CS422/622- HW 2

Classification Task with KNN Model Implementation for IRIS Data

In HW2, you will develop a classification pipeline using the K-Nearest Neighbors (KNN) model for the Iris dataset. By completing this task, you'll gain hands-on experience with data preprocessing, model training, and evaluation—key steps in machine learning. Follow the guidelines below to ensure structured and successful implementation.

1. Dataset

- The Iris dataset is available at the UCI Machine Learning Repository (Iris - UCI Machine Learning Repository). The dataset can be downloaded or directly imported into the Python library. For more details, refer to the website.

2. Data Preprocessing

- **Feature Selection**: The Iris dataset has four features and one label. You may select all of them, some significant features, or using a PCA to reduce the dimensions.
- **Normalization**: Consider normalizing the features if necessary to ensure they are on the same scale, which is important for distance-based algorithms like KNN.
- **Missing Values**: Address any missing data by either removing or imputing the samples with missing values.

3. Model Training

- **Distance Metric**: Choose an appropriate distance or similarity measure (e.g., Euclidean distance or Cosine similarity) for your KNN model. Undergraduate students may use an existing KNN library.
- **Tuning K**: Use a subset of the dataset (distinct from the test set) to find the optimal value for K (the number of neighbors). Experiment with different K values to see which works best.

4. Model Evaluation

- **Accuracy Calculation**: Evaluate the model's performance using the following accuracy formula:

$$accuracy = \frac{\#\ of\ your\ predictions\ correctly\ classified}{\#\ of\ total\ test\ data}$$

- Compare your model's performance with the "Baseline Model Performance" posted on the website.

Report

- Your report should provide a clear and structured explanation of each step, from dataset selection to evaluation. Discuss not only what you did, but why you made specific choices.

- Include test codes and execution results along with screenshots of your results for better clarity and visualization.
- Describe what you learned from this assignment—focusing on insights about KNN, data preprocessing, or any challenges faced during implementation.
- Feel free to describe anything else you want, such as additional observations, interesting findings, or how you might approach the task differently in the future.
- **There are no restrictions on using external libraries,** but make sure to document the libraries you used and their purposes in the report.

Additional requirements for graduate students in CS 622

Graduate students are expected to:

- Implement KNN algorithm using Matrix computation. Please refer to the lecture notes slide 24. Efficient implementation in "04.KNN.pptx".
- Implement **5-fold cross-validation** to ensure a robust evaluation of your KNN model.

Submission instructions:

You must submit the followings to UNLV WebCampus:

1. A report file
2. Source code file(s)
   - Must be well organized (function name, indentation, …)
   - **You need to upload the python text file (*.py.txt). Simply add ".txt" to the py extension. Don't upload jupyter notebook files**

You must submit the files SEPERATELY. DO NOT compress into a ZIP file. If you fail to provide all required information or files, you may be given zero score without grading.

Once you submit, Webcampus will perform similarity check for your submission and show you the result. Your similarity score must be lower than 50% unless something essential is described in the report. Otherwise, (the score -50%) will be deducted. Detecting any attempts, including adding excessive comments, to bypass the similarity check may result in receiving zero points.