

# Assignment 4

Name: N Lohitha

Reg No: 21BCE2807

## Import the Libraries

In [1]:

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

## Importing the dataset

In [2]:

```
df=pd.read_csv("WA_Fn-UseC_-HR-Employee-Attrition.csv")
```

## Data Preprocessing

In [3]:

```
df.head()
```

Out[3]:

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

5 rows × 35 columns

In [4]:

```
df.shape
```

Out[4]: (1470, 35)

In [5]:

```
df.info()
```

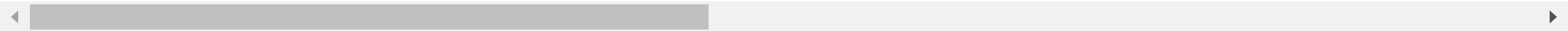
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Age                                   1470 non-null   int64
1   Attrition                           1470 non-null 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

```
In [6]: df.describe()
```

Out[6]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	Jobl
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	1470.000000	1470.000000	1470.000000	1470.00
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	2.721769	65.891156	2.729932	2.06
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	1.093082	20.329428	0.711561	1.10
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	1.000000	30.000000	1.000000	1.00
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	2.000000	48.000000	2.000000	1.00
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	3.000000	66.000000	3.000000	2.00
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	4.000000	83.750000	3.000000	3.00
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	4.000000	100.000000	4.000000	5.00

8 rows × 26 columns



# Checking for Null Values

```
In [7]: df.isnull().any()
```

```
Out[7]: 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 [8]: df.isnull().sum()
```

```
Out[8]: Age 0
Attrition 0
BusinessTravel 0
DailyRate 0
Department 0
DistanceFromHome 0
Education 0
EducationField 0
EmployeeCount 0
EmployeeNumber 0
EnvironmentSatisfaction 0
Gender 0
HourlyRate 0
JobInvolvement 0
JobLevel 0
JobRole 0
JobSatisfaction 0
MaritalStatus 0
MonthlyIncome 0
MonthlyRate 0
NumCompaniesWorked 0
Over18 0
OverTime 0
PercentSalaryHike 0
PerformanceRating 0
RelationshipSatisfaction 0
StandardHours 0
StockOptionLevel 0
TotalWorkingYears 0
TrainingTimesLastYear 0
WorkLifeBalance 0
YearsAtCompany 0
YearsInCurrentRole 0
YearsSinceLastPromotion 0
YearsWithCurrManager 0
dtype: int64
```

# Data Visualization.

## Distribution Plot

```
In [92]: sns.distplot(df["Age"])
```

```
C:\Users\lohitha\AppData\Local\Temp\ipykernel_22084\2732350774.py:1: UserWarning:
```

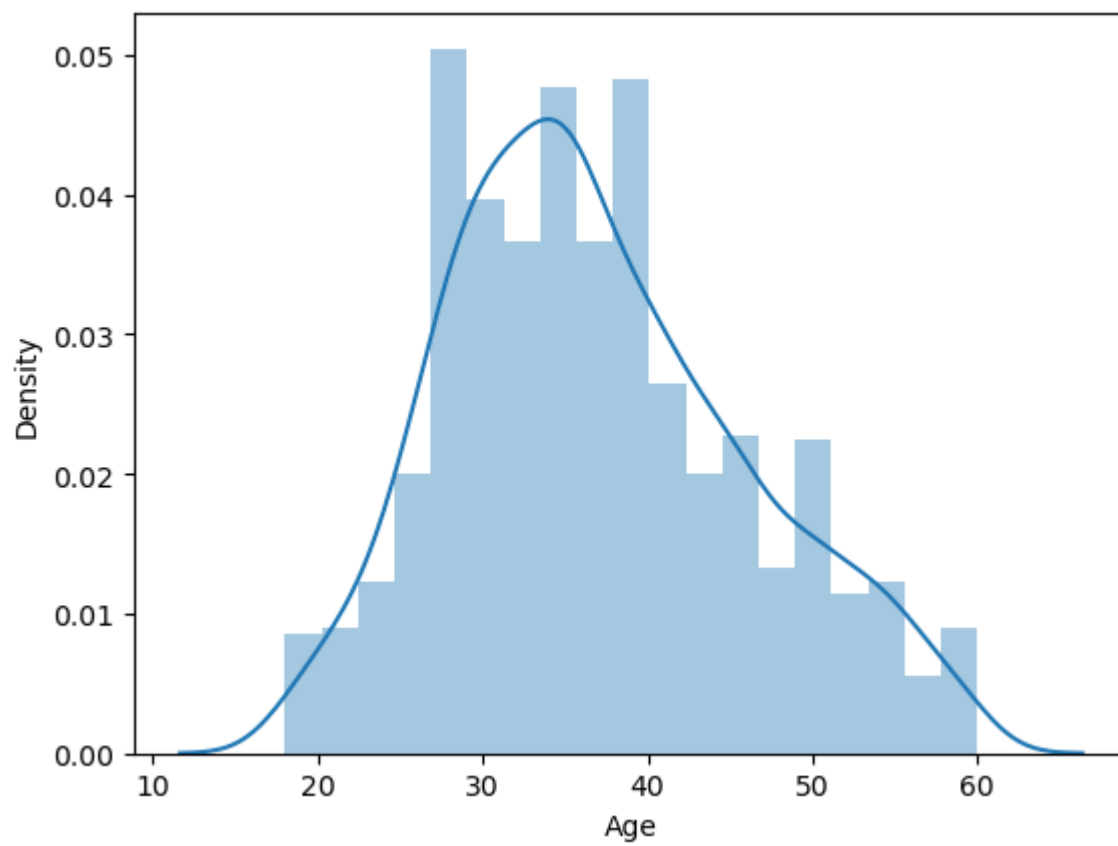
```
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
```

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

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

```
sns.distplot(df["Age"])
<Axes: xlabel='Age', ylabel='Density'>
```

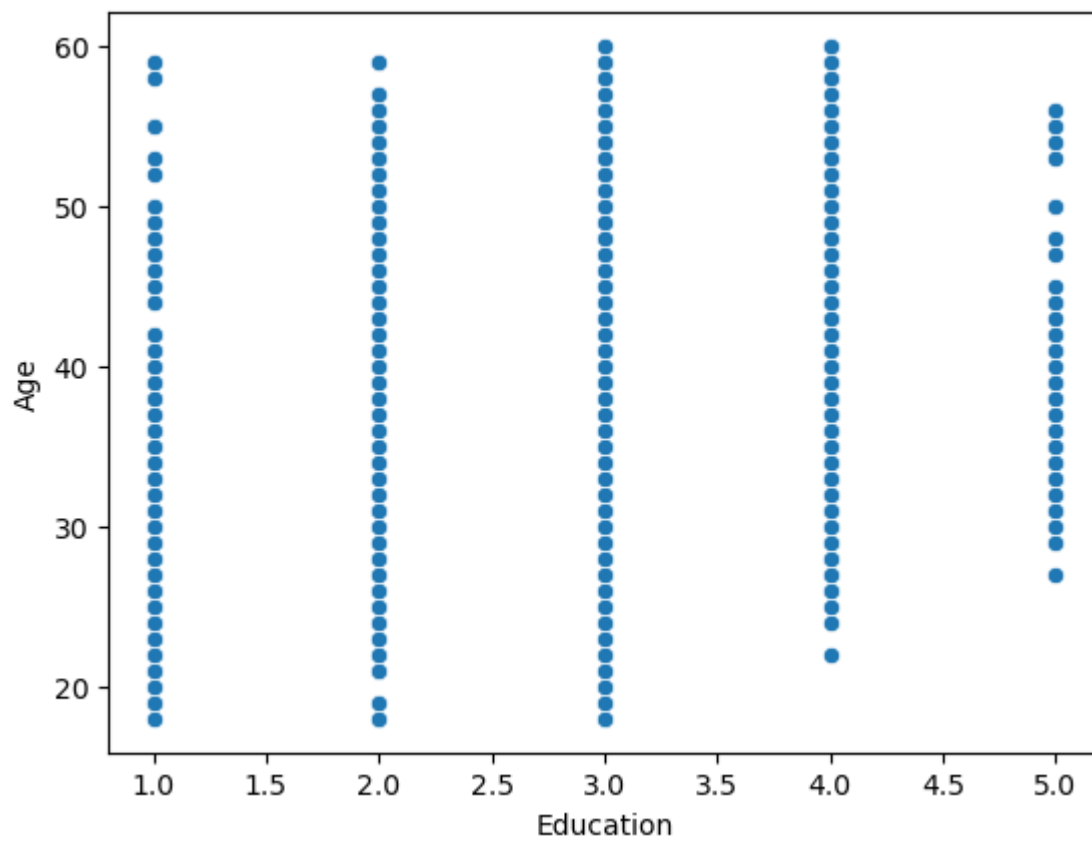
Out[92]:



## Scatterplot

```
In [87]: sns.scatterplot(x="Education", y="Age", data=df)
```

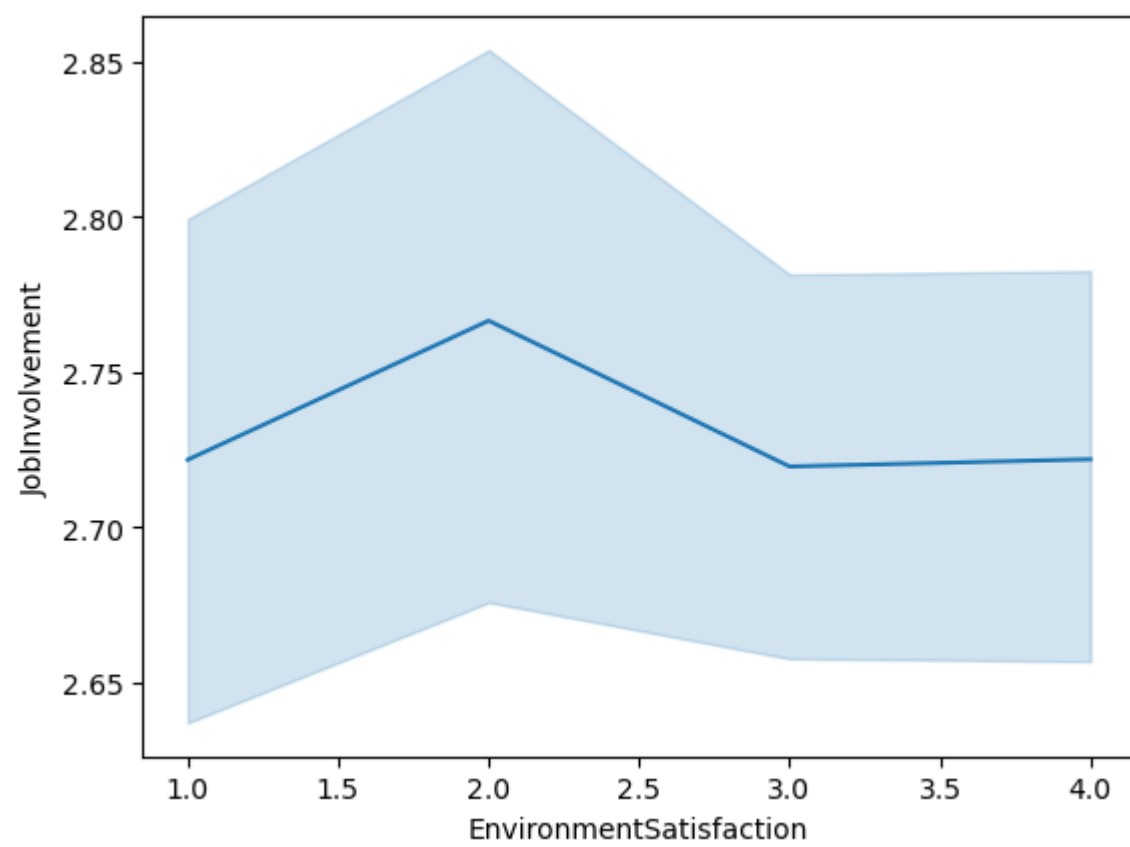
Out[87]: <Axes: xlabel='Education', ylabel='Age'>



## Line Plot

```
In [90]: sns.lineplot(x="EnvironmentSatisfaction", y="JobInvolvement", data=df)
```

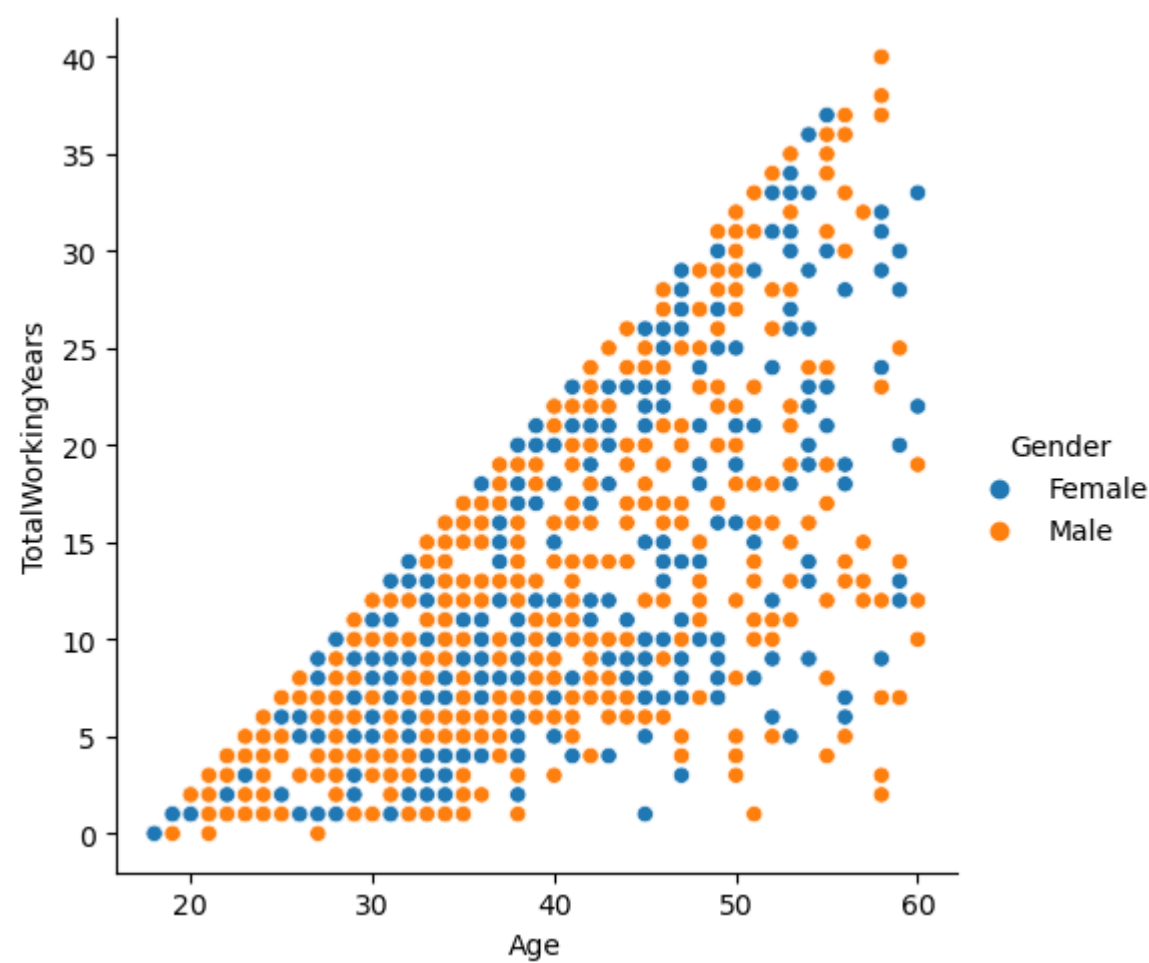
Out[90]: <Axes: xlabel='EnvironmentSatisfaction', ylabel='JobInvolvement'>



## Relational Plot

```
In [94]: sns.relplot(x="Age", y="TotalWorkingYears", data=df, hue="Gender")
```

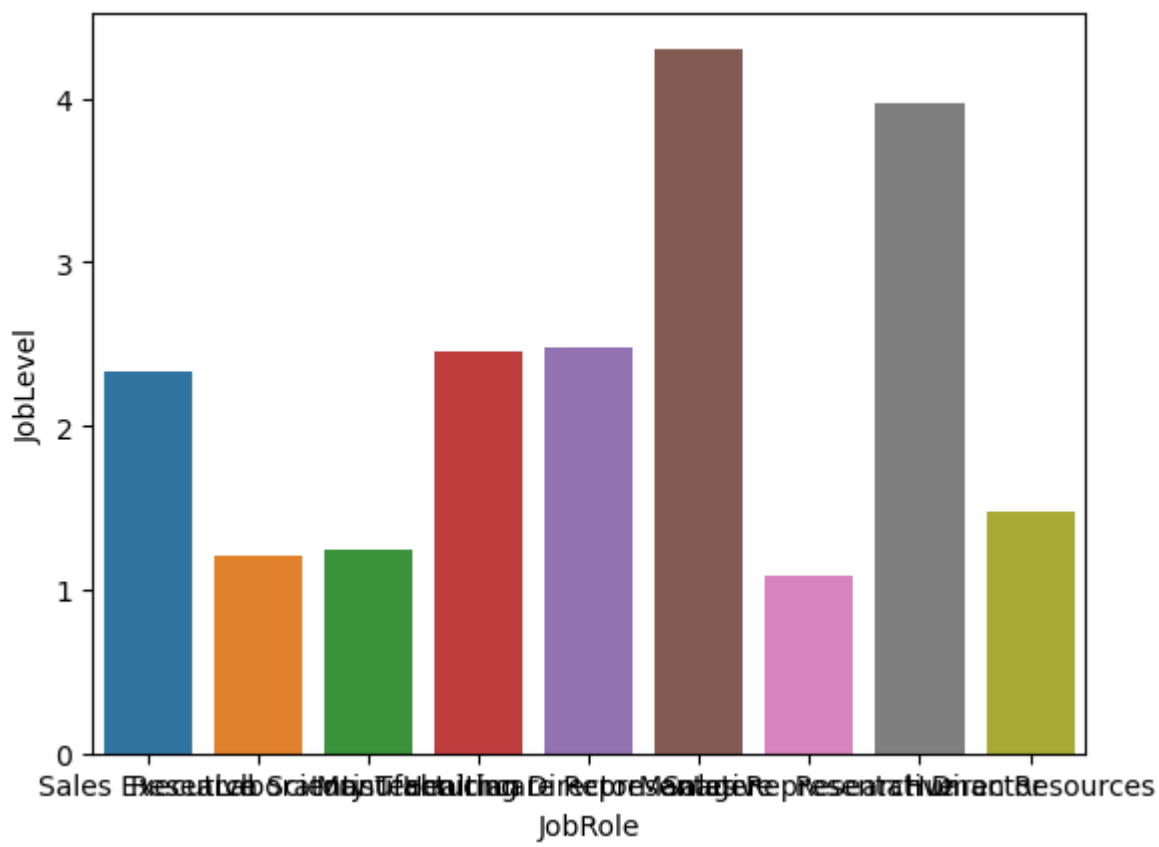
```
Out[94]: <seaborn.axisgrid.FacetGrid at 0x1da436f4990>
```



## Bar plot

```
In [101]: sns.barplot(data=df, x="JobRole", y="JobLevel", errorbar=None)  
plt.figure(figsize=(10,10))
```

```
Out[101]: <Figure size 1000x1000 with 0 Axes>
```

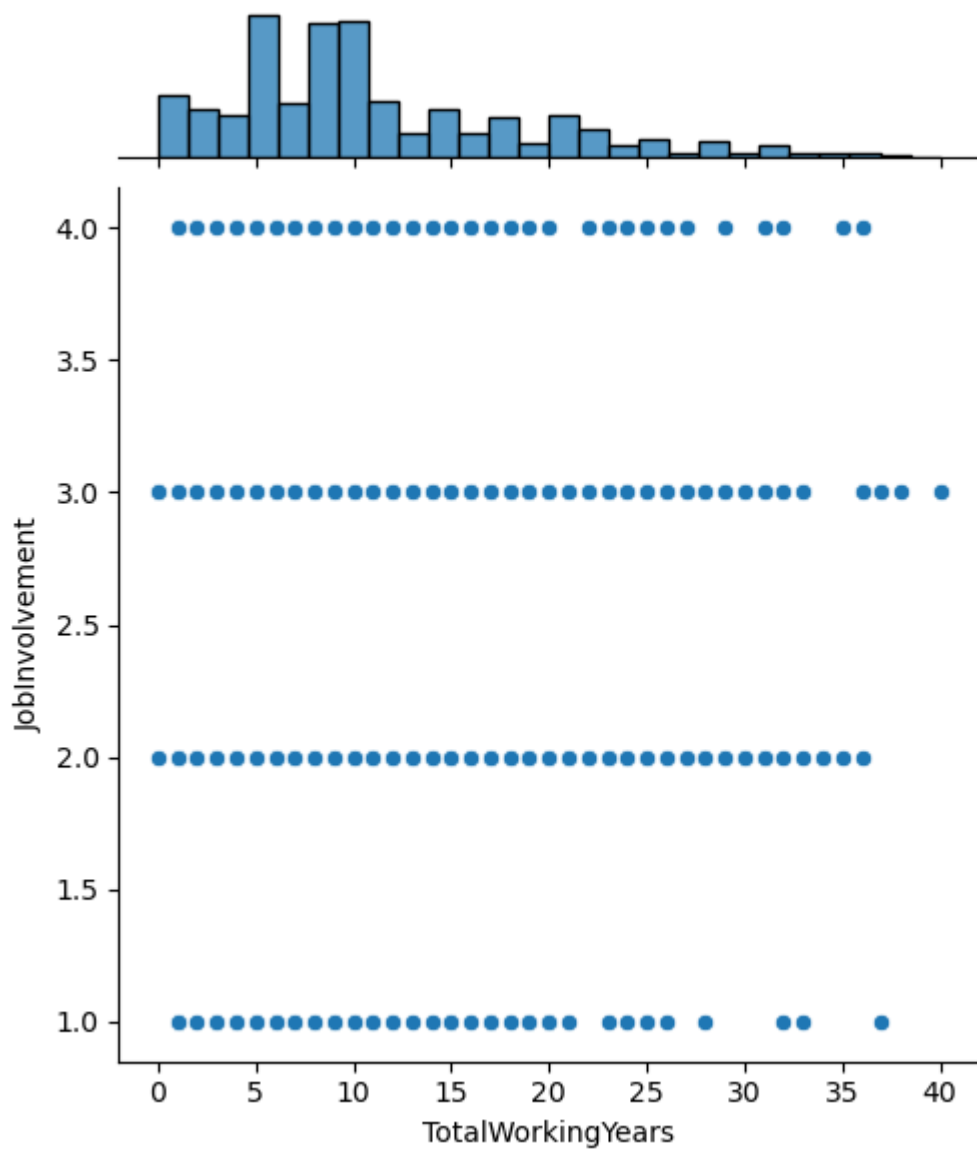


<Figure size 1000x1000 with 0 Axes>

## Joint Plot

```
In [102]: sns.jointplot(x="TotalWorkingYears", y="JobInvolvement", data=df)
```

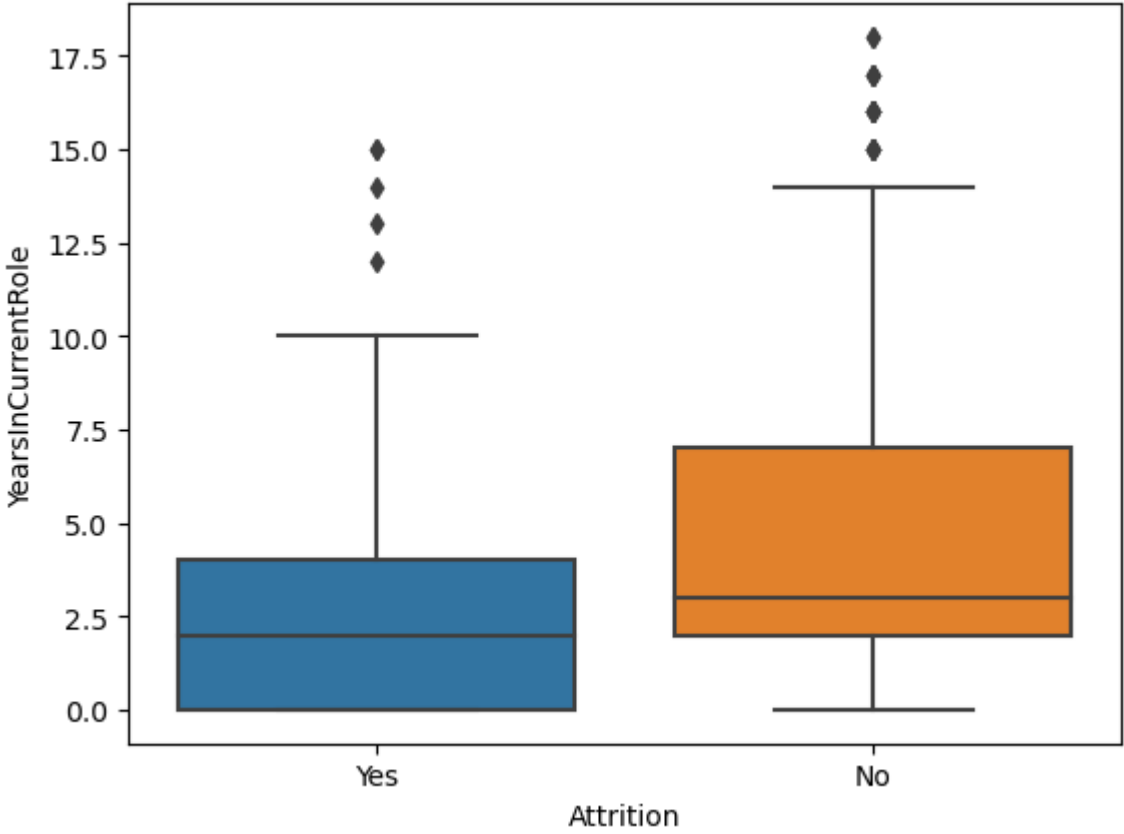
```
Out[102]: <seaborn.axisgrid.JointGrid at 0x1da438ed1d0>
```



## Box Plot

```
In [105]: sns.boxplot(x="Attrition", y="YearsInCurrentRole", data=df)
```

```
Out[105]: <Axes: xlabel='Attrition', ylabel='YearsInCurrentRole'>
```



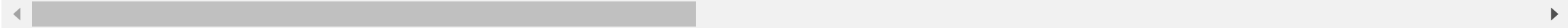
```
In [10]: df.corr()
```

C:\Users\lohitha\AppData\Local\Temp\ipykernel\_22084\1134722465.py:1: FutureWarning: The default value of numeric\_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric\_only to silence this warning.  
df.corr()

Out[10]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement
Age	1.000000	0.010661	-0.001686	0.208034	NaN	-0.010145	0.010146	0.024287	0.029820
DailyRate	0.010661	1.000000	-0.004985	-0.016806	NaN	-0.050990	0.018355	0.023381	0.046135
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042	NaN	0.032916	-0.016075	0.031131	0.008783
Education	0.208034	-0.016806	0.021042	1.000000	NaN	0.042070	-0.027128	0.016775	0.042438
EmployeeCount	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
EmployeeNumber	-0.010145	-0.050990	0.032916	0.042070	NaN	1.000000	0.017621	0.035179	-0.006888
EnvironmentSatisfaction	0.010146	0.018355	-0.016075	-0.027128	NaN	0.017621	1.000000	-0.049857	-0.008278
HourlyRate	0.024287	0.023381	0.031131	0.016775	NaN	0.035179	-0.049857	1.000000	0.042861
JobInvolvement	0.029820	0.046135	0.008783	0.042438	NaN	-0.006888	-0.008278	0.042861	1.000000
JobLevel	0.509604	0.002966	0.005303	0.101589	NaN	-0.018519	0.001212	-0.027853	-0.012661
JobSatisfaction	-0.004892	0.030571	-0.003669	-0.011296	NaN	-0.046247	-0.006784	-0.071335	-0.021438
MonthlyIncome	0.497855	0.007707	-0.017014	0.094961	NaN	-0.014829	-0.006259	-0.015794	-0.015229
MonthlyRate	0.028051	-0.032182	0.027473	-0.026084	NaN	0.012648	0.037600	-0.015297	-0.016317
NumCompaniesWorked	0.299635	0.038153	-0.029251	0.126317	NaN	-0.001251	0.012594	0.022157	0.015000
PercentSalaryHike	0.003634	0.022704	0.040235	-0.011111	NaN	-0.012944	-0.031701	-0.009062	-0.017229
PerformanceRating	0.001904	0.000473	0.027110	-0.024539	NaN	-0.020359	-0.029548	-0.002172	-0.029000
RelationshipSatisfaction	0.053535	0.007846	0.006557	-0.009118	NaN	-0.069861	0.007665	0.001330	0.034200
StandardHours	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
StockOptionLevel	0.037510	0.042143	0.044872	0.018422	NaN	0.062227	0.003432	0.050263	0.021500
TotalWorkingYears	0.680381	0.014515	0.004628	0.148280	NaN	-0.014365	-0.002693	-0.002334	-0.005500
TrainingTimesLastYear	-0.019621	0.002453	-0.036942	-0.025100	NaN	0.023603	-0.019359	-0.008548	-0.015300
WorkLifeBalance	-0.021490	-0.037848	-0.026556	0.009819	NaN	0.010309	0.027627	-0.004607	-0.014600
YearsAtCompany	0.311309	-0.034055	0.009508	0.069114	NaN	-0.011240	0.001458	-0.019582	-0.021300
YearsInCurrentRole	0.212901	0.009932	0.018845	0.060236	NaN	-0.008416	0.018007	-0.024106	0.008700
YearsSinceLastPromotion	0.216513	-0.033229	0.010029	0.054254	NaN	-0.009019	0.016194	-0.026716	-0.024100
YearsWithCurrManager	0.202089	-0.026363	0.014406	0.069065	NaN	-0.009197	-0.004999	-0.020123	0.025900

26 rows × 10 columns

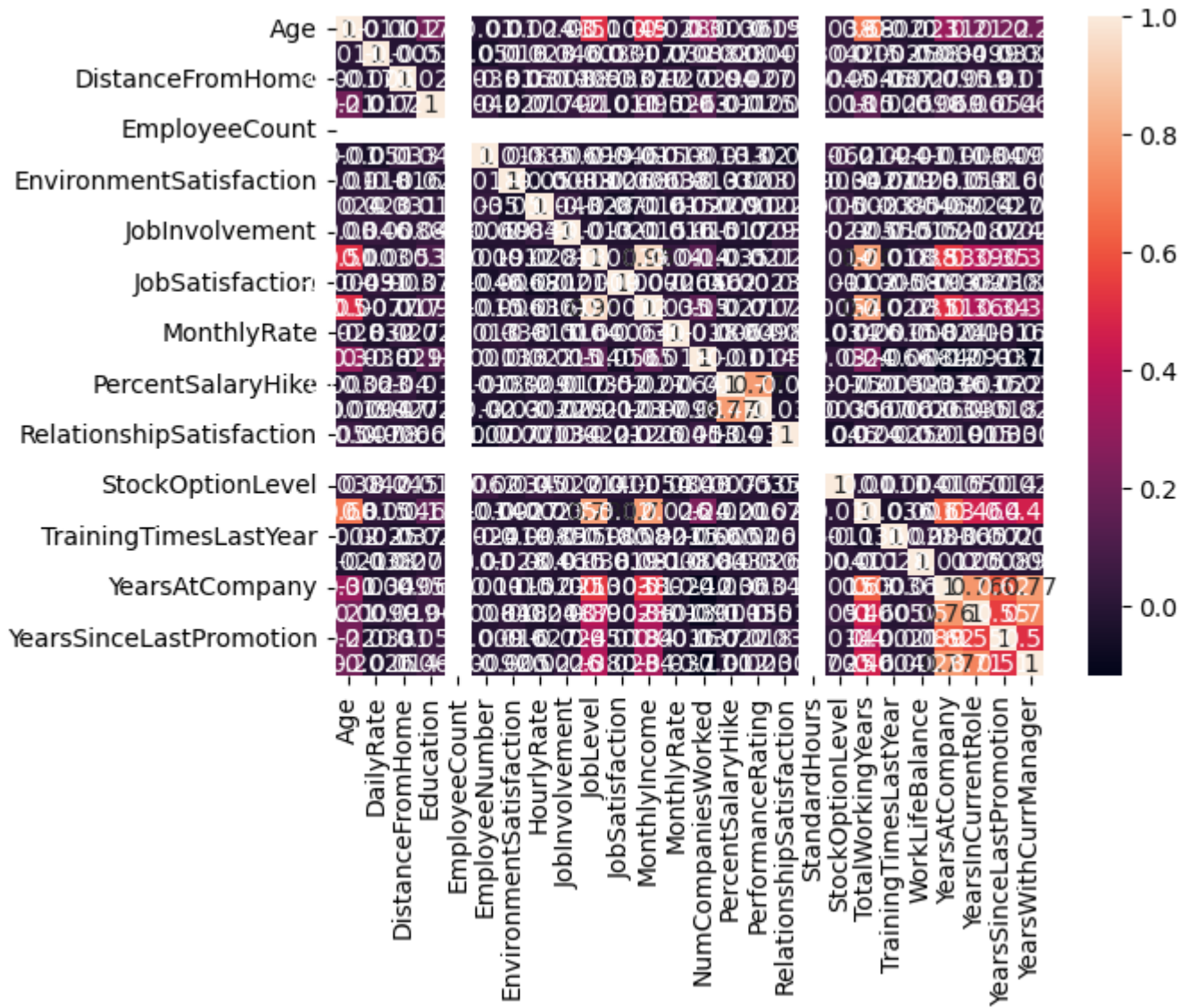


```
In [108]: sns.heatmap(df.corr(),annot=True)
```

C:\Users\lohitha\AppData\Local\Temp\ipykernel\_22084\2157429118.py:1: FutureWarning: The default value of numeric\_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric\_only to silence this warning.  
sns.heatmap(df.corr(),annot=True)

```
Out[108]: <Axes: >
```

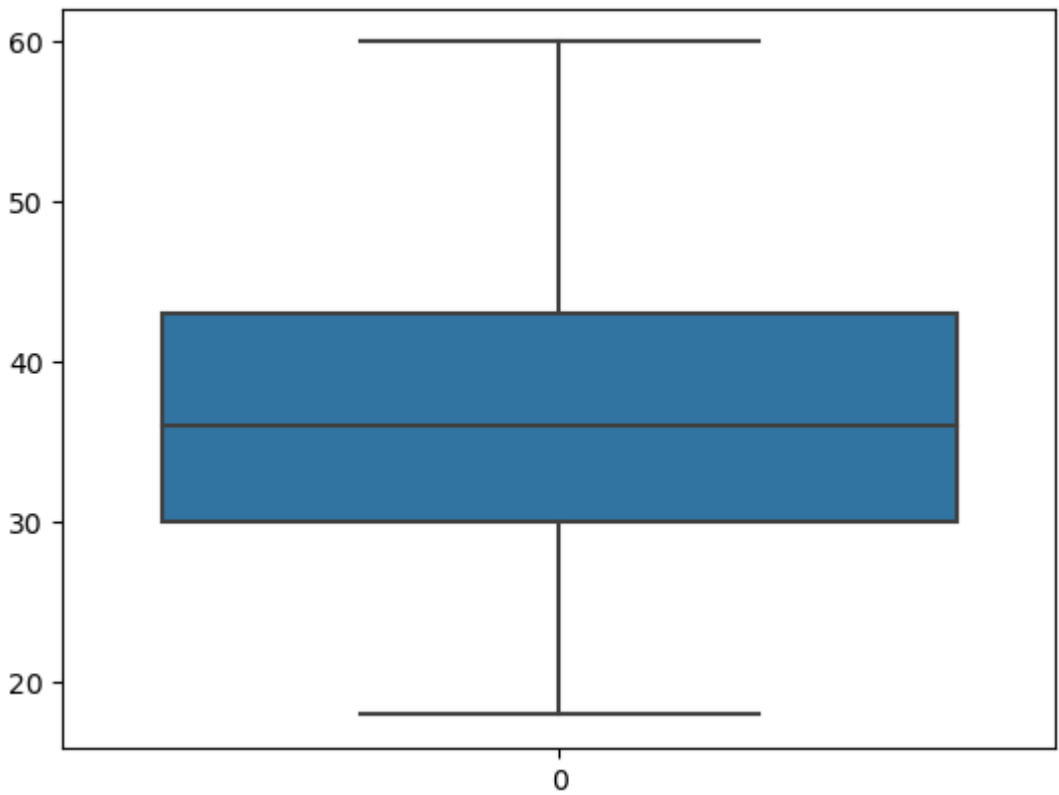




# Outlier Detection

```
In [12]: sns.boxplot(df.Age)
```

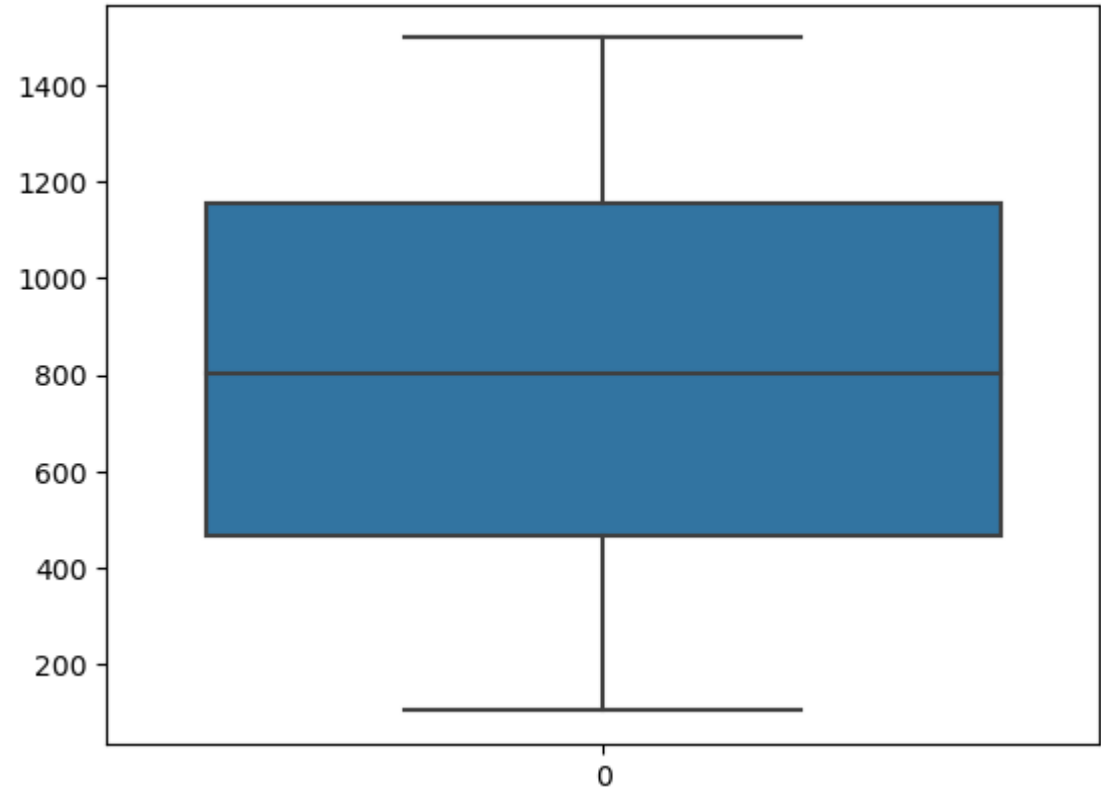
```
Out[12]: <Axes: >
```



```
In [13]: sns.boxplot(df.DailyRate)
```

```
Out[13]: <Axes: >
```





## Splitting Dependent and Independent variables

In [109...

```
x=df.drop(columns=['Attrition'], inplace=False) #independent variable
x
```

Out[109]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	...
0	41	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	2	...
1	49	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	3	...
2	37	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	4	...
3	33	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	4	...
4	27	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	1	...
...	...	...	...	...	...	...	...	...	...	...	...
1465	36	Travel_Frequently	884	Research & Development	23	2	Medical	1	2061	3	...
1466	39	Travel_Rarely	613	Research & Development	6	1	Medical	1	2062	4	...
1467	27	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	2064	2	...
1468	49	Travel_Frequently	1023	Sales	2	3	Medical	1	2065	4	...
1469	34	Travel_Rarely	628	Research & Development	8	3	Medical	1	2068	2	...

1470 rows × 34 columns

◀

▶

In [15]:

df.head()

Out[15]:

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

5 rows × 35 columns

◀

▶

In [110...

```
y=df.iloc[:,1:2] #dependent variable
y
```

Out[110]:

Attrition	
0	Yes
1	No
2	Yes
3	No
4	No
...	...
1465	No
1466	No
1467	No
1468	No
1469	No

1470 rows × 1 columns

## Encoding

In [17]:

```
df.info()
```

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

In [18]:

```
x["BusinessTravel"].value_counts()
```

Out[18]:

```
Travel_Rarely      1043  
Travel_Frequently   277  
Non-Travel         150  
Name: BusinessTravel, dtype: int64
```

In [19]:

```
x["Department"].value_counts()
```

Out[19]:

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

In [20]:

```
x["EducationField"].value_counts()
```

```
Out[20]: Life Sciences      606
Medical      464
Marketing     159
Technical Degree 132
Other         82
Human Resources 27
Name: EducationField, dtype: int64
```

Label encoding on BusinessTravel, Department, EducationField, Gender, JobRole, MaritalStatus, Over18, OverTime, Attrition

```
In [21]: from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
x.BusinessTravel=le.fit_transform(x.BusinessTravel)
x["BusinessTravel"]
```

```
Out[21]: 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: int32
```

```
In [22]: x.Department=le.fit_transform(x.Department)
x["Department"]
```

```
Out[22]: 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: int32
```

```
In [23]: x.EducationField=le.fit_transform(x.EducationField)
x["EducationField"]
```

```
Out[23]: 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: int32
```

```
In [24]: x.Gender=le.fit_transform(x.Gender)
x["Gender"]
```

```
Out[24]: 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: int32
```

```
In [25]: x.JobRole=le.fit_transform(x.JobRole)
x["JobRole"]
```

```
Out[25]: 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: int32
```

```
In [26]: x.MaritalStatus=le.fit_transform(x.MaritalStatus)
x["MaritalStatus"]
```

```
Out[26]: 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: int32
```

```
In [27]: x.Over18=le.fit_transform(x.Over18)
          x["Over18"]
```

```
Out[27]: 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: int32
```

```
In [28]: x.OverTime=le.fit_transform(x.OverTime)
          x["OverTime"]
```

```
Out[28]: 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: int32
```

```
In [29]: y["Attrition"]=le.fit_transform(y.Attrition)
          y["Attrition"]
```

```
Out[29]: 0      1
          1      0
          2      1
          3      0
          4      0
          ..
        1465     0
        1466     0
        1467     0
        1468     0
        1469     0
        Name: Attrition, Length: 1470, dtype: int32
```

## Feature scaling

```
In [30]: from sklearn.preprocessing import MinMaxScaler
          ms=MinMaxScaler()
          x_scaled=pd.DataFrame(ms.fit_transform(x),columns=x.columns)
```

```
In [31]: x_scaled
```

Out[31]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	
0	0.547619	1.0	0.715820	1.0	0.000000	0.25	0.2	0.0	0.000000	0.333333	
1	0.738095	0.5	0.126700	0.5	0.250000	0.00	0.2	0.0	0.000484	0.666667	
2	0.452381	1.0	0.909807	0.5	0.035714	0.25	0.8	0.0	0.001451	1.000000	
3	0.357143	0.5	0.923407	0.5	0.071429	0.75	0.2	0.0	0.001935	1.000000	
4	0.214286	1.0	0.350036	0.5	0.035714	0.00	0.6	0.0	0.002903	0.000000	
...	...	...	...	...	...	...	...	...	...	...	
1465	0.428571	0.5	0.559771	0.5	0.785714	0.25	0.6	0.0	0.996613	0.666667	
1466	0.500000	1.0	0.365784	0.5	0.178571	0.00	0.6	0.0	0.997097	1.000000	
1467	0.214286	1.0	0.037938	0.5	0.107143	0.50	0.2	0.0	0.998065	0.333333	
1468	0.738095	0.5	0.659270	1.0	0.035714	0.50	0.6	0.0	0.998549	1.000000	
1469	0.380952	1.0	0.376521	0.5	0.250000	0.50	0.6	0.0	1.000000	0.333333	

1470 rows × 34 columns

# Splitting Data into Train and Test.

In [32]:

from sklearn.model\_selection import train\_test\_split  
x\_train,x\_test,y\_train,y\_test=train\_test\_split(x\_scaled,y,test\_size=0.2,random\_state=0)

In [33]:

x\_train.shape,x\_test.shape,y\_train.shape,y\_test.shape

Out[33]:

((1176, 34), (294, 34), (1176, 1), (294, 1))

In [34]:

x\_train.head()

Out[34]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	
1374	0.952381	1.0	0.360057	1.0	0.714286	0.50	0.2	0.0	0.937107	1.000000	
1092	0.642857	1.0	0.607015	0.5	0.964286	0.50	1.0	0.0	0.747460	1.000000	
768	0.523810	1.0	0.141732	1.0	0.892857	0.50	0.4	0.0	0.515239	0.666667	
569	0.428571	0.0	0.953472	1.0	0.250000	0.75	0.2	0.0	0.381229	0.000000	
911	0.166667	0.5	0.355762	1.0	0.821429	0.00	0.2	0.0	0.615385	0.666667	

5 rows × 34 columns

# Model Building

## Import the model building Libraries

In [35]:

from sklearn.linear\_model import LogisticRegression  
model=LogisticRegression()

## Initializing the model

In [36]:

model.fit(x\_train,y\_train)

C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\utils\validation.py:1184: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().  
y = column\_or\_1d(y, warn=True)

Out[36]:

▼ LogisticRegression  
LogisticRegression()

# Training and testing the model

In [37]:

pred=model.predict(x\_test)

In [38]:

pred

```
Out[38]: array([0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
        1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 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, 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, 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, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 1, 0, 0])
```

```
In [39]: y_test
```

Out[39]:

	Attrition
442	0
1091	0
981	1
785	0
1332	1
...	...
1439	0
481	0
124	1
198	0
1229	0

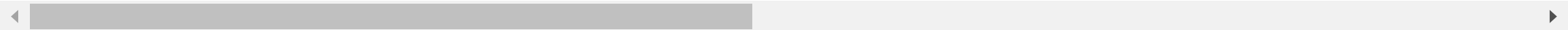
294 rows × 1 columns

```
In [40]: df
```

Out[40]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	RelationshipSati
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...	
...	...	...	...	...	...	...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	2061	...	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	2062	...	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	2064	...	
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	2065	...	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	2068	...	

1470 rows × 35 columns



# Evaluation of classification model

```
In [41]: from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,roc_auc_score,roc_curve
```

```
In [42]: accuracy_score(y_test,pred)
```

Out[42]: 0.8843537414965986

```
In [43]: confusion_matrix(y_test,pred)
```

Out[43]: array([[242, 3],  
[ 31, 18]], dtype=int64)

```
In [45]: print(classification_report(y_test,pred))
```

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

## Roc-AUC curve

```
In [46]: probability=model.predict_proba(x_test)
probability
```



```
Out[46]: array([[0.83999873, 0.16000127],
 [0.79399333, 0.20600667],
 [0.68467616, 0.31532384],
 [0.90757114, 0.09242886],
 [0.36332449, 0.63667551],
 [0.93846939, 0.06153061],
 [0.38180568, 0.61819432],
 [0.9242913 , 0.0757087 ],
 [0.99158628, 0.00841372],
 [0.6087931 , 0.3912069 ],
 [0.94601561, 0.05398439],
 [0.66706877, 0.33293123],
 [0.97979302, 0.02020698],
 [0.32784517, 0.67215483],
 [0.80213453, 0.19786547],
 [0.96545098, 0.03454902],
 [0.88956019, 0.11043981],
 [0.82898297, 0.17101703],
 [0.95522223, 0.04477777],
 [0.77216386, 0.22783614],
 [0.7664982 , 0.2335018 ],
 [0.98446095, 0.01553905],
 [0.93535508, 0.06464492],
 [0.94970044, 0.05029956],
 [0.41207587, 0.58792413],
 [0.55150536, 0.44849464],
 [0.92587286, 0.07412714],
 [0.95539065, 0.04460935],
 [0.32333368, 0.67666632],
 [0.9415617 , 0.0584383 ],
 [0.98400974, 0.01599026],
 [0.96478902, 0.03521098],
 [0.93036915, 0.06963085],
 [0.82602538, 0.17397462],
 [0.92169143, 0.07830857],
 [0.95711968, 0.04288032],
 [0.91849576, 0.08150424],
 [0.92893658, 0.07106342],
 [0.96377863, 0.03622137],
 [0.94776035, 0.05223965],
 [0.95137902, 0.04862098],
 [0.97908503, 0.02091497],
 [0.98180639, 0.01819361],
 [0.98637533, 0.01362467],
 [0.97126003, 0.02873997],
 [0.49763031, 0.50236969],
 [0.58446782, 0.41553218],
 [0.99693126, 0.00306874],
 [0.26023588, 0.73976412],
 [0.48617618, 0.51382382],
 [0.90362787, 0.09637213],
 [0.51154484, 0.48845516],
 [0.91963772, 0.08036228],
 [0.74242757, 0.25757243],
 [0.33483228, 0.66516772],
 [0.73691973, 0.26308027],
 [0.98035142, 0.01964858],
 [0.69801503, 0.30198497],
 [0.97080054, 0.02919946],
 [0.83961036, 0.16038964],
 [0.97897253, 0.02102747],
 [0.78329768, 0.21670232],
 [0.86018432, 0.13981568],
 [0.9641684 , 0.0358316 ],
 [0.62791597, 0.37208403],
 [0.96997683, 0.03002317],
 [0.70908814, 0.29091186],
 [0.83958858, 0.16041142],
 [0.89562503, 0.10437497],
 [0.91304823, 0.08695177],
 [0.91782411, 0.08217589],
 [0.69015482, 0.30984518],
 [0.91468638, 0.08531362],
 [0.92579311, 0.07420689],
 [0.87731349, 0.12268651],
 [0.93807448, 0.06192552],
 [0.95359096, 0.04640904],
 [0.92375288, 0.07624712],
 [0.80261517, 0.19738483],
 [0.96763684, 0.03236316],
 [0.99115561, 0.00884439],
 [0.9755892 , 0.0244108 ],
 [0.86364197, 0.13635803],
 [0.9739896 , 0.0260104 ],
 [0.96658992, 0.03341008],
 [0.91813112, 0.08186888],
 [0.99500603, 0.00499397],
 [0.96525148, 0.03474852],
 [0.96141973, 0.03858027],
 [0.85397306, 0.14602694],
 [0.73832335, 0.26167665],
 [0.83332643, 0.16667357],
 [0.72599891, 0.27400109],
 [0.75840435, 0.24159565],
 [0.97839579, 0.02160421],
```

[0.82251394, 0.17748606],  
[0.65923922, 0.34076078],  
[0.71977518, 0.28022482],  
[0.93085874, 0.06914126],  
[0.94996194, 0.05003806],  
[0.75562239, 0.24437761],  
[0.25301729, 0.74698271],  
[0.64561433, 0.35438567],  
[0.98079373, 0.01920627],  
[0.91221155, 0.08778845],  
[0.96744153, 0.03255847],  
[0.94538649, 0.05461351],  
[0.84876749, 0.15123251],  
[0.93156298, 0.06843702],  
[0.86247363, 0.13752637],  
[0.90415612, 0.09584388],  
[0.95330118, 0.04669882],  
[0.97506909, 0.02493091],  
[0.84616829, 0.15383171],  
[0.92918741, 0.07081259],  
[0.96910704, 0.03089296],  
[0.9462333 , 0.0537667 ],  
[0.88445684, 0.11554316],  
[0.99118384, 0.00881616],  
[0.98736729, 0.01263271],  
[0.82447747, 0.17552253],  
[0.94954766, 0.05045234],  
[0.91176762, 0.08823238],  
[0.17004243, 0.82995757],  
[0.96982244, 0.03017756],  
[0.9763181 , 0.0236819 ],  
[0.9912988 , 0.0087012 ],  
[0.8650411 , 0.1349589 ],  
[0.83525199, 0.16474801],  
[0.94797387, 0.05202613],  
[0.98475451, 0.01524549],  
[0.70721917, 0.29278083],  
[0.45232552, 0.54767448],  
[0.65724552, 0.34275448],  
[0.95370459, 0.04629541],  
[0.61033656, 0.38966344],  
[0.38666634, 0.61333366],  
[0.85447633, 0.14552367],  
[0.92597634, 0.07402366],  
[0.75856529, 0.24143471],  
[0.90581582, 0.09418418],  
[0.9310931 , 0.0689069 ],  
[0.89938044, 0.10061956],  
[0.80653673, 0.19346327],  
[0.79973707, 0.20026293],  
[0.96995061, 0.03004939],  
[0.85099576, 0.14900424],  
[0.99651154, 0.00348846],  
[0.88774851, 0.11225149],  
[0.84156845, 0.15843155],  
[0.93952427, 0.06047573],  
[0.81398118, 0.18601882],  
[0.93914131, 0.06085869],  
[0.87778683, 0.12221317],  
[0.96719816, 0.03280184],  
[0.97261201, 0.02738799],  
[0.93643575, 0.06356425],  
[0.91697618, 0.08302382],  
[0.98458284, 0.01541716],  
[0.985335 , 0.014665 ],  
[0.61482178, 0.38517822],  
[0.98735769, 0.01264231],  
[0.85038026, 0.14961974],  
[0.19491213, 0.80508787],  
[0.88401339, 0.11598661],  
[0.7157189 , 0.2842811 ],  
[0.82979857, 0.17020143],  
[0.8469417 , 0.1530583 ],  
[0.97235847, 0.02764153],  
[0.99386774, 0.00613226],  
[0.95808368, 0.04191632],  
[0.90217607, 0.09782393],  
[0.88448583, 0.11551417],  
[0.89622018, 0.10377982],  
[0.98220687, 0.01779313],  
[0.85628685, 0.14371315],  
[0.89384565, 0.10615435],  
[0.89701037, 0.10298963],  
[0.94867379, 0.05132621],  
[0.90938919, 0.09061081],  
[0.97102617, 0.02897383],  
[0.90075913, 0.09924087],  
[0.99487968, 0.00512032],  
[0.24891577, 0.75108423],  
[0.95703032, 0.04296968],  
[0.95937866, 0.04062134],  
[0.62481028, 0.37518972],  
[0.95436872, 0.04563128],  
[0.2748184 , 0.7251816 ],  
[0.89328335, 0.10671665],

[0.63050914, 0.36949086],  
[0.61853059, 0.38146941],  
[0.67904507, 0.32095493],  
[0.94733198, 0.05266802],  
[0.91827996, 0.08172004],  
[0.86052167, 0.13947833],  
[0.95665683, 0.04334317],  
[0.98530407, 0.01469593],  
[0.73586012, 0.26413988],  
[0.93669034, 0.06330966],  
[0.8385253 , 0.1614747 ],  
[0.84619483, 0.15380517],  
[0.32847643, 0.67152357],  
[0.94159207, 0.05840793],  
[0.72108177, 0.27891823],  
[0.95487436, 0.04512564],  
[0.53966135, 0.46033865],  
[0.99651569, 0.00348431],  
[0.85931033, 0.14068967],  
[0.97252599, 0.02747401],  
[0.87285867, 0.12714133],  
[0.82715754, 0.17284246],  
[0.92658934, 0.07341066],  
[0.89900173, 0.10099827],  
[0.83129115, 0.16870885],  
[0.97439158, 0.02560842],  
[0.98175969, 0.01824031],  
[0.91329204, 0.08670796],  
[0.97165763, 0.02834237],  
[0.86289785, 0.13710215],  
[0.91221065, 0.08778935],  
[0.7799939 , 0.2200061 ],  
[0.26598852, 0.73401148],  
[0.84061022, 0.15938978],  
[0.5904551 , 0.4095449 ],  
[0.98486155, 0.01513845],  
[0.88693691, 0.11306309],  
[0.78502494, 0.21497506],  
[0.67662425, 0.32337575],  
[0.96590734, 0.03409266],  
[0.95743682, 0.04256318],  
[0.67842469, 0.32157531],  
[0.94545535, 0.05454465],  
[0.97651521, 0.02348479],  
[0.83576648, 0.16423352],  
[0.67303853, 0.32696147],  
[0.77107937, 0.22892063],  
[0.99122841, 0.00877159],  
[0.91801181, 0.08198819],  
[0.98843639, 0.01156361],  
[0.8591309 , 0.1408691 ],  
[0.70764853, 0.29235147],  
[0.98729695, 0.01270305],  
[0.82670084, 0.17329916],  
[0.95918609, 0.04081391],  
[0.95905835, 0.04094165],  
[0.57228575, 0.42771425],  
[0.65041714, 0.34958286],  
[0.96233228, 0.03766772],  
[0.87974714, 0.12025286],  
[0.62301077, 0.37698923],  
[0.6807371 , 0.3192629 ],  
[0.20440662, 0.79559338],  
[0.94614341, 0.05385659],  
[0.78402963, 0.21597037],  
[0.93616272, 0.06383728],  
[0.99429009, 0.00570991],  
[0.33981813, 0.66018187],  
[0.64144714, 0.35855286],  
[0.62216394, 0.37783606],  
[0.63218602, 0.36781398],  
[0.96445488, 0.03554512],  
[0.78281797, 0.21718203],  
[0.94056378, 0.05943622],  
[0.93445515, 0.06554485],  
[0.89918525, 0.10081475],  
[0.99181287, 0.00818713],  
[0.73408684, 0.26591316],  
[0.57190325, 0.42809675],  
[0.93457165, 0.06542835],  
[0.90703197, 0.09296803],  
[0.98740174, 0.01259826],  
[0.85773349, 0.14226651],  
[0.94927338, 0.05072662],  
[0.97627742, 0.02372258],  
[0.97413077, 0.02586923],  
[0.93239573, 0.06760427],  
[0.75684352, 0.24315648],  
[0.73038568, 0.26961432],  
[0.80168267, 0.19831733],  
[0.7347704 , 0.2652296 ],  
[0.9834077 , 0.0165923 ],  
[0.84215764, 0.15784236],  
[0.91601018, 0.08398982],  
[0.97288225, 0.02711775],

```
[0.81249453, 0.18750547],  
[0.99216465, 0.00783535],  
[0.7155761 , 0.2844239 ],  
[0.99729258, 0.00270742],  
[0.97515031, 0.02484969],  
[0.77414255, 0.22585745],  
[0.27224395, 0.72775605],  
[0.92308453, 0.07691547],  
[0.73695641, 0.26304359]])
```

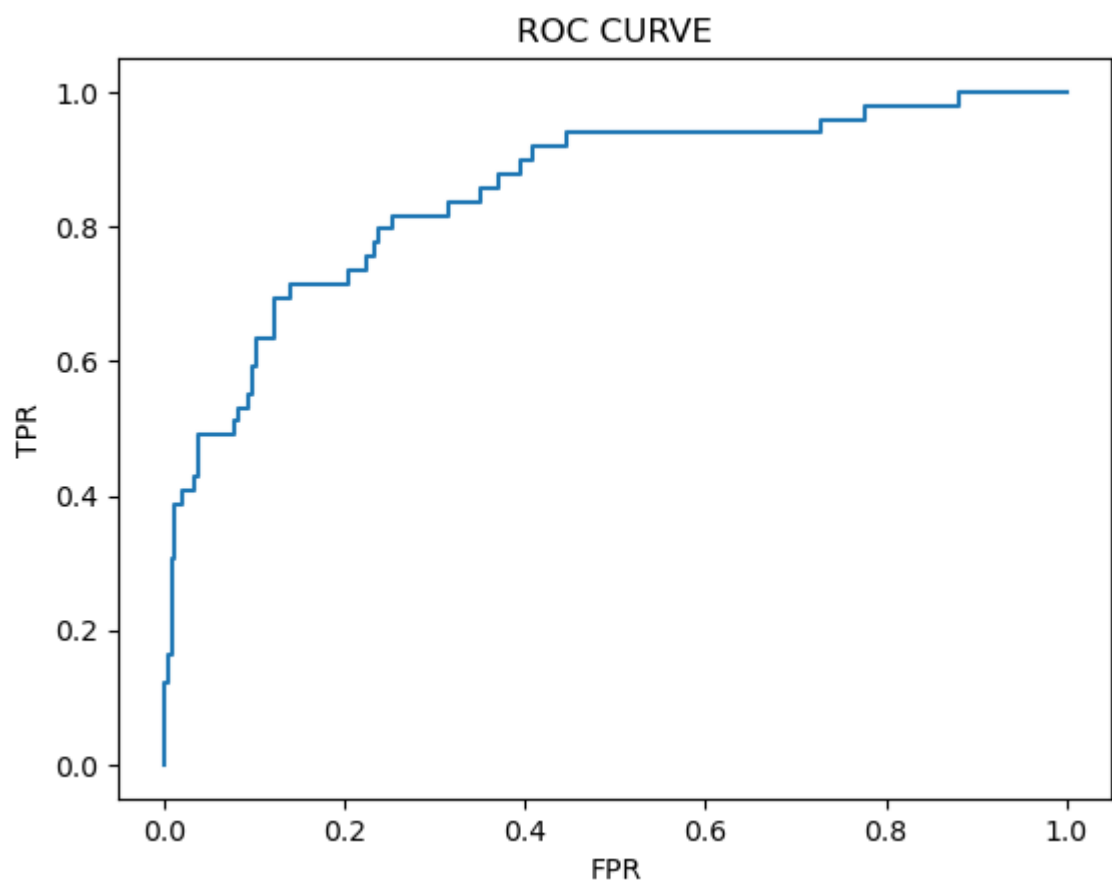
```
In [47]: probability=model.predict_proba(x_test)[: ,1]
```

```
In [48]: probability
```

```
Out[48]: array([0.16000127, 0.20600667, 0.31532384, 0.09242886, 0.63667551,  
0.06153061, 0.61819432, 0.0757087 , 0.00841372, 0.3912069 ,  
0.05398439, 0.33293123, 0.02020698, 0.67215483, 0.19786547,  
0.03454902, 0.11043981, 0.17101703, 0.04477777, 0.22783614,  
0.2335018 , 0.01553905, 0.06464492, 0.05029956, 0.58792413,  
0.44849464, 0.07412714, 0.04460935, 0.67666632, 0.0584383 ,  
0.01599026, 0.03521098, 0.06963085, 0.17397462, 0.07830857,  
0.04288032, 0.08150424, 0.07106342, 0.03622137, 0.05223965,  
0.04862098, 0.02091497, 0.01819361, 0.01362467, 0.02873997,  
0.50236969, 0.41553218, 0.00306874, 0.73976412, 0.51382382,  
0.09637213, 0.48845516, 0.08036228, 0.25757243, 0.66516772,  
0.26308027, 0.01964858, 0.30198497, 0.02919946, 0.16038964,  
0.02102747, 0.21670232, 0.13981568, 0.0358316 , 0.37208403,  
0.03002317, 0.29091186, 0.16041142, 0.10437497, 0.08695177,  
0.08217589, 0.30984518, 0.08531362, 0.07420689, 0.12268651,  
0.06192552, 0.04640904, 0.07624712, 0.19738483, 0.03236316,  
0.00884439, 0.0244108 , 0.13635803, 0.0260104 , 0.03341008,  
0.08186888, 0.00499397, 0.03474852, 0.03858027, 0.14602694,  
0.26167665, 0.16667357, 0.27400109, 0.24159565, 0.02160421,  
0.17748606, 0.34076078, 0.28022482, 0.06914126, 0.05003806,  
0.24437761, 0.74698271, 0.35438567, 0.01920627, 0.08778845,  
0.03255847, 0.05461351, 0.15123251, 0.06843702, 0.13752637,  
0.09584388, 0.04669882, 0.02493091, 0.15383171, 0.07081259,  
0.03089296, 0.0537667 , 0.11554316, 0.00881616, 0.01263271,  
0.17552253, 0.05045234, 0.08823238, 0.82995757, 0.03017756,  
0.0236819 , 0.0087012 , 0.1349589 , 0.16474801, 0.05202613,  
0.01524549, 0.29278083, 0.54767448, 0.34275448, 0.04629541,  
0.38966344, 0.61333366, 0.14552367, 0.07402366, 0.24143471,  
0.09418418, 0.0689069 , 0.10061956, 0.19346327, 0.20026293,  
0.03004939, 0.14900424, 0.00348846, 0.11225149, 0.15843155,  
0.06047573, 0.18601882, 0.06085869, 0.12221317, 0.03280184,  
0.02738799, 0.06356425, 0.08302382, 0.01541716, 0.014665 ,  
0.38517822, 0.01264231, 0.14961974, 0.80508787, 0.11598661,  
0.2842811 , 0.17020143, 0.1530583 , 0.02764153, 0.00613226,  
0.04191632, 0.09782393, 0.11551417, 0.10377982, 0.01779313,  
0.14371315, 0.10615435, 0.10298963, 0.05132621, 0.09061081,  
0.02897383, 0.09924087, 0.00512032, 0.75108423, 0.04296968,  
0.04062134, 0.37518972, 0.04563128, 0.7251816 , 0.10671665,  
0.36949086, 0.38146941, 0.32095493, 0.05266802, 0.08172004,  
0.13947833, 0.04334317, 0.01469593, 0.26413988, 0.06330966,  
0.1614747 , 0.15380517, 0.67152357, 0.05840793, 0.27891823,  
0.04512564, 0.46033865, 0.00348431, 0.14068967, 0.02747401,  
0.12714133, 0.17284246, 0.07341066, 0.10099827, 0.16870885,  
0.02560842, 0.01824031, 0.08670796, 0.02834237, 0.13710215,  
0.08778935, 0.2200061 , 0.73401148, 0.15938978, 0.4095449 ,  
0.01513845, 0.11306309, 0.21497506, 0.32337575, 0.03409266,  
0.04256318, 0.32157531, 0.05454465, 0.02348479, 0.16423352,  
0.32696147, 0.22892063, 0.00877159, 0.08198819, 0.01156361,  
0.1408691 , 0.29235147, 0.01270305, 0.17329916, 0.04081391,  
0.04094165, 0.42771425, 0.34958286, 0.03766772, 0.12025286,  
0.37698923, 0.3192629 , 0.79559338, 0.05385659, 0.21597037,  
0.06383728, 0.00570991, 0.66018187, 0.35855286, 0.37783606,  
0.36781398, 0.03554512, 0.21718203, 0.05943622, 0.06554485,  
0.10081475, 0.00818713, 0.26591316, 0.42809675, 0.06542835,  
0.09296803, 0.01259826, 0.14226651, 0.05072662, 0.02372258,  
0.02586923, 0.06760427, 0.24315648, 0.26961432, 0.19831733,  
0.2652296 , 0.0165923 , 0.15784236, 0.08398982, 0.02711775,  
0.18750547, 0.00783535, 0.2844239 , 0.00270742, 0.02484969,  
0.22585745, 0.72775605, 0.07691547, 0.26304359])
```

```
In [49]: # roc_curve  
fpr,tpr,threshholds = roc_curve(y_test,probability)
```

```
In [50]: plt.plot(fpr,tpr)  
plt.xlabel('FPR')  
plt.ylabel('TPR')  
plt.title('ROC CURVE')  
plt.show()
```



# Decision Tree

## Import the model building Libraries

```
In [51]: from sklearn.tree import DecisionTreeClassifier
dtc=DecisionTreeClassifier()
```

```
In [52]: dtc.fit(x_train,y_train)
```

```
Out[52]: ▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
In [53]: pred=dtc.predict(x_test)
```

```
In [54]: pred
```

```
Out[54]: 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, 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, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,
0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0,
1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1,
1, 1, 0, 0, 0, 0, 1, 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, 1,
0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
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, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0, 0])
```

```
In [55]: y_test
```

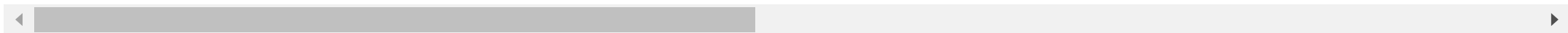
```
Out[55]:
```

	Attrition
442	0
1091	0
981	1
785	0
1332	1
...	...
1439	0
481	0
124	1
198	0
1229	0

294 rows × 1 columns

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	RelationshipSa
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...	
...	...	...	...	...	...	...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	2061	...	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	2062	...	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	2064	...	
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	2065	...	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	2068	...	

1470 rows × 35 columns



## Evaluation of model

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

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

0.7687074829931972

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

```
array([[210, 35],
       [ 33, 16]], dtype=int64)
```

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

	precision	recall	f1-score	support
0	0.86	0.86	0.86	245
1	0.31	0.33	0.32	49
accuracy			0.77	294
macro avg	0.59	0.59	0.59	294
weighted avg	0.77	0.77	0.77	294

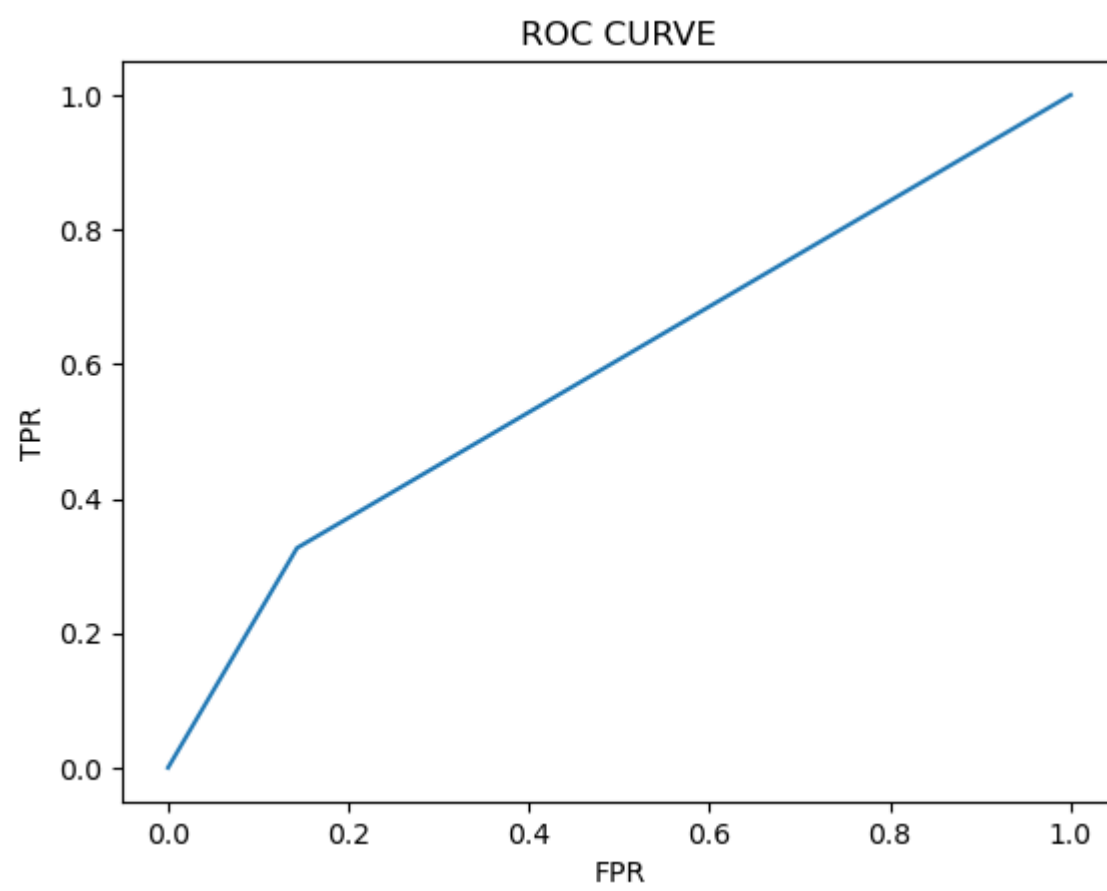
```
probability=dtc.predict_proba(x_test)[:,-1]
```

probability

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

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

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



## Hyper parameter tuning

```
In [67]: from sklearn import tree
plt.figure(figsize=(25,15))
tree.plot_tree(dtc,filled=True)
```

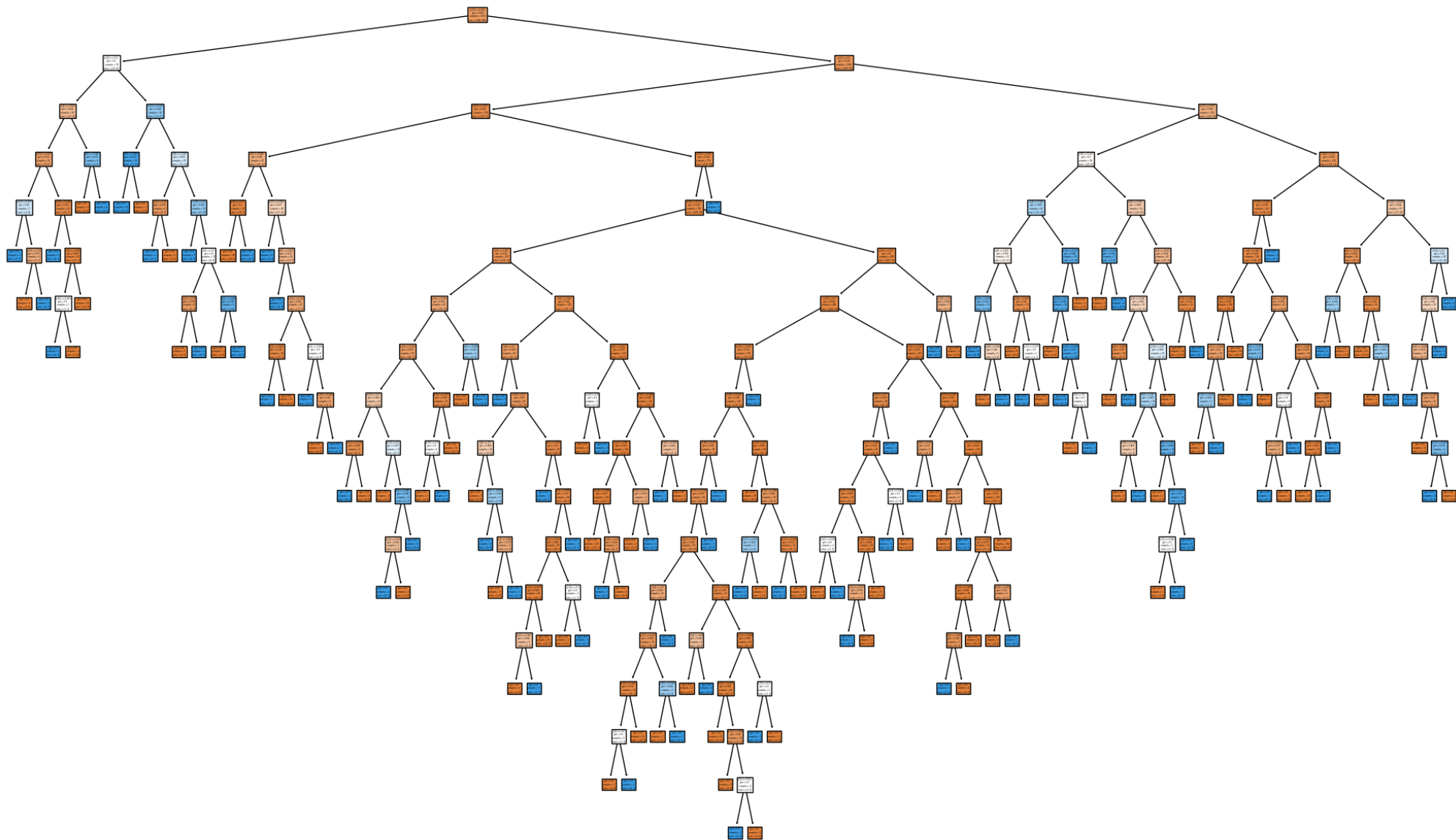


Out[67]: [Text(0.32534693238731216, 0.9722222222222222, 'x[27] <= 0.038\ngini = 0.269\nsamples = 1176\nvalue = [988, 188]'),  
Text(0.07345575959933222, 0.9166666666666666, 'x[16] <= 0.75\ngini = 0.5\nsamples = 78\nvalue = [39, 39]'),  
Text(0.04340567612687813, 0.8611111111111112, 'x[4] <= 0.554\ngini = 0.426\nsamples = 39\nvalue = [27, 12]'),  
Text(0.02671118530884808, 0.8055555555555556, 'x[15] <= 0.167\ngini = 0.312\nsamples = 31\nvalue = [25, 6]'),  
Text(0.01335559265442404, 0.75, 'x[17] <= 0.057\ngini = 0.49\nsamples = 7\nvalue = [3, 4]'),  
Text(0.00667779632721202, 0.6944444444444444, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),  
Text(0.02003338898163606, 0.6944444444444444, 'x[16] <= 0.25\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),  
Text(0.01335559265442404, 0.6388888888888888, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),  
Text(0.02671118530884808, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.04006677796327212, 0.75, 'x[19] <= 0.056\ngini = 0.153\nsamples = 24\nvalue = [22, 2]'),  
Text(0.0333889816360601, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.04674457429048414, 0.6944444444444444, 'x[9] <= 0.167\ngini = 0.083\nsamples = 23\nvalue = [22, 1]'),  
Text(0.04006677796327212, 0.6388888888888888, 'x[18] <= 0.283\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),  
Text(0.0333889816360601, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.04674457429048414, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.05342237061769616, 0.6388888888888888, 'gini = 0.0\nsamples = 21\nvalue = [21, 0]'),  
Text(0.06010016694490818, 0.8055555555555556, 'x[8] <= 0.385\ngini = 0.375\nsamples = 8\nvalue = [2, 6]'),  
Text(0.05342237061769616, 0.75, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),  
Text(0.0667779632721202, 0.75, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),  
Text(0.10350584307178631, 0.8611111111111112, 'x[11] <= 0.364\ngini = 0.426\nsamples = 39\nvalue = [12, 27]'),  
Text(0.08681135225375626, 0.8055555555555556, 'x[0] <= 0.369\ngini = 0.133\nsamples = 14\nvalue = [1, 13]'),  
Text(0.08013355592654424, 0.75, 'gini = 0.0\nsamples = 13\nvalue = [0, 13]'),  
Text(0.09348914858096828, 0.75, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.12020033388981637, 0.8055555555555556, 'x[8] <= 0.105\ngini = 0.493\nsamples = 25\nvalue = [11, 14]'),  
Text(0.10684474123539232, 0.75, 'x[1] <= 0.75\ngini = 0.278\nsamples = 6\nvalue = [5, 1]'),  
Text(0.1001669449081803, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.11352253756260434, 0.6944444444444444, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),  
Text(0.1335559265442404, 0.75, 'x[15] <= 0.5\ngini = 0.432\nsamples = 19\nvalue = [6, 13]'),  
Text(0.12687813021702837, 0.6944444444444444, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]'),  
Text(0.14023372287145242, 0.6944444444444444, 'x[6] <= 0.4\ngini = 0.5\nsamples = 12\nvalue = [6, 6]'),  
Text(0.12687813021702837, 0.6388888888888888, 'x[3] <= 0.75\ngini = 0.278\nsamples = 6\nvalue = [5, 1]'),  
Text(0.12020033388981637, 0.5833333333333334, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),  
Text(0.1335559265442404, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.15358931552587646, 0.6388888888888888, 'x[8] <= 0.249\ngini = 0.278\nsamples = 6\nvalue = [1, 5]'),  
Text(0.14691151919866444, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.16026711185308848, 0.5833333333333334, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),  
Text(0.5772381051752922, 0.9166666666666666, 'x[21] <= 0.5\ngini = 0.235\nsamples = 1098\nvalue = [949, 149]'),  
Text(0.32726419031719534, 0.8611111111111112, 'x[29] <= 0.167\ngini = 0.162\nsamples = 798\nvalue = [727, 71]'),  
Text(0.17362270450751252, 0.8055555555555556, 'x[8] <= 0.445\ngini = 0.38\nsamples = 47\nvalue = [35, 12]'),  
Text(0.16026711185308848, 0.75, 'x[16] <= 0.75\ngini = 0.1\nsamples = 19\nvalue = [18, 1]'),  
Text(0.15358931552587646, 0.6944444444444444, 'gini = 0.0\nsamples = 18\nvalue = [18, 0]'),  
Text(0.1669449081803005, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.18697829716193656, 0.75, 'x[17] <= 0.094\ngini = 0.477\nsamples = 28\nvalue = [17, 11]'),  
Text(0.18030050083472454, 0.6944444444444444, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),  
Text(0.19365609348914858, 0.6944444444444444, 'x[8] <= 0.524\ngini = 0.413\nsamples = 24\nvalue = [17, 7]'),  
Text(0.18697829716193656, 0.6388888888888888, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),  
Text(0.2003338898163606, 0.6388888888888888, 'x[33] <= 0.324\ngini = 0.351\nsamples = 22\nvalue = [17, 5]'),  
Text(0.18697829716193656, 0.5833333333333334, 'x[2] <= 0.025\ngini = 0.133\nsamples = 14\nvalue = [13, 1]'),  
Text(0.18030050083472454, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.19365609348914858, 0.5277777777777778, 'gini = 0.0\nsamples = 13\nvalue = [13, 0]'),  
Text(0.21368948247078465, 0.5833333333333334, 'x[2] <= 0.329\ngini = 0.5\nsamples = 8\nvalue = [4, 4]'),  
Text(0.20701168614357263, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),  
Text(0.22036727879799667, 0.5277777777777778, 'x[19] <= 0.444\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),  
Text(0.21368948247078465, 0.4722222222222222, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),  
Text(0.2270450751252087, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.4809056761268781, 0.8055555555555556, 'x[30] <= 0.963\ngini = 0.145\nsamples = 751\nvalue = [692, 59]'),  
Text(0.47422787979966613, 0.75, 'x[30] <= 0.113\ngini = 0.143\nsamples = 750\nvalue = [692, 58]'),  
Text(0.3418196994991653, 0.6944444444444444, 'x[9] <= 0.167\ngini = 0.218\nsamples = 257\nvalue = [225, 32]'),  
Text(0.2988313856427379, 0.6388888888888888, 'x[33] <= 0.147\ngini = 0.355\nsamples = 65\nvalue = [50, 15]'),  
Text(0.27712854757929883, 0.5833333333333334, 'x[33] <= 0.029\ngini = 0.303\nsamples = 59\nvalue = [48, 11]'),  
Text(0.25375626043405675, 0.5277777777777778, 'x[12] <= 0.5\ngini = 0.463\nsamples = 22\nvalue = [14, 8]'),  
Text(0.24040066777963273, 0.4722222222222222, 'x[11] <= 0.179\ngini = 0.198\nsamples = 9\nvalue = [8, 1]'),  
Text(0.2337228714524207, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.24707846410684475, 0.4166666666666667, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),  
Text(0.2671118530884808, 0.4722222222222222, 'x[11] <= 0.4\ngini = 0.497\nsamples = 13\nvalue = [6, 7]'),  
Text(0.2604340567612688, 0.4166666666666667, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),  
Text(0.27378964941569284, 0.4166666666666667, 'x[4] <= 0.286\ngini = 0.346\nsamples = 9\nvalue = [2, 7]'),  
Text(0.2671118530884808, 0.3611111111111111, 'x[11] <= 0.629\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),  
Text(0.2604340567612688, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.27378964941569284, 0.3055555555555556, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),  
Text(0.28046744574290483, 0.3611111111111111, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),  
Text(0.3005008347245409, 0.5277777777777778, 'x[15] <= 0.167\ngini = 0.149\nsamples = 37\nvalue = [34, 3]'),  
Text(0.2938230383973289, 0.4722222222222222, 'x[29] <= 0.5\ngini = 0.5\nsamples = 6\nvalue = [3, 3]'),  
Text(0.2871452420701169, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),  
Text(0.3005008347245409, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),  
Text(0.3071786310517529, 0.4722222222222222, 'gini = 0.0\nsamples = 31\nvalue = [31, 0]'),  
Text(0.32053422370617696, 0.5833333333333334, 'x[8] <= 0.065\ngini = 0.444\nsamples = 6\nvalue = [2, 4]'),  
Text(0.31385642737896496, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),  
Text(0.327212020033389, 0.5277777777777778, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),  
Text(0.38480801335559267, 0.6388888888888888, 'x[0] <= 0.321\ngini = 0.161\nsamples = 192\nvalue = [175, 17]'),  
Text(0.34724540901502504, 0.5833333333333334, 'x[6] <= 0.1\ngini = 0.294\nsamples = 67\nvalue = [55, 12]'),  
Text(0.34056761268781305, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),  
Text(0.35392320534223703, 0.5277777777777778, 'x[29] <= 0.5\ngini = 0.26\nsamples = 65\nvalue = [55, 10]'),  
Text(0.330550918196995, 0.4722222222222222, 'x[6] <= 0.5\ngini = 0.469\nsamples = 16\nvalue = [10, 6]'),  
Text(0.32387312186978295, 0.4166666666666667, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),  
Text(0.337228714524207, 0.4166666666666667, 'x[9] <= 0.833\ngini = 0.444\nsamples = 9\nvalue = [3, 6]'),  
Text(0.330550918196995, 0.3611111111111111, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),  
Text(0.34390651085141904, 0.3611111111111111, 'x[2] <= 0.566\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),  
Text(0.337228714524207, 0.3055555555555556, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),  
Text(0.35058430717863104, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.3772954924874791, 0.4722222222222222, 'x[2] <= 0.037\ngini = 0.15\nsamples = 49\nvalue = [45, 4]'),  
Text(0.37061769616026713, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.38397328881469117, 0.4166666666666667, 'x[2] <= 0.938\ngini = 0.117\nsamples = 48\nvalue = [45, 3]'),  
Text(0.3772954924874791, 0.3611111111111111, 'x[5] <= 0.875\ngini = 0.081\nsamples = 47\nvalue = [45, 2]'),  
Text(0.3639398998330551, 0.3055555555555556, 'x[12] <= 0.167\ngini = 0.043\nsamples = 45\nvalue = [44, 1]'),

Text(0.3572621035058431, 0.25, 'x[32] <= 0.067\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),  
Text(0.35058430717863104, 0.19444444444444445, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),  
Text(0.3639398998330551, 0.19444444444444445, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.37061769616026713, 0.25, 'gini = 0.0\nsamples = 42\nvalue = [42, 0]'),  
Text(0.39065108514190316, 0.3055555555555556, 'x[14] <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),  
Text(0.38397328881469117, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.3973288814691152, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.39065108514190316, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.42237061769616024, 0.5833333333333334, 'x[8] <= 0.022\ngini = 0.077\nsamples = 125\nvalue = [120, 5]'),  
Text(0.4040066777963272, 0.5277777777777778, 'x[14] <= 0.5\ngini = 0.5\nsamples = 4\nvalue = [2, 2]'),  
Text(0.3973288814691152, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),  
Text(0.41068447412353926, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),  
Text(0.44073455759599334, 0.5277777777777778, 'x[18] <= 0.968\ngini = 0.048\nsamples = 121\nvalue = [118, 3]'),  
Text(0.4240400667779633, 0.4722222222222222, 'x[2] <= 0.98\ngini = 0.033\nsamples = 118\nvalue = [116, 2]'),  
Text(0.41068447412353926, 0.4166666666666667, 'x[14] <= 0.938\ngini = 0.017\nsamples = 114\nvalue = [113, 1]'),  
Text(0.4040066777963272, 0.3611111111111111, 'gini = 0.0\nsamples = 107\nvalue = [107, 0]'),  
Text(0.41736227045075125, 0.3611111111111111, 'x[16] <= 0.25\ngini = 0.245\nsamples = 7\nvalue = [6, 1]'),  
Text(0.41068447412353926, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.4240400667779633, 0.3055555555555556, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),  
Text(0.4373956594323873, 0.4166666666666667, 'x[14] <= 0.75\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),  
Text(0.4307178631051753, 0.3611111111111111, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),  
Text(0.44407345575959933, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.4574290484140234, 0.4722222222222222, 'x[26] <= 0.167\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),  
Text(0.4507512520868113, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.46410684474123537, 0.4166666666666667, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),  
Text(0.6066360601001669, 0.6944444444444444, 'x[30] <= 0.787\ngini = 0.1\nsamples = 493\nvalue = [467, 26]'),  
Text(0.5671953255425709, 0.6388888888888888, 'x[15] <= 0.5\ngini = 0.094\nsamples = 486\nvalue = [462, 24]'),  
Text(0.508347245409015, 0.5833333333333334, 'x[14] <= 0.938\ngini = 0.154\nsamples = 191\nvalue = [175, 16]'),  
Text(0.501669449081803, 0.5277777777777778, 'x[18] <= 0.481\ngini = 0.145\nsamples = 190\nvalue = [175, 15]'),  
Text(0.48414023372287146, 0.4722222222222222, 'x[18] <= 0.47\ngini = 0.221\nsamples = 95\nvalue = [83, 12]'),  
Text(0.4774624373956594, 0.4166666666666667, 'x[33] <= 0.794\ngini = 0.207\nsamples = 94\nvalue = [83, 11]'),  
Text(0.4707846410684474, 0.3611111111111111, 'x[5] <= 0.375\ngini = 0.192\nsamples = 93\nvalue = [83, 10]'),  
Text(0.44908180300500833, 0.3055555555555556, 'x[6] <= 0.9\ngini = 0.363\nsamples = 21\nvalue = [16, 5]'),  
Text(0.44240400667779634, 0.25, 'x[17] <= 0.413\ngini = 0.266\nsamples = 19\nvalue = [16, 3]'),  
Text(0.4290484140233723, 0.19444444444444445, 'x[19] <= 0.056\ngini = 0.117\nsamples = 16\nvalue = [15, 1]'),  
Text(0.42237061769616024, 0.1388888888888889, 'x[6] <= 0.4\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),  
Text(0.41569282136894825, 0.08333333333333333, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.4290484140233723, 0.08333333333333333, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.4357262103505843, 0.1388888888888889, 'gini = 0.0\nsamples = 14\nvalue = [14, 0]'),  
Text(0.4557595993322204, 0.19444444444444445, 'x[18] <= 0.006\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),  
Text(0.44908180300500833, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.46243739565943237, 0.1388888888888889, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),  
Text(0.4557595993322204, 0.25, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),  
Text(0.49248747913188645, 0.3055555555555556, 'x[31] <= 0.139\ngini = 0.129\nsamples = 72\nvalue = [67, 5]'),  
Text(0.4757929883138564, 0.25, 'x[8] <= 0.68\ngini = 0.444\nsamples = 6\nvalue = [4, 2]'),  
Text(0.4691151919866444, 0.19444444444444445, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),  
Text(0.48247078464106846, 0.19444444444444445, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),  
Text(0.5091819699499165, 0.25, 'x[11] <= 0.993\ngini = 0.087\nsamples = 66\nvalue = [63, 3]'),  
Text(0.4958263772954925, 0.19444444444444445, 'x[28] <= 0.583\ngini = 0.061\nsamples = 64\nvalue = [62, 2]'),  
Text(0.48914858096828046, 0.1388888888888889, 'gini = 0.0\nsamples = 51\nvalue = [51, 0]'),  
Text(0.5025041736227045, 0.1388888888888889, 'x[14] <= 0.812\ngini = 0.26\nsamples = 13\nvalue = [11, 2]'),  
Text(0.4958263772954925, 0.08333333333333333, 'gini = 0.0\nsamples = 9\nvalue = [9, 0]'),  
Text(0.5091819699499165, 0.08333333333333333, 'x[2] <= 0.525\ngini = 0.5\nsamples = 4\nvalue = [2, 2]'),  
Text(0.5025041736227045, 0.027777777777777776, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),  
Text(0.5158597662771286, 0.027777777777777776, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),  
Text(0.5225375626043406, 0.19444444444444445, 'x[31] <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),  
Text(0.5158597662771286, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.5292153589315526, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.48414023372287146, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.49081803005008345, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.5191986644407346, 0.4722222222222222, 'x[19] <= 0.5\ngini = 0.061\nsamples = 95\nvalue = [92, 3]'),  
Text(0.5125208681135225, 0.4166666666666667, 'gini = 0.0\nsamples = 76\nvalue = [76, 0]'),  
Text(0.5258764607679466, 0.4166666666666667, 'x[33] <= 0.088\ngini = 0.266\nsamples = 19\nvalue = [16, 3]'),  
Text(0.5125208681135225, 0.3611111111111111, 'x[17] <= 0.493\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),  
Text(0.5058430717863105, 0.3055555555555556, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),  
Text(0.5191986644407346, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.5392320534223706, 0.3611111111111111, 'x[17] <= 0.108\ngini = 0.117\nsamples = 16\nvalue = [15, 1]'),  
Text(0.5325542570951586, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.5459098497495827, 0.3055555555555556, 'gini = 0.0\nsamples = 15\nvalue = [15, 0]'),  
Text(0.5150250417362271, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.6260434056761269, 0.5833333333333334, 'x[22] <= 0.036\ngini = 0.053\nsamples = 295\nvalue = [287, 8]'),  
Text(0.6026711185308848, 0.5277777777777778, 'x[32] <= 0.7\ngini = 0.159\nsamples = 46\nvalue = [42, 4]'),  
Text(0.5959933222036727, 0.4722222222222222, 'x[27] <= 0.688\ngini = 0.124\nsamples = 45\nvalue = [42, 3]'),  
Text(0.5792988313856428, 0.4166666666666667, 'x[12] <= 0.167\ngini = 0.089\nsamples = 43\nvalue = [41, 2]'),  
Text(0.5659432387312187, 0.3611111111111111, 'x[32] <= 0.533\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),  
Text(0.5592654424040067, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.5726210350584308, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.5926544240400667, 0.3611111111111111, 'x[14] <= 0.062\ngini = 0.048\nsamples = 41\nvalue = [40, 1]'),  
Text(0.5859766277128547, 0.3055555555555556, 'x[11] <= 0.707\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),  
Text(0.5792988313856428, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.5926544240400667, 0.25, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),  
Text(0.5993322203672788, 0.3055555555555556, 'gini = 0.0\nsamples = 37\nvalue = [37, 0]'),  
Text(0.6126878130217028, 0.4166666666666667, 'x[2] <= 0.36\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),  
Text(0.6060100166944908, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.6193656093489148, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.6093489148580968, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.6494156928213689, 0.5277777777777778, 'x[17] <= 0.056\ngini = 0.032\nsamples = 249\nvalue = [245, 4]'),  
Text(0.6327212020033389, 0.4722222222222222, 'x[8] <= 0.33\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),  
Text(0.6260434056761269, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.6393989983305509, 0.4166666666666667, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),  
Text(0.666110183639399, 0.4722222222222222, 'x[2] <= 0.015\ngini = 0.024\nsamples = 244\nvalue = [241, 3]'),  
Text(0.6527545909849749, 0.4166666666666667, 'x[4] <= 0.875\ngini = 0.278\nsamples = 6\nvalue = [5, 1]'),  
Text(0.6460767946577629, 0.3611111111111111, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),  
Text(0.659432387312187, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.679465776293823, 0.4166666666666667, 'x[24] <= 0.167\ngini = 0.017\nsamples = 238\nvalue = [236, 2]'),

Text(0.672787979966611, 0.3611111111111111, 'x[29] <= 0.833\ngini = 0.073\nsamples = 53\nvalue = [51, 2]'),  
Text(0.659432387312187, 0.3055555555555556, 'x[33] <= 0.088\ngini = 0.041\nsamples = 48\nvalue = [47, 1]'),  
Text(0.6527545909849749, 0.25, 'x[0] <= 0.345\ngini = 0.245\nsamples = 7\nvalue = [6, 1]'),  
Text(0.6460767946577629, 0.1944444444444445, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.659432387312187, 0.1944444444444445, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),  
Text(0.666110183639399, 0.25, 'gini = 0.0\nsamples = 41\nvalue = [41, 0]'),  
Text(0.6861435726210351, 0.3055555555555556, 'x[32] <= 0.367\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),  
Text(0.679465776293823, 0.25, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),  
Text(0.6928213689482471, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.6861435726210351, 0.3611111111111111, 'gini = 0.0\nsamples = 185\nvalue = [185, 0]'),  
Text(0.6460767946577629, 0.6388888888888888, 'x[10] <= 0.5\ngini = 0.408\nsamples = 7\nvalue = [5, 2]'),  
Text(0.6393989983305509, 0.5833333333333334, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),  
Text(0.6527545909849749, 0.5833333333333334, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),  
Text(0.48758347245409017, 0.75, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.82721202003389, 0.8611111111111112, 'x[17] <= 0.157\ngini = 0.385\nsamples = 300\nvalue = [222, 78]'),  
Text(0.7437395659432388, 0.8055555555555556, 'x[26] <= 0.167\ngini = 0.5\nsamples = 96\nvalue = [49, 47]'),  
Text(0.7095158597662772, 0.75, 'x[4] <= 0.161\ngini = 0.459\nsamples = 42\nvalue = [15, 27]'),  
Text(0.6861435726210351, 0.6944444444444444, 'x[8] <= 0.415\ngini = 0.499\nsamples = 23\nvalue = [12, 11]'),  
Text(0.672787979966611, 0.6388888888888888, 'x[18] <= 0.561\ngini = 0.355\nsamples = 13\nvalue = [3, 10]'),  
Text(0.666110183639399, 0.5833333333333334, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),  
Text(0.679465776293823, 0.5833333333333334, 'x[28] <= 0.583\ngini = 0.48\nsamples = 5\nvalue = [3, 2]'),  
Text(0.672787979966611, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),  
Text(0.6861435726210351, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),  
Text(0.6994991652754591, 0.6388888888888888, 'x[14] <= 0.875\ngini = 0.18\nsamples = 10\nvalue = [9, 1]'),  
Text(0.6928213689482471, 0.5833333333333334, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),  
Text(0.7061769616026711, 0.5833333333333334, 'x[17] <= 0.077\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),  
Text(0.6994991652754591, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.7128547579298832, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.7328881469115192, 0.6944444444444444, 'x[13] <= 0.125\ngini = 0.266\nsamples = 19\nvalue = [3, 16]'),  
Text(0.7262103505843072, 0.6388888888888888, 'x[11] <= 0.2\ngini = 0.198\nsamples = 18\nvalue = [2, 16]'),  
Text(0.7195325542570952, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.7328881469115192, 0.5833333333333334, 'x[32] <= 0.433\ngini = 0.111\nsamples = 17\nvalue = [1, 16]'),  
Text(0.7262103505843072, 0.5277777777777778, 'gini = 0.0\nsamples = 15\nvalue = [0, 15]'),  
Text(0.7395659432387313, 0.5277777777777778, 'x[5] <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),  
Text(0.7328881469115192, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.7462437395659433, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.7395659432387313, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.7779632721202003, 0.75, 'x[0] <= 0.202\ngini = 0.466\nsamples = 54\nvalue = [34, 20]'),  
Text(0.7595993322203672, 0.6944444444444444, 'x[8] <= 0.164\ngini = 0.245\nsamples = 7\nvalue = [1, 6]'),  
Text(0.7529215358931552, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.7662771285475793, 0.6388888888888888, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),  
Text(0.7963272120200334, 0.6944444444444444, 'x[2] <= 0.622\ngini = 0.418\nsamples = 47\nvalue = [33, 14]'),  
Text(0.7796327212020033, 0.6388888888888888, 'x[2] <= 0.145\ngini = 0.482\nsamples = 32\nvalue = [19, 13]'),  
Text(0.7662771285475793, 0.5833333333333334, 'x[4] <= 0.821\ngini = 0.18\nsamples = 10\nvalue = [9, 1]'),  
Text(0.7595993322203672, 0.5277777777777778, 'gini = 0.0\nsamples = 9\nvalue = [9, 0]'),  
Text(0.7729549248747913, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.7929883138564274, 0.5833333333333334, 'x[18] <= 0.87\ngini = 0.496\nsamples = 22\nvalue = [10, 12]'),  
Text(0.7863105175292153, 0.5277777777777778, 'x[8] <= 0.41\ngini = 0.465\nsamples = 19\nvalue = [7, 12]'),  
Text(0.7729549248747913, 0.4722222222222222, 'x[18] <= 0.715\ngini = 0.469\nsamples = 8\nvalue = [5, 3]'),  
Text(0.7662771285475793, 0.4166666666666667, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),  
Text(0.7796327212020033, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),  
Text(0.7996661101836394, 0.4722222222222222, 'x[0] <= 0.25\ngini = 0.298\nsamples = 11\nvalue = [2, 9]'),  
Text(0.7929883138564274, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.8063439065108514, 0.4166666666666667, 'x[18] <= 0.202\ngini = 0.18\nsamples = 10\nvalue = [1, 9]'),  
Text(0.7996661101836394, 0.3611111111111111, 'x[11] <= 0.286\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),  
Text(0.7929883138564274, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.8063439065108514, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.8130217028380634, 0.3611111111111111, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),  
Text(0.7996661101836394, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),  
Text(0.8130217028380634, 0.6388888888888888, 'x[19] <= 0.944\ngini = 0.124\nsamples = 15\nvalue = [14, 1]'),  
Text(0.8063439065108514, 0.5833333333333334, 'gini = 0.0\nsamples = 14\nvalue = [14, 0]'),  
Text(0.8196994991652755, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.9106844741235393, 0.8055555555555556, 'x[16] <= 0.75\ngini = 0.258\nsamples = 204\nvalue = [173, 31]'),  
Text(0.8647746243739566, 0.75, 'x[17] <= 0.992\ngini = 0.138\nsamples = 147\nvalue = [136, 11]'),  
Text(0.8580968280467446, 0.6944444444444444, 'x[4] <= 0.482\ngini = 0.128\nsamples = 146\nvalue = [136, 10]'),  
Text(0.8397328881469115, 0.6388888888888888, 'x[30] <= 0.063\ngini = 0.038\nsamples = 104\nvalue = [102, 2]'),  
Text(0.8330550918196995, 0.5833333333333334, 'x[11] <= 0.193\ngini = 0.32\nsamples = 10\nvalue = [8, 2]'),  
Text(0.8263772954924875, 0.5277777777777778, 'x[14] <= 0.562\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),  
Text(0.8196994991652755, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.8330550918196995, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),  
Text(0.8397328881469115, 0.5277777777777778, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),  
Text(0.8464106844741235, 0.5833333333333334, 'gini = 0.0\nsamples = 94\nvalue = [94, 0]'),  
Text(0.8764607679465777, 0.6388888888888888, 'x[9] <= 0.167\ngini = 0.308\nsamples = 42\nvalue = [34, 8]'),  
Text(0.8597662771285476, 0.5833333333333334, 'x[2] <= 0.736\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),  
Text(0.8530884808013356, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),  
Text(0.8664440734557596, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.8931552587646077, 0.5833333333333334, 'x[0] <= 0.393\ngini = 0.229\nsamples = 38\nvalue = [33, 5]'),  
Text(0.8797996661101837, 0.5277777777777778, 'x[2] <= 0.51\ngini = 0.5\nsamples = 6\nvalue = [3, 3]'),  
Text(0.8731218697829716, 0.4722222222222222, 'x[17] <= 0.192\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),  
Text(0.8664440734557596, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.8797996661101837, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),  
Text(0.8864774624373957, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),  
Text(0.9065108514190318, 0.5277777777777778, 'x[8] <= 0.992\ngini = 0.117\nsamples = 32\nvalue = [30, 2]'),  
Text(0.8998330550918197, 0.4722222222222222, 'x[28] <= 0.917\ngini = 0.062\nsamples = 31\nvalue = [30, 1]'),  
Text(0.8931552587646077, 0.4166666666666667, 'gini = 0.0\nsamples = 30\nvalue = [30, 0]'),  
Text(0.9065108514190318, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.9131886477462438, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.8714524207011686, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),  
Text(0.9565943238731218, 0.75, 'x[14] <= 0.812\ngini = 0.456\nsamples = 57\nvalue = [37, 20]'),  
Text(0.9265442404006677, 0.6944444444444444, 'x[8] <= 0.071\ngini = 0.238\nsamples = 29\nvalue = [25, 4]'),  
Text(0.9131886477462438, 0.6388888888888888, 'x[9] <= 0.5\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),  
Text(0.9065108514190318, 0.5833333333333334, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),  
Text(0.9198664440734557, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.9398998330550918, 0.6388888888888888, 'x[32] <= 0.4\ngini = 0.142\nsamples = 26\nvalue = [24, 2]'),  
Text(0.9332220367278798, 0.5833333333333334, 'gini = 0.0\nsamples = 23\nvalue = [23, 0]'),

Text(0.9465776293823038, 0.5833333333333334, 'x[8] <= 0.279\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),  
Text(0.9398998330550918, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.9532554257095158, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),  
Text(0.986644407345576, 0.6944444444444444, 'x[32] <= 0.1\ngini = 0.49\nsamples = 28\nvalue = [12, 16]'),  
Text(0.9799666110183639, 0.6388888888888888, 'x[12] <= 0.833\ngini = 0.48\nsamples = 20\nvalue = [12, 8]'),  
Text(0.9732888146911519, 0.5833333333333334, 'x[4] <= 0.018\ngini = 0.415\nsamples = 17\nvalue = [12, 5]'),  
Text(0.9666110183639399, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),  
Text(0.9799666110183639, 0.5277777777777778, 'x[17] <= 0.365\ngini = 0.32\nsamples = 15\nvalue = [12, 3]'),  
Text(0.9732888146911519, 0.4722222222222222, 'gini = 0.0\nsamples = 11\nvalue = [11, 0]'),  
Text(0.986644407345576, 0.4722222222222222, 'x[11] <= 0.829\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),  
Text(0.9799666110183639, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),  
Text(0.993322203672788, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),  
Text(0.986644407345576, 0.5833333333333334, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),  
Text(0.993322203672788, 0.6388888888888888, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]')]



```
In [69]: 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']
}
```

```
In [77]: grid_search=GridSearchCV(estimator=dtc,param_grid=parameter,cv=5,scoring="accuracy")
```

```
In [76]: grid_search.fit(x_train,y_train)
```

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

Below are more details about the failures:
-----
100 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in wrapper
    estimator._validate_params()
  File "C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _validate_params
    validate_parameter_constraints(
  File "C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.py", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'max_features' parameter of DecisionTreeClassifier must be an int in the range [1, inf), a float in the range (0.0, 1.0], a str among {'sqrt', 'log2'} or None. Got 'auto' instead.

    warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\model_selection\_search.py:976: UserWarning: One or more of the test scores are non-fi
nite: [
      nan      nan  0.83588172  0.84013704  0.84013704  0.84013704
      nan      nan  0.84437432  0.84269023  0.84354129  0.83843491
      nan      nan  0.83246664  0.84183195  0.84014785  0.83758745
      nan      nan  0.83671114  0.84524702  0.81888208  0.83417959
      nan      nan  0.81886765  0.84353768  0.83247746  0.8290696
      nan      nan  0.84013704  0.84013704  0.84013704  0.84013704
      nan      nan  0.83758384  0.83928597  0.83758384  0.83503065
      nan      nan  0.83844212  0.83588172  0.83333934  0.83672917
      nan      nan  0.83842409  0.83842409  0.83078255  0.83502344
      nan      nan  0.8248251   0.83928958  0.8205842   0.83589614]
    warnings.warn(
```

Out[76]:

GridSearchCV

estimator: DecisionTreeClassifier

DecisionTreeClassifier

```
In [72]: grid_search.best_params_
```

```
Out[72]: {'criterion': 'gini',
          'max_depth': 5,
          'max_features': 'log2',
          'splitter': 'random'}
```

```
In [74]: dtc_cv=DecisionTreeClassifier(criterion= 'gini',
          max_depth= 5,
          max_features= 'log2',
          splitter= 'random')
```

```
In [75]: dtc_cv.fit(x_train,y_train)
```

Out[75]:

DecisionTreeClassifier

DecisionTreeClassifier(max\_depth=5, max\_features='log2', splitter='random')

```
In [78]: pred=dtc_cv.predict(x_test)
```

```
In [79]: print(classification_report(y_test,pred))
```

	precision	recall	f1-score	support
0	0.85	0.96	0.90	245
1	0.47	0.18	0.26	49
accuracy			0.83	294
macro avg	0.66	0.57	0.58	294
weighted avg	0.79	0.83	0.80	294

# Random Forest

## Import the model building Libraries

```
In [80]: from sklearn.ensemble import RandomForestClassifier
          rfc=RandomForestClassifier()
```

```
In [81]: forest_params = [{'max_depth': list(range(10, 15)), 'max_features': list(range(0,14))}]
```

```
In [82]: rfc_cv= GridSearchCV(rfc,param_grid=forest_params,cv=10,scoring="accuracy")
```

```
In [83]: rfc_cv.fit(x_train,y_train)
```



[illegible]

[illegible]



[illegible]

[illegible]



[illegible]

[illegible]



[illegible]

[illegible]



[illegible]



[illegible]

[illegible]



[illegible]

[illegible]



[illegible]

[illegible]



[illegible]



[illegible]

[illegible]



[illegible]

[illegible]

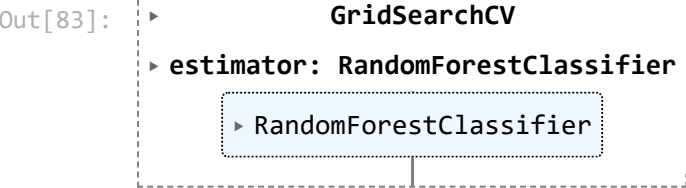


```
expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py:425: FitFailedWarning:
50 fits failed out of a total of 700.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score='raise'.
```

Below are more details about the failures:

```
-----
50 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\model_selection\_validation.py", line 732, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in wrapper
    estimator._validate_params()
  File "C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in _validate_params
    validate_parameter_constraints(
  File "C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\utils\_param_validation.py", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'max_features' parameter of RandomForestClassifier must be an int in the range
[1, inf), a float in the range (0.0, 1.0], a str among {'sqrt', 'log2'} or None. Got 0 instead.

    warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\model_selection\_search.py:976: UserWarning: One or more of the test scores are non-fi
nite: [
      nan 0.84779806 0.85374475 0.85799652 0.86311024 0.86308851
 0.85373751 0.85967695 0.8613936  0.8596842  0.85967695 0.86054614
 0.85797479 0.85712009          nan 0.84779806 0.85629436 0.85883674
 0.8622483  0.85289005 0.86221932 0.85881501 0.85882949 0.85883674
 0.8605389  0.85966971 0.8605389  0.85626539          nan 0.84779806
 0.85714907 0.86138635 0.86308851 0.86141533 0.86393597 0.85966971
 0.86137911 0.86308851 0.86138635 0.85796031 0.85625815 0.85459945
          nan 0.84608866 0.85629436 0.85884398 0.85796755 0.86136462
 0.86221932 0.85712009 0.85882225 0.85880776 0.86221208 0.85798204
 0.85795306 0.86050268          nan 0.84865276 0.85458496 0.8639577
 0.86140084 0.85628712 0.85798204 0.85796755 0.86137187 0.8596842
 0.85966247 0.85798928 0.85967695 0.85800377]
    warnings.warn(
C:\Users\lohitha\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
```



```
In [84]: pred=rfc_cv.predict(x_test)
```

```
In [85]: print(classification_report(y_test,pred))
```

	precision	recall	f1-score	support
0	0.85	1.00	0.92	245
1	1.00	0.12	0.22	49
accuracy			0.85	294
macro avg	0.93	0.56	0.57	294
weighted avg	0.88	0.85	0.80	294

```
In [86]: rfc_cv.best_params_
```

Out[86]: {'max\_depth': 14, 'max\_features': 3}

```
In [ ]:
```