Requirements:

1) The .ipynb file shall include not only the source code, but also necessary plots/figures and discussions which include your observations, thoughts and insights.

2) Please avoid using a single big block of code for everything then plotting all figures altogether. Instead, use a small block of code for each sub-task which is followed by its plots and discussions. This will make your homework more readable.

3) Please follow common software engineering practices, e.g., by including sufficient comments on functions, important statements, etc.

20% points will be deducted if you don't follow the previous requirements!!!

[10 pts]Question 1. Hands-on Linear Regression

In this question, you need to write a program to find the coefficients of a linear regression model for the dataset provided (data2.txt). Use NumPy to load the data and plot it [2 pts].

Assume a linear model: $y = w_0 + w_1 * x$. Use Python to implement the following methods to find its coefficients. Please follow the requirements step by step. Note: For Methods 1, 2, and 3, please implement the algorithms by yourself and do NOT use any function of any library to find the solutions.

[5 pts] Method 1: Normal equation

[3 pts] Split the dataset into 80% for training and 20% for testing.
Hints: You can generate a sequence having the same length as the data number. Shuffle it at random. Select the first 80% as the training set and the other as the testing set.

[20 pts] Method 2: Stochastic gradient Descent

1. **Use stochastic** Gradient Descent to find the coefficients of the linear model. Plot MSE vs. iteration for **both the training set and testing set in one figure**.

Hint: Your code should include

1. Weights initialization. [2]
2. A loop updating the parameters based on the gradient (See Slide 22). The termination conditions of the loop should at least include the maximum iteration number and the cost function threshold. [10]
3. For each loop, MSE on the training and testing dataset should be calculated and stored for plotting. [3]

2. Use different learning rates (using 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01) to fit the model. Determine the best learning rate. [5]

[20 pts] Method 3: **batch** gradient descent
Use **batch** gradient descent to repeat all the steps in SGD. [15]

Compare the best result of SGD with that of BGD in terms of accuracy (of the testing set) and speed of convergence **with discussion.** [5]

Question 2 Logistic Regression Practice [15 pts, 5 pts for each sub-question]
In this part, you will face a real-world dataset, and use Logistic Regression to make the classification.

1) Step 1: Download and read the data. Split the dataset and apply Logistic Regression to classify different categories.

   You need to download the file from the link. Recall the code in the Question 1. Please load the data and split the training set and the testing set.
   Scikit Learn is the most used machine learning package. Use the *sklearn.linear_model.LogisticRegression* to make the classification. You can refer to the doc to check the function's parameters and see one example.

2) Step2 Logistic Regression with penalty item

   Find the optimal lambda of logistic regression with l1 and l2 norm penalty with the function *sklearn.linear_model.LogisticRegressionCV*.

3) Compare the results of these models in terms of accuracy, and feature number. Try to analyze and give insights into the data by the models.