



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN - TP HCM
KHOA CÔNG NGHỆ THÔNG TIN
MÔN CƠ SỞ TRÍ TUỆ NHÂN TẠO
LỚP CỬ NHÂN TÀI NĂNG 2016

BÁO CÁO ĐỒ ÁN

Machine Learning

NHÓM SINH VIÊN THỰC HIỆN

ĐOÀN QUANG TUẤN 1612780

LÊ HOÀNG SANG 1612554

I. Tìm hiểu công cụ Weka.

1. Giới thiệu weka

Weka (viết tắt của Waikato Environment for Knowledge Analysis) là một bộ phần mềm học máy được Đại học Waikato, New Zealand phát triển bằng Java. Weka là phần mềm tự do phát hành theo giấy phép công cộng GNU.

Mục đích nhằm xây dựng một công cụ hiện đại để phát triển các kỹ thuật trong máy học và áp dụng chúng vào bài toán khai phá dữ liệu trong thực tế.

WEKA được xây dựng bằng ngôn ngữ Java, cấu trúc gồm hơn 600 lớp, tổ chức thành 10 packages.

Các chức năng chính của phần mềm :

- Khảo sát dữ liệu : tiền xử lý dữ liệu, phân lớp, gom nhóm dữ liệu, và khai thác luật kết hợp .
- Thực nghiệm mô hình: cung cấp phương tiện để kiểm chứng, đánh giá các mô hình học .
- Biểu diễn trực quan dữ liệu bằng nhiều dạng đồ thị khác nhau.

2. Các chức năng và cách sử dụng căn bản.

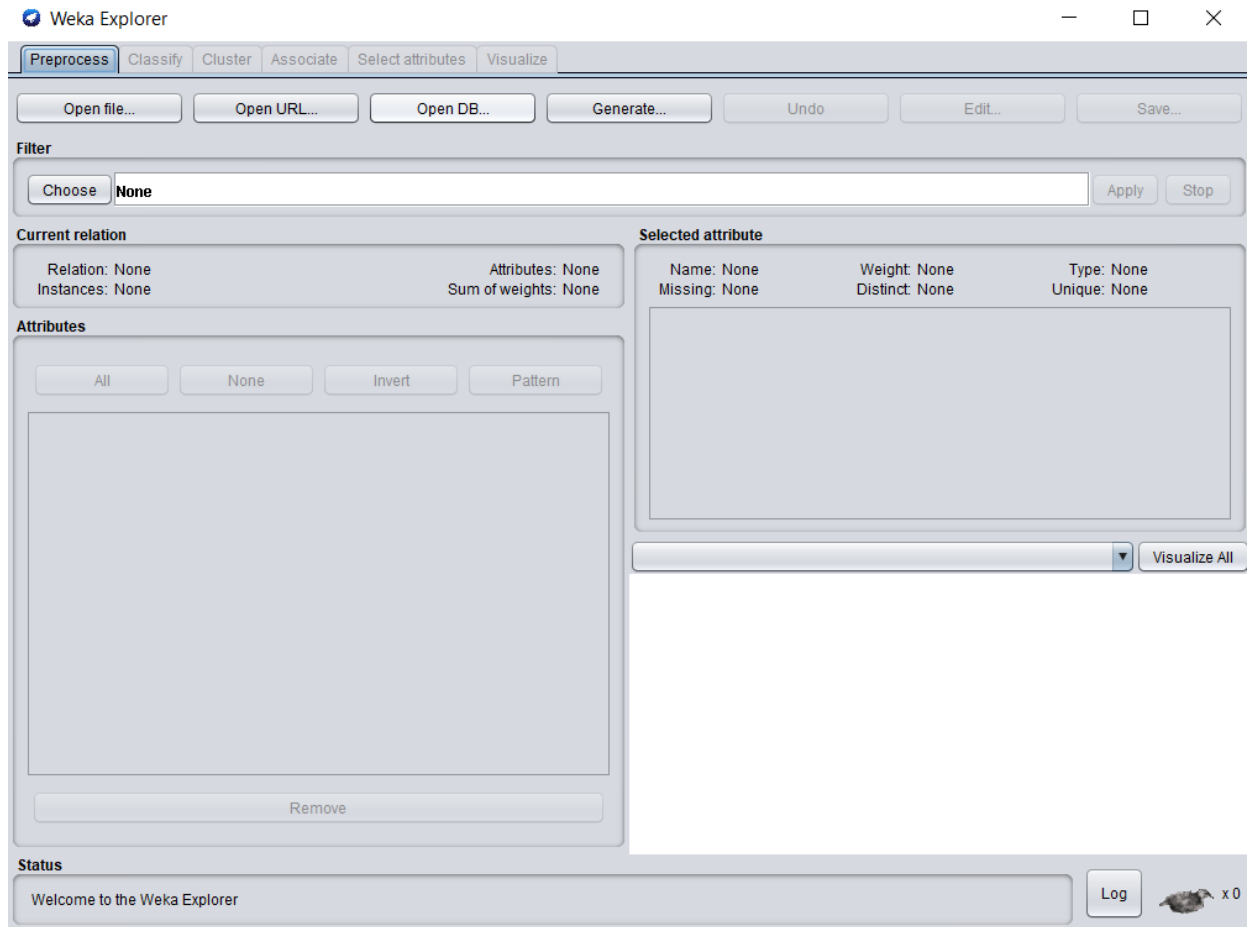
Phần mềm gồm 5 môi trường: Explorer, Experimenter, KnowledgeFlow, Workbench và Simple CLI.

➤ Bài báo cáo sẽ tập trung vào môi trường Explorer



a) Môi trường Explorer.

Giao diện môi trường Explorer gồm: preprocess, classify, cluster, associate, select attributes, visualize.

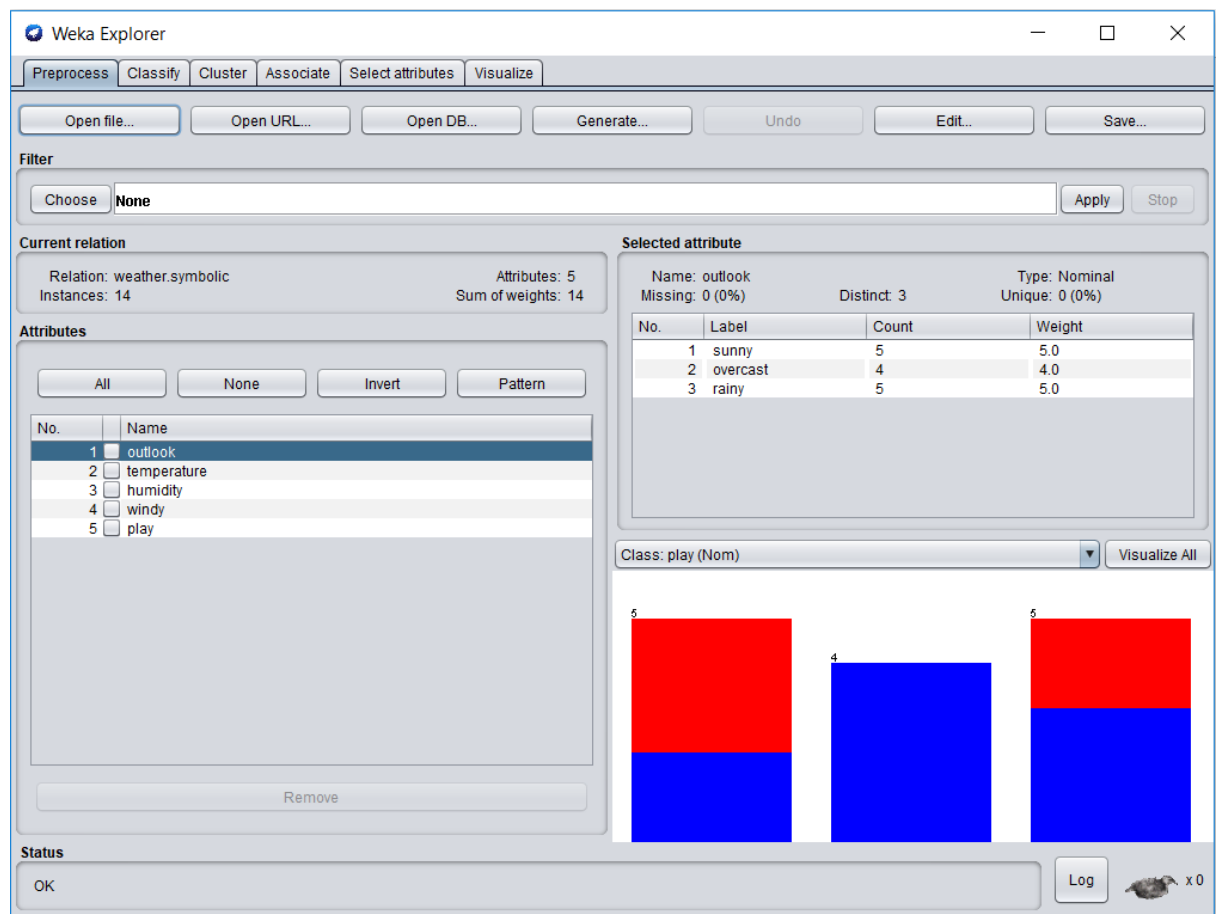


- **Preprocess – là bước tiền xử lý dữ liệu.**

Trong qui trình khai phá dữ liệu, công việc xử lý dữ liệu trước khi đưa vào các mô hình là rất cần thiết, bước này làm cho dữ liệu có được ban đầu qua thu thập dữ liệu (gọi là dữ liệu gốc ordinal data) có thể áp dụng được (thích hợp) với các mô hình khai phá dữ liệu (data mining model) cụ thể. Các công việc cụ thể của tiền xử lý dữ liệu bao gồm những công việc như:

- *Filtering Attributes:* Chọn các thuộc tính phù hợp với mô hình.

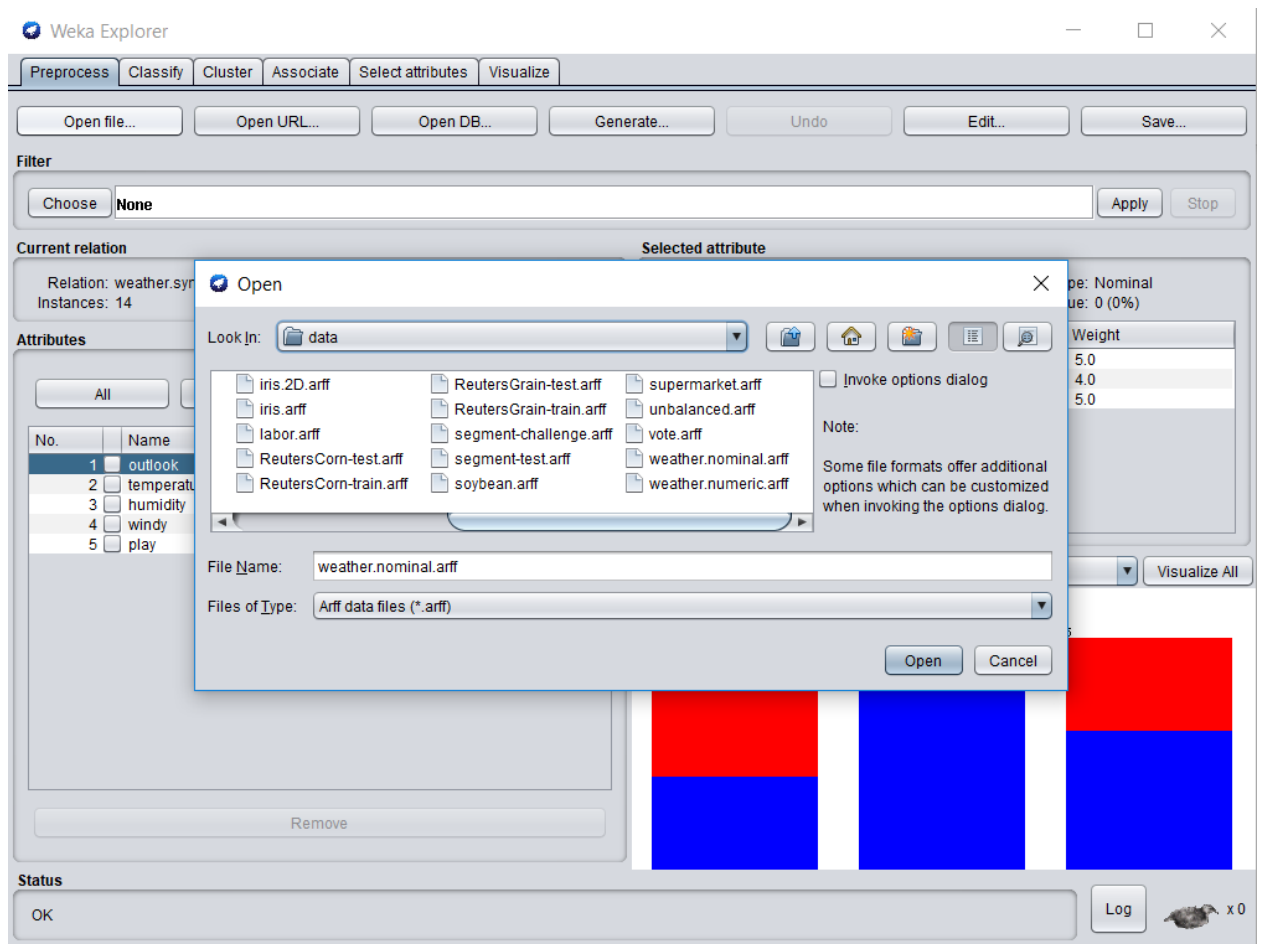
- *Filtering samples*: Lọc các mẫu (instances, patterns) dữ liệu cho mô hình.
- *Clean data*: Làm sạch dữ liệu như xóa bỏ các dữ liệu bất thường (Outlier).
- *Transformation*: Chuyển đổi dữ liệu cho phù hợp với các mô hình như chuyển đổi dữ liệu từ numeric qua nominal hay ordinal.
- *Discretization (rời rạc hóa dữ liệu)*: Nếu bạn có dữ liệu liên tục nhưng một vài mô hình chỉ áp dụng cho các dữ liệu rời rạc (luật kết hợp chẳng hạn) thì bạn phải thực hiện việc rời rạc hóa dữ liệu.



Dữ liệu có thể được nhập vào từ một tập tin có khuôn dạng ARFF, CSV.

Dữ liệu có thể được đọc từ một địa chỉ URL, hoặc từ một cơ sở dữ liệu thông qua JDBC.

- Chọn kiểu open ứng với nguồn dữ liệu để import một bộ dữ liệu vào weka.



■ Ví dụ của một tập dữ liệu

@relation weather

Tên của tập dữ liệu

@attribute outlook {sunny, overcast, rainy}

Thuộc tính kiểu định danh

@attribute temperature real

@attribute humidity real

Thuộc tính kiểu số

@attribute windy {TRUE, FALSE}

@attribute play {yes, no}

Thuộc tính phân lớp (mặc định là thuộc tính cuối cùng)

@data

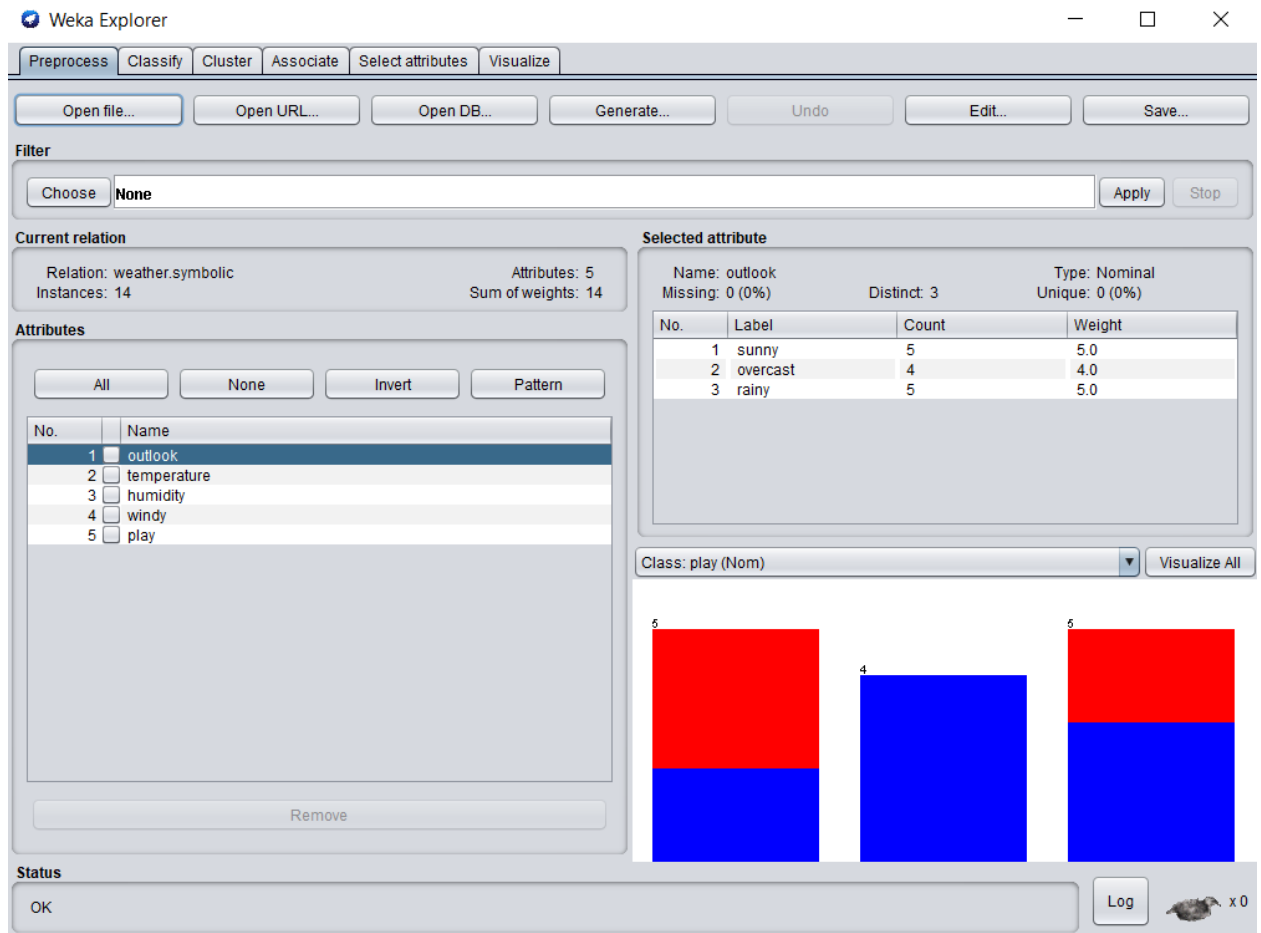
sunny, 85, 85, FALSE, no

overcast, 83, 86, FALSE, yes

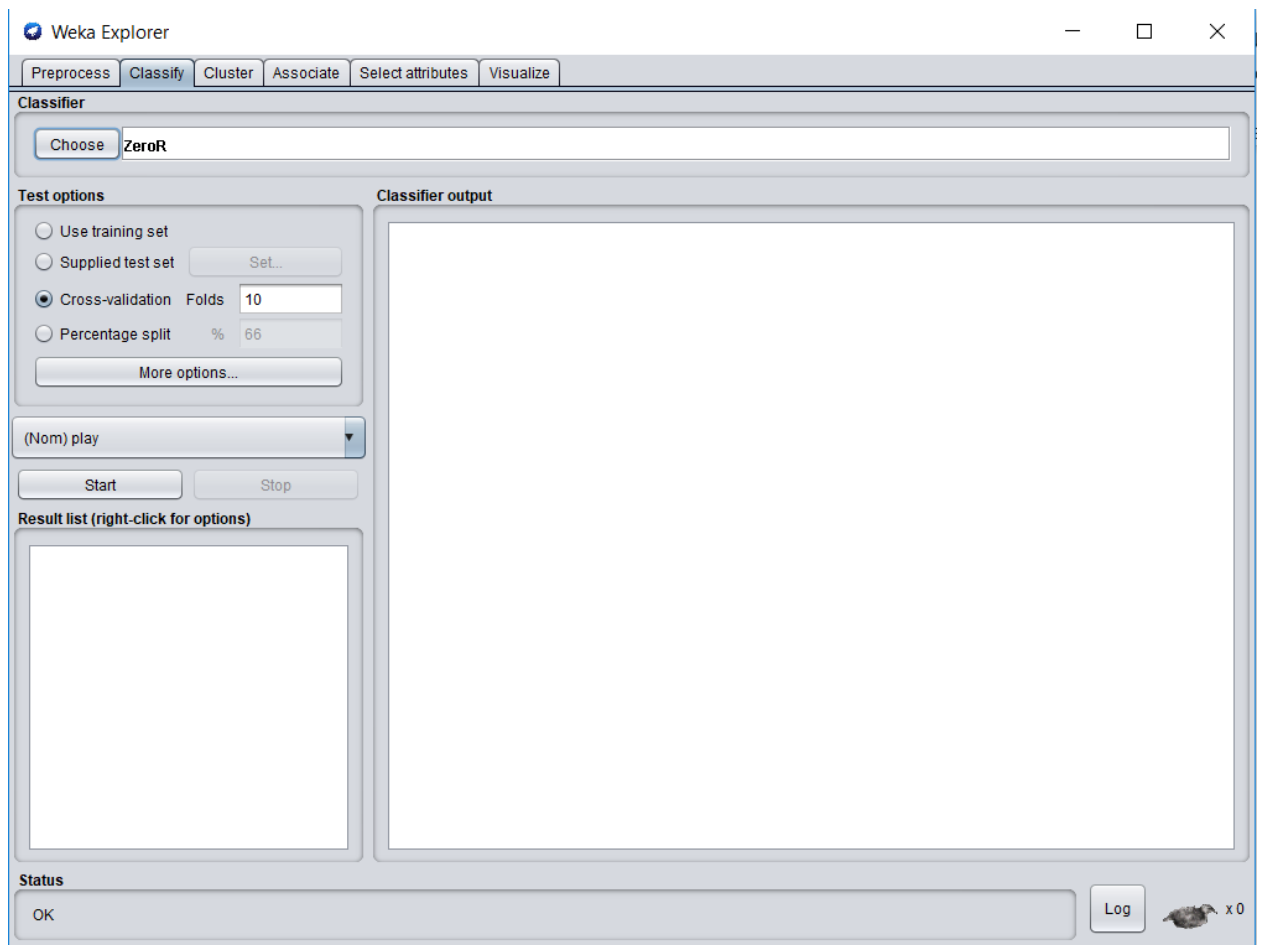
...

Các ví dụ (instances)

Sau khi mở tập dữ liệu ta có giao diện như sau:



- Phần **filter** thiết lập kiểu lọc (choose) và apply.
- Phần **Current relation** nêu các mô tả cơ bản của tập dữ liệu.
- Phần **Attributes** chứa các thuộc tính của đối tượng trong dataset.
- Phần **Selected attribute** cho biết một số đặc tính cơ bản của thuộc tính đang được chọn.
- Phần biểu đồ mô tả phân bố của thuộc tính đang chọn. Mỗi màu trên biểu đồ đại diện cho 1 class.
- **Classify – Phân lớp dữ liệu.**
Giao diện classify:



Các bộ phân lớp (Classifiers) của WEKA tương ứng với các mô hình dự đoán các đại lượng kiểu định danh (phân lớp) hoặc các đại lượng kiểu số (hồi quy/dự đoán).

Các kỹ thuật phân lớp được hỗ trợ bởi WEKA:

- Naïve Bayes classifier and Bayesian networks.
- Decision trees.
- Instance-based classifiers.
- Support vector machines.
- Neural networks.
- ...

🚦 Mô tả về giao diện Classify:

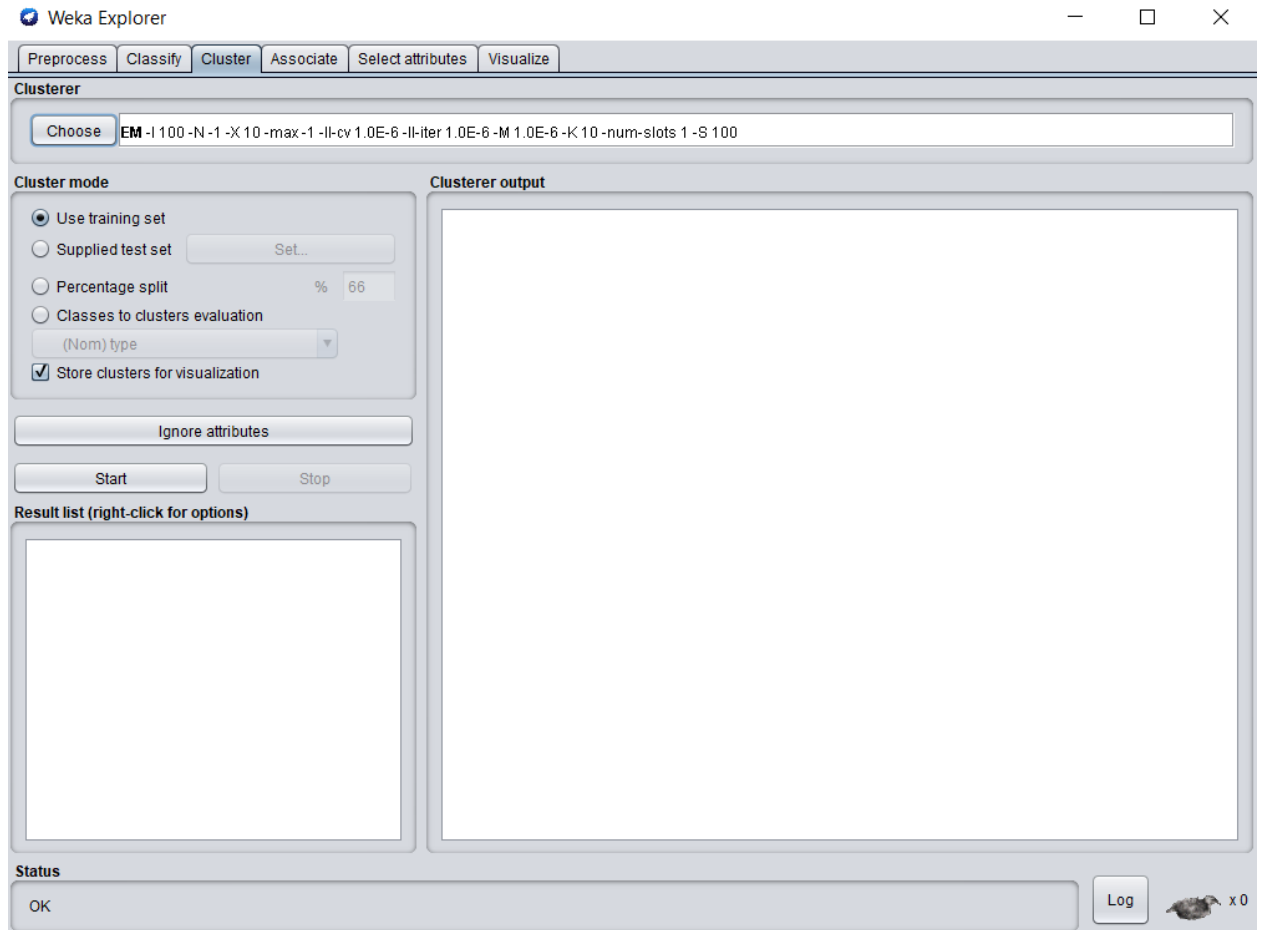
- **Classifiers** để chọn bộ phân lớp (gồm các bộ ở trên).
- **Test options** lựa chọn các tùy chọn cho việc kiểm tra.
 - **Use training set:** bộ phân loại học được sẽ được đánh giá trên tập học (training).

- **Supplied test set:** sử dụng một tập dữ liệu khác (với tập học) để cho việc đánh giá.
- **Cross-validation:** tập dữ liệu sẽ được chia đều thành k tập (folds) có kích thước xấp xỉ nhau, và bộ phân loại học được sẽ được đánh giá bởi phương pháp *cross-validation*.
- **Percentage split:** chỉ định tỷ lệ phân chia tập dữ liệu đối với việc đánh giá.
- **More options...**
 - **Output model:** hiển thị bộ phân lớp học được
 - **Output per-class stats:** hiển thị các thông tin thống kê về precision/recall đối với mỗi lớp
 - **Output entropy evaluation measures:** hiển thị đánh giá độ hỗn tạp (entropy) của tập dữ liệu.
 - **Output confusion matrix:** hiển thị thông tin về ma trận lỗi phân lớp (confusion matrix) đối với phân lớp học được.
 - **Store predictions for visualization:** các dự đoán của bộ phân lớp được lưu lại trong bộ nhớ, để có thể được hiển thị sau đó.
 - **Output predictions:** hiển thị chi tiết các dự đoán đối với tập kiểm tra
 - **Cost-sensitive evaluation:** các lỗi (của bộ phân lớp) được xác định dựa trên ma trận chi phí (cost matrix) chỉ định.
 - **Random seed for XVal / % Split:** chỉ định giá trị *random seed* được sử dụng cho quá trình lựa chọn ngẫu nhiên các ví dụ cho tập kiểm tra.
- **Classifier output** hiển thị các thông tin quan trọng
 - **Run information:** các tùy chọn đối với mô hình học tên của tập dữ liệu, số lượng các ví dụ, các thuộc tính, và phương pháp thí nghiệm.
 - **Classifier model (full training set):** biểu diễn (dạng text) của bộ phân lớp học được.

- **Predictions on test data:** thông tin chi tiết về các dự đoán của bộ phân lớp đối với tập kiểm tra.
 - **Summary:** các thống kê về mức độ chính xác của bộ phân lớp đối với phương pháp thí nghiệm đã chọn.
 - **Detailed Accuracy By Class:** thông tin chi tiết về mức độ chính xác của bộ phân lớp đối với mỗi lớp.
 - **Confusion Matrix:** các thành phần của ma trận này thể hiện số lượng các ví dụ kiểm tra (test instances) được phân lớp đúng và bị phân lớp sai.
- **Result list** cung cấp một số chức năng hữu ích:
 - **Save model:** lưu lại mô hình tương ứng với bộ phân lớp học được vào trong một tập tin nhị phân (binary file).
 - **Load model:** đọc lại một mô hình đã được học trước đó từ một tập tin nhị phân.
 - **Re-evaluate model on current test set:** đánh giá một mô hình (bộ phân lớp) học được trước đó đối với tập kiểm tra (test set) hiện tại.
 - **Visualize classifier errors:** hiển thị của số biểu đồ thể hiện các kết quả của việc phân lớp. Các ví dụ được phân lớp chính xác sẽ được biểu diễn bằng ký hiệu bởi dấu chéo (x), còn các ví dụ bị phân lớp sai sẽ được biểu diễn bằng ký hiệu ô vuông.
 - ...
- **Cluster – phân cụm dữ liệu.**

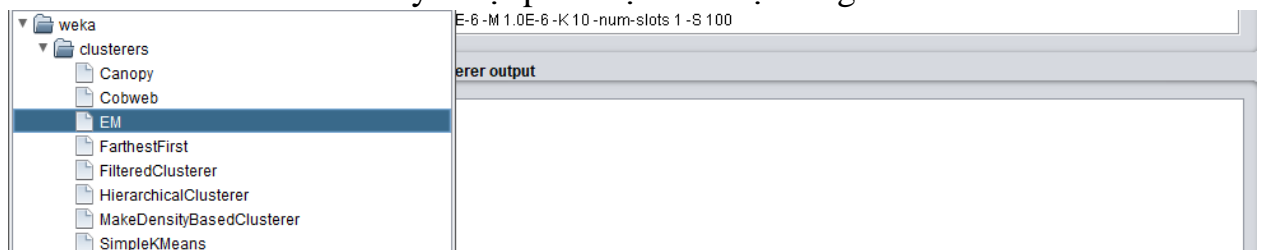
Các bộ phân cụm (Cluster builders) của WEKA tương ứng với các mô hình tìm các nhóm của các ví dụ tương tự đối với một tập dữ liệu.

Giao diện môi trường **cluster:**



✚ Mô tả giao diện **Cluster**:

- **Clusterers**: lựa chọn các kỹ thuật phân cụm.
Các kỹ thuật phân cụm hỗ trợ trong Weka:



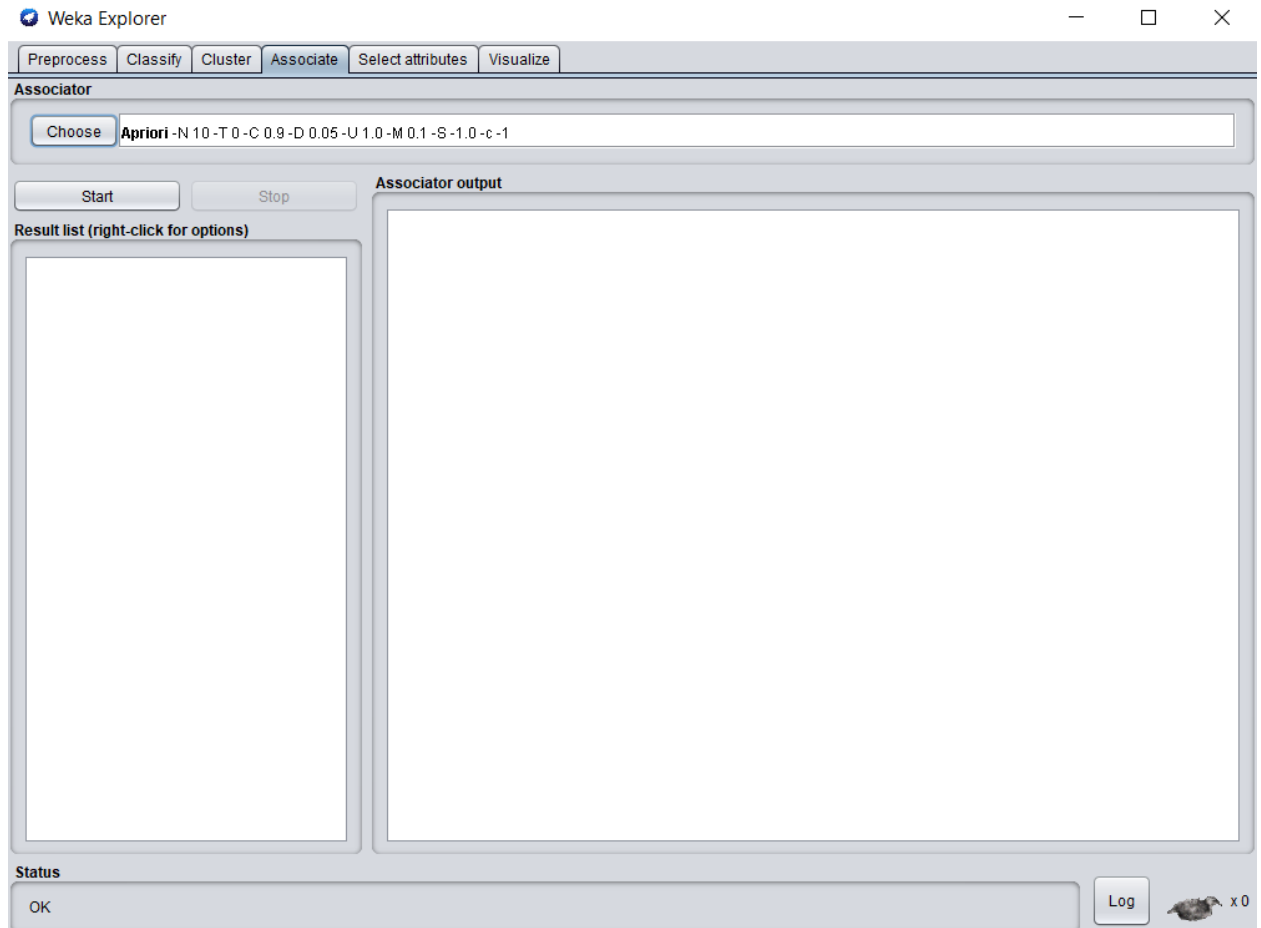
- **Cluster mode**: lựa chọn chế độ phân cụm.
 - **Use training set**: các cụm học được sẽ được kiểm tra đối với tập học.
 - **Supplied test set**: sử dụng một tập dữ liệu khác để kiểm tra các cụm học được.

- **Percentage split:** chỉ định tỷ lệ phân chia tập dữ liệu ban đầu cho việc xây dựng tập kiểm tra.
- **Classes to clusters evaluation:** so sánh độ chính xác của các cụm học được đối với các lớp được chỉ định.

✚ Sau khi thiết lập các kỹ thuật phân cụm cũng như mode ta nhấn **start** để bắt đầu phân cụm.

- **Associate – luật kết hợp.**

Giao diện **associate**:

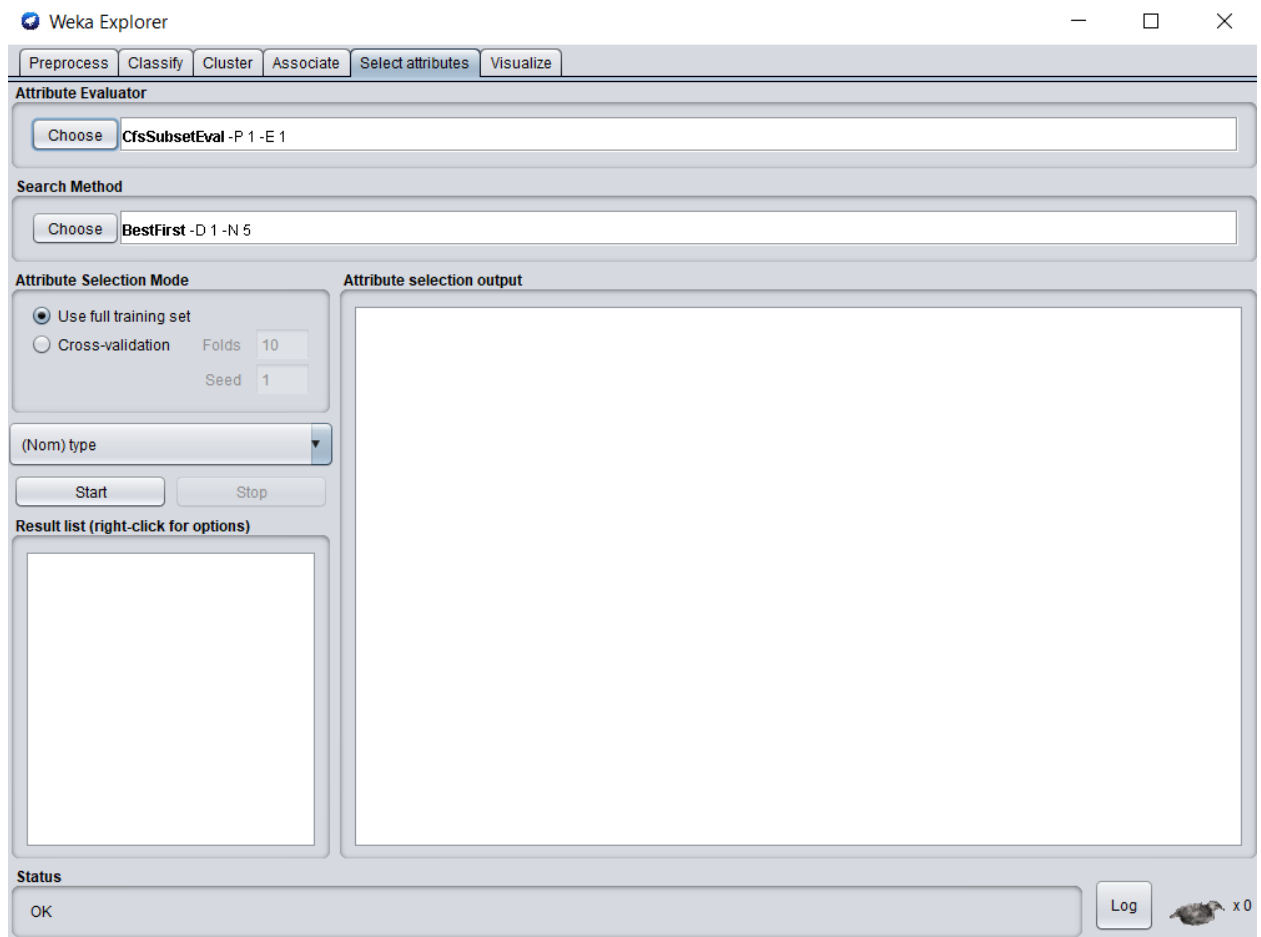


✚ Giải thích giao diện:

- **Associator:** lựa chọn một mô hình (giải thuật) phát hiện luật kết hợp.
- **Associator output** hiển thị các thông tin quan trọng:

- **Run information:** Các tùy chọn đối với mô hình phát hiện kết hợp, tên tập dữ liệu, thuộc tính, số lượng các ví dụ.
 - **Assocator mode:** Biểu diễn (dạng text) của tập các luật phát hiện kết hợp.
 - Độ hỗ trợ tối thiểu.
 - Độ tin cậy tối thiểu.
 - Liệt kê các luật kết hợp tìm được.
- **Select attributes – lựa chọn thuộc tính.**

Giao diện **Select attributes**:



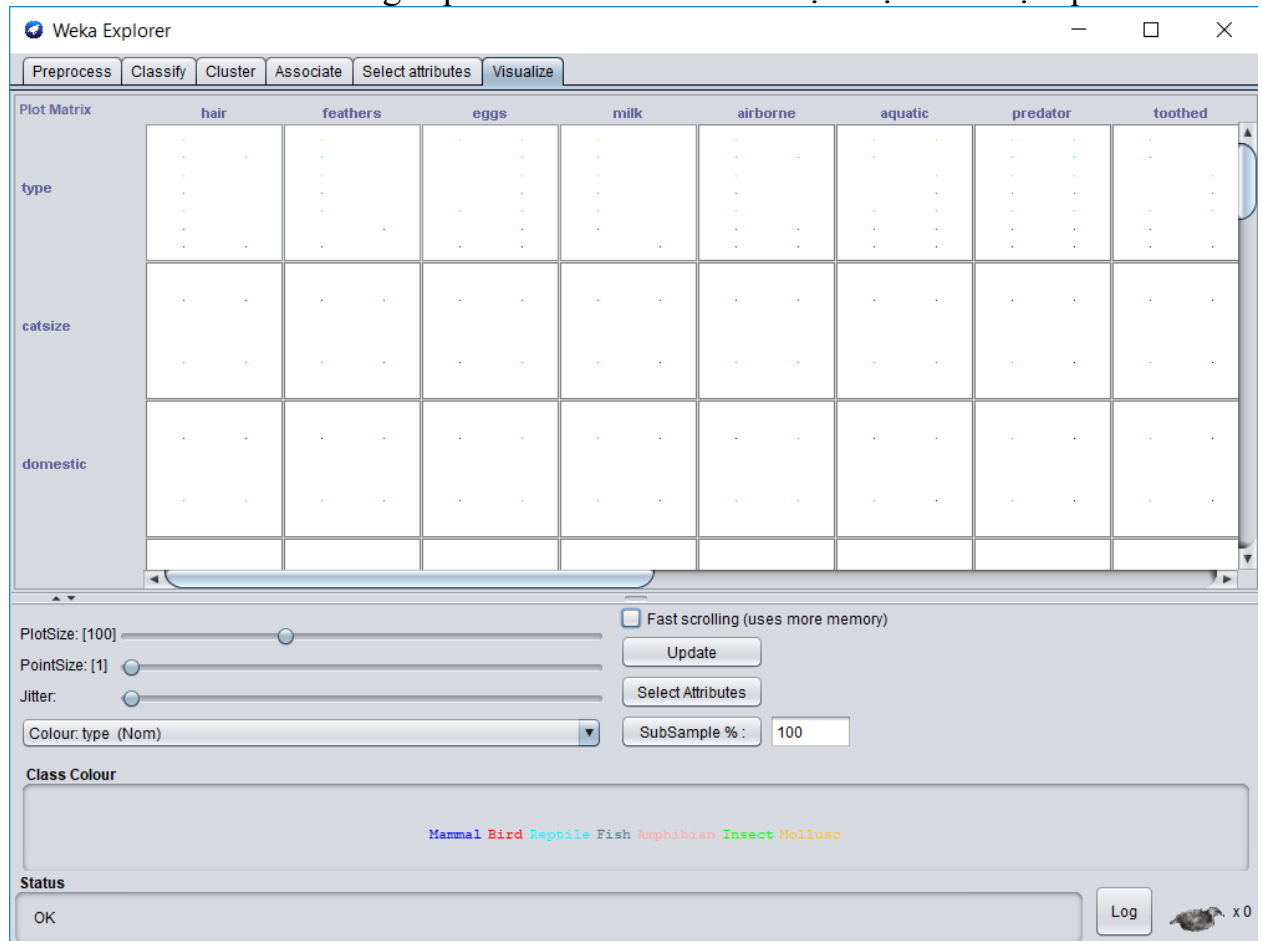
✚ Giới thiệu giao diện cơ bản:

- **Attribute Evaluator:** Để xác định một phương pháp đánh giá mức độ phù hợp của các thuộc tính
Vd: correlation-based, wrapper, information gain, chisquared,...

- **Search Method:** Để xác định một phương pháp (thứ tự) xét các thuộc tính
Vd: best-first, random, exhaustive, ranking,...

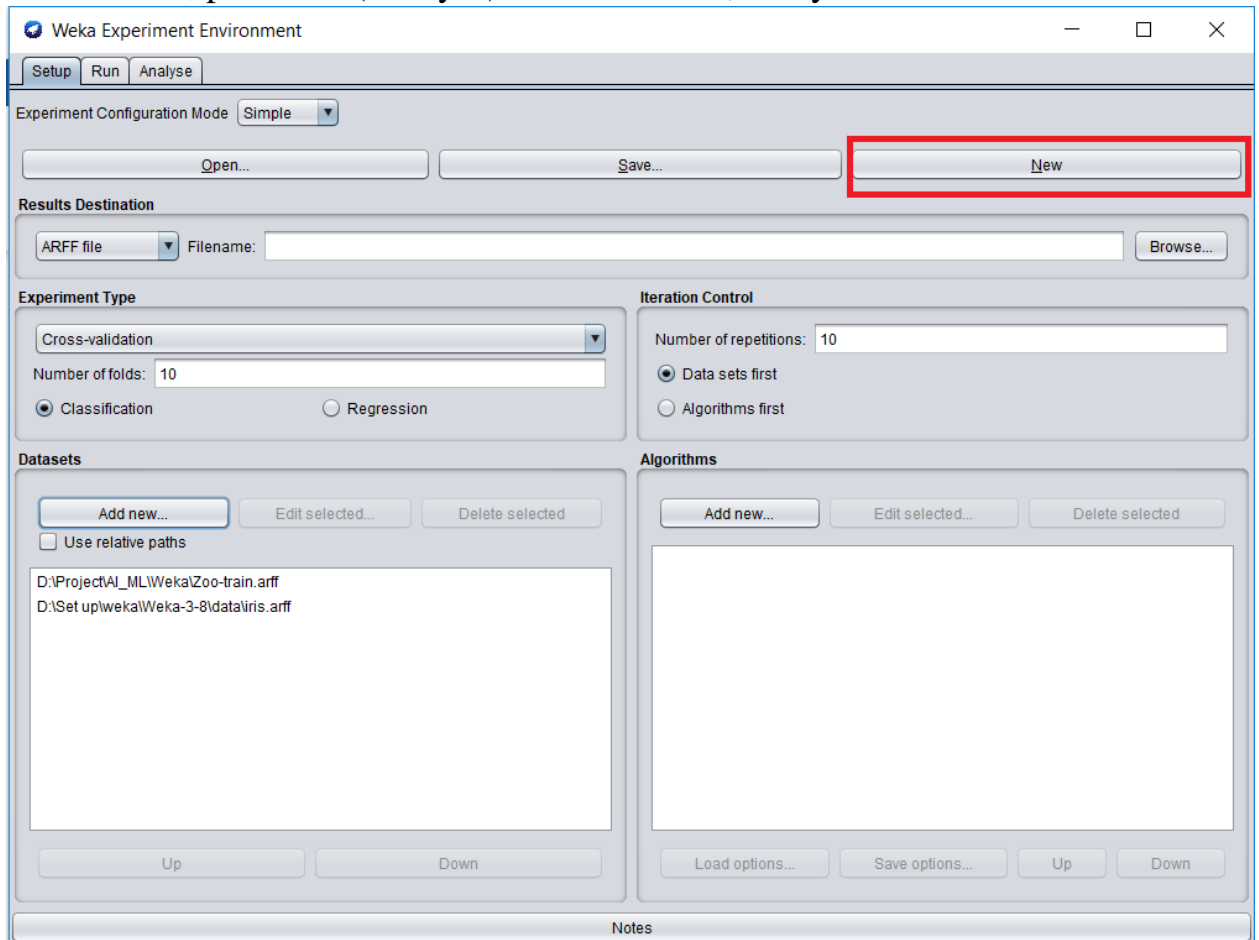
- **Visualize – hiển thị dữ liệu:**

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




- **Weka có thể hiển thị:**
 - Mỗi thuộc tính riêng lẻ (1-D visuallization)
 - Một cặp thuộc tính (2-D visuallization)
- Các giá trị (các nhãn) lớp khác nhau sẽ được hiển thị bằng các màu khác nhau.
- Thanh trượt **Jitter** hỗ trợ việc hiển thị rõ ràng hơn, khi có quá nhiều ví dụ (điểm) tập trung xung quanh một vị trí trên biểu đồ.
- Tính năng phóng to/thu nhỏ (bằng cách tăng/giảm giá trị của **PlotSize** và **PointSize**).

b) Experimenter: Môi trường cho phép thực nghiệm (Setup, Run), so sánh, phân tích (Analyse) các mô hình học máy.

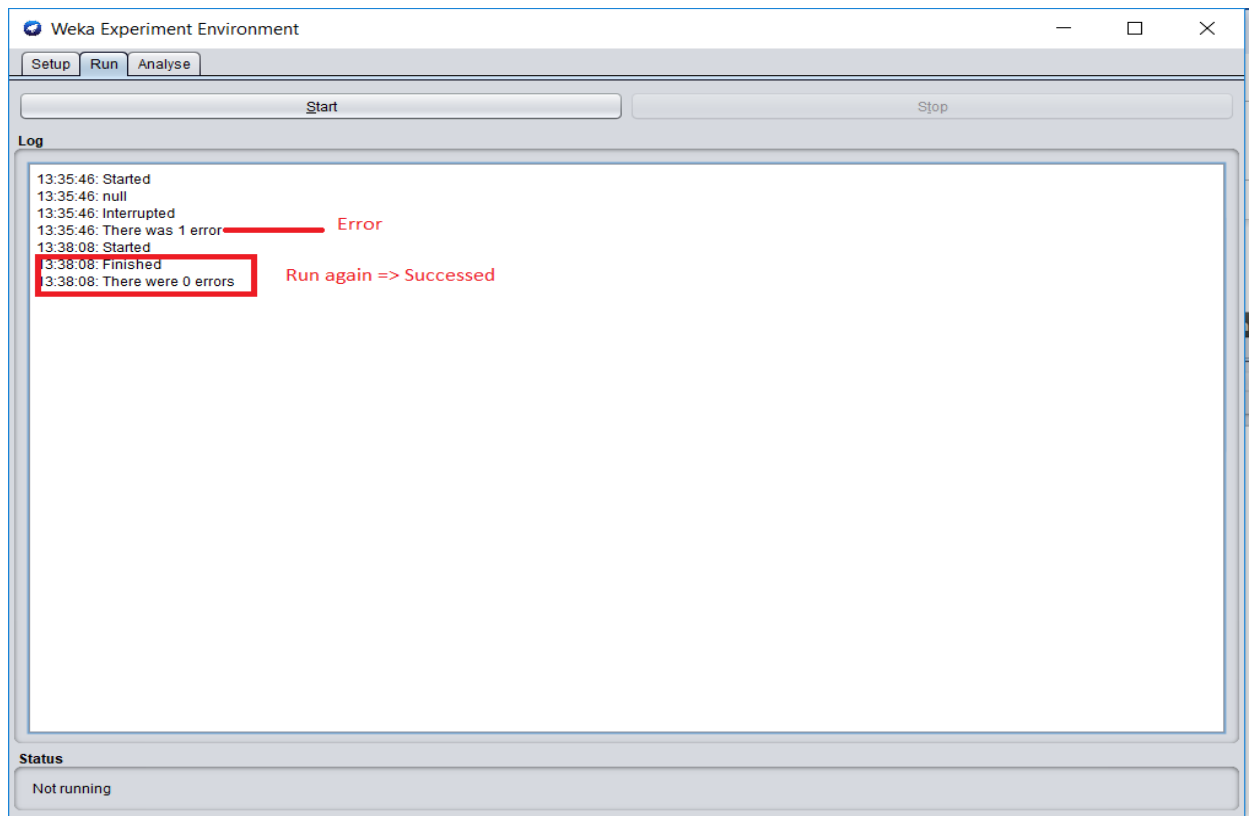


Tab Set up.

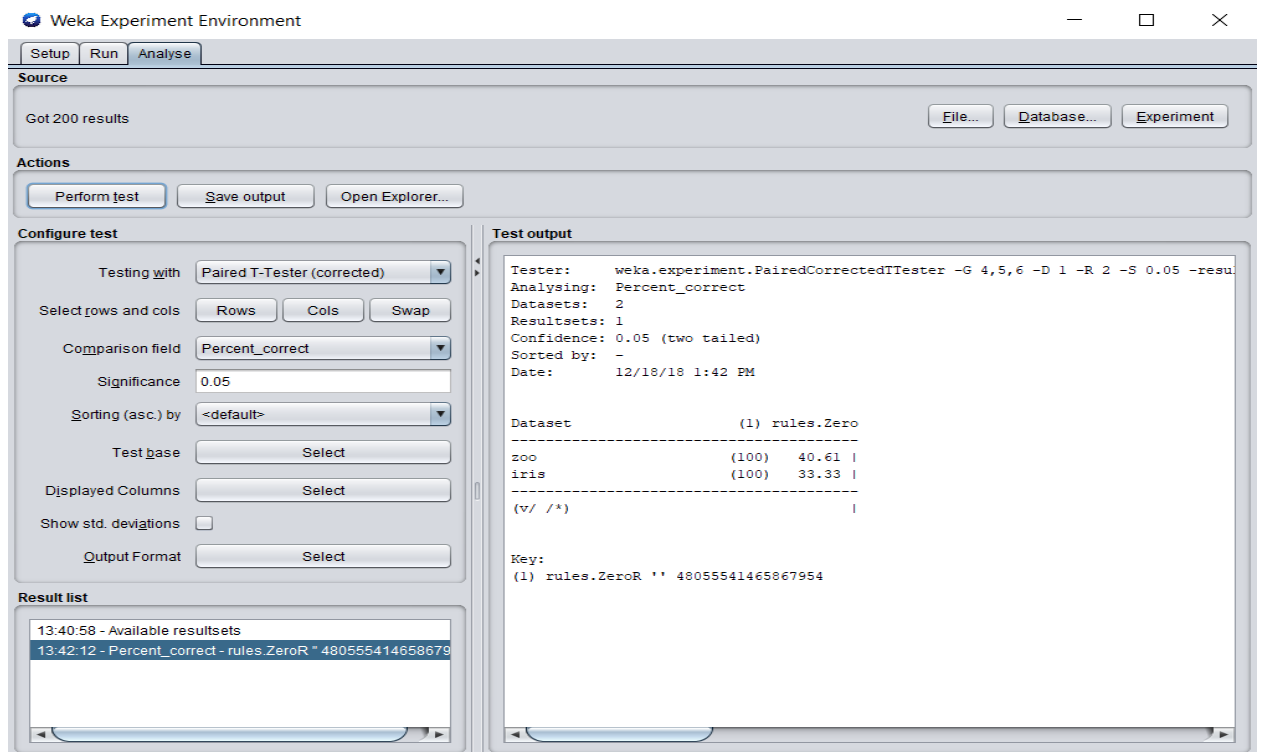
- Nhấn **New** để tạo một thử nghiệm mới, hoặc có thể thử nghiệm đã lưu trước đó qua **Open**.
- **Result Destination** lựa chọn format cũng như đường dẫn lưu file kết quả.
- **Experiment type**.
- **Iteration Control**.
- **Datasets** thêm dữ liệu phục vụ thử nghiệm.
- **Algorithms** lựa chọn thuật toán cho quá trình thử nghiệm.

 Sau khi **Set up** ta chuyển qua tab **Run** để tiến hành chạy thử nghiệm.

Nhấn **Start** để tiến hành chạy và xem log của chương trình đã chạy.



✚ Nếu đã chạy thành công ta chuyển sang tab **Analyze**.



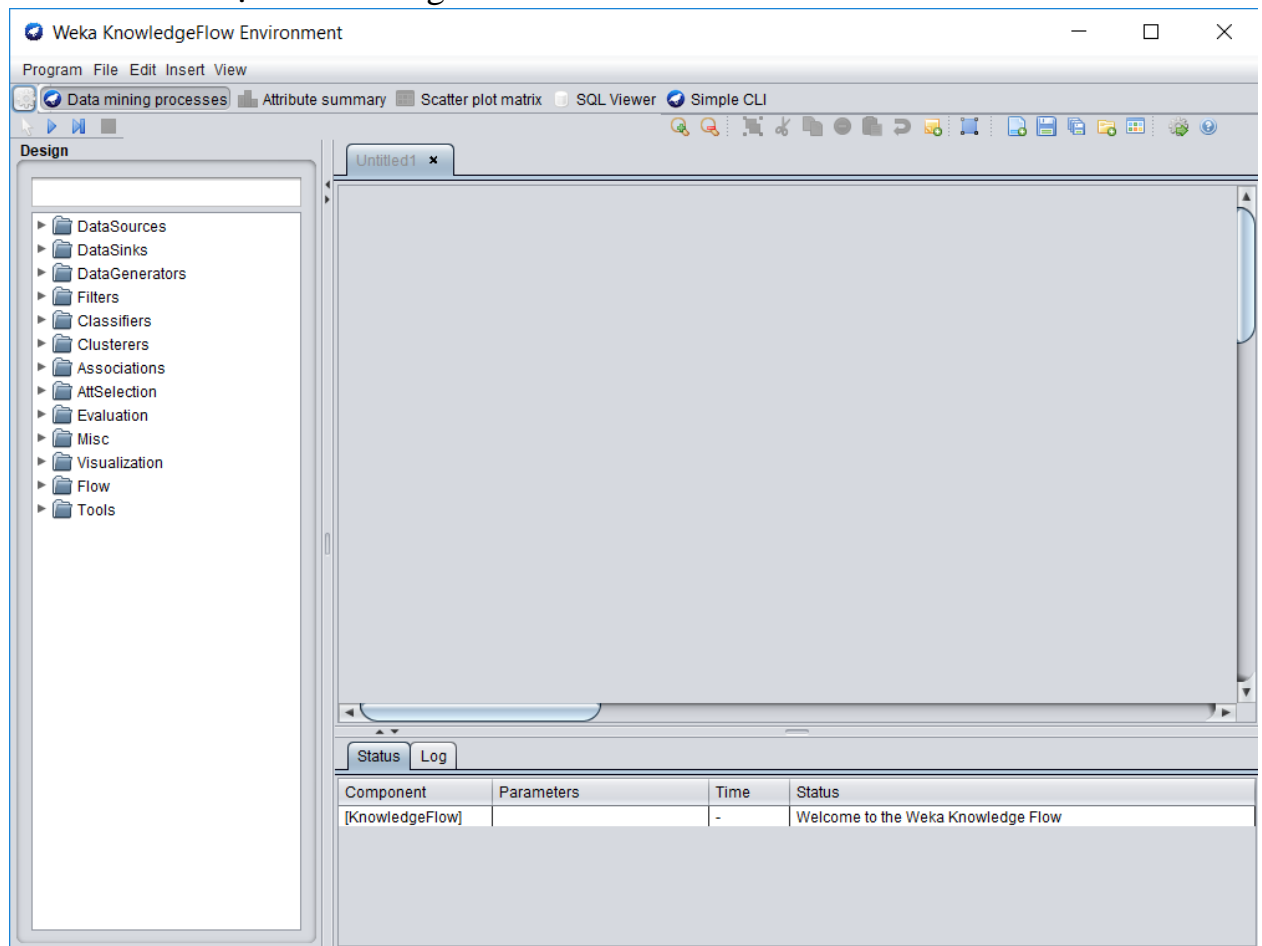
✚ Giới thiệu qua về giao diện:

- **Configure test** thiết lập tùy chọn muốn kiểm tra
- **Action:** nhấn **Perform test** để chạy theo config của mình.
- **Test output** hiển thị kết quả.

c) Môi trường KnowledgeFlow.

Môi trường cho phép bạn tương tác đồ họa kiểu kéo/thả để thiết kế các bước (các thành phần) của một thí nghiệm.

Giao diện môi trường:

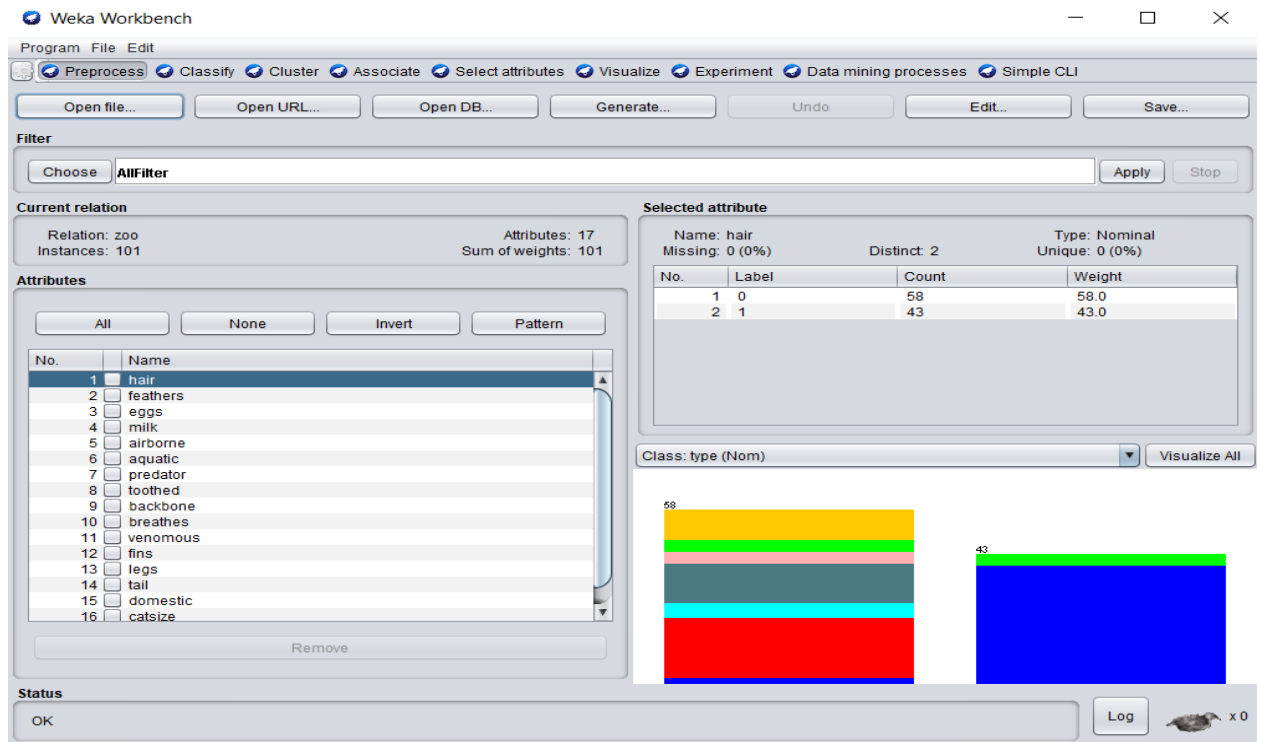


✚ Sơ lược về giao diện chính.

- Chúng ta chỉ cần kéo các đối tượng, thuật toán từ khung **Design** và thả vào khung đang thiết kế.
- Nhấn biểu tượng start để chạy chương trình.

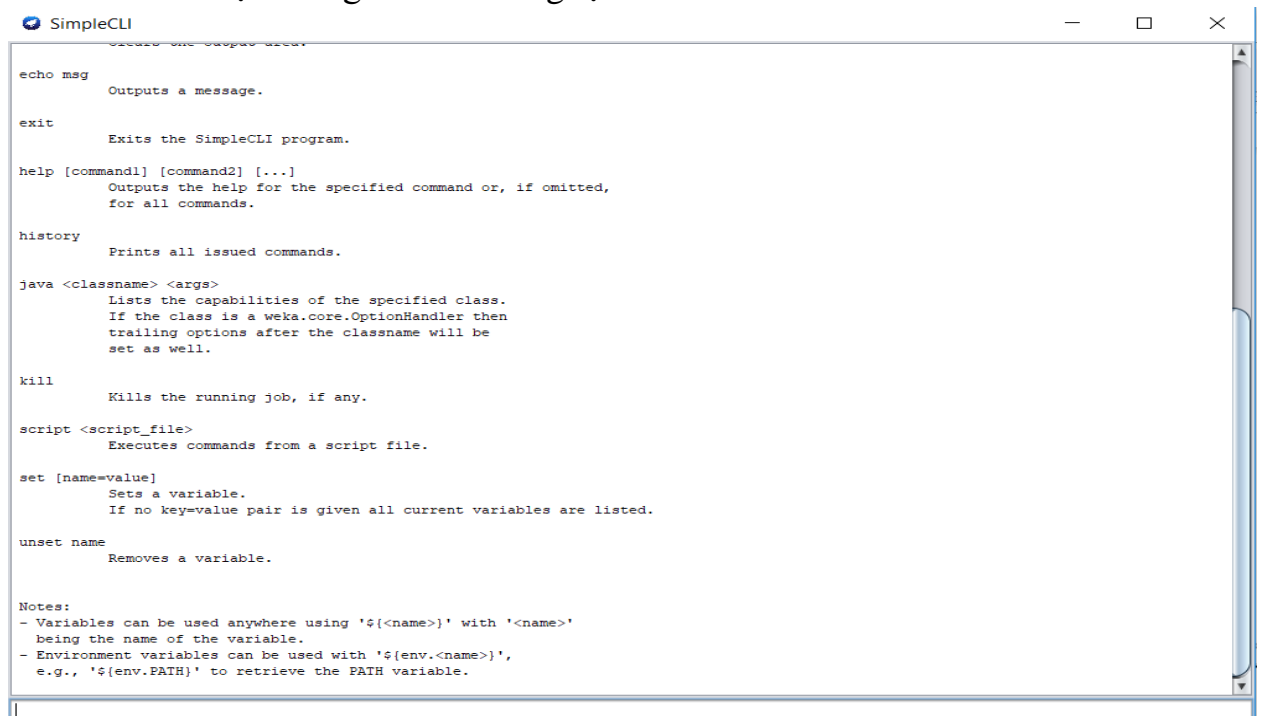
d) Workbench.

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



e) Simple CLI.

Giao diện đơn giản kiểu dòng lệnh.



II. Sử dụng Weka để chạy thuật toán ID3.

1. Mô tả dữ liệu Zoo.

- Số mẫu trong tập dữ liệu là 101 loại động vật.
- Tên và ý nghĩa các thuộc tính (18):

STT	Thuộc tính	Kiểu dữ liệu	Ý nghĩa	Miền giá trị
1	name	String	Tên động vật	Chuỗi
2	hair	Boolean	Tóc	1,0
3	feathers	Boolean	Lông	1,0
4	eggs	Boolean	Đẻ trứng	1,0
5	milk	Boolean	Có sữa	1,0
6	airbone	Boolean	Biết bay	1,0
7	aquatic	Boolean	Động vật sống dưới nước	1,0
8	predator	Boolean	Động vật ăn thịt	1,0
9	toothed	Boolean	Có răng	1,0
10	backbone	Boolean	Có xương sống	1,0
11	breathes	Boolean	Có thở	1,0
12	venomous	Boolean	Có độc	1,0
13	fins	Boolean	Có vây	1,0
14	legs	Numeric	Số chân	0,2,4,5,6,8
15	tail	Boolean	Có đuôi	1,0
16	domestic	Boolean	Động vật được nuôi	1,0
17	type	Numeric	Lớp động vật	1,2,3,4,5,6,7

- Danh sách phân lớp, đặt tên cho mỗi phân lớp:

Type	Đặt tên	Số lượng	Động vật thuộc lớp
1	Mammal	44	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	Bird	20	chicken, crow, dove, duck, flamingo, gull, hawk, kiwi, lark, ostrich, parakeet, penguin, pheasant, rhea, skimmer, skua, sparrow, swan, vulture, wren.
3	Reptile	5	pitviper, seasnake, slowworm, tortoise, tuatara.
4	Fish	13	bass, carp, catfish, chub, dogfish, haddock, herring, pike, piranha, seahorse, sole, stingray, tuna.
5	Amphibian	3	frog, newt, toad
6	Insect	8	flea, gnat, honeybee, housefly, ladybird, moth, termite, wasp.
7	Mollusc	10	clam, crab, crayfish, lobster, octopus, scorpion, seawasp, slug, starfish, worm

2. Sử dụng thuật toán ID3 để học ra cây quyết định.

Sử dụng thuật toán id3 cho thuộc tính type. Tên của động vật chúng ta không cần xét đến nên ta sử dụng bộ dữ liệu không có thuộc tính name: **Zoo-train.arff**

- Test option: 10-fold cross validation

Ta được cây quyết định:

```

Test mode:    10-fold cross-validation

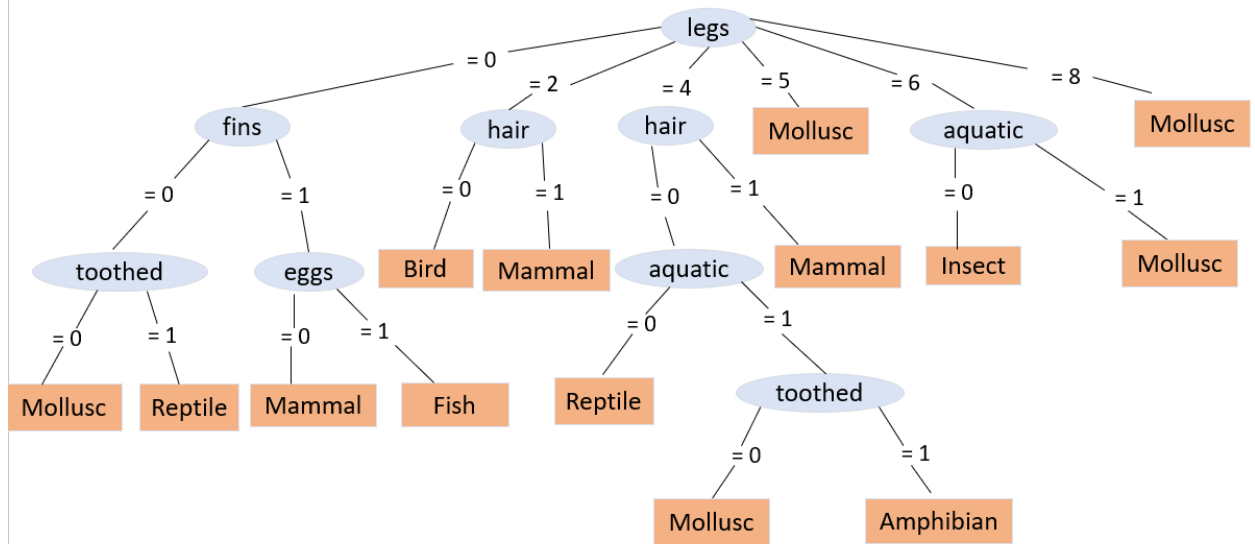
=== Classifier model (full training set) ===

Id3

legs = 0
| fins = 0
| | toothed = 0: Mollusc
| | toothed = 1: Reptile
| fins = 1
| | eggs = 0: Mammal
| | eggs = 1: Fish
legs = 2
| hair = 0: Bird
| hair = 1: Mammal
legs = 4
| hair = 0
| | aquatic = 0: Reptile
| | aquatic = 1
| | | toothed = 0: Mollusc
| | | toothed = 1: Amphibian
| hair = 1: Mammal
legs = 5: Mollusc
legs = 6
| aquatic = 0: Insect
| aquatic = 1: Mollusc
legs = 8: Mollusc

```

Cây được vẽ lại:



- Kết quả 5 mẫu đã cho dự đoán từ cây quyết định đã sinh ra là:

```

@relation zoo_test

@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 {0,2,4,5,6,8}
@attribute tail {0,1}
@attribute domestic {0,1}
@attribute catsize {0,1}
@attribute type {Mammal,Bird,Reptile,Fish,Amphibian,Insect,Mollusc}

@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,?
0,0,1,0,0,0,1,1,1,1,0,0,1,0,0,?
0,0,1,0,0,1,1,1,1,0,0,1,0,1,0,0,?
0,0,1,0,0,1,1,1,1,0,0,1,0,1,0,0,?
0,0,1,0,0,1,1,1,1,1,0,0,4,1,0,0,?

=== Predictions on test set ===

inst#,actual,predicted,error,prediction
1,1:?,1:Mammal,,1
2,1:?,2:Bird,,1
3,1:?,3:Reptile,,1
4,1:?,4:Fish,,1
5,1:?,5:Amphibian,,1

```

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

III. Chạy các thuật toán khác.

1. Chương trình python cho giải thuật Naïve Bayes.
2. Dùng weka để chạy thêm các thuật toán khác (3 thuật toán).

Vẫn sử dụng tập dữ liệu Zoo-train.arff (từ tập Zoo.arff nhưng loại bỏ thuộc tính name vì thuộc tính này không ảnh hưởng đến thí nghiệm).

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

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

Filter: Choose None Apply Stop

Current relation: Relation: zoo, Instances: 101, Attributes: 17, Sum of weights: 101

Attributes: All None Invert Pattern

No.	Name
1	hair
2	feathers
3	eggs
4	milk
5	airborne
6	aquatic
7	predator
8	toothed
9	backbone
10	breathes
11	venomous
12	fins
13	legs
14	tail
15	domestic
16	catsize
17	type

Remove

Status: OK Log x 0

Selected attribute: Name: hair, Missing: 0 (0%), Distinct: 2, Type: Nominal, Unique: 0 (0%)

No.	Label	Count	Weight
1	0	58	58.0
2	1	43	43.0

Class: type (Nom) Visualize All

Ở **Test option** ta chọn dữ liệu test từ tập train (Use training set) và tiến hành chạy 3 thuật toán:

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier: Choose NaiveBayes

Test options: Use training set (selected), Supplied test set, Cross-validation (Folds: 10), Percentage split (%: 66), More options...

(Nom) type Start Stop

Result list (right-click for options)

Status: OK Log x 0

➤ **Naïve Bayes:**

```
=== Run information ===

Scheme:      weka.classifiers.bayes.NaiveBayes
Relation:    zoo
Instances:   101
Attributes:  17
             hair
             feathers
             eggs
             milk
             airborne
             aquatic
             predator
             toothed
             backbone
             breathes
             venomous
             fins
             legs
             tail
             domestic
             catsize
             type
Test mode:    evaluate on training data
```

Thông tin chạy với thuật toán **Naïve Bayes** và test mode (use training set) như hình trên.

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

Naive Bayes Classifier

Attribute	Class Mammal (0.39)	Bird (0.19)	Reptile (0.06)	Fish (0.13)	Amphibian (0.05)	Insect (0.08)	Mollusc (0.1)
=====							
hair							
0	3.0	21.0	6.0	14.0	5.0	5.0	11.0
1	40.0	1.0	1.0	1.0	1.0	5.0	1.0
[total]	43.0	22.0	7.0	15.0	6.0	10.0	12.0
feathers							
0	42.0	1.0	6.0	14.0	5.0	9.0	11.0
1	1.0	21.0	1.0	1.0	1.0	1.0	1.0
[total]	43.0	22.0	7.0	15.0	6.0	10.0	12.0
eggs							
0	41.0	1.0	2.0	1.0	1.0	1.0	2.0
1	2.0	21.0	5.0	14.0	5.0	9.0	10.0
[total]	43.0	22.0	7.0	15.0	6.0	10.0	12.0
milk							
0	1.0	21.0	6.0	14.0	5.0	9.0	11.0
1	42.0	1.0	1.0	1.0	1.0	1.0	1.0
[total]	43.0	22.0	7.0	15.0	6.0	10.0	12.0
airborne							
0	40.0	5.0	6.0	14.0	5.0	3.0	11.0
1	3.0	17.0	1.0	1.0	1.0	7.0	1.0
[total]	43.0	22.0	7.0	15.0	6.0	10.0	12.0
aquatic							
0	36.0	15.0	5.0	1.0	1.0	9.0	5.0
1	7.0	7.0	2.0	14.0	5.0	1.0	7.0
[total]	43.0	22.0	7.0	15.0	6.0	10.0	12.0
predator							
0	20.0	12.0	2.0	5.0	2.0	8.0	3.0
1	23.0	10.0	5.0	10.0	4.0	2.0	9.0
[total]	43.0	22.0	7.0	15.0	6.0	10.0	12.0
toothed							
0	2.0	21.0	2.0	1.0	1.0	9.0	11.0
1	41.0	1.0	5.0	14.0	5.0	1.0	1.0
[total]	43.0	22.0	7.0	15.0	6.0	10.0	12.0
backbone							
0	1.0	1.0	1.0	1.0	1.0	9.0	11.0
1	42.0	21.0	6.0	14.0	5.0	1.0	1.0
[total]	43.0	22.0	7.0	15.0	6.0	10.0	12.0
breathes							
0	1.0	1.0	2.0	14.0	1.0	1.0	8.0
1	42.0	21.0	5.0	1.0	5.0	9.0	4.0
[total]	43.0	22.0	7.0	15.0	6.0	10.0	12.0
venomous							
0	42.0	21.0	4.0	13.0	4.0	7.0	9.0
1	1.0	1.0	3.0	2.0	2.0	3.0	3.0
[total]	43.0	22.0	7.0	15.0	6.0	10.0	12.0
fins							
0	38.0	21.0	6.0	1.0	5.0	9.0	11.0
1	5.0	1.0	1.0	14.0	1.0	1.0	1.0
[total]	43.0	22.0	7.0	15.0	6.0	10.0	12.0
legs							
0	4.0	1.0	4.0	14.0	1.0	1.0	5.0
2	8.0	21.0	1.0	1.0	1.0	1.0	1.0
4	32.0	1.0	3.0	1.0	5.0	1.0	2.0
5	1.0	1.0	1.0	1.0	1.0	1.0	2.0
6	1.0	1.0	1.0	1.0	1.0	9.0	3.0
8	1.0	1.0	1.0	1.0	1.0	1.0	3.0
[total]	47.0	26.0	11.0	19.0	10.0	14.0	16.0
tail							
0	7.0	1.0	1.0	1.0	4.0	9.0	10.0
1	36.0	21.0	6.0	14.0	2.0	1.0	2.0
[total]	43.0	22.0	7.0	15.0	6.0	10.0	12.0
domestic							
0	34.0	18.0	6.0	13.0	5.0	8.0	11.0
1	9.0	4.0	1.0	2.0	1.0	2.0	1.0
[total]	43.0	22.0	7.0	15.0	6.0	10.0	12.0
catsize							
0	10.0	15.0	5.0	10.0	5.0	9.0	10.0
1	33.0	7.0	2.0	5.0	1.0	1.0	2.0
[total]	43.0	22.0	7.0	15.0	6.0	10.0	12.0

Kết quả tổng kết cho ta biết:

- Số mẫu phân lớp đúng là 101 tương ứng 100%
- Số mẫu phân lớp sai 0 tương ứng 0%

- Các giá trị về độ đo lỗi như hình.
=> Độ chính xác thuật toán **Naïve Bayes** trong trường hợp này là 100%, tuy nhiên với số lượng mẫu ít như vậy chúng ta chưa kết luận được thuật toán nào tốt hơn.

=== Evaluation on training set ===

Time taken to test model on training data: 0.28 seconds

=== Summary ===

Correctly Classified Instances	101	100	%
Incorrectly Classified Instances	0	0	%
Kappa statistic	1		
Mean absolute error	0.0093		
Root mean squared error	0.0478		
Relative absolute error	4.2516	%	
Root relative squared error	14.4996	%	
Total Number of Instances	101		

➤ Locally weighted learning (LWL):

The screenshot shows the Weka Explorer window with the 'Classify' tab selected. The 'Classifier' dropdown is set to 'LWL -U 0 -K -1 -A "weka.core.neighboursearch.LinearNNSearch -A "weka.core.EuclideanDistance -R first-last"' -W weka.classifiers.trees.DecisionStump'. The 'Test options' section shows 'Use training set' selected. The 'Classifier output' pane displays the following information:

```

=== Run information ===

Scheme:      weka.classifiers.lazy.LWL -U 0 -K -1 -A "weka.core.neighboursearch.LinearNNSearch -A
Relation:    zoo
Instances:   101
Attributes:  17
             hair
             feathers
             eggs
             milk
             airborne
             aquatic
             predator
             toothed
             backbone
             breathes
             venomous
             fins
             legs
             tail
             domestic
             catsize
             type

Test mode:   evaluate on training data

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

Locally weighted learning
  
```

The 'Result list' shows a single entry: '10:12:17 - lazy.LWL'. The 'Status' bar at the bottom indicates 'OK'.

```
=== Evaluation on training set ===
```

```
Time taken to test model on training data: 0.04 seconds
```

```
=== Summary ===
```

Correctly Classified Instances	92	91.0891 %
Incorrectly Classified Instances	9	8.9109 %
Kappa statistic	0.8797	
Mean absolute error	0.05	
Root mean squared error	0.1437	
Relative absolute error	22.8477 %	
Root relative squared error	43.6307 %	
Total Number of Instances	101	

Kết quả chạy:

- Có 92 mẫu được phân lớp đúng – tương ứng 91,0891 %.
- Có 9 mẫu được phân lớp sai (8,9109 %).

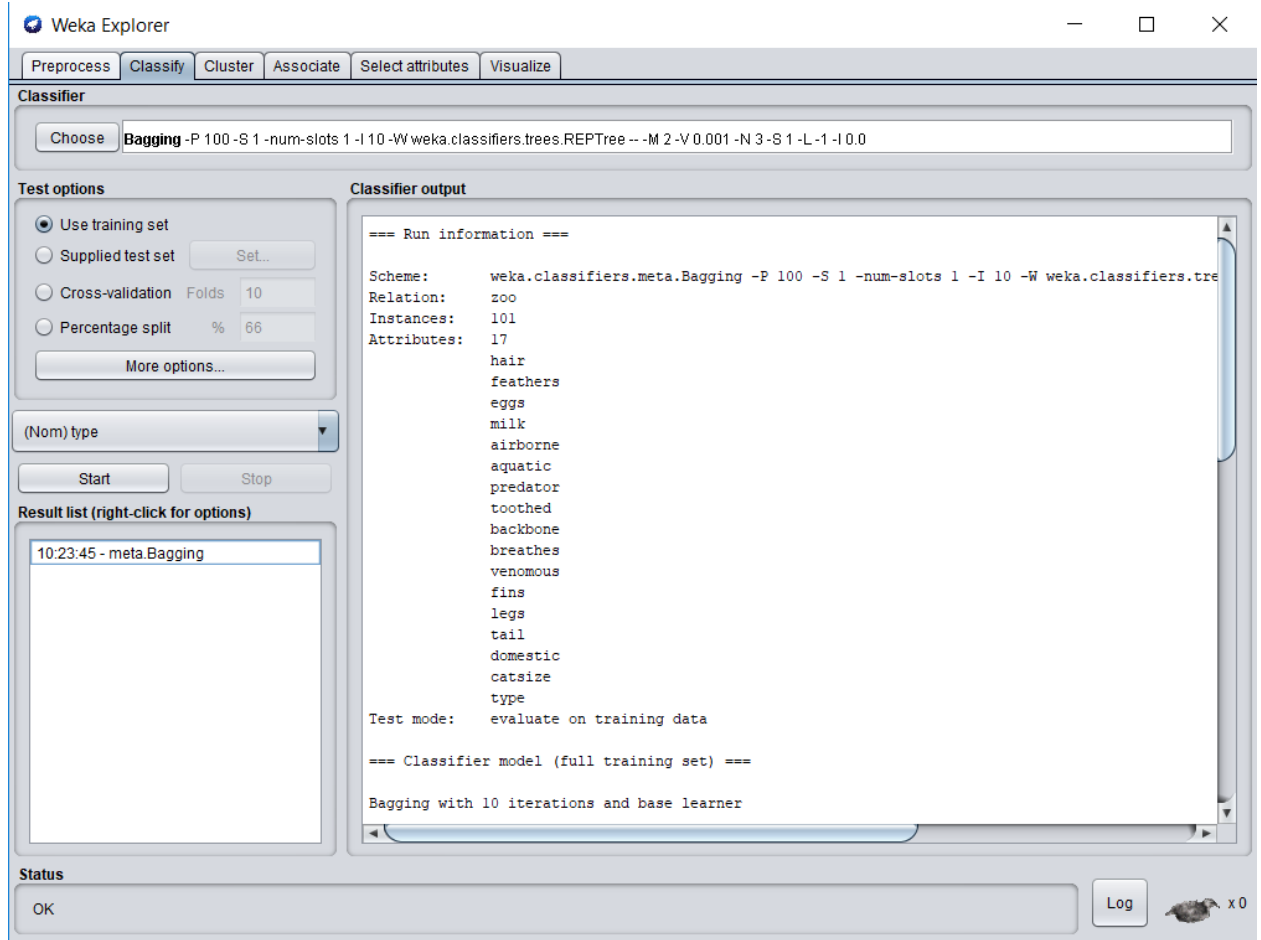
```
=== Confusion Matrix ===
```

	a	b	c	d	e	f	g	<-- classified as
a	41	0	0	0	0	0	0	a = Mammal
b	0	20	0	0	0	0	0	b = Bird
c	1	3	0	1	0	0	0	c = Reptile
d	0	0	0	13	0	0	0	d = Fish
e	2	1	0	1	0	0	0	e = Amphibian
f	0	0	0	0	0	8	0	f = Insect
g	0	0	0	0	0	0	10	g = Mollusc

Dựa vào confusion matrix ta thấy 9 loài bị phân lớp sai:

- Lớp **Reptile** (5 mẫu) đều phân lớp sai qua **Mammal** (1 mẫu), **Bird** (3 mẫu) và **Fish** (1 mẫu).
- Lớp **Amphibian** (4 mẫu) cũng đều bị phân lớp sai qua **Mammal** (2 mẫu), **Bird** (1 mẫu) và **Fish** (1 mẫu).

➤ Bagging:



Ta có kết quả sau khi chạy như sau:

=== Summary ===

Correctly Classified Instances	98	97.0297 %
Incorrectly Classified Instances	3	2.9703 %
Kappa statistic	0.9609	
Mean absolute error	0.0355	
Root mean squared error	0.1025	
Relative absolute error	16.2088 %	
Root relative squared error	31.1175 %	
Total Number of Instances	101	

Phân lớp 101 mẫu với thuật toán **Bagging** (test option: **use training set**):

98 mẫu được phân lớp đúng, 3 mẫu bị phân lớp sai.

=> độ chính xác của thuộc tính trong trường hợp này là 97,0297 %

=== Confusion Matrix ===

```
  a  b  c  d  e  f  g  <-- classified as
41  0  0  0  0  0  0 | a = Mammal
 0 20  0  0  0  0  0 | b = Bird
 0  0  5  0  0  0  0 | c = Reptile
 0  0  0 13  0  0  0 | d = Fish
 0  0  0  0  4  0  0 | e = Amphibian
 0  0  0  0  0  8  0 | f = Insect
 0  0  0  0  1  2  7 | g = Mollusc
```

Dựa vào **Confusion Matrix** có 3 mẫu bị phân lớp sai như sau:

- 1 mẫu **Moollusc (g)** bị phân thành lớp **Amphibian (e)**.
- 2 mẫu **Moollusc (g)** bị phân thành lớp **Insect (f)**.

IV. Tham khảo.

1. <https://tailieu.vn/doc/khai-pha-du-lieu-gioi-thieu-ve-cong-cu-weka-1254352.html>
2. [https://vi.wikipedia.org/wiki/Weka_\(h%E1%BB%8Dc_m%C3%A1y\)](https://vi.wikipedia.org/wiki/Weka_(h%E1%BB%8Dc_m%C3%A1y))
3. <https://vi.scribd.com/presentation/214406821/Introduction-to-Weka>