

Weka 3.6 Tutorial

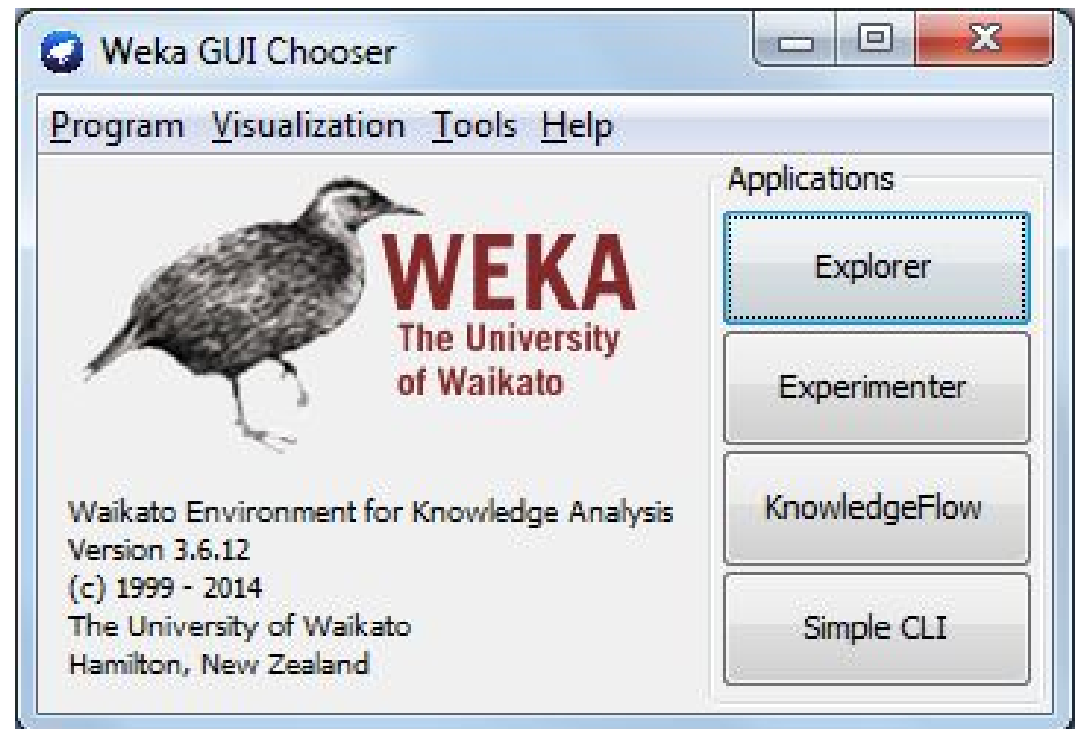
(**W**aikato **E**nvironment for **K**nowledge **A**nalysis)

WEKA

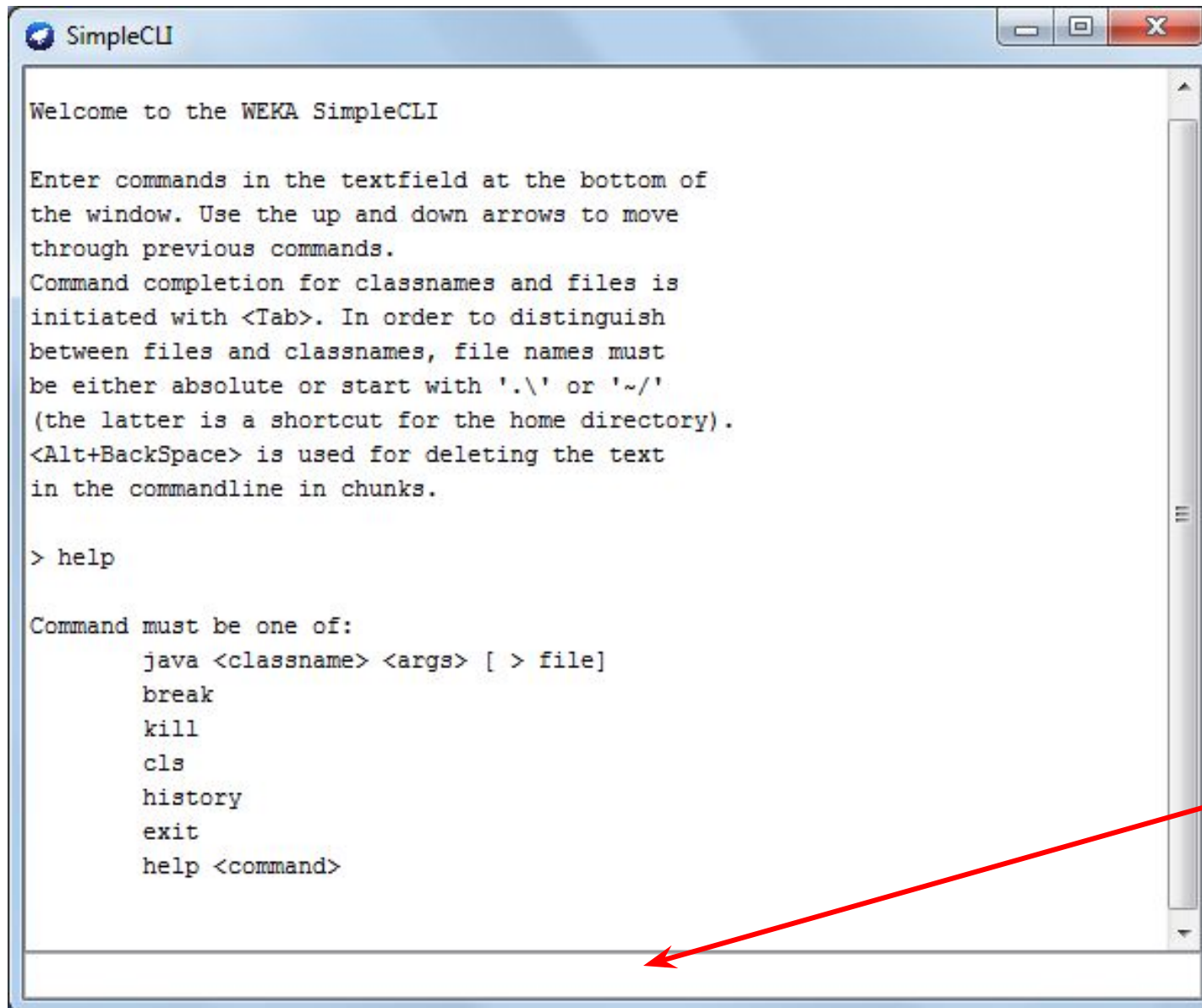
- It's a data mining/machine learning tool developed by University of Waikato.
- Main Features:
 - 49 data preprocessing tools
 - 76 classification/regression algorithms
 - 8 clustering algorithms
 - 3 algorithms for finding association rules
 - 15 attribute/subset evaluators + 10 search algorithms for feature selection

Starting WEKA

- 4 Options
 - Explorer
 - Experimenter
 - Knowledge Flow
 - Simple CLI



Weka Simple CLI



The image shows a window titled "SimpleCLI" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains a text area with the following text:

```
Welcome to the WEKA SimpleCLI

Enter commands in the textfield at the bottom of
the window. Use the up and down arrows to move
through previous commands.
Command completion for classnames and files is
initiated with <Tab>. In order to distinguish
between files and classnames, file names must
be either absolute or start with './' or '~/ '
(the latter is a shortcut for the home directory).
<Alt+BackSpace> is used for deleting the text
in the commandline in chunks.

> help

Command must be one of:
  java <classname> <args> [ > file]
  break
  kill
  cls
  history
  exit
  help <command>
```

At the bottom of the window, there is a text input field. A red arrow points from the text "Commands here" to this input field.

Commands here

Preprocessing Data

- Data can be imported from a file in various formats: ARFF, CSV etc.
- Data can also be read from a URL or from an SQL database (using JDBC)
- Pre-processing tools in WEKA are called “filters”

CSV (Comma Separated) File

```
Roll,Name,Percentage,Passed  
1,ABC,72,y  
2,abc,30.8,n  
3,xyz,44.3,n  
4,XYZ,52.3,y
```

Result.csv

Command for converting csv to arff:

```
java weka.core.converters.CSVLoader Result.csv > Result.arff
```

ARFF File

Description of dataset

@relation Result

@attribute Roll numeric

@attribute Name {ABC,abc,xyz,XYZ}

@attribute Percentage numeric

@attribute Passed {y,n}

Numeric

Nominal

@data

1,ABC,72,y

2,abc,30.8,n

3,xyz,44.3,n

4,XYZ,52.3,y

} weka.core.Instance

Weka.Classifiers

- A simple example

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

Classifier: Decision tree

Specifies training file

Classifier: NaiveBayes

```
java weka.classifiers.bayes.NaiveBayes -t data/weather.arff
```

Weka.Classifiers

- Classifier's option:
 - `weka.classifiers.trees.J48`
 - `weka.classifiers.bayes.NaiveBayes`
 - `weka.classifiers.functions.Logistic`
 - `weka.classifiers.functions.SMO`

Weka.Classifiers

- Other options:
 - -t : Specifies training file
 - -T: Specifies test file
 - -x: Number of cross validation
 - -c: set the class index
 - -d: Save the model
 - -i: Detailed performance description
 - -p #: Prediction and one attribute value

Few more complex examples

```
java -Xmx1024m weka.classifiers.trees.J48 -t data.arff -i -x 3
```

```
java -Xmx1024m weka.classifiers.trees.J48 -t data.arff -i -T  
testData.arff -p 1
```

```
java -Xmx1024m weka.classifiers.trees.J48 -t data.arff -i -T  
testData.arff -d J48-data.model >&! J48-data.out &
```

```
java -Xmx1024m weka.classifiers.meta.FilteredClassifier -t data.arff -i  
-T testData.arff -i -c last -F  
weka.filters.unsupervised.attribute.StringToWordVector -W  
weka.classifier.functions.SMO > Output
```

```
java -Xmx1024m -cp weka.jar:LibSVM.jar  
weka.classifiers.meta.FilteredClassifier -t data.arff -i -T  
testData.arff -i -c last -F  
weka.filters.unsupervised.attribute.StringToWordVector -W  
weka.classifier.functions.LibSVM > Output
```

Weka Filters

- Used to transform input data
 - Removing or adding attributes
 - Resampling the dataset
 - Removing examples
 - ...