Chapter 6

A Tour of the Weka Machine Learning Workbench

Weka is an easy to use and powerful machine learning platform. It provides a large number of machine learning algorithms, feature selection methods and data preparation filters. In this lesson you will discover the Weka machine learning workbench and take a tour of the key interfaces that you can use on your machine learning projects. After reading this lesson you will know about:

- The interfaces supported by the Weka machine learning workbench.
- Those interfaces that are recommended for beginners to work through their problems, and those that are not.
- How to at least click through each key interface you will need in Weka and generate a result.

Let's get started.

6.1 Weka GUI Chooser

The entry point into the Weka interface is the *Weka GUI Chooser*. It is an interface that let's you choose and launch a specific Weka environment.

Figure 6.1: Screenshot of the *Weka GUI Chooser*.

In addition to providing access to the core Weka tools, it also has a number of additional utilities and tools provided in the menu. There two important utilities to note in the *Tools* menu:

- 1. The *Package Manager* which let's you browse and install third party add-ons to Weka such as new algorithms.

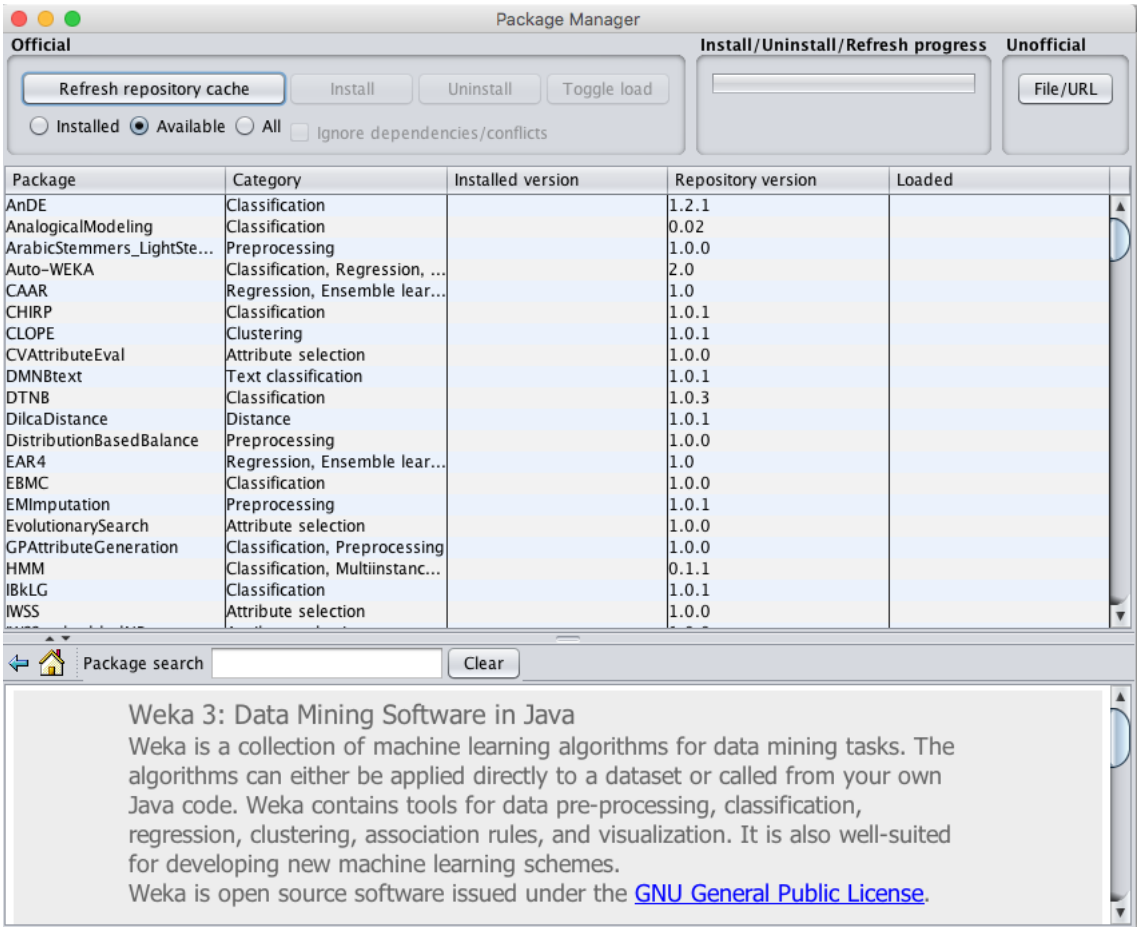
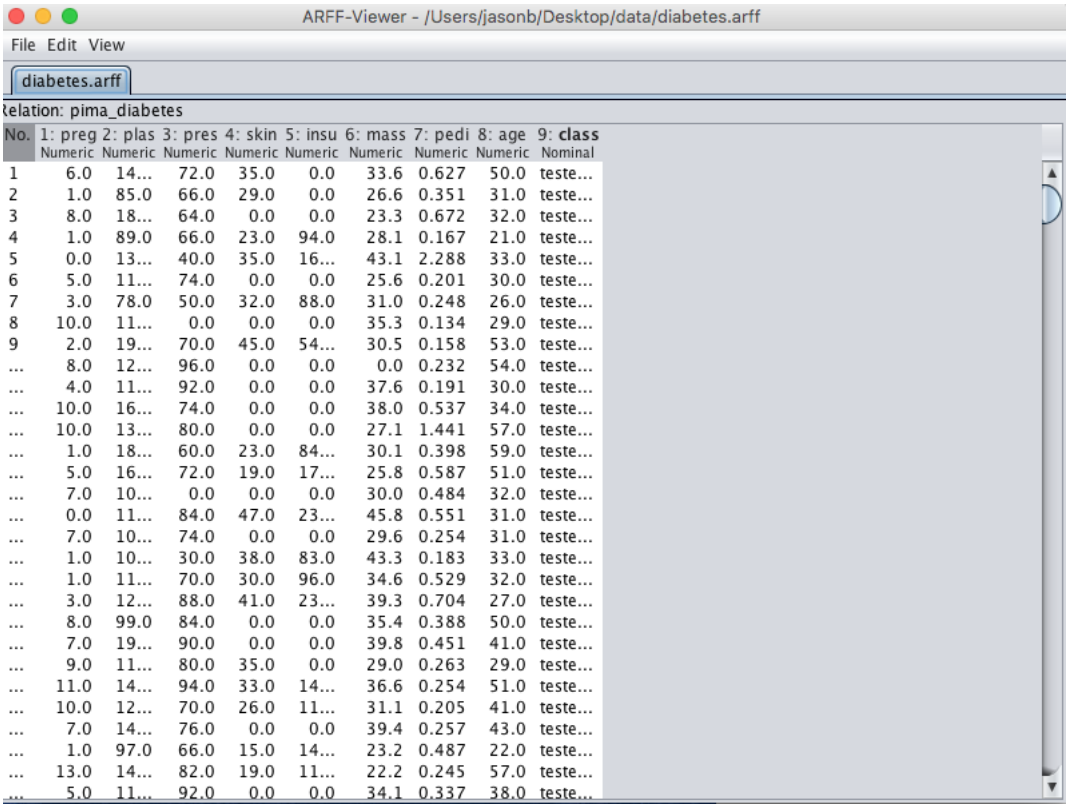


Figure 6.2: Screenshot of the Weka *Package Manager*.

- 2. The *ARFF-Viewer* that allows you to load and transform datasets and save them in ARFF format.



No.	1: preg Numeric	2: plas Numeric	3: pres Numeric	4: skin Numeric	5: insu Numeric	6: mass Numeric	7: pedi Numeric	8: age Numeric	9: class Nominal
1	6.0	14...	72.0	35.0	0.0	33.6	0.627	50.0	teste...
2	1.0	85.0	66.0	29.0	0.0	26.6	0.351	31.0	teste...
3	8.0	18...	64.0	0.0	0.0	23.3	0.672	32.0	teste...
4	1.0	89.0	66.0	23.0	94.0	28.1	0.167	21.0	teste...
5	0.0	13...	40.0	35.0	16...	43.1	2.288	33.0	teste...
6	5.0	11...	74.0	0.0	0.0	25.6	0.201	30.0	teste...
7	3.0	78.0	50.0	32.0	88.0	31.0	0.248	26.0	teste...
8	10.0	11...	0.0	0.0	0.0	35.3	0.134	29.0	teste...
9	2.0	19...	70.0	45.0	54...	30.5	0.158	53.0	teste...
...	8.0	12...	96.0	0.0	0.0	0.0	0.232	54.0	teste...
...	4.0	11...	92.0	0.0	0.0	37.6	0.191	30.0	teste...
...	10.0	16...	74.0	0.0	0.0	38.0	0.537	34.0	teste...
...	10.0	13...	80.0	0.0	0.0	27.1	1.441	57.0	teste...
...	1.0	18...	60.0	23.0	84...	30.1	0.398	59.0	teste...
...	5.0	16...	72.0	19.0	17...	25.8	0.587	51.0	teste...
...	7.0	10...	0.0	0.0	0.0	30.0	0.484	32.0	teste...
...	0.0	11...	84.0	47.0	23...	45.8	0.551	31.0	teste...
...	7.0	10...	74.0	0.0	0.0	29.6	0.254	31.0	teste...
...	1.0	10...	30.0	38.0	83.0	43.3	0.183	33.0	teste...
...	1.0	11...	70.0	30.0	96.0	34.6	0.529	32.0	teste...
...	3.0	12...	88.0	41.0	23...	39.3	0.704	27.0	teste...
...	8.0	99.0	84.0	0.0	0.0	35.4	0.388	50.0	teste...
...	7.0	19...	90.0	0.0	0.0	39.8	0.451	41.0	teste...
...	9.0	11...	80.0	35.0	0.0	29.0	0.263	29.0	teste...
...	11.0	14...	94.0	33.0	14...	36.6	0.254	51.0	teste...
...	10.0	12...	70.0	26.0	11...	31.1	0.205	41.0	teste...
...	7.0	14...	76.0	0.0	0.0	39.4	0.257	43.0	teste...
...	1.0	97.0	66.0	15.0	14...	23.2	0.487	22.0	teste...
...	13.0	14...	82.0	19.0	11...	22.2	0.245	57.0	teste...
...	5.0	11...	92.0	0.0	0.0	34.1	0.337	38.0	teste...

Figure 6.3: Screenshot of the Weka *ARFF-Viewer*.

6.2 Weka Explorer

The *Weka Explorer* is designed to investigate your machine learning dataset. It is useful when you are thinking about different data transforms and modeling algorithms that you could investigate with a controlled experiment later. It is excellent for getting ideas and playing what-if scenarios. The interface is divided into 6 tabs, each with a specific function:

The *Preprocess* tab is for loading your dataset and applying filters to transform the data into a form that better exposes the structure of the problem to the modeling processes. Also provides some summary statistics about loaded data. Load a standard dataset in the `data/` directory of your Weka installation, specifically `data/breast-cancer.arff`. This is a binary classification problem that we will use on this tour. You will learn more about this dataset in Section 8.2.2.

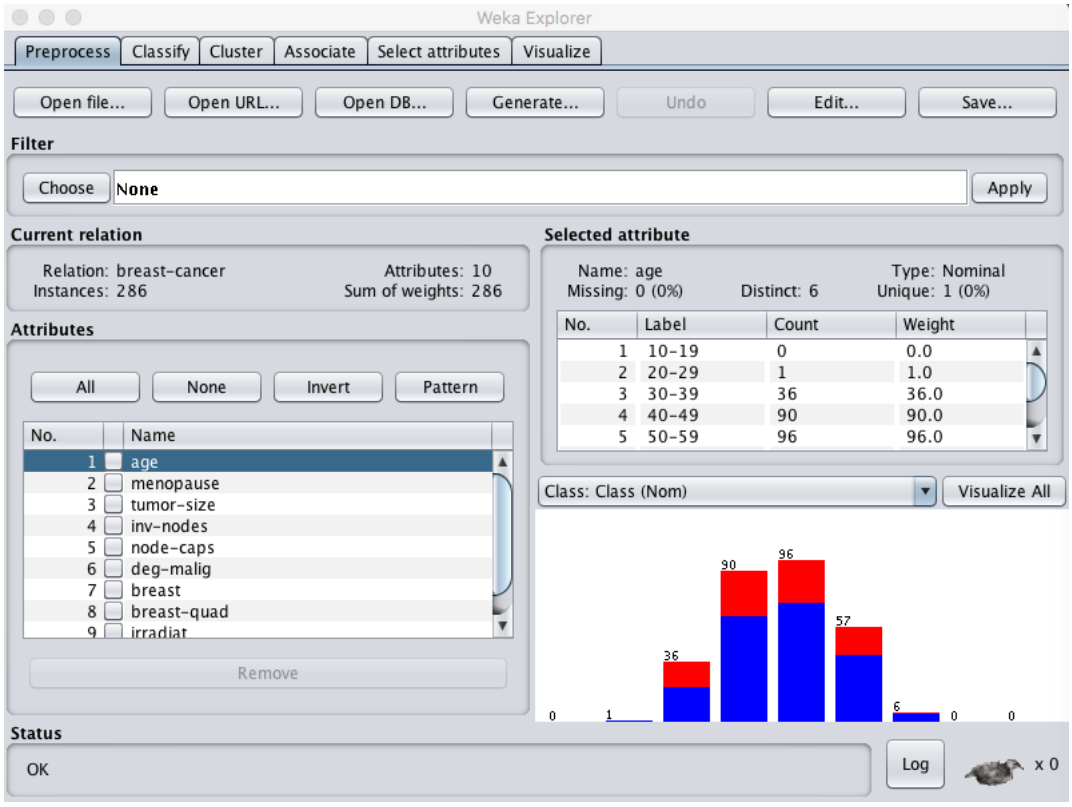


Figure 6.4: Screenshot of the *Weka Explorer Preprocess* Tab.

The *Classify* tab is for training and evaluating the performance of different machine learning algorithms on your classification or regression problem. Algorithms are divided up into groups, results are kept in a result list and summarized in the main *Classifier output* pane.

- Click the *Start* button to run the *ZeroR* classifier on the dataset and summarize the results.

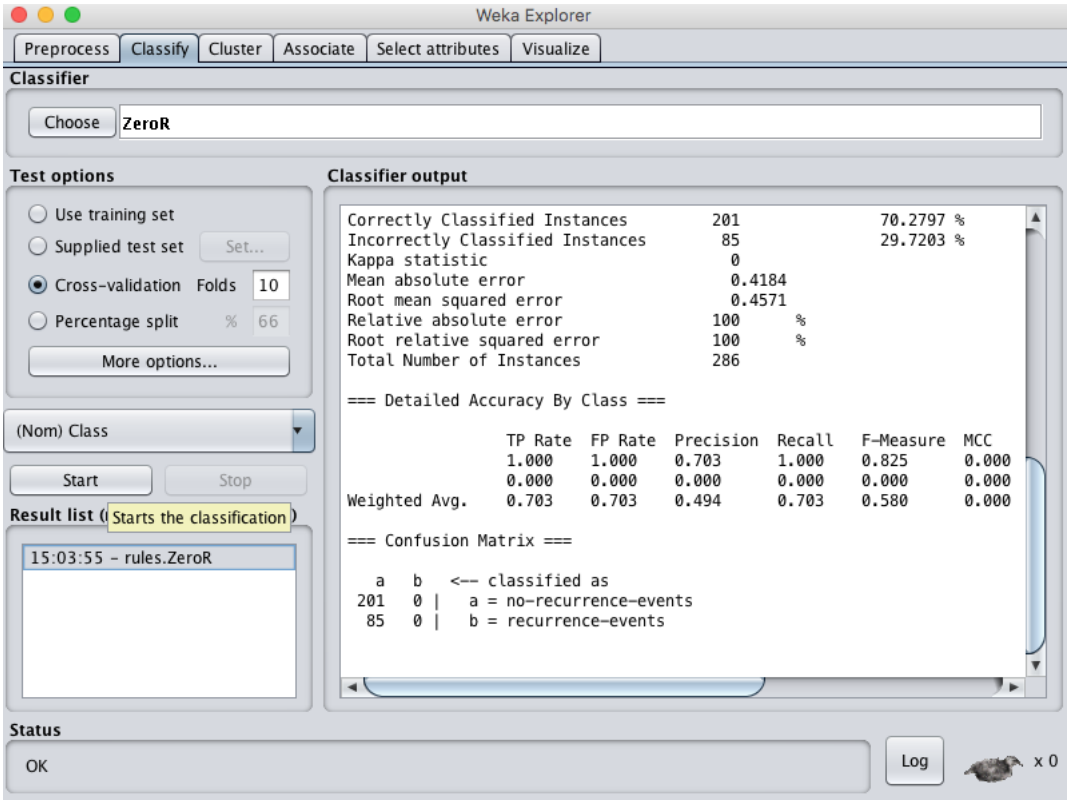


Figure 6.5: Screenshot of the *Weka Explorer Classify* Tab.

The *Cluster* tab is for training and evaluating the performance of different unsupervised clustering algorithms on your unlabeled dataset. Like the *Classify* tab, algorithms are divided into groups, results are kept in a result list and summarized in the main *Clusterer output* pane.

- Click the *Start* button to run the *EM* clustering algorithm on the dataset and summarize the results.

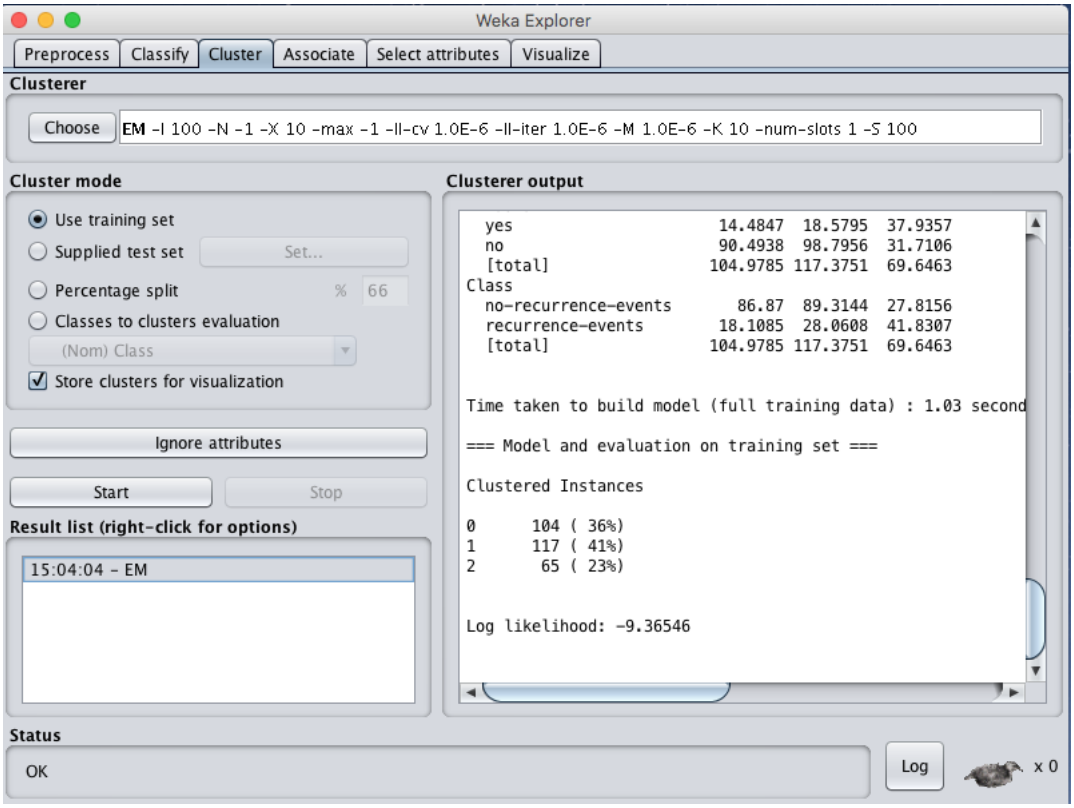


Figure 6.6: Screenshot of the *Weka Explorer Cluster* Tab.

The *Associate* tab is for automatically finding associations in a dataset. The techniques are often used for market basket analysis type data mining problems and require data where all attributes are categorical.

- Click the *Start* button to run the *Apriori* association algorithm on the dataset and summarize the results.

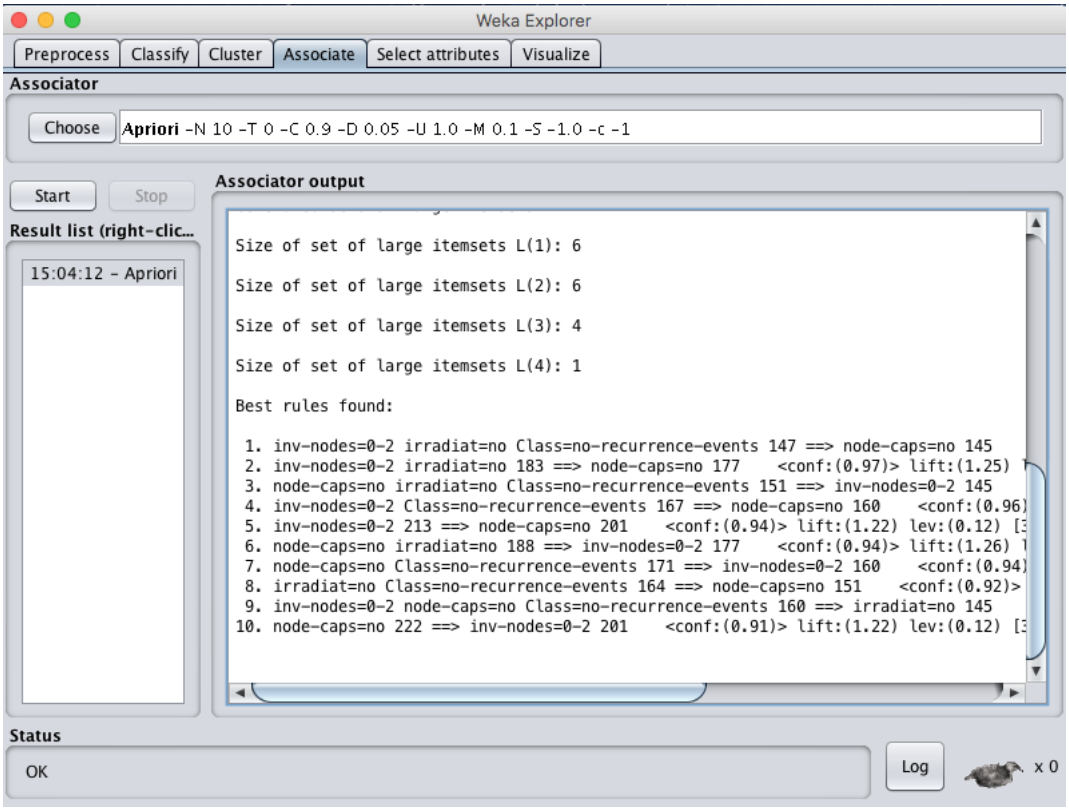


Figure 6.7: Screenshot of the *Weka Explorer Associate* Tab.

The *Select attributes* tab is for performing feature selection on the loaded dataset and identifying those features that are most likely to be relevant in developing a predictive model.

- Click the *Start* button to run the *CfsSubsetEval* algorithm with a *BestFirst* search on the dataset and summarize the results.

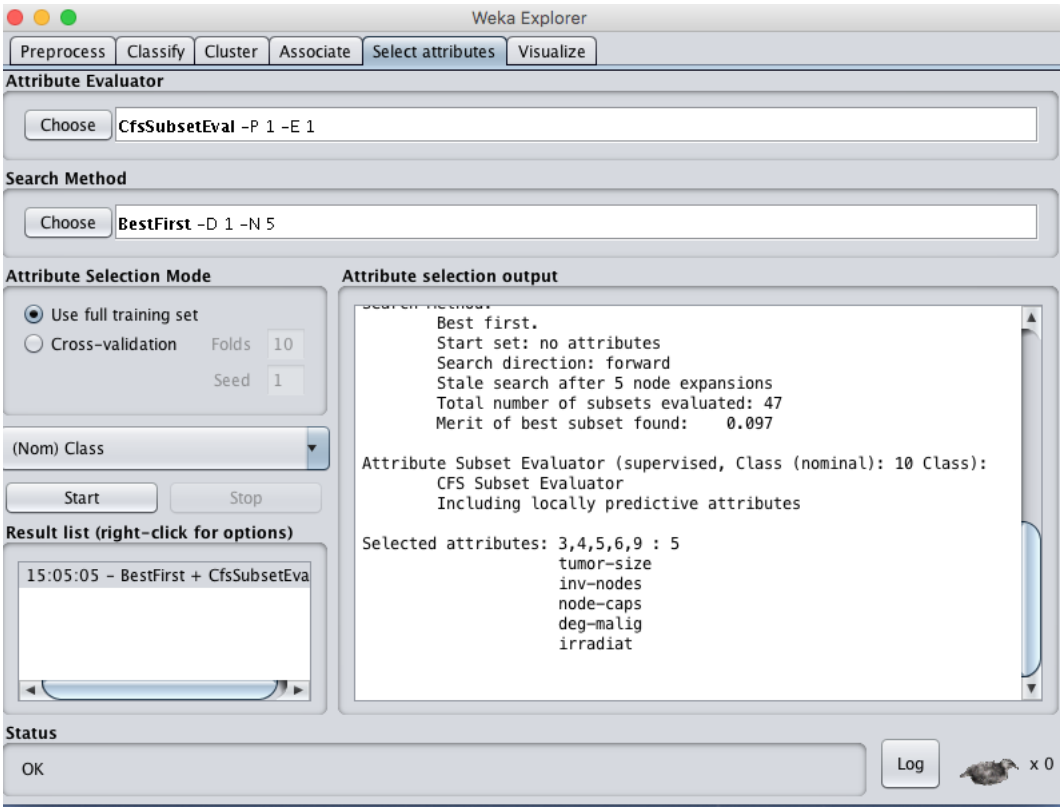


Figure 6.8: Screenshot of the *Weka Explorer Select Attributes* Tab.

The *Visualize* tab is for reviewing pairwise scatter plot matrix of each attribute plotted against every other attribute in the loaded dataset. It is useful to get an idea of the shape and relationship of attributes that may aid in data filtering, transformation and modeling.

- Increase the *PointSize* and the *Jitter* and click the *Update* button to set an improved plot of the categorical attributes of the loaded dataset.

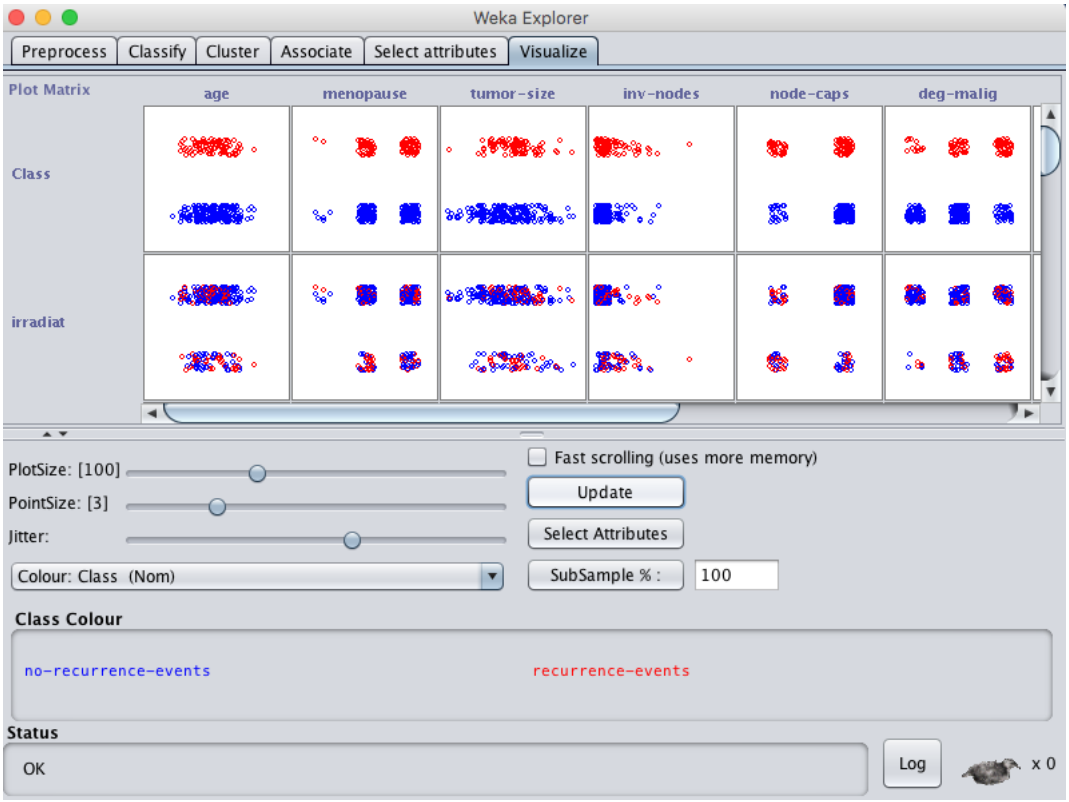


Figure 6.9: Screenshot of the *Weka Explorer* Visualize Tab.

6.3 Weka Experiment Environment

The *Weka Experiment Environment* is for designing controlled experiments, running them, then analyzing the results collected. It is the next step after using the *Weka Explorer*, where you can load up one or more views of your dataset and a suite of algorithms and design an experiment to find the combination that results in the best performance. The interface is split into 3 tabs.

The *Setup* tab is for designing an experiment. This includes the file where results are written, the test setup in terms of how algorithms are evaluated, the datasets to model and the algorithms to model them. The specifics of an experiment can be saved for later use and modification. In this section we will use the onset of diabetes binary classification problem. You will learn more about this dataset in Section 8.2.1.

- Click the *New* button to create a new experiment.
- Click the *Add New...* button in the *Datasets* pane and select the `data/diabetes.arff` dataset.
- Click the *Add New...* button in the *Algorithms* pane and click *OK* to add the *ZeroR* algorithm.

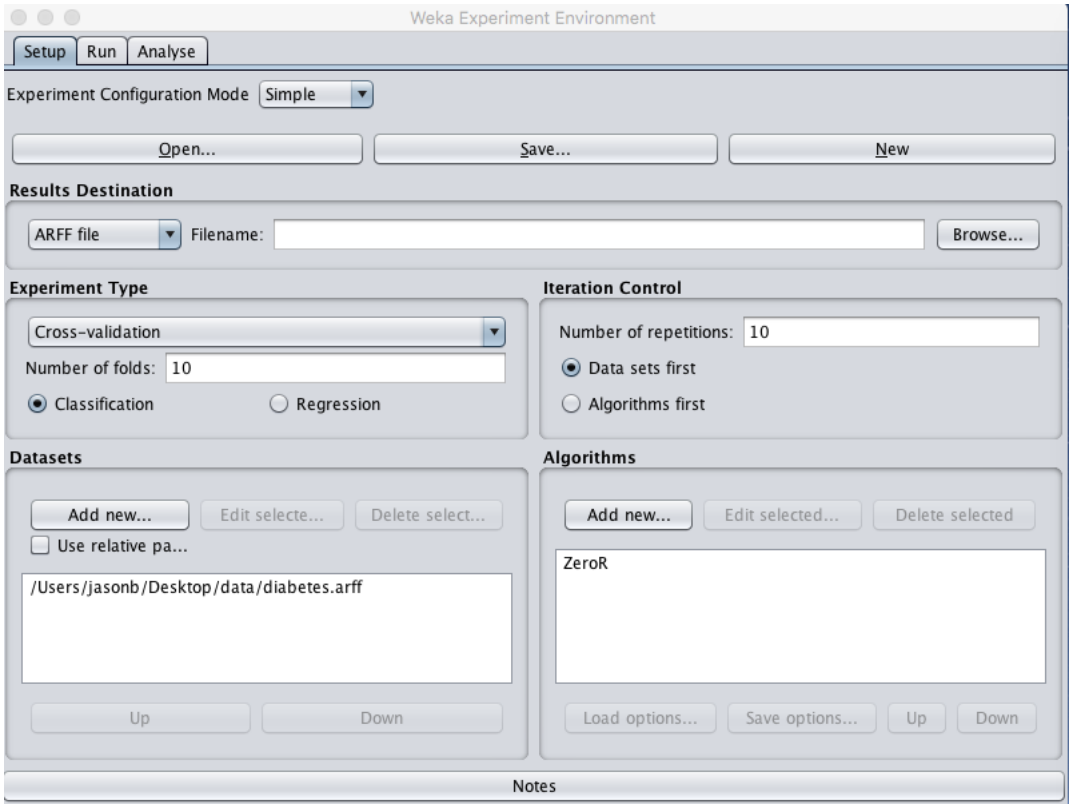


Figure 6.10: Screenshot of the *Weka Experiment Environment Setup* Tab.

The *Run* tab is for running your designed experiments. Experiments can be started and stopped. There is not a lot to it.

- Click the *Start* button to run the small experiment you designed.

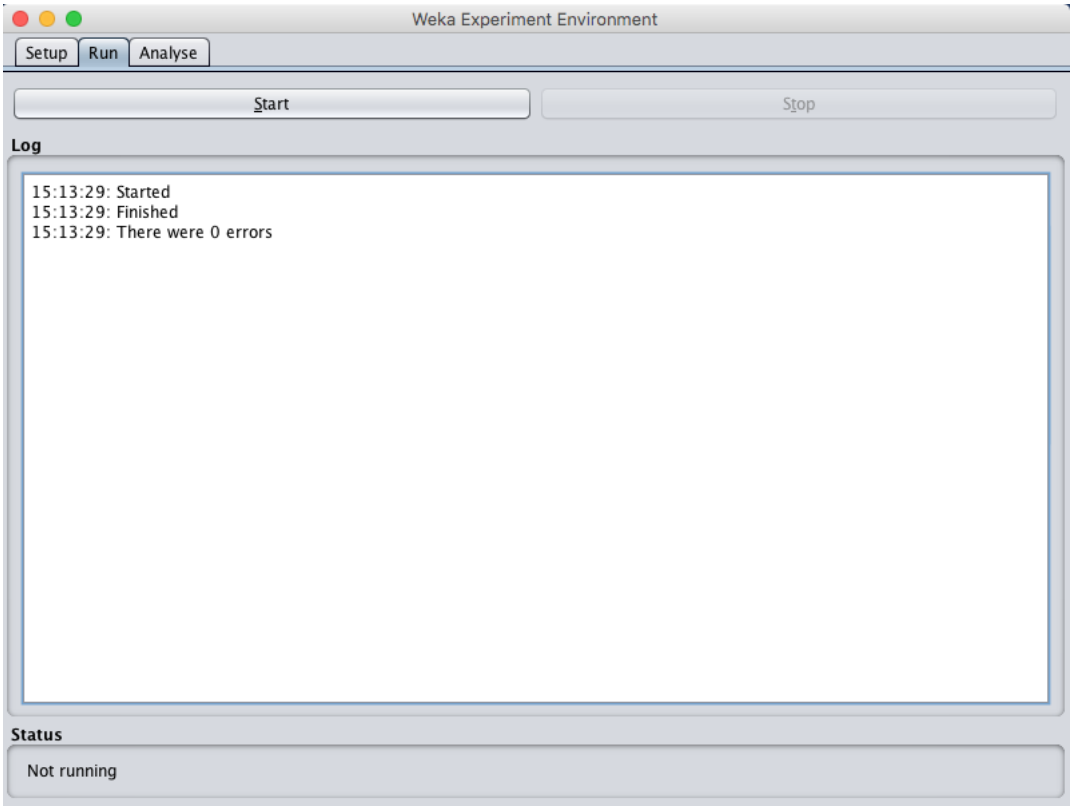


Figure 6.11: Screenshot of the *Weka Experiment Environment Run* Tab.

The *Analyse* tab is for analyzing the results collected from an experiment. Results can be loaded from a file, from the database or from an experiment just completed in the tool. A number of performance measures are collected from a given experiment which can be compared between algorithms using tools like statistical significance.

- Click the *Experiment* button the *Source* pane to load the results from the experiment you just ran.
- Click the *Perform Test* button to summary the classification accuracy results for the single algorithm in the experiment.

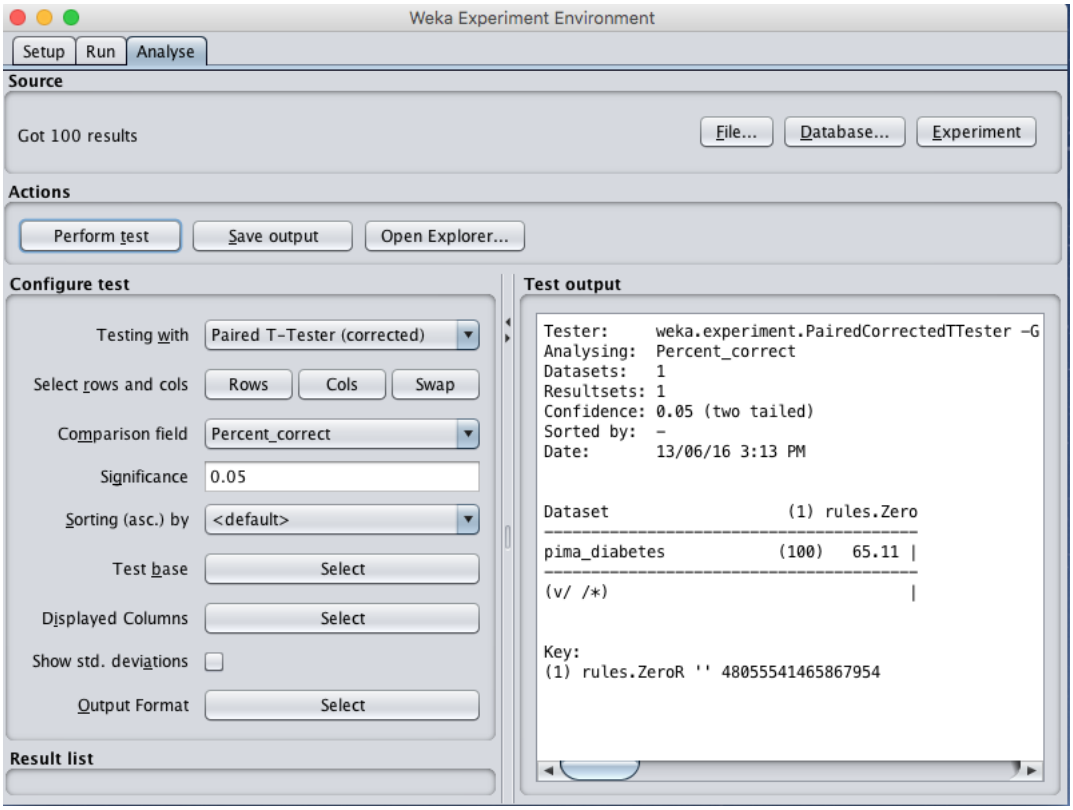


Figure 6.12: Screenshot of the *Weka Experiment Environment Analyse* Tab.

6.4 Weka KnowledgeFlow Environment

The *Weka KnowledgeFlow Environment* is a graphical workflow tool for designing a machine learning pipeline from data source to results summary, and much more. Once designed, the pipeline can be executed and evaluated within the tool.

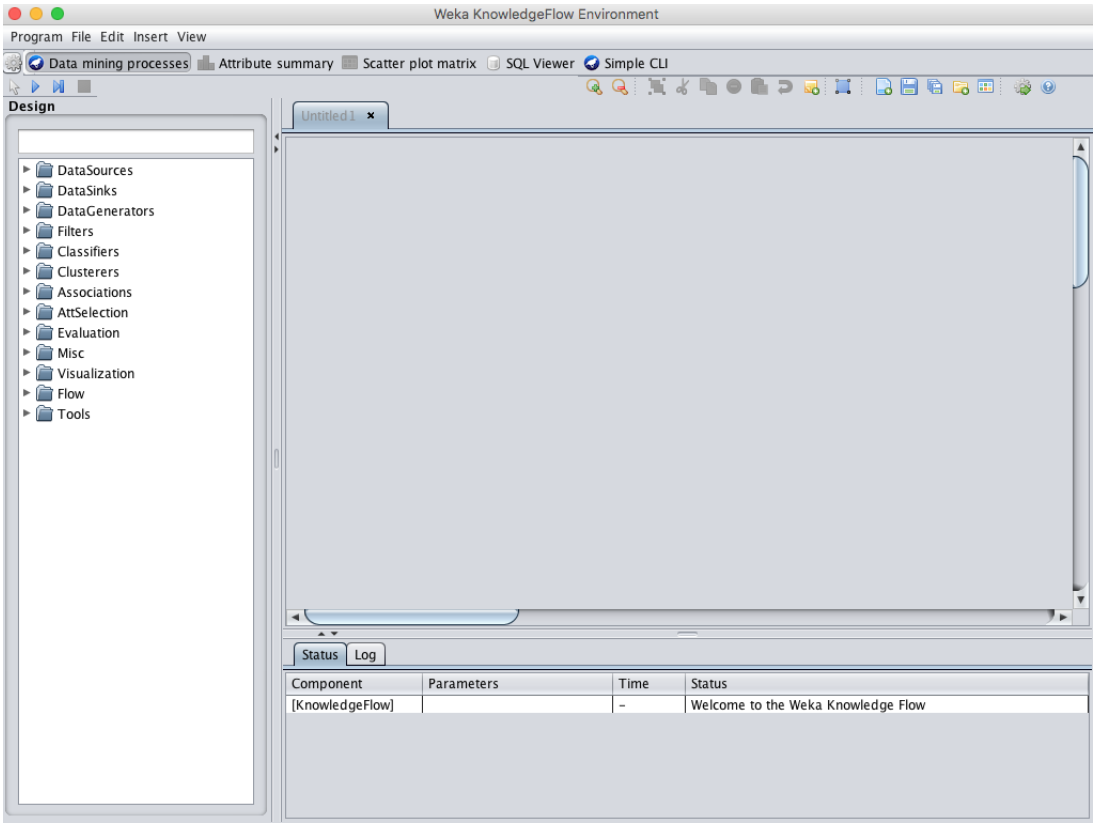


Figure 6.13: Screenshot of the Weka KnowledgeFlow Environment.

The *Weka KnowledgeFlow Environment* is a powerful tool that I do not recommend for beginners until after they have mastered use of the *Weka Explorer* and *Weka Experiment Environment*.

6.5 Weka Workbench

The *Weka Workbench* is an environment that combines all of the GUI interfaces into a single interface. It is useful if you find yourself jumping a lot between two or more different interfaces, such as between the *Weka Explorer* and the *Weka Experiment Environment*. This can happen if you try out a lot of what-if's in the Explorer and quickly take what you learn and put it into controlled experiments.

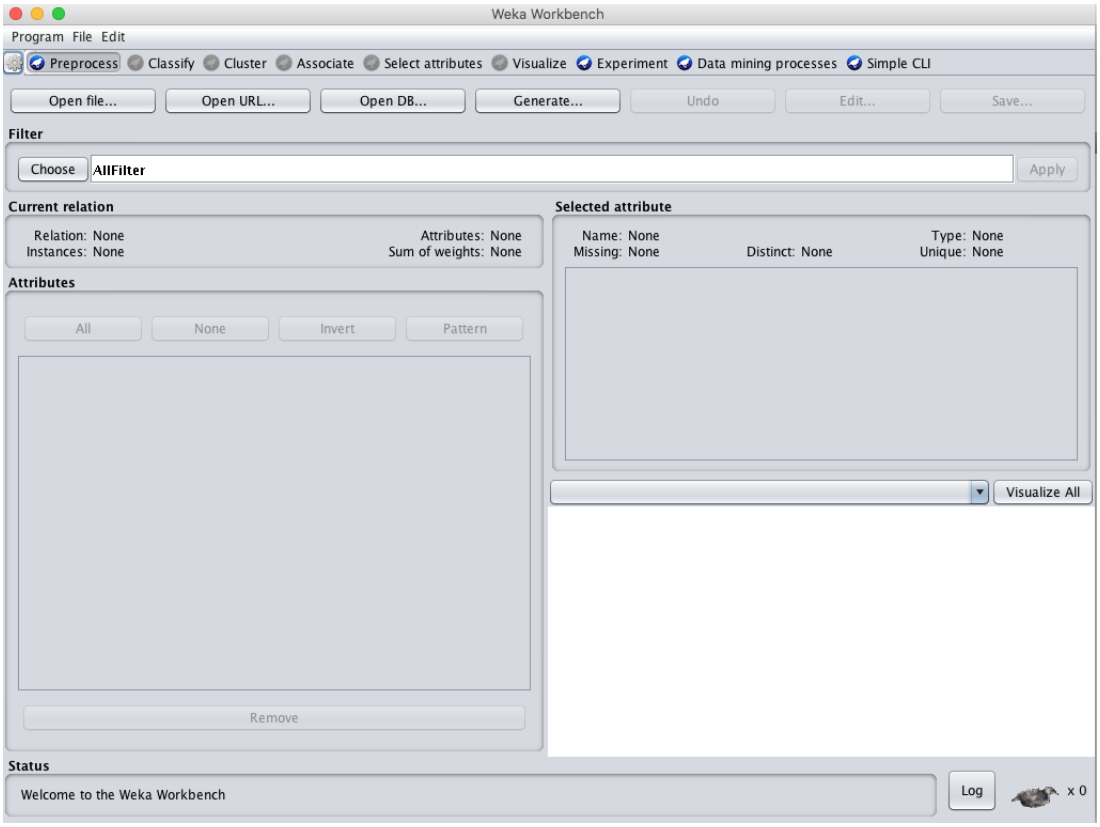


Figure 6.14: Screenshot of the Weka Workbench.

6.6 Weka SimpleCLI

Weka can be used from a simple Command Line Interface (CLI). This is powerful because you can write shell scripts to use the full API from command line calls with parameters, allowing you to build models, run experiments and make predictions without a graphical user interface. The *Weka SimpleCLI* provides an environment where you can quickly and easily experiment with the Weka command line interface commands.

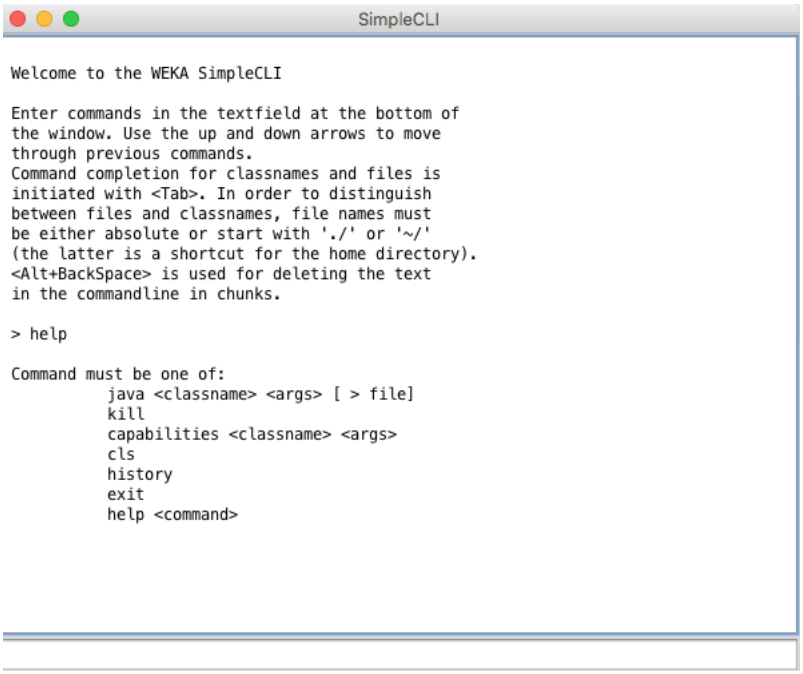


Figure 6.15: Screenshot of the Weka SimpleCLI.

Like the *Weka KnowledgeFlow Environment*, this is a powerful tool that I do not recommend for beginners until they have mastered use of the *Weka Explorer* and *Weka Experiment Environment*.

6.7 Weka Java API

Weka can also be used from the Java API. This is for Java programmers and can be useful when you want to incorporate learning or prediction into your own applications. This is an advanced feature that I do not recommend for beginners until they have mastered use of the *Weka Explorer* and *Weka Experiment Environment*.

6.8 Summary

In this lesson you discovered the Weka Machine Learning Workbench. You went on a tour of the key interfaces that you can use to explore and develop predictive machine learning models on your own problems. Specifically, you learned about:

- The *Weka Explorer* for data preparation, feature selection and evaluating algorithms.
- The *Weka Experiment Environment* for designing, running and analyzing the results from controlled experiments.
- The *Weka KnowledgeFlow Environment* for graphically designing and executing machine learning pipelines.
- The *Weka Workbench* that incorporates all of the Weka tools into a single convenient interface.

- The *Weka SimpleCLI* for using the Weka API from the command line.
- The Weka Java API that can be used to incorporate learning and prediction into your own applications.

6.8.1 Next

We are now familiar with the Weka interface. In the next lesson we will learn how to load machine learning data in CSV format into Weka.

Chapter 7

How To Load CSV Machine Learning Data

You must be able to load your data before you can start modeling it. In this lesson you will discover how you can load your CSV dataset in Weka. After reading this lesson, you will know:

- About the ARFF file format and how it is the default way to represent data in Weka.
- How to load a CSV file in the *Weka Explorer* and save it in ARFF format.
- How to load a CSV file in the *Weka ArffViewer* tool and save it in ARFF format.

Let's get started.

7.1 How to Talk About Data in Weka

Machine learning algorithms are primarily designed to work with arrays of numbers. This is called tabular or structured data because it is how data looks in a spreadsheet, comprised of rows and columns. Weka has a specific computer science centric vocabulary when describing data:

- **Instance**: A row of data is called an instance, as in an instance or observation from the problem domain.
- **Attribute**: A column of data is called a feature or attribute, as in feature of the observation.

Each attribute can have a different type, for example:

- **Real** for numeric values like 1.2.
- **Integer** for numeric values without a fractional part like 5.
- **Nominal** for categorical data like *dog* and *cat*.
- **String** for lists of words, like this sentence.

On classification problems, the output variable must be nominal. For regression problems, the output variable must be real.

7.2 Data in Weka

Weka prefers to load data in the ARFF format. ARFF is an acronym that stands for Attribute-Relation File Format. It is an extension of the CSV file format where a header is used that provides metadata about the data types in the columns. For example, the first few lines of the classic iris flowers dataset in CSV format looks as follows:

```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
```

Listing 7.1: CSV Data from the iris dataset.

The same file in ARFF format looks as follows:

```
@RELATION iris

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

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
```

Listing 7.2: ARFF version of the iris dataset.

You can see that directives start with the at symbol (`@`) and that there is one for the name of the dataset (e.g. `@RELATION iris`), there is a directive to define the name and datatype of each attribute (e.g. `@ATTRIBUTE sepallength REAL`) and there is a directive to indicate the start of the raw data (e.g. `@DATA`). Lines in an ARFF file that start with a percentage symbol (`%`) indicate a comment. Values in the raw data section that have a question mark symbol (`?`) indicate an unknown or missing value. The format supports numeric and categorical values as in the iris example above, but also supports dates and string values. Depending on your installation of Weka, you may or may not have some default datasets in your Weka installation directory under the `data/` subdirectory. These default datasets distributed with Weka are in the ARFF format and have the `.arff` file extension. You will learn more about these datasets in the next Lesson.

7.3 Load CSV Files in the *ARFF-Viewer*

Your data is not likely to be in ARFF format. In fact, it is much more likely to be in Comma Separated Value (CSV) format. This is a simple format where data is laid out in a table of rows and columns and a comma is used to separate the values on a row. Quotes may also be used to surround values, especially if the data contains strings of text with spaces. The CSV format is

easily exported from Microsoft Excel, so once you can get your data into Excel, you can easily convert it to CSV format.

Weka provides a handy tool to load CSV files and save them in ARFF. You only need to do this once with your dataset. Using the steps below you can convert your dataset from CSV format to ARFF format and use it with the Weka workbench. If you do not have a CSV file handy, you can use the iris flowers dataset. Download the file from the UCI Machine Learning repository[1] and save it to your current working directory as `iris.csv`. You will learn more about the iris dataset in Section 8.3.1.

- 1. Start the *Weka GUI Chooser*.

Figure 7.1: Screenshot of the *Weka GUI Chooser*.

- 2. Open the ARFF-Viewer by clicking *Tools* in the menu and select *ArffViewer*.
- 3. You will be presented with an empty *ARFF-Viewer* window.

[1] https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data

Figure 7.2: Screenshot of the Weka ARFF Viewer.

- 4. Open your CSV file in the *ARFF-Viewer* by clicking the *File* menu and select *Open*. Navigate to your current working directory. Change the *Files of Type:* filter to *CSV data files (*.csv)*. Select your file and click the *Open* button.

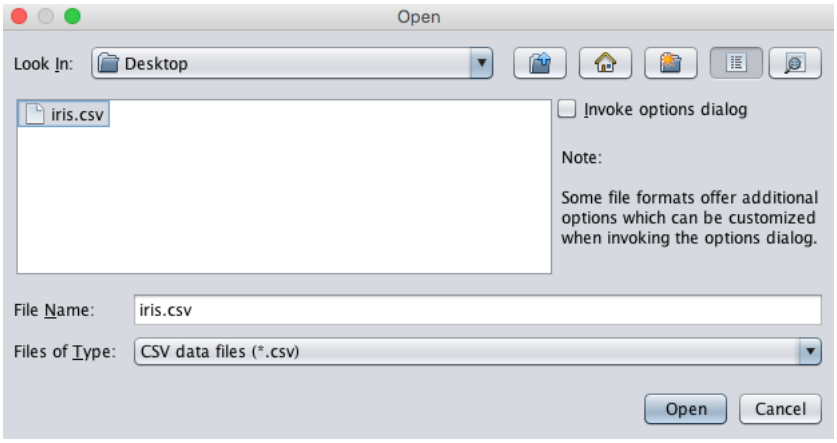


Figure 7.3: Load CSV In ARFF Viewer.

- 5. You should see a sample of your CSV file loaded into the *ARFF-Viewer*.
- 6. Save your dataset in ARFF format by clicking the *File* menu and selecting *Save as...*. Enter a filename with a `.arff` extension and click the *Save* button.

You can now load your saved `.arff` file directly into Weka. Note, the *ARFF-Viewer* provides options for modifying your dataset before saving. For example you can change values, change the name of attributes and change their data types. It is highly recommended that you specify the names of each attribute as this will help with analysis of your data later. Also, make sure that the data types of each attribute are correct.

7.4 Load CSV Files in the *Weka Explorer*

You can also load your CSV files directly in the *Weka Explorer* interface. This is handy if you are in a hurry and want to quickly test out an idea. This section shows you how you can load your CSV file in the *Weka Explorer* interface. You can use the iris dataset again, to practice if you do not have a CSV dataset to load.

- 1. Start the *Weka GUI Chooser*.
- 2. Launch the *Weka Explorer* by clicking the *Explorer* button.

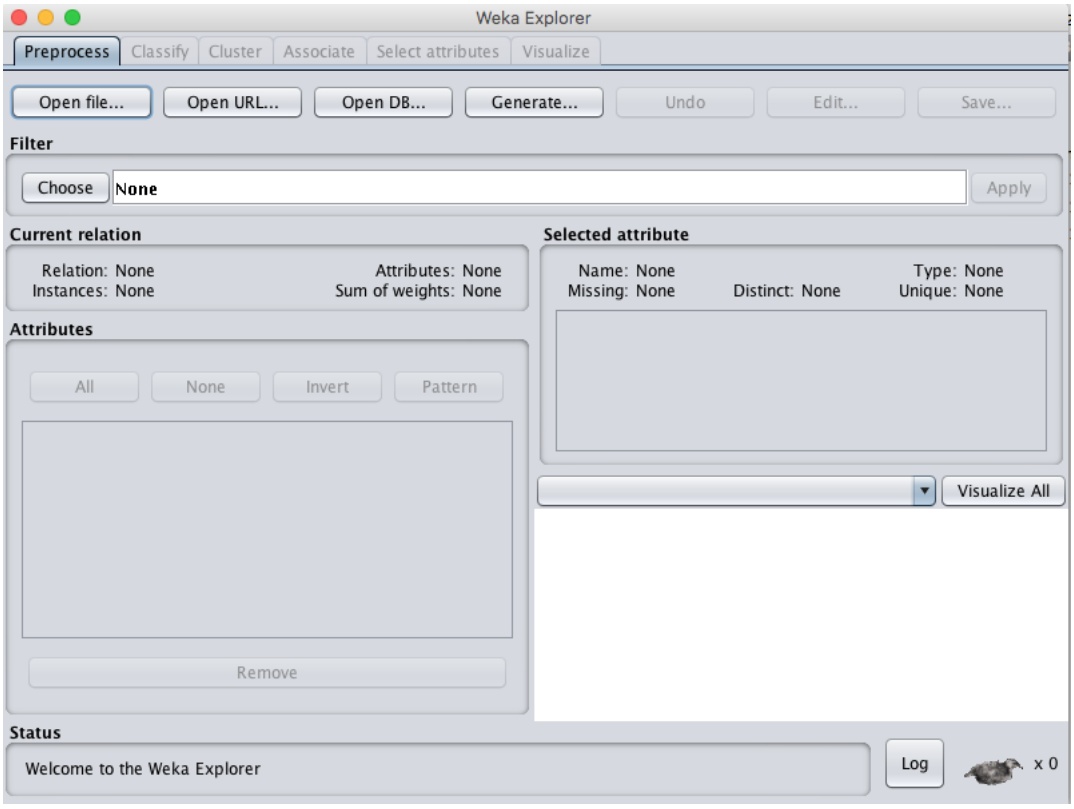


Figure 7.4: Screenshot of the *Weka Explorer*.

- 3. Click the *Open file...* button.
- 4. Navigate to your current working directory. Change the *Files of Type* to *CSV data files (*.csv)*. Select your file and click the *Open* button.

You can work with the data directly. You can also save your dataset in ARFF format by clicking the *Save* button and typing a filename.

7.5 Use Excel for Other File Formats

If you have data in another format, load it in Microsoft Excel first. It is common to get data in another format such as CSV using a different delimiter or fixed width fields. Excel has powerful tools for loading tabular data in a variety of formats. Use these tools and first load your data into Excel. Once you have loaded your data into Excel, you can export it into CSV format. You can then work with it in Weka, either directly or by first converting it to ARFF format.

7.6 Summary

In this lesson you discovered how to load your CSV data into Weka for machine learning. Specifically, you learned:

- About the ARFF file format and how Weka uses it to represent datasets for machine learning.
- How to load your CSV data using *ARFF-Viewer* and save it into ARFF format.
- How to load your CSV data directly in the *Weka Explorer* and use it for modeling.

7.6.1 Next

Weka is distributed with standard machine learning datasets that we can use as practice. In the next lesson you will learn about these standard datasets and specific examples that we can and should use as practice.

Chapter 8

How to Load Standard Machine Learning Datasets

It is a good idea to have small well understood datasets when getting started in machine learning and learning a new tool. The Weka machine learning workbench provides a directory of small well understood datasets in your Weka installation. In this lesson you will discover some of these small well understood datasets distributed with Weka, their details and where to learn more about them. We will focus on a handful of datasets of differing types. After reading this lesson you will know:

- Where the sample datasets are located or where to download them afresh if you need them.
- Specific standard datasets you can use to explore different aspects of classification and regression predictive models.
- Where to go for more information about specific datasets and state-of-the-art results.

Let's get started.

8.1 Standard Weka Datasets

An installation of the open source Weka machine learning workbench includes a `data/` directory full of standard machine learning problems.

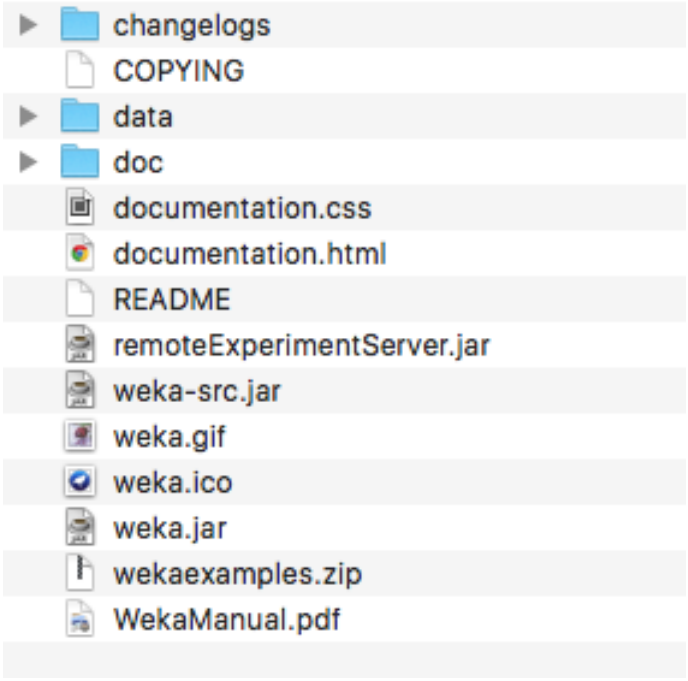


Figure 8.1: Weka Installation Directory.

This is very useful when you are getting started in machine learning or learning how to get started with the Weka platform. It provides standard machine learning datasets for common classification and regression problems, for example, below is a snapshot from this directory:

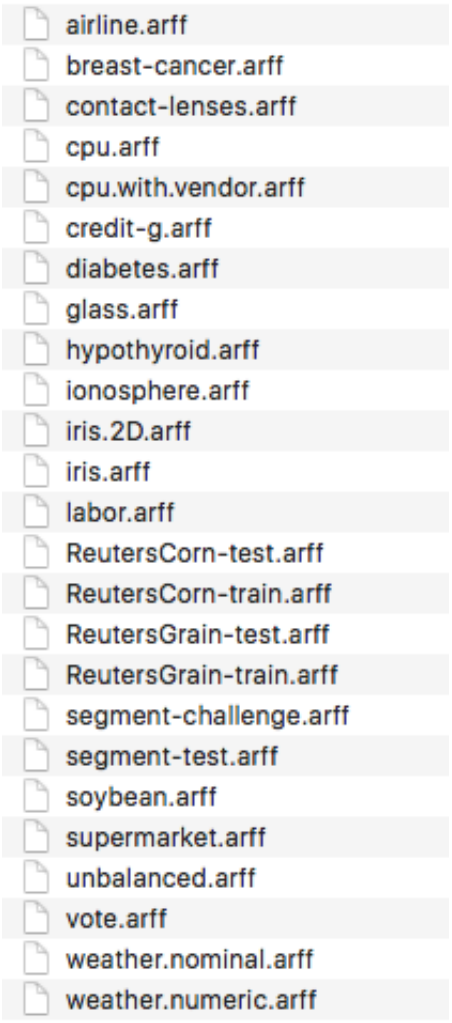


Figure 8.2: Provided Datasets in Weka Installation Directory.

All datasets are in the Weka native ARFF file format and can be loaded directly into Weka, meaning you can start developing practice models immediately. There are some special distributions of Weka that may not include the `data/` directory. If you have chosen to install one of these distributions, you can download the `.zip` distribution of Weka, unzip it and copy the `data/` directory to somewhere that you can access it easily from Weka. There are many datasets to play with in the `data/` directory, in the following sections I will point out a few that you can focus on for practicing and investigating predictive modeling problems.

8.2 Binary Classification Datasets

Binary classification is where the output variable to be predicted is nominal comprised of two classes. This is perhaps the most well studied type of predictive modeling problem and the type of problem that is good to start with. There are three standard binary classification problems in the `data/` directory that you can focus on:

8.2.1 Pima Indians Onset of Diabetes

Each instance represents medical details for one patient and the task is to predict whether the patient will have an onset of diabetes within the next five years. There are 8 numerical input variables all of which have varying scales.

- Dataset File: `data/diabetes.arff`
- More Info: https://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes
- Top results are in the order of 77% accuracy: http://www.is.umk.pl/projects/datasets.html#Diabetes

8.2.2 Breast Cancer

Each instance represents medical details of patients and samples of their tumor tissue and the task is to predict whether or not the patient has breast cancer. There are 9 input variables all of which a nominal.

- Dataset File: `data/breast-cancer.arff`
- More Info: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer
- Top results are in the order of 75% accuracy: http://www.is.umk.pl/projects/datasets.html#Ljubljana

8.2.3 Ionosphere

Each instance describes the properties of radar returns from the atmosphere and the task is to predict whether or not there is structure in the ionosphere. There are 34 numerical input variables of generally the same scale.

- Dataset File: `data/ionosphere.arff`
- More Info: https://archive.ics.uci.edu/ml/datasets/Ionosphere
- Top results are in the order of 98% accuracy: http://www.is.umk.pl/projects/datasets.html#Ionosphere

8.3 Multiclass Classification Datasets

There are many classification type problems, where the output variable has more than two classes. These are called multiclass classification problems. This is a good type of problem to look at after you have some confidence with binary classification. Three standard multiclass classification problems in the `data/` directory that you can focus on are:

8.3.1 Iris Flowers Classification

Each instance describes measurements of iris flowers and the task is to predict to which species of 3 iris flower the observation belongs. There are 4 numerical input variables with the same units and generally the same scale.

- Dataset File: `data/iris.arff`
- More Info: https://archive.ics.uci.edu/ml/datasets/Iris

8.3.2 Large Soybean Database

Each instance describes properties of a crop of soybeans and the task is to predict which of the 19 diseases the crop suffers. There are 35 nominal input variables.

- Dataset File: `data/soybean.arff`
- More Info: https://archive.ics.uci.edu/ml/datasets/Soybean+(Large)

8.3.3 Glass Identification

Each instance describes the chemical composition of samples of glass and the task is to predict the type or use of the class from one of 7 classes. There are 10 numeric attributes that describe the chemical properties of the glass and its refractive index.

- Dataset File: `data/glass.arff`
- More Info: https://archive.ics.uci.edu/ml/datasets/Glass+Identification

8.4 Regression Datasets

Regression problems are those where you must predict a real valued output. The selection of regression problems in the `data/` directory is small. Regression is an important class of predictive modeling problem. As such I recommend downloading the free add-on pack of regression problems collected from the UCI Machine Learning Repository. It is available from the datasets page on the Weka webpage[1] and is the first in the list called:

- A jar file containing 37 regression problems, obtained from various sources (`datasets-numeric.jar`)

It is a `.jar` file which is a type of compressed Java archive. You should be able to unzip it with most modern unzip programs. If you have Java installed (which you very likely do to use Weka), you can also unzip the `.jar` file manually on the command line using the following command in the directory where the jar was downloaded:

```
jar -xvf datasets-numeric.jar
```

Listing 8.1: Uncompress numerical datasets for Weka.

Unzipping the file will create a new directory called `numeric` that contains 37 regression datasets in ARFF native Weka format. Three regression datasets in the `numeric/` directory that you can focus on are:

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

8.4.1 Longley Economic Dataset

Each instance describes the gross economic properties of a nation for a given year and the task is to predict the number of people employed as an integer. There are 6 numeric input variables of varying scales.

- Dataset File: `numeric/longley.arff`

8.4.2 Boston House Price Dataset

Each instance describes the properties of a Boston suburb and the task is to predict the house prices in thousands of dollars. There are 13 numerical input variables with varying scales describing the properties of suburbs.

- Dataset File: `numeric/housing.arff`
- More Info: https://archive.ics.uci.edu/ml/datasets/Housing

8.4.3 Sleep in Mammals Dataset

Each instance describes the properties of different mammals and the task is to predict the number of hours of total sleep they require on average. There are 7 numeric input variables of different scales and measures.

- Dataset File: `numeric/sleep.arff`

8.5 Summary

In this lesson you discovered the standard machine learning datasets distributed with the Weka machine learning platform. Specifically, you learned:

- Three popular binary classification problems you can use for practice: diabetes, breast-cancer and ionosphere.
- Three popular multiclass classification problems you can use for practice: iris, soybean and glass.
- Three popular regression problems you can use for practice: longley, housing and sleep.

8.5.1 Next

Now that we know how to load data, we can start working through problems. In the next lesson you will discover how you can learn more about a load dataset by reviewing descriptive statistics and data visualizations.

Chapter 9

How to Better Understand Your Machine Learning Data

It is important to take your time to learn about your data when starting on a new machine learning problem. There are key things that you can look at to very quickly learn more about your dataset, such as descriptive statistics and data visualizations. In this lesson you will discover how you can learn more about your data in the Weka machine learning workbench by reviewing descriptive statistics and visualizations of your data. After reading this lesson you will know about:

- The distribution of attributes from reviewing statistical summaries.
- The distribution of attributes from reviewing univariate plots.
- The relationship between attributes from reviewing multivariate plots.

Let's get started

9.1 Descriptive Statistics

The *Weka Explorer* will automatically calculate descriptives statistics for numerical attributes.

1. Open the *Weka GUI Chooser*.
2. Click *Explorer* to open the *Weka Explorer*.
3. Load the Pima Indians datasets from `data/diabetes.arff`.

The Pima Indians dataset contains numeric input variables that we can use to demonstrate the calculation of descriptive statistics. You can learn more about this dataset in Section 8.2.1. Firstly, note that the dataset summary in the *Current Relation* section. This pane summarizes the following details about the loaded datasets:

- Dataset name (relation).
- The number of rows (instances).

- The number of columns (attributes).

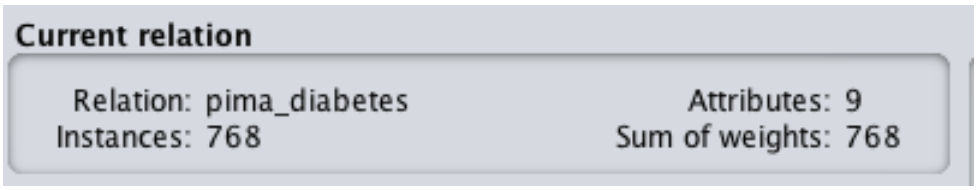


Figure 9.1: Weka Summary of Dataset.

- Click on the first attribute in the dataset in the *Attributes* pane.

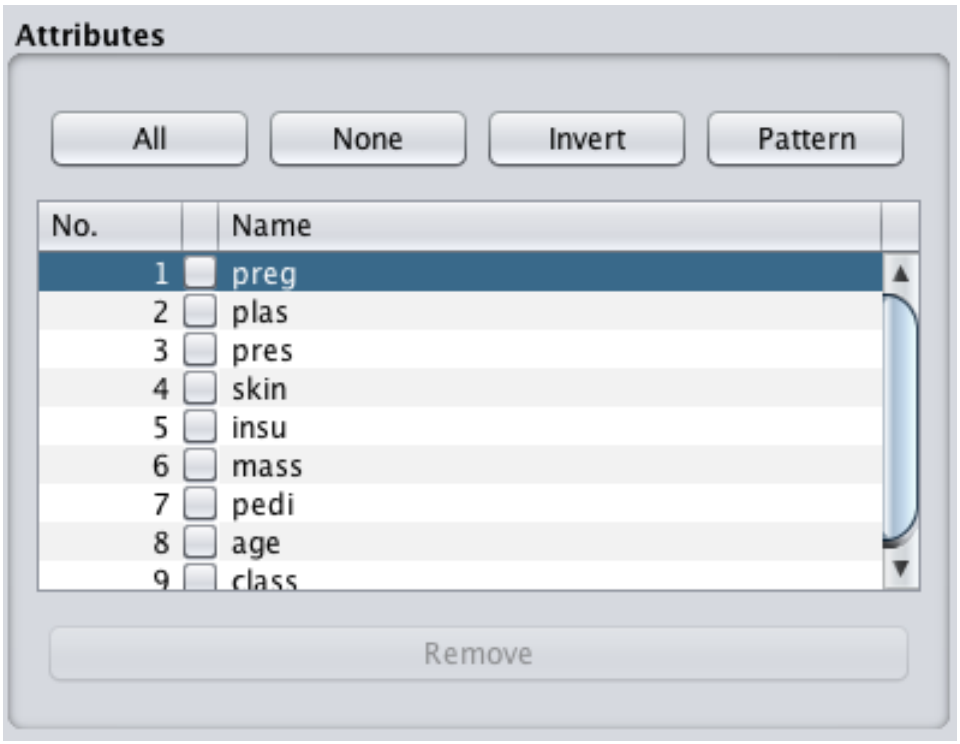


Figure 9.2: Weka List of Attributes.

Take note of the details in the *Selected attribute* pane. It lists a lot of information about the selected attribute, such as:

- The name of the attribute.
- The number of missing values and the ratio of missing values across the whole dataset.
- The number of distinct values.
- The data type.