# Importing the libraries

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Importing the dataset

ds=pd.read\_excel("data.xls")

from sklearn.preprocessing import LabelEncoder

LE= LabelEncoder()

ds["Attrition"]=LE.fit\_transform(ds["Attrition"])

ds["BusinessTravel"]=LE.fit\_transform(ds["BusinessTravel"])

ds["Department"]=LE.fit\_transform(ds["Department"])

ds["EducationField"]=LE.fit\_transform(ds["EducationField"])

ds["Gender"]=LE.fit\_transform(ds["Gender"])

ds["JobRole"]=LE.fit\_transform(ds["JobRole"])

ds["MaritalStatus"]=LE.fit\_transform(ds["MaritalStatus"])

ds["OverTime"]=LE.fit\_transform(ds["OverTime"])

X= ds.drop(["Attrition"],axis=1)

y= ds["Attrition"]

# Splitting the dataset into the Training set and Test set

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 = 0)

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

# Fitting Decision Tree Classification to the Training set

from sklearn.tree import DecisionTreeClassifier

classifier.fit(X\_train, y\_train)Out[223]:

DecisionTreeClassifier(ccp\_alpha=0.0, class\_weight='balanced',

criterion='entropy', max\_depth=None, max\_features=None,

max\_leaf\_nodes=None, min\_impurity\_decrease=0.0,

min\_impurity\_split=None, min\_samples\_leaf=1,

min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0,

presort='deprecated', random\_state=0, splitter='best')

# Predicting the Test set results

y\_pred = classifier.predict(X\_test)

# Making the Confusion Matrix

from sklearn.metrics import accuracy\_score,confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

cm

Out[229]:

array([[329, 42],

[ 47, 23]], dtype=int64)

from sklearn.metrics import classification\_report

print(classification\_report(y\_test, y\_pred))

precision recall f1-score support

0 0.88 0.89 0.88 371

1 0.35 0.33 0.34 70

accuracy 0.80 441

macro avg 0.61 0.61 0.61 441

weighted avg 0.79 0.80 0.80 441

accuracy\_sc=accuracy\_score(y\_test,y\_pred)

print(accuracy\_sc)

0.7981859410430839

#roc curve

from sklearn.metrics import roc\_auc\_score

from sklearn.metrics import roc\_curve

fpr, tpr, threshold = roc\_curve(y\_test, classifier.predict\_proba(X\_test)[:,1])

log\_roc\_auc2= roc\_auc\_score(y\_test, y\_pred)

print(log\_roc\_auc2)

0.6076819407008086

plt.figure()

plt.plot(fpr, tpr, label = 'Decission Tree (area = %0.2f)' %log\_roc\_auc2)

plt.plot([0,1], [0,1], 'r--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('false positive rate')

plt.ylabel('true positive rate')

plt.title('Receiver Operating Charecteristic')

plt.legend(loc = "lower right")

plt.show()

