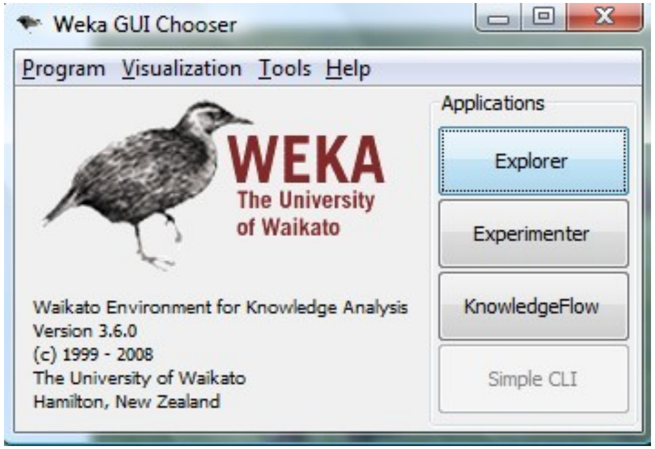
Introduction to Weka

WEKA
The University
of Waikato

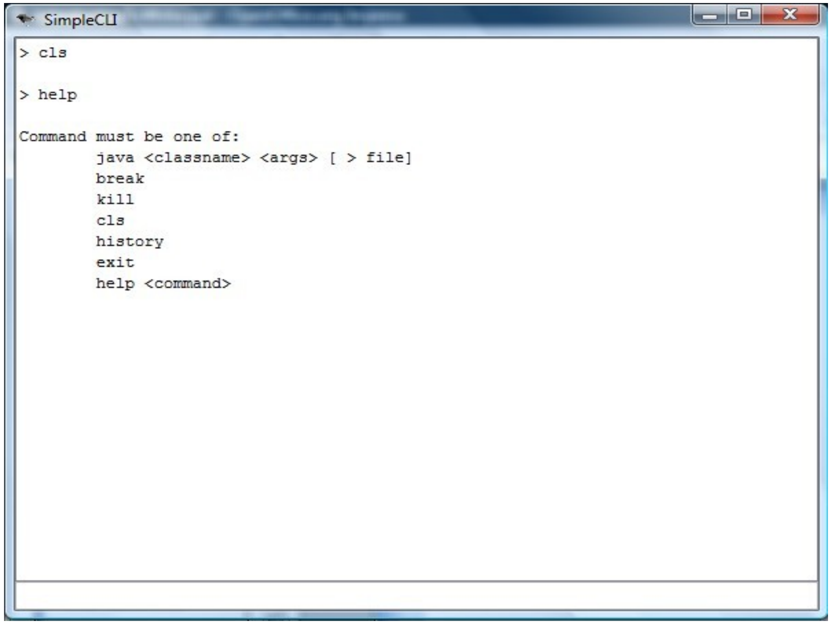


Waikato Environment for Knowledge Analysis
Version 3.6.0
(c) 1999 - 2008
The University of Waikato
Hamilton, New Zealand

What is Weka?

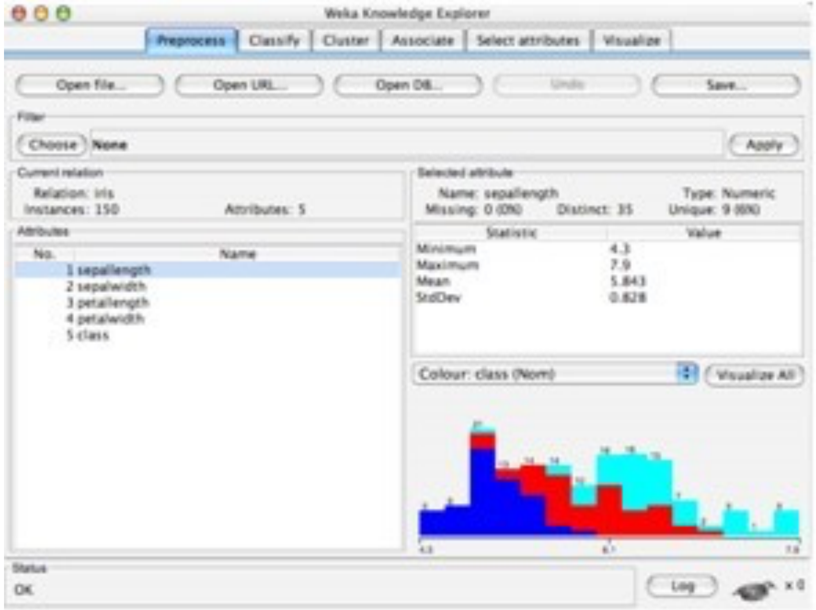
- Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code.
- Weka contains tools for data pre-processing, classification, regression, clustering,
- association rules, and visualization. It is also well-suited for developing new machine learning schemes.

CLI Vs GUI



```
> cls

> help

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

- Recommended for in-depth usage
- Offers some functionality not available via the GUI



Filter: None

Relation: iris
Instances: 150   Attributes: 5

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

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)

Status: OK

- Explorer
- Experimenter
- Knowledge Flow

Datasets in Weka

- Each entry in a dataset is an instance of the java class:
  - weka.core.Instance
- Each instance consists of a number of attributes

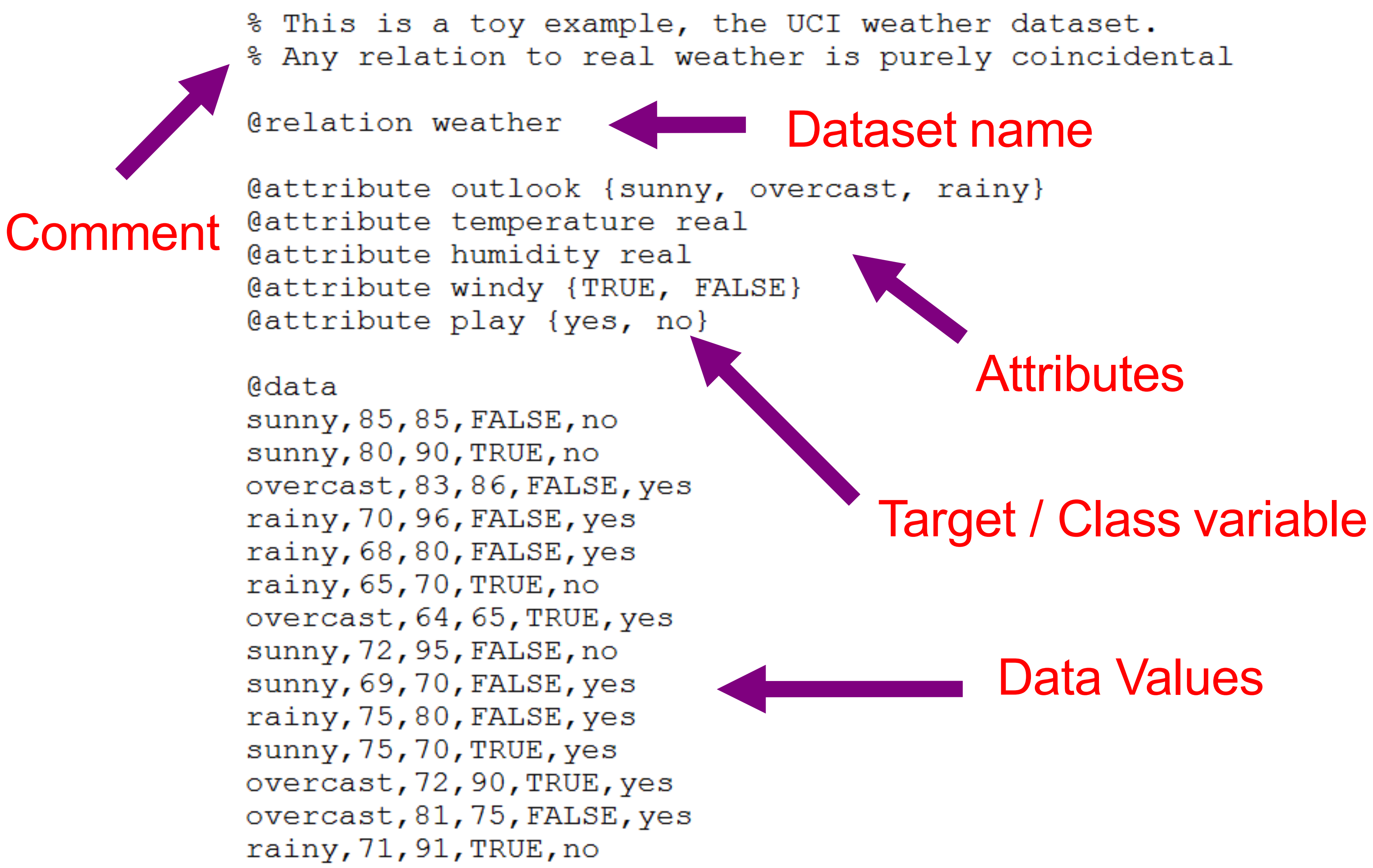
Attributes

- *Nominal*: one of a predefined list of values
  - e.g. red, green, blue
- *Numeric*: A real or integer number
- *String:* Enclosed in “double quotes”
- *Date*
- *Relational*

ARFF Files

- The external representation of an Instances class
- Consists of:
  - A header: Describes the attribute types
  - Data section: Comma separated list of data

ARFF File Example



```
% This is a toy example, the UCI weather dataset.
% Any relation to real weather is purely coincidental

@relation weather

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

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

Comment

Dataset name

Attributes

Target / Class variable

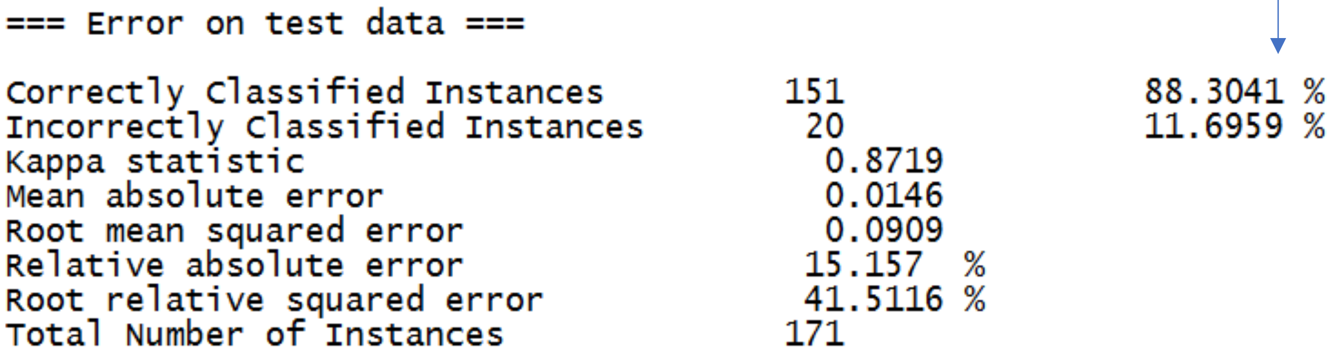
Data Values

Classifiers in Weka

- Learning algorithms in Weka are derived from the abstract class:
  - weka.classifiers.Classifier
- Simple classifier: ZeroR
  - Just determines the most common class
  - Or the median (in the case of numeric values)
  - Tests how well the class can be predicted without considering other attributes
  - Can be used as a Lower Bound on Performance.

Soybean Results

accuracy

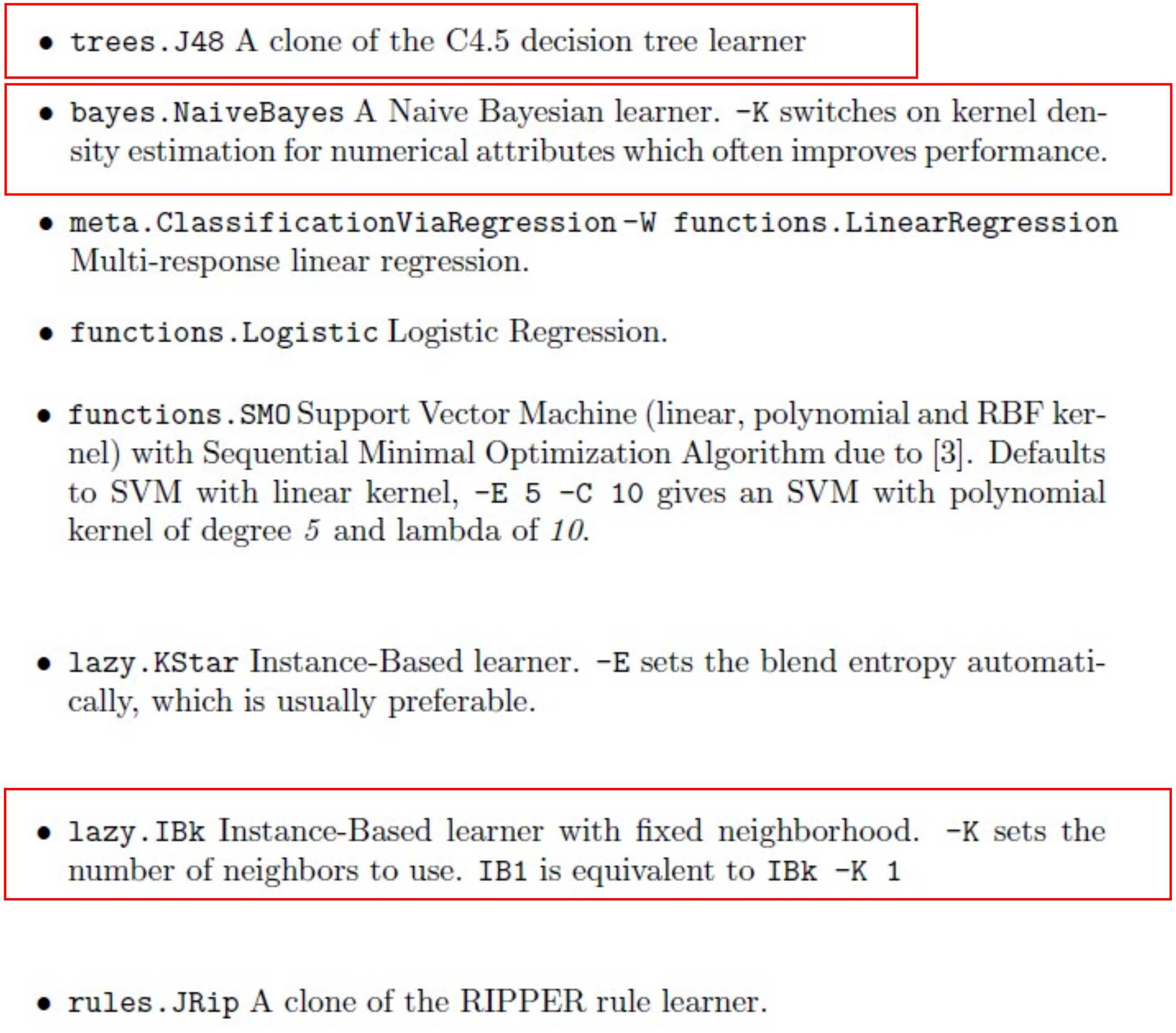


```
=== Error on test data ===

Correctly Classified Instances         151               88.3041 %
Incorrectly Classified Instances        20               11.6959 %
Kappa statistic                          0.8719
Mean absolute error                      0.0146
Root mean squared error                  0.0909
Relative absolute error                 15.157  %
Root relative squared error             41.5116 %
Total Number of Instances              171
```

Filters

- weka.filters package
- Transform datasets
- Support for data preprocessing
  - e.g. Removing/Adding Attributes
  - e.g. Discretize numeric attributes into nominal ones
- More info in Weka Manual p. 15 & 16.

More Classifiers



- `trees.J48` A clone of the C4.5 decision tree learner

- `bayes.NaiveBayes` A Naive Bayesian learner. `-K` switches on kernel density estimation for numerical attributes which often improves performance.

- `meta.ClassificationViaRegression -W functions.LinearRegression` Multi-response linear regression.

- `functions.Logistic` Logistic Regression.

- `functions.SMO` Support Vector Machine (linear, polynomial and RBF kernel) with Sequential Minimal Optimization Algorithm due to [3]. Defaults to SVM with linear kernel, `-E 5 -C 10` gives an SVM with polynomial kernel of degree *5* and lambda of *10*.

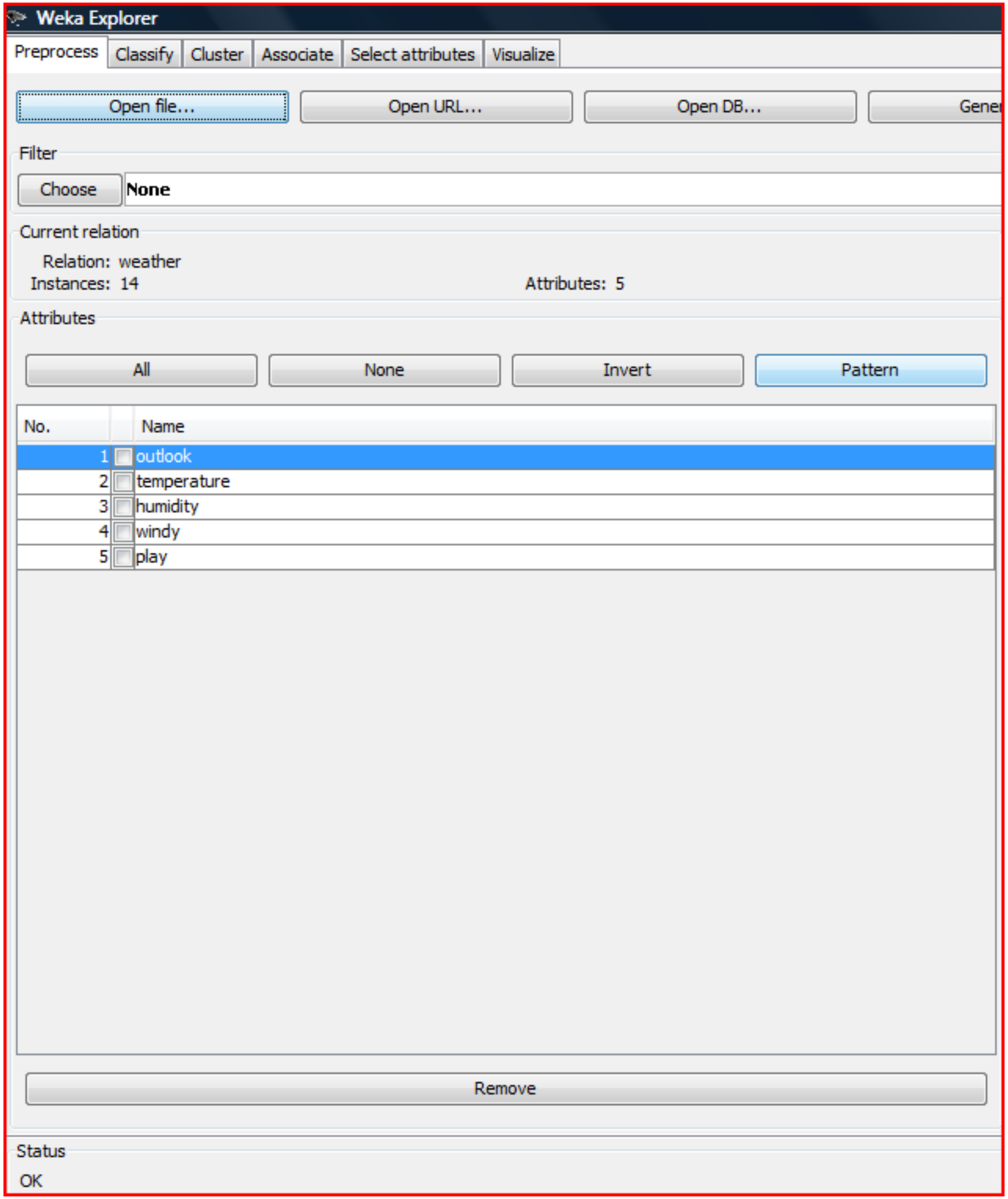
- `lazy.KStar` Instance-Based learner. `-E` sets the blend entropy automatically, which is usually preferable.

- `lazy.IBk` Instance-Based learner with fixed neighborhood. `-K` sets the number of neighbors to use. `IB1` is equivalent to `IBk -K 1`

- `rules.JRip` A clone of the RIPPER rule learner.

Preprocess

- Load Data
- Preprocess Data
- Analyse Attributes



Filter

None

Current relation

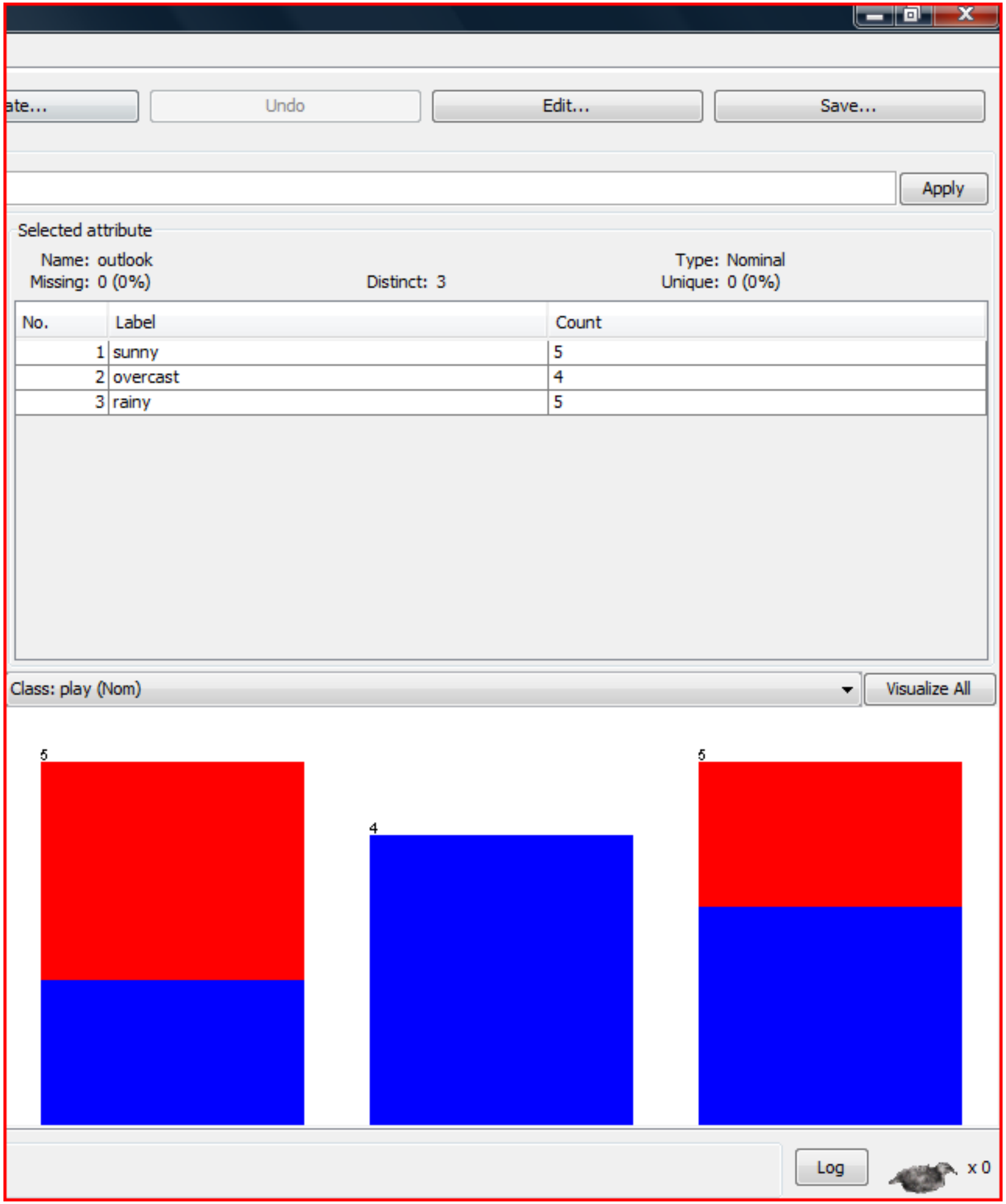
Relation: weather
Instances: 14
Attributes: 5

Attributes

No.		Name
1	□	outlook
2	□	temperature
3	□	humidity
4	□	windy
5	□	play

Status

OK



Selected attribute

Name: outlook
Missing: 0 (0%)
Distinct: 3
Type: Nominal
Unique: 0 (0%)

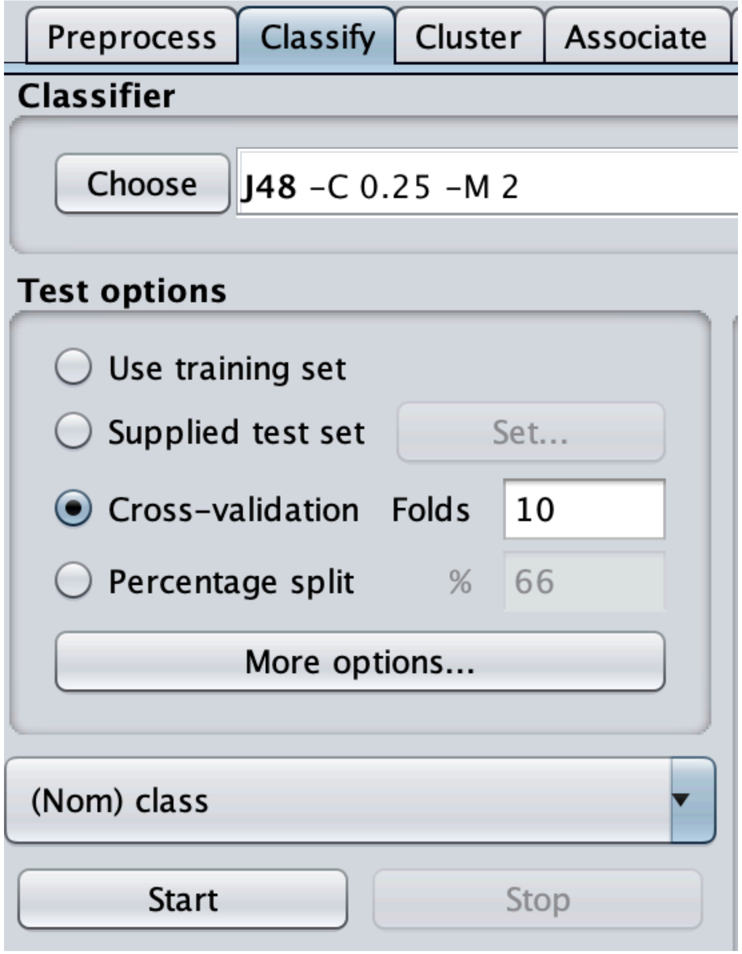
No.	Label	Count
1	sunny	5
2	overcast	4
3	rainy	5

Class: play (Nom)

Classify

- Select Test Options e.g:
  - Cross Validation...
  -
  -
- Run classifiers
- View results

Classify



Classifier

J48 -C 0.25 -M 2

Test options

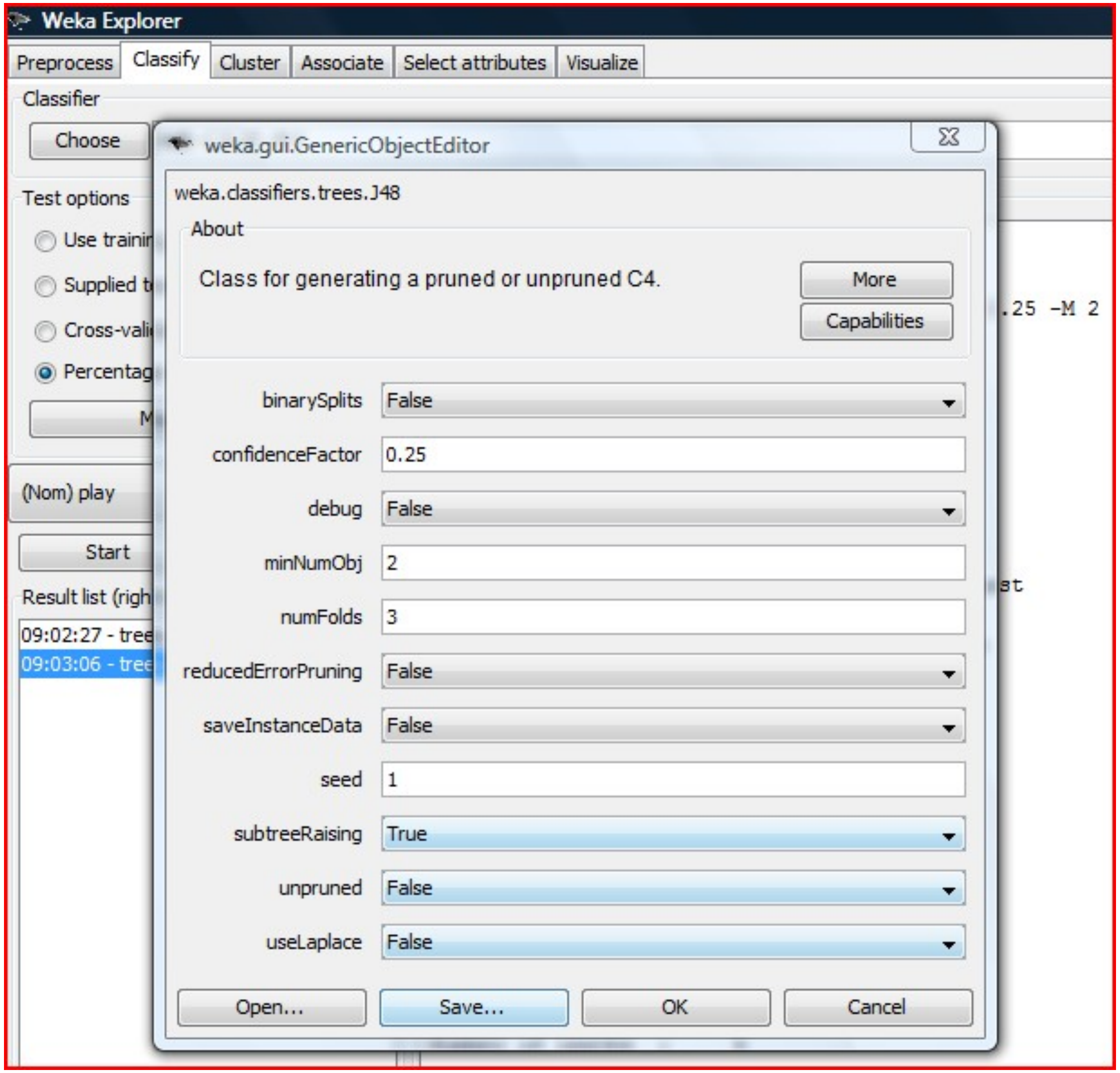
○ Use training set

○ Supplied test set

● Cross-validation Folds 10

○ Percentage split % 66

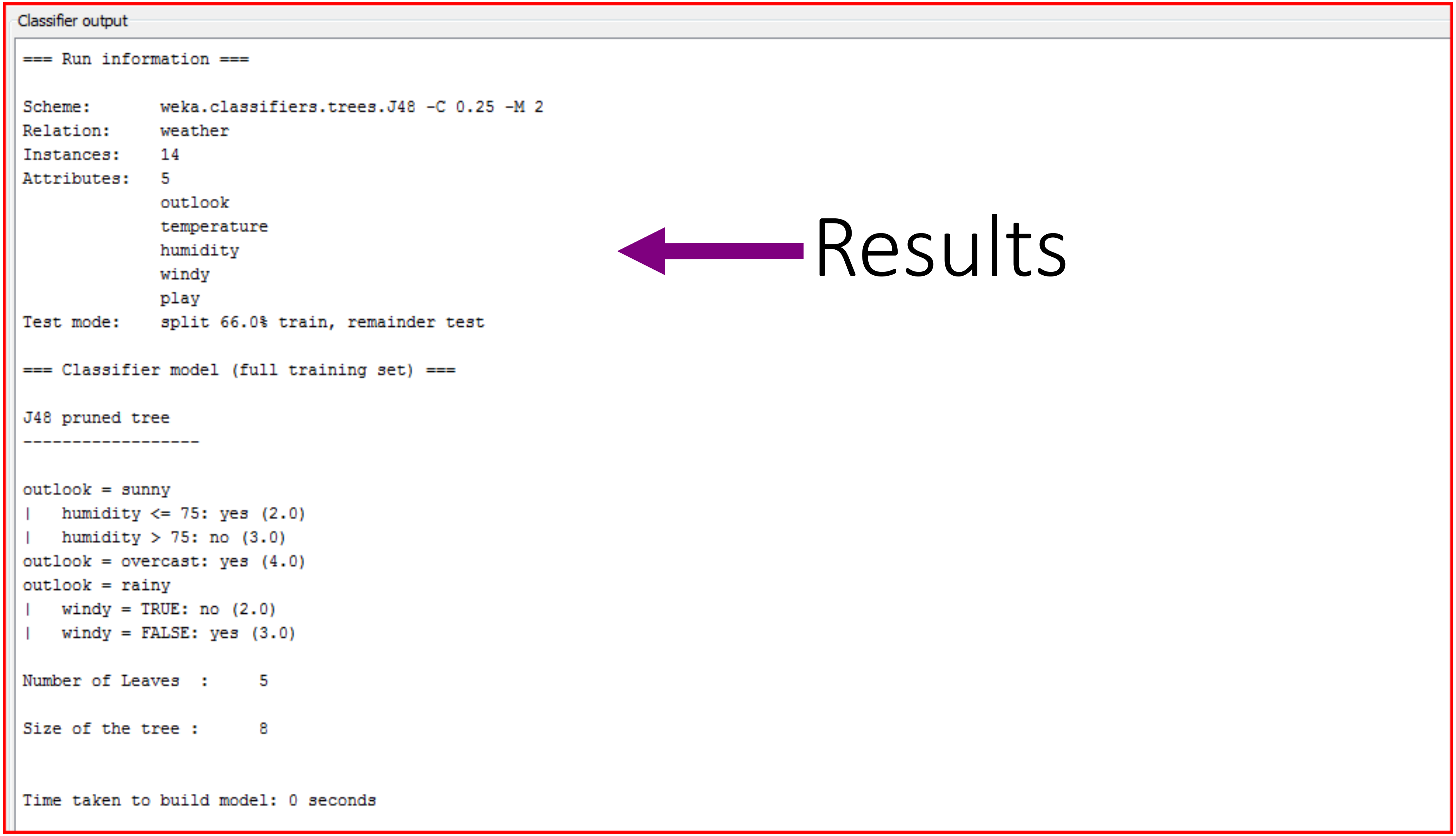
(Nom) class



weka.classifiers.trees.J48

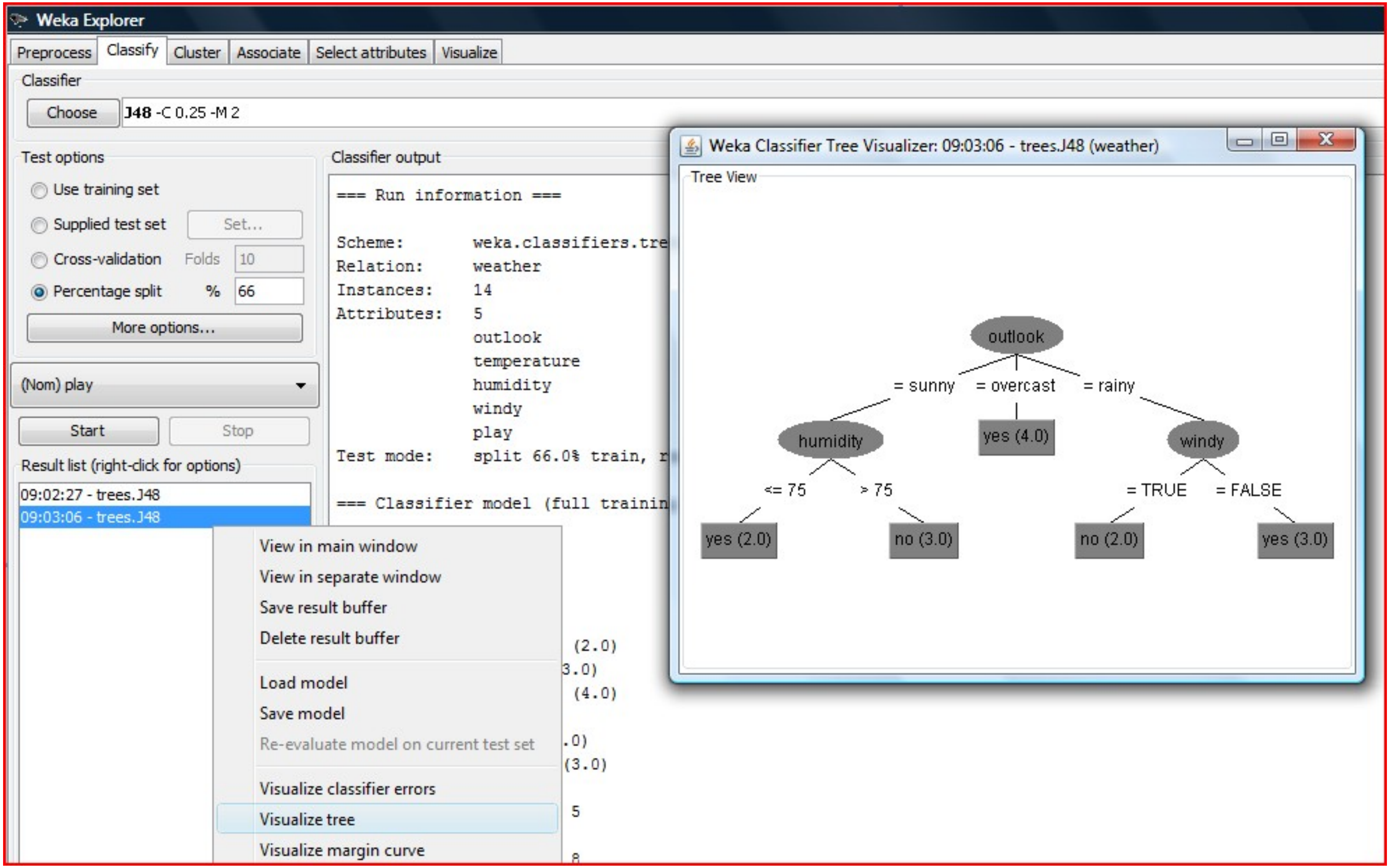
Class for generating a pruned or unpruned C4.

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



```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     weather
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    split 66.0% train, remainder test

=== Classifier model (full training set) ===

J48 pruned tree
------------------

outlook = sunny
|   humidity <= 75: yes (2.0)
|   humidity > 75: no (3.0)
outlook = overcast: yes (4.0)
outlook = rainy
|   windy = TRUE: no (2.0)
|   windy = FALSE: yes (3.0)

Number of Leaves  :     5

Size of the tree :      8


Time taken to build model: 0 seconds
```

Results



J48 -C 0.25 -M 2

○ Use training set
○ Supplied test set
○ Cross-validation Folds 10
● Percentage split % 66

(Nom) play

09:02:27 - trees.J48
09:03:06 - trees.J48

```
=== Run information ===

Scheme:       weka.classifiers.tre
Relation:     weather
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    split 66.0% train, r

=== Classifier model (full trainin
```

```
 (2.0)
3.0)
 (4.0)

.0)
(3.0)

 5

 8
```

Weka Classifier Tree Visualizer: 09:03:06 - trees.J48 (weather)