Assignment 2: K-Nearest Neighbors **(Total Marks: 25)**

Problem Statement:

A floriculture research team X is studying the use of multiple measurements to distinguish three different iris flower species. The dataset contains a set of 150 records under five attributes: sepal length, sepal width, petal length, petal width and species (see Fig. 1). Develop a K Nearest Neighbour (KNN) classifier that classifies the species according to the above measurements.

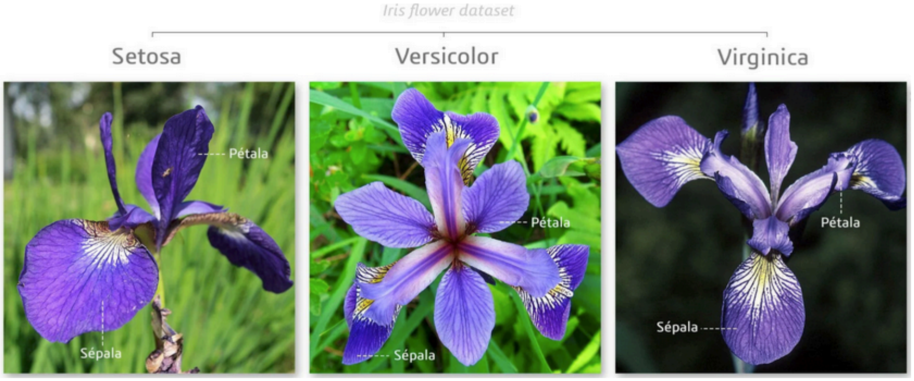

Figure 1: Different iris flower species and their attributes

Implementation: [3+2=5 Marks]

(a) Implementation of normal K-NN (KNN_Normal) from scratch (without using builtin functions or scikit-learn). The algorithm should be able to choose a particular K value.
(b) Implementation of distance weighted K-NN (KNN_Weighted) from scratch (without using builtin functions or scikit-learn). The algorithm should be able to choose a particular K value.

** Implement **[KNN_Normal]** and **[KNN_Weighted]** from scratch. You may make use of the numpy library to perform basic operations (e.g., sorting).

** Use **Euclidean Measure** for all cases.

** A custom weighting scheme that takes into account the distance. The weight for each neighbour should be calculated as 1 / d .

** In general, you may use libraries to process and handle data.

** Perform **Z-score Normalisation** before feeding the data in your model.

** To save computation, you may pre-compute pairwise distances of data-points and store that in a matrix.

Experiments: [3+3+3+3=12 Marks]

The dataset will be split into Train:Test with 70:30 ratio. Pl shuffle the data before splitting.

1. **Experiment 1:** Report the effect of varying K in [KNN_Normal] on Test data. Choose K values from [1, 3, 5, 10, 20]. Plot Percentage Accuracy vs K. Find the best value of the hyperparameter K. Further, plot the confusion matrix for the best K.

2. **Experiment 2:** Report the effect of varying K in [KNN_Weighted] on Test data. Choose K values from [1, 3, 5, 10, 20]. Plot Percentage Accuracy vs K. Find the best value of the hyperparameter K. Further, plot the confusion matrix for the best K.

3. **Experiment 3:** Add noise to only a fraction of the training data: Consider 10% of the training data for noise addition*. Choose a normal distribution with zero mean and standard deviation 1.0. Next, evaluate the new data using [KNN_Normal] and [KNN_Weighted] employing the optimal K found in the earlier experiments (Experiments 1 and 2). How do the performances vary as compared to that of the noiseless case (i.e., Experiments 1 and 2)?

   For noise, one may use: `numpy.random.normal(``loc=mean, scale=std_dev`*`, size=train_data.shape)`

4. **Experiment 4:** For the case of [KNN_Normal], study the effect of the curse of dimensionality. Using the optimal K obtained (in Experiment 1), consider (a) All four inputs (sepal length, sepal width, petal length, petal width), (b) Only petal parameters (i.e., petal length and petal width), (c) Only sepal parameters, (d) Only length parameters (sepal length and petal length) and (e) Only width parameters. Analyse whether the curse of dimensionality is imparted by petal parameters, sepal parameters, length parameters and width parameters.

Report your observations with appropriate explanations.

Datasets:

This dataset comprises three iris species with 50 samples each as well as some properties about each flower. You can find the dataset here.

- ID: Identification number of the flower
- Sepal length: Length of sepal in cm (in real numbers)
- Sepal Width: Width of flower sepal in cm (in real numbers)
- Petal length: Length of flower petal in cm (in real numbers)
- Petal Width: Width of flower petal in cm (in real numbers)
- Species: Three iris flower species (iris-setosa, iris-versicolor, and iris-virginica)

Problem: Predict the species of an iris flower

Submission:

A .zip file containing the python source code and a PDF report file. The final name should follow the template: <Assign-No>_<Your Roll No>.zip. For example, if your roll no is 15CE30021, the filename for Lab Asgn-2 will be: `LabAsgn-2_15ce30021.zip`

1. A **single python code (.py)** containing the implementations of the models and experiments with comments at function level. The first two lines should **contain your name and roll no**.

2. A report [PDF] containing : **[2+2+2+2 Marks]**
    a. Experiment 1: Plot of Percentage Accuracy vs K should be shown for [KNN_Normal]. Also mention the best choice for the K and the corresponding confusion matrix.
    b. Experiment 2: Plot of Percentage Accuracy vs K should be shown for [KNN_Weighted]. Also mention the best choice for the K and the corresponding confusion matrix.
    c. Experiment 3: Report the performance with and without noise levels. Comment on the robustness of [K-NN_Normal] and [KNN_Weighted] to noise in the training dataset.
    d. Experiment 4: Report the effect of curse of dimensionality in KNN_Normal based on points (a-e) as mentioned for the problem of Experiment 4.

Responsible TAs: Please write to the following TAs for any doubt or clarification regarding Assignment 2.

Trishita Mukherjee - trishitamukherjee77@gmail.com

Deadline: The deadline for submission is **22nd January (Monday), 11:55 PM, IST**. Irrespective of the time in your device, once submission in moodle is closed, no request for submission post-deadline will be entertained. No email submission will be considered. So, it is suggested that you start submitting the solution at least one hour before the deadline**.**

Plagiarism policy: Binary marking (two parties)