CSE471: Statistical Methods in AI -- Spring 2016

Assignment 1: K-NEAREST NEIGHBOUR CLASSIFIER

DUE: Before 12midnight on 20 Jan 2016

INSTRUCTIONS:

1. You may do the assignment in Matlab/Octave, R, Python, C/C++ or Java.
2. You need to upload a single pdf file in the Courses Portal. The file should contain your answers as well as the code you have written and its output.
3. At the top-right of the first page of your submission, include the assignment number, your name and roll number.
4. *IMPORTANT:* Make sure that the assignment that you submit is your own work. *Do not copy any part from any source* including your friends, seniors or the internet. Any breach of this rule could result in serious actions including an **F grade** in the course.
5. Your grade will depend on the correctness of answers and output. In addition, due consideration will be given to the clarity and details of your answers and the legibility and structure of your code.

PREAMBLE:

This assignment requires you to implement *K-nearest neighbour (kNN) classifier* (a baby first step into the world of data classification!) and test it on three different datasets from the UCI Machine learning repository (*http://archive.ics.uci.edu/ml/*). Given below are the instructions for downloading the datasets and carrying out the experiment. Once you conduct the experiments, calculate the metrics described and answer the questions below.

Experimental Procedure:

1. Download any three datasets from the UCI ML repository, one of them being the *Iris dataset* (http://archive.ics.uci.edu/ml/datasets/Iris).
2. Randomly partition each dataset into two parts of equal size named *Training set* and *Testing set*.
3. Use the training set as labelled examples and do *K*-nearest neighbor classification (do with *K*=1 and 3) of the Testing set samples. Deal with any ties appropriately but mention this in observations part of the report.
4. Compute the accuracy (percentage of correct classification) using the labels of the Test samples and the output from your classifier.
5. Repeat steps 2 - 4 for ten times, each time dividing the dataset differently. Report the mean and standard deviation of the accuracy over the ten test runs.
6. Report the *confusion matrix* of each dataset, which depicts the classes that are most confused.
7. The process described in steps 2-5 is *Random subsampling approach*. Repeat this with 5-fold cross validation and report the mean and standard deviation of each fold as well as the grand mean of all the folds.

Questions to be Answered:

1. Please list the names and salient characteristics (Number of features, Number of instances, Number of Classes, etc) of the datasets you chose from the UCI ML repository for your experiments. Mention any criteria you used in deciding on the datasets and the distance function used for each dataset.
2. For each of the datasets, give the results of classification using the (1- and 3-) nearest neighbor classifiers (mean and variance of accuracy and the confusion matrix). Give your observations related to the results.
3. Plot the Iris dataset using only two features, namely, *Petal width* and *Sepal width* features (2D Plot). Plot the decision boundaries of the 1-nearest neighbor classifier in this plot. You should generate the plot automatically (using code). A simple but crude method is to classify each point in a 2D grid and find the transition points in each row and column where the classification changes from one class to another.
4. Will the decision boundary of a 3-NN (nearest neighbor) classifier be piecewise linear? Argue the correctness of your answer.