Question 1 – Linear Regression Models

In the practical lab folder you will find a data file called data.xlsx.

There are two columns in the dataset. The first column contains all the (input) X data and the second column contains all the output (Y) data.

Instructions:

1) The objective of this question is to build and assess the performance of a linear regression model for predicting the Y value by considering only the input X value. Throughout the code, you should try to avoid the use of iterative 'for loops' and instead use NumPy as much as possible. You should only need to include a single *for* loop to control the number of iterations of gradient descent.You will also find a template py file in the folder, which will help you get started with the first few steps below.

   a. Read the dataset into your code file (you can use Pandas pd. read_excel method to read the contents of an Excel file into a DataFrame)
   b. Extract the X and Y values and store in separate NumPy arrays (remember you can convert a Dataframe or Series object to a NumPy array using .values).
   c. Visualize the relationship between the X feature and the target Y value.
   d. Perform standardization on the X feature.
   e. Write code that will build a linear regression model using gradient descent as explained in the lecture notes. You can use the following initial values for the parameters:
      i. bias = 0.0
      ii. lambda = 0.0
      iii. alpha = 0.005

   Set the number of iterations of gradient descent to 50.

   f. We need a way of determining the performance of the model as we iterate using gradient descent. Write a function that will calculate the Mean Squared Error for the linear model produced. Visualize the MSE value as you iterate through gradient descent. Do you think the current value of the learning rate is appropriate? Try the following values for alpha and observe the change in the graph [0.005, 0.05, 0.5, 5, 50].

g. Finally you might be interested in graphing the linear hypothesis against the training data as shown below. To do this you just have to (i) produce a scatter plot (plt.scatter(X, Y)) to plot all the training data and (ii) plot all the predicted y values for the training data as a line (plt.plot(X,yPredictions,'k-'))