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.neighbors import KNeighborsClassifier

knn=KNeighborsClassifier(n\_neighbors = 2)

knn.fit(X\_train, y\_train\_encoded)

KNeighborsClassifier(n\_neighbors=2)

y\_pred = knn.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\_pred)

from sklearn.metrics import accuracy\_score

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

0.02

from sklearn.model\_selection import GridSearchCV  
params = {'n\_neighbors':[2,3,4,5,6,7,8,9]}  
model = GridSearchCV(knn, params, cv=5)  
model.fit(X\_train, y\_train\_encoded)  
model.best\_params\_

C:\Users\siddisaa\Anaconda3\lib\site-packages\sklearn\model\_selection\\_split.py:666: UserWarning: The least populated class in y has only 1 members, which is less than n\_splits=5.  
 warnings.warn(("The least populated class in y has only %d"

{'n\_neighbors': 4}