Assignment 1: Logistic Regression and Neural Networks

Homework assignments will be done individually: each student must hand in their own answers. Use of partial or entire solutions obtained from others or online is strictly prohibited. Electronic submission on Canvas is mandatory.

1. **Document Classification** (100 points) In this homework, you need to classify text paragraphs into three categories: *Fyodor Dostoyevsky*, *Arthur Conan Doyle*, and *Jane Austen* by building your own classifiers. The data provided is from Project Gutenberg. Please follow the steps below:

   - (5pts) **Preprocess data**: Remove punctuation, irrelevant symbols, urls, and numbers. You can remove the unrelated text in the beginning of each file.
   - (5pts) **Construct examples**: Divide each document into multiple paragraphs. Each paragraph will be one example. Text that is not part of a paragraph can be discarded or preprocessed. Report the total number of examples for each category.
   - (5pts) **Data split**: Sample these paragraphs into training and testing data.
   - (5pts) **Feature extraction**: Build a vocabulary to represent each paragraph using only training data. Consider TF-IDF features for each input example.
   - (60pts) **Train** two classifiers (described below).
     1. Implement a Logistic Regression (LR) model with L_2 regularization from scratch:

$$J = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}\log\left(\frac{\exp f_k}{\sum_{c=1}^{K}\exp f_c}\right) + \lambda\sum_{j=1}^{d} w_{kj}^2$$

        - (10 pts) Given this formula, show the steps to derive the gradient of J with respect to $\mathbf{w}_k$.
        - (20 pts) Write a function for mini-batch gradient descent.
        - (20 pts) Write a function for stochastic gradient descent.

        Report the results and plots for both mini-batch and stochastic gradient descent.
     2. (10 pts) Build and train a Multilayer Perceptron (MLP) model (i.e., a two-layer neural network) using backpropagation. Please specify the settings of the model such as the network structure, the optimizer, the initial learning rate, the loss function.
   - (5pts) **Plot** training loss and validation loss every 100 epoches.
   - (5pts) Use **cross-validation** on the training data, report the recall and precision for each category on the test and validation sets, choose the best λ(in LR) and the number of neurons in the hidden layer (in MLP) using the validation set.
   - (10pts) **Compare** both classifiers and provide an analysis for the results.

Please follow the below instructions when you submit the assignment.

1. You are allowed to use packages for reading text, TF-IDF, plotting, and MLP, but you are not allowed to use packages for mini-batch gradient descent or stochastic gradient descent.
2. Your submission should consist of a zip file named Assignment1_LastName_FirstName.zip which contains:
   - a jupyter notebook file(.ipynb). The file should contain the code and the output after execution. You should also include detailed comments and analysis (plots, result tables, etc).
   - a pdf (or jpg) file to show the derivation steps of the gradient of J with respect to $\mathbf{w}_k$.