Attribute Discretization and Selection

Clustering

NIKOLA MILIKIĆ
nikola.milikic@fon.bg.ac.rs

UROŠ KRČADINAC
uros@krcadinac.com

Naive Bayes Features

- Intended primarily for the work with nominal attributes
- In case of numeric attributes
  - Use the propability distribution of attributes (Normal distribution is default) for probability estimation for the each attribute
  - Discretize the attribute’s values

Attribute Discretization

Discretization is the process of tranformation numeric data into nominal data, by putting the numeric values into distinct groups, which lenght is fixed.

Common approaches:

- Unsupervised:
  - Equal-width binning
  - Equal-frequency binning
- Supervised – classes are taken into account

Equal-Width Binning

Equal-width binning divides the scope of possible values into N subscopes (bins) of the same width:

width = (max value – min value) / N

Example: If the scope of the values is between 0 and 100, we should create 5 subscopes (bins) in the following manner:

Width = (100 – 0) / 5 = 20

Subscopes (bins): [0-20], (20-40], (40-60], (60-80], (80-100]

Usually, the first and the final subscope (bin) are being expended in order to include possible values outside the original scope.

Equal-frequency binning

Equal-frequency binning (or equal-height binning) divides the scope of possible values into N subscopes where each subscope (bin) carries the same number of instances:

Example: We want to put the following values in 5 subscopes (bins):

5, 7, 12, 35, 65, 82, 84, 88, 90, 95

So, each subscope will have 2 instances:

5, 7, | 12, 35, | 65, 82, | 84, 88, | 90, 95

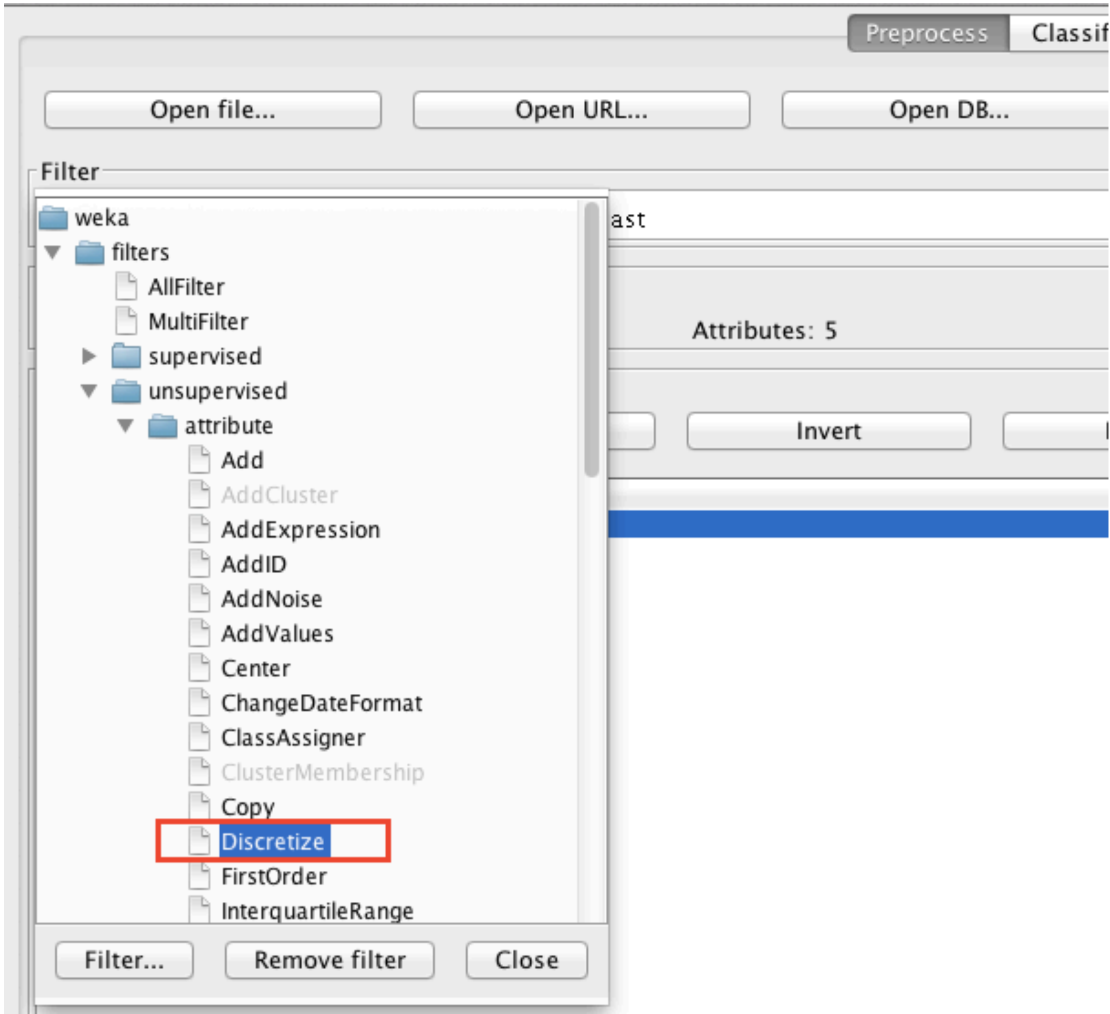
Discretization in Weka

We apply certain *Filters* to attributes we want to discretize.

Preprocess tab

Option: *Choose -> Filter*

filters/unsupervised/attribute ***Discretize***.

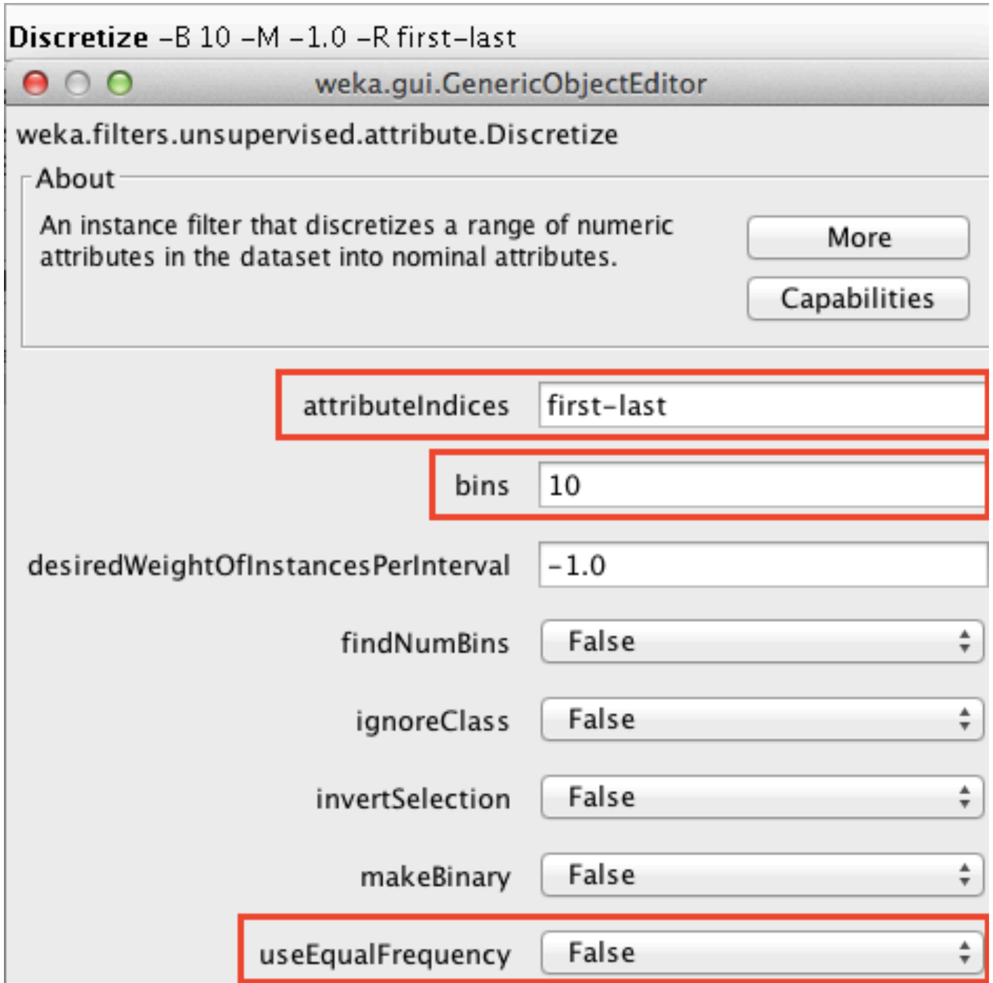


FishersIrisDataset.arff

Discretization in Weka

Equal-width binning is the default option.

- ***attributeIndices*** – the *first-last* value means that we are discretizing all values. We can also name the attribute numbers.
- ***bins*** – the desired number of scopes (bins)
- ***useEqualFrequency*** – *false* by default; *true* if we use Equal Frequency binning



Discretize -B 10 -M -1.0 -R first-last

weka.gui.GenericObjectEditor

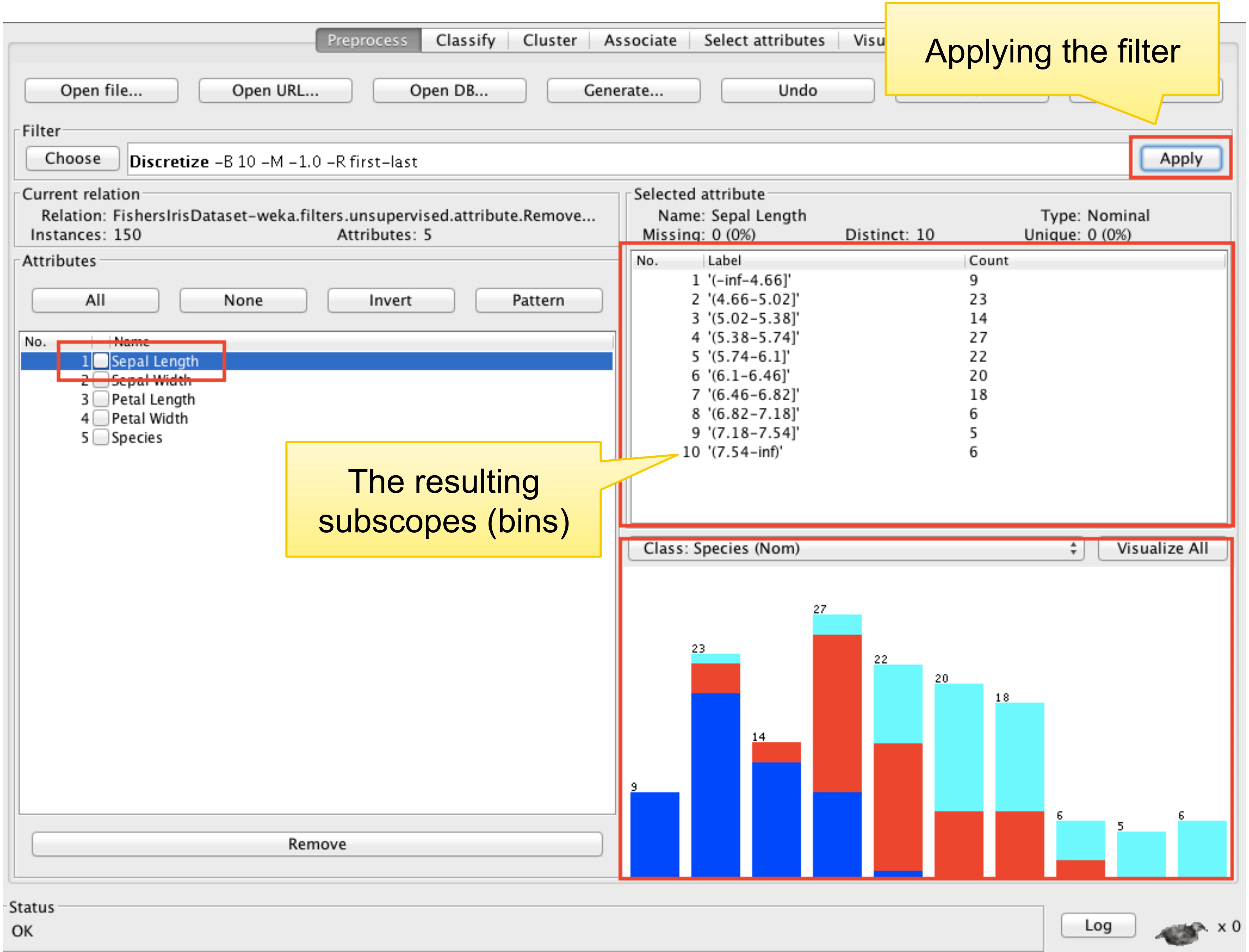
weka.filters.unsupervised.attribute.Discretize

About

An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.

attributeIndices: first-last
bins: 10
desiredWeightOfInstancesPerInterval: -1.0
findNumBins: False
ignoreClass: False
invertSelection: False
makeBinary: False
useEqualFrequency: False

Discretization in Weka



Applying the filter

Filter

Discretize -B 10 -M -1.0 -R first-last

Current relation

Relation: FishersIrisDataset-weka.filters.unsupervised.attribute.Remove...
Instances: 150 Attributes: 5

Attributes

No.		Name
1	□	Sepal Length
2	□	Sepal Width
3	□	Petal Length
4	□	Petal Width
5	□	Species

Selected attribute

Name: Sepal Length Type: Nominal
Missing: 0 (0%) Distinct: 10 Unique: 0 (0%)

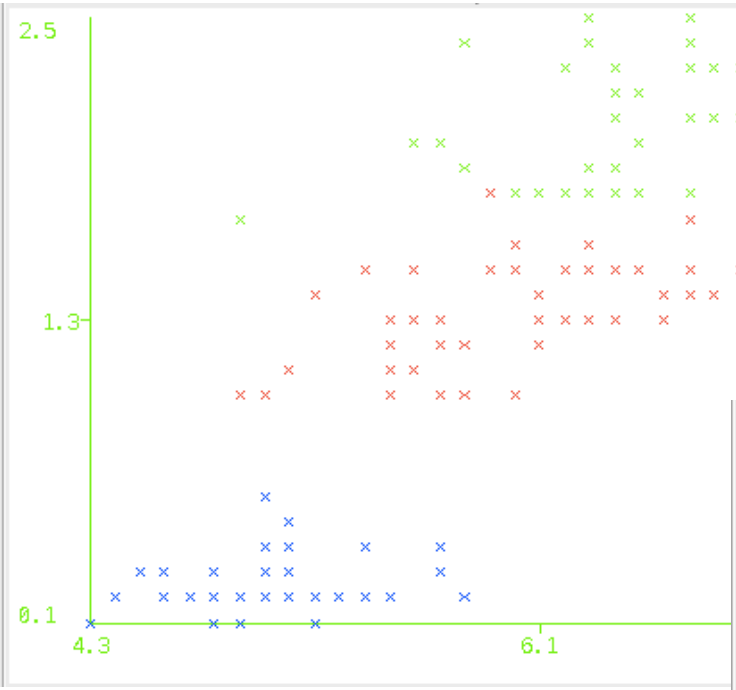
No.	Label	Count
1	'(-inf-4.66]'	9
2	'(4.66-5.02]'	23
3	'(5.02-5.38]'	14
4	'(5.38-5.74]'	27
5	'(5.74-6.1]'	22
6	'(6.1-6.46]'	20
7	'(6.46-6.82]'	18
8	'(6.82-7.18]'	6
9	'(7.18-7.54]'	5
10	'(7.54-inf)'	6

The resulting subscopes (bins)

Class: Species (Nom)

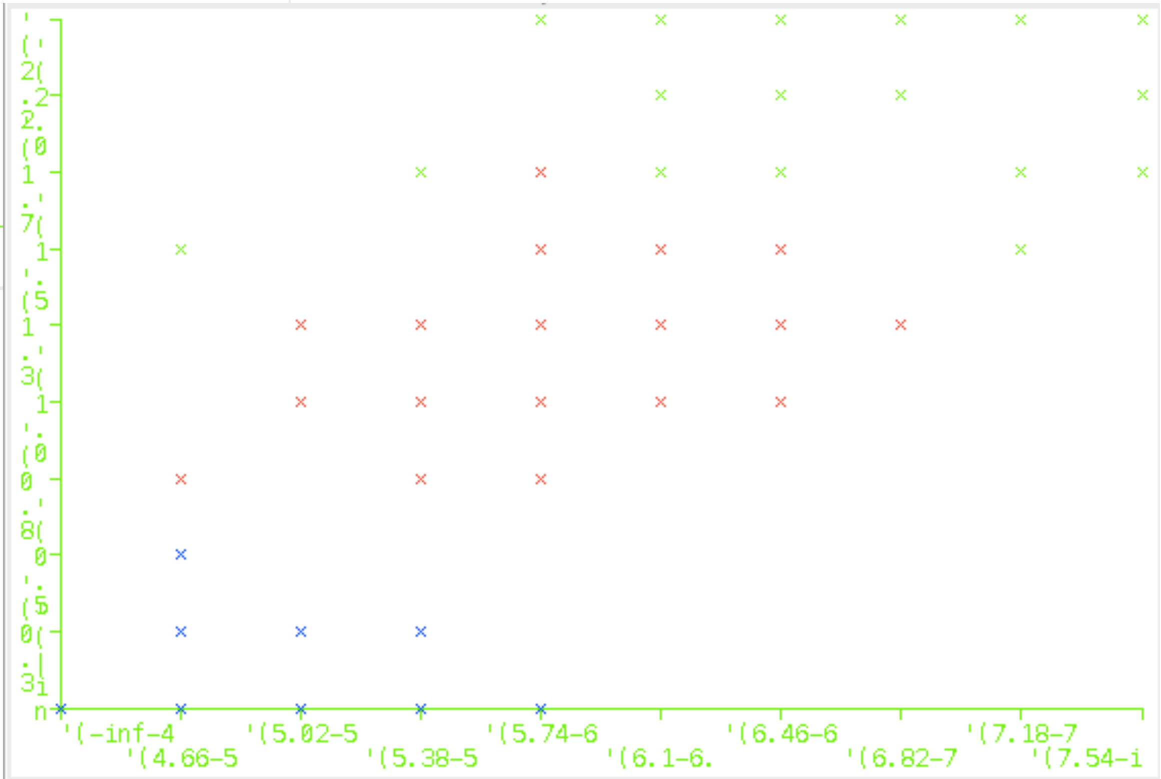
Status
OK

Data, before and after discretization



Before

After



Attribute Selection

Attribute Selection (or Feature Selection) is the process of choosing a subset of relevant attributes that will be used during the further analysis.

It is being applied in cases where the dataset contains attributes which are redudant and/or irrelevant.

- Redundant attributes are the ones that do not provide more information than the attributes we already have in our dataset.
- Irrelevant attributes are the ones that are useless in the context of the current analysis.

Attribute Selection Advantages

Excessive attributes can degrade the performance of the model.

Advantages:

- Advances the readability of the model (because now the model contains only the relevant attributes)
- Shortens the training time
- Generalization power is higher because it lowers the possibility of overfitting

If the problem is well-known, the best way to select attribute is to do it manually. However, automated apporaches also give good results.

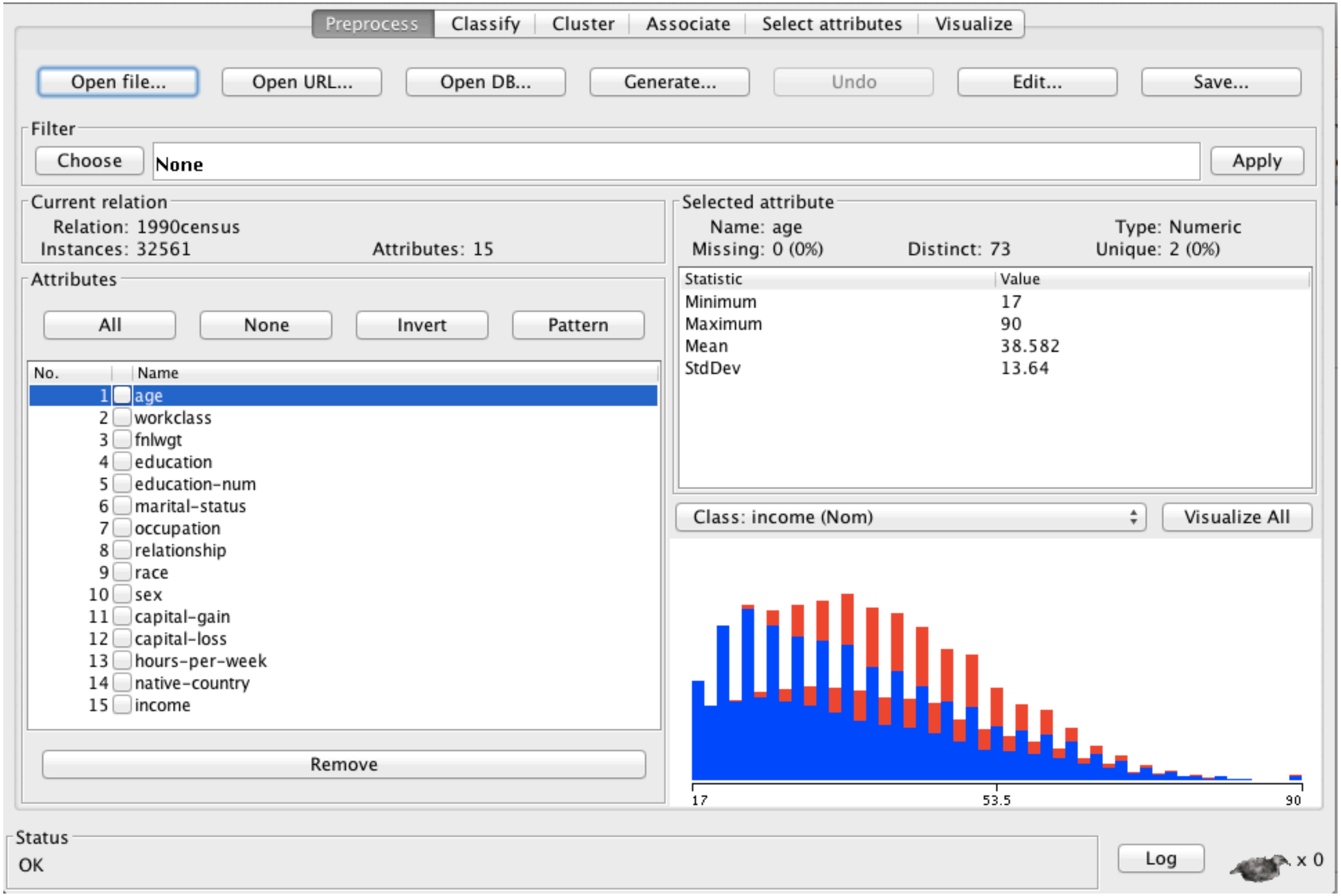
Approaches to Attribute Selection

Two approaches:

- *Filter* method – use the approximation based on the general features of the data.
- *Wrapper* method – attribute subsets are being evaluated by using the maching learning algorithm, applied to the dataset. The name *Wrapper* comes from the fact that the algorithm is wrapped within the process of selection. The chosen subset of attributes is the one for which the algorithm gives the best results.

Attribute Selection Example

census90-income.arff



Filter

None

Current relation

Relation: 1990census
Instances: 32561   Attributes: 15

Attributes

No.	Name
1	☐ age
2	☐ workclass
3	☐ fnlwgt
4	☐ education
5	☐ education-num
6	☐ marital-status
7	☐ occupation
8	☐ relationship
9	☐ race
10	☐ sex
11	☐ capital-gain
12	☐ capital-loss
13	☐ hours-per-week
14	☐ native-country
15	☐ income

Selected attribute

Name: age   Type: Numeric
Missing: 0 (0%)   Distinct: 73   Unique: 2 (0%)

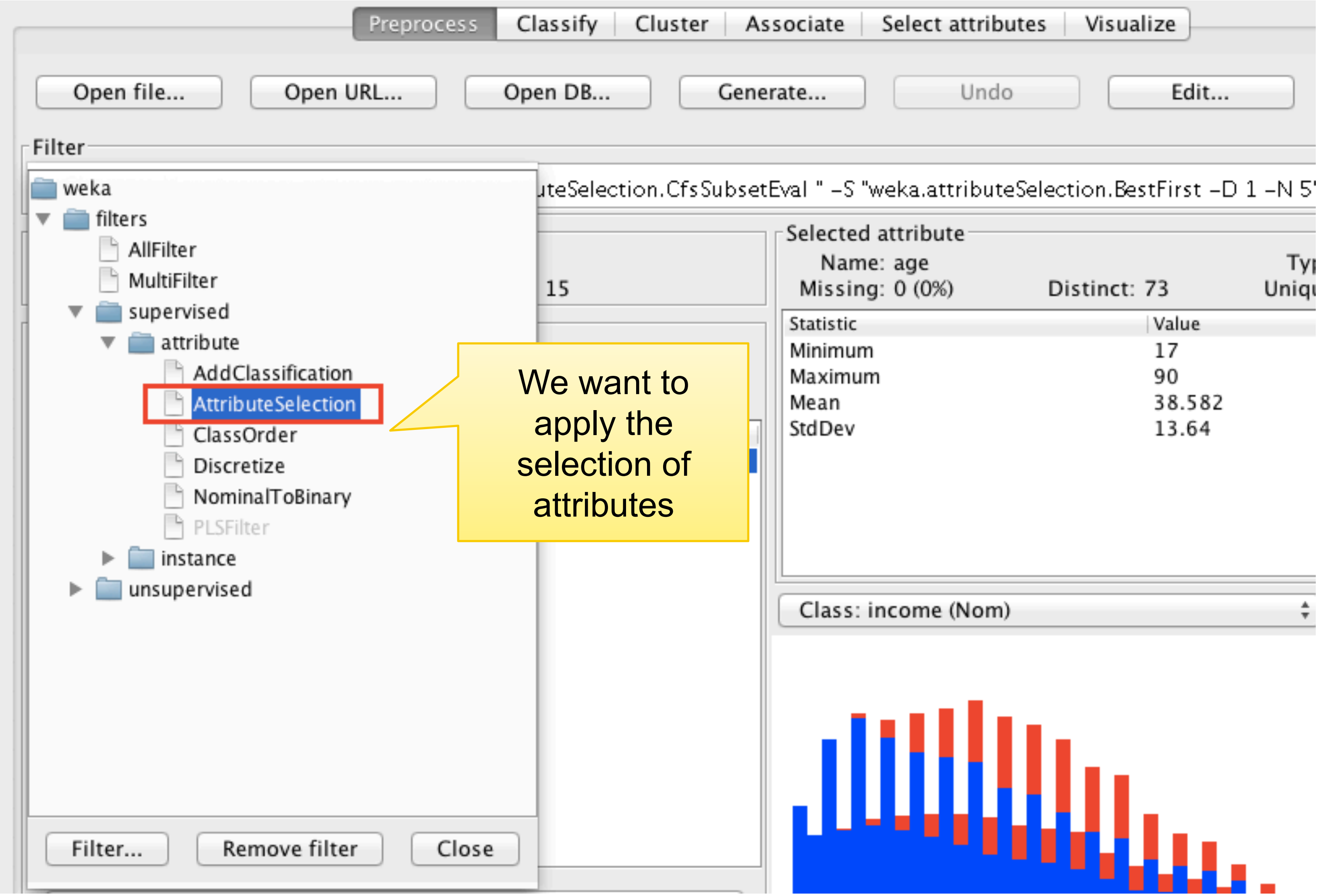
Statistic	Value
Minimum	17
Maximum	90
Mean	38.582
StdDev	13.64

Class: income (Nom)

Status

OK

Attribute Selection Example



uteSelection.CfsSubsetEval " -S "weka.attributeSelection.BestFirst -D 1 -N 5'

15

Selected attribute

Name: age Typ

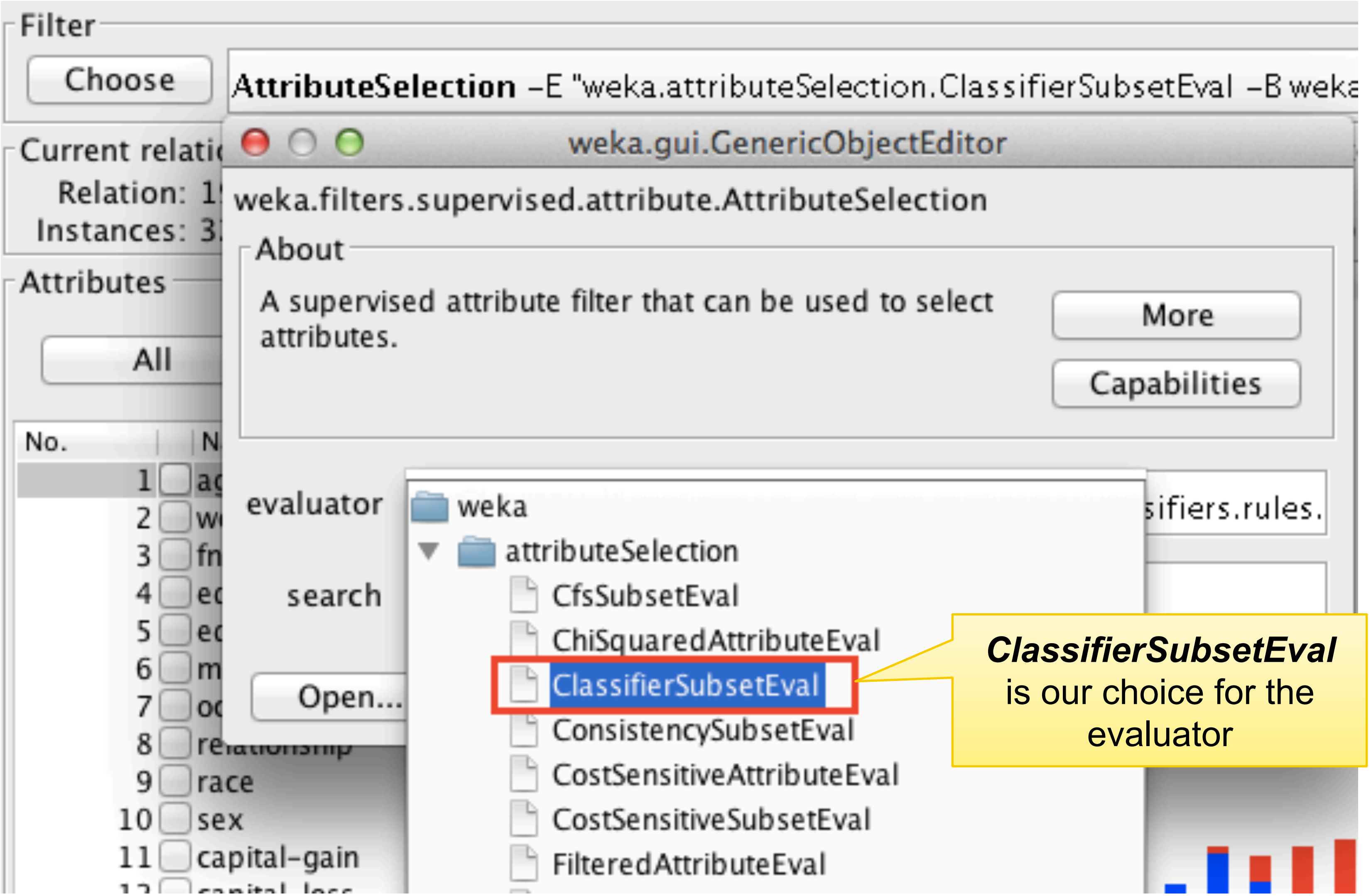
Missing: 0 (0%) Distinct: 73 Uniq

Statistic	Value
Minimum	17
Maximum	90
Mean	38.582
StdDev	13.64

Class: income (Nom)

We want to apply the selection of attributes

Attribute Selection Example



AttributeSelection -E "weka.attributeSelection.ClassifierSubsetEval -B weka

weka.gui.GenericObjectEditor

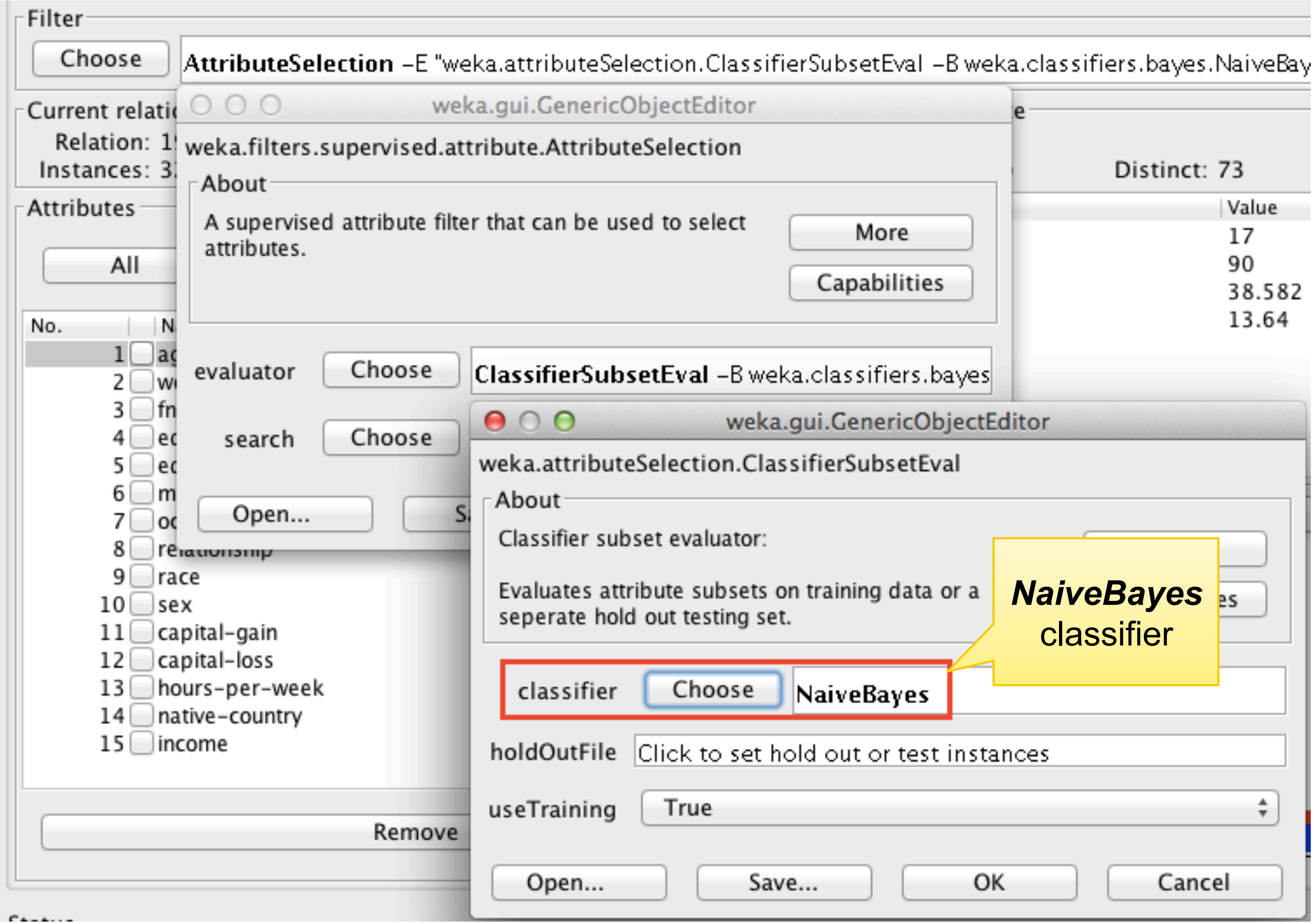
weka.filters.supervised.attribute.AttributeSelection

About

A supervised attribute filter that can be used to select attributes.

ClassifierSubsetEval is our choice for the evaluator

Attribute Selection Example



Filter

Choose | AttributeSelection -E "weka.attributeSelection.ClassifierSubsetEval -B weka.classifiers.bayes.NaiveBay

weka.gui.GenericObjectEditor

weka.filters.supervised.attribute.AttributeSelection

About

A supervised attribute filter that can be used to select attributes.

evaluator: ClassifierSubsetEval -B weka.classifiers.bayes

search

weka.gui.GenericObjectEditor

weka.attributeSelection.ClassifierSubsetEval

About

Classifier subset evaluator:

Evaluates attribute subsets on training data or a seperate hold out testing set.

classifier: NaiveBayes

holdOutFile: Click to set hold out or test instances

useTraining: True

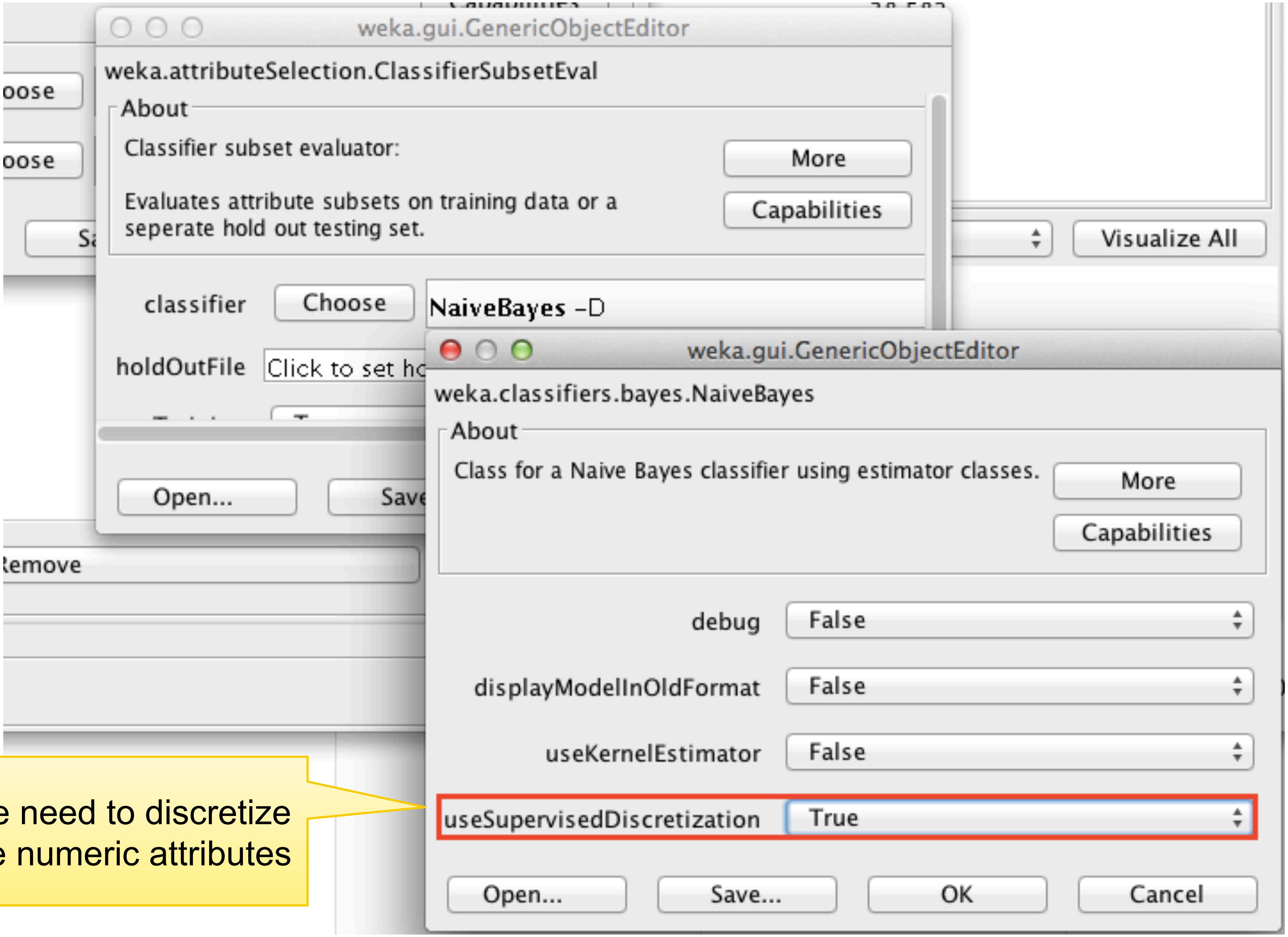
Distinct: 73

Value
17
90
38.582
13.64

No.	Name
1	ag
2	w
3	fn
4	ec
5	ec
6	m
7	oc
8	relationship
9	race
10	sex
11	capital-gain
12	capital-loss
13	hours-per-week
14	native-country
15	income

NaiveBayes **classifier**

Attribute Selection Example



weka.attributeSelection.ClassifierSubsetEval

About

Classifier subset evaluator:

Evaluates attribute subsets on training data or a seperate hold out testing set.

classifier: NaiveBayes -D

holdOutFile: Click to set hc

weka.classifiers.bayes.NaiveBayes

About

Class for a Naive Bayes classifier using estimator classes.

debug: False

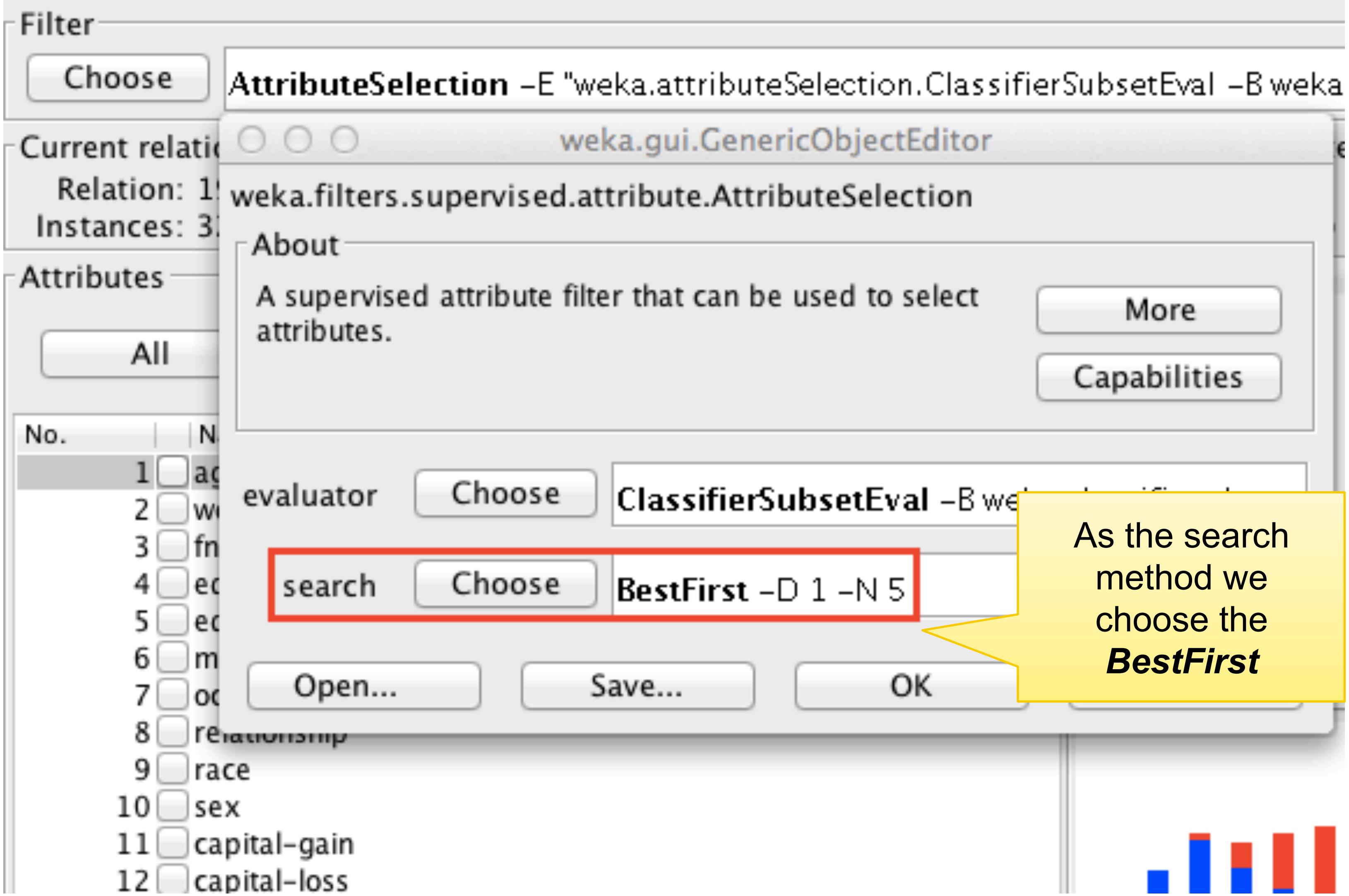
displayModelInOldFormat: False

useKernelEstimator: False

useSupervisedDiscretization: True

We need to discretize the numeric attributes

Attribute Selection Example



Filter

Choose | AttributeSelection -E "weka.attributeSelection.ClassifierSubsetEval -B weka

Current relati

Relation: 1

Instances: 3

Attributes

All

No.		N
1	□	ag
2	□	w
3	□	fn
4	□	ed
5	□	ed
6	□	m
7	□	oc
8	□	relationship
9	□	race
10	□	sex
11	□	capital-gain
12	□	capital-loss

weka.gui.GenericObjectEditor

weka.filters.supervised.attribute.AttributeSelection

About

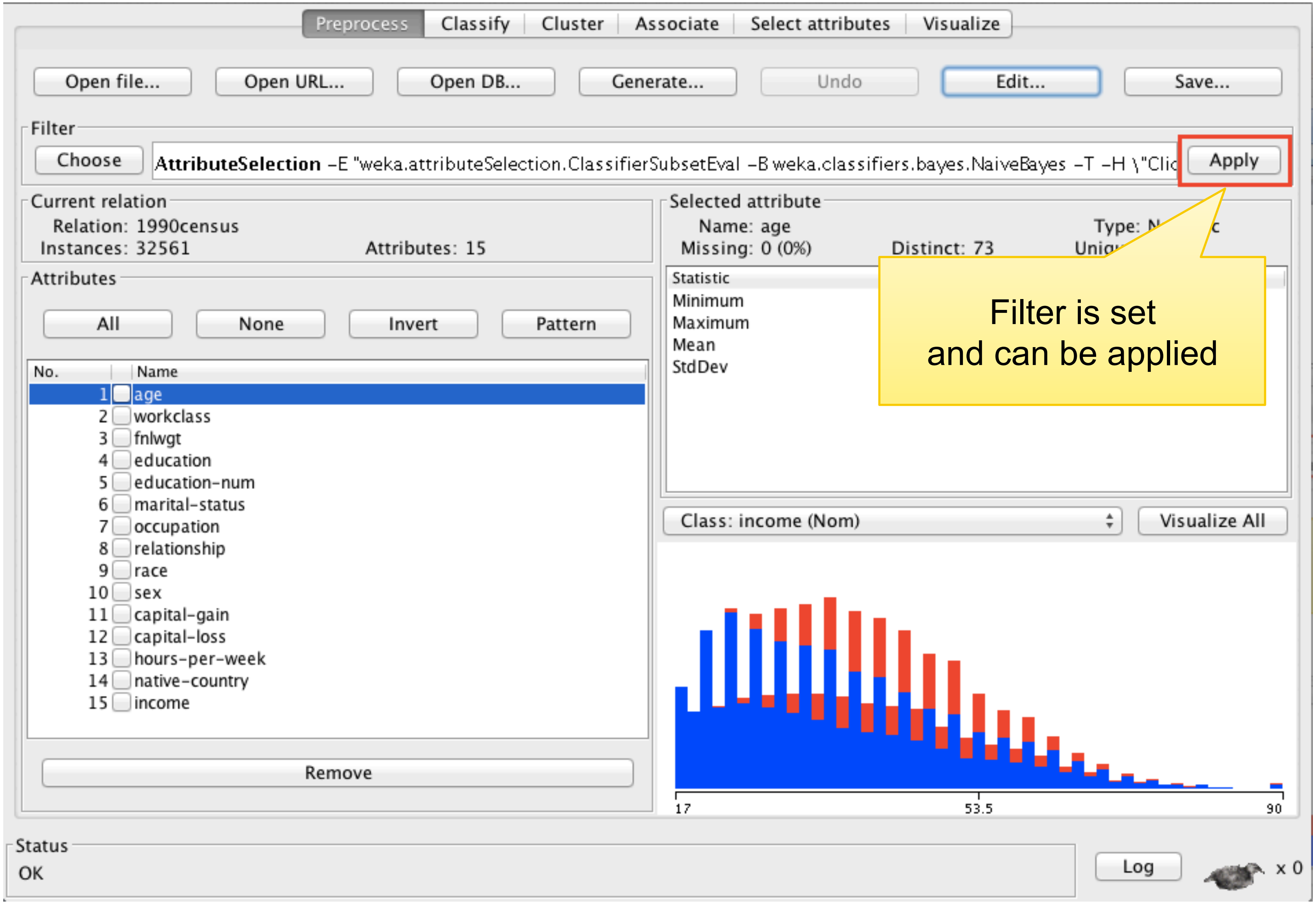
A supervised attribute filter that can be used to select attributes.

evaluator: Choose | ClassifierSubsetEval -B we

search: Choose | BestFirst -D 1 -N 5

As the search method we choose the ***BestFirst***

Attribute Selection Example



Filter

AttributeSelection -E "weka.attributeSelection.ClassifierSubsetEval -B weka.classifiers.bayes.NaiveBayes -T -H \"Clic

Current relation

Relation: 1990census

Instances: 32561 Attributes: 15

Selected attribute

Name: age Type: N c

Missing: 0 (0%) Distinct: 73 Uniq

Statistic
Minimum
Maximum
Mean
StdDev

Attributes

No.		Name
1	□	age
2	□	workclass
3	□	fnlwgt
4	□	education
5	□	education-num
6	□	marital-status
7	□	occupation
8	□	relationship
9	□	race
10	□	sex
11	□	capital-gain
12	□	capital-loss
13	□	hours-per-week
14	□	native-country
15	□	income

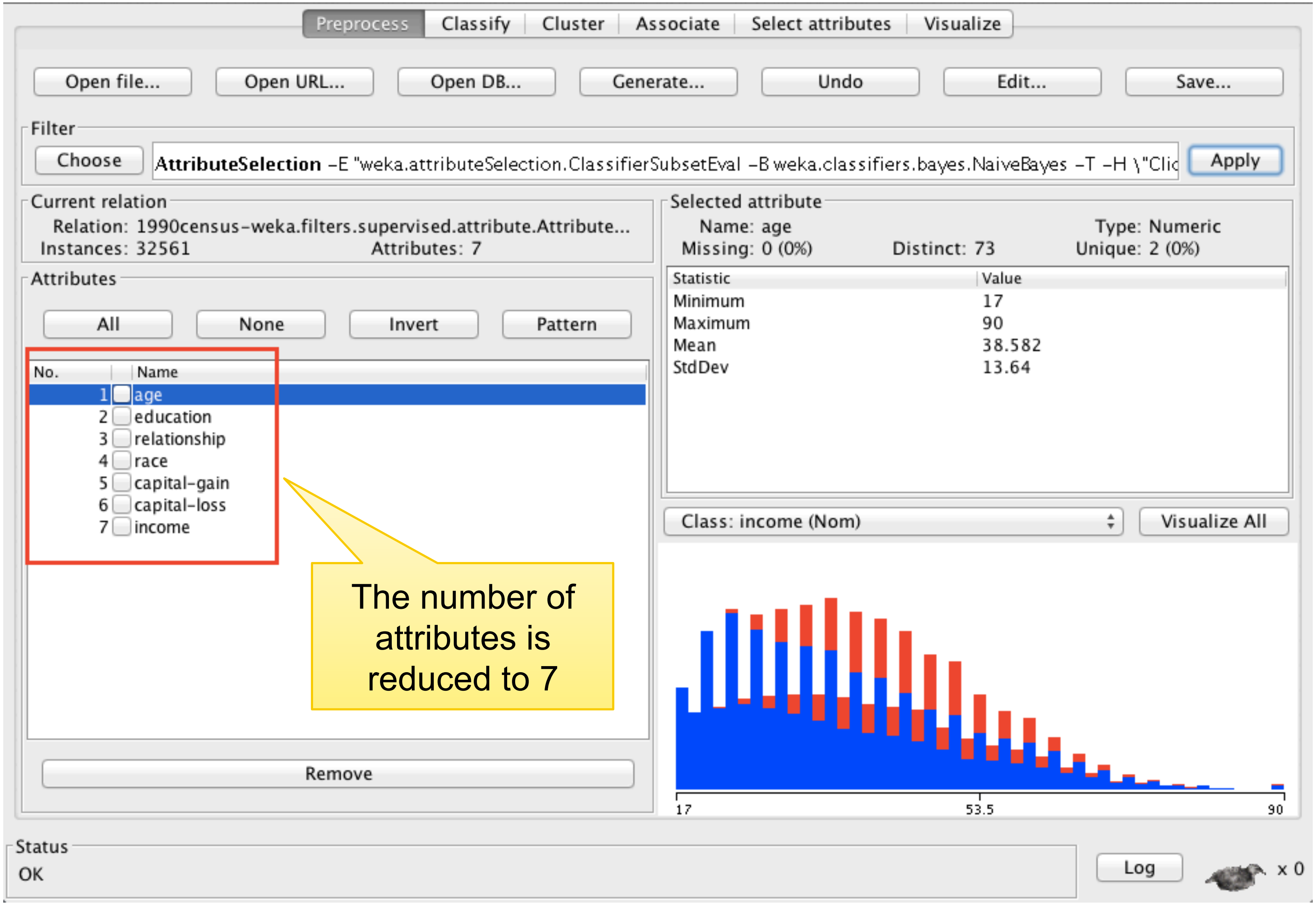
Class: income (Nom)

Status

OK

Filter is set
and can be applied

Attribute Selection Example



Filter

AttributeSelection -E "weka.attributeSelection.ClassifierSubsetEval -B weka.classifiers.bayes.NaiveBayes -T -H \"Clic

Current relation

Relation: 1990census-weka.filters.supervised.attribute.Attribute...

Instances: 32561 Attributes: 7

Attributes

No.	Name
1 □	age
2 □	education
3 □	relationship
4 □	race
5 □	capital-gain
6 □	capital-loss
7 □	income

Selected attribute

Name: age Type: Numeric

Missing: 0 (0%) Distinct: 73 Unique: 2 (0%)

Statistic	Value
Minimum	17
Maximum	90
Mean	38.582
StdDev	13.64

Class: income (Nom)

Status

OK

The number of attributes is reduced to 7

Clustering

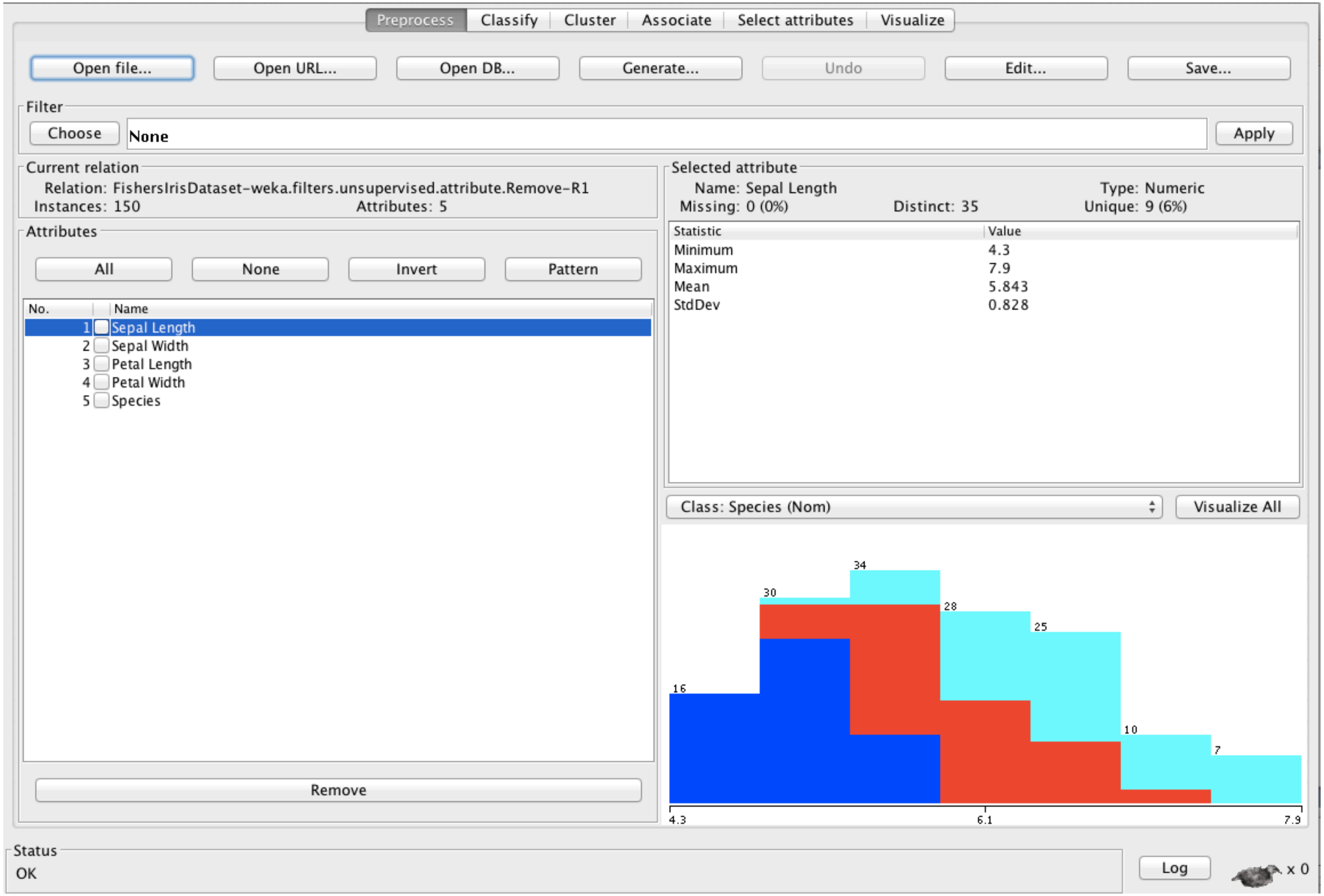
Clustering belongs to a group of techiques of unsupervised learning. It enables grouping instances into groups, where we know which are the possible groups *in advance*.

These groups are called **clusters**.

As the result of clustering each instance is being added *a new attribute* – the cluster to which it belongs. The clustering is said to be successful if the final clusters make sense, if they could be given meaningful names.

K-Means algorithm in Weka

FishersIrisDataset.arff



Filter
None

Current relation
Relation: FishersIrisDataset-weka.filters.unsupervised.attribute.Remove-R1
Instances: 150 Attributes: 5

Attributes

No.		Name
1	□	Sepal Length
2	□	Sepal Width
3	□	Petal Length
4	□	Petal Width
5	□	Species

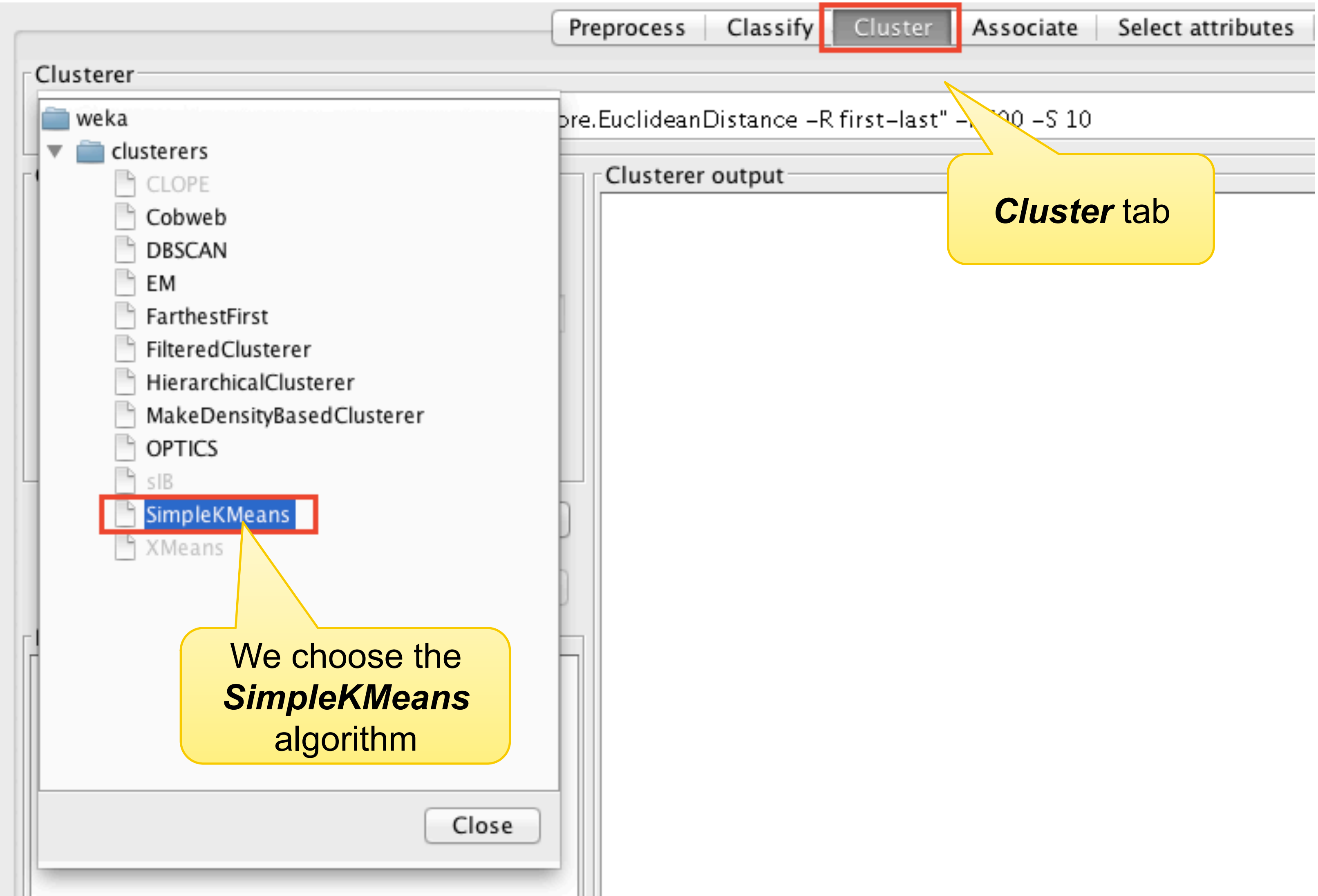
Selected attribute
Name: Sepal Length Type: Numeric
Missing: 0 (0%) Distinct: 35 Unique: 9 (6%)

Statistic	Value
Minimum	4.3
Maximum	7.9
Mean	5.843
StdDev	0.828

Class: Species (Nom)

Status
OK

Choosing the clustering algorithm



ore.EuclideanDistance -R first-last" -…00 -S 10

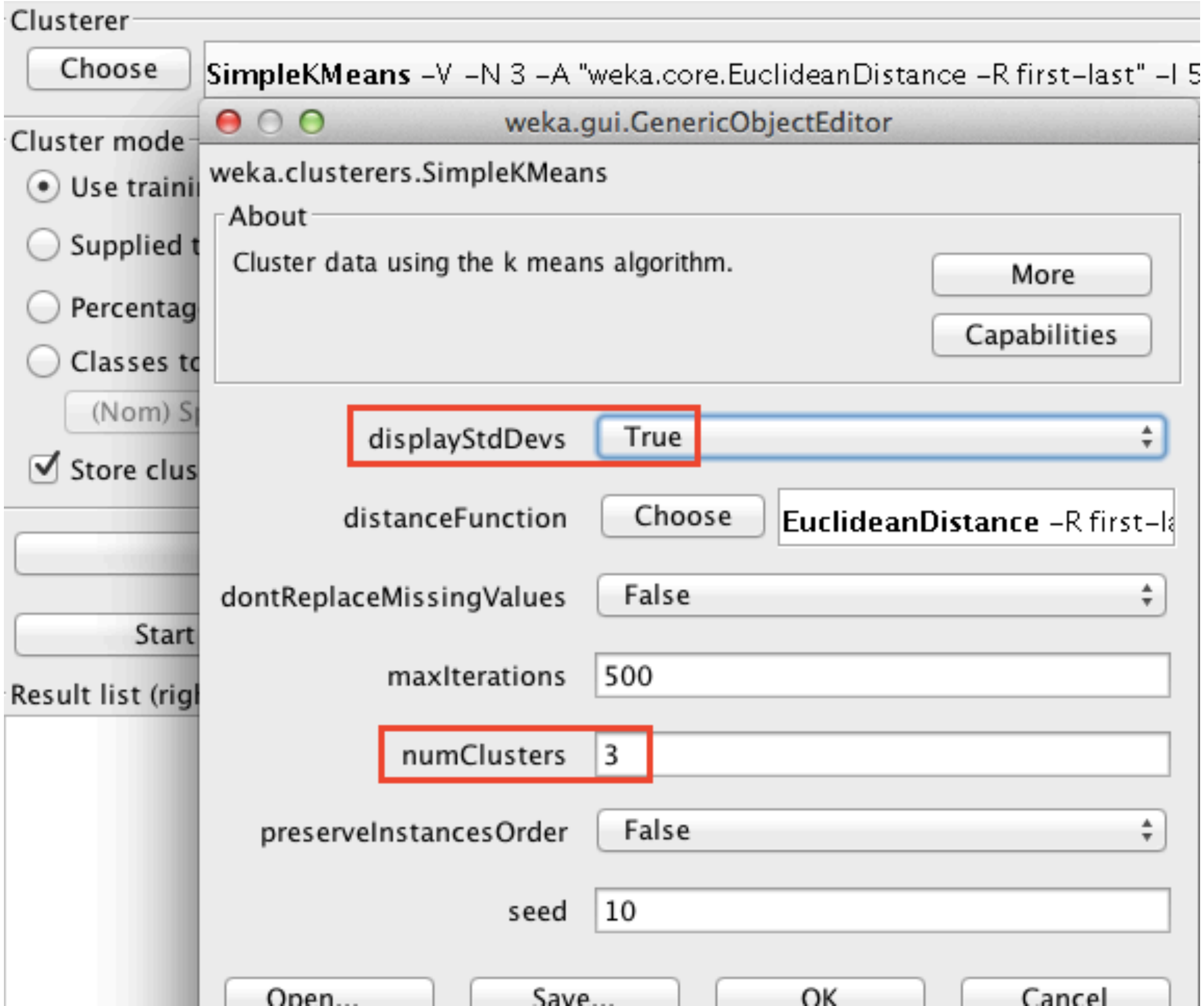
Cluster tab

We choose the ***SimpleKMeans*** algorithm

Parameter settings

numClusters – the number of desired clusters;
we set it to 3 because we have 3 kinds

displayStdDevs – if *true*, the standard deviation will be displayed



Clusterer

SimpleKMeans -V -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 5

weka.gui.GenericObjectEditor

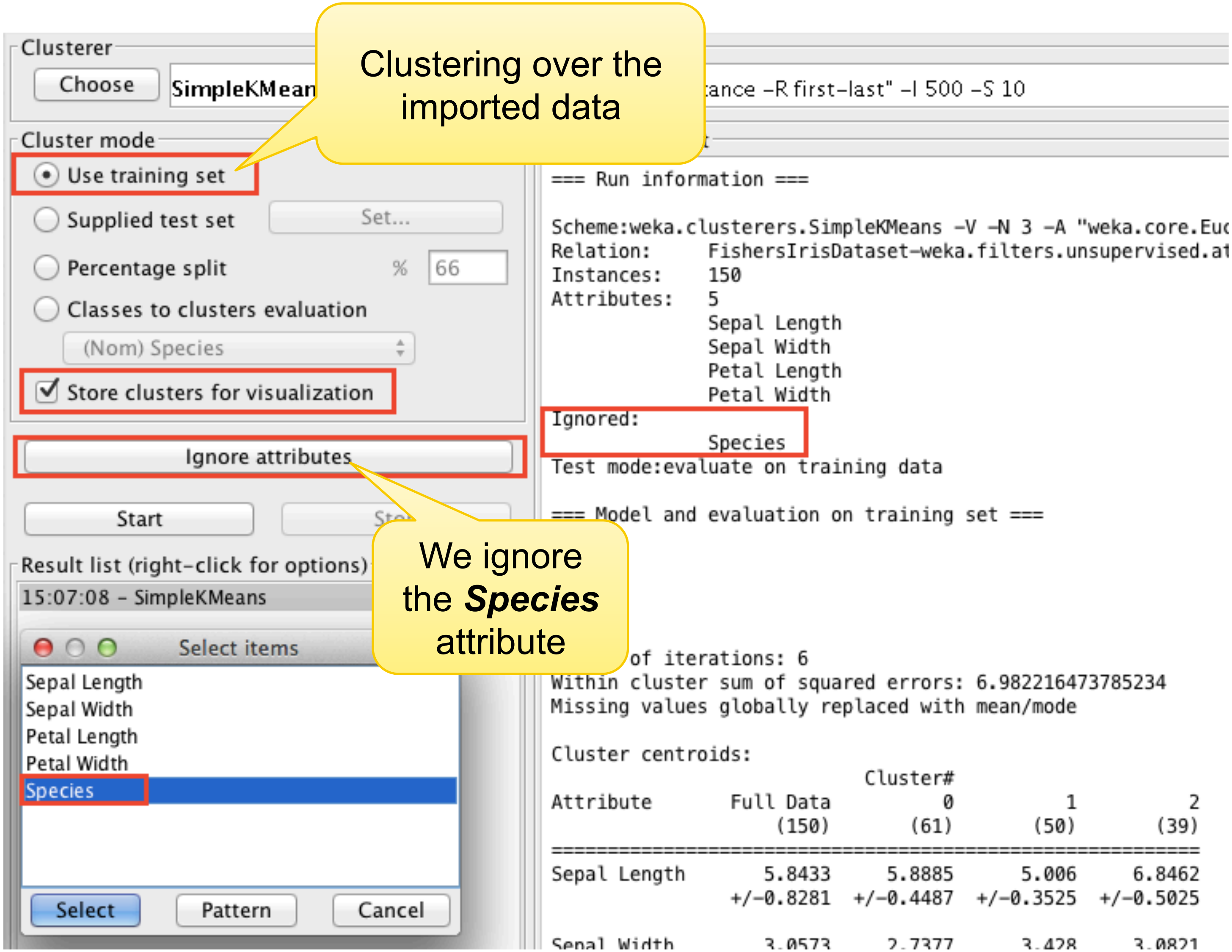
weka.clusterers.SimpleKMeans

About

Cluster data using the k means algorithm.

displayStdDevs	True
distanceFunction	EuclideanDistance -R first-la
dontReplaceMissingValues	False
maxIterations	500
numClusters	3
preserveInstancesOrder	False
seed	10

Running the Clustering



SimpleKMean… ance -R first-last" -I 500 -S 10

- ● Use training set
- ○ Supplied test set
- ○ Percentage split % 66
- ○ Classes to clusters evaluation (Nom) Species
- ☑ Store clusters for visualization

15:07:08 - SimpleKMeans

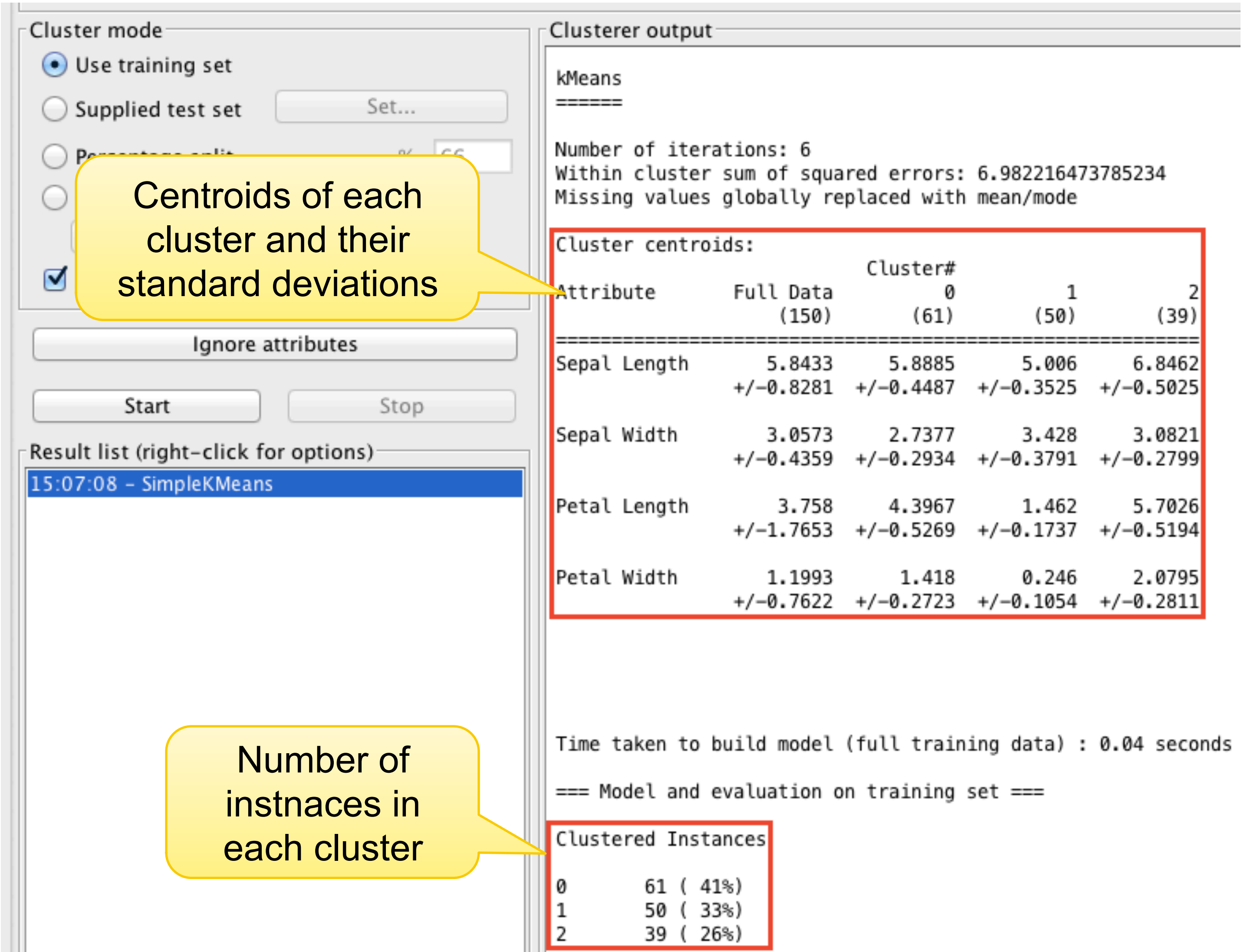
Sepal Length
Sepal Width
Petal Length
Petal Width
Species

```
=== Run information ===

Scheme:weka.clusterers.SimpleKMeans -V -N 3 -A "weka.core.Euc
Relation:     FishersIrisDataset-weka.filters.unsupervised.at
Instances:    150
Attributes:   5
              Sepal Length
              Sepal Width
              Petal Length
              Petal Width
Ignored:
              Species
Test mode:evaluate on training data

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




      of iterations: 6
Within cluster sum of squared errors: 6.982216473785234
Missing values globally replaced with mean/mode

Cluster centroids:
                           Cluster#
Attribute       Full Data         0          1          2
                    (150)      (61)       (50)       (39)
==========================================================
Sepal Length       5.8433     5.8885      5.006     6.8462
               +/-0.8281  +/-0.4487  +/-0.3525  +/-0.5025

Sepal Width        3.0573     2.7377      3.428     3.0821
```

Clustering over the imported data

We ignore the ***Species*** attribute

Results of Clustering



Cluster mode

● Use training set

○ Supplied test set Set...

○ Percentage split % 66

○

☑

Ignore attributes

Start Stop

Result list (right-click for options)

15:07:08 - SimpleKMeans

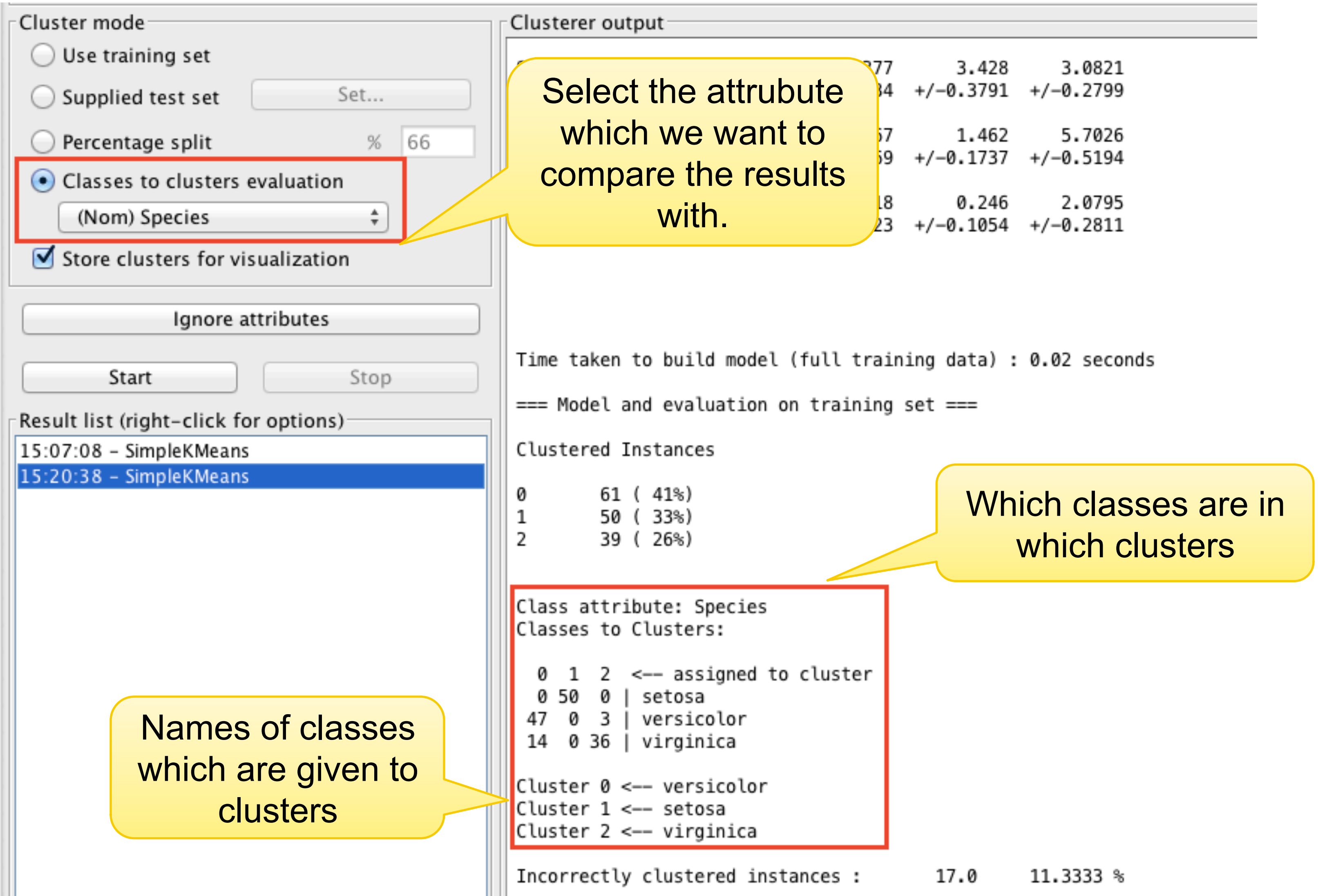
Clusterer output

```
kMeans
======

Number of iterations: 6
Within cluster sum of squared errors: 6.982216473785234
Missing values globally replaced with mean/mode

Cluster centroids:
                            Cluster#
Attribute       Full Data          0          1          2
                    (150)       (61)       (50)       (39)
===========================================================
Sepal Length       5.8433     5.8885      5.006     6.8462
               +/-0.8281  +/-0.4487  +/-0.3525  +/-0.5025

Sepal Width        3.0573     2.7377      3.428     3.0821
               +/-0.4359  +/-0.2934  +/-0.3791  +/-0.2799

Petal Length        3.758     4.3967      1.462     5.7026
               +/-1.7653  +/-0.5269  +/-0.1737  +/-0.5194

Petal Width        1.1993      1.418      0.246     2.0795
               +/-0.7622  +/-0.2723  +/-0.1054  +/-0.2811




Time taken to build model (full training data) : 0.04 seconds

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

Clustered Instances

0       61 ( 41%)
1       50 ( 33%)
2       39 ( 26%)
```

Centroids of each cluster and their standard deviations

Number of instnaces in each cluster

Evaluation of Results



Cluster mode

- ○ Use training set
- ○ Supplied test set
- ○ Percentage split % 66
- ● Classes to clusters evaluation
  - (Nom) Species
- ☑ Store clusters for visualization

Result list (right-click for options)

- 15:07:08 - SimpleKMeans
- 15:20:38 - SimpleKMeans

Clusterer output

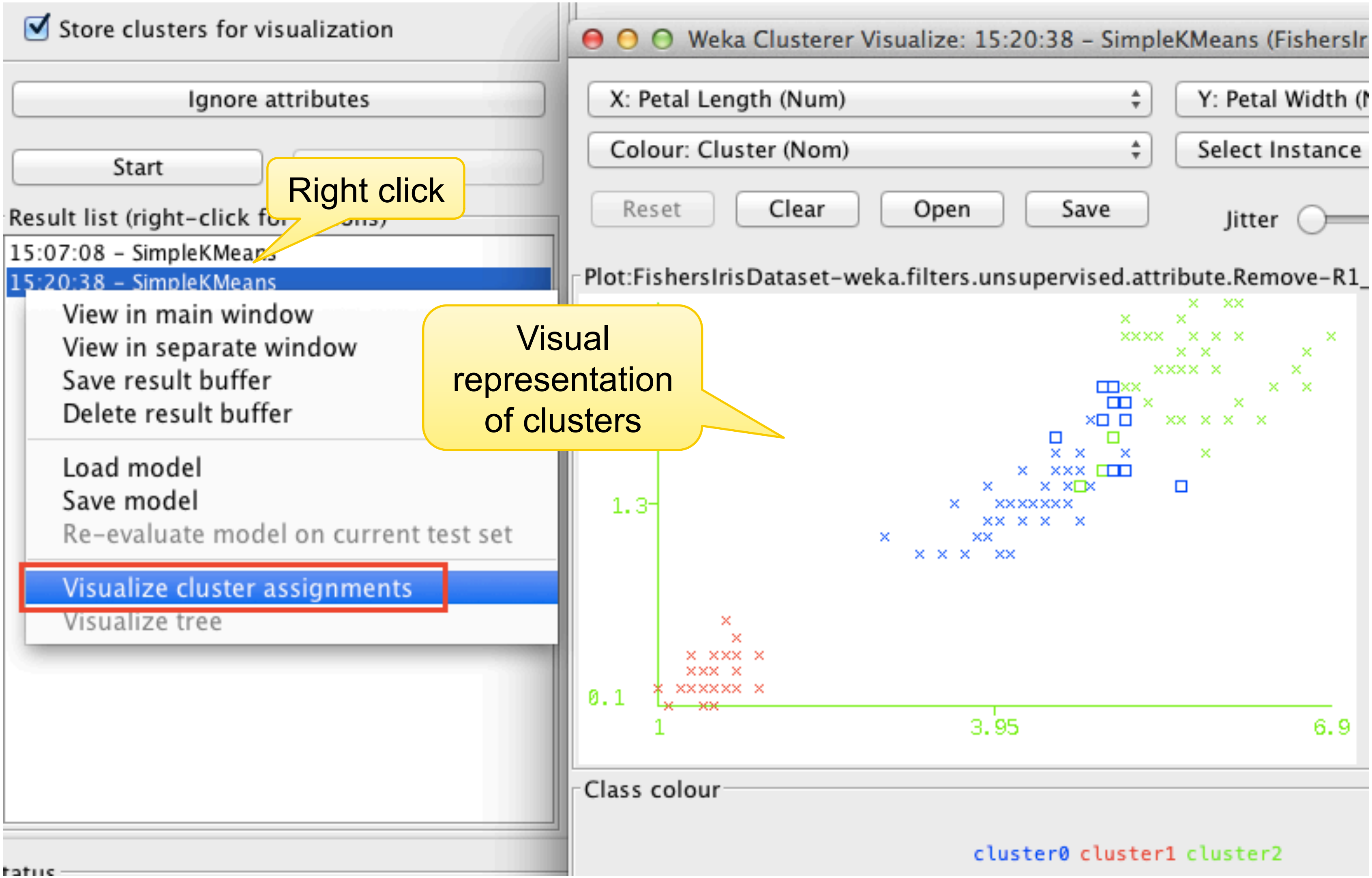
```
        77       3.428      3.0821
        34  +/-0.3791  +/-0.2799

        57       1.462      5.7026
        59  +/-0.1737  +/-0.5194

        18       0.246      2.0795
        23  +/-0.1054  +/-0.2811



Time taken to build model (full training data) : 0.02 seconds

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

Clustered Instances

0        61 ( 41%)
1        50 ( 33%)
2        39 ( 26%)

Class attribute: Species
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  0 50  0 | setosa
 47  0  3 | versicolor
 14  0 36 | virginica

Cluster 0 <-- versicolor
Cluster 1 <-- setosa
Cluster 2 <-- virginica

Incorrectly clustered instances :        17.0      11.3333 %
```

Select the attrubute which we want to compare the results with.

Which classes are in which clusters

Names of classes which are given to clusters

Visualization of Clusters



☑ Store clusters for visualization

Ignore attributes

Start

Result list (right-click for ...ons)

15:07:08 - SimpleKMeans

15:20:38 - SimpleKMeans

- 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
- Visualize cluster assignments
- Visualize tree

tatus

Weka Clusterer Visualize: 15:20:38 - SimpleKMeans (FishersIr

X: Petal Length (Num)

Y: Petal Width (N

Colour: Cluster (Nom)

Select Instance

Reset | Clear | Open | Save

Jitter

Plot:FishersIrisDataset-weka.filters.unsupervised.attribute.Remove-R1_

Class colour

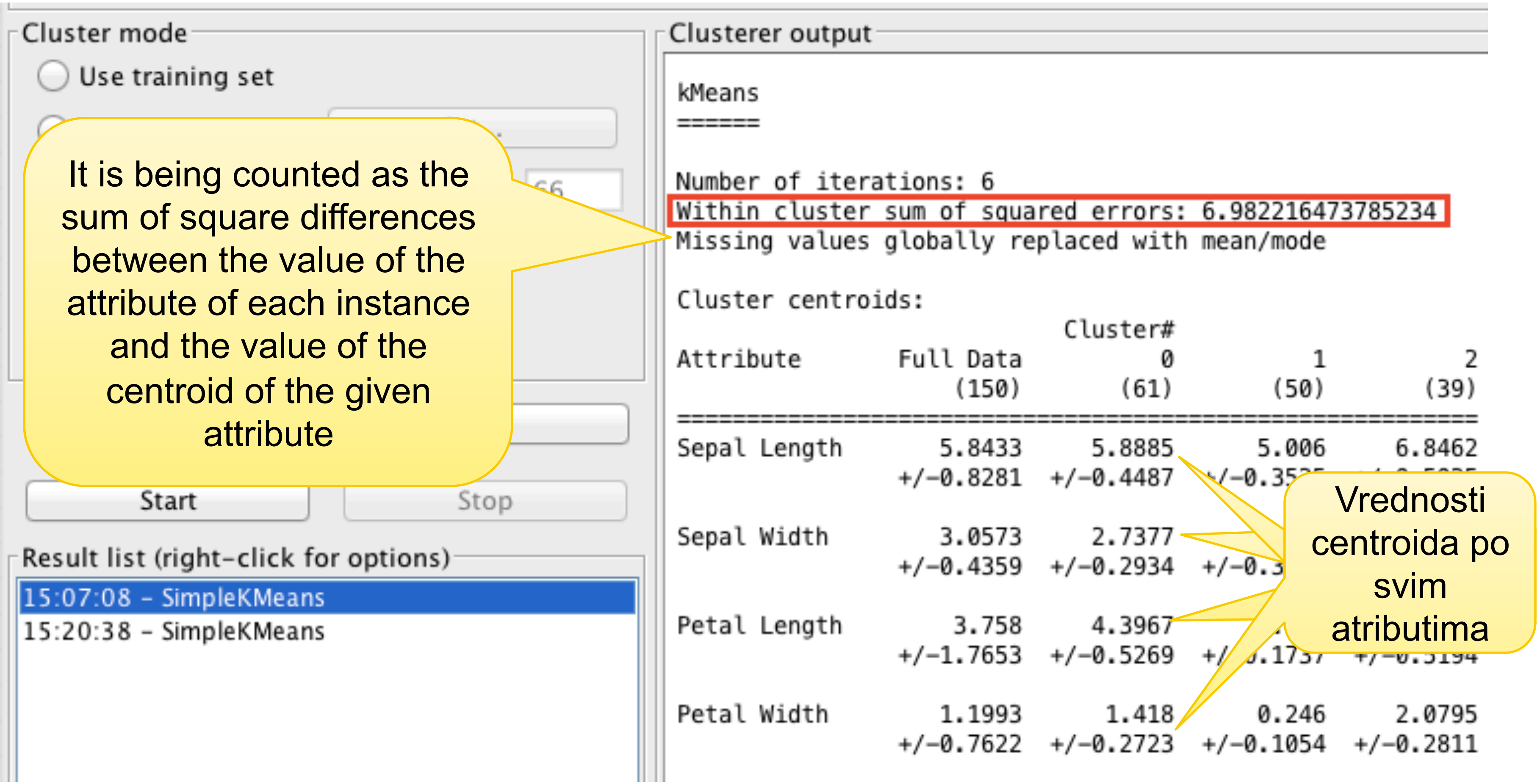
cluster0 cluster1 cluster2

Right click

Visual representation of clusters

Was clustering successful?

Within cluster sum of squared error gives us the assessment of quality

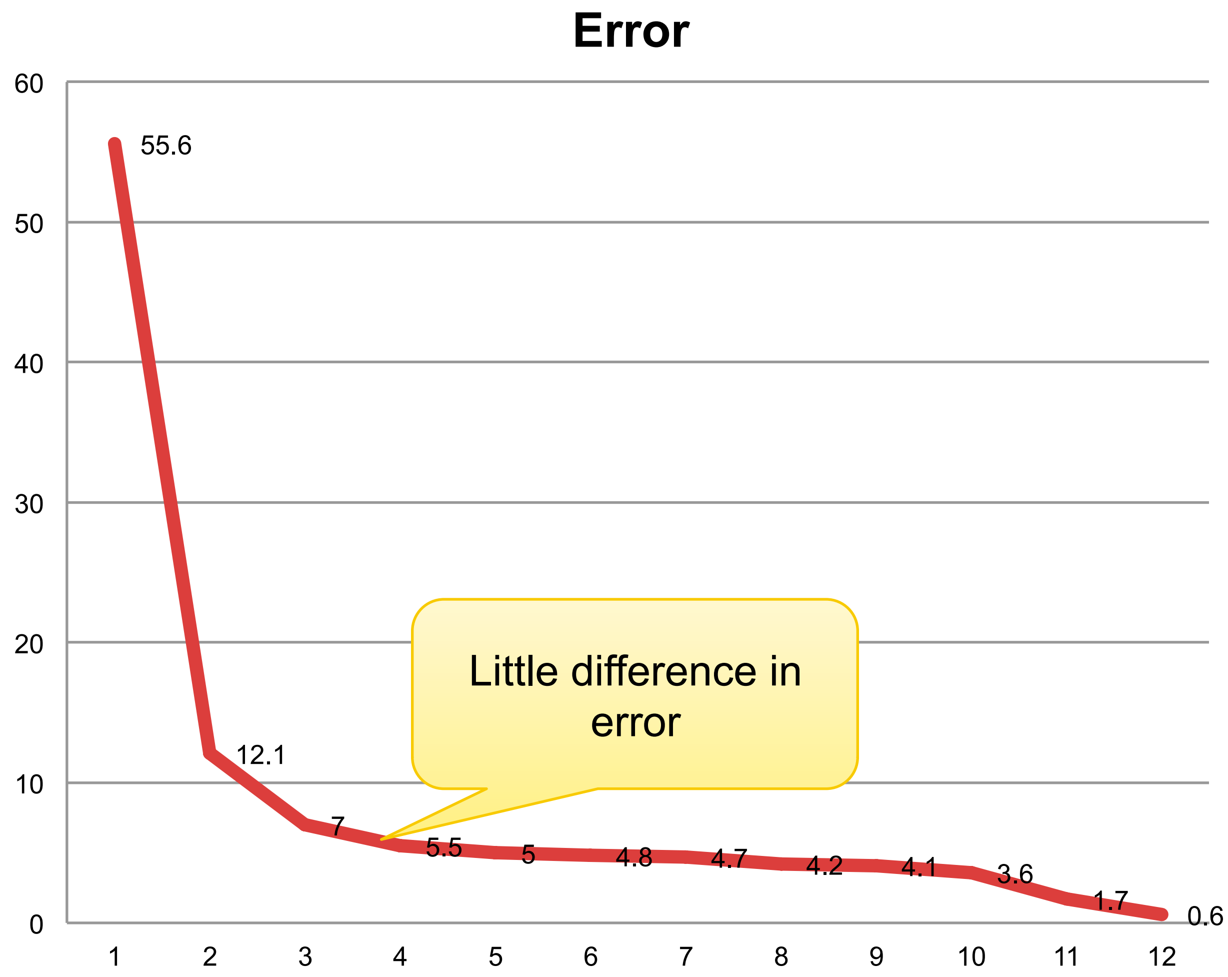


```
kMeans
======

Number of iterations: 6
Within cluster sum of squared errors: 6.982216473785234
Missing values globally replaced with mean/mode

Cluster centroids:
                                Cluster#
Attribute       Full Data          0          1          2
                    (150)       (61)       (50)       (39)
==============================================================
Sepal Length       5.8433     5.8885      5.006     6.8462
                +/-0.8281  +/-0.4487   +/-0.35

Sepal Width        3.0573     2.7377
                +/-0.4359  +/-0.2934   +/-0.3

Petal Length        3.758     4.3967
                +/-1.7653  +/-0.5269   +/ .1737  +/-0.5194

Petal Width        1.1993      1.418      0.246     2.0795
                +/-0.7622  +/-0.2723  +/-0.1054  +/-0.2811
```

It is being counted as the sum of square differences between the value of the attribute of each instance and the value of the centroid of the given attribute

Vrednosti centroida po svim atributima

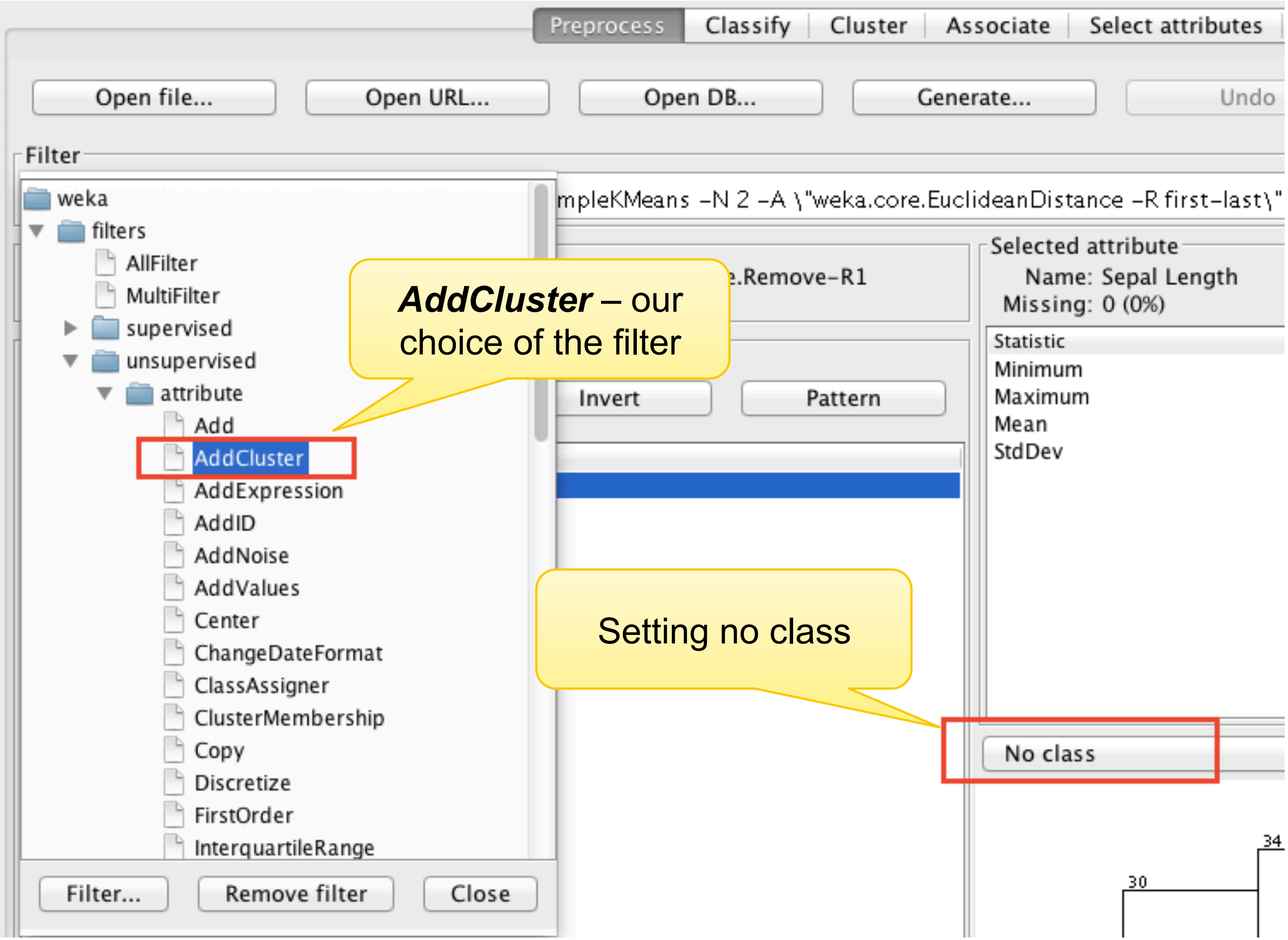
How to figure out the number of clusters?



Little difference in error

Clusters	Errors
1	55.6
2	12.1
3	7.0
4	5.5
5	5.0
6	4.8
7	4.7
8	4.2
9	4.1
10	3.6
20	1.7
50	0.6

Using Clusters for Classification



mpleKMeans -N 2 -A \"weka.core.EuclideanDistance -R first-last\"

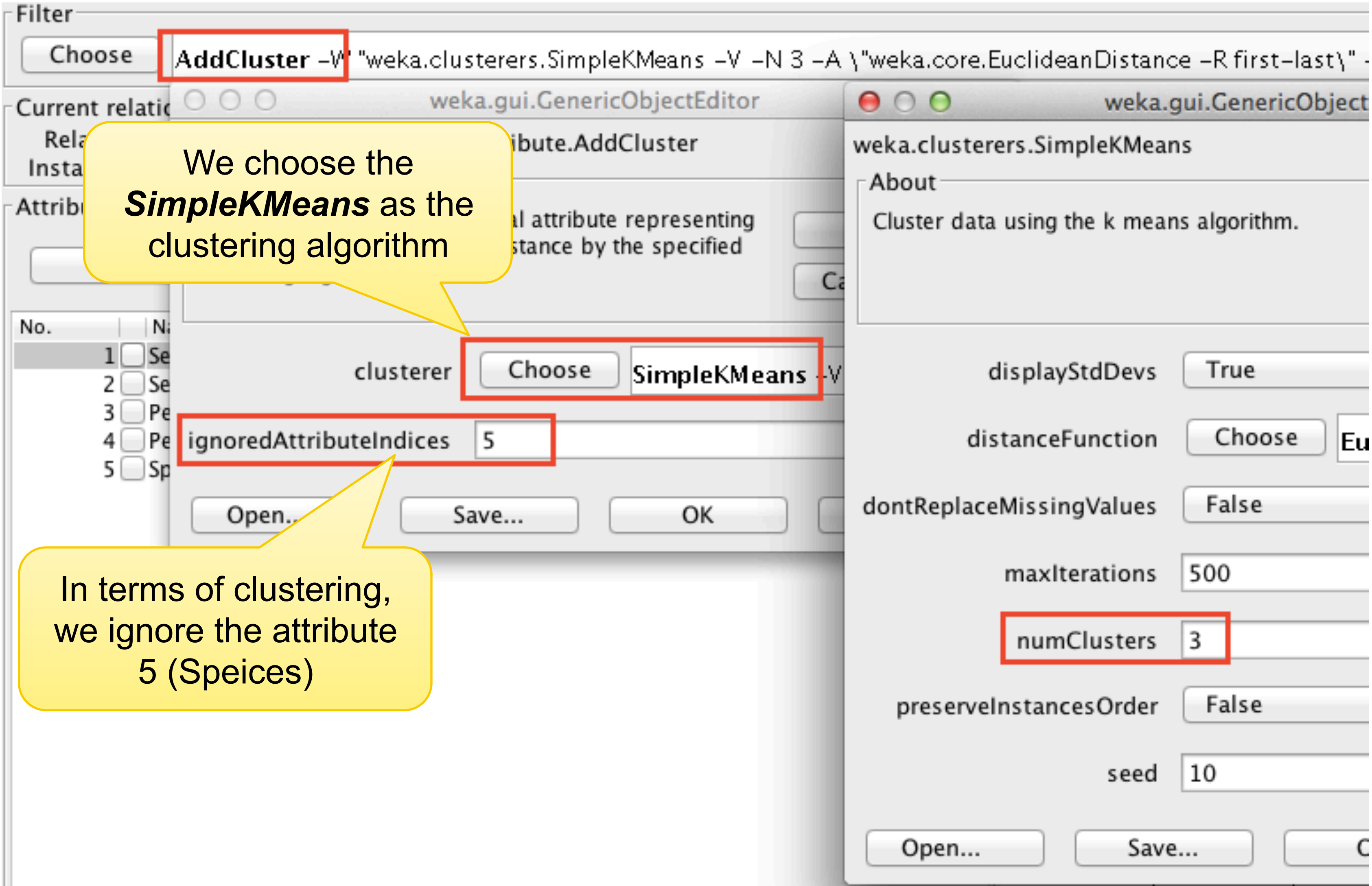
.Remove-R1

Name: Sepal Length
Missing: 0 (0%)

AddCluster – our choice of the filter

Setting no class

Using Clusters for Classification



Filter

Choose AddCluster -W "weka.clusterers.SimpleKMeans -V -N 3 -A \"weka.core.EuclideanDistance -R first-last\" -

weka.gui.GenericObjectEditor

ibute.AddCluster

al attribute representing
stance by the specified

clusterer: Choose SimpleKMeans -V

ignoredAttributeIndices: 5

weka.gui.GenericObject

weka.clusterers.SimpleKMeans

About

Cluster data using the k means algorithm.

displayStdDevs: True

distanceFunction: Choose Eu

dontReplaceMissingValues: False

maxIterations: 500

numClusters: 3

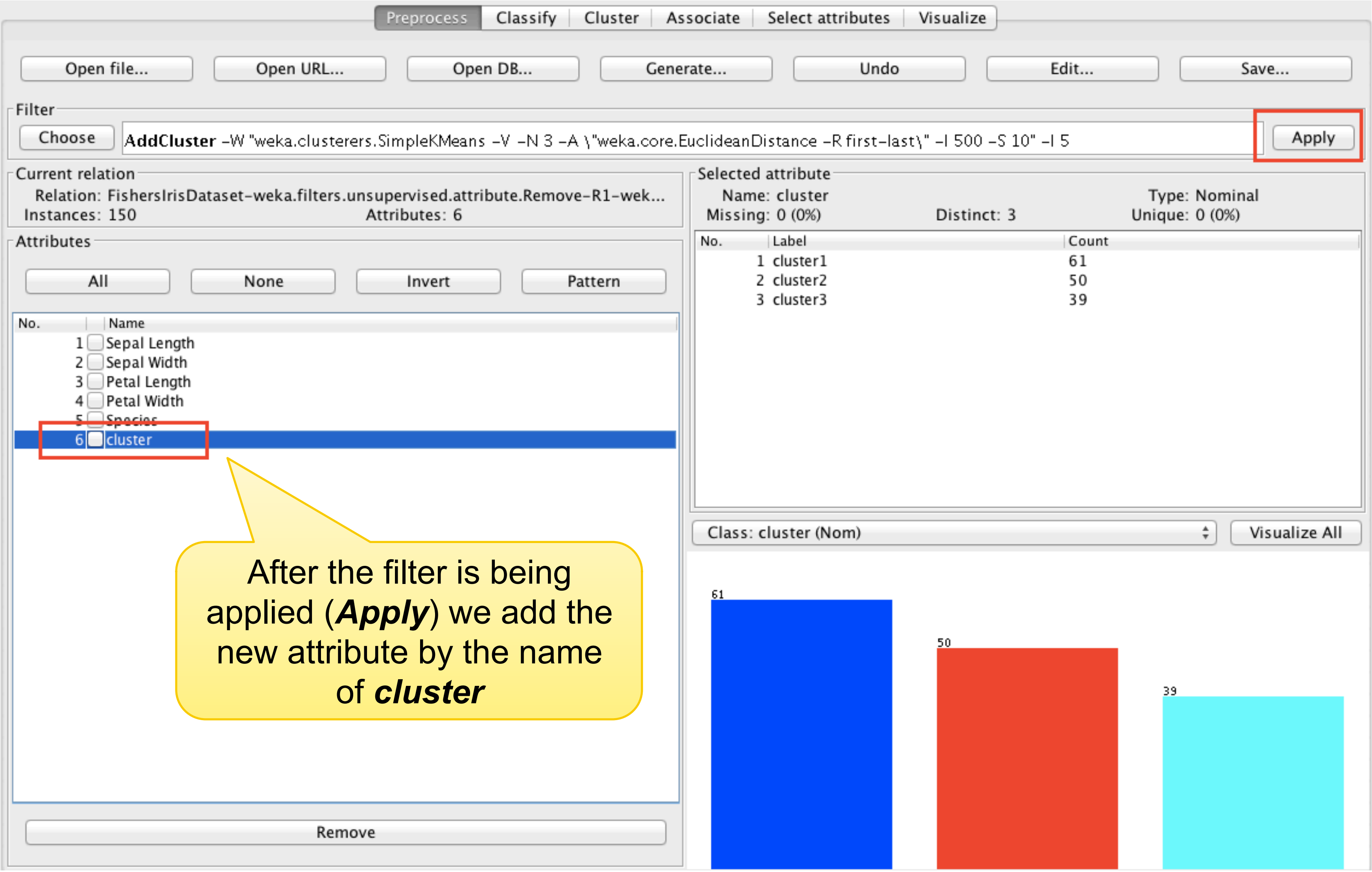
preserveInstancesOrder: False

seed: 10

We choose the ***SimpleKMeans*** as the clustering algorithm

In terms of clustering, we ignore the attribute 5 (Speices)

Using Clusters for Classification



Filter

AddCluster -W "weka.clusterers.SimpleKMeans -V -N 3 -A \"weka.core.EuclideanDistance -R first-last\" -I 500 -S 10" -I 5

Current relation

Relation: FishersIrisDataset-weka.filters.unsupervised.attribute.Remove-R1-wek...
Instances: 150 Attributes: 6

Attributes

No.	Name
1	□ Sepal Length
2	□ Sepal Width
3	□ Petal Length
4	□ Petal Width
5	□ Species
6	□ cluster

Selected attribute

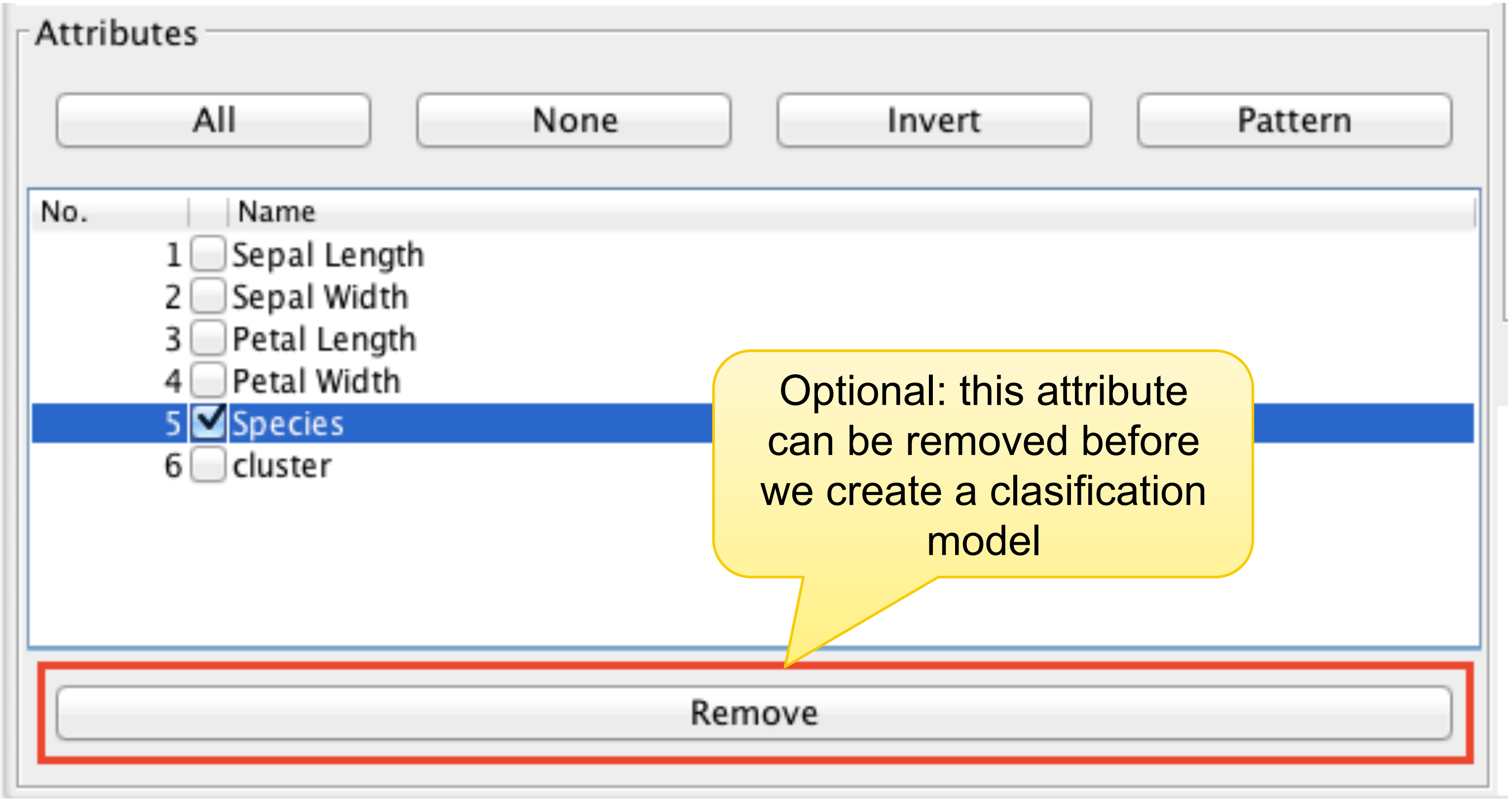
Name: cluster Type: Nominal
Missing: 0 (0%) Distinct: 3 Unique: 0 (0%)

No.	Label	Count
1	cluster1	61
2	cluster2	50
3	cluster3	39

Class: cluster (Nom)

After the filter is being applied (***Apply***) we add the new attribute by the name of ***cluster***

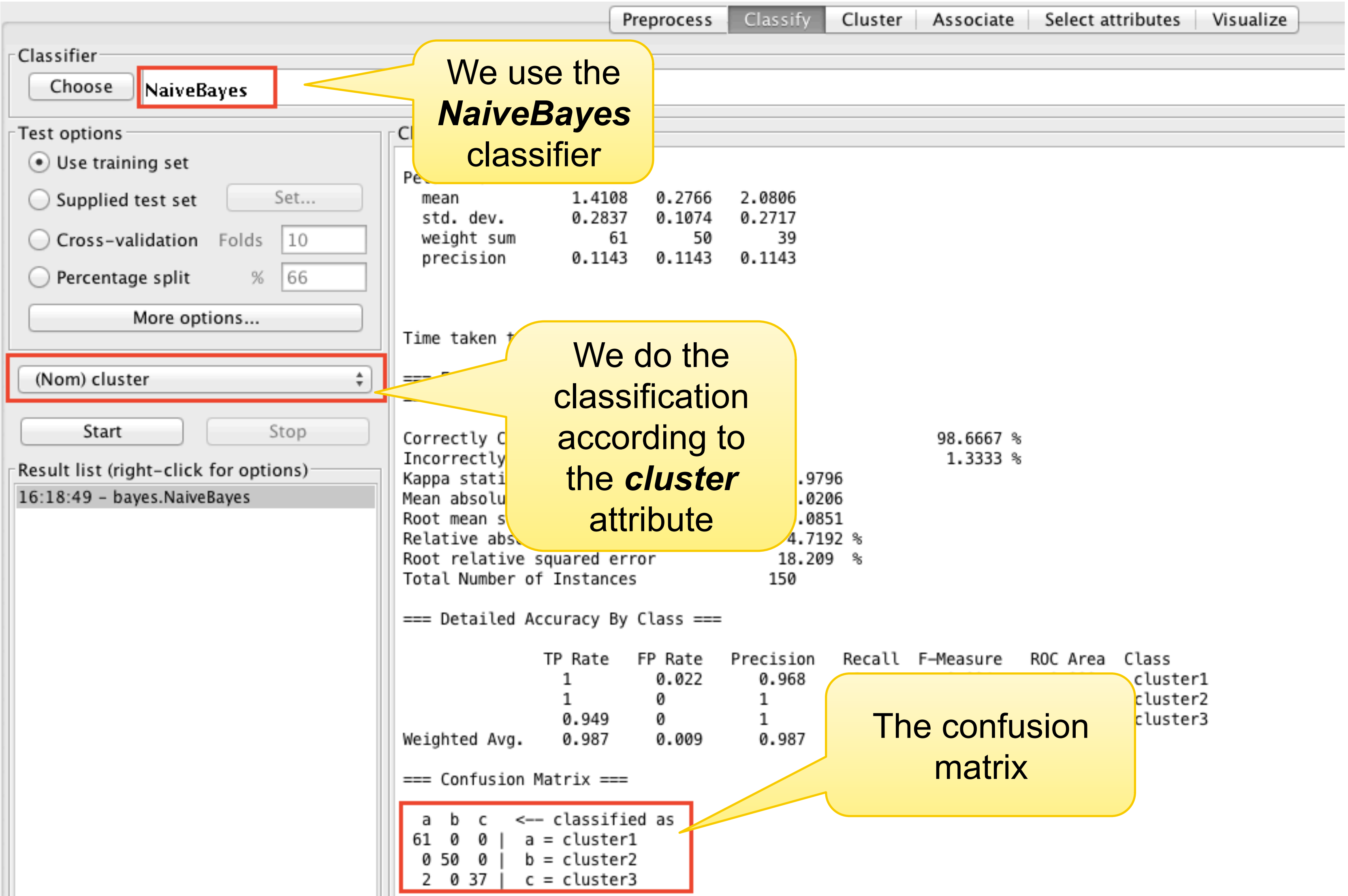
Using Clusters for Classification



No.		Name
1	□	Sepal Length
2	□	Sepal Width
3	□	Petal Length
4	□	Petal Width
5	☑	Species
6	□	cluster

Optional: this attribute can be removed before we create a clasification model

Using Clusters for Classification



Classifier

NaiveBayes

Test options

● Use training set

○ Supplied test set

○ Cross-validation Folds 10

○ Percentage split % 66

(Nom) cluster

Result list (right-click for options)

16:18:49 - bayes.NaiveBayes

```
Cl
Pe
  mean          1.4108   0.2766   2.0806
  std. dev.     0.2837   0.1074   0.2717
  weight sum        61       50       39
  precision     0.1143   0.1143   0.1143



Time taken t

=== 
=

Correctly C                                           98.6667 %
Incorrectly                                            1.3333 %
Kappa stati                      .9796
Mean absolu                      .0206
Root mean s                      .0851
Relative abs                    4.7192 %
Root relative squared error     18.209  %
Total Number of Instances      150

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0.022     0.968                                    cluster1
                 1         0         1                                        cluster2
                 0.949     0         1                                        cluster3
Weighted Avg.    0.987     0.009     0.987

=== Confusion Matrix ===

  a  b  c   <-- classified as
 61  0  0 |  a = cluster1
  0 50  0 |  b = cluster2
  2  0 37 |  c = cluster3
```

We use the ***NaiveBayes*** classifier

We do the classification according to the ***cluster*** attribute

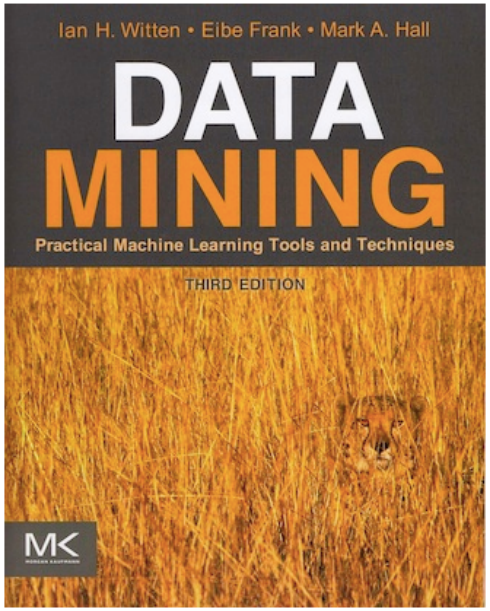
The confusion matrix

Thank you notes

Weka Tutorials and Assignments @ The Technology Forge

- Link: http://www.technologyforge.net/WekaTutorials/

Witten, Ian H., Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques*. Elsevier, 2011.


Ian H. Witten • Eibe Frank • Mark A. Hall
DATA
MINING
Practical Machine Learning Tools and Techniques
THIRD EDITION
MK

A survey for you, to judge us :)

http://goo.gl/cqdp3I

Any questions?

NIKOLA MILIKIĆ
nikola.milikic@fon.bg.ac.rs

UROŠ KRČADINAC
uros@krcadinac.com