Model Practical Examination

Reg No: 39110636

Name: Mohnish Devaraj

Branch: CSE

semester: IV sem

Subject Code: SCSA2601

Subject Name: Machine Learning & Data Analytics lab

Date of Exam: 18-04-2022

Batch-id: ML-7

Duration: 3 Hours

No. of pages: 7

① Aim

To write a python code to find the Least Common Multiple among given two numbers.

Algorithm:

step 1: start

step 2: start the function LCM by calling the two variables

step 3: check if the first number is greater than second number, if yes copy the first number to greater, else goto step 3.1

step 3.1: copy the second number to greater

step 4: start a while loop for True

step 4.1: check if greater % a is 0 and greater % b is 0

step 4.2: copy greater to LCM and break

step 4.3: increment greater by 1

step 5: return LCM

step 6: Get the input from the user

step 7: Call the function and print LCM

step 8: Stop

Program:

```
def LCM(a, b):
    if a > b:
        greater = a
    else:
        greater = b
    while (True):
        if ((greater % a == 0) and (greater % b == 0)):
            lcm = greater
            break
        greater += 1
    return lcm

a = int(input())
b = int(input())
print("The L.C.M. is", LCM(a,b))
```

Output:

The output is attached below.

Result:

The above program is executed and the output is verified.

(2) Aim:

To Evaluate the results of machine learning algorithm

Read Actual values vs Predicted values from the user

Compute the following

A) Confusion Matrix

B) Accuracy

C) Specificity

D) Sensitivity

E) Precision

F) Recall

G) Misclassification Error

Algorithm:

step 1: Start

step 2: Input the file Y into program

step 3: predict the Y

step 4: Initialize the TP, TN, FP, FN as '0'

step 5: Using the for loop calculate the confusion_matrix

step 6: Calculate the Accuracy, precision, Recall classification error, and specificity

step 7: print the result ACC, PRE, REC, SN, SP & MCB

step 8: Stop

Program:

```
y=['0','1','0','1','1','1','0','1','0','1','0','0','0','1','1','1','0',
   '1','1','0']

y_pred=['0','0','0','0','1','0','1','1','1','1','0','0','0','0','0',
   '1','0','1','1','0']

j=0
TP,TN,FP,FN=0,0,0,0
for i in y:
    if i=='1' and y_pred[j]=='1':
        TP+=1
    if i=='0' and y_pred[j]=='0':
        TN+=1
    if i=='1' and y_pred[j]=='0':
        FP+=1
    if i=='0' and y_pred[j]=='1':
        FN+=1
    j+=1

Confusion_matrix = [TP,TN,FP,FN]
print('A) Confusion Matrix: ', confusion_matrix)

ACC = (TP+TN)/(TP+FP+TN+FN)
print('B) Accuracy: ', ACC)

SP = TN/(TN+FP)
print('C) Specificity: ', SP)

SN = TP/(TP+FN)
print('D) Sensitivity: ', SN)
```

```
PREC = TP/(TP+FP)
print("E) Precision :", PREC)

REC = TP/(TP+FN)
print("F) Recall :", REC)

MCE = 1-ACC
print('G) Missclassification Error :', MCE)
```

Result:

The above program is executed successfully and the output is verified.

Output:

The output is attached below