Lab 2-1: Supervised Learning (Addendum)

**Due date:** Tuesday, May 2, 11:59pm. (Note: another lab assignment will start in the lab session on the due day.)

# Lab Assignment

This is the addendum to Lab 2. This addendum asks you to complete two additional tasks. First, we are asking you to complete the implementation of the `C4.5` algorithm so that it properly processes numeric attributes, and test the work on this algorithm on at least one dataset (Iris). Second, we are asking you to build an implementation of the Random Forest classifier based on your `C4.5` implementation, and compare the accuracy of the Random Forest on the available datasets to the accuracy of `C4.5` alone.

## Assignment Preparation

Continue working with your Lab 2 partner on this assignment.

## Datasets

One new dataset, Iris is made available. Iris is one of the most popular Machine Learning datasets. It is a simple dataset containing a few hundred records. Each record depicts physical dimensions of a specific iris flower from one of three different subspecies of iris: *Iris Setosa*, *Iris Versicolor*, or *Iris Virginica*. There are four physical parameters measured for each flower, listed below in the order in which they occur in the data file:

- Sepal length in cm
- Sepal width in cm
- Petal length in cm

- Petal width in cm

The dataset is available from the UCI Machine Learning repository. Lab 2 data page,

http://users.csc.calpoly.edu/∼dekhtyar/466-Spring2018/labs/lab02.html

contains the links to the data page and the data file.

## New Task 1: Complete C4.5 implementation

The original requirements of Lab 2 make it optional to implement handling of numeric attributes as part of your `C4.5` implementation. To give you incentive to do so, we are introducing the Iris dataset, which contains four numeric attributes (see above), and are asking you to classify it using your `C4.5` implementation.

Complete your `C4.5` implementation by adding the functionality to classify on numeric attributes.

## New Task 2: Random Forset implementation

Implement the Random Forest classifier. Your implementation shall behave as follows.

**Input Parameters.** Your implementation shall take as input the following parameters:

- m or NumAttributes: this parameter controls how many attributes each decision tree built by the Random Forest classifier shall contain.
- k or NumDataPoints: the number of data points selected randomly with replacement to form a dataset for each decision tree.
- N or NumTrees: the number of the decision trees to build.

**Dataset Selection.** Write a method/function that given a full dataset $D$ and the parameters NumAttributes and NumDataPoints selects the appropriate number of data points, and randomly selects NumAttributes and returns the constructed set back. This functionality will be used by your Random Forest classifier.

**Behavior.** Your Random Forest implementation shall take as input a training set $D$, and the three input parameters described above. It shall produce the requisite number of small decision trees, as guided by the input parameters. Each decision tree shall be produced by an appropriate call to your `C4.5` implementation.

**Evaluation.** You will use 10-fold cross-validation to compute the accuracy of the Random Forest classifier on your ELECTIONS and Iris datasets.

**The program.** Your Random Forest implementation shall be named randomForest.py or randomForest.java (or a similar file name for a programming language of your choice). It shall take as input the dataset filename, and the three input parameters described above. It shall produce, as output, a `results.txt` or `results.csv` file that produces predictions for each individual row in the dataset based on the 10-fold cross-validation evaluation. Separately, it shall also output the confusion matrix and the accuracy of prediction.

## Submission Instructions

The following is an **updated** list of deliverables. New deliverables are *in italics*.

- `README`. Shall contain the names and email addresses of all students in the team. Also, put any specific instructions and notes in this file. (e.g., if you choose a different implementation language, include translation/running instructions). *Include any instructions on how to run your Random Forest classifier.*
- Your programs: `InduceC45`, `classifier`, `Evaluate`, and any supplementary files. *Additionally, please include your Random Forest classifier.*
- The output of `Evaluate` on the `tree01-1000-numbers.csv` (or `tree01-1000-words.csv`) input with 10-fold cross-validation. Dump the output into a text file and submit.
- *A short report comparing the accuracy of predictions using 10-fold cross-validation for `C4.5` vs. Random Forest on all versions of the ELECTIONS dataset, and Iris dataset. Place the report file **outside** the archive that you are submitting.*

Place all other files into a `.zip` or a `.tar.gz` archive.

To submit use the following command:

```
$ handin dekhtyar lab02-466 <files>
```