

# ASSIGNMENT-4

September 28, 2023

KADE NAVANEESWAR GOWD VIT-AP

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[2]: df=pd.read_csv("Employee-Attrition.csv")
```

```
[3]: df.head()
```

```
[3]:   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]

[4]: 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]

```

## 1 Data Preprocessing

[5]: df.info()

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Age              1470 non-null    int64  
 1   Attrition        1470 non-null    object  
 2   BusinessTravel   1470 non-null    object  
 3   DailyRate         1470 non-null    int64  
 4   Department        1470 non-null    object  
 5   DistanceFromHome 1470 non-null    int64  
 6   Education         1470 non-null    int64  
 7   EducationField    1470 non-null    object  
 8   EmployeeCount     1470 non-null    int64  
 9   EmployeeNumber    1470 non-null    int64  
 10  EnvironmentSatisfaction 1470 non-null    int64

```

```

11 Gender           1470 non-null   object
12 HourlyRate      1470 non-null   int64
13 JobInvolvement  1470 non-null   int64
14 JobLevel         1470 non-null   int64
15 JobRole          1470 non-null   object
16 JobSatisfaction 1470 non-null   int64
17 MaritalStatus    1470 non-null   object
18 MonthlyIncome    1470 non-null   int64
19 MonthlyRate      1470 non-null   int64
20 NumCompaniesWorked 1470 non-null   int64
21 Over18           1470 non-null   object
22 Overtime          1470 non-null   object
23 PercentSalaryHike 1470 non-null   int64
24 PerformanceRating 1470 non-null   int64
25 RelationshipSatisfaction 1470 non-null   int64
26 StandardHours    1470 non-null   int64
27 StockOptionLevel 1470 non-null   int64
28 TotalWorkingYears 1470 non-null   int64
29 TrainingTimesLastYear 1470 non-null   int64
30 WorkLifeBalance  1470 non-null   int64
31 YearsAtCompany   1470 non-null   int64
32 YearsInCurrentRole 1470 non-null   int64
33 YearsSinceLastPromotion 1470 non-null   int64
34 YearsWithCurrManager 1470 non-null   int64
dtypes: int64(26), object(9)
memory usage: 402.1+ KB

```

[6]: df.corr()

```
C:\Users\mb419\AppData\Local\Temp\ipykernel_3768\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	
Age		StandardHours	StockOptionLevel	TotalWorkingYears \
DailyRate		NaN	0.037510	0.680381
DistanceFromHome		NaN	0.042143	0.014515
Education		NaN	0.044872	0.004628
EmployeeCount		NaN	0.018422	0.148280
EmployeeNumber		NaN	NaN	NaN
EnvironmentSatisfaction		NaN	0.062227	-0.014365
HourlyRate		NaN	0.003432	-0.002693
JobInvolvement		NaN	0.050263	-0.002334
JobLevel		NaN	0.021523	-0.005533
JobSatisfaction		NaN	0.013984	0.782208
MonthlyIncome		NaN	0.010690	-0.020185
MonthlyRate		NaN	0.005408	0.772893
NumCompaniesWorked		NaN	-0.034323	0.026442
PercentSalaryHike		NaN	0.030075	0.237639
PerformanceRating		NaN	0.007528	-0.020608
RelationshipSatisfaction		NaN	NaN	0.003506
StandardHours		NaN	-0.045952	0.006744
StockOptionLevel		NaN	NaN	0.024054
TotalWorkingYears		NaN	1.000000	NaN
TrainingTimesLastYear		NaN	0.010136	0.010136
WorkLifeBalance		NaN	0.011274	1.000000
YearsAtCompany		NaN	0.004129	-0.035662
YearsInCurrentRole		NaN	0.015058	0.001008
YearsSinceLastPromotion		NaN	0.050818	0.628133
YearsWithCurrManager		NaN	0.014352	0.460365
			NaN	0.404858
			0.024698	0.459188
Age		TrainingTimesLastYear	WorkLifeBalance \	
DailyRate		-0.019621	-0.021490	
DistanceFromHome		0.002453	-0.037848	
Education		-0.036942	-0.026556	
EmployeeCount		-0.025100	0.009819	
EmployeeNumber		NaN	NaN	
EnvironmentSatisfaction		0.023603	0.010309	
HourlyRate		-0.019359	0.027627	
JobInvolvement		-0.008548	-0.004607	
JobLevel		-0.015338	-0.014617	
JobSatisfaction		-0.018191	0.037818	
MonthlyIncome		-0.005779	-0.019459	
MonthlyRate		-0.021736	0.030683	
		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]

[7]: df.corr().DailyRate.sort\_values(ascending=False)

```
C:\Users\mb419\AppData\Local\Temp\ipykernel_3768\2954484028.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().DailyRate.sort\_values(ascending=False)

[7]:

DailyRate	1.000000
JobInvolvement	0.046135
StockOptionLevel	0.042143
NumCompaniesWorked	0.038153
JobSatisfaction	0.030571
HourlyRate	0.023381
PercentSalaryHike	0.022704
EnvironmentSatisfaction	0.018355
TotalWorkingYears	0.014515
Age	0.010661
YearsInCurrentRole	0.009932
RelationshipSatisfaction	0.007846
MonthlyIncome	0.007707
JobLevel	0.002966
TrainingTimesLastYear	0.002453

```
PerformanceRating      0.000473
DistanceFromHome     -0.004985
Education            -0.016806
YearsWithCurrManager -0.026363
MonthlyRate          -0.032182
YearsSinceLastPromotion -0.033229
YearsAtCompany        -0.034055
WorkLifeBalance      -0.037848
EmployeeNumber        -0.050990
EmployeeCount         NaN
StandardHours         NaN
Name: DailyRate, dtype: float64
```

```
[8]: df.isnull().any()
```

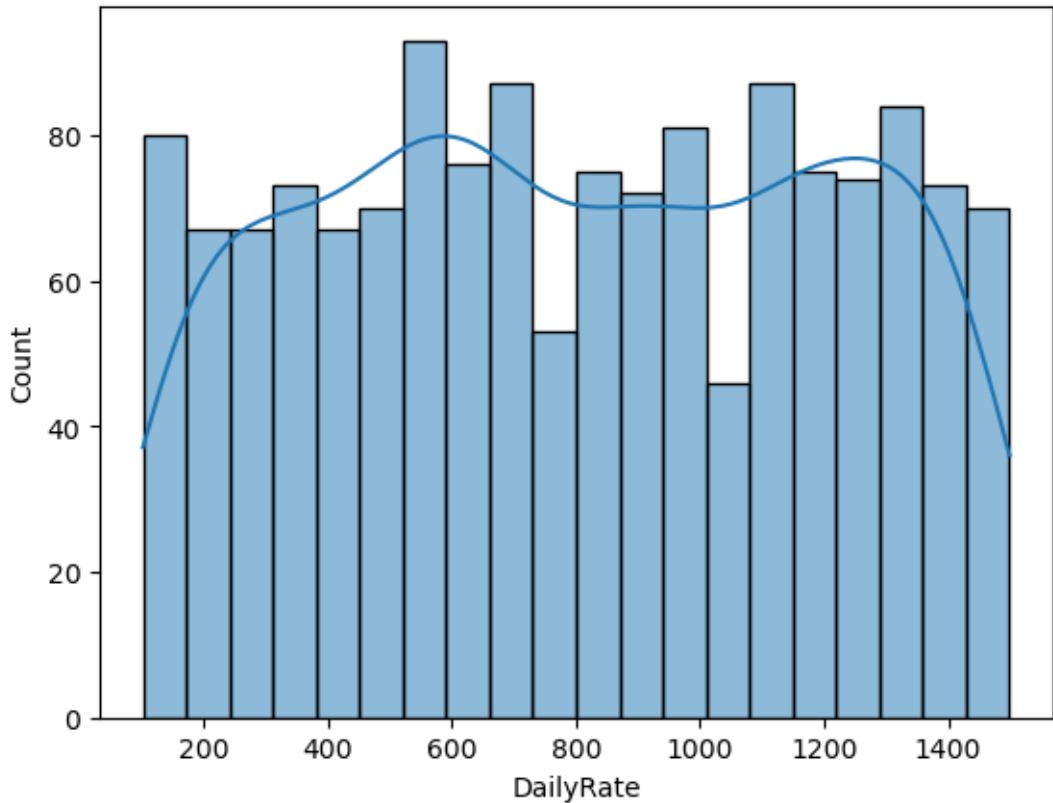
```
[8]: Age                  False
Attrition            False
BusinessTravel        False
DailyRate             False
Department           False
DistanceFromHome     False
Education             False
EducationField        False
EmployeeCount         False
EmployeeNumber        False
EnvironmentSatisfaction 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
```

```
[9]: df.isnull().sum()
```

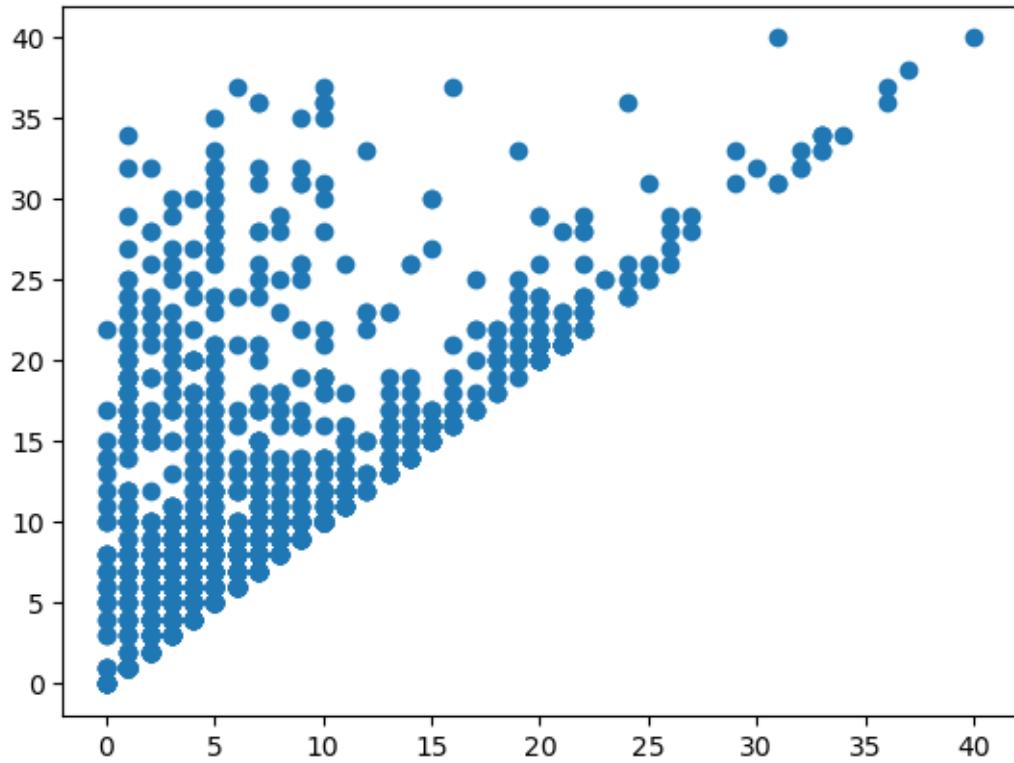
```
[9]: Age                  0
Attrition              0
BusinessTravel          0
DailyRate               0
Department              0
DistanceFromHome        0
Education               0
EducationField           0
EmployeeCount            0
EmployeeNumber           0
EnvironmentSatisfaction 0
Gender                  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
```

```
[10]: sns.histplot(df['DailyRate'], bins=20, kde=True)
plt.show()
```



```
[11]: plt.scatter(df["YearsAtCompany"],df["TotalWorkingYears"])
```

```
[11]: <matplotlib.collections.PathCollection at 0x2cd2dc6e310>
```

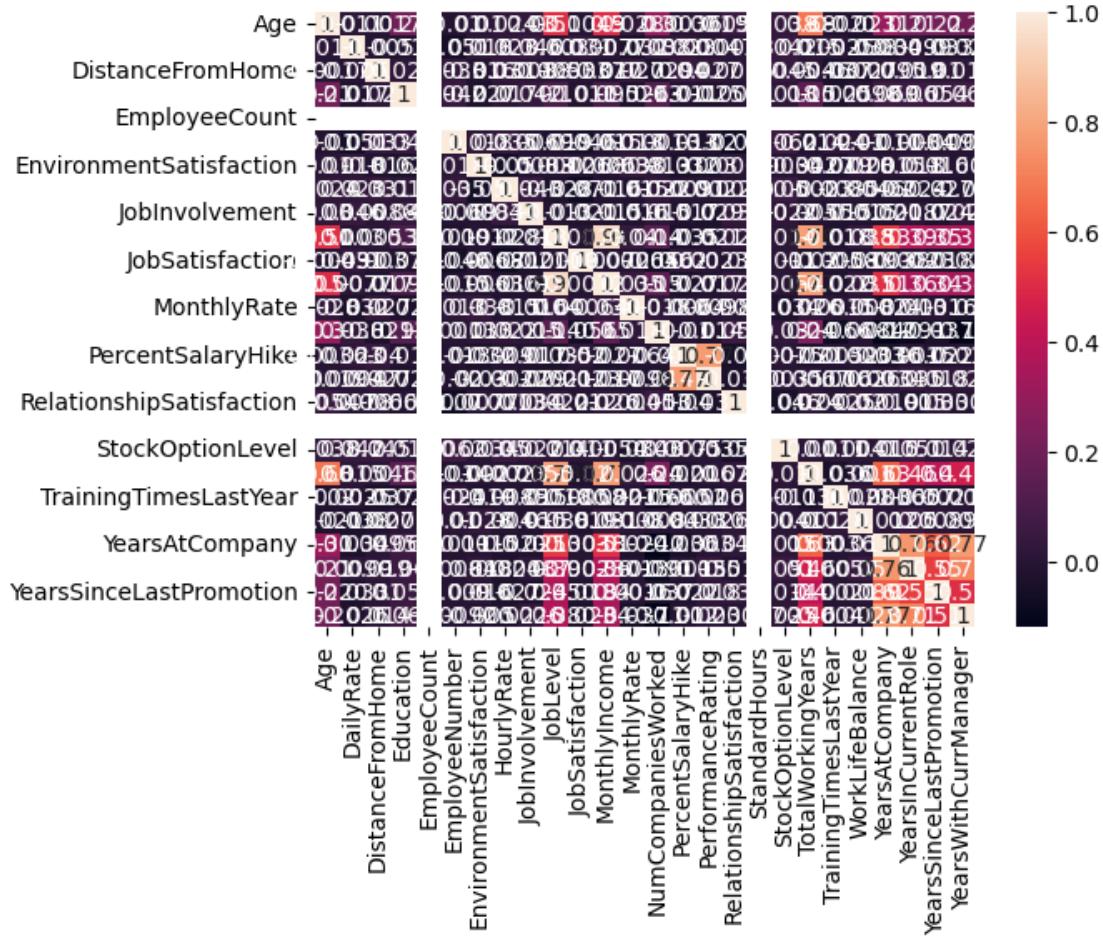


```
[12]: sns.heatmap(df.corr(), annot=True)
```

C:\Users\mb419\AppData\Local\Temp\ipykernel\_3768\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.

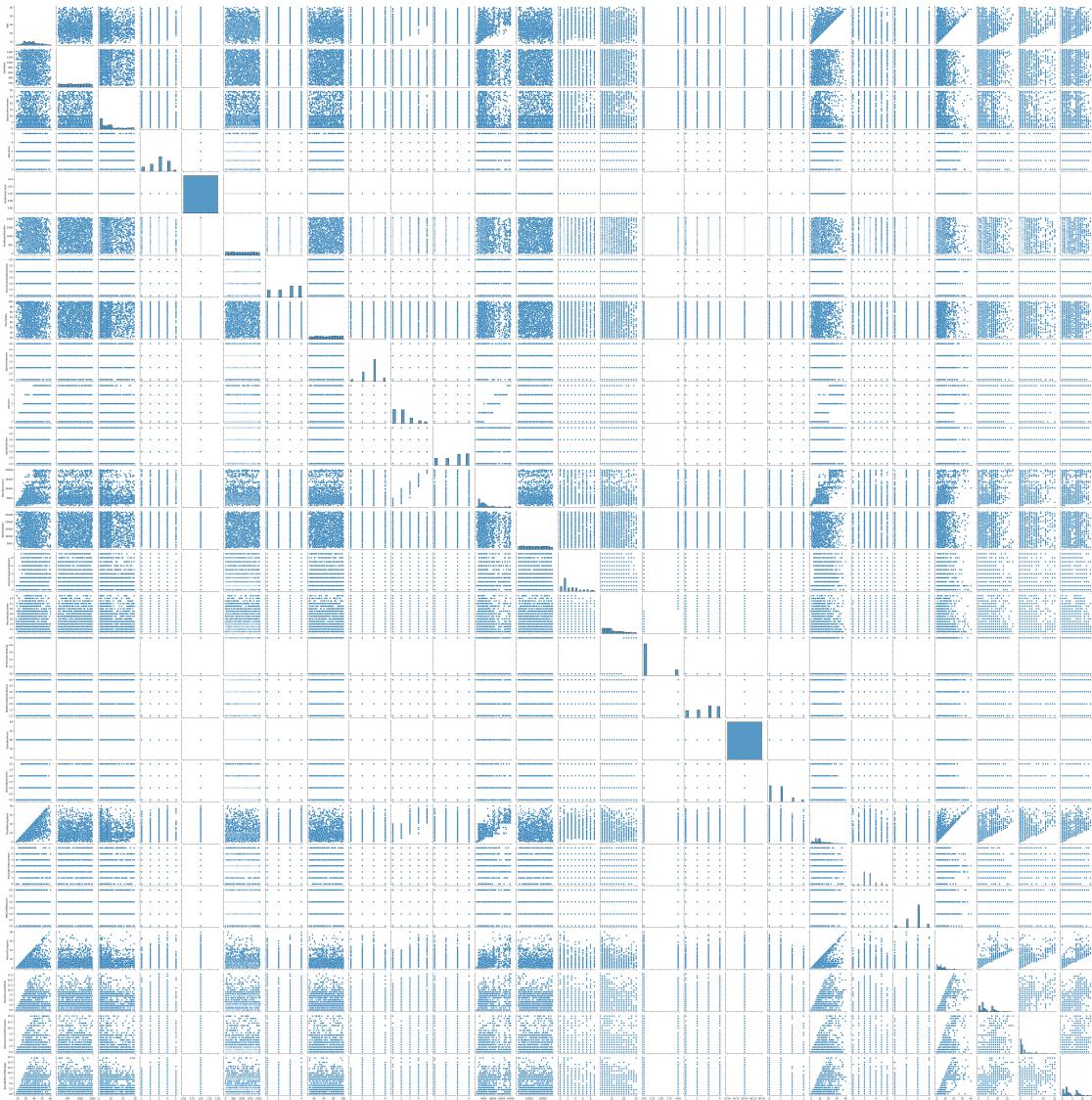
```
    sns.heatmap(df.corr(), annot=True)
```

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

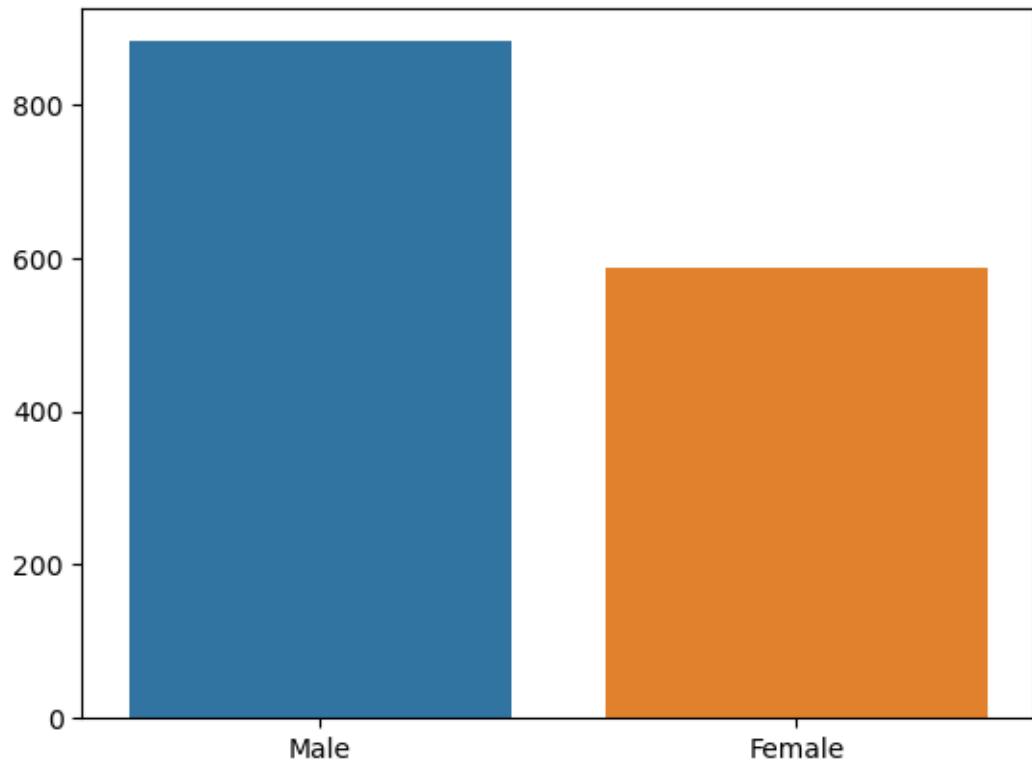


```
[13]: sns.pairplot(df)
```

```
[13]: <seaborn.axisgrid.PairGrid at 0x2cd2da62310>
```

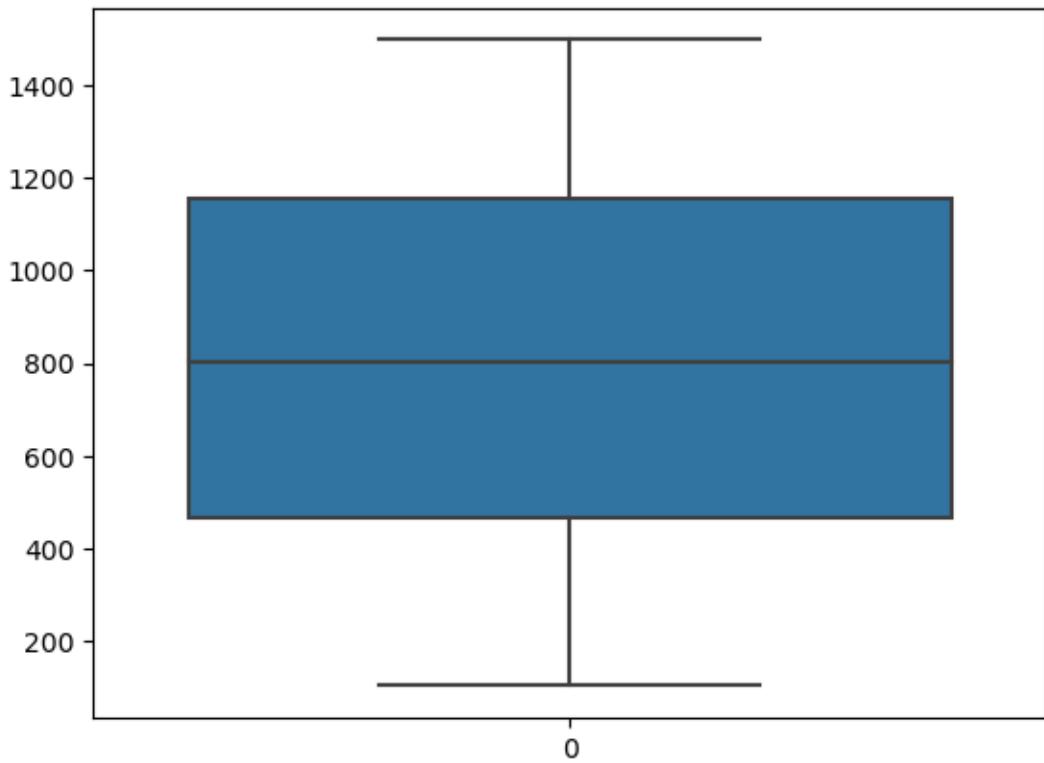


```
[14]: gender_count = df['Gender'].value_counts()
sns.barplot(x=gender_count.index, y=gender_count.values)
plt.show()
```



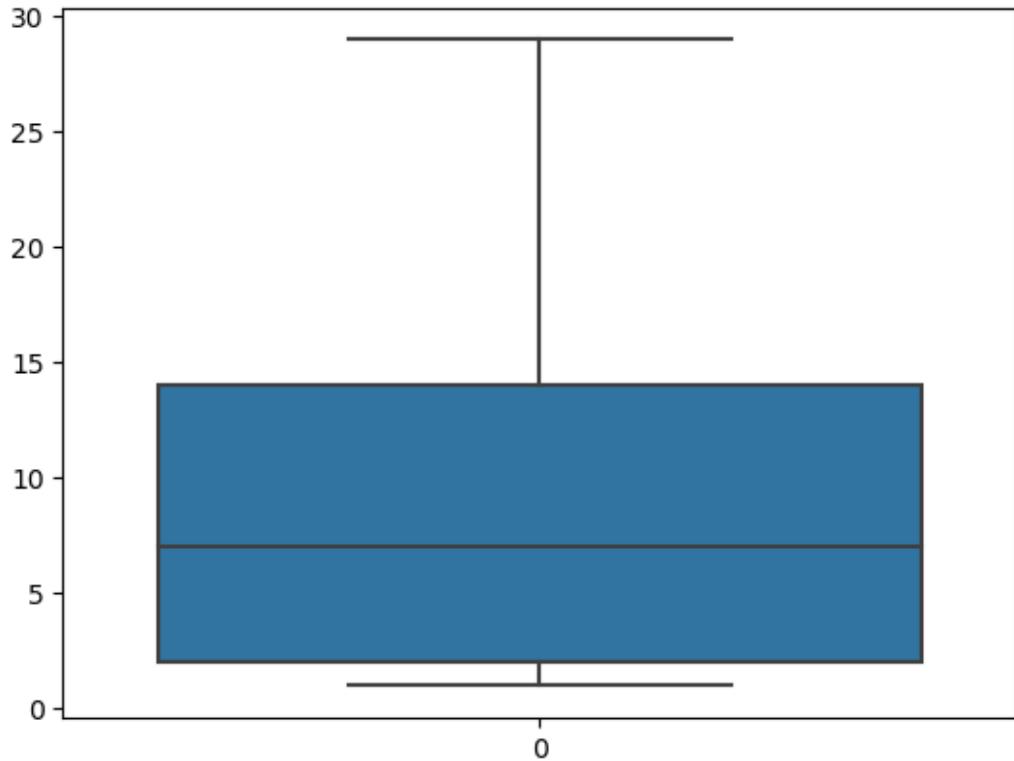
```
[15]: sns.boxplot(df.DailyRate)
```

```
[15]: <Axes: >
```



```
[16]: sns.boxplot(df.DistanceFromHome)
```

```
[16]: <Axes: >
```



[17]: df.head()

```
[17]:    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]

```
[18]: x = df.drop('Attrition', axis=1)
x.head()
```

```

[18]:   Age      BusinessTravel  DailyRate          Department \
0    41      Travel_Rarely     1102            Sales
1    49      Travel_Frequently  279  Research & Development
2    37      Travel_Rarely     1373  Research & Development
3    33      Travel_Frequently  1392  Research & Development
4    27      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

```

```

EnvironmentSatisfaction ... RelationshipSatisfaction  StandardHours \
0                   2 ...                           1                80
1                   3 ...                           4                80
2                   4 ...                           2                80
3                   4 ...                           3                80
4                   1 ...                           4                80

```

```

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

```

```

YearsAtCompany  YearsInCurrentRole  YearsSinceLastPromotion \
0              6                  4                      0
1              10                 7                      1
2              0                  0                      0
3              8                  7                      3
4              2                  2                      2

YearsWithCurrManager
0                  5
1                  7
2                  0
3                  0
4                  2

[5 rows x 34 columns]

[19]: x.head()

```

	Age	BusinessTravel	DailyRate	Department	
0	41	Travel_Rarely	1102	Sales	
1	49	Travel_Frequently	279	Research & Development	
2	37	Travel_Rarely	1373	Research & Development	
3	33	Travel_Frequently	1392	Research & Development	
4	27	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	

	EnvironmentSatisfaction	...	RelationshipSatisfaction	StandardHours	
0	2	...	1	80	
1	3	...	4	80	
2	4	...	2	80	
3	4	...	3	80	
4	1	...	4	80	

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

```
YearsAtCompany  YearsInCurrentRole  YearsSinceLastPromotion \
0              6                  4                  0
1              10                 7                  1
2              0                  0                  0
3              8                  7                  3
4              2                  2                  2

YearsWithCurrManager
0                  5
1                  7
2                  0
3                  0
4                  2

[5 rows x 34 columns]
```

```
[20]: x.shape
```

```
[20]: (1470, 34)
```

```
[21]: type(x)
```

```
[21]: pandas.core.frame.DataFrame
```

```
[22]: y = df['Attrition']
y.head()
```

```
[22]: 0    Yes
1    No
2    Yes
3    No
4    No
Name: Attrition, dtype: object
```

```
[23]: from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
[24]: x["BusinessTravel"]=le.fit_transform(x["BusinessTravel"])
x["Department"]=le.fit_transform(x["Department"])
x["EducationField"]=le.fit_transform(x["EducationField"])
x["Gender"]=le.fit_transform(x["Gender"])
x["JobRole"]=le.fit_transform(x["JobRole"])
x["MaritalStatus"]=le.fit_transform(x["MaritalStatus"])
x["Over18"]=le.fit_transform(x["Over18"])
x["OverTime"]=le.fit_transform(x["OverTime"])
```

```
[25]: x.head()
```

```
[25]:    Age BusinessTravel DailyRate Department DistanceFromHome Education \
0    41            2     1102           2                  1          2
1    49            1     279            1                  8          1
2    37            2     1373           1                  2          2
3    33            1     1392           1                  3          4
4    27            2      591            1                  2          1

   EducationField EmployeeCount EmployeeNumber EnvironmentSatisfaction \
0                 1             1              1                  2
1                 1             1              2                  3
2                 4             1              4                  4
3                 1             1              5                  4
4                 3             1              7                  1

   ... 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 34 columns]

```
[26]: categorical_cols = ['BusinessTravel', 'Department', 'EducationField', 'Gender', 'JobRole', 'MaritalStatus', 'OverTime']
numeric_cols = [col for col in df.columns if col not in categorical_cols]

# Encode categorical columns using label encoding
label_encoder = LabelEncoder()
for col in categorical_cols:
    df[col] = label_encoder.fit_transform(df[col])

# Encode the target variable "Attrition"
```

```

df['Attrition'] = label_encoder.fit_transform(df['Attrition'])

# Now, the entire dataset is encoded, including categorical columns and the target variable
X = df.drop(columns=['Attrition'])
y = df['Attrition']

```

[27]: x.head()

```

[27]:    Age BusinessTravel DailyRate Department DistanceFromHome Education \
0     41            2      1102          2                  1           2
1     49            1      279           1                  8           1
2     37            2     1373           1                  2           2
3     33            1     1392           1                  3           4
4     27            2      591           1                  2           1

   EducationField EmployeeCount EmployeeNumber EnvironmentSatisfaction \
0                 1             1              1                  2
1                 1             1              2                  3
2                 4             1              4                  4
3                 1             1              5                  4
4                 3             1              7                  1

   ... 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 34 columns]

[28]: y.head()

```
[28]: 0    1  
1    0  
2    1  
3    0  
4    0  
Name: Attrition, dtype: int32
```

```
[29]: from sklearn.preprocessing import MinMaxScaler  
ms=MinMaxScaler()
```

```
[30]: X_Scaled=ms.fit_transform(x)  
X_Scaled=pd.DataFrame(ms.fit_transform(x),columns=x.columns)
```

```
[31]: X_Scaled.head()
```

```
[31]:      Age  BusinessTravel  DailyRate  Department  DistanceFromHome  \\\n0  0.547619           1.0   0.715820           1.0        0.000000  
1  0.738095           0.5   0.126700           0.5        0.250000  
2  0.452381           1.0   0.909807           0.5        0.035714  
3  0.357143           0.5   0.923407           0.5        0.071429  
4  0.214286           1.0   0.350036           0.5        0.035714  
  
      Education  EducationField  EmployeeCount  EmployeeNumber  \\\n0       0.25          0.2          0.0        0.000000  
1       0.00          0.2          0.0        0.000484  
2       0.25          0.8          0.0        0.001451  
3       0.75          0.2          0.0        0.001935  
4       0.00          0.6          0.0        0.002903  
  
      EnvironmentSatisfaction  ...  RelationshipSatisfaction  StandardHours  \\\n0            0.333333  ...        0.000000        0.0  
1            0.666667  ...        1.000000        0.0  
2            1.000000  ...        0.333333        0.0  
3            1.000000  ...        0.666667        0.0  
4            0.000000  ...        1.000000        0.0  
  
      StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \\\n0            0.000000          0.200          0.0  
1            0.333333          0.250          0.5  
2            0.000000          0.175          0.5  
3            0.000000          0.200          0.5  
4            0.333333          0.150          0.5  
  
      WorkLifeBalance  YearsAtCompany  YearsInCurrentRole  \\\n0            0.000000          0.15        0.222222  
1            0.666667          0.25        0.388889  
2            0.666667          0.00        0.000000
```

3	0.666667	0.20	0.388889
4	0.666667	0.05	0.111111
	YearsSinceLastPromotion	YearsWithCurrManager	
0	0.000000	0.294118	
1	0.066667	0.411765	
2	0.000000	0.000000	
3	0.200000	0.000000	
4	0.133333	0.117647	

[5 rows x 34 columns]

```
[33]: print(x_train.shape,x_test.shape,y_train.shape,y_test.shape)
```

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

## 2 Logistic Regression

```
[34]: from sklearn.linear_model import LogisticRegression  
model=LogisticRegression()
```

```
[35]: model.fit(x_train,y_train)
```

[35]: LogisticRegression()

```
[36]: pred=model.predict(x_test)  
pred
```

```
[37]: y_test
```

```
[37]: 442      0  
1091     0  
981      1  
785      0  
1332     1  
..  
1439     0  
481      0  
124      1  
198      0  
1229     0  
Name: Attrition, Length: 294, dtype: int32
```

```
[38]: df
```

```
[38]:    Age Attrition BusinessTravel DailyRate Department DistanceFromHome \\  
0      41        1            2       1102          2                  1  
1      49        0            1       279           1                  8  
2      37        1            2       1373          1                  2  
3      33        0            1       1392          1                  3  
4      27        0            2       591           1                  2  
...   ...     ...     ...     ...     ...     ...  
1465    36        0            1       884           1                 23  
1466    39        0            2       613           1                  6  
1467    27        0            2       155           1                  4  
1468    49        0            1      1023          2                  2  
1469    34        0            2       628           1                  8  
  
    Education EducationField EmployeeCount EmployeeNumber ... \\  
0          2             1            1            1          1 ...  
1          1             1            1            1          2 ...  
2          2             4            1            1          4 ...  
3          4             1            1            1          5 ...  
4          1             3            1            1          7 ...  
...   ...     ...     ...     ...     ...     ...  
1465    2             3            1            1        2061 ...  
1466    1             3            1            1        2062 ...  
1467    3             1            1            1        2064 ...  
1468    3             3            1            1        2065 ...  
1469    3             3            1            1        2068 ...  
  
    RelationshipSatisfaction StandardHours StockOptionLevel \\  
0                      1            80            0  
1                      4            80            1  
2                      2            80            0
```

3	3	80	0
4	4	80	1
...	...	...	...
1465	3	80	1
1466	1	80	1
1467	2	80	1
1468	4	80	0
1469	1	80	0
\\			
0	8	0	1
1	10	3	3
2	7	3	3
3	8	3	3
4	6	3	3
...	...	...	...
1465	17	3	3
1466	9	5	3
1467	6	0	3
1468	17	3	2
1469	6	3	4
\\			
0	6	4	0
1	10	7	1
2	0	0	0
3	8	7	3
4	2	2	2
...	...	...	...
1465	5	2	0
1466	7	7	1
1467	6	2	0
1468	9	6	0
1469	4	3	1
\\			
YearsWithCurrManager			
0	5		
1	7		
2	0		
3	0		
4	2		
...	...		
1465	3		
1466	7		
1467	3		
1468	8		
1469	2		

```
[1470 rows x 35 columns]
```

```
[39]: x.head()
```

```
[39]:    Age BusinessTravel DailyRate Department DistanceFromHome Education \
0    41            2     1102           2                  1          2
1    49            1     279            1                  8          1
2    37            2     1373           1                  2          2
3    33            1     1392           1                  3          4
4    27            2      591            1                  2          1

   EducationField EmployeeCount EmployeeNumber EnvironmentSatisfaction \
0                 1             1              1                  2
1                 1             1              2                  3
2                 4             1              4                  4
3                 1             1              5                  4
4                 3             1              7                  1

   ... 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 34 columns]
```

```
[40]: model.predict(ms.transform([[41,2,1102,2,1,2,1,1,1,2,0,94,3,2,7,4,2,5993,19479,8,0,1,11,3,1,80,0,8,0,1,6,4,0,5]]))
```

```
C:\Users\mb419\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\mb419\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(
[40]: array([1])

[41]: from sklearn.metrics import
       accuracy_score,confusion_matrix,classification_report,roc_auc_score,roc_curve
[42]: accuracy_score(y_test,pred)
[42]: 0.8843537414965986
[43]: confusion_matrix(y_test,pred)
[43]: array([[242,    3],
           [ 31,   18]], dtype=int64)
[44]: pd.crosstab(y_test,pred)
[44]: col_0      0    1
      Attrition
      0        242    3
      1        31   18
[45]: a1=(242+18)/294 #accuracy
      a1
[45]: 0.8843537414965986
[46]: r=18/(18+31) #recall
      r
[46]: 0.3673469387755102
[47]: p=18/(3+18) #precision
      p
[47]: 0.8571428571428571
[48]: f1=2*p*r/(p+r) #F1 score
      f1
[48]: 0.5142857142857143
[49]: print(classification_report(y_test,pred))
```

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

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

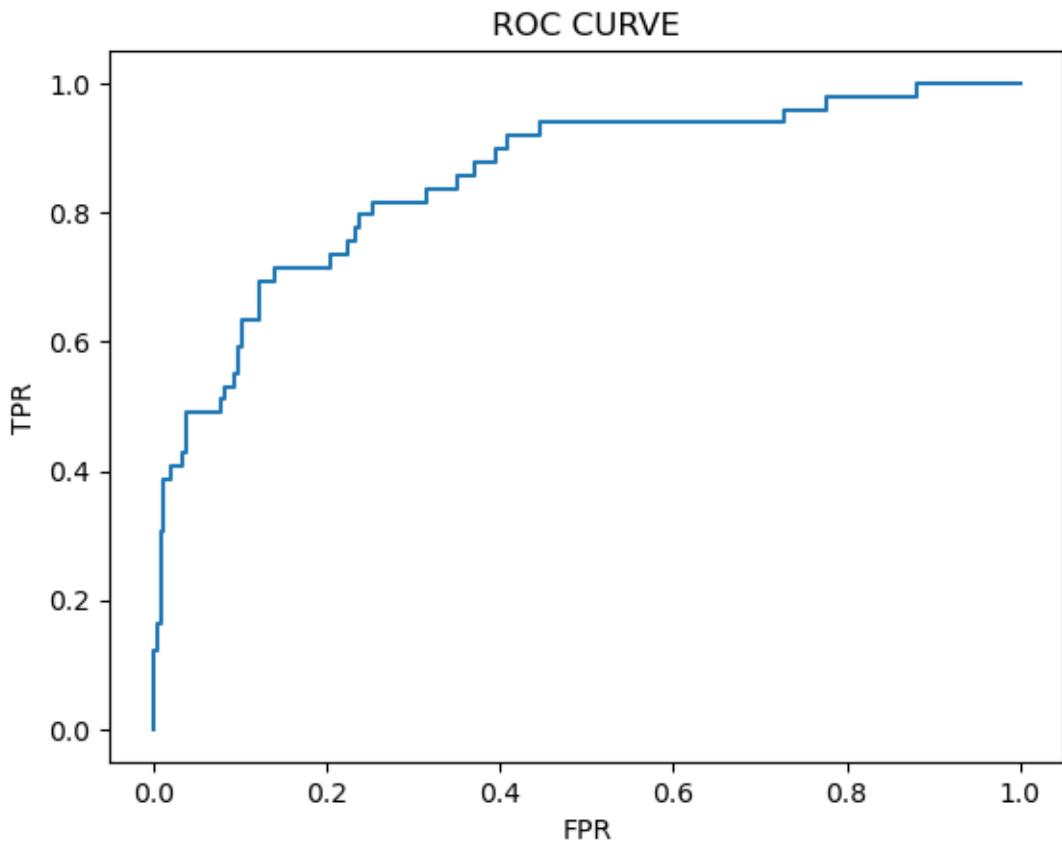
```
[51]: probability
```

```
[51]: array([0.16000127, 0.20600667, 0.31532384, 0.09242886, 0.63667551,
 0.06153061, 0.61819432, 0.0757087 , 0.00841372, 0.3912069 ,
 0.05398439, 0.33293123, 0.02020698, 0.67215483, 0.19786547,
 0.03454902, 0.11043981, 0.17101703, 0.04477777, 0.22783614,
 0.2335018 , 0.01553905, 0.06464492, 0.05029956, 0.58792413,
 0.44849464, 0.07412714, 0.04460935, 0.67666632, 0.0584383 ,
 0.01599026, 0.03521098, 0.06963085, 0.17397462, 0.07830857,
 0.04288032, 0.08150424, 0.07106342, 0.03622137, 0.05223965,
 0.04862098, 0.02091497, 0.01819361, 0.01362467, 0.02873997,
 0.50236969, 0.41553218, 0.00306874, 0.73976412, 0.51382382,
 0.09637213, 0.48845516, 0.08036228, 0.25757243, 0.66516772,
 0.26308027, 0.01964858, 0.30198497, 0.02919946, 0.16038964,
 0.02102747, 0.21670232, 0.13981568, 0.0358316 , 0.37208403,
 0.03002317, 0.29091186, 0.16041142, 0.10437497, 0.08695177,
 0.08217589, 0.30984518, 0.08531362, 0.07420689, 0.12268651,
 0.06192552, 0.04640904, 0.07624712, 0.19738483, 0.03236316,
 0.00884439, 0.0244108 , 0.13635803, 0.0260104 , 0.03341008,
 0.08186888, 0.00499397, 0.03474852, 0.03858027, 0.14602694,
 0.26167665, 0.16667357, 0.27400109, 0.24159565, 0.02160421,
 0.17748606, 0.34076078, 0.28022482, 0.06914126, 0.05003806,
 0.24437761, 0.74698271, 0.35438567, 0.01920627, 0.08778845,
 0.03255847, 0.05461351, 0.15123251, 0.06843702, 0.13752637,
 0.09584388, 0.04669882, 0.02493091, 0.15383171, 0.07081259,
 0.03089296, 0.0537667 , 0.11554316, 0.00881616, 0.01263271,
 0.17552253, 0.05045234, 0.08823238, 0.82995757, 0.03017756,
 0.0236819 , 0.0087012 , 0.1349589 , 0.16474801, 0.05202613,
 0.01524549, 0.29278083, 0.54767448, 0.34275448, 0.04629541,
 0.38966344, 0.61333366, 0.14552367, 0.07402366, 0.24143471,
 0.09418418, 0.0689069 , 0.10061956, 0.19346327, 0.20026293,
 0.03004939, 0.14900424, 0.00348846, 0.11225149, 0.15843155,
 0.06047573, 0.18601882, 0.06085869, 0.12221317, 0.03280184,
 0.02738799, 0.06356425, 0.08302382, 0.01541716, 0.014665 ,
 0.38517822, 0.01264231, 0.14961974, 0.80508787, 0.11598661,
```

```
0.2842811 , 0.17020143, 0.1530583 , 0.02764153, 0.00613226,
0.04191632, 0.09782393, 0.11551417, 0.10377982, 0.01779313,
0.14371315, 0.10615435, 0.10298963, 0.05132621, 0.09061081,
0.02897383, 0.09924087, 0.00512032, 0.75108423, 0.04296968,
0.04062134, 0.37518972, 0.04563128, 0.7251816 , 0.10671665,
0.36949086, 0.38146941, 0.32095493, 0.05266802, 0.08172004,
0.13947833, 0.04334317, 0.01469593, 0.26413988, 0.06330966,
0.1614747 , 0.15380517, 0.67152357, 0.05840793, 0.27891823,
0.04512564, 0.46033865, 0.00348431, 0.14068967, 0.02747401,
0.12714133, 0.17284246, 0.07341066, 0.10099827, 0.16870885,
0.02560842, 0.01824031, 0.08670796, 0.02834237, 0.13710215,
0.08778935, 0.2200061 , 0.73401148, 0.15938978, 0.4095449 ,
0.01513845, 0.11306309, 0.21497506, 0.32337575, 0.03409266,
0.04256318, 0.32157531, 0.05454465, 0.02348479, 0.16423352,
0.32696147, 0.22892063, 0.00877159, 0.08198819, 0.01156361,
0.1408691 , 0.29235147, 0.01270305, 0.17329916, 0.04081391,
0.04094165, 0.42771425, 0.34958286, 0.03766772, 0.12025286,
0.37698923, 0.3192629 , 0.79559338, 0.05385659, 0.21597037,
0.06383728, 0.00570991, 0.66018187, 0.35855286, 0.37783606,
0.36781398, 0.03554512, 0.21718203, 0.05943622, 0.06554485,
0.10081475, 0.00818713, 0.26591316, 0.42809675, 0.06542835,
0.09296803, 0.01259826, 0.14226651, 0.05072662, 0.02372258,
0.02586923, 0.06760427, 0.24315648, 0.26961432, 0.19831733,
0.2652296 , 0.0165923 , 0.15784236, 0.08398982, 0.02711775,
0.18750547, 0.00783535, 0.2844239 , 0.00270742, 0.02484969,
0.22585745, 0.72775605, 0.07691547, 0.26304359])
```

```
[52]: fpr,tpr,thresholds = roc_curve(y_test,probability)
```

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



### 3 DecisionTree

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

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

```
[55]: DecisionTreeClassifier()
```

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

```
[56]: array([0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 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, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

[57]: y\_test

```
[57]: 442      0  
    1091     0  
    981      1  
    785      0  
    1332     1  
    ..  
    1439     0  
    481      0  
    124      1  
    198      0  
    1229     0  
  
Name: Attrition, Length: 294, dtype: int32
```

[58] : df

[58]:	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	\
0	41	1	2	1102	2		1
1	49	0	1	279	1		8
2	37	1	2	1373	1		2
3	33	0	1	1392	1		3
4	27	0	2	591	1		2
...	...	...	...	...	...	...	\
1465	36	0	1	884	1		23
1466	39	0	2	613	1		6
1467	27	0	2	155	1		4
1468	49	0	1	1023	2		2
1469	34	0	2	628	1		8
	Education	EducationField	EmployeeCount	EmployeeNumber	...	...	\
0	2	1	1	1	1	...	\
1	1	1	1	1	2	...	\
2	2	4	1	1	4	...	\
3	4	1	1	1	5	...	\
4	1	3	1	1	7	...	\
...	...	...	...	...	...	...	\
1465	2	3	1	1	2061	...	\
1466	1	3	1	1	2062	...	\

1467	3	1	1	2064	...
1468	3	3	1	2065	...
1469	3	3	1	2068	...
0		1	80	0	
1		4	80	1	
2		2	80	0	
3		3	80	0	
4		4	80	1	
...	...	...	...	...	\\
1465		3	80	1	
1466		1	80	1	
1467		2	80	1	
1468		4	80	0	
1469		1	80	0	
0	TotalWorkingYears	TrainingTimesLastYear	WorkLifeBalance	\\	
1	8	0	1		
2	10	3	3		
3	7	3	3		
4	8	3	3		
...	...	...	...	...	\\
1465	17	3	3		
1466	9	5	3		
1467	6	0	3		
1468	17	3	2		
1469	6	3	4		
0	YearsAtCompany	YearsInCurrentRole	YearsSinceLastPromotion	\\	
1	6	4	0		
2	10	7	1		
3	0	0	0		
4	8	7	3		
5	2	2	2		
...	...	...	...	...	\\
1465	5	2	0		
1466	7	7	1		
1467	6	2	0		
1468	9	6	0		
1469	4	3	1		
0	YearsWithCurrManager				
1	5				
2	7				
3	0				

```
3          0
4          2
...
1465      ...
1466      3
1467      7
1467      3
1468      8
1469      2
```

[1470 rows x 35 columns]

```
[59]: dtc.predict(ms)
      ↪transform([[41,2,1102,2,1,2,1,1,1,2,0,94,3,2,7,4,2,5993,19479,8,0,1,11,3,1,80,0,0,8,0,1,6,4,0

C:\Users\mb419\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\mb419\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(
[59]: array([1])

[60]: from sklearn.metrics import_
      ↪accuracy_score,confusion_matrix,classification_report,roc_auc_score,roc_curve

[61]: accuracy_score(y_test,pred)

[61]: 0.7517006802721088

[62]: confusion_matrix(y_test,pred)

[62]: array([[205,  40],
           [ 33,  16]], dtype=int64)

[63]: pd.crosstab(y_test,pred)

[63]: col_0      0   1
      Attrition
      0        205  40
      1        33   16

[64]: a2 = (209+18)/294 #accuracy
      a2

[64]: 0.7721088435374149
```

```
[65]: r2 = 18/(31+18) #recall  
r2
```

[65]: 0.3673469387755102

```
[66]: p2 = 18/(36+18) #precision  
p2
```

[66]: 0.3333333333333333

```
[67]: f2 = 2*p2*r2/(p2+r2) # F1 score  
f2
```

[67]: 0.34951456310679613

```
[68]: print(classification_report(y_test,pred))
```

	precision	recall	f1-score	support
0	0.86	0.84	0.85	245
1	0.29	0.33	0.30	49
accuracy			0.75	294
macro avg	0.57	0.58	0.58	294
weighted avg	0.77	0.75	0.76	294

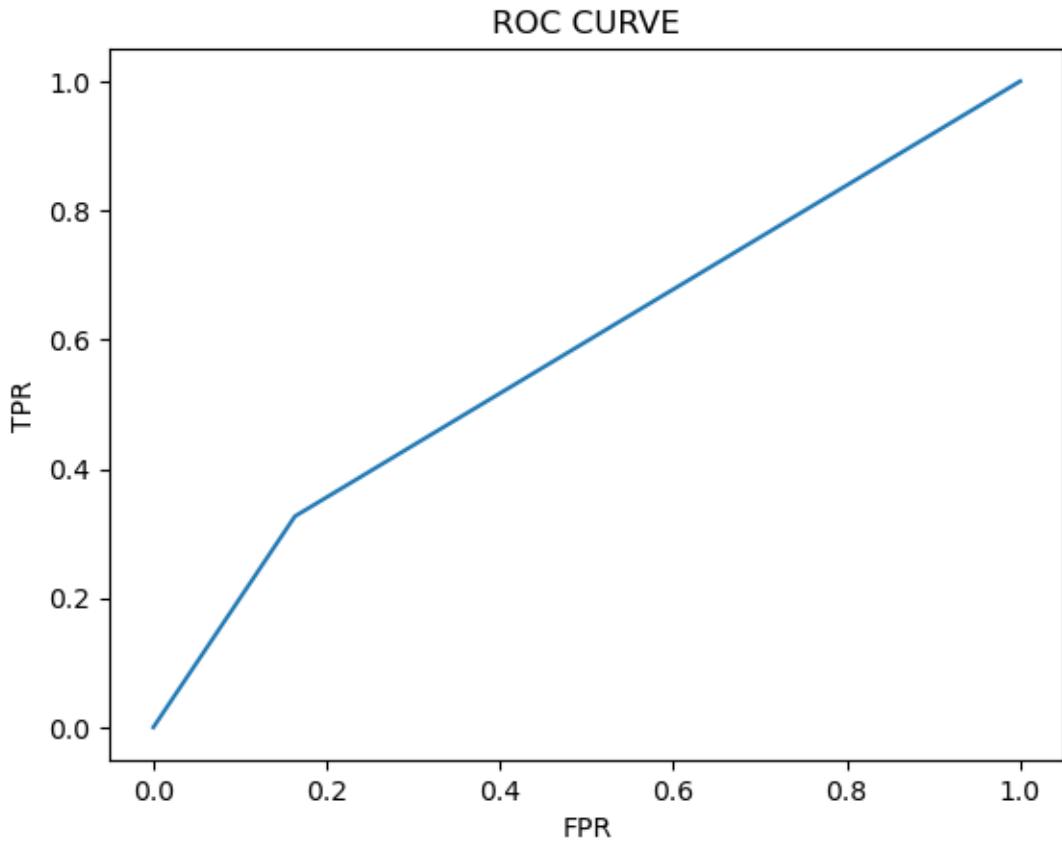
```
[69]: probability=dtc.predict_proba(x_test)[:,1]
```

[70]: probability

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

```
[71]: fpr,tpr,thresholds = roc_curve(y_test,probability)
```

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



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

```
[73]: [Text(0.32333245798319327, 0.9722222222222222, 'x[27] <= 0.038\nngini =  
0.269\nnsamples = 1176\nnvalue = [988, 188]'),  
Text(0.07394957983193277, 0.9166666666666666, 'x[16] <= 0.75\nngini =  
0.5\nnsamples = 78\nnvalue = [39, 39]'),  
Text(0.043697478991596636, 0.8611111111111112, 'x[4] <= 0.554\nngini =  
0.426\nnsamples = 39\nnvalue = [27, 12]'),
```

```

Text(0.02689075630252101, 0.8055555555555556, 'x[15] <= 0.167\ngini =
0.312\nsamples = 31\nvalue = [25, 6']),
Text(0.013445378151260505, 0.75, 'x[17] <= 0.057\ngini = 0.49\nsamples =
7\nvalue = [3, 4']),
Text(0.0067226890756302525, 0.6944444444444444, 'gini = 0.0\nsamples = 3\nvalue
= [0, 3']),
Text(0.020168067226890758, 0.6944444444444444, 'x[18] <= 0.251\ngini =
0.375\nsamples = 4\nvalue = [3, 1']),
Text(0.013445378151260505, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue
= [0, 1']),
Text(0.02689075630252101, 0.6388888888888888, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0']),
Text(0.040336134453781515, 0.75, 'x[19] <= 0.056\ngini = 0.153\nsamples =
24\nvalue = [22, 2']),
Text(0.03361344537815126, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1']),
Text(0.047058823529411764, 0.6944444444444444, 'x[9] <= 0.167\ngini =
0.083\nsamples = 23\nvalue = [22, 1']),
Text(0.040336134453781515, 0.6388888888888888, 'x[16] <= 0.25\ngini =
0.5\nsamples = 2\nvalue = [1, 1']),
Text(0.03361344537815126, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1']),
Text(0.047058823529411764, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0']),
Text(0.05378151260504202, 0.6388888888888888, 'gini = 0.0\nsamples = 21\nvalue =
[21, 0']),
Text(0.06050420168067227, 0.8055555555555556, 'x[22] <= 0.679\ngini =
0.375\nsamples = 8\nvalue = [2, 6']),
Text(0.05378151260504202, 0.75, 'gini = 0.0\nsamples = 6\nvalue = [0, 6']),
Text(0.06722689075630252, 0.75, 'gini = 0.0\nsamples = 2\nvalue = [2, 0']),
Text(0.10420168067226891, 0.8611111111111112, 'x[11] <= 0.364\ngini =
0.426\nsamples = 39\nvalue = [12, 27]),
Text(0.08739495798319327, 0.8055555555555556, 'x[17] <= 0.1\ngini =
0.133\nsamples = 14\nvalue = [1, 13']),
Text(0.08067226890756303, 0.75, 'gini = 0.0\nsamples = 13\nvalue = [0, 13']),
Text(0.09411764705882353, 0.75, 'gini = 0.0\nsamples = 1\nvalue = [1, 0']),
Text(0.12100840336134454, 0.8055555555555556, 'x[8] <= 0.105\ngini =
0.493\nsamples = 25\nvalue = [11, 14]),
Text(0.10756302521008404, 0.75, 'x[21] <= 0.5\ngini = 0.278\nsamples = 6\nvalue =
[5, 1']),
Text(0.10084033613445378, 0.6944444444444444, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0']),
Text(0.11428571428571428, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1']),
Text(0.13445378151260504, 0.75, 'x[15] <= 0.5\ngini = 0.432\nsamples =
19\nvalue = [6, 13]),
Text(0.12773109243697478, 0.6944444444444444, 'gini = 0.0\nsamples = 7\nvalue =

```

```

[0, 7]),
Text(0.1411764705882353, 0.6944444444444444, 'x[6] <= 0.4\ngini = 0.5\nsamples
= 12\nvalue = [6, 6]),
Text(0.12773109243697478, 0.6388888888888888, 'x[3] <= 0.75\ngini =
0.278\nsamples = 6\nvalue = [5, 1]),
Text(0.12100840336134454, 0.5833333333333334, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]),
Text(0.13445378151260504, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.1546218487394958, 0.6388888888888888, 'x[8] <= 0.249\ngini =
0.278\nsamples = 6\nvalue = [1, 5]),
Text(0.14789915966386555, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]),
Text(0.16134453781512606, 0.5833333333333334, 'gini = 0.0\nsamples = 5\nvalue =
[0, 5]),
Text(0.5727153361344538, 0.9166666666666666, 'x[21] <= 0.5\ngini =
0.235\nsamples = 1098\nvalue = [949, 149]),
Text(0.32358193277310926, 0.8611111111111112, 'x[29] <= 0.167\ngini =
0.162\nsamples = 798\nvalue = [727, 71]),
Text(0.17478991596638654, 0.8055555555555556, 'x[8] <= 0.445\ngini =
0.38\nsamples = 47\nvalue = [35, 12]),
Text(0.16134453781512606, 0.75, 'x[16] <= 0.75\ngini = 0.1\nsamples = 19\nvalue
= [18, 1]),
Text(0.1546218487394958, 0.6944444444444444, 'gini = 0.0\nsamples = 18\nvalue =
[18, 0]),
Text(0.16806722689075632, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.18823529411764706, 0.75, 'x[17] <= 0.094\ngini = 0.477\nsamples =
28\nvalue = [17, 11]),
Text(0.1815126050420168, 0.6944444444444444, 'gini = 0.0\nsamples = 4\nvalue =
[0, 4]),
Text(0.1949579831932773, 0.6944444444444444, 'x[8] <= 0.524\ngini =
0.413\nsamples = 24\nvalue = [17, 7]),
Text(0.18823529411764706, 0.6388888888888888, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]),
Text(0.20168067226890757, 0.6388888888888888, 'x[33] <= 0.324\ngini =
0.351\nsamples = 22\nvalue = [17, 5]),
Text(0.18823529411764706, 0.5833333333333334, 'x[2] <= 0.025\ngini =
0.133\nsamples = 14\nvalue = [13, 1]),
Text(0.1815126050420168, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.1949579831932773, 0.5277777777777778, 'gini = 0.0\nsamples = 13\nvalue =
[13, 0]),
Text(0.21512605042016808, 0.5833333333333334, 'x[2] <= 0.329\ngini =
0.5\nsamples = 8\nvalue = [4, 4]),
Text(0.20840336134453782, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]),

```

```

Text(0.2218487394957983, 0.5277777777777778, 'x[19] <= 0.444\ngini =
0.32\nsamples = 5\nvalue = [4, 1']),
Text(0.21512605042016808, 0.4722222222222222, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0']),
Text(0.22857142857142856, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1']),
Text(0.4723739495798319, 0.8055555555555556, 'x[30] <= 0.963\ngini =
0.145\nsamples = 751\nvalue = [692, 59']),
Text(0.4656512605042017, 0.75, 'x[30] <= 0.113\ngini = 0.143\nsamples =
750\nvalue = [692, 58']),
Text(0.34243697478991597, 0.6944444444444444, 'x[9] <= 0.167\ngini =
0.218\nsamples = 257\nvalue = [225, 32']),
Text(0.30084033613445377, 0.6388888888888888, 'x[33] <= 0.147\ngini =
0.355\nsamples = 65\nvalue = [50, 15']),
Text(0.27899159663865547, 0.5833333333333334, 'x[33] <= 0.029\ngini =
0.303\nsamples = 59\nvalue = [48, 11']),
Text(0.25546218487394956, 0.5277777777777778, 'x[12] <= 0.5\ngini =
0.463\nsamples = 22\nvalue = [14, 8']),
Text(0.24201680672268908, 0.4722222222222222, 'x[11] <= 0.179\ngini =
0.198\nsamples = 9\nvalue = [8, 1']),
Text(0.23529411764705882, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1']),
Text(0.24873949579831933, 0.4166666666666667, 'gini = 0.0\nsamples = 8\nvalue =
[8, 0']),
Text(0.2689075630252101, 0.4722222222222222, 'x[11] <= 0.4\ngini =
0.497\nsamples = 13\nvalue = [6, 7']),
Text(0.26218487394957984, 0.4166666666666667, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0']),
Text(0.27563025210084036, 0.4166666666666667, 'x[4] <= 0.286\ngini =
0.346\nsamples = 9\nvalue = [2, 7']),
Text(0.2689075630252101, 0.3611111111111111, 'x[32] <= 0.1\ngini =
0.444\nsamples = 3\nvalue = [2, 1']),
Text(0.26218487394957984, 0.3055555555555556, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0']),
Text(0.27563025210084036, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1']),
Text(0.2823529411764706, 0.3611111111111111, 'gini = 0.0\nsamples = 6\nvalue =
[0, 6']),
Text(0.3025210084033613, 0.5277777777777778, 'x[15] <= 0.167\ngini =
0.149\nsamples = 37\nvalue = [34, 3']),
Text(0.2957983193277311, 0.4722222222222222, 'x[29] <= 0.5\ngini = 0.5\nsamples =
6\nvalue = [3, 3']),
Text(0.28907563025210087, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0']),
Text(0.3025210084033613, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]),
Text(0.3092436974789916, 0.4722222222222222, 'gini = 0.0\nsamples = 31\nvalue =

```

```

[31, 0']),
Text(0.3226890756302521, 0.5833333333333334, 'x[8] <= 0.065\ngini =
0.444\nsamples = 6\nvalue = [2, 4']),
Text(0.31596638655462184, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0']),
Text(0.32941176470588235, 0.5277777777777778, 'gini = 0.0\nsamples = 4\nvalue =
[0, 4']),
Text(0.3840336134453782, 0.6388888888888888, 'x[0] <= 0.321\ngini =
0.161\nsamples = 192\nvalue = [175, 17']),
Text(0.3495798319327731, 0.5833333333333334, 'x[6] <= 0.1\ngini =
0.294\nsamples = 67\nvalue = [55, 12']),
Text(0.34285714285714286, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2']),
Text(0.3563025210084034, 0.5277777777777778, 'x[29] <= 0.5\ngini =
0.26\nsamples = 65\nvalue = [55, 10']),
Text(0.33949579831932775, 0.4722222222222222, 'x[11] <= 0.679\ngini =
0.469\nsamples = 16\nvalue = [10, 6']),
Text(0.33277310924369746, 0.4166666666666667, 'x[4] <= 0.018\ngini =
0.444\nsamples = 9\nvalue = [3, 6']),
Text(0.32605042016806723, 0.3611111111111111, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0']),
Text(0.33949579831932775, 0.3611111111111111, 'x[8] <= 0.306\ngini =
0.245\nsamples = 7\nvalue = [1, 6']),
Text(0.33277310924369746, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0']),
Text(0.346218487394958, 0.3055555555555556, 'gini = 0.0\nsamples = 6\nvalue =
[0, 6']),
Text(0.346218487394958, 0.4166666666666667, 'gini = 0.0\nsamples = 7\nvalue =
[7, 0']),
Text(0.373109243697479, 0.4722222222222222, 'x[2] <= 0.037\ngini =
0.15\nsamples = 49\nvalue = [45, 4']),
Text(0.3663865546218487, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1']),
Text(0.3798319327731092, 0.4166666666666667, 'x[2] <= 0.938\ngini =
0.117\nsamples = 48\nvalue = [45, 3']),
Text(0.373109243697479, 0.3611111111111111, 'x[5] <= 0.875\ngini =
0.081\nsamples = 47\nvalue = [45, 2']),
Text(0.3596638655462185, 0.3055555555555556, 'x[12] <= 0.167\ngini =
0.043\nsamples = 45\nvalue = [44, 1']),
Text(0.35294117647058826, 0.25, 'x[3] <= 0.75\ngini = 0.444\nsamples = 3\nvalue =
[2, 1']),
Text(0.346218487394958, 0.1944444444444445, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0']),
Text(0.3596638655462185, 0.1944444444444445, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1']),
Text(0.3663865546218487, 0.25, 'gini = 0.0\nsamples = 42\nvalue = [42, 0']),
Text(0.3865546218487395, 0.3055555555555556, 'x[18] <= 0.436\ngini =

```

```

0.5\nsamples = 2\nvalue = [1, 1]),
Text(0.3798319327731092, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [1, 0']),
Text(0.39327731092436974, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1']),
Text(0.3865546218487395, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.4184873949579832, 0.5833333333333334, 'x[8] <= 0.022\ngini =
0.077\nsamples = 125\nvalue = [120, 5]),
Text(0.4, 0.5277777777777778, 'x[2] <= 0.578\ngini = 0.5\nsamples = 4\nvalue =
[2, 2]),
Text(0.39327731092436974, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]),
Text(0.40672268907563025, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]),
Text(0.4369747899159664, 0.5277777777777778, 'x[18] <= 0.968\ngini =
0.048\nsamples = 121\nvalue = [118, 3]),
Text(0.42016806722689076, 0.4722222222222222, 'x[2] <= 0.98\ngini =
0.033\nsamples = 118\nvalue = [116, 2]),
Text(0.40672268907563025, 0.4166666666666667, 'x[14] <= 0.938\ngini =
0.017\nsamples = 114\nvalue = [113, 1]),
Text(0.4, 0.3611111111111111, 'gini = 0.0\nsamples = 107\nvalue = [107, 0]),
Text(0.4134453781512605, 0.3611111111111111, 'x[11] <= 0.193\ngini =
0.245\nsamples = 7\nvalue = [6, 1]),
Text(0.40672268907563025, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.42016806722689076, 0.3055555555555556, 'gini = 0.0\nsamples = 6\nvalue =
[6, 0]),
Text(0.4336134453781513, 0.4166666666666667, 'x[3] <= 0.75\ngini =
0.375\nsamples = 4\nvalue = [3, 1]),
Text(0.426890756302521, 0.3611111111111111, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]),
Text(0.4403361344537815, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.453781512605042, 0.4722222222222222, 'x[32] <= 0.033\ngini =
0.444\nsamples = 3\nvalue = [2, 1]),
Text(0.4470588235294118, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.46050420168067224, 0.4166666666666667, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]),
Text(0.5888655462184874, 0.6944444444444444, 'x[30] <= 0.787\ngini =
0.1\nsamples = 493\nvalue = [467, 26]),
Text(0.5542016806722689, 0.6388888888888888, 'x[15] <= 0.5\ngini =
0.094\nsamples = 486\nvalue = [462, 24]),
Text(0.5050420168067227, 0.5833333333333334, 'x[14] <= 0.938\ngini =
0.154\nsamples = 191\nvalue = [175, 16]),
Text(0.49831932773109244, 0.5277777777777778, 'x[18] <= 0.481\ngini =
0.145\nsamples = 190\nvalue = [175, 15]),
Text(0.48067226890756304, 0.4722222222222222, 'x[18] <= 0.47\ngini =

```

```

0.221\nsamples = 95\nvalue = [83, 12']),
Text(0.47394957983193275, 0.4166666666666667, 'x[33] <= 0.794\ngini =
0.207\nsamples = 94\nvalue = [83, 11']),
Text(0.4672268907563025, 0.3611111111111111, 'x[5] <= 0.375\ngini =
0.192\nsamples = 93\nvalue = [83, 10']),
Text(0.44537815126050423, 0.3055555555555556, 'x[6] <= 0.9\ngini =
0.363\nsamples = 21\nvalue = [16, 5']),
Text(0.43865546218487395, 0.25, 'x[17] <= 0.413\ngini = 0.266\nsamples =
19\nvalue = [16, 3']),
Text(0.42521008403361343, 0.1944444444444445, 'x[17] <= 0.141\ngini =
0.117\nsamples = 16\nvalue = [15, 1']),
Text(0.4184873949579832, 0.1388888888888889, 'x[5] <= 0.125\ngini =
0.5\nsamples = 2\nvalue = [1, 1']),
Text(0.4117647058823529, 0.0833333333333333, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0']),
Text(0.42521008403361343, 0.0833333333333333, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1']),
Text(0.4319327731092437, 0.1388888888888889, 'gini = 0.0\nsamples = 14\nvalue =
[14, 0']),
Text(0.45210084033613446, 0.1944444444444445, 'x[18] <= 0.006\ngini =
0.444\nsamples = 3\nvalue = [1, 2']),
Text(0.44537815126050423, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0']),
Text(0.4588235294117647, 0.1388888888888889, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2']),
Text(0.45210084033613446, 0.25, 'gini = 0.0\nsamples = 2\nvalue = [0, 2']),
Text(0.4890756302521008, 0.3055555555555556, 'x[31] <= 0.139\ngini =
0.129\nsamples = 72\nvalue = [67, 5']),
Text(0.4722689075630252, 0.25, 'x[8] <= 0.68\ngini = 0.444\nsamples = 6\nvalue =
[4, 2']),
Text(0.46554621848739497, 0.1944444444444445, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0']),
Text(0.4789915966386555, 0.1944444444444445, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2']),
Text(0.5058823529411764, 0.25, 'x[11] <= 0.993\ngini = 0.087\nsamples =
66\nvalue = [63, 3']),
Text(0.492436974789916, 0.1944444444444445, 'x[28] <= 0.583\ngini =
0.061\nsamples = 64\nvalue = [62, 2']),
Text(0.4857142857142857, 0.1388888888888889, 'gini = 0.0\nsamples = 51\nvalue =
[51, 0']),
Text(0.4991596638655462, 0.1388888888888889, 'x[3] <= 0.75\ngini =
0.26\nsamples = 13\nvalue = [11, 2']),
Text(0.492436974789916, 0.0833333333333333, 'gini = 0.0\nsamples = 9\nvalue =
[9, 0']),
Text(0.5058823529411764, 0.0833333333333333, 'x[9] <= 0.5\ngini = 0.5\nsamples =
4\nvalue = [2, 2']),
Text(0.4991596638655462, 0.02777777777777776, 'gini = 0.0\nsamples = 2\nvalue

```

```

= [0, 2]),
Text(0.5126050420168067, 0.027777777777777776, 'gini = 0.0\nsamples = 2\nvalue
= [2, 0]),
Text(0.519327731092437, 0.19444444444444445, 'x[2] <= 0.922\ngini =
0.5\nsamples = 2\nvalue = [1, 1]),
Text(0.5126050420168067, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]),
Text(0.5260504201680672, 0.1388888888888889, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.48067226890756304, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.48739495798319327, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.5159663865546219, 0.4722222222222222, 'x[19] <= 0.5\ngini =
0.061\nsamples = 95\nvalue = [92, 3]),
Text(0.5092436974789916, 0.4166666666666667, 'gini = 0.0\nsamples = 76\nvalue =
[76, 0]),
Text(0.5226890756302521, 0.4166666666666667, 'x[8] <= 0.161\ngini =
0.266\nsamples = 19\nvalue = [16, 3]),
Text(0.5092436974789916, 0.3611111111111111, 'x[18] <= 0.669\ngini =
0.444\nsamples = 3\nvalue = [1, 2]),
Text(0.5025210084033613, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]),
Text(0.5159663865546219, 0.3055555555555556, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]),
Text(0.5361344537815126, 0.3611111111111111, 'x[31] <= 0.639\ngini =
0.117\nsamples = 16\nvalue = [15, 1]),
Text(0.5294117647058824, 0.3055555555555556, 'gini = 0.0\nsamples = 14\nvalue =
[14, 0]),
Text(0.5428571428571428, 0.3055555555555556, 'x[19] <= 0.944\ngini =
0.5\nsamples = 2\nvalue = [1, 1]),
Text(0.5361344537815126, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]),
Text(0.5495798319327732, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]),
Text(0.5117647058823529, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.6033613445378151, 0.583333333333334, 'x[22] <= 0.036\ngini =
0.053\nsamples = 295\nvalue = [287, 8]),
Text(0.5798319327731093, 0.5277777777777778, 'x[32] <= 0.7\ngini =
0.159\nsamples = 46\nvalue = [42, 4]),
Text(0.573109243697479, 0.4722222222222222, 'x[11] <= 0.071\ngini =
0.124\nsamples = 45\nvalue = [42, 3]),
Text(0.5563025210084034, 0.4166666666666667, 'x[2] <= 0.489\ngini =
0.5\nsamples = 2\nvalue = [1, 1]),
Text(0.5495798319327732, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]),
Text(0.5630252100840336, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),

```

```

Text(0.5899159663865546, 0.4166666666666667, 'x[27] <= 0.688\ngini =
0.089\nsamples = 43\nvalue = [41, 2']),
Text(0.5764705882352941, 0.3611111111111111, 'x[14] <= 0.062\ngini =
0.048\nsamples = 41\nvalue = [40, 1']),
Text(0.5697478991596638, 0.3055555555555556, 'x[30] <= 0.237\ngini =
0.375\nsamples = 4\nvalue = [3, 1']),
Text(0.5630252100840336, 0.25, 'gini = 0.0\nsamples = 3\nvalue = [3, 0']),
Text(0.5764705882352941, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]),
Text(0.5831932773109244, 0.3055555555555556, 'gini = 0.0\nsamples = 37\nvalue =
[37, 0]),
Text(0.6033613445378151, 0.3611111111111111, 'x[8] <= 0.536\ngini =
0.5\nsamples = 2\nvalue = [1, 1']),
Text(0.5966386554621849, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]),
Text(0.6100840336134454, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.5865546218487395, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.626890756302521, 0.5277777777777778, 'x[17] <= 0.056\ngini =
0.032\nsamples = 249\nvalue = [245, 4]),
Text(0.6100840336134454, 0.4722222222222222, 'x[16] <= 0.75\ngini =
0.32\nsamples = 5\nvalue = [4, 1]),
Text(0.6033613445378151, 0.4166666666666667, 'gini = 0.0\nsamples = 4\nvalue =
[4, 0]),
Text(0.6168067226890757, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.6436974789915967, 0.4722222222222222, 'x[2] <= 0.015\ngini =
0.024\nsamples = 244\nvalue = [241, 3]),
Text(0.6302521008403361, 0.4166666666666667, 'x[18] <= 0.715\ngini =
0.278\nsamples = 6\nvalue = [5, 1]),
Text(0.6235294117647059, 0.3611111111111111, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]),
Text(0.6369747899159663, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.6571428571428571, 0.4166666666666667, 'x[24] <= 0.167\ngini =
0.017\nsamples = 238\nvalue = [236, 2]),
Text(0.6504201680672269, 0.3611111111111111, 'x[29] <= 0.833\ngini =
0.073\nsamples = 53\nvalue = [51, 2]),
Text(0.6369747899159663, 0.3055555555555556, 'x[33] <= 0.088\ngini =
0.041\nsamples = 48\nvalue = [47, 1]),
Text(0.6302521008403361, 0.25, 'x[18] <= 0.824\ngini = 0.245\nsamples =
7\nvalue = [6, 1]),
Text(0.6235294117647059, 0.1944444444444445, 'gini = 0.0\nsamples = 6\nvalue =
[6, 0]),
Text(0.6369747899159663, 0.1944444444444445, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.6436974789915967, 0.25, 'gini = 0.0\nsamples = 41\nvalue = [41, 0]),

```

```

Text(0.6638655462184874, 0.3055555555555556, 'x[0] <= 0.631\ngini =
0.32\nsamples = 5\nvalue = [4, 1']),
Text(0.6571428571428571, 0.25, 'gini = 0.0\nsamples = 4\nvalue = [4, 0']),
Text(0.6705882352941176, 0.25, 'gini = 0.0\nsamples = 1\nvalue = [0, 1']),
Text(0.6638655462184874, 0.3611111111111111, 'gini = 0.0\nsamples = 185\nvalue
= [185, 0]),
Text(0.6235294117647059, 0.6388888888888888, 'x[2] <= 0.366\ngini =
0.408\nsamples = 7\nvalue = [5, 2']),
Text(0.6168067226890757, 0.5833333333333334, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]),
Text(0.6302521008403361, 0.5833333333333334, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]),
Text(0.4790966386554622, 0.75, 'gini = 0.0\nsamples = 1\nvalue = [0, 1']),
Text(0.8218487394957983, 0.8611111111111112, 'x[17] <= 0.157\ngini =
0.385\nsamples = 300\nvalue = [222, 78']),
Text(0.7336134453781512, 0.8055555555555556, 'x[26] <= 0.167\ngini =
0.5\nsamples = 96\nvalue = [49, 47]),
Text(0.6974789915966386, 0.75, 'x[4] <= 0.161\ngini = 0.459\nsamples =
42\nvalue = [15, 27']),
Text(0.6705882352941176, 0.6944444444444444, 'x[8] <= 0.415\ngini =
0.499\nsamples = 23\nvalue = [12, 11]),
Text(0.6504201680672269, 0.6388888888888888, 'x[18] <= 0.561\ngini =
0.355\nsamples = 13\nvalue = [3, 10]),
Text(0.6436974789915967, 0.5833333333333334, 'gini = 0.0\nsamples = 8\nvalue =
[0, 8]),
Text(0.6571428571428571, 0.5833333333333334, 'x[17] <= 0.108\ngini =
0.48\nsamples = 5\nvalue = [3, 2]),
Text(0.6504201680672269, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0]),
Text(0.6638655462184874, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]),
Text(0.6907563025210084, 0.6388888888888888, 'x[27] <= 0.1\ngini =
0.18\nsamples = 10\nvalue = [9, 1]),
Text(0.6840336134453782, 0.5833333333333334, 'x[24] <= 0.333\ngini =
0.5\nsamples = 2\nvalue = [1, 1]),
Text(0.6773109243697479, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.6907563025210084, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]),
Text(0.6974789915966386, 0.5833333333333334, 'gini = 0.0\nsamples = 8\nvalue =
[8, 0]),
Text(0.7243697478991596, 0.6944444444444444, 'x[27] <= 0.35\ngini =
0.266\nsamples = 19\nvalue = [3, 16]),
Text(0.7176470588235294, 0.6388888888888888, 'x[11] <= 0.2\ngini =
0.198\nsamples = 18\nvalue = [2, 16]),
Text(0.7109243697478992, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]),

```

```

Text(0.7243697478991596, 0.5833333333333334, 'x[0] <= 0.595\ngini =
0.111\nsamples = 17\nvalue = [1, 16]'),
Text(0.7176470588235294, 0.5277777777777778, 'gini = 0.0\nsamples = 15\nvalue =
[0, 15]'),
Text(0.7310924369747899, 0.5277777777777778, 'x[24] <= 0.833\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.7243697478991596, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
Text(0.7378151260504202, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
Text(0.7310924369747899, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
Text(0.7697478991596639, 0.75, 'x[0] <= 0.202\ngini = 0.466\nsamples =
54\nvalue = [34, 20]'),
Text(0.7512605042016807, 0.6944444444444444, 'x[0] <= 0.107\ngini =
0.245\nsamples = 7\nvalue = [1, 6]'),
Text(0.7445378151260504, 0.6388888888888888, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
Text(0.7579831932773109, 0.6388888888888888, 'gini = 0.0\nsamples = 6\nvalue =
[0, 6]'),
Text(0.788235294117647, 0.6944444444444444, 'x[2] <= 0.622\ngini =
0.418\nsamples = 47\nvalue = [33, 14]'),
Text(0.7714285714285715, 0.6388888888888888, 'x[2] <= 0.145\ngini =
0.482\nsamples = 32\nvalue = [19, 13]'),
Text(0.7579831932773109, 0.5833333333333334, 'x[30] <= 0.237\ngini =
0.18\nsamples = 10\nvalue = [9, 1]'),
Text(0.7512605042016807, 0.5277777777777778, 'gini = 0.0\nsamples = 9\nvalue =
[9, 0]'),
Text(0.7647058823529411, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]'),
Text(0.7848739495798319, 0.5833333333333334, 'x[18] <= 0.87\ngini =
0.496\nsamples = 22\nvalue = [10, 12]'),
Text(0.7781512605042017, 0.5277777777777778, 'x[8] <= 0.41\ngini =
0.465\nsamples = 19\nvalue = [7, 12]'),
Text(0.7647058823529411, 0.4722222222222222, 'x[18] <= 0.715\ngini =
0.469\nsamples = 8\nvalue = [5, 3]'),
Text(0.7579831932773109, 0.4166666666666667, 'gini = 0.0\nsamples = 5\nvalue =
[5, 0]'),
Text(0.7714285714285715, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]'),
Text(0.7915966386554621, 0.4722222222222222, 'x[0] <= 0.25\ngini =
0.298\nsamples = 11\nvalue = [2, 9]'),
Text(0.7848739495798319, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]'),
Text(0.7983193277310925, 0.4166666666666667, 'x[14] <= 0.188\ngini =
0.18\nsamples = 10\nvalue = [1, 9]'),
Text(0.7915966386554621, 0.3611111111111111, 'x[0] <= 0.44\ngini = 0.5\nsamples

```

```

= 2\nvalue = [1, 1']),
Text(0.7848739495798319, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1']),
Text(0.7983193277310925, 0.3055555555555556, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0']),
Text(0.8050420168067227, 0.3611111111111111, 'gini = 0.0\nsamples = 8\nvalue =
[0, 8']),
Text(0.7915966386554621, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue =
[3, 0']),
Text(0.8050420168067227, 0.6388888888888888, 'x[19] <= 0.944\ngini =
0.124\nsamples = 15\nvalue = [14, 1']),
Text(0.7983193277310925, 0.5833333333333334, 'gini = 0.0\nsamples = 14\nvalue =
[14, 0']),
Text(0.8117647058823529, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1']),
Text(0.9100840336134454, 0.8055555555555556, 'x[16] <= 0.75\ngini =
0.258\nsamples = 204\nvalue = [173, 31']),
Text(0.8571428571428571, 0.75, 'x[17] <= 0.992\ngini = 0.138\nsamples =
147\nvalue = [136, 11']),
Text(0.8504201680672269, 0.6944444444444444, 'x[4] <= 0.482\ngini =
0.128\nsamples = 146\nvalue = [136, 10']),
Text(0.8319327731092437, 0.6388888888888888, 'x[30] <= 0.063\ngini =
0.038\nsamples = 104\nvalue = [102, 2']),
Text(0.8252100840336134, 0.5833333333333334, 'x[11] <= 0.193\ngini =
0.32\nsamples = 10\nvalue = [8, 2']),
Text(0.8184873949579832, 0.5277777777777778, 'x[17] <= 0.599\ngini =
0.444\nsamples = 3\nvalue = [1, 2']),
Text(0.8117647058823529, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]),
Text(0.8252100840336134, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0']),
Text(0.8319327731092437, 0.5277777777777778, 'gini = 0.0\nsamples = 7\nvalue =
[7, 0']),
Text(0.838655462184874, 0.5833333333333334, 'gini = 0.0\nsamples = 94\nvalue =
[94, 0]),
Text(0.8689075630252101, 0.6388888888888888, 'x[9] <= 0.167\ngini =
0.308\nsamples = 42\nvalue = [34, 8']),
Text(0.8521008403361344, 0.5833333333333334, 'x[2] <= 0.736\ngini =
0.375\nsamples = 4\nvalue = [1, 3']),
Text(0.8453781512605042, 0.5277777777777778, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]),
Text(0.8588235294117647, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]),
Text(0.8857142857142857, 0.5833333333333334, 'x[0] <= 0.393\ngini =
0.229\nsamples = 38\nvalue = [33, 5']),
Text(0.8722689075630252, 0.5277777777777778, 'x[11] <= 0.643\ngini =
0.5\nsamples = 6\nvalue = [3, 3']),

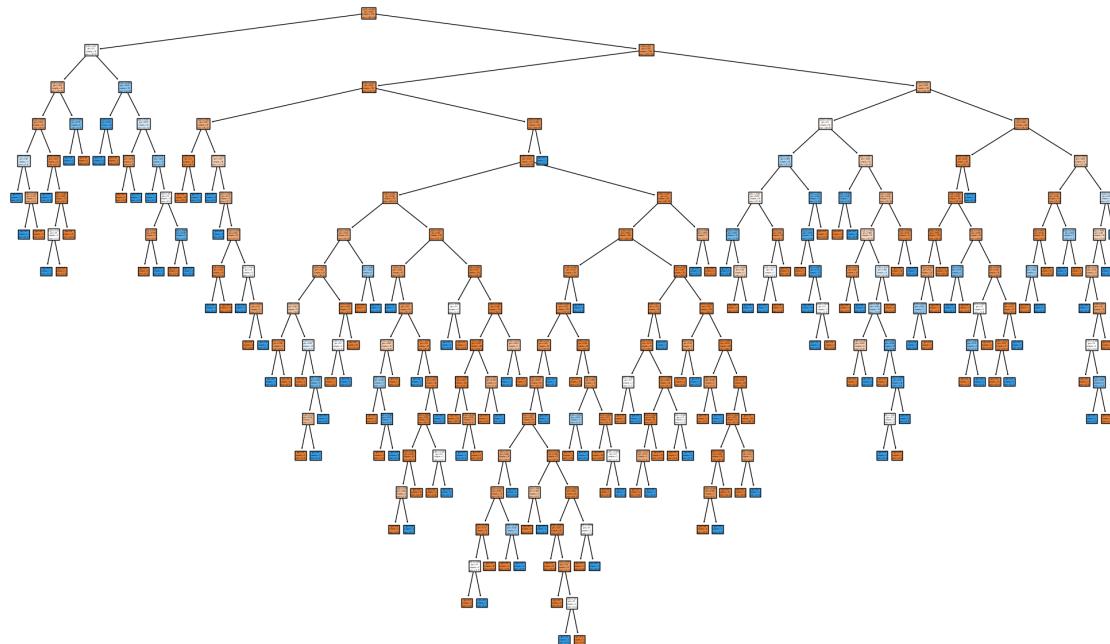
```

```

Text(0.865546218487395, 0.4722222222222222, 'x[2] <= 0.311\ngini =
0.375\nsamples = 4\nvalue = [1, 3]'),
Text(0.8588235294117647, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0']),
Text(0.8722689075630252, 0.4166666666666667, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3']),
Text(0.8789915966386554, 0.4722222222222222, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0']),
Text(0.8991596638655462, 0.5277777777777778, 'x[8] <= 0.992\ngini =
0.117\nsamples = 32\nvalue = [30, 2]'),
Text(0.892436974789916, 0.4722222222222222, 'x[28] <= 0.917\ngini =
0.062\nsamples = 31\nvalue = [30, 1]'),
Text(0.8857142857142857, 0.4166666666666667, 'gini = 0.0\nsamples = 30\nvalue =
[30, 0]),
Text(0.8991596638655462, 0.4166666666666667, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.9058823529411765, 0.4722222222222222, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.8638655462184874, 0.6944444444444444, 'gini = 0.0\nsamples = 1\nvalue =
[0, 1]),
Text(0.9630252100840336, 0.75, 'x[14] <= 0.812\ngini = 0.456\nsamples =
57\nvalue = [37, 20]'),
Text(0.9394957983193277, 0.6944444444444444, 'x[32] <= 0.4\ngini =
0.238\nsamples = 29\nvalue = [25, 4]'),
Text(0.9260504201680673, 0.6388888888888888, 'x[8] <= 0.071\ngini =
0.142\nsamples = 26\nvalue = [24, 2]'),
Text(0.9193277310924369, 0.5833333333333334, 'x[0] <= 0.321\ngini =
0.444\nsamples = 3\nvalue = [1, 2]'),
Text(0.9126050420168067, 0.5277777777777778, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]),
Text(0.9260504201680673, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]),
Text(0.9327731092436975, 0.5833333333333334, 'gini = 0.0\nsamples = 23\nvalue =
[23, 0]),
Text(0.9529411764705882, 0.6388888888888888, 'x[29] <= 0.833\ngini =
0.444\nsamples = 3\nvalue = [1, 2]),
Text(0.946218487394958, 0.5833333333333334, 'gini = 0.0\nsamples = 2\nvalue =
[0, 2]),
Text(0.9596638655462185, 0.5833333333333334, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]),
Text(0.9865546218487395, 0.6944444444444444, 'x[32] <= 0.1\ngini =
0.49\nsamples = 28\nvalue = [12, 16]'),
Text(0.9798319327731092, 0.6388888888888888, 'x[4] <= 0.804\ngini =
0.48\nsamples = 20\nvalue = [12, 8]'),
Text(0.973109243697479, 0.5833333333333334, 'x[30] <= 0.013\ngini =
0.415\nsamples = 17\nvalue = [12, 5]'),
Text(0.9663865546218487, 0.5277777777777778, 'gini = 0.0\nsamples = 2\nvalue =

```

```
[0, 2]),
Text(0.9798319327731092, 0.5277777777777778, 'x[24] <= 0.5\ngini =
0.32\nsamples = 15\nvalue = [12, 3']),
Text(0.973109243697479, 0.4722222222222222, 'x[17] <= 0.205\ngini =
0.5\nsamples = 6\nvalue = [3, 3']),
Text(0.9663865546218487, 0.4166666666666667, 'gini = 0.0\nsamples = 2\nvalue =
[2, 0]),
Text(0.9798319327731092, 0.4166666666666667, 'x[18] <= 0.706\ngini =
0.375\nsamples = 4\nvalue = [1, 3]),
Text(0.973109243697479, 0.3611111111111111, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]),
Text(0.9865546218487395, 0.3611111111111111, 'gini = 0.0\nsamples = 1\nvalue =
[1, 0]),
Text(0.9865546218487395, 0.4722222222222222, 'gini = 0.0\nsamples = 9\nvalue =
[9, 0]),
Text(0.9865546218487395, 0.5833333333333334, 'gini = 0.0\nsamples = 3\nvalue =
[0, 3]),
Text(0.9932773109243698, 0.6388888888888888, 'gini = 0.0\nsamples = 8\nvalue =
[0, 8])]
```



```
[74]: 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']

}

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

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

```

C:\Users\mb419\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\mb419\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\mb419\anaconda3\Lib\site-packages\sklearn\base.py", line 1144,  
   in wrapper  
     estimator.\_validate\_params()  
   File "C:\Users\mb419\anaconda3\Lib\site-packages\sklearn\base.py", line 637,  
   in \_validate\_params  
     validate\_parameter\_constraints()  
   File "C:\Users\mb419\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\mb419\anaconda3\Lib\site-packages\sklearn\model_selection\_search.py:976: UserWarning: One or more of the test scores are non-finite: [      nan      nan  0.83758384  0.83928597
  0.84013704  0.84013704
      nan      nan  0.83756942  0.83673278  0.83078615  0.83843491
      nan      nan  0.84352687  0.83673278  0.84437432  0.84013704
      nan      nan  0.82993148  0.84098449  0.84099892  0.83928237
      nan      nan  0.82227912  0.83078976  0.8384313   0.84185719
      nan      nan  0.84013704  0.84013704  0.84013704  0.84013704

```

```

        nan      nan 0.84436711 0.84097728 0.84098089 0.84269023
        nan      nan 0.84181392 0.84013704 0.82737829 0.84269023
        nan      nan 0.8384277 0.84015146 0.83673999 0.83756942
        nan      nan 0.83419041 0.83078615 0.84949153 0.82735305]
warnings.warn(
[76]: 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')

```

```
[77]: grid_search.best_params_
```

```
[77]: {'criterion': 'entropy',
       'max_depth': 5,
       'max_features': 'log2',
       'splitter': 'best'}
```

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

```
[78]: DecisionTreeClassifier(criterion='entropy', max_depth=3, max_features='sqrt')
```

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

```
[80]: print(classification_report(y_test,pred))
```

	precision	recall	f1-score	support
0	0.84	0.98	0.90	245
1	0.29	0.04	0.07	49
accuracy			0.82	294
macro avg	0.56	0.51	0.49	294
weighted avg	0.74	0.82	0.76	294

## 4 Random Forest

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

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

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

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

C:\Users\mb419\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\mb419\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\mb419\anaconda3\Lib\site-packages\sklearn\base.py", line 1144,
in wrapper
    estimator._validate_params()
  File "C:\Users\mb419\anaconda3\Lib\site-packages\sklearn\base.py", line 637,
in _validate_params
    validate_parameter_constraints(
  File "C:\Users\mb419\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\mb419\anaconda3\Lib\site-packages\sklearn\model\_selection\\_search.py:976: UserWarning: One or more of the test scores are non-finite: [ nan 0.8469506 0.85373026 0.85715631  
0.85458496 0.85799652  
0.85712009 0.86051717 0.86225554 0.85967695 0.86224105 0.85458496  
0.85797479 0.85714182 nan 0.84863827 0.85459945 0.85884398  
0.85458496 0.85882949 0.85883674 0.86052441 0.85797479 0.86305954  
0.86308851 0.85543966 0.85711285 0.86221932 nan 0.85034767  
0.85459945 0.85883674 0.8596842 0.8622483 0.85543966 0.86308127  
0.86138635 0.86137187 0.85967695 0.8596842 0.85967695 0.85882949

```

        nan 0.85034043 0.85204259 0.85799652 0.85714182 0.86562364
0.85798928 0.86054614 0.8596842 0.85798928 0.85797479 0.85712009
0.8596842 0.85711285      nan 0.84948573 0.85545415 0.85713458
0.85629436 0.85798204 0.86053165 0.8605389 0.85882225 0.86221932
0.86136462 0.85796031 0.85713458 0.85882225]
warnings.warn(
[84]: 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')

[85]: pred=rfc_cv.predict(x_test)

[86]: print(classification_report(y_test,pred))

          precision    recall  f1-score   support

           0       0.85      0.98      0.91      245
           1       0.64      0.14      0.23       49

    accuracy                           0.84      294
   macro avg       0.74      0.56      0.57      294
weighted avg       0.82      0.84      0.80      294

[87]: rfc_cv.best_params_

[87]: {'max_depth': 13, 'max_features': 5}

[88]: confusion_matrix(y_test,pred)

[88]: array([[241,    4],
             [ 42,    7]], dtype=int64)

[89]: pd.crosstab(y_test,pred)

[89]: col_0      0  1
Attrition
0           241  4
1            42  7

[90]: a3 = (240+11)/294 #accuracy
a3

[90]: 0.8537414965986394

```

```
[91]: r3 = 11/(11+38) # recall  
r3
```

[91]: 0.22448979591836735

```
[92]: p3 = 11/(5+11) #precision  
p3
```

[92]: 0.6875

```
[93]: f3 = 2*r3*p3/(r3+p3) #F1 score  
f3
```

[93]: 0.3384615384615384

```
[94]: print(classification_report(y_test,pred))
```

	precision	recall	f1-score	support
0	0.85	0.98	0.91	245
1	0.64	0.14	0.23	49
accuracy			0.84	294
macro avg	0.74	0.56	0.57	294
weighted avg	0.82	0.84	0.80	294