```
#G.Sai Spandana
# 21BCE7149
# VITAP MORNING SLOT
# ASSIGNMENT-4
# Data Preprocessing on Employee Attrition DataSet.
```

Import libraries

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

```
from google.colab import drive
drive.mount('/content/drive')
```

```
    Mounted at /content/drive
```

```
a=pd.read_csv("/content/drive/MyDrive/DATASETS/WA_Fn-UseC_-HR-Employee-Attrition.csv")
```

```
a
```

	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	
...	...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research & Development	23	
1466	39	No	Travel_Rarely	613	Research & Development	6	
1467	27	No	Travel_Rarely	155	Research & Development	4	
1468	49	No	Travel_Frequently	1023	Sales	2	
1469	34	No	Travel_Rarely	628	Research & Development	8	

1470 rows × 35 columns

Read the data types

```
a.dtypes
```

```
    Age                          int64
    Attrition                   object
    BusinessTravel              object
    DailyRate                    int64
    Department                  object
    DistanceFromHome             int64
    Education                    int64
    EducationField              object
    EmployeeCount                int64
    EmployeeNumber               int64
    EnvironmentSatisfaction      int64
    Gender                      object
    HourlyRate                   int64
    JobInvolvement               int64
    JobLevel                     int64
    JobRole                     object
    JobSatisfaction              int64
    MaritalStatus               object
    MonthlyIncome                int64
    MonthlyRate                  int64
    NumCompaniesWorked           int64
```

```
Over18                      object
OverTime                    object
PercentSalaryHike            int64
PerformanceRating            int64
RelationshipSatisfaction     int64
StandardHours                int64
StockOptionLevel             int64
TotalWorkingYears            int64
TrainingTimesLastYear        int64
WorkLifeBalance              int64
YearsAtCompany               int64
YearsInCurrentRole           int64
YearsSinceLastPromotion      int64
YearsWithCurrManager         int64
dtype: object
```

Shape of the dataset

```
a.shape
```

```
(1470, 35)
```

Information about the dataset

```
a.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
```

Statistics about the dataset

```
a.describe()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	J
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	
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	3.000000	66.000000	

Null values identification

```
a.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
```

```
a.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
```

```
# there are no null values
```

Data Visualization

```
d=a.corr()
d
```

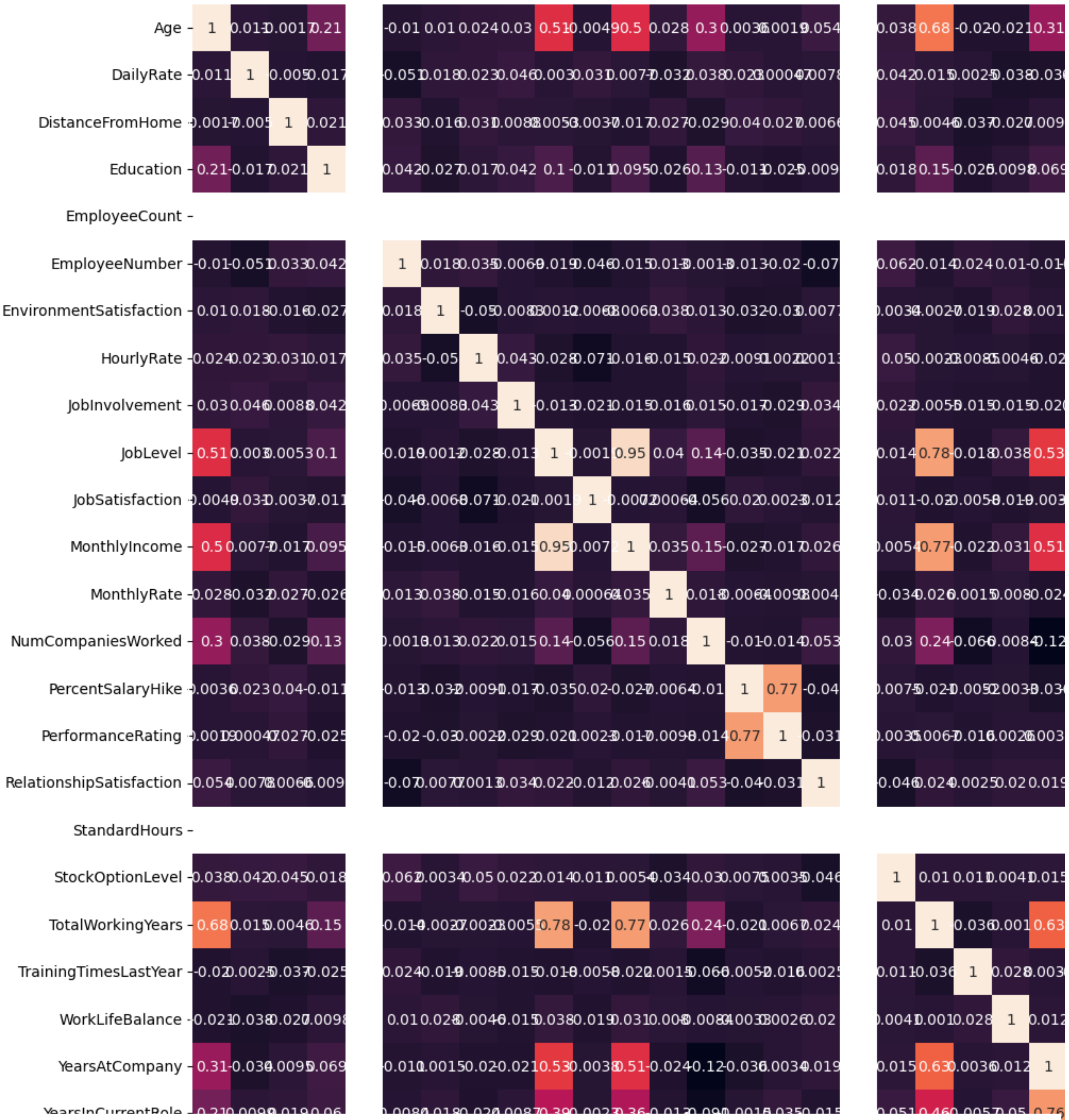
```
<ipython-input-12-385900cf86c7>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ver
  d=a.corr()
```

	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

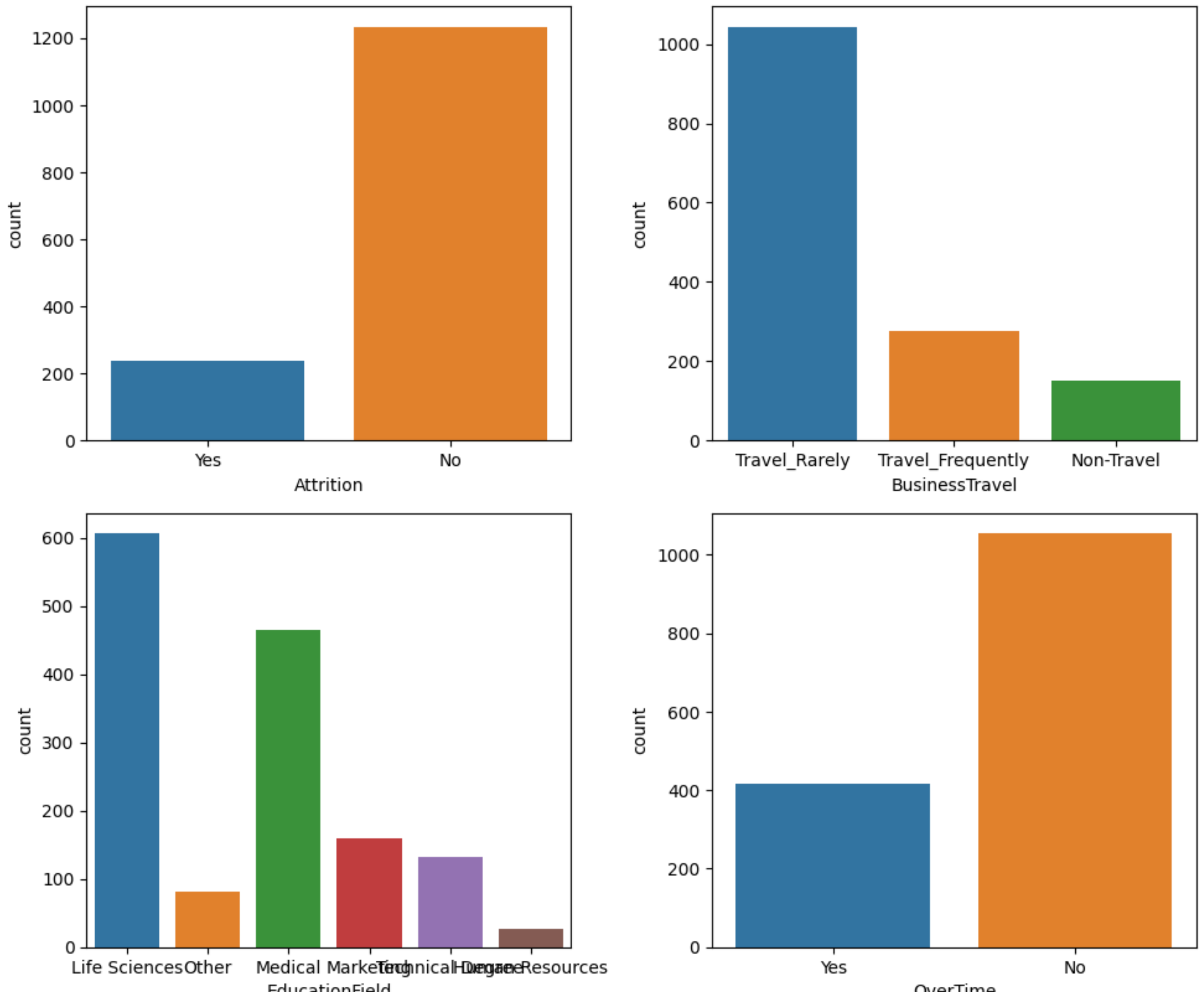
```
plt.subplots(figsize=(15,15))
sns.heatmap(d,annot=True)
```

```
<Axes: >
```



```
f = plt.figure()
f.set_figwidth(15)
f.set_figheight(12)

# Subplot 1
plt.subplot(3, 3, 1)
sns.countplot(x="Attrition", data=a)

# Subplot 2
plt.subplot(3, 3, 2)
sns.countplot(x="BusinessTravel", data=a)

# Subplot 5
plt.subplot(3, 3, 3)
sns.countplot(x="Department", data=a)

# Subplot 8
plt.subplot(3, 3, 4)
sns.countplot(x="EducationField", data=a)

# Subplot 9
plt.subplot(3, 3, 5)
sns.countplot(x="OverTime", data=a)

# Adjust layout
```

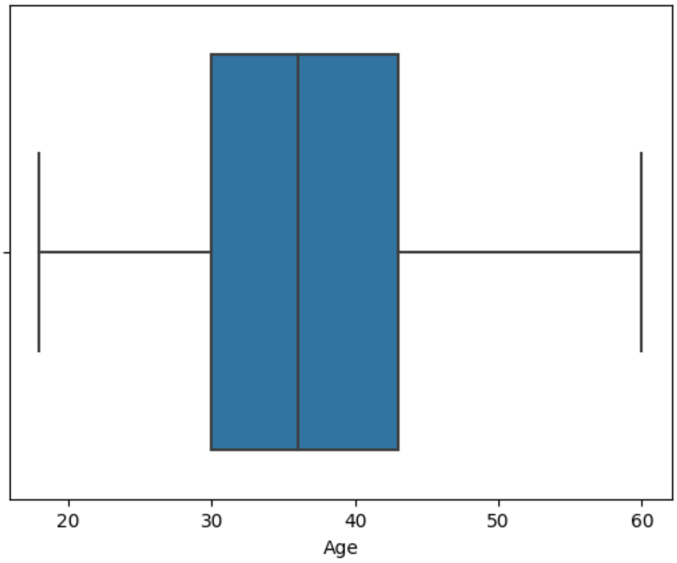
```
plt.tight_layout()

# Show the plots
plt.show()
```



Outlier Detection

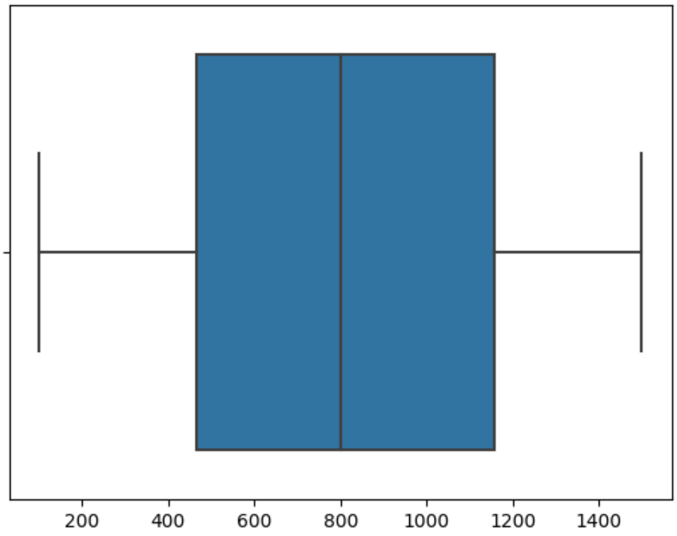
```
sns.boxplot(x="Age",data=a)
```

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



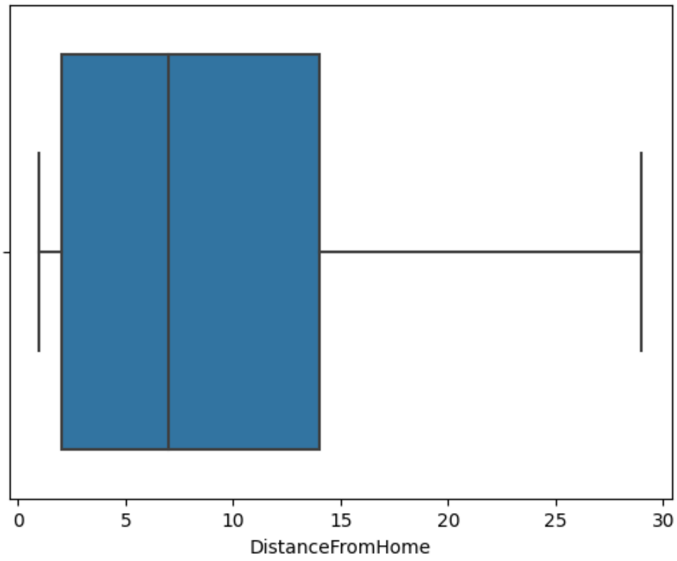
```
sns.boxplot(x="DailyRate",data=a)
```

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



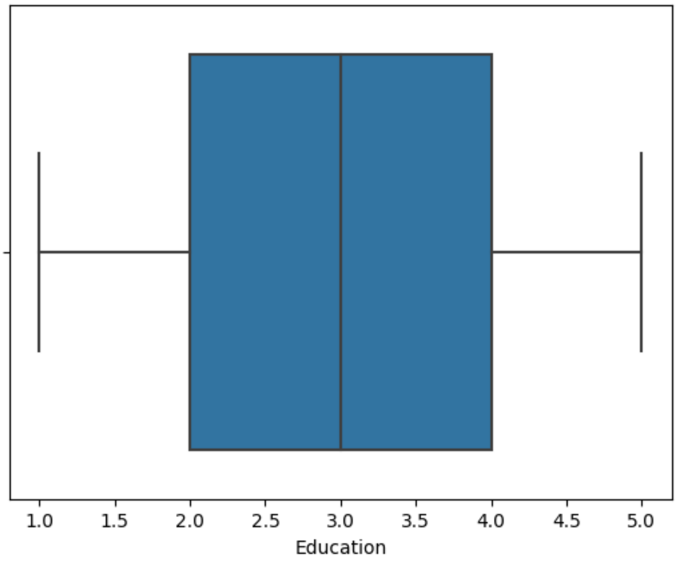
```
sns.boxplot(x="DistanceFromHome",data=a)
```

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



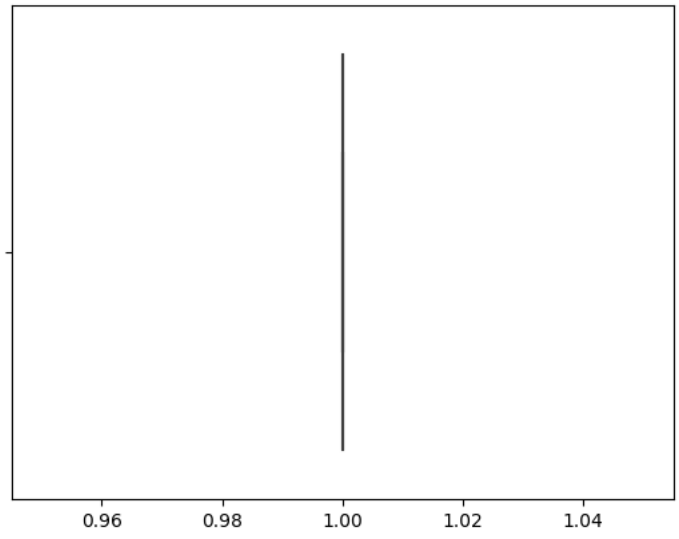
```
sns.boxplot(x="Education",data=a)
```

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



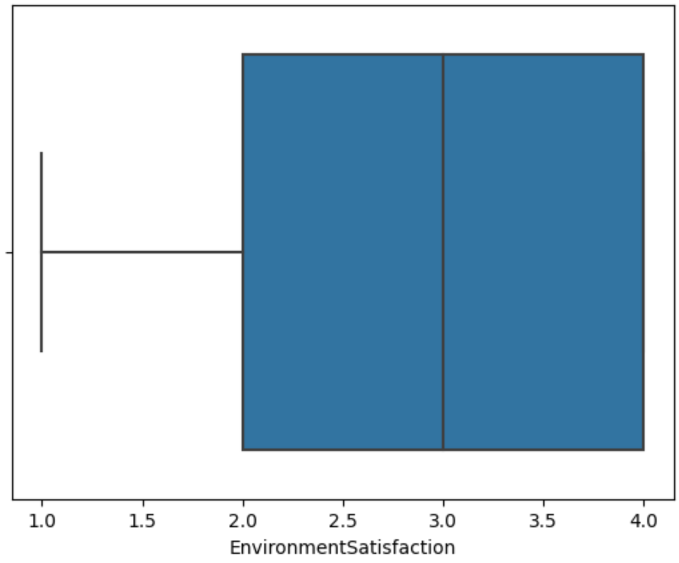
```
sns.boxplot(x="EmployeeCount",data=a)
```

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



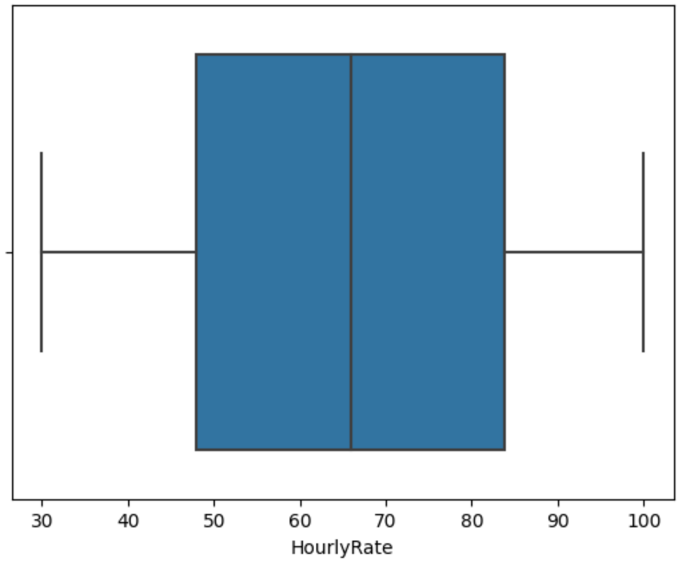
```
sns.boxplot(x="EnvironmentSatisfaction",data=a)
```

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



```
sns.boxplot(x="HourlyRate",data=a)
```

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



```
# there are no outliers , the data is clean
```

Splitting dependent and independent variables

```
x=a.drop(columns=["Attrition"],axis=1)
x.head()
```

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	Environme
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	

5 rows × 34 columns

```
x.shape
```

```
(1470, 34)
```

```
y=a["Attrition"]
y.head()
```

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

```
y.shape
```

```
(1470,)
```

Encoding

```
from sklearn.preprocessing import LabelEncoder
```

```
l=LabelEncoder()
```

```
x["Gender"]=l.fit_transform(x["Gender"])
x['Gender']
```

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

```
x['Gender'].value_counts()
```

```
1    882
0    588
Name: Gender, dtype: int64
```

```
x['Gender'].nunique()
```

```
2
```

```
x.head()
```

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	Environme
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	
				Research &						

```
Dept = pd.get_dummies(a, columns=["Department"])
print(Dept)
```

```
1466                        4  ...              9                    5
1467                        2  ...              6                    0
1468                        4  ...             17                    3
1469                        2  ...              6                    3

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

      YearsSinceLastPromotion YearsWithCurrManager  \
0                           0                    5
1                           1                    7
2                           0                    0
3                           3                    0
4                           2                    2
...                       ...                  ...
1465                        0                    3
1466                        1                    7
1467                        0                    3
1468                        0                    8
1469                        1                    2

      Department_Human Resources  Department_Research & Development  \
0                              0                                  0
1                              0                                  1
2                              0                                  1
3                              0                                  1
4                              0                                  1
...                          ...                                ...
1465                           0                                  1
1466                           0                                  1
1467                           0                                  1
1468                           0                                  0
1469                           0                                  1

      Department_Sales
0                    1
1                    0
2                    0
3                    0
4                    0
...                ...
1465                 0
1466                 0
1467                 0
1468                 1
1469                 0

[1470 rows x 37 columns]
```

```
print(x)
```

```
      Age     BusinessTravel  DailyRate              Department  \
0      41      Travel_Rarely       1102                   Sales
1      49  Travel_Frequently        279  Research & Development
2      37      Travel_Rarely       1373  Research & Development
3      33  Travel_Frequently       1392  Research & Development
4      27      Travel_Rarely        591  Research & Development
...   ...                ...        ...                     ...
1465   36  Travel_Frequently        884  Research & Development
1466   39      Travel_Rarely        613  Research & Development
1467   27      Travel_Rarely        155  Research & Development
1468   49  Travel_Frequently       1023                   Sales
```

```
1469   34      Travel_Rarely        628  Research & Development

      DistanceFromHome  Education EducationField  EmployeeCount  \
0                    1          2  Life Sciences              1
1                    8          1  Life Sciences              1
2                    2          2          Other              1
3                    3          4  Life Sciences              1
4                    2          1        Medical              1
...                ...        ...            ...            ...
1465                23          2        Medical              1
1466                 6          1        Medical              1
1467                 4          3  Life Sciences              1
1468                 2          3        Medical              1
1469                 8          3        Medical              1

      EmployeeNumber  EnvironmentSatisfaction  ...  RelationshipSatisfaction  \
0                  1                        2  ...                         1
1                  2                        3  ...                         4
2                  4                        4  ...                         2
3                  5                        4  ...                         3
4                  7                        1  ...                         4
...              ...                      ...  ...                       ...
1465            2061                        3  ...                         3
1466            2062                        4  ...                         1
1467            2064                        2  ...                         2
1468            2065                        4  ...                         4
1469            2068                        2  ...                         1

      StandardHours  StockOptionLevel  TotalWorkingYears  \
0                80                 0                  8
1                80                 1                 10
2                80                 0                  7
3                80                 0                  8
4                80                 1                  6
...             ...               ...                ...
1465             80                 1                 17
1466             80                 1                  9
1467             80                 1                  6
1468             80                 0                 17
1469             80                 0                  6

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

```
a.head()
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSat
0	41	Yes	Travel_Rarely	1102	Sales	1	2	1	1	
1	49	No	Travel_Frequently	279	Research & Development	8	1	1	2	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	1	4	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	1	5	
4	27	No	Travel_Rarely	591	Research & Development	2	1	1	7	

5 rows × 40 columns

```
x.head()
```

Age BusinessTravel DailyRate Department DistanceFromHome Education EducationField EmployeeCount EmployeeNumber Environme

```
Dept=pd.get_dummies(x["Department"],drop_first=True)
Dept
```

	Research & Development	Sales
0	0	1
1	1	0
2	1	0
3	1	0
4	1	0
...	...	...
1465	1	0
1466	1	0
1467	1	0
1468	0	1
1469	1	0

1470 rows × 2 columns

```
x=pd.concat([x,Dept],axis=1)
```

```
x.head()
```

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	Environme
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	

5 rows × 36 columns

Feature Scaling

```
from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()
```

```
X = a[['Age', 'MonthlyIncome', 'YearsAtCompany', 'JobSatisfaction', 'EnvironmentSatisfaction','YearsWithCurrManager','WorkLifeBalance']]
Y = a['Attrition']
```

```
X.head()
```

	Age	MonthlyIncome	YearsAtCompany	JobSatisfaction	EnvironmentSatisfaction	YearsWithCurrManager	WorkLifeBalance
0	41	5993	6	4	2	5	1
1	49	5130	10	2	3	7	3
2	37	2090	0	3	4	0	3
3	33	2909	8	3	4	0	3
4	27	3468	2	2	1	2	3

```
x.tail()
```

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	Enviro
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	

```
x
```

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	Enviro
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 × 36 columns

Splitting data into test and train

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
```

```
X_train,X_test,Y_train,Y_test.shape
```

```
(      Age  MonthlyIncome  YearsAtCompany  JobSatisfaction  \
 1097   24           2296               1                1
 727    18           1051               0                4
 254    29           6931               3                4
 1175   39           5295               5                2
 1341   31           4197              10                3
 ...   ...            ...             ...              ...
 1130   35           3407              10                3
 1294   41           6870               3                2
 860    22           2853               0                4
 1459   29           4025               4                2
 1126   50          19331               1                3

       EnvironmentSatisfaction  YearsWithCurrManager  WorkLifeBalance
 1097                        3                     0                3
 727                         2                     0                3
 254                         4                     2                3
 1175                        4                     0                3
 1341                        2                     2                3
 ...                       ...                   ...              ...
 1130                        2                     8                2
 1294                        2                     2                1
 860                         3                     0                3
 1459                        4                     3                3
 1126                        3                     0                3

 [1176 rows x 7 columns],
       Age  MonthlyIncome  YearsAtCompany  JobSatisfaction  \
 1041   28           8463               5                1
 184    53           4450               4                1
 1222   24           1555               1                3
 67     45           9724               1                1
 220    36           5914              13                2
 ...   ...            ...             ...              ...
 567    34           6274               6                4
 560    34           5121               0                1
 945    50          16880               3                1
 522    37           4680               1                4
 651    47           4537               7                4

       EnvironmentSatisfaction  YearsWithCurrManager  WorkLifeBalance
 1041                        4                     3                3
 184                         4                     3                3
 1222                        4                     0                3
 67                          2                     0                3
 220                         4                     7                4
 ...                       ...                   ...              ...
 567                         4                     4                3
 560                         2                     0                3
 945                         4                     2                3
 522                         4                     0                3
 651                         3                     7                3

 [294 rows x 7 columns],
 1097     No
 727      No
 254      No
 1175     No
```

Logistic Regression

Model Building & Import the model building Libraries

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

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

```
     ▾ LogisticRegression
     LogisticRegression()
```

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

```
pred
```

```
     array(['No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
            'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No'], dtype=object)
```

```
Y_test
```

```
     1041     No
     184      No
     1222    Yes
     67       No
     220      No
```

```
          ...
    567      No
    560      No
    945      No
    522      No
    651      No
    Name: Attrition, Length: 294, dtype: object
```

```
a
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	Environment
0	41	Yes	Travel_Rarely	1102	Sales	1	2	1	1	
1	49	No	Travel_Frequently	279	Research & Development	8	1	1	2	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	1	4	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	1	5	
4	27	No	Travel_Rarely	591	Research & Development	2	1	1	7	
...	...	...	...	...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research & Development	23	2	1	2061	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	1	2062	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	1	2064	
1468	49	No	Travel_Frequently	1023	Sales	2	3	1	2065	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	1	2068	

1470 rows × 40 columns

Evaluation of classification model

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

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

```
report = classification_report(Y_test, pred, zero_division=1)
```

```
print(f'Accuracy: {accuracy}')
print(f'Classification Report:\n{report}')
```

```
    Accuracy: 0.8673469387755102
    Classification Report:
                  precision    recall  f1-score   support

              No       0.87      1.00      0.93       255
             Yes       1.00      0.00      0.00        39

        accuracy                           0.87       294
       macro avg       0.93      0.50      0.46       294
    weighted avg       0.88      0.87      0.81       294
```

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

```
    array([[255,   0],
           [ 39,   0]])
```

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

```
col_0   No
```

Roc-AUC curve

```
No     255
```

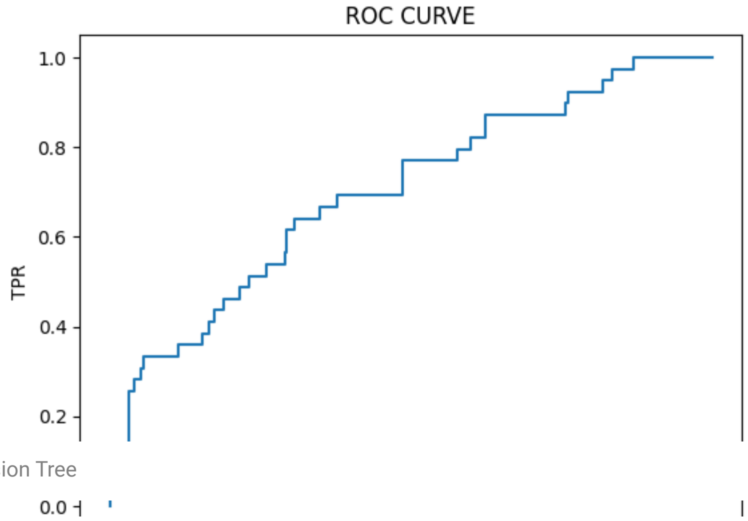
```
probability=model.predict_proba(X_test)[:,1]
```

```
probability
```

```
       0.14963007, 0.15969356, 0.20644099, 0.08193936, 0.18537088,
       0.16096129, 0.02189805, 0.15660552, 0.11782876, 0.18248771,
       0.13287268, 0.14334387, 0.0892007 , 0.06858367, 0.05708061,
       0.1753651 , 0.14395111, 0.10012064, 0.15057687, 0.2329628 ,
       0.03338823, 0.27116899, 0.15771847, 0.18762417, 0.10029771,
       0.10548668, 0.15048832, 0.12644386, 0.14778903, 0.2030313 ,
       0.06737083, 0.04935137, 0.35253675, 0.19926437, 0.23846212,
       0.08198467, 0.28864726, 0.23955634, 0.19282515, 0.22246873,
       0.11288909, 0.17545014, 0.24051176, 0.14059822, 0.32377579,
       0.08977525, 0.15148043, 0.01896052, 0.14635136, 0.20158982,
       0.10191406, 0.10573264, 0.08537077, 0.1631479 , 0.12443613,
       0.10510977, 0.33623452, 0.11027653, 0.05493965, 0.28005007,
       0.18450873, 0.12499531, 0.17197795, 0.17873294, 0.06110176,
       0.18127058, 0.08791989, 0.15005295, 0.15959692, 0.19866202,
       0.07388538, 0.19341696, 0.19100387, 0.08712656, 0.08033949,
       0.02928375, 0.13253218, 0.05956382, 0.16844953, 0.08753921,
       0.17957672, 0.12899389, 0.16872069, 0.16947305, 0.12397644,
       0.1099147 , 0.24576674, 0.07821105, 0.2716565 , 0.12140547,
       0.06524951, 0.1337184 , 0.14536957, 0.18726004, 0.10915274,
       0.04570312, 0.10169758, 0.07390408, 0.22704117, 0.07208355,
       0.08035364, 0.18593691, 0.16647288, 0.10818369, 0.05315879,
       0.17696614, 0.18973955, 0.22476227, 0.17342537, 0.21403334,
       0.16943373, 0.16771766, 0.09747364, 0.11387728, 0.2559594 ,
       0.32393512, 0.08431327, 0.13118746, 0.10751731, 0.09837008,
       0.25991497, 0.18954525, 0.11954205, 0.10534474, 0.09694665,
       0.07268098, 0.30507638, 0.06501248, 0.14080365, 0.1255734 ,
       0.11537899, 0.23299235, 0.17264787, 0.24765337, 0.06927027,
       0.21512755, 0.09901074, 0.16646941, 0.08047622, 0.03233445,
       0.15363939, 0.14131117, 0.25851265, 0.26761484, 0.1665985 ,
       0.10685997, 0.11549038, 0.19827264, 0.19076354, 0.13247131,
       0.26173972, 0.17180386, 0.21324175, 0.04115976, 0.15054569,
       0.16012435, 0.09434315, 0.09921354, 0.22000675, 0.06421677,
       0.16643204, 0.12016002, 0.14827189, 0.08450615, 0.05725373,
       0.12102272, 0.02681568, 0.18300015, 0.21076054, 0.11715199,
       0.16127828, 0.18483891, 0.09043029, 0.14086669, 0.20253644,
       0.0594472 , 0.10383826, 0.01617733, 0.15428555, 0.08595314,
       0.22434066, 0.11577713, 0.07998958, 0.07811109, 0.12006351,
       0.12845942, 0.14824842, 0.10405812, 0.19816497, 0.1162661 ,
       0.21477996, 0.24395257, 0.04972863, 0.2156586 , 0.16831872,
       0.17867722, 0.15398516, 0.21871738, 0.03416769, 0.07072713,
       0.22242289, 0.10244091, 0.10919764, 0.12517809, 0.0706504 ,
       0.07399615, 0.24438034, 0.17159597, 0.17617076, 0.10663942,
       0.13898632, 0.15178097, 0.10545546, 0.2723432 , 0.07462743,
       0.23465253, 0.26405405, 0.10124306, 0.3028089 , 0.12410107,
       0.1909214 , 0.20302625, 0.13276688, 0.0401135 , 0.18943046,
       0.23129363, 0.25951761, 0.08630086, 0.21347439, 0.20469075,
       0.13330949, 0.08581729, 0.10996842, 0.06690194, 0.04616928,
       0.18853288, 0.11542819, 0.21231547, 0.03597583, 0.07176025,
       0.17130681, 0.11593175, 0.23407496, 0.1533375 , 0.09696206,
       0.16256038, 0.06366454, 0.04689748, 0.0855508 , 0.23703024,
       0.07106702, 0.18067446, 0.2069784 , 0.22648723, 0.02715875,
       0.17170263, 0.14167865, 0.276632  , 0.10463943, 0.12037205,
       0.21133882, 0.02933273, 0.0973697 , 0.23466029, 0.23184945,
       0.1882965 , 0.04906958, 0.19036583, 0.1399965 , 0.11412922,
       0.22223015, 0.12517666, 0.24824295, 0.07113102, 0.07508479,
       0.14609486, 0.15491467, 0.18318556, 0.09382192, 0.04811606,
       0.20893659, 0.20088061, 0.23217748, 0.10747859, 0.11268901,
       0.25784861, 0.07464244, 0.1744561 , 0.09272658])
```

```
from sklearn.preprocessing import LabelBinarizer
lb = LabelBinarizer()
Y_test_bin = lb.fit_transform(Y_test)
fpr, tpr, thresholds = roc_curve(Y_test_bin, probability)
```

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



Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report
```

```
dt_model = DecisionTreeClassifier(random_state=50)
```

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

```
▾        DecisionTreeClassifier
DecisionTreeClassifier(random_state=50)
```

```
dt_predictions = dt_model.predict(X_test)
```

```
dt_accuracy = accuracy_score(Y_test, dt_predictions)
```

```
dt_report = classification_report(Y_test, dt_predictions)
```

```
print(f'Decision Tree Accuracy: {dt_accuracy}')
```

```
Decision Tree Accuracy: 0.7789115646258503
```

```
print(f'Decision Tree Classification Report:\n{dt_report}')
```

```
Decision Tree Classification Report:
              precision    recall  f1-score   support

          No       0.90      0.84      0.87       255
         Yes       0.28      0.41      0.33        39

    accuracy                           0.78       294
   macro avg       0.59      0.62      0.60       294
weighted avg       0.82      0.78      0.80       294
```

Random Forest Classifier

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

```
rf_model = RandomForestClassifier(random_state=50)
```

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

```
▾        RandomForestClassifier
RandomForestClassifier(random_state=50)
```

```
rf_predictions = rf_model.predict(X_test)
```

```
rf_accuracy = accuracy_score(Y_test, rf_predictions)
```

```
rf_report = classification_report(Y_test, rf_predictions)
```

```
print(f'Random Forest Accuracy: {rf_accuracy}')
```

```
Random Forest Accuracy: 0.8435374149659864
```

```
print(f'Random Forest Classification Report:\n{rf_report}')
```

```
Random Forest Classification Report:
              precision    recall  f1-score   support

          No       0.88      0.95      0.91       255
         Yes       0.33      0.18      0.23        39

    accuracy                           0.84       294
   macro avg       0.61      0.56      0.57       294
weighted avg       0.81      0.84      0.82       294
```