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("rp3.csv")  
grade.columns = ['SAT\_Score', 'FU\_GPA']

grade.head()

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

grade.shape

(1000, 2)

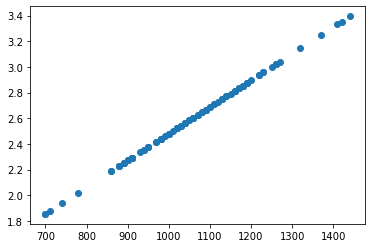
X = grade.drop('FU\_GPA', axis=1).values  
y = grade['FU\_GPA'].values

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

from sklearn.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)

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

plt.scatter(X\_test, y\_test\_predict)  
plt.show()



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)

0.5049939016030475

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

0.38828415797549165

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

0.03924149473072824

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

0.4161269811188133

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

0.13673394300643793

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': 100,  
 'learning\_rate': 'constant',  
 'max\_iter': 2000,  
 'random\_state': 1}