Perform Data preprocessing & Model Building on Employee Attrition Dataset 1.Data Collection. Please download the dataset from https://www.kaggle.com/datasets/patelprashant/employee-attrition

2.Data Preprocessing o Import the Libraries. o Importing the dataset. o Checking for Null Values. o Data Visualization. o Outlier Detection o Splitting Dependent and Independent variables o Perform Encoding o Feature Scaling. o Splitting Data into Train and Test

3.Model Building o Import the model building Libraries o Initializing the model o Training and testing the model o Evaluation of Model & Performance metrics o Save the Model

Data Collection

Collected Data from Kaggle - Employee Attrition Dataset

Data Preprocessing

Import the Libraries

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

Importing the DataSet

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

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Educati
0	41	Yes	Travel_Rarely	1102	Sales	1	
1	49	No	Travel_Frequently	279	Research & Development	8	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	
4	27	No	Travel_Rarely	591	Research & Development	2	

5 rows × 35 columns

```
df.info()
```

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

```
df.describe()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	Empl
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1
mean	36.923810	802.485714	9.192517	2.912925	1.0	1
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	1
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1
max	60.000000	1499.000000	29.000000	5.000000	1.0	2

8 rows × 26 columns

```
df.shape
```

```
(1470, 35)
```

Checking for null values

```
df.isnull().any()
```

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

```
YearsWithCurrManager        False
dtype: bool
```

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

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

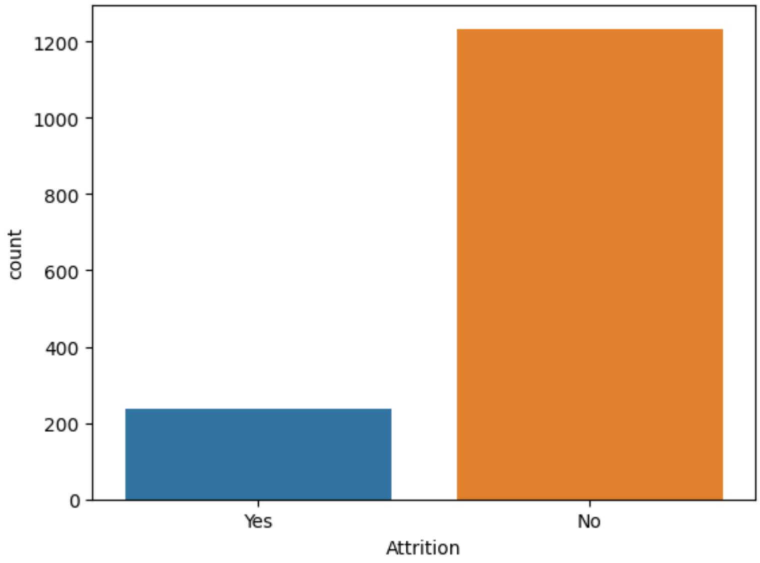
```
print("Null percentage in columns : ")
for i in df.columns:
    c=df[i].count()
    n=df[i].isnull().sum()
    print(i," : ",(n/(n+c)) * 100)
```

```
Null percentage in columns :
Age  :  0.0
Attrition  :  0.0
BusinessTravel  :  0.0
DailyRate  :  0.0
Department  :  0.0
DistanceFromHome  :  0.0
Education  :  0.0
EducationField  :  0.0
EmployeeCount  :  0.0
EmployeeNumber  :  0.0
EnvironmentSatisfaction  :  0.0
Gender  :  0.0
HourlyRate  :  0.0
JobInvolvement  :  0.0
JobLevel  :  0.0
JobRole  :  0.0
JobSatisfaction  :  0.0
MaritalStatus  :  0.0
MonthlyIncome  :  0.0
MonthlyRate  :  0.0
NumCompaniesWorked  :  0.0
Over18  :  0.0
OverTime  :  0.0
PercentSalaryHike  :  0.0
PerformanceRating  :  0.0
RelationshipSatisfaction  :  0.0
StandardHours  :  0.0
StockOptionLevel  :  0.0
TotalWorkingYears  :  0.0
TrainingTimesLastYear  :  0.0
WorkLifeBalance  :  0.0
YearsAtCompany  :  0.0
YearsInCurrentRole  :  0.0
YearsSinceLastPromotion  :  0.0
YearsWithCurrManager  :  0.0
```

Data Visualization

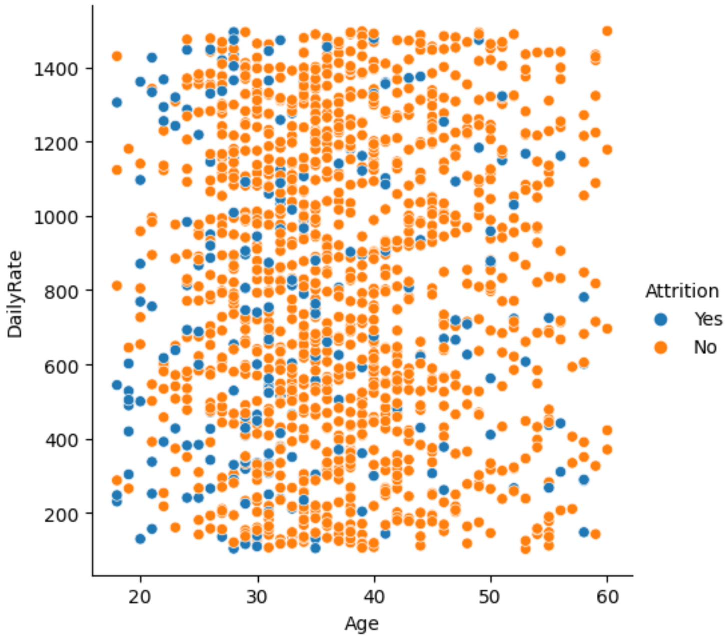
```
sns.countplot(x=df.Attrition,data=df)
```

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



```
sns.relplot(x = 'Age', y = 'DailyRate', hue = 'Attrition', data = df)
```

```
<seaborn.axisgrid.FacetGrid at 0x7eb9b111f130>
```



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

```
<ipython-input-11-2c2d81c89147>: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["TotalWorkingYears"])
<Axes: xlabel='TotalWorkingYears', ylabel='Density'>
```

```
# fig=plt.figure(figsize=(20,6))
# sns.pairplot(df)
```

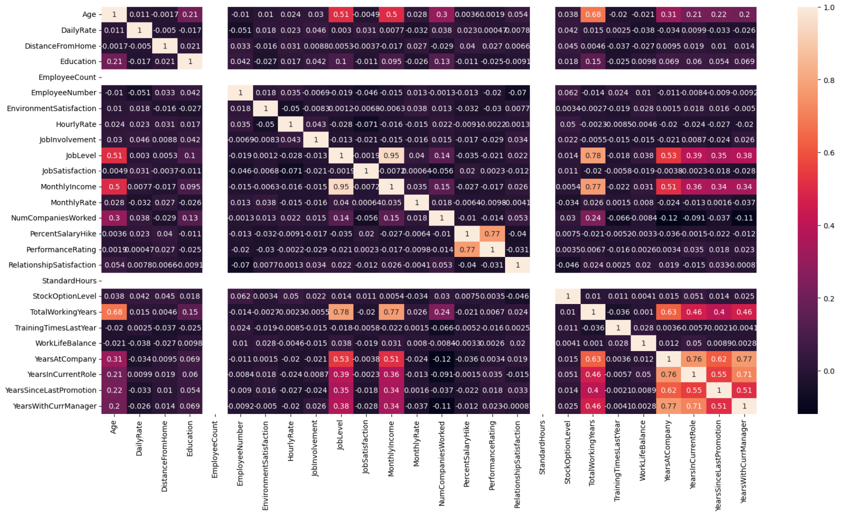
```
corr=df.corr(numeric_only=True)
corr
```

	Age	DailyRate	DistanceFromHome	Education	Employe
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	

26 rows × 26 columns

```
fig=plt.figure(figsize=(20,10))
sns.heatmap(corr,annot=True,)
```

```
<Axes: >
```



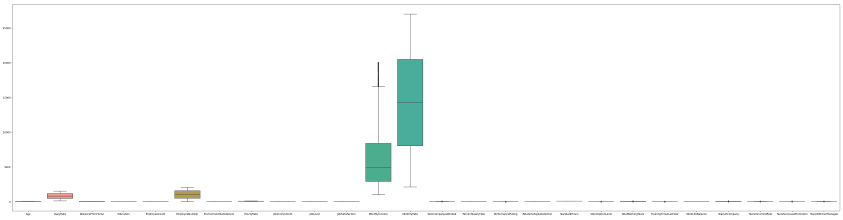
Outlier Detection

```
df.columns
```

```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate',
       'JobInvolvement', 'JobLevel', 'JobRole', 'JobSatisfaction',
       'MaritalStatus', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked',
       'Over18', 'OverTime', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

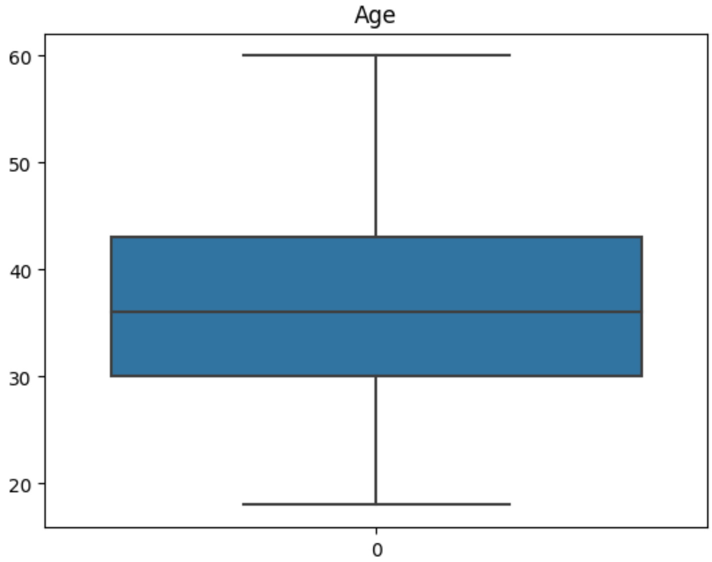
```
fig=plt.figure(figsize=(60,15))
sns.boxplot(df)
```

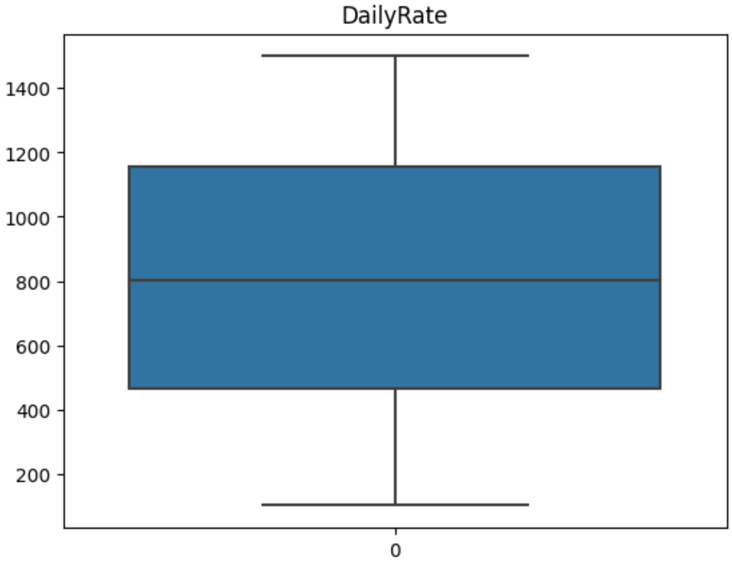
```
<Axes: >
```

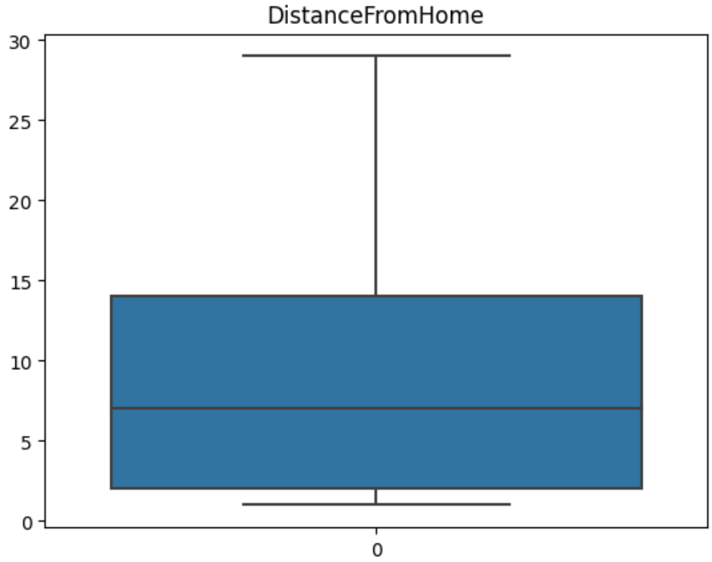


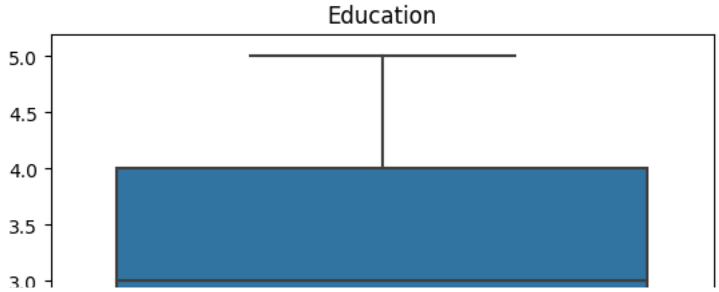
```
for i in df.columns:
    if(df[i].dtype!=object):
        plt.figure()
        plt.title(i)
        sns.boxplot(df[i])
```

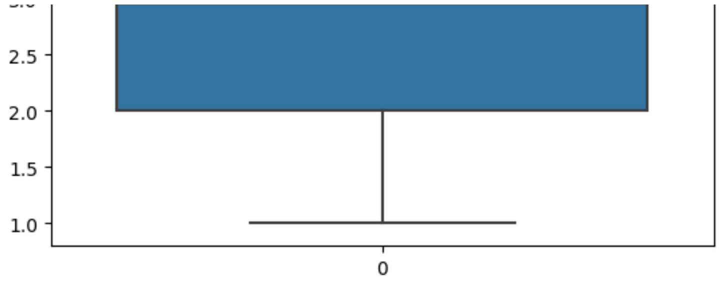
```
<ipython-input-17-d0c2f0b5eb22>:3: RuntimeWarning: More than 20 figures have been
  plt.figure()
```

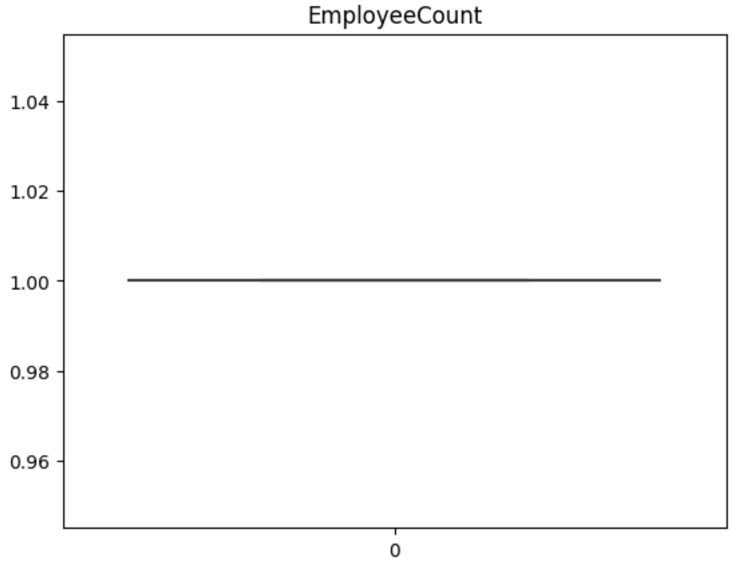


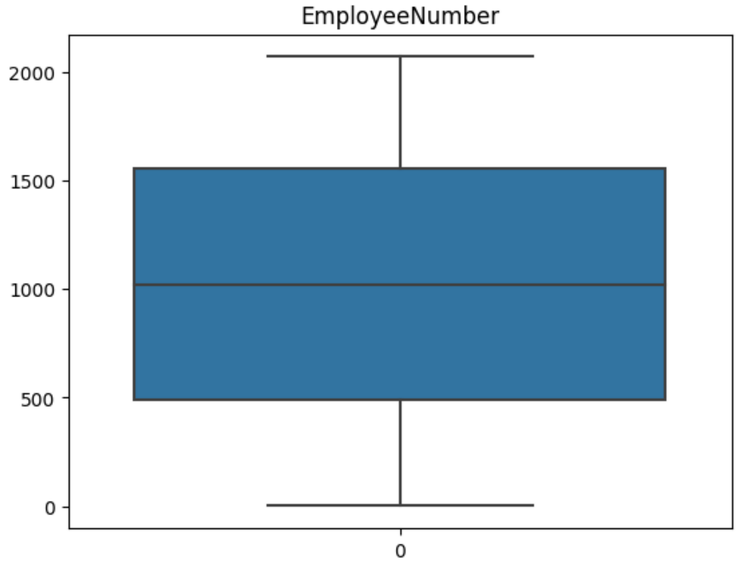


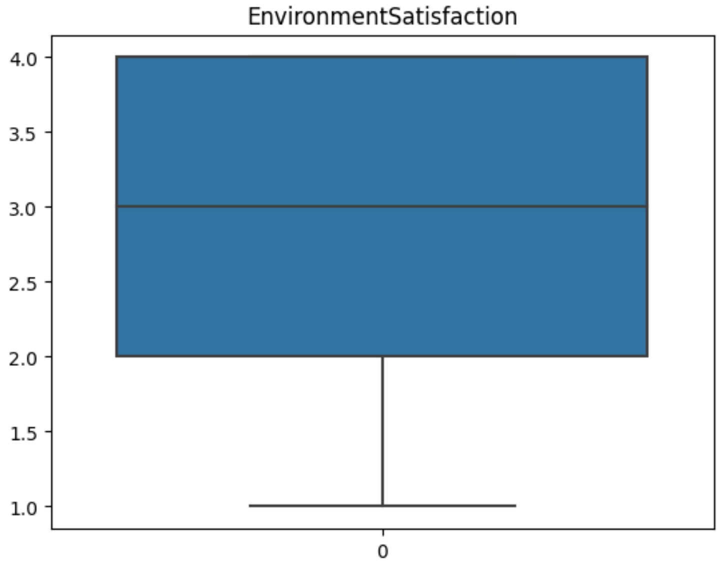


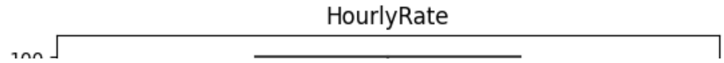


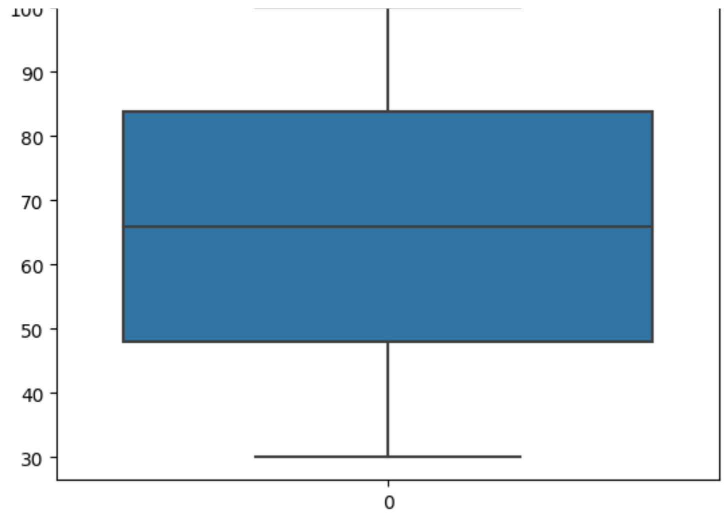



EmployeeCount

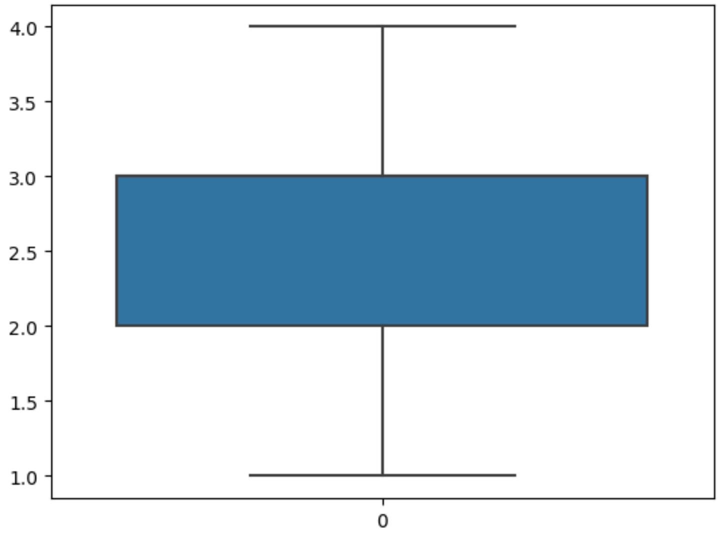

EmployeeNumber


EnvironmentSatisfaction

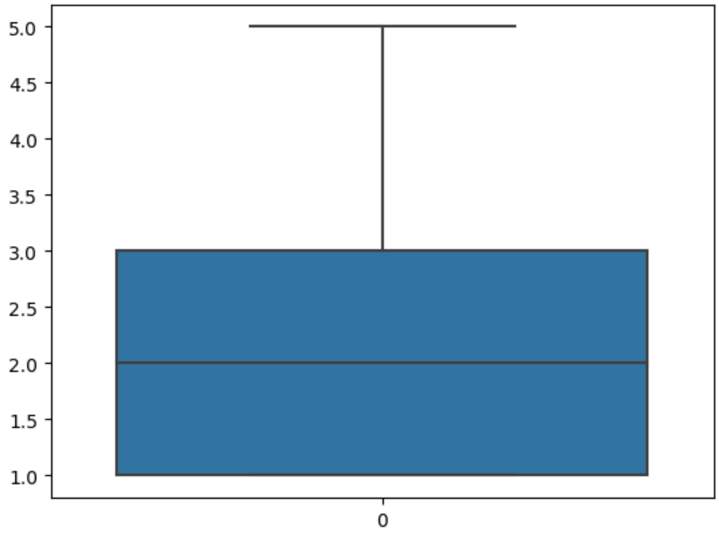

HourlyRate



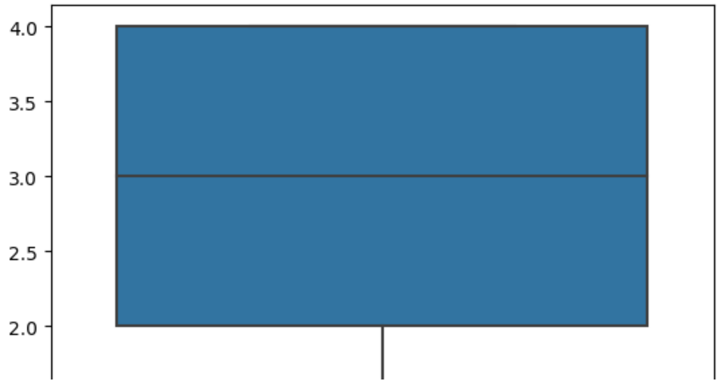
JobInvolvement

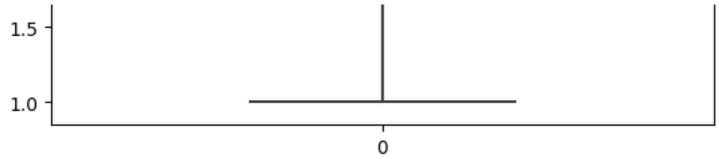


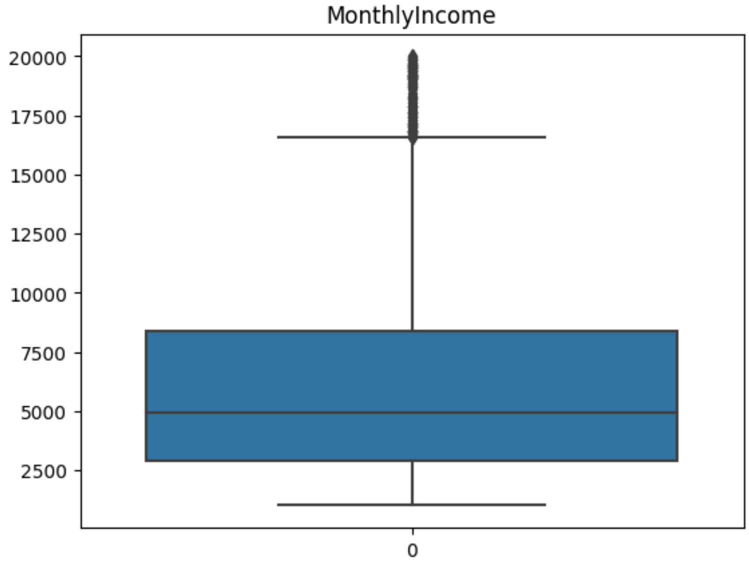
JobLevel

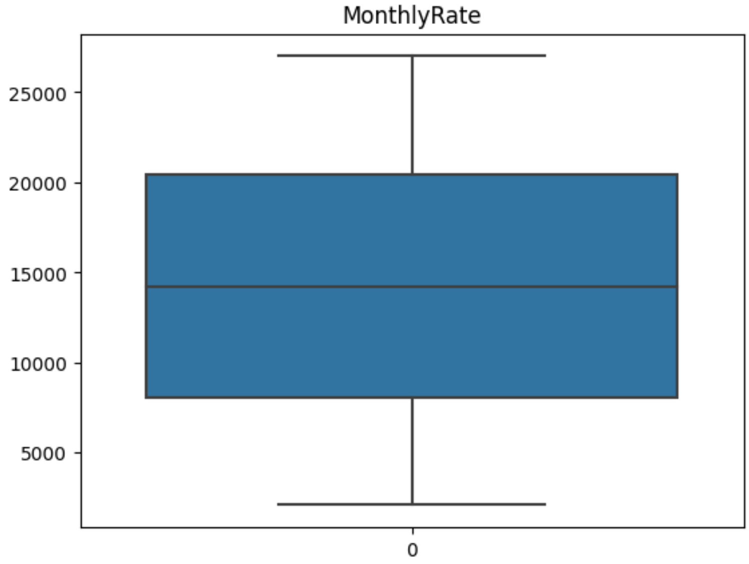


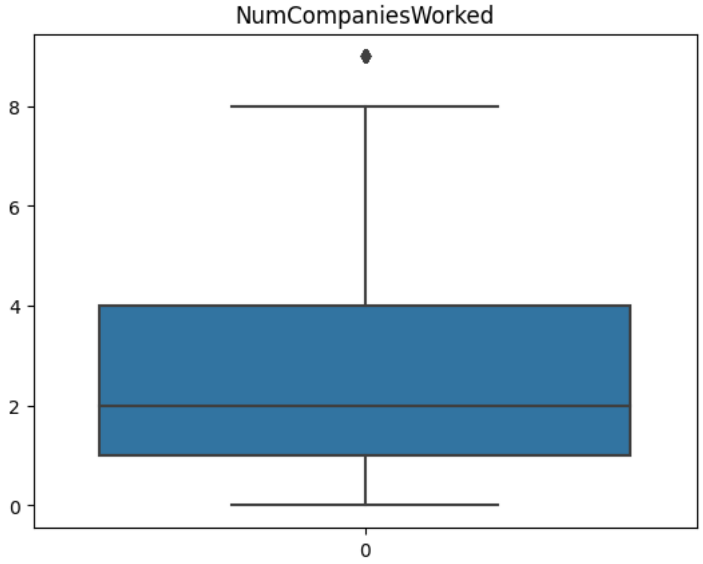
JobSatisfaction

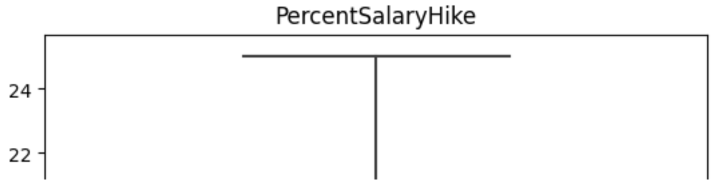


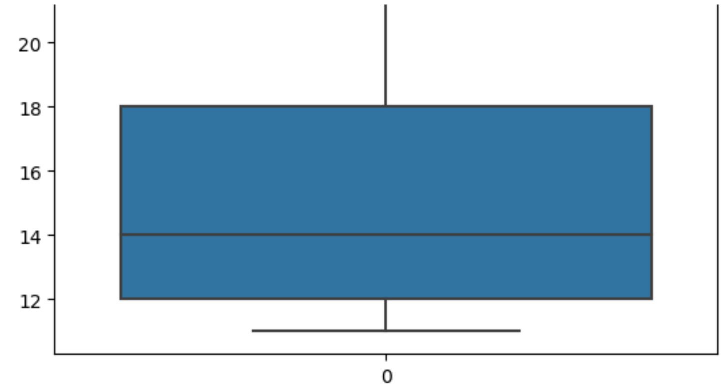

1.5
1.0
0

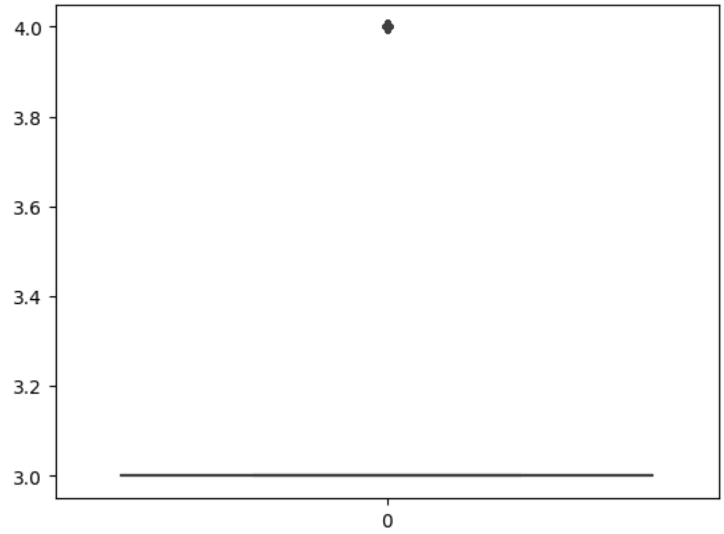

MonthlyIncome
20000
17500
15000
12500
10000
7500
5000
2500
0

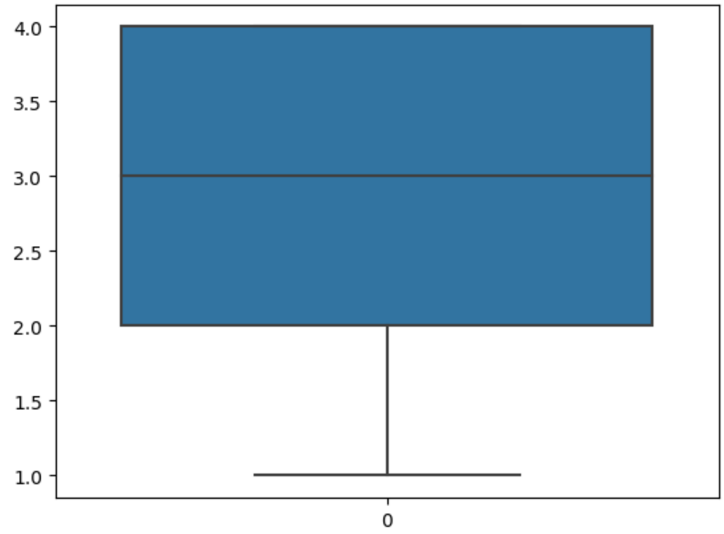

MonthlyRate
25000
20000
15000
10000
5000
0

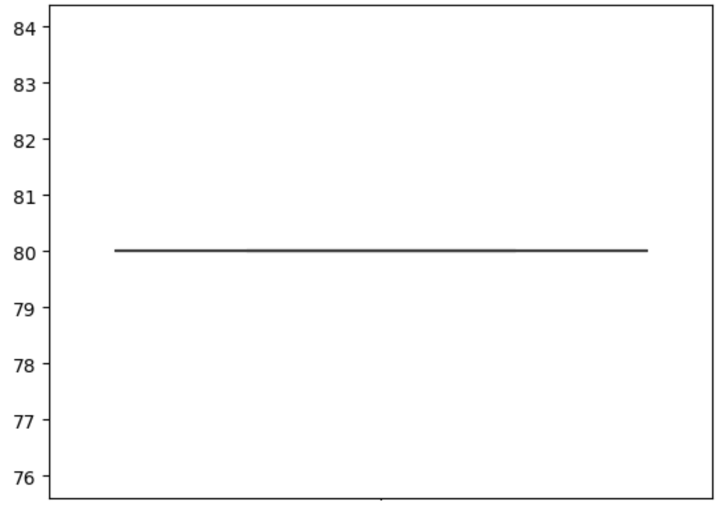

NumCompaniesWorked
8
6
4
2
0
0


PercentSalaryHike
24
22

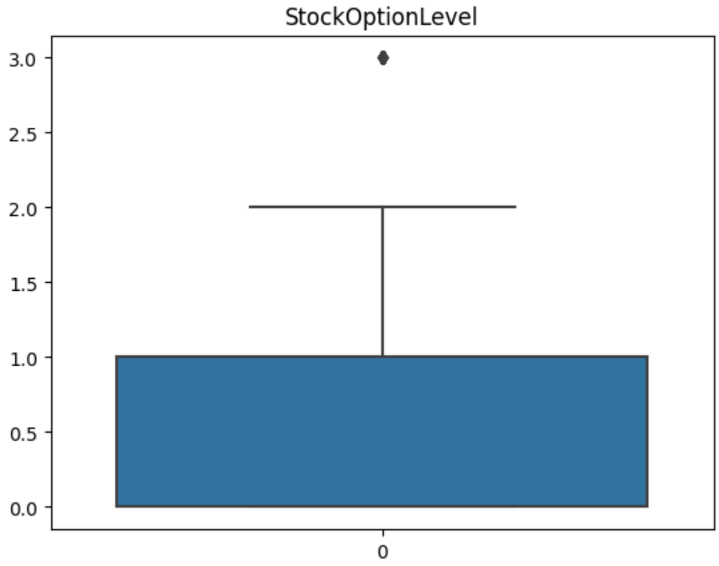


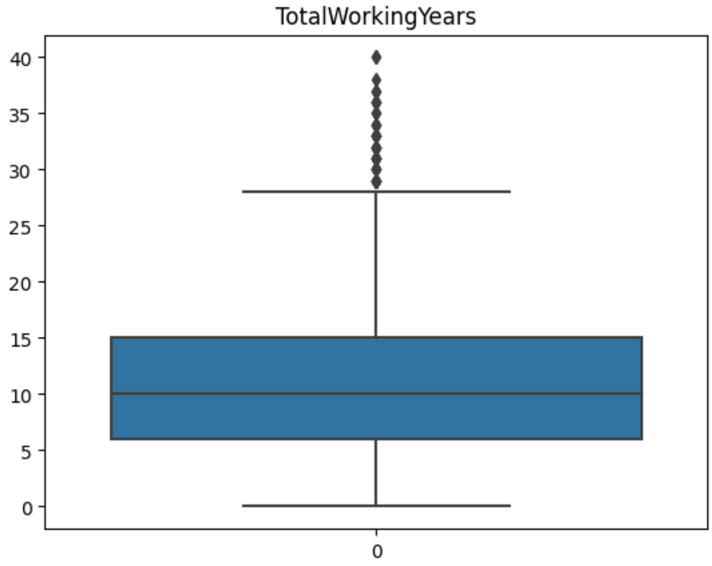

PerformanceRating

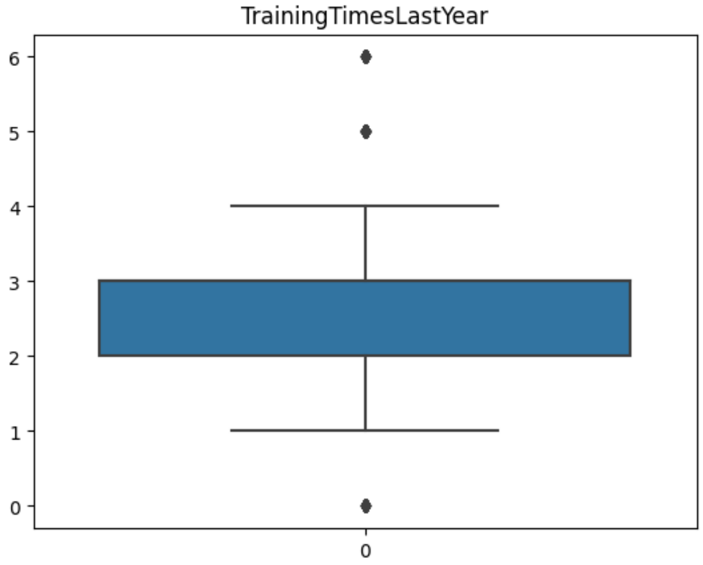

RelationshipSatisfaction

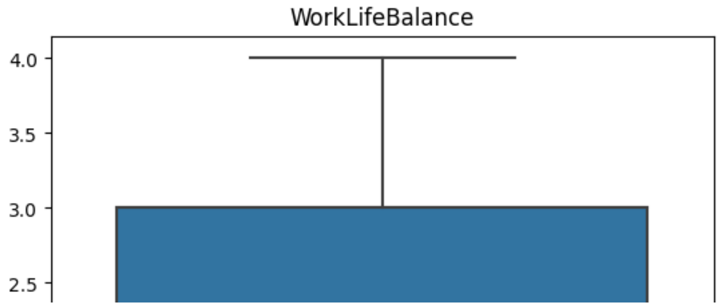

StandardHours

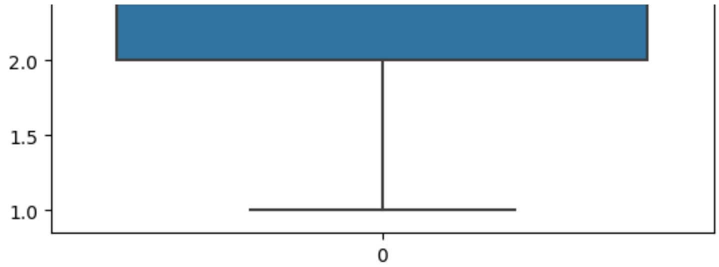
0

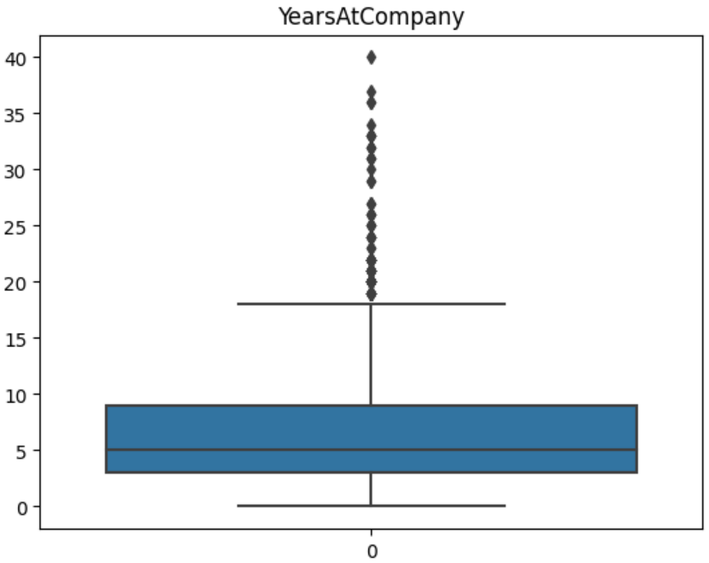

StockOptionLevel
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0

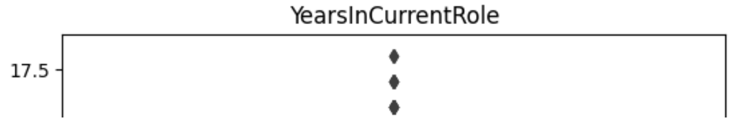

TotalWorkingYears
40
35
30
25
20
15
10
5
0
0


TrainingTimesLastYear
6
5
4
3
2
1
0
0


WorkLifeBalance
4.0
3.5
3.0
2.5



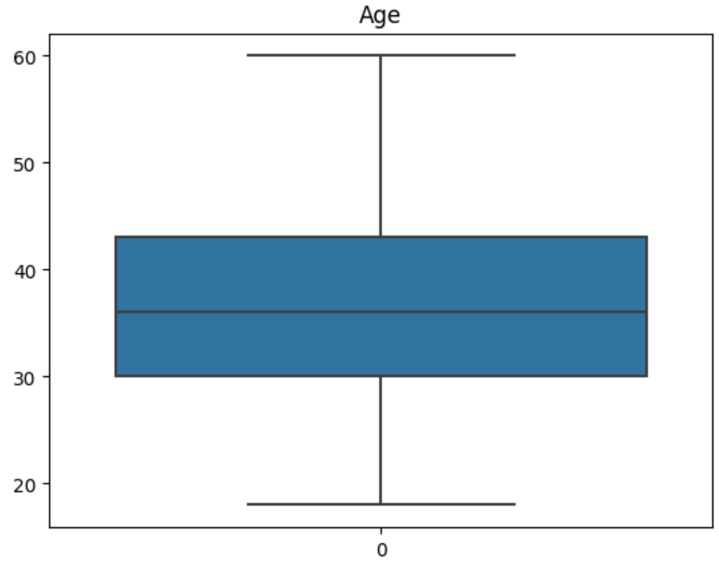

YearsAtCompany

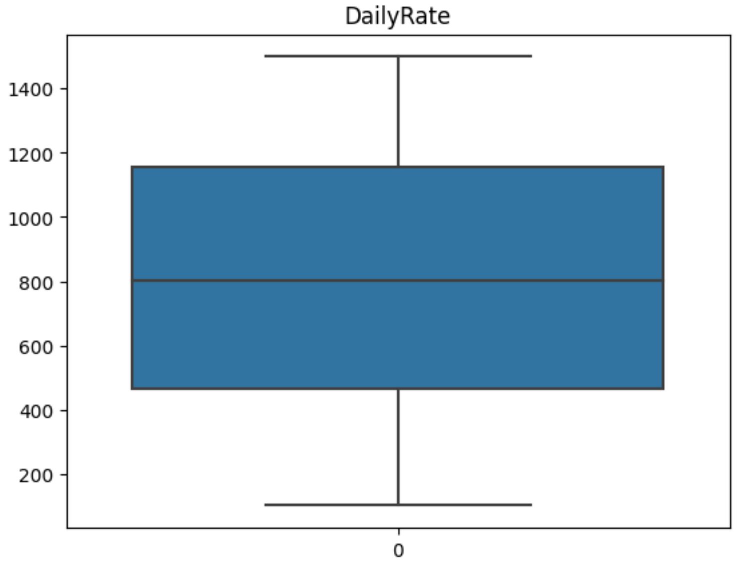

YearsInCurrentRole

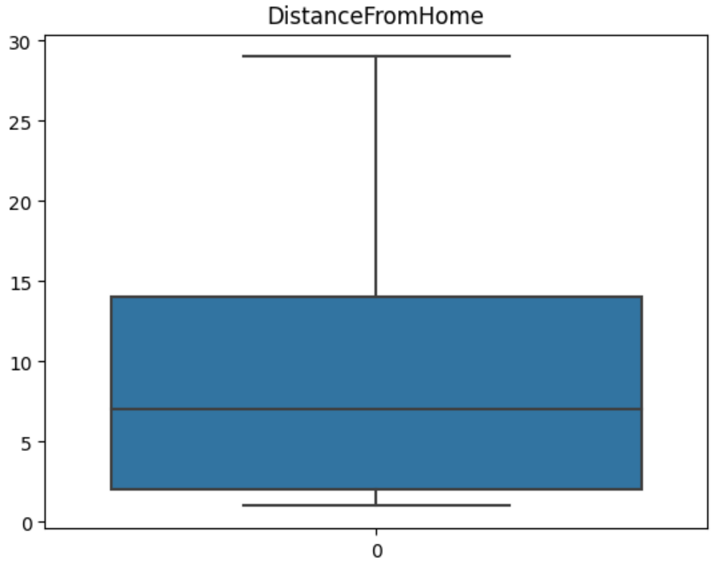
```
l=["MonthlyIncome","NumCompaniesWorked","StockOptionLevel","TotalWorkingYears","TrainingTimesLastYear","YearsAtCompany", "YearsInCurrentR
       "YearsWithCurrManager"]
for i in l:
    q3=df[i].quantile(0.75)
    q1=df[i].quantile(0.25)
    IQR=q3-q1
    upper_limit=q3+(1.5*IQR)
    lower_limit=q1-(1.5*IQR)
    df[i]=np.where(df[i]>upper_limit,df[i].median(),df[i])
```

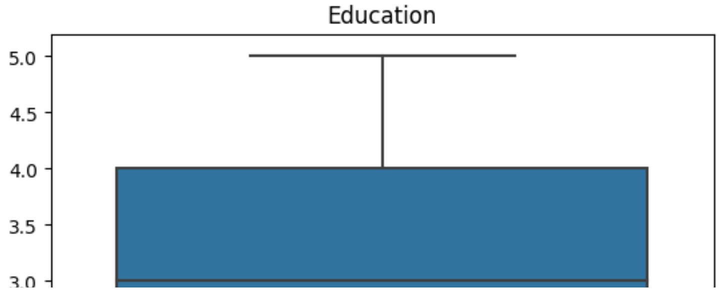
```
for i in df.columns:
    if(df[i].dtype!=object):
        plt.figure()
        plt.title(i)
        sns.boxplot(df[i])
```

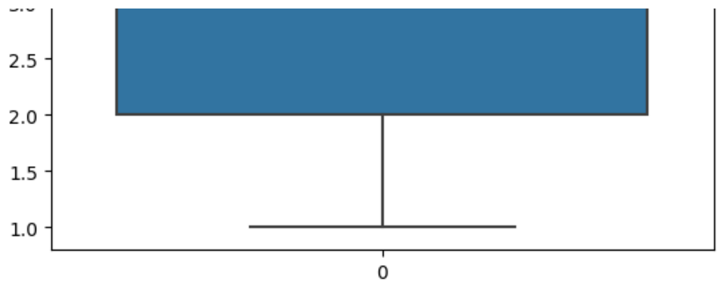
```
<ipython-input-19-d0c2f0b5eb22>:3: RuntimeWarning: More than 20 figures have been
  plt.figure()
```

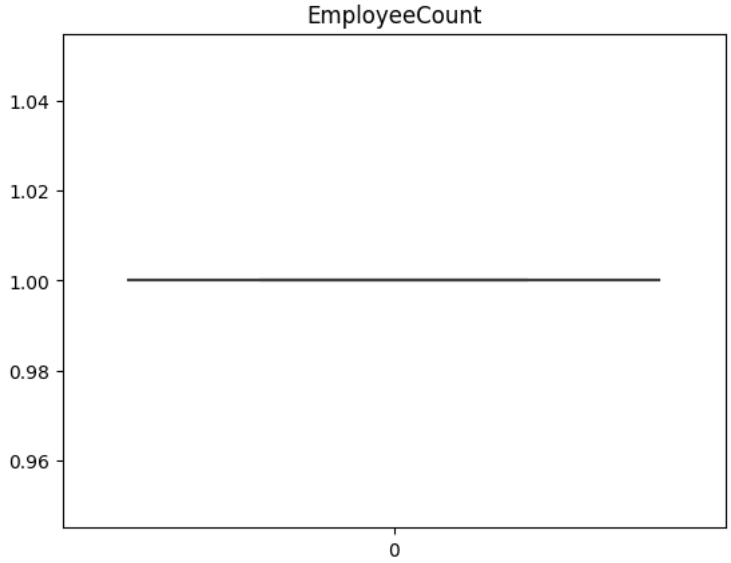


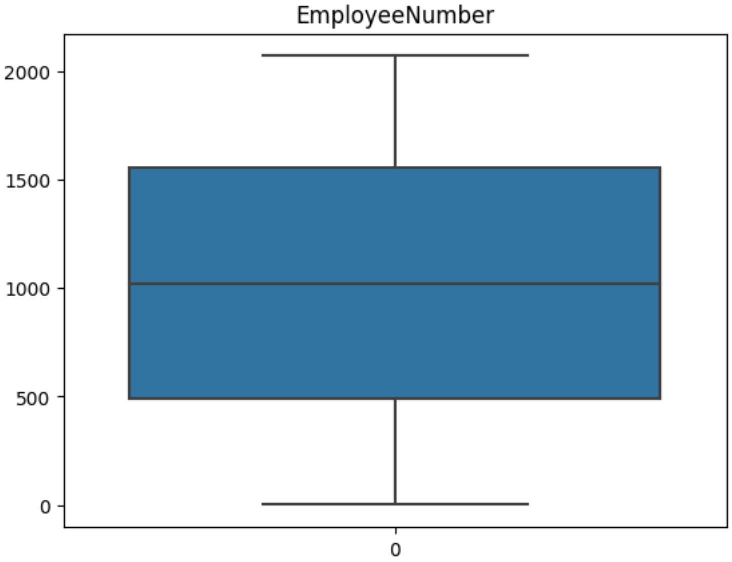


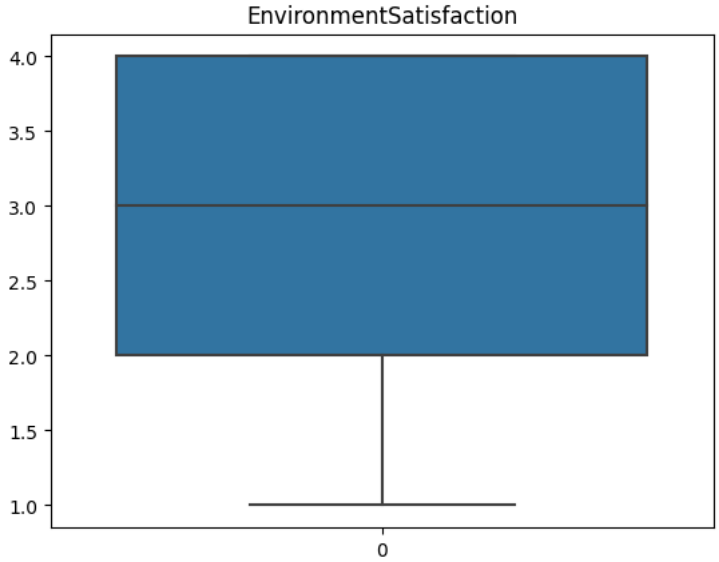


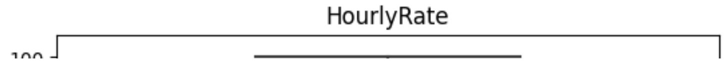


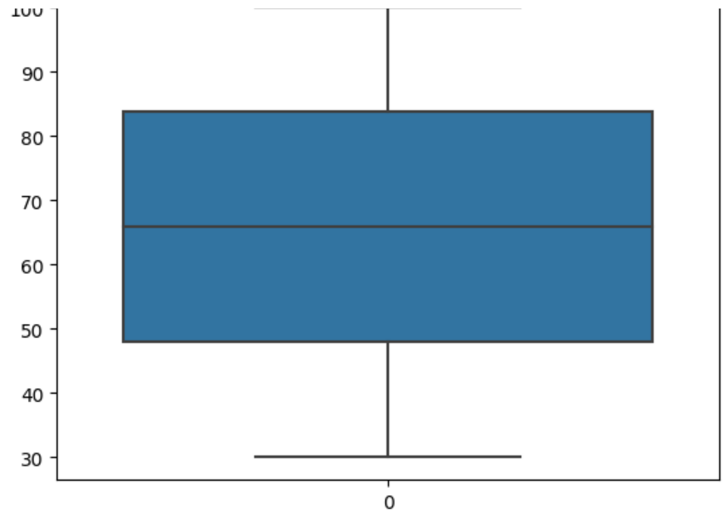

2.5
2.0
1.5
1.0
0

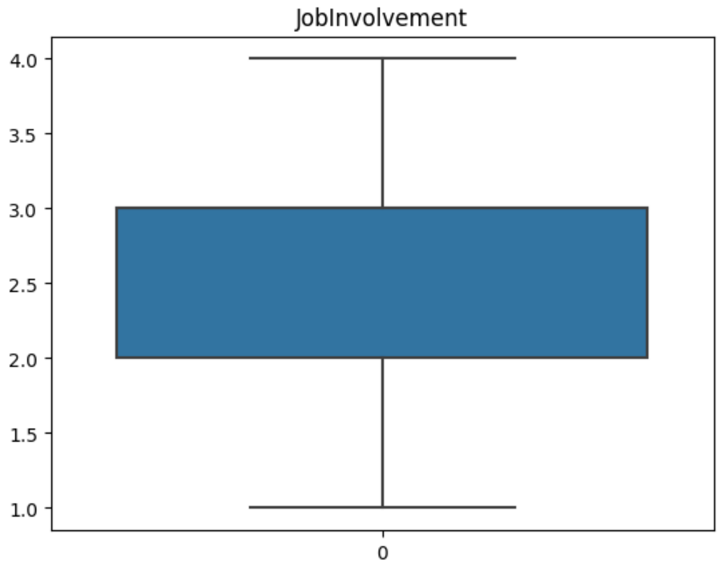

EmployeeCount
1.04
1.02
1.00
0.98
0.96
0

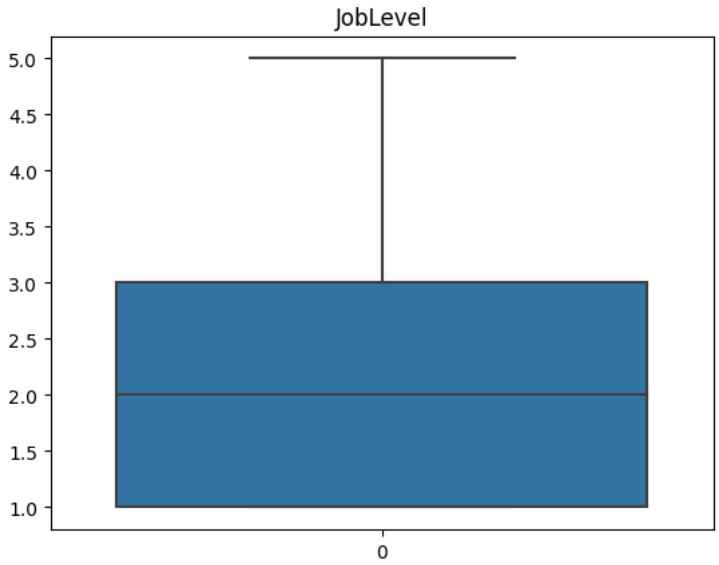

EmployeeNumber
2000
1500
1000
500
0
0

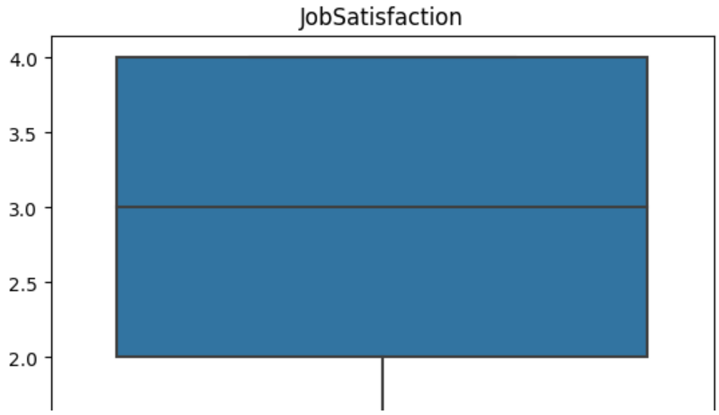

EnvironmentSatisfaction
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0

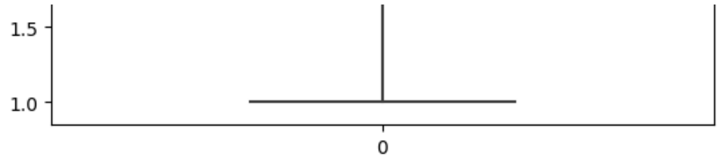

HourlyRate


100
90
80
70
60
50
40
30
0

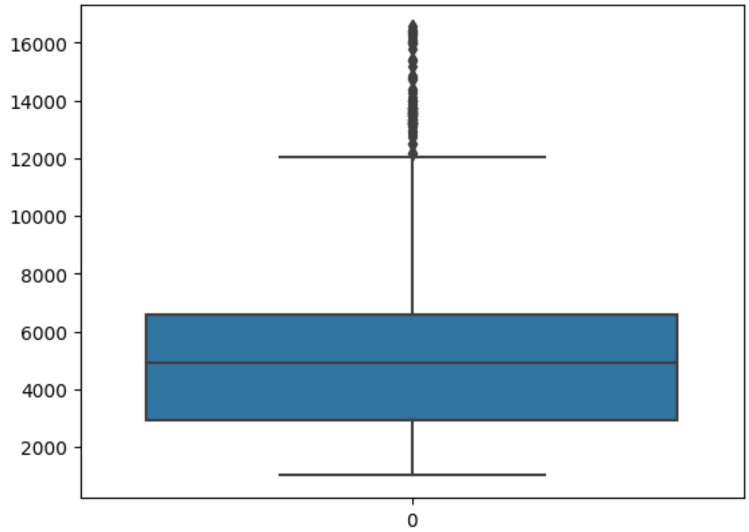

JobInvolvement
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0


JobLevel
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0

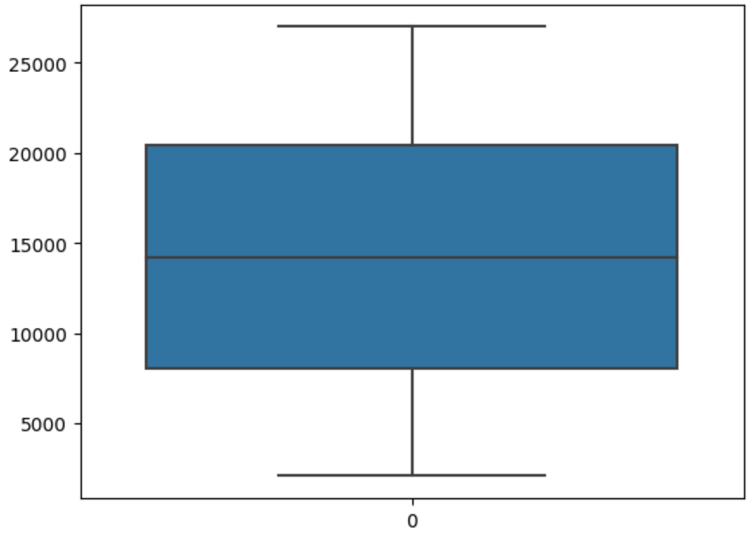

JobSatisfaction
4.0
3.5
3.0
2.5
2.0



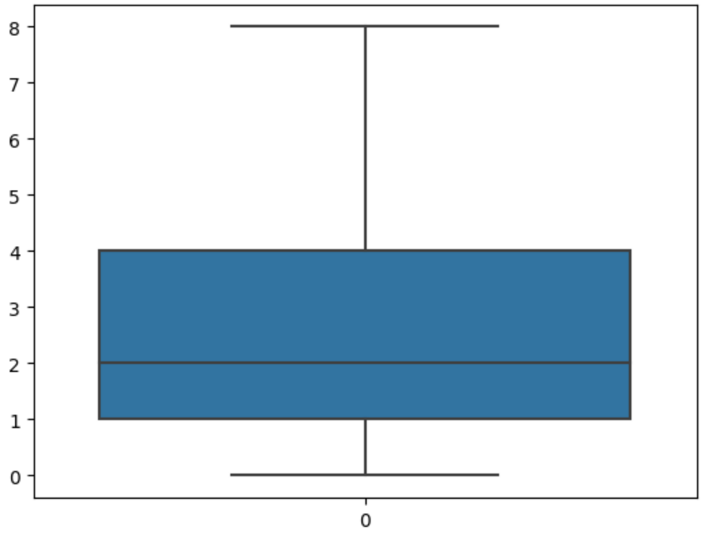
MonthlyIncome



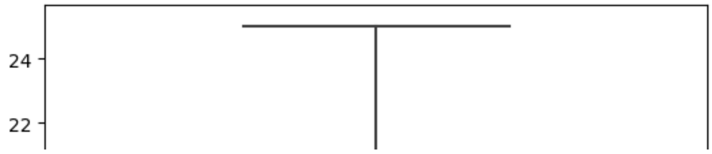
MonthlyRate

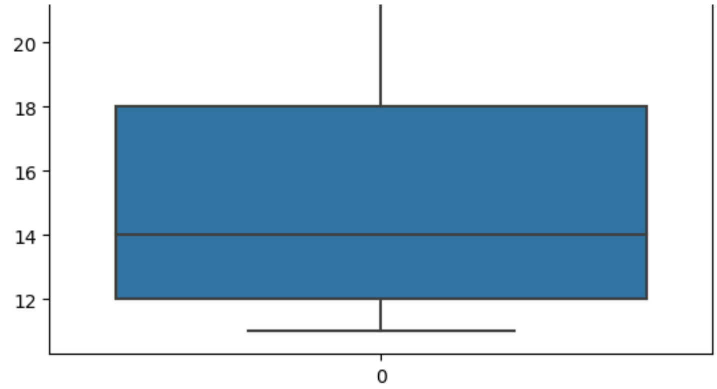


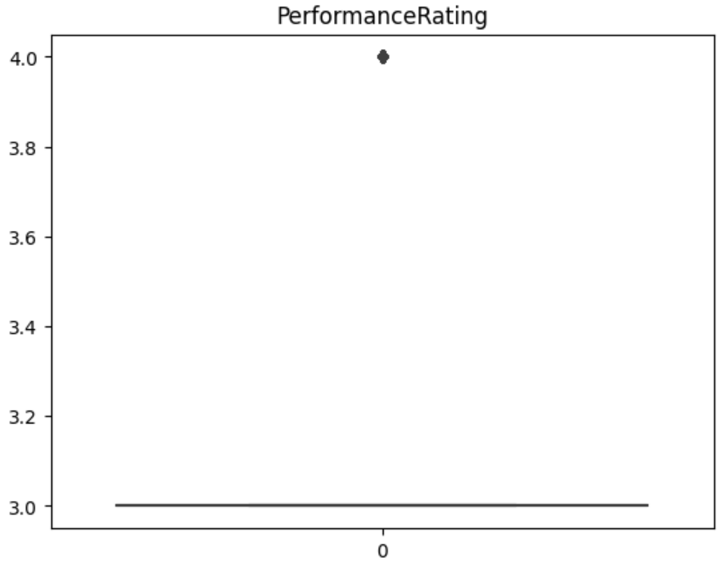
NumCompaniesWorked

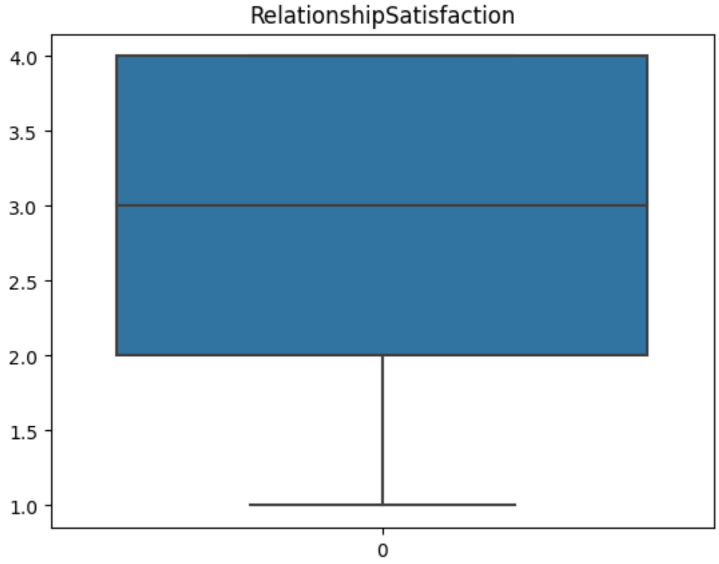


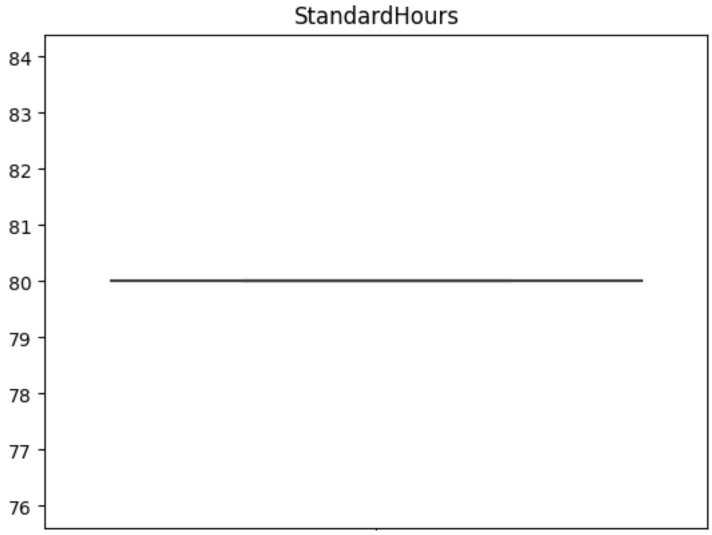
PercentSalaryHike



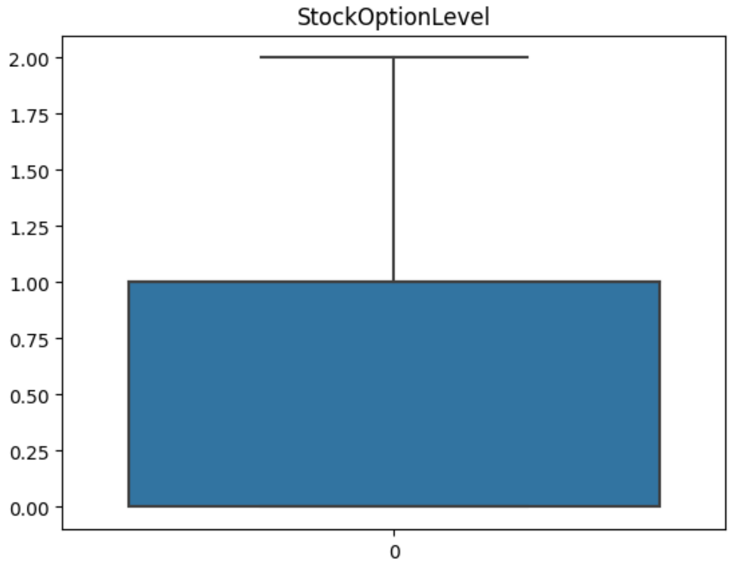

20
18
16
14
12
0

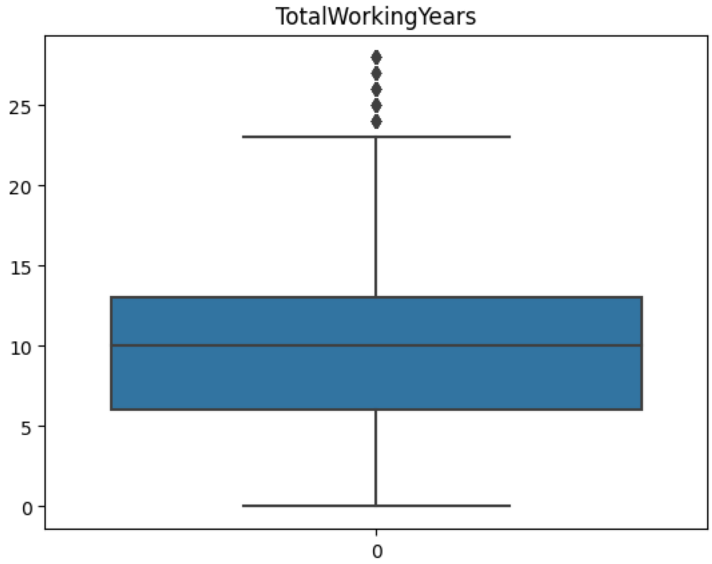

PerformanceRating
4.0
3.8
3.6
3.4
3.2
3.0
0

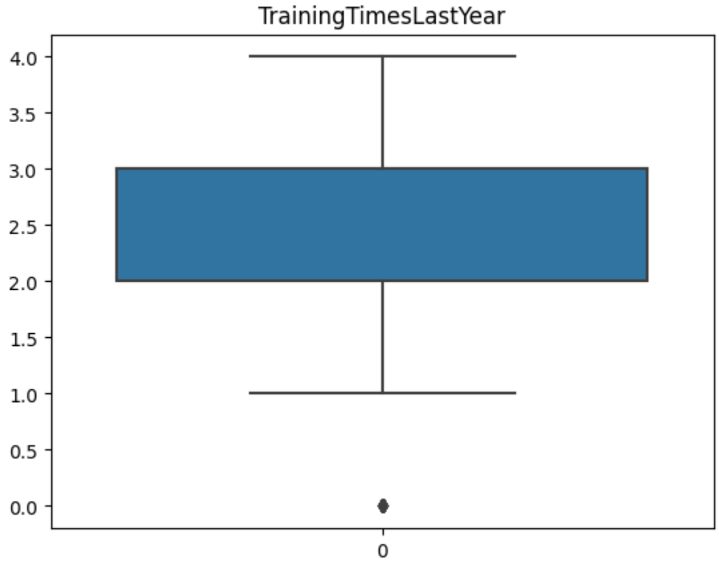

RelationshipSatisfaction
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0

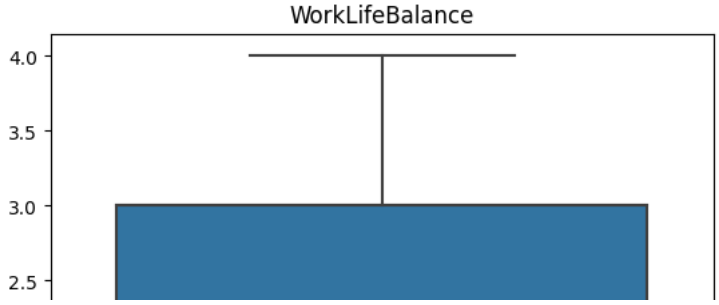

StandardHours
84
83
82
81
80
79
78
77
76

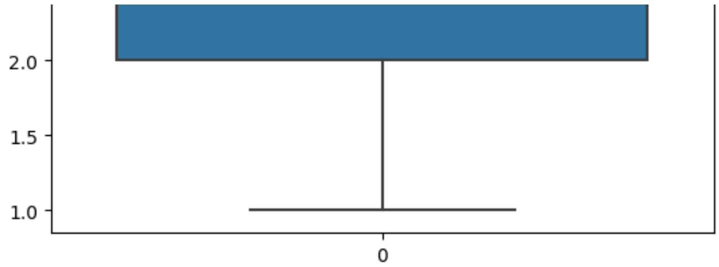
0

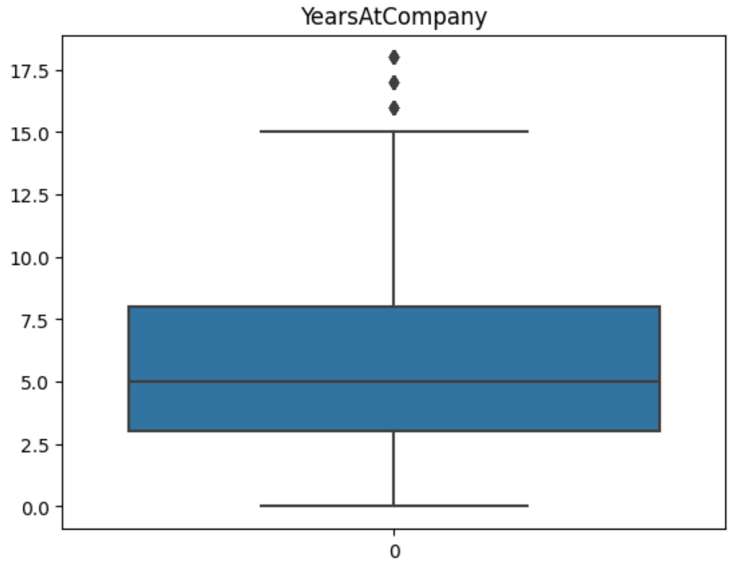

StockOptionLevel
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0

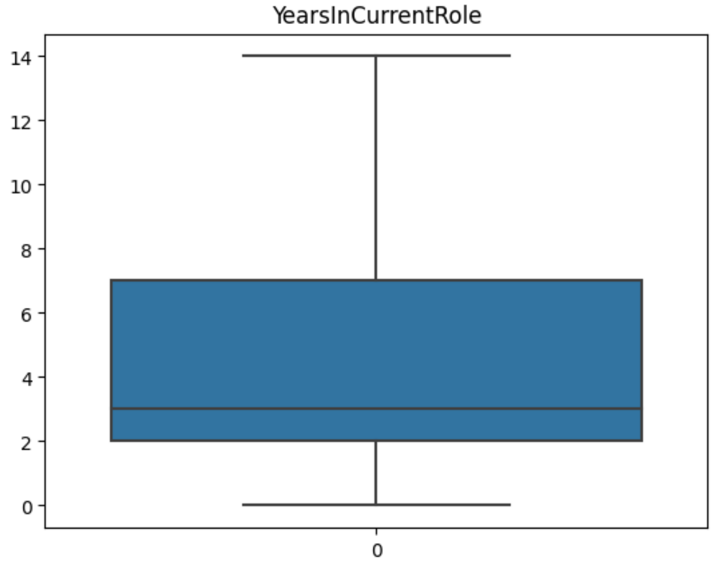

TotalWorkingYears
25
20
15
10
5
0
0

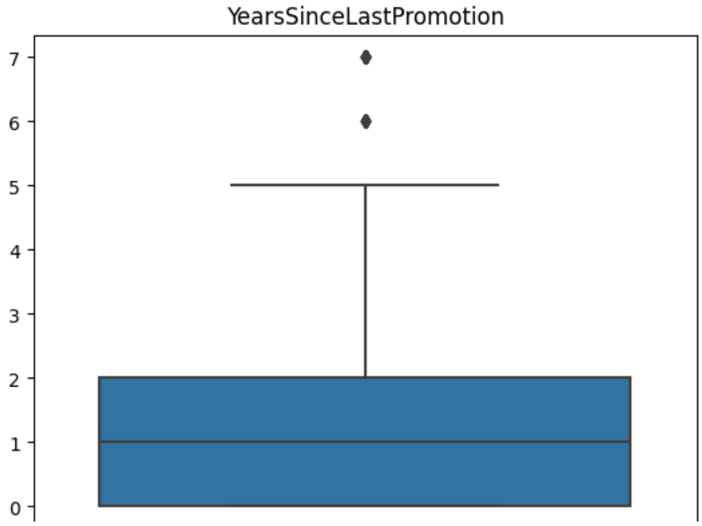

TrainingTimesLastYear
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0


WorkLifeBalance
4.0
3.5
3.0
2.5









Splitting Dependent and Independent variables

```
df.drop(columns=['EmployeeCount','StandardHours','EmployeeNumber','Over18'],axis=1,inplace=True)
```

```
df.shape
```

```
(1470, 31)
```

```
df.head(5)
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Educa
0	41	Yes	Travel_Rarely	1102	Sales	1	
1	49	No	Travel_Frequently	279	Research & Development	8	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	
4	27	No	Travel_Rarely	591	Research & Development	2	

5 rows × 31 columns

```
x=df.drop("Attrition",axis=1)
y=df.iloc[:,1:2]
```

```
x.head()
```

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Educa
0	41	Travel_Rarely	1102	Sales	1	2	Li
1	49	Travel_Frequently	279	Research & Development	8	1	Li
2	37	Travel_Rarely	1373	Research & Development	2	2	
3	33	Travel_Frequently	1392	Research & Development	3	4	Li
4	27	Travel_Rarely	591	Research & Development	2	1	

5 rows × 30 columns

```
y.head()
```

	Attrition
0	Yes
1	No
2	Yes
3	No
4	No

```
y=np.squeeze(y)
y.head()
```

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

```
type(x)
```

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

```
type(y)
```

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

Perform Encoding

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
y_encoded=pd.Series(le.fit_transform(y))
```

```
x.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 30 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Age                       1470 non-null   int64  
 1   BusinessTravel            1470 non-null   object 
 2   DailyRate                 1470 non-null   int64  
 3   Department                1470 non-null   object 
 4   DistanceFromHome          1470 non-null   int64  
 5   Education                 1470 non-null   int64  
 6   EducationField            1470 non-null   object 
 7   EnvironmentSatisfaction   1470 non-null   int64  
 8   Gender                    1470 non-null   object 
 9   HourlyRate                1470 non-null   int64  
 10  JobInvolvement            1470 non-null   int64  
 11  JobLevel                  1470 non-null   int64  
 12  JobRole                   1470 non-null   object 
 13  JobSatisfaction           1470 non-null   int64  
 14  MaritalStatus             1470 non-null   object 
 15  MonthlyIncome             1470 non-null   float64
 16  MonthlyRate               1470 non-null   int64  
 17  NumCompaniesWorked        1470 non-null   float64
 18  OverTime                  1470 non-null   object 
 19  PercentSalaryHike         1470 non-null   int64  
 20  PerformanceRating         1470 non-null   int64  
 21  RelationshipSatisfaction  1470 non-null   int64  
 22  StockOptionLevel          1470 non-null   float64
 23  TotalWorkingYears         1470 non-null   float64
 24  TrainingTimesLastYear     1470 non-null   float64
 25  WorkLifeBalance           1470 non-null   int64  
 26  YearsAtCompany            1470 non-null   float64
 27  YearsInCurrentRole        1470 non-null   float64
 28  YearsSinceLastPromotion   1470 non-null   float64
 29  YearsWithCurrManager      1470 non-null   float64
dtypes: float64(9), int64(14), object(7)
memory usage: 344.7+ KB
```

```
Business_Travel1=pd.get_dummies(df["BusinessTravel"],drop_first=True).astype(int)
Department1=pd.get_dummies(df["Department"],drop_first=True).astype(int)
EducationField1=pd.get_dummies(df["EducationField"],drop_first=True).astype(int)
Gender1=pd.get_dummies(df["Gender"],drop_first=True).astype(int)
JobRole1=pd.get_dummies(df["JobRole"],drop_first=True).astype(int)
MaritalStatus1=pd.get_dummies(df["MaritalStatus"],drop_first=True).astype(int)
OverTime1=pd.get_dummies(df["OverTime"],drop_first=True).astype(int)
```

```
x=pd.concat([x,Business_Travel1],axis=1)
x=pd.concat([x,Department1],axis=1)
x=pd.concat([x,EducationField1],axis=1)
x=pd.concat([x,Gender1],axis=1)
x=pd.concat([x,JobRole1],axis=1)
x=pd.concat([x,MaritalStatus1],axis=1)
x=pd.concat([x,OverTime1],axis=1)
```

```
x.drop(['BusinessTravel', 'Department', 'EducationField','Gender', 'JobRole', 'MaritalStatus', 'OverTime'],axis = 1, inplace = True)
```

```
x.head()
```

```
y_encoded.head()
```

```
0    1
1    0
2    1
3    0
4    0
dtype: int64
```

Feature Scaling.

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

```
x_scaled.head()
```

	Age	DailyRate	DistanceFromHome	Education	EnvironmentSatisfaction	Hour
0	0.446350	0.742527	-1.010909	-0.891688	-0.660531	1.
1	1.322365	-1.297775	-0.147150	-1.868426	0.254625	-0.
2	0.008343	1.414363	-0.887515	-0.891688	1.169781	1.
3	-0.429664	1.461466	-0.764121	1.061787	1.169781	-0.
4	-1.086676	-0.524295	-0.887515	-1.868426	-1.575686	-1.

5 rows × 44 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_scaled,y_encoded,test_size=0.2,random_state=0)
```

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

```
(1176, 44) (294, 44) (1176,) (294,)
```

Preprocessing Done

-->Model Building - Logistic regression

[ ] ↳ *17 cells hidden*

--> Model Building - Decision Tree

Import the Model Builiding Libraries

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV
```

Initializing the Model

```
dtc=DecisionTreeClassifier()
```

Hyper parametering and Training of Model

```
parameters=[{
```

```
parameters=[{
    'criterion':['gini','Entropy'],
    'splitter':['best','random'],
    'max_depth':[1,2,3,4,5],
    'max_features':['auto', 'sqrt', 'log2'],
    'random_state':[0,42],
}]

griddtc=GridSearchCV(dtc,param_grid=parameters,cv=5,scoring='accuracy')
```

Training the Model

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

```
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
```

```
griddtc.best_params_
```

```
{'criterion': 'gini',
 'max_depth': 4,
 'max_features': 'auto',
 'random_state': 42,
 'splitter': 'random'}

/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
```

Testing the Model

```
/usr/local/lib/python3.10/dist-packages/sklearn/tree/_classes.py:269: FutureWarnin
```

```
y_pred1=griddtc.predict(x_test)
```

```
  warnings.warn(
```

```
pd.DataFrame({"Actual_values":y_test,"Predicted_values":y_pred1})
```

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

294 rows × 2 columns

```
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_search.py:952: Us
 0.84013704 0.84013704 0.84013704 0.84013704 0.84013704 0.84013704
```

Evaluation of Model & Performance metrics

```
 0.84014064 0.83929318 0.84096646 0.84183916 0.84185359 0.84269383
```

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

```
 0.81634331 0.84268662 0.83588172 0.84013343 0.83164443 0.84012982
```

```
print("Accuracy of model :",accuracy_score(y_test,y_pred1))
```

```
Accuracy of model : 0.8367346938775511

        nan        nan        nan        nan        nan        nan
```

```
confusion_matrix(y_test,y_pred1)
```

```
array([[240,   5],
       [ 43,   6]])

  warnings.warn(
```

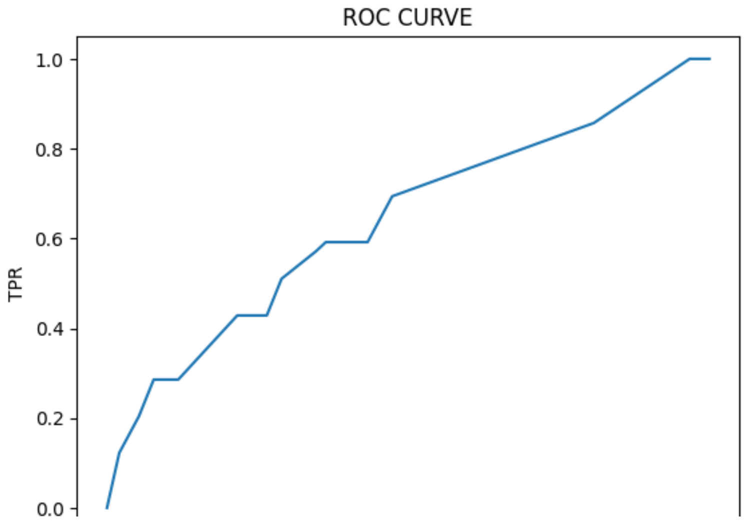
```
print(classification_report(y_test,y_pred1))
```

```
              precision    recall  f1-score   support

           0       0.85      0.98      0.91       245
           1       0.55      0.12      0.20        49

    accuracy                           0.84       294
   macro avg       0.70      0.55      0.55       294
weighted avg       0.80      0.84      0.79       294
```

```
#ROC-AUC Curve
probability=griddtc.predict_proba(x_test)[:,1]
fpr,tpr,threshsholds = roc_curve(y_test,probability)
plt.plot(fpr,tpr)
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('ROC CURVE')
plt.show()
```



```
threshsholds
```

```
     array([1.67647059, 0.67647059, 0.39583333, 0.37037037, 0.33823529,
            0.23577236, 0.2       , 0.18867925, 0.18604651, 0.16666667,
            0.15517241, 0.08474576, 0.07715134, 0.05936073, 0.        ])
```

```
#Tree Visualization using basic Decision Tree
dtc.fit(x_train,y_train)
from sklearn import tree
plt.figure(figsize=(25,15))
tree.plot_tree(dtc,filled=True)
```

```
[Text(0.31958512931034483, 0.96875, 'x[16] <= -1.397\ngini = 0.269\nsamples =
1176\nvalue = [988, 188]'),
 Text(0.0896551724137931, 0.90625, 'x[42] <= 0.387\ngini = 0.5\nsamples =
78\nvalue = [39, 39]'),
 Text(0.05172413793103448, 0.84375, 'x[2] <= 0.902\ngini = 0.426\nsamples =
39\nvalue = [27, 12]'),
 Text(0.034482758620689655, 0.78125, 'x[23] <= 0.797\ngini = 0.312\nsamples =
31\nvalue = [25, 6]'),
 Text(0.020689655172413793, 0.71875, 'x[8] <= -1.114\ngini = 0.198\nsamples =
27\nvalue = [24, 3]'),
 Text(0.013793103448275862, 0.65625, 'x[43] <= 0.482\ngini = 0.5\nsamples =
6\nvalue = [3, 3]'),
 Text(0.006896551724137931, 0.59375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.020689655172413793, 0.59375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.027586206896551724, 0.65625, 'gini = 0.0\nsamples = 21\nvalue = [21,
0]'),
 Text(0.04827586206896552, 0.71875, 'x[4] <= 0.712\ngini = 0.375\nsamples =
4\nvalue = [1, 3]'),
 Text(0.041379310344827586, 0.65625, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.05517241379310345, 0.65625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.06896551724137931, 0.78125, 'x[12] <= 1.446\ngini = 0.375\nsamples =
8\nvalue = [2, 6]'),
 Text(0.06206896551724138, 0.71875, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.07586206896551724, 0.71875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.12758620689655173, 0.84375, 'x[38] <= 0.755\ngini = 0.426\nsamples =
39\nvalue = [12, 27]'),
 Text(0.10344827586206896, 0.78125, 'x[29] <= 0.397\ngini = 0.26\nsamples =
26\nvalue = [4, 22]'),
 Text(0.0896551724137931, 0.71875, 'x[5] <= 1.482\ngini = 0.095\nsamples =
20\nvalue = [1, 19]'),
 Text(0.08275862068965517, 0.65625, 'gini = 0.0\nsamples = 18\nvalue = [0, 18]'),
 Text(0.09655172413793103, 0.65625, 'x[4] <= 0.712\ngini = 0.5\nsamples =
2\nvalue = [1, 1]'),
 Text(0.0896551724137931, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.10344827586206896, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.11724137931034483, 0.71875, 'x[24] <= -0.462\ngini = 0.5\nsamples =
6\nvalue = [3, 3]'),
 Text(0.1103448275862069, 0.65625, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.12413793103448276, 0.65625, 'x[0] <= -1.196\ngini = 0.375\nsamples =
4\nvalue = [3, 1]'),
 Text(0.11724137931034483, 0.59375, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.1310344827586207, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.15172413793103448, 0.78125, 'x[10] <= 1.103\ngini = 0.473\nsamples =
13\nvalue = [8, 5]'),
 Text(0.14482758620689656, 0.71875, 'x[1] <= 0.712\ngini = 0.32\nsamples =
10\nvalue = [8, 2]'),
 Text(0.13793103448275862, 0.65625, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(0.15172413793103448, 0.65625, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.15862068965517243, 0.71875, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.5495150862068966, 0.90625, 'x[43] <= 0.482\ngini = 0.235\nsamples =
1098\nvalue = [949, 149]'),
 Text(0.2964439655172414, 0.84375, 'x[18] <= -1.786\ngini = 0.162\nsamples =
798\nvalue = [727, 71]'),
 Text(0.19310344827586207, 0.78125, 'x[17] <= 1.161\ngini = 0.38\nsamples =
47\nvalue = [35, 12]'),
 Text(0.18620689655172415, 0.71875, 'x[6] <= -0.323\ngini = 0.325\nsamples =
44\nvalue = [35, 9]'),
 Text(0.16551724137931034, 0.65625, 'x[1] <= 0.852\ngini = 0.498\nsamples =
15\nvalue = [8, 7]'),
 Text(0.15862068965517243, 0.59375, 'x[2] <= 0.532\ngini = 0.42\nsamples =
10\nvalue = [3, 7]'),
 Text(0.15172413793103448, 0.53125, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.16551724137931034, 0.53125, 'x[0] <= -0.703\ngini = 0.375\nsamples =
4\nvalue = [3, 1]'),
 Text(0.15862068965517243, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.1724137931034483, 0.46875, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.1724137931034483, 0.59375, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.20689655172413793, 0.65625, 'x[0] <= -0.539\ngini = 0.128\nsamples =
29\nvalue = [27, 2]'),
 Text(0.2, 0.59375, 'x[2] <= -0.702\ngini = 0.408\nsamples = 7\nvalue = [5, 2]'),
 Text(0.19310344827586207, 0.53125, 'x[11] <= 0.253\ngini = 0.444\nsamples =
3\nvalue = [1, 2]'),
 Text(0.18620689655172415, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.2, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.20689655172413793, 0.53125, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.21379310344827587, 0.59375, 'gini = 0.0\nsamples = 22\nvalue = [22, 0]'),
 Text(0.2, 0.71875, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.39978448275862066, 0.78125, 'x[19] <= -0.309\ngini = 0.145\nsamples =
751\nvalue = [692, 59]'),
 Text(0.3129310344827586, 0.71875, 'x[4] <= -1.118\ngini = 0.218\nsamples =
257\nvalue = [225, 32]'),
 Text(0.27413793103448275, 0.65625, 'x[22] <= -0.445\ngini = 0.355\nsamples =
65\nvalue = [50, 15]'),
 Text(0.2517241379310345, 0.59375, 'x[22] <= -1.039\ngini = 0.303\nsamples =
59\nvalue = [48, 11]'),
 Text(0.22758620689655173, 0.53125, 'x[6] <= -0.323\ngini = 0.463\nsamples =
22\nvalue = [14, 8]'),
 Text(0.21379310344827587, 0.46875, 'x[5] <= -1.151\ngini = 0.198\nsamples =
9\nvalue = [8, 1]'),
 Text(0.20689655172413793, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.2206896551724138, 0.40625, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(0.2413793103448276, 0.46875, 'x[5] <= -0.388\ngini = 0.497\nsamples =
13\nvalue = [6, 7]'),
 Text(0.23448275862068965, 0.40625, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.2482758620689655, 0.40625, 'x[2] <= -0.024\ngini = 0.346\nsamples =
9\nvalue = [2, 7]'),
 Text(0.2413793103448276, 0.34375, 'x[25] <= -0.323\ngini = 0.444\nsamples =
3\nvalue = [2, 1]'),
 Text(0.23448275862068965, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.2482758620689655, 0.28125, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.25517241379310346, 0.34375, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.27586206896551724, 0.53125, 'x[8] <= -1.114\ngini = 0.149\nsamples =
37\nvalue = [34, 3]'),
 Text(0.2689655172413793, 0.46875, 'x[19] <= -0.566\ngini = 0.5\nsamples =
6\nvalue = [3, 3]'),
 Text(0.2620689655172414, 0.40625, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.27586206896551724, 0.40625, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.2827586206896552, 0.46875, 'gini = 0.0\nsamples = 31\nvalue = [31, 0]'),
 Text(0.296551724137931, 0.59375, 'x[6] <= -1.729\ngini = 0.444\nsamples =
6\nvalue = [2, 4]'),
 Text(0.2896551724137931, 0.53125, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.30344827586206896, 0.53125, 'x[8] <= -0.661\ngini = 0.444\nsamples =
3\nvalue = [2, 1]'),
 Text(0.296551724137931, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3103448275862069, 0.46875, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.35172413793103446, 0.65625, 'x[27] <= 0.178\ngini = 0.161\nsamples =
192\nvalue = [175, 17]'),
 Text(0.3448275862068966, 0.59375, 'x[18] <= -0.37\ngini = 0.24\nsamples =
122\nvalue = [105, 17]'),
 Text(0.3310344827586207, 0.53125, 'x[5] <= 0.399\ngini = 0.463\nsamples =
22\nvalue = [14, 8]'),
 Text(0.32413793103448274, 0.46875, 'x[0] <= -0.156\ngini = 0.444\nsamples =
12\nvalue = [4, 8]'),
 Text(0.31724137931034485, 0.40625, 'x[1] <= -1.301\ngini = 0.198\nsamples =
9\nvalue = [1, 8]'),
 Text(0.3103448275862069, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.32413793103448274, 0.34375, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
 Text(0.3310344827586207, 0.40625, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.33793103448275863, 0.46875, 'gini = 0.0\nsamples = 10\nvalue = [10, 0]'),
 Text(0.3586206896551724, 0.53125, 'x[1] <= -1.711\ngini = 0.164\nsamples =
100\nvalue = [91, 9]'),
 Text(0.35172413793103446, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.36551724137931035, 0.46875, 'x[9] <= -0.745\ngini = 0.149\nsamples =
99\nvalue = [91, 8]'),
 Text(0.35172413793103446, 0.40625, 'x[3] <= 1.55\ngini = 0.283\nsamples =
41\nvalue = [34, 7]'),
 Text(0.3448275862068966, 0.34375, 'x[0] <= 1.706\ngini = 0.224\nsamples =
39\nvalue = [34, 5]'),
 Text(0.33793103448275863, 0.28125, 'x[16] <= 1.219\ngini = 0.188\nsamples =
38\nvalue = [34, 4]'),
 Text(0.3310344827586207, 0.21875, 'x[9] <= -0.848\ngini = 0.149\nsamples =
37\nvalue = [34, 3]'),
 Text(0.32413793103448274, 0.15625, 'gini = 0.0\nsamples = 29\nvalue = [29, 0]'),
 Text(0.33793103448275863, 0.15625, 'x[12] <= -0.467\ngini = 0.469\nsamples =
8\nvalue = [5, 3]'),
 Text(0.3310344827586207, 0.09375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.3448275862068966, 0.09375, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.3448275862068966, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.35172413793103446, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3586206896551724, 0.34375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.3793103448275862, 0.40625, 'x[1] <= 1.626\ngini = 0.034\nsamples =
58\nvalue = [57, 1]'),
 Text(0.3724137931034483, 0.34375, 'gini = 0.0\nsamples = 57\nvalue = [57, 0]'),
 Text(0.38620689655172413, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.3586206896551724, 0.59375, 'gini = 0.0\nsamples = 70\nvalue = [70, 0]'),
 Text(0.4866379310344828, 0.71875, 'x[9] <= 0.385\ngini = 0.103\nsamples =
494\nvalue = [467, 27]'),
 Text(0.4482758620689655, 0.65625, 'x[20] <= 2.837\ngini = 0.056\nsamples =
345\nvalue = [335, 10]'),
 Text(0.4413793103448276, 0.59375, 'x[22] <= 2.822\ngini = 0.051\nsamples =
344\nvalue = [335, 9]'),
 Text(0.4206896551724138, 0.53125, 'x[5] <= 0.227\ngini = 0.046\nsamples =
342\nvalue = [334, 8]'),
 Text(0.4, 0.46875, 'x[11] <= 1.854\ngini = 0.01\nsamples = 202\nvalue = [201,
1]'),
 Text(0.3931034482758621, 0.40625, 'gini = 0.0\nsamples = 184\nvalue = [184,
0]'),
 Text(0.4068965517241379, 0.40625, 'x[9] <= -0.642\ngini = 0.105\nsamples =
18\nvalue = [17, 1]'),
 Text(0.4, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.41379310344827586, 0.34375, 'gini = 0.0\nsamples = 17\nvalue = [17, 0]'),
 Text(0.4413793103448276, 0.46875, 'x[40] <= 1.922\ngini = 0.095\nsamples =
140\nvalue = [133, 7]'),
 Text(0.43448275862068964, 0.40625, 'x[20] <= 0.137\ngini = 0.083\nsamples =
139\nvalue = [133, 6]'),
 Text(0.42758620689655175, 0.34375, 'x[23] <= 0.797\ngini = 0.161\nsamples =
68\nvalue = [62, 6]'),
 Text(0.4, 0.28125, 'x[14] <= -1.122\ngini = 0.098\nsamples = 58\nvalue = [55,
3]'),
 Text(0.3793103448275862, 0.21875, 'x[1] <= 0.186\ngini = 0.346\nsamples =
9\nvalue = [7, 2]'),
9\nvalue = [7, 2]'),
 Text(0.3724137931034483, 0.15625, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.38620689655172413, 0.15625, 'x[22] <= -1.039\ngini = 0.444\nsamples =
3\nvalue = [1, 2]'),
 Text(0.3793103448275862, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.3931034482758621, 0.09375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.4206896551724138, 0.21875, 'x[9] <= -1.063\ngini = 0.04\nsamples =
49\nvalue = [48, 1]'),
 Text(0.41379310344827586, 0.15625, 'x[17] <= -1.173\ngini = 0.444\nsamples =
3\nvalue = [2, 1]'),
 Text(0.4068965517241379, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.4206896551724138, 0.09375, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.42758620689655175, 0.15625, 'gini = 0.0\nsamples = 46\nvalue = [46, 0]'),
 Text(0.45517241379310347, 0.28125, 'x[8] <= -0.207\ngini = 0.42\nsamples =
10\nvalue = [7, 3]'),
 Text(0.4482758620689655, 0.21875, 'x[9] <= -0.837\ngini = 0.375\nsamples =
4\nvalue = [1, 3]'),
 Text(0.4413793103448276, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.45517241379310347, 0.15625, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.46206896551724136, 0.21875, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.4413793103448276, 0.34375, 'gini = 0.0\nsamples = 71\nvalue = [71, 0]'),
 Text(0.4482758620689655, 0.40625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.46206896551724136, 0.53125, 'x[35] <= 1.695\ngini = 0.5\nsamples =
2\nvalue = [1, 1]'),
 Text(0.45517241379310347, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.4689655172413793, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.45517241379310347, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.525, 0.65625, 'x[5] <= 1.654\ngini = 0.202\nsamples = 149\nvalue = [132,
17]'),
 Text(0.5181034482758621, 0.59375, 'x[9] <= 0.391\ngini = 0.193\nsamples =
148\nvalue = [132, 16]'),
 Text(0.5112068965517241, 0.53125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.525, 0.53125, 'x[1] <= -1.621\ngini = 0.183\nsamples = 147\nvalue = [132,
15]'),
 Text(0.4827586206896552, 0.46875, 'x[14] <= -0.196\ngini = 0.49\nsamples =
7\nvalue = [4, 3]'),
 Text(0.47586206896551725, 0.40625, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.4896551724137931, 0.40625, 'x[16] <= 1.061\ngini = 0.375\nsamples =
4\nvalue = [1, 3]'),
 Text(0.4827586206896552, 0.34375, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.496551724137931, 0.34375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5672413793103448, 0.46875, 'x[41] <= 0.085\ngini = 0.157\nsamples =
140\nvalue = [128, 12]'),
 Text(0.5275862068965518, 0.40625, 'x[1] <= -1.47\ngini = 0.07\nsamples =
82\nvalue = [79, 3]'),
 Text(0.5103448275862069, 0.34375, 'x[7] <= 0.846\ngini = 0.5\nsamples = 2\nvalue
= [1, 1]'),
 Text(0.503448275862069, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5172413793103449, 0.28125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5448275862068965, 0.34375, 'x[16] <= 2.726\ngini = 0.049\nsamples =
80\nvalue = [78, 2]'),
 Text(0.5310344827586206, 0.28125, 'x[2] <= 1.765\ngini = 0.025\nsamples =
78\nvalue = [77, 1]'),
 Text(0.5241379310344828, 0.21875, 'gini = 0.0\nsamples = 76\nvalue = [76, 0]'),
 Text(0.5379310344827586, 0.21875, 'x[9] <= 0.724\ngini = 0.5\nsamples = 2\nvalue
= [1, 1]'),
 Text(0.5310344827586206, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.5448275862068965, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5586206896551724, 0.28125, 'x[27] <= 0.178\ngini = 0.5\nsamples =
2\nvalue = [1, 1]'),
 Text(0.5517241379310345, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5655172413793104, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.6068965517241379, 0.40625, 'x[9] <= 1.606\ngini = 0.262\nsamples =
58\nvalue = [49, 9]'),
 Text(0.6, 0.34375, 'x[0] <= 0.611\ngini = 0.375\nsamples = 36\nvalue = [27,
9]'),
 Text(0.5862068965517241, 0.28125, 'x[20] <= 2.087\ngini = 0.211\nsamples =
25\nvalue = [22, 3]'),
 Text(0.5793103448275863, 0.21875, 'x[5] <= -1.52\ngini = 0.153\nsamples =
24\nvalue = [22, 2]'),
 Text(0.5724137931034483, 0.15625, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5862068965517241, 0.15625, 'x[17] <= 1.161\ngini = 0.083\nsamples =
23\nvalue = [22, 1]'),
 Text(0.5793103448275863, 0.09375, 'gini = 0.0\nsamples = 21\nvalue = [21, 0]'),
 Text(0.593103448275862, 0.09375, 'x[2] <= -0.024\ngini = 0.5\nsamples = 2\nvalue
= [1, 1]'),
 Text(0.5862068965517241, 0.03125, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6, 0.03125, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.593103448275862, 0.21875, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.6137931034482759, 0.28125, 'x[11] <= 0.025\ngini = 0.496\nsamples =
11\nvalue = [5, 6]'),
 Text(0.6068965517241379, 0.21875, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.6206896551724138, 0.21875, 'x[2] <= -0.024\ngini = 0.408\nsamples =
7\nvalue = [5, 2]'),
 Text(0.6137931034482759, 0.15625, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.6275862068965518, 0.15625, 'x[15] <= -0.271\ngini = 0.444\nsamples =
3\nvalue = [1, 2]'),
 Text(0.6206896551724138, 0.09375, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.6344827586206897, 0.09375, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.6137931034482759, 0.34375, 'gini = 0.0\nsamples = 22\nvalue = [22, 0]'),
 Text(0.531896551724138, 0.59375, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8025862068965517, 0.84375, 'x[9] <= -0.458\ngini = 0.385\nsamples =
300\nvalue = [222, 78]'),
 Text(0.7008620689655173, 0.78125, 'x[15] <= -0.271\ngini = 0.5\nsamples =
96\nvalue = [49, 47]'),
 Text(0.6655172413793103, 0.71875, 'x[2] <= -0.456\ngini = 0.459\nsamples =
42\nvalue = [15, 27]'),
 Text(0.6413793103448275, 0.65625, 'x[10] <= -0.283\ngini = 0.499\nsamples =
23\nvalue = [12, 11]'),
 Text(0.6275862068965518, 0.59375, 'x[29] <= 0.397\ngini = 0.426\nsamples =
13\nvalue = [4, 9]'),
 Text(0.6206896551724138, 0.53125, 'x[17] <= -1.173\ngini = 0.298\nsamples =
11\nvalue = [2, 9]'),
 Text(0.6137931034482759, 0.46875, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.6275862068965518, 0.46875, 'x[10] <= -0.638\ngini = 0.18\nsamples =
10\nvalue = [1, 9]'),
 Text(0.6206896551724138, 0.40625, 'gini = 0.0\nsamples = 8\nvalue = [0, 8]'),
```

--> Model Building - Random Forest

```
 Text(0.6344827586206897, 0.53125, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
```

Import the Model Builiding Libraries

```
 Text(0.6620689655172414, 0.53125, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
 Text(0.6827586206896552, 0.59375, 'x[5] <= -1.077\ngini = 0.198\nsamples =
```

Initializing the Model

```
 Text(0.6827586206896552, 0.46875, 'gini = 0.0\nsamples = 15\nvalue = [0, 15]'),
```

```
rfc=RandomForestClassifier()
```

Hyper parametering and Training of Model

```
54\nvalue = [34, 20]'),
```

```
from sklearn.model_selection import GridSearchCV
parameters=[{
    'max_depth': list(range(10, 15)),
    'max_features': list(range(0,14))
}]
gridrfc=GridSearchCV(rfc,param_grid=parameters,cv=5,scoring='accuracy')
```

Training the Model

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

```
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_validation.py:378
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_sco

Below are more details about the failures:
```

```
gridrfc.best_params_
```

```
{'max_depth': 12, 'max_features': 12}
    estimator.fit(X_train, y_train, **fit_params)
```

Testing the Model

```
y_pred2=gridrfc.predict(x_test)
```

```
pd.DataFrame({"Actual_values":y_test,"Predicted_values":y_pred2})
```

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

294 rows × 2 columns

Evaluation of Model & Performance metrics

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

```
print("Accuracy of model :",accuracy_score(y_test,y_pred2))
```

```
Accuracy of model : 0.8503401360544217
```

```
confusion_matrix(y_test,y_pred2)
```

```
array([[241,   4],
       [ 40,   9]])
```

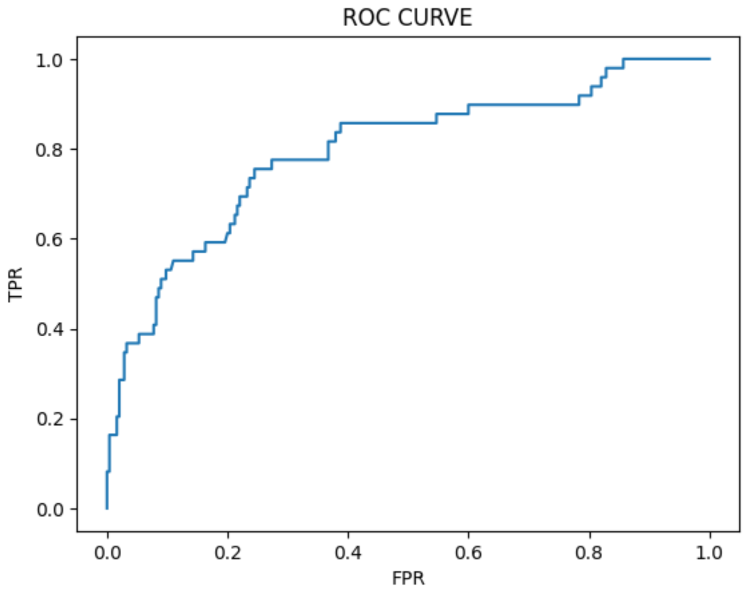
```
print(classification_report(y_test,y_pred2))
```

```
              precision    recall  f1-score   support

           0       0.86      0.98      0.92       245
           1       0.69      0.18      0.29        49

    accuracy                           0.85       294
   macro avg       0.77      0.58      0.60       294
weighted avg       0.83      0.85      0.81       294
```

```
#ROC-AUC Curve
probability=gridrfc.predict_proba(x_test)[:,1]
fpr,tpr,threshsholds = roc_curve(y_test,probability)
plt.plot(fpr,tpr)
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('ROC CURVE')
plt.show()
```



```
threshsholds
```

```
array([1.82000000e+00, 8.20000000e-01, 6.26666667e-01, 6.05666667e-01,
       5.64659091e-01, 5.21428571e-01, 4.70000000e-01, 4.59868903e-01,
       4.10388350e-01, 4.00000000e-01, 3.85641026e-01, 3.80000000e-01,
       3.69187062e-01, 3.40000000e-01, 3.34132505e-01, 3.26514286e-01,
       3.21818182e-01, 3.12213675e-01, 3.10000000e-01, 3.07346999e-01,
       3.06666667e-01, 3.01071429e-01, 3.00860806e-01, 2.96666667e-01,
       2.92538414e-01, 2.85833333e-01, 2.80000000e-01, 2.46279540e-01,
       2.43822608e-01, 2.32267856e-01, 2.28766776e-01, 2.00400000e-01,
       1.97239394e-01, 1.96000000e-01, 1.91515785e-01, 1.90666667e-01,
       1.90388350e-01, 1.90000000e-01, 1.88721683e-01, 1.86493506e-01,
       1.85322581e-01, 1.82995161e-01, 1.82174064e-01, 1.80845715e-01,
       1.78533333e-01, 1.77268623e-01, 1.76743661e-01, 1.65325000e-01,
       1.62129797e-01, 1.33476383e-01, 1.31818182e-01, 1.28613943e-01,
       1.28346711e-01, 1.26294949e-01, 1.23433092e-01, 8.63571300e-02,
       8.58860748e-02, 8.13957805e-02, 8.09610768e-02, 5.17257313e-02,
       5.14163965e-02, 4.64544339e-02, 4.58530435e-02, 4.39191206e-02,
       4.38987475e-02, 4.35495240e-02, 4.34359223e-02, 4.17571413e-02,
       4.16258628e-02, 1.73860421e-03])
```