EMPIRICAL SOFTWARE ENGINEERING (VERSION 2.0) AND DATA MINING

SWINBURNE

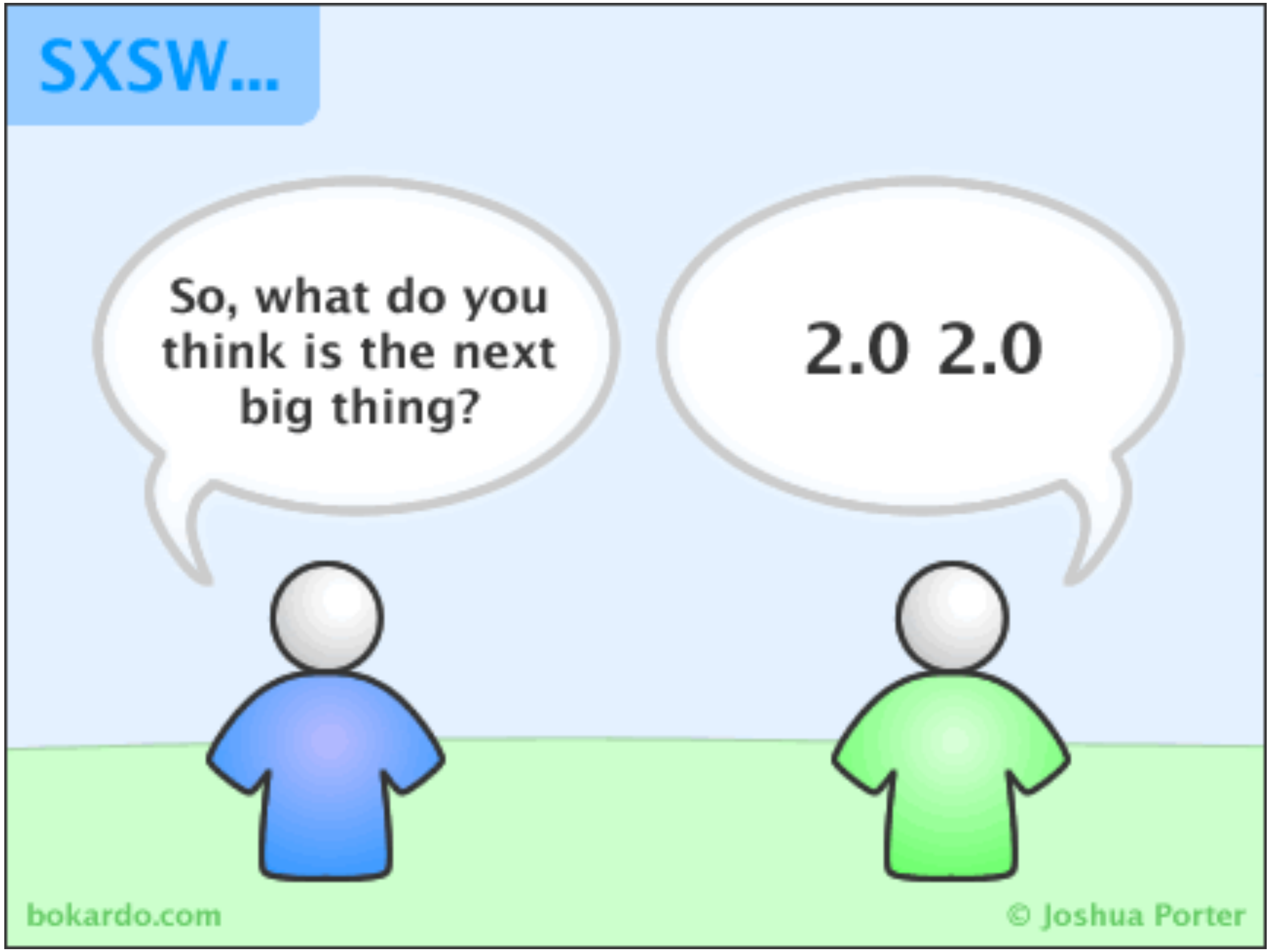
CENTRE FOR COMPUTING AND ENGINEERING SOFTWARE SYSTEMS

NOV 24, 2010

Tim Menzies,
WVU, USA, tim@menzies.us, http://menzies.us

Download: http://unbox.org/wisp/var/timm/10/swinburne

<u>Version history</u>
V1: Aug18 '10
V1a: Aug28 '10
V1b: Sept02 '10
V1c: Sept11'10
V2: Nov23'10


SXSW...
So, what do you think is the next big thing?
2.0 2.0
bokardo.com
© Joshua Porter

Road map

1. Data mining & SE (overview)
2. Data mining tools (guided tour of "WEKA")
3. Data "carving" (core operators of DM)

---

4. Generality (or not)
5. Bias (is your friend)
6. Evaluation (does it really work?)

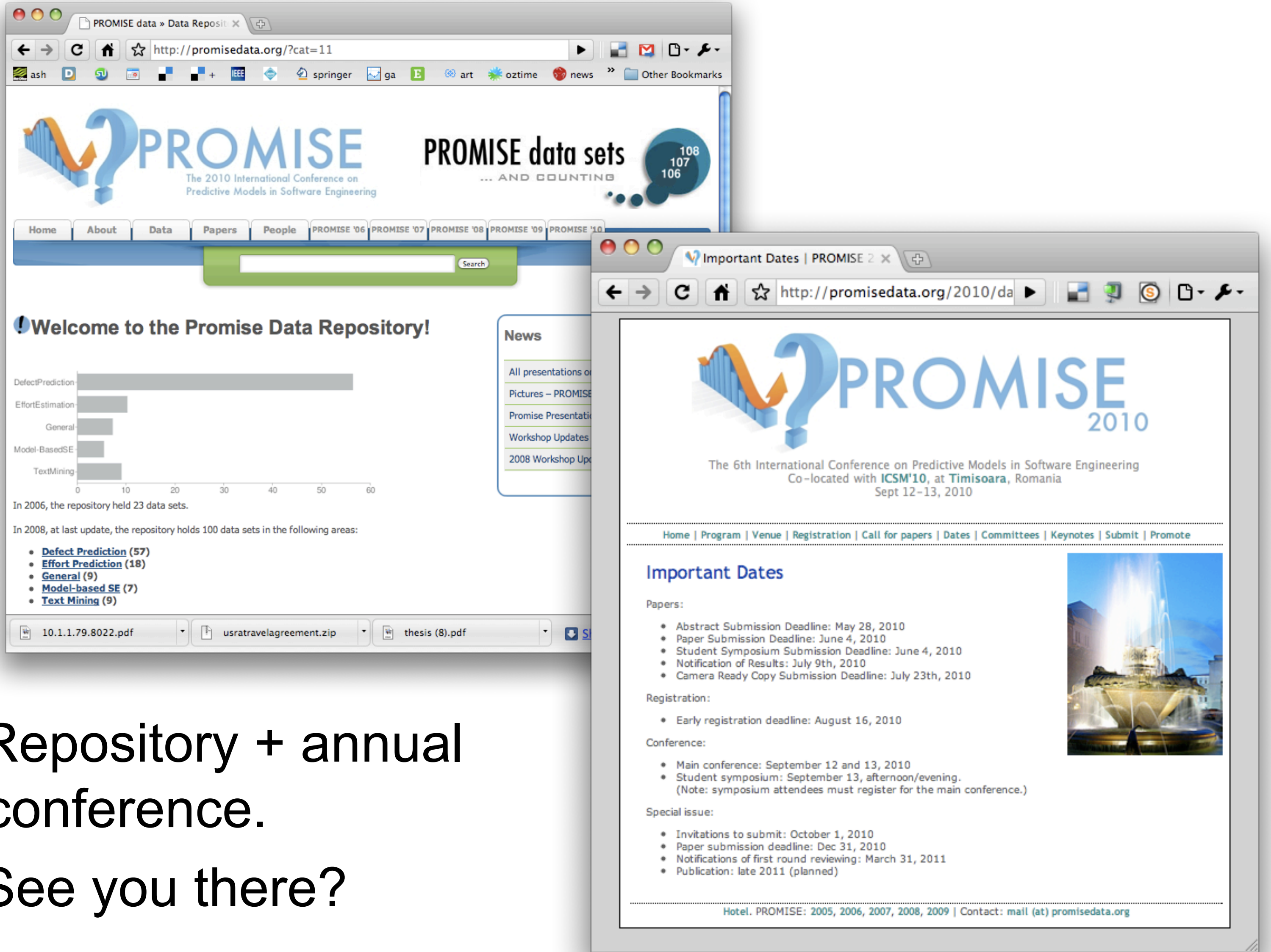
Change log

- Version 1: Aug 18, 2010
  - Version 1a: Aug 28, 2010
    - 2 more slides on "why empirical SE v2.0"
  - Version 1b: Sept 2: minor edits
  - Version 1c: minor edits
  - Version 1d: new conclusion
- Version 2: Nov 21, 2010
  - Shorter version for Swinburne

About the author

- Dr. Tim Menzies (tim@menzies.us) has worked on advanced modeling + AI since 1986.
  - PhD from Uni. New South Wales, Sydney, Oz
  - Assoc/prof at WVU CS &EE
- Former research chair for NASA
- Author of 200 refereed papers: http://menzies.us/papers.php
- Co-founder and organizer of the PROMISE conferences on repeatable experiments in SE
- For more, see http://menzies.us

http://promisedata.org/data



Repository + annual conference.

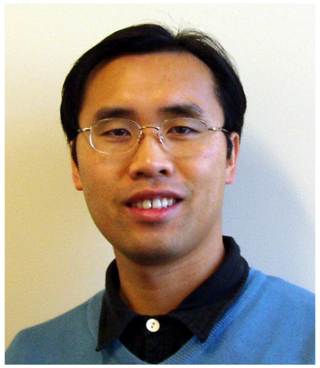
See you there?

New trend

- Ph.D. students, finishing up their studies, using PROMISE to archive their data
  - E.g. 30 new OO data sets from Marian Jureczko
- Are you next?

For other view on DM + SE

- ICSE 2010 Tutorial T18 Tuesday, 4 May 2010 (afternoon)
- Mining Software Engineering Data
  - Ahmed E. Hassan: Queen's University, Canada
  - Tao Xie: North Carolina State University, USA



- Tutorial Slides:
  - https://sites.google.com/site/asergrp/dmse/dmse-icse08-tutorial.ppt?attredirects=0

DATA MINING & SE (OVERVIEW)

Road map

1. Data mining & SE (overview)
2. Data mining tools (guided tour of "WEKA")
3. Data "carving" (core operators of DM)

A proposal

- Add roller skates to scientific analysis

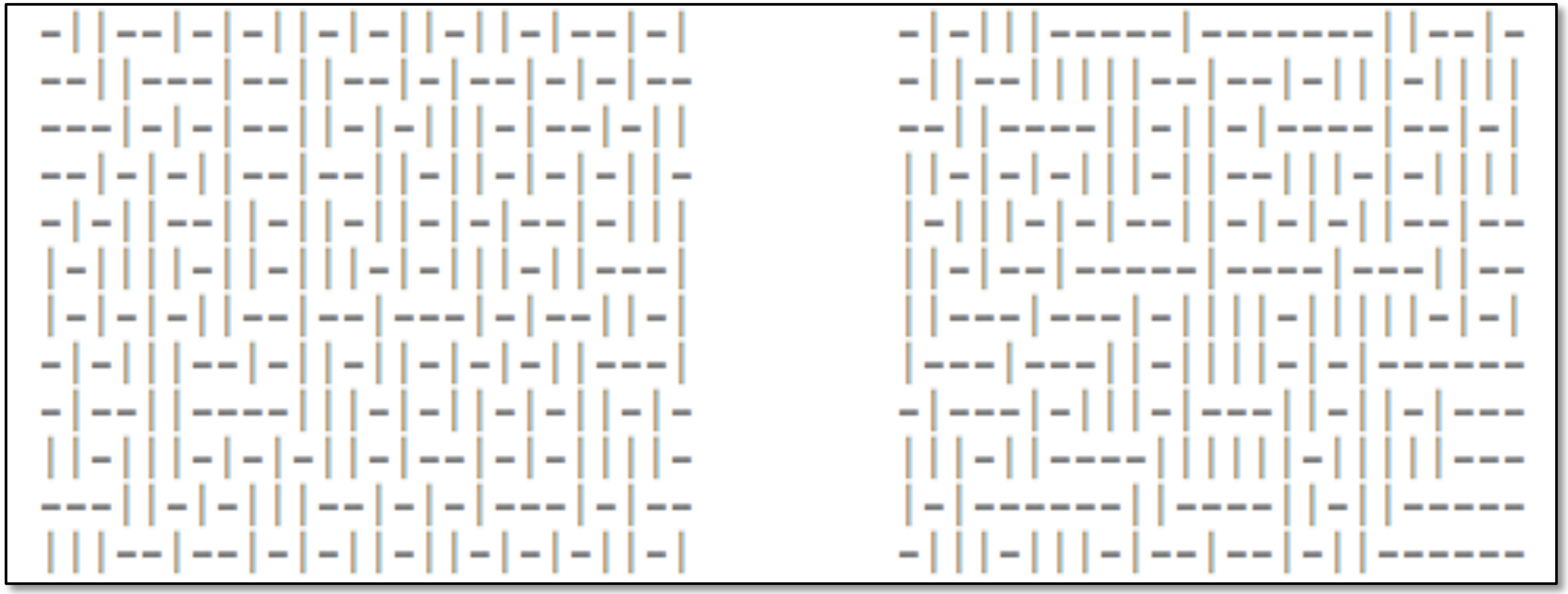
- Always use data mining on SE data

Data Mining: definition

- Finding patterns in (lots of) data
  - Diamonds in the dust
- Combines statistics, AI, visualization, ….
- Synonyms
  - Machine learning
  - Business intelligence
  - Predictive analytics
- The art of the approximate scalable analysis
  - Bigger is better
- Used for… anything
  - The review of current beliefs w.r.t. new data is the hallmark of human rationality.
  - It is irrational <u>NOT</u> to data mine.

Exercise #1

- One these these things is not like the other
  - One was generating by selecting "-" or "|" at random, 300 times.
- Which one?



Exercise #2

- A little experiment from http://www.youtube.com/v/vJG698U2Mvo&hl=en_US&fs=1&rel=0

- Rules
  - No one talks for the next 4 minutes
  - If you know what is about to happen, see (1)

- This is a selective attention test
  - Count the number of times the team with the white shirt passes the ball.

What have we learned?

- Lesson #1:
  - Algorithms can be pretty dumb
  - If they don't focus on X, they see any Y, at random.
- Lesson #2:
  - Humans can be pretty dumb
  - If they mono-focus on X, you can miss Y
- Maybe, any induction process is a guess
  - And while guessing can be useful
  - Guesses can also be wrong
- Lets us a create community of agents, each with novel insights and limitations
  - Data miners working with humans
  - Maybe in combination, we can see more that separately

Wikipedia:
List of cognitive biases
http://en.wikipedia.org/wiki/List_of_cognitive_biases
- 38 decision making biases
- 30 biases in probability
- 18 social biases,
- 10 memory biases

Applications

- Effort estimation
- Defect prediction
- Optimization of discrete systems
- Test case generation
- Fault localization
- Text mining
- Temporal sequence mining
  - Learning software processes
  - Learning APIs
- Etc
- Welcome to Empirical SE, Version 2.0

Applications

- Effort estimation
- Defect prediction
- Optimization of discrete systems
- Test case generation
- Fault localization
- Text mining
- Temporal sequence mining
  - Learning software processes
  - Learning APIs
- Etc
- Welcome to Empirical SE, Version 2.0

Data mining applications explored by me since 2007.

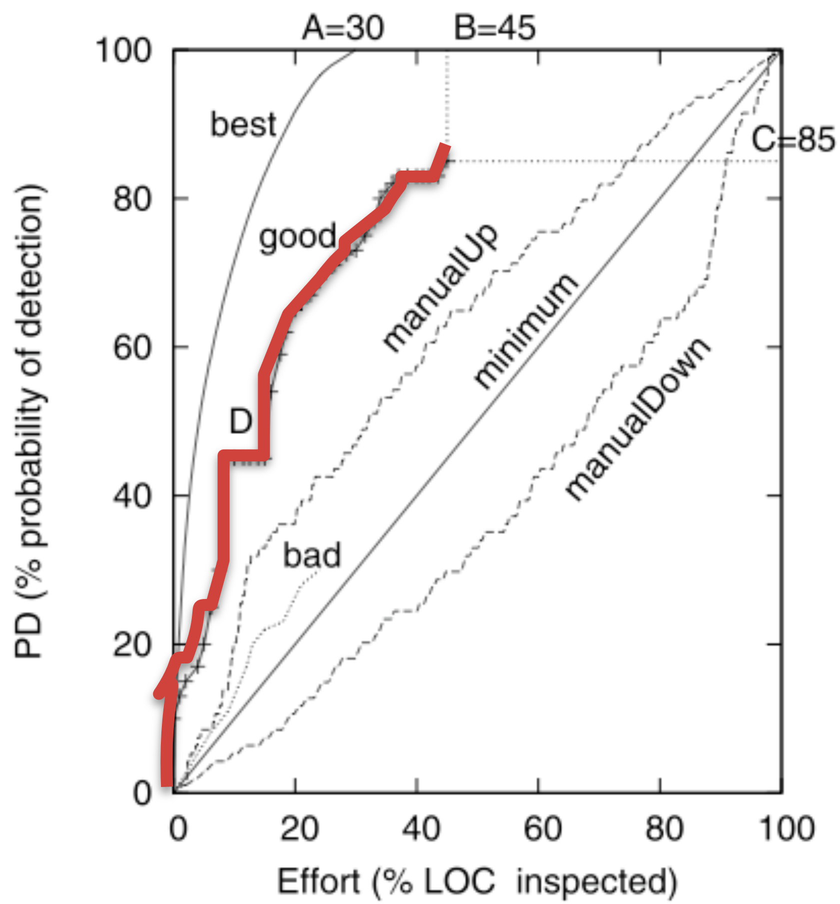
A career in data mining is a very diverse career, indeed

Application: Effort estimation

- Can we predict development effort (time * staff)?
- E.g. using linear regression; effort = $a*KLOC^b$ c
  - Boehm, B.W. 1981 Software Engineering Economics
  - Boehm, B.W., Clark, Horowitz, Brown, Reifer, Chulani, Madachy, R., and Steece, B. 2000 Software Cost Estimation with Cocomo II
  - Sunita Chulani, Barry W. Boehm, Bert Steece: Bayesian Analysis of Empirical Software Engineering Cost Models IEEE Trans. Software Eng. 25(4): 573-583 (1999)
- E.g. using analogy
  - Describe past projects according to N dimensions
  - Float all known projects in an N-dimensional space
  - To estimate a project, insert into that space; query its nearest neighbors
  - For the classic estimation via analogy, see
    - Martin J. Shepperd, Chris Schofield: Estimating Software Project Effort Using Analogies IEEE Trans. Software Eng. 23(11): 736-743 (1997)
  - For 12,000+ variants to that process, see
    - Fig1 of http://menzies.us/pdf/10stable.pdf
- E.g. using other methods:
  - See 154 variants in http://menzies.us/pdf/10stable.pdf

Application: Defect Prediction

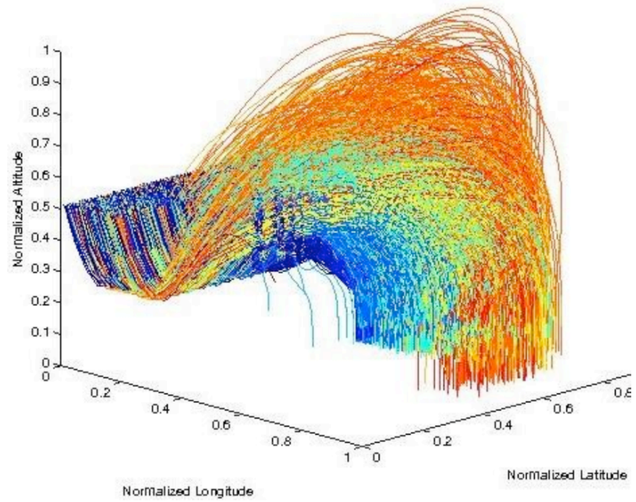
- Limited QA budgets, can't check everything.
  - Where should we place our inspection effort?
- For a review, see Section Two of
  - http://menzies.us/pdf/10which.pdf
- Practical value:
  - How to inspect less, and find more bugs



Application:
Optimizations of discrete systems

- Standard numeric optimizers assume continuous, possibly even linear, equations
- Data miners much happier to work in discrete spaces.
- What factors predict for landing closest to the target?
  - State-of-the-art optimizer
  - Simulated annealing
  - the TAR3 data miner
  - TAR3 45 times faster, found better solutions

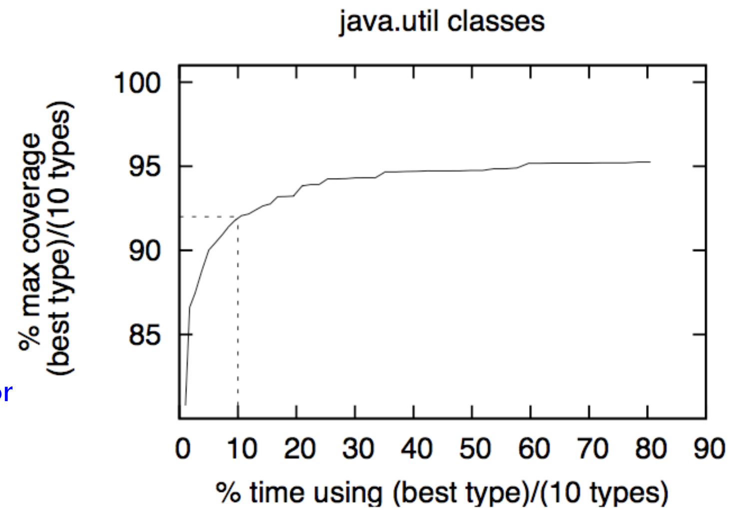
http://menzies.us/pdf/10keys.pdf



Application: Test Case Generation

- NIGHTHAWK: A genetic algorithm that mutates sequences of method calls in order to maximize code coverage.
- RELIEF: a data mining technique to find "interesting features"
    - Same attribute same values in all classes?
        - Boring
    - Same Attribute, different values in different classes?
        - Interesting
- RELIEF found that 90% of NIGHTHAWK's mutators were "boring"
    - Order of magnitude speed up in test generation

Rank	Gene type t	$avgMerit$
1	numberOfCalls	85
2	valuePoolActivityBitSet	83
3	upperBound	64
4	chanceOfTrue	50
5	methodWeight	50
6	numberOfValuePools	49
7	lowerBound	44
8	chanceOfNull	40
9	numberOfValues	40
10	candidateBitSet	34



- James H. Andrews, Tim Menzies, Felix C.H. Li, "Genetic Algorithms for Randomized Unit Testing," IEEE Transactions on Software Engineering, 25 Mar. 2010.

Application: Fault Localization

- 100,000 JAVA methods
  - In a matrix T*D
  - T = "terms" = all the method calls in each method
  - D = "documents" = all the methods

- Bug report
  - Replace text with just the method calls it mentions
  - Add edited report as row D+one in the matrix
  - Compute similarity of D+one to other rows (cosine similarity)
  - The actual buggy method is in the closest 100 methods
  - Use relevancy feedback to narrow down the search

- Gregory Gay, Sonia Haiduc, Andrian Marcus Tim Menzies: On the use of relevance feedback in IR-based concept location ICSM 2009: 351-360

Application: Text Mining

- 80% of data in organizations is unstructured
  - Not in databases, or XML schemas
  - But in the natural language of (say) Word documents
- Given enough of these seemingly unstructured documents, structures can be discovered
- E.g.
  - Thousands of natural language bug reports from NASA
  - Used "feature reduction" to find the top 100 most important words
  - Used standard data mining to learn predictors for defect severity from that top-100
  - Tim Menzies, Andrian Marcus: Automated severity assessment of software defect reports. ICSM 2008: 346-355

Application: Temporal Sequence Mining

- Learning software process descriptions
  - No more prescriptions of what we think goes on inside software projects
  - Lets look at see at what actually happens
    - Li, Mingshu and Boehm, Barry and Osterweil, Leon and Jensen, Chris and Scacchi, Walt "Experiences in Discovering, Modeling, and Reenacting Open Source Software Development Processes", Unifying the Software Process Spectrum, Lecture Notes in Computer Science, 2006, page 449 to 462
- Learning APIs from method sequence calls
  - Tao Xie and Jian Pei. MAPO: Mining API Usages from Open Source Repositories. In *Proceedings of the 3rd International Workshop on Mining Software Repositories (MSR 2006), Shanghai, China, pp. 54-57, May 2006*
- Learning patches from method sequence calls
  - Suresh Thummalapenta and Tao Xie. Mining exception-handling rules as sequence association rules. In ICSE '09: Proceedings of the 31st International Conference on Software Engineering, pages 496– 506, Washington, DC, USA, 2009. IEEE Computer Society.
- Obtaining sequence miners:
  - https://illimine.cs.uiuc.edu/
  - Another tool set is at http://himalaya-tools.sourceforge.net/
  - See more tools at https://sites.google.com/site/asergrp/dmse/resources

Application: etc etc etc

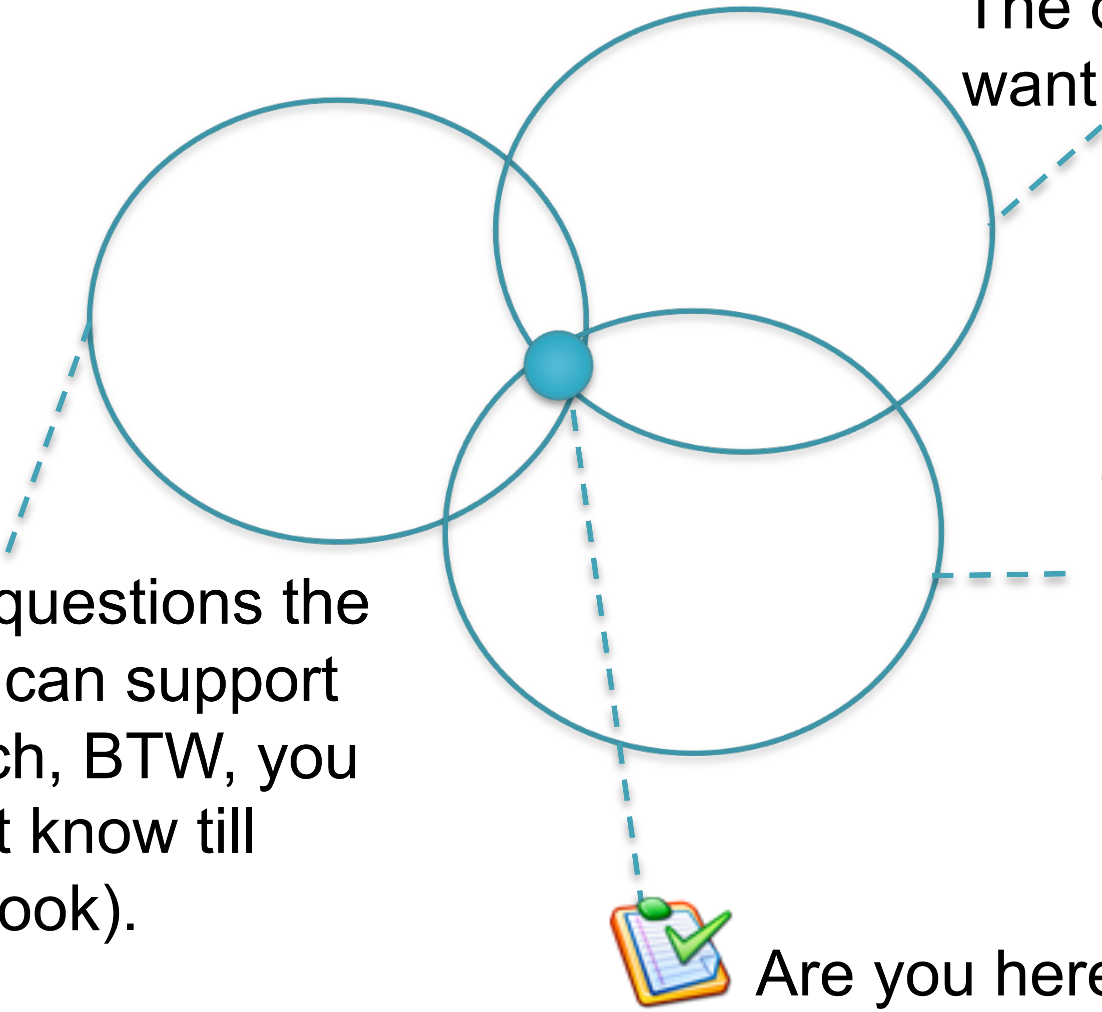
- Data mining + SE a very active area
  - PROMISE conference
  - Mining Software Repository conference
- See also
  - ESEM conference
  - Search-based software engineering
- Hint: to get ahead of the curve…
  - … learn sequence mining
- Welcome to Empirical SE, version 2.0

Empirical SE, Version 2.0

- Open Science movement
  - Open Data
    - Everyone places their data on-line, all the time
  - Open Access publishing
    - Death to subscription-based services
- Shneiderman, B. (2008) "Science 2.0" Science 319(5868):1349-50
  - Science meets web 2.0
  - International team of researchers posting and analyzing data
  - Research at internet speed
- Anda, Markus et al (*) distinguish between
  - **Case studies:** that collect new context variables from project data
  - **Experiments**: that explore case study data
  - Currently, very few case studies generating publicly available data
    - But very many researchers wanting to experiment on that data
    - Perfect setting for data mining

- (*) Bente Anda Audris Mockus and Dag I.K. Sjoberg. Experiences from replicating a case study to investigate reproducibility of software development. In First International Workshop on Replication in Empirical Software Engineering Research, ICSE'09,

Empirical Science 2.0 adjusts its questions to the available data



The lamppost objection

- Where did you lose your keys?
  - Over by the car.
- So why are you looking here?
  - Cause the light is better

"I'm searching for my keys."

- Traditional view
  - Frame the question
  - Then collect the data
  - Then explore data w.r.t. that question
  - Assumes control of data collection

The lamppost objection (more)

- Where did you lose your keys?
  - Over by the car.
- So why are you looking here?
  - Cause the light is better

"I'm searching for my keys."

- Another view
  - Study the SE literature looking for open questions
  - Study the SE data looking for what answers they support
  - When answers intersect with questions, then report
- E.g. Kitchenham TSE 2007: use cross-vs-within company data
- E.g. Shepperd TSE 2002, 2005: stability of ranking of estimation methods

Does it work? Any general conclusions?

Only a small minority of PROMISE papers (11/64) discuss results that repeated in data sets from multiple projects

E.g. Ostrand, Weyuker, Bell PROMISE '08, '09

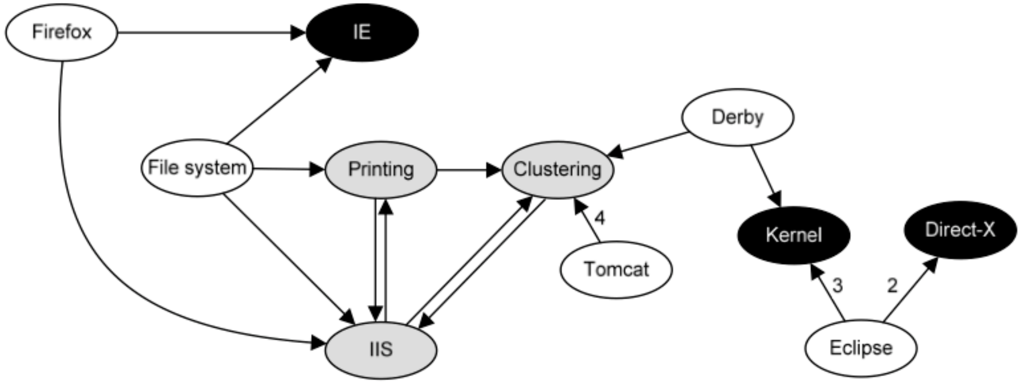
- Same functional form
- Predicts defects for generations of AT&T software

E.g. Turhan, Menzies, Bener PROMISE '08, '09

- 10 projects
  - Learn on 9
  - Apply to the 10th
- Defect models learned from NASA projects work for Turkish white goods software
  - Caveat: need to filter irrelevant training examples. See also
    - •When to Use Data from Other Projects for Effort Estimation Ekrem Kocaguneli, Gregory Gay, Tim Menzies, Ye Yang, Jacky W. Keung , ASE 2010
    - •B. Turhan, T. Menzies, A. Bener, and J. Distefano. On the relative value of cross-company and within- company data for defect prediction. Empirical Software Engineering, 68(2):278–290, 2009

General conclusions: very rare

- The usual conclusion is that we learn that we can learn very little
- FSE'09: Zimmerman et
  - Defect models not generalizable
    - Learn "there", apply "here" only works in 4! of their 600+ experim
  - Opposite to Turhan'09
    - ?add relevancy filter



- ASE'09: Green, Menzies et al.
  - AI search for better software project options
  - Conclusions highly dependent on local business value proposition
- And others
  - TSE '01, '05: Shepperd et al
    - Any conclusion regarding "best" effort estimator varies by data sets, performance criteria, random selection train/test set
  - TSE'06: Menzies, Greenwald:
    - attributes selected by column selection vary wildly across projects

The local lesson hypothesis

- Software construction is a very varied process
- General principles may not hold at local sites
- Need methods for quickly finding those local lessons
- Enter data mining

Q: Why Empirical SE 2.0?
A: Case study results may not generalize

- What is true at one site,
  - May not be true for another
  - E.g. local sites have different goals, different biases, that changes what is "best" for that site
    - II, P. G., Menzies, T., Williams, S., and El-Rawas, O. 2009. Understanding the Value of Software Engineering Technologies. In *Proceedings of the 2009 IEEE/ACM international Conference on Automated Software Engineering(November 16 - 20, 2009)*
  - E.g. general policies perform worse than locally generated policies
    - Menzies, T., Williams, S., Boehm, B., and Hihn, J. 2009. How to avoid drastic software process change (using stochastic stability). In *Proceedings of the 31st international Conference on Software Engineering (May 16 - 24, 2009*
- So we need to audit the conclusions of one case study w.r.t. to data taken from other sites.
- Data mining is one technology that can (at least partially) automate that audit process

Q: Why Empirical SE 2.0?
A: Sharing is a good thing

- WC= Within- company data
  - Locally collected, locally applied
- CC= Cross- company data
  - Collected elsewhere, applied here.
- Filtered CC works nearly as well as WC
  - Turhan, B., Menzies, T., Bener, A. B., and Di Stefano, J. 2009. On the relative value of cross-company and within-company data for defect prediction. *Empirical Softw. Engg. 14, 5 (Oct. 2009), 540-578*
  - Ekrem Kocaguneli, Gregory Gay, Tim Menzies, Ye Yang, Jacky Keung, When to Use Data from Other Projects for Effort Estimation, *IEEE ASE 2010*
- So if ever you are doing new work,
  - and lack local data,
  - you can apply other people's data
- But only if it is available
  - Open data !!!

Q: Why Empirical SE 2.0? A: Like it or not, here it comes

- Everyone has access to this technology
  - In three minutes, on your machine, I can install
    - 200 data miners * pre-processors * meta-learners
    - http://www.cs.waikato.ac.nz/ml/weka/index_downloading.html
- We must explore how communities can responsibly use that technology
- Otherwise, we run the risk of
  - Inexperienced analysts hiding in a corner,
  - Making private conclusions
  - Which they don't share

Q: Why Empirical SE 2.0
A: Changing nature of data

- In the 21st century
  - we can access more data collected by others than we can ever can collect by ourselves.
- In the 20th century,
  - research was focused on case studies where researchers collected special purpose data sets for their particular questions.
- In the 21st century,
  - much research is devoted to experimentation with the data generated by the case studies,
  - possibly investigating hypotheses not originally considered when the data was collected.
  - Data mining is one way to experiment with data.

Q: Why Empirical SE 2.0?
A: Increasing pace of change

- New developments are radically changing SE: open source toolkits, agile development, cloud-based computing, etc.

- 20th century Empirical SE used "big science"
  - Research questions, data collection, analysis took years
  - Big science is too slow to keep up with changes to contemporary SE. e.g.
    - Increasing pace of organization change at NASA was fatal to the "big science" approach of Victor Basili's Software Engineering Laboratory (*)
    - V. Basili, F. McGarry, R. Pajerski, and M. Zelkowitz. Lessons learned from 25 years of process improvement: The rise and fall of the NASA software engineering laboratory. In Proceedings of the 24th International Conference on Software Engineering (ICSE) 2002, Orlando, Florida, 2002.

- Data mining is one response to the open and urgent issue of
  - how to reason faster about SE data.

Q: Why Empirical SE 2.0?
A: Changing nature of SE theories

- 20th century SE: the struggle for the single theory
  - E.g. Boehm's COCOMO effort estimation project
  - E.g. SEI capability maturity model [130];

- 21st century: faster pace = more diversity
  - Less likely that there exists a single over-arching grand theory of SE
- Recent reports [1,2,3,4,5] say that while such generality may elude us, we can still find the special lessons that work best on the local projects

1. Rombach A. Endres, H.D. A Handbook of Software and Systems Engineering: Empirical Observa- tions, Laws and Theories. Addison Wesley, 2003.
2. B. Kitchenham D. Budgen, P. Brereton. Is evidence based software engineering mature enough for practice & policy? In 33rd Annual IEEE Software Engineering Workshop 2009 (SEW-33), Skvde, Sweden, 2009.
3. B.A. Kitchenham, E. Mendes, and G. H. Travassos. Cross- vs. within-company cost estimation studies: A systematic review. IEEE Transactions on Software Engineering, pages 316–329, May 2007.
4. Tim Menzies and Forrest Shull. The quest for convincing evidence. In A. Oram and G. Wilson, editors, Making Software: What Really Works, and Why We Believe It. O'Reilly, 2010.
5. H. Gall E. Giger T. Zimmermann, N. Nagappan and B. Murphy. Cross-project defect prediction. In ESEC/FSE'09, August 2009.

- Data mining is one way to rapidly find and verify the special practices that best work on the local projects.

Q: Why Empirical SE 2.0?
A: Changing nature of data analysis

- A contemporary empirical SE paper might explore gigabytes of core dumps looking for the method calls that lead to a crash.
- Faced with such large and complex data, analysis methods are becoming more intricate; e.g.
  - Model trees for multi-model data
  - Latent Dirichlet allocation (LDA) for document clustering
  - Mining sequences to learn exception handling rules
- It is now possible to find <u>new</u> insights in <u>old</u> data, just by applying a <u>new</u> analysis method.
  - E.g. see later, the "W" tool

Why Data Mining for SE?

- Natural tool to help a community:
  - racing to keep up with the pace of change in SE;
  - while finding and verifying the special theories that work best on local projects …
  - … from a new kind data sources …
  - … using a large menagerie of new data analysis tools.

Coming next…

- Enough generalities
- Details of using a data mining tool suite
  - The “WEKA”

DATA MINING TOOLS (GUIDED TOUR OF "WEKA")

Road map

1. Data mining & SE (overview)
2. Data mining tools (guided tour of "WEKA")
3. Data "carving" (core operators of DM)

WEKA

- Machine learning/data mining software written in Java
  - Used for research, education, and applications
  - Complements Data Mining: Practical Machine Learning Tools and Techniques (Second Edition) Ian H. Witten, Eibe Frank, Morgan Kaufmann June 2005 525 pages ISBN 0-12-088407-0
- 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

Access

- 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

Data Files

```
@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
...
```

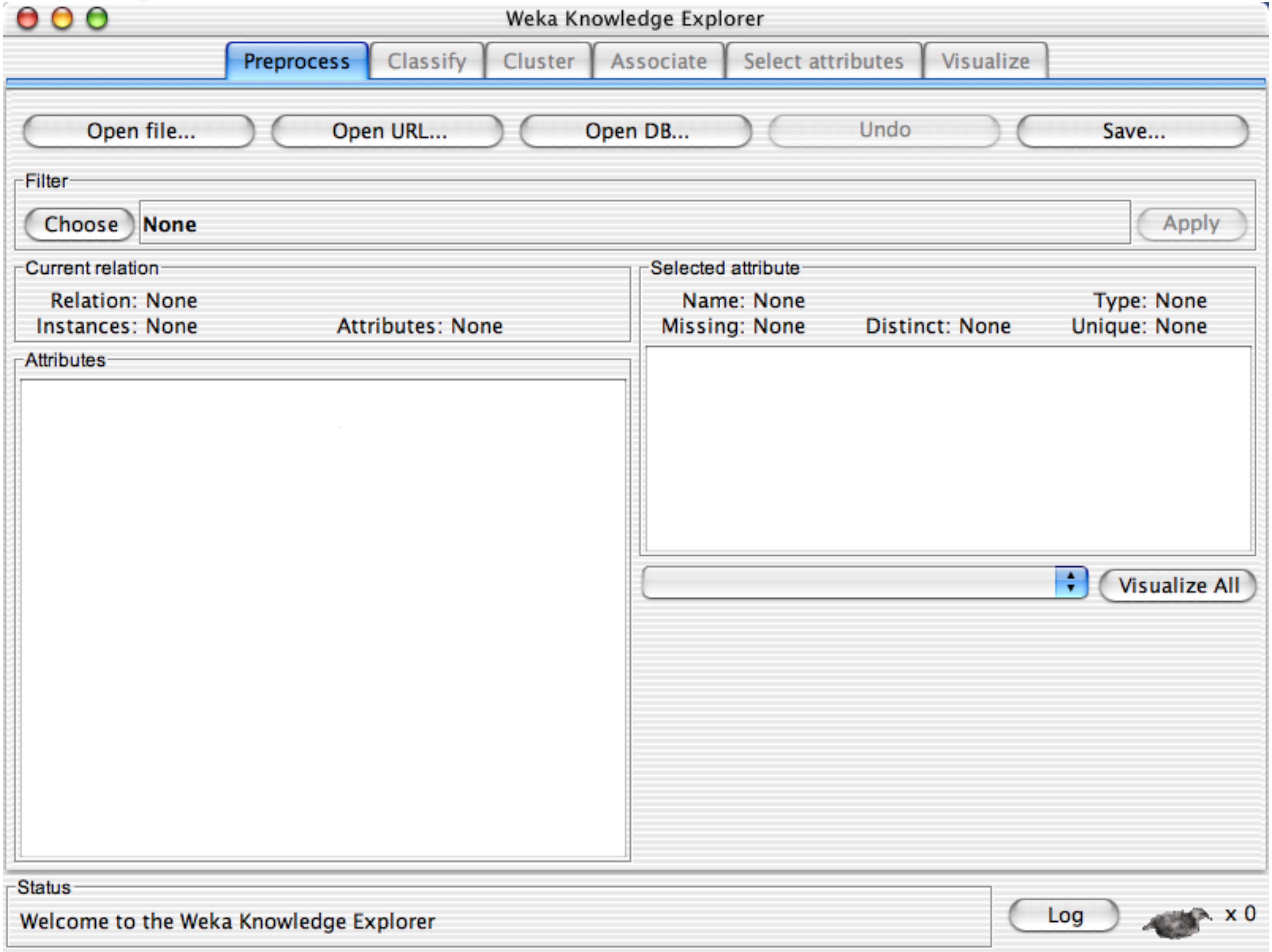
numeric attribute

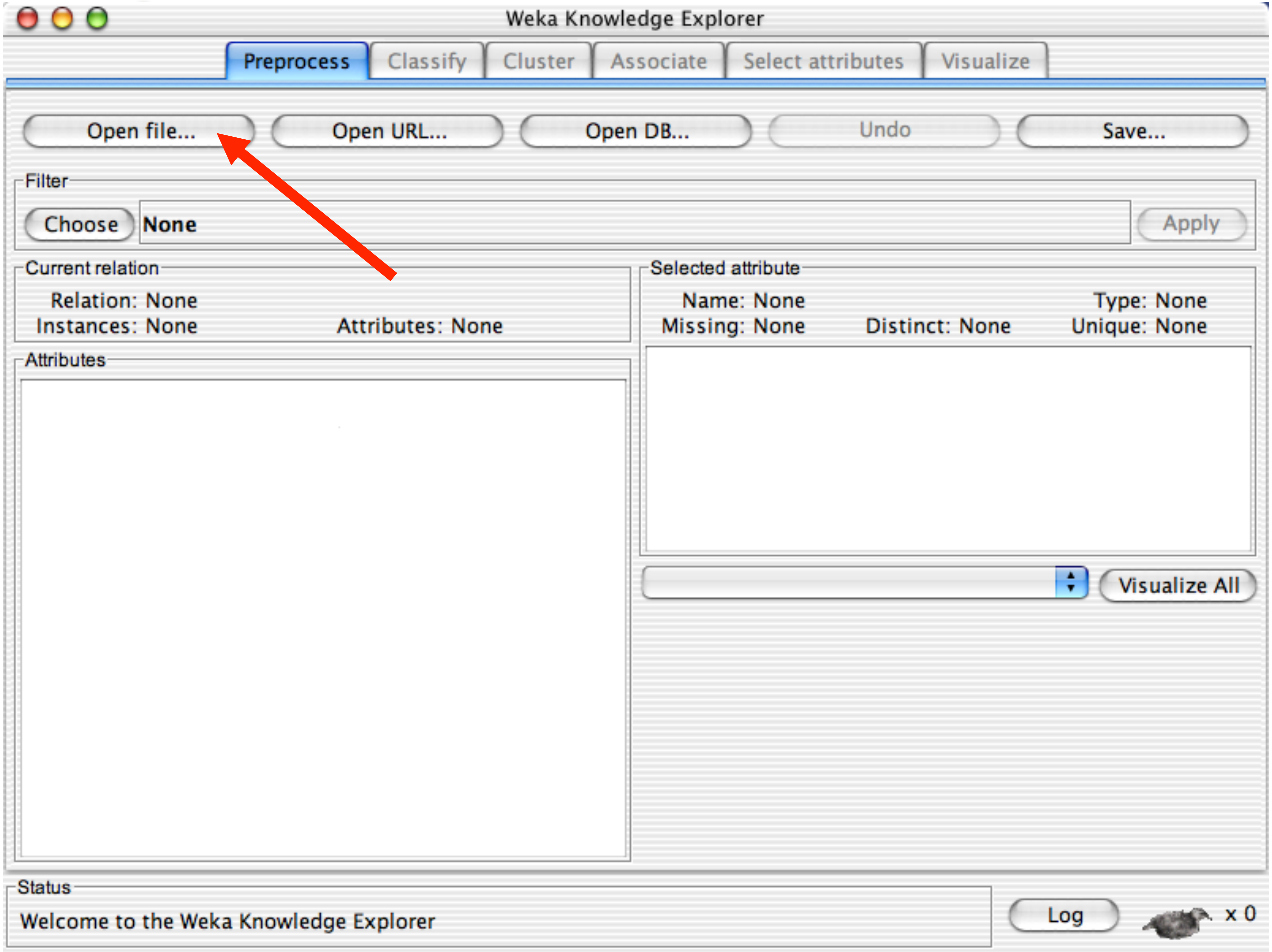
nominal attribute

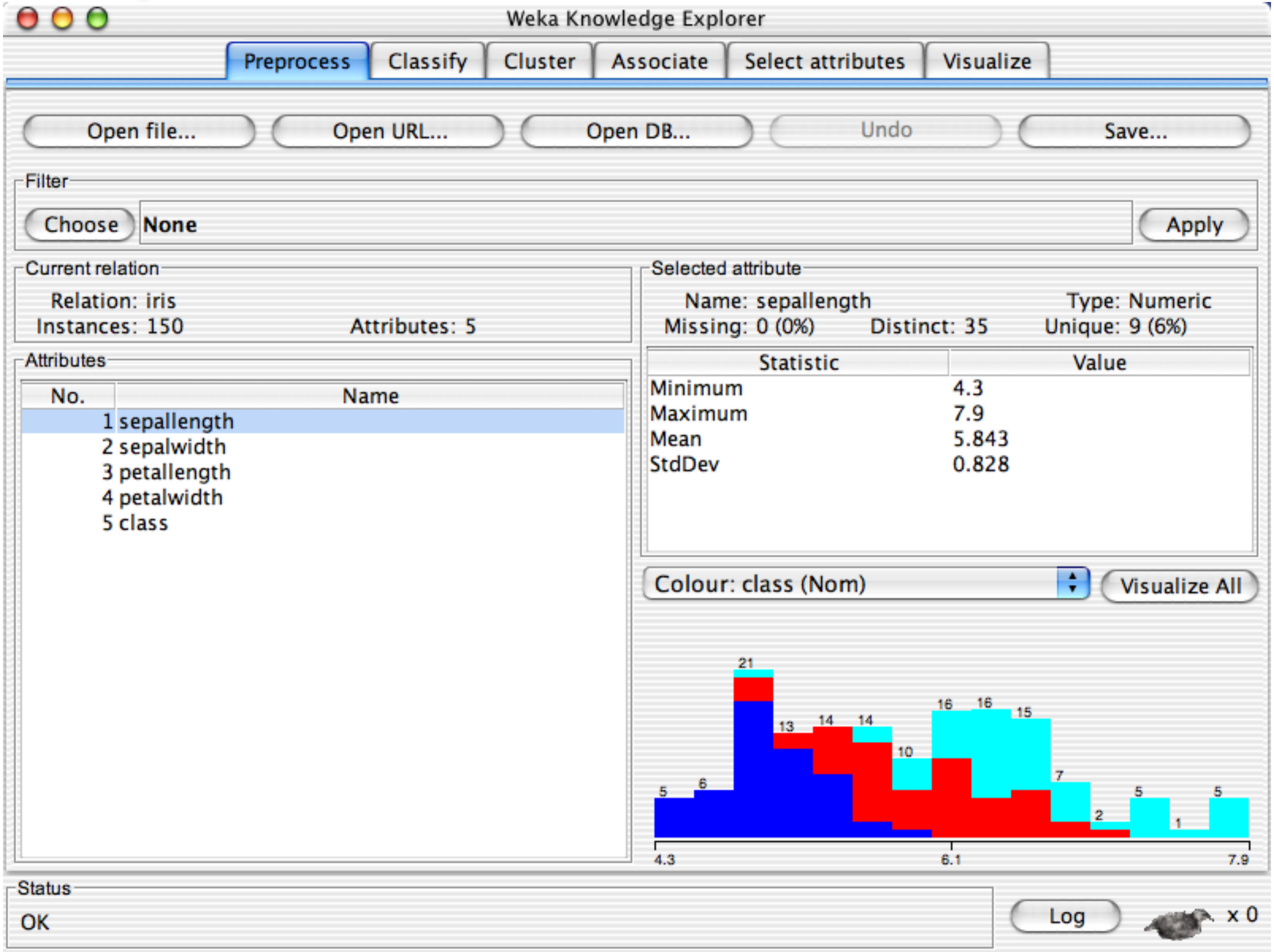
Flat file in ARFF format

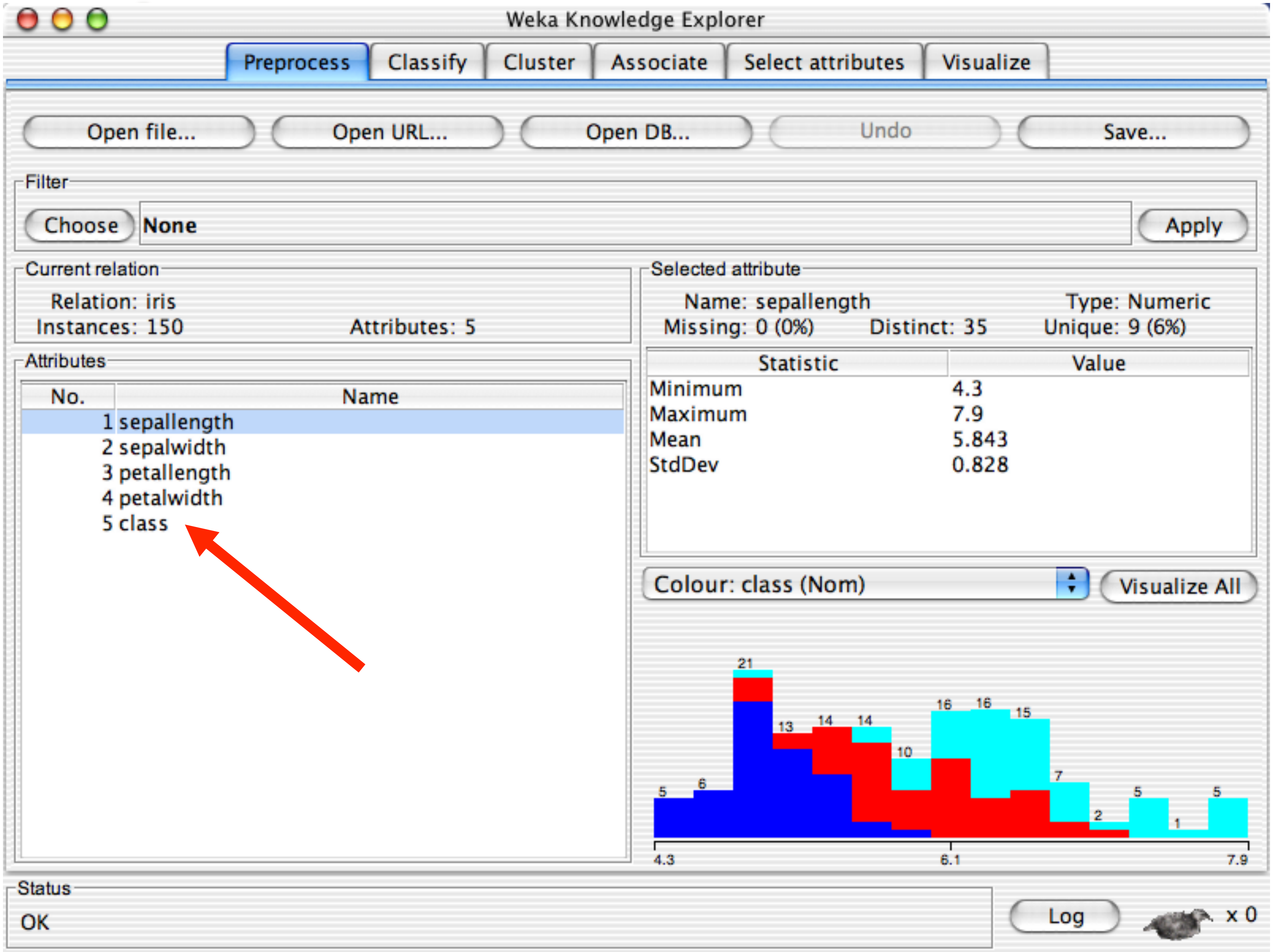
Explorer: pre-processing

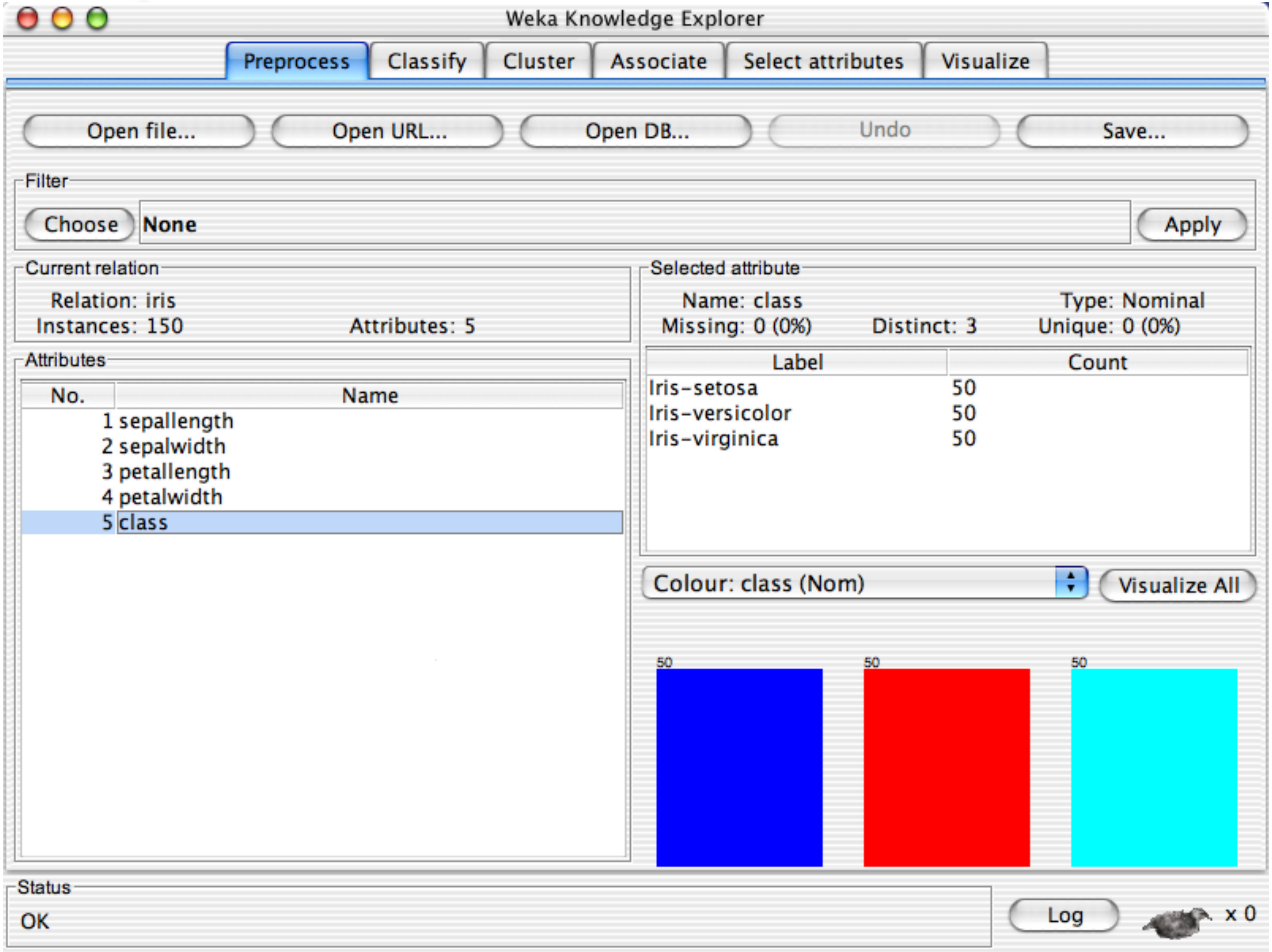
- Source
  - Data can be imported from a file in various formats: ARFF, CSV, C4.5, binary
  - Data can also be read from a URL or from an SQL database (using JDBC)
- Pre-processing tools
  - Called "filters"
  - Discretization, normalization, resampling, attribute selection, transforming and combining attributes, …

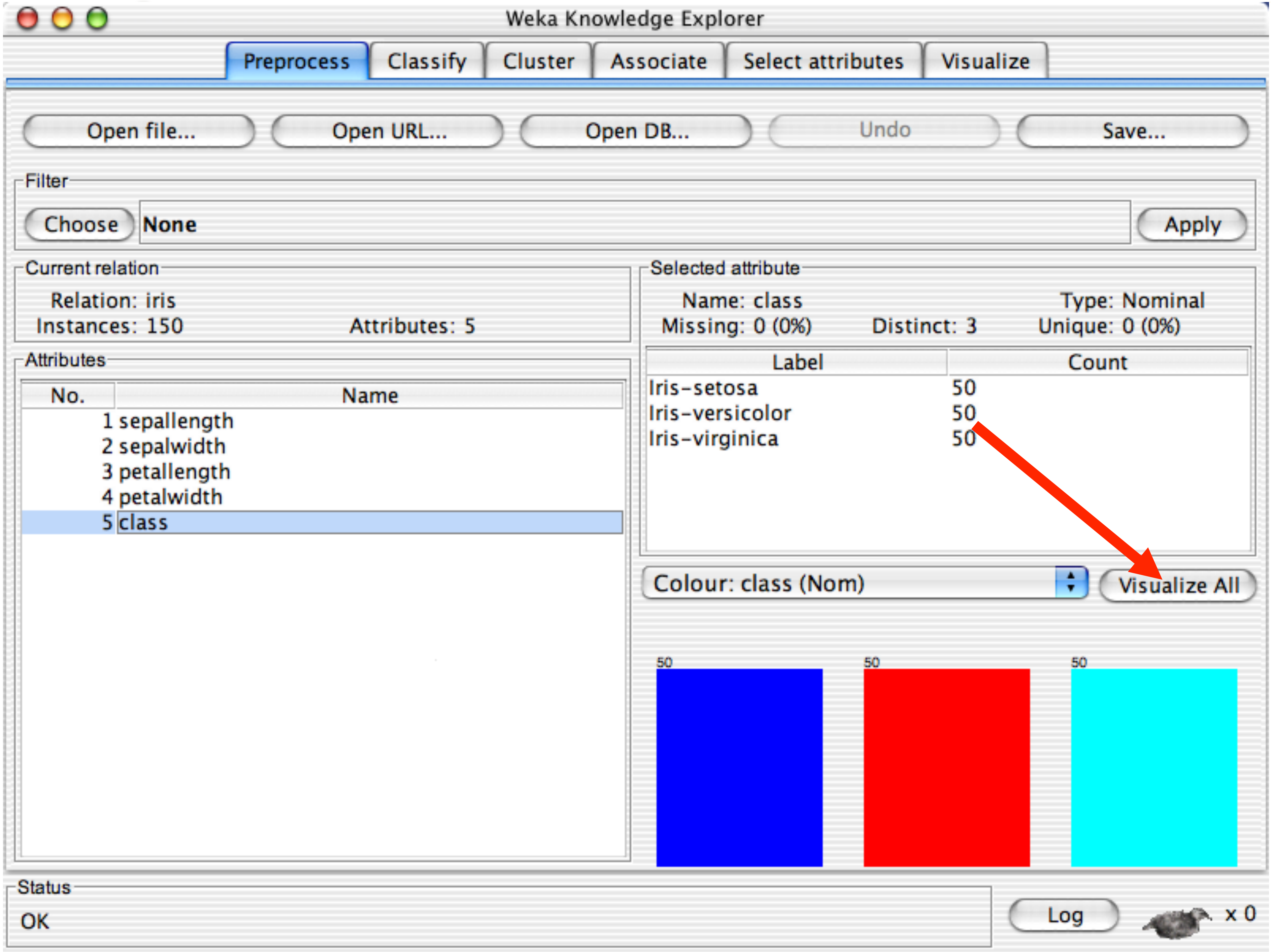

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
None
Apply
Current relation
Relation: None
Instances: None
Attributes: None
Attributes
Selected attribute
Name: None
Type: None
Missing: None
Distinct: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

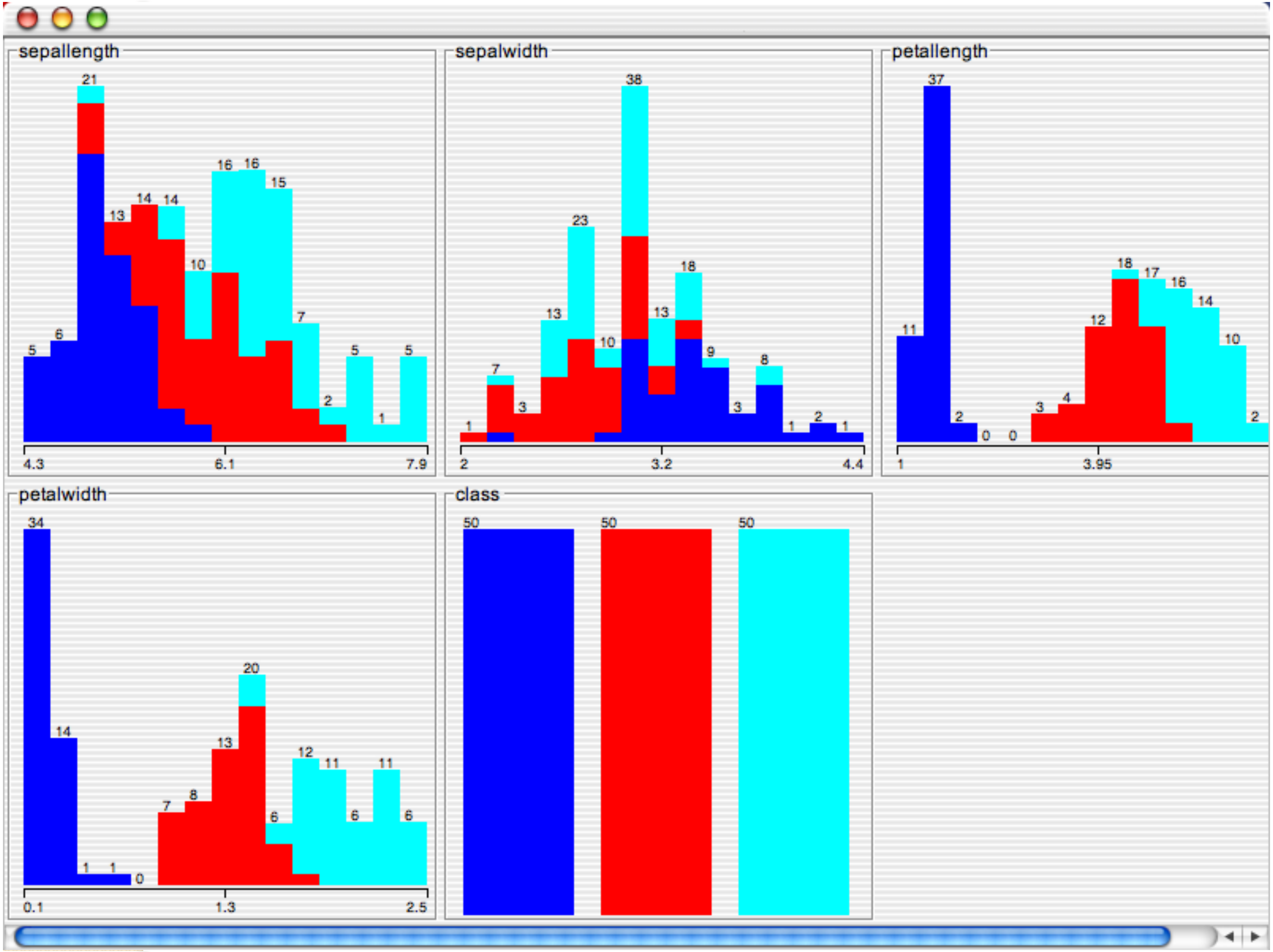

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
None
Apply
Current relation
Relation: None
Instances: None
Attributes: None
Attributes
Selected attribute
Name: None
Type: None
Missing: None
Distinct: None
Unique: None
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
21
5
6
13
14
14
10
16
16
15
7
2
5
1
5
4.3
6.1
7.9
Status
OK
Log
x 0

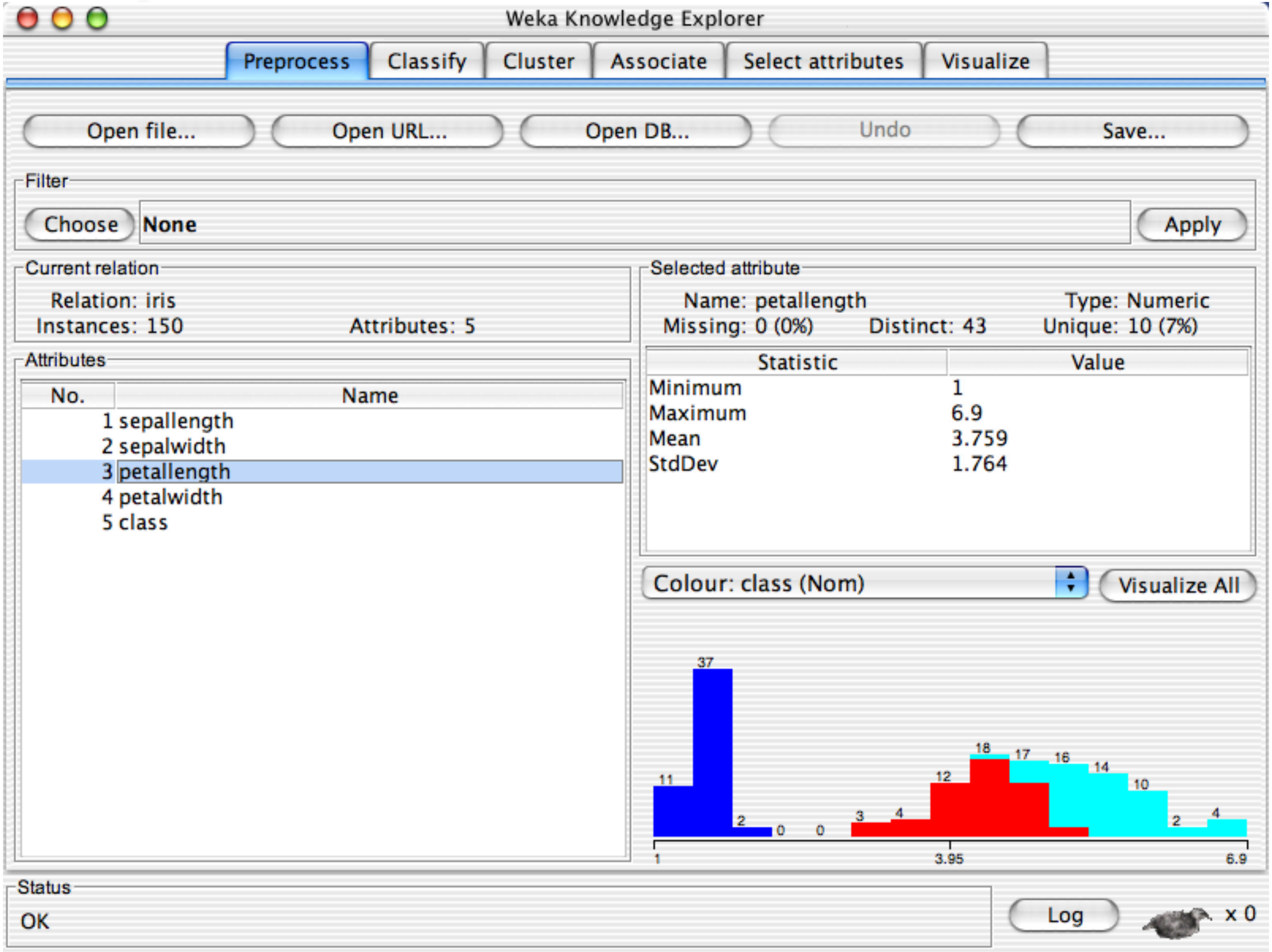

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
None
Apply
Current relation
Relation: iris
Instances: 150
Attributes: 5
Attributes
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 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
5
6
21
13
14
14
10
16
16
15
7
2
5
1
5
4.3
6.1
7.9
Status
OK
Log
x 0

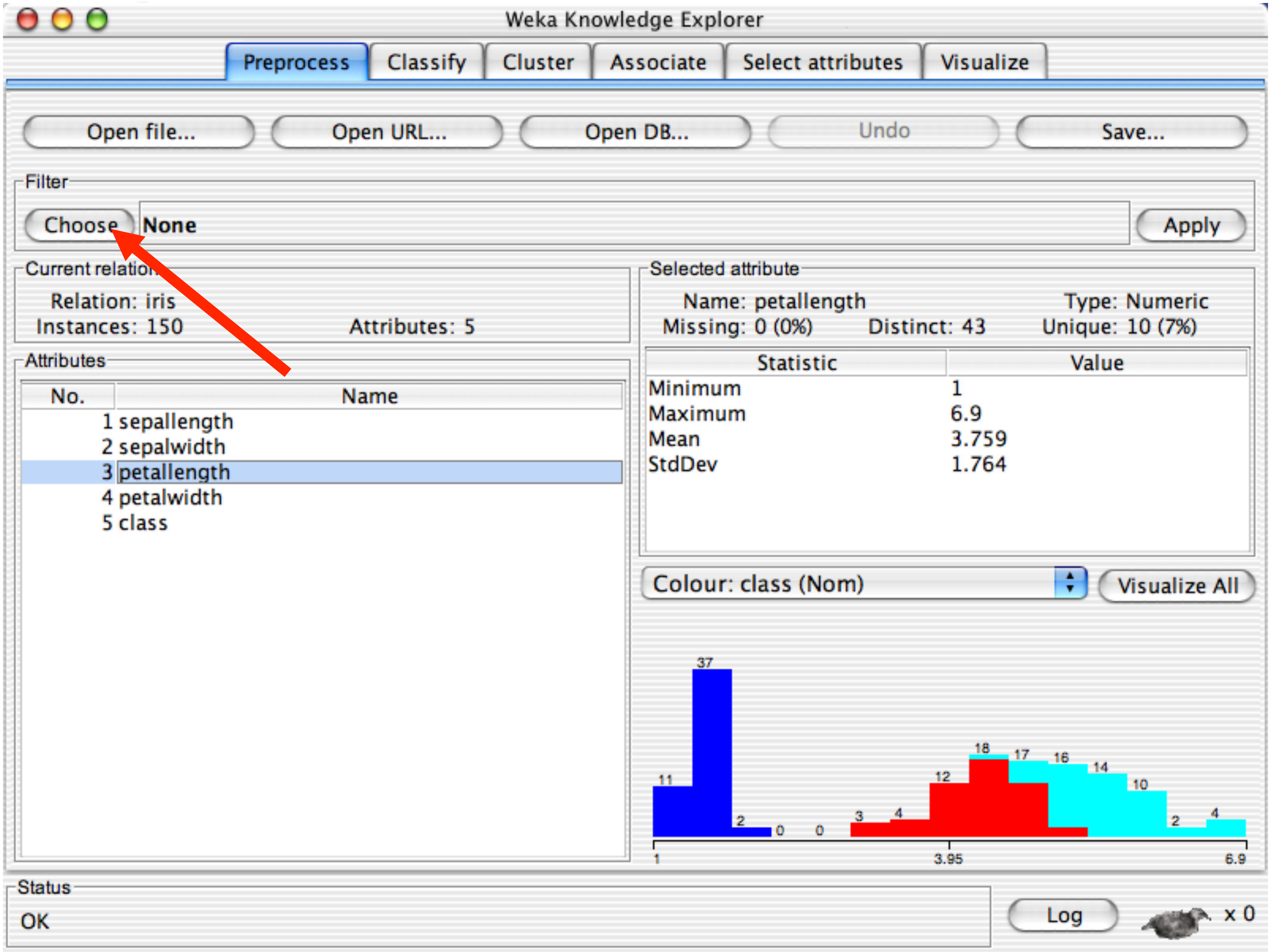

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
None
Apply
Current relation
Relation: iris
Instances: 150
Attributes: 5
Attributes
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 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

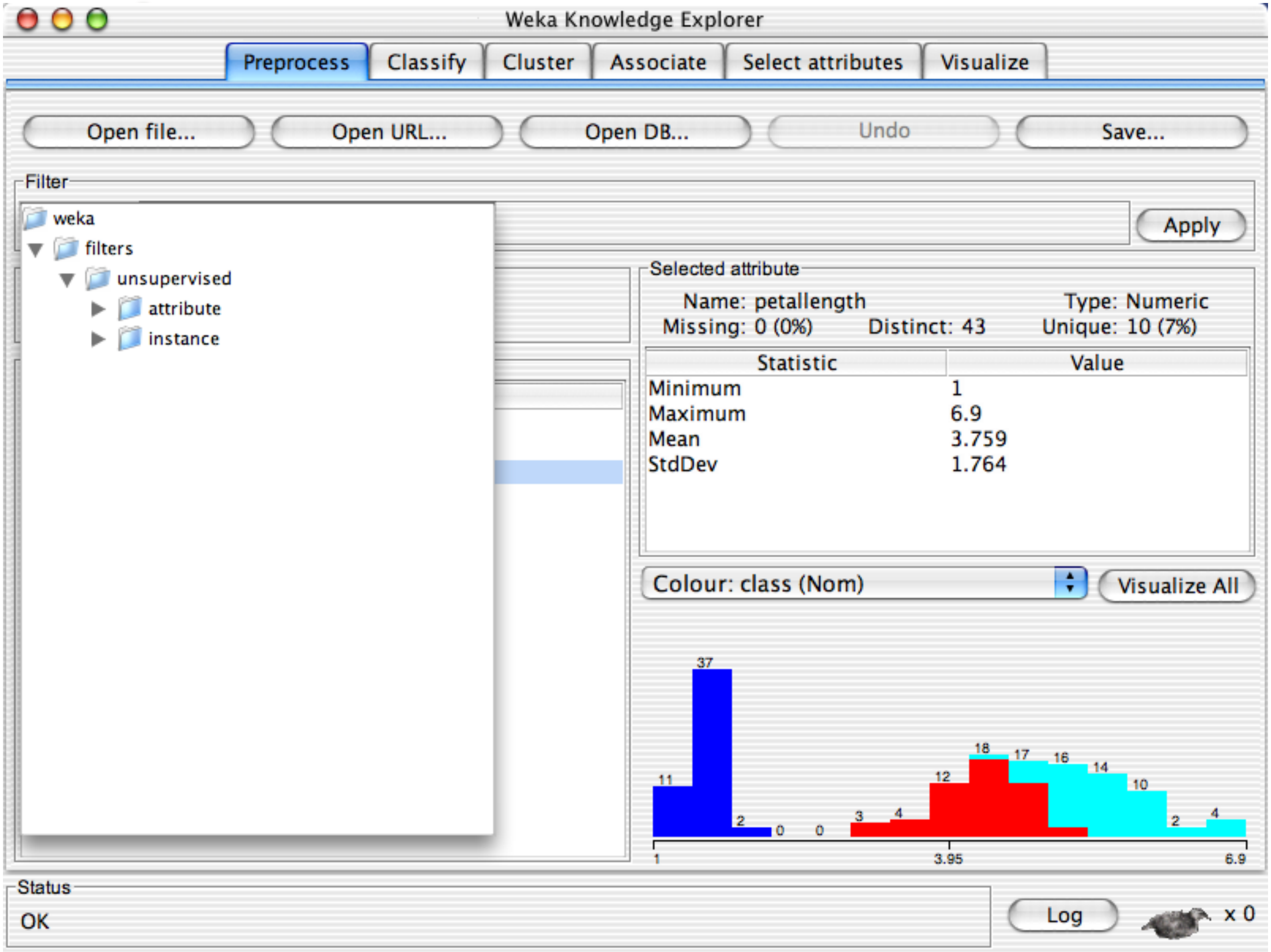

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
None
Apply
Current relation
Relation: iris
Instances: 150
Attributes: 5
Attributes
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 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

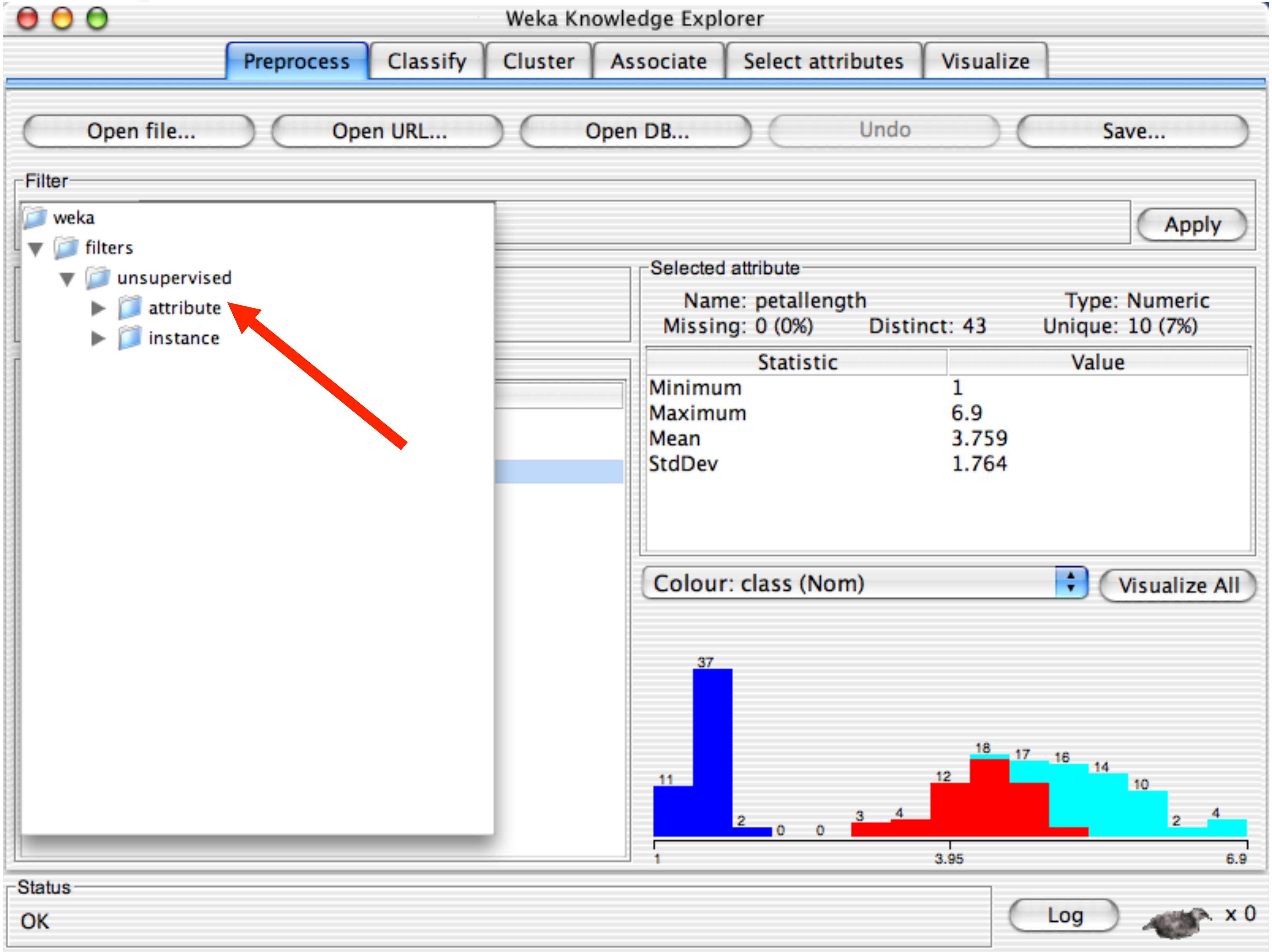


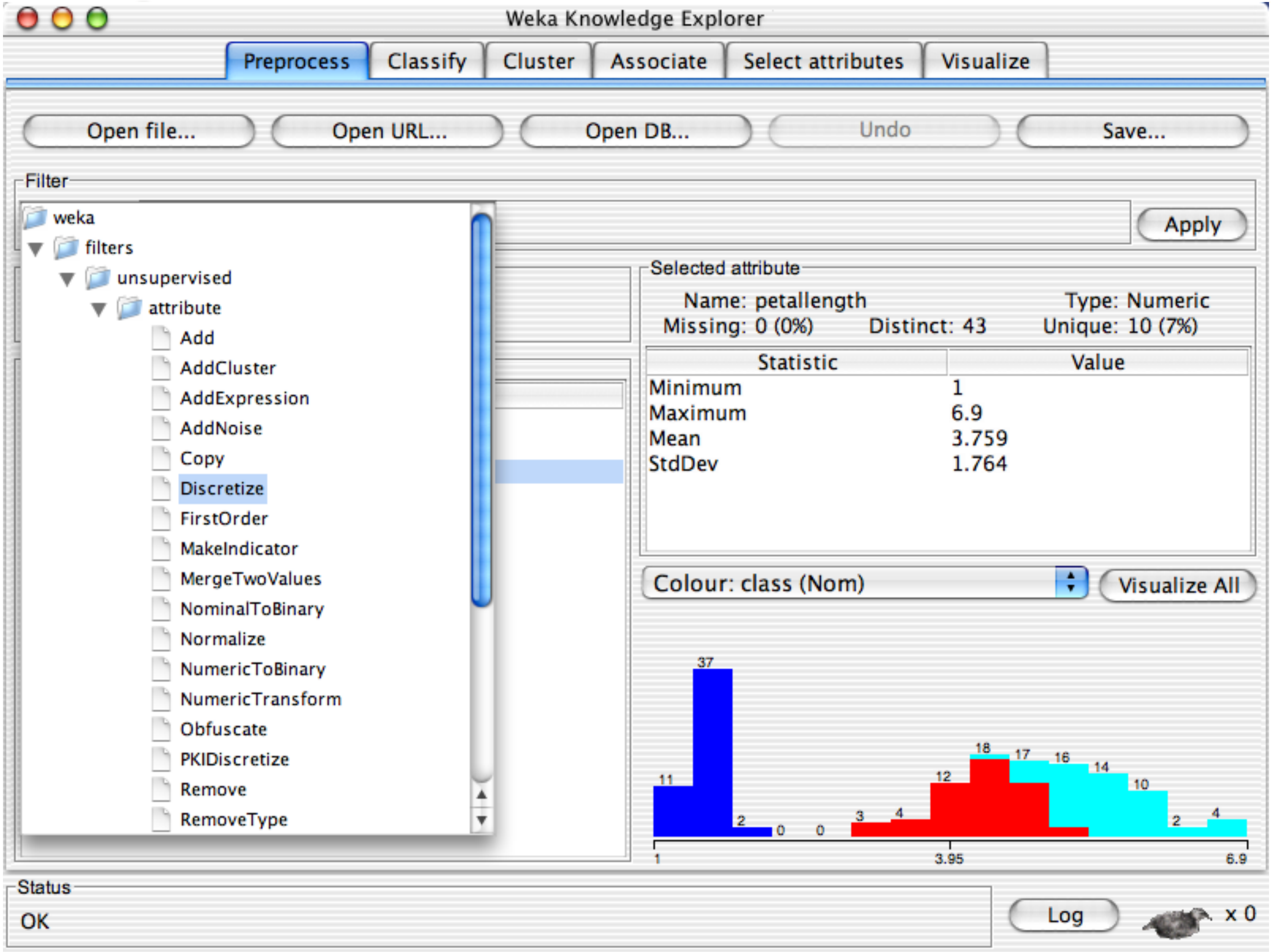
sepallength
21
16
16
15
14
14
13
10
7
6
5
5
5
2
1
4.3
6.1
7.9
sepalwidth
38
23
18
13
13
10
9
8
7
3
3
2
1
1
1
2
3.2
4.4
petallength
37
18
17
16
14
12
11
10
4
3
2
2
0
0
1
3.95
petalwidth
34
20
14
13
12
11
11
8
7
6
6
6
1
1
0
0.1
1.3
2.5
class
50
50
50

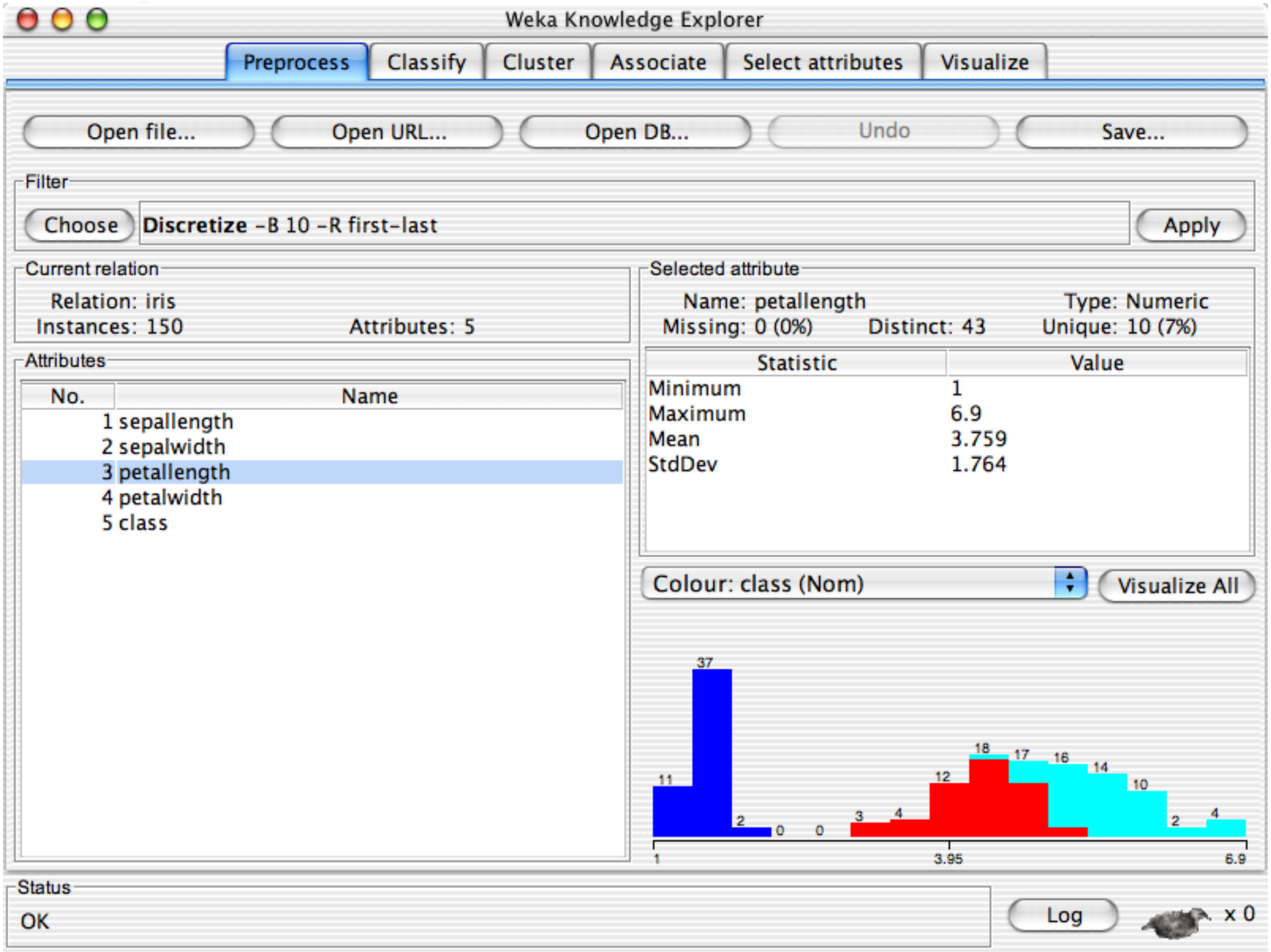

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
None
Apply
Current relation
Relation: iris
Instances: 150
Attributes: 5
Attributes
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Selected attribute
Name: petallength
Type: Numeric
Missing: 0 (0%)
Distinct: 43
Unique: 10 (7%)
Statistic
Value
Minimum 1
Maximum 6.9
Mean 3.759
StdDev 1.764
Colour: class (Nom)
Visualize All
11
37
2
0
0
3
4
12
18
17
16
14
10
2
4
1
3.95
6.9
Status
OK
Log
x 0

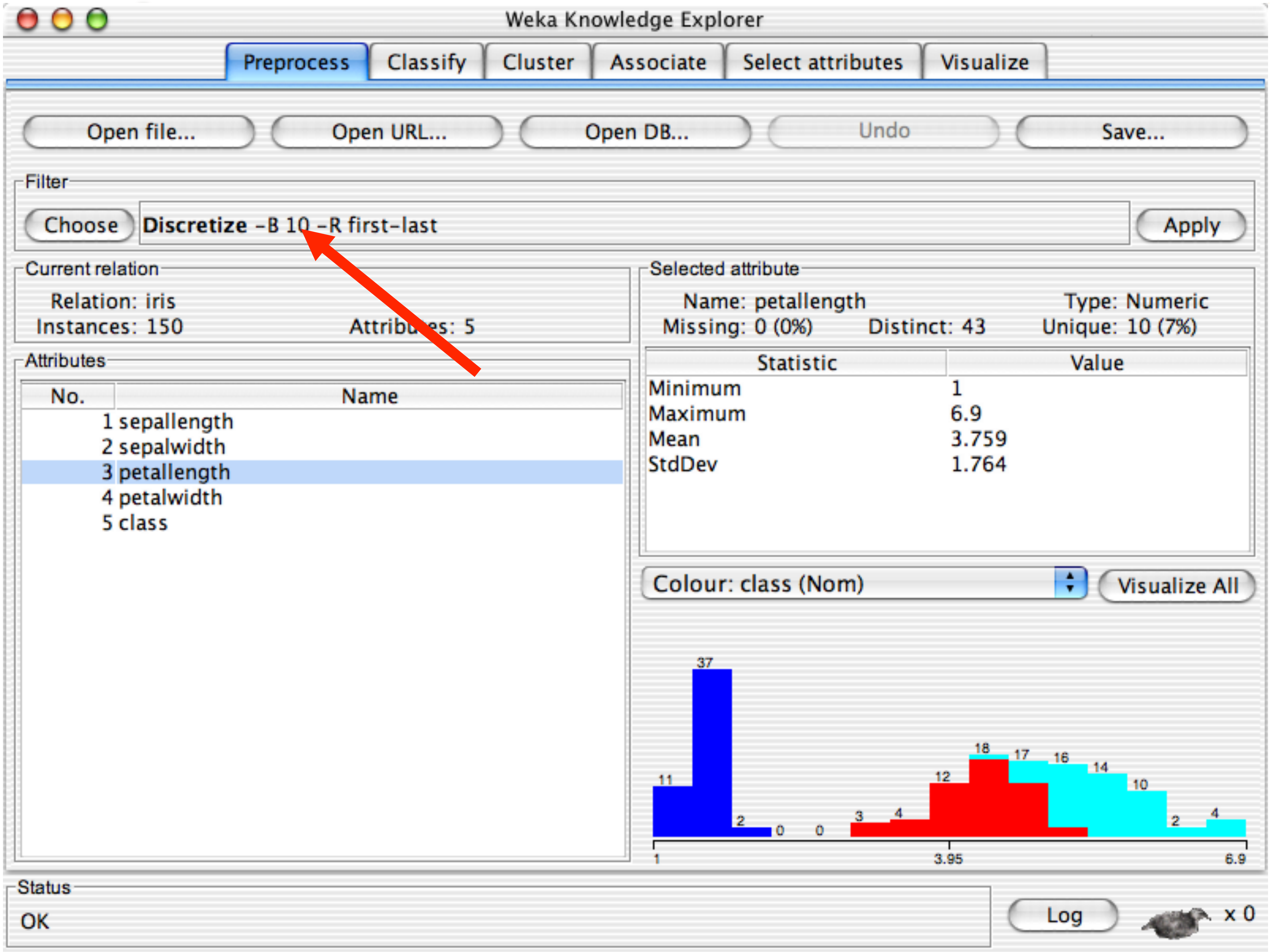

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
None
Apply
Current relation
Relation: iris
Instances: 150
Attributes: 5
Attributes
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Selected attribute
Name: petallength
Type: Numeric
Missing: 0 (0%)
Distinct: 43
Unique: 10 (7%)
Statistic
Value
Minimum 1
Maximum 6.9
Mean 3.759
StdDev 1.764
Colour: class (Nom)
Visualize All
Status
OK
Log
x 0

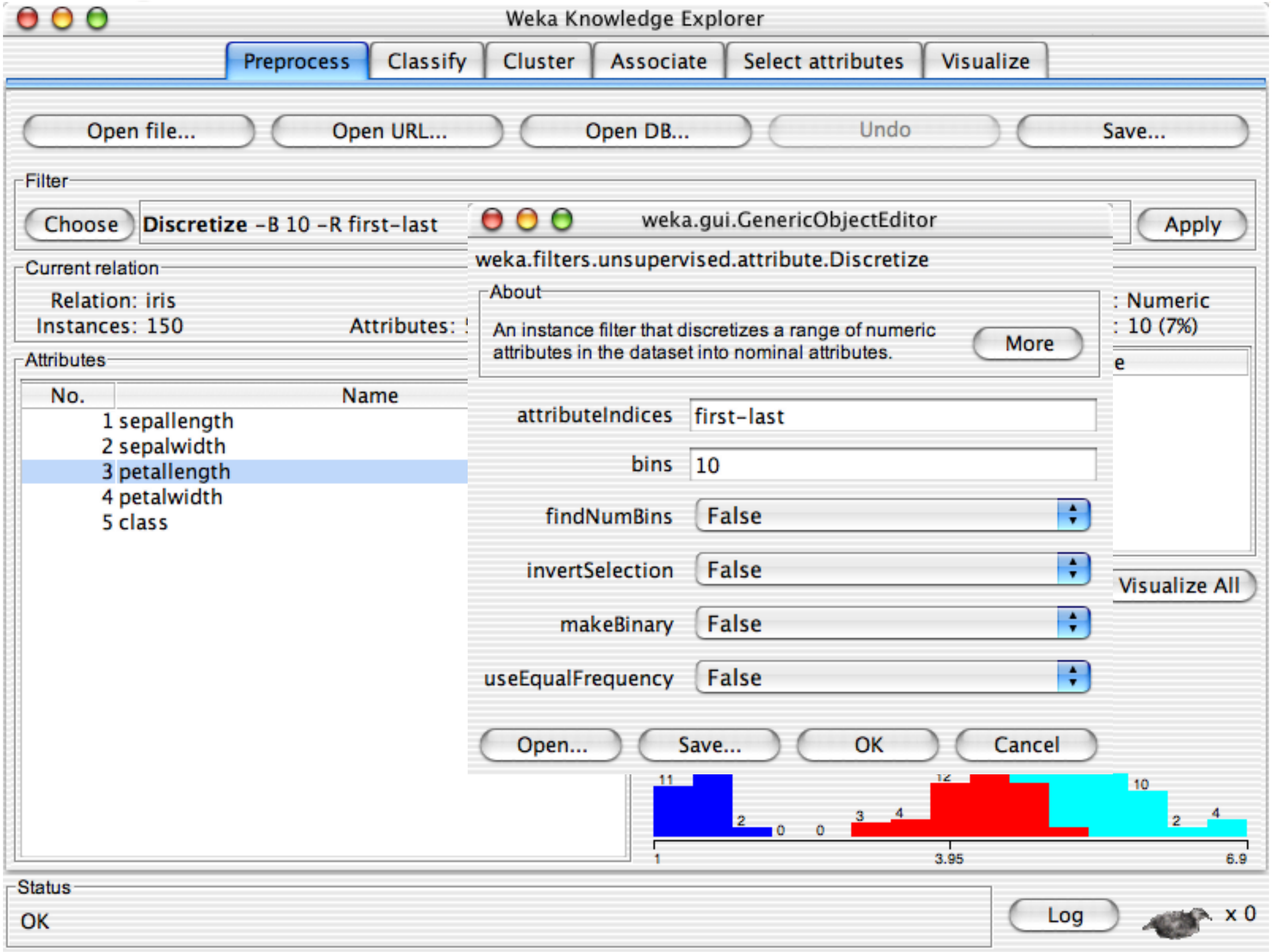

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
weka
filters
unsupervised
attribute
instance
Apply
Selected attribute
Name: petallength
Type: Numeric
Missing: 0 (0%)
Distinct: 43
Unique: 10 (7%)
Statistic
Value
Minimum 1
Maximum 6.9
Mean 3.759
StdDev 1.764
Colour: class (Nom)
Visualize All
11
37
2
0
0
3
4
12
18
17
16
14
10
2
4
1
3.95
6.9
Status
OK
Log
x 0

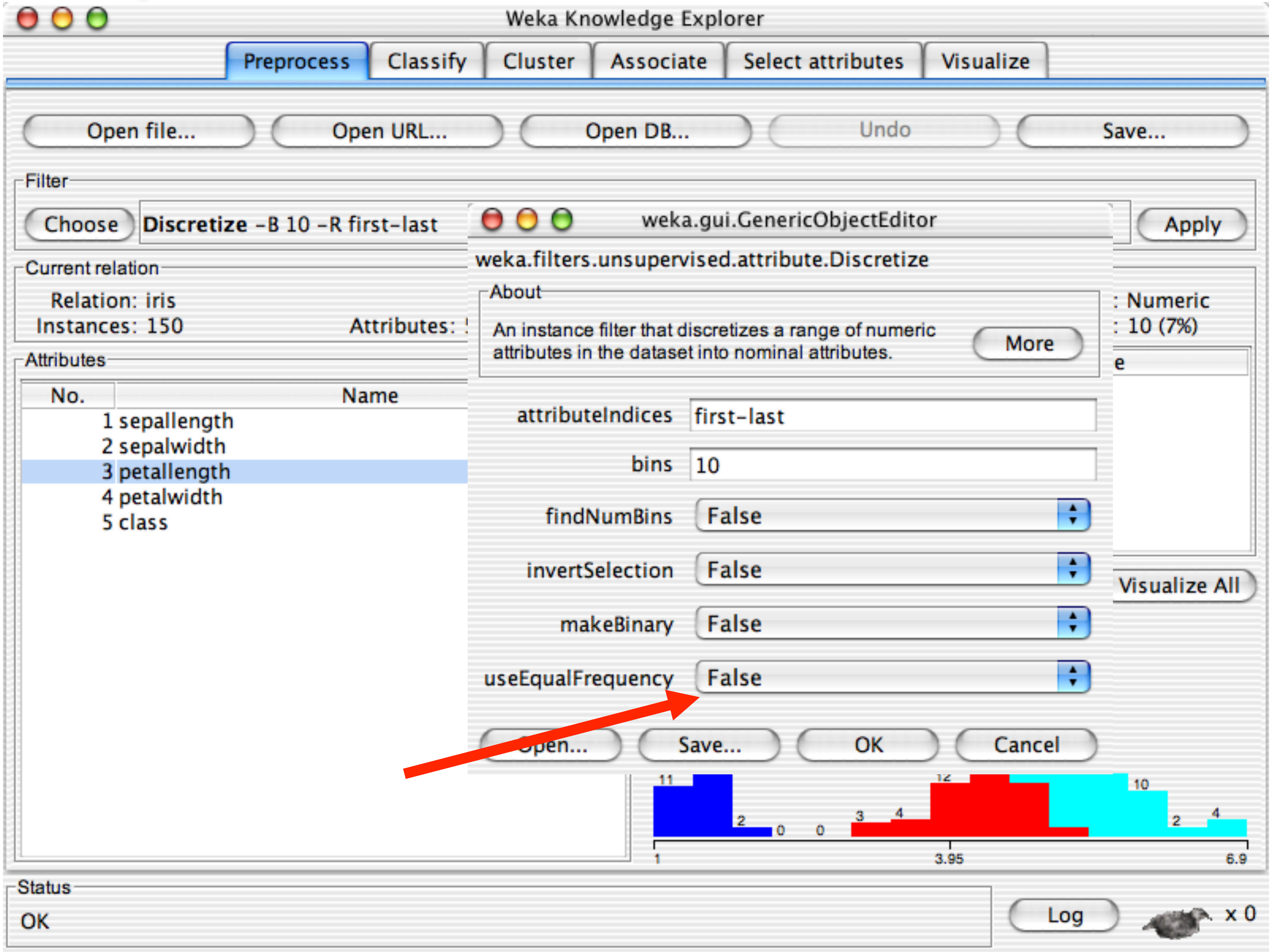

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
weka
filters
unsupervised
attribute
instance
Apply
Selected attribute
Name: petallength
Type: Numeric
Missing: 0 (0%)
Distinct: 43
Unique: 10 (7%)
Statistic
Value
Minimum 1
Maximum 6.9
Mean 3.759
StdDev 1.764
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
Apply
weka
filters
unsupervised
attribute
Add
AddCluster
AddExpression
AddNoise
Copy
Discretize
FirstOrder
MakeIndicator
MergeTwoValues
NominalToBinary
Normalize
NumericToBinary
NumericTransform
Obfuscate
PKIDiscretize
Remove
RemoveType
Selected attribute
Name: petallength
Type: Numeric
Missing: 0 (0%)
Distinct: 43
Unique: 10 (7%)
Statistic
Value
Minimum
1
Maximum
6.9
Mean
3.759
StdDev
1.764
Colour: class (Nom)
Visualize All
Status
OK
Log
x 0

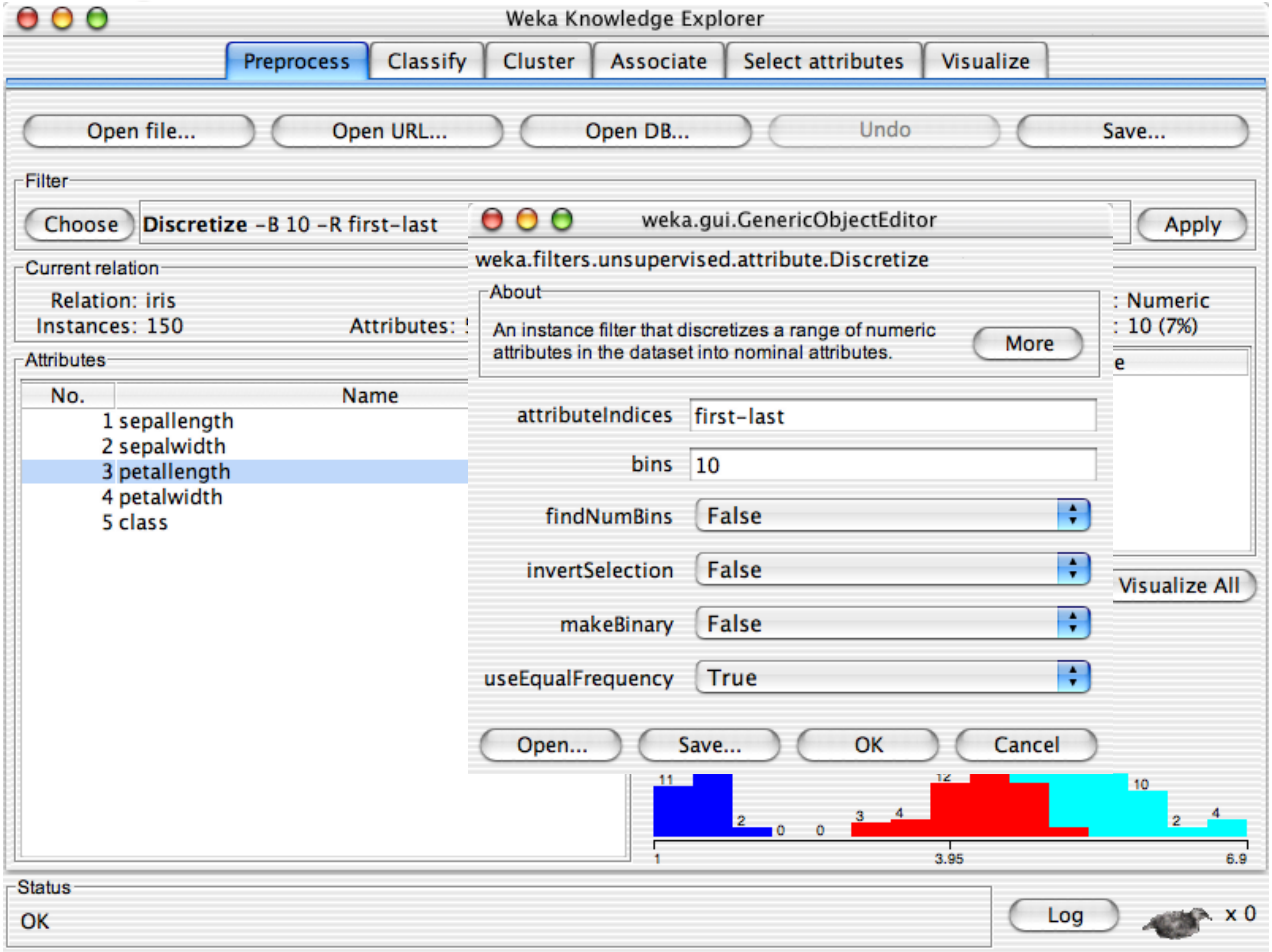

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
Discretize -B 10 -R first-last
Apply
Current relation
Relation: iris
Instances: 150
Attributes: 5
Attributes
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Selected attribute
Name: petallength
Type: Numeric
Missing: 0 (0%)
Distinct: 43
Unique: 10 (7%)
Statistic
Value
Minimum
1
Maximum
6.9
Mean
3.759
StdDev
1.764
Colour: class (Nom)
Visualize All
37
11
2
0
0
3
4
12
18
17
16
14
10
2
4
1
3.95
6.9
Status
OK
Log
x 0

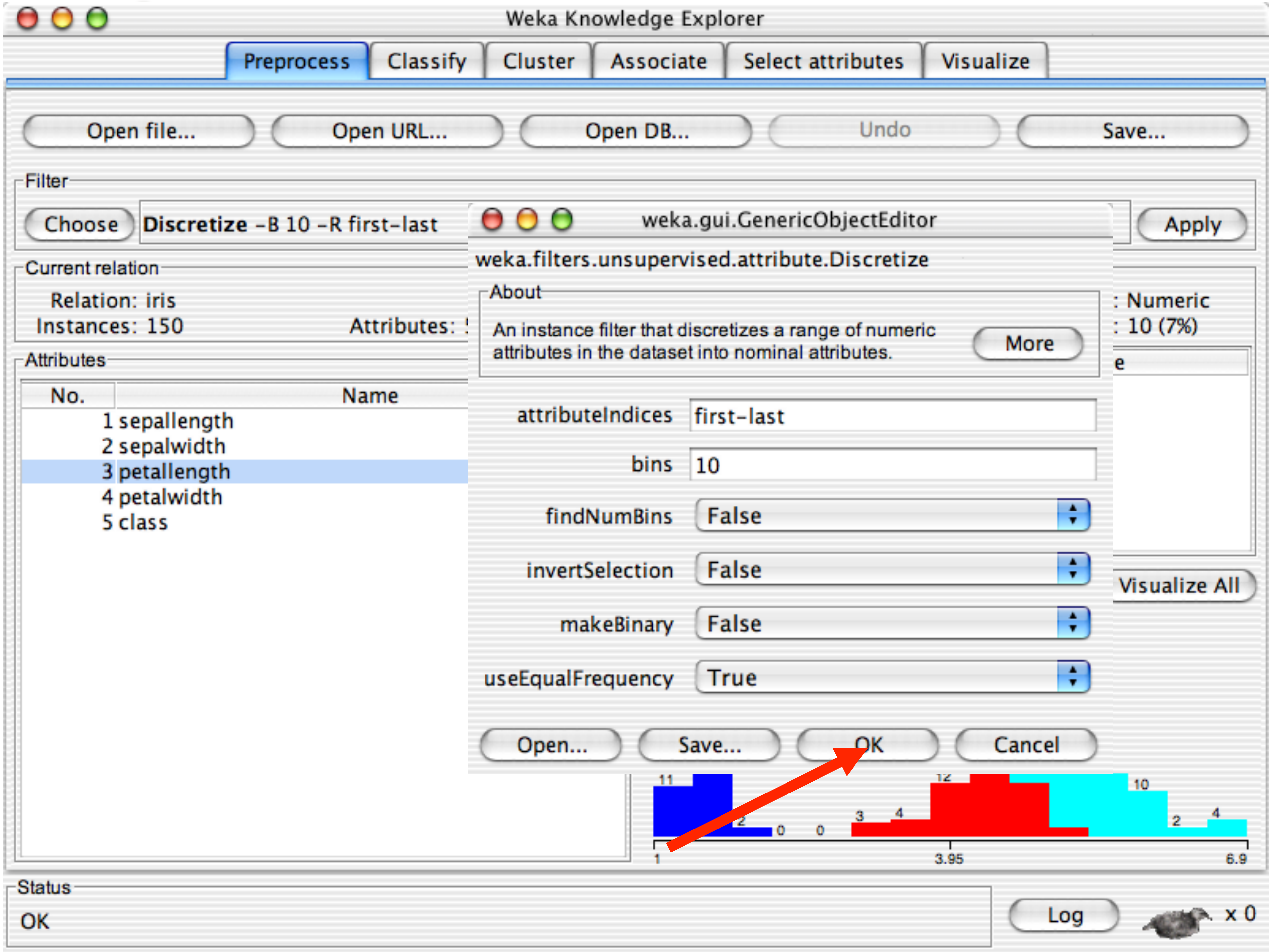

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
Discretize -B 10 -R first-last
Apply
Current relation
Relation: iris
Instances: 150
Attributes: 5
Attributes
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Selected attribute
Name: petallength
Type: Numeric
Missing: 0 (0%)
Distinct: 43
Unique: 10 (7%)
Statistic
Value
Minimum 1
Maximum 6.9
Mean 3.759
StdDev 1.764
Colour: class (Nom)
Visualize All
Status
OK
Log
x 0

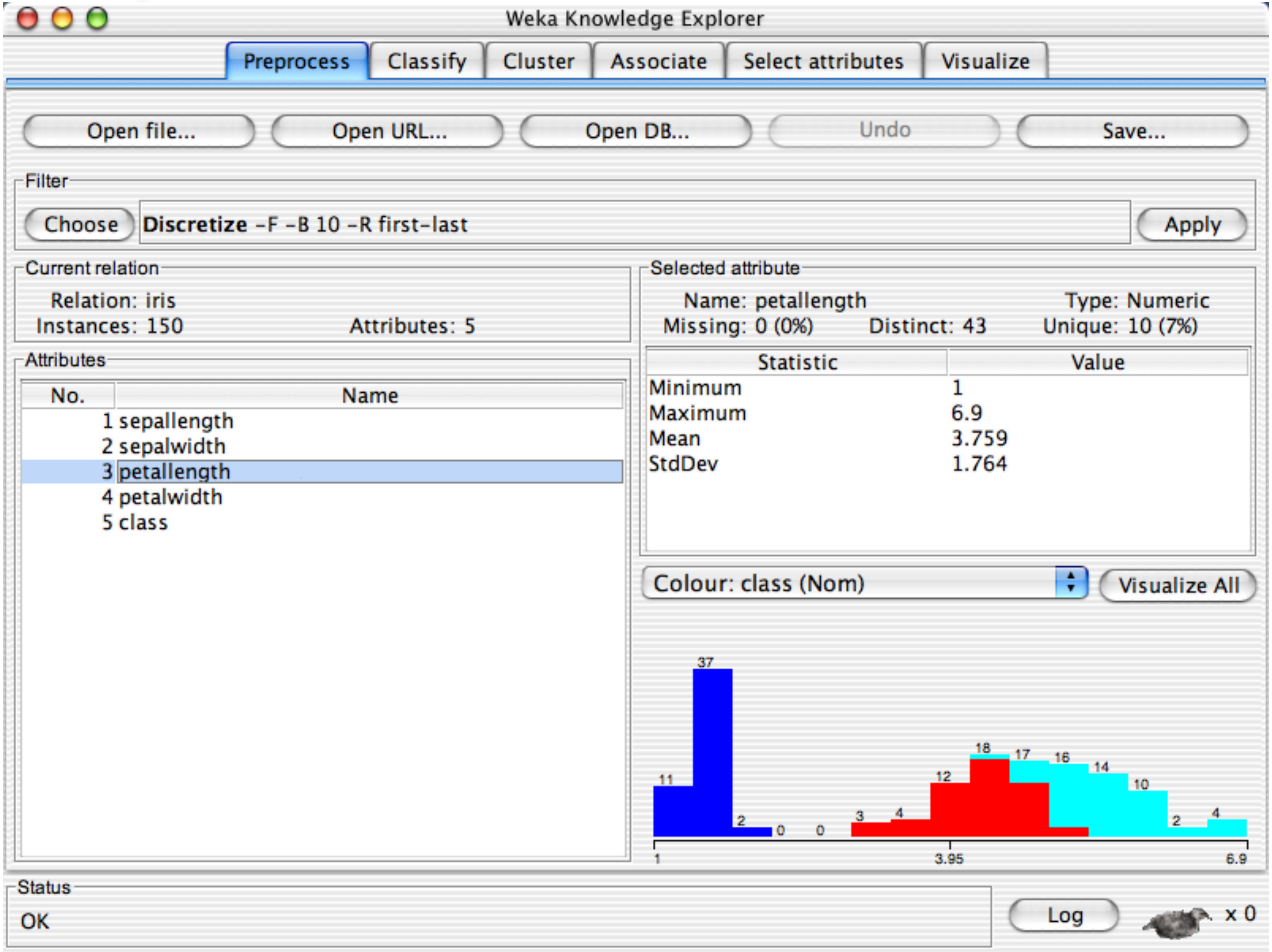

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
Discretize -B 10 -R first-last
Apply
Current relation
Relation: iris
Instances: 150
Attributes:
: Numeric
: 10 (7%)
Attributes
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
weka.gui.GenericObjectEditor
weka.filters.unsupervised.attribute.Discretize
About
An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.
More
attributeIndices
first-last
bins
10
findNumBins
False
invertSelection
False
makeBinary
False
useEqualFrequency
False
Open...
Save...
OK
Cancel
Visualize All
1
3.95
6.9
Status
OK
Log
x 0

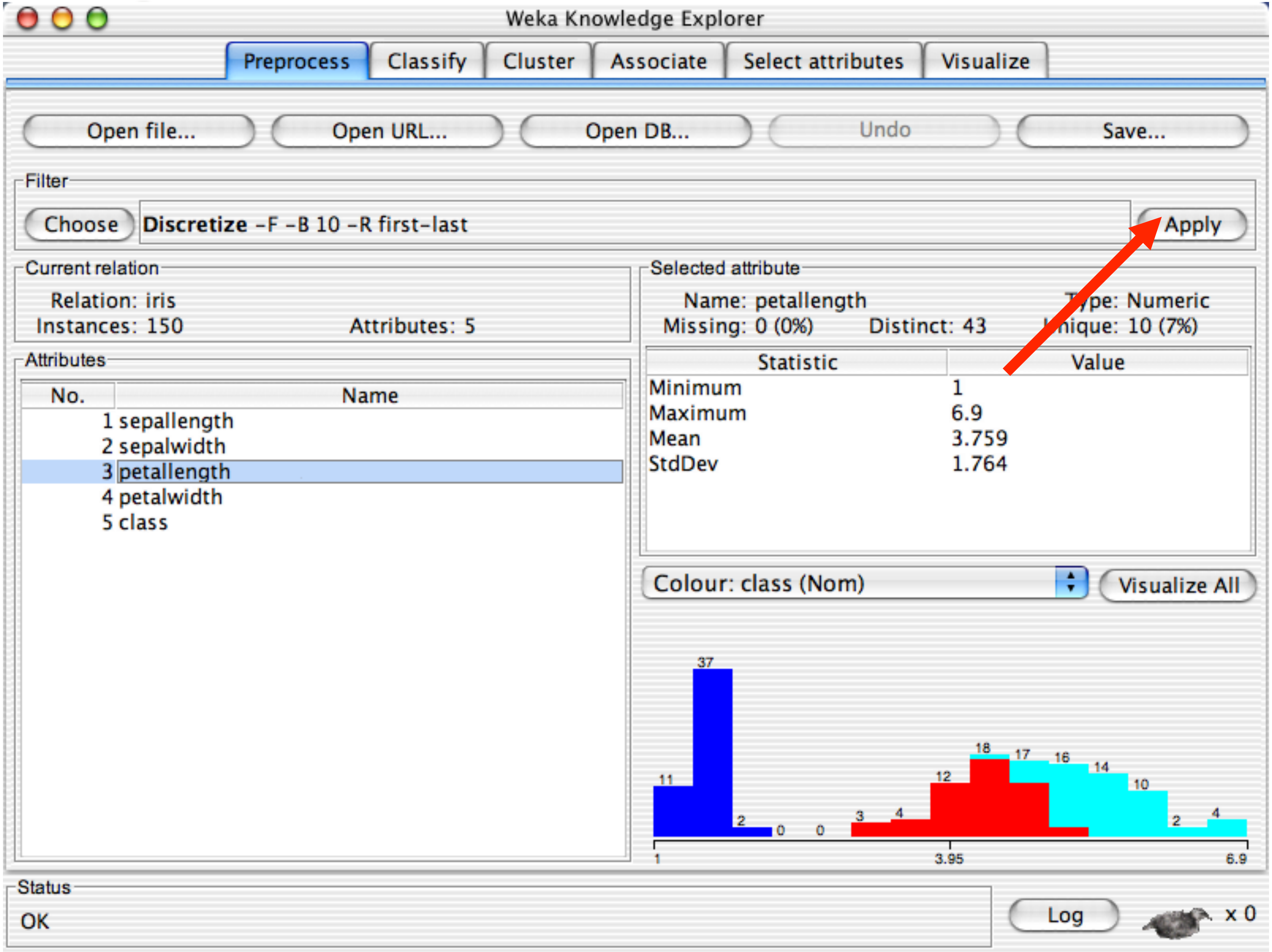


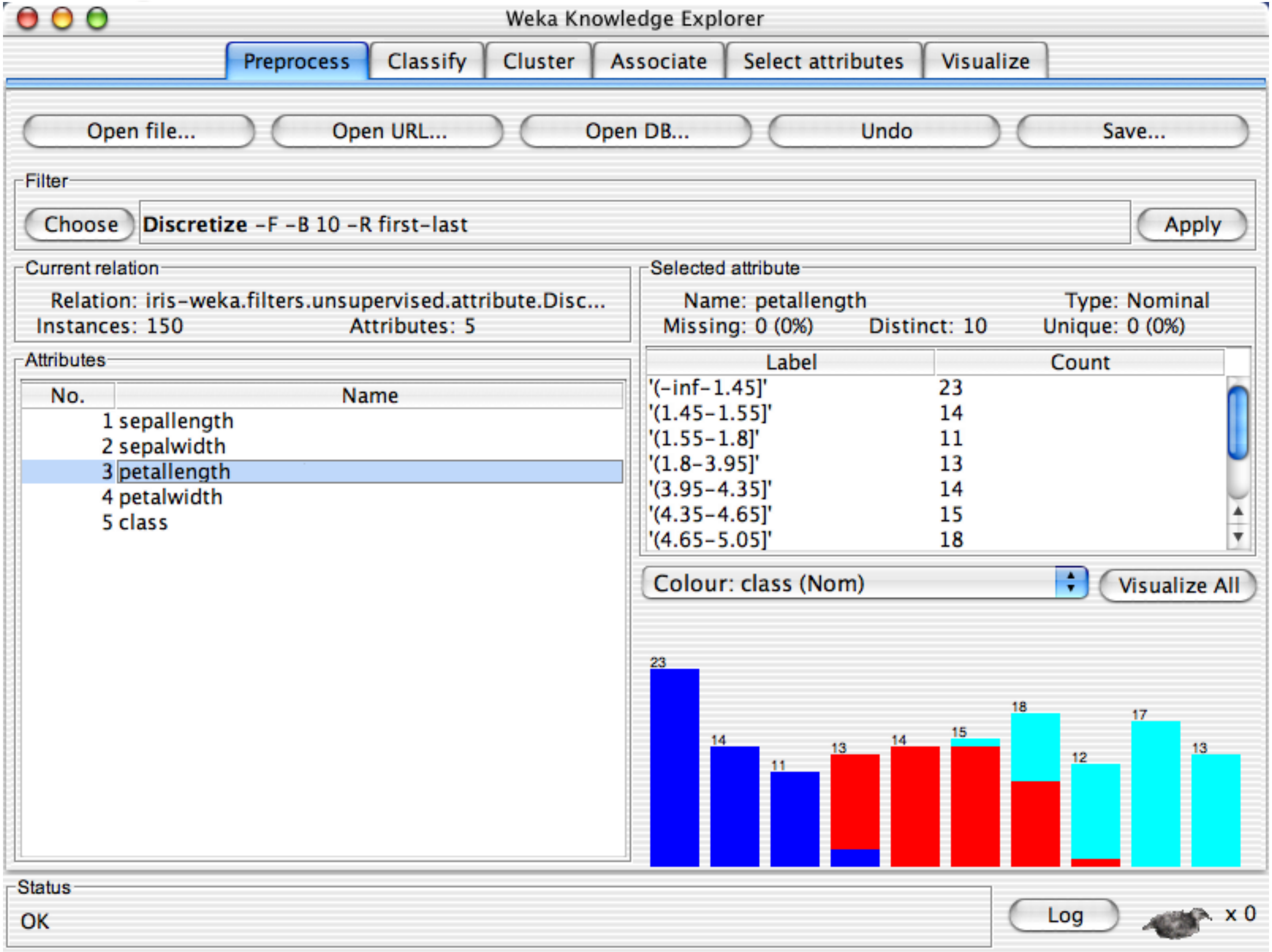
Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
Discretize -B 10 -R first-last
Apply
Current relation
Relation: iris
Instances: 150
Attributes:
Attributes
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
weka.gui.GenericObjectEditor
weka.filters.unsupervised.attribute.Discretize
About
An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.
More
attributeIndices
first-last
bins
10
findNumBins
False
invertSelection
False
makeBinary
False
useEqualFrequency
False
Open...
Save...
OK
Cancel
: Numeric
: 10 (7%)
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
Discretize -B 10 -R first-last
Apply
Current relation
Relation: iris
Instances: 150
Attributes:
Attributes
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
: Numeric
: 10 (7%)
Visualize All
weka.gui.GenericObjectEditor
weka.filters.unsupervised.attribute.Discretize
About
An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.
More
attributeIndices
first-last
bins
10
findNumBins
False
invertSelection
False
makeBinary
False
useEqualFrequency
True
Open...
Save...
OK
Cancel
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
Discretize -B 10 -R first-last
Apply
Current relation
Relation: iris
Instances: 150
Attributes:
: Numeric
: 10 (7%)
Attributes
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Visualize All
weka.gui.GenericObjectEditor
weka.filters.unsupervised.attribute.Discretize
About
An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.
More
attributeIndices
first-last
bins
10
findNumBins
False
invertSelection
False
makeBinary
False
useEqualFrequency
True
Open...
Save...
OK
Cancel
Status
OK
Log
x 0

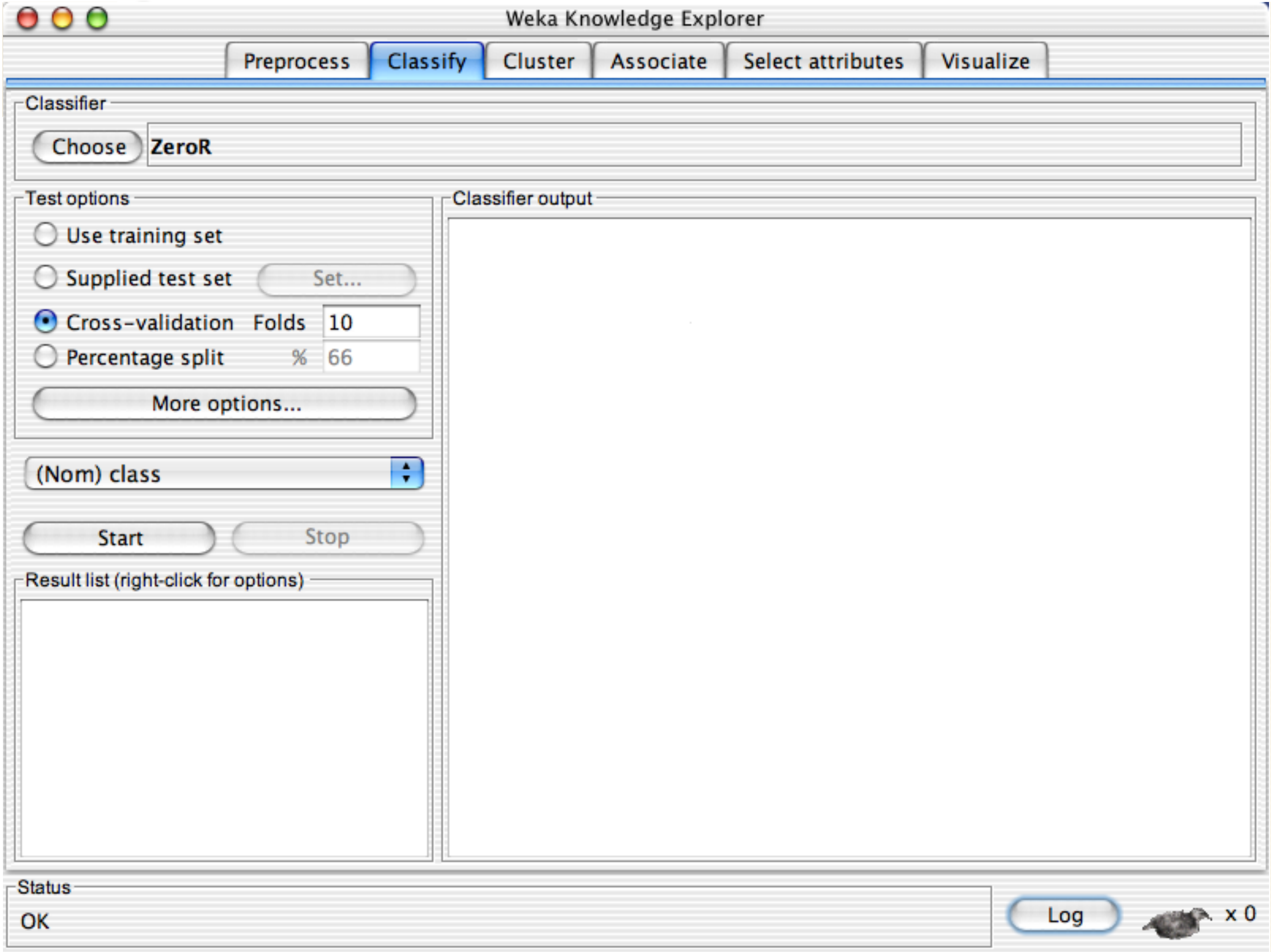

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
Discretize -F -B 10 -R first-last
Apply
Current relation
Relation: iris
Instances: 150
Attributes: 5
Attributes
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Selected attribute
Name: petallength
Type: Numeric
Missing: 0 (0%)
Distinct: 43
Unique: 10 (7%)
Statistic
Value
Minimum
1
Maximum
6.9
Mean
3.759
StdDev
1.764
Colour: class (Nom)
Visualize All
11
37
2
0
0
3
4
12
18
17
16
14
10
2
4
1
3.95
6.9
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
Discretize -F -B 10 -R first-last
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: petallength
Type: Numeric
Missing: 0 (0%)
Distinct: 43
Unique: 10 (7%)
Statistic
Value
Minimum 1
Maximum 6.9
Mean 3.759
StdDev 1.764
Colour: class (Nom)
Visualize All
11
37
2
0
0
3
4
12
18
17
16
14
10
2
4
1
3.95
6.9
Status
OK
Log
x 0

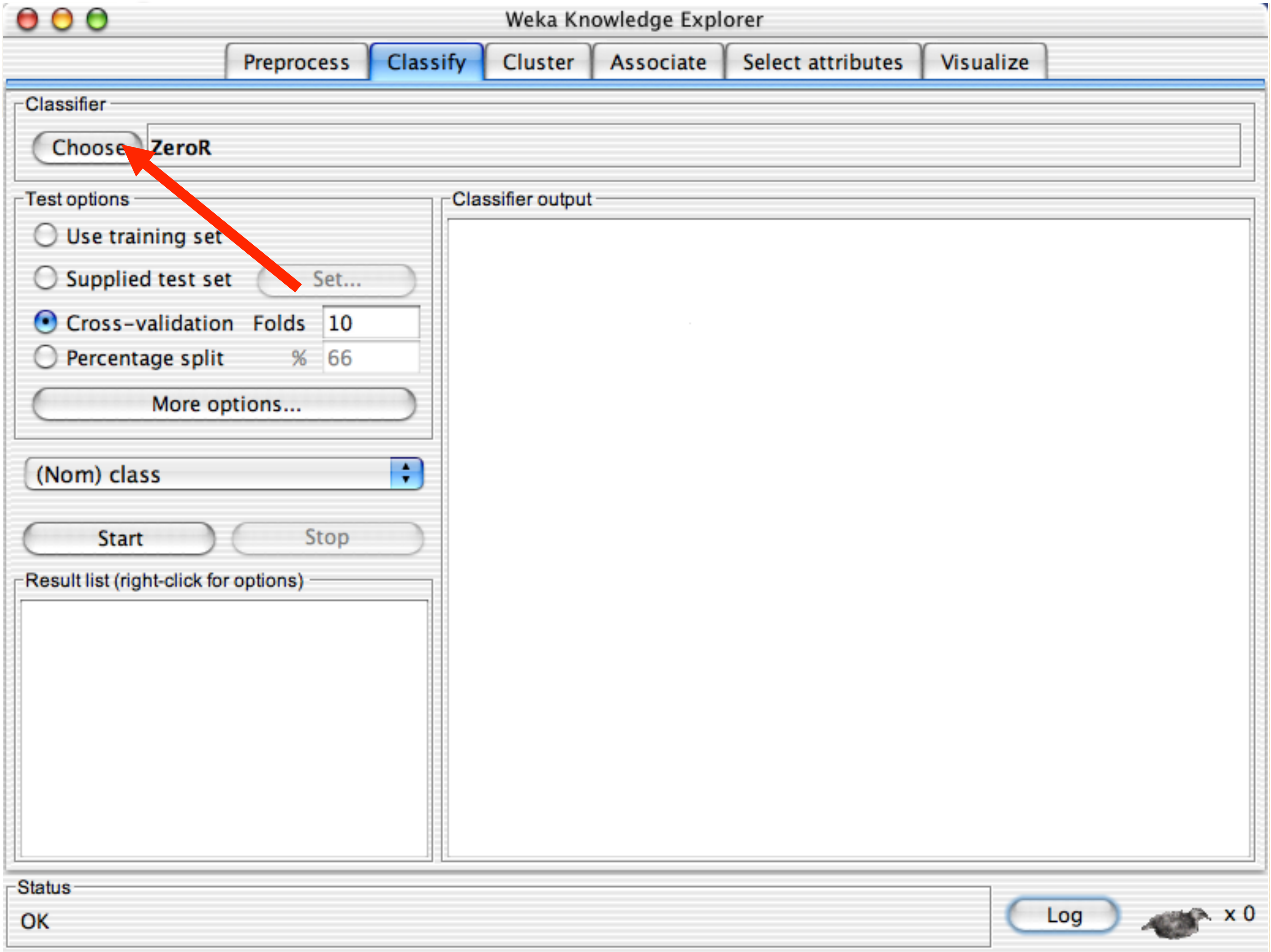

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
Discretize -F -B 10 -R first-last
Apply
Current relation
Relation: iris-weka.filters.unsupervised.attribute.Disc...
Instances: 150
Attributes: 5
Attributes
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Selected attribute
Name: petallength
Type: Nominal
Missing: 0 (0%)
Distinct: 10
Unique: 0 (0%)
Label
Count
'(-inf-1.45]' 23
'(1.45-1.55]' 14
'(1.55-1.8]' 11
'(1.8-3.95]' 13
'(3.95-4.35]' 14
'(4.35-4.65]' 15
'(4.65-5.05]' 18
Colour: class (Nom)
Visualize All
23
14
11
13
14
15
18
12
17
13
Status
OK
Log
x 0

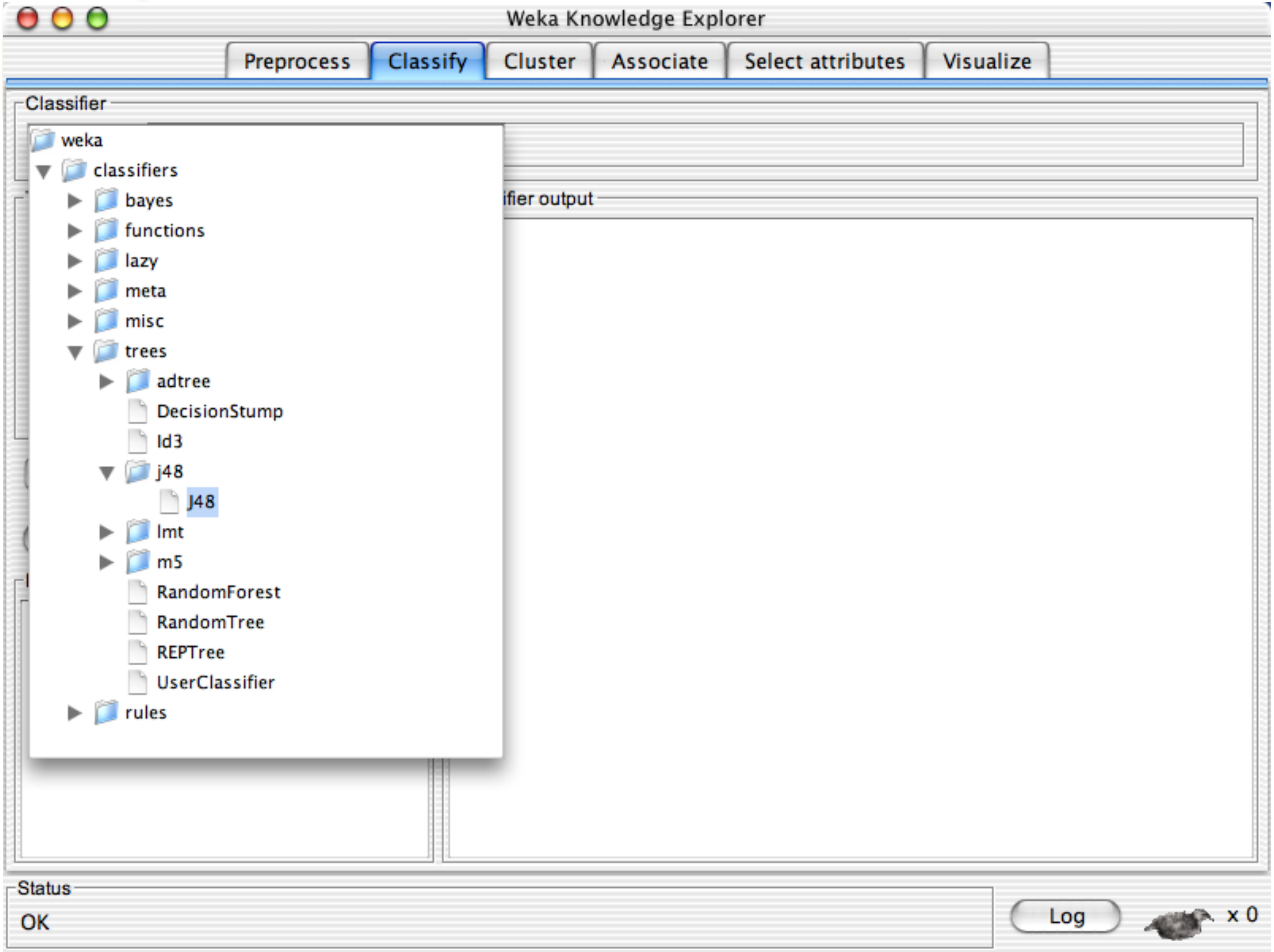
Explorer: building “classifiers”

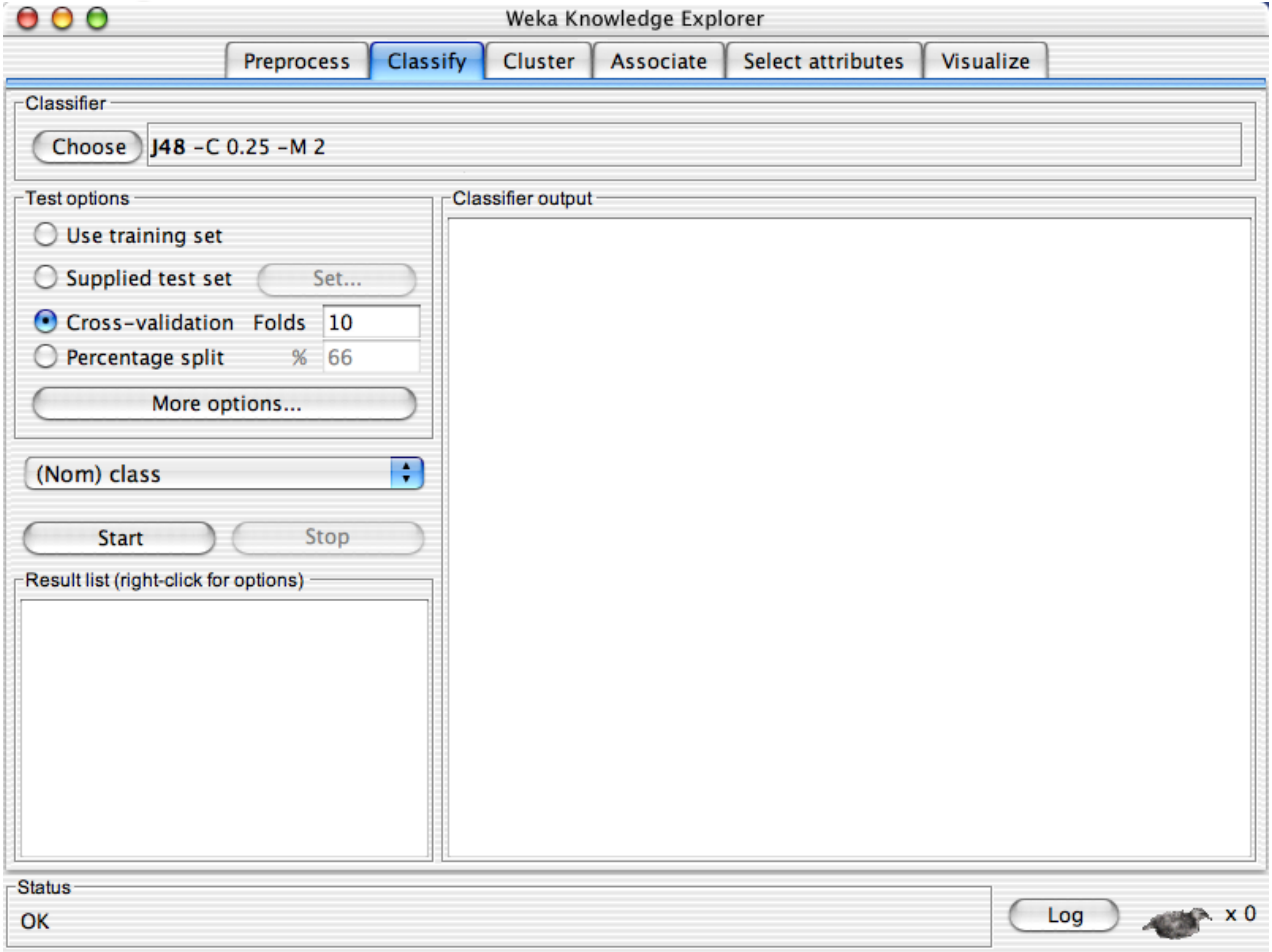
- Classifiers in WEKA are models for predicting nominal or numeric quantities
- Implemented learning schemes include:
  - Decision trees and lists, instance-based classifiers, support vector machines, multi-layer perceptrons, logistic regression, Bayes’ nets, …
- “Meta”-classifiers include:
  - Bagging,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

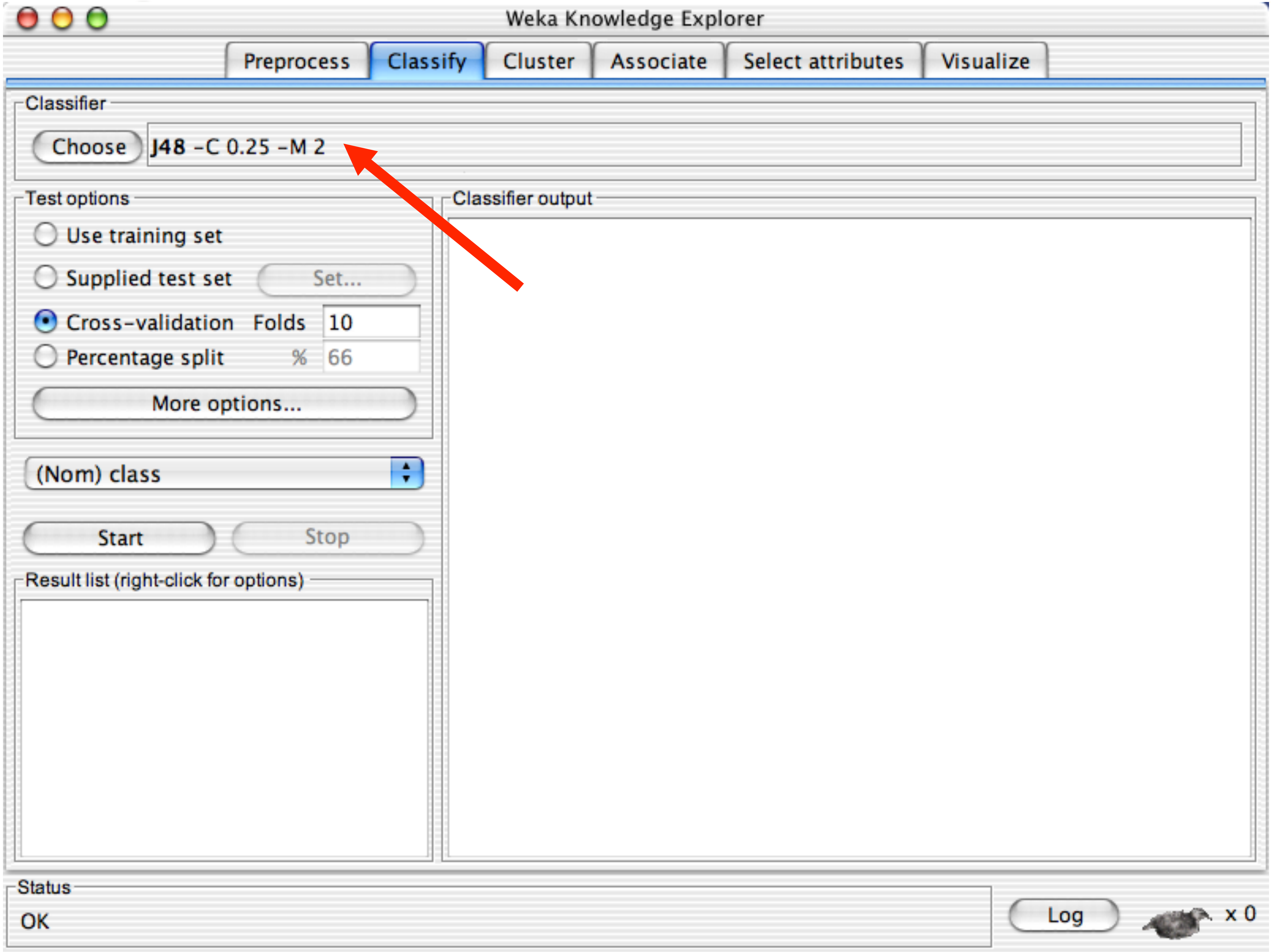


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
ZeroR
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
Classifier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

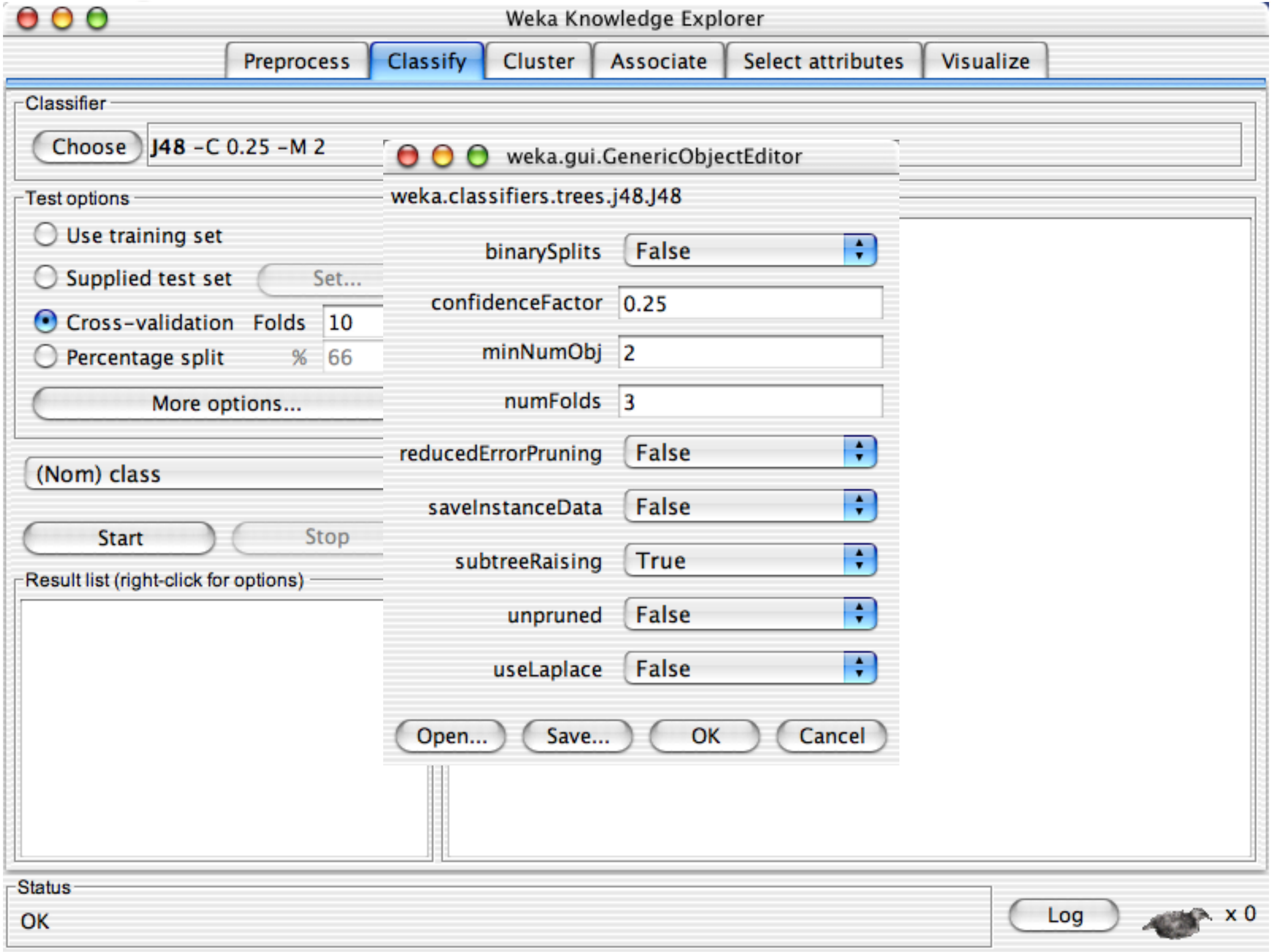

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
ZeroR
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
Classifier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

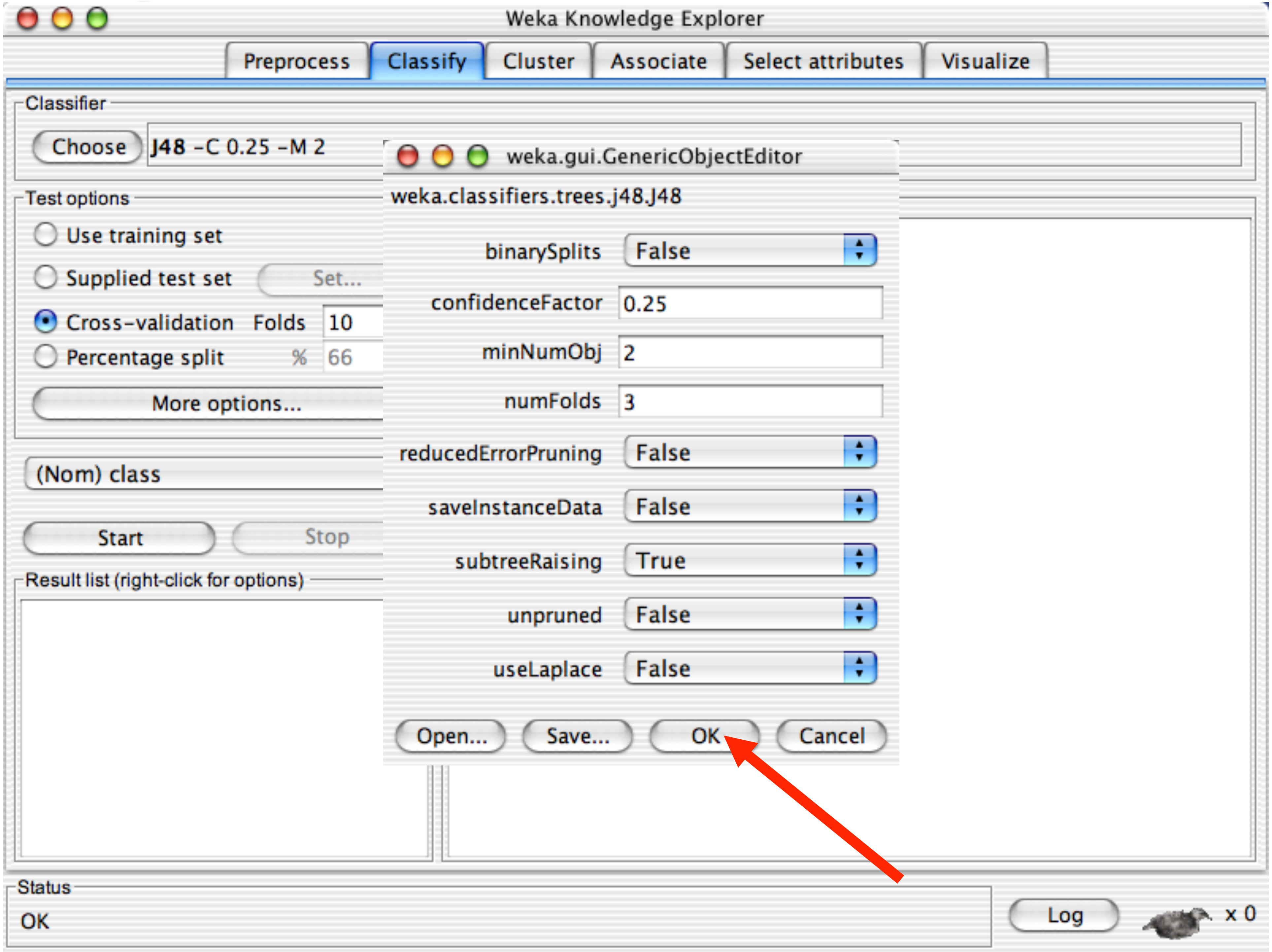

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
weka
classifiers
bayes
functions
lazy
meta
misc
trees
adtree
DecisionStump
Id3
j48
J48
lmt
m5
RandomForest
RandomTree
REPTree
UserClassifier
rules
ifier output
Status
OK
Log
x 0

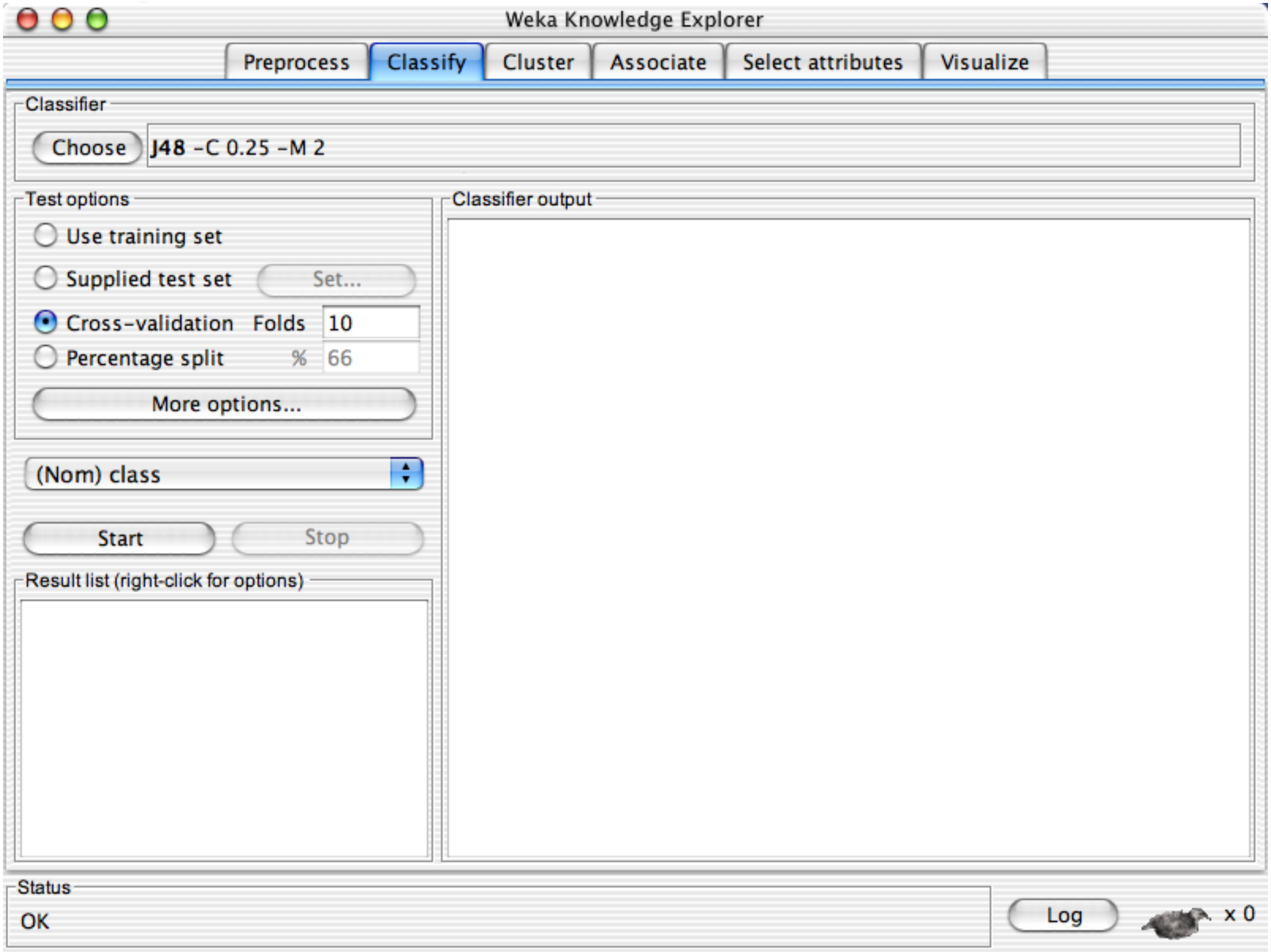

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)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

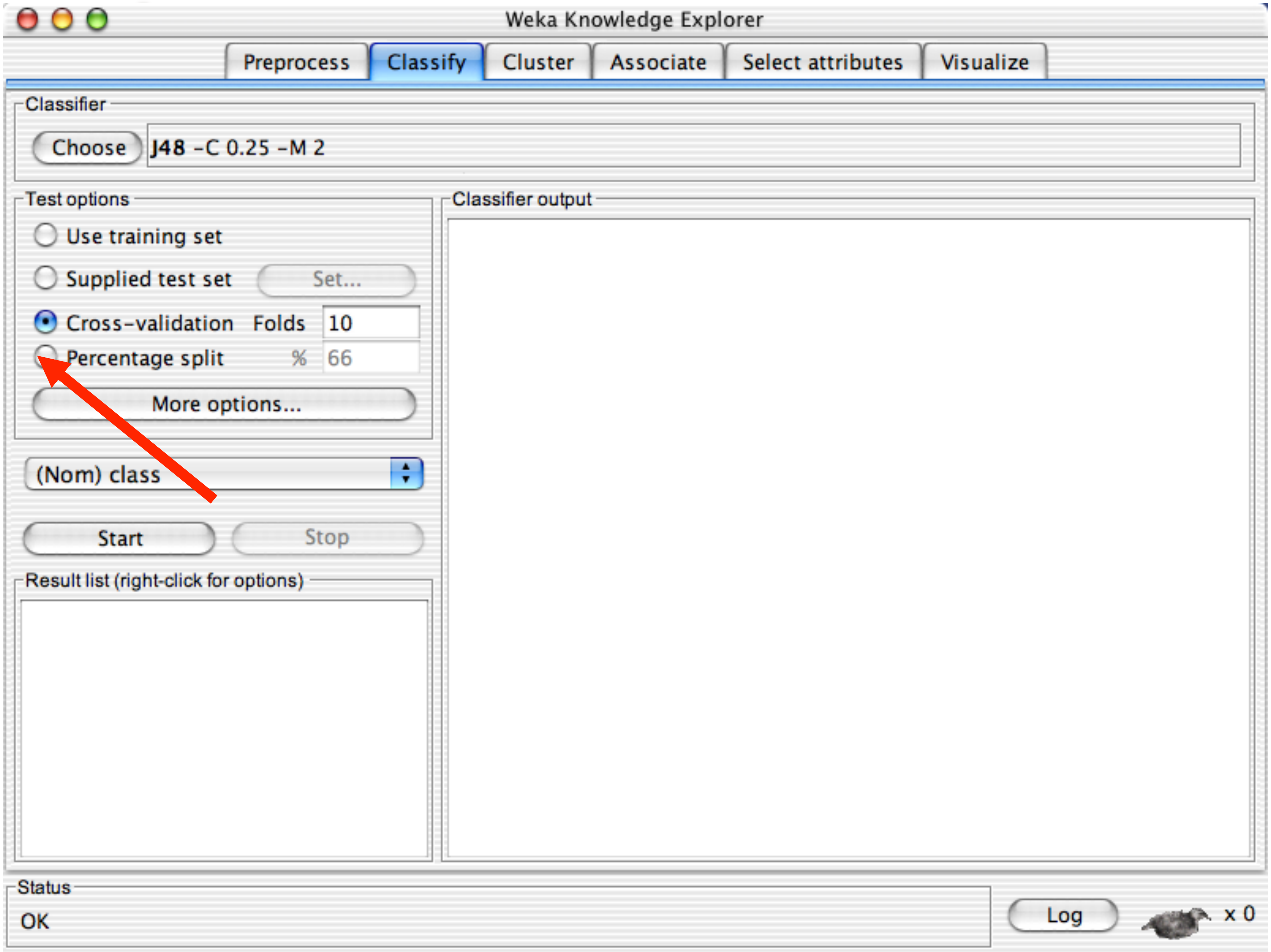


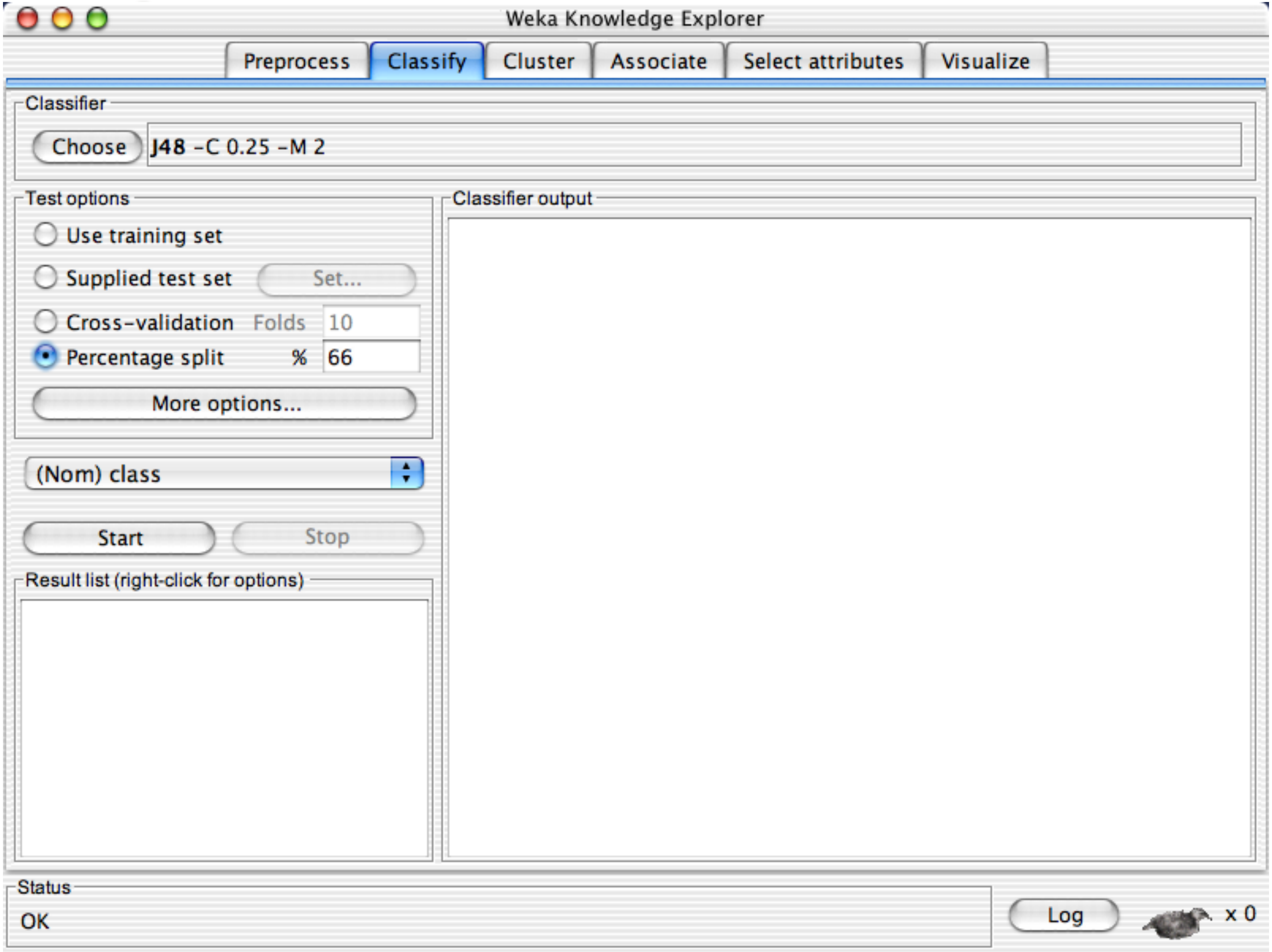
Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)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

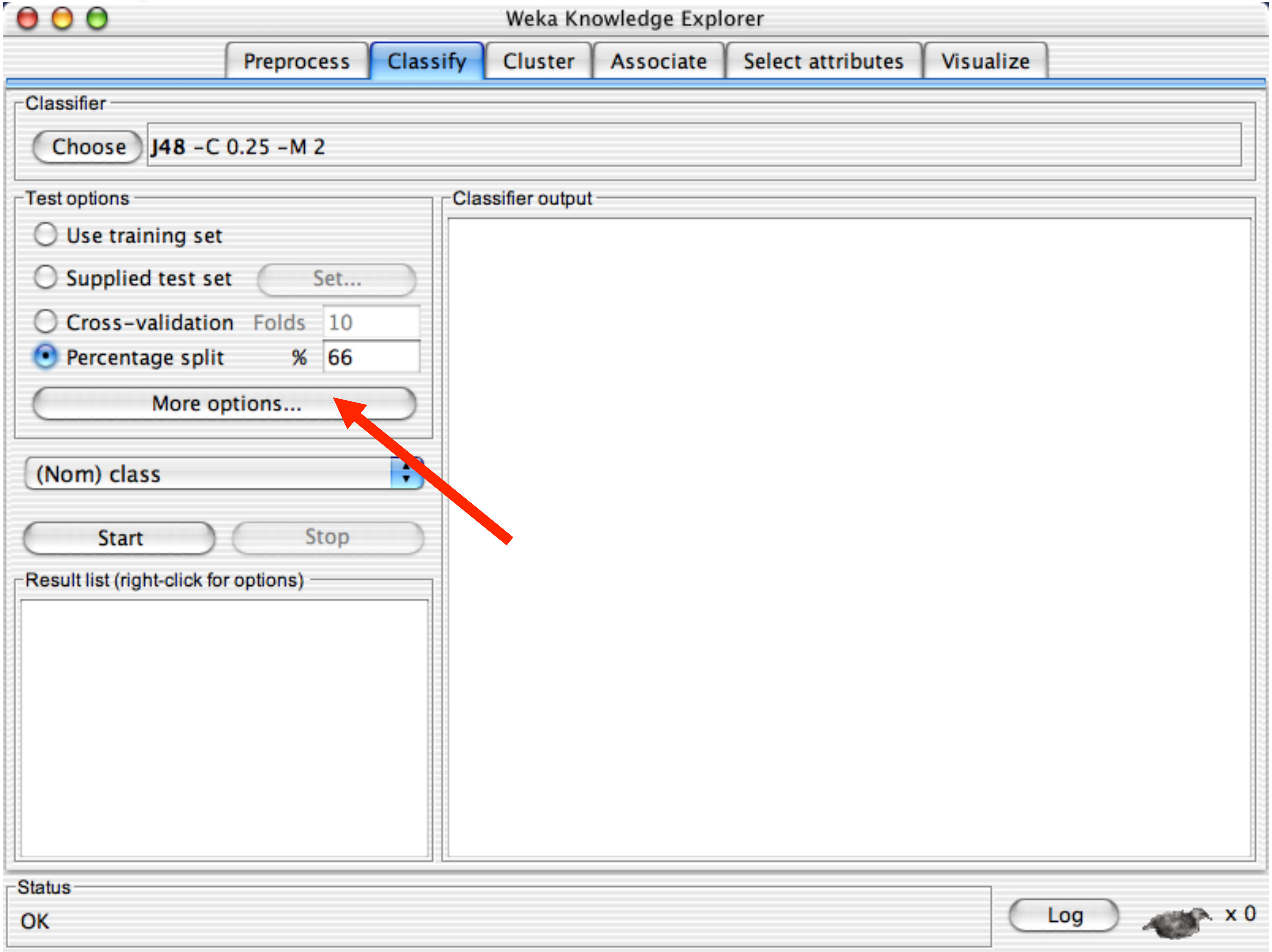

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
weka.gui.GenericObjectEditor
weka.classifiers.trees.j48.J48
binarySplits
False
confidenceFactor
0.25
minNumObj
2
numFolds
3
reducedErrorPruning
False
saveInstanceData
False
subtreeRaising
True
unpruned
False
useLaplace
False
Open...
Save...
OK
Cancel
Status
OK
Log
x 0

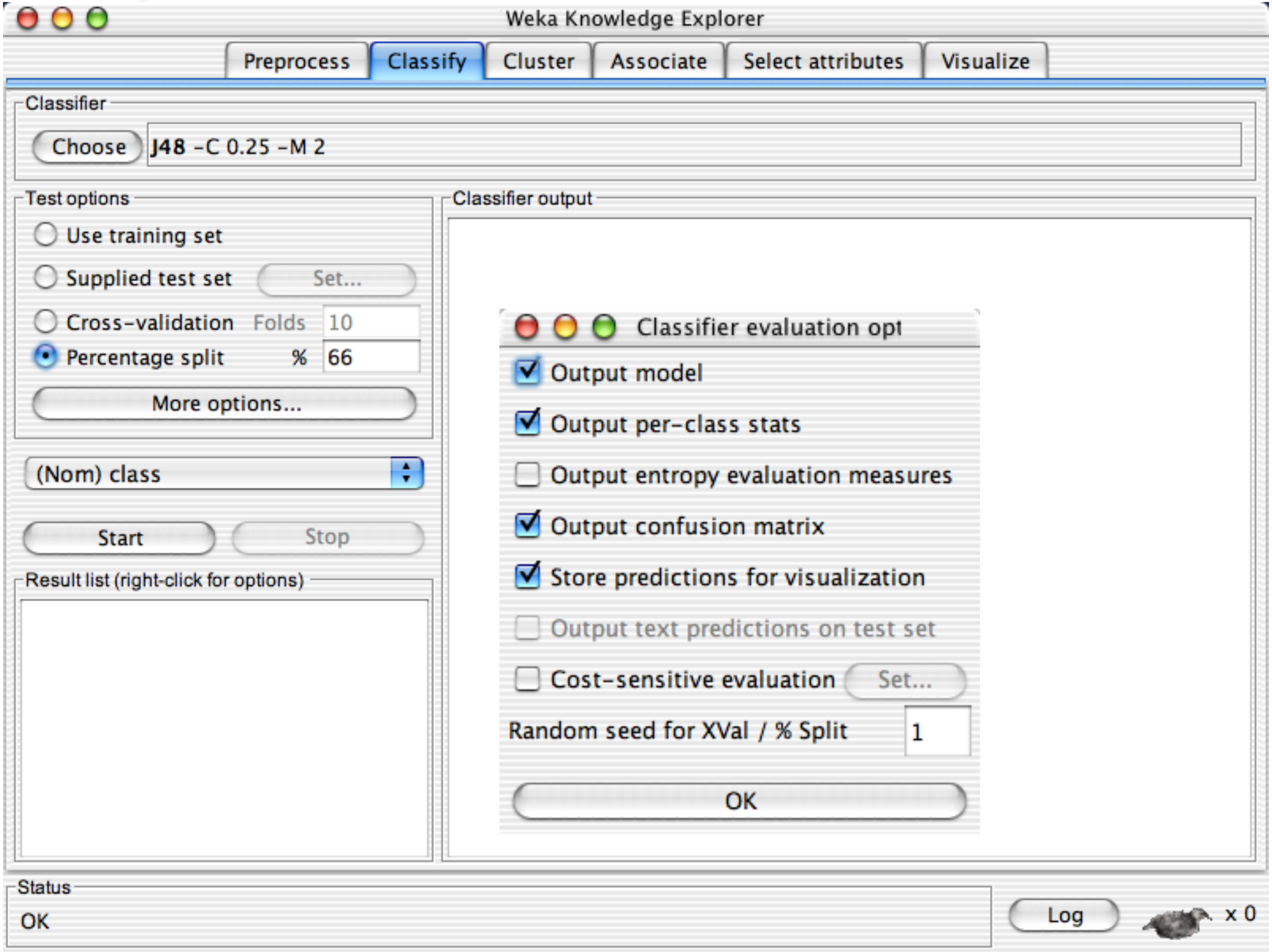

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
weka.gui.GenericObjectEditor
weka.classifiers.trees.j48.J48
binarySplits
False
confidenceFactor
0.25
minNumObj
2
numFolds
3
reducedErrorPruning
False
saveInstanceData
False
subtreeRaising
True
unpruned
False
useLaplace
False
Open...
Save...
OK
Cancel
Status
OK
Log
x 0

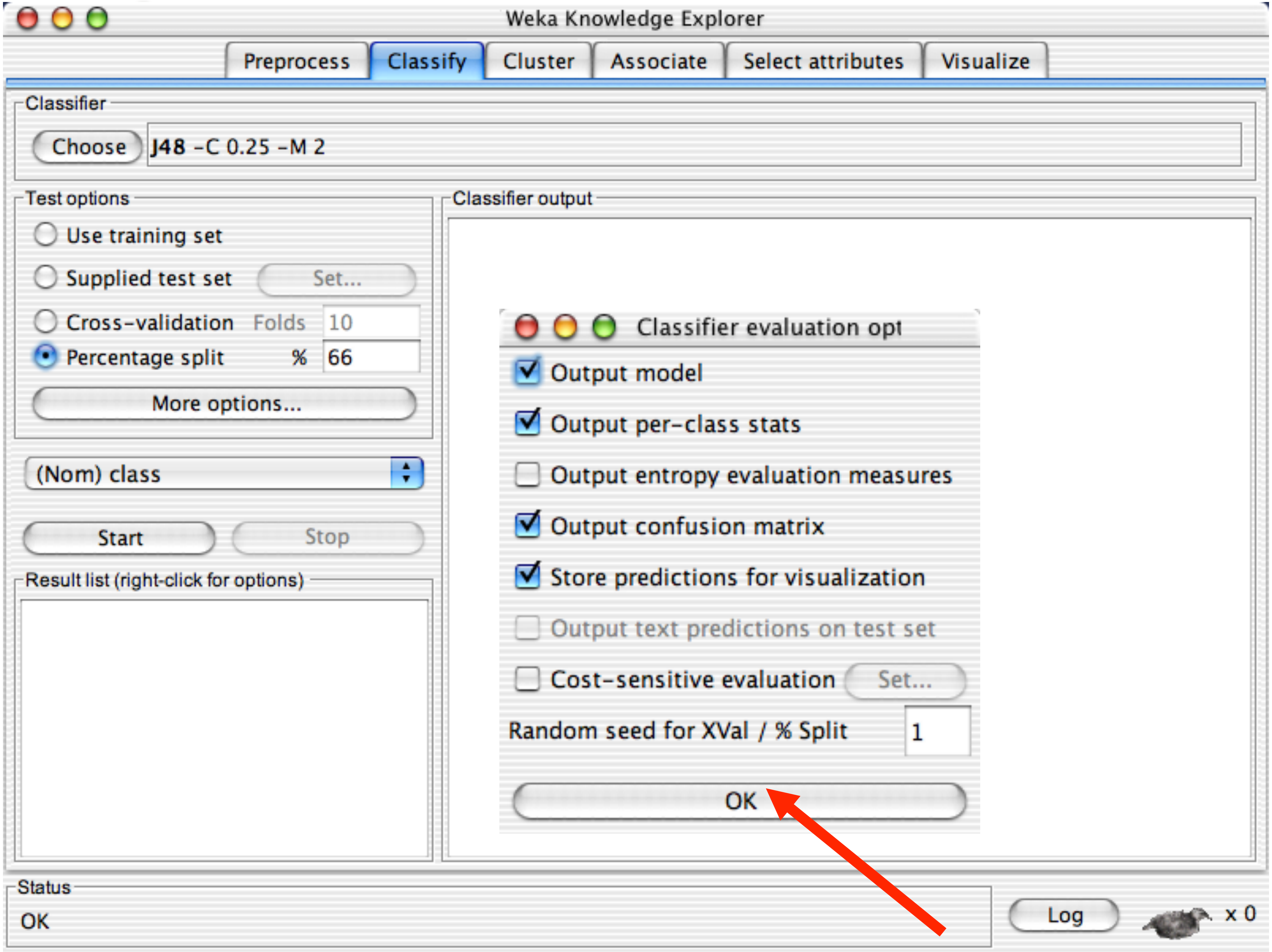

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)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

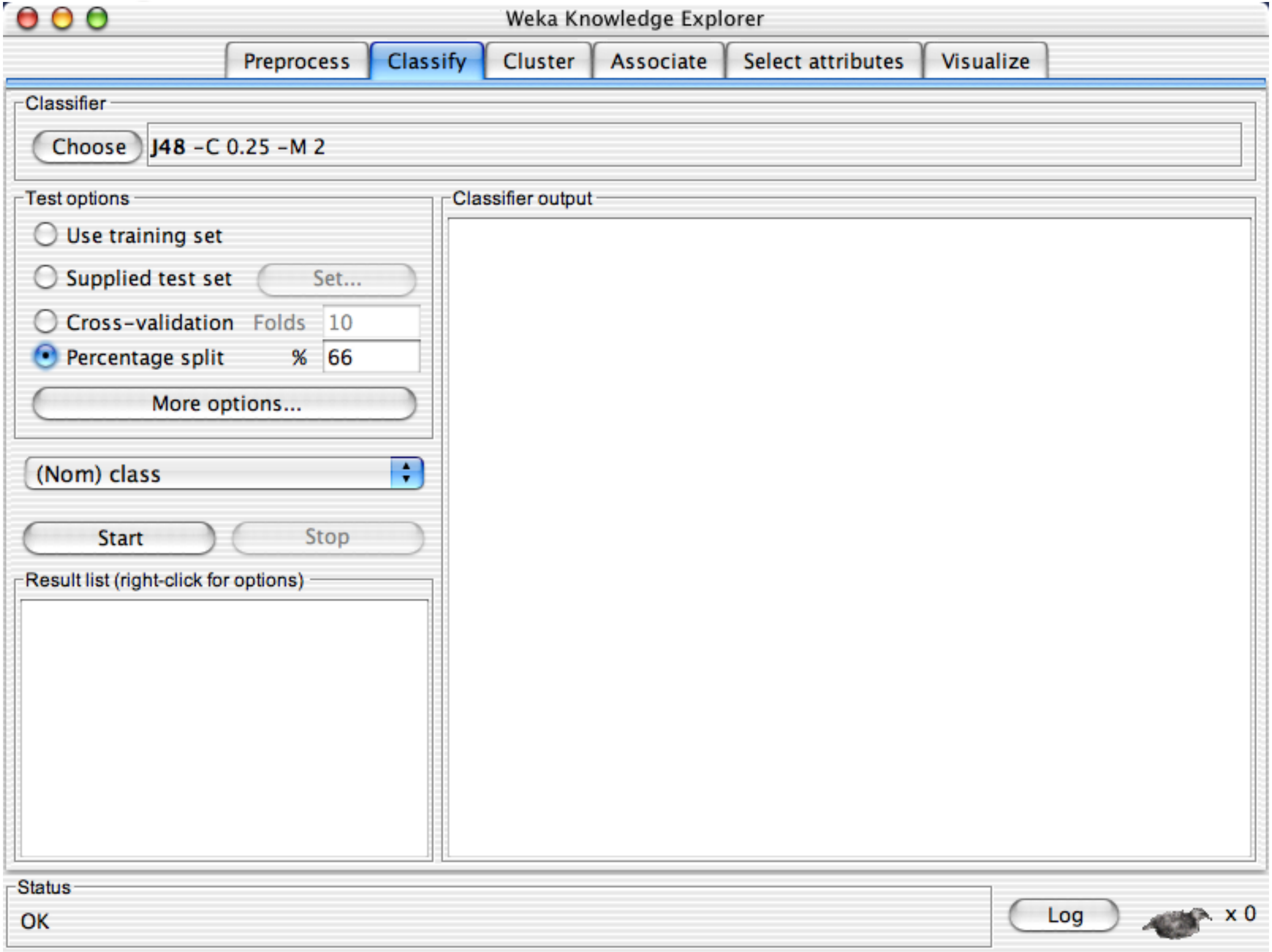

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)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

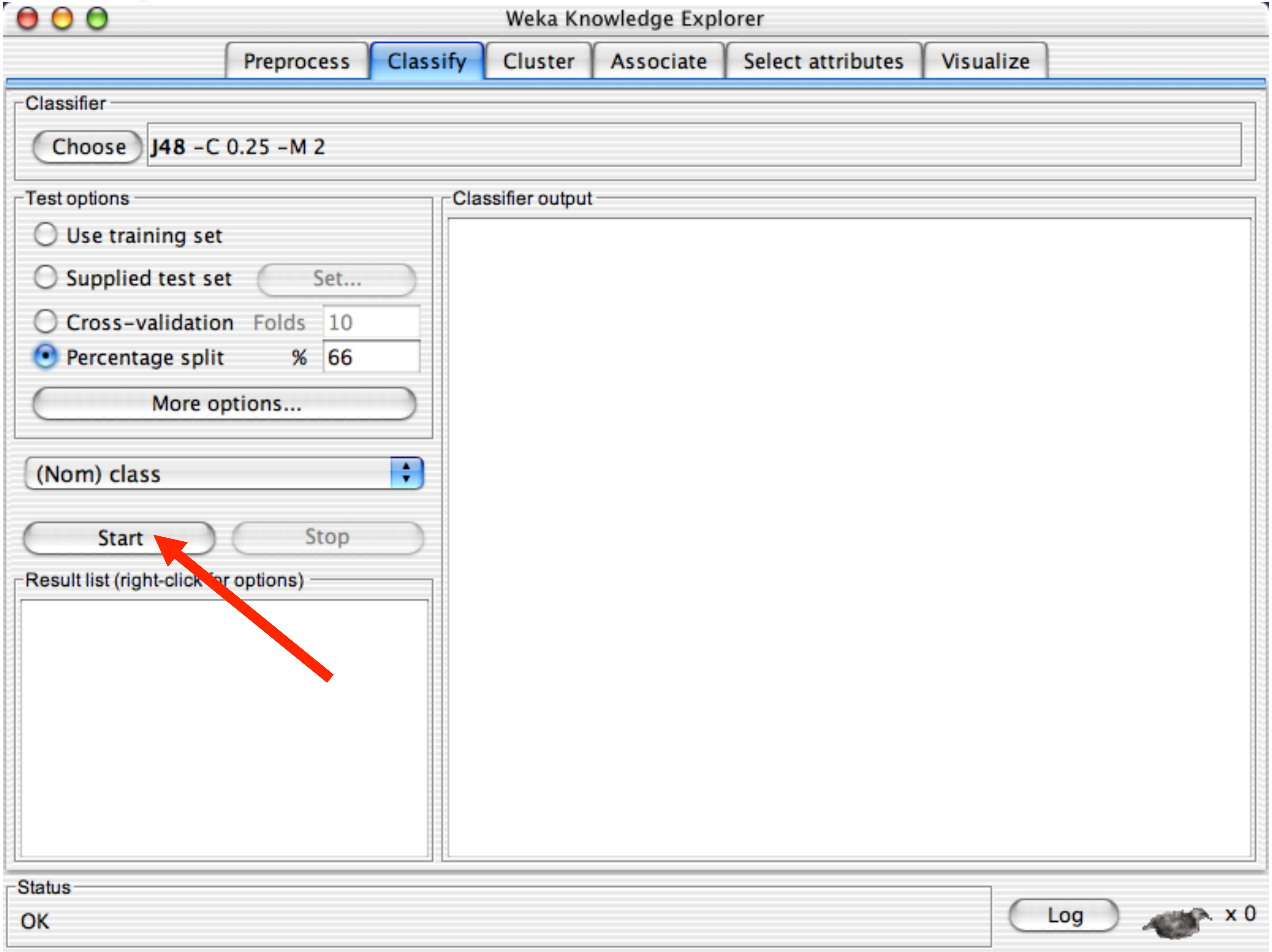

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)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

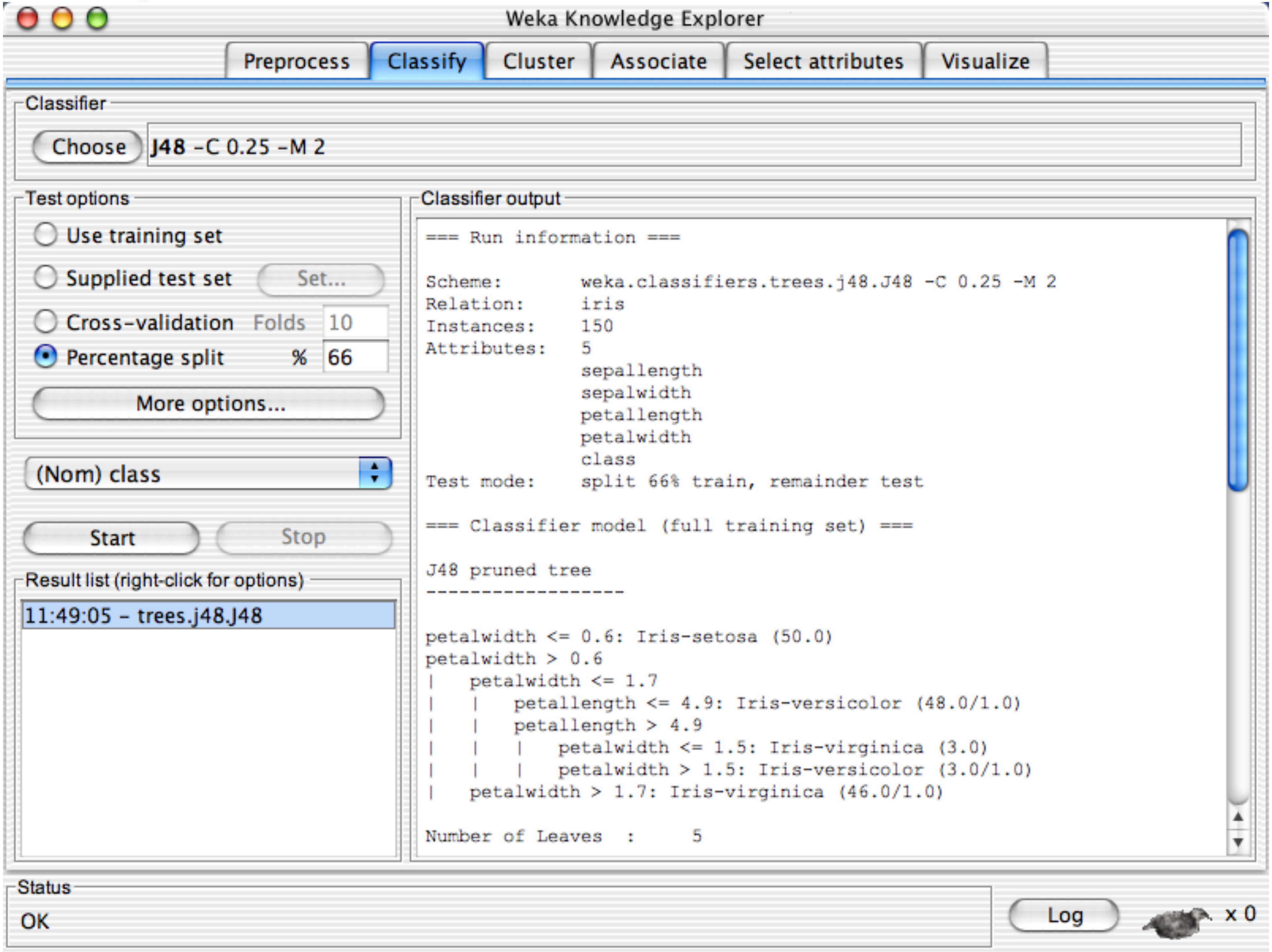

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
Classifier output
Classifier evaluation opt
Output model
Output per-class stats
Output entropy evaluation measures
Output confusion matrix
Store predictions for visualization
Output text predictions on test set
Cost-sensitive evaluation
Set...
Random seed for XVal / % Split
1
OK
Status
OK
Log
x 0

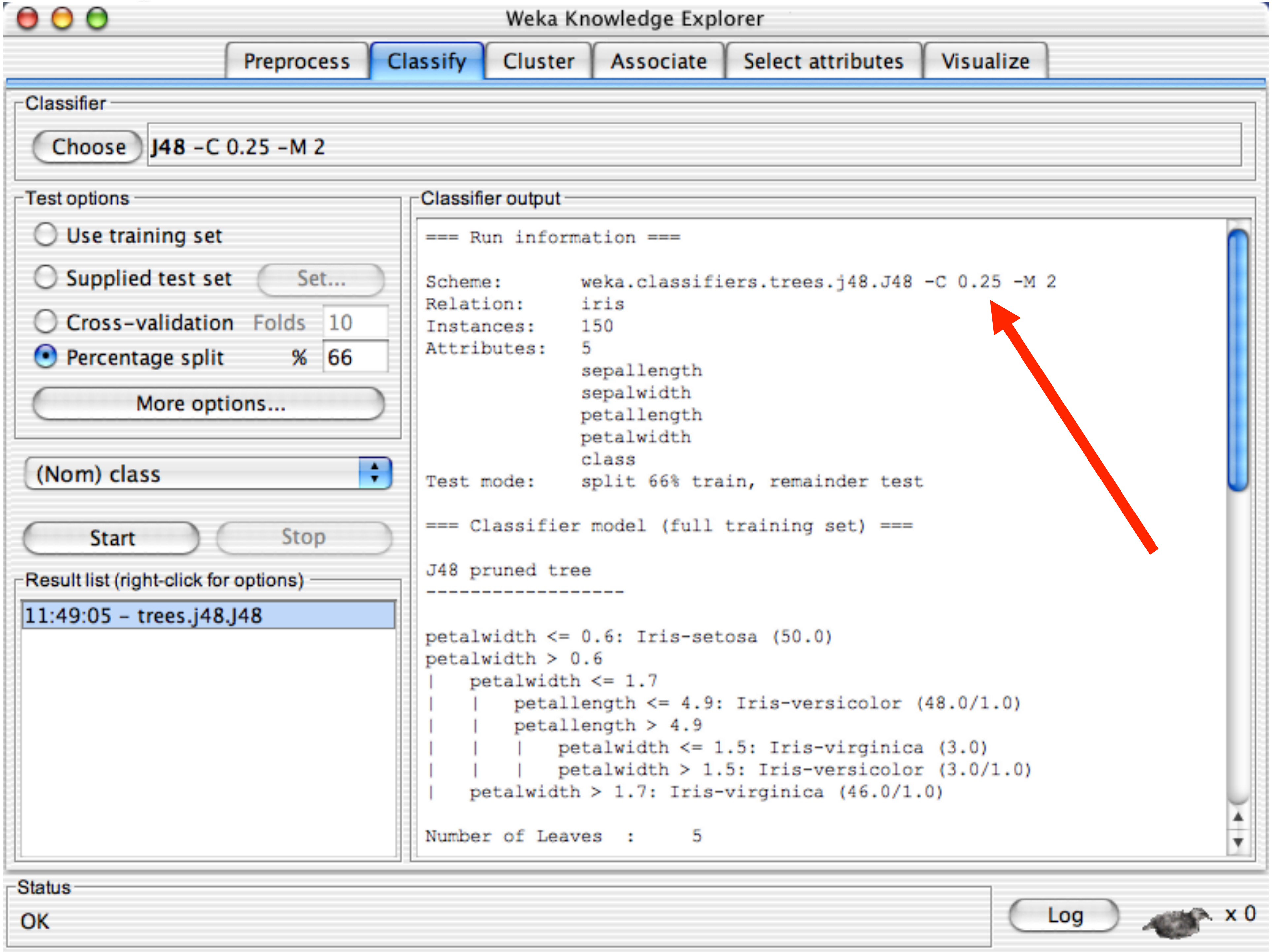

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
Classifier output
Classifier evaluation opt
Output model
Output per-class stats
Output entropy evaluation measures
Output confusion matrix
Store predictions for visualization
Output text predictions on test set
Cost-sensitive evaluation
Set...
Random seed for XVal / % Split
1
OK
Status
OK
Log
x 0

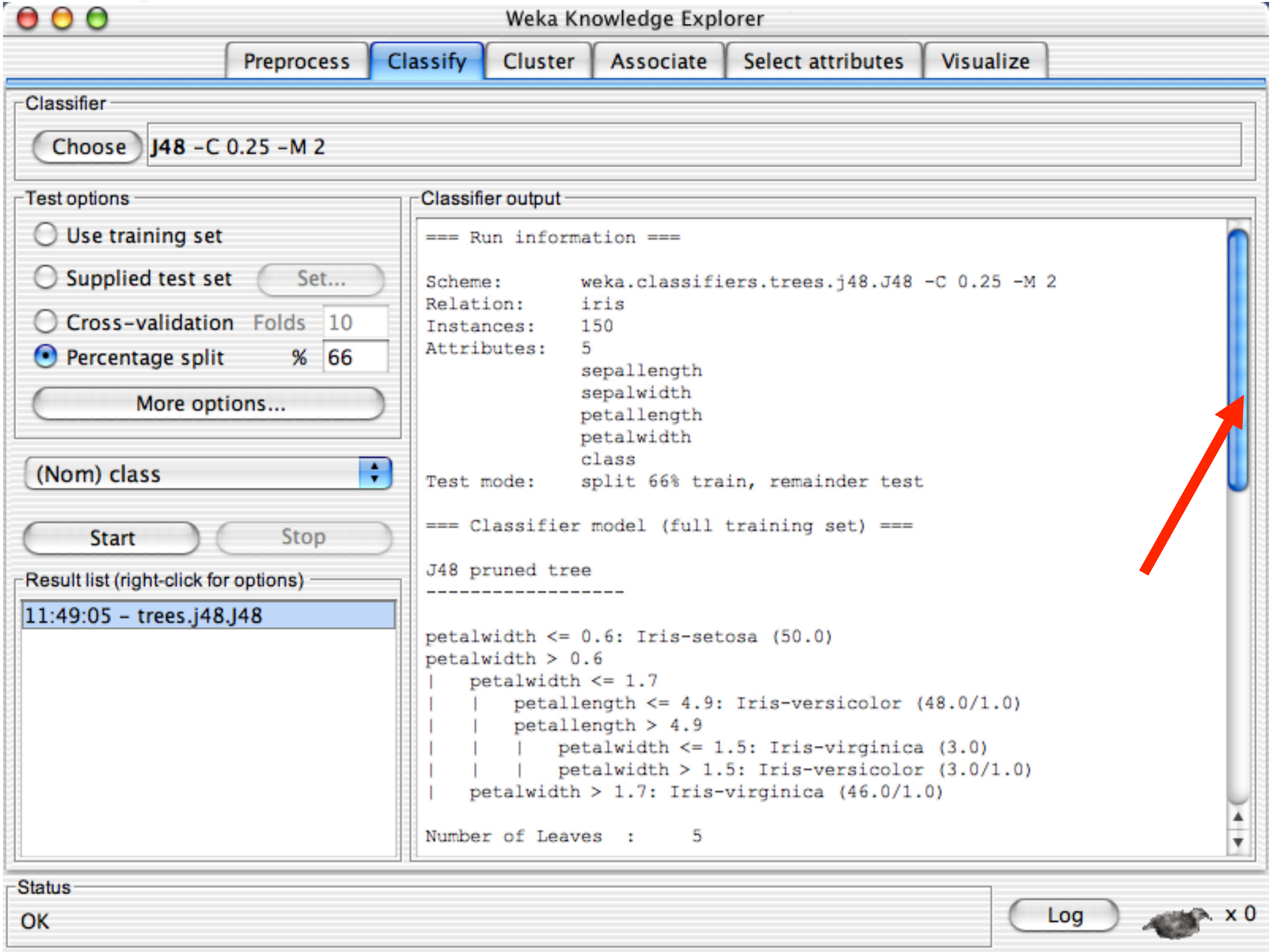

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)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



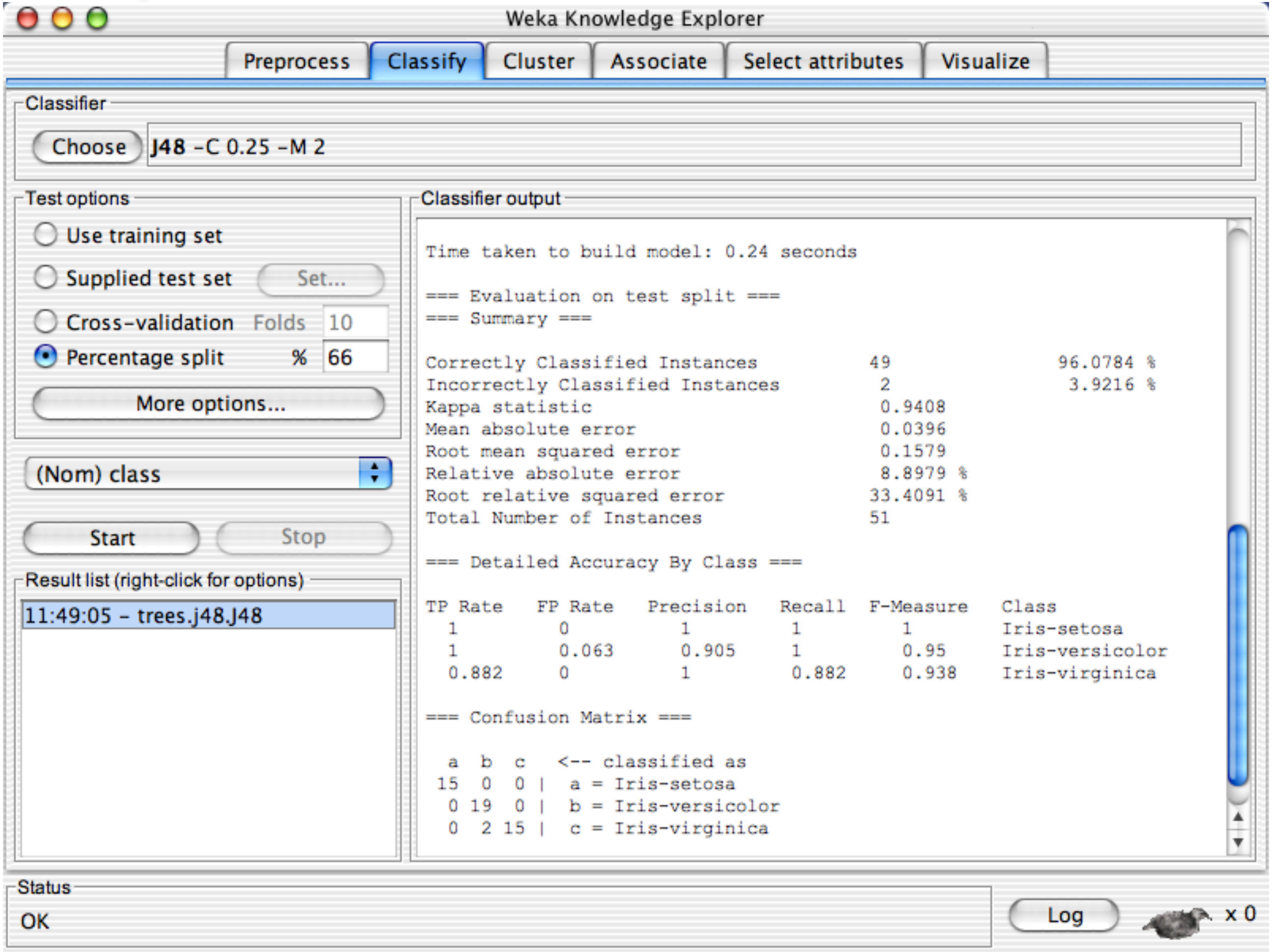
Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)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

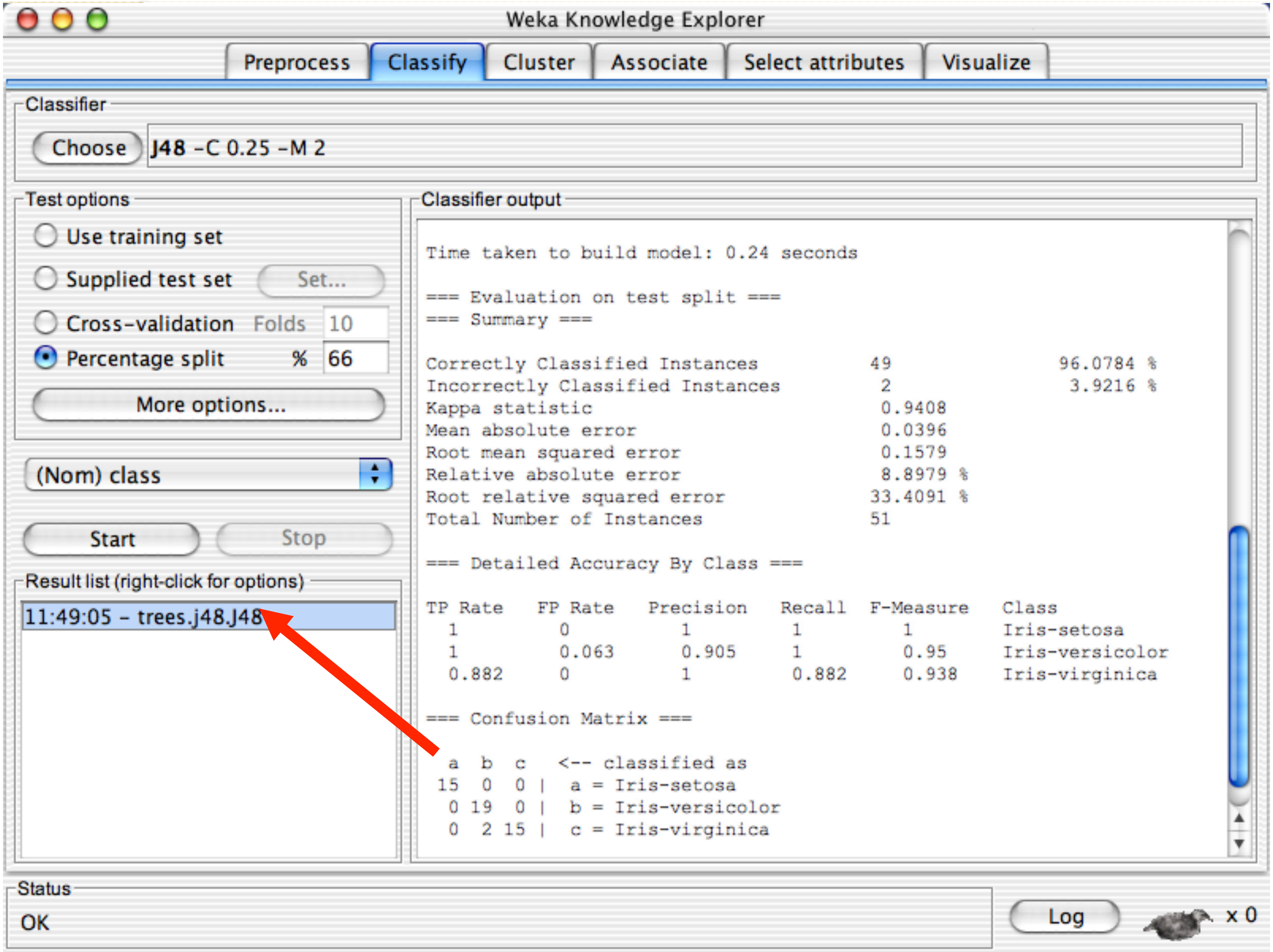

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)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

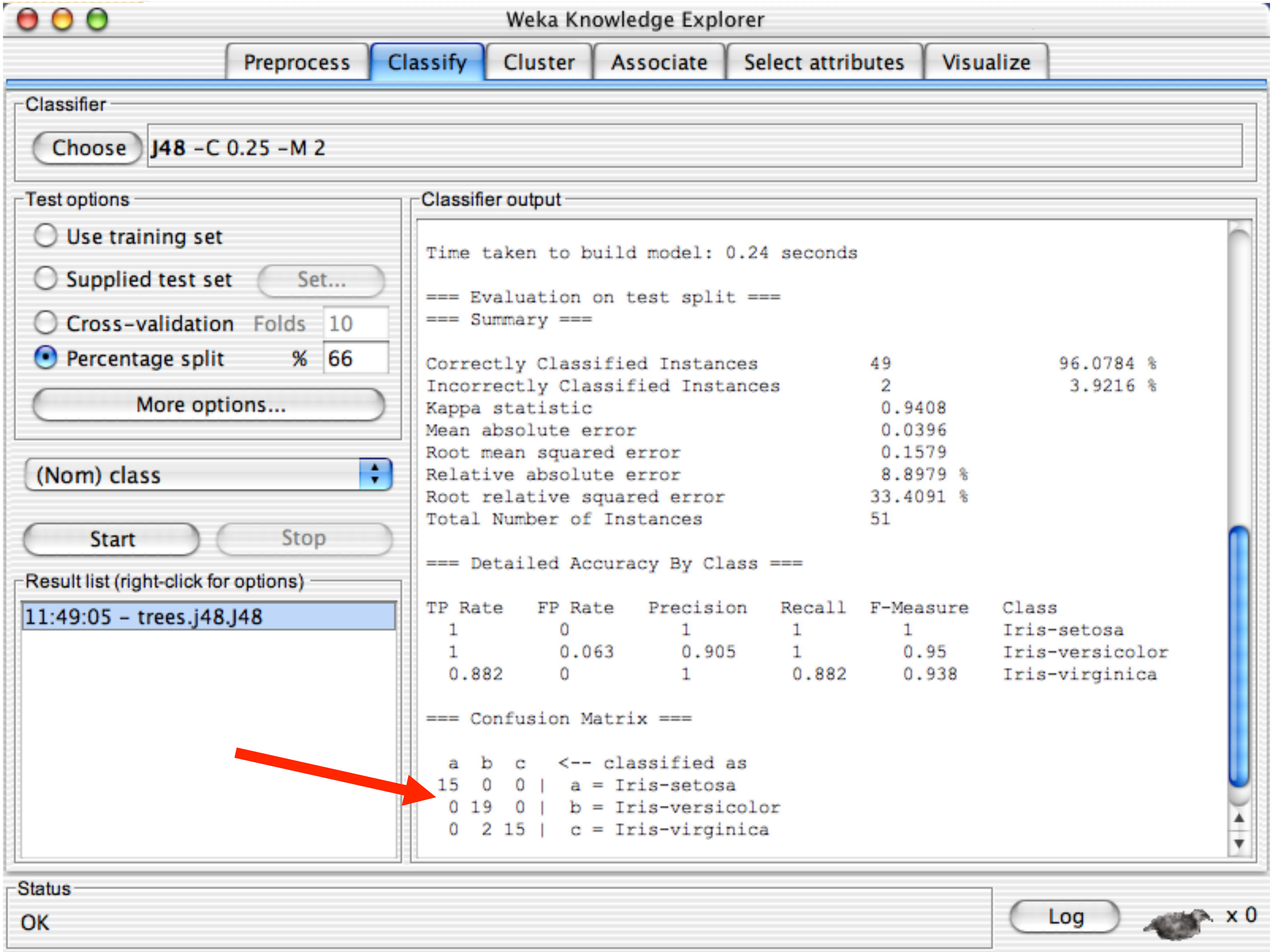

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
11:49:05 - trees.j48.J48
Classifier output
=== 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

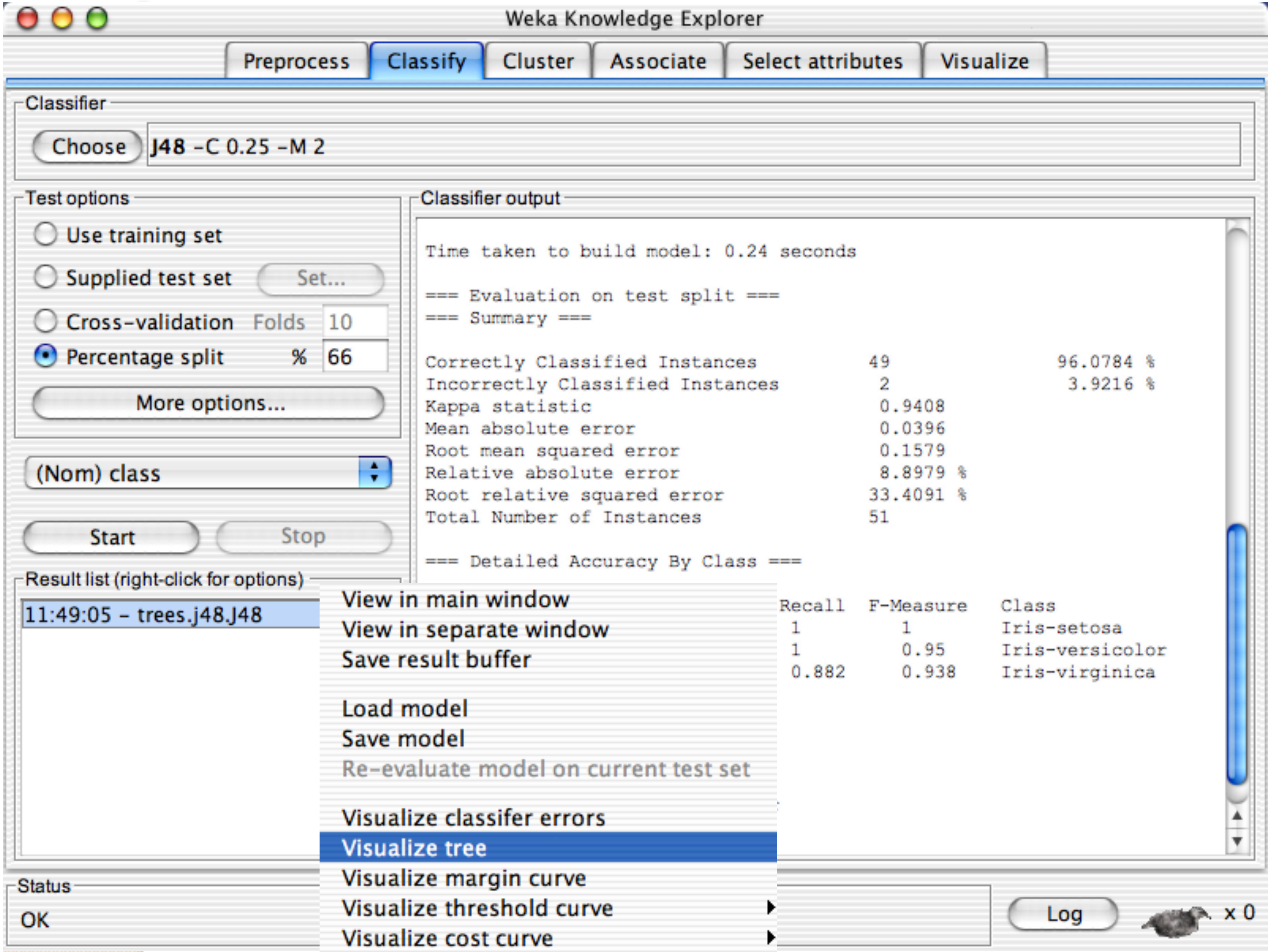


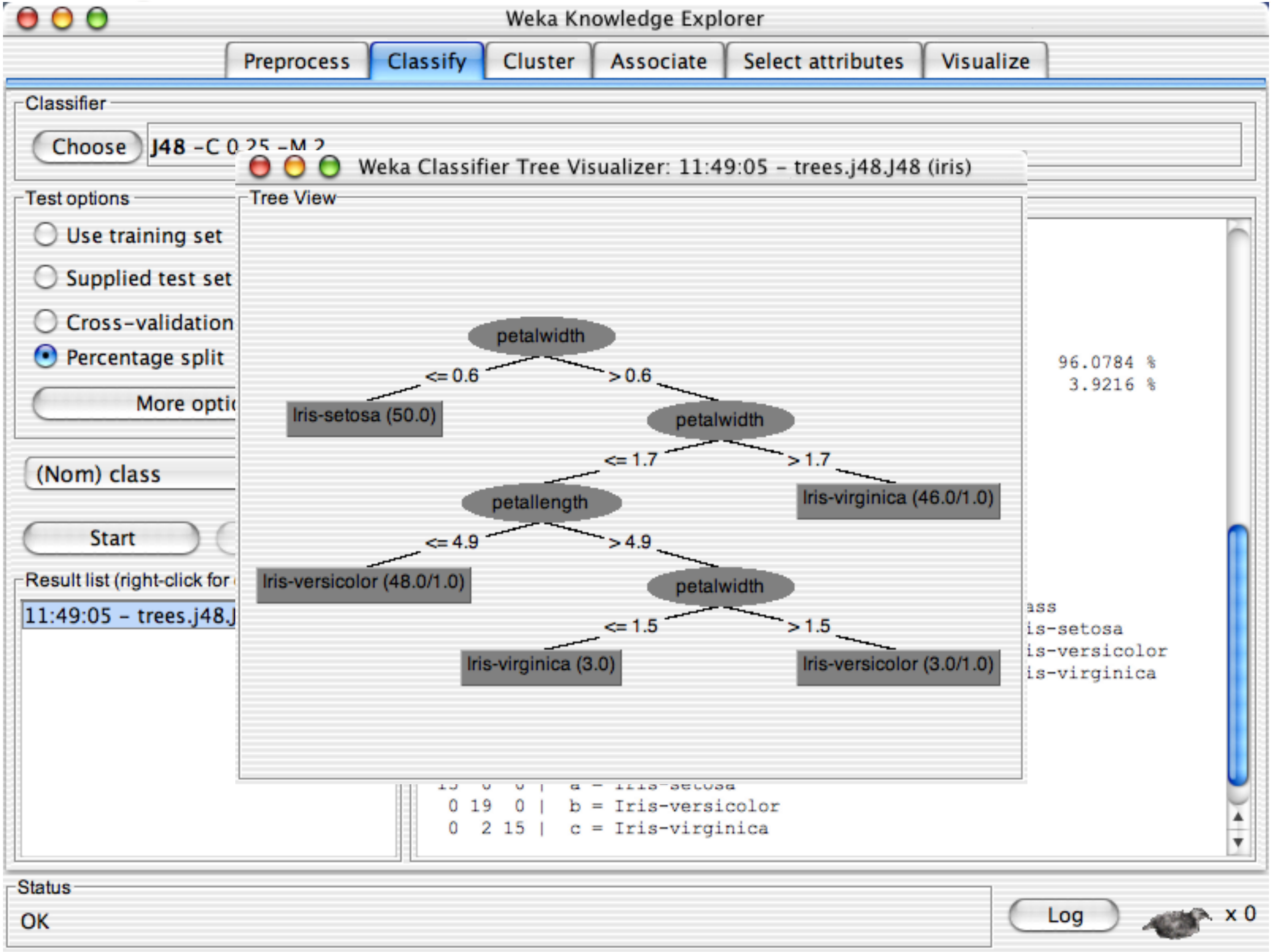
Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
11:49:05 - trees.j48.J48
Classifier output
=== 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

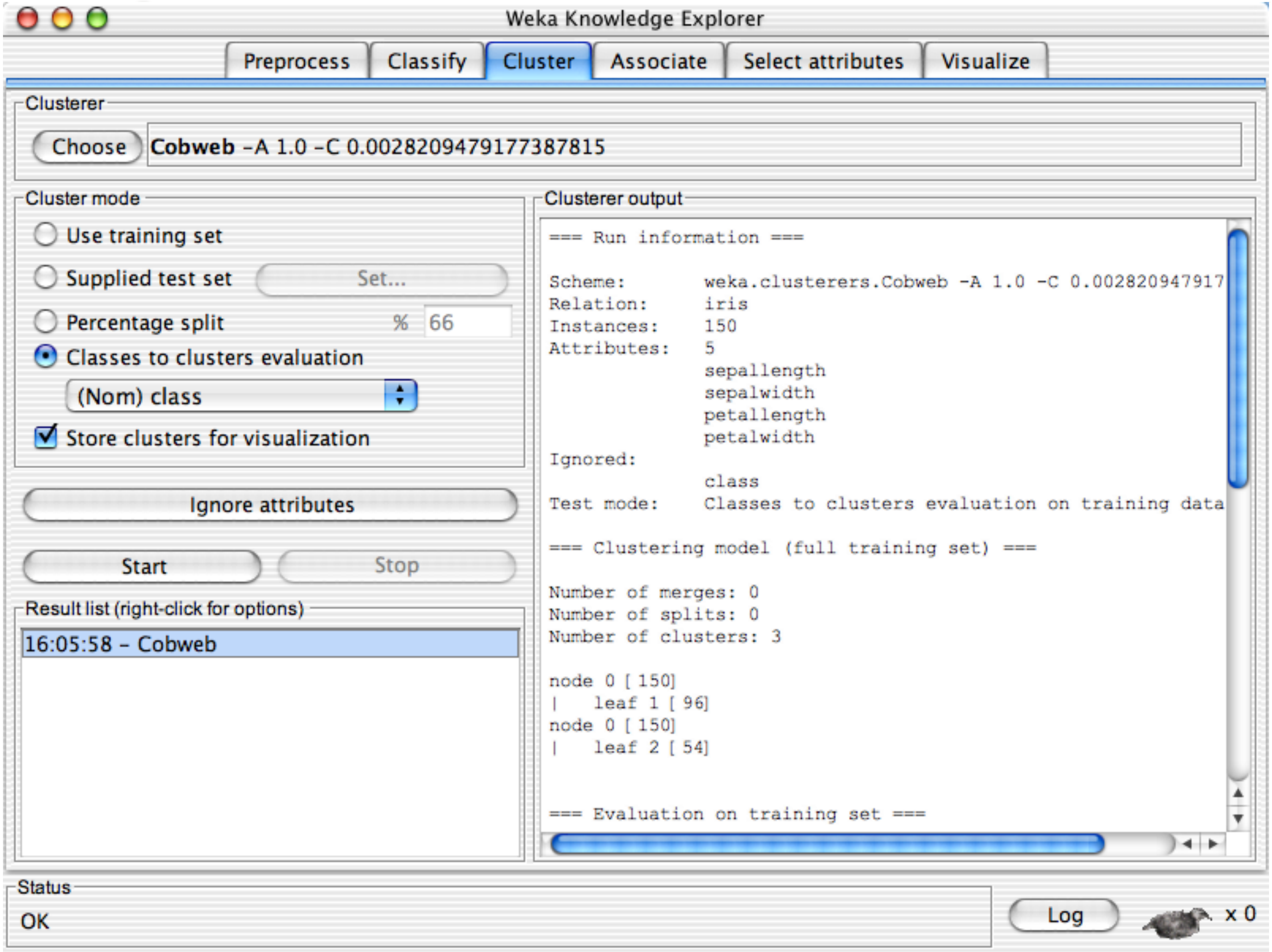

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
% 66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
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 ===
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
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
(Nom) class
Start
Result list (right-click for
11:49:05 - trees.j48.J
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)
96.0784 %
3.9216 %
0 19 0 | b = Iris-versicolor
0 2 15 | c = Iris-virginica
Status
OK
Log
x 0

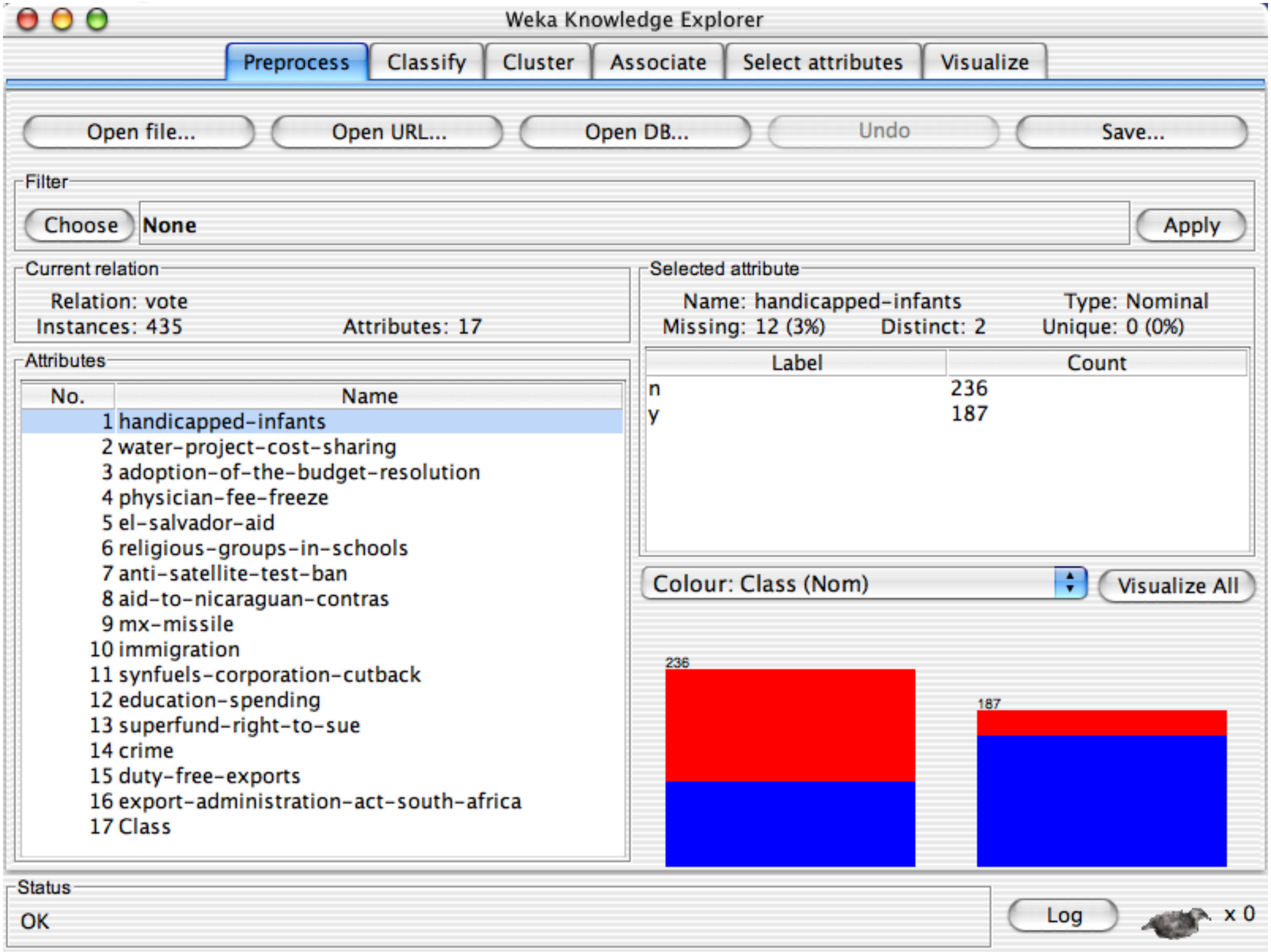
Explorer: clustering data

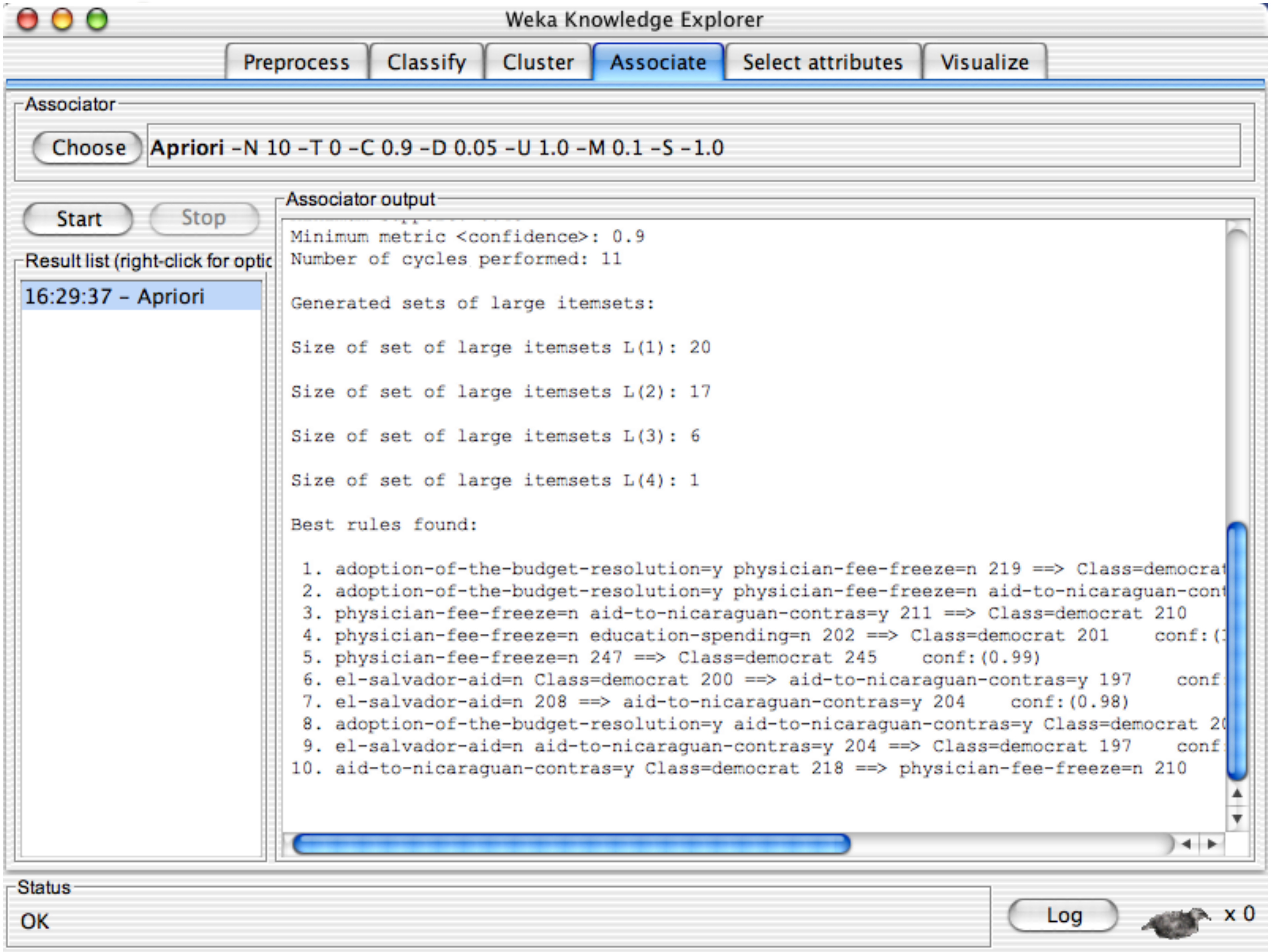
- WEKA contains "clusterers" for finding groups of similar instances in a dataset
- Implemented schemes are:
  - *k*-Means, EM, Cobweb, *X*-means, FarthestFirst
- Clusters can be visualized and compared to "true" clusters (if given)
- Evaluation based on loglikelihood if clustering scheme produces a probability distribution


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
Cobweb -A 1.0 -C 0.0028209479177387815
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)
16:05:58 - Cobweb
Clusterer output
=== Run information ===
Scheme: weka.clusterers.Cobweb -A 1.0 -C 0.002820947917
Relation: iris
Instances: 150
Attributes: 5
sepallength
sepalwidth
petallength
petalwidth
Ignored:
class
Test mode: Classes to clusters evaluation on training data
=== Clustering model (full training set) ===
Number of merges: 0
Number of splits: 0
Number of clusters: 3
node 0 [150]
| leaf 1 [96]
node 0 [150]
| leaf 2 [54]
=== Evaluation on training set ===
Status
OK
Log
x 0

Explorer: finding associations

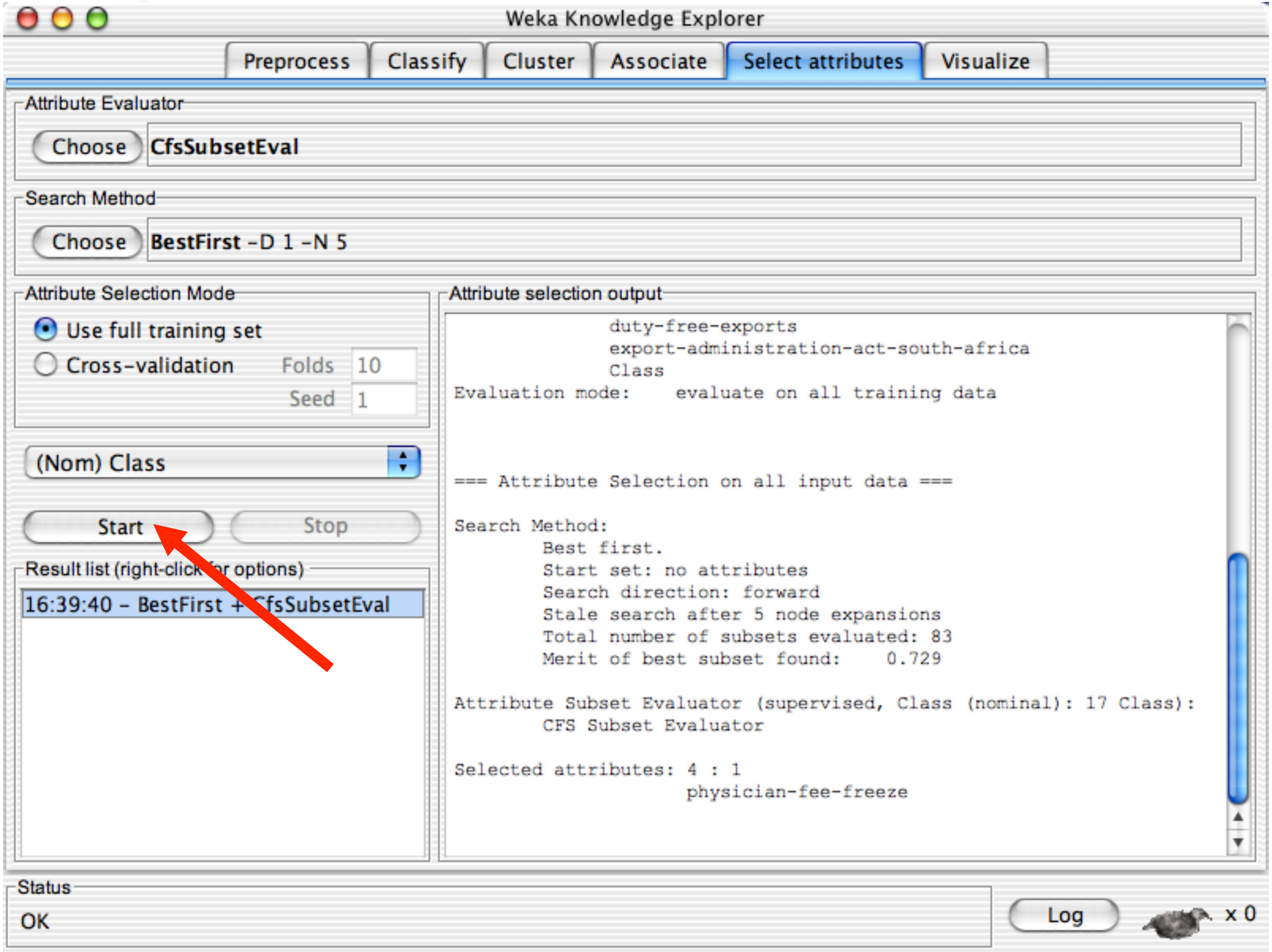
- WEKA contains an implementation of the Apriori algorithm for learning association rules
  - Works only with discrete data
- Can identify statistical dependencies between groups of attributes:
  - milk, butter $\Rightarrow$ bread, eggs (with confidence 0.9 and support 2000)
- Apriori can compute all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

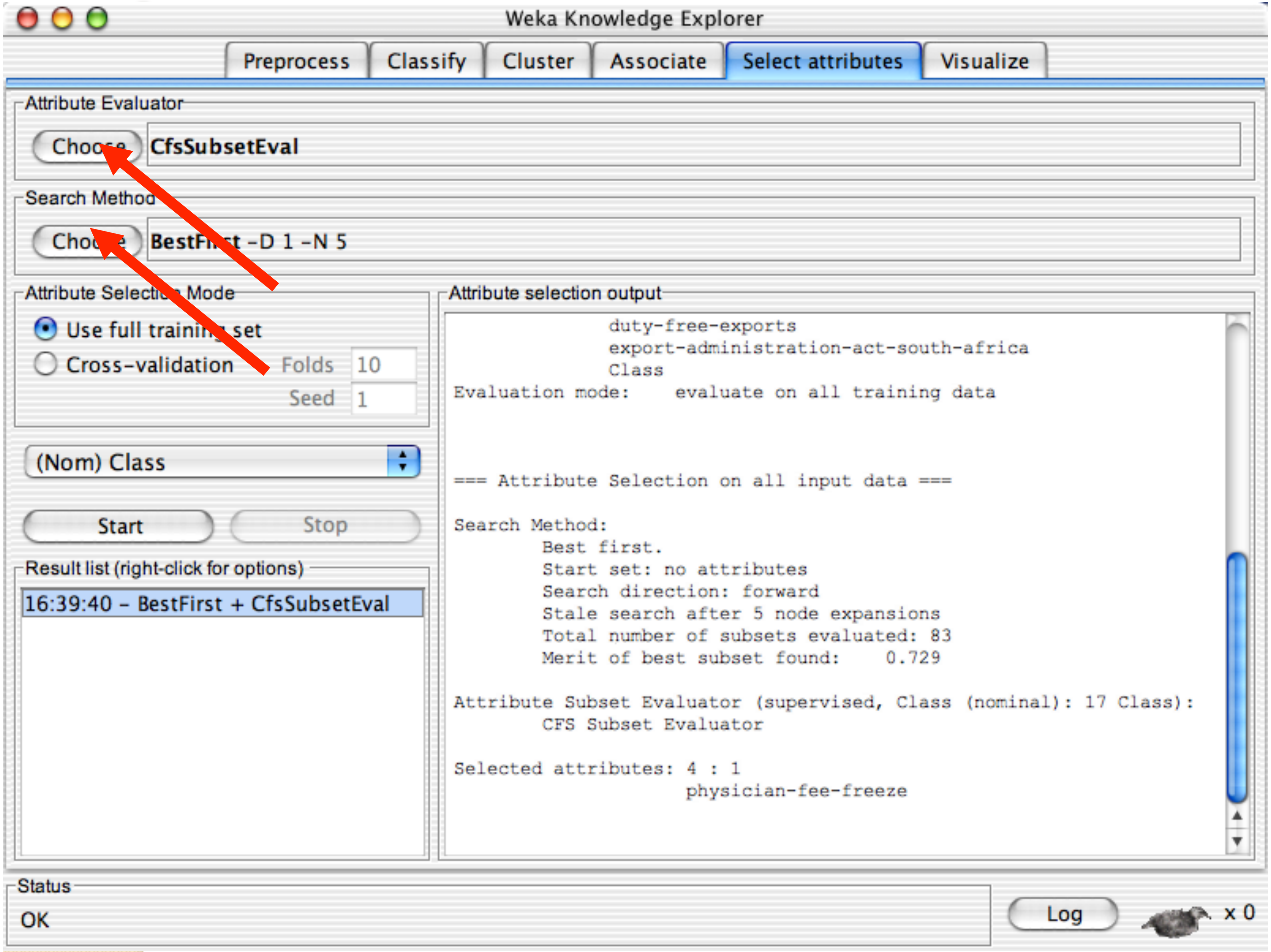

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Undo
Save...
Filter
Choose
None
Apply
Current relation
Relation: vote
Instances: 435
Attributes: 17
Attributes
No.
Name
1 handicapped-infants
2 water-project-cost-sharing
3 adoption-of-the-budget-resolution
4 physician-fee-freeze
5 el-salvador-aid
6 religious-groups-in-schools
7 anti-satellite-test-ban
8 aid-to-nicaraguan-contras
9 mx-missile
10 immigration
11 synfuels-corporation-cutback
12 education-spending
13 superfund-right-to-sue
14 crime
15 duty-free-exports
16 export-administration-act-south-africa
17 Class
Selected attribute
Name: handicapped-infants
Type: Nominal
Missing: 12 (3%)
Distinct: 2
Unique: 0 (0%)
Label
Count
n
236
y
187
Colour: Class (Nom)
Visualize All
236
187
Status
OK
Log
x 0

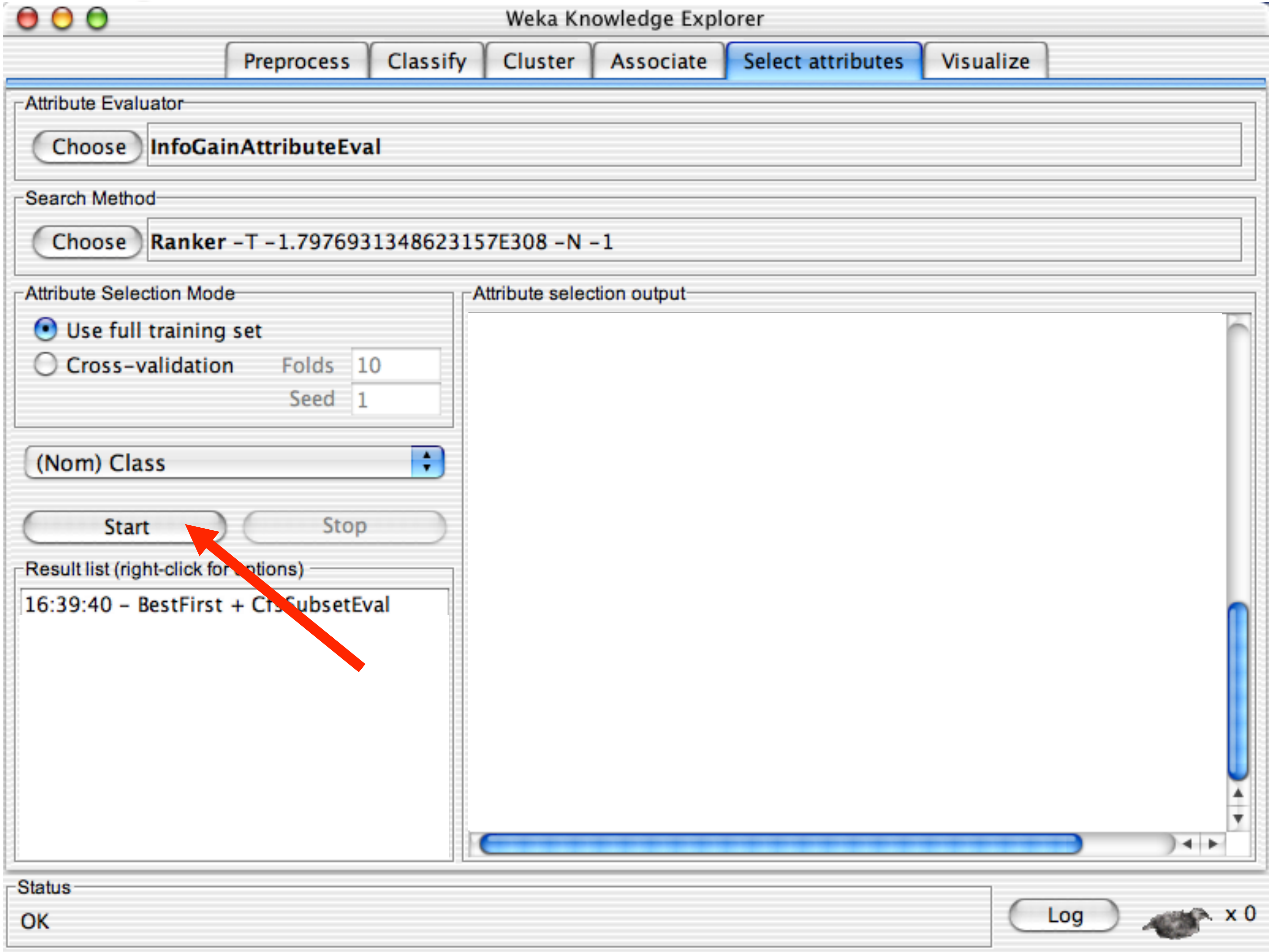

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Associator
Choose
Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0
Start
Stop
Result list (right-click for optic
16:29:37 - Apriori
Associator output
Minimum metric <confidence>: 0.9
Number of cycles performed: 11
Generated sets of large itemsets:
Size of set of large itemsets L(1): 20
Size of set of large itemsets L(2): 17
Size of set of large itemsets L(3): 6
Size of set of large itemsets L(4): 1
Best rules found:
1. adoption-of-the-budget-resolution=y physician-fee-freeze=n 219 ==> Class=democrat
2. adoption-of-the-budget-resolution=y physician-fee-freeze=n aid-to-nicaraguan-cont
3. physician-fee-freeze=n aid-to-nicaraguan-contras=y 211 ==> Class=democrat 210
4. physician-fee-freeze=n education-spending=n 202 ==> Class=democrat 201 conf:(
5. physician-fee-freeze=n 247 ==> Class=democrat 245 conf:(0.99)
6. el-salvador-aid=n Class=democrat 200 ==> aid-to-nicaraguan-contras=y 197 conf
7. el-salvador-aid=n 208 ==> aid-to-nicaraguan-contras=y 204 conf:(0.98)
8. adoption-of-the-budget-resolution=y aid-to-nicaraguan-contras=y Class=democrat 2
9. el-salvador-aid=n aid-to-nicaraguan-contras=y 204 ==> Class=democrat 197 conf
10. aid-to-nicaraguan-contras=y Class=democrat 218 ==> physician-fee-freeze=n 210
Status
OK
Log
x 0

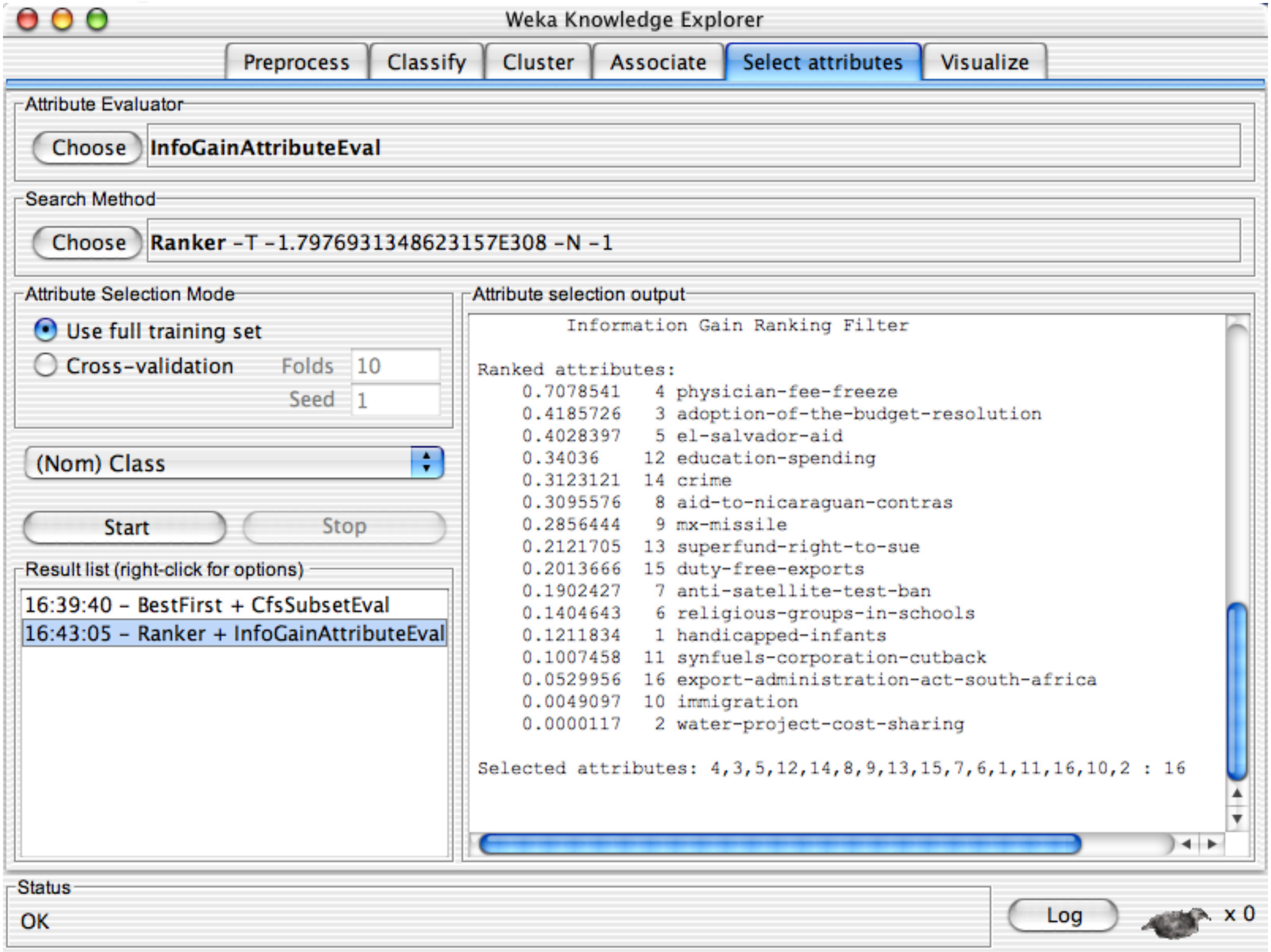
Explorer: attribute selection

- 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


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
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)
16:39:40 - BestFirst + CfsSubsetEval
Attribute selection output
duty-free-exports
export-administration-act-south-africa
Class
Evaluation mode: evaluate on all training data
=== Attribute Selection on all input data ===
Search Method:
Best first.
Start set: no attributes
Search direction: forward
Stale search after 5 node expansions
Total number of subsets evaluated: 83
Merit of best subset found: 0.729
Attribute Subset Evaluator (supervised, Class (nominal): 17 Class):
CFS Subset Evaluator
Selected attributes: 4 : 1
physician-fee-freeze
Status
OK
Log
x 0


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
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)
16:39:40 - BestFirst + CfsSubsetEval
Attribute selection output
duty-free-exports
export-administration-act-south-africa
Class
Evaluation mode: evaluate on all training data
=== Attribute Selection on all input data ===
Search Method:
Best first.
Start set: no attributes
Search direction: forward
Stale search after 5 node expansions
Total number of subsets evaluated: 83
Merit of best subset found: 0.729
Attribute Subset Evaluator (supervised, Class (nominal): 17 Class):
CFS Subset Evaluator
Selected attributes: 4 : 1
physician-fee-freeze
Status
OK
Log
x 0


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Attribute Evaluator
Choose
InfoGainAttributeEval
Search Method
Choose
Ranker -T -1.7976931348623157E308 -N -1
Attribute Selection Mode
Use full training set
Cross-validation
Folds
10
Seed
1
(Nom) Class
Start
Stop
Result list (right-click for options)
16:39:40 - BestFirst + CfsSubsetEval
Attribute selection output
Status
OK
Log
x 0


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Attribute Evaluator
Choose
InfoGainAttributeEval
Search Method
Choose
Ranker -T -1.7976931348623157E308 -N -1
Attribute Selection Mode
Use full training set
Cross-validation
Folds 10
Seed 1
(Nom) Class
Start
Stop
Result list (right-click for options)
16:39:40 - BestFirst + CfsSubsetEval
16:43:05 - Ranker + InfoGainAttributeEval
Attribute selection output
Information Gain Ranking Filter
Ranked attributes:
0.7078541 4 physician-fee-freeze
0.4185726 3 adoption-of-the-budget-resolution
0.4028397 5 el-salvador-aid
0.34036 12 education-spending
0.3123121 14 crime
0.3095576 8 aid-to-nicaraguan-contras
0.2856444 9 mx-missile
0.2121705 13 superfund-right-to-sue
0.2013666 15 duty-free-exports
0.1902427 7 anti-satellite-test-ban
0.1404643 6 religious-groups-in-schools
0.1211834 1 handicapped-infants
0.1007458 11 synfuels-corporation-cutback
0.0529956 16 export-administration-act-south-africa
0.0049097 10 immigration
0.0000117 2 water-project-cost-sharing
Selected attributes: 4,3,5,12,14,8,9,13,15,7,6,1,11,16,10,2 : 16
Status
OK
Log
x 0

Which attribute selector?

- Best: WRAPPER
  - Slow: O(2^N) search through all attribute combinations
  - The "wrapped" learner called to assess each combination
  - Some heuristics to prune the search; but does not scale

- If not WRAPPER
  - Use InfoGain / OneR for very big datasets
  - Use CFS otherwise

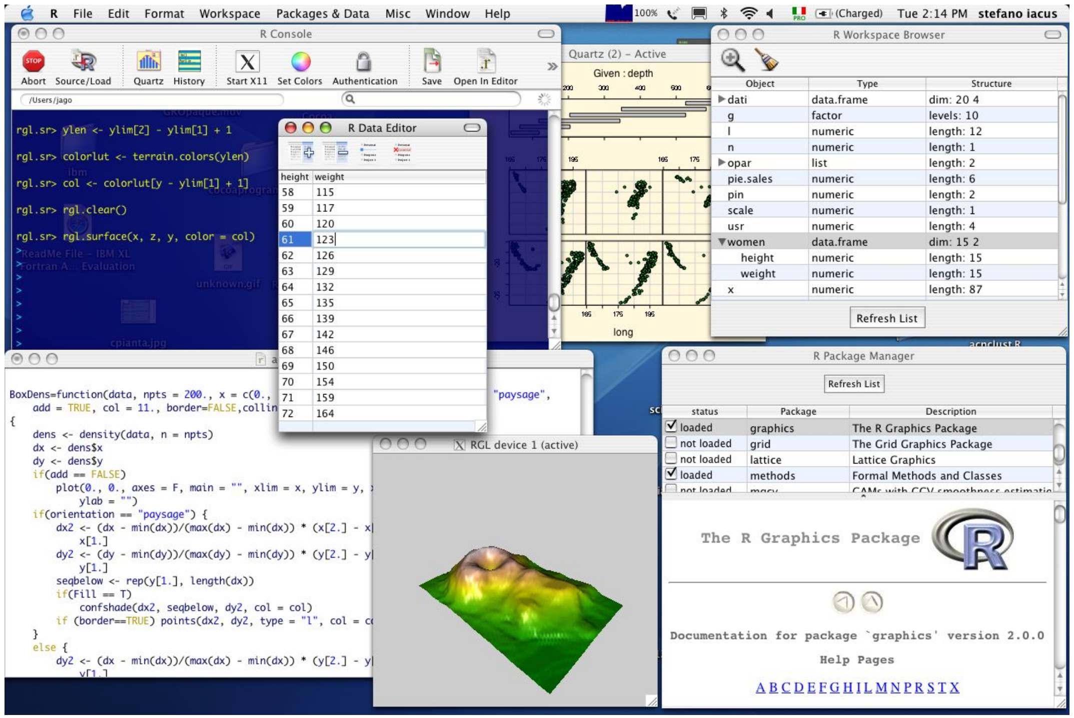
- Don't use PCA
  - This is an unsupervised selector
  - So it is uninformed on how dimensions help classification

Limitations

- Loads all data into ram prior to learning
  - Problem for large data sets
- Not good for complex experiments
- IMHO, discourages experimentation with new learners
  - The "WEKA effect"
    - Try every learner till something works
- Still, very useful for
  - Initial investigations
  - Learning data mining
  - Or as a sub-routine of other systems

Alternate tools:"R"

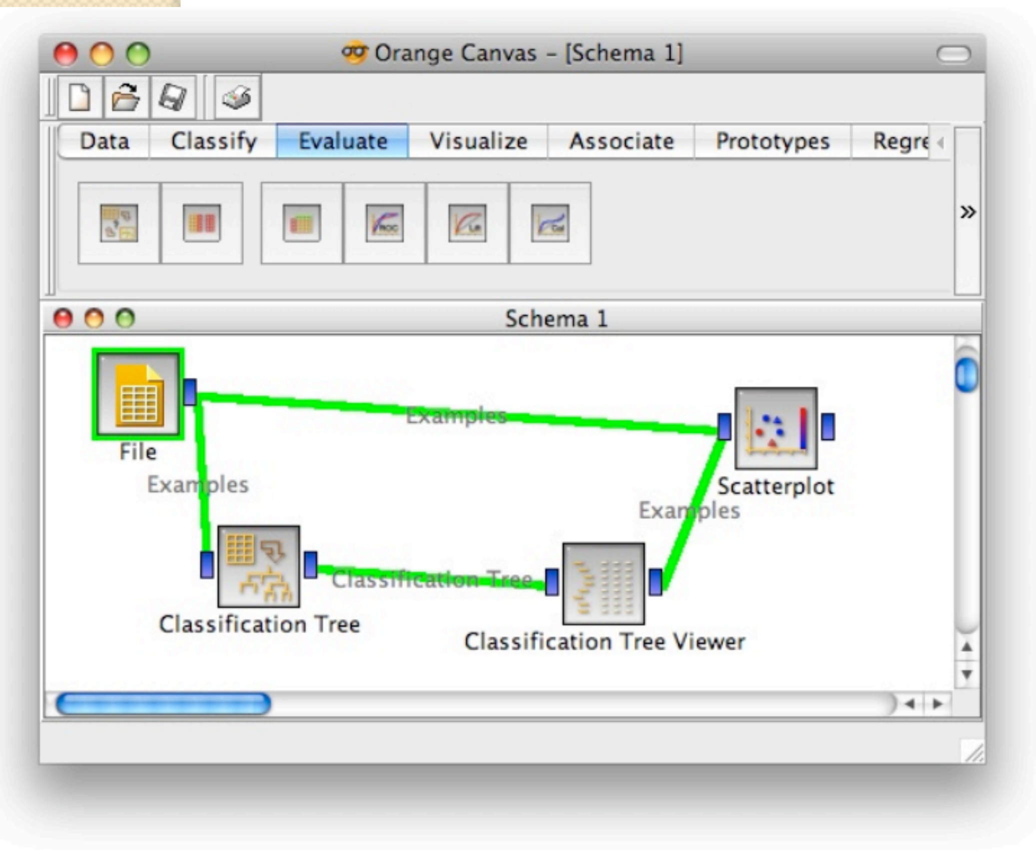
- Leading open-source system for statistical computing and graphics,
- http://www.r-project.org/



Alternate tools: Matlab

- For me: just say no
- Open science, open tools

Alternate tools: Orange

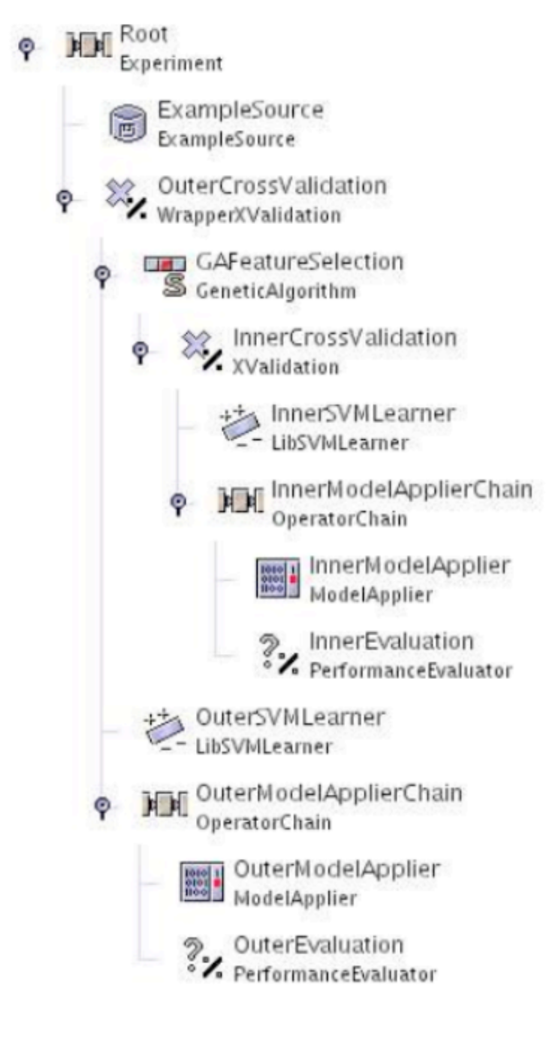


Written in Python

Simpler specification (but see WEKA's KnowledgeFlow Environment).

Also, less community support/debugging. So sometimes frustrated by random bugs

Alternate tools: RapidMiner



Experiments specified in an XML tree syntax

In theory, possible to share experimental descriptions

Alternate tools: OurMine

```
Java=$Base/lib/java
Weka="java -Xmx2048M -cp $Java/weka.jar "
Clusterers="java -Xmx1024M -jar $Java/Clusterers.jar "
Reducers="java -Xmx1024M -jar $Java/Reduce.jar "

nb() {
   local learner=weka.classifiers.bayes.NaiveBayes
   $Weka $learner -p 0 -t $1 -T $2
}

nb10() {
   local learner=weka.classifiers.bayes.NaiveBayes
   $Weka $learner -i -t $1
}

j48() {
   local learner=weka.classifiers.trees.J48
   $Weka $learner -p 0 -C 0.25 -M 2 -t $1 -T $2
}
```

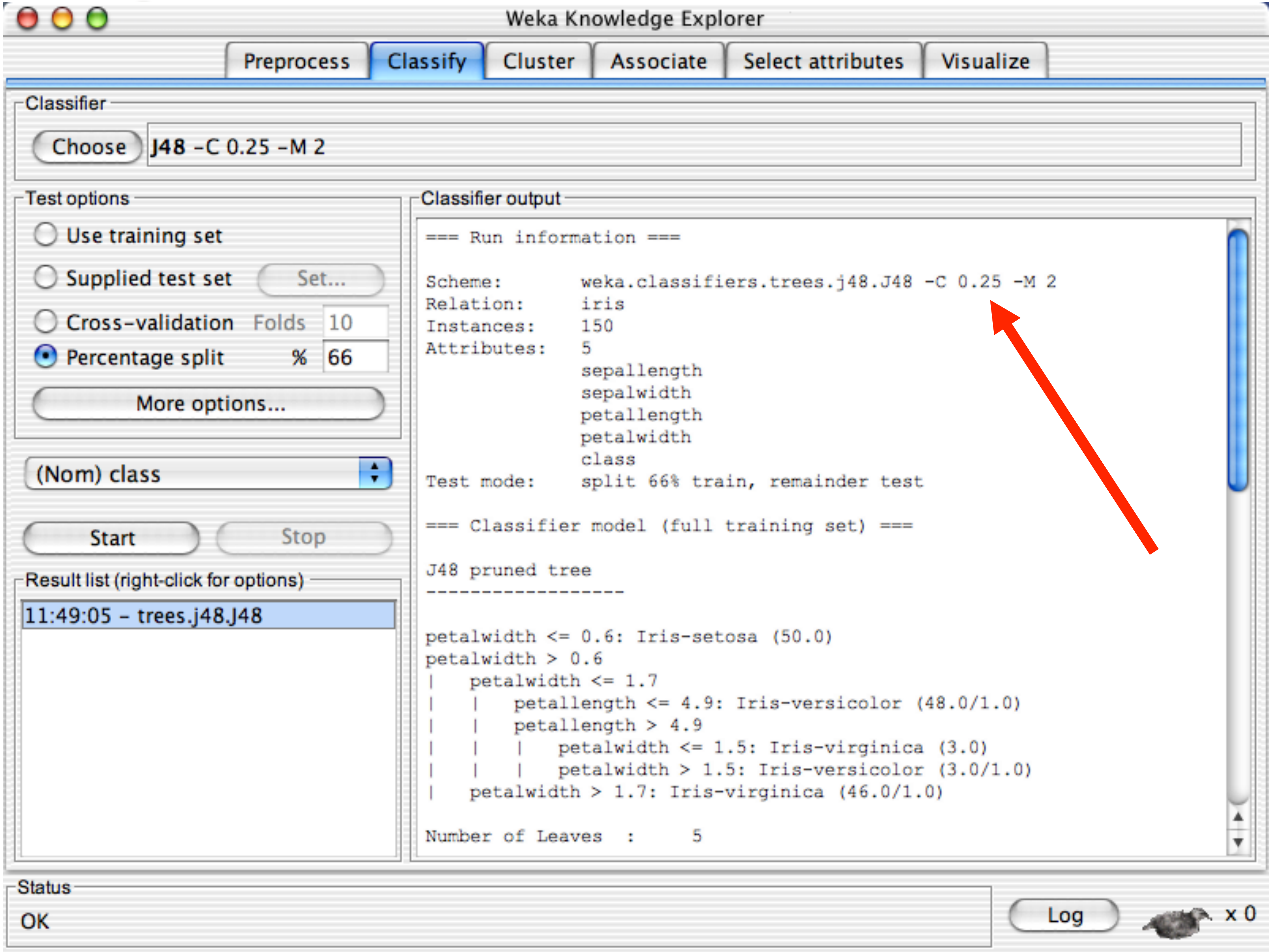
Adam Nelson, Tim Menzies, Gregory Gay,
Sharing Experiments Using Open Source Software,
Softw. Pract. Exper. 2011

Forget the visuals.

Make WEKA a sub-routine
inside Bash script

Now you can mix WEKA's JAVA with learners written in your
favorite language.

But how do you find the magic
command strings?


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

Why go to all that trouble?

```
analysis1(){
   local origdata=$1
   local outstats=$2
   local nattrs="2 4 6 8 10 12 14 16 18 20"
   local learners="nb10 j4810 zeror10 oner10 adtree10"
   local reducers="infogain chisquared oneR"
   local tmpred=$Tmp/red
   echo "n,reducer,learner,accuracy" > $outstats

   for n in $nattrs; do
      for reducer in $reducers; do
           $reducer $origdata $n $tmpred
           for learner in $learners; do
              accur=`$learner $tmpred.arff | acc
              out="$n,$reducer,$learner,$accur"
              blabln $out
             echo $out >> $outstats
           done
      done
   done
}
```

Complex experiments, specified succinctly.

Experiments can now be reviewed, audited, by others.

Also, in 12 months time when Reviewer2 wants a tiny extension to the old paper, you don't have to remember all that clicking you did: just rerun the script.

Coming next…

- Enough details
- So many tools in WEKA, R, Rapid-Miner, Orange, OURMINE…
- The great secret
  - All those "different" tools do the same thing.
    - Carve up vector space.

DATA CARVING (THE CORE OPERATORS OF DM)

Road map

1. Data mining & SE (overview)
2. Data mining tools (guided tour of "WEKA")
3. Data "carving" (core operators of DM)

“Data Carving”: A geometric view of data mining

- Data is like a block of marble,
  - waiting for a sculptor (that’s you)
  - to find the shape within

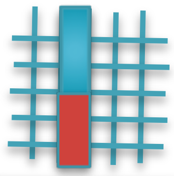
- So “data mining” is really “data carving”
  - chipping away the irrelevancies
  - To find what lies beneath.

Four operators of data carving

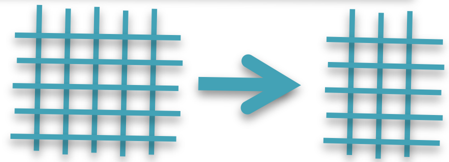
```
@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}
@data
sunny,      85,85,FALSE,  no
sunny,      80,90,TRUE,   no
overcast,   83,86,FALSE,  yes
rainy,      70,96,FALSE,  no
rainy,      68,80,FALSE,  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
```

- Each example is a row in a table
- What can can we do change the table geometry?

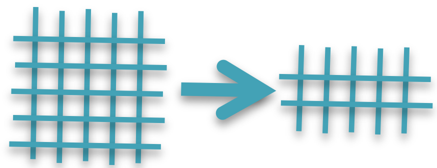
1. Clump



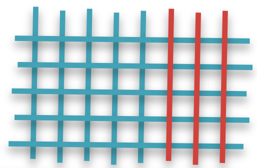
2. Select columns



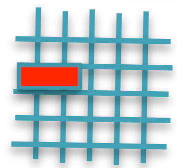
3. Select rows



4. Rotate (add new columns)



5. Clump columns



The field is called "data mining", not "algorithm mining"

- To understand data mining, look at the <u>data</u>, not the <u>algorithms</u>

- Why? We do data mining not to study algorithms.
  - But to study data

- Our results should be insights about data,
  - not trivia about (say) decision tree algorithms

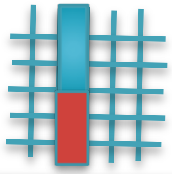
- Besides, the thing that most predicts for performance is the data, not the algorithm,
  - Pedro Domingos and Michael J. Pazzani, On the Optimality of the Simple Bayesian Classifier under Zero-One Loss, Machine Learning, Volume 29, number 2-3, pages 103-130, 1997

Table 1. Classification accuracies and sample standard deviations, averaged over 20 random training/test splits. "Bayes" is the Bayesian classifier with discretization and "Gauss" is the Bayesian classifier with Gaussian distributions. Superscripts denote confidence levels for the difference in accuracy between the Bayesian classifier and the corresponding algorithm, using a one-tailed paired t test: 1 is 99.5%, 2 is 99%, 3 is 97.5%, 4 is 95%, 5 is 90%, and 6 is below 90%.

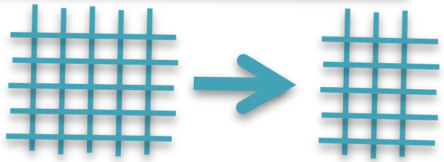
Data Set	Bayes	Gauss	C4.5	PEBLS	CN2	Def.
Audiology	73.0 ± 6.1	73.0 ± 6.1^{6}	72.5 ± 5.8^{6}	75.8 ± 5.4^{3}	71.0 ± 5.1^{5}	21.3
Annealing	95.3 ± 1.2	84.3 ± 3.8^{1}	90.5 ± 2.2^{1}	98.8 ± 0.8^{1}	81.2 ± 5.4^{1}	76.4
Breast cancer	71.6 ± 4.7	71.3 ± 4.3^{6}	70.1 ± 6.8^{5}	65.6 ± 4.7^{1}	67.9 ± 7.1^{1}	67.6
Credit	84.5 ± 1.8	78.9 ± 2.5^{1}	85.9 ± 2.1^{3}	82.2 ± 1.9^{1}	82.0 ± 2.2^{1}	57.4
Chess endgames	88.0 ± 1.4	88.0 ± 1.4^{6}	99.2 ± 0.1^{1}	96.9 ± 0.7^{1}	98.1 ± 1.0^{1}	52.0
Diabetes	74.5 ± 2.4	75.2 ± 2.1^{6}	73.5 ± 3.4^{5}	71.1 ± 2.4^{1}	73.8 ± 2.7^{6}	66.0
Echocardiogram	69.1 ± 5.4	73.4 ± 4.9^{1}	64.7 ± 6.3^{1}	61.7 ± 6.4^{1}	68.2 ± 7.2^{6}	67.8
Glass	61.9 ± 6.2	50.6 ± 8.2^{1}	63.9 ± 8.7^{6}	62.0 ± 7.4^{6}	63.8 ± 5.5^{6}	31.7
Heart disease	81.9 ± 3.4	84.1 ± 2.8^{1}	77.5 ± 4.3^{1}	78.9 ± 4.0^{1}	79.7 ± 2.9^{3}	55.0
Hepatitis	85.3 ± 3.7	85.2 ± 4.0^{6}	79.2 ± 4.3^{1}	79.0 ± 5.1^{1}	80.3 ± 4.2^{1}	78.1
Horse colic	80.7 ± 3.7	79.3 ± 3.7^{1}	85.1 ± 3.8^{1}	75.7 ± 5.0^{1}	82.5 ± 4.2^{2}	63.6
Hypothyroid	97.5 ± 0.3	97.9 ± 0.4^{1}	99.1 ± 0.2^{1}	95.9 ± 0.7^{1}	98.8 ± 0.4^{1}	95.3
Iris	93.2 ± 3.5	93.9 ± 1.9^{6}	92.6 ± 2.7^{6}	93.5 ± 3.0^{6}	93.3 ± 3.6^{6}	26.5
Labor	91.3 ± 4.9	88.7 ± 10.6^{6}	78.1 ± 7.9^{1}	89.7 ± 5.0^{6}	82.1 ± 6.9^{1}	65.0
Lung cancer	46.8 ± 13.3	46.8 ± 13.3^{6}	40.9 ± 16.3^{5}	42.3 ± 17.3^{6}	38.6 ± 13.5^{3}	26.8
Liver disease	63.0 ± 3.3	54.8 ± 5.5^{1}	65.9 ± 4.4^{1}	61.3 ± 4.3^{6}	65.0 ± 3.8^{3}	58.1
LED	62.9 ± 6.5	62.9 ± 6.5^{6}	61.2 ± 8.4^{6}	55.3 ± 6.1^{1}	58.6 ± 8.1^{2}	8.0
Lymphography	81.6 ± 5.9	81.1 ± 4.8^{6}	75.0 ± 4.2^{1}	82.9 ± 5.6^{6}	78.8 ± 4.9^{3}	57.3
Post-operative	64.7 ± 6.8	67.2 ± 5.0^{3}	70.0 ± 5.2^{1}	59.2 ± 8.0^{2}	60.8 ± 8.2^{4}	71.2
Promoters	87.9 ± 7.0	87.9 ± 7.0^{6}	74.3 ± 7.8^{1}	91.7 ± 5.9^{3}	75.9 ± 8.8^{1}	43.1
Primary tumor	44.2 ± 5.5	44.2 ± 5.5^{6}	35.9 ± 5.8^{1}	30.9 ± 4.7^{1}	39.8 ± 5.2^{1}	24.6
Solar flare	68.5 ± 3.0	68.2 ± 3.7^{6}	70.6 ± 2.9^{1}	67.6 ± 3.5^{6}	70.4 ± 3.0^{2}	25.2
Sonar	69.4 ± 7.6	63.0 ± 8.3^{1}	69.1 ± 7.4^{6}	73.8 ± 7.4^{1}	66.2 ± 7.5^{5}	50.8
Soybean	100.0 ± 0.0	100.0 ± 0.0^{6}	95.0 ± 9.0^{3}	100.0 ± 0.0^{6}	96.9 ± 5.9^{3}	30.0
Splice junctions	95.4 ± 0.6	95.4 ± 0.6^{6}	93.4 ± 0.8^{1}	94.3 ± 0.5^{1}	81.5 ± 5.5^{1}	52.4
Voting records	91.2 ± 1.7	91.2 ± 1.7^{6}	96.3 ± 1.3^{1}	94.9 ± 1.2^{1}	95.8 ± 1.6^{1}	60.5
Wine	96.4 ± 2.2	97.8 ± 1.2^{3}	92.4 ± 5.6^{1}	97.2 ± 1.8^{6}	90.8 ± 4.7^{1}	36.4
Zoology	94.4 ± 4.1	94.1 ± 3.8^{6}	89.6 ± 4.7^{1}	94.6 ± 4.3^{6}	90.6 ± 5.0^{1}	39.4

The rest of this hour

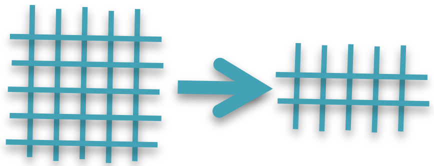
1. Clump



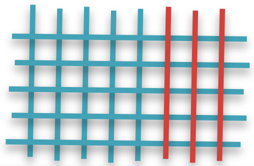
2. Select columns



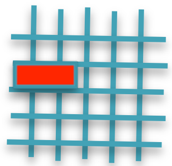
3. Select rows



4. Rotate (add new columns)



5. Clump columns



Carving can be dangerous

- While carving the training data is recommended
  - It is a methodological error to carve the test data
- Whatever is learned from the training data
  - Should be assessed on "raw" (i.e. uncarved) test data

Clumping column data (a.k.a. discretization)

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

- Learning = compression
  - Take a target concept that is spread out across all the data
  - Squeeze it together till it is dense enough to be visible.
- Discretization: clump together observations taken over a continuous range
  - into a small number of regions.
- E.g. "toddlers" If age =1,2,3
- Discretization improves the performance of a learner
  - Gives a learner a smaller space to reason about,
  - With more examples in each part of the space

Supervised Discretization

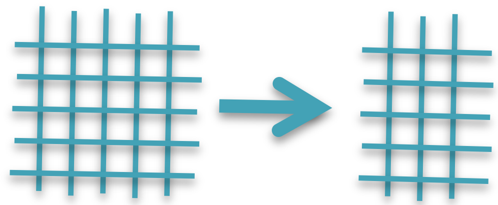
```
overcast,  64,65,TRUE,  yes
Rainy,     65,70,TRUE,  no
sunny,     69,70,FALSE,yes
sunny,     75,70,TRUE,  yes
overcast,  81,75,FALSE,yes
rainy,     68,80,FALSE,yes
rainy,     75,80,FALSE,yes
sunny,     85,85,FALSE,no
overcast,  83,86,FALSE,yes
overcast,  72,90,TRUE,  yes
sunny,     80,90,TRUE,  no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

- Standard method:
  - Find a break that most reduces class diversity either side of the break
  - Recurse on data:
    - above break,
    - below break
  - Fayyad and Irani, Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning IJCAI'93, pp1022-1027

Unsupervised Discretization

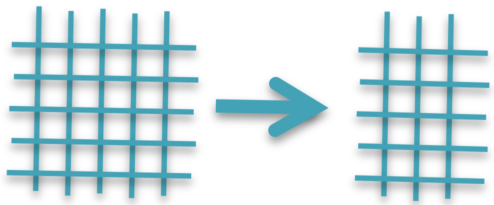
- Divide into "B" bins
  - (X – Min) / ((Max – Min)/ B)
  - B=3 or 10 very common
- Divide into P percentile groups
  - Each bins contains (say) 25% of the rows
- For Bayesian methods
  - Divide into groups of N items
  - Ying and Webb recommends N= sqrt(rows)
  - Ying Yang and Geoff Webb, Weighted Proportional k-Interval Discretization of Naïve Bayes classifeirs, PAKADD'03, p501-512, 2003

Select columns

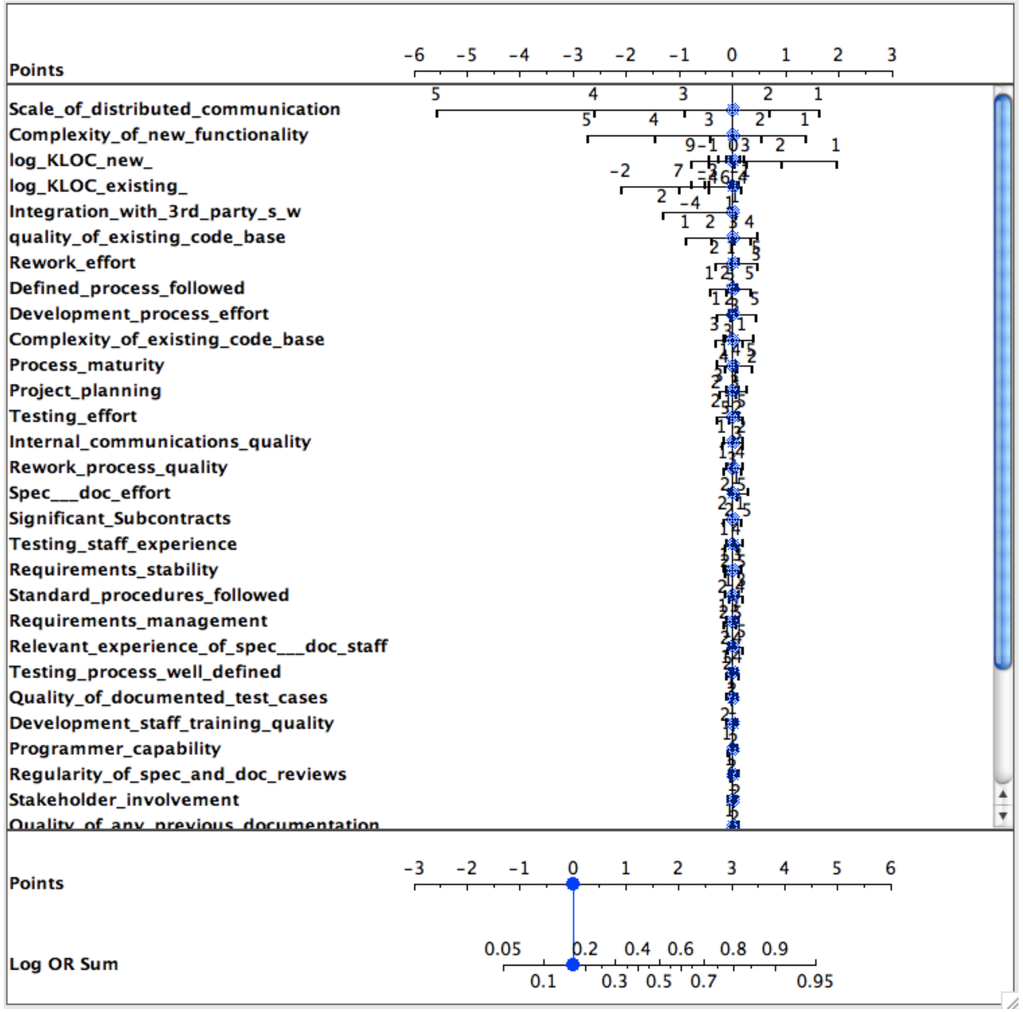


- Occam's Razor - Entia non sunt multiplicanda praeter necessitatem. ("Entities should not be multiplied more than necessary").
  - the fewer features used to explain something, the better
- Log(OR):
  - Discrete every feature. For all pairs of target / other of size C1, C2 count frequency of range N1, N2 in each class
  - Log(odds ratio) = log((N1/C1) / (N2/C2)) > 0 if more frequent in target
  - "Pivots" are the ranges with high Log (OR)
  - Možina, M., Demšar, J., Kattan, M., and Zupan, B. 2004. Nomograms for visualization of naive Bayesian classifier. In*Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (Pisa, Italy, September 20 - 24, 2004)*
- InfoGain:
  - Use Fayyad Irani trick: assses each column by how well it divides up the data
  - Takes linear time : O(C)
- Wrapper:
  - Explore 2^C subsets of C columns: takes time $O(2^C)$
  - Call a learner on each subset
  - Use the columns that maximize learner performance
  - Not practical for large data sets
- For more, see Hall, M. and Holmes, G. (2003). Benchmarking attribute selection techniques for discrete class data mining. IEEE Transactions on Knowledge and Data Engineering. 15(3), November/December 2003

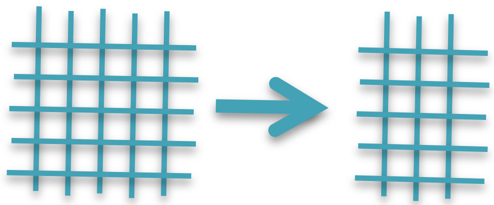
Select columns with log(OR)



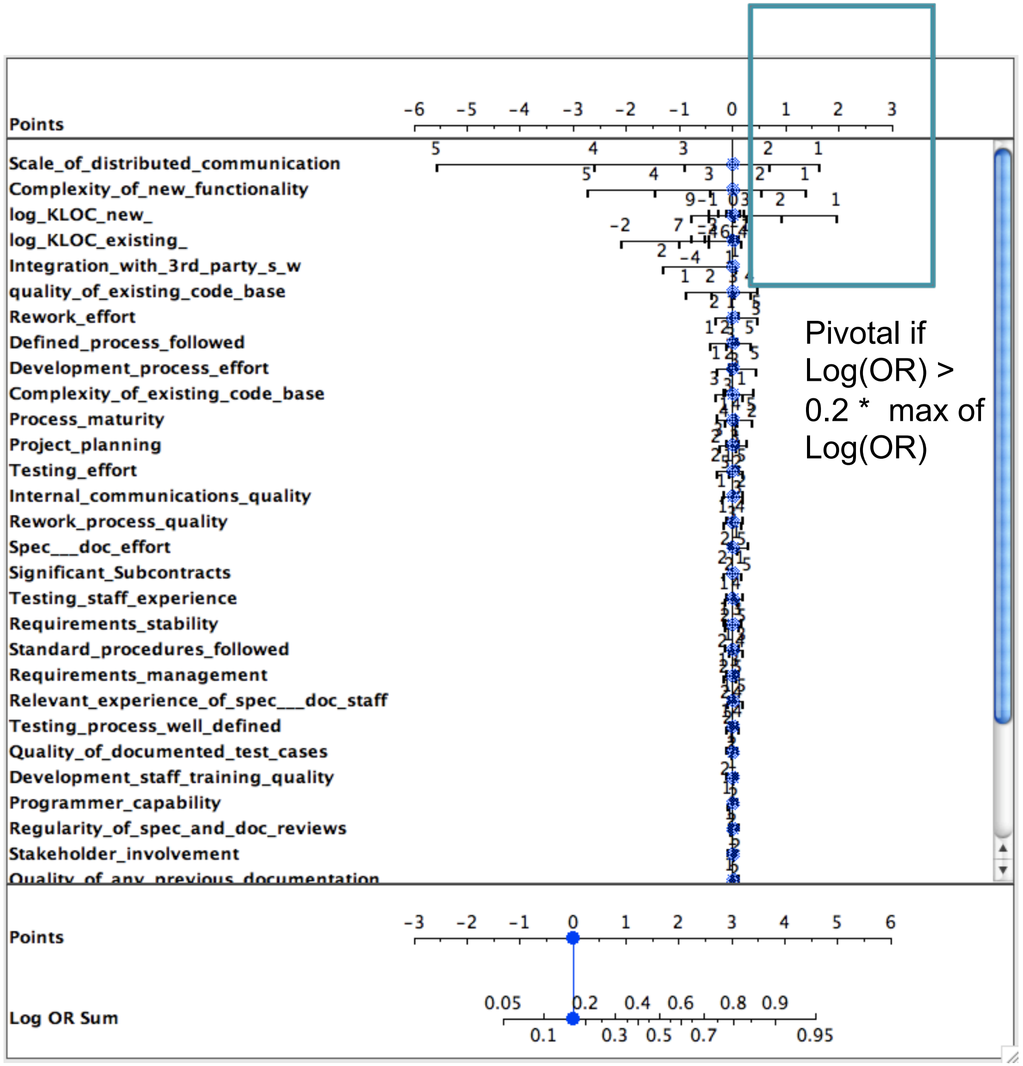
- Data from Norman Fenton's Bayes Net
  - Project Data Incorporating Qualitative Factors for Improved Software Defect Prediction Norman Fenton, Martin Neil, William Marsh, Peter Hearty, Lukasz Radlinski and Paul Krause., PROMISE 2008
- Target class. worse defects
- Only a few features matter
- Only a few ranges of those features matter



Select columns with log(OR)

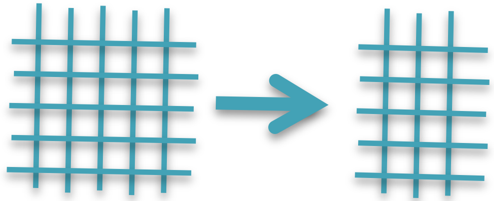


- Data from Norman Fenton's Bayes Net
  - Project Data Incorporating Qualitative Factors for Improved Software Defect Prediction Norman Fenton, Martin Neil, William Marsh, Peter Hearty, Lukasz Radlinski and Paul Krause., PROMISE 2008
- Target class. worse defects
- Only a few features matter
- Only a few ranges of those features matter

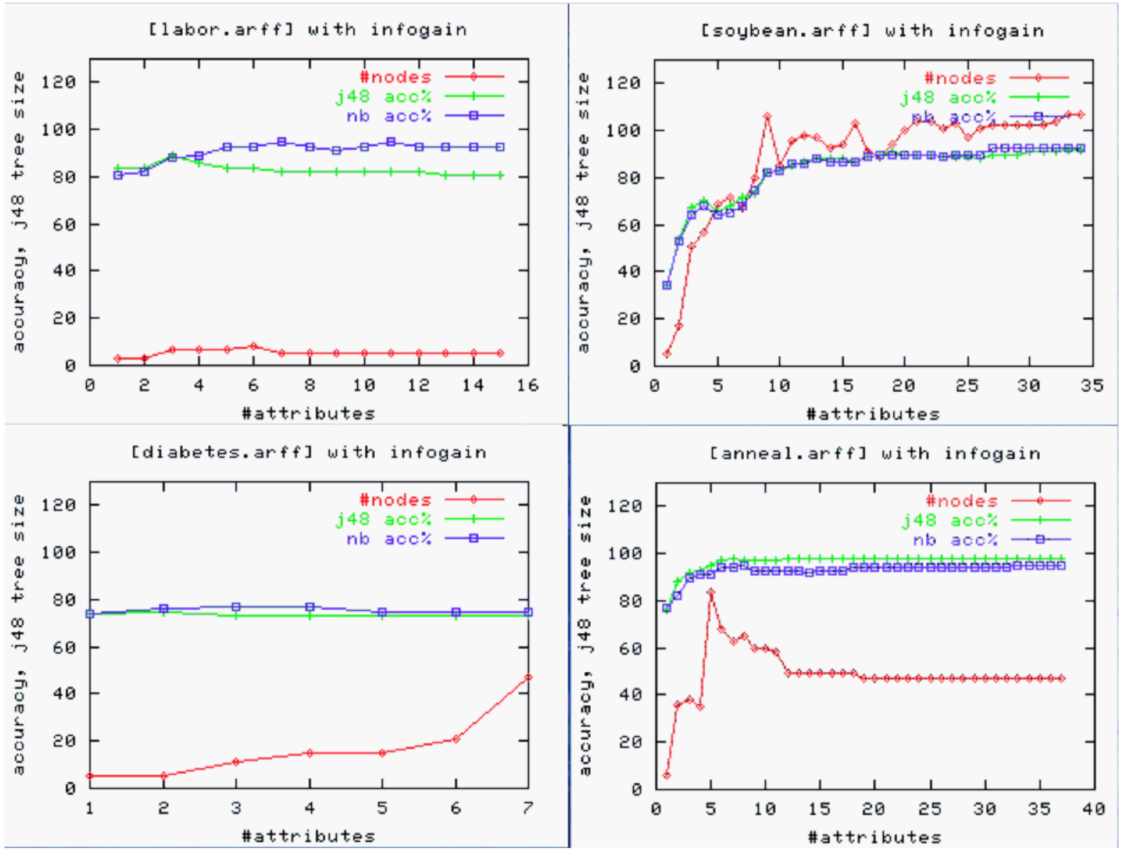


Pivotal if Log(OR) > 0.2 * max of Log(OR)

Select columns with InfoGain

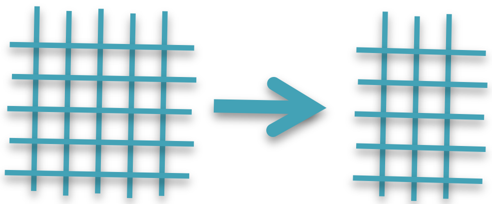


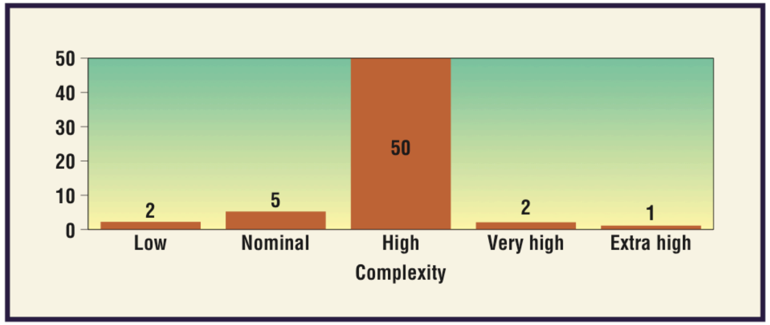
X-axis sorted by sum(-p*log(p))

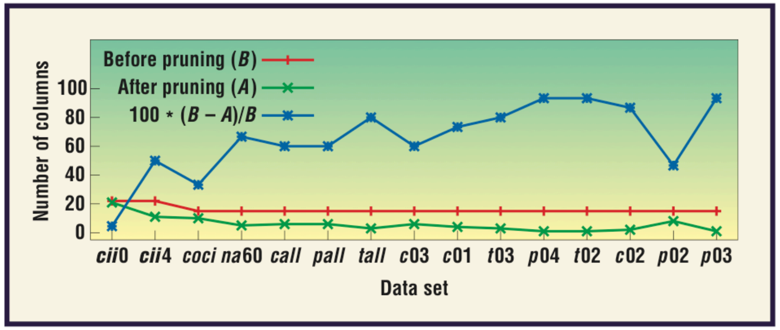


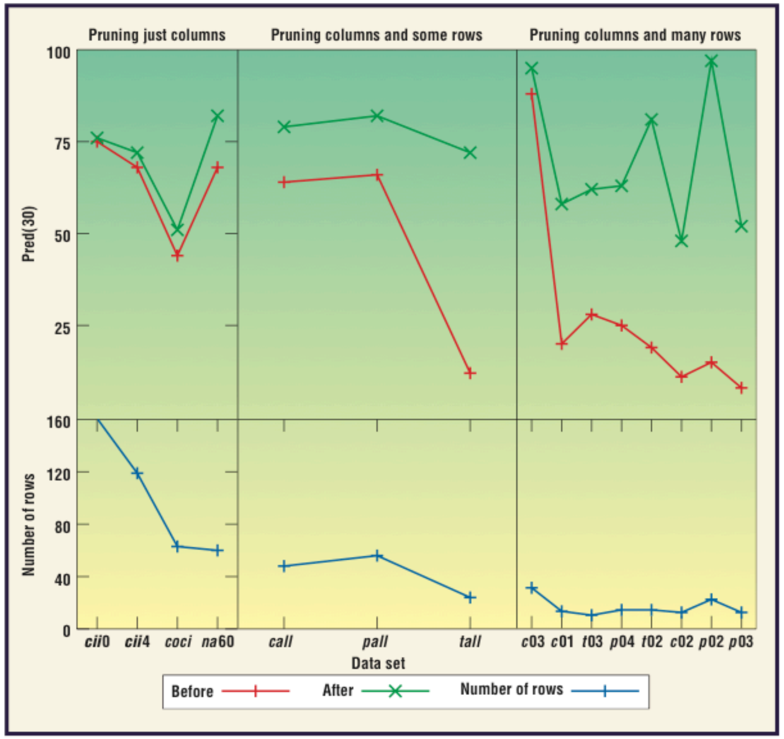
Simpler theories after column selection, work just as well as using everything

Select columns with WRAPPER



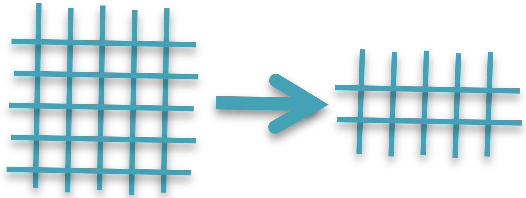






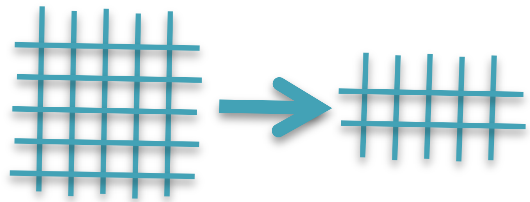
- Finding the Right Data for Software Cost Modeling Chen, Menzies, Port, Boehm, IEEE Software Nov/Dec 2005

Select rows

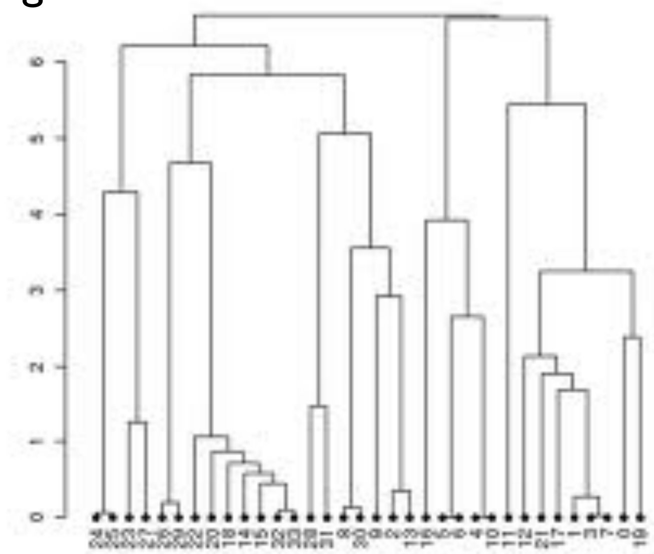


- Replace N rows
  - with M < N rows
  - that best exemplify the data
- Typical result:
  - Can throw out 80 to 90% of the rows without lossing accuracy
  - C. Chang, "Finding prototypes for nearest neighbor classifiers," IEEE Trans. on Computers, pp. 1179–1185, 1974.
- Benefits:
  - Outlier removal
  - Any downstream processing is faster
    - E.g. any $O(N^2)$ process is 100 times faster on N/10 of the data
  - Less errors in conclusions
    - Instance learner: classify according to nearest neighbors
    - If nearest neighbors further away, harder for data collection errors to cause wrong classifications
  - Easier to visualize
    - Fewer things to look at

Select rows

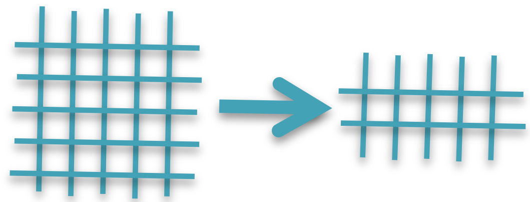


- Exponential time
  - Genetic algorithm to explore the 2^R subsets of rows.
    - When more rows than columns, even slower than the WRAPPER's $O(2^C)$ search
    - Y.Li, M.Xie, and T.Goh, "A study of project selection and feature weighting for analogy based software cost estimation," Journal of Systems and Software, vol. 82, pp. 241–252, 2009.
- Polynomial time: Greedy agglomerative clustering
  - Push every instance to its closest neighbor.
  - Build a synthetic example at each pair's median
  - Repeat for the synthetic points.
  - Prototypes are all nodes at level X of GAC tree
  - For R rows, $O(R^2)$



- TEAK = GAC plus …
  - Prune sub-trees with large variance
  - When to Use Data from Other Projects for Effort Estimation Ekrem Kocaguneli, Gregory Gay, Tim Menzies, Ye Yang, Jacky W. Keung, ASE 2010
- Linear-time
  - Rank ranges by frequency delta in different classes
  - Discard all rows that do not have the top R pivots

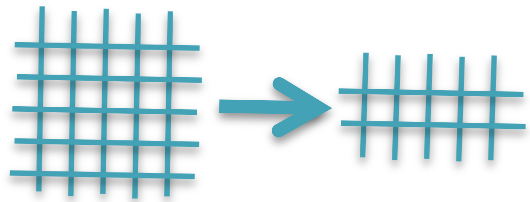
Select rows (with TEAK)



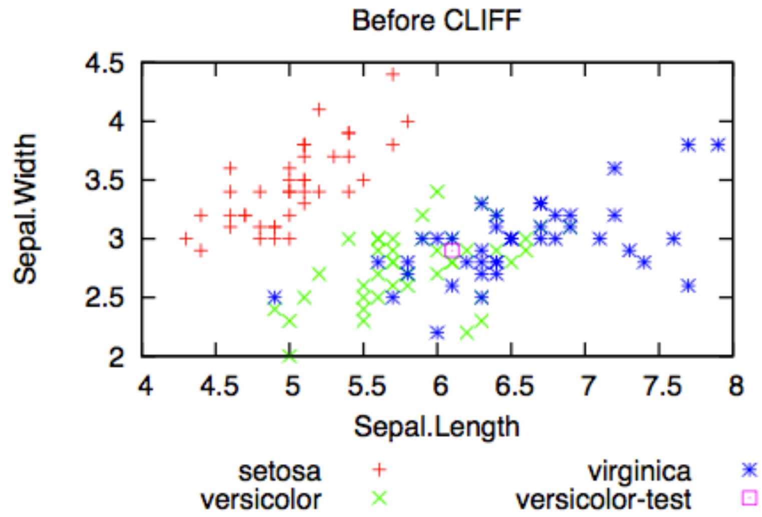
- To effort estimate a test instance, start at root of GAC tree
  - Move to nearest child
  - Stop at leaf or when sub-tree variance greater than super-tree
  - Estimate = median of instances in that sub-tree
- Compared with
  - linear regression,
  - neural nets,
  - analogy methods that use K=1,2,4 nearest neighbors (no variance pruning)
- Compared using
  - 20 * {shuffle rows, 3-way cross-val)
  - #wins - # losses (in a Wilcoxon, 95%)
  - Count number of times ranked first by this procedure
- Conclusion: row-selection using clustering + variance pruning is a good thing

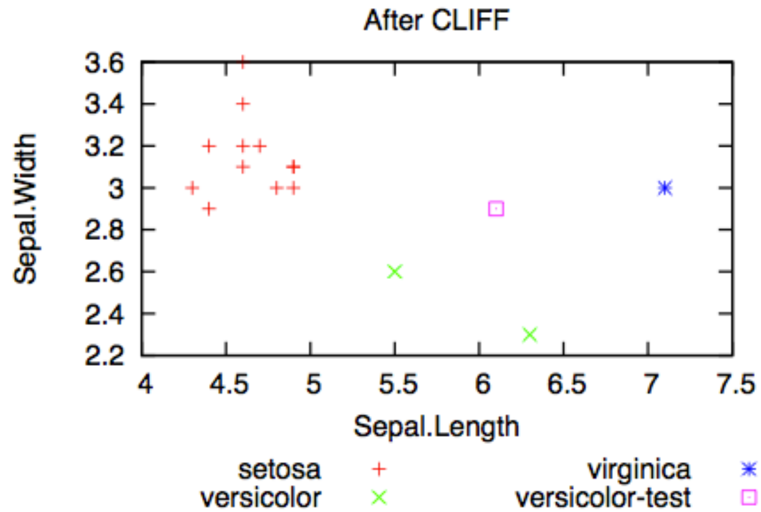
	TEAK	LR	NNet	Best(K)	k=1	k=16	k=2	k=4	k=8
MRE									
Cocomo81	▲								
Cocomo81e	▲								
Cocomo81o	▲								
Nasa93		▲							
Nasa93c2		▲							
Nasa93c5	▲								
Desharnais		▲							
Sdr	▲								
ISBSG-Banking	▲								
Count	6	3	0	0	0	0	0	0	0
Pred(25)									
Cocomo81	▲								
Cocomo81e			▲						
Cocomo81o	▲								
Nasa93		▲							
Nasa93c2		▲							
Nasa93c5	▲								
Desharnais		▲							
Sdr	▲								
ISBSG-Banking	▲								
Count	5	3	1	0	0	0	0	0	0
AR									
Cocomo81				▲					
Cocomo81e	▲								
Cocomo81o	▲								
Nasa93		▲							
Nasa93c2		▲							
Nasa93c5	▲								
Desharnais		▲							
Sdr	▲								
ISBSG-Banking	▲								
Count	6	3	0	1	0	0	0	0	0

Select rows (with range pruning)

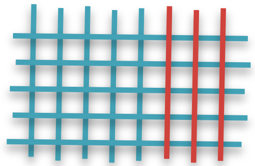


- For class in classes
  - Rank attribute ranges by how often they appear in class vs notClass
- For each attribute
  - Find its range with top rank
- Sort attributes by top ranked range
- Remove rows that do not have the top range of the top N ranked attribute
- Linear time (much faster than any other instance selector)
- ProbDefections not harmed
- As noise added to data, this method's PF changes the least)

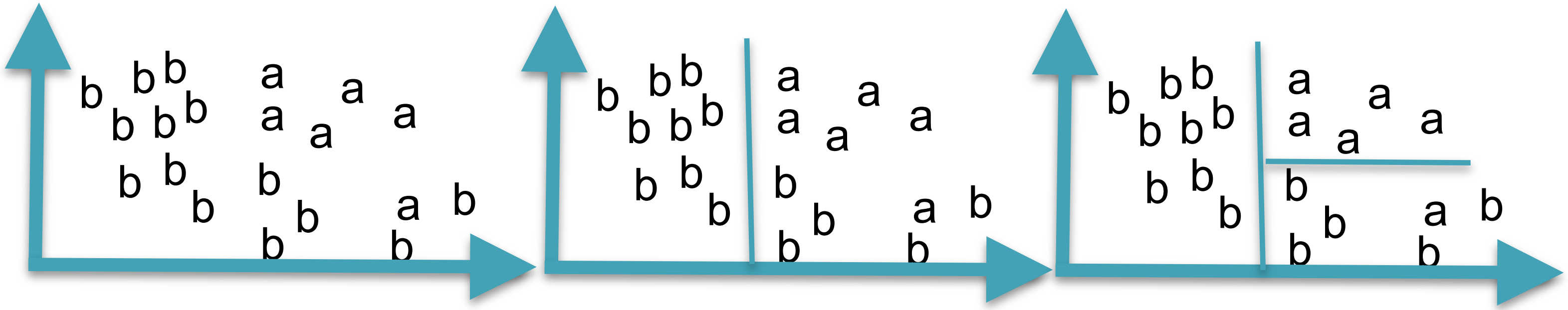




Rotate
(add columns)



- Sometimes, the data's raw dimensions suffice for isolating the target concept..



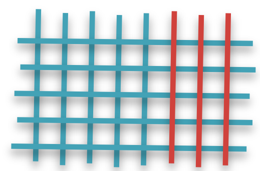
Theory 1:
if true, then "a"

Theory 2::
if x > 1, then "a"

Theory 3::
if x > 1, and y > 1 then "a"

- But what if the target concept falls across, and not along, the raw dimensions?

Rotate (add columns)



- Synthesize a new dimension that combines the raw into something new
- Apply single-valued decomposition (SVD) to
  - the covariance matrix (principal component analysis, or PCA)
  - or the data table (latent semantic indexing, or LSI)
- PCA that produces a set of orthogonal "components"
  - Transforms C correlated variables into fewer uncorrelated "components".
  - Component[i] accounts for as much variability as possible.
  - Component[i+1] accounts for as much of the remaining variability as possible.

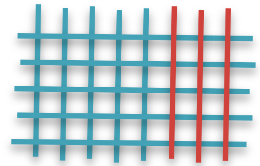
- Much easier to learn rules when dimensions match the data. E.g. a defect predictor:

- if comp[1] ≤ 0.180
  then NoDefects
  else if comp[1] > 0.180
  then if comp[1] ≤ 0.371 then NoDefects
  else if com[1] > 0.371 then Defects

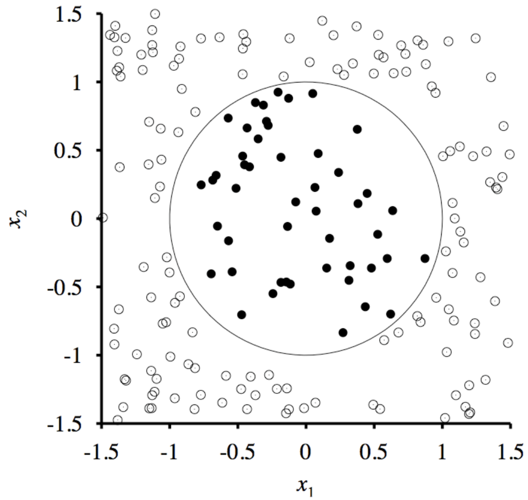
- But it can be hard to explain that predictor:

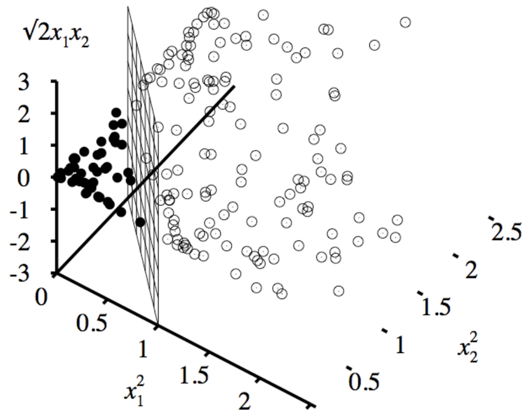
  Comp[1] = 0.236*v(g) +0.222*ev(g)+0.236*iv(g) +0.241*n +0.238*v -0.086*l +0.199*d +0.216*i +0.225*e +0.236*b +0.221*t +0.241*lOCode +0.179*lOComment +0.221*lOBlank +0.158*lOCodeAndComment +0.163*uniq_Op +0.234*uniq_Opnd +0.241*total_Op +0.241*total_Opnd +0.236*branchCount

Rotate (add columns)

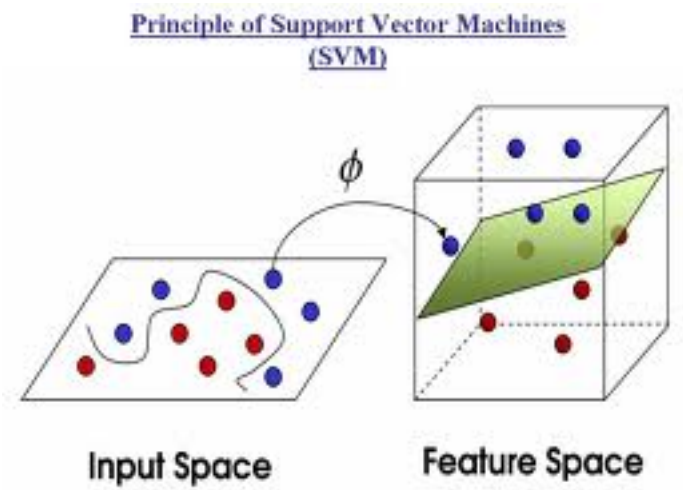


- Special transforms





- Support vector machines: construct a hyper-plane that separates classes



Clump rows (a.k.a. generalize)

- Ever notice that rows and rules have (nearly) the same syntax?
  - Age=young and wealth=rich and iq=high and class=happy
  - If age=old and wealth=rich and iq=high then happy
- But when we write rules, we only do it for frequently occurring patterns in the other rows
- "Clump rows" : replace them with a rule that covers many rows, but many only mention some of the columns
  - **If** age=old and wealth=rich **then** happy
- If you do this after clumping columns and selecting good rows and selecting good columns and (maybe) adding in good columns
  - Then the search space is very small
  - The exploring can be heavily biased by the other steps (e.g. look at great rows before dull ones)
  - And, hey presto, you've got a working data miner

Hints and tips (note: only my view)

- Always try clumping with discretization
  - So very simple
  - So experiment with / without discretization
- Always try column selection
  - Usually, massive reduction in the columns
- If the data won't fit in RAM,
  - try column selection first (use a linear-time approach)
  - then you can explore row selection by (say)
    - Era1: read first 1000 instances and apply row selection
    - Era[i+1]: read next 1000 records and ignore instances that fall close to the instances selected at Era[i]
- Try these last: PCA / Support vector machines
  - Benefits of PCA often achieved, or beaten by other column selectors
    - Hall, M. and Holmes, G. (2003). Benchmarking attribute selection techniques for discrete class data mining. IEEE Trans on Knowledge and Data Engineering. 15(3), November/December 2003
  - The FASTMAP heuristic FASTMAP, can do what PCA does, faster, scalable.
    - Faloutsos, C. and Lin, K. 1995. FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the 1995 ACM SIGMOD international Conference on Management of Data*
  - For text mining (PCA / LDA) vs TF*IDF never benchmarked

tim@menzies.us

WHAT HAVE WE LEARNED?

Fatal flaws in data mining for SE?

- Barbara Kitchenham et al, ESE journal, 2008
  - Replications can replicate stupid errors
- Vic Basili, LASER, 2010
  - If we give people our data, they can make stupid mistakes, cause they don't understand our context
- Well get back to this….

Data mining =
a diverse and lucrative career

- Effort estimation
- Defect prediction
- Optimization of discrete systems
- Test case generation
- Fault localization
- Text mining
- Temporal sequence mining
  - Learning software processes
  - Learning APIs
- Etc

Data mining applications explored by me since 2007.

A career in data mining is a very diverse career, indeed

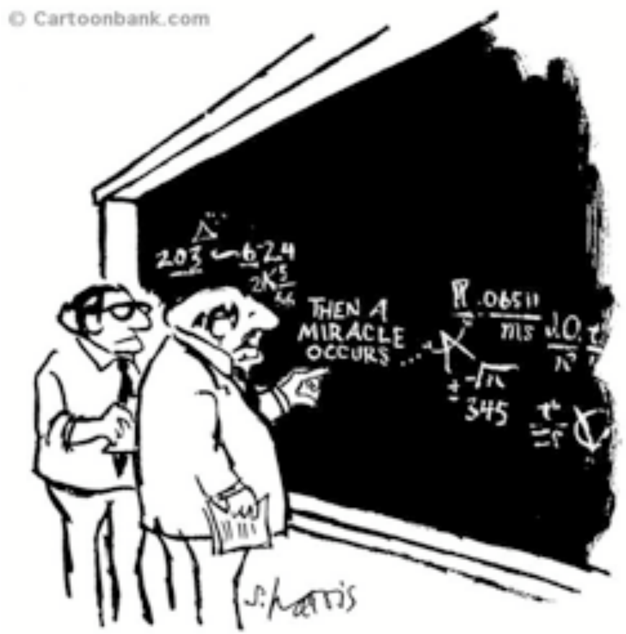
We need help

- A little experiment from http://www.youtube.com/v/vJG698U2Mvo&hl=en_US&fs=1&rel=0
- Rules
  - No one talks for the next 4 minutes
  - If you know what is about to happen, see (1)
- This is a selective attention test
  - Count the number of times the team with the white shirt passes the ball.

Data analysis deserves (much) more than zero pages

Easterbrook et al. (2007)

- 9 pages: selecting methods
- 3 pages: research questions
- 2 pages: empirical validity
- 2 pages: different forms of "empirical truth"
- 1 page: role of theory building
- 1 page: conclusions
- 1 page: data collection techniques
- 0 pages: data analysis
  - and then a miracle happens



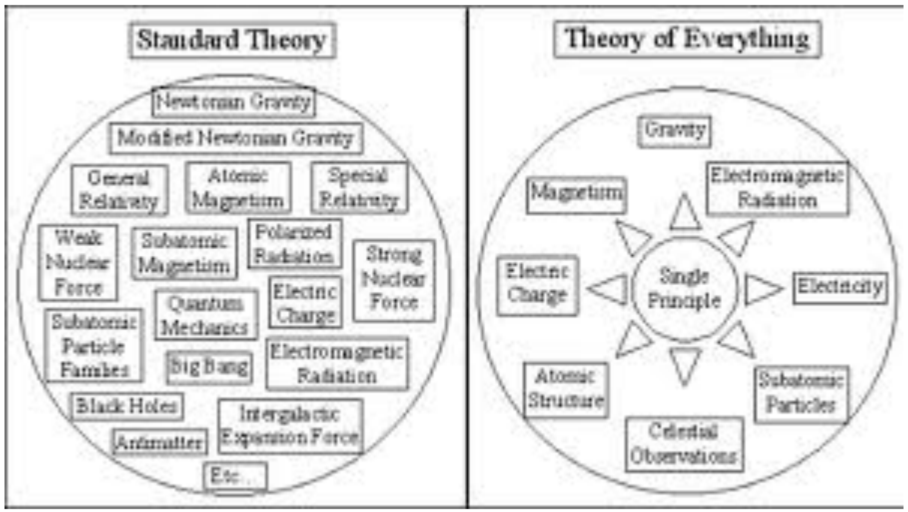
"I think you should be more explicit here in step two."

Don't just do data mining

- Be of the empirical research community
  - Go to LASER, ICSE, etc
  - Talk
- Find current hypothesis that of interest
  - E.g. max AUC(effort,pdf)
  - E.g. cross-vs-within data
  - E.g. TOE
  - E.g. text mining for structured reviews
    - Juristo, Menzies, 2011

Don't do data mining once

- Continuous process monitoring
  1. Learn expectations
  2. Stale smell policy: when good ideas go bad
  3. Repair policies: how to modify old ideas (more mining)
  4. Escalation policy: recognize when you need to call for help
- Bt the way,
  - 1,2,3,4 can all be implemented by data miners.
- Welcome to TOE



Other Do-s and Don'ts

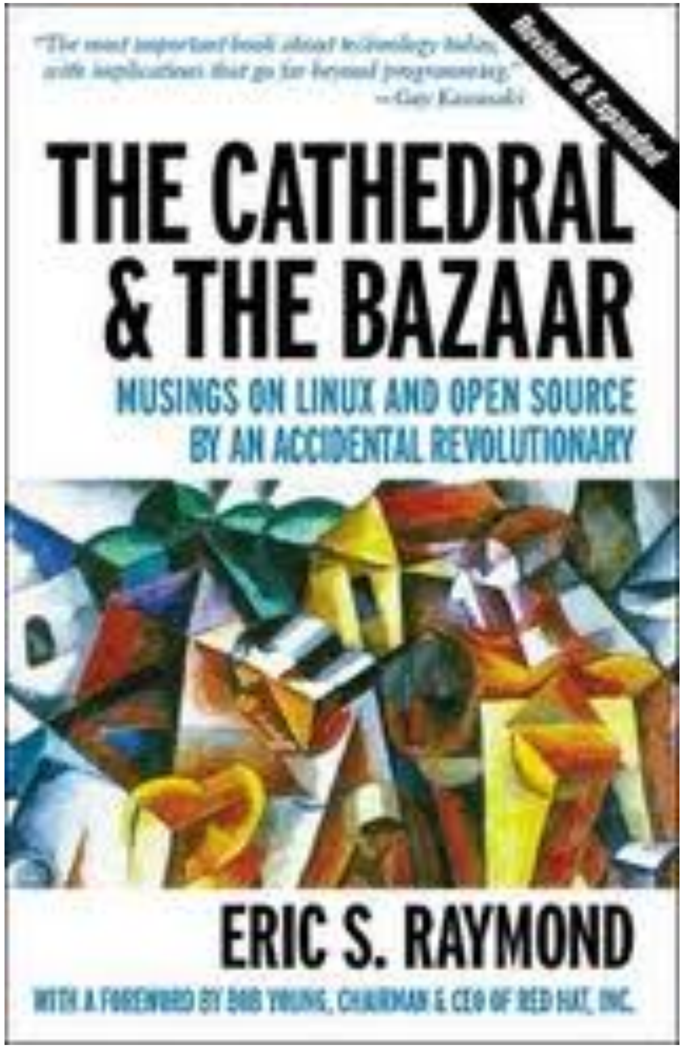
- Do <u>learn</u> about data mining
  - People make mistakes
  - Need communities of agents (human and otherwise)
  - New algorithms, old data, new insights
- Don't used <u>dumb</u> data mining:
  - correlation, PCA??
  - Forgettaboutit
- Don't quote old <u>dumb studies</u>:
  - E.g. Mccabe
- Do study <u>stability</u>:
  - 20 * 66% of the data
- Do model <u>bias</u>
  - Bias is where the business meets the learning

Exploit the crowd source advantage

- Crowd source
  - Join the community of people studying the data
  - Be there for them
- Lead, follow, or get out of the way
  - What's fair got to do with it? Its going to happen
  - Wolfgang Grieskamp from Microsoft, at Dagstuhl 2010

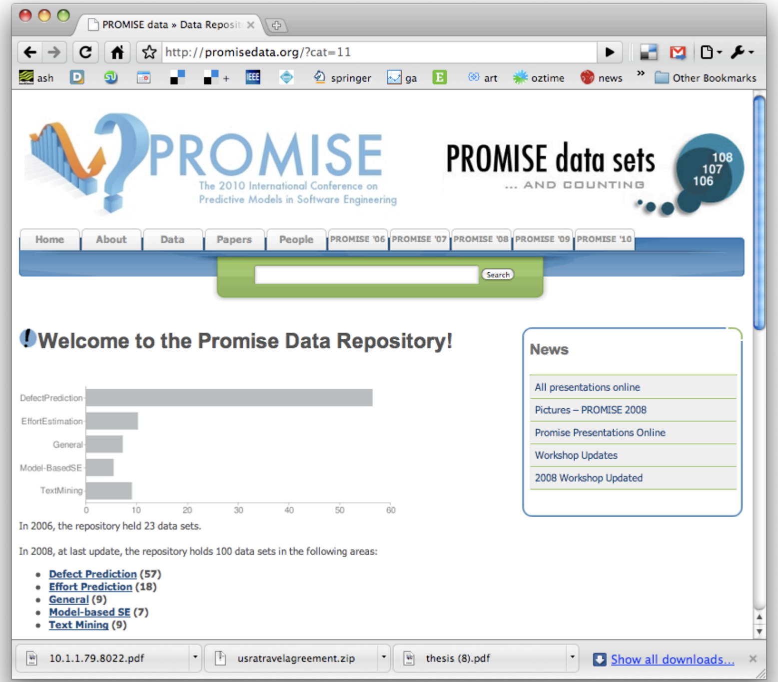
Open data initiatives

- Open source?
  That'll never work
- Menzies = bazaar!
- Are you the high priest in a cathedral?



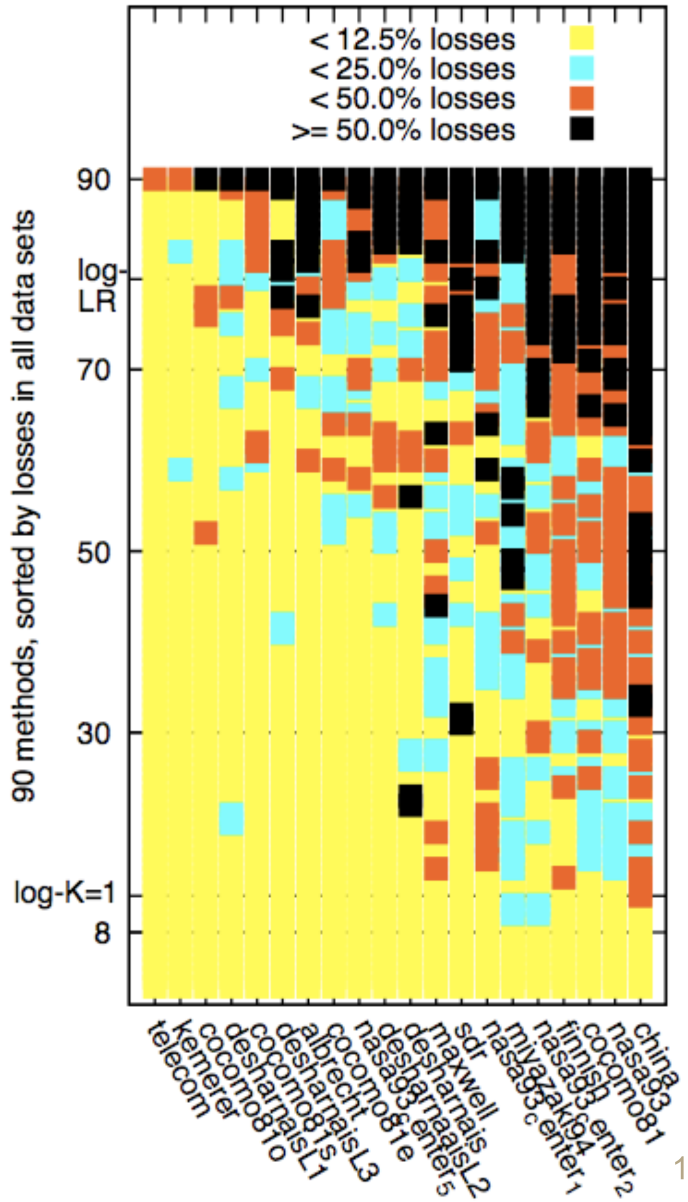
Lighten up!

- Do put data on the web
- Do collect data with "sunset clauses" (when it can go public)
  - The COCOMO experience
- Do collect data that joins
  - performance indicators
  - with things you can change
- Much inaccessible empirical data:
  - Data from the 152/ 154 MSR papers
  - ISERN 2007, ISERN 2008, ISERN 200
  - COCOMO-II
  - SEL
  - CeBase.org
- No propriety software
  - Static pages (no code that needs maintaining)
  - Password free sites (after the sunset)



Generate better results, faster

- Empirical SE results greatly lag the pace of innovation in the field.
- In too many cases
  - A trusted body of empirical results….
  - …Only appears after the innovative is already well on their way to obsolescence or standard practice.
- The generality of a result from any one case study is highly questioned. We urgently need:
  - Faster ways to learn local lessons
  - Faster ways to study data from multiple sources
- Can't always afford N people*Y years
  - Managers need answers yesterday
  - Funding bodies want progress
- Every time someone says "it depends"…
  - A grad student dies.



If a tree falls in a forest….

- Pooh and Piglet were walking together in the Thousand Acre Wood.
- The wind was blowing ferociously and the treetops were swaying.
- Somewhat disconcerted, Piglet asked Pooh, "What if a tree falls on us?"
- Pooh considered for a moment, before replying "What if it doesn't?"
- Barbara Kitchenham et al, ESEj, 2008
  - Replications can replicate stupid errors
  - Me: and sometimes, they don't
- Vic Basili, LASER, 2010
  - If we give people our data, they can make stupid mistakes, cause they don't understand the context
  - Me: and sometimes, they won't

Dude! Chill out!

By the way….

I am happy to report that there is no book called "data mining for dummies"