import numpy as np  
import pandas as pd  
import scipy  
import sklearn  
from sklearn import preprocessing  
from sklearn import neighbors  
from sklearn.model\_selection import train\_test\_split  
from sklearn import metrics  
import matplotlib.pyplot as plt

grade = pd.read\_csv("rp1.csv")  
grade.columns = ['SAT\_Score', 'HS\_GPA', 'FU\_GPA']

X = grade.drop('FU\_GPA', axis=1).values  
y = grade['FU\_GPA'].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y,test\_size = 0.1, shuffle=False)

lab\_enc = preprocessing.LabelEncoder()  
y\_train\_encoded = lab\_enc.fit\_transform(y\_train)

from sklearn.neural\_network import MLPClassifier

mlp = MLPClassifier(hidden\_layer\_sizes=(100,50,25), activation="identity", alpha=0.0001, batch\_size = 'auto',   
 learning\_rate = 'constant', learning\_rate\_init = 0.001, random\_state=1, max\_iter=2000)  
mlp.fit(X\_train, y\_train\_encoded)  
y\_predict\_mlp = mlp.predict(X\_test)

lab\_eno = preprocessing.LabelEncoder()  
y\_test\_encoded = lab\_enc.fit\_transform(y\_test)  
lab\_enc = preprocessing.LabelEncoder()  
y\_pred\_encoded = lab\_enc.fit\_transform(y\_predict\_mlp)

from sklearn.metrics import accuracy\_score

accuracy\_score(y\_test\_encoded,y\_pred\_encoded, normalize=True,sample\_weight=None)

0.01

from sklearn.model\_selection import GridSearchCV  
from sklearn.model\_selection import ShuffleSplit  
cv=ShuffleSplit(n\_splits=10, test\_size=0.1, random\_state=1)  
params = {'hidden\_layer\_sizes':[100,50,25], 'activation':['identity'], 'max\_iter':[2000], 'random\_state':[1],   
 'learning\_rate':['constant']}  
model = GridSearchCV(MLPClassifier(), params, cv=cv)  
model.fit(X\_train, y\_train\_encoded)  
model.best\_params\_

{'activation': 'identity',  
 'hidden\_layer\_sizes': 25,  
 'learning\_rate': 'constant',  
 'max\_iter': 2000,  
 'random\_state': 1}