

WEKA only deals with “flat” files

Data Mining with WEKA (Data Preprocessing)

```

@relation heart-disease-simplified

@attribute age numeric
@attribute sex { female, male}
@attribute chest_pain_type { typ_angina, asympt, non_anginal, atyp_angina}
@attribute cholesterol numeric
@attribute exercise_induced_angina { no, yes}
@attribute class { present, not_present}

@data
63,male,typ_angina,233,no,not_present
67,male,asympt,286,yes,present
67,male,asympt,229,yes,present
38,female,non_anginal,?,no,not_present
...

```

Flat file in ARFF format

WEKA: the software

- WEKA: Data Mining Software in Java
- The Explorer
 - Data Preprocessing
 - Classification and Regression
 - Clustering
 - Association Rules
 - Attribute Selection
 - Data Visualization
- The Experimenter
- The Knowledge Flow GUI

WEKA only deals with “flat” files

```

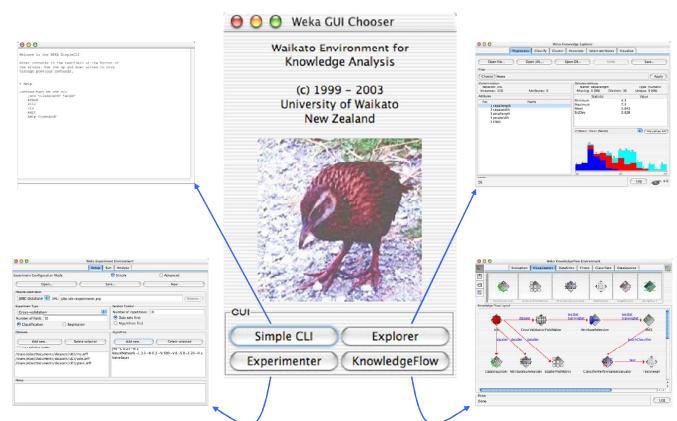
@relation heart-disease-simplified
numeric attribute
@attribute age numeric
nominal attribute
@attribute sex { female, male}
@attribute chest_pain_type { typ_angina, asympt, non_anginal, atyp_angina}
@attribute cholesterol numeric
@attribute exercise_induced_angina { no, yes}
@attribute class { present, not_present}

@data
63,male,typ_angina,233,no,not_present
67,male,asympt,286,yes,present
67,male,asympt,229,yes,present
38,female,non_anginal,?,no,not_present
...

```

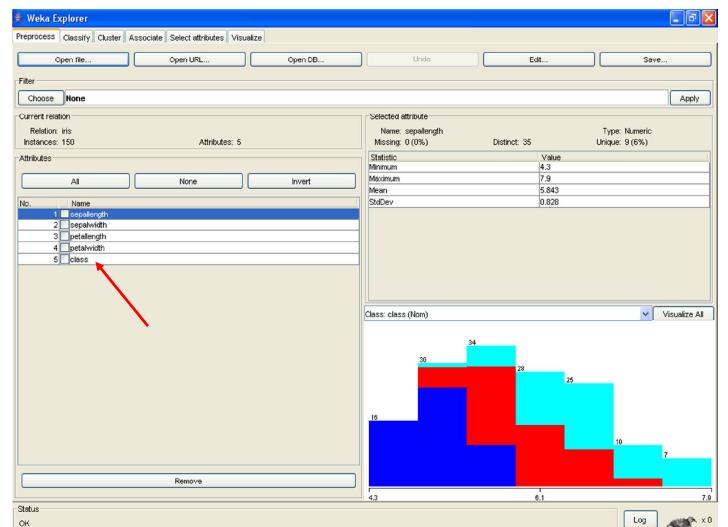
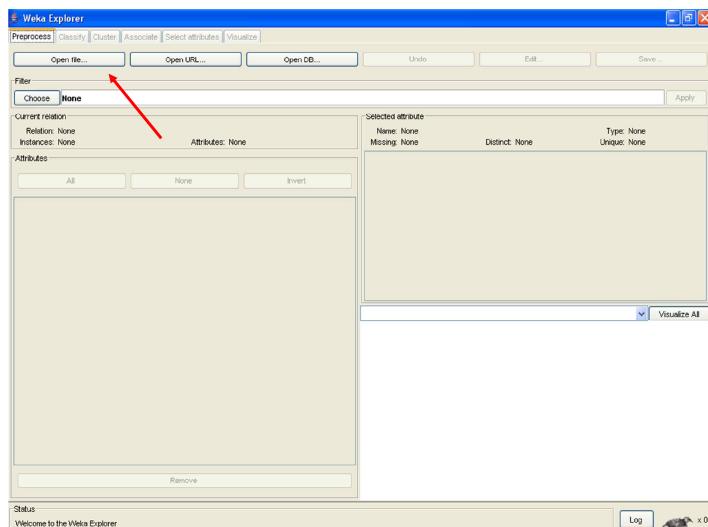
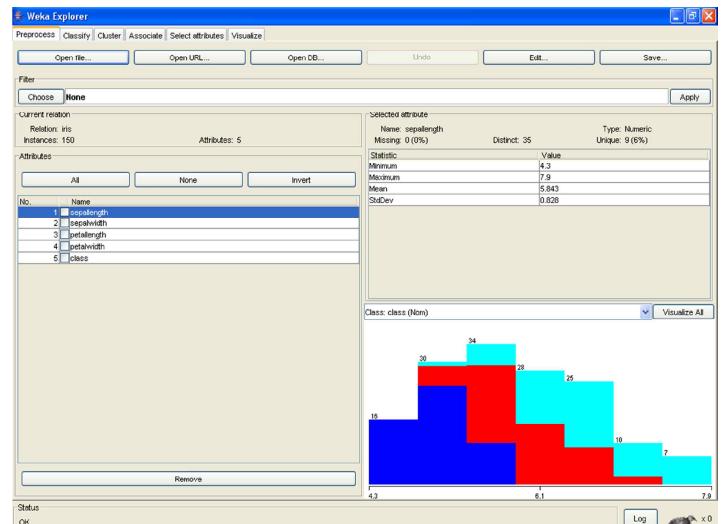
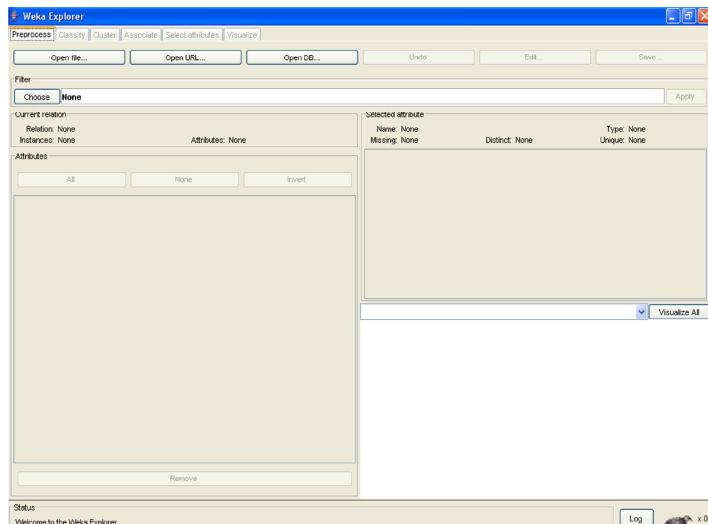
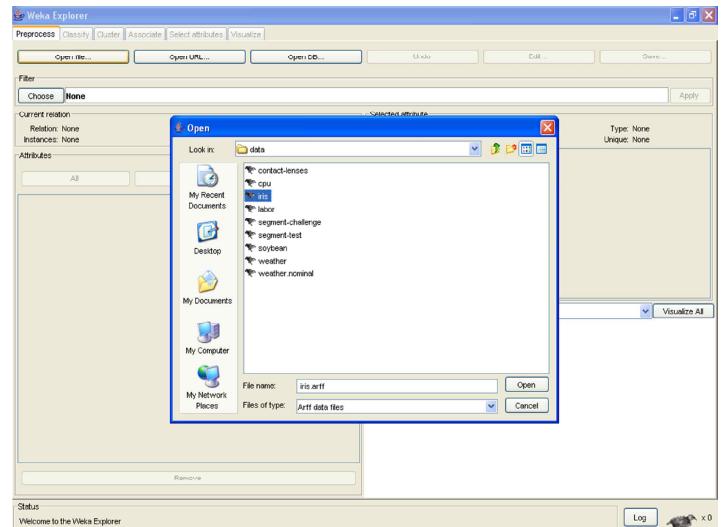
WEKA: the software (2)

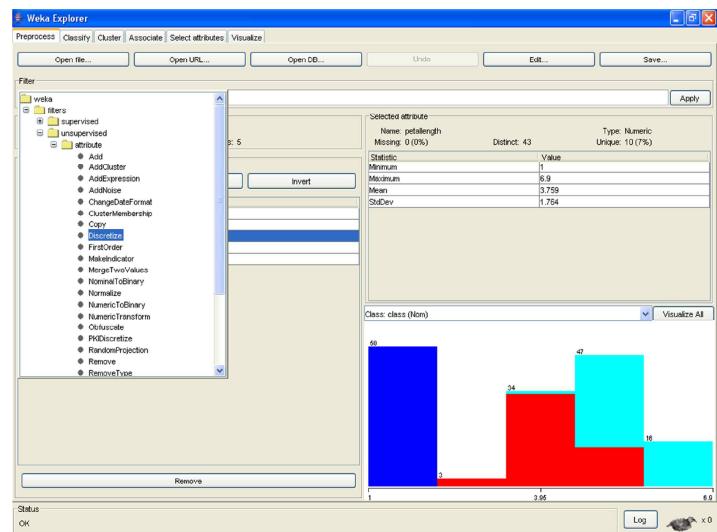
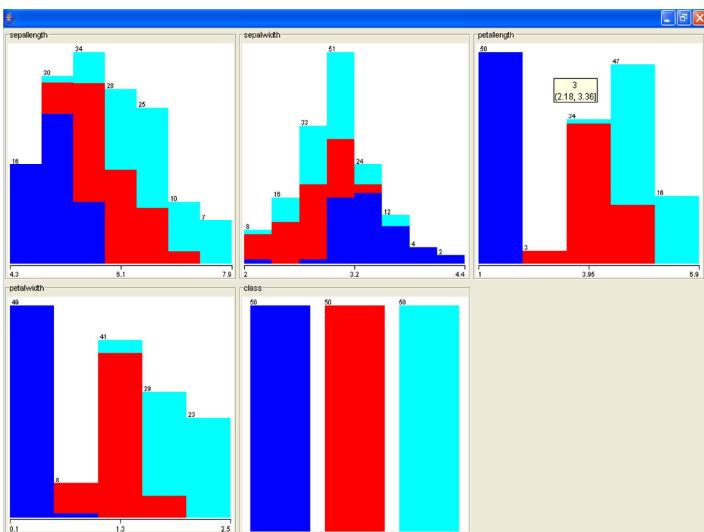
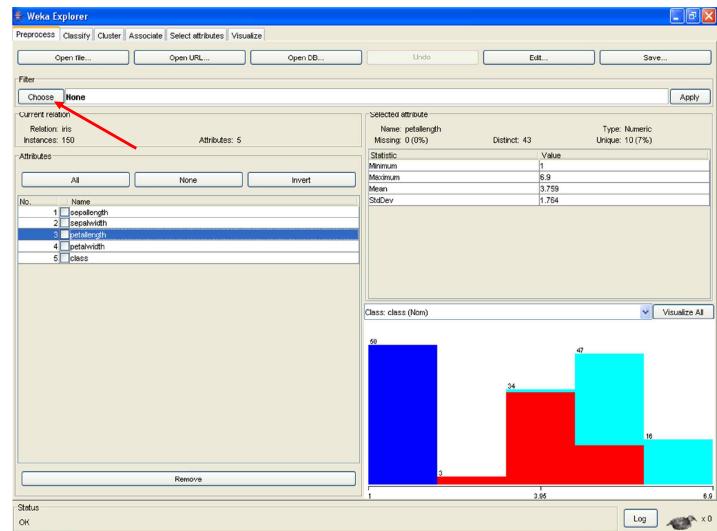
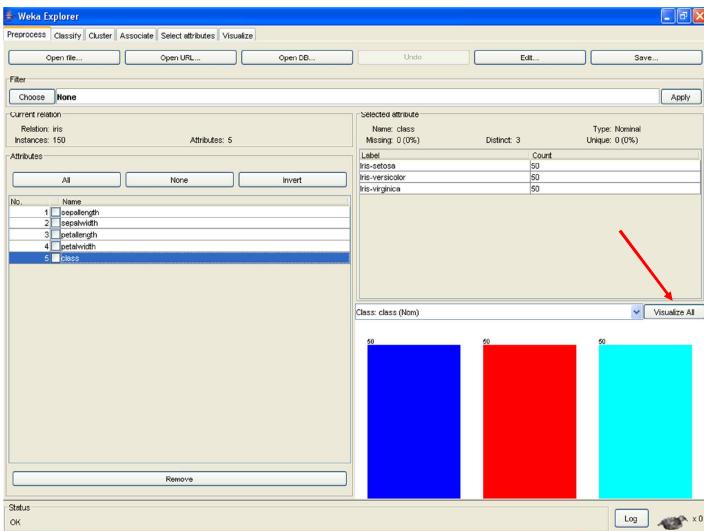
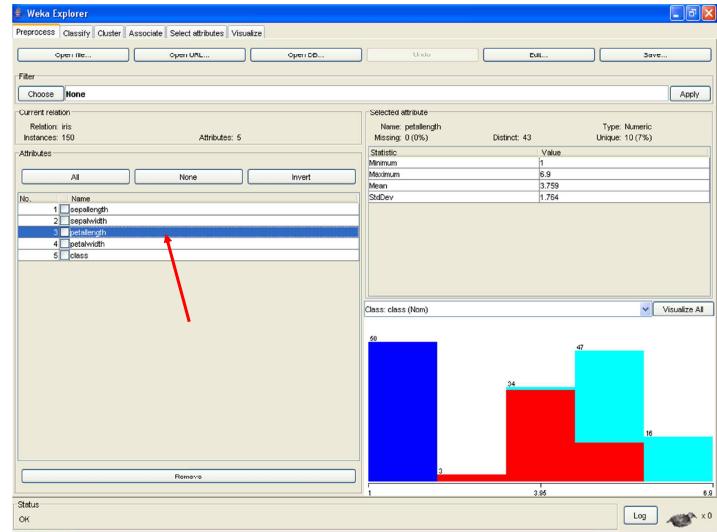
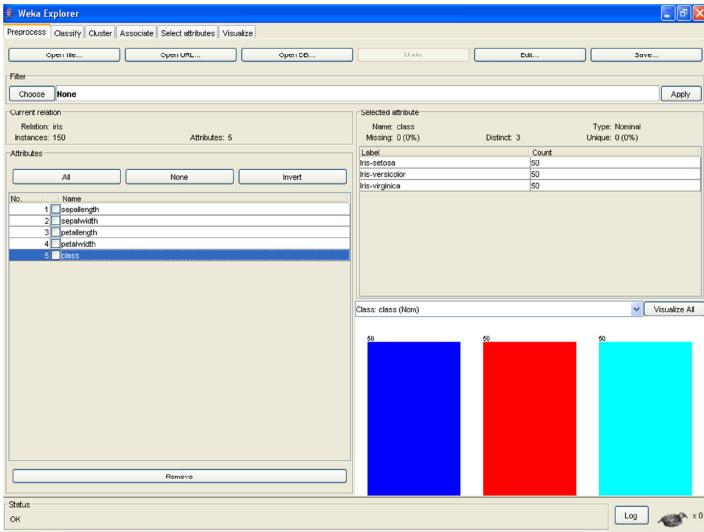
- Machine learning/data mining software written in Java (distributed under the GNU Public License)
- Used for research, education, and applications
- Complements “Data Mining” by Witten & Frank
- Main features:
 - Comprehensive set of data pre-processing tools, learning algorithms and evaluation methods
 - Graphical user interfaces (incl. data visualization)
 - Environment for comparing learning algorithms

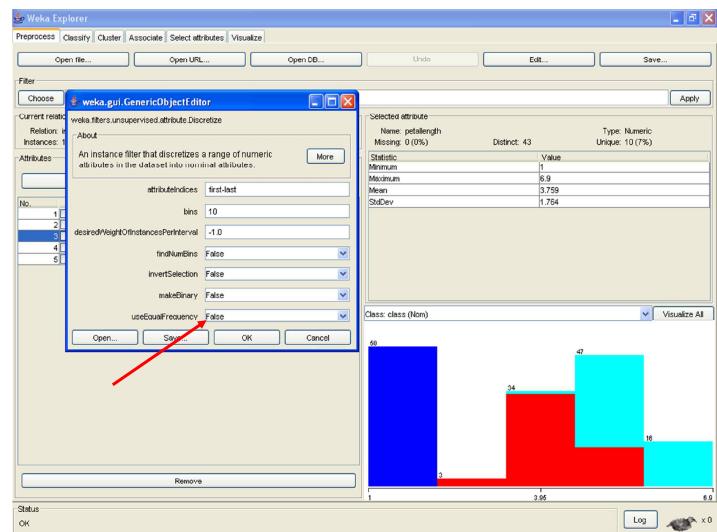
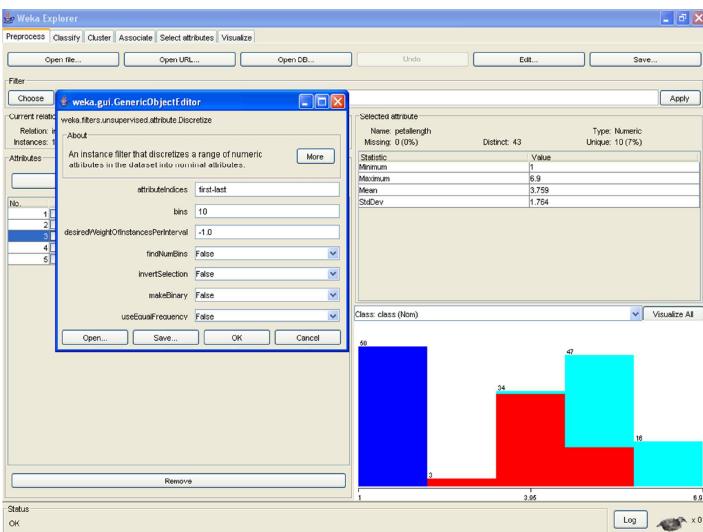
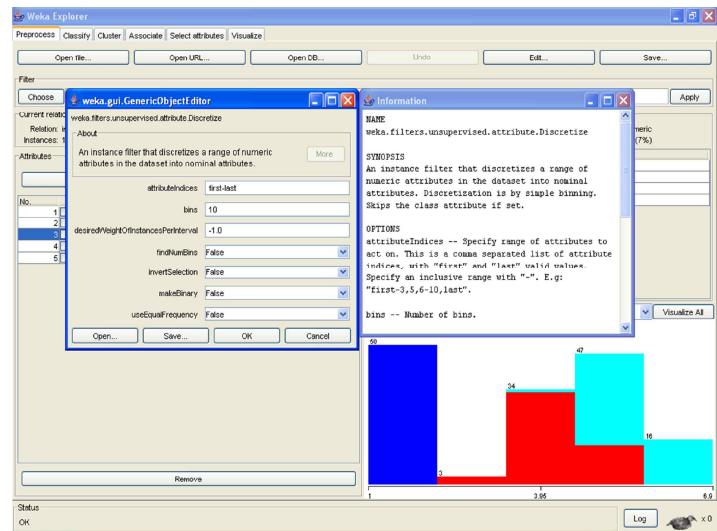
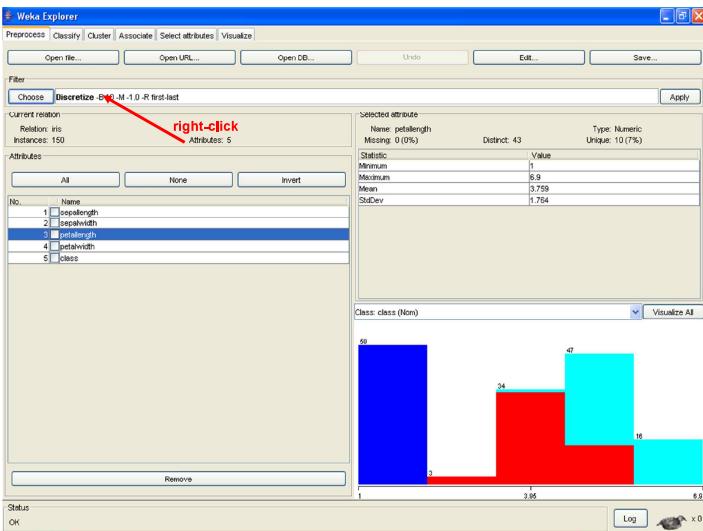
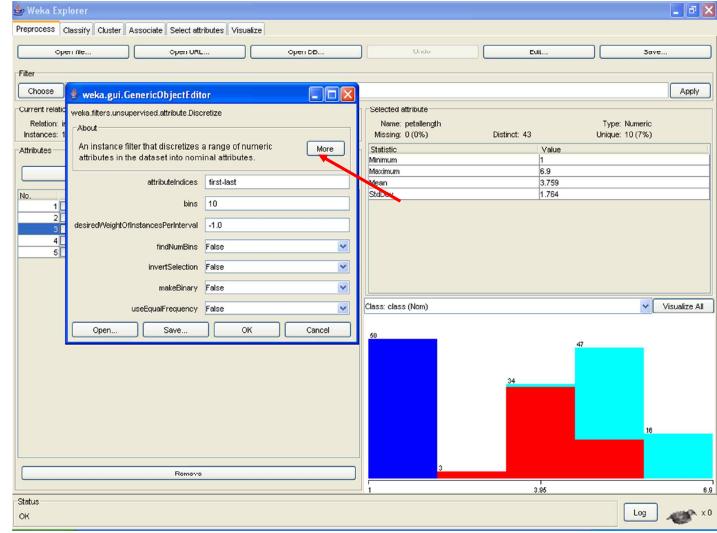
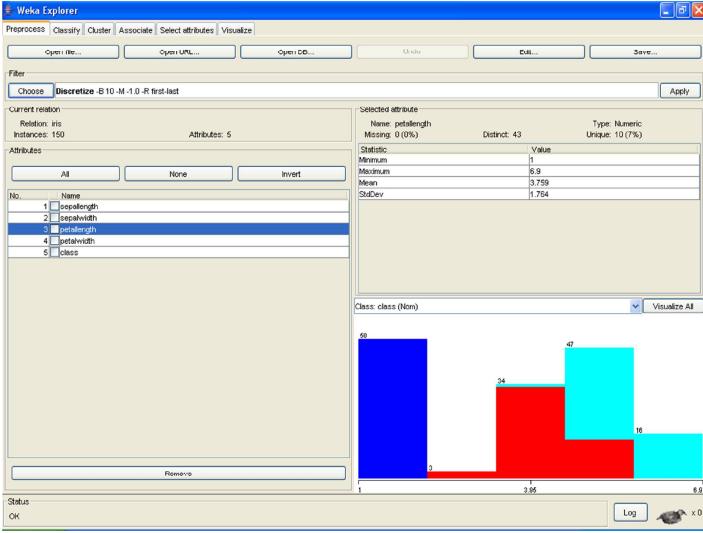


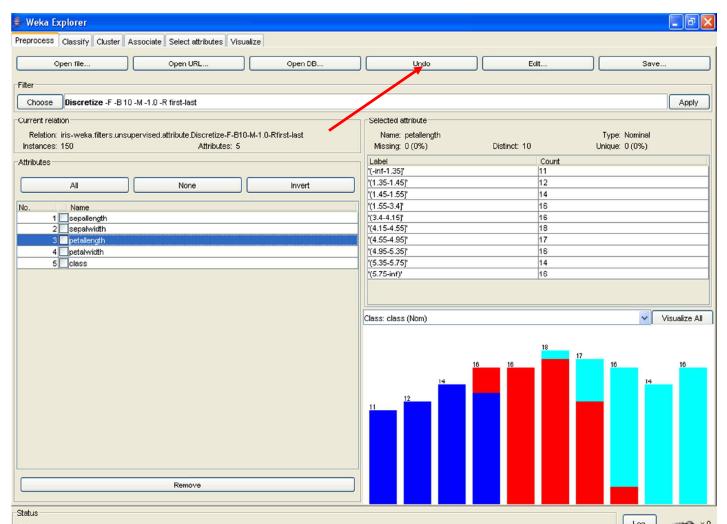
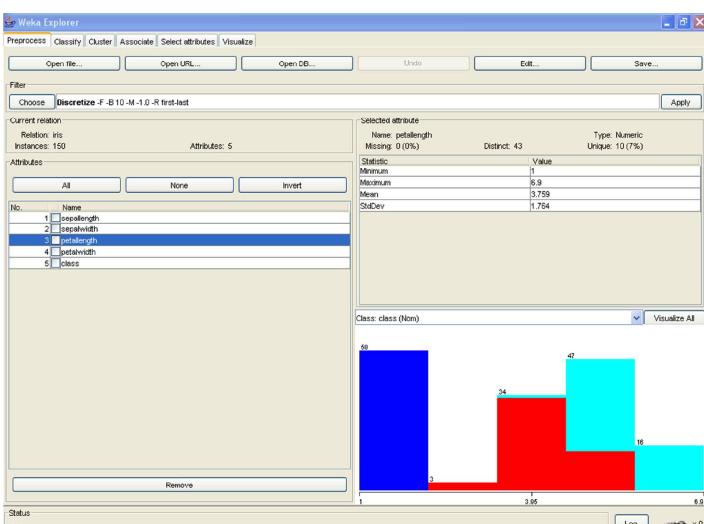
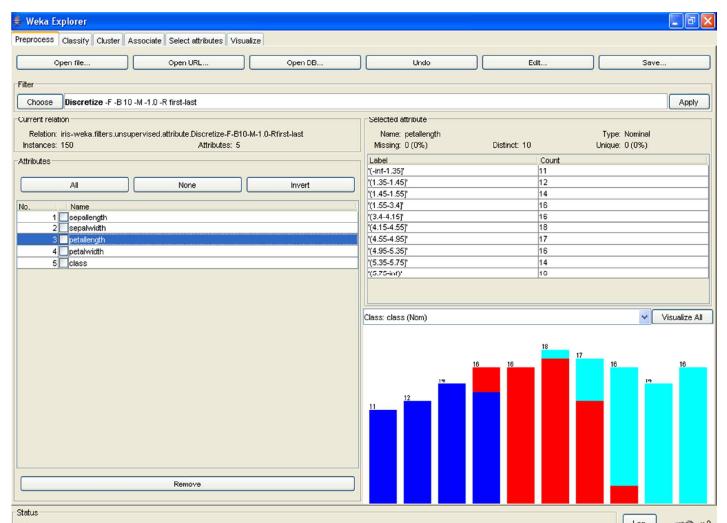
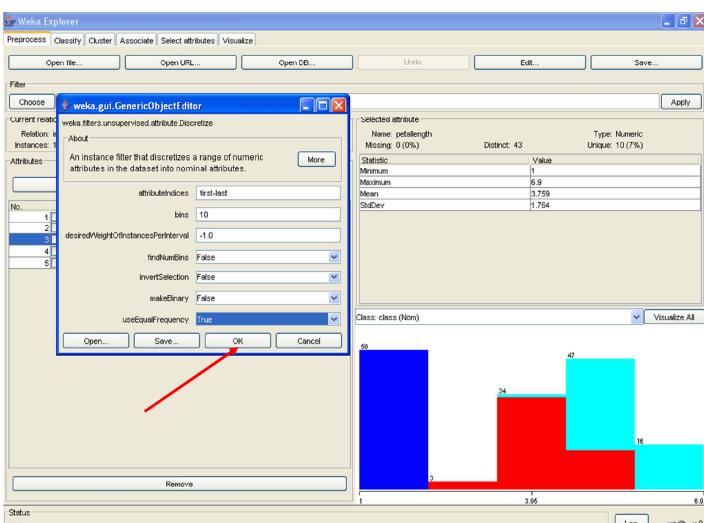
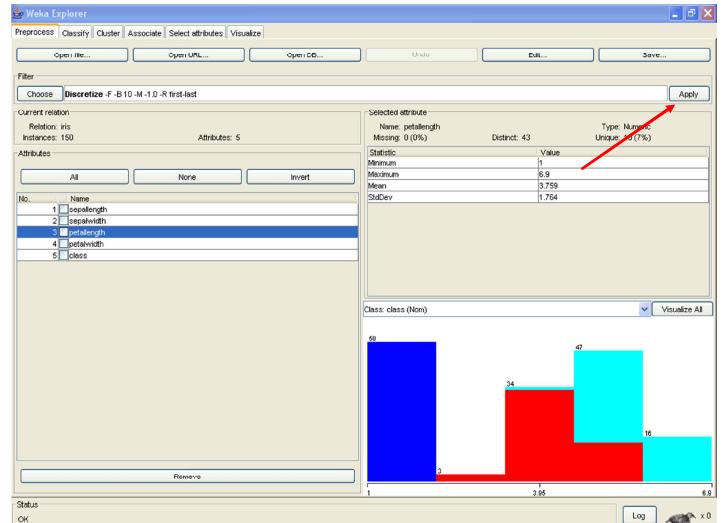
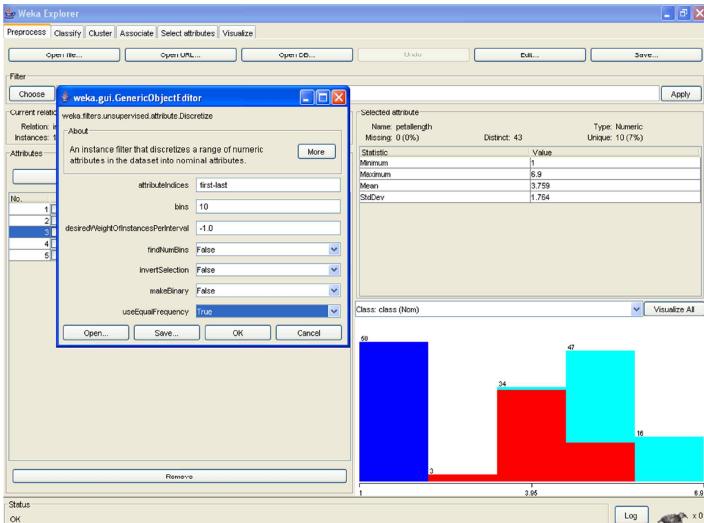
Explorer: pre-processing the data

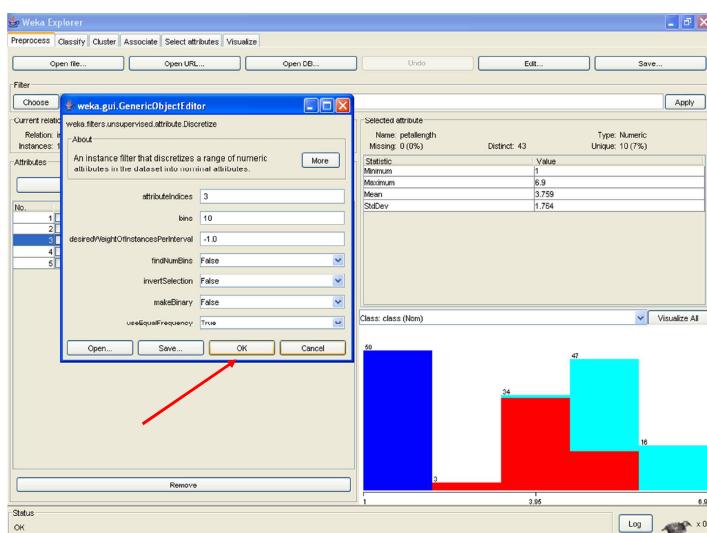
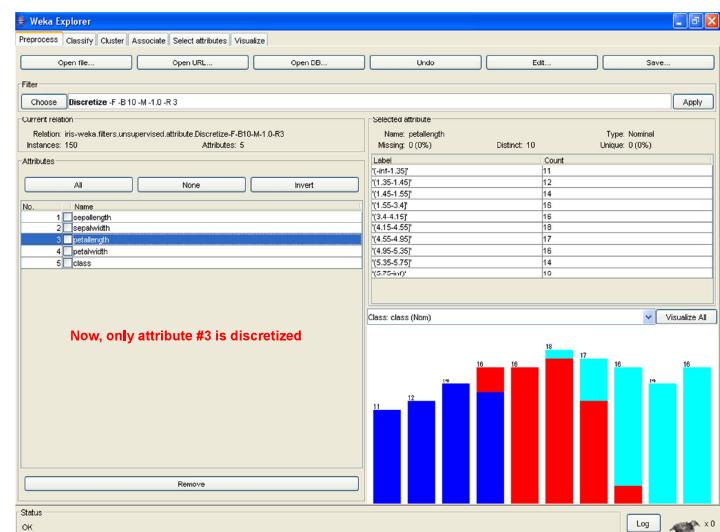
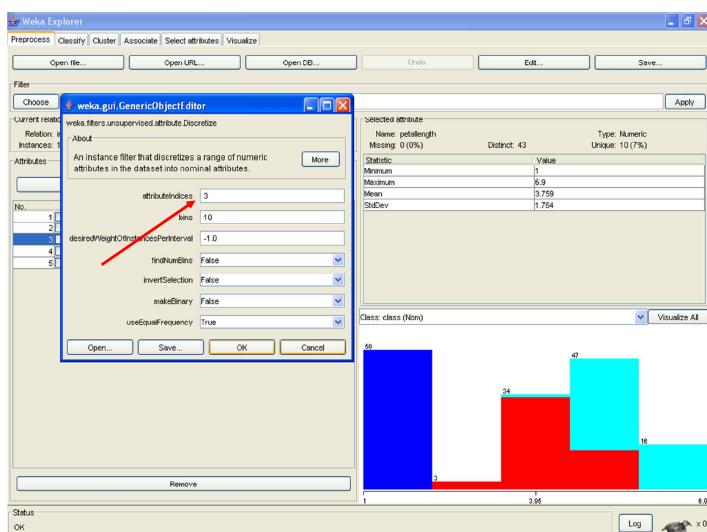
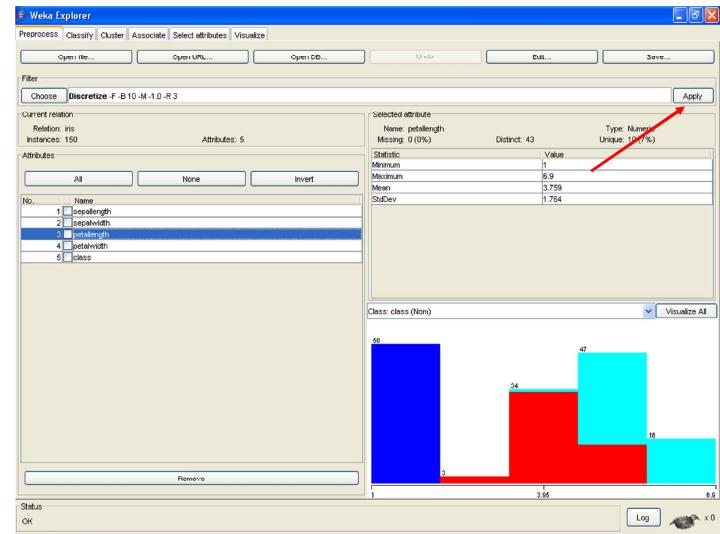
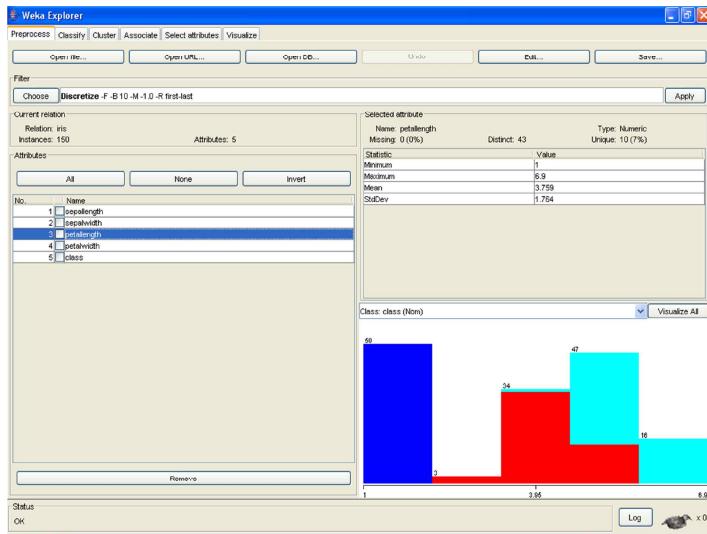
- Data can be imported from a file in various formats: ARFF, CSV, C4.5, binary
- Data can also be read from a URL or from an SQL database (using JDBC)
- Pre-processing tools in WEKA are called “filters”
- WEKA contains filters for:
 - Discretization, normalization, resampling, attribute selection, transforming and combining attributes, ...







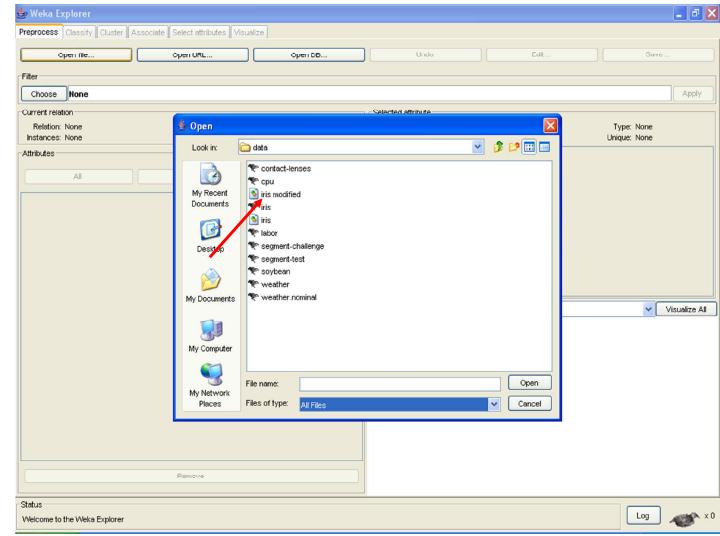




iris - WordPad

```
S.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
5.1,3.6,1.5,0.2,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.3,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.0,3.4,1.6,0.3,Iris-setosa
4.9,3.0,1.4,0.1,Iris-setosa
4.3,3.0,1.3,0.1,Iris-setosa
5.8,4.0,1.2,0.2,Iris-setosa
5.7,4.4,1.4,0.2,Iris-setosa
5.4,4.5,1.5,0.4,Iris-setosa
5.1,3.5,1.4,0.3,Iris-setosa
5.7,3.8,1.7,0.3,Iris-setosa
5.1,3.8,1.5,0.3,Iris-setosa
5.4,3.4,1.5,0.2,Iris-setosa
5.1,3.5,1.5,0.2,Iris-setosa
4.6,3.4,1.4,0.2,Iris-setosa
5.1,3.1,1.7,0.5,Iris-setosa
4.9,3.4,1.9,0.2,Iris-setosa
5.0,3.0,1.6,0.2,Iris-setosa
5.0,3.4,1.6,0.2,Iris-setosa
5.2,3.5,1.5,0.2,Iris-setosa
5.2,3.4,1.4,0.2,Iris-setosa
4.7,3.2,1.6,0.2,Iris-setosa
4.8,3.1,1.6,0.2,Iris-setosa
5.4,3.4,1.5,0.4,Iris-setosa
5.1,3.5,1.4,0.2,Iris-setosa
5.5,4.2,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
5.0,3.2,1.2,0.2,Iris-setosa
5.5,3.5,1.5,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
4.6,3.4,1.5,0.2,Iris-setosa
5.1,3.4,1.5,0.2,Iris-setosa
5.1,3.4,1.5,0.2,Iris-setosa
For Help, press F1
```

iris.data

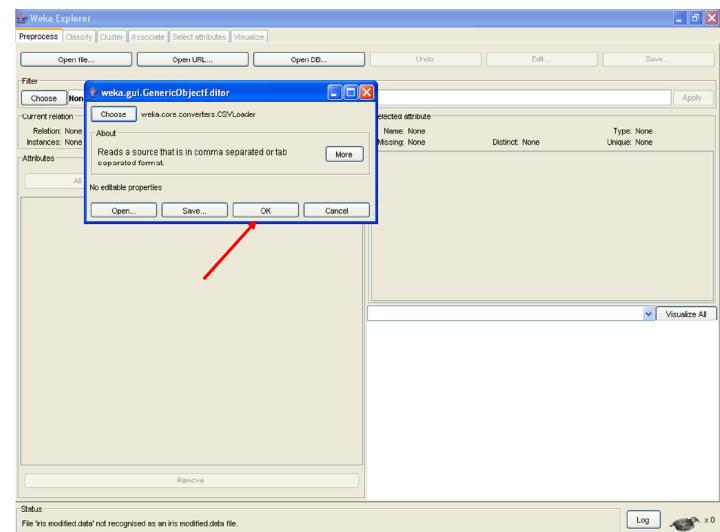
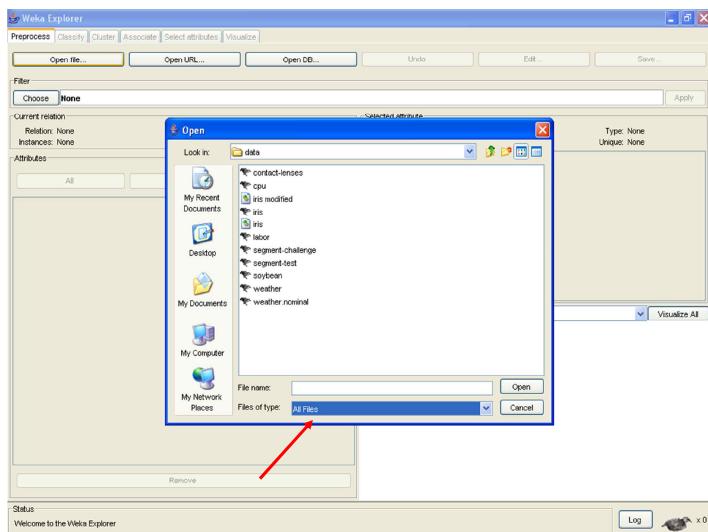
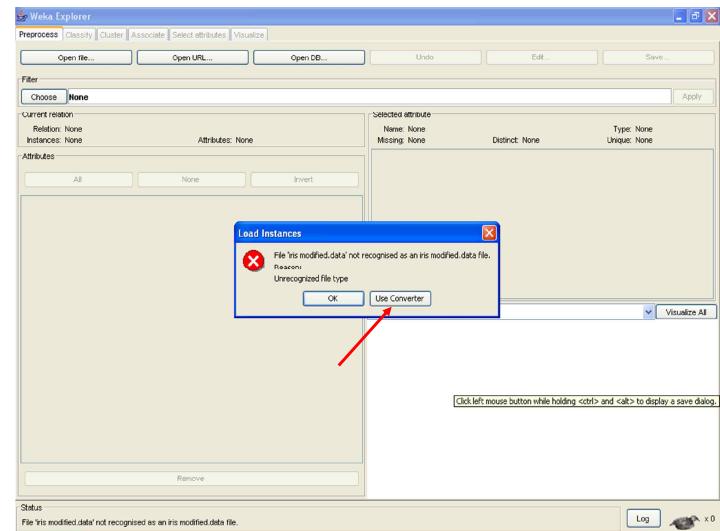


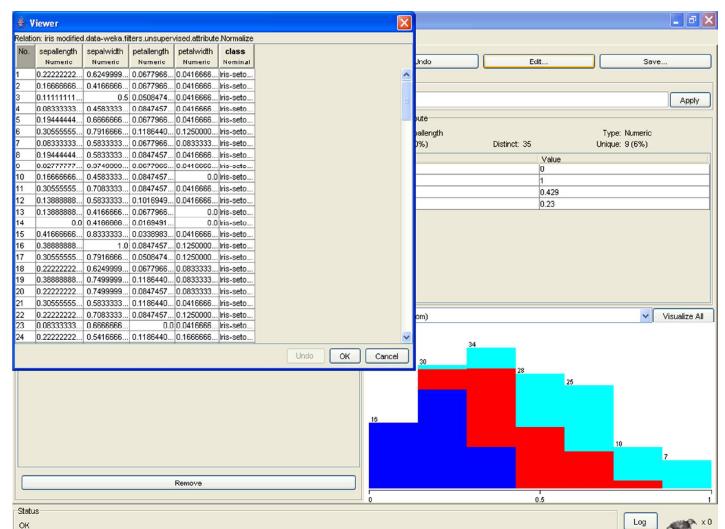
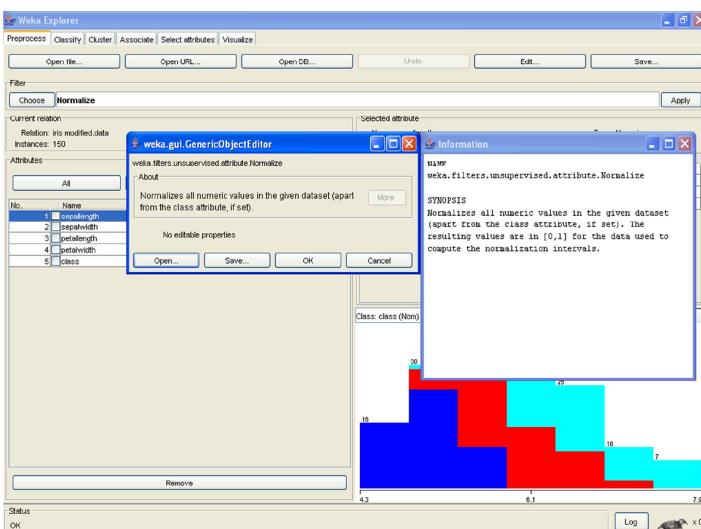
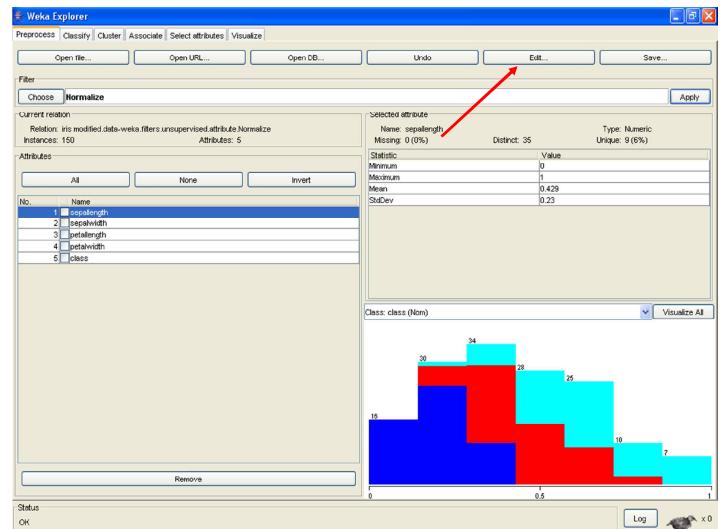
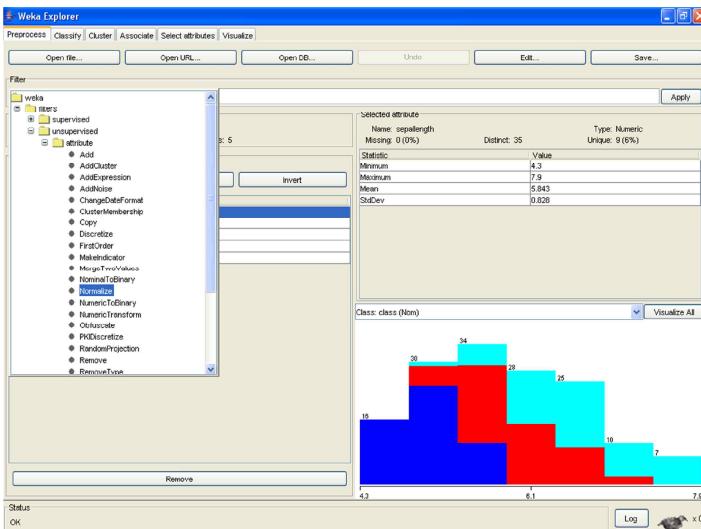
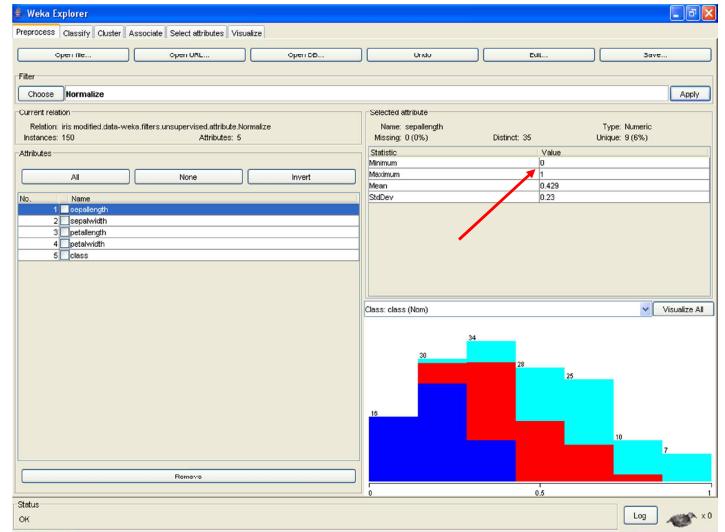
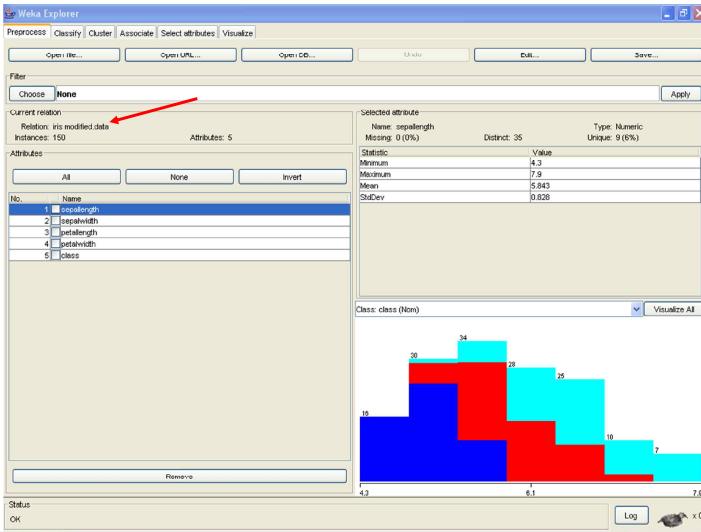
iris modified - WordPad

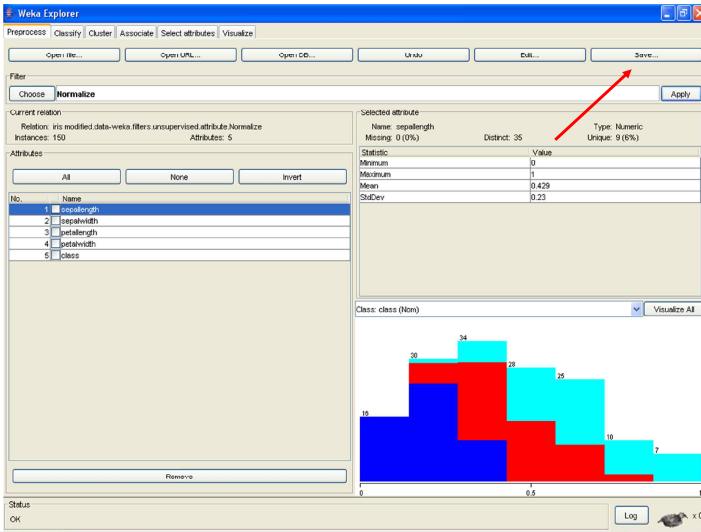
```
sepallength,sepalwidth,petallength,petalwidth,class
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.1,3.6,1.5,0.2,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.3,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
5.2,3.5,1.5,0.2,Iris-setosa
5.2,3.4,1.4,0.2,Iris-setosa
4.7,3.2,1.6,0.2,Iris-setosa
4.8,3.1,1.6,0.2,Iris-setosa
5.4,3.4,1.5,0.4,Iris-setosa
5.1,3.5,1.5,0.2,Iris-setosa
5.7,3.8,1.7,0.3,Iris-setosa
5.1,3.8,1.5,0.3,Iris-setosa
5.4,3.4,1.5,0.2,Iris-setosa
5.1,3.5,1.5,0.2,Iris-setosa
4.6,3.4,1.4,0.2,Iris-setosa
5.1,3.1,1.7,0.5,Iris-setosa
4.9,3.4,1.9,0.2,Iris-setosa
5.0,3.0,1.6,0.2,Iris-setosa
5.0,3.4,1.6,0.2,Iris-setosa
5.2,3.5,1.5,0.2,Iris-setosa
5.2,3.4,1.4,0.2,Iris-setosa
4.6,3.6,1.0,0.2,Iris-setosa
4.9,3.1,1.7,0.5,Iris-setosa
4.5,3.0,1.2,0.2,Iris-setosa
5.0,3.0,1.6,0.2,Iris-setosa
5.0,3.4,1.6,0.4,Iris-setosa
5.2,3.5,1.5,0.2,Iris-setosa
5.2,3.4,1.4,0.2,Iris-setosa
4.7,3.2,1.6,0.2,Iris-setosa
4.9,3.1,1.6,0.2,Iris-setosa
5.5,4.2,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
5.0,3.0,1.3,0.2,Iris-setosa
4.4,3.0,1.3,0.2,Iris-setosa
For Help, press F1
```

first row

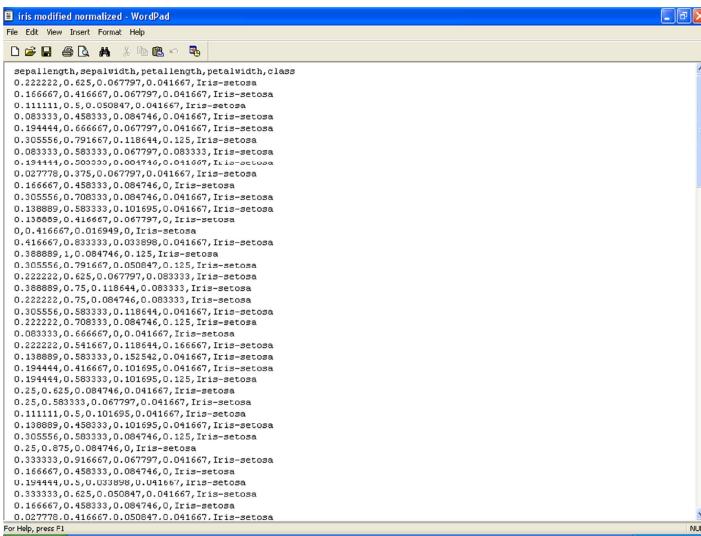
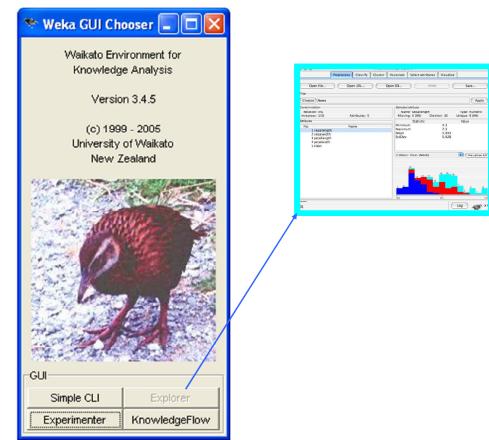
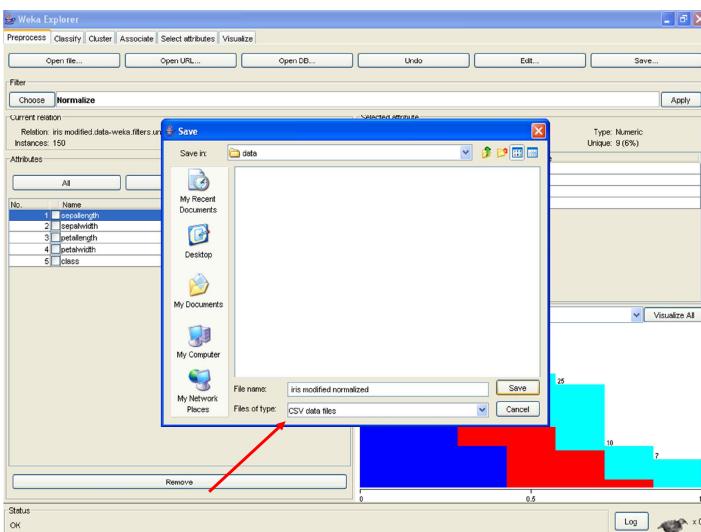
iris modified.data





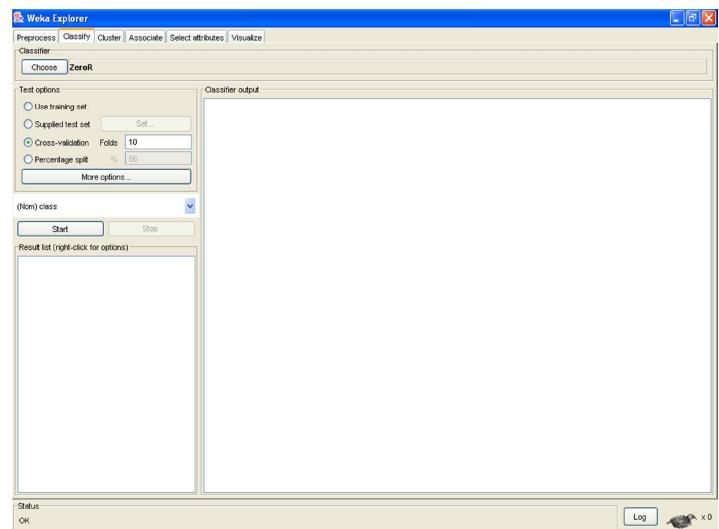
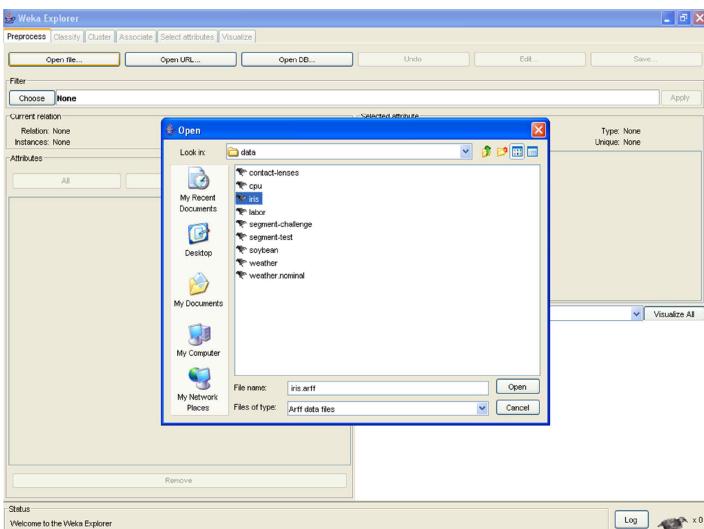
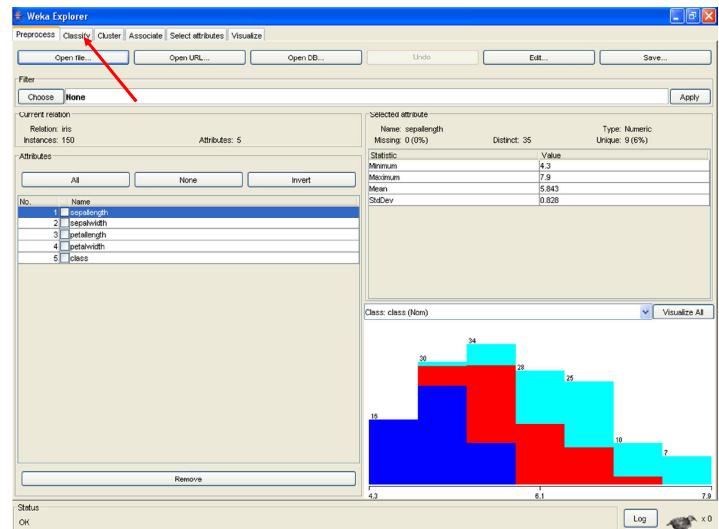
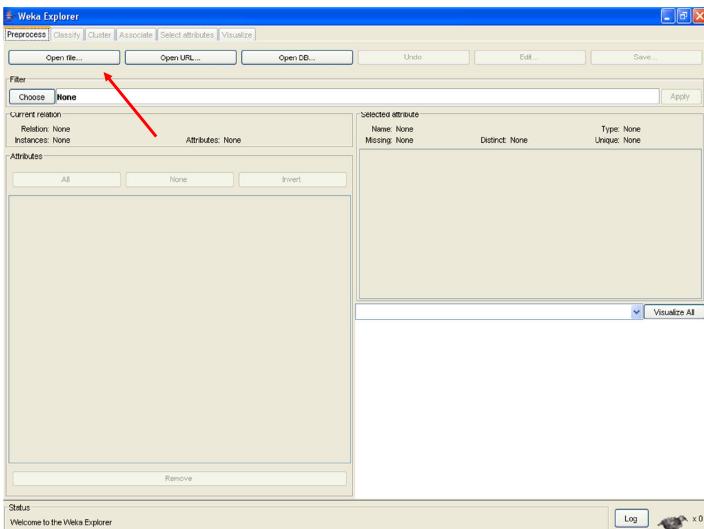
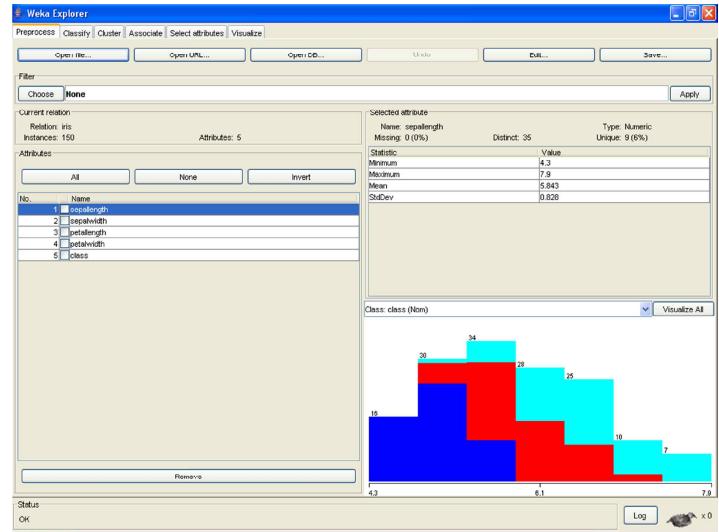
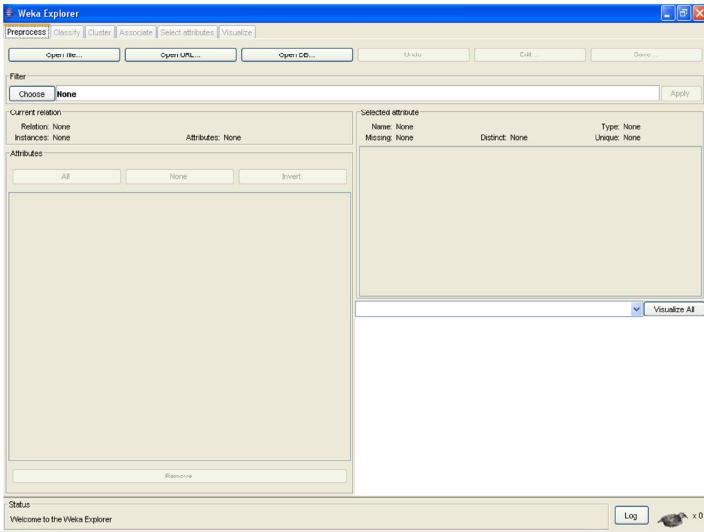


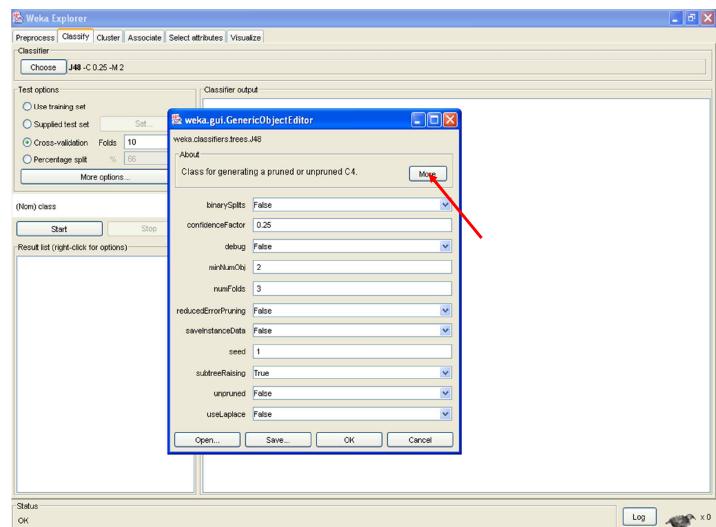
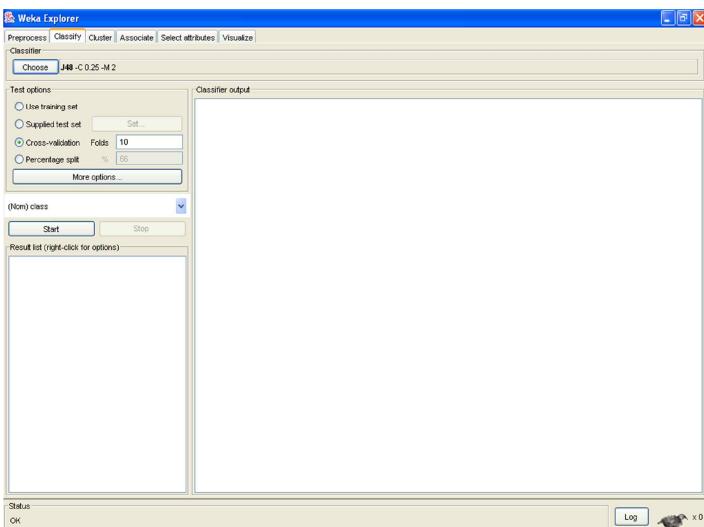
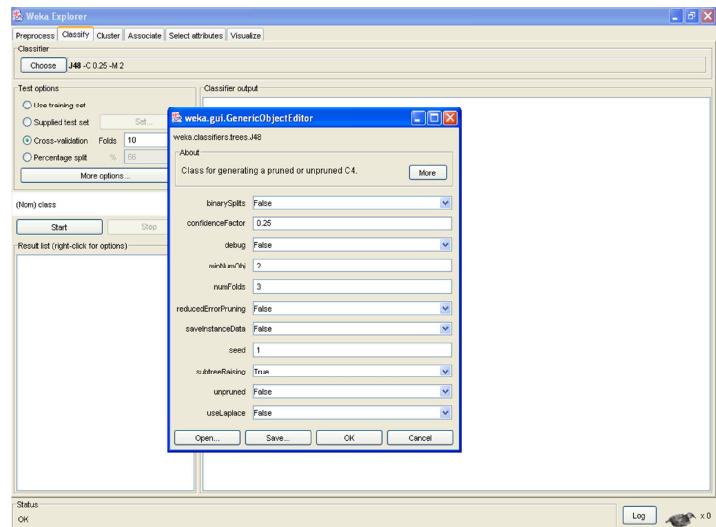
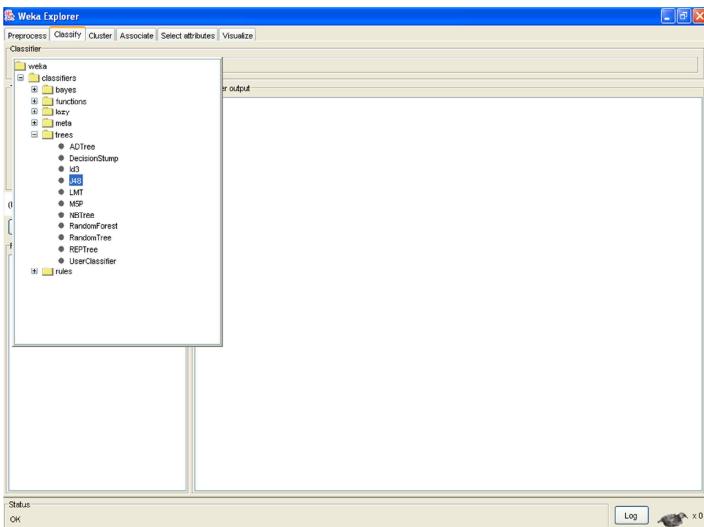
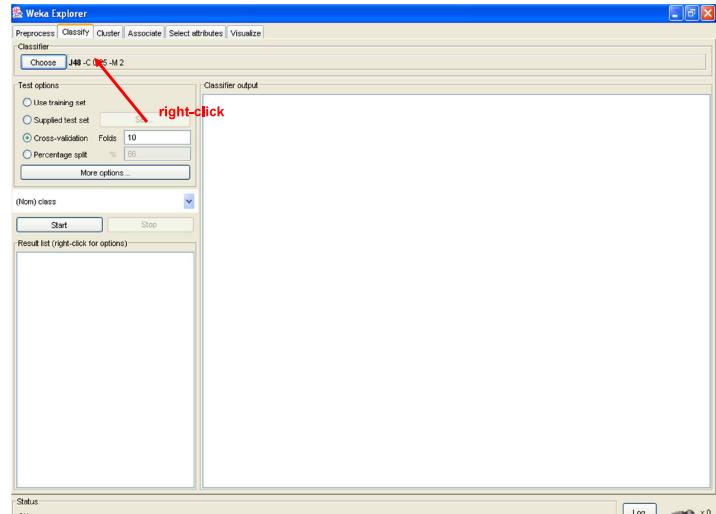
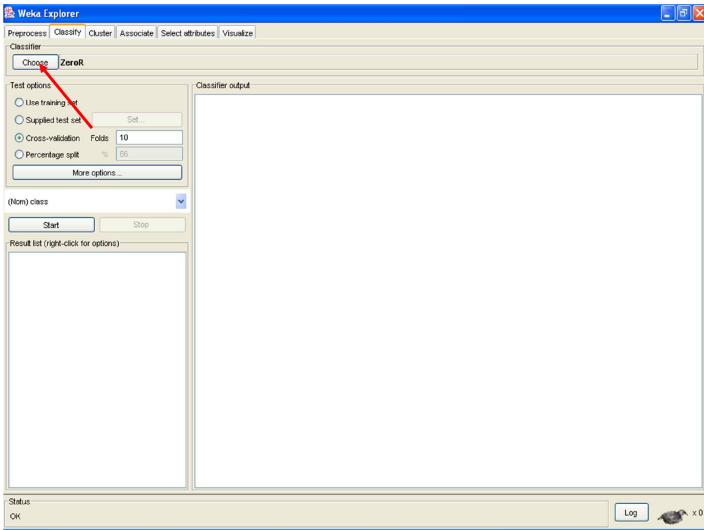
Data Mining with WEKA (Classification – Decision Tree)

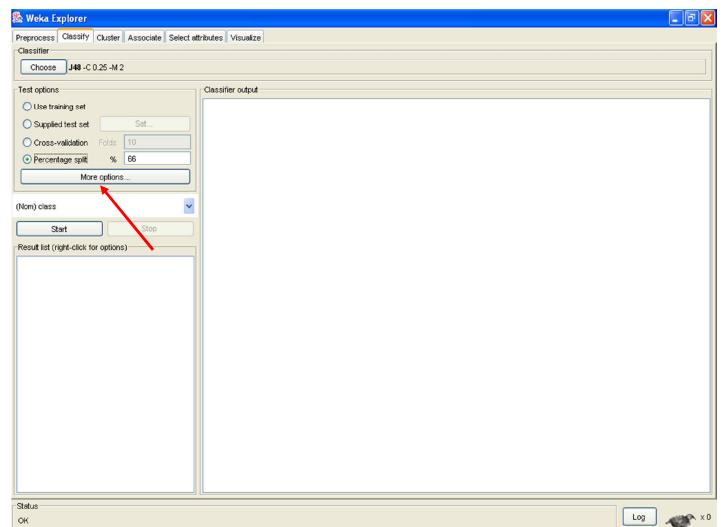
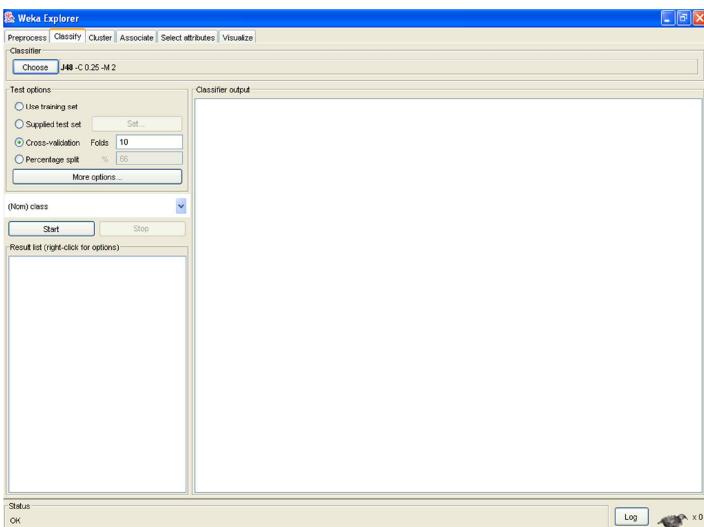
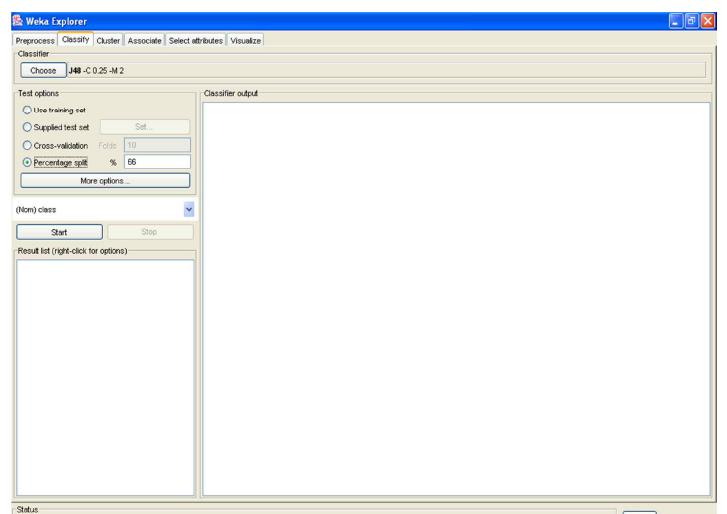
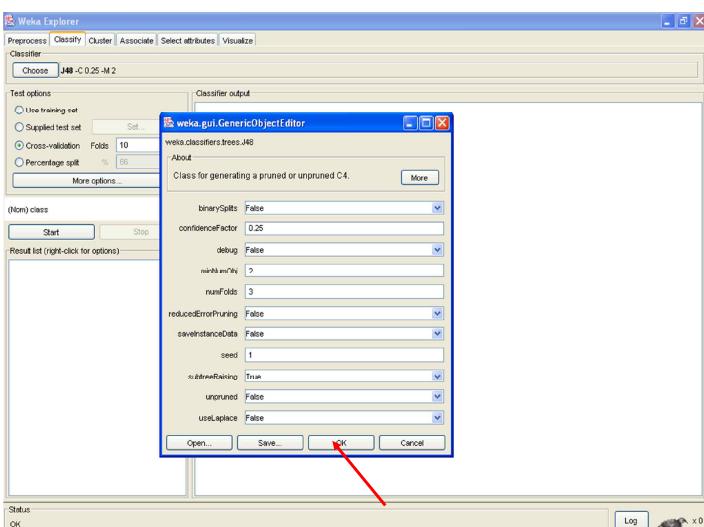
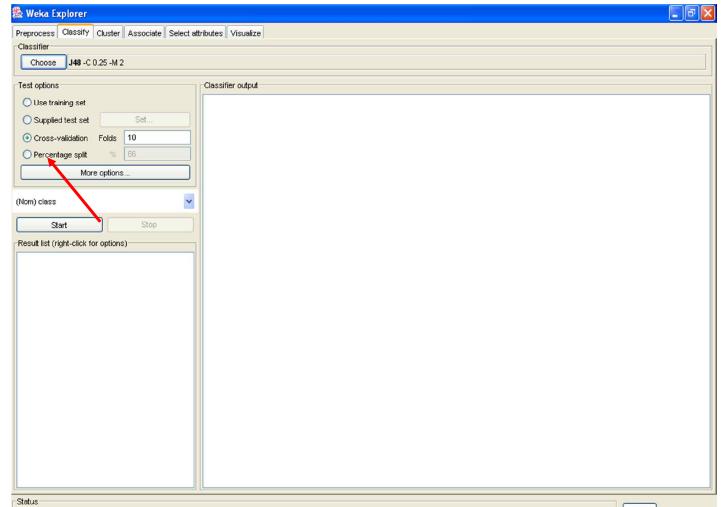
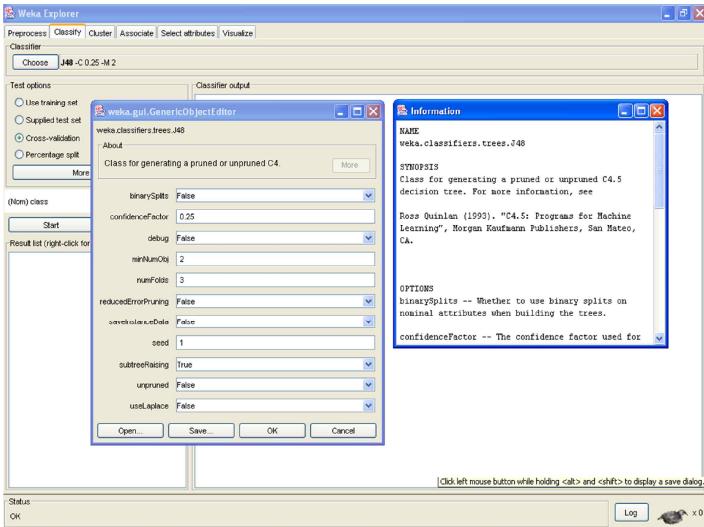


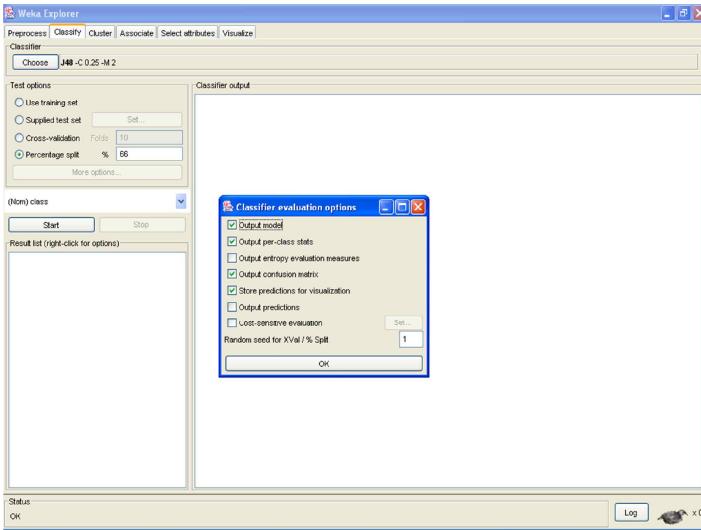
WEKA Explorer: building “classifiers”

- Classifiers in WEKA are models for classifying nominal or numeric data
- Implemented learning schemes include:
 - Decision trees, naïve Bayes, instance-based classifiers, support vector machines, multi-layer neural networks, logistic regression, Bayes’ nets, ...
- “Meta”-classifiers include:
 - Bagging, boosting, stacking, error-correcting output codes, locally weighted learning, ...







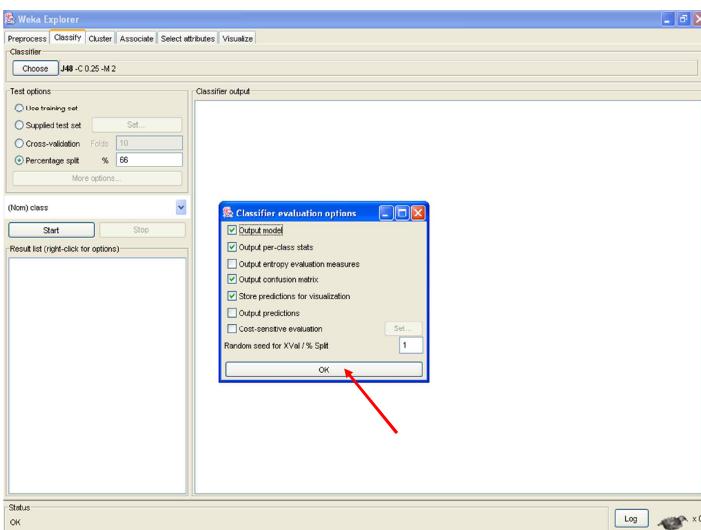


```

Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Classifier Choose J48-C 0.25-M 2
Test options
 Use training set
 Supplied test set Set...
 Cross-validation Folds 10
 Percentage split % 66
More options...
(Nom) class Start Stop
Result list (right-click for options)
Status OK Log x 0

Classifier output
--- Run information ---
Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: iris
Instances: 150
Attributes: 5
sepalwidth
sepalwidth
petallength
petalwidth
class
Test mode: split 66% train, remainder test
--- Classifier model (full training set) ---
J48 pruned tree
-----
petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
| petallength <= 1.7
| | petallength <= 4.9: Iris-versicolor (48.0/1.0)
| | | petallength > 4.9
| | | petalwidth <= 1.5: Iris-virginica (3.0)
| | | petalwidth > 1.5: Iris-versicolor (3.0/1.0)
| | petalwidth > 1.7: Iris-virginica (46.0/1.0)
Number of Leaves : 5
Size of the tree : 9
Status OK Log x 0

```

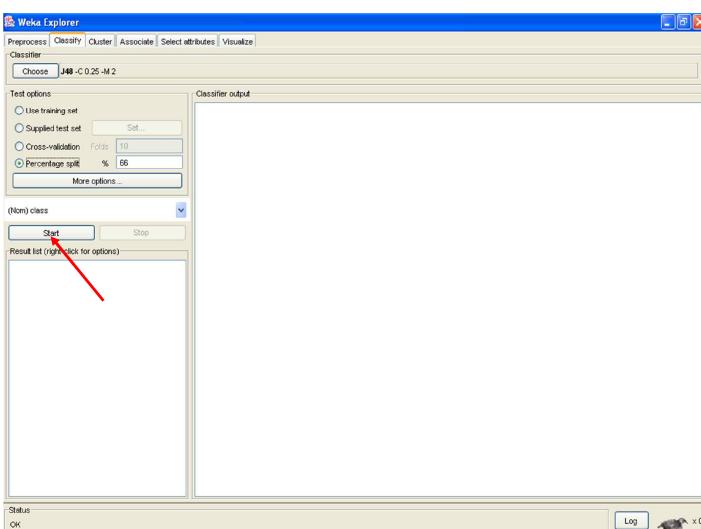


```

Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Classifier Choose J48-C 0.25-M 2
Test options
 Use training set
 Supplied test set Set...
 Cross-validation Folds 10
 Percentage split % 66
More options...
(Nom) class Start Stop
Result list (right-click for options)
Status OK Log x 0

Classifier output
--- Run information ---
Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: iris
Instances: 150
Attributes: 5
sepalwidth
sepalwidth
petallength
petalwidth
class
Test mode: split 66% train, remainder test
--- Classifier model (full training set) ---
J48 pruned tree
-----
petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
| petallength <= 1.7
| | petallength <= 4.9: Iris-versicolor (48.0/1.0)
| | | petallength > 4.9
| | | petalwidth <= 1.5: Iris-virginica (3.0)
| | | petalwidth > 1.5: Iris-versicolor (3.0/1.0)
| | petalwidth > 1.7: Iris-virginica (46.0/1.0)
Number of Leaves : 5
Size of the tree : 9
Status OK Log x 0

```

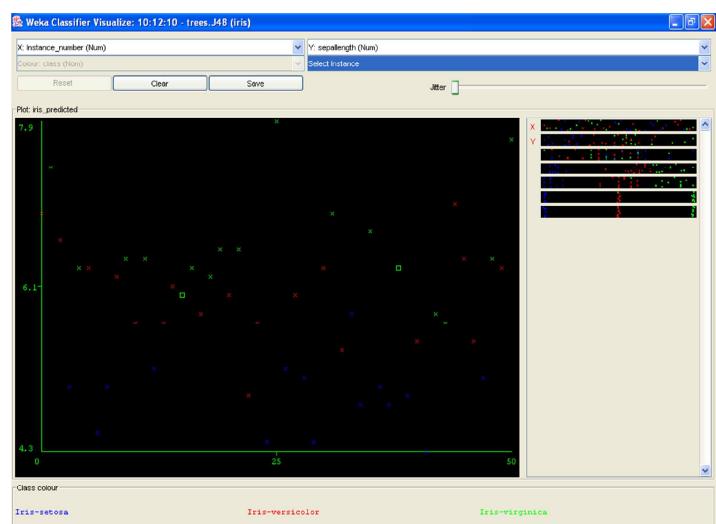
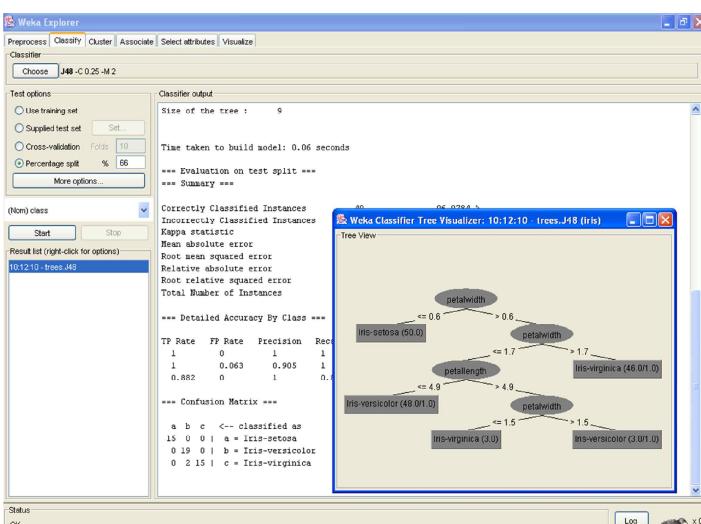
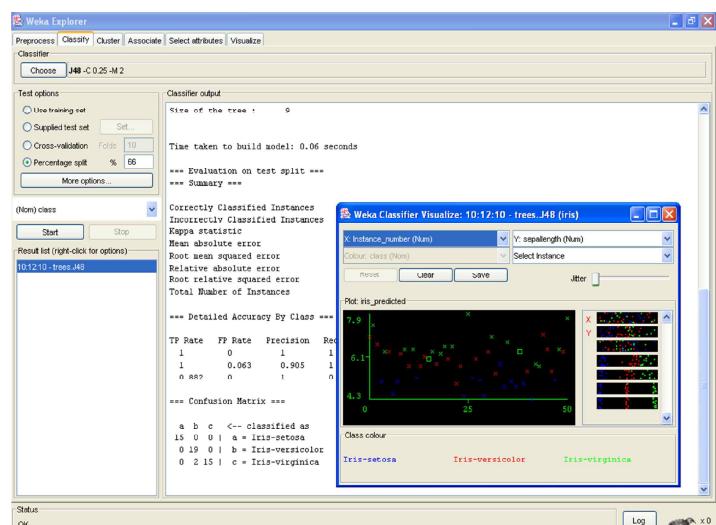
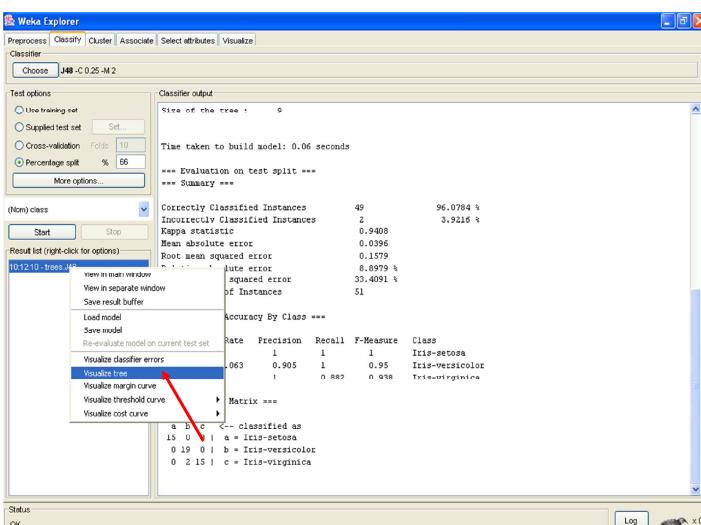
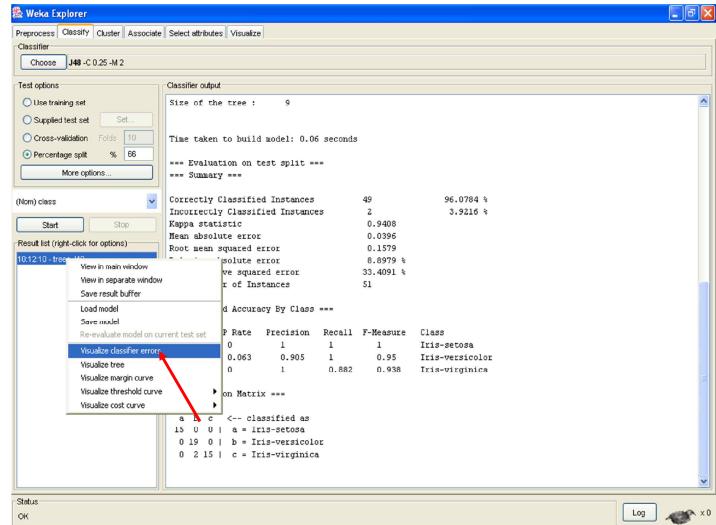
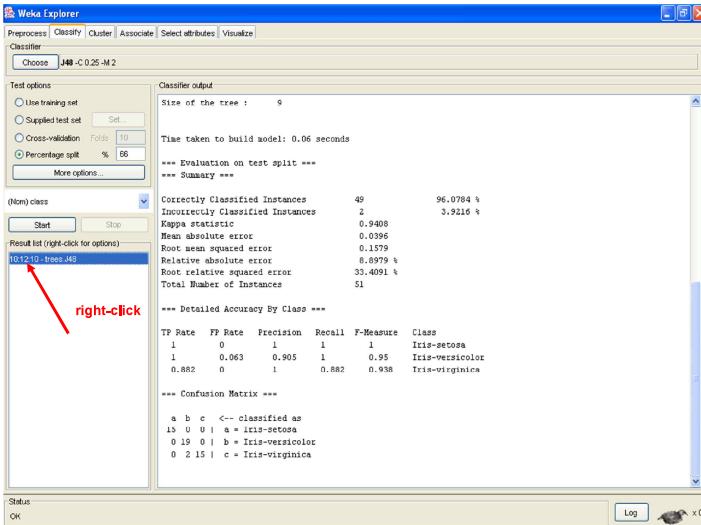


```

Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Classifier Choose J48-C 0.25-M 2
Test options
 Use training set
 Supplied test set Set...
 Cross-validation Folds 10
 Percentage split % 66
More options...
(Nom) class Start Stop
Result list (right-click for options)
Status OK Log x 0

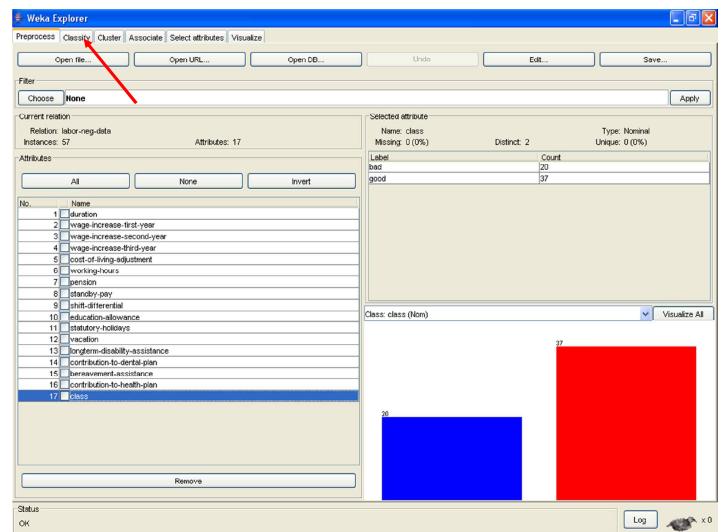
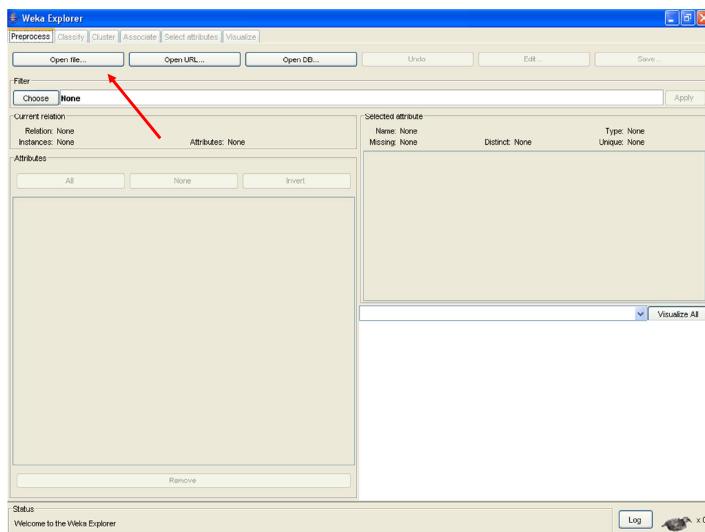
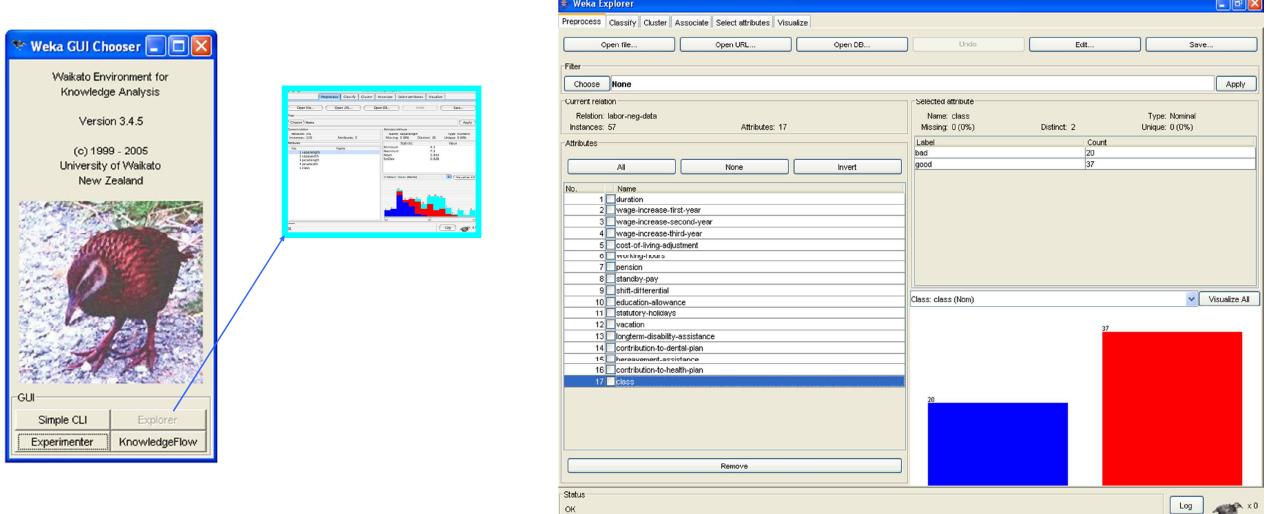
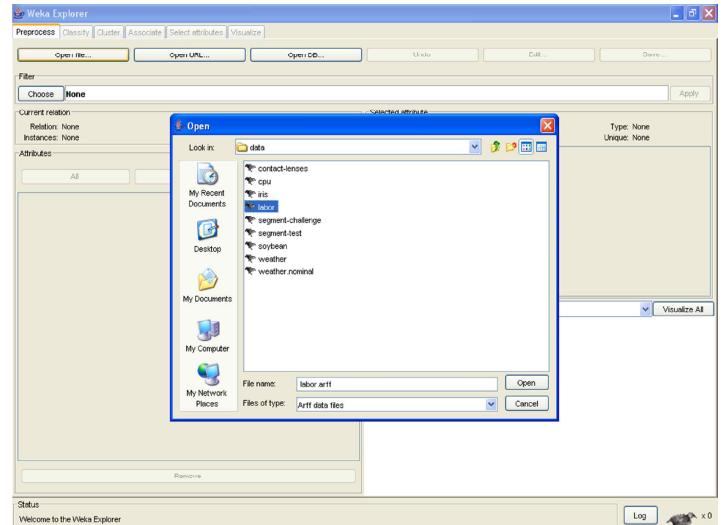
Classifier output
Size of the tree : 9
Time taken to build model: 0.06 seconds
--- Evaluation on test split ---
--- Summary ---
Correctly Classified Instances 49 96.0784 %
Incorrectly Classified Instances 2 3.9216 %
Kappa statistic 0.9408
Mean absolute error 0.0396
Root mean square error 0.1579
Relative absolute error 8.8979 %
Root relative squared error 33.4091 %
Total Number of Instances 51
--- Detailed Accuracy By Class ---
TP Rate FP Rate Precision Recall F-Measure Class
1 0 1 1 1 Iris-setosa
1 0.063 0.905 1 0.95 Iris-versicolor
0.882 0 1 0.882 0.938 Iris-virginica
--- Confusion Matrix ---
a b c <- classified as
15 0 0 | a = Iris-setosa
0 19 0 | b = Iris-versicolor
0 2 15 | c = Iris-virginica
Status OK Log x 0

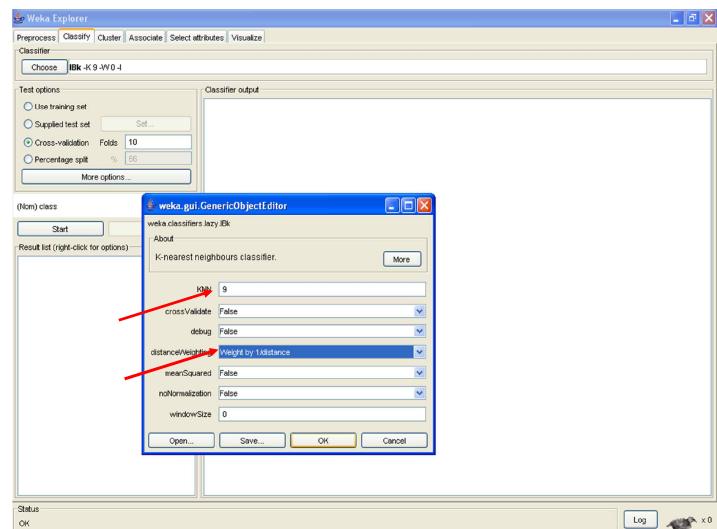
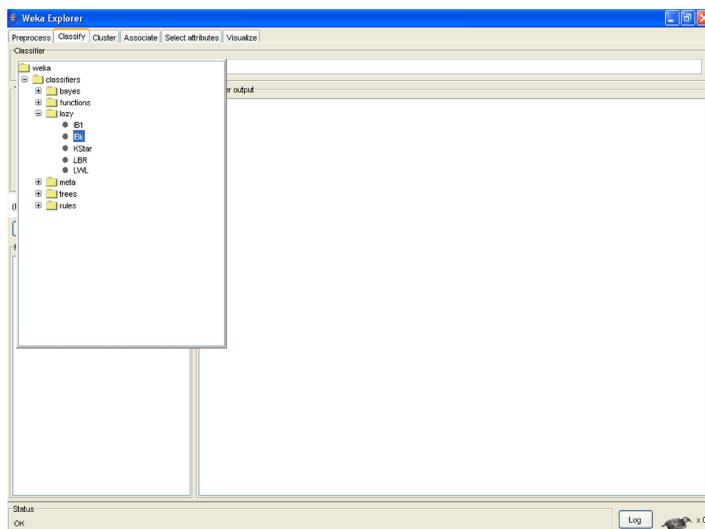
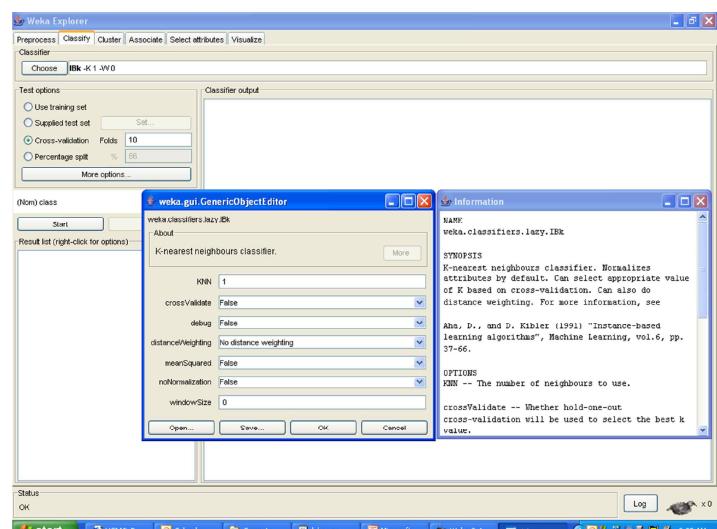
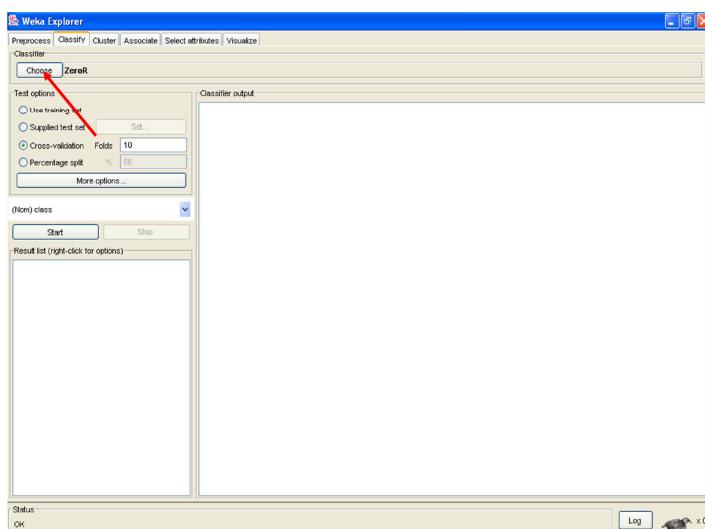
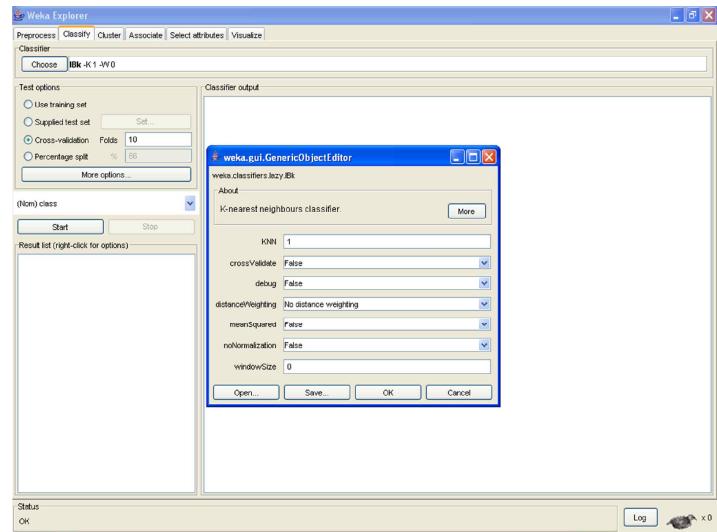
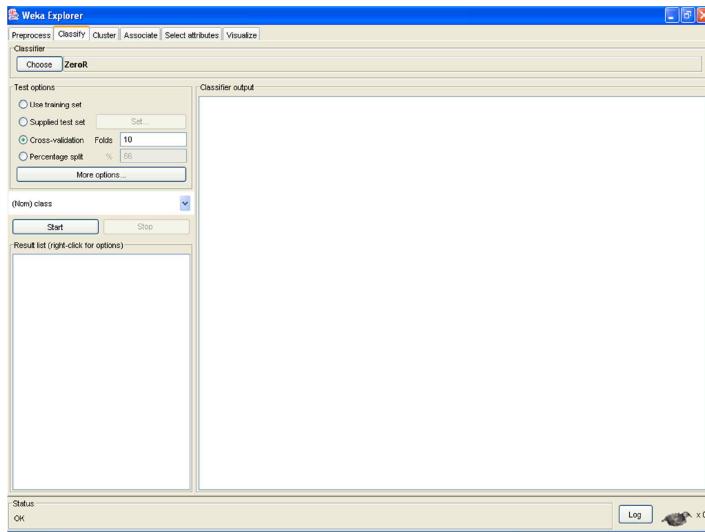
```

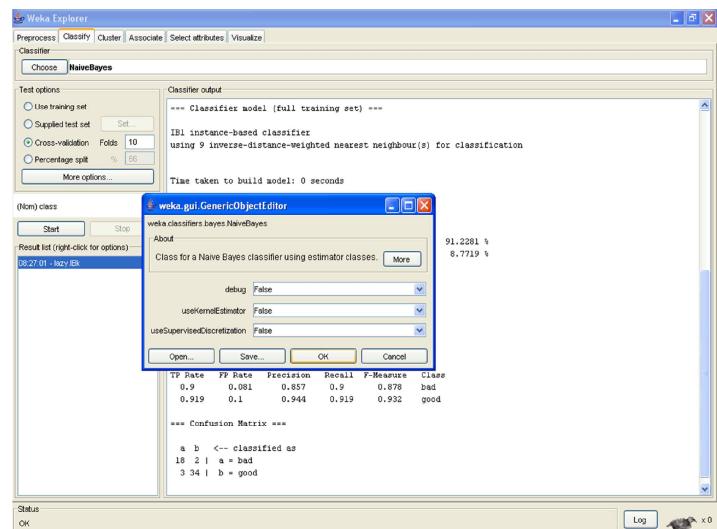
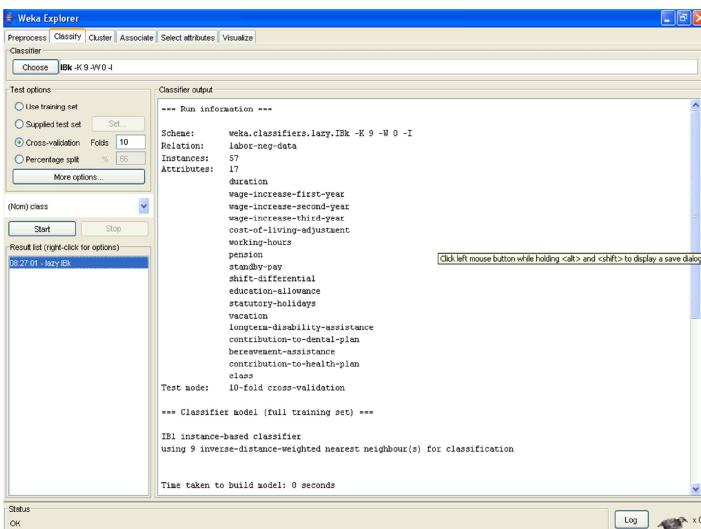
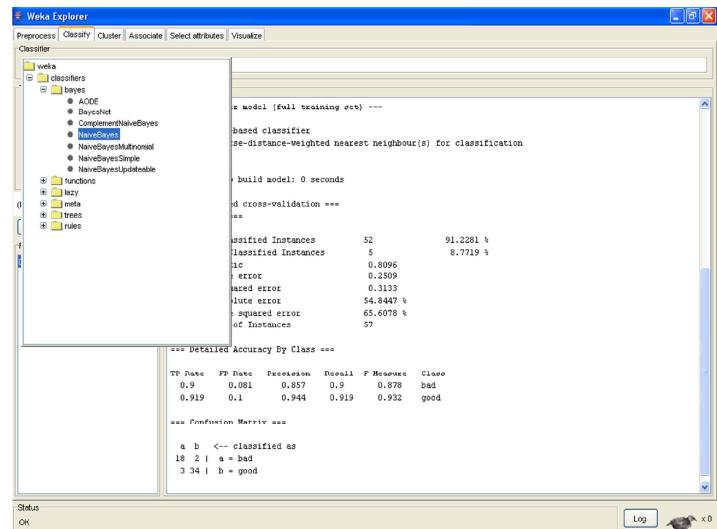
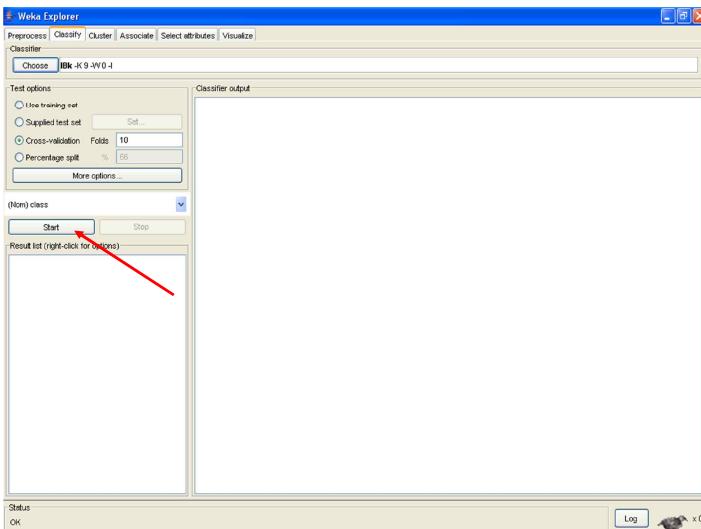
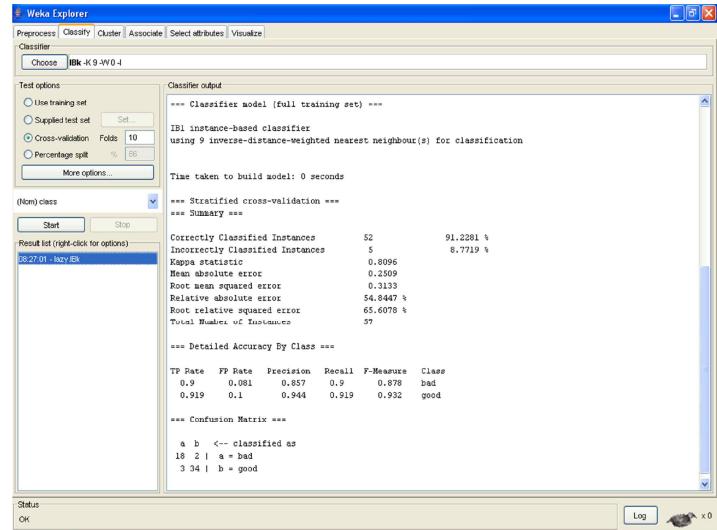
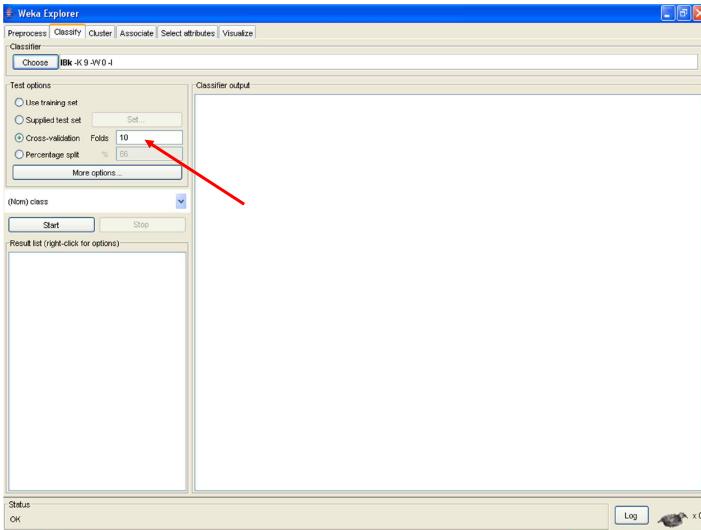


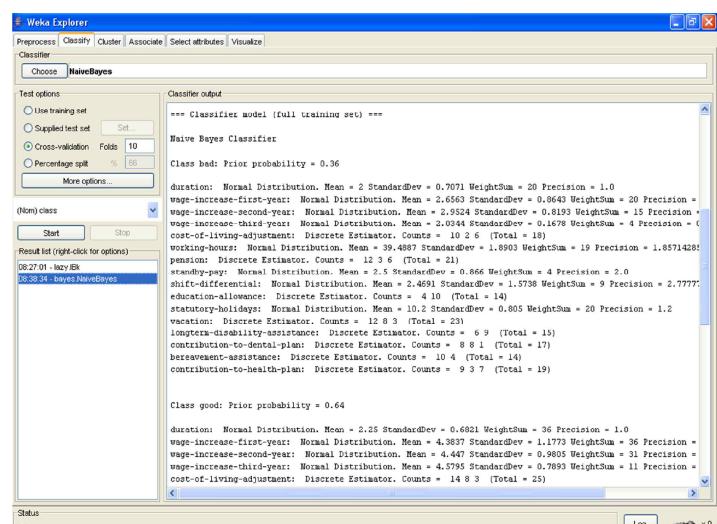
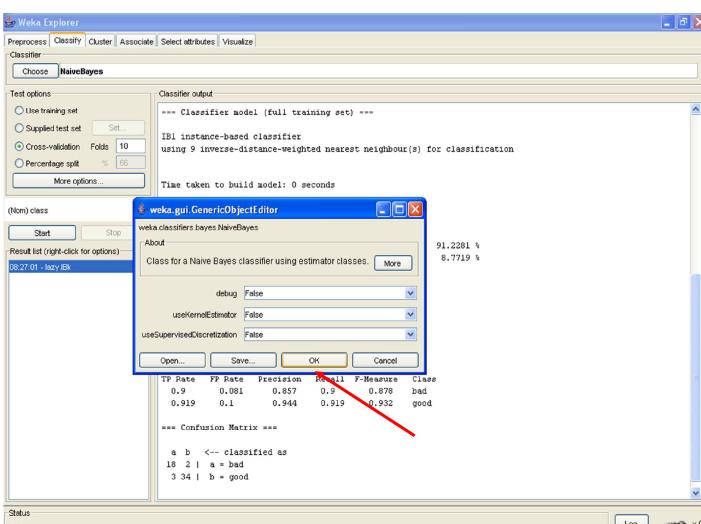
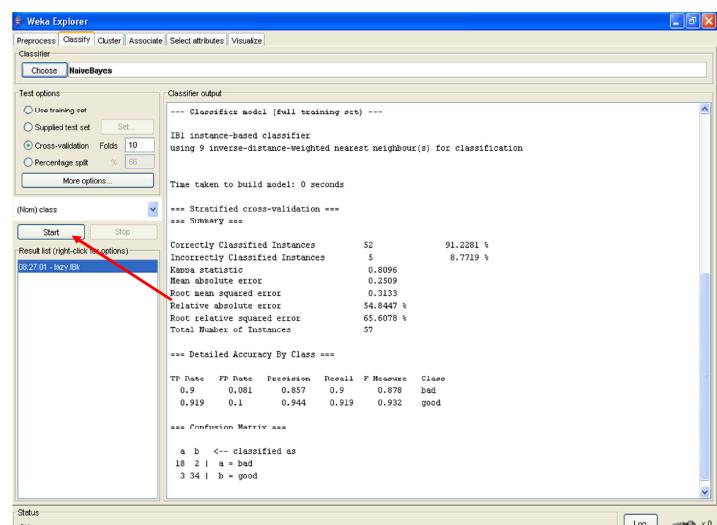
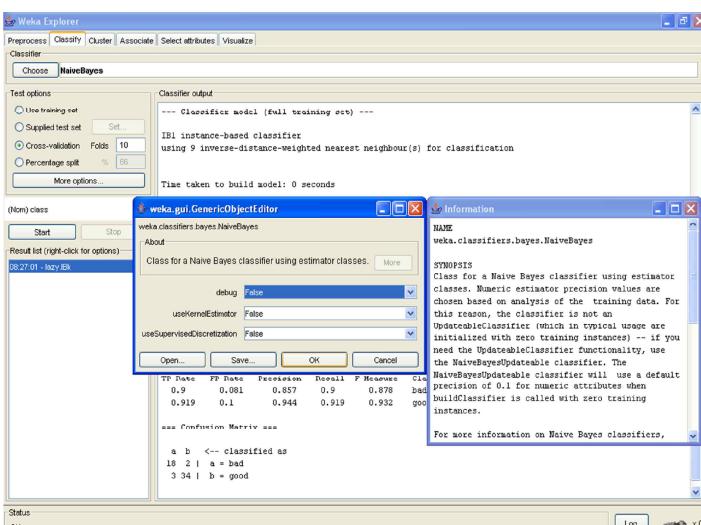
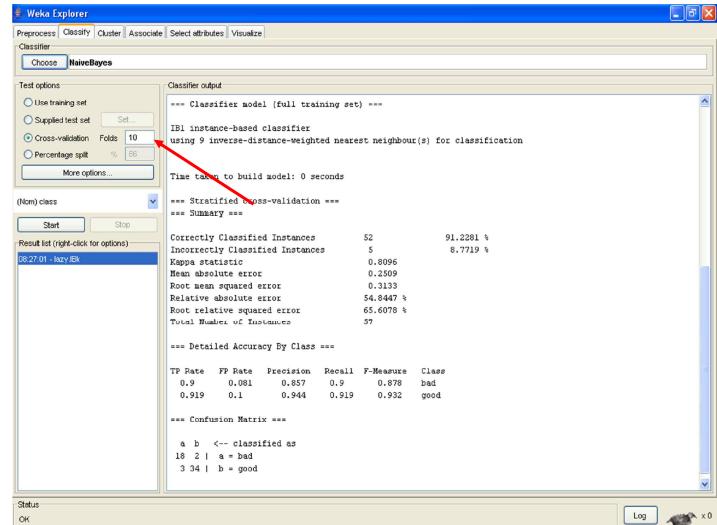
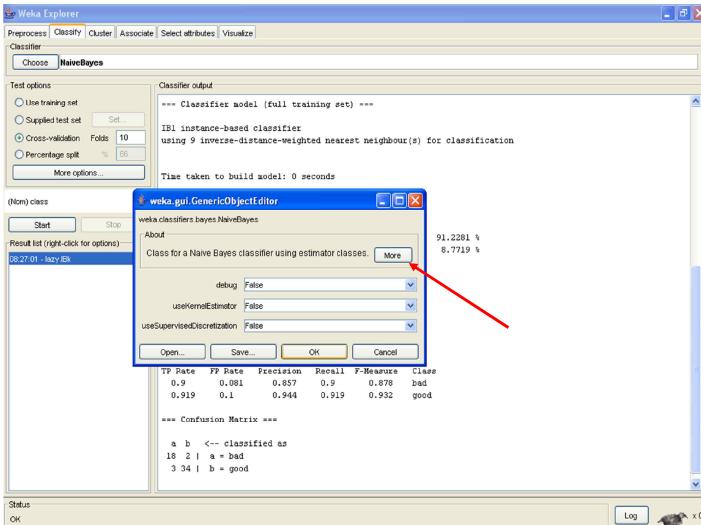
Data Mining with WEKA

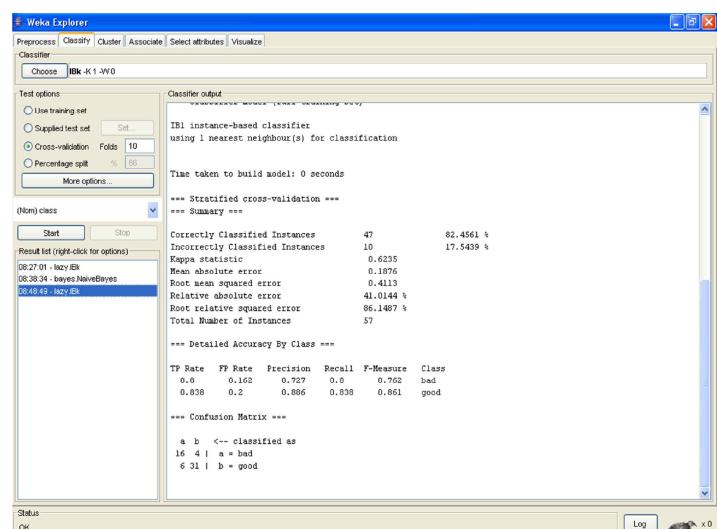
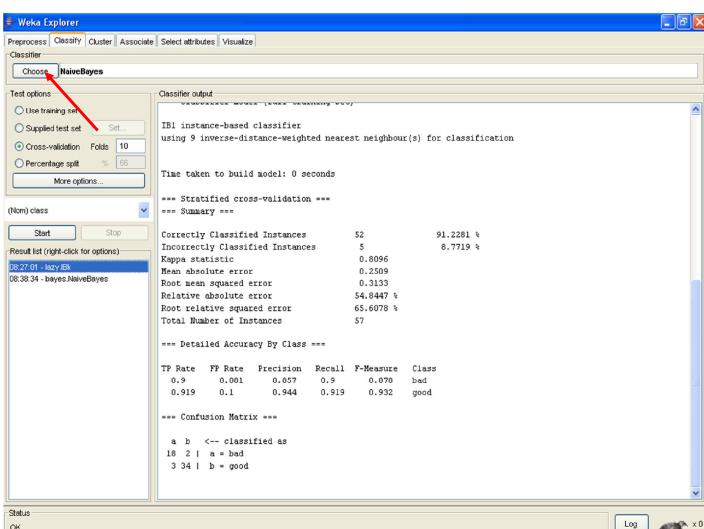
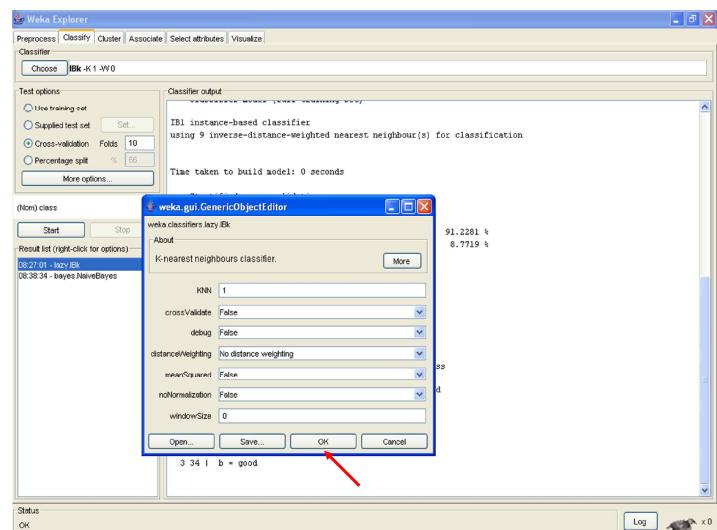
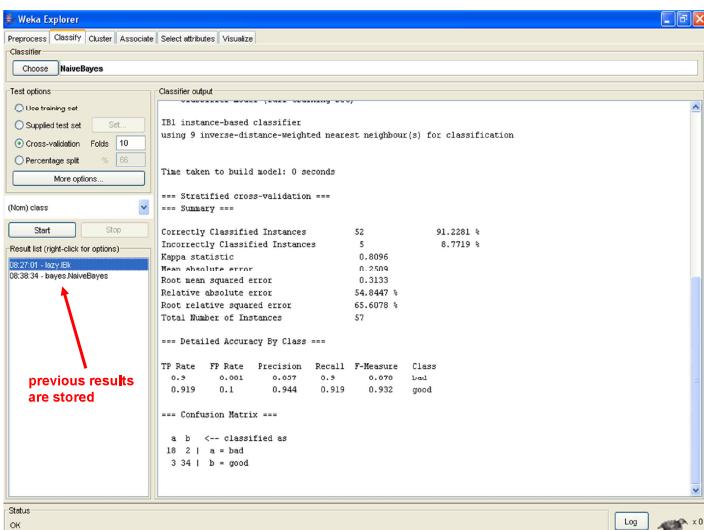
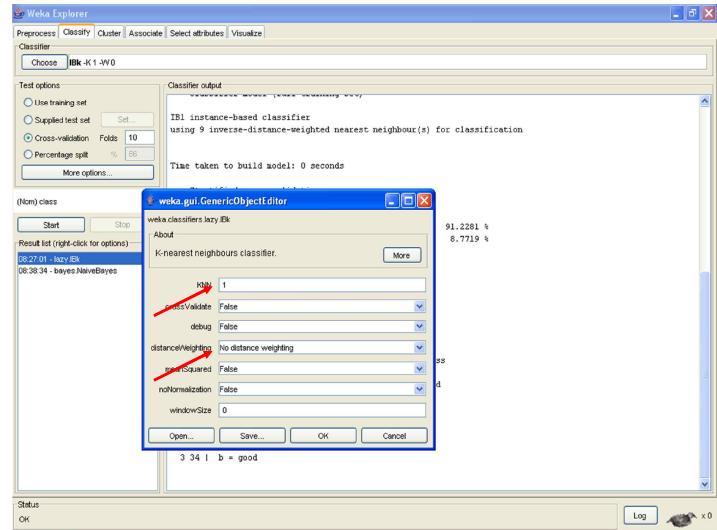
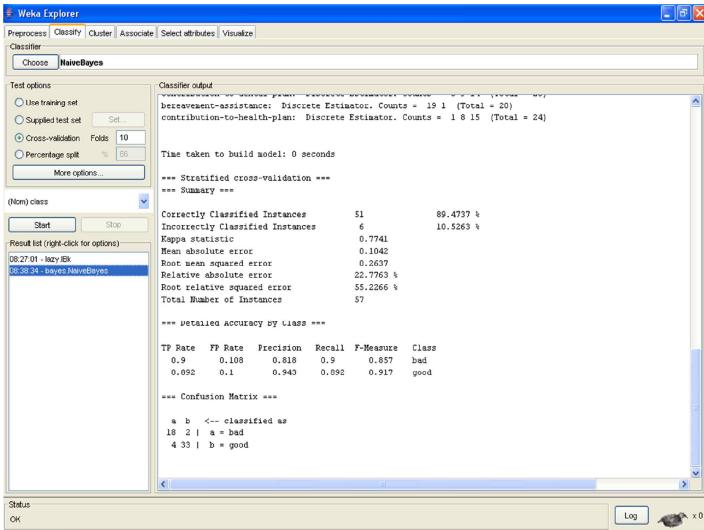
(Classification – k-NN & Naïve Bayes;
Comparing Classifiers)

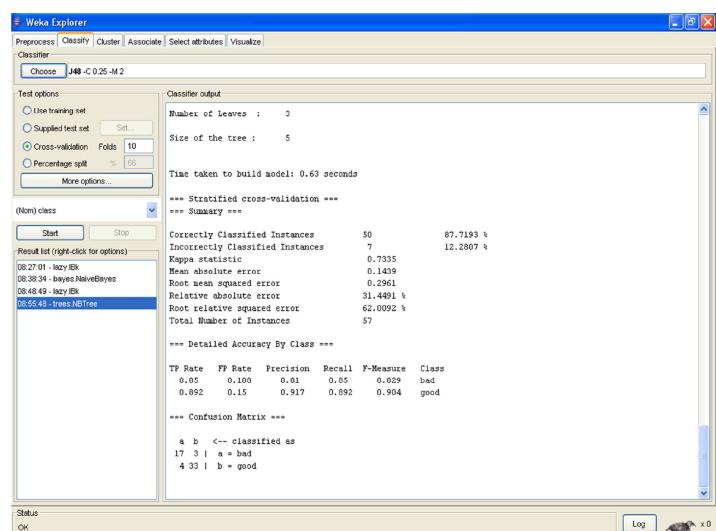
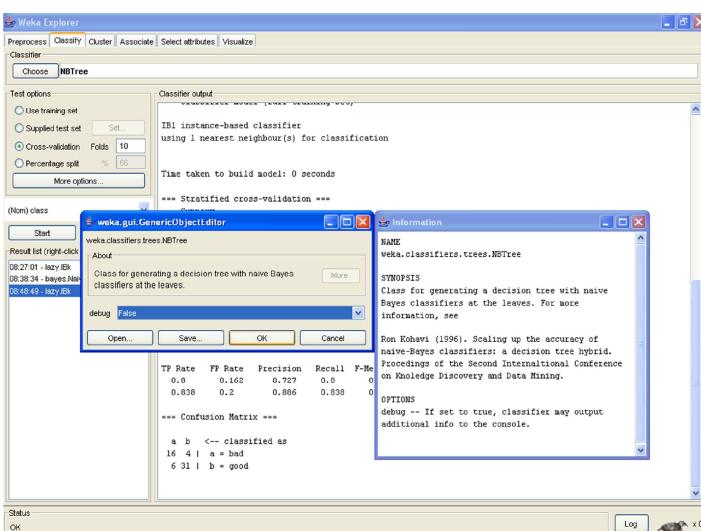
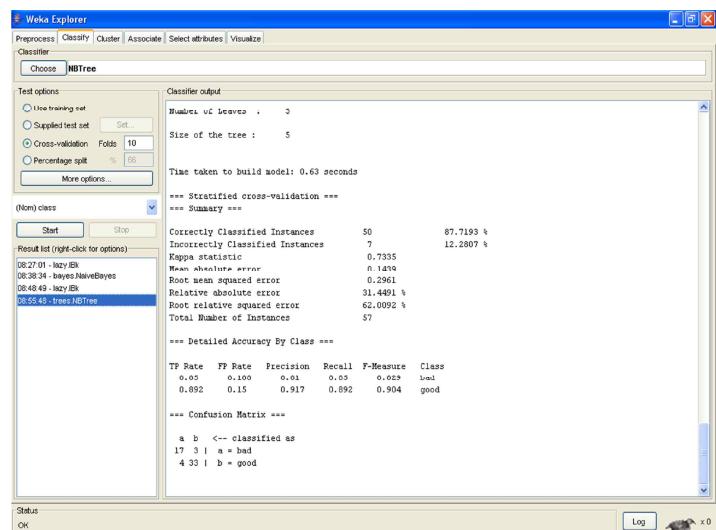
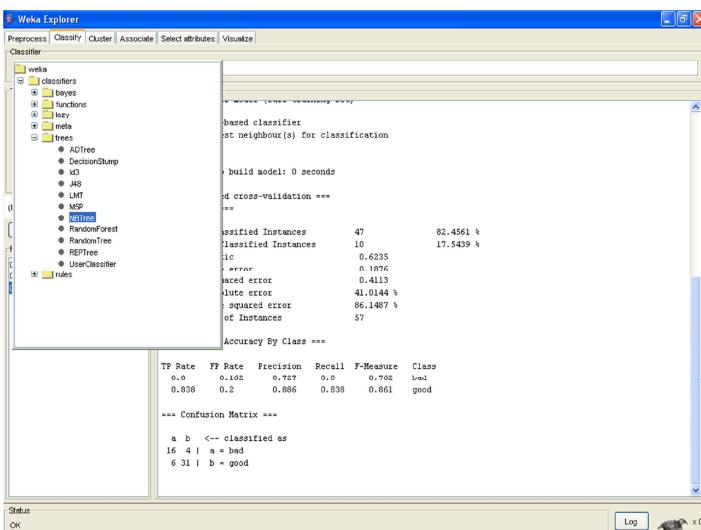
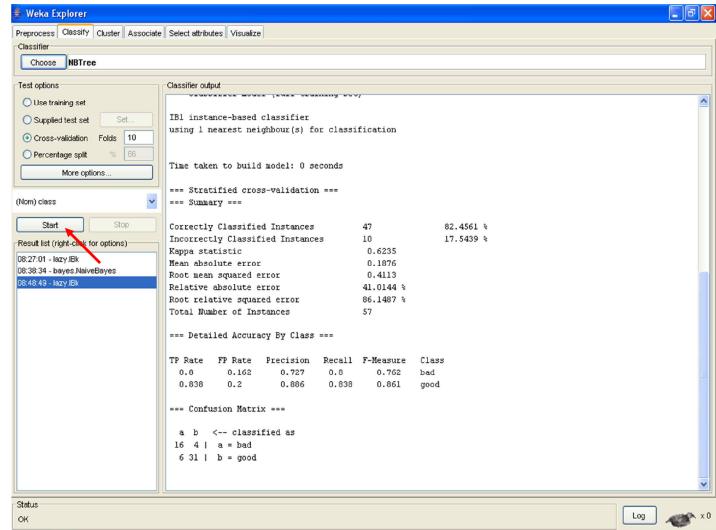
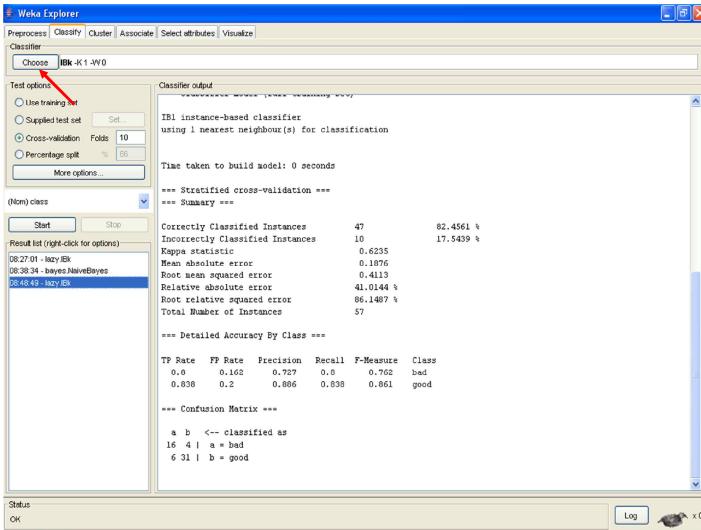


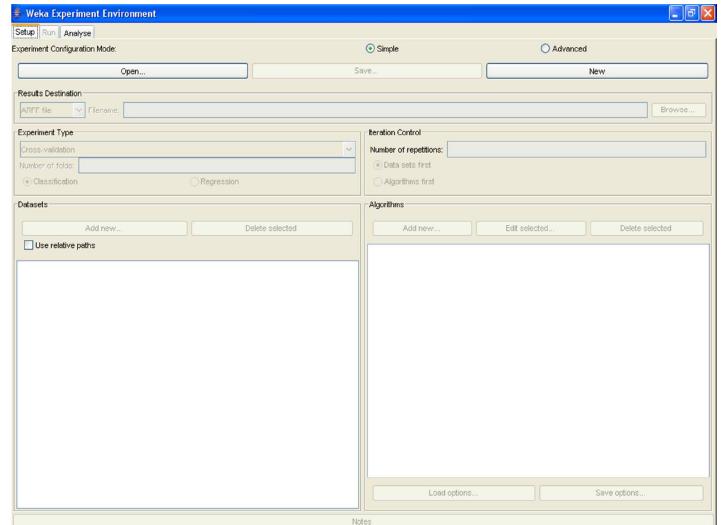
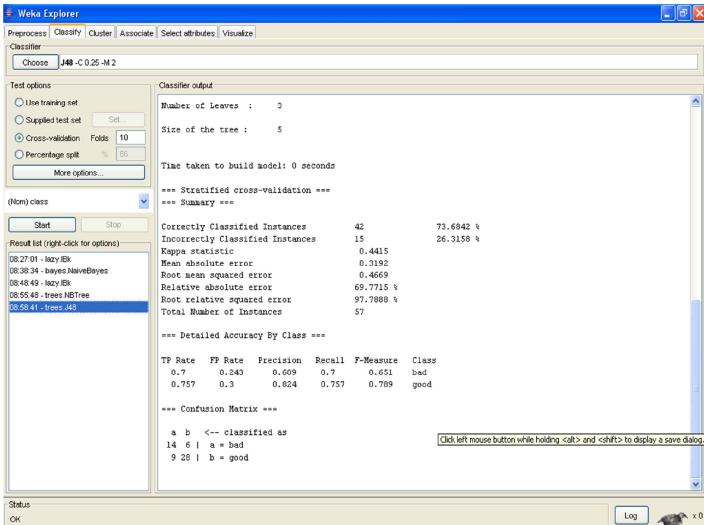






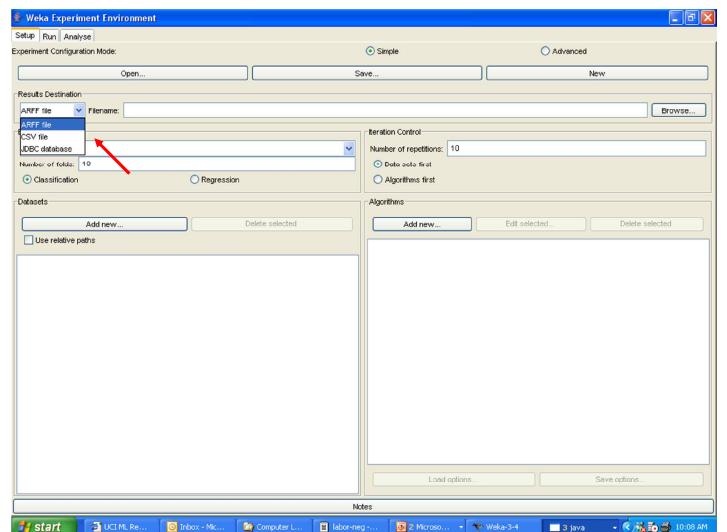
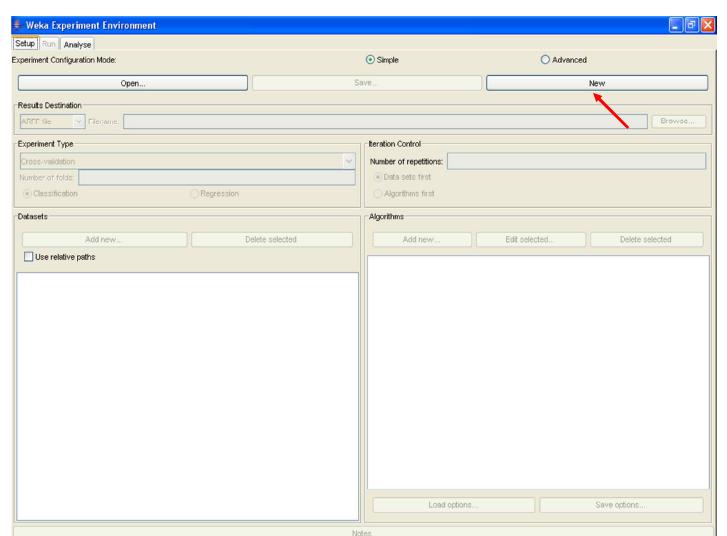


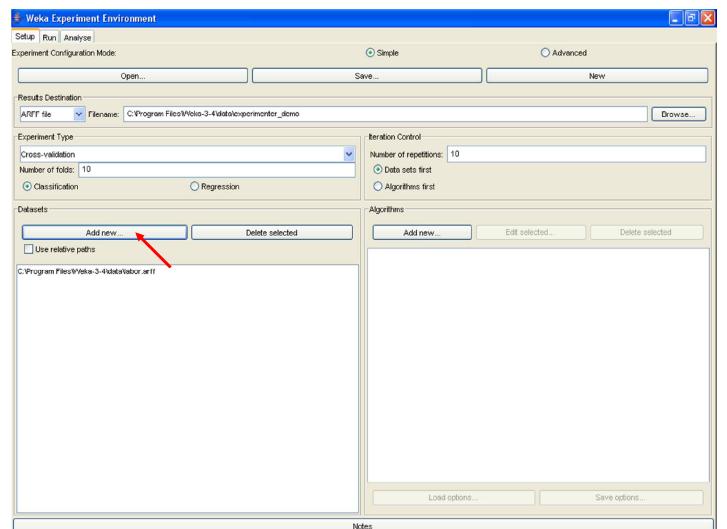
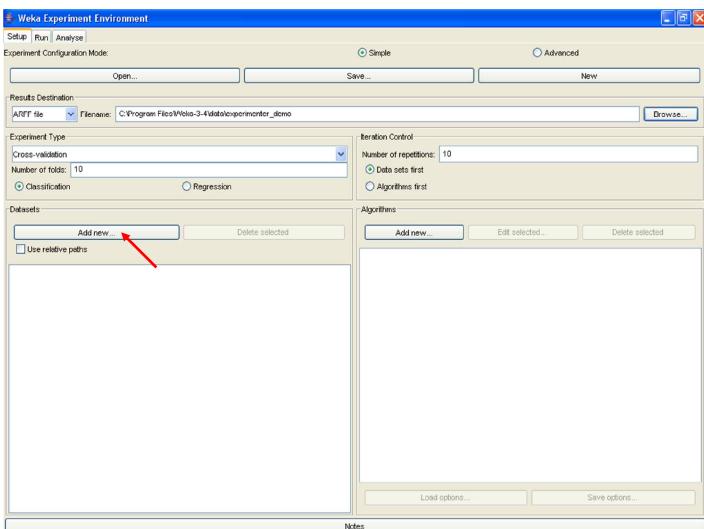
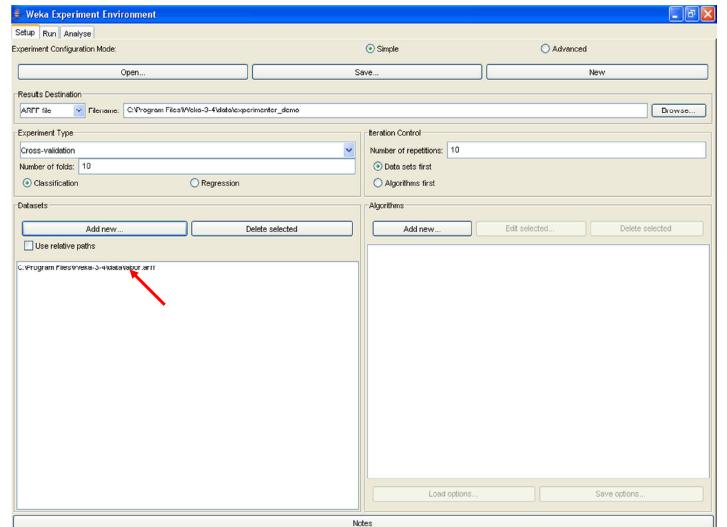
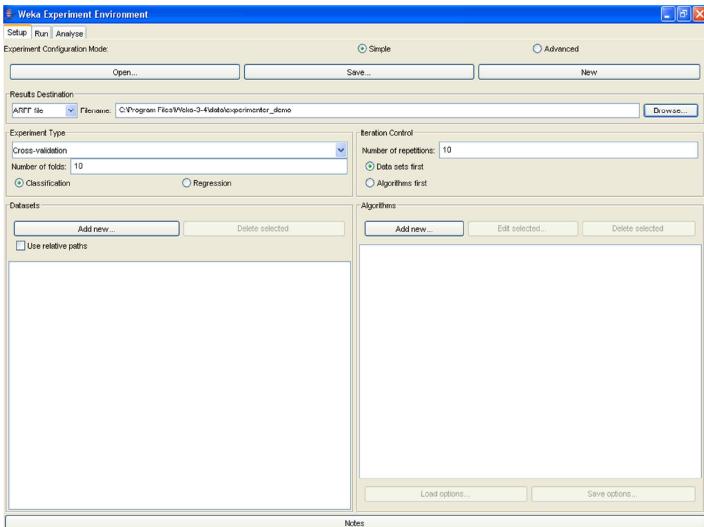
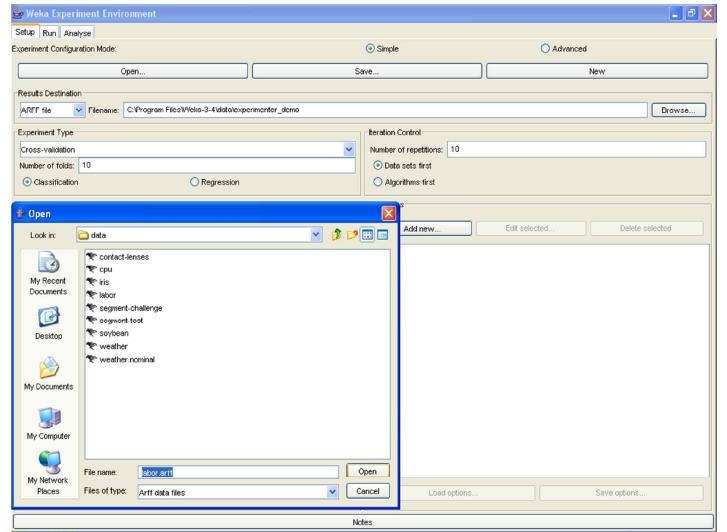
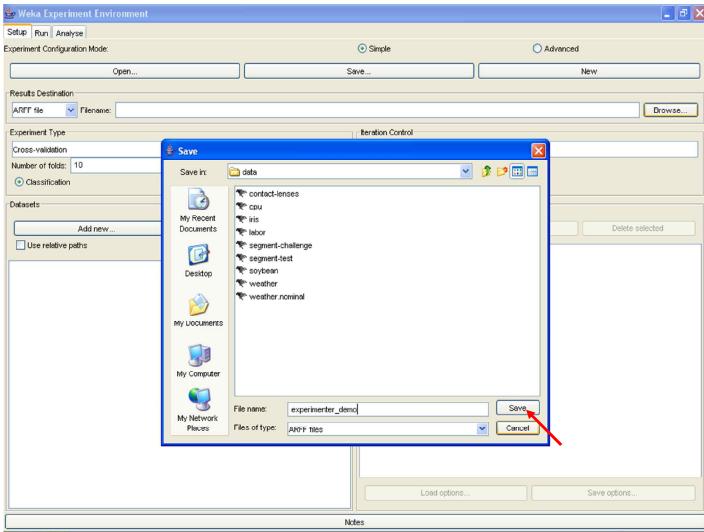


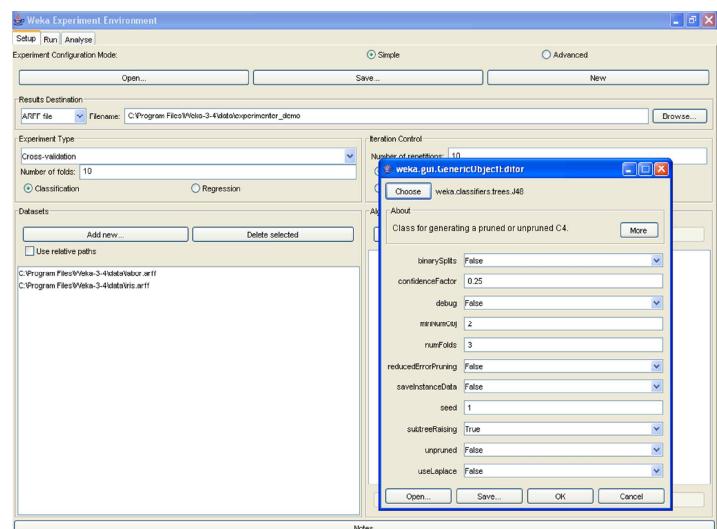
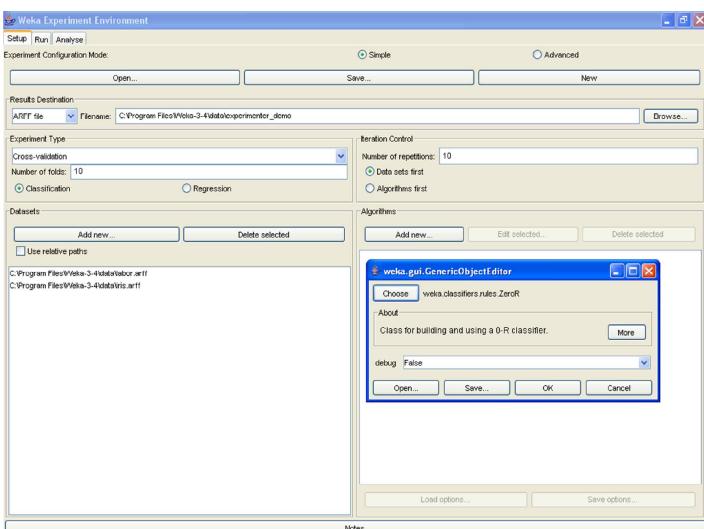
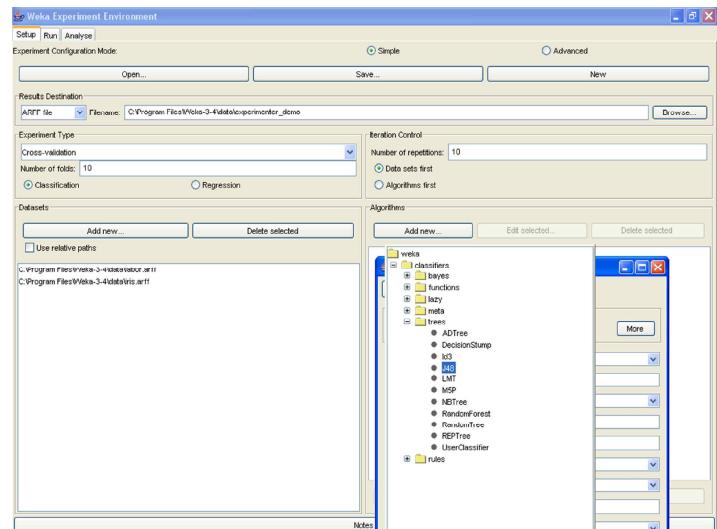
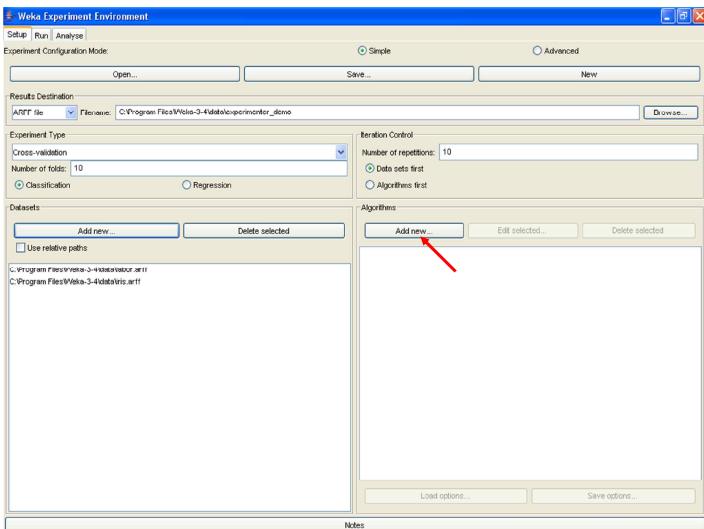
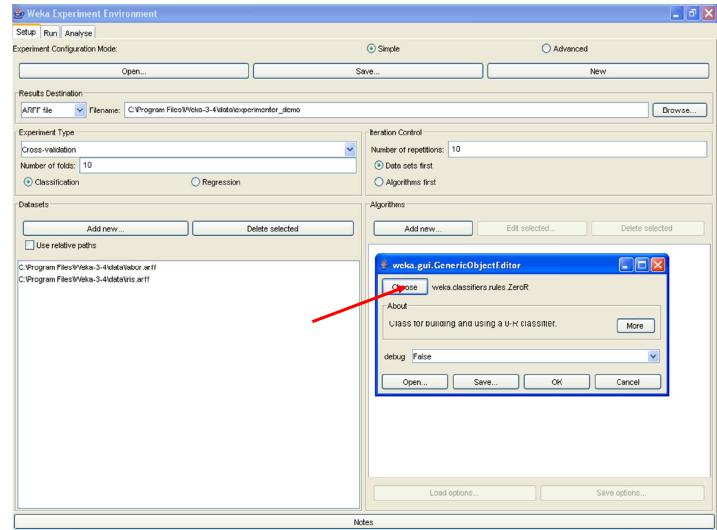
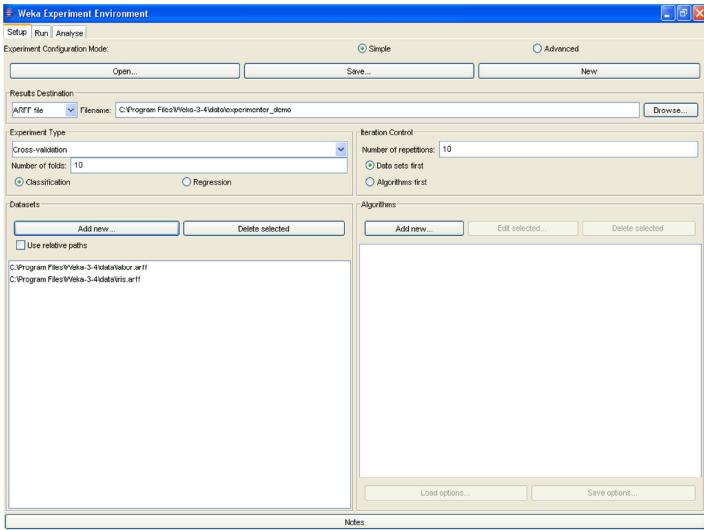


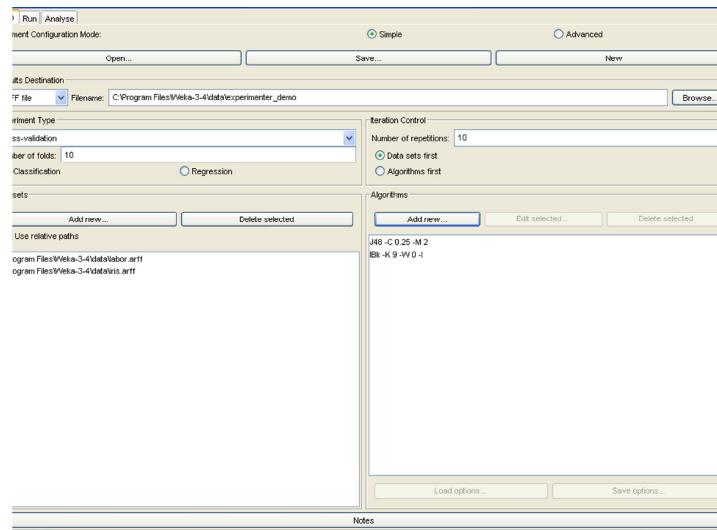
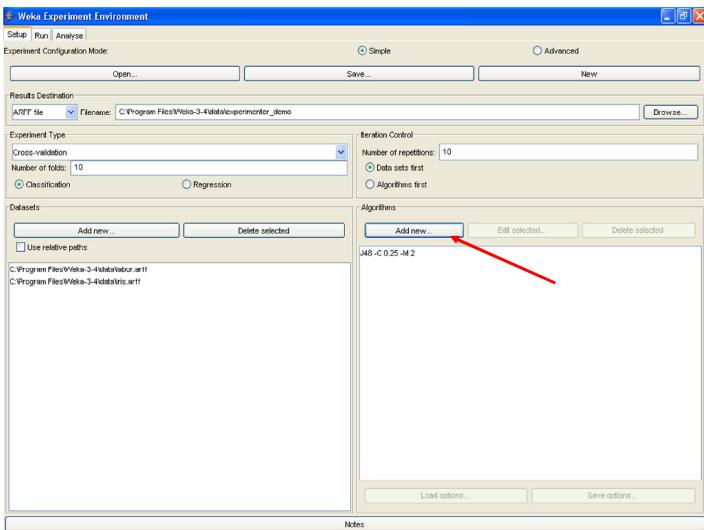
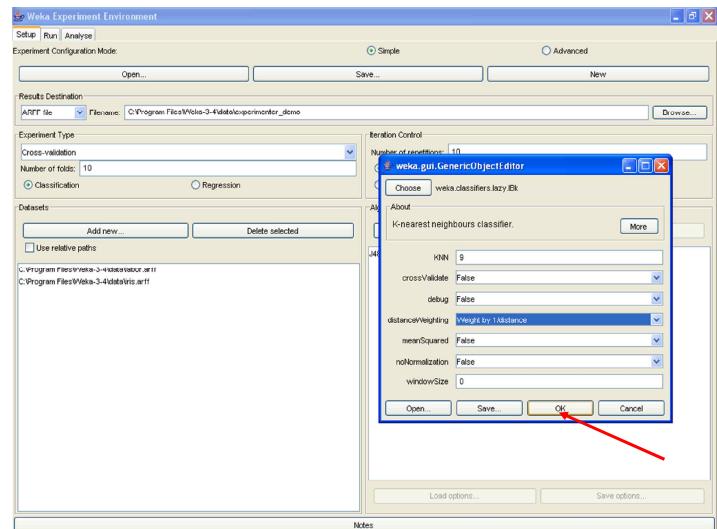
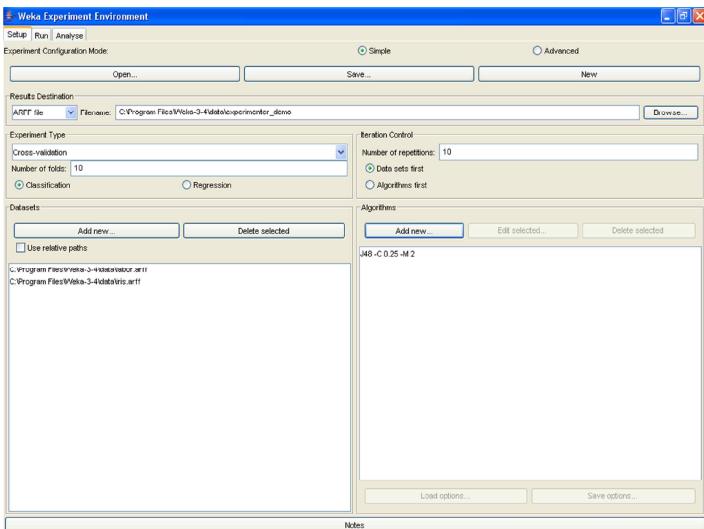
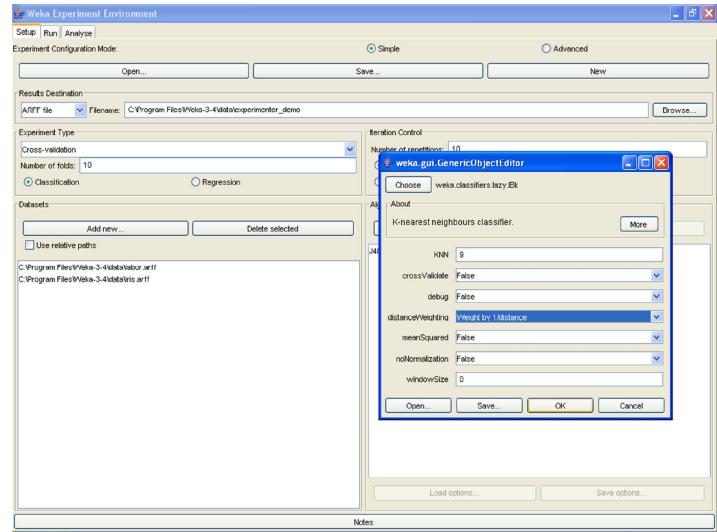
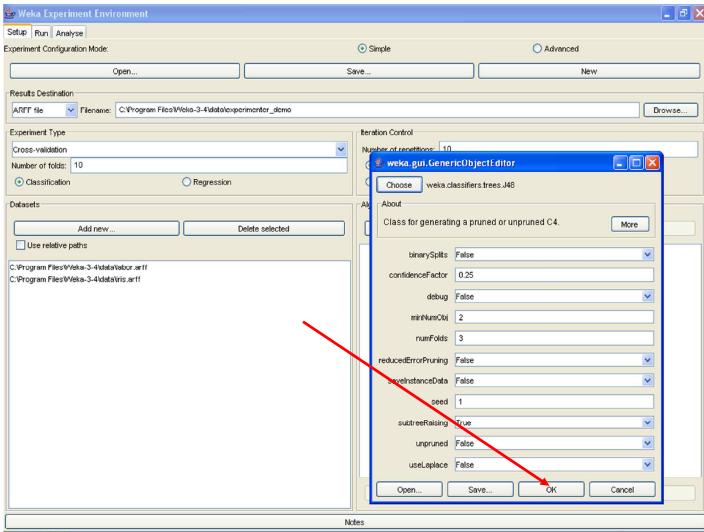
Performing experiments

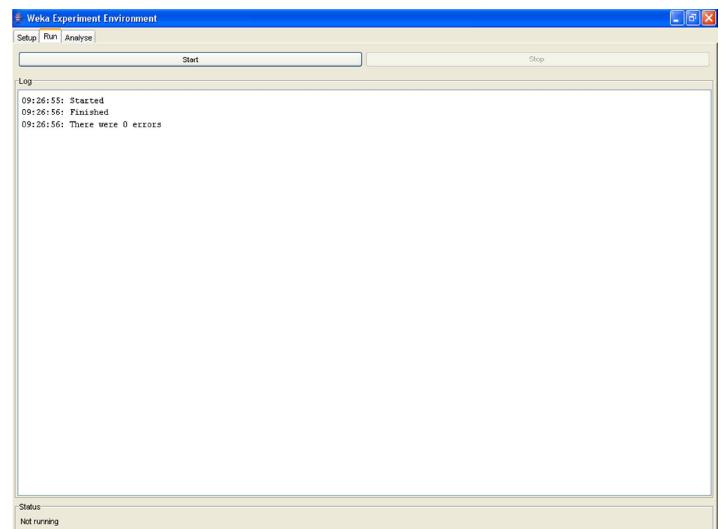
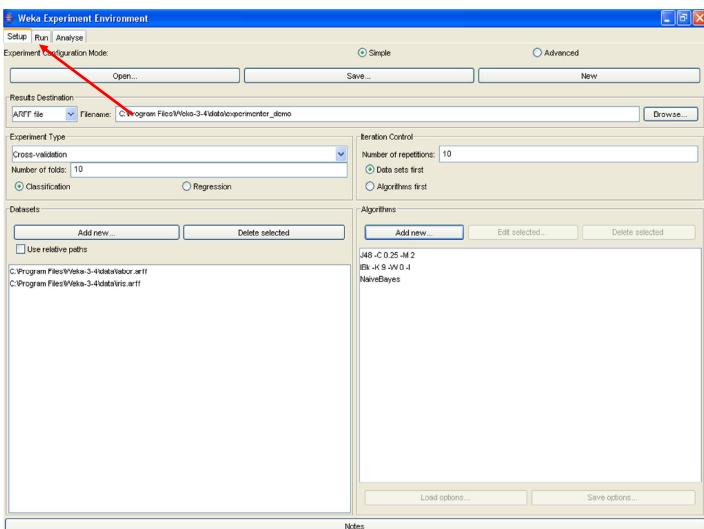
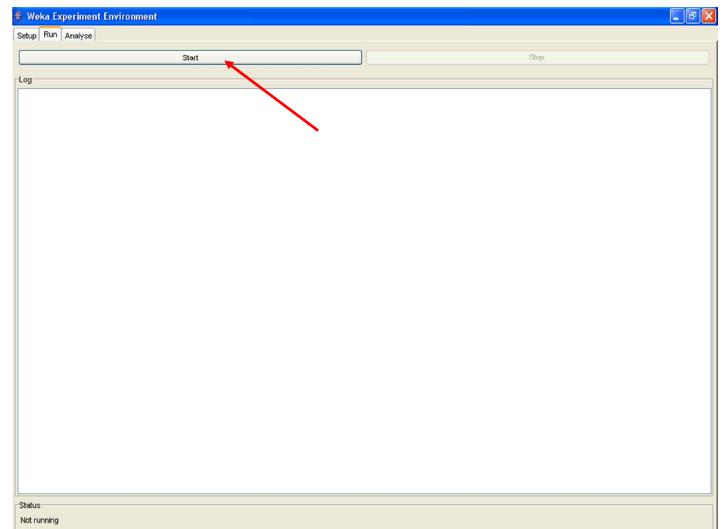
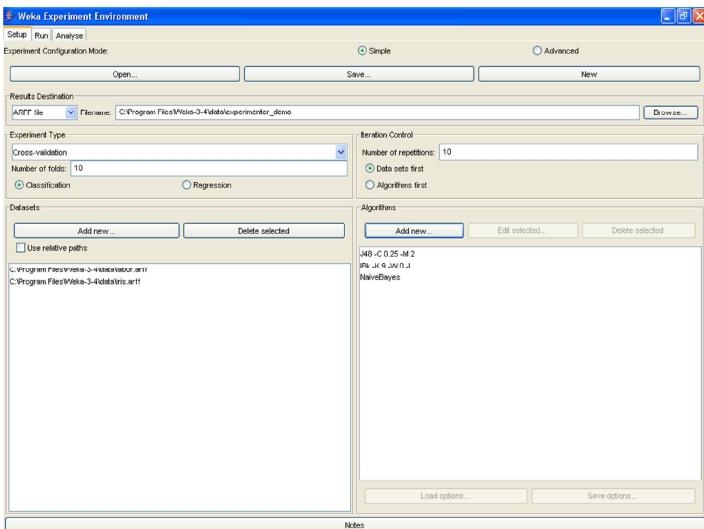
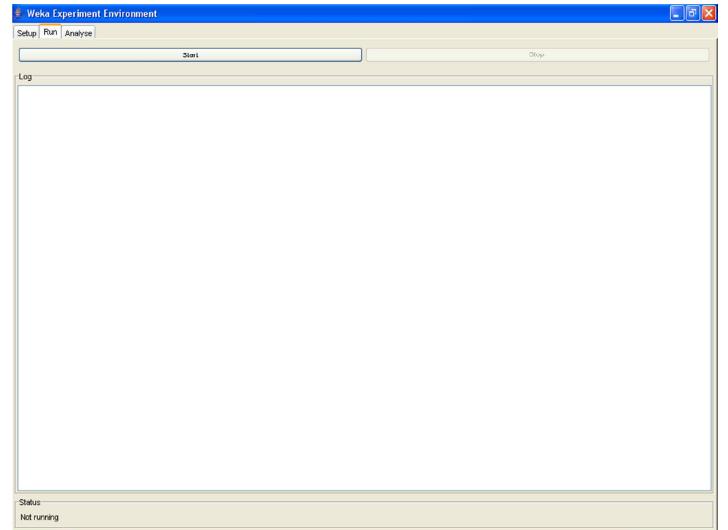
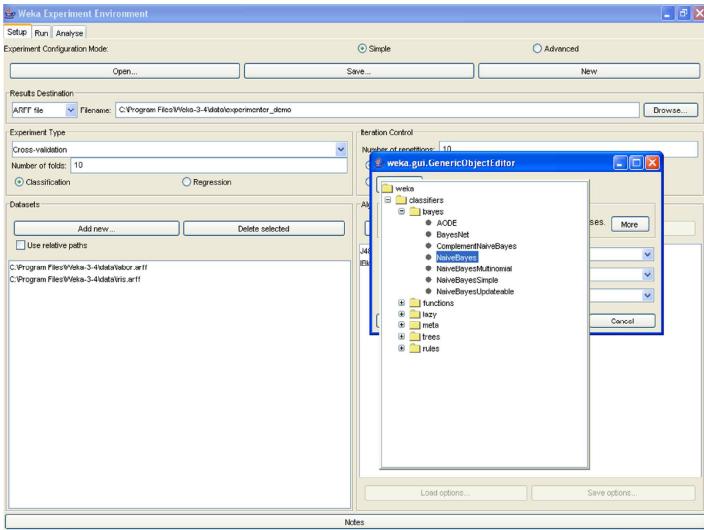
- Experimenter GUI makes it easy to compare the performance of different learning schemes
- For classification and regression problems
- Results can be written into file or database
- Evaluation options: cross-validation, learning curve, hold-out
- Can also iterate over different parameter settings
- Significance-testing built in!

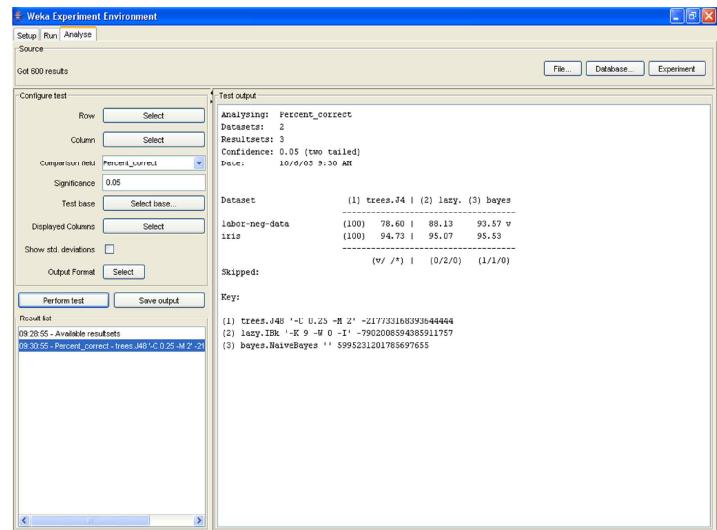
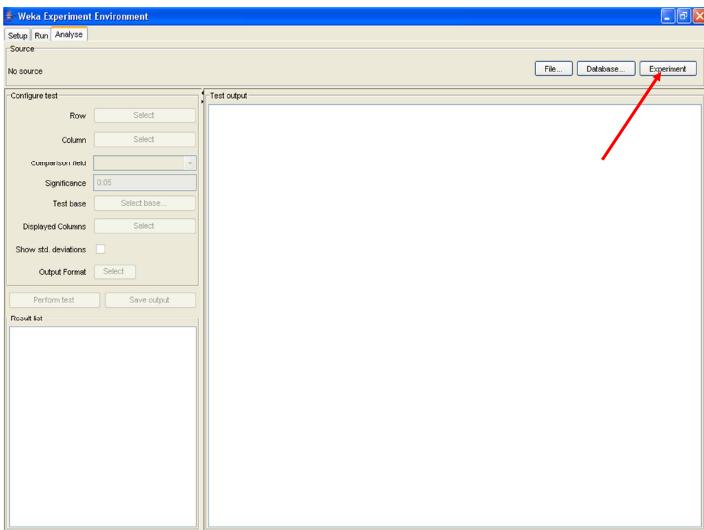
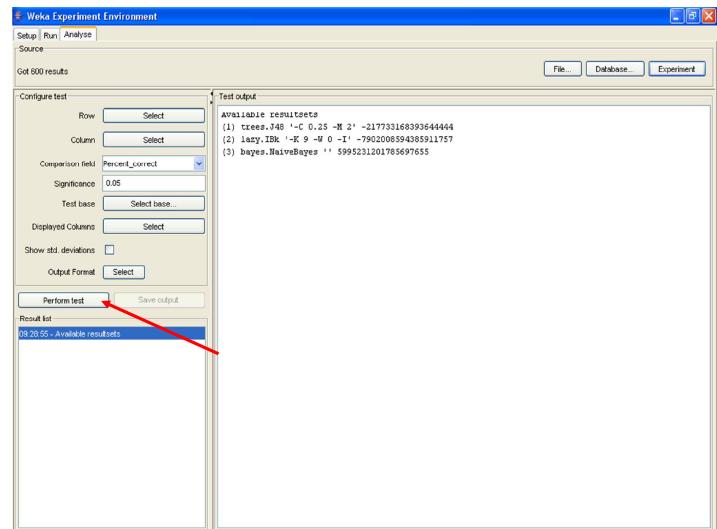
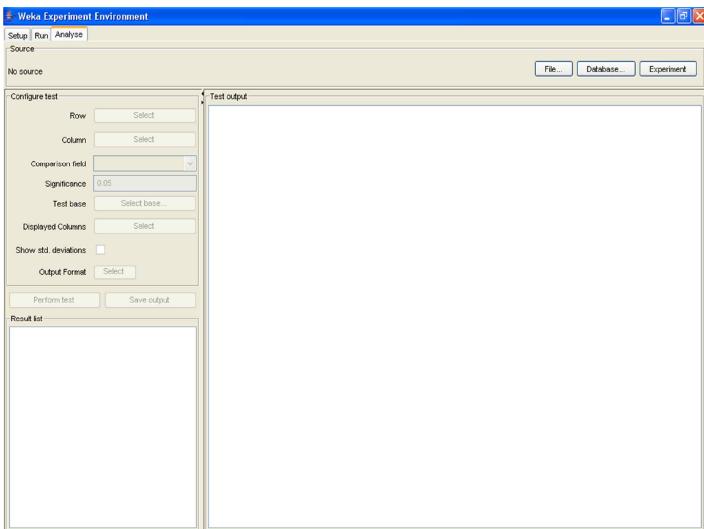
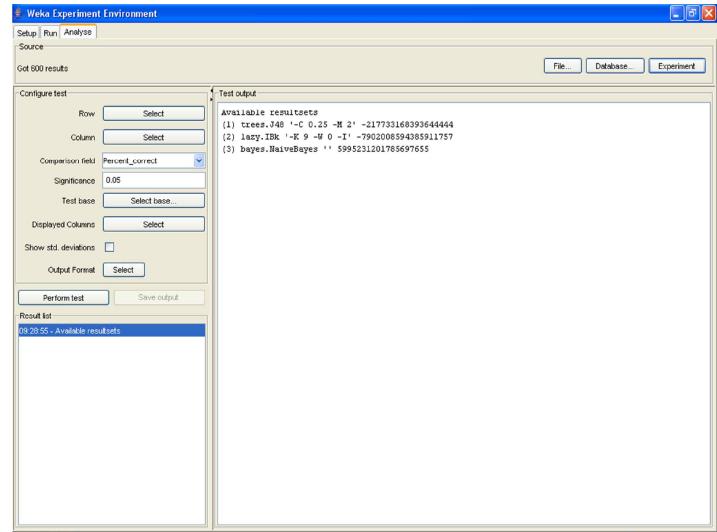
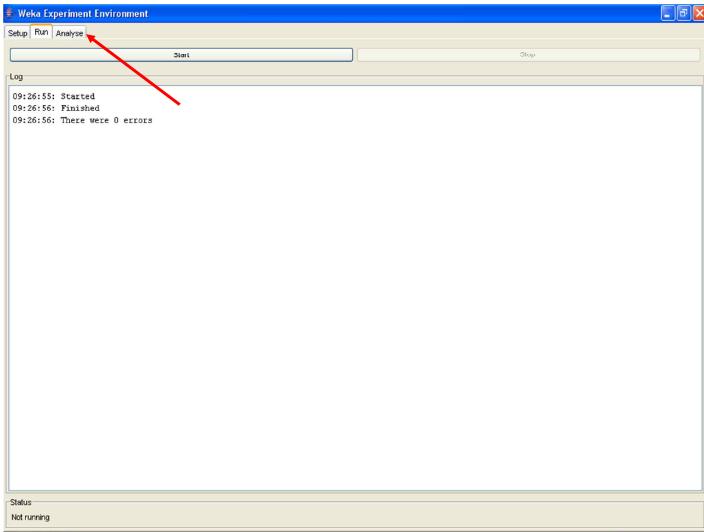




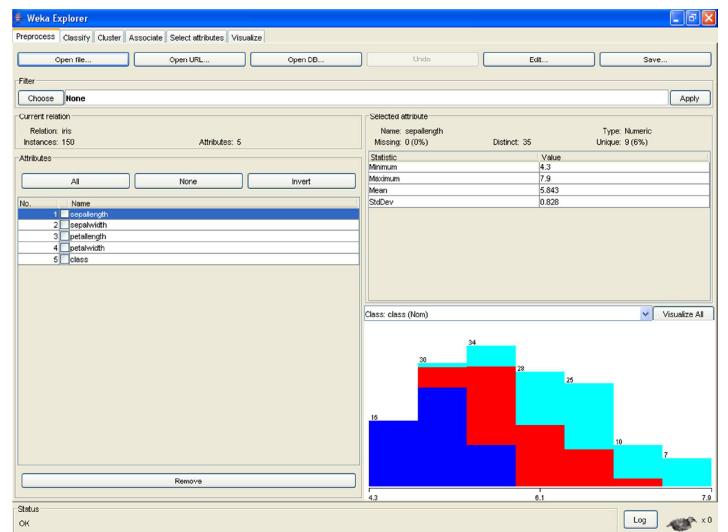
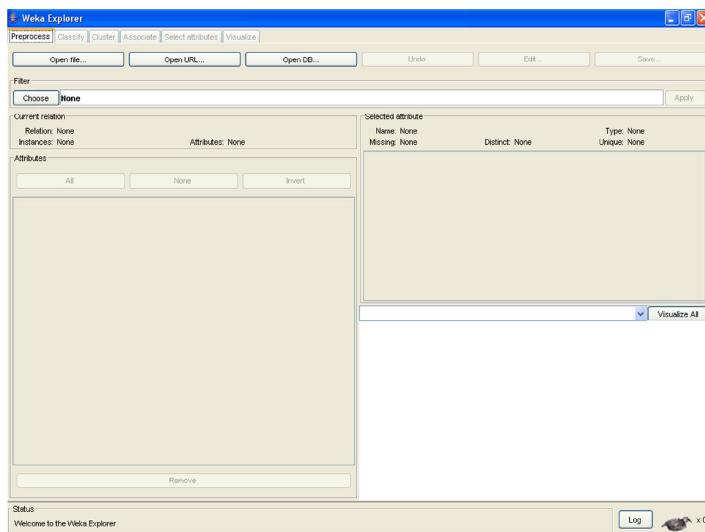
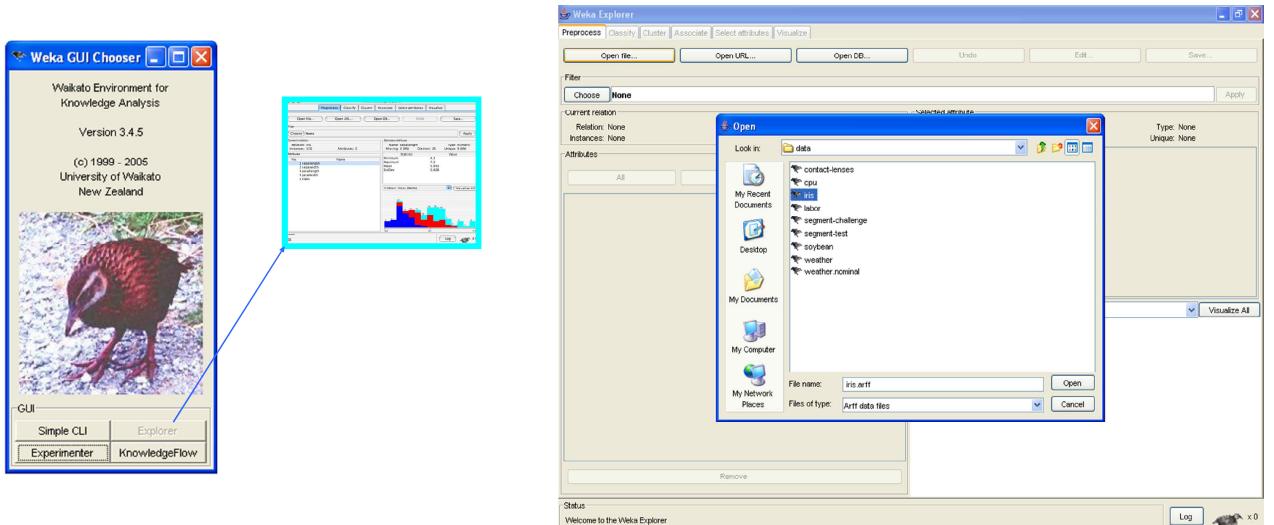
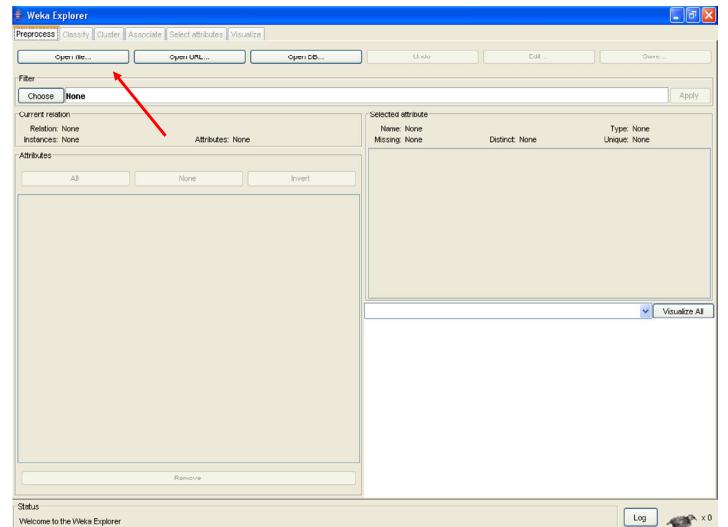


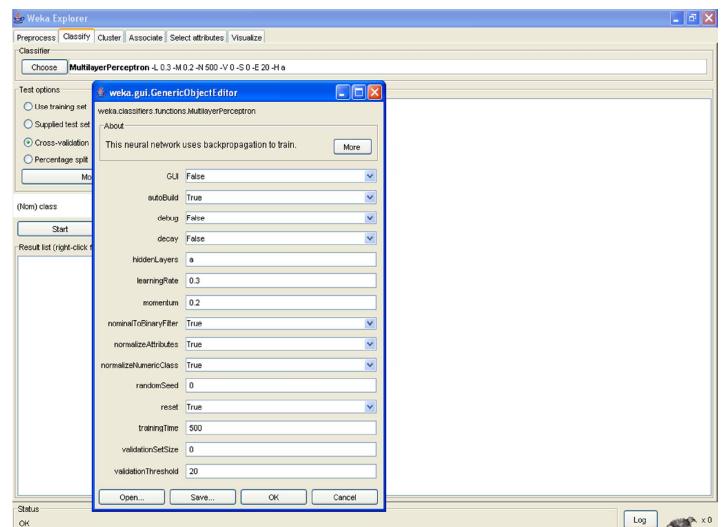
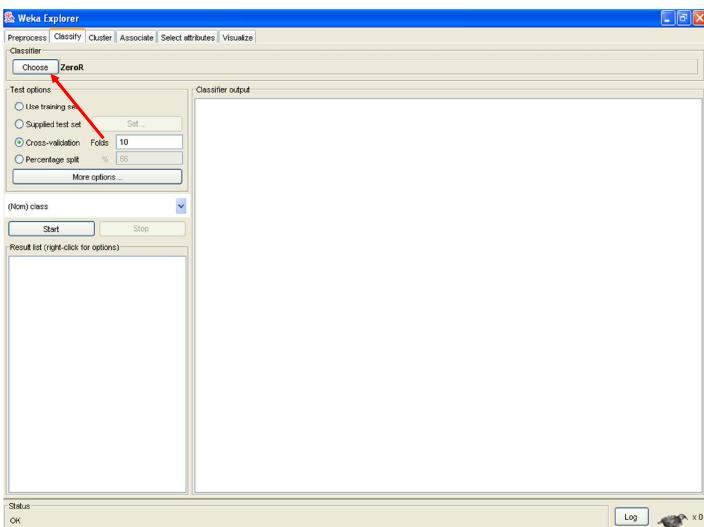
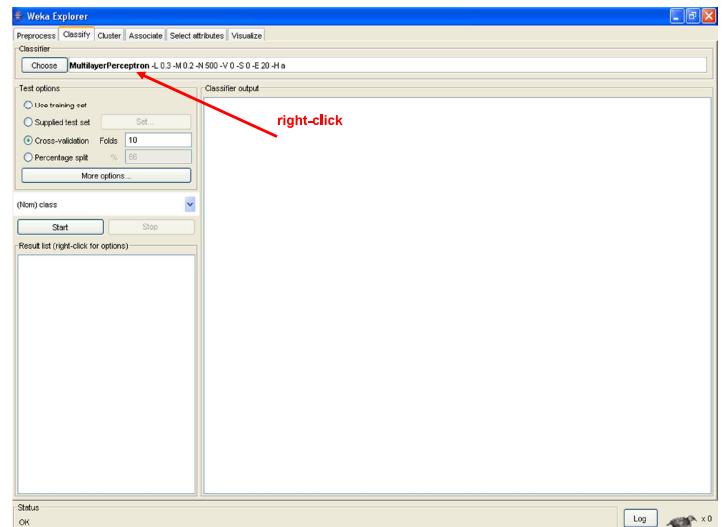
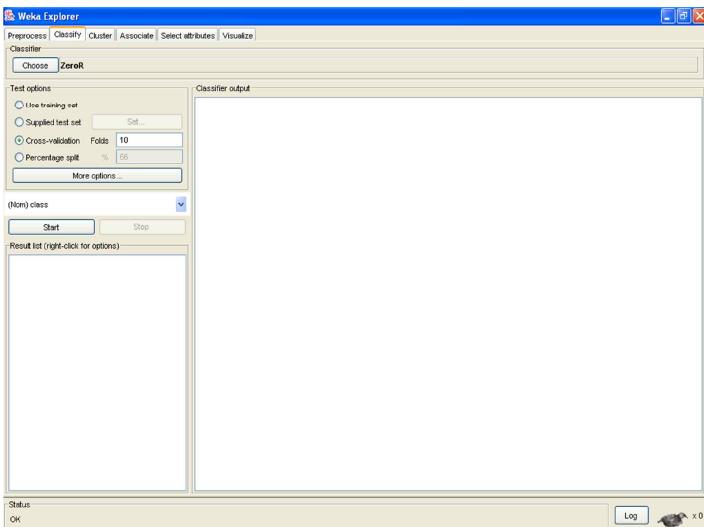
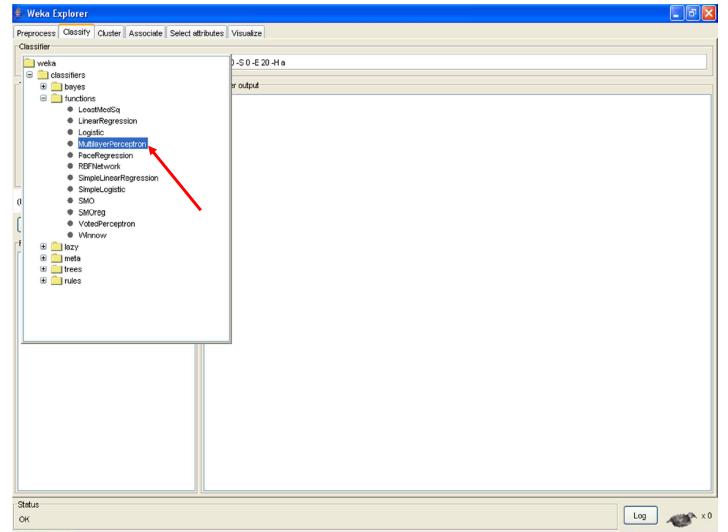
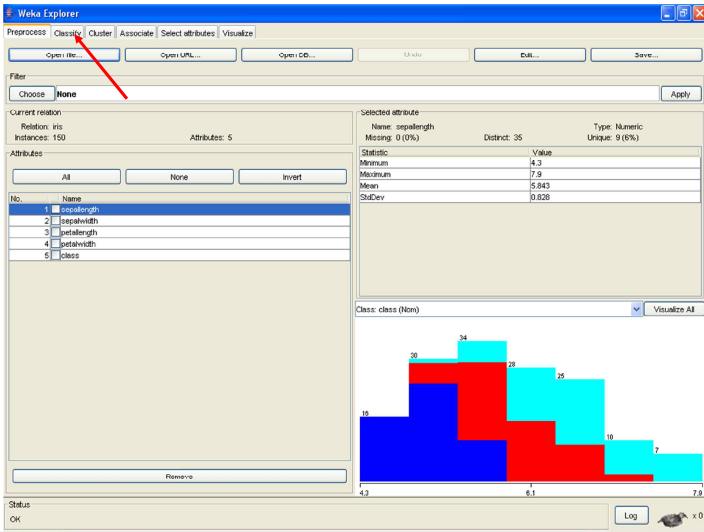


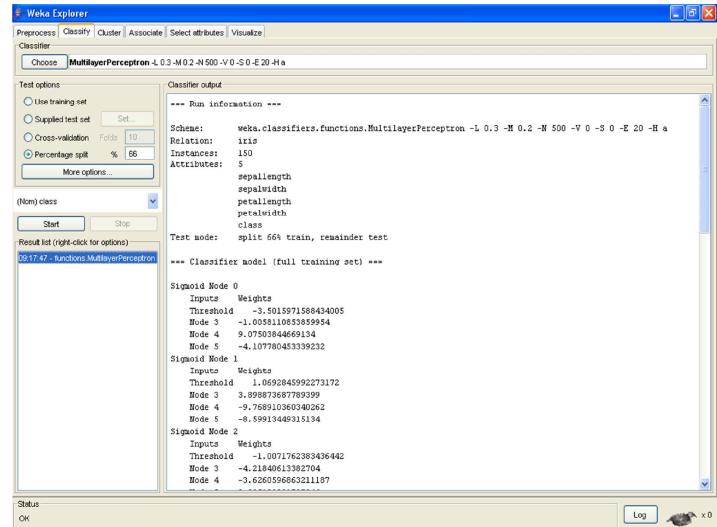
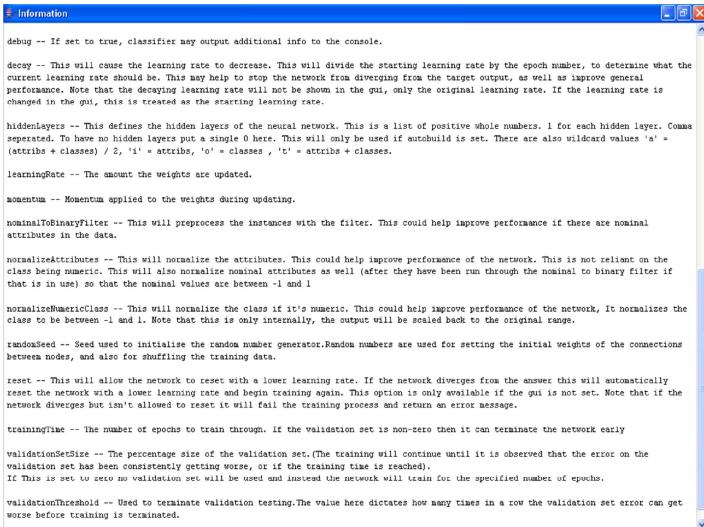
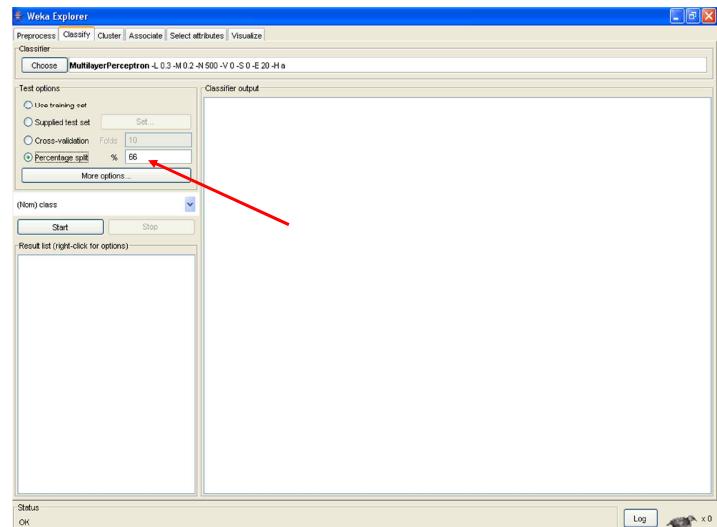
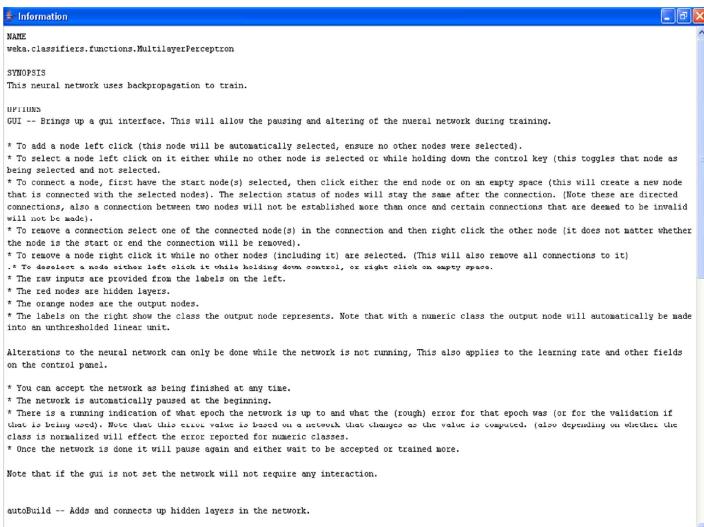
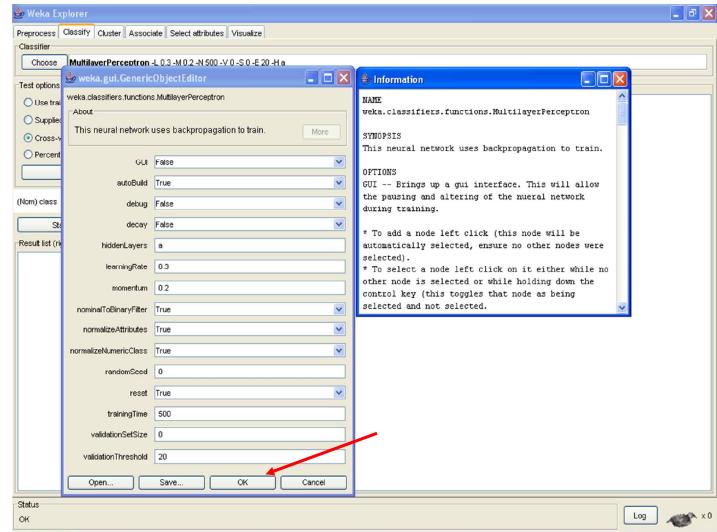
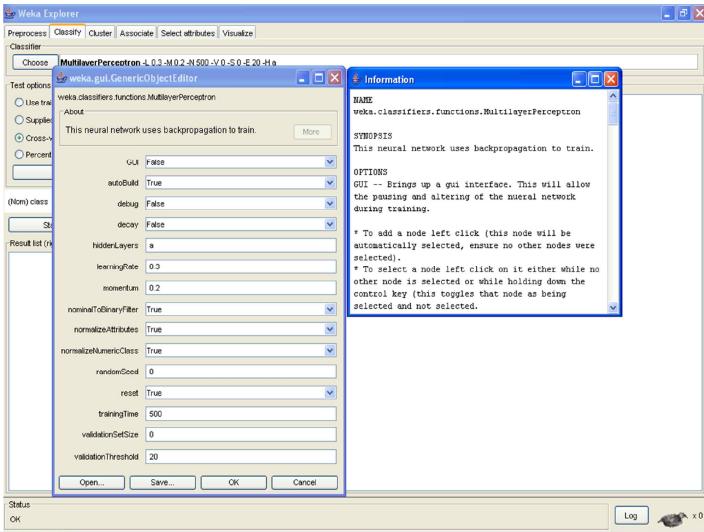




Neural Networks and SVM







Weka Explorer

Classifier chosen: MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Test options: Use training set (radio button selected)

Result list (right-click for options): 2017-47 - functions MultilayerPerceptron

Controls: Start, Stop, More options...

Log: Log icon

Status: OK

Neural Network

Controls: Start, Epoch 0, Num Of Epochs 500, Error per Epoch = 0, Accept, Learning Rate = 0.3, Momentum = 0.2

Status: Building model on training data...

Weka Explorer

Classifier chosen: MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -G -B

Test options: Use training set (radio button selected)

Result list (right-click for options): 2017-47 - weka.gui.GenericObjectEditor

Controls: Start, More options...

Log: Log icon

Status: OK

Neural Network

Controls: Start, Epoch 0, Num Of Epochs 500, Error per Epoch = 0, Accept, Learning Rate = 0.3, Momentum = 0.2

Status: Building model on training split.

Weka Explorer

Classifier chosen: MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -G -R

Test options: Use training set (radio button selected)

Result list (right-click for options): 2022-58 - functions MultilayerPerceptron

Controls: Start, Stop, More options...

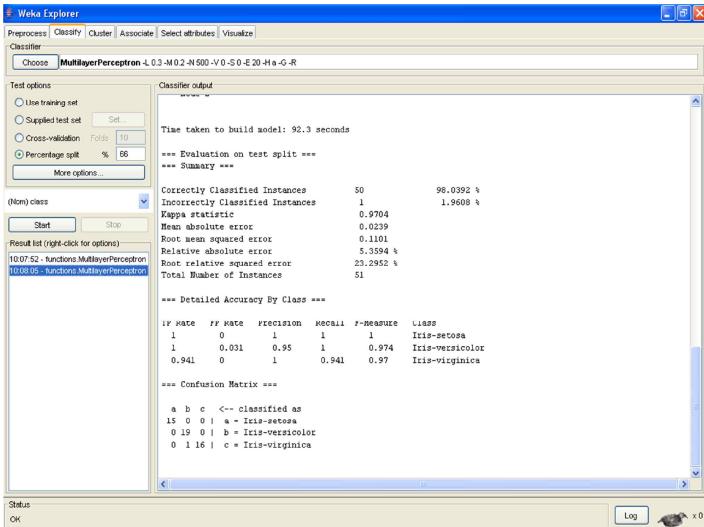
Log: Log icon

Status: OK

Neural Network

Controls: Start, Epoch 500, Num Of Epochs 500, Error per Epoch = 0.0062655, Accept, Learning Rate = 0.3, Momentum = 0.2

Status: Building model on training split...



Using "backprop.cpp"

- Download the executable file as well as the example training and testing data sets
- The training and data set were obtained from the UCI Machine Learning Repository's Wine Recognition Database
- The original data set was randomly divided to create the training (2/3) and testing (1/3) data sets

Using "backprop.cpp"

- The training data set contains the input and output values.
- The testing data set contains **only the input** values

Try with different network topologies

Other Neural Network Software

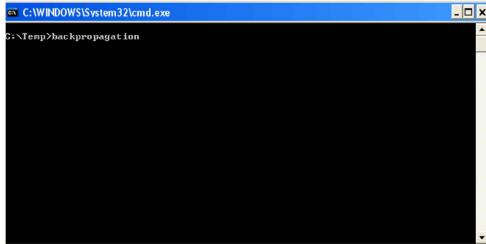
- SNNS- Stuttgart Neural Network Simulator
<http://www-ra.informatik.uni-tuebingen.de/SNNS/>
- Alyuda NeuroIntelligence
<http://www.alyuda.com/neural-networks-software.htm>
- "backprop.cpp" from V. Rao and H. Rao
C++ Neural Network and Fuzzy Logic
Backpropagation Simulation version 1

Using "backprop.cpp"

- Please note the changes made to the output values for this particular example.
 - Class 1: output node 1 = 0; output node 2 = 0
 - Class 2: output node 1 = 0; output node 2 = 1
 - Class 3: output node 1 = 1; output node 2 = 1

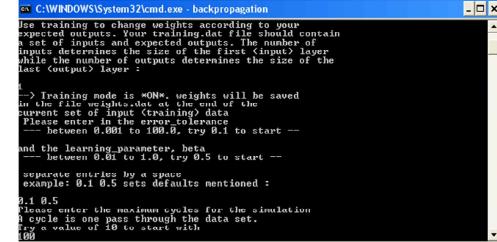
Using "backprop.cpp" (TRAINING)

- On the DOS prompt, type "backpropagation" to run the application



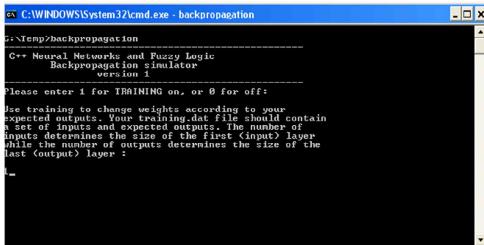
Using "backprop.cpp" (TRAINING)

- Enter the maximum number of cycles. One cycle is one pass through the data set.



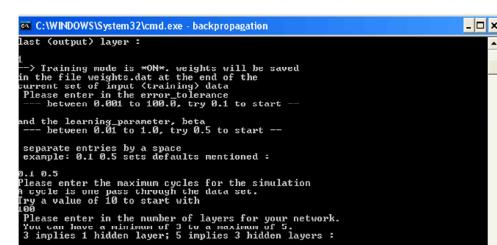
Using "backprop.cpp" (TRAINING)

- Enter "1" for TRAINING



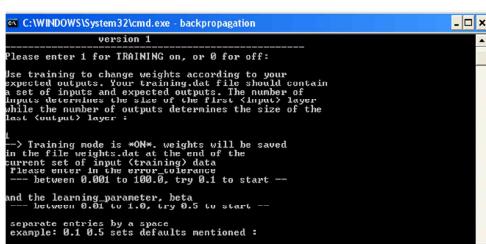
Using "backprop.cpp" (TRAINING)

- Enter the number of layers for your network.



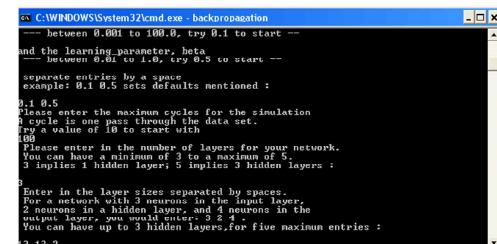
Using "backprop.cpp" (TRAINING)

- Enter the error tolerance and the learning rate



Using "backprop.cpp" (TRAINING)

- Enter the layer sizes separated by spaces.
- IMPORTANT:** the number of input and output nodes must match the training data (on the "training.dat" file)



Using "backprop.cpp" (TRAINING)

- Run the neural network. Weights were saved on the "weights.dat" file. They will be used later during testing

```
done: results in file output.dat
training: last vector only
not training: full cycle
weights saved in file weights.dat
>average error per cycle = 1.61664 <---
>error last cycle = 1.24692 <---
>error last cycle per pattern= 0.111977 <---
----->total cycles = 100 <---
----->total patterns = 12400 <---

C:\Temp>
```

Using "backprop.cpp" (TESTING)

- Enter the number of layers for your network.

IMPORTANT: This must be the same used for training

```
C:\WINDOWS\system32\cmd.exe - backpropagation
C++ Neural Networks and Fuzzy Logic
Backpropagation simulator
version 1

Please enter 1 for TRAINING on, or 0 for off:
Use training to change weights according to your
expected outputs. Your training.dat file should contain
a set of inputs and expected outputs. The number of
inputs determines the size of the first (input) layer
while the number of outputs determines the size of the
last (output) layer :
-> Training mode is *OFF*. weights will be loaded
from the file weights.dat and the current
test.dat file will be used. For the test
data set, the test.dat file should contain
only inputs, and no expected outputs.
Please enter in the number of layers for your network.
You can have a minimum of 3 to a maximum of 5;
3 implies 1 hidden layer, 5 implies 3 hidden layers :
3

C:\Temp>
```

Using "backprop.cpp" (TESTING)

- On the DOS prompt, type "backpropagation" to run the application again

```
done: results in file output.dat
training: last vector only
not training: full cycle
weights saved in file weights.dat
>average error per cycle = 1.61664 <---
>error last cycle = 1.24692 <---
>error last cycle per pattern= 0.111977 <---
----->total cycles = 100 <---
----->total patterns = 12400 <---

C:\Temp>backpropagation
```

Using "backprop.cpp" (TESTING)

- Enter the layer sizes separated by spaces.

IMPORTANT: This must be the same used for training

```
C:\WINDOWS\system32\cmd.exe - backpropagation
Use training to change weights according to your
expected outputs. Your training.dat file should contain
a set of inputs and expected outputs. The number of
inputs determines the size of the first (input) layer
while the number of outputs determines the size of the
last (output) layer :
-> Training mode is *OFF*. weights will be loaded
from the file weights.dat and the current
test.dat file will be used. For the test
data set, the test.dat file should contain
only inputs, and no expected outputs.
Please enter in the layer sizes separated by spaces.
For a network with 3 neurons in the input layer,
2 neurons in a hidden layer, and 1 neurons in the
output layer, you would enter 3 2 1.
You can have up to 3 hidden layers,for five maximum entries :
13 13 2

C:\Temp>
```

Using "backprop.cpp" (TESTING)

- Enter "0" for TESTING

```
weights saved in file weights.dat
>average error per cycle = 1.24692 <---
>error last cycle per pattern= 0.111977 <---
----->total cycles = 100 <---
----->total patterns = 12400 <---

C:\Temp>backpropagation
C++ Neural Networks and Fuzzy Logic
Backpropagation simulator
version 1

Please enter 1 for TRAINING on, or 0 for off:
Use training to change weights according to your
expected outputs. Your training.dat file should contain
a set of inputs and expected outputs. The number of
inputs determines the size of the first (input) layer
while the number of outputs determines the size of the
last (output) layer :
0

C:\Temp>
```

Using "backprop.cpp" (TESTING)

- Run the neural network. Output results from the testing data set were saved on the "output.dat" file.

```
done: results in file output.dat
training: last vector only
not training: full cycle
----->total cycles = 1 <---
----->total patterns = 54 <---

C:\Temp>
```

Using “backprop.cpp” (OUTPUT)

Using “SVM^{*light*}”

- Training file ("trainsvm.dat"):

Using “SVM^{*light*}”

- Testing file ("testsvm.dat"):

SVM^{light} (<http://svmlight.joachims.org/>)

Using “SVM^{light}”

Using "SVM^{light}" (TRAINING)

- Download the executable file as well as the example training and testing data sets
 - Reference : T. Joachims, Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

`svm_learn` is called with the following parameters:

```
svm_learn [options] trainingexamples_file model_file
```

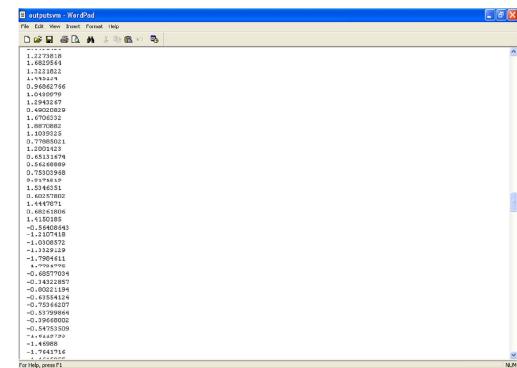
```
C:\Windows\System32\cmd.exe
C:\Temp>svm_learn trainsvm.dat modelsvm.dat
```

Using "SVM^{light}" (TRAINING)

Run `svm_learn`. The model is saved at the “`model_file`”. Later, this file will be used for testing

Using "SVM^{light}" (RESULTS)

- Output file ("outputsvm.dat"):



Using "SVM^{light}" (TESTING)

`svm_classify` is called with the following parameters:

```
svm_classify [options] testingexamples_file model_file output_file
```

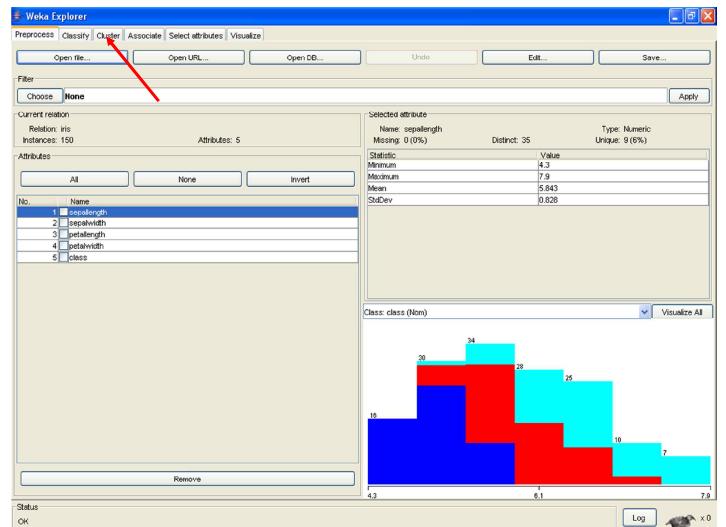
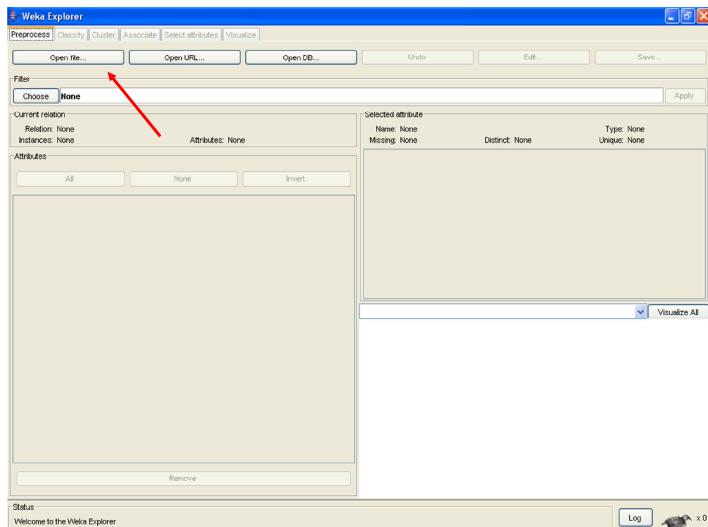
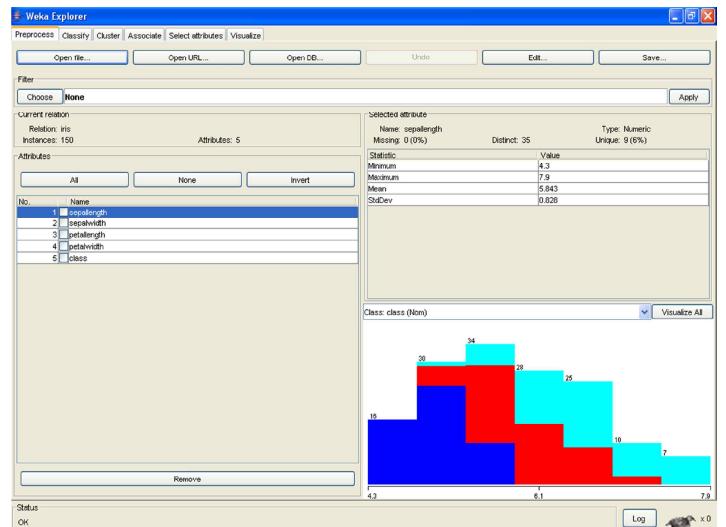
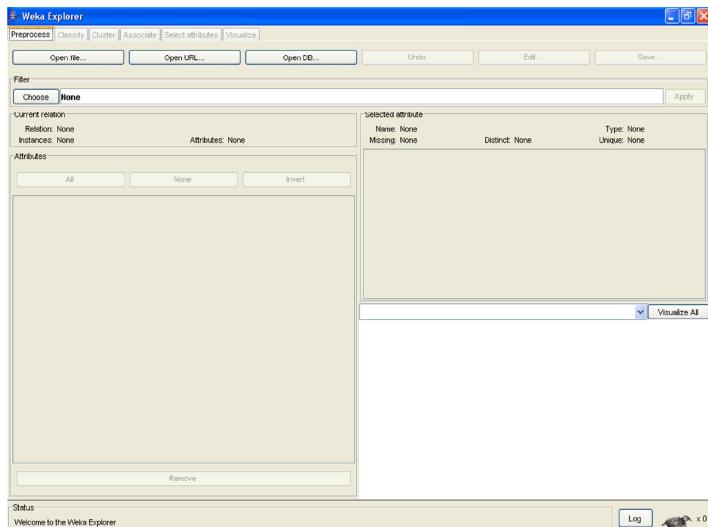
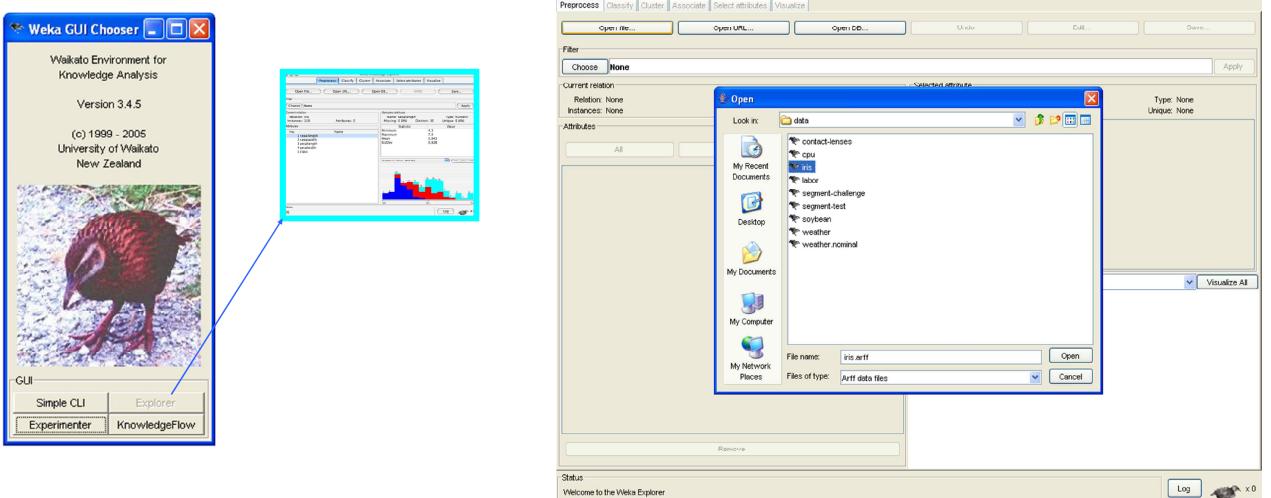
Clustering

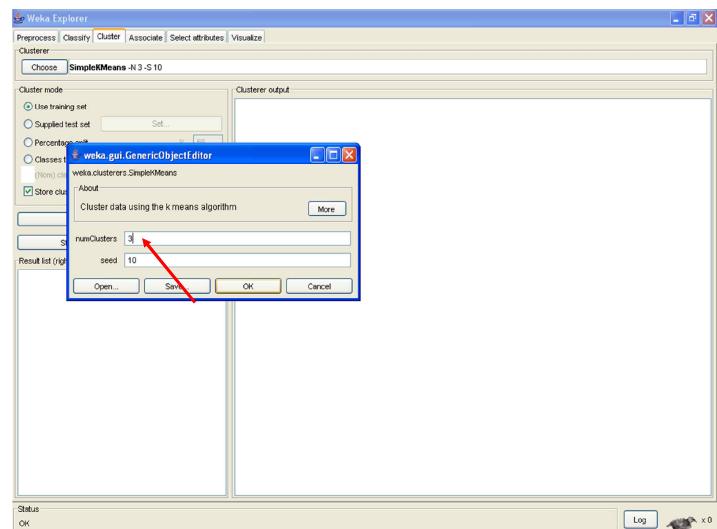
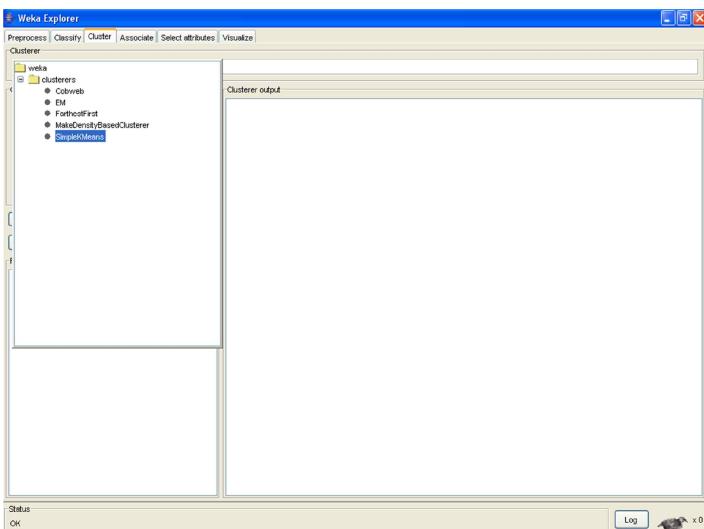
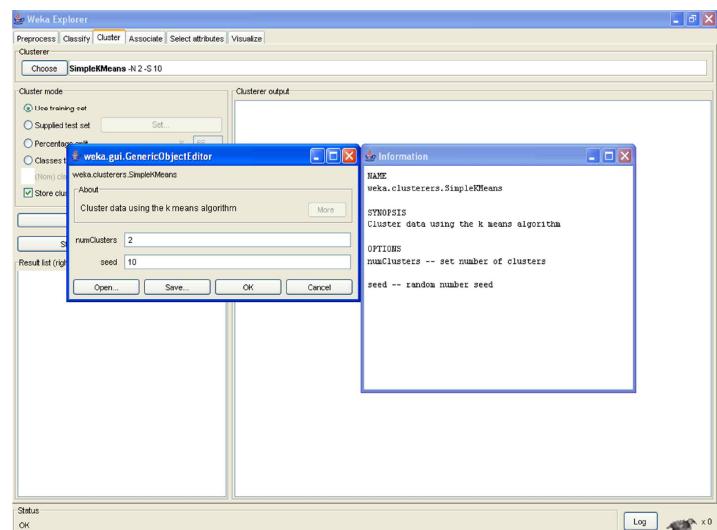
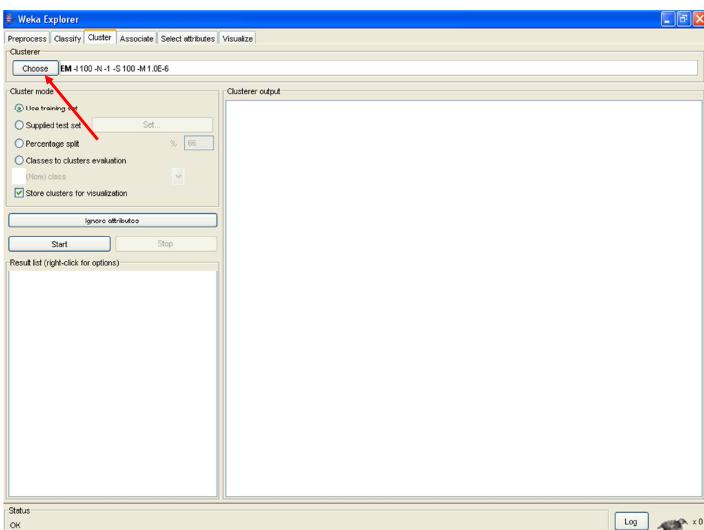
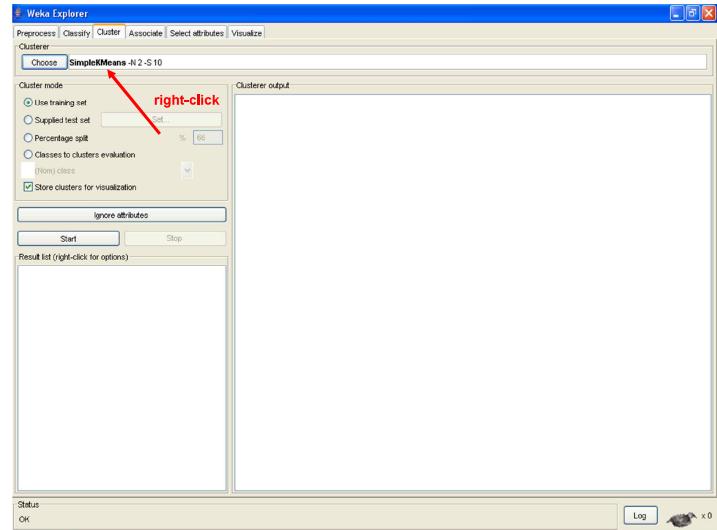
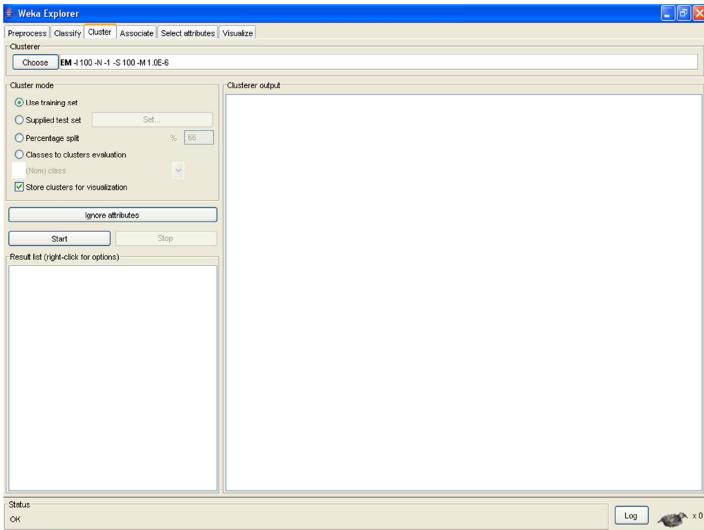
Using "SVM^{light}" (TESTING)

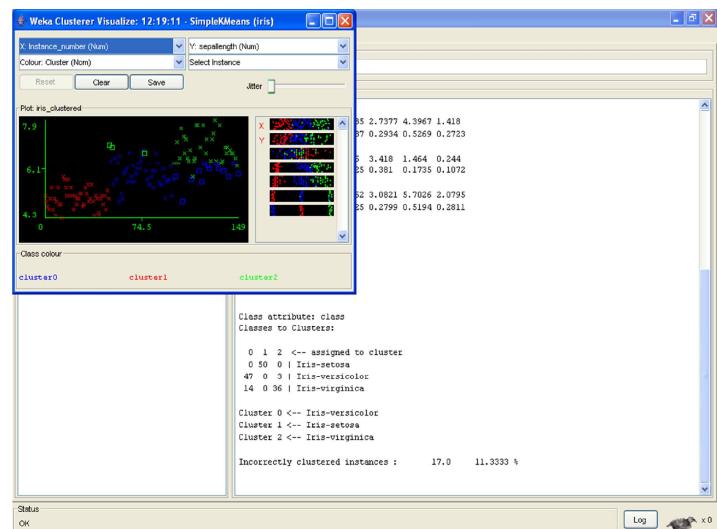
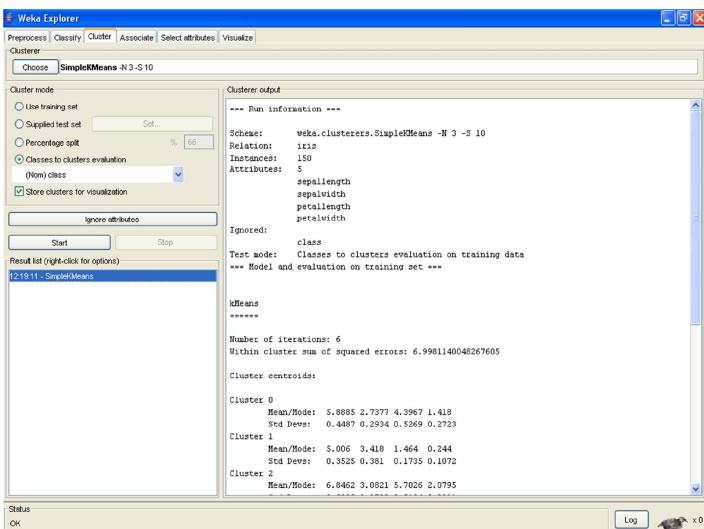
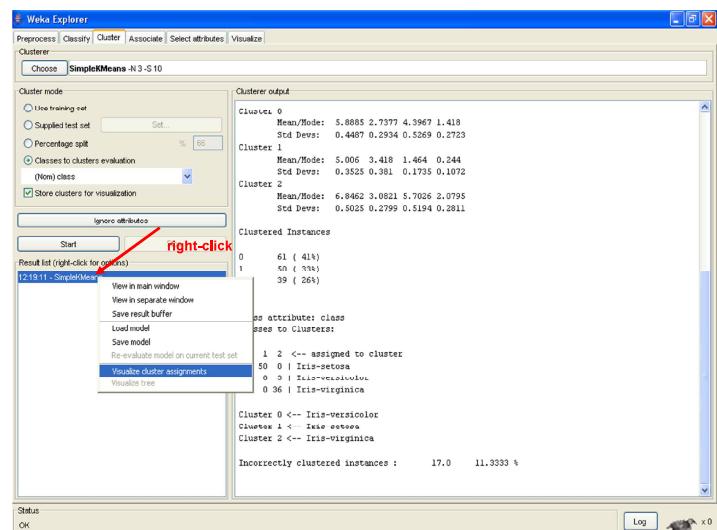
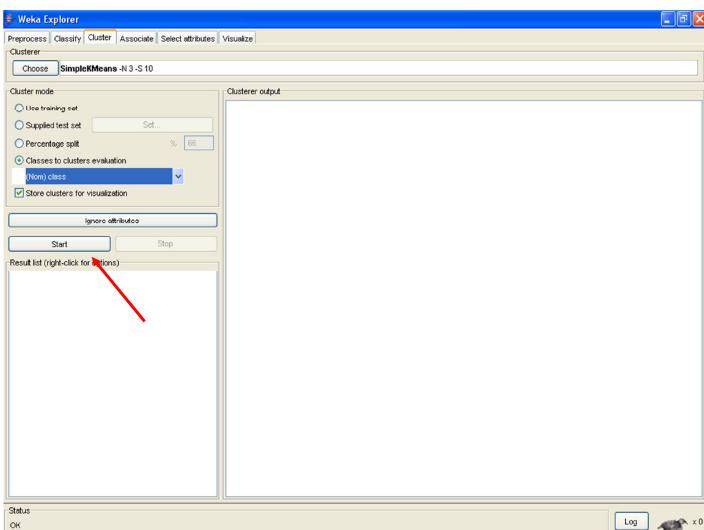
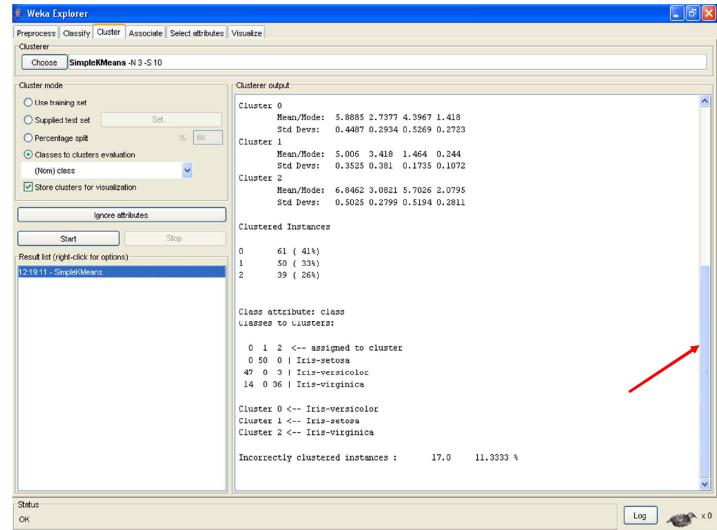
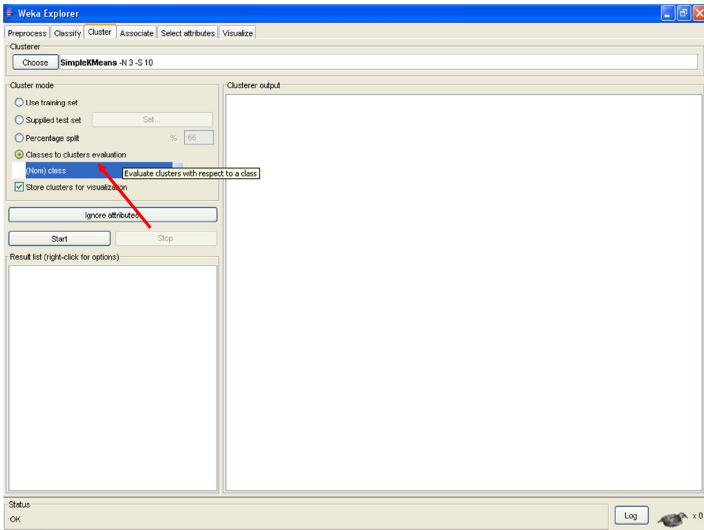
Run `svm_classify`. Results will be saved at the "output_file"

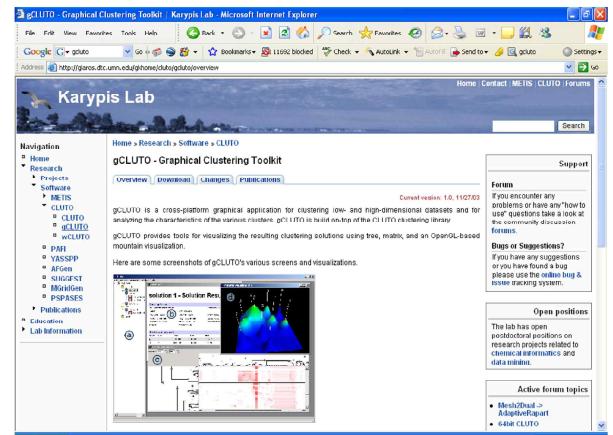
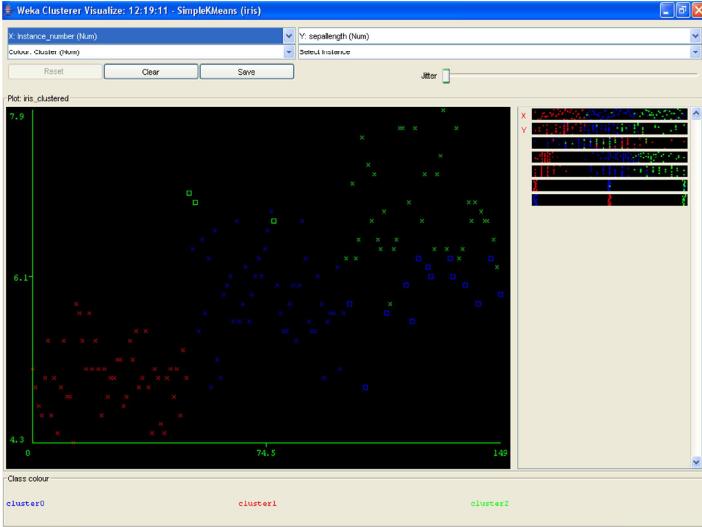
Explorer: clustering data

- WEKA contains “clusterers” for finding groups of similar instances in a dataset
 - Implemented schemes are:
 - k -Means, EM, Cobweb, X -means, FarthestFirst
 - Clusters can be visualized and compared to “true” clusters (if given)

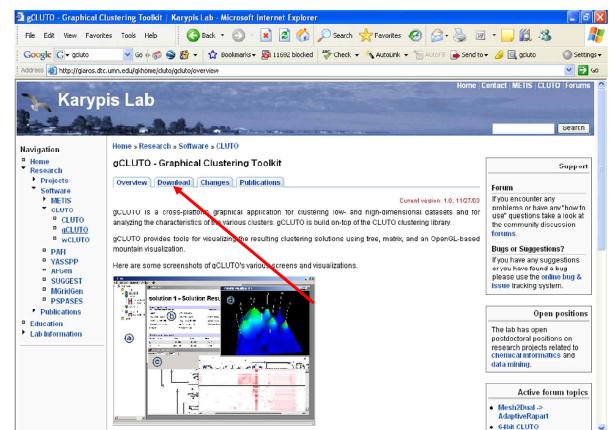






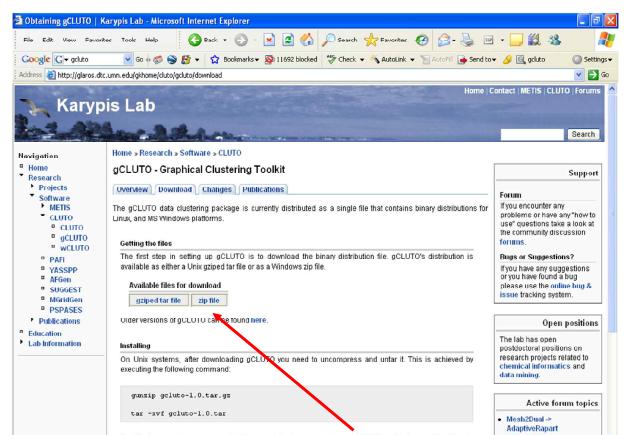


<http://glaros.dtc.umn.edu/gkhome/cluto/gcluto/overview>



<http://glaros.dtc.umn.edu/gkhome/cluto/gcluto/overview>

Experiment other visualization options



<http://glaros.dtc.umn.edu/gkhome/cluto/gcluto/overview>

Experiment different clustering parameters and algorithms

Download and Unzip "gcluto_1_0.zip"

gCLUTO Documentation

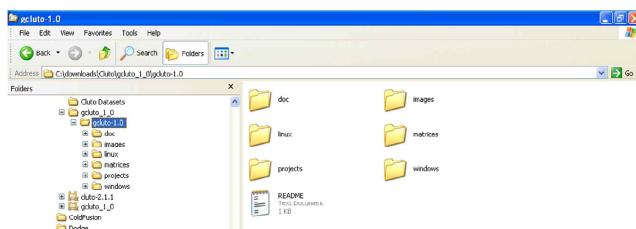
Version 1.0

Matt Rasmussen, Mark Newman, George Karypis
University of Minnesota, Copyright 2003
Last Modified: Wed Nov 19 15:16:53 CST 2003
<http://www.cs.umn.edu/~mrasmus/gcluto>

Table of Contents

- [Introduction](#)
 - [1.1 What is gCLUTO](#)

gCLUTO Folders and Files (extracted)



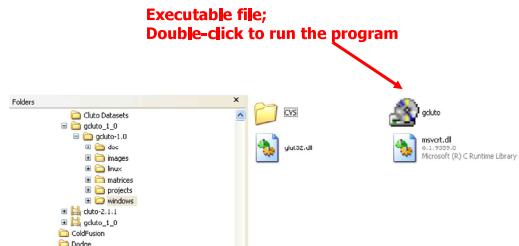
For more information on CLUTO (original non-graphical version), download and Unzip "cluto-2.1.1.zip" at:

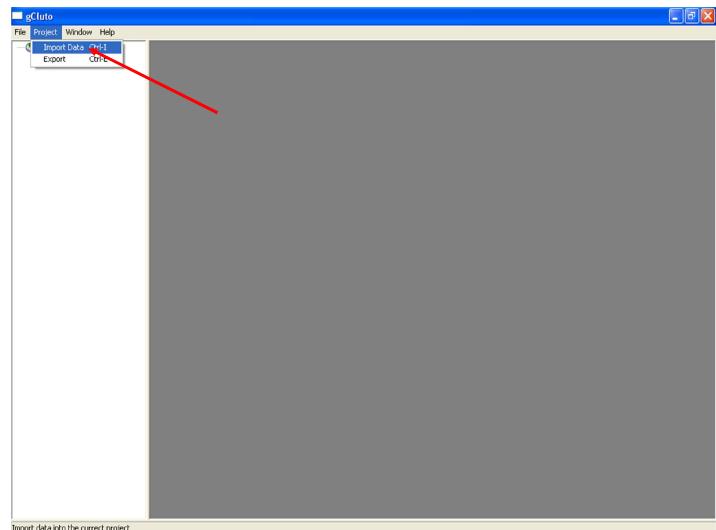
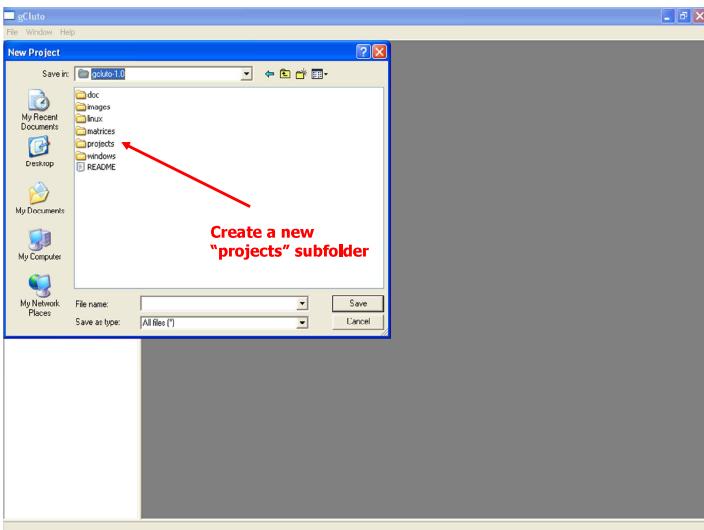
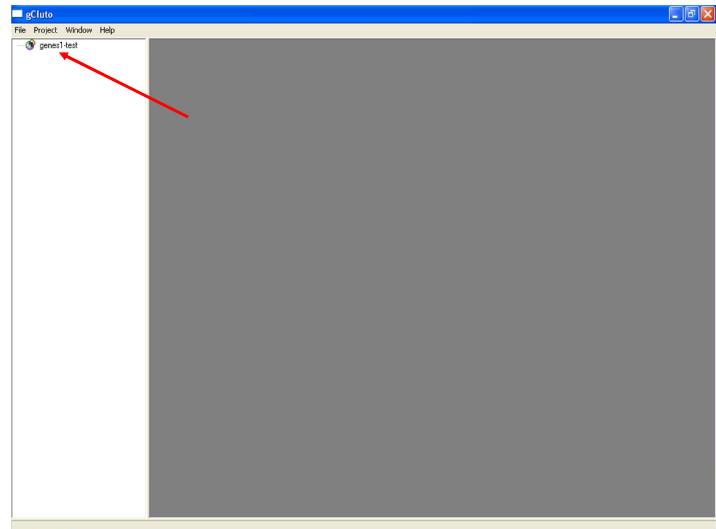
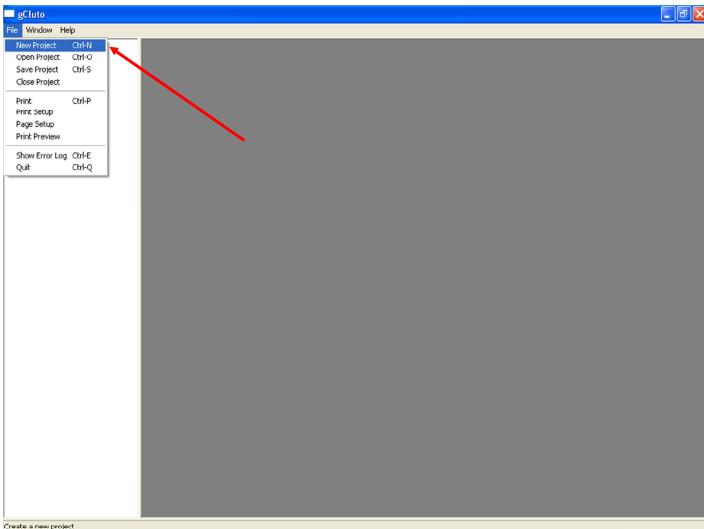
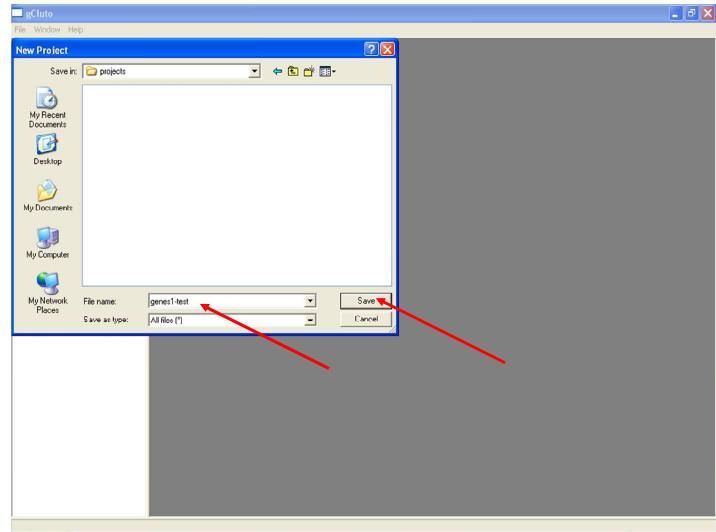
<http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download>

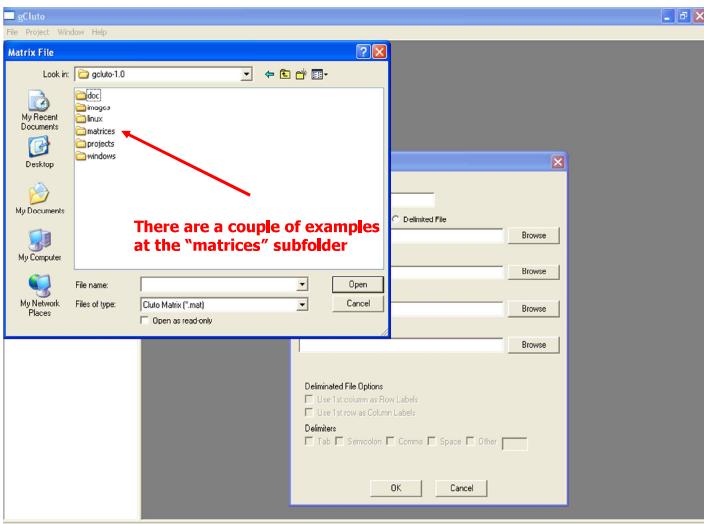
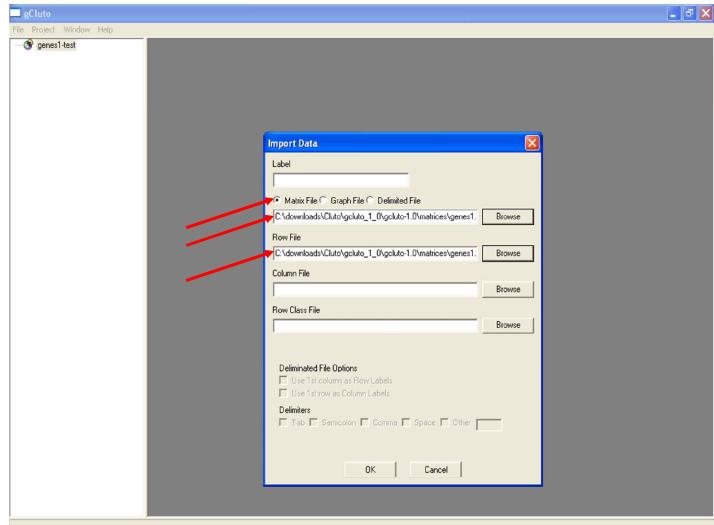
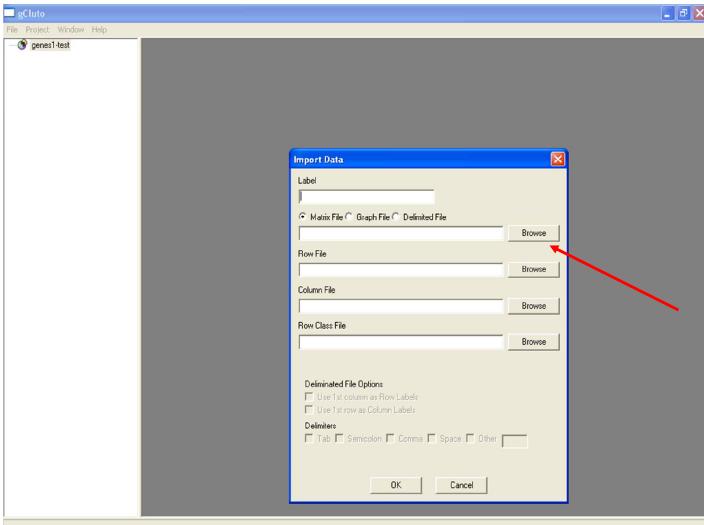
gCLUTO "doc" Subfolder and Files



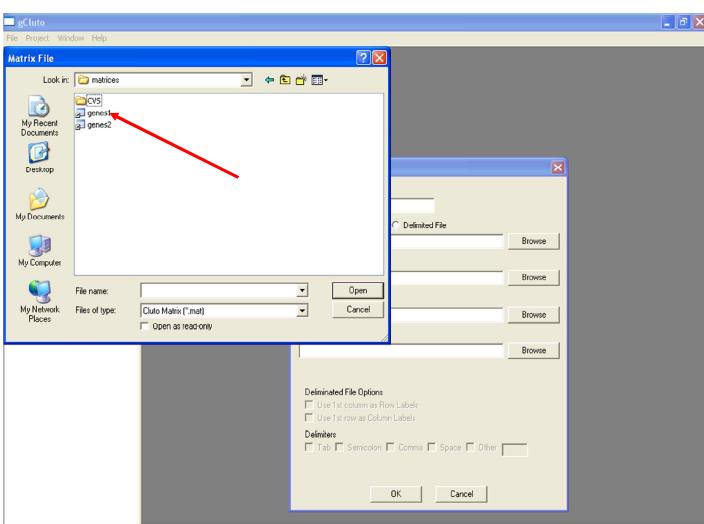
gCLUTO "windows" Subfolder and Files



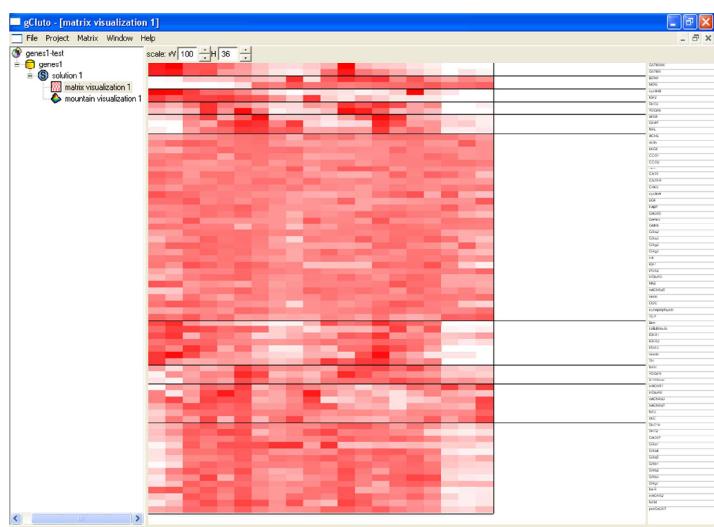
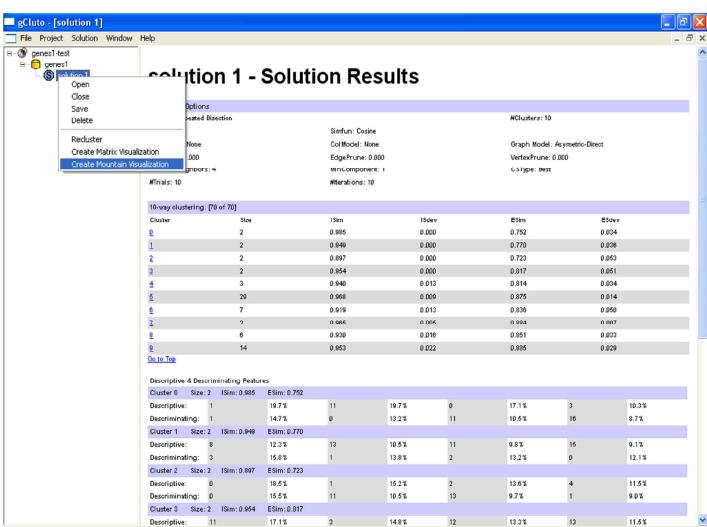
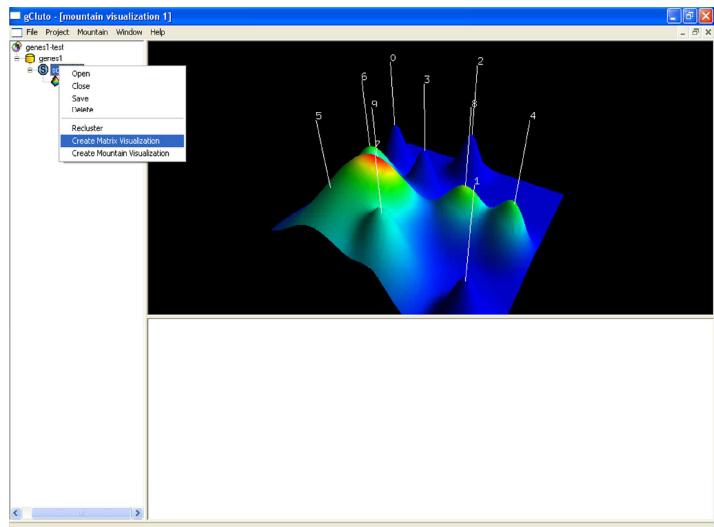
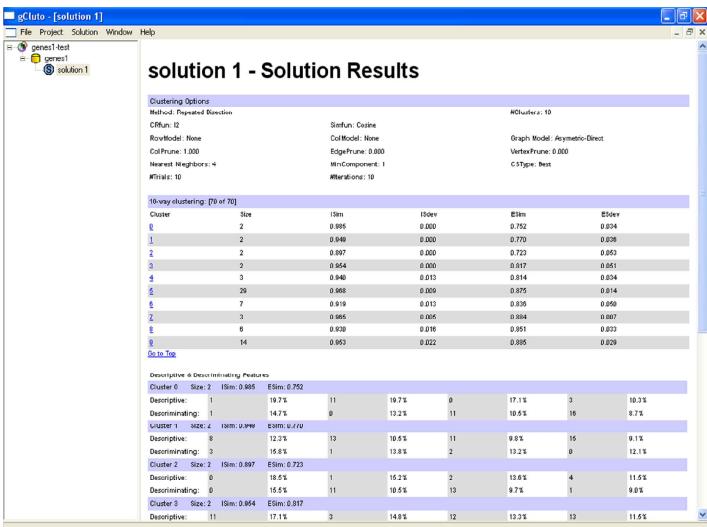
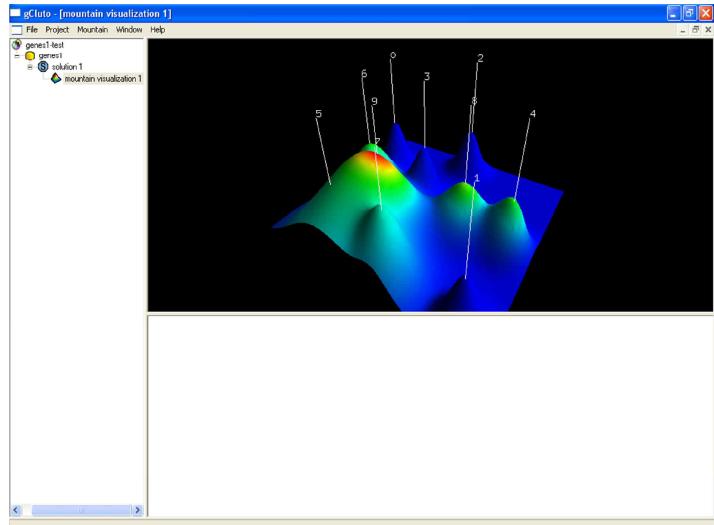
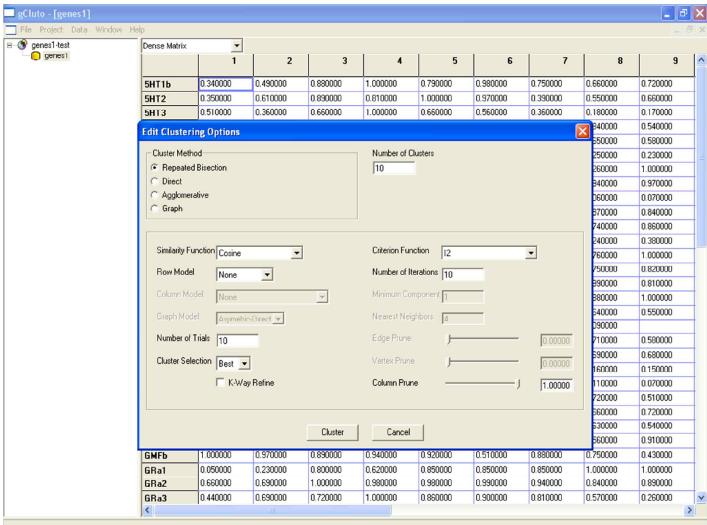




	1	2	3	4	5	6	7	8	9
SHT1b	0.340000	0.490000	0.880000	1.000000	0.730000	0.980000	0.750000	0.680000	0.720000
SHT2	0.350000	0.610000	0.890000	0.810000	1.000000	0.570000	0.390000	0.590000	0.690000
SHT3	0.510000	0.360000	0.660000	1.000000	0.660000	0.560000	0.360000	0.180000	0.170000
ACHE	0.570000	0.530000	0.750000	0.760000	0.830000	0.960000	1.000000	0.840000	0.540000
actn	0.590000	1.000000	0.930000	0.750000	0.850000	0.880000	0.890000	0.590000	0.580000
aFGF	0.160000	0.210000	0.530000	0.770000	0.320000	0.650000	1.000000	0.290000	0.230000
BDNF	0.020000	0.400000	0.290000	0.260000	0.280000	0.430000	0.330000	0.260000	1.000000
bFGF	0.630000	0.530000	0.850000	0.900000	1.000000	0.930000	0.950000	0.540000	0.970000
Bm	0.870000	1.000000	0.520000	0.360000	0.360000	0.230000	0.130000	0.030000	0.070000
CCX1	0.870000	0.520000	0.960000	0.950000	0.610000	1.000000	0.930000	0.870000	0.840000
CCO2	1.000000	0.890000	0.740000	0.700000	0.810000	0.820000	0.960000	0.740000	0.680000
cellulasevin	0.700000	1.000000	0.990000	0.920000	0.750000	0.820000	0.760000	0.240000	0.300000
cjan	0.710000	0.680000	0.670000	0.690000	0.690000	0.890000	0.700000	0.760000	1.000000
CNTF	1.000000	0.920000	0.880000	0.890000	0.910000	0.960000	0.910000	0.240000	0.620000
CNTFR	1.000000	0.900000	0.780000	0.830000	0.970000	0.820000	0.930000	0.980000	0.810000
CX43	0.890000	0.790000	0.980000	1.000000	0.930000	0.970000	0.980000	0.880000	1.000000
cyclinA	1.000000	0.890000	0.820000	0.720000	0.750000	0.610000	0.640000	0.590000	0.590000
cyclinB	1.000000	0.890000	0.770000	0.590000	0.590000	0.480000	0.280000	0.090000	0.070000
EGF	0.860000	0.020000	1.000000	0.720000	0.790000	0.680000	0.790000	0.710000	0.590000
FABP	0.730000	0.760000	0.880000	0.950000	0.980000	1.000000	0.890000	0.560000	0.680000
GG70909c	0.460000	1.000000	0.680000	0.680000	0.760000	0.410000	0.790000	0.680000	0.140000
GG70905	0.970000	1.000000	0.730000	0.770000	0.440000	0.420000	0.260000	0.110000	0.070000
GG70906	0.970000	1.000000	0.680000	0.680000	0.760000	0.440000	0.330000	0.260000	0.110000
GAD65	0.910000	0.830000	0.860000	0.970000	0.870000	1.000000	0.790000	0.510000	0.510000
GAD67	0.930000	0.600000	0.950000	0.830000	0.810000	1.000000	0.810000	0.690000	0.720000
GAP43	0.580000	0.690000	1.000000	0.630000	0.630000	0.640000	0.650000	0.530000	0.540000
GFAP	0.860000	0.890000	0.880000	0.890000	0.810000	1.000000	0.940000	0.910000	0.430000
GMFb	1.000000	0.970000	0.880000	0.940000	0.920000	0.510000	0.880000	0.750000	0.430000
GRA1	0.690000	0.230000	0.880000	0.620000	0.850000	0.850000	0.850000	1.000000	0.100000
GRA2	0.680000	0.680000	1.000000	0.980000	0.980000	0.990000	0.940000	0.840000	0.890000
GRA3	0.440000	0.690000	0.720000	1.000000	0.860000	0.860000	0.500000	0.570000	0.260000

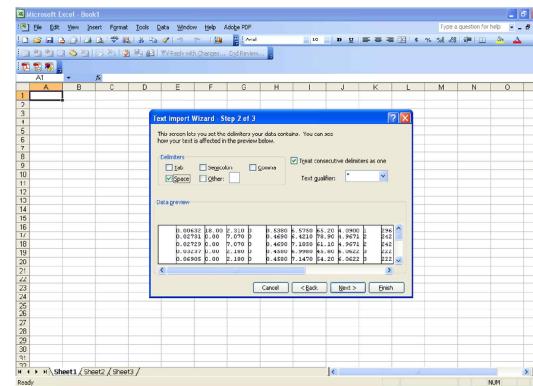


	1	2	3	4	5	6	7	8	9
SHT1b	0.340000	0.490000	0.880000	1.000000	0.730000	0.980000	0.750000	0.680000	0.720000
SHT2	0.350000	0.610000	0.890000	0.810000	1.000000	0.570000	0.390000	0.550000	0.690000
SHT3	0.510000	0.360000	0.660000	1.000000	0.660000	0.560000	0.360000	0.180000	0.170000
ACHE	0.570000	0.530000	0.750000	0.760000	0.830000	0.960000	1.000000	0.840000	0.540000
actn	0.590000	1.000000	0.930000	0.750000	0.850000	0.880000	0.890000	0.590000	0.580000
aFGF	0.160000	0.210000	0.530000	0.770000	0.320000	0.650000	1.000000	0.290000	0.230000
BDNF	0.020000	0.400000	0.290000	0.260000	0.280000	0.430000	0.330000	0.260000	1.000000
bFGF	0.630000	0.530000	0.850000	0.900000	1.000000	0.930000	0.950000	0.540000	0.970000
Bm	0.870000	1.000000	0.520000	0.360000	0.360000	0.230000	0.130000	0.050000	0.070000
CCX1	0.870000	0.520000	0.960000	0.950000	0.610000	1.000000	0.930000	0.870000	1.000000
CCO2	1.000000	0.890000	0.740000	0.700000	0.810000	0.870000	0.980000	0.720000	0.860000
cellulasevin	0.700000	1.000000	0.990000	0.920000	0.750000	0.820000	0.760000	0.240000	0.300000
cjan	0.710000	0.680000	0.670000	0.690000	0.690000	0.890000	0.700000	0.760000	1.000000
CNTF	1.000000	0.920000	0.880000	0.890000	0.910000	0.790000	0.810000	0.790000	0.830000
CNTFR	1.000000	0.900000	0.780000	0.830000	0.970000	0.830000	0.930000	0.990000	0.810000
CX43	0.890000	0.790000	0.980000	1.000000	0.900000	0.970000	0.980000	0.880000	1.000000
cyclinA	1.000000	0.890000	0.830000	0.720000	0.750000	0.710000	0.640000	0.640000	0.550000
cyclinB	1.000000	0.980000	0.770000	0.590000	0.610000	0.480000	0.290000	0.090000	0.070000
EGF	0.860000	0.670000	1.000000	0.720000	0.790000	0.690000	0.790000	0.710000	0.590000
FABP	0.730000	0.760000	0.830000	0.900000	0.890000	1.000000	0.850000	0.590000	0.690000
GG70909c	0.460000	1.000000	0.680000	0.680000	0.590000	0.410000	0.290000	0.160000	0.190000
GG70905	0.970000	1.000000	0.730000	0.770000	0.440000	0.420000	0.260000	0.110000	0.070000
GG70906	0.970000	1.000000	0.680000	0.680000	0.590000	0.410000	0.260000	0.110000	0.070000
GAD65	0.910000	0.830000	0.880000	0.970000	0.870000	1.000000	0.790000	0.720000	0.510000
GAD67	0.930000	0.460000	0.950000	0.030000	0.010000	1.000000	0.610000	0.600000	0.720000
GAP43	0.580000	0.650000	1.000000	0.630000	0.640000	0.540000	0.530000	0.540000	0.910000
GFAP	0.860000	0.530000	0.280000	0.810000	0.940000	1.000000	0.940000	0.560000	0.910000
GMFb	1.000000	0.970000	0.890000	0.940000	0.920000	0.510000	0.880000	0.750000	0.430000
GRA1	0.690000	0.230000	0.880000	0.620000	0.850000	0.850000	1.000000	0.860000	1.000000
GRA2	0.680000	0.690000	1.000000	0.980000	0.980000	0.990000	0.940000	0.840000	0.890000
GRA3	0.440000	0.690000	0.720000	1.000000	0.860000	0.500000	0.810000	0.570000	0.260000



In this data set “space” is the delimiter
That's not always the case

Using other data sets



Check the data

- make sure that all columns are complete
- preprocess the data, if needed
- eliminate unnecessary columns

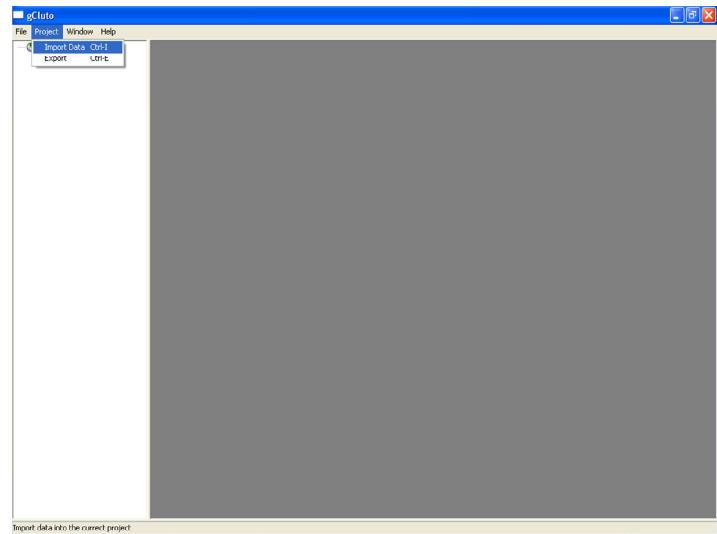
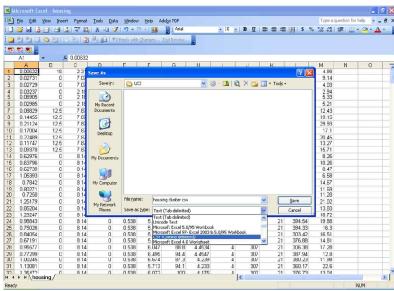
Preparing the data

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	206	16.2	305.3	44.36	24	
2	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	240	16.2	305.3	44.36	24	
3	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	242	17.8	305.3	43.32	24	
4	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	242	17.8	305.3	43.32	24	
5	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
6	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
7	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
8	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
9	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
10	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
11	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
12	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
13	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
14	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
15	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
16	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
17	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
18	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
19	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
20	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
21	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
22	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
23	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
24	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
25	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
26	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
27	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
28	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
29	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	
30	0.00032	18	2.31	0	0.0000	5.0750	35.21	4.00	1	222	16.1	306.4	43.32	24	

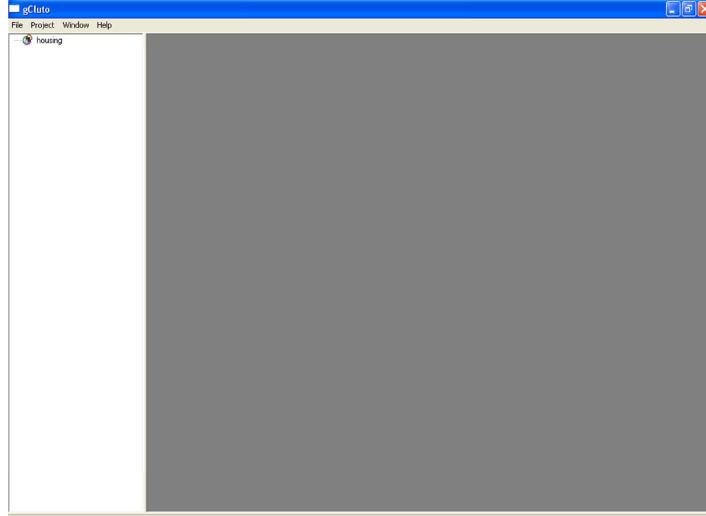
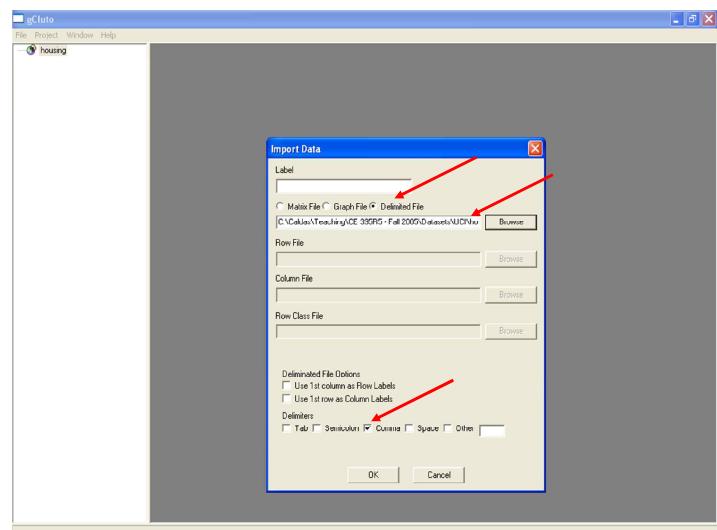
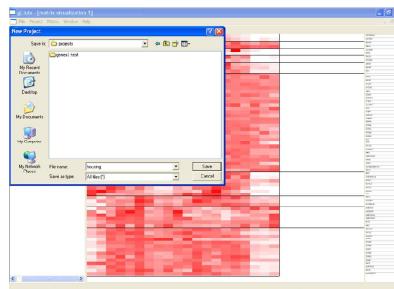
Opening “housing.dat” in Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2															
3															
4															
5															
6															
7															
8															
9															
10															
11															
12															
13															
14															
15															
16															
17															
18															
19															
20															
21															
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															
32															
33															
34															
35															
36															
37															
38															
39															
40															
41															
42															
43															
44															
45															
46															
47															
48															
49															
50															
51															
52															
53															
54															
55															
56															
57															
58															
59															
60															
61															
62															
63															
64															
65															
66															
67															
68															
69															
70															
71															
72															
73															
74															
75															
76															
77															
78															
79															
80															
81															
82															
83															
84															

Save the file as CSV



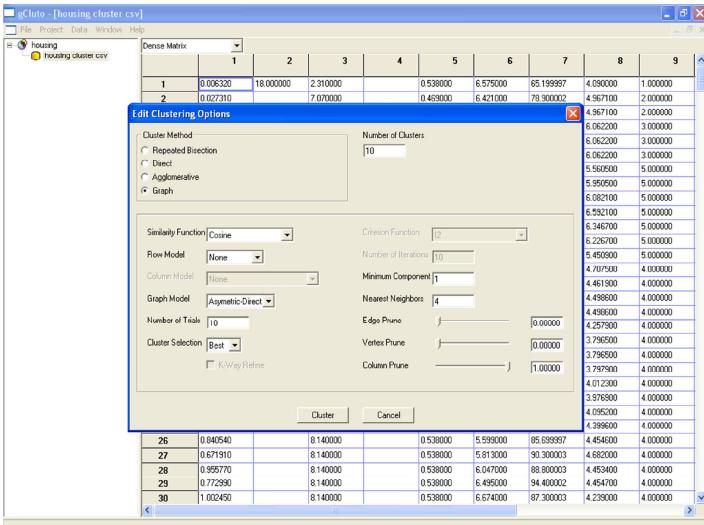
Create a new project in gCLUTO



gCLUTO - [housing cluster.csv]

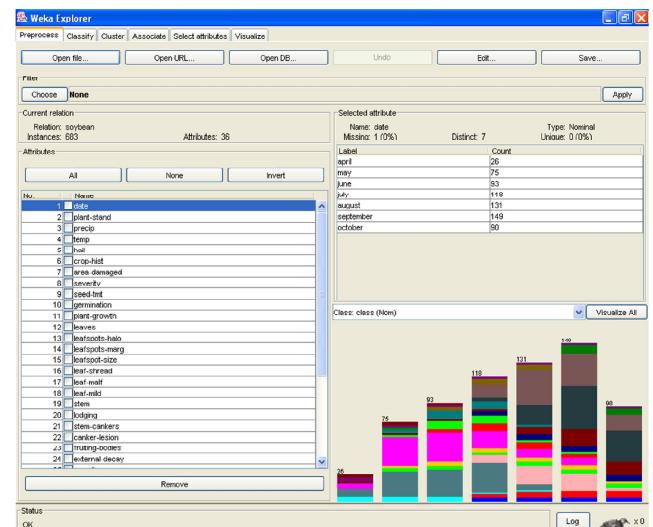
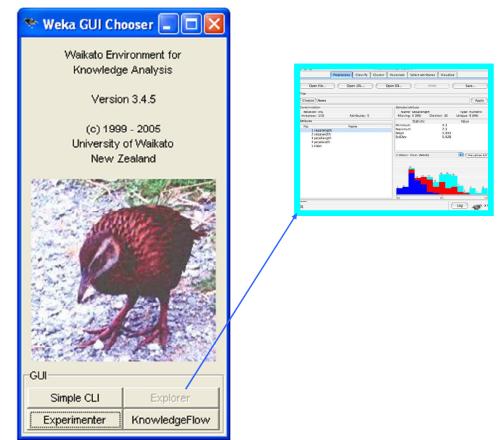
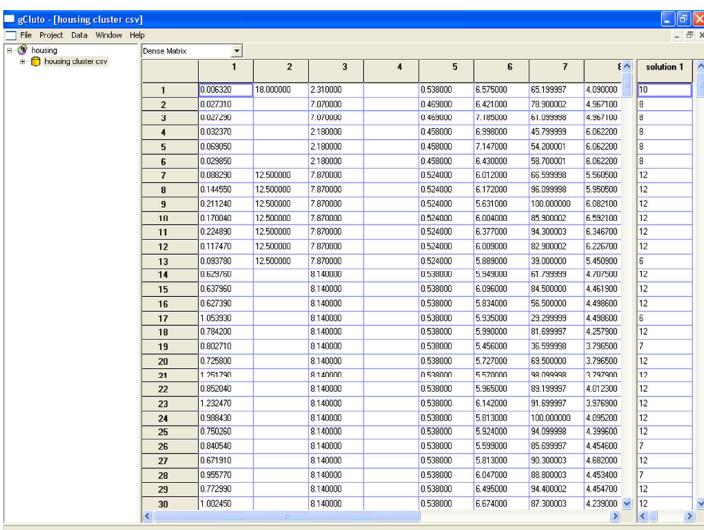
Dense Metric

	1	2	3	4	5	6	7	8	9
1	0.006320	18.000000	2.310000		0.538000	6.575000	65.199957	4.090000	1.000000
2	0.027310		7.070000		0.469000	6.421000	73.000022	4.957100	
3	0.027290		7.070000		0.469000	7.185000	61.099958	4.957100	2.000000
4	0.023270		2.180000		0.458000	6.980000	45.799959	6.052200	3.000000
5	0.069050		2.180000		0.458000	7.147000	54.200001	6.052200	
6	0.028650		2.180000		0.458000	6.430000	58.700001	6.062200	3.000000
7	0.089290	12.500000	7.870000		0.524000	6.012000	65.599958	5.595000	5.000000
8	0.144550	12.500000	7.870000		0.524000	6.172000	36.999958	5.595000	5.000000
9	0.211240	12.500000	7.870000		0.524000	5.631000	100.000000	6.082100	5.000000
10	0.170940	12.500000	7.870000		0.524000	6.044000	85.300022	6.532100	5.000000
11	0.224890	12.500000	7.870000		0.524000	6.377000	94.300003	6.346700	5.000000
12	0.117470	12.500000	7.870000		0.524000	6.059000	92.300022	6.226700	5.000000
13	0.093790	12.500000	7.870000		0.524000	5.889000	39.000000	5.459000	5.000000
14	0.625970		81.400000		0.538000	5.943000	61.799957	4.707000	4.000000
15	0.637960		81.400000		0.538000	6.056000	84.500000	4.461900	4.000000
16	0.627390		81.400000		0.538000	5.934000	55.500000	4.498600	4.000000
17	1.053930		81.400000		0.538000	5.936000	29.266999	4.498600	4.000000
18	0.794200		81.400000		0.538000	5.990000	81.599957	4.257900	4.000000
19	0.802710		81.400000		0.538000	5.456000	35.599957	3.795000	4.000000
20	0.725600		81.400000		0.538000	5.727000	63.500000	3.795000	4.000000
21	1.251730		81.400000		0.538000	5.570000	98.999957	3.797900	4.000000
22	0.852040		81.400000		0.538000	5.965000	83.199957	4.012300	4.000000
23	1.232470		81.400000		0.538000	6.142000	91.599957	3.757600	4.000000
24	0.989430		81.400000		0.538000	5.013000	100.000000	4.095200	4.000000
25	0.769360		81.400000		0.538000	5.934000	94.599957	4.396600	4.000000
26	0.906450		81.400000		0.538000	5.599000	85.599957	4.464600	4.000000
27	0.671910		81.400000		0.538000	5.812000	93.300003	4.682000	4.000000
28	0.955770		81.400000		0.538000	6.047000	98.800003	4.453400	4.000000
29	0.772930		81.400000		0.538000	6.495000	94.300022	4.454700	4.000000
30	1.002450		81.400000		0.538000	6.674000	97.300003	4.239000	4.000000

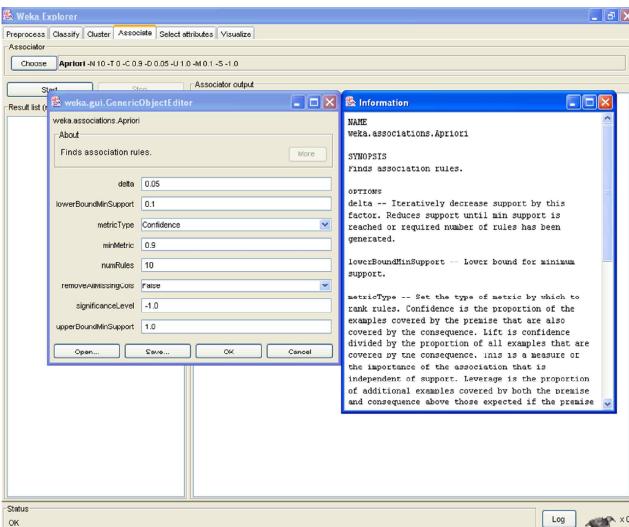
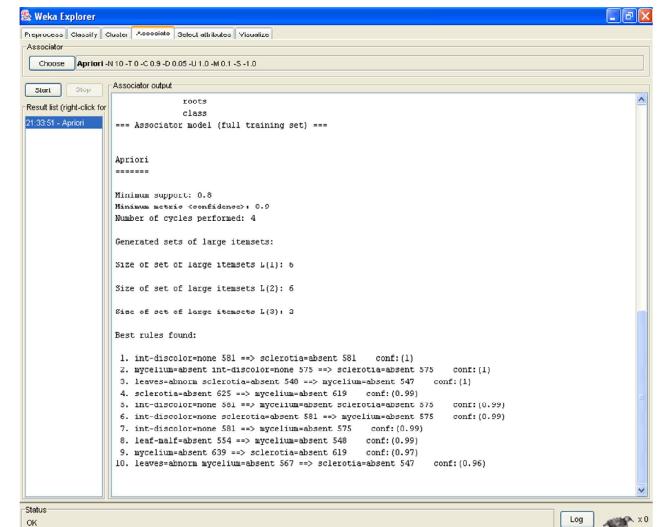
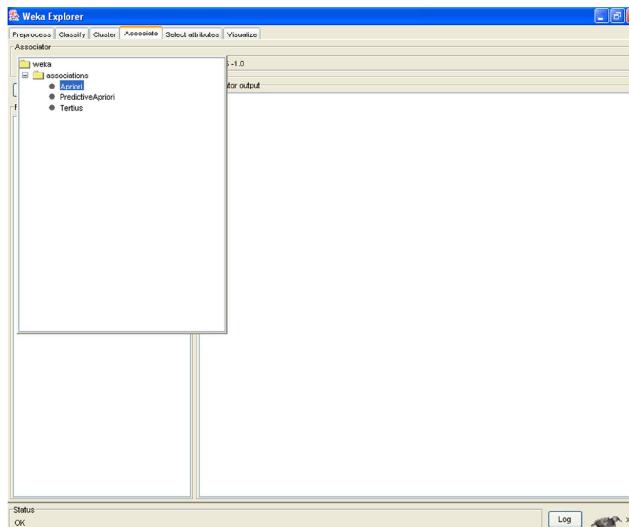
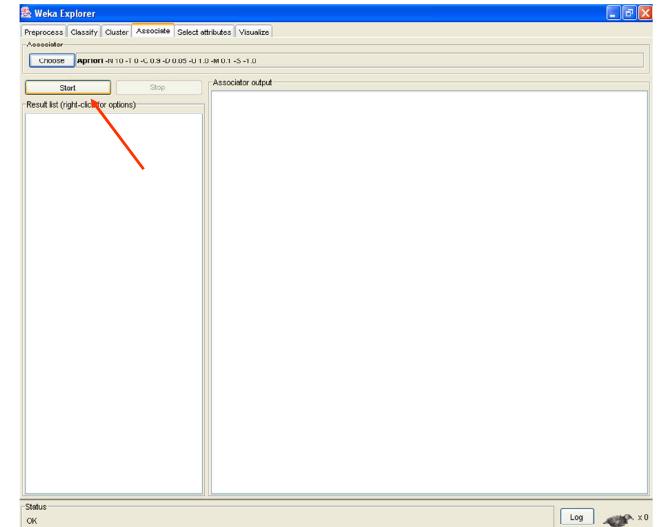
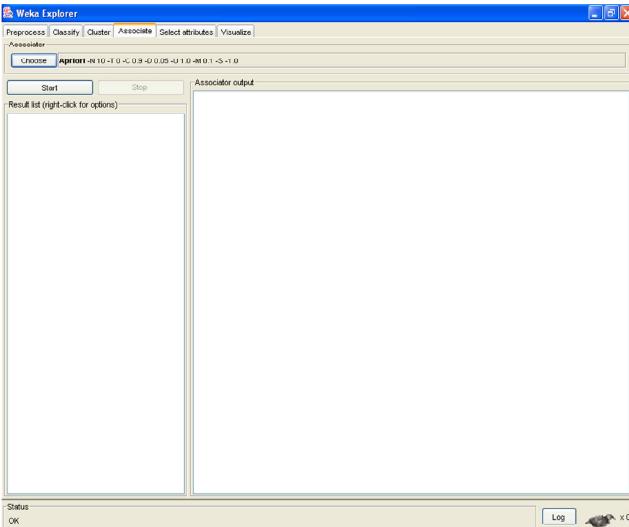


Explorer: finding associations

- WEKA contains an implementation of the Apriori algorithm for learning association rules
 - Works only with discrete data
- Can identify statistical dependencies between groups of attributes:
 - milk, butter \Rightarrow bread, eggs (with confidence 0.9 and support 2000)
- Apriori can compute all rules that have a given minimum support and exceed a given confidence



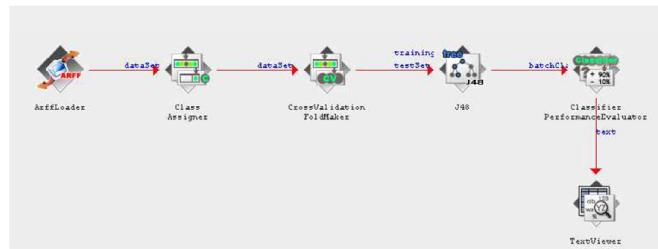
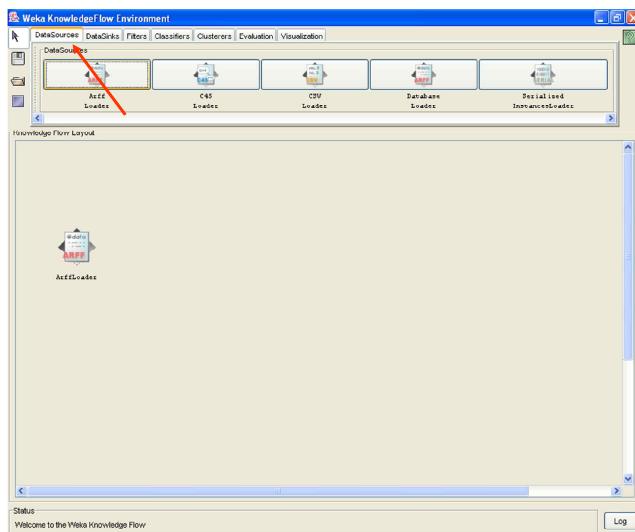
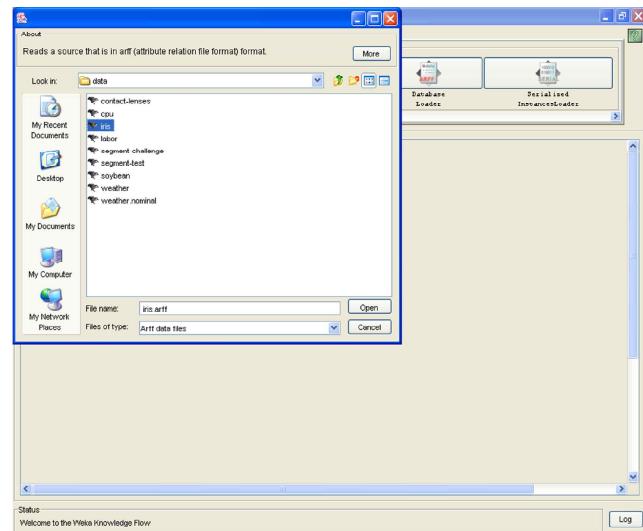
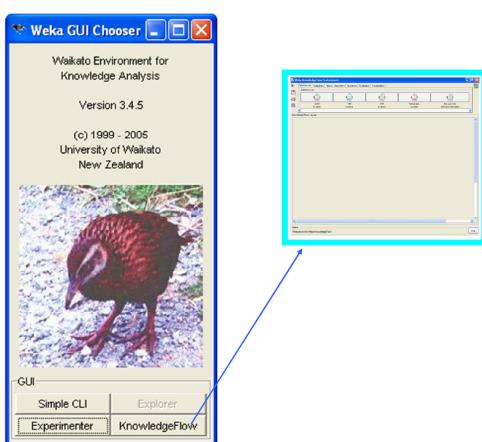
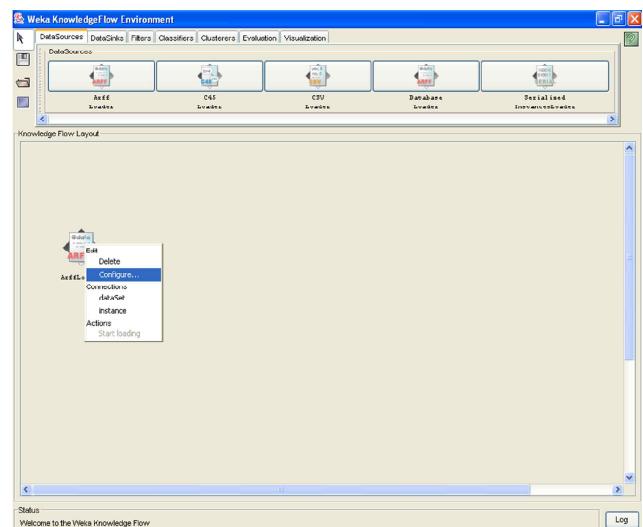
Data Mining with WEKA (Mining Association Rules)

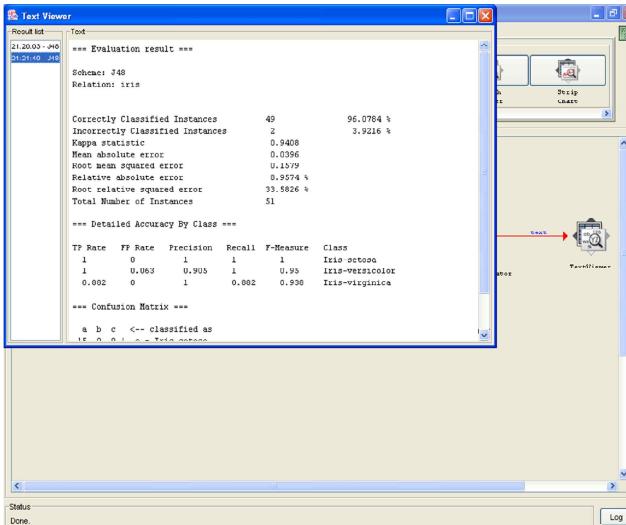


Data Mining with WEKA (Knowledge Flow GUI)

The Knowledge Flow GUI

- Graphical user interface for WEKA
- Java-Beans-based interface for setting up and running machine learning experiments
- Data sources, classifiers, etc. are beans and can be connected graphically
- Data “flows” through components: e.g., “data source” -> “filter” -> “classifier” -> “evaluator”
- Layouts can be saved and loaded again later





Introduction

The KnowledgeFlow redesigns an alternative to the E-lorer as a graphical interface. The end-to-WEKA core algorithm is the KnowledgeFlow is a work in progress so some of the functionality from the E-lorer is not yet available at the moment and there are things that are done in the KnowledgeFlow but not in the E-lorer.



The KnowledgeFlow presents a *data-flow* oriented interface to WEKA. The user can select the WEKA components from a toolbar at the top of the interface and connect them together in order to form a knowledge flow for processing and analyzing data. At present, all of WEKA's classifiers, filters, learners, and loaders and save as a file in the KnowledgeFlow along with some extra tools.

To handle data in relational databases, KnowledgeFlow provides an interface called `RelationalDataHandler`. This interface defines methods for reading data from a database and writing data back to it. The `RelationalDataHandler` interface requires a classifier that can be updated on an instance-by-instance basis. Currently, in WEKA there are ten classifiers that can handle data in relational databases.

- A E
 -
 -
 - K tar
 - ai e a es ultimo ial datea
 - ai e a es datea le
 - ge
 - Winnow

- Winnow

And two of them are meta classifiers:

- a cd n re ental ogit oost - t at an use of an reg to learn fro dis re lass data in re entall
- W - lo all weig ted learning

Feature

The KnowledgeFlow offers the following features:

- intuitive data flow style layout
 - to res data in at es or in re entall
 - ro ess ult le at es or strea sin araled ea se ate flow executes in its own t read
 - chain filters together
 - view models produced by classifiers for each fold in a cross validation
 - visualize performance of incremental classifiers during pre osing: s rolling plots of classification accuracy, RMS error, predictions et
 - begin filt for allowing eas addition of new o ments to t e Knowl-edgeFlow


 THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

WEKA KnowledgeFlow Tutorial
for Version 3-5-8

ar all
eter eute ann
ul 8

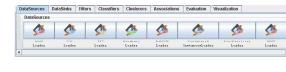


3 components

o onents available in the KnowledgeFlow

3 ata our e

All of WEKA's loaders are available.



All of WEKA's sa

DataSources DataSinks Filters Classifiers Clean



3.4 Classification

3.4 Classifiers

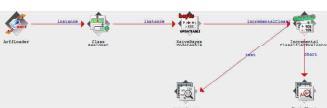


4

3.5 lusterers



Some classifiers, clusterers and filters in Weka can handle data in real time. In a streaming fashion there is an example of training and testing naïve Bayes classifier. The results are sent to a TestViewer and predictions are plotted in a tripartite component.



- l i s t e n t o a t a o u r e s t a n d o n e **Afflooder** fro m t o o l a r t e
 - Next place the **Afflooder** component on the layout area by clicking so where on the layout (a copy of the **Afflooder** icon will appear on the layout area).
 - Next specify an **AMF** file to load by first right-clicking the mouse or the **Afflooder** icon on the layout. A pop-up menu will appear. Select **Configure** under Edit in the context menu and route to el o ato your **AMF** file.
 - e t h i c t e a l u a t i o n t a t t e c t o f e w i n d o w a n d o s s e t t e l a s s i g n e r a l l o w s o u t t o o w s i l u n t o e t e l a s s i g n e r o n f r o m t o o l a r t e i n t h e l a t o u t
 - Now connect the **Afflooder** to the **ClassAssigner**: first right click on the **Afflooder** and select the data under **outputs** in t e c h n i c a l and line will a car t e s e t o n e o n e **ClassAssigner** o n e and let h a red line the el a data and will su n t e t o o o n e
 - e t r i g h t h i l o c t e l a s s i g n e r a n d o n e **Configure** fro m t e e n u T i s w i l l o u t o w h i f o r m o n a s c i f i w l o n i s t e l a s in our data last is t e def aul
 - o n g r a a c i a e s d a t a s l e o n f r o m t e a c e s se t o n t e t **Classifiers** and la e i t o n t e l a out
 - e t o n t e t l a s s i g n e r a c i a e s d a t a s le t o u s t a n e i s t o n e t o n
 - e t h e a n n o n e r a l l a s s i g n e r a l a t o r t e e a l u a t i o n a n d o n t e l a e a n d o n e t a s i c a e s d a t a le t o i t u s i n g a i n re medialClassifier one t o n

eferen es

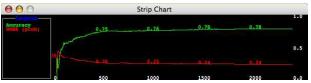
Witten and Fran E. Mataining rational active learning.

ing tools and techniques 2nd edition organ Kaufmann

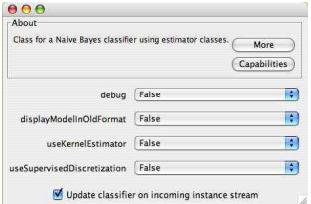
We a o tt we a sour e forge net we ado

3 We awi i tt we a sour efor ge net we awi i
lotting ulti le ur es on We awi i
tt we a sour efor ge net wi i lind lotting ulti le ur es

- e t a l e a T e V i e w e r o n e n t f r o m e V i s u a l i z a t i o n a n d o n t e a o u t o n n e t h e *IncrementalClassifierEvaluator* to it using a te t o n n i c e
 - e t a l e a T r a i n e r o n e n t f r o m e V i s u a l i z a t i o n a n d o n t e a o u t a n d o n n e *IncrementalClassifierEvaluator* to it using a t o n n i c e
 - i s t h e t r a i n e r s a r t r i g g e r i n g o r i t a n d o n n i n g o w a r t h o t e r o u t e s o n n e t h e t r a i n e r c a n b e c o n f i g u r e d w i t h o n t a t o n t r a i n o n f e a t u r e s o n t a i n i n g t h e a l s o d i s c a d
 - Finally, start the flow by right-clicking over the *ArtifactLoader* and selecting *Start Task* from the context menu.



ote at int is a le re di ran is o tained fro nai e as for ea n in insta nce before the classifier is trained (updated) with the instance. If you have a pre-trained classifier, you can specify that the classifier not be updated on incoming instances unless tuning the configuration dialog for the classifier. If the pre-trained classifier is a neural classifier (i.e. it is not a linear model), then you will only be able to test it in an internal function.



5 lugin Fa ilit

Beans.props file:

```

gnus    all  ls -l   E  nowledgeFlow lugins  ettle
total
-rw-r--r--    all  all      Fe      5  cans  ro s
-rw-r--r--    all  all     8      Fe          ettleKF ar

gnus    all  ar t f   sers  all  nowledgeFlow lugins  ettle

```

Wd Fe 3 T S ETA- F
Wd Fe 3 T S ETA- F A F E T F

Tue Fe 5 8 T 8 we a
Tue Fe 5 8 T 8 we a gui
we a

Wed Fe 3 55 5 T 8 we a gui cans
Wed Fe 3 5 3 T 8 we a gui cans i ons

S Wd Fe T S we a - gui eans i ons Kettle n u
S Wd Fe S T S we a - gui eans i ons Kettle n u

83 Wed Fe 3 5 8 T 8 we a gui cans Kettle nut last
Tue Fe 5 T 8 we a gui cans Kettle nut can

gnus all ore sers all now edgeFlow lugins ettle can

gnus are servers and now easier to plug in. The little example specifies the tools to go into the plug-in toolbar.

We can now use the KnowledgeFlow plug-in to create a GUI application.

we a gun cans know edges now fugitives we a gun cans kettle nut