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']

grade.head()

SAT\_Score HS\_GPA FU\_GPA  
0 1270 3.4 3.18  
1 1220 4.0 3.33  
2 1160 3.8 3.25  
3 950 3.8 2.42  
4 1070 4.0 2.63

grade.shape

(1000, 3)

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)

from sklearn.neighbors import KNeighborsRegressor

KNR = KNeighborsRegressor(6)  
KNR.fit(X\_train, y\_train)

KNeighborsRegressor(n\_neighbors=6)

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

from sklearn.metrics import mean\_absolute\_error  
from sklearn.metrics import mean\_squared\_error  
from sklearn.metrics import mean\_squared\_log\_error  
from sklearn.metrics import median\_absolute\_error  
from sklearn.metrics import r2\_score

mean\_absolute\_error(y\_test, y\_test\_predict)

0.46276666666666666

mean\_squared\_error(y\_test, y\_test\_predict)

0.3319418333333333

mean\_squared\_log\_error(y\_test, y\_test\_predict)

0.033371124016016364

median\_absolute\_error(y\_test, y\_test\_predict)

0.4158333333333335

r2\_score(y\_test, y\_test\_predict)

0.2619989465782736

from sklearn.model\_selection import GridSearchCV  
params = {'n\_neighbors':[2,3,4,5,6,7,8,9]}

model = GridSearchCV(KNR, params, cv=5)

model.fit(X\_train, y\_train)  
model.best\_params\_

{'n\_neighbors': 9}