ENCS3340 **Project # 2** **Second Semester 2023**

Due date:	July 15, 2023
Late submission:	20% per day.
Teams:	You can do the project individually or in teams of 2. You can work with students from all sections. Teams must submit only 1 copy of the project.
Purpose:	The purpose of this project is to train and test different ML models on real dataset.

In this project, you will write a python code to classify emails as spam or not-spam.

Getting Started

- Download the attached code template and dataset in spam.zip and unzip it.
- Install the `scikit-learn` package if it isn't already installed, which you'll need for this project.

  (You can run `pip install scikit-learn` in the terminal to install it)

The Dataset

The provided `spambase.csv` file contains a set of 4601 examples. Each example is represented in one row which contains 58 numbers. The last number in each row denotes whether the e-mail was considered spam (1) or not (0). The first 57 numbers are the attributes. For details about the meaning of these attributes, please refer to https://archive.ics.uci.edu/dataset/94/spambase

Your Task

You need to test a k-NN and a multi-layer perceptron classifier on the provided dataset. For each classifier, you need to train it on the training set, and report the accuracy, precision, recall, and F1-score on the test set. The splitting of the data into train and test sets is already done for you in the template code. To make things simple, you only have to implement the k-NN classifier, whereas for the MLP, you will use the implementation from the `scikit-learn` package.

To structure your project, you only need to complete the missing functions in the provided template, which include:

- `load_data` to read the provided examples from the csv file.
- `preprocess` to preprocess the features of the dataset examples by normalizing each feature according to the following formula.

$$f_i = \frac{f_i - \bar{f_i}}{\sigma_i}$$

  Where f_i is the i-th feature, $\bar{f_i}$ is the mean value of the i-th feature (i-th column) computed from all examples, and σ_i is the standard deviation of the i-th feature.
- `train_mlp_model` which will be used to train the MLP classifier using the `scikit-learn` package.
- `evaluate` which computes the evaluation measures.
- And finally, complete the NN class which implements the k-NN classifier.

For the k-NN classifier, use k=3 and use Euclidean distance to find the nearest neighbors. For MLP, train a network with two hidden layers, the first layer with 10 neurons, and the second layer with 5 neurons, and use the logistic function (sigmoid) for the activation function.

Submission:

- You need to submit the completed template code file `template.py`

  Make sure to rename the file using the following structure `{firstname1}_{ID1}_{firstname2}_{ID2}.py`
- A report (~3 pages) to discuss the achieved results and any experiment that you tried. It should also contain the confusion matrix for both classifiers, and suggest possible ways to improve the performance of the tested models.

You have to submit a runnable code. Submissions with syntax or runtime errors will not be accepted.