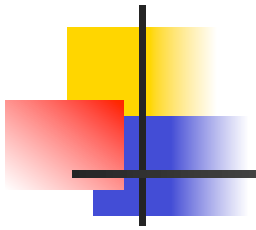
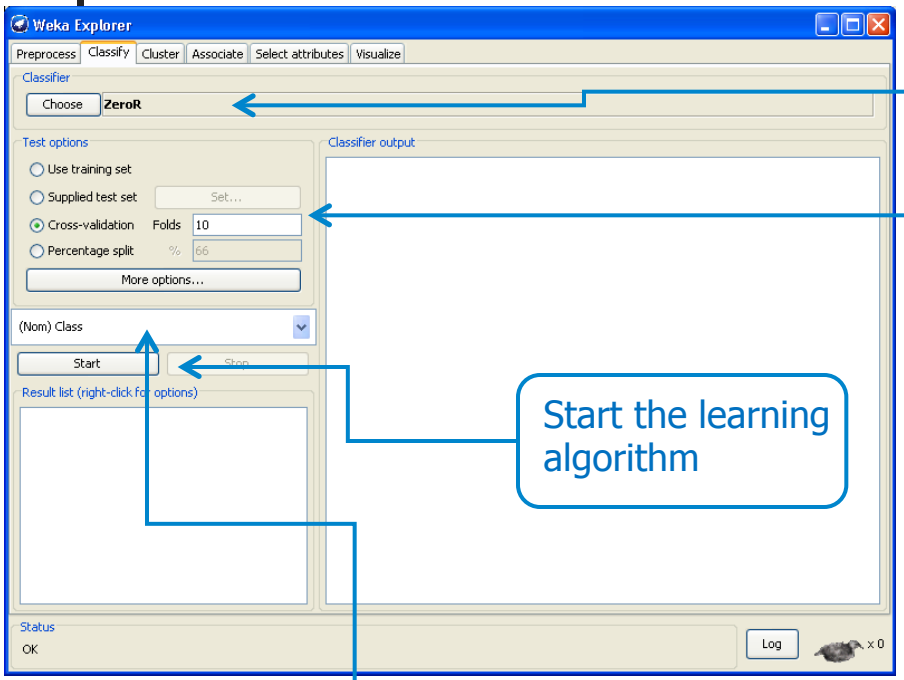


WEKA
Waikato Environment for Knowledge Analysis

Classification

Prof. Pietro Ducange

Classification



Name of the currently selected classifier and its options

Start the learning algorithm

Attribute to be predicted (By default, the class is taken to be the last attribute in the data).

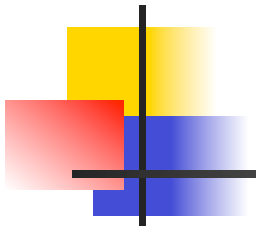
Options for testing the results of the chosen classifier

→**Use 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

→**Percentage split:** The classifier is evaluated on how well it predicts a certain percentage of the data which is held out for testing

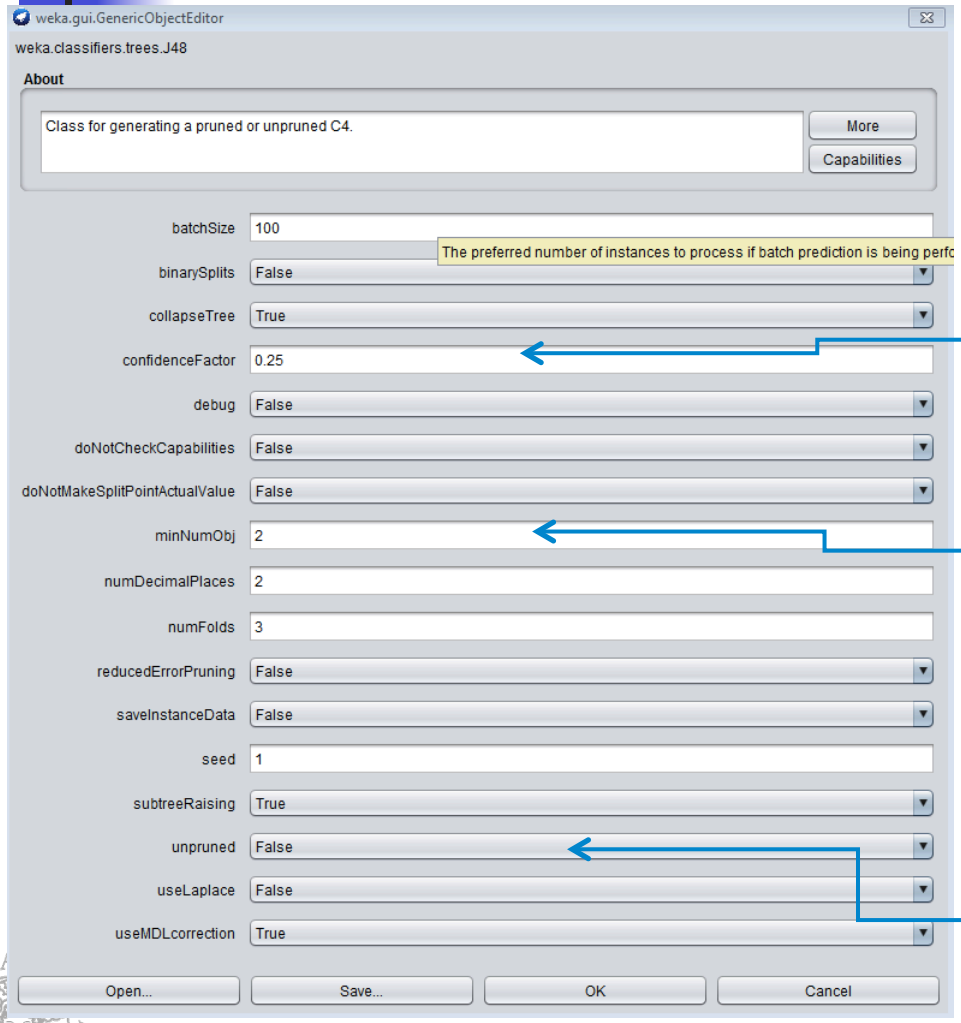


Classification: C4.5 Example (1)

Using C4.5 for generating the decision tree

- Choose a classification dataset (IRIS)
- Select the J.48
- Set the Percentage Split to 66%
- Set the Parameters
- Visualize the decision tree
- Analyze the results

Classification: C4.5 Example (2)

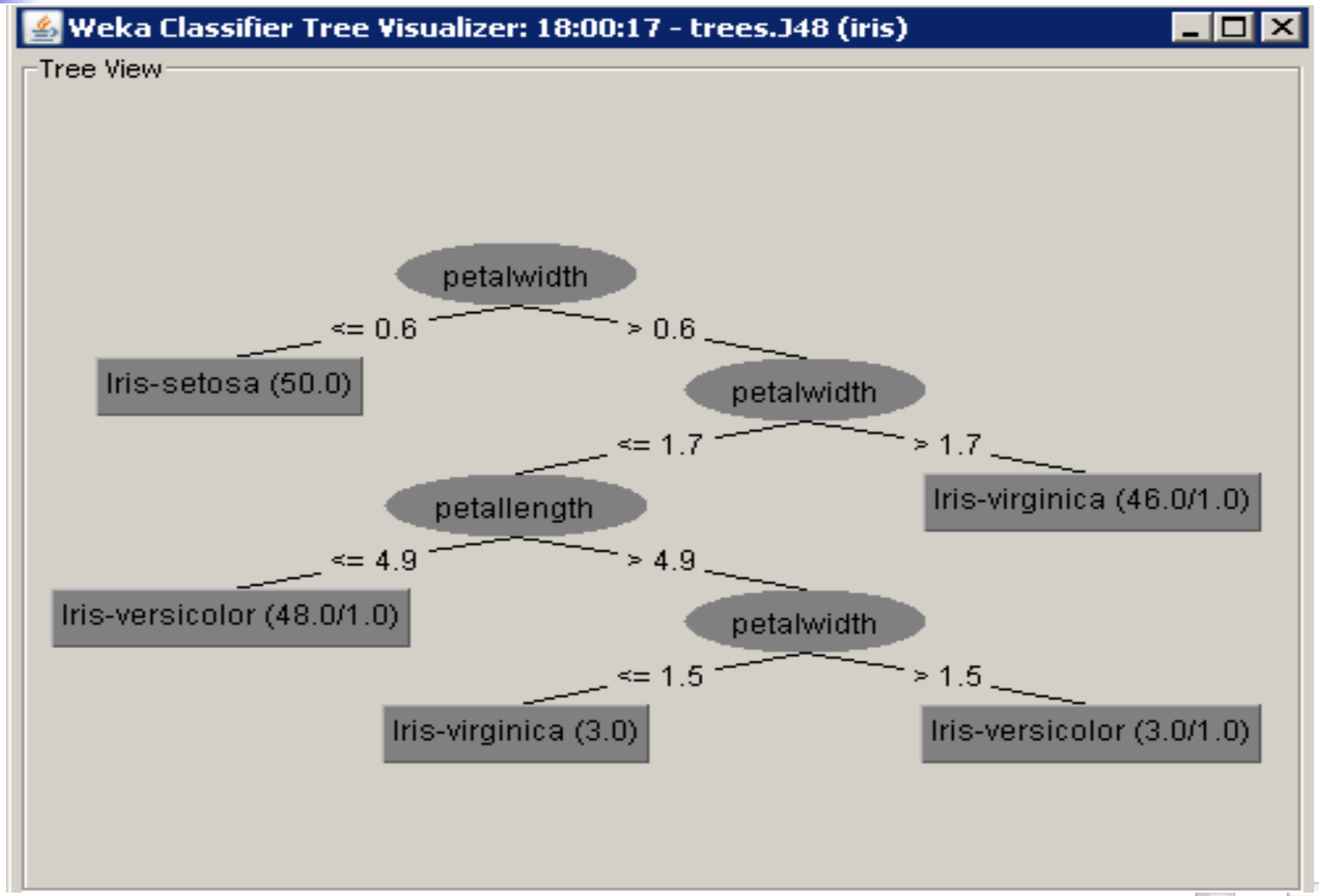


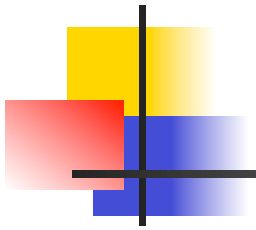
The confidence factor used for pruning (smaller values incur more pruning)

The minimum number of instances per leaf

Whether pruning is performed

Classification: C4.5 Example (3)

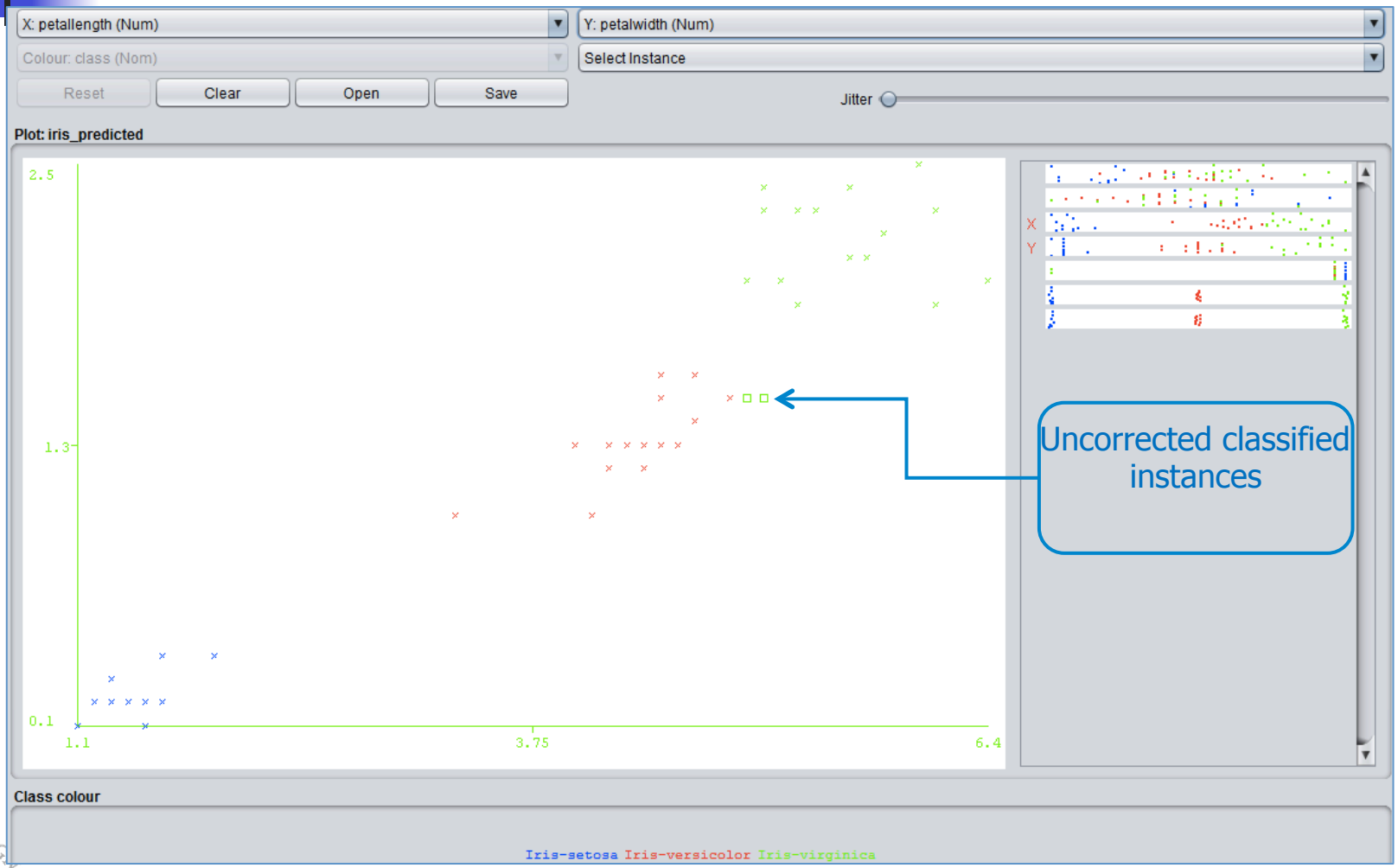


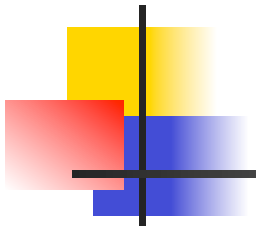


Classification: C4.5 Example (4)

```
=== 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   ROC Area  Class
                 1         0          1         1         1          1        Iris-setosa
                 1         0.063      0.905     1         0.95       0.969    Iris-versicolor
                 0.882     0          1         0.882     0.938      0.967    Iris-virginica
Weighted Avg.    0.961     0.023      0.965     0.961     0.961      0.977

=== 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
```

Classification: C4.5 Example (5)





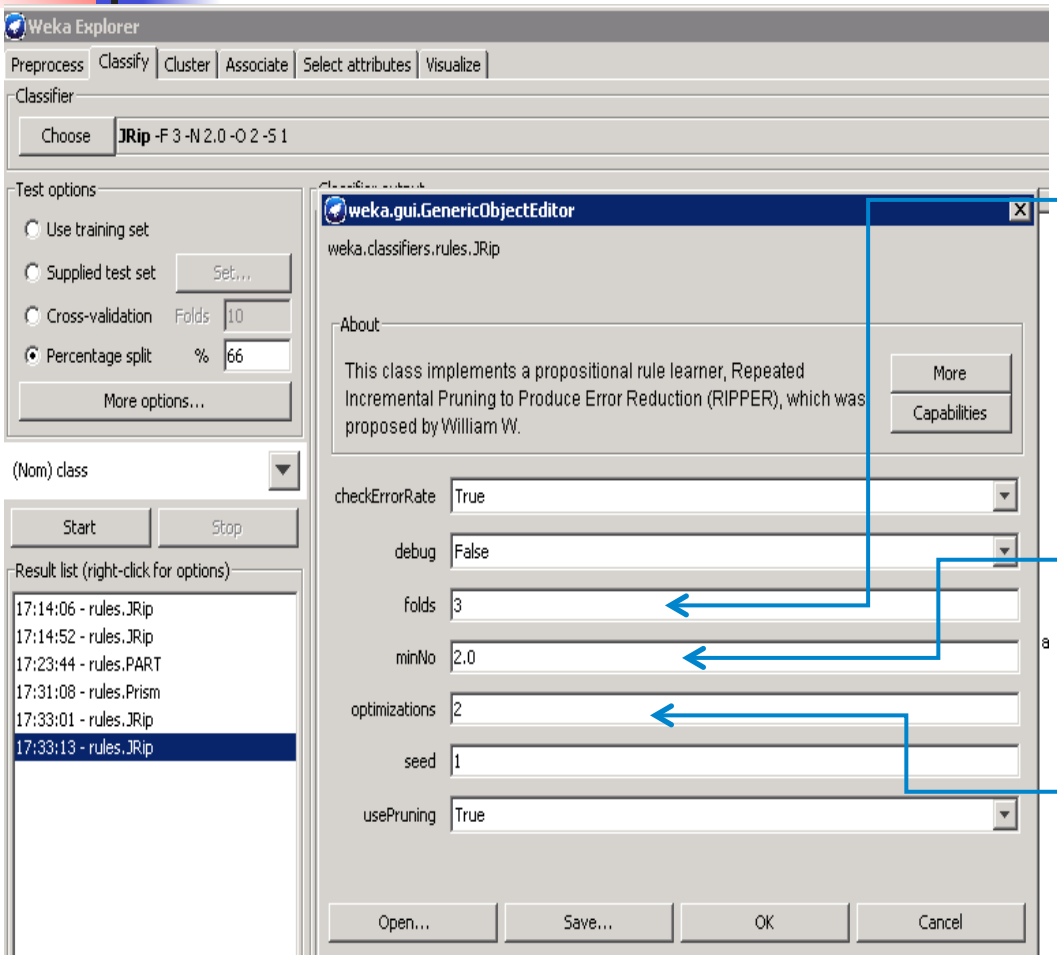
Exercise I

Perform the classification by using the following classifiers (default parameters) and the iris dataset (66% spit):

- Jrip (rules)
- KNN(lazy)
- Naive Bayes (Bayes)

Which is the most accurate classifier on the test set?

Classification: Jrip Example (1)



Determines the amount of data used for pruning
One fold is used for pruning,
the rest for growing the rules

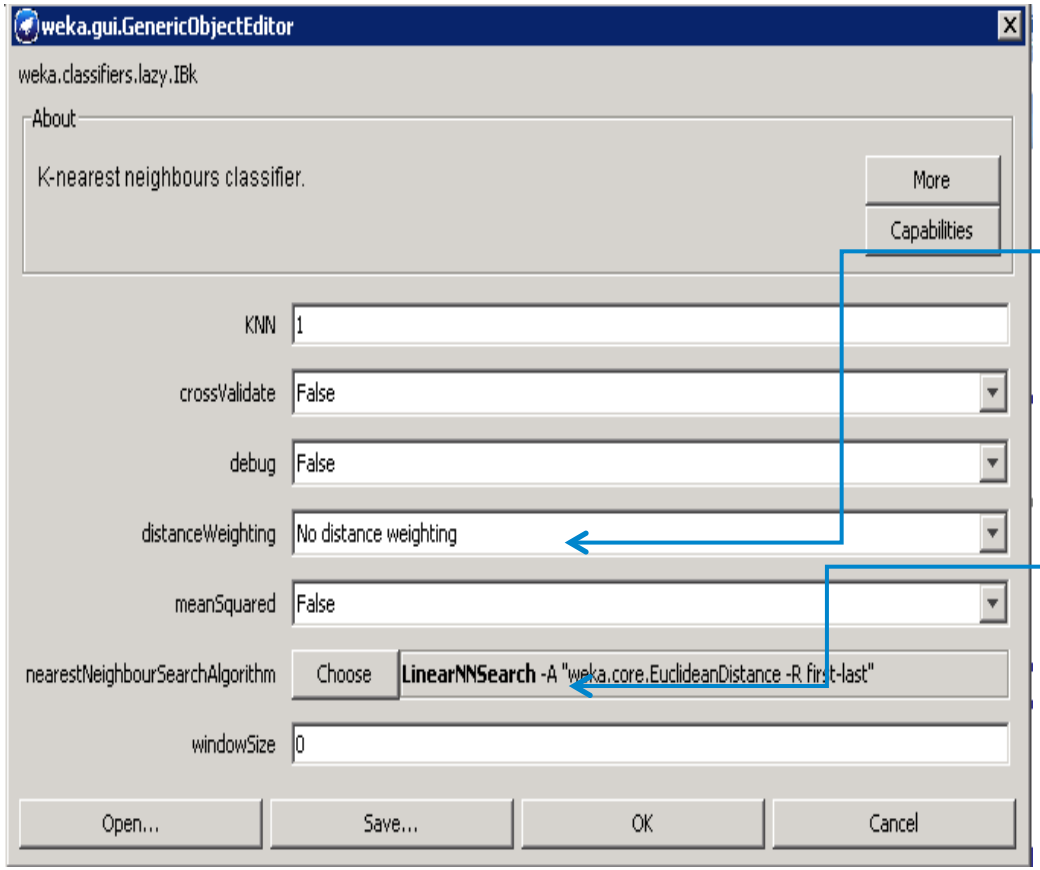
The minimum total weight
of the instances in a rule

The number of optimization runs

Classification: Jrip Example (2)

```
JRIP rules:
===========

(petallength >= 3.3) and (petalwidth <= 1.6) and (petallength <= 4.9) => class=Iris-versicolor (46.0/0.0)
(petallength <= 1.9) => class=Iris-setosa (50.0/0.0)
 => class=Iris-virginica (54.0/4.0)

Number of Rules : 3


Time taken to build model: 0.01 seconds

=== Evaluation on test split ===

Time taken to test model on training split: 0 seconds

=== Summary ===

Correctly Classified Instances          47               92.1569 %
Incorrectly Classified Instances         4                7.8431 %
Kappa statistic                          0.8821
Mean absolute error                      0.0808
Root mean squared error                  0.2233
Relative absolute error                 18.1437 %
Root relative squared error             47.2352 %
Coverage of cases (0.95 level)          92.1569 %
Mean rel. region size (0.95 level)      54.2484 %
Total Number of Instances               51

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,056    0,882      1,000    0,938      0,913    0,972     0,882     Iris-setosa
                 0,895    0,063    0,895      0,895    0,895      0,832    0,891     0,840     Iris-versicolor
                 0,882    0,000    1,000      0,882    0,938      0,913    0,971     0,941     Iris-virginica
Weighted Avg.    0,922    0,040    0,926      0,922    0,922      0,883    0,942     0,886

=== Confusion Matrix ===

  a  b  c   <-- classified as
 15  0  0 |  a = Iris-setosa
  2 17  0 |  b = Iris-versicolor
  0  2 15 |  c = Iris-virginica
```

Classification: KNN Example (1)



Gets the distance weighting method used: Weight by 1/distance or by 1-distance

The nearest neighbor search algorithm to use

Classification: KNN Example (2)

Time taken to build model: 0 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	0.0382	
Root mean squared error	0.1599	
Relative absolute error	8.5739 %	
Root relative squared error	33.8182 %	
Total Number of Instances	51	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1	0	1	1	1	1	Iris-setosa
	1	0.063	0.905	1	0.95	0.969	Iris-versicolor
	0.882	0	1	0.882	0.938	0.943	Iris-virginica
Weighted Avg.	0.961	0.023	0.965	0.961	0.961	0.969	

=== 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

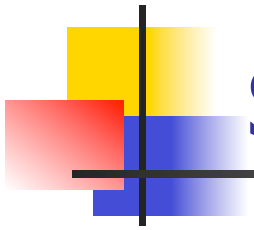
Classification: NaiveBayes Example (1)

Naive Bayes Classifier

Attribute	Class Iris-setosa (0.33)	Iris-versicolor (0.33)	Iris-virginica (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
petallength			
mean	1.4694	4.2452	5.5516
std. dev.	0.1782	0.4712	0.5529
weight sum	50	50	50
precision	0.1405	0.1405	0.1405
petalwidth			
mean	0.2743	1.3097	2.0343
std. dev.	0.1096	0.1915	0.2646
weight sum	50	50	50
precision	0.1143	0.1143	0.1143

Classification: NaiveBayes Example (2)

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

Correctly Classified Instances          48               94.1176 %
Incorrectly Classified Instances         3                5.8824 %
Kappa statistic                          0.9113
Mean absolute error                      0.0447
Root mean squared error                  0.1722
Relative absolute error                 10.0365 %
Root relative squared error             36.4196 %
Total Number of Instances               51

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        Iris-setosa
                 0.947     0.063      0.9       0.947     0.923      0.988    Iris-versicolor
                 0.882     0.029      0.938     0.882     0.909      0.988    Iris-virginica
Weighted Avg.    0.941     0.033      0.942     0.941     0.941      0.992

=== Confusion Matrix ===

  a  b  c   <-- classified as
 15  0  0 |  a = Iris-setosa
  0 18  1 |  b = Iris-versicolor
  0  2 15 |  c = Iris-virginica
```



Simple Comparison Among Classifiers

C4.5

Correctly Classified Instances 49 96.0784 %

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
1	0	1	1	1	1	Iris-setosa
1	0.063	0.905	1	0.95	0.969	Iris-versicolor
0.882	0	1	0.882	0.938	0.967	Iris-virginica

Jrip

Correctly Classified Instances 47 92.1569 %

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
1	0.056	0.882	1	0.938	0.972	Iris-setosa
0.895	0.063	0.895	0.895	0.895	0.891	Iris-versicolor
0.882	0	1	0.882	0.938	0.971	Iris-virginica

KNN

Correctly Classified Instances 49 96.0784 %

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
1	0	1	1	1	1	Iris-setosa
1	0.063	0.905	1	0.95	0.969	Iris-versicolor
0.882	0	1	0.882	0.938	0.943	Iris-virginica

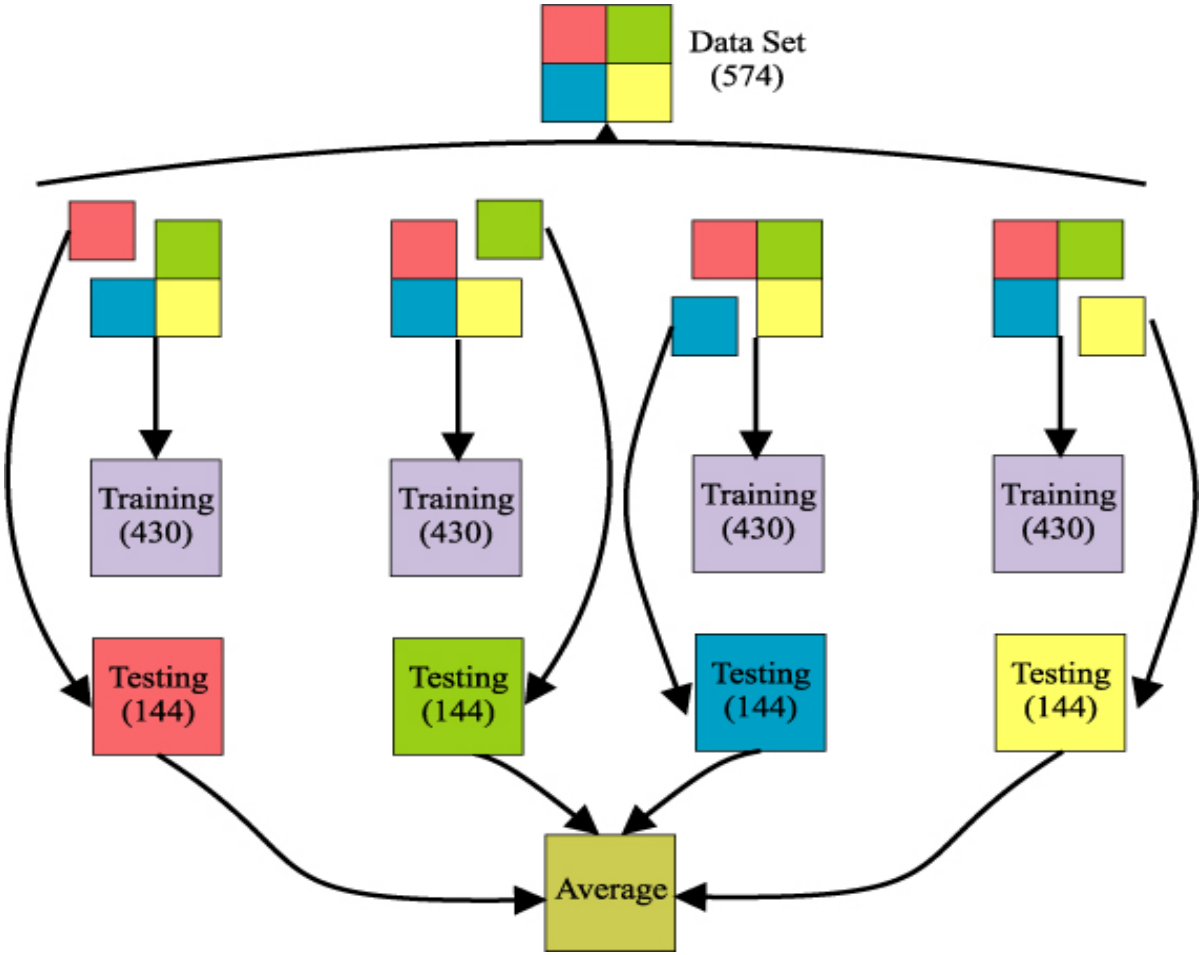
Naïve Bayes

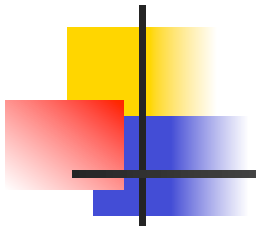
Correctly Classified Instances 48 94.1176 %

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
1	0	1	1	1	1	Iris-setosa
0.947	0.063	0.9	0.947	0.923	0.988	Iris-versicolor
0.882	0.029	0.938	0.882	0.909	0.988	Iris-virginica

K-fold Cross Validation

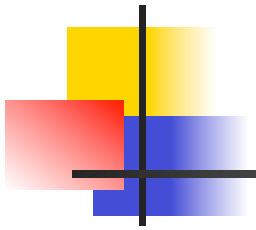




Exercise II

Perform the classification by using the following classifiers (default parameters) and the Pima Diabets dataset with a 5-fold cross validation:

- Jrip (rules)
- J48
- KNN(lazy)
- Naive Bayes (Bayes)



J48

Correctly Classified Instances	547	71.224 %
Incorrectly Classified Instances	221	28.776 %

Jrip

Correctly Classified Instances	572	74.4792 %
Incorrectly Classified Instances	196	25.5208 %

IBK

Correctly Classified Instances	540	70.3125 %
Incorrectly Classified Instances	228	29.6875 %

Naïve Bayes

Correctly Classified Instances	587	76.4323 %
Incorrectly Classified Instances	181	23.5677 %

Command Line Filtering and Classification: setting the classpth

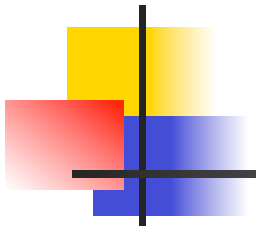
- In Unix based operating systems the classpath can be set by typing the following command:

```
export CLASSPATH=$CLASSPATH:/CompletePathOfweka/weka.jar
```

- For Windows OS:

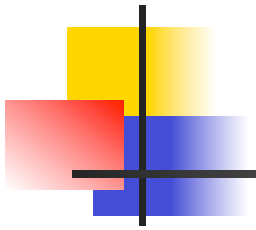
1. In the Control Panel click on System (or right click on My Computer and select Properties) and then go to the Advanced tab. There you will find a button called Environment Variables, click it.
2. Enter the following name for the variable CLASSPATH
3. Add this value /CompletePathOfweka/weka.jar, where CompletePathOfweka is your own path in which weka.jar file is located.

Check on the web instructions on how set Environmental Variables in your specific WIN OS. You will find also videos on you tube.



Command Line Filtering

- The **weka.filters** package is concerned with classes that transform datasets by removing or adding attributes, resampling the dataset, removing examples and so on.
- All filters offer the options -*i* for specifying the input dataset, and -*o* for specifying the output dataset. All others including specific parameters can be found via -*h*



Command Line Filtering: Examples

1) Write on the console, after adding weka.jar to the CLASSPATH

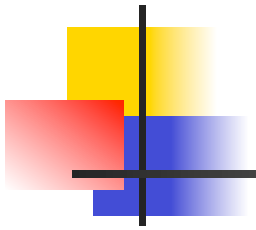
java weka.filters.unsupervised.attribute.PrincipalComponents

-I yourPahtOfDataset/iris.arff -o iris-PC.arff -c last

2) Resample creates a stratified subsample of the given dataset. This means that overall class distributions are approximately retained within the sample. A bias towards uniform class distribution can be specified via -B.

java weka.filters.supervised.instance.Resample -i yourPahtOfDataset/soybean.arff -o soybean-5%.arff -c last -Z 5

java weka.filters.supervised.instance.Resample -i yourPahtOfDataset/soybean.arff -o soybean-uniform-5%.arff -c last -Z 5 -B 1



Command Line Classification

- Any learning algorithm in WEKA is derived from the abstract weka.classifiers.Classifier class

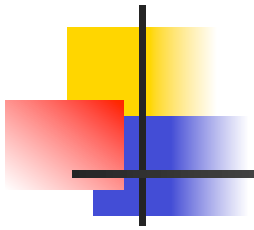
Three simple routines are needed for a basic classifier:

- a routine which generates a classifier model from a training dataset (= buildClassifier)
- a routine which evaluates the generated model on an unseen test dataset (= classifyInstance)
- a routine which generates a probability distribution for all classes (=distributionForInstance)

Example: java weka.classifiers.trees.J48 -t data/iris.arff

Command Line Classification: Parameters

-t	specifies the training file (ARFF format)
-T	specifies the test file in (ARFF format). If this parameter is missing, a crossvalidation will be performed (default: ten-fold cv)
-x	This parameter determines the number of folds for the cross-validation. A cv will only be performed if -T is missing.
-c	As we already know from the weka.filters section, this parameter sets the class variable with a one-based index.
-d	The model after training can be saved via this parameter. Each classifier has a different binary format for the model, so it can only be read back by the exact same classifier on a compatible dataset. Only the model on the training set is saved, not the multiple models generated via cross-validation.
-l	Loads a previously saved model, usually for testing on new, previously unseen data. In that case, a compatible test file should be specified, i.e. the same attributes in the same order.



Command Line Classification: An Example

- Creating a Model

java weka.classifiers.trees.J48 -t data/appendicitis-10-1tra.arff -d modelApp

- Using a Model

java weka.classifiers.trees.J48 -T data/appendicitis-10-1tst.arff -l modelApp