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)

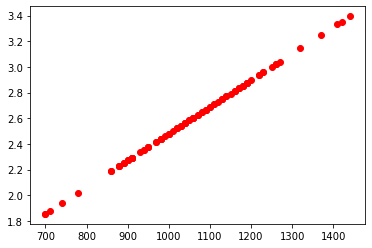
X = grade.drop('FU\_GPA', axis=1).values  
y = grade['FU\_GPA'].values  
  
X1 = grade.drop(['HS\_GPA', 'FU\_GPA'], axis=1).values  
X2 = grade.drop(['SAT\_Score', 'FU\_GPA'], axis=1).values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y,test\_size = 0.1, shuffle=False)  
X\_train1, X\_test1, y\_train1, y\_test1 = train\_test\_split(X1,y,test\_size = 0.1, shuffle=False)  
X\_train2, X\_test2, y\_train2, y\_test2 = train\_test\_split(X2,y,test\_size = 0.1, shuffle=False)

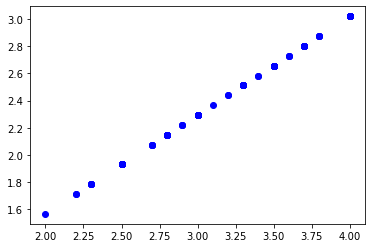
from sklearn.neural\_network import MLPRegressor  
reg = MLPRegressor(hidden\_layer\_sizes=(100,50,25), activation="identity", alpha=0.0001, batch\_size = 'auto',   
 learning\_rate = 'constant', learning\_rate\_init = 0.001, random\_state=1, max\_iter=2000).fit(X\_train, y\_train)  
reg1 = MLPRegressor(hidden\_layer\_sizes=(100,50,25), activation="identity", alpha=0.0001, batch\_size = 'auto',   
 learning\_rate = 'constant', learning\_rate\_init = 0.001, random\_state=1, max\_iter=2000).fit(X\_train1, y\_train1)  
reg2 = MLPRegressor(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).fit(X\_train2, y\_train1)

y\_test\_predict = reg.predict(X\_test)  
y\_test\_predict1 = reg1.predict(X\_test1)  
y\_test\_predict2 = reg2.predict(X\_test2)

plt.scatter(X\_test1, y\_test\_predict1, color='red')  
plt.show()  
plt.scatter(X\_test2, y\_test\_predict2, color='blue')  
plt.show



<function matplotlib.pyplot.show(close=None, block=None)>



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)

0.4451902839609075

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

0.3098429871354448

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

0.028818519543335874

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

0.35473933780599

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

0.3111309635032664

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(MLPRegressor(), params, cv=cv)  
model.fit(X\_test, y\_test\_predict)  
model.best\_params\_

{'activation': 'identity',  
 'hidden\_layer\_sizes': 50,  
 'learning\_rate': 'constant',  
 'max\_iter': 2000,  
 'random\_state': 1}