P J N D M PRAKASH

21BCE8942

ASSIGNMENT-4

AI AND ML MORNING SLOT

durga.21bce8942@vitapstudent.ac.in

Import necessary libraries

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

Import dataset

```
df=pd.read_csv("Employee_Attrition.csv")
```

```
df
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Educatio
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life S
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life S
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life S
4	27	No	Travel_Rarely	591	Research & Development	2	1	
...	...	...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research & Development	23	2	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life S
1468	49	No	Travel_Frequently	1023	Sales	2	3	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	

1470 rows × 35 columns

Information and statistics about dataset

```
df.info()
```

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

```
df.describe()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	Envir
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	

8 rows × 26 columns

```
df.shape
```

```
(1470, 35)
```

Check if any null values present in the dataset

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

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

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

```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
MonthlyRate                 0
NumCompaniesWorked          0
```

```
Over18                      0
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StandardHours               0
StockOptionLevel            0
TotalWorkingYears           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

** Data visualzation**

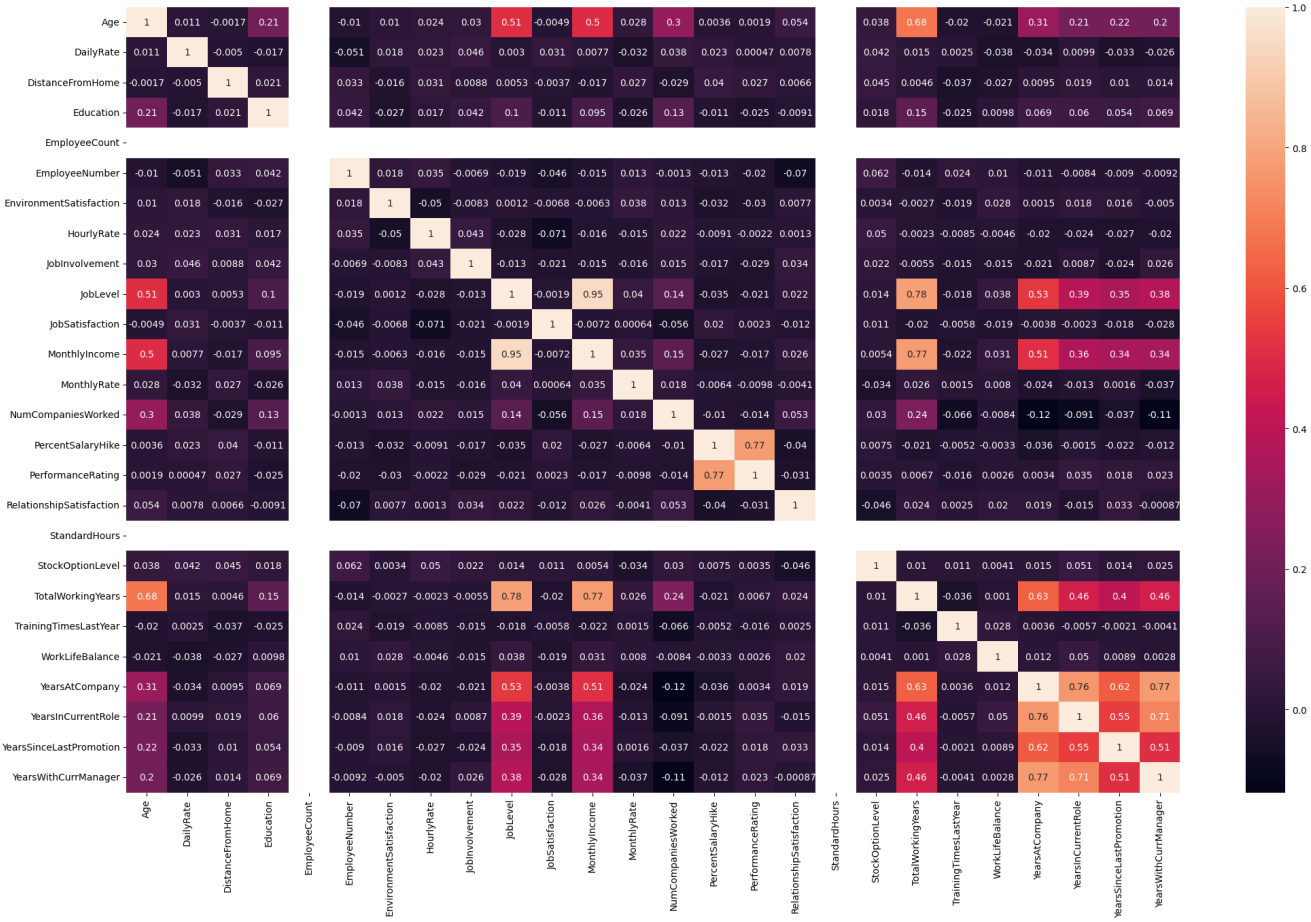
```
df.corr()
```

```
<ipython-input-10-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in DataFrame.co
  df.corr()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeN
Age	1.000000	0.010661	-0.001686	0.208034	NaN	-0.0
DailyRate	0.010661	1.000000	-0.004985	-0.016806	NaN	-0.0

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

```
<ipython-input-11-d92c41bf2ee0>:2: FutureWarning: The default value of numeric_only in DataFrame.co
  sns.heatmap(df.corr(),annot=True)
<Axes: >
```



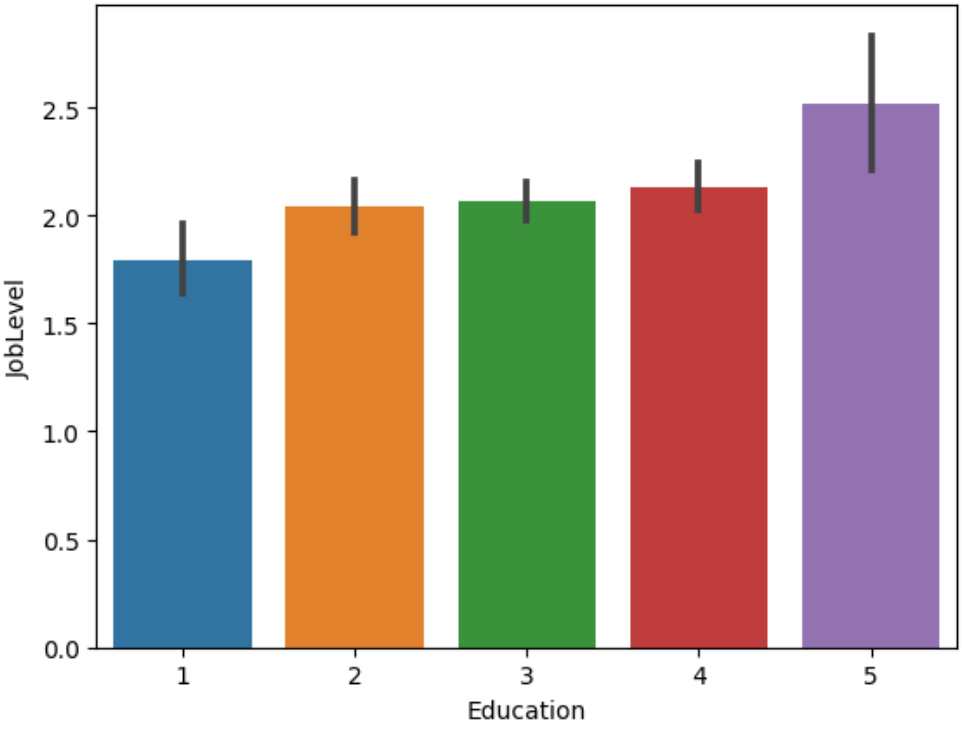
YearsWithCurrManager	0.202089	-0.026363	0.014406	0.069065	NaN	-0.0

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

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


```
sns.barplot(x="Education",y="JobLevel",data=df)
```

```
<Axes: xlabel='Education', ylabel='JobLevel'>
```



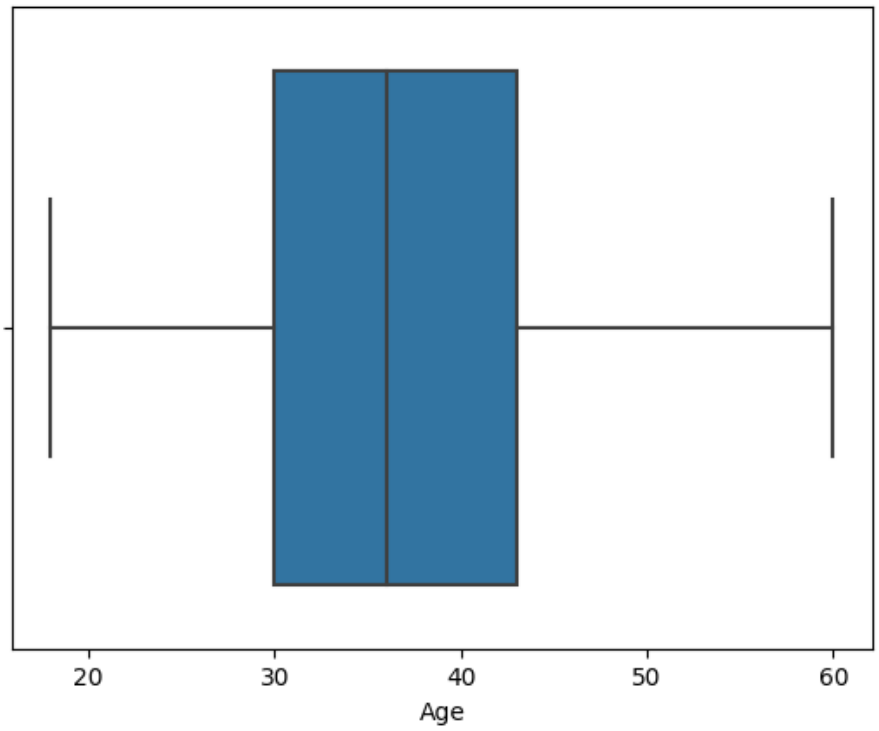

```
sns.pointplot(x="JobInvolvement",y="JobSatisfaction",data=df)
```

```
<Axes: xlabel='JobInvolvement', ylabel='JobSatisfaction'>
```

Outlier Detection

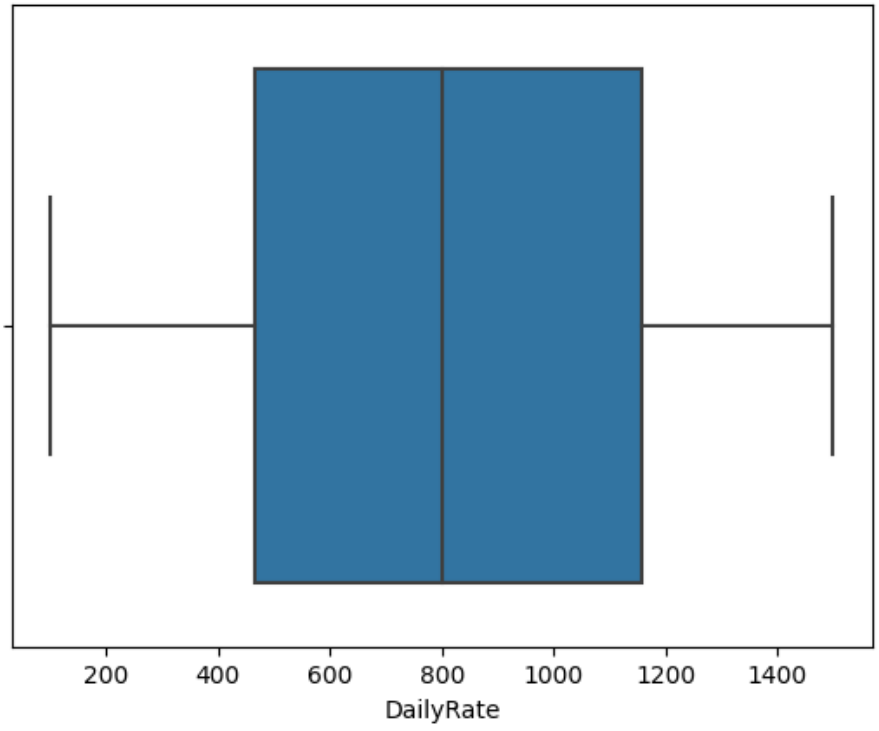
```
sns.boxplot(x="Age",data=df)
```

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



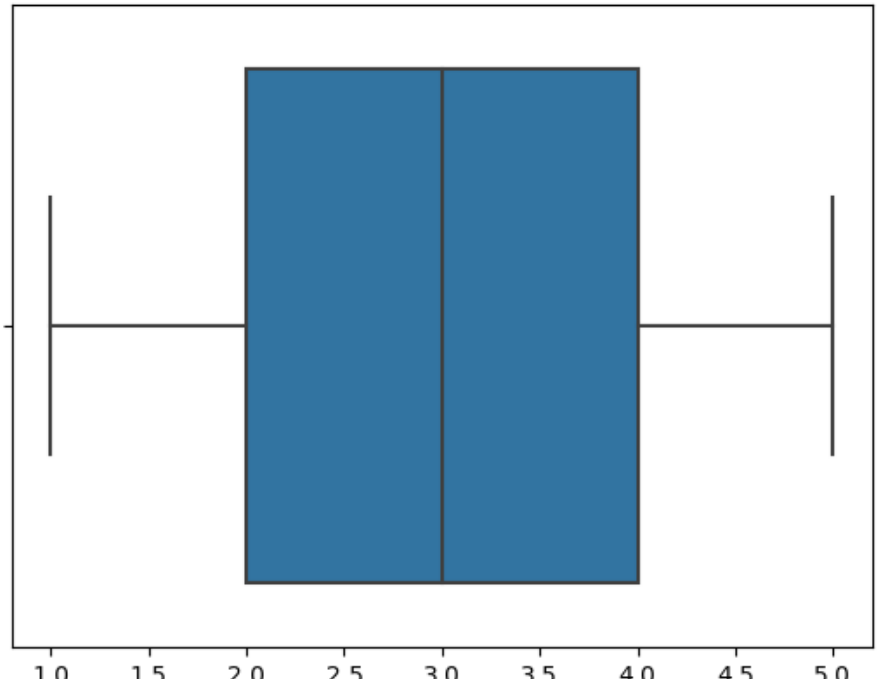

```
sns.boxplot(x="DailyRate",data=df)
```

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



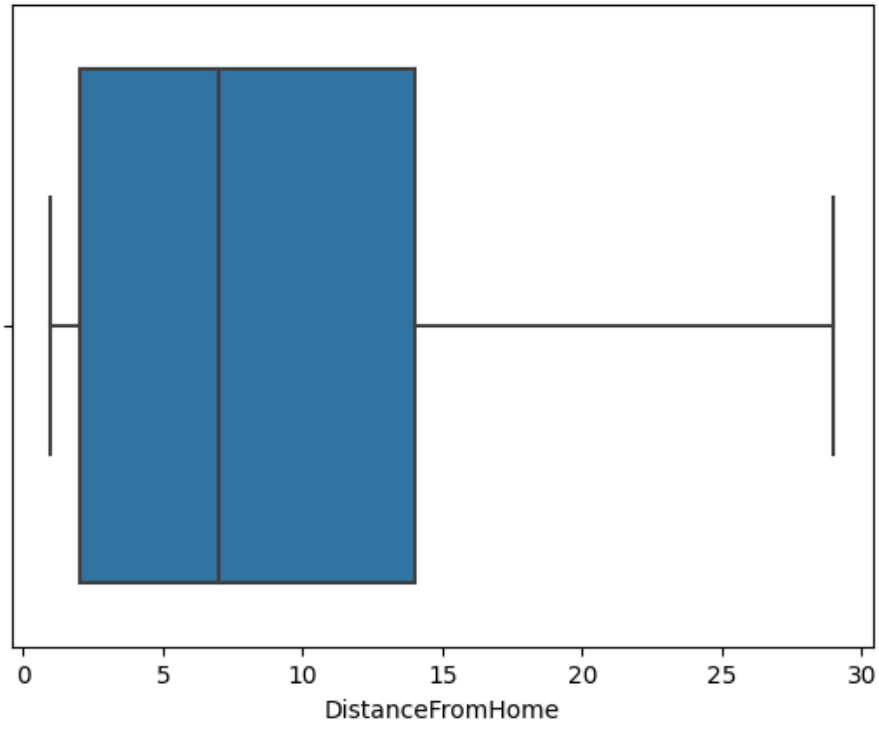

```
sns.boxplot(x="Education",data=df)
```

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



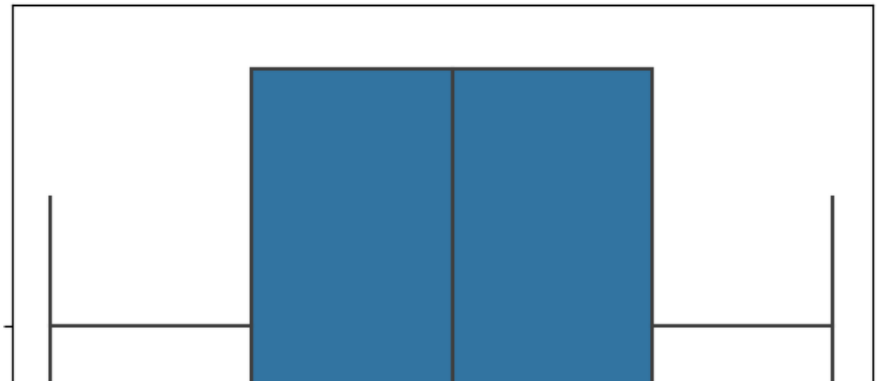

```
sns.boxplot(x="DistanceFromHome",data=df)
```

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



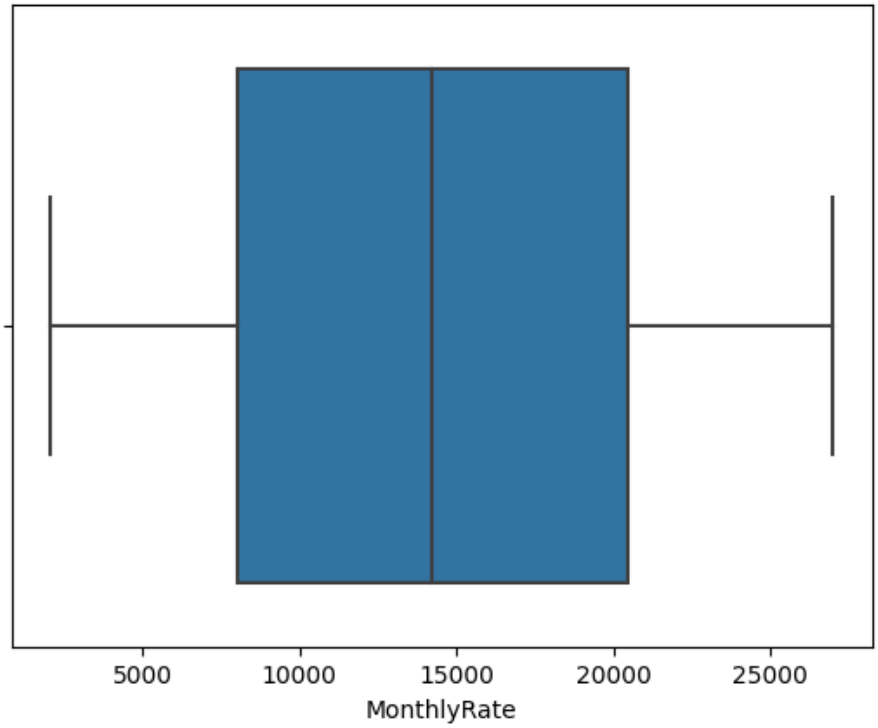

```
sns.boxplot(x="HourlyRate",data=df)
```

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



```
sns.boxplot(x="MonthlyRate",data=df)
```

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




```
sns.boxplot(x="JobSatisfaction",data=df)
```

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

There are no outliers in the data.

Splitting Dependent and Independent Variables

```
x=df[['Age','Gender','Department','EmployeeCount','EmployeeNumber' , 'EnvironmentSatisfaction','HourlyRat
```

```
x.head(10)
```

	Age	Gender	Department	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	Job
0	41	Female	Sales	1	1	2	94	
1	49	Male	Research & Development	1	2	3	61	
2	37	Male	Research & Development	1	4	4	92	
3	33	Female	Research & Development	1	5	4	56	
4	27	Male	Research & Development	1	7	1	40	
5	32	Male	Research & Development	1	8	4	79	
6	59	Female	Research & Development	1	10	3	81	
7	30	Male	Research & Development	1	11	4	67	
8	38	Male	Research & Development	1	12	4	44	
9	36	Male	Research & Development	1	13	3	94	

```
x.shape
```

```
(1470, 18)
```

```
y=df.Attrition
```

```
y.head(10)
```

```
0    Yes
1     No
2    Yes
3     No
4     No
5     No
6     No
7     No
8     No
9     No
Name: Attrition, dtype: object
```

```
y.shape
```

```
(1470,)
```

Encoding

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

```
x["Gender"] = le.fit_transform(x["Gender"])
```

```
<ipython-input-83-5f2403693ec8>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/index
  x["Gender"] = le.fit_transform(x["Gender"])
```

```
x
```

	Age	Gender	Department	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate
0	41	0	Sales	1	1	2	94
1	49	1	Research & Development	1	2	3	61
2	37	1	Research & Development	1	4	4	92
3	33	0	Research & Development	1	5	4	56
4	27	1	Research & Development	1	7	1	40
...	...	...	...	...	...	...	...
1465	36	1	Research & Development	1	2061	3	41
1466	39	1	Research & Development	1	2062	4	42
1467	27	1	Research & Development	1	2064	2	87
1468	49	1	Sales	1	2065	4	63
1469	34	1	Research & Development	1	2068	2	82

1470 rows × 18 columns

```
print(le.classes_)
```

```
['Female' 'Male']
```

```
le1=LabelEncoder()
```

```
x["Department"] = le1.fit_transform(x["Department"])
x.head()
```

```
<ipython-input-87-9d9230f200b4>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/index
  x["Department"] = le1.fit_transform(x["Department"])
```

	Age	Gender	Department	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	Job
0	41	0	2	1	1	2	94	
1	49	1	1	1	2	3	61	
2	37	1	1	1	4	4	92	
3	33	0	1	1	5	4	56	
4	27	1	1	1	7	1	40	

```
dict(zip(le1.classes_,range(len(le1.classes_))))
```

```
{'Human Resources': 0, 'Research & Development': 1, 'Sales': 2}
```

```
x.shape
```

```
(1470, 18)
```

Feature Scaling

```
from sklearn.preprocessing import StandardScaler
std=StandardScaler()
```

```
x_scaled=pd.DataFrame(std.fit_transform(x),columns=x.columns)
```

```
x_scaled
```

	Age	Gender	Department	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	Hourl
1	1.322365	0.816497	0.493817	0.0	1.699621	0.254625	0.2

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=0.3, random_state=42)
```

3	-0.429664	-1.224745	-0.493817	0.0	-1.694636	1.169781	-0.4

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

```
((1029, 18), (441, 18), (1029,), (441,))
```

1465	0.101159	0.816497	0.493817	0.0	1.721670	0.254625	1.2

```
X_train.head()
```

	Age	Gender	Department	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate
714	50	1	1	1	997	4	66
135	36	1	1	1	178	2	84
1271	21	1	2	1	1780	2	31
477	50	1	0	1	644	1	99
806	52	1	1	1	1118	2	87

Model Building

LOGISTIC REGRESSION

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

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

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning:
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

▾ 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', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
```

```
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', '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
        ...
1047     No
331      No
323     Yes
654      No
1110    Yes
Name: Attrition, Length: 441, dtype: object
```

```
x
```

	Age	Gender	Department	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate
0	41	0	2	1	1	2	94
1	49	1	1	1	2	3	61
2	37	1	1	1	4	4	92
3	33	0	1	1	5	4	56

Evalution Of Classification Model

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

```
accuracy_score(Y_test,pred)
```

```
0.8639455782312925
```

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

```
array([[379,   1],
       [ 59,   2]])
```

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

col_0	No	Yes
Attrition		
No	379	1
Yes	59	2

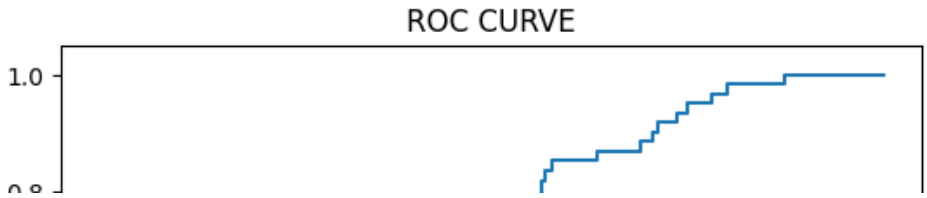
Roc-AUC curve for Logistic Regression

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

```
0.05371578, 0.02219113, 0.11955788, 0.19031586, 0.31650394,
0.15279411, 0.02729826, 0.12249097, 0.12075813, 0.09558155,
0.27679347, 0.11025601, 0.33044913, 0.06668938, 0.07795657,
0.11924351, 0.20871763, 0.39389321, 0.05694522, 0.0532045 ,
0.21490644, 0.25324812, 0.36315922, 0.05429224, 0.09960449,
0.25102764, 0.03669868, 0.19522668, 0.14519451, 0.35956027,
0.35581818, 0.09610838, 0.17997788, 0.09761989, 0.23488566,
0.03357664, 0.17046244, 0.37296518, 0.16225315, 0.34054488,
0.02426574, 0.13201919, 0.13899548, 0.12885304, 0.27099242,
0.12672625, 0.27652686, 0.20411269, 0.21120591, 0.02566868,
0.11040152, 0.02739579, 0.32812678, 0.0356868 , 0.0555287 ,
0.04246302, 0.03945161, 0.08572675, 0.13737275, 0.14899748,
0.08851646, 0.04640608, 0.08757664, 0.31435635, 0.11874548,
0.19312344, 0.19903991, 0.00434757, 0.01359597, 0.02896461,
0.21525374, 0.07043031, 0.04936885, 0.2555012 , 0.03367993,
0.15839323, 0.08941318, 0.04304069, 0.09092879, 0.41638816,
0.21080791, 0.25378281, 0.11505416, 0.12806476, 0.08431388,
0.24187452, 0.23523082, 0.05611581, 0.10727518, 0.1747152 ,
0.15892058, 0.31031297, 0.30196505, 0.05540075, 0.16610599,
0.26126156, 0.11849072, 0.12130789, 0.31989303, 0.30308456,
0.10829714, 0.32275972, 0.05964667, 0.22507908, 0.19373778,
0.09161063, 0.05019815, 0.0443391 , 0.10394145, 0.1993514 ,
0.37193017, 0.20830325, 0.02336753, 0.15002127, 0.21238251,
0.44973521, 0.28104863, 0.0945913 , 0.25606712, 0.08455771,
0.09952959, 0.4043509 , 0.10053673, 0.31357501, 0.1417907 ,
0.31489395, 0.08284036, 0.22872485, 0.17175544, 0.11778603,
0.20956943, 0.16501444, 0.28859664, 0.29058634, 0.03547303,
0.08896462, 0.26335701, 0.15721688, 0.26717469, 0.29359793,
0.04326776, 0.16511205, 0.11401487, 0.19270145, 0.17325272,
0.25608343, 0.12117371, 0.10788852, 0.05577113, 0.23356136,
0.31319097, 0.15116921, 0.04650322, 0.26054039, 0.15149241,
0.19499282, 0.14898636, 0.2600236 , 0.20077498, 0.06410256,
0.05802142, 0.06155586, 0.23733205, 0.03068845, 0.33748808,
0.12874656, 0.15462816, 0.13922545, 0.35578153, 0.11710508,
0.0611714 , 0.17872668, 0.15302434, 0.341877  , 0.05541642,
0.43956966])
```

```
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
dt=DecisionTreeClassifier()
```

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

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
pre=dt.predict(X_test)
pre
```

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

```
Y_test
```

```
1041     No
184      No
1222    Yes
67       No
220      No
        ...
1047     No
331      No
323     Yes
654      No
1110    Yes
Name: Attrition, Length: 441, dtype: object
```

```
accuracy_score(Y_test,pre)
```

```
0.7528344671201814
```

```
print(classification_report(Y_test,pre))
```

```
                  precision    recall  f1-score   support

              No       0.87      0.84      0.85       380
             Yes       0.18      0.21      0.19        61

        accuracy                           0.75       441
       macro avg       0.52      0.53      0.52       441
    weighted avg       0.77      0.75      0.76       441
```

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

col_0	No	Yes
Attrition		
No	319	61
Yes	48	13

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

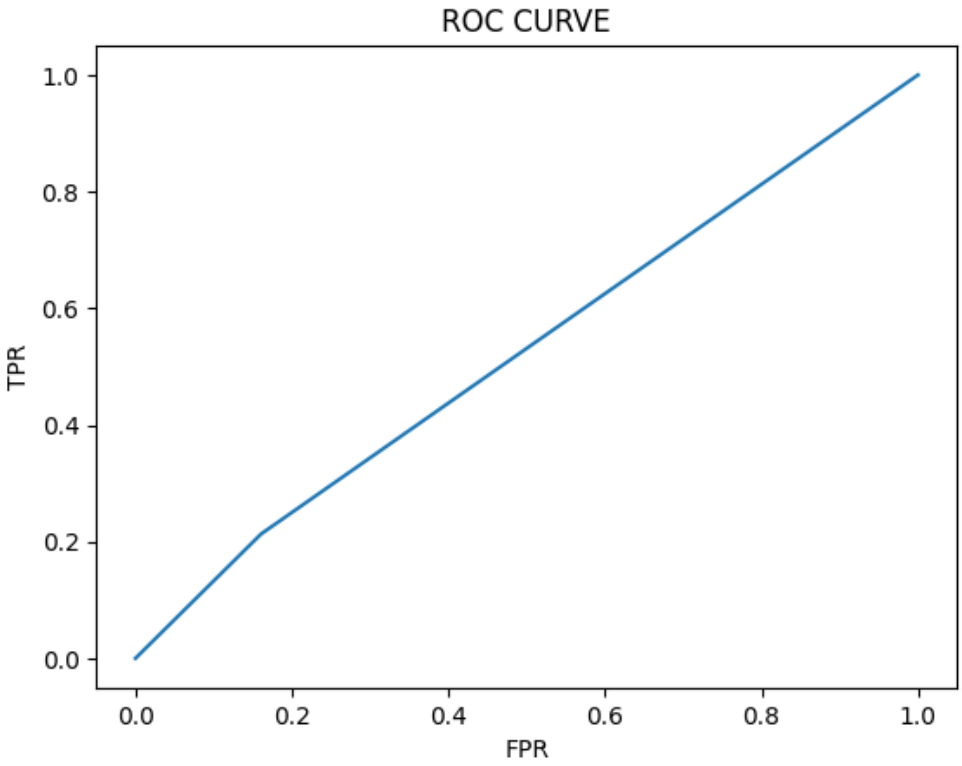
```
array([0., 0., 1., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 1., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 1., 0., 0., 0., 1., 0., 0., 0., 1., 0., 1., 0., 1., 0.,
       0., 0., 1., 0., 0., 0., 0., 1., 0., 0., 1., 0., 0., 0., 0., 0., 0.,
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

```
0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1.,
0., 0., 0., 1., 1., 1., 1., 0., 0., 0., 0., 1., 0., 0., 0., 0., 1.,
0., 0., 0., 1., 0., 1., 0., 0., 0., 0., 0., 1., 0., 0., 0., 1., 0.,
0., 0., 0., 0., 1., 0., 1., 0., 1., 0., 0., 0., 0., 1., 0., 0.])
```

ROC - CURVE for Decision tree

```
Y_test_bin2 = lb.fit_transform(Y_test)
fpr1,tpr1,threshsholds = roc_curve(Y_test_bin2,prob)
```

```
plt.plot(fpr1,tpr1)
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('ROC CURVE')
plt.show()
```



Parameter Tuning

```
from sklearn import tree
plt.figure(figsize=(25,15))
tree.plot_tree(dt, filled=True)
```

```
 Text(0.14697286012526095, 0.7941176470588235, 'x[14] <= 1.5\ngini = 0.426\nsamples = 39\nvalue
= [27, 12]'),
 Text(0.1336116910229645, 0.7352941176470589, 'x[15] <= 0.5\ngini = 0.499\nsamples = 19\nvalue =
[10, 9]'),
 Text(0.1269311064718163, 0.6764705882352942, 'x[12] <= 2.5\ngini = 0.444\nsamples = 15\nvalue =
[10, 5]'),
 Text(0.12025052192066805, 0.6176470588235294, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.1336116910229645, 0.6176470588235294, 'x[11] <= 2372.5\ngini = 0.355\nsamples =
13\nvalue = [10, 3]'),
 Text(0.1269311064718163, 0.5588235294117647, 'x[1] <= 0.5\ngini = 0.278\nsamples = 12\nvalue =
[10, 2]'),
 Text(0.12025052192066805, 0.5, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.1336116910229645, 0.5, 'x[9] <= 17.0\ngini = 0.444\nsamples = 6\nvalue = [4, 2]'),
 Text(0.1269311064718163, 0.4411764705882353, 'x[11] <= 2217.5\ngini = 0.444\nsamples = 3\nvalue
= [1, 2]'),
 Text(0.12025052192066805, 0.38235294117647056, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.1336116910229645, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.14029227557411272, 0.4411764705882353, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.14029227557411272, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.14029227557411272, 0.6764705882352942, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.1603340292275574, 0.7352941176470589, 'x[16] <= 7.5\ngini = 0.255\nsamples = 20\nvalue =
[17, 3]'),
```

```
 Text(0.15365344467640918, 0.6764705882352942, 'x[0] <= 20.5\ngini = 0.188\nsamples = 19\nvalue
= [17, 2]'),
 Text(0.14697286012526095, 0.6176470588235294, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.1603340292275574, 0.6176470588235294, 'x[2] <= 1.5\ngini = 0.105\nsamples = 18\nvalue =
[17, 1]'),
 Text(0.15365344467640918, 0.5588235294117647, 'gini = 0.0\nsamples = 15\nvalue = [15, 0]'),
 Text(0.16701461377870563, 0.5588235294117647, 'x[9] <= 12.0\ngini = 0.444\nsamples = 3\nvalue =
[2, 1]'),
 Text(0.1603340292275574, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.17369519832985386, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.16701461377870563, 0.6764705882352942, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.19373695198329854, 0.8529411764705882, 'x[12] <= 1.5\ngini = 0.289\nsamples = 57\nvalue
= [47, 10]'),
 Text(0.1803757828810021, 0.7941176470588235, 'x[4] <= 569.0\ngini = 0.444\nsamples = 6\nvalue =
[2, 4]'),
 Text(0.17369519832985386, 0.7352941176470589, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.18705636743215032, 0.7352941176470589, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]'),
 Text(0.207098121085595, 0.7941176470588235, 'x[4] <= 45.5\ngini = 0.208\nsamples = 51\nvalue =
[45, 6]'),
 Text(0.20041753653444677, 0.7352941176470589, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.21377870563674323, 0.7352941176470589, 'x[7] <= 1.5\ngini = 0.15\nsamples = 49\nvalue =
[45, 4]'),
 Text(0.207098121085595, 0.6764705882352942, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.22045929018789143, 0.6764705882352942, 'x[6] <= 98.5\ngini = 0.117\nsamples = 48\nvalue
= [45, 3]'),
 Text(0.21377870563674323, 0.6176470588235294, 'x[11] <= 2436.5\ngini = 0.081\nsamples =
47\nvalue = [45, 2]'),
 Text(0.20041753653444677, 0.5588235294117647, 'x[17] <= 7.5\ngini = 0.043\nsamples = 45\nvalue
= [44, 1]'),
 Text(0.19373695198329854, 0.5, 'gini = 0.0\nsamples = 43\nvalue = [43, 0]'),
 Text(0.207098121085595, 0.5, 'x[6] <= 73.0\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.20041753653444677, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.21377870563674323, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.22713987473903965, 0.5588235294117647, 'x[5] <= 2.0\ngini = 0.5\nsamples = 2\nvalue =
[1, 1]'),
 Text(0.22045929018789143, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.23382045929018788, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.22713987473903965, 0.6176470588235294, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5618704331941545, 0.9117647058823529, 'x[16] <= 2.5\ngini = 0.226\nsamples = 868\nvalue
= [755, 113]'),
 Text(0.38538622129436323, 0.8529411764705882, 'x[0] <= 33.5\ngini = 0.378\nsamples = 166\nvalue
= [124, 42]'),
 Text(0.32317327766179543, 0.7941176470588235, 'x[14] <= 0.5\ngini = 0.493\nsamples = 68\nvalue
= [38, 30]'),
 Text(0.26555323590814195, 0.7352941176470589, 'x[4] <= 861.5\ngini = 0.482\nsamples = 32\nvalue
= [13, 19]'),
 Text(0.24718162839248434, 0.6764705882352942, 'x[12] <= 1.5\ngini = 0.375\nsamples = 8\nvalue =
[6, 2]'),
 Text(0.2405010438413361, 0.6176470588235294, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.2538622129436326, 0.6176470588235294, 'x[15] <= 0.5\ngini = 0.245\nsamples = 7\nvalue =
[6, 1]'),
 Text(0.24718162839248434, 0.5588235294117647, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.2605427974947808, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.2839248434237996, 0.6764705882352942, 'x[5] <= 1.5\ngini = 0.413\nsamples = 24\nvalue =
[7, 17]'),
 Text(0.27724425887265136, 0.6176470588235294, 'gini = 0.0\nsamples = 7\nvalue = [0, 7]'),
 Text(0.2906054279749478, 0.6176470588235294, 'x[11] <= 2794.5\ngini = 0.484\nsamples =
17\nvalue = [7, 10]'),
 Text(0.2739039665970772, 0.5588235294117647, 'x[7] <= 3.5\ngini = 0.245\nsamples = 7\nvalue =
[1, 6]'),
 Text(0.267223382045929, 0.5, 'gini = 0.0\nsamples = 6\nvalue = [0, 6]'),
 Text(0.28058455114822545, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.30730688935281836, 0.5588235294117647, 'x[9] <= 11.5\ngini = 0.48\nsamples = 10\nvalue =
[6, 4]'),
 Text(0.2939457202505219, 0.5, 'x[2] <= 0.5\ngini = 0.375\nsamples = 4\nvalue = [1, 3]'),
 Text(0.2872651356993737, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.30062630480167013, 0.4411764705882353, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.3206680584551148, 0.5, 'x[2] <= 0.5\ngini = 0.278\nsamples = 6\nvalue = [5, 1]'),
 Text(0.3139874739039666, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.32734864300626304, 0.4411764705882353, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
```

```
 Text(0.38079331941544886, 0.7352941176470589, 'x[11] <= 4340.0\ngini = 0.424\nsamples =
36\nvalue = [25, 11]'),
 Text(0.35407098121085595, 0.6764705882352942, 'x[11] <= 2549.0\ngini = 0.311\nsamples =
26\nvalue = [21, 5]'),
 Text(0.3407098121085595, 0.6176470588235294, 'x[9] <= 16.5\ngini = 0.444\nsamples = 3\nvalue =
[1, 2]'),
 Text(0.33402922755741127, 0.5588235294117647, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.3473903966597077, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.3674321503131524, 0.6176470588235294, 'x[6] <= 98.0\ngini = 0.227\nsamples = 23\nvalue =
[20, 3]'),
 Text(0.3607515657620042, 0.5588235294117647, 'x[7] <= 1.5\ngini = 0.165\nsamples = 22\nvalue =
[20, 2]'),
 Text(0.3473903966597077, 0.5, 'x[10] <= 3.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.3407098121085595, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.35407098121085595, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.37411273486430063, 0.5, 'x[11] <= 2633.5\ngini = 0.095\nsamples = 20\nvalue = [19, 1]'),
 Text(0.3674321503131524, 0.4411764705882353, 'x[15] <= 1.0\ngini = 0.444\nsamples = 3\nvalue =
[2, 1]'),
 Text(0.3607515657620042, 0.38235294117647056, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.37411273486430063, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.38079331941544886, 0.4411764705882353, 'gini = 0.0\nsamples = 17\nvalue = [17, 0]'),
 Text(0.37411273486430063, 0.5588235294117647, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.40751565762004177, 0.6764705882352942, 'x[5] <= 3.5\ngini = 0.48\nsamples = 10\nvalue =
[4, 6]'),
 Text(0.40083507306889354, 0.6176470588235294, 'x[4] <= 528.5\ngini = 0.375\nsamples = 8\nvalue
= [2, 6]'),
 Text(0.3941544885177453, 0.5588235294117647, 'x[4] <= 179.0\ngini = 0.444\nsamples = 3\nvalue =
[2, 1]'),
 Text(0.3874739039665971, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.40083507306889354, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.40751565762004177, 0.5588235294117647, 'gini = 0.0\nsamples = 5\nvalue = [0, 5]'),
 Text(0.41419624217119, 0.6176470588235294, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.4475991649269311, 0.7941176470588235, 'x[8] <= 3.5\ngini = 0.215\nsamples = 98\nvalue =
[86, 12]'),
 Text(0.44091858037578285, 0.7352941176470589, 'x[13] <= 2.5\ngini = 0.312\nsamples = 62\nvalue
= [50, 12]'),
 Text(0.4342379958246347, 0.6764705882352942, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.4475991649269311, 0.6764705882352942, 'x[5] <= 1.5\ngini = 0.278\nsamples = 60\nvalue =
[50, 10]'),
 Text(0.42755741127348645, 0.6176470588235294, 'x[11] <= 10487.5\ngini = 0.457\nsamples =
17\nvalue = [11, 6]'),
 Text(0.4208768267223382, 0.5588235294117647, 'x[4] <= 708.0\ngini = 0.496\nsamples = 11\nvalue
= [5, 6]'),
 Text(0.41419624217119, 0.5, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.42755741127348645, 0.5, 'x[4] <= 1037.0\ngini = 0.469\nsamples = 8\nvalue = [5, 3]'),
 Text(0.4208768267223382, 0.4411764705882353, 'gini = 0.0\nsamples = 3\nvalue = [3, 0]'),
 Text(0.4342379958246347, 0.4411764705882353, 'x[2] <= 1.5\ngini = 0.48\nsamples = 5\nvalue =
[2, 3]'),
 Text(0.42755741127348645, 0.38235294117647056, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.44091858037578285, 0.38235294117647056, 'x[0] <= 48.5\ngini = 0.444\nsamples = 3\nvalue
= [2, 1]'),
 Text(0.4342379958246347, 0.3235294117647059, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.4475991649269311, 0.3235294117647059, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.4342379958246347, 0.5588235294117647, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.46764091858037576, 0.6176470588235294, 'x[8] <= 2.5\ngini = 0.169\nsamples = 43\nvalue =
[39, 4]'),
 Text(0.4475991649269311, 0.5588235294117647, 'x[13] <= 16.5\ngini = 0.071\nsamples = 27\nvalue
= [26, 1]'),
 Text(0.44091858037578285, 0.5, 'gini = 0.0\nsamples = 20\nvalue = [20, 0]'),
 Text(0.4542797494780793, 0.5, 'x[6] <= 65.0\ngini = 0.245\nsamples = 7\nvalue = [6, 1]'),
 Text(0.4475991649269311, 0.4411764705882353, 'gini = 0.0\nsamples = 6\nvalue = [6, 0]'),
 Text(0.46096033402922754, 0.4411764705882353, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.48768267223382045, 0.5588235294117647, 'x[7] <= 3.5\ngini = 0.305\nsamples = 16\nvalue =
[13, 3]'),
 Text(0.4810020876826722, 0.5, 'x[2] <= 1.5\ngini = 0.231\nsamples = 15\nvalue = [13, 2]'),
 Text(0.474321503131524, 0.4411764705882353, 'gini = 0.0\nsamples = 11\nvalue = [11, 0]'),
 Text(0.48768267223382045, 0.4411764705882353, 'x[5] <= 3.5\ngini = 0.5\nsamples = 4\nvalue =
[2, 2]'),
 Text(0.4810020876826722, 0.38235294117647056, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
```

```
 Text(0.4943632567849687, 0.38235294117647056, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.4943632567849687, 0.5, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.4542797494780793, 0.7352941176470589, 'gini = 0.0\nsamples = 36\nvalue = [36, 0]'),
 Text(0.7383546450939458, 0.8529411764705882, 'x[2] <= 1.5\ngini = 0.182\nsamples = 702\nvalue =
[631, 71]'),
 Text(0.6142875782881002, 0.7941176470588235, 'x[13] <= 38.5\ngini = 0.134\nsamples = 471\nvalue
= [437, 34]'),
 Text(0.6076069937369519, 0.7352941176470589, 'x[8] <= 1.5\ngini = 0.131\nsamples = 470\nvalue =
[437, 33]'),
 Text(0.5319415448851774, 0.6764705882352942, 'x[4] <= 51.5\ngini = 0.213\nsamples = 99\nvalue =
[87, 12]'),
 Text(0.5252609603340292, 0.6176470588235294, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.5386221294363257, 0.6176470588235294, 'x[2] <= 0.5\ngini = 0.185\nsamples = 97\nvalue =
[87, 10]'),
 Text(0.5177453027139874, 0.5588235294117647, 'x[4] <= 1164.5\ngini = 0.5\nsamples = 4\nvalue =
[2, 2]'),
 Text(0.5110647181628393, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.5244258872651357, 0.5, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.5594989561586639, 0.5588235294117647, 'x[0] <= 54.5\ngini = 0.157\nsamples = 93\nvalue =
[85, 8]'),
 Text(0.5377870563674322, 0.5, 'x[14] <= 3.5\ngini = 0.127\nsamples = 88\nvalue = [82, 6]'),
 Text(0.5144050104384134, 0.4411764705882353, 'x[17] <= 2.5\ngini = 0.245\nsamples = 35\nvalue =
[30, 5]'),
 Text(0.5077244258872652, 0.38235294117647056, 'gini = 0.0\nsamples = 20\nvalue = [20, 0]'),
 Text(0.5210855949895616, 0.38235294117647056, 'x[6] <= 87.5\ngini = 0.444\nsamples = 15\nvalue
= [10, 5]'),
 Text(0.5077244258872652, 0.3235294117647059, 'x[12] <= 1.5\ngini = 0.198\nsamples = 9\nvalue =
[8, 1]'),
 Text(0.5010438413361169, 0.2647058823529412, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5144050104384134, 0.2647058823529412, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(0.534446764091858, 0.3235294117647059, 'x[16] <= 4.5\ngini = 0.444\nsamples = 6\nvalue =
[2, 4]'),
 Text(0.5277661795407098, 0.2647058823529412, 'x[11] <= 4216.0\ngini = 0.444\nsamples = 3\nvalue
= [2, 1]'),
 Text(0.5210855949895616, 0.20588235294117646, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.534446764091858, 0.20588235294117646, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5411273486430063, 0.2647058823529412, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.5611691022964509, 0.4411764705882353, 'x[0] <= 28.5\ngini = 0.037\nsamples = 53\nvalue =
[52, 1]'),
 Text(0.5544885177453027, 0.38235294117647056, 'x[5] <= 2.5\ngini = 0.198\nsamples = 9\nvalue =
[8, 1]'),
 Text(0.5478079331941544, 0.3235294117647059, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.5611691022964509, 0.3235294117647059, 'gini = 0.0\nsamples = 8\nvalue = [8, 0]'),
 Text(0.5678496868475992, 0.38235294117647056, 'gini = 0.0\nsamples = 44\nvalue = [44, 0]'),
 Text(0.5812108559498956, 0.5, 'x[17] <= 3.0\ngini = 0.48\nsamples = 5\nvalue = [3, 2]'),
 Text(0.5745302713987473, 0.4411764705882353, 'gini = 0.0\nsamples = 2\nvalue = [2, 0]'),
 Text(0.5878914405010438, 0.4411764705882353, 'x[4] <= 1391.5\ngini = 0.444\nsamples = 3\nvalue
= [1, 2]'),
 Text(0.5812108559498956, 0.38235294117647056, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.5945720250521921, 0.38235294117647056, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.6832724425887265, 0.6764705882352942, 'x[0] <= 23.5\ngini = 0.107\nsamples = 371\nvalue
= [350, 21]'),
 Text(0.6556367432150313, 0.6176470588235294, 'x[11] <= 3642.0\ngini = 0.444\nsamples = 6\nvalue
= [4, 2]'),
 Text(0.6489561586638831, 0.5588235294117647, 'gini = 0.0\nsamples = 4\nvalue = [4, 0]'),
 Text(0.6623173277661796, 0.5588235294117647, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.7109081419624217, 0.6176470588235294, 'x[15] <= 5.5\ngini = 0.099\nsamples = 365\nvalue
= [346, 19]'),
 Text(0.675678496868476, 0.5588235294117647, 'x[16] <= 32.0\ngini = 0.072\nsamples = 296\nvalue
= [285, 11]'),
 Text(0.648643006263048, 0.5, 'x[7] <= 1.5\ngini = 0.066\nsamples = 294\nvalue = [284, 10]'),
 Text(0.6146137787056367, 0.4411764705882353, 'x[8] <= 3.5\ngini = 0.305\nsamples = 16\nvalue =
[13, 3]'),
 Text(0.6079331941544885, 0.38235294117647056, 'gini = 0.0\nsamples = 10\nvalue = [10, 0]'),
 Text(0.621294363256785, 0.38235294117647056, 'x[9] <= 16.5\ngini = 0.5\nsamples = 6\nvalue =
[3, 3]'),
 Text(0.6146137787056367, 0.3235294117647059, 'x[5] <= 3.0\ngini = 0.375\nsamples = 4\nvalue =
[1, 3]'),
 Text(0.6079331941544885, 0.2647058823529412, 'gini = 0.0\nsamples = 3\nvalue = [0, 3]'),
 Text(0.621294363256785, 0.2647058823529412, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
```