

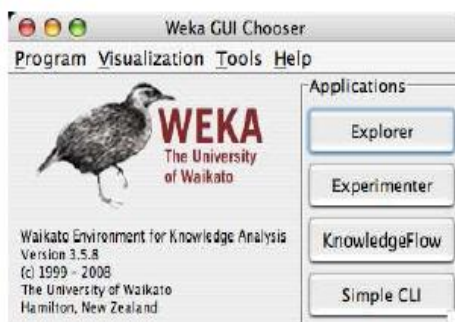
LAUNCHING WEKA:

WEKA → Waikato Environment for Knowledge Analysis

The WEKA GUI Chooser provides a starting point for launching WEKA'S main GUI applications and supporting tools. The GUI Chooser consists of four buttons—one for each of the four major Weka applications—and four menus.

The buttons can be used to start the following applications:

1. **Explorer:** An environment for exploring data with WEKA.
2. **Experimenter:** An environment for performing experiments and conducting statistical tests between learning schemes.
3. **Knowledge Flow:** It supports essentially the same functions as the explorer but with a drag and drop interface. One advantage is that it supports incremental learning.
4. **Simple CLI:** Provides a simple command-line interface that allows direct execution of WEKA commands for operating systems that do not provide their own command line interface.



The menu consists of four sections like Program, Tools, Visualization, and Help.

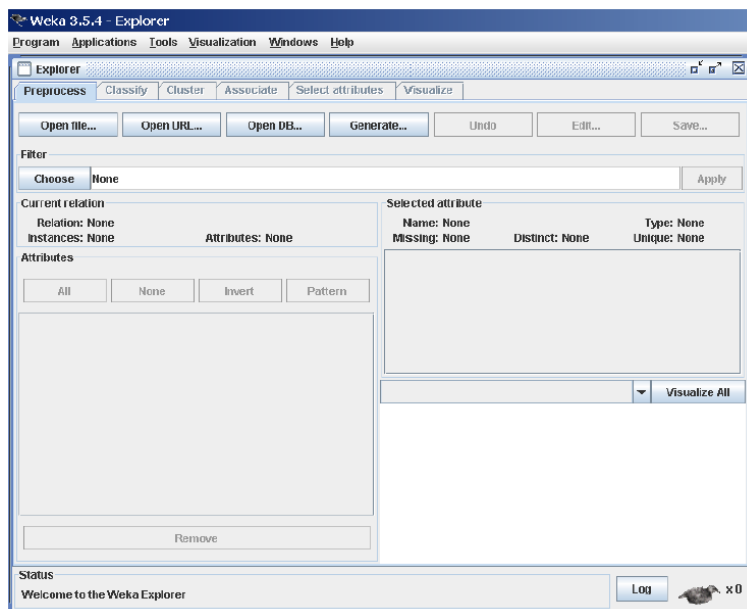
EXPLORER:

It is a user interface which contains a group of tabs just below the title bar. The tabs are as follows:

1. Preprocess
2. Classify
3. Cluster
4. Associate
5. Select Attributes
6. Visualize

The bottom of the window contains status box, log and WEKA bird.

1. PREPROCESSING:



LOADING DATA

The first four buttons at the top of the preprocess section enable you to load Data into WEKA:

1. **Open file:** It shows a dialog box allowing you to browse for the data file on the local file system.
2. **Open URL:** Asks for a Uniform Resource Locator address for where the data is stored.
3. **Open DB:** Reads data from a database.
4. **Generate:** It is used to generate artificial data from a variety of Data Generators.

Using the Open file button we can read files in a variety of formats like WEKA's ARFF format, CSV format. Typically ARFF files have .arff extension and CSV files .csv extension.

THE CURRENT RELATION

The Current relation box contains the currently loaded data i.e. interpreted as a single relational table in database terminology, which has three entries:

1. **Relation:** It provides the name of the relation in the file from which it was loaded. Filters are used modify the name of a relation.
2. **Instances:** The number of instances (data points/records) in the data.
3. **Attributes:** The number of attributes (features) in the data.

ATTRIBUTES

It is located below the current relation box which contains four buttons, they are:

- 1) **All** is used to tick all boxes
- 2) **None** is used to clear all boxes
- 3) **Invert** is used make ticked boxes unticked.
- 4) **Pattern** is used to select attributes by representing an expression. E.g. a.* is used to select all the attributes that begins with a.

SELECTED ATTRIBUTE:

It is located beside the current relation box which contains the following:

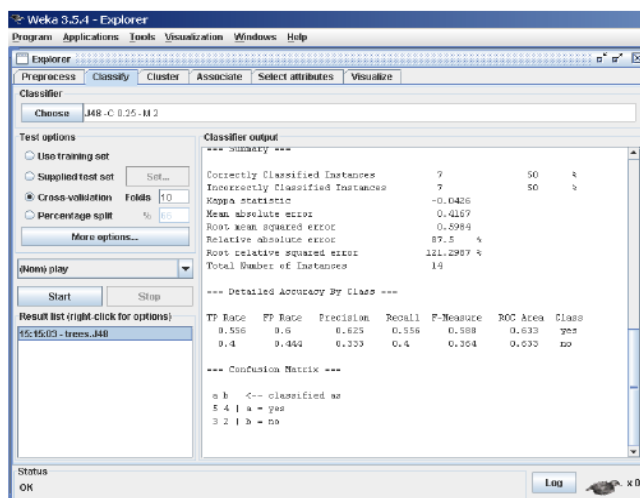
1. **Name:** It specifies the name of the attribute i.e. same as in the attribute list.
2. **Type:** It specifies the type of attribute, most commonly Nominal or Numeric.

3. **Missing:** It provides a numeric value of instances in the data for which an attribute is missing.
4. **Distinct:** It provides the number of different values that the data contains for an attribute.
5. **Unique:** it provides the number of instances in the data having a value for an attribute that no other instances have.

FILTERS

By clicking the **Choose** button at the left of the Filter box, it is possible to select one of the filters in WEKA. Once a filter has been selected, its name and options are shown in the field next to the Choose button, by clicking on this box with the left mouse button it shows a GenericObjectEditor dialog box which is used to configure the filter.

2. CLASSIFICATION



Classification has a text box which gives the name of currently selected classifier, and its options. By clicking it with the left mouse button it shows a GenericObjectEditor dialog box, which is same as for filters i.e. used to configure the current classifier options.

TEST OPTIONS

The result of applying the chosen classifier will be tested according to the options that are set by clicking in the Test options box. There are four test modes:

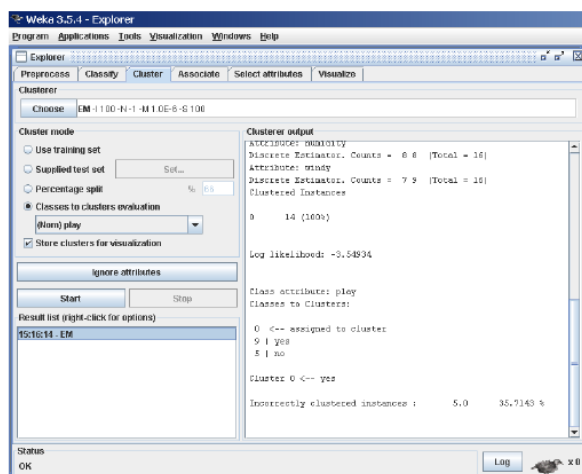
1. Use training set.
2. Supplied test set.
3. Cross-validation.
4. Percentage split.

Once the classifier, test options and class have all been set, the learning process is started by clicking on the **Start** button. We can stop the training process at any time by clicking on the **Stop** button.

The **Classifier output** area to the right of the display is filled with text describing the results of training and testing.

After training several classifiers, the **Result List** will contain several entries using which we can move over various results that have been generated. By pressing Delete we can remove a selected entry from the results.

3. CLUSTERING



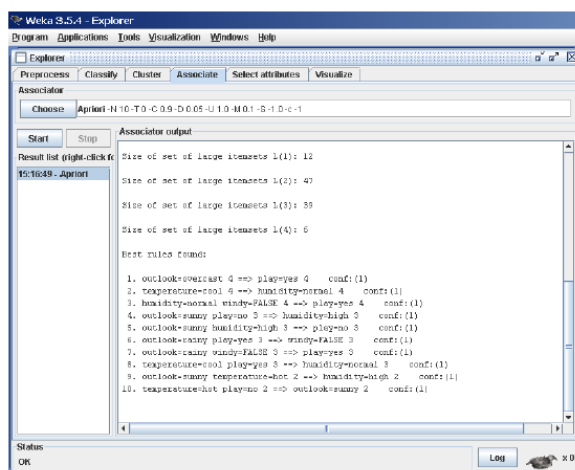
By clicking the text box beside the choose button in the **Clusterer box**, it shows a dialog box used to choose a new clustering scheme.

The **Cluster mode** box is used to choose what to cluster and how to evaluate the results. The first three options in it are same as in classification like **Use training set**, **Supplied test set** and **Percentage split**. The fourth option is **classes to clusters evaluation**

An additional option in the Cluster mode box is the **Store clusters for visualization** which finds whether or not it will be possible to visualize the clusters once training is complete.

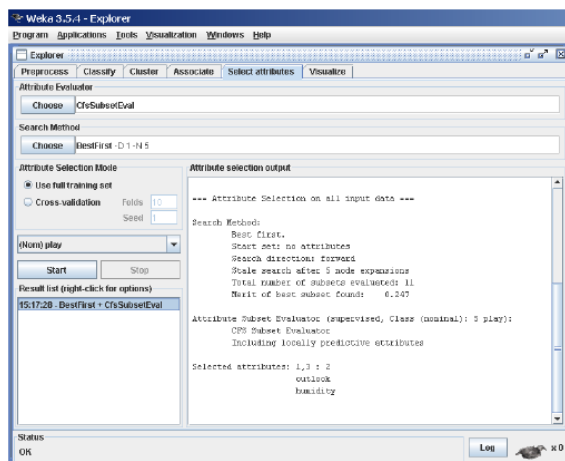
Ignore Attributes: when clustering, some attributes in the data should be ignored. It shows a small window that allows you to select which attributes are ignored.

4. ASSOCIATING



It contains schemes for learning association rules, and the learners are chosen and configured in the same way as the clusterer, filters, and classifiers in the other panels.

5. SELECTING ATTRIBUTES

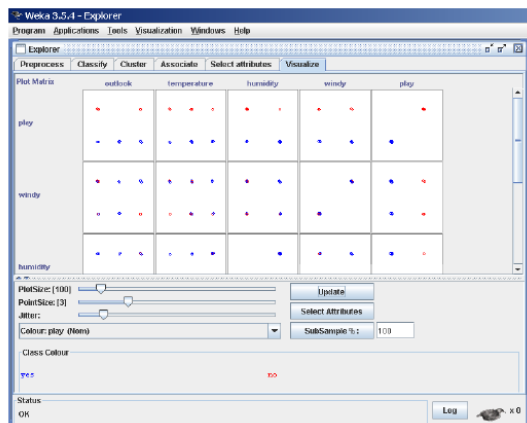


Attribute selection involves searching through all possible combinations of attributes in the data to find which subset of attributes works best for prediction. To do this, two objects must be set up: an attribute evaluator and a search method. The evaluator determines what method is used to assign a worth to each subset of attributes. The search method determines what style of search is performed.

The **Attribute Selection Mode** box has two options:

1. **Use full training set:** The worth of the attribute subset is determined using the full set of training data.
2. **Cross-validation:** The worth of the attribute subset is determined by a process of cross-validation. The Fold and Seed fields set the number of folds to use and the random seed used when shuffling the data.

6. VISUALIZING



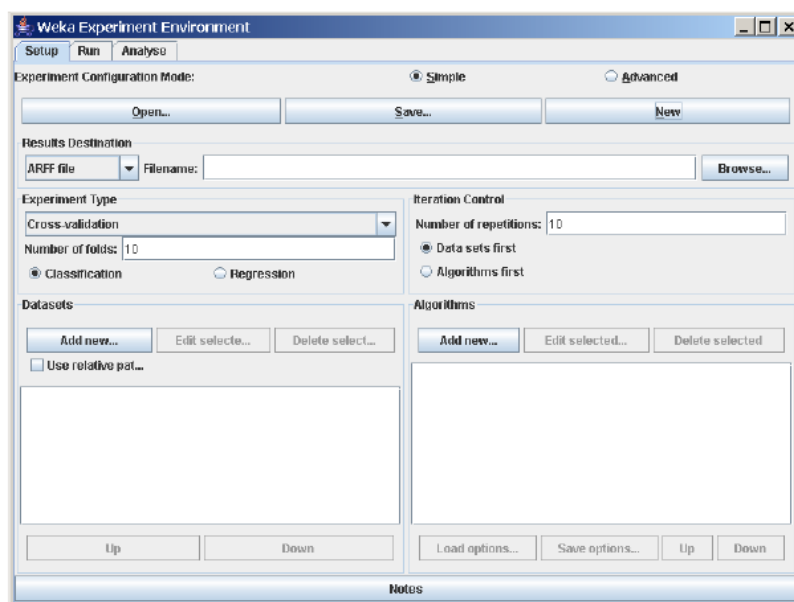
WEKA's visualization section allows you to visualize 2D plots of the current relation.

EXPERIMENTER: The Weka Experiment Environment enables the user to create, run, modify, and analyses experiments in a more convenient manner. It can also be run from the command line using the Simple CLI.

New Experiment:

After clicking on **new** default parameters for an Experiment are defined.

We can choose the experiment in two different modes 1) Simple and 2) Advanced



Result Destination:

By default, an ARFF file is the destination for the results output. But we can also choose CSV file as the destination for output file. The advantage of ARFF or CSV files is that they can be created without any additional classes. The drawback is the lack of ability to resume the interrupted experiment.

Experiment type:

The user can choose between the following three different types:

1. Cross-validation: it is a default type and it performs stratified cross-validation with the given number of folds.
2. Train/Test Percentage Split: it splits a dataset according to the given percentage into a train and a test file after the order of the data has been randomized and stratified.
3. Train/Test Percentage Split: As it is impossible to specify an explicit train/test files pair, one can abuse this type to un-merge previously merged train and test file into the two original files.

Additionally, one can choose between Classification and Regression, depending on the datasets and classifiers one uses.

Data Sets:

One can add dataset files either with an absolute path or with a relative path.

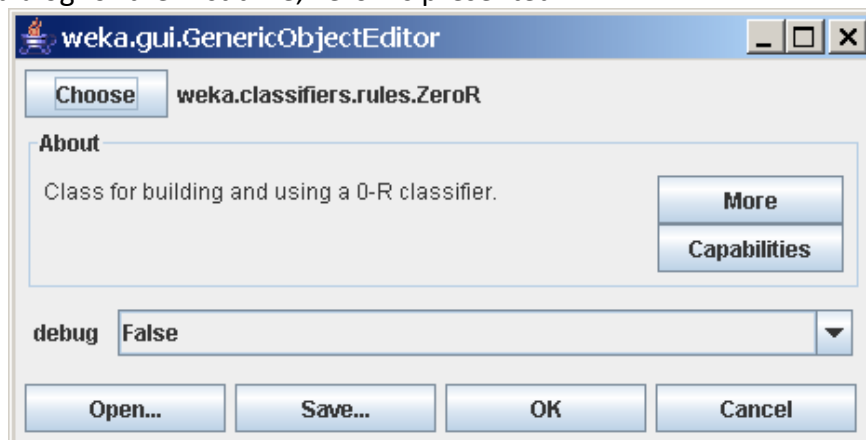
Iteration control:

1. *Number of repetitions:* In order to get statistically meaningful results, the default number of iterations is 10.

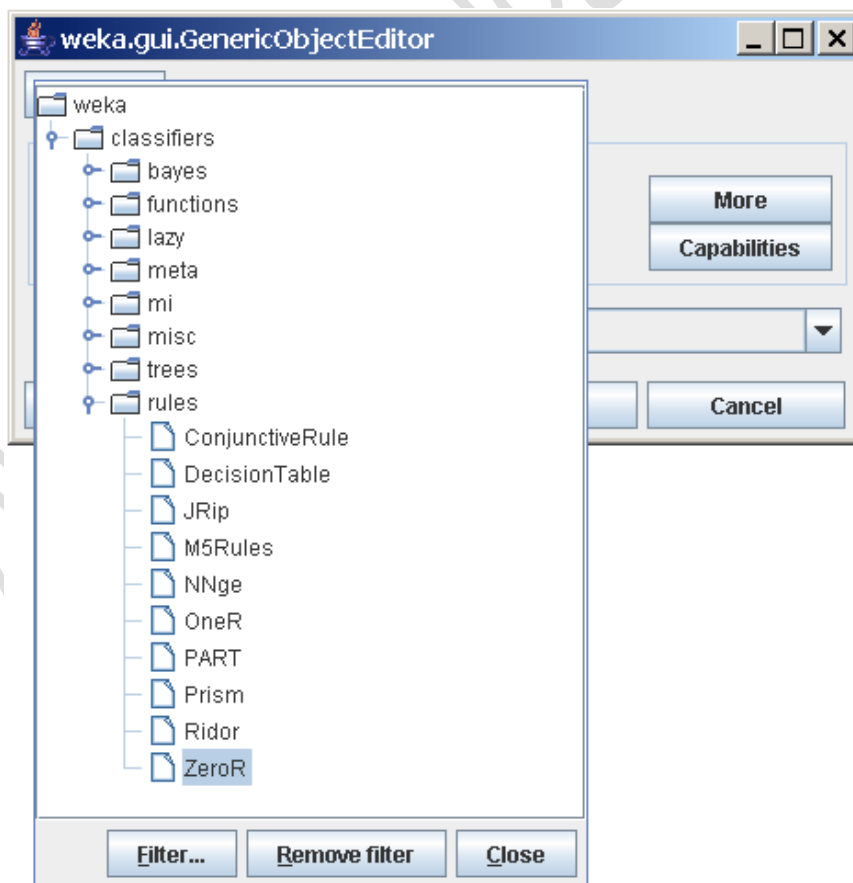
2. *Data sets first/Algorithms first*: As soon as one has more than one dataset and algorithm, it can be useful to switch from datasets being iterated over first to algorithms.

Algorithms: New algorithms can be added via the “Add New” button.

Opening this dialog for the first time, ZeroR is presented.



By clicking on the Choose button one can choose another classifier which is as shown in the below diagram:

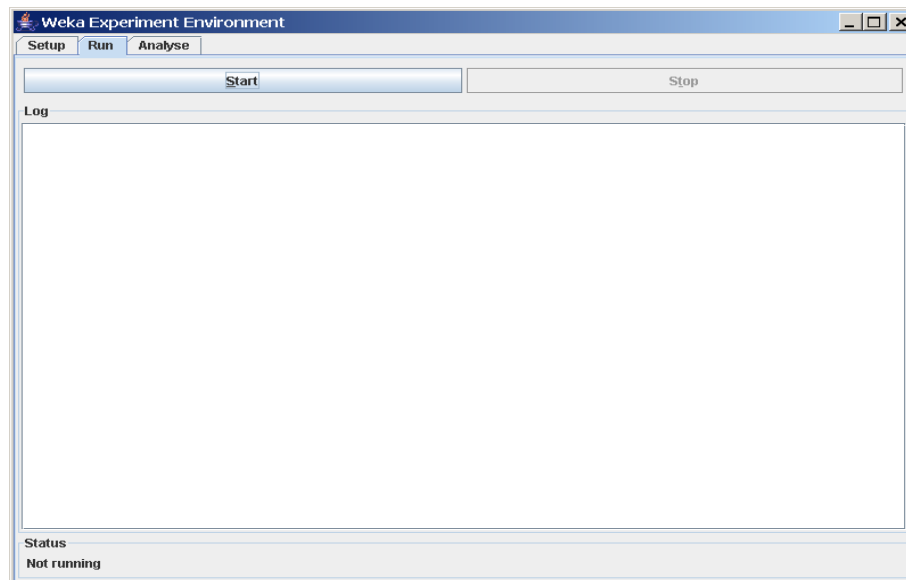


The “Filter...” button enables us to highlight classifiers that can handle certain attributes and class types. With “Remove Filter” button one can clear the classifiers that are highlighted earlier.

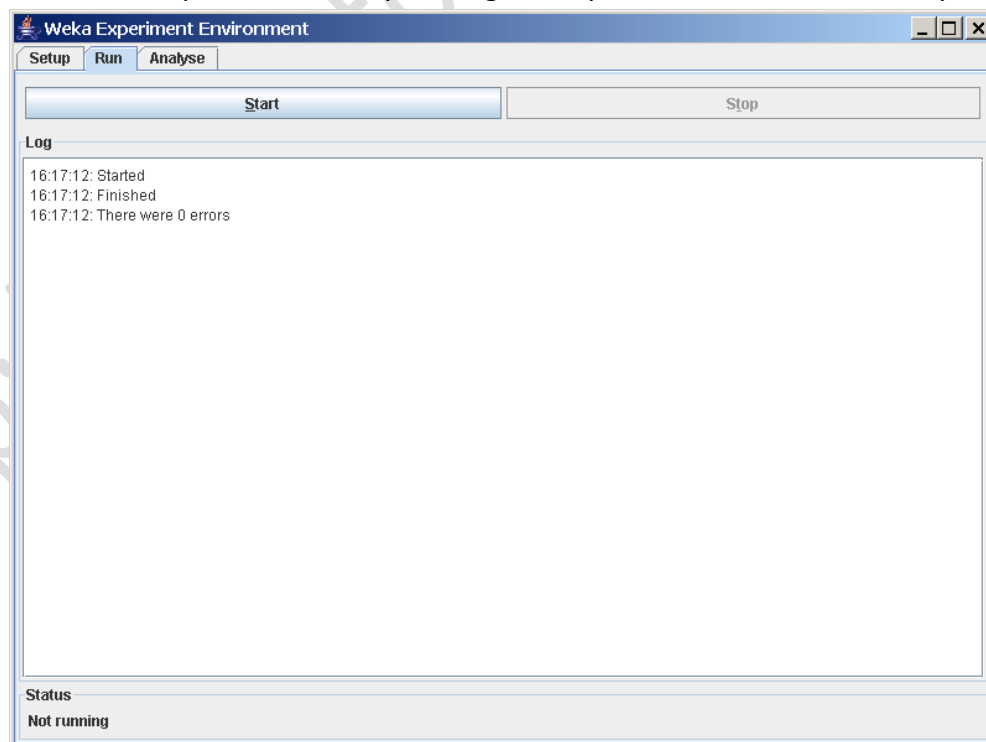
With the Load options... and Save options... buttons one can load and save the setup of a selected classifier from and to XML.

Running an Experiment:

To run the current experiment, click the Run tab at the top of the Experiment Environment window. The current experiment performs 10 runs of 10-fold stratified cross-validation.



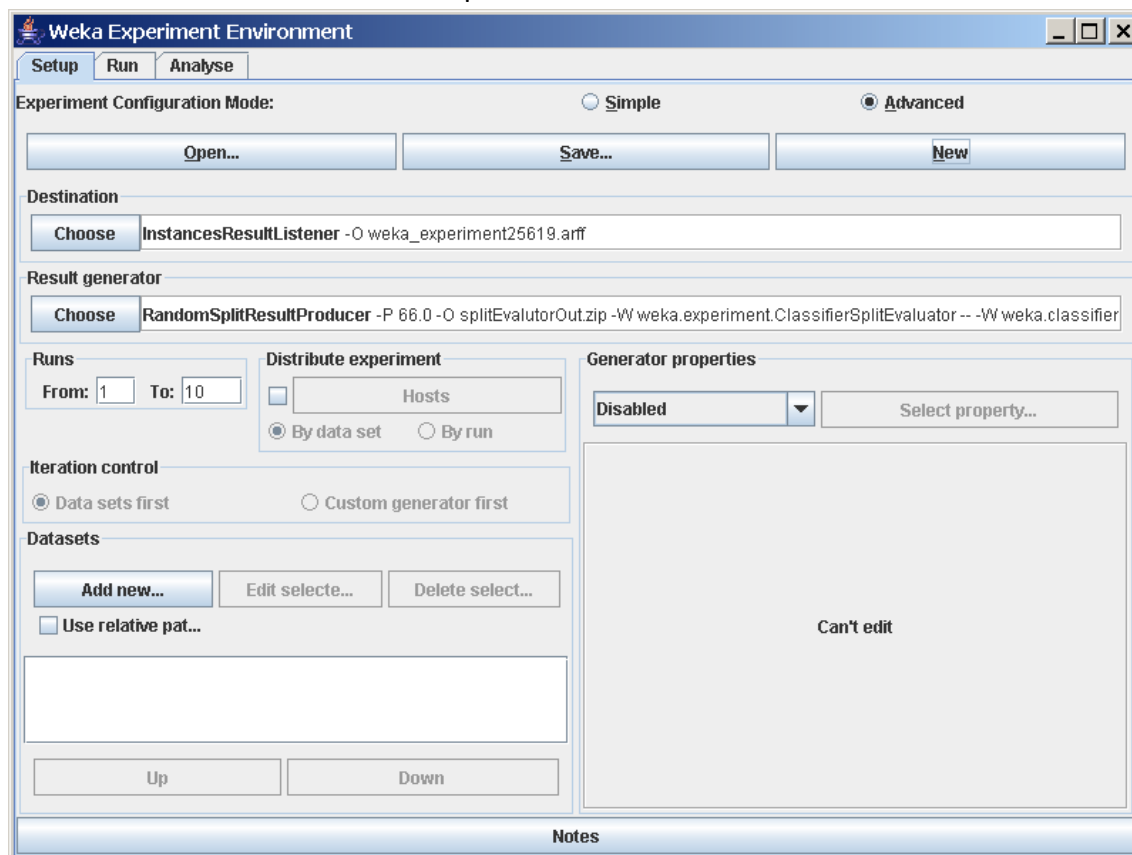
After clicking the Run tab, it shows a window with start button and stop button, by clicking on start button we can run the experiment and by clicking on stop button we can run the experiment.



If the experiment was defined correctly, the 3 messages shown above will be displayed in the Log panel.

Advanced

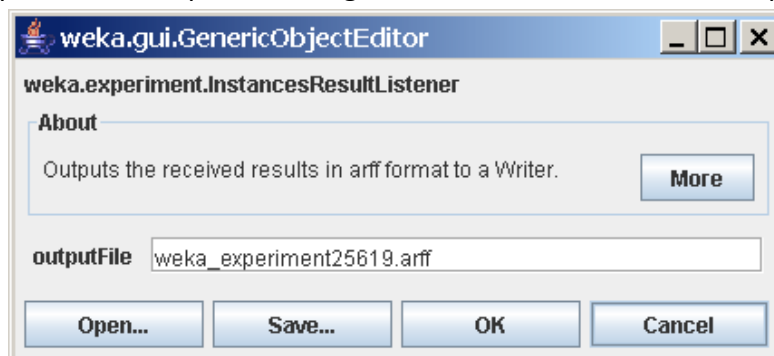
Defining an experiment: When the Experimenter is started in Advanced mode, the Setup tab is displayed. Now click New to initialize an experiment.



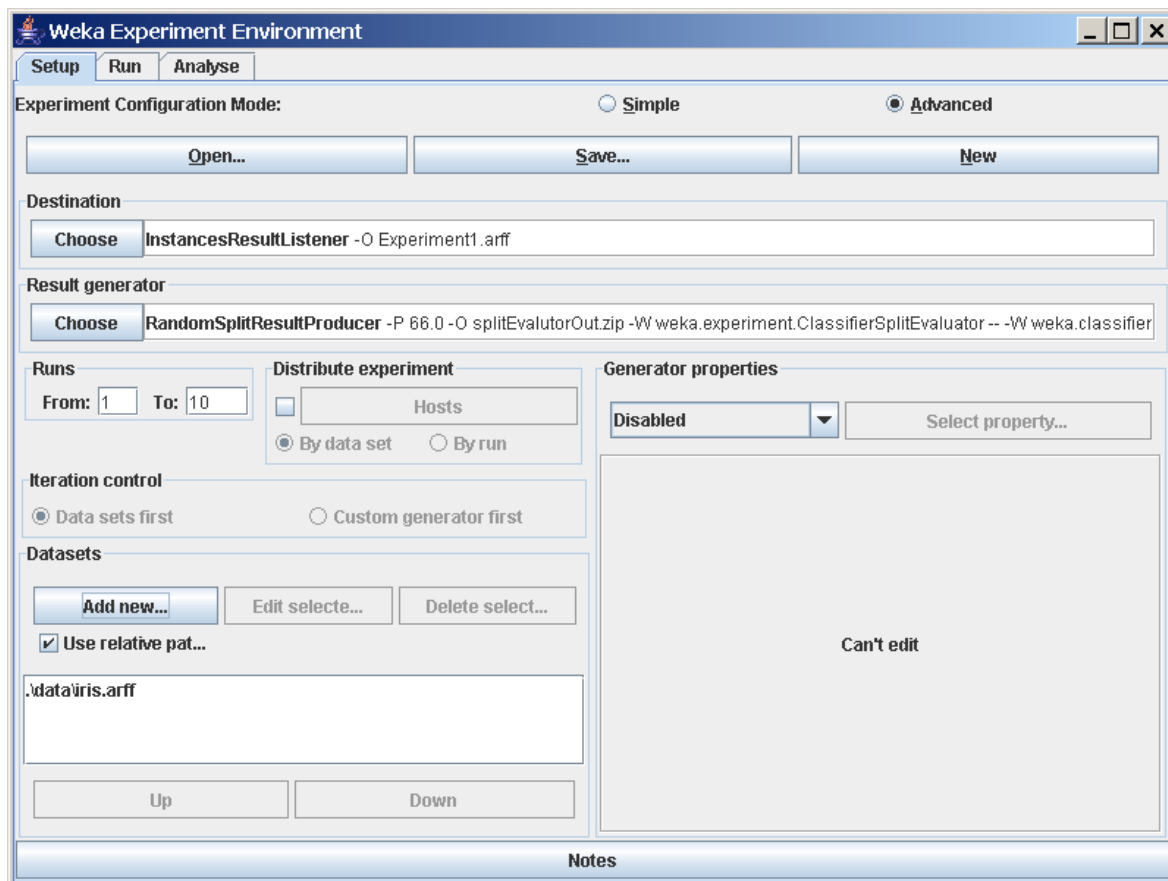
To define the dataset to be processed by a scheme, first select Use relative paths in the Datasets panel of the Setup tab and then click on Add new... button.

Saving Results of the experiment

To identify a dataset to which the results are to be sent, click on the Instances- ResultListener entry in the Destination panel, which opens a dialog box with a label named as “output file”.



Now give the name of the output file and click on OK button. The dataset name is now displayed in the Datasets panel of the Setup tab. This is as shown in the following figure:



Now we can run the experiment by clicking the Run tab at the top of the experiment environment window. The current experiment performs 10 randomized train and test runs.

To change from random train and test experiments to cross-validation experiments, click on the *Result generator* entry.

Using analysis tab in experiment environment window one can analyze the results of experiments using experiment analyzer.

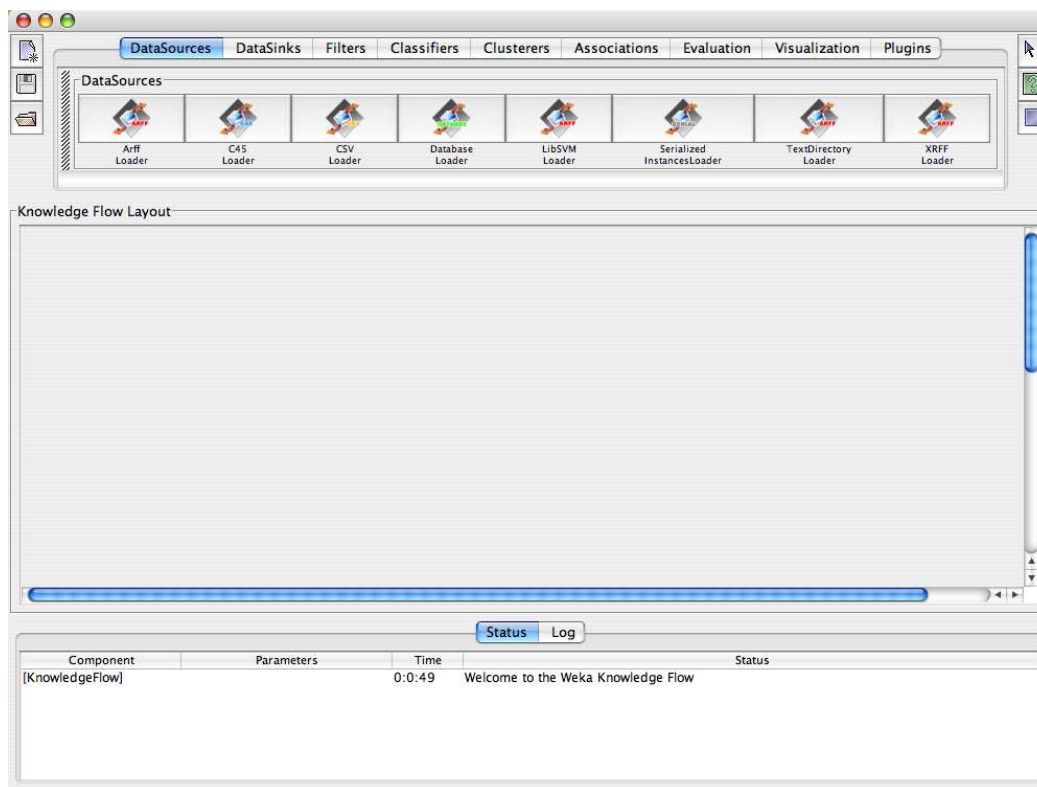
KNOWLEDGE FLOW:

The Knowledge Flow provides an alternative to the Explorer as a graphical front end to WEKA's core algorithms. It is represented as shown in the following figure. The Knowledge Flow presents a data-flow inspired interface to WEKA.

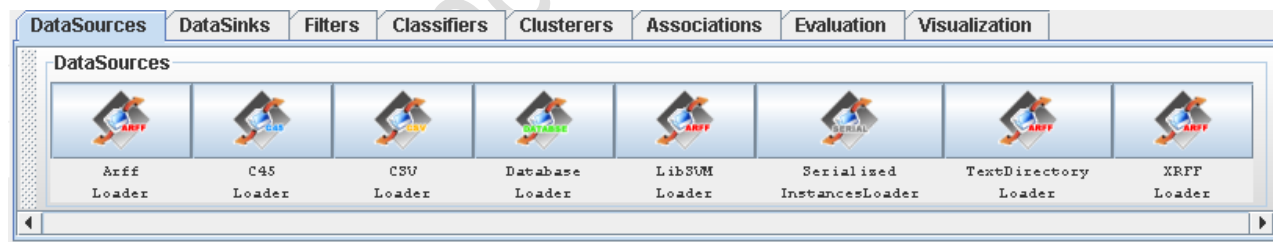
The Knowledge Flow offers the following features:

1. Intuitive data flow style layout.
2. Process data in batches or incrementally.
3. Process multiple batches or streams in parallel (each separate flow executes in its own thread).
4. Chain filters together.
5. View models produced by classifiers for each fold in a cross validation.
6. visualize performance of incremental classifiers during processing

7. Plug-in facility for allowing easy addition of new components to the Knowledge Flow.



Components:



1. Data Sources: All WEKA loaders are available.
2. Data Sinks: All WEKA savers are available.
3. Filters: All WEKA's filters are available.
4. Classifiers: All WEKA classifiers are available.
5. Clusterers: All WEKA clusterers are available.
6. Evaluation: It contains different kinds of techniques like TrainingSetMaker, TestSetMaker, CrossValidationFoldMaker, TrainTestSplitMaker, ClassAssigner, ClassValuePicker, ClassifierPerformanceEvaluator, IncrementalClassifierEvaluator, ClustererPerformanceEvaluator, and PredictionAppender.

7. Visualization: It contains different models like DataVisualizer, ScatterPlotMatrix, AttributeSummarizer, ModelPerformanceChart, TextViewer, GraphViewerbased, and StripChart.

Plug-in Facility:

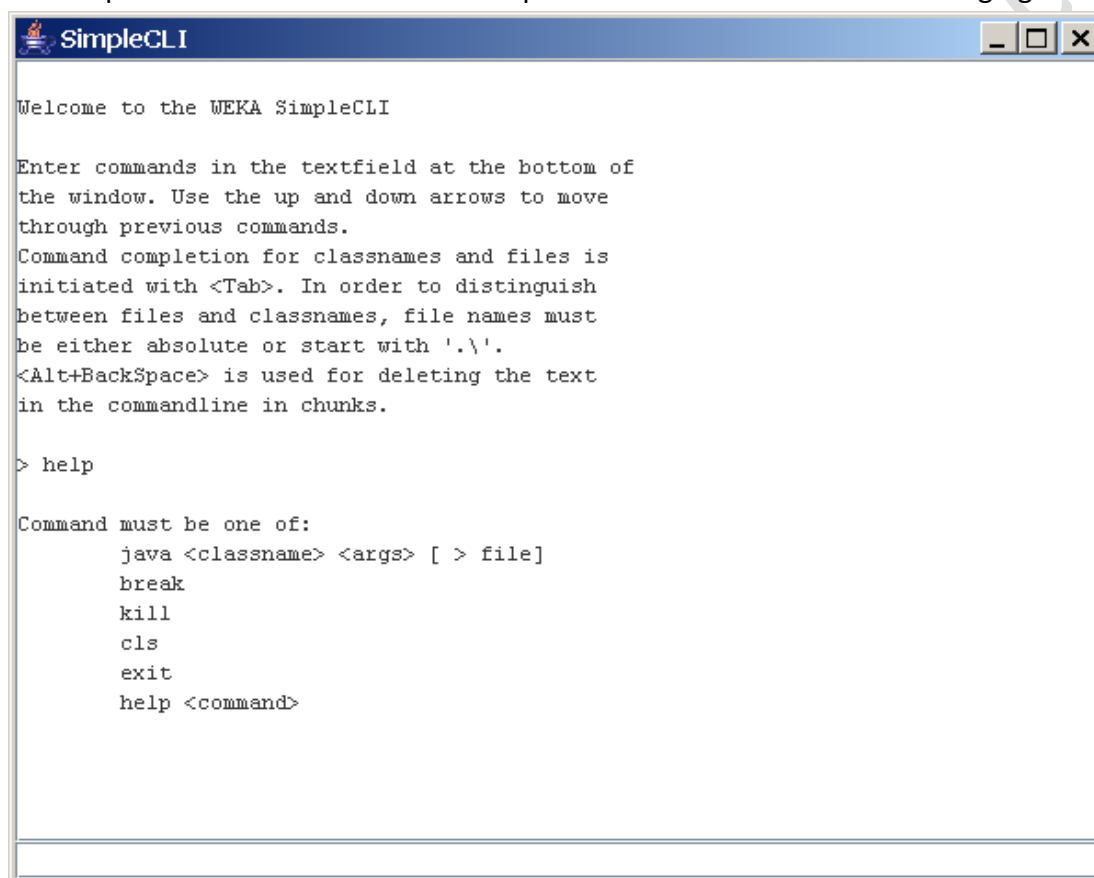
The Knowledge Flow offers the ability to easily add new components via a plug-in mechanism.

SIMPLE CLI:

The Simple CLI provides full access to all Weka classes like classifiers, filters, clusterers, etc., but without the hassle of the CLASSPATH.

It offers a simple Weka shell with separated command line and output.

The simple command line interface is represented as shown in the following figure:

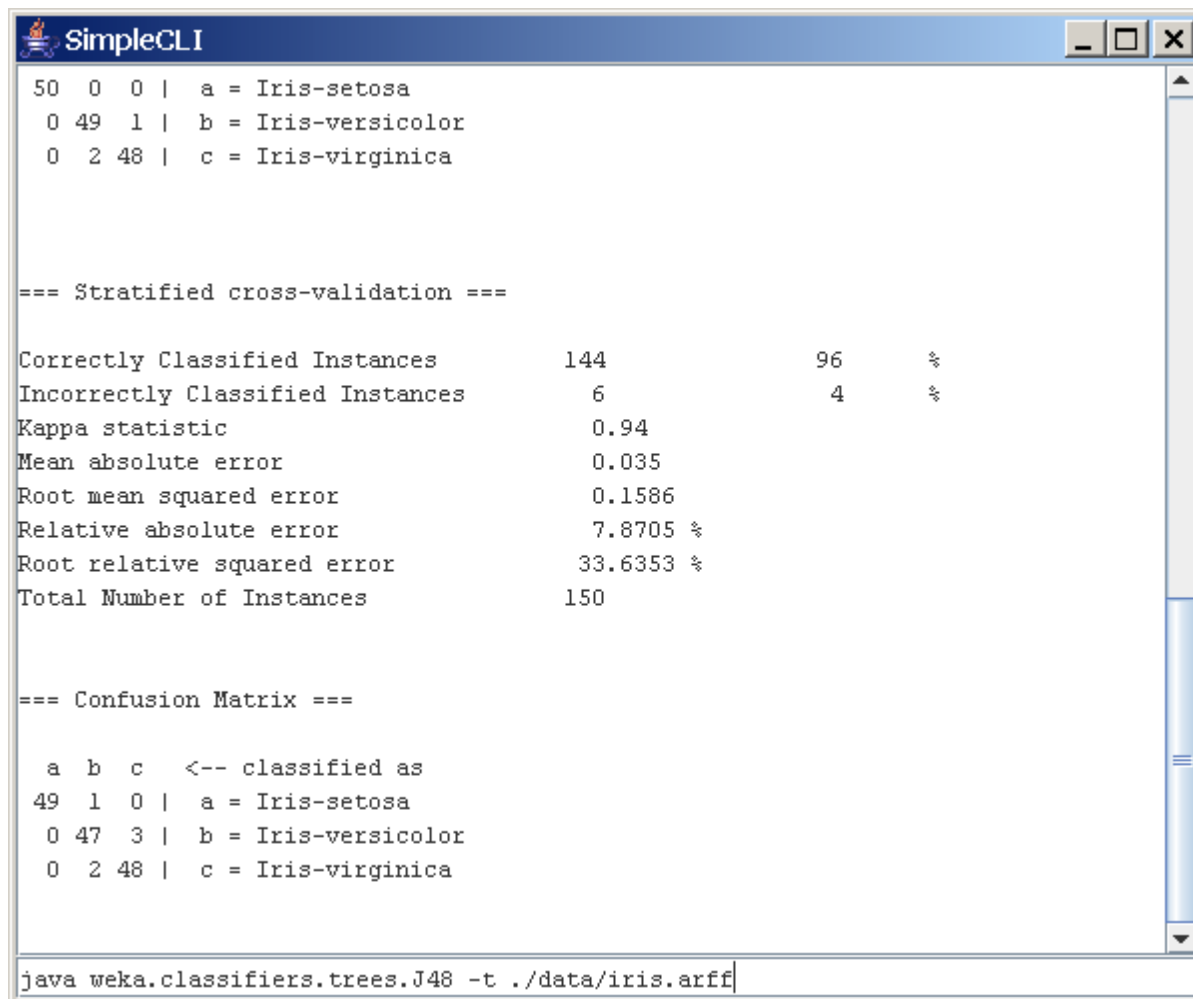


The following commands are available in the Simple CLI:

1. **java <classname> [<args>]:** - invokes a java class with the given arguments (if any)
2. **break:** - it stops the current thread in a friendly manner. e.g., a running classifier
3. **kill:** - stops the current thread in an unfriendly fashion
4. **cls:-** clears the output area
5. **exit:-** exits the Simple CLI
6. **help [<command>]:** - provides an information about the command available in the simple CLI. Also it provides an overview of all commands available if the command is not specified as an argument.

In order to invoke a Weka class, only the way is one has to prefix the class with "java". This command tells the Simple CLI to load a class and execute it with any given parameters.

For example: **java weka.classifiers.trees.J48 -t c:/temp/iris.arff** which results in the following output.



```
SimpleCLI
50  0  0 | a = Iris-setosa
  0 49  1 | b = Iris-versicolor
  0  2 48 | c = Iris-virginica

=== Stratified cross-validation ===

Correctly Classified Instances      144           96    %
Incorrectly Classified Instances     6            4    %
Kappa statistic                     0.94
Mean absolute error                  0.035
Root mean squared error              0.1586
Relative absolute error              7.8705 %
Root relative squared error          33.6353 %
Total Number of Instances           150

=== Confusion Matrix ===

  a  b  c  <-- classified as
49  1  0 | a = Iris-setosa
  0 47  3 | b = Iris-versicolor
  0  2 48 | c = Iris-virginica

java weka.classifiers.trees.J48 -t ./data/iris.arff
```

Using simple CLI we can also perform command redirection using the operator ">". For example: **java weka.classifiers.trees.J48 -t test.arff > j48.txt**