uc3m
Departamento de Informática
Universidad Carlos III de Madrid

Bachelor in Data Science and Engineering

Machine Learning I

Course 2020-2021

Tutorial 2: Introduction to Weka

Data preprocessing and classification using Weka Explorer

February 14, 2022

- The aim of this tutorial is to get used to Weka Explorer.
- You can check the Weka webpage *http://www.cs.waikato.ac.nz/ml/weka/* to find more documentation and examples.
- It is important to solve the exercises in order.

1 Introduction

Weka (Waikato Environment for Knowledge Analysis) is an open source platform for machine learning and data mining. It is written in Java and is developed by the University of Waikato (New Zealand).

Figure 1: Weka is an endemic bird of New Zealand.

1.1 Weka Applications

Figure 2 shows the main interface of Weka with its four applications:

- **Explorer**: is the main workbench, where the data is loaded. It contains a set of components that allows to: apply different filters to the loaded data ("Preprocess"), use regression and classification algorithms ("Classify"), identify relation between attributes ("Associate"), use clustering techniques ("Cluster"), identify the most relevant attributes for the classification and regression tasks ("Select attributes"), and visualize some features about the attributes ("Visualize").
- **Experimenter**: it allows to systematically evaluate different algorithm's configurations over a data set.
- **KnowledgeFlow**: is a graphical interface with the same options as the Explorer.
- **Simple CLI**: a command line to execute all the Weka functions.

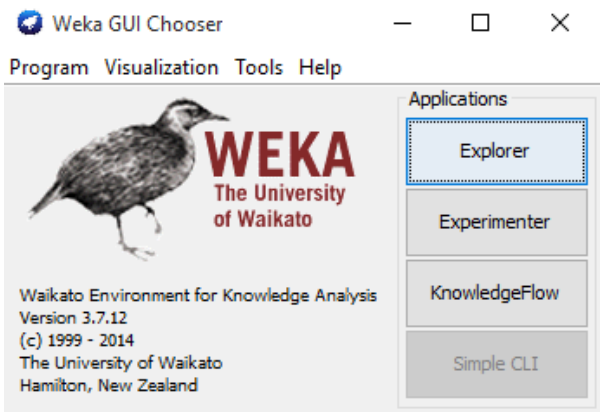

Figure 2: Interfaz principal de WEKA.

2 Instructions

You should answer each question in the given order, indicating the exercise number. **The document should not contain screenshots with code or the Weka output**. We strongly recommend you to make tables when comparing different algorithms, or the behavior of the same algorithm on different data sets. It is necessary to store each modified file separately, and attach them with the final document.

3 Exercises

Ex. 1: Data Files

- Download the data set `badges_plain.arff`.
- Open the file with any editor and study its content.

1. How many input attributes does the file have? What type are they?
2. Could a machine learning algorithm identify a function able to predict the class of an instance with the current data? Why?

Ex. 2: Classify with ID3

- Launch Weka.
- Open the Explorer.
- Open the file `badges_plain.arff`.

1. On the *Classify* tab, select the classifier *trees/ID3* [1]. On *Test options* select *Use training set* and press the *Start* button to generate the model. How good are the results?

Ex. 3: Generating new attributes

In Machine Learning, we can often improve the training data by selecting or extracting features from the raw data that we initially have. It is important to use the proper attributes for each task, so many times we will have to generate new attributes that have more information to improve the learning.

Based on the previous exercise:

1. Consider at least 6 attributes that you think may be relevant to solve this problem. These attributes should be extracted just from the previous attribute *name*. Write and explain them. Why have you chosen these attributes?

[1] In case you do not find it, you can include it through *tools/package manager*, installing the package *simpleEducationalLearningSchemes*

2. Open the file `badges1.arff` with Weka. How many input attributes does the file have? What type are they?
3. Which type of statistic information is shown? Press the "Visualize All" button. What do you see?
4. Generate a classifier using *tree/ID3*. What happen? What can you do to avoid this problem with ID3?

Ex. 4: Solving problems when using ID3

- Using the same data file, go back to the preprocess window.
- Select the attribute *name* and remove it.

1. Select the filter *Filter/unsupervised/attribute/Discretize*, fix the number of *bins* to 5, and apply the filter to the data set. What effect does this filter have?

- In the *Classifiers* tab, choose ID3 and set *Use training set* in the *Test options*. Generate the classifier again.

2. How many instances are correctly classified? What is the percentage?
3. What do you think the confusion matrix shows?
4. How many instances of each type are misclassified?
5. Click the *More Options* button and select the *Output Predictions (PlainText)* option. Classify it again and check the results. Which is the first misclassified instance? Why?
6. How would the instance "Donald Trump" be classified? Which are the attributes of this name? What happen to the values of this instance if you use the previous filter?

Ex. 5: Classifying with ZeroR

- In the *Classifiers* tab, choose *rules/ZeroR* y select *Use training set* in *Test options*. Generate the classifier again.

1. What model does ZeroR generate?
2. What is the success rate of this model?
3. How would the instance "Donald Trump" be classified?

Ex. 6: Classifying with J48 (C4.5)

- Go back to the preprocess window and load the file `badges1.arff` again.
- On the *Classifiers* tab choose J48 and select *Use training set* in *Test options*. Generate the classifier again.

1. How many leaves does the tree generated by J48 have?
2. How many instances are correctly classified?
3. What is the success rate?
4. How many instances of each type are incorrectly classified?
5. How would the instance "Donald Trump" be classified?
6. Would you prefer this model or the one generated by ID3? Why?
7. Have we found the perfect function to label the instances? Why?

Ex. 7: Using more attributes with J48 (C4.5)

1. Go back to the preprocess window and select the filter *Filter/unsupervised/attribute/AddExpression* to generate a new attribute that computes the number of vowels.
2. Store the data set `badges-Ej7.arff`.
3. Could you tell which is the most common range of vowels in the provided file?
4. Generate a new classifier with J48 and the previous data set. In this case you have to select the class in the drop-down menu of the *Classify* tab.
5. Write down the percentage of correctly classified instances and the confusion matrix.
6. Right click on the generated model that appears in *Result list*. Visualize the tree by clicking on *Visualize Tree*. What does the number on the leaves show?
7. Given these results, what features do you think attributes should have to maximize the success of machine learning algorithms?

Ex. 8: Balancing data, feature selection and other filters

1. Load in Weka the file `adult-data.arff`.
2. How many input attributes does the file have? How many training instances?
3. Execute the classifier J48. Select the option "Cross-validation" in Test Options. What results appear? Explain the result.
4. Now, evaluate the classifier only in instances from the `adult-test.arff` file. To do so, select "Supplied test set" in Test Options. What results appear? Are these results comparable with the previous ones? Why?
5. Go back to Preprocess and click the attribute salary. What proportion of the data is from each class? Do you think this percentage is suitable for a machine learning algorithm to properly learn?
6. Now we are going to modify the training instances to have a similar proportion of each class. To do so, select **supervised/instance/Resample** changing the parameter **biasToUniformClass** to 1.0 What does happen with the salary attribute? What does happen with the number of training instances?
7. After applying this filter, evaluate again with *cross-validation* and *supplied test set* the algorithm J48. What is the result? Is it better or worse?
8. Finally apply the standardisation filter **unsupervised/instance/Normalize** for the numeric attributes. What are the results?
9. After applying several filters to the data as you have just done, do you think this helps the learning process? Why?
10. What is the best results you have obtained? Justify it.
11. Save the data set as `badges-Ej8.arff`.

4 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: **Thursday, February 25th at 8:00am**. 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 of **no more than 10 pages** with:
   - Cover page with the names and NIAs of both students.
   - Answers to all the questions appearing in the Exercises section.
   - Conclusions.
2. The files generated during this tutorial: badges-Ej7.arff and badges-Ej8.arff

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