import pandas as pd

import numpy as np

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

from sklearn.preprocessing import StandardScaler

from sklearn.neural\_network import MLPClassifier

from sklearn.metrics import classification\_report,confusion\_matrix

import matplotlib.pyplot as plt

scaler = StandardScaler()

import csv

with open("train.csv", 'r') as f:

can = list(csv.reader(f, delimiter=","))

can = np.array(can[1:], dtype=np.float)

X=can[:,0:20]

y=can[:,20]

file = open("output.csv", mode="w")

#X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,random\_state=0)

#scaler.fit(X\_train)

scaler.fit(X)

#X\_train = scaler.transform(X\_train)

#X\_test = scaler.transform(X\_test)

X=scaler.transform(X)

mlp = MLPClassifier(hidden\_layer\_sizes=(50),learning\_rate\_init=.0005, alpha=1, max\_iter=2000,random\_state=42)

#mlp.fit(X\_train,y\_train)

mlp.fit(X,y)

#print("Accuracy {:3f}".format(mlp.score(X\_test,y\_test)))

#print("Accuracy {:3f}".format(mlp.score(X\_train,y\_train)))

with open("test.csv", 'r') as f:

can = list(csv.reader(f, delimiter=","))

can = np.array(can[1:], dtype=np.float)

X1=can[:,0:20]

scaler.fit(X1)

X1 = scaler.transform(X1)

output=mlp.predict(X1)

print(output)

for i in output:

file.write("%d\n"%i)