Data Analysis & Data Mining with WEKA

Lan Man

Department of Computer Science and Technology

East China Normal University

Synopsis

- About WEKA
- Data Format and Preprocessing
- WEKA Usage (version 3.8)
  - Explorer
  - Experimenter
  - Knowledge Flow
- Other Data Mining Tools
- Summary

About WEKA

- Developed at the machine learning group of CS Department, University of Waikato, New Zealand from 1993
- Waikato Environment for Knowledge Analysis
- Written in Java. Open Source, issued under the GNU General Public License.
- Provides multiple implementations of learning algorithms
  - Classification
  - Clustering
  - Association Rule Mining
  - Attribute Selection (Feature Selection)

  which can be directly applied to the dataset

About WEKA (contd..)

- Main features
  - Comprehensive set of data pre-processing tools, learning algorithms and evaluation methods
  - Graphical user interfaces (incl. data visualization)
  - Environment for comparing learning algorithms
- Usage
  - Apply a learning method to a dataset & analyze the result
  - Use a learned model to make predictions on new instances
  - Apply different learners to a dataset & compare results
- WEKA downloaded from http://www.cs.waikato.ac.nz/~ml/weka/
- This lesson is based on the latest snapshot of WEKA 3.8

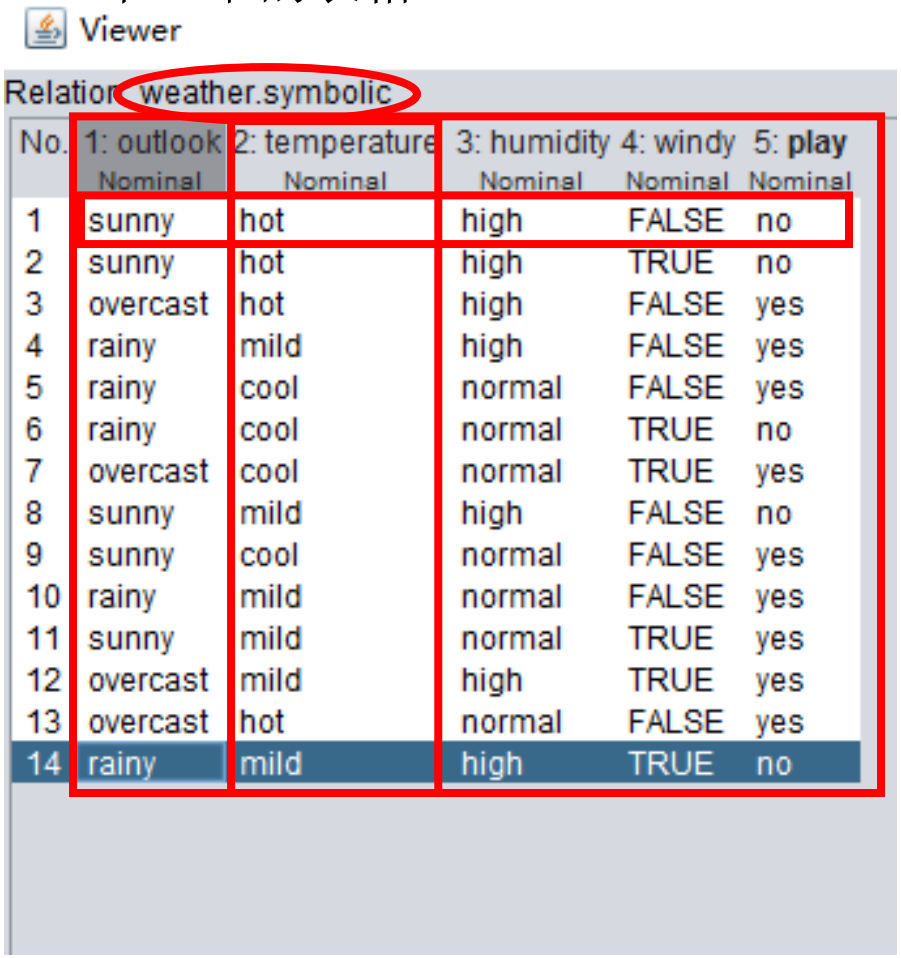
Launching Weka 3.8

Machine Learning with WEKA 3.8

- About WEKA
- Data Format and Preprocessing
- WEKA Usage (version 3.8)
  - Explorer
  - Experimenter
  - Knowledge Flow
- Other Data Mining Tools
- Summary

WEKA 数据格式

- 跟很多电子表格或数据分析软件一样，WEKA所处理的数据集是如图的一个二维的表格



- WEKA中的术语：表格里的一行称作一个实例（Instance）；
- 一列称作一个属性（Attrbute），相当于统计学中的一个变量，或者数据库中的一个字段。
- 这样一个表格，或者叫数据集(dataset)，呈现属性之间的一种关系(Relation)。
- 14个实例，5个属性，关系名称为“weather.symbolic”

WEKA 数据格式

```
% ARFF file for the weather data
@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
% 14 instances

sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```

- WEKA存储数据的格式是ARFF (**A**ttribute-**R**elation **F**ile **F**ormat)文件，这是一种ASCII文本文件。上页图中所示的二维表格存储在如左图示的ARFF文件中。这也就是WEKA自带的“weather.arff”文件，在WEKA安装目录的“data”子目录下可以找到。
- 识别ARFF文件的重要依据是分行，因此不能在这种文件里随意的断行。空行（或全是空格的行）将被忽略。
- 以“%”开始的行是注释，WEKA将忽略这些行。

WEKA 数据格式

```
% ARFF file for the weather data
@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
% 14 instances

sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```

- 头信息（Head information），包括了对关系的声明和对属性的声明
- 关系名称在ARFF文件的第一个有效行来定义；如果这个字符串包含空格，它必须加上引号（指英文标点的单引号或双引号）
- 属性声明用一列以“@attribute”开头的语句表示。数据集中的每一个属性都有它对应的“@attribute”语句，来定义它的属性名称和数据类型。
- 数据信息（Data information），即数据集中给出的数据。从“@data”标记开始，后面的就是数据信息了。

WEKA 数据格式

```
% ARFF file for the weather data
@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
% 14 instances

sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```

- 属性声明语句的顺序很重要。首先表明该项属性在数据部分的位置。其次，最后一个声明的属性被称作class属性，在分类或回归任务中，它是默认的目标变量。
- <attribute-name>是必须以字母开头的字符串。和关系名称一样，如果这个字符串包含空格，它必须加上引号。
- WEKA支持的<datatype>有四种：
  numeric: 数值型
  nominal-specification: 分类(nominal)
  string: 字符串型
  date [<date-format>]: 日期和时间型
  默认的字符串是ISO-8601所给的日期时间组合格式 “yyyy-MM-ddTHH:mm:ss”

WEKA 数据格式

```
% ARFF file for the weather data
@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
% 14 instances

sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```

- 数据信息中“@data”标记独占一行，剩下的是各个实例的数据。 每个实例占一行。实例的各属性值用逗号“,”隔开。如果某个属性的值是缺失值（missing value），用问号“?”表示，不能省略。例如：
  sunny,hot,high,FALSE,no
  ?,hot,high,?,yes
- 字符串属性和分类属性的值是区分大小写。若含有空格，必须被引号括起来。如：
  AG5, ‘Encyclopedias and dictionaries‘
  ,’Twentieth century'
- 日期属性的值必须与属性声明中给定的相一致。例如：
  @ATTRIBUTE timestamp DATE "yyyy-MM-dd HH:mm:ss"
  @DATA
  "2001-04-03 12:12:12"

WEKA 数据格式-稀疏数据

- 有的数据集中含有大量的0值（比如超市购物篮数据），这时用稀疏格式的数据存贮更加省空间
- 稀疏格式是针对数据信息中某个实例的表示而言，不需要修改ARFF文件的其它部分。如下：

@data
0, X, 0, Y, 0, 0, “class A”
0, 0, W, 0, 0, 0, “class B”

用稀疏格式表达:

@data
{1 X, 3 Y, 6 “class A”}
{2 W, 6 “class B”}

每个实例用花括号括起来。实例中每一个非0的属性值用<index> <空格><value>表示。<index>是属性的序号，从0开始计；<value>是属性值。属性值之间仍用逗号隔开。这里每个实例的数值必须按属性的顺序来写，如 {1 X, 3 Y, 6 “class A”}，不能写成{3 Y, 1 X, 6 “class A“}

- 注意在稀疏格式中没有注明的属性值不是缺失值，而是0值。若要表示缺失值必须显式的用问号（？）表示出来。

数据预处理

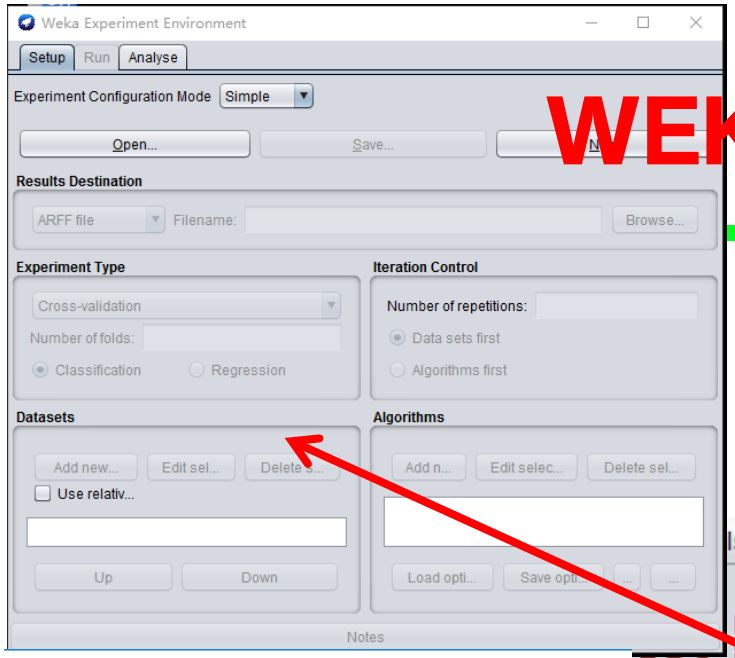
- 使用WEKA作数据挖掘，面临的第一个问题往往是我们的数据不是ARFF格式的。WEKA提供了对CSV文件的支持，而这种格式是被很多其他软件所支持的。此外，WEKA还提供了通过JDBC访问数据库的功能。
- 根据ARFF文件格式，我们也可以先用自己的预处理方法将原有的文件转化成ARFF文件，毕竟后者才是WEKA支持得最好的文件格式。
- 面对一个ARFF文件，我们仍有一些预处理要做，才能进行挖掘任务。

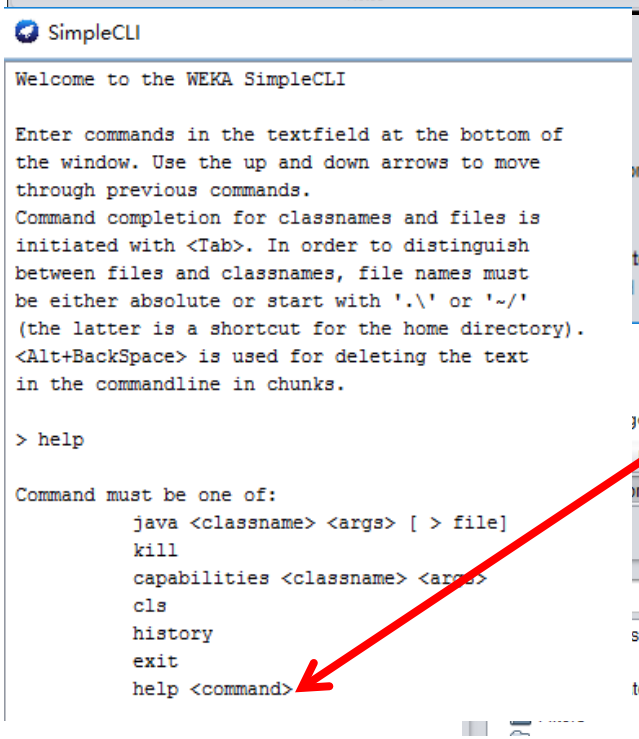
Machine Learning with WEKA 3.8

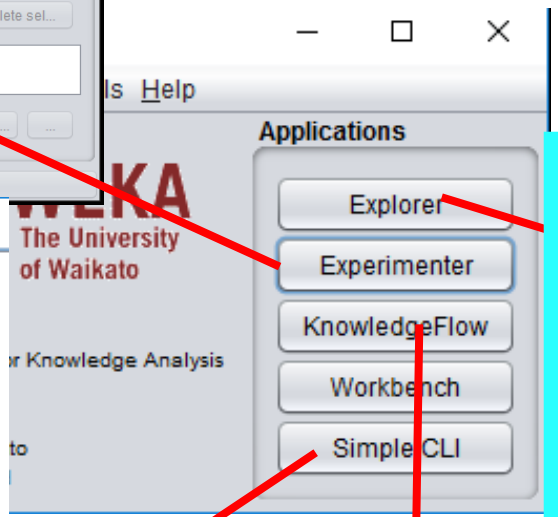
- About WEKA
- Data Format and Preprocessing
- WEKA Usage (version 3.8)
  - Explorer
  - Experimenter
  - Knowledge Flow
- Other Data Mining Tools
- Summary

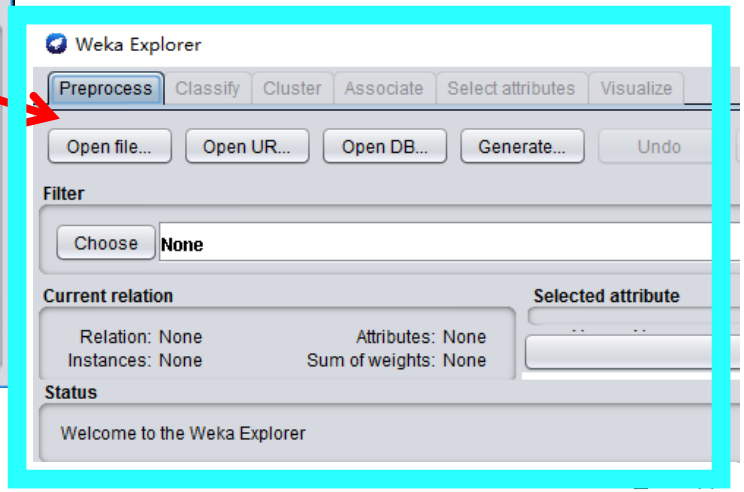
WEKA窗口

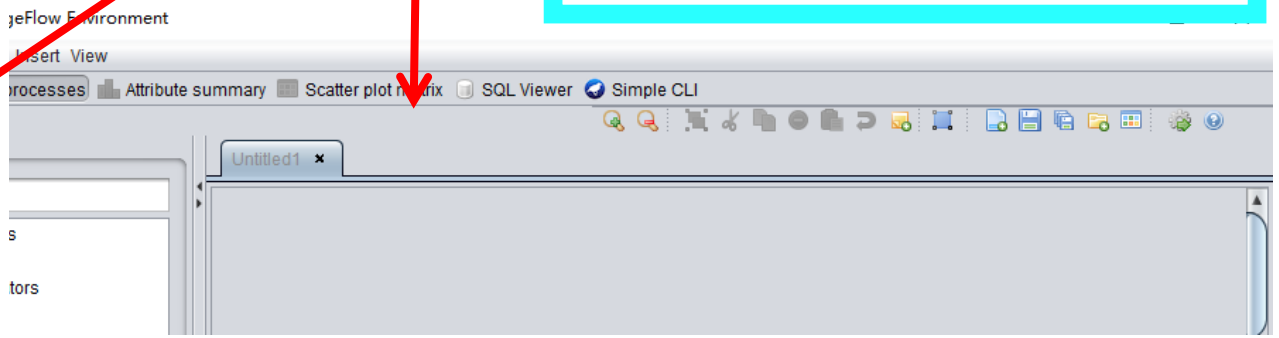
- Applications
  - Explorer
  - Experimenter
  - Knowledge Flow
  - Command line Interface (SimpleCLI)











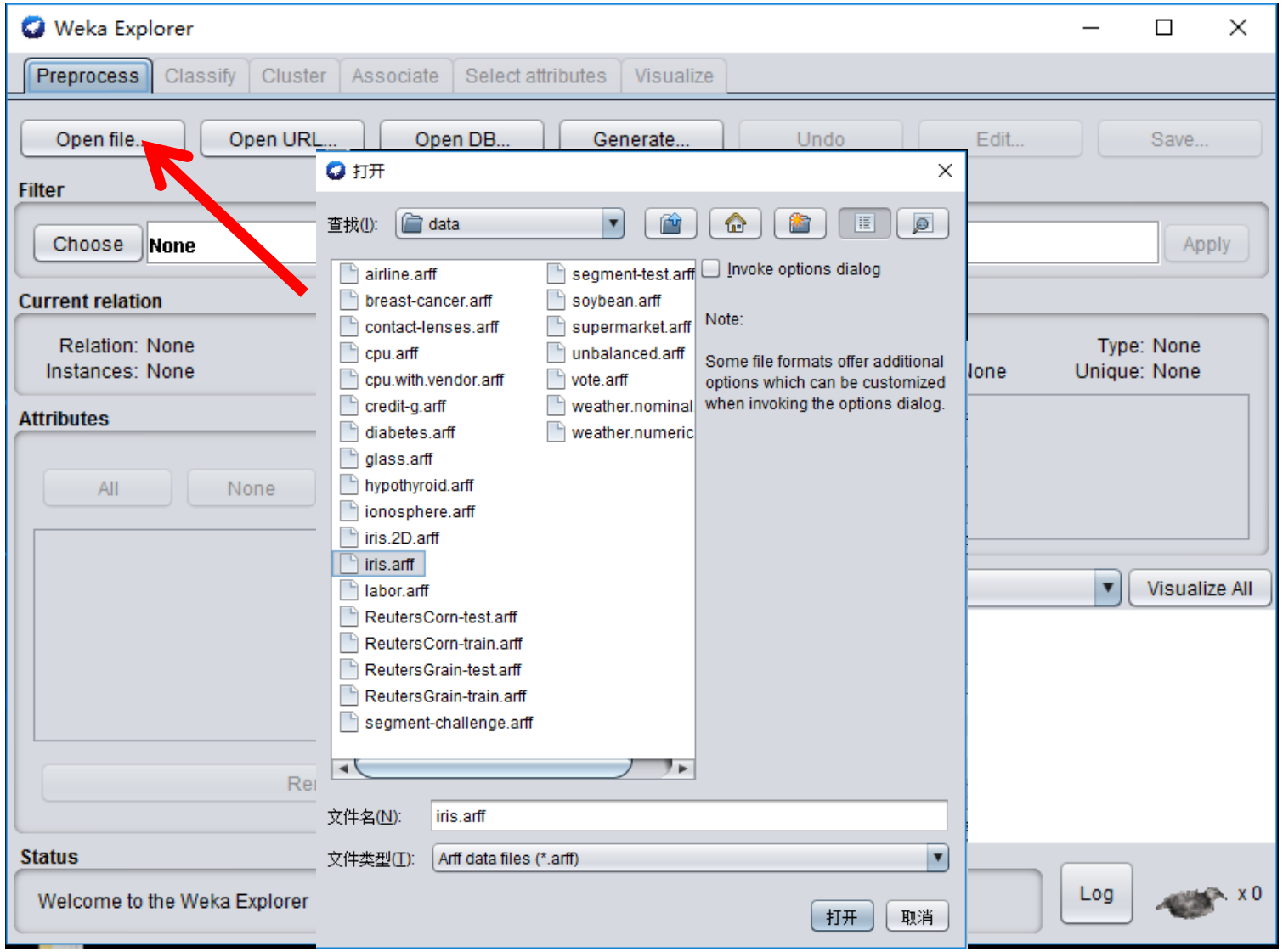
Roadmap: WEKA Usage

- WEKA Usage (version 3.8)
  - Explorer
    - Data preprocessing
    - Classification
    - Clustering
    - Association rule
    - Attribute selection (Feature selection)
    - Data visualization
  - Experimenter
  - Knowledge Flow

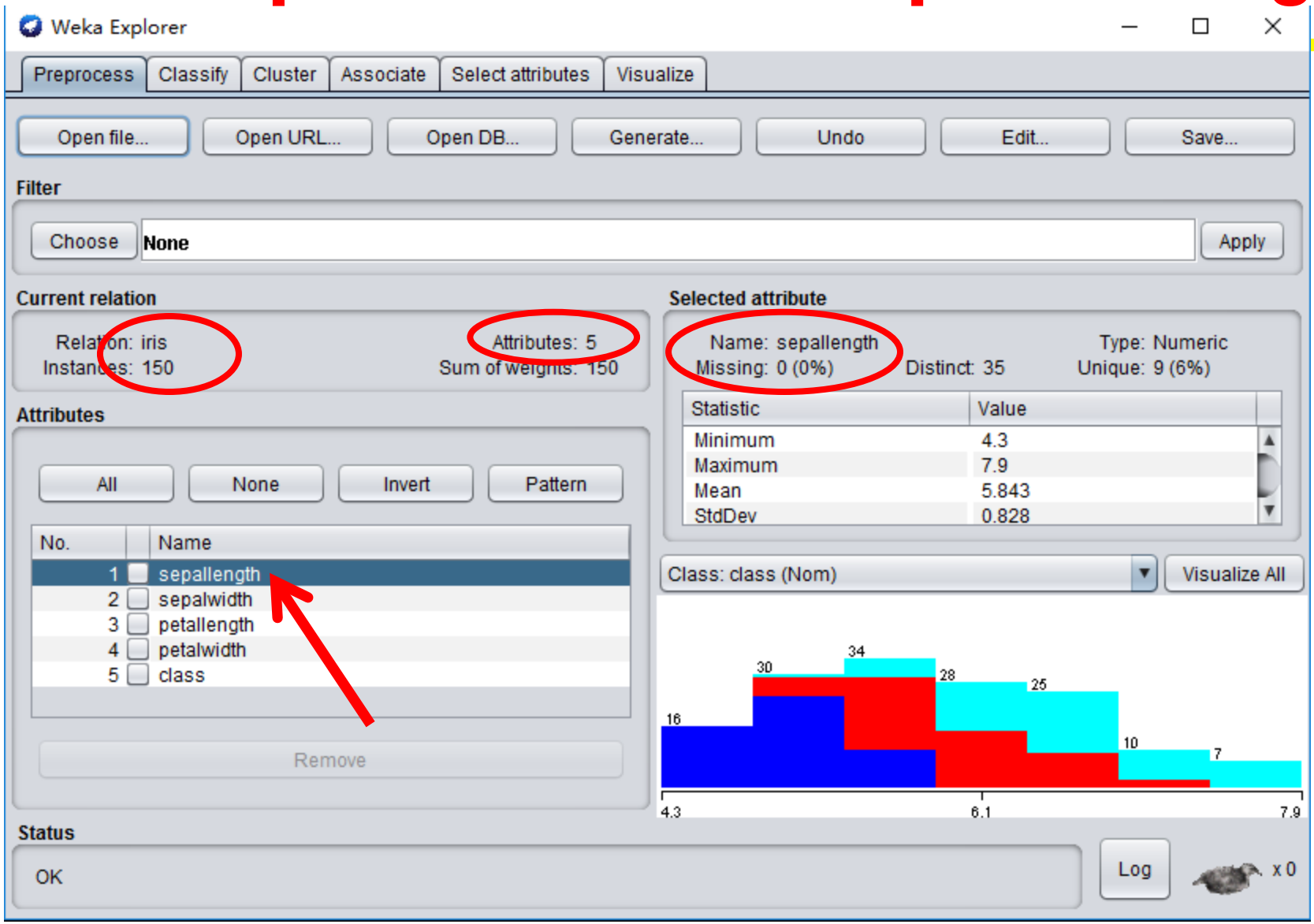
Explorer: Data Pre-processing

- Data can be imported from a file in various formats: ARFF, CSV, C4.5, binary
- Data can also be read from a URL or from an SQL database (using JDBC)
- Pre-processing tools in WEKA are called “filters”
- WEKA contains filters for:
  - Discretization, normalization, resampling, attribute selection, transforming and combining attributes, ...

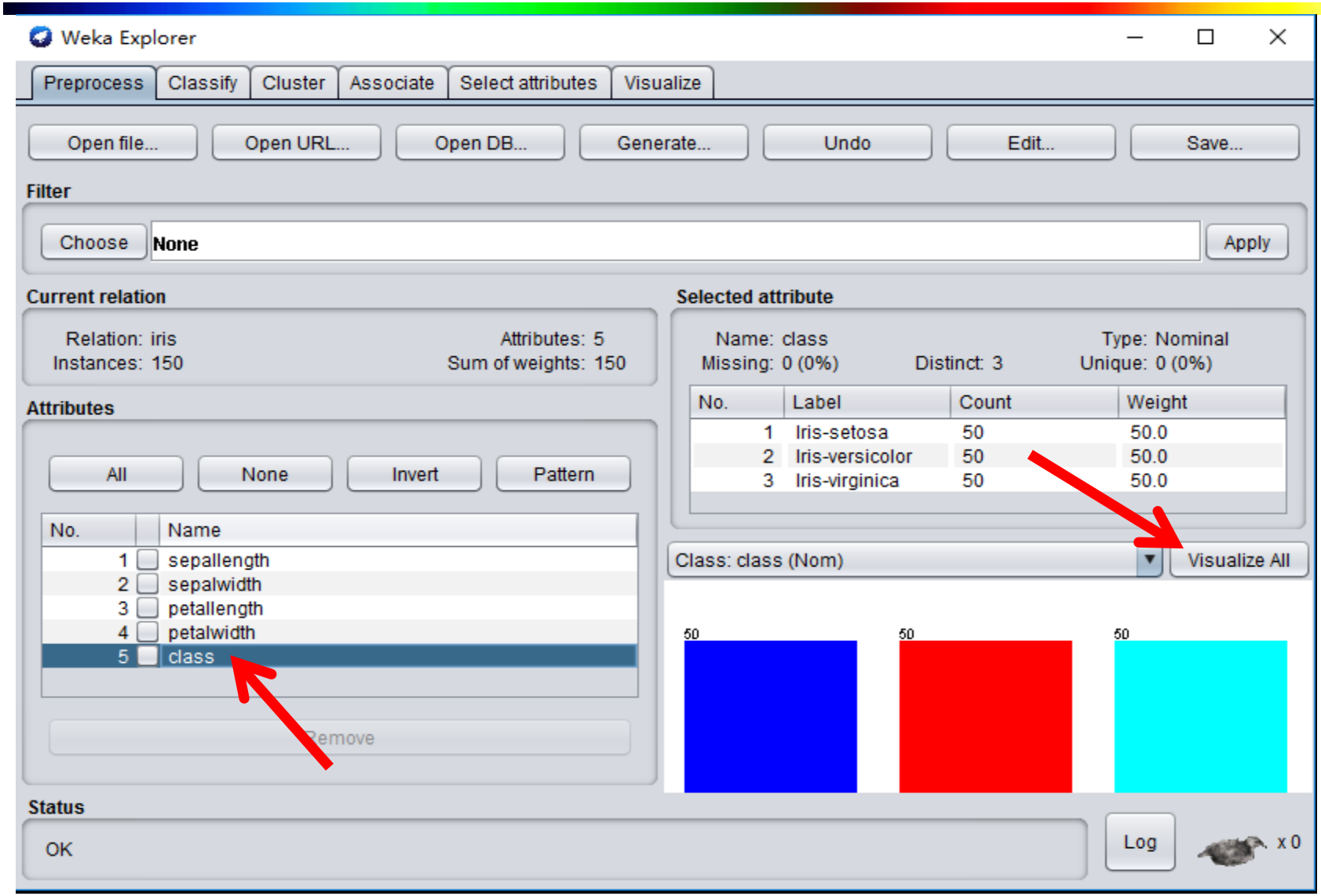
Explorer: Data Pre-processing



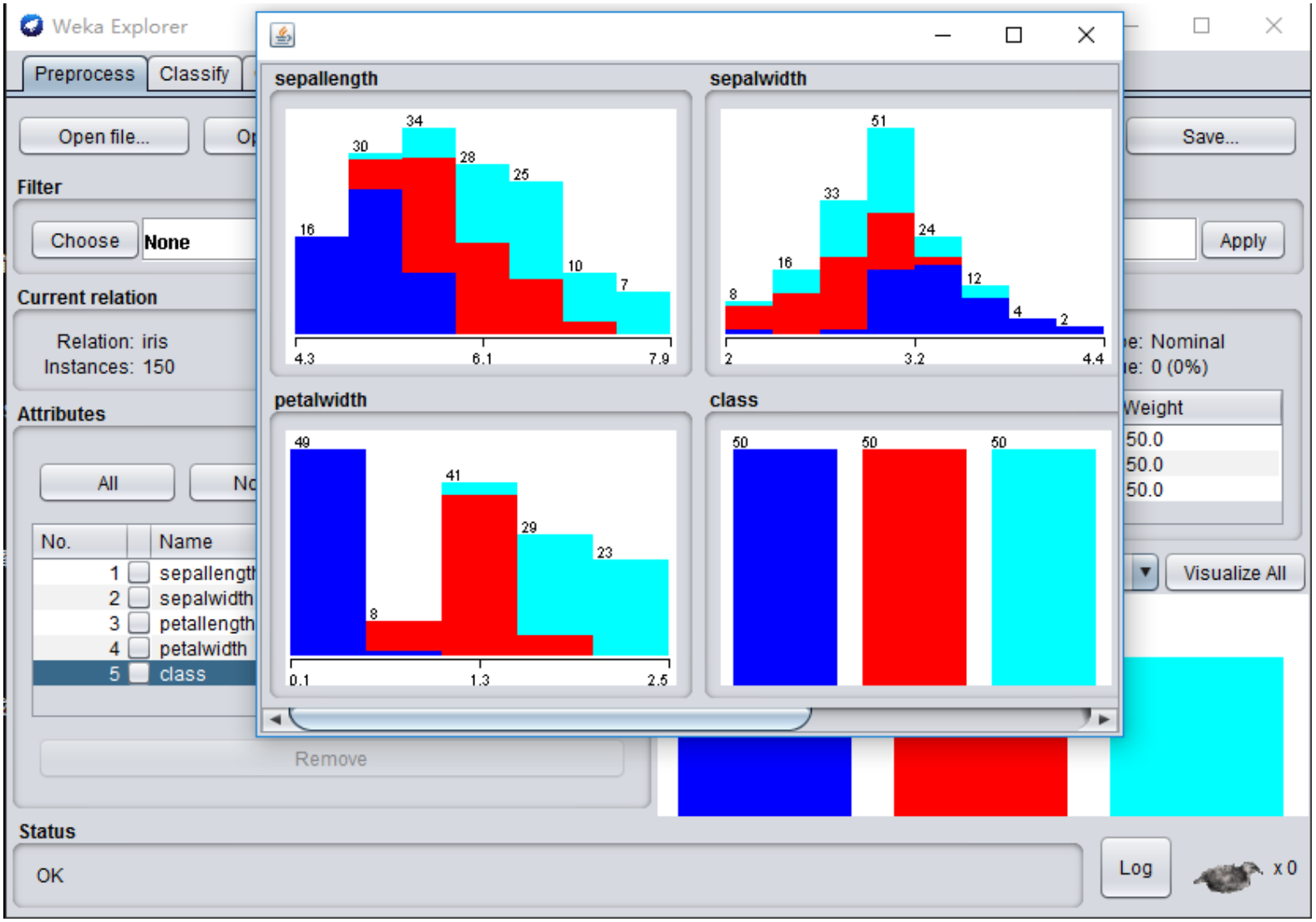
Explorer: Data Pre-processing



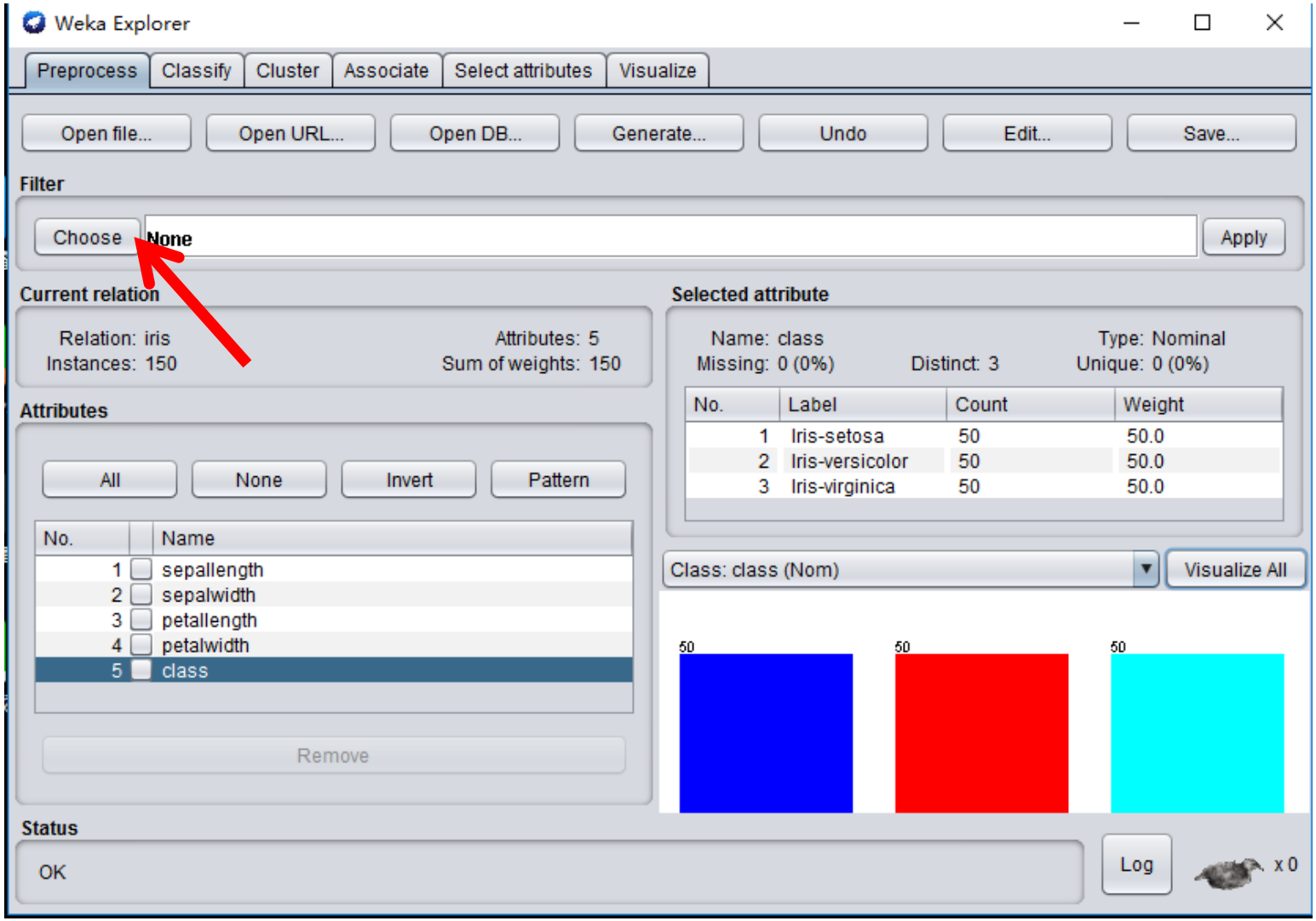
Explorer: Data Pre-processing



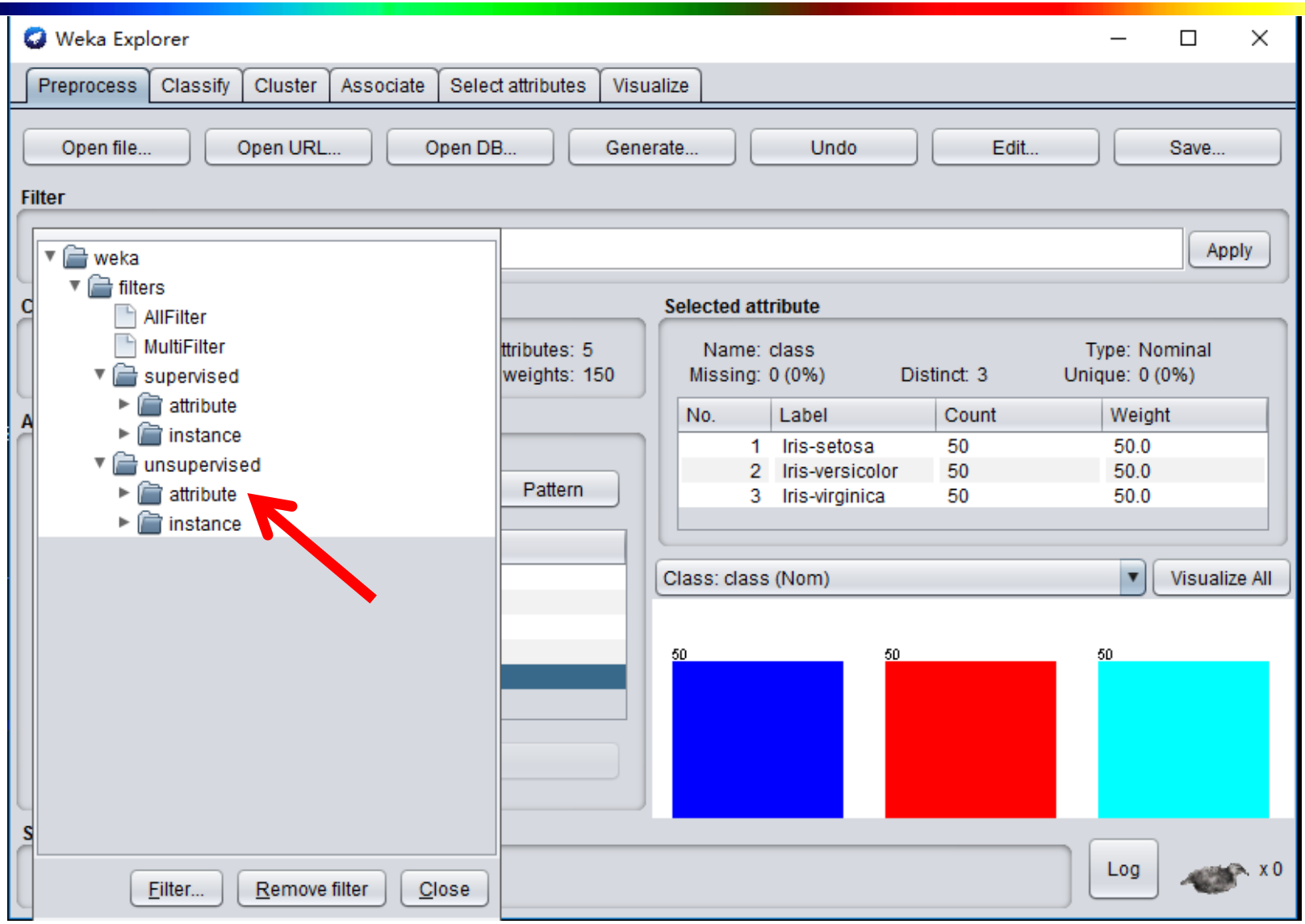
Explorer: Data Pre-processing



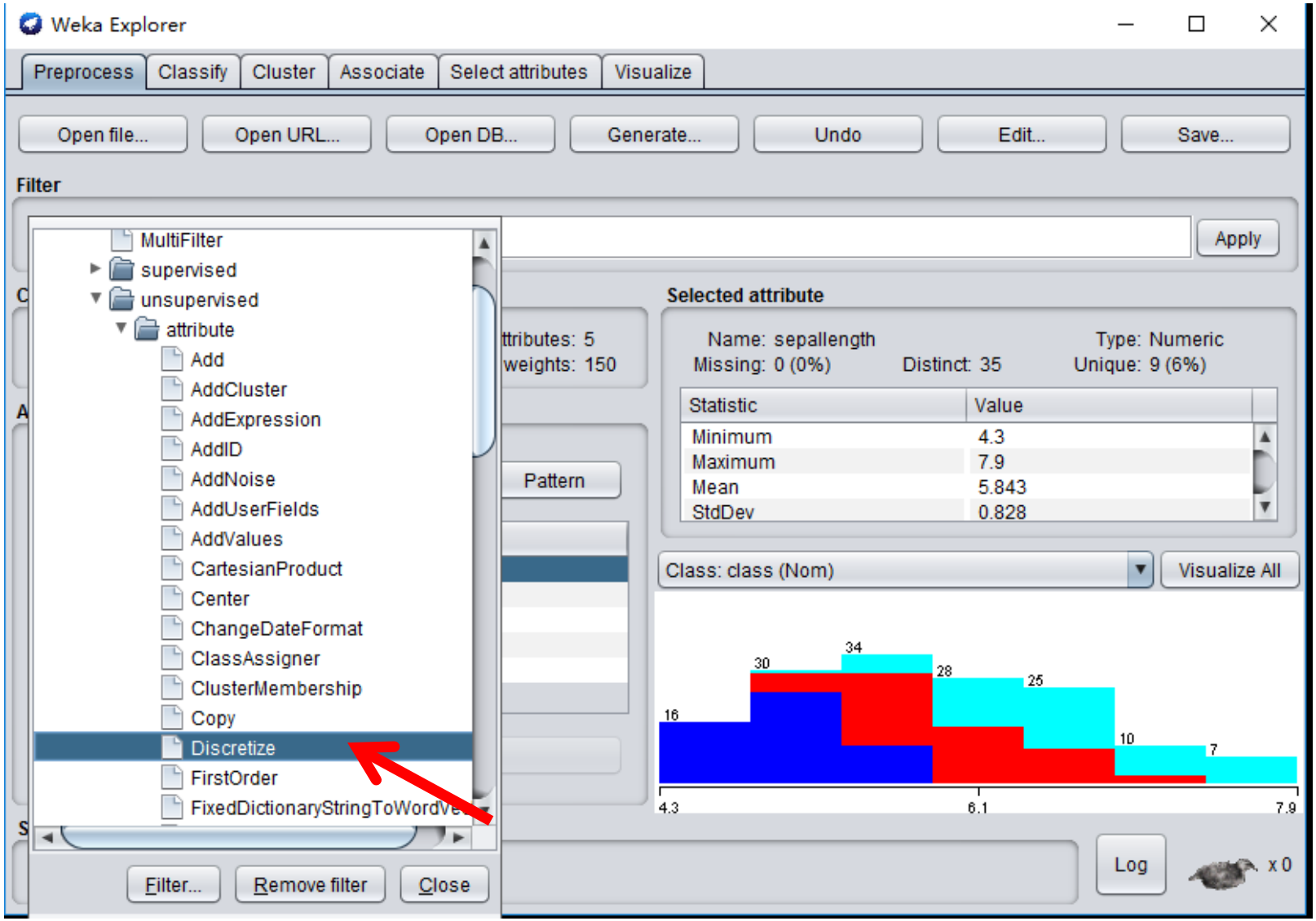
Explorer: Data Pre-processing



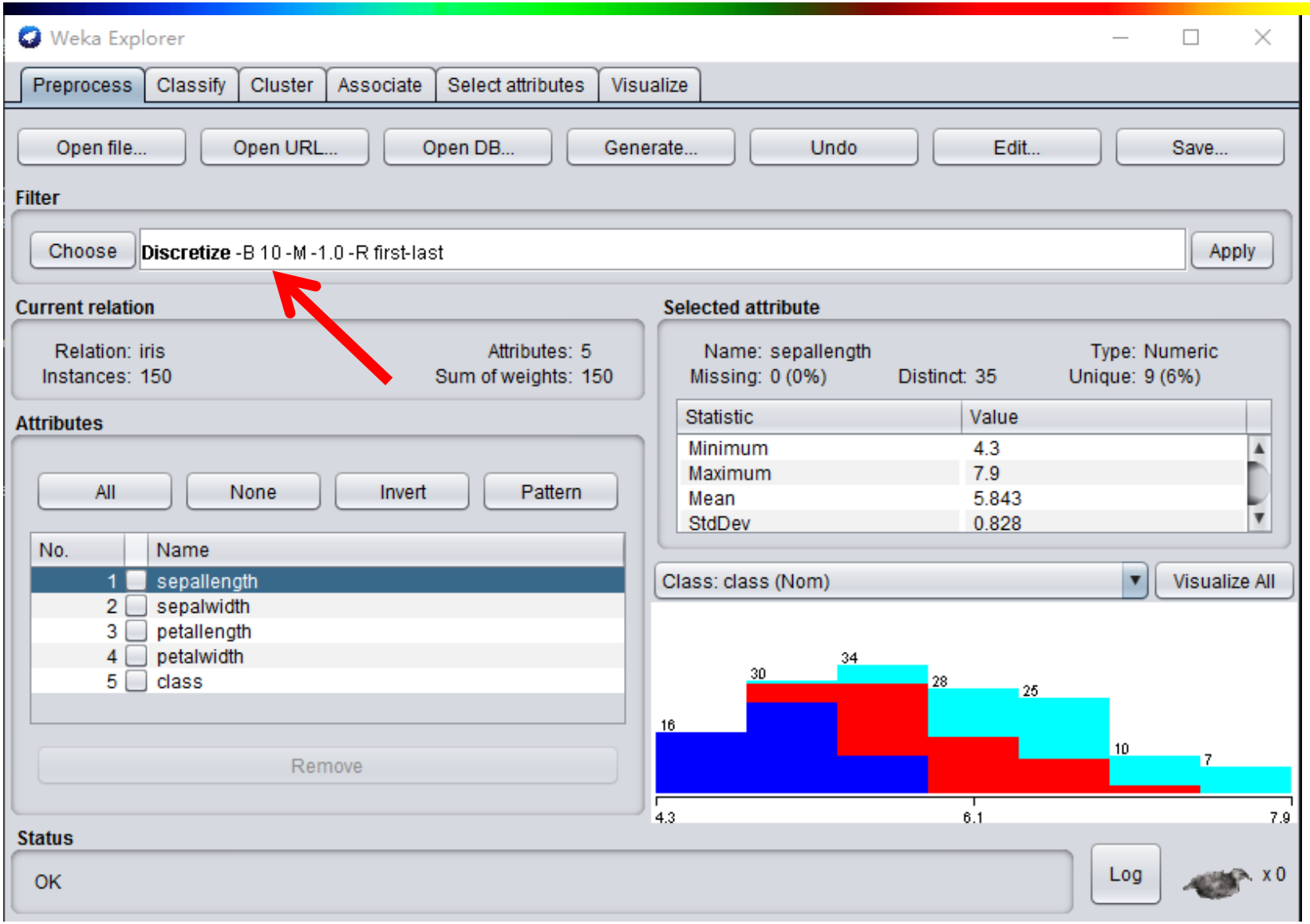
Explorer: Data Pre-processing



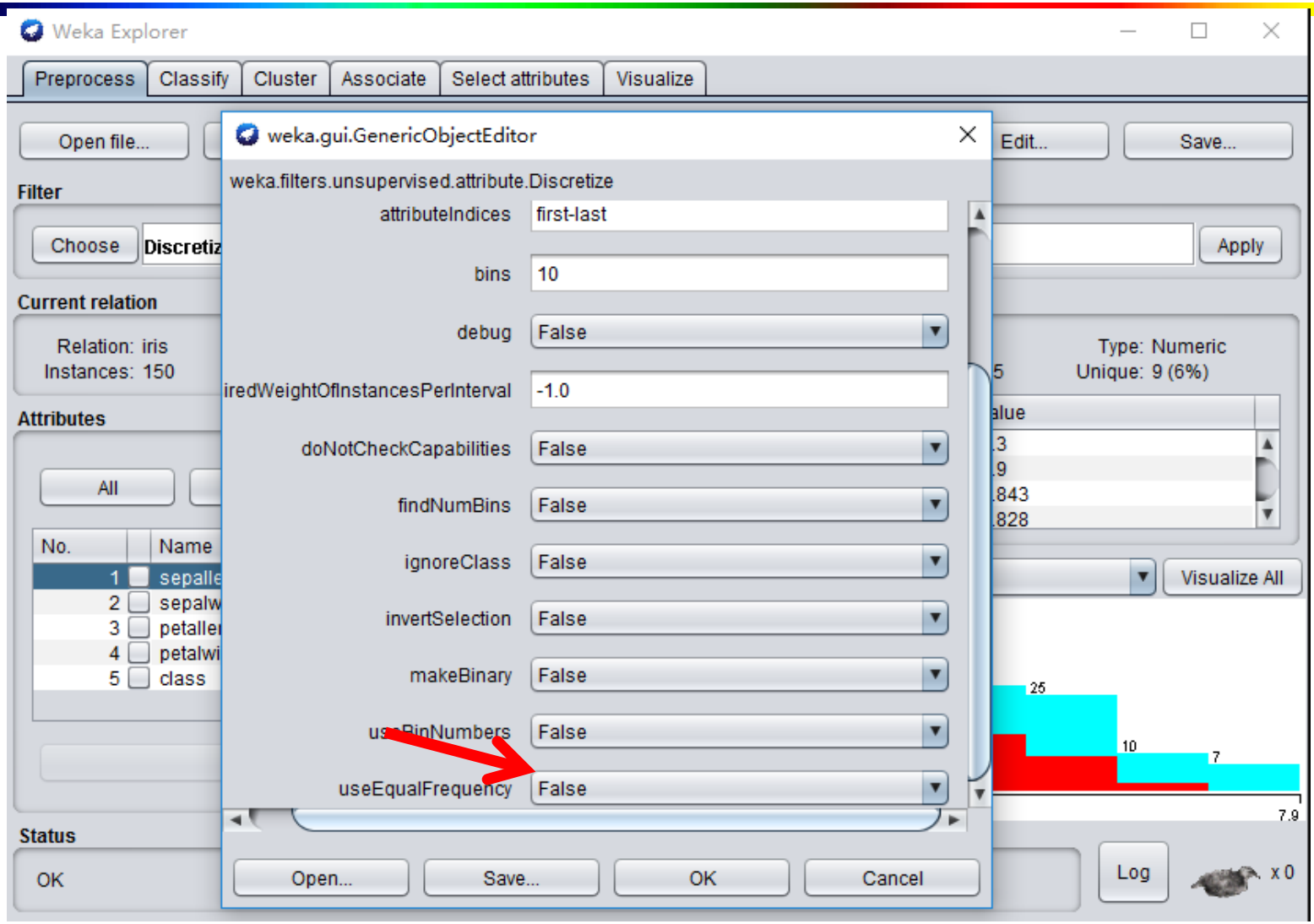
Explorer: Data Pre-processing



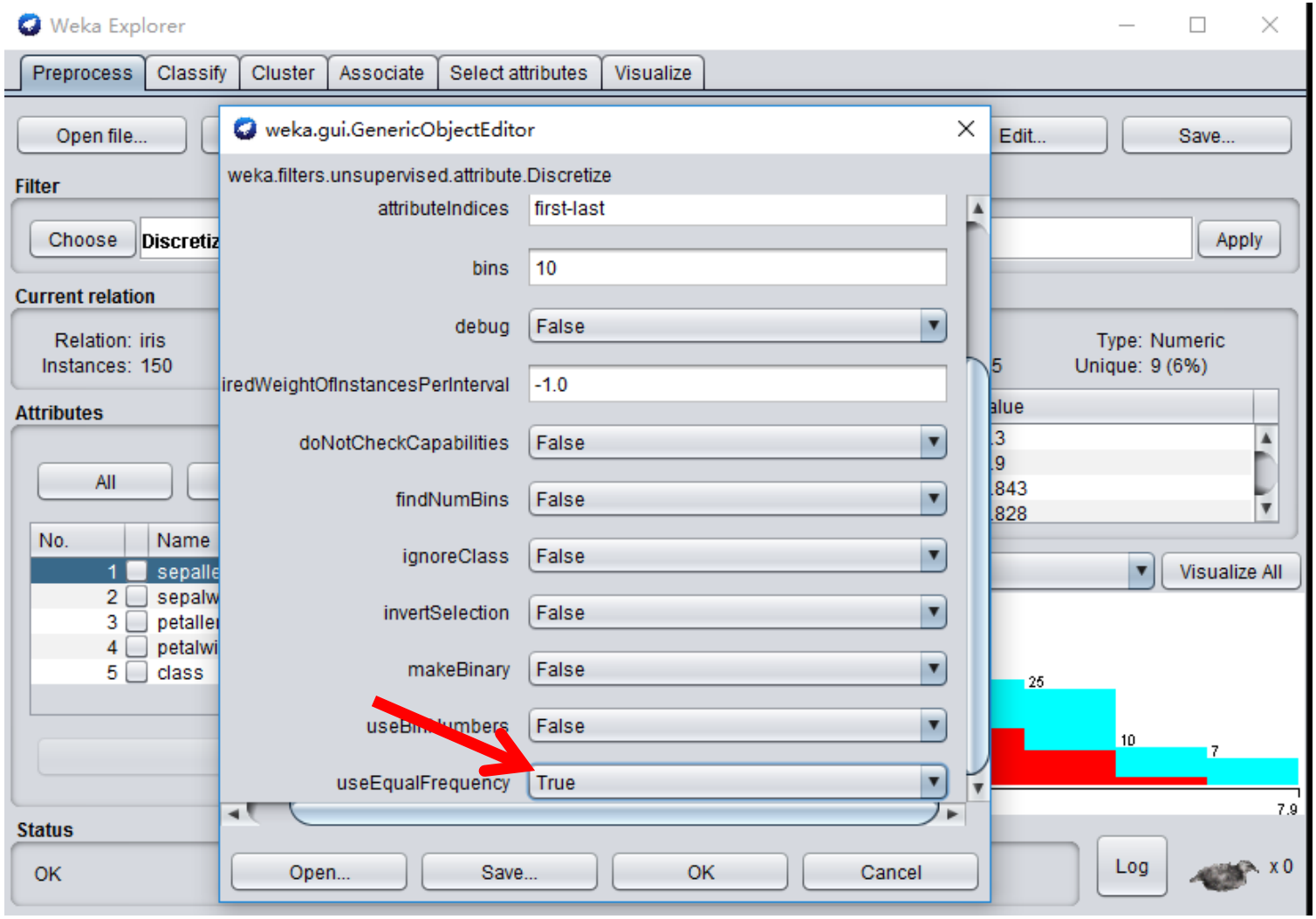
Explorer: Data Pre-processing



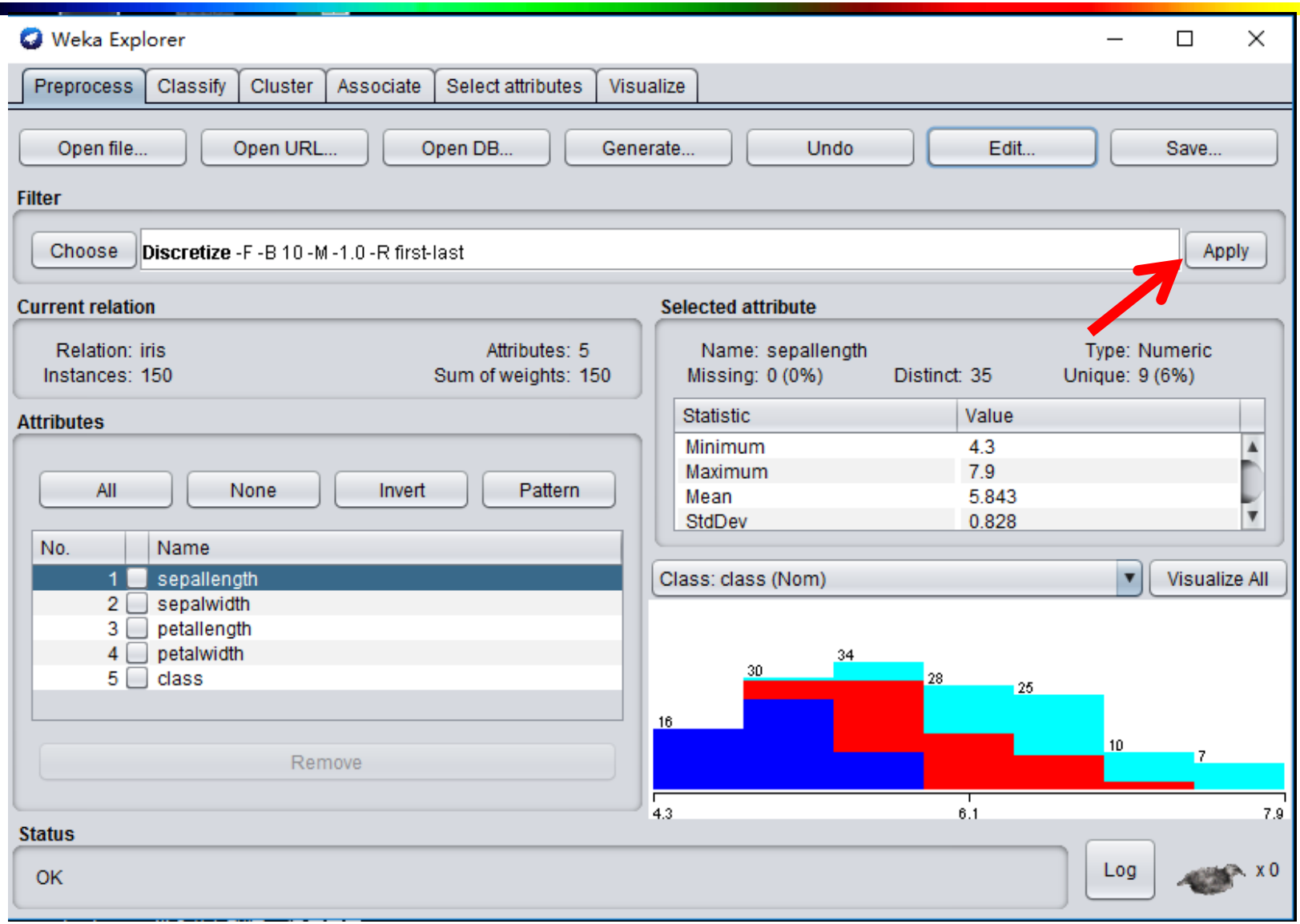
Explorer: Data Pre-processing



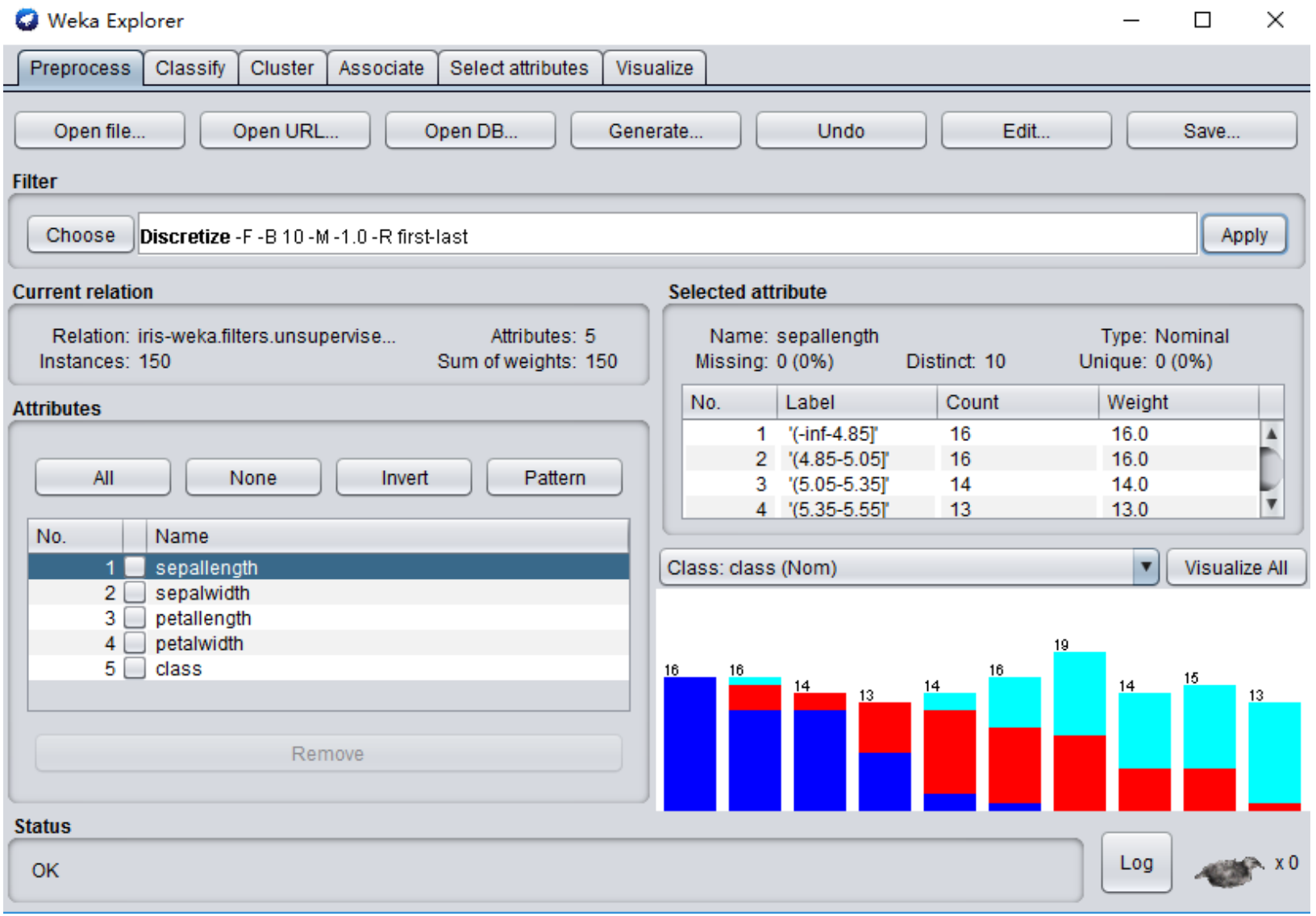
Explorer: Data Pre-processing



Explorer: Data Pre-processing



Explorer: Data Pre-processing



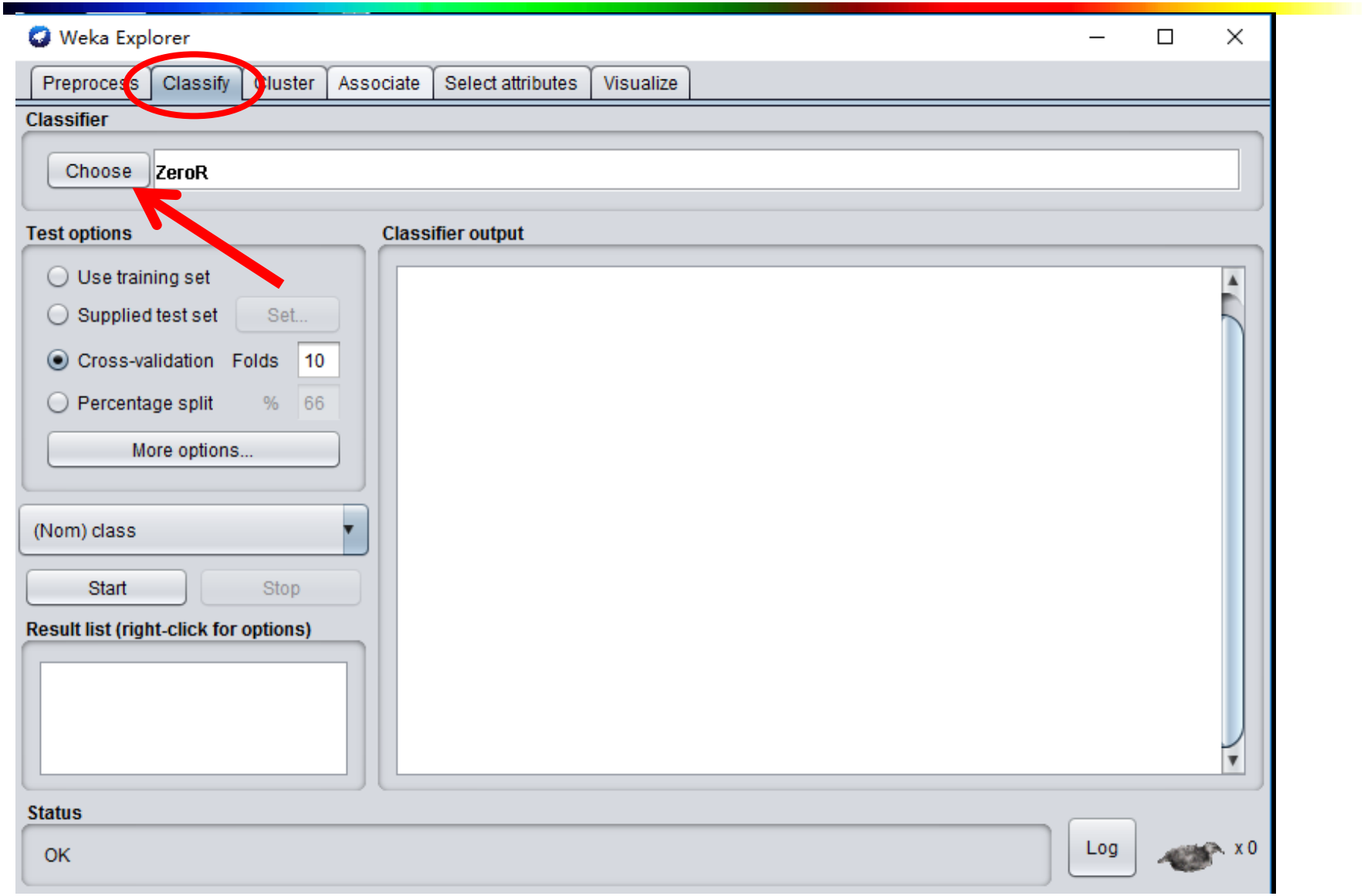
Roadmap: WEKA Usage

- WEKA Usage (version 3.8)
  - Explorer
    - Data preprocessing
    - Classification
    - Clustering
    - Association rule
    - Attribute selection (Feature selection)
    - Data visualization
  - Experimenter
  - Knowledge Flow

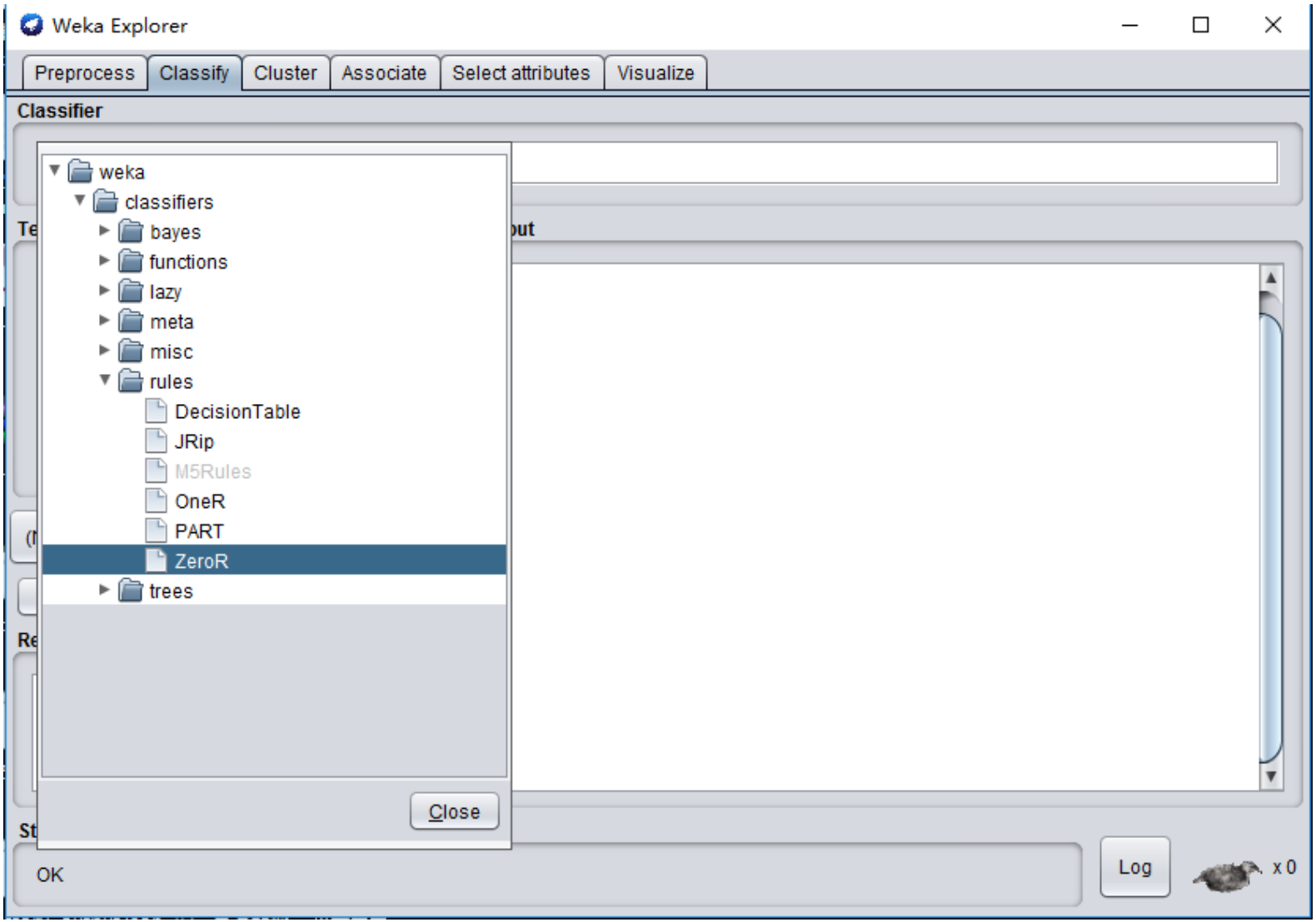
Explorer: Building Classifiers

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

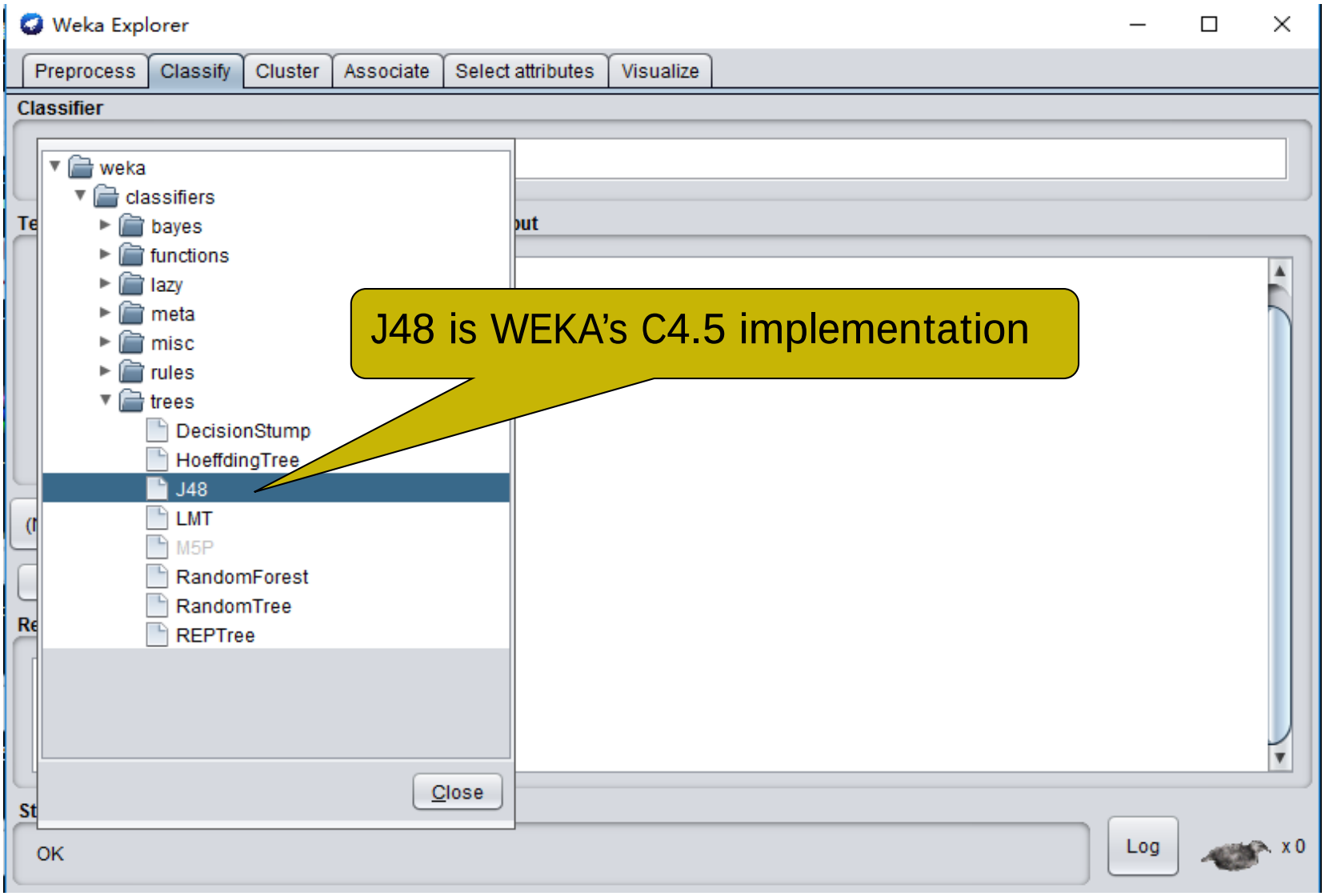
Explorer: Classification



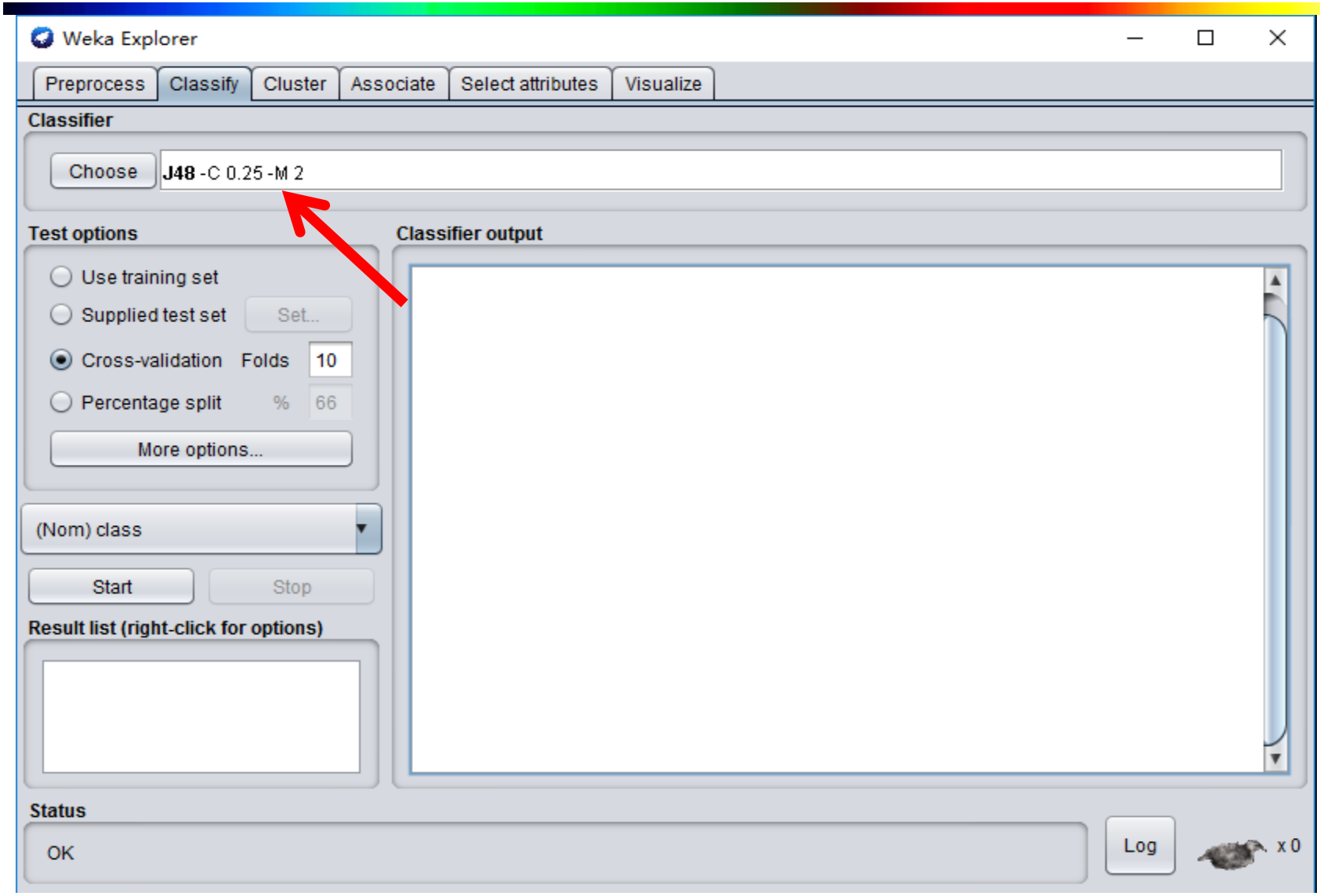
Explorer: Classification



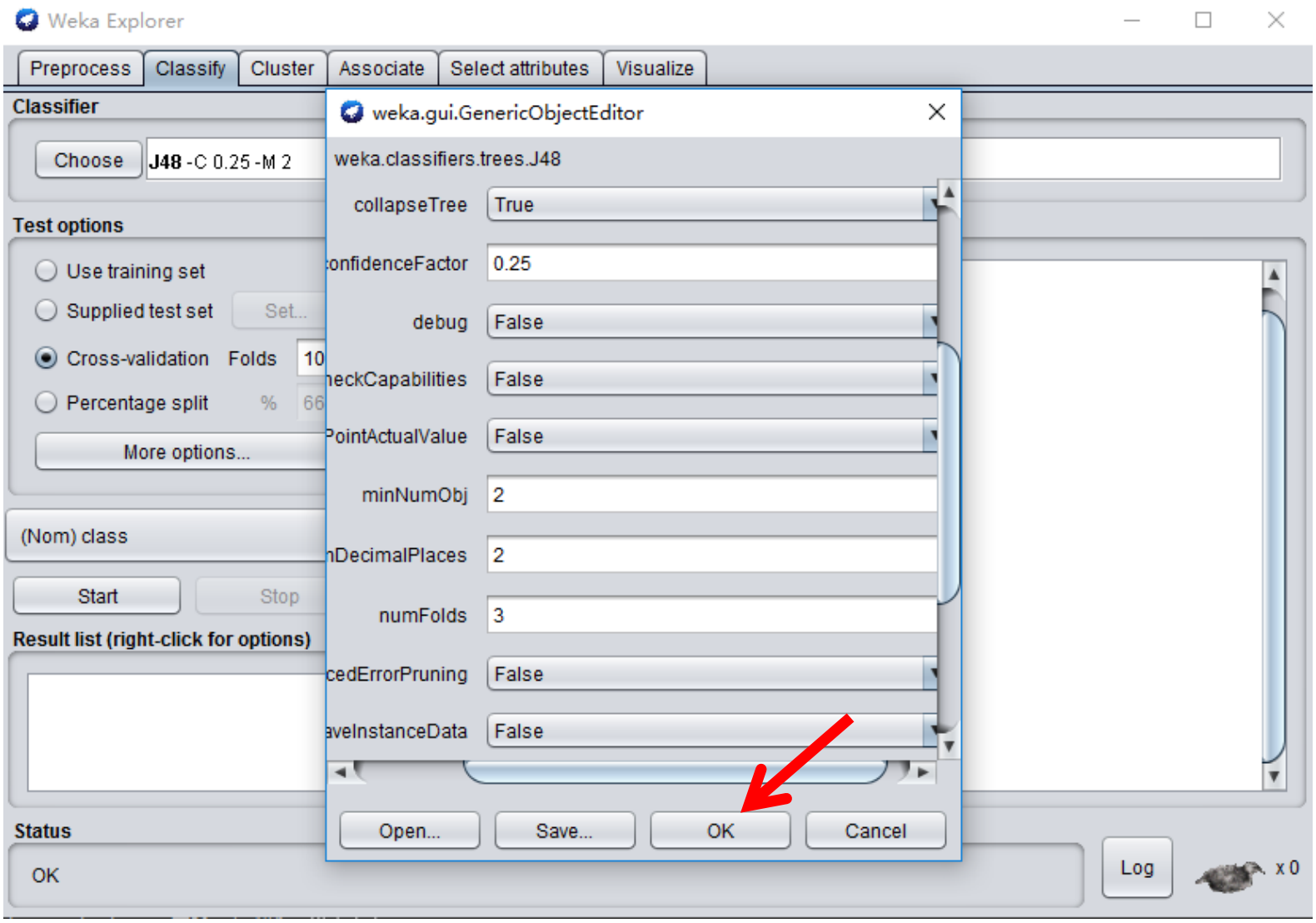
Explorer: Classification



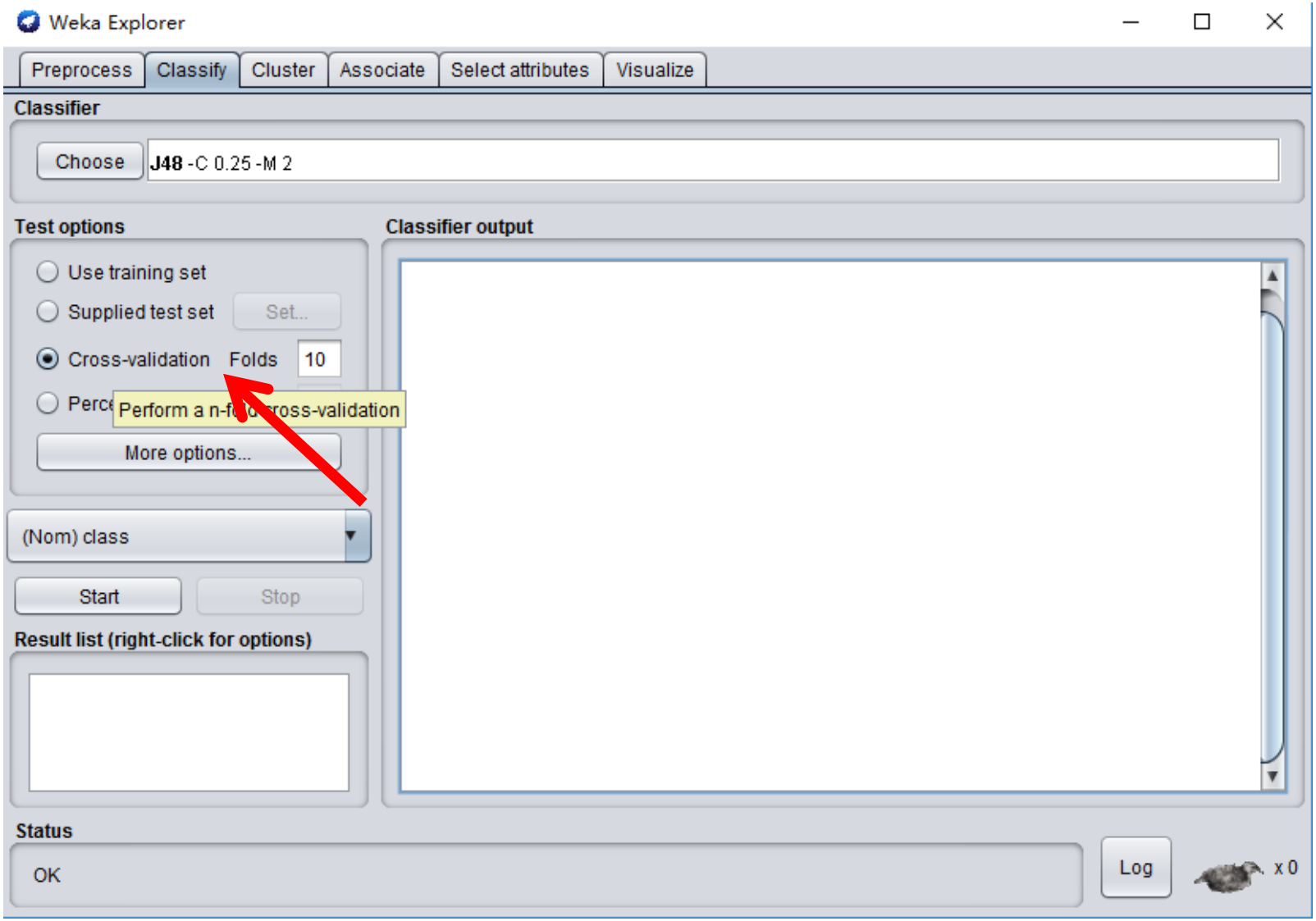
Explorer: Classification



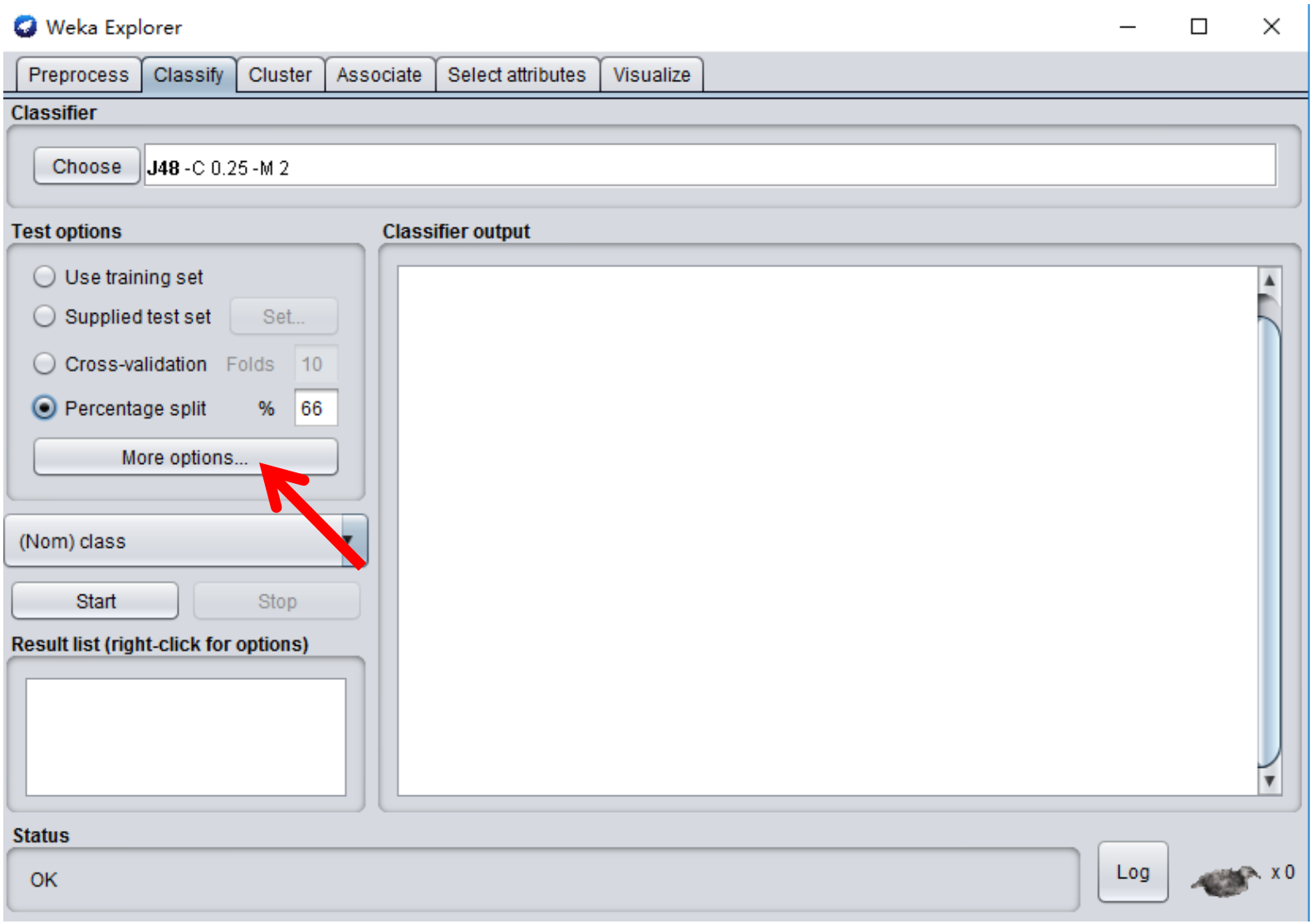
Explorer: Classification



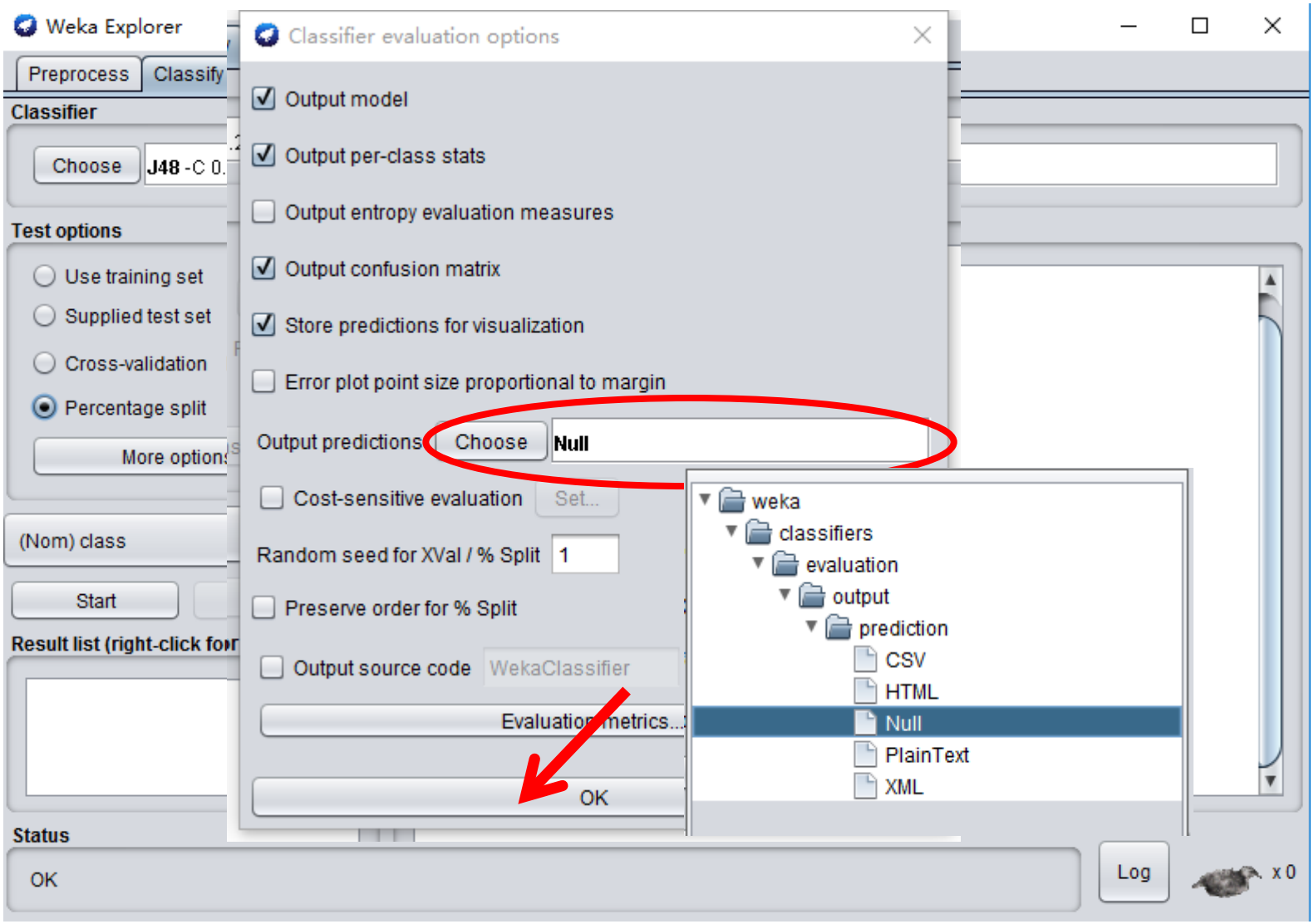
Explorer: Classification



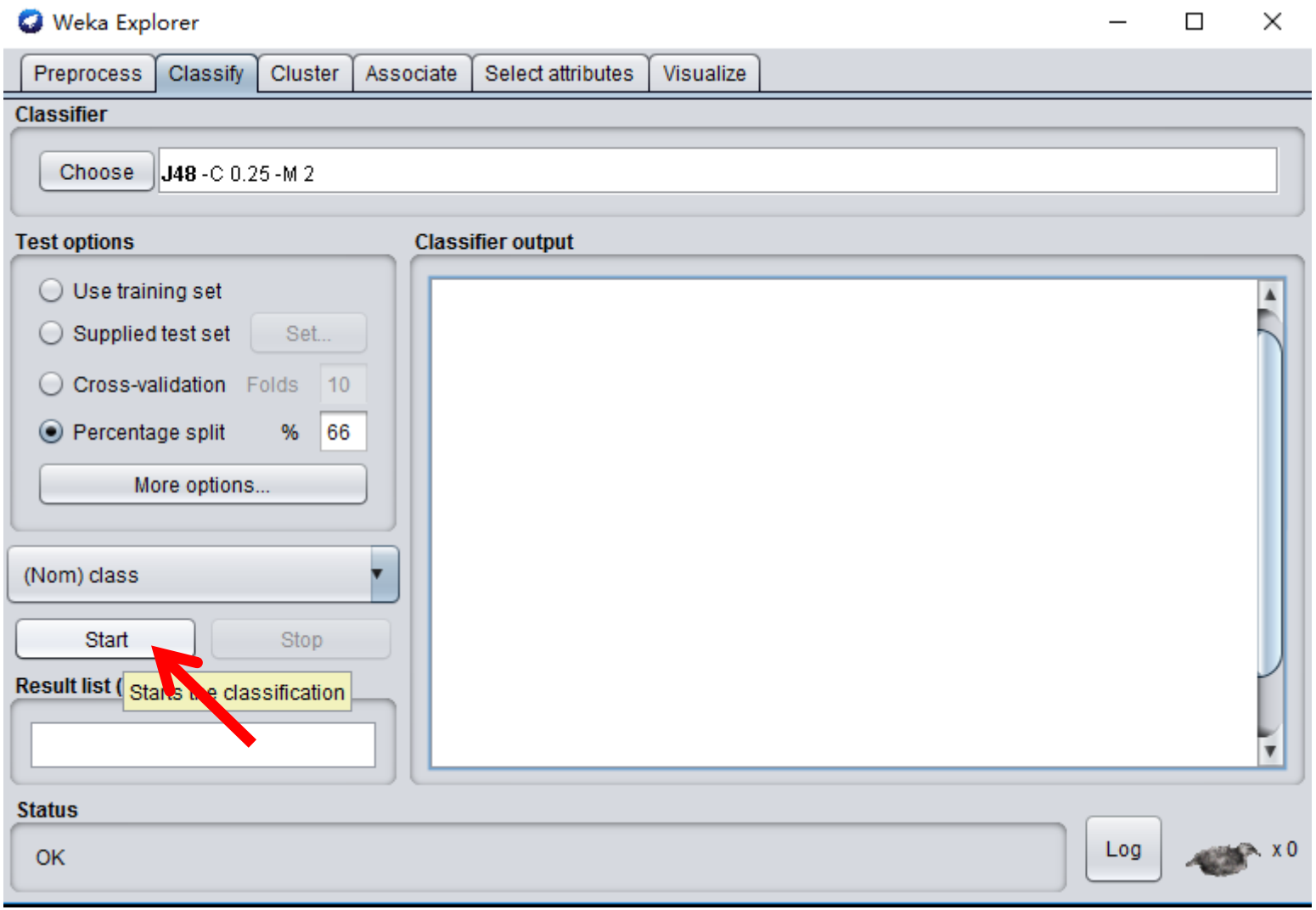
Explorer: Classification



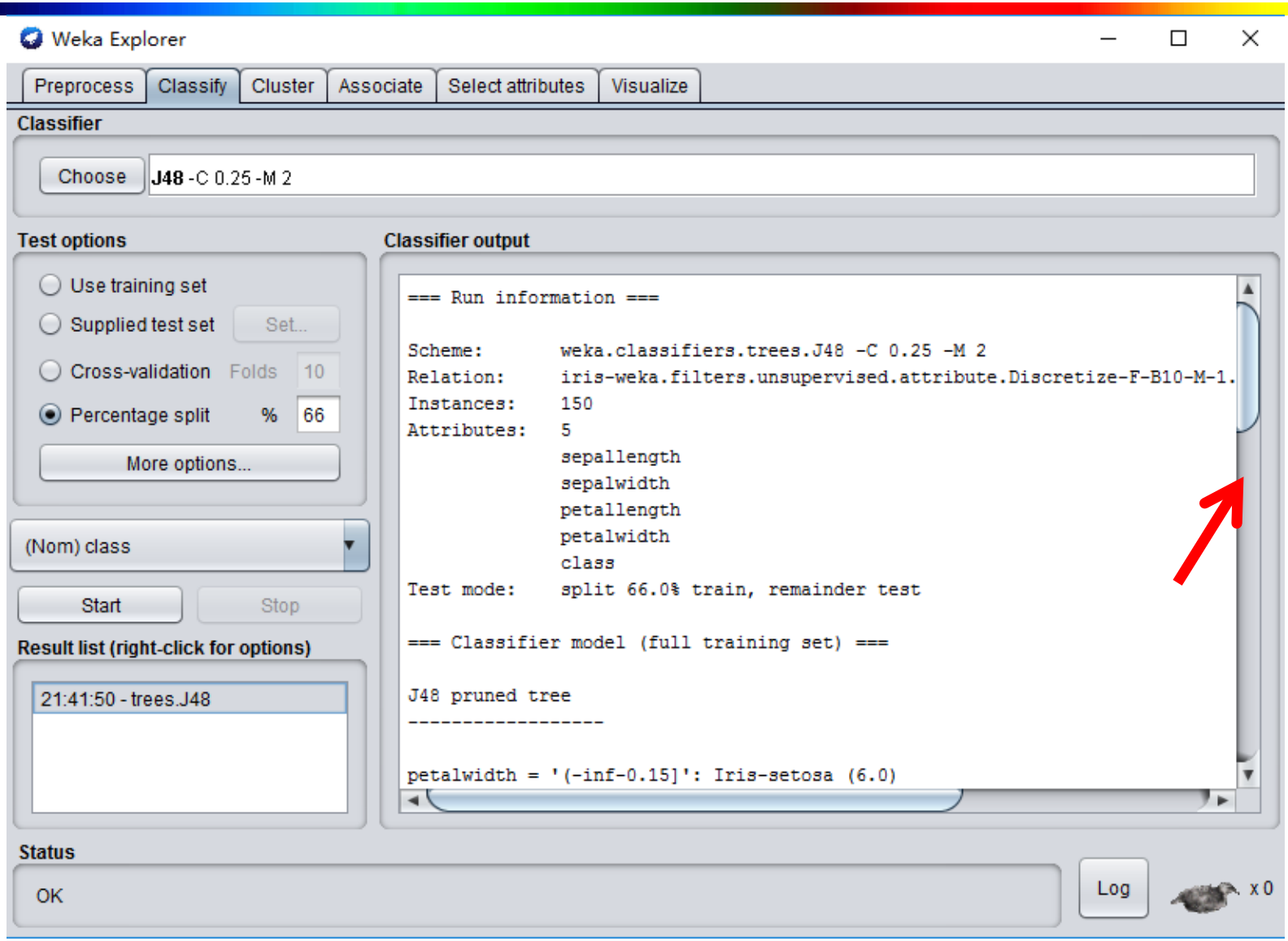
Explorer: Classification



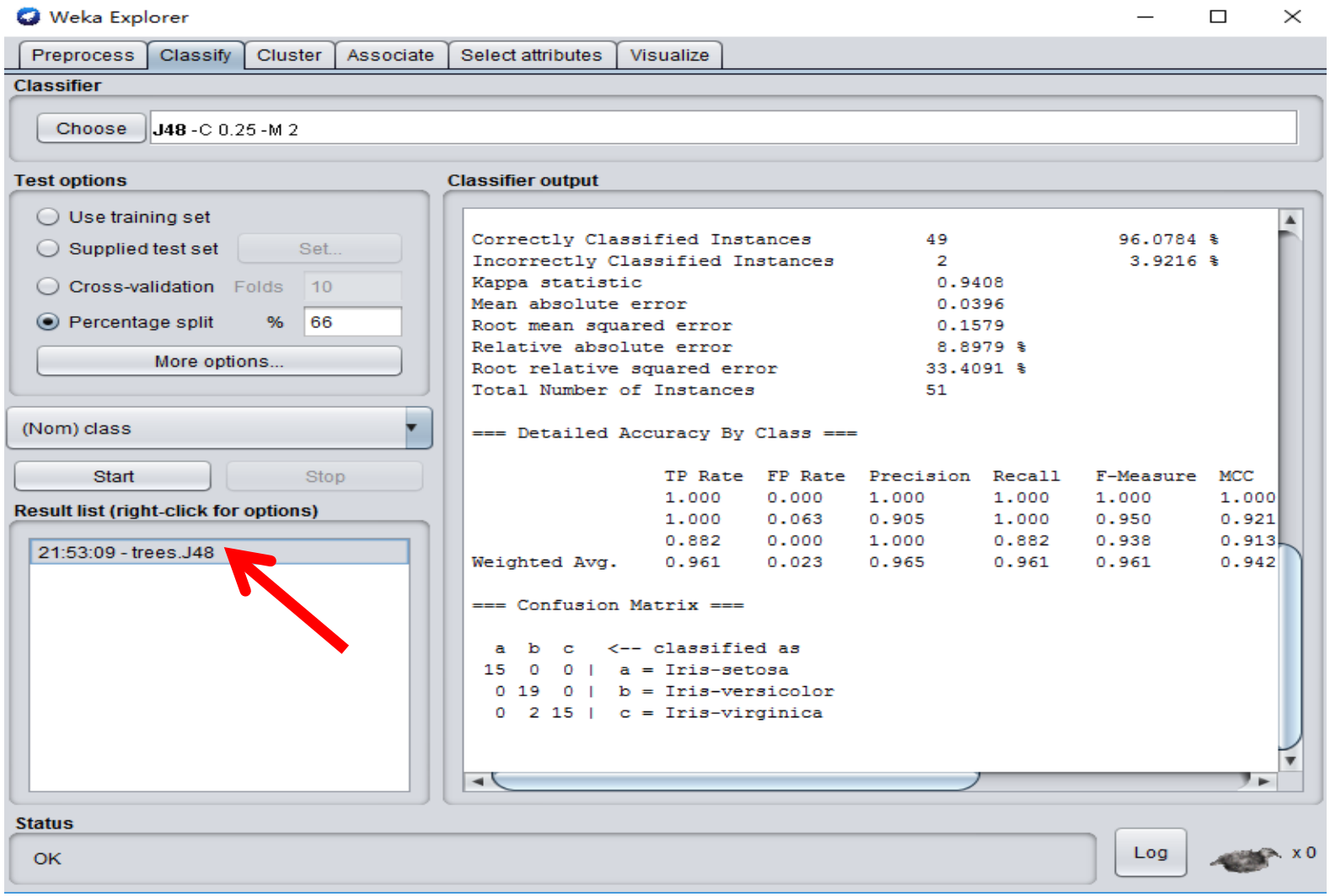
Explorer: Classification



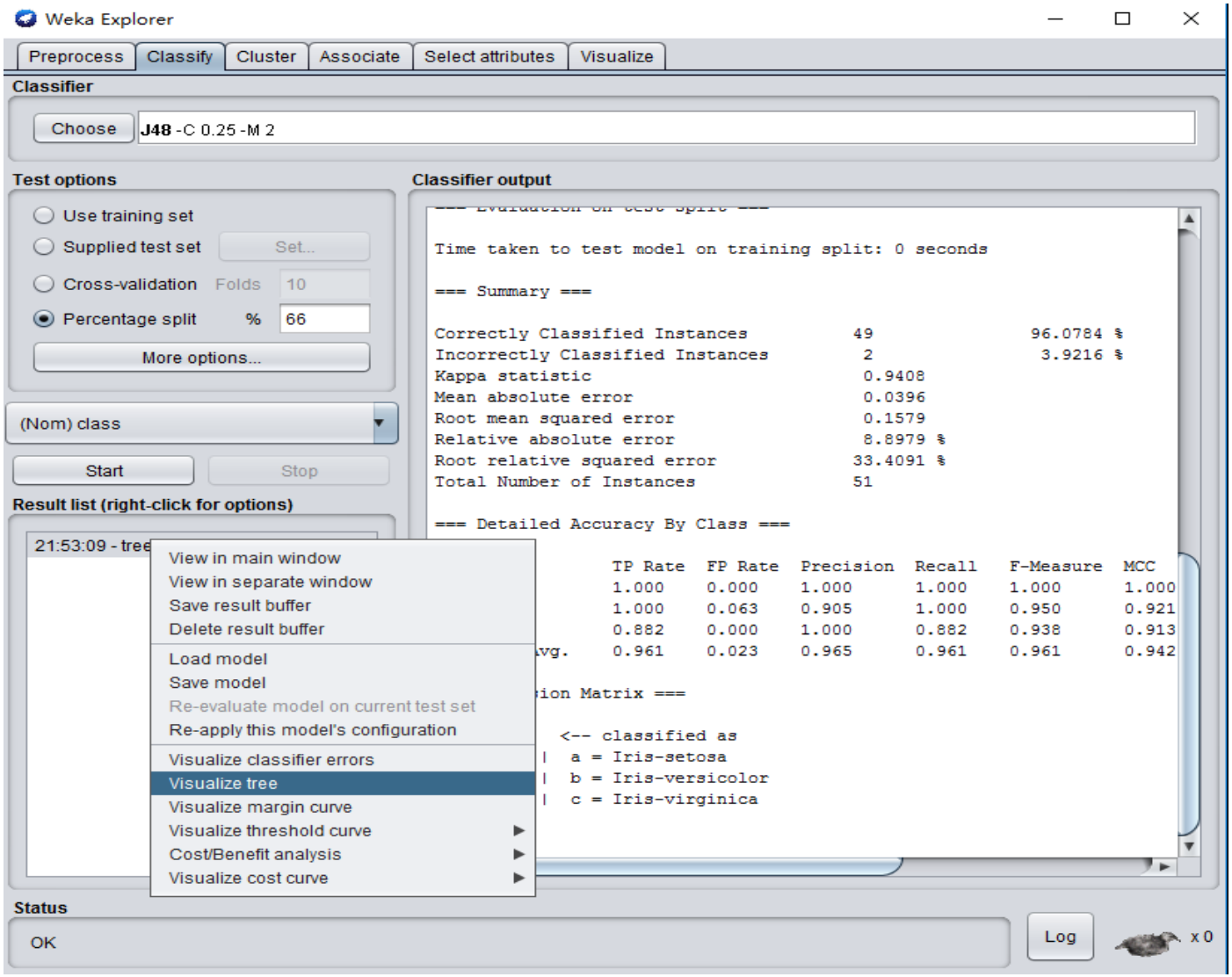
Explorer: Classification



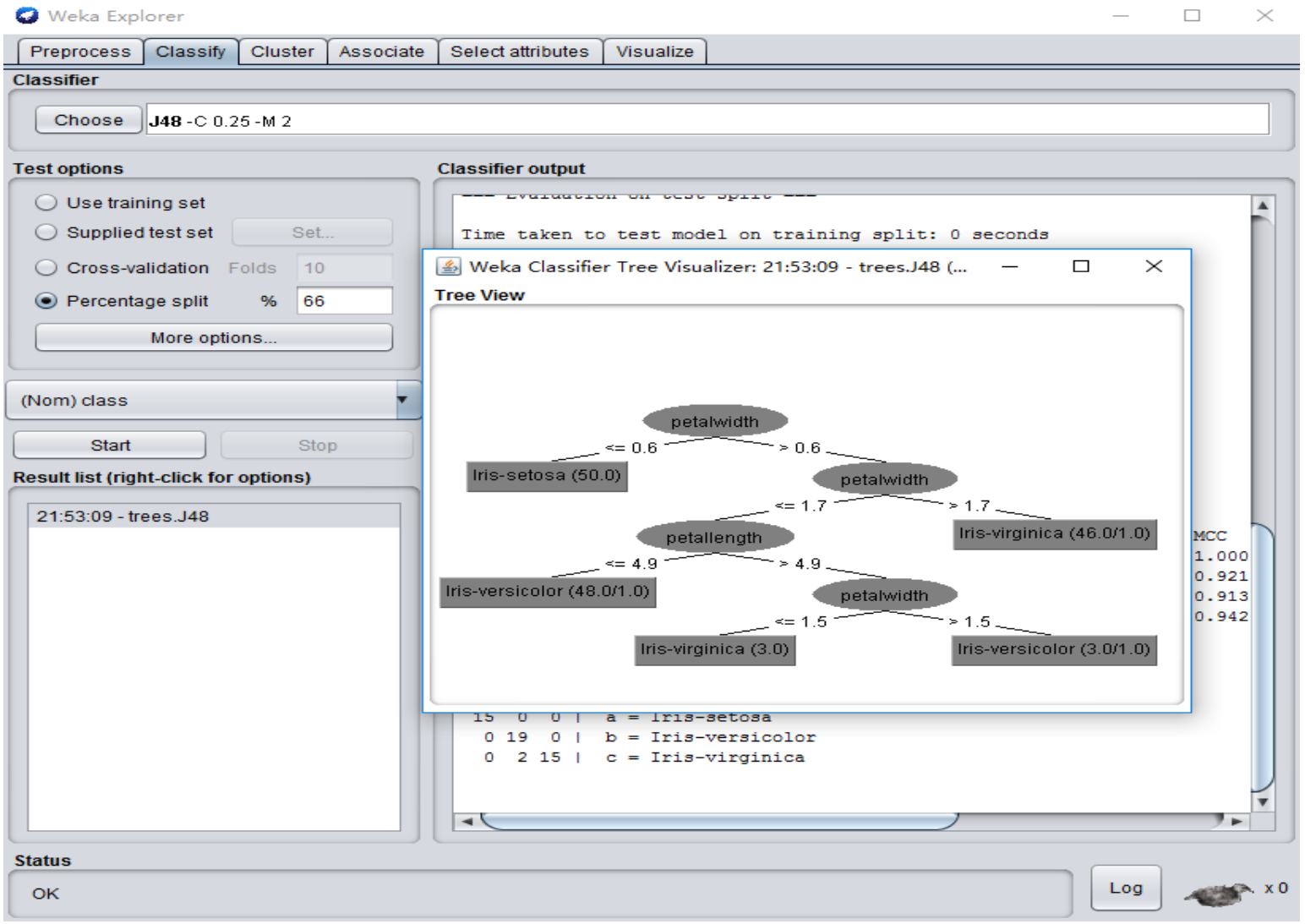
Explorer: Classification



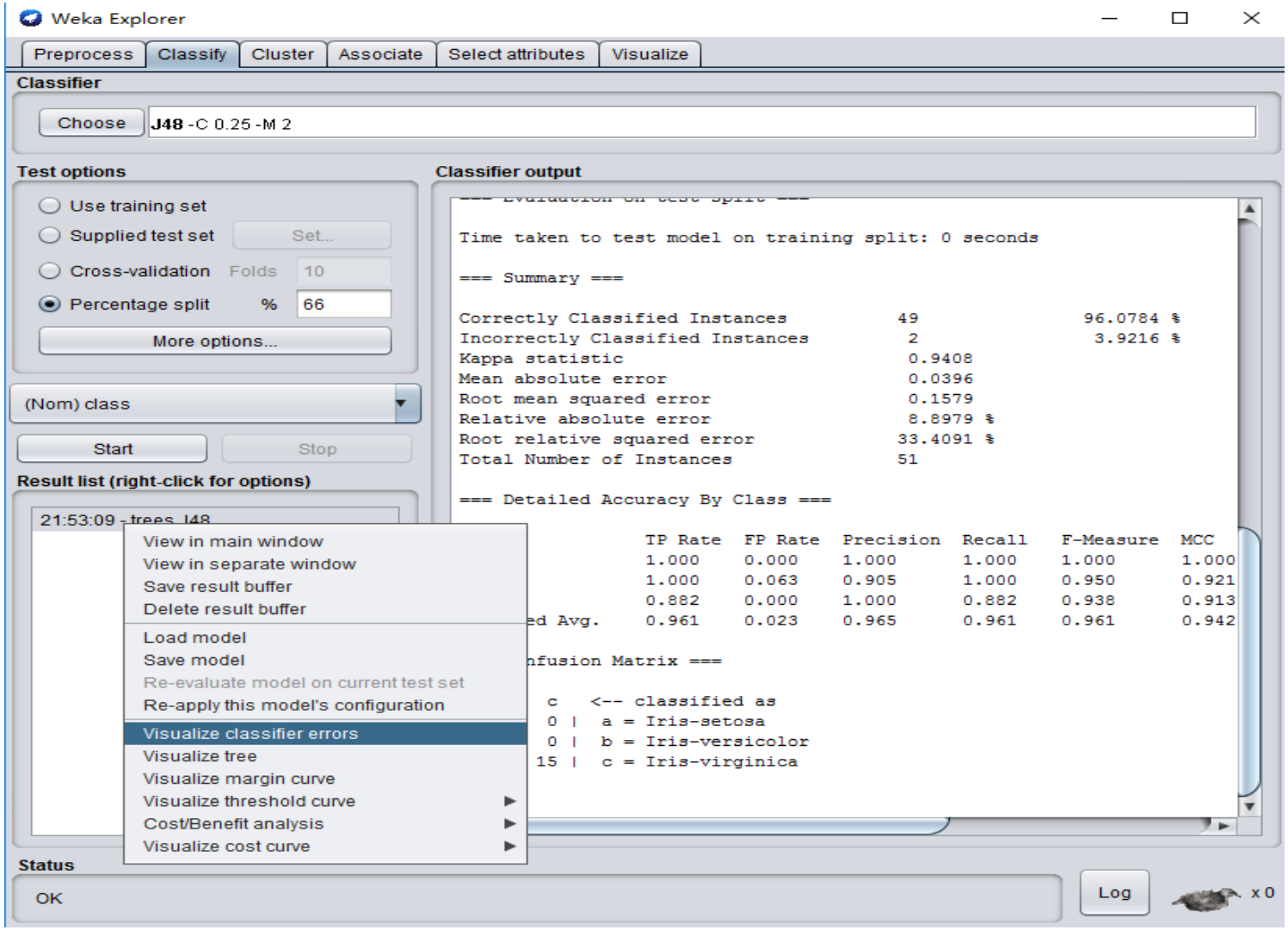
Explorer: Classification



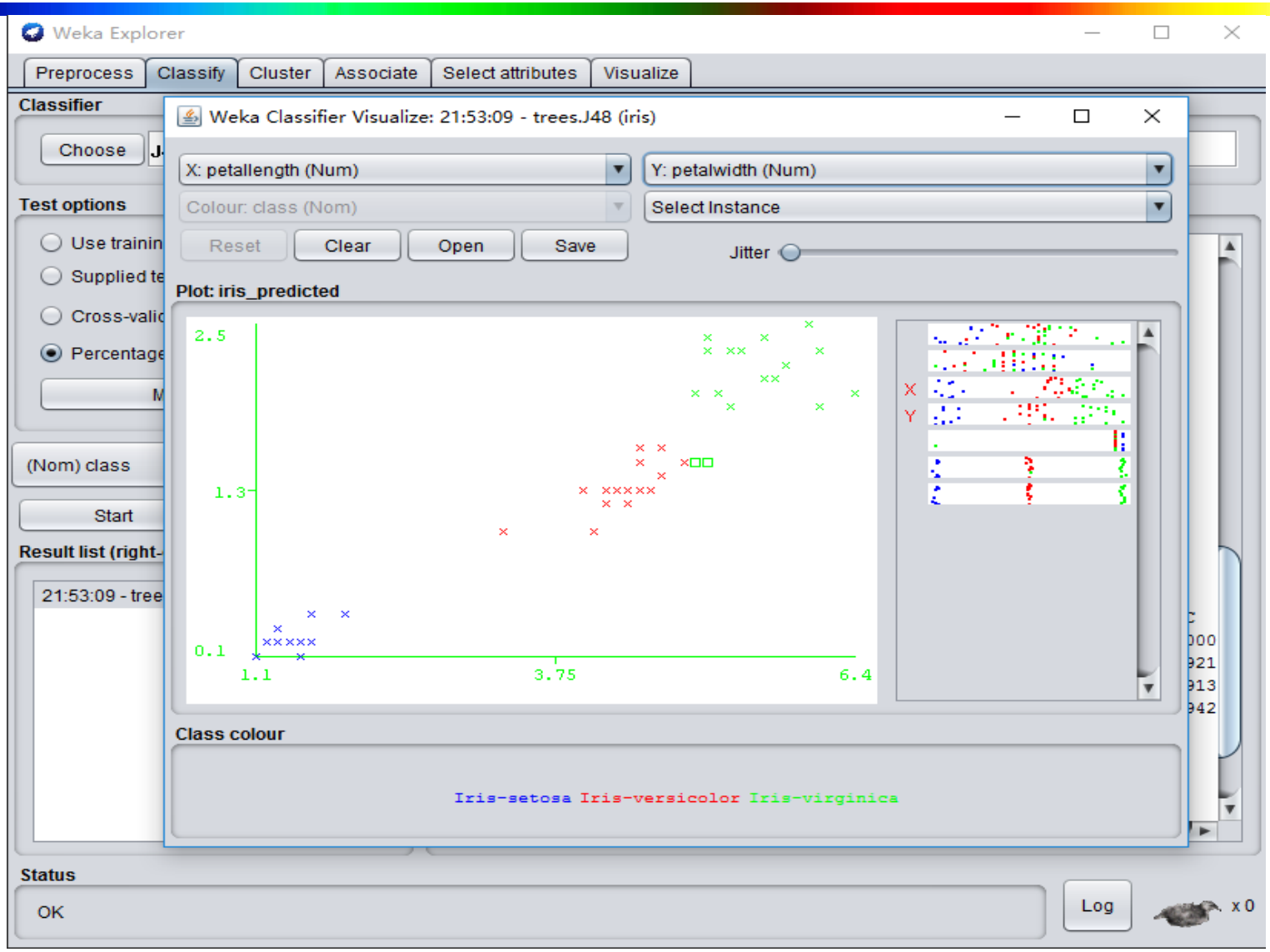
Explorer: Classification



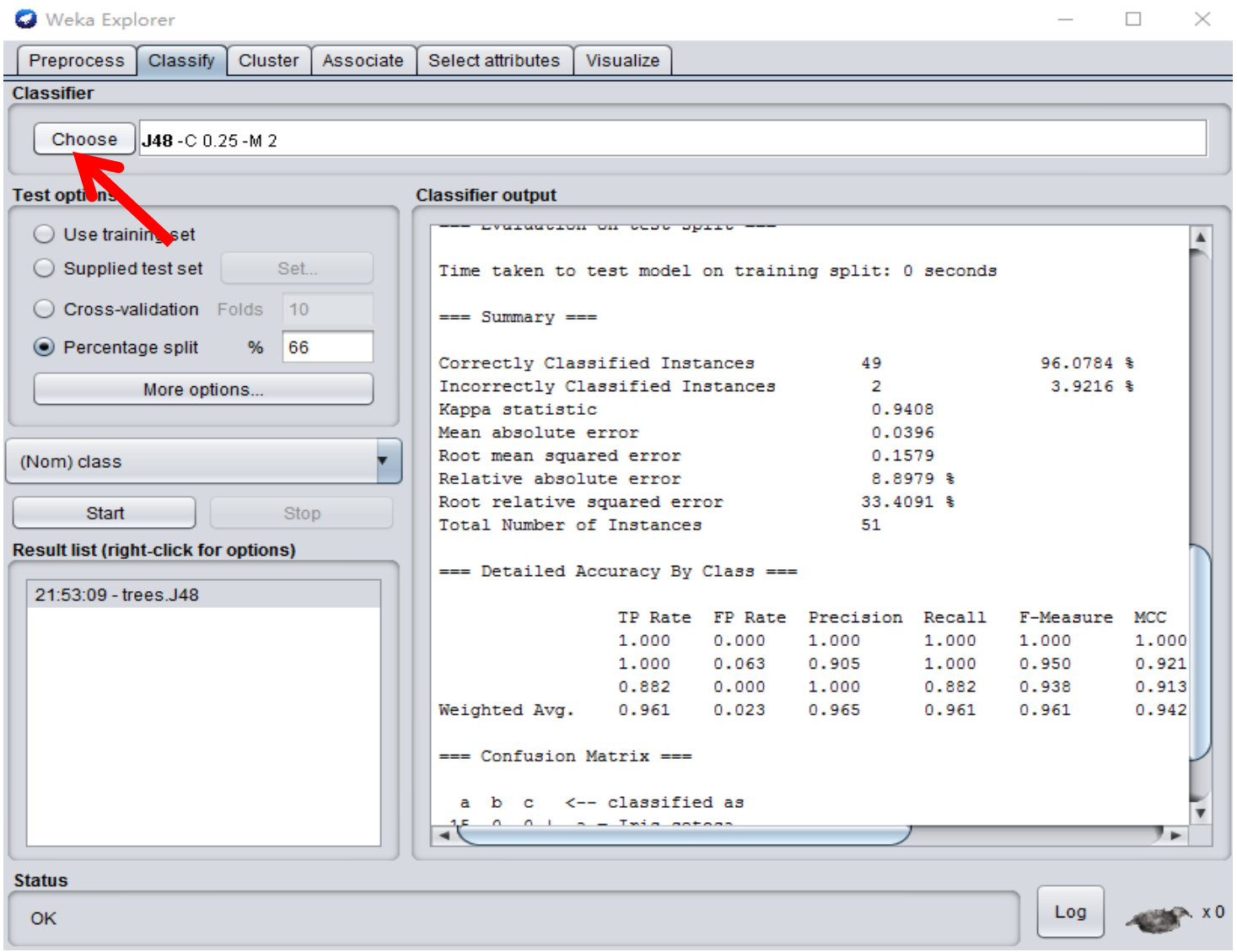
Explorer: Classification



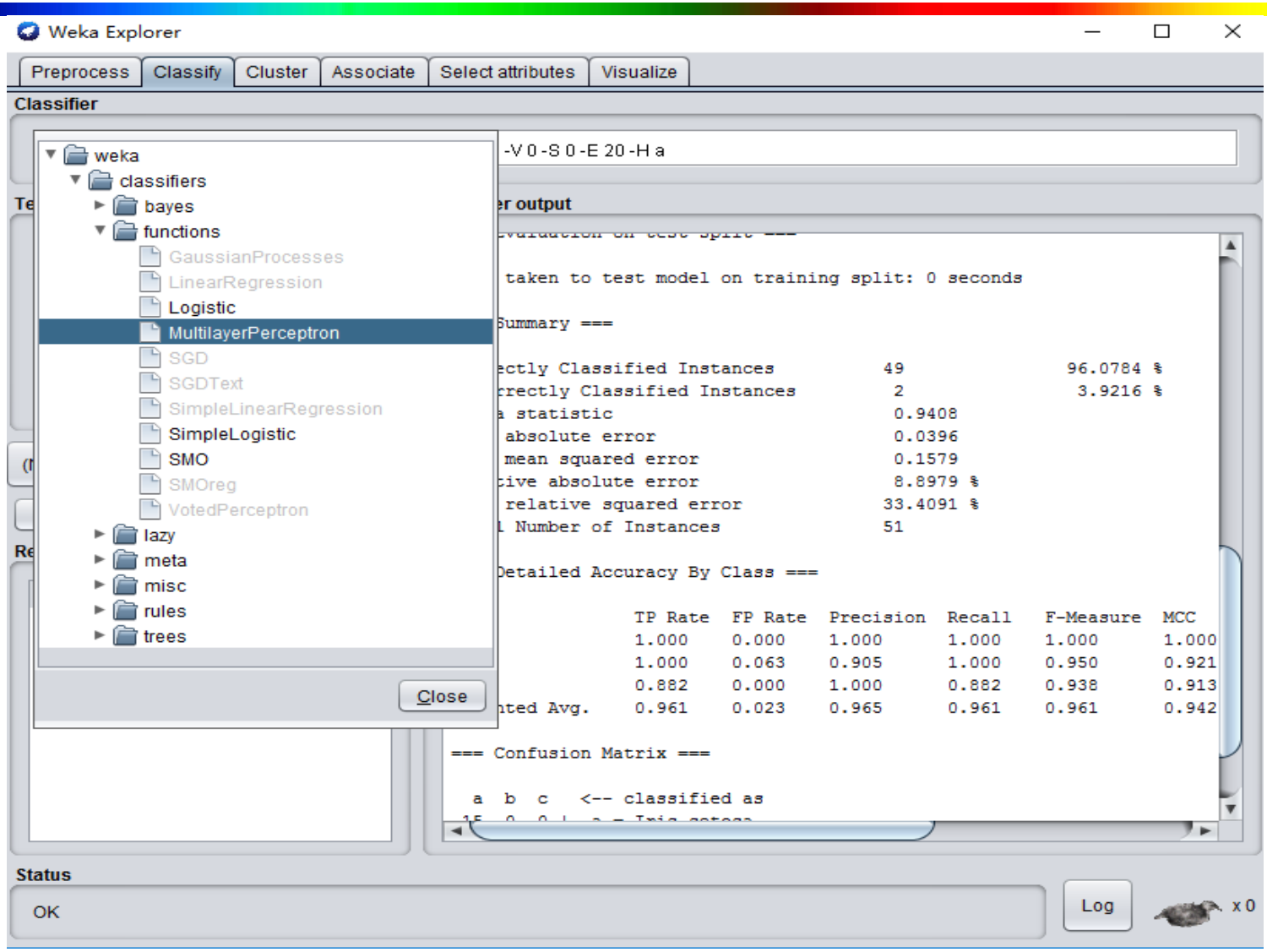
Explorer: Classification



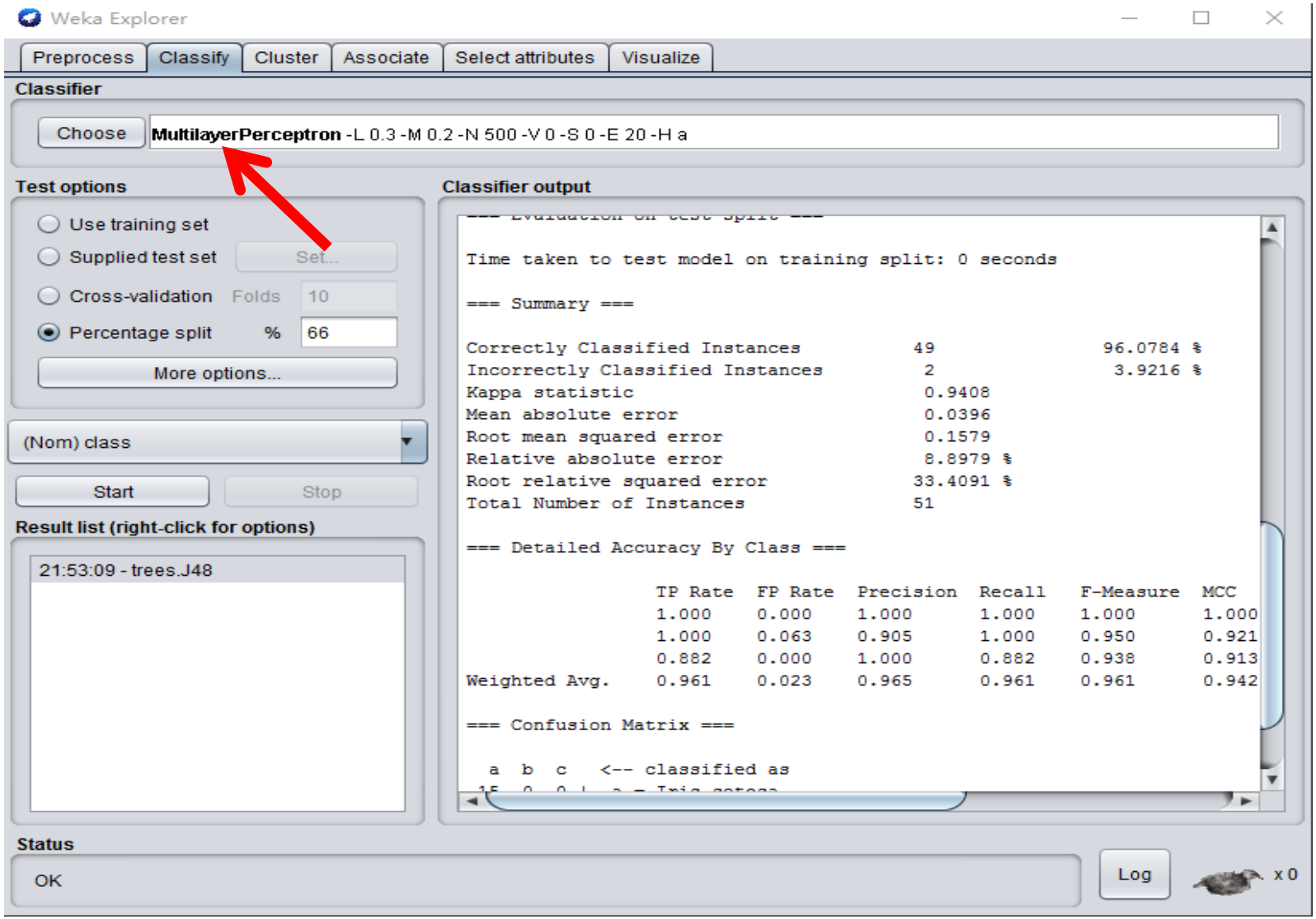
Explorer: Classification



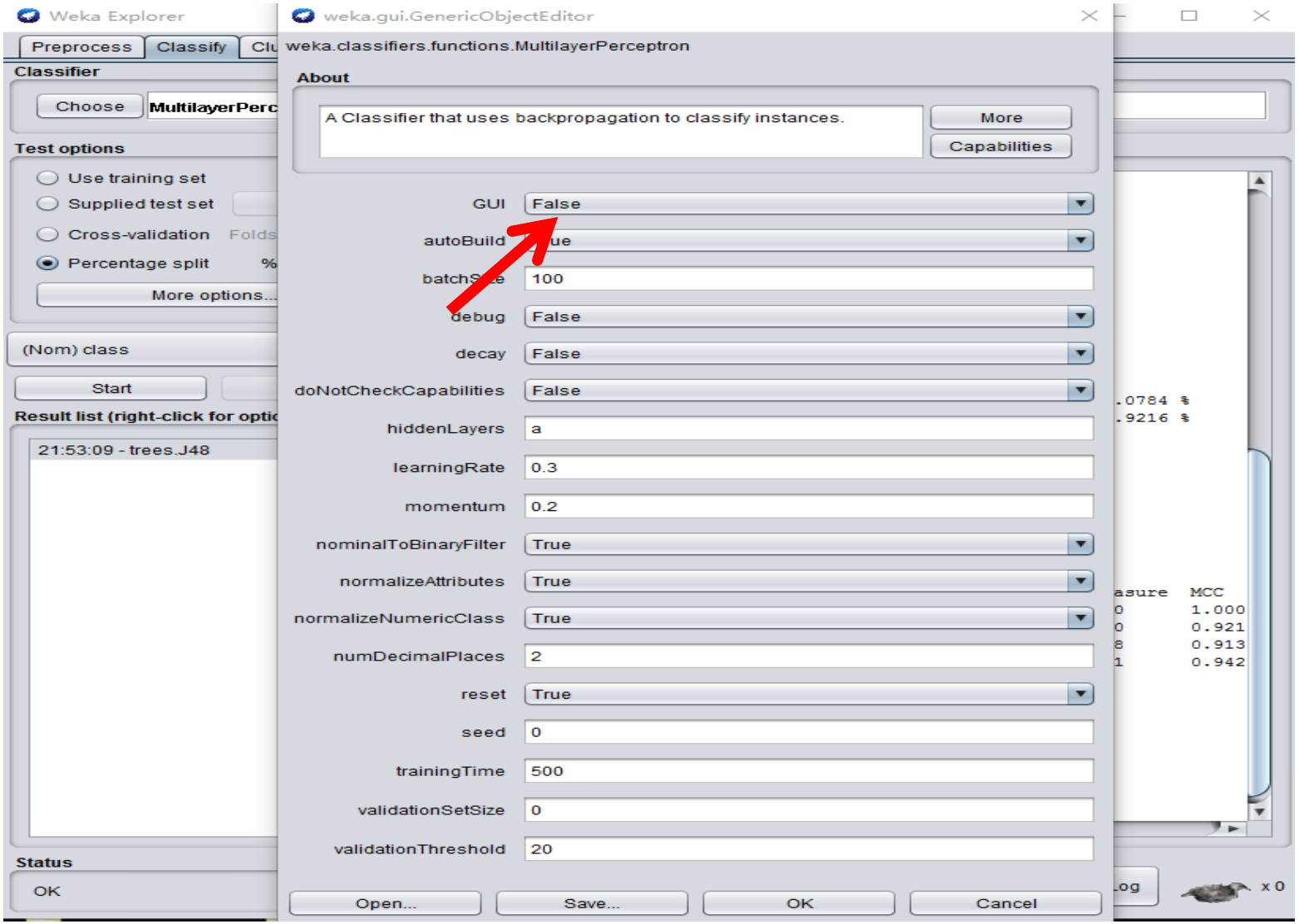
Explorer: Classification



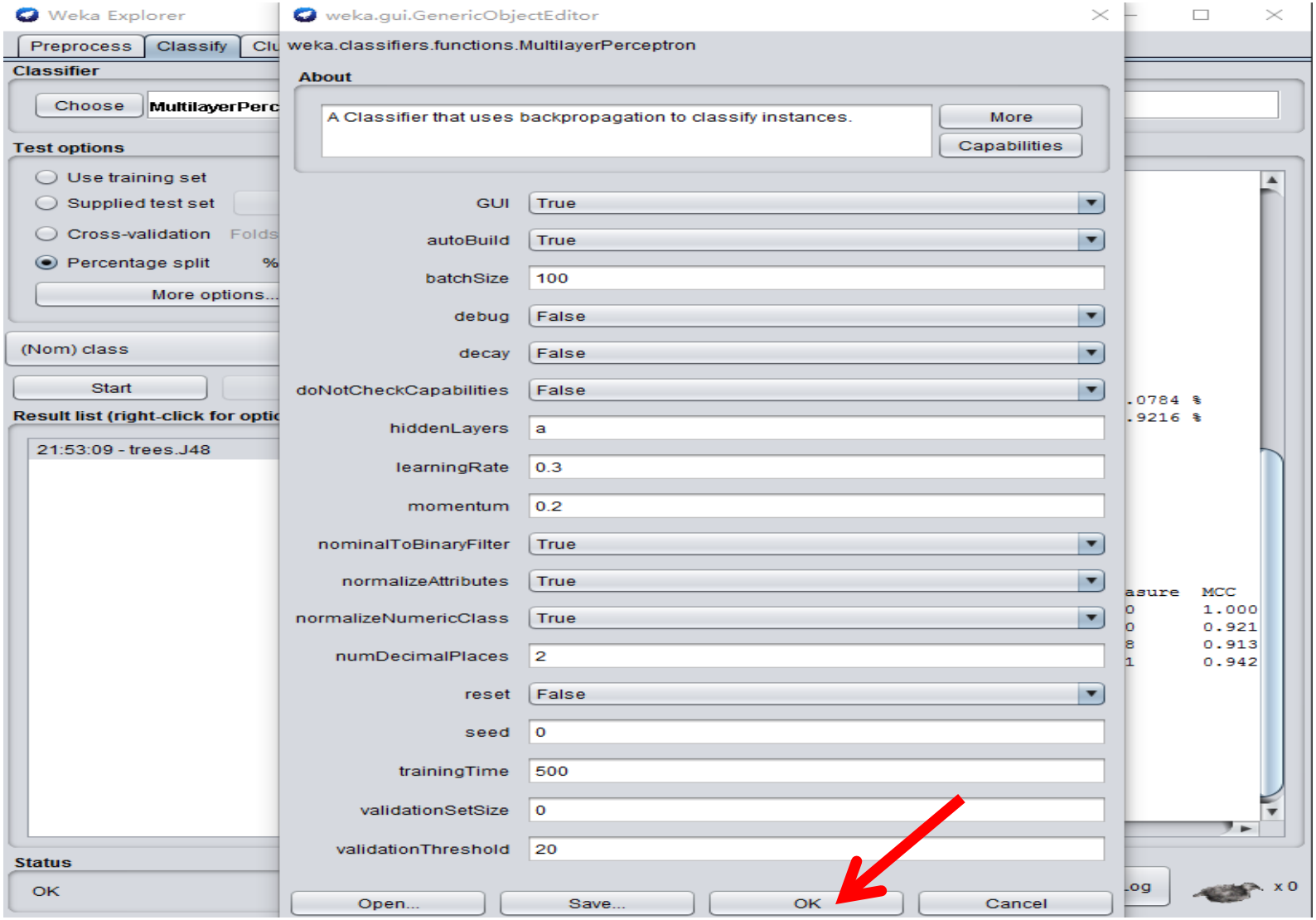
Explorer: Classification



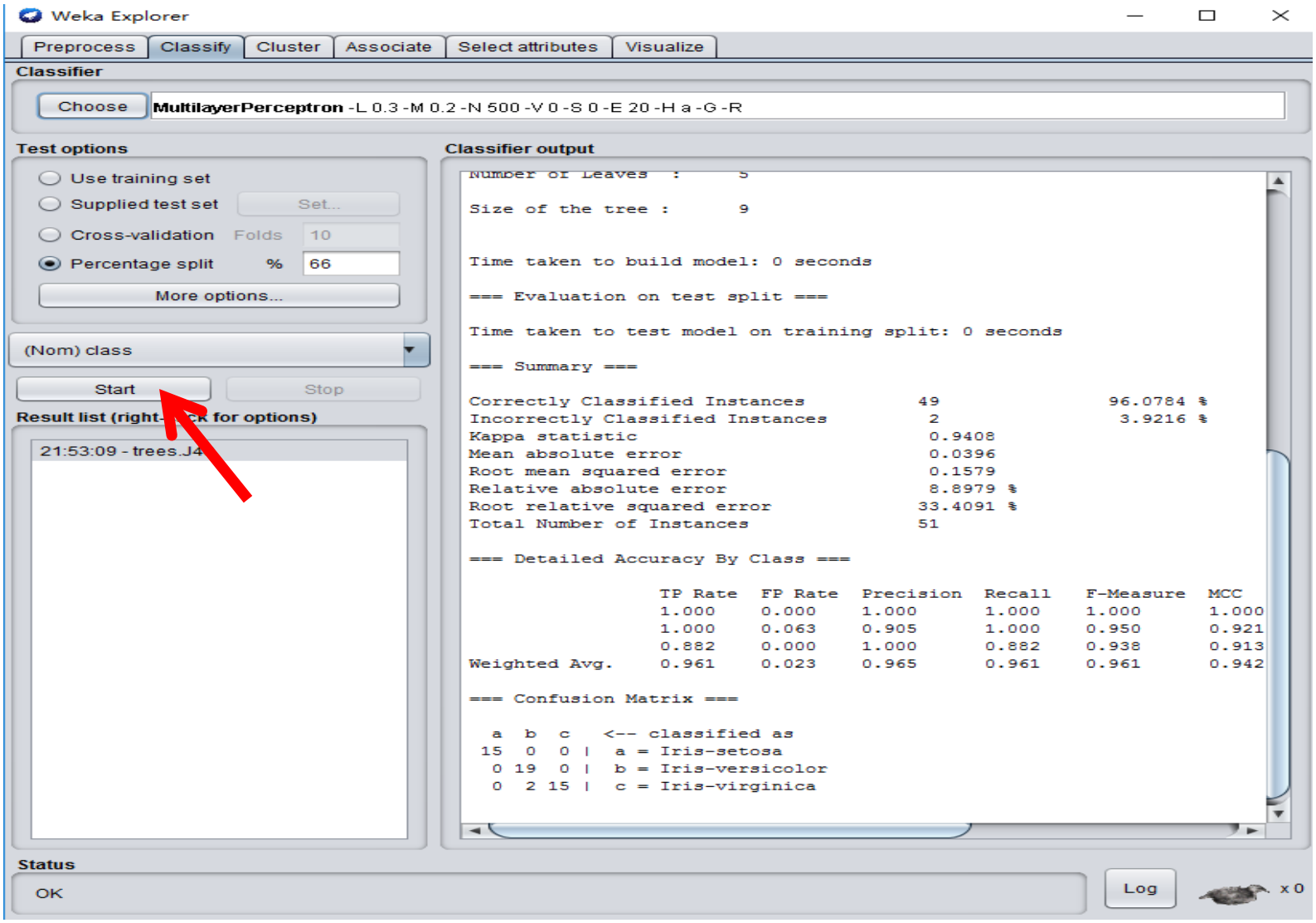
Explorer: Classification



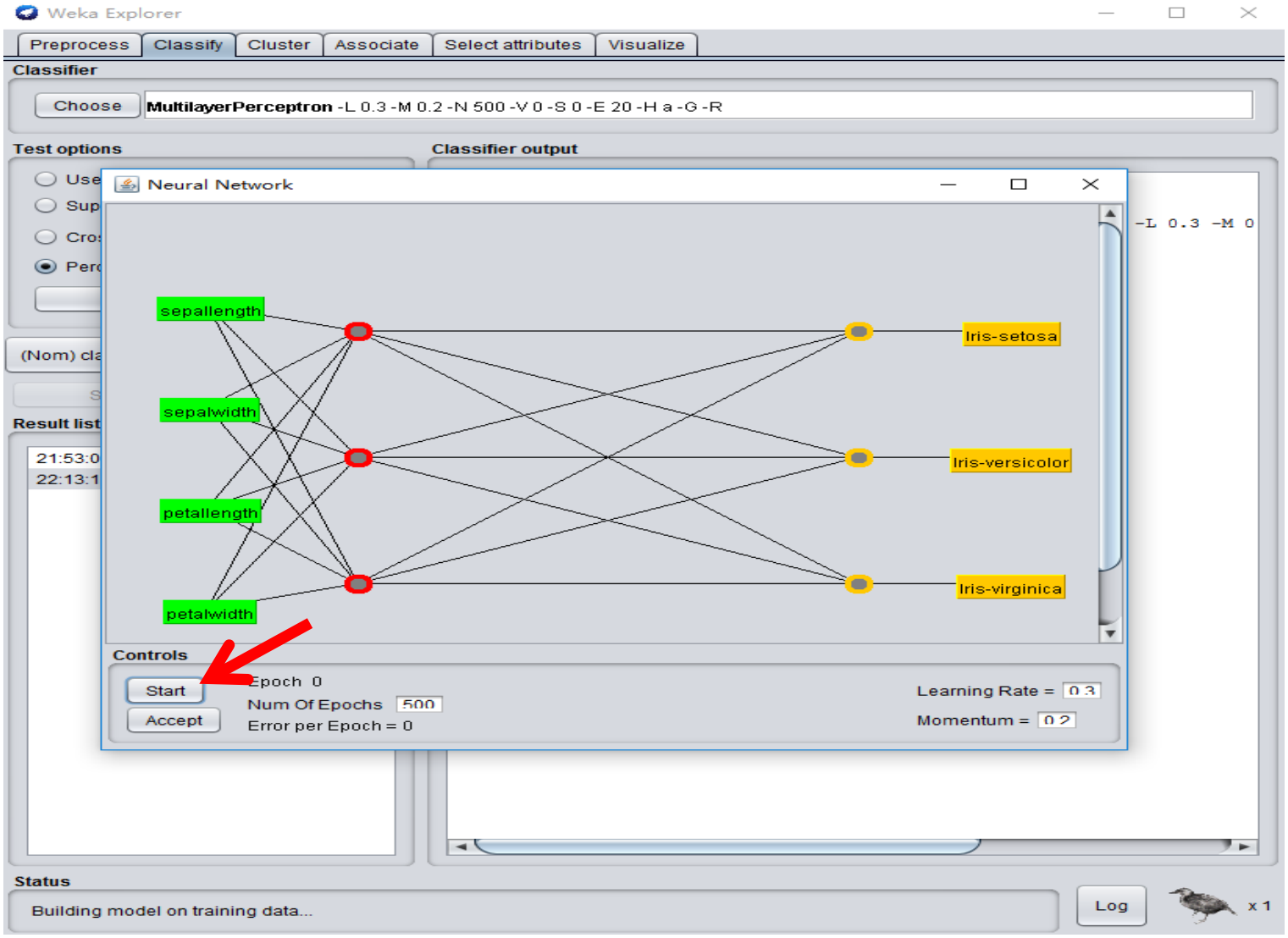
Explorer: Classification



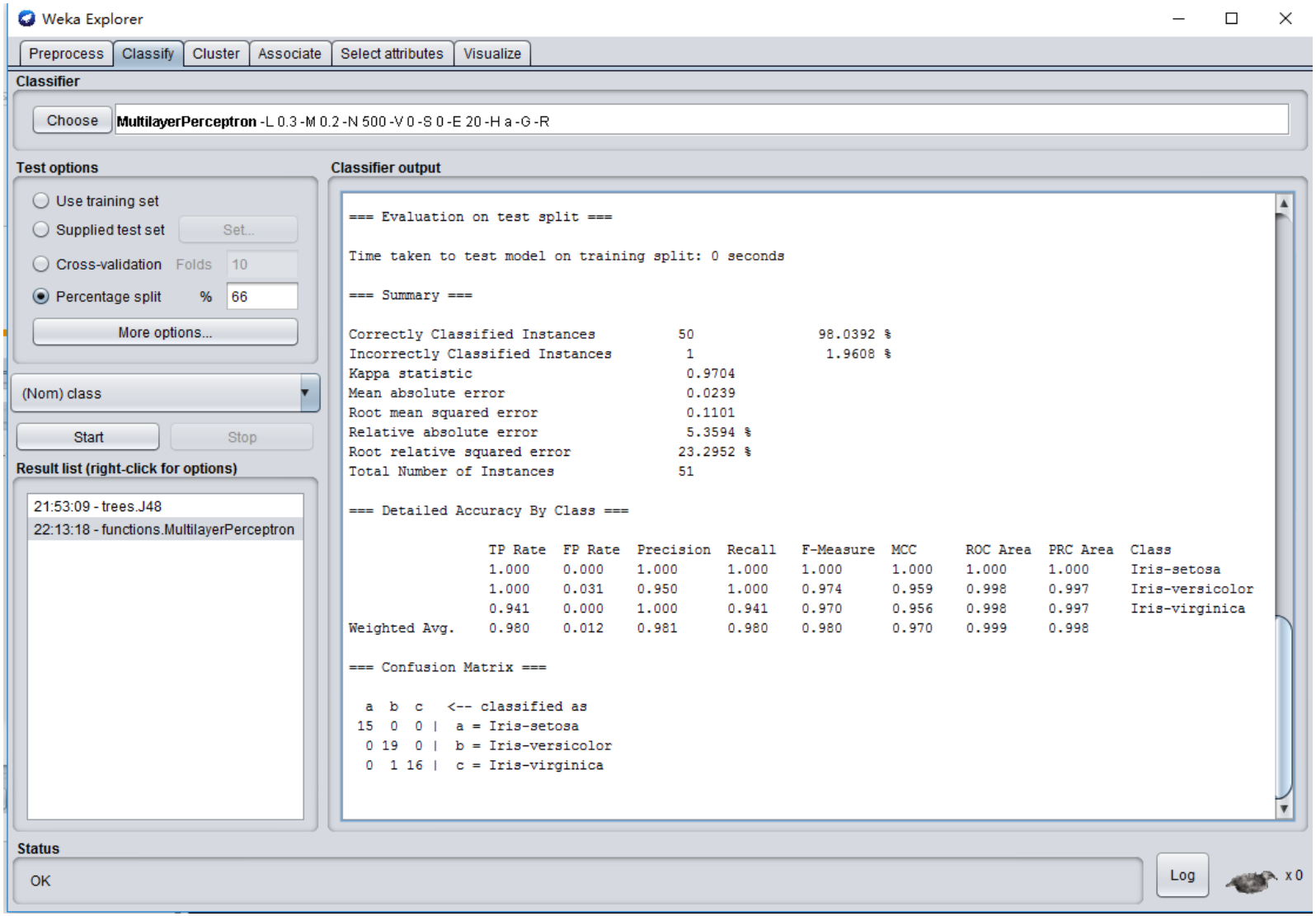
Explorer: Classification



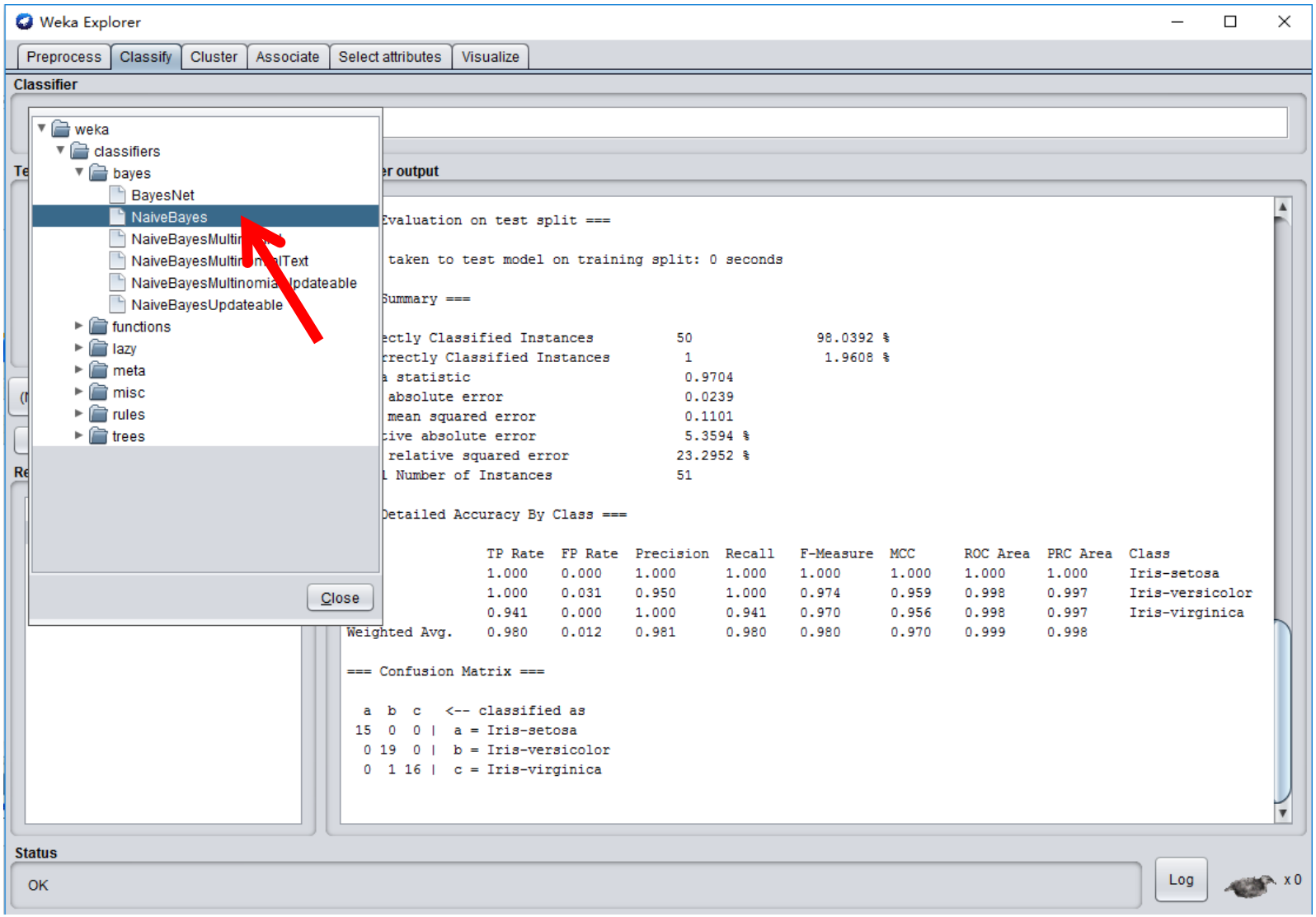
Explorer: Classification



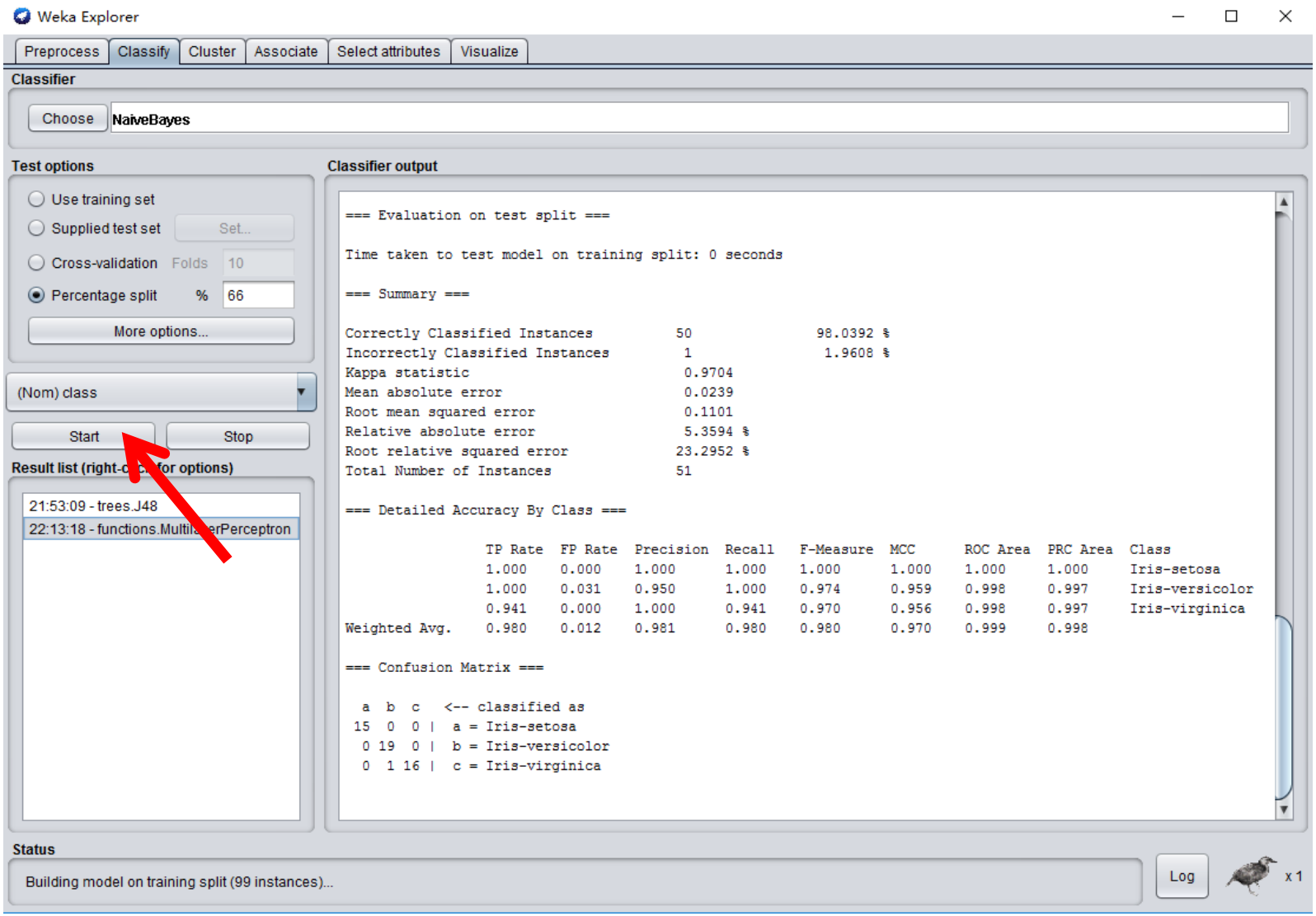
Explorer: Classification



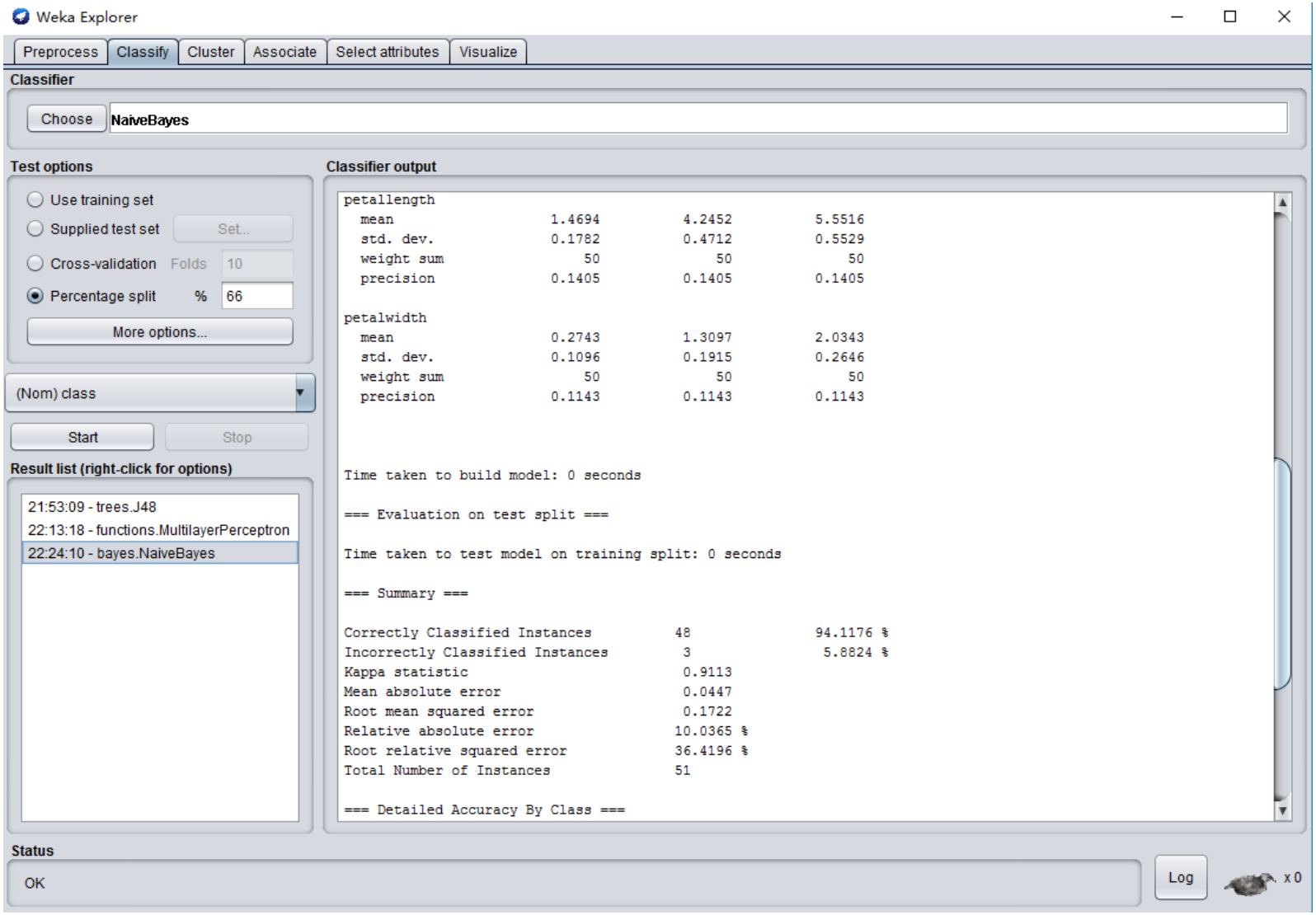
Explorer: Classification



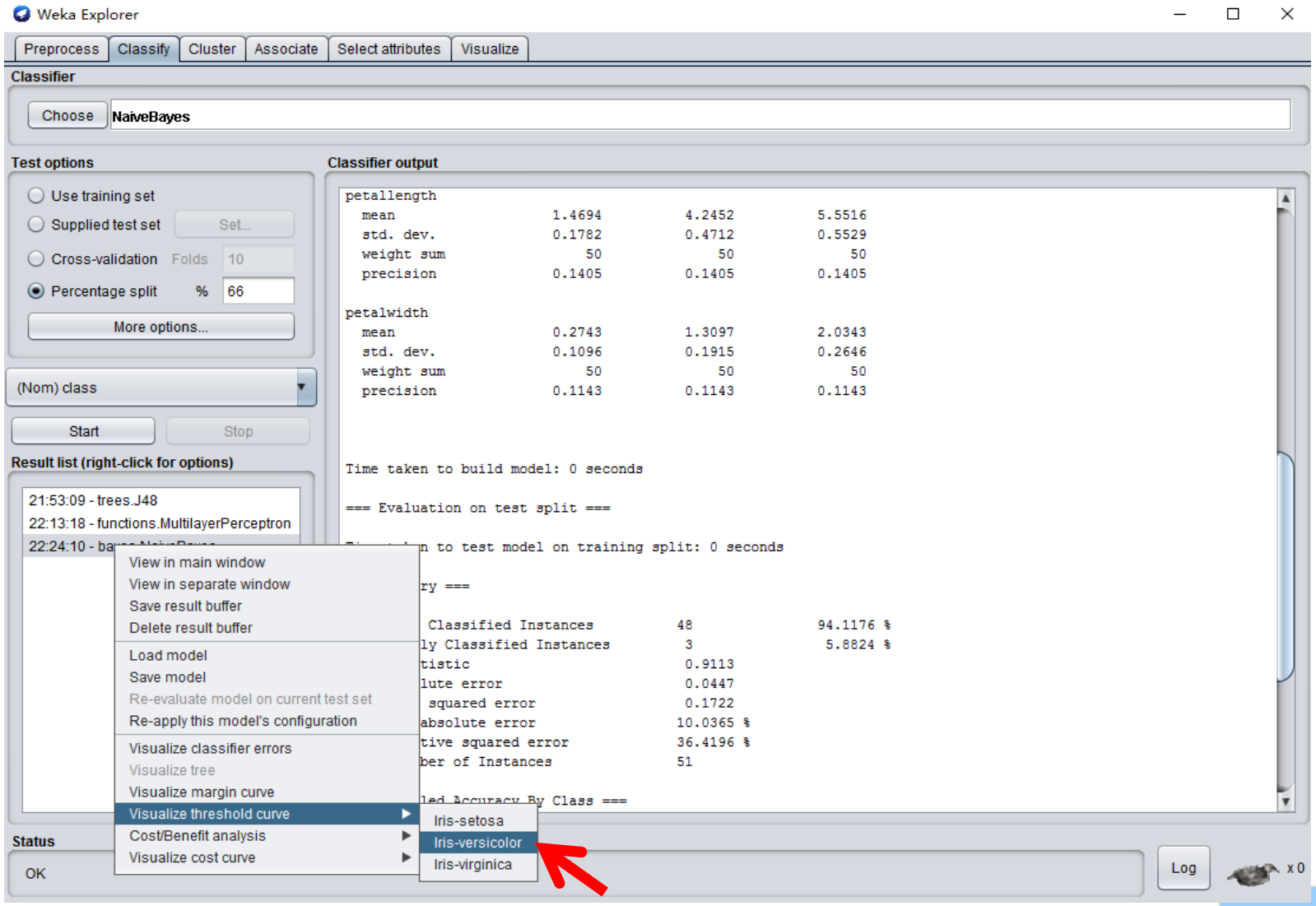
Explorer: Classification



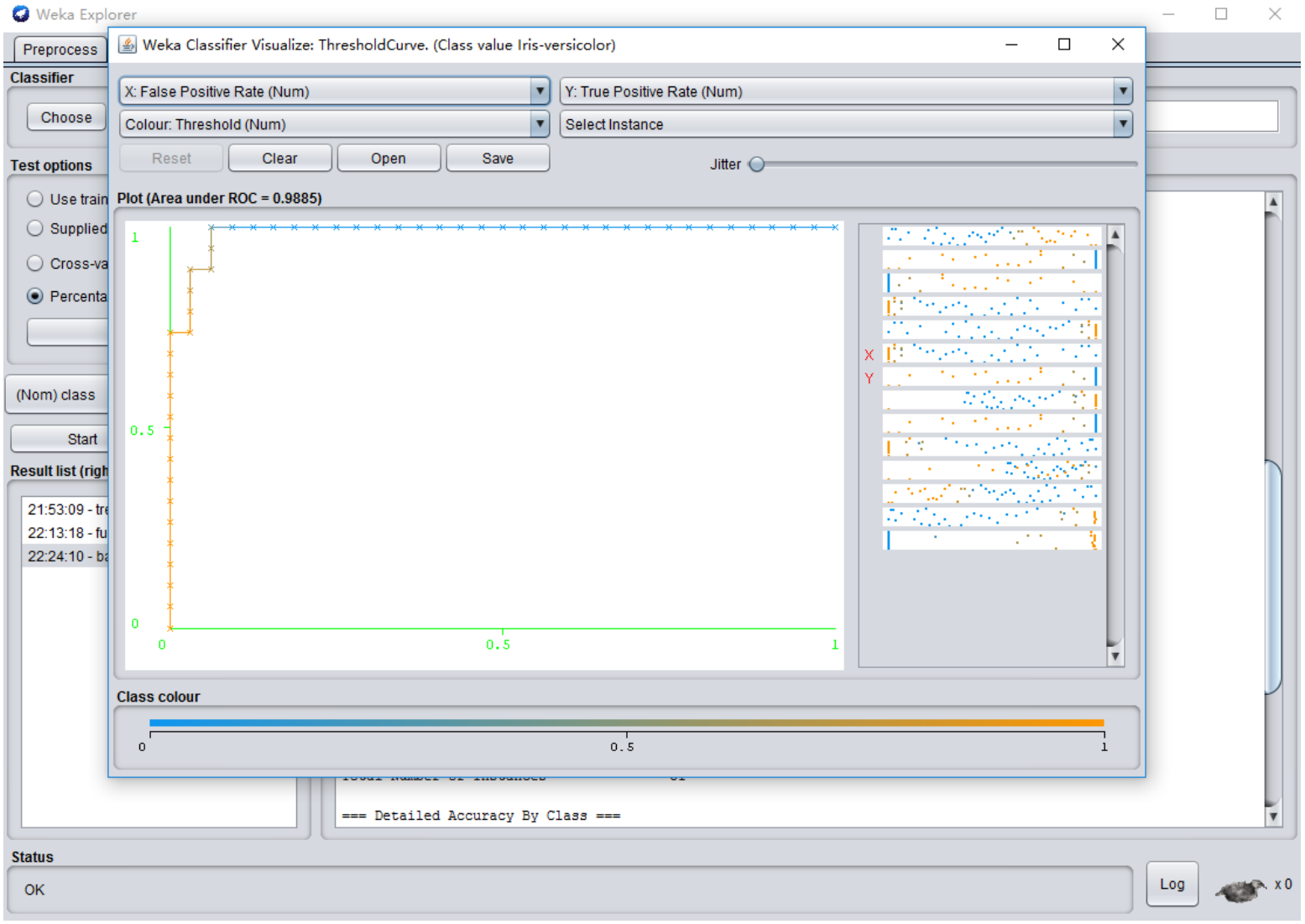
Explorer: Classification



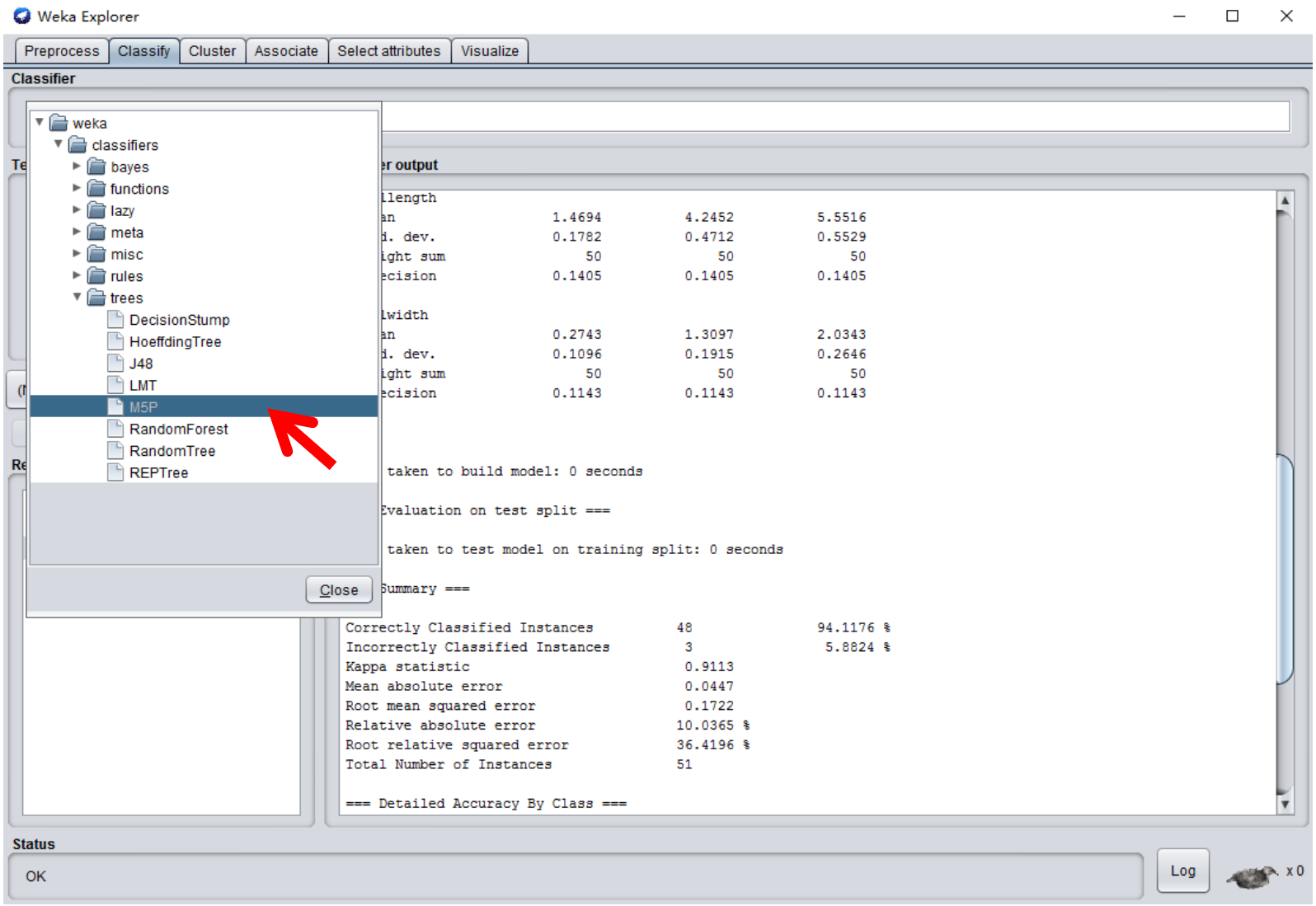
Explorer: Classification



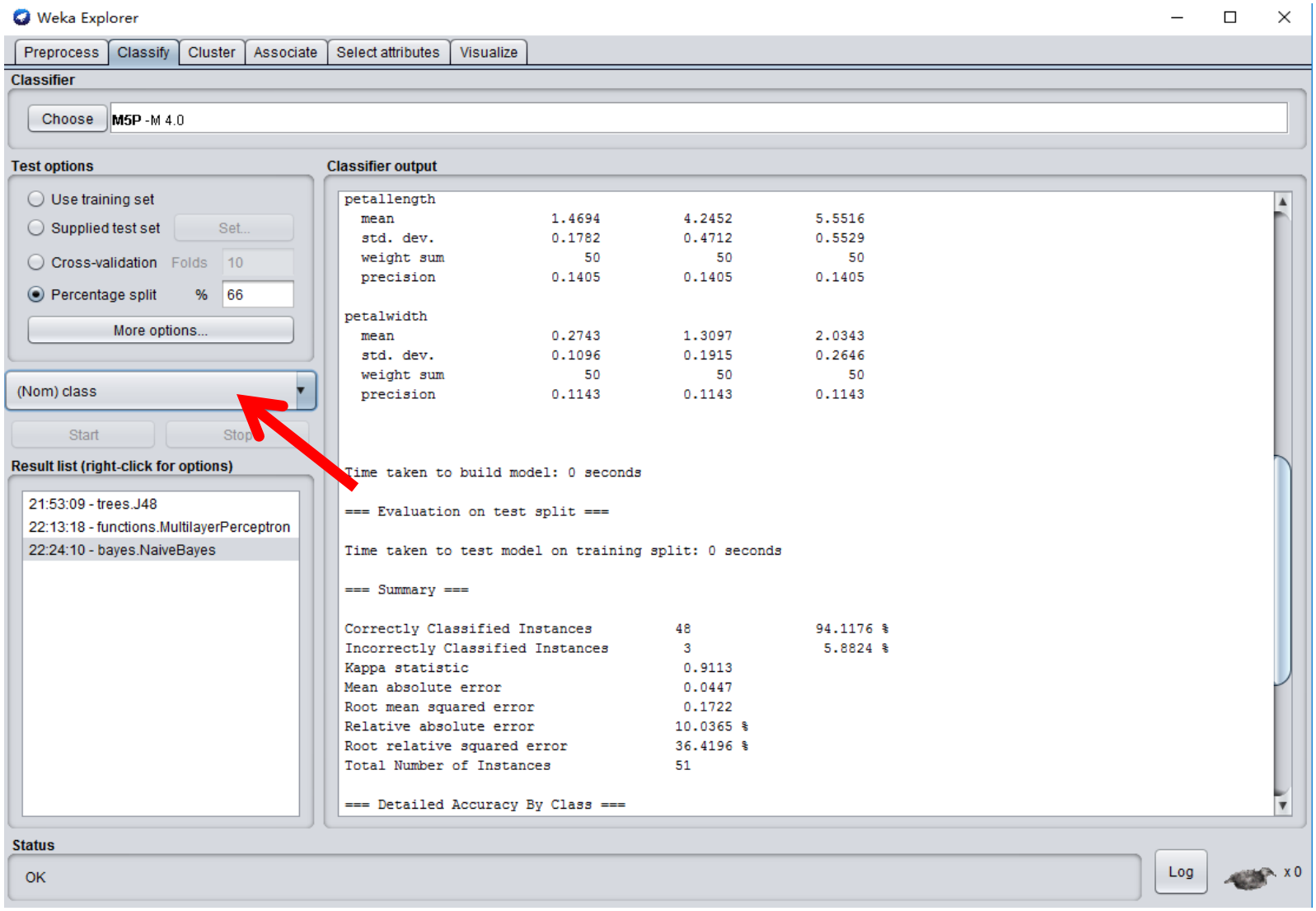
Explorer: Classification



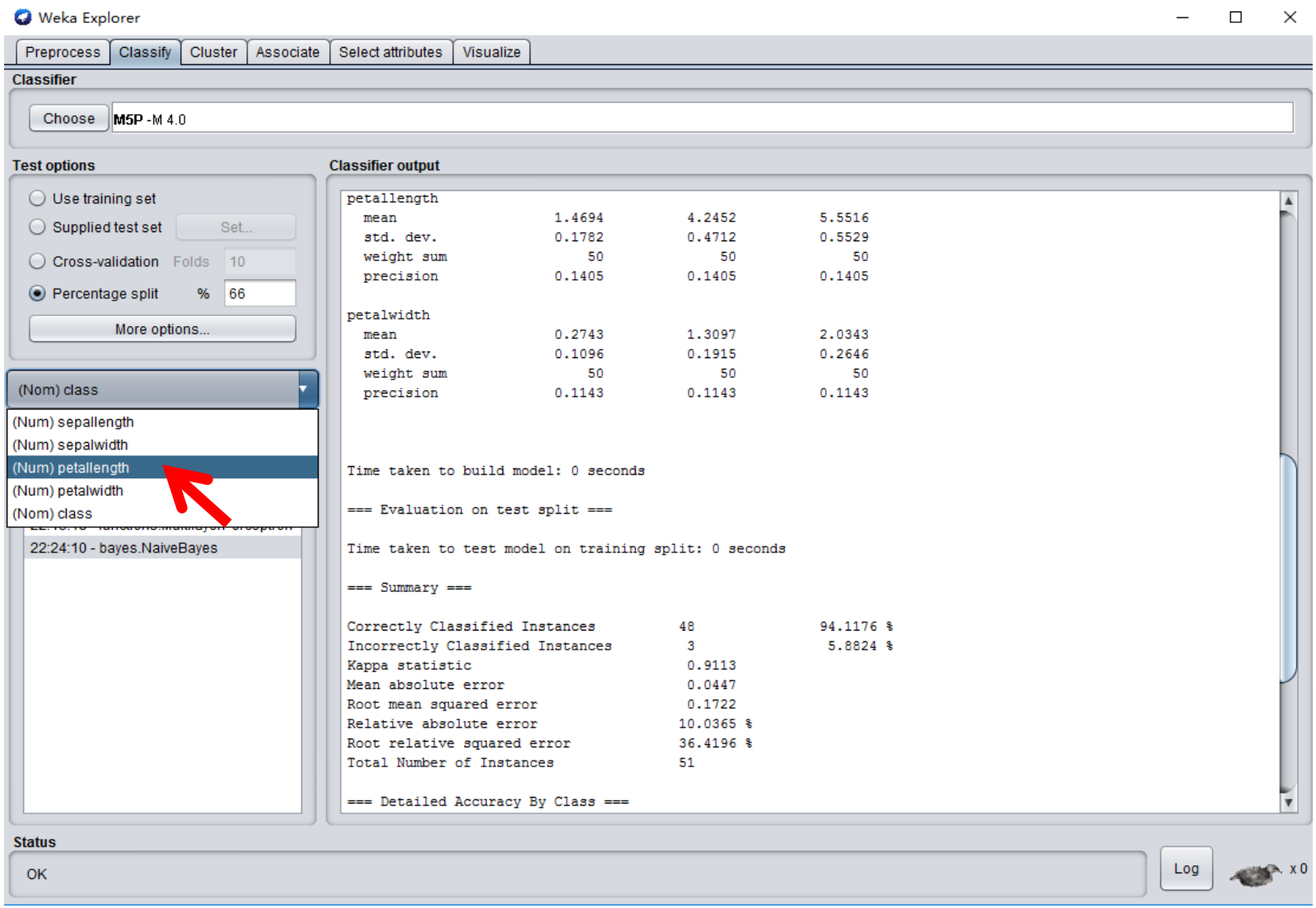
Explorer: Classification



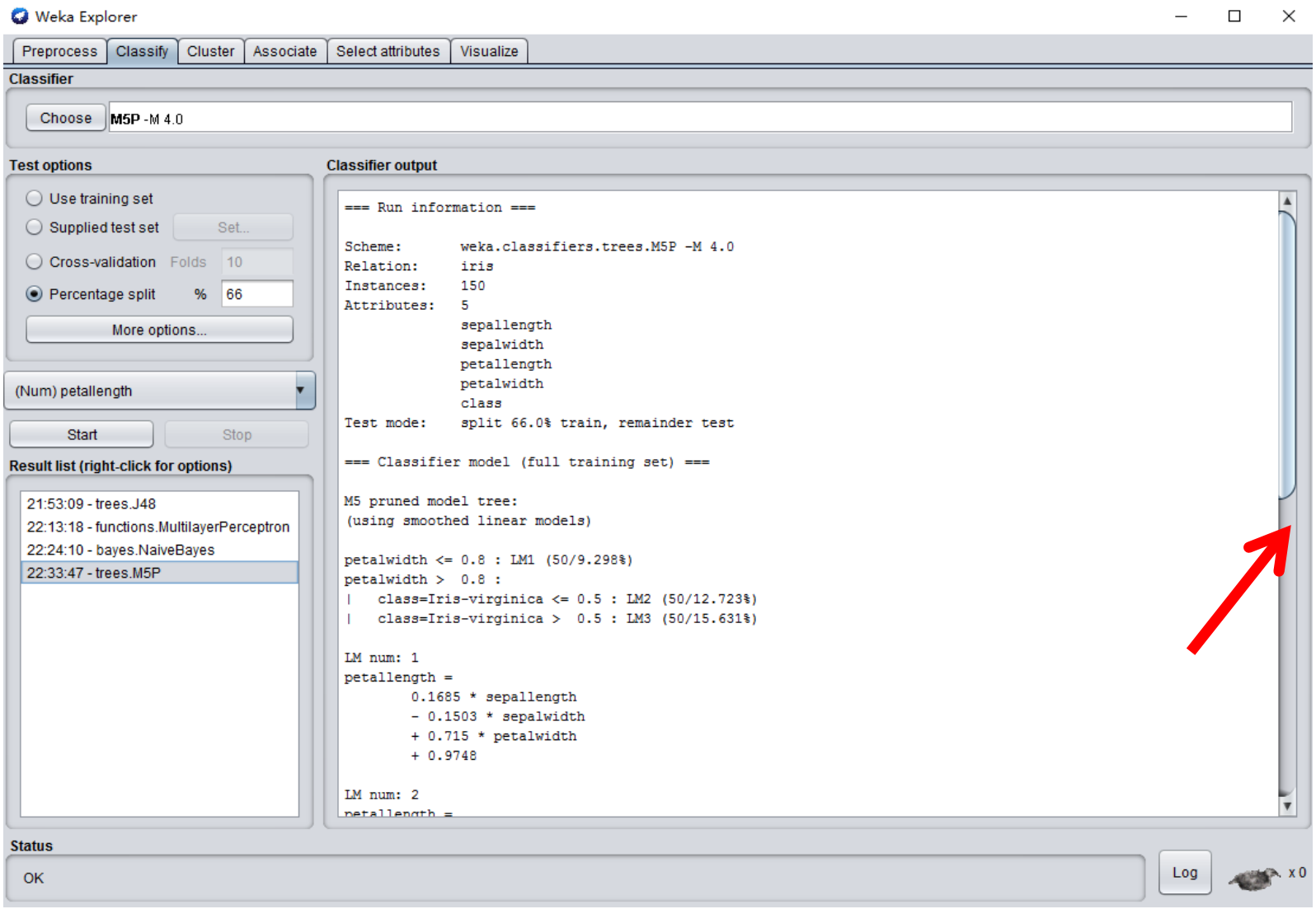
Explorer: Classification



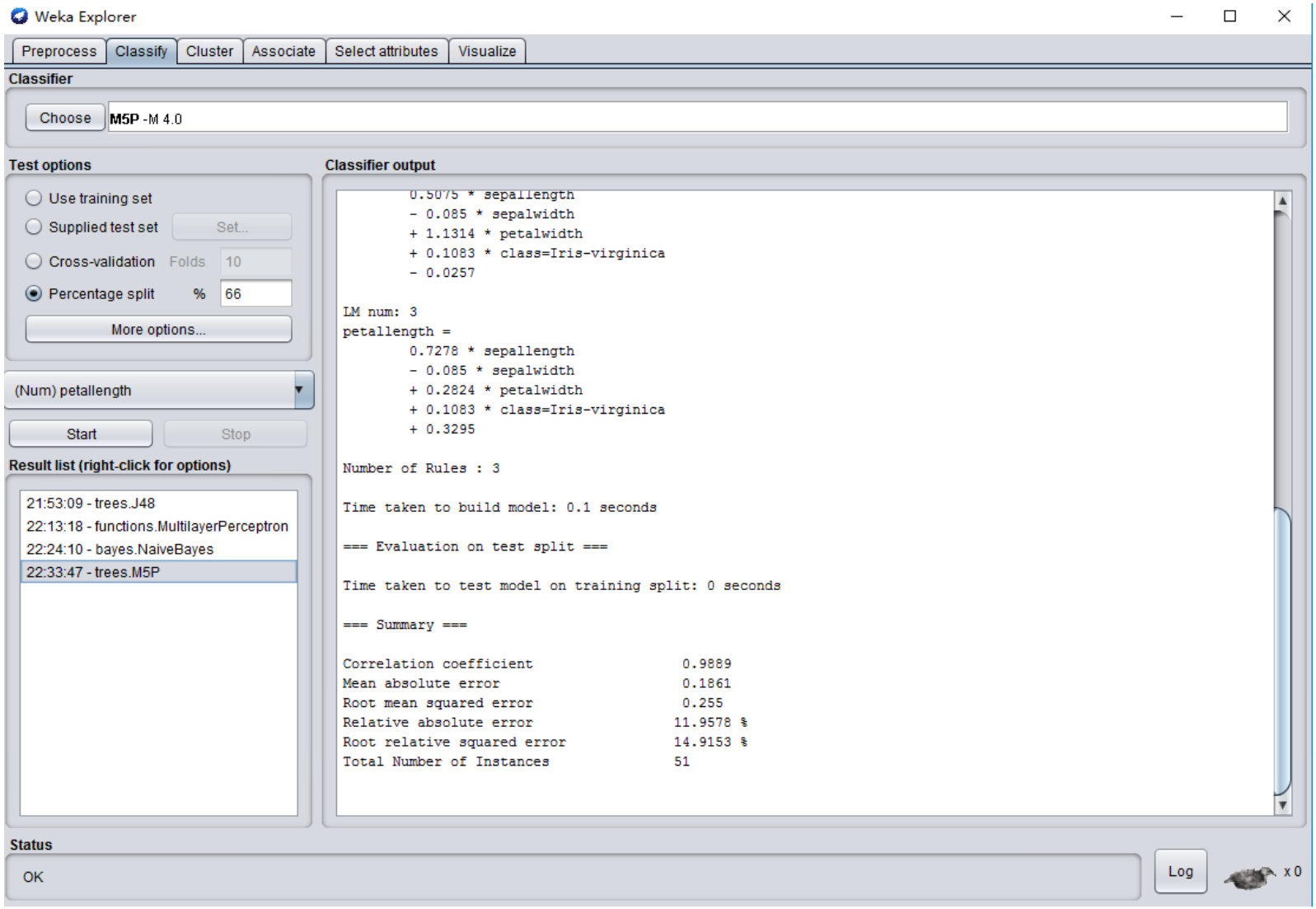
Explorer: Classification



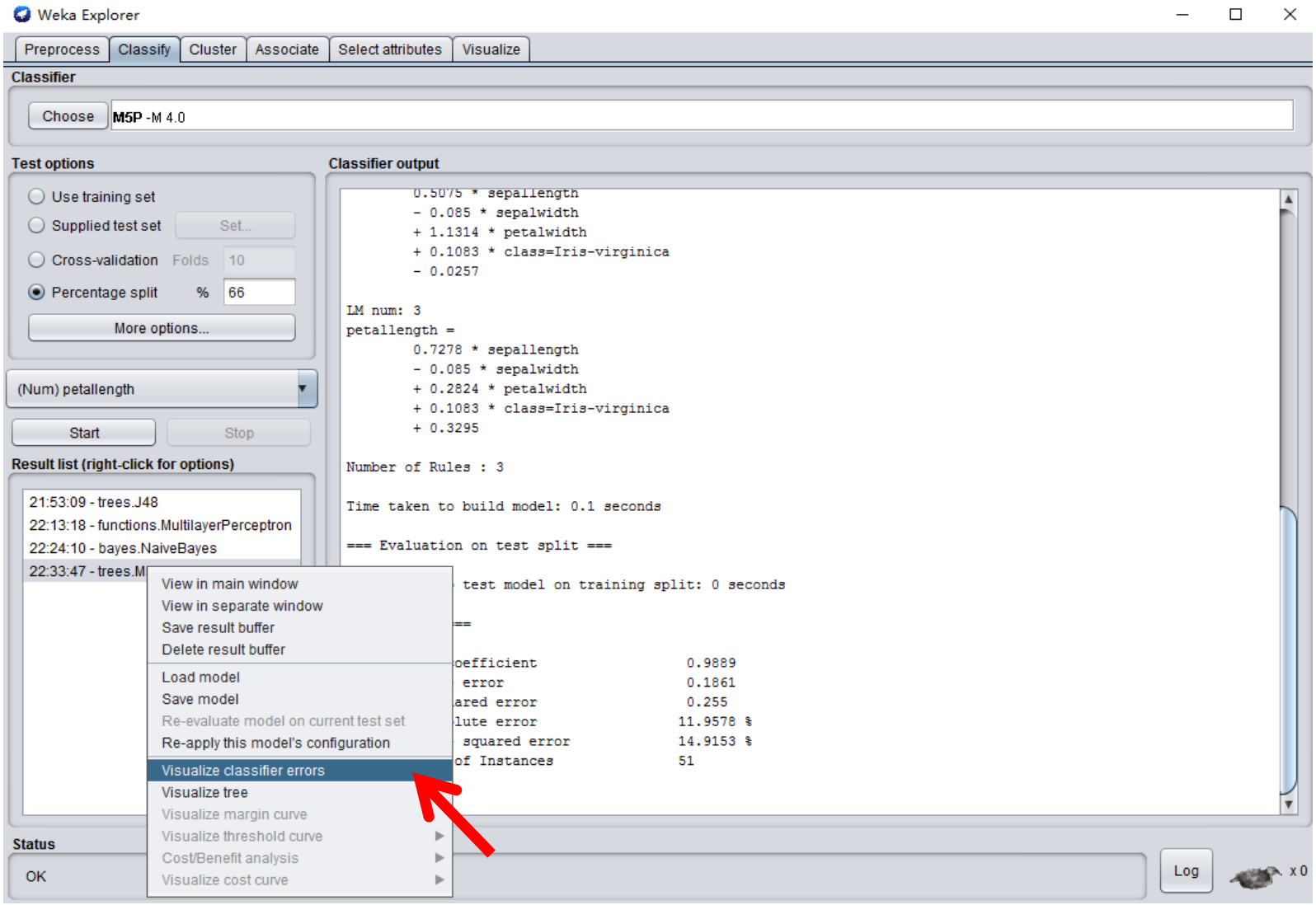
Explorer: Classification



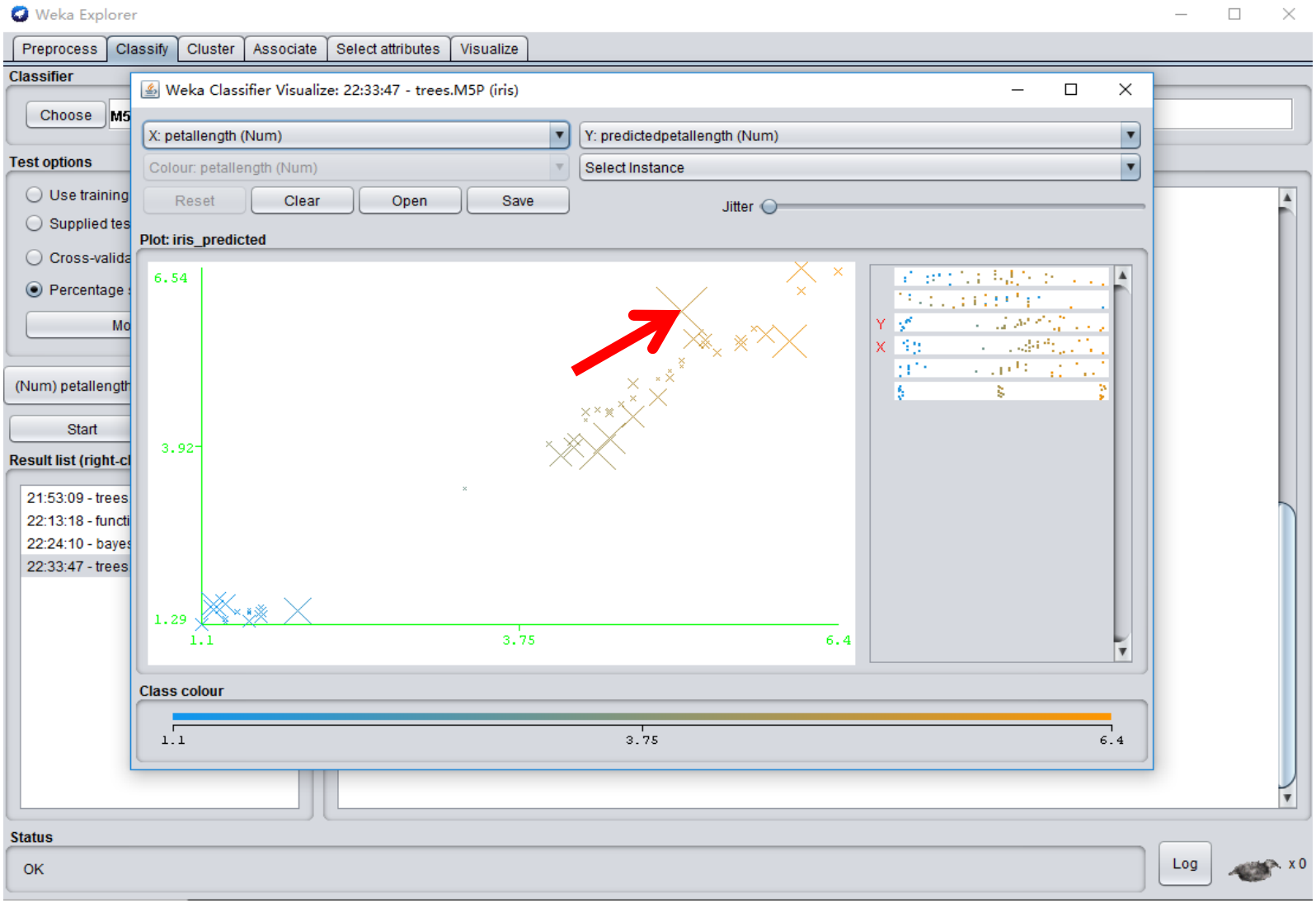
Explorer: Classification



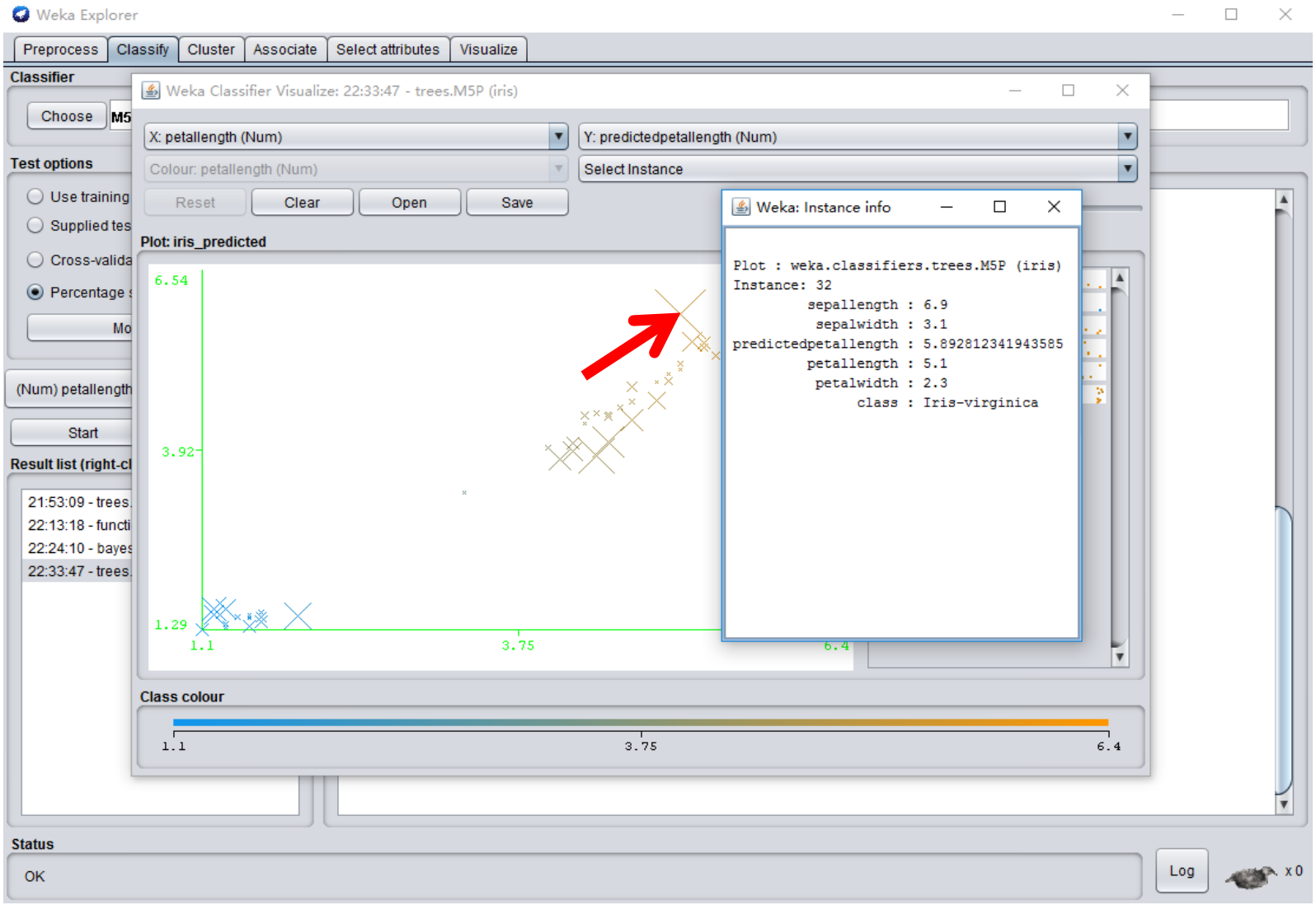
Explorer: Classification



Explorer: Classification



Explorer: Classification



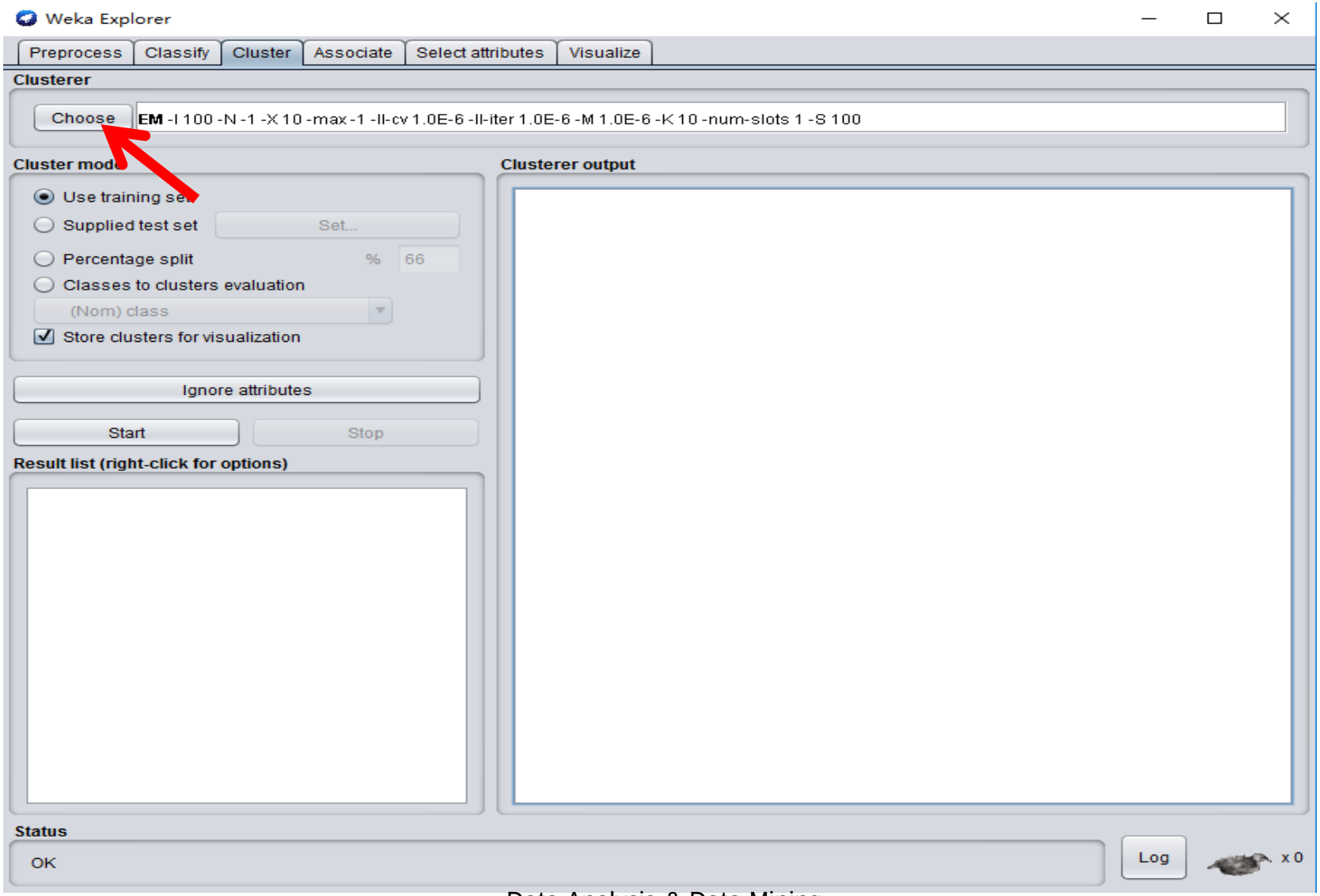
Roadmap: WEKA Usage

- WEKA Usage (version 3.8)
  - Explorer
    - Data preprocessing
    - Classification
    - Clustering
    - Association rule
    - Attribute selection (Feature selection)
    - Data visualization
  - Experimenter
  - Knowledge Flow

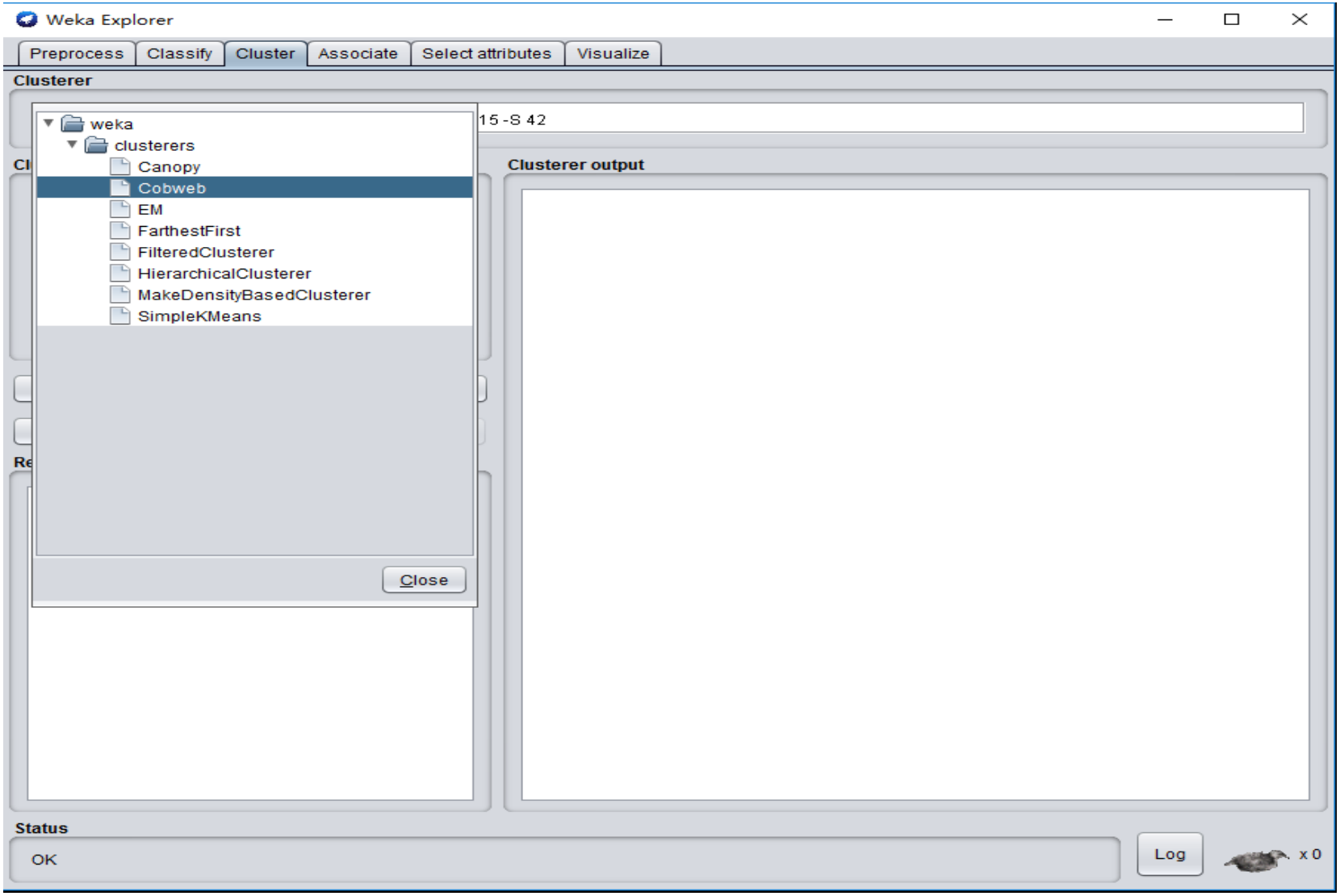
Explorer: Clustering Data

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

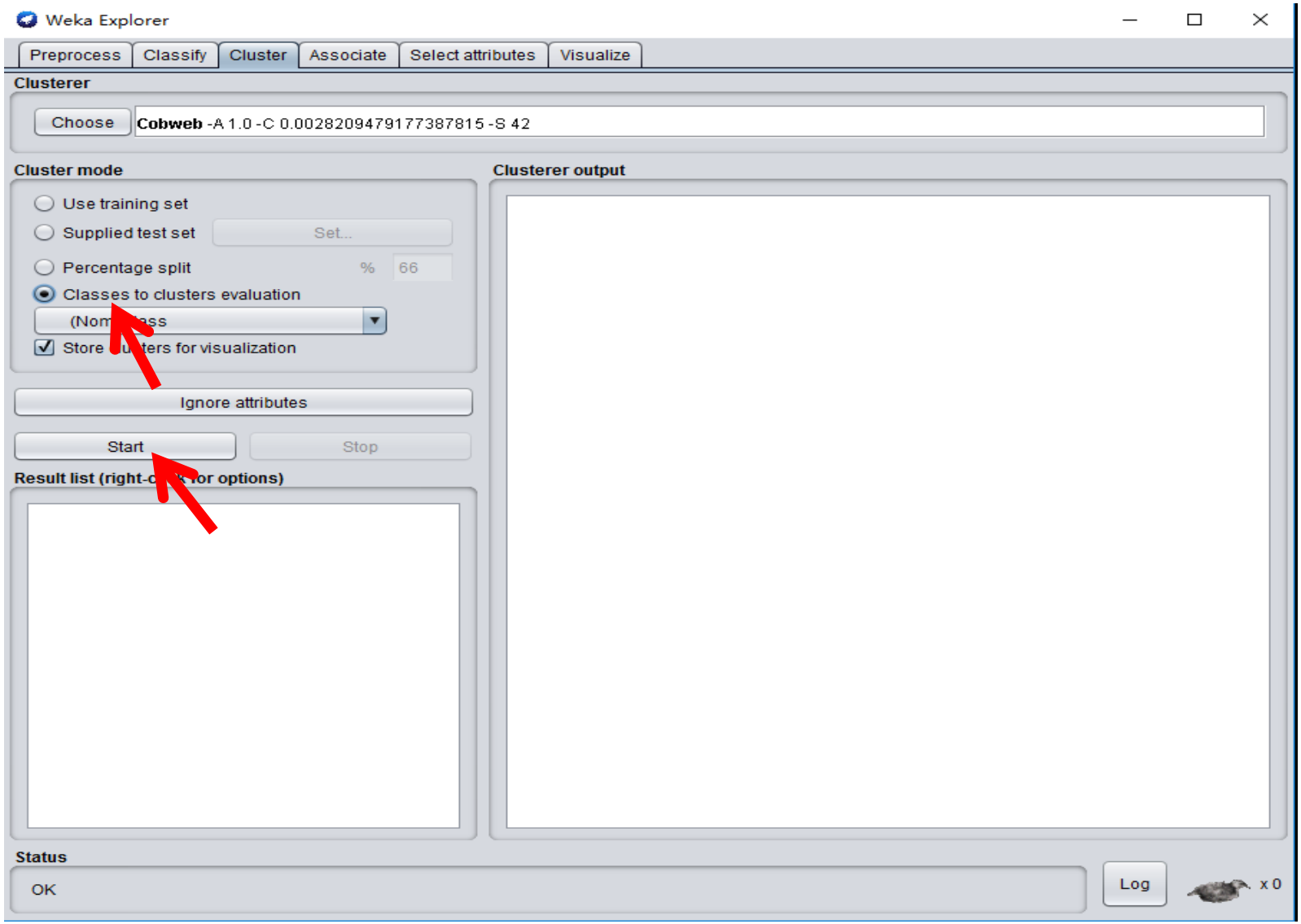
Explorer: Clustering



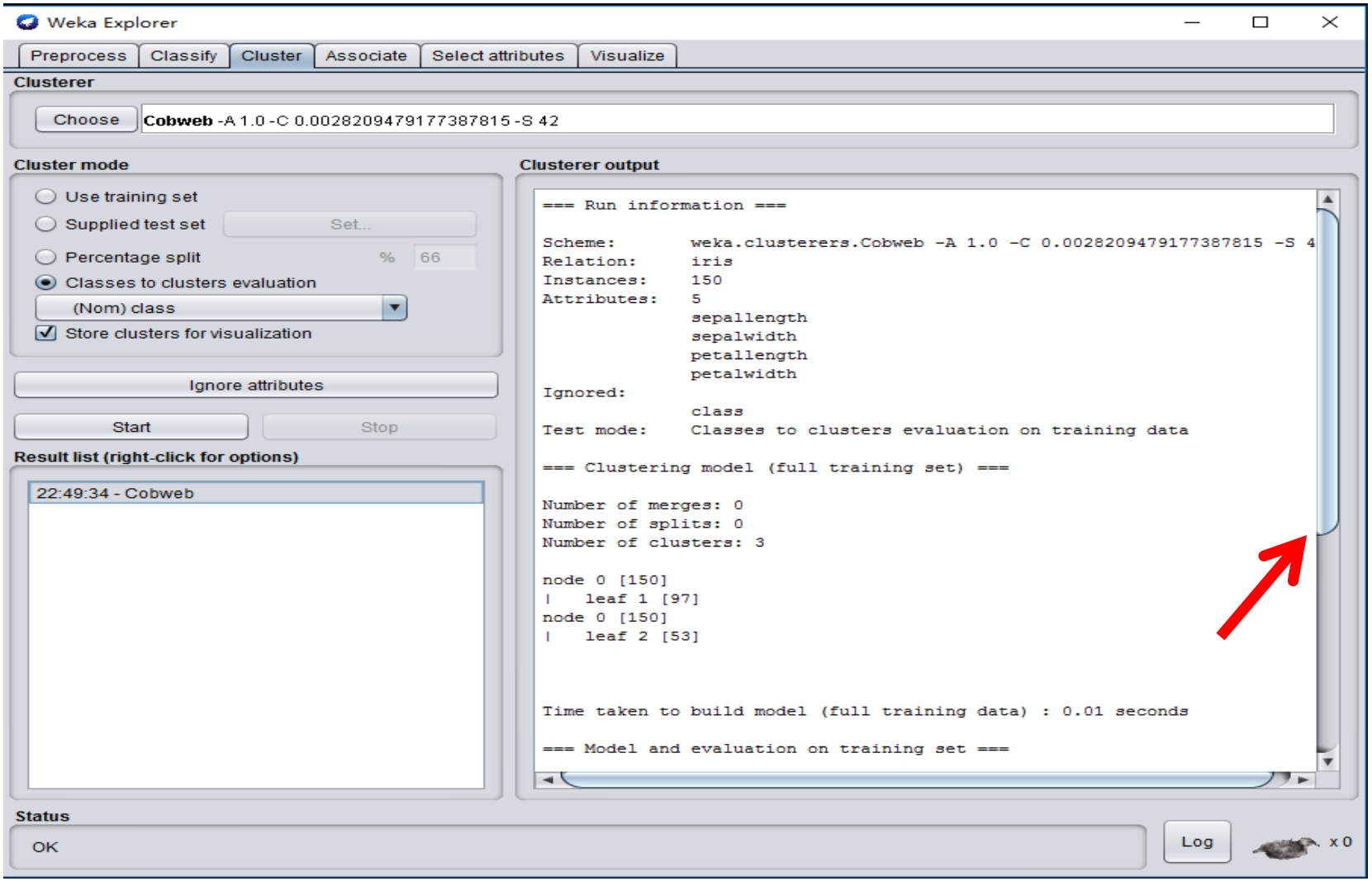
Explorer: Clustering



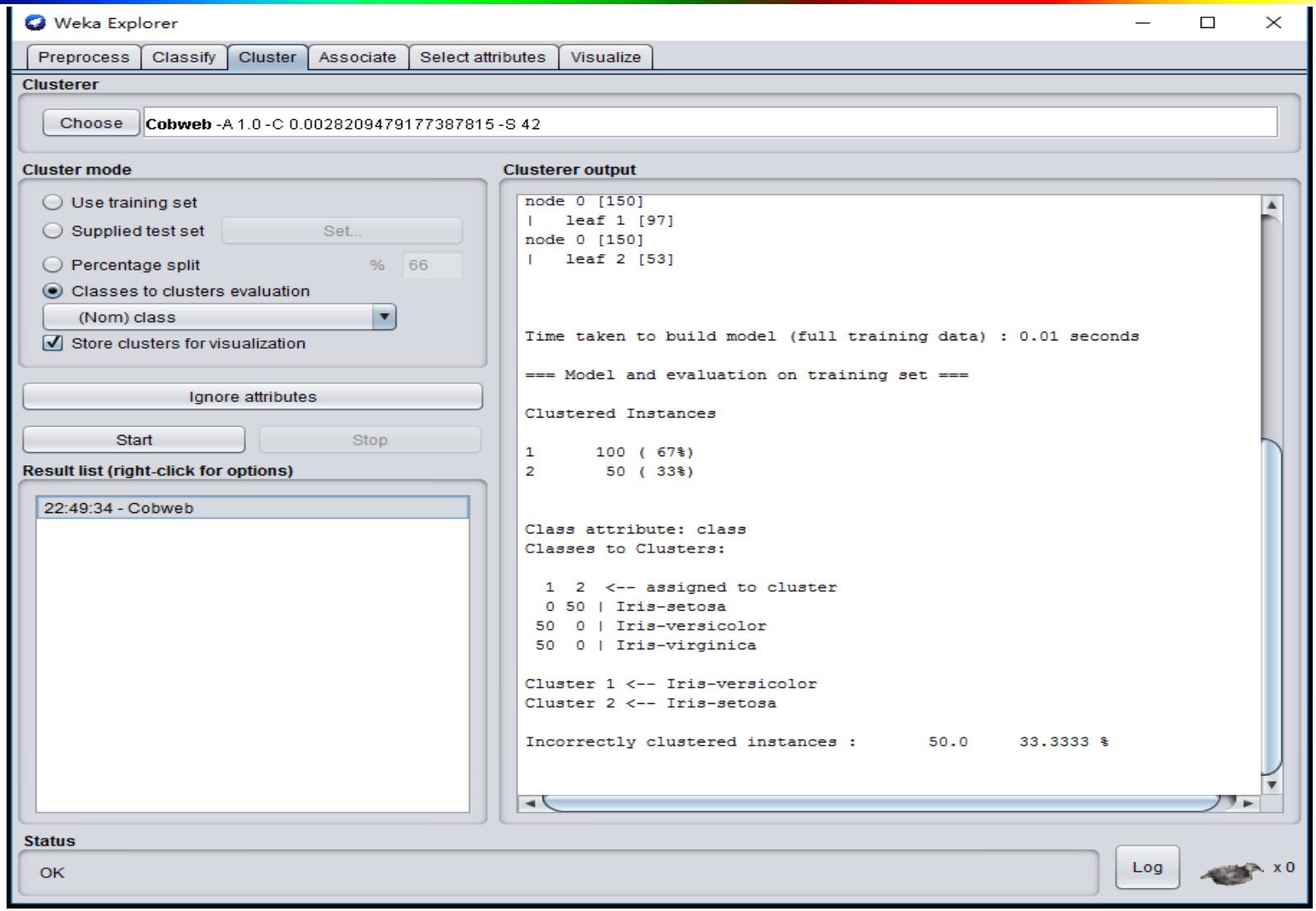
Explorer: Clustering



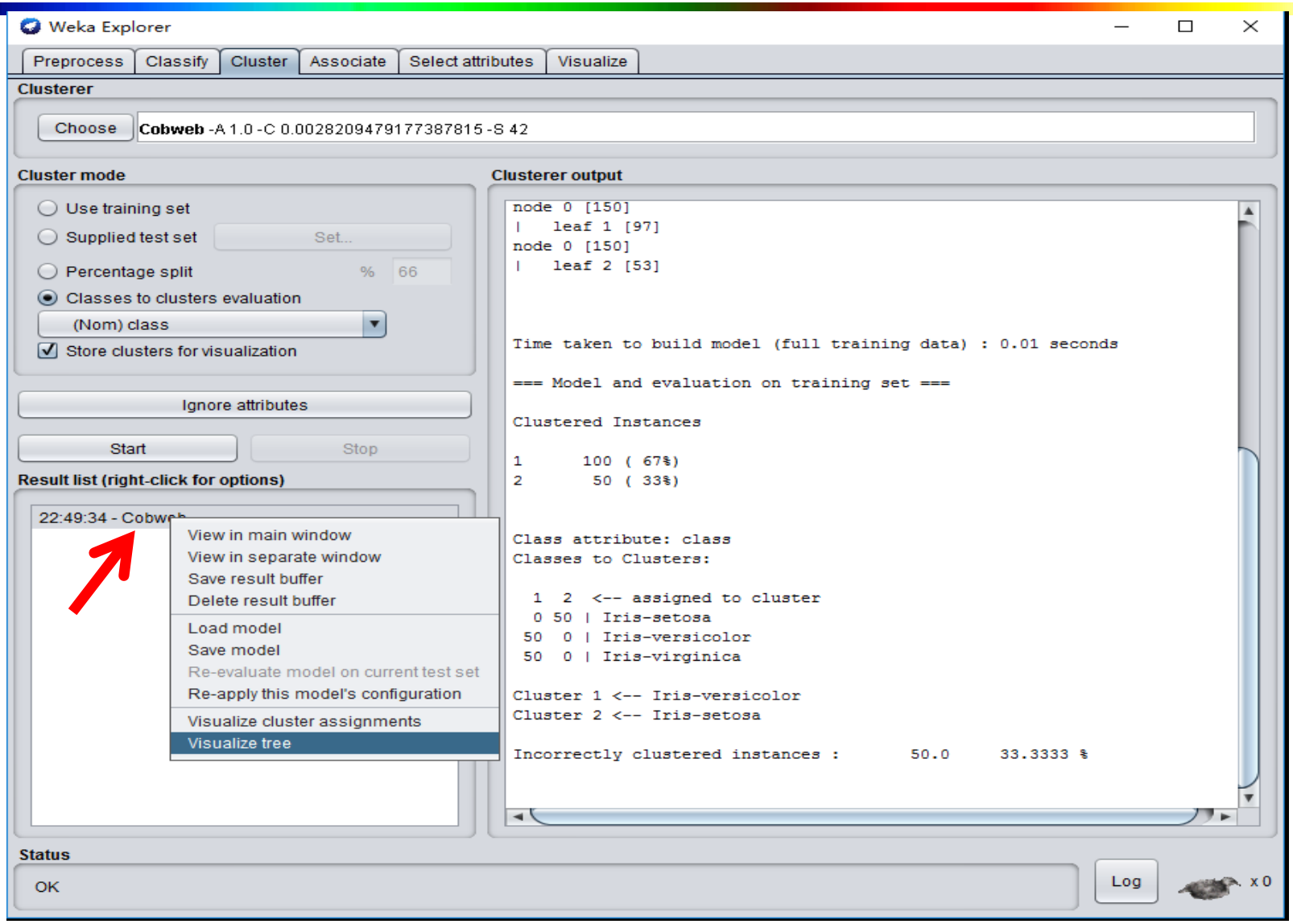
Explorer: Clustering



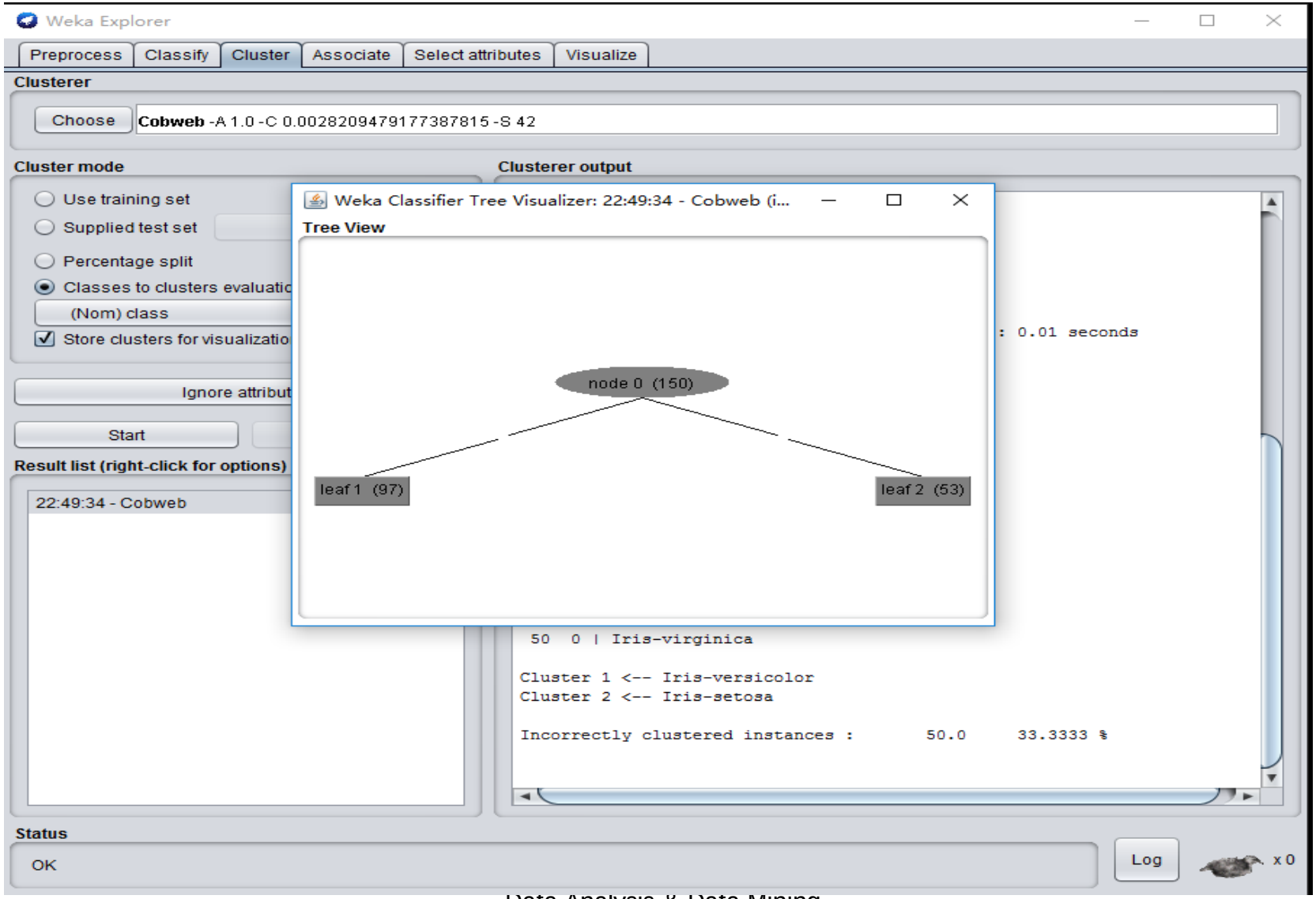
Explorer: Clustering



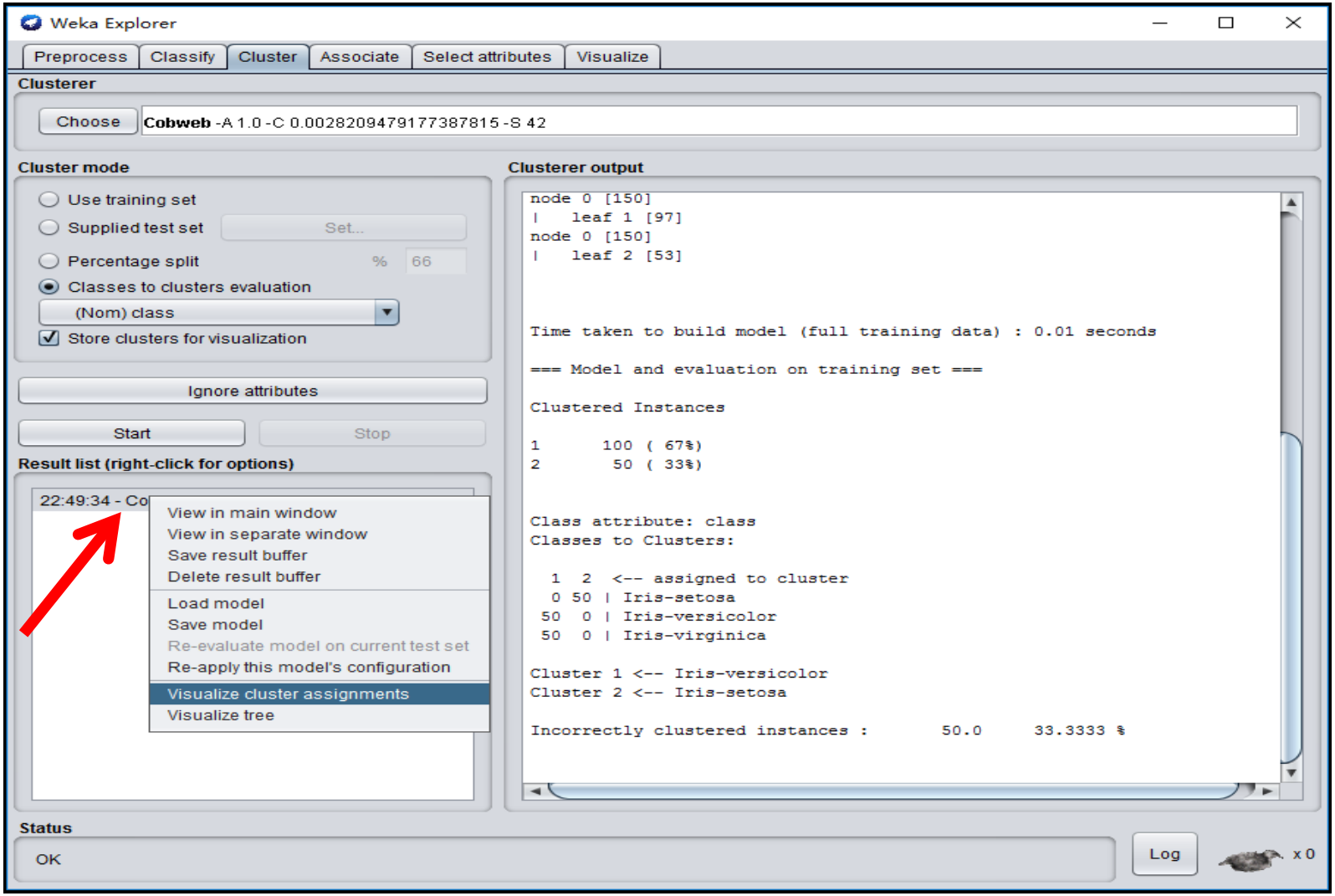
Explorer: Clustering



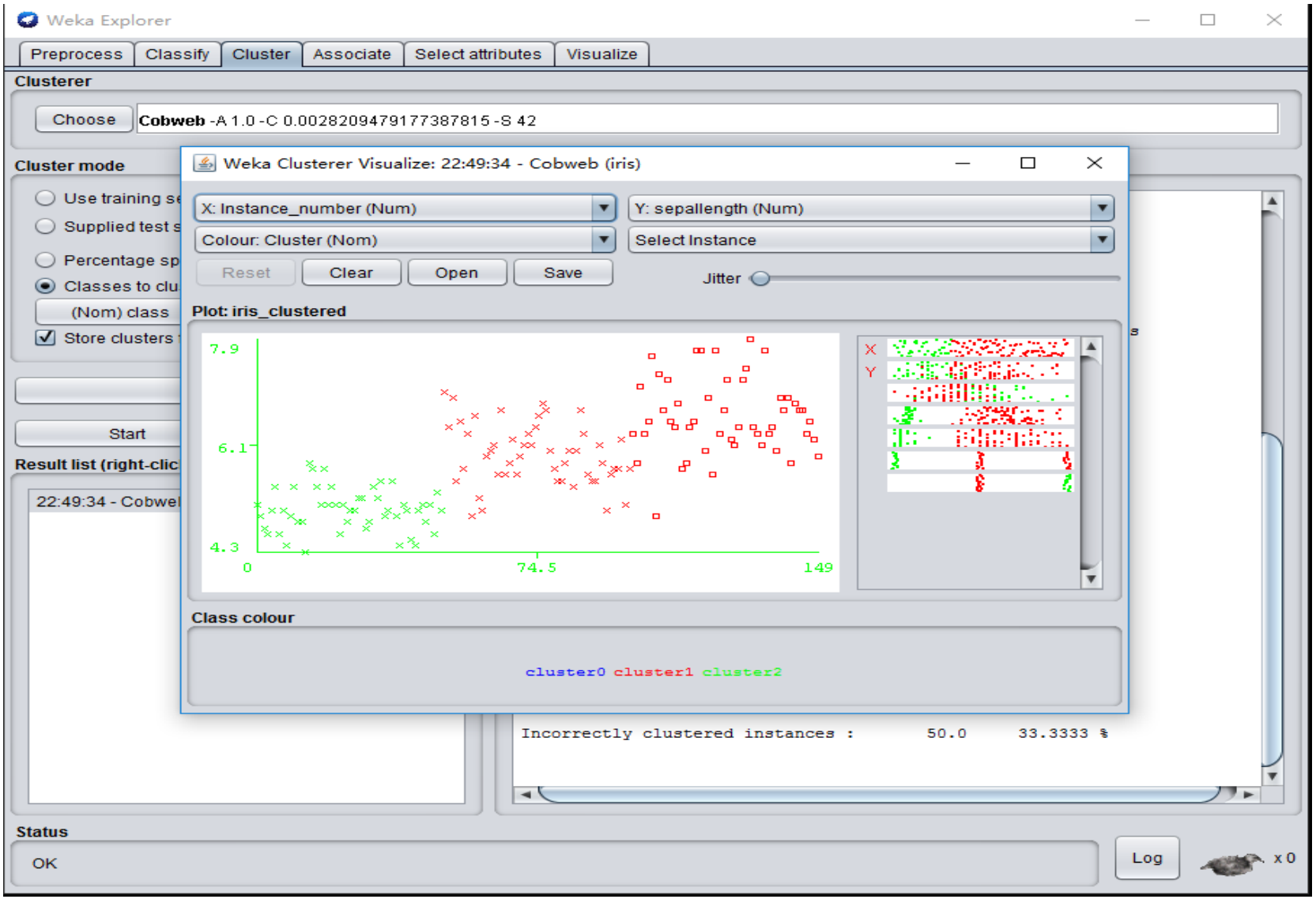
Explorer: Clustering



Explorer: Clustering



Explorer: Clustering



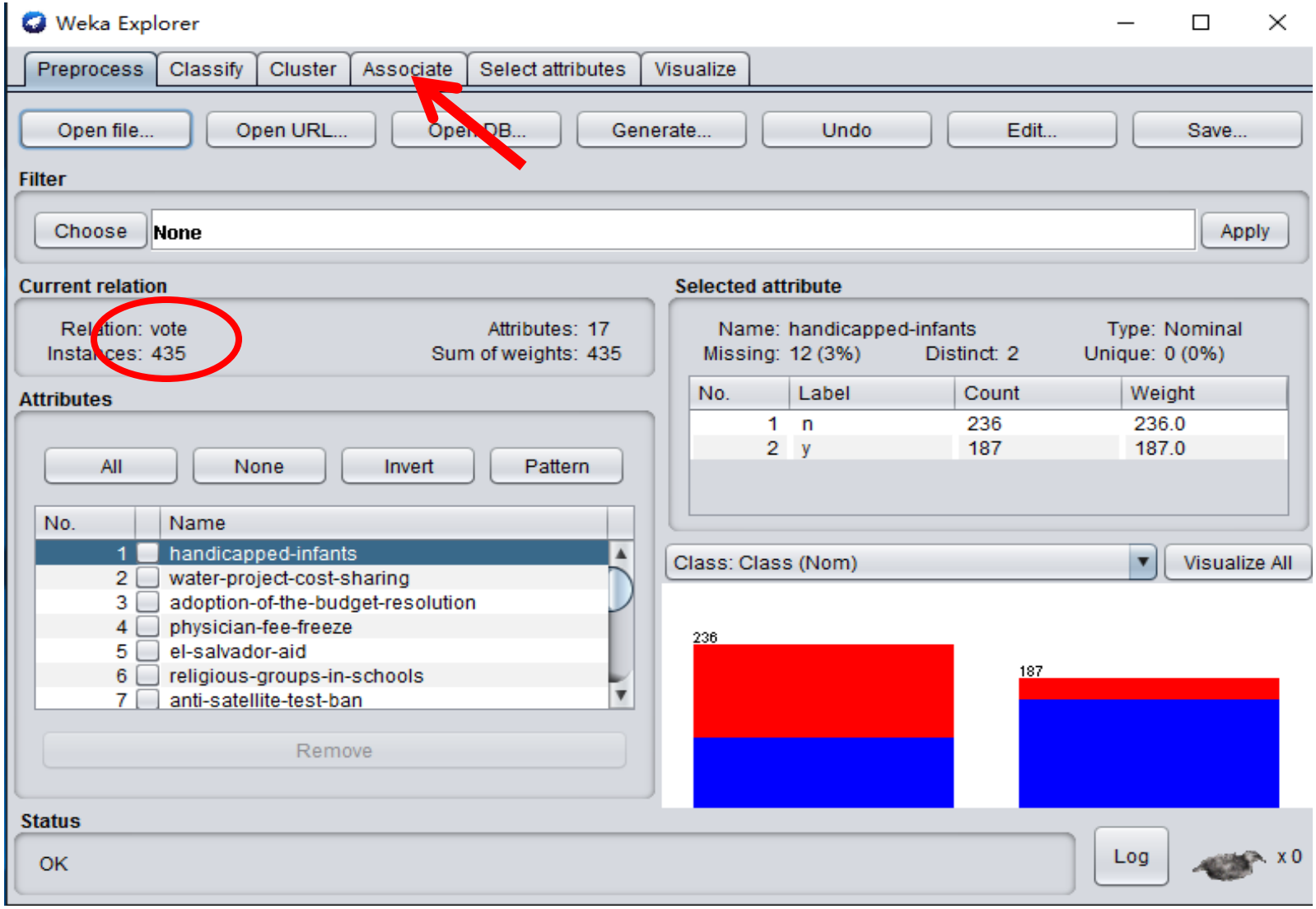
Roadmap: WEKA Usage

- WEKA Usage (version 3.8)
  - Explorer
    - Data preprocessing
    - Classification
    - Clustering
    - Association rule
    - Attribute selection (Feature selection)
    - Data visualization
  - Experimenter
  - Knowledge Flow

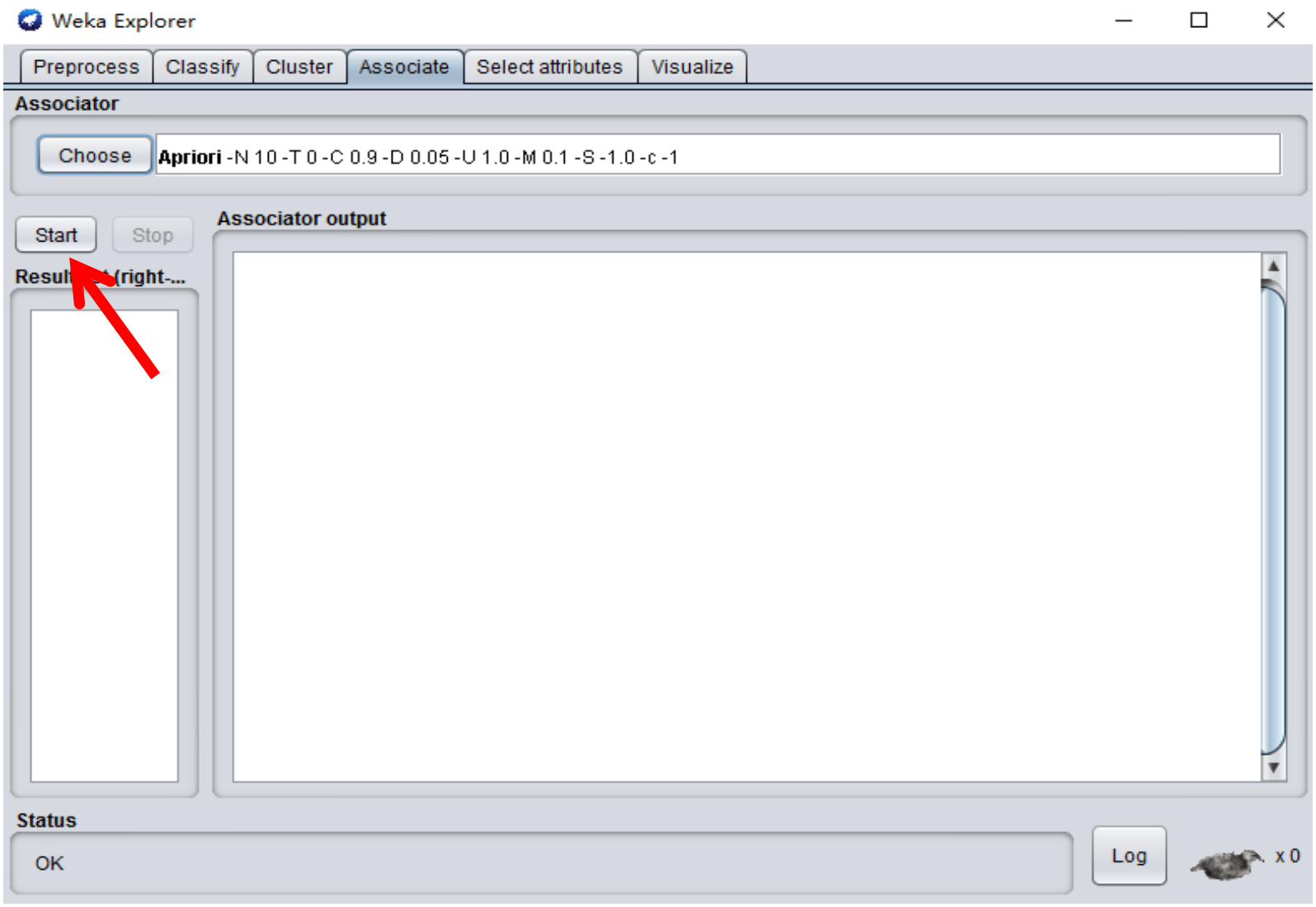
Explorer: Finding Associations

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

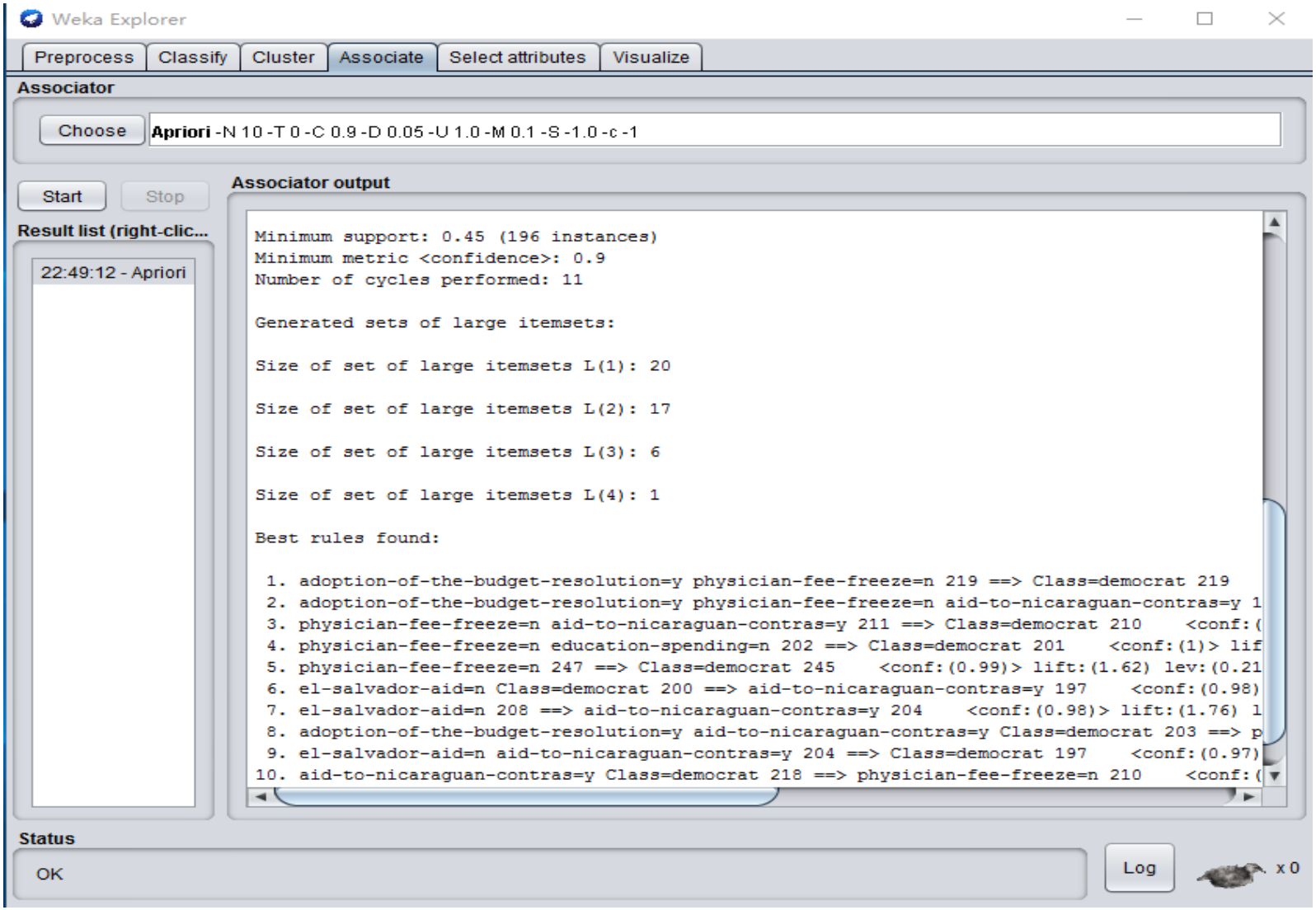
Explorer: Association Rule



Explorer: Association Rule



Explorer: Association Rule



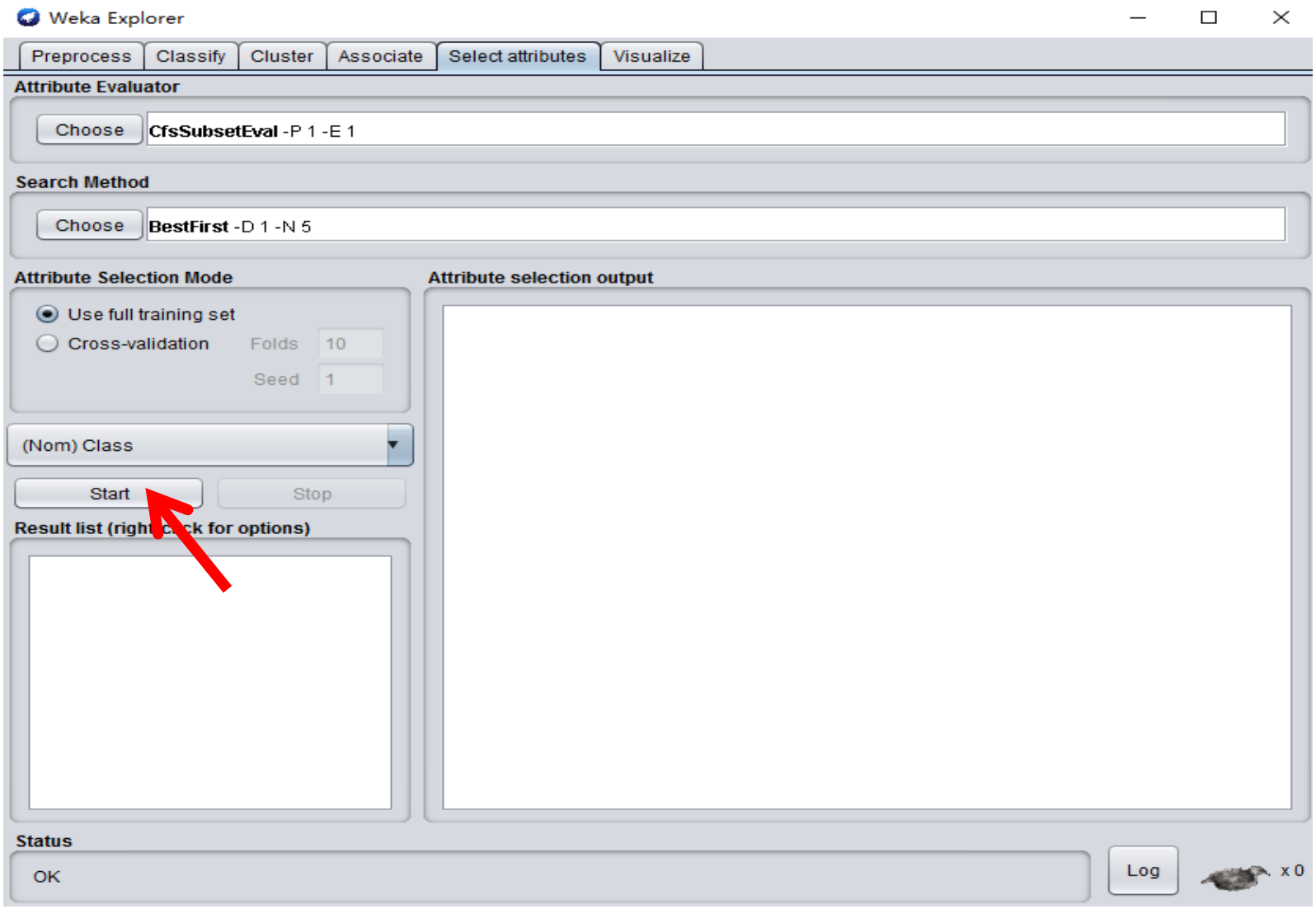
Roadmap: WEKA Usage

- WEKA Usage (version 3.8)
  - Explorer
    - Data preprocessing
    - Classification
    - Clustering
    - Association rule
    - Attribute selection (Feature selection)
    - Data visualization
  - Experimenter
  - Knowledge Flow

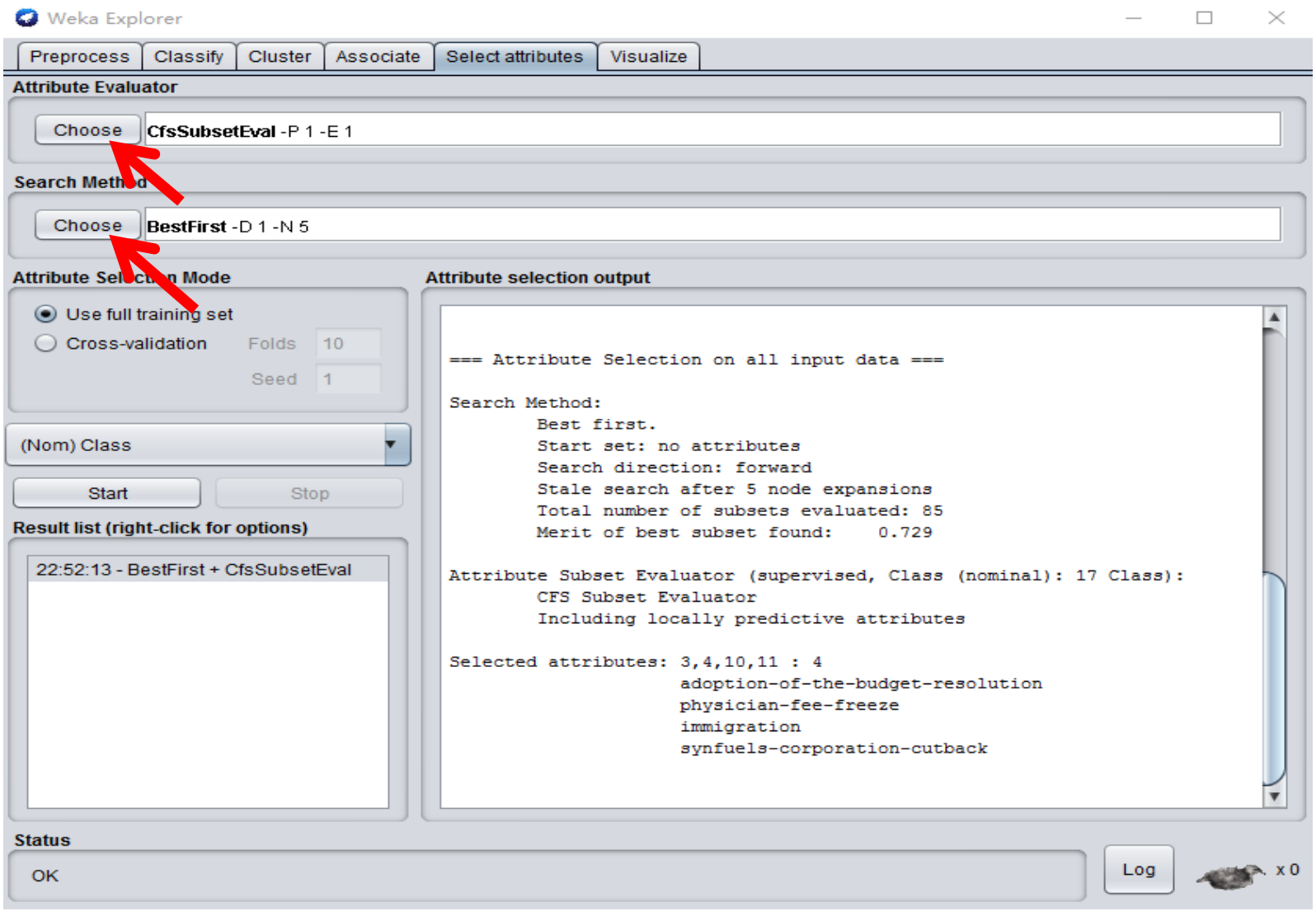
Explorer: Attribute Selection

- Panel that can be used to investigate which (subsets of) attributes are the most predictive ones
- Attribute selection methods contain two parts:
  - A search method: best-first, forward selection, random, exhaustive, genetic algorithm, ranking
  - An evaluation method: correlation-based, wrapper, information gain, chi-square, ...
- Very flexible: WEKA allows (almost) arbitrary combinations of these two

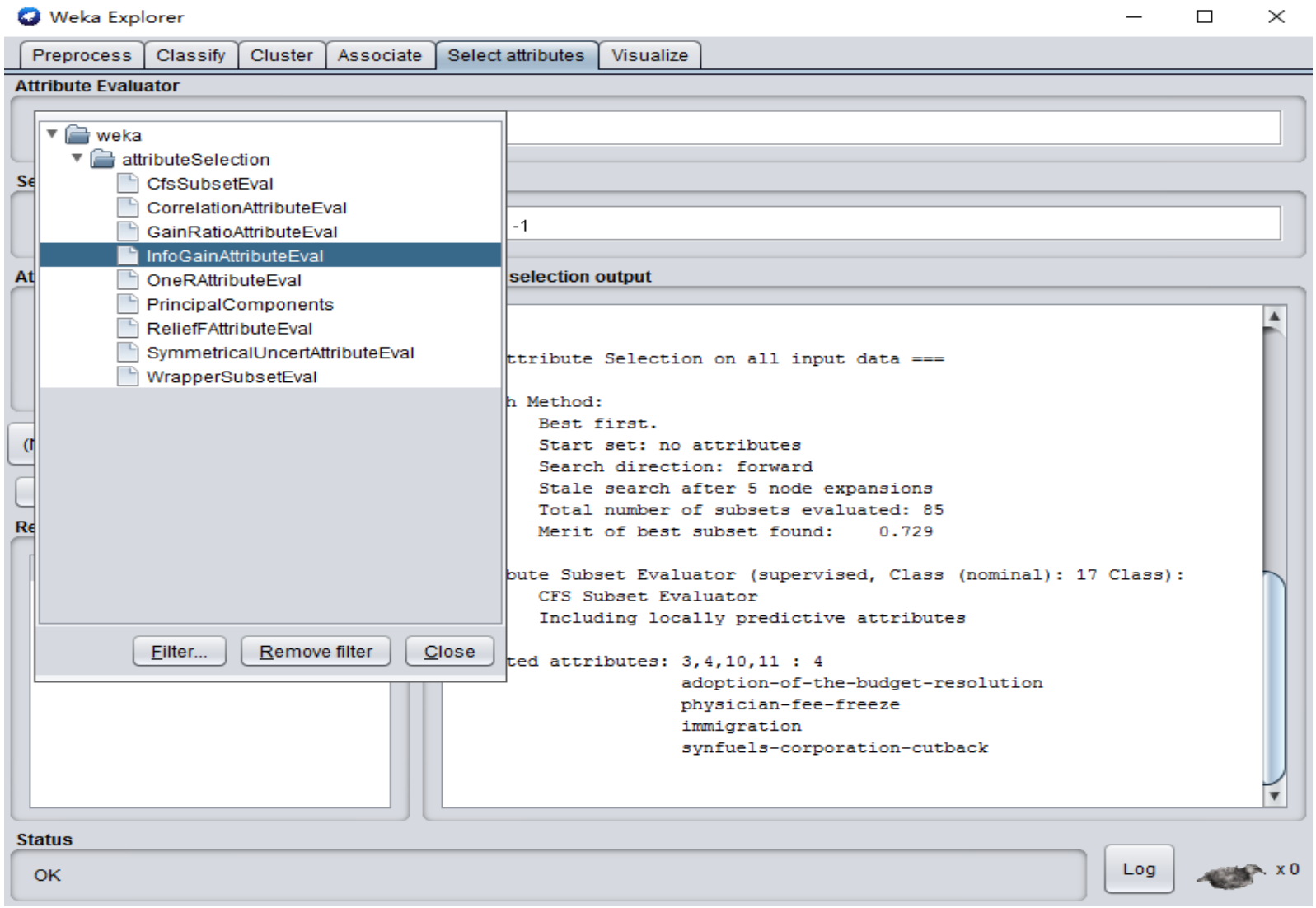
Explorer: Attribute Selection



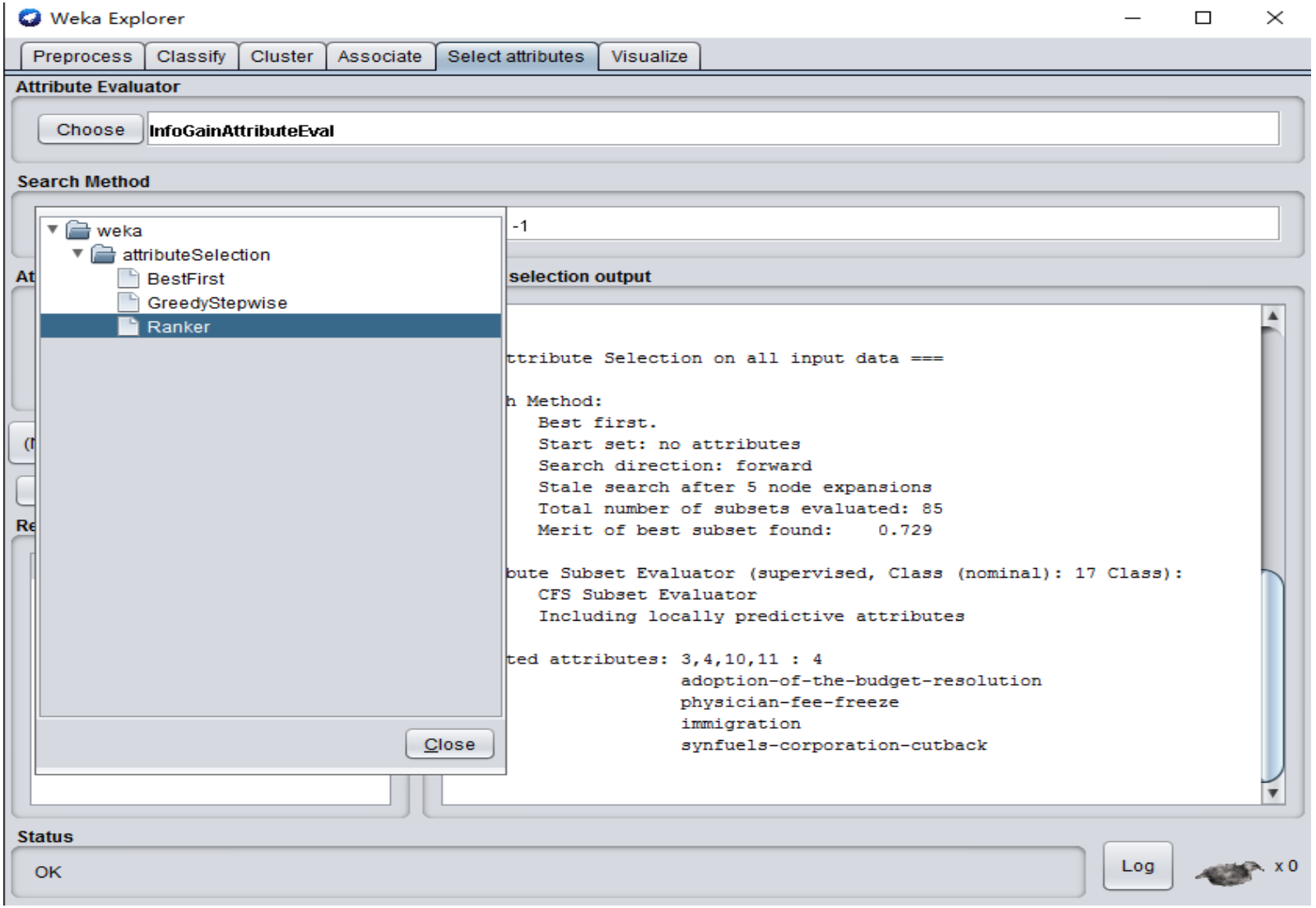
Explorer: Attribute Selection



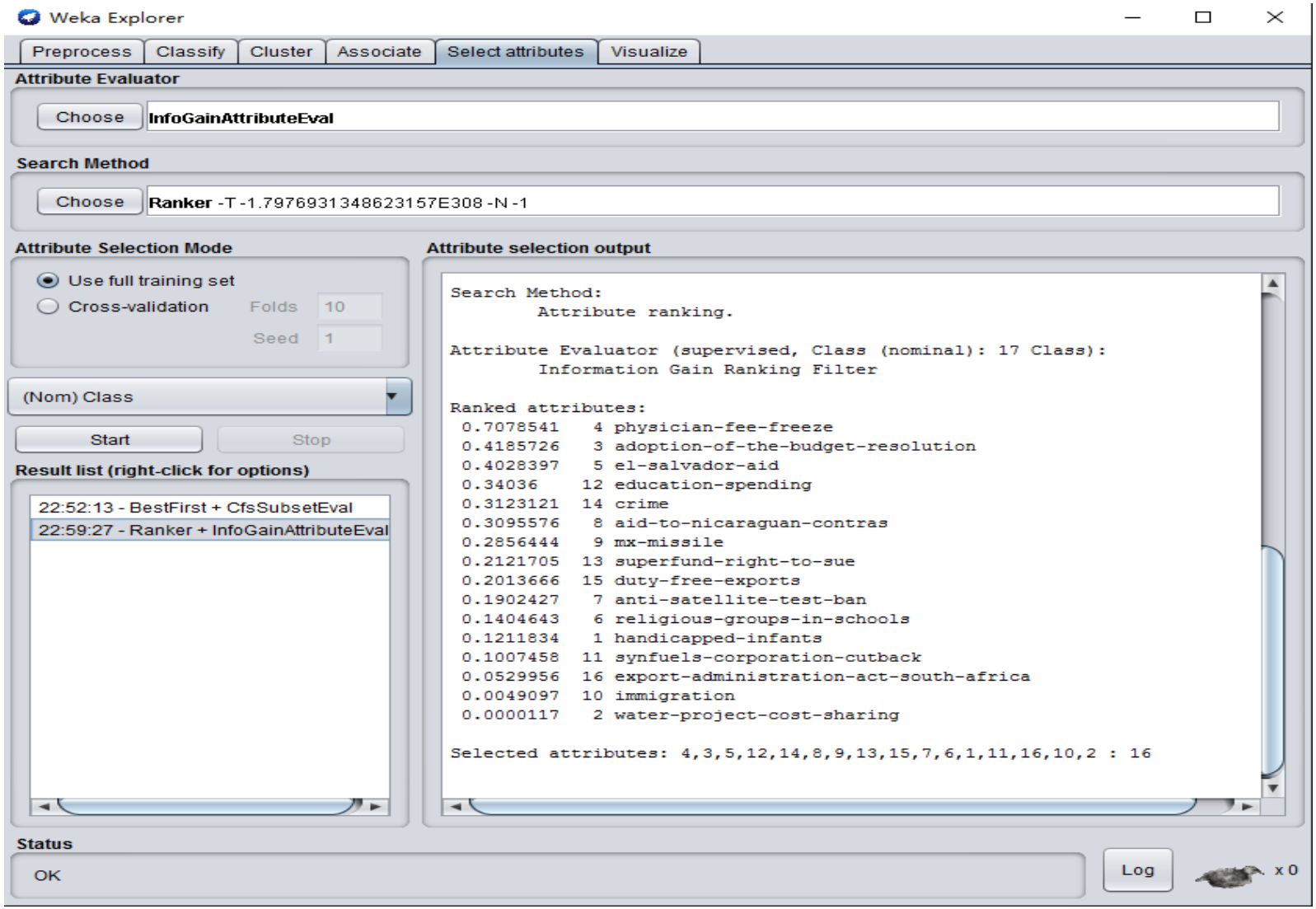
Explorer: Attribute Selection



Explorer: Attribute Selection



Explorer: Attribute Selection



Roadmap: WEKA Usage

- WEKA Usage (version 3.8)
  - Explorer
    - Data preprocessing
    - Classification
    - Clustering
    - Association rule
    - Attribute selection (Feature selection)
    - Data visualization
  - Experimenter
  - Knowledge Flow

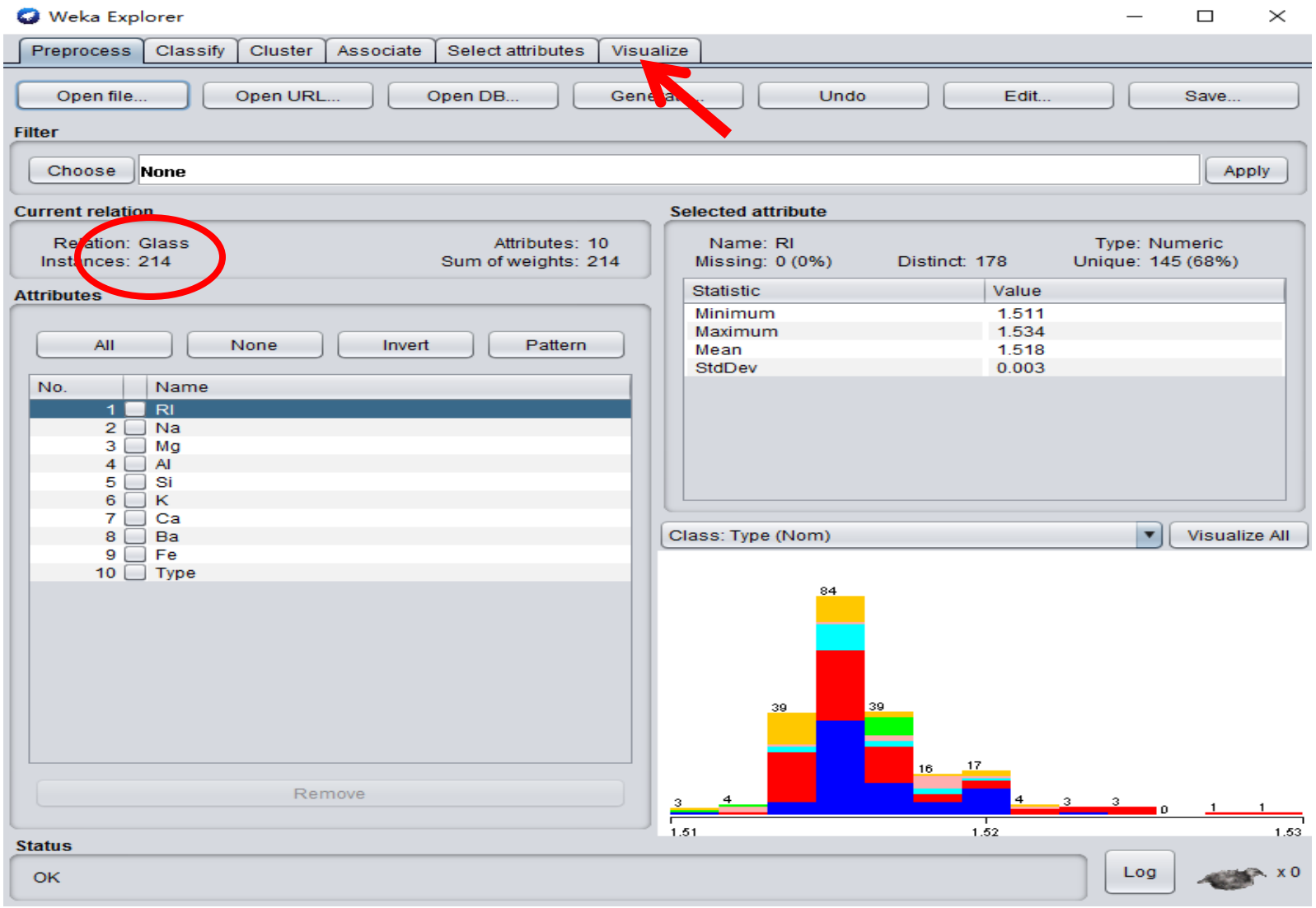
Explorer: Data Visualization

- So how do visualization helps in mining ?
  - Displayed are scatter plots.
  - It helps answer the questions regarding the involved variables X & Y such as
    - Related?
    - Positively correlated?
    - Negatively correlated?
    - Does Y's variation depends X's variation?
    - Are there outliers ?

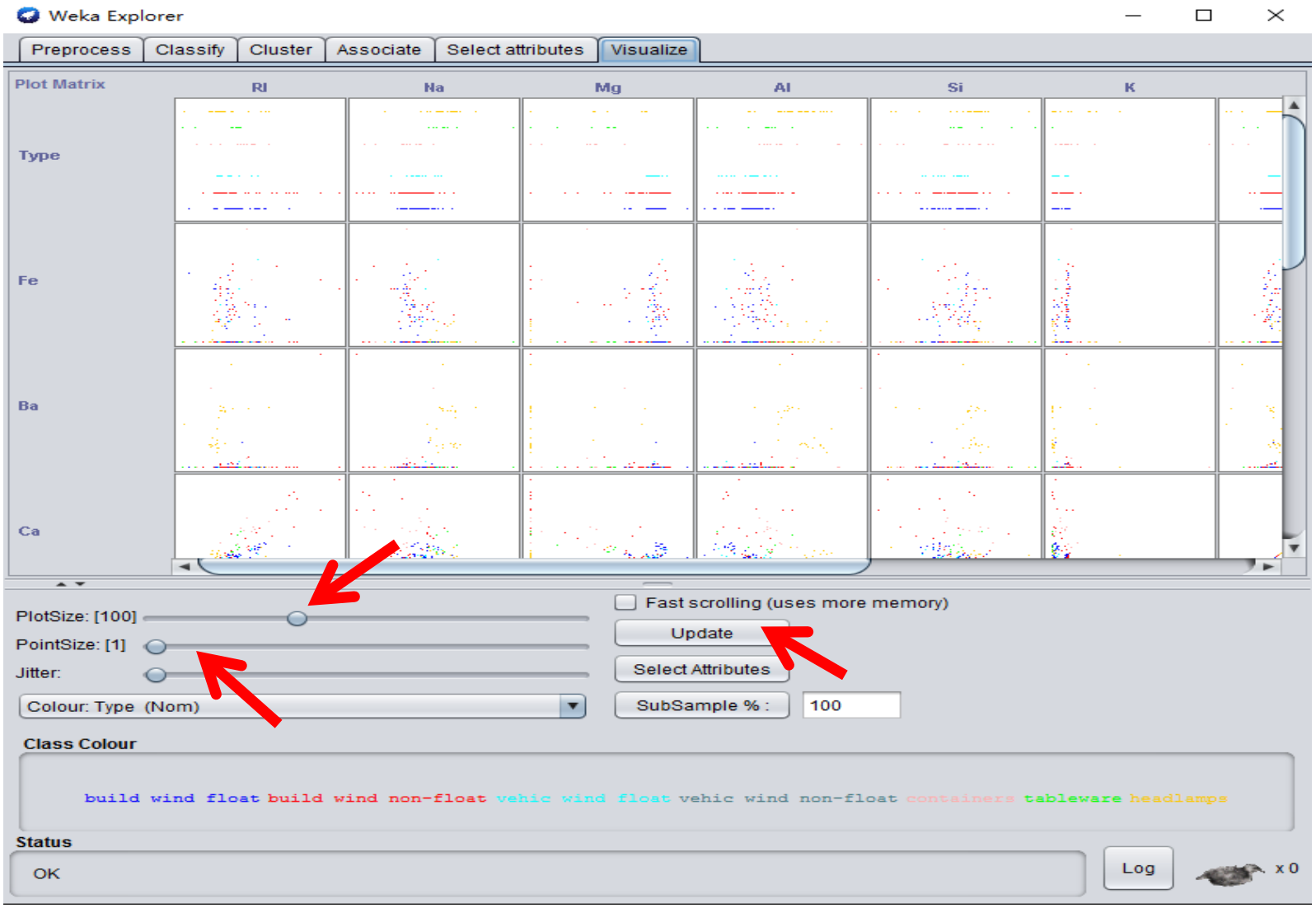
Explorer: Data Visualization

- Visualization very useful in practice: e.g. helps to determine difficulty of the learning problem
- WEKA can visualize single attributes (1-d) and pairs of attributes (2-d)
  - To do: rotating 3-d visualizations (Xgobi-style)
- Color-coded class values
- “Jitter” option to deal with nominal attributes (and to detect “hidden” data points)
- “Zoom-in” function

Explorer: Data Visualization



Explorer: Data Visualization

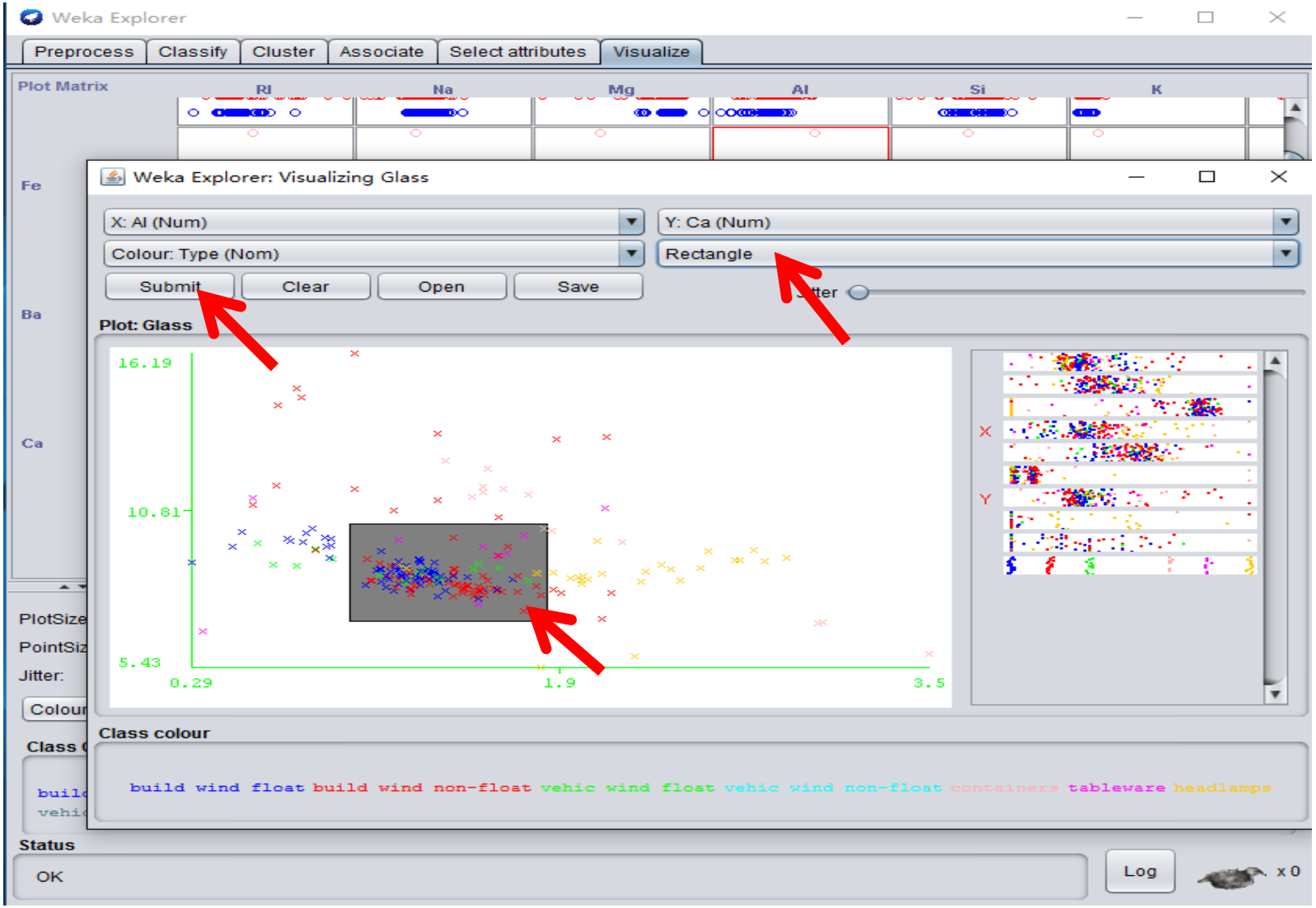


Explorer: Data Visualization

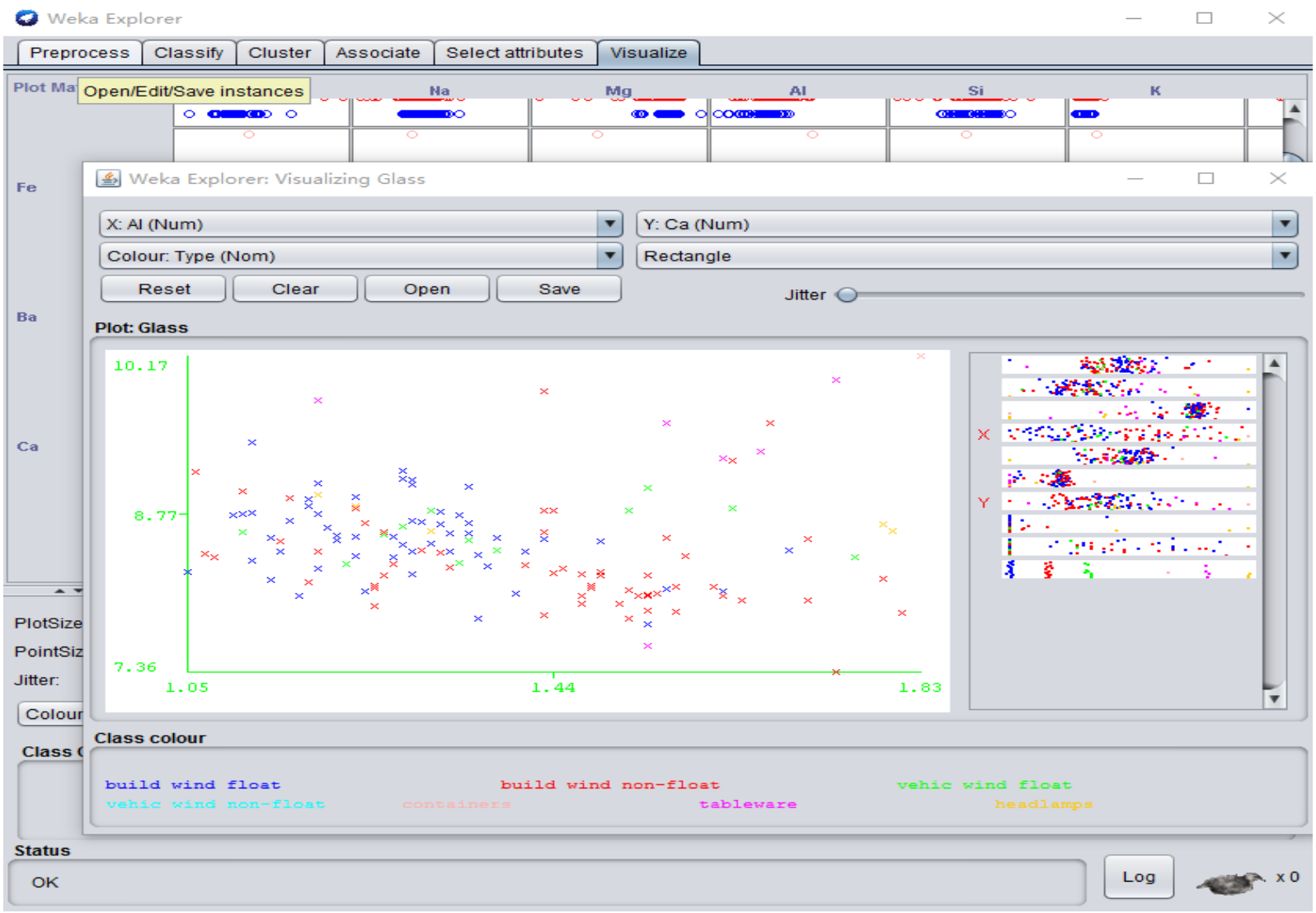
Explorer: Data Visualization

Explorer: Data Visualization

Explorer: Data Visualization

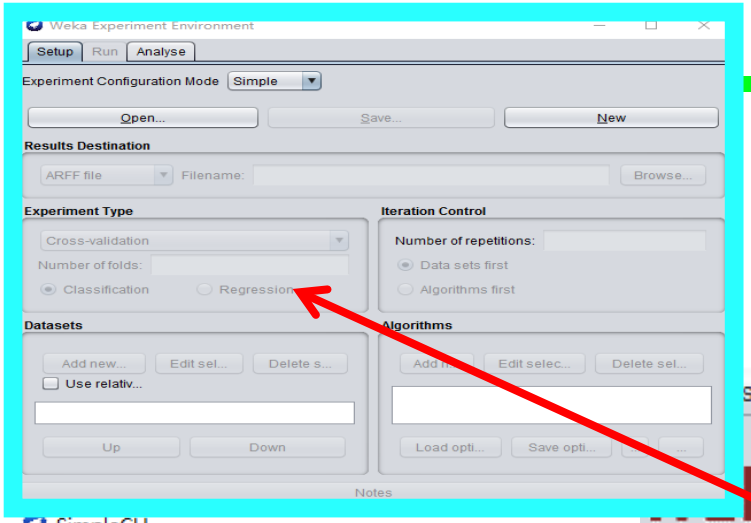


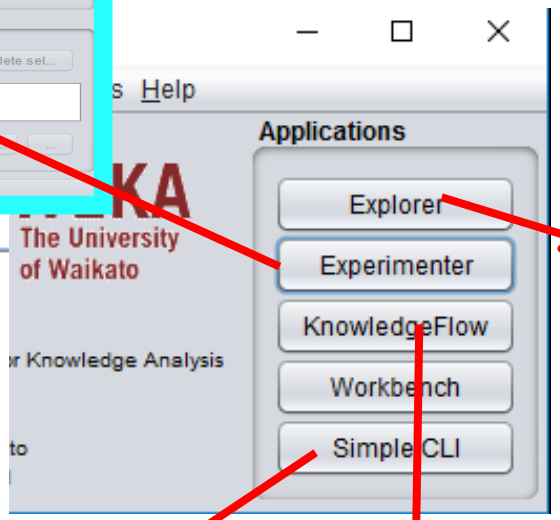
Explorer: Data Visualization

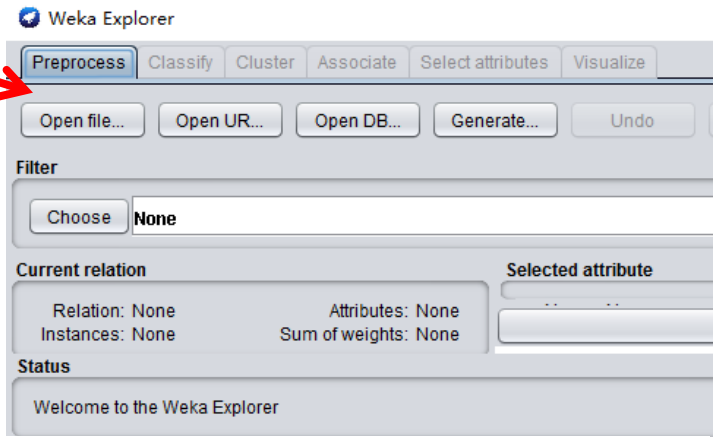


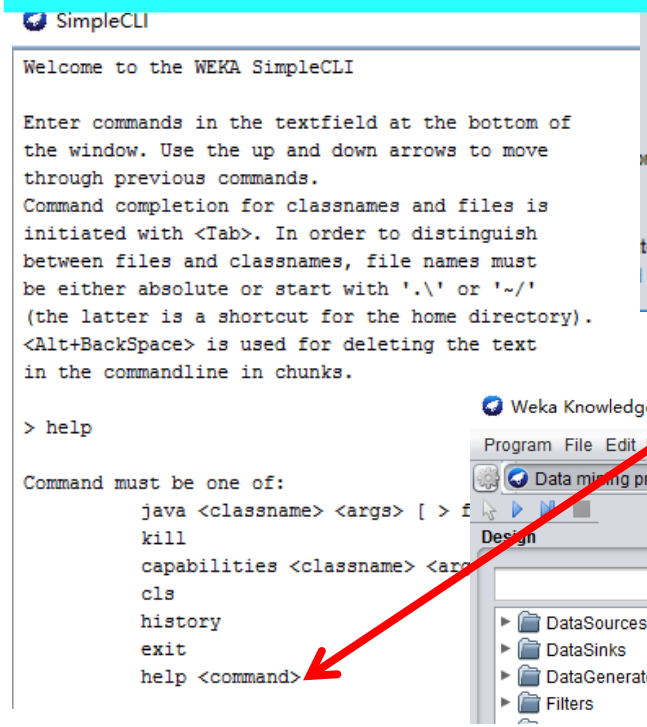
WEKA窗口

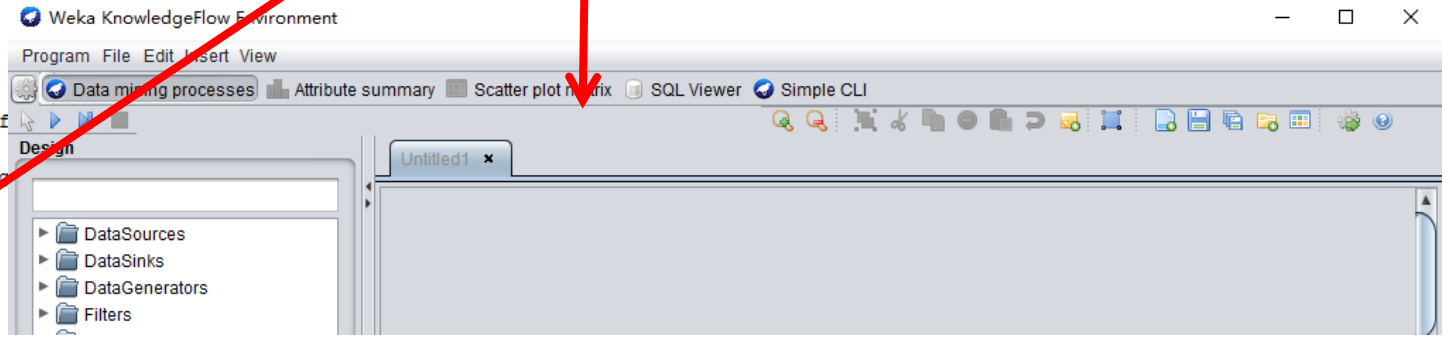
- Applications
  - Explorer
  - **Experimenter**
  - Knowledge Flow
  - Command line Interface (SimpleCLI)











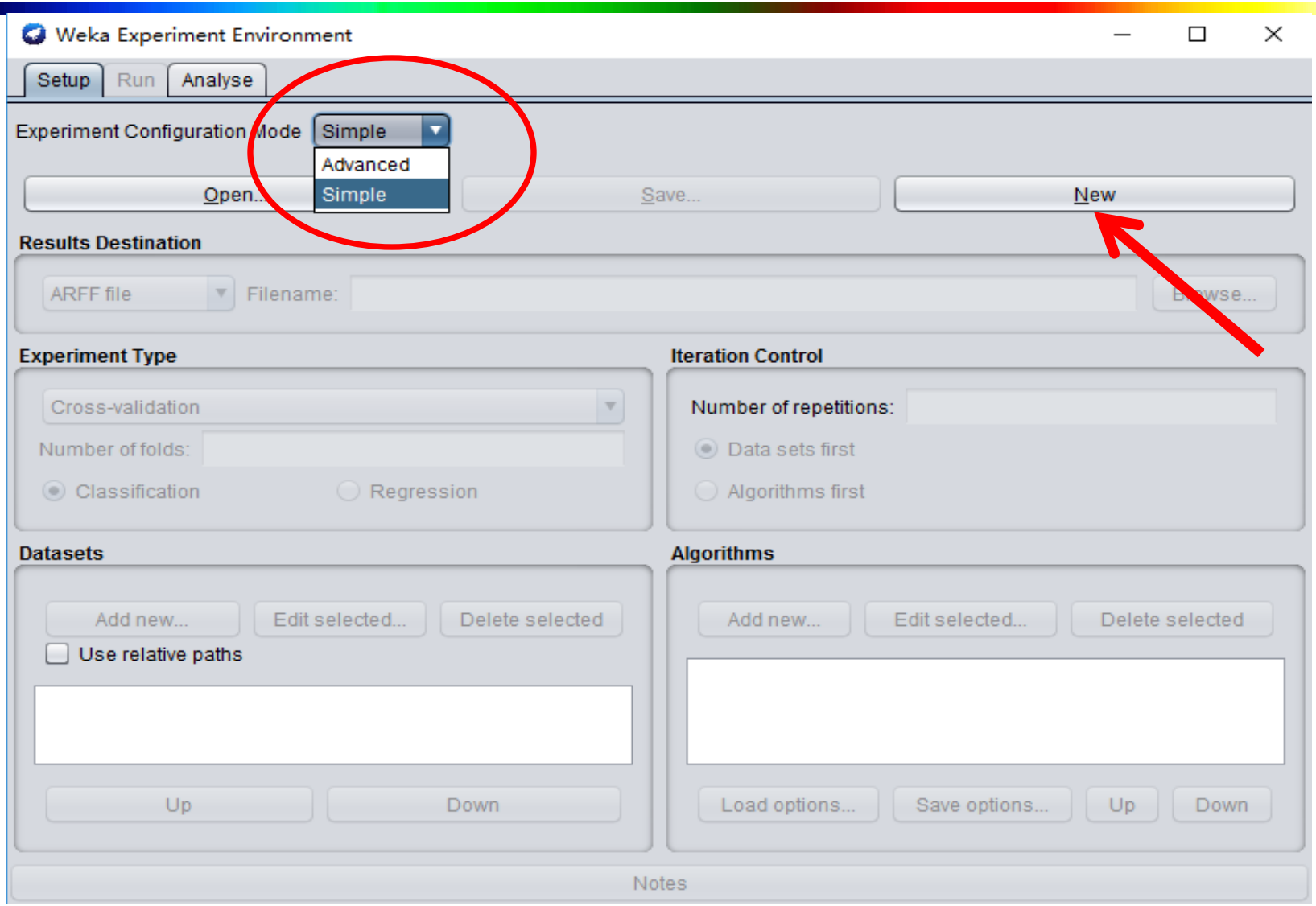
Roadmap: WEKA Usage

- WEKA Usage (version 3.8)
  - Explorer
  - Experimenter
  - Knowledge Flow

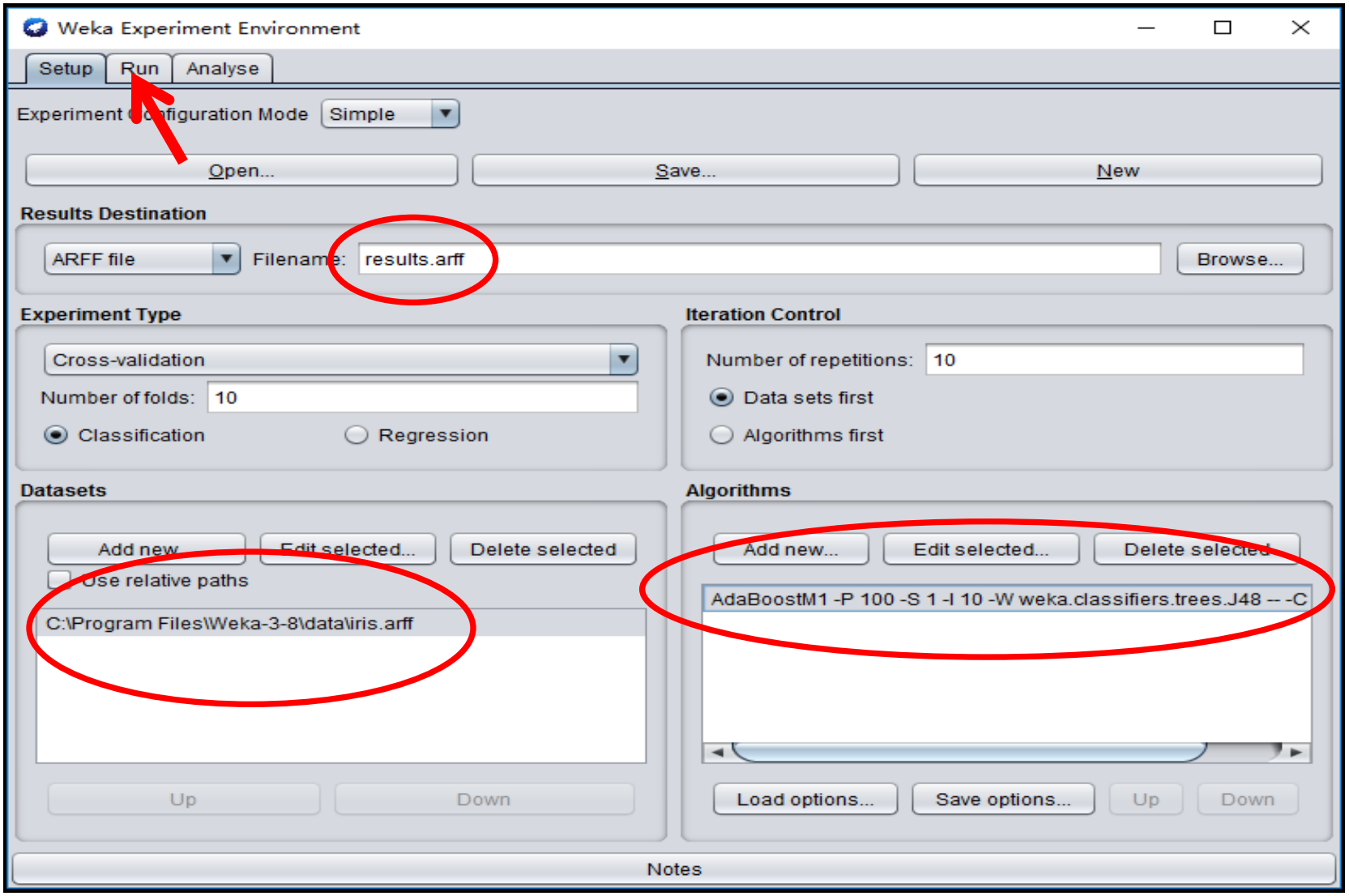
Performing Experiments

- 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
- Significance-testing built in!

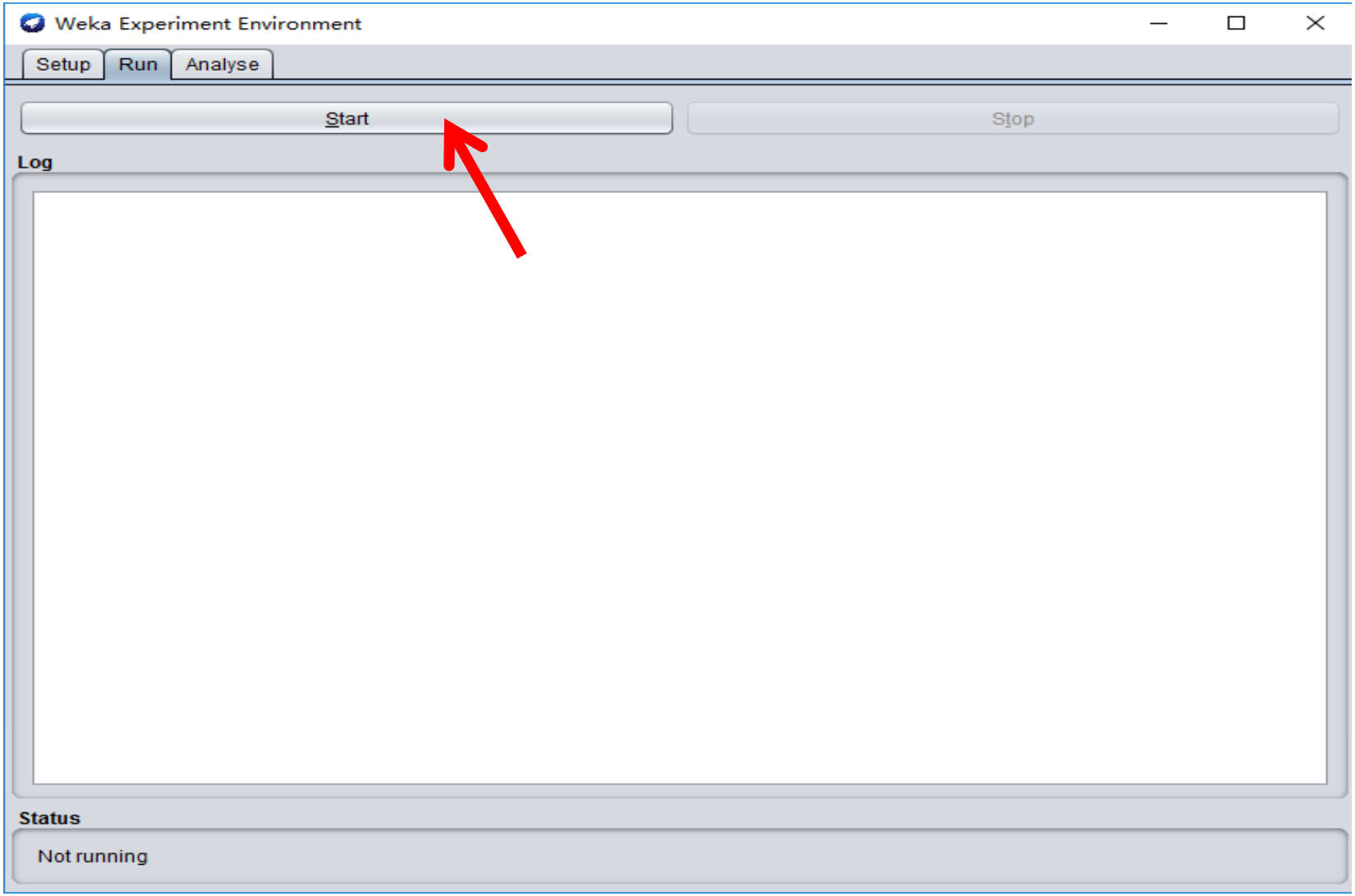
Experimenter



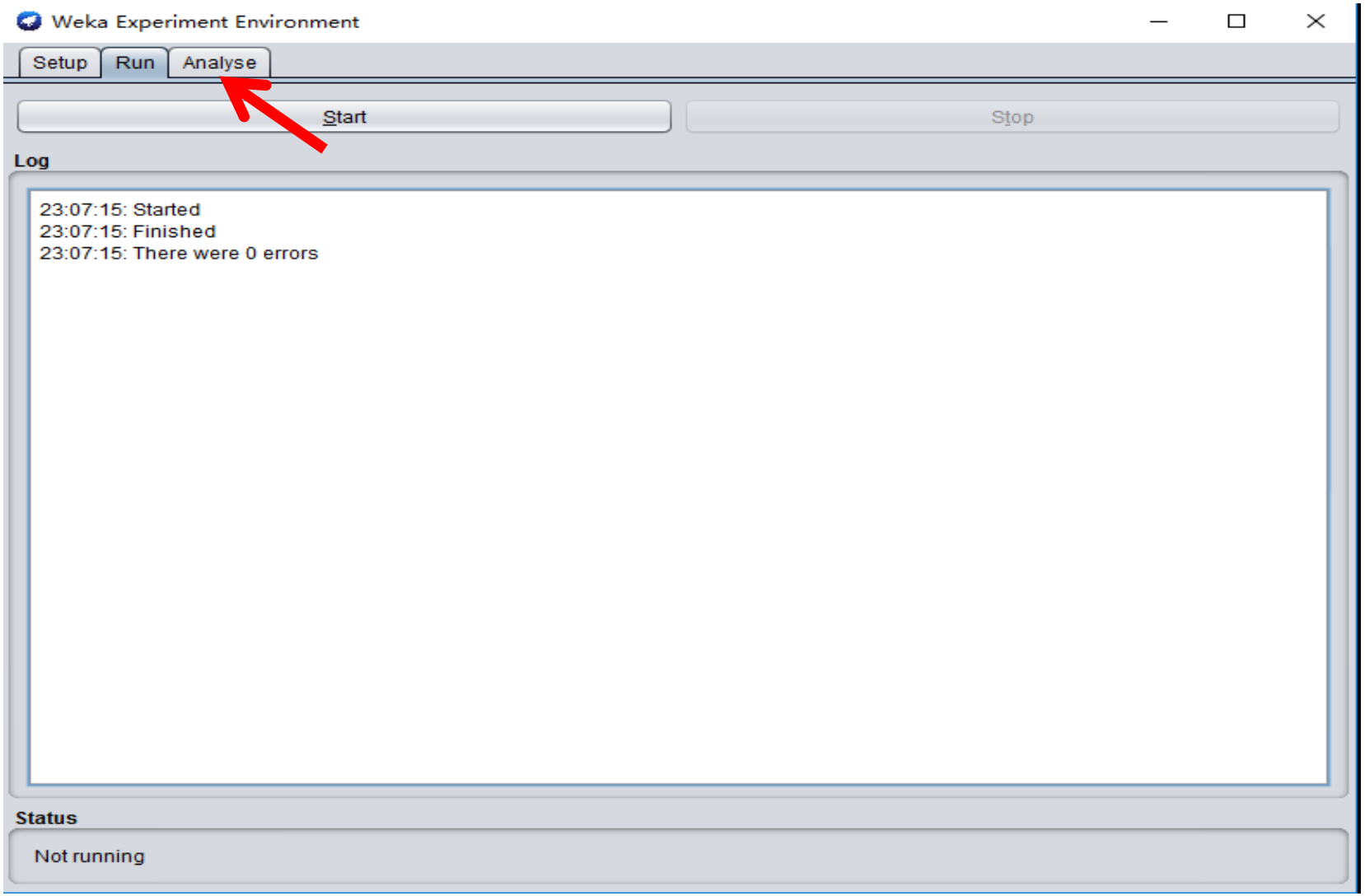
Experimenter



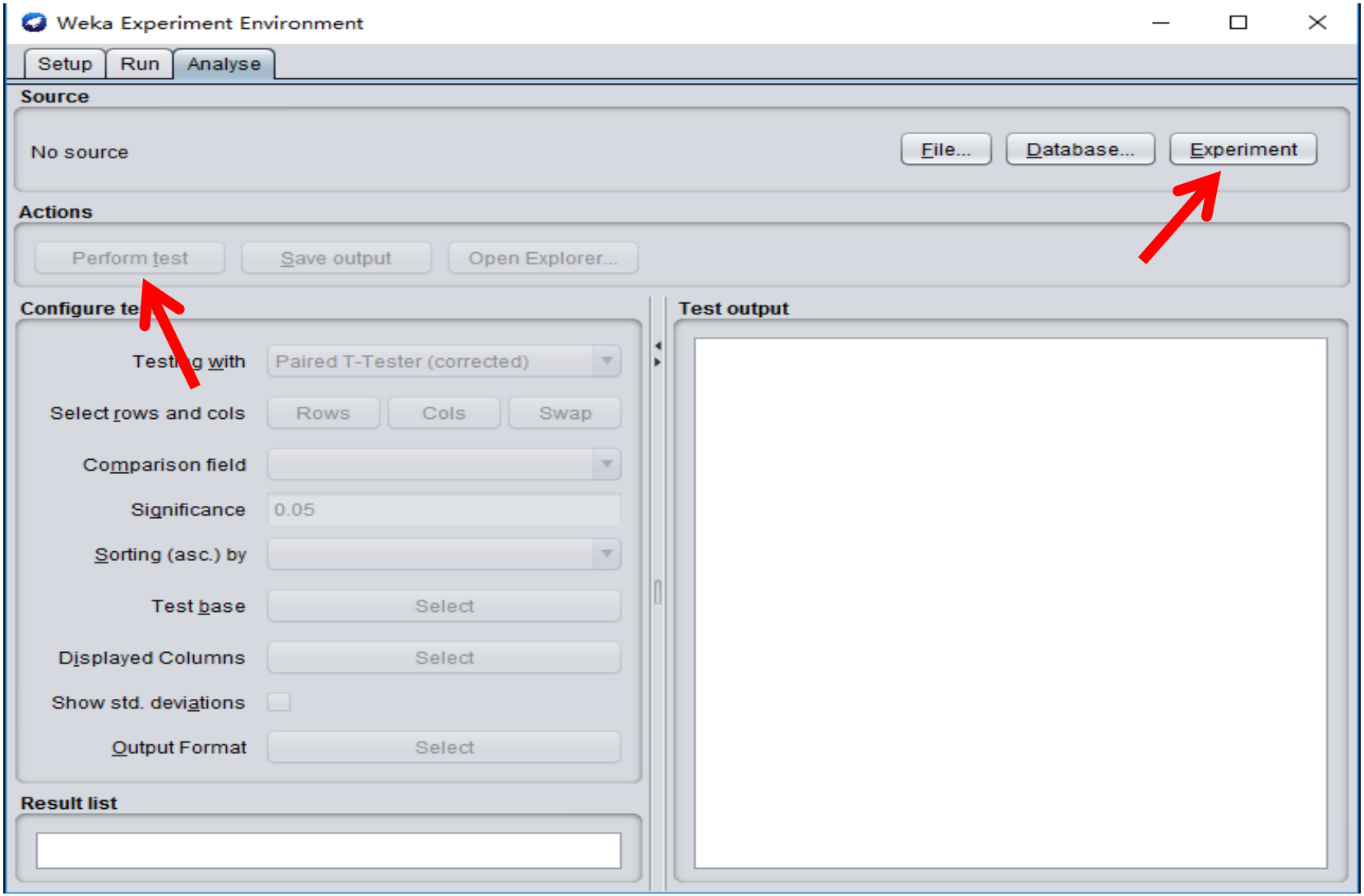
Experimenter



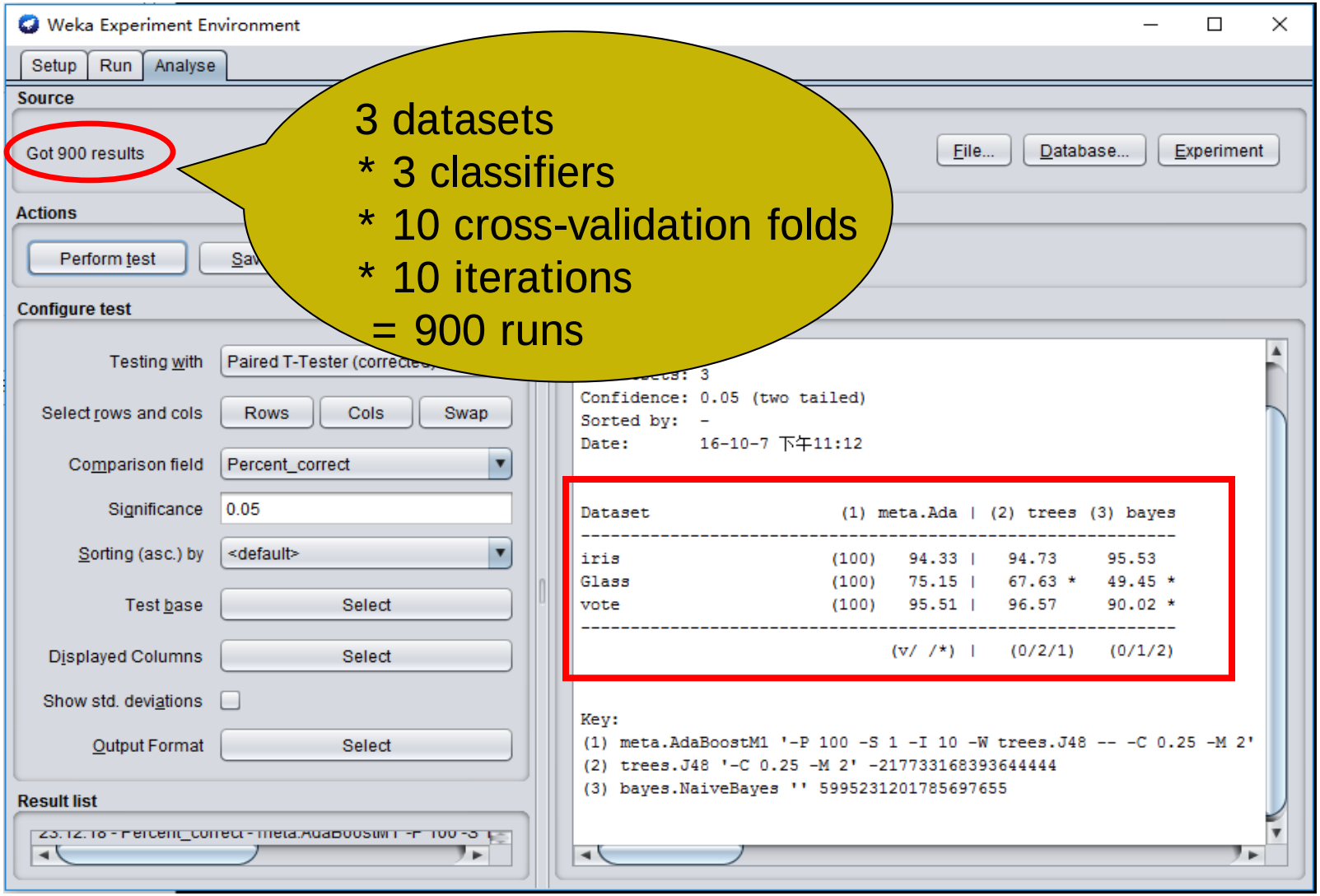
Experimenter



Experimenter



Experimenter

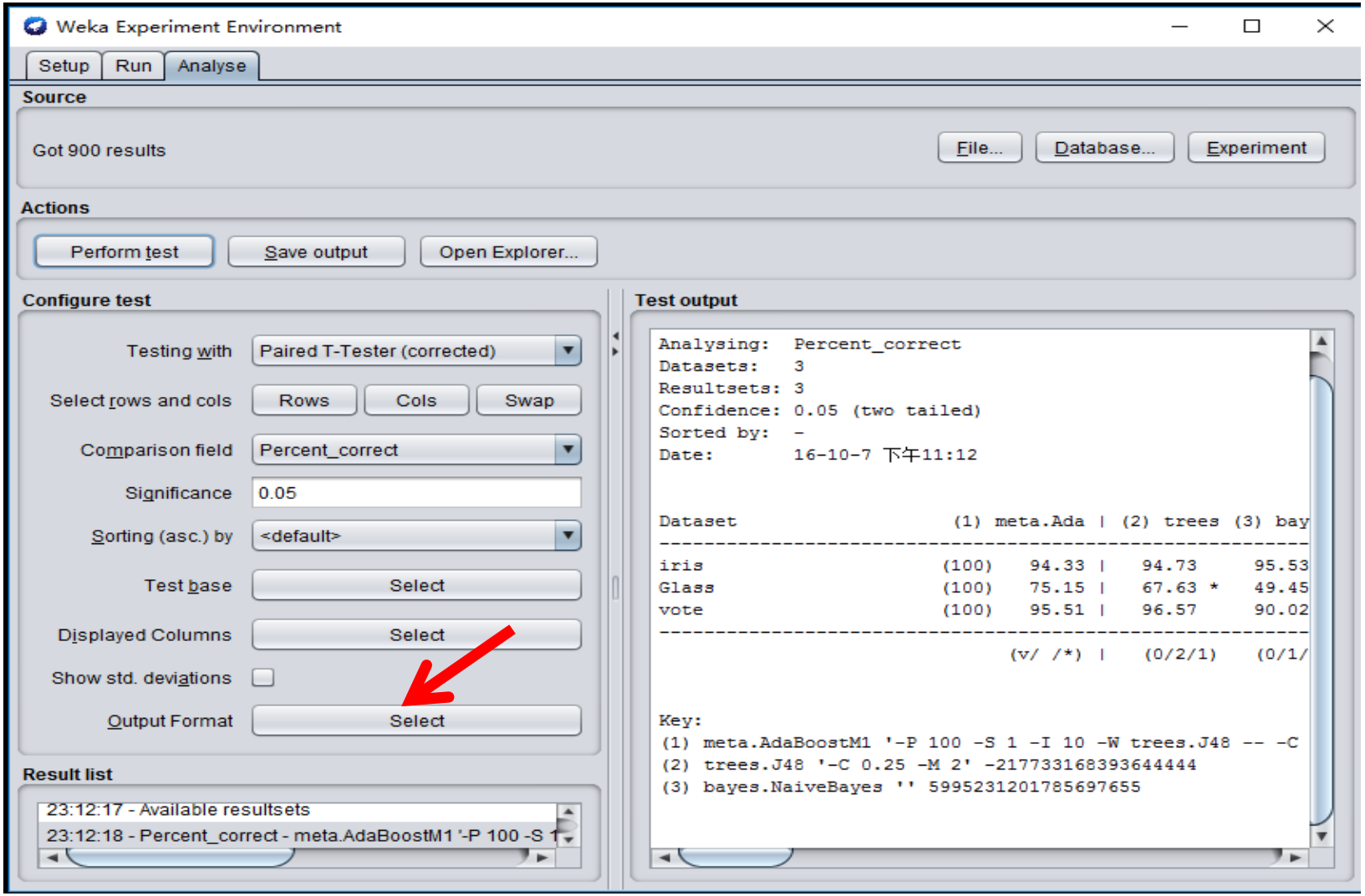


Close-Up: the result

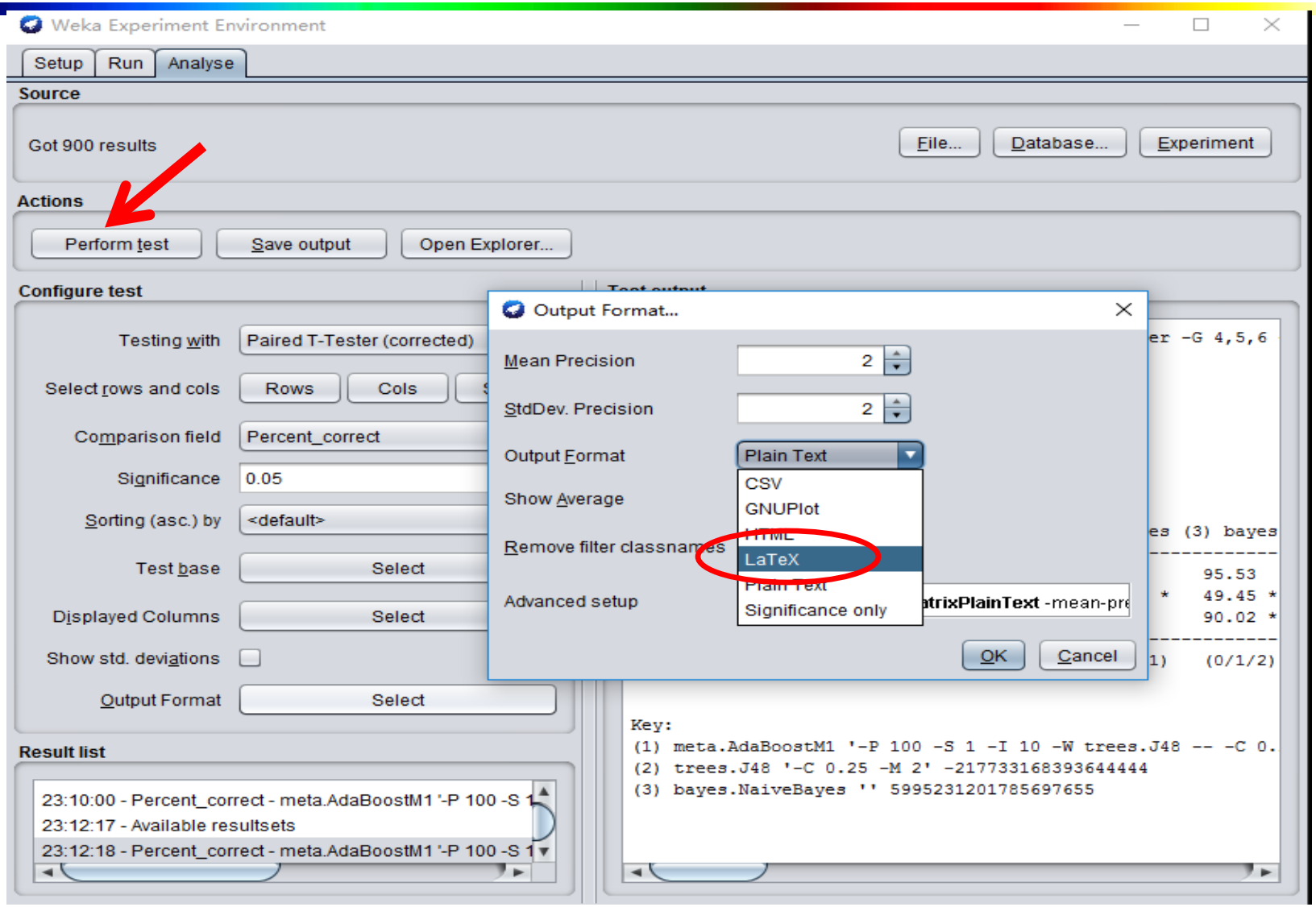
Dataset		(1) meta.Ada	(2) trees	(3) bayes
iris	(100)	94.33	94.73	95.53
Glass	(100)	75.15	67.63 *	49.45 *
vote	(100)	95.51	96.57	90.02 *
		(v/ /*)	(0/2/1)	(0/1/2)

- The first classifier acts as baseline classifier
- *v* denotes the significantly better than baseline at the significance level $\alpha = 0.05$
- * denotes the significantly worse than baseline
- (0/1/2): significantly better in 0 dataset; 1 dataset has no significant difference; 2 datasets significantly worse

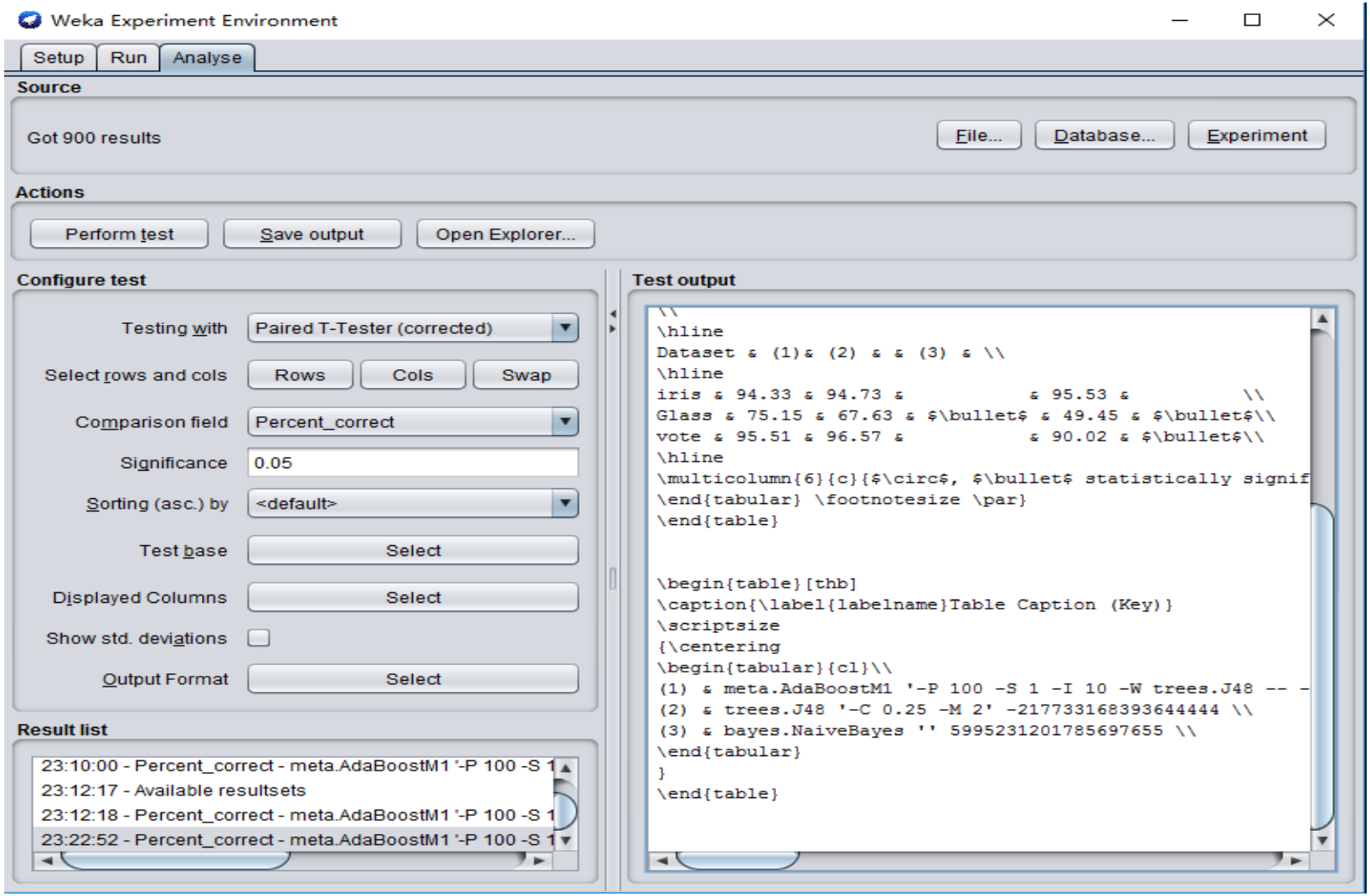
Experimenter



Experimenter

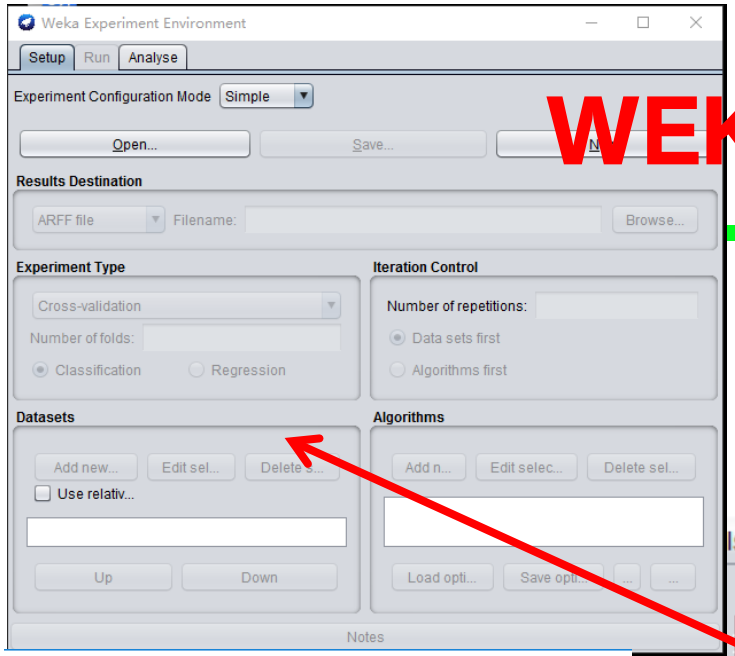


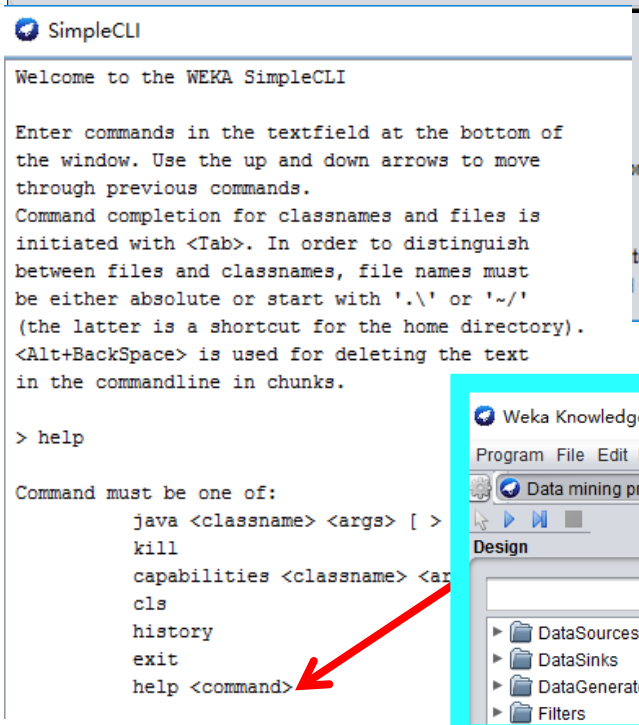
Experimenter

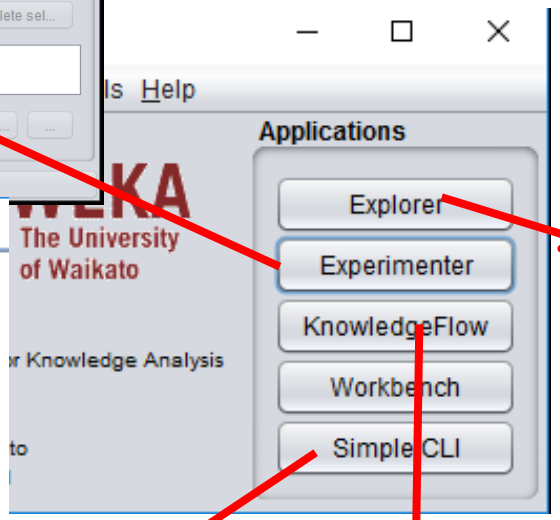


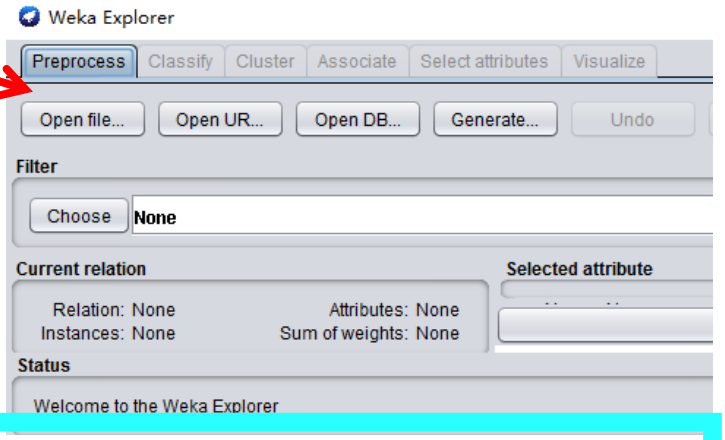
WEKA窗口

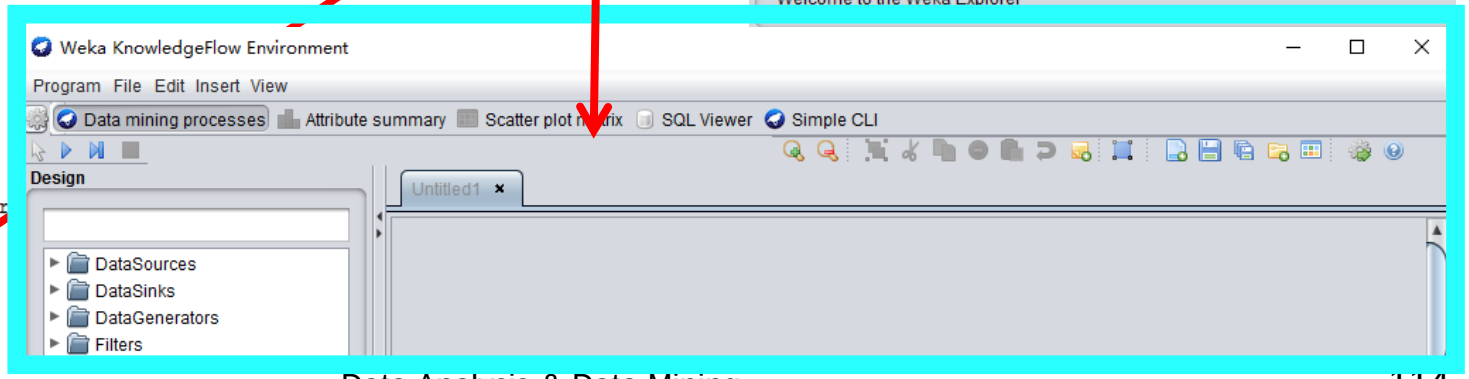
- Applications
  - Explorer
  - Experimenter
  - **Knowledge Flow**
  - Command line Interface (SimpleCLI)











Roadmap: WEKA Usage

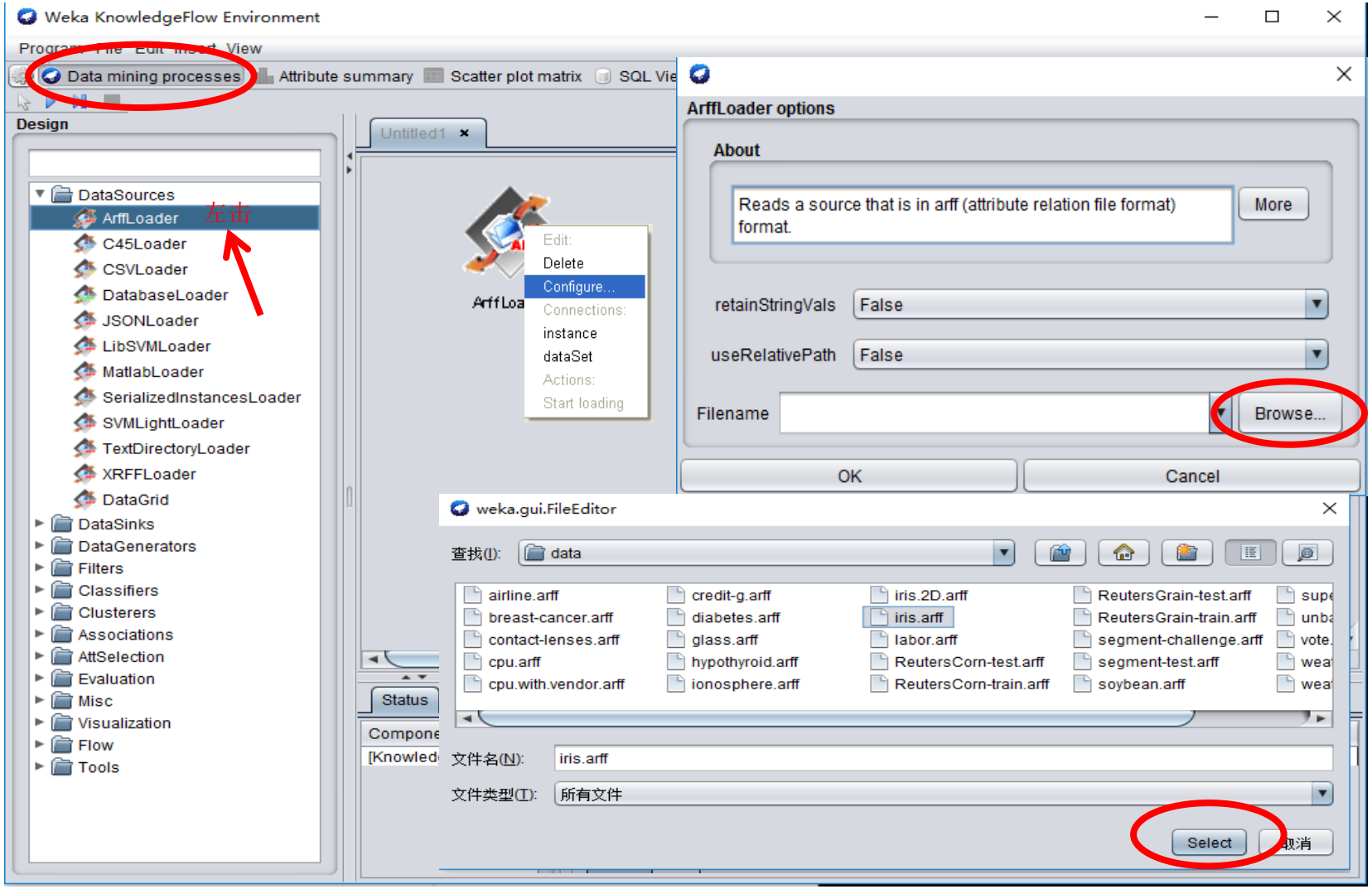
- WEKA Usage (version 3.8)
  - Explorer
  - Experimenter
  - Knowledge Flow

The Knowledge Flow GUI

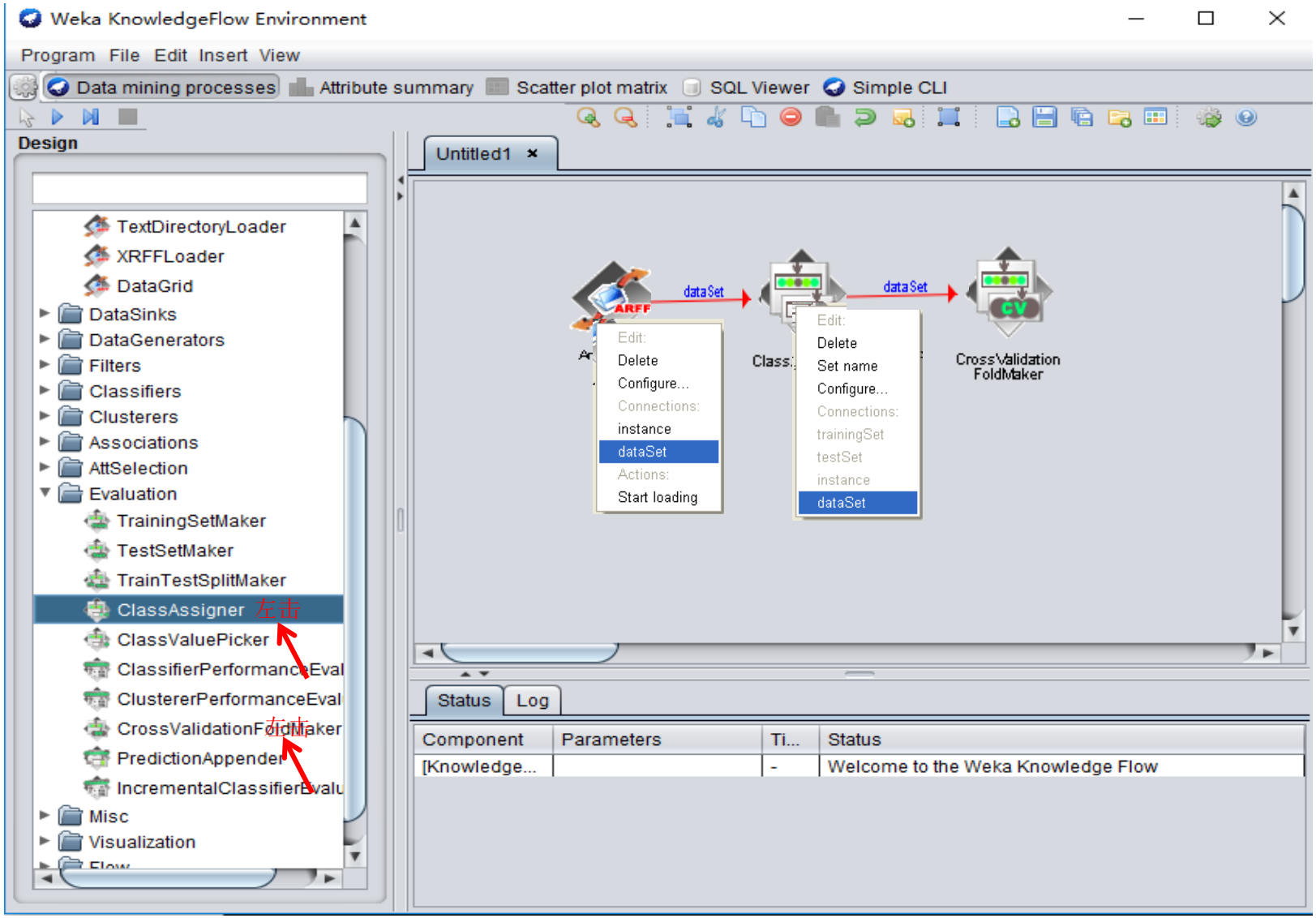
- Java-Beans-based interface for setting up and running machine learning experiments
- Data sources, classifiers, etc. are beans and can be connected graphically
- Data “flows” through components: e.g.,

  “data source” → “filter” → “classifier” → “evaluator”
- Layouts can be saved and loaded again later

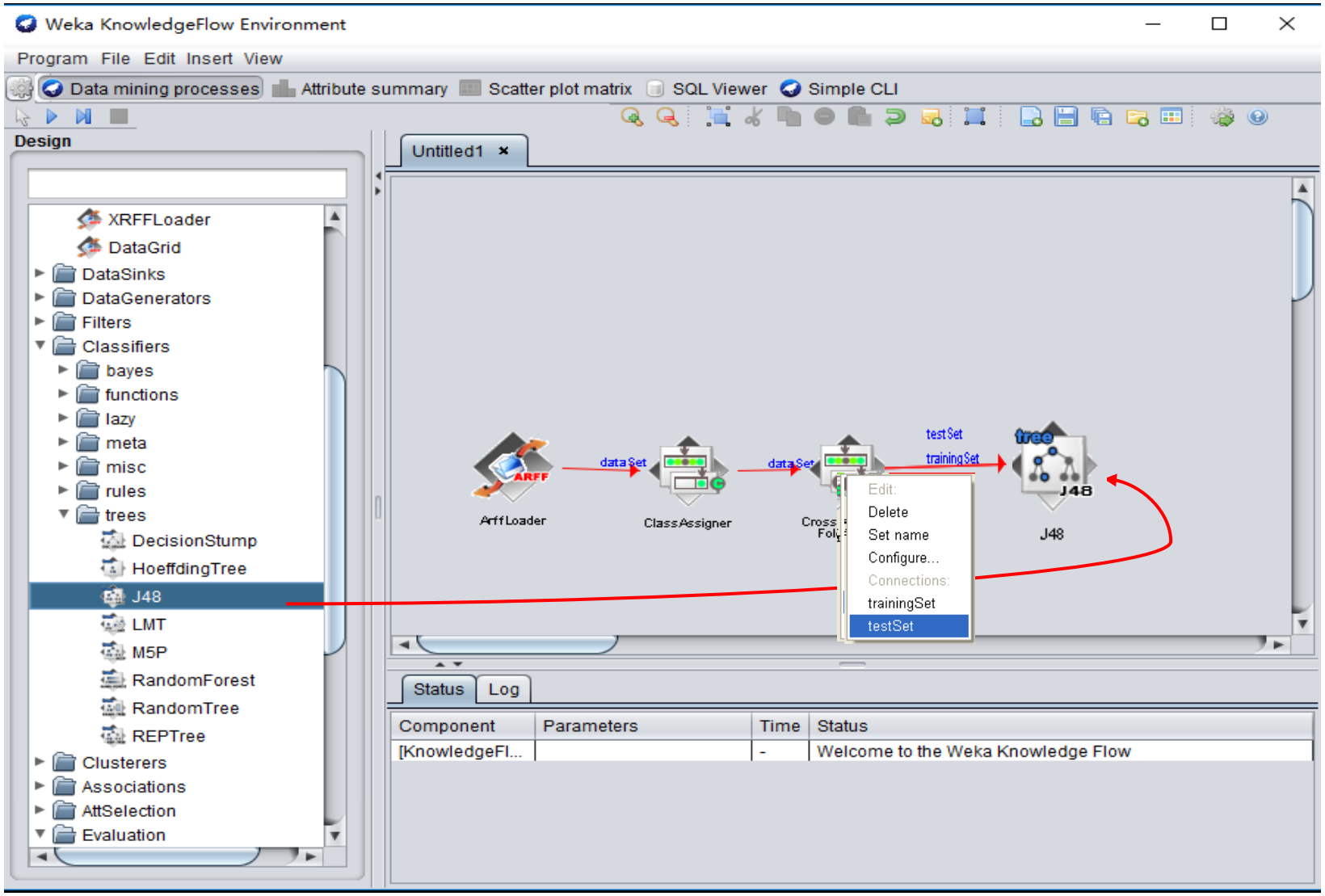
Ex1: Cross-validated J48 (C4.5)



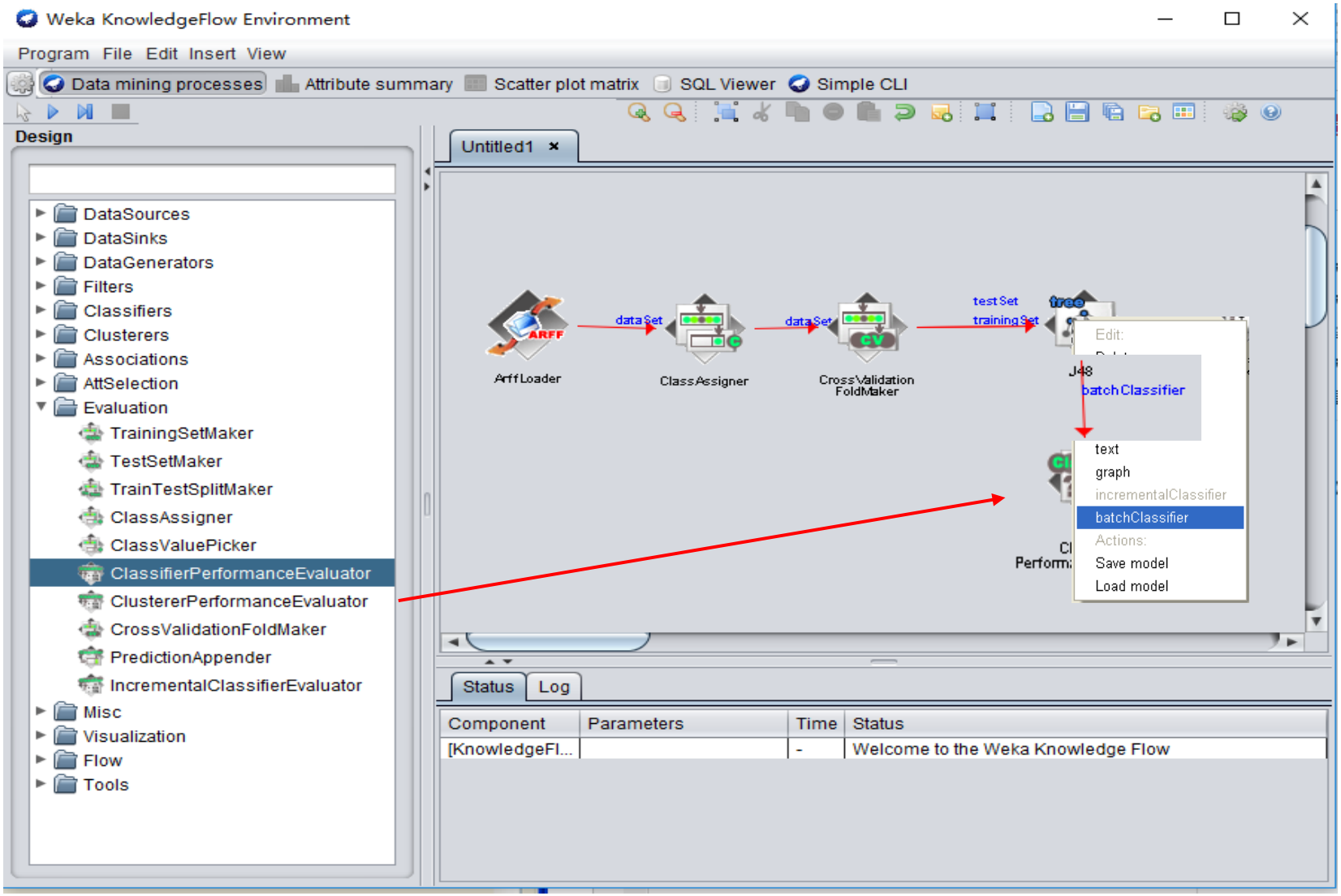
Ex1: Cross-validated J48 (C4.5)



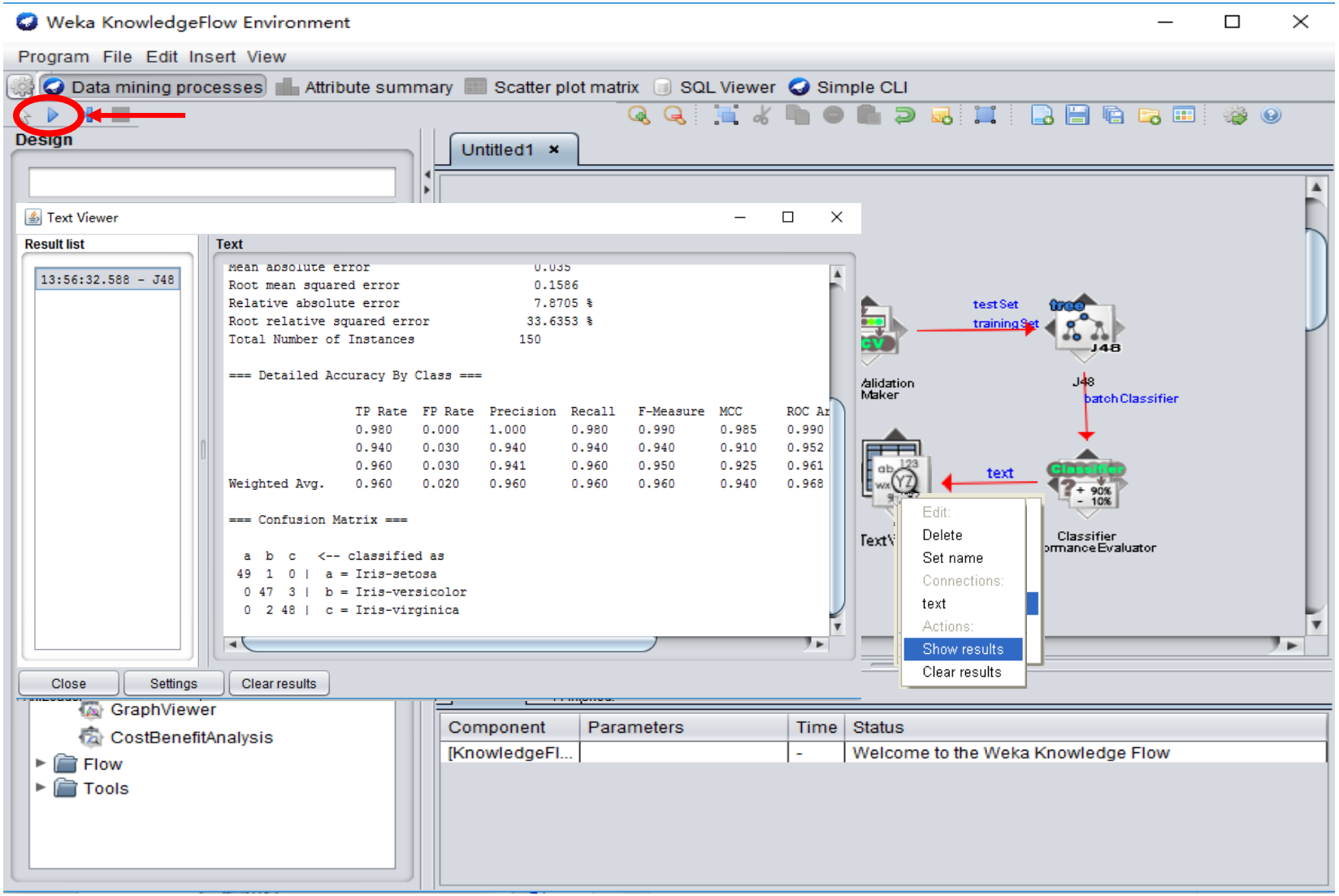
Ex1: Cross-validated J48 (C4.5)



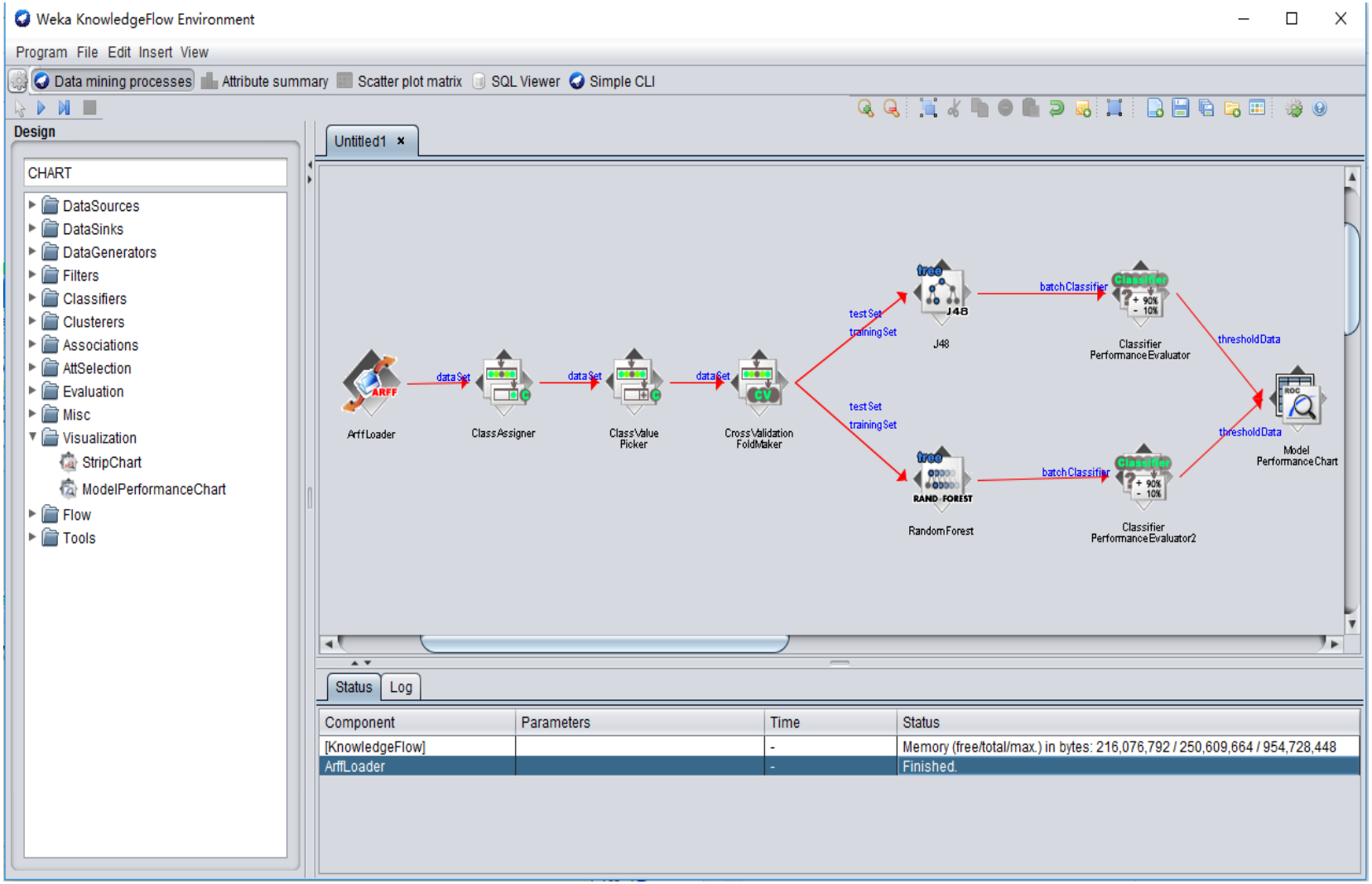
Ex1: Cross-validated J48 (C4.5)



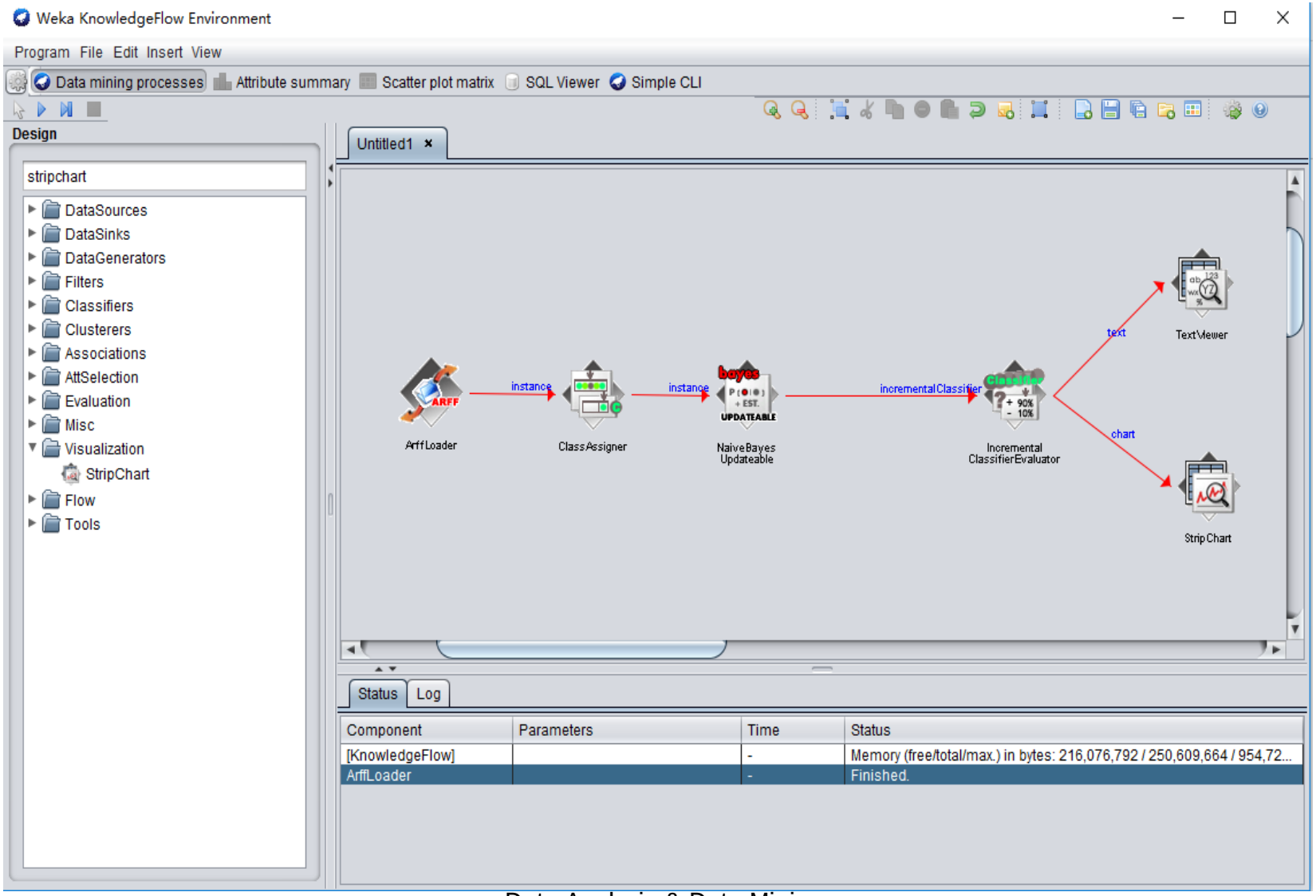
Ex1: Cross-validated J48 (C4.5)



Ex2：Plotting multiple ROC curves



Ex3: Processing data incrementally



Machine Learning with WEKA 3.8

- About WEKA
- Data Format and Preprocessing
- WEKA Usage (version 3.8)
  - Explorer
  - Experimenter
  - Knowledge Flow
- Other Data Mining Tools
- Summary

Other Data Mining Tools

- R (http://www.r-project.org)
- Tanagra (http://eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html)
- YALE (Yet Another Learning Environment, http://rapid-i.com/, new name: RapidMiner)
- KNIME (Konstanz Information Miner, http://www.knime.org)
- Orange (http://orange.biolab.si/)
- GGobi (http://www.ggobi.org/)

Machine Learning with WEKA 3.8

- About WEKA
- Data Format and Preprocessing
- WEKA Usage (version 3.8)
  - Explorer
  - Experimenter
  - Knowledge Flow
- Other Data Mining Tools
- Summary

Summary

- Java-based open source
- Provide various data preprocessing, data mining functions and evaluation measures
- Incremental functions and components
- WEKA is available at

  http://www.cs.waikato.ac.nz/ml/weka

Projects based on WEKA

- 45 projects currently (30/01/07) listed on the WekaWiki
- Incorporate/wrap WEKA
  - GRB Tool Shed - a tool to aid gamma ray burst research
  - YALE - facility for large scale ML experiments
  - GATE - NLP workbench with a WEKA interface
  - Judge - document clustering and classification
  - RWeka - an R interface to Weka
- Extend/modify WEKA
  - BioWeka - extension library for knowledge discovery in biology
  - WekaMetal - meta learning extension to WEKA
  - Weka-Parallel - parallel processing for WEKA
  - Grid Weka - grid computing using WEKA
  - Weka-CG - computational genetics tool library

Limitations of WEKA

- Traditional algorithms need to have all data in main memory
- ==> big datasets are an issue
- Solution:
  - Incremental schemes
  - Stream algorithms

    MOA "**M**assive **O**nline **A**nalysis"

    (not only a *flightless* bird, but also *extinct*!)

Conclusion: try it yourself!

- WEKA related reference
  - WekaWiki – http://weka.sourceforge.net/wiki/
  - WekaDoc – http://weka.sourceforge.net/wekadoc/
  - Ensemble Selection on WekaDoc –
    - http://weka.sourceforge.net/wekadoc/index.php/en:Ensemble Selection
  - Extensions for Weka's main GUI on WekaWiki –
    - http://weka.sourceforge.net/wiki/index.php/Extensions for Weka%27s main GUI
  - Adding tabs in the Explorer on WekaWiki –
    - http://weka.sourceforge.net/wiki/index.php/Adding tabs in the Explorer
  - Explorer visualization plugins on WekaWiki –
    - http://weka.sourceforge.net/wiki/index.php/Explorer visualization plugins
  - Other related reference
    - http://www.itl.nist.gov/div898/handbook/index.htm
    - http://wiki.wekacn.org/index.php/%E6%95%B0%E6%8D%AE%E6%8C%96%E6%8E%98
    - http://forum.wekacn.org/
    - http://www.wekacn.org/