**Department of Software Engineering**

**CS471: Machine Learning**

**Class: BESE-12AB**

**Lab 06: Nearest Neighbors**

**Date: 15 Mar 2024**

**Instructor: Dr. Hashir Kiani and Dr Duad Abdullah**

**Lab Engineer: Ms Iram Tariq Bhatti**

# Lab 06: Nearest Neighbors

**Introduction:**

The nearest neighbor algorithm is most suitable for classification tasks. In nearest neighbor approach to learning, the whole training dataset is used for the purposes of prediction for a new data point. For each new data point, its distance from all the data points in the training set is calculated. The prediction is then the label of the data point in the training examples that is closest to the new data point. In the k-nearest neighbor approach, we find k training examples that are closest to the new data point. The most common label/output of those k training examples is the prediction for the new data point.

**Objective:**

* Understand and implement KNN classifier from scratch and using off-the-shelf libraries.
* Compare custom implementation with scikit-learn implementation in terms of classification results and execution time.

**Tools:**

Google colab or equivalent environment

**Lab Task:**

In this lab, you are supposed to implement the k-Nearest Neighbor Classifier. Data in excel files (both the training and test sets) are uploaded on LMS. In the said training and test data files, each row contains data about one instance of a plant category where four predictors/attributes are recorded for each plant (namely, leaf length, leaf width, flower length, and flower width), while “plant” is the target class which could be any one of the following at a time: “Arctica” or “Harlequin” or “Caroliniana”.

You are supposed to perform classification with the following values of k.

1. k = 3
2. k = 5
3. k = 7

Figure out how the results vary on the test data (How results vary by increasing the number of k). Please submit your results for the complete test.

**Note:** You are expected to code the kNN algorithm from scratch in Python. However, you are allowed (in fact encouraged) to compare your results with standard implementations available in off-the-shelf libraries and note if how/why your results vary in terms of classification and execution time from that of the standard implementation available in the library you used for comparison.

**Deliverable:**

Jupyter notebook with codes for both implementations, output result on test data and analysis on results obtained and execution time.