**Student Names: Ghadir Alfadhl & Li Baio**

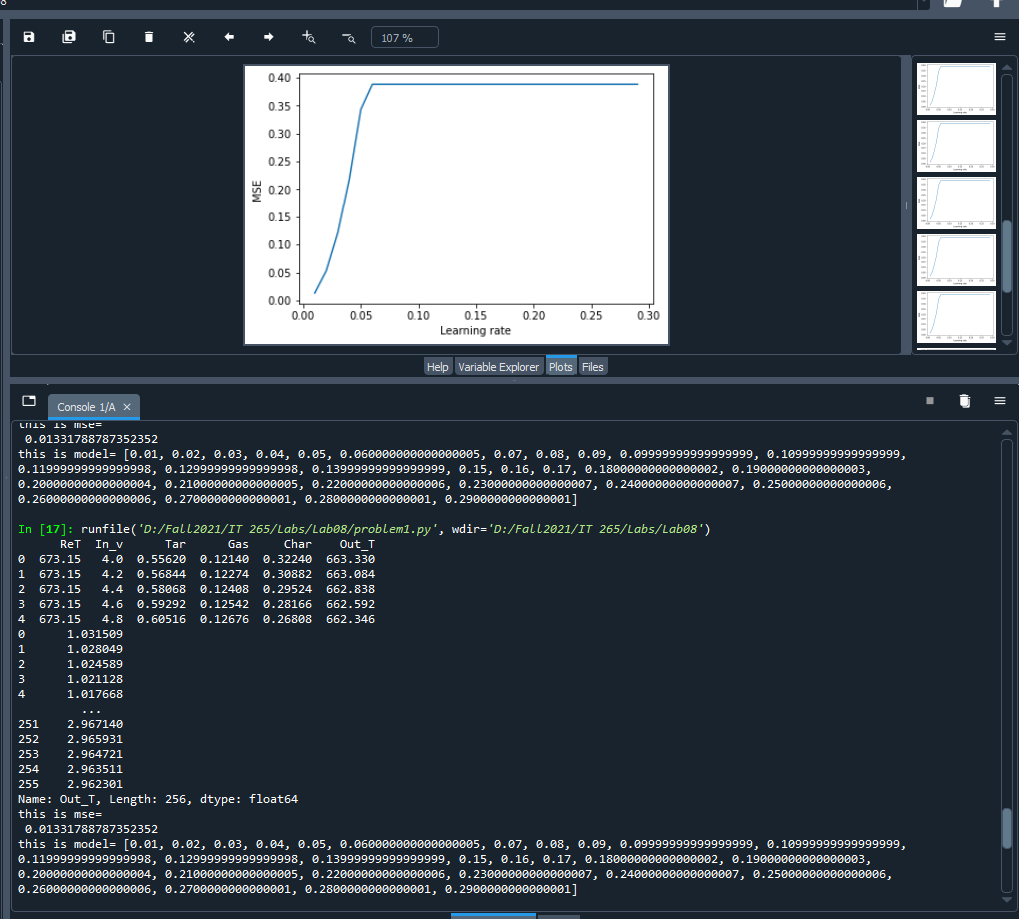
**Course Name: ITS265** – **Lab08**

**Problem 1:** Use data from the file “data-Lab10.csv” containing data collected and processed from a simulation study for a reheat furnace. The data has 6 columns where the first two columns are input variables and the remaining columns are the outputs of the simulation. In this lab, two input data variables, Reheat Temperature (‘ReT’) and Injection velocity (‘In\_v’) will be used to predict the Output Temperature (‘Out\_T’) using a Linear Regression model, called LASSO. Implement this in python using SKLearn.

Split the data into training and test sets. Before using the model to predict the output, the model should be optimized first. Use a variety of different learning rates (alpha) to optimize this model. Use MSE (mean squared error) to evaluate the model with different learning rates in order to select an optimal value.

Note that data should be normalized in the range of 1 and 3.

Show your output and plot of mean MSEs of training and validation for different learning rate values (alpha). And you should conclude with your selection of learning rate value and explain why your selection optimizes the model.



import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import MinMaxScaler

#from sklearn.linear\_model import LinearRegression

import matplotlib.pyplot as plt

df= pd.read\_csv("data-Lab10.csv")

print(df.head())

# Normalization in the range 1 to 3

def normalize(values, actual\_bounds, desired\_bounds):

return [desired\_bounds[0] + (x - actual\_bounds[0]) \* (desired\_bounds[1] - desired\_bounds[0]) / (actual\_bounds[1] - actual\_bounds[0])

for x in values]

#normalized in the range of 1 and 3

for i in df.columns:

df[i]= normalize(df[i], [min(list(df[i])), max(list(df[i]))], [1,3] )

print(df[i])

X= df[['ReT', 'In\_v']]

y= df['Out\_T']

# splitting the data into train and test data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,

y,

test\_size=0.5,

random\_state=1)

# Implementing Lasso

from sklearn.linear\_model import Lasso

model= Lasso(alpha= 0.01, normalize=True)

model = model.fit(X\_train, y\_train)

MSE = np.square(np.subtract(y\_test, model.predict(X\_test))).mean()

print("this is mse=\n", MSE)

# plotting the MSE vs learning rate

#Show your output and plot of mean MSEs of training and

#validation for different learning rate values (alpha).

#And you should conclude with your selection of learning rate value and

# explain why your selection optimizes the model.

alpha=[]

mse=[]

i=0.01

while i<0.3:

alpha.append(i)

model= Lasso(alpha=i, normalize=True)

model= model.fit(X\_train, y\_train)

MSE= np.square(np.subtract(y\_test, model.predict(X\_test))).mean()

mse.append(MSE)

i += 0.01

print("this is model=", alpha)

plt.plot(alpha, mse)

plt.xlabel("Learning rate")

plt.ylabel("MSE")

plt.show()