Tools for Machine Learning

- **Weka:** Waikato Environment for Knowledge Analysis
- **Scikit-learn:** Machine learning in Python
- **R:** Free software environment for statistical computing and visualization
- **Matlab:** Tools for solving engineering and scientific problems
- **Octave:** A free open source environment similar to Matlab
- **MOA:** Massive online analysis – similar to Weka, for big data
- **Scilab:** Another free open source software for numerical computations
- **LibSVM:** A library for Support Vector Machines
- **Deep learning:**
  - Tensorflow
  - Theano
  - Torch7
  - Caffe
  - Keras

Weka

- **W**aikato **E**nvironment for **K**nowledge **A**nalysis
  - It's a data mining/machine learning tool developed by Department of Computer Science, University of Waikato, New Zealand.
  - The Weka, or woodhen, is a bird native to New Zealand.

- Website: http://www.cs.waikato.ac.nz/ml/weka/
- Supports both 32 bit and 64 bit architectures
- Download the developer's version (3.8) which has the package manager (used to install packages more easily)

Weka

- History:
  - 1st version (version 2.1) 1996
  - Version 2.3: 1998
  - Version 3.0: 1999
  - Version 3.4: 2003
  - Version 3.6: 2008
  - Version 3.7: 2012
  - Version 3.8: current

- WEKA provides a collection of data mining, machine learning algorithms and preprocessing tools.
- It includes algorithms for regression, classification, clustering, association rule mining and attribute selection.
- WEKA is an environment for:
  - data visualization
  - comparing learning algorithms
  - implementing new data mining algorithms to add to the Weka as a package
- WEKA is the best-known open-source data mining software.

Weka

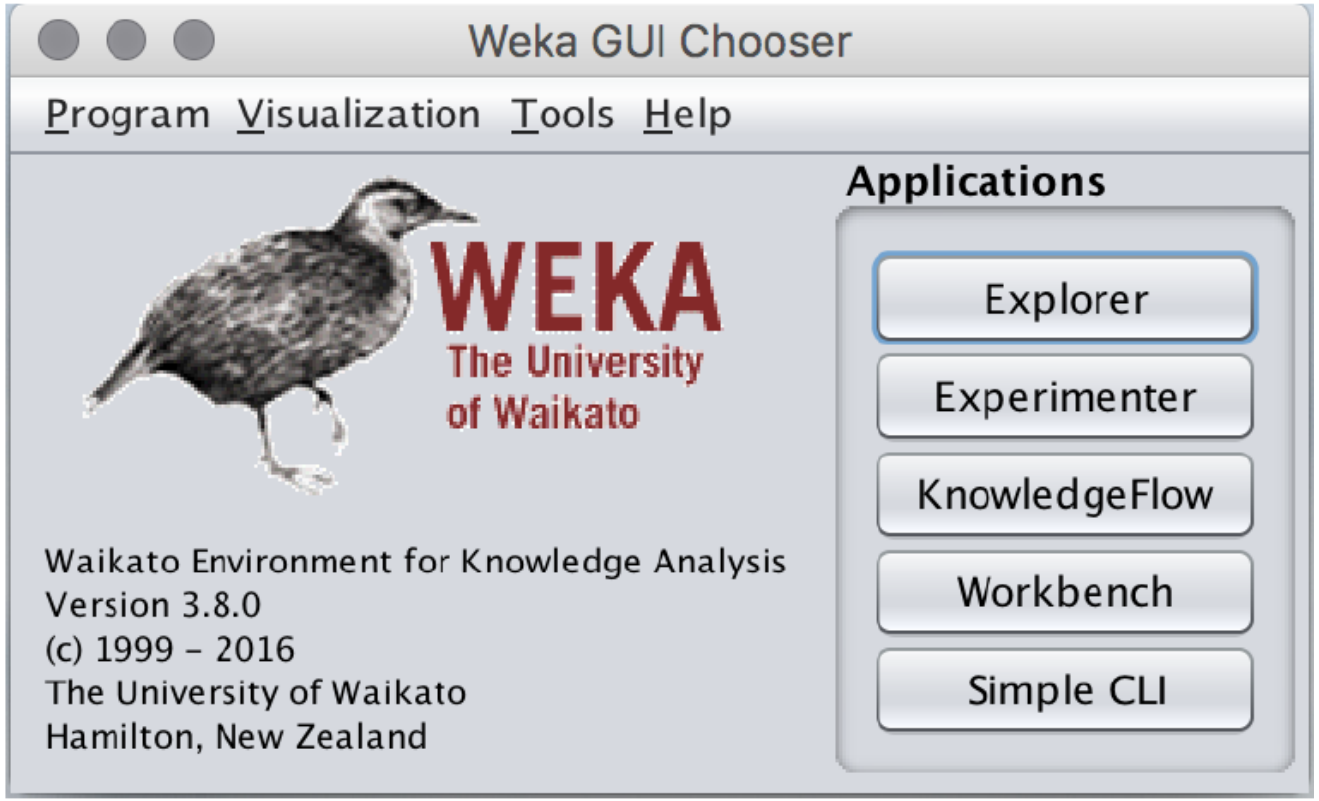
- WEKA is written in Java.
  - WEKA 3.4 consists of 271,477 lines of code.
  - WEKA 3.6 consists of more than 500,000 lines of code.
- It works on Windows, Linux and Macintosh.
- Users can access its components through Java programming or through a command-line interface as well as a GUI.

- 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

Weka – Main GUI

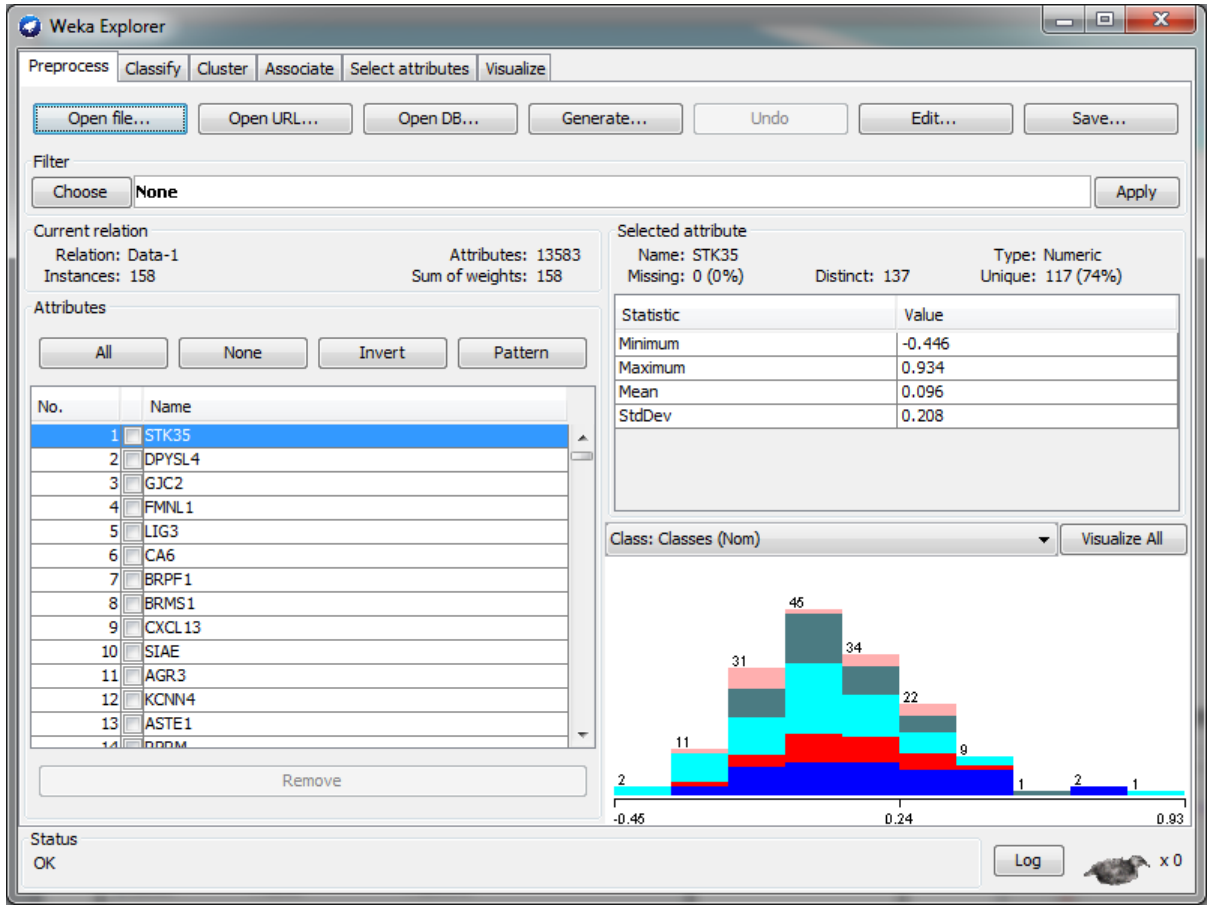
The GUI Chooser

- "The Explorer" (exploratory data analysis)
- "The Experimenter" (experimental environment)
- "The KnowledgeFlow" (new process model inspired interface)
- " The Simple CLI"- provides users without a graphic interface option the ability to execute commands from a terminal window



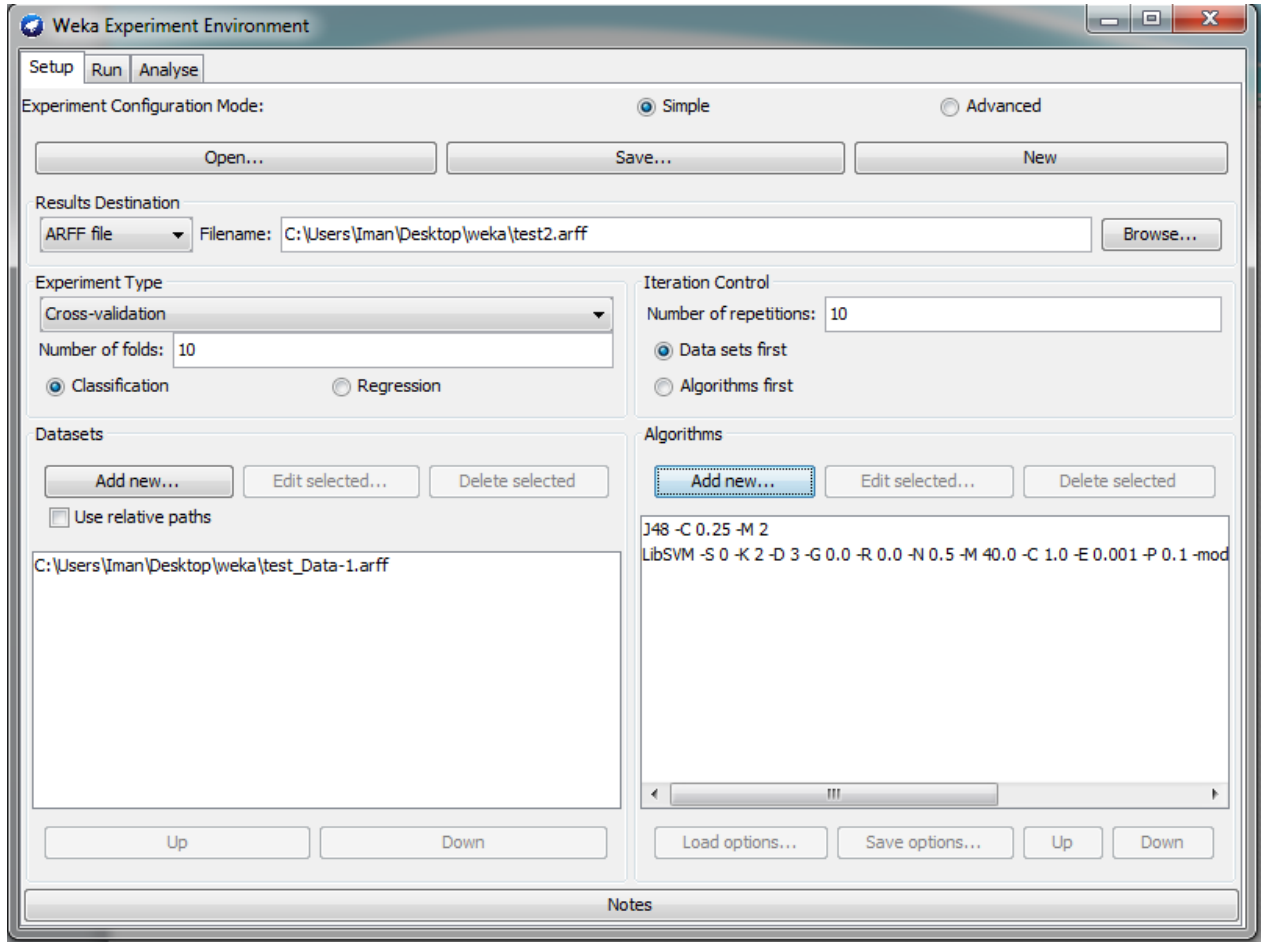
Weka - Explorer

- Consists of 6 panels, each for one data mining task:
  - *Preprocess*
  - *Classify*
  - *Cluster*
  - *Associate*
  - *Select Attributes*
  - *Visualize*.



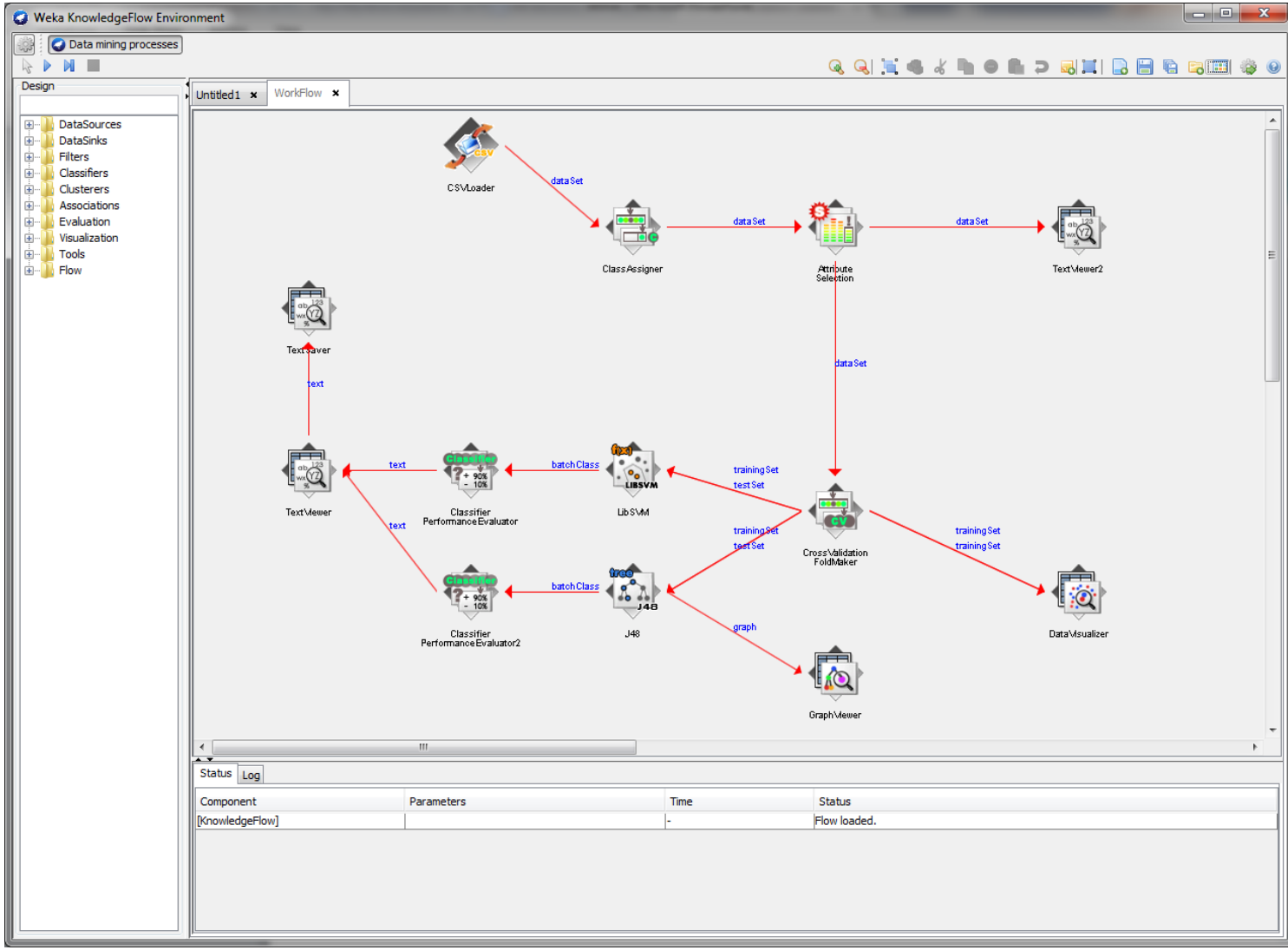
Weka - Experimenter

- This interface is designed to facilitate *experimental comparisons* of the performance of algorithms based on many different evaluation criteria.
- Experiments can involves many algorithms that are run on multiple datasets.
- Can also iterate over different parameter settings
- Experiments can also be distributed across different computer nodes in a network.
- Once an experiment has been set up, it can be saved in either XML or binary form, so that it can be re-visited.

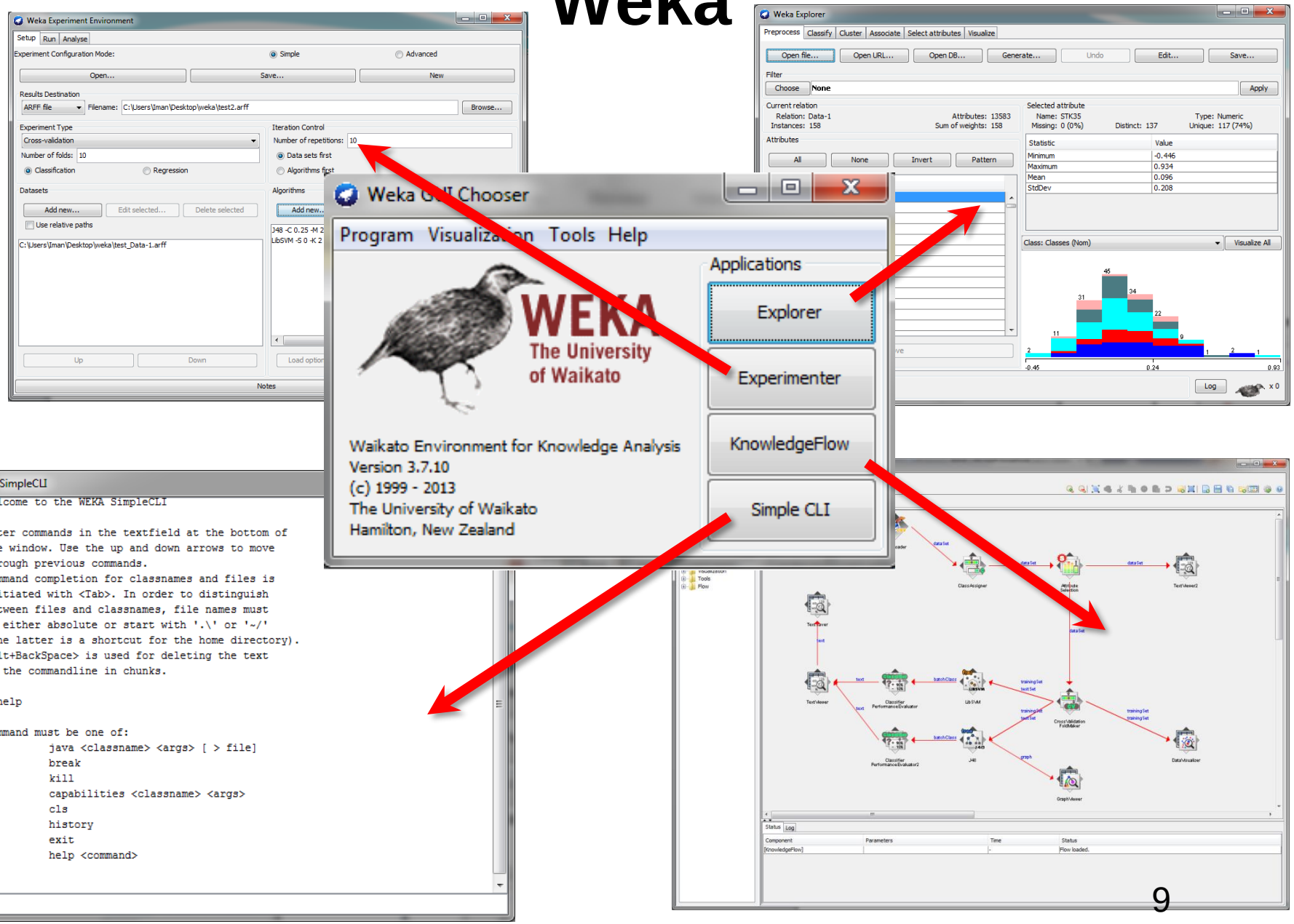


Weka – Knowledge Flow Interface

- The Explorer is designed for batch-based data processing: training data is loaded into memory and then processed.
- However WEKA has implemented some *incremental algorithms*.
- Knowledge-flow interface can handle *incremental updates*. It can load and preprocess individual instances before feeding them into incremental learning algorithms.
- Knowledge-flow also provides nodes for visualization and evaluation.



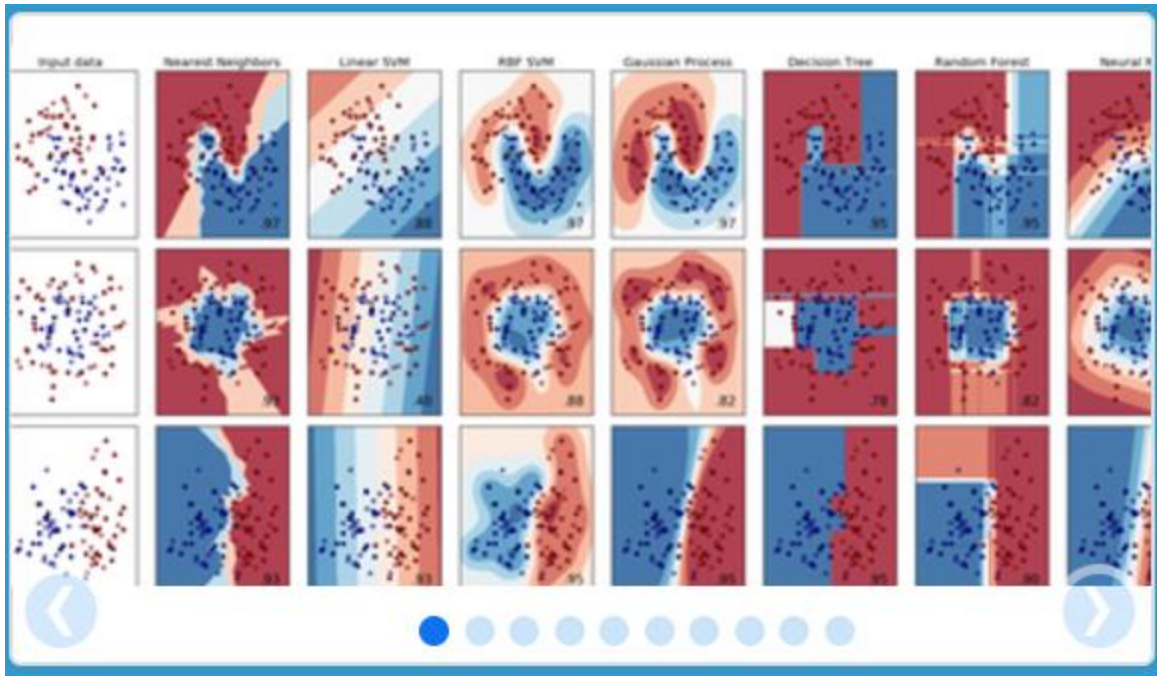
Weka



Scikit-learn: Machine Learning in Python

- Tools for data mining and data analysis
- Developed in Python and used in Python
- Open source, free, and can be used for commercial purposes
- Need not be installed separately
  - Included in latest versions of Python (e.g., >= 2.6)
- Includes ML algorithms for
  - classification, regression, clustering, dimensionality reduction, model selection, preprocessing

http://scikit-learn.org/stable/



Scikit-learn

- Main website:
  - http://scikit-learn.org/stable/
- Includes:
  - Installation instructions
  - Quick start
  - Tutorials
  - User guide
  - Resources
  - Full examples

- Who uses Scikit?
  - Spotify
  - Evernote
  - Booking.com
  - Peerindex
  - Many others…

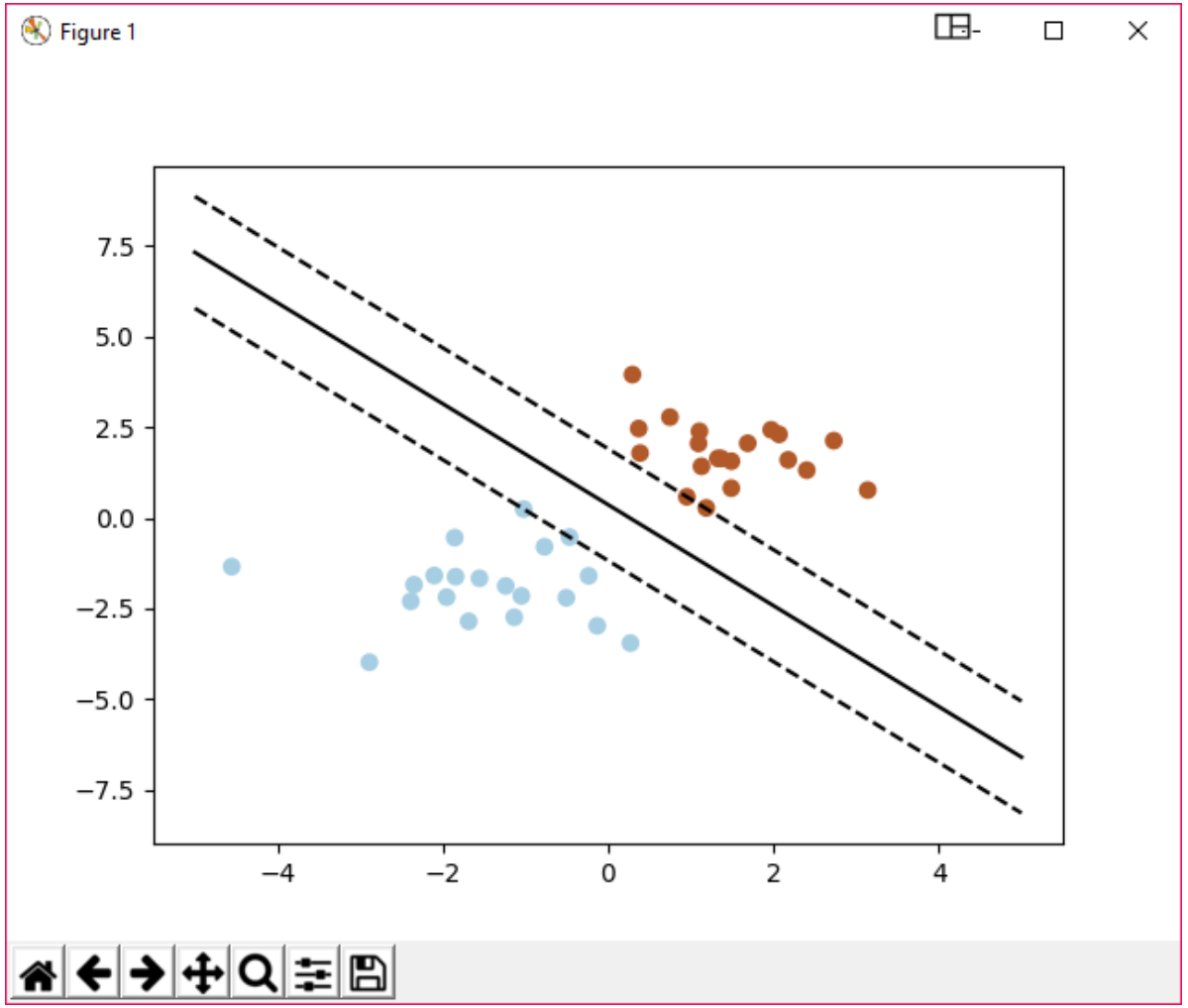
Scikit-learn – Algorithms/Models

- Supervised learning
  - Linear, quadratic
  - SVM
  - kNN
  - Naïve Bayes
  - Decision trees, random forest
  - Neural networks
  - Feature selection
- Unsupervised learning
  - Gaussian mixture – EM, k-Means
  - Regression
  - Bi-clustering, co-clustering
  - Outlier detection
  - Density estimation
  - Neural networks
- Model selection/evaluation
  - Cross validation
  - Validation curves
  - Parameter selection, tuning
- Data transformation
  - Feature extraction
  - Dimensionality reduction
  - Kernel approximation
  - Random projection
  - Supervised dimensionality reduction - LDA
- Incremental learning
  - Big data mining

Scikit-learn - example

python.exe - Shortcut

```
>>> a = -w[0] / w[1]
>>> xx = np.linspace(-5, 5)
>>> yy = a * xx - (clf.intercept_[0]) / w[1]
>>>
>>> # plot the parallels to the separating hyperplane that pass through the
... # support vectors
... b = clf.support_vectors_[0]
>>> yy_down = a * xx + (b[1] - a * b[0])
>>> b = clf.support_vectors_[-1]
>>> yy_up = a * xx + (b[1] - a * b[0])
>>>
>>>
>>> # plot the line, the points, and the nearest vectors to the plane
... plt.plot(xx, yy, 'k-')
[<matplotlib.lines.Line2D object at 0x000002163E320C50>]
>>> plt.plot(xx, yy_down, 'k--')
[<matplotlib.lines.Line2D object at 0x000002163A41B6A0>]
>>> plt.plot(xx, yy_up, 'k--')
[<matplotlib.lines.Line2D object at 0x000002163E320E48>]
>>>
>>> plt.scatter(clf.support_vectors_[:, 0], clf.support_vectors_[:, 1],
...             s=80, facecolors='none')
<matplotlib.collections.PathCollection object at 0x000002163E32DE48>
>>> plt.scatter(X[:, 0], X[:, 1], c=Y, cmap=plt.cm.Paired)
<matplotlib.collections.PathCollection object at 0x000002163E33ADA0>
>>>
>>> plt.axis('tight')
(-5.5, 5.5, -8.9908292748568677, 9.7145234464309773)
>>> plt.show()
```



Tools for Deep Learning

- **TensorFlow:** Open source library for machine learning and deep learning.
- **Keras:** NN API that runs on top of TensorFlow, **CNTK**, or **Theano**.
- **Caffe:** Open source written in C++ -- developed at Berkeley.
- **Torch 7:** Based on scripting language LuaJIT – good for CNN.
- Theano: Written in C++/Python – good for RNNs; slow compiling.

	Caffe	Theano	Torch7	TensorFlow
Core language	C++	Python, C++	LuaJIT	C++
Interfaces	Python, Matlab	Python	C	Python
Wrappers		Lasagne, Keras, sklearn-theano		Keras, Pretty Tensor, Scikit Flow
Programming paradigm	Imperative	Declarative	Imperative	Declarative
Well suited for	CNNs, Reusing existing models, Computer vision	Custom models, RNNs	Custom models, CNNs, Reusing existing models	Custom models, Parallelization, RNNs

References

- Weka Workbench: Online reference

http://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf

- Data mining book:
  - http://www.cs.waikato.ac.nz/ml/weka/book.html
- A. Muller, S. Guido. Introduction to Machine Learning with Python, O'Reilly, 2016.
- Scikit-learn:
  - http://scikit-learn.org/stable
- See the Resources tab for more useful links