ASSIGNMENT - 4

Name : R.Nikhila Manogna

Reg no : 21BCE7281

Data preprocessing on employee attrition dataset

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

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

df.head()
```

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

1   49        No  Travel_Frequently        279  Research & Development

2   37       Yes      Travel_Rarely       1373  Research & Development

3   33        No  Travel_Frequently       1392  Research & Development

4   27        No      Travel_Rarely        591  Research & Development


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

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

```
3  ...                          3            80                  0
4  ...                          4            80                  1

   TotalWorkingYears  TrainingTimesLastYear WorkLifeBalance
YearsAtCompany  \
0                  8                      0               1
6
1                 10                      3               3
10
2                  7                      3               3
0
3                  8                      3               3
8
4                  6                      3               3
2

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

[5 rows x 35 columns]
```

```
df.describe()
```

```
               Age    DailyRate  DistanceFromHome    Education
EmployeeCount  \
count  1470.000000  1470.000000       1470.000000  1470.000000
1470.0
mean     36.923810   802.485714          9.192517     2.912925
1.0
std       9.135373   403.509100          8.106864     1.024165
0.0
min      18.000000   102.000000          1.000000     1.000000
1.0
25%      30.000000   465.000000          2.000000     2.000000
1.0
50%      36.000000   802.000000          7.000000     3.000000
1.0
75%      43.000000  1157.000000         14.000000     4.000000
1.0
max      60.000000  1499.000000         29.000000     5.000000
1.0

       EmployeeNumber  EnvironmentSatisfaction   HourlyRate
JobInvolvement  \
count     1470.000000              1470.000000  1470.000000
1470.000000
```

```
mean      1024.865306                  2.721769    65.891156
2.729932
std        602.024335                  1.093082    20.329428
0.711561
min          1.000000                  1.000000    30.000000
1.000000
25%        491.250000                  2.000000    48.000000
2.000000
50%       1020.500000                  3.000000    66.000000
3.000000
75%       1555.750000                  4.000000    83.750000
3.000000
max       2068.000000                  4.000000   100.000000
4.000000

          JobLevel  ...  RelationshipSatisfaction  StandardHours  \
count  1470.000000  ...               1470.000000         1470.0
mean      2.063946  ...                  2.712245           80.0
std       1.106940  ...                  1.081209            0.0
min       1.000000  ...                  1.000000           80.0
25%       1.000000  ...                  2.000000           80.0
50%       2.000000  ...                  3.000000           80.0
75%       3.000000  ...                  4.000000           80.0
max       5.000000  ...                  4.000000           80.0

       StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
count       1470.000000        1470.000000            1470.000000
mean           0.793878          11.279592               2.799320
std            0.852077           7.780782               1.289271
min            0.000000           0.000000               0.000000
25%            0.000000           6.000000               2.000000
50%            1.000000          10.000000               3.000000
75%            1.000000          15.000000               3.000000
max            3.000000          40.000000               6.000000

       WorkLifeBalance  YearsAtCompany  YearsInCurrentRole  \
count      1470.000000     1470.000000         1470.000000
mean          2.761224        7.008163            4.229252
std           0.706476        6.126525            3.623137
min           1.000000        0.000000            0.000000
25%           2.000000        3.000000            2.000000
50%           3.000000        5.000000            3.000000
75%           3.000000        9.000000            7.000000
max           4.000000       40.000000           18.000000

       YearsSinceLastPromotion  YearsWithCurrManager
count              1470.000000           1470.000000
mean                  2.187755              4.123129
std                   3.222430              3.568136
min                   0.000000              0.000000
```

```
25%                    0.000000              2.000000
50%                    1.000000              3.000000
75%                    3.000000              7.000000
max                   15.000000             17.000000

[8 rows x 26 columns]
```

```
df.info()
```

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

```
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

```
df.shape
```

```
(1470, 35)
```

```
df.corr()
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_22072\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()
```

	Age	DailyRate	DistanceFromHome	Education
Age	1.000000	0.010661	-0.001686	0.208034
DailyRate	0.010661	1.000000	-0.004985	-0.016806
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042
Education	0.208034	-0.016806	0.021042	1.000000
EmployeeCount	NaN	NaN	NaN	NaN
EmployeeNumber	-0.010145	-0.050990	0.032916	0.042070
EnvironmentSatisfaction	0.010146	0.018355	-0.016075	-0.027128
HourlyRate	0.024287	0.023381	0.031131	0.016775
JobInvolvement	0.029820	0.046135	0.008783	0.042438
JobLevel	0.509604	0.002966	0.005303	0.101589
JobSatisfaction	-0.004892	0.030571	-0.003669	-0.011296
MonthlyIncome	0.497855	0.007707	-0.017014	0.094961
MonthlyRate	0.028051	-0.032182	0.027473	-0.026084
NumCompaniesWorked	0.299635	0.038153	-0.029251	0.126317
PercentSalaryHike	0.003634	0.022704	0.040235	-0.011111
PerformanceRating	0.001904	0.000473	0.027110	-0.024539

```
RelationshipSatisfaction  0.053535   0.007846          0.006557  -
0.009118
StandardHours                  NaN        NaN               NaN
NaN
StockOptionLevel          0.037510   0.042143          0.044872
0.018422
TotalWorkingYears         0.680381   0.014515          0.004628
0.148280
TrainingTimesLastYear    -0.019621   0.002453         -0.036942  -
0.025100
WorkLifeBalance          -0.021490  -0.037848         -0.026556
0.009819
YearsAtCompany            0.311309  -0.034055          0.009508
0.069114
YearsInCurrentRole        0.212901   0.009932          0.018845
0.060236
YearsSinceLastPromotion   0.216513  -0.033229          0.010029
0.054254
YearsWithCurrManager      0.202089  -0.026363          0.014406
0.069065

                          EmployeeCount  EmployeeNumber  \
Age                                 NaN       -0.010145
DailyRate                           NaN       -0.050990
DistanceFromHome                    NaN        0.032916
Education                           NaN        0.042070
EmployeeCount                       NaN             NaN
EmployeeNumber                      NaN        1.000000
EnvironmentSatisfaction             NaN        0.017621
HourlyRate                          NaN        0.035179
JobInvolvement                      NaN       -0.006888
JobLevel                            NaN       -0.018519
JobSatisfaction                     NaN       -0.046247
MonthlyIncome                       NaN       -0.014829
MonthlyRate                         NaN        0.012648
NumCompaniesWorked                  NaN       -0.001251
PercentSalaryHike                   NaN       -0.012944
PerformanceRating                   NaN       -0.020359
RelationshipSatisfaction            NaN       -0.069861
StandardHours                       NaN             NaN
StockOptionLevel                    NaN        0.062227
TotalWorkingYears                   NaN       -0.014365
TrainingTimesLastYear               NaN        0.023603
WorkLifeBalance                     NaN        0.010309
YearsAtCompany                      NaN       -0.011240
YearsInCurrentRole                  NaN       -0.008416
YearsSinceLastPromotion             NaN       -0.009019
YearsWithCurrManager                NaN       -0.009197
```

```
                          EnvironmentSatisfaction  HourlyRate
JobInvolvement  \
Age                                      0.010146    0.024287
0.029820
DailyRate                                0.018355    0.023381
0.046135
DistanceFromHome                        -0.016075    0.031131
0.008783
Education                               -0.027128    0.016775
0.042438
EmployeeCount                                 NaN         NaN
NaN
EmployeeNumber                           0.017621    0.035179       -
0.006888
EnvironmentSatisfaction                  1.000000   -0.049857       -
0.008278
HourlyRate                              -0.049857    1.000000
0.042861
JobInvolvement                          -0.008278    0.042861
1.000000
JobLevel                                 0.001212   -0.027853       -
0.012630
JobSatisfaction                         -0.006784   -0.071335       -
0.021476
MonthlyIncome                           -0.006259   -0.015794       -
0.015271
MonthlyRate                              0.037600   -0.015297       -
0.016322
NumCompaniesWorked                       0.012594    0.022157
0.015012
PercentSalaryHike                       -0.031701   -0.009062       -
0.017205
PerformanceRating                       -0.029548   -0.002172       -
0.029071
RelationshipSatisfaction                 0.007665    0.001330
0.034297
StandardHours                                 NaN         NaN
NaN
StockOptionLevel                         0.003432    0.050263
0.021523
TotalWorkingYears                       -0.002693   -0.002334       -
0.005533
TrainingTimesLastYear                   -0.019359   -0.008548       -
0.015338
WorkLifeBalance                          0.027627   -0.004607       -
0.014617
YearsAtCompany                           0.001458   -0.019582       -
0.021355
YearsInCurrentRole                       0.018007   -0.024106
```

```
0.008717
YearsSinceLastPromotion                  0.016194   -0.026716       -
0.024184
YearsWithCurrManager                    -0.004999   -0.020123
0.025976

                           JobLevel  ...  RelationshipSatisfaction  \
Age                        0.509604  ...                  0.053535
DailyRate                  0.002966  ...                  0.007846
DistanceFromHome           0.005303  ...                  0.006557
Education                  0.101589  ...                 -0.009118
EmployeeCount                   NaN  ...                       NaN
EmployeeNumber            -0.018519  ...                 -0.069861
EnvironmentSatisfaction    0.001212  ...                  0.007665
HourlyRate                -0.027853  ...                  0.001330
JobInvolvement            -0.012630  ...                  0.034297
JobLevel                   1.000000  ...                  0.021642
JobSatisfaction           -0.001944  ...                 -0.012454
MonthlyIncome              0.950300  ...                  0.025873
MonthlyRate                0.039563  ...                 -0.004085
NumCompaniesWorked         0.142501  ...                  0.052733
PercentSalaryHike         -0.034730  ...                 -0.040490
PerformanceRating         -0.021222  ...                 -0.031351
RelationshipSatisfaction   0.021642  ...                  1.000000
StandardHours                   NaN  ...                       NaN
StockOptionLevel           0.013984  ...                 -0.045952
TotalWorkingYears          0.782208  ...                  0.024054
TrainingTimesLastYear     -0.018191  ...                  0.002497
WorkLifeBalance            0.037818  ...                  0.019604
YearsAtCompany             0.534739  ...                  0.019367
YearsInCurrentRole         0.389447  ...                 -0.015123
YearsSinceLastPromotion    0.353885  ...                  0.033493
YearsWithCurrManager       0.375281  ...                 -0.000867

                           StandardHours  StockOptionLevel
TotalWorkingYears  \
Age                                  NaN          0.037510
0.680381
DailyRate                            NaN          0.042143
0.014515
DistanceFromHome                     NaN          0.044872
0.004628
Education                            NaN          0.018422
0.148280
EmployeeCount                        NaN               NaN
NaN
EmployeeNumber                       NaN          0.062227          -
0.014365
EnvironmentSatisfaction              NaN          0.003432          -
```

```
0.002693
HourlyRate                              NaN           0.050263          -
0.002334
JobInvolvement                          NaN           0.021523          -
0.005533
JobLevel                                NaN           0.013984
0.782208
JobSatisfaction                         NaN           0.010690          -
0.020185
MonthlyIncome                           NaN           0.005408
0.772893
MonthlyRate                             NaN          -0.034323
0.026442
NumCompaniesWorked                      NaN           0.030075
0.237639
PercentSalaryHike                       NaN           0.007528          -
0.020608
PerformanceRating                       NaN           0.003506
0.006744
RelationshipSatisfaction                NaN          -0.045952
0.024054
StandardHours                           NaN                NaN
NaN
StockOptionLevel                        NaN           1.000000
0.010136
TotalWorkingYears                       NaN           0.010136
1.000000
TrainingTimesLastYear                   NaN           0.011274          -
0.035662
WorkLifeBalance                         NaN           0.004129
0.001008
YearsAtCompany                          NaN           0.015058
0.628133
YearsInCurrentRole                      NaN           0.050818
0.460365
YearsSinceLastPromotion                 NaN           0.014352
0.404858
YearsWithCurrManager                    NaN           0.024698
0.459188

                         TrainingTimesLastYear  WorkLifeBalance  \
Age                                  -0.019621        -0.021490
DailyRate                             0.002453        -0.037848
DistanceFromHome                     -0.036942        -0.026556
Education                            -0.025100         0.009819
EmployeeCount                              NaN              NaN
EmployeeNumber                        0.023603         0.010309
EnvironmentSatisfaction              -0.019359         0.027627
HourlyRate                           -0.008548        -0.004607
```

```
JobInvolvement                -0.015338        -0.014617
JobLevel                      -0.018191         0.037818
JobSatisfaction               -0.005779        -0.019459
MonthlyIncome                 -0.021736         0.030683
MonthlyRate                    0.001467         0.007963
NumCompaniesWorked            -0.066054        -0.008366
PercentSalaryHike             -0.005221        -0.003280
PerformanceRating             -0.015579         0.002572
RelationshipSatisfaction       0.002497         0.019604
StandardHours                       NaN              NaN
StockOptionLevel               0.011274         0.004129
TotalWorkingYears             -0.035662         0.001008
TrainingTimesLastYear          1.000000         0.028072
WorkLifeBalance                0.028072         1.000000
YearsAtCompany                 0.003569         0.012089
YearsInCurrentRole            -0.005738         0.049856
YearsSinceLastPromotion       -0.002067         0.008941
YearsWithCurrManager          -0.004096         0.002759
```

```
                          YearsAtCompany  YearsInCurrentRole  \
Age                             0.311309            0.212901
DailyRate                      -0.034055            0.009932
DistanceFromHome                0.009508            0.018845
Education                       0.069114            0.060236
EmployeeCount                        NaN                 NaN
EmployeeNumber                 -0.011240           -0.008416
EnvironmentSatisfaction         0.001458            0.018007
HourlyRate                     -0.019582           -0.024106
JobInvolvement                 -0.021355            0.008717
JobLevel                        0.534739            0.389447
JobSatisfaction                -0.003803           -0.002305
MonthlyIncome                   0.514285            0.363818
MonthlyRate                    -0.023655           -0.012815
NumCompaniesWorked             -0.118421           -0.090754
PercentSalaryHike              -0.035991           -0.001520
PerformanceRating               0.003435            0.034986
RelationshipSatisfaction        0.019367           -0.015123
StandardHours                        NaN                 NaN
StockOptionLevel                0.015058            0.050818
TotalWorkingYears               0.628133            0.460365
TrainingTimesLastYear           0.003569           -0.005738
WorkLifeBalance                 0.012089            0.049856
YearsAtCompany                  1.000000            0.758754
YearsInCurrentRole              0.758754            1.000000
YearsSinceLastPromotion         0.618409            0.548056
YearsWithCurrManager            0.769212            0.714365
```

```
                          YearsSinceLastPromotion
YearsWithCurrManager
```

```
Age                              0.216513
0.202089
DailyRate                       -0.033229          -
0.026363
DistanceFromHome                 0.010029
0.014406
Education                        0.054254
0.069065
EmployeeCount                         NaN
NaN
EmployeeNumber                  -0.009019          -
0.009197
EnvironmentSatisfaction          0.016194          -
0.004999
HourlyRate                      -0.026716          -
0.020123
JobInvolvement                  -0.024184
0.025976
JobLevel                         0.353885
0.375281
JobSatisfaction                 -0.018214          -
0.027656
MonthlyIncome                    0.344978
0.344079
MonthlyRate                      0.001567          -
0.036746
NumCompaniesWorked              -0.036814          -
0.110319
PercentSalaryHike               -0.022154          -
0.011985
PerformanceRating                0.017896
0.022827
RelationshipSatisfaction         0.033493          -
0.000867
StandardHours                         NaN
NaN
StockOptionLevel                 0.014352
0.024698
TotalWorkingYears                0.404858
0.459188
TrainingTimesLastYear           -0.002067          -
0.004096
WorkLifeBalance                  0.008941
0.002759
YearsAtCompany                   0.618409
0.769212
YearsInCurrentRole               0.548056
0.714365
YearsSinceLastPromotion          1.000000
0.510224
```

```
YearsWithCurrManager                      0.510224
1.000000

[26 rows x 26 columns]
```

```
# Checking for null values
df.isnull().any()
```

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

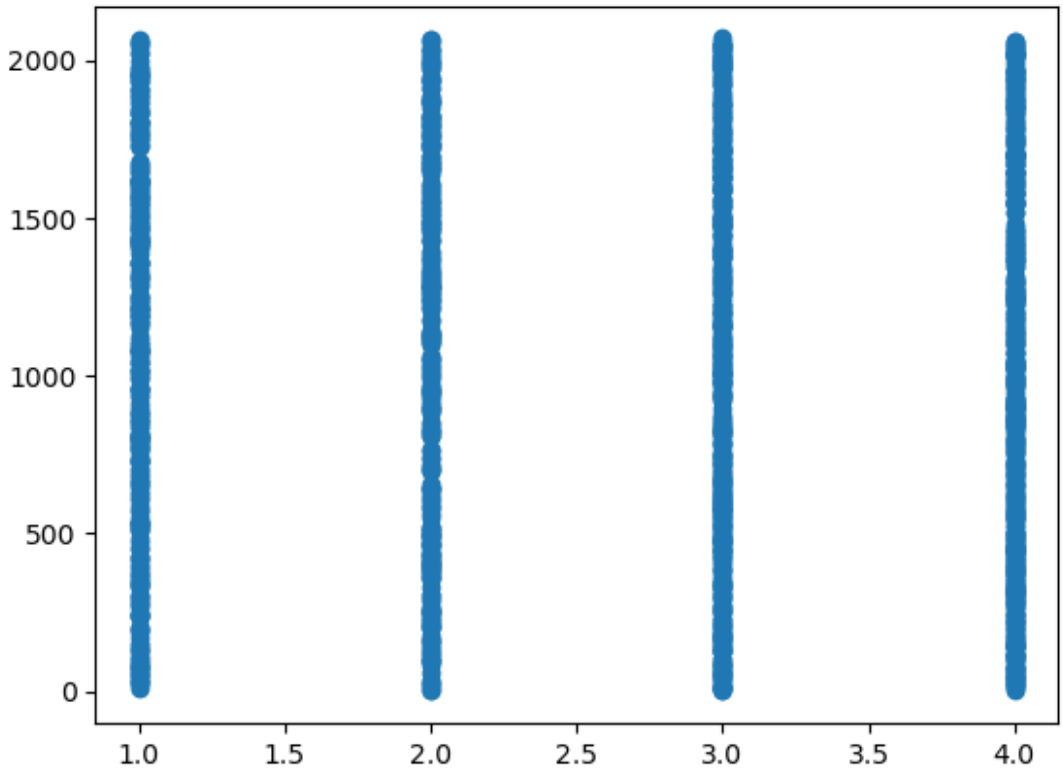
```
df.isnull().sum()
```

```
Age                         0
Attrition                   0
BusinessTravel              0
```

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

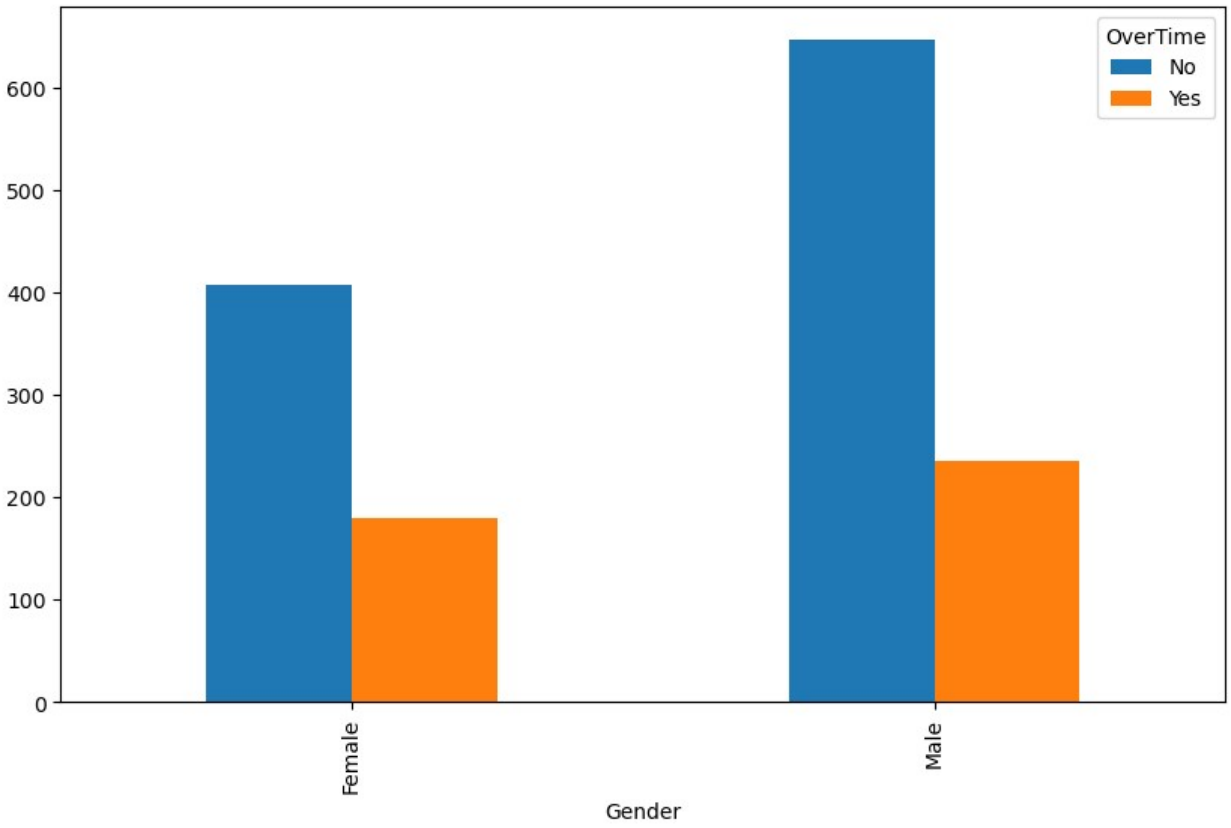
```
# Data visualisation
plt.scatter(df["JobSatisfaction"],df["EmployeeNumber"])
```

```
<matplotlib.collections.PathCollection at 0x1e0ea5a1a10>
```



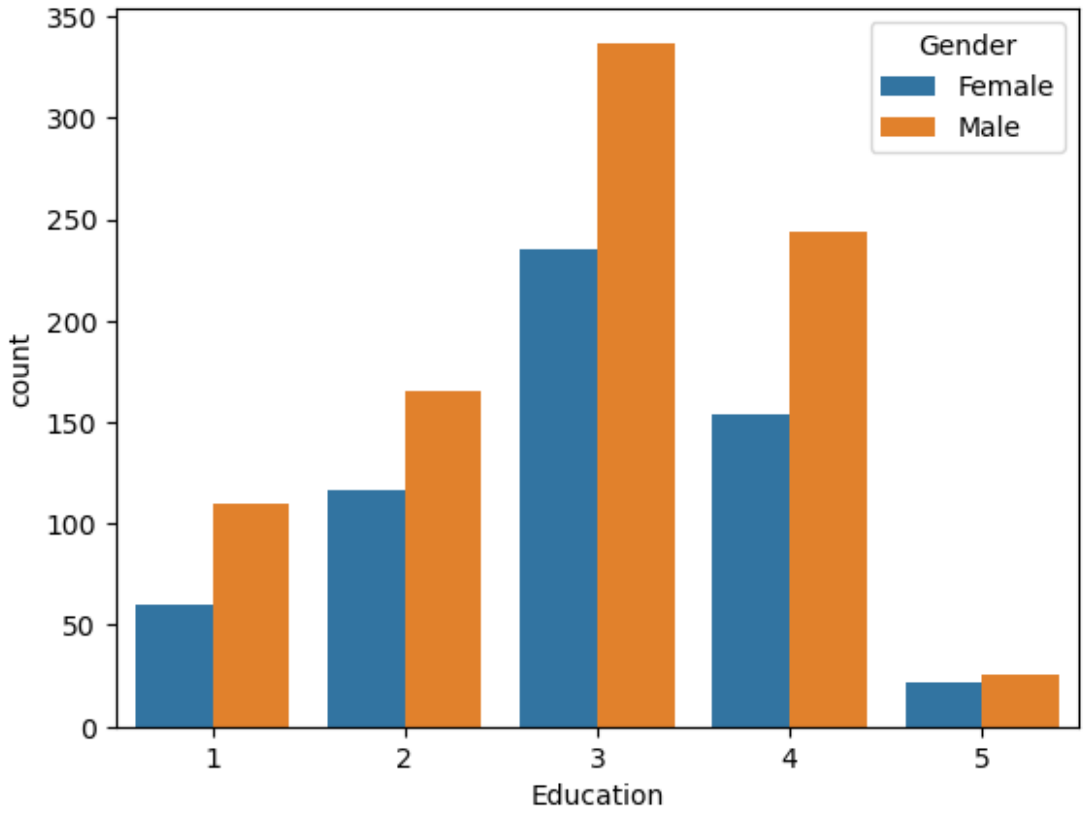

```
pd.crosstab(df['Gender'],df['OverTime']).plot(kind="bar",figsize=(10,6
))
```

```
<Axes: xlabel='Gender'>
```



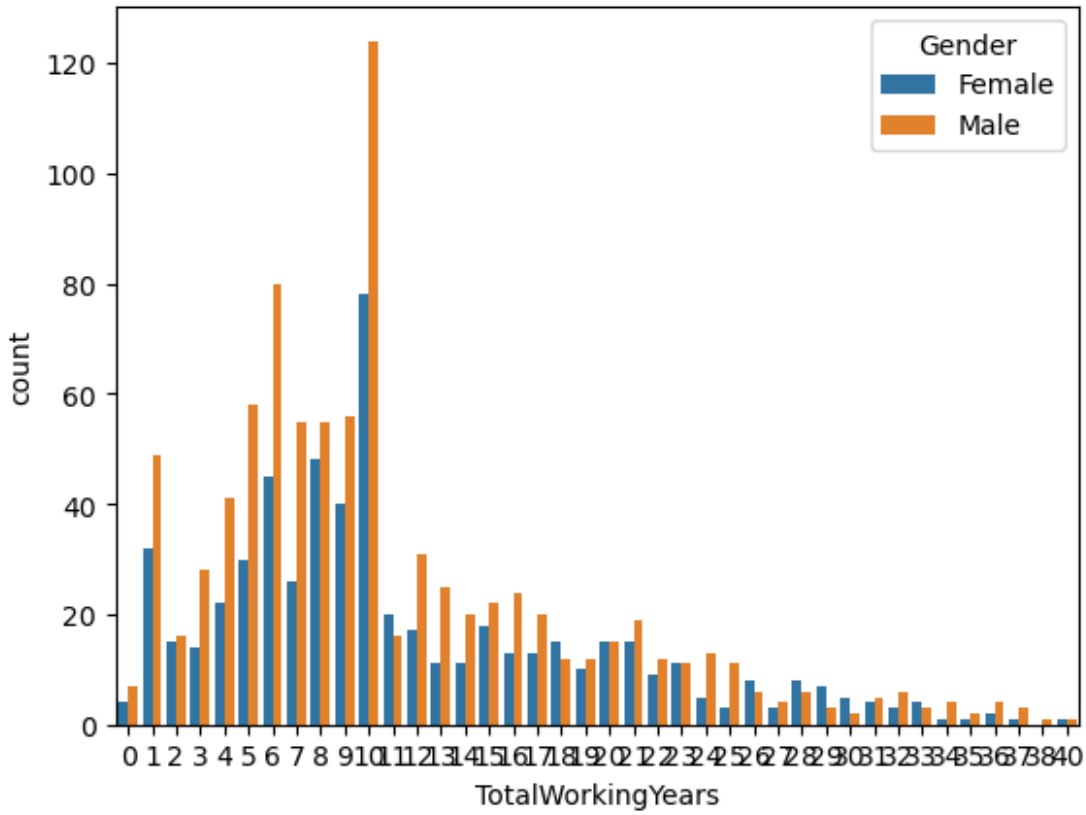

```
sns.countplot(x="Education",data=df,hue="Gender")
```

```
<Axes: xlabel='Education', ylabel='count'>
```



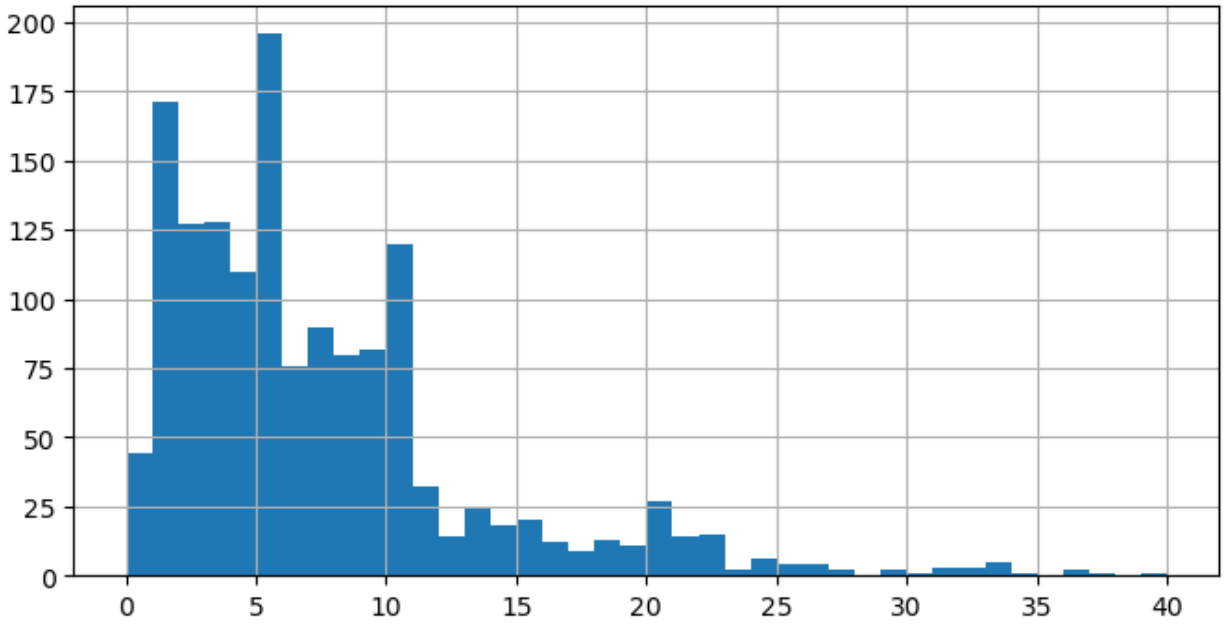

```
sns.countplot(x="TotalWorkingYears",data=df,hue="Gender")
```

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




```
df["YearsAtCompany"].hist(bins=40,figsize=(8,4))
```

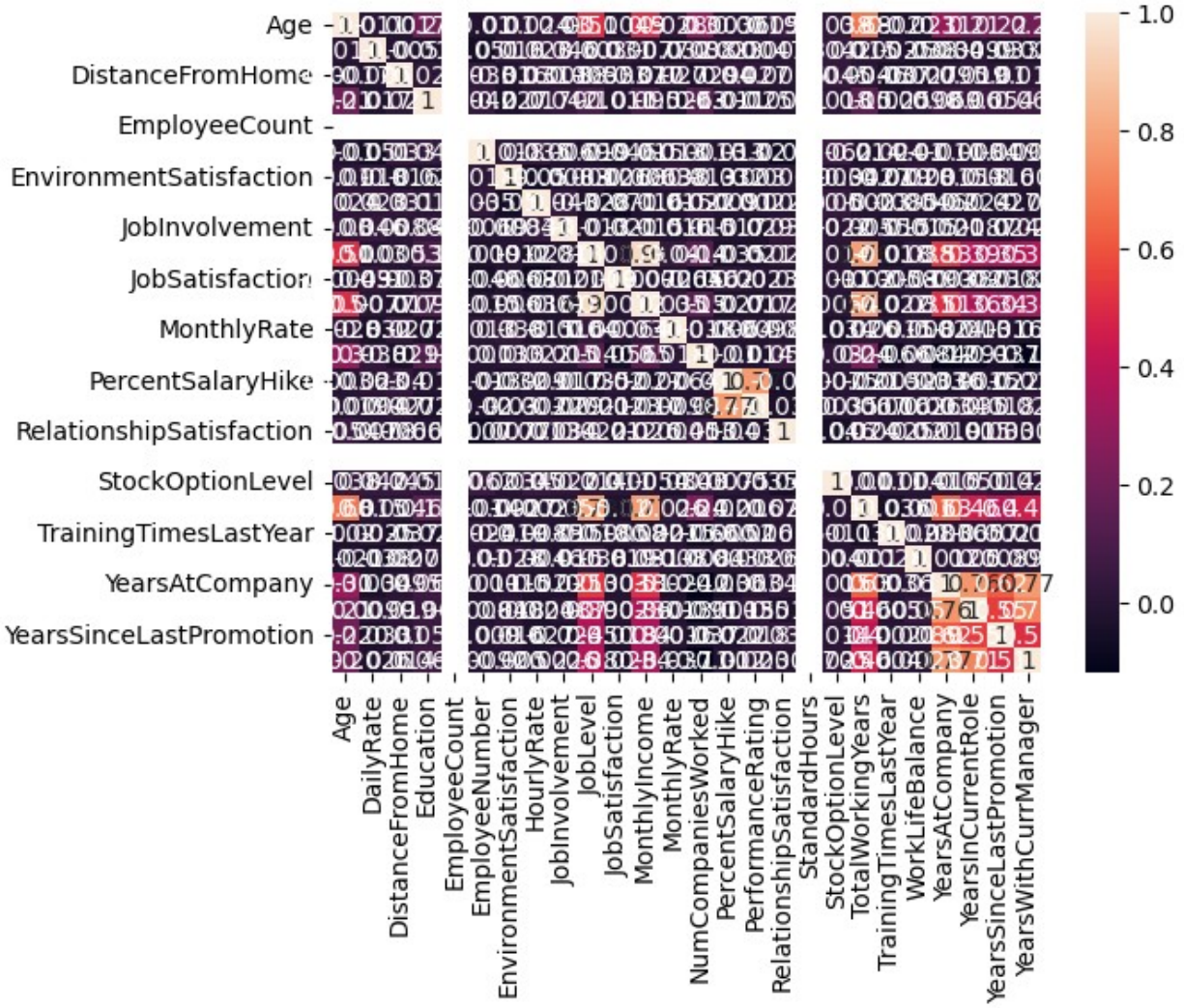
```
<Axes: >
```



```
sns.heatmap(df.corr(),annot=True)
```

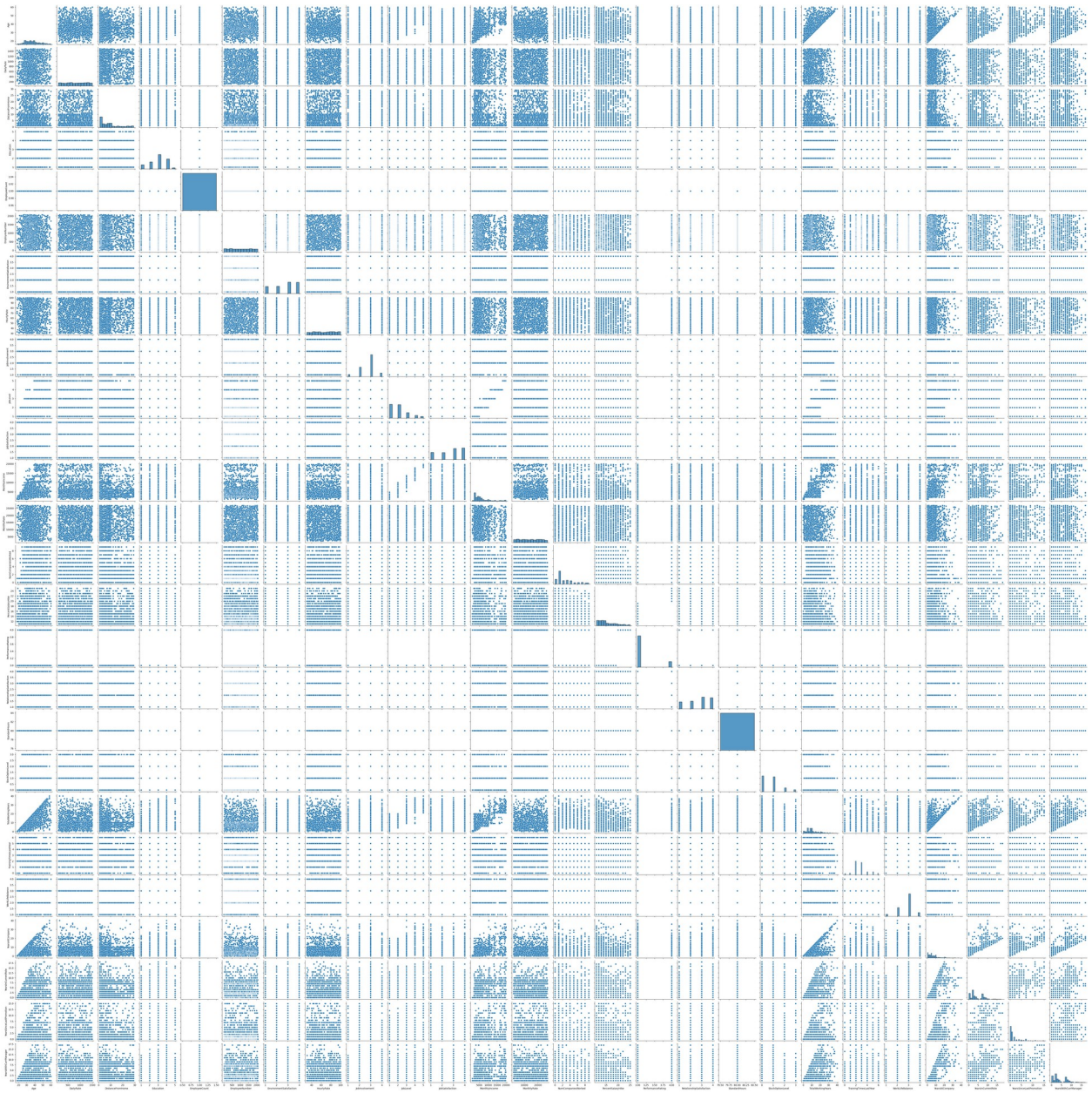
```
C:\Users\DELL\AppData\Local\Temp\ipykernel_22072\4277794465.py:1:
FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only
valid columns or specify the value of numeric_only to silence this
warning.
  sns.heatmap(df.corr(),annot=True)
```

```
<Axes: >
```



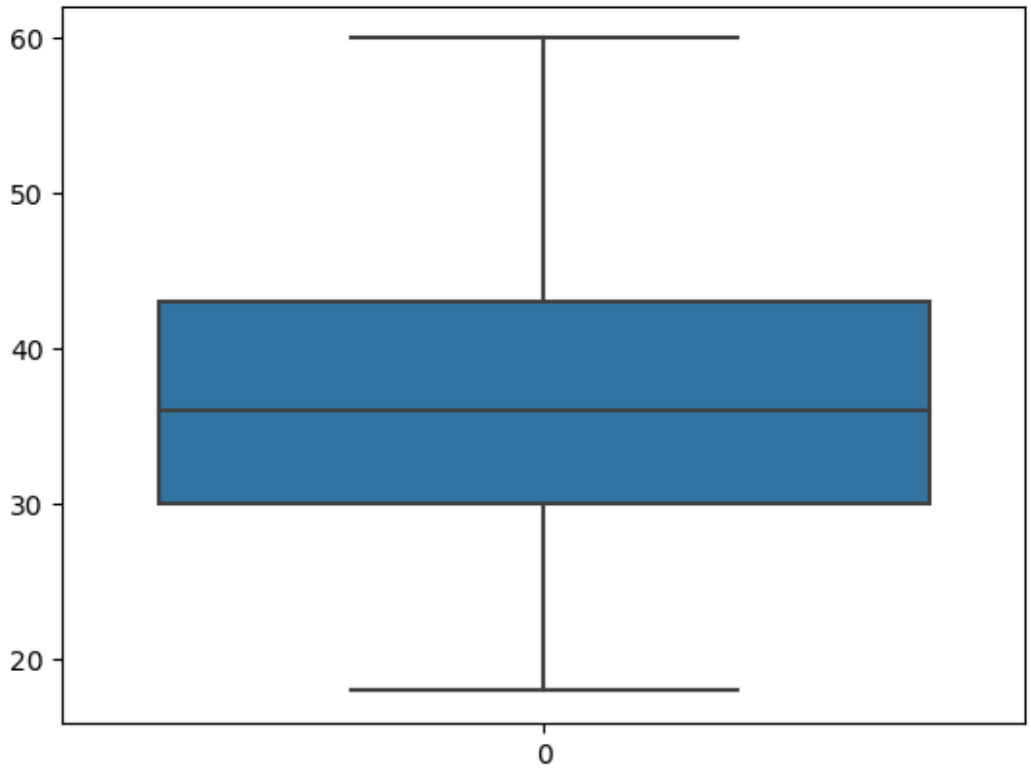

```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x2331e44b890>
```



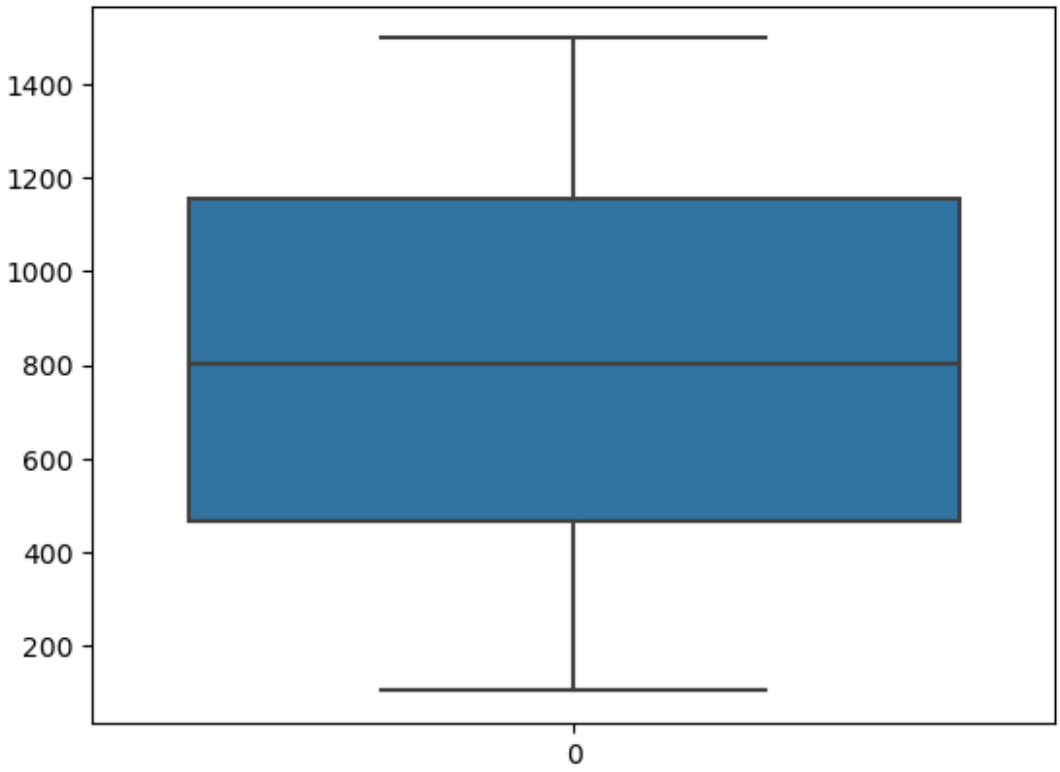
```
# Outlier detection
sns.boxplot(df["Age"])
```

```
<Axes: >
```



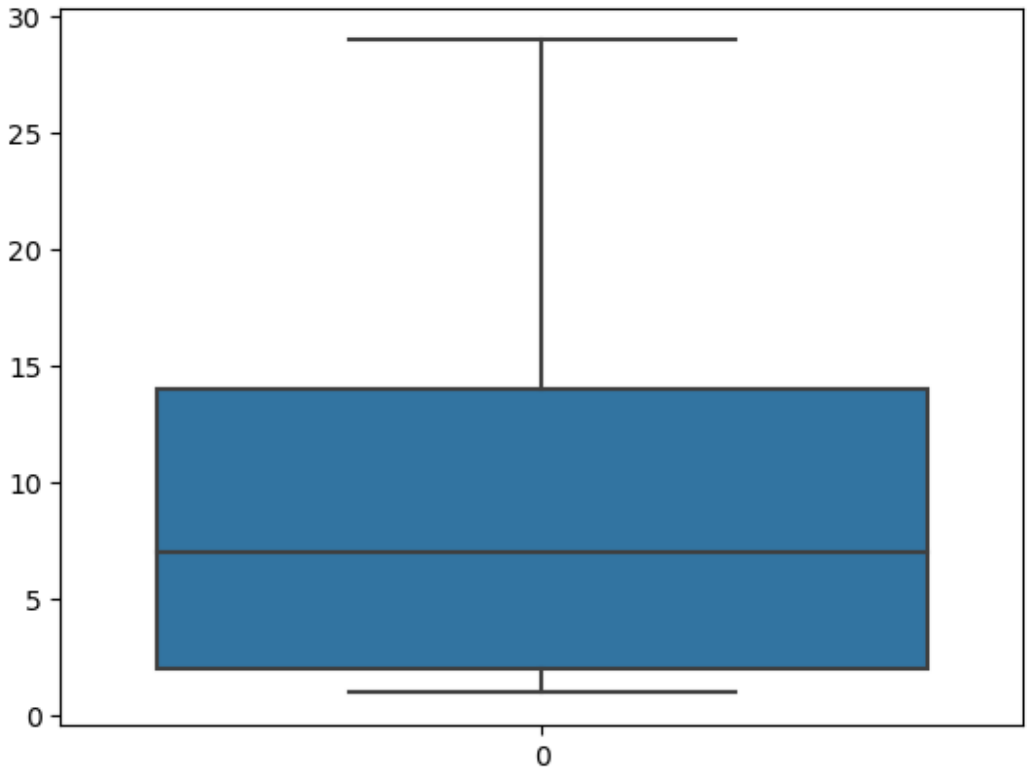

```
sns.boxplot(df["DailyRate"])
```

```
<Axes: >
```



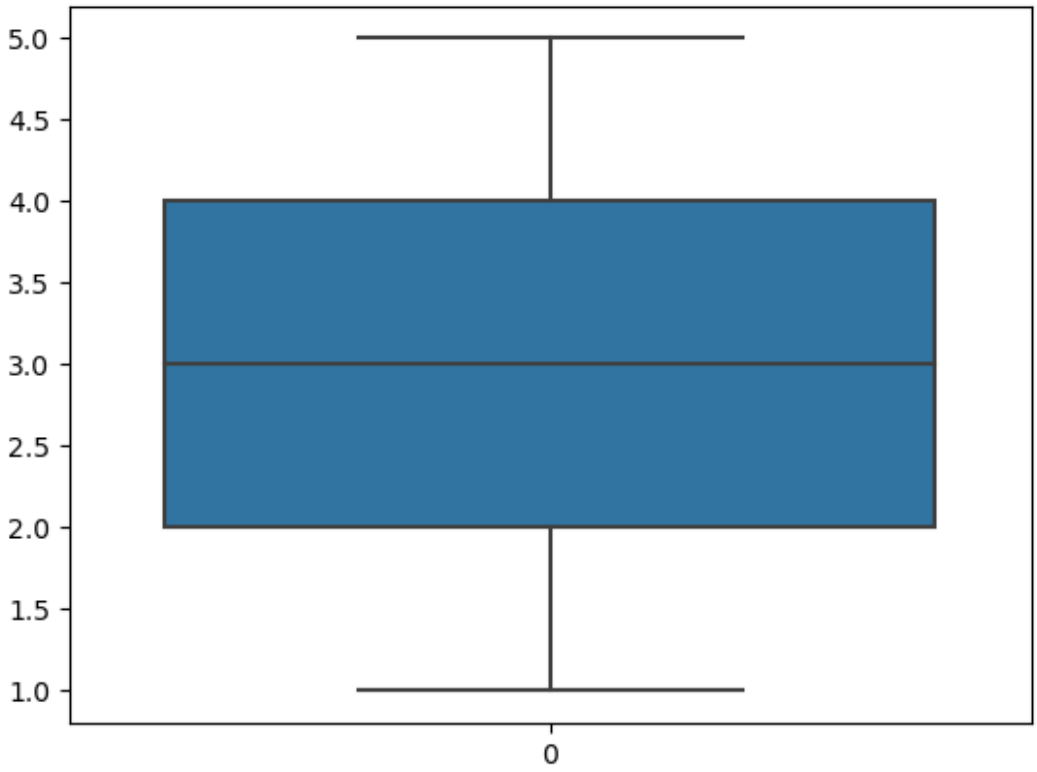

```
sns.boxplot(df["DistanceFromHome"])
```

```
<Axes: >
```



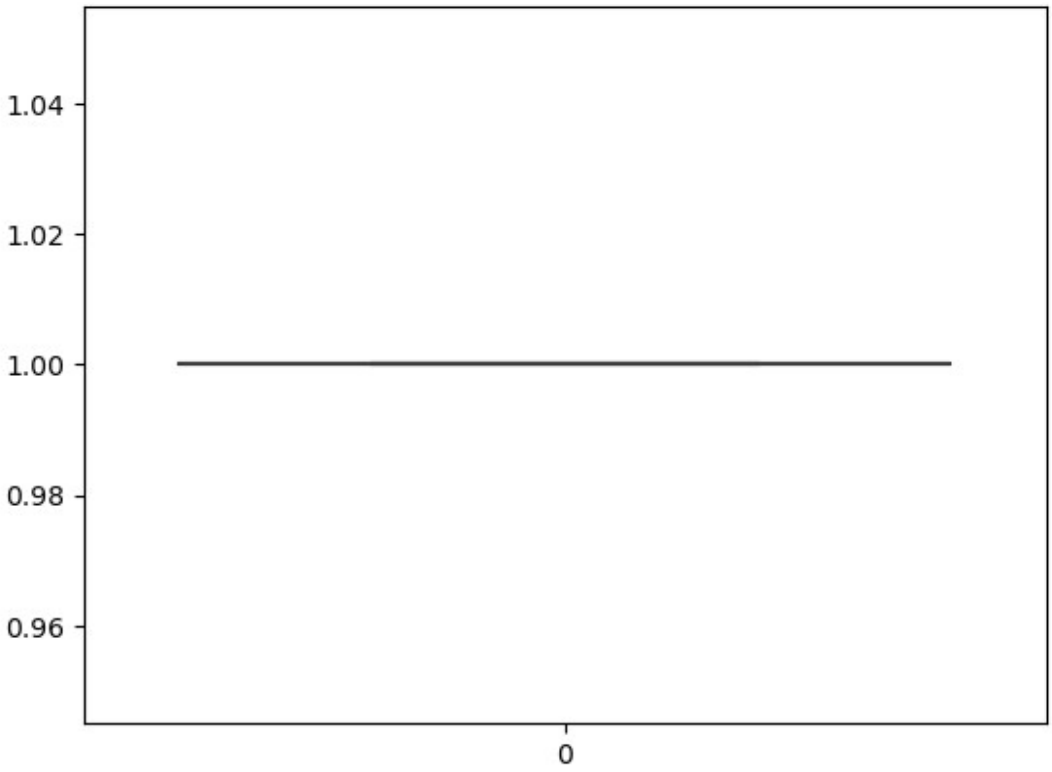

```
sns.boxplot(df["Education"])
```

```
<Axes: >
```



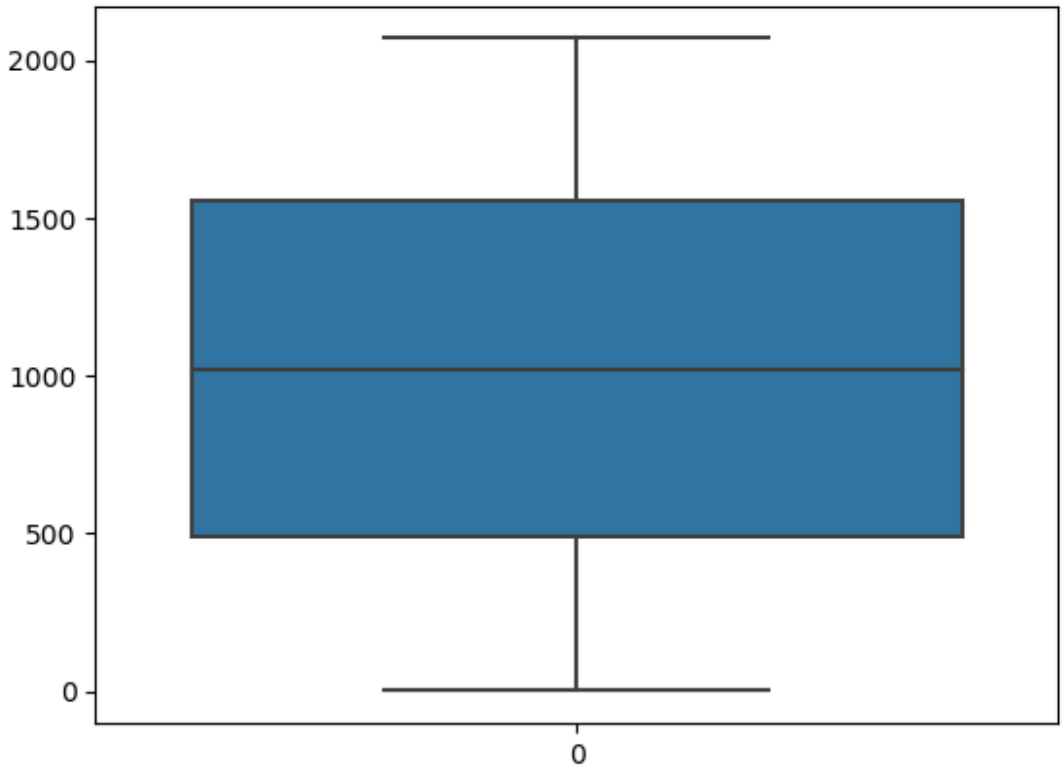

```
sns.boxplot(df["EmployeeCount"])
```

```
<Axes: >
```



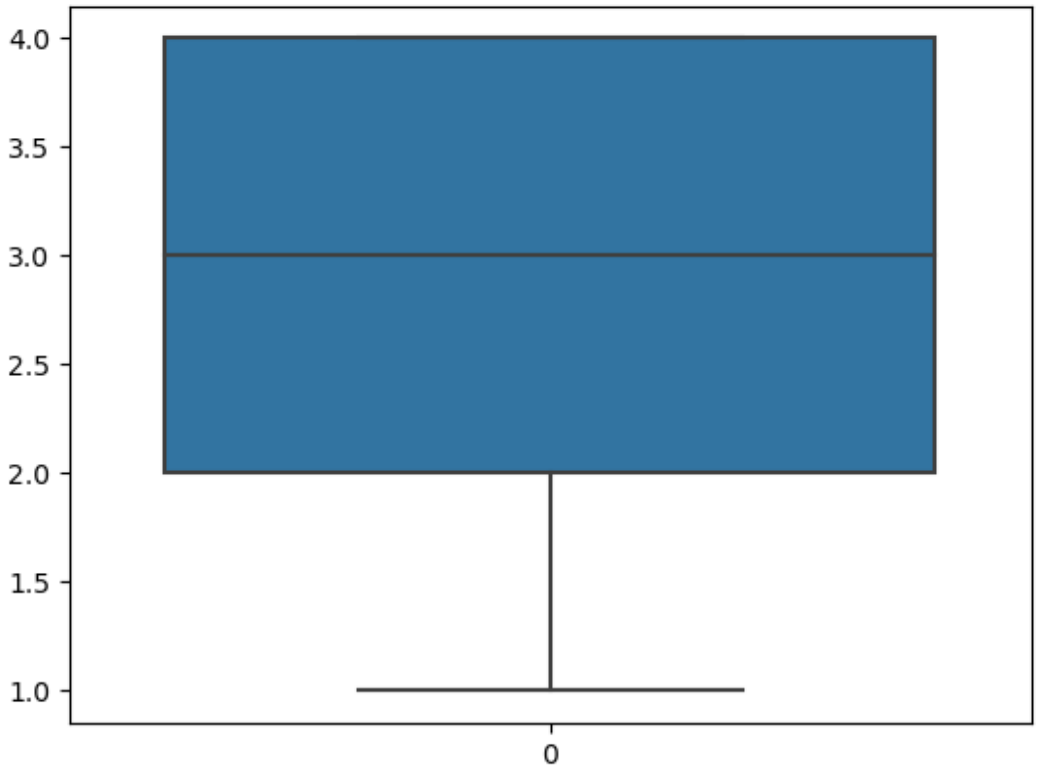

```
sns.boxplot(df["EmployeeNumber"])
```

```
<Axes: >
```



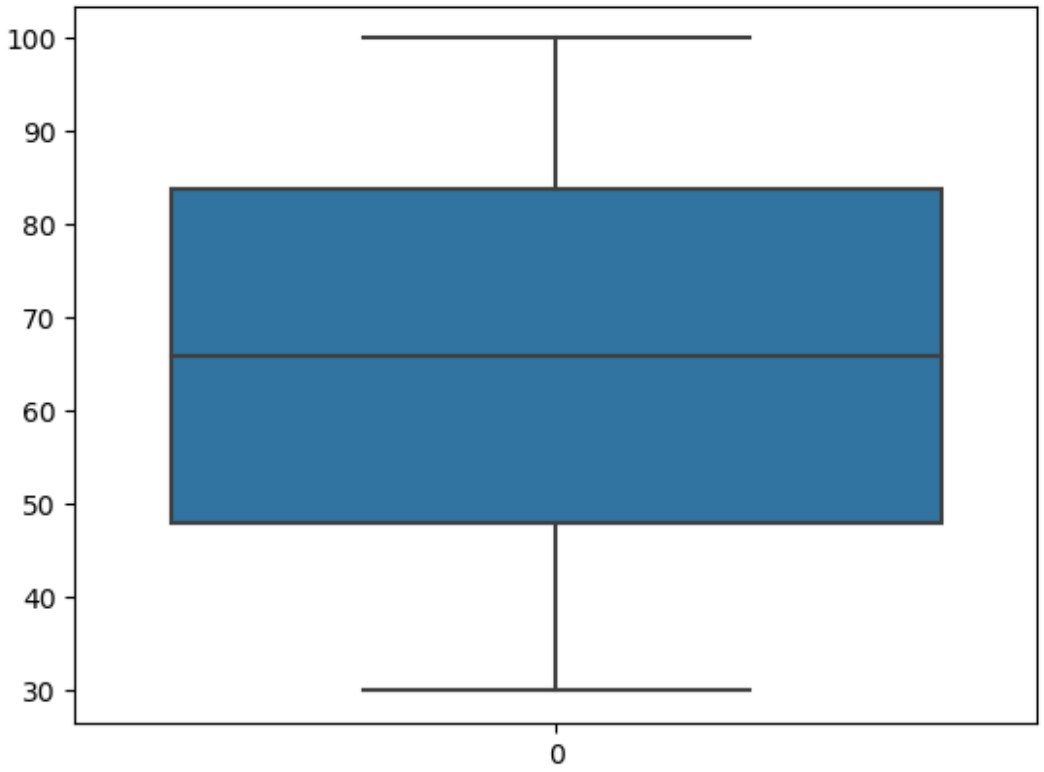

```
sns.boxplot(df["EnvironmentSatisfaction"])
```

```
<Axes: >
```



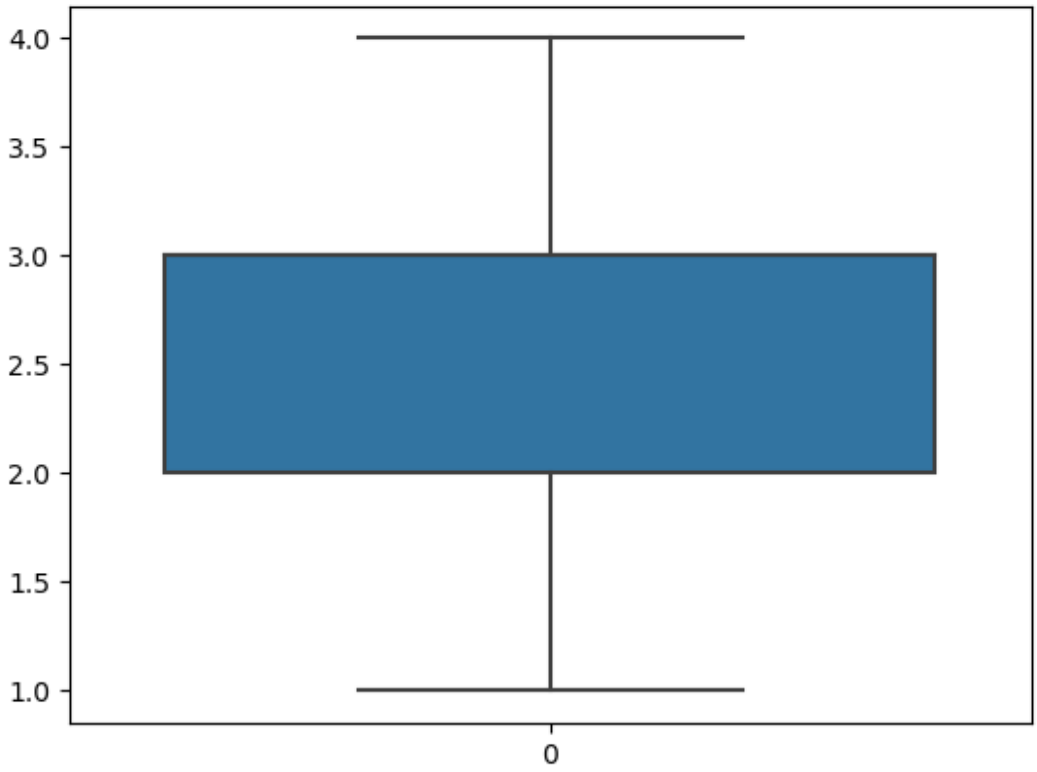

```
sns.boxplot(df["HourlyRate"])
```

```
<Axes: >
```



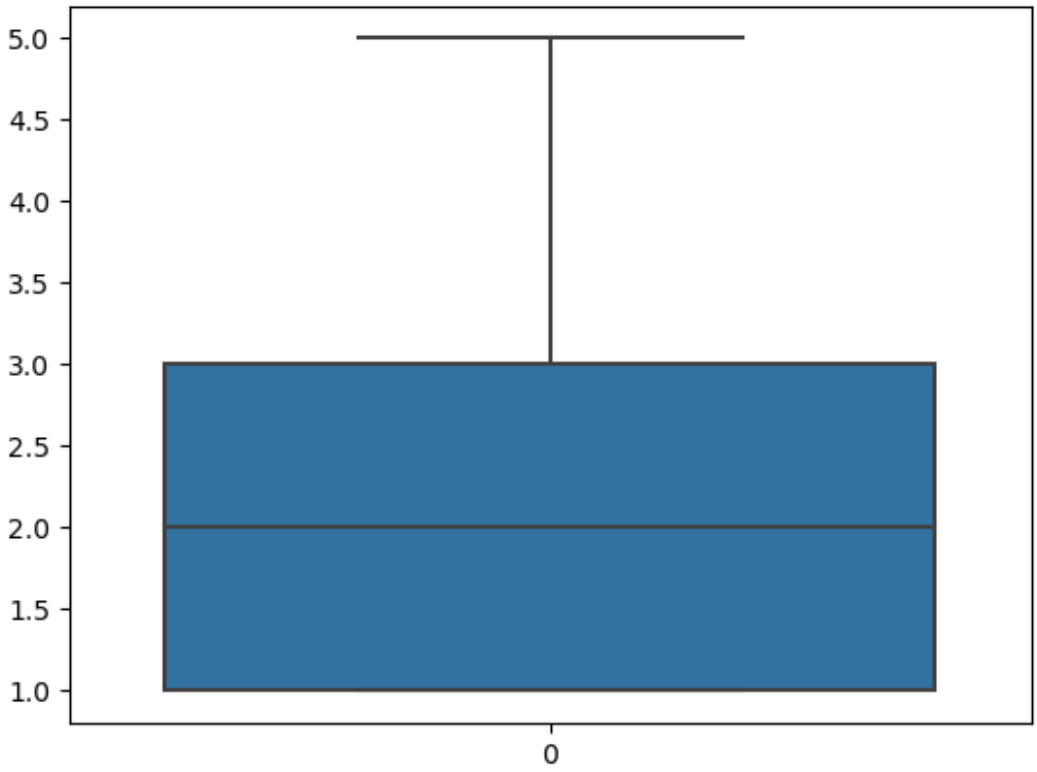

```
sns.boxplot(df["JobInvolvement"])
```

```
<Axes: >
```



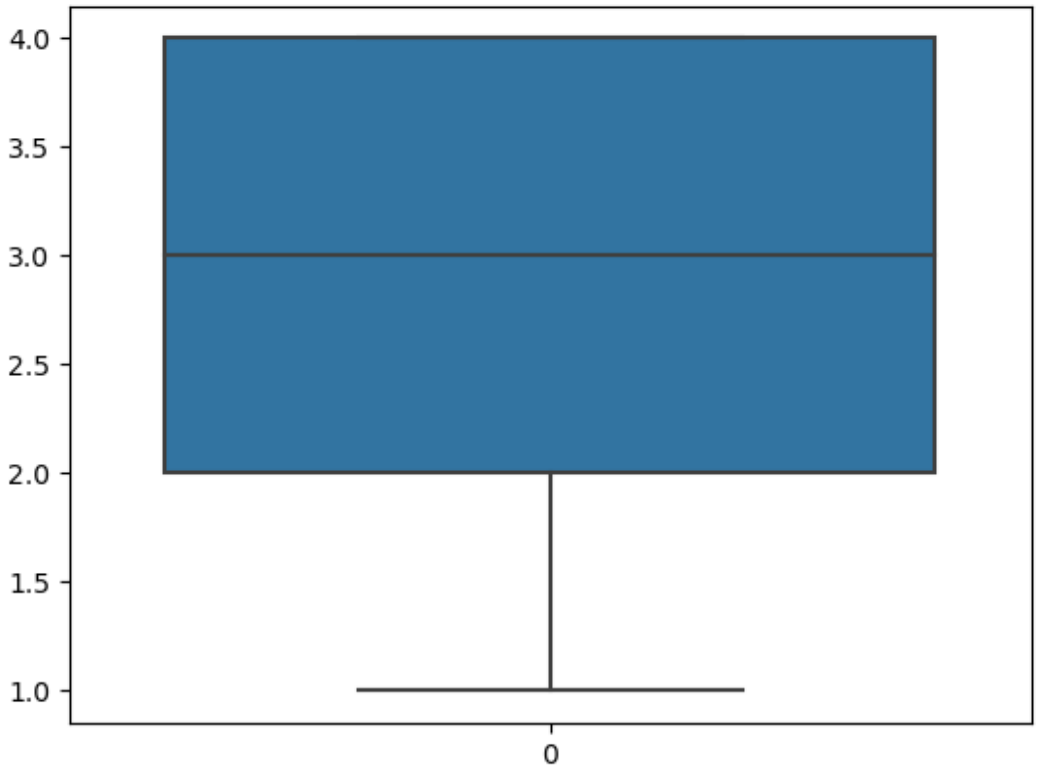

```
sns.boxplot(df["JobLevel"])
```

```
<Axes: >
```



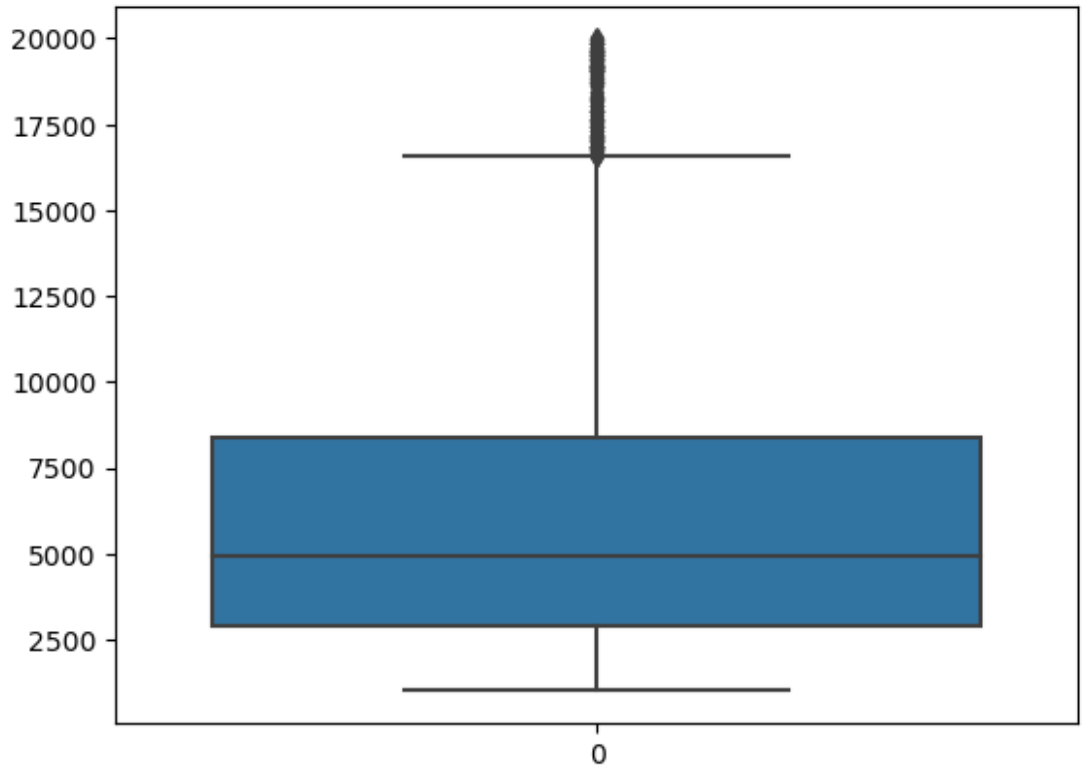

```
sns.boxplot(df["JobSatisfaction"])
```

```
<Axes: >
```




```
sns.boxplot(df["MonthlyIncome"])
```

```
<Axes: >
```




```
# Outlier removal by replacement with median
q1=df.MonthlyIncome.quantile(0.25)
q3=df.MonthlyIncome.quantile(0.75)

q1

2911.0

q3

8379.0

IQR=q3-q1

IQR

5468.0

upper_limit=q3+1.5*IQR
upper_limit

16581.0

lower_limit=q1-1.5*IQR
lower_limit

-5291.0
```

```
df.median()
```

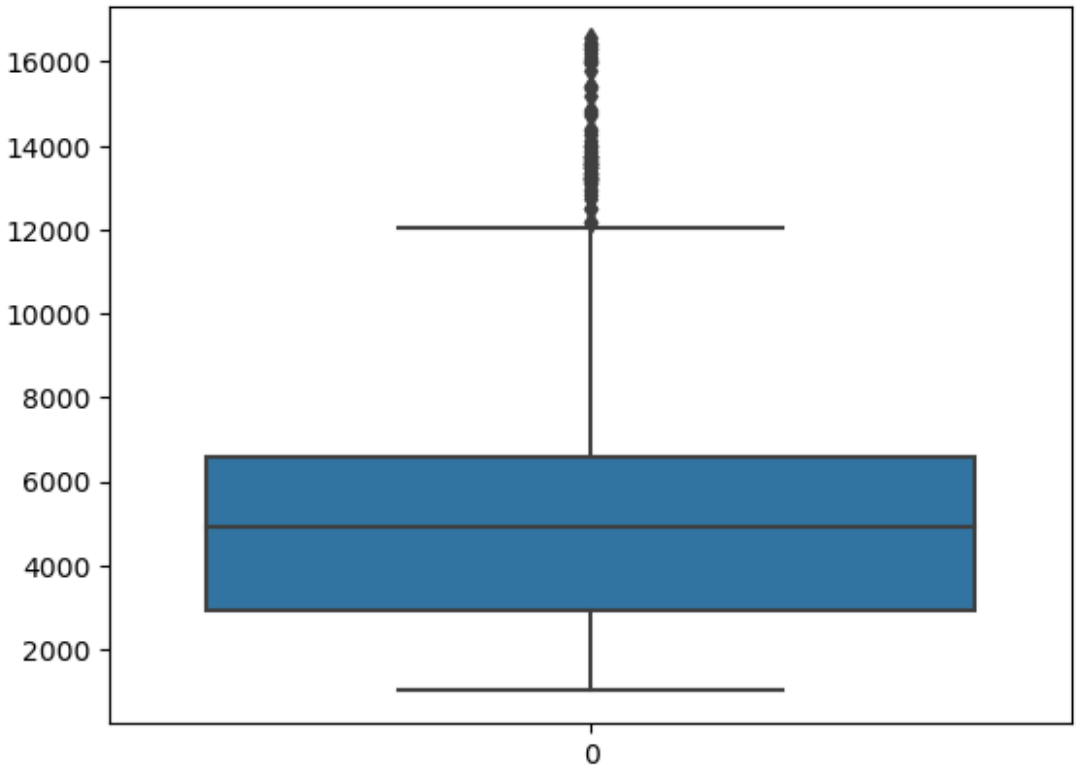
```
C:\Users\DELL\AppData\Local\Temp\ipykernel_22072\530051474.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median
is deprecated. In a future version, it will default to False. In
addition, specifying 'numeric_only=None' is deprecated. Select only
valid columns or specify the value of numeric_only to silence this
warning.
  df.median()
```

```
Age                         36.0
DailyRate                  802.0
DistanceFromHome             7.0
Education                    3.0
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
```

```
df['MonthlyIncome']=np.where(df['MonthlyIncome']>upper_limit,4919,df['
MonthlyIncome'])
```

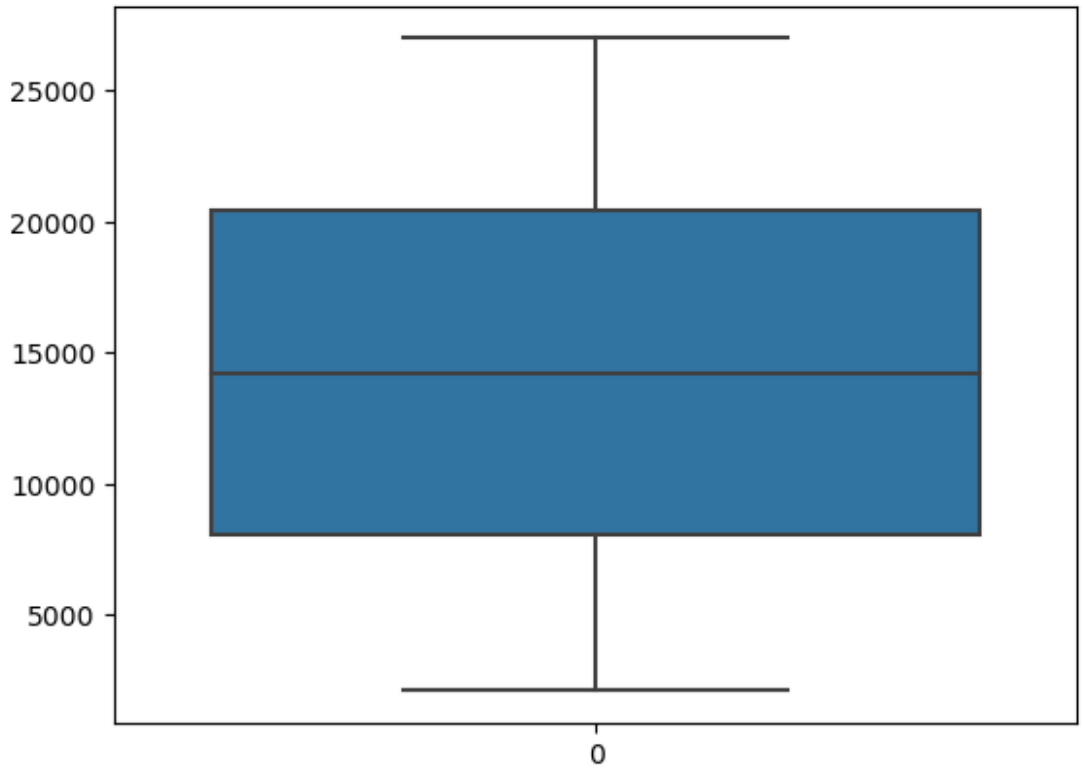
```
sns.boxplot(df.MonthlyIncome)
```

```
<Axes: >
```



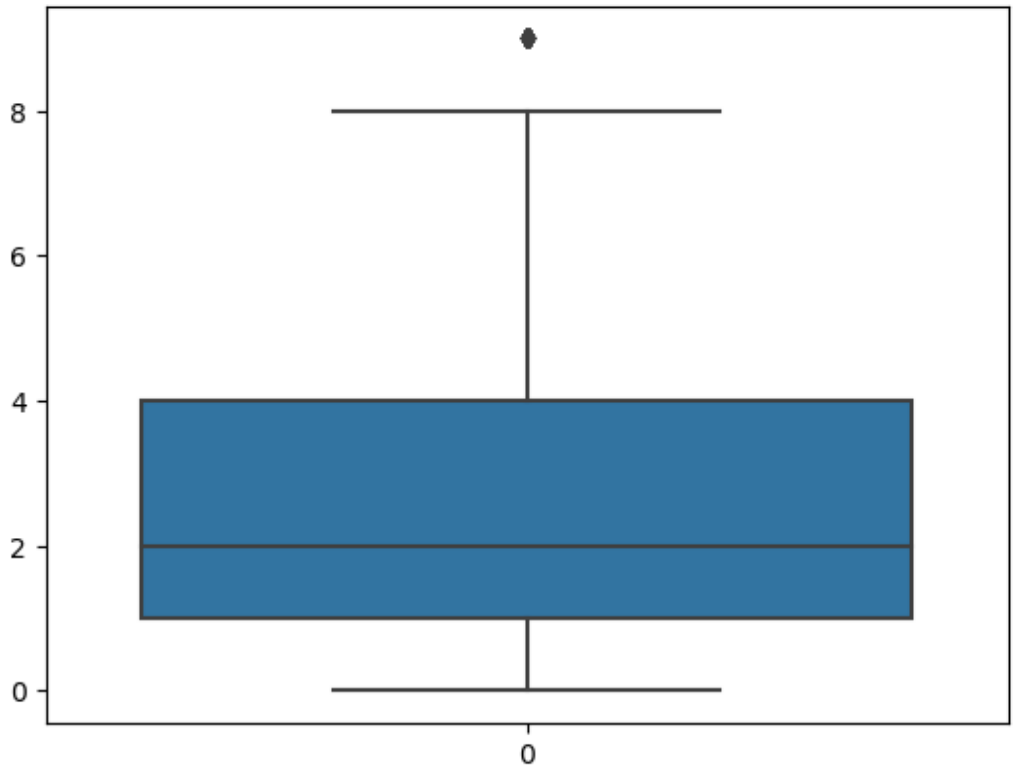

```
sns.boxplot(df["MonthlyRate"])
```

```
<Axes: >
```




```
sns.boxplot(df["NumCompaniesWorked"])
```

```
<Axes: >
```



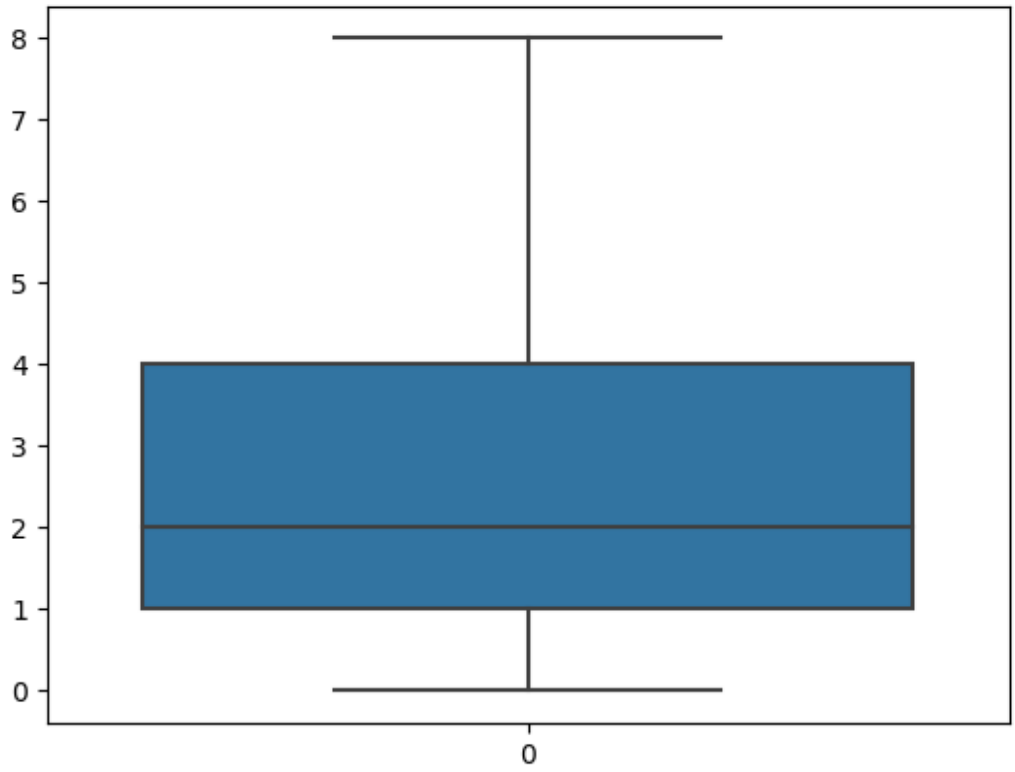

```
# Outlier removal by replacement with median
q1=df.NumCompaniesWorked.quantile(0.25)
q3=df.NumCompaniesWorked.quantile(0.75)

q1

1.0

q3

4.0

IQR=q3-q1
IQR

3.0

upper_limit=q3+1.5*IQR
upper_limit

8.5

lower_limit=q1-1.5*IQR
lower_limit

-3.5

df.median()
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_22072\530051474.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median
is deprecated. In a future version, it will default to False. In
addition, specifying 'numeric_only=None' is deprecated. Select only
valid columns or specify the value of numeric_only to silence this
warning.
  df.median()
```

```
Age                         36.0
DailyRate                  802.0
DistanceFromHome             7.0
Education                    3.0
EmployeeCount                1.0
EmployeeNumber            1020.5
EnvironmentSatisfaction      3.0
HourlyRate                  66.0
JobInvolvement               3.0
JobLevel                     2.0
JobSatisfaction              3.0
MonthlyIncome             4913.5
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
```

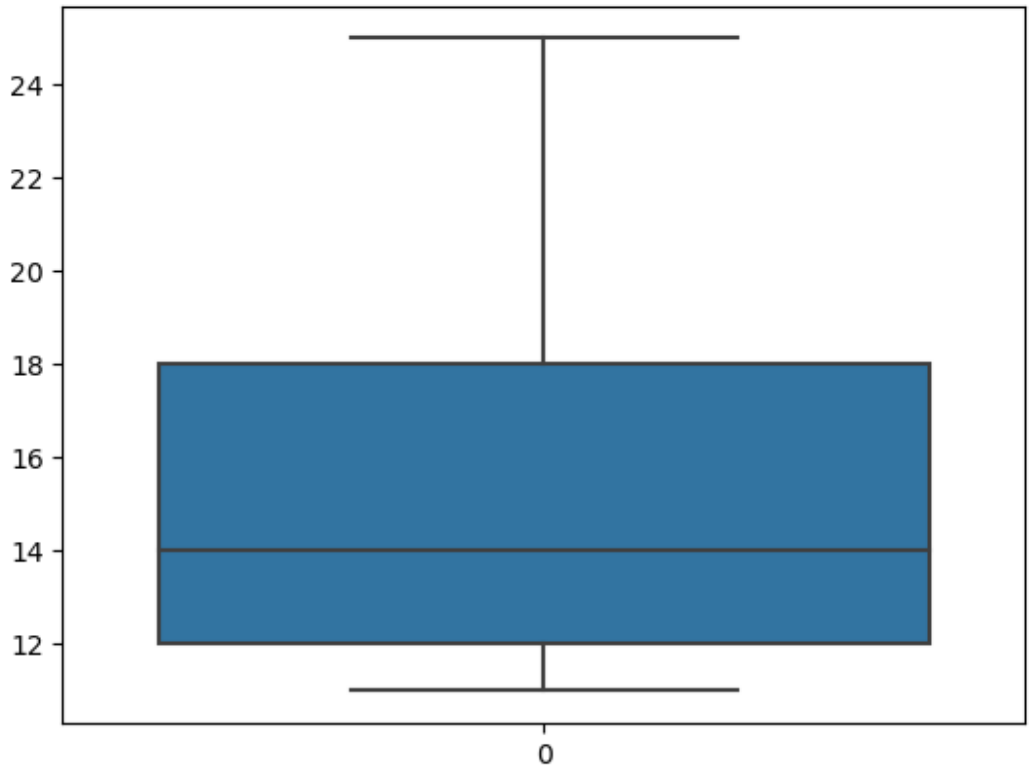
```
df['NumCompaniesWorked']=np.where(df['NumCompaniesWorked']>upper_limit
,2,df['NumCompaniesWorked'])

sns.boxplot(df["NumCompaniesWorked"])
```

```
<Axes: >
```



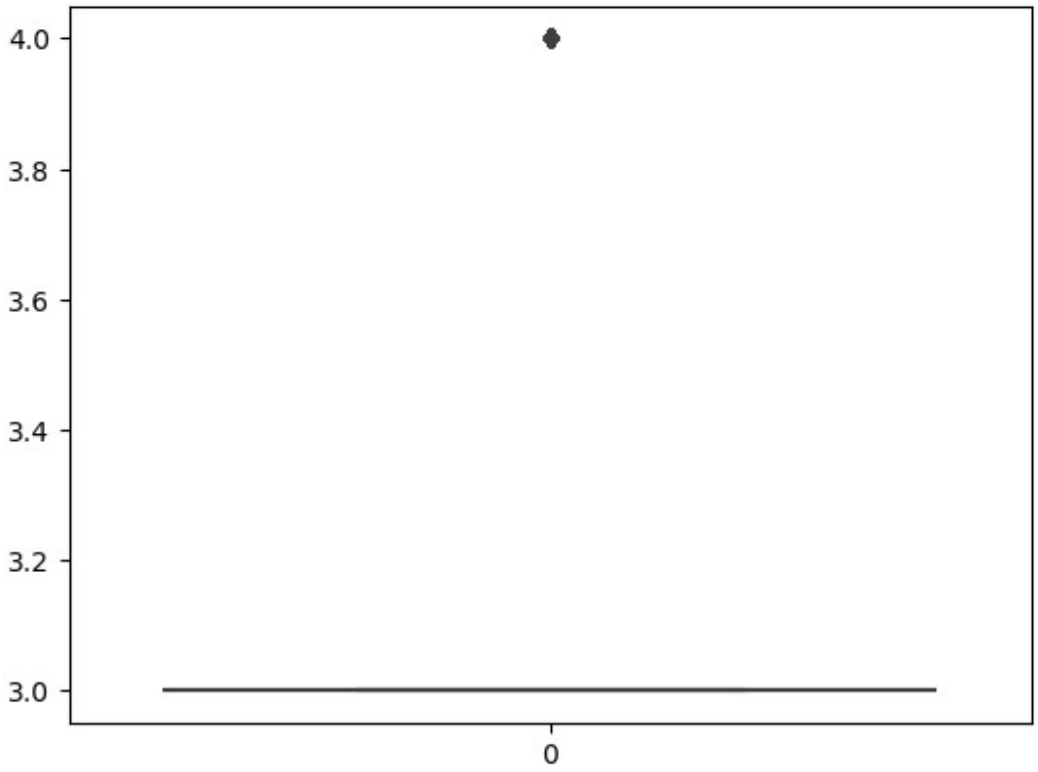

```
sns.boxplot(df["PercentSalaryHike"])
```

```
<Axes: >
```




```
sns.boxplot(df["PerformanceRating"])
```

```
<Axes: >
```



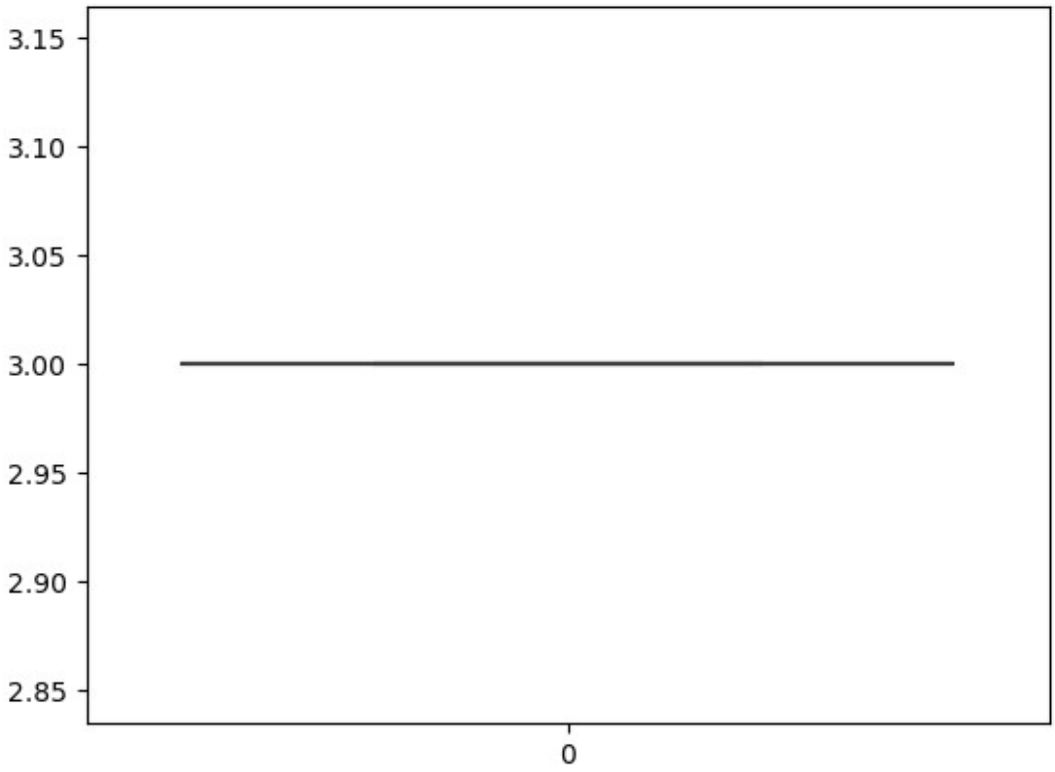

```
# Outlier removal by replacement with median
q1=df.PerformanceRating.quantile(0.25)
q3=df.PerformanceRating.quantile(0.75)

q1

3.0

q3

3.0

IQR=q3-q1
IQR

0.0

upper_limit=q3+1.5*IQR
upper_limit

3.0

lower_limit=q1-1.5*IQR
lower_limit

3.0

df.median()
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_22072\530051474.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median
is deprecated. In a future version, it will default to False. In
addition, specifying 'numeric_only=None' is deprecated. Select only
valid columns or specify the value of numeric_only to silence this
warning.
  df.median()
```

```
Age                         36.0
DailyRate                  802.0
DistanceFromHome             7.0
Education                    3.0
EmployeeCount                1.0
EmployeeNumber            1020.5
EnvironmentSatisfaction      3.0
HourlyRate                  66.0
JobInvolvement               3.0
JobLevel                     2.0
JobSatisfaction              3.0
MonthlyIncome             4913.5
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
```

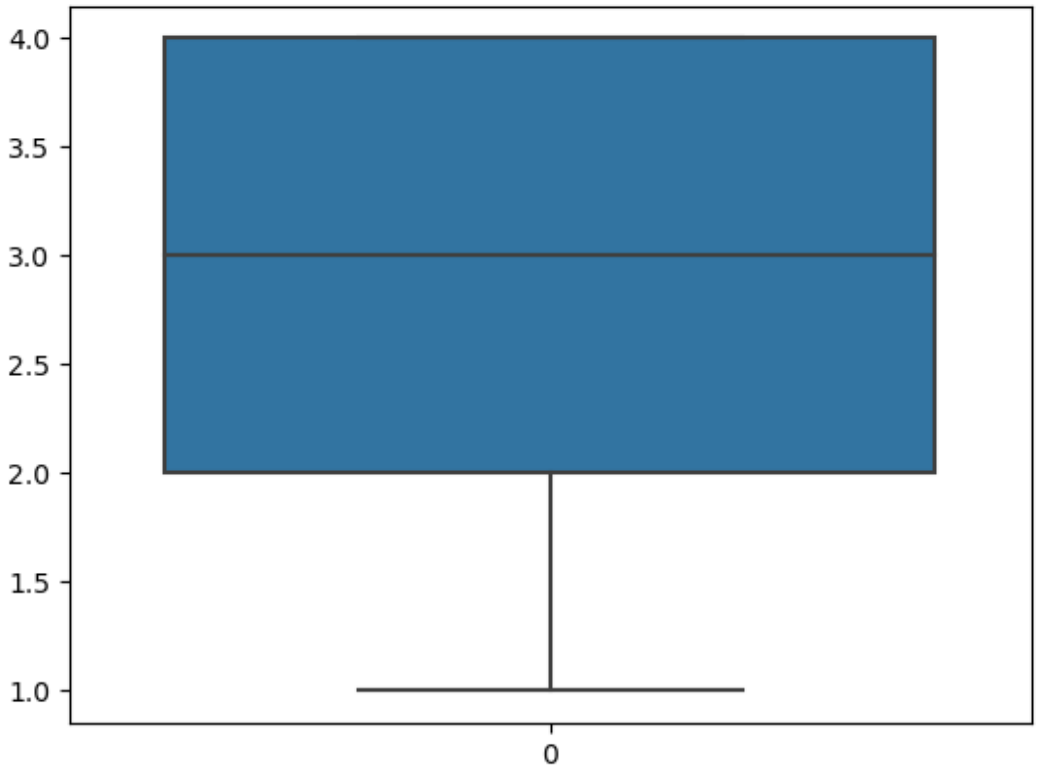
```
df['PerformanceRating']=np.where(df['PerformanceRating']>upper_limit,3
,df['PerformanceRating'])

sns.boxplot(df["PerformanceRating"])
```

```
<Axes: >
```



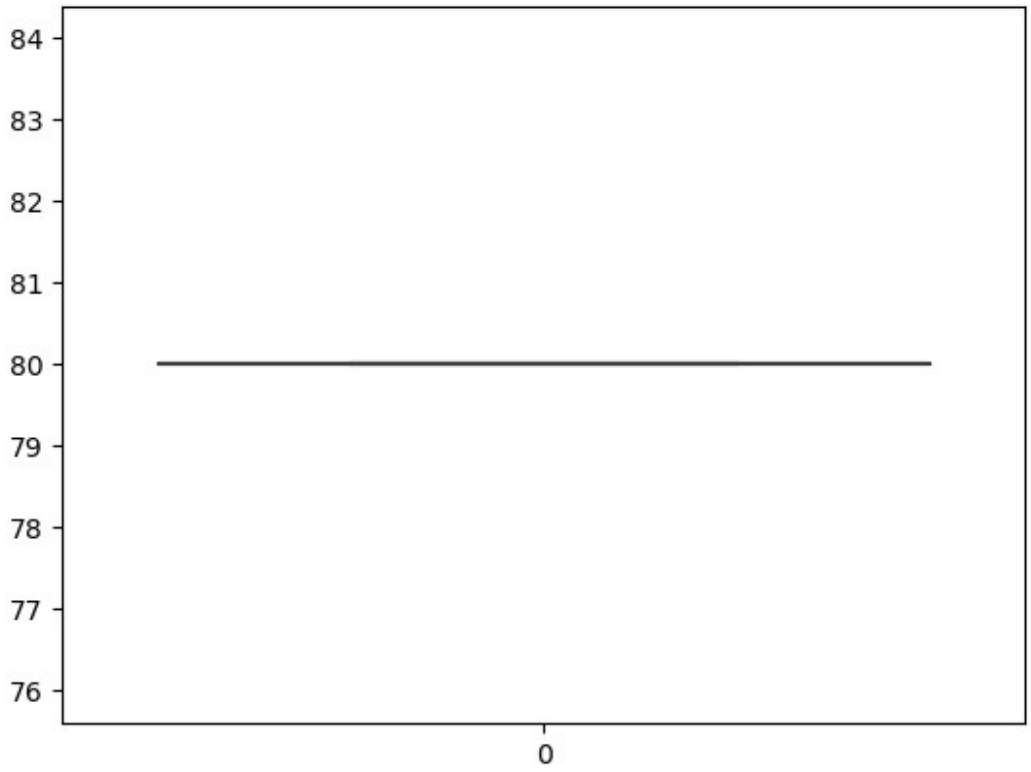

```
sns.boxplot(df["RelationshipSatisfaction"])
```

```
<Axes: >
```



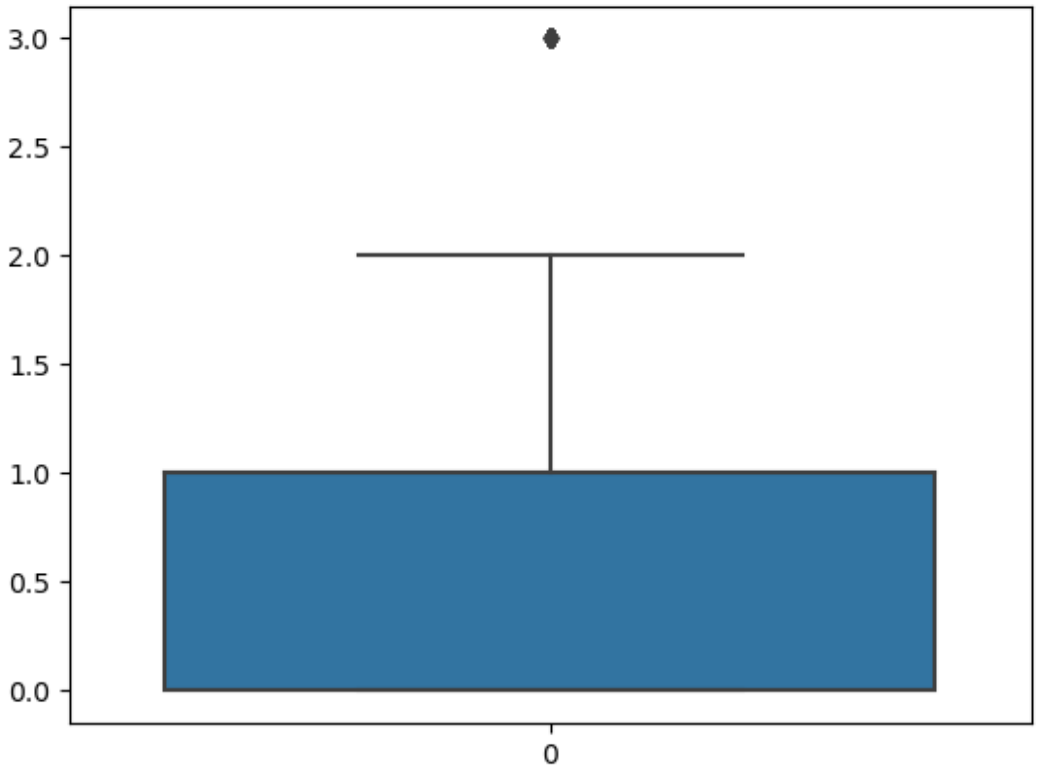

```
sns.boxplot(df["StandardHours"])
```

```
<Axes: >
```




```
sns.boxplot(df["StockOptionLevel"])
```

```
<Axes: >
```




```
# Outlier removal by replacement with median
q1=df.StockOptionLevel.quantile(0.25)
q3=df.StockOptionLevel.quantile(0.75)

q1
```

```
0.0
```

```
q3
```

```
1.0
```

```
IQR=q3-q1
IQR
```

```
1.0
```

```
upper_limit=q3+1.5*IQR
upper_limit
```

```
2.5
```

```
lower_limit=q1-1.5*IQR
lower_limit
```

```
-1.5
```

```
df.median()
```

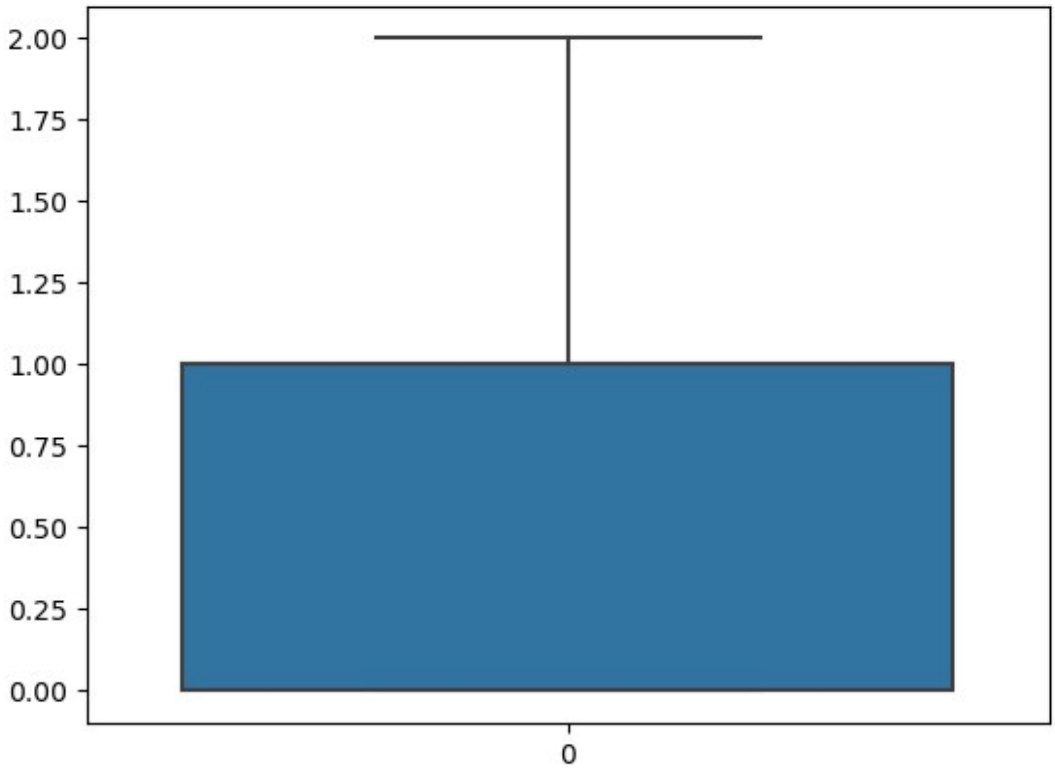
```
C:\Users\DELL\AppData\Local\Temp\ipykernel_22072\530051474.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median
is deprecated. In a future version, it will default to False. In
addition, specifying 'numeric_only=None' is deprecated. Select only
valid columns or specify the value of numeric_only to silence this
warning.
  df.median()
```

```
Age                         36.0
DailyRate                  802.0
DistanceFromHome             7.0
Education                    3.0
EmployeeCount                1.0
EmployeeNumber            1020.5
EnvironmentSatisfaction      3.0
HourlyRate                  66.0
JobInvolvement               3.0
JobLevel                     2.0
JobSatisfaction              3.0
MonthlyIncome             4913.5
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
```

```
df['StockOptionLevel']=np.where(df['StockOptionLevel']>upper_limit,1,d
f['StockOptionLevel'])
```

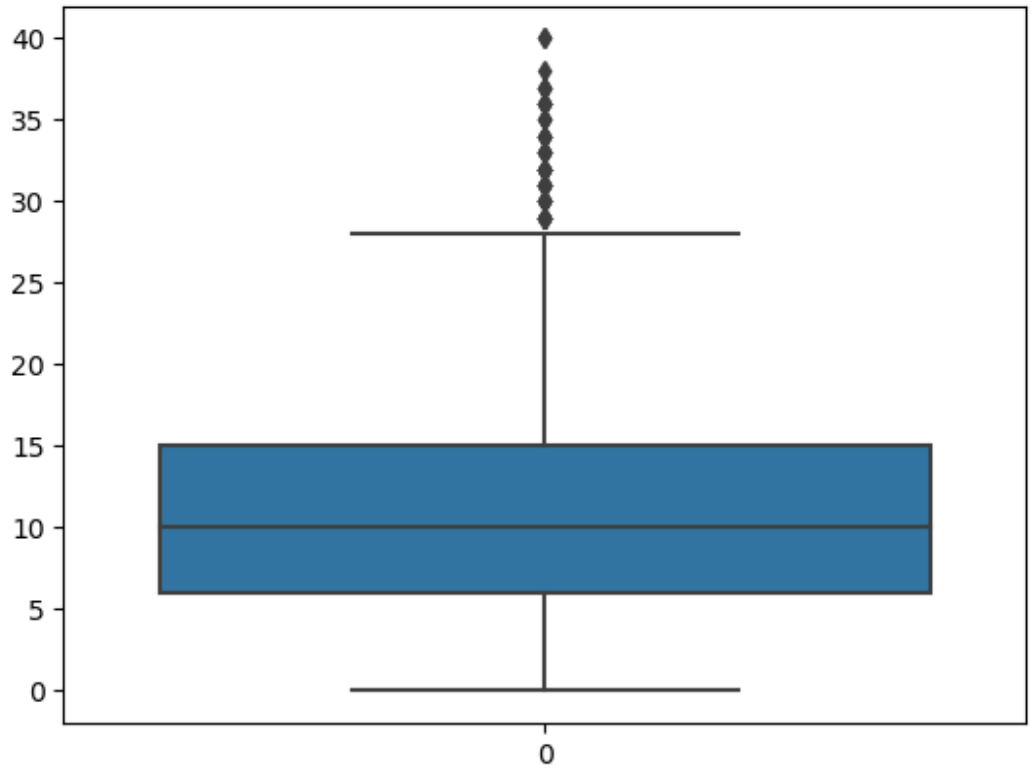
```
sns.boxplot(df["StockOptionLevel"])
```

```
<Axes: >
```




```
sns.boxplot(df["TotalWorkingYears"])
```

```
<Axes: >
```




```
# Outlier removal by replacement with median
q1=df.TotalWorkingYears.quantile(0.25)
q3=df.TotalWorkingYears.quantile(0.75)

q1
```

```
6.0
```

```
q3
```

```
15.0
```

```
IQR=q3-q1
IQR
```

```
9.0
```

```
upper_limit=q3+1.5*IQR
upper_limit
```

```
28.5
```

```
lower_limit=q1-1.5*IQR
lower_limit
```

```
-7.5
```

```
df.median()
```

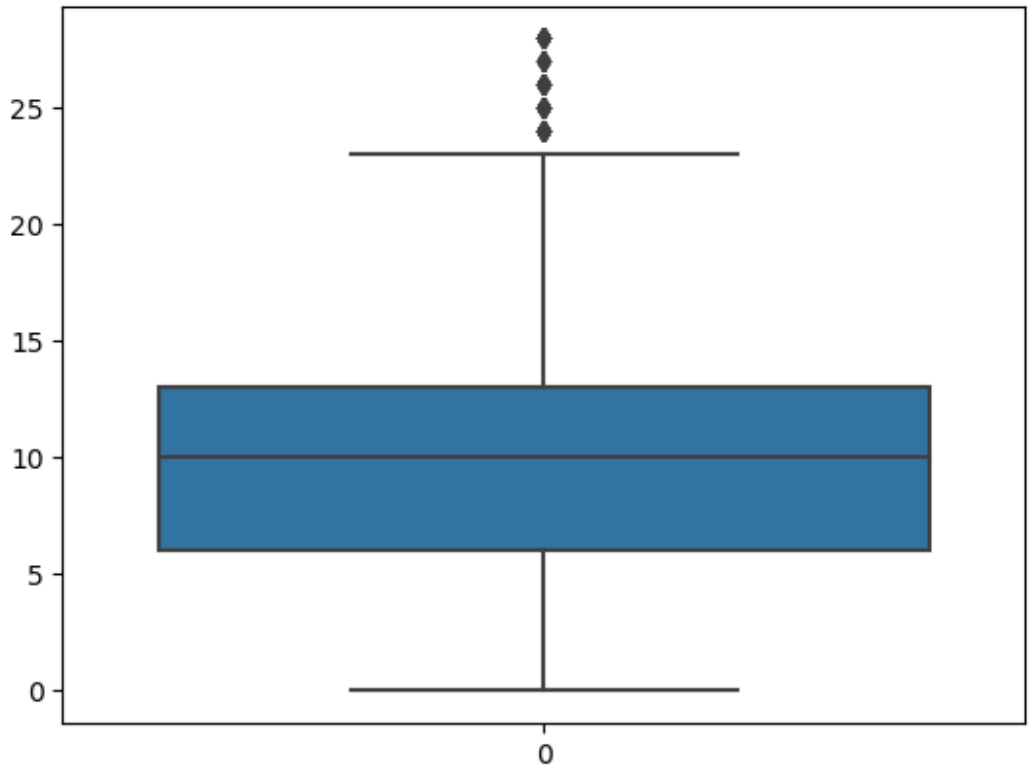
```
C:\Users\DELL\AppData\Local\Temp\ipykernel_22072\530051474.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median
is deprecated. In a future version, it will default to False. In
addition, specifying 'numeric_only=None' is deprecated. Select only
valid columns or specify the value of numeric_only to silence this
warning.
  df.median()
```

```
Age                         36.0
DailyRate                  802.0
DistanceFromHome             7.0
Education                    3.0
EmployeeCount                1.0
EmployeeNumber            1020.5
EnvironmentSatisfaction      3.0
HourlyRate                  66.0
JobInvolvement               3.0
JobLevel                     2.0
JobSatisfaction              3.0
MonthlyIncome             4913.5
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
```

```
df['TotalWorkingYears']=np.where(df['TotalWorkingYears']>upper_limit,10,df['TotalWorkingYears'])
```

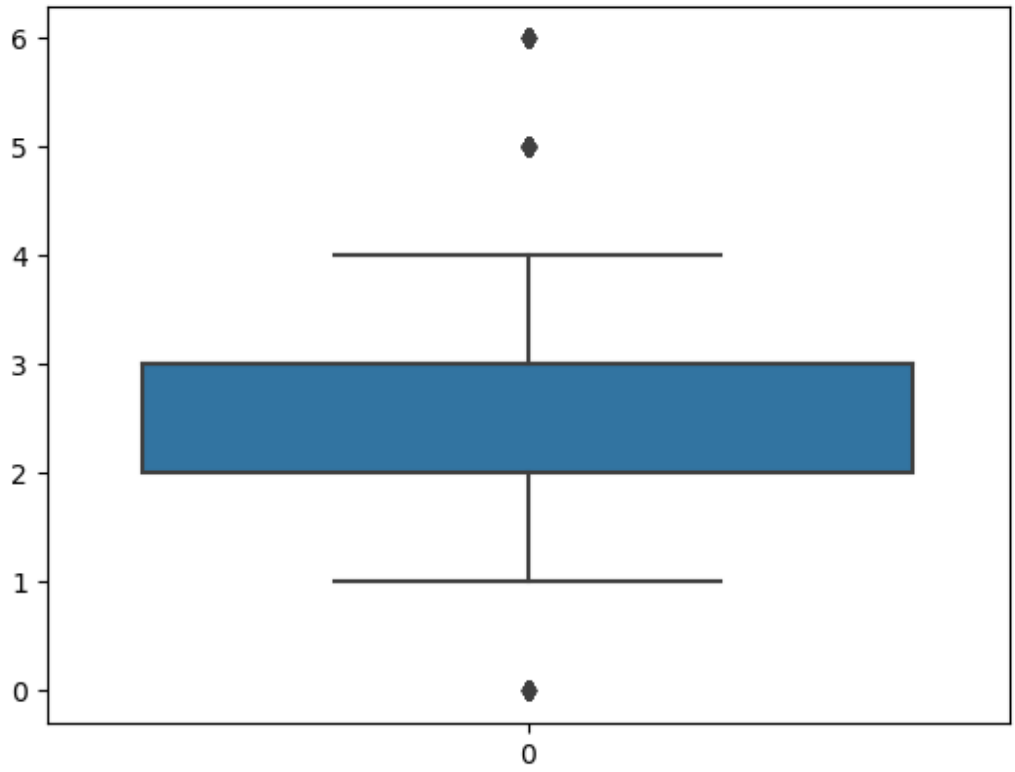
```
sns.boxplot(df.TotalWorkingYears)
```

```
<Axes: >
```




```
sns.boxplot(df["TrainingTimesLastYear"])
```

```
<Axes: >
```




```
# Outlier removal by replacement with median
q1=df.TrainingTimesLastYear.quantile(0.25)
q3=df.TrainingTimesLastYear.quantile(0.75)

q1
```

```
2.0
```

```
q3
```

```
3.0
```

```
IQR=q3-q1
IQR
```

```
1.0
```

```
upper_limit=q3+1.5*IQR
upper_limit
```

```
4.5
```

```
lower_limit=q1-1.5*IQR
lower_limit
```

```
0.5
```

```
df.median()
```

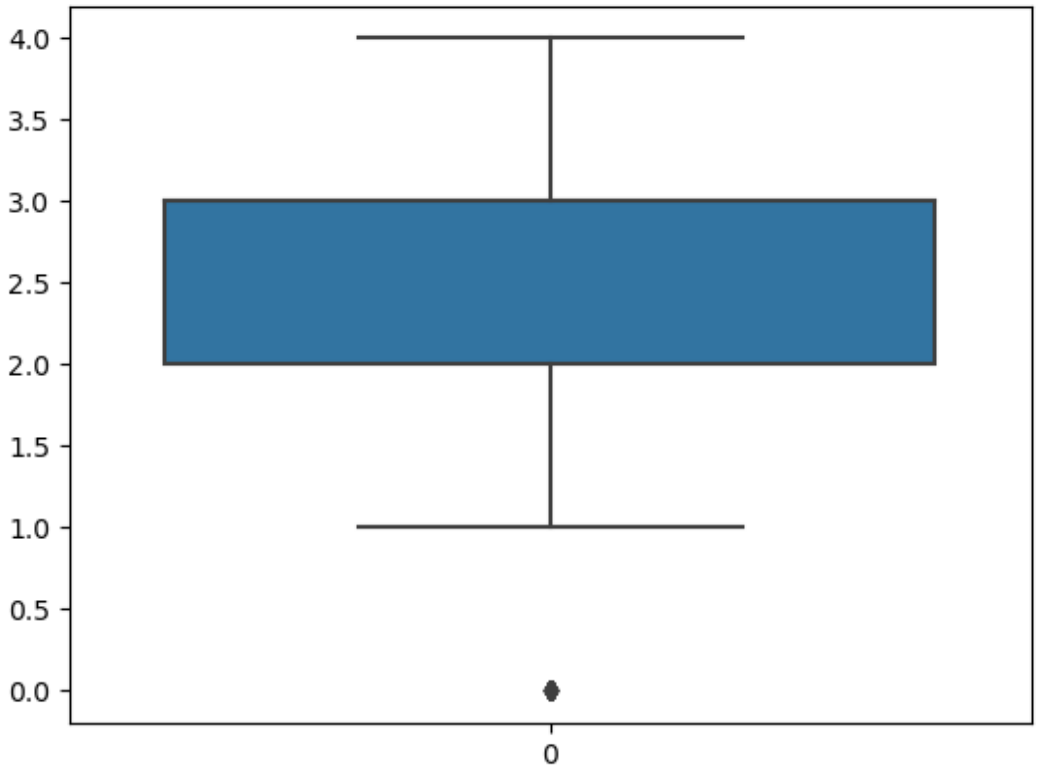
```
C:\Users\DELL\AppData\Local\Temp\ipykernel_22072\530051474.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median
is deprecated. In a future version, it will default to False. In
addition, specifying 'numeric_only=None' is deprecated. Select only
valid columns or specify the value of numeric_only to silence this
warning.
  df.median()
```

```
Age                         36.0
DailyRate                  802.0
DistanceFromHome             7.0
Education                    3.0
EmployeeCount                1.0
EmployeeNumber            1020.5
EnvironmentSatisfaction      3.0
HourlyRate                  66.0
JobInvolvement               3.0
JobLevel                     2.0
JobSatisfaction              3.0
MonthlyIncome             4913.5
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
```

```
df['TrainingTimesLastYear']=np.where(df['TrainingTimesLastYear']>upper
_limit,3,df['TrainingTimesLastYear'])
```

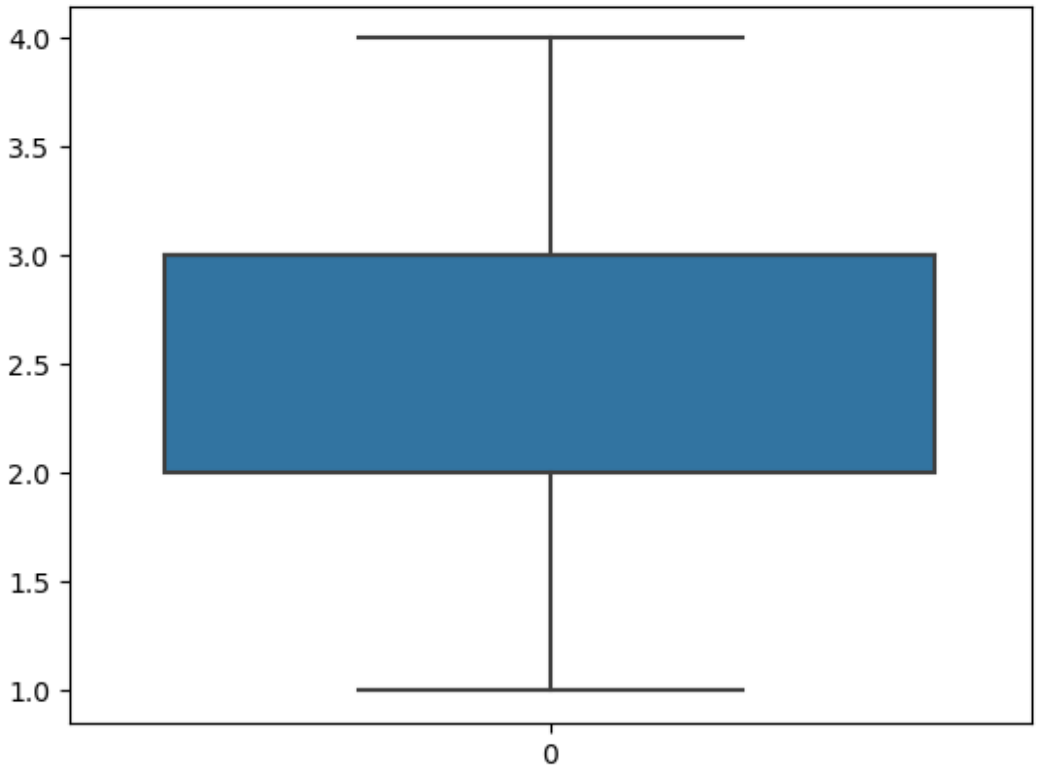
```
sns.boxplot(df["TrainingTimesLastYear"])
```

```
<Axes: >
```



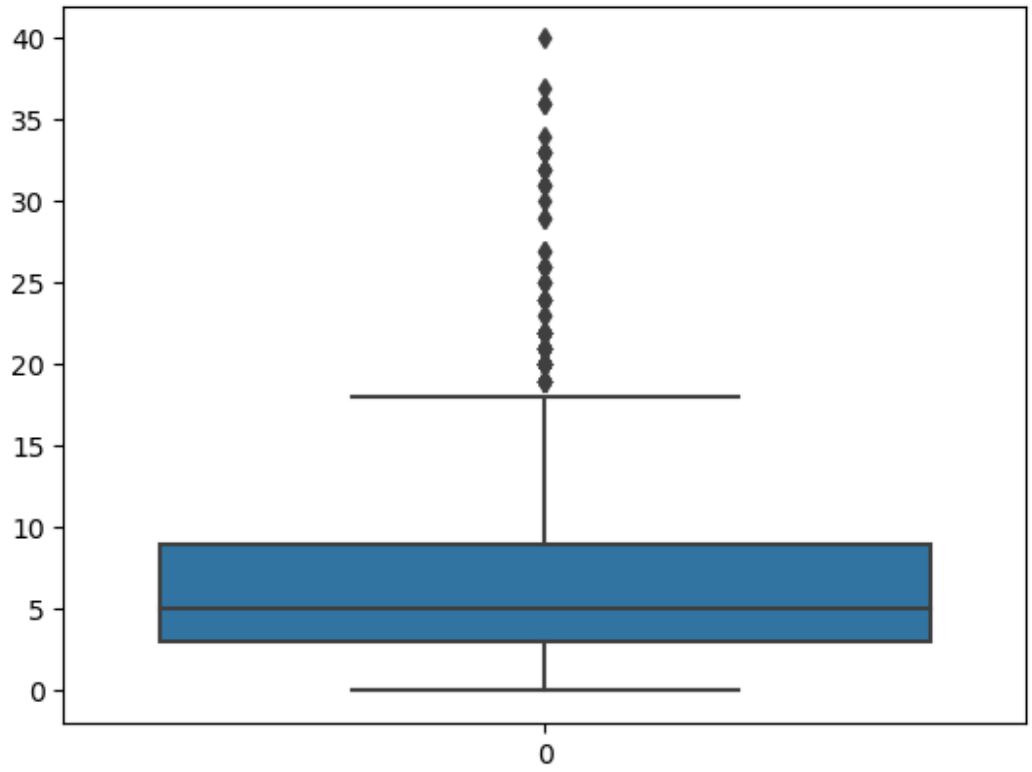

```
sns.boxplot(df["WorkLifeBalance"])
```

```
<Axes: >
```




```
sns.boxplot(df["YearsAtCompany"])
```

```
<Axes: >
```




```
# Outlier removal by replacement with median
q1=df.YearsAtCompany.quantile(0.25)
q3=df.YearsAtCompany.quantile(0.75)

q1
```

```
3.0
```

```
q3
```

```
9.0
```

```
IQR=q3-q1
IQR
```

```
6.0
```

```
upper_limit=q3+1.5*IQR
upper_limit
```

```
18.0
```

```
lower_limit=q1-1.5*IQR
lower_limit
```

```
-6.0
```

```
df.median()
```

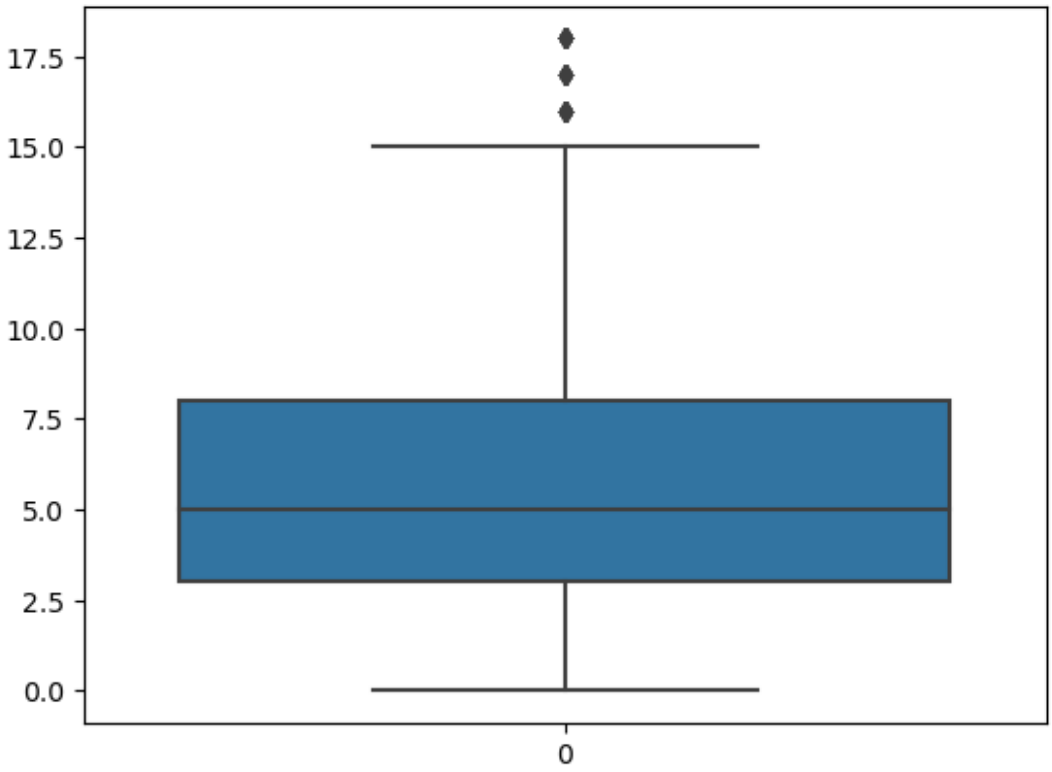
```
C:\Users\DELL\AppData\Local\Temp\ipykernel_22072\530051474.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median
is deprecated. In a future version, it will default to False. In
addition, specifying 'numeric_only=None' is deprecated. Select only
valid columns or specify the value of numeric_only to silence this
warning.
  df.median()
```

```
Age                         36.0
DailyRate                  802.0
DistanceFromHome             7.0
Education                    3.0
EmployeeCount                1.0
EmployeeNumber            1020.5
EnvironmentSatisfaction      3.0
HourlyRate                  66.0
JobInvolvement               3.0
JobLevel                     2.0
JobSatisfaction              3.0
MonthlyIncome             4913.5
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
```

```
df['YearsAtCompany']=np.where(df['YearsAtCompany']>upper_limit,5,df['Y
earsAtCompany'])
```

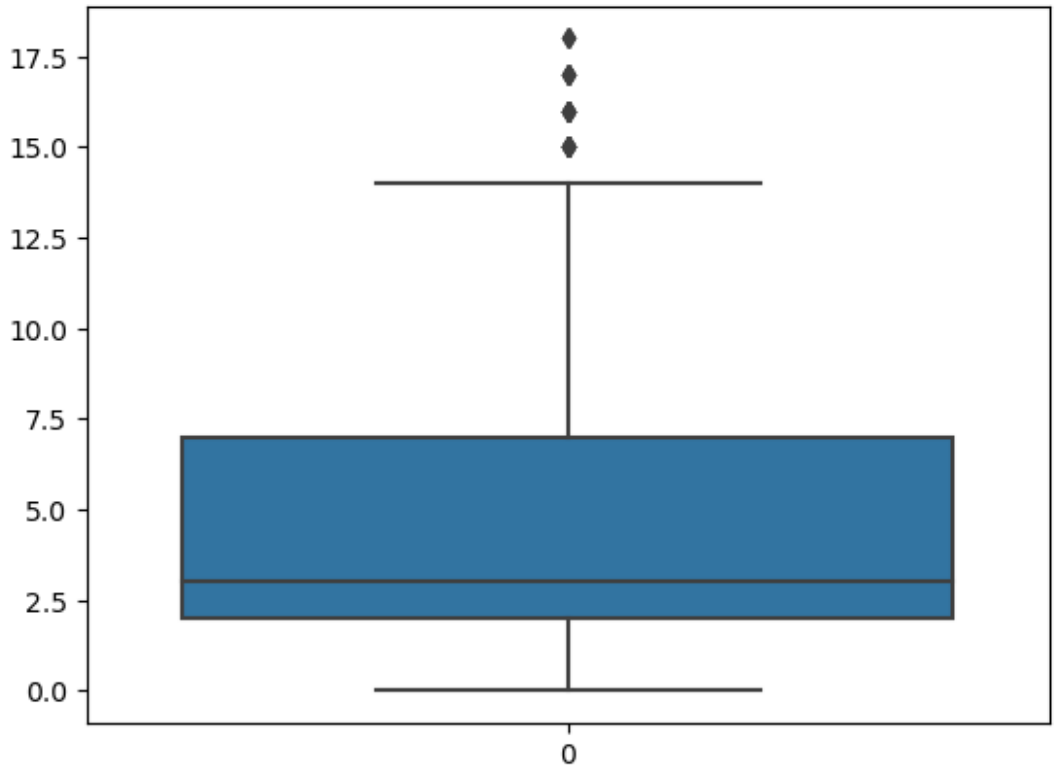
```
sns.boxplot(df["YearsAtCompany"])
```

```
<Axes: >
```




```
sns.boxplot(df["YearsInCurrentRole"])
```

```
<Axes: >
```




```
# Outlier removal by replacement with median
q1=df.YearsInCurrentRole.quantile(0.25)
q3=df.YearsInCurrentRole.quantile(0.75)

q1

2.0

q3

7.0

IQR=q3-q1
IQR

5.0

upper_limit=q3+1.5*IQR
upper_limit

14.5

lower_limit=q1-1.5*IQR
lower_limit

-5.5

df.median()
```

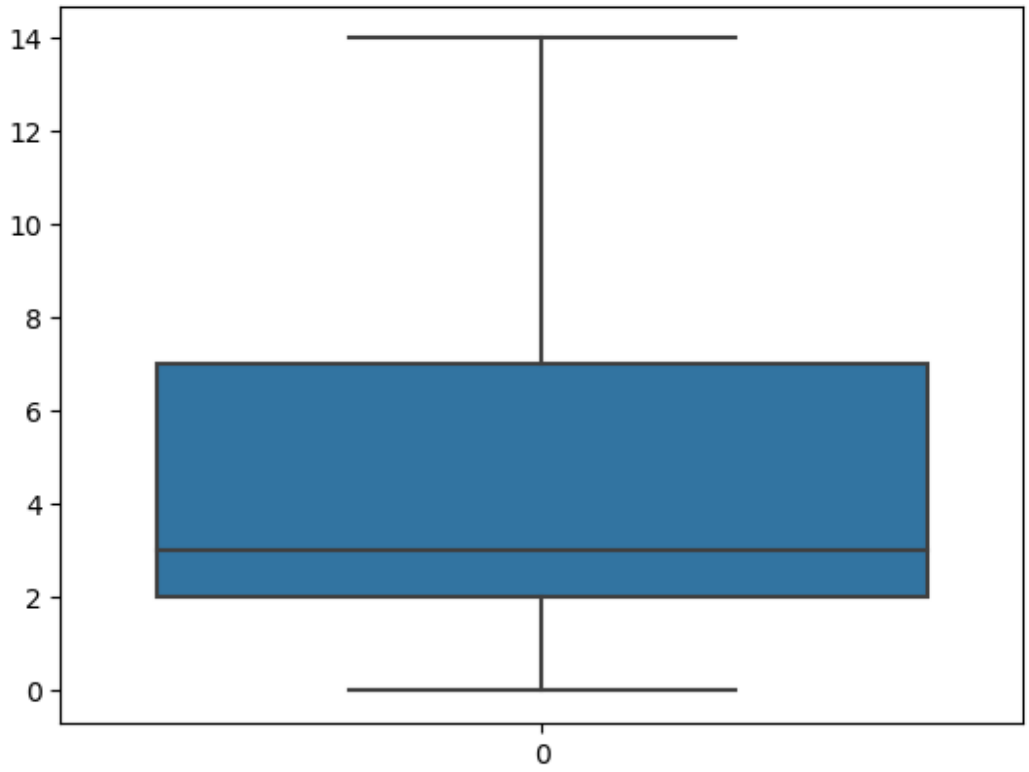
```
C:\Users\DELL\AppData\Local\Temp\ipykernel_22072\530051474.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median
is deprecated. In a future version, it will default to False. In
addition, specifying 'numeric_only=None' is deprecated. Select only
valid columns or specify the value of numeric_only to silence this
warning.
  df.median()
```

```
Age                          36.0
DailyRate                   802.0
DistanceFromHome              7.0
Education                     3.0
EmployeeCount                 1.0
EmployeeNumber             1020.5
EnvironmentSatisfaction       3.0
HourlyRate                   66.0
JobInvolvement                3.0
JobLevel                      2.0
JobSatisfaction               3.0
MonthlyIncome              4913.5
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
```

```
df['YearsInCurrentRole']=np.where(df['YearsInCurrentRole']>upper_limit
,3,df['YearsInCurrentRole'])
```

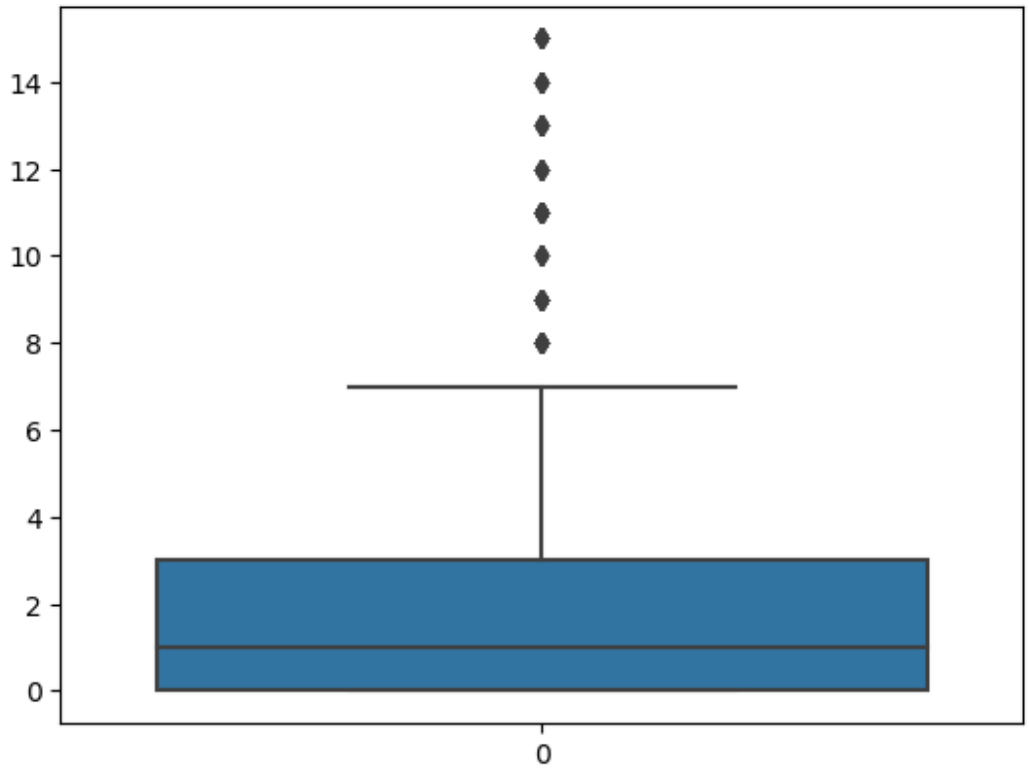
```
sns.boxplot(df["YearsInCurrentRole"])
```

```
<Axes: >
```




```
sns.boxplot(df["YearsSinceLastPromotion"])
```

```
<Axes: >
```




```
# Outlier removal by replacement with median
q1=df.YearsSinceLastPromotion.quantile(0.25)
q3=df.YearsSinceLastPromotion.quantile(0.75)

q1
```

```
0.0
```

```
q3
```

```
3.0
```

```
IQR=q3-q1
IQR
```

```
3.0
```

```
upper_limit=q3+1.5*IQR
upper_limit
```

```
7.5
```

```
lower_limit=q1-1.5*IQR
lower_limit
```

```
-4.5
```

```
df.median()
```

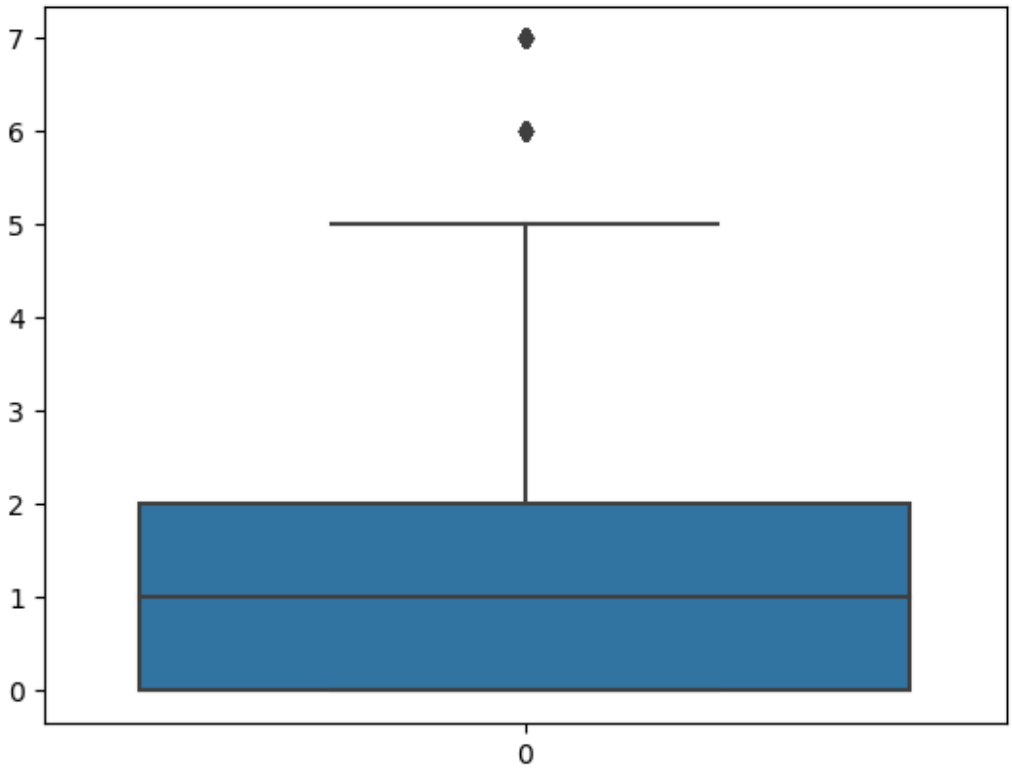
```
C:\Users\DELL\AppData\Local\Temp\ipykernel_22072\530051474.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median
is deprecated. In a future version, it will default to False. In
addition, specifying 'numeric_only=None' is deprecated. Select only
valid columns or specify the value of numeric_only to silence this
warning.
  df.median()
```

```
Age                         36.0
DailyRate                  802.0
DistanceFromHome             7.0
Education                    3.0
EmployeeCount                1.0
EmployeeNumber            1020.5
EnvironmentSatisfaction      3.0
HourlyRate                  66.0
JobInvolvement               3.0
JobLevel                     2.0
JobSatisfaction              3.0
MonthlyIncome             4913.5
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
```

```
df['YearsSinceLastPromotion']=np.where(df['YearsSinceLastPromotion']>u
pper_limit,1,df['YearsSinceLastPromotion'])
```

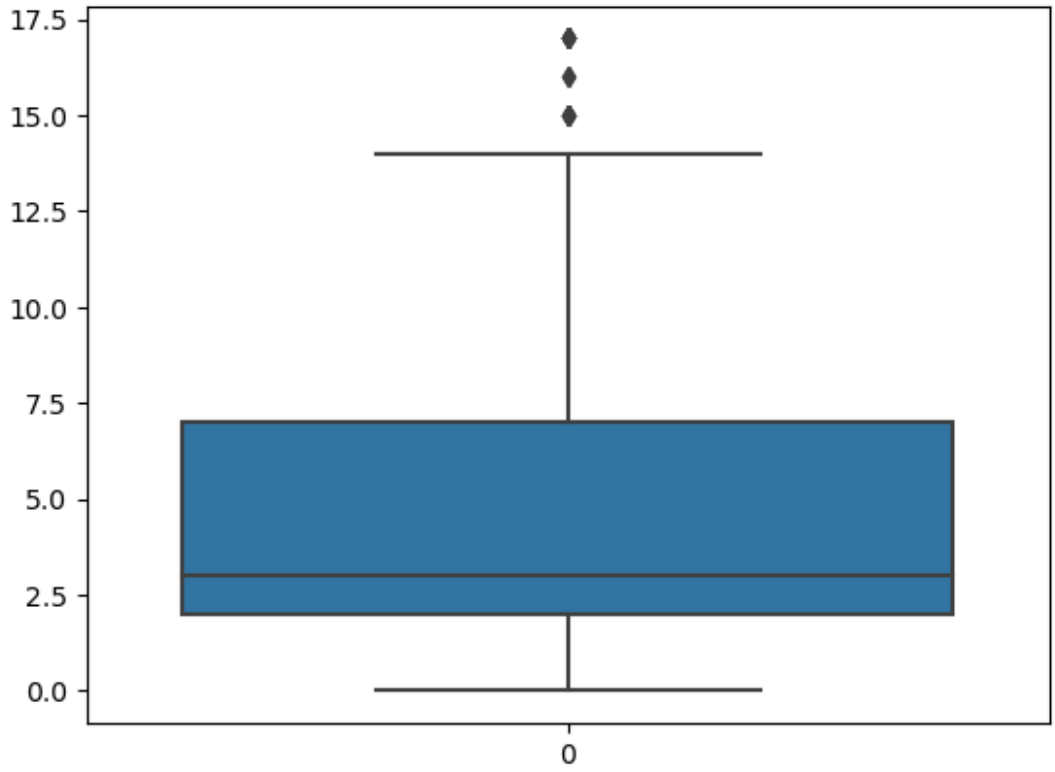
```
sns.boxplot(df["YearsSinceLastPromotion"])
```

```
<Axes: >
```




```
sns.boxplot(df["YearsWithCurrManager"])
```

```
<Axes: >
```




```
# Outlier removal by replacement with median
q1=df.YearsWithCurrManager.quantile(0.25)
q3=df.YearsWithCurrManager.quantile(0.75)

q1
```

```
2.0
```

```
q3
```

```
7.0
```

```
IQR=q3-q1

IQR
```

```
5.0
```

```
upper_limit=q3+1.5*IQR
upper_limit
```

```
14.5
```

```
lower_limit=q1-1.5*IQR
lower_limit
```

```
-5.5
```

```
df.median()
```

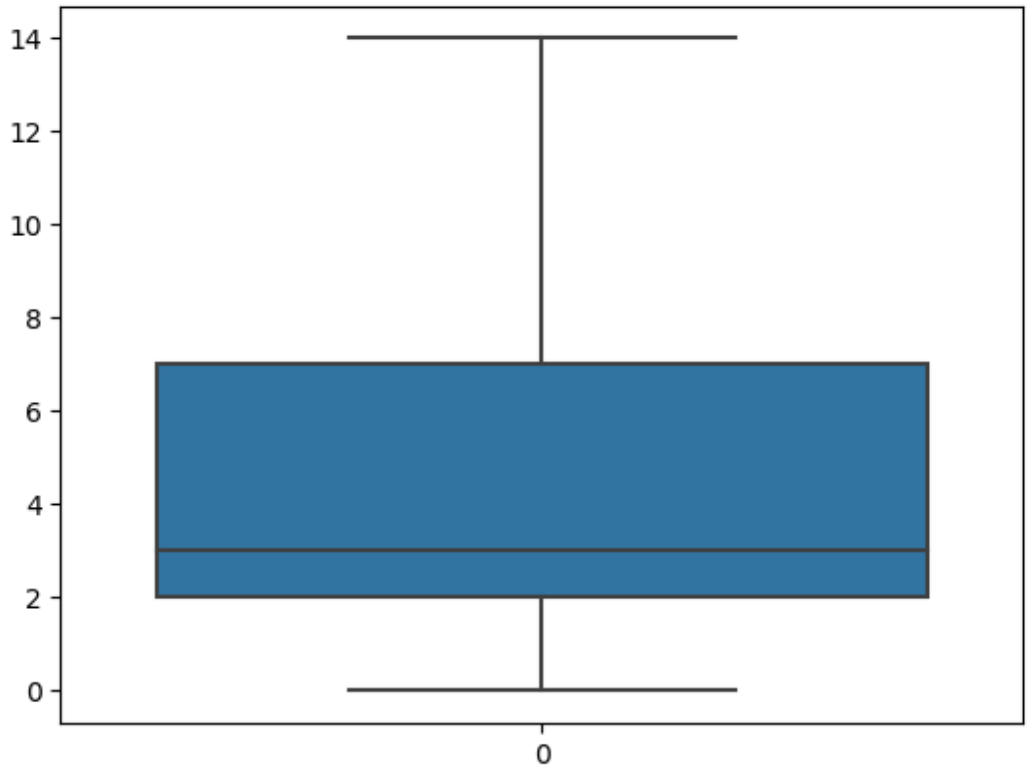
```
C:\Users\DELL\AppData\Local\Temp\ipykernel_22072\530051474.py:1:
FutureWarning: The default value of numeric_only in DataFrame.median
is deprecated. In a future version, it will default to False. In
addition, specifying 'numeric_only=None' is deprecated. Select only
valid columns or specify the value of numeric_only to silence this
warning.
  df.median()
```

```
Age                         36.0
DailyRate                  802.0
DistanceFromHome             7.0
Education                    3.0
EmployeeCount                1.0
EmployeeNumber            1020.5
EnvironmentSatisfaction      3.0
HourlyRate                  66.0
JobInvolvement               3.0
JobLevel                     2.0
JobSatisfaction              3.0
MonthlyIncome             4913.5
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
```

```
df['YearsWithCurrManager']=np.where(df['YearsWithCurrManager']>upper_l
imit,3,df['YearsWithCurrManager'])
```

```
sns.boxplot(df["YearsWithCurrManager"])
```

```
<Axes: >
```




```
# Removing unnecessary columns from the dataset
df=df.drop(['BusinessTravel','DailyRate','Department','DistanceFromHom
e','EducationField','HourlyRate','MonthlyRate','NumCompaniesWorked','O
verTime','PercentSalaryHike','RelationshipSatisfaction','StandardHours
','StockOptionLevel','YearsInCurrentRole','YearsSinceLastPromotion','E
mployeeNumber','PerformanceRating','EmployeeCount','Over18'],axis=1)
df.head()
```

```
   Age Attrition  Education  EnvironmentSatisfaction  Gender
JobInvolvement  \
0   41       Yes          2                        2  Female
3
1   49        No          1                        3    Male
2
2   37       Yes          2                        4    Male
2
3   33        No          4                        4  Female
3
4   27        No          1                        1    Male
3

   JobLevel                 JobRole  JobSatisfaction MaritalStatus  \
0         2       Sales Executive                4        Single
1         2    Research Scientist                2       Married
2         1  Laboratory Technician                3        Single
```

```
3          1     Research Scientist                 3       Married
4          1  Laboratory Technician                 2       Married

   MonthlyIncome  TotalWorkingYears  TrainingTimesLastYear
WorkLifeBalance  \
0           5993                  8                      0
1
1           5130                 10                      3
3
2           2090                  7                      3
3
3           2909                  8                      3
3
4           3468                  6                      3
3

   YearsAtCompany  YearsWithCurrManager
0               6                     5
1              10                     7
2               0                     0
3               8                     0
4               2                     2
```

```
df.shape
```

```
(1470, 16)
```

```
# Independent variables should be 2d array or dataframe
X=df.drop(columns=["Attrition"],axis=1)
X.head()
```

```
   Age  Education  EnvironmentSatisfaction  Gender  JobInvolvement
JobLevel  \
0   41          2                        2  Female               3
2
1   49          1                        3    Male               2
2
2   37          2                        4    Male               2
1
3   33          4                        4  Female               3
1
4   27          1                        1    Male               3
1

                  JobRole  JobSatisfaction MaritalStatus  MonthlyIncome
\
0        Sales Executive                4        Single           5993

1     Research Scientist                2       Married           5130

2  Laboratory Technician                3        Single           2090
```

```
3     Research Scientist                  3       Married           2909

4  Laboratory Technician                  2       Married           3468


   TotalWorkingYears  TrainingTimesLastYear  WorkLifeBalance
YearsAtCompany  \
0                  8                      0                1
6
1                 10                      3                3
10
2                  7                      3                3
0
3                  8                      3                3
8
4                  6                      3                3
2

   YearsWithCurrManager
0                     5
1                     7
2                     0
3                     0
4                     2
```

```
X.shape
```

```
(1470, 15)
```

```
type(X)
```

```
pandas.core.frame.DataFrame
```

```
# Dependent variable should be 1d array or series
y=df["Attrition"]
y.head()
```

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

```
y.shape
```

```
(1470,)
```

```
type(y)
```

```
pandas.core.series.Series
```

```
# Encoding
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()

X["Gender"]=le.fit_transform(X["Gender"])
X.head()
```

```
   Age  Education  EnvironmentSatisfaction  Gender  JobInvolvement
JobLevel  \
0   41          2                        2       0               3
2
1   49          1                        3       1               2
2
2   37          2                        4       1               2
1
3   33          4                        4       0               3
1
4   27          1                        1       1               3
1

   JobRole  JobSatisfaction  MaritalStatus  MonthlyIncome
TotalWorkingYears  \
0        7                4              2           5993
8
1        6                2              1           5130
10
2        2                3              2           2090
7
3        6                3              1           2909
8
4        2                2              1           3468
6

   TrainingTimesLastYear  WorkLifeBalance  YearsAtCompany  \
0                      0                1               6
1                      3                3              10
2                      3                3               0
3                      3                3               8
4                      3                3               2

   YearsWithCurrManager
0                     5
1                     7
2                     0
3                     0
4                     2
```

```
print(le.classes_)
```

```
['Female' 'Male']
```

```
mapping=dict(zip(le.classes_,range(len(le.classes_))))
mapping
```

```
{'Female': 0, 'Male': 1}
```

```
X["JobRole"]=le.fit_transform(X["JobRole"].values)
X.head()
```

```
   Age  Education  EnvironmentSatisfaction  Gender  JobInvolvement
JobLevel  \
0   41          2                        2       0               3
2
1   49          1                        3       1               2
2
2   37          2                        4       1               2
1
3   33          4                        4       0               3
1
4   27          1                        1       1               3
1

   JobRole  JobSatisfaction  MaritalStatus  MonthlyIncome
TotalWorkingYears  \
0        7                4              2           5993
8
1        6                2              1           5130
10
2        2                3              2           2090
7
3        6                3              1           2909
8
4        2                2              1           3468
6

   TrainingTimesLastYear  WorkLifeBalance  YearsAtCompany  \
0                      0                1               6
1                      3                3              10
2                      3                3               0
3                      3                3               8
4                      3                3               2

   YearsWithCurrManager
0                     5
1                     7
2                     0
3                     0
4                     2
```

```
print(le.classes_)
```

```
['Healthcare Representative' 'Human Resources' 'Laboratory Technician'
 'Manager' 'Manufacturing Director' 'Research Director'
 'Research Scientist' 'Sales Executive' 'Sales Representative']
```

```
mapping=dict(zip(le.classes_,range(len(le.classes_))))
mapping
```

```
{'Healthcare Representative': 0,
 'Human Resources': 1,
 'Laboratory Technician': 2,
 'Manager': 3,
 'Manufacturing Director': 4,
 'Research Director': 5,
 'Research Scientist': 6,
 'Sales Executive': 7,
 'Sales Representative': 8}
```

```
X["MaritalStatus"]=le.fit_transform(X["MaritalStatus"])
X.head()
```

```
   Age  Education  EnvironmentSatisfaction  Gender  JobInvolvement
JobLevel  \
0   41          2                        2       0               3
2
1   49          1                        3       1               2
2
2   37          2                        4       1               2
1
3   33          4                        4       0               3
1
4   27          1                        1       1               3
1

   JobRole  JobSatisfaction  MaritalStatus  MonthlyIncome
TotalWorkingYears  \
0        7                4              2           5993
8
1        6                2              1           5130
10
2        2                3              2           2090
7
3        6                3              1           2909
8
4        2                2              1           3468
6

   TrainingTimesLastYear  WorkLifeBalance  YearsAtCompany  \
0                      0                1               6
1                      3                3              10
2                      3                3               0
```

```
3                       3                  3                 8
4                       3                  3                 2

   YearsWithCurrManager
0                     5
1                     7
2                     0
3                     0
4                     2
```

```
print(le.classes_)
```

```
['Divorced' 'Married' 'Single']
```

```
mapping=dict(zip(le.classes_,range(len(le.classes_))))
mapping
```

```
{'Divorced': 0, 'Married': 1, 'Single': 2}
```

```
# Feature scaling
from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
```

```
X_Scaled=pd.DataFrame(ms.fit_transform(X),columns=X.columns)
X_Scaled.head()
```

```
        Age  Education  EnvironmentSatisfaction  Gender
JobInvolvement  \
0  0.547619       0.25                 0.333333     0.0
0.666667
1  0.738095       0.00                 0.666667     1.0
0.333333
2  0.452381       0.25                 1.000000     1.0
0.333333
3  0.357143       0.75                 1.000000     0.0
0.666667
4  0.214286       0.00                 0.000000     1.0
0.666667

   JobLevel  JobRole  JobSatisfaction  MaritalStatus  MonthlyIncome  \
0      0.25    0.875         1.000000            1.0       0.320597
1      0.25    0.750         0.333333            0.5       0.265084
2      0.00    0.250         0.666667            1.0       0.069536
3      0.00    0.750         0.666667            0.5       0.122218
4      0.00    0.250         0.333333            0.5       0.158176

   TotalWorkingYears  TrainingTimesLastYear  WorkLifeBalance
YearsAtCompany  \
0           0.285714                   0.00         0.000000
0.333333
1           0.357143                   0.75         0.666667
```

```
0.555556
2           0.250000                    0.75         0.666667
0.000000
3           0.285714                    0.75         0.666667
0.444444
4           0.214286                    0.75         0.666667
0.111111

   YearsWithCurrManager
0              0.357143
1              0.500000
2              0.000000
3              0.000000
4              0.142857
```

```
# Splitting Data into Train and Test
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.2,rando
m_state=2)

print(x_train.shape,x_test.shape,y_train.shape,y_test.shape)
```

```
(1176, 15) (294, 15) (1176,) (294,)
```

```
y_train=le.fit_transform(y_train)
y_test=le.transform(y_test)

x_train
```

```
      Age  Education  EnvironmentSatisfaction  Gender  JobInvolvement
\
285    37          3                        4       0               3

194    45          2                        1       1               2

323    28          4                        1       1               1

1015   34          4                        4       1               3

1003   25          3                        1       1               3

...   ...        ...                      ...     ...             ...

466    41          5                        2       0               3

299    51          3                        4       1               1

493    44          4                        1       0               2

527    32          3                        4       1               3

1192   49          3                        4       0               3
```

```
      JobLevel  JobRole  JobSatisfaction  MaritalStatus  MonthlyIncome
\
285          1        6                4              2           2115

194          4        3                4              1           4919

323          1        6                4              1           3464

1015         1        6                1              0           2996

1003         1        2                4              1           3229

...        ...      ...              ...            ...            ...

466          4        3                1              1           4919

299          2        4                2              0           5482

493          2        1                3              2           5985

527          2        7                4              2           5396

1192         1        2                1              0           2587


      TotalWorkingYears  TrainingTimesLastYear  WorkLifeBalance  \
285                  17                      3                3
194                  22                      1                3
323                   5                      4                2
1015                 10                      2                3
1003                  7                      2                2
...                 ...                    ...              ...
466                  22                      2                3
299                  13                      3                3
493                  10                      1                4
527                  10                      2                2
1192                 17                      2                2

      YearsAtCompany  YearsWithCurrManager
285               17                     7
194                5                     8
323                3                     2
1015               4                     3
1003               3                     2
...              ...                   ...
466               18                     8
299                4                     2
493                2                     2
527               10                     8
```

```
1192               2                       2

[1176 rows x 15 columns]

x_test

      Age  Education  EnvironmentSatisfaction  Gender  JobInvolvement
\
721    50          3                        4       1               3

843    26          4                        1       1               4

627    52          4                        3       0               2

1368   34          4                        3       1               2

305    36          4                        2       0               3

...   ...        ...                      ...     ...             ...

61     38          5                        4       0               3

498    22          1                        1       1               3

993    25          1                        1       1               4

308    58          4                        4       1               1

400    39          1                        2       1               3


      JobLevel  JobRole  JobSatisfaction  MaritalStatus  MonthlyIncome
\
721          4        4                3              1          13973

843          1        2                4              1           4420

627          4        4                4              1          13826

1368         2        6                4              1           5747

305          2        2                2              1           5674

...        ...      ...              ...            ...            ...

61           2        2                4              2           2406

498          1        6                3              1           2773

993          2        7                3              1           6232

308          2        0                3              0           5660
```

```
400          5        3                  3               0           4919


      TotalWorkingYears  TrainingTimesLastYear  WorkLifeBalance  \
721                  22                      2                3
843                   8                      2                3
627                  10                      3                3
1368                 16                      3                3
305                  11                      3                3
...                 ...                    ...              ...
61                   10                      2                3
498                   3                      3                3
993                   6                      3                2
308                  12                      2                3
400                  21                      3                3

      YearsAtCompany  YearsWithCurrManager
721               12                     5
843                8                     7
627                9                     0
1368              15                    11
305                9                     8
...              ...                   ...
61                10                     9
498                2                     2
993                3                     2
308                5                     2
400                5                     6

[294 rows x 15 columns]

y_train

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

y_test

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

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

```
# Import model building libraries
from sklearn.linear_model import LogisticRegression

# Initialising the model
model=LogisticRegression()

# Training the model
model.fit(x_train,y_train)
```

```
C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\linear_model\
_logistic.py:460: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

```
LogisticRegression()
```

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

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

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

```
y_test
```

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

```
df

      Age Attrition  Education  EnvironmentSatisfaction  Gender  \
0      41       Yes          2                        2  Female
1      49        No          1                        3    Male
2      37       Yes          2                        4    Male
3      33        No          4                        4  Female
4      27        No          1                        1    Male
...   ...       ...        ...                      ...     ...
1465   36        No          2                        3    Male
1466   39        No          1                        4    Male
1467   27        No          3                        2    Male
1468   49        No          3                        4    Male
1469   34        No          3                        2    Male

      JobInvolvement  JobLevel                    JobRole
JobSatisfaction  \
0                  3         2            Sales Executive
4
1                  2         2         Research Scientist
2
2                  2         1      Laboratory Technician
3
3                  3         1         Research Scientist
3
4                  3         1      Laboratory Technician
2
...              ...       ...                        ...
...
1465               4         2      Laboratory Technician
4
1466               2         3  Healthcare Representative
1
1467               4         2     Manufacturing Director
2
1468               2         2            Sales Executive
2
1469               4         2      Laboratory Technician
3

     MaritalStatus  MonthlyIncome  TotalWorkingYears
TrainingTimesLastYear  \
0           Single           5993                  8
0
1          Married           5130                 10
3
2           Single           2090                  7
3
3          Married           2909                  8
3
```

```
4          Married           3468                    6
3
...            ...            ...                  ...
...
1465       Married           2571                   17
3
1466       Married           9991                    9
3
1467       Married           6142                    6
0
1468       Married           5390                   17
3
1469       Married           4404                    6
3
```

```
      WorkLifeBalance  YearsAtCompany  YearsWithCurrManager
0                   1               6                     5
1                   3              10                     7
2                   3               0                     0
3                   3               8                     0
4                   3               2                     2
...               ...             ...                   ...
1465                3               5                     3
1466                3               7                     7
1467                3               6                     3
1468                2               9                     8
1469                4               4                     2
```

```
[1470 rows x 16 columns]
```

```
model.predict(ms.transform([[49,1,3,1,2,2,6,2,1,5130,10,3,3,10,7]]))
```

```
C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\base.py:464:
UserWarning: X does not have valid feature names, but MinMaxScaler was
fitted with feature names
  warnings.warn(
C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\base.py:464:
UserWarning: X does not have valid feature names, but
LogisticRegression was fitted with feature names
  warnings.warn(
```

```
array([0])
```

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

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

```
0.8435374149659864
```

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

```
array([[246,   0],
       [ 46,   2]], dtype=int64)
```

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

```
col_0    0  1
row_0
0      246  0
1       46  2
```

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

```
              precision    recall  f1-score   support

          No       0.88      1.00      0.93       255
         Yes       0.80      0.10      0.18        39

    accuracy                           0.88       294
   macro avg       0.84      0.55      0.56       294
weighted avg       0.87      0.88      0.83       294
```

Performance metrics

```
# Accuracy
# Accuracy=(TP+TN)/(TP+TN+FP+FN)
(246+2)/(246+2+0+46)
```

```
0.8435374149659864
```

```
# Precision=TP/(TP+FP)
(246)/(246+0)
```

```
1.0
```

```
# Recall=TP/(FN+TP)
(246)/(246+46)
```

```
0.8424657534246576
```

```
# F1 score=2*Precision*Recall/(Precision+Recall)
(2*(1.0)*(0.8424657534246576))/(1.0+0.8424657534246576)
```

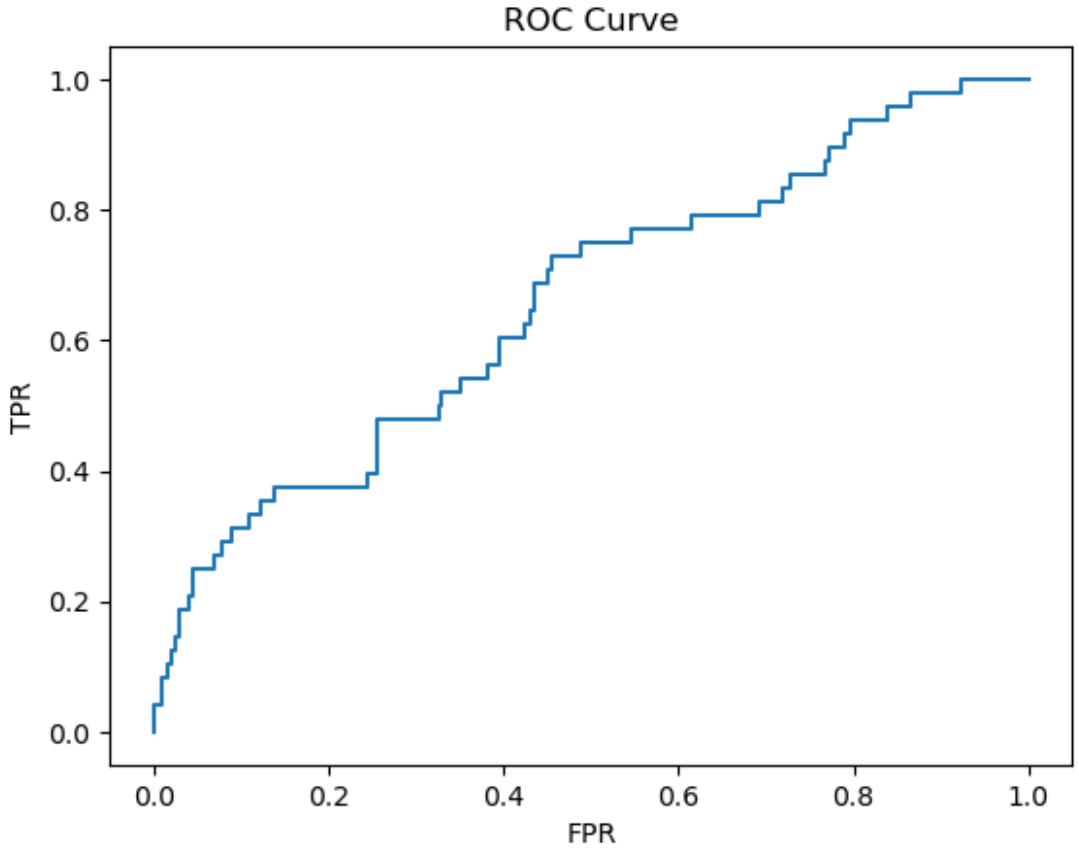
```
0.9144981412639406
```

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

```
array([0.02403142, 0.09445797, 0.07654444, 0.06064433, 0.10148443,
       0.34233309, 0.12725673, 0.06919687, 0.20709283, 0.11372964,
       0.08314631, 0.0412354 , 0.06491955, 0.08394682, 0.26297851,
       0.06484908, 0.01720105, 0.02589436, 0.09137199, 0.15075918,
       0.04507812, 0.01855805, 0.09960144, 0.2834942 , 0.07949573,
       0.0808416 , 0.18105098, 0.05390051, 0.18111516, 0.3793593 ,
       0.07549996, 0.33250979, 0.20368194, 0.01494276, 0.23136722,
       0.07205727, 0.04056071, 0.08960441, 0.28826796, 0.13931592,
       0.2396192 , 0.1205202 , 0.17424522, 0.10245817, 0.19144624,
       0.14573917, 0.50077028, 0.09737055, 0.06134597, 0.07695119,
       0.0646936 , 0.11396589, 0.16340508, 0.05094316, 0.21571305,
       0.30228028, 0.15815536, 0.15091795, 0.05766565, 0.19820098,
       0.14276109, 0.107298  , 0.11492072, 0.15108016, 0.0645181 ,
       0.1781897 , 0.15089305, 0.4067094 , 0.30635655, 0.18177287,
       0.09397118, 0.09367974, 0.07183057, 0.03706485, 0.10186304,
       0.18325816, 0.16806538, 0.2040328 , 0.13649306, 0.12886518,
       0.29610066, 0.05066106, 0.03524866, 0.12629425, 0.02899862,
       0.03930277, 0.02380177, 0.02392398, 0.14225553, 0.06476465,
       0.13760121, 0.33976273, 0.08409621, 0.04104775, 0.04285854,
       0.02990783, 0.34096974, 0.22464599, 0.36035625, 0.09197099,
       0.14054254, 0.1372768 , 0.05689064, 0.19127233, 0.15124034,
       0.03901433, 0.08072146, 0.35887245, 0.19090518, 0.20490332,
       0.10069357, 0.11571031, 0.30199654, 0.1347411 , 0.12824171,
       0.18982858, 0.14856528, 0.15252271, 0.01170449, 0.0601033 ,
       0.38493219, 0.20774456, 0.12504444, 0.3381428 , 0.16350242,
       0.03927928, 0.05686355, 0.2587833 , 0.10649944, 0.10170878,
       0.10833344, 0.38775056, 0.09239781, 0.37281429, 0.03513365,
       0.02552965, 0.07041695, 0.23165476, 0.06781167, 0.36567094,
       0.20117603, 0.09156185, 0.03276371, 0.09724294, 0.29253649,
       0.19110933, 0.09339161, 0.06718603, 0.11092964, 0.08738097,
       0.2339715 , 0.11519131, 0.27433358, 0.18746237, 0.16424103,
       0.19528037, 0.19277357, 0.24328856, 0.0588372 , 0.16854064,
       0.13578893, 0.35533901, 0.13716653, 0.07105806, 0.17597616,
       0.21227275, 0.06419271, 0.08756476, 0.0754363 , 0.13953337,
       0.41682252, 0.15939375, 0.12607455, 0.04256013, 0.09707126,
       0.05237973, 0.03431458, 0.04500934, 0.40358837, 0.25899091,
       0.12323135, 0.06434512, 0.07975735, 0.11425787, 0.3599286 ,
       0.16382687, 0.11681062, 0.16060263, 0.0825488 , 0.35856775,
       0.07503177, 0.03395941, 0.03786847, 0.0294548 , 0.14796439,
       0.27499469, 0.08194015, 0.17106979, 0.0838629 , 0.14009019,
       0.16257525, 0.2229384 , 0.17729351, 0.02684841, 0.1028664 ,
       0.19742187, 0.16483514, 0.12884445, 0.26476188, 0.06584164,
       0.27421741, 0.04377848, 0.11380416, 0.07227381, 0.01189707,
       0.06546546, 0.12970361, 0.15420974, 0.13219821, 0.13522648,
       0.16631532, 0.06052613, 0.18504565, 0.04236855, 0.05633362,
       0.06211668, 0.22205859, 0.02277011, 0.06825874, 0.09254447,
       0.24516644, 0.03359641, 0.07843682, 0.09505613, 0.13138298,
       0.3504319 , 0.29010163, 0.03918853, 0.1290916 , 0.31060748,
       0.0289438 , 0.21718332, 0.14321047, 0.11451947, 0.10250799,
       0.25463103, 0.16697332, 0.21764557, 0.24706491, 0.25473136,
```

```
       0.14737471, 0.22454757, 0.12766487, 0.14611059, 0.08291559,
       0.32673455, 0.0595181 , 0.21761503, 0.13254516, 0.13766186,
       0.09249018, 0.1889244 , 0.25118676, 0.05661062, 0.10480195,
       0.0431618 , 0.18729946, 0.14917859, 0.23735463, 0.07257554,
       0.0356213 , 0.03261861, 0.08875539, 0.0896364 , 0.25198156,
       0.13954989, 0.06516022, 0.32069518, 0.13145889, 0.18394341,
       0.09020818, 0.26266061, 0.11798205, 0.12693783, 0.50313143,
       0.03670132, 0.13929559, 0.11870062, 0.19477779, 0.05859857,
       0.29327098, 0.25157755, 0.05271322, 0.0330197 ])
```

```
# ROC curve
fpr,tpr,thresholds=roc_curve(y_test,probability)

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




```
from sklearn.metrics import roc_curve, roc_auc_score

auc=roc_auc_score(y_test,probability)
```

```
print('AUC-ROC score:', auc)
```

```
AUC-ROC score: 0.6566734417344173
```

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

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

```
DecisionTreeClassifier()
```

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

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

```
y_test
```

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

```
df.head()
```

```
   Age Attrition  Education  EnvironmentSatisfaction  Gender
JobInvolvement  \
0   41       Yes          2                        2  Female
3
1   49        No          1                        3    Male
2
2   37       Yes          2                        4    Male
2
3   33        No          4                        4  Female
3
4   27        No          1                        1    Male
3

   JobLevel                JobRole  JobSatisfaction MaritalStatus  \
0         2        Sales Executive                4        Single
1         2     Research Scientist                2       Married
2         1  Laboratory Technician                3        Single
3         1     Research Scientist                3       Married
4         1  Laboratory Technician                2       Married

   MonthlyIncome  TotalWorkingYears  TrainingTimesLastYear
WorkLifeBalance  \
0           5993                  8                      0
1
1           5130                 10                      3
3
2           2090                  7                      3
```

```
3
3           2909                    8                       3
3
4           3468                    6                       3
3
```

```
   YearsAtCompany  YearsWithCurrManager
0               6                     5
1              10                     7
2               0                     0
3               8                     0
4               2                     2
```

```
dtc.predict(ms.transform([[37,2,4,1,2,1,2,3,2,2090,7,3,3,0,0]]))
```

```
C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\base.py:464:
UserWarning: X does not have valid feature names, but MinMaxScaler was
fitted with feature names
  warnings.warn(
C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\base.py:464:
UserWarning: X does not have valid feature names, but
DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

```
array([1])
```

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

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

```
0.7551020408163265
```

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

```
array([[211,  35],
       [ 37,  11]], dtype=int64)
```

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

```
col_0    0   1
row_0
0      211  35
1       37  11
```

```
# Accuracy=(TP+TN)/(TP+TN+FP+FN)
(211+11)/(211+11+35+37)
```

```
0.7551020408163265
```

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

```
              precision    recall  f1-score   support

           0       0.85      0.86      0.85       246
           1       0.24      0.23      0.23        48

    accuracy                           0.76       294
   macro avg       0.54      0.54      0.54       294
weighted avg       0.75      0.76      0.75       294
```

```
# Precision=TP/(TP+FP)
(211)/(211+35)
```

```
0.8577235772357723
```

```
# Recall=TP/(TP+FN)
(211)/(211+37)
```

```
0.8508064516129032
```

```
# F1 score=2*Precision*Recall/(Precision+Recall)
(2*(0.8577235772357723)*(0.8508064516129032))/(0.8577235772357723+0.85
08064516129032)
```

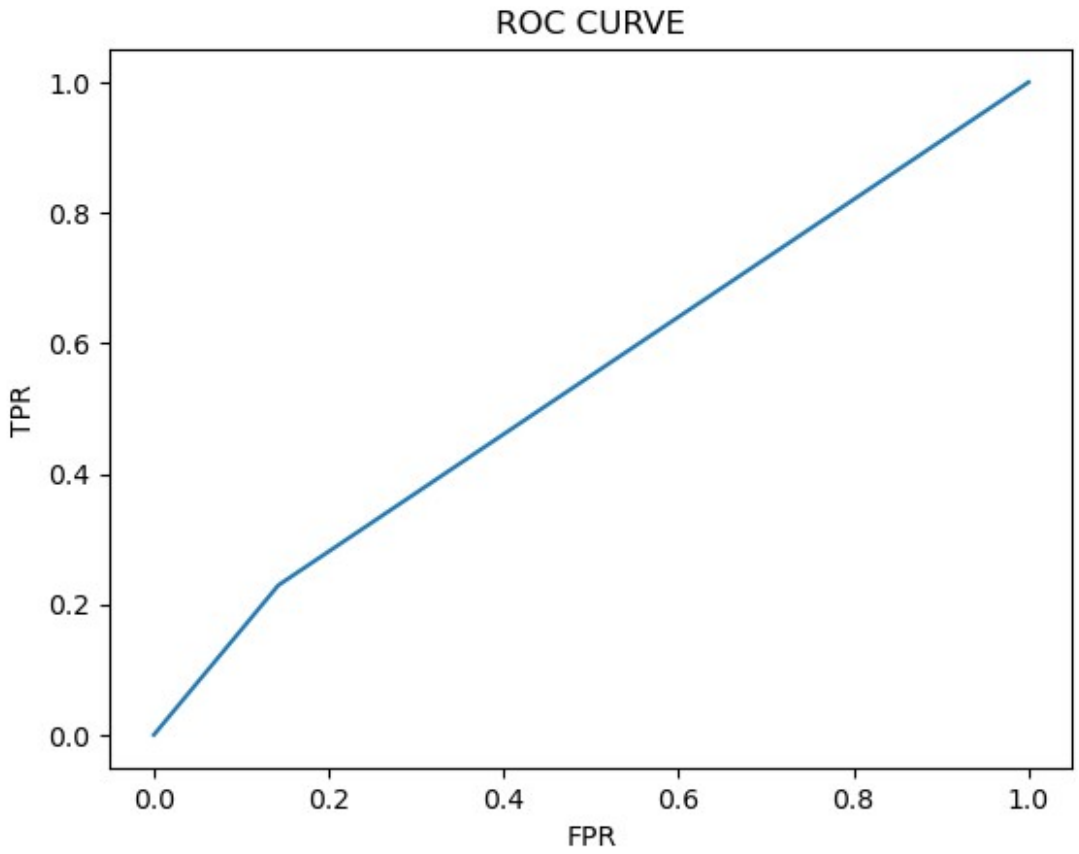
```
0.854251012145749
```

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

```
probability
```

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

# ROC-Curve
fpr,tpr,threshsholds = roc_curve(y_test,probability)

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




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

```
[Text(0.31082099780701755, 0.9722222222222222, 'x[10] <= 1.5\ngini =
0.27\nsamples = 1176\nvalue = [987, 189]'),
 Text(0.06871345029239766, 0.9166666666666666, 'x[0] <= 32.5\ngini =
0.5\nsamples = 73\nvalue = [37, 36]'),
 Text(0.043859649122807015, 0.8611111111111112, 'x[6] <= 4.0\ngini =
0.489\nsamples = 61\nvalue = [26, 35]'),
 Text(0.011695906432748537, 0.8055555555555556, 'x[11] <= 2.5\ngini =
0.34\nsamples = 23\nvalue = [5, 18]'),
 Text(0.005847953216374269, 0.75, 'gini = 0.0\nsamples = 12\nvalue =
[0, 12]'),
 Text(0.017543859649122806, 0.75, 'x[12] <= 2.5\ngini = 0.496\nsamples
= 11\nvalue = [5, 6]'),
 Text(0.011695906432748537, 0.6944444444444444, 'gini = 0.0\nsamples =
4\nvalue = [0, 4]'),
 Text(0.023391812865497075, 0.6944444444444444, 'x[3] <= 0.5\ngini =
0.408\nsamples = 7\nvalue = [5, 2]'),
 Text(0.017543859649122806, 0.6388888888888888, 'gini = 0.0\nsamples =
4\nvalue = [4, 0]'),
```

```
 Text(0.029239766081871343, 0.6388888888888888, 'x[2] <= 3.5\ngini =
0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.023391812865497075, 0.5833333333333334, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.03508771929824561, 0.5833333333333334, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.07602339181286549, 0.8055555555555556, 'x[8] <= 1.5\ngini =
0.494\nsamples = 38\nvalue = [21, 17]'),
 Text(0.06432748538011696, 0.75, 'x[1] <= 3.5\ngini = 0.346\nsamples =
18\nvalue = [14, 4]'),
 Text(0.05847953216374269, 0.6944444444444444, 'x[0] <= 25.5\ngini =
0.219\nsamples = 16\nvalue = [14, 2]'),
 Text(0.05263157894736842, 0.6388888888888888, 'x[0] <= 24.5\ngini =
0.5\nsamples = 4\nvalue = [2, 2]'),
 Text(0.04678362573099415, 0.5833333333333334, 'x[7] <= 1.5\ngini =
0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.04093567251461988, 0.5277777777777778, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.05263157894736842, 0.5277777777777778, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.05847953216374269, 0.5833333333333334, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.06432748538011696, 0.6388888888888888, 'gini = 0.0\nsamples =
12\nvalue = [12, 0]'),
 Text(0.07017543859649122, 0.6944444444444444, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.08771929824561403, 0.75, 'x[1] <= 1.5\ngini = 0.455\nsamples =
20\nvalue = [7, 13]'),
 Text(0.08187134502923976, 0.6944444444444444, 'gini = 0.0\nsamples =
5\nvalue = [0, 5]'),
 Text(0.0935672514619883, 0.6944444444444444, 'x[12] <= 3.5\ngini =
0.498\nsamples = 15\nvalue = [7, 8]'),
 Text(0.08771929824561403, 0.6388888888888888, 'x[6] <= 7.0\ngini =
0.486\nsamples = 12\nvalue = [7, 5]'),
 Text(0.07602339181286549, 0.5833333333333334, 'x[4] <= 2.5\ngini =
0.278\nsamples = 6\nvalue = [5, 1]'),
 Text(0.07017543859649122, 0.5277777777777778, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.08187134502923976, 0.5277777777777778, 'gini = 0.0\nsamples =
5\nvalue = [5, 0]'),
 Text(0.09941520467836257, 0.5833333333333334, 'x[2] <= 2.5\ngini =
0.444\nsamples = 6\nvalue = [2, 4]'),
 Text(0.0935672514619883, 0.5277777777777778, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.10526315789473684, 0.5277777777777778, 'x[7] <= 2.0\ngini =
0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.09941520467836257, 0.4722222222222222, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.1111111111111111, 0.4722222222222222, 'gini = 0.0\nsamples =
```

```
2\nvalue = [2, 0]'),
 Text(0.09941520467836257, 0.6388888888888888, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.0935672514619883, 0.8611111111111112, 'x[7] <= 1.5\ngini =
0.153\nsamples = 12\nvalue = [11, 1]'),
 Text(0.08771929824561403, 0.8055555555555556, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.09941520467836257, 0.8055555555555556, 'gini = 0.0\nsamples =
11\nvalue = [11, 0]'),
 Text(0.5529285453216374, 0.9166666666666666, 'x[8] <= 1.5\ngini =
0.239\nsamples = 1103\nvalue = [950, 153]'),
 Text(0.2876918859649123, 0.8611111111111112, 'x[9] <= 2444.0\ngini =
0.184\nsamples = 752\nvalue = [675, 77]'),
 Text(0.1652046783625731, 0.8055555555555556, 'x[14] <= 1.5\ngini =
0.353\nsamples = 83\nvalue = [64, 19]'),
 Text(0.13450292397660818, 0.75, 'x[4] <= 2.5\ngini = 0.5\nsamples =
16\nvalue = [8, 8]'),
 Text(0.12280701754385964, 0.6944444444444444, 'x[9] <= 2301.5\ngini =
0.375\nsamples = 8\nvalue = [2, 6]'),
 Text(0.11695906432748537, 0.6388888888888888, 'x[2] <= 2.0\ngini =
0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.1111111111111111, 0.5833333333333334, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.12280701754385964, 0.5833333333333334, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.1286549707602339, 0.6388888888888888, 'gini = 0.0\nsamples =
5\nvalue = [0, 5]'),
 Text(0.14619883040935672, 0.6944444444444444, 'x[11] <= 2.5\ngini =
0.375\nsamples = 8\nvalue = [6, 2]'),
 Text(0.14035087719298245, 0.6388888888888888, 'x[6] <= 3.5\ngini =
0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.13450292397660818, 0.5833333333333334, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.14619883040935672, 0.5833333333333334, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.15204678362573099, 0.6388888888888888, 'gini = 0.0\nsamples =
5\nvalue = [5, 0]'),
 Text(0.195906432748538, 0.75, 'x[9] <= 2361.0\ngini = 0.274\nsamples
= 67\nvalue = [56, 11]'),
 Text(0.17543859649122806, 0.6944444444444444, 'x[7] <= 2.5\ngini =
0.198\nsamples = 54\nvalue = [48, 6]'),
 Text(0.1695906432748538, 0.6388888888888888, 'x[9] <= 2183.0\ngini =
0.386\nsamples = 23\nvalue = [17, 6]'),
 Text(0.15789473684210525, 0.5833333333333334, 'x[1] <= 3.5\ngini =
0.5\nsamples = 10\nvalue = [5, 5]'),
 Text(0.15204678362573099, 0.5277777777777778, 'x[0] <= 26.5\ngini =
0.408\nsamples = 7\nvalue = [5, 2]'),
 Text(0.14619883040935672, 0.4722222222222222, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
```

```
 Text(0.15789473684210525, 0.4722222222222222, 'gini = 0.0\nsamples =
5\nvalue = [5, 0]'),
 Text(0.16374269005847952, 0.5277777777777778, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.18128654970760233, 0.5833333333333334, 'x[0] <= 42.5\ngini =
0.142\nsamples = 13\nvalue = [12, 1]'),
 Text(0.17543859649122806, 0.5277777777777778, 'gini = 0.0\nsamples =
11\nvalue = [11, 0]'),
 Text(0.1871345029239766, 0.5277777777777778, 'x[13] <= 11.0\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.18128654970760233, 0.4722222222222222, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.19298245614035087, 0.4722222222222222, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.18128654970760233, 0.6388888888888888, 'gini = 0.0\nsamples =
31\nvalue = [31, 0]'),
 Text(0.21637426900584794, 0.6944444444444444, 'x[11] <= 2.5\ngini =
0.473\nsamples = 13\nvalue = [8, 5]'),
 Text(0.21052631578947367, 0.6388888888888888, 'x[13] <= 2.5\ngini =
0.408\nsamples = 7\nvalue = [2, 5]'),
 Text(0.2046783625730994, 0.5833333333333334, 'x[7] <= 2.0\ngini =
0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.19883040935672514, 0.5277777777777778, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.21052631578947367, 0.5277777777777778, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.21637426900584794, 0.5833333333333334, 'gini = 0.0\nsamples =
4\nvalue = [0, 4]'),
 Text(0.2222222222222222, 0.6388888888888888, 'gini = 0.0\nsamples =
6\nvalue = [6, 0]'),
 Text(0.41017909356725146, 0.8055555555555556, 'x[6] <= 6.5\ngini =
0.158\nsamples = 669\nvalue = [611, 58]'),
 Text(0.30646929824561403, 0.75, 'x[10] <= 2.5\ngini = 0.118\nsamples
= 477\nvalue = [447, 30]'),
 Text(0.23976608187134502, 0.6944444444444444, 'x[4] <= 2.5\ngini =
0.49\nsamples = 7\nvalue = [4, 3]'),
 Text(0.23391812865497075, 0.6388888888888888, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
 Text(0.24561403508771928, 0.6388888888888888, 'x[9] <= 2597.5\ngini =
0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.23976608187134502, 0.5833333333333334, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.25146198830409355, 0.5833333333333334, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.37317251461988304, 0.6944444444444444, 'x[2] <= 2.5\ngini =
0.108\nsamples = 470\nvalue = [443, 27]'),
 Text(0.2989766081871345, 0.6388888888888888, 'x[4] <= 1.5\ngini =
0.171\nsamples = 180\nvalue = [163, 17]'),
 Text(0.2631578947368421, 0.5833333333333334, 'x[7] <= 3.5\ngini =
```

```
0.426\nsamples = 13\nvalue = [9, 4]'),
 Text(0.2573099415204678, 0.5277777777777778, 'gini = 0.0\nsamples =
7\nvalue = [7, 0]'),
 Text(0.26900584795321636, 0.5277777777777778, 'x[10] <= 12.5\ngini =
0.444\nsamples = 6\nvalue = [2, 4]'),
 Text(0.2631578947368421, 0.4722222222222222, 'gini = 0.0\nsamples =
4\nvalue = [0, 4]'),
 Text(0.27485380116959063, 0.4722222222222222, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.3347953216374269, 0.5833333333333334, 'x[9] <= 9774.0\ngini =
0.144\nsamples = 167\nvalue = [154, 13]'),
 Text(0.30701754385964913, 0.5277777777777778, 'x[0] <= 55.5\ngini =
0.086\nsamples = 133\nvalue = [127, 6]'),
 Text(0.28654970760233917, 0.4722222222222222, 'x[13] <= 2.5\ngini =
0.073\nsamples = 131\nvalue = [126, 5]'),
 Text(0.2631578947368421, 0.4166666666666667, 'x[12] <= 1.5\ngini =
0.227\nsamples = 23\nvalue = [20, 3]'),
 Text(0.2573099415204678, 0.3611111111111111, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.26900584795321636, 0.3611111111111111, 'x[0] <= 30.0\ngini =
0.165\nsamples = 22\nvalue = [20, 2]'),
 Text(0.2573099415204678, 0.3055555555555556, 'x[7] <= 3.0\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.25146198830409355, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.2631578947368421, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.2807017543859649, 0.3055555555555556, 'x[11] <= 1.5\ngini =
0.095\nsamples = 20\nvalue = [19, 1]'),
 Text(0.27485380116959063, 0.25, 'x[1] <= 2.5\ngini = 0.444\nsamples =
3\nvalue = [2, 1]'),
 Text(0.26900584795321636, 0.19444444444444445, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.2807017543859649, 0.19444444444444445, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.28654970760233917, 0.25, 'gini = 0.0\nsamples = 17\nvalue =
[17, 0]'),
 Text(0.30994152046783624, 0.4166666666666667, 'x[14] <= 0.5\ngini =
0.036\nsamples = 108\nvalue = [106, 2]'),
 Text(0.2982456140350877, 0.3611111111111111, 'x[1] <= 3.0\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.29239766081871343, 0.3055555555555556, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.30409356725146197, 0.3055555555555556, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.3216374269005848, 0.3611111111111111, 'x[9] <= 2716.0\ngini =
0.019\nsamples = 106\nvalue = [105, 1]'),
 Text(0.3157894736842105, 0.3055555555555556, 'x[9] <= 2699.0\ngini =
0.198\nsamples = 9\nvalue = [8, 1]'),
```

```
 Text(0.30994152046783624, 0.25, 'gini = 0.0\nsamples = 8\nvalue = [8,
0]'),
 Text(0.3216374269005848, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.32748538011695905, 0.3055555555555556, 'gini = 0.0\nsamples =
97\nvalue = [97, 0]'),
 Text(0.32748538011695905, 0.4722222222222222, 'x[10] <= 12.5\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.3216374269005848, 0.4166666666666667, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.3333333333333333, 0.4166666666666667, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.36257309941520466, 0.5277777777777778, 'x[9] <= 10148.0\ngini
= 0.327\nsamples = 34\nvalue = [27, 7]'),
 Text(0.3508771929824561, 0.4722222222222222, 'x[0] <= 30.0\ngini =
0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.34502923976608185, 0.4166666666666667, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.3567251461988304, 0.4166666666666667, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.3742690058479532, 0.4722222222222222, 'x[13] <= 17.0\ngini =
0.231\nsamples = 30\nvalue = [26, 4]'),
 Text(0.3684210526315789, 0.4166666666666667, 'x[14] <= 5.5\ngini =
0.185\nsamples = 29\nvalue = [26, 3]'),
 Text(0.36257309941520466, 0.3611111111111111, 'gini = 0.0\nsamples =
18\nvalue = [18, 0]'),
 Text(0.3742690058479532, 0.3611111111111111, 'x[6] <= 4.0\ngini =
0.397\nsamples = 11\nvalue = [8, 3]'),
 Text(0.3684210526315789, 0.3055555555555556, 'x[14] <= 7.5\ngini =
0.5\nsamples = 6\nvalue = [3, 3]'),
 Text(0.36257309941520466, 0.25, 'gini = 0.0\nsamples = 2\nvalue = [0,
2]'),
 Text(0.3742690058479532, 0.25, 'x[2] <= 1.5\ngini = 0.375\nsamples =
4\nvalue = [3, 1]'),
 Text(0.3684210526315789, 0.19444444444444445, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.38011695906432746, 0.19444444444444445, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
 Text(0.38011695906432746, 0.3055555555555556, 'gini = 0.0\nsamples =
5\nvalue = [5, 0]'),
 Text(0.38011695906432746, 0.4166666666666667, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.4473684210526316, 0.6388888888888888, 'x[0] <= 57.5\ngini =
0.067\nsamples = 290\nvalue = [280, 10]'),
 Text(0.43567251461988304, 0.5833333333333334, 'x[3] <= 0.5\ngini =
0.055\nsamples = 282\nvalue = [274, 8]'),
 Text(0.4298245614035088, 0.5277777777777778, 'gini = 0.0\nsamples =
105\nvalue = [105, 0]'),
 Text(0.4415204678362573, 0.5277777777777778, 'x[9] <= 6225.5\ngini =
```

```
0.086\nsamples = 177\nvalue = [169, 8]'),
 Text(0.43567251461988304, 0.4722222222222222, 'x[9] <= 6128.0\ngini =
0.121\nsamples = 124\nvalue = [116, 8]'),
 Text(0.4298245614035088, 0.4166666666666667, 'x[7] <= 2.5\ngini =
0.107\nsamples = 123\nvalue = [116, 7]'),
 Text(0.40350877192982454, 0.3611111111111111, 'x[12] <= 1.5\ngini =
0.206\nsamples = 43\nvalue = [38, 5]'),
 Text(0.39766081871345027, 0.3055555555555556, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.4093567251461988, 0.3055555555555556, 'x[0] <= 51.5\ngini =
0.172\nsamples = 42\nvalue = [38, 4]'),
 Text(0.39766081871345027, 0.25, 'x[9] <= 4848.0\ngini = 0.102\
nsamples = 37\nvalue = [35, 2]'),
 Text(0.391812865497076, 0.19444444444444445, 'gini = 0.0\nsamples =
21\nvalue = [21, 0]'),
 Text(0.40350877192982454, 0.19444444444444445, 'x[9] <= 4865.5\ngini
= 0.219\nsamples = 16\nvalue = [14, 2]'),
 Text(0.39766081871345027, 0.1388888888888889, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.4093567251461988, 0.1388888888888889, 'x[6] <= 2.5\ngini =
0.124\nsamples = 15\nvalue = [14, 1]'),
 Text(0.40350877192982454, 0.08333333333333333, 'x[7] <= 1.5\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.39766081871345027, 0.027777777777777776, 'gini = 0.0\nsamples
= 1\nvalue = [0, 1]'),
 Text(0.4093567251461988, 0.027777777777777776, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.4152046783625731, 0.08333333333333333, 'gini = 0.0\nsamples =
13\nvalue = [13, 0]'),
 Text(0.42105263157894735, 0.25, 'x[10] <= 10.5\ngini = 0.48\nsamples
= 5\nvalue = [3, 2]'),
 Text(0.4152046783625731, 0.19444444444444445, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
 Text(0.4269005847953216, 0.19444444444444445, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.45614035087719296, 0.3611111111111111, 'x[0] <= 32.5\ngini =
0.049\nsamples = 80\nvalue = [78, 2]'),
 Text(0.4502923976608187, 0.3055555555555556, 'x[0] <= 31.5\ngini =
0.147\nsamples = 25\nvalue = [23, 2]'),
 Text(0.4444444444444444, 0.25, 'x[12] <= 3.5\ngini = 0.08\nsamples =
24\nvalue = [23, 1]'),
 Text(0.43859649122807015, 0.19444444444444445, 'gini = 0.0\nsamples =
21\nvalue = [21, 0]'),
 Text(0.4502923976608187, 0.19444444444444445, 'x[9] <= 3751.5\ngini =
0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.4444444444444444, 0.1388888888888889, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.45614035087719296, 0.1388888888888889, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
```

```
 Text(0.45614035087719296, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.4619883040935672, 0.3055555555555556, 'gini = 0.0\nsamples =
55\nvalue = [55, 0]'),
 Text(0.4415204678362573, 0.4166666666666667, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.4473684210526316, 0.4722222222222222, 'gini = 0.0\nsamples =
53\nvalue = [53, 0]'),
 Text(0.4590643274853801, 0.5833333333333334, 'x[14] <= 5.5\ngini =
0.375\nsamples = 8\nvalue = [6, 2]'),
 Text(0.45321637426900585, 0.5277777777777778, 'gini = 0.0\nsamples =
6\nvalue = [6, 0]'),
 Text(0.4649122807017544, 0.5277777777777778, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.5138888888888888, 0.75, 'x[12] <= 1.5\ngini = 0.249\nsamples =
192\nvalue = [164, 28]'),
 Text(0.49122807017543857, 0.6944444444444444, 'x[14] <= 5.5\ngini =
0.5\nsamples = 8\nvalue = [4, 4]'),
 Text(0.4853801169590643, 0.6388888888888888, 'x[0] <= 31.5\ngini =
0.32\nsamples = 5\nvalue = [4, 1]'),
 Text(0.47953216374269003, 0.5833333333333334, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.49122807017543857, 0.5833333333333334, 'gini = 0.0\nsamples =
4\nvalue = [4, 0]'),
 Text(0.49707602339181284, 0.6388888888888888, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.5365497076023392, 0.6944444444444444, 'x[13] <= 4.5\ngini =
0.227\nsamples = 184\nvalue = [160, 24]'),
 Text(0.5087719298245614, 0.6388888888888888, 'x[9] <= 8400.0\ngini =
0.361\nsamples = 55\nvalue = [42, 13]'),
 Text(0.5029239766081871, 0.5833333333333334, 'x[2] <= 2.5\ngini =
0.422\nsamples = 43\nvalue = [30, 13]'),
 Text(0.47953216374269003, 0.5277777777777778, 'x[0] <= 37.0\ngini =
0.499\nsamples = 19\nvalue = [10, 9]'),
 Text(0.4678362573099415, 0.4722222222222222, 'x[0] <= 24.5\ngini =
0.375\nsamples = 8\nvalue = [2, 6]'),
 Text(0.4619883040935672, 0.4166666666666667, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.47368421052631576, 0.4166666666666667, 'x[13] <= 0.5\ngini =
0.245\nsamples = 7\nvalue = [1, 6]'),
 Text(0.4678362573099415, 0.3611111111111111, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.47953216374269003, 0.3611111111111111, 'gini = 0.0\nsamples =
6\nvalue = [0, 6]'),
 Text(0.49122807017543857, 0.4722222222222222, 'x[0] <= 44.5\ngini =
0.397\nsamples = 11\nvalue = [8, 3]'),
 Text(0.4853801169590643, 0.4166666666666667, 'gini = 0.0\nsamples =
7\nvalue = [7, 0]'),
 Text(0.49707602339181284, 0.4166666666666667, 'x[4] <= 3.5\ngini =
```

```
0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.49122807017543857, 0.3611111111111111, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.5029239766081871, 0.3611111111111111, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.5263157894736842, 0.5277777777777778, 'x[11] <= 0.5\ngini =
0.278\nsamples = 24\nvalue = [20, 4]'),
 Text(0.52046783625731, 0.4722222222222222, 'gini = 0.0\nsamples = 1\
nvalue = [0, 1]'),
 Text(0.5321637426900585, 0.4722222222222222, 'x[13] <= 3.5\ngini =
0.227\nsamples = 23\nvalue = [20, 3]'),
 Text(0.52046783625731, 0.4166666666666667, 'x[10] <= 11.5\ngini =
0.111\nsamples = 17\nvalue = [16, 1]'),
 Text(0.5146198830409356, 0.3611111111111111, 'gini = 0.0\nsamples =
15\nvalue = [15, 0]'),
 Text(0.5263157894736842, 0.3611111111111111, 'x[9] <= 5362.0\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.52046783625731, 0.3055555555555556, 'gini = 0.0\nsamples = 1\
nvalue = [0, 1]'),
 Text(0.5321637426900585, 0.3055555555555556, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.543859649122807, 0.4166666666666667, 'x[9] <= 3486.0\ngini =
0.444\nsamples = 6\nvalue = [4, 2]'),
 Text(0.5380116959064327, 0.3611111111111111, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.5497076023391813, 0.3611111111111111, 'x[9] <= 4540.0\ngini =
0.32\nsamples = 5\nvalue = [4, 1]'),
 Text(0.543859649122807, 0.3055555555555556, 'gini = 0.0\nsamples = 4\
nvalue = [4, 0]'),
 Text(0.5555555555555556, 0.3055555555555556, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.5146198830409356, 0.5833333333333334, 'gini = 0.0\nsamples =
12\nvalue = [12, 0]'),
 Text(0.564327485380117, 0.6388888888888888, 'x[9] <= 5611.0\ngini =
0.156\nsamples = 129\nvalue = [118, 11]'),
 Text(0.5584795321637427, 0.5833333333333334, 'gini = 0.0\nsamples =
47\nvalue = [47, 0]'),
 Text(0.5701754385964912, 0.5833333333333334, 'x[4] <= 1.5\ngini =
0.232\nsamples = 82\nvalue = [71, 11]'),
 Text(0.5584795321637427, 0.5277777777777778, 'x[14] <= 5.0\ngini =
0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.5526315789473685, 0.4722222222222222, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.564327485380117, 0.4722222222222222, 'gini = 0.0\nsamples = 2\
nvalue = [0, 2]'),
 Text(0.5818713450292398, 0.5277777777777778, 'x[9] <= 5645.0\ngini =
0.202\nsamples = 79\nvalue = [70, 9]'),
 Text(0.5760233918128655, 0.4722222222222222, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
```

```
 Text(0.5877192982456141, 0.4722222222222222, 'x[14] <= 0.5\ngini =
0.184\nsamples = 78\nvalue = [70, 8]'),
 Text(0.5672514619883041, 0.4166666666666667, 'x[11] <= 2.5\ngini =
0.444\nsamples = 6\nvalue = [4, 2]'),
 Text(0.5614035087719298, 0.3611111111111111, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
 Text(0.5730994152046783, 0.3611111111111111, 'x[13] <= 5.5\ngini =
0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.5672514619883041, 0.3055555555555556, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.5789473684210527, 0.3055555555555556, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.6081871345029239, 0.4166666666666667, 'x[10] <= 26.5\ngini =
0.153\nsamples = 72\nvalue = [66, 6]'),
 Text(0.5964912280701754, 0.3611111111111111, 'x[9] <= 7304.5\ngini =
0.133\nsamples = 70\nvalue = [65, 5]'),
 Text(0.5906432748538012, 0.3055555555555556, 'gini = 0.0\nsamples =
26\nvalue = [26, 0]'),
 Text(0.6023391812865497, 0.3055555555555556, 'x[9] <= 7738.5\ngini =
0.201\nsamples = 44\nvalue = [39, 5]'),
 Text(0.5847953216374269, 0.25, 'x[4] <= 2.5\ngini = 0.5\nsamples = 6\
nvalue = [3, 3]'),
 Text(0.5789473684210527, 0.19444444444444445, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.5906432748538012, 0.19444444444444445, 'x[13] <= 7.5\ngini =
0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.5847953216374269, 0.1388888888888889, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.5964912280701754, 0.1388888888888889, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.6198830409356725, 0.25, 'x[11] <= 3.5\ngini = 0.1\nsamples =
38\nvalue = [36, 2]'),
 Text(0.6140350877192983, 0.19444444444444445, 'x[13] <= 15.5\ngini =
0.053\nsamples = 37\nvalue = [36, 1]'),
 Text(0.6081871345029239, 0.1388888888888889, 'gini = 0.0\nsamples =
33\nvalue = [33, 0]'),
 Text(0.6198830409356725, 0.1388888888888889, 'x[1] <= 2.0\ngini =
0.375\nsamples = 4\nvalue = [3, 1]'),
 Text(0.6140350877192983, 0.08333333333333333, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.6257309941520468, 0.08333333333333333, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
 Text(0.6257309941520468, 0.19444444444444445, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.6198830409356725, 0.3611111111111111, 'x[0] <= 46.5\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.6140350877192983, 0.3055555555555556, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.6257309941520468, 0.3055555555555556, 'gini = 0.0\nsamples =
```

```
1\nvalue = [1, 0]'),
 Text(0.8181652046783626, 0.8611111111111112, 'x[7] <= 3.5\ngini =
0.339\nsamples = 351\nvalue = [275, 76]'),
 Text(0.7101608187134503, 0.8055555555555556, 'x[2] <= 1.5\ngini =
0.401\nsamples = 223\nvalue = [161, 62]'),
 Text(0.6403508771929824, 0.75, 'x[0] <= 34.5\ngini = 0.496\nsamples =
55\nvalue = [30, 25]'),
 Text(0.6169590643274854, 0.6944444444444444, 'x[9] <= 9115.5\ngini =
0.444\nsamples = 24\nvalue = [8, 16]'),
 Text(0.6111111111111112, 0.6388888888888888, 'x[7] <= 2.5\ngini =
0.363\nsamples = 21\nvalue = [5, 16]'),
 Text(0.6052631578947368, 0.5833333333333334, 'gini = 0.0\nsamples =
11\nvalue = [0, 11]'),
 Text(0.6169590643274854, 0.5833333333333334, 'x[0] <= 27.5\ngini =
0.5\nsamples = 10\nvalue = [5, 5]'),
 Text(0.6052631578947368, 0.5277777777777778, 'x[12] <= 1.5\ngini =
0.32\nsamples = 5\nvalue = [4, 1]'),
 Text(0.5994152046783626, 0.4722222222222222, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.6111111111111112, 0.4722222222222222, 'gini = 0.0\nsamples =
4\nvalue = [4, 0]'),
 Text(0.6286549707602339, 0.5277777777777778, 'x[13] <= 14.5\ngini =
0.32\nsamples = 5\nvalue = [1, 4]'),
 Text(0.6228070175438597, 0.4722222222222222, 'gini = 0.0\nsamples =
4\nvalue = [0, 4]'),
 Text(0.6345029239766082, 0.4722222222222222, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.6228070175438597, 0.6388888888888888, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
 Text(0.6637426900584795, 0.6944444444444444, 'x[4] <= 1.5\ngini =
0.412\nsamples = 31\nvalue = [22, 9]'),
 Text(0.6578947368421053, 0.6388888888888888, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.6695906432748538, 0.6388888888888888, 'x[9] <= 9316.5\ngini =
0.366\nsamples = 29\nvalue = [22, 7]'),
 Text(0.6578947368421053, 0.5833333333333334, 'x[13] <= 4.5\ngini =
0.269\nsamples = 25\nvalue = [21, 4]'),
 Text(0.652046783625731, 0.5277777777777778, 'x[14] <= 2.5\ngini =
0.48\nsamples = 10\nvalue = [6, 4]'),
 Text(0.6461988304093568, 0.4722222222222222, 'x[10] <= 10.5\ngini =
0.375\nsamples = 8\nvalue = [6, 2]'),
 Text(0.6403508771929824, 0.4166666666666667, 'gini = 0.0\nsamples =
5\nvalue = [5, 0]'),
 Text(0.652046783625731, 0.4166666666666667, 'x[9] <= 3252.0\ngini =
0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.6461988304093568, 0.3611111111111111, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.6578947368421053, 0.3611111111111111, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.6578947368421053, 0.4722222222222222, 'gini = 0.0\nsamples =
```

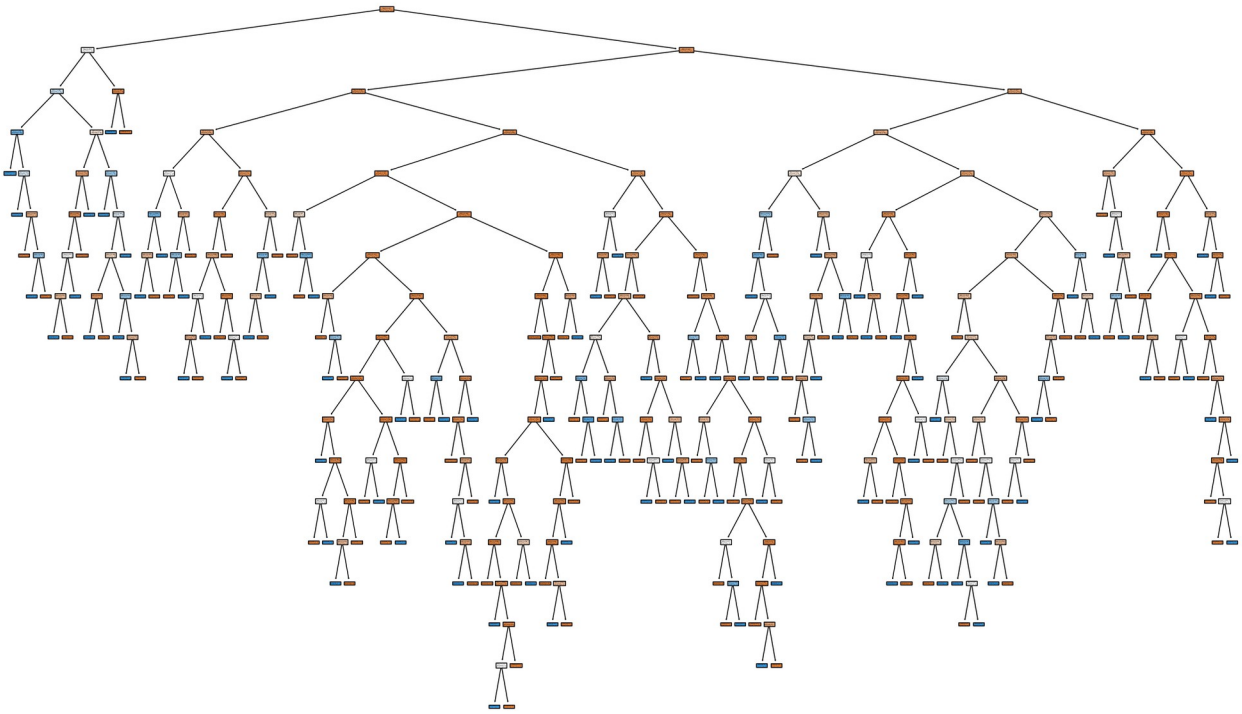
```
2\nvalue = [0, 2]'),
 Text(0.6637426900584795, 0.5277777777777778, 'gini = 0.0\nsamples =
15\nvalue = [15, 0]'),
 Text(0.6812865497076024, 0.5833333333333334, 'x[12] <= 3.5\ngini =
0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.6754385964912281, 0.5277777777777778, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.6871345029239766, 0.5277777777777778, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.7799707602339181, 0.75, 'x[6] <= 5.5\ngini = 0.343\nsamples =
168\nvalue = [131, 37]'),
 Text(0.716374269005848, 0.6944444444444444, 'x[10] <= 4.5\ngini =
0.225\nsamples = 85\nvalue = [74, 11]'),
 Text(0.6988304093567251, 0.6388888888888888, 'x[13] <= 2.5\ngini =
0.5\nsamples = 8\nvalue = [4, 4]'),
 Text(0.6929824561403509, 0.5833333333333334, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.7046783625730995, 0.5833333333333334, 'x[9] <= 2431.0\ngini =
0.32\nsamples = 5\nvalue = [4, 1]'),
 Text(0.6988304093567251, 0.5277777777777778, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.7105263157894737, 0.5277777777777778, 'gini = 0.0\nsamples =
4\nvalue = [4, 0]'),
 Text(0.7339181286549707, 0.6388888888888888, 'x[0] <= 57.0\ngini =
0.165\nsamples = 77\nvalue = [70, 7]'),
 Text(0.7280701754385965, 0.5833333333333334, 'x[9] <= 2040.0\ngini =
0.145\nsamples = 76\nvalue = [70, 6]'),
 Text(0.7222222222222222, 0.5277777777777778, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.7339181286549707, 0.5277777777777778, 'x[14] <= 13.5\ngini =
0.124\nsamples = 75\nvalue = [70, 5]'),
 Text(0.7280701754385965, 0.4722222222222222, 'x[1] <= 4.5\ngini =
0.102\nsamples = 74\nvalue = [70, 4]'),
 Text(0.7134502923976608, 0.4166666666666667, 'x[13] <= 0.5\ngini =
0.08\nsamples = 72\nvalue = [69, 3]'),
 Text(0.7017543859649122, 0.3611111111111111, 'x[9] <= 3029.0\ngini =
0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.695906432748538, 0.3055555555555556, 'gini = 0.0\nsamples = 1\
nvalue = [0, 1]'),
 Text(0.7076023391812866, 0.3055555555555556, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.7251461988304093, 0.3611111111111111, 'x[13] <= 9.5\ngini =
0.056\nsamples = 69\nvalue = [67, 2]'),
 Text(0.7192982456140351, 0.3055555555555556, 'gini = 0.0\nsamples =
52\nvalue = [52, 0]'),
 Text(0.7309941520467836, 0.3055555555555556, 'x[11] <= 3.5\ngini =
0.208\nsamples = 17\nvalue = [15, 2]'),
 Text(0.7251461988304093, 0.25, 'x[0] <= 28.5\ngini = 0.117\nsamples =
16\nvalue = [15, 1]'),
```

```
 Text(0.7192982456140351, 0.19444444444444445, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.7309941520467836, 0.19444444444444445, 'gini = 0.0\nsamples =
15\nvalue = [15, 0]'),
 Text(0.7368421052631579, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.7426900584795322, 0.4166666666666667, 'x[2] <= 3.5\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.7368421052631579, 0.3611111111111111, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.7485380116959064, 0.3611111111111111, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.7397660818713451, 0.4722222222222222, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.7397660818713451, 0.5833333333333334, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.8435672514619883, 0.6944444444444444, 'x[9] <= 8941.5\ngini =
0.43\nsamples = 83\nvalue = [57, 26]'),
 Text(0.8157894736842105, 0.6388888888888888, 'x[10] <= 9.5\ngini =
0.394\nsamples = 74\nvalue = [54, 20]'),
 Text(0.7777777777777778, 0.5833333333333334, 'x[0] <= 23.5\ngini =
0.444\nsamples = 54\nvalue = [36, 18]'),
 Text(0.7719298245614035, 0.5277777777777778, 'gini = 0.0\nsamples =
7\nvalue = [7, 0]'),
 Text(0.783625730994152, 0.5277777777777778, 'x[11] <= 2.5\ngini =
0.473\nsamples = 47\nvalue = [29, 18]'),
 Text(0.7602339181286549, 0.4722222222222222, 'x[1] <= 2.5\ngini =
0.499\nsamples = 19\nvalue = [9, 10]'),
 Text(0.7543859649122807, 0.4166666666666667, 'gini = 0.0\nsamples =
4\nvalue = [0, 4]'),
 Text(0.7660818713450293, 0.4166666666666667, 'x[13] <= 3.5\ngini =
0.48\nsamples = 15\nvalue = [9, 6]'),
 Text(0.7602339181286549, 0.3611111111111111, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
 Text(0.7719298245614035, 0.3611111111111111, 'x[9] <= 5806.5\ngini =
0.5\nsamples = 12\nvalue = [6, 6]'),
 Text(0.7660818713450293, 0.3055555555555556, 'x[9] <= 3471.0\ngini =
0.48\nsamples = 10\nvalue = [4, 6]'),
 Text(0.7543859649122807, 0.25, 'x[14] <= 4.5\ngini = 0.48\nsamples =
5\nvalue = [3, 2]'),
 Text(0.7485380116959064, 0.19444444444444445, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
 Text(0.7602339181286549, 0.19444444444444445, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.7777777777777778, 0.25, 'x[14] <= 5.0\ngini = 0.32\nsamples =
5\nvalue = [1, 4]'),
 Text(0.7719298245614035, 0.19444444444444445, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
 Text(0.783625730994152, 0.19444444444444445, 'x[2] <= 3.5\ngini =
```

```
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.7777777777777778, 0.1388888888888889, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.7894736842105263, 0.1388888888888889, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.7777777777777778, 0.3055555555555556, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.8070175438596491, 0.4722222222222222, 'x[14] <= 2.5\ngini =
0.408\nsamples = 28\nvalue = [20, 8]'),
 Text(0.7894736842105263, 0.4166666666666667, 'x[1] <= 1.5\ngini =
0.475\nsamples = 18\nvalue = [11, 7]'),
 Text(0.783625730994152, 0.3611111111111111, 'gini = 0.0\nsamples = 4\
nvalue = [4, 0]'),
 Text(0.7953216374269005, 0.3611111111111111, 'x[9] <= 2459.0\ngini =
0.5\nsamples = 14\nvalue = [7, 7]'),
 Text(0.7894736842105263, 0.3055555555555556, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
 Text(0.8011695906432749, 0.3055555555555556, 'x[5] <= 1.5\ngini =
0.463\nsamples = 11\nvalue = [4, 7]'),
 Text(0.7953216374269005, 0.25, 'gini = 0.0\nsamples = 5\nvalue = [0,
5]'),
 Text(0.8070175438596491, 0.25, 'x[0] <= 32.0\ngini = 0.444\nsamples =
6\nvalue = [4, 2]'),
 Text(0.8011695906432749, 0.19444444444444445, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.8128654970760234, 0.19444444444444445, 'gini = 0.0\nsamples =
4\nvalue = [4, 0]'),
 Text(0.8245614035087719, 0.4166666666666667, 'x[9] <= 2758.0\ngini =
0.18\nsamples = 10\nvalue = [9, 1]'),
 Text(0.8187134502923976, 0.3611111111111111, 'x[0] <= 29.0\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.8128654970760234, 0.3055555555555556, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.8245614035087719, 0.3055555555555556, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.8304093567251462, 0.3611111111111111, 'gini = 0.0\nsamples =
8\nvalue = [8, 0]'),
 Text(0.8538011695906432, 0.5833333333333334, 'x[2] <= 2.5\ngini =
0.18\nsamples = 20\nvalue = [18, 2]'),
 Text(0.847953216374269, 0.5277777777777778, 'x[1] <= 3.5\ngini =
0.444\nsamples = 6\nvalue = [4, 2]'),
 Text(0.8421052631578947, 0.4722222222222222, 'x[4] <= 3.5\ngini =
0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.8362573099415205, 0.4166666666666667, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.847953216374269, 0.4166666666666667, 'gini = 0.0\nsamples = 1\
nvalue = [1, 0]'),
 Text(0.8538011695906432, 0.4722222222222222, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
```

```
 Text(0.8596491228070176, 0.5277777777777778, 'gini = 0.0\nsamples =
14\nvalue = [14, 0]'),
 Text(0.8713450292397661, 0.6388888888888888, 'x[10] <= 9.5\ngini =
0.444\nsamples = 9\nvalue = [3, 6]'),
 Text(0.8654970760233918, 0.5833333333333334, 'gini = 0.0\nsamples =
4\nvalue = [0, 4]'),
 Text(0.8771929824561403, 0.5833333333333334, 'x[10] <= 21.5\ngini =
0.48\nsamples = 5\nvalue = [3, 2]'),
 Text(0.8713450292397661, 0.5277777777777778, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
 Text(0.8830409356725146, 0.5277777777777778, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.9261695906432749, 0.8055555555555556, 'x[10] <= 4.5\ngini =
0.195\nsamples = 128\nvalue = [114, 14]'),
 Text(0.8947368421052632, 0.75, 'x[11] <= 2.5\ngini = 0.408\nsamples =
14\nvalue = [10, 4]'),
 Text(0.8888888888888888, 0.6944444444444444, 'gini = 0.0\nsamples =
6\nvalue = [6, 0]'),
 Text(0.9005847953216374, 0.6944444444444444, 'x[13] <= 1.5\ngini =
0.5\nsamples = 8\nvalue = [4, 4]'),
 Text(0.8947368421052632, 0.6388888888888888, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.9064327485380117, 0.6388888888888888, 'x[9] <= 2499.5\ngini =
0.444\nsamples = 6\nvalue = [4, 2]'),
 Text(0.9005847953216374, 0.5833333333333334, 'x[6] <= 4.0\ngini =
0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.8947368421052632, 0.5277777777777778, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.9064327485380117, 0.5277777777777778, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.9122807017543859, 0.5833333333333334, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
 Text(0.9576023391812866, 0.75, 'x[0] <= 50.5\ngini = 0.16\nsamples =
114\nvalue = [104, 10]'),
 Text(0.9385964912280702, 0.6944444444444444, 'x[12] <= 1.5\ngini =
0.128\nsamples = 102\nvalue = [95, 7]'),
 Text(0.9327485380116959, 0.6388888888888888, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.9444444444444444, 0.6388888888888888, 'x[13] <= 7.5\ngini =
0.112\nsamples = 101\nvalue = [95, 6]'),
 Text(0.9239766081871345, 0.5833333333333334, 'x[4] <= 3.5\ngini =
0.029\nsamples = 67\nvalue = [66, 1]'),
 Text(0.9181286549707602, 0.5277777777777778, 'gini = 0.0\nsamples =
63\nvalue = [63, 0]'),
 Text(0.9298245614035088, 0.5277777777777778, 'x[10] <= 9.5\ngini =
0.375\nsamples = 4\nvalue = [3, 1]'),
 Text(0.9239766081871345, 0.4722222222222222, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.935672514619883, 0.4722222222222222, 'gini = 0.0\nsamples = 3\
```

```
nvalue = [3, 0]'),
 Text(0.9649122807017544, 0.5833333333333334, 'x[0] <= 29.5\ngini =
0.251\nsamples = 34\nvalue = [29, 5]'),
 Text(0.9532163742690059, 0.5277777777777778, 'x[12] <= 2.5\ngini =
0.5\nsamples = 4\nvalue = [2, 2]'),
 Text(0.9473684210526315, 0.4722222222222222, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.9590643274853801, 0.4722222222222222, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.9766081871345029, 0.5277777777777778, 'x[3] <= 0.5\ngini =
0.18\nsamples = 30\nvalue = [27, 3]'),
 Text(0.9707602339181286, 0.4722222222222222, 'gini = 0.0\nsamples =
16\nvalue = [16, 0]'),
 Text(0.9824561403508771, 0.4722222222222222, 'x[9] <= 3112.5\ngini =
0.337\nsamples = 14\nvalue = [11, 3]'),
 Text(0.9766081871345029, 0.4166666666666667, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.9883040935672515, 0.4166666666666667, 'x[12] <= 3.5\ngini =
0.26\nsamples = 13\nvalue = [11, 2]'),
 Text(0.9824561403508771, 0.3611111111111111, 'x[0] <= 44.5\ngini =
0.153\nsamples = 12\nvalue = [11, 1]'),
 Text(0.9766081871345029, 0.3055555555555556, 'gini = 0.0\nsamples =
10\nvalue = [10, 0]'),
 Text(0.9883040935672515, 0.3055555555555556, 'x[9] <= 7695.5\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.9824561403508771, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.9941520467836257, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
 Text(0.9941520467836257, 0.3611111111111111, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.9766081871345029, 0.6944444444444444, 'x[2] <= 2.0\ngini =
0.375\nsamples = 12\nvalue = [9, 3]'),
 Text(0.9707602339181286, 0.6388888888888888, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.9824561403508771, 0.6388888888888888, 'x[1] <= 2.0\ngini =
0.18\nsamples = 10\nvalue = [9, 1]'),
 Text(0.9766081871345029, 0.5833333333333334, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.9883040935672515, 0.5833333333333334, 'gini = 0.0\nsamples =
9\nvalue = [9, 0]')]
```



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

}

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

grid_search.fit(x_train,y_train)
```

```
C:\Users\DELL\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\DELL\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\DELL\anaconda3\Lib\site-packages\sklearn\base.py",
line 1144, in wrapper
    estimator._validate_params()
  File "C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\base.py",
line 637, in _validate_params
    validate_parameter_constraints(
  File "C:\Users\DELL\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
{'log2', 'sqrt'} or None. Got 'auto' instead.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\model_selection\
_search.py:976: UserWarning: One or more of the test scores are non-
finite: [       nan        nan 0.83503065 0.83928597 0.83588172
0.83928597
        nan        nan 0.8409881  0.84013704 0.83673278 0.83758384
        nan        nan 0.83504868 0.8409881  0.8341832  0.83928597
        nan        nan 0.83759106 0.84013704 0.83588532 0.83928958
        nan        nan 0.81461955 0.84182113 0.83759466 0.8367436
        nan        nan 0.83928597 0.83928597 0.83758384 0.83928597
        nan        nan 0.83843851 0.83928597 0.84269023 0.83928597
        nan        nan 0.83503065 0.83332853 0.8341868  0.83673999
        nan        nan 0.83673639 0.83417959 0.83845655 0.83335016
        nan        nan 0.82909845 0.82654526 0.83672196 0.8341868 ]
  warnings.warn(
```

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

```
grid_search.best_params_
```

```
{'criterion': 'entropy',
 'max_depth': 2,
 'max_features': 'log2',
 'splitter': 'best'}
```

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

```
 splitter='best')
dtc_cv.fit(x_train,y_train)
```

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

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

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

```
              precision    recall  f1-score   support

           0       0.84      1.00      0.91       246
           1       0.50      0.02      0.04        48

    accuracy                           0.84       294
   macro avg       0.67      0.51      0.48       294
weighted avg       0.78      0.84      0.77       294
```

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

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

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

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

```
C:\Users\DELL\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\DELL\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\DELL\anaconda3\Lib\site-packages\sklearn\base.py",
line 1144, in wrapper
    estimator._validate_params()
  File "C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\base.py",
line 637, in _validate_params
```

```
    validate_parameter_constraints(
  File "C:\Users\DELL\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
{'log2', 'sqrt'} or None. Got 0 instead.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\model_selection\
_search.py:976: UserWarning: One or more of the test scores are non-
finite: [       nan 0.84355353 0.84779806 0.84608866 0.8452412
0.84269883
 0.84437926 0.84440099 0.84015645 0.84100391 0.84269883 0.8418224
 0.84100391 0.84524844        nan 0.8460959  0.84950022 0.84525569
 0.84014197 0.83843981 0.84353904 0.84355353 0.84269883 0.83759235
 0.83843981 0.83762857 0.84014197 0.84440099        nan 0.84440099
 0.84950746 0.84269158 0.84608866 0.84014197 0.84269883 0.8384543
 0.84017094 0.83504274 0.84522671 0.8410184  0.84183688 0.84098942
        nan 0.85119513 0.84356801 0.8410184  0.84440099 0.84185861
 0.84099667 0.84526293 0.83589744 0.83760684 0.83589744 0.84017094
 0.84015645 0.84522671        nan 0.84779806 0.85120238 0.84527017
 0.84101115 0.84610314 0.84100391 0.84266985 0.84524844 0.84270607
 0.83759959 0.84013472 0.83932348 0.84185861]
  warnings.warn(
```

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

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

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

```
              precision    recall  f1-score   support

           0       0.85      0.98      0.91       246
           1       0.50      0.08      0.14        48

    accuracy                           0.84       294
   macro avg       0.67      0.53      0.53       294
weighted avg       0.79      0.84      0.78       294
```

```
rfc_cv.best_params_
```

```
{'max_depth': 14, 'max_features': 2}
```

```
# Evaluation the model
#Accuracy score
from sklearn.metrics import
accuracy_score,confusion_matrix,classification_report,roc_auc_score,ro
c_curve
accuracy=accuracy_score(y_test,pred)
print('Accuracy:',accuracy)
precision=precision_score(y_test,pred)
print('Precision:',precision)
recall=recall_score(y_test,pred)
print('Recall:',recall)
f1=f1_score(y_test,pred)
print('F1 score:', f1)
```

```
Accuracy: 0.8367346938775511
Precision: 0.5
Recall: 0.08333333333333333
F1 score: 0.14285714285714285
```

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

```
array([[242,   4],
       [ 44,   4]], dtype=int64)
```

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

col_0	0	1
row_0		
0	242	4
1	44	4

```
# Accuracy=(TP+TN)/(TP+TN+FP+FN)
(242+4)/(242+4+4+44)
```

```
0.8367346938775511
```

```
# Precision=TP/(TP+FP)
(242)/(242+4)
```

```
0.983739837398374
```

```
# Recall=TP/(TP+FN)
(242)/(242+44)
```

```
0.8461538461538461
```

```
# F1 score=2*Precision*Recall/(Precision+Recall)
(2*(0.983739837398374)*(0.8461538461538461))/(0.983739837398374+0.8461
538461538461)
```

```
0.9097744360902256
```