

BÁO CÁO ĐỒ ÁN 3 MÁY HỌC

Môn học: Cơ sở trí tuệ nhân tạo

Lớp: Cử nhân tài năng

Thực hiện: Hoàng Thiên Nữ - Dương Nguyễn Thái Bảo



Tháng 12/2018

I. Thông tin chung

1. Thông tin thành viên

| MSSV | Họ và tên |
|---------|-----------------------|
| 1612880 | Hoàng Thiên Nữ |
| 1612840 | Dương Nguyễn Thái Bảo |

2. Phân công và đánh giá mức độ hoàn thành

| STT | Công việc | Người thực hiện | Mức độ hoàn thành |
|-----|---|-----------------------|-------------------|
| 1 | Tìm hiểu công cụ Weka. | Dương Nguyễn Thái Bảo | 100 % |
| 2 | Sử dụng Weka để chạy thuật toán ID3. | Hoàng Thiên Nữ | 100 % |
| 3 | Code Naive Bayes bằng Python. | Hoàng Thiên Nữ | 100 % |
| 4 | Chạy thêm các thuật toán khác trên Weka | Dương Nguyễn Thái Bảo | 100 % |

II. Yêu cầu 1

1. Giới thiệu công cụ Weka

Weka (Waikato Environment for Knowledge Analysis) là một bộ phần mềm cung cấp các chức năng Học Máy (Machine Learning) được viết bằng Java, phát triển bởi đại học Waikato, New Zealand. Weka là phần mềm tự do phát hành theo Giấy phép Công cộng GNU.

Weka hỗ trợ một bộ các công cụ visualize và thuật toán cho phân tích dữ liệu và xây dựng mô hình dự đoán, với điểm nổi trội là giao diện đồ họa (GUI) thân thiện khiến việc thao tác với các công cụ dễ dàng hơn cho người dùng.

2. Cài đặt

Tải và chạy/cài đặt phiên bản Weka tương ứng với hệ điều hành đang sử dụng ở địa chỉ: <https://www.cs.waikato.ac.nz/ml/weka/downloading.html>.

3. Các chức năng

Khi chạy Weka, giao diện khởi động sẽ gồm các lựa chọn ứng dụng chức năng bao gồm Explorer, Experimenter, KnowledgeFlow, Workbench và Simple CLI:

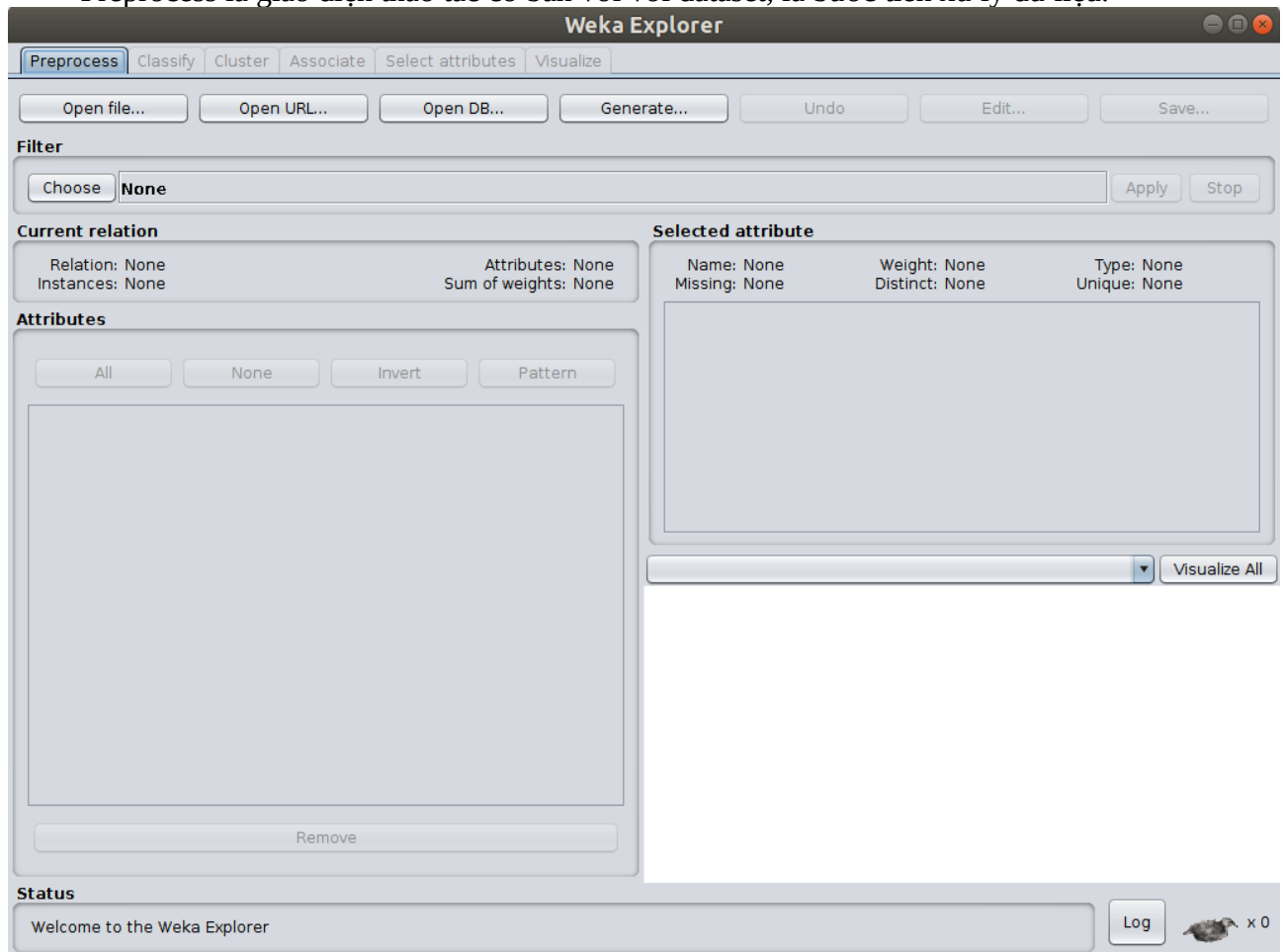


Explorer:

Explorer là ứng dụng cung cấp các chức năng Học Máy chính của Weka. Ở đây chứa các nhóm công cụ bao gồm Preprocess, Classify, Cluster, Associate, Select attributes và Visualize.

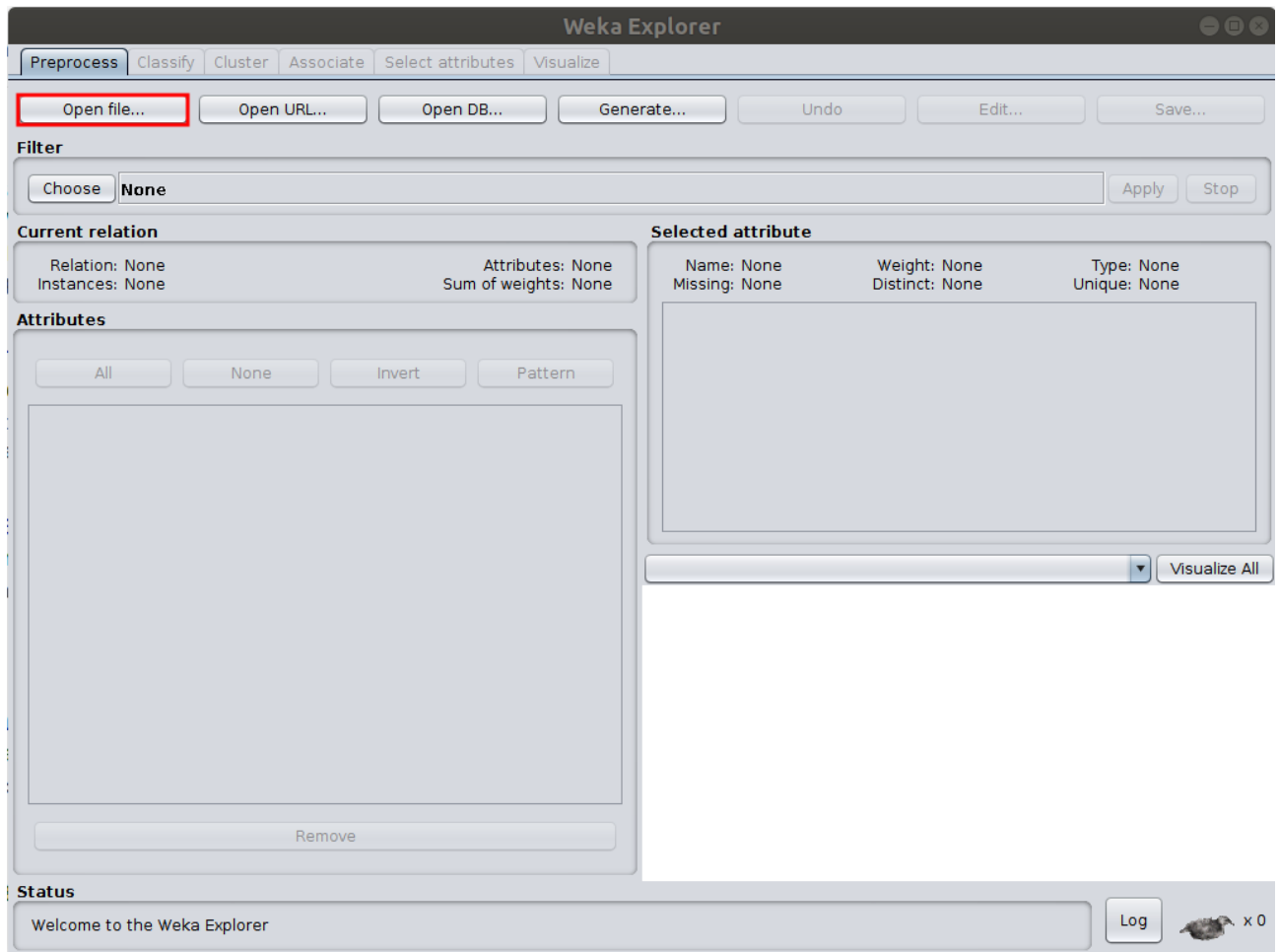
Preprocess

Preprocess là giao diện thao tác cơ bản với với dataset, là bước tiền xử lý dữ liệu.

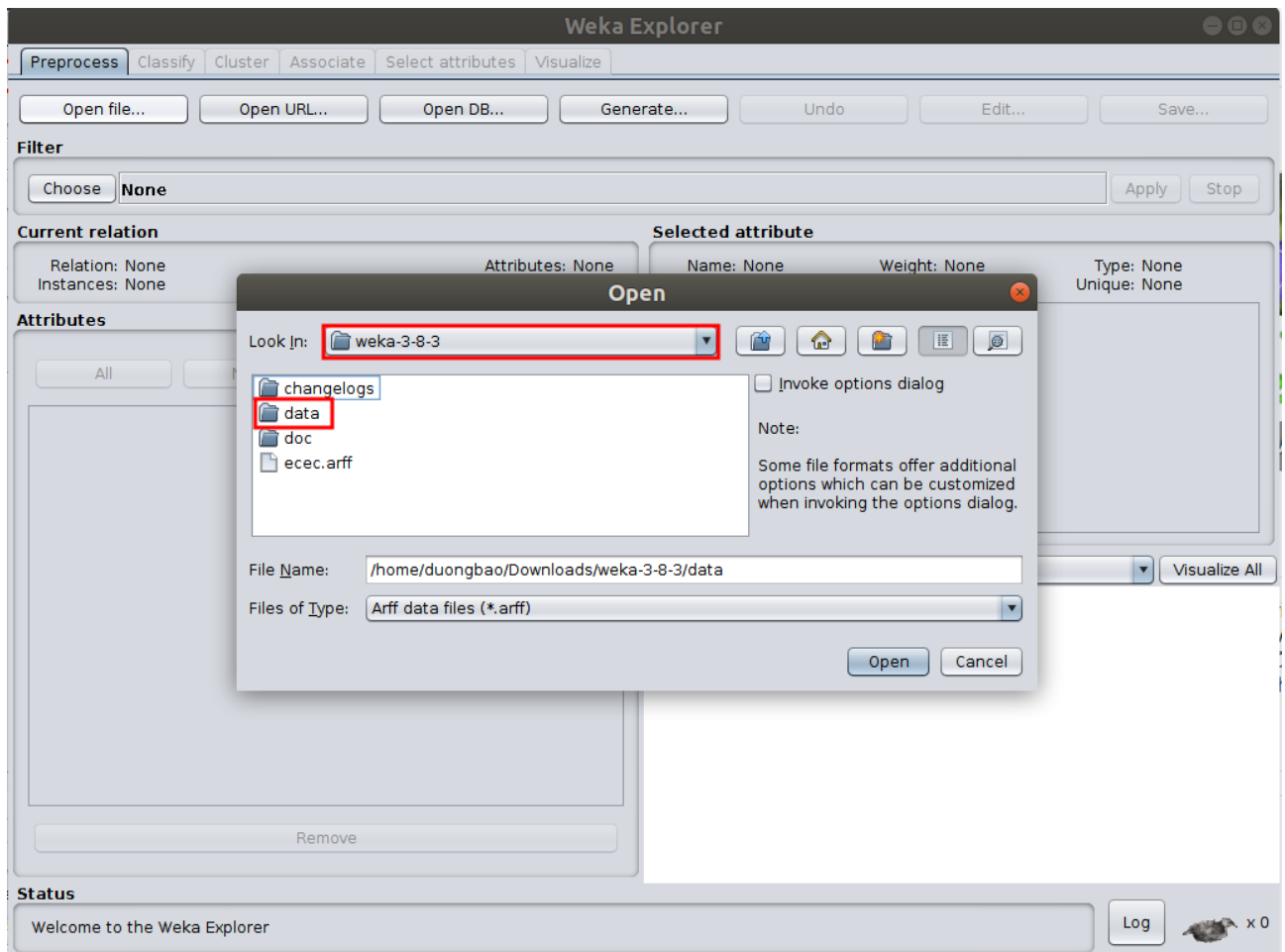


Ví dụ muốn làm việc với bộ dataset Iris (bộ dữ liệu hoa Iris là một bộ dữ liệu đa biến được sử dụng phổ biến trong việc minh họa về data classification, bộ dữ liệu này có sẵn trong tập data đi kèm với phần mềm Weka tải về):

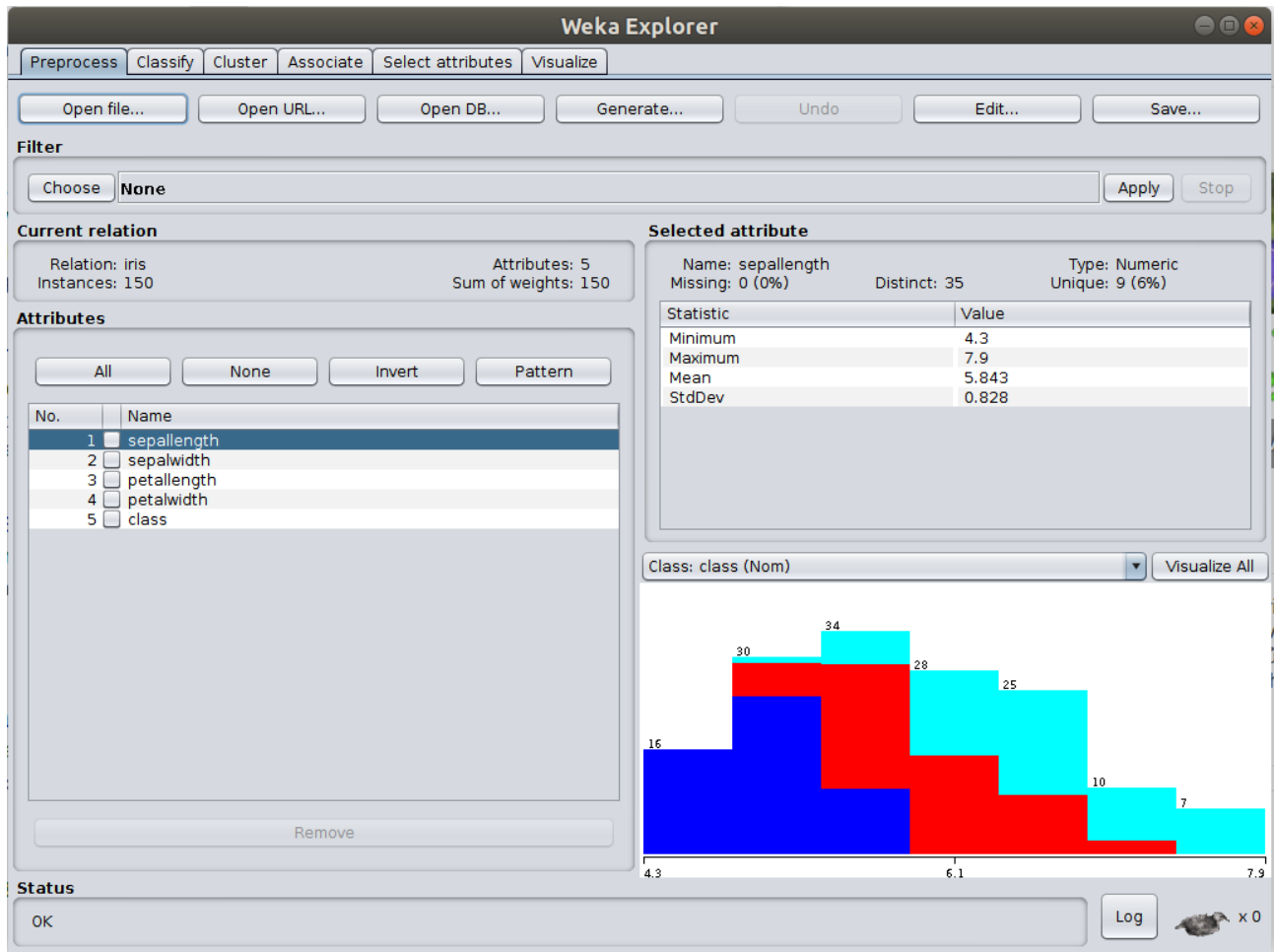
- Chọn Open file:



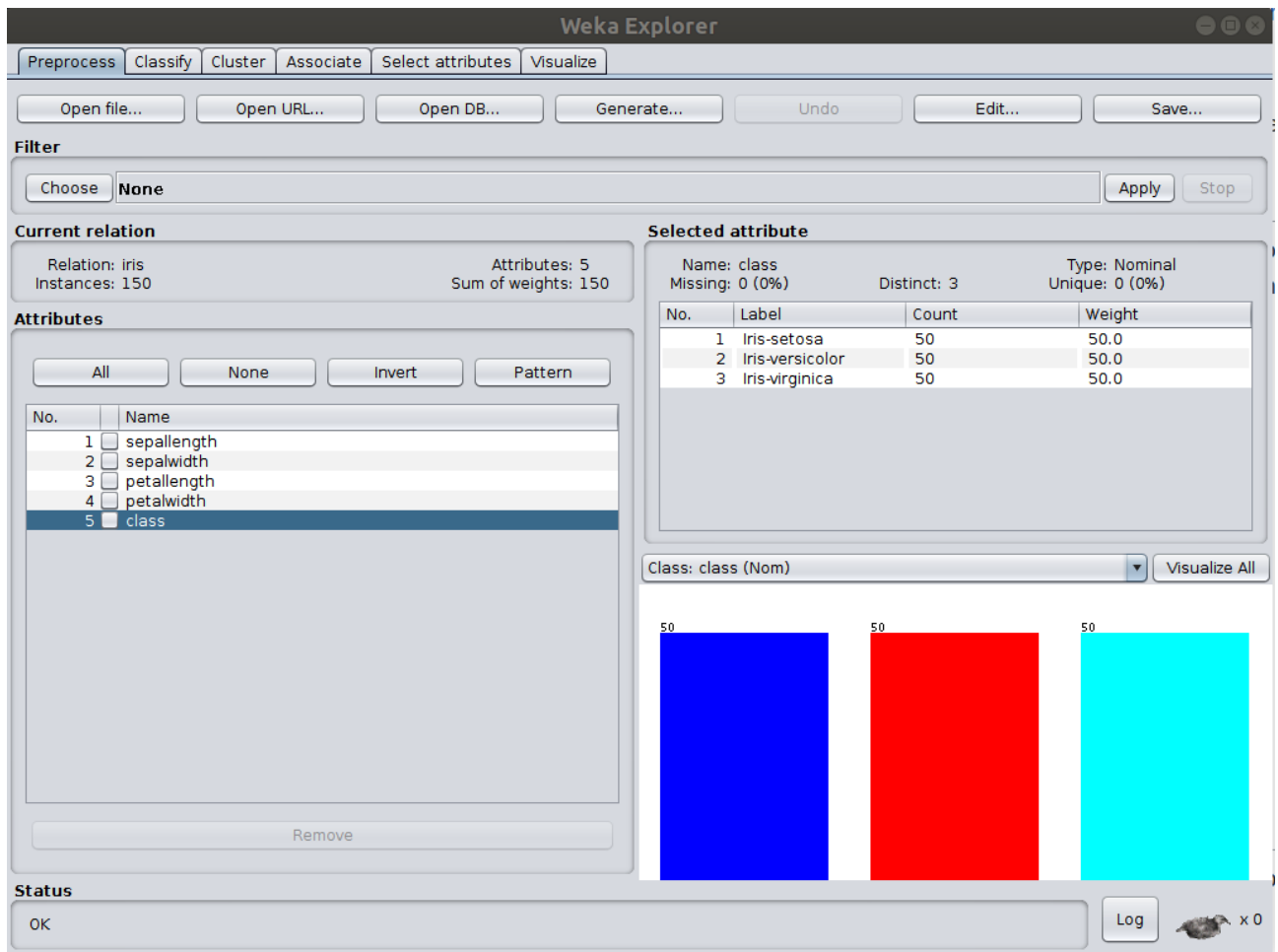
- Di chuyển tới thư mục Weka, chọn thư mục data:



- Chọn mở tập tin Iris.arff (arff là tên mở rộng riêng của tập tin dữ liệu của Weka nên có thể thấy là tất cả các file dataset trong thư mục data đều có đuôi là arff).
- Sau khi mở Iris.arff, dữ liệu được load ra màn hình như sau:

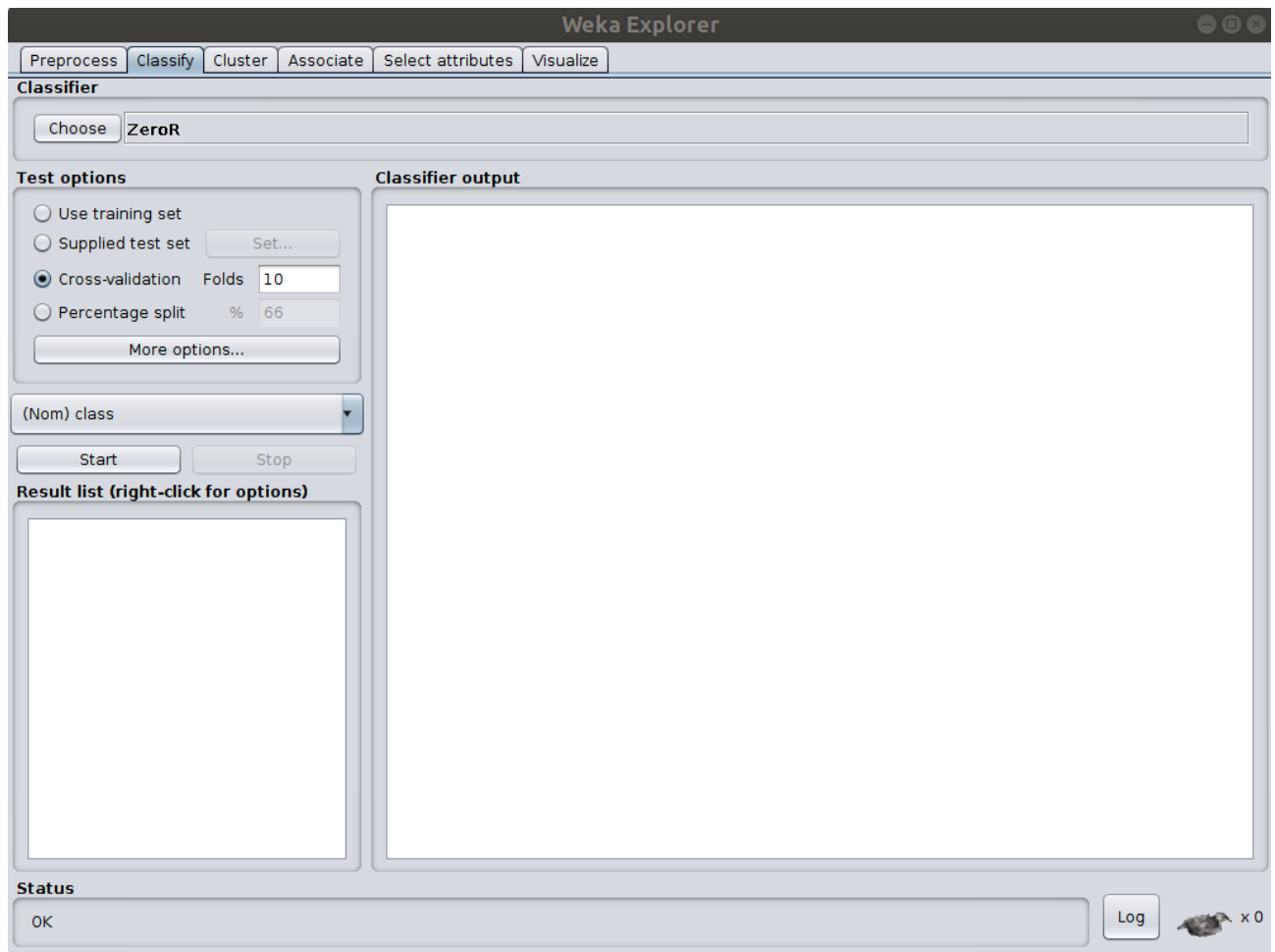


- Giải thích:
 - Khung Current relation nêu các mô tả cơ bản của tập dữ liệu. Như bộ Iris ở trên thì có thể thấy:
 - Relation: Iris - tên relation là Iris.
 - Attributes: 5 - có 5 thuộc tính cho mỗi điểm dữ liệu.
 - Instance: 150 - có 150 điểm dữ liệu.
 - Khung Attributes chứa các thuộc tính của các điểm dữ liệu trong dataset. Theo ví dụ trên thì 5 thuộc tính (tính cả class) của mỗi điểm dữ liệu là *sepalength*, *sepalwidth*, *petallength*, *petalwidth* và *class*.
 - Khung Selected attribute cho biết một số đặc tính cơ bản của thuộc tính đang được chọn. Với thuộc tính đang được chọn là *sepalength* như trên, ta biết:
 - Type: numeric - kiểu của thuộc tính này là số.
 - Missing: 0% - không có điểm dữ liệu nào bị khuyết thuộc tính này.
 - Distinct: 35 - có 35 giá trị khác nhau của thuộc tính này trong toàn bộ dữ liệu.
 - Unique: 9 - có 9 giá trị của thuộc tính này mà chỉ xuất hiện tại đúng 1 điểm dữ liệu.
 - Minimum: 4.3 - giá trị nhỏ nhất là 4.3.
 - Maximum: 7.9 - giá trị lớn nhất là 7.9.
 - Mean: 5.843 - giá trị trung bình là 5.843.
 - StdDev: 0.828 - độ lệch chuẩn là 0.828.
 - Ở dưới khung Selected attribute có một biểu đồ mô tả phân bố của thuộc tính đang chọn. Mỗi màu trên biểu đồ đại diện cho 1 class. Bộ dữ liệu này gồm 3 class được chia đều nhau, mỗi class có 50 điểm dữ liệu. Màu xanh đậm đại diện cho class Iris-setosa, màu đỏ đại diện cho Iris-versicolor, màu xanh nhạt đại diện cho Iris-virginica:



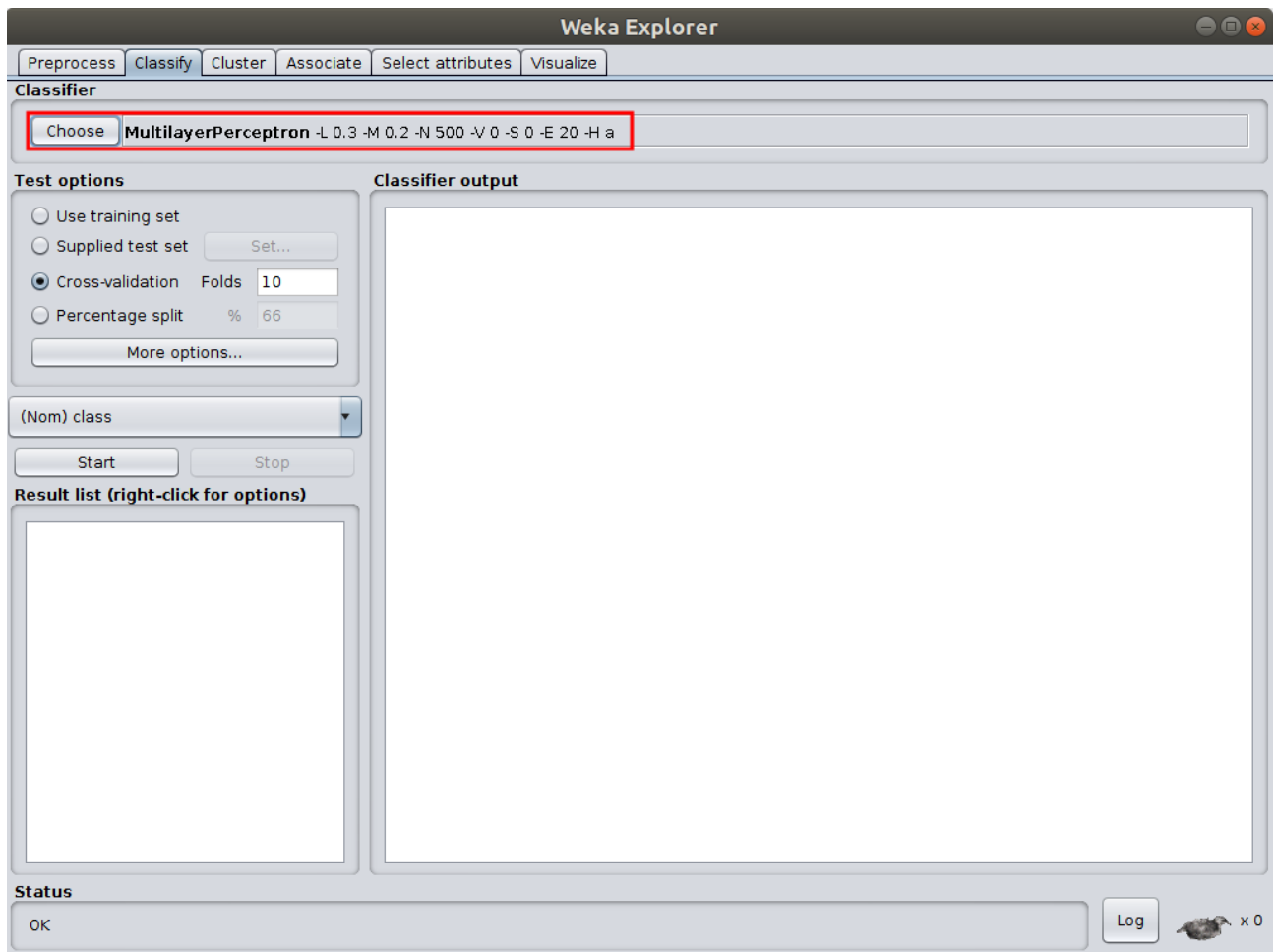
Classify

Công cụ Classify cung cấp các thuật toán về phân loại dữ liệu.

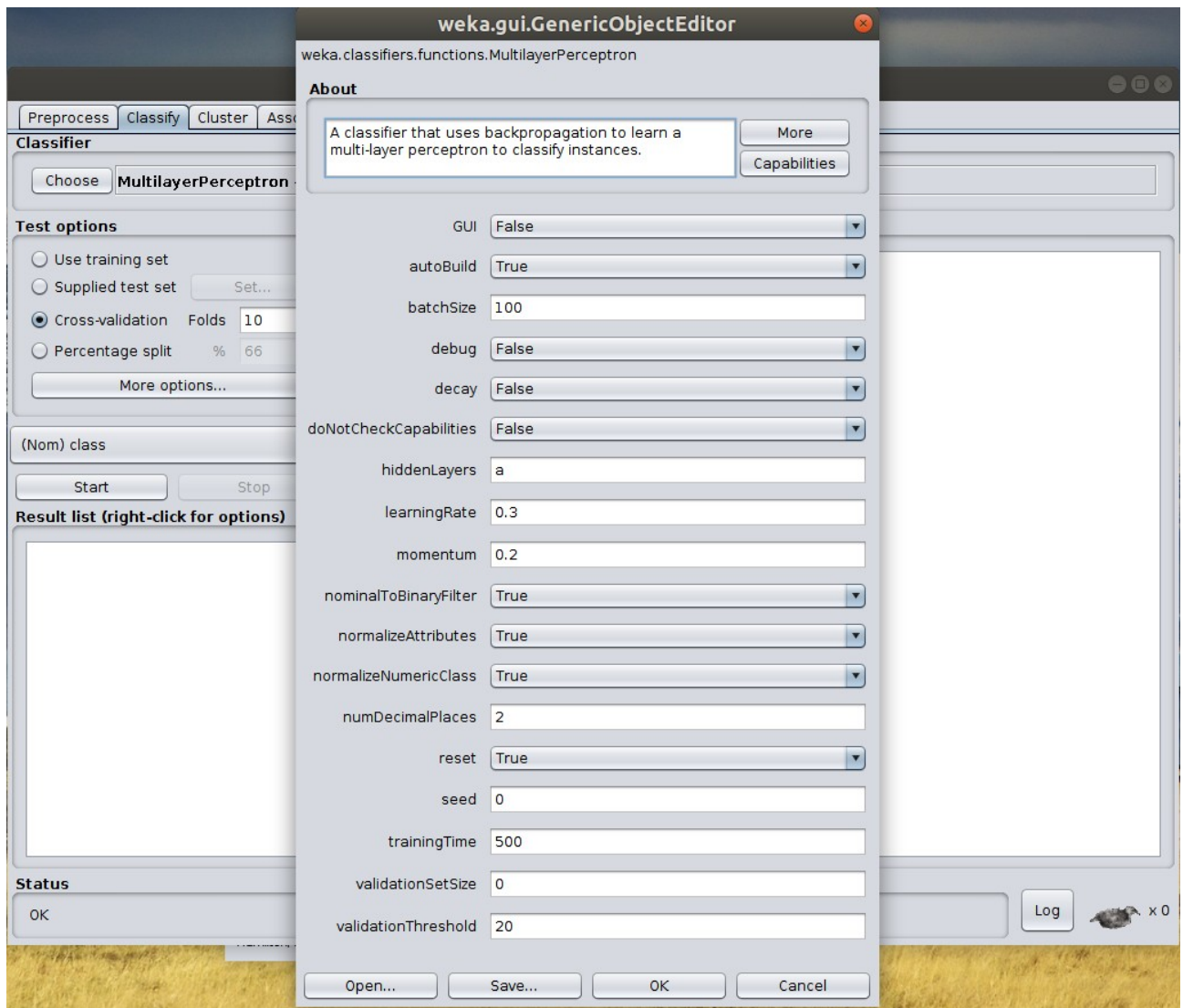


Ví dụ sử dụng thuật toán MultilayerPerceptron để phân loại tập dữ liệu Iris:

- Chọn Choose và chọn thuật toán MultilayerPerceptron trong tab functions:



- Tùy chỉnh các tham số: ấn vào chữ MultilayerPerception, một cửa sổ sẽ hiện ra để ta có thể có các tùy chỉnh riêng biệt cho thuật toán:



- Giải thích một số tham số:
 - GUI: nếu là True thì khi chạy thuật toán sẽ có visualization cho mỗi bước chạy.
 - HiddenLayers: số nút ở lớp ẩn.
 - LearningRate: learning rate.
 - Momentum: momentum.
 - TrainingTime: số epoch.
 - ...
- Cuối cùng ấn OK để chấp nhận tùy chỉnh.
- Chọn test options: Các tùy chọn bộ dữ liệu để test:
 - Use training set: test trên bộ dữ liệu train.
 - Supplied test set: dùng bộ test khác.
 - Cross-validation: dùng kỹ thuật kiểm chéo.
 - Percentage split: chia bộ training set thành 2 phần theo tỉ lệ phần trăm, 1 phần để train, 1 phần để test.
- Bấm Start và đợi kết quả:

Weka Explorer

Preprocess **Classify** Cluster Associate Select attributes Visualize

Classifier

Choose **MultilayerPerceptron** -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -R

Test options

☐ Use training set
☐ Supplied test set Set...
☒ Cross-validation Folds
☐ Percentage split %
 More options...

(Nom) class

Start Stop

Result list (right-click for options)

```

01:19:44 - functions.MultilayerPerceptron
01:21:05 - functions.MultilayerPerceptron
01:22:48 - functions.MultilayerPerceptron
01:24:10 - functions.MultilayerPerceptron
01:27:27 - functions.MultilayerPerceptron
01:27:44 - functions.MultilayerPerceptron
01:27:45 - functions.MultilayerPerceptron
01:37:33 - functions.MultilayerPerceptron
  
```

Classifier output

Input
Node 2

Time taken to build model: 0.09 seconds

=== Stratified cross-validation ===
 === Summary ===

| | | |
|----------------------------------|-----|-----------|
| Correctly Classified Instances | 146 | 97.3333 % |
| Incorrectly Classified Instances | 4 | 2.6667 % |

Kappa statistic 0.96
 Mean absolute error 0.0327
 Root mean squared error 0.1291
 Relative absolute error 7.3555 %
 Root relative squared error 27.3796 %
 Total Number of Instances 150

=== Detailed Accuracy By Class ===

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area |
|---------------|---------|---------|-----------|--------|-----------|-------|----------|----------|
| Weighted Avg. | 0.973 | 0.013 | 0.973 | 0.973 | 0.973 | 0.960 | 0.998 | 0.995 |

=== Confusion Matrix ===

```

a b c <-- classified as
50 0 0 | a = Iris-setosa
0 48 2 | b = Iris-versicolor
0 2 48 | c = Iris-virginica
  
```

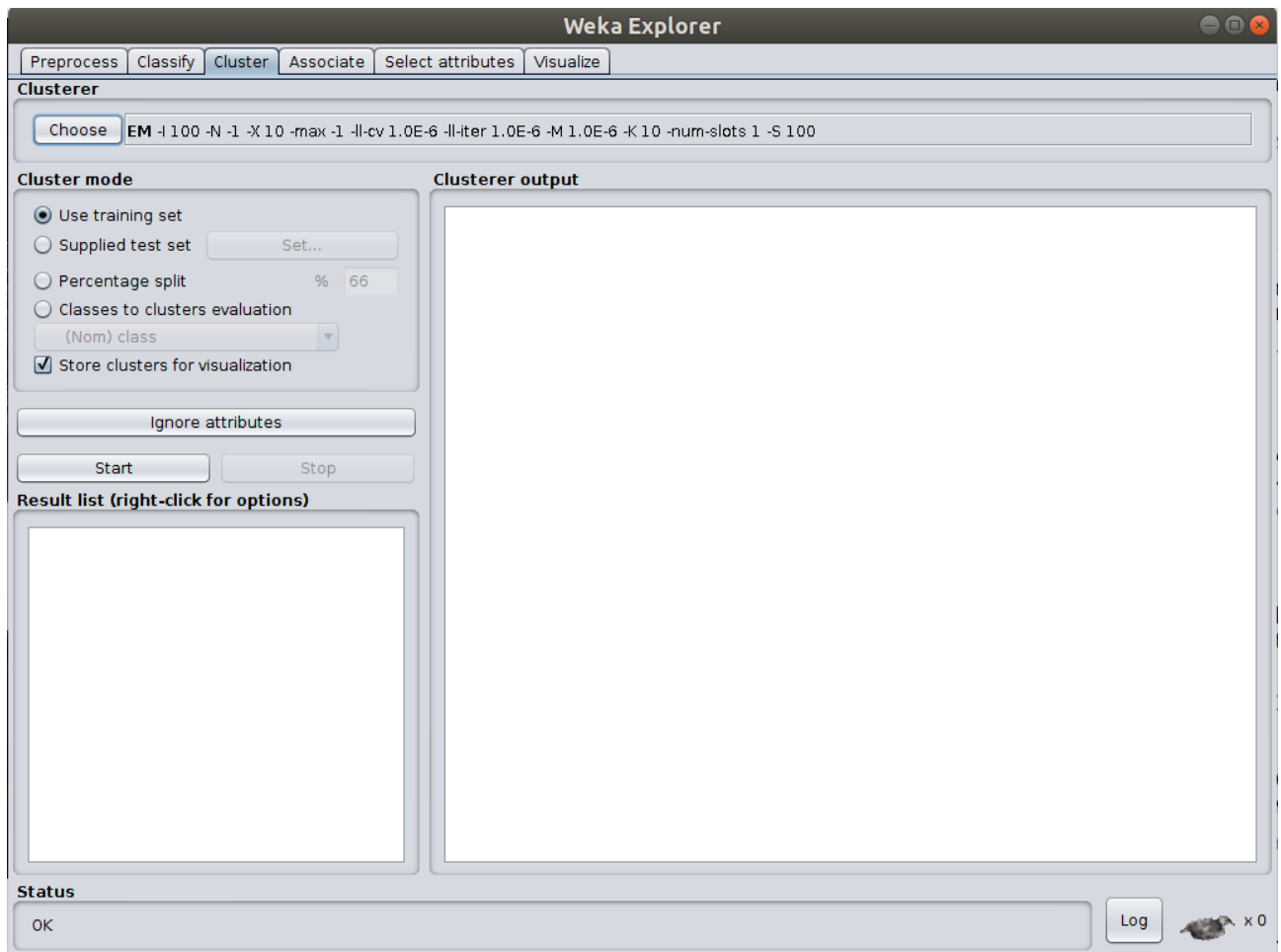
Status

OK Log x 0

- Kết quả cho thấy thuật toán cho độ chính xác của MultilayerPerceptron đạt được tới 97.33%.
- Lịch sử các lần chạy được liệt kê trong khung Result list.

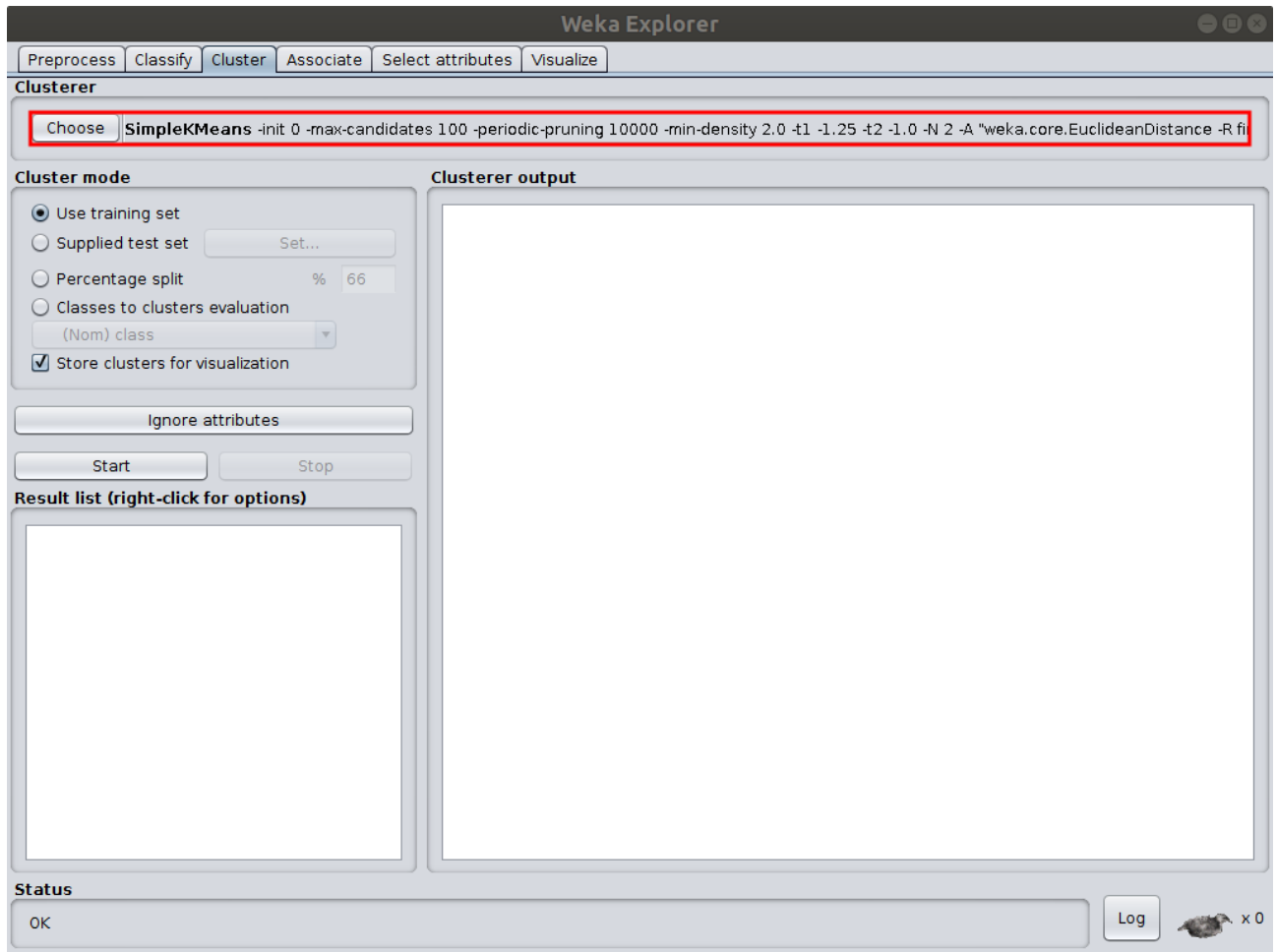
Cluster

Nhóm Cluster hỗ trợ các thuật toán gom nhóm dữ liệu.

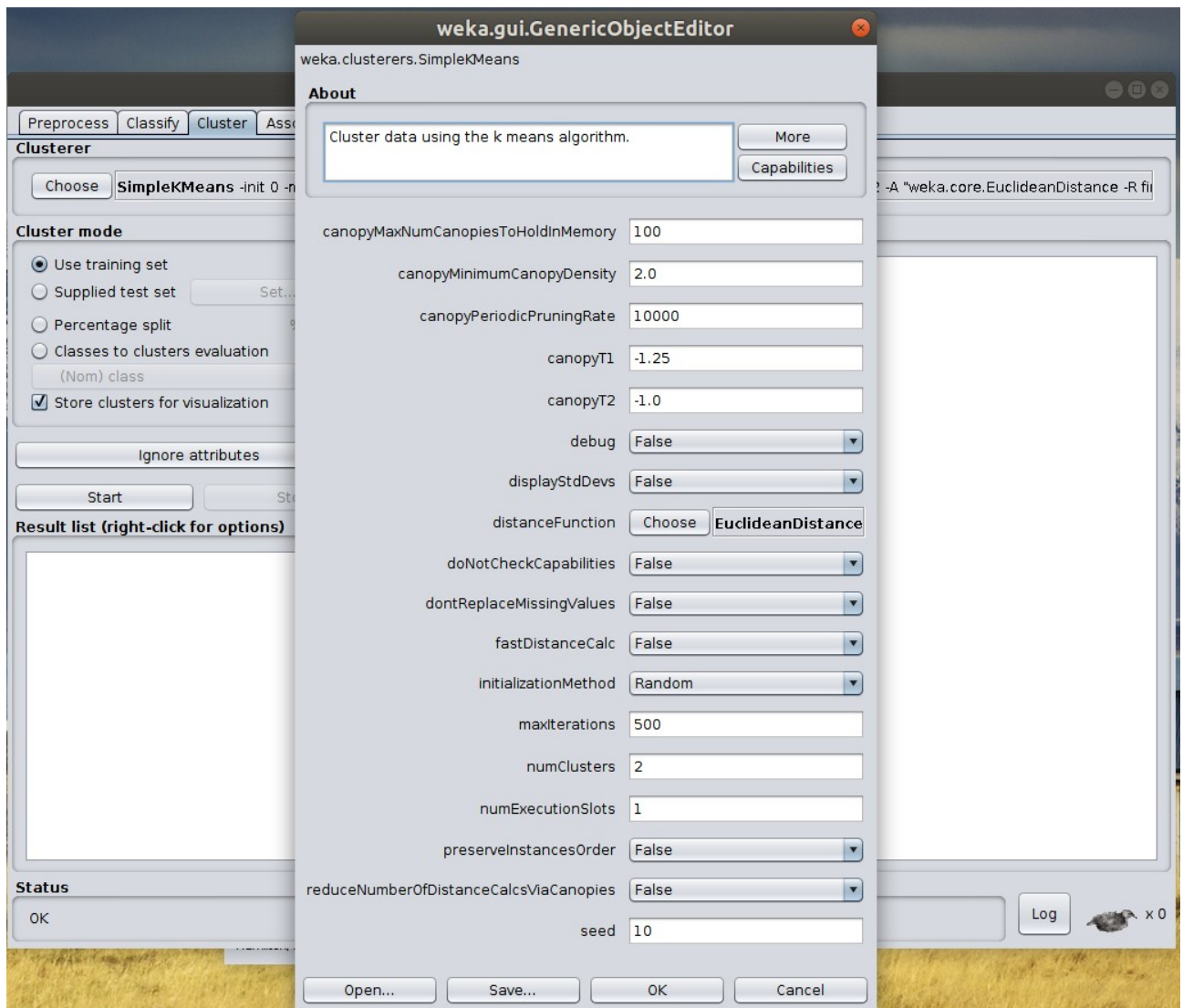


Ví dụ gom nhóm dữ liệu bằng thuật toán KmeanClustering:

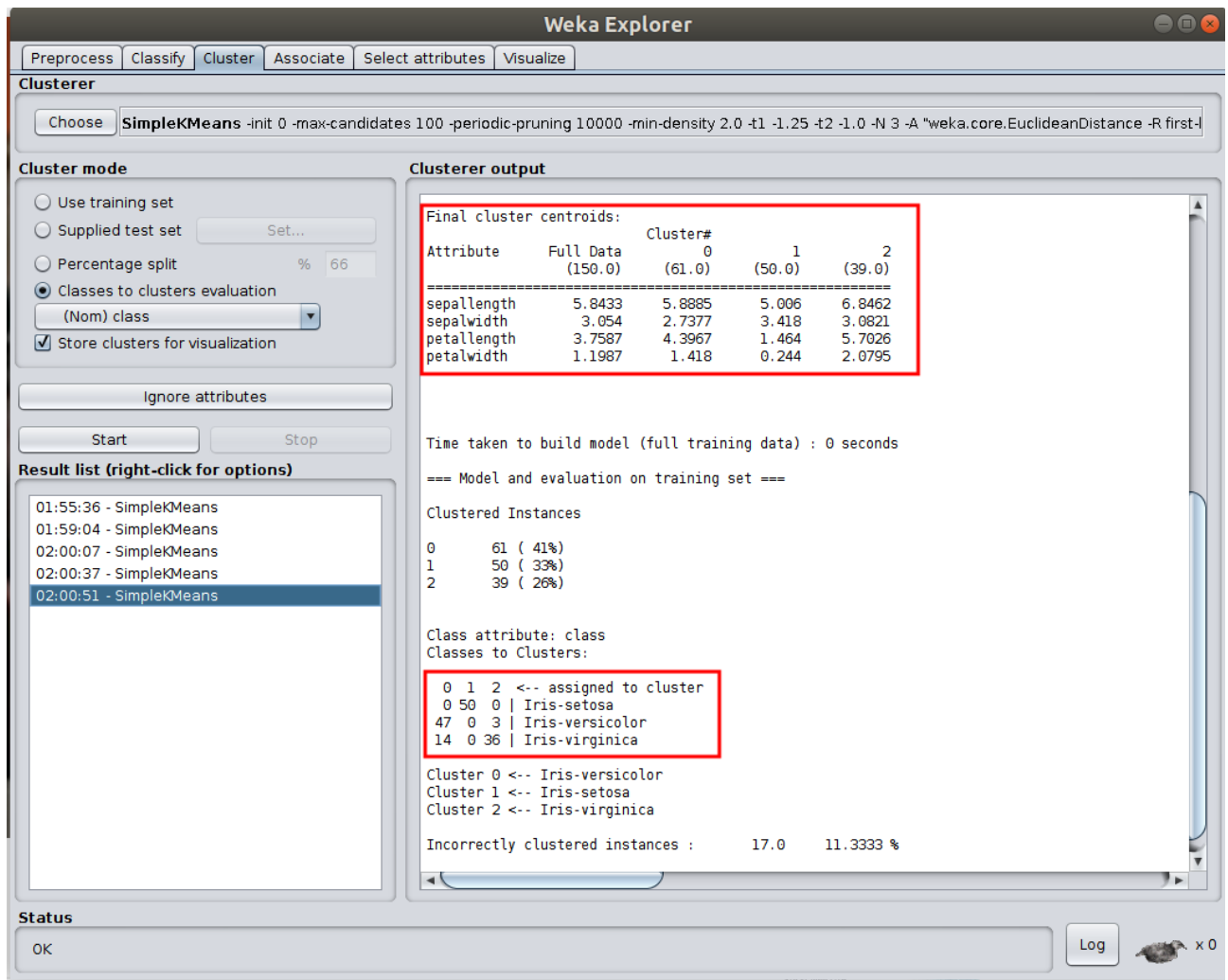
- Chọn Choose và chọn thuật toán SimpleKMeans trong mục clusterers.



- Ấn vào SimpleKMeans để tùy chỉnh riêng cho thuật toán KMean:



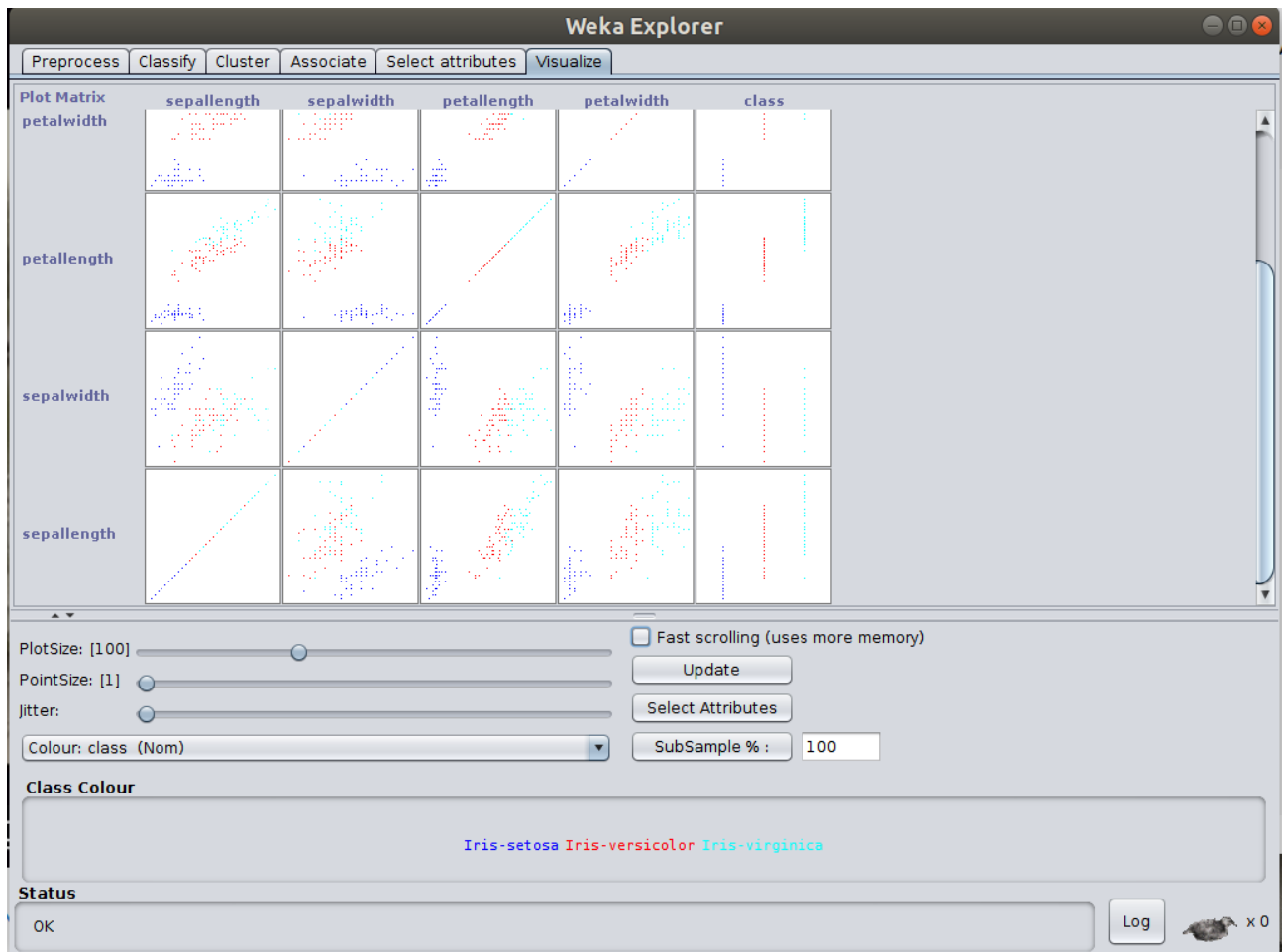
- Ở đây số lớp đang phân loại là 3 nên ta đặt $\text{numClusters} = 3$.
- Nhấn OK để chấp nhận tùy chỉnh.
- Ở Cluster mode chọn Classess to clusters evaluation là (Nom) class.
- Chọn Start để chạy thuật toán và đợi kết quả:



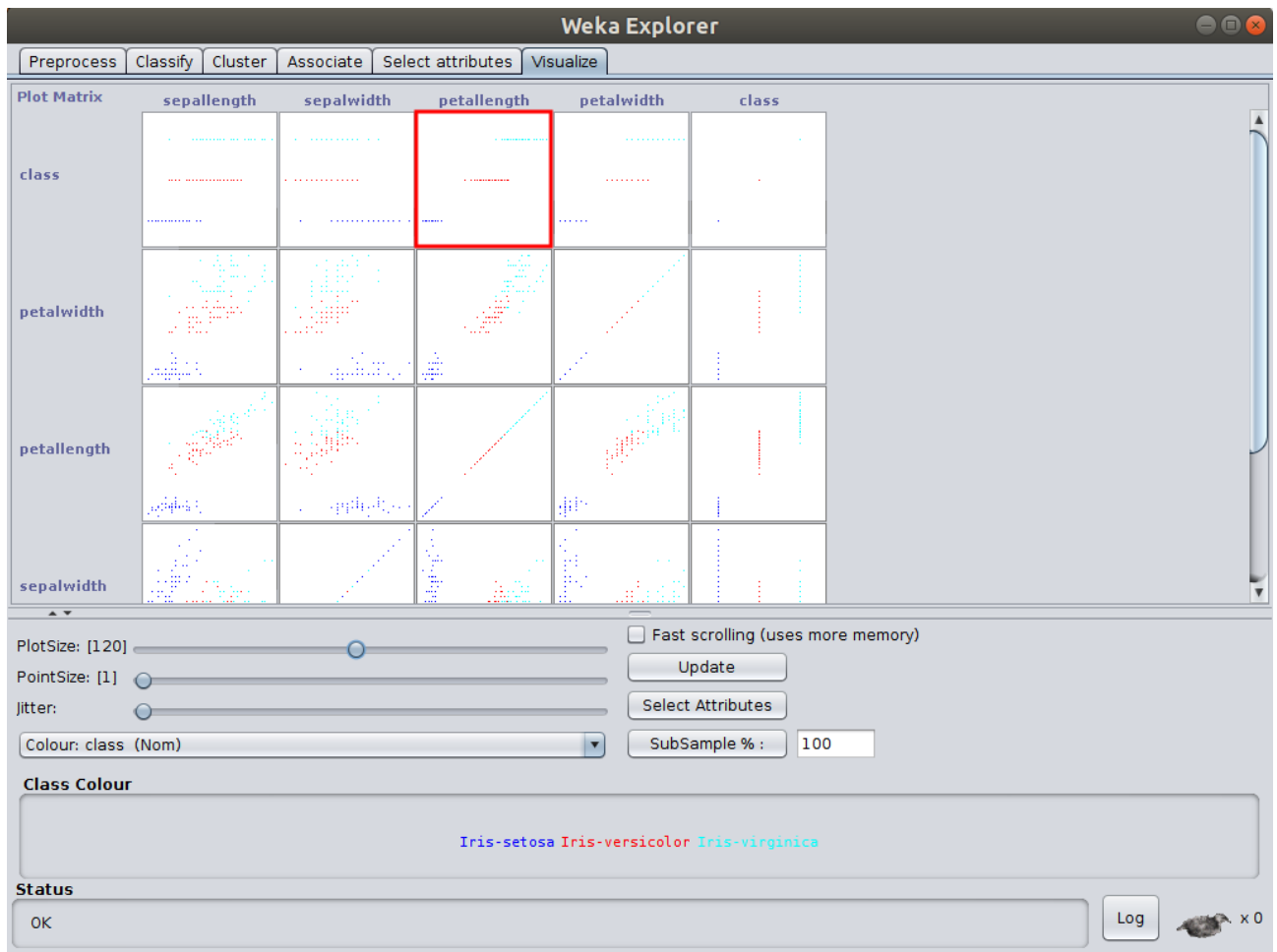
- Kết quả trên cho biết trọng tâm của 3 cụm dữ liệu (khung đỏ ở trên), và class tương ứng với mỗi cụm dữ liệu (khung đỏ ở dưới).

Visualize

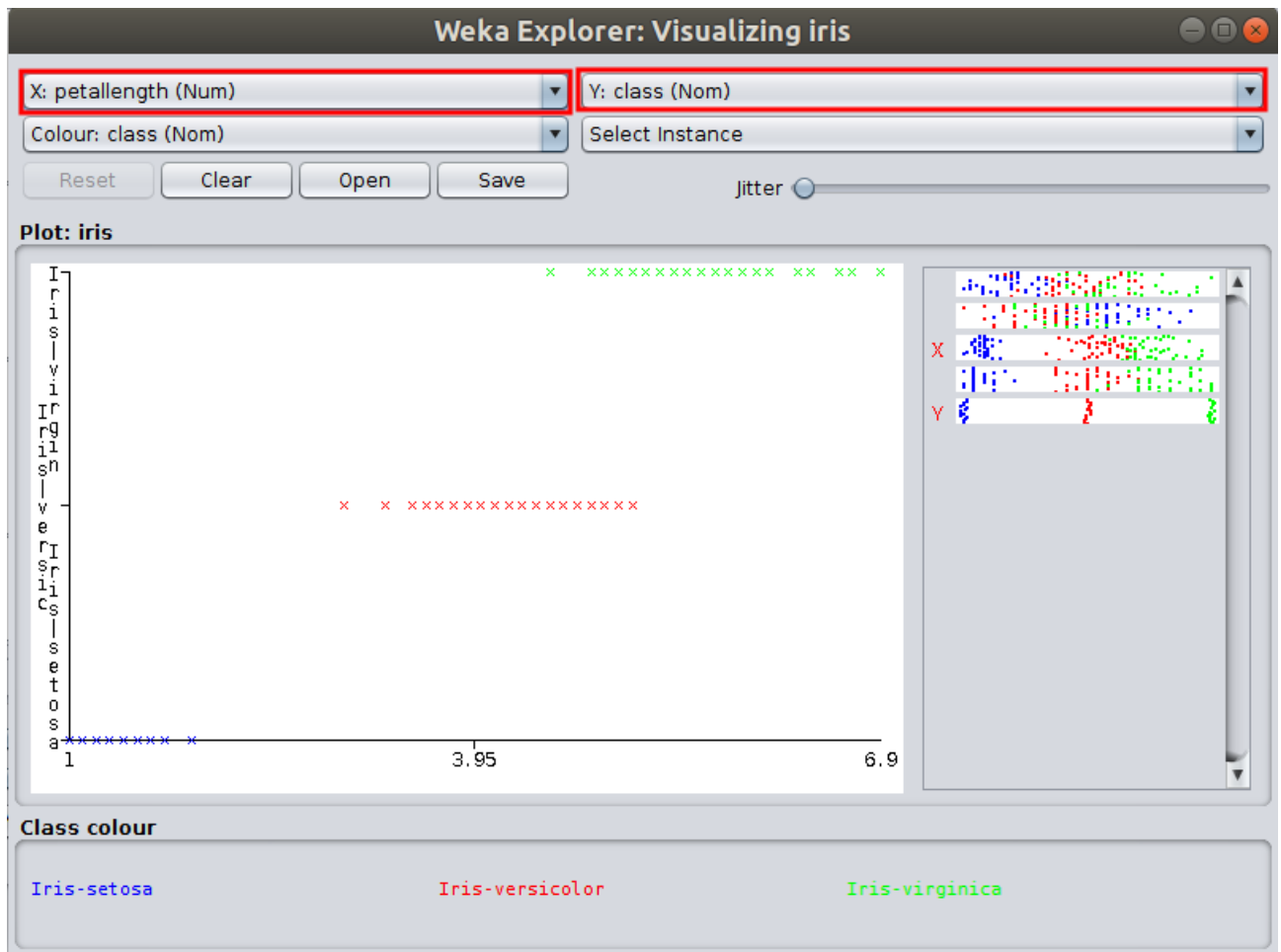
Visualize cung cấp các biểu đồ mô tả dữ liệu một cách trực quan.



- Có thể thấy với mỗi cặp thuộc tính trong bộ dữ liệu, ta có một biểu đồ quan hệ tương ứng.
- Ví dụ với quan hệ giữa class với petallength:
- Chọn biểu đồ ở hàng 'class' và cột 'petallength':

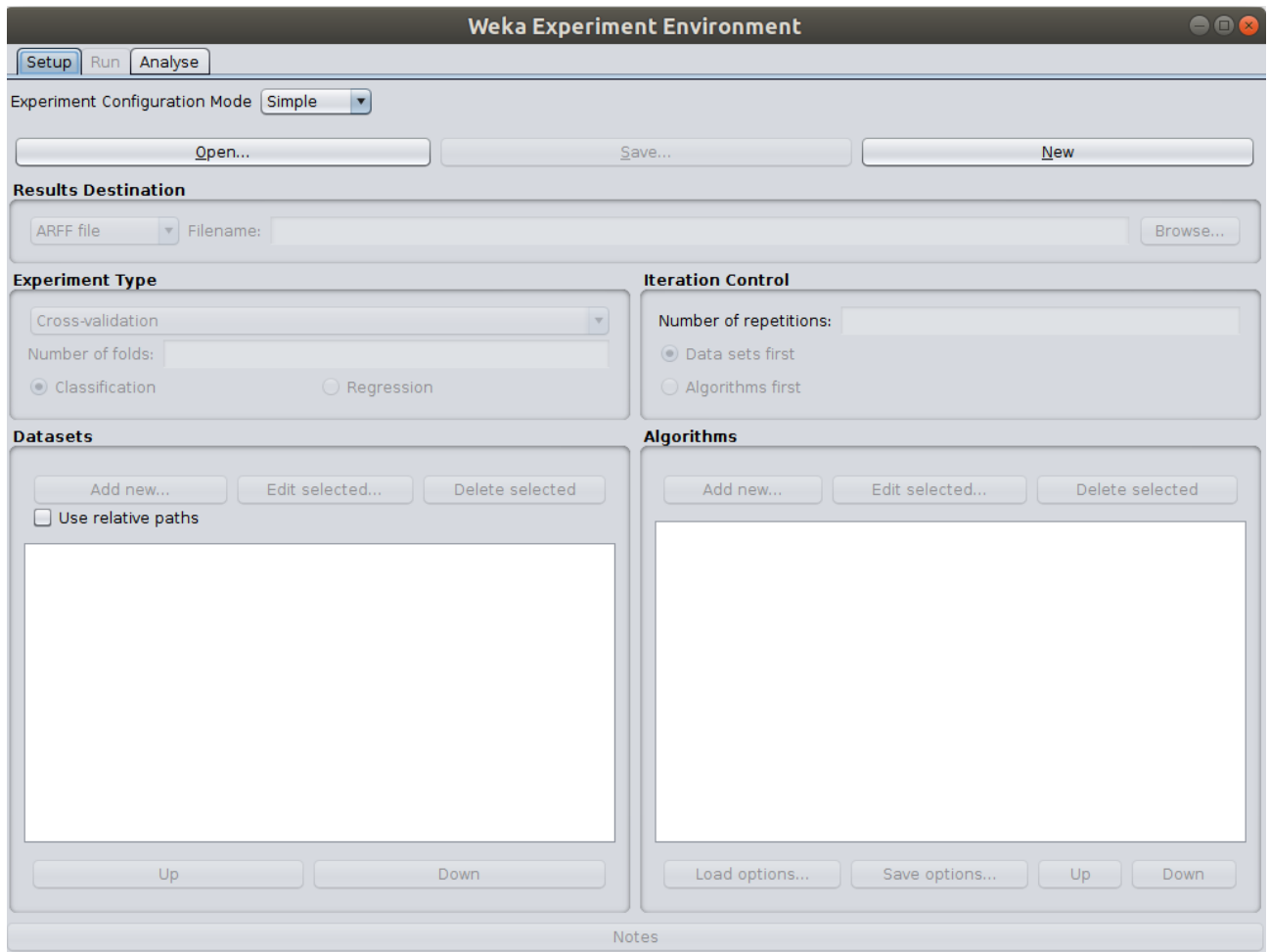


- Ở cửa sổ hiện ra ta thấy rõ hơn biểu đồ quan hệ giữa class và petalwidth trong khung Plot (trục X là petalwidth, Y là class):



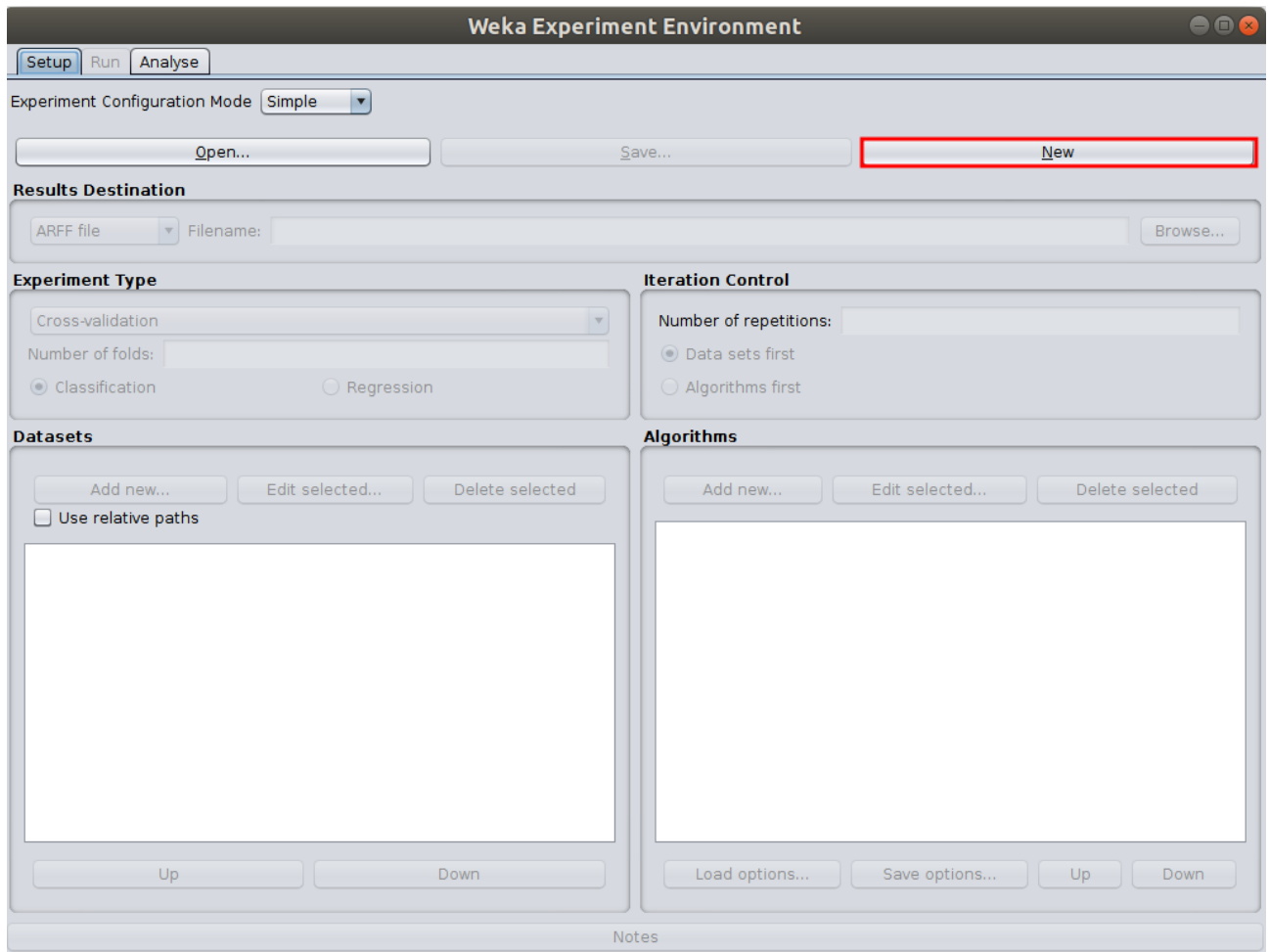
Experimenter

Ứng dụng Experimenter dùng để so sánh kết quả của các thuật toán trên các bộ dữ liệu khác nhau:

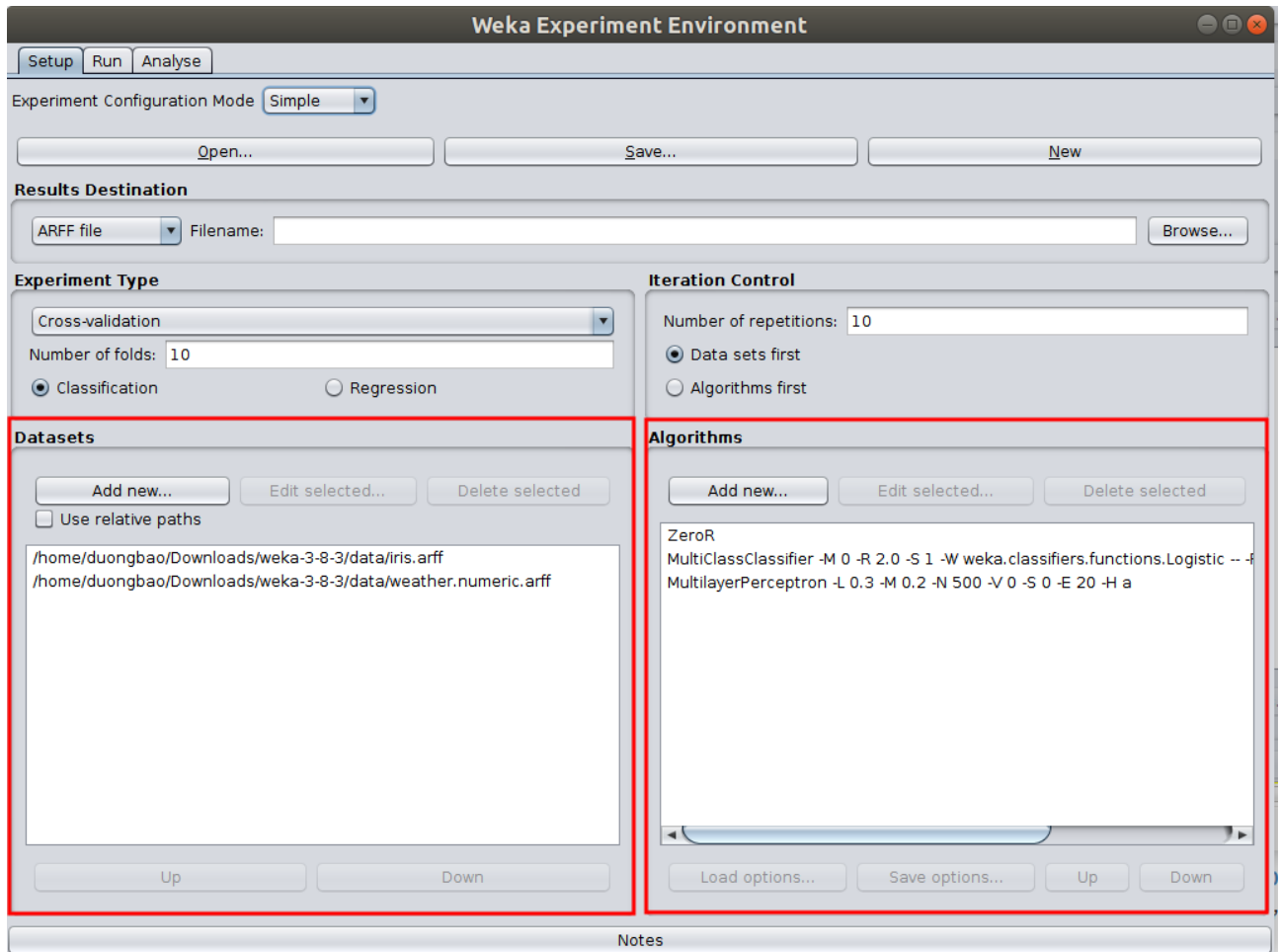


Ví dụ so sánh 3 thuật toán ZeroR, MultiClassClassifier và MultilayerPerceptron trên 2 bộ dữ liệu iris và weather:

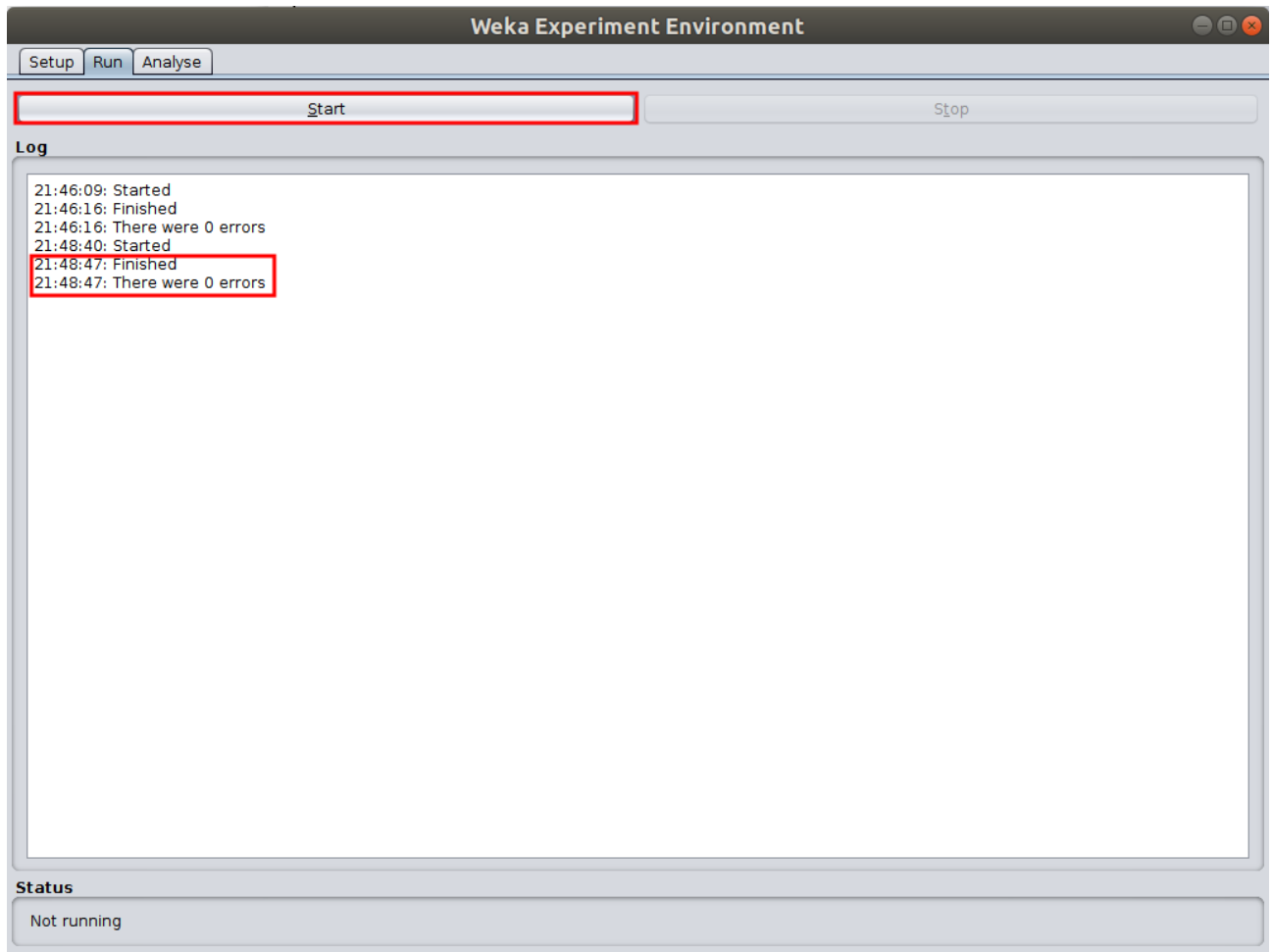
- Chọn New để tạo một thử nghiệm mới:



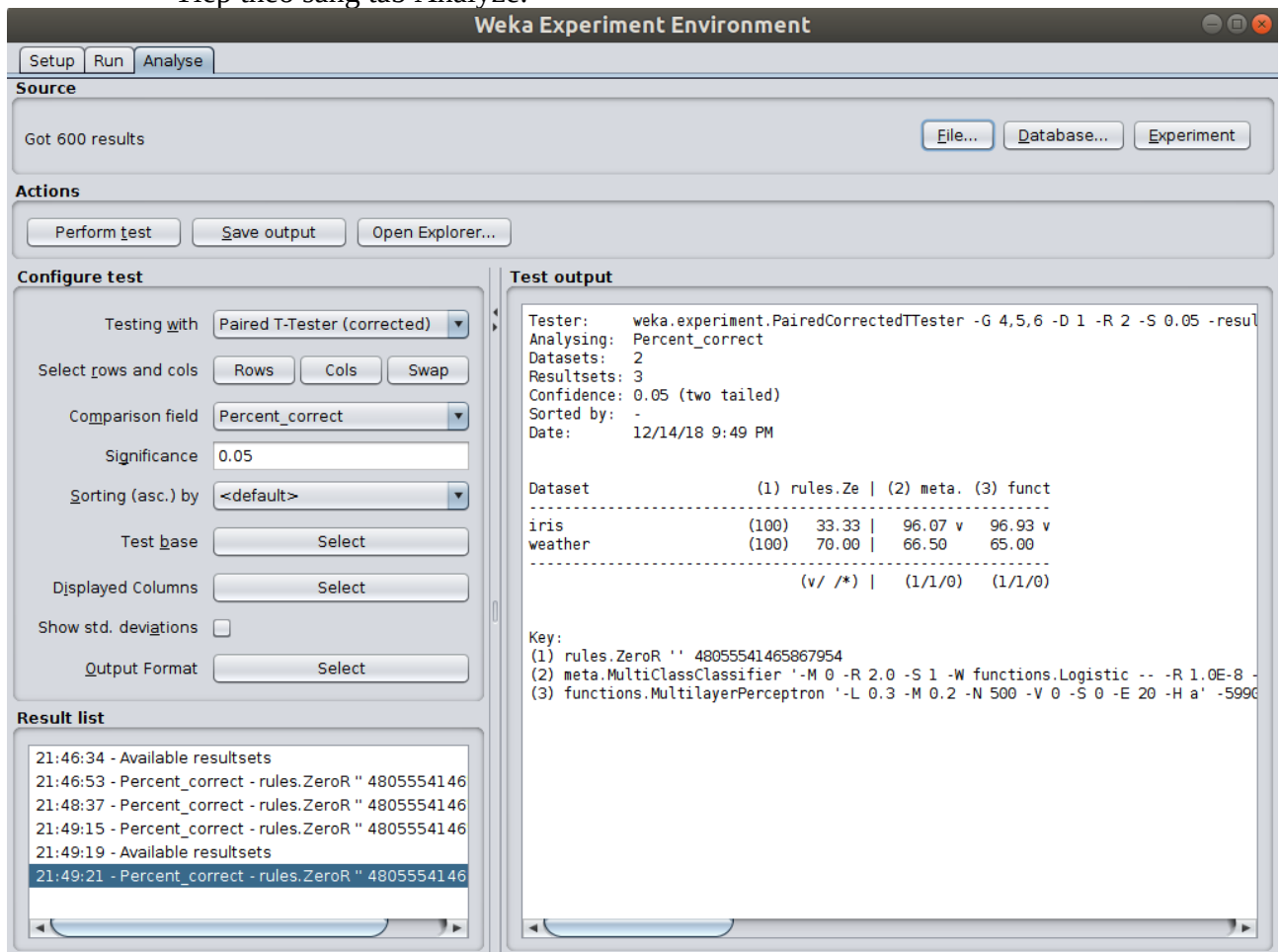
- Trong khung Dataset chọn Add new... và chọn load 2 file iris.arff và weather.numeric.arff trong thư mục data của thư mục giải nén weka. Trong Khung Algorithms chọn Add new... và chọn 3 thuật toán ZeroR, MultiClassClassifier và MultilayerPerceptron:



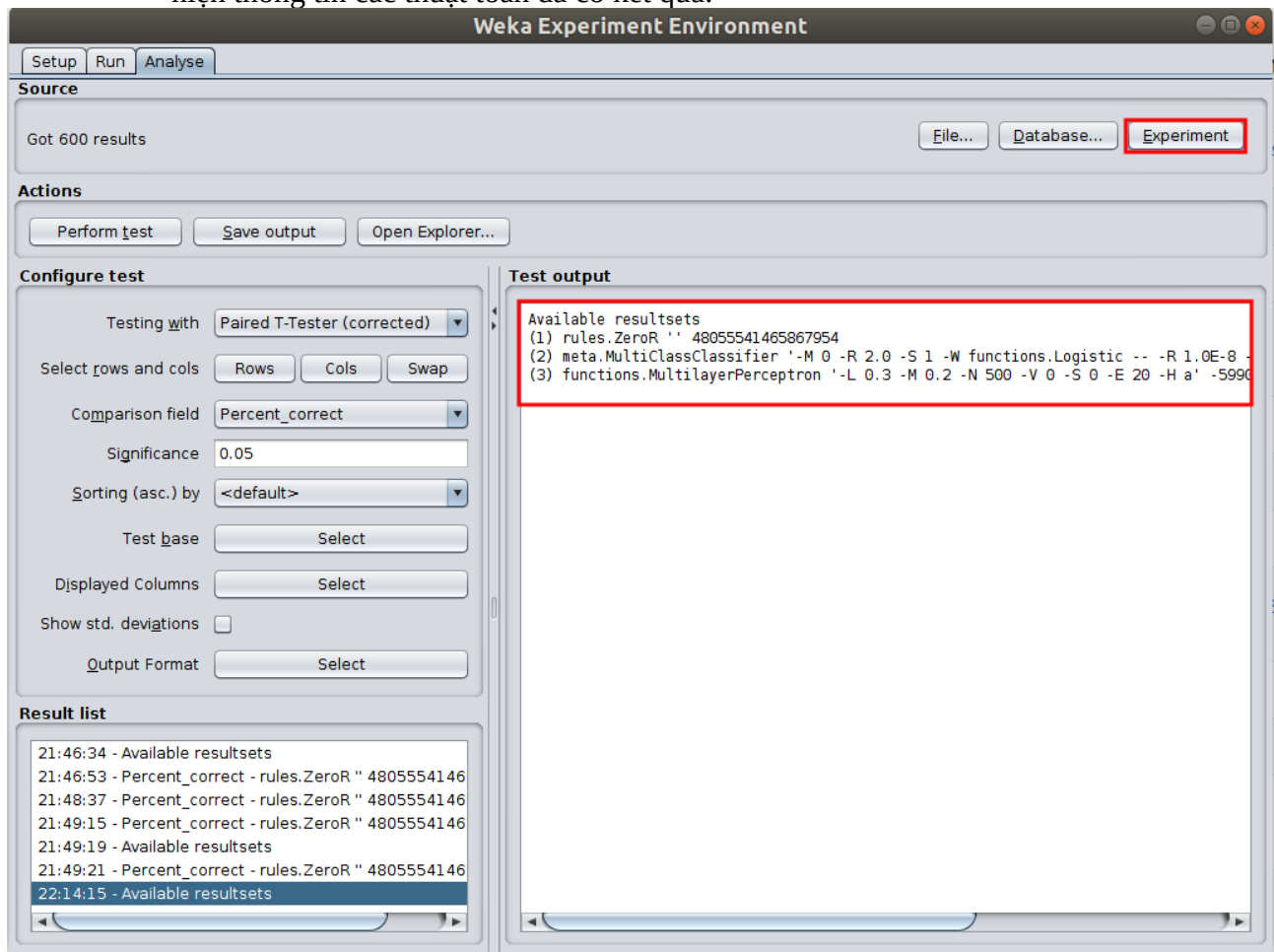
- Sau đó di chuyển sang tab Run, chọn Start và đợi các thuật toán chạy xong và thành công:



- Tiếp theo sang tab Analyze:



- Trong khung Source chọn Experimenter, quan sát thấy trong khung Test output hiện thông tin các thuật toán đã có kết quả:



- Trong khung Actions bấm Perform test để chạy so sánh và đợi kết quả:

Weka Experiment Environment

Setup Run Analyse

Source: Got 600 results File... Database... Experiment

Actions: Perform test Save output Open Explorer...

Configure test

Testing with: Paired T-Tester (corrected)

Select rows and cols: Rows Cols Swap

Comparison field: Percent_correct

Significance: 0.05

Sorting (asc.) by: <default>

Test base: Select

Displayed Columns: Select

Show std. deviations: ☐

Output Format: Select

Test output

```

Tester: weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result
Analysing: Percent_correct
Datasets: 2
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by: -
Date: 12/14/18 10:16 PM

```

| Dataset | (1) rules.Ze | (2) meta. | (3) funct |
|---------|--------------|-----------|-----------|
| iris | (100) 33.33 | 96.07 v | 96.93 v |
| weather | (100) 70.00 | 66.50 | 65.00 |
| | (v/ /*) | (1/1/0) | (1/1/0) |

Key:

```

(1) rules.ZeroR '' 48055541465867954
(2) meta.MultiClassClassifier '-M 0 -R 2.0 -S 1 -W functions.Logistic -- -R 1.0E-8 -
(3) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a' -5996

```

Result list

```

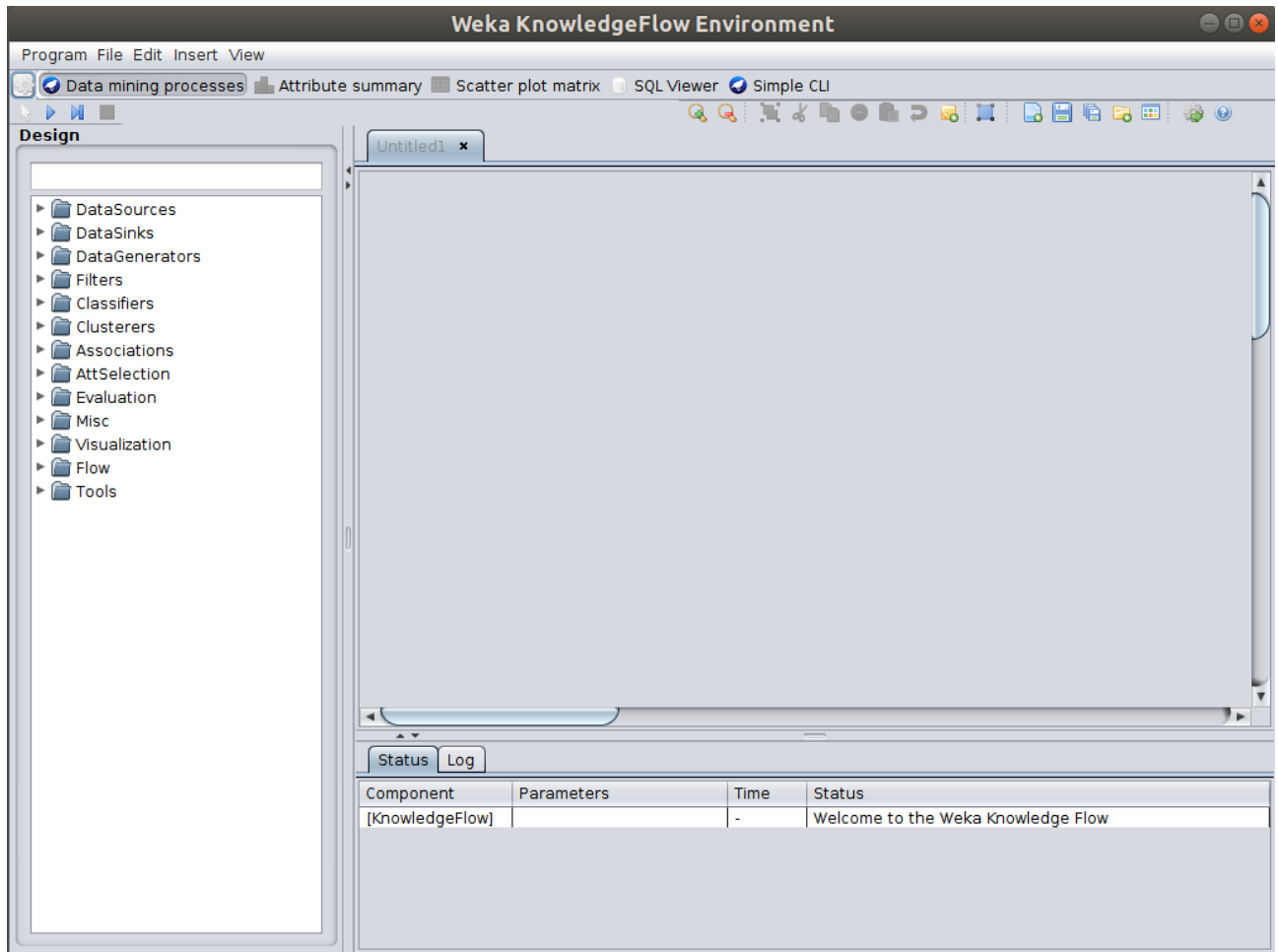
21:46:53 - Percent_correct - rules.ZeroR " 48055541
21:48:37 - Percent_correct - rules.ZeroR " 48055541
21:49:15 - Percent_correct - rules.ZeroR " 48055541
21:49:19 - Available resultsets
21:49:21 - Percent_correct - rules.ZeroR " 48055541
22:14:15 - Available resultsets
22:16:53 - Percent_correct - rules.ZeroR " 48055541

```

- Trong Test output lúc này có một bảng 2 hàng 3 cột mô tả kết quả so sánh giữa 3 thuật toán trên 2 bộ dữ liệu Iris và Weather, giá trị kết quả là phần trăm độ chính xác.

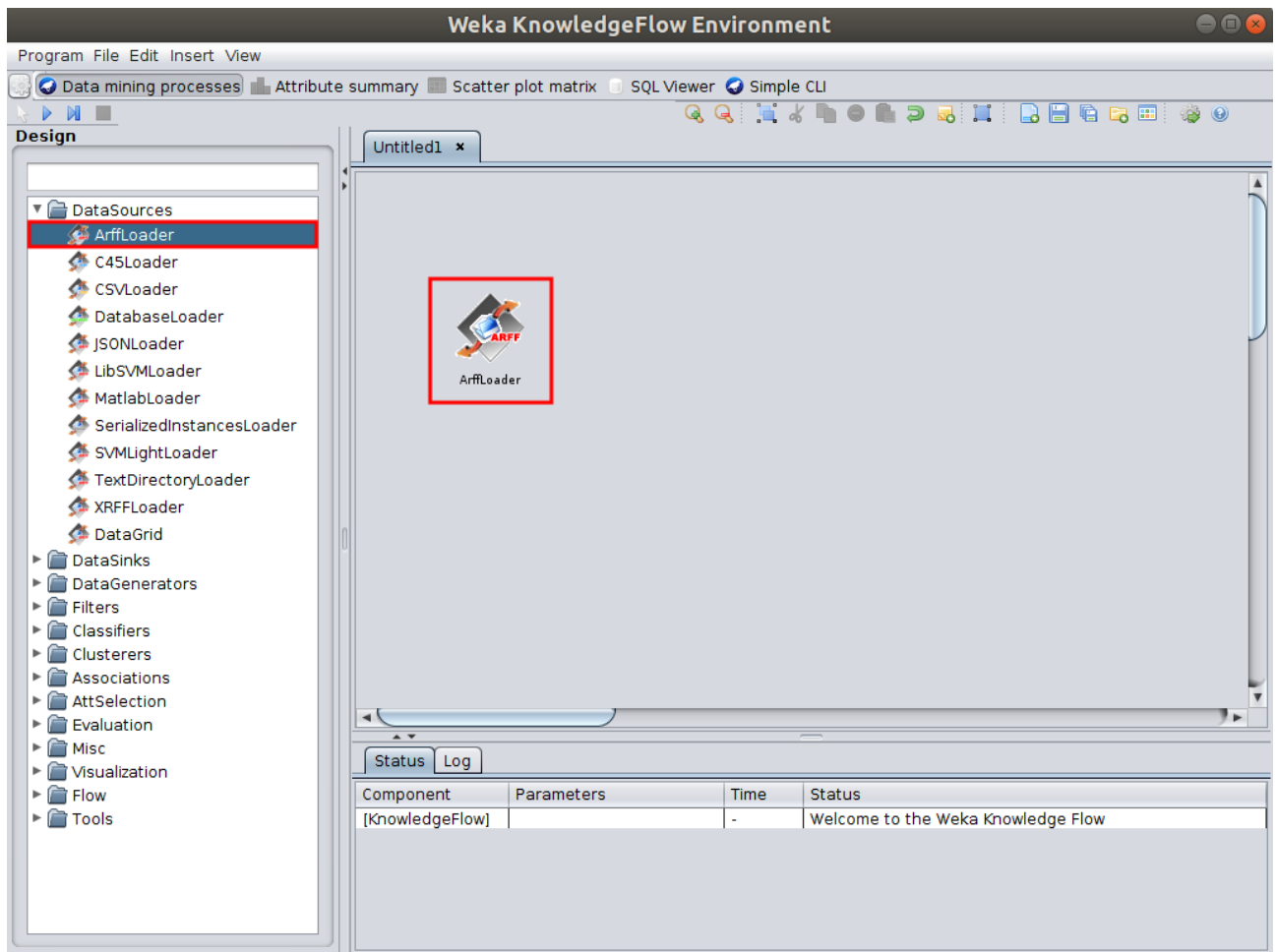
KnowledgeFlow

KnowledgeFlow cung cấp cho người dùng các chức năng giống như Explorer nhưng với môi trường kéo thả trực quan.

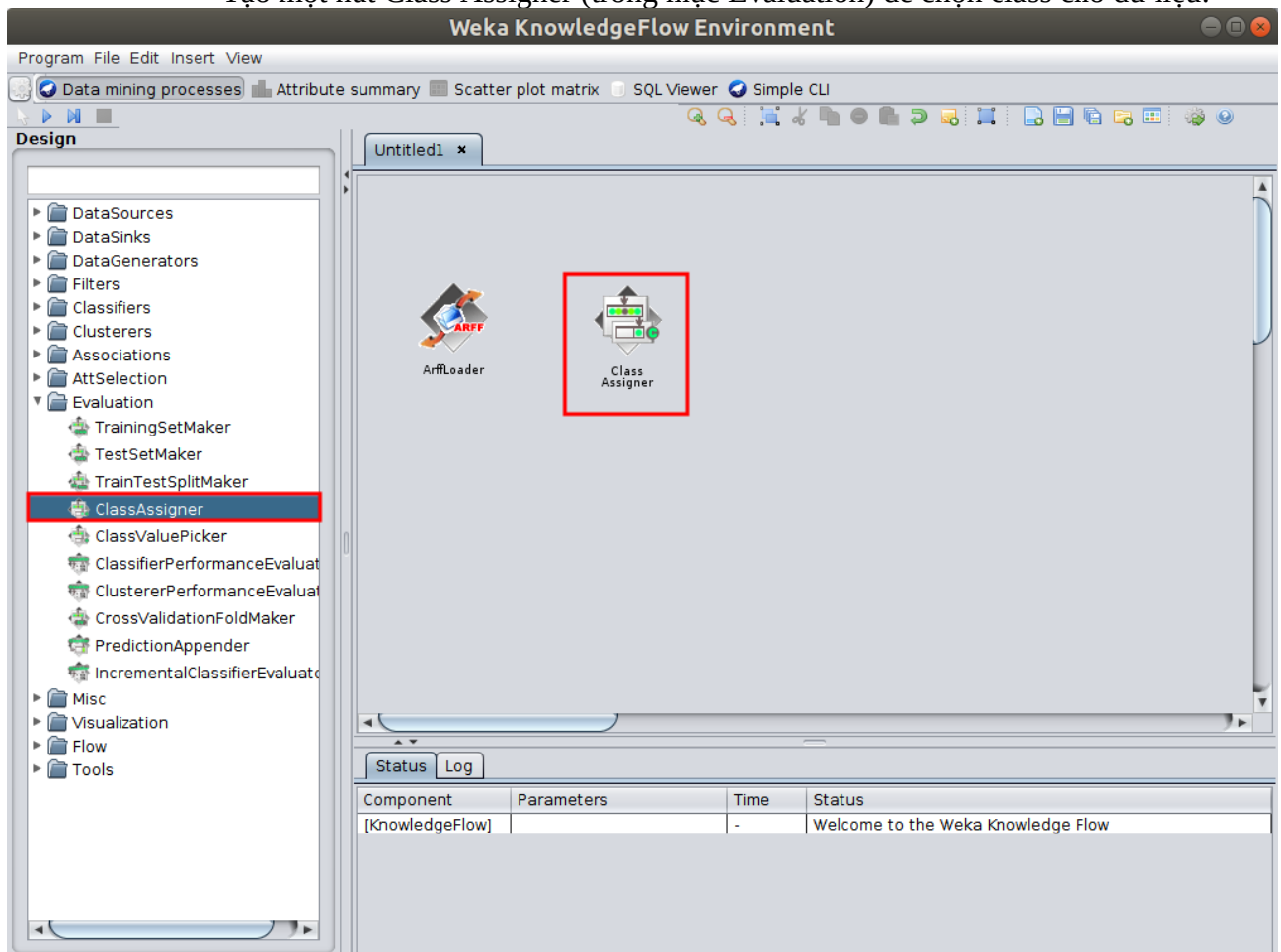


Lấy ví dụ với mô hình MultilayerPerceptron trên bộ dữ liệu Iris. Bước đầu tiên là xây dựng luồng thuật toán, sau đó mới có thể chạy thuật toán:

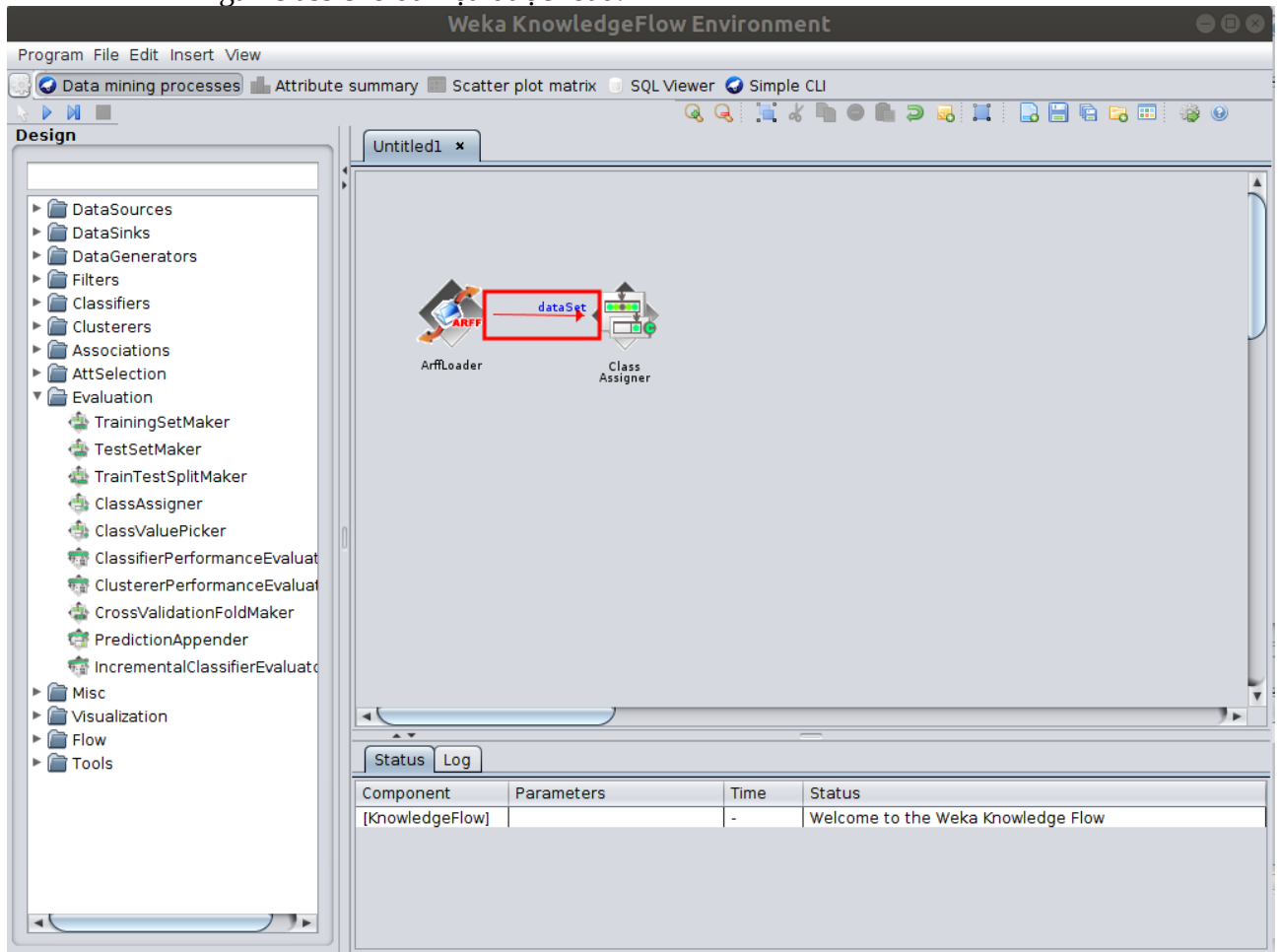
- Xây dựng thuật toán:
 - Đầu tiên cần xác định nguồn dữ liệu. Trong khung Design, chọn mục DataSources và chọn ArffLoader, sau đó nhấp chuột vào một vị trí trên khung làm việc bên phải để tạo đối tượng ArffLoader:



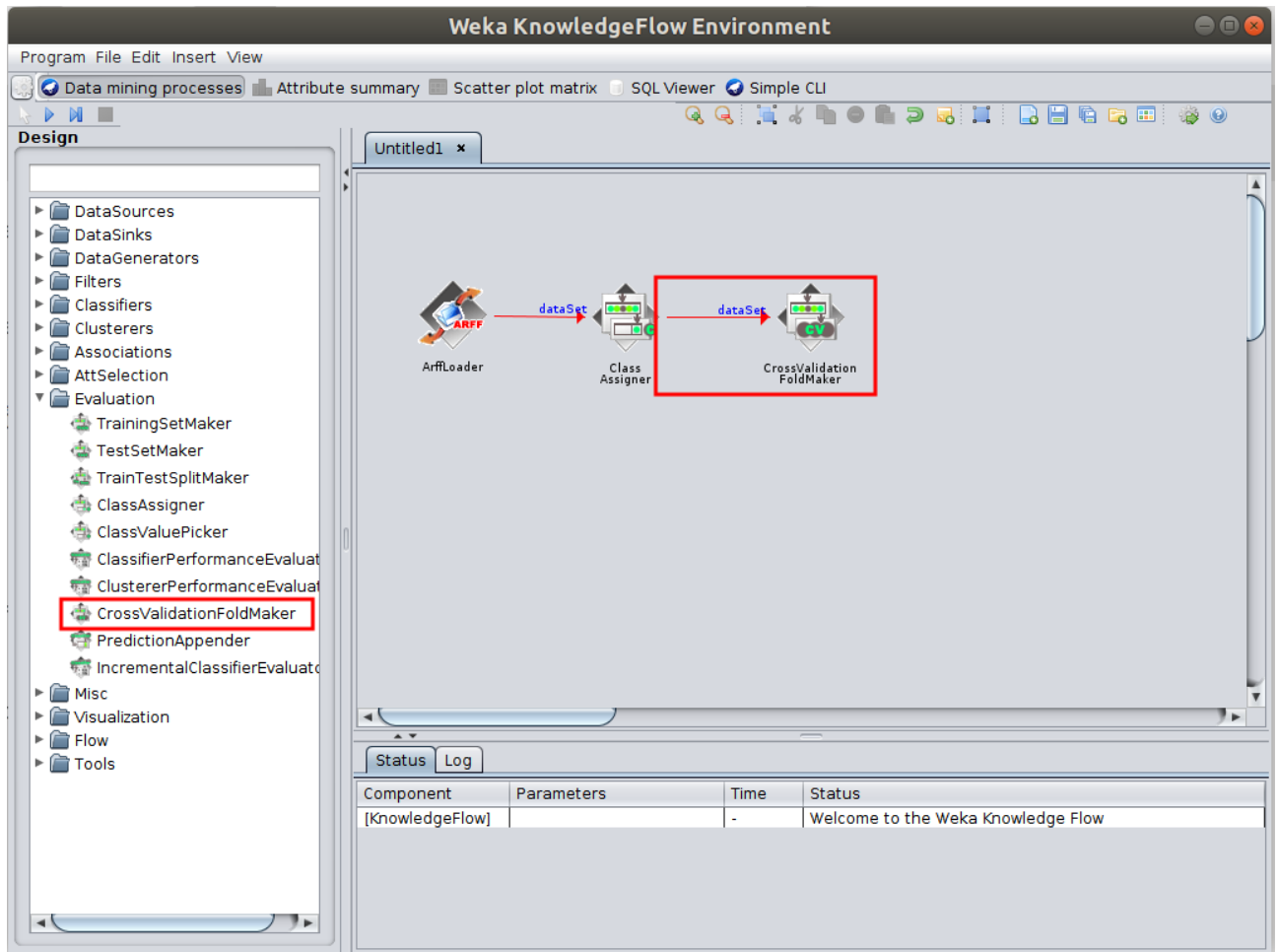
- Tạo một nút Class Assigner (trong mục Evaluation) để chọn class cho dữ liệu:



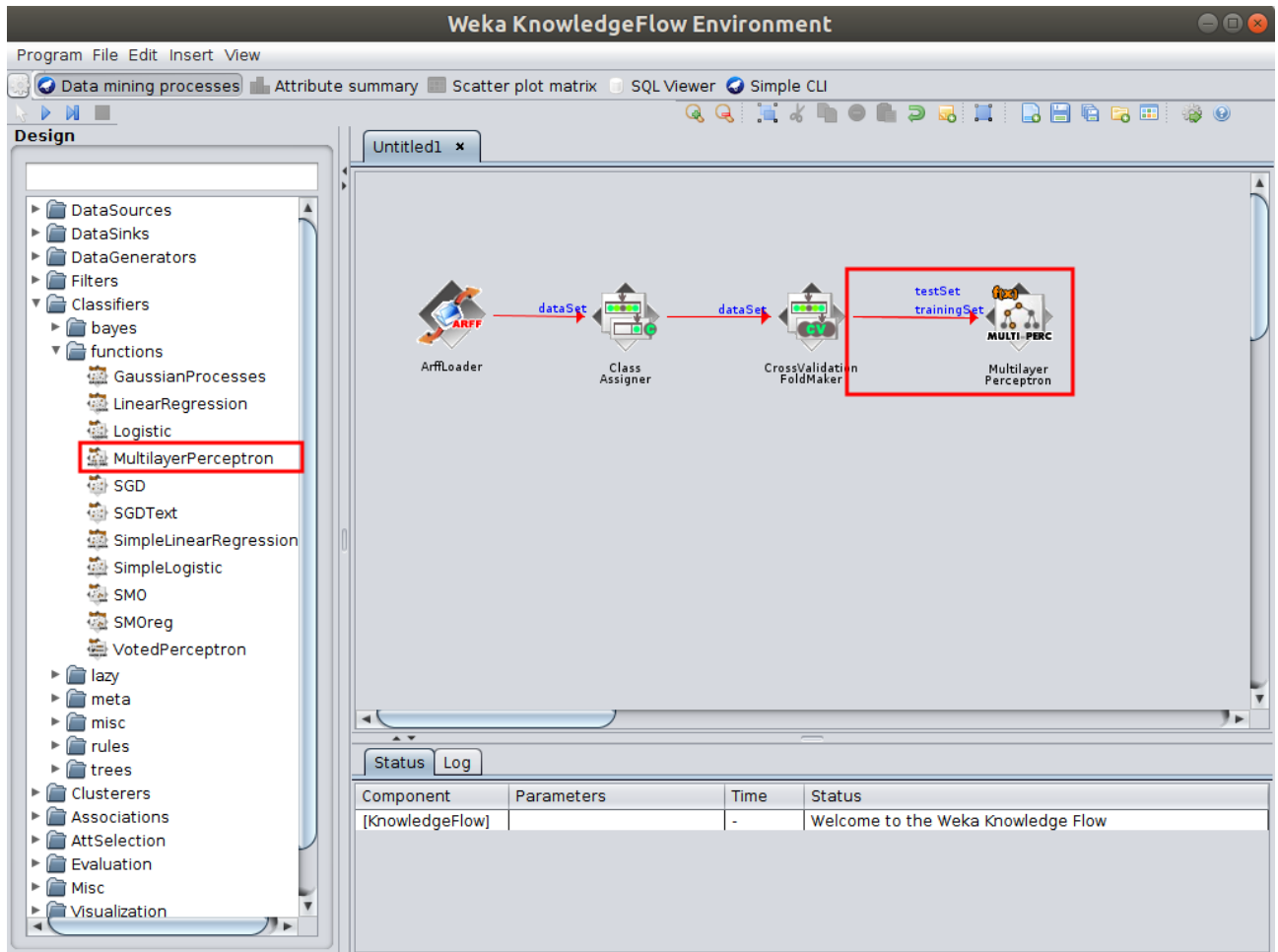
- Chuột phải ở nút ArffLoader, chọn dataSet và nối với nút Class Assigner để gán class cho dữ liệu được load.



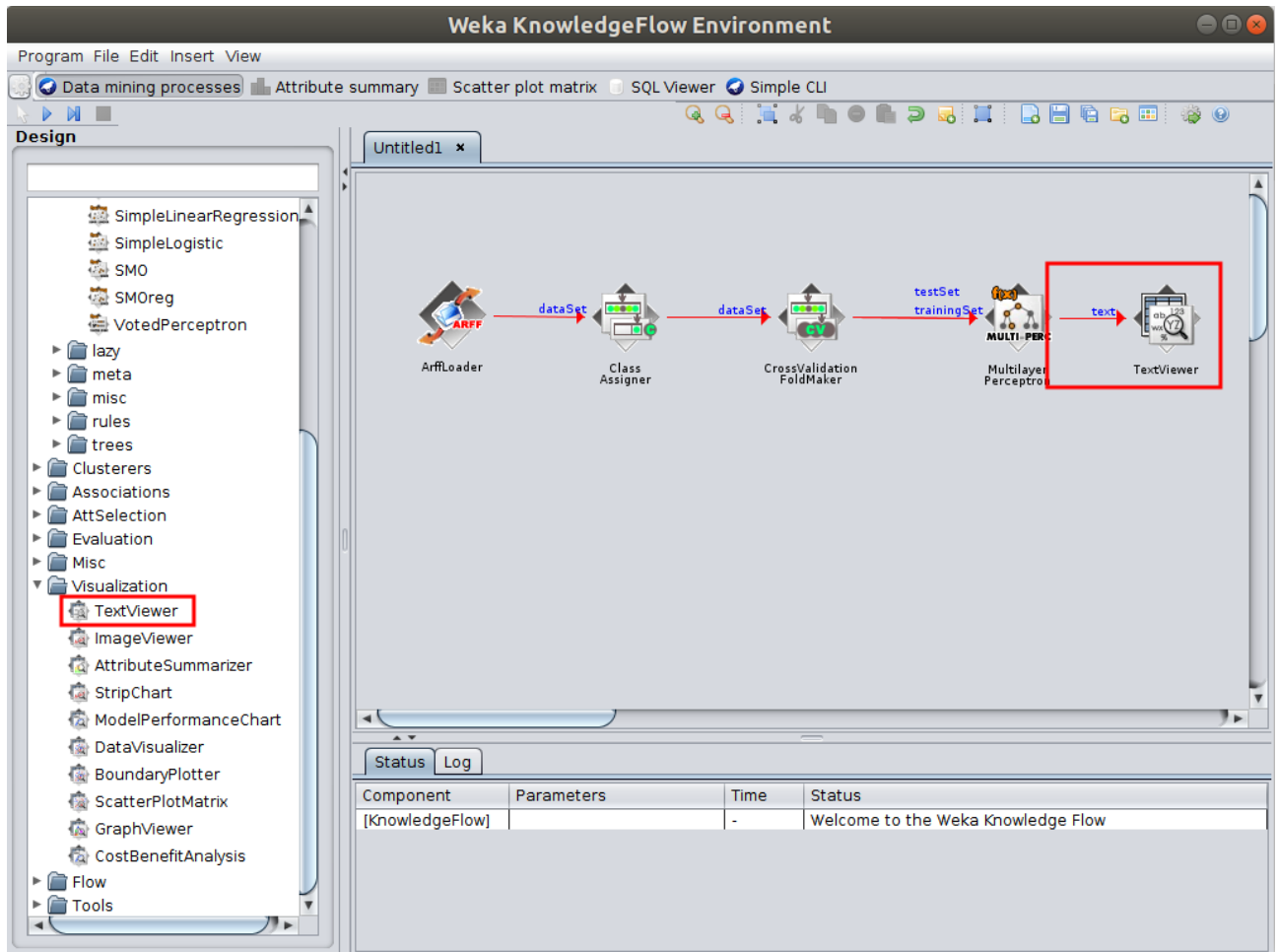
- Sau khi dữ liệu đã được gán lớp, ta chuyển nó qua nút CrossValidation FoldMaker (trong mục Evaluation) để chia dữ liệu thành các bộ train và test.



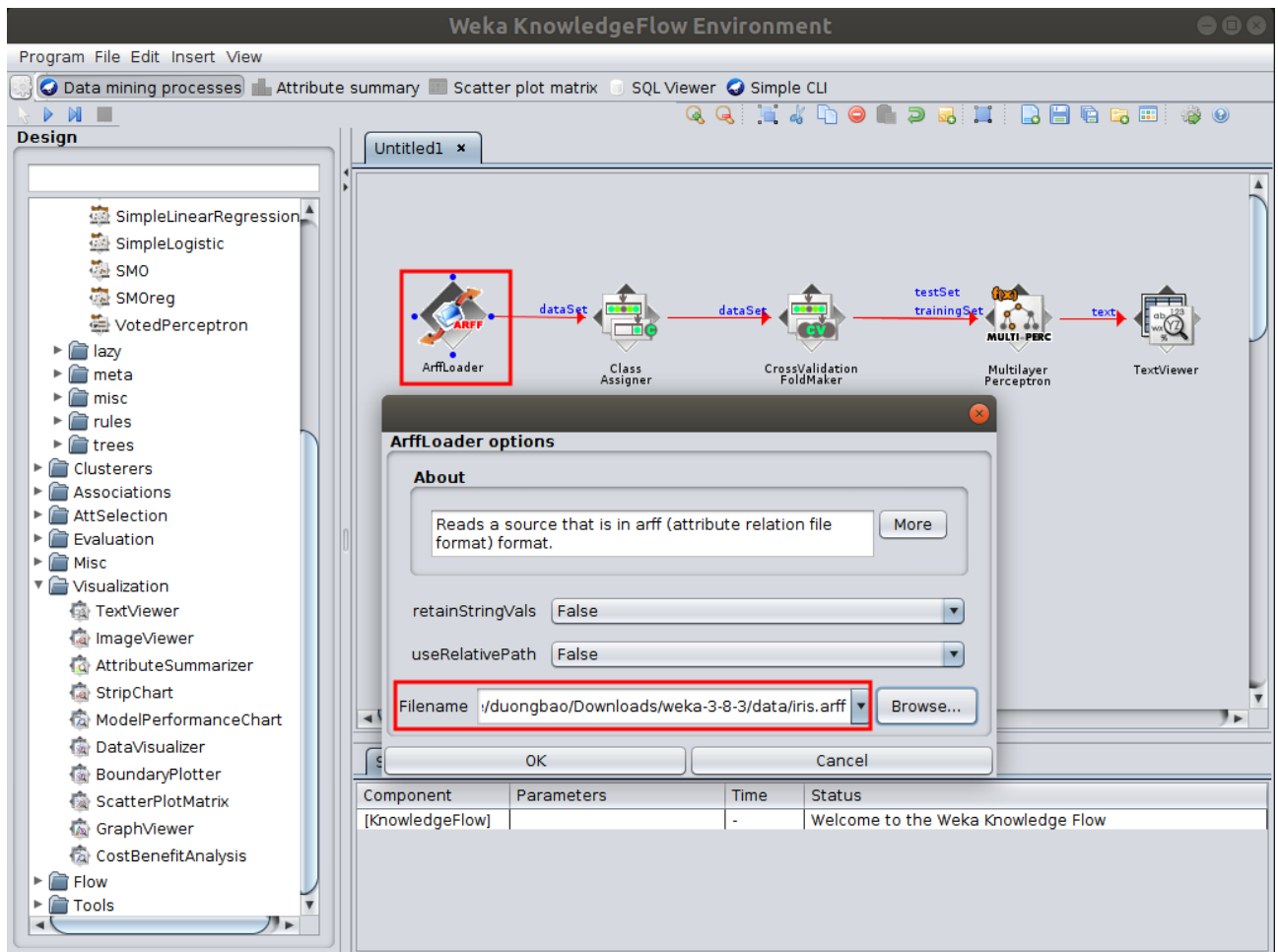
- Tiếp đến ta chọn thuật toán để thao tác trên dữ liệu. Ta sẽ chọn thuật toán MultilayerPerceptron (trong mục Classifier/functions). Sau đó chuyển cả 2 bộ trainingSet và testSet từ CrossValidation FoldMaker sang Multilayer Perceptron:



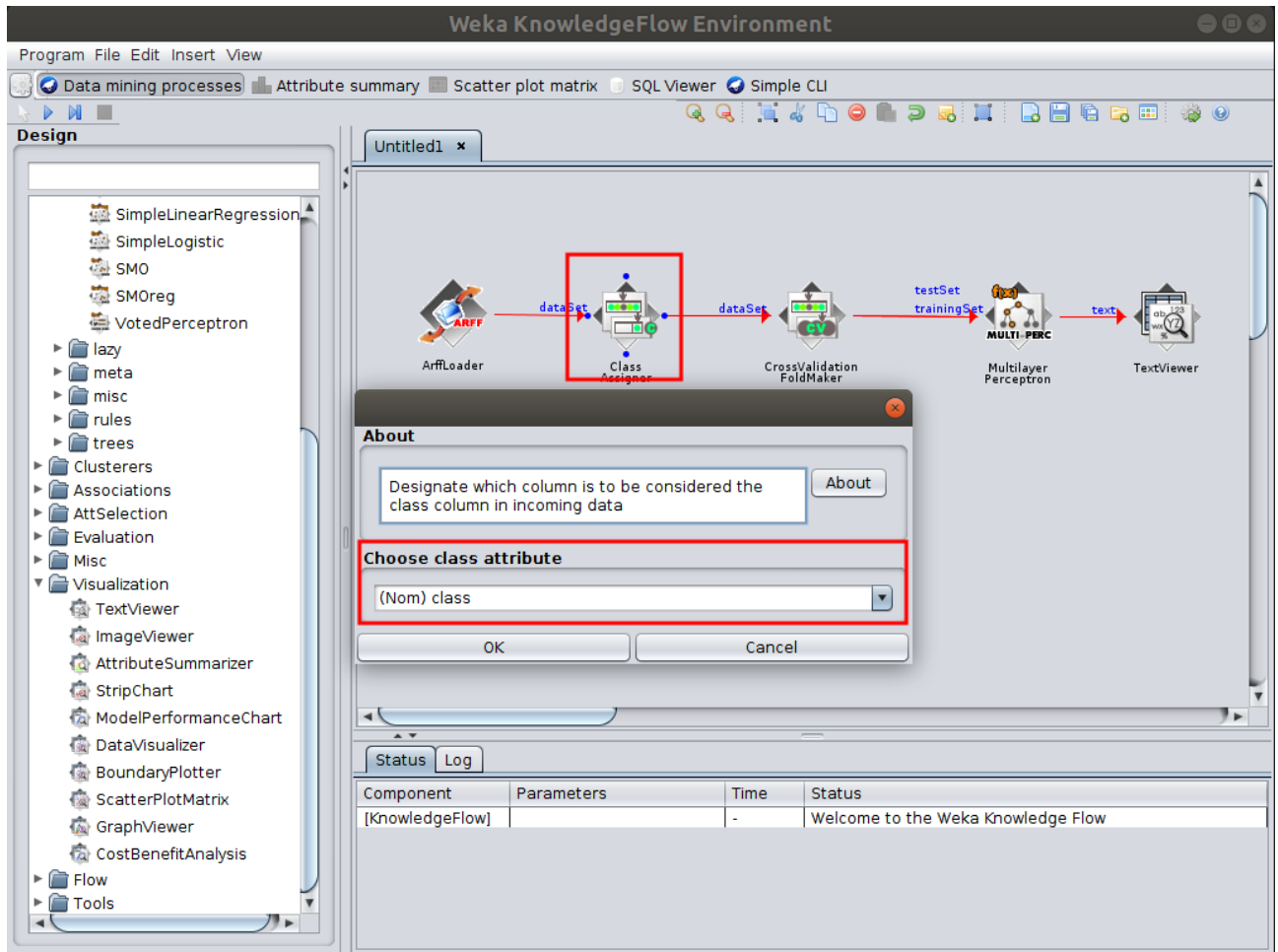
- Cuối cùng để nhận được kết quả từ thuật toán, ta lấy kết quả dạng text của MultilayerPerceptron truyền ra một nút TextViewer (trong mục Visualization):



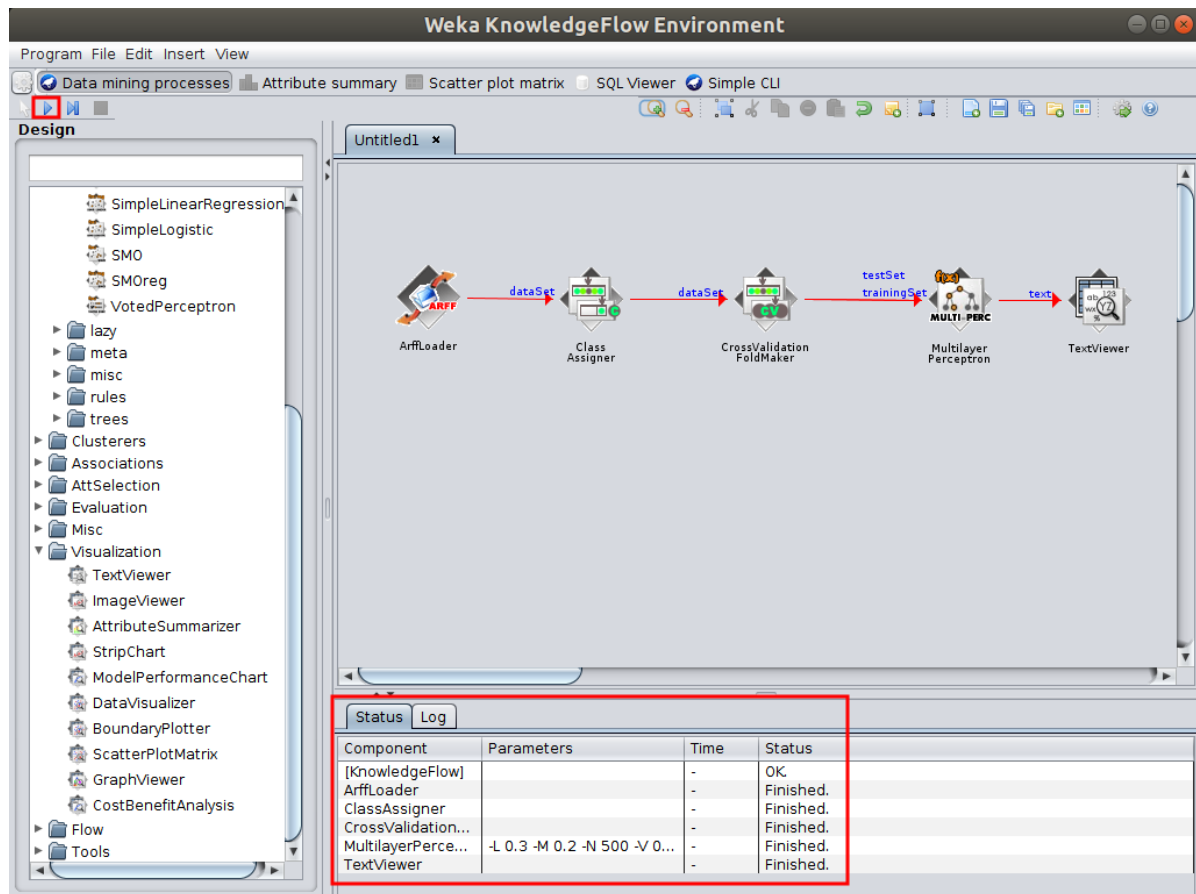
- Chạy thuật toán:
 - Nhấp đúp chuột tại ArffLoader, mục Filename chọn load file iris.arff:



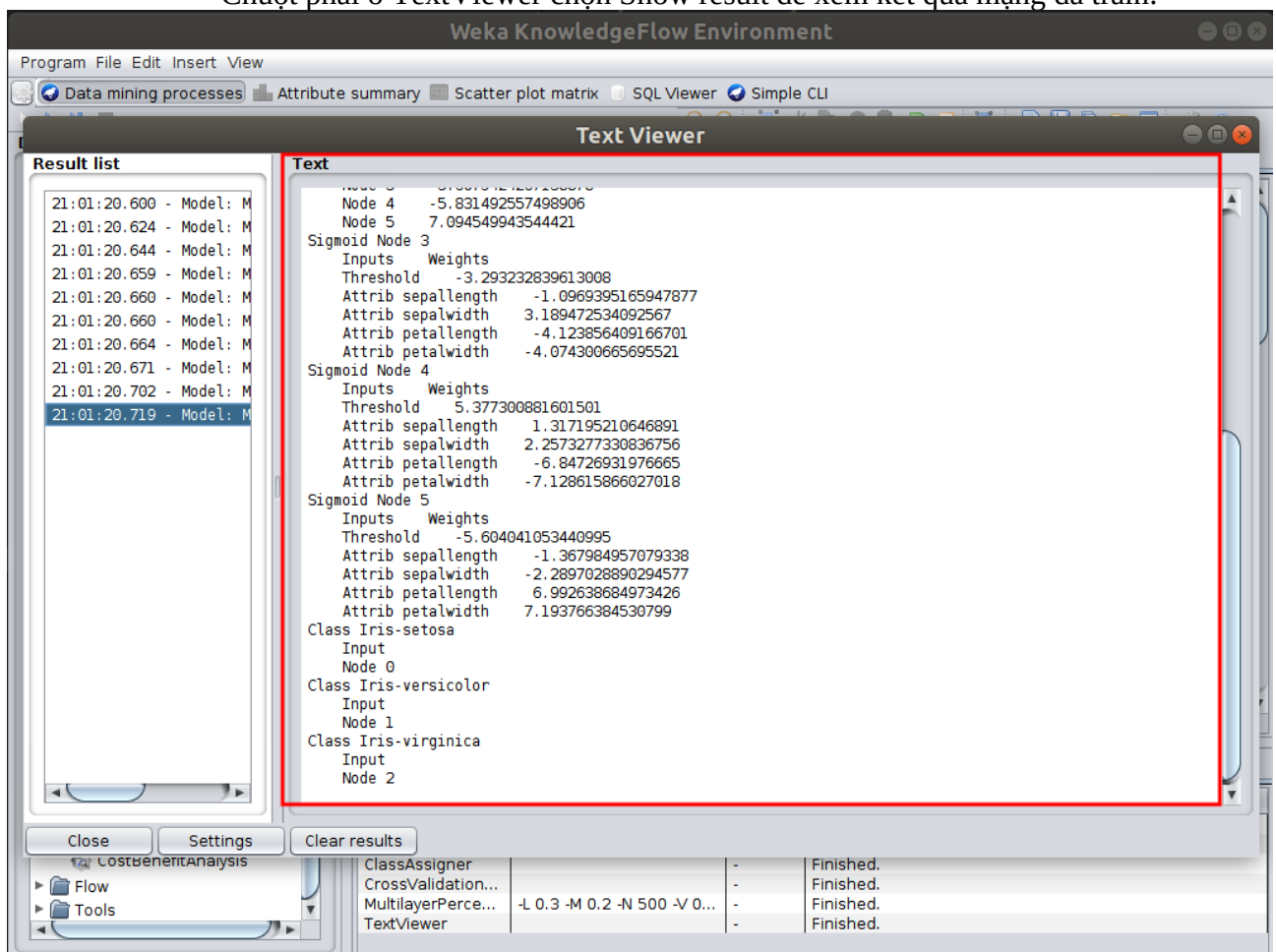
- Nhấp đúp ở Class Assigner, trong mục Choose class attribute chọn (Nom) class:



- Có thể tùy chỉnh một số thuộc tính tùy chọn khác trong CrossValidation FoldMaker, MultilayerPerceptron và TextViewer.
- Chọn nút Run this flow ở góc trái trên cửa sổ để chạy mô hình thuật toán. Nếu chạy thành công, ở mục Status, các bước chạy đều đạt được trạng thái Finished.

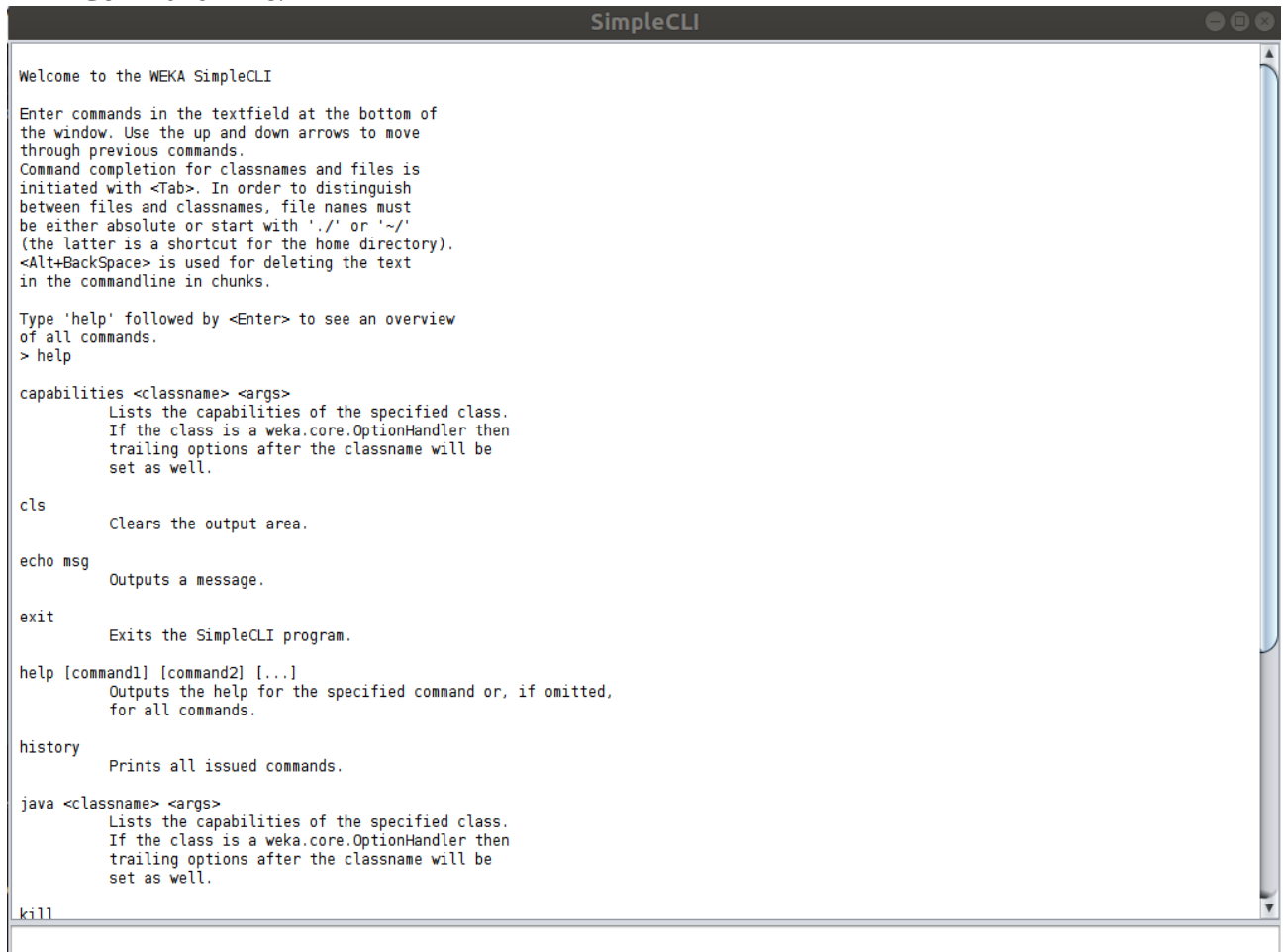


- Chuột phải ở TextViewer chọn Show result để xem kết quả mạng đã train:



Simple CLI

Đây là giao diện cho phép sử dụng các công cụ của Weka chỉ bằng các dòng lệnh Command Line.



```
SimpleCLI

Welcome to the WEKA SimpleCLI

Enter commands in the textfield at the bottom of
the window. Use the up and down arrows to move
through previous commands.
Command completion for classnames and files is
initiated with <Tab>. In order to distinguish
between files and classnames, file names must
be either absolute or start with './' or '~/ '
(the latter is a shortcut for the home directory).
<Alt+BackSpace> is used for deleting the text
in the commandline in chunks.

Type 'help' followed by <Enter> to see an overview
of all commands.
> help

capabilities <classname> <args>
    Lists the capabilities of the specified class.
    If the class is a weka.core.OptionHandler then
    trailing options after the classname will be
    set as well.

cls
    Clears the output area.

echo msg
    Outputs a message.

exit
    Exits the SimpleCLI program.

help [command1] [command2] [...]
    Outputs the help for the specified command or, if omitted,
    for all commands.

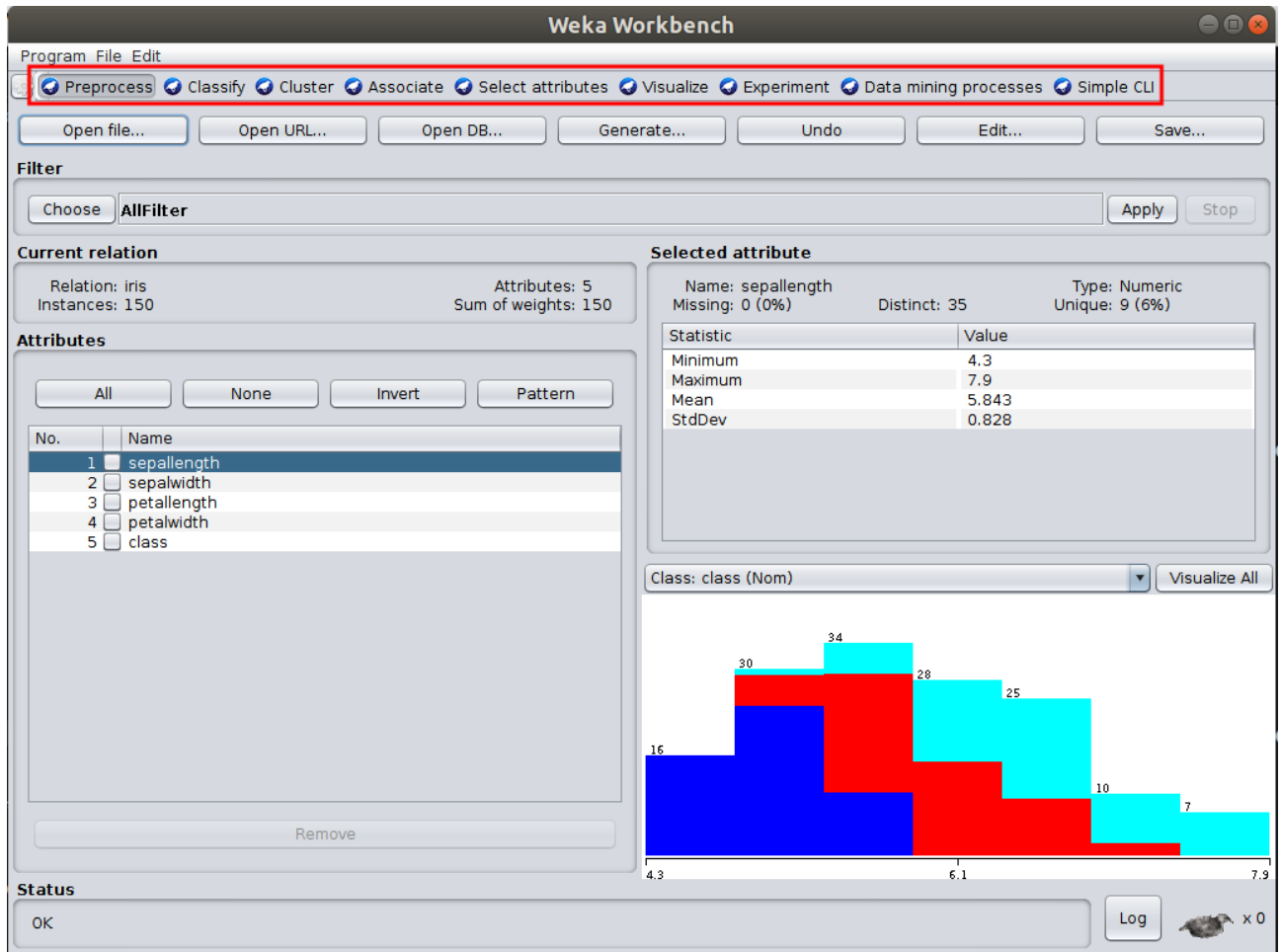
history
    Prints all issued commands.

java <classname> <args>
    Lists the capabilities of the specified class.
    If the class is a weka.core.OptionHandler then
    trailing options after the classname will be
    set as well.

kill
```

Workbench

Giao diện Workbench chỉ đơn giản là tổng hợp của tất cả các công cụ khác trong Explorer, Experimenter, Knowledge và Simple CLI.



III. Yêu cầu 2: Sử dụng weka để chạy thuật toán ID3

Mô tả dữ liệu:

- Số mẫu: 101
- Số thuộc tính: Có 18 thuộc tính như sau
 - @ATTRIBUTE animal
 {aardvark,antelope,bass,bear,boar,buffalo,calf,carp,catfish,cavy,cheetah,chicken,chub,clam,crab,crayfish,crow,deer,dogfish,dolphin,dove,duck,elephant,flamingo,flea,frog,fruitbat,giraffe,girl,gnat,goat,gorilla,gull,haddock,hamster,hare,hawk,herring,honeybee,housefly,kiwi,ladybird,lark,leopard,lion,lobster,lynx,mink,mole,mongoose,moth,newt,octopus,opossum,oryx,ostrich,parakeet,penguin,pheasant,pike,piranha,pitviper,platypus,polecat,pony,porpoise,puma,pussycat,raccoon,reindeer,rhea,scorpion,seahorse,seal,sealion,seasnake,seawasp,skimmer,skua,slowworm,slug,sole,sparrow,squirrel,starfish,stingray,swan,termite,toad,tortoise,tuatara,tuna,vampire,vole,vulture,wallaby,wasps,wolf,worm,wren}
 - @ATTRIBUTE hair {0, 1}
 - @ATTRIBUTE feathers {0, 1}
 - @ATTRIBUTE eggs {0, 1}
 - @ATTRIBUTE milk {0, 1}
 - @ATTRIBUTE airborne {0, 1}
 - @ATTRIBUTE aquatic {0, 1}
 - @ATTRIBUTE predator {0, 1}
 - @ATTRIBUTE toothed {0, 1}
 - @ATTRIBUTE backbone {0, 1}

- @ATTRIBUTE breathes {0, 1}
- @ATTRIBUTE venomous {0, 1}
- @ATTRIBUTE fins {0, 1}
- @ATTRIBUTE legs INTEGER [0,9]
- @ATTRIBUTE tail {0, 1}
- @ATTRIBUTE domestic {0, 1}
- @ATTRIBUTE catsize {0, 1}
- @ATTRIBUTE type INTEGER [1, 7]
- Chúng ta có danh sách 7 lớp động vật như sau:
 - #1: aardvark, antelope, bear, boar, buffalo, calf, cavy, cheetah, deer, dolphin, elephant, fruitbat, giraffe, girl, goat, gorilla, hamster, hare, leopard, lion, lynx, mink, mole, mongoose, opossum, oryx, platypus, polecat, pony, porpoise, puma, pussycat, raccoon, reindeer, seal, sealion, squirrel, vampire, vole, wallaby, wolf
 - #2: chicken, crow, dove, duck, flamingo, gull, hawk, kiwi, lark, ostrich, parakeet, penguin, pheasant, rhea, skimmer, skua, sparrow, swan, vulture, wren
 - #3: pitviper, seasnake, slowworm, tortoise, tuatara
 - #4: bass, carp, catfish, chub, dogfish, haddock, herring, pike, piranha, seahorse, sole, stingray, tuna
 - #5: frog, frog, newt, toad
 - #6: flea, gnat, honeybee, housefly, ladybird, moth, termite, wasp
 - #7: clam, crab, crayfish, lobster, octopus, scorpion, seawasp, slug, starfish, worm

Đặt tên lại cho các phân lớp, ta có bảng phân bố như sau:

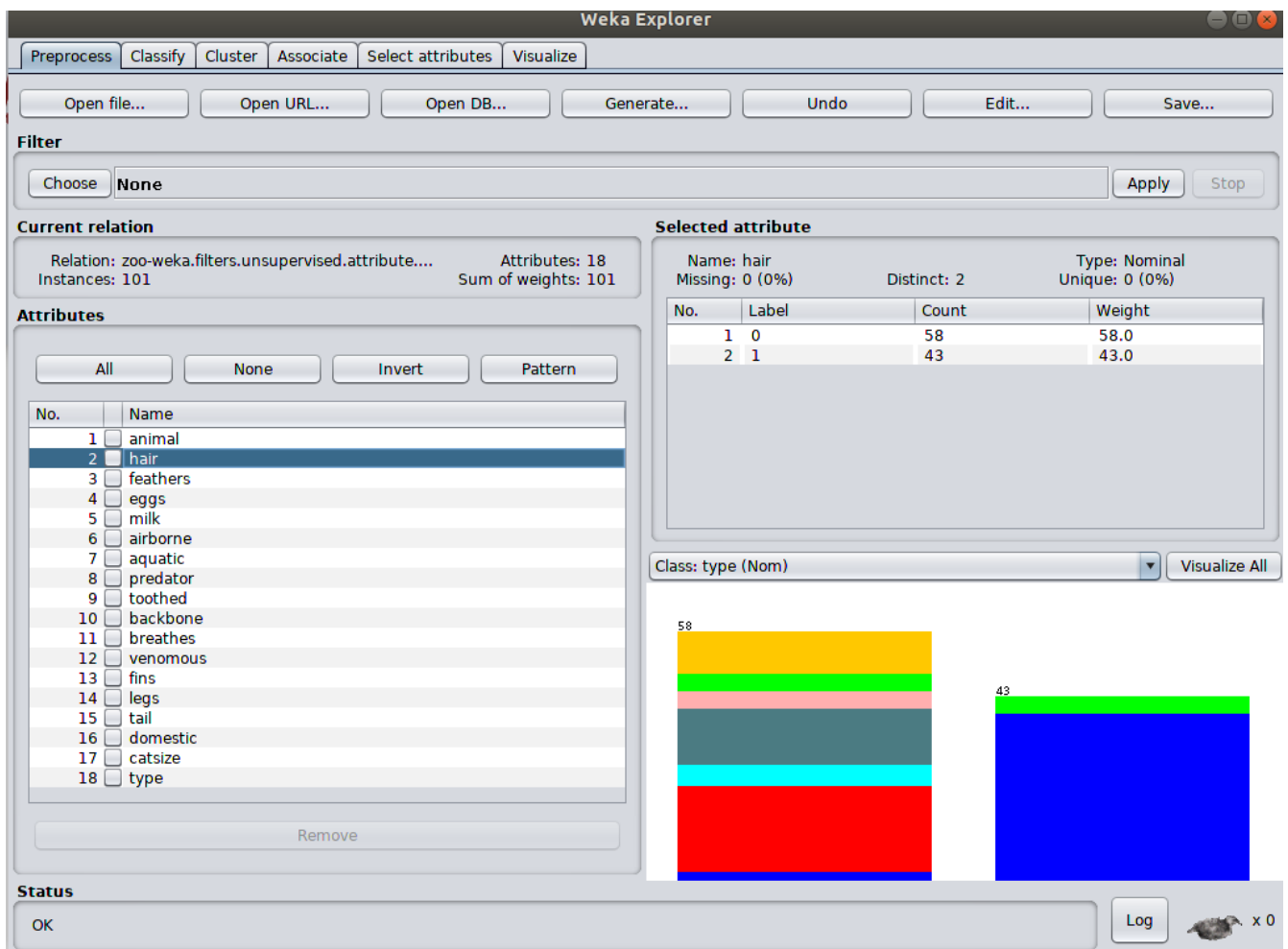
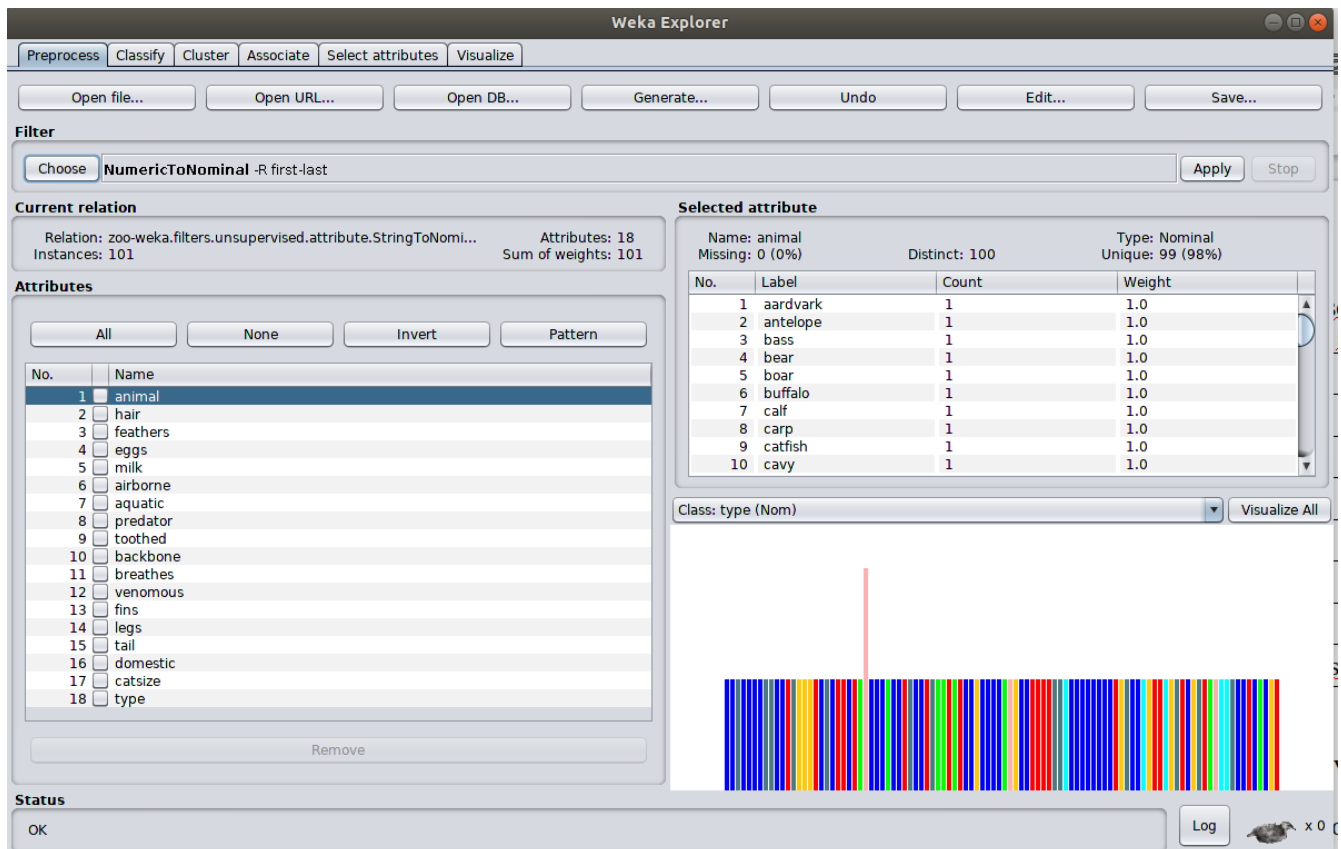
| Số thứ tự lớp | Tên lớp | Số lượng mẫu |
|---------------|--|--------------|
| 1 | mamal (thú) | 41 |
| 2 | bird (chim) | 20 |
| 3 | Reptile (bò sát) | 5 |
| 4 | fish (cá) | 13 |
| 5 | amphibian (lưỡng cư) | (4) |
| 6 | insect (côn trùng) | 8 |
| 7 | Invertebrate (động vật không xương sống) | 10 |

Do vậy thuộc tính type được sửa lại như sau:

type {mamal, bird, reptile, fish, amphibian, insect, invertebrate}

Chạy thuật toán ID3 trên weka

- Đầu tiên ta cần chuyển dữ liệu về dạng định danh (Nominal).



- Sử dụng lựa chọn cách test phân chia tập dữ liệu: 70% dữ liệu trainning, còn lại 30% là test.
- Lần lượt chạy với các điều kiện phân lớp:
 - Phân lớp với thuộc tính animal, ta thu được kết quả:

=== Classifier model for training split (71 instances) ===

Id3

type = mamal

| predator = 0

| | domestic = 0

| | | legs = 0: null

| | | legs = 2

| | | | airborne = 0: gorilla

| | | | airborne = 1: fruitbat

| | | legs = 4

| | | | catsize = 0: vole

| | | | catsize = 1: antelope

| | | legs = 5: null

| | | legs = 6: null

| | | legs = 8: null

| domestic = 1

| | catsize = 0

| | | tail = 0: cavy

| | | tail = 1: hamster

| | | catsize = 1: calf

| predator = 1

| | aquatic = 0

| | | tail = 0: aardvark

| | | tail = 1

| | | | domestic = 0: boar

| | | | domestic = 1: pussycat

| | aquatic = 1

| | | legs = 0: porpoise

| | | legs = 2: sealion

| | | legs = 4

| | | | eggs = 0: mink

| | | | eggs = 1: platypus

| | | legs = 5: null

| | | legs = 6: null

| | | legs = 8: null

type = bird

| predator = 0

| | domestic = 0

| | | catsize = 0

| | | | aquatic = 0: pheasant

| | | | aquatic = 1: duck

| | | catsize = 1

| | | | airborne = 0: ostrich

| | | | airborne = 1

| | | | | aquatic = 0: flamingo

| | | | | aquatic = 1: swan

| | domestic = 1: chicken

| predator = 1

| | aquatic = 0

| | | airborne = 0: rhea

| | | airborne = 1: crow

| | aquatic = 1

| | | airborne = 0: penguin

| | | airborne = 1: gull

type = reptile

| legs = 0

| | eggs = 0: seasnake

| | eggs = 1: slowworm

| legs = 2: null

| legs = 4

| | predator = 0: tortoise

| | predator = 1: tuatara

| legs = 5: null

| legs = 6: null

```

| legs = 8: null
type = fish
| catsize = 0
| | predator = 0
| | | domestic = 0: haddock
| | | domestic = 1: carp
| | predator = 1: bass
| catsize = 1
| | venomous = 0: dogfish
| | venomous = 1: stingray
type = amphibian
| predator = 0: toad
| predator = 1
| | tail = 0: frog
| | tail = 1: newt
type = insect
| hair = 0: gnat
| hair = 1: moth
type = invertebrate
| legs = 0
| | aquatic = 0
| | | predator = 0: slug
| | | predator = 1: clam
| | aquatic = 1: seawasp
| legs = 2: null
| legs = 4: crab
| legs = 5: null
| legs = 6: crayfish
| legs = 8: scorpion
=== Summary ===

```

| | | |
|----------------------------------|------------|-----------|
| Correctly Classified Instances | 0 | 0 % |
| Incorrectly Classified Instances | 29 | 96.6667 % |
| Kappa statistic | 0 | |
| Mean absolute error | 0.02 | |
| Root mean squared error | 0.1316 | |
| Relative absolute error | 104.0568 % | |
| Root relative squared error | 133.8798 % | |
| UnClassified Instances | 1 | 3.3333 % |
| Total Number of Instances | 30 | |

- Phân lớp với thuộc tính type, ta thu được kết quả:

=== Classifier model for training split (71 instances) ===

Id3

```

animal = aardvark: mamal
animal = antelope: mamal
animal = bass: fish
animal = bear: mamal
animal = boar: mamal
animal = buffalo: mamal
animal = calf: mamal
animal = carp: fish
animal = catfish: fish
animal = cavy: mamal
animal = cheetah: null
animal = chicken: bird
animal = chub: null
animal = clam: invertebrate
animal = crab: invertebrate
animal = crayfish: invertebrate
animal = crow: bird
animal = deer: null
animal = dogfish: fish
animal = dolphin: null
animal = dove: bird

```

animal = duck: bird
animal = elephant: mamal
animal = flamingo: bird
animal = flea: null
animal = frog: amphibian
animal = fruitbat: mamal
animal = giraffe: mamal
animal = girl: null
animal = gnat: insect
animal = goat: mamal
animal = gorilla: mamal
animal = gull: bird
animal = haddock: fish
animal = hamster: mamal
animal = hare: null
animal = hawk: bird
animal = herring: fish
animal = honeybee: null
animal = housefly: null
animal = kiwi: null
animal = ladybird: null
animal = lark: null
animal = leopard: mamal
animal = lion: mamal
animal = lobster: invertebrate
animal = lynx: mamal
animal = mink: mamal
animal = mole: null
animal = mongoose: mamal
animal = moth: insect
animal = newt: amphibian
animal = octopus: null
animal = opossum: null
animal = oryx: mamal
animal = ostrich: bird
animal = parakeet: bird
animal = penguin: bird
animal = pheasant: bird
animal = pike: fish
animal = piranha: null
animal = pitviper: null
animal = platypus: mamal
animal = polecat: null
animal = pony: mamal
animal = porpoise: mamal
animal = puma: mamal
animal = pussycat: mamal
animal = raccoon: mamal
animal = reindeer: mamal
animal = rhea: bird
animal = scorpion: invertebrate
animal = seahorse: null
animal = seal: null
animal = sealion: mamal
animal = seasnake: reptile
animal = seawasp: invertebrate
animal = skimmer: bird
animal = skua: null
animal = slowworm: reptile
animal = slug: invertebrate
animal = sole: null
animal = sparrow: bird
animal = squirrel: null
animal = starfish: null
animal = stingray: fish


```

animal = swan: bird
animal = termite: null
animal = toad: amphibian
animal = tortoise: reptile
animal = tuatara: reptile
animal = tuna: fish
animal = vampire: null
animal = vole: mamal
animal = vulture: null
animal = wallaby: null
animal = wasp: null
animal = wolf: mamal
animal = worm: null
animal = wren: bird
=== Summary ===

```

```

Correctly Classified Instances      0      0  %
Incorrectly Classified Instances    0      0  %
Kappa statistic                    1
Mean absolute error                NaN
Root mean squared error            NaN
Relative absolute error            NaN  %
Root relative squared error        NaN  %
UnClassified Instances             30     100  %
Total Number of Instances          30

```

Nhận xét:

- Ở đây ta thấy thuộc tính animal (tên động vật) là một thuộc tính phân biệt.
- Mỗi một động vật chỉ thuộc duy nhất một lớp (type) theo như định nghĩa.
→ Thuộc tính animal là một thuộc tính gây nhiễu khi huấn luyện. Do vậy chúng ta cần bước tiền xử lý dữ liệu loại bỏ thuộc tính animal.

Sau khi loại bỏ thuộc tính animal, phân lớp trên type ta thu được:

=== Classifier model for training split (71 instances) ===

Id3

legs = 0

| fins = 0

| | toothed = 0: invertebrate

| | toothed = 1: reptile

| fins = 1

| | eggs = 0: mamal

| | eggs = 1: fish

legs = 2

| hair = 0: bird

| hair = 1: mamal

legs = 4

| hair = 0

| | aquatic = 0: reptile

| | aquatic = 1

| | | toothed = 0: invertebrate

| | | toothed = 1: amphibian

| hair = 1: mamal

legs = 5: invertebrate

legs = 6

| aquatic = 0: insect

| aquatic = 1: invertebrate

legs = 8: invertebrate

=== Summary ===

```

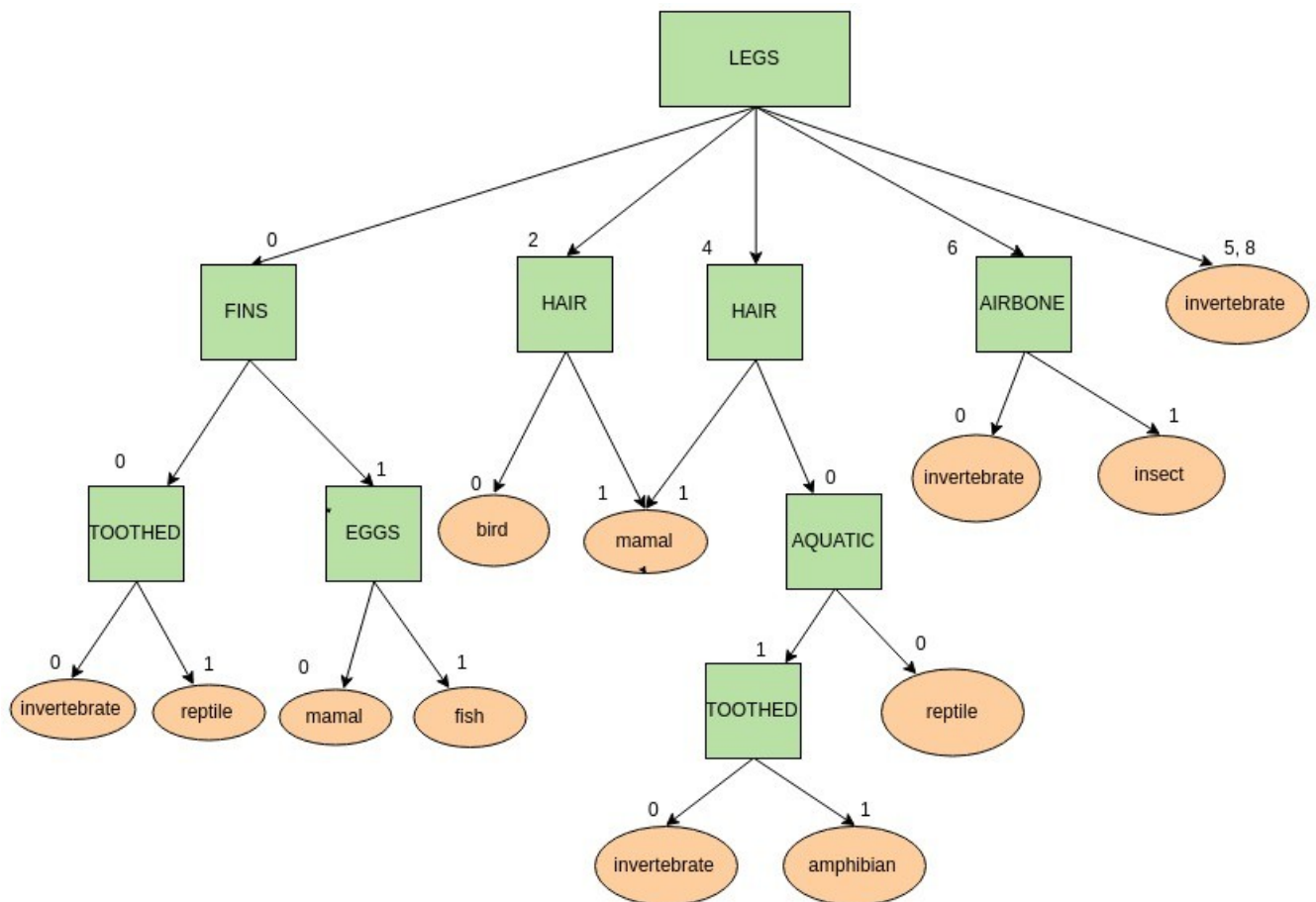
Correctly Classified Instances      27      90  %
Incorrectly Classified Instances      2      6.6667  %
Kappa statistic                    0.9082
Mean absolute error                0.0197
Root mean squared error            0.1404

```

| | | |
|-----------------------------|-----------|----------|
| Relative absolute error | 9.1954 % | |
| Root relative squared error | 42.6666 % | |
| UnClassified Instances | 1 | 3.3333 % |
| Total Number of Instances | 30 | |

Sau các thử nghiệm trên ta thấy được bước xử lý tiền dữ liệu rất quan trọng trong việc huấn luyện mô hình Máy học.

Dựa trên kết quả chạy được, ta có cây quyết định được sinh ra như sau:



Chạy test trên mô hình cây quyết định

- Tạo file dữ liệu test.arff như sau:

```

@attribute hair {0,1}
@attribute feathers {0,1}
@attribute eggs {0,1}
@attribute milk {0,1}
@attribute airborne {0,1}
@attribute aquatic {0,1}
@attribute predator {0,1}
@attribute toothed {0,1}
@attribute backbone {0,1}
@attribute breathes {0,1}
@attribute venomous {0,1}
@attribute fins {0,1}
@attribute legs {0,2,4}
@attribute tail {0,1}
@attribute domestic {0,1}
@attribute catsize {0,1}
@attribute type {mamal,bird,reptile,fish,amphibian,insect,invertebrate}

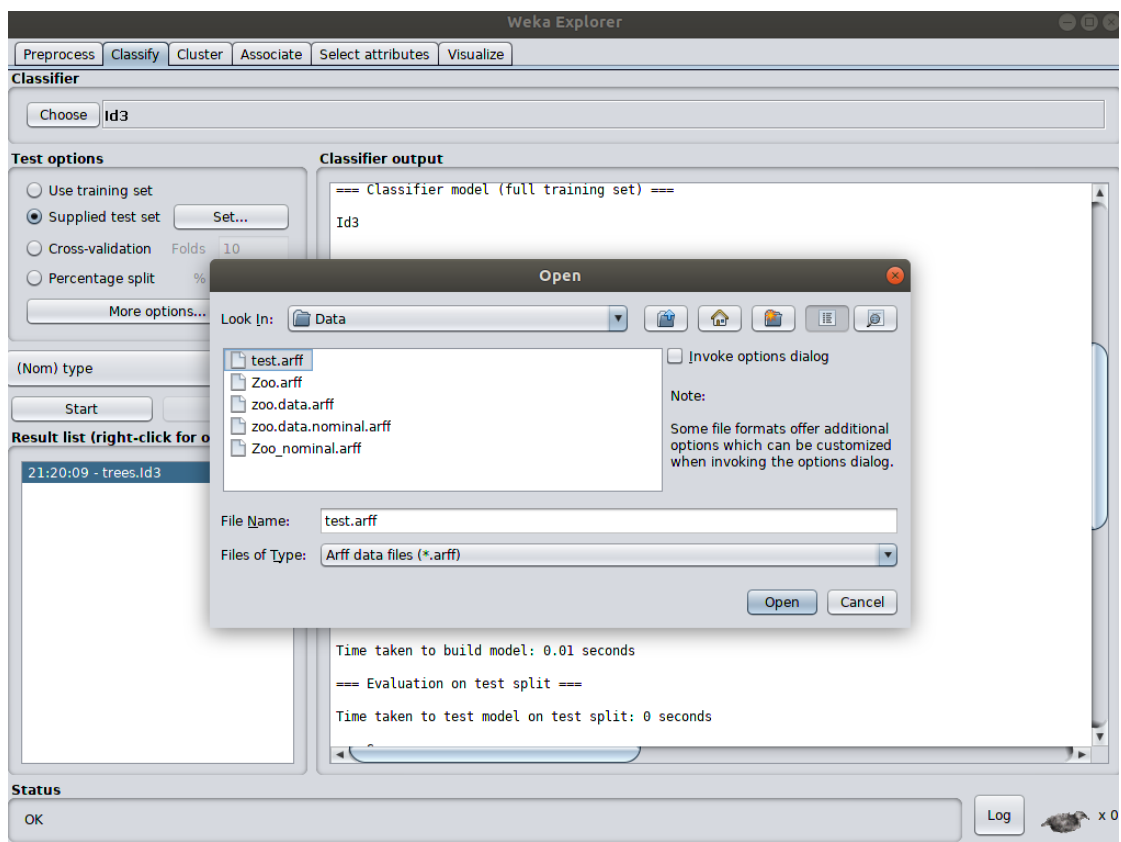
```

```

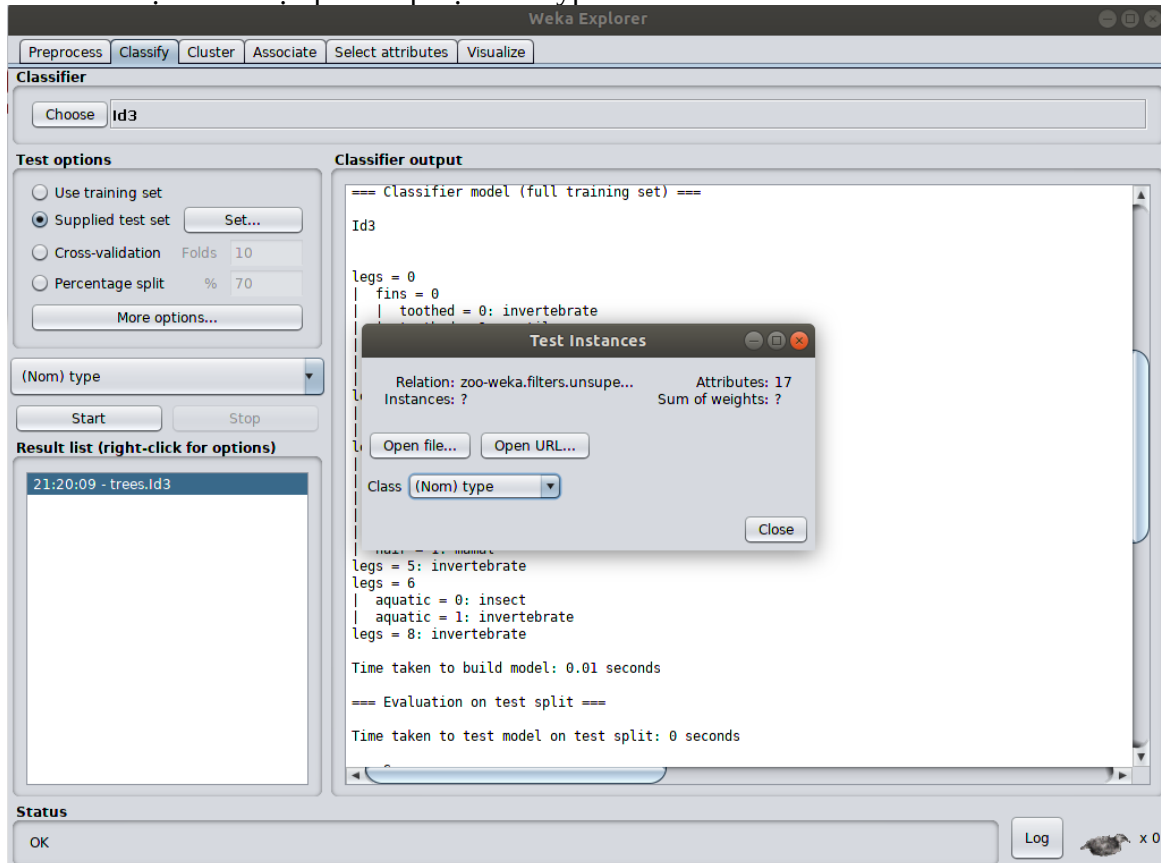
@data
1,0,0,1,0,0,0,1,1,1,0,0,4,1,0,1,?
0,1,1,0,1,0,0,0,1,1,0,0,2,1,1,0,?
0,0,1,0,0,0,1,1,1,1,1,0,0,1,0,0,?
0,0,1,0,0,1,1,1,1,0,0,1,0,1,0,0,?
0,0,1,0,0,1,1,1,1,0,0,4,1,0,0,?

```

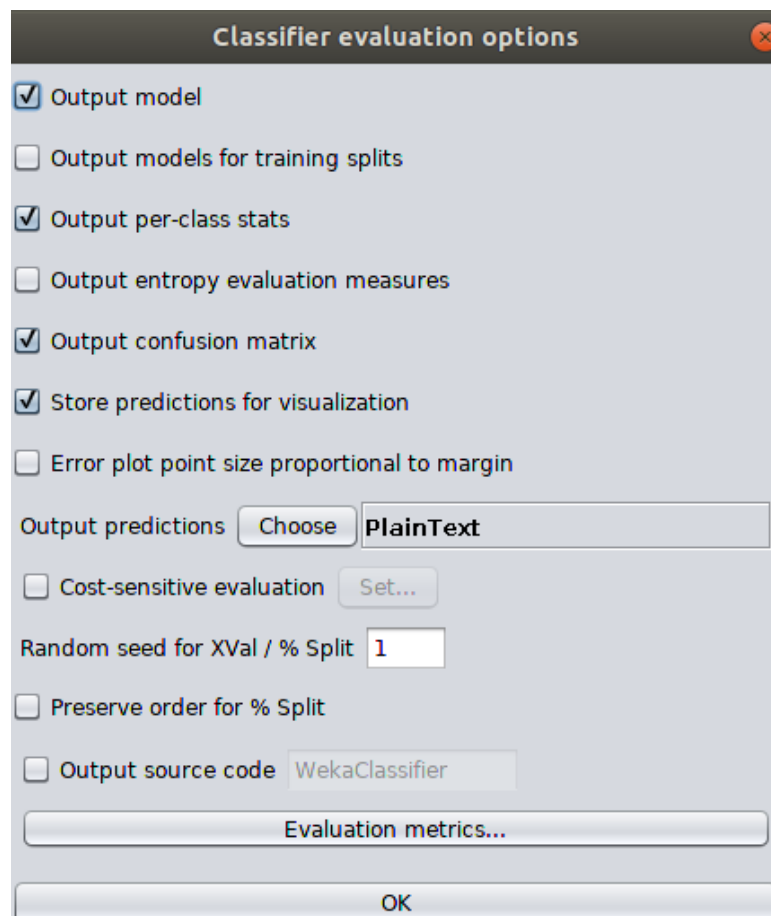
- Lưu ý vì thuộc tính animal không quan trọng nên đã được lược bỏ.
- Sau đó chọn chế độ Supplied test set ở mục chọn Test, tiến hành thêm file test.arff đã tạo bên trên vào.



- Chọn điều kiện phân lớp dựa trên type.



- Chọn Output Predictions là PlainText để xuất kết quả test ra màn hình.



- Nhấn start, ta thu được kết quả như sau:

```

=== Predictions on test set ===

inst#    actual    predicted error prediction
  1      1:?      1:mamal    1
  2      1:?      2:bird     1
  3      1:?      3:reptile  1
  4      1:?      4:fish     1
  5      1:?      5:amphibian 1

```

- Ta có kết quả dự đoán của 5 mẫu đã cho:
 - 1. NameIsSecret,1,0,0,1,0,0,0,1,1,1,0,0,4,1,0,1, ? → mamal
 - 2. NameIsSecret,0,1,1,0,1,0,0,0,1,1,0,0,2,1,1,0, ? → bird
 - 3. NameIsSecret,0,0,1,0,0,0,1,1,1,1,0,0,1,0,0, ? → reptile
 - 4. NameIsSecret,0,0,1,0,0,1,1,1,1,0,0,1,0,1,0,0, ? → fish
 - 5. NameIsSecret,0,0,1,0,0,1,1,1,1,1,0,0,4,1,0,0, ? → amphibian
- Kiểm tra lại với cây quyết định bên trên, kết quả hoàn toàn trùng khớp.

IV. Yêu cầu 3

Cài đặt thuật toán Naive Bayes Python:

Ý tưởng của thuật toán xuất phát từ công thức tính xác suất có điều kiện Bayes.

Dựa trên đó chúng ta chỉ cần tính xác suất từng class cho vector thuộc tính mà ta có. Sau đó ta sẽ đưa ra dự đoán là class có xác suất cao nhất:

$$c = \arg \max_{c \in \{1, \dots, C\}} p(c) \prod_{i=1}^d p(x_i | c)$$

Khi d lớn và các xác suất nhỏ, biểu thức ở vế phải của công thức trên sẽ là một số rất nhỏ, khi tính toán có thể gặp sai số. Để giải quyết việc này, thường được viết lại dạng tương đương bằng cách lấy log của vế phải.

$$c = \arg \max_{c \in \{1, \dots, C\}} = \log(p(c)) + \sum_{i=1}^d \log(p(x_i | c))$$

Dựa vào tập dữ liệu mà ta có thể chọn cách tính xác suất bằng các phân phối khác nhau. Tuy nhiên, phân phối mà chúng em chọn cho bài toán này Multinomial Naive Bayes, vì tập dữ liệu Zoo được cho là tập dữ liệu rời rạc.

Thuật toán chính được thể hiện qua class Multinomial được viết như sau:

```

class MultinomialNB(object):
    #alpha for Laplace smoothing
    def __init__(self, alpha = 1.0):
        self.alpha = alpha
    def fit(self, X, y):
        self.classes = np.unique(y)
        seperated = [[x for x, t in zip(X, y) if t == c] for c in self.classes]
        count_sample = X.shape[0]
        self.class_prior_ = [np.log(len(i) / count_sample) for i in seperated]
        # self.class_prior_ = (len(i) / count_sample) for i in seperated]
        count = np.array([np.array(i).sum(axis=0) for i in seperated]) + self.alpha
        self.feature_prob = np.log(count / (count.sum(axis = 1)[np.newaxis].T))
        # self.feature_prob = (count / (count.sum(axis = 1)[np.newaxis].T))
        return self

    def predict_prob(self, X):
        return [(self.feature_prob * x).sum(axis = 1) + self.class_prior_ for x in X]

    def predict(self, X):
        pred = np.argmax(self.predict_prob(X), axis = 1)
        prob = np.max(self.predict_prob(X), axis = 1)
        return [self.classes[p] for p in pred]

```

Hàm fit(X, y) được dùng để train một tập dữ liệu với X là danh sách các thuộc tính được chứa biểu diễn dưới dạng vector, y là danh sách các class phân lớp.

Đầu tiên ta sẽ tách danh sách X theo các class cụ thể và lưu nó vào seperated.

class_prior được tính với công thức:

$$\hat{P}(c) = \frac{N_c}{N}$$

Như đã đề cập, để tránh sai số nên ta lấy log của công thức trên.

Chạy các thuật toán khác

Thiết lập dữ liệu:

- Mở file Zoo.arff:

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Generate... | Undo | Edit... | Save...

Filter

Choose Apply Stop

Current relation

Relation: zoo
Instances: 101

Attributes: 18
Sum of weights: 101

Attributes

All None Invert Pattern

| No. | Name |
|-----|--|
| 1 | <input checked="" type="checkbox"/> animal |
| 2 | <input type="checkbox"/> hair |
| 3 | <input type="checkbox"/> feathers |
| 4 | <input type="checkbox"/> eggs |
| 5 | <input type="checkbox"/> milk |
| 6 | <input type="checkbox"/> airborne |
| 7 | <input type="checkbox"/> aquatic |
| 8 | <input type="checkbox"/> predator |
| 9 | <input type="checkbox"/> toothed |
| 10 | <input type="checkbox"/> backbone |
| 11 | <input type="checkbox"/> breathes |
| 12 | <input type="checkbox"/> venomous |
| 13 | <input type="checkbox"/> fins |
| 14 | <input type="checkbox"/> legs |
| 15 | <input type="checkbox"/> tail |
| 16 | <input type="checkbox"/> domestic |
| 17 | <input type="checkbox"/> catsize |
| 18 | <input type="checkbox"/> type |

Remove

Selected attribute

Name: animal
Missing: 0 (0%)
Distinct: 100
Type: Nominal
Unique: 99 (98%)

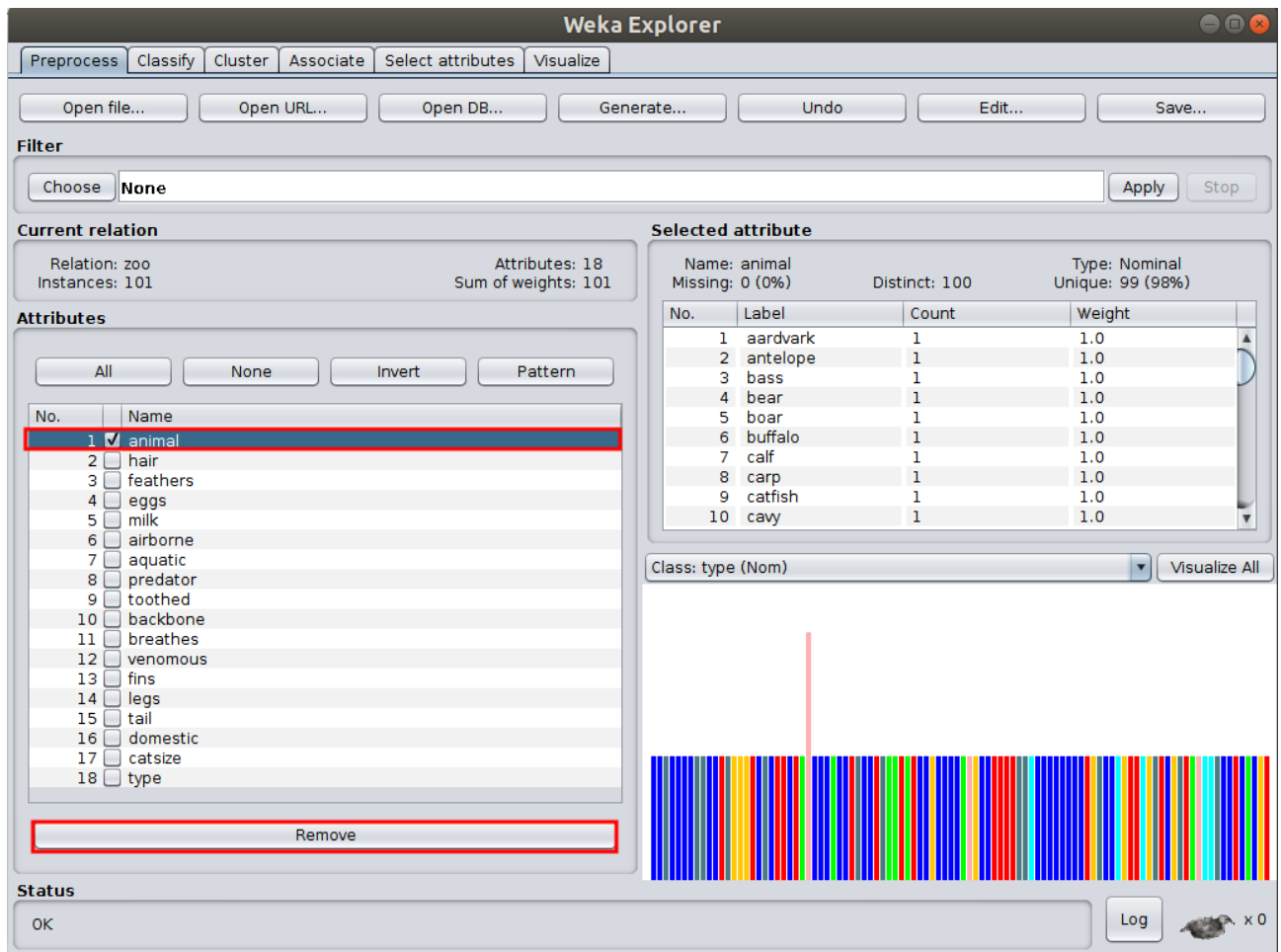
| No. | Label | Count | Weight |
|-----|----------|-------|--------|
| 1 | aardvark | 1 | 1.0 |
| 2 | antelope | 1 | 1.0 |
| 3 | bass | 1 | 1.0 |
| 4 | bear | 1 | 1.0 |
| 5 | boar | 1 | 1.0 |
| 6 | buffalo | 1 | 1.0 |
| 7 | calf | 1 | 1.0 |
| 8 | carp | 1 | 1.0 |
| 9 | catfish | 1 | 1.0 |
| 10 | cavy | 1 | 1.0 |

Class: type (Nom) Visualize All

Status

OK Log x 0

- Loại bỏ thuộc tính animal vì nó không có ý nghĩa phân loại:

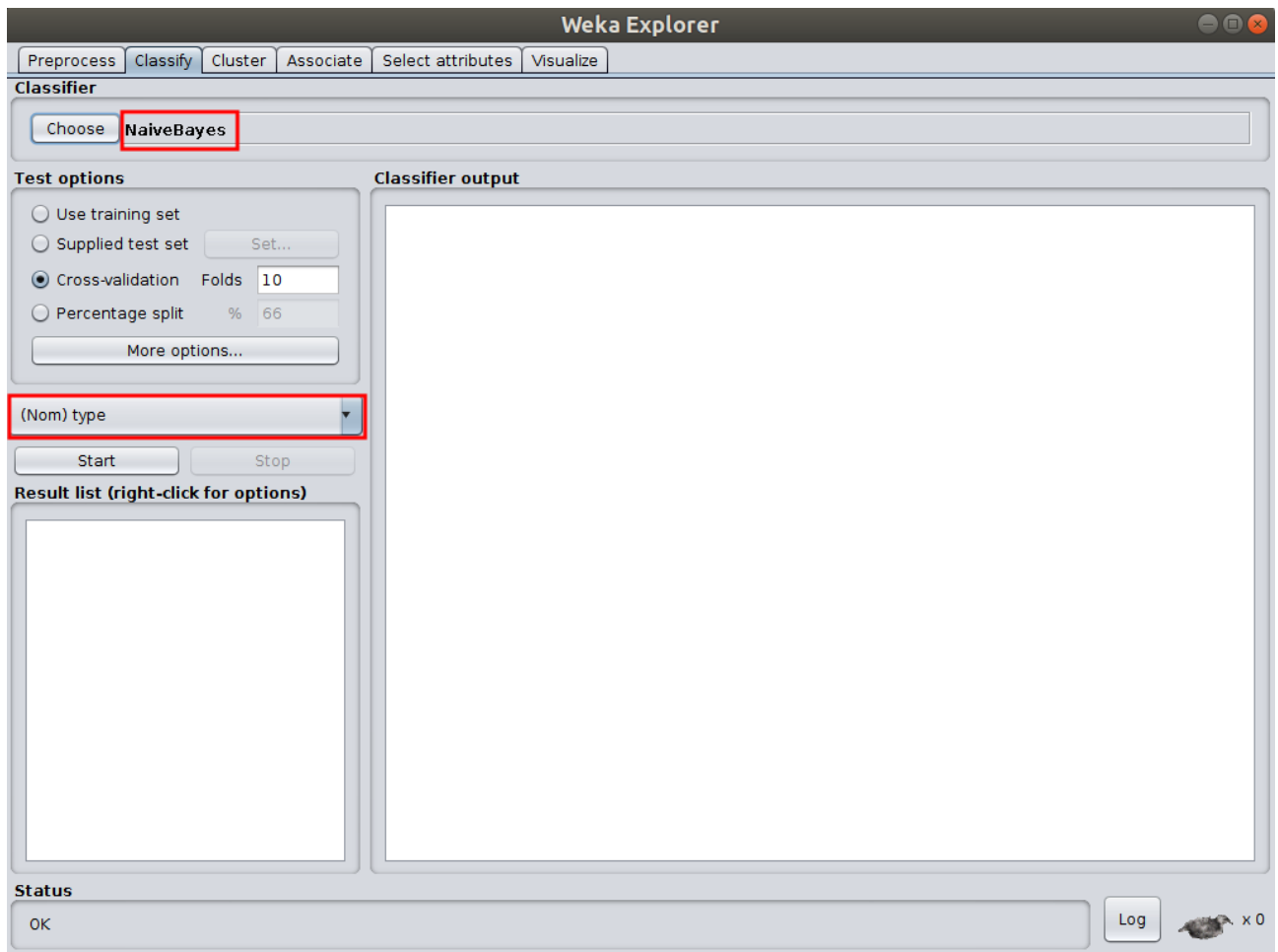


1. Thuật toán Naive Bayes:

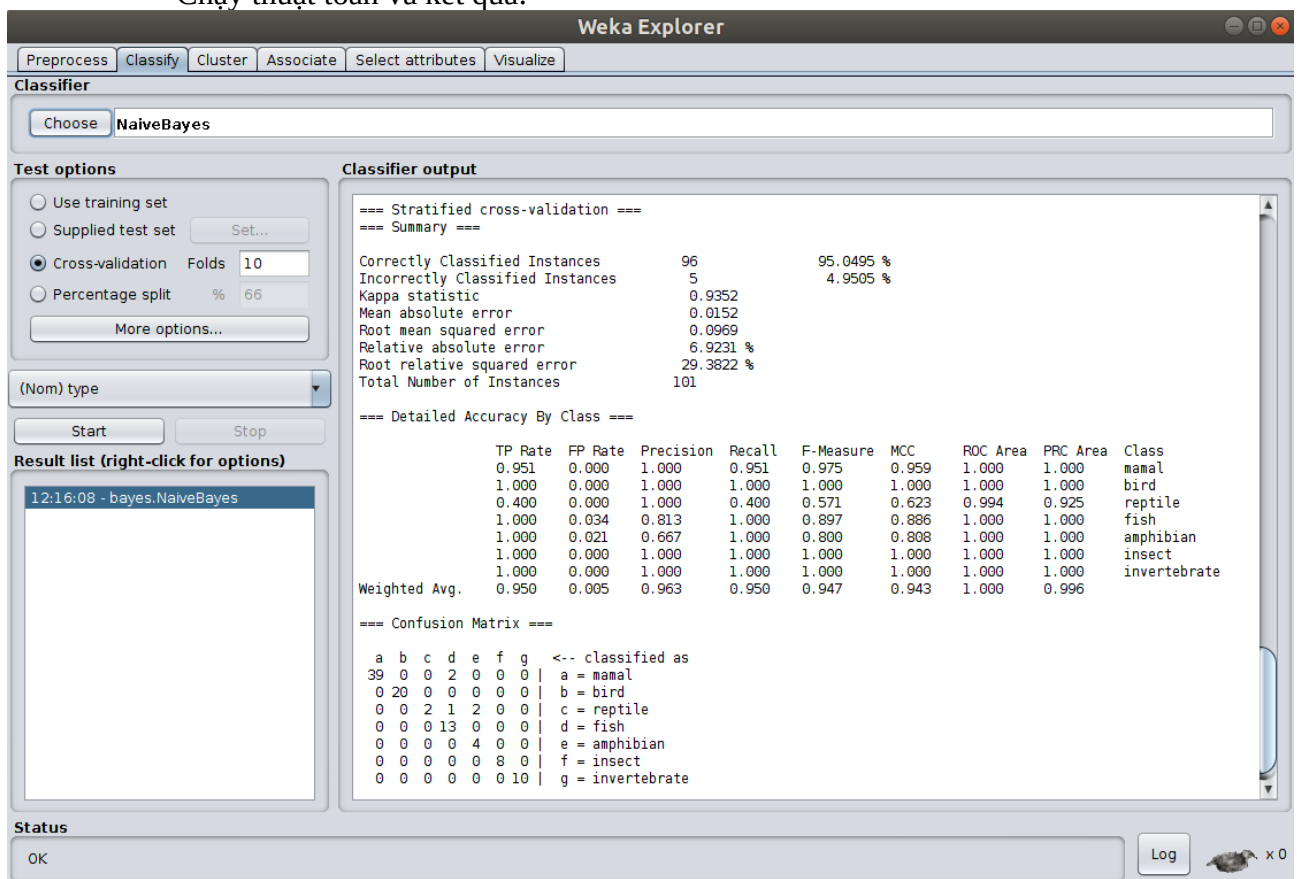
Thiết lập thuật toán: Các bước thiết lập cơ bản giống như NaiveBayes.

Chạy thuật toán:

- Chọn tab ứng dụng Classify. Trong khung Classifier chọn thuật toán NaiveBayes (trong mục bayes). Đồng thời đảm bảo tên thuộc tính đang dùng để phân loại là type:



- Chạy thuật toán và kết quả:



Giải thích kết quả:

- Thuật toán phân loại 101 loài động vật, phân loại chính xác 96 loài, sai 5 loài, độ chính xác 95.0495%.

- Dựa vào confusion matrix ta thấy 5 loài bị phân loại sai là:
 - 2 mamal (a) bị phân loại thành fish (d).
 - 1 reptile (c) bị phân loại thành fish (d).
 - 2 reptile (c) bị phân loại thành amphibian (e).

2. Thuật toán MultilayerPerceptron

Thiết lập thuật toán: Các bước thiết lập cơ bản giống như NaiveBayes.

Chạy thuật toán:

The screenshot shows the Weka Explorer interface with the MultilayerPerceptron classifier selected. The 'Test options' section shows 'Cross-validation' with 'Folds' set to 10. The 'Classifier output' section displays the following results:

| Metric | Value | Percentage |
|----------------------------------|-----------|------------|
| Correctly Classified Instances | 96 | 95.0495 % |
| Incorrectly Classified Instances | 5 | 4.9505 % |
| Kappa statistic | 0.9346 | |
| Mean absolute error | 0.02 | |
| Root mean squared error | 0.0991 | |
| Relative absolute error | 9.1345 % | |
| Root relative squared error | 30.0495 % | |
| Total Number of Instances | 101 | |

Below the summary statistics, a 'Detailed Accuracy By Class' table is shown:

| | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC | ROC Area | PRC Area |
|---------------|---------|---------|-----------|--------|-----------|-------|----------|----------|
| a | 1.000 | 0.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| b | 1.000 | 0.012 | 0.952 | 1.000 | 0.976 | 0.970 | 0.999 | 0.998 |
| c | 0.400 | 0.010 | 0.667 | 0.400 | 0.500 | 0.498 | 0.871 | 0.699 |
| d | 1.000 | 0.011 | 0.929 | 1.000 | 0.963 | 0.958 | 1.000 | 1.000 |
| e | 0.750 | 0.010 | 0.750 | 0.750 | 0.750 | 0.740 | 0.997 | 0.950 |
| f | 1.000 | 0.011 | 0.889 | 1.000 | 0.941 | 0.938 | 1.000 | 1.000 |
| g | 0.900 | 0.000 | 1.000 | 0.900 | 0.947 | 0.944 | 0.969 | 0.926 |
| Weighted Avg. | 0.950 | 0.006 | 0.946 | 0.950 | 0.946 | 0.943 | 0.990 | 0.975 |

At the bottom, the 'Confusion Matrix' is displayed:

```

a b c d e f g <-- classified as
41 0 0 0 0 0 0 | a = mamal
0 20 0 0 0 0 0 | b = bird
0 1 2 1 1 0 0 | c = reptile
0 0 0 13 0 0 0 | d = fish
0 0 1 0 3 0 0 | e = amphibian
0 0 0 0 8 0 0 | f = insect
0 0 0 0 0 1 9 | g = invertebrate
  
```

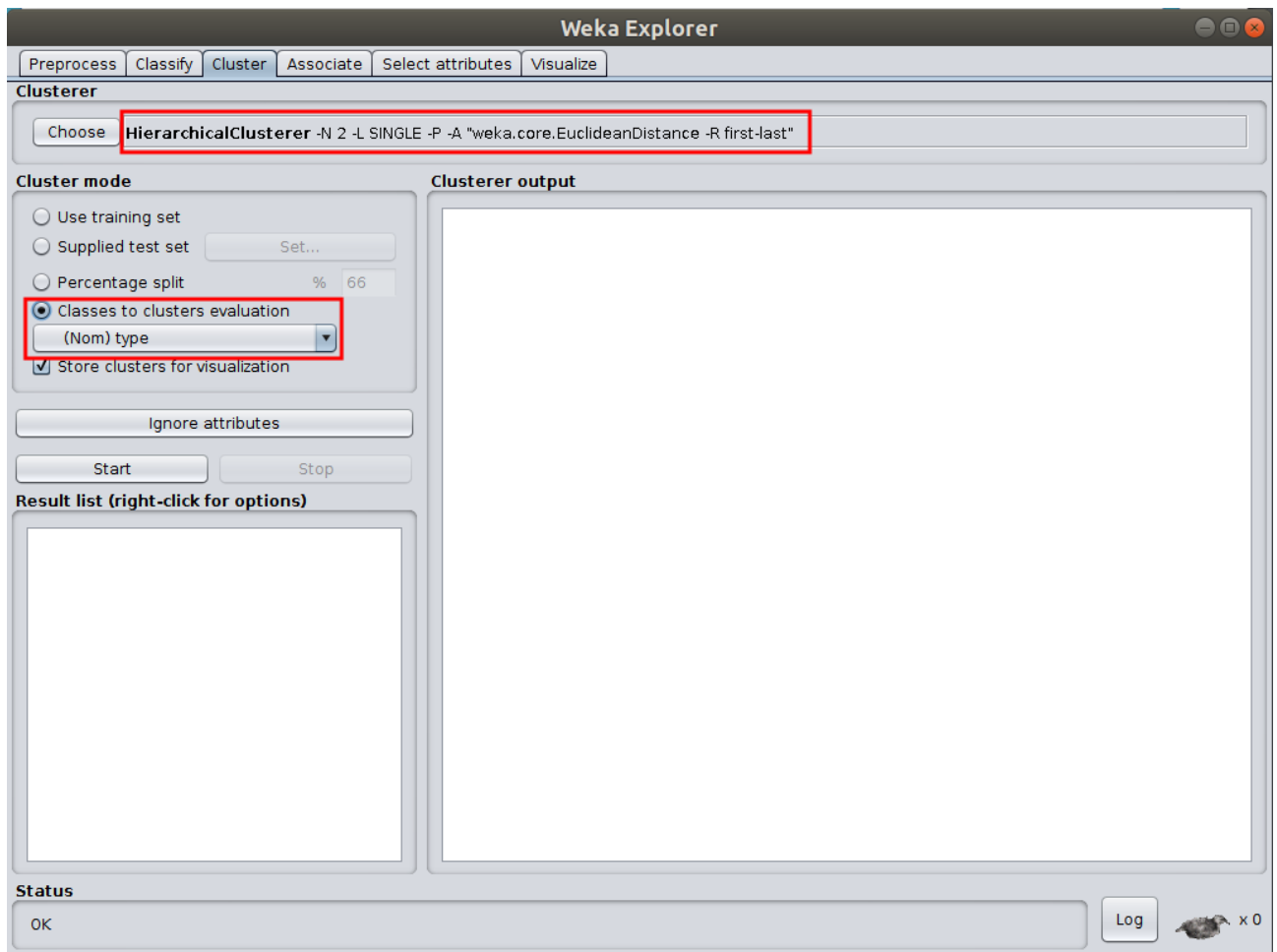
Giải thích kết quả:

- Kết quả phân loại 101 loài động vật, phân loại chính xác 96 loài, sai 5 loài, độ chính xác 95.0495%.
- Dựa vào confusion matrix ta thấy 5 loài bị phân loại sai là:
 - 1 reptile (c) bị phân loại thành bird (b).
 - 1 reptile (c) bị phân loại thành fish (d).
 - 1 reptile (c) bị phân loại thành amphibian (e).
 - 1 amphibian (e) bị phân loại thành reptile (c).
 - 1 invertebrate (g) bị phân loại thành insect (f).

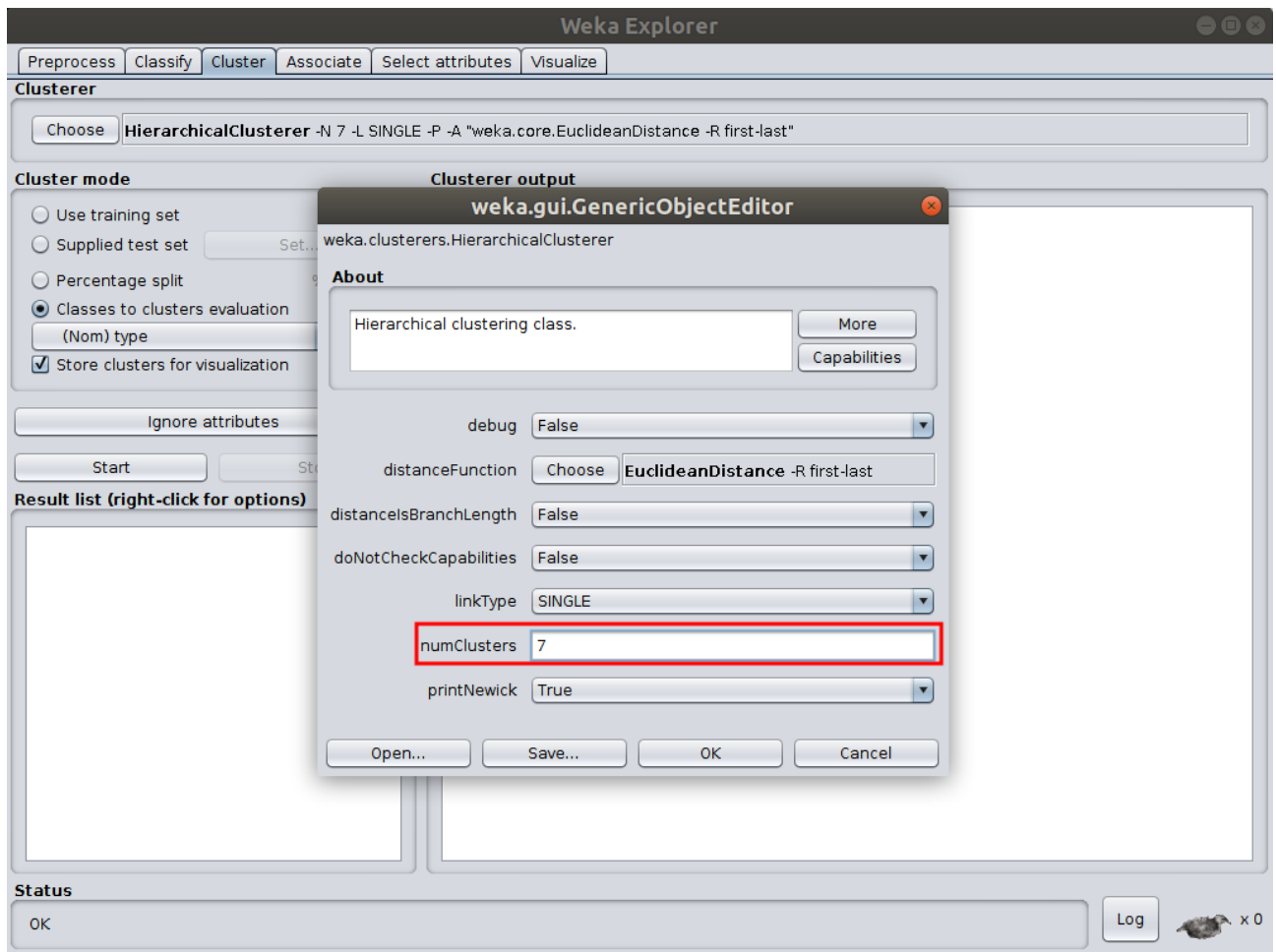
3. Thuật toán KMeans:

Thiết lập thuật toán:

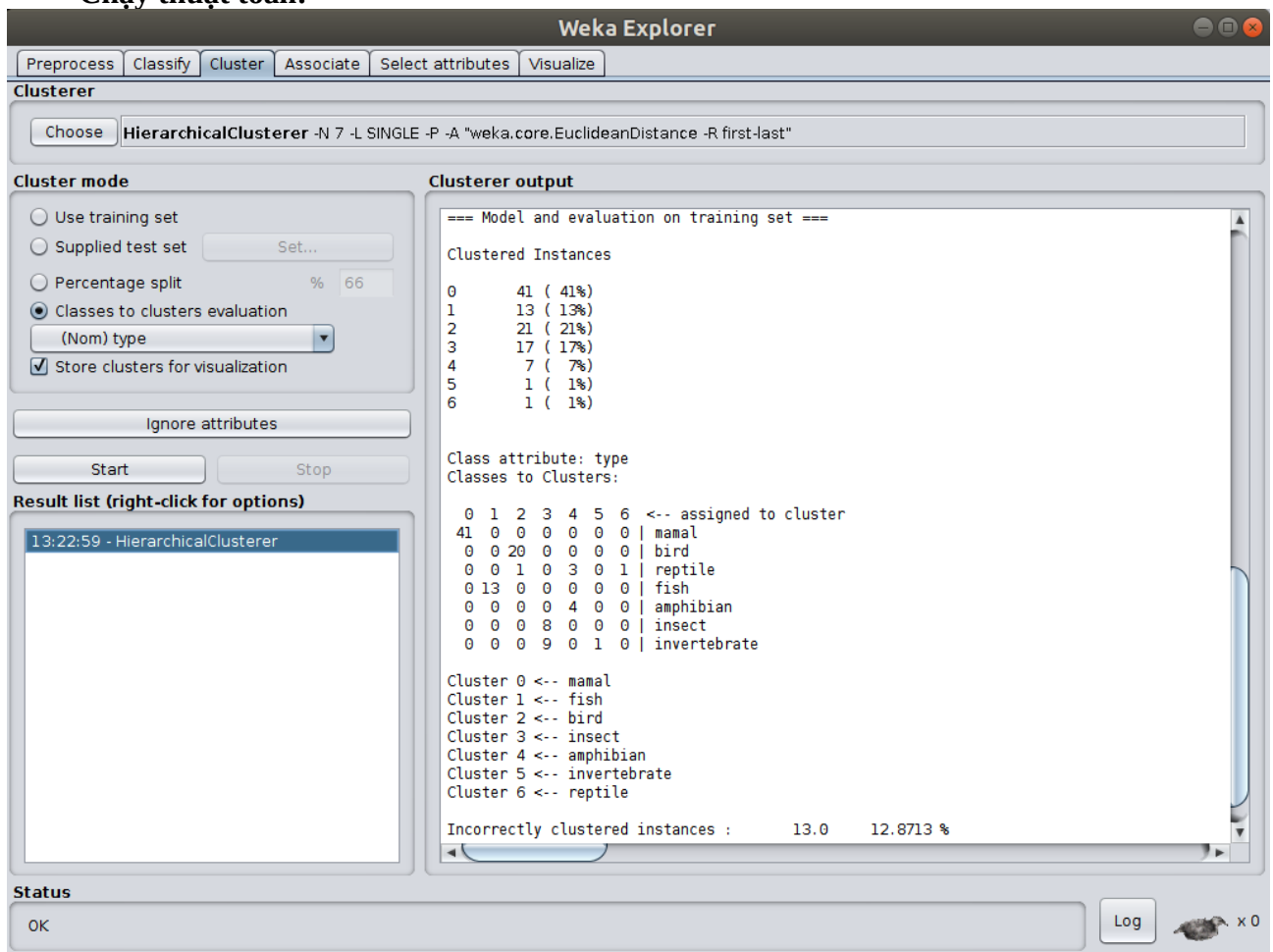
- Chọn tab ứng dụng Cluster. Trong khung Clusterer chọn thuật toán HierarchicalClusterer. Đồng thời đảm bảo tên thuộc tính dùng làm class là type.



- Tùy chỉnh tham số numClusters của thuật toán HierarchicalClusterer thành 7 vì chúng ta đang có 7 loại động vật cần phân loại.



Chạy thuật toán:



Giải thích kết quả:

Có 13 loài bị phân loại sai, chiếm tỉ lệ 12.8713%, cụ thể:

- 1 reptile bị phân loại thành bird (class 2).
- 3 reptile bị phân loại thành amphibian (class 4).
- 9 invertebrate bị phân loại thành reptile (class 3).