

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

```
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

EmployeeNumber	DistanceFromHome	Education	EducationField	EmployeeCount
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.shape
```

```
(1470, 35)
```

```
df.BusinessTravel.value_counts()
```

```

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

```

```
df.Attrition.value_counts()
```

```

No      1233
Yes      237
Name: Attrition, dtype: int64

```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 1470 entries, 0 to 1469
```

```
Data columns (total 35 columns):
```

#	Column	Non-Null Count	Dtype
0	Age	1470 non-null	int64
1	Attrition	1470 non-null	object
2	BusinessTravel	1470 non-null	object
3	DailyRate	1470 non-null	int64
4	Department	1470 non-null	object
5	DistanceFromHome	1470 non-null	int64
6	Education	1470 non-null	int64
7	EducationField	1470 non-null	object
8	EmployeeCount	1470 non-null	int64
9	EmployeeNumber	1470 non-null	int64
10	EnvironmentSatisfaction	1470 non-null	int64
11	Gender	1470 non-null	object
12	HourlyRate	1470 non-null	int64
13	JobInvolvement	1470 non-null	int64

14	JobLevel	1470	non-null	int64
15	JobRole	1470	non-null	object
16	JobSatisfaction	1470	non-null	int64
17	MaritalStatus	1470	non-null	object
18	MonthlyIncome	1470	non-null	int64
19	MonthlyRate	1470	non-null	int64
20	NumCompaniesWorked	1470	non-null	int64
21	Over18	1470	non-null	object
22	OverTime	1470	non-null	object
23	PercentSalaryHike	1470	non-null	int64
24	PerformanceRating	1470	non-null	int64
25	RelationshipSatisfaction	1470	non-null	int64
26	StandardHours	1470	non-null	int64
27	StockOptionLevel	1470	non-null	int64
28	TotalWorkingYears	1470	non-null	int64
29	TrainingTimesLastYear	1470	non-null	int64
30	WorkLifeBalance	1470	non-null	int64
31	YearsAtCompany	1470	non-null	int64
32	YearsInCurrentRole	1470	non-null	int64
33	YearsSinceLastPromotion	1470	non-null	int64
34	YearsWithCurrManager	1470	non-null	int64

dtypes: int64(26), object(9)

memory usage: 402.1+ KB

df.describe()

	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.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
```

```
sns.distplot(df["Age"])
```

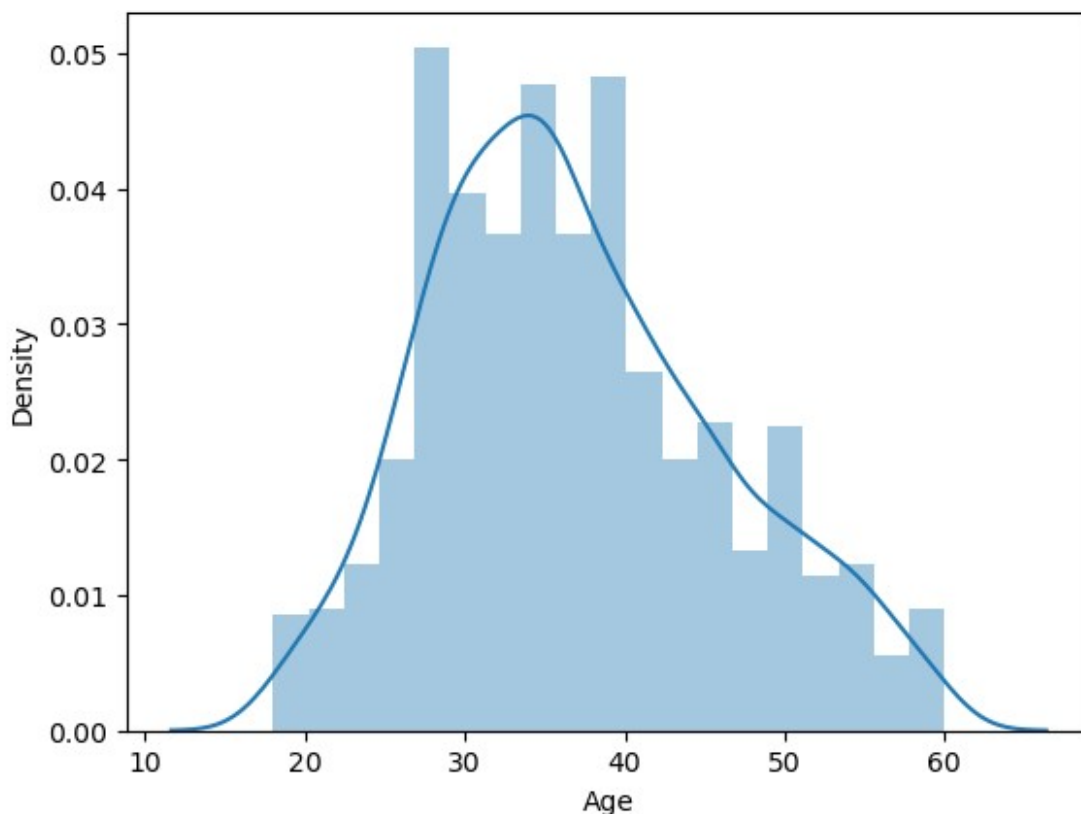
```
C:\Users\himaj\AppData\Local\Temp\ipykernel_58416\2732350774.py:1:
UserWarning:
```

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

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

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

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



```
df.corr()
```

C:\Users\himaj\AppData\Local\Temp\ipykernel\_58416\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		

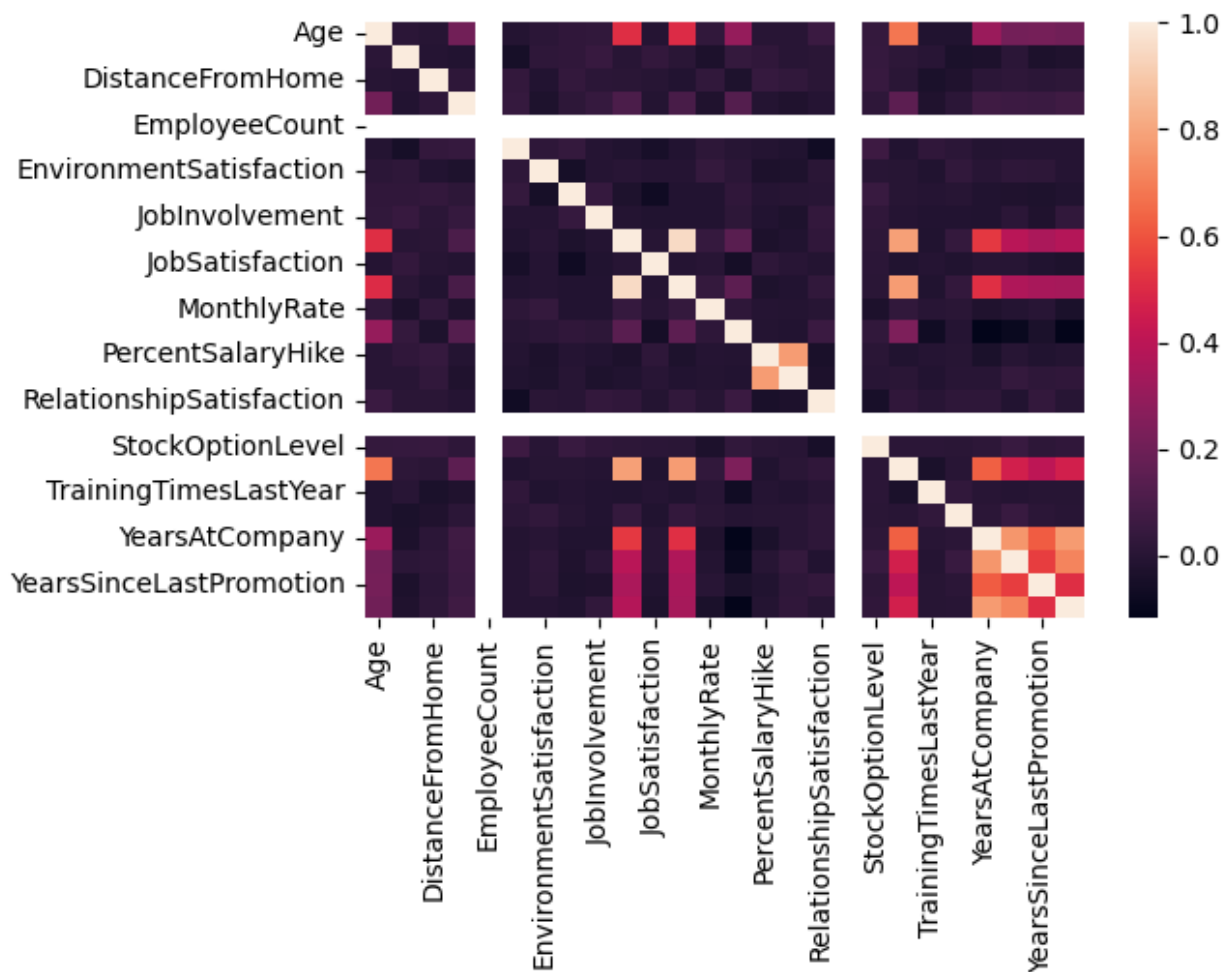
[26 rows x 26 columns]

```
plt.subplots(figsize=(6,4))
sns.heatmap(df.corr())
```

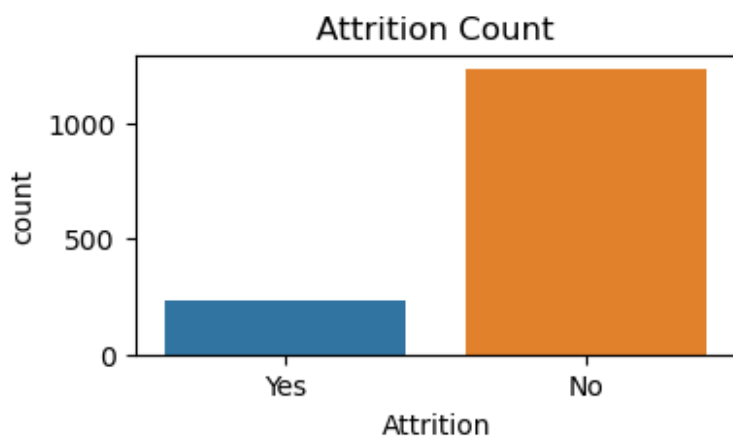
C:\Users\himaj\AppData\Local\Temp\ipykernel\_58416\3989471373.py:2:  
FutureWarning: The default value of numeric\_only in DataFrame.corr is  
deprecated. In a future version, it will default to False. Select only  
valid columns or specify the value of numeric\_only to silence this  
warning.

```
sns.heatmap(df.corr())
```

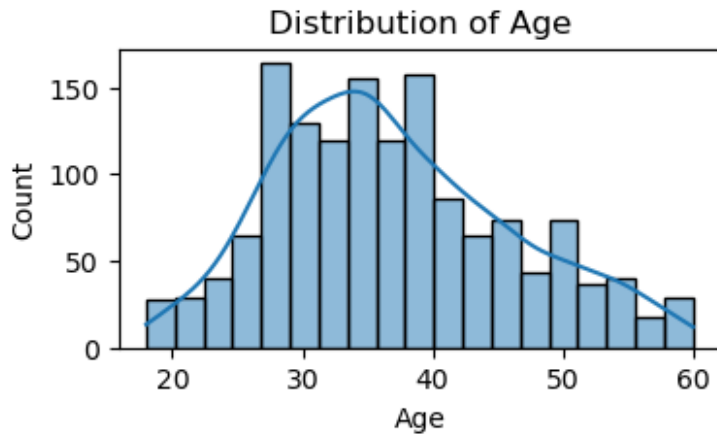
<Axes: >



```
plt.figure(figsize=(4,2))
sns.countplot(x="Attrition", data=df)
plt.title("Attrition Count")
plt.show()
```

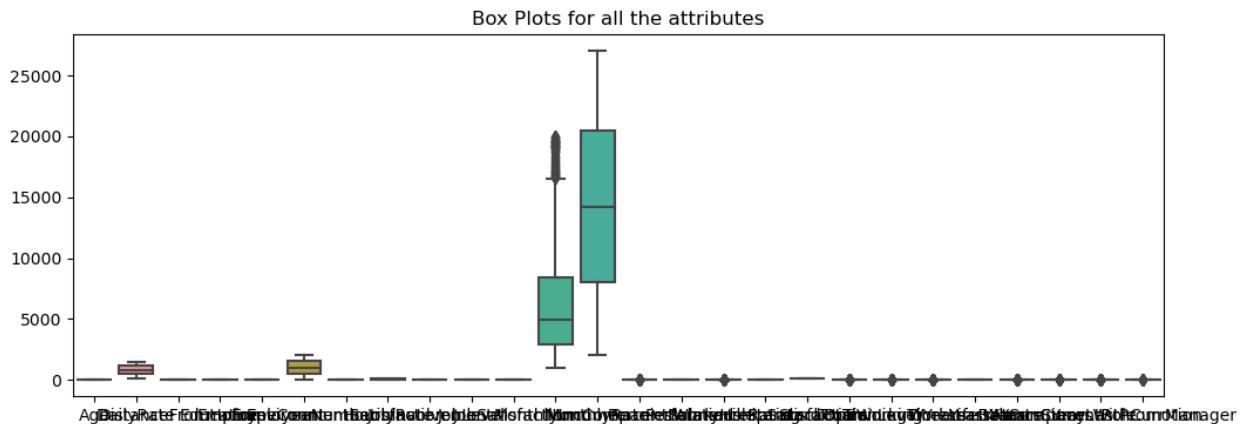


```
plt.figure(figsize=(4,2))
sns.histplot(data=df, x="Age", kde=True)
plt.title("Distribution of Age")
plt.show()
```



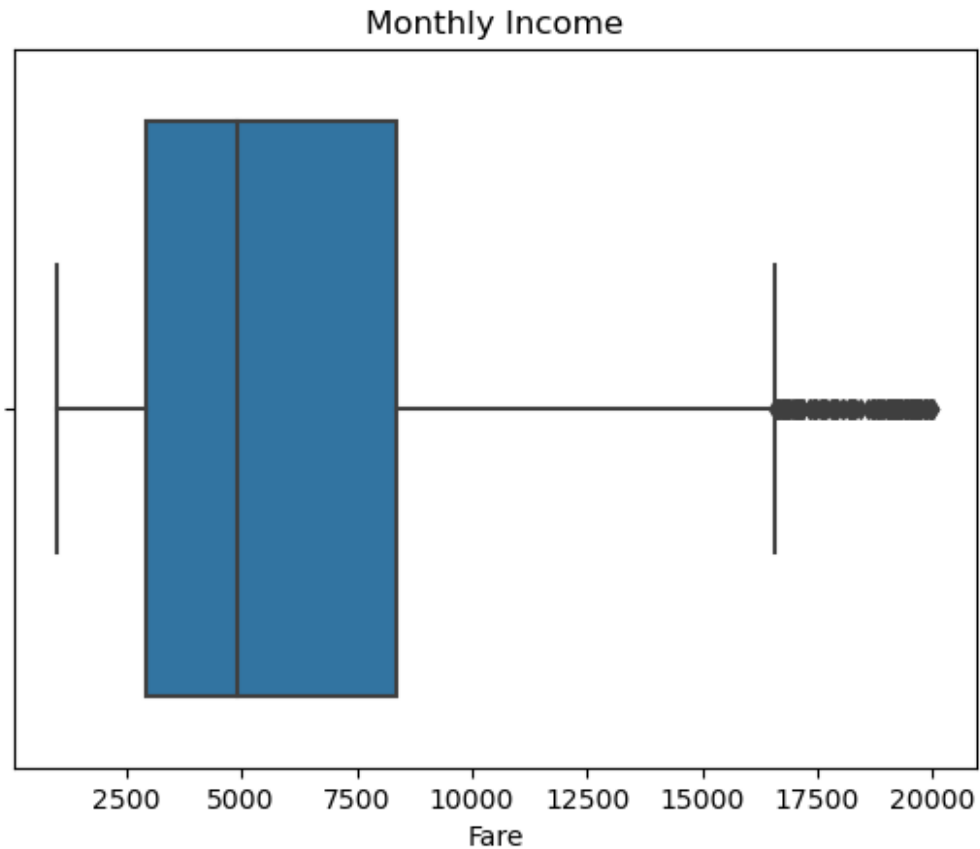
# Outlier detection

```
plt.figure(figsize=(12,4))
sns.boxplot(data=df)
plt.title('Box Plots for all the attributes')
plt.show()
```



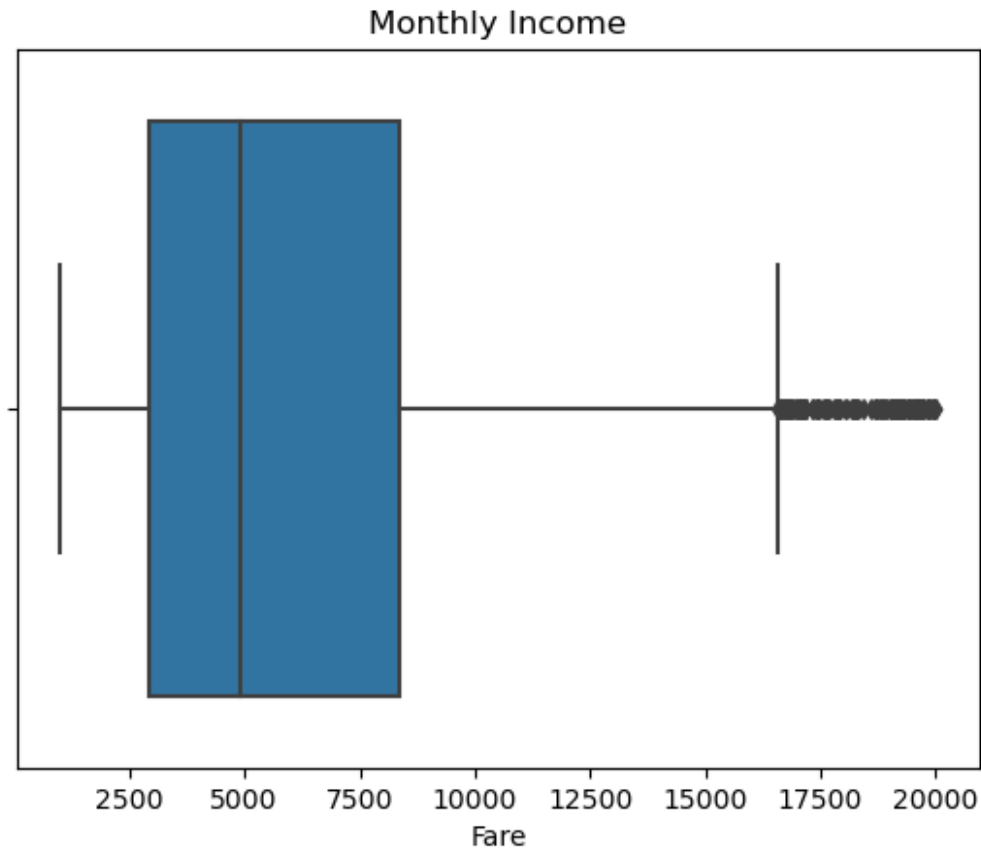
```
sns.boxplot(data=df, x='MonthlyIncome')
plt.title('Monthly Income')
plt.xlabel('Fare')
plt.show()
```





```
from scipy import stats
z_scores = stats.zscore(df['MonthlyIncome'])
z_score_threshold = 3
df_cleaned = df[(np.abs(z_scores) <= z_score_threshold)]

sns.boxplot(data=df_cleaned, x='MonthlyIncome')
plt.title('Monthly Income')
plt.xlabel('Fare')
plt.show()
```



```
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

EmployeeNumber	DistanceFromHome	Education	EducationField	EmployeeCount
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]

```

## Splitting dependent and independent variables

```

x=df.iloc[:,2:]
x.head()

```

```

BusinessTravel DailyRate Department
DistanceFromHome \
0 Travel_Rarely 1102 Sales
1
1 Travel_Frequently 279 Research & Development
8
2 Travel_Rarely 1373 Research & Development
2
3 Travel_Frequently 1392 Research & Development
3

```

```
4      Travel_Rarely      591  Research & Development
2
```

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

```
      EnvironmentSatisfaction Gender ... RelationshipSatisfaction \
0                          2  Female ...              1
1                          3   Male ...              4
2                          4   Male ...              2
3                          4  Female ...              3
4                          1   Male ...              4
```

```
      StandardHours StockOptionLevel TotalWorkingYears
TrainingTimesLastYear \
0          80              0              8
0
1          80              1             10
3
2          80              0              7
3
3          80              0              8
3
4          80              1              6
3
```

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

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

```
[5 rows x 33 columns]
```

```
x.head()
```

```
      BusinessTravel DailyRate      Department
DistanceFromHome \
```

0	Travel_Rarely	1102	Sales
1			
1	Travel_Frequently	279	Research & Development
8			
2	Travel_Rarely	1373	Research & Development
2			
3	Travel_Frequently	1392	Research & Development
3			
4	Travel_Rarely	591	Research & Development
2			

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

	EnvironmentSatisfaction	Gender	...	RelationshipSatisfaction	\
0		2	Female	...	1
1		3	Male	...	4
2		4	Male	...	2
3		4	Female	...	3
4		1	Male	...	4

	StandardHours	StockOptionLevel	TotalWorkingYears
TrainingTimesLastYear	\		
0	80	0	8
0			
1	80	1	10
3			
2	80	0	7
3			
3	80	0	8
3			
4	80	1	6
3			

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

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

```

4                                2                                2

[5 rows x 33 columns]
y = df['Attrition']
y.head()
0    Yes
1    No
2    Yes
3    No
4    No
Name: Attrition, dtype: object

x.shape
(1470, 33)
y.shape
(1470,)
type(x)
pandas.core.frame.DataFrame
type(y)
pandas.core.series.Series

```

## Label encoding

```

from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
x.Gender=le.fit_transform(x.Gender)
x.BusinessTravel=le.fit_transform(x.BusinessTravel)
x.Department=le.fit_transform(x.Department)
x.EducationField=le.fit_transform(x.EducationField)
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)
x.head()

```

	BusinessTravel	DailyRate	Department	DistanceFromHome	Education
0	2	1102	2	1	2
1	1	279	1	8	1

2	2	1373	1	2	2
3	1	1392	1	3	4
4	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			

	Gender	...	RelationshipSatisfaction	StandardHours
StockOptionLevel \				
0	0	...	1	80
0				
1	1	...	4	80
1				
2	1	...	2	80
0				
3	0	...	3	80
0				
4	1	...	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 33 columns]

## Feature Scaling

```
from sklearn.preprocessing import MinMaxScaler

numericalcolumns =
df.select_dtypes(include=['int64','float64']).columns

numericalcolumns

Index(['Age', 'DailyRate', 'DistanceFromHome', 'Education',
      'EmployeeCount',
      'EmployeeNumber', 'EnvironmentSatisfaction', 'HourlyRate',
      'JobInvolvement', 'JobLevel', 'JobSatisfaction',
      'MonthlyIncome',
      'MonthlyRate', 'NumCompaniesWorked', 'PercentSalaryHike',
      'PerformanceRating', 'RelationshipSatisfaction',
      'StandardHours',
      'StockOptionLevel', 'TotalWorkingYears',
      'TrainingTimesLastYear',
      'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole',
      'YearsSinceLastPromotion', 'YearsWithCurrManager'],
      dtype='object')
```

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

x\_scaled

	BusinessTravel	DailyRate	Department	DistanceFromHome
Education \				
0	1.0	0.715820	1.0	0.000000
0.25				
1	0.5	0.126700	0.5	0.250000
0.00				
2	1.0	0.909807	0.5	0.035714
0.25				
3	0.5	0.923407	0.5	0.071429
0.75				
4	1.0	0.350036	0.5	0.035714
0.00				
...	...	...	...	...
..				.



1465	0.5	0.559771	0.5	0.785714
0.25				
1466	1.0	0.365784	0.5	0.178571
0.00				
1467	1.0	0.037938	0.5	0.107143
0.50				
1468	0.5	0.659270	1.0	0.035714
0.50				
1469	1.0	0.376521	0.5	0.250000
0.50				

EducationField	EmployeeCount	EmployeeNumber
EnvironmentSatisfaction \		
0	0.2	0.0
0.333333		0.000000
1	0.2	0.0
0.666667		0.000484
2	0.8	0.0
1.000000		0.001451
3	0.2	0.0
1.000000		0.001935
4	0.6	0.0
0.000000		0.002903
...	...	...
...		...
1465	0.6	0.0
0.666667		0.996613
1466	0.6	0.0
1.000000		0.997097
1467	0.2	0.0
0.333333		0.998065
1468	0.6	0.0
1.000000		0.998549
1469	0.6	0.0
0.333333		1.000000

Gender	...	RelationshipSatisfaction	StandardHours
StockOptionLevel \			
0	0.0	...	0.000000
0.000000			0.0
1	1.0	...	1.000000
0.333333			0.0
2	1.0	...	0.333333
0.000000			0.0
3	0.0	...	0.666667
0.000000			0.0
4	1.0	...	1.000000
0.333333			0.0
...	...	...	...

...				
1465	1.0	...	0.666667	0.0
0.333333				
1466	1.0	...	0.000000	0.0
0.333333				
1467	1.0	...	0.333333	0.0
0.333333				
1468	1.0	...	1.000000	0.0
0.000000				
1469	1.0	...	0.000000	0.0
0.000000				

	TotalWorkingYears	TrainingTimesLastYear	WorkLifeBalance	\
0	0.200	0.000000	0.000000	
1	0.250	0.500000	0.666667	
2	0.175	0.500000	0.666667	
3	0.200	0.500000	0.666667	
4	0.150	0.500000	0.666667	
...	...	...	...	
1465	0.425	0.500000	0.666667	
1466	0.225	0.833333	0.666667	
1467	0.150	0.000000	0.666667	
1468	0.425	0.500000	0.333333	
1469	0.150	0.500000	1.000000	

	YearsAtCompany	YearsInCurrentRole	YearsSinceLastPromotion	\
0	0.150	0.222222	0.000000	
1	0.250	0.388889	0.066667	
2	0.000	0.000000	0.000000	
3	0.200	0.388889	0.200000	
4	0.050	0.111111	0.133333	
...	...	...	...	
1465	0.125	0.111111	0.000000	
1466	0.175	0.388889	0.066667	
1467	0.150	0.111111	0.000000	
1468	0.225	0.333333	0.000000	
1469	0.100	0.166667	0.066667	

	YearsWithCurrManager
0	0.294118
1	0.411765
2	0.000000
3	0.000000
4	0.117647
...	...
1465	0.176471
1466	0.411765
1467	0.176471
1468	0.470588
1469	0.117647

```
[1470 rows x 33 columns]
```

## 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, random_state=40)
```

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

```
x_train.head()
```

	BusinessTravel	DailyRate	Department	DistanceFromHome
Education \				
442	0	635	2	10
4				
388	2	201	1	1
4				
35	2	1273	1	2
2				
254	2	1247	2	20
2				
617	2	625	1	4
3				

	EducationField	EmployeeCount	EmployeeNumber
EnvironmentSatisfaction \			
442	3	1	592
2			
388	1	1	517
2			
35	3	1	46
4			
254	2	1	349
4			
617	3	1	852
4			

	Gender	...	RelationshipSatisfaction	StandardHours
StockOptionLevel \				
442	1	...	4	80
0				
388	0	...	2	80

```

1
35      0  ...      4      80
2
254     1  ...      4      80
1
617     1  ...      4      80
0

      TotalWorkingYears  TrainingTimesLastYear  WorkLifeBalance \
442                   10                      3                2
388                    8                      5                3
35                     6                      3                2
254                   10                      2                3
617                   10                      2                2

      YearsAtCompany  YearsInCurrentRole  YearsSinceLastPromotion \
442                 10                   3                      9
388                  5                   2                      1
35                   5                   3                      1
254                  3                   2                      0
617                  5                   2                      2

      YearsWithCurrManager
442                        7
388                        2
35                         4
254                        2
617                        3

[5 rows x 33 columns]

```

## Model Building

Model Building using Logistic Regression

```

from sklearn.linear_model import LogisticRegression

model = LogisticRegression()

model.fit(x_train, y_train)

C:\Users\himaj\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:

```



[illegible]

y\_test

1456	No
236	Yes
70	No
42	Yes
454	No

	...
1392	No
627	No
1407	No
731	Yes
722	No

Name: Attrition, Length: 294, dtype: object

```
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

EmployeeNumber	DistanceFromHome	Education	EducationField	EmployeeCount
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]

```
model.predict(ms.fit_transform(x))
```

C:\Users\himaj\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(['Yes', 'No', 'Yes', ..., 'No', 'No', 'No'], dtype=object)
```

Evaluating of classification model

```

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

accuracy_score(y_test, y_pred)

0.8333333333333334

confusion_matrix(y_test, y_pred)

array([[245,  0],
       [ 49,  0]], dtype=int64)

pd.crosstab(y_test, y_pred)

col_0      No
Attrition
No          245
Yes          49

print(classification_report(y_test, y_pred))

```

	precision	recall	f1-score	support
No	0.83	1.00	0.91	245
Yes	0.00	0.00	0.00	49
accuracy			0.83	294
macro avg	0.42	0.50	0.45	294
weighted avg	0.69	0.83	0.76	294

```

C:\Users\himaj\anaconda3\Lib\site-packages\sklearn\metrics\
_classification.py:1469: UndefinedMetricWarning: Precision and F-score
are ill-defined and being set to 0.0 in labels with no predicted
samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\himaj\anaconda3\Lib\site-packages\sklearn\metrics\
_classification.py:1469: UndefinedMetricWarning: Precision and F-score
are ill-defined and being set to 0.0 in labels with no predicted
samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\himaj\anaconda3\Lib\site-packages\sklearn\metrics\
_classification.py:1469: UndefinedMetricWarning: Precision and F-score
are ill-defined and being set to 0.0 in labels with no predicted
samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))

```

Roc-AUC Curve

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



y\_pred\_prob

```
array([0.1269117 , 0.1367298 , 0.08856111, 0.42399048, 0.39142484,
       0.04436196, 0.12890411, 0.20344523, 0.26170413, 0.23426568,
       0.3287974 , 0.10470663, 0.05334631, 0.09040092, 0.33922967,
       0.15805223, 0.10705361, 0.19171757, 0.25904029, 0.34761996,
       0.17120528, 0.01543825, 0.15259077, 0.12472736, 0.08046627,
       0.23232198, 0.20895101, 0.14486701, 0.2463008 , 0.26465362,
       0.02642624, 0.03956622, 0.15630196, 0.14229448, 0.30933548,
       0.3412925 , 0.13341011, 0.21786506, 0.09159417, 0.03927247,
       0.29752254, 0.0954887 , 0.12847056, 0.04198665, 0.12300917,
       0.26118401, 0.16829683, 0.11495276, 0.13149545, 0.12057169,
       0.09205675, 0.21574212, 0.15050669, 0.17794209, 0.13905351,
       0.16661292, 0.06797552, 0.27443612, 0.04793437, 0.14967738,
       0.19977231, 0.08934343, 0.11789569, 0.12111361, 0.08614477,
       0.09389582, 0.27950817, 0.03460094, 0.07547985, 0.13657322,
       0.0445866 , 0.27795223, 0.05438318, 0.14335678, 0.33013084,
       0.27014799, 0.03808675, 0.02555547, 0.19124183, 0.03189124,
       0.18555733, 0.21903053, 0.10769021, 0.12598711, 0.12377976,
       0.27746651, 0.25756312, 0.16316891, 0.02937981, 0.27617148,
       0.28367978, 0.2024671 , 0.0202049 , 0.0670186 , 0.23514057,
       0.18396591, 0.26705957, 0.05884434, 0.12618696, 0.11580528,
       0.246875 , 0.03968388, 0.11627215, 0.05997782, 0.14761594,
       0.13073625, 0.18400454, 0.25986668, 0.01603003, 0.15814518,
       0.3465885 , 0.13916891, 0.08361326, 0.10993309, 0.14776334,
       0.04212779, 0.04821943, 0.15928801, 0.33022753, 0.15746768,
       0.28897042, 0.01012801, 0.04668629, 0.19671219, 0.07940416,
       0.27628363, 0.08804126, 0.04839881, 0.19751203, 0.10361103,
       0.19349545, 0.15067797, 0.28900828, 0.06360858, 0.19156746,
       0.2087438 , 0.03516907, 0.28254939, 0.05298114, 0.05355734,
       0.13722247, 0.38777911, 0.15183027, 0.13810171, 0.25078261,
       0.01759044, 0.13648495, 0.14213917, 0.35005366, 0.06179861,
       0.10994663, 0.10901876, 0.04099997, 0.12791252, 0.39506975,
       0.07220022, 0.14272372, 0.09277771, 0.2750967 , 0.17065144,
       0.01622879, 0.33478478, 0.17624919, 0.10471147, 0.13344414,
       0.04714757, 0.38408563, 0.0815325 , 0.24275551, 0.1256278 ,
       0.3341296 , 0.042477 , 0.47427039, 0.34150924, 0.22971191,
       0.05419324, 0.34753921, 0.18454739, 0.03520548, 0.00564107,
       0.06236659, 0.09420622, 0.05009921, 0.19895088, 0.03456291,
       0.22519921, 0.04292082, 0.01971212, 0.4295647 , 0.07563821,
       0.08155498, 0.10841429, 0.00923005, 0.04202215, 0.18118998,
       0.13857939, 0.12256346, 0.15251999, 0.23385248, 0.03531807,
       0.16640489, 0.08431189, 0.06664876, 0.12999292, 0.0663151 ,
       0.10769755, 0.23775319, 0.21236001, 0.15441227, 0.11001043,
       0.00829649, 0.27174263, 0.0767256 , 0.24139314, 0.01362718,
       0.20068367, 0.07881528, 0.21246527, 0.11028945, 0.07313669,
       0.40089405, 0.28323103, 0.03105265, 0.08681372, 0.04575705,
       0.13197757, 0.02015167, 0.07405819, 0.1275278 , 0.09579005,
       0.01166289, 0.11290404, 0.10709683, 0.08578317, 0.10123403,
       0.09493532, 0.22075754, 0.10020879, 0.10574884, 0.18520986,
```

```

0.25331549, 0.15996964, 0.23229787, 0.19468418, 0.10232503,
0.30347488, 0.0237353 , 0.47803552, 0.05803445, 0.12734958,
0.14421286, 0.20674013, 0.16470074, 0.18309886, 0.09793091,
0.12897375, 0.06815628, 0.04284429, 0.3856836 , 0.23627879,
0.23145463, 0.39234388, 0.07047972, 0.20224778, 0.27596604,
0.16435912, 0.27154998, 0.20101782, 0.28828216, 0.22989958,
0.13776628, 0.19709854, 0.22745295, 0.23436059, 0.08639474,
0.02796938, 0.25629211, 0.08963719, 0.25773142, 0.06254773,
0.07209341, 0.26313086, 0.06705019, 0.02379169, 0.06525368,
0.17444298, 0.16154772, 0.10449492, 0.10457458, 0.08549116,
0.06982678, 0.17576048, 0.22292339, 0.20697072])

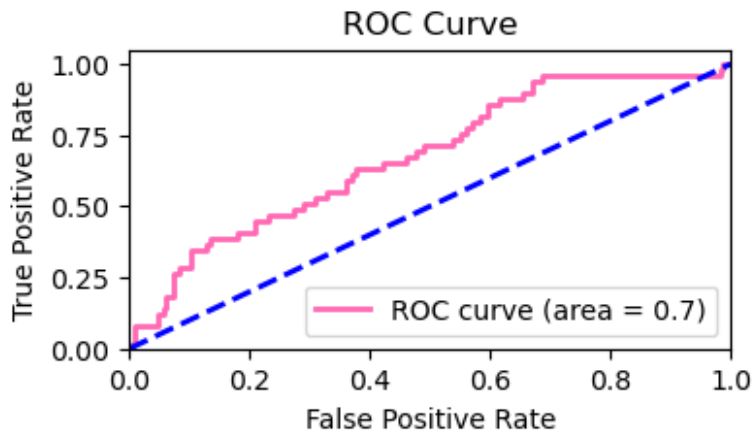
```

```

fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob, pos_label="Yes")
# Specify pos_label as 1
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(4, 2))
plt.plot(fpr, tpr, color='hotpink', lw=2, label=f'ROC curve (area =
{roc_auc:.1f})')
plt.plot([0, 1], [0, 1], color='blue', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend(loc='lower right')
plt.show()

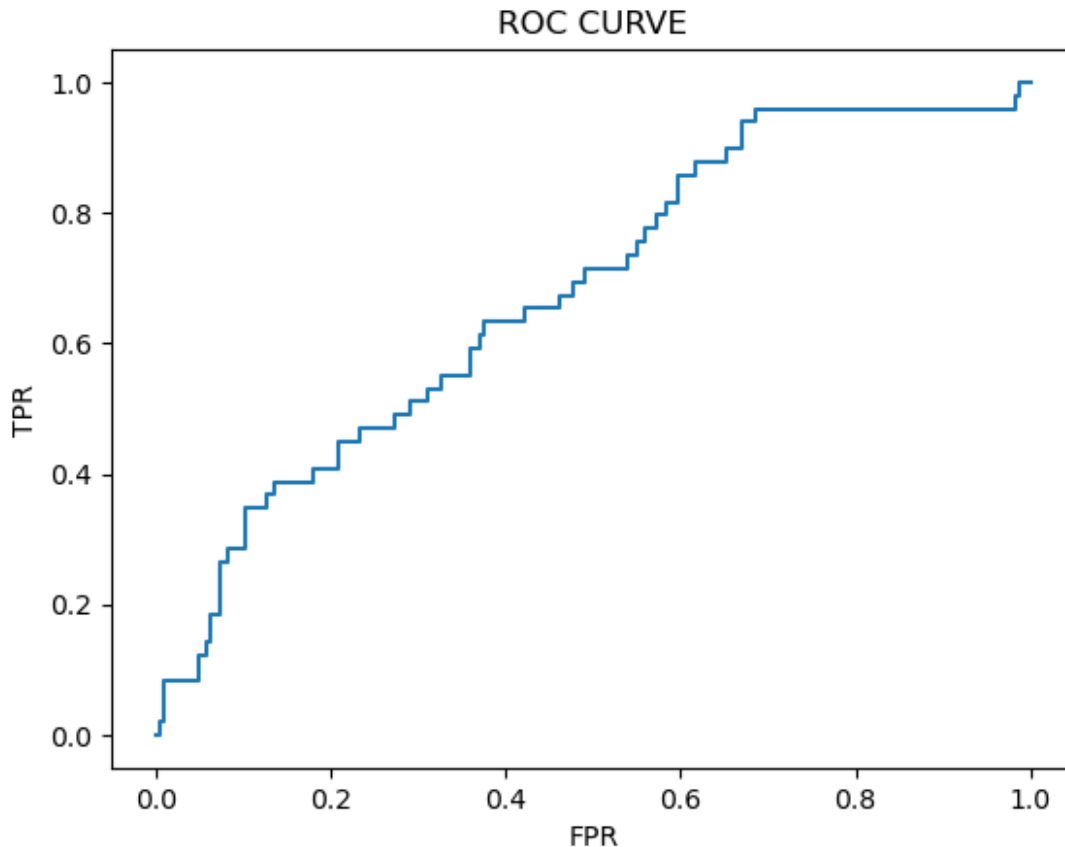
```



```

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

```



## Model Building using Decision Tree

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

dtc.fit(x_train,y_train)

DecisionTreeClassifier()

pred=dtc.predict(x_test)

pred

array(['Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No',
       'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'Yes', 'Yes', 'No', 'No', 'Yes', 'No', 'No', 'Yes',
       'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No',
```

```

'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No',
'No',
'No', 'Yes', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No',
'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
'No',
'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No',
'No',
'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
'No',
'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes',
'No',
'Yes', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
'No',
'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
'No',
'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No',
'No',
'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes',
'No',
'No', 'No', 'Yes', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No',
'No',
'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No',
'No',
'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
'No',
'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes',
'No',
'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'Yes', 'No', 'No',
'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'Yes',
'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
'No',
'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
'No',
'No', 'No', 'No', 'Yes', 'No', 'No', 'Yes', 'No', 'Yes', 'No',
'Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes',
'No',
'No', 'No', 'Yes', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No',
'No',
'No', 'Yes', 'No'], dtype=object)

```

y\_test

```

1456    No
236     Yes
70      No
42     Yes
454    No
...
1392    No
627    No

```

```
1407      No
731      Yes
722      No
Name: Attrition, Length: 294, dtype: object
```

```
model.predict(ms.fit_transform(x))
```

```
C:\Users\himaj\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(['Yes', 'No', 'Yes', ..., 'No', 'No', 'No'], dtype=object)
```

```
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]

## Evaluating of classification model

```
from sklearn.metrics import
accuracy_score, confusion_matrix, classification_report, roc_auc_score, ro
c_curve
```

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

```
0.7925170068027211
```

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

```
array([[217,  28],
       [ 33,  16]], dtype=int64)
```

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

col_0	No	Yes
Attrition		
No	217	28
Yes	33	16

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

	precision	recall	f1-score	support
No	0.87	0.89	0.88	245
Yes	0.36	0.33	0.34	49
accuracy			0.79	294

macro avg	0.62	0.61	0.61	294
weighted avg	0.78	0.79	0.79	294

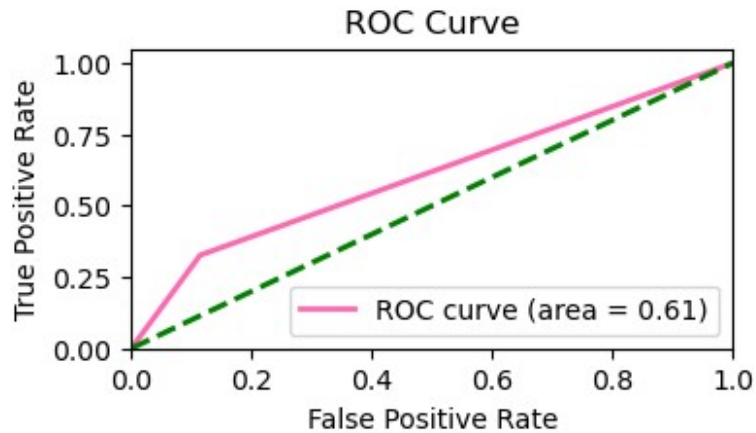
```
y_pred_probs = dtc.predict_proba(x_test)[: , 1]
y_pred_probs
```

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

```
fpr, tpr, thresholds = roc_curve(y_test, y_pred_probs, pos_label="Yes")
roc_auc = auc(fpr, tpr)

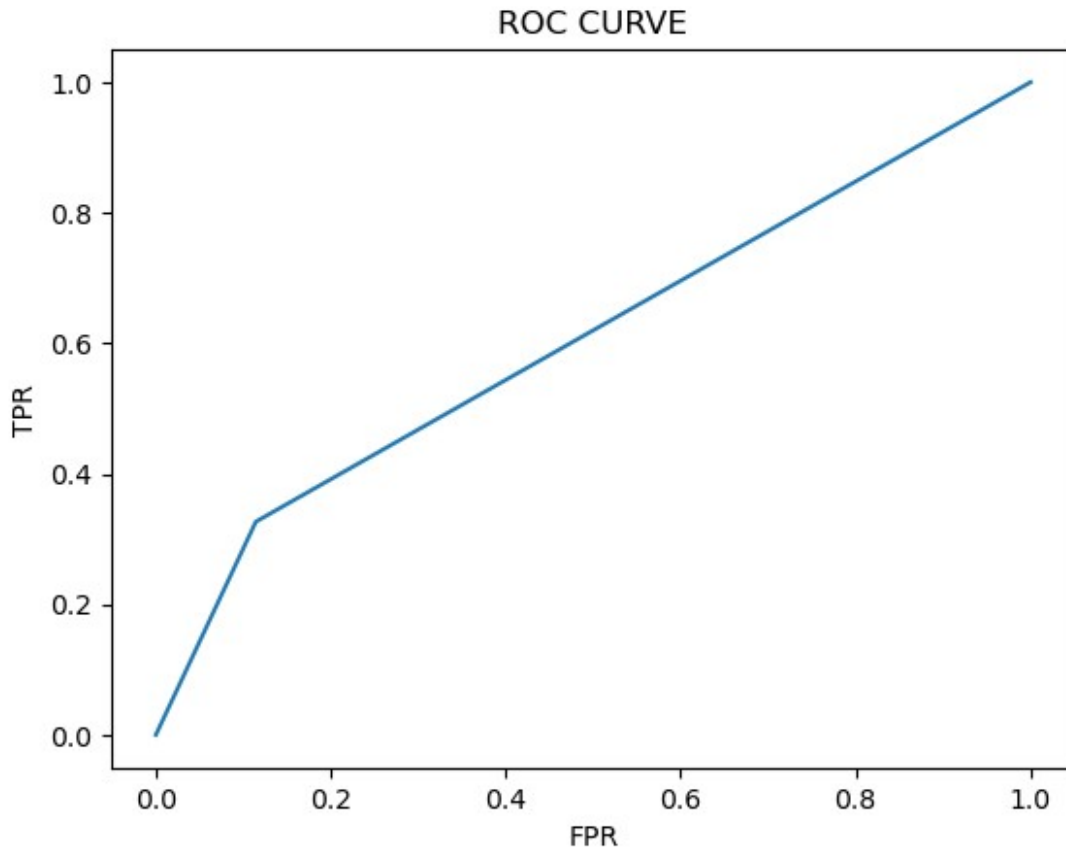
plt.figure(figsize=(4, 2))
plt.plot(fpr, tpr, color='hotpink', lw=2, label=f'ROC curve (area =
{roc_auc:.2f})')
```

```
plt.plot([0, 1], [0, 1], color='green', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend(loc='lower right')
plt.show()
```



```
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(0.3271022727272727, 0.9705882352941176, 'x[26] <= 1.5\ngini =
0.269\nsamples = 1176\nvalue = [988, 188]'),
Text(0.09772727272727273, 0.9117647058823529, 'x[3] <= 17.5\ngini =
0.5\nsamples = 72\nvalue = [36, 36]'),
Text(0.057272727272727274, 0.8529411764705882, 'x[10] <= 55.5\ngini =
0.481\nsamples = 57\nvalue = [34, 23]'),
Text(0.014545454545454545, 0.7941176470588235, 'x[0] <= 0.5\ngini =
0.457\nsamples = 17\nvalue = [6, 11]'),
Text(0.007272727272727273, 0.7352941176470589, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
Text(0.02181818181818182, 0.7352941176470589, 'x[11] <= 2.5\ngini =
0.337\nsamples = 14\nvalue = [3, 11]'),
Text(0.014545454545454545, 0.6764705882352942, 'gini = 0.0\nsamples =
8\nvalue = [0, 8]'),
Text(0.02909090909090909, 0.6764705882352942, 'x[31] <= 0.5\ngini =
0.5\nsamples = 6\nvalue = [3, 3]'),
Text(0.02181818181818182, 0.6176470588235294, 'gini = 0.0\nsamples =
```

```
3\nvalue = [3, 0]'),
Text(0.03636363636363636, 0.6176470588235294, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
Text(0.1, 0.7941176470588235, 'x[20] <= 0.5\ngini = 0.42\nsamples =
40\nvalue = [28, 12]'),
Text(0.07636363636363637, 0.7352941176470589, 'x[1] <= 249.5\ngini =
0.32\nsamples = 30\nvalue = [24, 6]'),
Text(0.05818181818181818, 0.6764705882352942, 'x[21] <= 16.5\ngini =
0.444\nsamples = 3\nvalue = [1, 2]'),
Text(0.05090909090909091, 0.6176470588235294, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
Text(0.06545454545454546, 0.6176470588235294, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
Text(0.09454545454545454, 0.6764705882352942, 'x[28] <= 3.5\ngini =
0.252\nsamples = 27\nvalue = [23, 4]'),
Text(0.08, 0.6176470588235294, 'x[7] <= 1646.5\ngini = 0.153\nsamples
= 24\nvalue = [22, 2]'),
Text(0.07272727272727272, 0.5588235294117647, 'gini = 0.0\nsamples =
20\nvalue = [20, 0]'),
Text(0.08727272727272728, 0.5588235294117647, 'x[16] <= 1700.5\ngini
= 0.5\nsamples = 4\nvalue = [2, 2]'),
Text(0.08, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.09454545454545454, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [2,
0]'),
Text(0.10909090909090909, 0.6176470588235294, 'x[14] <= 2.5\ngini =
0.444\nsamples = 3\nvalue = [1, 2]'),
Text(0.10181818181818182, 0.5588235294117647, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
Text(0.11636363636363636, 0.5588235294117647, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
Text(0.12363636363636364, 0.7352941176470589, 'x[25] <= 0.5\ngini =
0.48\nsamples = 10\nvalue = [4, 6]'),
Text(0.11636363636363636, 0.6764705882352942, 'gini = 0.0\nsamples =
4\nvalue = [0, 4]'),
Text(0.13090909090909092, 0.6764705882352942, 'x[14] <= 2.0\ngini =
0.444\nsamples = 6\nvalue = [4, 2]'),
Text(0.12363636363636364, 0.6176470588235294, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
Text(0.13818181818181818, 0.6176470588235294, 'gini = 0.0\nsamples =
4\nvalue = [4, 0]'),
Text(0.13818181818181818, 0.8529411764705882, 'x[7] <= 89.0\ngini =
0.231\nsamples = 15\nvalue = [2, 13]'),
Text(0.13090909090909092, 0.7941176470588235, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
Text(0.14545454545454545, 0.7941176470588235, 'x[16] <= 2928.5\ngini
= 0.133\nsamples = 14\nvalue = [1, 13]'),
Text(0.13818181818181818, 0.7352941176470589, 'gini = 0.0\nsamples =
13\nvalue = [0, 13]'),
Text(0.15272727272727274, 0.7352941176470589, 'gini = 0.0\nsamples =
```

```
1\nvalue = [1, 0]'),
Text(0.5564772727272728, 0.9117647058823529, 'x[20] <= 0.5\ngini =
0.237\nsamples = 1104\nvalue = [952, 152]'),
Text(0.3390909090909091, 0.8529411764705882, 'x[28] <= 1.5\ngini =
0.163\nsamples = 795\nvalue = [724, 71]'),
Text(0.18181818181818182, 0.7941176470588235, 'x[31] <= 6.0\ngini =
0.375\nsamples = 44\nvalue = [33, 11]'),
Text(0.17454545454545456, 0.7352941176470589, 'x[32] <= 0.5\ngini =
0.289\nsamples = 40\nvalue = [33, 7]'),
Text(0.16, 0.6764705882352942, 'x[26] <= 4.5\ngini = 0.278\nsamples =
6\nvalue = [1, 5]'),
Text(0.15272727272727274, 0.6176470588235294, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
Text(0.16727272727272727, 0.6176470588235294, 'gini = 0.0\nsamples =
5\nvalue = [0, 5]'),
Text(0.1890909090909091, 0.6764705882352942, 'x[26] <= 2.5\ngini =
0.111\nsamples = 34\nvalue = [32, 2]'),
Text(0.18181818181818182, 0.6176470588235294, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.19636363636363635, 0.6176470588235294, 'x[3] <= 22.0\ngini =
0.059\nsamples = 33\nvalue = [32, 1]'),
Text(0.1890909090909091, 0.5588235294117647, 'gini = 0.0\nsamples =
29\nvalue = [29, 0]'),
Text(0.20363636363636364, 0.5588235294117647, 'x[21] <= 13.5\ngini =
0.375\nsamples = 4\nvalue = [3, 1]'),
Text(0.19636363636363635, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
Text(0.2109090909090909, 0.5, 'gini = 0.0\nsamples = 3\nvalue = [3,
0]'),
Text(0.1890909090909091, 0.7352941176470589, 'gini = 0.0\nsamples =
4\nvalue = [0, 4]'),
Text(0.49636363636363634, 0.7941176470588235, 'x[13] <= 7.5\ngini =
0.147\nsamples = 751\nvalue = [691, 60]'),
Text(0.38545454545454544, 0.7352941176470589, 'x[29] <= 37.0\ngini =
0.133\nsamples = 713\nvalue = [662, 51]'),
Text(0.378181818181818182, 0.6764705882352942, 'x[18] <= 4.5\ngini =
0.131\nsamples = 712\nvalue = [662, 50]'),
Text(0.2772727272727273, 0.6176470588235294, 'x[25] <= 0.5\ngini =
0.098\nsamples = 540\nvalue = [512, 28]'),
Text(0.23272727272727273, 0.5588235294117647, 'x[31] <= 14.0\ngini =
0.158\nsamples = 220\nvalue = [201, 19]'),
Text(0.22545454545454546, 0.5, 'x[3] <= 28.5\ngini = 0.151\nsamples =
219\nvalue = [201, 18]'),
Text(0.19636363636363635, 0.4411764705882353, 'x[17] <= 21394.0\ngini
= 0.138\nsamples = 214\nvalue = [198, 16]'),
Text(0.16, 0.38235294117647056, 'x[14] <= 2.5\ngini = 0.091\nsamples
= 167\nvalue = [159, 8]'),
Text(0.13818181818181818, 0.3235294117647059, 'x[10] <= 99.5\ngini =
0.191\nsamples = 56\nvalue = [50, 6]'),
```

```
Text(0.13090909090909092, 0.2647058823529412, 'x[7] <= 541.5\ngini = 0.165\nsamples = 55\nvalue = [50, 5]'),
Text(0.11636363636363636, 0.20588235294117646, 'x[13] <= 5.5\ngini = 0.375\nsamples = 16\nvalue = [12, 4]'),
Text(0.10909090909090909, 0.14705882352941177, 'gini = 0.0\nsamples = 11\nvalue = [11, 0]'),
Text(0.12363636363636364, 0.14705882352941177, 'x[10] <= 92.0\ngini = 0.32\nsamples = 5\nvalue = [1, 4]'),
Text(0.11636363636363636, 0.08823529411764706, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
Text(0.13090909090909092, 0.08823529411764706, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.14545454545454545, 0.20588235294117646, 'x[18] <= 3.5\ngini = 0.05\nsamples = 39\nvalue = [38, 1]'),
Text(0.13818181818181818, 0.14705882352941177, 'gini = 0.0\nsamples = 34\nvalue = [34, 0]'),
Text(0.15272727272727274, 0.14705882352941177, 'x[8] <= 1.5\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),
Text(0.14545454545454545, 0.08823529411764706, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.16, 0.08823529411764706, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
Text(0.14545454545454545, 0.2647058823529412, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.18181818181818182, 0.3235294117647059, 'x[2] <= 0.5\ngini = 0.035\nsamples = 111\nvalue = [109, 2]'),
Text(0.16727272727272727, 0.2647058823529412, 'x[4] <= 2.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.16, 0.20588235294117646, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.17454545454545456, 0.20588235294117646, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.19636363636363635, 0.2647058823529412, 'x[7] <= 123.5\ngini = 0.018\nsamples = 109\nvalue = [108, 1]'),
Text(0.1890909090909091, 0.20588235294117646, 'x[16] <= 9572.5\ngini = 0.198\nsamples = 9\nvalue = [8, 1]'),
Text(0.18181818181818182, 0.14705882352941177, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
Text(0.19636363636363635, 0.14705882352941177, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.20363636363636364, 0.20588235294117646, 'gini = 0.0\nsamples = 100\nvalue = [100, 0]'),
Text(0.23272727272727273, 0.38235294117647056, 'x[17] <= 21522.5\ngini = 0.282\nsamples = 47\nvalue = [39, 8]'),
Text(0.22545454545454546, 0.3235294117647059, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.24, 0.3235294117647059, 'x[23] <= 1.5\ngini = 0.231\nsamples = 45\nvalue = [39, 6]'),
Text(0.22545454545454546, 0.2647058823529412, 'x[29] <= 5.5\ngini =
```

```
0.473\nsamples = 13\nvalue = [8, 5]'),
Text(0.218181818181817, 0.20588235294117646, 'x[28] <= 3.5\ngini =
0.278\nsamples = 6\nvalue = [1, 5]'),
Text(0.2109090909090909, 0.14705882352941177, 'gini = 0.0\nsamples =
5\nvalue = [0, 5]'),
Text(0.22545454545454546, 0.14705882352941177, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
Text(0.23272727272727273, 0.20588235294117646, 'gini = 0.0\nsamples =
7\nvalue = [7, 0]'),
Text(0.2545454545454545, 0.2647058823529412, 'x[11] <= 1.5\ngini =
0.061\nsamples = 32\nvalue = [31, 1]'),
Text(0.24727272727272728, 0.20588235294117646, 'x[16] <= 8245.5\ngini
= 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.24, 0.14705882352941177, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
Text(0.2545454545454545, 0.14705882352941177, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.26181818181818184, 0.20588235294117646, 'gini = 0.0\nsamples =
30\nvalue = [30, 0]'),
Text(0.2545454545454545, 0.4411764705882353, 'x[30] <= 3.5\ngini =
0.48\nsamples = 5\nvalue = [3, 2]'),
Text(0.24727272727272728, 0.38235294117647056, 'gini = 0.0\nsamples =
3\nvalue = [3, 0]'),
Text(0.26181818181818184, 0.38235294117647056, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
Text(0.24, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.32181818181818184, 0.5588235294117647, 'x[1] <= 378.0\ngini =
0.055\nsamples = 320\nvalue = [311, 9]'),
Text(0.29818181818181816, 0.5, 'x[1] <= 371.0\ngini = 0.15\nsamples =
61\nvalue = [56, 5]'),
Text(0.2909090909090909, 0.4411764705882353, 'x[23] <= 3.5\ngini =
0.124\nsamples = 60\nvalue = [56, 4]'),
Text(0.27636363636363637, 0.38235294117647056, 'x[18] <= 3.5\ngini =
0.041\nsamples = 48\nvalue = [47, 1]'),
Text(0.2690909090909091, 0.3235294117647059, 'gini = 0.0\nsamples =
41\nvalue = [41, 0]'),
Text(0.28363636363636363, 0.3235294117647059, 'x[26] <= 10.5\ngini =
0.245\nsamples = 7\nvalue = [6, 1]'),
Text(0.27636363636363637, 0.2647058823529412, 'gini = 0.0\nsamples =
6\nvalue = [6, 0]'),
Text(0.2909090909090909, 0.2647058823529412, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.3054545454545455, 0.38235294117647056, 'x[3] <= 18.5\ngini =
0.375\nsamples = 12\nvalue = [9, 3]'),
Text(0.29818181818181816, 0.3235294117647059, 'gini = 0.0\nsamples =
8\nvalue = [8, 0]'),
Text(0.31272727272727274, 0.3235294117647059, 'x[11] <= 3.5\ngini =
0.375\nsamples = 4\nvalue = [1, 3]'),
Text(0.3054545454545455, 0.2647058823529412, 'gini = 0.0\nsamples =
```

```
3\nvalue = [0, 3]'),
Text(0.32, 0.2647058823529412, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
Text(0.3054545454545455, 0.4411764705882353, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.3454545454545456, 0.5, 'x[7] <= 1291.5\ngini = 0.03\nsamples
= 259\nvalue = [255, 4]'),
Text(0.3381818181818182, 0.4411764705882353, 'gini = 0.0\nsamples =
160\nvalue = [160, 0]'),
Text(0.3527272727272727, 0.4411764705882353, 'x[7] <= 1299.5\ngini =
0.078\nsamples = 99\nvalue = [95, 4]'),
Text(0.3454545454545456, 0.38235294117647056, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.36, 0.38235294117647056, 'x[29] <= 1.5\ngini = 0.059\nsamples
= 98\nvalue = [95, 3]'),
Text(0.3418181818181818, 0.3235294117647059, 'x[17] <= 21527.0\ngini
= 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.33454545454545453, 0.2647058823529412, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.3490909090909091, 0.2647058823529412, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
Text(0.3781818181818182, 0.3235294117647059, 'x[7] <= 1310.0\ngini =
0.041\nsamples = 96\nvalue = [94, 2]'),
Text(0.36363636363636365, 0.2647058823529412, 'x[32] <= 1.5\ngini =
0.444\nsamples = 3\nvalue = [2, 1]'),
Text(0.3563636363636364, 0.20588235294117646, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.3709090909090909, 0.20588235294117646, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
Text(0.3927272727272727, 0.2647058823529412, 'x[17] <= 5355.5\ngini =
0.021\nsamples = 93\nvalue = [92, 1]'),
Text(0.38545454545454544, 0.20588235294117646, 'x[17] <= 5271.5\ngini
= 0.124\nsamples = 15\nvalue = [14, 1]'),
Text(0.3781818181818182, 0.14705882352941177, 'gini = 0.0\nsamples =
14\nvalue = [14, 0]'),
Text(0.3927272727272727, 0.14705882352941177, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.4, 0.20588235294117646, 'gini = 0.0\nsamples = 78\nvalue =
[78, 0]'),
Text(0.4790909090909091, 0.6176470588235294, 'x[26] <= 6.5\ngini =
0.223\nsamples = 172\nvalue = [150, 22]'),
Text(0.43636363636363634, 0.5588235294117647, 'x[5] <= 3.5\ngini =
0.453\nsamples = 26\nvalue = [17, 9]'),
Text(0.4218181818181818, 0.5, 'x[27] <= 4.5\ngini = 0.332\nsamples =
19\nvalue = [15, 4]'),
Text(0.41454545454545455, 0.4411764705882353, 'x[5] <= 0.5\ngini =
0.278\nsamples = 18\nvalue = [15, 3]'),
Text(0.4072727272727273, 0.38235294117647056, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
```

```
Text(0.4218181818181818, 0.38235294117647056, 'x[7] <= 841.0\ngini = 0.208\nsamples = 17\nvalue = [15, 2]'),
Text(0.4145454545454545, 0.3235294117647059, 'x[32] <= 1.5\ngini = 0.444\nsamples = 6\nvalue = [4, 2]'),
Text(0.4072727272727273, 0.2647058823529412, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.4218181818181818, 0.2647058823529412, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
Text(0.4290909090909091, 0.3235294117647059, 'gini = 0.0\nsamples = 11\nvalue = [11, 0]'),
Text(0.4290909090909091, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.4509090909090909, 0.5, 'x[31] <= 1.5\ngini = 0.408\nsamples = 7\nvalue = [2, 5]'),
Text(0.4436363636363636, 0.4411764705882353, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
Text(0.4581818181818182, 0.4411764705882353, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
Text(0.5218181818181818, 0.5588235294117647, 'x[11] <= 1.5\ngini = 0.162\nsamples = 146\nvalue = [133, 13]'),
Text(0.48, 0.5, 'x[16] <= 5414.0\ngini = 0.49\nsamples = 7\nvalue = [3, 4]'),
Text(0.4727272727272727, 0.4411764705882353, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
Text(0.4872727272727273, 0.4411764705882353, 'x[29] <= 1.5\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),
Text(0.48, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.4945454545454545, 0.38235294117647056, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
Text(0.5636363636363636, 0.5, 'x[7] <= 2037.5\ngini = 0.121\nsamples = 139\nvalue = [130, 9]'),
Text(0.5381818181818182, 0.4411764705882353, 'x[3] <= 8.5\ngini = 0.11\nsamples = 137\nvalue = [129, 8]'),
Text(0.5090909090909091, 0.38235294117647056, 'x[17] <= 2349.5\ngini = 0.045\nsamples = 87\nvalue = [85, 2]'),
Text(0.4945454545454545, 0.3235294117647059, 'x[32] <= 5.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.4872727272727273, 0.2647058823529412, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.5018181818181818, 0.2647058823529412, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.5236363636363637, 0.3235294117647059, 'x[1] <= 204.5\ngini = 0.023\nsamples = 85\nvalue = [84, 1]'),
Text(0.5163636363636364, 0.2647058823529412, 'x[32] <= 4.0\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),
Text(0.5090909090909091, 0.20588235294117646, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
Text(0.5236363636363637, 0.20588235294117646, 'gini = 0.0\nsamples =
```

```
1\nvalue = [0, 1]'),
Text(0.5309090909090909, 0.2647058823529412, 'gini = 0.0\nsamples =
80\nvalue = [80, 0]'),
Text(0.5672727272727273, 0.38235294117647056, 'x[7] <= 361.5\ngini =
0.211\nsamples = 50\nvalue = [44, 6]'),
Text(0.5527272727272727, 0.3235294117647059, 'x[32] <= 5.5\ngini =
0.5\nsamples = 6\nvalue = [3, 3]'),
Text(0.5454545454545454, 0.2647058823529412, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
Text(0.56, 0.2647058823529412, 'gini = 0.0\nsamples = 3\nvalue = [3,
0]'),
Text(0.5818181818181818, 0.3235294117647059, 'x[30] <= 11.5\ngini =
0.127\nsamples = 44\nvalue = [41, 3]'),
Text(0.5745454545454546, 0.2647058823529412, 'x[7] <= 1842.5\ngini =
0.089\nsamples = 43\nvalue = [41, 2]'),
Text(0.56, 0.20588235294117646, 'x[30] <= 0.5\ngini = 0.049\nsamples
= 40\nvalue = [39, 1]'),
Text(0.5527272727272727, 0.14705882352941177, 'x[26] <= 9.5\ngini =
0.278\nsamples = 6\nvalue = [5, 1]'),
Text(0.5454545454545454, 0.08823529411764706, 'x[23] <= 1.5\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.5381818181818182, 0.029411764705882353, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
Text(0.5527272727272727, 0.029411764705882353, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.56, 0.08823529411764706, 'gini = 0.0\nsamples = 4\nvalue = [4,
0]'),
Text(0.5672727272727273, 0.14705882352941177, 'gini = 0.0\nsamples =
34\nvalue = [34, 0]'),
Text(0.5890909090909091, 0.20588235294117646, 'x[31] <= 0.5\ngini =
0.444\nsamples = 3\nvalue = [2, 1]'),
Text(0.5818181818181818, 0.14705882352941177, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.5963636363636363, 0.14705882352941177, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
Text(0.5890909090909091, 0.2647058823529412, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.5890909090909091, 0.4411764705882353, 'x[7] <= 2045.0\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.5818181818181818, 0.38235294117647056, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.5963636363636363, 0.38235294117647056, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
Text(0.3927272727272727, 0.6764705882352942, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.6072727272727273, 0.7352941176470589, 'x[11] <= 1.5\ngini =
0.361\nsamples = 38\nvalue = [29, 9]'),
Text(0.6, 0.6764705882352942, 'gini = 0.0\nsamples = 2\nvalue = [0,
2]'),
```



```
Text(0.6145454545454545, 0.6764705882352942, 'x[14] <= 1.5\ngini = 0.313\nsamples = 36\nvalue = [29, 7]'),
Text(0.5963636363636363, 0.6176470588235294, 'x[32] <= 1.0\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
Text(0.5890909090909091, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.6036363636363636, 0.5588235294117647, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
Text(0.6327272727272727, 0.6176470588235294, 'x[28] <= 3.5\ngini = 0.219\nsamples = 32\nvalue = [28, 4]'),
Text(0.6181818181818182, 0.5588235294117647, 'x[16] <= 2070.0\ngini = 0.069\nsamples = 28\nvalue = [27, 1]'),
Text(0.6109090909090909, 0.5, 'x[7] <= 485.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.6036363636363636, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.6181818181818182, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.6254545454545455, 0.5, 'gini = 0.0\nsamples = 26\nvalue = [26, 0]'),
Text(0.6472727272727272, 0.5588235294117647, 'x[10] <= 48.5\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
Text(0.64, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.6545454545454545, 0.5, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
Text(0.7738636363636363, 0.8529411764705882, 'x[16] <= 2488.0\ngini = 0.387\nsamples = 309\nvalue = [228, 81]'),
Text(0.6836363636363636, 0.7941176470588235, 'x[1] <= 482.5\ngini = 0.465\nsamples = 38\nvalue = [14, 24]'),
Text(0.6763636363636364, 0.7352941176470589, 'gini = 0.0\nsamples = 11\nvalue = [0, 11]'),
Text(0.6909090909090909, 0.7352941176470589, 'x[3] <= 15.5\ngini = 0.499\nsamples = 27\nvalue = [14, 13]'),
Text(0.6836363636363636, 0.6764705882352942, 'x[18] <= 3.5\ngini = 0.463\nsamples = 22\nvalue = [14, 8]'),
Text(0.6690909090909091, 0.6176470588235294, 'x[10] <= 71.5\ngini = 0.245\nsamples = 14\nvalue = [12, 2]'),
Text(0.6618181818181819, 0.5588235294117647, 'gini = 0.0\nsamples = 11\nvalue = [11, 0]'),
Text(0.6763636363636364, 0.5588235294117647, 'x[4] <= 1.5\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),
Text(0.6690909090909091, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.6836363636363636, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
Text(0.6981818181818182, 0.6176470588235294, 'x[7] <= 1186.5\ngini = 0.375\nsamples = 8\nvalue = [2, 6]'),
Text(0.6909090909090909, 0.5588235294117647, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
```

```
Text(0.7054545454545454, 0.5588235294117647, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
Text(0.6981818181818182, 0.6764705882352942, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
Text(0.8640909090909091, 0.7941176470588235, 'x[15] <= 1.5\ngini = 0.332\nsamples = 271\nvalue = [214, 57]'),
Text(0.8045454545454546, 0.7352941176470589, 'x[13] <= 7.5\ngini = 0.248\nsamples = 193\nvalue = [165, 28]'),
Text(0.78, 0.6764705882352942, 'x[7] <= 2022.5\ngini = 0.23\nsamples = 189\nvalue = [164, 25]'),
Text(0.7527272727272727, 0.6176470588235294, 'x[27] <= 0.5\ngini = 0.217\nsamples = 186\nvalue = [163, 23]'),
Text(0.72, 0.5588235294117647, 'x[28] <= 2.5\ngini = 0.48\nsamples = 10\nvalue = [6, 4]'),
Text(0.7127272727272728, 0.5, 'x[1] <= 293.0\ngini = 0.32\nsamples = 5\nvalue = [1, 4]'),
Text(0.7054545454545454, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.72, 0.4411764705882353, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
Text(0.7272727272727273, 0.5, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
Text(0.7854545454545454, 0.5588235294117647, 'x[8] <= 2.5\ngini = 0.193\nsamples = 176\nvalue = [157, 19]'),
Text(0.7527272727272727, 0.5, 'x[29] <= 7.5\ngini = 0.346\nsamples = 54\nvalue = [42, 12]'),
Text(0.7345454545454545, 0.4411764705882353, 'x[31] <= 1.5\ngini = 0.438\nsamples = 34\nvalue = [23, 11]'),
Text(0.72, 0.38235294117647056, 'x[10] <= 59.0\ngini = 0.337\nsamples = 28\nvalue = [22, 6]'),
Text(0.7127272727272728, 0.3235294117647059, 'x[9] <= 0.5\ngini = 0.496\nsamples = 11\nvalue = [5, 6]'),
Text(0.7054545454545454, 0.2647058823529412, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
Text(0.72, 0.2647058823529412, 'x[5] <= 4.5\ngini = 0.278\nsamples = 6\nvalue = [5, 1]'),
Text(0.7127272727272728, 0.20588235294117646, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
Text(0.7272727272727273, 0.20588235294117646, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
Text(0.7272727272727273, 0.3235294117647059, 'gini = 0.0\nsamples = 17\nvalue = [17, 0]'),
Text(0.7490909090909091, 0.38235294117647056, 'x[18] <= 2.5\ngini = 0.278\nsamples = 6\nvalue = [1, 5]'),
Text(0.7418181818181818, 0.3235294117647059, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.7563636363636363, 0.3235294117647059, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
Text(0.7709090909090909, 0.4411764705882353, 'x[7] <= 1923.5\ngini =
```

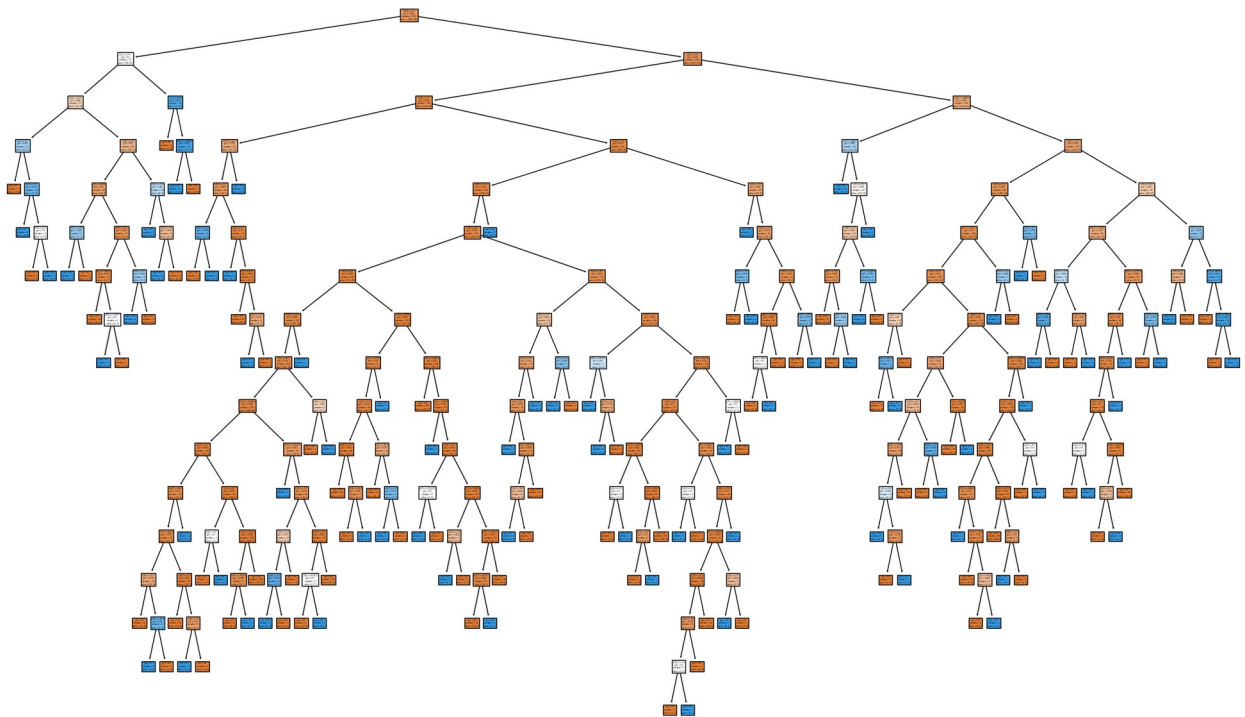
```
0.095\nsamples = 20\nvalue = [19, 1]'),
Text(0.7636363636363637, 0.38235294117647056, 'gini = 0.0\nsamples =
19\nvalue = [19, 0]'),
Text(0.7781818181818182, 0.38235294117647056, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.8181818181818182, 0.5, 'x[16] <= 19780.0\ngini = 0.108\
nsamples = 122\nvalue = [115, 7]'),
Text(0.8109090909090909, 0.4411764705882353, 'x[27] <= 5.5\ngini =
0.094\nsamples = 121\nvalue = [115, 6]'),
Text(0.7927272727272727, 0.38235294117647056, 'x[10] <= 41.5\ngini =
0.066\nsamples = 117\nvalue = [113, 4]'),
Text(0.7781818181818182, 0.3235294117647059, 'x[17] <= 5462.0\ngini =
0.227\nsamples = 23\nvalue = [20, 3]'),
Text(0.7709090909090909, 0.2647058823529412, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.7854545454545454, 0.2647058823529412, 'x[21] <= 17.5\ngini =
0.165\nsamples = 22\nvalue = [20, 2]'),
Text(0.7781818181818182, 0.20588235294117646, 'gini = 0.0\nsamples =
16\nvalue = [16, 0]'),
Text(0.7927272727272727, 0.20588235294117646, 'x[1] <= 576.0\ngini =
0.444\nsamples = 6\nvalue = [4, 2]'),
Text(0.7854545454545454, 0.14705882352941177, 'gini = 0.0\nsamples =
4\nvalue = [4, 0]'),
Text(0.8, 0.14705882352941177, 'gini = 0.0\nsamples = 2\nvalue = [0,
2]'),
Text(0.8072727272727273, 0.3235294117647059, 'x[10] <= 97.5\ngini =
0.021\nsamples = 94\nvalue = [93, 1]'),
Text(0.8, 0.2647058823529412, 'gini = 0.0\nsamples = 89\nvalue = [89,
0]'),
Text(0.8145454545454546, 0.2647058823529412, 'x[28] <= 2.5\ngini =
0.32\nsamples = 5\nvalue = [4, 1]'),
Text(0.8072727272727273, 0.20588235294117646, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.8218181818181818, 0.20588235294117646, 'gini = 0.0\nsamples =
4\nvalue = [4, 0]'),
Text(0.8290909090909091, 0.38235294117647056, 'x[9] <= 0.5\ngini =
0.5\nsamples = 4\nvalue = [2, 2]'),
Text(0.8218181818181818, 0.3235294117647059, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
Text(0.8363636363636363, 0.3235294117647059, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
Text(0.8254545454545454, 0.4411764705882353, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.8072727272727273, 0.6176470588235294, 'x[32] <= 2.5\ngini =
0.444\nsamples = 3\nvalue = [1, 2]'),
Text(0.8, 0.5588235294117647, 'gini = 0.0\nsamples = 2\nvalue = [0,
2]'),
Text(0.8145454545454546, 0.5588235294117647, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
Text(0.8290909090909091, 0.6764705882352942, 'x[4] <= 3.5\ngini =
```

```
0.375\nsamples = 4\nvalue = [1, 3]'),
Text(0.8218181818181818, 0.6176470588235294, 'gini = 0.0\nsamples =
3\nvalue = [0, 3]'),
Text(0.8363636363636363, 0.6176470588235294, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
Text(0.9236363636363636, 0.7352941176470589, 'x[3] <= 11.0\ngini =
0.467\nsamples = 78\nvalue = [49, 29]'),
Text(0.8836363636363637, 0.6764705882352942, 'x[16] <= 3703.0\ngini =
0.375\nsamples = 56\nvalue = [42, 14]'),
Text(0.8545454545454545, 0.6176470588235294, 'x[14] <= 3.5\ngini =
0.486\nsamples = 12\nvalue = [5, 7]'),
Text(0.84, 0.5588235294117647, 'x[10] <= 98.0\ngini = 0.245\nsamples
= 7\nvalue = [1, 6]'),
Text(0.8327272727272728, 0.5, 'gini = 0.0\nsamples = 6\nvalue = [0,
6]'),
Text(0.8472727272727273, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
Text(0.8690909090909091, 0.5588235294117647, 'x[30] <= 7.0\ngini =
0.32\nsamples = 5\nvalue = [4, 1]'),
Text(0.8618181818181818, 0.5, 'gini = 0.0\nsamples = 4\nvalue = [4,
0]'),
Text(0.8763636363636363, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
Text(0.9127272727272727, 0.6176470588235294, 'x[17] <= 25737.0\ngini
= 0.268\nsamples = 44\nvalue = [37, 7]'),
Text(0.8981818181818182, 0.5588235294117647, 'x[29] <= 22.5\ngini =
0.18\nsamples = 40\nvalue = [36, 4]'),
Text(0.8909090909090909, 0.5, 'x[21] <= 21.5\ngini = 0.142\nsamples =
39\nvalue = [36, 3]'),
Text(0.8836363636363637, 0.4411764705882353, 'x[27] <= 0.5\ngini =
0.1\nsamples = 38\nvalue = [36, 2]'),
Text(0.8690909090909091, 0.38235294117647056, 'x[31] <= 3.5\ngini =
0.5\nsamples = 2\nvalue = [1, 1]'),
Text(0.8618181818181818, 0.3235294117647059, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
Text(0.8763636363636363, 0.3235294117647059, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.8981818181818182, 0.38235294117647056, 'x[16] <= 4035.0\ngini
= 0.054\nsamples = 36\nvalue = [35, 1]'),
Text(0.8909090909090909, 0.3235294117647059, 'x[3] <= 5.0\ngini =
0.444\nsamples = 3\nvalue = [2, 1]'),
Text(0.8836363636363637, 0.2647058823529412, 'gini = 0.0\nsamples =
2\nvalue = [2, 0]'),
Text(0.8981818181818182, 0.2647058823529412, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
Text(0.9054545454545454, 0.3235294117647059, 'gini = 0.0\nsamples =
33\nvalue = [33, 0]'),
Text(0.8981818181818182, 0.4411764705882353, 'gini = 0.0\nsamples =
1\nvalue = [0, 1]'),
```

```

Text(0.9054545454545454, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0,
1]'),
Text(0.9272727272727272, 0.5588235294117647, 'x[13] <= 4.5\ngini =
0.375\nsamples = 4\nvalue = [1, 3]'),
Text(0.92, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
Text(0.9345454545454546, 0.5, 'gini = 0.0\nsamples = 3\nvalue = [0,
3]'),
Text(0.9636363636363636, 0.6764705882352942, 'x[2] <= 1.5\ngini =
0.434\nsamples = 22\nvalue = [7, 15]'),
Text(0.9490909090909091, 0.6176470588235294, 'x[4] <= 1.5\ngini =
0.408\nsamples = 7\nvalue = [5, 2]'),
Text(0.9418181818181818, 0.5588235294117647, 'gini = 0.0\nsamples =
2\nvalue = [0, 2]'),
Text(0.9563636363636364, 0.5588235294117647, 'gini = 0.0\nsamples =
5\nvalue = [5, 0]'),
Text(0.9781818181818182, 0.6176470588235294, 'x[7] <= 184.0\ngini =
0.231\nsamples = 15\nvalue = [2, 13]'),
Text(0.9709090909090909, 0.5588235294117647, 'gini = 0.0\nsamples =
1\nvalue = [1, 0]'),
Text(0.9854545454545455, 0.5588235294117647, 'x[10] <= 34.0\ngini =
0.133\nsamples = 14\nvalue = [1, 13]'),
Text(0.9781818181818182, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1,
0]'),
Text(0.9927272727272727, 0.5, 'gini = 0.0\nsamples = 13\nvalue = [0,
13]')]]

```



```

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,scoring="accuracy")

```

```

grid_search.fit(x_train,y_train)

```

C:\Users\himaj\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\himaj\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\himaj\anaconda3\Lib\site-packages\sklearn\base.py", line 1144, in wrapper

estimator.\_validate\_params()

File "C:\Users\himaj\anaconda3\Lib\site-packages\sklearn\base.py", line 637, in \_validate\_params  
 validate\_parameter\_constraints(

File "C:\Users\himaj\anaconda3\Lib\site-packages\sklearn\utils\\_param\_validation.py", line 95, in validate\_parameter\_constraints  
 raise InvalidParameterError(

sklearn.utils.\_param\_validation.InvalidParameterError: The 'max\_features' parameter of DecisionTreeClassifier must be an int in the range [1, inf), a float in the range (0.0, 1.0], a str among {'sqrt', 'log2'} or None. Got 'auto' instead.

warnings.warn(some\_fits\_failed\_message, FitFailedWarning)

C:\Users\himaj\anaconda3\Lib\site-packages\sklearn\model\_selection\\_search.py:976: UserWarning: One or more of the test scores are non-finite: [ nan nan 0.84013704 0.84013704 0.84013704 0.84013704

nan nan 0.83248107 0.84013704 0.83588532 0.84013704

```

nan      nan 0.83417959 0.84013704 0.83163722 0.83843491
nan      nan 0.83671475 0.84524342 0.82312297 0.83759827
nan      nan 0.82481789 0.82311937 0.82482149 0.83672917
nan      nan 0.84013704 0.84013704 0.84013704 0.84013704
nan      nan 0.84013704 0.83673278 0.83673278 0.84013704
nan      nan 0.82992066 0.83758384 0.83929679 0.83503787
nan      nan 0.84438154 0.8409881  0.83247025 0.84439596
nan      nan 0.83161558 0.83164443 0.83842409 0.83247386]
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': 'gini',
 'max_depth': 4,
 'max_features': 'sqrt',
 'splitter': 'random'}

```

```

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
No	0.84	1.00	0.91	245
Yes	0.50	0.02	0.04	49
accuracy			0.83	294
macro avg	0.67	0.51	0.47	294
weighted avg	0.78	0.83	0.76	294

# Model Building using Random Forest

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

accuracy_score(y_test,pred)

0.8333333333333334

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

rfc_cv=
GridSearchCV(rfc,param_grid=forest_params,cv=5,scoring="accuracy")

rfc_cv.fit(x_train,y_train)

C:\Users\himaj\anaconda3\Lib\site-packages\sklearn\model_selection\
_validation.py:425: FitFailedWarning:
25 fits failed out of a total of 350.
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:
-----
-----
25 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\himaj\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\himaj\anaconda3\Lib\site-packages\sklearn\base.py",
line 1144, in wrapper
    estimator._validate_params()
  File "C:\Users\himaj\anaconda3\Lib\site-packages\sklearn\base.py",
line 637, in _validate_params
    validate_parameter_constraints(
  File "C:\Users\himaj\anaconda3\Lib\site-packages\sklearn\utils\
_param_validation.py", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The
'max_features' parameter of RandomForestClassifier must be an int in
the range [1, inf), a float in the range (0.0, 1.0], a str among
{'sqrt', 'log2'} or None. Got 0 instead.

warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\Users\himaj\anaconda3\Lib\site-packages\sklearn\model_selection\
_search.py:976: UserWarning: One or more of the test scores are non-
finite: [          nan  0.84524342  0.84864407  0.84949513  0.85033898
```



```

0.85119005
0.85034259 0.84608727 0.84864407 0.8520339 0.85033538 0.8520375
0.85033177 0.85458348 nan 0.84524342 0.84609448 0.84523621
0.84608366 0.85033898 0.84778579 0.85373963 0.85713307 0.85033177
0.85204472 0.85288857 0.84864046 0.8562784 nan 0.84609448
0.84694555 0.84523621 0.85033898 0.85119365 0.85458709 0.85119005
0.85032816 0.8520375 0.85288857 0.85883159 0.85118644 0.85288496
nan 0.84524342 0.84609088 0.84949874 0.84778579 0.84694194
0.85204832 0.85118283 0.84439235 0.85033177 0.85033538 0.85628922
0.84948792 0.85204111 nan 0.84609448 0.8503462 0.84694194
0.84864407 0.84863686 0.8520375 0.85117923 0.85033898 0.85033177
0.85714028 0.85798774 0.85119005 0.84948792]
warnings.warn(

GridSearchCV(cv=5, 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
No	0.86	0.99	0.92	245
Yes	0.83	0.20	0.33	49
accuracy			0.86	294
macro avg	0.85	0.60	0.63	294
weighted avg	0.86	0.86	0.82	294

```

rfc_cv.best_params_
{'max_depth': 12, 'max_features': 11}

```