EX NO : 07

DATE : 09.05.2022

ROC CURVE PLOT

Aim:

To write python code to plot ROC curve used in ANN.

Equipments Required:

1. Hardware – PCs
2. Anaconda – Python 3.7 Installation / Moodle-Code Runner /Google Colab

Related Theory Concept:

The receiver operating characteristic (ROC) curve is frequently used for evaluating the performance of binary classification algorithms. It provides a graphical representation of a classifier's performance, rather than a single value like most other metrics. First, let's establish that in binary classification, there are four possible outcomes for a test prediction: true positive, false positive, true negative, and false negative. The ROC curve is produced by calculating and plotting the true positive rate against the false positive rate for a single classifier at a variety of thresholds.

Uses of ROC Curve :

One advantage presented by ROC curves is that they aid us in finding a classification threshold that suits our specific problem.

On the other hand, if our classifier is predicting whether someone has a terminal illness, we might be ok with a higher number of false positives (incorrectly diagnosing the illness), just to make sure that we don't miss any true positives (people who actually have the illness).

Algorithm

1. Import Necessary Packages
2. Load the Data
3. Create Training and Test Samples
4. Fit the Logistic Regression Model
5. Model Diagnostics

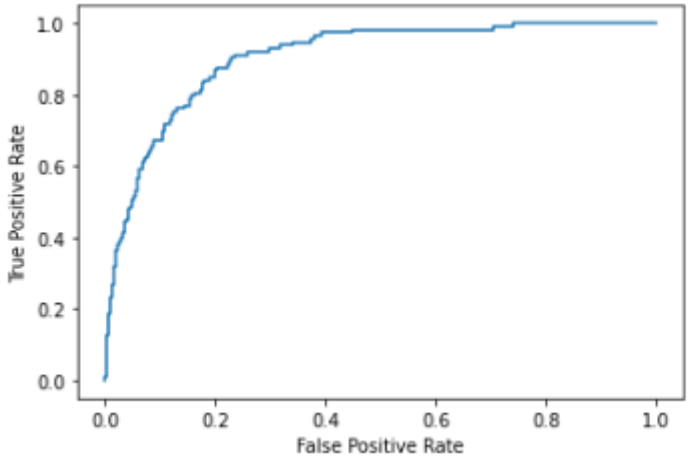
Program:

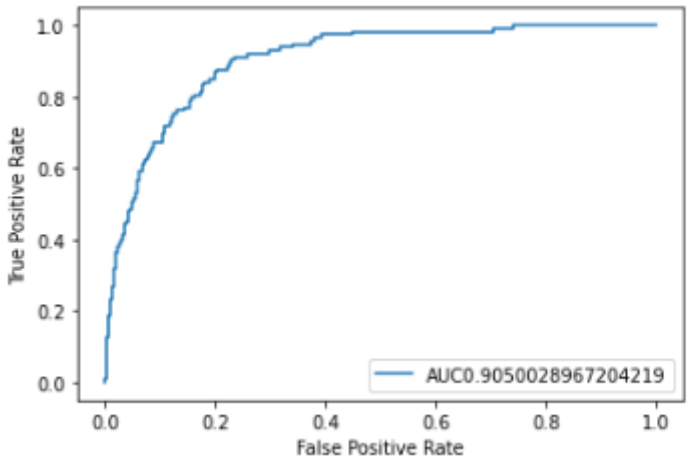
```
/*
Program to plot Receiver Operating Characteristic [ROC] Curve.
Developed by   : Monica M
RegisterNumber :  212219040082
*/
```

```
import pandas as pd
import numpy as np
from sklearn import metrics
from sklearn.linear_model import LogisticRegression
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

url = "https://raw.githubusercontent.com/Statology/Python-Guides/main/default.csv
data = pd.read_csv(url)
```

```
x=data[['student','balance','income']]

y=data['default']
x_train,x_test,y_train,y_test,= train_test_split(x,y,test_size=0.3,random_state=0
log_regression= LogisticRegression()
log_regression.fit(x_train,y_train)
#define metrics
y_pred_proba=log_regression.predict_proba(x_test)[::,1]
fpr,tpr, _ = metrics.roc_curve(y_test,y_pred_proba)

plt.plot(fpr,tpr)
plt.ylabel("True Positive Rate")
plt.xlabel("False Positive Rate")

plt.show()

#define metrics
y_pred_proba=log_regression.predict_proba(x_test)[::,1]
fpr,tpr, _ = metrics.roc_curve(y_test,y_pred_proba)
auc = metrics.roc_auc_score(y_test,y_pred_proba)


plt.plot(fpr,tpr, label="AUC" + str(auc))
plt.ylabel("True Positive Rate")
plt.xlabel("False Positive Rate")
plt.legend(loc=4)

plt.show()
```

Output:





Result:

Thus the python program successully plotted Receiver Operating Characteristic [ROC] Curve.