Istanbul Technical University- Fall 2017
BLG527E Machine Learning, Homework 3

Purpose: nonparametric methods, decision trees, linear discrimination, MLPs
Total worth: 6% of your grade.
Handed out: Wednesday, April 12, 2017.
Due: Wednesday, April 25, 2017 23:00. (through ninova!)
Instructor: Zehra Cataltepe (cataltepe@itu.edu.tr),
Assistant: Hakan Gündüz (hakangunduz@itu.edu.tr),
Policy: Collaboration in the form of discussions is acceptable, but you should write your own answer/code by yourself. Cheating is highly discouraged for it could mean a zero or negative grade from the homework.
If a question is not clear, please let us know (via email, during office hour or in class).

Submission Instructions: Please submit through the class ninova site.
Please zip and upload all your files using filename studentID_HW3.zip. You must provide all functions you wrote with your zipped file. Functions you do not submit may cause you lose a portion of your grade. You must also include a .doc or pdf file with answers to the questions and how to call your python or matlab functions for each question so that we can run and check the results.

QUESTIONS:

Dataset:
Optdigits data by Alpaydin and Kaynak, from UCI Machine Learning Repository:
ftp://ftp.ics.uci.edu/pub/ml-repos/machine-learning-databases/optdigits/
You need the files:

optdigits.names	explanation of data
optdigits.tra	training data
optdigits.tes	test data

Q1) [6pts] [knn (1pt), decision tree (1pt), linear discrimination (1pt), multilayer perceptron (3pt)]

Partition the optdigits.tra data randomly into 90% train and 10% validation sets.

For the following 4 methods: knn, decision tree, linear discrimination, multilayer perceptron **(you can use any available implementation, make sure you indicate what you used in your report)**:

- Train on training data and determine the best hyper-parameters **(e.g. k for knn, max tree depth for decision tree, etc.)** based on the validation accuracy.
- Using those parameters, train on the whole training data. Measure the training time.
- Test the trained models on the test data, measure the test time.
- Output the training and test confusion matrices and training and test times.

Q2) [2pts] Compare the training and test confusion matrices between the 4 models in terms of accuracy for each class and which classes are confused with each other the most. Also compare the training and test times of the 4 models.

Q3) [4pts] *(eliminate noisy instances)* Identify which training instances are most misclassified by the models. Eliminate 10% of training instances based on how many models have them misclassified. If there are ties, decide by random coin toss.
Train, validate the test your models again, as in Q1). Did test accuracies get better for each class after the removal of the 10% of noisy training instances.