**TEST CODE-**

import numpy

from sklearn.svm import SVC

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import accuracy\_score

from sklearn.ensemble import RandomForestClassifier

import joblib

import tkinter

from tkinter import filedialog

import os

#from pygame import mixer # Load the popular external library

import time

#mixer.init()

root = tkinter.Tk()

root.withdraw()

while True:

currdir = os.getcwd()

file = filedialog.askopenfilename()

a = numpy.genfromtxt(file, delimiter=' ')

a=a.reshape(1,-1)

clf=joblib.load('valance.pkl')

val = int(clf.predict(a))

clf=joblib.load('Arousal.pkl')

aro = int(clf.predict(a))

if val==0 and aro==0:

print('Sad')

if val==0 and aro==1:

print('Depressed')

if val==1 and aro==0:

print('Relaxed')

if val==1 and aro==1:

print('Happy')

**SVM CLASSifier-**

import numpy

from sklearn.svm import SVC

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import accuracy\_score

from sklearn.ensemble import RandomForestClassifier

import joblib

def svm\_classifier():

file\_x = 'data/features\_sampled.dat'

file\_y = 'data/label\_class\_0.dat'

X = numpy.genfromtxt(file\_x, delimiter=' ')

y = numpy.genfromtxt(file\_y, delimiter=' ')

# Split the data into training/testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.1, random\_state=100)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

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

X\_test = sc.transform(X)

for i in range(0,len(X\_train)):

v=str(i)

fn='Test data/'+v+'.txt'

#print(X\_train[i])

c = numpy.savetxt(fn, X\_train[i], delimiter =', ')

# SVM Classifier

clf = SVC()

#clf = RandomForestClassifier(n\_estimators=10)

print(X\_train)

clf.fit(X\_train, y)

y\_predict = clf.predict(X\_train)

#print(y\_predict)

joblib.dump(clf, 'valance.pkl')

cm = confusion\_matrix(y, y\_predict)

print("Accuracy score of Valence ")

print(accuracy\_score(y, y\_predict)\*100)

###############################################################

file\_x = 'data/features\_sampled.dat'

file\_y = 'data/label\_class\_1.dat'

X = numpy.genfromtxt(file\_x, delimiter=' ')

y = numpy.genfromtxt(file\_y, delimiter=' ')

# Split the data into training/testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.1, random\_state=42)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

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

X\_test = sc.transform(X)

# SVM Classifier

clf = SVC()

clf.fit(X\_train, y)

y\_predict = clf.predict(X\_test)

cm = confusion\_matrix(y, y\_predict)

joblib.dump(clf, 'Arousal.pkl')

print(cm)

print("Accuracy score of Arousal ")

print(accuracy\_score(y, y\_predict)\*100)

if \_\_name\_\_ == '\_\_main\_\_':

svm\_classifier()