Introduction to Weka

Daniel Rodríguez, University of Alcalá

Universidad de Alcalá

OXFORD BROOKES UNIVERSITY

Weka

Weka Toolkit

- http://www.cs.waikato.ac.nz/ml/weka/

Developed in Java and Open source

- GNU Licence

Well supported by the community through mailing lists and Wiki:

- http://weka.wikispaces.com/

Also, supported commercially by Pentaho

- http://weka.pentaho.com/
- Acknowledgement: J Hernandez & C Ferri (UPV) for some some examples

Weka Tookit

Composed of several tools

- GUI Chooser
  - Package manager,
    Visualisation and Viewers
- Explorer (the one to use most of the time)
- Experimenter
- KnowlegeFlow
- SimpleCLI
  - (Command Line Interface) - Obsolete

ARFF, XRFF, and sparse formats

ARFF (Attribute-Relation File Format) is Weka's native file format

- Composed of Header and Data
  - Header of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types (numeric, nominal, date).
  - Data is compsed of comma separated values after @data

XRFF (eXtensible attribute-Relation File Format) is an XML-based extension of the ARFF format. It has the advantage of the following additional capabilities:

- class attribute specification
- attribute weights (So far, only Naïve Bayes can handle attribute weights)

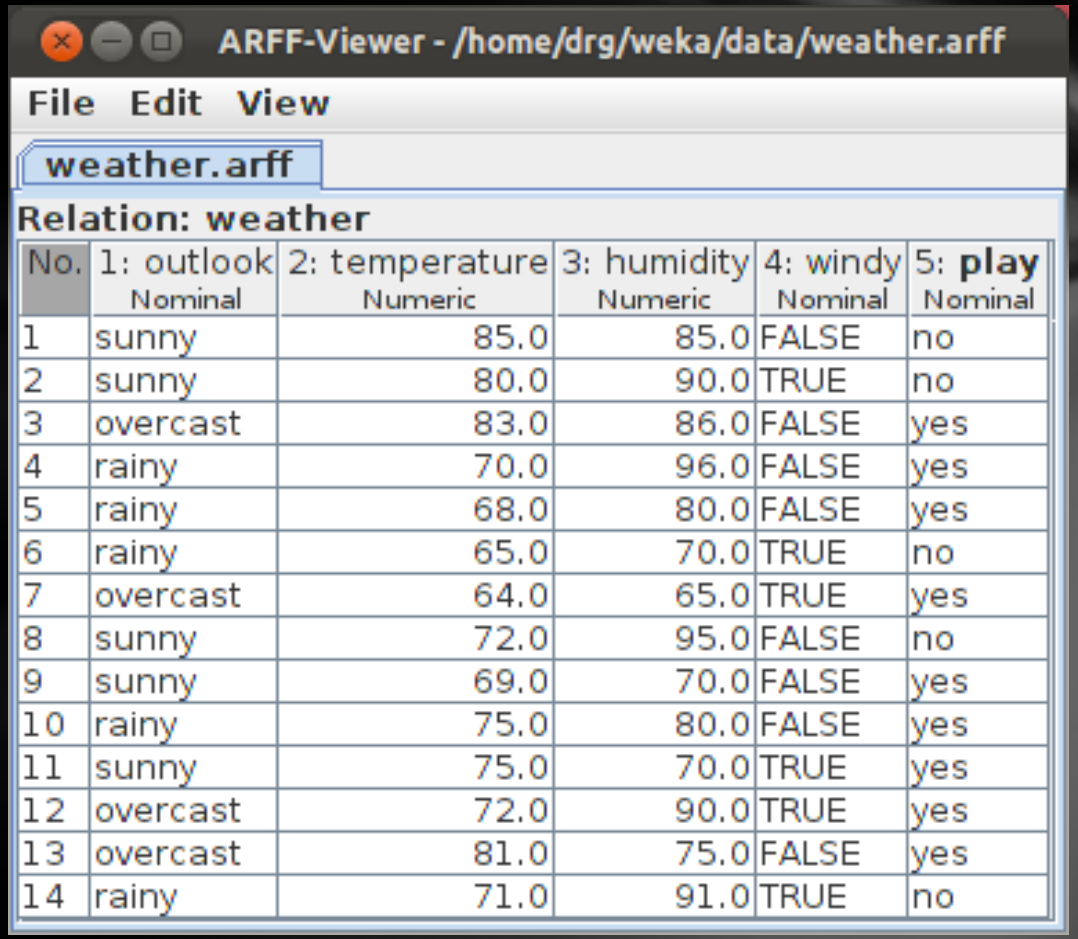
  and instance weights are also supported in ARFF de
- It can be compressed using the xrff.gz extension

Both ARFF and XRFF formats support sparse data when most data values are zeros

- (e.g. supermarket dataset)

ARFF Example

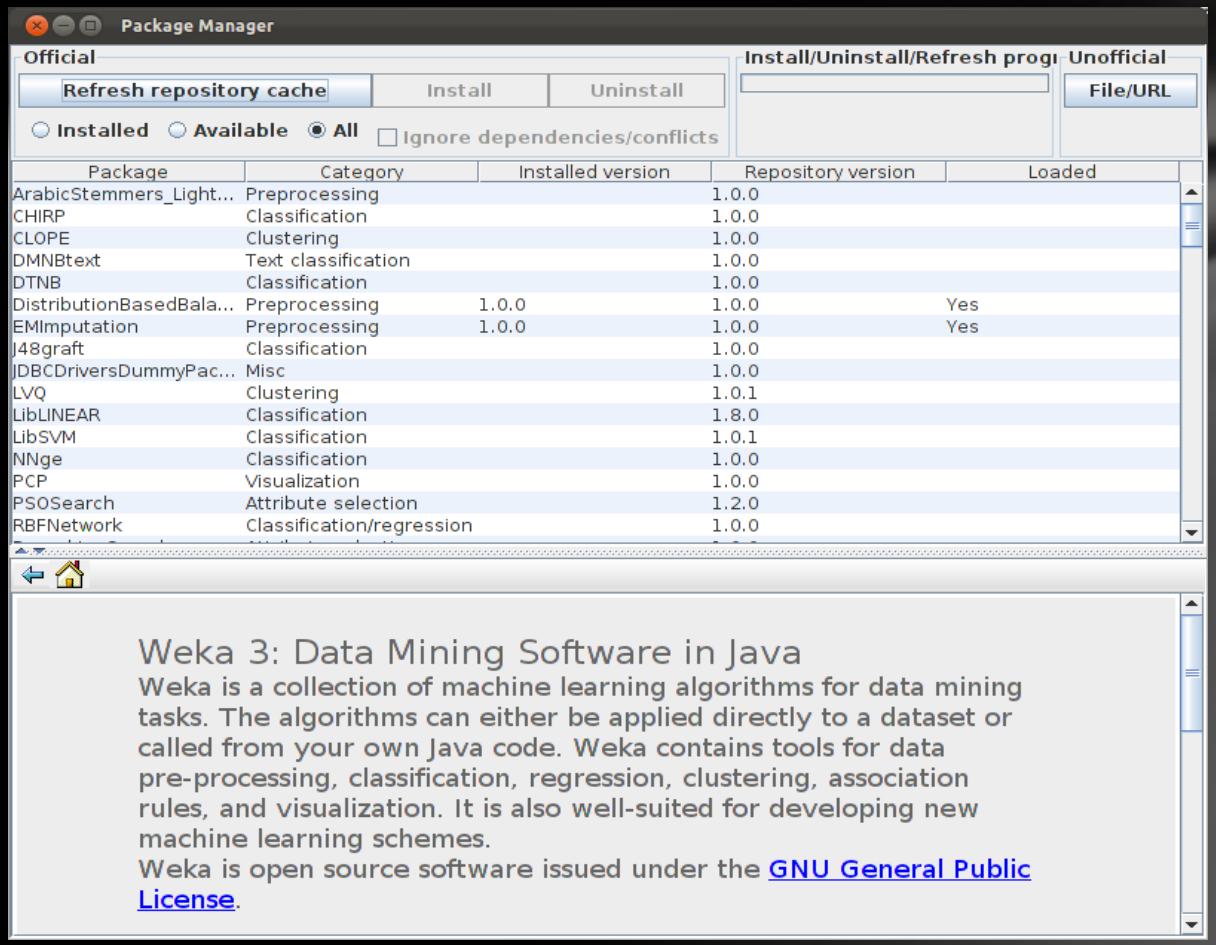
```
@relation weather % Comment

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no
```



No.	1: outlook Nominal	2: temperature Numeric	3: humidity Numeric	4: windy Nominal	5: play Nominal
1	sunny	85.0	85.0	FALSE	no
2	sunny	80.0	90.0	TRUE	no
3	overcast	83.0	86.0	FALSE	yes
4	rainy	70.0	96.0	FALSE	yes
5	rainy	68.0	80.0	FALSE	yes
6	rainy	65.0	70.0	TRUE	no
7	overcast	64.0	65.0	TRUE	yes
8	sunny	72.0	95.0	FALSE	no
9	sunny	69.0	70.0	FALSE	yes
10	rainy	75.0	80.0	FALSE	yes
11	sunny	75.0	70.0	TRUE	yes
12	overcast	72.0	90.0	TRUE	yes
13	overcast	81.0	75.0	FALSE	yes
14	rainy	71.0	91.0	TRUE	no

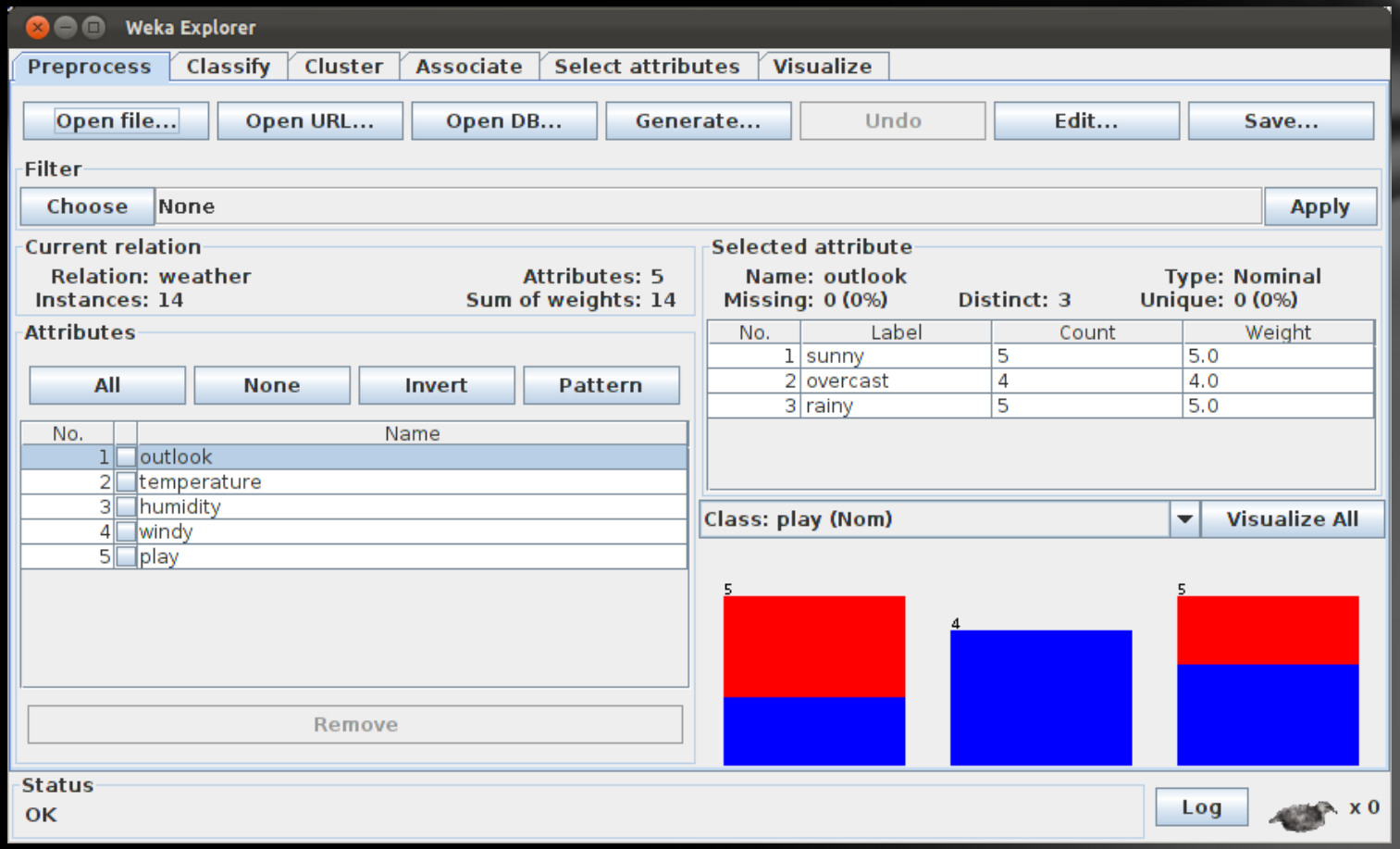
Package Manager (version > 3.7.2)

Just the most common algorithms are included with the download but others can be installed via the Package Manager.



Explorer

The one to use most of the time. It can be used to find the most appropriate algorithm and parameters for a given dataset (usually trial & error approach)



Explorer's panels

Explorer has 6 panels to analyse and prepare data:

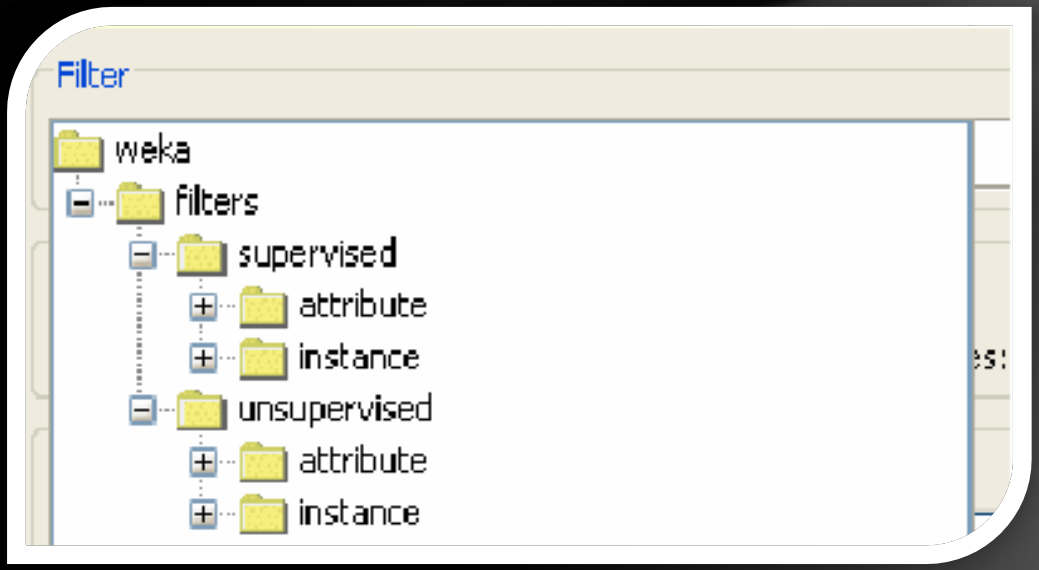
- **Preprocess**: Tools and filters for data manipulation
- **Classification**: Classification and regresión techniques
- **Cluster**: Clustering techniques
- **Associate**: Association techniques
- **Select Attributes**: Permite aplicar diversas técnicas para la reducción del número de atributos
- **Visualize**: Visualisation techniques
- **Other panels** can be added (e.g., Time series analysis)

Preprocess Panel

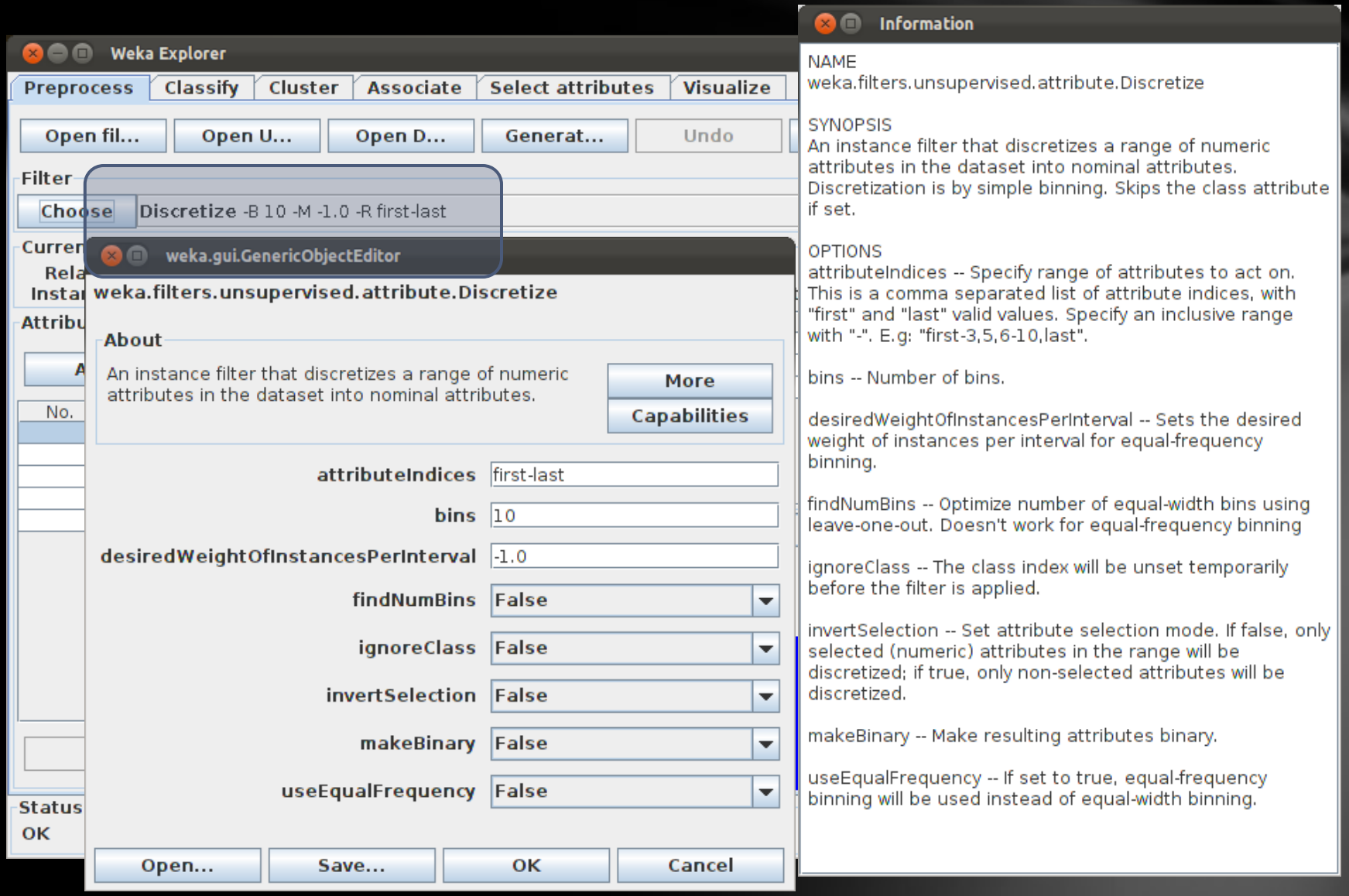
Filter algorithms are classifed as

- Supervised – if the filter takes into account the class
- Unsuvervised – otherwise
- An in turn they are also classifed according to whether the filter applies to instances or attributes

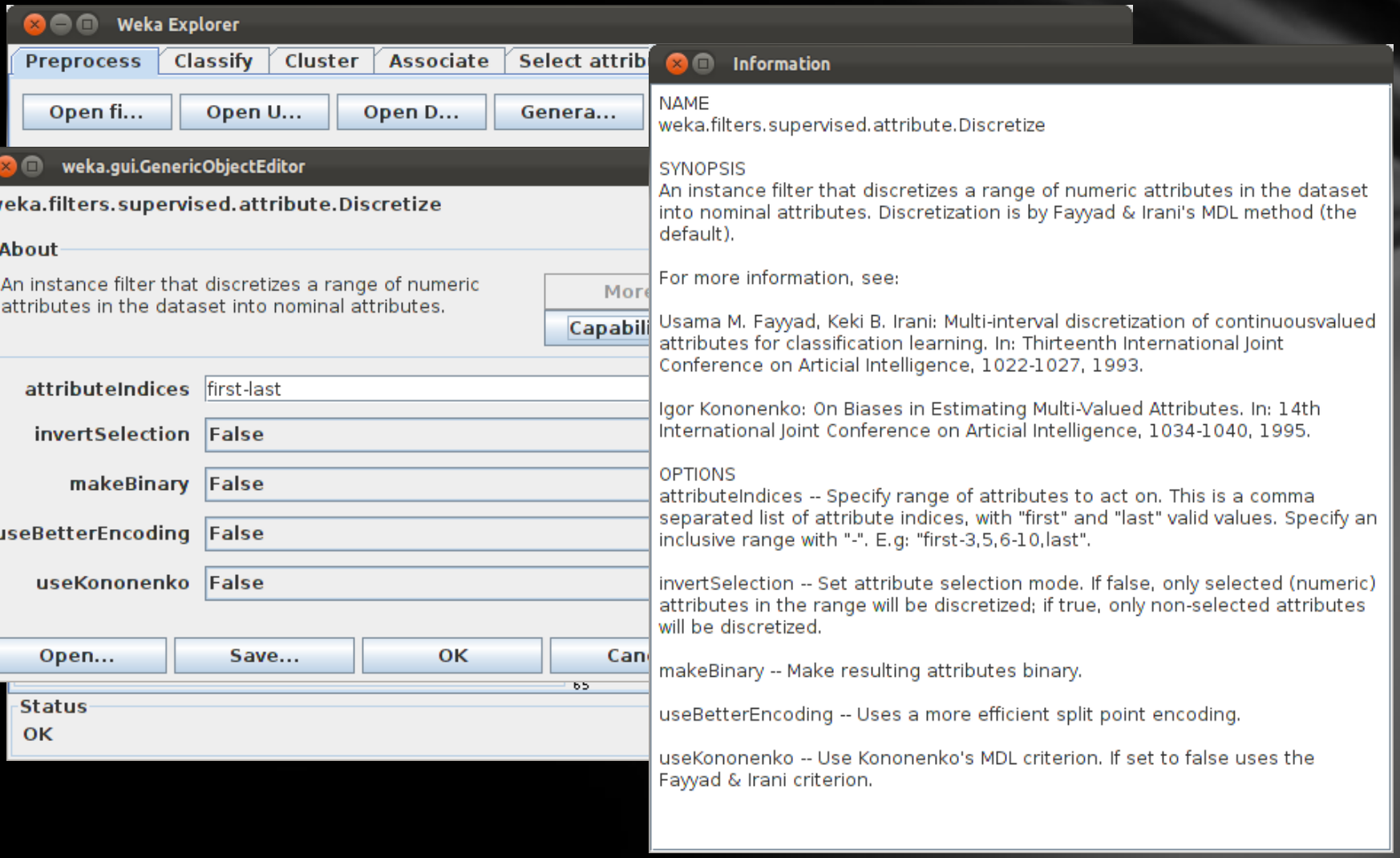
Filter examples include discretisation (can be supervised or unsupervised, attribute selection (we will cover this one later in detail), etc.



Unsupervised Discretize Filter Example



Discretize Supervised Filter Example



‘Classify’ Panel

Classification and regression Techniques:

Decission Trees: ID3, C4.5 (J48), …

Regresssion Trees: LMT (M5), …

Rules: PART, CN2, …

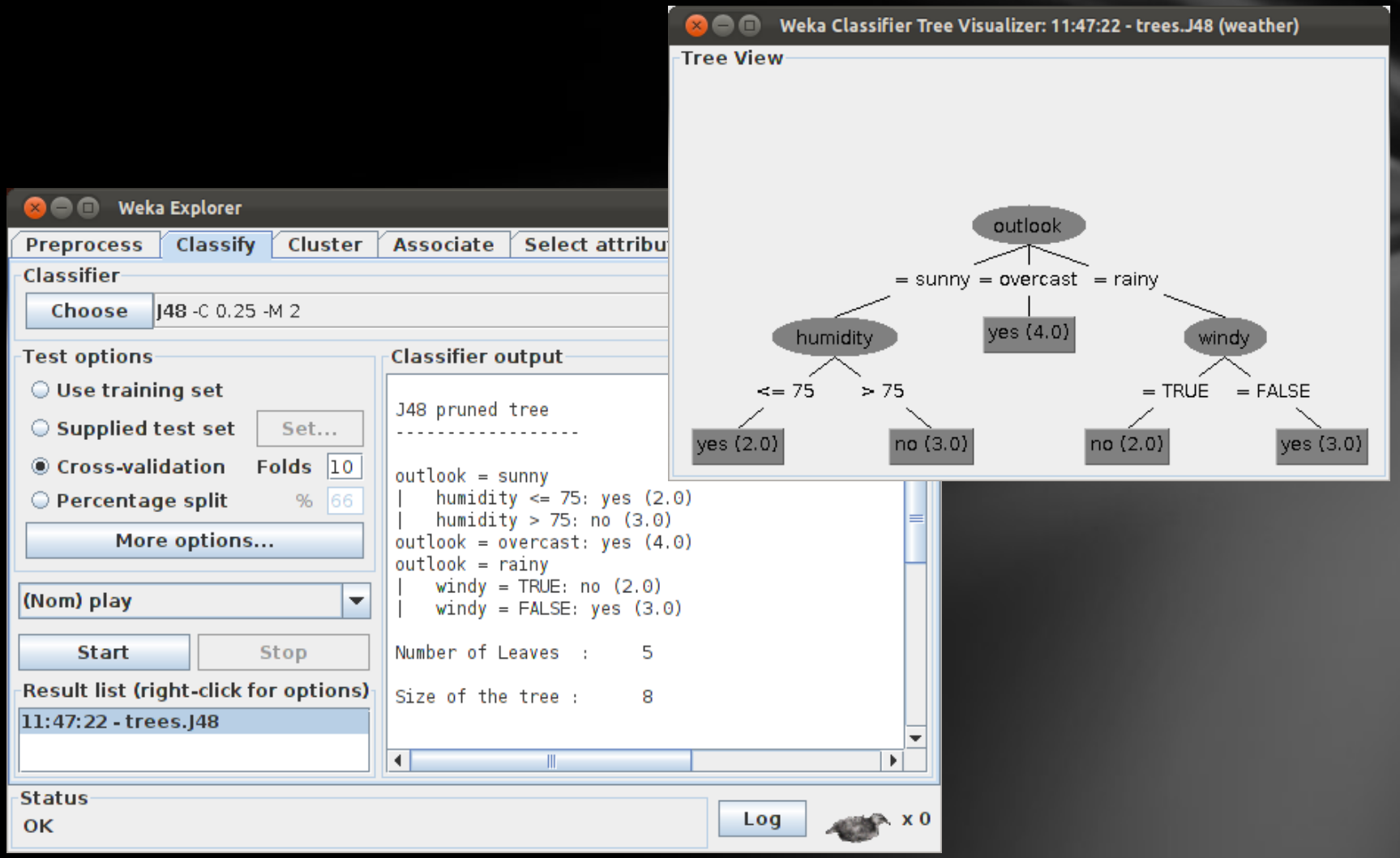
Functions: Regression, Neural Networks, logistic regression, Support Vector Machines (SMO), …

Lazy Techniques: IB1, IBK, …

Bayesian Techniques: Naive Bayes

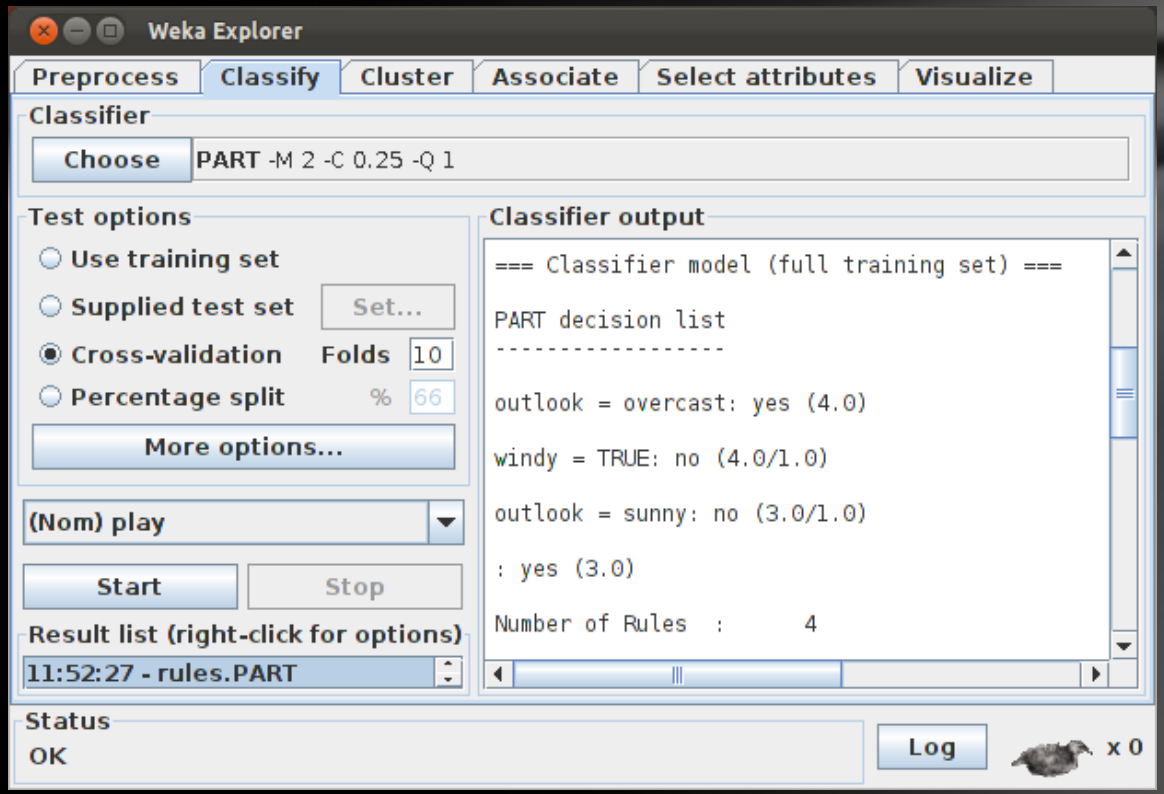
Meta-techniques

‘Classify’ Panel – Trees (C4.5 – J48)

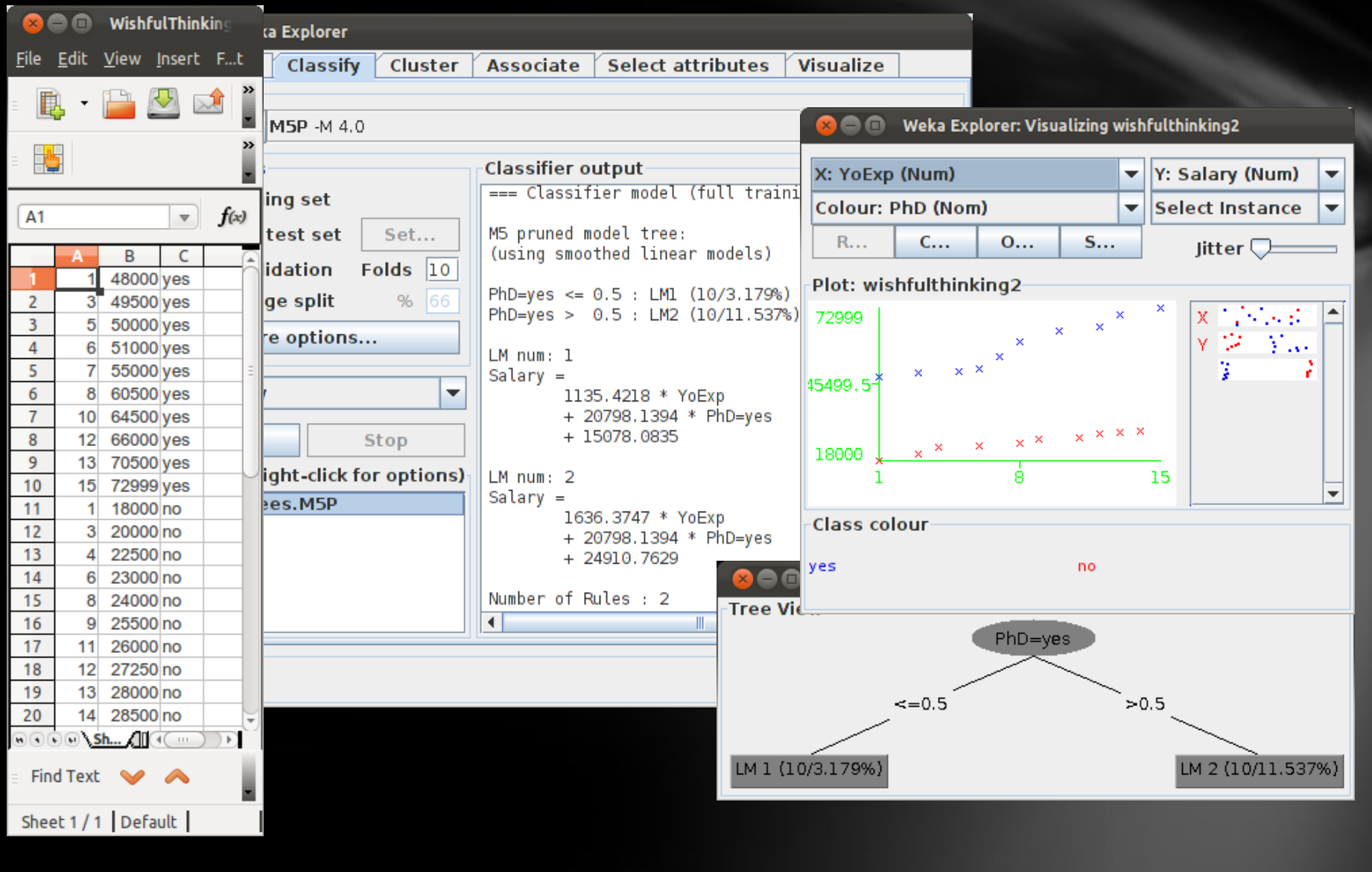


‘Classify’ Panel – Rules

We can generate rules from trees or there are other algorithms to generate rules directly



‘Classify Panel – Regression trees (M5P)



Lazy Techniques

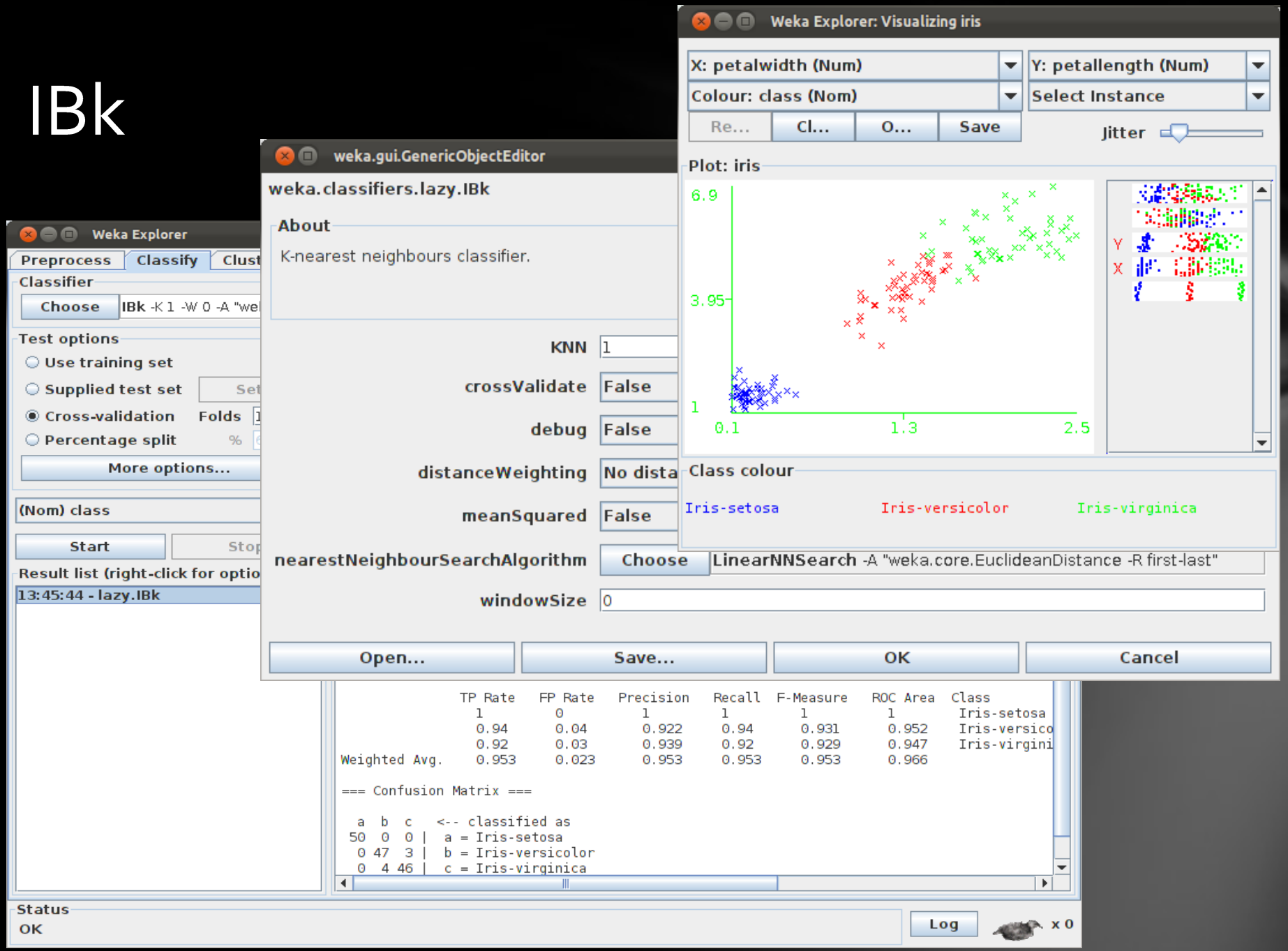
Lazy techniques (also known as Collaborative Filtering or Instance-based Learning) do not build models, just retain instances.

Once a new instance needs to be classifed, these techniques search for 'similar' instances in the repository

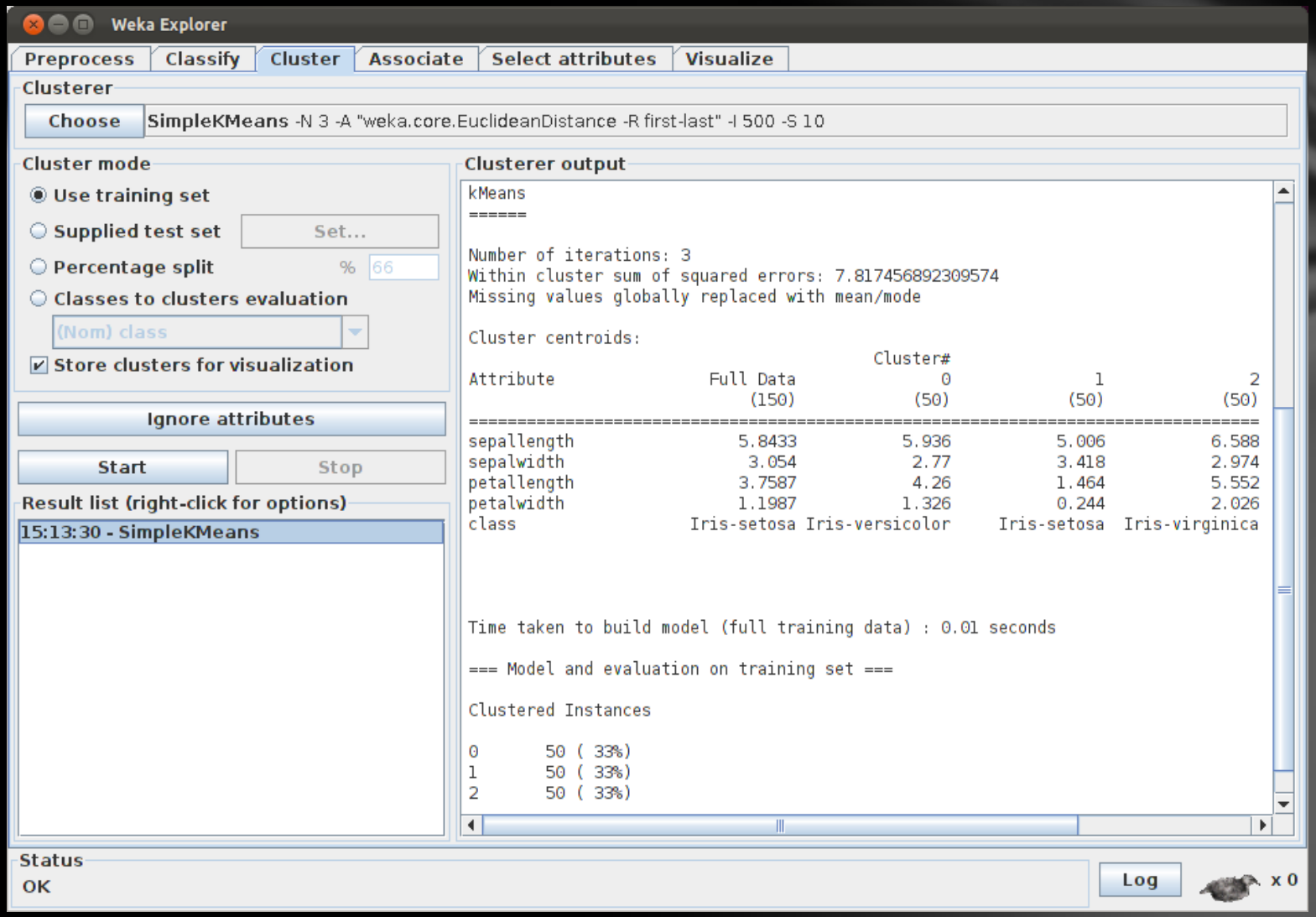
Weka's IBk implements k-NN.

- Weka's IBk uses the Euclidean distance by default.
- The sum of the squared differences between normalized attribute values is computed; this is then normalized by the number of attributes in the data; finally the square root is taken.
- Following this, weighting can be applied to the distances (if selected).
  - Normalising distances for all attributes so that attributes have the same impact on the distance function.
- It may return *k* neighbours. If there are ties in the distance, neighbours are voted to form the final classification.

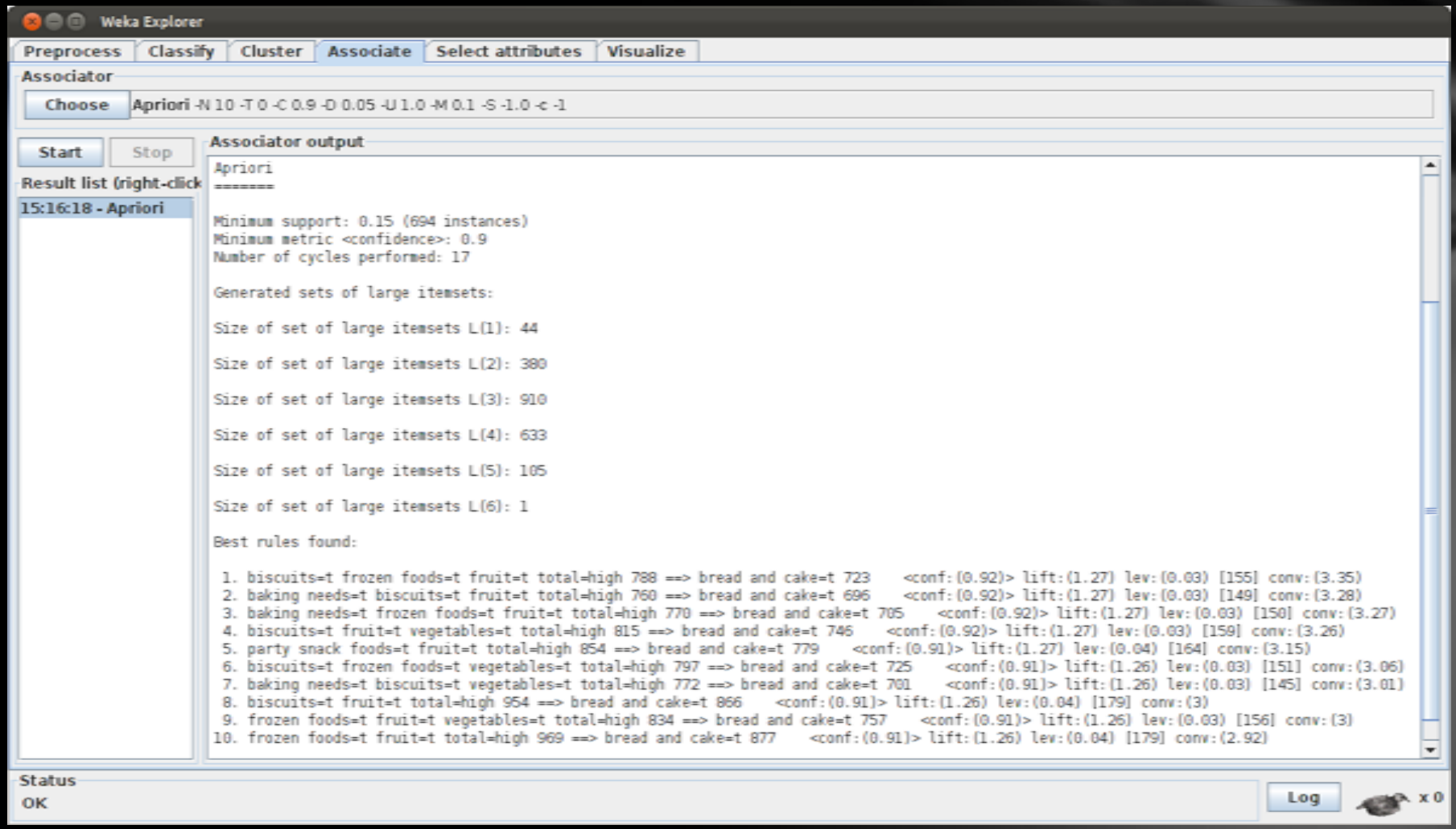
IBk


Weka Explorer: Visualizing iris
X: petalwidth (Num)
Y: petallength (Num)
Colour: class (Nom)
Select Instance
Re...
Cl...
O...
Save
Jitter
Plot: iris
6.9
3.95
1
0.1
1.3
2.5
Y
X
Class colour
Iris-setosa
Iris-versicolor
Iris-virginica
weka.gui.GenericObjectEditor
weka.classifiers.lazy.IBk
About
K-nearest neighbours classifier.
KNN
1
crossValidate
False
debug
False
distanceWeighting
No dista
meanSquared
False
nearestNeighbourSearchAlgorithm
Choose
LinearNNSearch -A "weka.core.EuclideanDistance -R first-last"
windowSize
0
Open...
Save...
OK
Cancel
Weka Explorer
Preprocess
Classify
Clust
Classifier
Choose
IBk -K 1 -W 0 -A "wel
Test options
Use training set
Supplied test set
Set
Cross-validation
Folds
Percentage split
%
More options...
(Nom) class
Start
Stop
Result list (right-click for optio
13:45:44 - lazy.IBk
TP Rate FP Rate Precision Recall F-Measure ROC Area Class
1 0 1 1 1 1 Iris-setosa
0.94 0.04 0.922 0.94 0.931 0.952 Iris-versico
0.92 0.03 0.939 0.92 0.929 0.947 Iris-virgini
Weighted Avg. 0.953 0.023 0.953 0.953 0.953 0.966
=== Confusion Matrix ===
a b c <-- classified as
50 0 0 | a = Iris-setosa
0 47 3 | b = Iris-versicolor
0 4 46 | c = Iris-virginica
Status
OK
Log
x 0

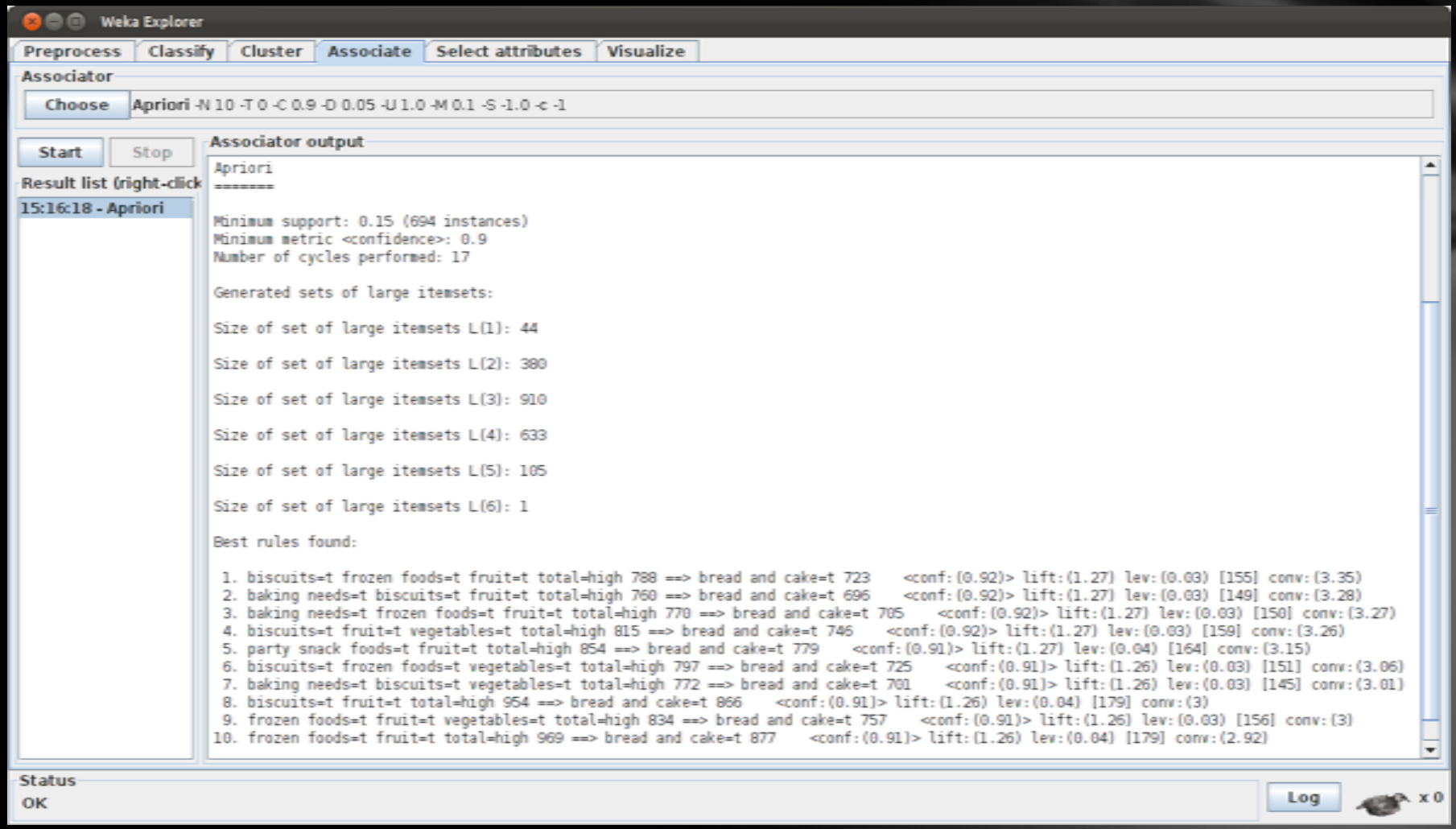
Cluster Tab


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
15:13:30 - SimpleKMeans
Clusterer output
kMeans
======
Number of iterations: 3
Within cluster sum of squared errors: 7.817456892309574
Missing values globally replaced with mean/mode
Cluster centroids:
Cluster#
Attribute Full Data 0 1 2
(150) (50) (50) (50)
sepallength 5.8433 5.936 5.006 6.588
sepalwidth 3.054 2.77 3.418 2.974
petallength 3.7587 4.26 1.464 5.552
petalwidth 1.1987 1.326 0.244 2.026
class Iris-setosa Iris-versicolor Iris-setosa Iris-virginica
Time taken to build model (full training data) : 0.01 seconds
=== Model and evaluation on training set ===
Clustered Instances
0 50 (33%)
1 50 (33%)
2 50 (33%)
Status
OK
Log
x 0

Association



Association



Attribute Selection

Feature selection is important in different ways:

- A reduced volume of data allows different data mining or searching techniques to be applied.
- Irrelevant and redundant attributes can generate less accurate and more complex models. Furthermore, data mining algorithms can be executed faster.
- We can avoid the collection of data for those irrelevant and redundant attributes in the future.

There exist two major approach in feature selection from the method's output point of view depending on the way that features are evaluated:

- **Feature ranking**, a.k.a. feature weighting, assesses individual features and assigns them weights according to their degrees of relevance
- **Feature subset selection** (FSS) evaluate the goodness of each found feature subset (Unusually, some search strategies in combination with subset evaluation can provide a ranked list).

Feature Ranking

Feature ranking algorithms category, one can expect a ranked list of features which are ordered according to evaluation measures

Each attribute correlation with the class is evaluated independenly of other attributes according to a statistical test (e.g.: χ-squared, Infogain, etc.)

- A subset of features is often selected from the top of a ranking list. A feature is good and thus will be selected if its weight of relevance is greater than a user-specified threshold value,
- or we can simply select the first *k* features from the ranked list.

This approach is efficient for high-dimensional data due to its linear time complexity in terms of dimensionality

Very fast method.

However, it cannot detect redundant attributes

Feature Subset Selection (FSS)

In the FSS category, candidate subsets are generated based on a certain search strategy:

- **Exhaustive**, **heuristic** and **random** search

Each candidate subset is evaluated by a certain **evaluation measure**. If a new subset turns out to be better, it replaces the previous best subset. The process of subset generation and evaluation is repeated until a given stopping criterion is satisfied. Existing evaluation measures include:

- Consistency measure attempts to find a minimum number of features that separate classes as consistently as the full set of features can. An inconsistency is defined as to instances having the same feature values but different class labels.
- Correlation measure evaluates the goodness of feature subsets based on the hypothesis that good feature subsets contain features highly correlated to the class, yet uncorrelated to each other.
- Accuracy of a learning algorithm (Wrapper-based attribute selection) uses the target learning algorithm to estimate the worth of attribute subsets. The feature subset selection algorithm conducts a search for a good subset using the induction algorithm itself as part of the evaluation function.

The time complexity in terms of data dimensionality is, exponential for exhaustive search, quadratic for heuristic search and linear to the number of iterations in a random search

FSS – CFS (Correlation Based FS)

Correlation measure is applied in an algorithm called CFS that exploit heuristic search (best first) to search for candidate feature subsets.

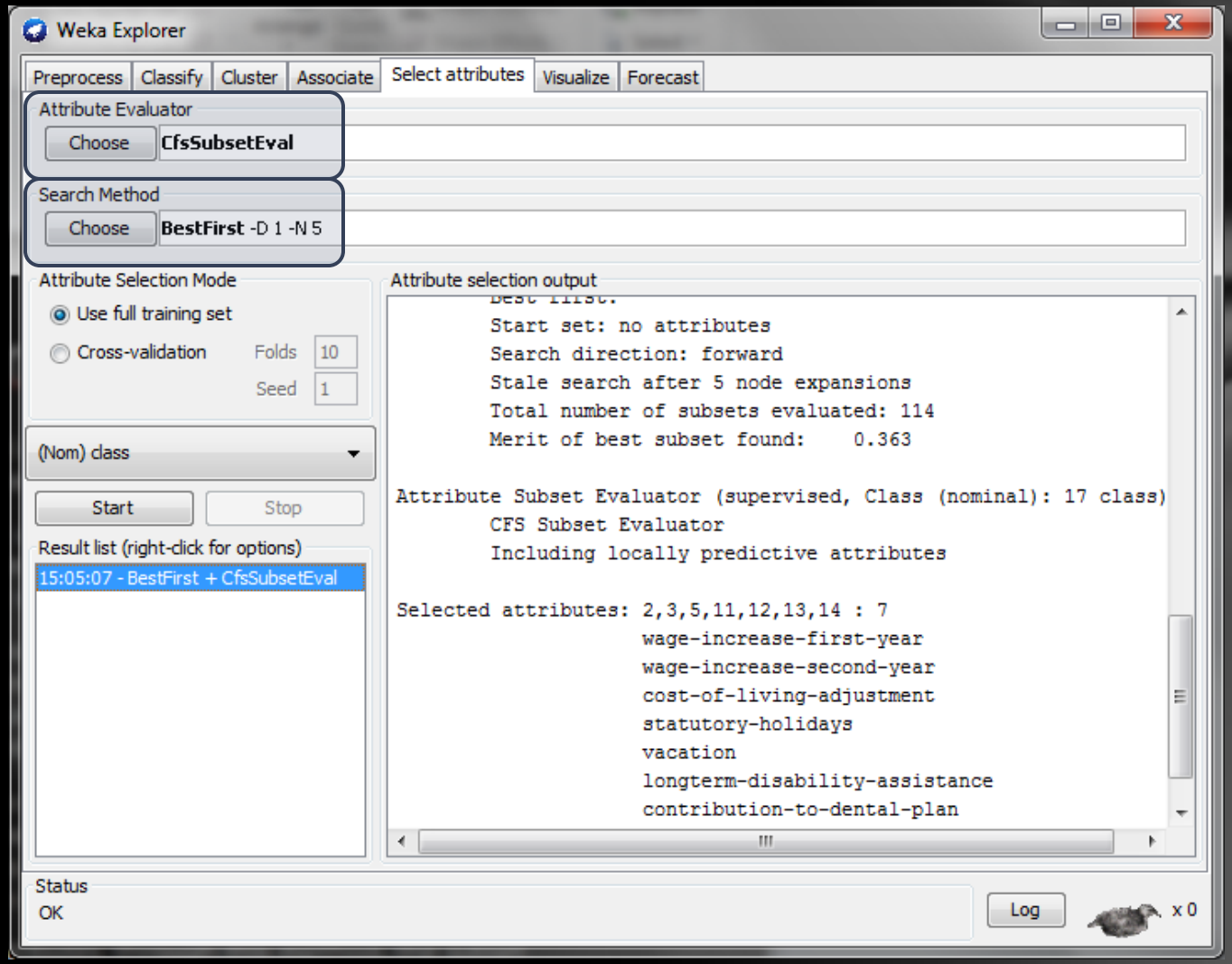
- One of the most frequently used search techniques is hill-climbing (greedy). It starts with an empty set and evaluates each attribute individually to find the best single attribute.

It then tries each of the remaining attributes in conjunction with the best to find the most suited pair of attributes. In the next iteration, each of the remaining attributes are tried in conjunction with the best pair to find the most suited group of three attributes.

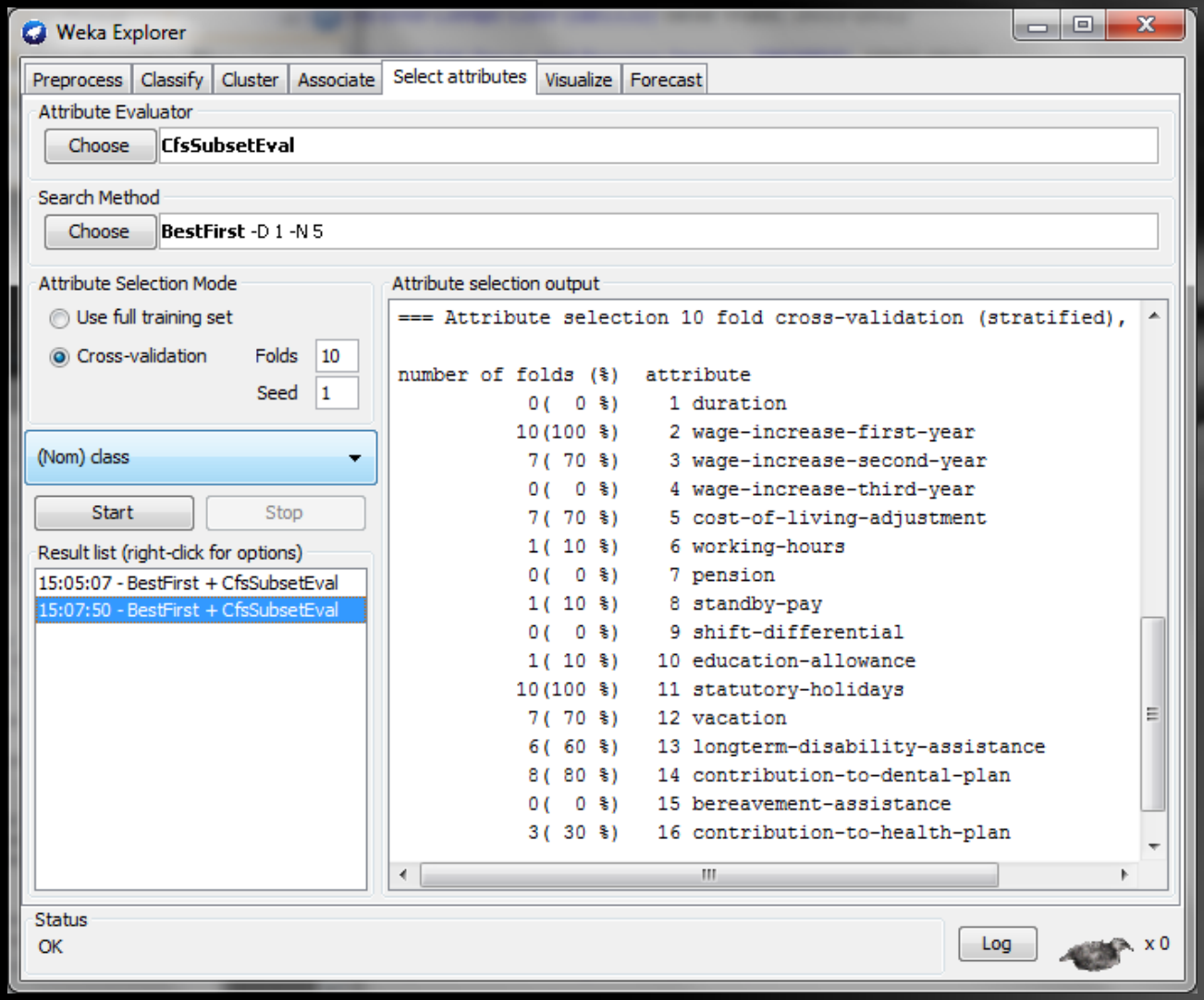
This process continues until no single attribute addition improves the evaluation of the subset; i.e., subset evaluator is run M times to choose the best single attribute, *M-1* times to find the best pair of attributes, *M-2* times the best group of three, and so on.

- E.g., if we have chosen 5 attributes through this method, the subset evaluator has been run *M+(M-1)+(M-2)+(M-3)+(M-4)* times.

CFS Example


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Forecast
Attribute Evaluator
Choose
CfsSubsetEval
Search Method
Choose
BestFirst -D 1 -N 5
Attribute Selection Mode
Use full training set
Cross-validation
Folds
10
Seed
1
(Nom) class
Start
Stop
Result list (right-click for options)
15:05:07 - BestFirst + CfsSubsetEval
Attribute selection output
Start set: no attributes
Search direction: forward
Stale search after 5 node expansions
Total number of subsets evaluated: 114
Merit of best subset found: 0.363
Attribute Subset Evaluator (supervised, Class (nominal): 17 class)
CFS Subset Evaluator
Including locally predictive attributes
Selected attributes: 2,3,5,11,12,13,14 : 7
wage-increase-first-year
wage-increase-second-year
cost-of-living-adjustment
statutory-holidays
vacation
longterm-disability-assistance
contribution-to-dental-plan
Status
OK
Log
x 0

CFS – 10 CV


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Forecast
Attribute Evaluator
Choose
CfsSubsetEval
Search Method
Choose
BestFirst -D 1 -N 5
Attribute Selection Mode
Use full training set
Cross-validation
Folds
10
Seed
1
(Nom) class
Start
Stop
Result list (right-click for options)
15:05:07 - BestFirst + CfsSubsetEval
15:07:50 - BestFirst + CfsSubsetEval
Attribute selection output
=== Attribute selection 10 fold cross-validation (stratified),
number of folds (%) attribute
0(0 %) 1 duration
10(100 %) 2 wage-increase-first-year
7(70 %) 3 wage-increase-second-year
0(0 %) 4 wage-increase-third-year
7(70 %) 5 cost-of-living-adjustment
1(10 %) 6 working-hours
0(0 %) 7 pension
1(10 %) 8 standby-pay
0(0 %) 9 shift-differential
1(10 %) 10 education-allowance
10(100 %) 11 statutory-holidays
7(70 %) 12 vacation
6(60 %) 13 longterm-disability-assistance
8(80 %) 14 contribution-to-dental-plan
0(0 %) 15 bereavement-assistance
3(30 %) 16 contribution-to-health-plan
Status
OK
Log
x 0

Exercise - Hay fever

Find the best possible model to recommend the type of drug for hay fever depending on the patient. The attributes collected from historical patients include:

- Age
- Sex
- BP Blood Pressure
- Cholesterol level
- Na: Blood sodium level
- K: Blook potasium level
- There are 5 possible drugs: DrugA, DrugB, DrugC, DrugX, DrugY

The dataset can be downloaded from:

- http://www.cc.uah.es/drg/courses/datamining/datasets.zip

Hints:

- Try C4.5, can you simplify the generated tree combining attributes?
- Naive Bayes: can you improve the 'default' results with Feature Selection?

Exercise – Cost Sensitive Example

Using the German credit dataset:

http://www.cc.uah.es/drg/courses/datamining/datasets.zip

This dataset is composed of 20 attributes (7 numeric and 13 nominal) of clients of a bank requesting a credit.

Please, check for details of the in the file itself, including the provided cost matrix:

	actual		
		good	bad
	good	0	1
	bad	5	0

This cost matrix means that it is 5 times more costly to give credit to a person that won't pay back than the other way around.

Check what happens with ZeroR, Naïve Bayes, Bagging

Association Rules and dependencies

The Titanic dataset (titanic.arff) is composed of 4 attributes

- Class (1st, 2nd, 3rd)
- Age (adult, child)
- Sex (male, femele)
- Survived (yes, no)

describing the actual characteristics of the 2,201 passengers of The Titanic (*"Report on the Loss of the 'Titanic' (S.S.)" (1990), British Board of Trade Inquiry Report_ (reprint), Gloucester, UK: Allan Sutton Publishing*)

Using the Apriori algorithm extract information contained in this dataset

- Modify the default parameters to obtain rules that consider 'child' in the Age attribute. As there is a small number of values containing samples of age = 0, those rules are filtered out with due to their low coverage

Exercise – Clustering

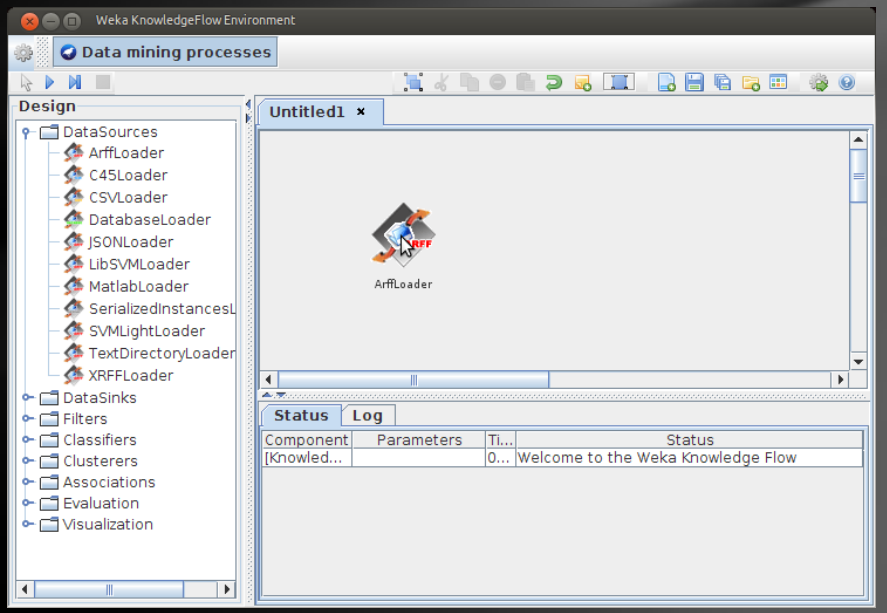
Using the employees dataset:

http://www.cc.uah.es/drg/courses/datamining/datasets.zip

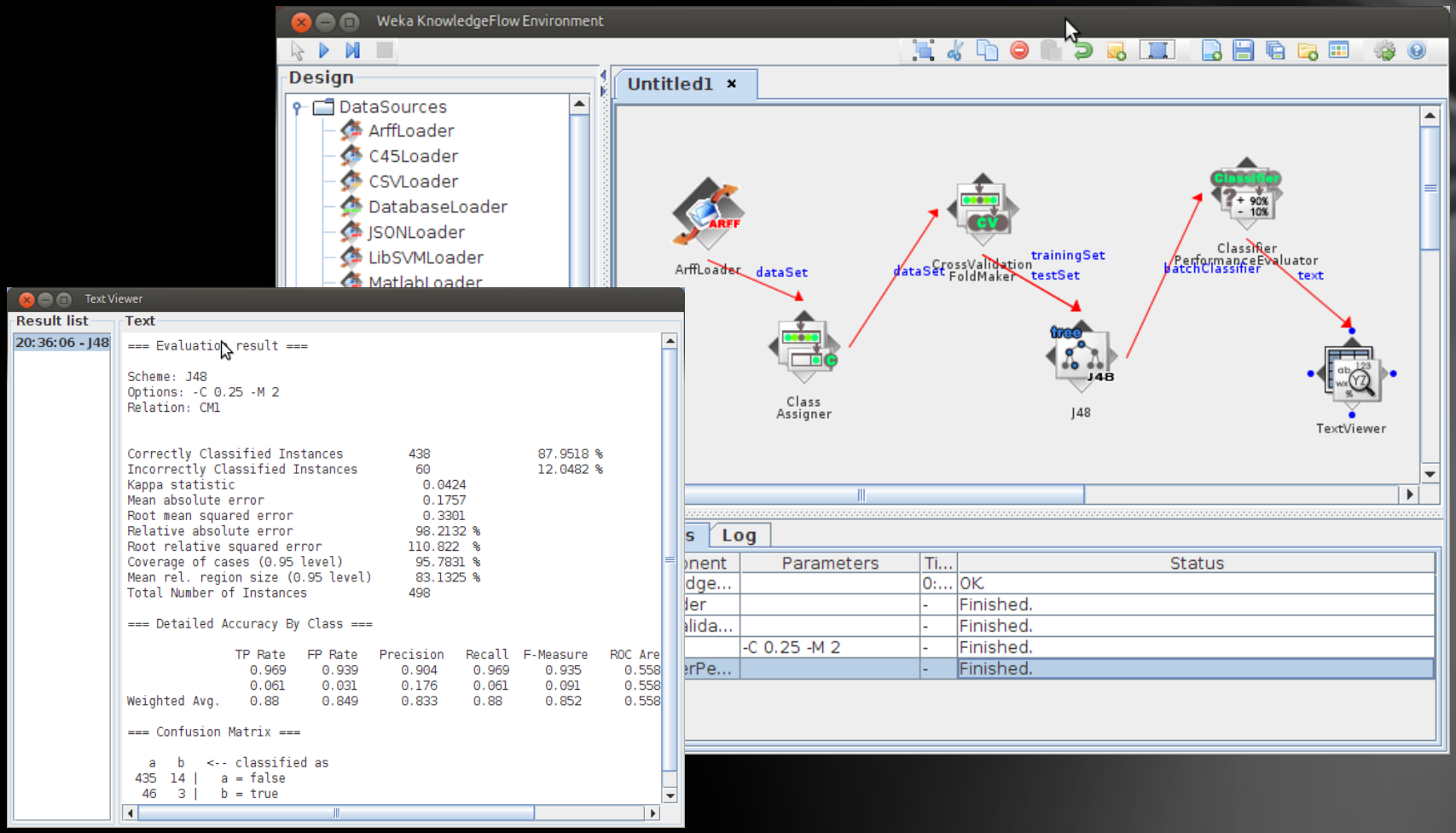
Checks the results for k Means usign 3 clusters.

KnowlegeFlow

Visually



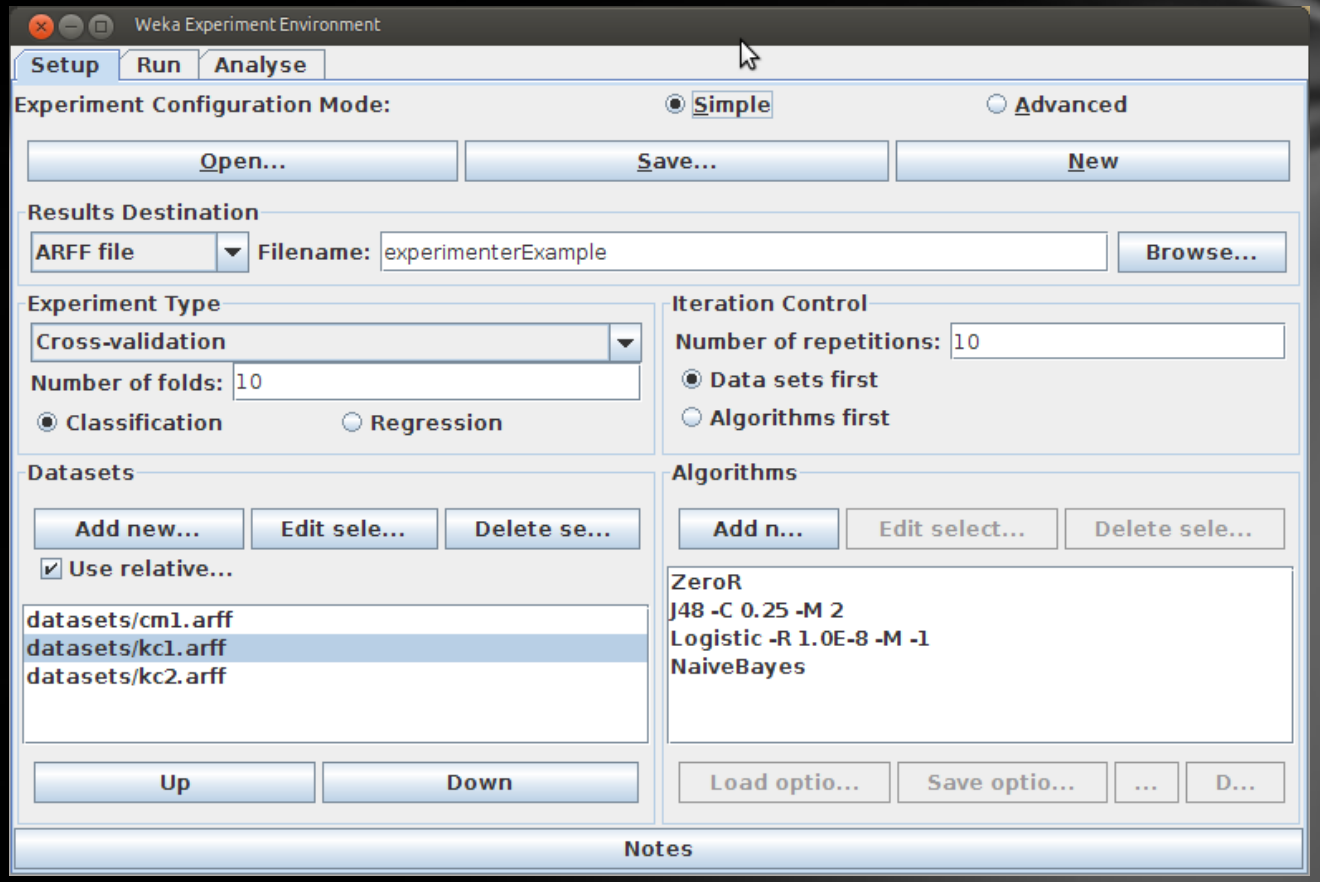
Example of Running J4.8 (C4.5) using KnowledgeFlow



Experimenter

Allow us to compare multiple algorithms

Typically, 10 times 10-CV



Experimenter – Output

After Running the experiment (Run tab) and loading the generated file (Analyze tab)