ECE 7995: ST AI for NLP
Assignment 1
Due date: October 02, 2022

The goal of this assignment is to implement a "logistic regression" model from scratch for email classification. Emails are available on canvas "spam_ham_dataset.csv." Each email is labeled as either "spam" or "ham."
In this assignment, students have to use **Python only.** You are not allowed to use other tools, such as **sklearn, nltk, etc.**

Load the data using pandas DataFrame. Check how many samples are "spam" and how many are "ham." (**Note:** you need to remove unlabeled samples (if any).)
Then, clean the data and implement a classification model as follows:

1. Develop a class/object in Python for "data preprocessing". The class includes a word tokenization function/method, and at least four cleaning functions (such as lowercasing, remove punctuation/URLs/names/noisy characters, etc.)
   Note: For each function, you need to explain why it is used for the target problem.

2. Develop a class/object in Python for "features extraction" from the preprocessed text. Specifically, for each data sample, you need to extract two features. The first feature is the number of spam words in an email. The second feature is the number of ham words in an email.
   This class includes two functions/methods: 1) build the frequency dictionary which maps from (word, class) to frequency, and 2) extract features (bias, sum of spam frequencies, and sum of ham frequencies).

3. Implement a function to shuffle the corpus and split it into the train set (for model training) and test set (for model testing).

4. Develop a class/object in Python for "Logistic Regression". The class includes two methods/functions (fit and evaluate).

   "fit": this function is used to train the model. It includes 1) parameter initialization (random), 2) apply the sigmoid function, 3) calculate the cost function, and 4) update the parameters using gradient descent. The last step will be repeated until finding the minimum cost.

   "evaluate": this function is used to test the trained model and returns the evaluation metrics (such as accuracy, confusion matrix, etc.).

Finally, implement the **"Logistic Regression"** using sklearn and compare its performance with your implementation.

Note: submit one "Jupyter Notebook" that includes the whole code.