Instituto Superior de Engenharia do Porto

INTELIGÊNCIA ARTIFICIAL APLICADA (IAAPLI)

APPLIED ARTIFICIAL INTELLIGENCE

DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA

INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

POLITÉCNICO DO PORTO

Lab Classes Script:

Weka - A tool for data science
Weka - Uma ferramenta para a ciência de dados

Version Control

Version number	Date issued	Authors	Update information
V1.0	14 Jan 2022	Veríssimo Santos (VMS)	Original version of the script.

Table of Contents

1. Introduction

1.1 Context

Data science is a broad and multidisciplinary field that has become omnipresent in the modern world. Nowadays, almost there aren't processes where some data science subfield hasn't been used to improve it. Artificial intelligence, machine learning, data analytics, and big data, among many others, one or more of these data-driven fields has been or is highly probably currently being used to improve the state of the art in almost all fields of knowledge.

1.2 Motivation

To process and extract knowledge from data, a myriad of software frameworks and tools are available. Some frameworks like Python and R-based ones offer a plethora of tools suited to deal with both simple and complex data science (and subfields) problems. Their main strengths arise from the possibility to configure high-level parameters down to the smallest details of any kind of data science pipeline, but they come with the cost of frequently requiring solid programming skills and extensive knowledge of those smallest details involved. Another set of tools are the graphical interface-based ones. They offer an effective approach to learning and use data science and machine learning. Since they aim to solve data science problems via a graphical user interface, the general aspects of those problems can be configured, but their complexities are hidden from the user thus reducing the complexity of use and avoiding implementation mistakes. Weka is one of those tools and its characteristics make it perfect to be used in a variety of data science scenarios.

1.3 Objectives

This lab script aims to present and use Weka functionalities to view, study, inspect and correct datasets.

2. Weka

Weka (which stands for Waikato Environment for Knowledge Analysis) is a software package that contains a collection of tools and algorithms for data analysis and predictive modelling [1].

2.1 Key features

Among a myriad of data science software packages, WEKA has gained a solid position in the academia and enterprise environment due to key features such as:

- it enables fast testing and prototyping of ML algorithms;

- it supports a wide range of standard ML tasks;
- it has a decent GUI, with a smooth learning curve that enables quick and effective entry into the world of data processing, visualization, and ML algorithms;
- the user can test the algorithms on its datasets and in a significant set of built-in/existing datasets, resulting from contributions from the international community.
- it can be used not only from this graphical user interface (GUI), but also from the command line (useful to automate data processing), or using its Java API (to automate, add Weka features to other software or develop new features/models);
- it is bundled with a wide range of ML algorithms and new ones can be added (via GUI or programming interface);
- it enables access to widely known ML toolboxes, such as R and scikit-learn;
- it is licensed under the GNU General Public License;

While WEKA's GUI takes some time to master, it provides a very comfortable environment to experiment with, from the simplest to very complex ML algorithms. Note that despite the wide portfolio of algorithms already available, when a specific algorithm is not available, people are welcome to implement it and afterwards freely distribute it to the community.

2.2 Setup

Given that it is implemented in the Java programming language, installing Weka is a very straightforward process, as follows:

- In Microsoft Windows, just download the software package and install it.
- In Linux, just download and extract the software zip archive, enter the newly extracted directory and run: `./weka.sh`
- In macOS, download and open the `.dmg file`, copy the contents to a directory of your choice and open the `weka-3-8-6-azul-zulu-osx` file.

Any additional info about the installation procedure can be found here.

2.3 First steps

 After starting WEKA, the **WEKA GUI Chooser** window shows up on the PC screen:

Figure 1 – The Graphical User Interface (GUI) Chooser window

The WEKA GUI Chooser is the main interface, enabling access to the Weka Explorer, the Experimenter, the KnowleddgeFlow, the Workbench and the Simple CLI applications, as briefly summarised next:

- The **Explorer** is used to explore datasets and machine learning algorithms and analyze results. Its functionality it's going to be described in the following sections.
- The **Experimenter** is used for designing experiments for performance evaluation and comparison of several machine learning algorithms (on one or more datasets).
- The **KnowleddgeFlow** is "an advanced Explorer application" to graphically design ML pipelines (from the dataset to results synthesis), using a data-flow-inspired interface.
- The **Workbench** application aims at combining the functionality of **Explorer** and **Experimenter** into a single GUI.
- The **Simple CLI** is a command-line interface to access Weka ML algorithms. ML tasks can be automated by storing sets of commands in a batch file.

Additionally, in the **Tools Menu** of the `Weka GUi Chooser`, there are the:

Package Manager application: This application enables a convenient way to add/remove packages (models) to Weka.

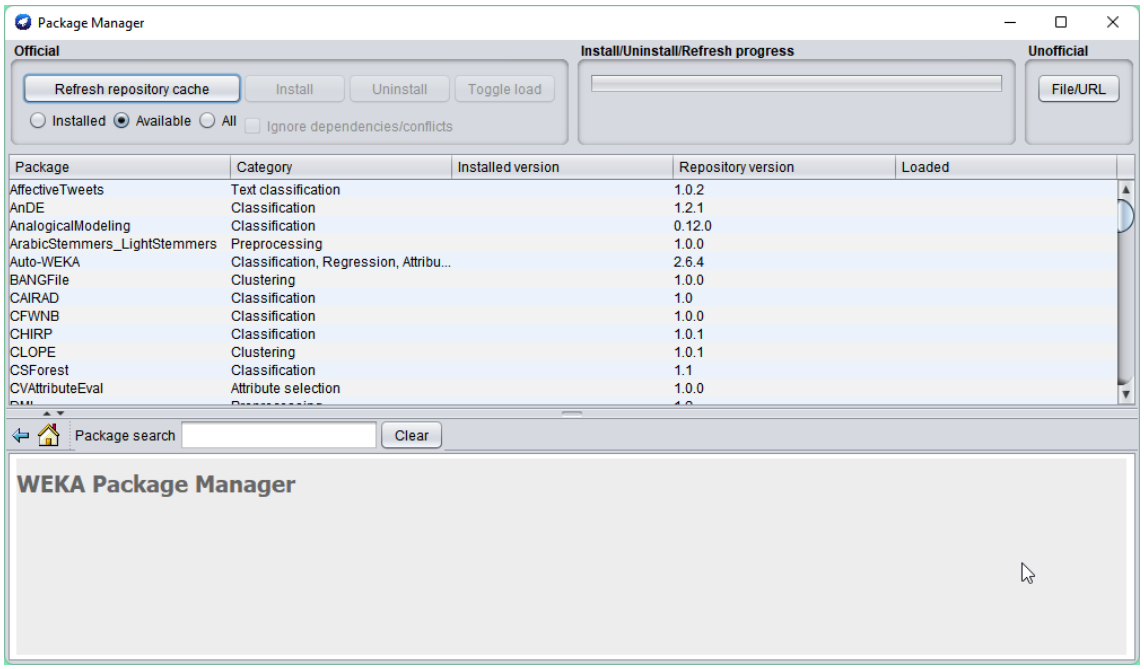

Figure 2 - Package Manager application

ARFF Explorer: This application provides a convenient way to **view/inspect/edit ARFF files**. This app provides a simple interface to edit the feature values, add/remove data instances (lines), and features (columns), among other operations.

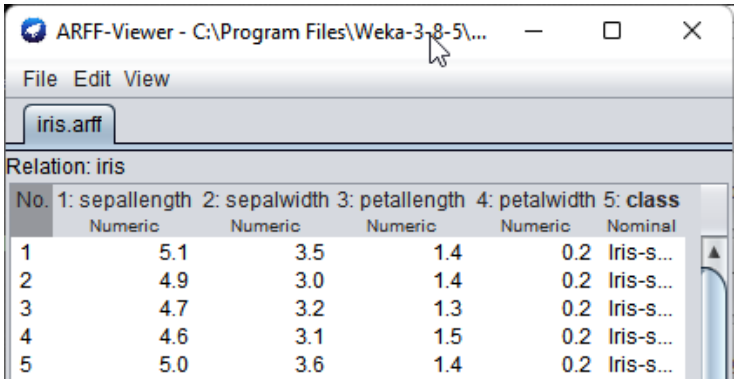

Figure 3 - ARFF Explorer

3. The Explorer application

The Explorer application is the main GUI and as the name says, allows exploring ML algorithms on built-in or user-added datasets. In this lab script we will focus our attention on the Explorer application, so in the Weka GUI Chooser click the Explorer button to open the Explorer window.

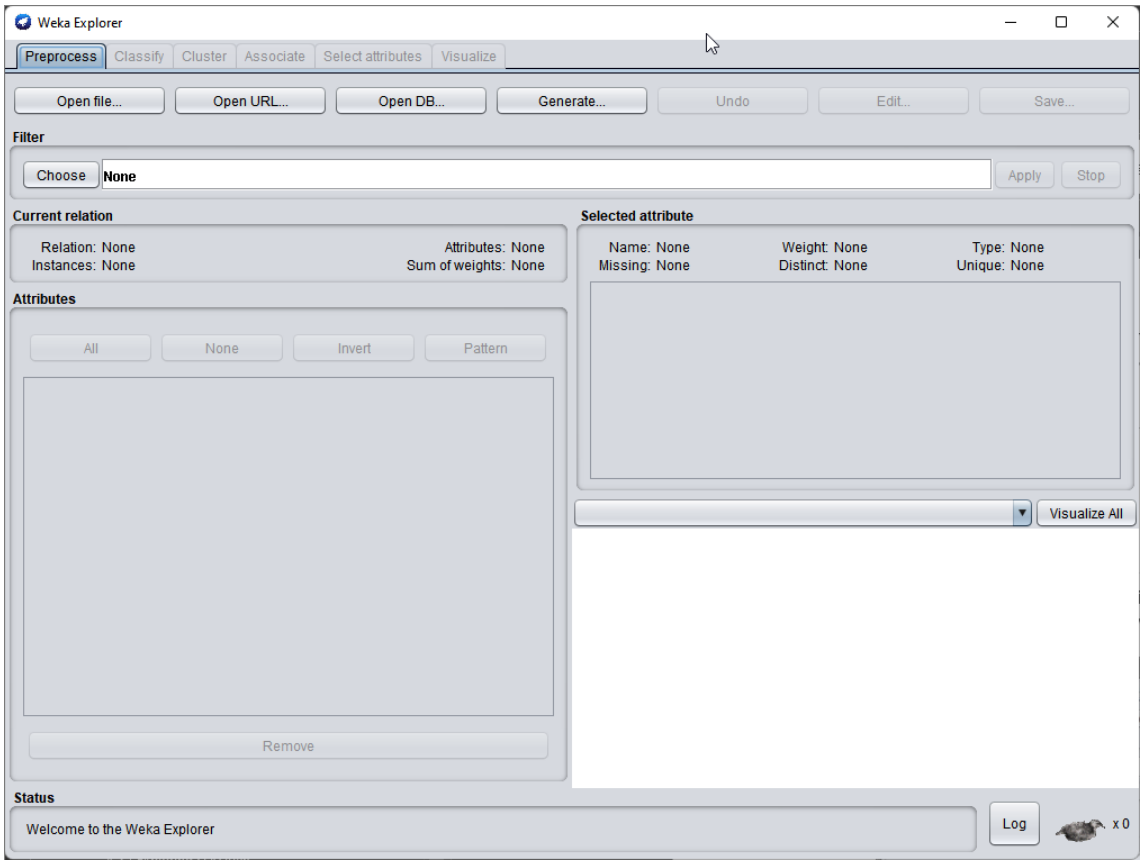

Figure 4 - The Weka Explorer window

4. Working with datasets

Data is collected to form datasets. These, are collections of instances, each of which is composed of a set of variables (called features) each describing one particular characteristic of the phenomenon/case of study. For example in sports data analytics related study, features can be a measure of the player's performance (number of scored points, maximum speed, maximum or mean strength, etc), physical characteristics (height, weight, abdominal diameter, bicep diameter, etc), team achievements (number of points scored/suffered per game, mean distance travelled by players, percentage of time used to defend/attack). In computer engineering features can be the idle/mean/max CPU time, the number of processes/threads per CPU core, the mean temperature for each CPU core, etc. In the image domain, common relevant features are related to colour and texture (MPEG-7 colour and texture descriptors, Haralick textural features, Local Binary Patterns).

First and foremost let’s understand how Weka handles data.

4.1 Weka data formats

Weka accepts data in a variety of data formats, but the default data format is the ARFF (Attribute-Relation File Format) [3].

In the **Weka Explorer Windows** on the **Preprocess tab** click on the **Open file …** button and in the new window that opens in the **Files of Type:** button to check all the supported data formats.

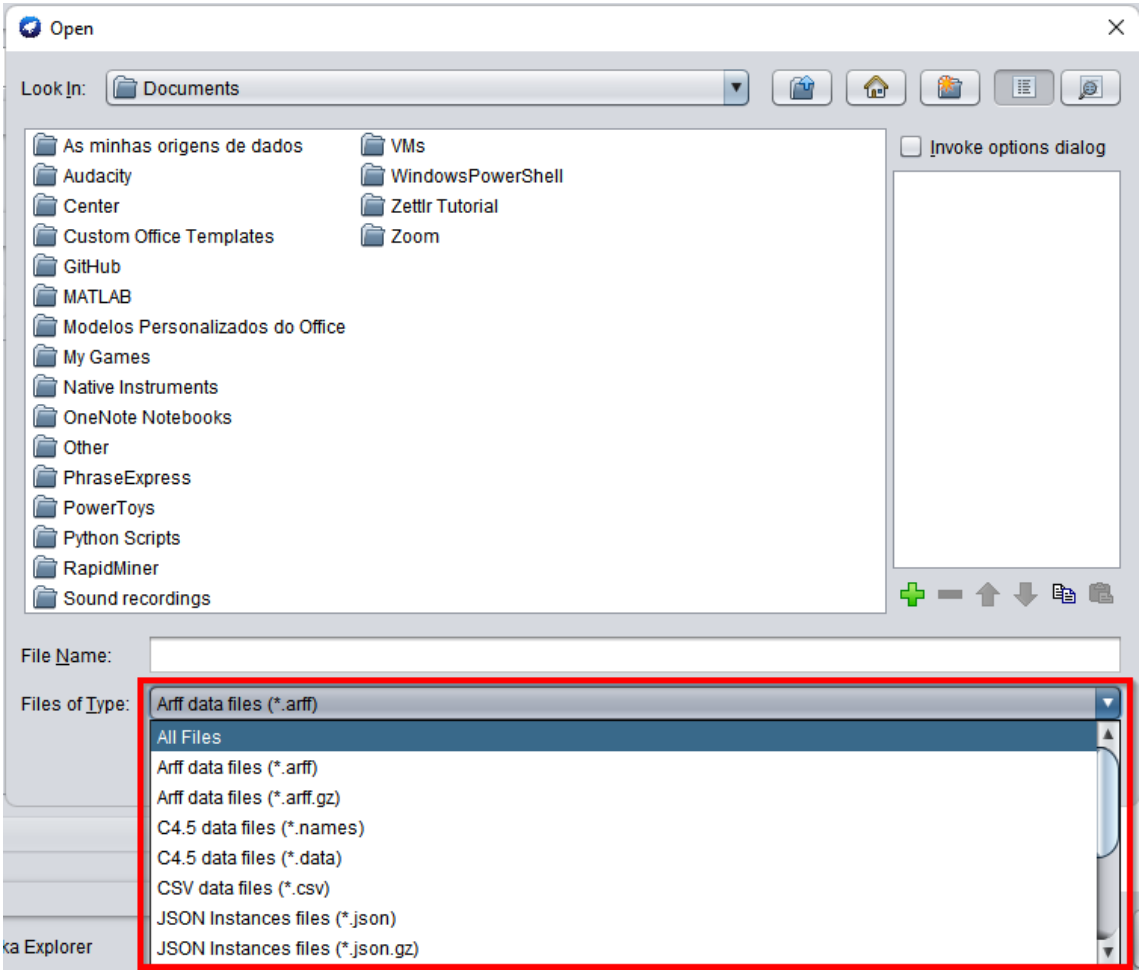

Figure 5 – Weka supported data formats

4.2 Attribute-relation file format (ARFF)

ARFF files are ASCII text files in which a dataset is described by listing all data instances and the corresponding set of attributes[1] using a specific format [3].

[1] Attributes are generically known as features, but in Weka frequently they are named this way. Both term will be used interchangibly in the scope of IAAPLI.

ARFF files are composed of two distinct sections. The first, the Header section, is where the dataset is characterized by presenting a list of attributes (the columns in the data section) and their corresponding data types. The second, the Data section, contains the actual data instances.

As an example, an extract from the standard IRIS dataset is displayed next[2]:

Table 1 - Extract from the IRIS dataset ARFF file

```
% 1. Title: Iris Plants Database
%
% 2. Sources:
%      (a) Creator: R.A. Fisher
%      (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
%      (c) Date: July, 1988
%
% 5. Number of Instances: 150 (50 in each of three classes)
%
% 6. Number of Attributes: 4 numeric, predictive attributes and the class
%
% 7. Attribute Information:
%    1. sepal length in cm
%    2. sepal width in cm
%    3. petal length in cm
%    4. petal width in cm
%    5. class:
%       -- Iris Setosa
%       -- Iris Versicolour
%       -- Iris Virginica
%

@RELATION iris

@ATTRIBUTE sepallength  NUMERIC
@ATTRIBUTE sepalwidth   REAL
@ATTRIBUTE petallength  INTEGER
@ATTRIBUTE petalwidth   REAL
@ATTRIBUTE class        {Iris-setosa,Iris-versicolor,Iris-virginica}

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
7.0,3.2,4.7,1.4,Iris-versicolor
6.3,3.3,6.0,2.5,Iris-virginica
```

Weka is bundled with a diverse set of datasets that are contained in the **data** folder inside the Weka installation folder.

Lines started with a `$ character` are comments.

The header section begins with the `@RELATION` statement with which the name of the dataset is declared. Attributes are declared in an ordered sequence, one in each line using the `@ATTRIBUTE` statement at the beginning of the line, followed by the name

[2] Note: The original arff file has been slightly modified to meet the explanation that follows.

and the data type. The order in which the attributes are declared indicates the column position in the data section of the file.

4.2.1 ARFF Valid datatypes

ARFF files currently support four data types (of attributes).

4.2.1.1.Numeric

Numeric attributes can be real or integer numbers (both are treated as NUMERIC).

```
@ATTRIBUTE sepallength  NUMERIC
@ATTRIBUTE sepalwidth   REAL
@ATTRIBUTE petallength  INTEGER
```

4.2.1.2.Nominal

```
@ATTRIBUTE class        {Iris-setosa,Iris-versicolor,Iris-virginica}
```

4.2.1.3.String

```
@ATTRIBUTE LCC    string
```

4.2.1.4.Date

```
@ATTRIBUTE “start date” DATE
```

The date default format is the ISO-8601 combined date and time format: `yyyy-MM-dd'T'HH:mm:ss`. Additional date formats can be specified via:

```
@ATTRIBUTE “start date” DATE [[date-format]]
```

5. Data sources

In Weka, data can be obtained from a variety of sources. In the **Preprocess tab** of the **Weka Explorer** (Figure 4) data can be imported directly from an **ARFF file** (click the `Open File …` button) or in **other valid formats** (listed in Figure 5), directly from an **URL** (click the `Open URL …` button) or from a **database** (click the `Open DB …` button). Additionally, there is a **data generator** (click the `Generate …` button) that can be used to generate synthetic data[3] using several data generation algorithms.

Despite all these format possibilities, whenever possible we will use the ARFF file format since it's native to the Weka Framework. Since data is not always available in this format, tools are necessary to perform this conversion. Several options are available. The native solution is to use the ARFF Viewer application to perform the conversion. However, this tool not always is capable of performing an effective conversion and so, other tools have

[3] Data that is generated by algorithms (not collected from real-world events).

been developed to help in this task. A quick search "arff converter" or "FORMAT[4] to arff converter" on the internet will return a nice set of options available. A very convenient and effective CSV to ARFF Converter is available online HERE.

6. Tutorial example

To explain the data visualization and preprocessing capabilities of the Weka Explorer we will use a tutorial example using the Iris Dataset which is bundled with Weka.

6.1 The baseline dataset - Iris

The Iris dataset (`iris.arff`) is a labelled dataset that has been collected by Edgar Anderson [7] and was introduced in 1936 by Ronald Fisher [8]. It contains 150 samples of Iris flowers, 50 of each of three species: *Iris setosa*, *Iris virginica* and *Iris versicolor* (see Figure 6 below).

Figure 6 – Three species of Iris [9]

Each data instance is described using the *length* and *width* of both the sepals and petals, measured in centimetres (see Figure 6). The features are `X={sepallength, sepalwidth, petallength, petalwidth}` and additionally, every instance has the corresponding class label (ground truth), i.e. the correct Iris species. You can learn more about this dataset here.

6.2 Opening the Iris dataset

To import the Iris dataset, press the `Open file …` button in the `preprocess tab` and select the `iris.arff` file (in windows: `C:\Program Files\Weka-3-8-6\data`).

[4] FORMAT: Replace for the format data is in.

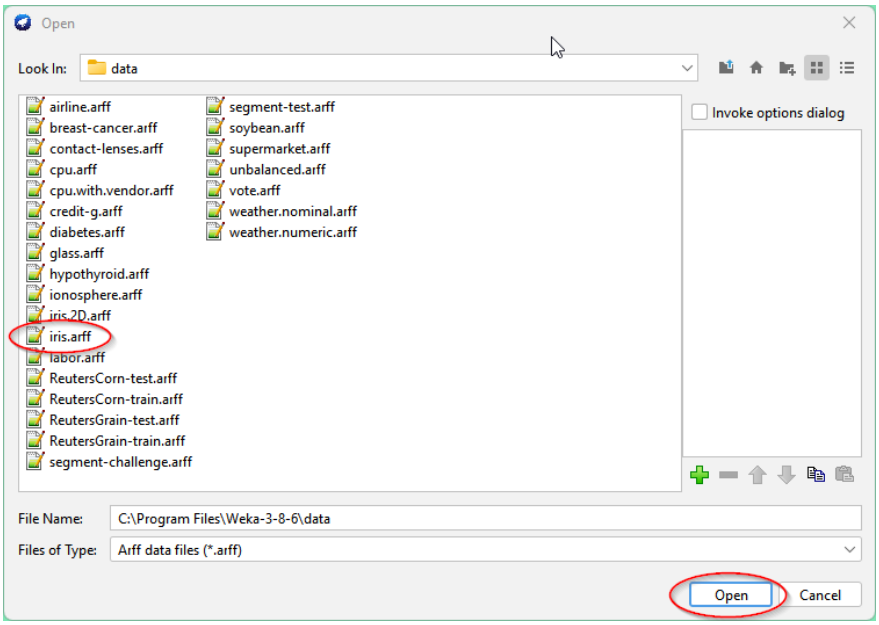

Figure 7 – Selecting and opening the Iris dataset

Upon loading the Iris dataset, the Explorer will look like this in Figure 8:

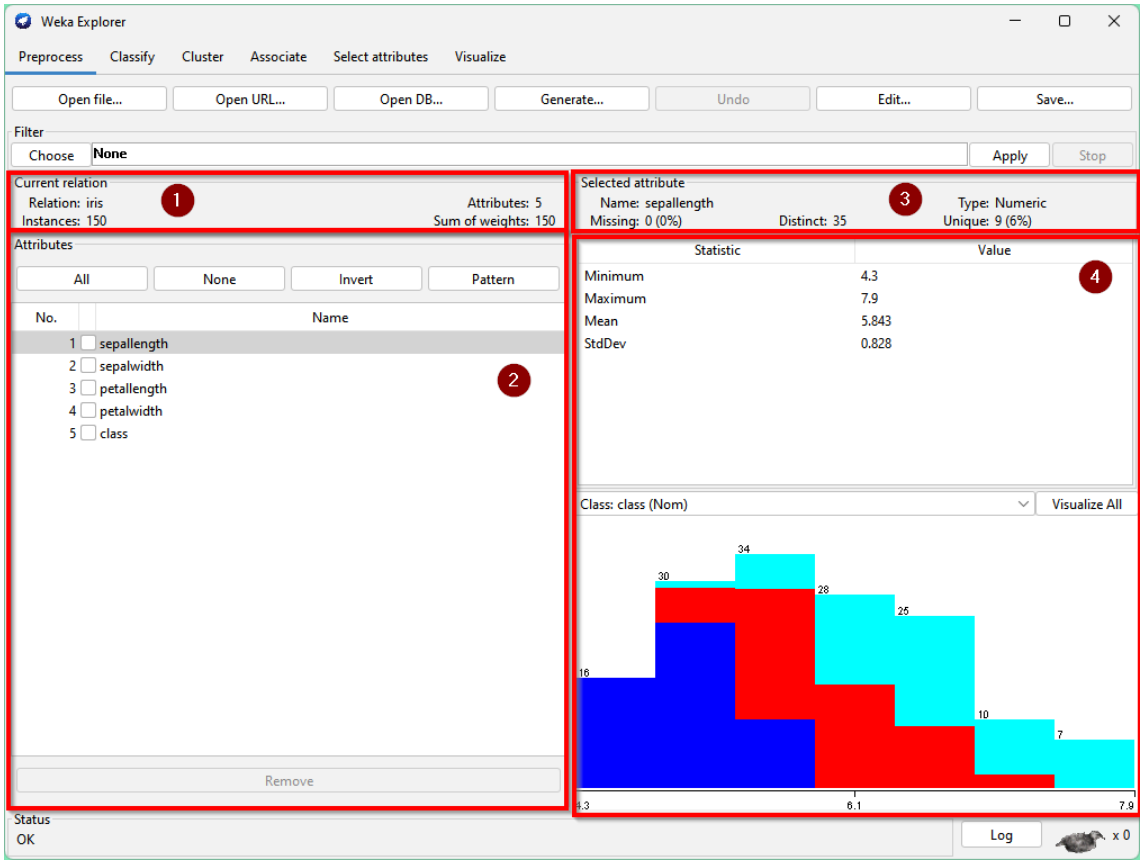

Figure 8 - Weka explorer - Preprocess tab - Iris dataset - general view

In the Explorer window (Figure 8), some properties of the current dataset (relation) are shown.

1 The name is `iris`, the number of instances is 150 and the number of features is 5.

2 Bellow, the attributes {`sepallength, sepalwidth, petallength, petalwidth, class`} are listed in the same order they are declared in the ARFF file.

Selecting each attribute with the mouse will display on the `Selected attribute` area 3 (upper right) its:

- **Name**;
- **Type** (numerical, nominal, etc);
- the number (and percentage) of data instances where this **attribute value is missing**;
- the **number of different values for this attribute** has on this dataset;
- the number (and percentage) of **data instances with a unique value for this attribute** (i.e. no other data instance has the same value for this attribute).

Bellow, if the attribute is numeric, **four statistics** (`minimum`, `maximum`, `mean` and `standard deviation`) describing the distribution of values will be shown. If the attribute is nominal, each value is listed with the number of instances that have that value.

4 Below, a histogram is shown displaying the distribution of the values of the selected attribute. The histogram is coloured according to any nominal attribute chosen as "Class" (This choice is done in the drop-down menu above the histogram). The order in which the attributes are listed in 3 is kept in the histogram (from the left to the right).

To check this select the `class` attribute.

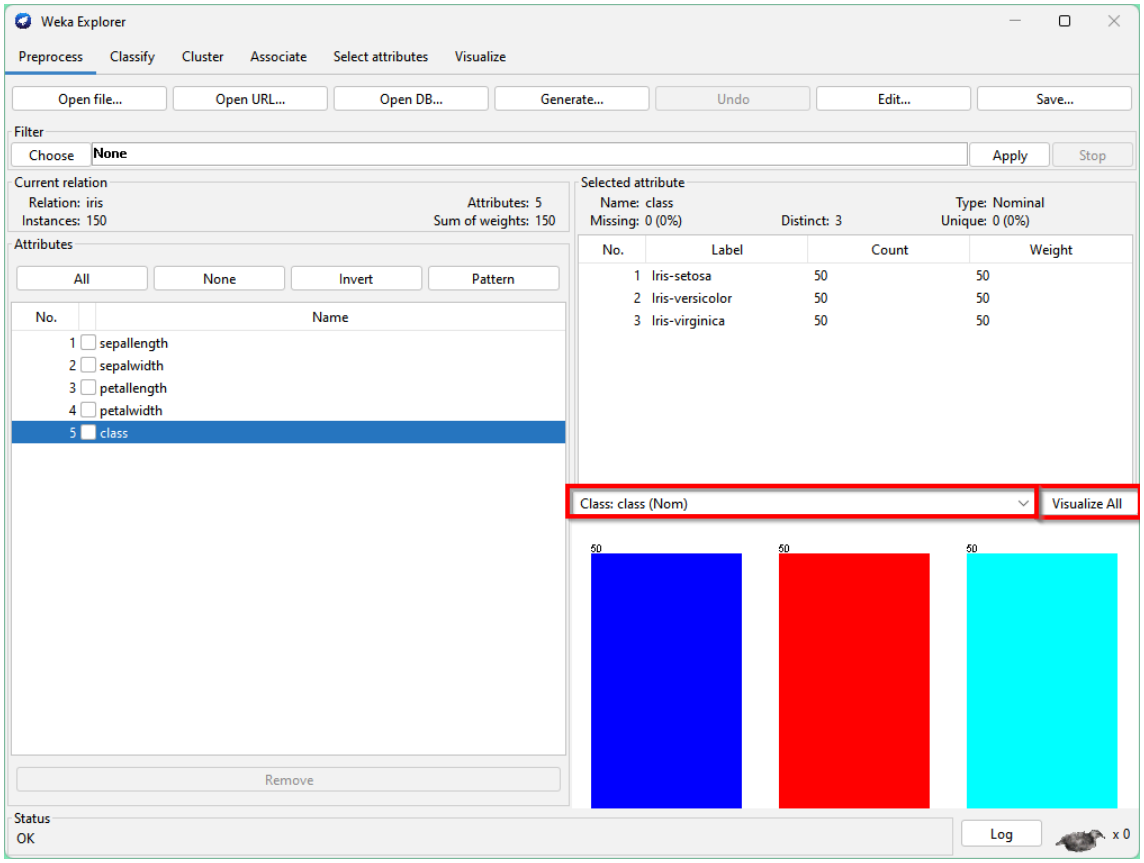

Figure 9 - Weka Explorer - Preprocess tab - Iris dataset - class attribute selected

In Figure 9 we can see that the first class (`iris-setosa`) is represented in blue, the second (`iris-versicolor`) in red, and the third (`iris-virginica`) in (cyan). Please note the confirmation of the existence of 50 instances (samples) of each class in the dataset.

For easier visualization click on the `Visualize All` button, to display in a separate window all the histograms for all the attributes in the data (Figure 10).

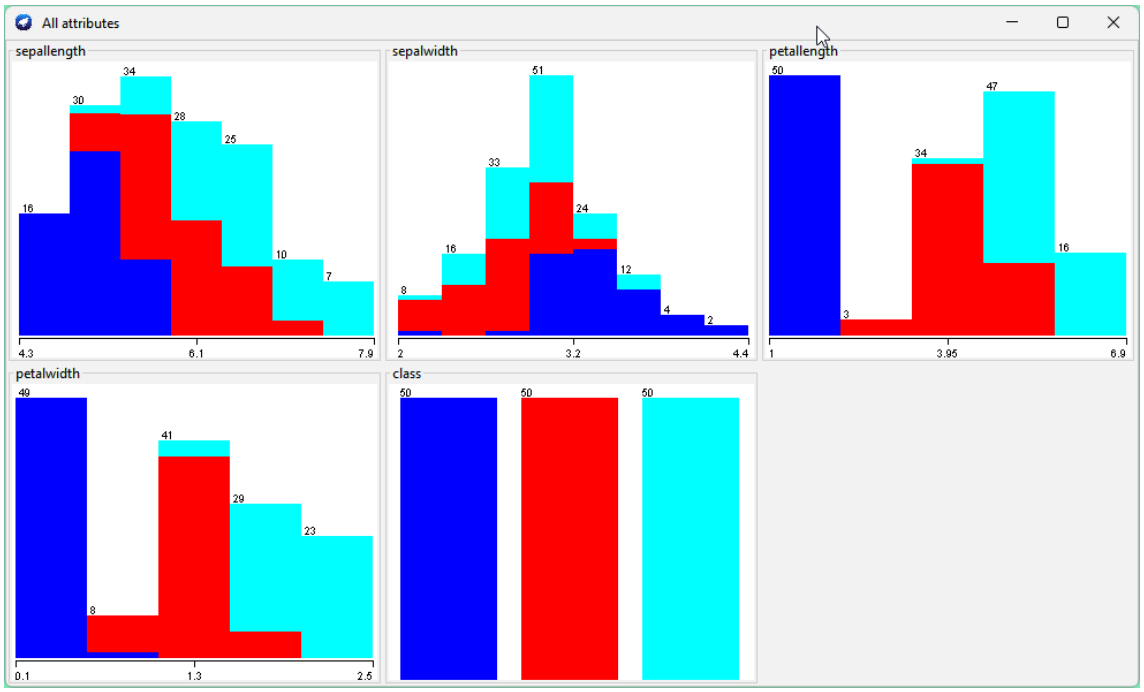

Figure 10 - Visualize the histograms for all attributes in the data

6.3 Visual data inspection

Figure 10 illustrates all class-wise histograms for each feature. Each histogram shows the values of the corresponding feature for each of the classes. By inspecting the feature values, can you find a threshold for one or more features that allow the separation (discrimination) between the Iris species?

For `sepallength`, `sepalwidth` there is considerable overlapping between the values of the three classes. However, only a few `petalwidth` values of the `iris-setosa` overlap the `iris-versicolor` ones. Looking at all histograms, the overlapping between the `iris-versicolor` and `iris-virginica` feature values is clear. Consequently, these two classes are less "separable" between themselves (i.e. the discriminative function will be more complex) while the `iris-setosa` is more easily separable from the other two (i.e. the discriminative function will be simpler).

Now let us dig deeper into the dataset. Change to the Weka Explorer `visualize tab` (Figure 11), and you will see the `Plot Matrix`, the plot controls and the association between colours and classes.

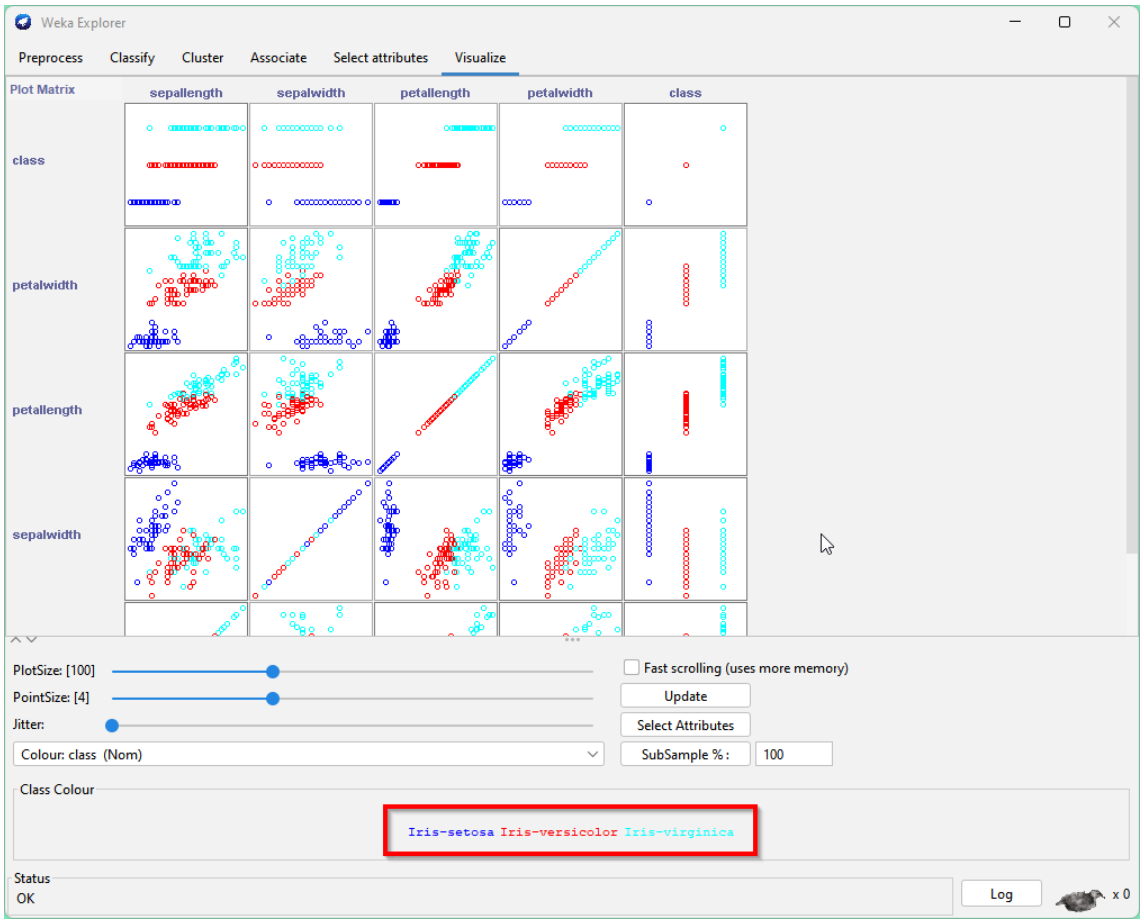

Figure 11 - Weka Explorer 'Visualize' tab – general view

If needed, for optimal visualization of the **scatter graphs** in the **plot matrix**, adjust the `PlotSize` and `PointSize` and do not forget to **click on the update button** (to update the graph) after adjusting the values in the sliders (Figure 12).

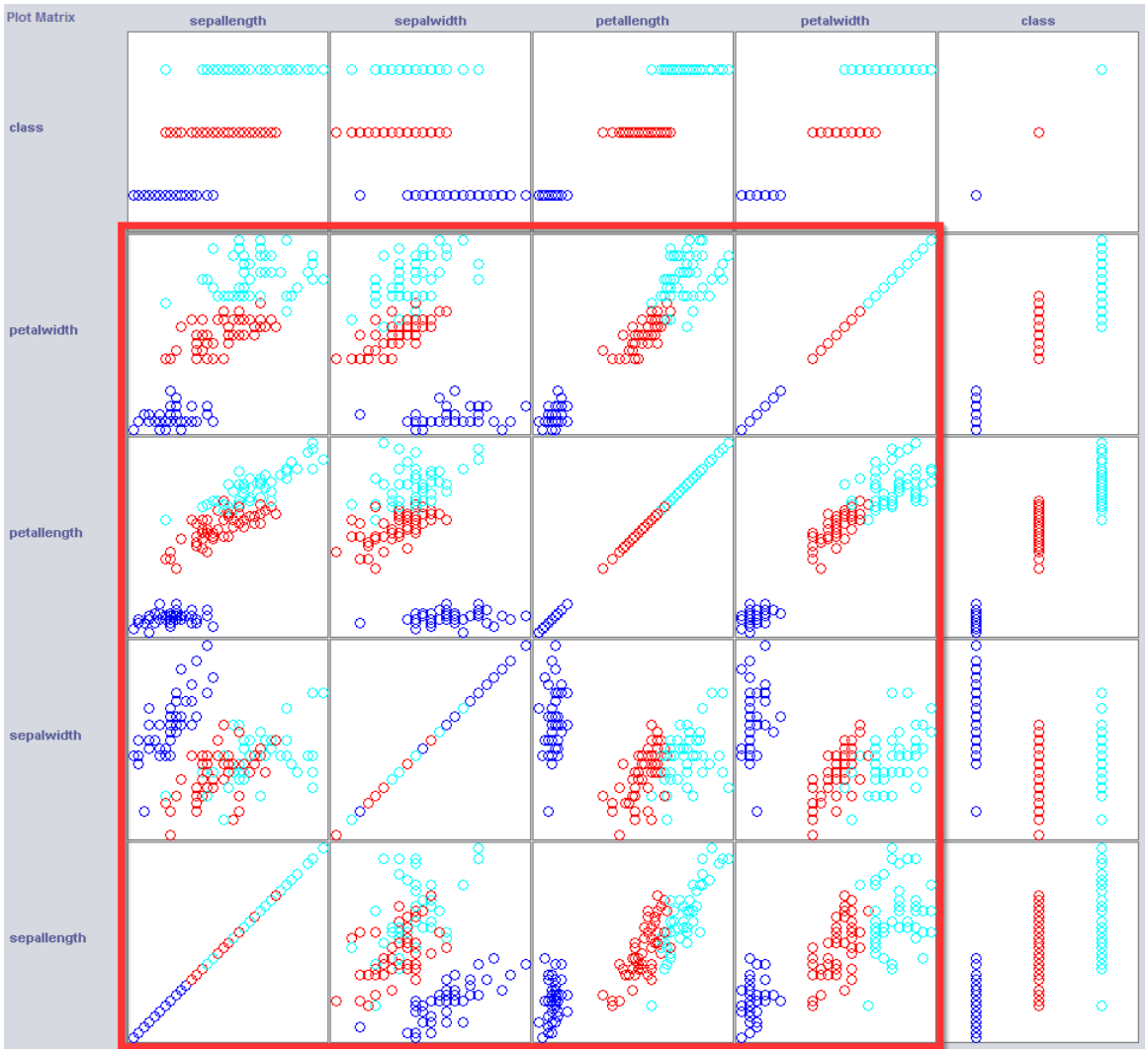

Figure 12 – Iris dataset Plot Matrix

The `Plot Matrix` is a matrix of **scatter graphs** displaying the relative distribution of each pair of features. The graphs on the secondary diagonal (from the top-right corner to the bottom-left corner) represent the same attribute for both axes.

These graphs display another perspective of the attribute distribution. We only need to consider the graphs below or above the secondary diagonal because they are repeated, given that the Plot Matrix is symmetrical over the secondary diagonal, i.e. if you fold it over this line the graphs overlap. Considering the upper part, in the first row given that class is one of the features involved, those graphs display a classwise distribution of each feature (the information that appears compiled in the histograms).

In the other rows are scatter graphs displaying the relative distribution of two pairs of different features. By inspecting these scatter graphs we can confirm that `iris-setosa` feature values “are more separable” than the other two species of Iris, given that a linear function is enough to separate (discriminate) its feature values from the other two classes.

Now, try to identify the most discriminative attributes, i.e. the ones that enable better class separation. Write them down and later we will see if your guess matches the algorithms… ☺

Note that we are using the class labels as the model will use during training. Nonetheless, when the model is deployed, labels are not available and the model must estimate the most likely class considering just the features. It is like looking at a grey scatter graph. Considering Figure 13 how easy is it to identify to which class each data instance belongs?

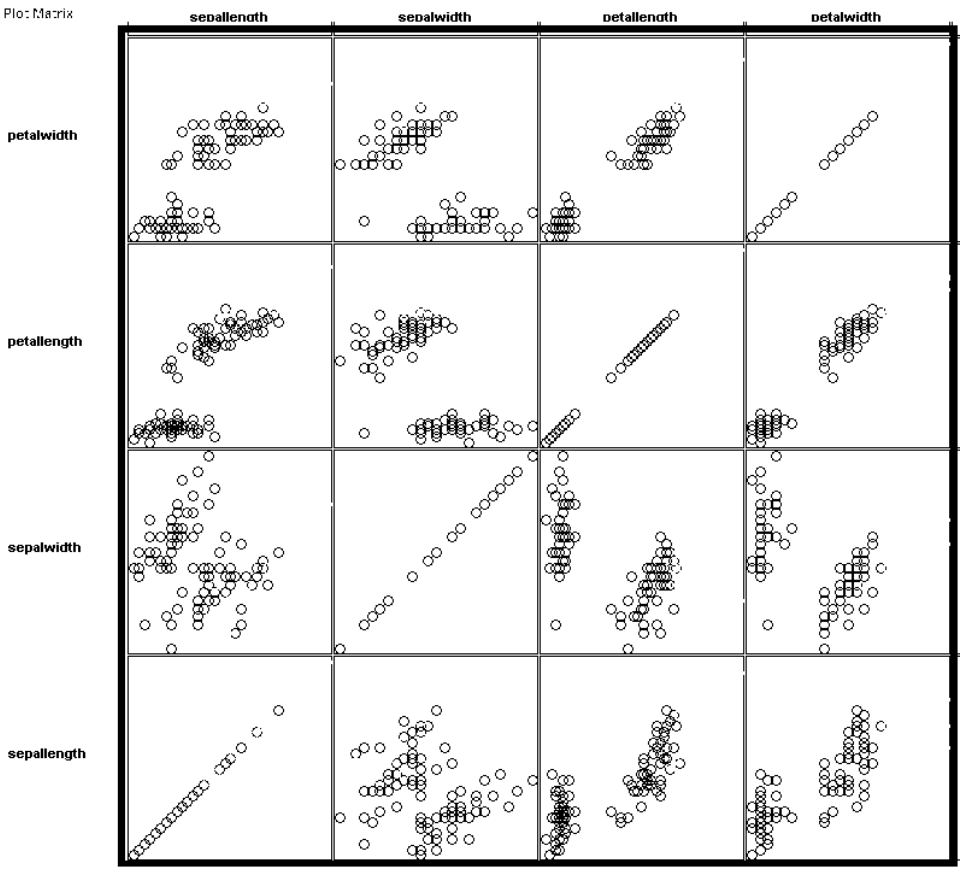

Figure 13 - Iris dataset Plot Matrix (without class information)

6.4 Assessing the discriminative potential of each attribute

At this point, we are going to use *Attribute Evaluator* algorithms to automatically identify the most discriminative features among the feature set (4 attributes). To evaluate that, go to the `Select Attributes tab`. You will get the window illustrated in Figure 14

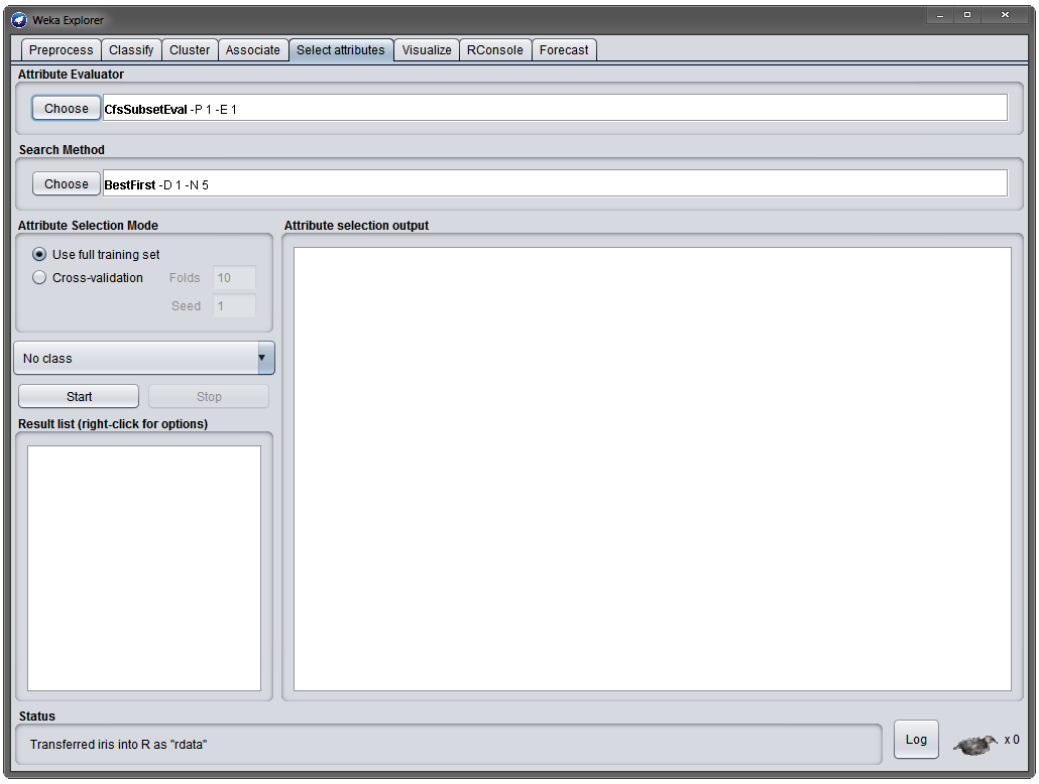

Figure 14 - Weka Explorer 'Select attributes' tab

Use the default attribute evaluator algorithm `CfsSubsetEval` [10] with the default options (`BestFirst`) and press the `Start` button.

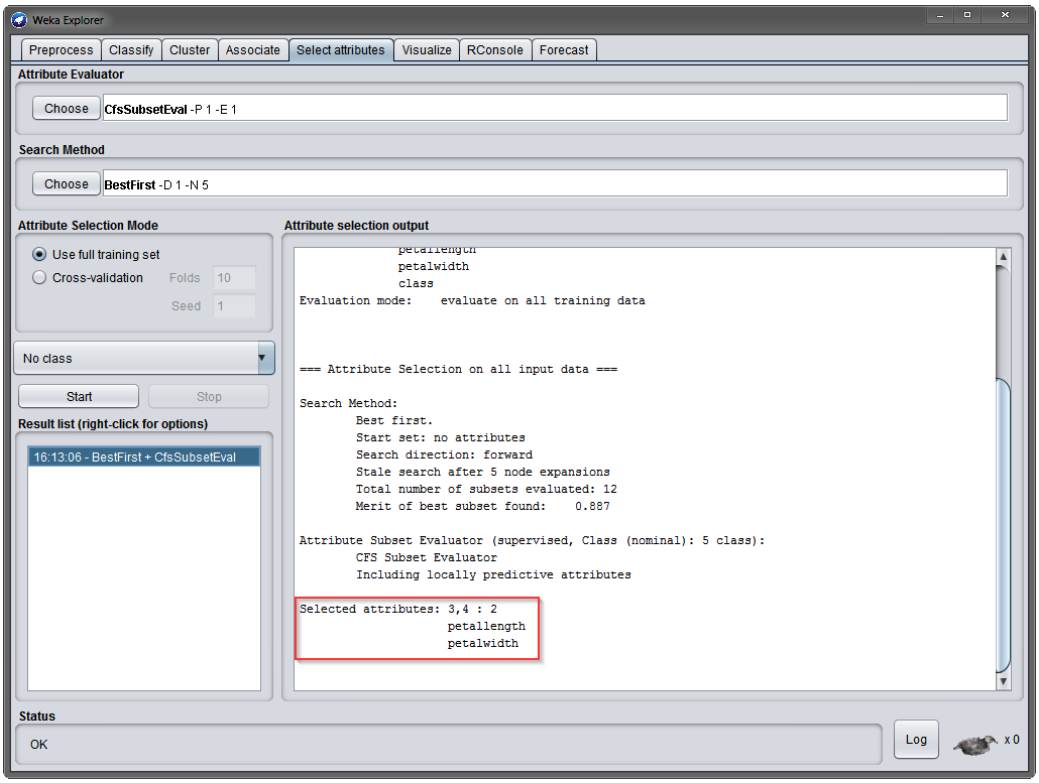

Figure 15 - Weka Explorer 'Select attributes' tab – `CfsSubsetEval` output

According to the `CfsSubsetEval`, the most discriminative features are `petallength` and `petalwidth`.

At this point, let us change the `search method` to `Ranker`. Let Weka change the evaluator for you when it prompts.

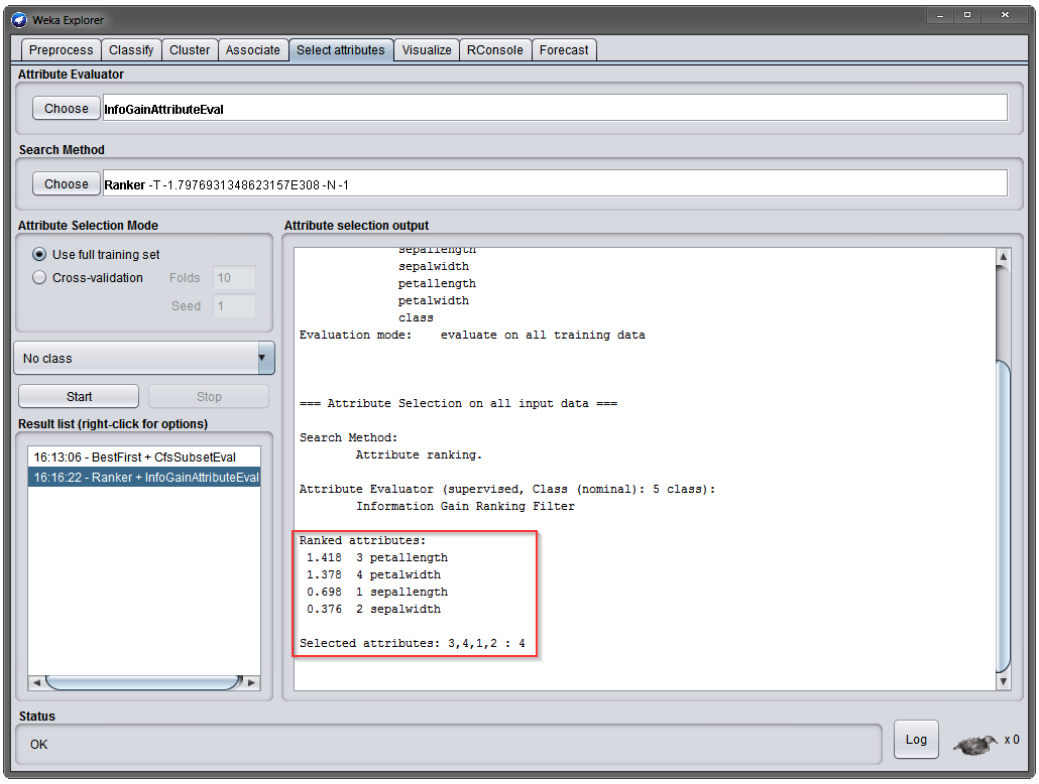

Figure 16 - Weka Explorer 'Select attributes' tab – Ranker algorithm output

The 'Ranker' assigns a rank to each feature, quantifying its discriminative capability [11]. According to the rank, both features `petallength` and `petalwidth` are individually more discriminative than `sepallength` and `sepalwidth` combined.

7. Exercises

To explore and apply the previously described concepts the following two examples are proposed in this section.

7.1 Problem A

Open the Weather dataset (`weather.nominal.arff`) in the Weka Explorer. This dataset has only five features and fourteen data instances. It's a register of the weather outlook, temperature, humidity, wind and if a person plays outside. The aim is to predict if the person plays outside depending on the other features. There are two versions of this dataset: the `weather.numeric.arff` with temperature and humidity with real values and the `weather.nominal.arff` with discretized nominal values (`temperature = {hot, mild, cool}` and `humidity={high, normal}`).

Carefully inspect the data histograms and visualize the scatter graphs available on the Weka Explorer `Visualize` tab and predict if the person will play based on the other features.

Dig you manage to do it? _____

Assess each attribute's discriminative potential using the previously mentioned algorithms available on the `Select Attributes` tab. Did your predictions match the algorithms? _____

7.2 Problem B

Download the **Simple Gender Classification** dataset (link). Convert it from CSV to the arff format. Open it on the Weka Explorer study the dataset, its features and their distribution and answer the following questions.

What's the goal of this dataset:

_____

_____

How many people are on the dataset: _____

Home many males: _____ and females: _____

How many education levels are considered: _____

What is the most frequent occupation in the dataset: _____

What is the most frequent occupation for males: _____

What is the most frequent occupation for females: _____

Describe the income distribution between genders:

_____
_____
_____
_____

Describe the income distribution between genders considering the education level:

_____
_____
_____
_____

Rank the most relevant attributes to identify the class:

_____

Remember that in the Weka Explorer, you can select the variable you want to use as a class. By selecting `no class´ the class information will be ignored and the histograms

will appear in black and white. Without this information can you identify the class by looking at the different features in the data? This is precisely what machine learning does! Create models from data and detect patterns in it, enabling conclusions such as the prediction of the gender (classification) from a set of diverse data variables (features) to be drawn.

8. Challenge

In a posterior lab script, a predictive model will be trained and used to automatically classify data instances. As humans are curious by nature, so go to the `Classify` tab and for each of the previously used datasets select and train a Multilayer Perceptron (MLP) Artificial Neural Network (ANN) to classify a subset of samples from the dataset. How about now? How well do you believe each model performed in each task?

9. References

[1] The University of Waikato, 'Weka 3: Machine Learning Software in Java', 2015. http://www.cs.waikato.ac.nz/ml/weka/

[2] The University of Waikato, 'Weka 3: Machine Learning Software in Java'. 2015. [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/

[3] 'Machine Learning Project', 'Attribute-relation file format (ARFF)', *Dep. Comput. Sci. Univ. Waikato*.

[4] 'electricity @ datahub.io'. https://datahub.io/machine-learning/electricity

[5] M. Harries, 'SPLICE-2 Comparative Evaluation: Electricity Pricing', 1999.

[6] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, 'Learning with drift detection', *Lect. Notes Comput. Sci. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinforma.*, vol. 3171, no. September, pp. 286–295, 2004, doi: 10.1007/978-3-540-28645-5_29.