ASSIGNMENT 4:

HARSH KUMAR

harsh.kumar2021d@vitstudent.ac.in

```
# HARSH KUMAR
# harsh.kumar2021d@vitstudent.ac.in
```

```
ASSIGNMENT 4:
1.Download the Employee Attrition Dataset
https://www.kaggle.com/datasets/patelprashant/employee-attrition
2.Perfrom Data Preprocessing
3.Model Building using Logistic Regression and Decision Tree and Random Forest
4.Calculate Performance metrics
```

DATA COLLECTION

```
-> Collect the dataset or create the dataset
```

DATA PREPROCESSING

```
-> Import the libraries.
-> Import the dataset.
-> Check for null values.
-> Data Visualization.
-> Outlier Detection.
-> Splitting Dependent and Independent variables.
-> Encoding.
-> Feature Scaling.
-> Splitting Dataset into Train and Test.
```

MODEL BUILDING

```
-> Import the model building libraries.
-> Initializing the model.
-> Training and testing the model.
-> Evaluation of Model.
-> Save the model.
```

APPLICATION BUILDING

```
-> Create an HTML file.
-> Build a Python code.
```

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

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

```
df.head()
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1
1	49	No	Travel_Frequently	279	Research &	8	1	Life Sciences	1	2

```
df.shape
```

```
(1470, 35)
```

```
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	EmployeeNumber	EnvironmentSatisfaction	HourlyRate
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	1470.000000	1470.000000
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	2.721769	65.891156
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	1.093082	20.329428
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	1.000000	30.000000
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	2.000000	48.000000
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	3.000000	66.000000
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	4.000000	83.750000
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	4.000000	100.000000

8 rows × 26 columns

```
# Detection of null values
df.isnull().any()
```

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

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

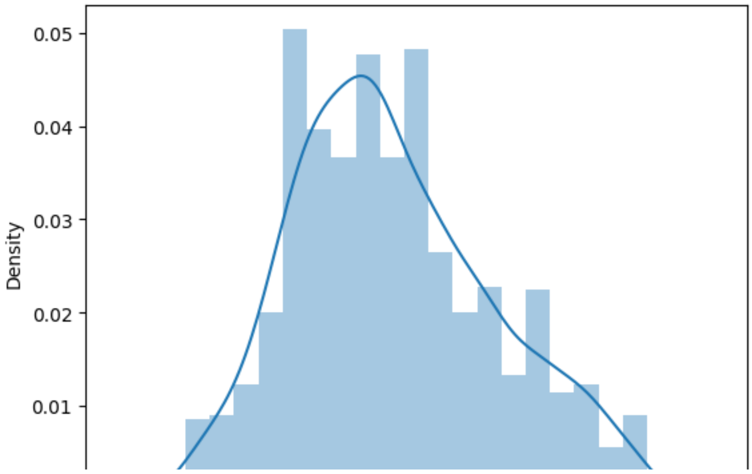
```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber            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

DATA VISUALIZATION

```
# @title DATA VISUALIZATION
```

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

```
<ipython-input-779-eb5d36fb3f65>:3: 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()
```

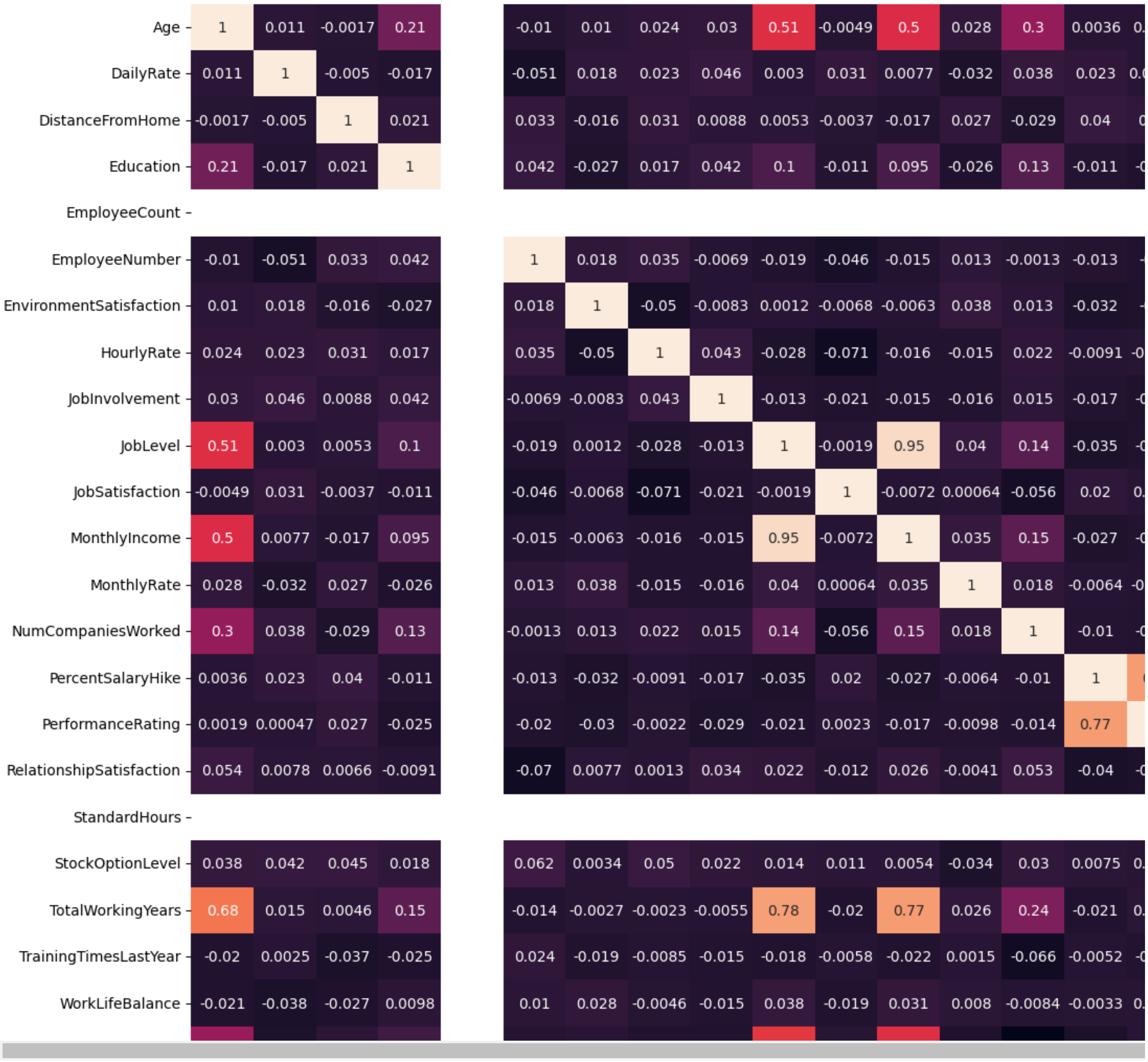
```
<ipython-input-780-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ve
  df.corr()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	Hou
Age	1.000000	0.010661	-0.001686	0.208034	NaN	-0.010145	0.010146	(
DailyRate	0.010661	1.000000	-0.004985	-0.016806	NaN	-0.050990	0.018355	(
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042	NaN	0.032916	-0.016075	(
Education	0.208034	-0.016806	0.021042	1.000000	NaN	0.042070	-0.027128	(
EmployeeCount	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
EmployeeNumber	-0.010145	-0.050990	0.032916	0.042070	NaN	1.000000	0.017621	(
EnvironmentSatisfaction	0.010146	0.018355	-0.016075	-0.027128	NaN	0.017621	1.000000	-(
HourlyRate	0.024287	0.023381	0.031131	0.016775	NaN	0.035179	-0.049857	1
JobInvolvement	0.029820	0.046135	0.008783	0.042438	NaN	-0.006888	-0.008278	(
JobLevel	0.509604	0.002966	0.005303	0.101589	NaN	-0.018519	0.001212	-(
JobSatisfaction	-0.004892	0.030571	-0.003669	-0.011296	NaN	-0.046247	-0.006784	-(
MonthlyIncome	0.497855	0.007707	-0.017014	0.094961	NaN	-0.014829	-0.006259	-(
MonthlyRate	0.028051	-0.032182	0.027473	-0.026084	NaN	0.012648	0.037600	-(
NumCompaniesWorked	0.299635	0.038153	-0.029251	0.126317	NaN	-0.001251	0.012594	(
PercentSalaryHike	0.003634	0.022704	0.040235	-0.011111	NaN	-0.012944	-0.031701	-(
PerformanceRating	0.001904	0.000473	0.027110	-0.024539	NaN	-0.020359	-0.029548	-(
RelationshipSatisfaction	0.053535	0.007846	0.006557	-0.009118	NaN	-0.069861	0.007665	(
StandardHours	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
StockOptionLevel	0.037510	0.042143	0.044872	0.018422	NaN	0.062227	0.003432	(
TotalWorkingYears	0.680381	0.014515	0.004628	0.148280	NaN	-0.014365	-0.002693	-(
TrainingTimesLastYear	-0.019621	0.002453	-0.036942	-0.025100	NaN	0.023603	-0.019359	-(
WorkLifeBalance	-0.021490	-0.037848	-0.026556	0.009819	NaN	0.010309	0.027627	-(
YearsAtCompany	0.311309	-0.034055	0.009508	0.069114	NaN	-0.011240	0.001458	-(
YearsInCurrentRole	0.212901	0.009932	0.018845	0.060236	NaN	-0.008416	0.018007	-(
YearsSinceLastPromotion	0.216513	-0.033229	0.010029	0.054254	NaN	-0.009019	0.016194	-(
YearsWithCurrManager	0.202089	-0.026363	0.014406	0.069065	NaN	-0.009197	-0.004999	-(

26 rows × 26 columns

```
plt.subplots(figsize=(25,15))
sns.heatmap(df.corr(), annot = True)
```

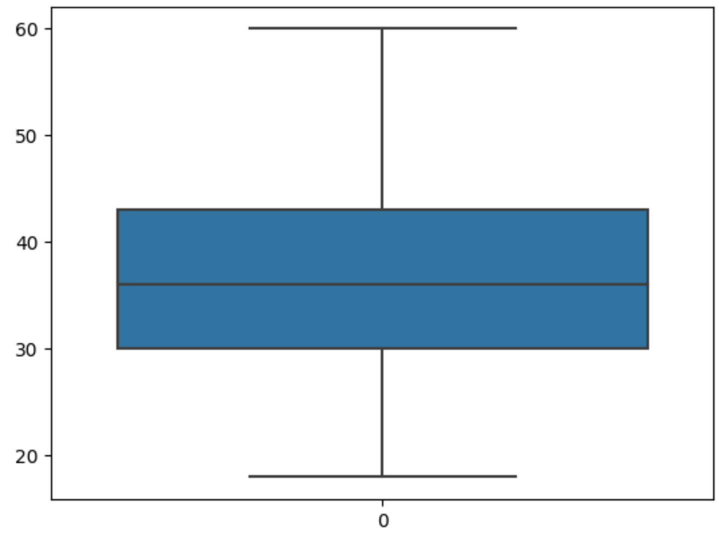
```
<ipython-input-781-dec6949f63e9>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ve
  sns.heatmap(df.corr(), annot = True)
<Axes: >
```



OUTLIER TREATMENT

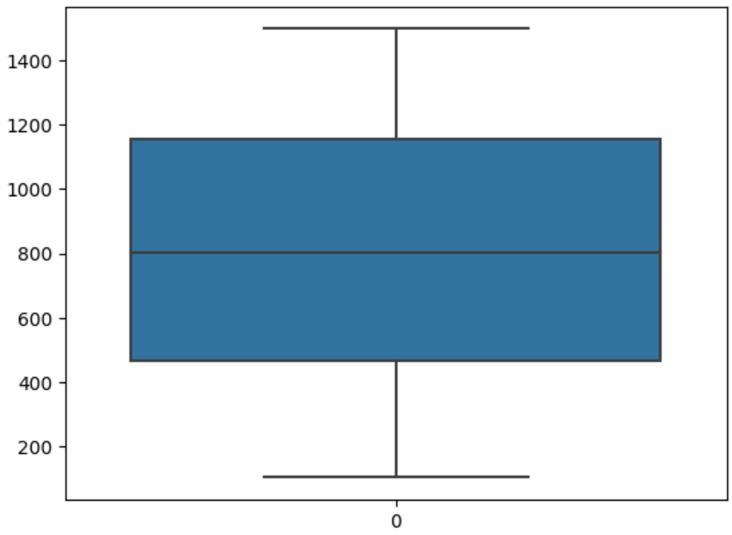
```
sns.boxplot(df["Age"])
```

```
<Axes: >
```



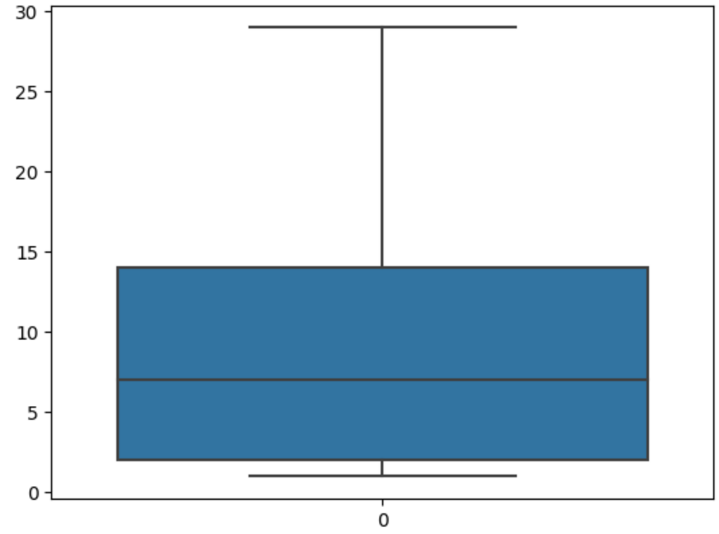
```
sns.boxplot(df["DailyRate"])
```

```
<Axes: >
```



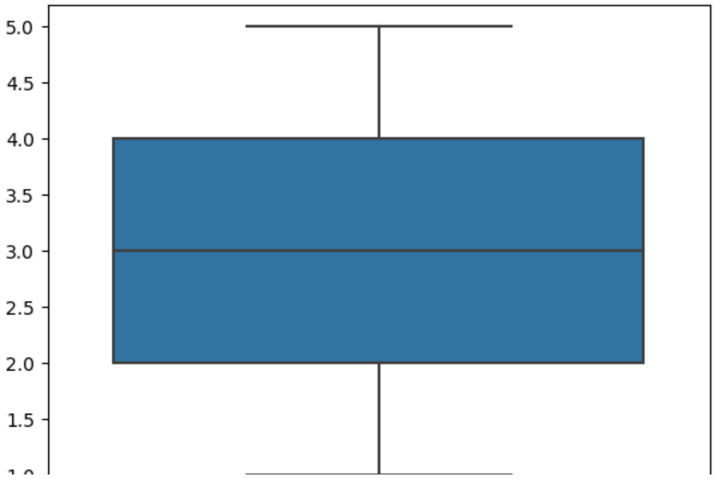
```
sns.boxplot(df["DistanceFromHome"])
```

```
<Axes: >
```



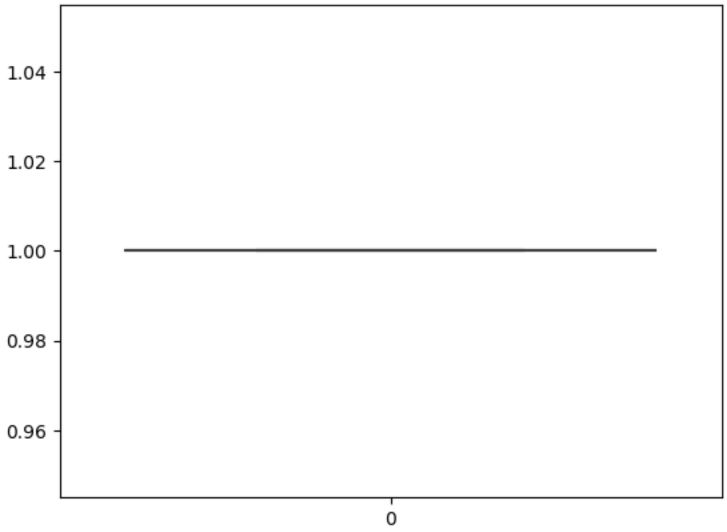
```
sns.boxplot(df["Education"])
```

```
<Axes: >
```



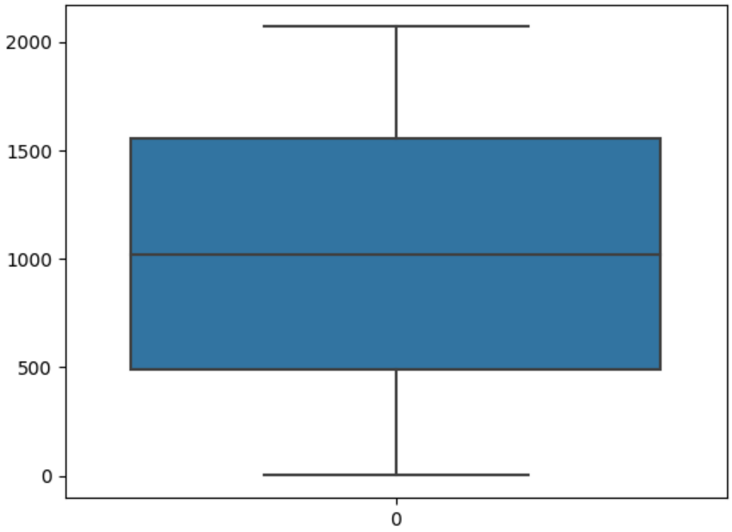
```
sns.boxplot(df["EmployeeCount"])
```

```
<Axes: >
```



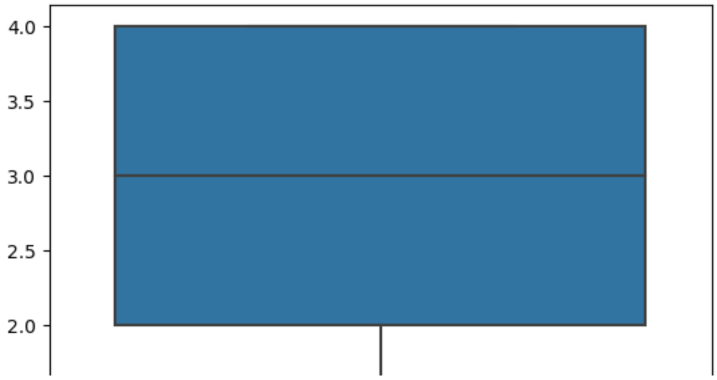
```
sns.boxplot(df["EmployeeNumber"])
```

```
<Axes: >
```



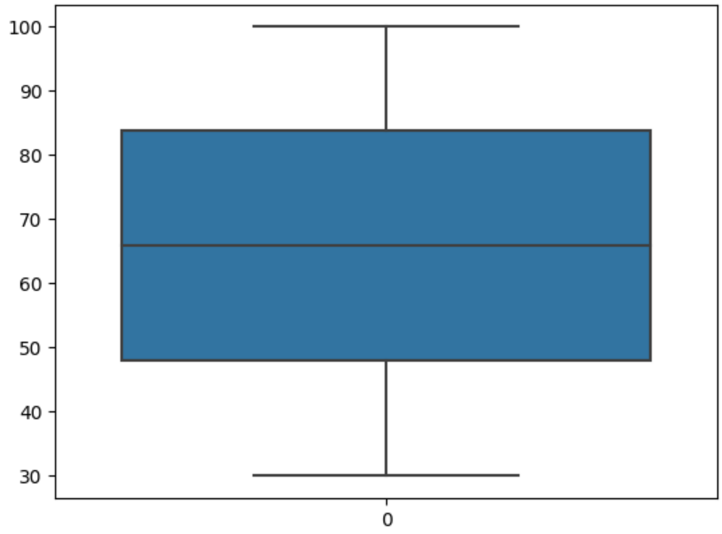
```
sns.boxplot(df["EnvironmentSatisfaction"])
```

```
<Axes: >
```



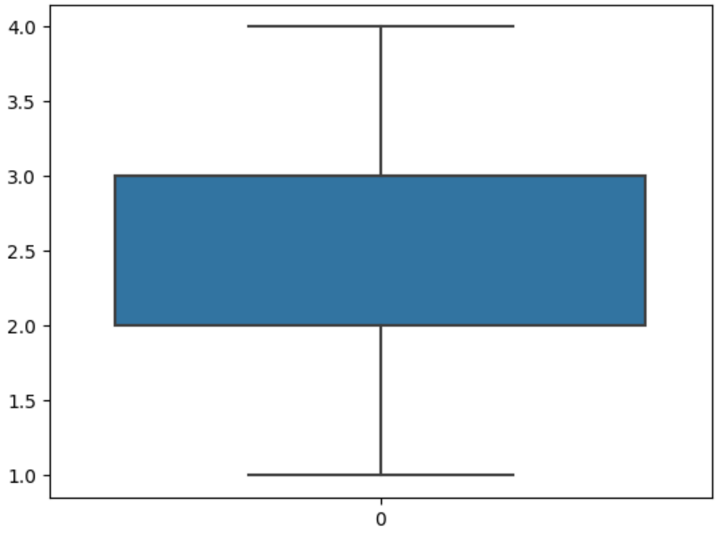
```
sns.boxplot(df["HourlyRate"])
```

```
<Axes: >
```



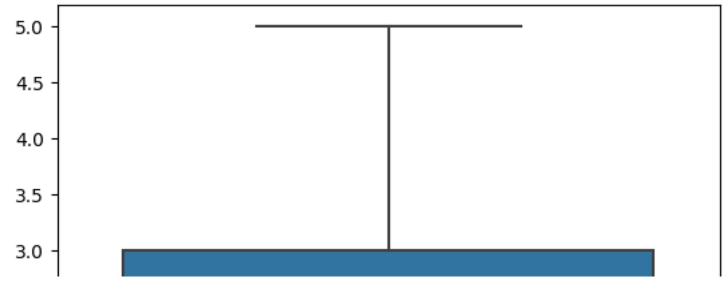
```
sns.boxplot(df["JobInvolvement"])
```

```
<Axes: >
```



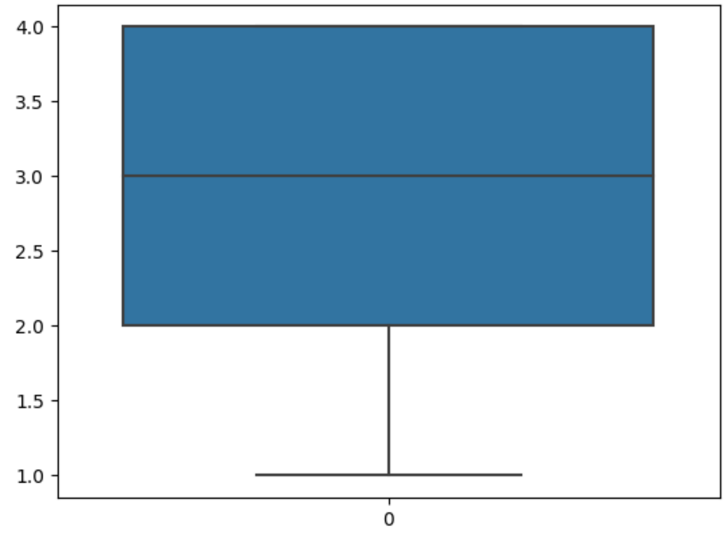
```
sns.boxplot(df["JobLevel"])
```

```
<Axes: >
```



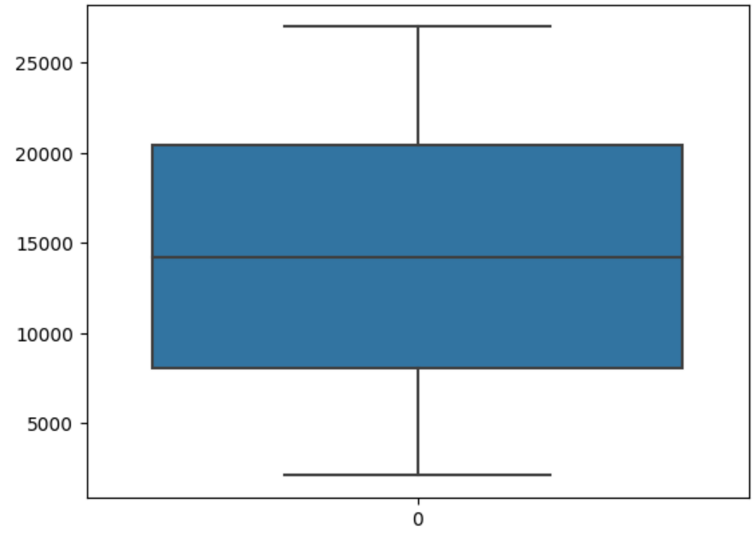
```
sns.boxplot(df["JobSatisfaction"])
```

```
<Axes: >
```



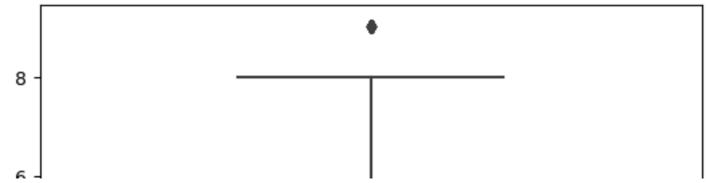
```
sns.boxplot(df["MonthlyRate"])
```

```
<Axes: >
```



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

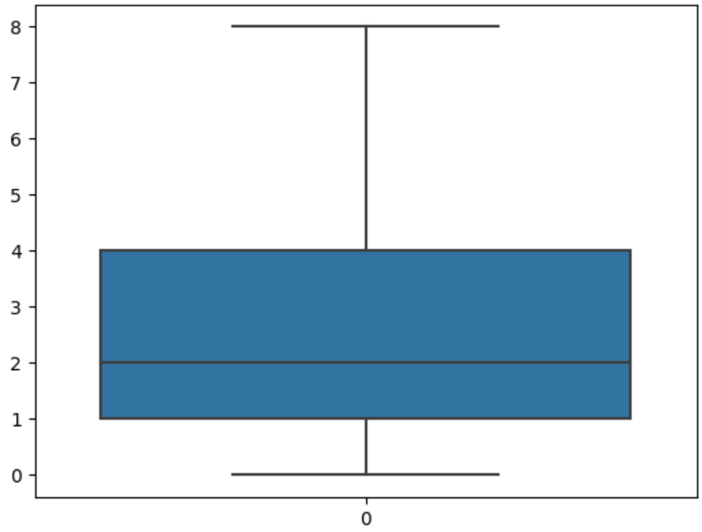
```
<Axes: >
```



```
Q1 = df.NumCompaniesWorked.quantile(0.25)
Q3 = df.NumCompaniesWorked.quantile(.75)
IQR = Q3-Q1
upper_lt = Q3 + 1.5*IQR
lower_lt = Q1 - 1.5*IQR

median = df["NumCompaniesWorked"].median()

df['NumCompaniesWorked'] = np.where(df['NumCompaniesWorked']>upper_lt, median, df['NumCompaniesWorked'])
df['NumCompaniesWorked'] = np.where(df['NumCompaniesWorked']<lower_lt, median, df['NumCompaniesWorked'])
```

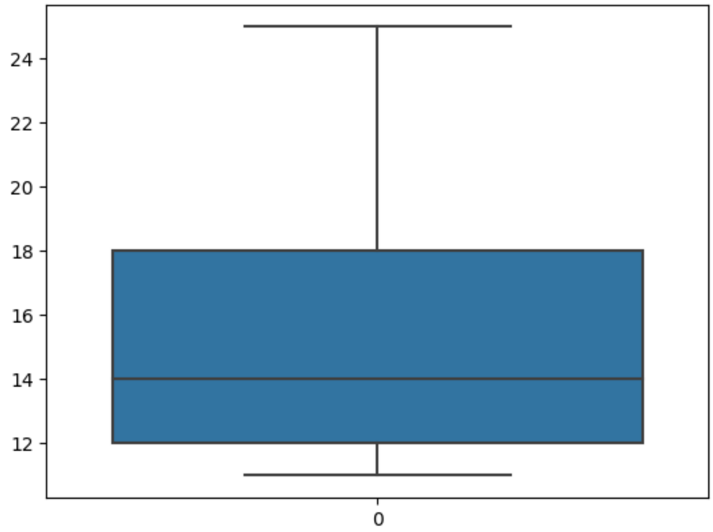
```
sns.boxplot(df["NumCompaniesWorked"])
```

```
<Axes: >
```



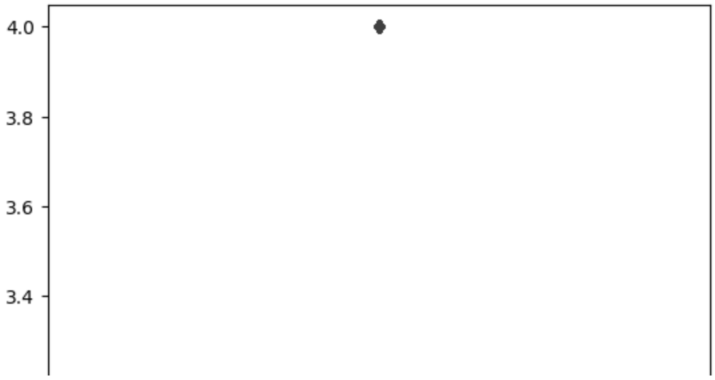
```
sns.boxplot(df["PercentSalaryHike"])
```

```
<Axes: >
```



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

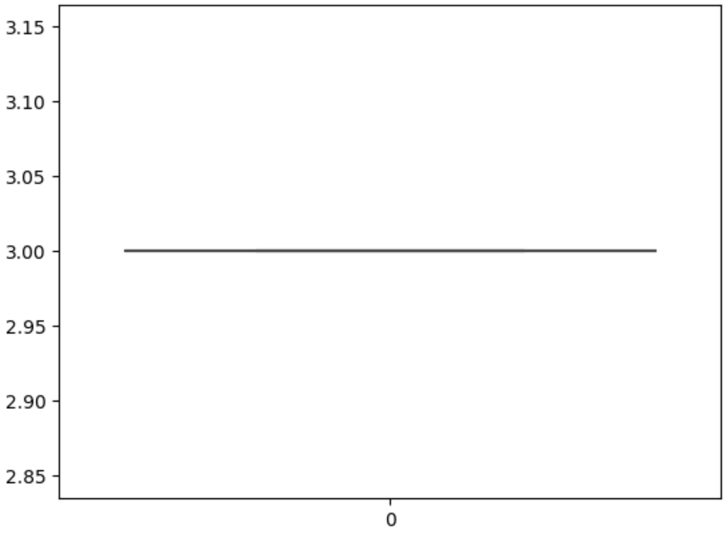
```
<Axes: >
```



```
Q1 = df.PerformanceRating.quantile(0.25)
Q3 = df.PerformanceRating.quantile(.75)
IQR = Q3-Q1
upper_lt = Q3 + 1.5*IQR
lower_lt = Q1 - 1.5*IQR

median = df["PerformanceRating"].median()

df['PerformanceRating'] = np.where(df['PerformanceRating']>upper_lt, median, df['PerformanceRating'])
df['PerformanceRating'] = np.where(df['PerformanceRating']<lower_lt, median, df['PerformanceRating'])
```

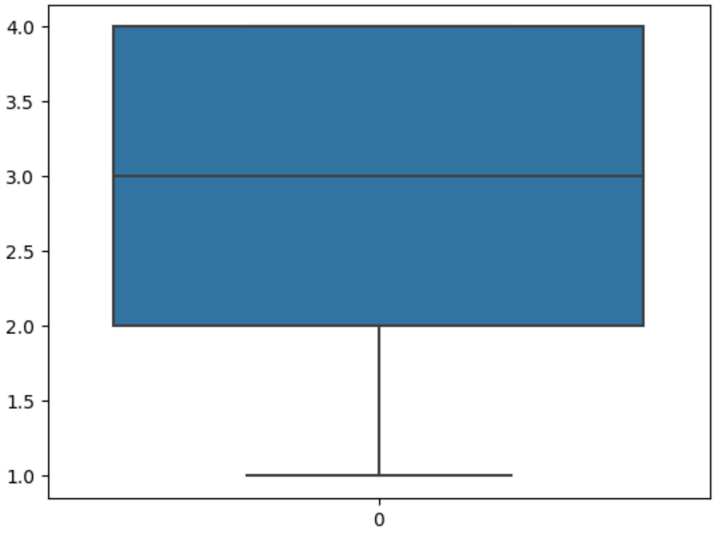
```
sns.boxplot(df["PerformanceRating"], color = 'red')
```

```
<Axes: >
```



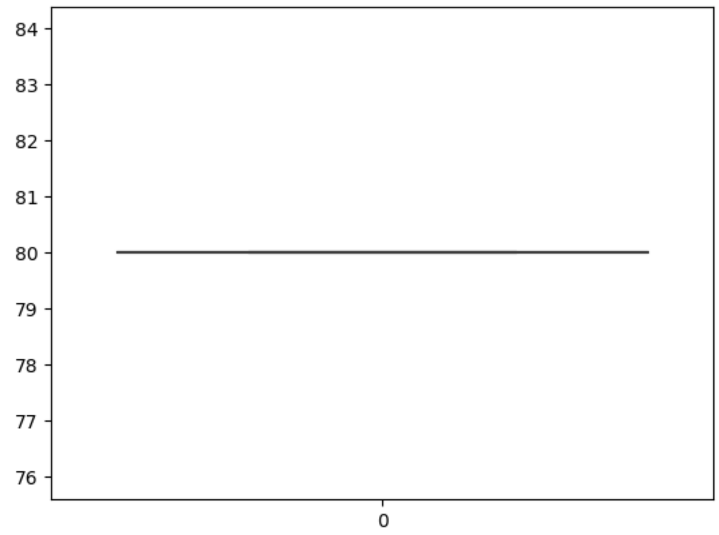
```
sns.boxplot(df["RelationshipSatisfaction"])
```

```
<Axes: >
```



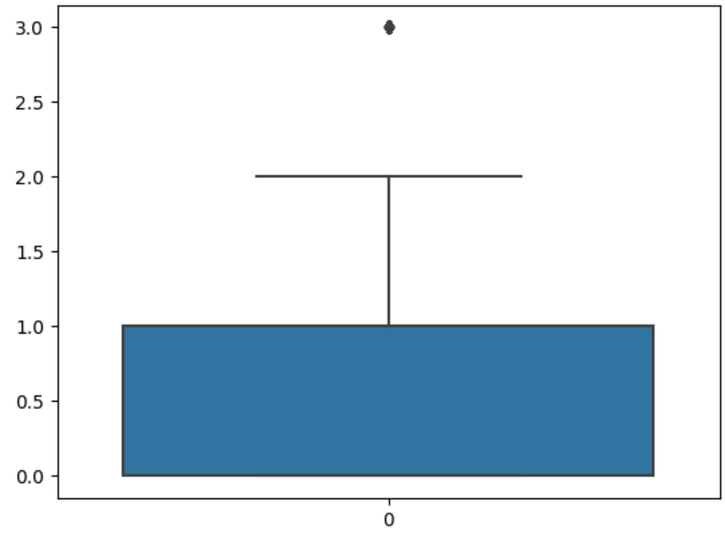
```
sns.boxplot(df["StandardHours"])
```

```
<Axes: >
```



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

```
<Axes: >
```



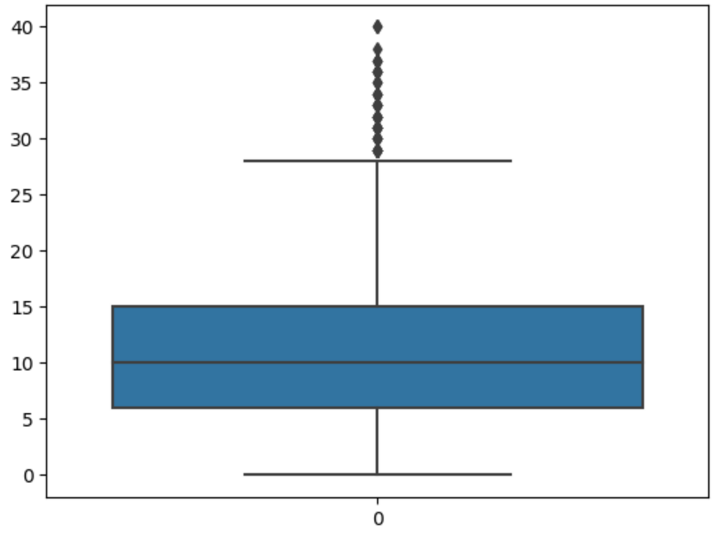
```
Q1 = df.StockOptionLevel.quantile(0.25)
Q3 = df.StockOptionLevel.quantile(.75)
IQR = Q3-Q1
upper_lt = Q3 + 1.5*IQR
lower_lt = Q1 - 1.5*IQR

median = df["StockOptionLevel"].median()

df['StockOptionLevel'] = np.where(df['StockOptionLevel']>upper_lt, median, df['StockOptionLevel'])
df['StockOptionLevel'] = np.where(df['StockOptionLevel']<lower_lt, median, df['StockOptionLevel'])
```

```
sns.boxplot(df["StockOptionLevel"], color = 'red')
```

```
<Axes: >
```

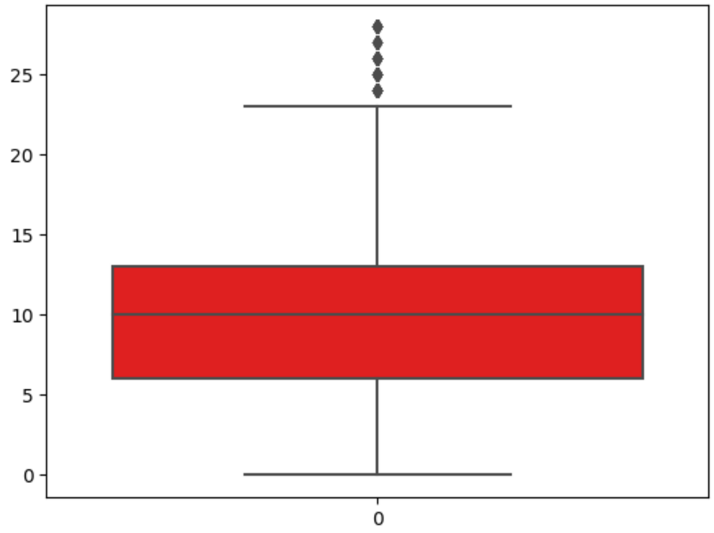
```
sns.boxplot(df["TotalWorkingYears"])
```

```
<Axes: >
```



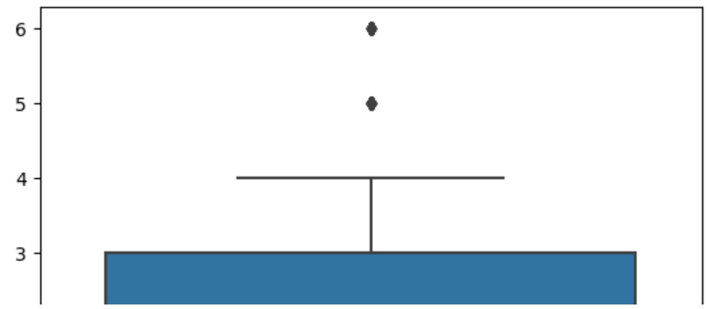
```
Q1 = df.TotalWorkingYears.quantile(0.25)
Q3 = df.TotalWorkingYears.quantile(.75)
IQR = Q3-Q1
upper_lt = Q3 + 1.5*IQR
lower_lt = Q1 - 1.5*IQR
median = df["TotalWorkingYears"].median()

df['TotalWorkingYears'] = np.where(df['TotalWorkingYears']>upper_lt, median, df['TotalWorkingYears'])
df['TotalWorkingYears'] = np.where(df['TotalWorkingYears']<lower_lt, median, df['TotalWorkingYears'])


# after outlier treatment
sns.boxplot(df["TotalWorkingYears"], color = 'red')
```

```
<Axes: >
```



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

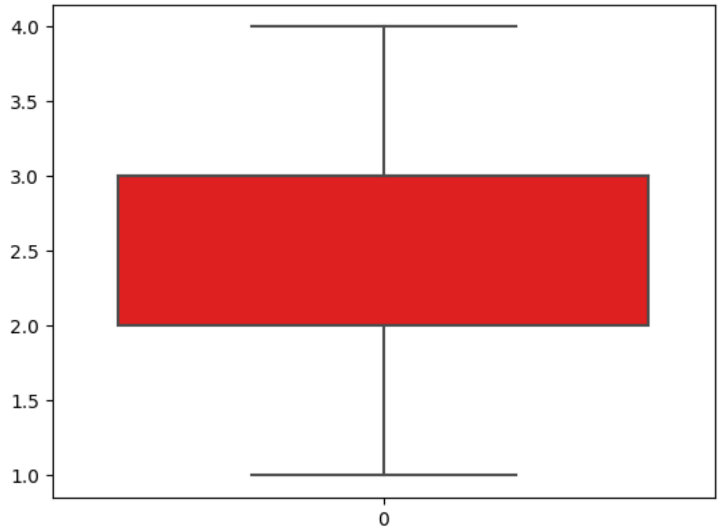
```
<Axes: >
```



```
Q1 = df.TrainingTimesLastYear.quantile(0.25)
Q3 = df.TrainingTimesLastYear.quantile(.75)
IQR = Q3-Q1
upper_lt = Q3 + 1.5*IQR
lower_lt = Q1 - 1.5*IQR
median = df["TrainingTimesLastYear"].median()

df['TrainingTimesLastYear'] = np.where(df['TrainingTimesLastYear']>upper_lt, median, df['TrainingTimesLastYear'])
df['TrainingTimesLastYear'] = np.where(df['TrainingTimesLastYear']<lower_lt, median, df['TrainingTimesLastYear'])
```

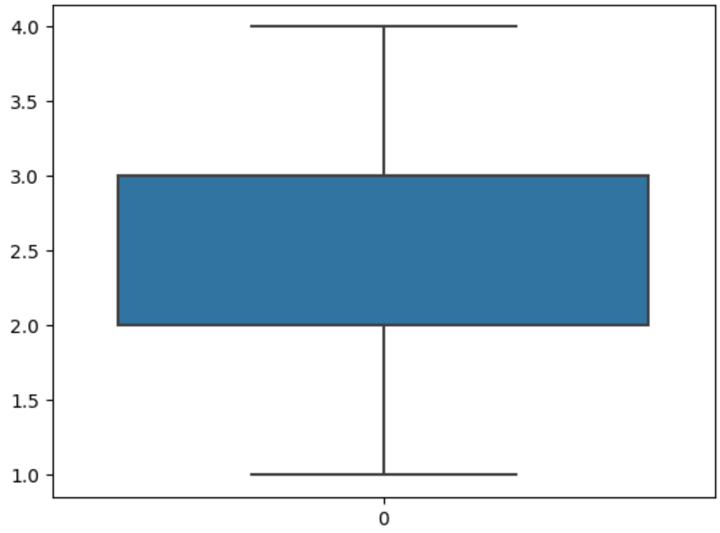
```
# after outlier treatment
sns.boxplot(df["TrainingTimesLastYear"], color = 'red')
```

```
<Axes: >
```



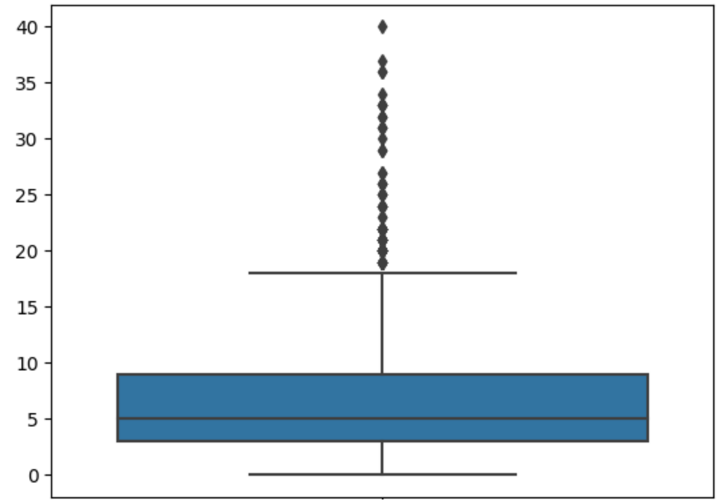
```
sns.boxplot(df["WorkLifeBalance"])
```

```
<Axes: >
```



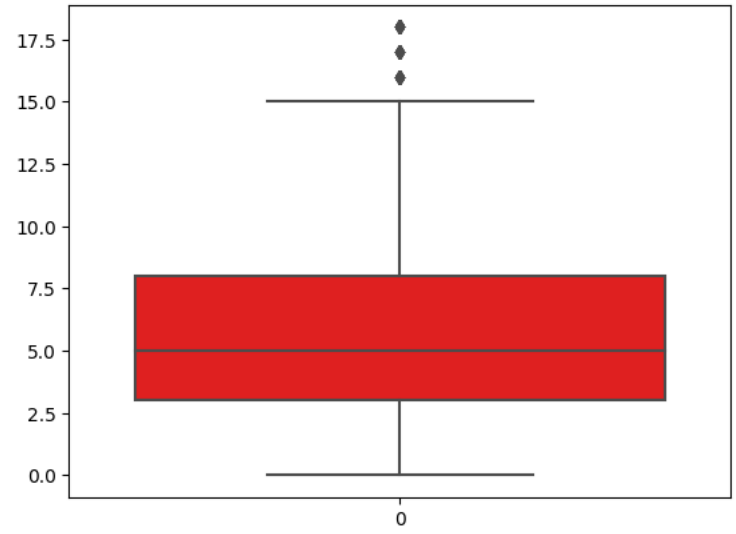
```
sns.boxplot(df["YearsAtCompany"])
```

```
<Axes: >
```



```
Q1 = df.YearsAtCompany.quantile(0.25)
Q3 = df.YearsAtCompany.quantile(.75)
IQR = Q3-Q1
upper_lt = Q3 + 1.5*IQR
lower_lt = Q1 - 1.5*IQR
median = df["YearsAtCompany"].median()

df['YearsAtCompany'] = np.where(df['YearsAtCompany']>upper_lt, median, df['YearsAtCompany'])
df['YearsAtCompany'] = np.where(df['YearsAtCompany']<lower_lt, median, df['YearsAtCompany'])


# after outlier treatment
sns.boxplot(df["YearsAtCompany"],  color = 'red')
```

```
<Axes: >
```



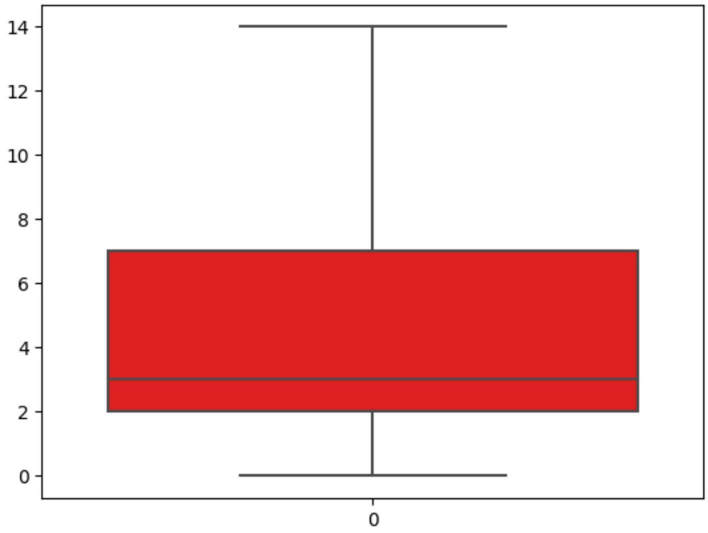
```
sns.boxplot(df["YearsInCurrentRole"])
```

```
<Axes: >
```

```
Q1 = df.YearsInCurrentRole.quantile(0.25)
Q3 = df.YearsInCurrentRole.quantile(.75)
IQR = Q3-Q1
upper_lt = Q3 + 1.5*IQR
lower_lt = Q1 - 1.5*IQR
median = df["YearsInCurrentRole"].median()

df['YearsInCurrentRole'] = np.where(df['YearsInCurrentRole']>upper_lt, median, df['YearsInCurrentRole'])
df['YearsInCurrentRole'] = np.where(df['YearsInCurrentRole']<lower_lt, median, df['YearsInCurrentRole'])
```

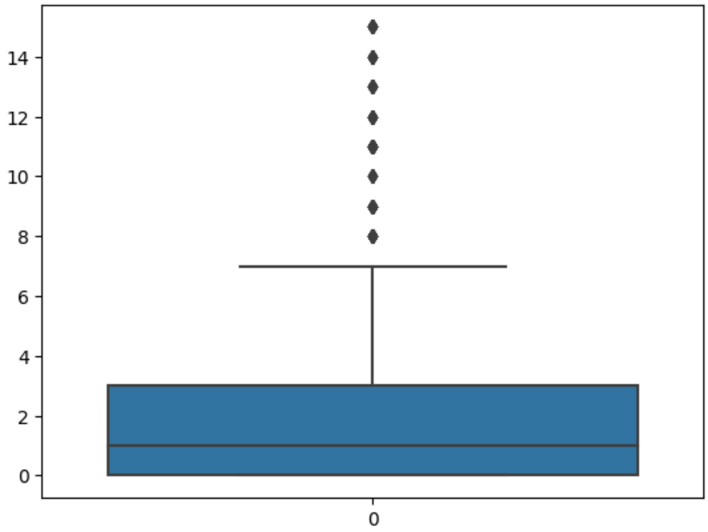
```
# after outlier treatment
sns.boxplot(df["YearsInCurrentRole"],  color = 'red')
```

```
<Axes: >
```



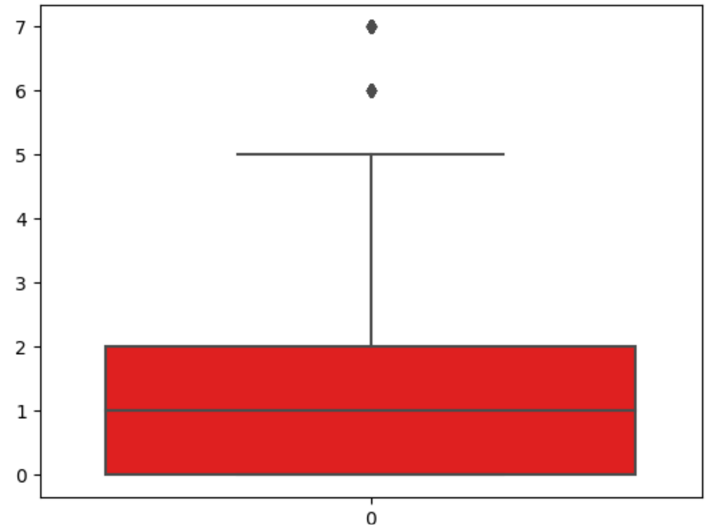
```
sns.boxplot(df["YearsSinceLastPromotion"])
```

```
<Axes: >
```



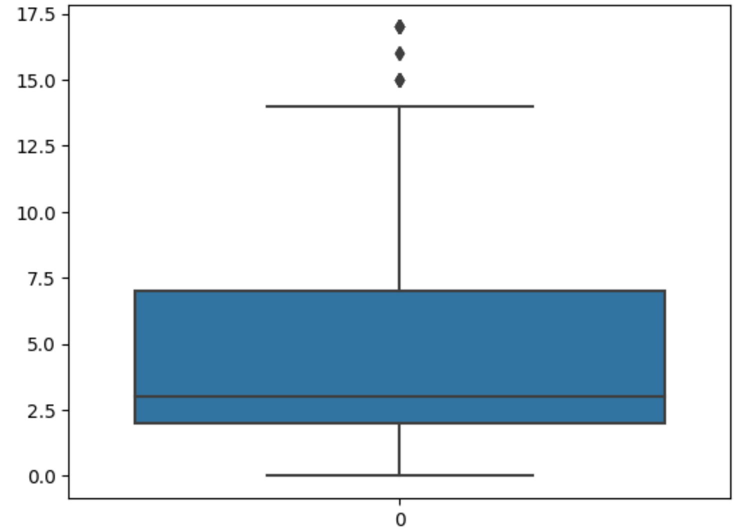
```
Q1 = df.YearsSinceLastPromotion.quantile(0.25)
Q3 = df.YearsSinceLastPromotion.quantile(.75)
IQR = Q3-Q1
upper_lt = Q3 + 1.5*IQR
lower_lt = Q1 - 1.5*IQR
median = df["YearsSinceLastPromotion"].median()

df['YearsSinceLastPromotion'] = np.where(df['YearsSinceLastPromotion']>upper_lt, median, df['YearsSinceLastPromotion'])
df['YearsSinceLastPromotion'] = np.where(df['YearsSinceLastPromotion']<lower_lt, median, df['YearsSinceLastPromotion'])


# after outlier treatment
sns.boxplot(df["YearsSinceLastPromotion"],  color = 'red')
```

```
<Axes: >
```



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

```
<Axes: >
```



```
Q1 = df.YearsWithCurrManager.quantile(0.25)
Q3 = df.YearsWithCurrManager.quantile(.75)
IQR = Q3-Q1
upper_lt = Q3 + 1.5*IQR
lower_lt = Q1 - 1.5*IQR
median = df["YearsWithCurrManager"].median()

df['YearsWithCurrManager'] = np.where(df['YearsWithCurrManager']>upper_lt, median, df['YearsWithCurrManager'])
df['YearsWithCurrManager'] = np.where(df['YearsWithCurrManager']<lower_lt, median, df['YearsWithCurrManager'])


# after outlier treatment
sns.boxplot(df["YearsWithCurrManager"],  color = 'red')
```

```
<Axes: >
```

Splitting Dependent and Independent Variable

```
# @title Splitting Dependent and Independent Variable

#X = df.iloc[:, 2:]
X = df.drop(['Attrition'], axis=1)
X
```

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

1470 rows × 34 columns

```
type(X)
```

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

```
Y = df.iloc[:, 1]
Y
```

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

```
type(Y)
```

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

Label Encoding

```
# @title Label Encoding

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

```
Y = le.fit_transform(Y)
```

```
Y
```

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

```
type(Y)
```

```
numpy.ndarray
```

```
Y = pd.Series(Y, name = 'Attrition')
```

```
Y
```

```
0       1
1       0
2       1
3       0
4       0
       ..
1465    0
1466    0
1467    0
1468    0
1469    0
Name: Attrition, Length: 1470, dtype: int64
```

```
type(Y)
```

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

```
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.OverTime = le.fit_transform(X.OverTime)
```

```
X.EducationField = le.fit_transform(X.EducationField)
```

```
X.Over18 = le.fit_transform(X.Over18)
```

```
X
```

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

1470 rows × 34 columns

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

```
# applying fit transform on all columns with 1 command
X_Scaled = pd.DataFrame(ms.fit_transform(X), columns = X.columns)
```

```
X_Scaled
```

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

1470 rows × 34 columns

```
#Splitting dataset into train and test

from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X_Scaled, Y, test_size=0.2, random_state = 0)
```

```
print(X_train.shape, X_test.shape, Y_train.shape, Y_test.shape)
```

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

MODEL BUILDING

1) Logistic Regression

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
```

```
lr.fit(X_train, Y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
pred = lr.predict(X_test)
```

```
pred
```

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

```
Y_test
```

```
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: int64
```

EVALUATION OF LOGISTIC REGRESSION MODEL

```
accuracy = accuracy_score(Y_test, pred)
accuracy
```

```
0.8775510204081632
```

```
confusion_matrix(Y_test, pred)
```

```
array([[240,   5],
       [ 31,  18]])
```

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

col_0	0	1
Attrition		
0	240	5
1	31	18

```
print(classification_report(Y_test, pred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.98      0.93       245
           1       0.78      0.37      0.50        49

    accuracy                           0.88       294
   macro avg       0.83      0.67      0.72       294
weighted avg       0.87      0.88      0.86       294
```

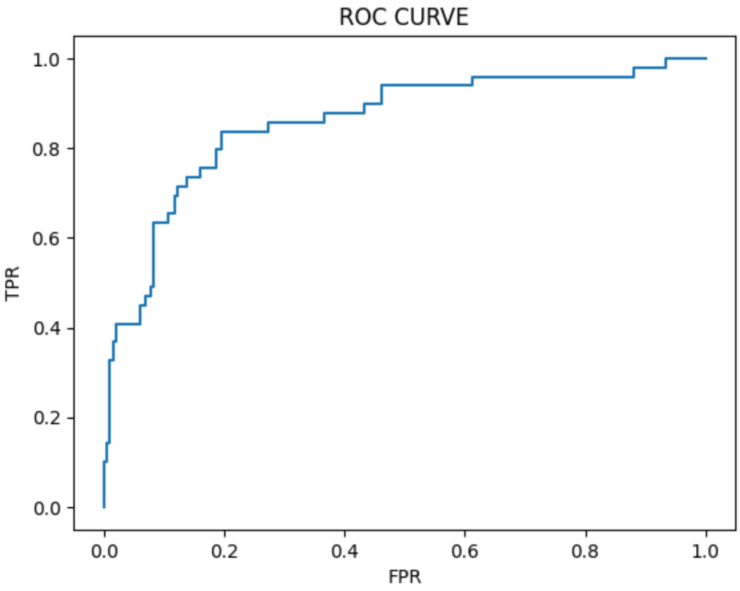
```
prob = lr.predict_proba(X_test)[:,1]
```

```
prob
```

```
       0.05676869, 0.30527903, 0.06380828, 0.00595047, 0.31380025,
       0.10429547, 0.33245617, 0.01998986, 0.70947895, 0.24794078,
       0.0360448 , 0.11540785, 0.19099221, 0.06516867, 0.1790507 ,
       0.2589084 , 0.02432868, 0.05030312, 0.07689715, 0.56886403,
       0.23296053, 0.06290349, 0.05389915, 0.74340672, 0.07925537,
       0.01583639, 0.02864397, 0.07861393, 0.35128791, 0.13789068,
       0.05708495, 0.03507929, 0.09451264, 0.05595278, 0.03671325,
       0.035616  , 0.02273025, 0.02963269, 0.01479639, 0.04009473,
       0.48767355, 0.19876799, 0.00436051, 0.74220658, 0.4680317 ,
       0.17045322, 0.52988098, 0.13832449, 0.26216984, 0.62155628,
       0.21414729, 0.02622294, 0.30058166, 0.04111191, 0.16207279,
       0.03370787, 0.27528827, 0.07107923, 0.05182189, 0.54261956,
       0.04264961, 0.23487739, 0.14701782, 0.07479887, 0.11537307,
       0.02300648, 0.21190048, 0.06772688, 0.10704887, 0.06176002,
       0.1003125 , 0.04184002, 0.18974364, 0.20453703, 0.03407722,
       0.00715255, 0.01622941, 0.16602592, 0.02548361, 0.04608612,
       0.08087786, 0.06041798, 0.02650985, 0.03486111, 0.10643701,
       0.35496811, 0.11770772, 0.19535174, 0.24275894, 0.01508851,
       0.22523028, 0.36675156, 0.29655802, 0.06393108, 0.05870711,
       0.35811197, 0.7698731 , 0.29150179, 0.01667642, 0.10890206,
       0.03754651, 0.03470175, 0.21009494, 0.03998842, 0.10919276,
       0.10808936, 0.0493002 , 0.03383773, 0.1677363 , 0.06762935,
       0.05743164, 0.05088314, 0.14647426, 0.00688177, 0.01362541,
       0.17135655, 0.04057648, 0.06784162, 0.85751252, 0.02736216,
       0.01298357, 0.00962036, 0.11610702, 0.1516965 , 0.08454238,
       0.02209819, 0.25728446, 0.58330047, 0.3984843 , 0.09482722,
       0.4400553 , 0.56922942, 0.2215613 , 0.05595997, 0.14503804,
       0.12458319, 0.07372571, 0.10975582, 0.07534216, 0.17660223,
       0.06435814, 0.11181529, 0.04075243, 0.20228697, 0.07980074,
       0.02351601, 0.01621434, 0.07857288, 0.03167707, 0.01177251,
       0.26684397, 0.00708518, 0.16607682, 0.83355576, 0.13668737,
       0.27016504, 0.13154952, 0.11162142, 0.05424089, 0.00684904,
       0.03639676, 0.09558948, 0.13291639, 0.09346684, 0.01258016,
       0.1199055 , 0.10625927, 0.05831006, 0.09153553, 0.08750728,
       0.03025245, 0.146812  , 0.01068085, 0.79556632, 0.03514599,
       0.03962226, 0.38848045, 0.06037585, 0.77262447, 0.13738206,
       0.43953659, 0.53967335, 0.17529952, 0.05286558, 0.04592524,
       0.14052106, 0.05549132, 0.01089639, 0.36059135, 0.05715995,
       0.17629379, 0.1424216 , 0.69568084, 0.05367088, 0.20697085,
       0.02818173, 0.46066677, 0.00915616, 0.14115345, 0.03003976,
       0.07539821, 0.11656754, 0.05036671, 0.09009807, 0.2495464 ,
       0.02533313, 0.02676806, 0.10527759, 0.0255894 , 0.15653797,
       0.16187907, 0.22097983, 0.79363802, 0.07603148, 0.50428861,
       0.01324733, 0.10075963, 0.24069048, 0.34126423, 0.04303388,
       0.0123262 , 0.3484587 , 0.04632451, 0.02116938, 0.1376797 ,
       0.4586222 , 0.20989952, 0.01129264, 0.07275431, 0.0242586 ,
       0.21843379, 0.29278726, 0.04439659, 0.17762422, 0.09326981,
       0.04458596, 0.44716043, 0.35302322, 0.0421056 , 0.15808294,
       0.34792218, 0.44110432, 0.8631254 , 0.0376125 , 0.22628   ,
       0.10334198, 0.02781656, 0.68815644, 0.17545869, 0.31002285,
       0.45018771, 0.02486529, 0.27695696, 0.06369293, 0.06756454,
       0.17135499, 0.00604792, 0.24771961, 0.57153276, 0.0912968 ,
       0.11724733, 0.01644254, 0.23538889, 0.05308161, 0.02923285,
       0.0311262 , 0.07946052, 0.33957449, 0.13195021, 0.18304628,
       0.29781204, 0.01860209, 0.14674178, 0.09711585, 0.11045661,
       0.26327301, 0.01357499, 0.19532227, 0.01099183, 0.01629828,
       0.20759601, 0.83454478, 0.04483892, 0.23713331])
```

ROC - AUC CURVE FOR LOGISTIC REGRESSION MODEL

```
# @title ROC - AUC CURVE FOR LOGISTIC REGRESSION MODEL
fpr, tpr, thresholds = roc_curve(Y_test, prob)


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



2) Decision Tree Classifier

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


dtc.fit(X_train, Y_train)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

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

```
pred
```

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

```
Y_test
```

```
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: int64
```

Evaluation of Decison Tree Classifier model

```
# @title Evaluation of Decison Tree Classifier model
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, roc_auc_score, roc_curve
```

```
# Accuracy Score
accuracy = accuracy_score(Y_test, pred)
accuracy
```

```
0.7312925170068028
```

```
confusion_matrix(Y_test, pred)
```

```
array([[201,  44],
       [ 35,  14]])
```

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

col_0	0	1
Attrition		
0	201	44
1	35	14

```
report = classification_report(Y_test, pred)
print(report)
```

```
              precision    recall  f1-score   support

           0       0.85      0.82      0.84       245
           1       0.24      0.29      0.26        49

    accuracy                           0.73       294
   macro avg       0.55      0.55      0.55       294
weighted avg       0.75      0.73      0.74       294
```

ROC - AUC CURVE FOR DECISION TREE CLASSIFIER MODEL

```
# @title ROC - AUC CURVE FOR DECISION TREE CLASSIFIER MODEL
```

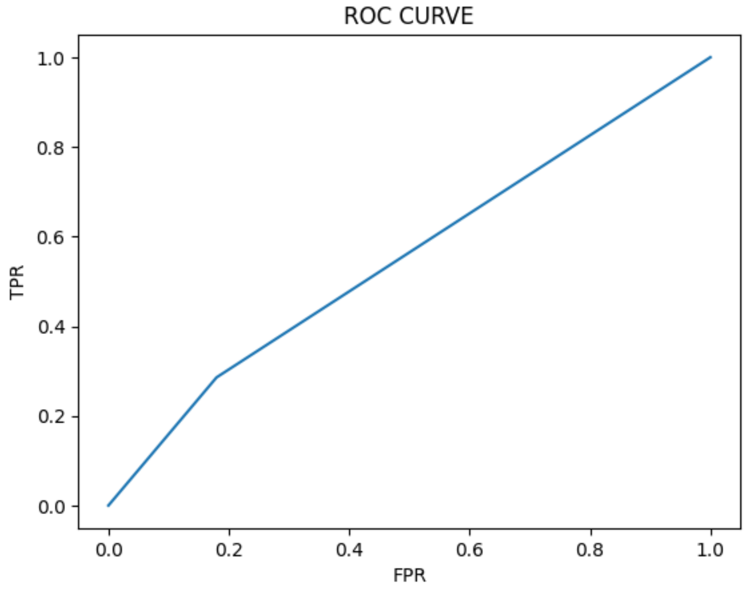
```
prob = dtc.predict_proba(X_test)[:,1]
```

```
prob
```

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

```
# roc_curve
fpr, tpr, thresholds = roc_curve(Y_test, prob)
```

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

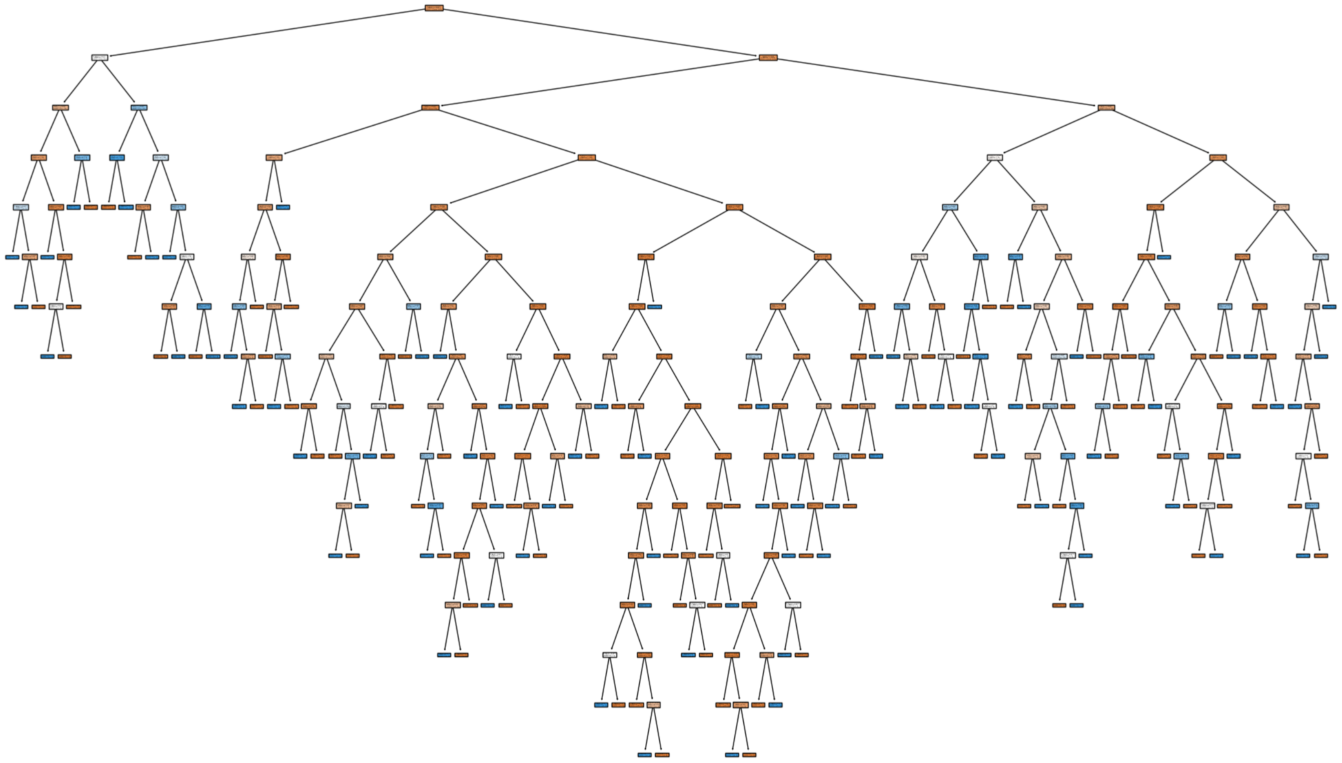


HYPER PARAMETER TUNING

```
from sklearn import tree
```

```
plt.figure(figsize=(25,15))
tree.plot_tree(dtc, filled=True)
```

```
[Text(0.3224826388888889, 0.96875, 'x[27] <= 0.054\ngini = 0.269\nsamples = 1176\nvalue = [988, 188]'),
 Text(0.0718954248366013, 0.90625, 'x[16] <= 0.75\ngini = 0.5\nsamples = 78\nvalue = [39, 39]'),
 Text(0.042483660130718956, 0.84375, 'x[4] <= 0.554\ngini = 0.426\nsamples = 39\nvalue = [27, 12]'),
 Text(0.026143790849673203, 0.78125, 'x[15] <= 0.167\ngini = 0.312\nsamples = 31\nvalue = [25, 6]'),
 Text(0.013071895424836602, 0.71875, 'x[17] <= 0.057\ngini = 0.49\nsamples = 7\nvalue = [3, 4]'),
 Text(0.006535947712418301, 0.65625, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.0196078431372549, 0.65625, 'x[22] <= 0.036\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),
 Text(0.013071895424836602, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.026143790849673203, 0.59375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.0392156862745098, 0.71875, 'x[19] <= 0.062\ngini = 0.153\nsamples = 24\nvalue = [22, 2]'),
 Text(0.032679738562091505, 0.65625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.0457516339869281, 0.65625, 'x[9] <= 0.167\ngini = 0.083\nsamples = 23\nvalue = [22, 1]'),
 Text(0.0392156862745098, 0.59375, 'x[1] <= 0.75\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.032679738562091505, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.0457516339869281, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.05228758169934641, 0.59375, 'gini = 0.0\nsamples = 21\nvalue = [21, 0]'),
 Text(0.058823529411764705, 0.78125, 'x[22] <= 0.679\ngini = 0.375\nsamples = 8\nvalue = [2, 6]'),
 Text(0.05228758169934641, 0.71875, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.06535947712418301, 0.71875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.10130718954248366, 0.84375, 'x[11] <= 0.364\ngini = 0.426\nsamples = 39\nvalue = [12, 27]'),
 Text(0.08496732026143791, 0.78125, 'x[29] <= 0.167\ngini = 0.133\nsamples = 14\nvalue = [1, 13]'),
 Text(0.0784313725490196, 0.71875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.0915032679738562, 0.71875, 'gini = 0.0\nsamples = 13\nvalue = [0, 13]'),
 Text(0.11764705882352941, 0.78125, 'x[8] <= 0.105\ngini = 0.493\nsamples = 25\nvalue = [11, 14]'),
 Text(0.10457516339869281, 0.71875, 'x[22] <= 0.464\ngini = 0.278\nsamples = 6\nvalue = [5, 1]'),
 Text(0.09803921568627451, 0.65625, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.1111111111111111, 0.65625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.13071895424836602, 0.71875, 'x[15] <= 0.5\ngini = 0.432\nsamples = 19\nvalue = [6, 13]'),
 Text(0.12418300653594772, 0.65625, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(0.13725490196078433, 0.65625, 'x[6] <= 0.4\ngini = 0.5\nsamples = 12\nvalue = [6, 6]'),
 Text(0.12418300653594772, 0.59375, 'x[28] <= 0.833\ngini = 0.278\nsamples = 6\nvalue = [5, 1]'),
 Text(0.11764705882352941, 0.53125, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.13071895424836602, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.1503267973856209, 0.59375, 'x[8] <= 0.249\ngini = 0.278\nsamples = 6\nvalue = [1, 5]'),
 Text(0.1437908496732026, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.1568627450980392, 0.53125, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(0.5730698529411765, 0.90625, 'x[21] <= 0.5\ngini = 0.235\nsamples = 1098\nvalue = [949, 149]'),
 Text(0.319750816993464, 0.84375, 'x[29] <= 0.167\ngini = 0.162\nsamples = 798\nvalue = [727, 71]'),
 Text(0.20261437908496732, 0.78125, 'x[28] <= 0.833\ngini = 0.38\nsamples = 47\nvalue = [35, 12]'),
 Text(0.19607843137254902, 0.71875, 'x[12] <= 0.5\ngini = 0.325\nsamples = 44\nvalue = [35, 9]'),
 Text(0.1830065359477124, 0.65625, 'x[2] <= 0.747\ngini = 0.498\nsamples = 15\nvalue = [8, 7]'),
 Text(0.17647058823529413, 0.59375, 'x[4] <= 0.446\ngini = 0.42\nsamples = 10\nvalue = [3, 7]'),
 Text(0.16993464052287582, 0.53125, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.1830065359477124, 0.53125, 'x[0] <= 0.298\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),
 Text(0.17647058823529413, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.1895424836601307, 0.46875, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.1895424836601307, 0.59375, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.20915032679738563, 0.65625, 'x[0] <= 0.333\ngini = 0.128\nsamples = 29\nvalue = [27, 2]'),
 Text(0.20261437908496732, 0.59375, 'x[17] <= 0.125\ngini = 0.408\nsamples = 7\nvalue = [5, 2]'),
 Text(0.19607843137254902, 0.53125, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.20915032679738563, 0.53125, 'x[13] <= 0.125\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.20261437908496732, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.21568627450980393, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.21568627450980393, 0.59375, 'gini = 0.0\nsamples = 22\nvalue = [22, 0]'),
 Text(0.20915032679738563, 0.71875, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.4368872549019608, 0.78125, 'x[30] <= 0.25\ngini = 0.145\nsamples = 751\nvalue = [692, 59]'),
 Text(0.3263888888888889, 0.71875, 'x[9] <= 0.167\ngini = 0.218\nsamples = 257\nvalue = [225, 32]'),
 Text(0.28594771241830064, 0.65625, 'x[33] <= 0.179\ngini = 0.355\nsamples = 65\nvalue = [50, 15]'),
 Text(0.2647058823529412, 0.59375, 'x[33] <= 0.036\ngini = 0.303\nsamples = 59\nvalue = [48, 11]'),
 Text(0.24183006535947713, 0.53125, 'x[12] <= 0.5\ngini = 0.463\nsamples = 22\nvalue = [14, 8]'),
 Text(0.22875816993464052, 0.46875, 'x[11] <= 0.179\ngini = 0.198\nsamples = 9\nvalue = [8, 1]'),
 Text(0.2222222222222222, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.23529411764705882, 0.40625, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(0.2549019607843137, 0.46875, 'x[11] <= 0.4\ngini = 0.497\nsamples = 13\nvalue = [6, 7]'),
 Text(0.24836601307189543, 0.40625, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.26143790849673204, 0.40625, 'x[4] <= 0.286\ngini = 0.346\nsamples = 9\nvalue = [2, 7]'),
 Text(0.2549019607843137, 0.34375, 'x[19] <= 0.312\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.24836601307189543, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.26143790849673204, 0.28125, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.2679738562091503, 0.34375, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.2875816993464052, 0.53125, 'x[15] <= 0.167\ngini = 0.149\nsamples = 37\nvalue = [34, 3]'),
 Text(0.28104575163398693, 0.46875, 'x[30] <= 0.194\ngini = 0.5\nsamples = 6\nvalue = [3, 3]'),
 Text(0.27450980392156865, 0.40625, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.2875816993464052, 0.40625, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.29411764705882354, 0.46875, 'gini = 0.0\nsamples = 31\nvalue = [31, 0]'),
 Text(0.30718954248366015, 0.59375, 'x[8] <= 0.065\ngini = 0.444\nsamples = 6\nvalue = [2, 4]'),
 Text(0.3006535947712418, 0.53125, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.3137254901960784, 0.53125, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.3668300653594771, 0.65625, 'x[0] <= 0.321\ngini = 0.161\nsamples = 192\nvalue = [175, 17]'),
 Text(0.3333333333333333, 0.59375, 'x[6] <= 0.1\ngini = 0.294\nsamples = 67\nvalue = [55, 12]'),
 Text(0.32679738562091504, 0.53125, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.33986928104575165, 0.53125, 'x[29] <= 0.5\ngini = 0.26\nsamples = 65\nvalue = [55, 10]'),
 Text(0.3235294117647059, 0.46875, 'x[11] <= 0.679\ngini = 0.469\nsamples = 16\nvalue = [10, 6]'),
 Text(0.31699346405228757, 0.40625, 'x[6] <= 0.4\ngini = 0.444\nsamples = 9\nvalue = [3, 6]'),
 Text(0.3104575163398693, 0.34375, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.3235294117647059, 0.34375, 'x[22] <= 0.464\ngini = 0.245\nsamples = 7\nvalue = [1, 6]'),
 Text(0.31699346405228757, 0.28125, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.3300653594771242, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.3300653594771242, 0.40625, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(0.3562091503267974, 0.46875, 'x[2] <= 0.037\ngini = 0.15\nsamples = 49\nvalue = [45, 4]'),
 Text(0.34967320261437906, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3627450980392157, 0.40625, 'x[2] <= 0.938\ngini = 0.117\nsamples = 48\nvalue = [45, 3]'),
 Text(0.3562091503267974, 0.34375, 'x[5] <= 0.875\ngini = 0.081\nsamples = 47\nvalue = [45, 2]'),
 Text(0.3431372549019608, 0.28125, 'x[12] <= 0.167\ngini = 0.043\nsamples = 45\nvalue = [44, 1]'),
 Text(0.3366013071895425, 0.21875, 'x[22] <= 0.214\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.3300653594771242, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3431372549019608, 0.15625, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.34967320261437906, 0.21875, 'gini = 0.0\nsamples = 42\nvalue = [42, 0]'),
 Text(0.369281045751634, 0.28125, 'x[2] <= 0.337\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.3627450980392157, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3758169934640523, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.369281045751634, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.40032679738562094, 0.59375, 'x[8] <= 0.022\ngini = 0.077\nsamples = 125\nvalue = [120, 5]'),
 Text(0.3823529411764706, 0.53125, 'x[2] <= 0.578\ngini = 0.5\nsamples = 4\nvalue = [2, 2]'),
 Text(0.3758169934640523, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.3888888888888889, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.41830065359477125, 0.53125, 'x[18] <= 0.968\ngini = 0.048\nsamples = 121\nvalue = [118, 3]'),
 Text(0.4019607843137255, 0.46875, 'x[2] <= 0.98\ngini = 0.033\nsamples = 118\nvalue = [116, 2]'),
 Text(0.3888888888888889, 0.40625, 'x[14] <= 0.938\ngini = 0.017\nsamples = 114\nvalue = [113, 1]'),
 Text(0.3823529411764706, 0.34375, 'gini = 0.0\nsamples = 107\nvalue = [107, 0]'),
 Text(0.3954248366013072, 0.34375, 'x[11] <= 0.193\ngini = 0.245\nsamples = 7\nvalue = [6, 1]'),
 Text(0.3888888888888889, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.4019607843137255, 0.28125, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.4150326797385621, 0.40625, 'x[0] <= 0.405\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),
 Text(0.4084967320261438, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.4215686274509804, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.434640522875817, 0.46875, 'x[30] <= 0.083\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.42810457516339867, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.4411764705882353, 0.40625, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.5473856209150327, 0.71875, 'x[17] <= 0.299\ngini = 0.103\nsamples = 494\nvalue = [467, 27]'),
 Text(0.4812091503267974, 0.65625, 'x[33] <= 0.964\ngini = 0.06\nsamples = 291\nvalue = [282, 9]'),
 Text(0.47467320261437906, 0.59375, 'x[8] <= 0.016\ngini = 0.054\nsamples = 290\nvalue = [282, 8]'),
 Text(0.4542483660130719, 0.53125, 'x[4] <= 0.411\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.4477124183006536, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.46078431372549017, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.4950980392156863, 0.53125, 'x[17] <= 0.056\ngini = 0.048\nsamples = 287\nvalue = [280, 7]'),
 Text(0.4738562091503268, 0.46875, 'x[17] <= 0.054\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),
 Text(0.4673202614379085, 0.40625, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.4803921568627451, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5163398692810458, 0.46875, 'x[15] <= 0.5\ngini = 0.042\nsamples = 282\nvalue = [276, 6]'),
 Text(0.4934640522875817, 0.40625, 'x[2] <= 0.34\ngini = 0.092\nsamples = 104\nvalue = [99, 5]'),
 Text(0.4803921568627451, 0.34375, 'x[2] <= 0.334\ngini = 0.219\nsamples = 32\nvalue = [28, 4]'),
 Text(0.4738562091503268, 0.28125, 'x[19] <= 0.812\ngini = 0.175\nsamples = 31\nvalue = [28, 3]'),
 Text(0.4673202614379085, 0.21875, 'x[31] <= 0.107\ngini = 0.124\nsamples = 30\nvalue = [28, 2]'),
 Text(0.4542483660130719, 0.15625, 'x[2] <= 0.192\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.4477124183006536, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.46078431372549017, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.4803921568627451, 0.15625, 'x[4] <= 0.946\ngini = 0.069\nsamples = 28\nvalue = [27, 1]'),
 Text(0.4738562091503268, 0.09375, 'gini = 0.0\nsamples = 25\nvalue = [25, 0]'),
 Text(0.4869281045751634, 0.09375, 'x[1] <= 0.75\ngini = 0.444\nsamples = 3\nvalue = [2, 1]'),
 Text(0.4803921568627451, 0.03125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.4934640522875817, 0.03125, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.4803921568627451, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.4869281045751634, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5065359477124183, 0.34375, 'x[26] <= 0.75\ngini = 0.027\nsamples = 72\nvalue = [71, 1]'),
 Text(0.5, 0.28125, 'gini = 0.0\nsamples = 64\nvalue = [64, 0]'),
 Text(0.5130718954248366, 0.28125, 'x[2] <= 0.528\ngini = 0.219\nsamples = 8\nvalue = [7, 1]'),
 Text(0.5065359477124183, 0.21875, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.5196078431372549, 0.21875, 'x[14] <= 0.625\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.5130718954248366, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5261437908496732, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5392156862745098, 0.40625, 'x[33] <= 0.107\ngini = 0.011\nsamples = 178\nvalue = [177, 1]'),
 Text(0.5326797385620915, 0.34375, 'x[30] <= 0.528\ngini = 0.133\nsamples = 14\nvalue = [13, 1]'),
 Text(0.5261437908496732, 0.28125, 'gini = 0.0\nsamples = 12\nvalue = [12, 0]'),
 Text(0.5392156862745098, 0.28125, 'x[10] <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.5326797385620915, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.545751633986928, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.545751633986928, 0.34375, 'gini = 0.0\nsamples = 164\nvalue = [164, 0]'),
 Text(0.4877450980392157, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.613562091503268, 0.65625, 'x[17] <= 0.5\ngini = 0.162\nsamples = 203\nvalue = [185, 18]'),
 Text(0.5800653594771242, 0.59375, 'x[2] <= 0.033\ngini = 0.275\nsamples = 97\nvalue = [81, 16]'),
 Text(0.5620915032679739, 0.53125, 'x[17] <= 0.346\ngini = 0.48\nsamples = 5\nvalue = [2, 3]'),
 Text(0.5555555555555556, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.5686274509803921, 0.46875, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.5980392156862745, 0.53125, 'x[0] <= 0.607\ngini = 0.243\nsamples = 92\nvalue = [79, 13]'),
 Text(0.5816993464052288, 0.46875, 'x[18] <= 0.952\ngini = 0.157\nsamples = 70\nvalue = [64, 6]'),
 Text(0.5751633986928104, 0.40625, 'x[17] <= 0.3\ngini = 0.134\nsamples = 69\nvalue = [64, 5]'),
 Text(0.5686274509803921, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5816993464052288, 0.34375, 'x[11] <= 0.971\ngini = 0.111\nsamples = 68\nvalue = [64, 4]'),
 Text(0.5751633986928104, 0.28125, 'x[31] <= 0.786\ngini = 0.086\nsamples = 67\nvalue = [64, 3]'),
 Text(0.5588235294117647, 0.21875, 'x[6] <= 0.9\ngini = 0.06\nsamples = 65\nvalue = [63, 2]'),
 Text(0.545751633986928, 0.15625, 'x[28] <= 0.833\ngini = 0.032\nsamples = 61\nvalue = [60, 1]'),
 Text(0.5392156862745098, 0.09375, 'gini = 0.0\nsamples = 56\nvalue = [56, 0]'),
 Text(0.5522875816993464, 0.09375, 'x[22] <= 0.071\ngini = 0.32\nsamples = 5\nvalue = [4, 1]'),
 Text(0.545751633986928, 0.03125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5588235294117647, 0.03125, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.5718954248366013, 0.15625, 'x[8] <= 0.412\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),
 Text(0.565359477124183, 0.09375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.5784313725490197, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5915032679738562, 0.21875, 'x[31] <= 0.929\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.5849673202614379, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]')
 Text(0.5849673202614379, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5980392156862745, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5882352941176471, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5882352941176471, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6143790849673203, 0.46875, 'x[4] <= 0.268\ngini = 0.434\nsamples = 22\nvalue = [15, 7]'),
 Text(0.6013071895424836, 0.40625, 'x[19] <= 0.062\ngini = 0.231\nsamples = 15\nvalue = [13, 2]'),
 Text(0.5947712418300654, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6078431372549019, 0.34375, 'x[17] <= 0.498\ngini = 0.133\nsamples = 14\nvalue = [13, 1]'),
 Text(0.6013071895424836, 0.28125, 'gini = 0.0\nsamples = 13\nvalue = [13, 0]'),
 Text(0.6143790849673203, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6274509803921569, 0.40625, 'x[33] <= 0.571\ngini = 0.408\nsamples = 7\nvalue = [2, 5]'),
 Text(0.6209150326797386, 0.34375, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(0.6339869281045751, 0.34375, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.6470588235294118, 0.59375, 'x[31] <= 0.964\ngini = 0.037\nsamples = 106\nvalue = [104, 2]'),
 Text(0.6405228758169934, 0.53125, 'x[30] <= 0.972\ngini = 0.019\nsamples = 105\nvalue = [104, 1]'),
 Text(0.6339869281045751, 0.46875, 'gini = 0.0\nsamples = 101\nvalue = [101, 0]'),
 Text(0.6470588235294118, 0.46875, 'x[24] <= 0.833\ngini = 0.375\nsamples = 4\nvalue = [3, 1]'),
 Text(0.6405228758169934, 0.40625, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.6535947712418301, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6535947712418301, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8263888888888888, 0.84375, 'x[17] <= 0.157\ngini = 0.385\nsamples = 300\nvalue = [222, 78]'),
 Text(0.7426470588235294, 0.78125, 'x[26] <= 0.25\ngini = 0.5\nsamples = 96\nvalue = [49, 47]'),
 Text(0.7091503267973857, 0.71875, 'x[4] <= 0.161\ngini = 0.459\nsamples = 42\nvalue = [15, 27]'),
 Text(0.6862745098039216, 0.65625, 'x[8] <= 0.415\ngini = 0.499\nsamples = 23\nvalue = [12, 11]'),
 Text(0.673202614379085, 0.59375, 'x[18] <= 0.561\ngini = 0.355\nsamples = 13\nvalue = [3, 10]'),
 Text(0.6666666666666666, 0.53125, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
 Text(0.6797385620915033, 0.53125, 'x[9] <= 0.333\ngini = 0.48\nsamples = 5\nvalue = [3, 2]'),
 Text(0.673202614379085, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.6862745098039216, 0.46875, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.6993464052287581, 0.59375, 'x[14] <= 0.875\ngini = 0.18\nsamples = 10\nvalue = [9, 1]'),
 Text(0.6928104575163399, 0.53125, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(0.7058823529411765, 0.53125, 'x[15] <= 0.667\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.6993464052287581, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.7124183006535948, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.7320261437908496, 0.65625, 'x[27] <= 0.5\ngini = 0.266\nsamples = 19\nvalue = [3, 16]'),
 Text(0.7254901960784313, 0.59375, 'x[11] <= 0.2\ngini = 0.198\nsamples = 18\nvalue = [2, 16]'),
 Text(0.7189542483660131, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.7320261437908496, 0.53125, 'x[32] <= 0.929\ngini = 0.111\nsamples = 17\nvalue = [1, 16]'),
 Text(0.7254901960784313, 0.46875, 'gini = 0.0\nsamples = 15\nvalue = [0, 15]'),
 Text(0.738562091503268, 0.46875, 'x[5] <= 0.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.7320261437908496, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.7450980392156863, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.738562091503268, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.7761437908496732, 0.71875, 'x[0] <= 0.202\ngini = 0.466\nsamples = 54\nvalue = [34, 20]'),
 Text(0.7581699346405228, 0.65625, 'x[8] <= 0.164\ngini = 0.245\nsamples = 7\nvalue = [1, 6]'),
 Text(0.7516339869281046, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.7647058823529411, 0.59375, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.7941176470588235, 0.65625, 'x[2] <= 0.622\ngini = 0.418\nsamples = 47\nvalue = [33, 14]'),
 Text(0.7777777777777778, 0.59375, 'x[2] <= 0.145\ngini = 0.482\nsamples = 32\nvalue = [19, 13]'),
 Text(0.7647058823529411, 0.53125, 'x[2] <= 0.024\ngini = 0.18\nsamples = 10\nvalue = [9, 1]'),
 Text(0.7581699346405228, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.7712418300653595, 0.46875, 'gini = 0.0\nsamples = 9\nvalue = [9, 0]'),
 Text(0.7908496732026143, 0.53125, 'x[18] <= 0.87\ngini = 0.496\nsamples = 22\nvalue = [10, 12]'),
 Text(0.7843137254901961, 0.46875, 'x[8] <= 0.41\ngini = 0.465\nsamples = 19\nvalue = [7, 12]'),
 Text(0.7712418300653595, 0.40625, 'x[18] <= 0.715\ngini = 0.469\nsamples = 8\nvalue = [5, 3]'),
 Text(0.7647058823529411, 0.34375, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.7777777777777778, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.7973856209150327, 0.40625, 'x[0] <= 0.25\ngini = 0.298\nsamples = 11\nvalue = [2, 9]'),
 Text(0.7908496732026143, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.803921568627451, 0.34375, 'x[4] <= 0.018\ngini = 0.18\nsamples = 10\nvalue = [1, 9]'),
 Text(0.7973856209150327, 0.28125, 'x[11] <= 0.229\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.7908496732026143, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.803921568627451, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8104575163398693, 0.28125, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
 Text(0.7973856209150327, 0.46875, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.8104575163398693, 0.59375, 'x[11] <= 0.064\ngini = 0.124\nsamples = 15\nvalue = [14, 1]'),
 Text(0.803921568627451, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8169934640522876, 0.53125, 'gini = 0.0\nsamples = 14\nvalue = [14, 0]'),
 Text(0.9101307189542484, 0.78125, 'x[16] <= 0.75\ngini = 0.258\nsamples = 204\nvalue = [173, 31]'),
 Text(0.8627450980392157, 0.71875, 'x[17] <= 0.992\ngini = 0.138\nsamples = 147\nvalue = [136, 11]'),
 Text(0.8562091503267973, 0.65625, 'x[4] <= 0.482\ngini = 0.128\nsamples = 146\nvalue = [136, 10]'),
 Text(0.8366013071895425, 0.59375, 'x[30] <= 0.139\ngini = 0.038\nsamples = 104\nvalue = [102, 2]'),
 Text(0.8300653594771242, 0.53125, 'x[11] <= 0.193\ngini = 0.32\nsamples = 10\nvalue = [8, 2]'),
 Text(0.8235294117647058, 0.46875, 'x[13] <= 0.625\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.8169934640522876, 0.40625, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.8300653594771242, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.8366013071895425, 0.46875, 'gini = 0.0\nsamples = 7\nvalue = [7, 0]'),
 Text(0.8431372549019608, 0.53125, 'gini = 0.0\nsamples = 94\nvalue = [94, 0]'),
 Text(0.8758169934640523, 0.59375, 'x[9] <= 0.167\ngini = 0.308\nsamples = 42\nvalue = [34, 8]'),
 Text(0.8562091503267973, 0.53125, 'x[18] <= 0.194\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.8496732026143791, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.8627450980392157, 0.46875, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.8954248366013072, 0.53125, 'x[0] <= 0.393\ngini = 0.229\nsamples = 38\nvalue = [33, 5]'),
 Text(0.8758169934640523, 0.46875, 'x[1] <= 0.25\ngini = 0.5\nsamples = 6\nvalue = [3, 3]'),
 Text(0.869281045751634, 0.40625, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.8823529411764706, 0.40625, 'x[11] <= 0.643\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.8758169934640523, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.8888888888888888, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.9150326797385621, 0.46875, 'x[8] <= 0.992\ngini = 0.117\nsamples = 32\nvalue = [30, 2]'),
 Text(0.9084967320261438, 0.40625, 'x[18] <= 0.174\ngini = 0.062\nsamples = 31\nvalue = [30, 1]'),
 Text(0.9019607843137255, 0.34375, 'x[14] <= 0.688\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.8954248366013072, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.9084967320261438, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.9150326797385621, 0.34375, 'gini = 0.0\nsamples = 29\nvalue = [29, 0]'),
 Text(0.9215686274509803, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.869281045751634, 0.65625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.9575163398692811, 0.71875, 'x[14] <= 0.812\ngini = 0.456\nsamples = 57\nvalue = [37, 20]'),
 Text(0.9281045751633987, 0.65625, 'x[8] <= 0.071\ngini = 0.238\nsamples = 29\nvalue = [25, 4]'),
 Text(0.9150326797385621, 0.59375, 'x[0] <= 0.321\ngini = 0.444\nsamples = 3\nvalue = [1, 2]'),
 Text(0.9084967320261438, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.9215686274509803, 0.53125, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.9411764705882353, 0.59375, 'x[28] <= 0.167\ngini = 0.142\nsamples = 26\nvalue = [24, 2]'),
 Text(0.934640522875817, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.9477124183006536, 0.53125, 'x[17] <= 0.956\ngini = 0.077\nsamples = 25\nvalue = [24, 1]'),
 Text(0.9411764705882353, 0.46875, 'gini = 0.0\nsamples = 24\nvalue = [24, 0]'),
 Text(0.954248366013072, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.9869281045751634, 0.65625, 'x[32] <= 0.214\ngini = 0.49\nsamples = 28\nvalue = [12, 16]'),
 Text(0.9803921568627451, 0.59375, 'x[4] <= 0.804\ngini = 0.48\nsamples = 20\nvalue = [12, 8]'),
 Text(0.9738562091503268, 0.53125, 'x[30] <= 0.028\ngini = 0.415\nsamples = 17\nvalue = [12, 5]'),
 Text(0.9673202614379085, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.9803921568627451, 0.46875, 'x[24] <= 0.5\ngini = 0.32\nsamples = 15\nvalue = [12, 3]'),
 Text(0.9738562091503268, 0.40625, 'x[27] <= 0.196\ngini = 0.5\nsamples = 6\nvalue = [3, 3]'),
 Text(0.9673202614379085, 0.34375, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.9803921568627451, 0.34375, 'x[6] <= 0.7\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.9738562091503268, 0.28125, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.9869281045751634, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.9869281045751634, 0.40625, 'gini = 0.0\nsamples = 9\nvalue = [9, 0]'),
 Text(0.9869281045751634, 0.53125, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.9934640522875817, 0.59375, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]')]
```



```
from sklearn.model_selection import GridSearchCV

# search documentation of parameters for Decision Tree Classifier
parameters = {
    'criterion' : ['gini', 'entropy'],
    'splitter' : ['best', 'random'],
    'max_depth' : [1, 2, 3, 4, 5],
    'max_features' : ['auto', 'sqrt', 'log2']
}


# using GridSearchCV to figure out the best hyperparameters
grid_search = GridSearchCV(estimator = dtc, param_grid = parameters, cv=5, scoring ="accuracy")


grid_search.fit(X_train, Y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarning: `max_features='auto'` has been deprecated in
  warnings.warn(
```