Departamento de Informática
Universidad Carlos III de Madrid

Bachelor in Data Science and Engineering

Machine Learning I
Course 2020-2021

Tutorial 3: Multiple Experimentation

KnowledgeFlow and Experimenter Weka Interfaces

February 27, 2023

- The aim of this tutorial is to get used to two Weka interfaces that facilitates the design and execution of multiple experiments at once: KnowledgeFlow and Experimenter.
- You can check the Weka webpage *http://www.cs.waikato.ac.nz/ml/weka/* to find more documentation and examples.

1 Exercise 1: KnowledgeFlow

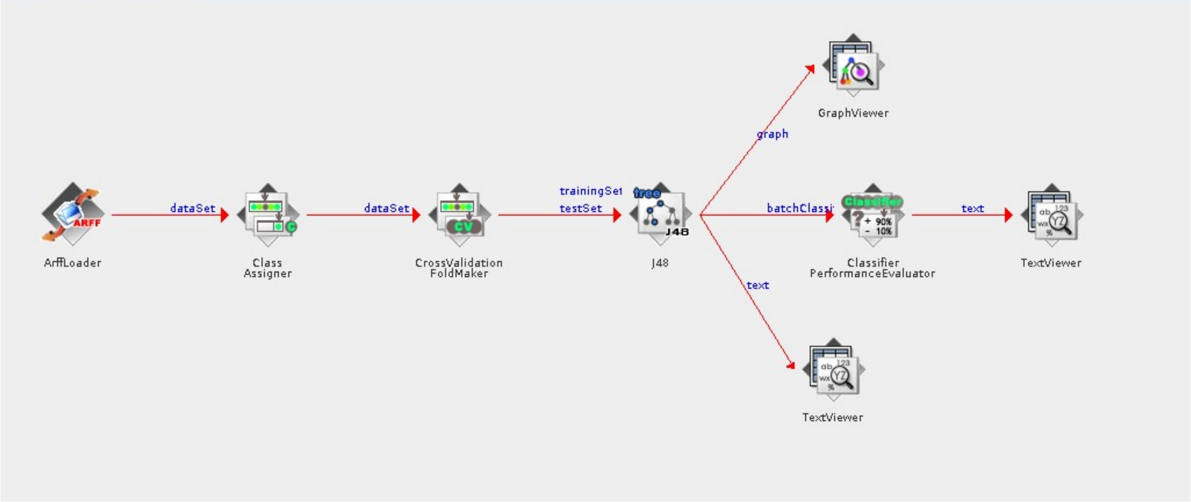

Figure 1: Knowledge flow to analyze the data in `adult-data.arff`.

1. Launch Weka.
2. Execute the KnowledgeFlow module. This interface allows to graphically design experiments. The user can choose the Weka components from a toolbar, place them in the screen and connect them to build a knowledge flow that load, process and analyze the data. The next sections explain how to build the knowledge flow shown in Figure 1.
3. Insert an `Arff Loader` node from the `DataSources` tab. Configure it to read the file `adult-data.arff` (right click on the node, `Configure` option).

4. Insert a `Class Assigner` node from the `Evaluation` tab. Link it with the previous node sending `dataSet` (right click on the previous node, `dataSet` option). Configure the node so the class is the variable called `salary` (it should be the by default option).

5. Insert a `Cross Validation FoldMaker` node from the `Evaluation` tab. Link it with the previous node sending `dataSet` (right click on the previous node, `dataSet` option). In the configuration we can choose the number of `Folds` (10 by default) and the random seed (1 by default).

6. Insert a `J48` node from the `Classifier` tab. Link it with the previous node sending `trainingSet` (right click on the previous node, `trainingSet` option). Link it again with the previous node sending `testSet`. It is necessary to end both the training and test set. Otherwise, the cross validation will not properly work.

7. Insert a `Classifier PerformanceEvaluator` node from the `Evaluation` tab. Link it with the previous node sending `batchClassifier`.

8. Insert a `TextViewer` node from the `Visualization` tab. Link it with the node `J48` sending `text`.

9. Insert a `TextViewer` node from the `Visualization` tab. Link it with the node `Classifier Performance Evaluator` sending `text`.

10. Insert a `GraphViewer` node from the `Visualization` tab. Link it with the node `J48`, sending `graph`.

11. Execute the KnowledgeFlow. To do so, you can either (1) press the start button placed in the top right part of the interface; or (2) click the `Start loading` option from the `Arff Loader` node. After that, select the option `Show Results` from the `TextViewer` nodes. What information is shown on each of them? What is percentage of correctly classified instances?

12. Save the knowledge flow as .kf or .kfml. Include both files in the submission.

13. What is the utility of creating knowledge flows with this Weka interface?

2 Exercise 2: Experimenter

This part of the tutorial focuses on building models that classify the direction Pac-Man is taken given a state of the game. This exercise is based on the basic feature extraction function programmed in tutorial 1.

2.1 Pac-Man data generation and pre-processing

1. Modify the printLineData() function you programmed in Tutorial 1, so it generates the .arff files[1] Weka receives as input. It should contain the proper header, and each line of data should contain an instance of the game state (with the attributes you consider relevant) plus the direction Pac-Man has just taken.

2. Generate a training data file called "all_data_pacman.arff" with more than 5000 instances. To do so, execute the game with the following parameters: *python busters.py -g RandomGhost -l openHunt*. Note that since we want you to collect the instances with the KeyBoardAgent (the one by default), you should write the printLineData() function for that particular agent. If the training data file exists, then you should only append the new lines. Do not write the header several times in the same file.

3. Generate two new files from the one you just have created. On each of them you should select a subset of different attributes. Save these files as "filter_data_pacman_manual1.arff" and "filter_data_pacman_manual2.arff".

By doing this, you should have 3 different .arff files with more than 5000 instances each.

2.2 Design and execution of the experiment

Now we proceed to design and execute the experiment using the Experimenter interface:

1. Launch Weka.

2. Open the *Experimenter*.

[1] http://www.cs.waikato.ac.nz/ml/weka/arff.html

3. Click on *New* to create a new experiment.
4. Select *Classification* as the type of experiment.
5. Select the three previous datasets.
6. Select the algorithms J48, IbK (with k=1,3,5), PART, ZeroR and NaiveBayes.
7. On *Results Destination* select ARFF and click on *Browse* to select the file. This file will contain the results and data from the experiment, and you will be able to open it as a spreadsheet when *Experimenter* finishes.
8. Save the experiment clicking on *Save*, choosing the name you want for the experiment.
9. Click the *Run* tan and press *Start*.

2.3 Results analysis

1. Click on the *Analyse* tab to analyze the results.
2. Click on *Experimenter* to select the results of the current experiment.
3. Select *Percent_correct* on the *Comparison field*, and then select *Perform_test*.
4. The v and * characters indicates if the result is statiscally better (v), worse (*) or equal ( ) than the base scheme, with the specified significance value (0.05 by default). The numbers between brackets (v/ /*) appearing below each scheme indicate the number of times a scheme is better, equal or worse than the base scheme.
5. Is there a data set that seems more appropriate?
6. Which algorithm do you think is the best?
7. Are the results of the best algorithm much better than the others?
8. Save in a file both the configuration and the results of the analysis.
9. What do you think the Weka Experimenter is suitable for?

3 Files to Submit

All the lab assignments **must** be done in groups of 2 people. You must submit a .zip file containing the required material through Aula Global before the following deadline: **Tuesday, March 7 at 8:00**. The name of the zip file must contain the last 6 digits of both student's NIA, `i.e., tutorial1-123456-234567.zip`

The zip file must contain the following files:

1. A **PDF** document with:
   - Cover page with the names and NIAs of both students.
   - Answers to all the questions.
   - Conclusions.
2. The files generated during this tutorial.

Please, **be very careful and respect the submission rules**.