drop(labels='Attrition',axis=1)
```

In [188…
```python
x.head()
```

Out[188]:

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

5 rows × 34 columns

In [189…
```python
y
```

```
Out[189]:  0       Yes
           1        No
           2       Yes
           3        No
           4        No
                   ...
           1465     No
           1466     No
           1467     No
           1468     No
           1469     No
           Name: Attrition, Length: 1470, dtype: object
```

6. Encoding

In [190…] `dataset.head()`

Out[190]:

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

5 rows × 35 columns

In [191…]
```python
s = (dataset.dtypes == 'object')
object_cols = list(s[s].index)

print("Categorical variables:")
print(object_cols)
```
```
Categorical variables:
['Attrition', 'BusinessTravel', 'Department', 'EducationField', 'Gender', 'JobRol
e', 'MaritalStatus', 'Over18', 'OverTime']
```

In [192…]
```python
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

In [193…]
```python
x['Gender'].value_counts()
```

Out[193]:
```
Male      882
Female    588
Name: Gender, dtype: int64
```

In [194…]
```python
x["Gender"]=le.fit_transform(x["Gender"])
x["Gender"]
```

```
Out[194]:  0       0
           1       1
           2       1
           3       0
           4       1
                   ..
           1465    1
           1466    1
           1467    1
           1468    1
           1469    1
           Name: Gender, Length: 1470, dtype: int64
```

In [195…
```python
x["Gender"].value_counts()
```

```
Out[195]:  1    882
           0    588
           Name: Gender, dtype: int64
```

1- male

2- female

In [196…
```python
x["Gender"].nunique()
```

Out[196]: 2

In [197…
```python
x.BusinessTravel.value_counts()
```

```
Out[197]:  Travel_Rarely        1043
           Travel_Frequently     277
           Non-Travel            150
           Name: BusinessTravel, dtype: int64
```

In [198…
```python
x["BusinessTravel"]=le.fit_transform(x["BusinessTravel"])
x["BusinessTravel"]
```

```
Out[198]:  0       2
           1       1
           2       2
           3       1
           4       2
                   ..
           1465    1
           1466    2
           1467    2
           1468    1
           1469    2
           Name: BusinessTravel, Length: 1470, dtype: int64
```

In [199…
```python
x.BusinessTravel.value_counts()
```

```
Out[199]:  2    1043
           1     277
           0     150
           Name: BusinessTravel, dtype: int64
```

0- Non-Travel

1- Travel-frequently

2- Travel-Rarely

In [200… `x.Department.value_counts()`

Out[200]:
```
Research & Development    961
Sales                     446
Human Resources            63
Name: Department, dtype: int64
```

In [201…
```python
x["Department"]=le.fit_transform(x["Department"])
x["Department"]
```

Out[201]:
```
0       2
1       1
2       1
3       1
4       1
       ..
1465    1
1466    1
1467    1
1468    2
1469    1
Name: Department, Length: 1470, dtype: int64
```

In [202… `x.Department.value_counts()`

Out[202]:
```
1    961
2    446
0     63
Name: Department, dtype: int64
```

0- Human Resources

1- Sales

2- Research & Development

In [203… `x['EducationField'].value_counts()`

Out[203]:
```
Life Sciences       606
Medical             464
Marketing           159
Technical Degree    132
Other                82
Human Resources      27
Name: EducationField, dtype: int64
```

In [204…
```python
x["EducationField"]=le.fit_transform(x["EducationField"])
x["EducationField"]
```

Out[204]:
```
0       1
1       1
2       4
3       1
4       3
       ..
1465    3
1466    3
1467    1
1468    3
1469    3
Name: EducationField, Length: 1470, dtype: int64
```

In [205… `x['EducationField'].value_counts()`

Out[205]:
```
1    606
3    464
2    159
5    132
4     82
0     27
Name: EducationField, dtype: int64
```

0-Human Resources

1-Life Sciences

2-Marketing

3-Medical

4-Other

5-Technical Degree

In [206…
```python
x.JobRole.value_counts()
```

Out[206]:
```
Sales Executive              326
Research Scientist           292
Laboratory Technician        259
Manufacturing Director       145
Healthcare Representative    131
Manager                      102
Sales Representative          83
Research Director             80
Human Resources               52
Name: JobRole, dtype: int64
```

In [207…
```python
x["JobRole"]=le.fit_transform(x["JobRole"])
x["JobRole"]
```

Out[207]:
```
0       7
1       6
2       2
3       6
4       2
       ..
1465    2
1466    0
1467    4
1468    7
1469    2
Name: JobRole, Length: 1470, dtype: int64
```

In [208…
```python
x.JobRole.value_counts()
```

Out[208]:
```
7    326
6    292
2    259
4    145
0    131
3    102
8     83
5     80
1     52
Name: JobRole, dtype: int64
```

0-Helathcare Representative

1-Human Resource

2-Laboratory Technician

3-Manager

4-Manufacturing Director

5-Research Director

6-Research Scientist

7-Sales Executive

8-Sales Representative

```
In [209... x.MaritalStatus.value_counts()
```

```
Out[209]: Married     673
          Single      470
          Divorced    327
          Name: MaritalStatus, dtype: int64
```

```
In [210... x["MaritalStatus"]=le.fit_transform(x["MaritalStatus"])
          x["MaritalStatus"]
```

```
Out[210]: 0       2
          1       1
          2       2
          3       1
          4       1
                 ..
          1465    1
          1466    1
          1467    1
          1468    1
          1469    1
          Name: MaritalStatus, Length: 1470, dtype: int64
```

```
In [211... x.MaritalStatus.value_counts()
```

```
Out[211]: 1    673
          2    470
          0    327
          Name: MaritalStatus, dtype: int64
```

0- Divorced

1- Married

2- Single

```
In [212... x.Over18.value_counts()
```

```
Out[212]: Y    1470
          Name: Over18, dtype: int64
```

```
In [213... x["Over18"]=le.fit_transform(x["Over18"])
          x["Over18"]
```

```
Out[213]:  0         0
           1         0
           2         0
           3         0
           4         0
                    ..
           1465      0
           1466      0
           1467      0
           1468      0
           1469      0
           Name: Over18, Length: 1470, dtype: int64
```

In [214… `x.Over18.value_counts()`

```
Out[214]:  0    1470
           Name: Over18, dtype: int64
```

0- Yes

In [215… `x.OverTime.value_counts()`

```
Out[215]:  No     1054
           Yes     416
           Name: OverTime, dtype: int64
```

In [216…
```
x["OverTime"]=le.fit_transform(x["OverTime"])
x["OverTime"]
```

```
Out[216]:  0         1
           1         0
           2         1
           3         1
           4         0
                    ..
           1465      0
           1466      0
           1467      1
           1468      0
           1469      0
           Name: OverTime, Length: 1470, dtype: int64
```

In [217… `x.OverTime.value_counts()`

```
Out[217]:  0    1054
           1     416
           Name: OverTime, dtype: int64
```

0- No

1- Yes

In [218… `x.shape`

Out[218]: `(1470, 34)`

In [219… `x.head()`

Out[219]:

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

5 rows × 34 columns

In [220... `y.head()`

Out[220]:
```
0    Yes
1     No
2    Yes
3     No
4     No
Name: Attrition, dtype: object
```

In [222... `y.value_counts()`

Out[222]:
```
No     1233
Yes     237
Name: Attrition, dtype: int64
```

In [224...
```
y=le.fit_transform(y)
y
```

Out[224]: `array([1, 0, 1, ..., 0, 0, 0])`

In [227...
```
#feature scaling
from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
x_scaled=pd.DataFrame(ms.fit_transform(x),columns=x.columns)
x_scaled
```

Out[227]:

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

1470 rows × 34 columns

In [228…
```python
#spliting 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_stat
```

In [229…
```python
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[229]:
```
((1176, 34), (294, 34), (1176,), (294,))
```

Model Building

o Import the model building Libraries

o Initializing the model

o Training and testing the model

o Evaluation of Model

o Save the Model

Logistic Regression

In [230…
```python
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
```

In [231…
```python
model.fit(x_train,y_train)
```

Out[231]:
▾ LogisticRegression
LogisticRegression()

In [232...
```python
pred=model.predict(x_test)
pred
```

Out[232]:
```
array([0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 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, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0])
```

In [233...
```python
y_test
```

Out[233]:
```
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, 1, 0, 0, 0, 0, 1, 0, 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, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 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])
```

In [234...
```python
#evaluation of classification model
#Accuracy score
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,
```

In [235...
```python
accuracy_score(y_test,pred)
```

Out[235]:
```
0.8809523809523809
```

In [236...
```python
confusion_matrix(y_test,pred)
```

Out[236]:
```
array([[242,   3],
       [ 32,  17]])
```

In [237...
```python
pd.crosstab(y_test,pred)
```

Out[237]:

col_0	0	1
row_0		
0	242	3
1	32	17

In [238...
```python
(242+17)/(242+17+32+3) #accuracy score
```

Out[238]:
```
0.8809523809523809
```

In [239...   `print(classification_report(y_test,pred))`

```
              precision    recall  f1-score   support

           0       0.88      0.99      0.93       245
           1       0.85      0.35      0.49        49

    accuracy                           0.88       294
   macro avg       0.87      0.67      0.71       294
weighted avg       0.88      0.88      0.86       294
```

In [240...
```
#Roc-AUC Curve
probability=model.predict_proba(x_test)[:,1]
probability
```

```
Out[240]:  array([0.15217402, 0.21632349, 0.31448937, 0.11552231, 0.63195007,
                  0.06110795, 0.61205171, 0.08073788, 0.00740246, 0.42236585,
                  0.05133183, 0.32469008, 0.02237374, 0.6920699 , 0.18976212,
                  0.02894311, 0.11110158, 0.17159697, 0.05136821, 0.21685107,
                  0.23350228, 0.01976108, 0.07290817, 0.05153163, 0.60410648,
                  0.49152701, 0.08029735, 0.05034051, 0.67674278, 0.05779457,
                  0.01808775, 0.03116965, 0.05629576, 0.17137281, 0.07754204,
                  0.04011103, 0.08277844, 0.066222  , 0.04473409, 0.05074883,
                  0.0453298 , 0.02518108, 0.01390803, 0.01430808, 0.02992714,
                  0.47717853, 0.42463578, 0.00342564, 0.73804972, 0.48329632,
                  0.08750612, 0.50973356, 0.08356101, 0.26756878, 0.63342172,
                  0.22261944, 0.01664791, 0.27789879, 0.02738251, 0.1516253 ,
                  0.02255381, 0.21819211, 0.11921312, 0.03932104, 0.32671669,
                  0.03013093, 0.32903292, 0.15813444, 0.09364017, 0.09358887,
                  0.0910726 , 0.31432242, 0.09789755, 0.07483434, 0.12863802,
                  0.05837452, 0.0493793 , 0.06218263, 0.19638662, 0.03325248,
                  0.00761511, 0.02692652, 0.14003752, 0.02550772, 0.03405416,
                  0.08306553, 0.00372029, 0.03841149, 0.0385455 , 0.15582131,
                  0.25915316, 0.16446146, 0.2605143 , 0.2170274 , 0.02308859,
                  0.17411709, 0.34105565, 0.2836754 , 0.06979948, 0.05412718,
                  0.23058355, 0.71467057, 0.29623588, 0.01981856, 0.09155609,
                  0.03062145, 0.06306983, 0.14829133, 0.07471827, 0.131319  ,
                  0.09039381, 0.04550021, 0.0234031 , 0.14007801, 0.07915213,
                  0.03325378, 0.05904139, 0.10244746, 0.00769902, 0.01356191,
                  0.1899012 , 0.04854069, 0.09989016, 0.82471562, 0.02395718,
                  0.02480647, 0.00740718, 0.13995083, 0.1676497 , 0.06179059,
                  0.0153163 , 0.27844187, 0.53392829, 0.36999997, 0.04365391,
                  0.36760733, 0.62429365, 0.1278725 , 0.07357853, 0.20034431,
                  0.08198194, 0.07412797, 0.11514013, 0.22048825, 0.21971106,
                  0.03120375, 0.13280402, 0.00281934, 0.11143174, 0.16183701,
                  0.0632159 , 0.17069567, 0.05761521, 0.12594416, 0.03701899,
                  0.03134031, 0.05938977, 0.0852415 , 0.01621121, 0.01882977,
                  0.3645953 , 0.01470022, 0.1478167 , 0.80600097, 0.11850163,
                  0.28135372, 0.19690363, 0.15121366, 0.02658214, 0.00620571,
                  0.03477834, 0.11262055, 0.10009029, 0.09904116, 0.01267428,
                  0.13933536, 0.10267279, 0.13081418, 0.05184247, 0.07418267,
                  0.02732516, 0.10911227, 0.00416416, 0.75017934, 0.04437636,
                  0.04324425, 0.3940623 , 0.05143664, 0.71301171, 0.09775088,
                  0.37841904, 0.37731749, 0.34241499, 0.04783579, 0.08374629,
                  0.15060733, 0.047484  , 0.01525795, 0.27533347, 0.06190423,
                  0.15515891, 0.15822481, 0.66396633, 0.06007485, 0.30507983,
                  0.05444403, 0.45863441, 0.00260234, 0.14983251, 0.03055447,
                  0.11599359, 0.16925832, 0.08265742, 0.09971406, 0.15622642,
                  0.02095842, 0.01859019, 0.08868262, 0.02462549, 0.13927623,
                  0.09226204, 0.21627391, 0.72737143, 0.15794236, 0.41508798,
                  0.01633093, 0.09689161, 0.21311155, 0.31087529, 0.03492884,
                  0.04588433, 0.32368325, 0.05863244, 0.02807638, 0.17020541,
                  0.28403941, 0.24319541, 0.00853641, 0.08857797, 0.01190675,
                  0.14307901, 0.28821893, 0.01166466, 0.18415389, 0.04799148,
                  0.04083938, 0.40349113, 0.34270419, 0.03801019, 0.13633343,
                  0.3579905 , 0.34773119, 0.79705153, 0.04762894, 0.22929036,
                  0.05777684, 0.00532319, 0.65735633, 0.3691595 , 0.38214768,
                  0.35711771, 0.03375219, 0.22133642, 0.0614842 , 0.07525809,
                  0.09793662, 0.00859601, 0.26368491, 0.39075382, 0.06385042,
                  0.10236432, 0.01395833, 0.14225579, 0.05150353, 0.0246592 ,
                  0.03247425, 0.07929253, 0.25427181, 0.28548567, 0.19978486,
                  0.26904127, 0.01644442, 0.15255598, 0.07893817, 0.02843614,
                  0.19025185, 0.00800355, 0.25237563, 0.00220511, 0.02847355,
                  0.24199687, 0.73687967, 0.06887902, 0.22136975])
```

```python
In [241…  #roc_curve
          fpr,tpr,thresholds=roc_curve(y_test,probability)
```

```
In [242...  plt.plot(fpr,tpr)
            plt.xlabel('FPR')
            plt.ylabel('TPR')
            plt.title('ROC CURVE')
            plt.show()
```
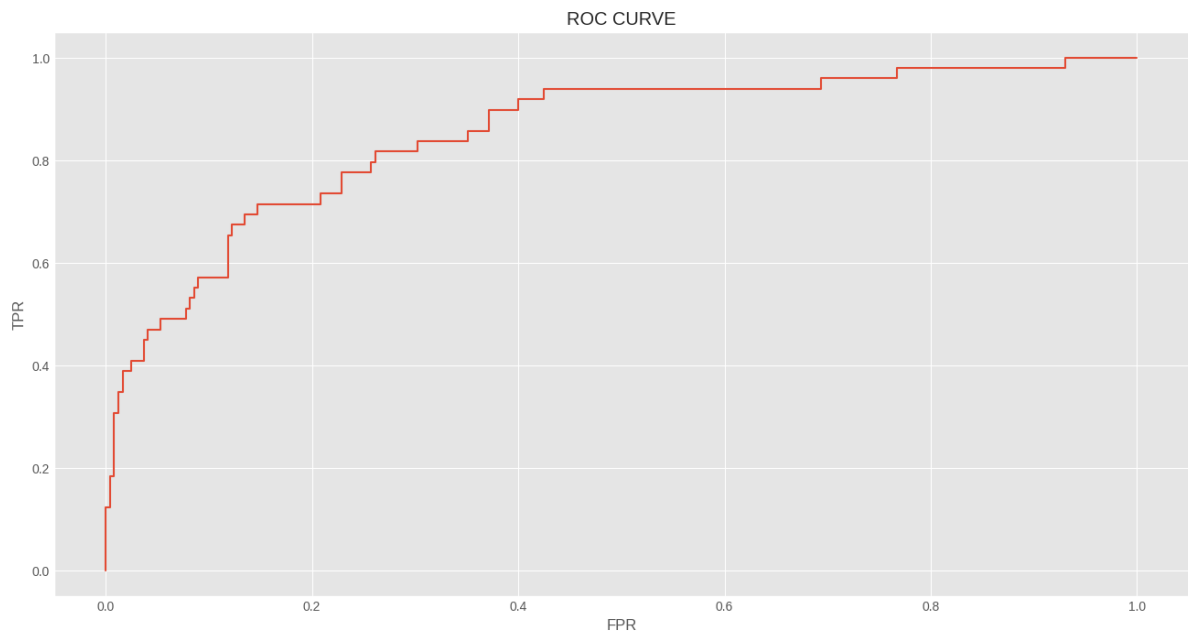

Decision Tree

```
In [243...  from sklearn.tree import DecisionTreeClassifier
            dtc=DecisionTreeClassifier()
```

```
In [244...  dtc.fit(x_train,y_train)
```

```
Out[244]:  ▾ DecisionTreeClassifier
            DecisionTreeClassifier()
```

```
In [246...  pred=dtc.predict(x_test)
            pred
```

```
Out[246]:  array([0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 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, 0, 0,
                   0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
                   0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 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, 0, 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, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
                   0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
                   0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 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, 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, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1,
                   0, 0, 1, 0, 0, 0, 0, 0])
```

```
In [247...  y_test
```

```
Out[247]:  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])
```

In [248...
```python
#evaluation of classification model
#Accuracy score
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,
```

In [249...
```python
accuracy_score(y_test,pred)
```

Out[249]: 0.7619047619047619

In [250...
```python
confusion_matrix(y_test,pred)
```

Out[250]:
```
array([[208,  37],
       [ 33,  16]])
```

In [251...
```python
pd.crosstab(y_test,pred)
```

Out[251]:

col_0	0	1
row_0		
0	208	37
1	33	16

In [252...
```python
#Roc-AUC Curve
probability=dtc.predict_proba(x_test)[:,1]
probability
```

Out[252]:
```
array([0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 1., 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., 0., 0., 0., 0., 0., 0., 1., 1., 1.,
       1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 0., 0.,
       0., 0., 0., 0., 0., 1., 0., 0., 1., 0., 0., 0., 0., 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., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 1., 1., 1., 1., 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., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 1., 0., 1., 1., 0., 0., 0., 0., 1., 0., 0., 1.,
       0., 0., 1., 0., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0.,
       1., 0., 0., 1., 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., 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., 1., 0., 1., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 1., 1., 0., 1., 0., 0., 1.,
       0., 0., 0., 0., 0.])
```

In [253...
```python
#roc_curve
```

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

In [254…
```python
plt.plot(fpr,tpr)
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('ROC CURVE')
plt.show()
```
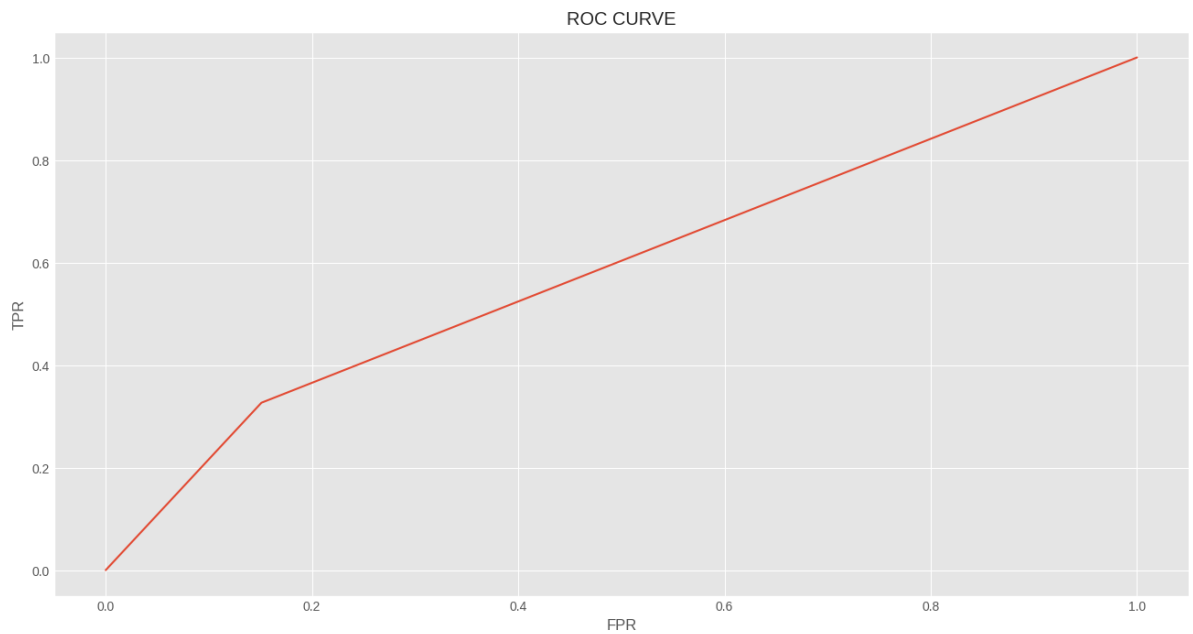

Pre Parameter Tuning

In [256…
```python
from sklearn import tree
plt.figure(figsize=(25,15))
tree.plot_tree(dtc,filled=True)
```

Out[256]:
```
[Text(0.32785292288557216, 0.9722222222222222, 'x[27] <= 0.05\ngini = 0.269\nsampl
es = 1176\nvalue = [988, 188]'),
 Text(0.07960199004975124, 0.9166666666666666, 'x[16] <= 0.75\ngini = 0.5\nsamples
= 78\nvalue = [39, 39]'),
 Text(0.04975124378109453, 0.8611111111111112, 'x[4] <= 0.554\ngini = 0.426\nsampl
es = 39\nvalue = [27, 12]'),
 Text(0.03316749585406302, 0.8055555555555556, 'x[15] <= 0.167\ngini = 0.312\nsamp
les = 31\nvalue = [25, 6]'),
 Text(0.01990049751243781, 0.75, 'x[21] <= 0.5\ngini = 0.49\nsamples = 7\nvalue =
[3, 4]'),
 Text(0.013266998341625208, 0.6944444444444444, 'x[10] <= 0.5\ngini = 0.375\nsampl
es = 4\nvalue = [3, 1]'),
 Text(0.006633499170812604, 0.6388888888888888, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
 Text(0.01990049751243781, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.026533996683250415, 0.6944444444444444, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.04643449419568822, 0.75, 'x[19] <= 0.056\ngini = 0.153\nsamples = 24\nvalu
e = [22, 2]'),
 Text(0.03980099502487562, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.05306799336650083, 0.6944444444444444, 'x[9] <= 0.167\ngini = 0.083\nsampl
es = 23\nvalue = [22, 1]'),
 Text(0.04643449419568822, 0.6388888888888888, 'x[23] <= 0.5\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.03980099502487562, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.05306799336650083, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.05970149253731343, 0.6388888888888888, 'gini = 0.0\nsamples = 21\nvalue =
[21, 0]'),
 Text(0.06633499170812604, 0.8055555555555556, 'x[8] <= 0.385\ngini = 0.375\nsampl
es = 8\nvalue = [2, 6]'),
 Text(0.05970149253731343, 0.75, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.07296849087893864, 0.75, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.10945273631840796, 0.8611111111111112, 'x[11] <= 0.364\ngini = 0.426\nsamp
les = 39\nvalue = [12, 27]'),
 Text(0.09286898839137644, 0.8055555555555556, 'x[29] <= 0.167\ngini = 0.133\nsamp
les = 14\nvalue = [1, 13]'),
 Text(0.08623548922056384, 0.75, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.09950248756218906, 0.75, 'gini = 0.0\nsamples = 13\nvalue = [0, 13]'),
 Text(0.12603648424543948, 0.8055555555555556, 'x[8] <= 0.105\ngini = 0.493\nsampl
es = 25\nvalue = [11, 14]'),
 Text(0.11276948590381426, 0.75, 'x[22] <= 0.464\ngini = 0.278\nsamples = 6\nvalue
= [5, 1]'),
 Text(0.10613598673300166, 0.6944444444444444, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]'),
 Text(0.11940298507462686, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.13930348258706468, 0.75, 'x[15] <= 0.5\ngini = 0.432\nsamples = 19\nvalue
= [6, 13]'),
 Text(0.13266998341625208, 0.6944444444444444, 'gini = 0.0\nsamples = 7\nvalue =
[0, 7]'),
 Text(0.14593698175787728, 0.6944444444444444, 'x[6] <= 0.4\ngini = 0.5\nsamples =
12\nvalue = [6, 6]'),
 Text(0.13266998341625208, 0.6388888888888888, 'x[3] <= 0.75\ngini = 0.278\nsample
s = 6\nvalue = [5, 1]'),
 Text(0.12603648424543948, 0.5833333333333334, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]'),
 Text(0.13930348258706468, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.15920398009950248, 0.6388888888888888, 'x[8] <= 0.249\ngini = 0.278\nsampl
es = 6\nvalue = [1, 5]'),
```

```
 Text(0.15257048092868988, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.16583747927031509, 0.5833333333333334, 'gini = 0.0\nsamples = 5\nvalue =
[0, 5]'),
 Text(0.576103855721393, 0.9166666666666666, 'x[21] <= 0.5\ngini = 0.235\nsamples
= 1098\nvalue = [949, 149]'),
 Text(0.3279954394693201, 0.8611111111111112, 'x[29] <= 0.167\ngini = 0.162\nsampl
es = 798\nvalue = [727, 71]'),
 Text(0.1791044776119403, 0.8055555555555556, 'x[8] <= 0.445\ngini = 0.38\nsamples
= 47\nvalue = [35, 12]'),
 Text(0.16583747927031509, 0.75, 'x[16] <= 0.75\ngini = 0.1\nsamples = 19\nvalue =
[18, 1]'),
 Text(0.15920398009950248, 0.6944444444444444, 'gini = 0.0\nsamples = 18\nvalue =
[18, 0]'),
 Text(0.1724709784411277, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.19237147595356552, 0.75, 'x[17] <= 0.094\ngini = 0.477\nsamples = 28\nvalu
e = [17, 11]'),
 Text(0.1857379767827529, 0.6944444444444444, 'gini = 0.0\nsamples = 4\nvalue =
[0, 4]'),
 Text(0.19900497512437812, 0.6944444444444444, 'x[8] <= 0.524\ngini = 0.413\nsampl
es = 24\nvalue = [17, 7]'),
 Text(0.19237147595356552, 0.6388888888888888, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.20563847429519072, 0.6388888888888888, 'x[33] <= 0.324\ngini = 0.351\nsamp
les = 22\nvalue = [17, 5]'),
 Text(0.19237147595356552, 0.5833333333333334, 'x[2] <= 0.025\ngini = 0.133\nsampl
es = 14\nvalue = [13, 1]'),
 Text(0.1857379767827529, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.19900497512437812, 0.5277777777777778, 'gini = 0.0\nsamples = 13\nvalue =
[13, 0]'),
 Text(0.21890547263681592, 0.5833333333333334, 'x[2] <= 0.329\ngini = 0.5\nsamples
= 8\nvalue = [4, 4]'),
 Text(0.21227197346600332, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.22553897180762852, 0.5277777777777778, 'x[12] <= 0.333\ngini = 0.32\nsampl
es = 5\nvalue = [4, 1]'),
 Text(0.21890547263681592, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.23217247097844113, 0.4722222222222222, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0]'),
 Text(0.47688640132669985, 0.8055555555555556, 'x[30] <= 0.963\ngini = 0.145\nsamp
les = 751\nvalue = [692, 59]'),
 Text(0.4702529021558872, 0.75, 'x[30] <= 0.113\ngini = 0.143\nsamples = 750\nvalu
e = [692, 58]'),
 Text(0.34618573797678276, 0.6944444444444444, 'x[9] <= 0.167\ngini = 0.218\nsampl
es = 257\nvalue = [225, 32]'),
 Text(0.3034825870646766, 0.6388888888888888, 'x[33] <= 0.147\ngini = 0.355\nsampl
es = 65\nvalue = [50, 15]'),
 Text(0.28192371475953565, 0.5833333333333334, 'x[33] <= 0.029\ngini = 0.303\nsamp
les = 59\nvalue = [48, 11]'),
 Text(0.2587064676616915, 0.5277777777777778, 'x[12] <= 0.5\ngini = 0.463\nsample
s = 22\nvalue = [14, 8]'),
 Text(0.24543946932006633, 0.4722222222222222, 'x[11] <= 0.179\ngini = 0.198\nsamp
les = 9\nvalue = [8, 1]'),
 Text(0.23880597014925373, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.25207296849087896, 0.4166666666666667, 'gini = 0.0\nsamples = 8\nvalue =
[8, 0]'),
 Text(0.27197346600331673, 0.4722222222222222, 'x[11] <= 0.4\ngini = 0.497\nsample
s = 13\nvalue = [6, 7]'),
 Text(0.26533996683250416, 0.4166666666666667, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0]'),
```

```
          Text(0.27860696517412936, 0.4166666666666667, 'x[4] <= 0.286\ngini = 0.346\nsampl
es = 9\nvalue = [2, 7]'),
          Text(0.27197346600331673, 0.3611111111111111, 'x[19] <= 0.944\ngini = 0.444\nsamp
les = 3\nvalue = [2, 1]'),
          Text(0.26533996683250416, 0.3055555555555556, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
          Text(0.27860696517412936, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
          Text(0.28524046434494194, 0.3611111111111111, 'gini = 0.0\nsamples = 6\nvalue =
[0, 6]'),
          Text(0.30514096185737977, 0.5277777777777778, 'x[15] <= 0.167\ngini = 0.149\nsamp
les = 37\nvalue = [34, 3]'),
          Text(0.29850746268656714, 0.4722222222222222, 'x[29] <= 0.5\ngini = 0.5\nsamples
= 6\nvalue = [3, 3]'),
          Text(0.29187396351575456, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
          Text(0.30514096185737977, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
          Text(0.3117744610281924, 0.4722222222222222, 'gini = 0.0\nsamples = 31\nvalue =
[31, 0]'),
          Text(0.3250414593698176, 0.5833333333333334, 'x[8] <= 0.065\ngini = 0.444\nsample
s = 6\nvalue = [2, 4]'),
          Text(0.31840796019900497, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
          Text(0.33167495854063017, 0.5277777777777778, 'gini = 0.0\nsamples = 4\nvalue =
[0, 4]'),
          Text(0.3888888888888889, 0.6388888888888888, 'x[0] <= 0.321\ngini = 0.161\nsample
s = 192\nvalue = [175, 17]'),
          Text(0.351575456053068, 0.5833333333333334, 'x[6] <= 0.1\ngini = 0.294\nsamples =
67\nvalue = [55, 12]'),
          Text(0.3449419568822554, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
          Text(0.3582089552238806, 0.5277777777777778, 'x[29] <= 0.5\ngini = 0.26\nsamples
= 65\nvalue = [55, 10]'),
          Text(0.33499170812603646, 0.4722222222222222, 'x[6] <= 0.5\ngini = 0.469\nsamples
= 16\nvalue = [10, 6]'),
          Text(0.3283582089552239, 0.4166666666666667, 'gini = 0.0\nsamples = 7\nvalue =
[7, 0]'),
          Text(0.3416252072968491, 0.4166666666666667, 'x[9] <= 0.833\ngini = 0.444\nsample
s = 9\nvalue = [3, 6]'),
          Text(0.33499170812603646, 0.3611111111111111, 'gini = 0.0\nsamples = 5\nvalue =
[0, 5]'),
          Text(0.3482587064676617, 0.3611111111111111, 'x[18] <= 0.072\ngini = 0.375\nsampl
es = 4\nvalue = [3, 1]'),
          Text(0.3416252072968491, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
          Text(0.3548922056384743, 0.3055555555555556, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
          Text(0.3814262023217247, 0.4722222222222222, 'x[2] <= 0.037\ngini = 0.15\nsamples
= 49\nvalue = [45, 4]'),
          Text(0.3747927031509121, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
          Text(0.3880597014925373, 0.4166666666666667, 'x[2] <= 0.938\ngini = 0.117\nsample
s = 48\nvalue = [45, 3]'),
          Text(0.3814262023217247, 0.3611111111111111, 'x[5] <= 0.875\ngini = 0.081\nsample
s = 47\nvalue = [45, 2]'),
          Text(0.3681592039800995, 0.3055555555555556, 'x[12] <= 0.167\ngini = 0.043\nsampl
es = 45\nvalue = [44, 1]'),
          Text(0.3615257048092869, 0.25, 'x[8] <= 0.839\ngini = 0.444\nsamples = 3\nvalue =
[2, 1]'),
          Text(0.3548922056384743, 0.19444444444444445, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
          Text(0.3681592039800995, 0.19444444444444445, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
```

```
 Text(0.3747927031509121, 0.25, 'gini = 0.0\nsamples = 42\nvalue = [42, 0]'),
 Text(0.3946932006633499, 0.3055555555555556, 'x[8] <= 0.246\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.3880597014925373, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.4013266998341625, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3946932006633499, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.4262023217247098, 0.5833333333333334, 'x[8] <= 0.022\ngini = 0.077\nsample
s = 125\nvalue = [120, 5]'),
 Text(0.4079601990049751, 0.5277777777777778, 'x[27] <= 0.25\ngini = 0.5\nsamples
= 4\nvalue = [2, 2]'),
 Text(0.4013266998341625, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.41459369817578773, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.4444444444444444, 0.5277777777777778, 'x[18] <= 0.968\ngini = 0.048\nsampl
es = 121\nvalue = [118, 3]'),
 Text(0.42786069651741293, 0.4722222222222222, 'x[2] <= 0.98\ngini = 0.033\nsample
s = 118\nvalue = [116, 2]'),
 Text(0.41459369817578773, 0.4166666666666667, 'x[14] <= 0.938\ngini = 0.017\nsamp
les = 114\nvalue = [113, 1]'),
 Text(0.4079601990049751, 0.3611111111111111, 'gini = 0.0\nsamples = 107\nvalue =
[107, 0]'),
 Text(0.42122719734660036, 0.3611111111111111, 'x[12] <= 0.167\ngini = 0.245\nsamp
les = 7\nvalue = [6, 1]'),
 Text(0.41459369817578773, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.42786069651741293, 0.3055555555555556, 'gini = 0.0\nsamples = 6\nvalue =
[6, 0]'),
 Text(0.44112769485903813, 0.4166666666666667, 'x[3] <= 0.75\ngini = 0.375\nsample
s = 4\nvalue = [3, 1]'),
 Text(0.43449419568822556, 0.3611111111111111, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
 Text(0.44776119402985076, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.46102819237147596, 0.4722222222222222, 'x[19] <= 0.278\ngini = 0.444\nsamp
les = 3\nvalue = [2, 1]'),
 Text(0.45439469320066334, 0.4166666666666667, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.46766169154228854, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.5943200663349917, 0.6944444444444444, 'x[30] <= 0.787\ngini = 0.1\nsamples
= 493\nvalue = [467, 26]'),
 Text(0.5601160862354893, 0.6388888888888888, 'x[15] <= 0.5\ngini = 0.094\nsamples
= 486\nvalue = [462, 24]'),
 Text(0.511608623548922, 0.5833333333333334, 'x[14] <= 0.938\ngini = 0.154\nsample
s = 191\nvalue = [175, 16]'),
 Text(0.5049751243781094, 0.5277777777777778, 'x[18] <= 0.481\ngini = 0.145\nsampl
es = 190\nvalue = [175, 15]'),
 Text(0.48756218905472637, 0.4722222222222222, 'x[18] <= 0.47\ngini = 0.221\nsampl
es = 95\nvalue = [83, 12]'),
 Text(0.48092868988391374, 0.4166666666666667, 'x[33] <= 0.794\ngini = 0.207\nsamp
les = 94\nvalue = [83, 11]'),
 Text(0.47429519071310117, 0.3611111111111111, 'x[5] <= 0.375\ngini = 0.192\nsampl
es = 93\nvalue = [83, 10]'),
 Text(0.4527363184079602, 0.3055555555555556, 'x[6] <= 0.9\ngini = 0.363\nsamples
= 21\nvalue = [16, 5]'),
 Text(0.4461028192371476, 0.25, 'x[17] <= 0.413\ngini = 0.266\nsamples = 19\nvalue
= [16, 3]'),
 Text(0.43283582089552236, 0.19444444444444445, 'x[8] <= 0.215\ngini = 0.117\nsamp
les = 16\nvalue = [15, 1]'),
 Text(0.4262023217247098, 0.1388888888888889, 'x[0] <= 0.369\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.41956882255389716, 0.08333333333333333, 'gini = 0.0\nsamples = 1\nvalue =
```

```
[0, 1]'),
 Text(0.43283582089552236, 0.08333333333333333, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.439469320066335, 0.1388888888888889, 'gini = 0.0\nsamples = 14\nvalue = [1
4, 0]'),
 Text(0.4593698175787728, 0.19444444444444445, 'x[24] <= 0.833\ngini = 0.444\nsamp
les = 3\nvalue = [1, 2]'),
 Text(0.4527363184079602, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.4660033167495854, 0.1388888888888889, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.4593698175787728, 0.25, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.49585406301824214, 0.3055555555555556, 'x[31] <= 0.083\ngini = 0.129\nsamp
les = 72\nvalue = [67, 5]'),
 Text(0.4792703150912106, 0.25, 'x[8] <= 0.68\ngini = 0.444\nsamples = 6\nvalue =
[4, 2]'),
 Text(0.472636815920398, 0.19444444444444445, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0]'),
 Text(0.4859038142620232, 0.19444444444444445, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.5124378109452736, 0.25, 'x[11] <= 0.993\ngini = 0.087\nsamples = 66\nvalue
= [63, 3]'),
 Text(0.49917081260364843, 0.19444444444444445, 'x[28] <= 0.583\ngini = 0.061\nsam
ples = 64\nvalue = [62, 2]'),
 Text(0.4925373134328358, 0.1388888888888889, 'gini = 0.0\nsamples = 51\nvalue =
[51, 0]'),
 Text(0.5058043117744611, 0.1388888888888889, 'x[9] <= 0.5\ngini = 0.26\nsamples =
13\nvalue = [11, 2]'),
 Text(0.4991708126036484 3, 0.08333333333333333, 'x[17] <= 0.335\ngini = 0.5\nsampl
es = 4\nvalue = [2, 2]'),
 Text(0.4925373134328358, 0.027777777777777776, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.5058043117744611, 0.027777777777777776, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.5124378109452736, 0.08333333333333333, 'gini = 0.0\nsamples = 9\nvalue =
[9, 0]'),
 Text(0.5257048092868989, 0.19444444444444445, 'x[1] <= 0.25\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.5190713101160862, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.5323383084577115, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.48756218905472637, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.494195688225539, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.5223880597014925, 0.4722222222222222, 'x[19] <= 0.5\ngini = 0.061\nsamples
= 95\nvalue = [92, 3]'),
 Text(0.5157545605306799, 0.4166666666666667, 'gini = 0.0\nsamples = 76\nvalue =
[76, 0]'),
 Text(0.5290215588723052, 0.4166666666666667, 'x[33] <= 0.088\ngini = 0.266\nsampl
es = 19\nvalue = [16, 3]'),
 Text(0.5157545605306799, 0.3611111111111111, 'x[3] <= 0.75\ngini = 0.444\nsamples
= 3\nvalue = [1, 2]'),
 Text(0.5091210613598673, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.5223880597014925, 0.3055555555555556, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.5422885572139303, 0.3611111111111111, 'x[17] <= 0.108\ngini = 0.117\nsampl
es = 16\nvalue = [15, 1]'),
 Text(0.5356550580431177, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.548922056384743, 0.3055555555555556, 'gini = 0.0\nsamples = 15\nvalue = [1
5, 0]'),
```

```
 Text(0.5182421227197347, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.6086235489220564, 0.5833333333333334, 'x[22] <= 0.036\ngini = 0.053\nsampl
es = 295\nvalue = [287, 8]'),
 Text(0.5854063018242123, 0.5277777777777778, 'x[32] <= 0.7\ngini = 0.159\nsamples
= 46\nvalue = [42, 4]'),
 Text(0.5787728026533997, 0.4722222222222222, 'x[12] <= 0.167\ngini = 0.124\nsampl
es = 45\nvalue = [42, 3]'),
 Text(0.5621890547263682, 0.4166666666666667, 'x[27] <= 0.4\ngini = 0.5\nsamples =
2\nvalue = [1, 1]'),
 Text(0.5555555555555556, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.5688225538971807, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.5953565505804311, 0.4166666666666667, 'x[27] <= 0.917\ngini = 0.089\nsampl
es = 43\nvalue = [41, 2]'),
 Text(0.582089552238806, 0.3611111111111111, 'x[14] <= 0.062\ngini = 0.048\nsample
s = 41\nvalue = [40, 1]'),
 Text(0.5754560530679934, 0.3055555555555556, 'x[9] <= 0.167\ngini = 0.375\nsample
s = 4\nvalue = [3, 1]'),
 Text(0.5688225538971807, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.582089552238806, 0.25, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.5887230514096186, 0.3055555555555556, 'gini = 0.0\nsamples = 37\nvalue =
[37, 0]'),
 Text(0.6086235489220564, 0.3611111111111111, 'x[30] <= 0.212\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.6019900497512438, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.615257048092869, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.5920398009950248, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.6318407960199005, 0.5277777777777778, 'x[17] <= 0.056\ngini = 0.032\nsampl
es = 249\nvalue = [245, 4]'),
 Text(0.615257048092869, 0.4722222222222222, 'x[16] <= 0.75\ngini = 0.32\nsamples
= 5\nvalue = [4, 1]'),
 Text(0.6086235489220564, 0.4166666666666667, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0]'),
 Text(0.6218905472636815, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.648424543946932, 0.4722222222222222, 'x[2] <= 0.015\ngini = 0.024\nsamples
= 244\nvalue = [241, 3]'),
 Text(0.6351575456053068, 0.4166666666666667, 'x[11] <= 0.129\ngini = 0.278\nsampl
es = 6\nvalue = [5, 1]'),
 Text(0.6285240464344942, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.6417910447761194, 0.3611111111111111, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]'),
 Text(0.6616915422885572, 0.4166666666666667, 'x[24] <= 0.167\ngini = 0.017\nsampl
es = 238\nvalue = [236, 2]'),
 Text(0.6550580431177446, 0.3611111111111111, 'x[29] <= 0.833\ngini = 0.073\nsampl
es = 53\nvalue = [51, 2]'),
 Text(0.6417910447761194, 0.3055555555555556, 'x[33] <= 0.088\ngini = 0.041\nsampl
es = 48\nvalue = [47, 1]'),
 Text(0.6351575456053068, 0.25, 'x[14] <= 0.312\ngini = 0.245\nsamples = 7\nvalue
= [6, 1]'),
 Text(0.6285240464344942, 0.19444444444444445, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.6417910447761194, 0.19444444444444445, 'gini = 0.0\nsamples = 6\nvalue =
[6, 0]'),
 Text(0.648424543946932, 0.25, 'gini = 0.0\nsamples = 41\nvalue = [41, 0]'),
 Text(0.6683250414593698, 0.3055555555555556, 'x[31] <= 0.25\ngini = 0.32\nsamples
= 5\nvalue = [4, 1]'),
 Text(0.6616915422885572, 0.25, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
```

```
 Text(0.6749585406301825, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6683250414593698, 0.3611111111111111, 'gini = 0.0\nsamples = 185\nvalue =
[185, 0]'),
 Text(0.6285240464344942, 0.6388888888888888, 'x[2] <= 0.366\ngini = 0.408\nsample
s = 7\nvalue = [5, 2]'),
 Text(0.6218905472636815, 0.5833333333333334, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.6351575456053068, 0.5833333333333334, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]'),
 Text(0.4835199004975124, 0.75, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.824212271973466, 0.8611111111111112, 'x[17] <= 0.157\ngini = 0.385\nsample
s = 300\nvalue = [222, 78]'),
 Text(0.7371475953565506, 0.8055555555555556, 'x[26] <= 0.167\ngini = 0.5\nsamples
= 96\nvalue = [49, 47]'),
 Text(0.7014925373134329, 0.75, 'x[4] <= 0.161\ngini = 0.459\nsamples = 42\nvalue
= [15, 27]'),
 Text(0.6749585406301825, 0.6944444444444444, 'x[8] <= 0.415\ngini = 0.499\nsample
s = 23\nvalue = [12, 11]'),
 Text(0.6550580431177446, 0.6388888888888888, 'x[18] <= 0.561\ngini = 0.355\nsampl
es = 13\nvalue = [3, 10]'),
 Text(0.648424543946932, 0.5833333333333334, 'gini = 0.0\nsamples = 8\nvalue = [0,
8]'),
 Text(0.6616915422885572, 0.5833333333333334, 'x[28] <= 0.583\ngini = 0.48\nsample
s = 5\nvalue = [3, 2]'),
 Text(0.6550580431177446, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
 Text(0.6683250414593698, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.6948590381426202, 0.6388888888888888, 'x[24] <= 0.167\ngini = 0.18\nsample
s = 10\nvalue = [9, 1]'),
 Text(0.6882255389718076, 0.5833333333333334, 'x[32] <= 0.067\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.681592039800995, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.6948590381426202, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.7014925373134329, 0.5833333333333334, 'gini = 0.0\nsamples = 8\nvalue =
[8, 0]'),
 Text(0.7280265339966833, 0.6944444444444444, 'x[13] <= 0.125\ngini = 0.266\nsampl
es = 19\nvalue = [3, 16]'),
 Text(0.7213930348258707, 0.6388888888888888, 'x[11] <= 0.2\ngini = 0.198\nsamples
= 18\nvalue = [2, 16]'),
 Text(0.714759535655058, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.7280265339966833, 0.5833333333333334, 'x[31] <= 0.183\ngini = 0.111\nsampl
es = 17\nvalue = [1, 16]'),
 Text(0.7213930348258707, 0.5277777777777778, 'gini = 0.0\nsamples = 15\nvalue =
[0, 15]'),
 Text(0.7346600331674958, 0.5277777777777778, 'x[18] <= 0.162\ngini = 0.5\nsamples
= 2\nvalue = [1, 1]'),
 Text(0.7280265339966833, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.7412935323383084, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.7346600331674958, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.7728026533996684, 0.75, 'x[0] <= 0.202\ngini = 0.466\nsamples = 54\nvalue
= [34, 20]'),
 Text(0.7545605306799337, 0.6944444444444444, 'x[12] <= 0.833\ngini = 0.245\nsampl
es = 7\nvalue = [1, 6]'),
 Text(0.7479270315091211, 0.6388888888888888, 'gini = 0.0\nsamples = 6\nvalue =
[0, 6]'),
 Text(0.7611940298507462, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
```

 Text(0.7910447761194029, 0.6944444444444444, 'x[2] <= 0.622\ngini = 0.418\nsample
s = 47\nvalue = [33, 14]'),
 Text(0.7744610281923715, 0.6388888888888888, 'x[2] <= 0.145\ngini = 0.482\nsample
s = 32\nvalue = [19, 13]'),
 Text(0.7611940298507462, 0.5833333333333334, 'x[2] <= 0.024\ngini = 0.18\nsamples
= 10\nvalue = [9, 1]'),
 Text(0.7545605306799337, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.7678275290215588, 0.5277777777777778, 'gini = 0.0\nsamples = 9\nvalue =
[9, 0]'),
 Text(0.7877280265339967, 0.5833333333333334, 'x[18] <= 0.87\ngini = 0.496\nsample
s = 22\nvalue = [10, 12]'),
 Text(0.7810945273631841, 0.5277777777777778, 'x[8] <= 0.41\ngini = 0.465\nsamples
= 19\nvalue = [7, 12]'),
 Text(0.7678275290215588, 0.4722222222222222, 'x[18] <= 0.715\ngini = 0.469\nsampl
es = 8\nvalue = [5, 3]'),
 Text(0.7611940298507462, 0.4166666666666667, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]'),
 Text(0.7744610281923715, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.7943615257048093, 0.4722222222222222, 'x[0] <= 0.25\ngini = 0.298\nsamples
= 11\nvalue = [2, 9]'),
 Text(0.7877280265339967, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.8009950248756219, 0.4166666666666667, 'x[18] <= 0.202\ngini = 0.18\nsample
s = 10\nvalue = [1, 9]'),
 Text(0.7943615257048093, 0.3611111111111111, 'x[15] <= 0.5\ngini = 0.5\nsamples =
2\nvalue = [1, 1]'),
 Text(0.7877280265339967, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.8009950248756219, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.8076285240464345, 0.3611111111111111, 'gini = 0.0\nsamples = 8\nvalue =
[0, 8]'),
 Text(0.7943615257048093, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]'),
 Text(0.8076285240464345, 0.6388888888888888, 'x[11] <= 0.064\ngini = 0.124\nsampl
es = 15\nvalue = [14, 1]'),
 Text(0.8009950248756219, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.814262023217247, 0.5833333333333334, 'gini = 0.0\nsamples = 14\nvalue = [1
4, 0]'),
 Text(0.9112769485903814, 0.8055555555555556, 'x[16] <= 0.75\ngini = 0.258\nsample
s = 204\nvalue = [173, 31]'),
 Text(0.8590381426202321, 0.75, 'x[17] <= 0.992\ngini = 0.138\nsamples = 147\nvalu
e = [136, 11]'),
 Text(0.8524046434494196, 0.6944444444444444, 'x[4] <= 0.482\ngini = 0.128\nsample
s = 146\nvalue = [136, 10]'),
 Text(0.8341625207296849, 0.6388888888888888, 'x[30] <= 0.063\ngini = 0.038\nsampl
es = 104\nvalue = [102, 2]'),
 Text(0.8275290215588723, 0.5833333333333334, 'x[11] <= 0.193\ngini = 0.32\nsample
s = 10\nvalue = [8, 2]'),
 Text(0.8208955223880597, 0.5277777777777778, 'x[28] <= 0.417\ngini = 0.444\nsampl
es = 3\nvalue = [1, 2]'),
 Text(0.814262023217247, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.8275290215588723, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.8341625207296849, 0.5277777777777778, 'gini = 0.0\nsamples = 7\nvalue =
[7, 0]'),
 Text(0.8407960199004975, 0.5833333333333334, 'gini = 0.0\nsamples = 94\nvalue =
[94, 0]'),
 Text(0.8706467661691543, 0.6388888888888888, 'x[9] <= 0.167\ngini = 0.308\nsample
s = 42\nvalue = [34, 8]'),

```
        Text(0.8540630182421227, 0.5833333333333334, 'x[29] <= 0.833\ngini = 0.375\nsampl
es = 4\nvalue = [1, 3]'),
 Text(0.8474295190713101, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.8606965174129353, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.8872305140961857, 0.5833333333333334, 'x[0] <= 0.393\ngini = 0.229\nsample
s = 38\nvalue = [33, 5]'),
 Text(0.8739635157545605, 0.5277777777777778, 'x[9] <= 0.5\ngini = 0.5\nsamples =
6\nvalue = [3, 3]'),
 Text(0.867330016583748, 0.4722222222222222, 'x[5] <= 0.625\ngini = 0.375\nsamples
= 4\nvalue = [1, 3]'),
 Text(0.8606965174129353, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.8739635157545605, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.8805970149253731, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]'),
 Text(0.900497512437811, 0.5277777777777778, 'x[8] <= 0.992\ngini = 0.117\nsamples
= 32\nvalue = [30, 2]'),
 Text(0.8938640132669984, 0.4722222222222222, 'x[28] <= 0.917\ngini = 0.062\nsampl
es = 31\nvalue = [30, 1]'),
 Text(0.8872305140961857, 0.4166666666666667, 'gini = 0.0\nsamples = 30\nvalue =
[30, 0]'),
 Text(0.900497512437811, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.9071310116086235, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.8656716417910447, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
 Text(0.9635157545605307, 0.75, 'x[14] <= 0.812\ngini = 0.456\nsamples = 57\nvalue
= [37, 20]'),
 Text(0.9402985074626866, 0.6944444444444444, 'x[32] <= 0.4\ngini = 0.238\nsamples
= 29\nvalue = [25, 4]'),
 Text(0.9270315091210614, 0.6388888888888888, 'x[8] <= 0.071\ngini = 0.142\nsample
s = 26\nvalue = [24, 2]'),
 Text(0.9203980099502488, 0.5833333333333334, 'x[33] <= 0.412\ngini = 0.444\nsampl
es = 3\nvalue = [1, 2]'),
 Text(0.9137645107794361, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.9270315091210614, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.9336650082918739, 0.5833333333333334, 'gini = 0.0\nsamples = 23\nvalue =
[23, 0]'),
 Text(0.9535655058043118, 0.6388888888888888, 'x[2] <= 0.324\ngini = 0.444\nsample
s = 3\nvalue = [1, 2]'),
 Text(0.9469320066334992, 0.5833333333333334, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]'),
 Text(0.9601990049751243, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.9867330016583747, 0.6944444444444444, 'x[32] <= 0.1\ngini = 0.49\nsamples
= 28\nvalue = [12, 16]'),
 Text(0.9800995024875622, 0.6388888888888888, 'x[4] <= 0.804\ngini = 0.48\nsamples
= 20\nvalue = [12, 8]'),
 Text(0.9734660033167496, 0.5833333333333334, 'x[30] <= 0.013\ngini = 0.415\nsampl
es = 17\nvalue = [12, 5]'),
 Text(0.966832504145937, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue = [0,
2]'),
 Text(0.9800995024875622, 0.5277777777777778, 'x[24] <= 0.5\ngini = 0.32\nsamples
= 15\nvalue = [12, 3]'),
 Text(0.9734660033167496, 0.4722222222222222, 'x[0] <= 0.286\ngini = 0.5\nsamples
= 6\nvalue = [3, 3]'),
 Text(0.966832504145937, 0.4166666666666667, 'gini = 0.0\nsamples = 2\nvalue = [2,
0]'),
```

```
 Text(0.9800995024875622, 0.4166666666666667, 'x[2] <= 0.197\ngini = 0.375\nsample
s = 4\nvalue = [1, 3]'),
 Text(0.9734660033167496, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
 Text(0.9867330016583747, 0.3611111111111111, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.9867330016583747, 0.4722222222222222, 'gini = 0.0\nsamples = 9\nvalue =
[9, 0]'),
 Text(0.9867330016583747, 0.5833333333333334, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
 Text(0.9933665008291874, 0.6388888888888888, 'gini = 0.0\nsamples = 8\nvalue =
[0, 8]')]
```
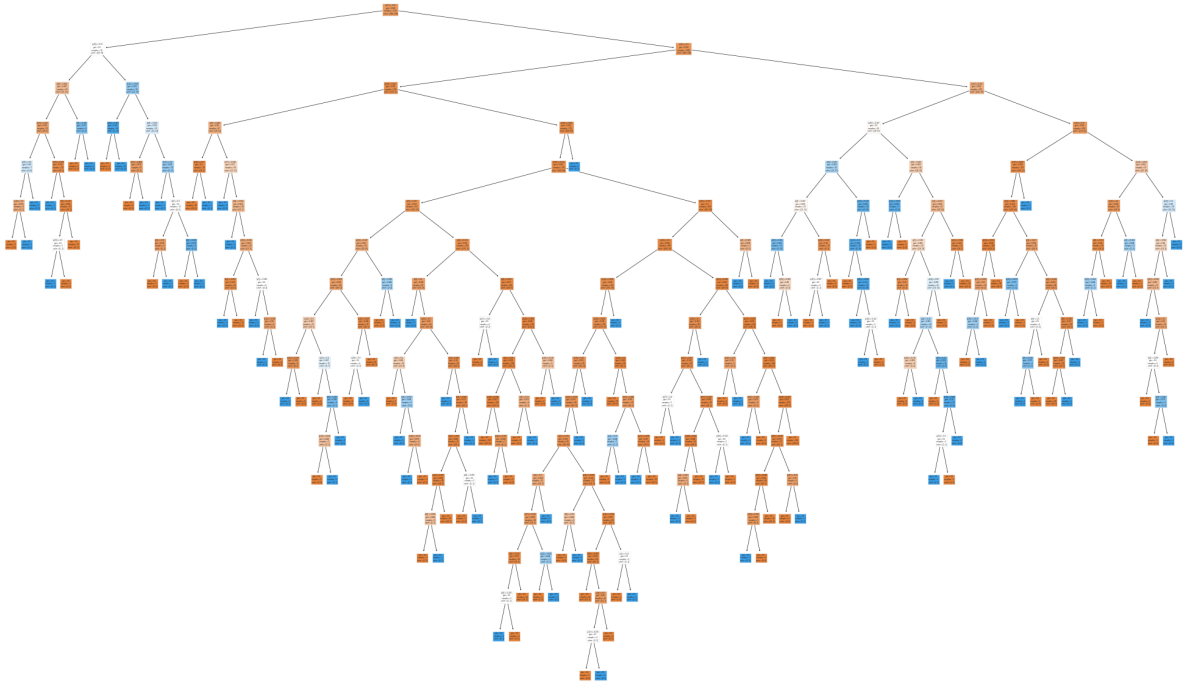


```python
from sklearn.model_selection import GridSearchCV
parameter={
    'criterion':['gini','entropy'],
    'splitter':['best','random'],
    'max_depth':[1,2,3,4,5],
    'max_features':['auto','sqrt','log2']
}
```

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

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

```
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
g: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
```

Out[260]:
```
▸        GridSearchCV

▸ estimator: DecisionTreeClassifier

    ▸ DecisionTreeClassifier
```

In [261…]
```python
grid_search.best_params_
```

Out[261]:
```
{'criterion': 'entropy',
 'max_depth': 3,
 'max_features': 'sqrt',
 'splitter': 'best'}
```

In [265…]
```python
dtc_cv=DecisionTreeClassifier(criterion= 'entropy',
 max_depth= 3,
 max_features= 'sqrt',
 splitter= 'best')
dtc_cv.fit(x_train,y_train)
```

Out[265]:
```
▾                    DecisionTreeClassifier

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

In [268…]
```python
pred=dtc_cv.predict(x_test)
```

In [270…]
```python
print(classification_report(y_test,pred))
```

```
              precision    recall  f1-score   support

           0       0.83      1.00      0.91       245
           1       0.00      0.00      0.00        49

    accuracy                           0.83       294
   macro avg       0.42      0.50      0.45       294
weighted avg       0.69      0.83      0.76       294
```

After hyperparameter tuning, accuracy is 83%

Random Forest

In [271…]
```python
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
```

In [272...
```python
forest_params=[{'max_depth': list(range(10,15)), 'max_features': list(range(0,14))}
```

In [273...
```python
rfc_cv=GridSearchCV(rfc,forest_params,cv=10,scoring="accuracy")
```

In [274...
```python
rfc_cv.fit(x_train,y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_validation.py:37
8: 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_sco
re='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/_validatio
n.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 _va
lidate_params
    validate_parameter_constraints(
  File "/usr/local/lib/python3.10/dist-packages/sklearn/utils/_param_validation.p
y", line 97, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'max_features' paramete
r of RandomForestClassifier must be an int in the range [1, inf), a float in the r
ange (0.0, 1.0], a str among {'sqrt', 'auto' (deprecated), 'log2'} or None. Got 0
instead.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_search.py:952: Us
erWarning: One or more of the test scores are non-finite: [       nan 0.85120238
 0.85630885 0.85459221 0.85291178 0.86053165
 0.86137911 0.85798204 0.85882949 0.85966971 0.85798204 0.85966247
 0.8605389  0.85797479        nan 0.8443865  0.85204259 0.86224105
 0.86052441 0.85967695 0.85969144 0.86138635 0.86477618 0.86137911
 0.86393597 0.86137911 0.86052441 0.86307403        nan 0.85034043
 0.85373026 0.85714182 0.85712734 0.85712009 0.85714182 0.85627264
 0.86221932 0.85457048 0.86222657 0.85627988 0.85542518 0.85795306
        nan 0.85204983 0.85969868 0.85712734 0.85457048 0.85966247
 0.86137911 0.86051717 0.86478343 0.85287556 0.85628712 0.85798204
 0.86052441 0.86647834        nan 0.84950746 0.85546139 0.85971317
 0.86138635 0.85627988 0.85967695 0.85882225 0.86223381 0.85628712
 0.86221932 0.85458496 0.85880776 0.85882225]
  warnings.warn(
```
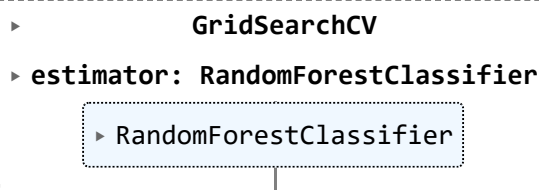
Out[274]:
```
 ▸          GridSearchCV
 ▸ estimator: RandomForestClassifier

      ▸ RandomForestClassifier
```

In [275...
```python
pred=rfc_cv.predict(x_test)
```

In [276...
```python
print(classification_report(y_test,pred))
```

```
              precision    recall  f1-score   support

           0       0.86      0.98      0.91       245
           1       0.62      0.20      0.31        49

    accuracy                           0.85       294
   macro avg       0.74      0.59      0.61       294
weighted avg       0.82      0.85      0.81       294
```

Accuracy of this model is 85%

In [ ]: