**Weka**

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. Weka is open source software issued under the GNU General Public License.  
<https://www.cs.waikato.ac.nz/ml/weka/index.html>

All of Weka's techniques are predicated on the assumption that the data is available as one flat file or relation, where each data point is described by a fixed number of attributes (normally, numeric or nominal attributes, but some other attribute types are also supported).

**Attribute Relation File Format (ARFF)** is the default file type for data analysis in Weka but data can also be imported from various other formats, such as CSV and JSON. Data can also be read from a database using ODBC connectivity. ARFF has two parts:|  
 - the header section, which defines the relation (data set) name, attribute name and the type  
 - the data section, which lists the data instances

An ARFF file requires the declaration of the relation, attribute and data.

Weka's main user interface is the **Explorer**, but essentially the same functionality can be accessed through the component-based **Knowledge Flow** interface and from the command line. There is also the **Experimenter**, which allows the systematic comparison of the predictive performance of Weka's machine learning algorithms on a collection of datasets.  
<https://en.wikipedia.org/wiki/Weka_(machine_learning)#User_interfaces>

The Weka explorer contains various tabs:  
 - Preprocess: This allows us to choose the data file.  
 - Classify: This allows us to apply and experiment with different algorithms on preprocessed data files.  
 - Cluster: This allows us to apply different clustering tools, which identify clusters within the data file.  
 - Association: This allows us to apply association rules, which identify the association within the data.  
 - Select attributes: These allow us to see the changes on the inclusion and exclusion of attributes from the experiment.  
 - Visualize: This allows us to see the possible visualisation produced on the data set in a 2D format, in scatter plot and bar graph output.

Data preprocessing is a must. There are three ways to inject the data for preprocessing:

- Open File – enables the user to select the file from the local machine

- Open URL – enables the user to select the data file from different locations

- pen Database – enables users to retrieve a data file from a database source

**Classification**: To predict nominal or numeric quantities, we have classifiers in Weka. Available learning schemes are decision-trees and lists, support vector machines, instance-based classifiers, logistic regression and Bayes’ nets. Once the data has been loaded, all the tabs are enabled. Based on the requirements and by trial and error, we can find out the most suitable algorithm to produce an easily understandable representation of data. Before running any classification algorithm, we need to set test options.

**Clustering**: The cluster tab enables the user to identify similarities or groups of occurrences within the data set. Clustering can provide data for the user to analyse. The training set, percentage split, supplied test set and classes are used for clustering, for which the user can ignore some attributes from the data set, based on the requirements. Available clustering schemes in Weka are k-Means, EM, Cobweb, X-means and FarthestFirst.

**Association**: The only available scheme for association in Weka is the Apriori algorithm. It identifies statistical dependencies between clusters of attributes, and only works with discrete data. The Apriori algorithm computes all the rules having minimum support and exceeding a given confidence level.

**Attribute selection**: Attribute selection crawls through all possible combinations of attributes in the data to decide which of these will best fit the desired calculation—which subset of attributes works best for prediction.

**Visualisation**: The user can see the final piece of the puzzle, derived throughout the process. It allows users to visualise a 2D representation of data, and is used to determine the difficulty of the learning problem. We can visualise single attributes (1D) and pairs of attributes (2D), and rotate 3D visualisations in Weka. It has the Jitter option to deal with nominal attributes and to detect ‘hidden’ data points.

Using the graphical tools, like the Explorer, or just the command-line is in most cases suﬃcient for the normal user. But WEKA’s clearly deﬁned API makes it very easy to “embed” it in another projects.  
 http://weka.sourceforge.net/doc.dev/

**Useful links**

**Documentation:** <https://vorboss.dl.sourceforge.net/project/weka/documentation/3.9.x/WekaManual-3-9-2.pdf>  
 <http://weka.wikispaces.com/> <https://wiki.pentaho.com/display/DATAMINING/Pentaho+Data+Mining+Community+Documentation>  
 <http://markahall.blogspot.ro/>

**API tutorials and code examples**

<https://weka.wikispaces.com/Use+WEKA+in+your+Java+code#Examples>

<https://www.youtube.com/watch?v=6o19TPn181g&list=PLea0WJq13cnBVfsPVNyRAus2NK-KhCuzJ>

<https://www.youtube.com/watch?v=QdoBHuXQc-0> (decison tree using Weka)

<https://www.youtube.com/watch?v=COpXNK0O8As> (regression and clustering)