National Institute of Technology Calicut
Department of Computer Science and Engineering
CS4038D DATA MINING MONSOON 2024 - Assignment 2

Submission deadline (on or before):
10th November, 2024 10:00:00 PM

Complete the following Assignment Questions, and submit your assignment in the moodle (Eduserver) course page, on or before the submission deadline. Only one member among your team should make a submission in Eduserver on behalf of the entire team. Include a README.PDF which contains the name and roll number of the group members. Total of 7 files (3 Zip files, 3 PDF files related with each of the 3 following tasks and the README file) is expected to be submitted as part of this assignment.

During evaluation, the genuinity of the submission and contribution of each member will be checked either through viva/quiz. The total marks for the assignment is 12. The marks awarded will be based on the uploaded documents and the viva/quiz.

Assignment Questions

Perform the following tasks and submit the outcomes described for each task.

Dataset: It consists of 150 samples of three species of Iris flowers (Iris setosa, Iris versicolor, and Iris virginica). From each class, take 75% for training (including 5% validation) and 25% for testing randomly. Store it in train.csv, train-valid.csv and test.csv. These files should be included in the respective python code folders.

1. **Task 1:** Implement Naive Bayes classifier from scratch (do not use libraries) by discretizing the numeric feature into three equal-width bins. You should use Laplacian correction. After calculating the likelihood and prior probabilities in trainNBayes.py, save the probability values to a txtfile. The testNBayes.py should read the probabilities from the txtfile, and can use it for predicting the class labels for test dataset, and finally find accuracy, precision and recall by creating the confusion matrix.
   Outcome: Zip file (T1CODE_<TeamNumber>.zip) containing all the python codes. Document (T1_<TeamNumber>.PDF) which describes the experiment setup (if needed) and tabulates the evaluation measures - Accuracy, Precision and Recall, obtained for the task. Also, write down your inference on the misclassified samples.

2. **Task2.1:** Implement k-Nearest Neighbor classifier from scratch without using libraries for ‘k’ = 5, and Euclidean distance measure.
   Task2.2: Use the scikit-learn library to create k-Nearest Neighbor classifier. Experiment with different distance measures, and different ‘k’ values (Use grid search for fine tuning parameters). You are allowed to do any sort of tuning on the parameters.
   Outcome: Zip file (T2CODE_<TeamNumber>.zip) containing python codes. Document (T2_<TeamNumber>.PDF) which describes the experiment setup (if needed) and tabulates the evaluation measures - Accuracy, Precision and Recall, obtained for tasks. Also, write down your inference on the misclassified samples. The optimized model parameters and the performance of the model should be tabulated well.

3. **Task3**: Implement k-means clustering using scikit-learn library. Make use of the ground truth class labels only for cluster evaluation. Find the optimal number of clusters using any automated method. Experiment with different distance measures.
   Outcome: Zip file (T3CODE_<TeamNumber>.zip) containing python codes. Document (T3_<TeamNumber>.PDF) which describes the experiment setup (if needed) and tabulates the extrinsic and intrinsic cluster evaluation measures obtained for the task. The optimized model parameters and the performance of the model should be tabulated well.