CentraleSupélec

DATA STREAM MANAGEMENT SYSTEM
SOCIAL NETWORK ANALYSIS AND MINING

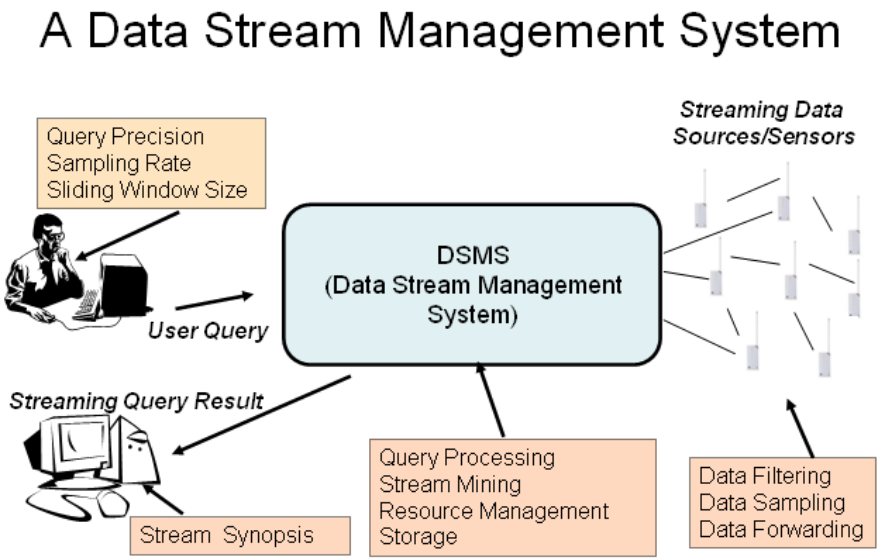
MARIO CATALDI

WWW.AI.UNIV-PARIS8.FR/~CATALDI/

M.CATALDI@IUT.UNIV-PARIS8.FR

A data stream management system (**DSMS**) is a computer software system to manage continuous data streams.

It is similar to a database management system (DBMS), which is, however, designed for static data in conventional **databases.**



TO RESUME

TO RESUME

A Data Stream Management System needs:

1. A data stream extraction system to handle, ***in real time***, a flow of information
2. A data store system to properly store the relevant information into a limited amount of space
3. A data formalization model for studying the information and the structure of the extrated data.

DATA STREAM MANAGEMENT SYSTEM

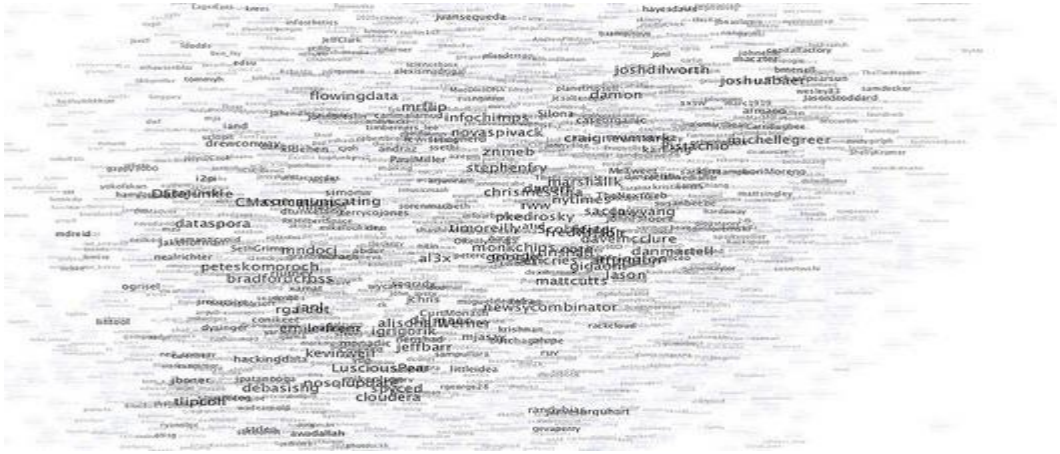
One of the biggest challenges for a DSMS is to handle **potentially infinite** data streams using a fixed amount of memory and no random access to the data.

Two main classes of approaches:

1. *compression techniques that try to summarize the information*
2. *window techniques that try to portion the data into (finite) parts*.

PROBLEM:

too much information.



SOLUTIONS:

- Threads (parallel computing)
- Volatile solutions (not everything needs to be threated and/or stored)
- Target the extraction before extracting the data

TO RESUME: PROBLEMS

Apache Hadoop

- Hadoop Distributed File System (HDFS)
- MapReduce

Apache Spark

These are changing rapidly – active area of use and growth. These are *big hot areas* today

- "Silicon Valley investors have poured $2 billion into companies based on the data-collection software known as Hadoop."– Wall Street Journal, June 15, 2015
- IBM to invest few hundred million dollars a year in Spark
- Not including investments by Facebook, Google, Yahoo!, Baidu, and others

APACHE HADOOP/SPARK

APACHE HADOOP: PURPOSE

"Framework that allows distributed processing of large data sets across clusters of computers…sing simple programming models.

It is designed to scale up from single servers to thousands of machines, each offering local computation and storage.

Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures."

Fast, expressive cluster computing system compatible with Apache Hadoop

- Works with any Hadoop-supported storage system (HDFS, S3, Avro, …)

Improves efficiency through:

- In-memory computing primitives
- General computation graphs

→ Up to 100× faster

Improves usability through:

- Rich APIs in Java, Scala, Python
- Interactive shell

→ Often 2-10× less code

PARALLEL COMPUTING: HADOOP/SPARK

APACHE SPARK

Processing engine; instead of just “map” and “reduce”, defines a large set of *operations* (transformations & actions)

- Operations can be arbitrarily combined in any order

Open source software

Supports Java, Scala and Python

Key construct: Resilient Distributed Dataset (RDD)

RESILIENT DISTRIBUTED DATASET (RDD)

RDDs represent data or transformations on data

RDDs can be created from Hadoop InputFormats (such as HDFS files), “parallelize()” datasets, or by transforming other RDDs (you can stack RDDs)

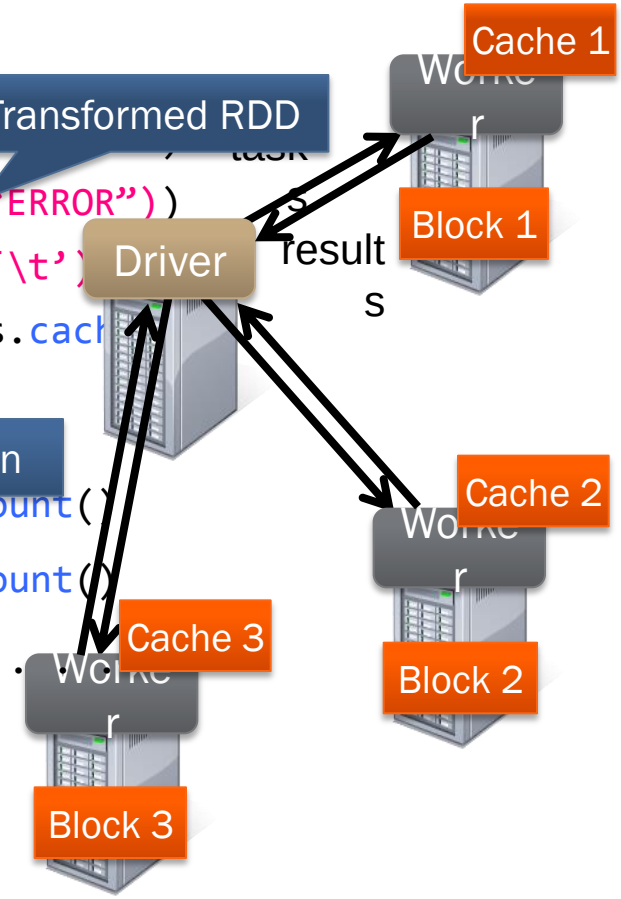
Actions can be applied to RDDs; actions force calculations and return values

Lazy evaluation: Nothing computed until an action requires it

RDDs are best suited for applications that apply the same operation to all elements of a dataset

Load error messages from a log into memory, then interactively search for patterns

```
lines = spark.textFile(“hd
errors = lines.filter(lambda s: s.startswith(“ERROR”))
messages = errors.map(lambda s: s.split(‘\t’)
messages.cach

messages.filter(lambda s: “foo” in s).count()
messages.filter(lambda s: “bar” in s).count()
```



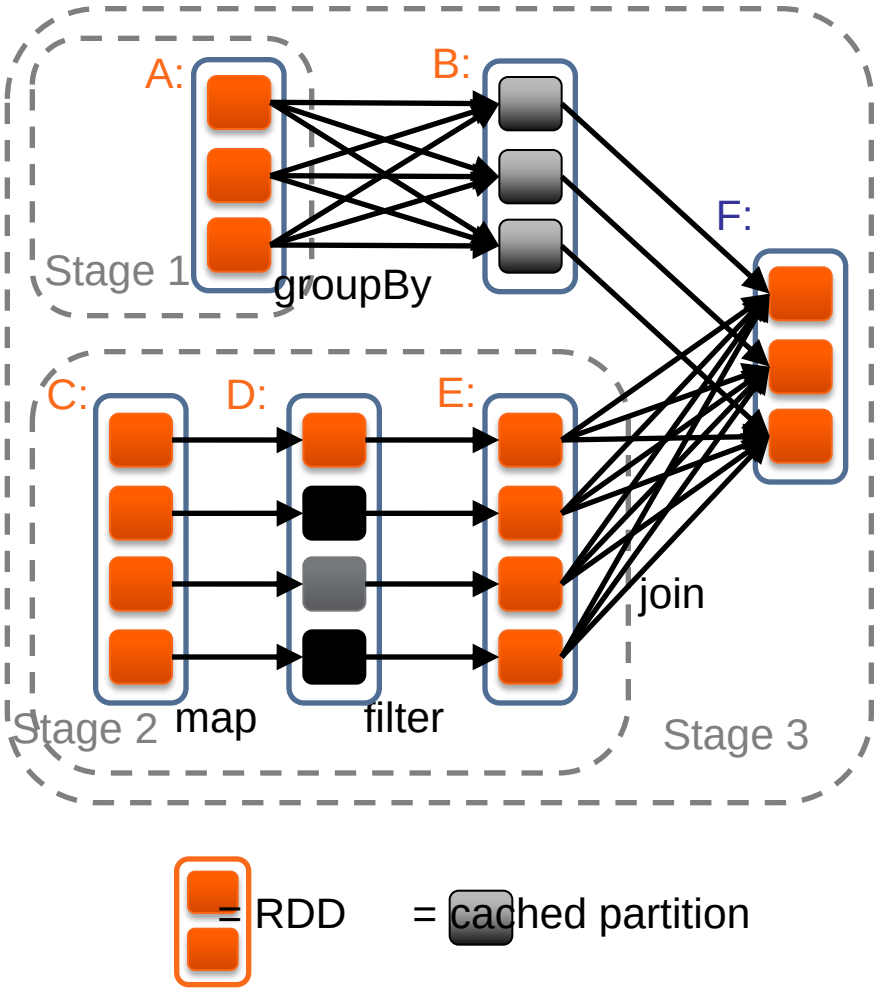
Result: scaled to 1 TB data in 5-7 sec
(vs 170 sec for on-disk data)

PARALLEL COMPUTING: SPARK

TASK SCHEDULER

Supports general task graphs

Pipelines functions where possible Cache-aware data reuse & locality Partitioning-aware to avoid shuffles



SPARK: ARCHITECTURE

WHY PAGERANK?

Good example of a more complex algorithm

- Multiple stages of map & reduce

Benefits from Spark's in-memory caching

- Multiple iterations over the same data

SPARK: EXAMPLE PAGE RANK

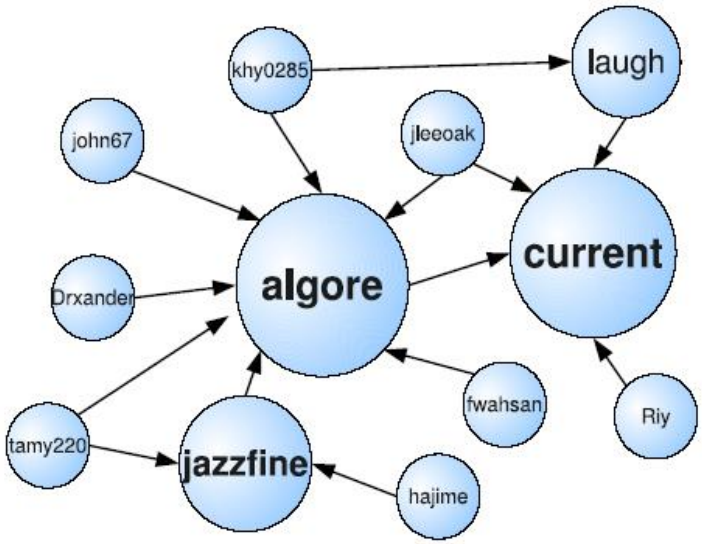
LINK ANALYSIS: PAGE RANK

In a graph, we could also take into account the existing links among the nodes to perform alternative analysis.

The page rank algorithm could help estimate the popularity of a node

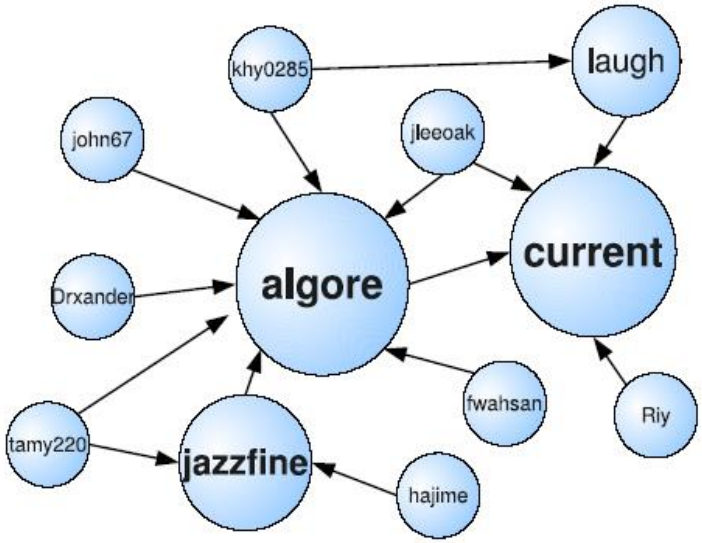
Links can bring a lot of information

Link between nodes= perticene relation



LINK ANALYSIS: PAGE RANK

We define an author-based graph ***G(U, F)*** where ***U*** is the set of users and ***F*** is the set of directed edges; thus, given two users *u1* and *u2*, the edge <*u1, u2*> exists only if u1 is a follower of u2 .

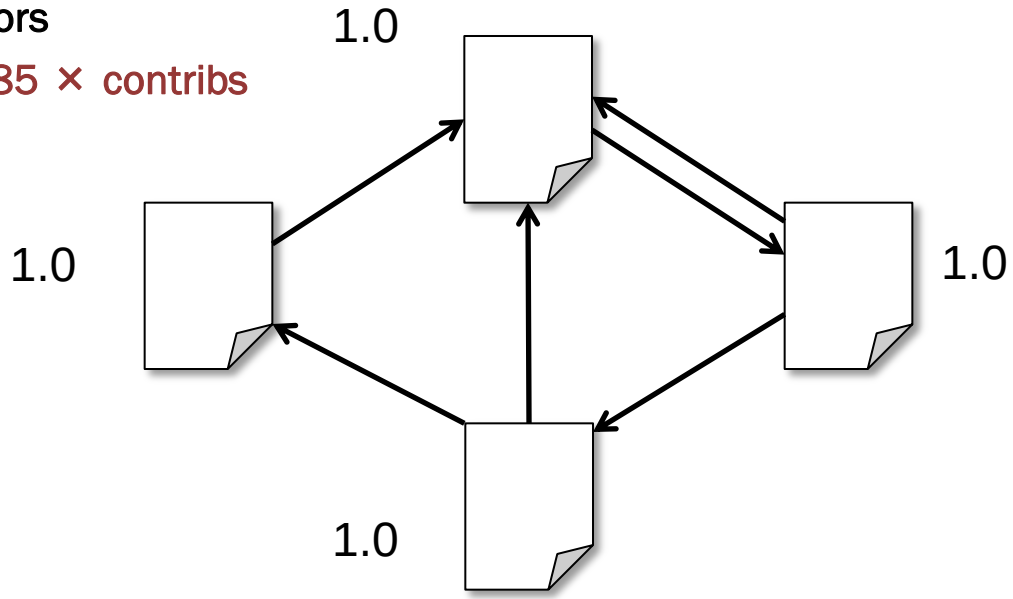


..thus, we measure the degree of importance of each user by analyzing the connectivity in G;

Page Rank on G

ALGORITHM

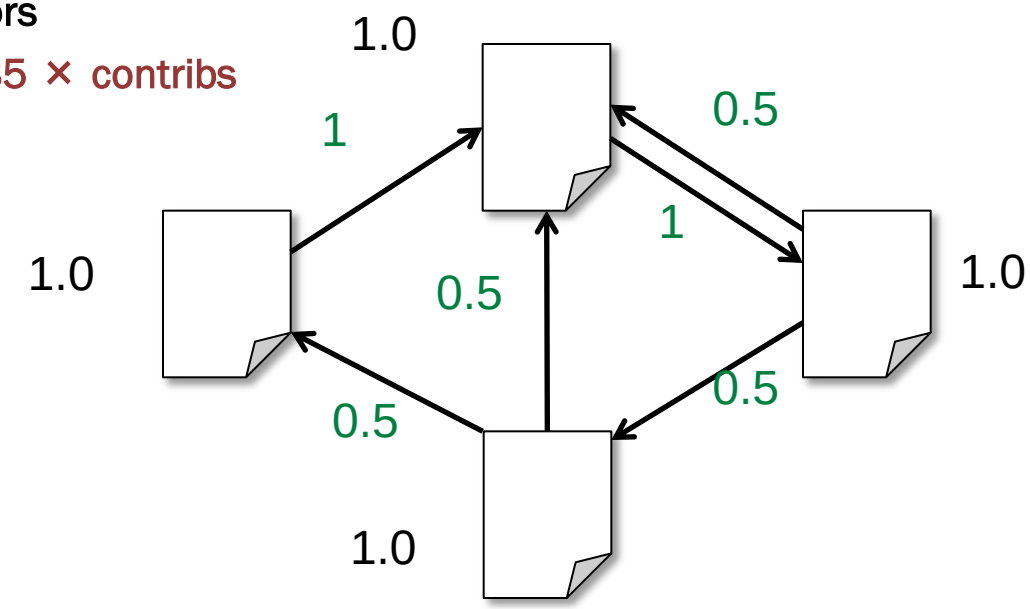
1. Start each page at a rank of 1
2. On each iteration, have page p contribute $rank_p / |neighbors_p|$ to its neighbors
3. Set each page's rank to $0.15 + 0.85 \times contribs$



LINK ANALYSIS: PAGE RANK

ALGORITHM

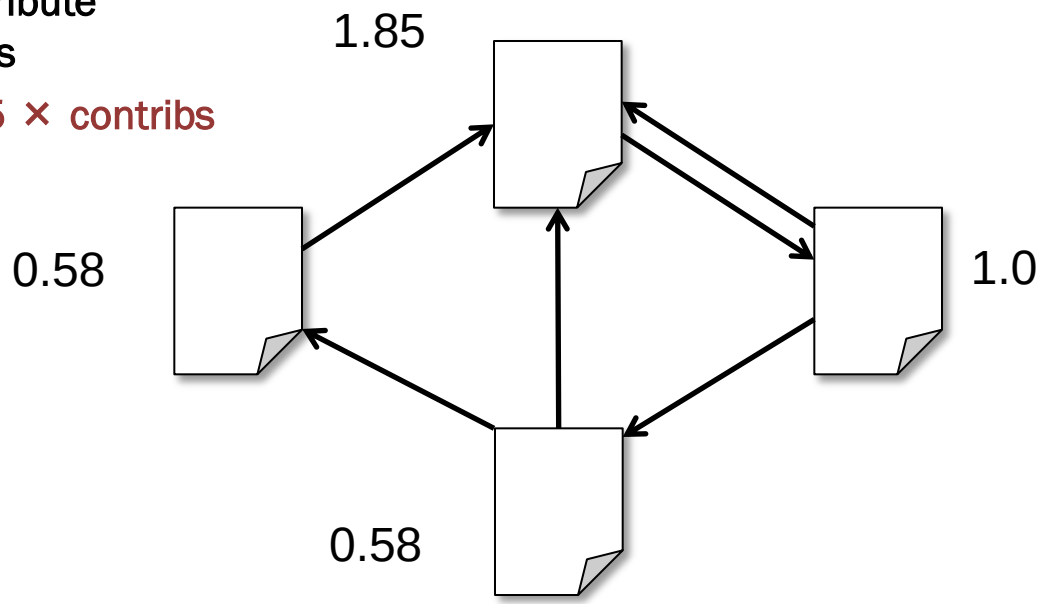
1. Start each page at a rank of 1
2. On each iteration, have page p contribute $rank_p / |neighbors_p|$ to its neighbors
3. Set each page's rank to $0.15 + 0.85 \times contribs$



LINK ANALYSIS: PAGE RANK

ALGORITHM

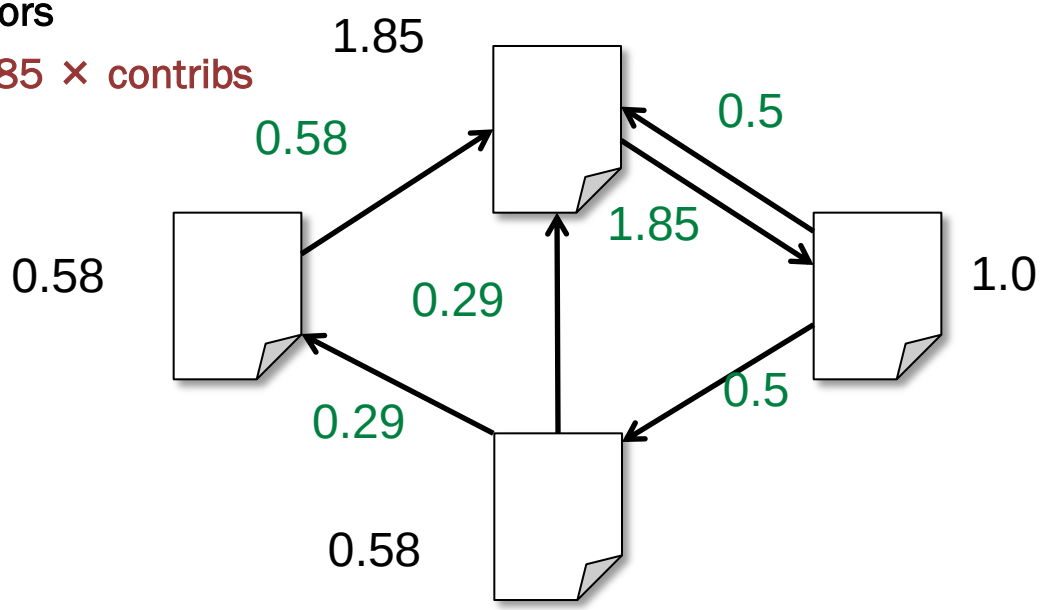
1. Start each page at a rank of 1
2. On each iteration, have page p contribute $rank_p / |neighbors_p|$ to its neighbors
3. Set each page's rank to $0.15 + 0.85 \times contribs$



LINK ANALYSIS: PAGE RANK

ALGORITHM

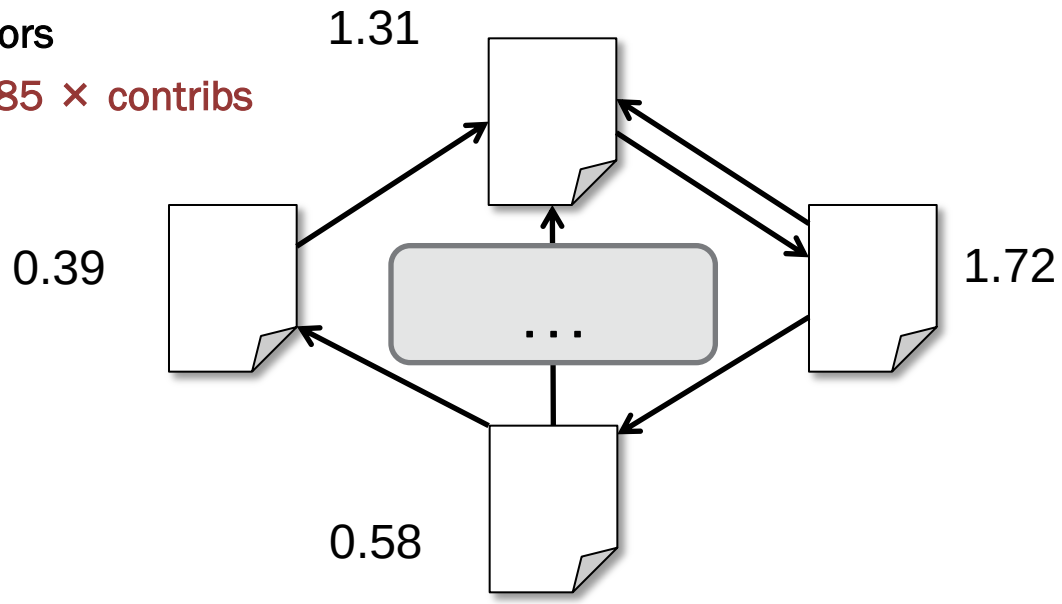
1. Start each page at a rank of 1
2. On each iteration, have page p contribute $rank_p / |neighbors_p|$ to its neighbors
3. Set each page's rank to $0.15 + 0.85 \times contribs$



LINK ANALYSIS: PAGE RANK

ALGORITHM

1. Start each page at a rank of 1
2. On each iteration, have page p contribute $rank_p / |neighbors_p|$ to its neighbors
3. Set each page's rank to $0.15 + 0.85 \times contribs$

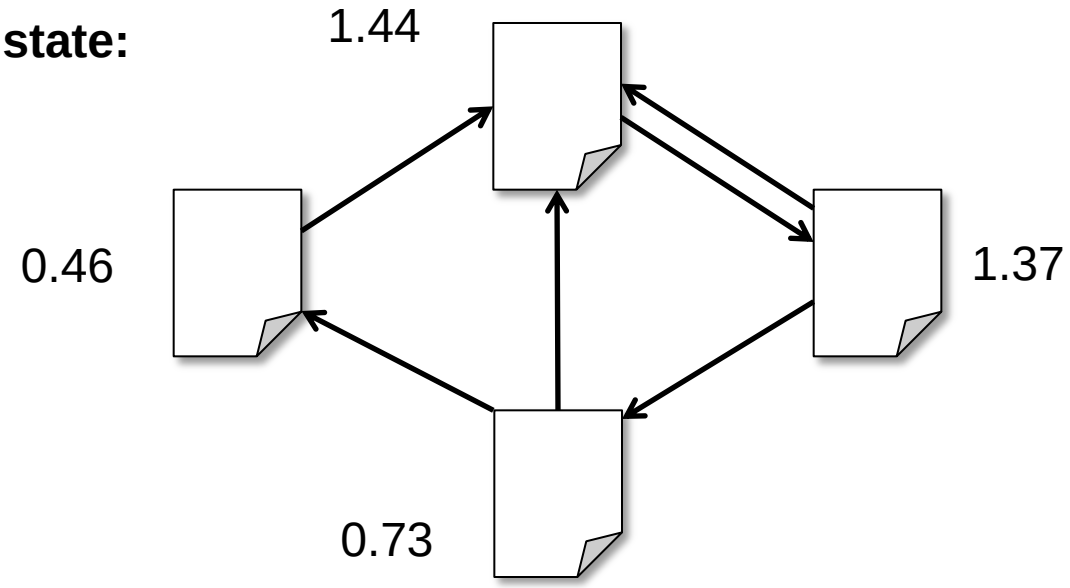


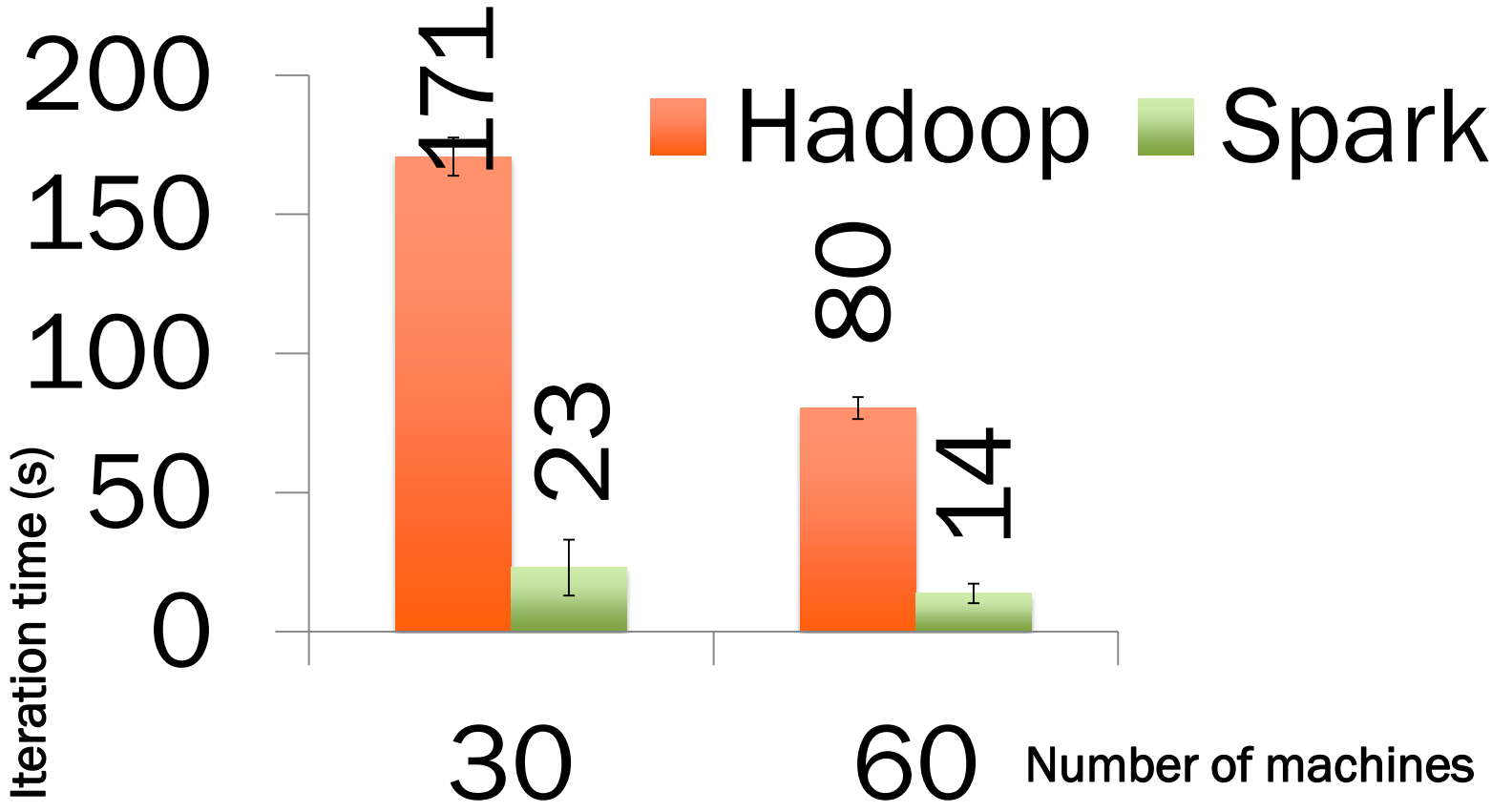
LINK ANALYSIS: PAGE RANK

LINK ANALYSIS: PAGE RANK

1. Start each page at a rank of 1
2. On each iteration, have page p contribute $rank_p / |neighbors_p|$ to its neighbors
3. Set each page's rank to $0.15 + 0.85 \times contribs$

Final state:





LINK ANALYSIS: PAGE RANK

Waikato Environment for Knowledge Analysis

- It's a data mining/machine learning tool developed by Department of Computer Science, University of Waikato, New Zealand.
- Weka is also a bird found only on the islands of New Zealand.

WHAT IS WEKA?

DOWNLOAD AND INSTALL WEKA

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

Support multiple platforms (written in java):

- Windows, Mac OS X and Linux

MAIN FEATURES

49 data preprocessing tools

76 classification/regression algorithms

8 clustering algorithms

3 algorithms for finding association rules

15 attribute/subset evaluators + 10 search algorithms for feature selection

MAIN GUI

Three graphical user interfaces

- "The Explorer" (exploratory data analysis)
- "The Experimenter" (experimental environment)
- "The KnowledgeFlow" (new process model inspired interface)

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”

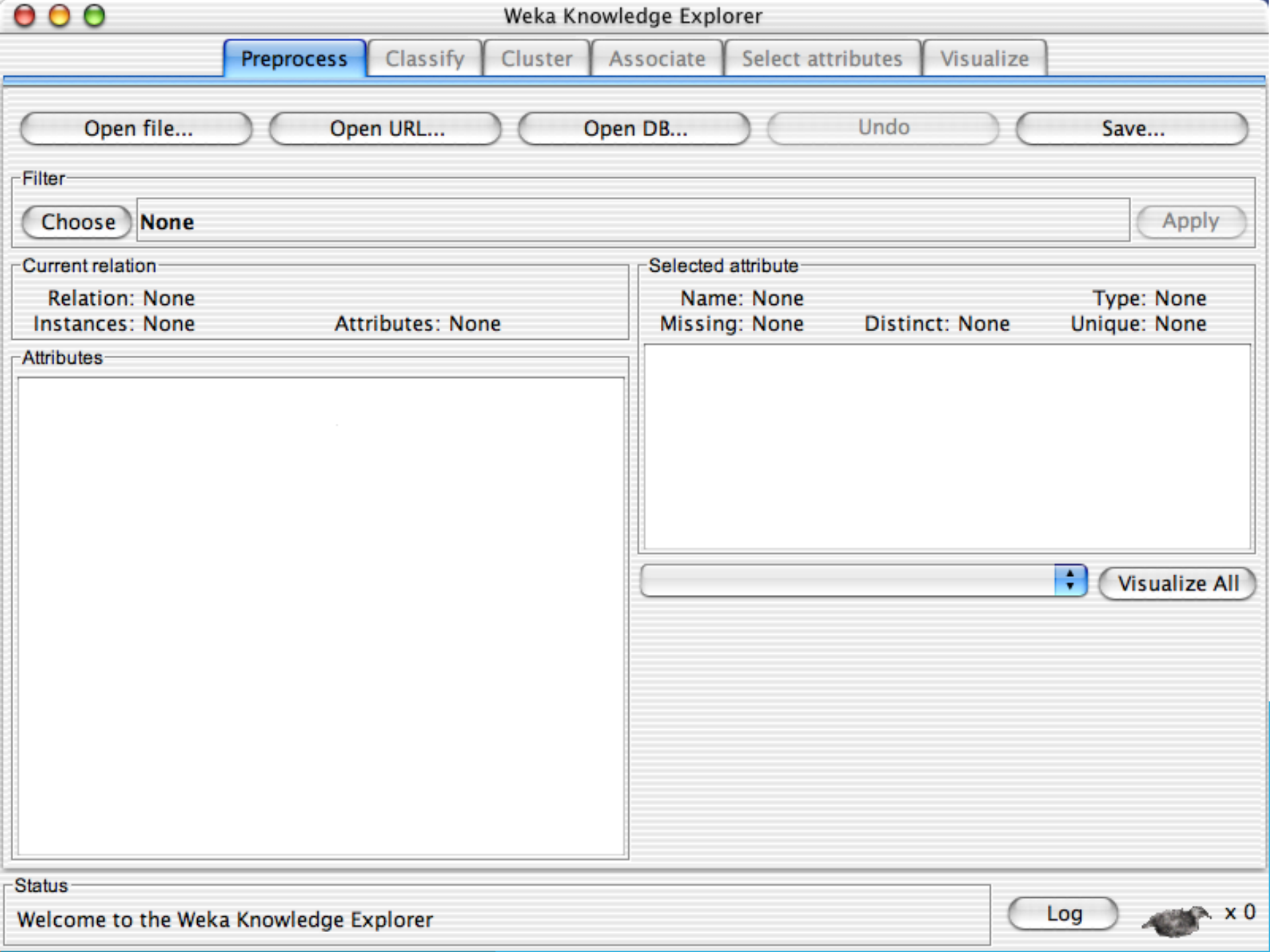
WEKA contains filters for:

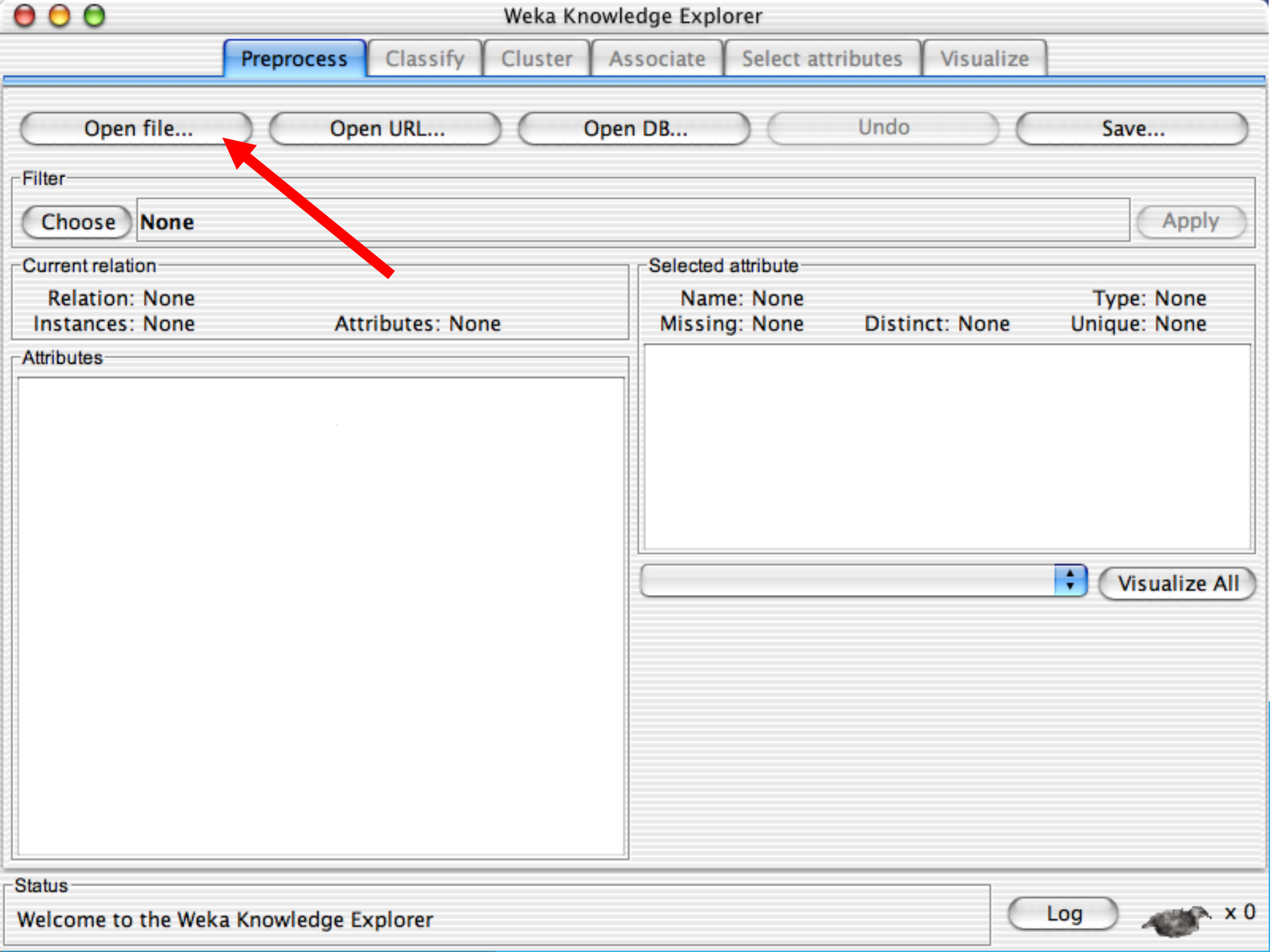
- Discretization, normalization, resampling, attribute selection, transforming and combining attributes, ...

```
@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: ARFF FILE


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
None
Apply
Current relation
Relation: None
Instances: None
Attributes: None
Selected attribute
Name: None
Type: None
Missing: None
Distinct: None
Unique: None
Attributes
Visualize All
Status
Welcome to the Weka Knowledge Explorer
Log
x 0


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
None
Apply
Current relation
Relation: None
Instances: None
Attributes: None
Selected attribute
Name: None
Type: None
Missing: None
Distinct: None
Unique: None
Attributes
Visualize All
Status
Welcome to the Weka Knowledge Explorer
Log
x 0

Weka Knowledge Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Undo | Save...

Filter

Choose **None** | Apply

Current relation

Relation: iris
Instances: 150 Attributes: 5

Attributes

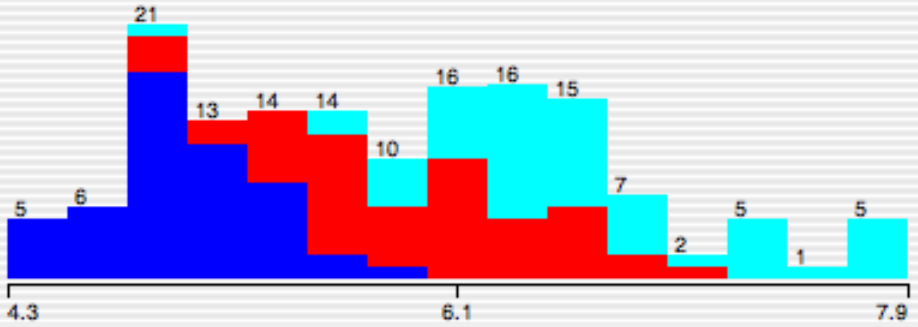
No.	Name
1	sepallength
2	sepalwidth
3	petallength
4	petalwidth
5	class

Selected attribute

Name: sepallength Type: Numeric
Missing: 0 (0%) Distinct: 35 Unique: 9 (6%)

Statistic	Value
Minimum	4.3
Maximum	7.9
Mean	5.843
StdDev	0.828

Colour: class (Nom) | Visualize All



Status

OK

Log x 0

Weka Knowledge Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Undo | Save...

Filter

Choose **None** | Apply

Current relation

Relation: iris
Instances: 150 Attributes: 5

Attributes

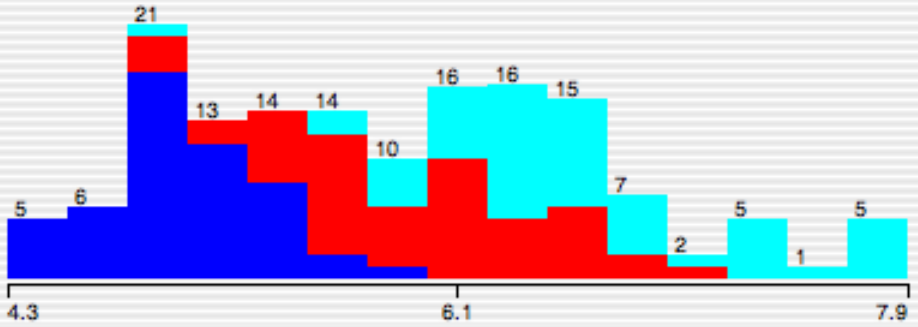
No.	Name
1	sepallength
2	sepalwidth
3	petallength
4	petalwidth
5	class

Selected attribute

Name: sepallength Type: Numeric
Missing: 0 (0%) Distinct: 35 Unique: 9 (6%)

Statistic	Value
Minimum	4.3
Maximum	7.9
Mean	5.843
StdDev	0.828

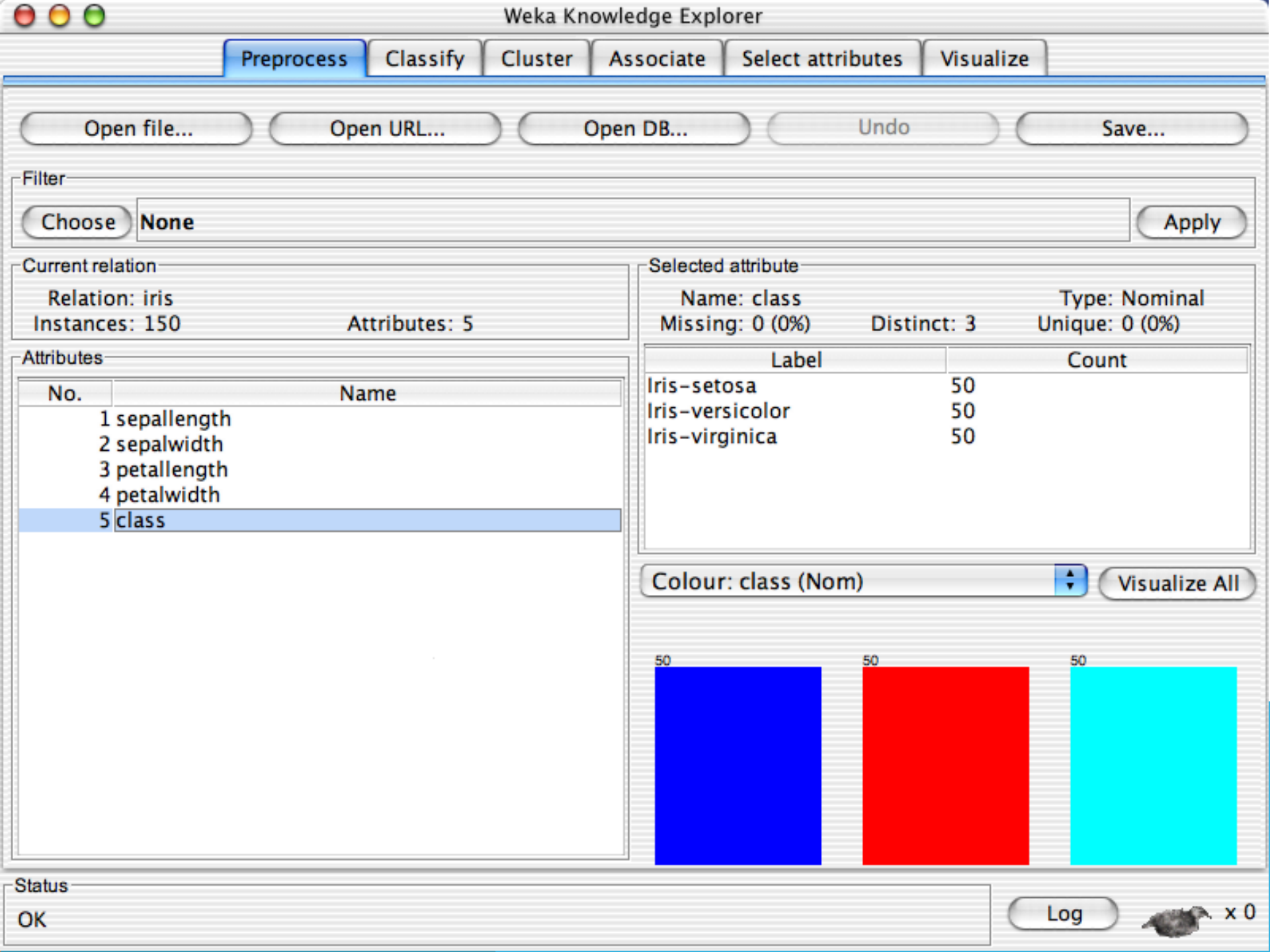
Colour: class (Nom) | Visualize All



Status

OK

Log | x 0


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
None
Apply
Current relation
Relation: iris
Instances: 150
Attributes: 5
Attributes
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Selected attribute
Name: class
Type: Nominal
Missing: 0 (0%)
Distinct: 3
Unique: 0 (0%)
Label
Count
Iris-setosa 50
Iris-versicolor 50
Iris-virginica 50
Colour: class (Nom)
Visualize All
50
50
50
Status
OK
Log
x 0

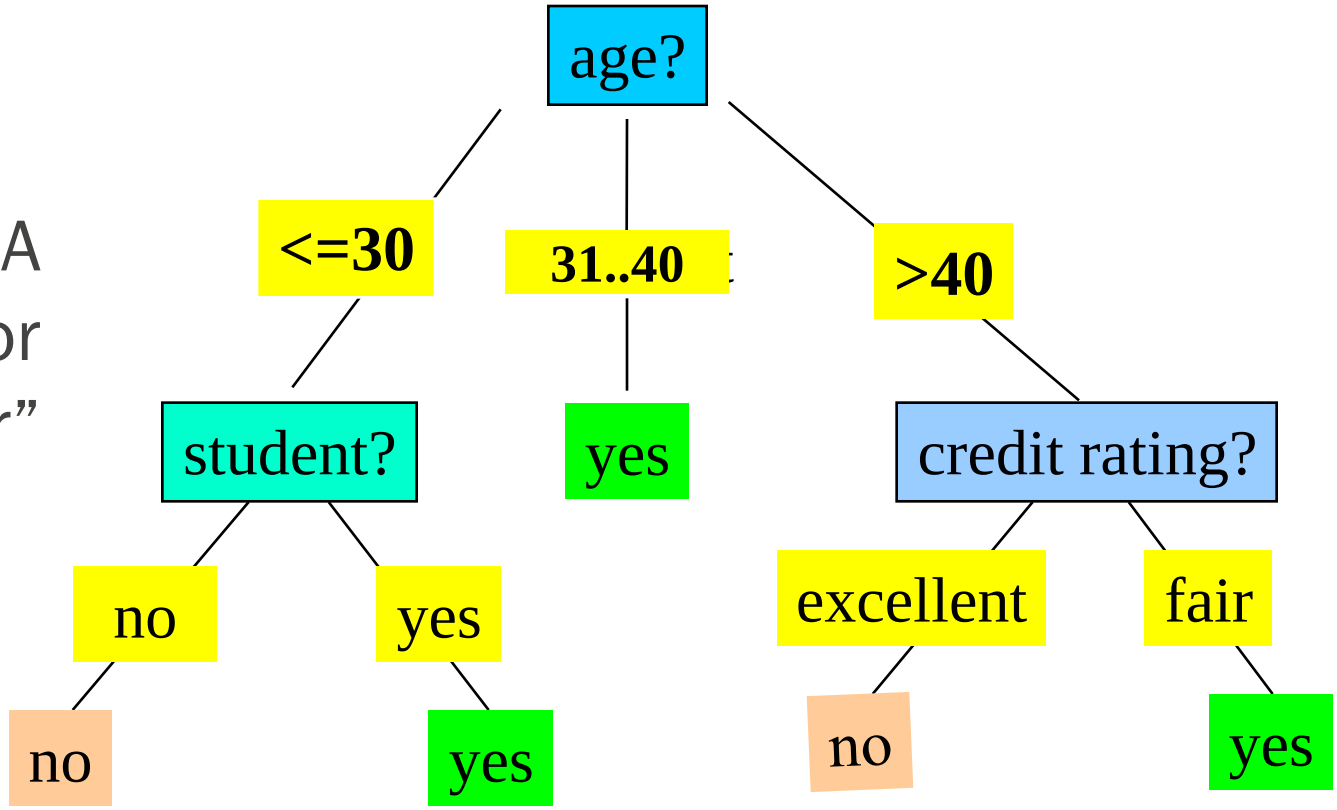
EXPLORER: BUILDING "CLASSIFIERS"

Classifiers in WEKA are models for predicting nominal or numeric quantities

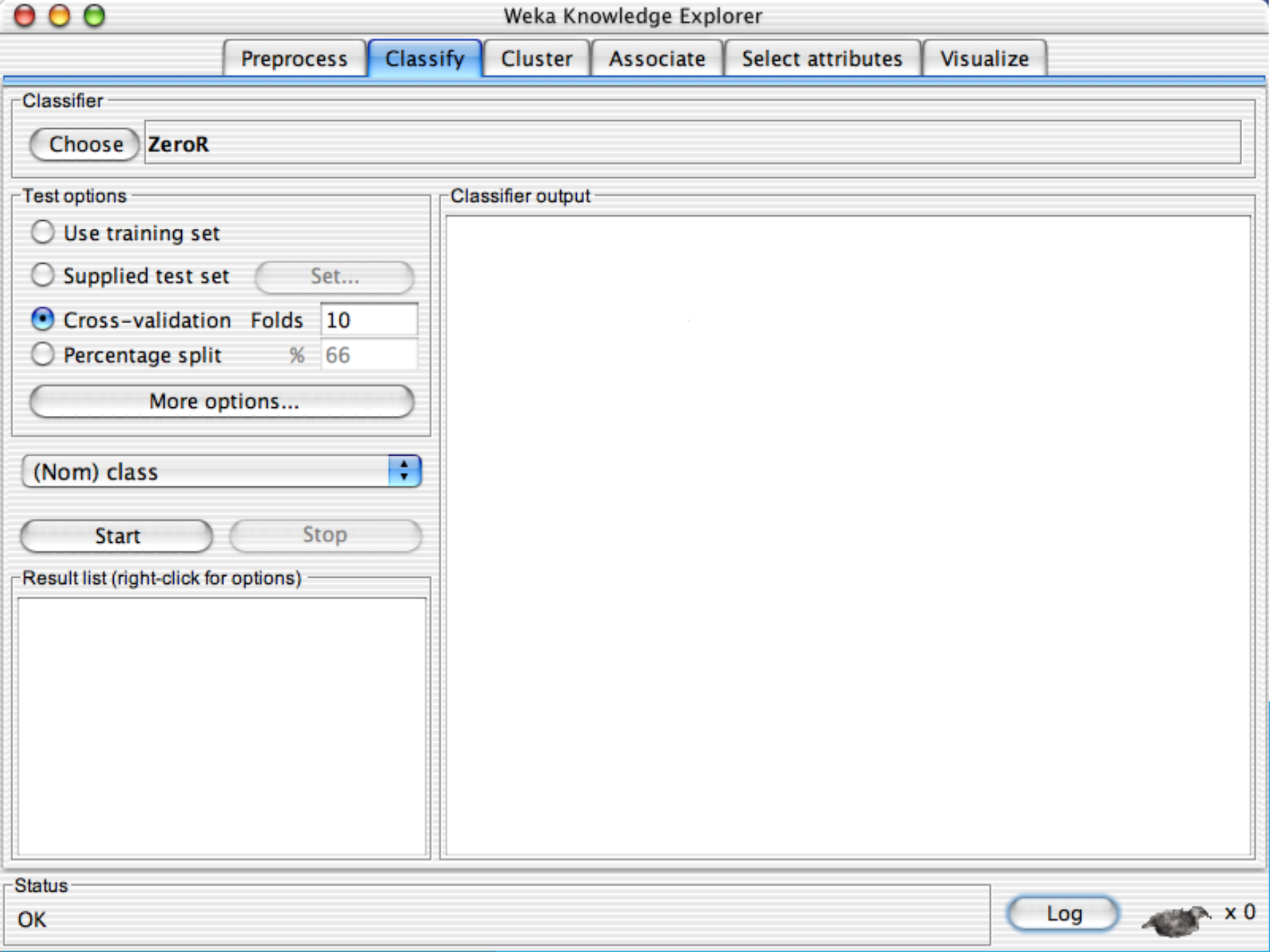
Implemented learning schemes include:

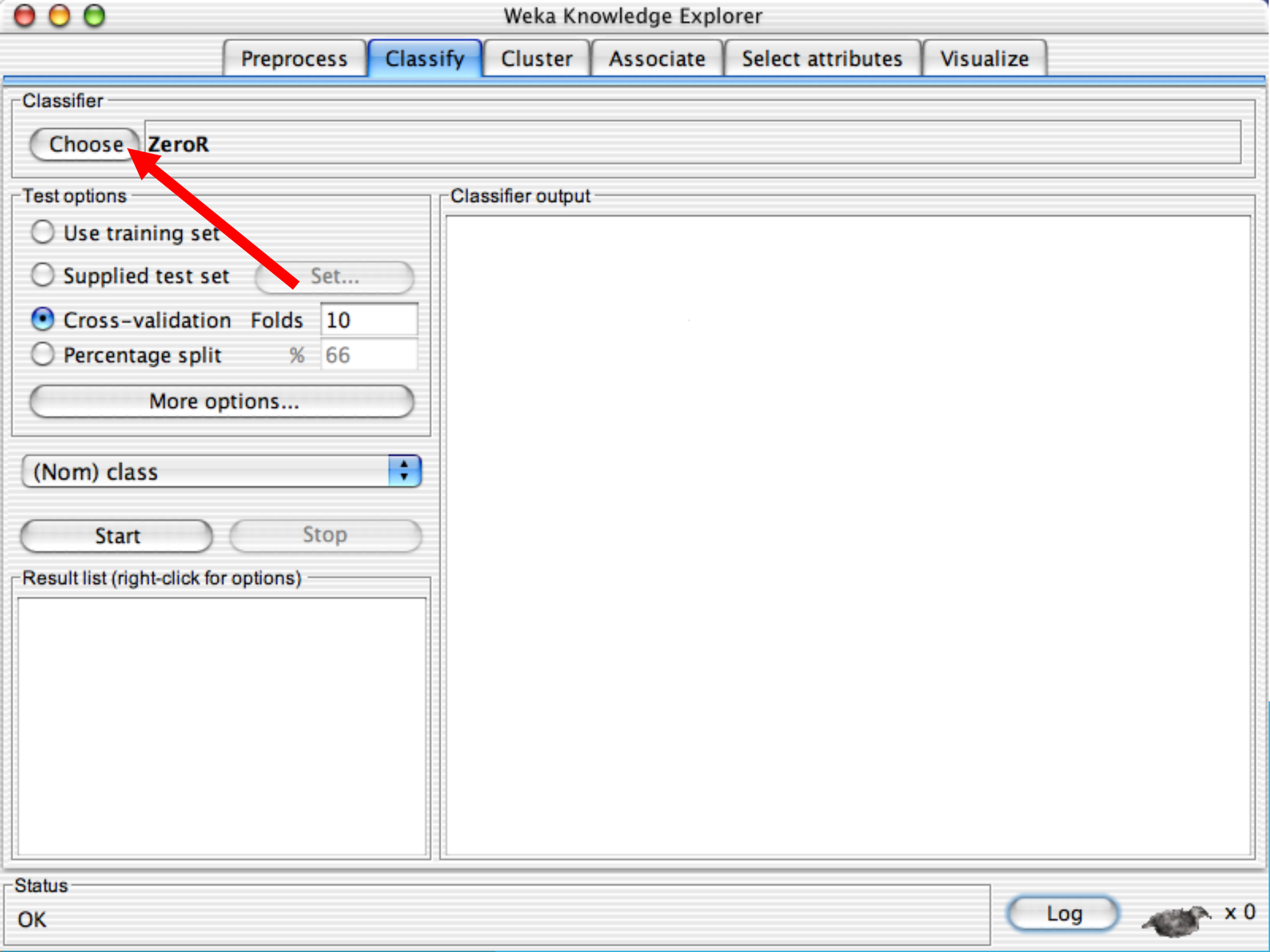
- Decision trees and lists, instance-based classifiers, support vector machines, multi-layer perceptrons, logistic regression, Bayes' nets, …

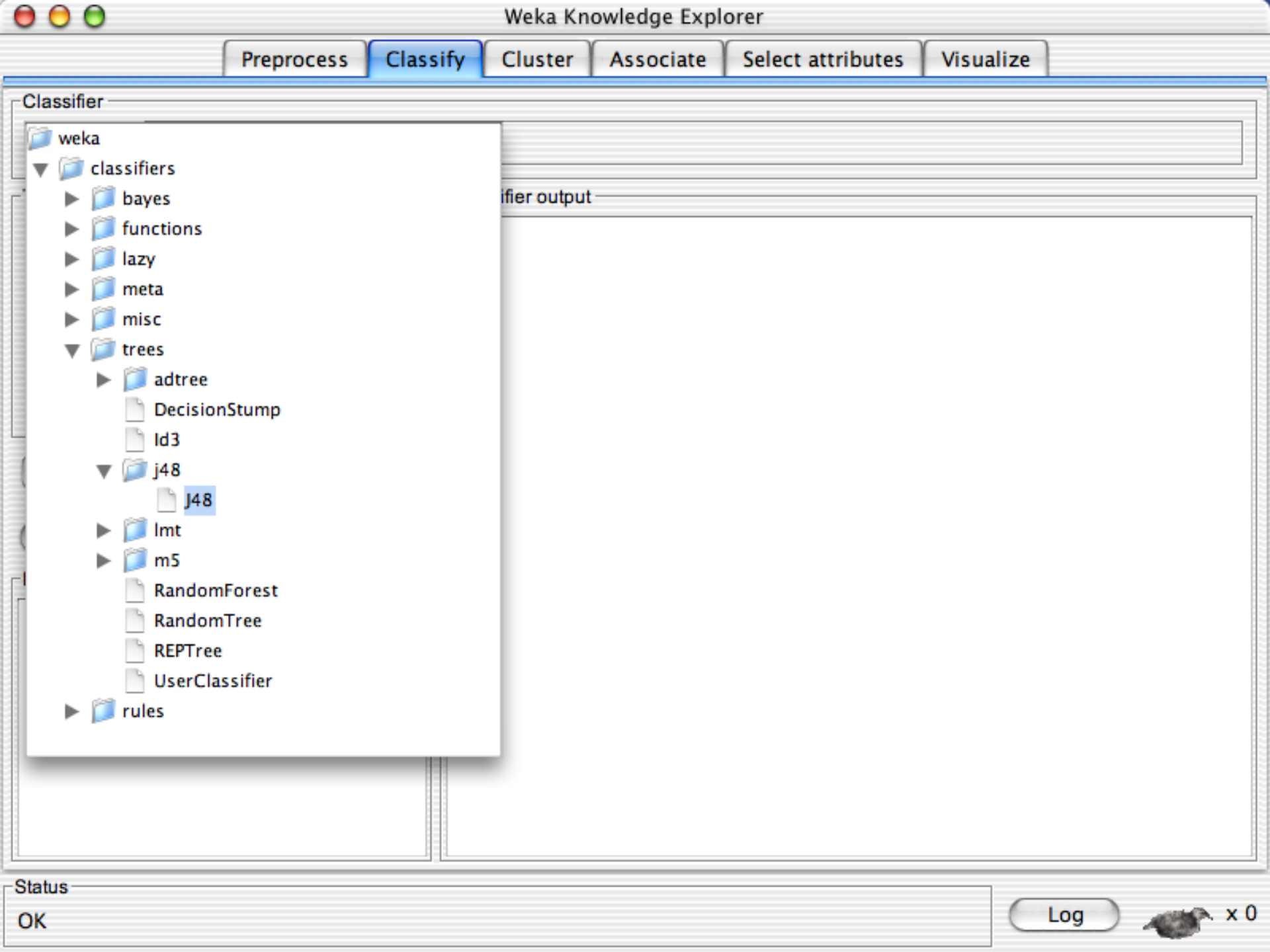
Output: A Decision Tree for "buys_computer"

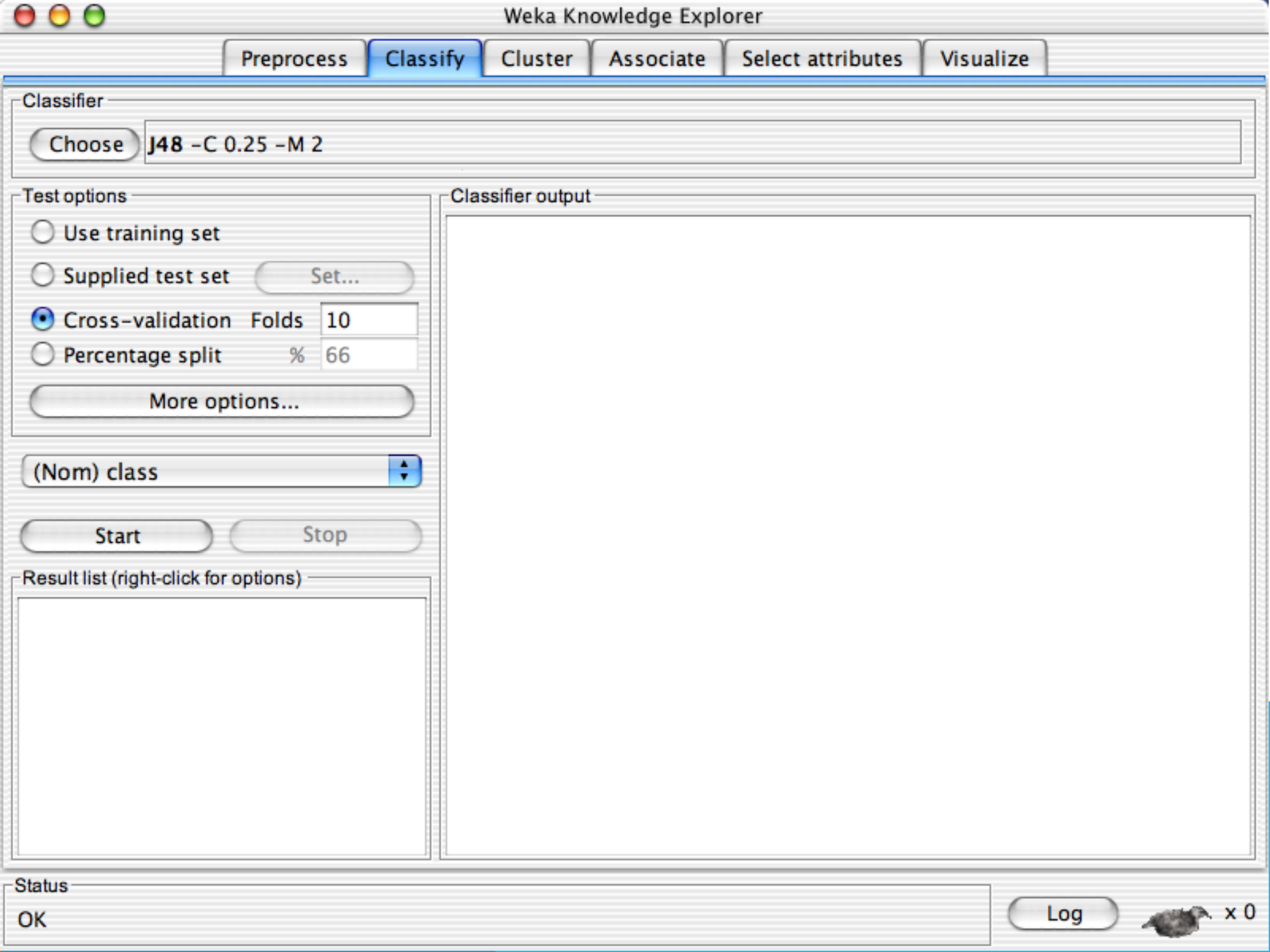


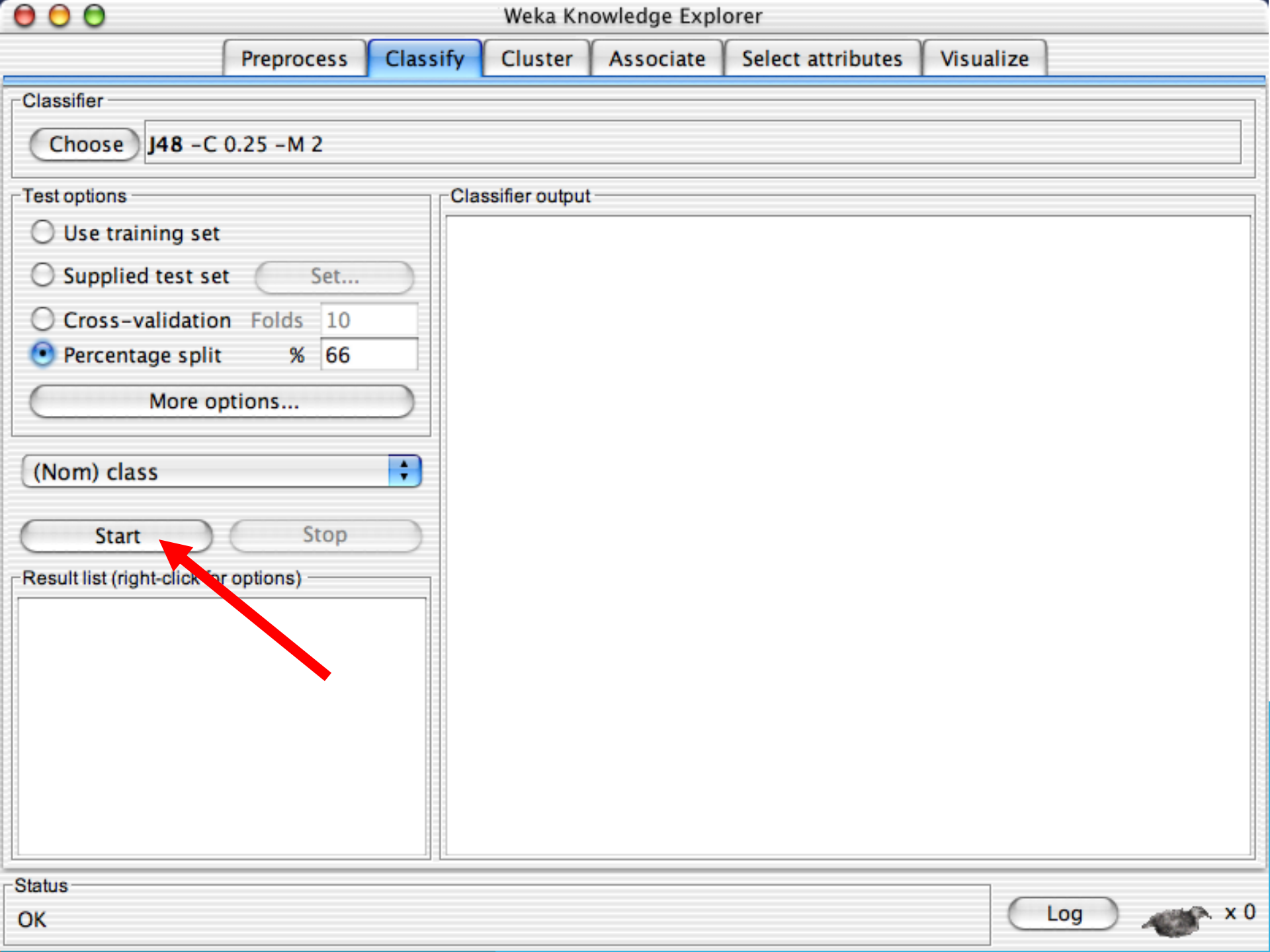
WEKA: DECISION TREE


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


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


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
weka
classifiers
bayes
functions
lazy
meta
misc
trees
adtree
DecisionStump
Id3
j48
J48
lmt
m5
RandomForest
RandomTree
REPTree
UserClassifier
rules
ifier output
Status
OK
Log
x 0


Weka Knowledge 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...
Classifier output
(Nom) class
Start
Stop
Result list (right-click for options)
Status
OK
Log
x 0


Weka Knowledge 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...
Classifier output
(Nom) class
Start
Stop
Result list (right-click for options)
Status
OK
Log
x 0

Weka Knowledge 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)

11:49:05 - trees.j48.J48

Classifier output

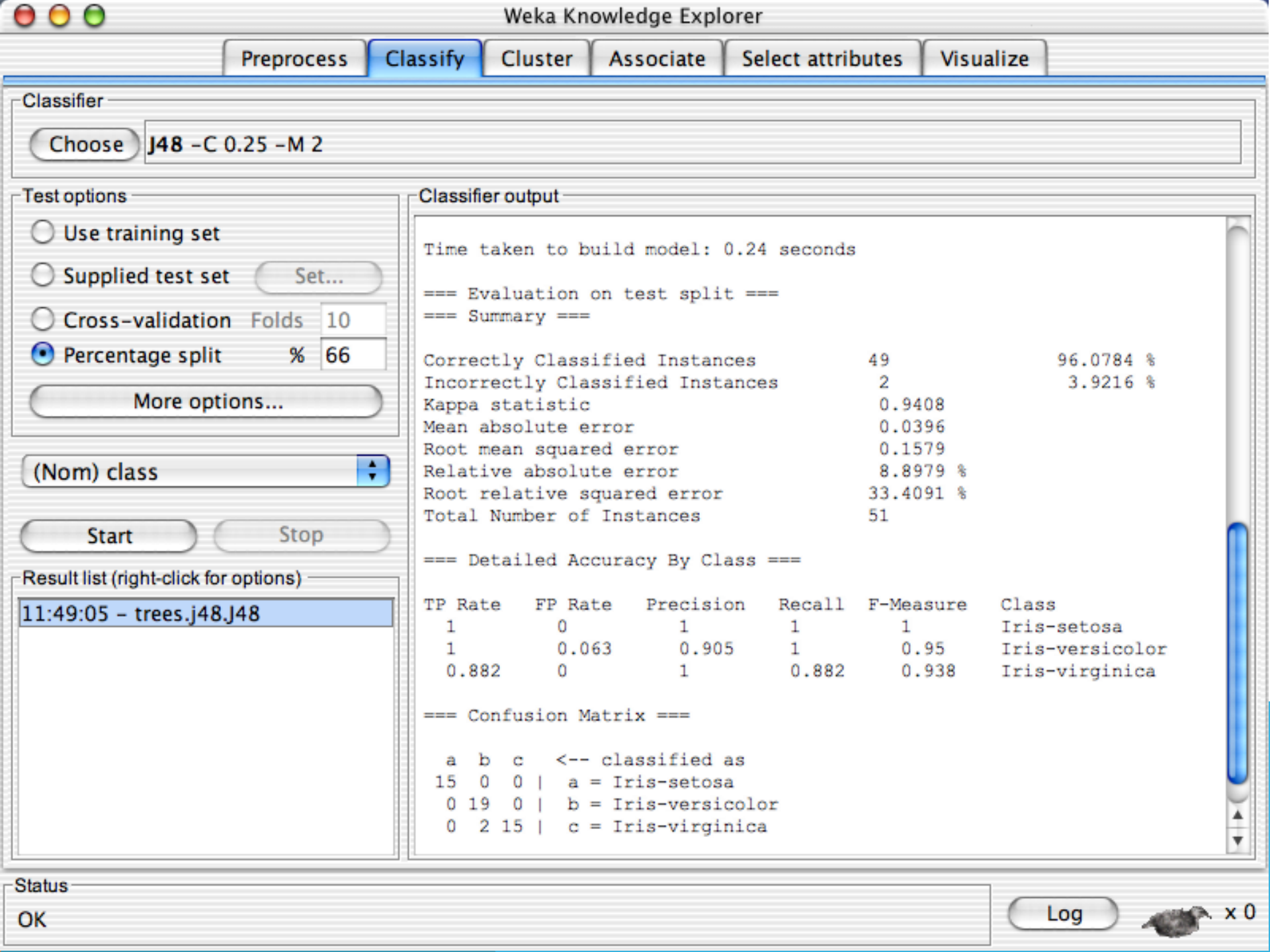
```
=== Run information ===

Scheme:       weka.classifiers.trees.j48.J48 -C 0.25 -M 2
Relation:     iris
Instances:    150
Attributes:   5
              sepallength
              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
|   petalwidth <= 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
```

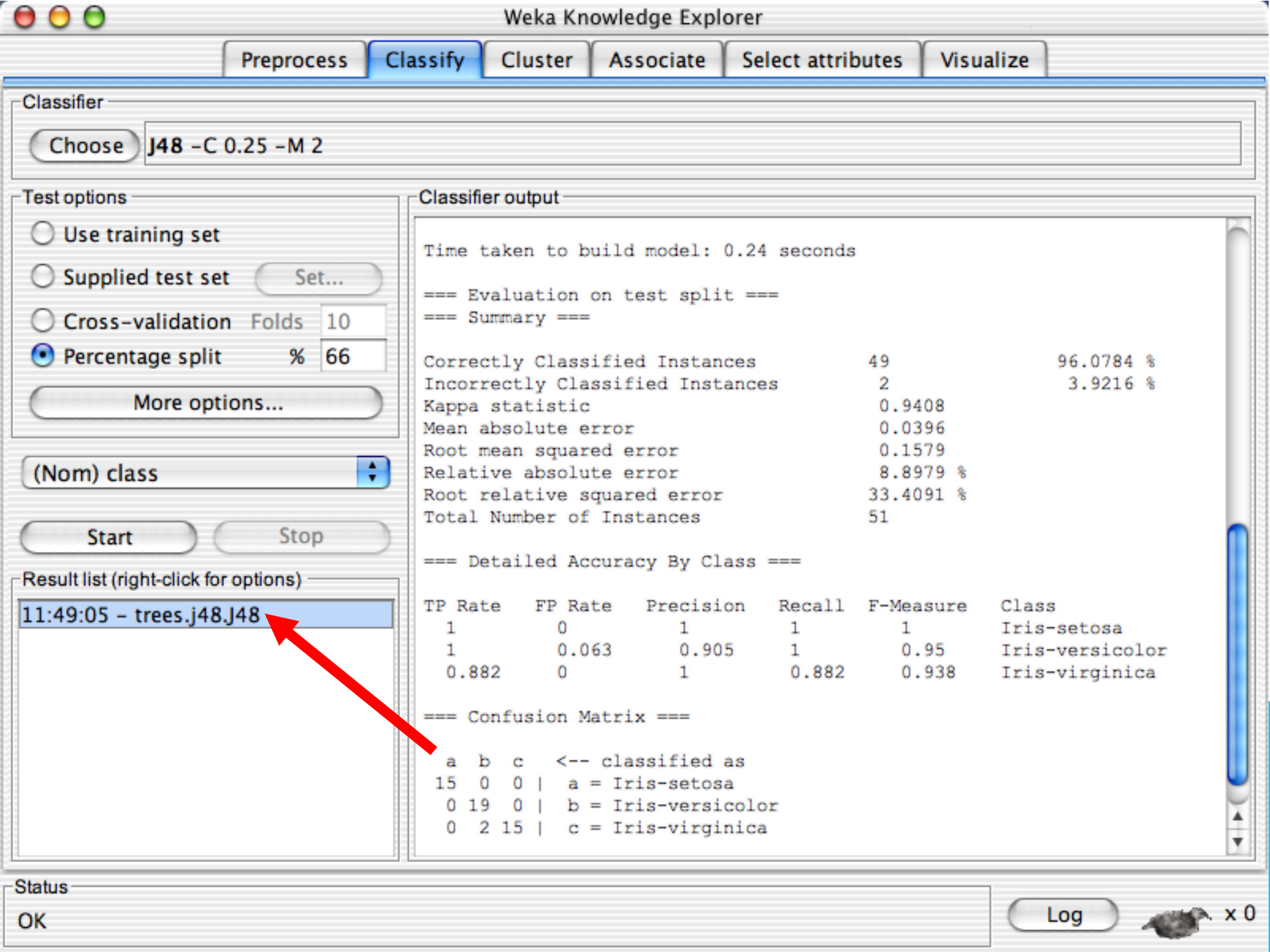
Status

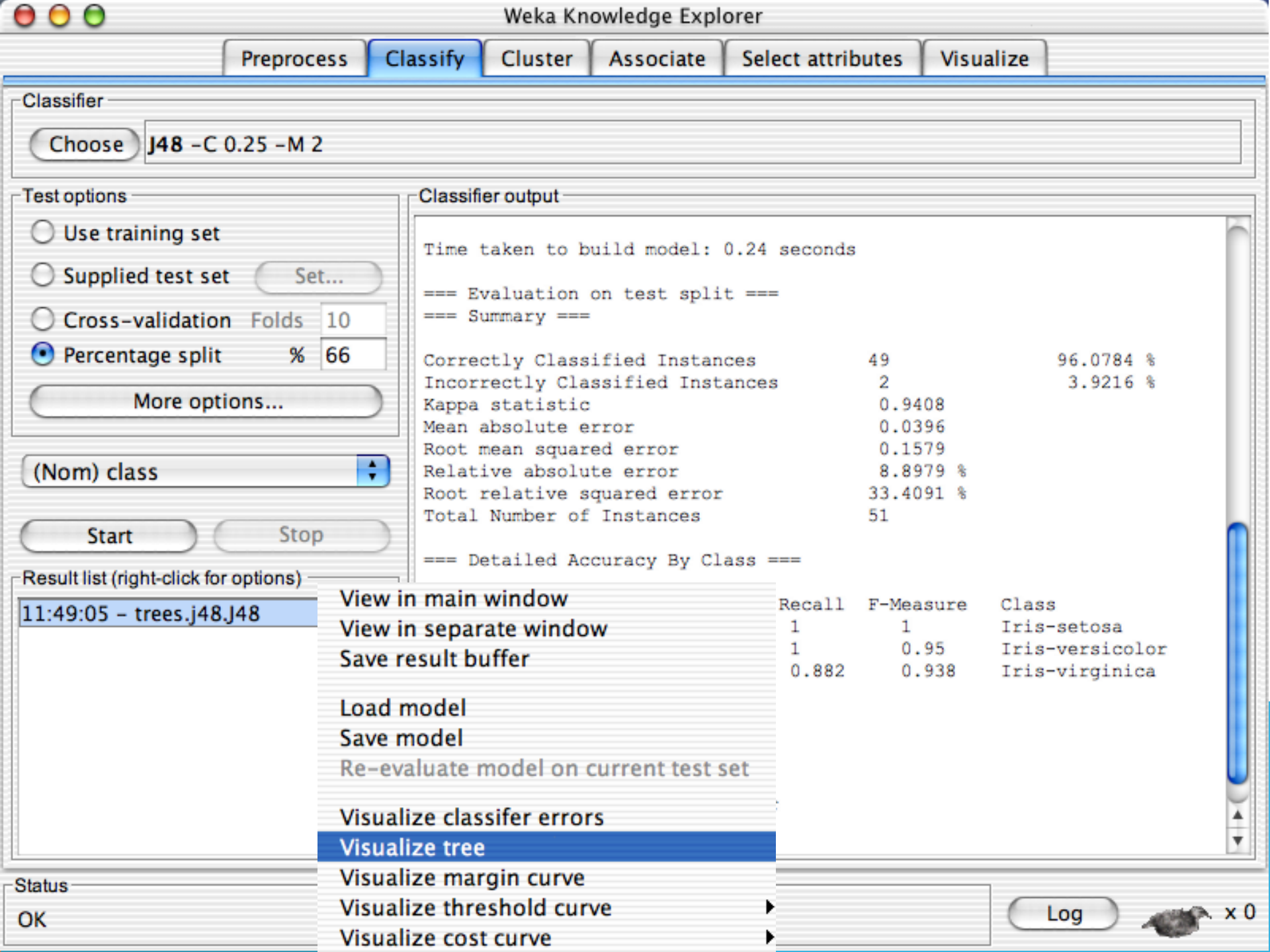
OK

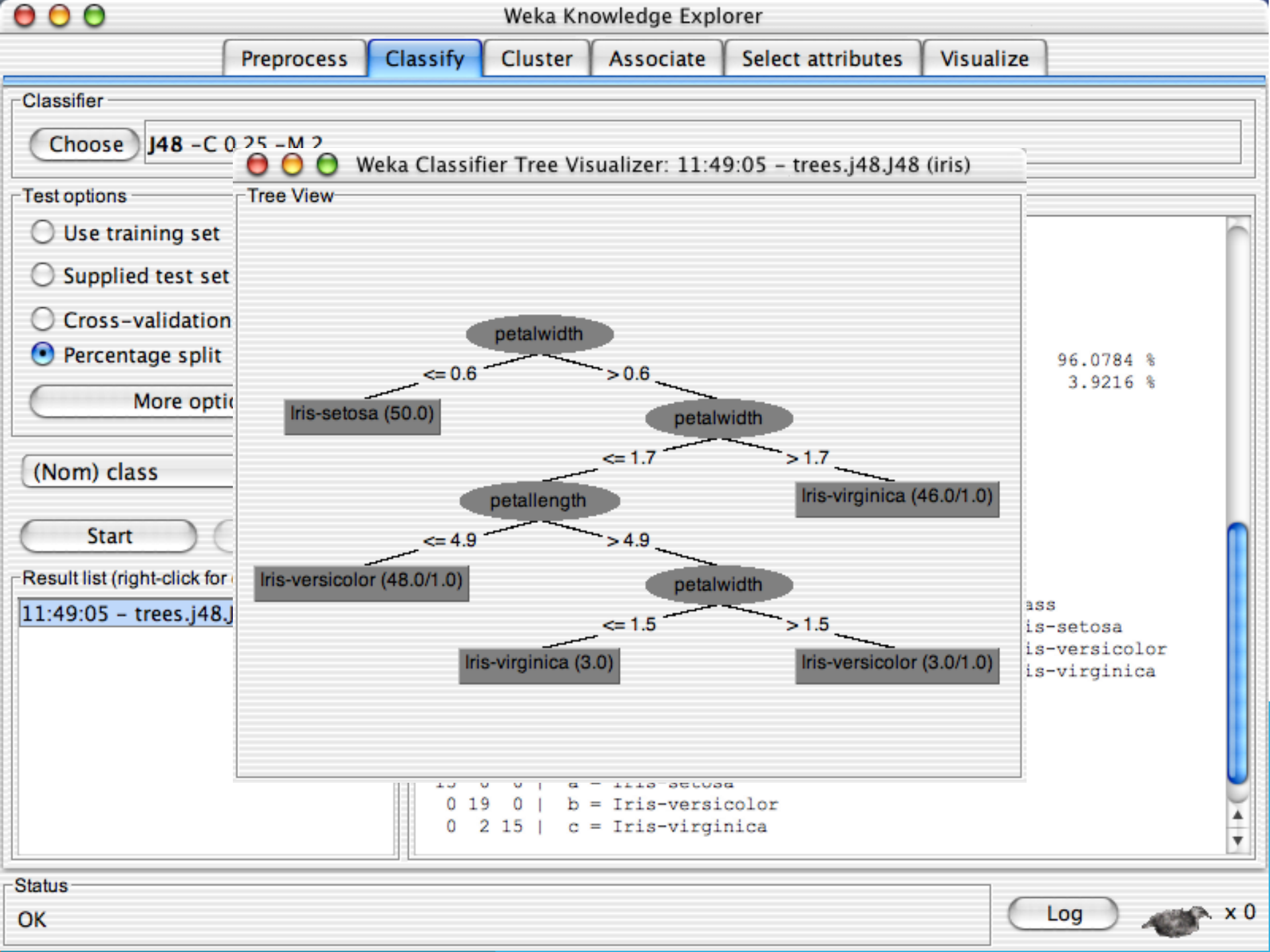
Log

x 0


Weka Knowledge 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)
11:49:05 - trees.j48.J48
Classifier output
Time taken to build model: 0.24 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 squared 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


Weka Knowledge 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)
11:49:05 - trees.j48.J48
Classifier output
Time taken to build model: 0.24 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 squared 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


Weka Knowledge 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)
11:49:05 - trees.j48.J48
Classifier output
Time taken to build model: 0.24 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 squared error 0.1579
Relative absolute error 8.8979 %
Root relative squared error 33.4091 %
Total Number of Instances 51
=== Detailed Accuracy By Class ===
Recall F-Measure Class
1 1 Iris-setosa
1 0.95 Iris-versicolor
0.882 0.938 Iris-virginica
View in main window
View in separate window
Save result buffer
Load model
Save model
Re-evaluate model on current test set
Visualize classifer errors
Visualize tree
Visualize margin curve
Visualize threshold curve
Visualize cost curve
Status
OK
Log
x 0


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Weka Classifier Tree Visualizer: 11:49:05 - trees.j48.J48 (iris)
Tree View
petalwidth
<= 0.6
> 0.6
Iris-setosa (50.0)
petalwidth
<= 1.7
> 1.7
petallength
Iris-virginica (46.0/1.0)
<= 4.9
> 4.9
Iris-versicolor (48.0/1.0)
petalwidth
<= 1.5
> 1.5
Iris-virginica (3.0)
Iris-versicolor (3.0/1.0)
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More optic
(Nom) class
Start
Result list (right-click for
11:49:05 - trees.j48.J
96.0784 %
3.9216 %
ass
is-setosa
is-versicolor
is-virginica
0 19 0 | b = Iris-versicolor
0 2 15 | c = Iris-virginica
Status
OK
Log
x 0

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)

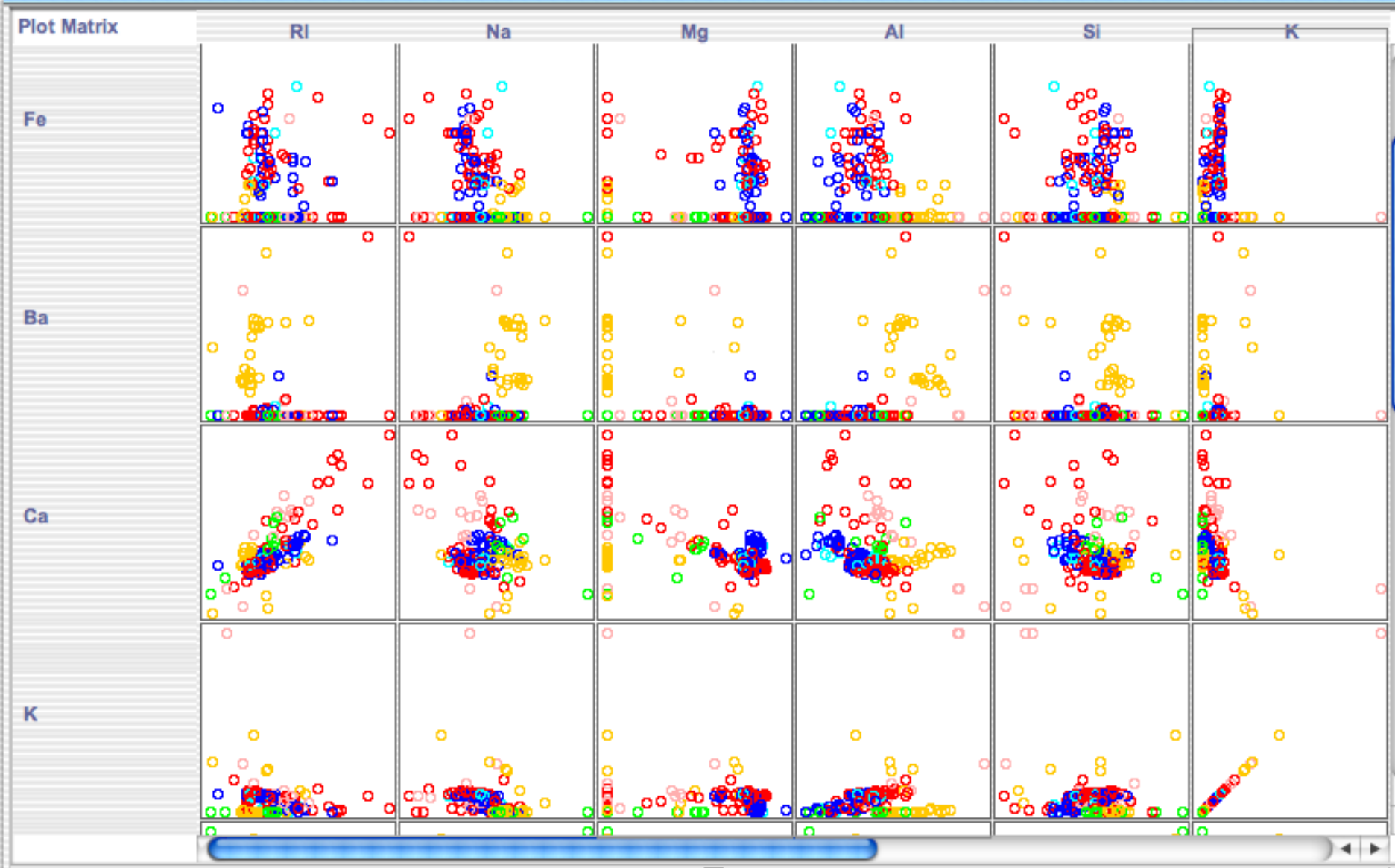
Color-coded class values

"Jitter" option to deal with nominal attributes (and to detect "hidden" data points)

"Zoom-in" function

Weka Knowledge Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize



Status

OK

Log

x 0

Weka Knowledge Explorer: Visualizing Glass

X: Al (Num)

Y: Ca (Num)

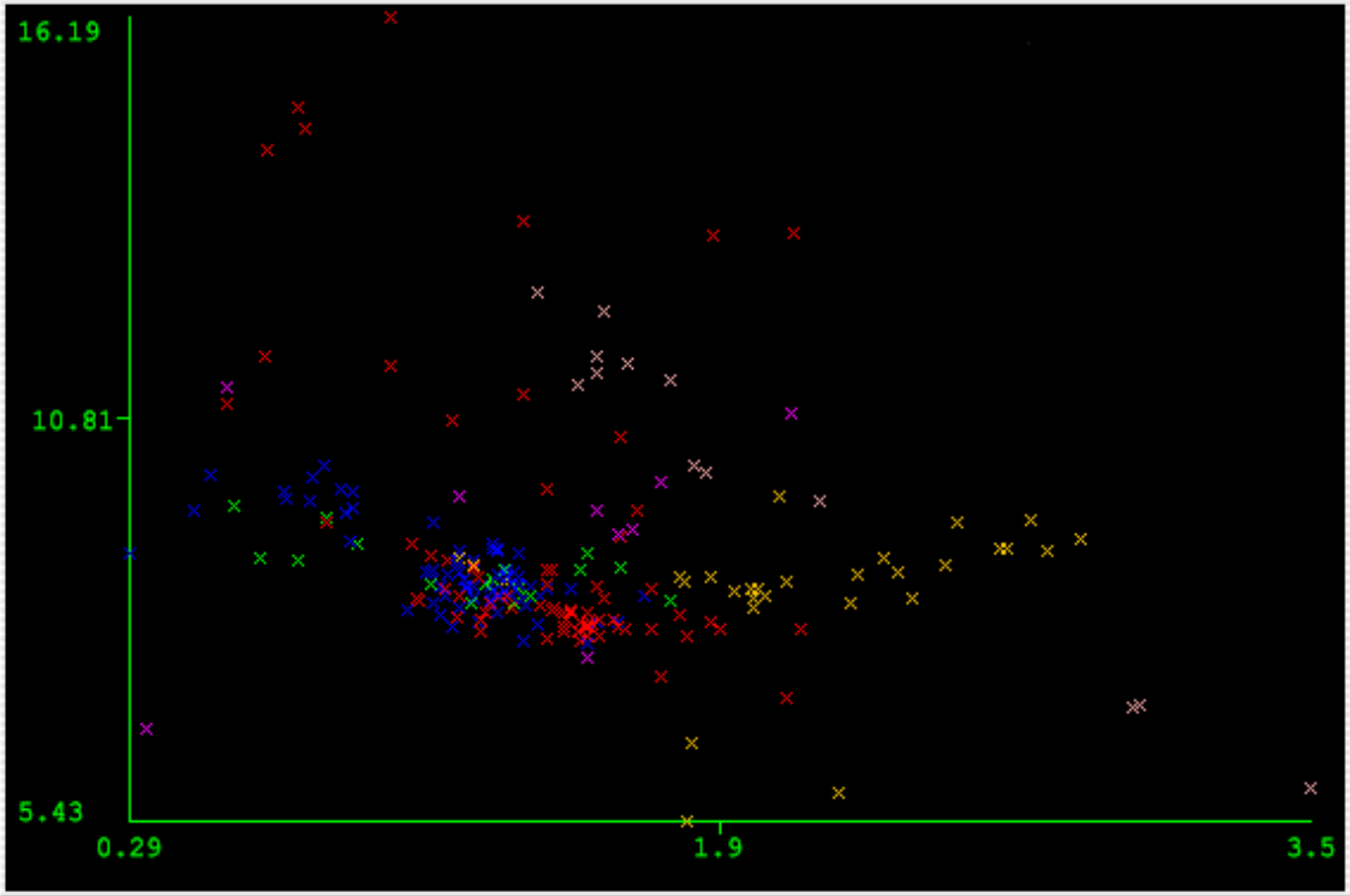
Colour: Type (Nom)

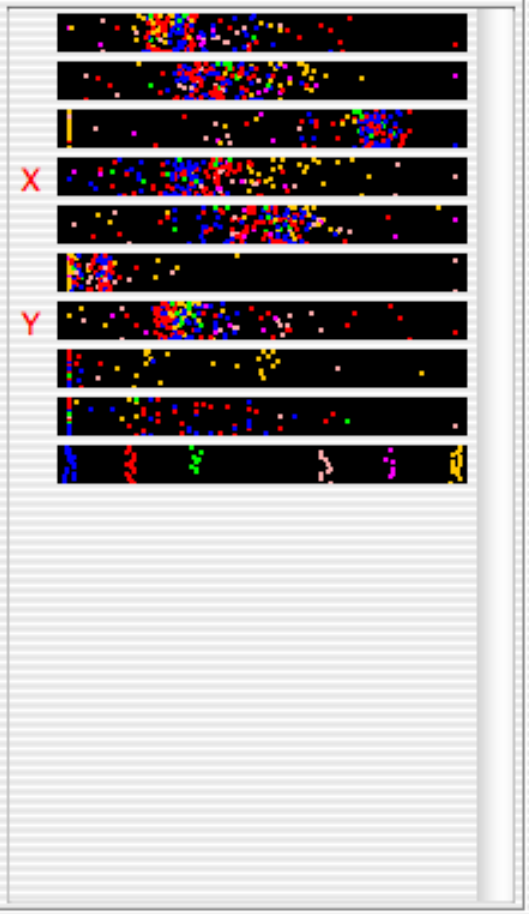
Select Instance

Reset | Clear | Save

Jitter

Plot: Glass





Class colour

build wind float build wind non-float vehic wind float

vehic wind non-float containers tableware headlamps

REFERENCES AND RESOURCES

References:

- WEKA website: http://www.cs.waikato.ac.nz/~ml/weka/index.html
- WEKA Tutorial:
  - Machine Learning with WEKA: A presentation demonstrating all graphical user interfaces (GUI) in Weka.
  - A presentation which explains how to use Weka for exploratory data mining.
- WEKA Data Mining Book:
  - Ian H. Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)
- WEKA Wiki: http://weka.sourceforge.net/wiki/index.php/Main_Page
- Others:
  - Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, 2nd ed.