# CS-331 PROJECT 2: MACHINE LEARNING

**DUE DATE: 05/05/17 at 11:59 pm**

**80 points + 5 bonus**

***You will solve the project in groups of two people. Submit one zip file per group with the information required via Dropbox (through myCourses).***

## PART 1 (80 points)

Perform different experiments in KEEL (7 in total), in order to test the behavior of the required algorithms when applied to the following datasets:

| | | | | | |
|---|---|---|---|---|---|
| Bupa | Monk-2 | Abalone | Contraceptive | Splice | Coil2000 |
| Ecoli | New-Thyroid | Australian | Heart | Thyroid | Flare |
| Glass | Pima | Balance | Led7digits | Tic-tac-toe | German |
| Haberman | Vehicle | Car | Lymphography | Titanic | Ionosphere |
| Iris | Wine | Chess | Ring | Vowel | Saheart |

The algorithms are listed below:

- Decision Trees:
  - AdaBoost.NC-C
  - C45-C
- Crisp Rule Learning
  - Ripper-C
- Evolutionary Crisp Rule Learning
  - GAssist-Intervalar-C
- Evolutionary Fuzzy Rule Learning
  - GFS-GCCL-C
- Fuzzy Rule Learning
  - Chi-RW-C
  - FURIA-C

You will also perform an experimental study in order to compare the algorithms among them. Use the **Friedman, Holm and Shaffer** statistical tests as convenient so that we can compare the best algorithm against the rest and all against all. You will find the details about how each one of the statistical tests work in the file **"statistics.pdf"**.

## Tasks

a) Add to KEEL the datasets needed in order to complete the experiments. **(10 points)**
b) Run each one of the experiments. Some of them may take a long time to finish. **(20 points)**
c) Collect the results in a .csv file (see **examples/FC_tra.csv** for clarification). Later, you will use this file to perform the statistical analysis. **(10 points)**
d) Write a document in which you show the results of the algorithms on each dataset together with the interpretation of the statistical analysis. Follow the ideas mentioned below: **(40 points)**

1. The results of each algorithm on each dataset for a given parameter (accuracy on training/test) should be shown through a table, i.e.:

| Data | Training | | | |
|---|---|---|---|---|
| | NSLV | NSLV-AR | NSLV-FR | SLAVE3 |
| appendicitis | **93.3 (1)** | 91.9 (4) | 92 (3) | 92.5 (2) |
| australian | 90.3 (2) | 87.4 (4) | **90.7 (1)** | 89.2 (3) |
| automobile | 97.6 (3) | 97.3 (4) | **99.1 (1)** | 98.1 (2) |
| balance | 85.8 (3) | 81.1 (4) | **98.4 (1)** | 96.6 (2) |

In the previous table, each row shows the accuracy on training for a specific dataset provided by each of the algorithms considered.

2. When doing the experimental study, use the tables obtained by the statistical analysis to support your interpretation of the results.

Table 6.18: Adjusted $p$-values (accuracy on testing set).

| i | hypothesis | unadjusted $p$ | $p_{Shaf}$ |
|---|---|---|---|
| 1 | SGERD vs .FURIA | 0 | 0 |
| 2 | GCCL vs .FURIA | 0 | 0 |
| 3 | SGERD vs .FARC-HD | 0 | 0 |
| 4 | SGERD vs .SLAVE3 | 0 | 0.000001 |
| 5 | C45 vs .SGERD | 0 | 0.000002 |
| 6 | GCCL vs .FARC-HD | 0 | 0.000004 |
| 7 | GCCL vs .SLAVE3 | 0.000019 | 0.000135 |
| 8 | GCCL vs .C45 | 0.000029 | 0.000201 |
| 9 | C45 vs .FURIA | 0.055829 | 0.390805 |
| 10 | FURIA vs .SLAVE3 | 0.068345 | 0.410072 |
| 11 | FARC-HD vs .FURIA | 0.309656 | 1.238624 |
| 12 | GCCL vs .SGERD | 0.324102 | 1.296408 |
| 13 | C45 vs .FARC-HD | 0.370028 | 1.296408 |
| 14 | FARC-HD vs .SLAVE3 | 0.419794 | 1.296408 |
| 15 | C45 vs .SLAVE3 | 0.928572 | 1.296408 |

| Algorithm | Ranking |
|---|---|
| GCCL | 4.8 |
| C45 | 3.05 |
| SGERD | 5.2125 |
| FARC-HD | 2.675 |
| FURIA | 2.25 |
| SLAVE3 | 3.0125 |

| Friedman p-value |
|---|
| 4.43937109295689E-11 |

For PART 1, provide a pdf document with all the information required.

## PART 2 (BONUS)* (5 points)

Implement a classification algorithm in **Java** that is able to provide at least a **60%** of accuracy on test when working with the dataset given in the file **"koronia_dataset.csv"** (in the file, commas are used for decimals and semicolons to separate attributes). This dataset is related to a remotely sensed imagery problem.

The name of the attributes considered in the problem represented by the dataset are (from left to right in the "koronia_dataset.csv" file):

**blue**, **green**, **red**, **nearIR**, **conB**, **asmB**, **corB**, **homB**, **conG**, **asmG**, **corG**, **homG**, **conR**, **asmR**, **corR**, **homR**, **conIR**, **asmIR**, **corIR**, **homIR**, **brightness**, **greeness**, **wetness**, **intensity**, **hue** and **class**.

The algorithm must take as inputs two files (training and test), and must also return as outputs two different files, one for training results and another one for test results. Each one of these output files will show one line for each of the examples in the set (either training or test), showing the correct classification and the estimated classification (given by your algorithm). The file **examples/result0s0.tra** shows an example of the output for an algorithm when dealing with one partition of the "Iris" dataset in KEEL.

The algorithm must also prompt through the standard output the accuracy on training and test.

For PART 2, create a folder and include the Java source files together with the training and test partitions that your program uses. Please, provide also a ReadMe file specifying the commands to correctly run the program.

*Only in case that this part works completely, then you will receive full credit. Otherwise, you will receive no credit.