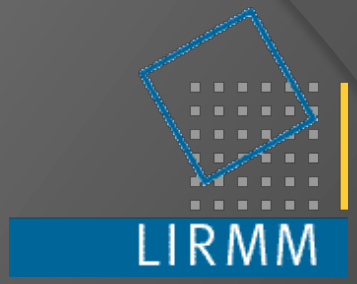


INTRODUCTION TO WEKA

FMIN311

Master DECOL-2013

Université de Montpellier 2

Hugo Alatrista-Salas : hugo.alatrista-salas@teledetection.fr

Introduction

WEKA

- Gallirallus australis : Endemic bird (New Zeland)

Characteristics

- Waikato university
- Weka is a collection of machine learning algorithms for data mining tasks
- Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.
- Under GPL license

Links

- http://www.cs.waikato.ac.nz/ml/weka/
- http://transact.dl.sourceforge.net/sourceforge/weka/WekaManual-3.6.0.pdf

How to run Weka?

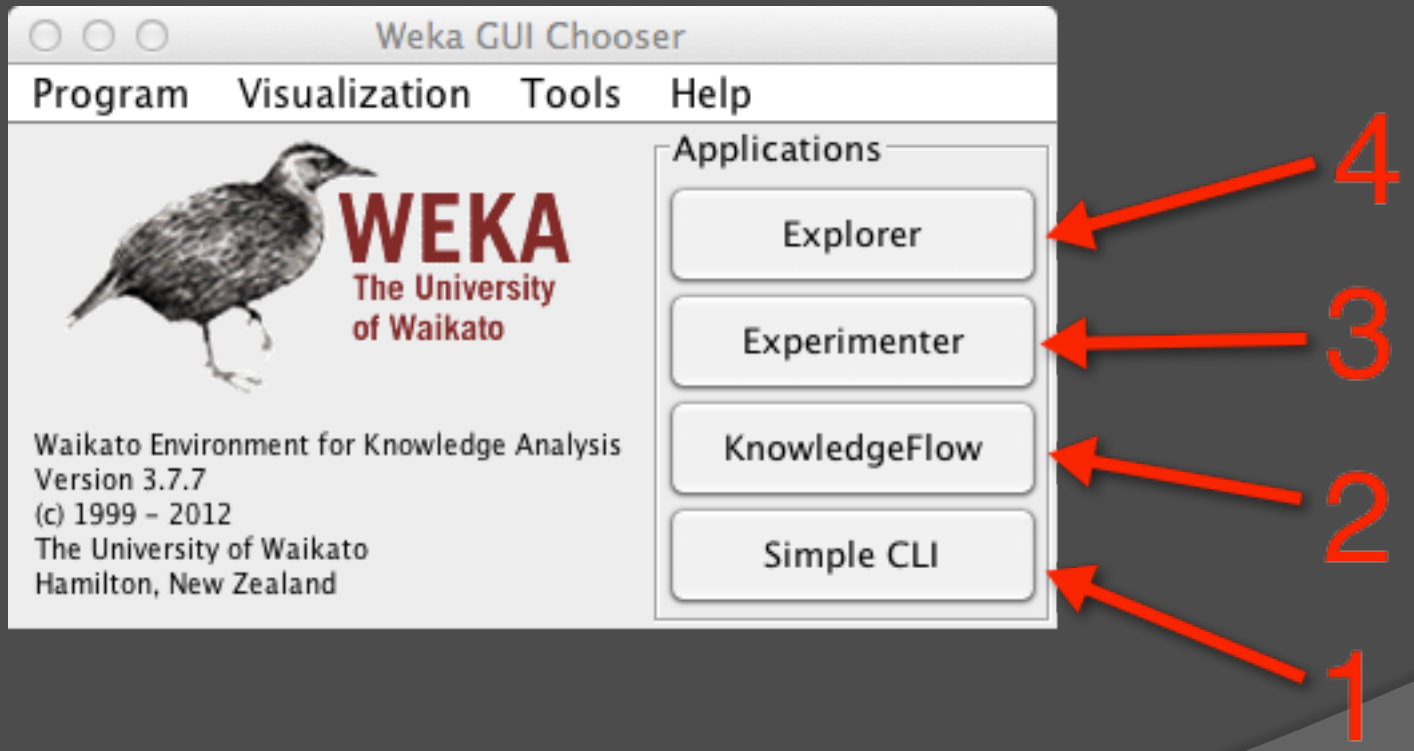
- Using the icon

- Using the command line

java -Xmx1024m -jar weka.jar

Interface

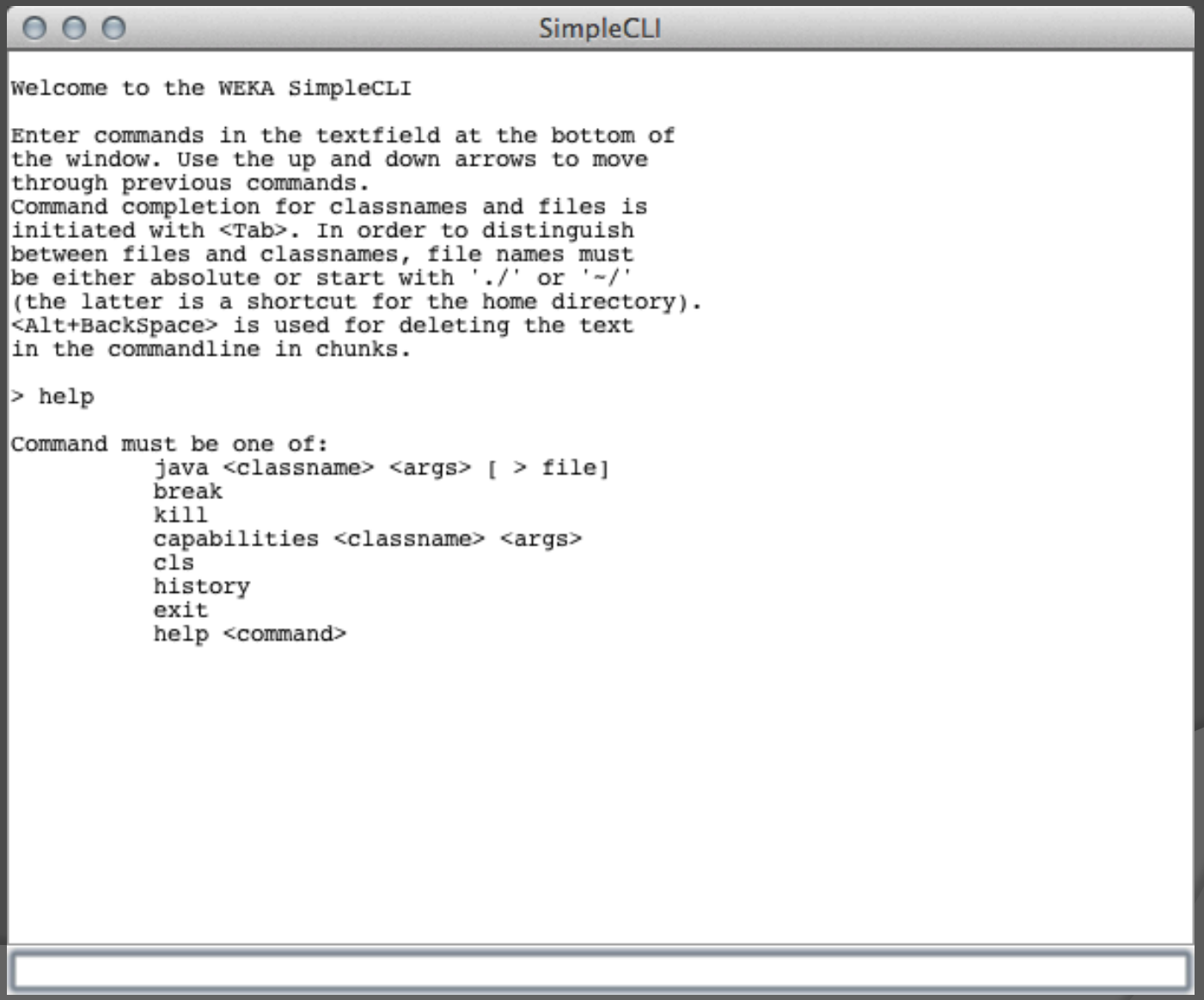
Main interface



Simple CLI (1)

- Support all operation proposed by WEKA
- E.g.
  - java <class><param>
  - break
  - kill
  - cls
  - exit
  - help <command>
  - …

Simple CLI (2)

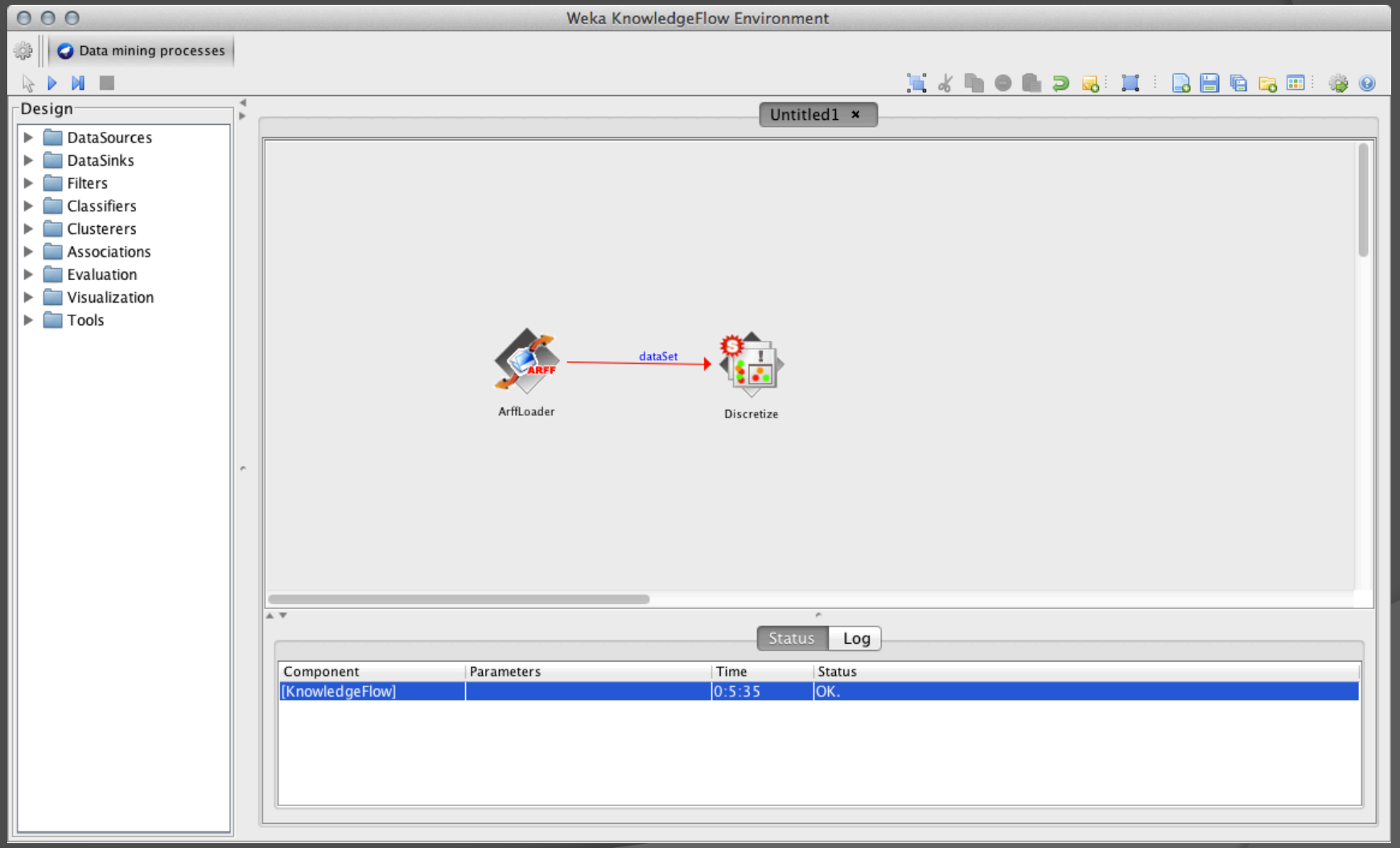

SimpleCLI
Welcome to the WEKA SimpleCLI
Enter commands in the textfield at the bottom of
the window. Use the up and down arrows to move
through previous commands.
Command completion for classnames and files is
initiated with <Tab>. In order to distinguish
between files and classnames, file names must
be either absolute or start with './' or '~/'
(the latter is a shortcut for the home directory).
<Alt+BackSpace> is used for deleting the text
in the commandline in chunks.
> help
Command must be one of:
java <classname> <args> [> file]
break
kill
capabilities <classname> <args>
cls
history
exit
help <command>

Knowledge Flow (1)

- Alternative to the Explorer as a graphical front end
- Intuition:

  The user can select WEKA components from a tool bar, place them on a layout canvas and connect them together in order to form a knowledge flow for processing and analyzing data.

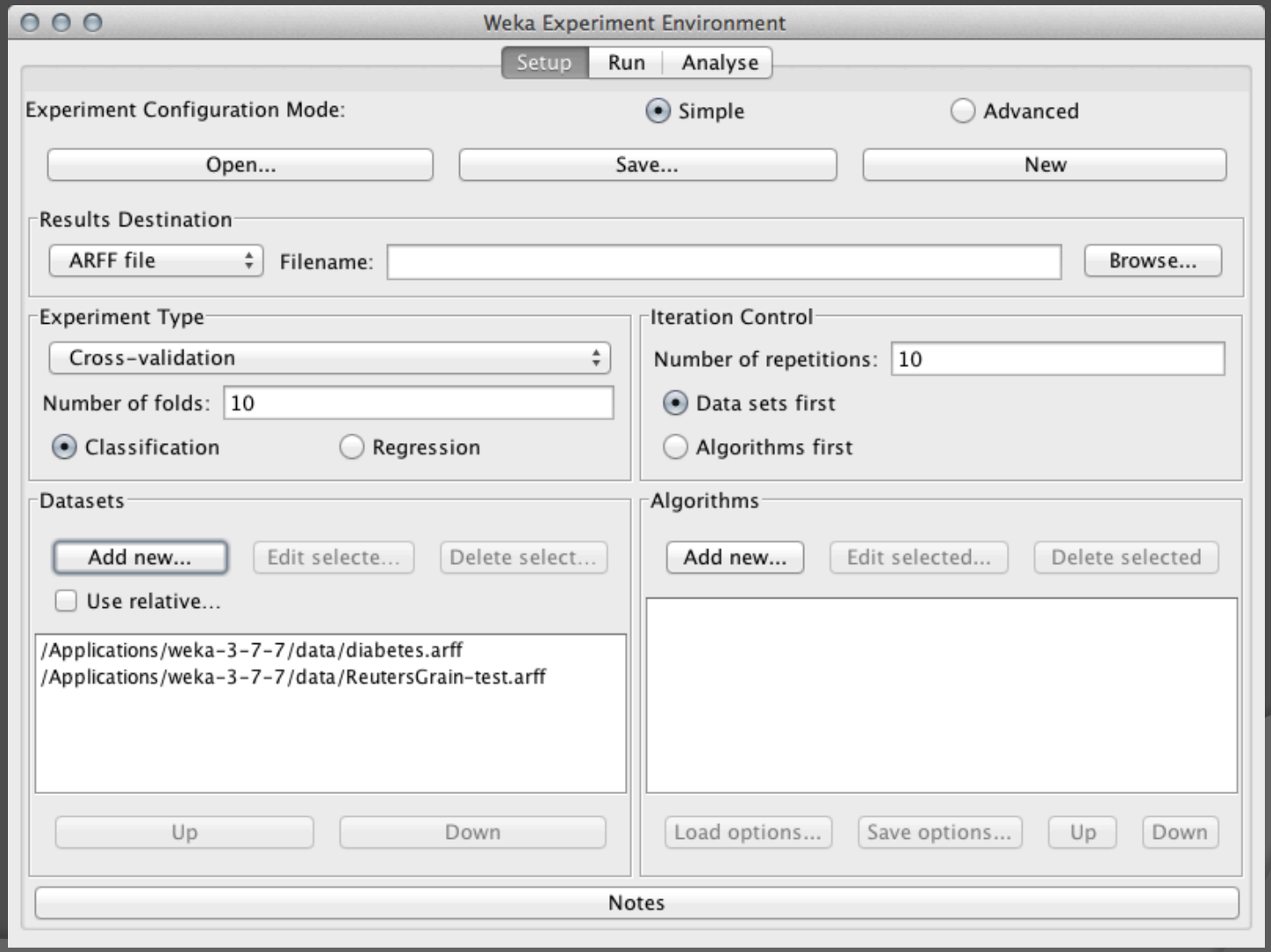
Knowledge Flow (2)



Experimenter (1)

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

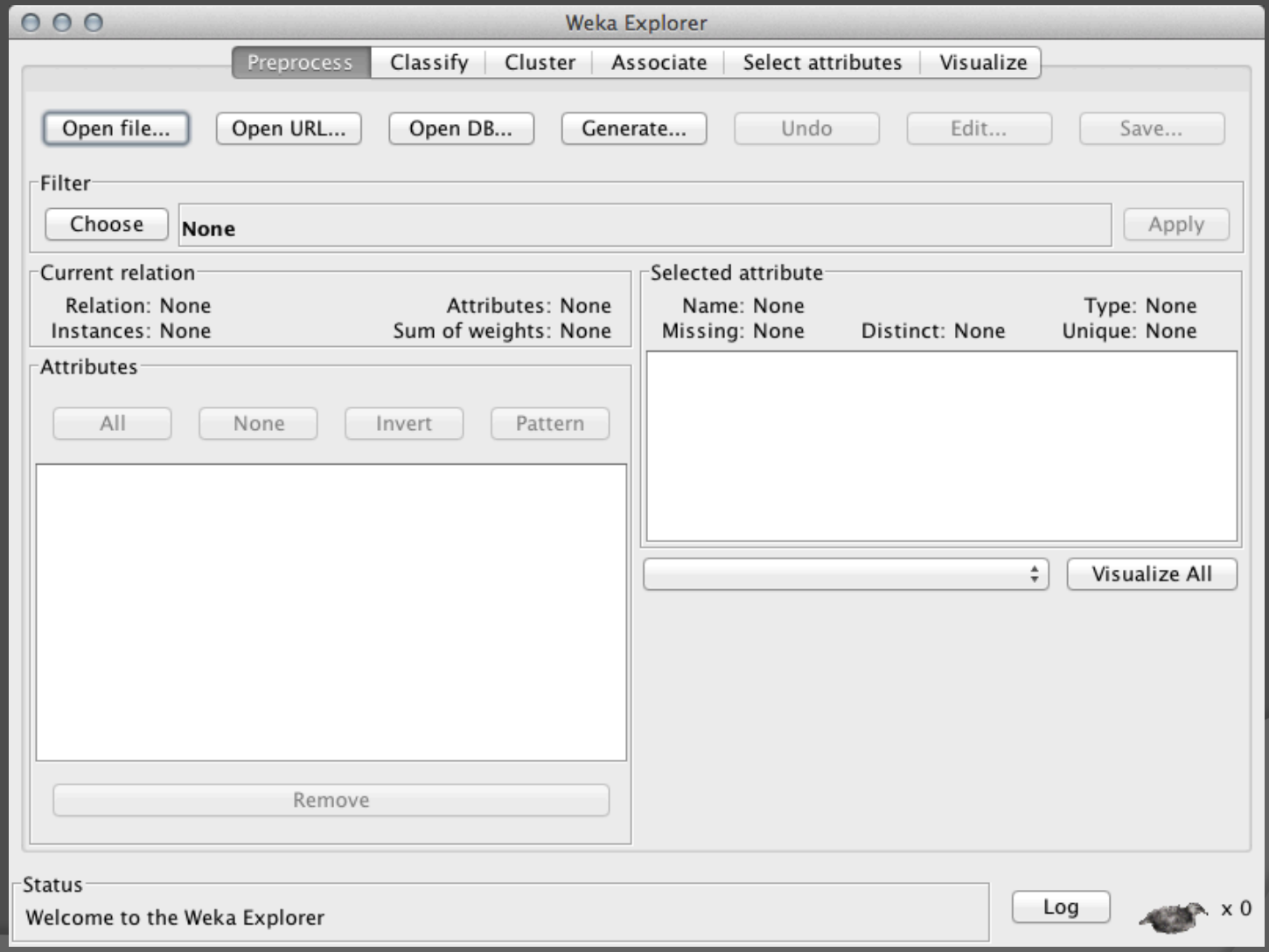
Experimenter (2)



Explorer (1)

- Preprocess
- Classify
- Cluster
- Associate
- Select attributes
- Visualize

Explorer (2)


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

Using WEKA

Data format supported by Weka

- Data can be imported from a file in various formats:
  - ARFF – default format file
  - CSV – separated by comas or tabulations
  - C4.5 – codify under C4.5 format (.names to store the names and .data to store the data)
  - JSON – data files used by Javascript

ARFF files (1)

diabetes.arff

```
@relation pima_diabetes
@attribute 'preg' real
@attribute 'plas' real
@attribute 'pres' real
@attribute 'skin' real
@attribute 'insu' real
@attribute 'mass' real
@attribute 'pedi' real
@attribute 'age' real
@attribute 'class' { tested_negative, tested_positive}
@data
6,148,72,35,0,33.6,0.627,50,tested_positive
1,85,66,29,0,26.6,0.351,31,tested_negative
8,183,64,0,0,23.3,0.672,32,tested_positive
1,89,66,23,94,28.1,0.167,21,tested_negative
0,137,40,35,168,43.1,2.288,33,tested_positive
5,116,74,0,0,25.6,0.201,30,tested_negative
3,78,50,32,88,31,0.248,26,tested_positive
10,115,0,0,0,35.3,0.134,29,tested_negative
2,197,70,45,543,30.5,0.158,53,tested_positive
8,125,96,0,0,0,0.232,54,tested_positive
4,110,92,0,0,37.6,0.191,30,tested_negative
10,168,74,0,0,38,0.537,34,tested_positive
10,139,80,0,0,27.1,1.441,57,tested_negative
1,189,60,23,846,30.1,0.398,59,tested_positive
5,166,72,19,175,25.8,0.587,51,tested_positive
7,100,0,0,0,30,0.484,32,tested_positive
0,118,84,47,230,45.8,0.551,31,tested_positive
7,107,74,0,0,29.6,0.254,31,tested_positive
1,103,30,38,83,43.3,0.183,33,tested_negative
```

ARFF files (2)

- Header:

  @relation <relation name>
- Attributes declaration:

  @attribute <name> <type>

  where <type> can be a value (numeric, string, date, etc) or nominal (set of values, e.g. {female, male})
- Data

  @data

  …

ARFF file example

```
% file to test.
@relation test
@attribute name STRING
@attribute health {good, bad}
@attribute weight NUMERIC
@attribute date_analyse DATE "dd-MM-yyyy HH:mm”
@data
Alice, good, 38.43, "12-04-2003 12:23”
’Maria Jose’, ?, 34.53, "14-05-2003 13:45”
Alex, good, 43, "01-01-2004 08:04”
Richard, ?, ?, "03-04-2003 11:03"
```

ARFF files (sparse format)

- Considering only the non 0 values
- Represent each values with: POSITION VALUE information
- Each couple (POSITION - VALUE) is separated with a comma
- Useful for documents representation in ARFF format

e.g.

@data		@data
0, X, Y, “male”	→	{1 X, 2 Y, 3 “male”}
0, 0, W, “male”	→	{2 W, 3 “male”}

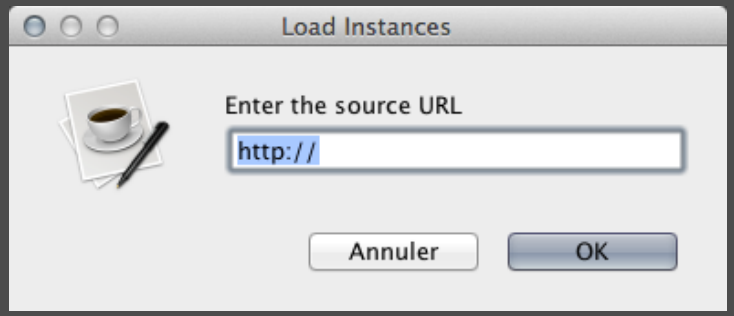
Exercise

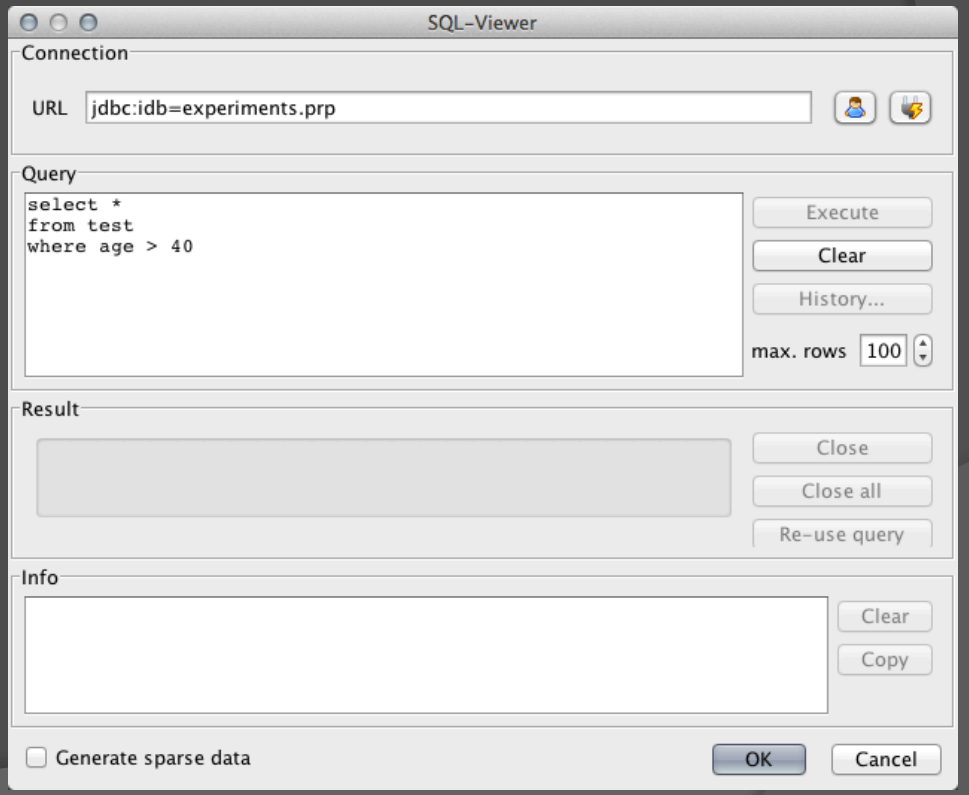
- Represent the following table in ARFF file (simple and sparse formats)

City	Date	Temperature	Humidity	Wind	Emergency
Alès	03/14	14.4	68	57	Yes
Paris	03/15	18.4	60		No
Nîmes	03/14	20.3	72	45	Yes
Nice	04/01	15.6	68	11	No
Lunel	03/18	28.0	71		No

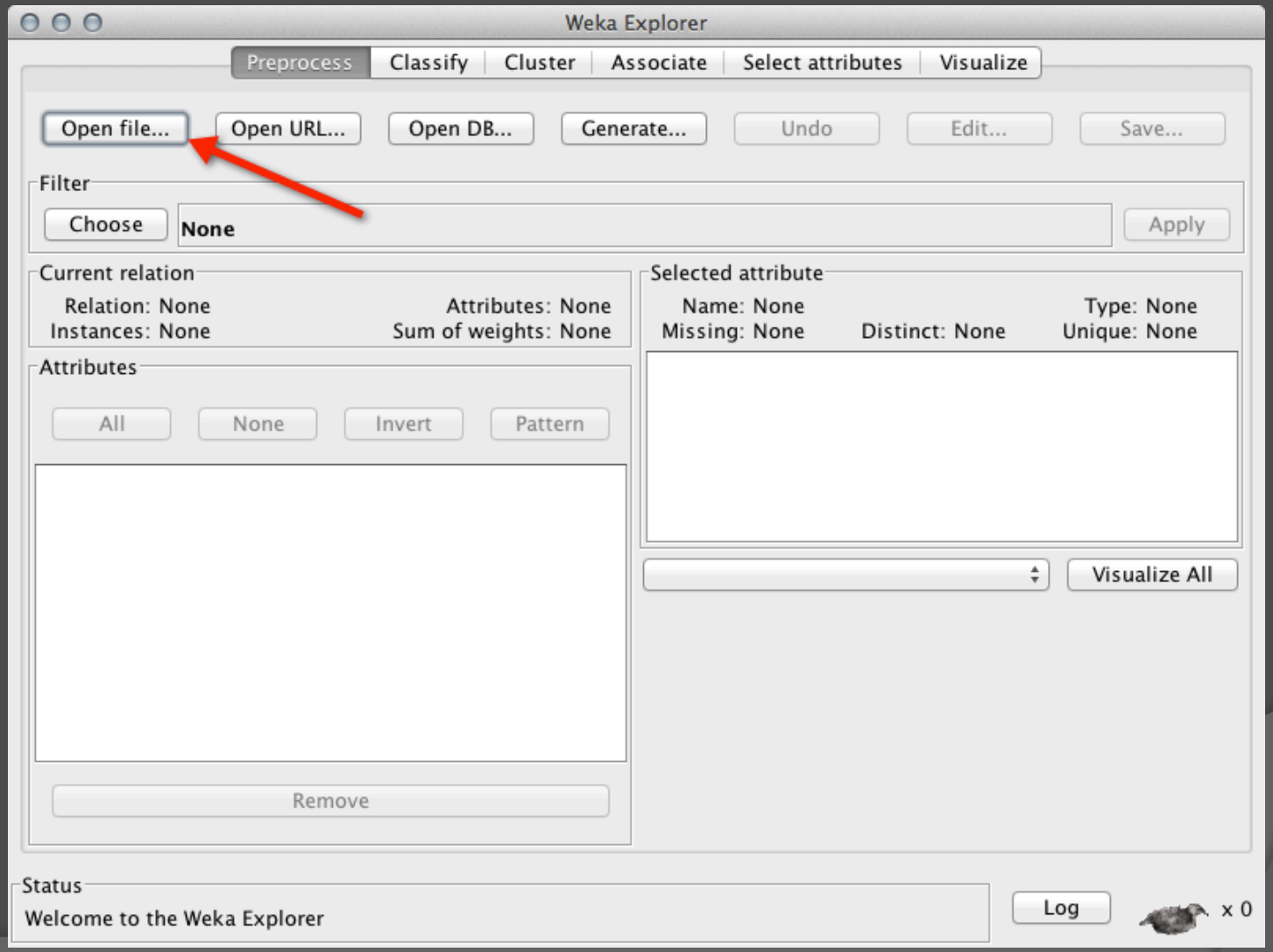
Open a DB or URL

- Data can also be read from a URL or from an SQL database (using JDBC)

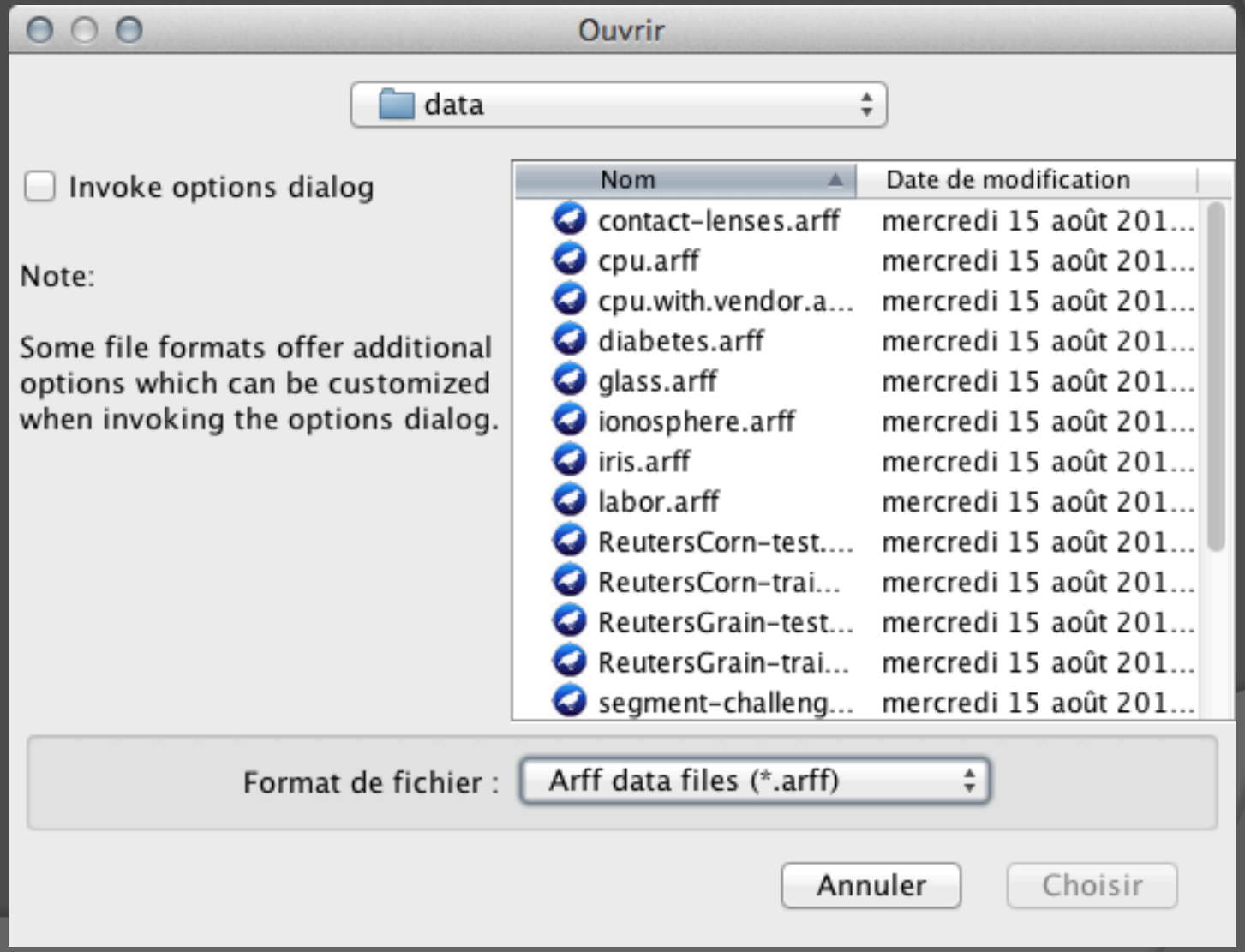




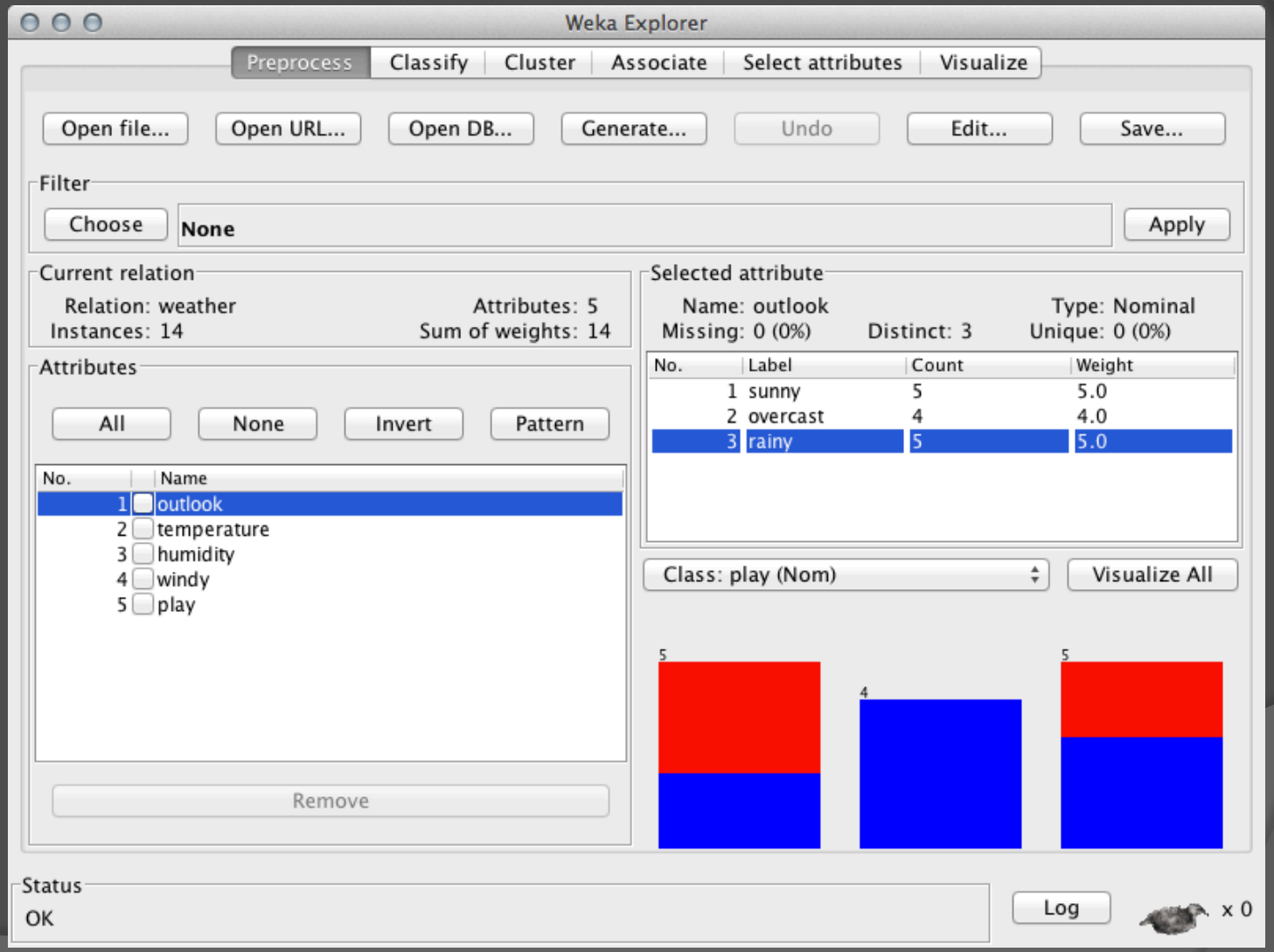
Open file using Explorer (1)



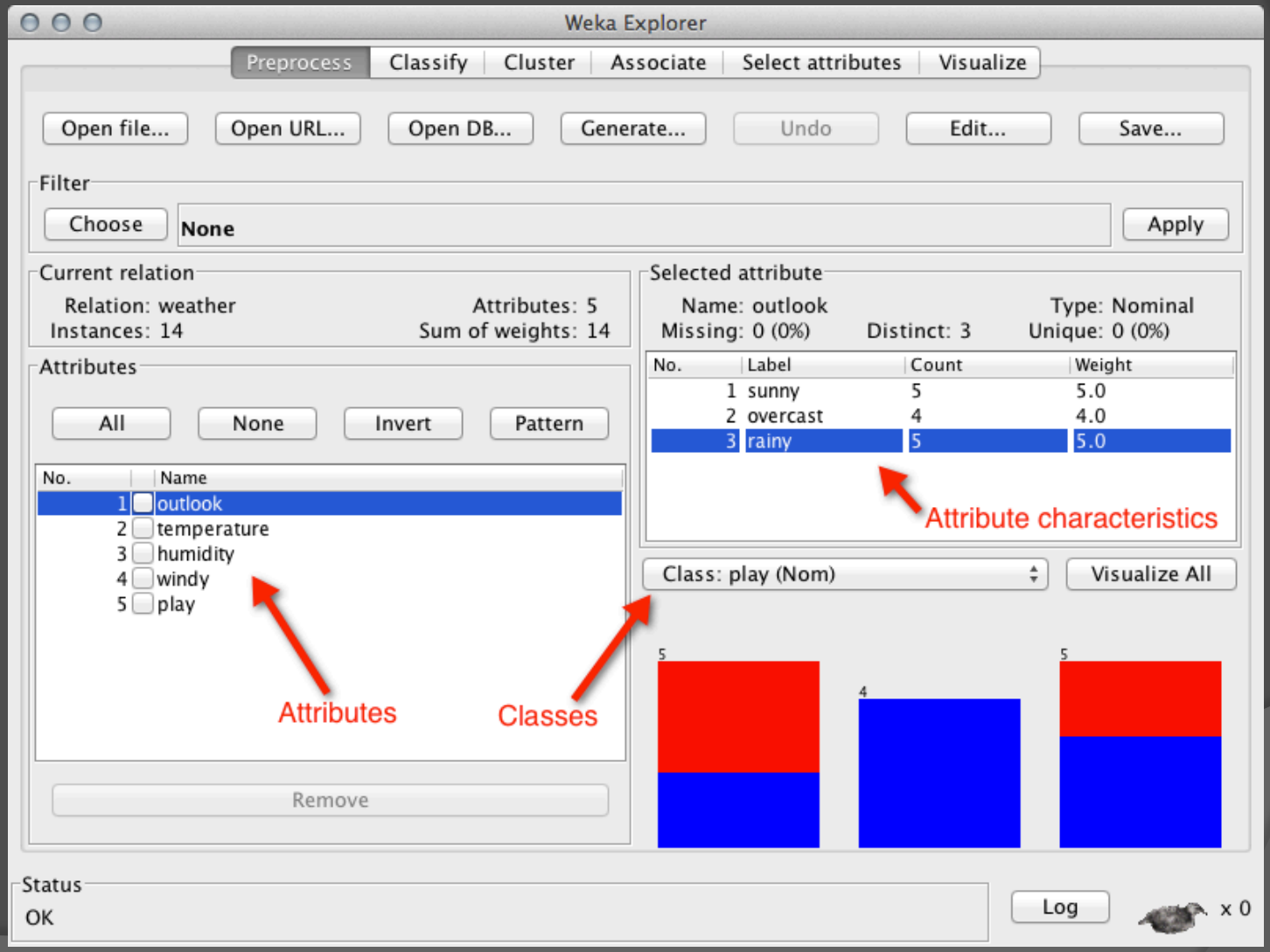
Open file using Explorer (2)



Open file using Explorer (3)



Open file using Explorer (4)

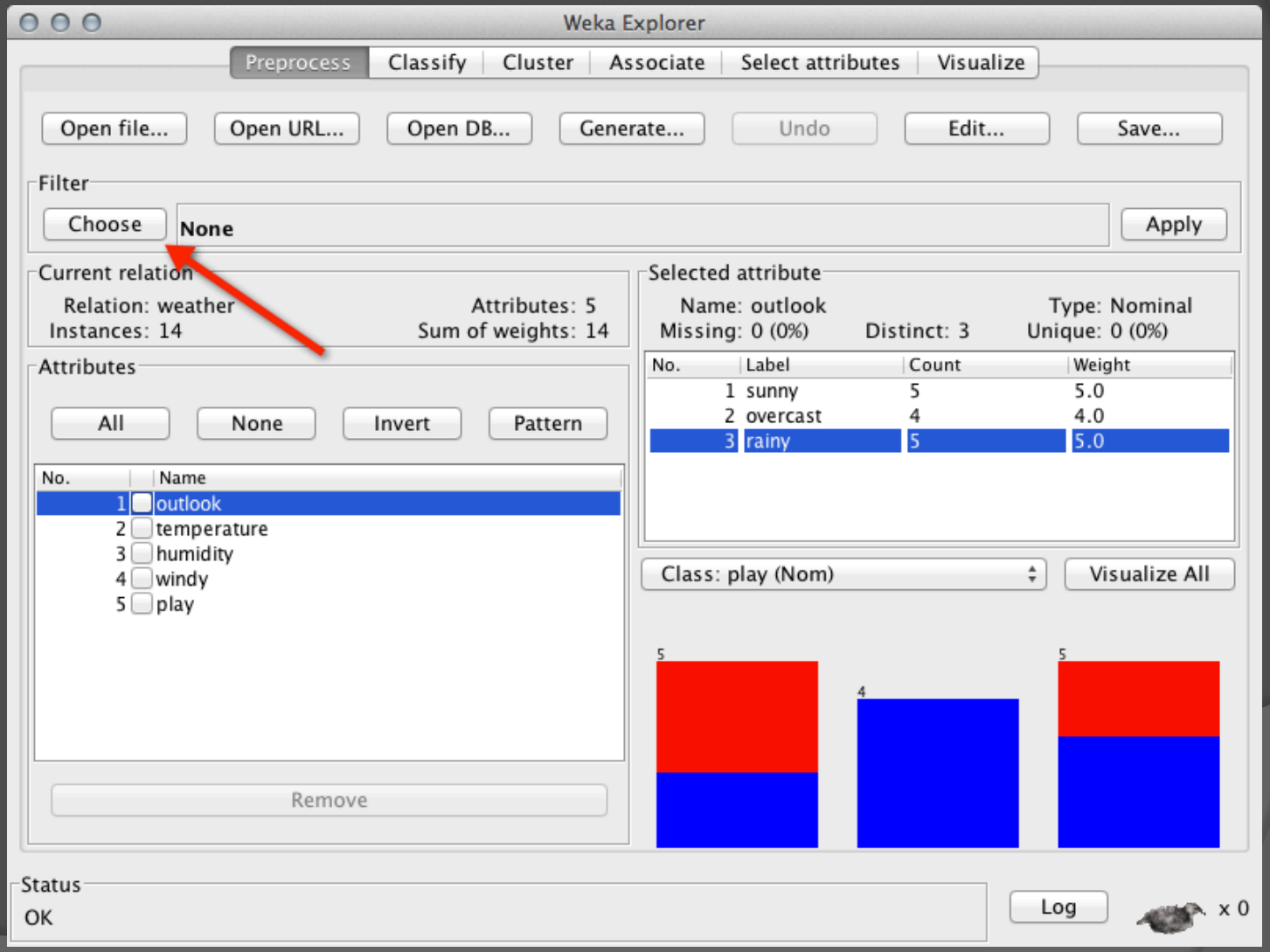


Pre-processing tools (1)

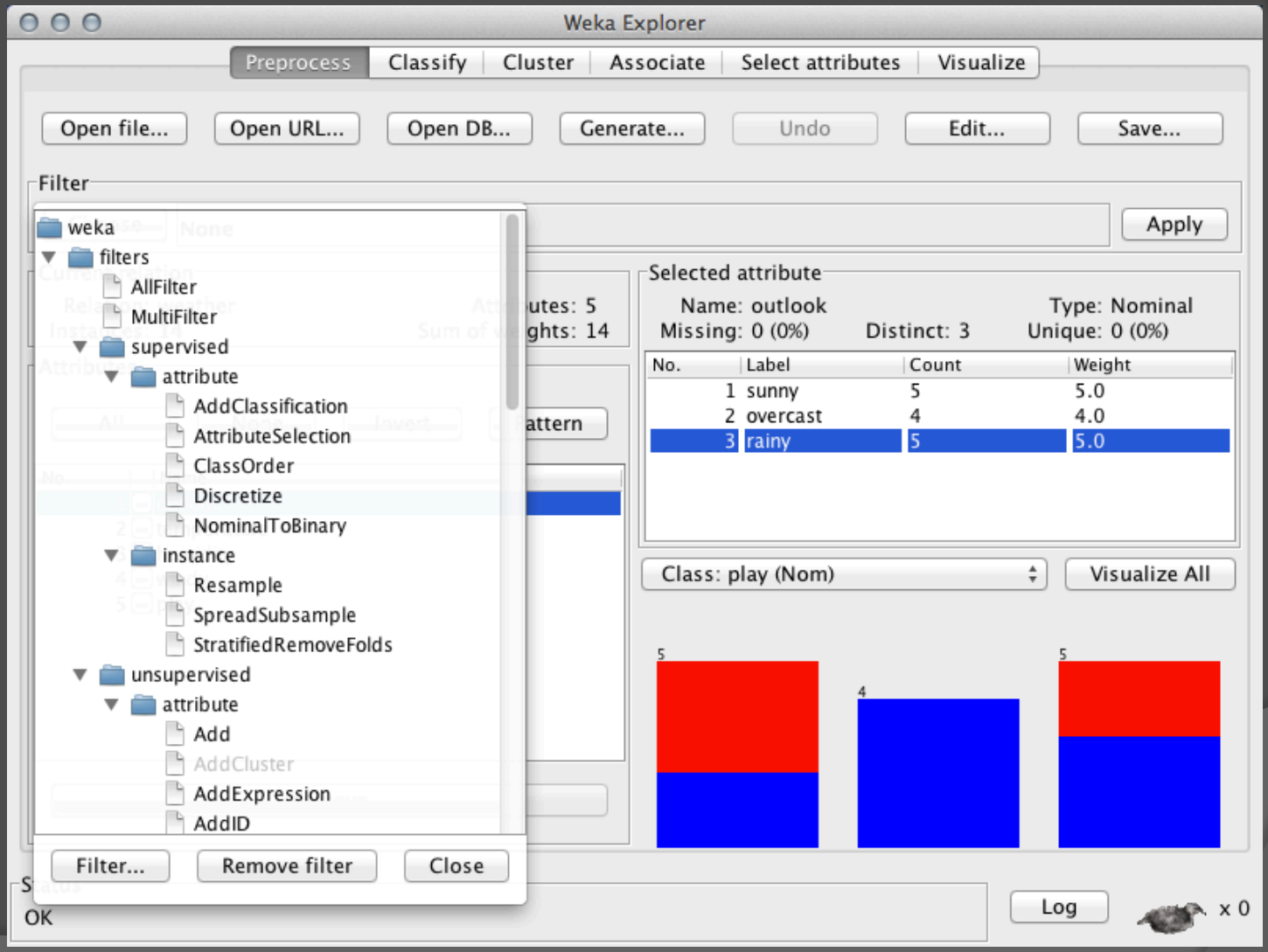
- Pre-processing tools in WEKA are called “filters”
- WEKA contains filters for:

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

Pre-processing tools (2)



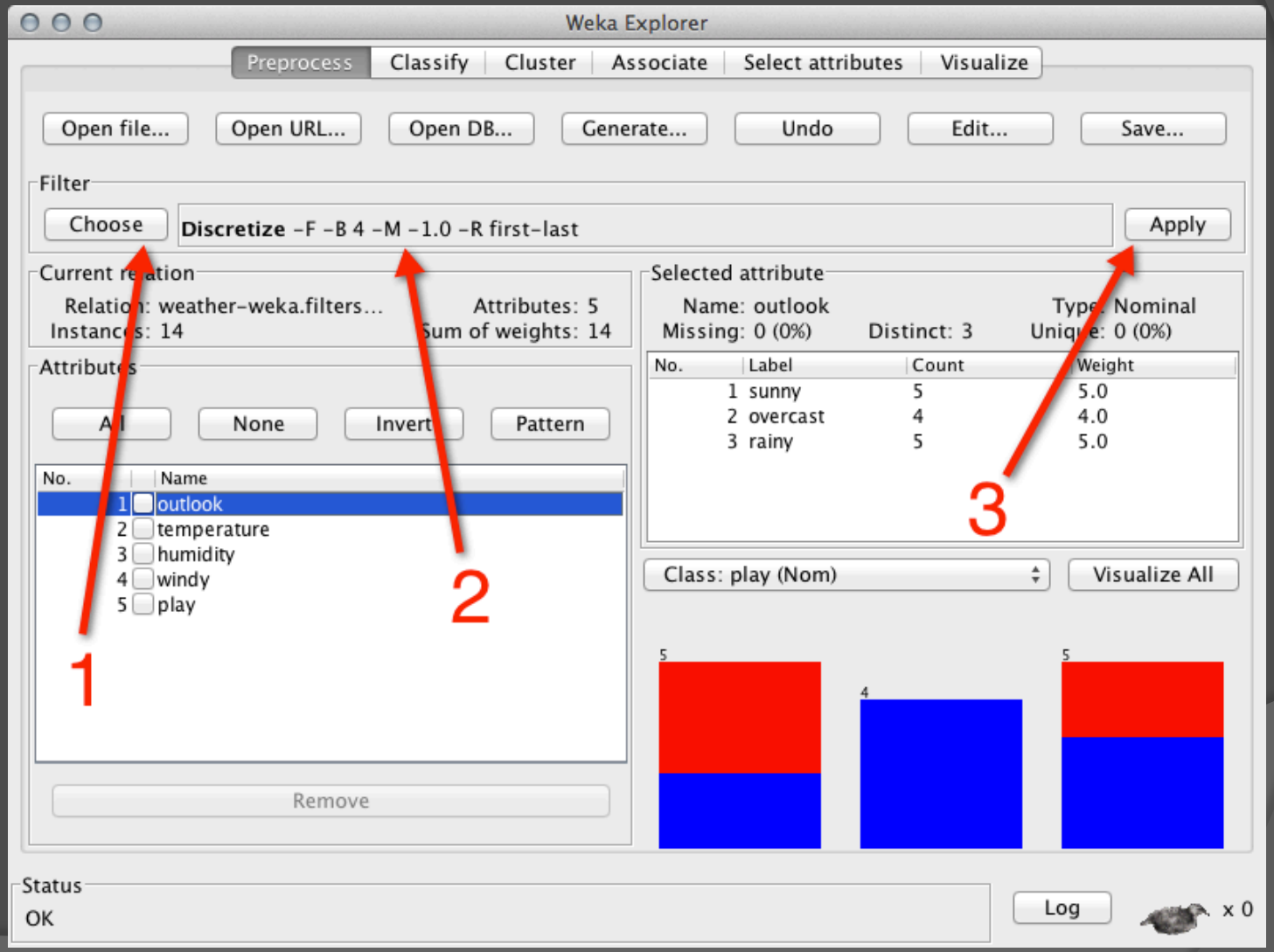
Pre-processing tools (3)



Example – Discretization (1)

- To obtain categorical data
- Used on numerical attributes
  1. Open a file (weather.arff for example)
  2. Choose a filter : *Filters* → *unsupervised* → *discretize* 1
  3. Left-click on properties 2
  4. Change the number of *binds* and *useEqualFrequency*
  5. Click on OK and APPLY 3

Example – Discretization (2)



Exercise

In the example, compare the characteristics of attributes before and after discretization:

- Evaluate the *outlook* attribute characteristics: Comment the results
- Evaluate the *temperature* attribute characteristics: Comment the results
- Export the results into *arff* and *csv* files

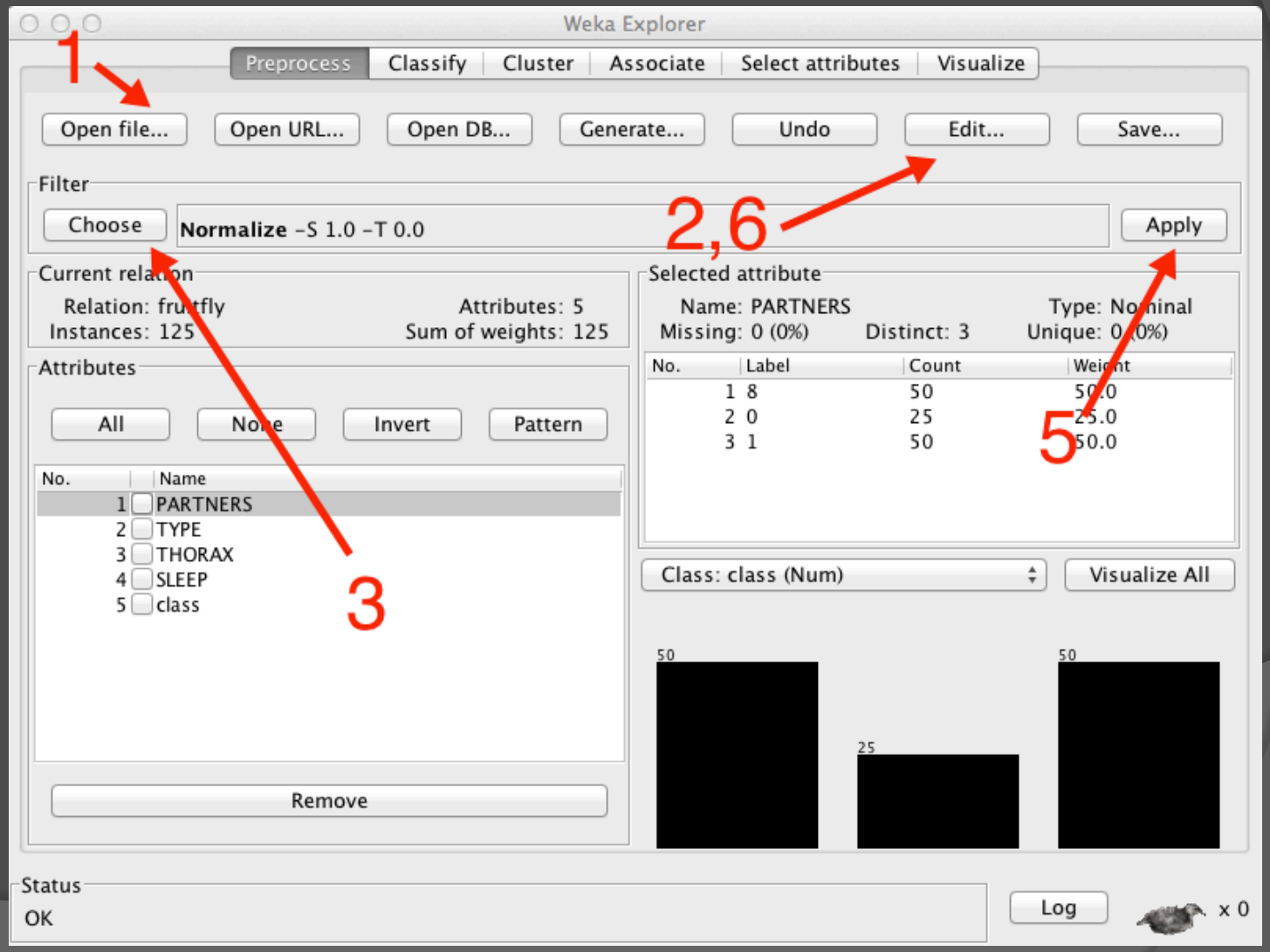
Data normalization

- Pre-processing technique
- The filter *standardization* allow us standardize all numerical values of the data set into values belonging the interval [0, 1]
- For more information, see “More”

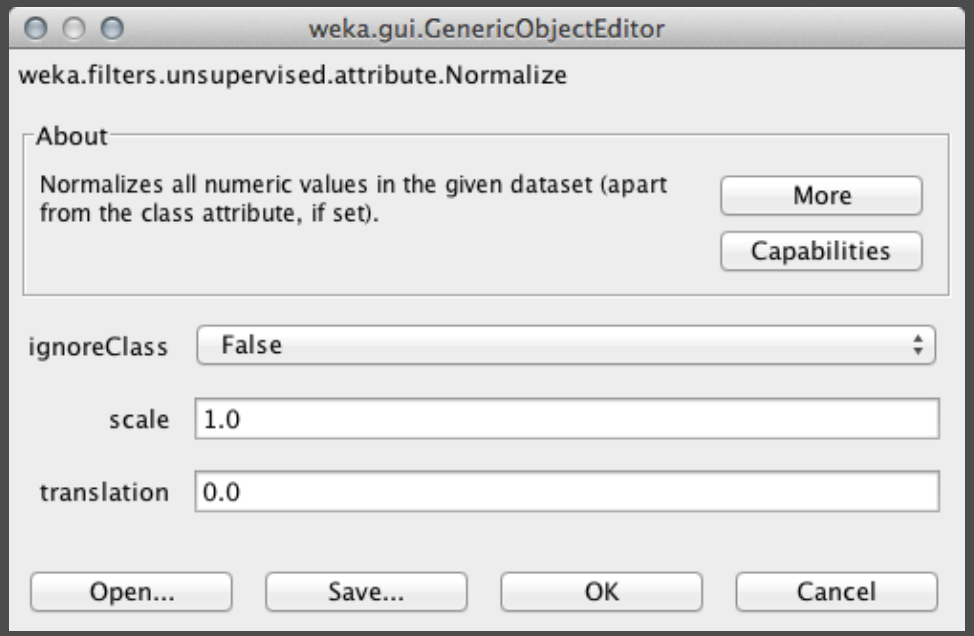
Example (1)

1. Open *fruitfly.arff* 1
2. See the dataset using the button *Edit* 2
3. Choose: *filters* → *unsupervised* → *attribute* → *normalize* 3
4. Set *scale* to 1.0 into options 4
5. Click on *Apply* 5
6. See the data using the button *Edit* 6

Example (2)


Weka Explorer
1
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
Normalize -S 1.0 -T 0.0
2,6
Apply
Current relation
Relation: fruitfly
Instances: 125
Attributes: 5
Sum of weights: 125
Selected attribute
Name: PARTNERS
Missing: 0 (0%)
Distinct: 3
Type: Nominal
Unique: 0 (0%)
No. Label Count Weight
1 8 50 50.0
2 0 25 25.0
3 1 50 50.0
5
Attributes
All
None
Invert
Pattern
No. Name
1 PARTNERS
2 TYPE
3 THORAX
4 SLEEP
5 class
3
Remove
Class: class (Num)
Visualize All
50
25
50
Status
OK
Log
x 0

Example (3)


weka.gui.GenericObjectEditor
weka.filters.unsupervised.attribute.Normalize
About
Normalizes all numeric values in the given dataset (apart from the class attribute, if set).
More
Capabilities
ignoreClass
False
scale
1.0
translation
0.0
Open...
Save...
OK
Cancel

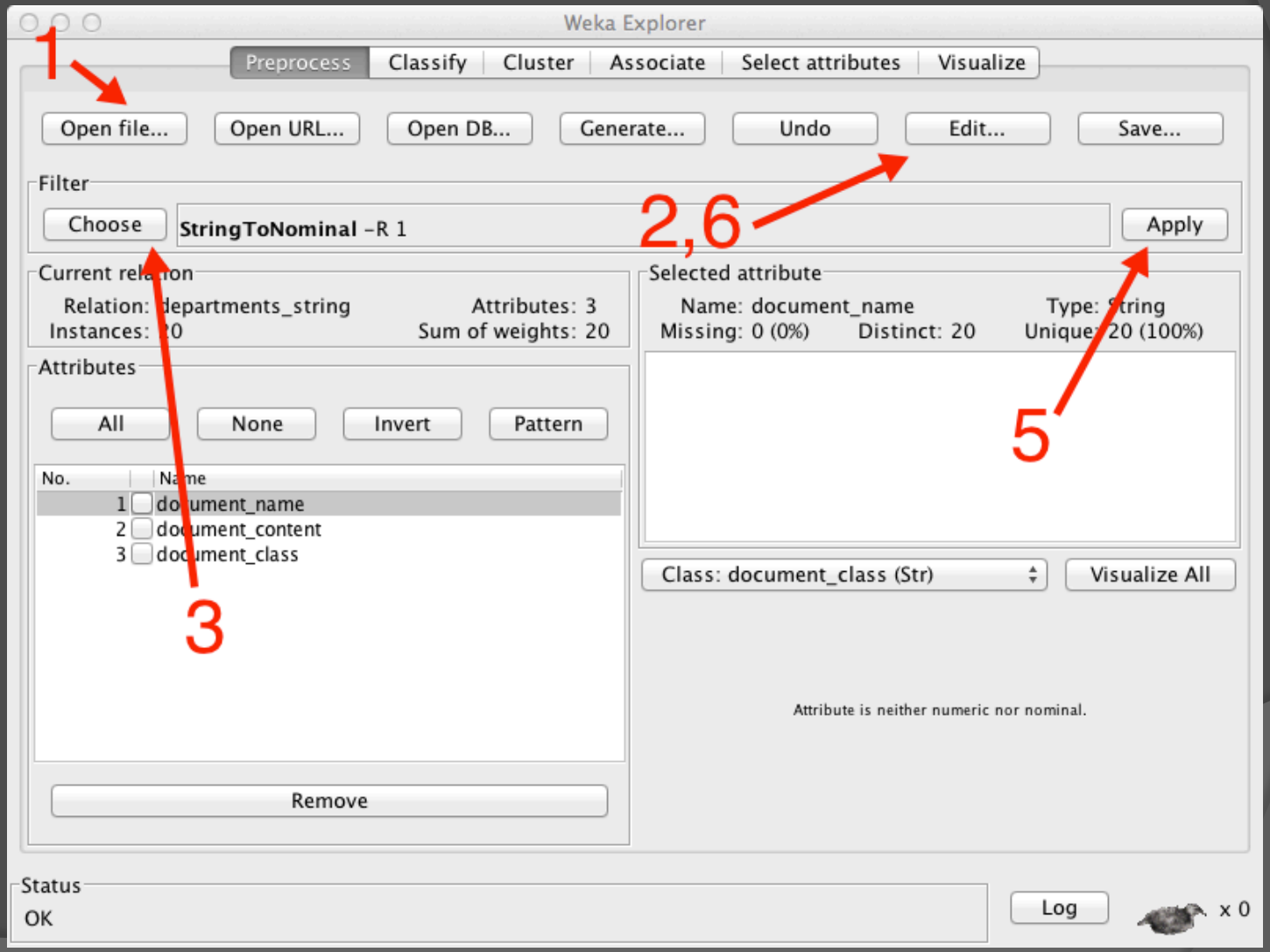
String attribute into nominal

- Pre-processing tool
- Converting a string attribute into nominal
- Finite number of values (string)
- For more information, see “More”

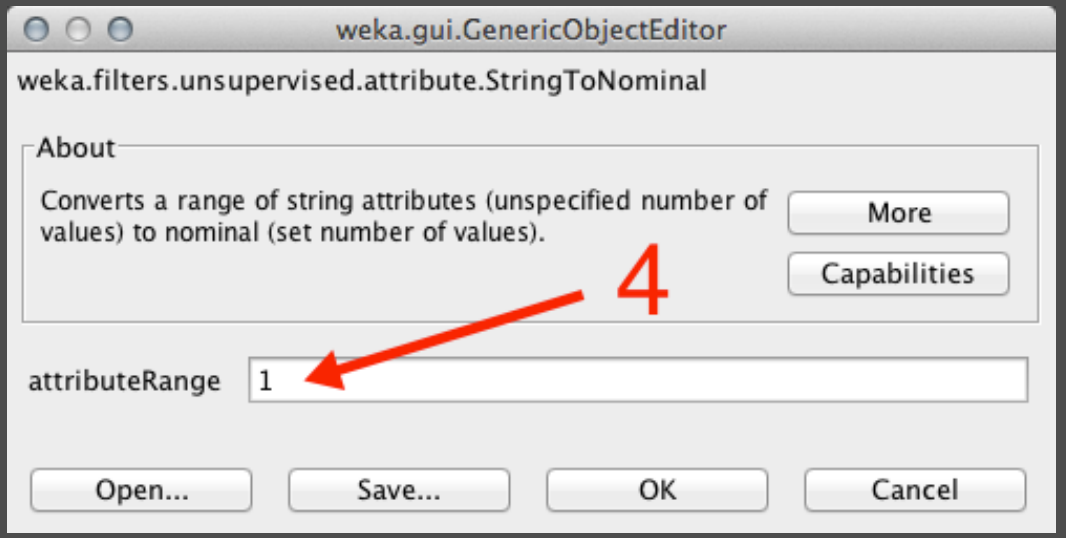
Example (1)

1. Open *Departments-string.arff* 1
2. See the dataset using the button *Edit* 2
3. Choose: *filters* → *unsupervised* → *attribute* → *StringToNominal* 3
4. Set *attributeRange* to 1 into options 4
5. Click on *Apply* 5
6. See the dataset using the button *Edit* 6

Example (2)


Weka Explorer
1
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
StringToNominal -R 1
2,6
Apply
Current relation
Relation: departments_string
Instances: 20
Attributes: 3
Sum of weights: 20
Selected attribute
Name: document_name
Type: String
Missing: 0 (0%)
Distinct: 20
Unique: 20 (100%)
Attributes
All
None
Invert
Pattern
No.
Name
1 document_name
2 document_content
3 document_class
3
5
Class: document_class (Str)
Visualize All
Attribute is neither numeric nor nominal.
Remove
Status
OK
Log
x 0

Example (3)


weka.gui.GenericObjectEditor
weka.filters.unsupervised.attribute.StringToNominal
About
Converts a range of string attributes (unspecified number of values) to nominal (set number of values).
More
Capabilities
4
attributeRange
1
Open...
Save...
OK
Cancel

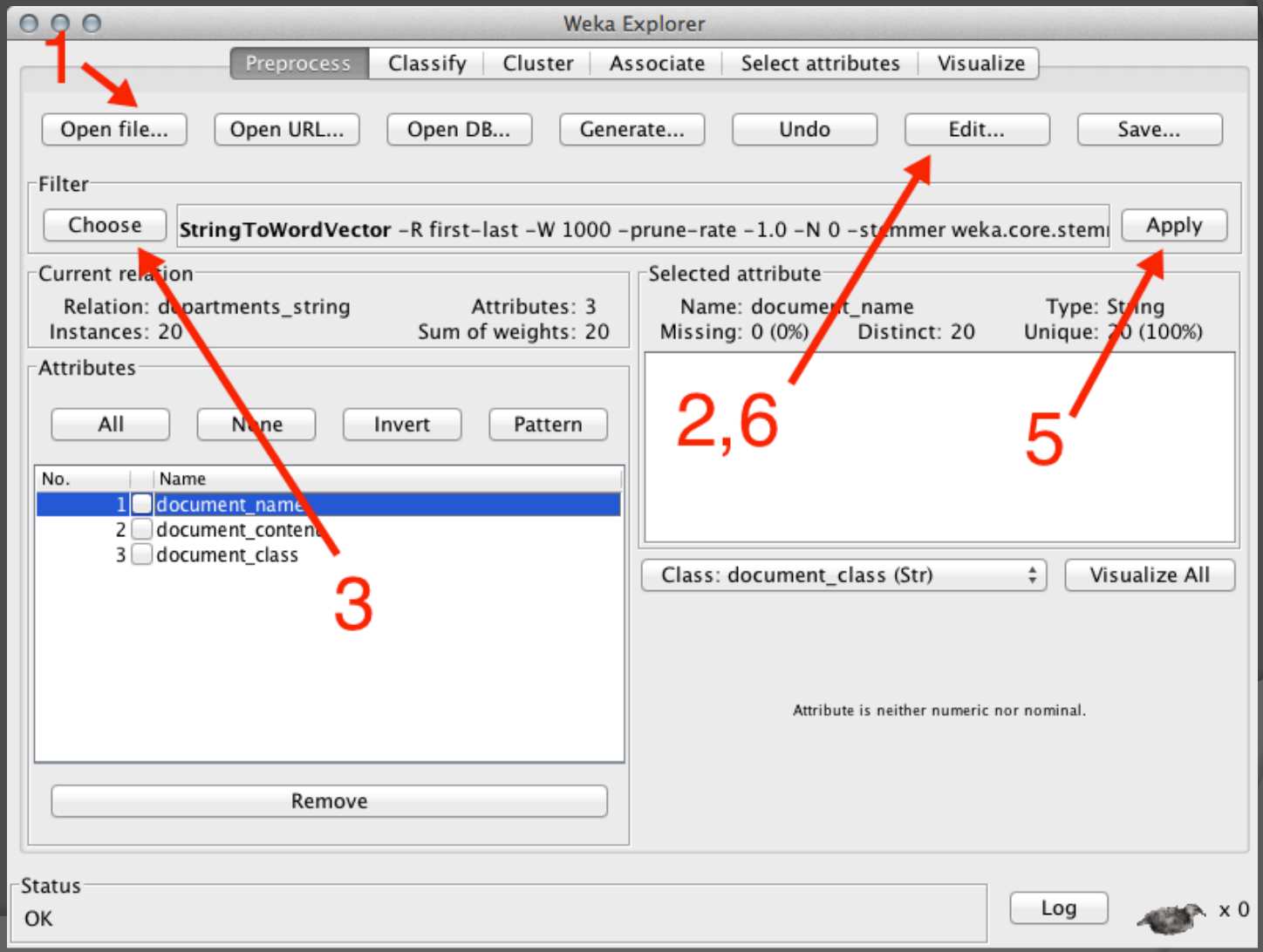
Text data to String vector (1)

- Pre-processing technique
- Converting text data into TF-IDF (Term Frequency – Inverted Document Frequency) attribute format
- Used on string data
- For more information, see “More”

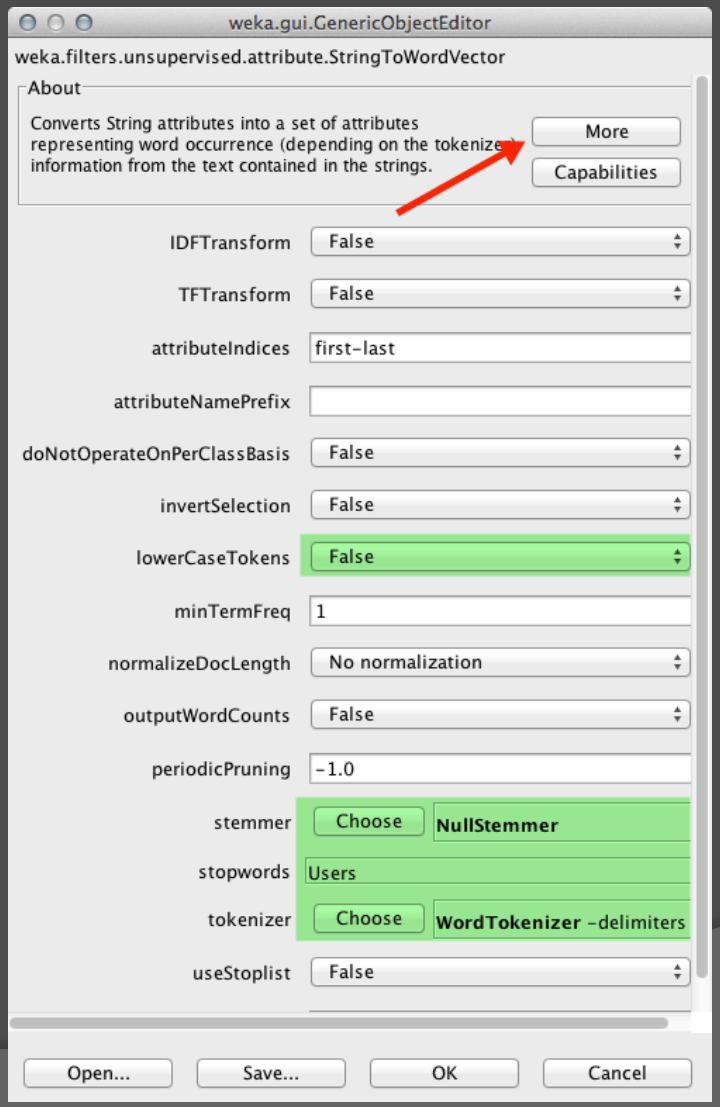
Example (1)

1. Open *Departments-string.arff* 1
2. See the dataset using the button *Edit* 2
3. Choose: *filters* → *unsupervised* → *attribute* → *StringToWordVector* 3
4. Set some options 4
5. Click on *Apply* 5
6. See the dataset using the button *Edit* 6

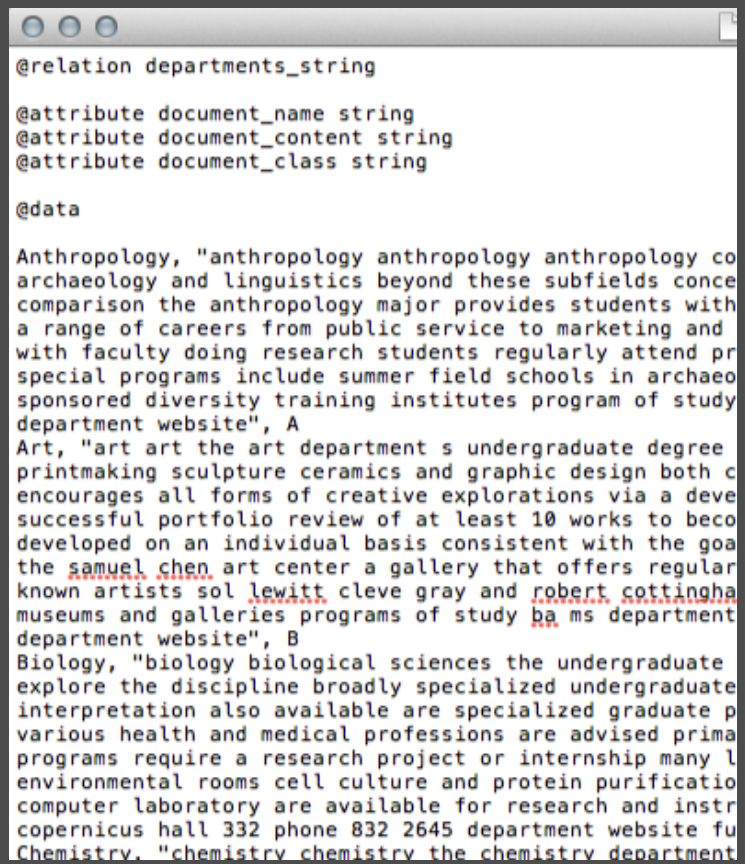
Example (2)


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
1
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -N 0 -st mmer weka.core.stem
Apply
Current relation
Relation: departments_string
Instances: 20
Attributes: 3
Sum of weights: 20
Selected attribute
Name: document_name
Missing: 0 (0%)
Distinct: 20
Type: String
Unique: 20 (100%)
Attributes
All
None
Invert
Pattern
No.
Name
1 document_name
2 document_content
3 document_class
2,6
5
3
Class: document_class (Str)
Visualize All
Attribute is neither numeric nor nominal.
Remove
Status
OK
Log
x 0

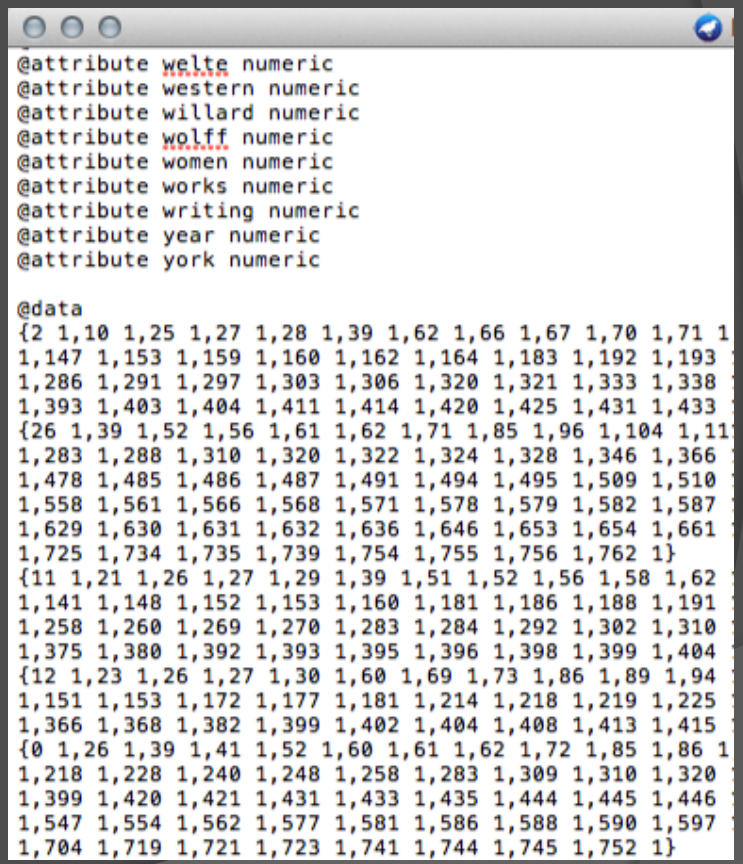
Example (3)


weka.gui.GenericObjectEditor
weka.filters.unsupervised.attribute.StringToWordVector
About
Converts String attributes into a set of attributes representing word occurrence (depending on the tokenizer) information from the text contained in the strings.
More
Capabilities
IDFTransform
False
TFTransform
False
attributeIndices
first-last
attributeNamePrefix
doNotOperateOnPerClassBasis
False
invertSelection
False
lowerCaseTokens
False
minTermFreq
1
normalizeDocLength
No normalization
outputWordCounts
False
periodicPruning
-1.0
stemmer
Choose
NullStemmer
stopwords
Users
tokenizer
Choose
WordTokenizer -delimiters
useStoplist
False
Open...
Save...
OK
Cancel

Example (4)



```
@relation departments_string

@attribute document_name string
@attribute document_content string
@attribute document_class string

@data

Anthropology, "anthropology anthropology anthropology co
archaeology and linguistics beyond these subfields conce
comparison the anthropology major provides students with
a range of careers from public service to marketing and
with faculty doing research students regularly attend pr
special programs include summer field schools in archaeo
sponsored diversity training institutes program of study
department website", A
Art, "art art the art department s undergraduate degree
printmaking sculpture ceramics and graphic design both c
encourages all forms of creative explorations via a deve
successful portfolio review of at least 10 works to beco
developed on an individual basis consistent with the goa
the samuel chen art center a gallery that offers regular
known artists sol lewitt cleve gray and robert cottingha
museums and galleries programs of study ba ms department
department website", B
Biology, "biology biological sciences the undergraduate
explore the discipline broadly specialized undergraduate
interpretation also available are specialized graduate p
various health and medical professions are advised prima
programs require a research project or internship many l
environmental rooms cell culture and protein purificatio
computer laboratory are available for research and instr
copernicus hall 332 phone 832 2645 department website fu
Chemistry, "chemistry chemistry the chemistry department
```



```
@attribute welte numeric
@attribute western numeric
@attribute willard numeric
@attribute wolff numeric
@attribute women numeric
@attribute works numeric
@attribute writing numeric
@attribute year numeric
@attribute york numeric

@data
{2 1,10 1,25 1,27 1,28 1,39 1,62 1,66 1,67 1,70 1,71 1
1,147 1,153 1,159 1,160 1,162 1,164 1,183 1,192 1,193
1,286 1,291 1,297 1,303 1,306 1,320 1,321 1,333 1,338
1,393 1,403 1,404 1,411 1,414 1,420 1,425 1,431 1,433
{26 1,39 1,52 1,56 1,61 1,62 1,71 1,85 1,96 1,104 1,11
1,283 1,288 1,310 1,320 1,322 1,324 1,328 1,346 1,366
1,478 1,485 1,486 1,487 1,491 1,494 1,495 1,509 1,510
1,558 1,561 1,566 1,568 1,571 1,578 1,579 1,582 1,587
1,629 1,630 1,631 1,632 1,636 1,646 1,653 1,654 1,661
1,725 1,734 1,735 1,739 1,754 1,755 1,756 1,762 1}
{11 1,21 1,26 1,27 1,29 1,39 1,51 1,52 1,56 1,58 1,62
1,141 1,148 1,152 1,153 1,160 1,181 1,186 1,188 1,191
1,258 1,260 1,269 1,270 1,283 1,284 1,292 1,302 1,310
1,375 1,380 1,392 1,393 1,395 1,396 1,398 1,399 1,404
{12 1,23 1,26 1,27 1,30 1,60 1,69 1,73 1,86 1,89 1,94
1,151 1,153 1,172 1,177 1,181 1,214 1,218 1,219 1,225
1,366 1,368 1,382 1,399 1,402 1,404 1,408 1,413 1,415
{0 1,26 1,39 1,41 1,52 1,60 1,61 1,62 1,72 1,85 1,86 1
1,218 1,228 1,240 1,248 1,258 1,283 1,309 1,310 1,320
1,399 1,420 1,421 1,431 1,433 1,435 1,444 1,445 1,446
1,547 1,554 1,562 1,577 1,581 1,586 1,588 1,590 1,597
1,704 1,719 1,721 1,723 1,741 1,744 1,745 1,752 1}
```

Attribute selection (1)

- The most useful part of this is attribute selection (also called feature selection)
- Select relevant attributes
- Remove redundant and/or irrelevant attributes

Attribute selection (1)

Objectives:

- Simpler model
  - More transparent
  - Easier to interpret
- Faster model induction
- Structural knowledge
  - Knowing which attributes are important may be inherently important to the application

Attribute selection (2)

Search Method	Attribute Evaluator: Attributes	Attribute Evaluator: Subsets of attributes
Best First		YES
Greedy Stepwise		YES
Ranker	YES	

Attribute selection (3)

Filters:

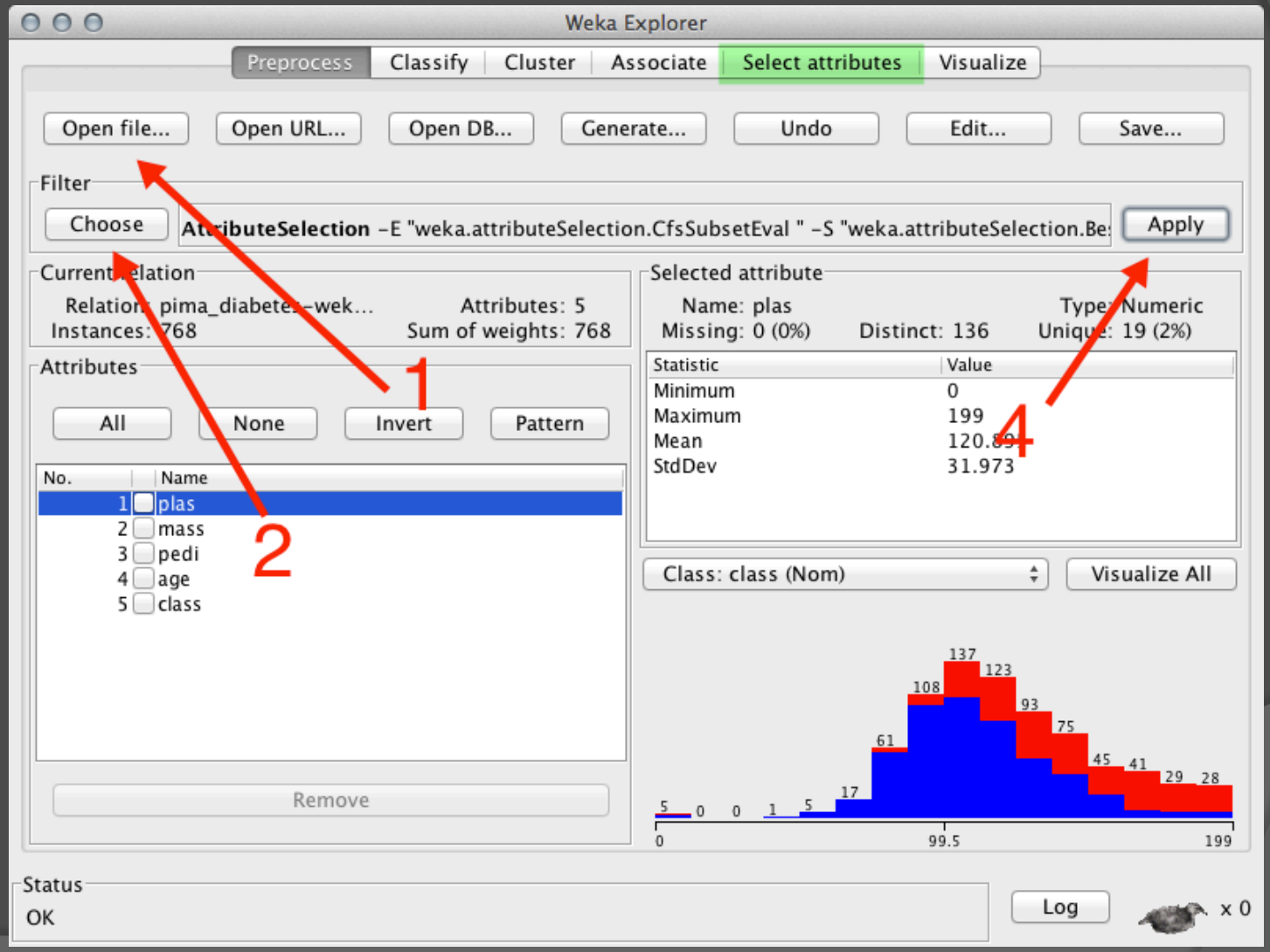
- Ranked list of attributes
  - Typical when each attribute is evaluated individually
- A selected subset of attributes
  - Greedy Stepwise and Best first
  - Random search such as genetic algorithm

Example (1)

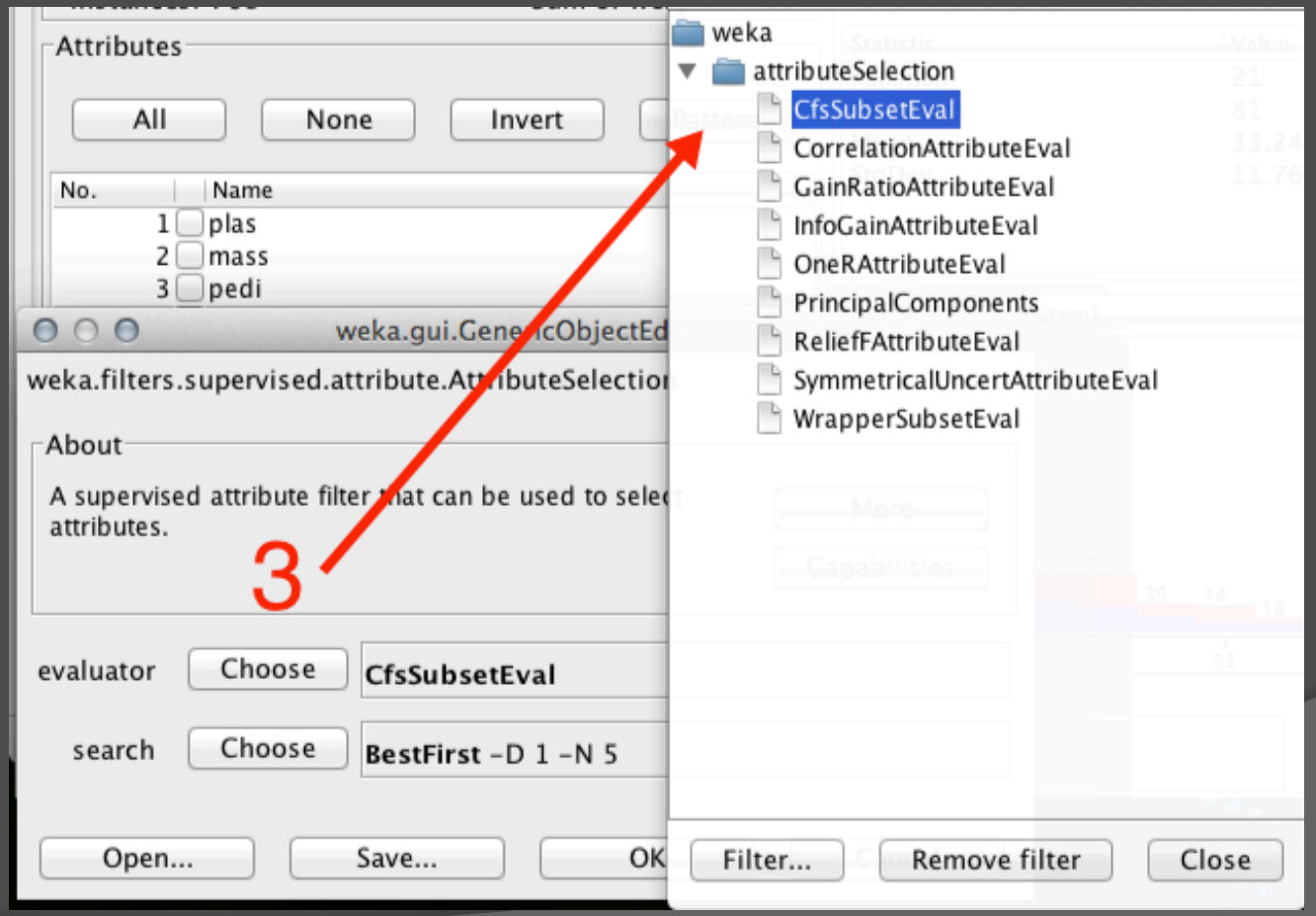
1. Open *diabetes.arff* 1
2. Choose: *filters* → *supervised* → *attribute* → *AttributeSelection* 2

   It's possible to use "*SelectAttributes*" tab
3. Set some options 3
4. Click on *Apply* 4

Example (2)


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
AttributeSelection -E "weka.attributeSelection.CfsSubsetEval " -S "weka.attributeSelection.Be
Apply
Current relation
Relation: pima_diabete-wek...
Instances: 768
Attributes: 5
Sum of weights: 768
Attributes
All
None
Invert
Pattern
No.
Name
1 plas
2 mass
3 pedi
4 age
5 class
Remove
Selected attribute
Name: plas
Missing: 0 (0%)
Distinct: 136
Type: Numeric
Unique: 19 (2%)
Statistic
Value
Minimum 0
Maximum 199
Mean 120.89
StdDev 31.973
Class: class (Nom)
Visualize All
137
123
108
93
75
61
45
41
29
28
17
5
0
0
1
5
0
99.5
199
Status
OK
Log
x 0
1
2
4

Example (3)


Attributes
All
None
Invert
No.
Name
1 plas
2 mass
3 pedi
weka.gui.GenericObjectEd
weka.filters.supervised.attribute.AttributeSelection
About
A supervised attribute filter that can be used to select attributes.
3
evaluator
Choose
CfsSubsetEval
search
Choose
BestFirst -D 1 -N 5
Open...
Save...
OK
weka
attributeSelection
CfsSubsetEval
CorrelationAttributeEval
GainRatioAttributeEval
InfoGainAttributeEval
OneRAttributeEval
PrincipalComponents
ReliefFAttributeEval
SymmetricalUncertAttributeEval
WrapperSubsetEval
Filter...
Remove filter
Close

Exercise

- Discuss the results after applying *AttributeSelection* on *diabetes.arff* using "default" parameters
- Change some parameters and compare the results
- Apply a classification algorithm (e.g., J48) aux datasets with/without attribute selection. Compare the results

Conclusion

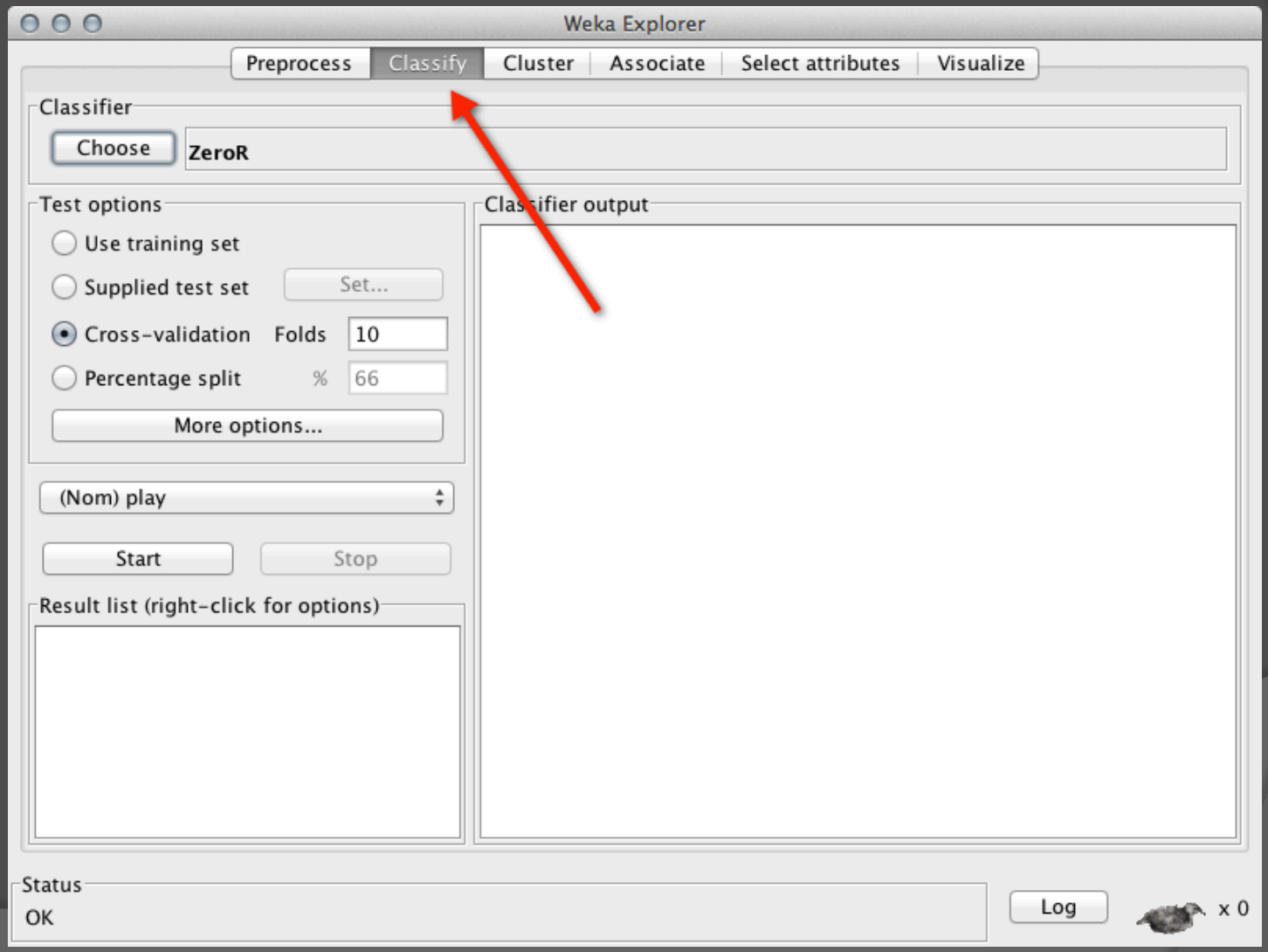
Data preprocessing is very important, and it has an important impact on the quality of learning process

Classifiers

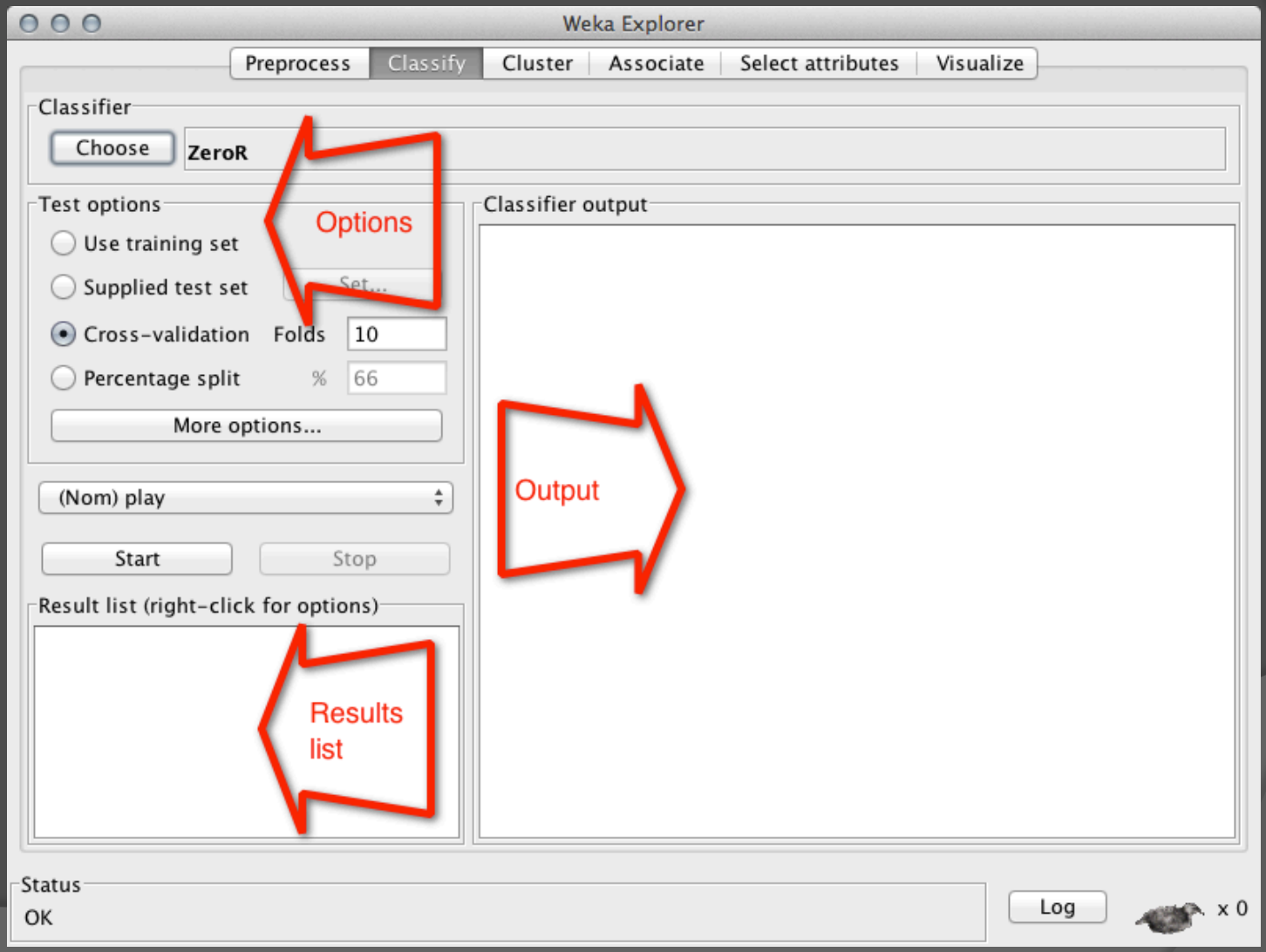
Classifiers in Weka (1)

- Classifiers in WEKA are models for predicting nominal or numeric quantities
- Classification algorithms include:
  - Decision trees
  - Naïve Bayes Classification
  - Support vector machine (SVM)
  - Multi-layer perceptron
  - Bayes network, etc.
- Meta-classifiers:
  - Combination
  - Bagging
  - Boosting, etc.

Classifiers in Weka (2)



Classifiers in Weka (3)



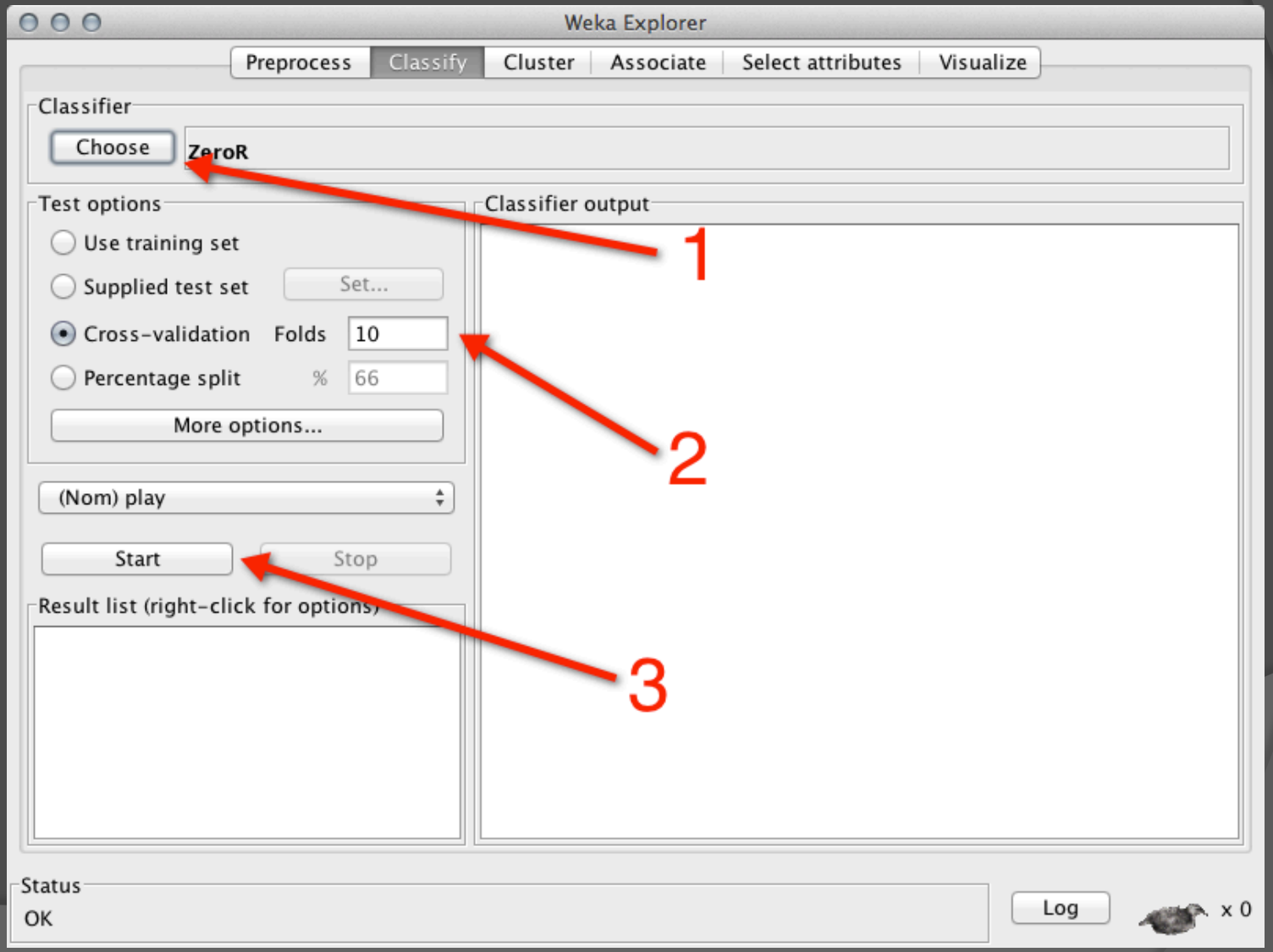
Classifiers : options

- ***Training set:*** the classifier is evaluated on how well it predicts the class of the instances it was trained on
- ***Supplied test set:*** the classifier is evaluated on how well it predicts the class of a set of instances loaded from a file
- ***Cross-Validation:*** the classifier is evaluated by cross-validation, using the number of folds that are entered in the Folds text field
- ***Percentage Split:*** the classifier is evaluated on how well it predicts a certain percentage of the data which is held out for testing

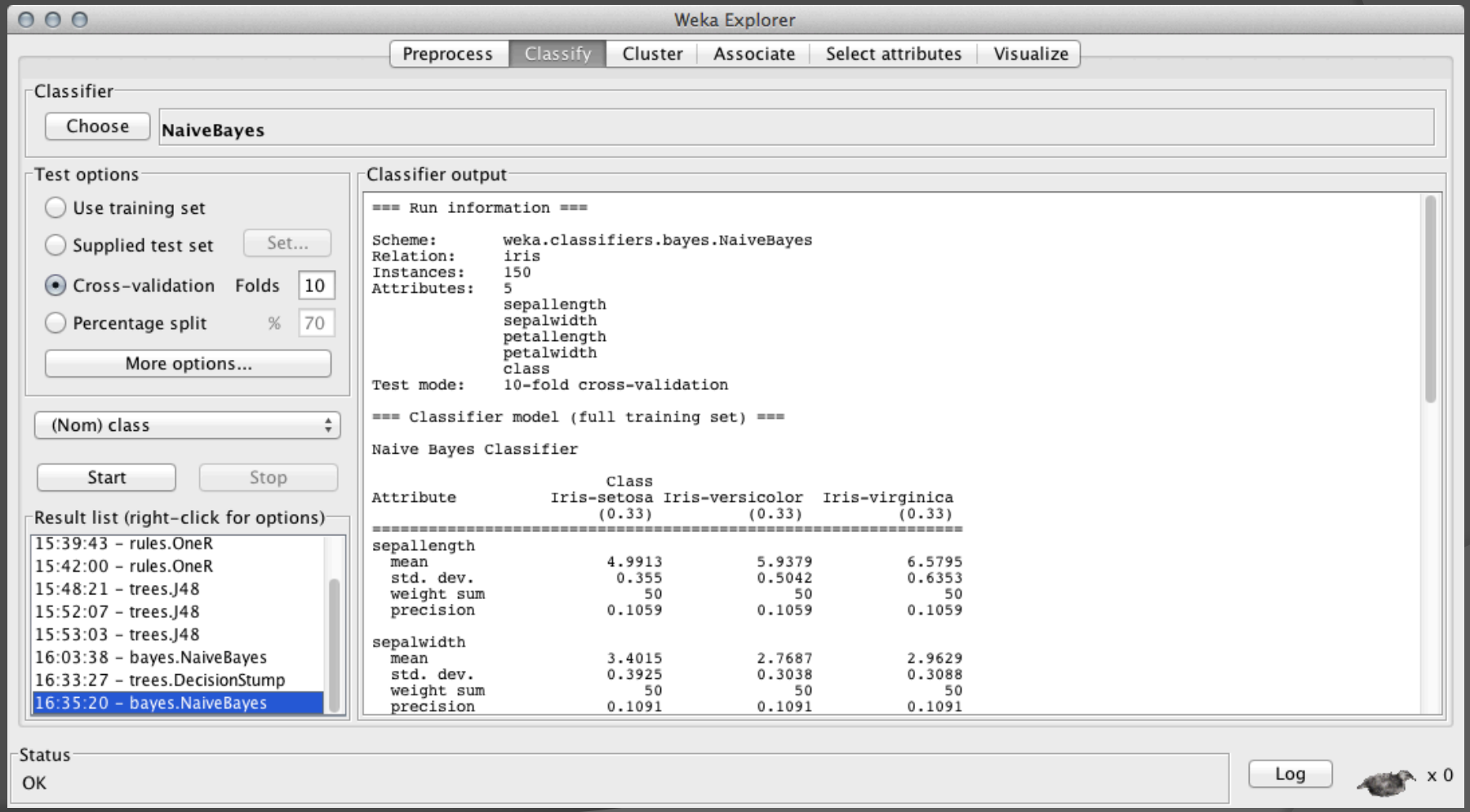
Example (1)

1. Open iris.arff
2. Go to *Classify* tab
3. Choose a classifier : *Classifier* → *Bayes* → *NaiveBayes* 1
4. Set *Cross-Validation* value to 10 2
5. Click on *Start* button 3

Example (2)


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
ZeroR
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
Classifier output
(Nom) play
Start
Stop
Result list (right-click for options)
Status
OK
Log
x 0
1
2
3

Example (3)


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
70
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
15:39:43 - rules.OneR
15:42:00 - rules.OneR
15:48:21 - trees.J48
15:52:07 - trees.J48
15:53:03 - trees.J48
16:03:38 - bayes.NaiveBayes
16:33:27 - trees.DecisionStump
16:35:20 - bayes.NaiveBayes
Classifier output
=== Run information ===
Scheme: weka.classifiers.bayes.NaiveBayes
Relation: iris
Instances: 150
Attributes: 5
sepallength
sepalwidth
petallength
petalwidth
class
Test mode: 10-fold cross-validation
=== Classifier model (full training set) ===
Naive Bayes Classifier
Class
Attribute Iris-setosa Iris-versicolor Iris-virginica
(0.33) (0.33) (0.33)
sepallength
mean 4.9913 5.9379 6.5795
std. dev. 0.355 0.5042 0.6353
weight sum 50 50 50
precision 0.1059 0.1059 0.1059
sepalwidth
mean 3.4015 2.7687 2.9629
std. dev. 0.3925 0.3038 0.3088
weight sum 50 50 50
precision 0.1091 0.1091 0.1091
Status
OK
Log
x 0

Interpretation of results (1)

```
=== Evaluation on test split ===

Time taken to test model on training split: 0.04 seconds

=== Summary ===

Correctly Classified Instances          43               95.5556 %
Incorrectly Classified Instances         2                4.4444 %
Kappa statistic                          0.9331
Mean absolute error                      0.0375
Root mean squared error                  0.158
Relative absolute error                  8.422  %
Root relative squared error             33.4987 %
Coverage of cases (0.95 level)          97.7778 %
Mean rel. region size (0.95 level)      37.037  %
Total Number of Instances               45

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 1        0        1          1       1          1        1         1         Iris-setosa
                 1        0.069    0.889      1       0.941      0.91     0.987     0.976     Iris-versicolor
                 0.867    0        1          0.867   0.929      0.901    0.987     0.979     Iris-virginica
Weighted Avg.    0.956    0.025    0.96       0.956   0.955      0.935    0.991     0.984

=== Confusion Matrix ===

  a  b  c   <-- classified as
 14  0  0 |  a = Iris-setosa
  0 16  0 |  b = Iris-versicolor
  0  2 13 |  c = Iris-virginica
```

1

2

3

Interpretation of results (2)

=== Summary === 1

- This gives the error levels when applying the classifier.
- The most important figures here are the numbers of correctly and incorrectly classified instances.
- With the exception of the Kappa statistic, the remaining statistics compute various error measures based on the class probabilities assigned by the tree.

Interpretation of results (3)

=== Detailed Accuracy By Class === 2

- The percentage of correctly classified instances is often called accuracy or sample accuracy.
- Accuracy has some disadvantages as a performance estimate (not chance corrected, not sensitive to class distribution)
- Area under the ROC curve is an interesting measure.

Interpretation of results (4)

=== Confusion Matrix === 3

- This shows for each class, how instances from that class received the various classifications.
- a, b and c representing the class labels. Here there were 45 instances, so the percentages and raw numbers add up, aa+bb+cc = 43, ab+ba+ac+ca+… = 2 .

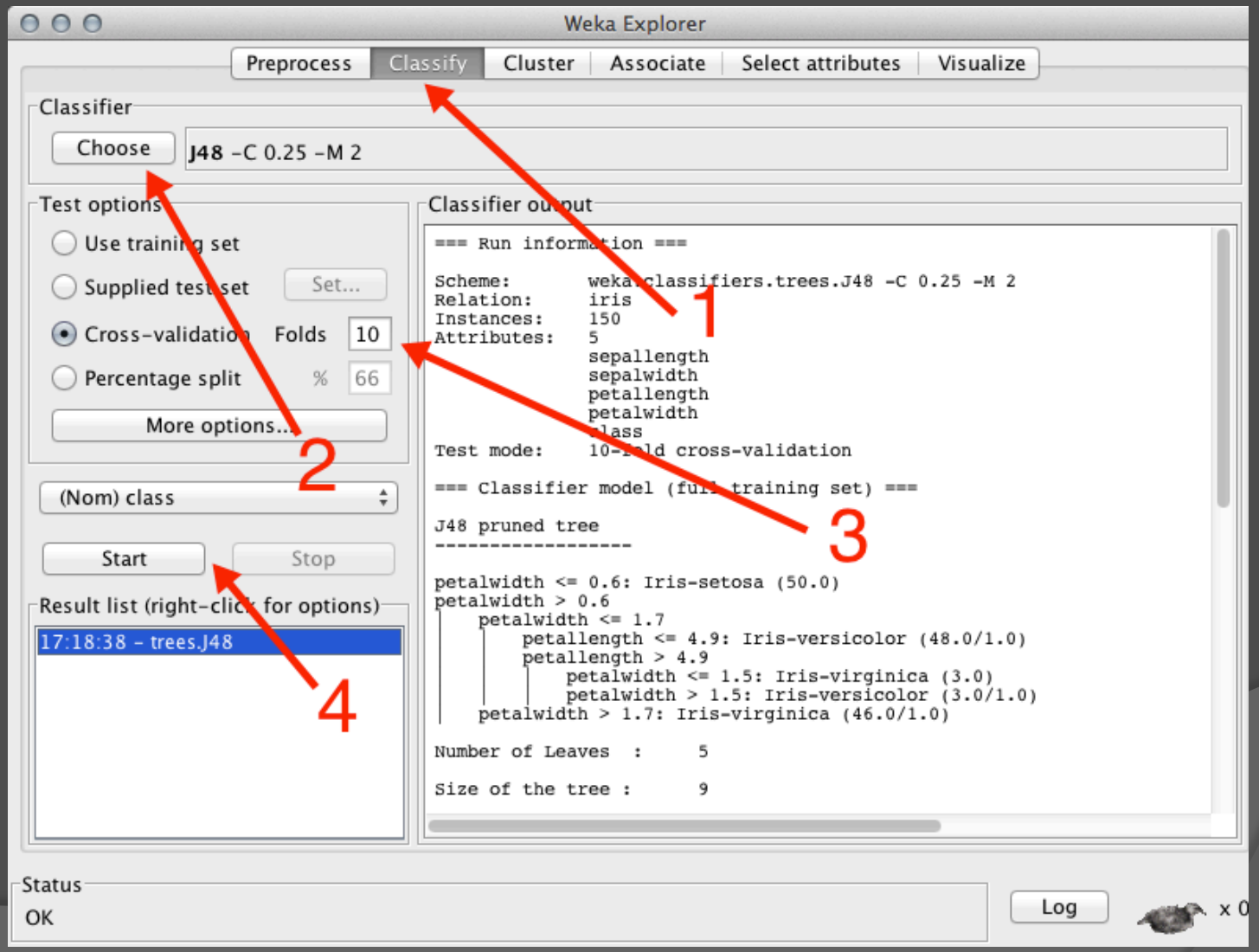
Decision trees

- Learning by partitioning
- We want to build homogeneous subgroups in terms of a nominal variable to be predicted (target) using a set of discriminant variables
- Result must be readable
- It must be able to automatically select discriminating variables

Example (1)

1. Open iris.arff
2. Go to *Classify* tab
3. Choose a classifier : *Classifier* → *trees* → *J48* 1
4. Set *Cross-Validation* value to 10 2
5. Click on *Start* button 3

Example (2)


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
Classifier output
=== Run information ===
Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: iris
Instances: 150
Attributes: 5
sepallength
sepalwidth
petallength
petalwidth
class
Test mode: 10-fold cross-validation
=== 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
Size of the tree : 9
(Nom) class
Start
Stop
Result list (right-click for options)
17:18:38 - trees.J48
Status
OK
Log
x 0
1
2
3
4

Interpretation of tree

J48 pruned tree

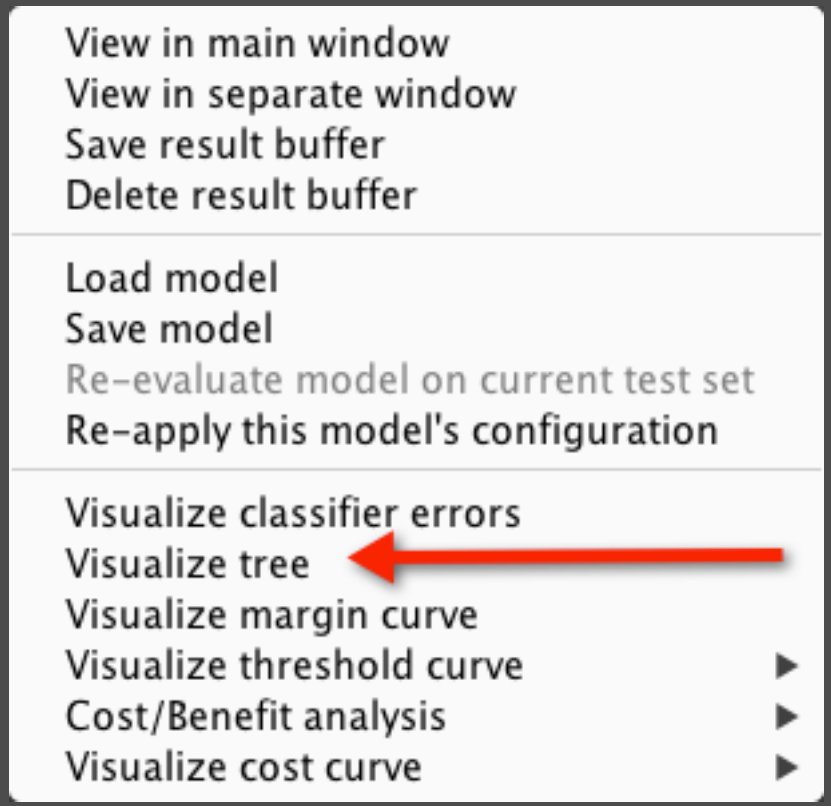
---

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
Size of the tree : 9

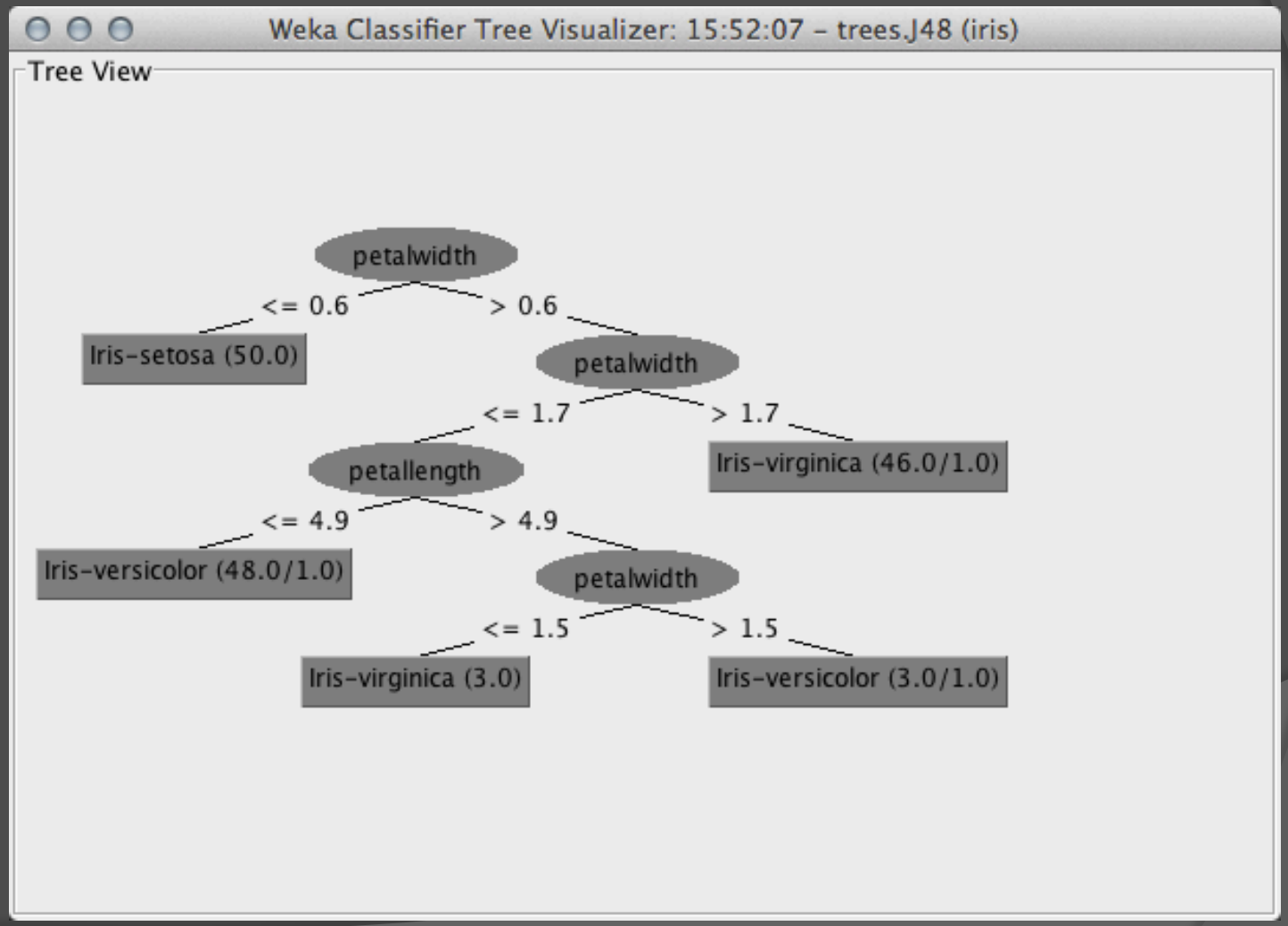
Interpretation of tree

- This indicates how the classifier uses the attributes to make a decision.
- The leaf nodes indicate which class an instance will be assigned to should that node be reached.
- The numbers in brackets after the leaf nodes indicate the number of instances assigned to that node, followed by how many of those instances are incorrectly classified as a result.

Tree visualization (1)



Tree visualization (2)



Exercise 1

- Using the Weka explorer environment and load the training file *“diabetes.arff”* Perform classification with Naive Bayes, Decision Tree and K-NN (with K=3) Use the following setting :
  - 10 Fold Cross validation
  - 70% Training and 30% Test (percentage split)
- Build a comparative table with the 2 different settings and the 3 classifiers and comment the results

Exercise 2

- Using the Weka explorer environment and load the training file *“diabetes.arff”* Perform classification with K-NN with different values of K (3,5,7,9,11,13) with 10 Fold Cross validation.
- Put the accuracy results in a table and comment the results. Emphasize how the results change in relation to the value of K

Exercise 3

- Show the tree decision for “weather.arff data using the following parameters:
  - Method: J48
  - Cross-validation: fixed on 5 and 10
- Discuss the results (figure)

Clustering

Clustering

- The process of grouping physical or abstract objects into classes of similar objects i.e., given a set of records (instances, examples, objects, observations, …), organize them into clusters (groups, classes)
- Works with both discrete and numerical data*

Classification vs clustering

- ***Classification: Supervised learning***

Learns a method for predicting the instance class from pre-labeled (classified) instances

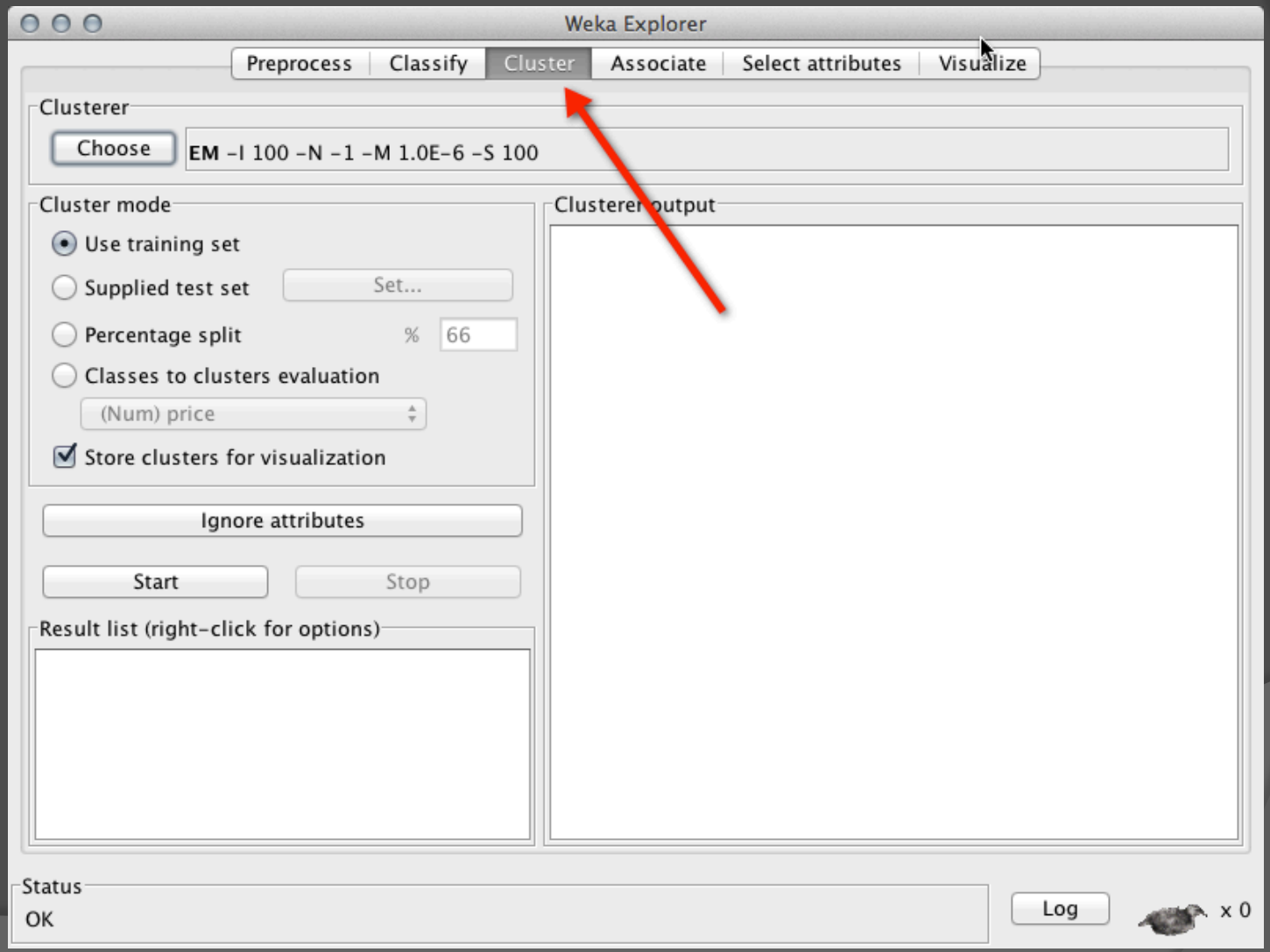
- ***Clustering: Unsupervised learning***

Finds “natural” grouping of instances given un-labeled data

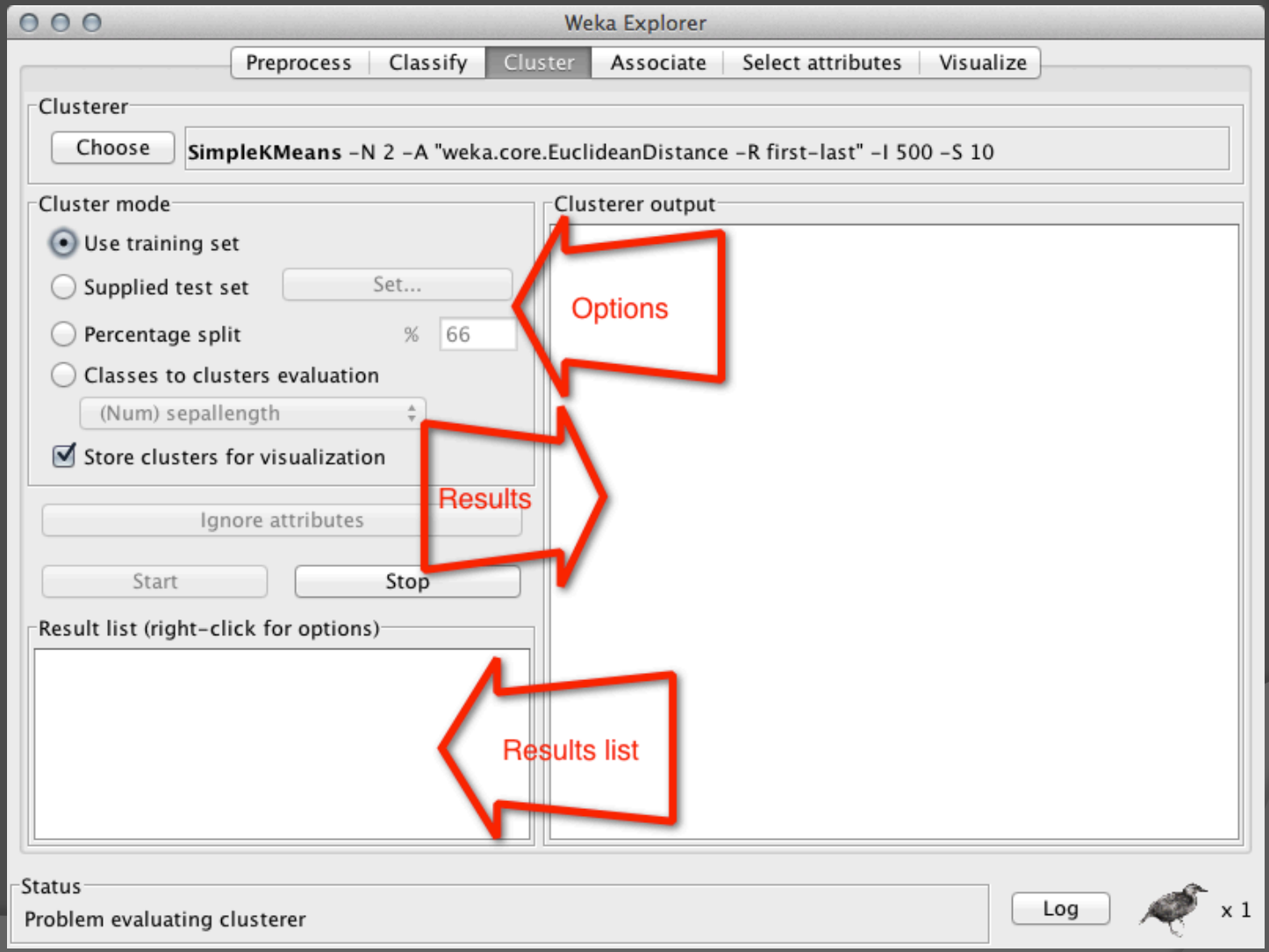
Cluster definition

- A cluster is a subset of objects which are “similar”
- A subset of objects such that the distance between any two objects in the cluster is less than the distance between any object in the cluster and any object not located inside it
- A connected region of a multidimensional space containing a relatively high density of objects

Clustering with Weka (1)



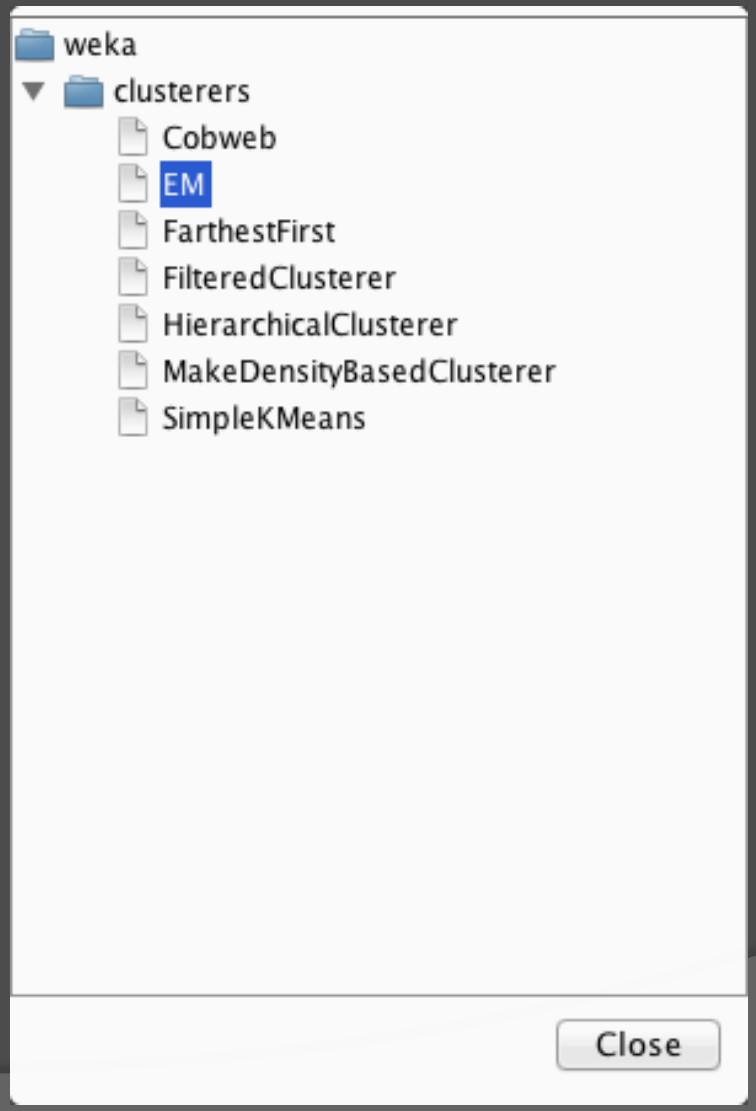
Clustering with Weka (2)



Clustering: options

- ***Use training set:*** After generating the clustering Weka classifies the training instances into clusters according to the cluster representation and computes the percentage of instances falling in each cluster
- ***Supplied test set or Percentage split:*** Weka can evaluate clusterings on separate test data if the cluster representation is probabilistic (e.g. for EM).
- ***Classes to clusters evaluation:*** In this mode Weka first ignores the class attribute and generates the clustering. Then during the test phase it assigns classes to the clusters, based on the majority value of the class attribute within each cluster (e.g. for k-Means)

Clustering with Weka (2)



EM

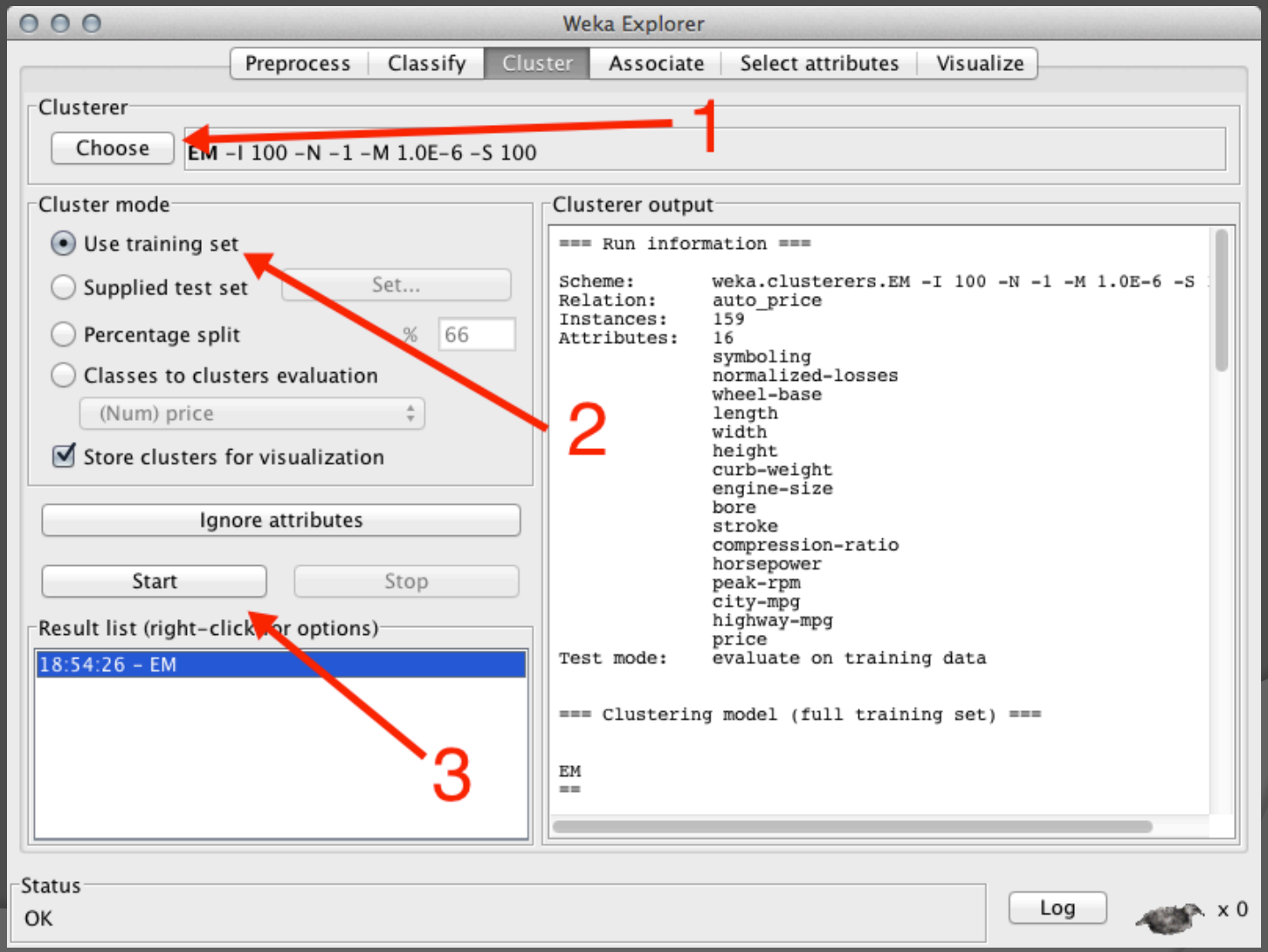
- The EM clustering scheme generates probabilistic descriptions of the clusters in terms of mean and standard deviation for the **numeric attributes** and value **counts*** for the nominal ones

* incremented by 1 and modified with a small value to avoid zero probabilities

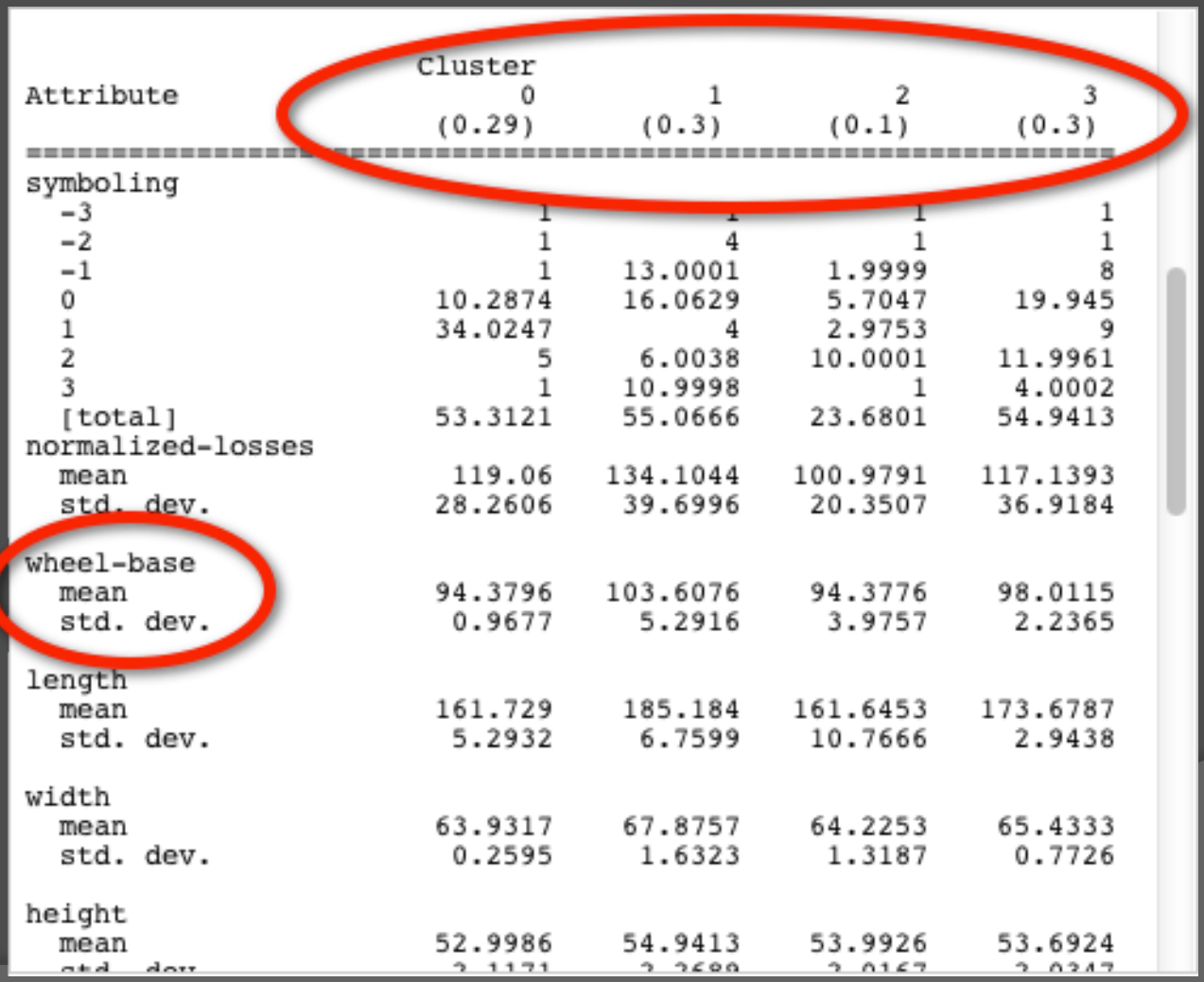
Example (1)

1. Open auto_price.arff
2. Go to *Clustering* tab
3. Choose a clustering method: *EM* 1
4. Set *Use training set* 2
5. Click on *Start* button 3

Example (2)


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
EM -I 100 -N -1 -M 1.0E-6 -S 100
1
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Num) price
Store clusters for visualization
2
Ignore attributes
Start
Stop
Result list (right-click for options)
18:54:26 - EM
3
Clusterer output
=== Run information ===
Scheme: weka.clusterers.EM -I 100 -N -1 -M 1.0E-6 -S
Relation: auto_price
Instances: 159
Attributes: 16
symboling
normalized-losses
wheel-base
length
width
height
curb-weight
engine-size
bore
stroke
compression-ratio
horsepower
peak-rpm
city-mpg
highway-mpg
price
Test mode: evaluate on training data
=== Clustering model (full training set) ===
EM
==
Status
OK
Log
x 0

Example (3)

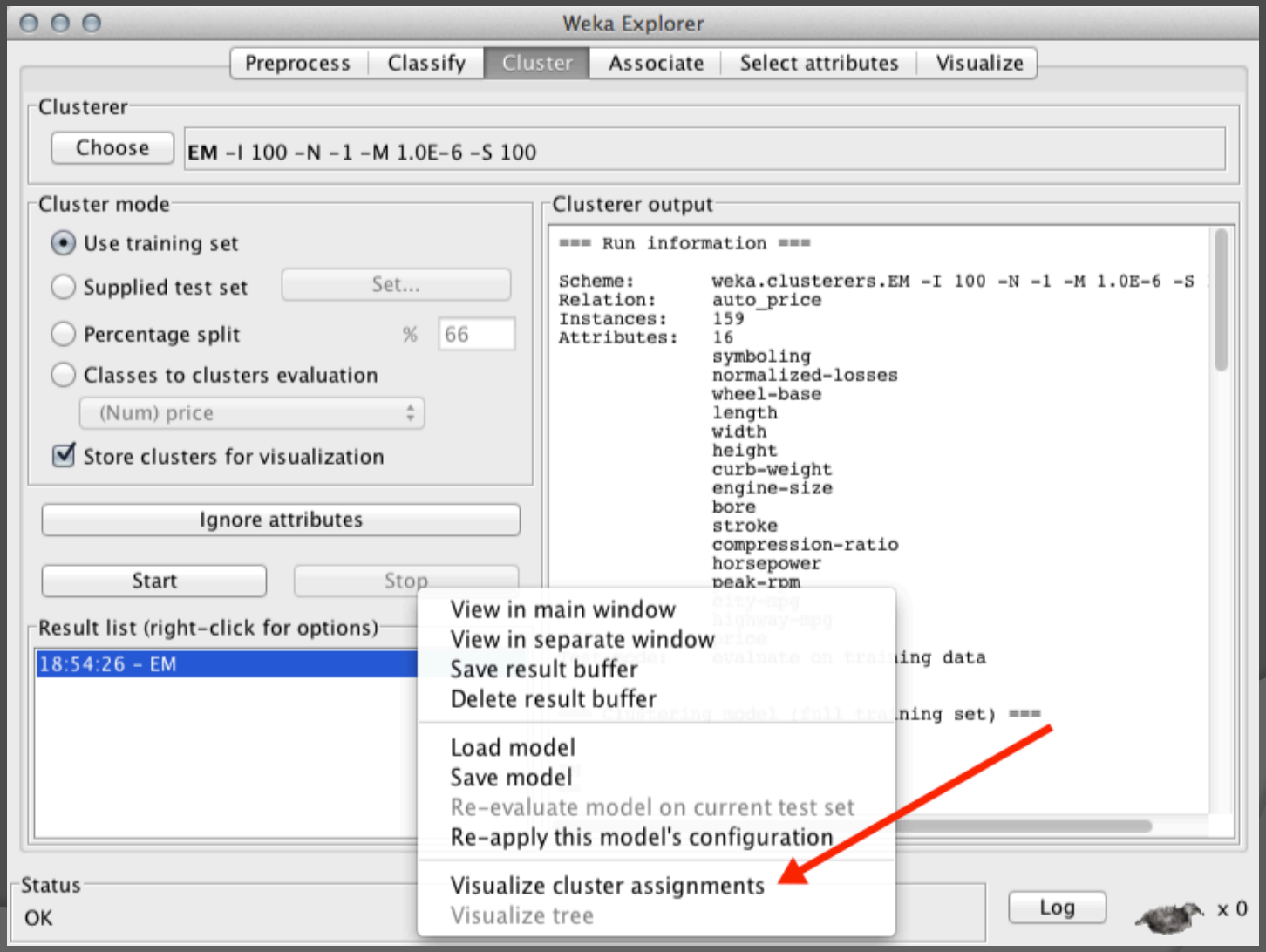


Attribute	Cluster 0 (0.29)	1 (0.3)	2 (0.1)	3 (0.3)
symboling				
-3	1	1	1	1
-2	1	4	1	1
-1	1	13.0001	1.9999	8
0	10.2874	16.0629	5.7047	19.945
1	34.0247	4	2.9753	9
2	5	6.0038	10.0001	11.9961
3	1	10.9998	1	4.0002
[total]	53.3121	55.0666	23.6801	54.9413
normalized-losses				
mean	119.06	134.1044	100.9791	117.1393
std. dev.	28.2606	39.6996	20.3507	36.9184
wheel-base				
mean	94.3796	103.6076	94.3776	98.0115
std. dev.	0.9677	5.2916	3.9757	2.2365
length				
mean	161.729	185.184	161.6453	173.6787
std. dev.	5.2932	6.7599	10.7666	2.9438
width				
mean	63.9317	67.8757	64.2253	65.4333
std. dev.	0.2595	1.6323	1.3187	0.7726
height				
mean	52.9986	54.9413	53.9926	53.6924
std. dev.	2.1171	2.2689	2.0167	2.0347

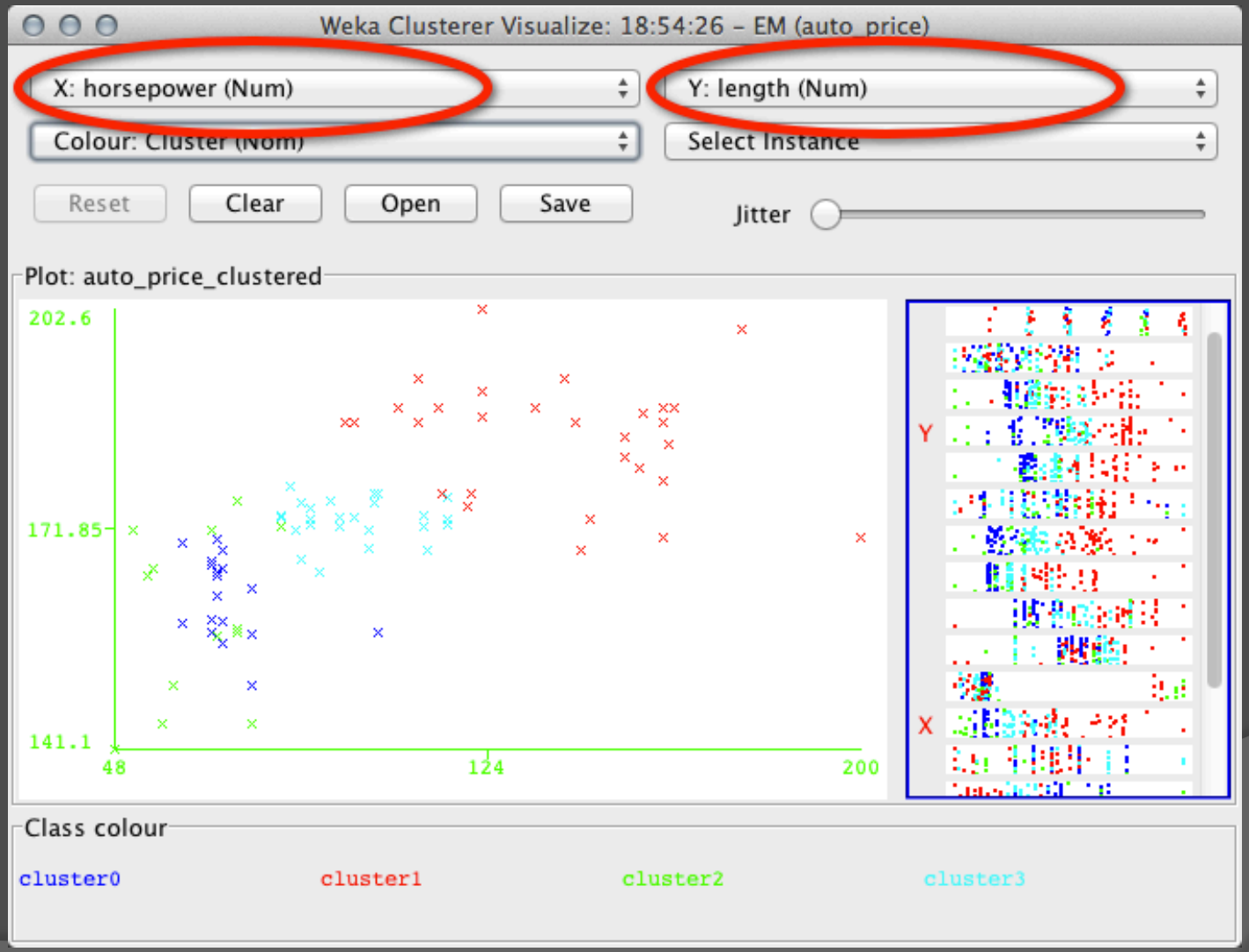
Example (4)

```
peak rpm
  mean                 5255.9762  5046.5452  4816.9411  5147.2922
  std. dev.             355.5054   429.7359   401.8338   543.7446

city-mpg
  mean                   30.9262     20.358    35.4016    25.3582
  std. dev.               3.0629     3.0307     6.7859      1.614

highway-mpg
  mean                   36.7524      25.518    41.1052    31.0112
  std. dev.               3.1128     3.1878       7.46     2.2009

price
  mean                 6959.1625 18703.0111   7412.919   9906.699
  std. dev.             902.7482  5306.2492  1423.7368  1965.4932


Time taken to build model (full training data) : 1.88 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       46 ( 29%)
1       48 ( 30%)
2       17 ( 11%)
3       48 ( 30%)


Log likelihood: -53.41921
```

Example (5)


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
EM -I 100 -N -1 -M 1.0E-6 -S 100
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Num) price
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
18:54:26 - EM
Clusterer output
=== Run information ===
Scheme: weka.clusterers.EM -I 100 -N -1 -M 1.0E-6 -S
Relation: auto_price
Instances: 159
Attributes: 16
symboling
normalized-losses
wheel-base
length
width
height
curb-weight
engine-size
bore
stroke
compression-ratio
horsepower
peak-rpm
View in main window
View in separate window
Save result buffer
Delete result buffer
Load model
Save model
Re-evaluate model on current test set
Re-apply this model's configuration
Visualize cluster assignments
Visualize tree
Status
OK
Log
x 0

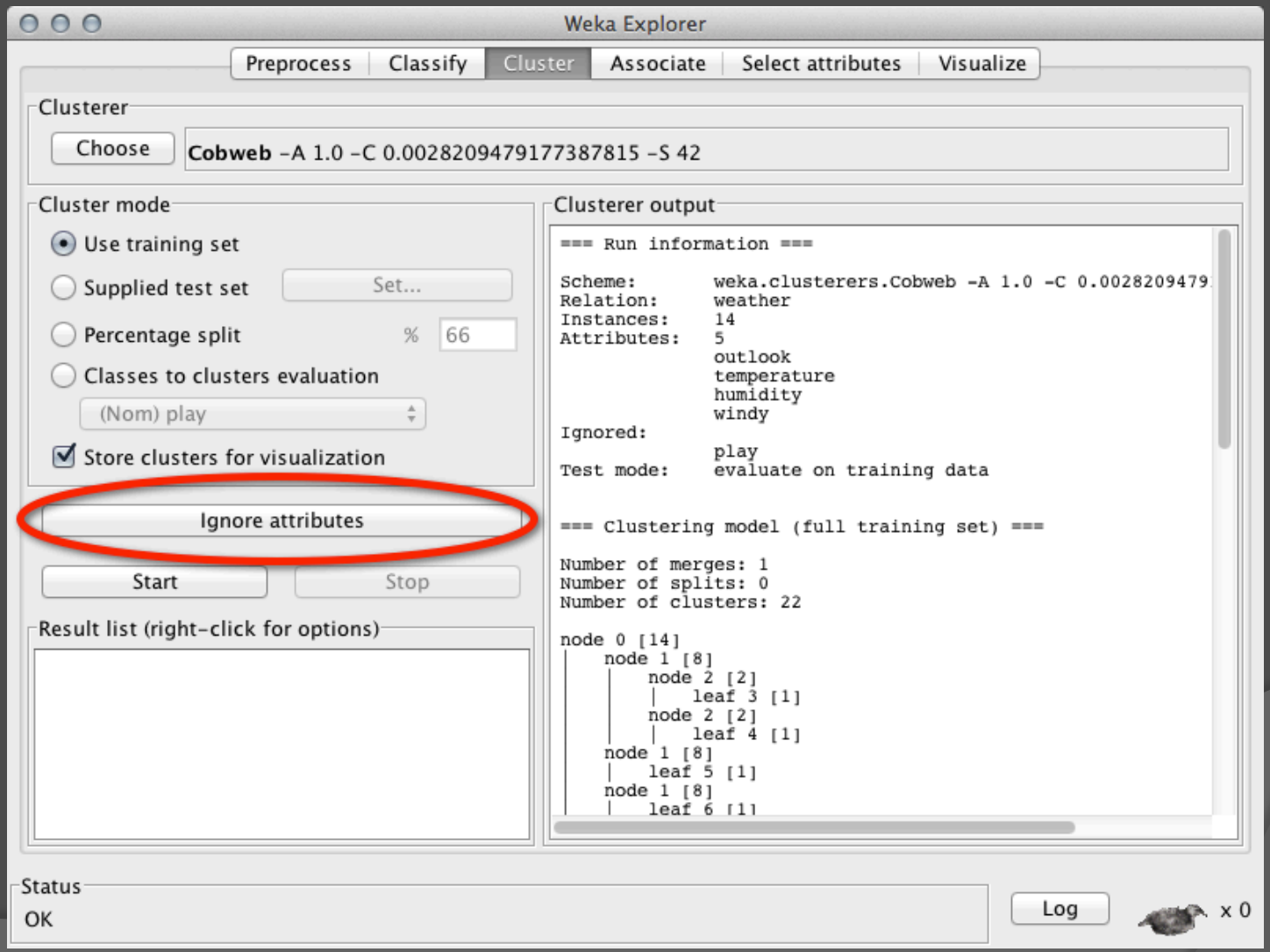
Example (6)


Weka Clusterer Visualize: 18:54:26 - EM (auto_price)
X: horsepower (Num)
Y: length (Num)
Colour: Cluster (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: auto_price_clustered
202.6
171.85
141.1
48
124
200
Y
X
Class colour
cluster0
cluster1
cluster2
cluster3

Cobweb (1)

- Cobweb generates hierarchical clustering, where clusters are described probabilistically. The class attribute is ignored in order to allow later classes to clusters evaluation

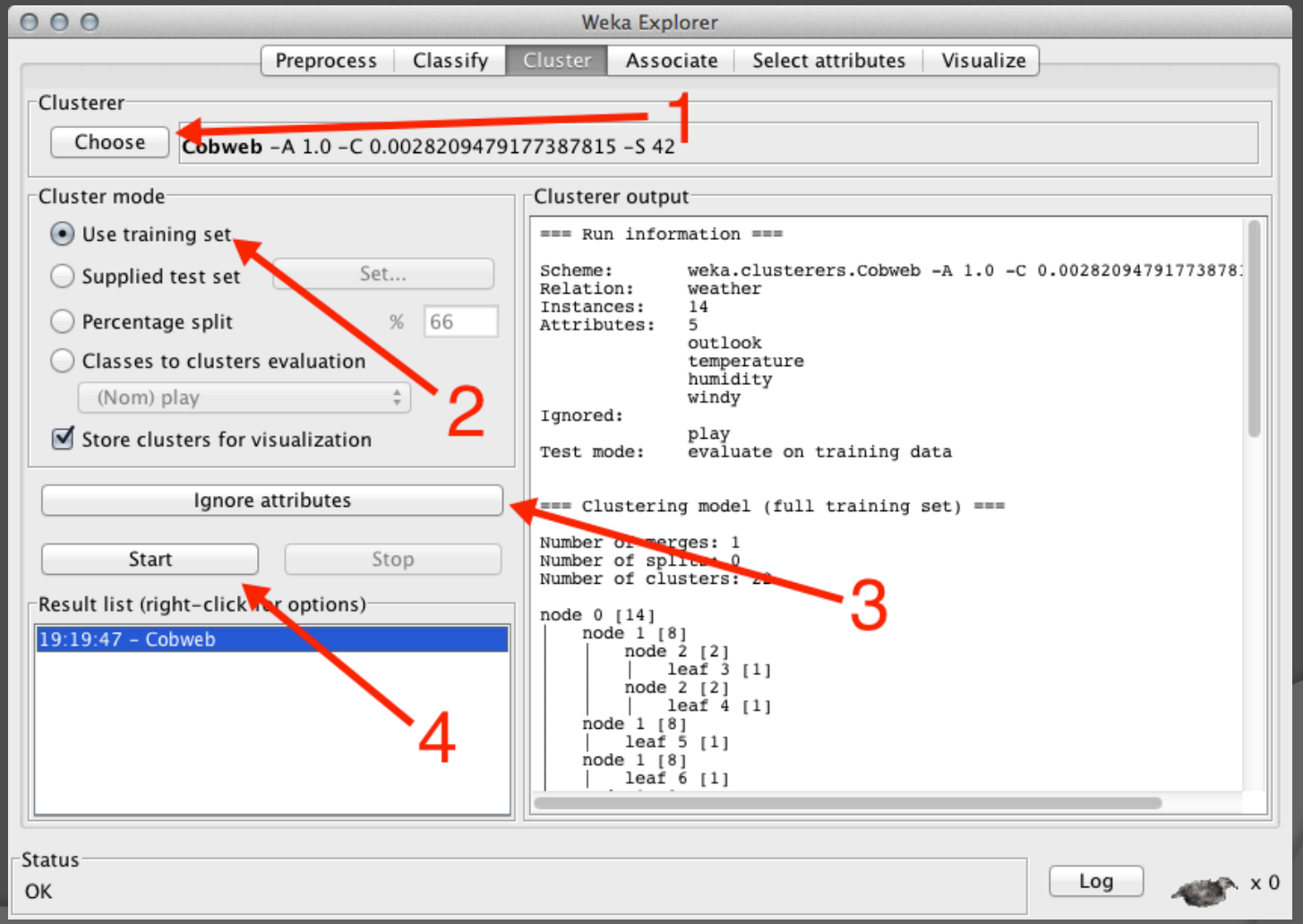
Cobweb (2)


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
Cobweb -A 1.0 -C 0.0028209479177387815 -S 42
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) play
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
Clusterer output
=== Run information ===
Scheme: weka.clusterers.Cobweb -A 1.0 -C 0.0028209479
Relation: weather
Instances: 14
Attributes: 5
outlook
temperature
humidity
windy
Ignored:
play
Test mode: evaluate on training data
=== Clustering model (full training set) ===
Number of merges: 1
Number of splits: 0
Number of clusters: 22
node 0 [14]
node 1 [8]
node 2 [2]
leaf 3 [1]
node 2 [2]
leaf 4 [1]
node 1 [8]
leaf 5 [1]
node 1 [8]
leaf 6 [1]
Status
OK
Log
x 0

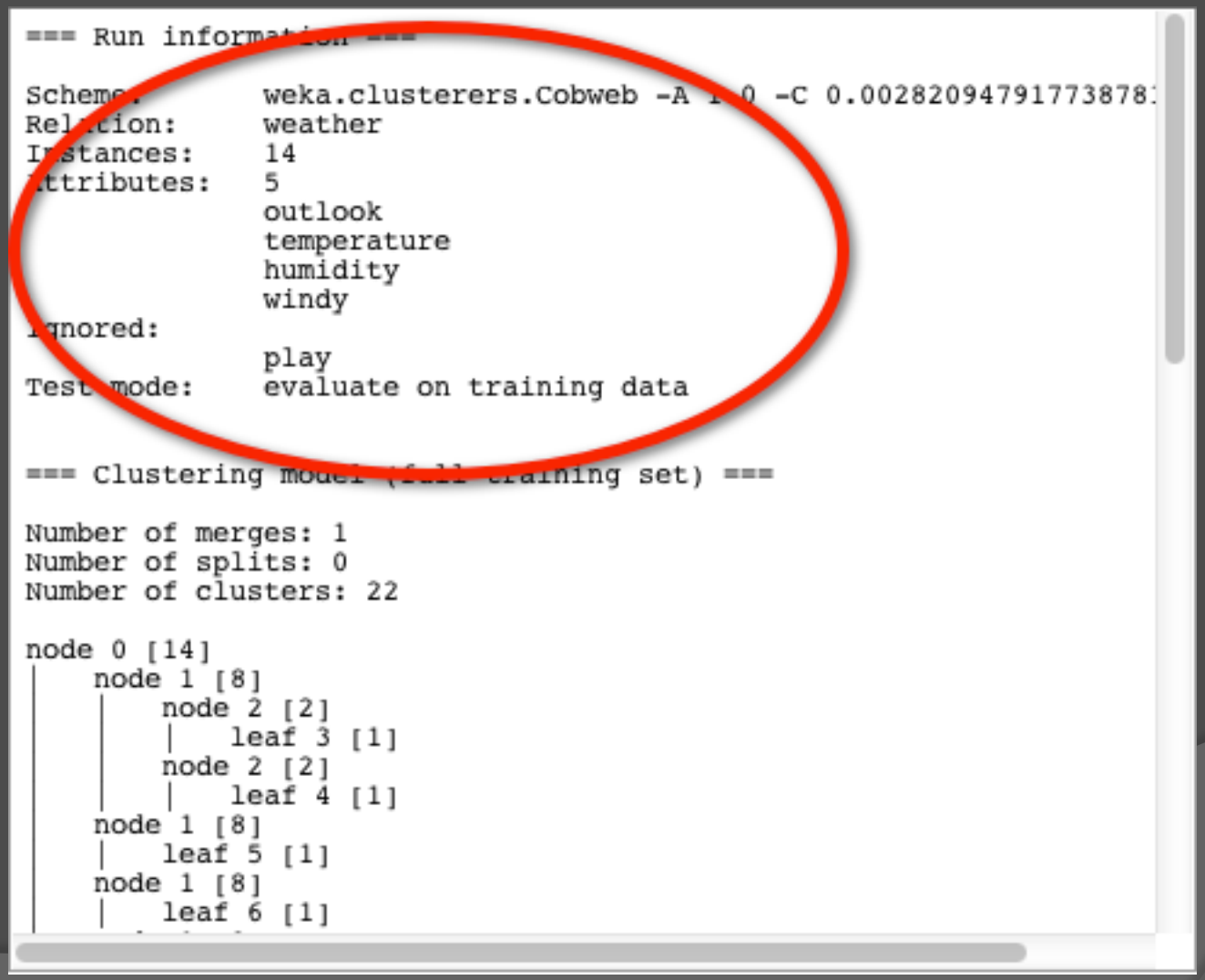
Example (1)

1. Open weather.arff
2. Go to *Clustering* tab
3. Choose a clustering method: *Cobweb* 1
4. Set *Use training set* 2
5. Chose the “class” attribute on *Ignore Attribute* panel *3*
6. Click on *Start* button 4

Example (2)


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
Cobweb -A 1.0 -C 0.0028209479177387815 -S 42
1
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) play
Store clusters for visualization
2
Ignore attributes
Start
Stop
Result list (right-click for options)
19:19:47 - Cobweb
4
Clusterer output
=== Run information ===
Scheme: weka.clusterers.Cobweb -A 1.0 -C 0.00282094791773878
Relation: weather
Instances: 14
Attributes: 5
outlook
temperature
humidity
windy
Ignored:
play
Test mode: evaluate on training data
=== Clustering model (full training set) ===
Number of merges: 1
Number of splits: 0
Number of clusters:
node 0 [14]
node 1 [8]
node 2 [2]
leaf 3 [1]
node 2 [2]
leaf 4 [1]
node 1 [8]
leaf 5 [1]
node 1 [8]
leaf 6 [1]
3
Status
OK
Log
x 0

Meaning of results (1)



Meaning of results (2)

```
=== Clustering model (full training set) ===

Number of merges: 1
Number of splits: 0
Number of clusters: 22

node 0 [14]
|    node 1 [8]
|    |    node 2 [2]
|    |    |   leaf 3 [1]
|    |    node 2 [2]
|    |    |   leaf 4 [1]
|    node 1 [8]
|    |    leaf 5 [1]
|    node 1 [8]
|    |    leaf 6 [1]
|    node 1 [8]
|    |    node 7 [3]
|    |    |   leaf 8 [1]
|    |    node 7 [3]
|    |    |   leaf 9 [1]
|    |    node 7 [3]
|    |    |   leaf 10 [1]
|    node 1 [8]
|    |    leaf 11 [1]
node 0 [14]
|    node 12 [6]
|    |    node 13 [2]
|    |    |   leaf 14 [1]
|    |    node 13 [2]
|    |    |   leaf 15 [1]
```

Meaning of results (3)

- Node N or leaf N represents a subcluster, whose parent cluster is N
- The clustering tree structure is shown as a *horizontal tree*, where subclusters are aligned at the same column
- The *root cluster is 0*. Each line with node 0 defines a subcluster of the root

Meaning of results (4)

- The number in *square brackets* after node N represents the number of instances in the parent cluster N
- Clusters with [1] at the end of the line are *instances*
- To view the clustering tree right click on the last line in the result list window and then select *Visualize tree*

Exercise 1

- *Right click* on the last line in the result list window
- Visualize cluster assignments - you get the Weka cluster *visualize* window
- Put *Instance_number* on *X* and *Cluster* on *Y*
- Click on *Save* and choose a file name (*.arff)
- Explore the *arff* file and comment

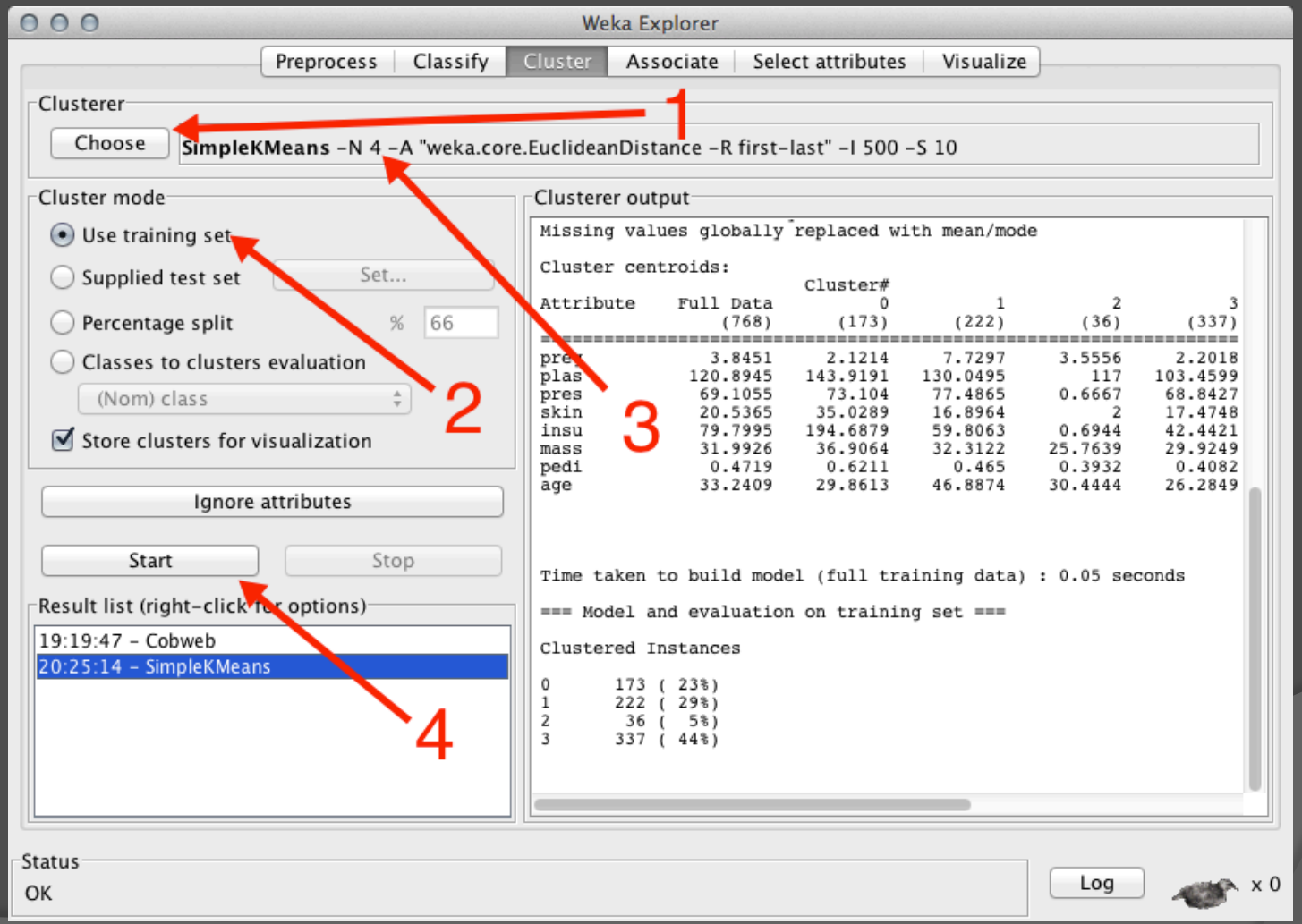
k-Means

- *"k"* stands for number of clusters, it is typically a user input to the algorithm; some criteria can be used to automatically estimate *k*
- Works only for numerical data

Example (1)

1. Open weather.arff
2. Go to *Clustering* tab
3. Choose a clustering method: *SimpleKMeans* 1
4. Set *Use training set* 2
5. Set *numCluster* (k) to 4 *3*
6. Click on *Start* button 4

Example (2)


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -N 4 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
19:19:47 - Cobweb
20:25:14 - SimpleKMeans
Clusterer output
Missing values globally replaced with mean/mode
Cluster centroids:
Cluster#
Attribute Full Data 0 1 2 3
(768) (173) (222) (36) (337)
preg 3.8451 2.1214 7.7297 3.5556 2.2018
plas 120.8945 143.9191 130.0495 117 103.4599
pres 69.1055 73.104 77.4865 0.6667 68.8427
skin 20.5365 35.0289 16.8964 2 17.4748
insu 79.7995 194.6879 59.8063 0.6944 42.4421
mass 31.9926 36.9064 32.3122 25.7639 29.9249
pedi 0.4719 0.6211 0.465 0.3932 0.4082
age 33.2409 29.8613 46.8874 30.4444 26.2849
Time taken to build model (full training data) : 0.05 seconds
=== Model and evaluation on training set ===
Clustered Instances
0 173 (23%)
1 222 (29%)
2 36 (5%)
3 337 (44%)
Status
OK
Log
x 0
1
2
3
4

Example (3)

```
Number of iterations: 31
Within cluster sum of squared errors: 95.23652346839076
Missing values globally replaced with mean/mode

Cluster centroids:
                         Cluster#
Attribute    Full Data          0          1          2          3
                 (768)      (173)      (222)       (36)      (337)
=============================================================================
preg              3.8451     2.1214     7.7297     3.5556     2.2018
plas            120.8945   143.9191   130.0495        117   103.4599
pres             69.1055     73.104    77.4865     0.6667    68.8427
skin             20.5365    35.0289    16.8964          2    17.4748
insu             79.7995   194.6879    59.8063     0.6944    42.4421
mass             31.9926    36.9064    32.3122    25.7639    29.9249
pedi              0.4719     0.6211      0.465     0.3932     0.4082
age              33.2409    29.8613    46.8874    30.4444    26.2849

Time taken to build model (full training data) : 0.05 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       173 ( 23%)
1       222 ( 29%)
2        36 (  5%)
3       337 ( 44%)
```

Centroids

Meaning of results

- The first column gives you the overall population centroid. The second to fifth columns give you the centroids for cluster 0 to 4, respectively. Each row gives the centroid coordinate for the specific dimension.

Exercise 1

- Go to the WEKA explorer environment and load the training file *iris.arff*
- Cluster the iris dataset using the *k-Means* clustering algorithm with k=5. Watch the result given by WEKA (Cobweb).

Exercise 2

- Cluster the *"iris.arff"* dataset using the k-Means Clustering algorithm with k=3, k=4 and k=5, with the same ten different value of the seed parameter. Use the option: Classes to cluster evaluation to evaluate the accuracy and store the results on an excel file. Compute the mean of the three different k values for the k-Means.

Association Rules

Association rules mining

- Method for discovering interesting relations between variables in large databases
- For example, the rule {onion, potatoes} → {burger} would indicate that if a customer buys onions and potatoes together, he is likely to also buy hamburger meat

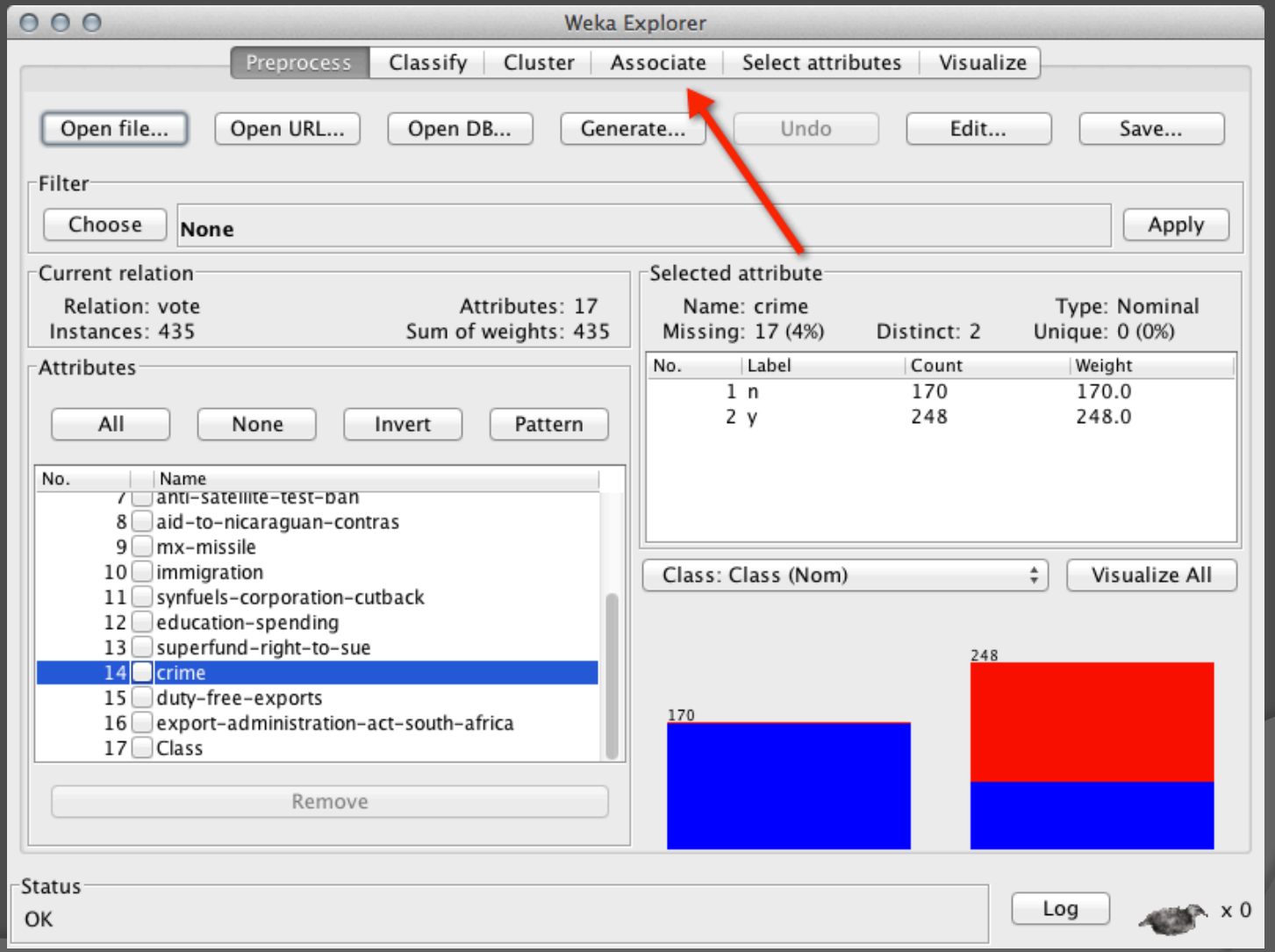
Classification vs Association Rules

- Classification
  - Focus on one target field
  - Specify class in all cases
  - Measures: Accuracy
- Association Rules
  - Many target fields
  - Applicable in some cases
  - Measures: Support, Confidence, Lift

Association rules

- Association rule R : itemset1 => itemset2
  - Itemset1, itemset2 are disjoint and Itemset2 is non-empty
  - meaning: if transaction includes Itemset1 then it also has Itemset2
- Example
  - A, B => E, C

Association rules with Weka (1)

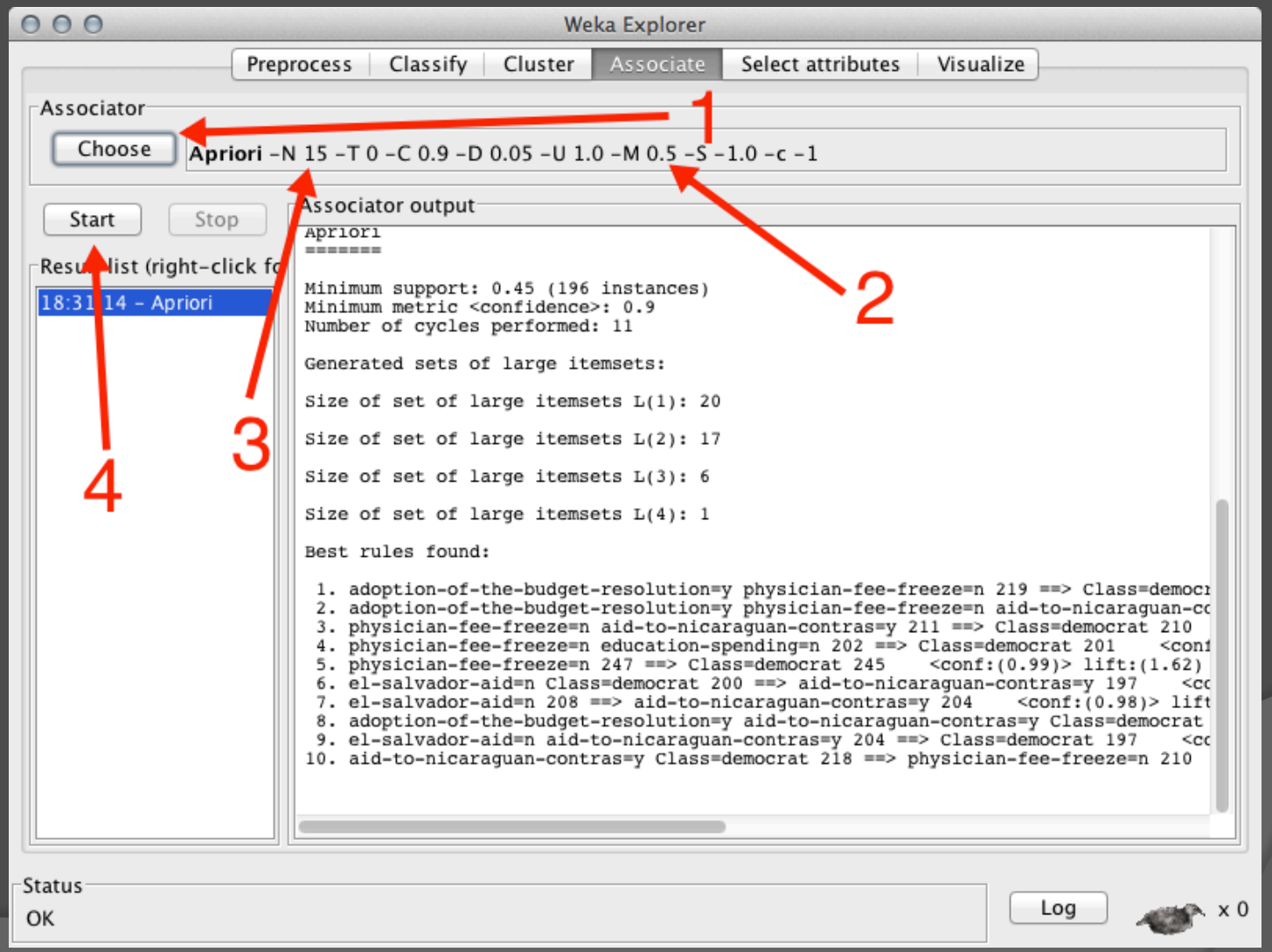


Association rules with Weka (2)

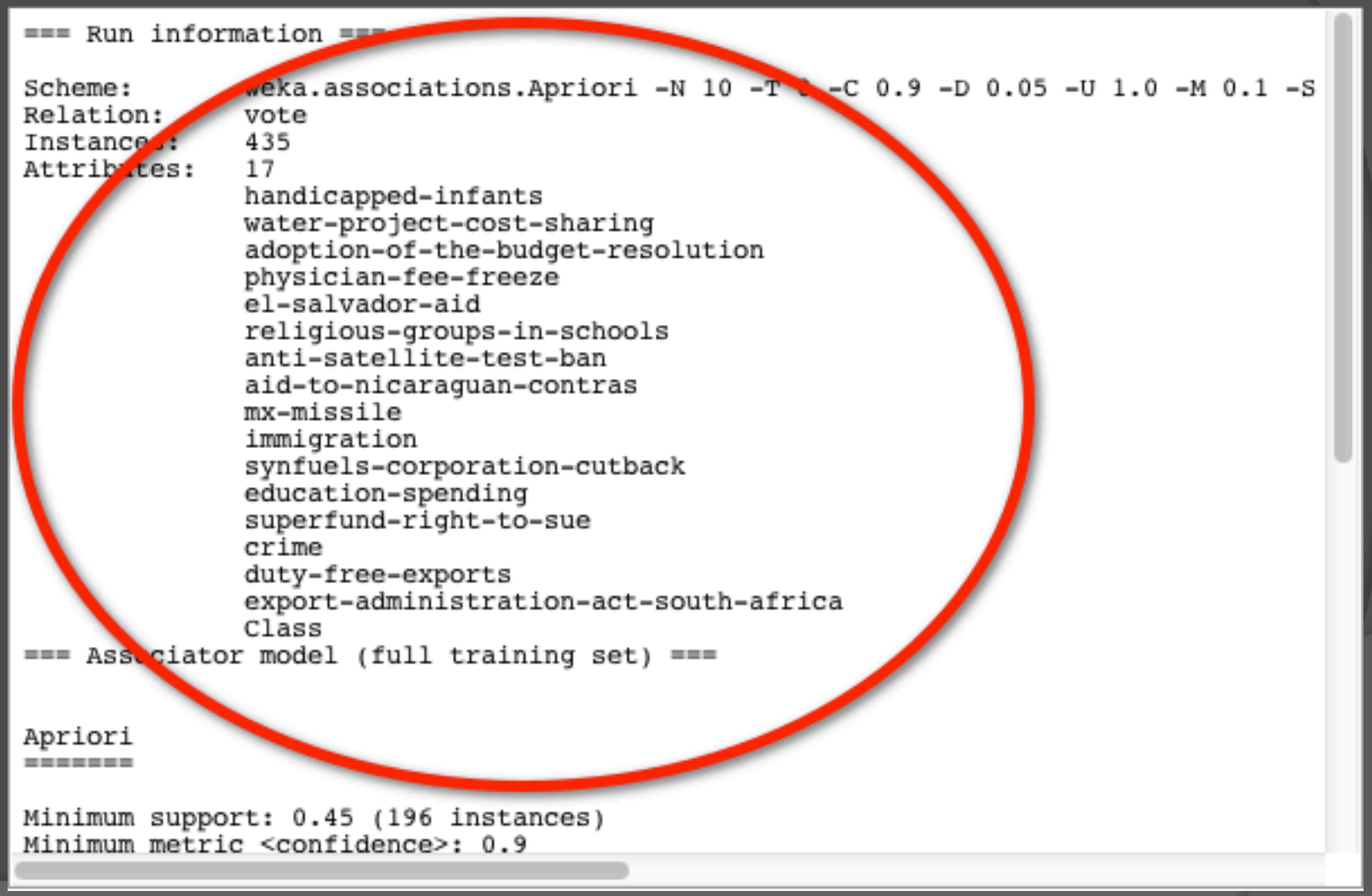
Example (1)

1. Open vote.arff
2. Go to *Associate* tab
3. Choose a Association Rules method: *Apriori* 1
4. Set *lowerMinBoundSupport* to 0.5 2
5. Set *numRules* to 15 *3*
6. Click on *Start* button 4

Example (2)

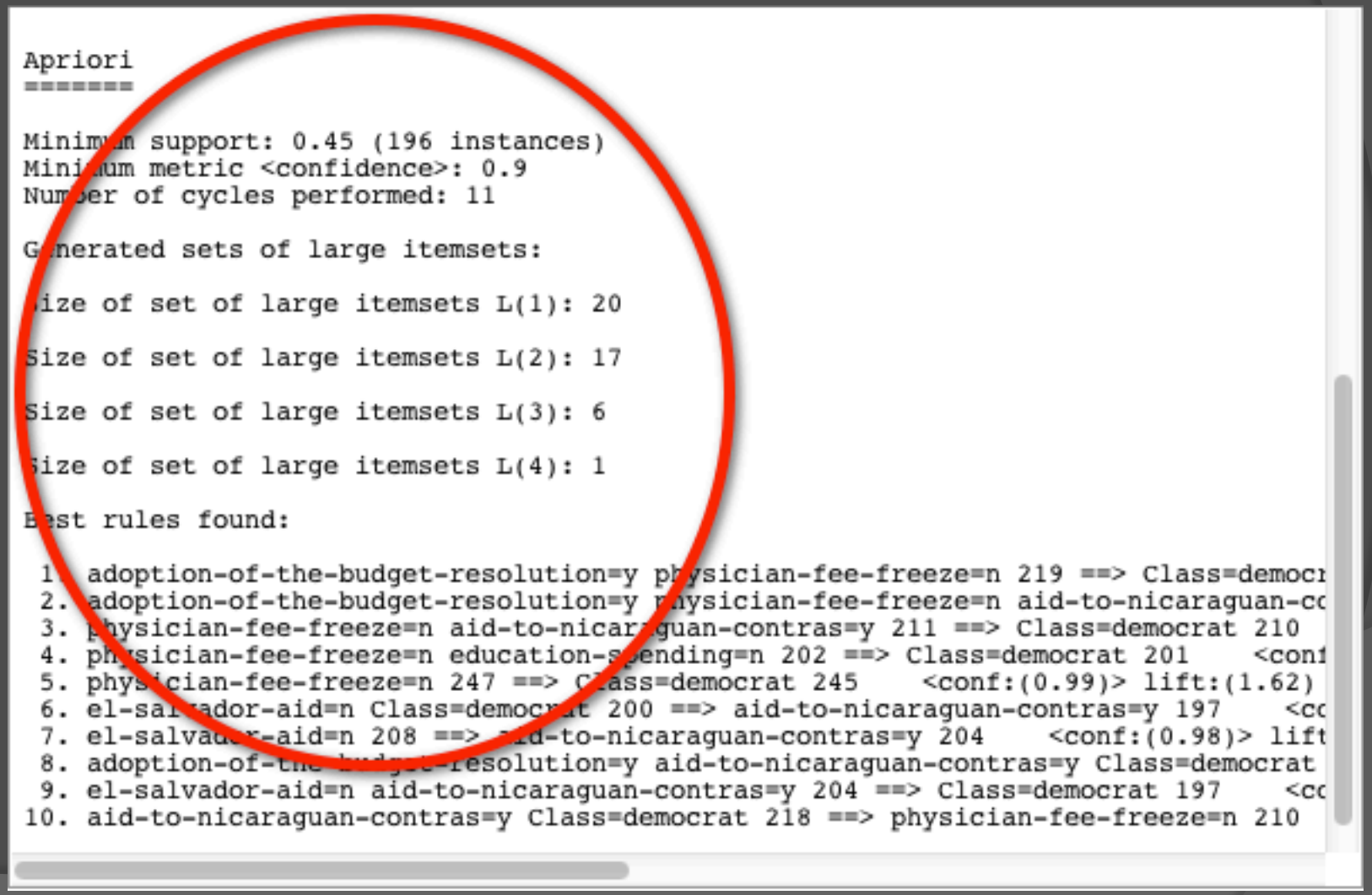

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Associator
Choose
Apriori -N 15 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.5 -S -1.0 -c -1
Start
Stop
Result list (right-click fo
18:31:14 - Apriori
Associator output
Apriori
=======
Minimum support: 0.45 (196 instances)
Minimum metric <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=democr
2. adoption-of-the-budget-resolution=y physician-fee-freeze=n aid-to-nicaraguan-cc
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)> lift:(1.62)
6. el-salvador-aid=n Class=democrat 200 ==> aid-to-nicaraguan-contras=y 197 <cc
7. el-salvador-aid=n 208 ==> aid-to-nicaraguan-contras=y 204 <conf:(0.98)> lift
8. adoption-of-the-budget-resolution=y aid-to-nicaraguan-contras=y Class=democrat
9. el-salvador-aid=n aid-to-nicaraguan-contras=y 204 ==> Class=democrat 197 <cc
10. aid-to-nicaraguan-contras=y Class=democrat 218 ==> physician-fee-freeze=n 210
Status
OK
Log
x 0
1
2
3
4

Meaning of results (1)



```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S
Relation:     vote
Instances:    435
Attributes:   17
              handicapped-infants
              water-project-cost-sharing
              adoption-of-the-budget-resolution
              physician-fee-freeze
              el-salvador-aid
              religious-groups-in-schools
              anti-satellite-test-ban
              aid-to-nicaraguan-contras
              mx-missile
              immigration
              synfuels-corporation-cutback
              education-spending
              superfund-right-to-sue
              crime
              duty-free-exports
              export-administration-act-south-africa
              Class
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.45 (196 instances)
Minimum metric <confidence>: 0.9
```

Meaning of results (2)



```
Apriori
=======

Minimum support: 0.45 (196 instances)
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=democr
 2. adoption-of-the-budget-resolution=y physician-fee-freeze=n aid-to-nicaraguan-co
 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)> lift:(1.62)
 6. el-salvador-aid=n Class=democrat 200 ==> aid-to-nicaraguan-contras=y 197    <co
 7. el-salvador-aid=n 208 ==> aid-to-nicaraguan-contras=y 204    <conf:(0.98)> lift
 8. adoption-of-the-budget-resolution=y aid-to-nicaraguan-contras=y Class=democrat
 9. el-salvador-aid=n aid-to-nicaraguan-contras=y 204 ==> Class=democrat 197    <co
10. aid-to-nicaraguan-contras=y Class=democrat 218 ==> physician-fee-freeze=n 210
```

Meaning of results (3)

```
Apriori
=======

Minimum support: 0.45 (196 instances)
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=democ
 2. adoption-of-the-budget-resolution=y physician-fee-freeze=n aid-to-nicaraguan-c
 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    <con
 5. physician-fee-freeze=n 247 ==> Class=democrat 245    <conf:(0.99)> lift:(1.62)
 6. el-salvador-aid=n Class=democrat 200 ==> aid-to-nicaraguan-contras=y 197    <c
 7. el-salvador-aid=n 208 ==> aid-to-nicaraguan-contras=y 204    <conf:(0.98)> lif
 8. adoption-of-the-budget-resolution=y aid-to-nicaraguan-contras=y Class=democrat
 9. el-salvador-aid=n aid-to-nicaraguan-contras=y 204 ==> Class=democrat 197    <c
10. aid-to-nicaraguan-contras=y Class=democrat 218 ==> physician-fee-freeze=n 210
```

Exercise (1)

- Mining the file ’supermarket.arff’
- Open with a text editor this file and look at the value inside the file.
- Which is the particularity of this file?
- Try to understand why this file is particularly adapted for the Association Rules task.

Exercise (2)

- Create an “arff”-file containing the following document-word representation (binary mode).

  t1 = {machine, learning, classifier}

  t2 = {data, mining, associative, classifier}

  t3 = {mining, decision, tree}

  t4 = {association, mining, data}

  t5 = {decision, tree, classifier}

Exercise (3)

- Extract the top 10 Association Rules from your ‘arff’-file (Exercise 2)
- Discuss the results

Questions concerning to final project…