Lời nói đầu

Sự phát triển của công nghệ thông tin và việc ứng dụng công nghệ thông tin trong nhiều lĩnh vực của đời sống, kinh tế xã hội trong nhiều năm qua cũng đồng nghĩa với lượng dữ liệu đã được các cơ quan thu thập và lưu trữ ngày một tích luỹ nhiều lên. Họ lưu trữ các dữ liệu này vì cho rằng trong nó ẩn chứa những giá trị nhất định nào đó. Mặt khác, trong môi trường cạnh tranh, người ta ngày càng cần có nhiều thông tin với tốc độ nhanh để trợ giúp việc ra quyết định và ngày càng có nhiều câu hỏi mang tính chất định tính cần phải trả lời dựa trên một khối lượng dữ liệu khổng lồ đã có. Với những lý do như vậy, các phương pháp quản trị và khai thác cơ sở dữ liệu truyền thống ngày càng không đáp ứng được thực tế đã làm phát triển một khuynh hướng kỹ thuật mới đó là Kỹ thuật phát hiện tri thức và khai phá dữ liệu.

Khai phá dữ liệu đã và đang được nghiên cứu, ứng dụng trong nhiều lĩnh vực khác nhau ở các nước trên thế giới, tại Việt Nam kỹ thuật này tương đối còn mới mẻ tuy nhiên cũng đang được nghiên cứu và dần đưa vào ứng dụng. Khai phá dữ liệu là một bước trong qui trình phát hiện tri thức gồm có các thuật toán khai thác dữ liệu chuyên dùng dưới một số qui định về hiệu quả tính toán chấp nhận được để tìm ra các mẫu hoặc các mô hình trong dữ liệu. Nói một cách khác, mục đích của phát hiện tri thức và khai phá dữ liệu chính là tìm ra các mẫu và/hoặc các mô hình đang tồn tại trong các cơ sở dữ liệu nhưng vẫn còn bị che khuất bởi hàng núi dữ liệu.

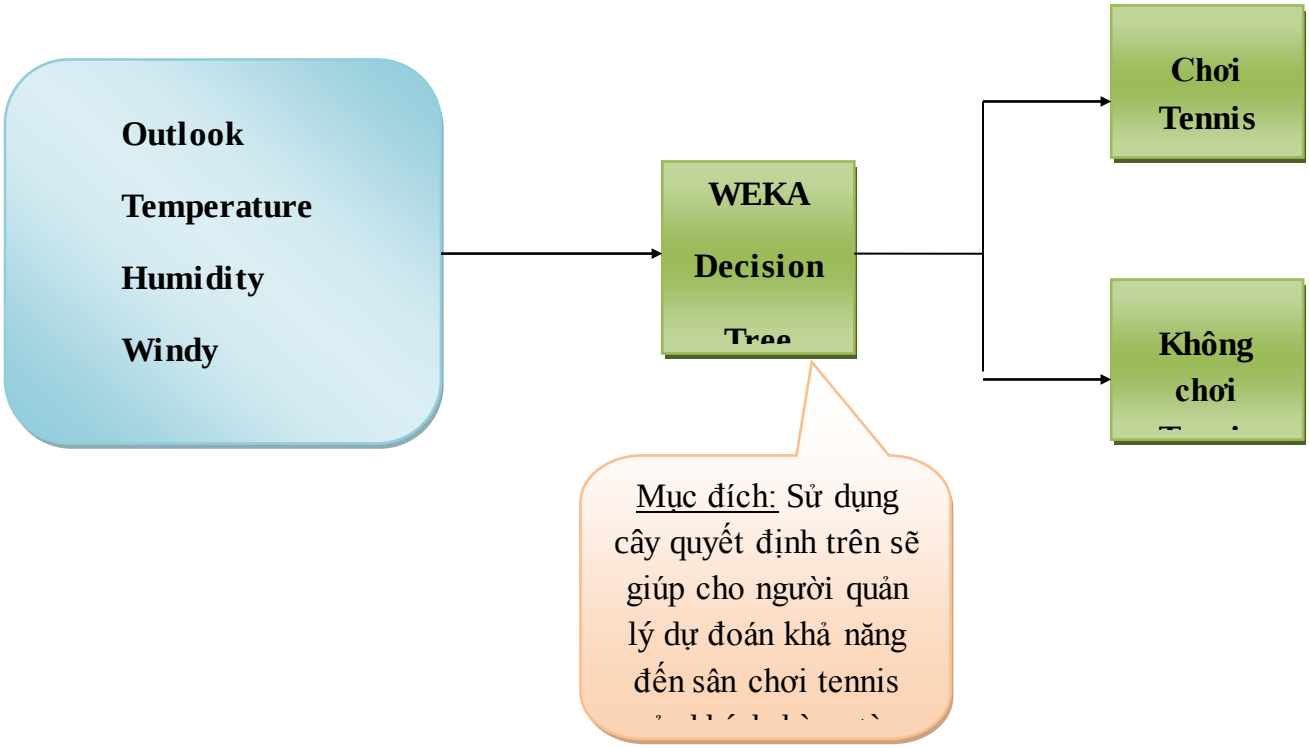
Trong bài viết này, em sẽ trình bày một cách tổng quan về Kỹ thuật khai phá dữ liệu. Trên cơ sở đó đưa ra một bài toán dự báo về khả năng chơi thể thao dựa vào thời tiết và giải quyết bài toán bằng phương pháp phân lớp nhằm cung cấp cho bạn đọc một cách nhìn khái quát về kỹ thuật mới này cũng như mối tương quan với phương pháp thống kê truyền thống.

1. Mô tả bài toán

Trong bài tập này, em sẽ xây dựng mô hình phân lớp (Classification Model) bằng cây quyết định trong weka. Dữ liệu được dùng trong ví dụ này là file **weather.arff** có 4 thuộc tính **Outlook, Temperature, Humidity, Windy** và thuộc tính phân loại là **Play** có 2 giá trị là Yes và No. Đây là dữ liệu mô tả về khả năng có đến sân để chơi thể thao (tennis chẳng hạn) hay không của những người chơi thể thao phụ thuộc vào thời tiết.

Vấn đề đặt ra là với thời tiết như vậy, người chơi tennis có đến sân để chơi hay không ? Việc dự đoán này sẽ giúp cho người quản lý sân giảm được rất nhiều chi phí quản lý sân như điều chỉnh (tăng hoặc giảm) số nhân viên phục vụ cũng như các dịch vụ khác nhằm phục vụ tốt hơn nhu cầu của người chơi tennis.

Bằng cách sử dụng WEKA để thực thi một cây quyết định ,chúng ta có thể xây dựng 1 công cụ hỗ trợ được các yêu cầu trên dựa vào những dữ liệu khách quan thu thập trước đó.



2. 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 nomial 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 (như luật kết hợp chẳn hạn) thì bạn phải thực hiện việc rời rạc hóa dữ liệu.

2.1. Trích chọn thuộc tính

Trích chọn thuộc tính là nhiệm vụ rất quan trọng giai đoạn tiền xử lý dữ liệu khi triển khai các mô hình khai phá dữ liệu. Một vấn đề gặp phải là các dataset dùng để xây dựng các Data mining Models thường chứa nhiều thông tin không cần thiết cho việc xây dựng mô hình. Chẳng hạn, một dataset gồm hàng trăm thuộc tính dùng để mô tả về khách hàng của một doanh nghiệp được thu thập, tuy nhiên khi xây dựng một Data mining model nào đó chỉ cần khoảng 50 thuộc tính từ hàng trăm thuộc tính đó. Nếu ta sử dụng tất cả các thuộc tính (hàng trăm) của khách hàng để xây dựng mô hình thì ta cần nhiều CPU, nhiều bộ nhớ trong quá trình Training model, thậm chí các thuộc tính không cần thiết đó làm giảm độ chính xác của mô hình và gây khó khăn trong việc phát hiện tri thức.

Các phương pháp trích chọn thuộc tính thường tính trọng số (score) của các thuộc tính và sau đó chỉ chọn các thuộc tính có trọng số tốt nhất để sử dụng cho mô hình. Các phương pháp này cho phép bạn hiệu chỉnh ngưỡng (threshold) để lấy ra các thuộc tính có Score trên ngưỡng cho phép. Quá trình trích chọn thuộc tính luôn được thực hiện trước quá trình Training Model.

❖ *Một số phương pháp chọn thuộc tính (Feature Selection Methods)*

Có rất nhiều phương pháp để lựa chọn thuộc tính tùy thuộc vào cấu trúc của dữ liệu dùng cho mô hình và thuật toán được dùng để xây dựng mô hình. Sau đây là một số phương pháp phổ biến dùng trong trích chọn thuộc tính:

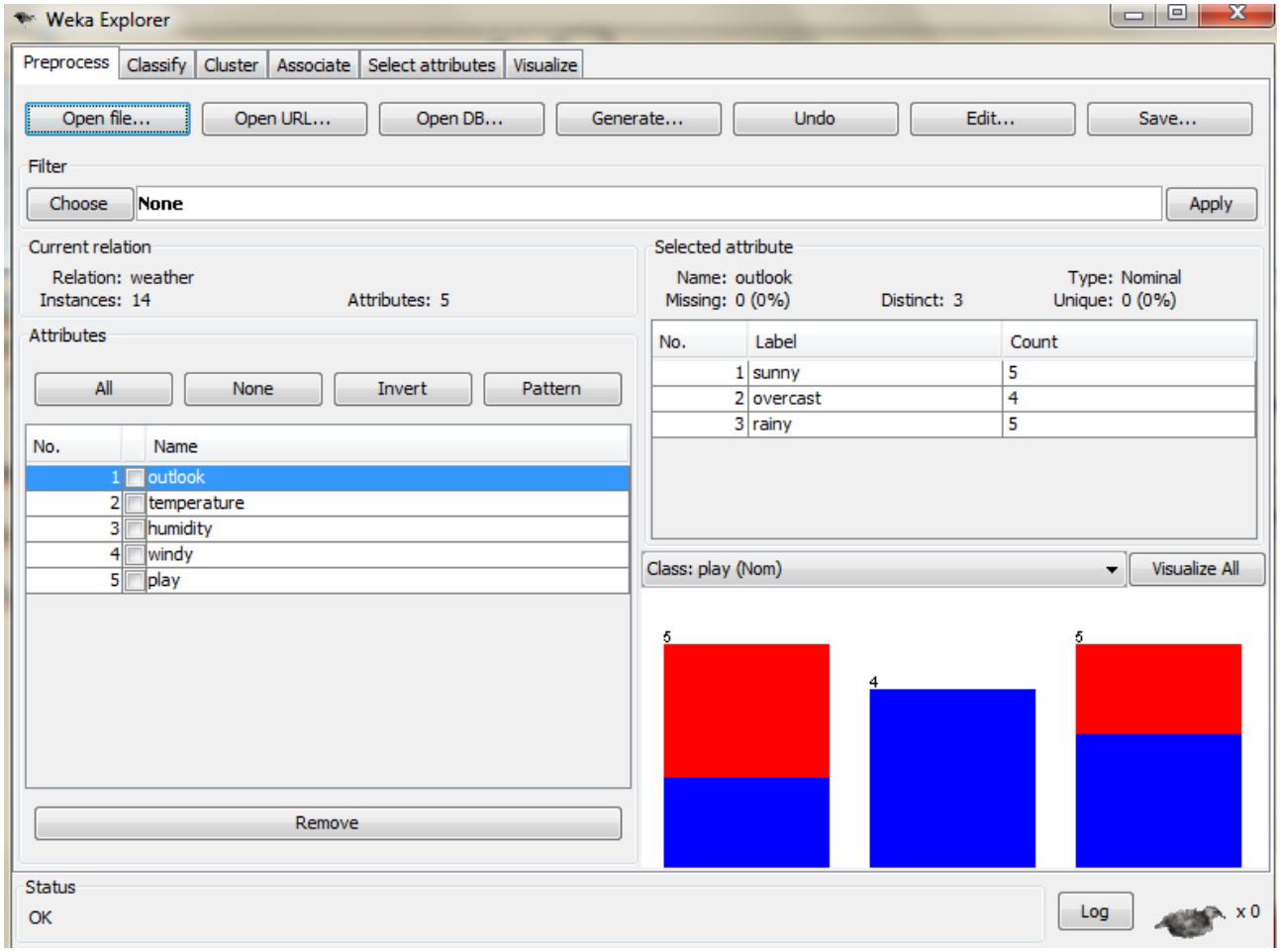
- *Interestingness score*: Được sử dụng để xếp hạng (rank) các thuộc tính đối với các thuộc tính có kiểu dữ liệu liên tục (continuous).

- *Shannon's Entropy*: Được sử dụng đối với các dữ liệu kiểu rời rạc (discretized data).
- Ngoài ra còn có một số phương pháp khác cũng thường được sử dụng trong lựa chọn thuộc tính như *Bayesian with K2 Prior*, *Bayesian Dirichlet Equivalent with Uniform Prior*.

❖ *Trích chọn thuộc tính với phần mềm WeKa*

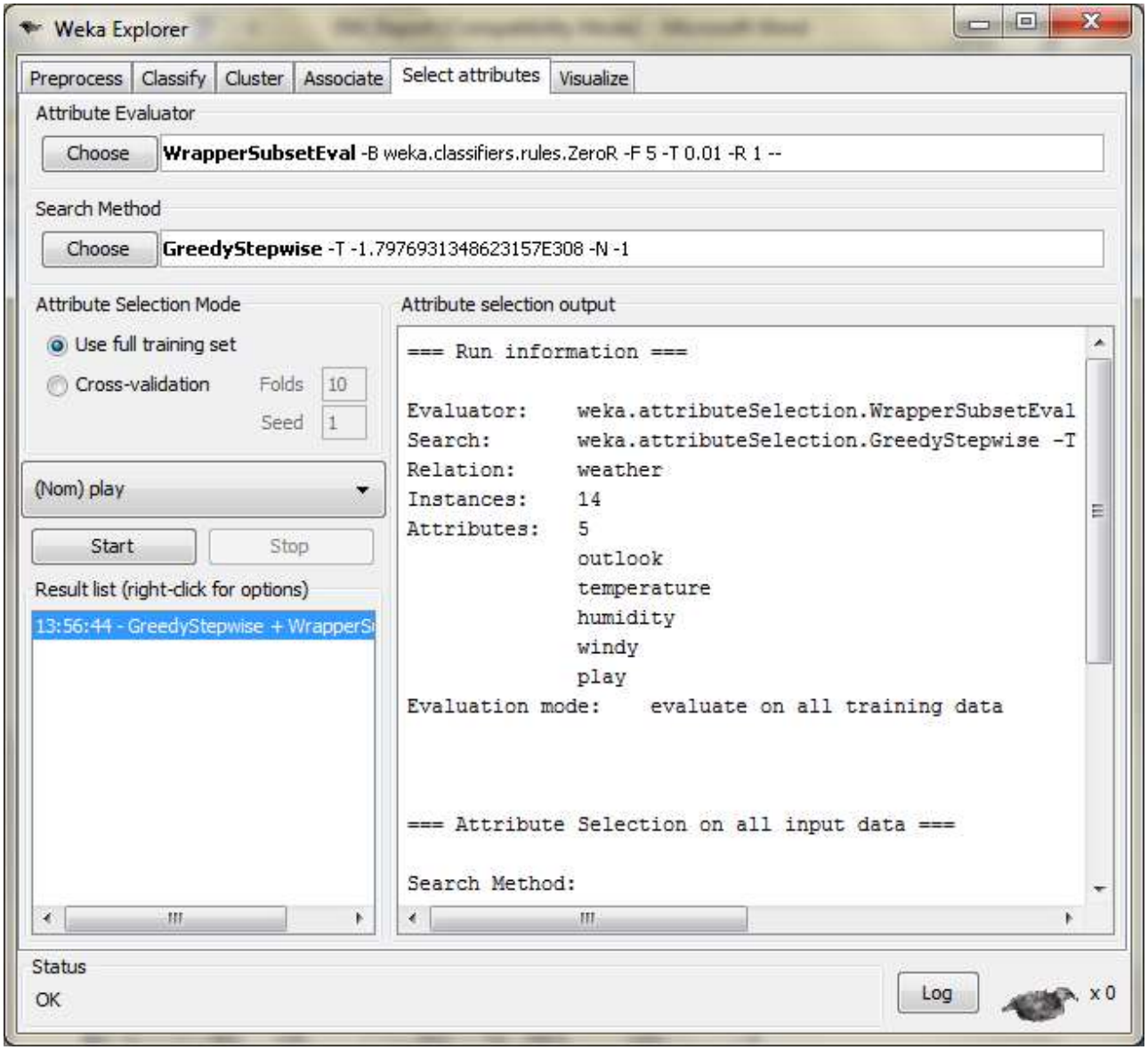
Dataset được dùng là file định dạng chuẩn của weka Weather.arff

Khởi động Weka > Chọn Explorer > Chọn Open file > Chọn Dataset "Weather.arff" kết quả như sau:



- Chọn Tab "Select attributes".
- Trong mục Attribute Evaluator chọn WrapperSubsetEval.
- Trong mục Search Method chọn GreedyStepwise .
- Chon Tab "Classify": Trong mục classifier chọn NaiveBayes

- *Bấm **Start** để thực hiện, kết quả như sau:*



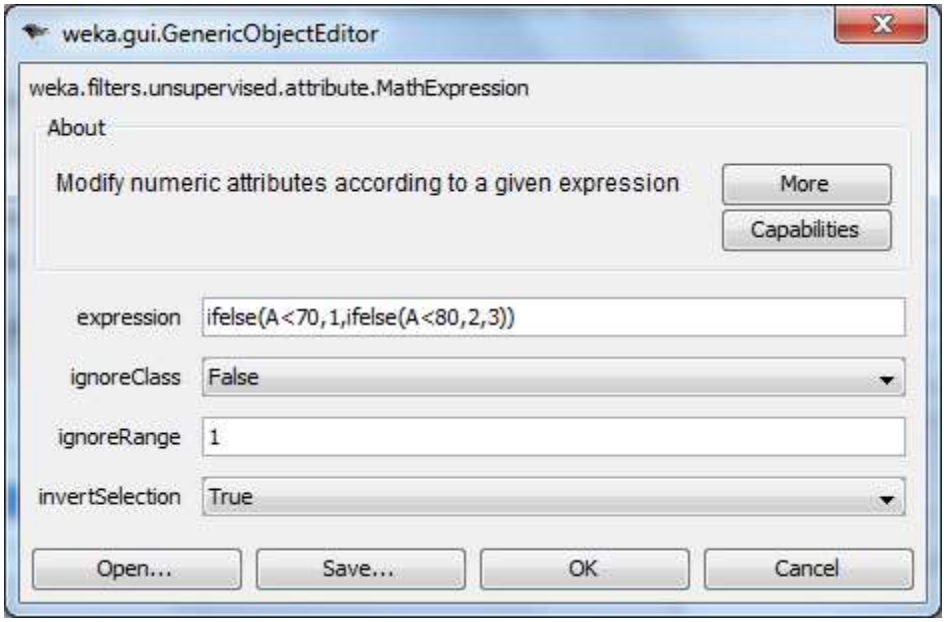
Vậy 5 thuộc tính được chọn đó là: *outlook, temperature, humidity, windy* và *play* (quang cảnh, nhiệt độ, độ ẩm, gió, chơi).

2.2. Chuẩn hóa dữ liệu

- Chuyển kiểu dữ liệu của thuộc tính ***temperature*** thành kiểu Nominal với các giá trị tương ứng cool(temperature ≤ 70.0), mild (70.0 < temperature <80.0), hot (80.0≤ temperature).

+ Trong Filter chọn Choose trong cây thư mục chọn MathExpression.

+ Nhập biểu thức lọc vào nhấn OK để chấp nhận.Chọn Apply để áp dụng lên trường dữ liệu temperature.



+ Chọn tiếp mục NumericToNominal trong cây thư mục. Chọn Apply

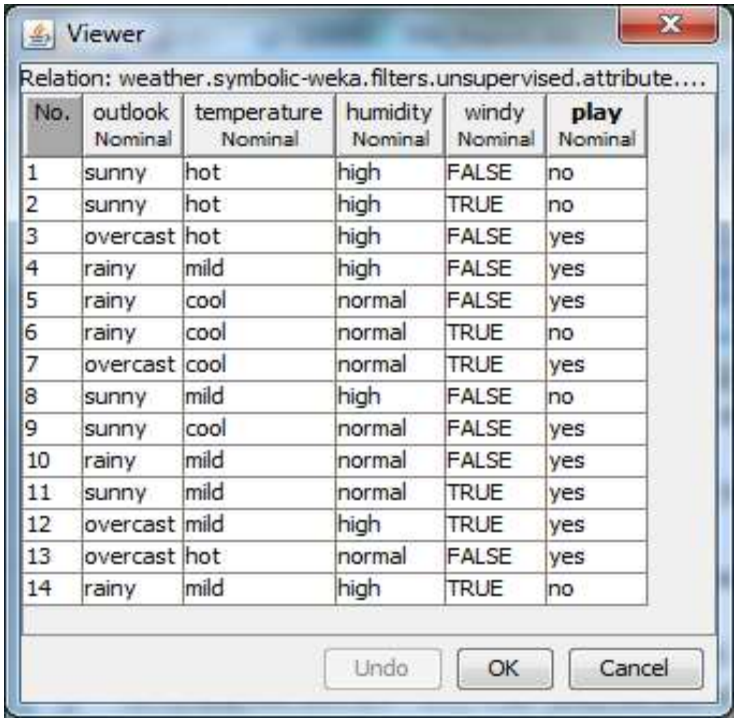
+ Thêm các gía trị *cool, mild, hot* vào thuộc tính temperature. Trong cây thư mục chọn AddValue. Tiến hành nhập vào các giá trị tương ứng như sau:

+ Đưa dữ liệu vào bảng ta chọ Edit trong vùng Filter xuất hiện bảng dữ liệu

+ Click chuột phải vào cột temperature chọn Replace Value With… gõ các giá trị tương ứng muốn thay thế vào.

- Tương tự chuyển kiểu dữ liệu của thuộc tính ***humidity*** thành kiểu Nominal với các giá trị tương ứng normal (humidity ≤ 80.0), hight (80.0 < humidity)

Sau khi tiến hành quá trình Preprocess ta thu được bảng dữ liệu chỉ toàn kiểu Nominal như sau:



No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	play Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

3. Chọn thuật toán J48 để xây dựng cây quyết định

✓ Giới thiệu về cây quyết định

Cây quyết định (Decision Tree) là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật (series of rules). Các thuộc tính của đối tượng (ngoại trừ thuộc tính phân lớp – Category attribute) có thể thuộc các kiểu dữ liệu khác nhau (Binary, Nominal, ordinal, quantitative values) trong khi đó thuộc tính phân lớp phải có kiểu dữ liệu là Binary hoặc Ordinal.Tóm lại, cho dữ liệu về các đối tượng gồm các thuộc tính cùng với lớp (classes) của nó, cây quyết định sẽ sinh ra các luật để dự đoán lớp của các đối tượng chưa biết (unseen data).

✓ Qui trình Train và Test một classifier

- Dữ liệu để xây dựng mô hình: dữ liệu gốc (original dataset), dữ liệu này phải có thuộc tính phân lớp gọi là categorical attribute

- Dữ liệu gốc sẽ được chia thành 2 phần là Training Set (để xây dựng model) và Testing Set (để kiểm định Model)

- Cuối cùng là tính toán lỗi để đánh giá Model

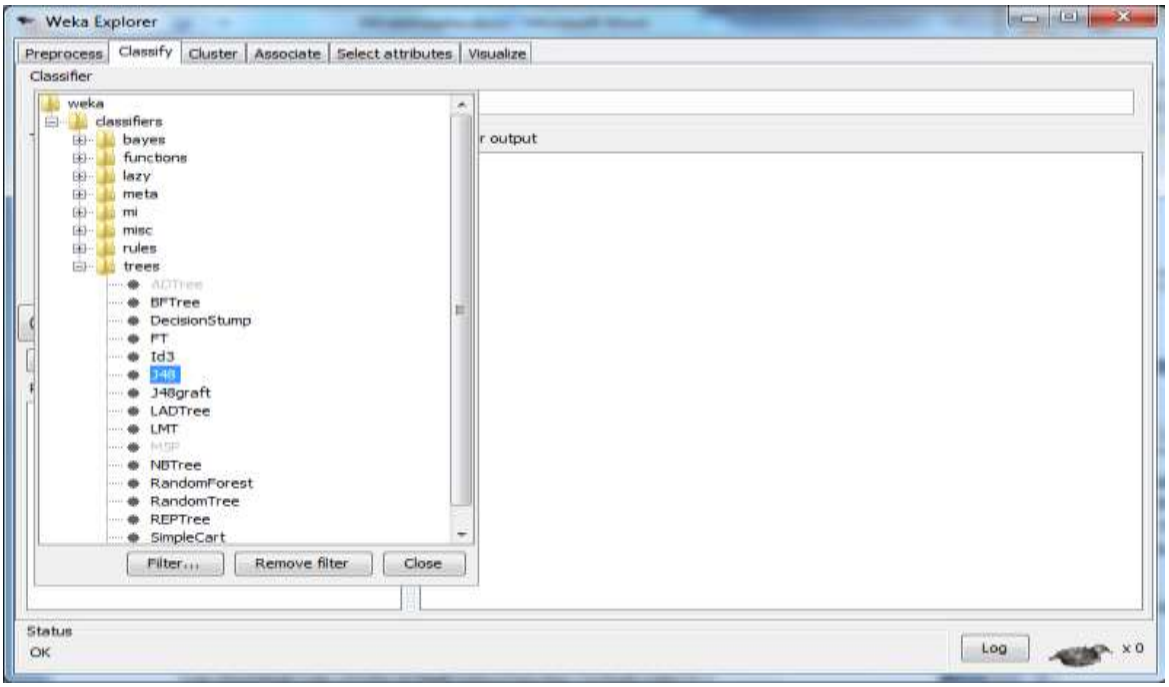
✓ Tại sao lại sử dụng thuật toán này

Có rất nhiều thuật toán phân lớp như ID3, J48, C4.5, CART (Classification and Regression Tree),… Việc chọn thuật toán nào để có hiệu quả phân lớp cao tuy thuộc vào rất nhiều yếu tố, trong đó cấu trúc dữ liệu ảnh hưởng rất lớn đến kết quả của các thuật toán.

Với thuật toán ID3 và CART cho hiệu quả phân lớp rất cao đối với các trường dữ liệu số (quantitative value) trong khi đó các thuật toán như J48, C4.5 có hiệu quả hơn đối với các dữ liệu Qualititive value (ordinal, Binary, nominal).Sau khi đã chuẩn hóa dữ liệu thì được bảng dữ liệu chỉ toàn kiểu Nominal, vì vậy ta sử dụng thuật toán J48 để đạt hiệu quả phân lớp cao.

✓ Sử dụng thuật toán với phần mềm Weka

Nhấn vào tab Classify chọn thuật toán sử dụng bằng cách nhấn vào nút Choose; khi cây thư mục hiện thư mục Trees/J48:



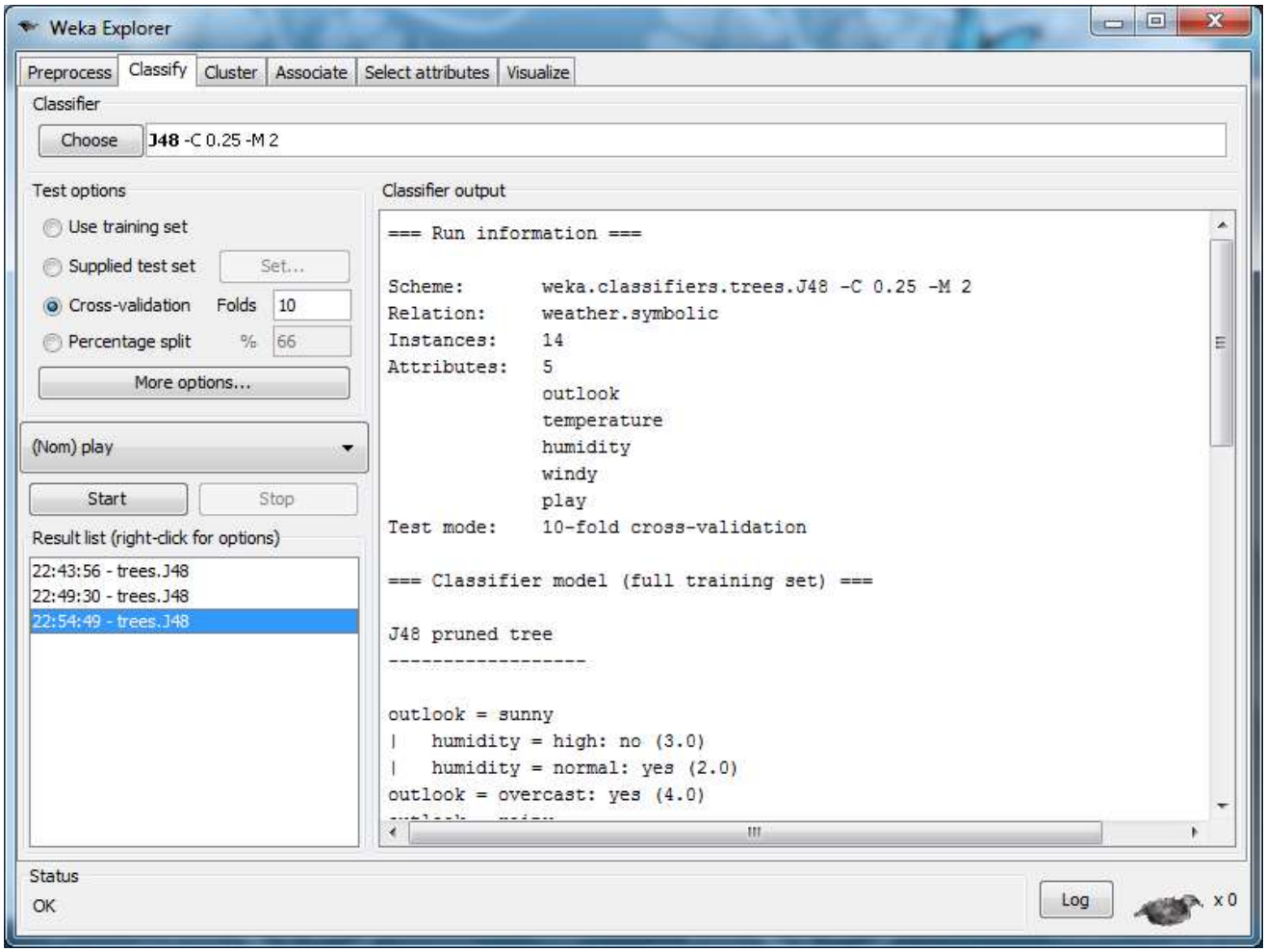
Đánh giá hiệu quả phân lớp của thuật toán đối với tập dữ liệu được cho theo hai phương pháp:

a. 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*.

Đây là kỹ thuật chủ yếu được sử dụng trong xây dựng predictive Model. Trong đó dữ liệu gốc sẽ được chia thành n phần bằng nhau (n-fold), và quá trình Train/Test Model thực hiện lặp lại n lần. Tại mỗi lần Train/Test Model, 1 phần dữ liệu dùng để Test và (n-1) phần còn lại dùng để Train.(Người ta đã chứng minh 10-fold Cross –Validation là tối ưu)

Với phương pháp này ta thu được kết quả hiển thị ở khung **Classifier Output** như sau:



Nội dung kết quả :

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     weather.symbolic
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    10-fold cross-validation

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

J48 pruned tree
------------------

outlook = sunny
|   humidity = high: no (3.0)
|   humidity = normal: yes (2.0)
outlook = overcast: yes (4.0)
outlook = rainy
|   windy = TRUE: no (2.0)
|   windy = FALSE: yes (3.0)

Number of Leaves  :         5

Size of the tree :        8


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
```

```
=== Summary ===

Correctly Classified Instances          7               50      %
Incorrectly Classified Instances        7               50      %
Kappa statistic                          -0.0426
Mean absolute error                      0.4167
Root mean squared error                  0.5984
Relative absolute error                 87.5    %
Root relative squared error            121.2987 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.556    0.6        0.625     0.556    0.588      0.633    yes
                 0.4      0.444      0.333     0.4      0.364      0.633    no
Weighted Avg.    0.5      0.544      0.521     0.5      0.508      0.633

=== Confusion Matrix ===

 a b   <-- classified as
 5 4 | a = yes
 3 2 | b = no
```

Đọc nội dung kết quả: Kết quả được trả về theo 3 vùng dữ liệu:

- ***Vùng Run Information:*** Cho biết thông tin về dữ liệu nguồn.
  - Đề án sử dụng: weka.classifiers.trees.J48 -C 0.25 -M 2
  - Cơ sở dữ liệu: weather.symbolic
  - Số trường: 14
  - Số thuộc tính: 5
    - outlook
    - temperature
    - humidity
    - windy
    - play

- Chế độ kiểm tra: 10-fold cross-validation

➢ ***Vùng hiển thị kết quả trainning:***

- Chế độ phân lớp: toàn bộ dữ liệu
- Cây J48 sau khi tiến hành trainning:

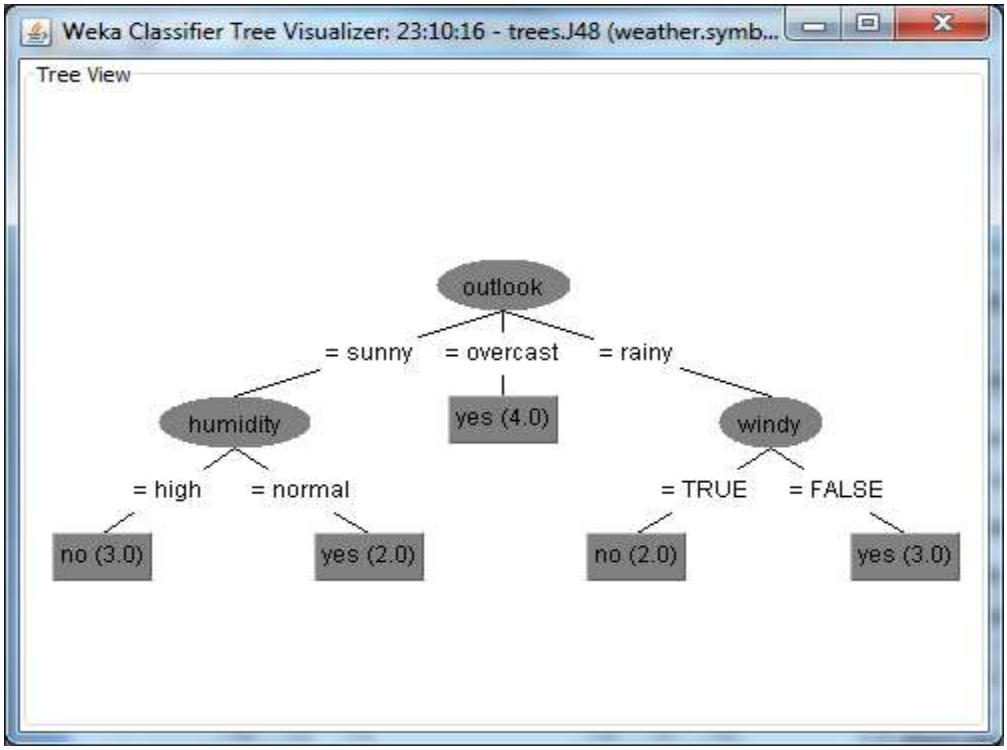
outlook = sunny
| humidity = high: no (3.0)
| humidity = normal: yes (2.0)
outlook = overcast: yes (4.0)
outlook = rainy
| windy = TRUE: no (2.0)
| windy = FALSE: yes (3.0)

- Số lượng lá: 5
- Kích thước cây: 8
- Thời gian tiến hành: 0 giây

➢ ***Tóm tắt kết quả xác nhận phân lớp:***

- Trường hợp phân lớp chính xác: 7 chiếm 50%
- Trường hợp không chính xác: 7 chiếm 50%
- Các thống kê lỗi.

Kết quả hiển thị cây như sau:



Các luật được sinh ra:

Rule 1: If outlook = "sunny" and humidity = hight then Play = "no"

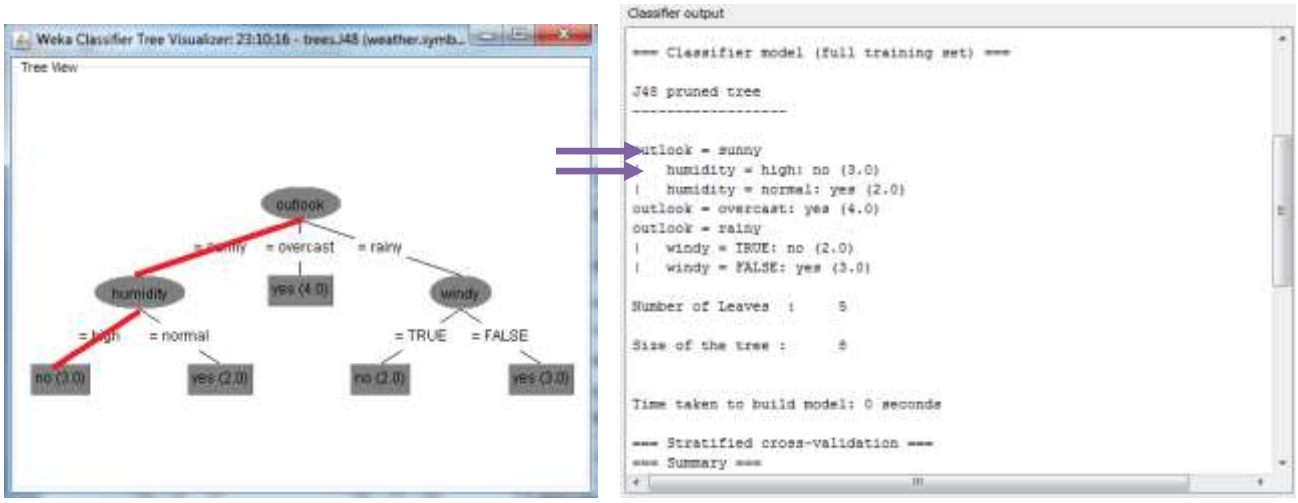
Rule 2: If outlook = "sunny" and humidity = normal then Play = "yes"

Rule 3: If outlook = "overcast" then Play= "yes"

Rule 4: If outlook = "rainy" and windy = TRUE then Play ="no"

Rule 5: If outlook = "rainy" and windy = FALSE then Play ="yes"

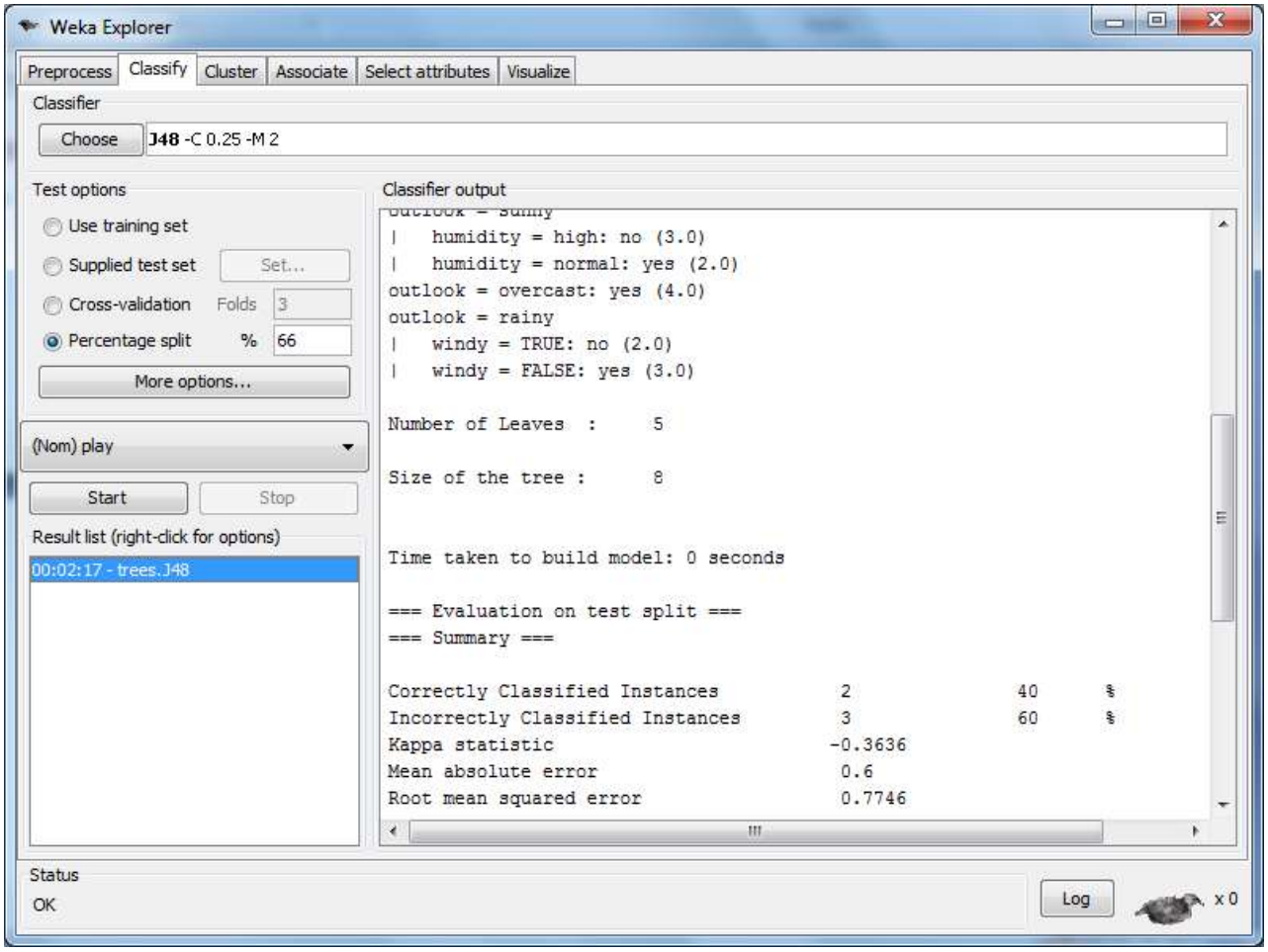
Đọc cây quyết định:



Tương tự với các trường hợp còn lại.

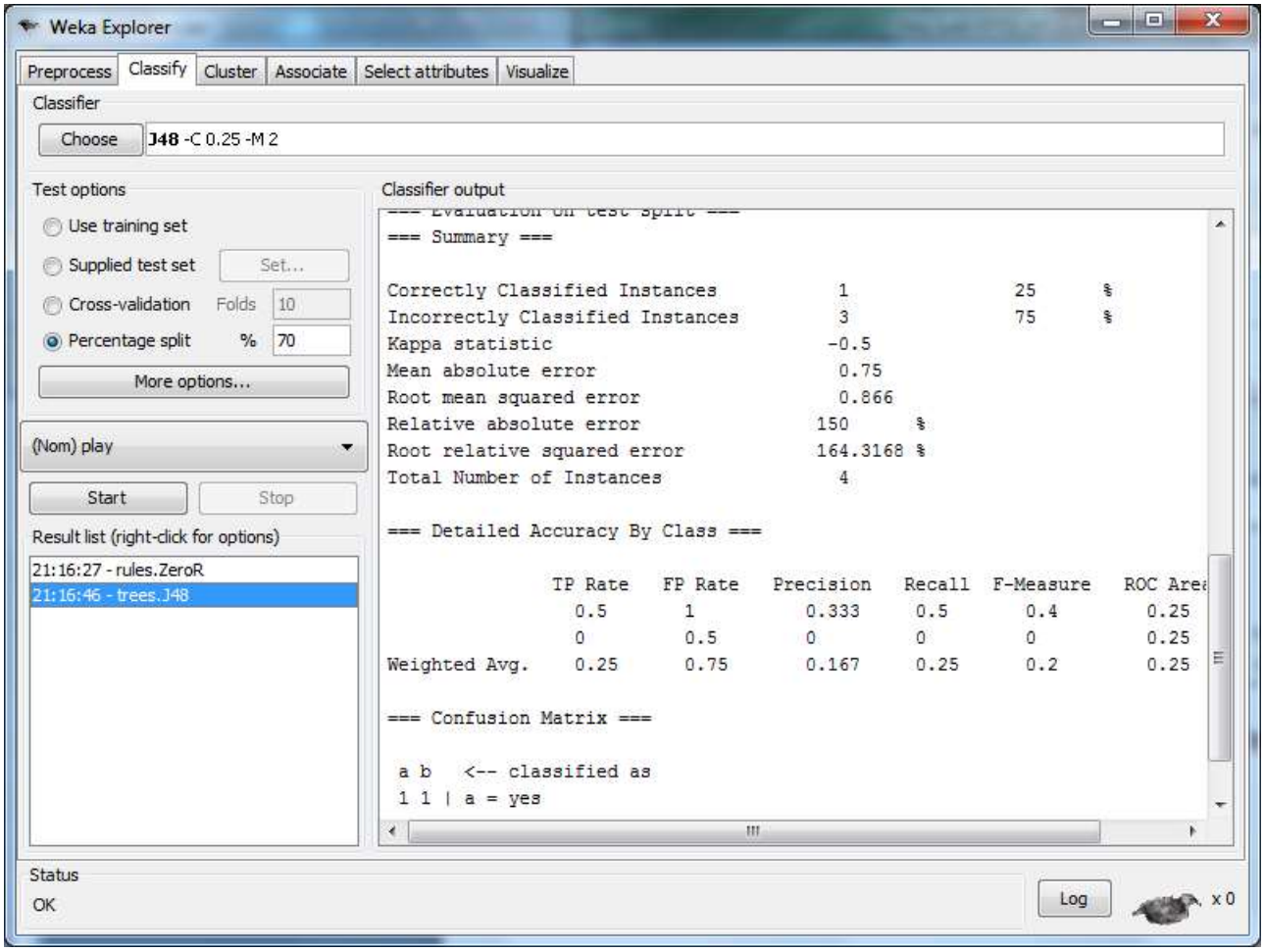
b. ***Phương pháp Percentage split:*** Cho biết tỉ lệ phân chia là bao nhiêu % thì đạt hiệu quả phân lớp cao nhất.

- *Lần thứ nhất:* với tỉ lệ phân chia là 66% thì ta có kết quả như sau:



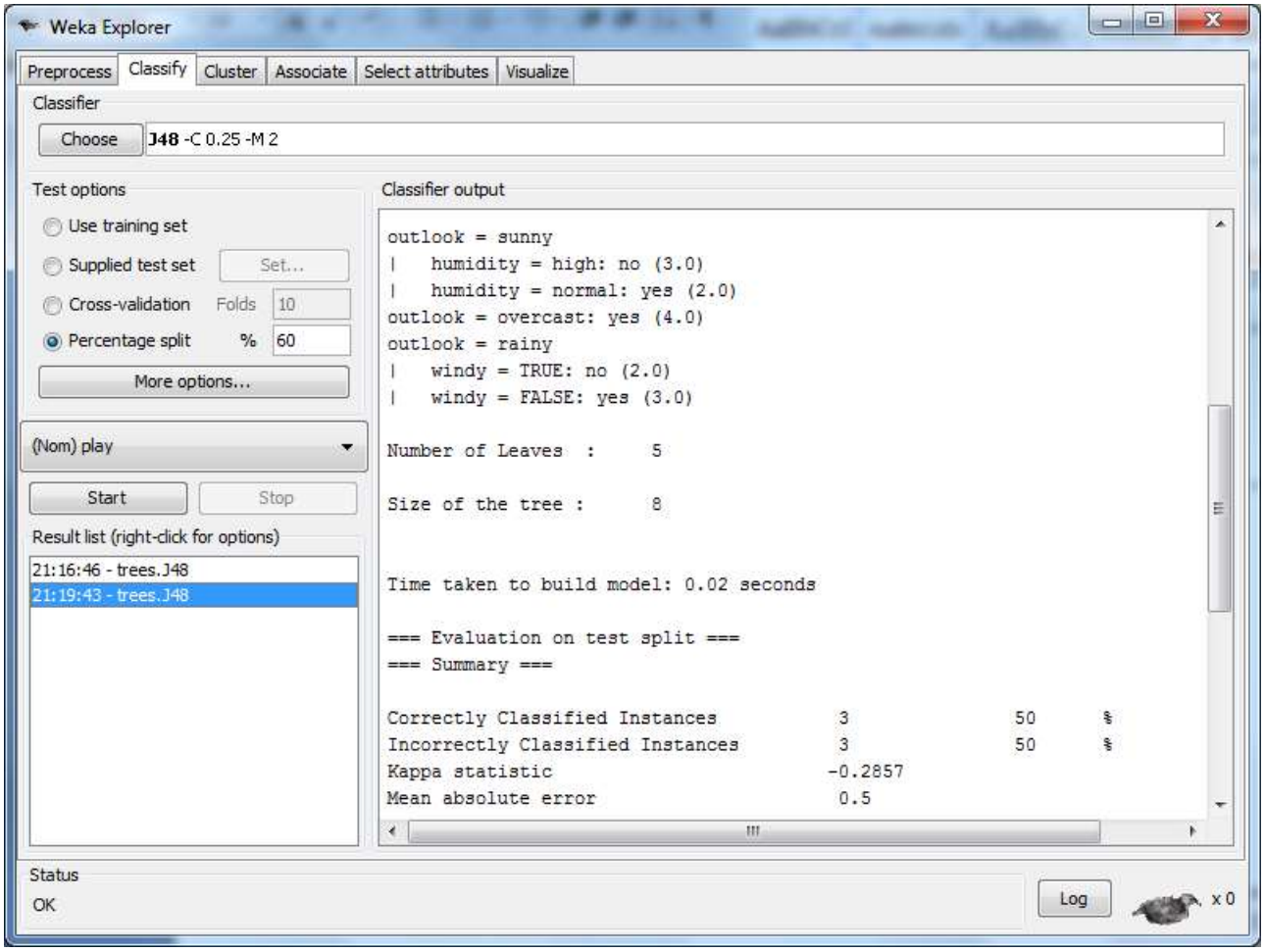
=== Evaluation on test split ===			
=== Summary ===			
Correctly Classified Instances	2	40	%
Incorrectly Classified Instances	3	60	%

- Lần thứ 2: tỉ lệ phân chia là 70% thì kết quả thu được là:



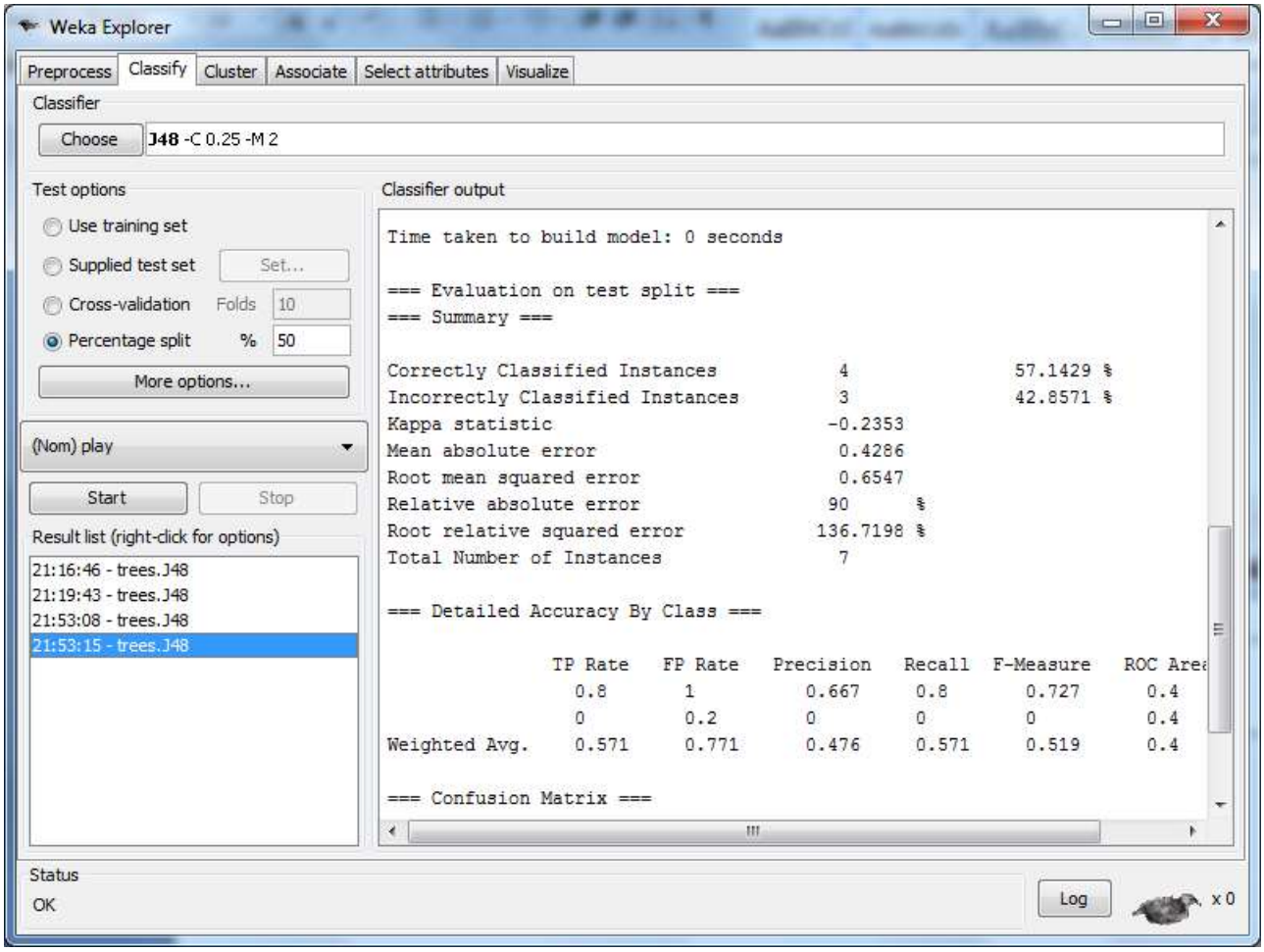

```
=== Evaluation  on test split ===
=== Summary  ===

Correctly Classified Instances         1               25      %
Incorrectly Classified Instances       3               75      %
```

- Lần thứ 3: tỉ lệ phân chia là 60% thì kết quả là:



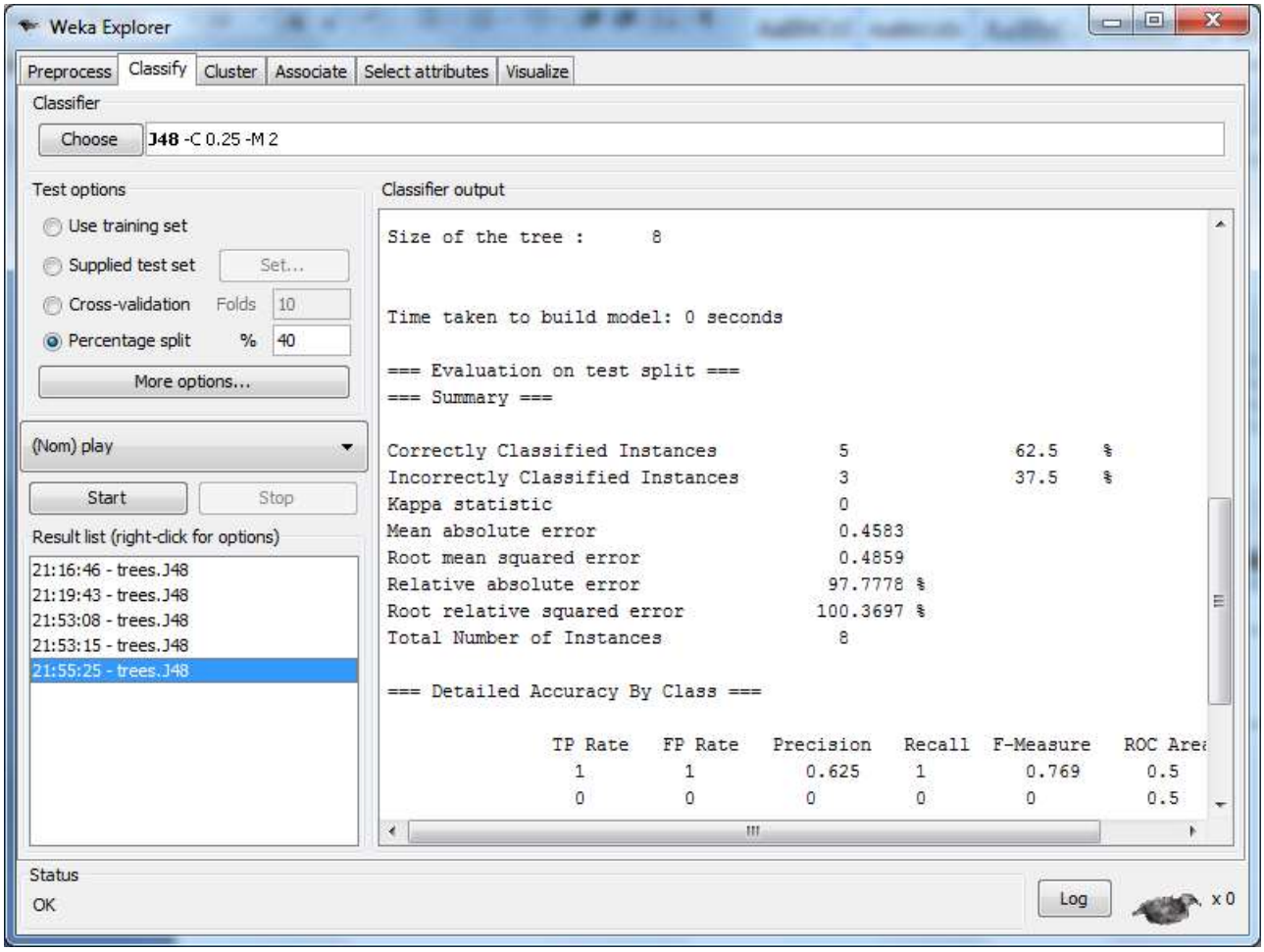
Correctly Classified Instances	3	50	%
Incorrectly Classified Instances	3	50	%

- Lần thứ 4: tỉ lệ phân chia là 50%



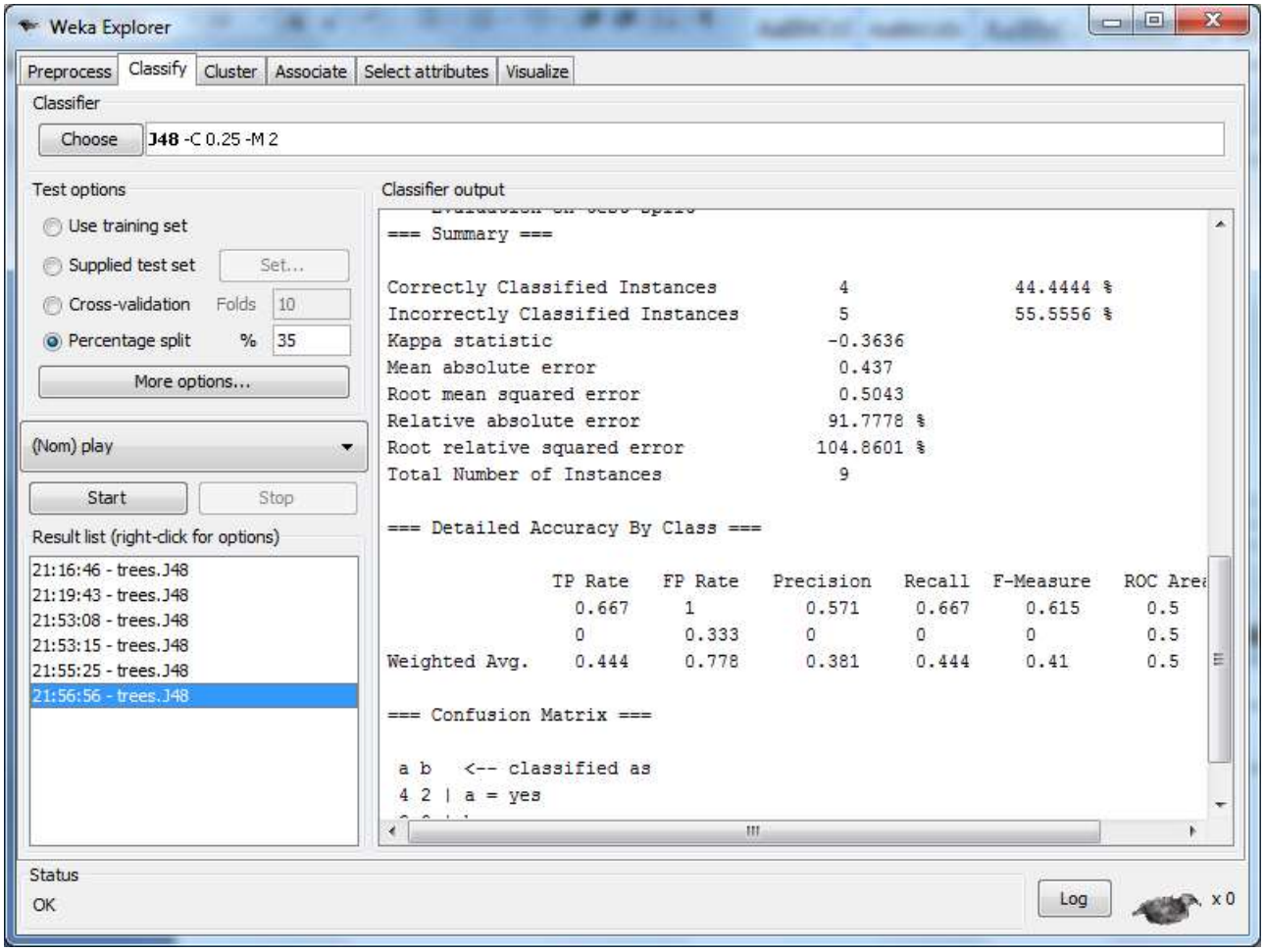
Correctly Classified Instances	4	57.1429 %
Incorrectly Classified Instances	3	42.8571 %

- Lần thứ 5: tỉ lệ phân chia là 40%



Correctly Classified Instances	5	62.5	%
Incorrectly Classified Instances	3	37.5	%

- Lần thứ 6: tỉ lệ phân chia là 35%

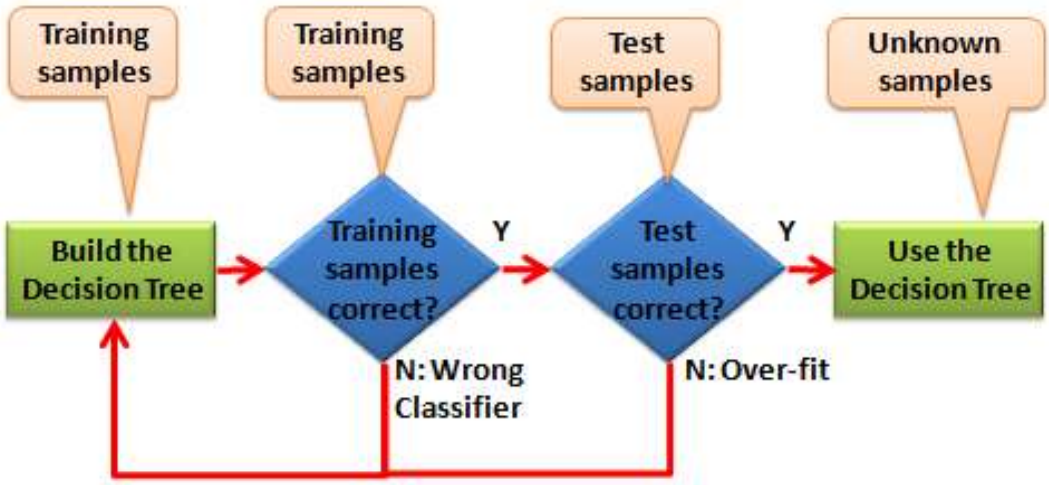


Correctly Classified Instances	4	44.4444 %
Incorrectly Classified Instances	5	55.5556 %

4. Đánh giá và kết luận

Sau khi thực hiện chạy 6 lần J48 để xây dựng cây với các tham số đầu vào khác nhau ta thấy ứng với lần chạy thứ 5 với các tham số lựa chọn là : bộ dữ liệu dùng khởi tạo cây 40 %, bộ dữ liệu test là 60% đạt được tỉ lệ phân lớp chính xác là 62,5%. Số lượng mẫu test không quá bé. Nên ta chọn cây quyết định sinh ra tại lần chạy thứ 5 để sử dụng cho các mẫu thử bất kỳ sau này.

Sau quá trình thực hành tạo cây quyết định trên WEKA bằng thuật toán J48 (C4.5) ta có thể hình dung được quá trình khởi tạo và hoạt động của một cây quyết định như sau:


Training samples
Training samples
Test samples
Unknown samples
Build the Decision Tree
Training samples correct?
Y
Test samples correct?
Y
Use the Decision Tree
N: Wrong Classifier
N: Over-fit