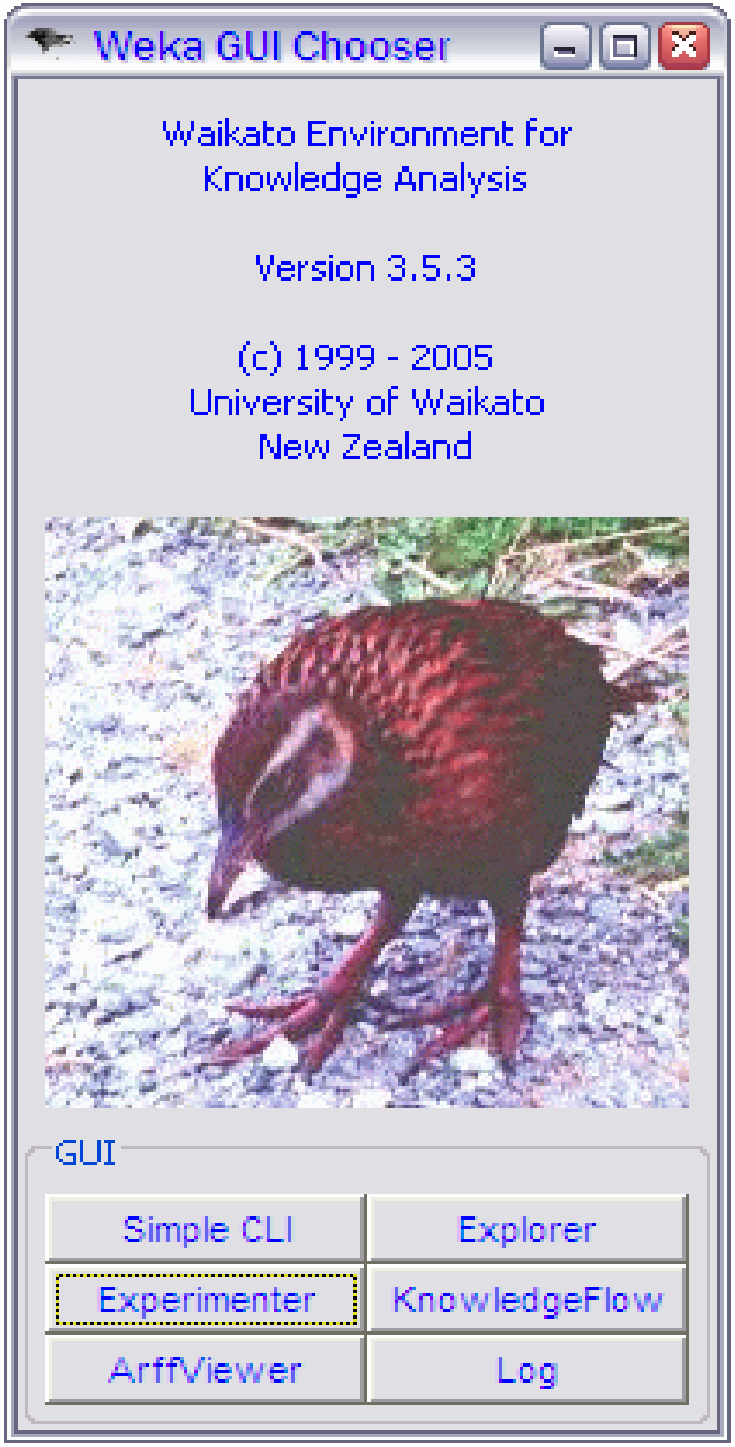
# Machine Learning algorithms and methods in Weka

Presented by: William Elazmeh

PhD. candidate at the Ottawa-Carleton Institute for Computer Science, University of Ottawa, Canada

Abstract:

this workshop presents a review of concepts and methods used in machine learning. The workshop aims to illustrate such ideas using the Weka software. The workshop is divided into 3 parts; (1) an illustration of data processing and using machine learning algorithms in Weka, (2) a demonstration of experiment administrations in Weka, and (3) a talk on evaluating machine learning algorithms using ROC and Cost Curves.



Weka GUI Chooser

Waikato Environment for
Knowledge Analysis

Version 3.5.3

(c) 1999 - 2005
University of Waikato
New Zealand

- 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:

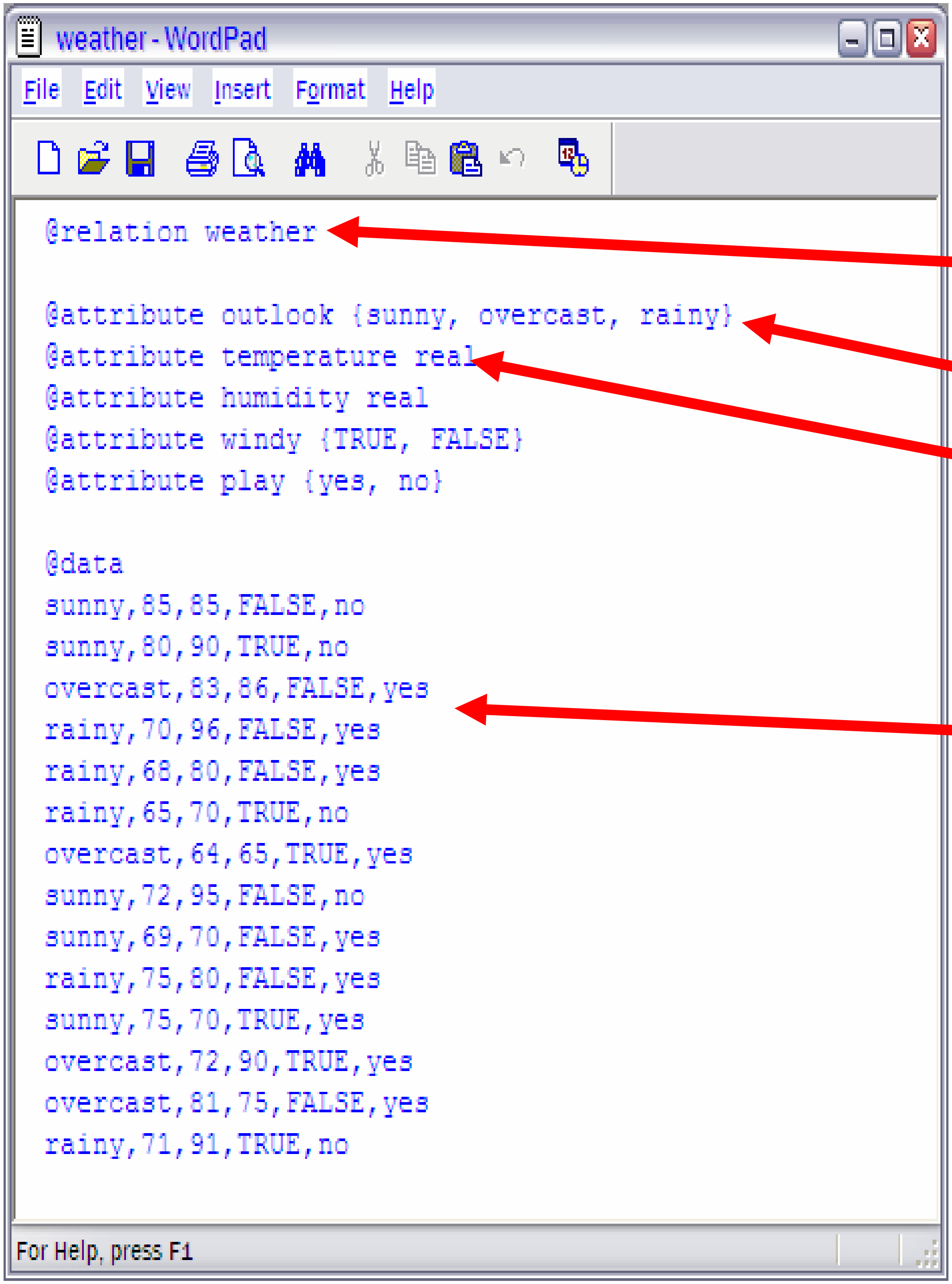
- Data pre-processing tools
- Learning algorithms
- evaluation methods
  - Graphical user interfaces
  - An environment experimenting

WEKA is a Machine Learning Toolkit that consists of:

- The Explorer
  - Classification and Regression
  - Clustering
  - Finding Associations
  - Attribute Selection
  - Data Visualization
- The Experimenter
- The Knowledge Flow GUI

Note: the content of this presentation is based on a Weka presentation prepared by Eibe Frank at the Department of Computer Science, University of Waikato, New Zealand

# The ARFF Flat File Format



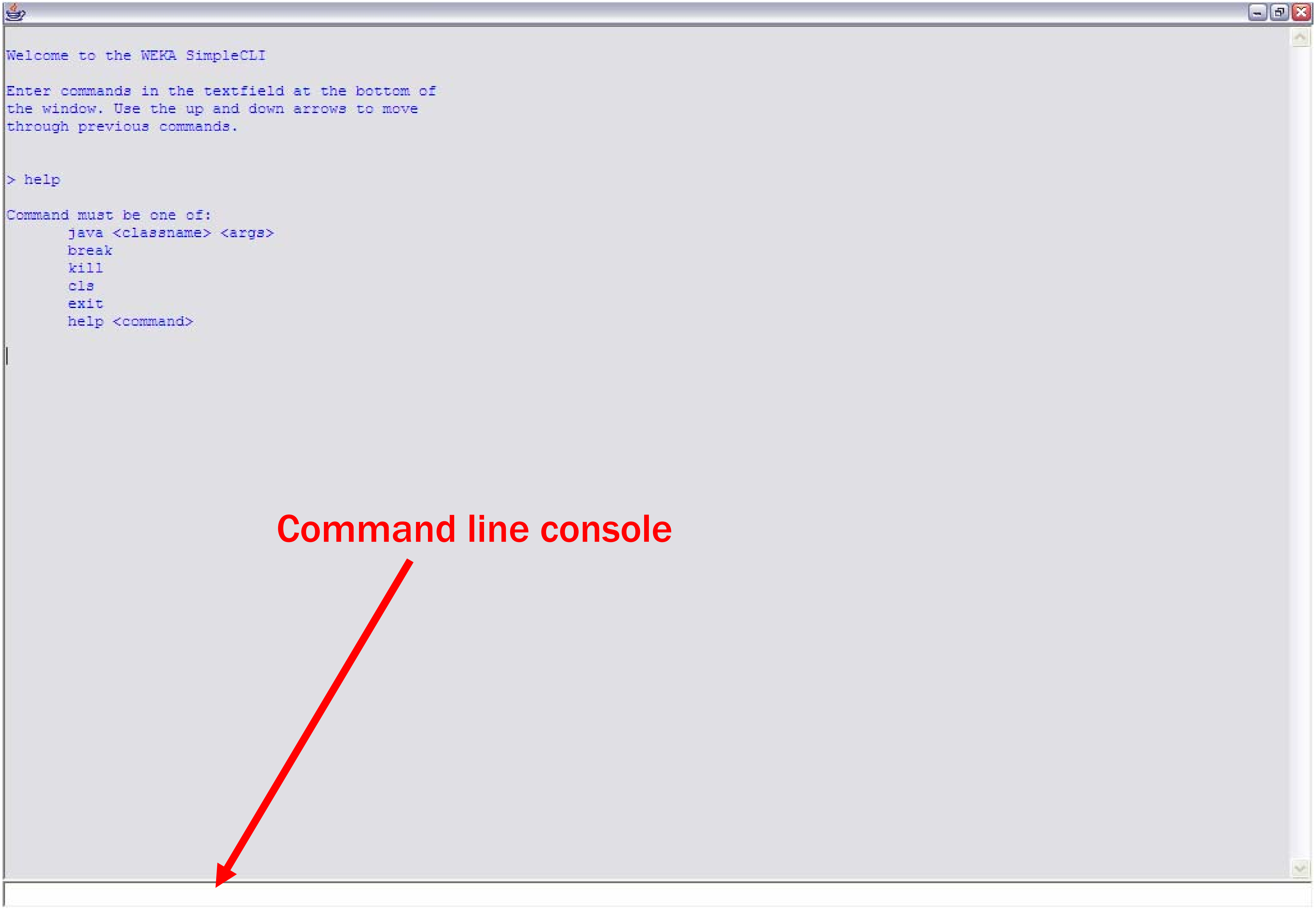
```
@relation weather

@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
```

class

Nominal attribute

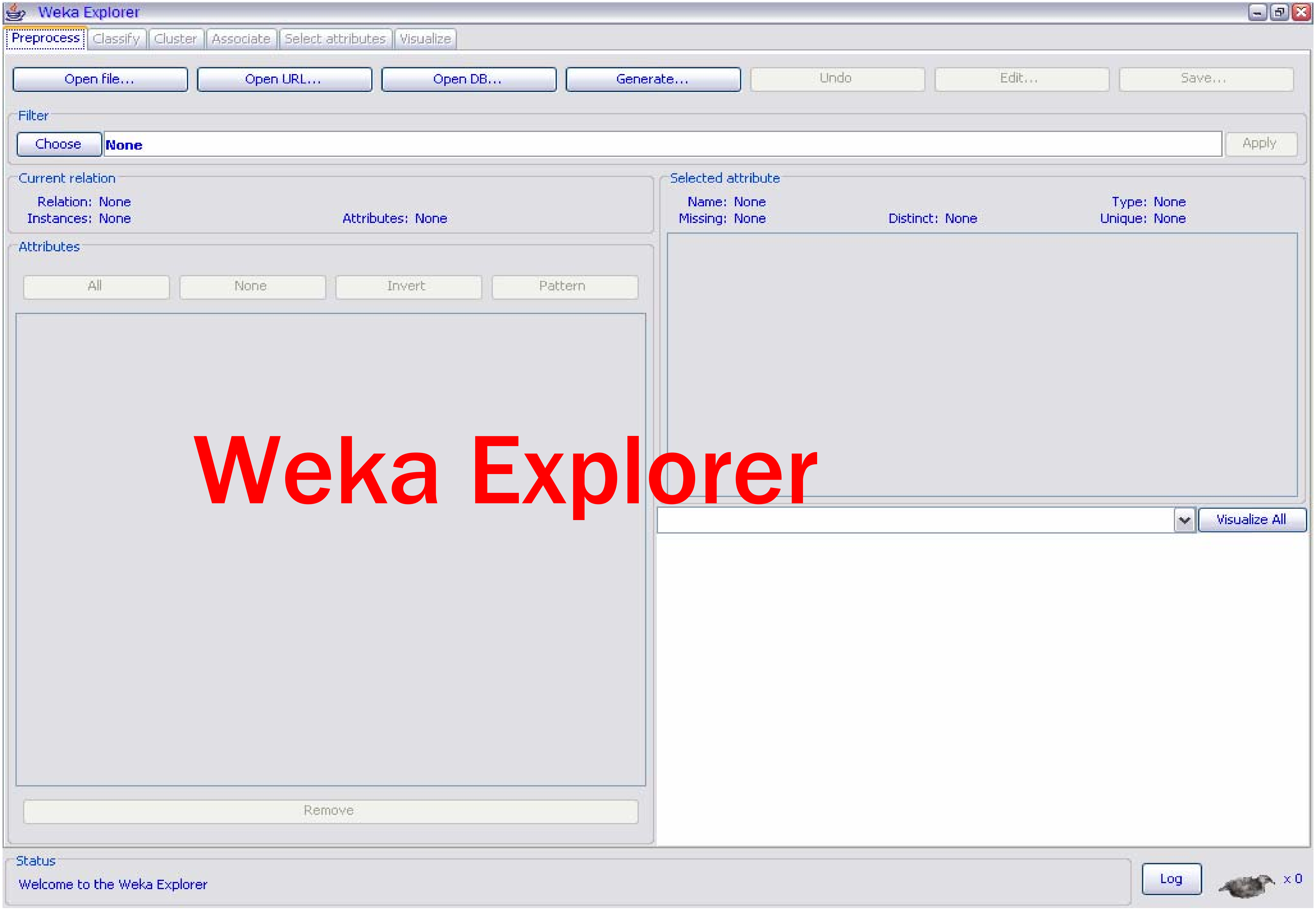
Numeric attribute

Data records



```
Welcome to the WEKA SimpleCLI

Enter commands in the textfield at the bottom of
the window. Use the up and down arrows to move
through previous commands.


> help

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

Command line console



# Weka Explorer

Filter

None

Current relation

Relation: None
Instances: None
Attributes: None

Attributes

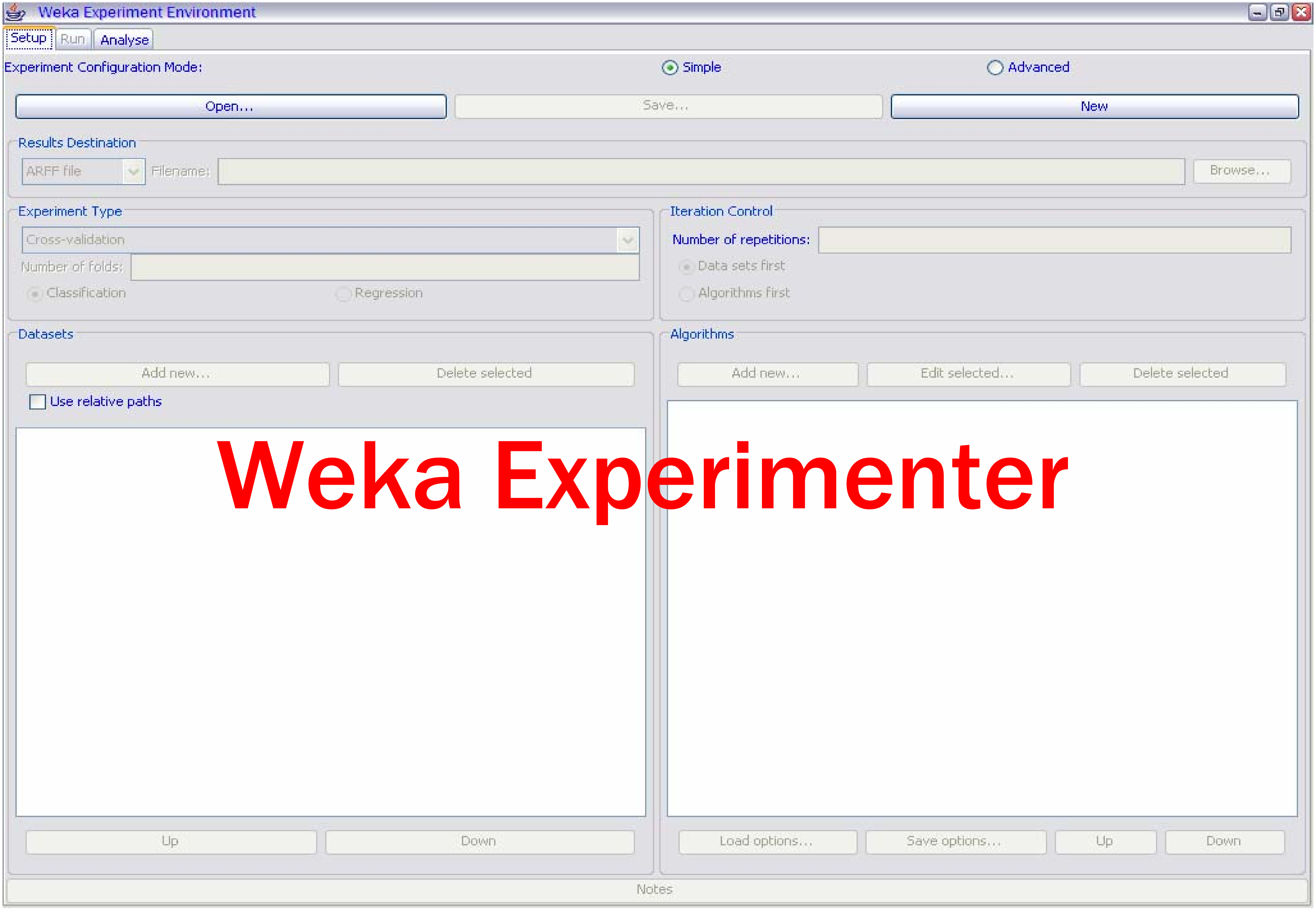
Selected attribute

Name: None
Missing: None
Distinct: None
Type: None
Unique: None

Status

Welcome to the Weka Explorer

# Weka Experimenter



Weka Experiment Environment

Experiment Configuration Mode: ● Simple ○ Advanced

Results Destination

ARFF file Filename:

Experiment Type

Cross-validation

Number of folds:

● Classification ○ Regression

Iteration Control

Number of repetitions:

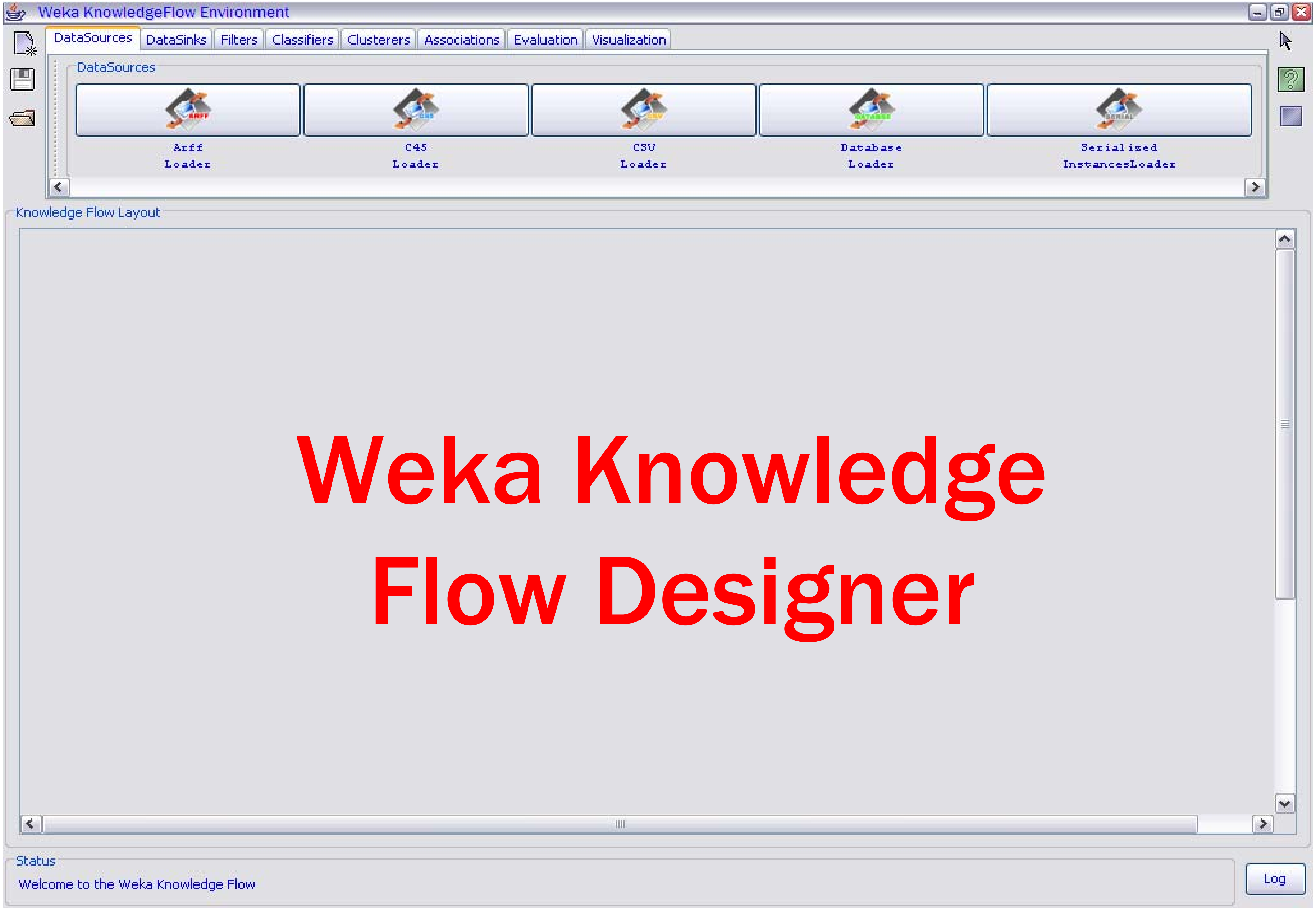
● Data sets first

○ Algorithms first

Datasets

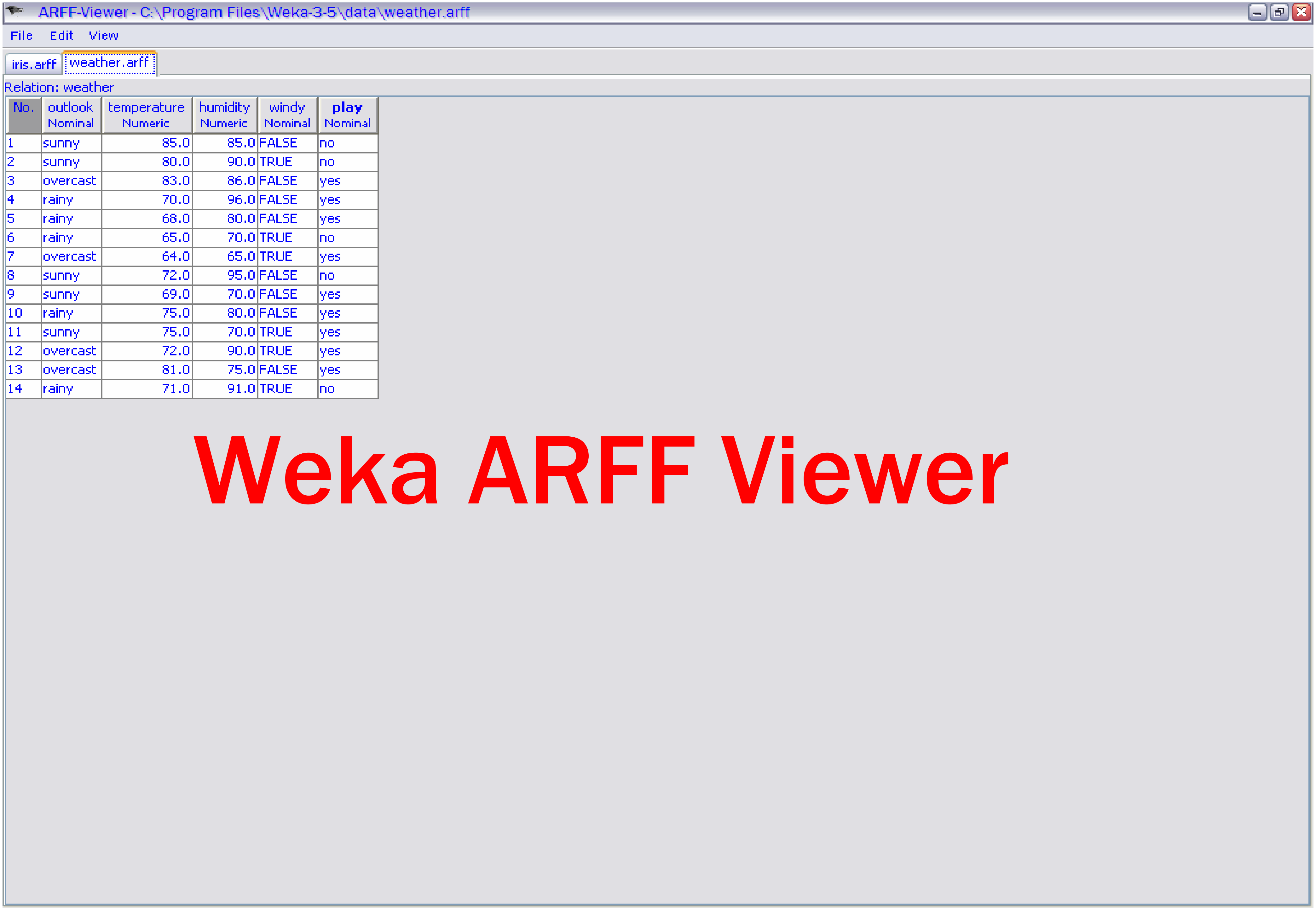
□ Use relative paths

Algorithms



# Weka Knowledge Flow Designer

Welcome to the Weka Knowledge Flow

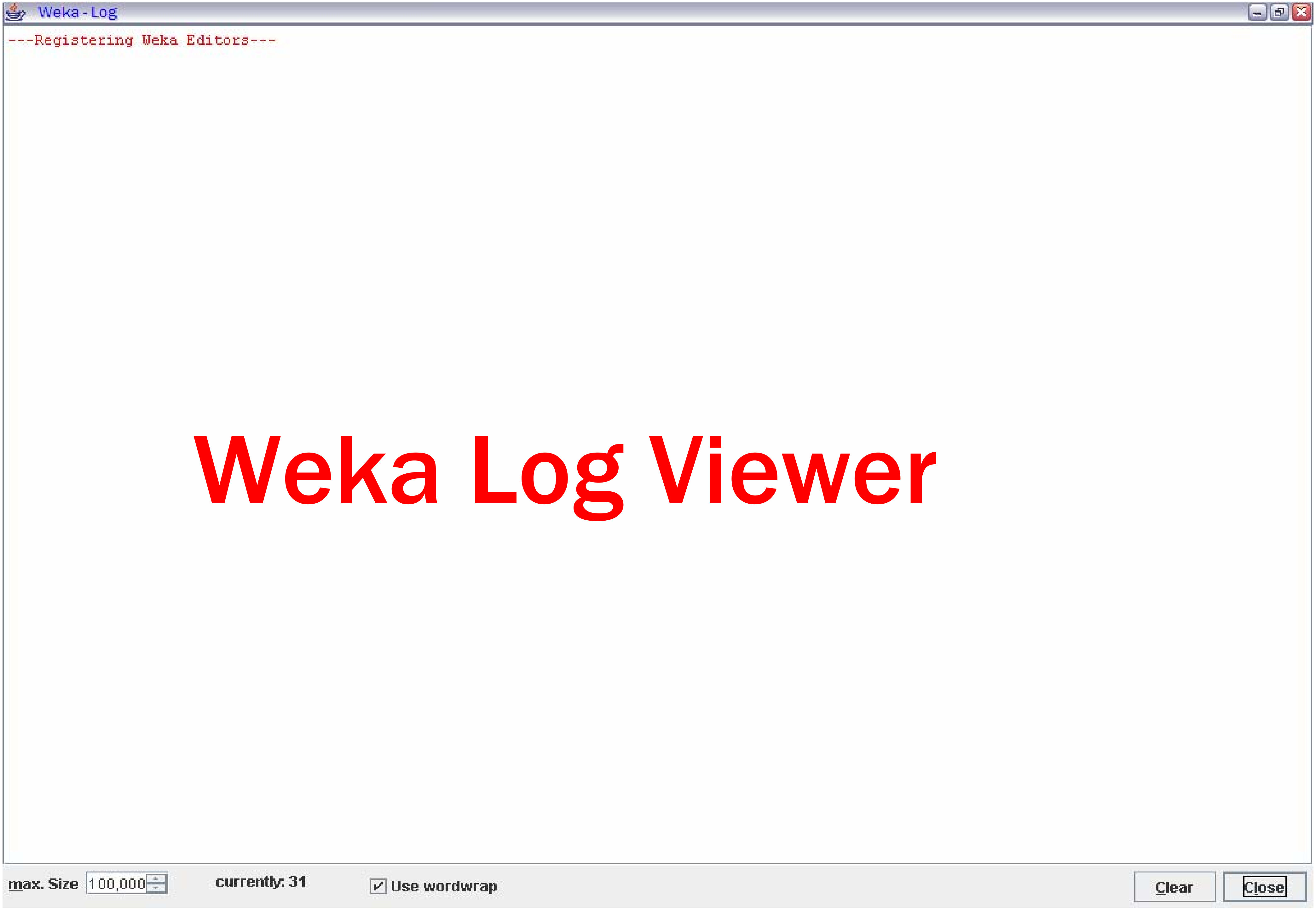


Relation: weather

| No. | outlook<br>Nominal | temperature<br>Numeric | humidity<br>Numeric | windy<br>Nominal | **play**<br>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 |

# Weka ARFF Viewer

# Weka Log Viewer



```
---Registering Weka Editors---
```

max. Size 100,000

currently: 31

☑ Use wordwrap

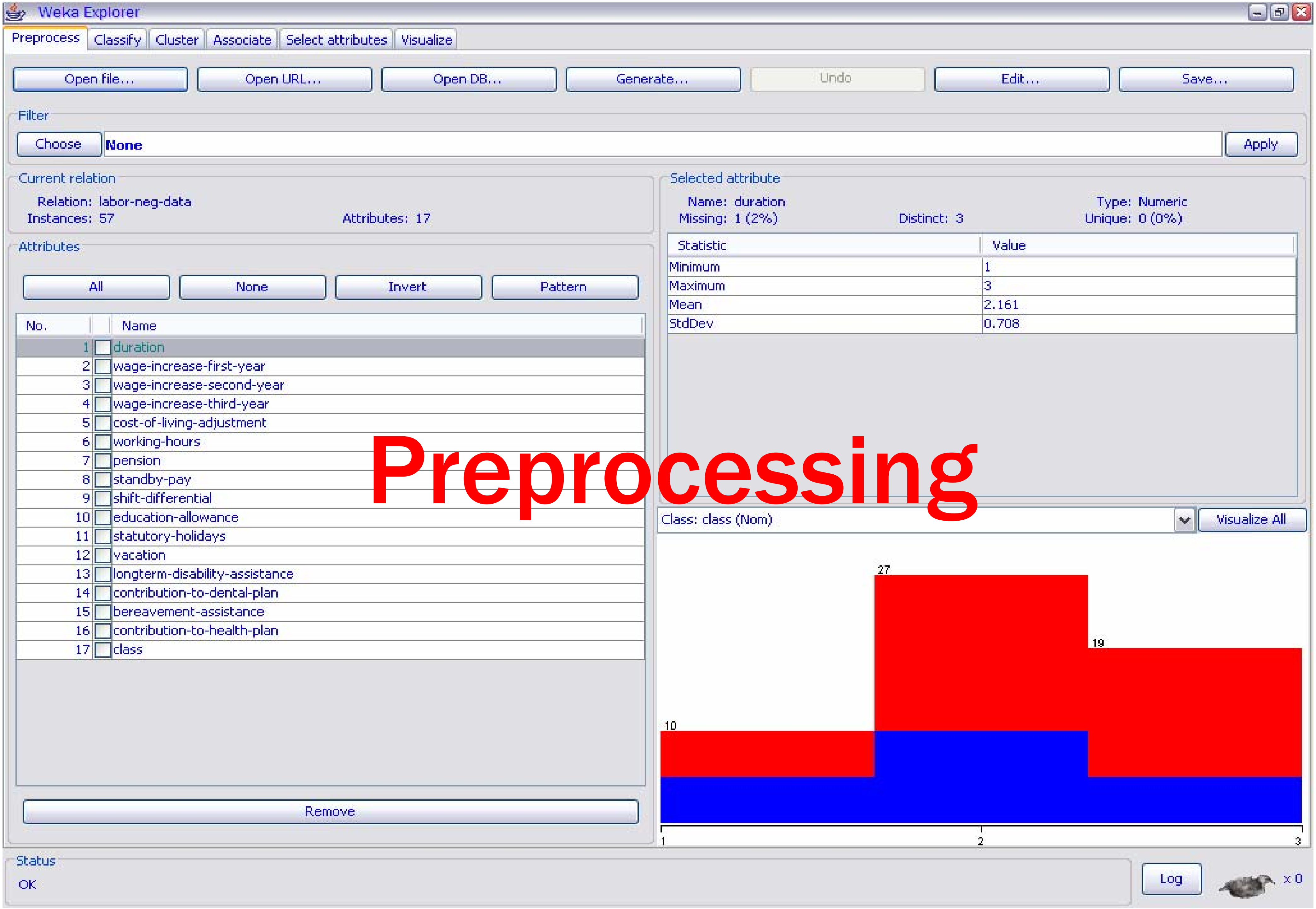
# **Part I:** data processing and using machine learning algorithms in Weka

# 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, …

# Preprocessing



Filter

None

Current relation

Relation: labor-neg-data
Instances: 57
Attributes: 17

Selected attribute

Name: duration
Type: Numeric
Missing: 1 (2%)
Distinct: 3
Unique: 0 (0%)

| Statistic | Value |
|---|---|
| Minimum | 1 |
| Maximum | 3 |
| Mean | 2.161 |
| StdDev | 0.708 |

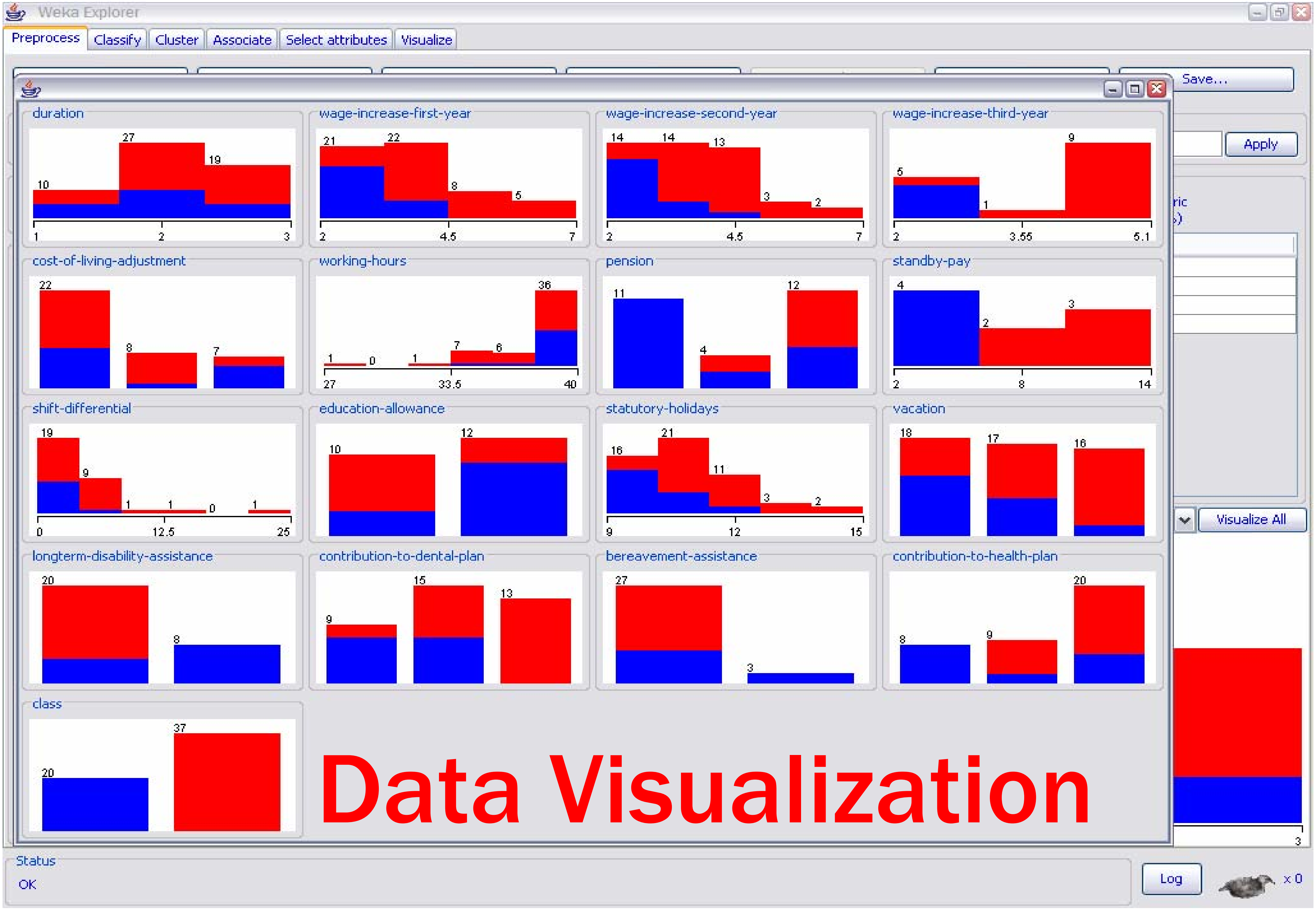
Attributes

| No. | | Name |
|---|---|---|
| 1 | □ | duration |
| 2 | □ | wage-increase-first-year |
| 3 | □ | wage-increase-second-year |
| 4 | □ | wage-increase-third-year |
| 5 | □ | cost-of-living-adjustment |
| 6 | □ | working-hours |
| 7 | □ | pension |
| 8 | □ | standby-pay |
| 9 | □ | shift-differential |
| 10 | □ | education-allowance |
| 11 | □ | statutory-holidays |
| 12 | □ | vacation |
| 13 | □ | longterm-disability-assistance |
| 14 | □ | contribution-to-dental-plan |
| 15 | □ | bereavement-assistance |
| 16 | □ | contribution-to-health-plan |
| 17 | □ | class |

Class: class (Nom)

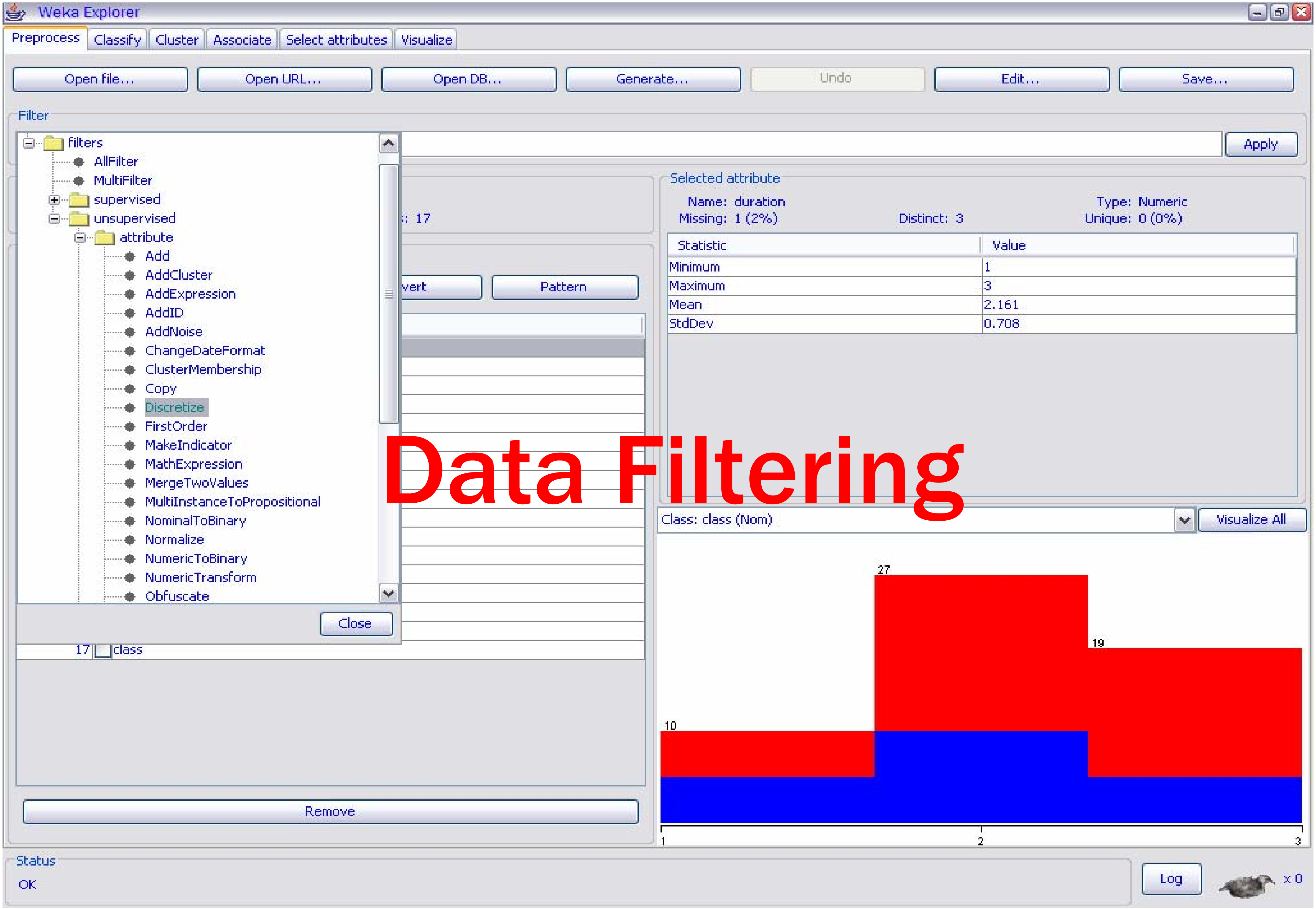
Status

OK



# Data Visualization

# Data Filtering



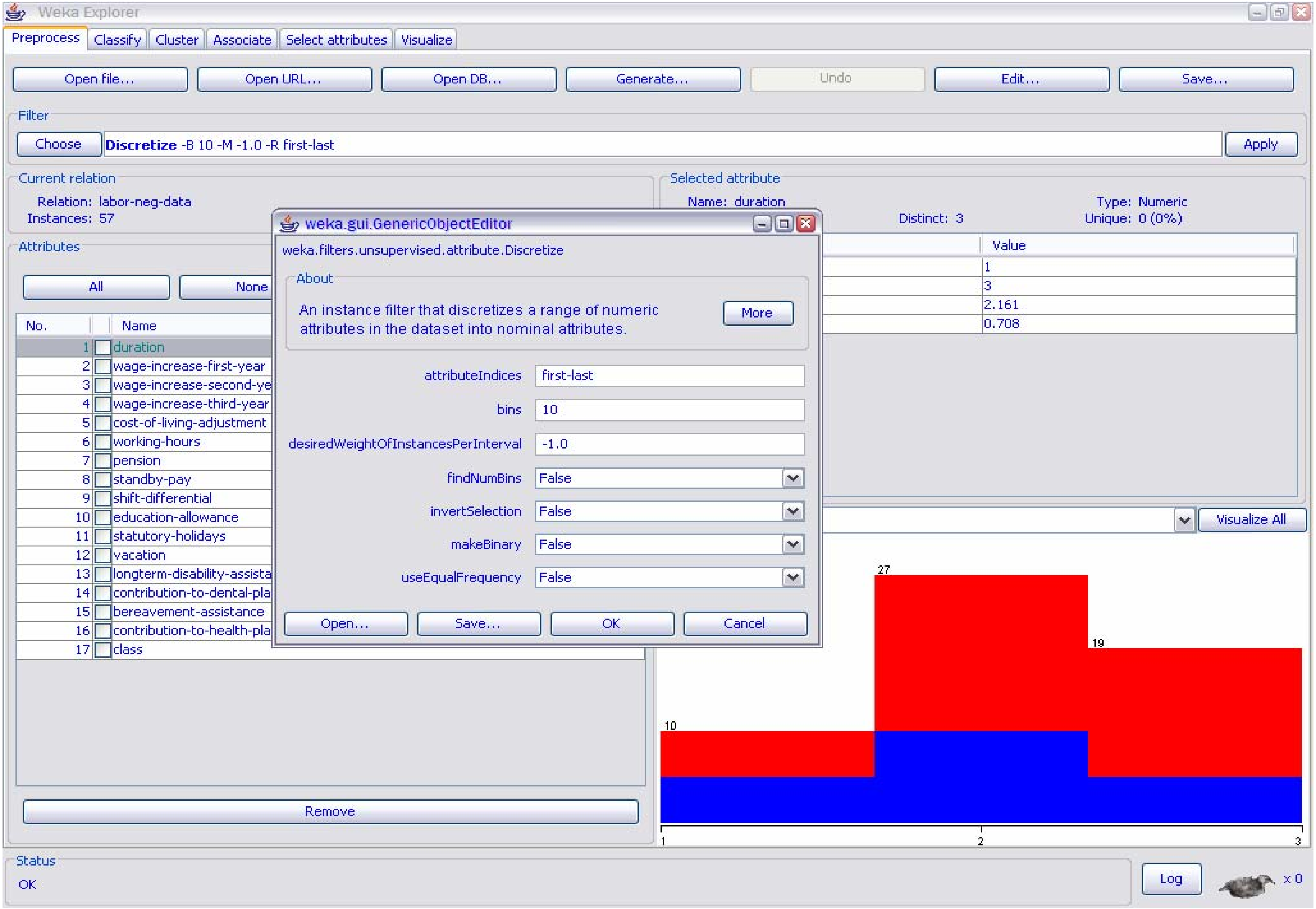
Selected attribute

Name: duration  
Missing: 1 (2%)  
Distinct: 3  
Type: Numeric  
Unique: 0 (0%)

| Statistic | Value |
|---|---|
| Minimum | 1 |
| Maximum | 3 |
| Mean | 2.161 |
| StdDev | 0.708 |

Class: class (Nom)

Status  
OK



**Discretize** -B 10 -M -1.0 -R first-last

Relation: labor-neg-data
Instances: 57

Name: duration
Type: Numeric
Distinct: 3
Unique: 0 (0%)

| Value |
|---|
| 1 |
| 3 |
| 2.161 |
| 0.708 |

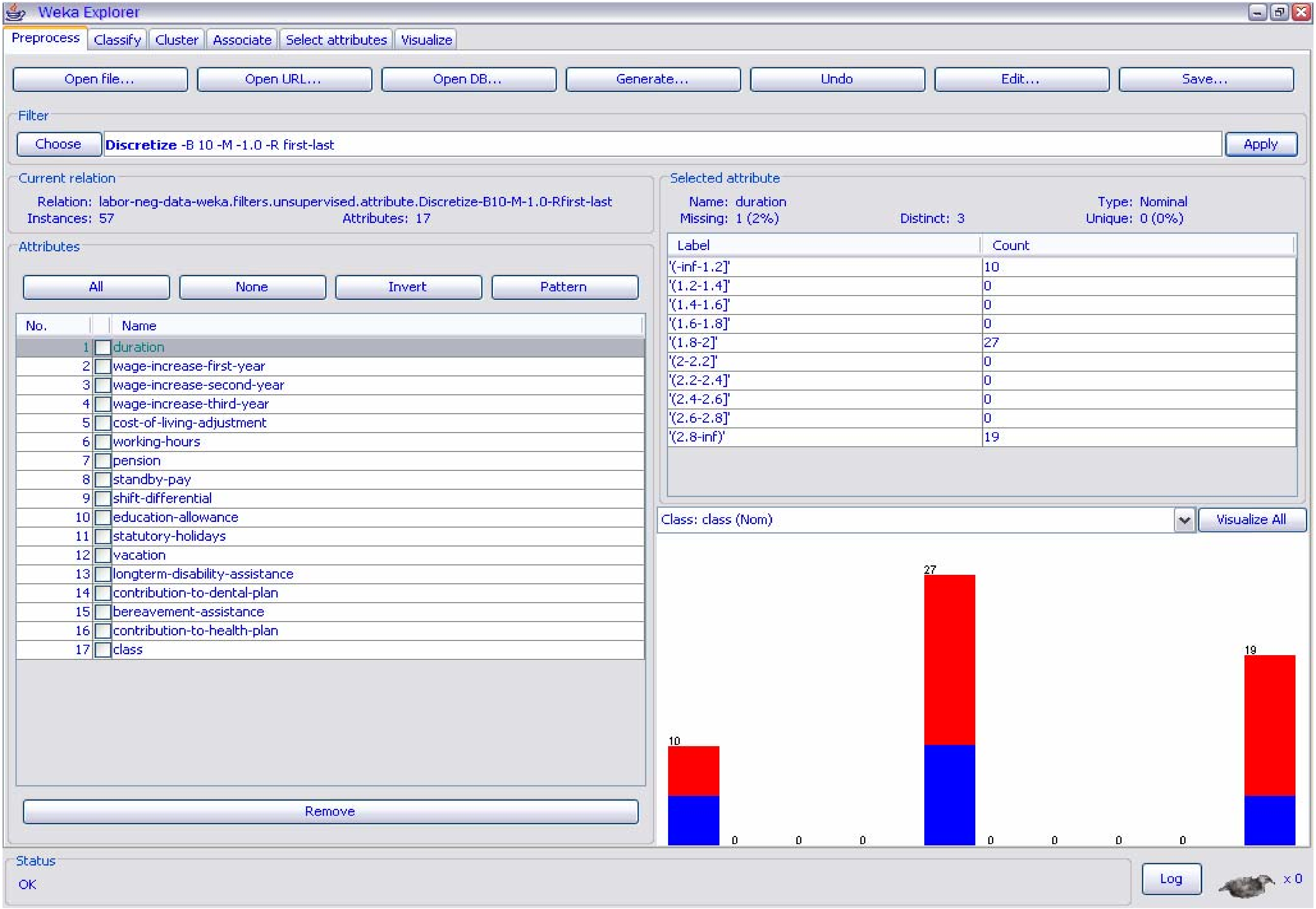
| No. | | Name |
|---|---|---|
| 1 | □ | duration |
| 2 | □ | wage-increase-first-year |
| 3 | □ | wage-increase-second-ye |
| 4 | □ | wage-increase-third-year |
| 5 | □ | cost-of-living-adjustment |
| 6 | □ | working-hours |
| 7 | □ | pension |
| 8 | □ | standby-pay |
| 9 | □ | shift-differential |
| 10 | □ | education-allowance |
| 11 | □ | statutory-holidays |
| 12 | □ | vacation |
| 13 | □ | longterm-disability-assista |
| 14 | □ | contribution-to-dental-pla |
| 15 | □ | bereavement-assistance |
| 16 | □ | contribution-to-health-pla |
| 17 | □ | class |

weka.filters.unsupervised.attribute.Discretize

An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.

| | |
|---|---|
| attributeIndices | first-last |
| bins | 10 |
| desiredWeightOfInstancesPerInterval | -1.0 |
| findNumBins | False |
| invertSelection | False |
| makeBinary | False |
| useEqualFrequency | False |

OK



Filter

**Discretize** -B 10 -M -1.0 -R first-last

Current relation

Relation: labor-neg-data-weka.filters.unsupervised.attribute.Discretize-B10-M-1.0-Rfirst-last
Instances: 57
Attributes: 17

Attributes

| No. | | Name |
|---|---|---|
| 1 | □ | duration |
| 2 | □ | wage-increase-first-year |
| 3 | □ | wage-increase-second-year |
| 4 | □ | wage-increase-third-year |
| 5 | □ | cost-of-living-adjustment |
| 6 | □ | working-hours |
| 7 | □ | pension |
| 8 | □ | standby-pay |
| 9 | □ | shift-differential |
| 10 | □ | education-allowance |
| 11 | □ | statutory-holidays |
| 12 | □ | vacation |
| 13 | □ | longterm-disability-assistance |
| 14 | □ | contribution-to-dental-plan |
| 15 | □ | bereavement-assistance |
| 16 | □ | contribution-to-health-plan |
| 17 | □ | class |

Selected attribute

Name: duration
Type: Nominal
Missing: 1 (2%)
Distinct: 3
Unique: 0 (0%)

| Label | Count |
|---|---|
| '(-inf-1.2]' | 10 |
| '(1.2-1.4]' | 0 |
| '(1.4-1.6]' | 0 |
| '(1.6-1.8]' | 0 |
| '(1.8-2]' | 27 |
| '(2-2.2]' | 0 |
| '(2.2-2.4]' | 0 |
| '(2.4-2.6]' | 0 |
| '(2.6-2.8]' | 0 |
| '(2.8-inf)' | 19 |

Class: class (Nom)

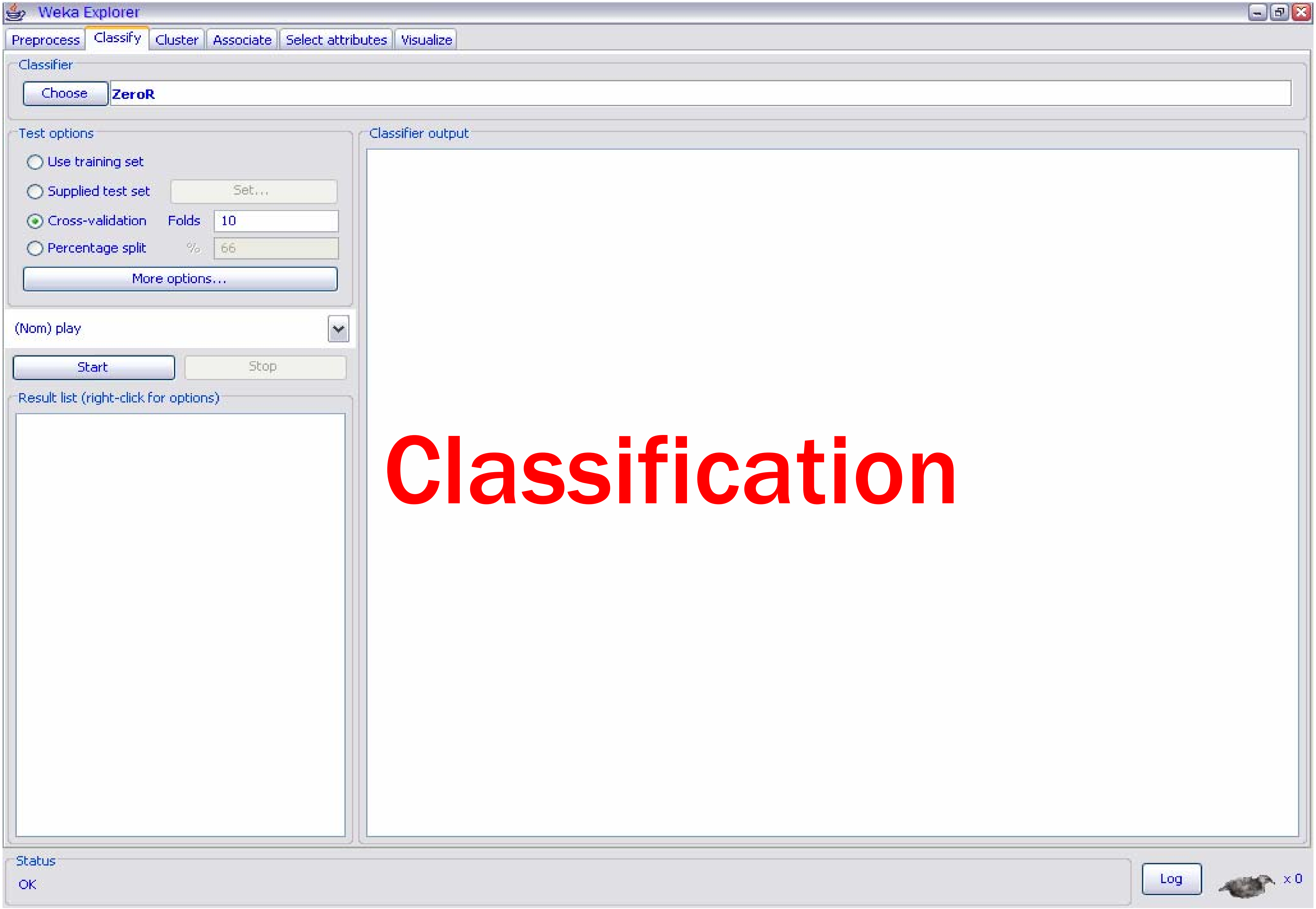
Status

OK

# Explorer: building “classifiers”

- Classifiers in WEKA are models for predicting nominal or numeric quantities
- Implemented learning schemes include:
  - Decision trees and lists, instance-based classifiers, support vector machines, multi-layer perceptrons, logistic regression, Bayes’ nets, …
- “Meta”-classifiers include:
  - Bagging, boosting, stacking, error-correcting output codes, locally weighted learning, …

# Classification



Classifier

ZeroR

Test options

○ Use training set

○ Supplied test set

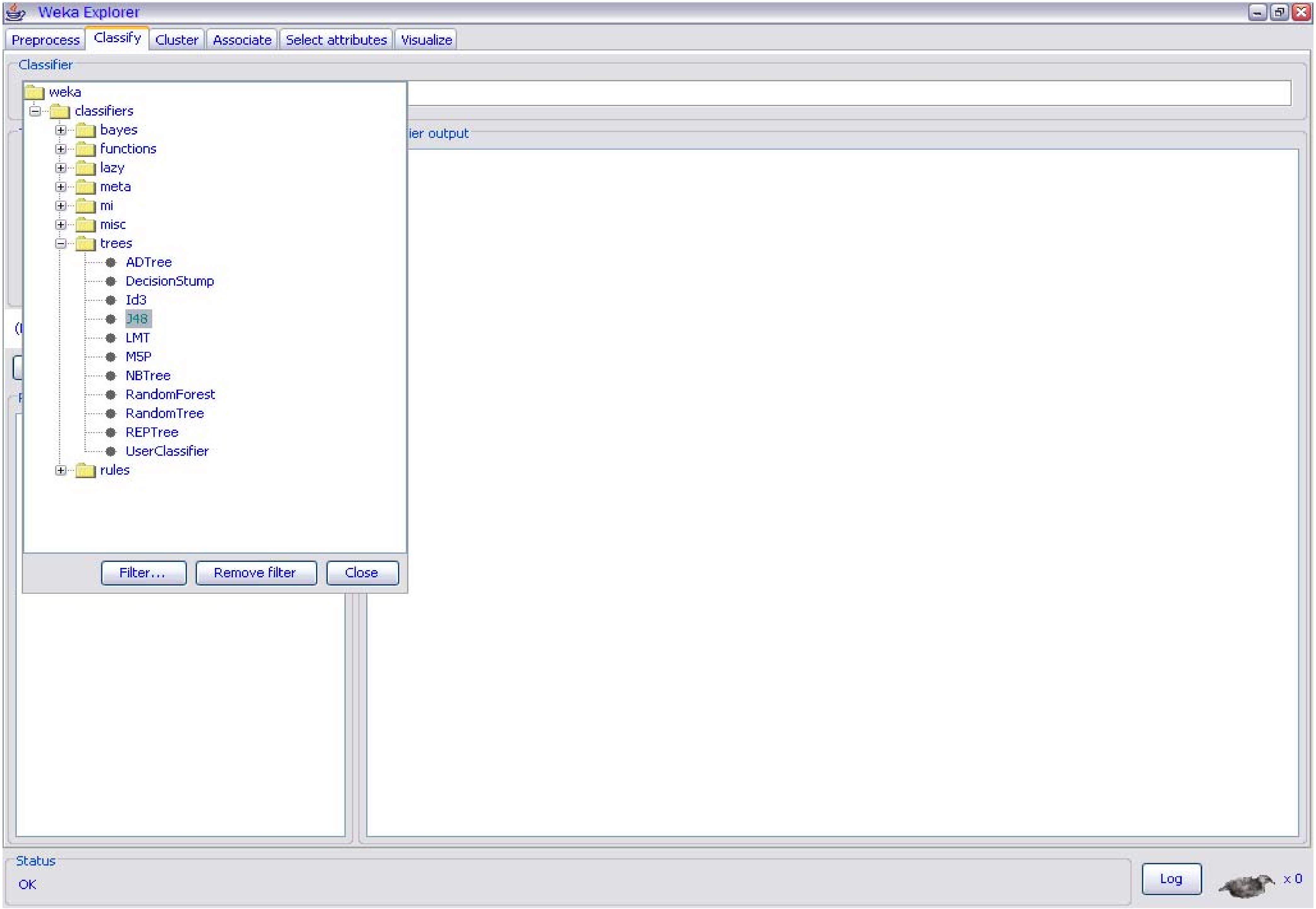
● Cross-validation Folds 10

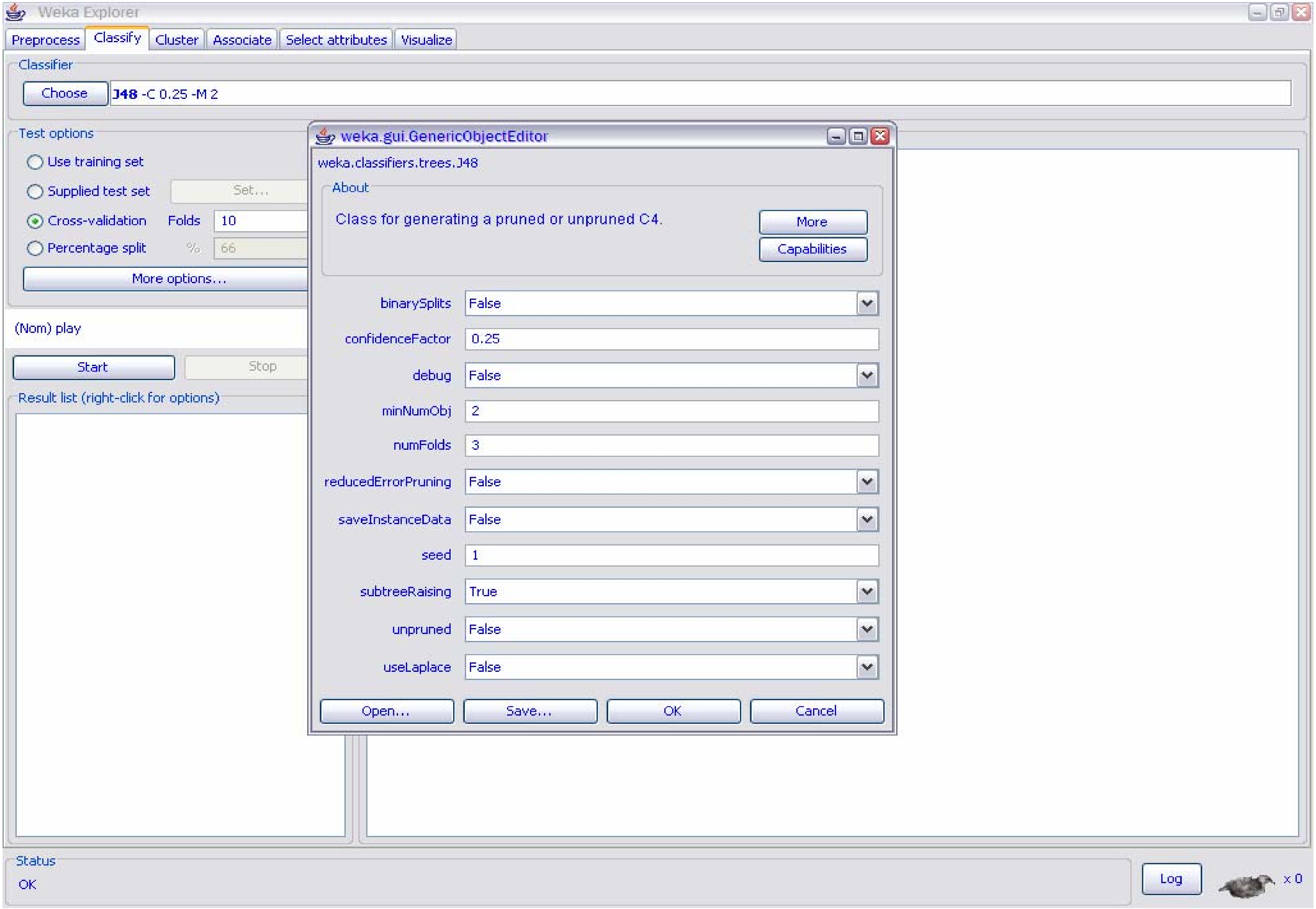
○ Percentage split % 66

(Nom) play

Status

OK


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
weka
classifiers
bayes
functions
lazy
meta
mi
misc
trees
ADTree
DecisionStump
Id3
J48
LMT
M5P
NBTree
RandomForest
RandomTree
REPTree
UserClassifier
rules
Filter...
Remove filter
Close
ier output
Status
OK
Log
x 0



Classifier

J48 -C 0.25 -M 2

Test options

○ Use training set

○ Supplied test set

● Cross-validation Folds 10

○ Percentage split % 66

(Nom) play

Result list (right-click for options)

weka.classifiers.trees.J48

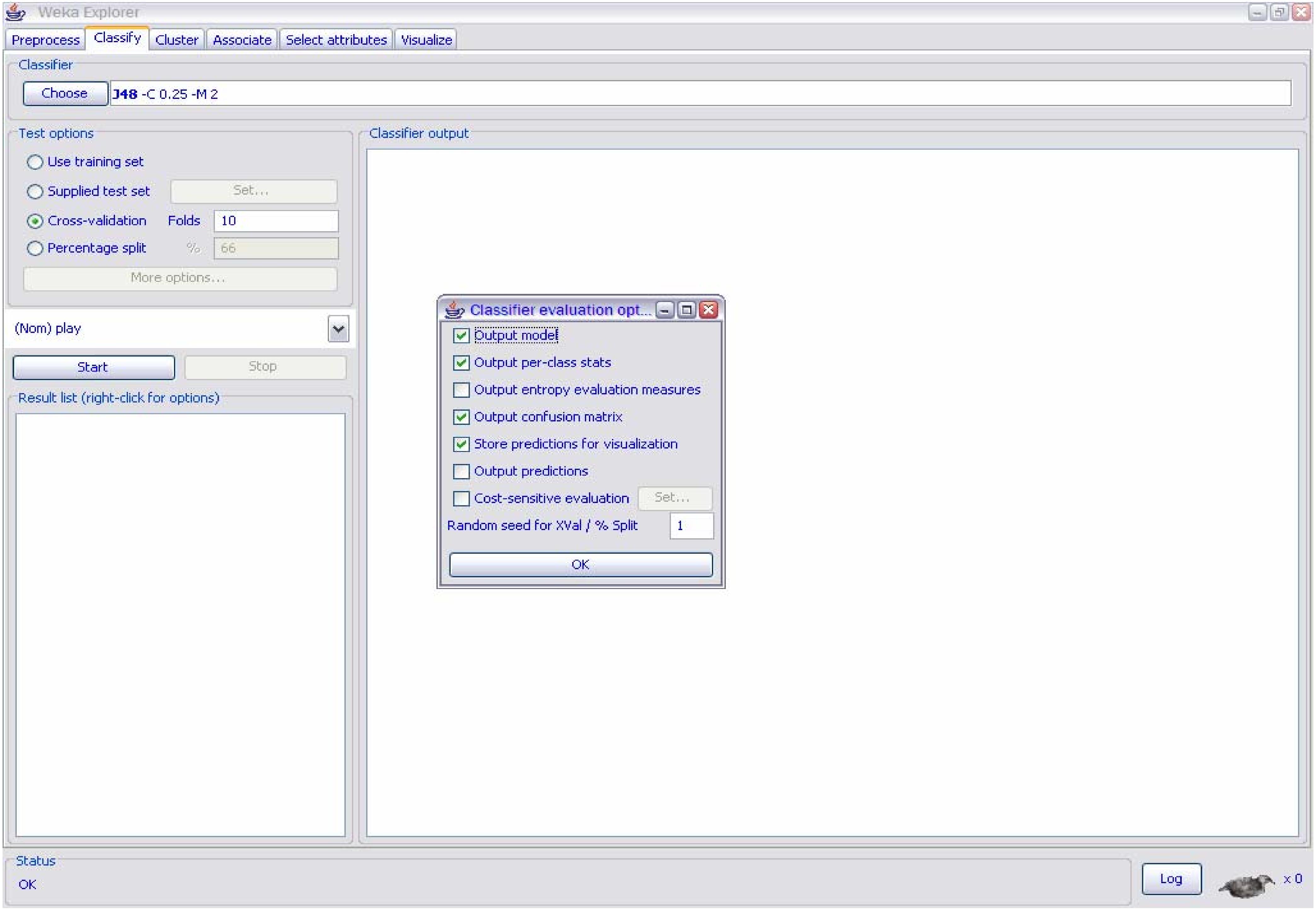
About

Class for generating a pruned or unpruned C4.

| | |
|---|---|
| binarySplits | False |
| confidenceFactor | 0.25 |
| debug | False |
| minNumObj | 2 |
| numFolds | 3 |
| reducedErrorPruning | False |
| saveInstanceData | False |
| seed | 1 |
| subtreeRaising | True |
| unpruned | False |
| useLaplace | False |

Status

OK



Classifier

J48 -C 0.25 -M 2

Test options

○ Use training set

○ Supplied test set

● Cross-validation Folds 10

○ Percentage split % 66

(Nom) play

Classifier evaluation opt...

☑ Output model

☑ Output per-class stats

□ Output entropy evaluation measures

☑ Output confusion matrix

☑ Store predictions for visualization

□ Output predictions

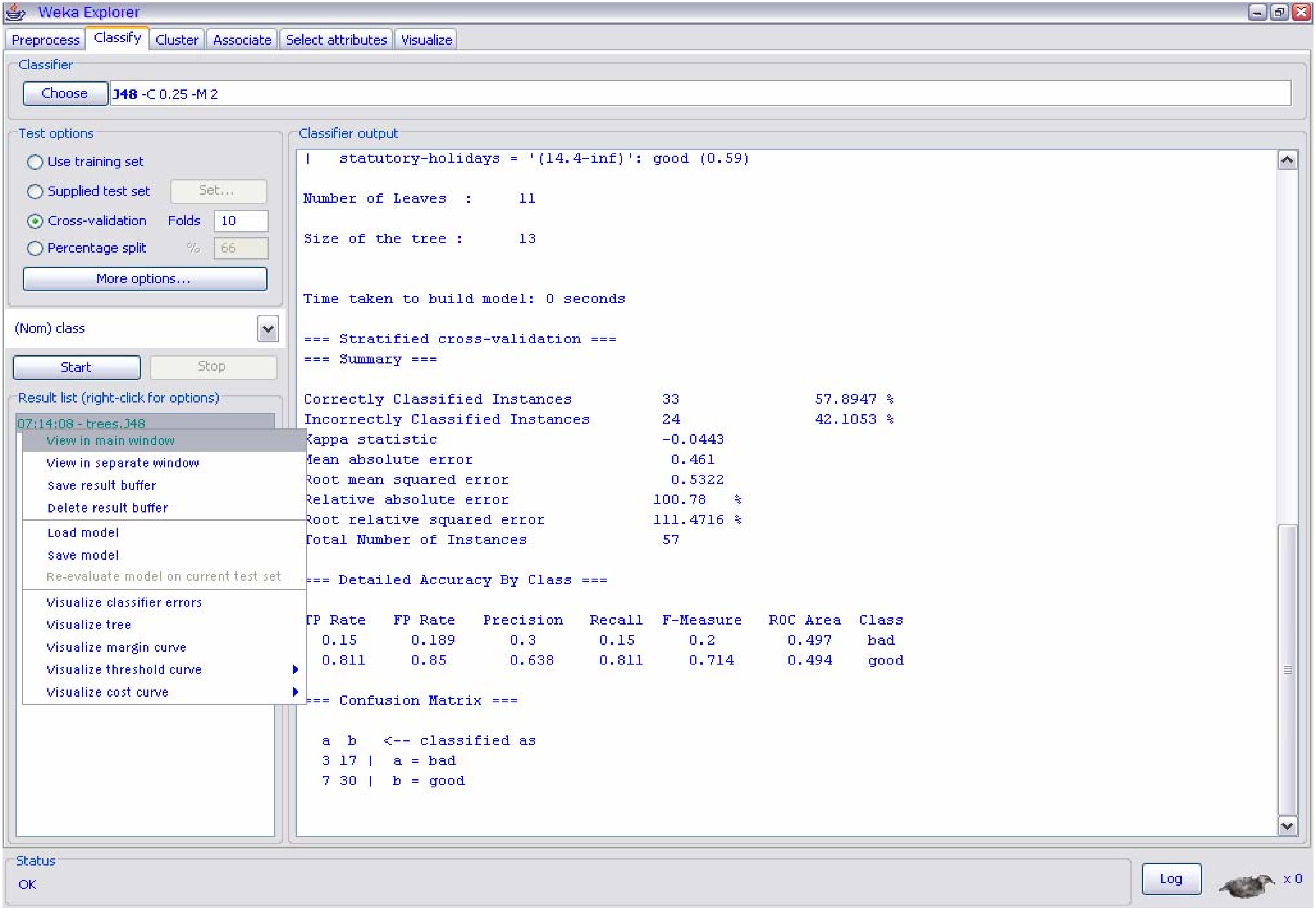
□ Cost-sensitive evaluation Set...

Random seed for XVal / % Split 1

OK

Status

OK



Classifier: J48 -C 0.25 -M 2

Test options

○ Use training set

○ Supplied test set

● Cross-validation Folds 10

○ Percentage split % 66

(Nom) class

Result list (right-click for options)

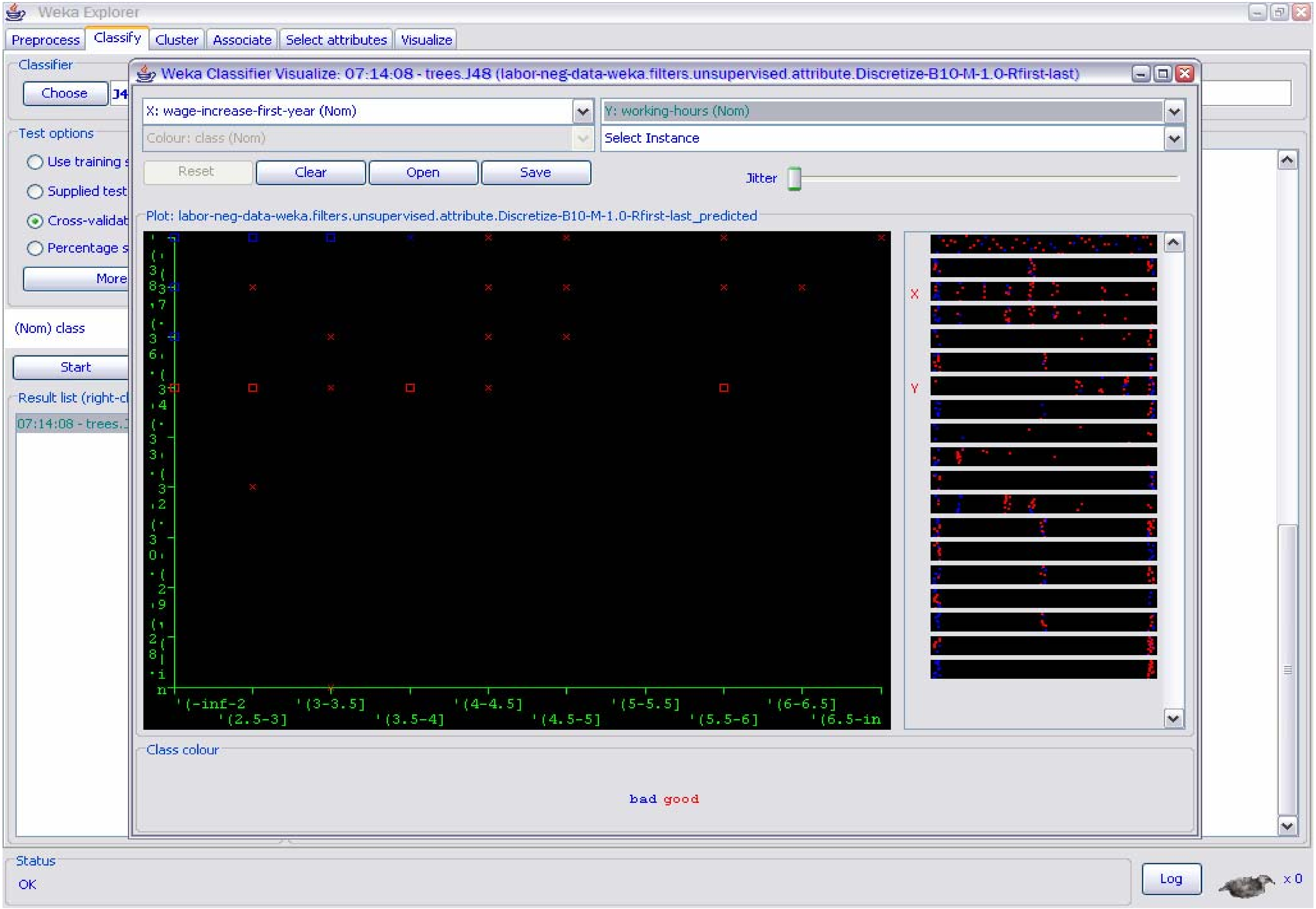
07:14:08 - trees.J48

Classifier output

```
|   statutory-holidays = '(14.4-inf)': good (0.59)

Number of Leaves  :     11

Size of the tree :      13


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          33               57.8947 %
Incorrectly Classified Instances        24               42.1053 %
Kappa statistic                         -0.0443
Mean absolute error                      0.461
Root mean squared error                  0.5322
Relative absolute error                100.78    %
Root relative squared error            111.4716 %
Total Number of Instances               57

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
  0.15      0.189      0.3       0.15      0.2         0.497    bad
  0.811     0.85       0.638     0.811     0.714       0.494    good

=== Confusion Matrix ===

  a  b   <-- classified as
  3 17 |  a = bad
  7 30 |  b = good
```

Status

OK



Weka Explorer

Weka Classifier Visualize: 07:14:08 - trees.J48 (labor-neg-data-weka.filters.unsupervised.attribute.Discretize-B10-M-1.0-Rfirst-last)

X: wage-increase-first-year (Nom)

Y: working-hours (Nom)

Colour: class (Nom)

Select Instance

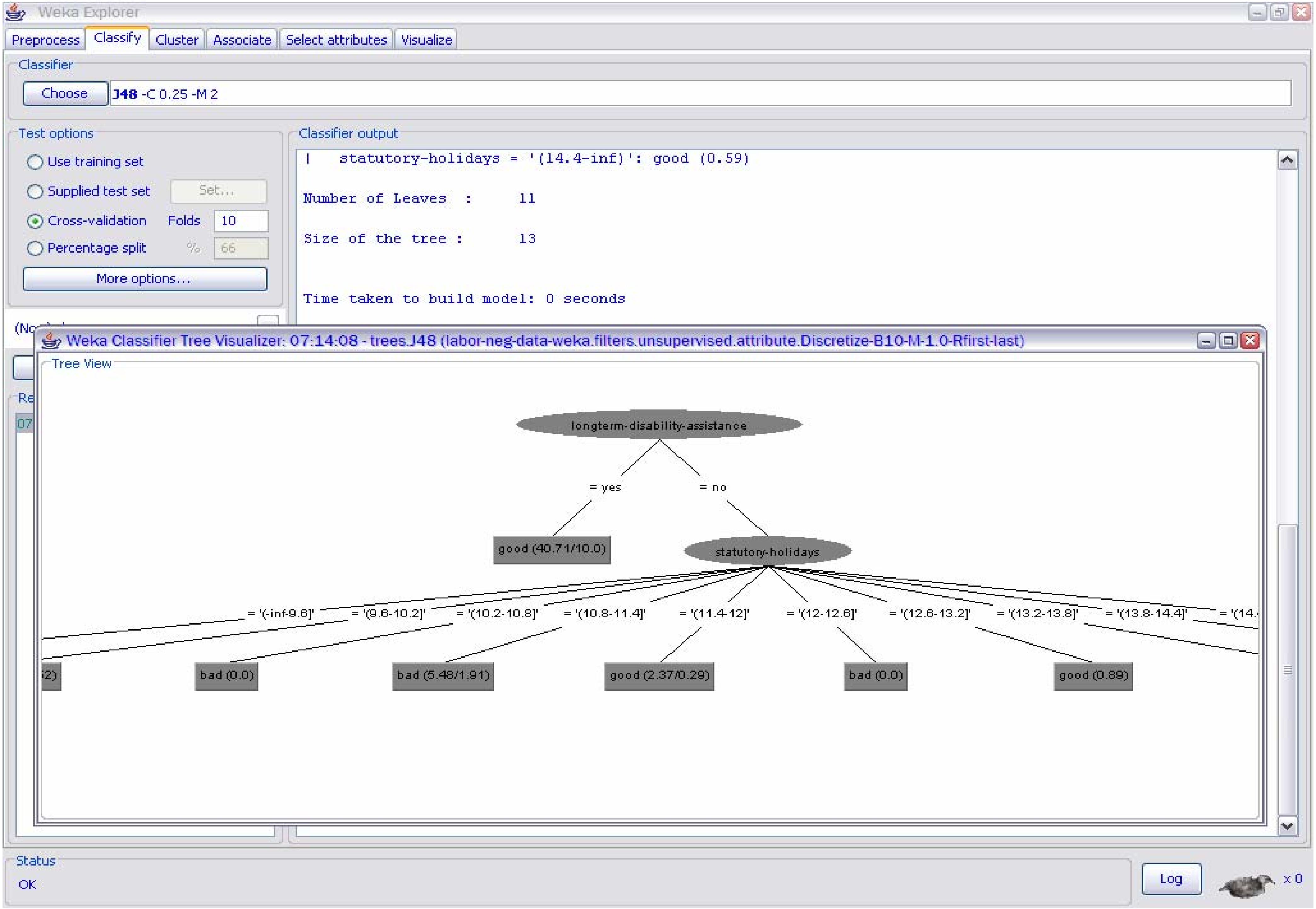
Plot: labor-neg-data-weka.filters.unsupervised.attribute.Discretize-B10-M-1.0-Rfirst-last_predicted

(Nom) class

07:14:08 - trees.J

Status

OK



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...

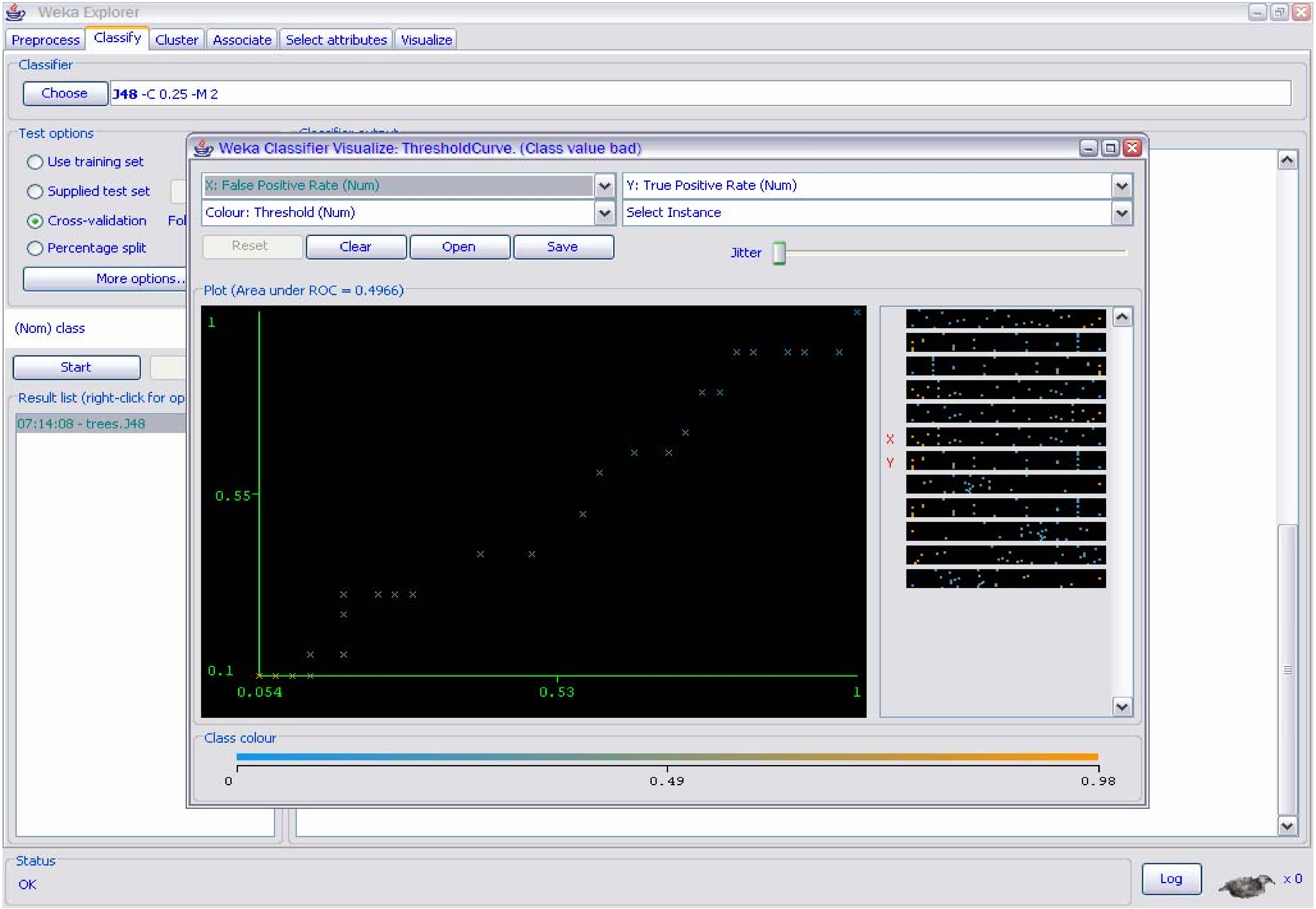
Classifier output

```
|   statutory-holidays = '(14.4-inf)': good (0.59)

Number of Leaves  :     11

Size of the tree :      13


Time taken to build model: 0 seconds
```

Weka Classifier Tree Visualizer: 07:14:08 - trees.J48 (labor-neg-data-weka.filters.unsupervised.attribute.Discretize-B10-M-1.0-Rfirst-last)

Status

OK



Weka Explorer

Classifier: J48 -C 0.25 -M 2

Test options: ○ Use training set ○ Supplied test set ● Cross-validation ○ Percentage split

(Nom) class

Result list (right-click for op

07:14:08 - trees.J48

Weka Classifier Visualize: ThresholdCurve. (Class value bad)

X: False Positive Rate (Num)

Y: True Positive Rate (Num)

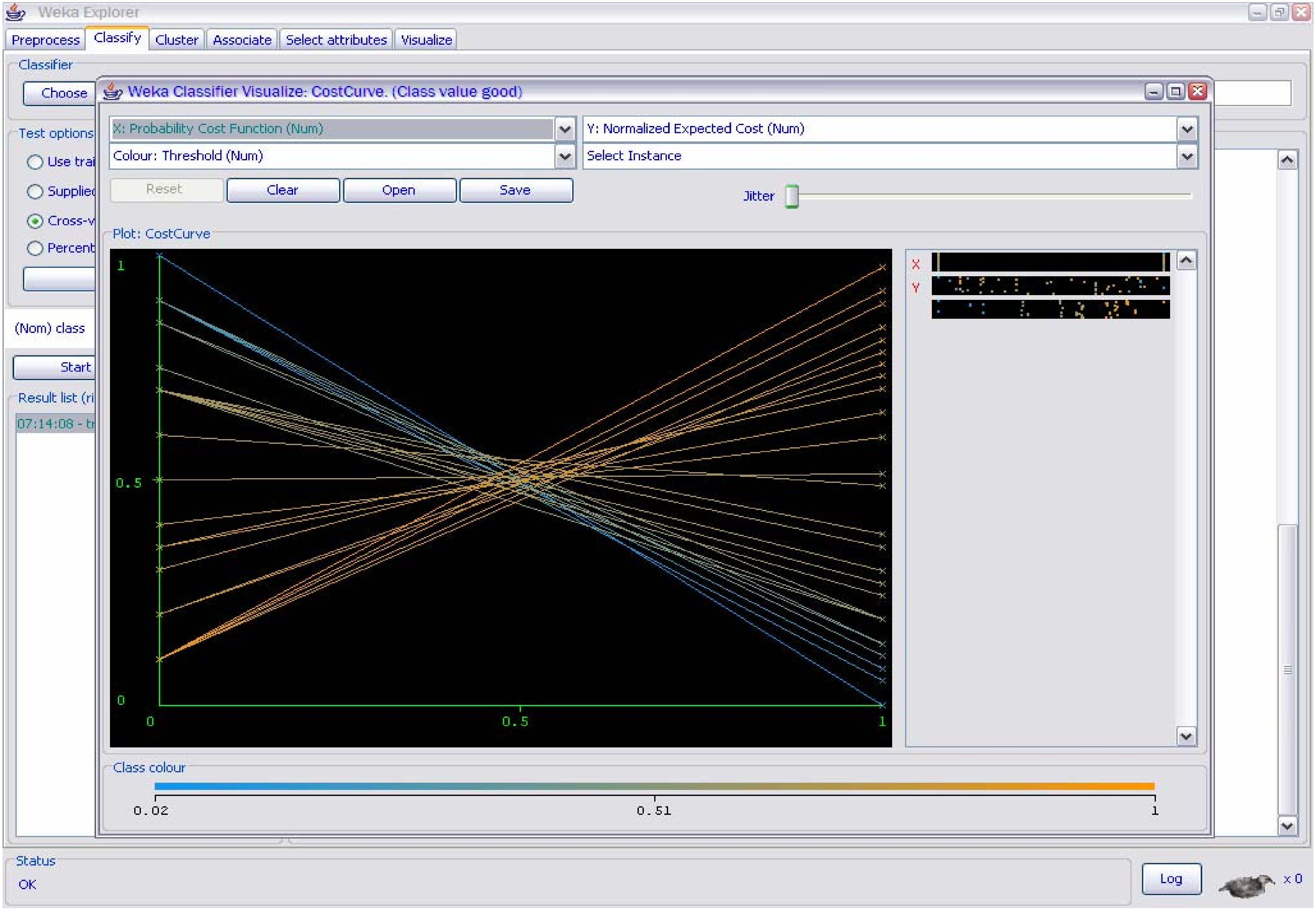
Colour: Threshold (Num)

Select Instance

Plot (Area under ROC = 0.4966)

Status

OK



Weka Explorer

Weka Classifier Visualize: CostCurve. (Class value good)

X: Probability Cost Function (Num)

Y: Normalized Expected Cost (Num)

Colour: Threshold (Num)

Select Instance

Plot: CostCurve

Class colour

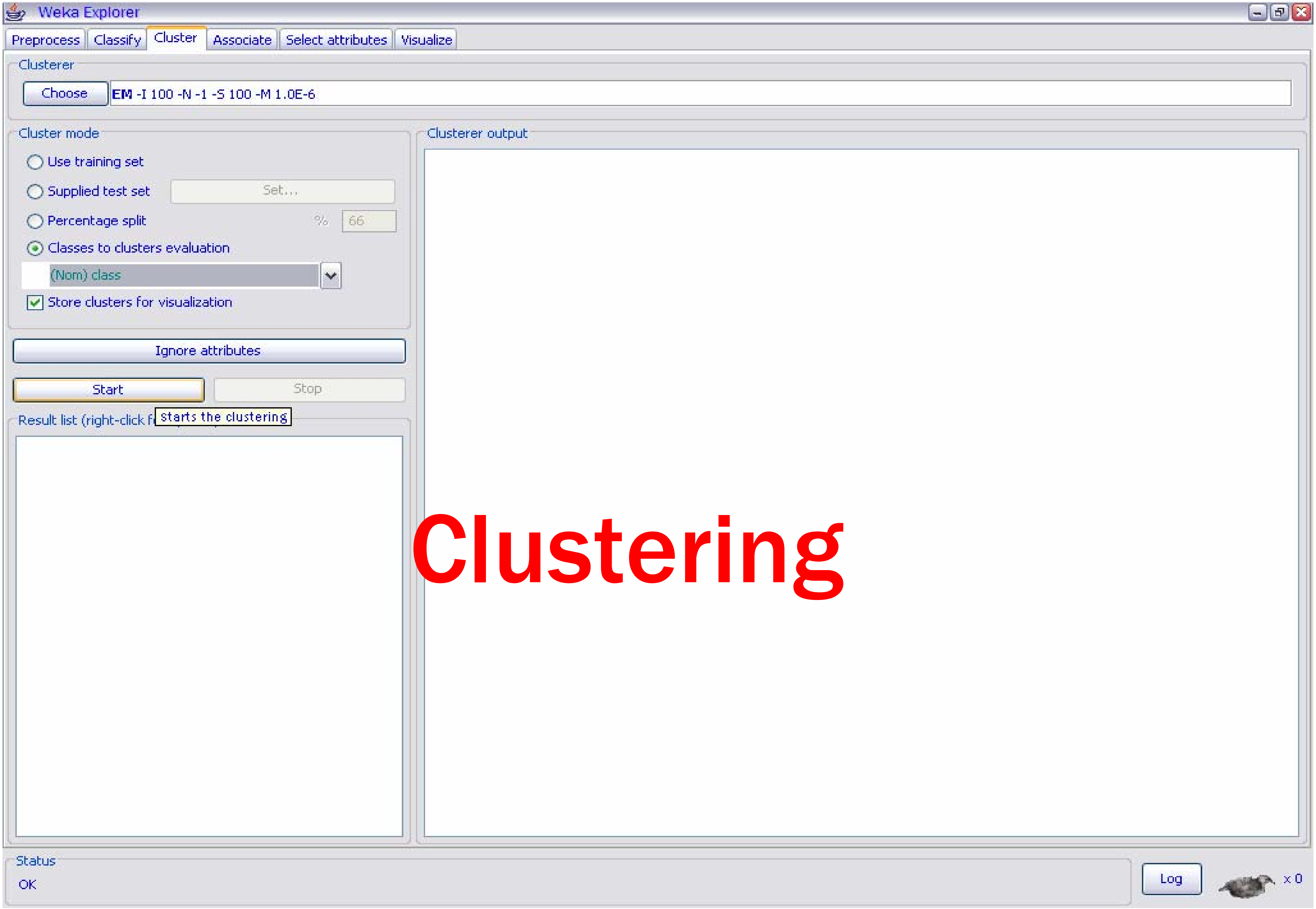
Status

OK

# 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)
- Evaluation based on loglikelihood if clustering scheme produces a probability distribution

# Clustering

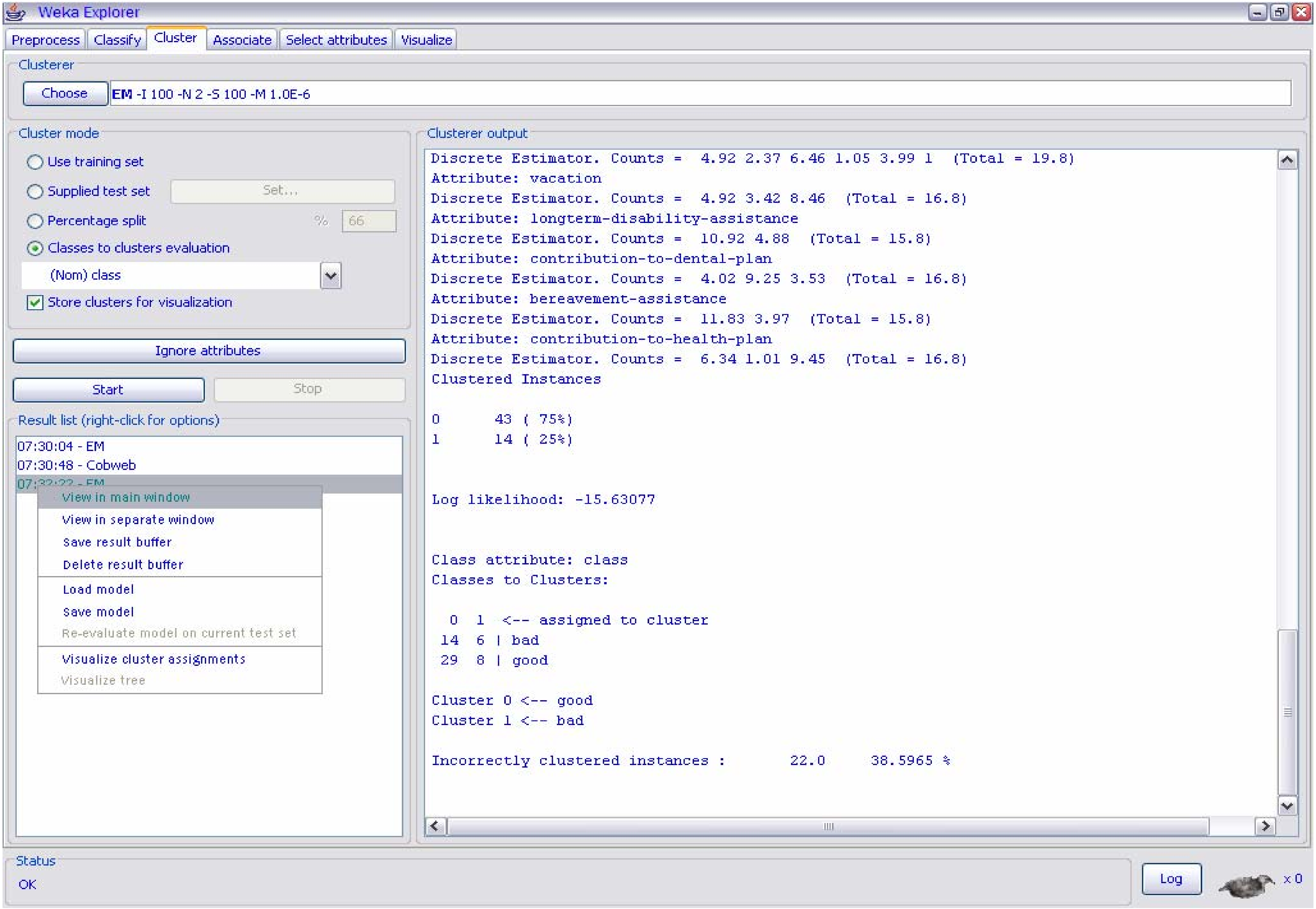


Clusterer: EM -I 100 -N -1 -S 100 -M 1.0E-6

Cluster mode:

- ○ Use training set
- ○ Supplied test set
- ○ Percentage split % 66
- ● Classes to clusters evaluation
  (Nom) class
- ☑ Store clusters for visualization

Status
OK



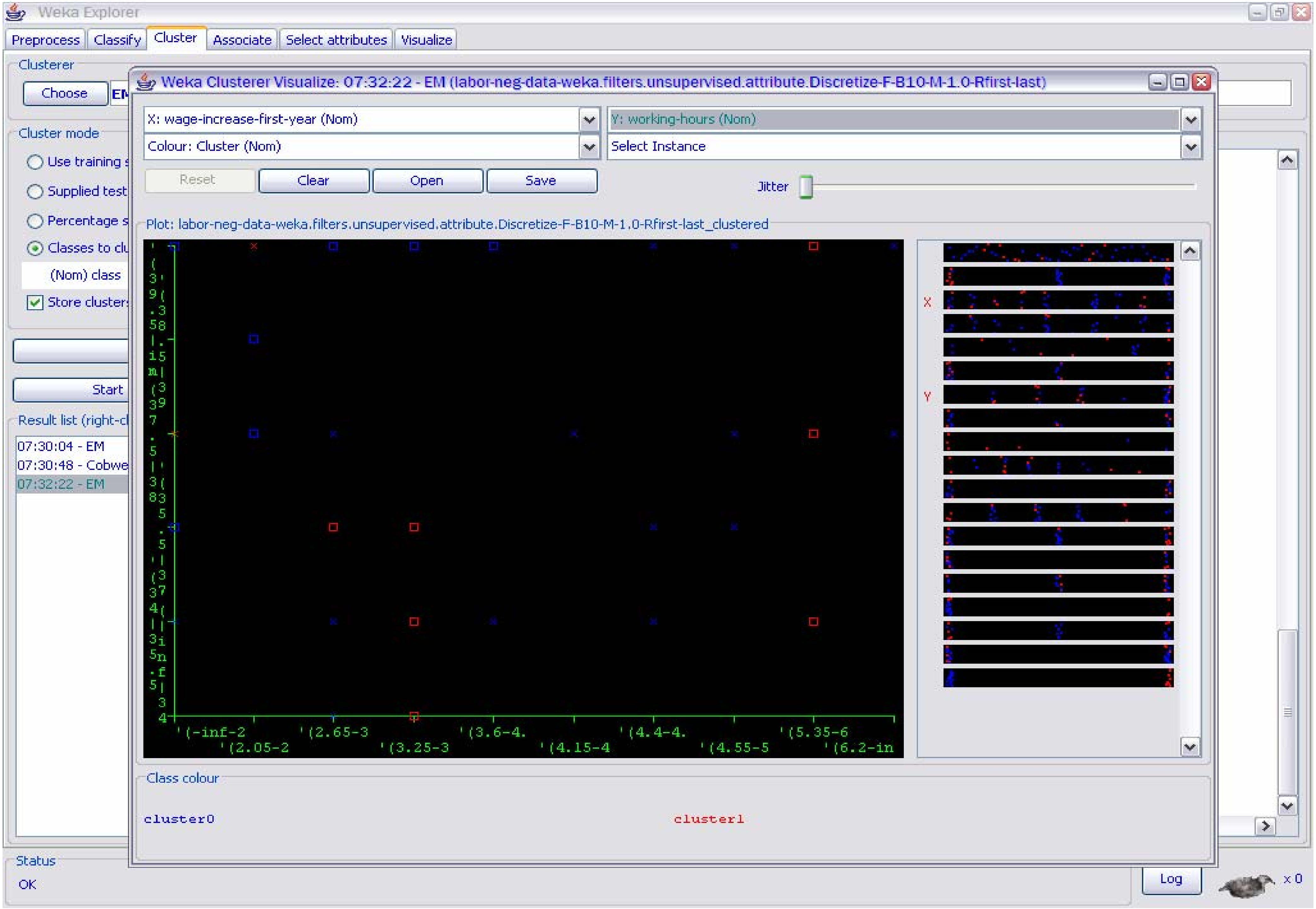
EM -I 100 -N 2 -S 100 -M 1.0E-6

○ Use training set
○ Supplied test set
○ Percentage split % 66
● Classes to clusters evaluation
(Nom) class
☑ Store clusters for visualization

07:30:04 - EM
07:30:48 - Cobweb
07:32:22 - EM

```
Discrete Estimator. Counts =  4.92 2.37 6.46 1.05 3.99 1  (Total = 19.8)
Attribute: vacation
Discrete Estimator. Counts =  4.92 3.42 8.46  (Total = 16.8)
Attribute: longterm-disability-assistance
Discrete Estimator. Counts =  10.92 4.88  (Total = 15.8)
Attribute: contribution-to-dental-plan
Discrete Estimator. Counts =  4.02 9.25 3.53  (Total = 16.8)
Attribute: bereavement-assistance
Discrete Estimator. Counts =  11.83 3.97  (Total = 15.8)
Attribute: contribution-to-health-plan
Discrete Estimator. Counts =  6.34 1.01 9.45  (Total = 16.8)
Clustered Instances

0       43 ( 75%)
1       14 ( 25%)


Log likelihood: -15.63077


Class attribute: class
Classes to Clusters:

  0  1  <-- assigned to cluster
 14  6 | bad
 29  8 | good

Cluster 0 <-- good
Cluster 1 <-- bad

Incorrectly clustered instances :       22.0     38.5965 %
```

OK



Weka Explorer

Weka Clusterer Visualize: 07:32:22 - EM (labor-neg-data-weka.filters.unsupervised.attribute.Discretize-F-B10-M-1.0-Rfirst-last)

X: wage-increase-first-year (Nom)

Y: working-hours (Nom)

Colour: Cluster (Nom)

Plot: labor-neg-data-weka.filters.unsupervised.attribute.Discretize-F-B10-M-1.0-Rfirst-last_clustered

Class colour

cluster0 cluster1

Result list (right-cl

07:30:04 - EM
07:30:48 - Cobwe
07:32:22 - EM

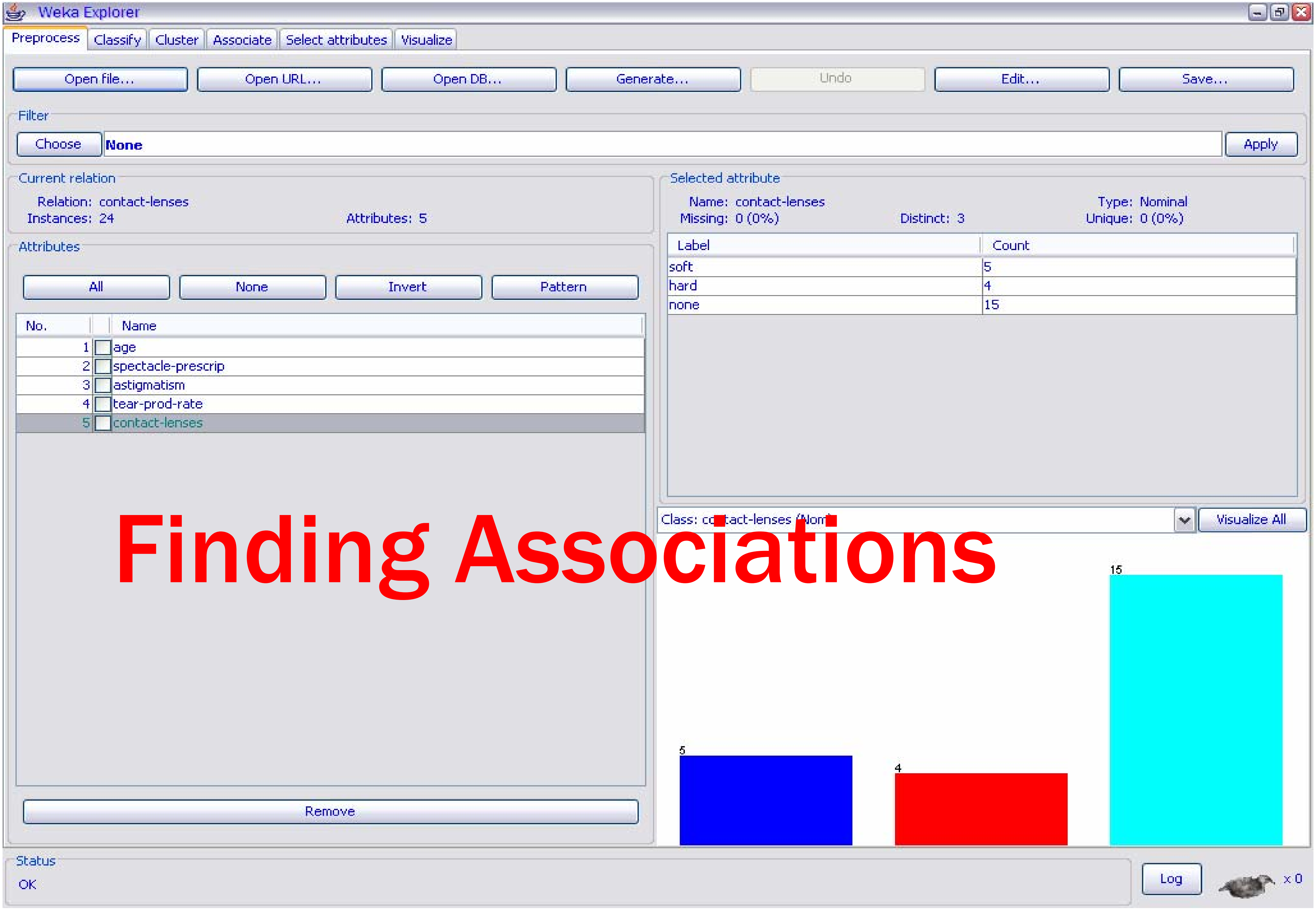
Status

OK

# 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

# Finding Associations



Filter: None

Current relation

Relation: contact-lenses
Instances: 24 Attributes: 5

Attributes

| No. | | Name |
|---|---|---|
| 1 | □ | age |
| 2 | □ | spectacle-prescrip |
| 3 | □ | astigmatism |
| 4 | □ | tear-prod-rate |
| 5 | □ | contact-lenses |

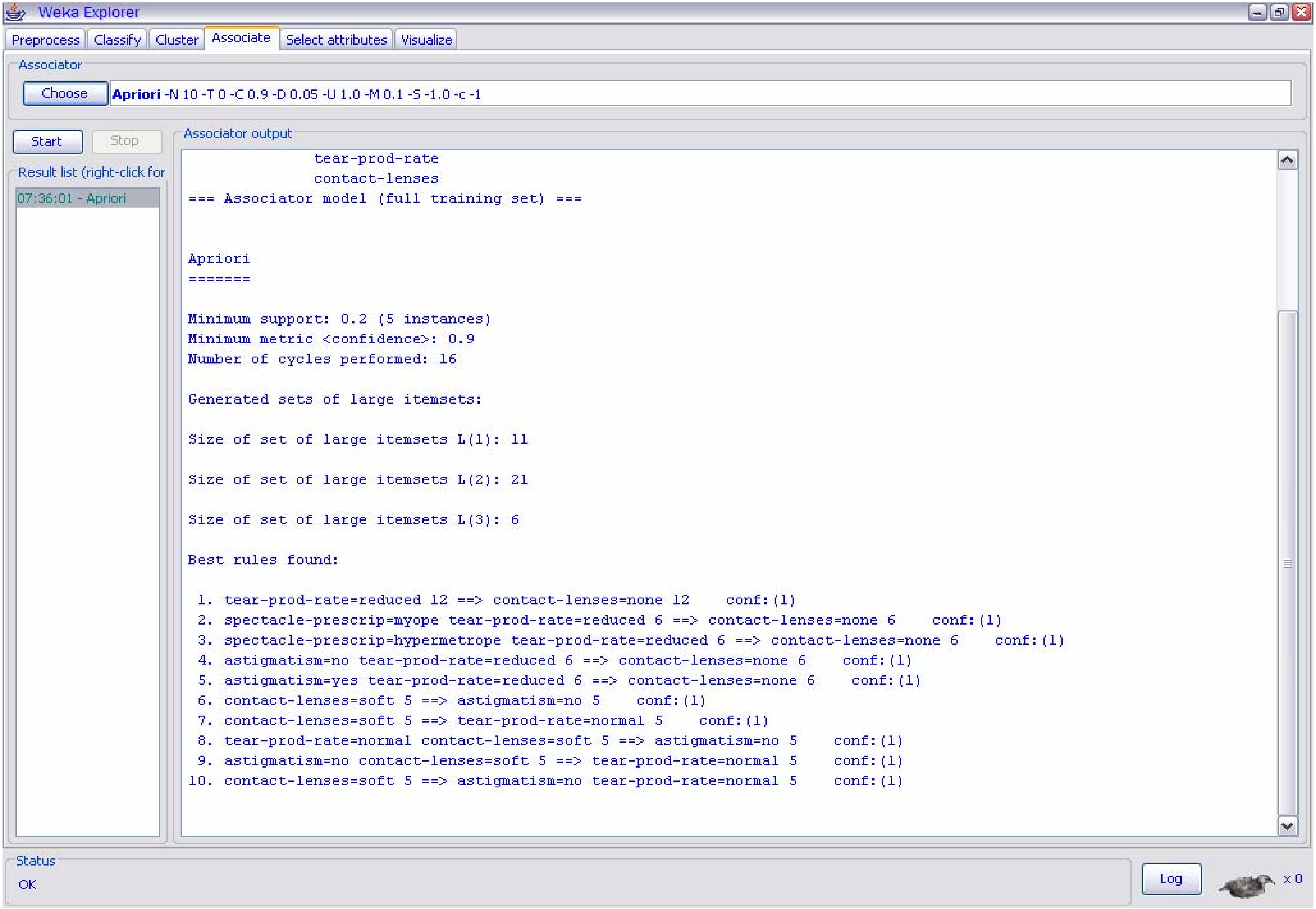
Selected attribute

Name: contact-lenses Type: Nominal
Missing: 0 (0%) Distinct: 3 Unique: 0 (0%)

| Label | Count |
|---|---|
| soft | 5 |
| hard | 4 |
| none | 15 |

Class: contact-lenses (Nom)

Status
OK

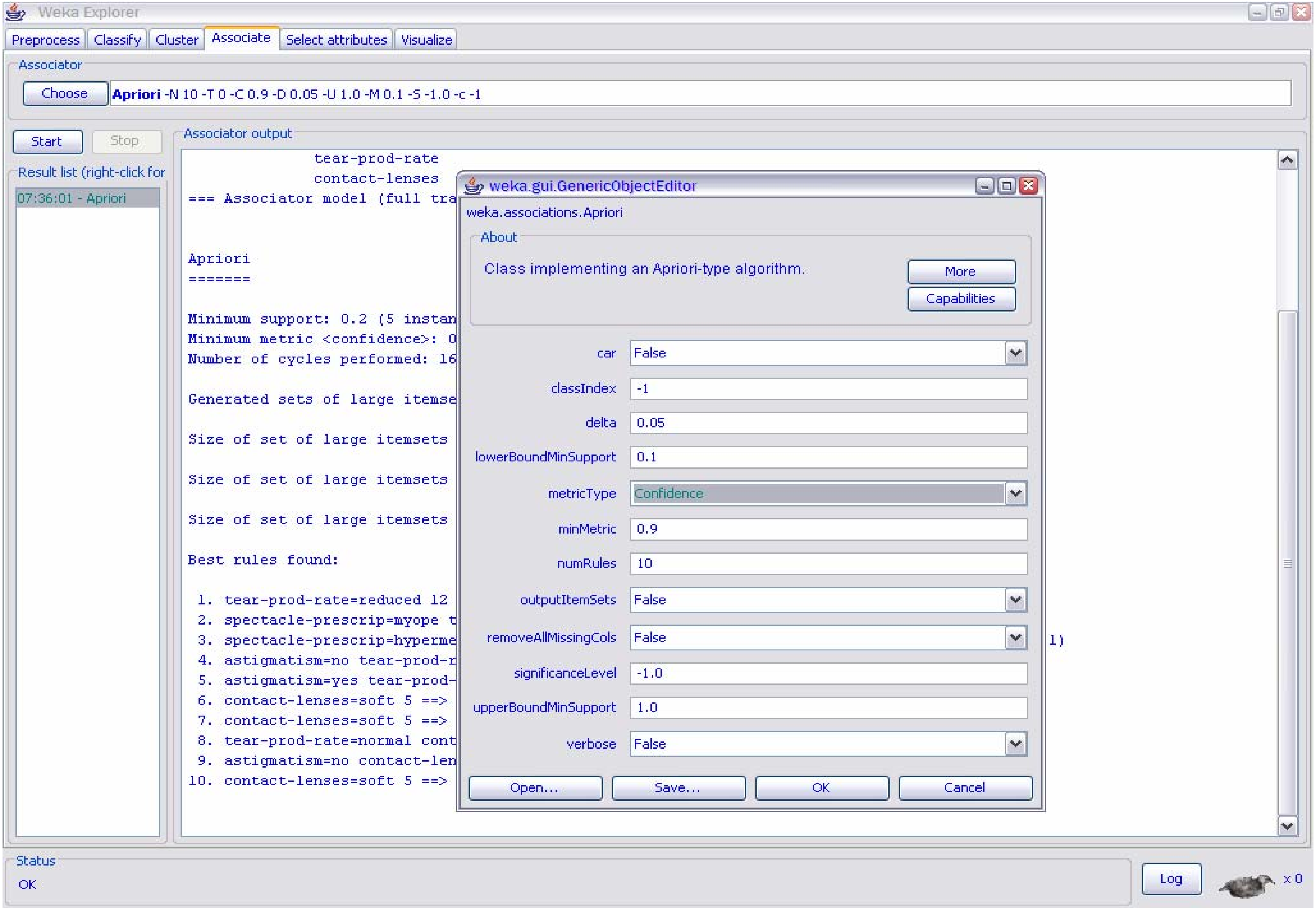


Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

07:36:01 - Apriori

```
              tear-prod-rate
              contact-lenses
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.2 (5 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 16

Generated sets of large itemsets:

Size of set of large itemsets L(1): 11

Size of set of large itemsets L(2): 21

Size of set of large itemsets L(3): 6

Best rules found:

 1. tear-prod-rate=reduced 12 ==> contact-lenses=none 12    conf:(1)
 2. spectacle-prescrip=myope tear-prod-rate=reduced 6 ==> contact-lenses=none 6    conf:(1)
 3. spectacle-prescrip=hypermetrope tear-prod-rate=reduced 6 ==> contact-lenses=none 6    conf:(1)
 4. astigmatism=no tear-prod-rate=reduced 6 ==> contact-lenses=none 6    conf:(1)
 5. astigmatism=yes tear-prod-rate=reduced 6 ==> contact-lenses=none 6    conf:(1)
 6. contact-lenses=soft 5 ==> astigmatism=no 5    conf:(1)
 7. contact-lenses=soft 5 ==> tear-prod-rate=normal 5    conf:(1)
 8. tear-prod-rate=normal contact-lenses=soft 5 ==> astigmatism=no 5    conf:(1)
 9. astigmatism=no contact-lenses=soft 5 ==> tear-prod-rate=normal 5    conf:(1)
10. contact-lenses=soft 5 ==> astigmatism=no tear-prod-rate=normal 5    conf:(1)
```

OK



Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

07:36:01 - Apriori

```
                 tear-prod-rate
                 contact-lenses
=== Associator model (full tra


Apriori
=======

Minimum support: 0.2 (5 instan
Minimum metric <confidence>: 0
Number of cycles performed: 16

Generated sets of large itemse

Size of set of large itemsets

Size of set of large itemsets

Size of set of large itemsets

Best rules found:

 1. tear-prod-rate=reduced 12
 2. spectacle-prescrip=myope t
 3. spectacle-prescrip=hyperme                                                      1)
 4. astigmatism=no tear-prod-r
 5. astigmatism=yes tear-prod-
 6. contact-lenses=soft 5 ==>
 7. contact-lenses=soft 5 ==>
 8. tear-prod-rate=normal cont
 9. astigmatism=no contact-len
10. contact-lenses=soft 5 ==>
```

weka.gui.GenericObjectEditor

weka.associations.Apriori

About

Class implementing an Apriori-type algorithm.

| | |
|---|---|
| car | False |
| classIndex | -1 |
| delta | 0.05 |
| lowerBoundMinSupport | 0.1 |
| metricType | Confidence |
| minMetric | 0.9 |
| numRules | 10 |
| outputItemSets | False |
| removeAllMissingCols | False |
| significanceLevel | -1.0 |
| upperBoundMinSupport | 1.0 |
| verbose | False |

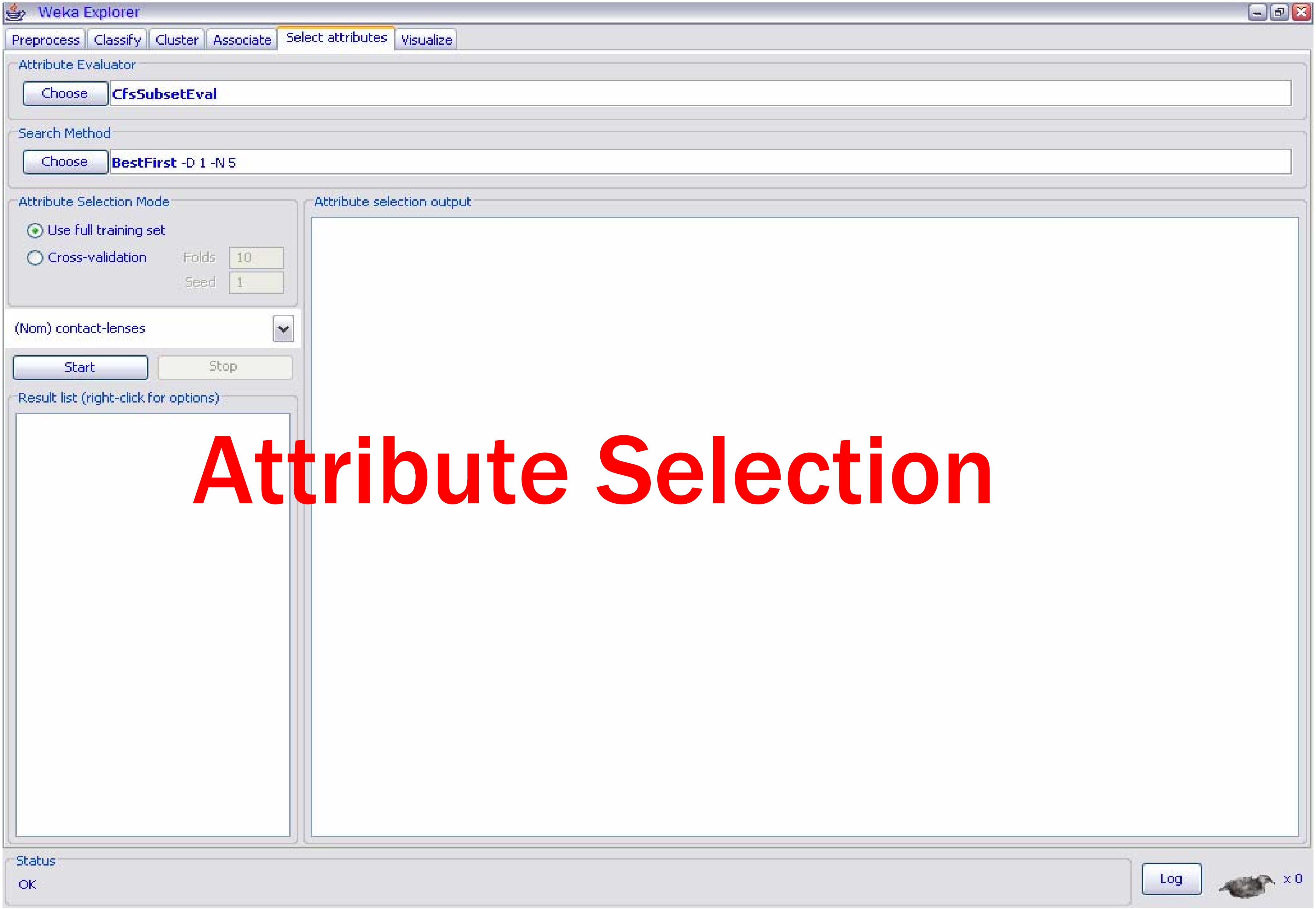
Status

OK

# Explorer: attribute selection

- Panel that can be used to investigate which (subsets of) attributes are the most predictive ones
- Attribute selection methods contain two parts:
  - A search method: best-first, forward selection, random, exhaustive, genetic algorithm, ranking
  - An evaluation method: correlation-based, wrapper, information gain, chi-squared, …
- Very flexible: WEKA allows (almost) arbitrary combinations of these two

# Attribute Selection



Attribute Evaluator: CfsSubsetEval

Search Method: BestFirst -D 1 -N 5

Attribute Selection Mode

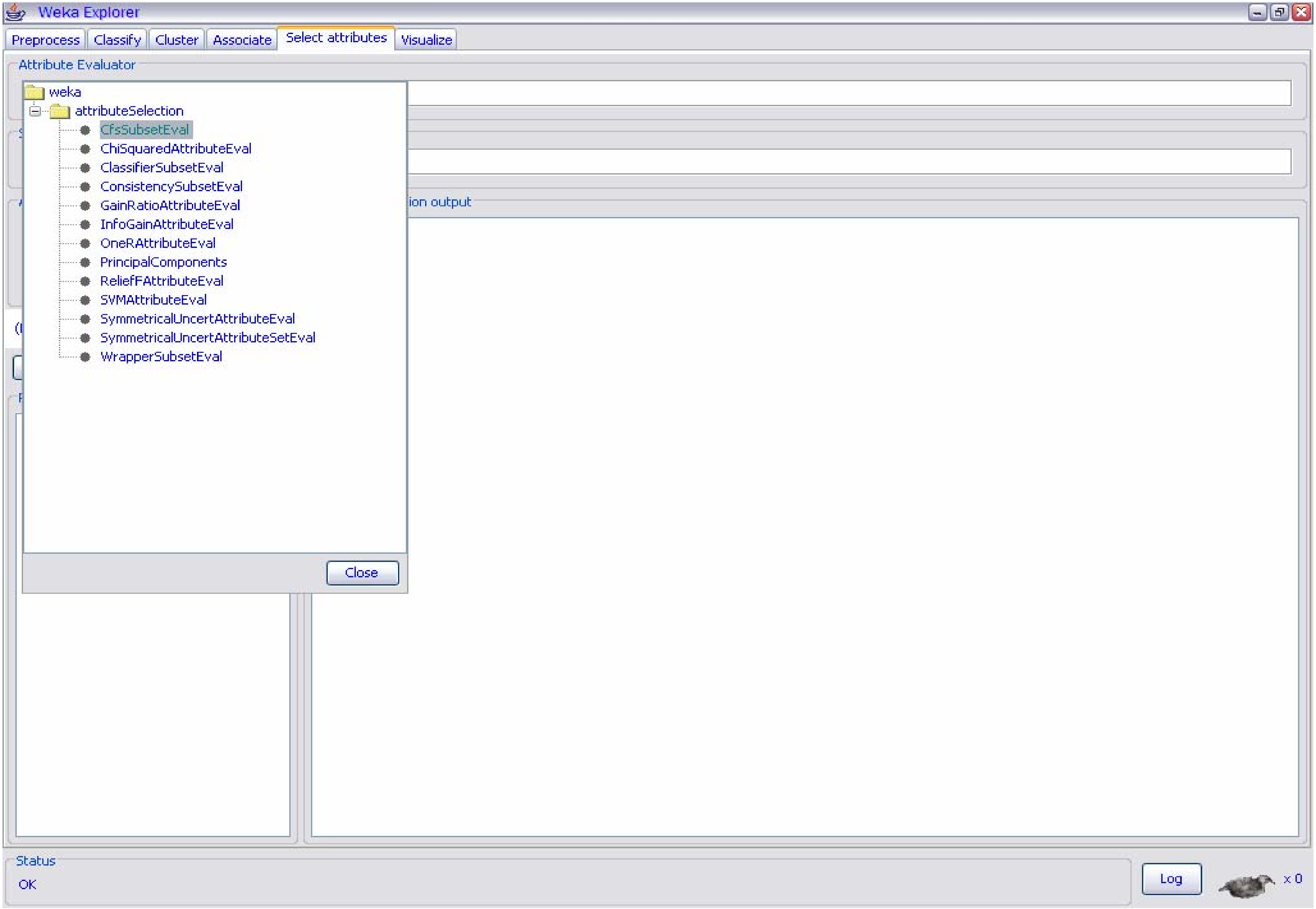
● Use full training set

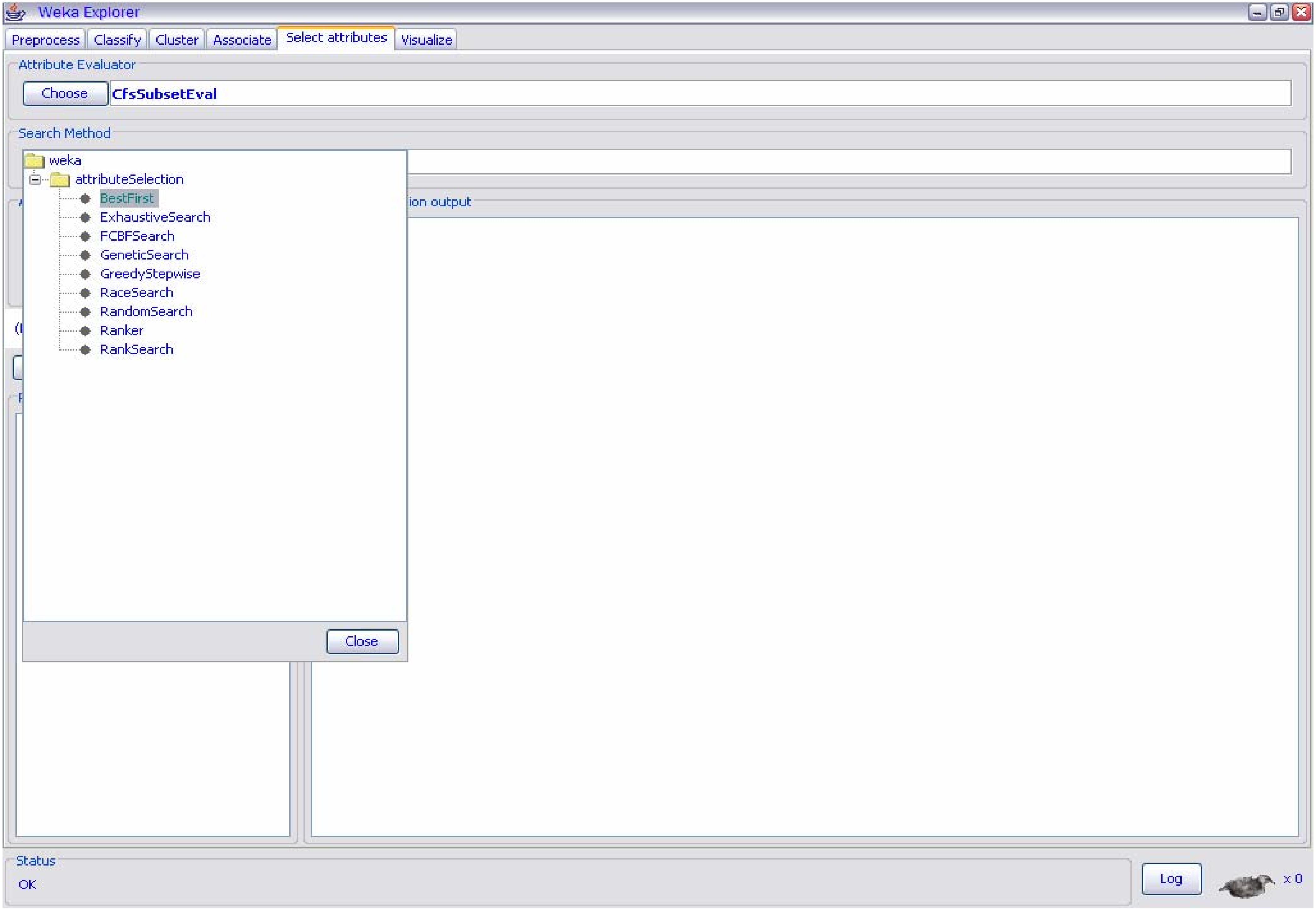
○ Cross-validation Folds 10 Seed 1

(Nom) contact-lenses

Status

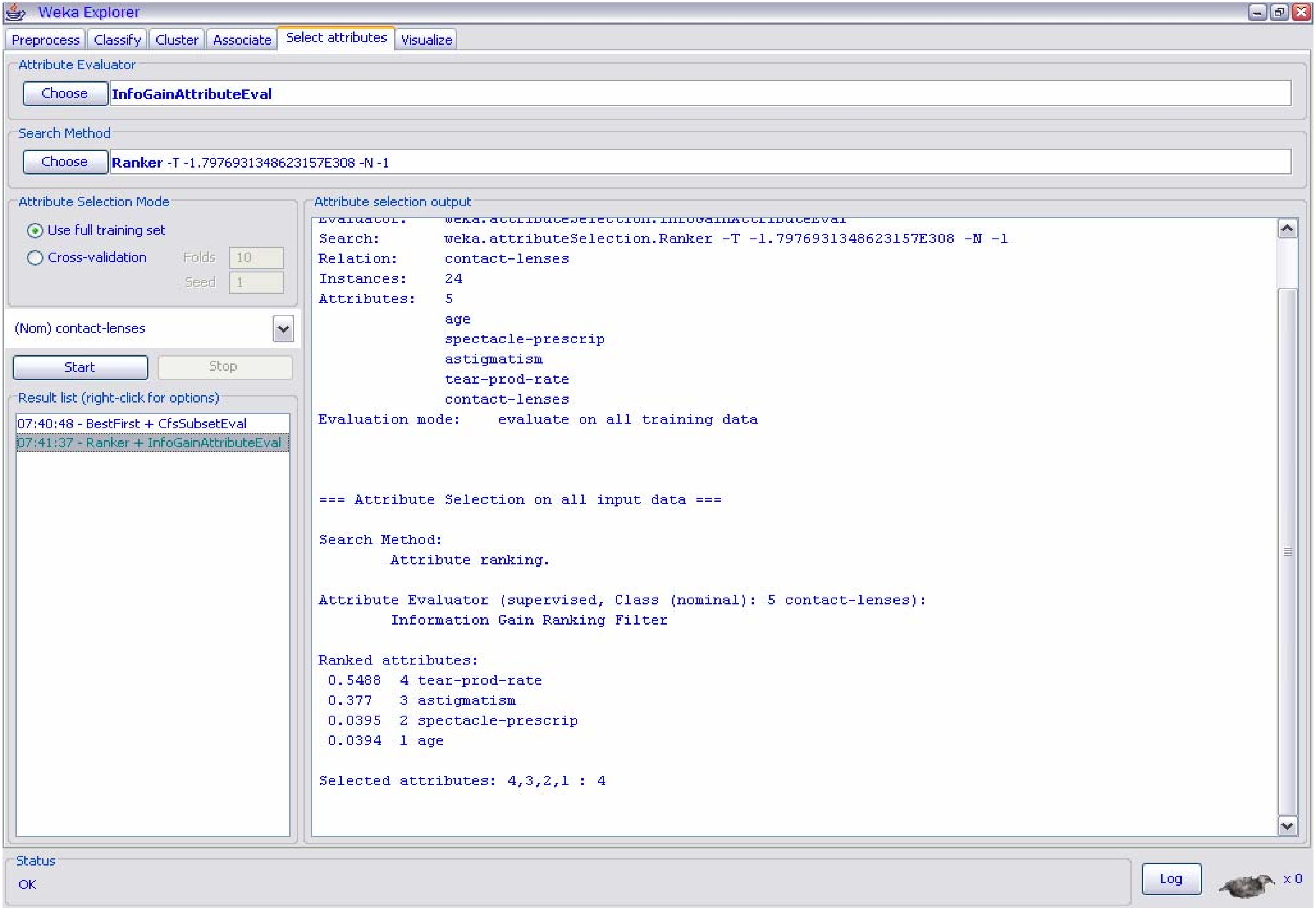
OK


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Attribute Evaluator
weka
attributeSelection
CfsSubsetEval
ChiSquaredAttributeEval
ClassifierSubsetEval
ConsistencySubsetEval
GainRatioAttributeEval
InfoGainAttributeEval
OneRAttributeEval
PrincipalComponents
ReliefFAttributeEval
SVMAttributeEval
SymmetricalUncertAttributeEval
SymmetricalUncertAttributeSetEval
WrapperSubsetEval
Close
ion output
Status
OK
Log
x 0



Attribute Evaluator: Choose CfsSubsetEval

Status: OK



Attribute Evaluator

InfoGainAttributeEval

Search Method

Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode

● Use full training set

○ Cross-validation    Folds 10    Seed 1

(Nom) contact-lenses

Result list (right-click for options)

07:40:48 - BestFirst + CfsSubsetEval

07:41:37 - Ranker + InfoGainAttributeEval

Attribute selection output

```
Evaluator:    weka.attributeSelection.InfoGainAttributeEval
Search:       weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1
Relation:     contact-lenses
Instances:    24
Attributes:   5
              age
              spectacle-prescrip
              astigmatism
              tear-prod-rate
              contact-lenses
Evaluation mode:    evaluate on all training data



=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 5 contact-lenses):
        Information Gain Ranking Filter

Ranked attributes:
 0.5488  4 tear-prod-rate
 0.377   3 astigmatism
 0.0395  2 spectacle-prescrip
 0.0394  1 age

Selected attributes: 4,3,2,1 : 4
```

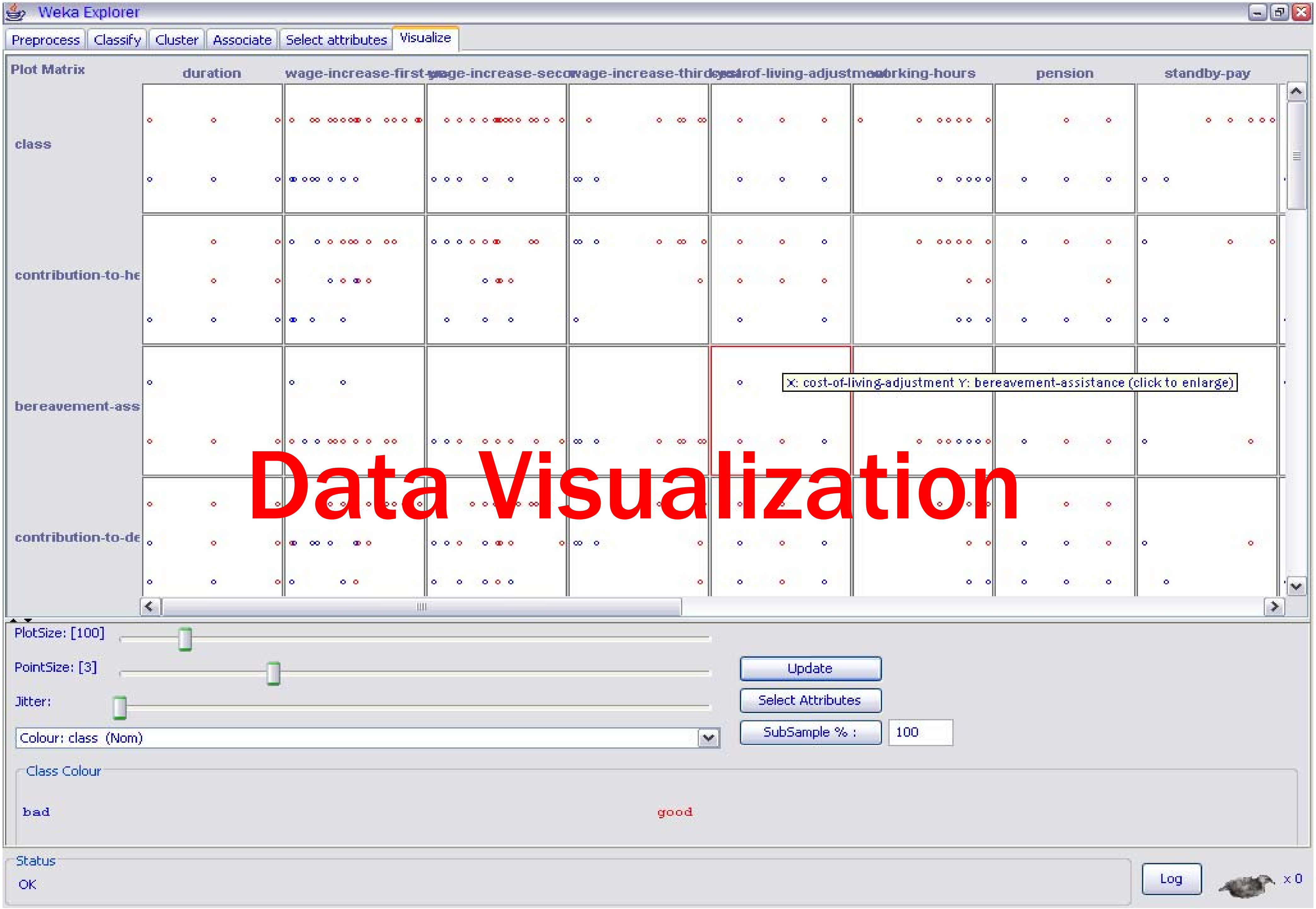
Status

OK

# Explorer: data visualization

- Visualization very useful in practice: e.g. helps to determine difficulty of the learning problem
- WEKA can visualize single attributes (1-d) and pairs of attributes (2-d)
  - To do: rotating 3-d visualizations (Xgobi-style)
- Color-coded class values
- “Jitter” option to deal with nominal attributes (and to detect “hidden” data points)
- “Zoom-in” function

# Data Visualization



PlotSize: [100]

PointSize: [3]

Jitter:

Colour: class (Nom)

SubSample % : 100

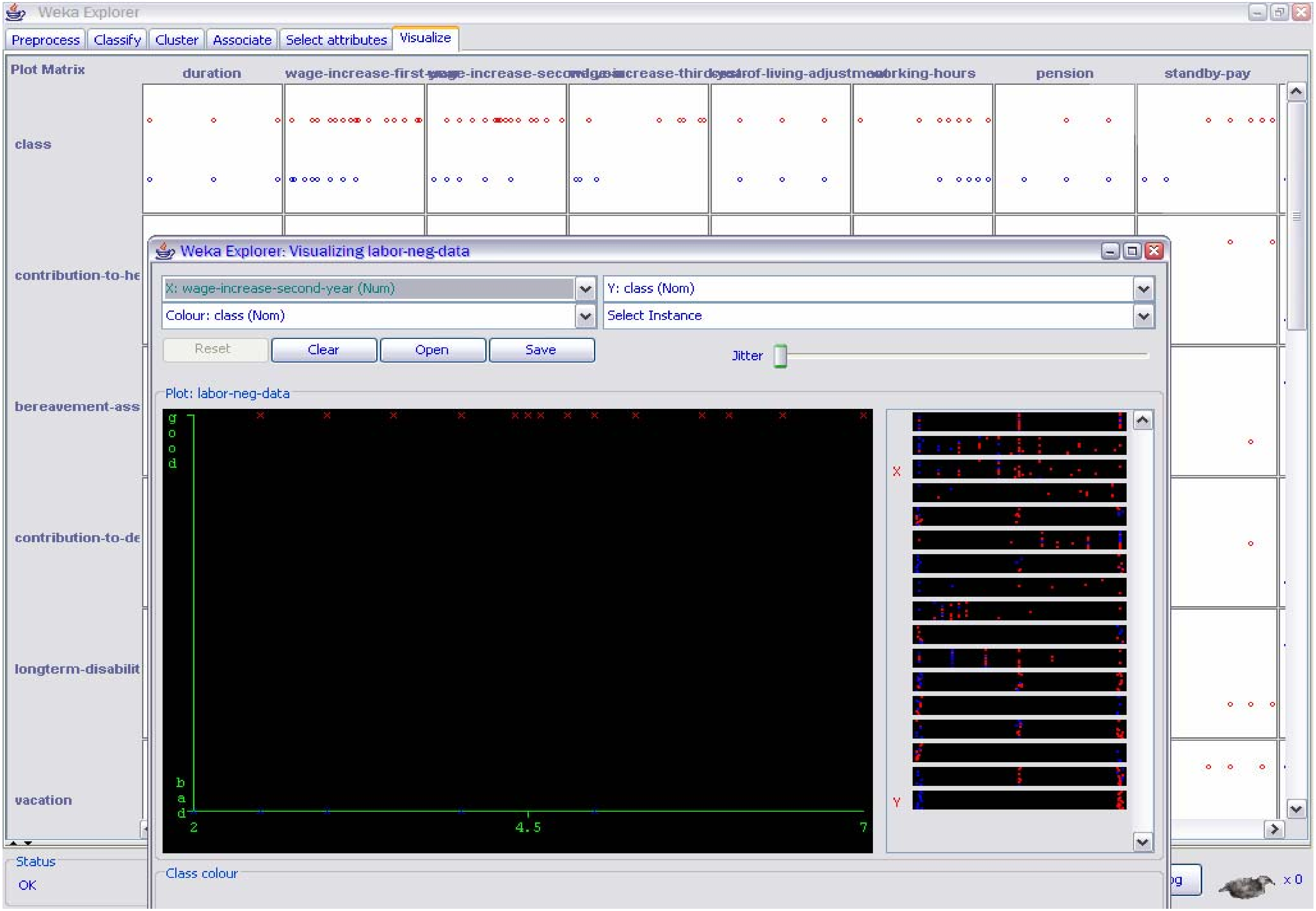
Class Colour

bad

good

Status

OK

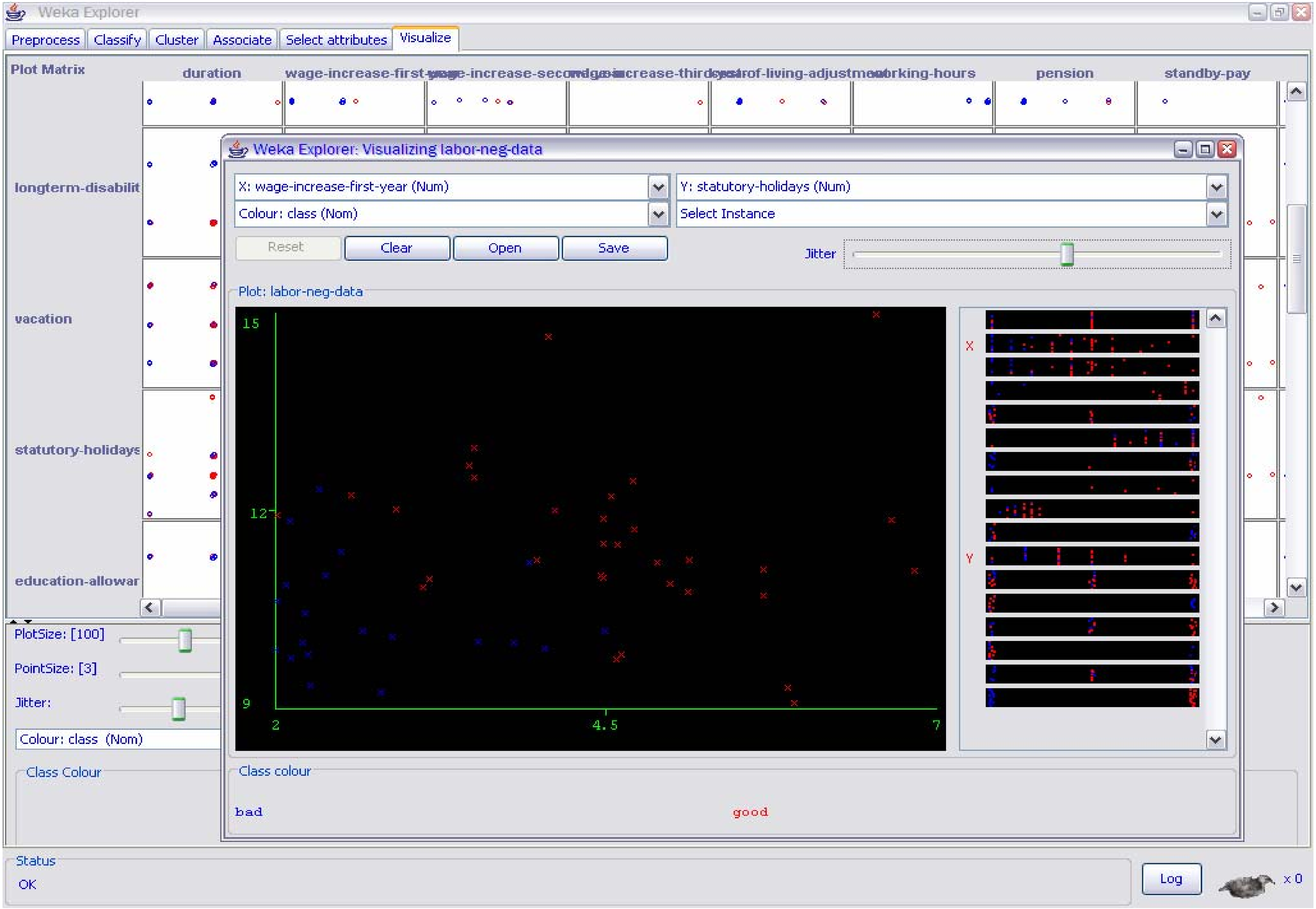


X: wage-increase-second-year (Num)

Y: class (Nom)

Colour: class (Nom)

Select Instance



Weka Explorer

Weka Explorer: Visualizing labor-neg-data

X: wage-increase-first-year (Num)

Y: statutory-holidays (Num)

Colour: class (Nom)

Select Instance

Plot: labor-neg-data

PlotSize: [100]

PointSize: [3]

Jitter:

Colour: class (Nom)

Status

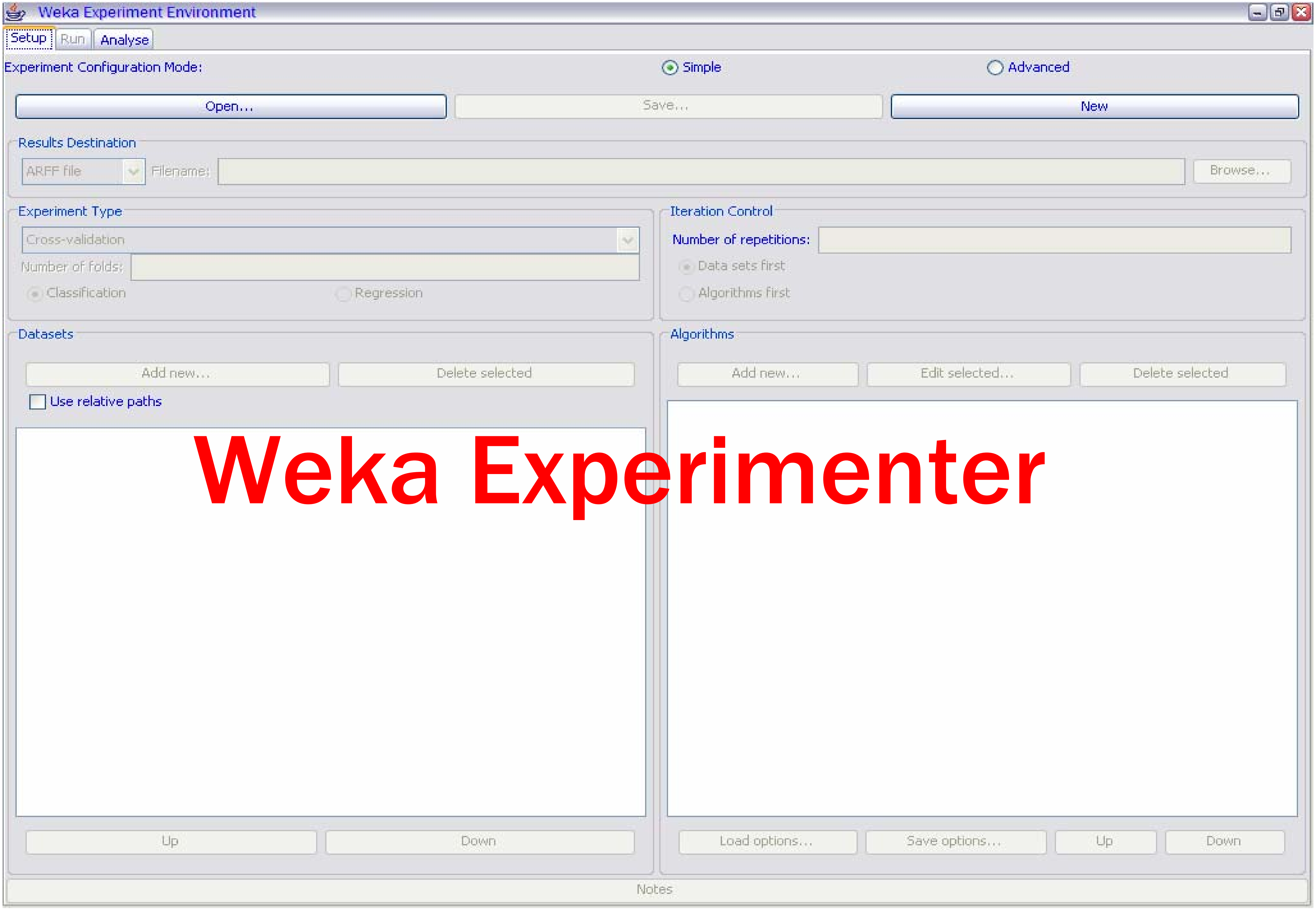
OK

# **Part II:** experiment administrations in Weka

# Performing experiments

- Experimenter 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!

# Weka Experimenter



Weka Experiment Environment

Experiment Configuration Mode: ● Simple ○ Advanced

Results Destination: ARFF file  Filename:

Experiment Type: Cross-validation

Number of folds:

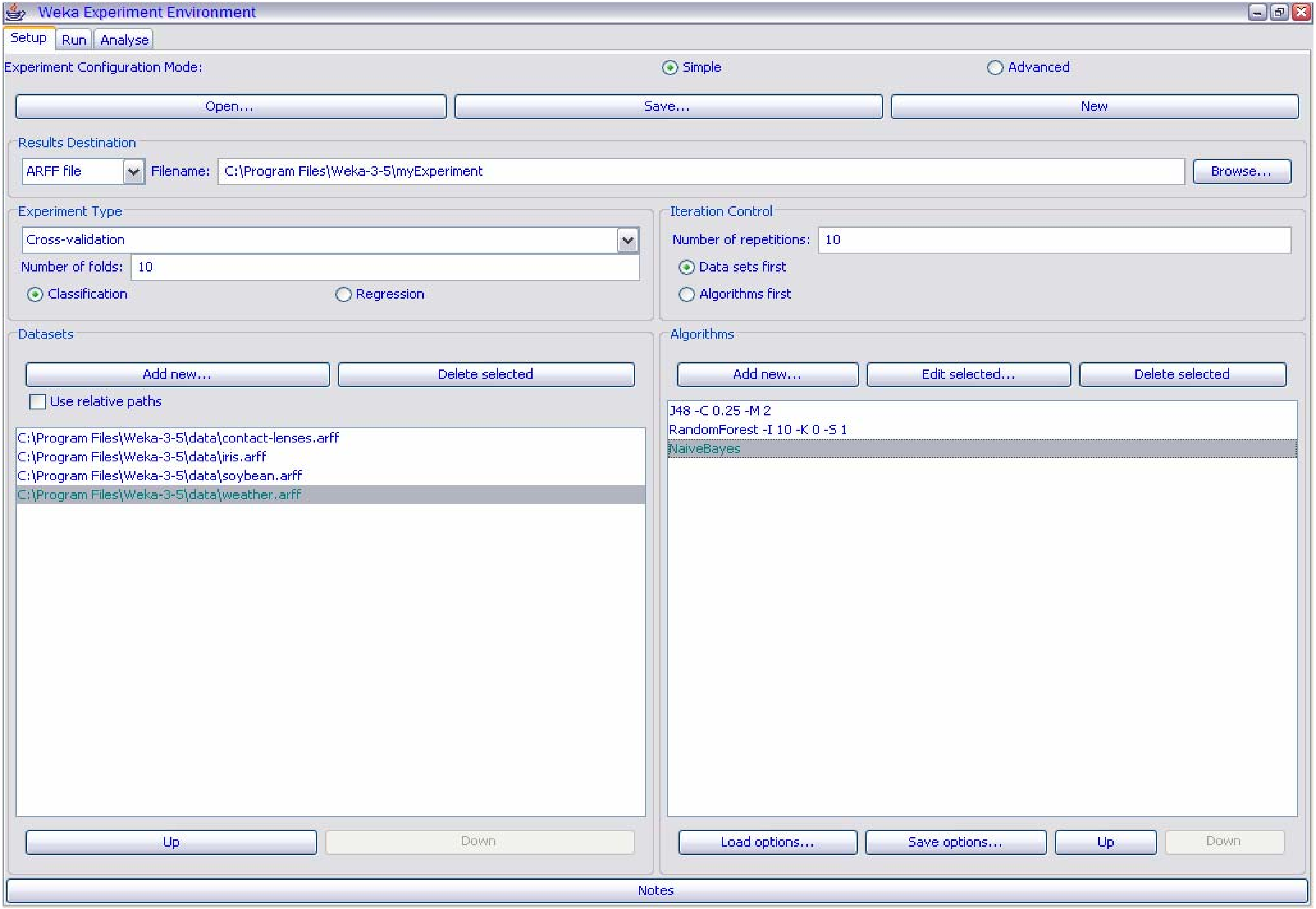
● Classification ○ Regression

Iteration Control

Number of repetitions:

● Data sets first ○ Algorithms first

□ Use relative paths



Experiment Configuration Mode: ● Simple ○ Advanced

Results Destination

ARFF file Filename: C:\Program Files\Weka-3-5\myExperiment

Experiment Type

Cross-validation

Number of folds: 10

● Classification ○ Regression

Iteration Control

Number of repetitions: 10

● Data sets first

○ Algorithms first

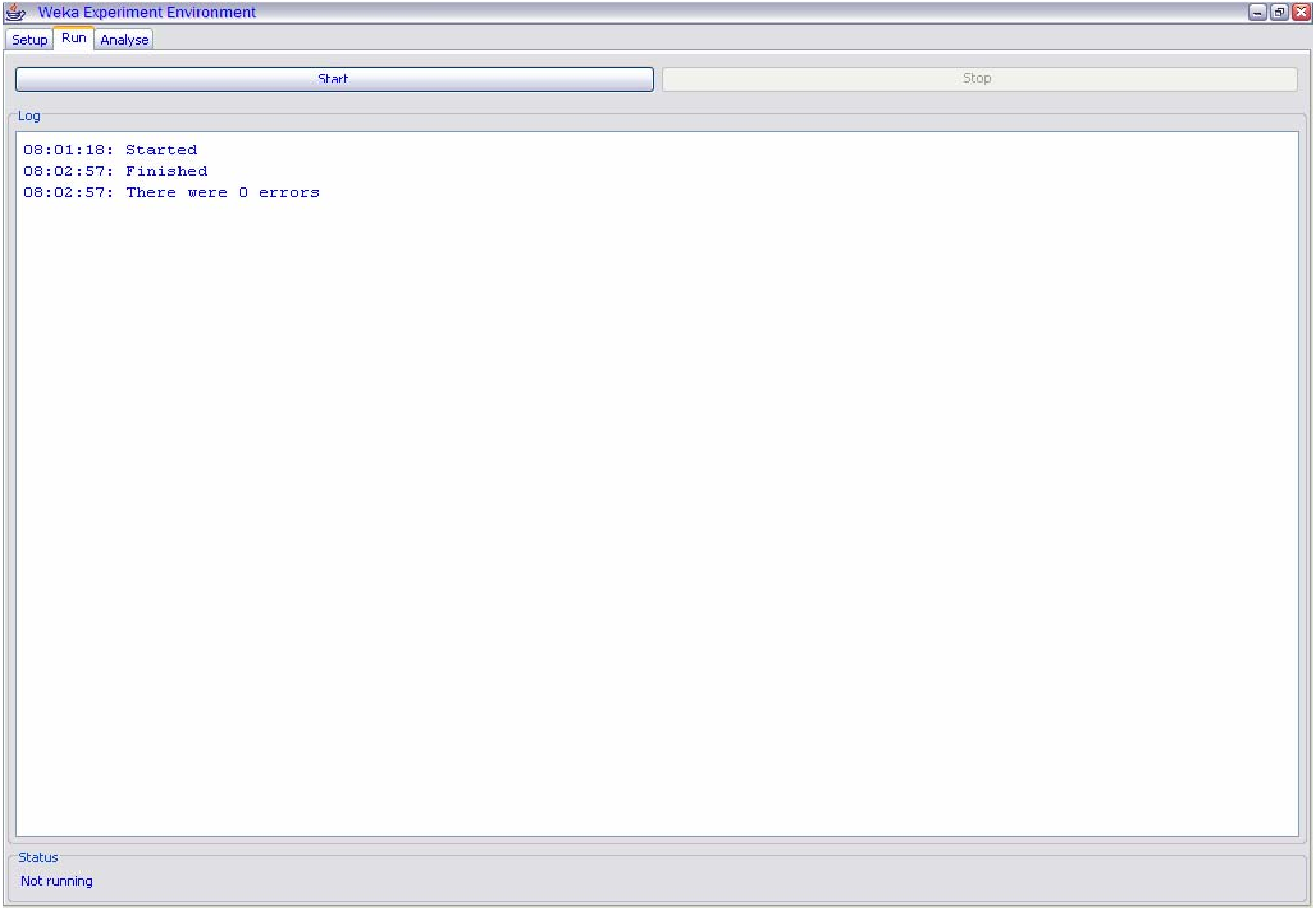
Datasets

□ Use relative paths

C:\Program Files\Weka-3-5\data\contact-lenses.arff
C:\Program Files\Weka-3-5\data\iris.arff
C:\Program Files\Weka-3-5\data\soybean.arff
C:\Program Files\Weka-3-5\data\weather.arff

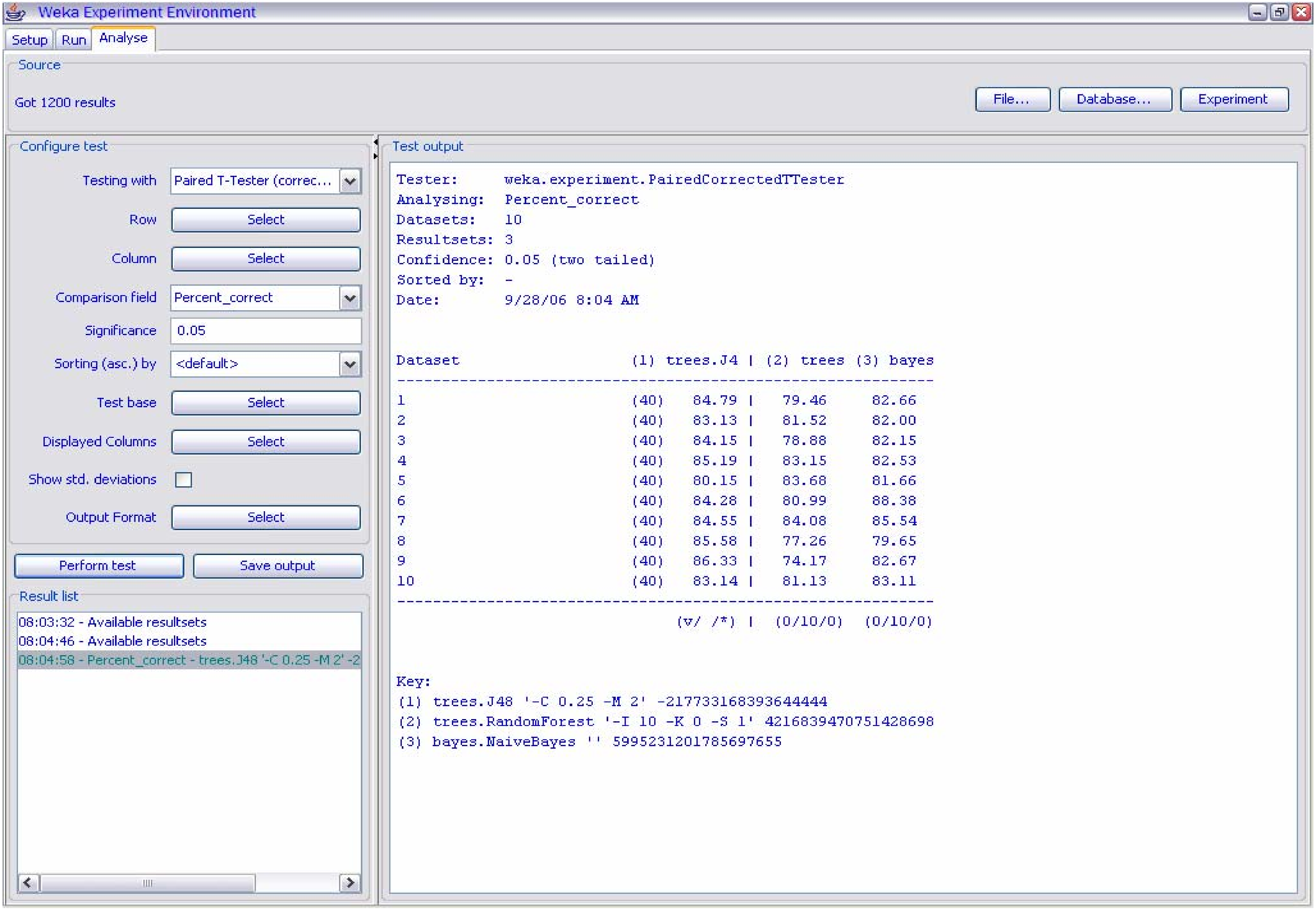
Algorithms

J48 -C 0.25 -M 2
RandomForest -I 10 -K 0 -S 1
NaiveBayes



```
08:01:18: Started
08:02:57: Finished
08:02:57: There were 0 errors
```

Not running



Got 1200 results

Testing with: Paired T-Tester (correc...

Row: Select

Column: Select

Comparison field: Percent_correct

Significance: 0.05

Sorting (asc.) by: <default>

Test base: Select

Displayed Columns: Select

Show std. deviations: □

Output Format: Select

```
08:03:32 - Available resultsets
08:04:46 - Available resultsets
08:04:58 - Percent_correct - trees.J48 '-C 0.25 -M 2' -2
```

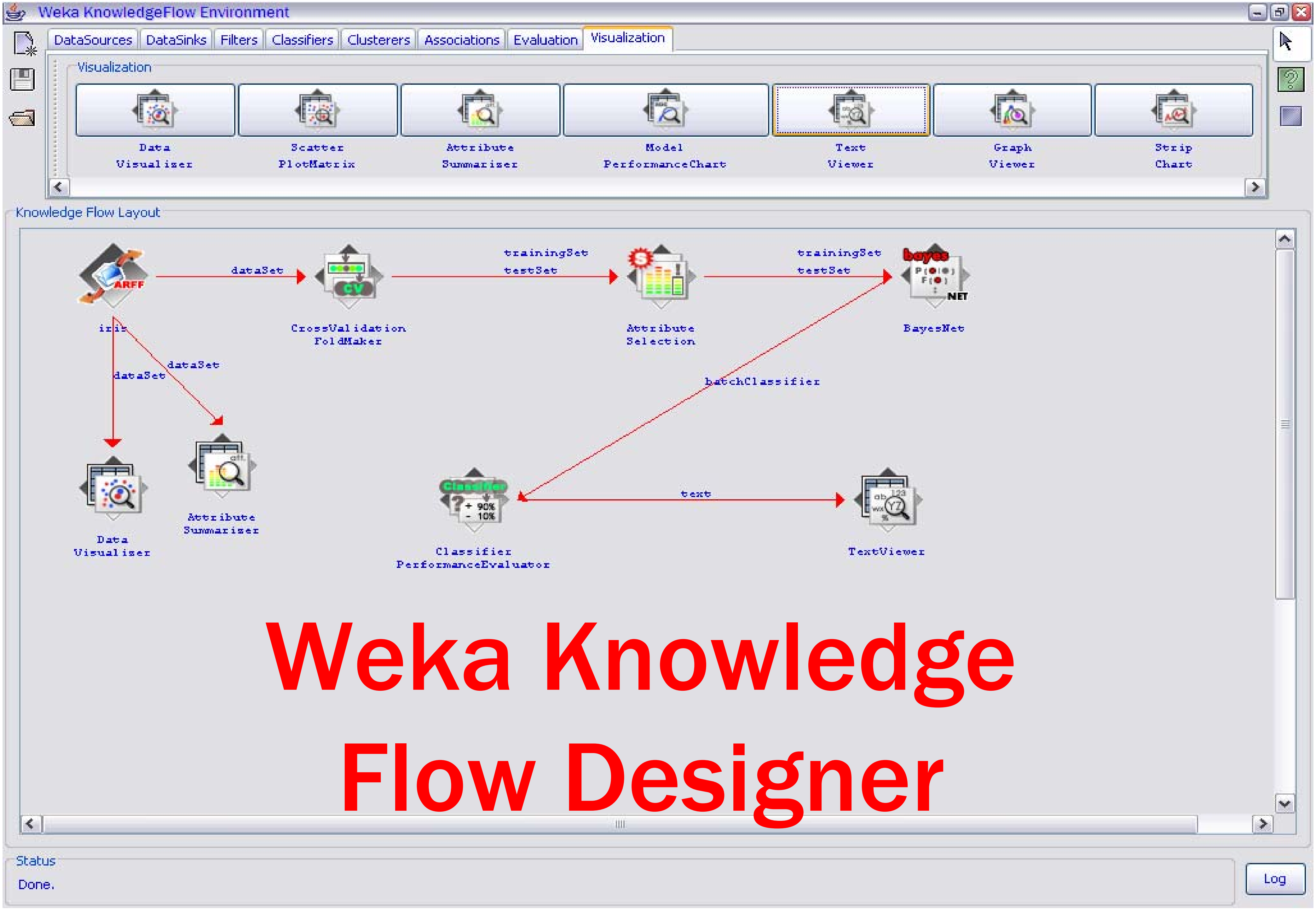
```
Tester:     weka.experiment.PairedCorrectedTTester
Analysing:  Percent_correct
Datasets:   10
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       9/28/06 8:04 AM


Dataset                   (1) trees.J4 | (2) trees (3) bayes
------------------------------------------------------------
1                         (40)   84.79 |   79.46     82.66
2                         (40)   83.13 |   81.52     82.00
3                         (40)   84.15 |   78.88     82.15
4                         (40)   85.19 |   83.15     82.53
5                         (40)   80.15 |   83.68     81.66
6                         (40)   84.28 |   80.99     88.38
7                         (40)   84.55 |   84.08     85.54
8                         (40)   85.58 |   77.26     79.65
9                         (40)   86.33 |   74.17     82.67
10                        (40)   83.14 |   81.13     83.11
------------------------------------------------------------
                               (v/ /*) |  (0/10/0)  (0/10/0)


Key:
(1) trees.J48 '-C 0.25 -M 2' -217733168393644444
(2) trees.RandomForest '-I 10 -K 0 -S 1' 4216839470751428698
(3) bayes.NaiveBayes '' 5995231201785697655
```

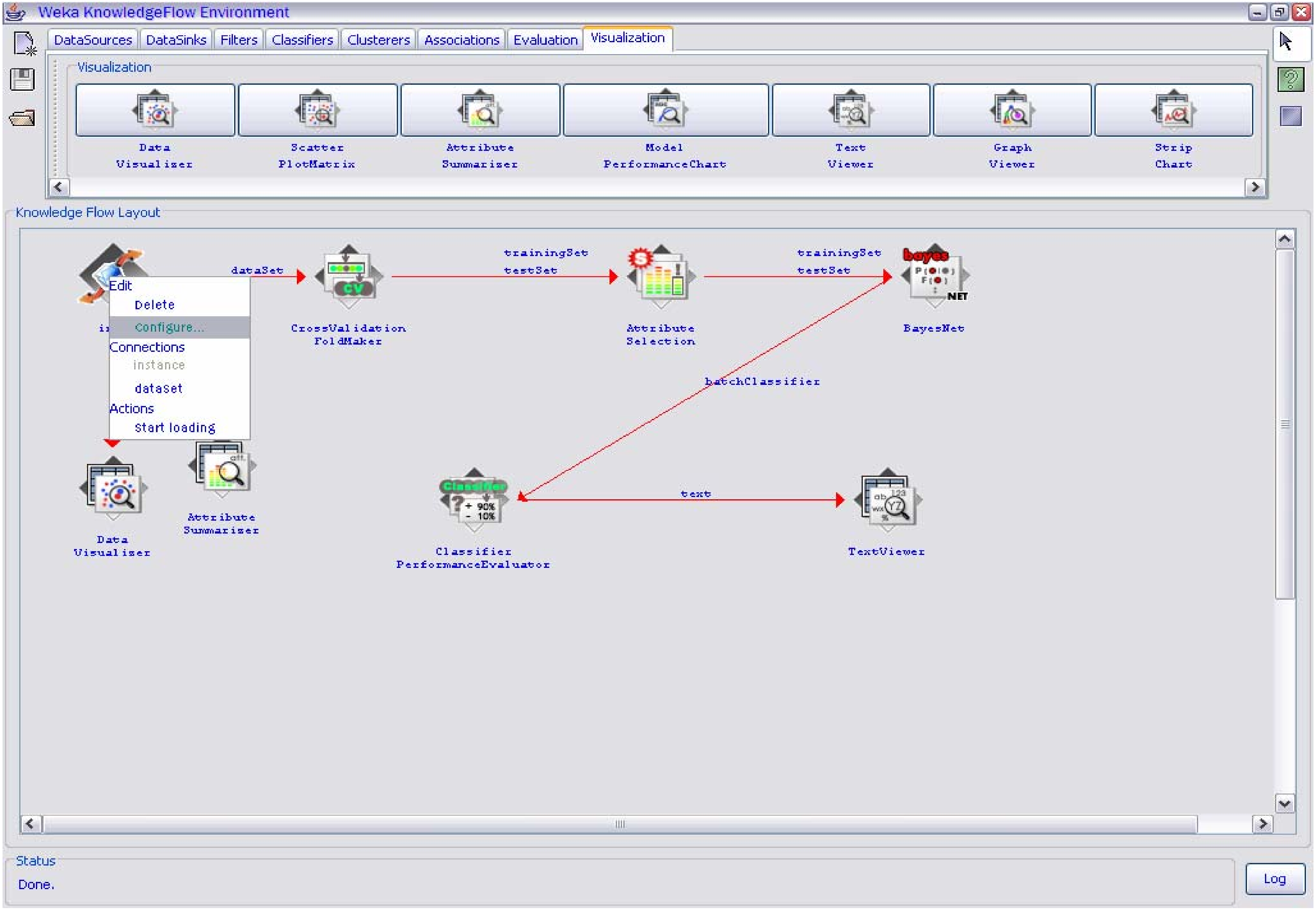
# The Knowledge Flow GUI

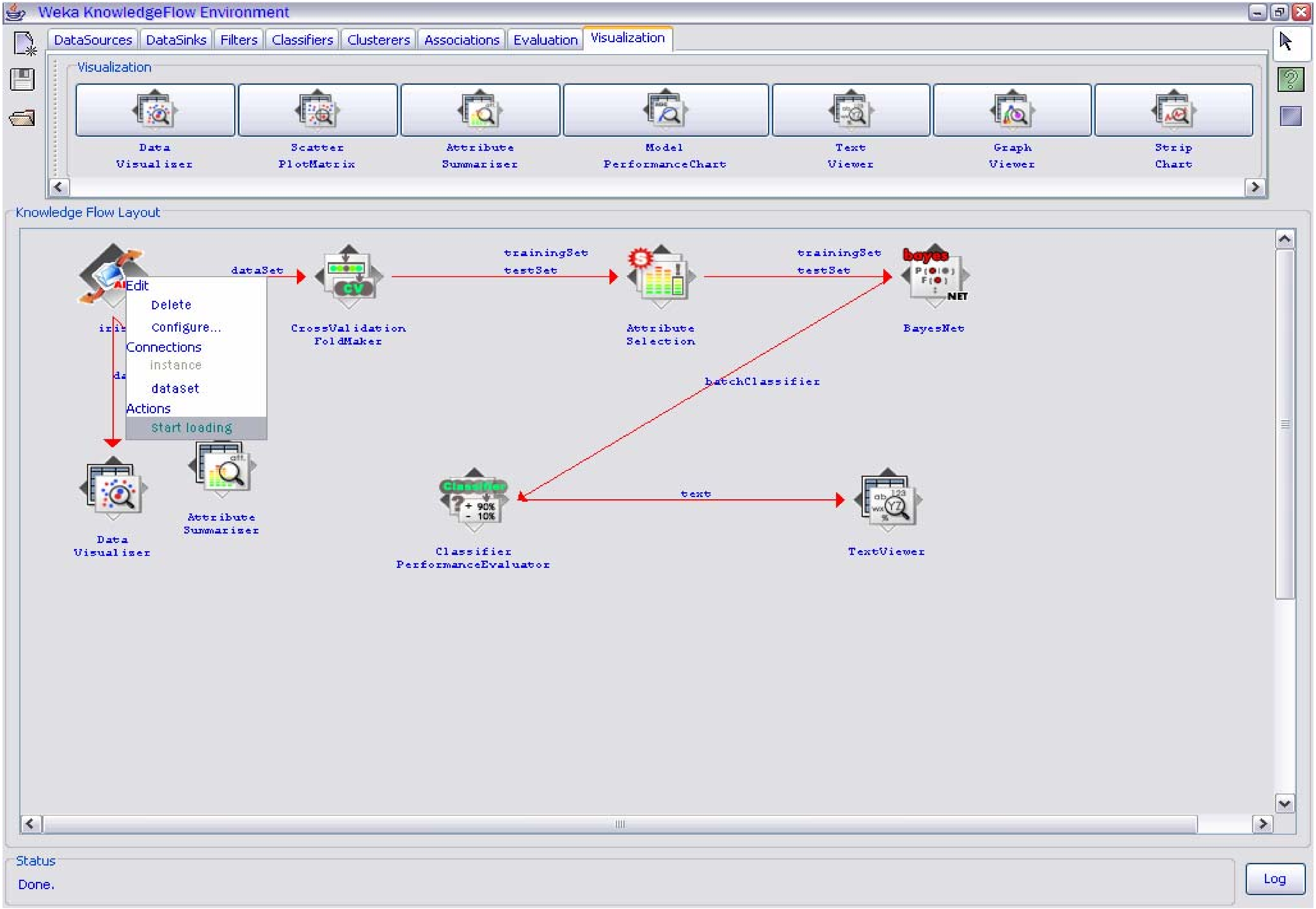
- New graphical user interface for WEKA
- Java-Beans-based interface for setting up and running machine learning experiments
- Data sources, classifiers, etc. are beans and can be connected graphically
- Data “flows” through components: e.g.,

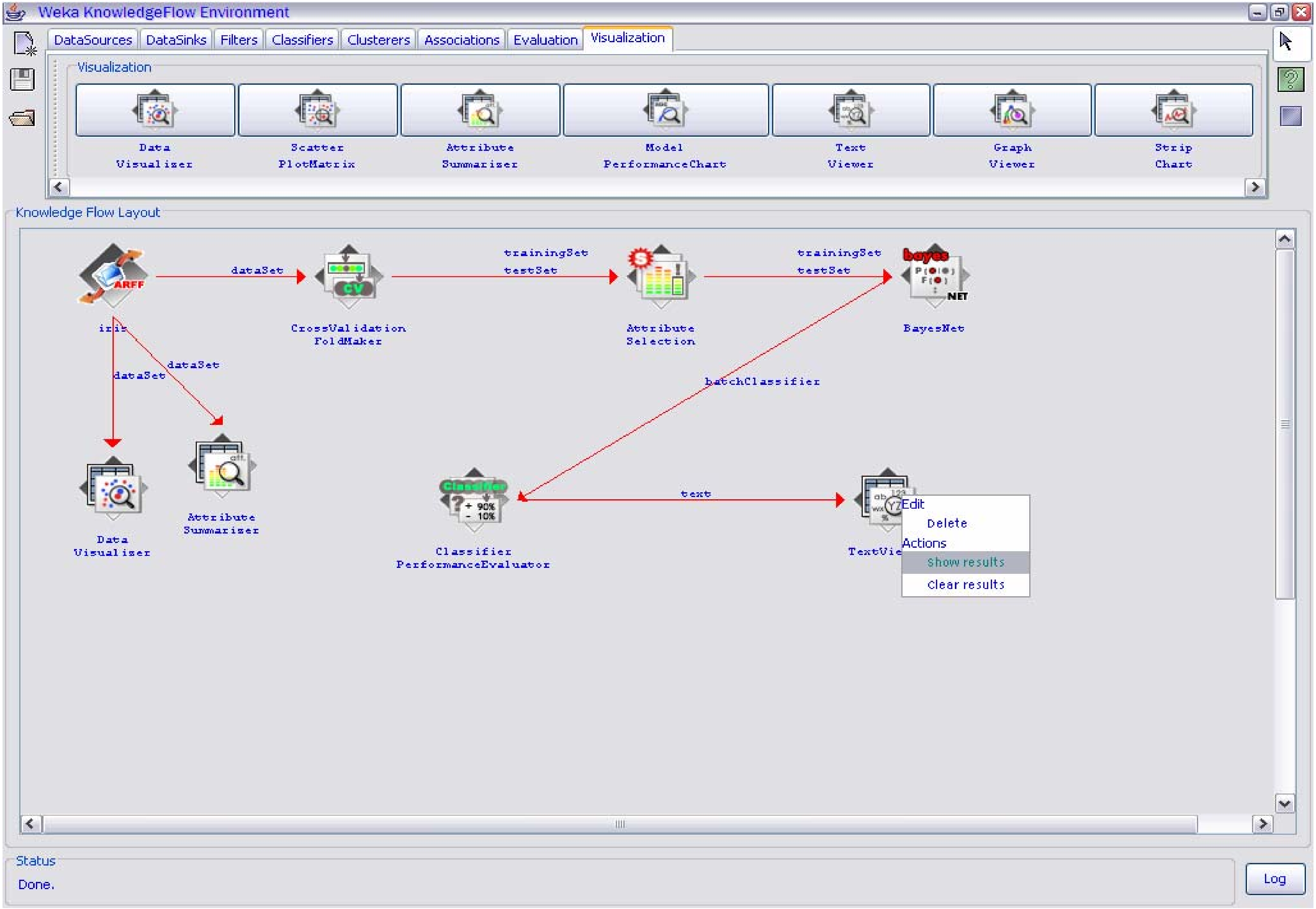
  “data source” -> “filter” -> “classifier” -> “evaluator”
- Layouts can be saved and loaded again later

# Weka Knowledge Flow Designer

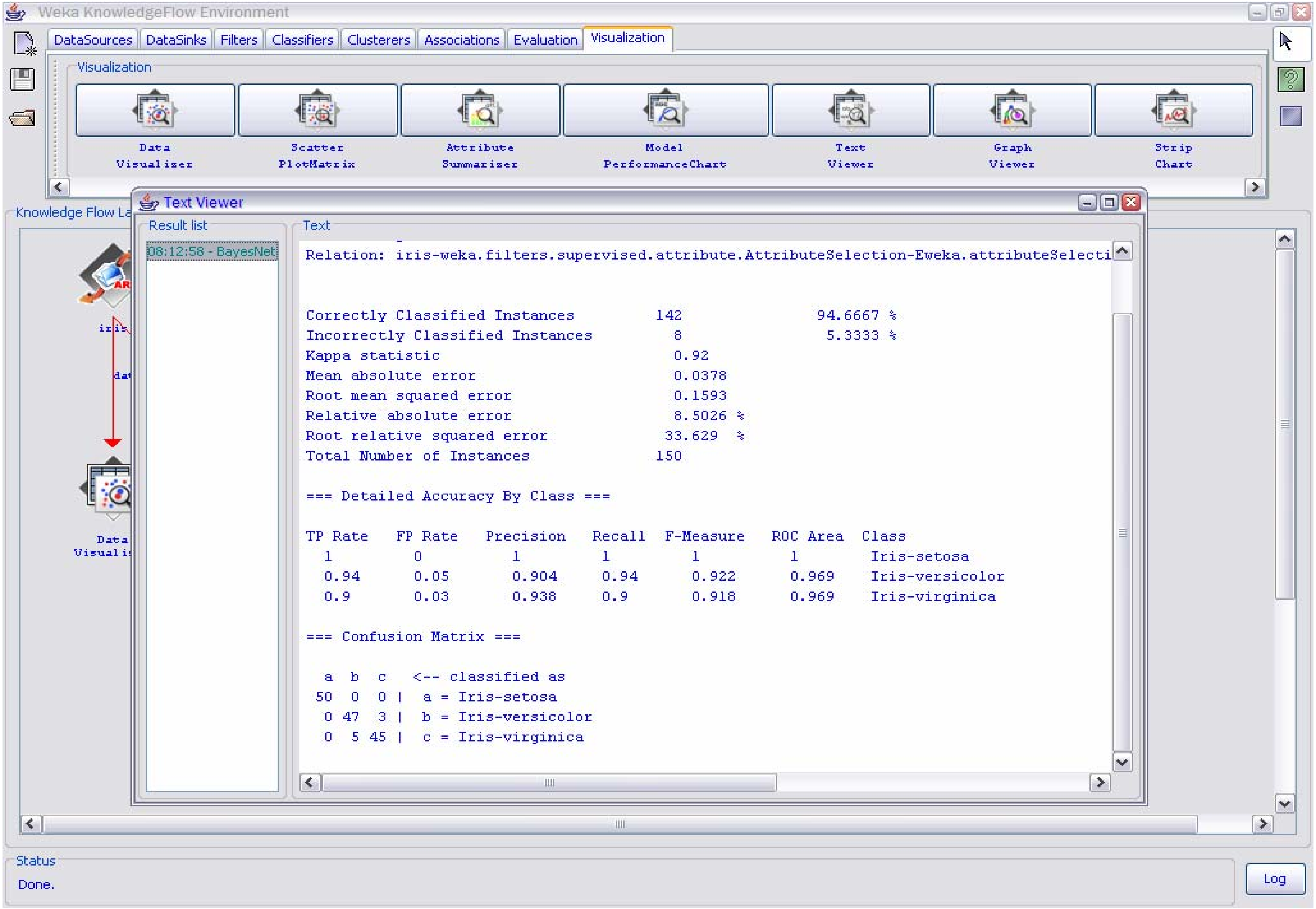



Weka KnowledgeFlow Environment
DataSources
DataSinks
Filters
Classifiers
Clusterers
Associations
Evaluation
Visualization
Visualization
Data Visualizer
Scatter PlotMatrix
Attribute Summarizer
Model PerformanceChart
Text Viewer
Graph Viewer
Strip Chart
Knowledge Flow Layout
Edit
Delete
Configure...
Connections
instance
dataset
Actions
Start loading
dataSet
CrossValidation FoldMaker
trainingSet
testSet
Attribute Selection
trainingSet
testSet
BayesNet
batchClassifier
Data Visualizer
Attribute Summarizer
Classifier PerformanceEvaluator
text
TextViewer
Status
Done.
Log


Weka KnowledgeFlow Environment
DataSources
DataSinks
Filters
Classifiers
Clusterers
Associations
Evaluation
Visualization
Visualization
Data Visualizer
Scatter PlotMatrix
Attribute Summarizer
Model PerformanceChart
Text Viewer
Graph Viewer
Strip Chart
Knowledge Flow Layout
dataSet
trainingSet
testSet
trainingSet
testSet
CrossValidation FoldMaker
Attribute Selection
BayesNet
batchClassifier
iris
Edit
Delete
Configure...
Connections
instance
dataSet
Actions
Start loading
Data Visualizer
Attribute Summarizer
Classifier PerformanceEvaluator
text
TextViewer
Status
Done.
Log


Weka KnowledgeFlow Environment
DataSources
DataSinks
Filters
Classifiers
Clusterers
Associations
Evaluation
Visualization
Visualization
Data Visualizer
Scatter PlotMatrix
Attribute Summariser
Model PerformanceChart
Text Viewer
Graph Viewer
Strip Chart
Knowledge Flow Layout
iris
dataSet
CrossValidation FoldMaker
trainingSet
testSet
Attribute Selection
trainingSet
testSet
BayesNet
dataSet
dataSet
batchClassifier
Data Visualizer
Attribute Summariser
Classifier PerformanceEvaluator
text
TextViewer
Edit
Delete
Actions
Show results
Clear results
Status
Log

Done.



```
Relation: iris-weka.filters.supervised.attribute.AttributeSelection-Eweka.attributeSelecti


Correctly Classified Instances         142               94.6667 %
Incorrectly Classified Instances         8                5.3333 %
Kappa statistic                          0.92
Mean absolute error                      0.0378
Root mean squared error                  0.1593
Relative absolute error                  8.5026 %
Root relative squared error             33.629  %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
  1         0          1         1         1          1        Iris-setosa
  0.94      0.05       0.904     0.94      0.922      0.969    Iris-versicolor
  0.9       0.03       0.938     0.9       0.918      0.969    Iris-virginica

=== Confusion Matrix ===

  a  b  c   <-- classified as
 50  0  0 |  a = Iris-setosa
  0 47  3 |  b = Iris-versicolor
  0  5 45 |  c = Iris-virginica
```

# Finally!

WEKA is available at

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

Also has a list of projects based on WEKA

WEKA contributors:

Abdelaziz Mahoui, Alexander K. Seewald, Ashraf M. Kibriya, Bernhard Pfahringer , Brent Martin, Peter Flach, Eibe Frank ,Gabi Schmidberger ,Ian H. Witten , J. Lindgren, Janice Boughton, Jason Wells, Len Trigg, Lucio de Souza Coelho, Malcolm Ware, Mark Hall ,Remco Bouckaert , Richard Kirkby, Shane Butler, Shane Legg, Stuart Inglis, Sylvain Roy, Tony Voyle, Xin Xu, Yong Wang, Zhihai Wang

# Part III: evaluating machine learning algorithms using ROC and Cost Curves

# Introduction to ROC Curves (Drummond et. al.)

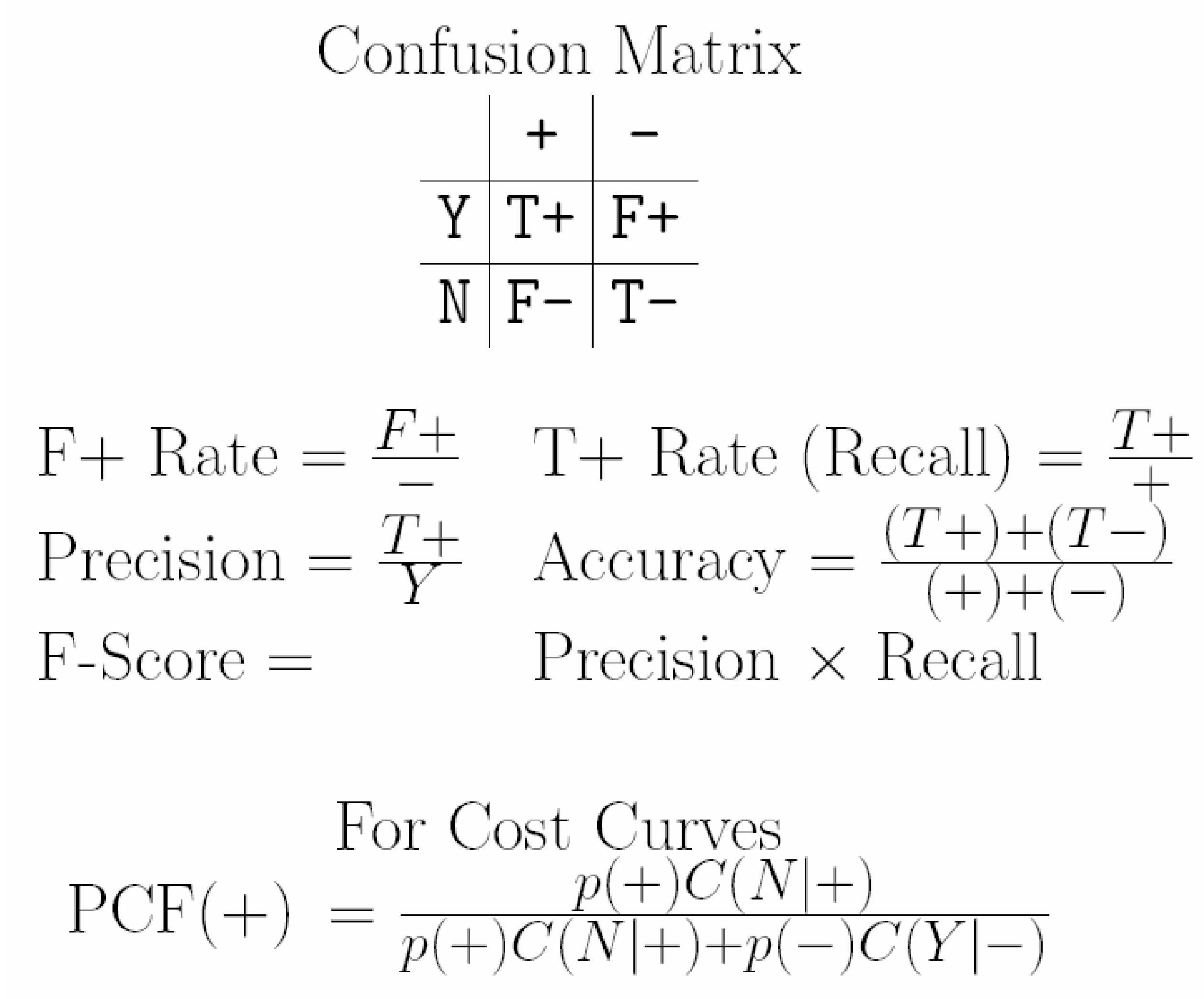
- The focus is on visualization of classifier's performance
- ROC curves show the tradeoff between false positive
- and true positive rates
- We want to know when and by how much a classifier outperforms another
- The analysis is restricted to a two class classifier

# Introduction to Cost Curves (Drummond et al.)

- The focus is on visualization of classifier's performance
- ROC curves show the tradeoff between false positive
- and true positive rates
- We want to know when and by how much a classifier outperforms another
- The analysis is restricted to a two class classifier

# Cost Curves (Drummond et al. 2004)

- Given a specific misclassification cost and class probabilities, what is the expected cost of classification?
- For what misclassification costs and class probabilities
- does a classifier outperform the trivial classifiers?
- For what misclassification costs and class probabilities
- does a classifier outperform another?
- What is the difference in performance between two classifiers?
- What is the average performance of several independent classifiers?
- What is the 90% confidence interval for a particular classifier's performance?
- What is the significance of the difference between the performances of two classifiers?



## Confusion Matrix

|   | + | - |
|---|---|---|
| Y | T+ | F+ |
| N | F- | T- |

$$\text{F+ Rate} = \frac{F+}{-} \qquad \text{T+ Rate (Recall)} = \frac{T+}{+}$$

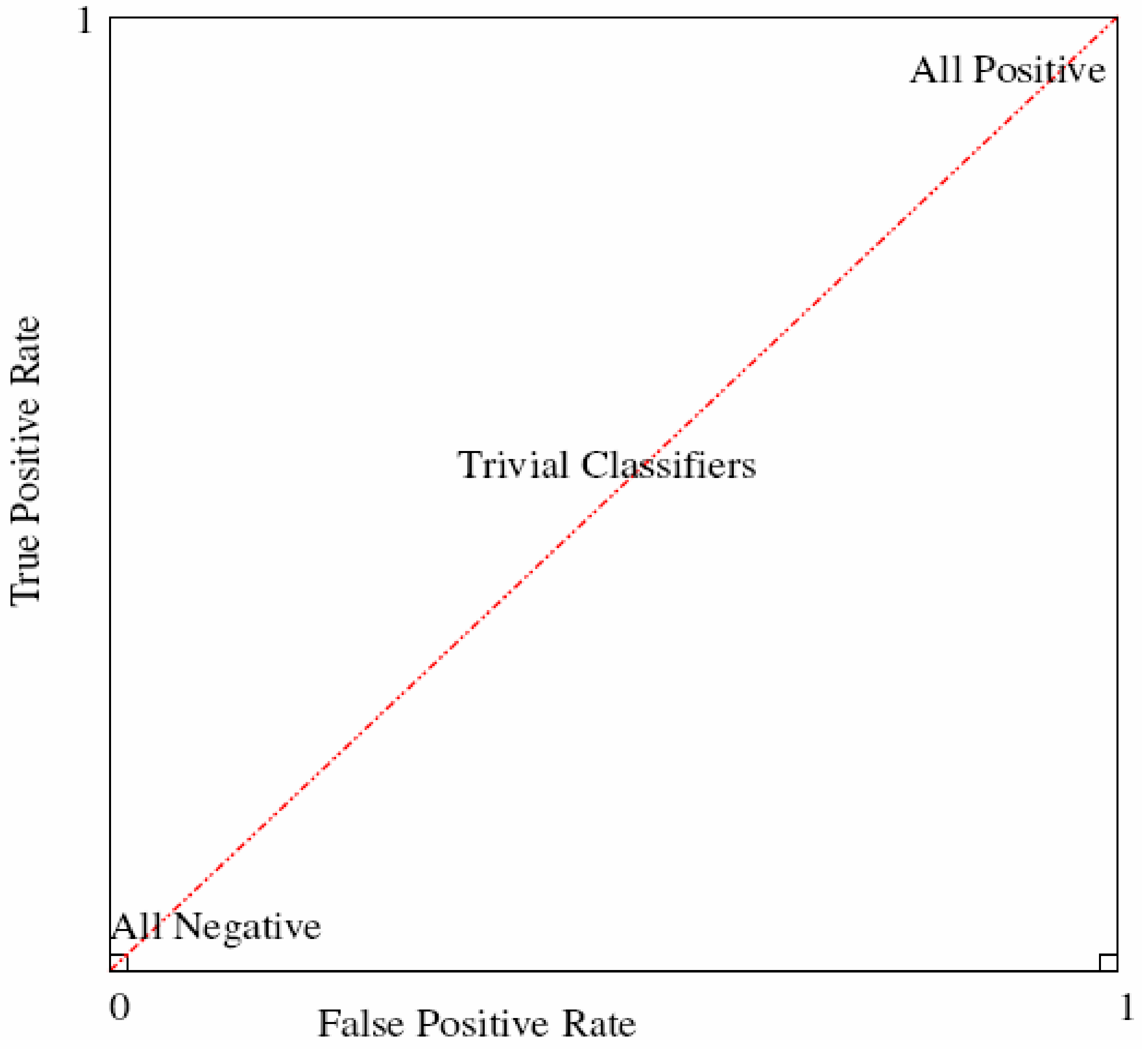
$$\text{Precision} = \frac{T+}{Y} \qquad \text{Accuracy} = \frac{(T+)+(T-)}{(+)+(-)}$$

$$\text{F-Score} = \qquad \text{Precision} \times \text{Recall}$$

## For Cost Curves

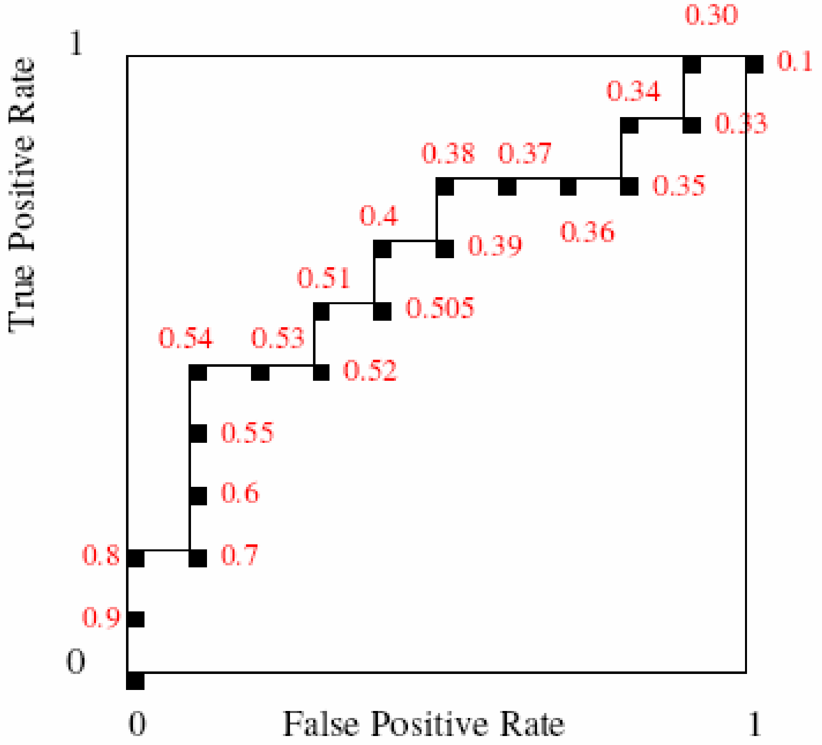
$$\text{PCF}(+) = \frac{p(+)C(N|+)}{p(+)C(N|+)+p(-)C(Y|-)}$$

# ROC Space



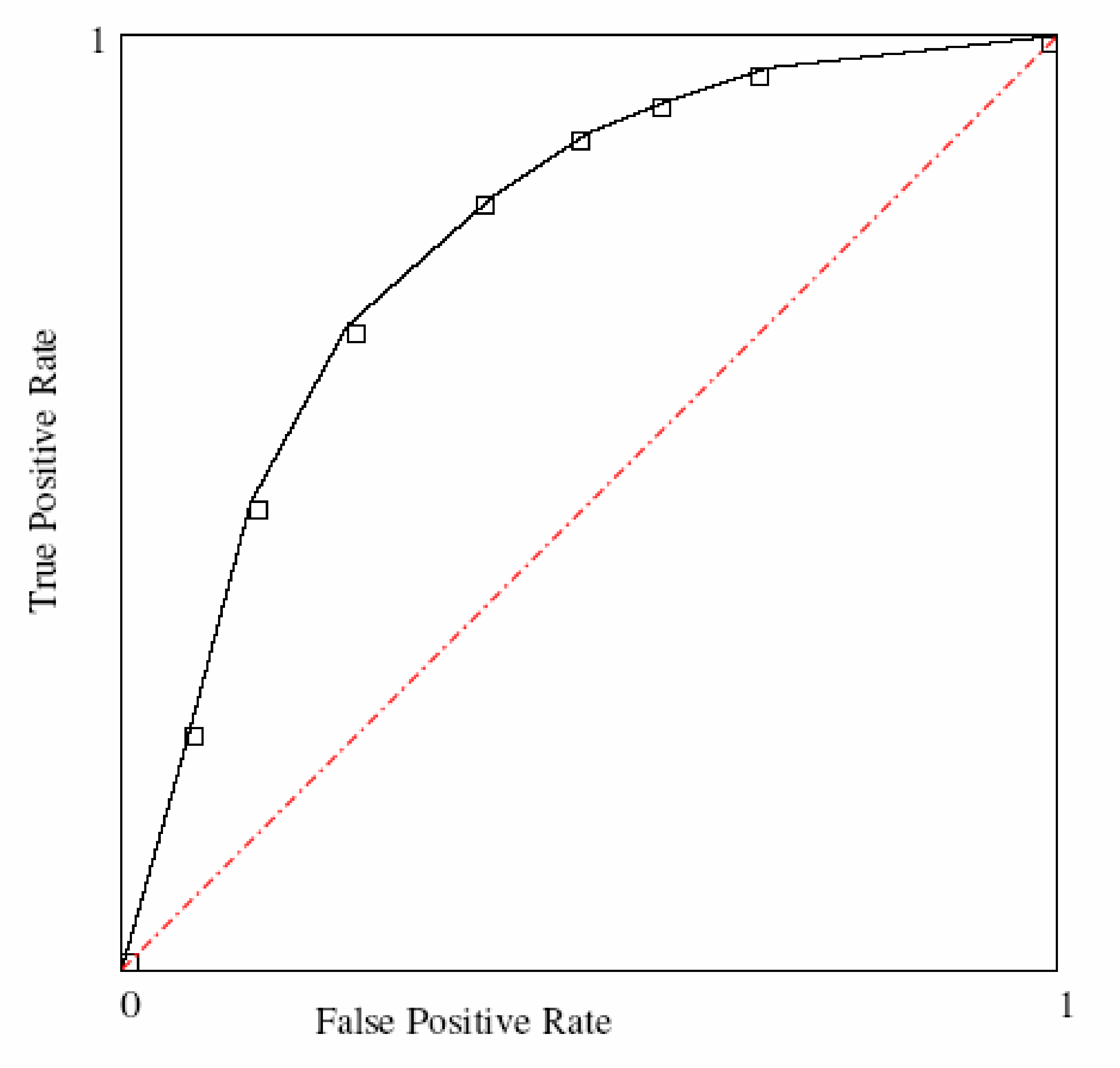
1
All Positive
Trivial Classifiers
True Positive Rate
All Negative
0
False Positive Rate
1

# Generating ROC curves

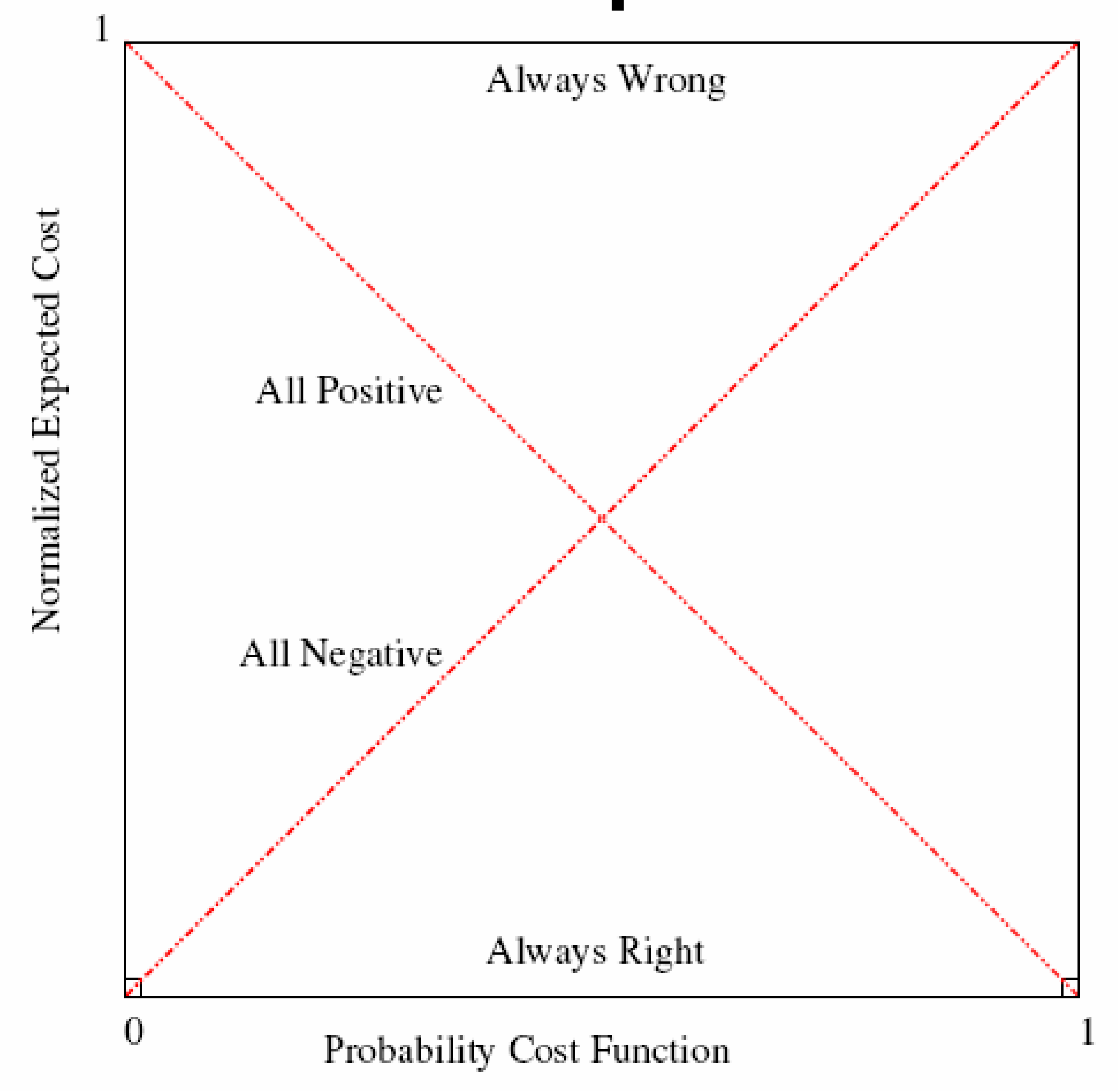


| # | Class | Score | # | Class | Score |
|---|---|---|---|---|---|
| 1 | + | 0.9 | 11 | + | 0.4 |
| 2 | + | 0.8 | 12 | - | 0.39 |
| 3 | - | 0.7 | 13 | + | 0.38 |
| 4 | + | 0.6 | 14 | - | 0.37 |
| 5 | + | 0.55 | 15 | - | 0.36 |
| 6 | + | 0.54 | 16 | - | 0.35 |
| 7 | - | 0.53 | 17 | + | 0.34 |
| 8 | - | 0.52 | 18 | - | 0.33 |
| 9 | + | 0.51 | 19 | + | 0.30 |
| 10 | - | 0.505 | 20 | - | 0.1 |

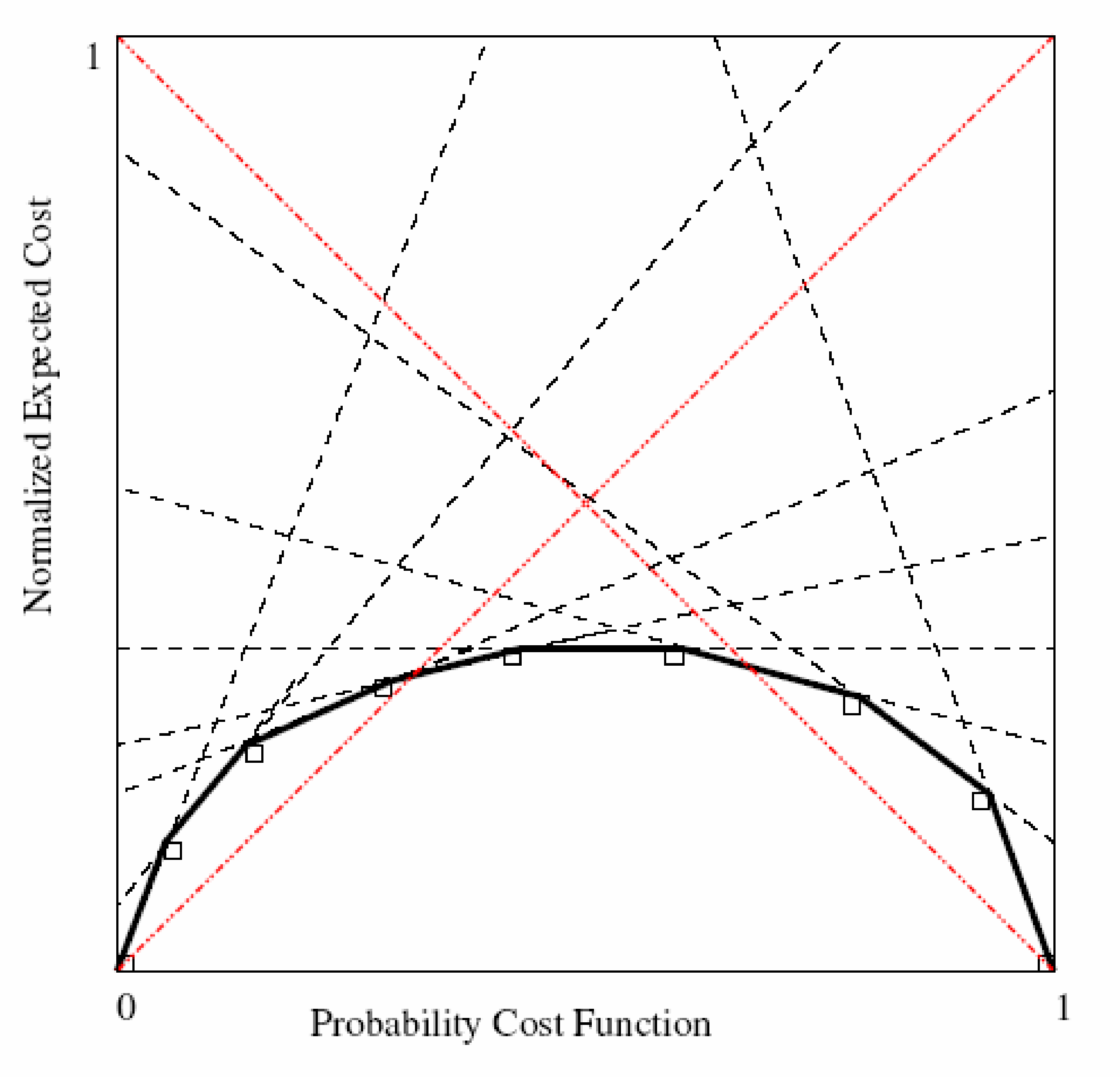
# ROC curves



# Cost Space


1
Always Wrong
All Positive
All Negative
Always Right
0
1
Probability Cost Function
Normalized Expected Cost

# Cost Curves


Probability Cost Function
Normalized Expected Cost