

Trường Đại học Khoa học Tự nhiên  
Khoa Công nghệ Thông tin



TÀI LIỆU LÝ THUYẾT KTDL & UD

# Phân Lớp Dữ Liệu (P1)

**Giảng viên: ThS. Lê Ngọc Thành**  
**Email: [lnthanh@fit.hcmus.edu.vn](mailto:lnthanh@fit.hcmus.edu.vn)**

Summer 2012

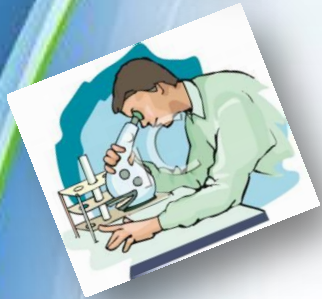
# Nội dung

---

- **Khái niệm cơ sở về phân lớp**
- Phân lớp dựa trên cây quyết định
- Phân lớp dựa trên luật

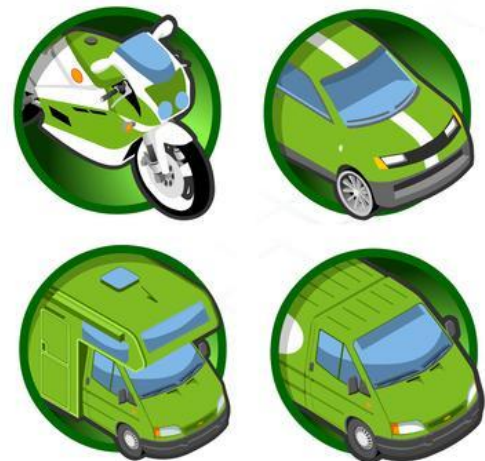
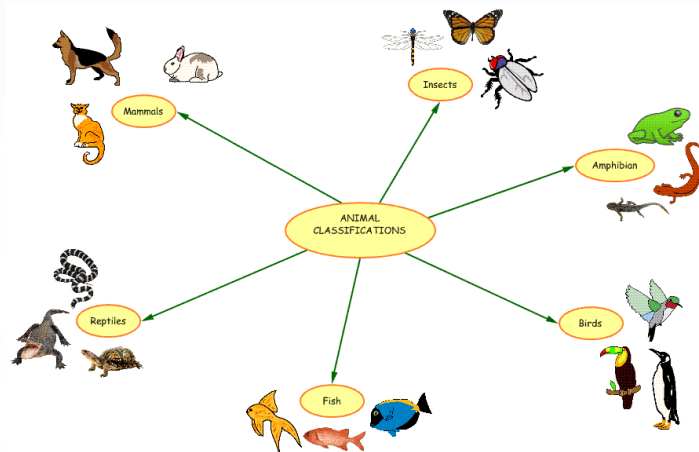
# Case Study

- Ngân hàng đánh giá việc cho vay là “*an toàn*” hay “*rủi ro*”.
- Quản lý cửa hàng dự đoán khách hàng “*mua*” hay “*không mua*”.
- Bác sĩ quyết định “*một trong ba phương pháp*” điều trị nào thích hợp với bệnh nhân.
- Phân loại tin tức thuộc về chủ đề “*thể thao*”, “*chính trị*”, “*văn hóa*” hay “*giải trí*”.



# Phân lớp (1/3)

- Phân lớp là quá trình gán nhãn (đã xác định) cho các mẫu dữ liệu mới với độ chính xác có thể.
  - Ví dụ: gán nhãn “an toàn” hay “rủi ro” cho khách hàng; gán nhãn “mua” hay “không mua”; gán nhãn “pp A”, “pp B” hay “pp C”; gán nhãn “thể thao”,...cho từng tin tức mới.



# Phân lớp (2/3)

- Cho CSDL  $D = \{t_1, t_2, \dots, t_n\}$  và tập các lớp  $C = \{c_1, c_2, \dots, c_m\}$ , phân lớp là bài toán xác định ánh xạ  $f : D \rightarrow C$  sao cho mỗi  $t_i$  được gán vào một lớp  $c_j$ .



*Hãy đưa các ví dụ thể hiện bài toán phân lớp.*

# Phân lớp (3/3)

- Phân lớp là dạng học có giám sát (supervised learning) – Tại sao?
- Phân lớp (classification) và dự đoán giá trị số (numeric prediction) là hai dạng chính của bài toán dự đoán (prediction) nhưng:

Classification	Numeric Prediction
<ul style="list-style-type: none"><li>- Các nhãn là các giá trị rời rạc hay định danh</li><li>- Mục tiêu là phân lớp về các nhãn đã định</li><li>- Ví dụ: dự đoán khách hàng có “mua” hay “không mua” đồ?</li></ul>	<ul style="list-style-type: none"><li>- Đầu ra là hàm giá trị liên tục hay giá trị có thứ tự</li><li>- Mục tiêu là dự đoán các giá trị bị thiếu hay chưa biết</li><li>- Ví dụ: dự đoán số tiền một khách hàng xác định sẽ bỏ ra trong một lần mua sắm</li></ul>

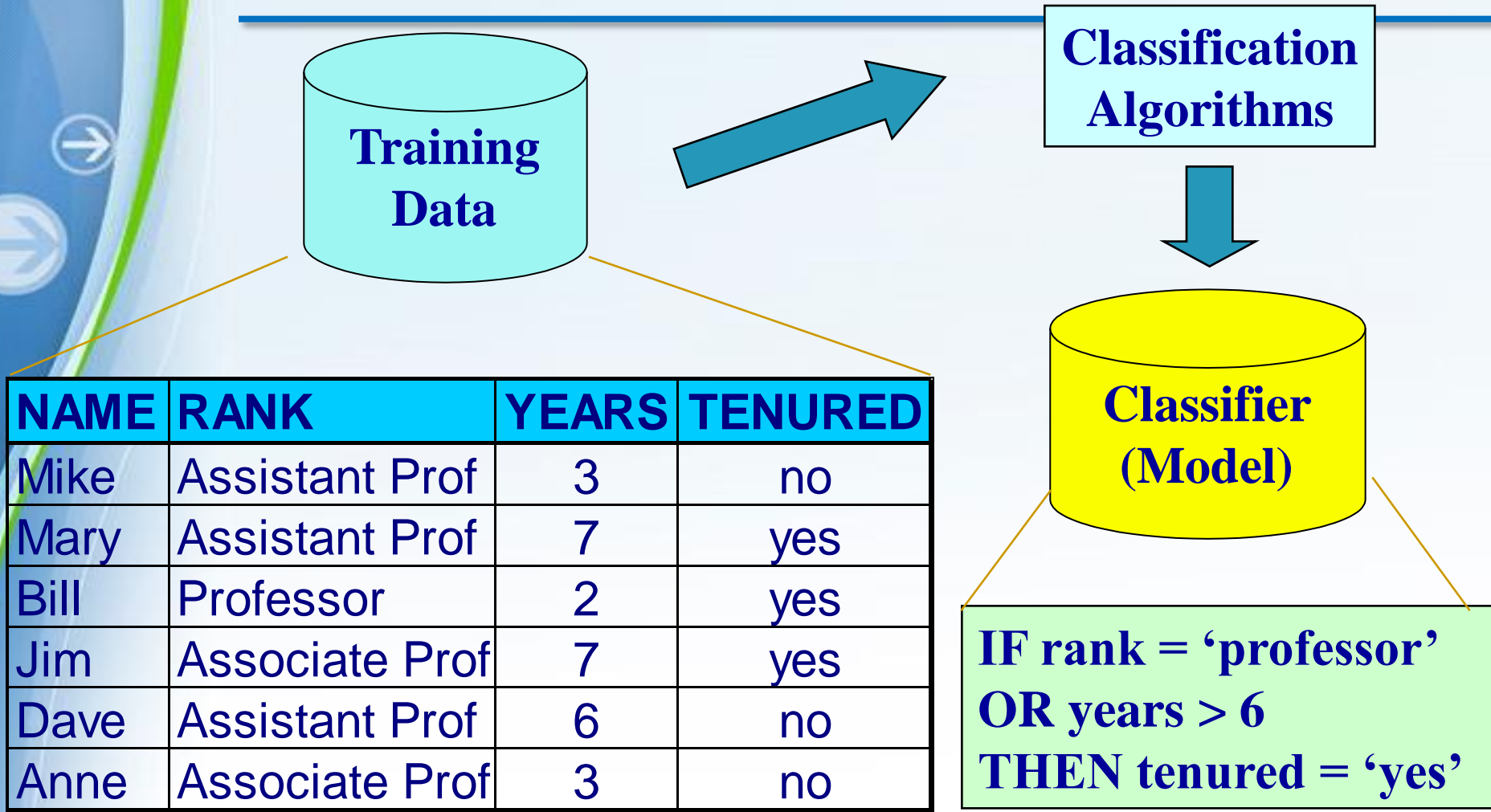




# Quá trình phân lớp

- **Bước 1: *Xây dựng mô hình (bước học)***
  - Mô tả tập các nhãn/lớp
  - Tập huấn luyện: các mẫu đã gán nhãn lớp
  - Đầu ra: mô hình phân lớp ví dụ như luật phân lớp, cây quyết định hoặc công thức toán mô tả lớp
- **Bước 2: *Sử dụng mô hình (b.phân lớp)***
  - Áp dụng mô hình vào dữ liệu kiểm thử (tách biệt và đã có nhãn) để đánh giá độ chính xác.
  - Nếu độ chính xác chấp nhận được -> áp dụng mô hình để phân lớp các mẫu mới

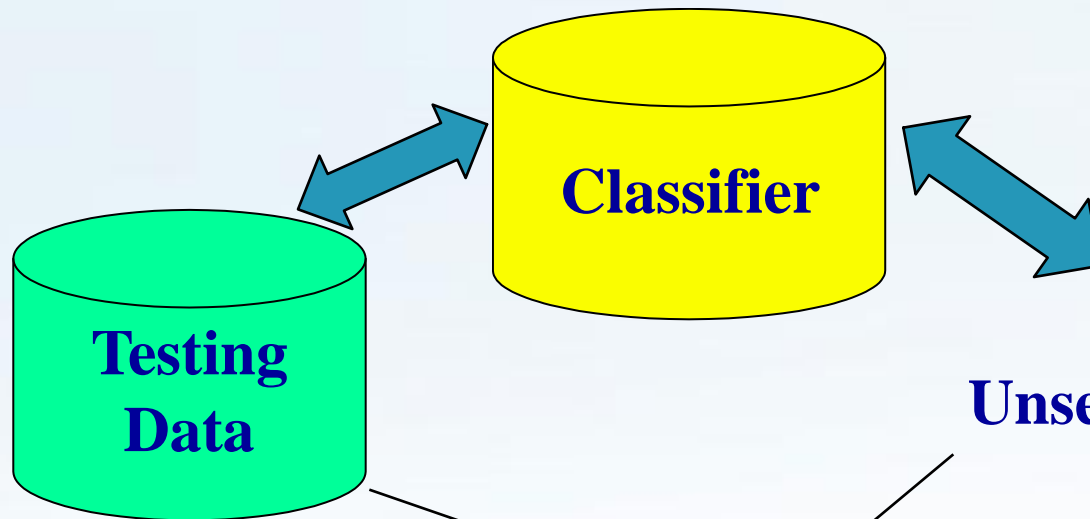
# Ví dụ về bước học





# Ví dụ về bước phân lớp

*IF rank = 'professor' OR years > 6 THEN tenured = 'yes'*



(Jeff, Professor, 4)

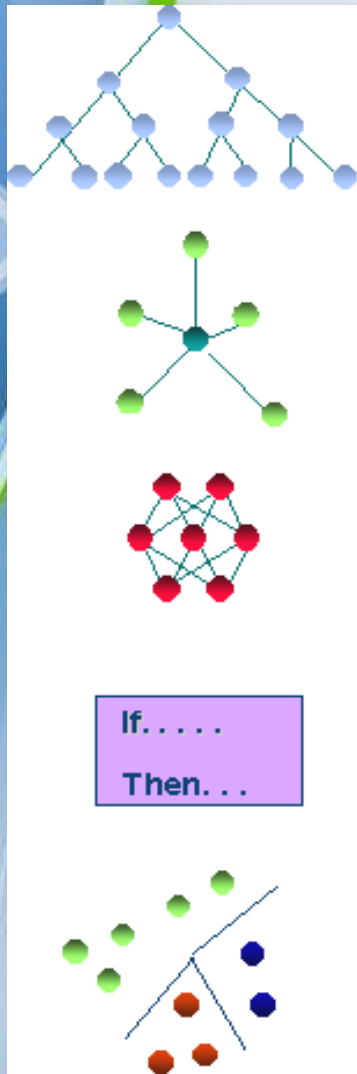
NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

Tenured?

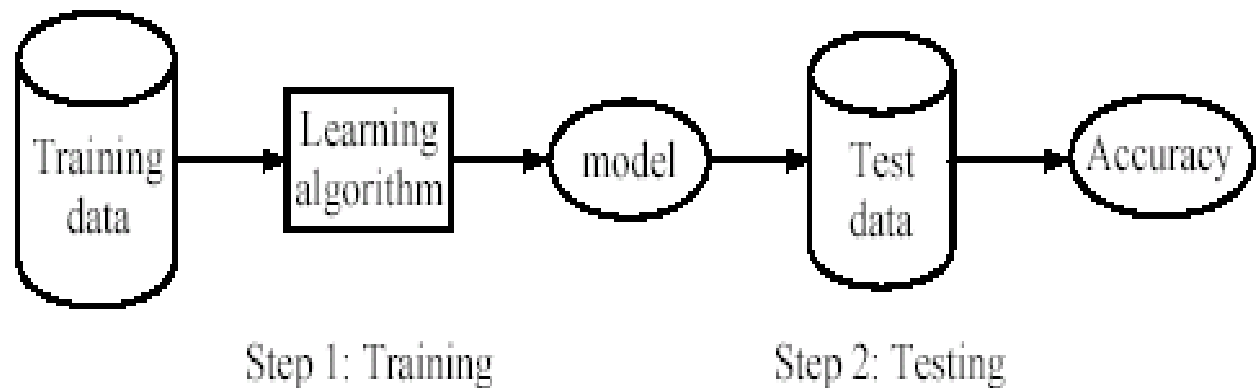


**Yes**

# Một số phương pháp phân lớp



- Phương pháp dựa trên cây quyết định
- Phương pháp dựa trên luật
- Phương pháp Naïve Bayes
- Phương pháp dựa trên thể hiện
- Mạng Noron
- SVM (support vector machine)
- Tập thô...



# Đánh giá mô hình phân lớp

- Độ chính xác của dự đoán
- Tốc độ
- Khả năng chịu lỗi (dữ liệu nhiều/thiếu)
- Tính dễ hiểu, dễ cài đặt
- Độ tốt của luật (kích thước, số lượng,...)
- ...

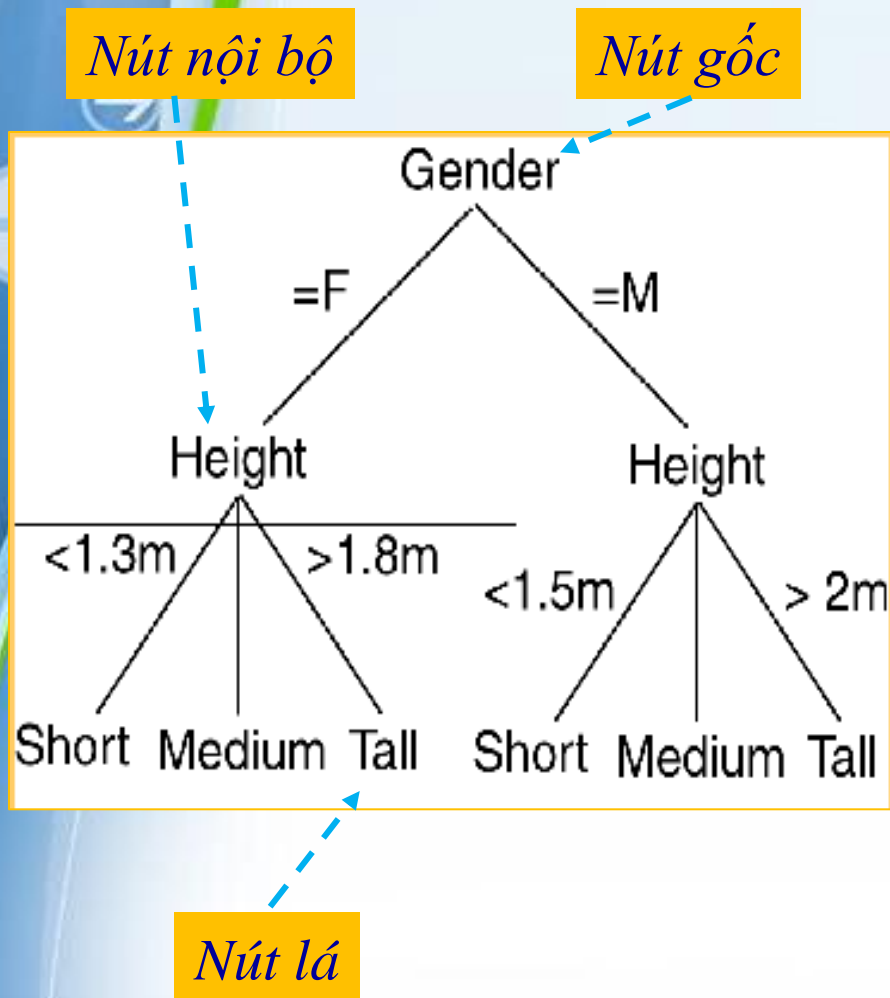


# Nội dung

---

- Khái niệm cơ sở về phân lớp
- **Phân lớp dựa trên cây quyết định**
  - Khái niệm cây quyết định
  - Các phương pháp dựa trên cây quyết định
  - Xây dựng cây quyết định
  - Tỉa cây
- Phân lớp dựa trên luật

# Định nghĩa cây quyết định



- Cây quyết định là một cấu trúc phân cấp của các nút và các nhánh
- 2 loại nút trên cây:
  - Nút nội bộ: mang tên thuộc tính của CSDL
  - Nút lá: mang tên lớp
- Nhánh: mang giá trị của thuộc tính

# Giới thiệu PP cây quyết định

---

- 1970 – 1980: J.Ross Quinlan đề xuất thuật toán cây quyết định ID3. Sau đó, đề xuất thuật toán C4.5 cải tiến từ ID3.
- 1984: L.Breiman và đồng sự đề xuất CART cho việc phát sinh cây quyết định nhị phân.
- Ngoài ra còn một số thuật toán khác như SLIQ (Mehta 1996), SPRINT (J.Shafer 1996), PUBLIC (Rastogi 1998), RainForest (Gehrke 1998)
- Các phương pháp chủ yếu dựa trên mô hình top-down và chia để trị.



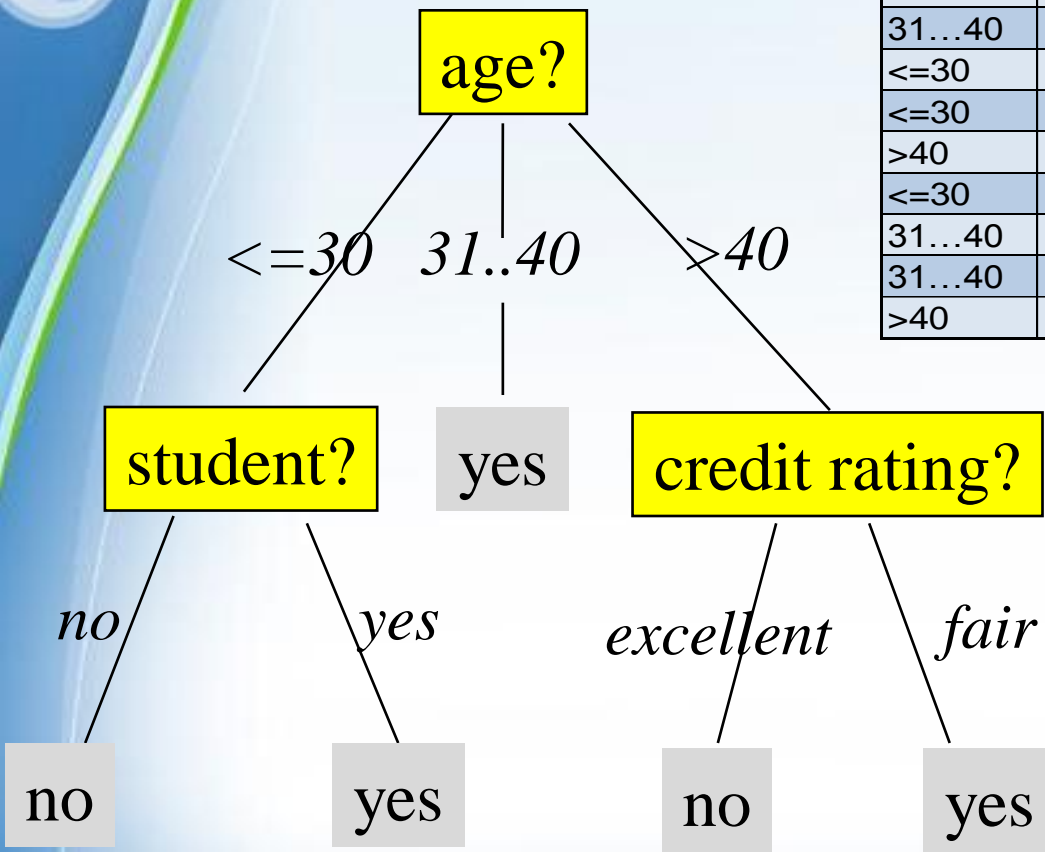
# Phát sinh cây quyết định

---

- Gồm 2 bước chính:
  - Bước 1: Xây dựng cây quyết định
    - Bắt đầu, toàn bộ tập dữ liệu huấn luyện được sử dụng để chọn thuộc tính cho gốc
    - Tập huấn luyện được phân chia đệ quy dựa trên thuộc tính được chọn.
  - Bước 2 : Tỉa cây
    - Xác định và loại bỏ bớt các nhánh gây nhiễu hay ngoại lai

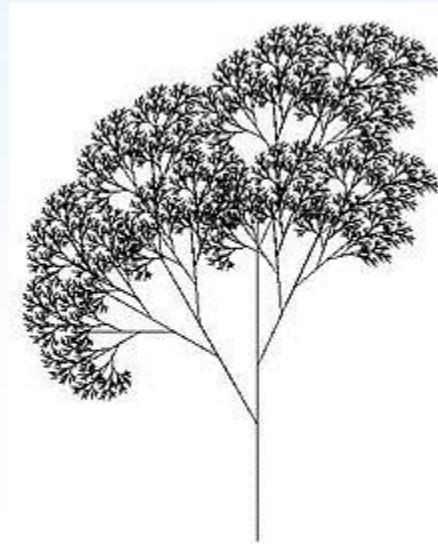
# Ví dụ phát sinh cây quyết định

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no



- Dữ liệu huấn luyện từ cửa hàng bán máy tính.
- Cây quyết định được tạo ra từ ID3

# Xây dựng Cây Quyết Định



# TT xây dựng cây quyết định

- (1) create a node  $N$ ;
- (2) **if** tuples in  $D$  are all of the same class,  $C$ , **then**
- (3)     return  $N$  as a leaf node labeled with the class  $C$ ;
- (4) **if** *attribute\_list* is empty **then**
- (5)     return  $N$  as a leaf node labeled with the majority class in  $D$ ; // majority voting
- (6) apply **Attribute\_selection\_method**( $D$ , *attribute\_list*) to find the “best” *splitting\_criterion*;
- (7) label node  $N$  with *splitting\_criterion*;
- (8) **if** *splitting\_attribute* is discrete-valued **and**  
      multiway splits allowed **then** // not restricted to binary trees
- (9)     *attribute\_list*  $\leftarrow$  *attribute\_list* – *splitting\_attribute*; // remove *splitting\_attribute*
- (10) **for each** outcome  $j$  of *splitting\_criterion*  
      // partition the tuples and grow subtrees for each partition
- (11)     let  $D_j$  be the set of data tuples in  $D$  satisfying outcome  $j$ ; // a partition
- (12)     **if**  $D_j$  is empty **then**
- (13)         attach a leaf labeled with the majority class in  $D$  to node  $N$ ;
- (14)     **else** attach the node returned by **Generate\_decision\_tree**( $D_j$ , *attribute\_list*) to node  $N$ ;
- endfor**
- (15) return  $N$ ;

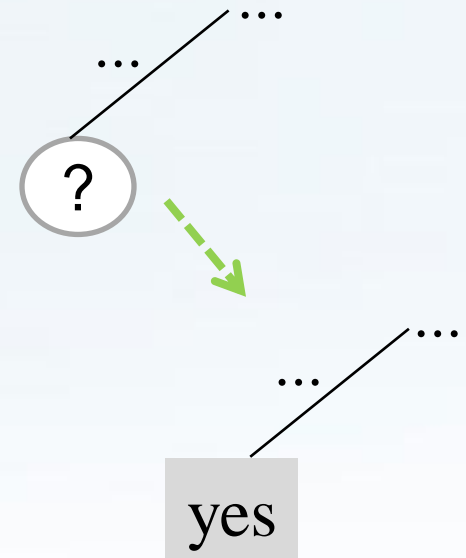
- (1) create a node  $N$ ;
- (2) if tuples in  $D$  are all of the same class,  $C$ , then
- (3) return  $N$  as a leaf node labeled with the class  $C$ ;

**Đầu vào: cơ sở dữ liệu  $D$ , tập các thuộc tính, phương pháp chọn thuộc tính**

1. Tạo ra một node  $N$
  2. Nếu các dòng trong  $D$  thuộc về cùng 1 lớp, thì node  $N$  trở thành lá và được đánh nhãn với lớp này.
- multiway splits allowed then // not restricted to binary trees
- (9)  $attribute\_list \leftarrow attribute\_list - splitting\_attribute$ ; // remove  $splitting\_attribute$
  - (10) for each outcome  $j$  of  $splitting\_criterion$   
// partition the tuples and grow subtrees for each partition
  - (11) let  $D_j$  be the set of data tuples in  $D$  satisfying outcome  $j$ ; // a partition
  - (12) if  $D_j$  is empty then
  - (13) attach a leaf labeled with the majority class in  $D$  to node  $N$ ;
  - (14) else attach the node returned by  $Generate\_decision\_tree(D_j, attribute\_list)$  to node  $N$ ;
  - endfor
  - (15) return  $N$ ;

# Ví dụ x/d cây quyết định

age	income	student	credit_rating	buys_computer
...	...	...	...	yes
...	...	...	...	yes
...	...	...	...	yes
...	...	...	...	yes
...	...	...	...	yes
...	...	...	...	yes
...	...	...	...	yes
...	...	...	...	yes



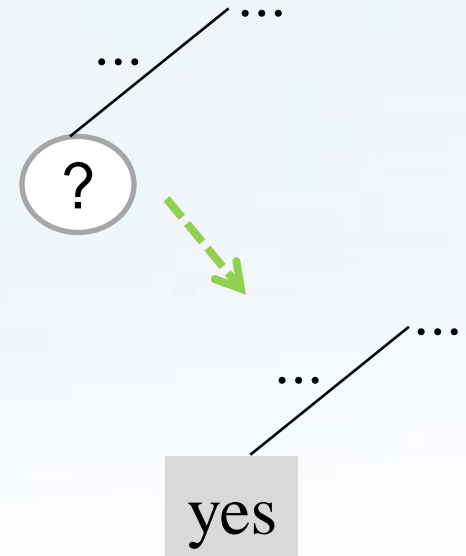
- Các dòng dữ liệu trong D đều có thuộc tính phân lớp *buys\_computer* là 'yes' nên node N trở thành node lá với giá trị là nhãn của lớp này



- (1) create a node  $N$ ;
- (2) if tuples in  $D$  are all of the same class,  $C$ , then
- (3) return  $N$  as a leaf node labeled with the class  $C$ ;
- (4) **if  $attribute\_list$  is empty then**
- (5) return  $N$  as a leaf node labeled with the majority class in  $D$ ; // majority voting
- (6) apply Attribute Selection
- (7) label node  $N$  with the majority class in  $D$
- (8) **if  $splitting\_attribute$  is chosen then**
  4. Nếu danh sách thuộc tính (không tính thuộc tính phân lớp) là rỗng thì  $N$  là node lá với nhãn của lớp xuất hiện nhiều nhất trong  $D$
- multiway splits allowed then // not restricted to binary trees
- (9)  $attribute\_list \leftarrow attribute\_list - splitting\_attribute$ ; // remove  $splitting\_attribute$
- (10) **for each** outcome  $j$  of  $splitting\_criterion$   
// partition the tuples and grow subtrees for each partition
- (11) let  $D_j$  be the set of data tuples in  $D$  satisfying outcome  $j$ ; // a partition
- (12) if  $D_j$  is empty then
- (13) attach a leaf labeled with the majority class in  $D$  to node  $N$ ;
- (14) else attach the node returned by  $Generate\_decision\_tree(D_j, attribute\_list)$  to node  $N$ ;
- endfor**
- (15) return  $N$ ;

# Ví dụ x/d cây quyết định

buys_computer
yes
yes
no
yes
yes
yes
no
yes

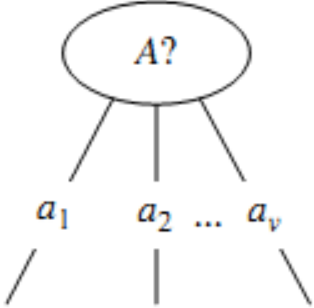

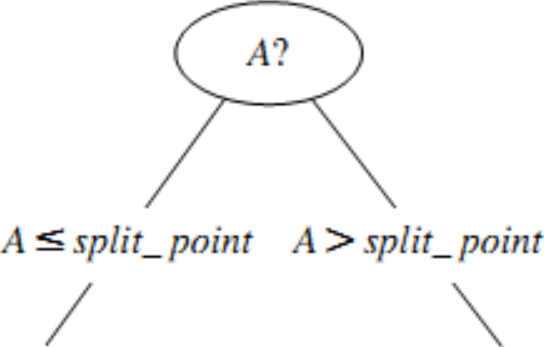
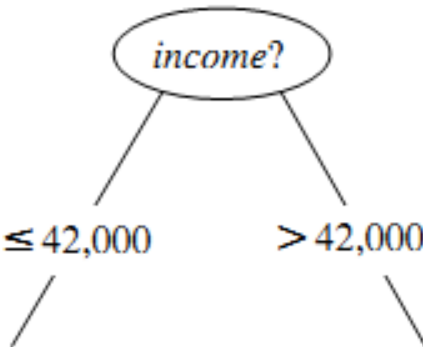


- D chỉ có thuộc tính phân lớp nên node N trở thành node lá với giá trị là nhãn của lớp xuất hiện nhiều nhất

- (1) create a node  $N$ ;
- (2) if tuples in  $D$  are all of the same class,  $C$ , then
- (3)     return  $N$  as a leaf node labeled with the class  $C$ ;
- (4) if *attribute\_list* is empty then
- (5)     return  $N$  as a leaf node labeled with the majority class in  $D$ ; // majority voting
- (6) apply **Attribute\_selection\_method**( $D$ , *attribute\_list*) to find the “best” *splitting\_criterion*;
- (7) label node  $N$  with *splitting\_criterion*;
- (8) if *splitting\_attribute* is discrete-valued and  
    multiway splits allowed then // not restricted to binary trees
- (9)     *attribute\_list*  $\leftarrow$  *attribute\_list* – *splitting\_attribute*; // remove *splitting\_attribute*
- (10) for each output class  $c$  in  $D$  do  
    // partition  $D$  into  $D_j$  based on *splitting\_attribute*  
    let  $D_j$  be the subset of  $D$  that belong to class  $c$ ;  
    if  $D_j$  is empty then  
    // no tuples in  $D_j$  to split on  
    // do not split on *splitting\_attribute*  
    // attach  $D_j$  to node  $N$  as a leaf node labeled with class  $c$ ;  
    else attach  $D_j$  to node  $N$  as a subtree;  
endfor
- (15) return  $N$ ;

6. Áp dụng phương pháp heuristic để chọn thuộc tính phân chia tốt nhất.
7. Node  $N$  được gán là thuộc tính này kèm với các tiêu chí chia (nếu thuộc tính liên tục thì tiêu chí chia là các điểm dữ liệu để từ đó chia dữ liệu)
8. Nếu thuộc tính chia là rời rạc thì bỏ nó ra khỏi danh sách thuộc tính

# Ví dụ x/d cây quyết định

Partitioning scenarios	Examples
(a) 	
(b) 	

(a) Thuộc tính chia ‘tốt nhất’ A là rời rạc

(b) Thuộc tính chia ‘tốt nhất’ A là liên tục, split\_point là điểm dữ liệu chia

```

(1) create a node  $N$ ;
(2) if tuples in  $D$  are all of the same class,  $C$ , then
(3)     return  $N$ 
(4) if attribute_list is empty then
(5)     return  $N$ 
(6) apply Attribute_selection_criteria to  $D$ 
(7) label node  $N$  with the majority class in  $D$ 
(8) if splitting_attribute is not null then
(9)     multiway split  $D$  on splitting_attribute
(10)    for each outcome  $j$  of splitting_criterion
(11)        // partition the tuples and grow subtrees for each partition
(12)        let  $D_j$  be the set of data tuples in  $D$  satisfying outcome  $j$ ; // a partition
(13)        if  $D_j$  is empty then
(14)            attach a leaf labeled with the majority class in  $D$  to node  $N$ ;
(15)        else attach the node returned by Generate_decision_tree( $D_j$ , attribute_list) to node  $N$ ;
(16)    endfor
(17) return  $N$ ;

```

10. Với mỗi tiêu chí chia của thuộc tính được chọn ở bước trước

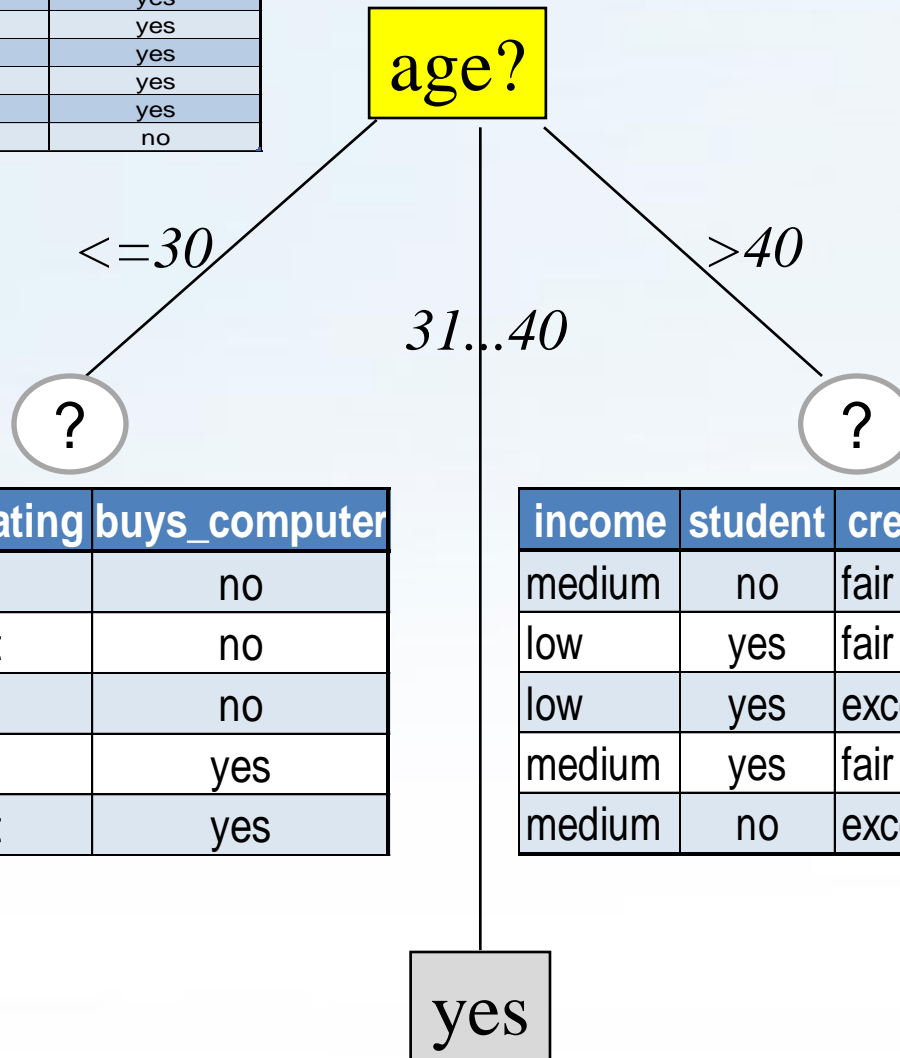
11. Chia tập dữ liệu  $D$  thành các tập dữ liệu con  $D_j$  theo từng tiêu chí

12. Nếu  $D_j$  rỗng thì  $N$  là node lá với nhãn của lớp xuất hiện nhiều nhất trong  $D$

13. Nếu không, gọi đệ quy lại hàm để tìm thuộc tính phân chia tốt nhất cho  $D_j$

# Ví dụ x/d cây quyết định

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no



income	student	credit_rating	buys_computer
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

$D_1$

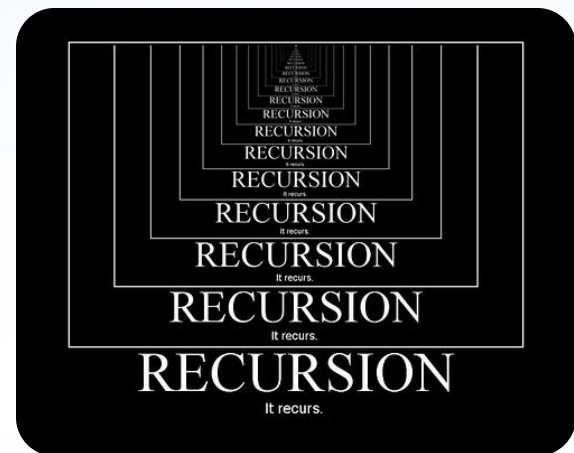
income	student	credit_rating	buys_computer
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

$D_2$



# Điểm dừng thuật toán

- Quá trình đệ quy dừng khi gặp một trong các điều kiện sau:
  - Tất cả các dòng dữ liệu trong D đều thuộc về cùng một lớp
  - Không còn thuộc tính để tiếp tục phân chia.
  - D rỗng



# Phương Pháp Chọn Thuộc Tính



# Phương pháp chọn thuộc tính

---

- Là một heuristic để chọn thuộc tính sao cho nó phân chia “tốt nhất” dữ liệu được cho vào các lớp.
- Một số heuristic:
  - Information Gain
  - Gain Ratio
  - Gini Index

# Information Gain (ID3/C4.5)

- Chọn thuộc tính có độ lợi thông tin (information gain) cao nhất
- Độ đo thông tin cần thiết để có thể phân lớp các mẫu trong D (cũng được gọi là độ entropy)

$$Info(D) = - \sum_{i=1}^m p_i \log_2(p_i)$$

D: là tập huấn luyện

$C_{i,D}$ : là các nhãn phân lớp trong D ( $i=1, \dots, m$ )

$p_i$ : xác suất một mẫu trong D thuộc về lớp  $C_i$  và bằng  $\frac{|C_{i,D}|}{|D|}$

# Ví dụ information gain

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

Info(D) = ?

$$Info(D) = I(9,5) = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.940$$

# Information Gain (ID3/C4.5)

- Thuộc tính A có các giá trị : $\{a_1, a_2, \dots, a_v\}$
- Dùng thuộc tính A để phân chia tập huấn luyện D thành v tập con  $\{D_1, D_2, \dots, D_v\}$
- Thông tin cần thiết để phân chia D theo thuộc tính A :

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times Info(D_j)$$



# Ví dụ information gain

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

$Info_{age}(D) = ?$

$$Info_{age}(D) = \frac{5}{14} Info_{D_{\leq 30}} + \frac{4}{14} Info_{D_{31..40}} + \frac{5}{14} Info_{D_{>40}} = 0.694$$

# Information Gain (ID3/C4.5)

- Độ lợi thông tin khi phân chia D dựa trên thuộc tính A:

$$Gain(A) = Info(D) - Info_A(D)$$

*Ví dụ:*

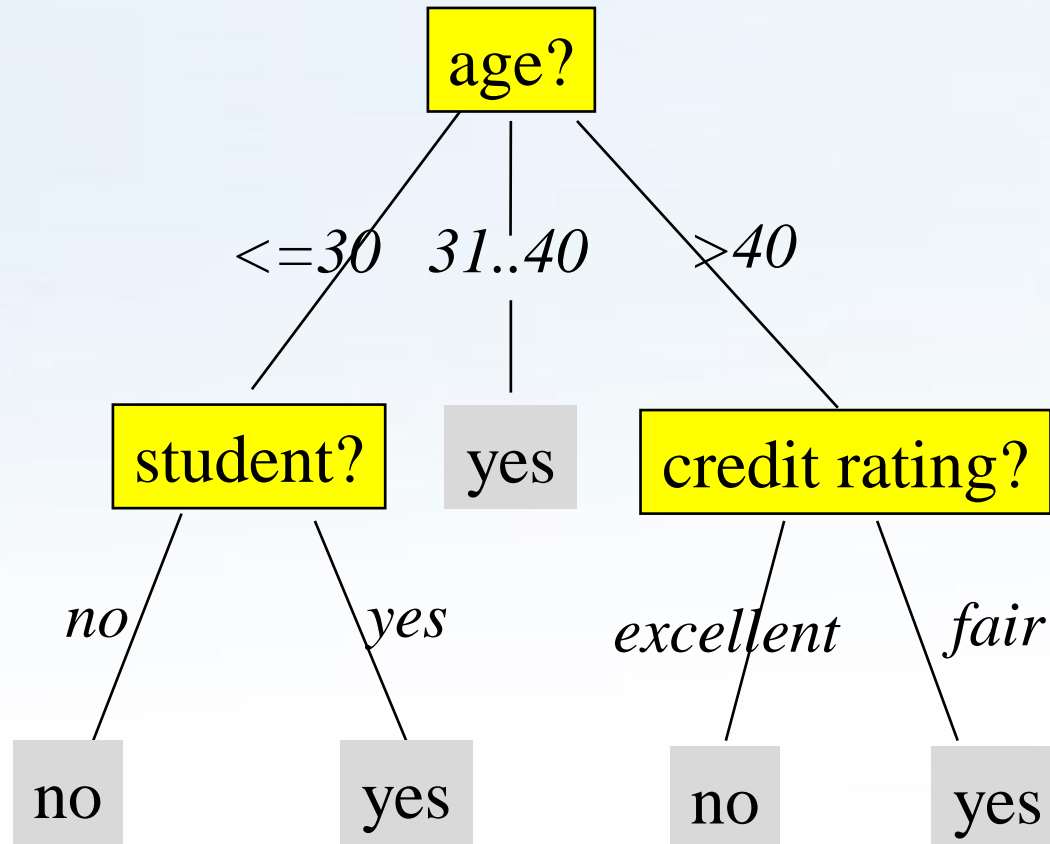
$$\begin{aligned} Gain(age) &= Info(D) - Info_{age}(D) \\ &= 0.940 - 0.694 \\ &= 0.246 \end{aligned}$$

# Bài tập 1

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

- Xây dựng cây quyết định cho dữ liệu sau với phương pháp chọn thuộc tính là Information Gain

# Bài tập 1 – Đáp án

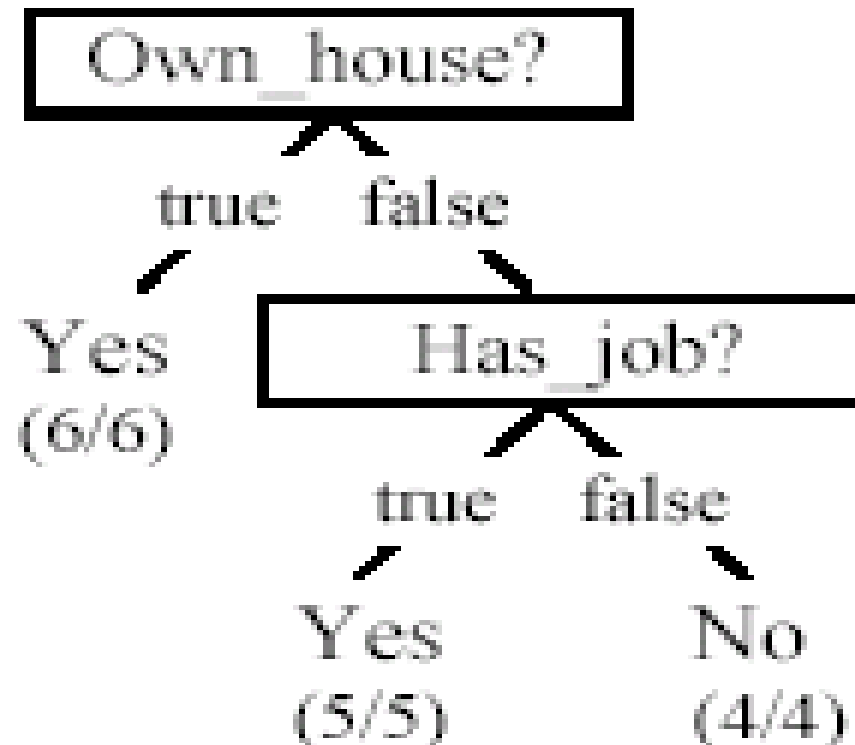


# Bài tập 2

- Yêu cầu tương tự bài tập 1

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

# Bài tập 2 – Đáp án



# Vấn đề của Information Gain

- Độ đo Information Gain thường hướng về các thuộc tính có nhiều giá trị → một số trường hợp các mẫu chia thuần nhất và không có ích cho việc phân lớp.
- Ví dụ: thuộc tính id của sản phẩm

	A	B
1	ProductID	Name
2		1 Adjustable Race
3	879	All-Purpose Bike Stand
4	712	AWC Logo Cap
5	3	BB Ball Bearing
6	2	Bearing Ball
7	877	Bike Wash - Dissolver
8	316	Blade
9	843	Cable Lock
10	952	Chain
11	324	Chain Stays
12	322	Chainring
13	320	Chainring Bolts

# Gain Ratio (C4.5)

- Giá trị chuẩn hóa (“chia thông tin”):

$$SplitInfo_A(D) = - \sum_{j=1}^v \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

- Độ đo Gain Ratio:

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}$$

- Thuộc tính có Gain Ratio lớn nhất sẽ chọn để chia



# Ví dụ Gain Ratio (C4.5)

- Ví dụ:

$$SplitInfo_{income}(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 1.557$$

- $GainRatio(income) = 0.029/1.557 = 0.019$
- $GainRatio(student)$ ?  $GainRatio(credit.)$ ?

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

# Gini Index (CART)

- Đánh giá độ *không thuần nhất* của dữ liệu:

$$Gini(D) = 1 - \sum_{i=1}^m p_i^2$$

$p_i$ : xác suất một mẫu trong  $D$  thuộc về lớp  $C_i$  và bằng  $\frac{|C_{i,D}|}{|D|}$

- Tương tự IG, độ Gini để phân chia  $D$  theo thuộc tính  $A \{a_1, a_2, \dots, a_v\}$ :

$$Gini_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times Gini(D_j)$$

*Chọn thuộc tính có độ Gini nhỏ nhất*

# Ví dụ Gini Index (CART)

$$Gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

$$Gini_{age}(D) = ?$$

$$Gini_{income}(D) = ?$$

$$Gini_{student}(D) = ?$$

$$Gini_{credit\_rating}(D) = ?$$

# Ví dụ Gini Index (CART) – Đáp án

---

$$Gini_{age}(D) = \frac{5}{14} gini(2,3) + \frac{4}{14} gini(4,0) + \frac{5}{14} gini(3,2) \\ = 0.343$$

$$Gini_{income}(D) = 0.44$$

$$Gini_{student}(D) = 0.367$$

$$Gini_{credit\_rating}(D) = 0.429$$

# Bài tập 3

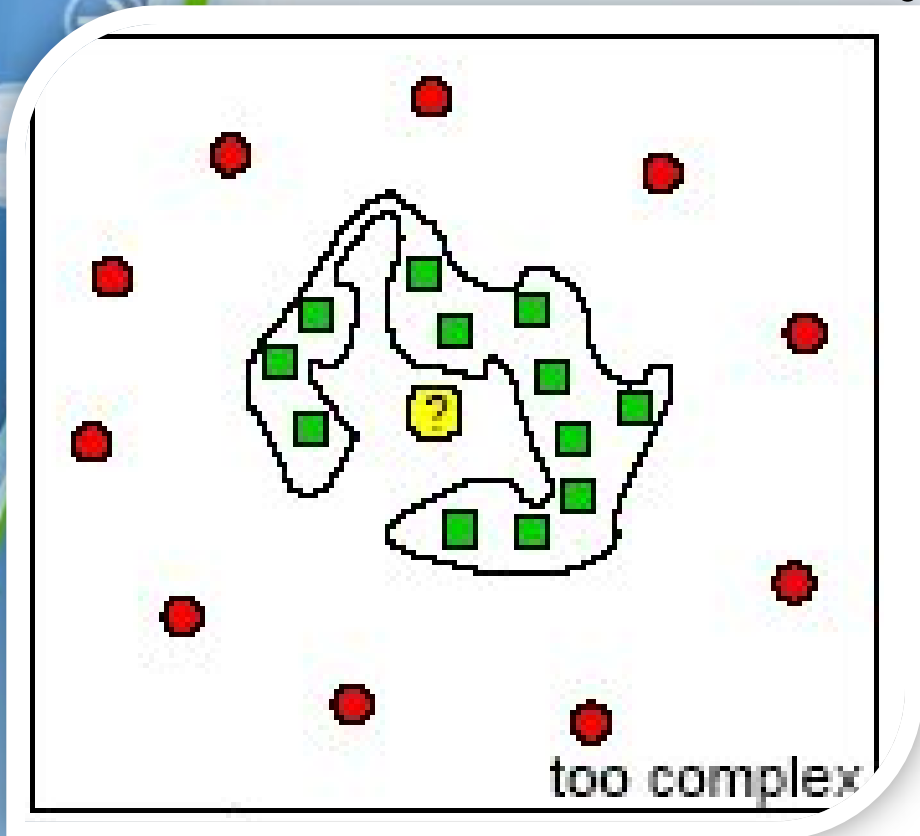
age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

• Xây dựng cây quyết định cho dữ liệu sau với phương pháp chọn thuộc tính là Gain Ratio và Gini Index

# Tỉa Cây

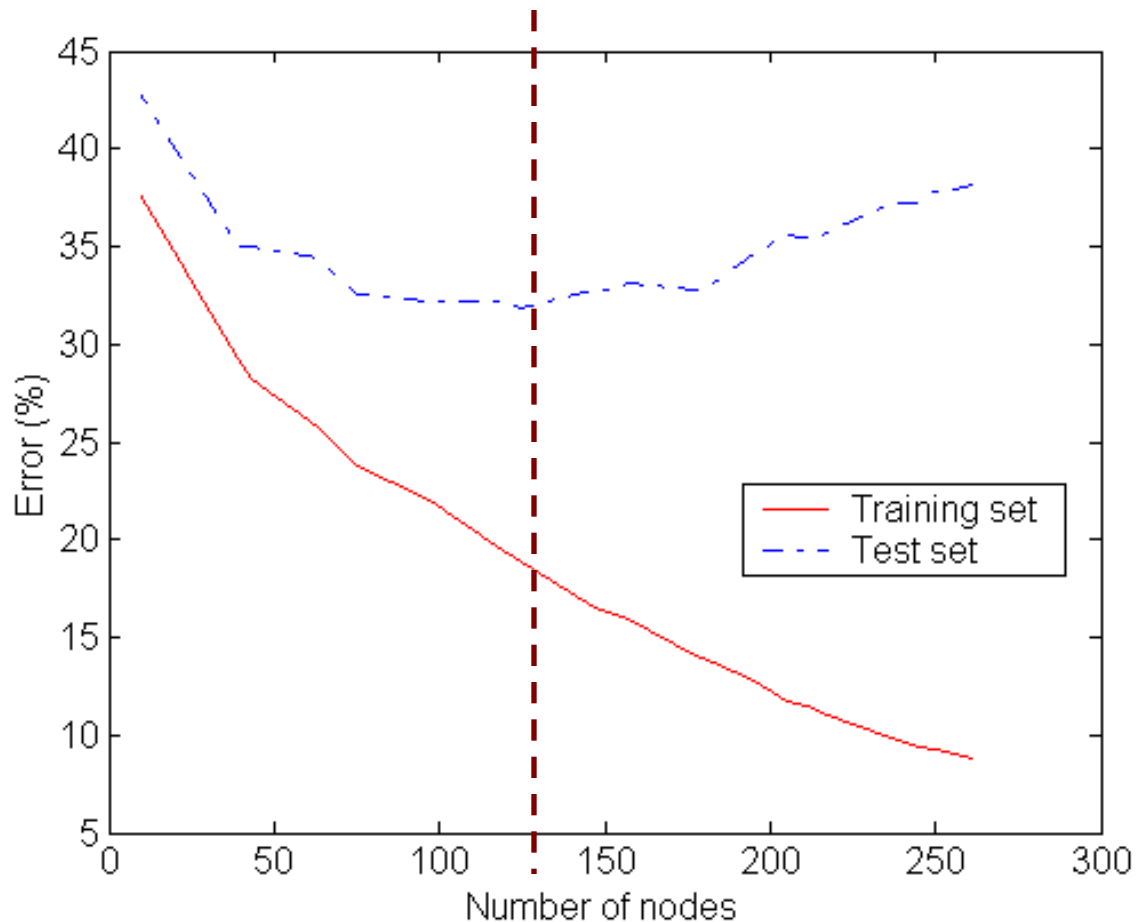


# Vấn đề cây quyết định (1/2)



- Quá khớp (overfitting):
  - Quá nhiều nhánh, một số nhánh bất thường do được tạo bởi dữ liệu nhiễu hay dữ liệu biên
  - Gây nên độ chính xác thấp cho mẫu chưa gặp bao giờ.

# Vấn đề cây quyết định (2/2)



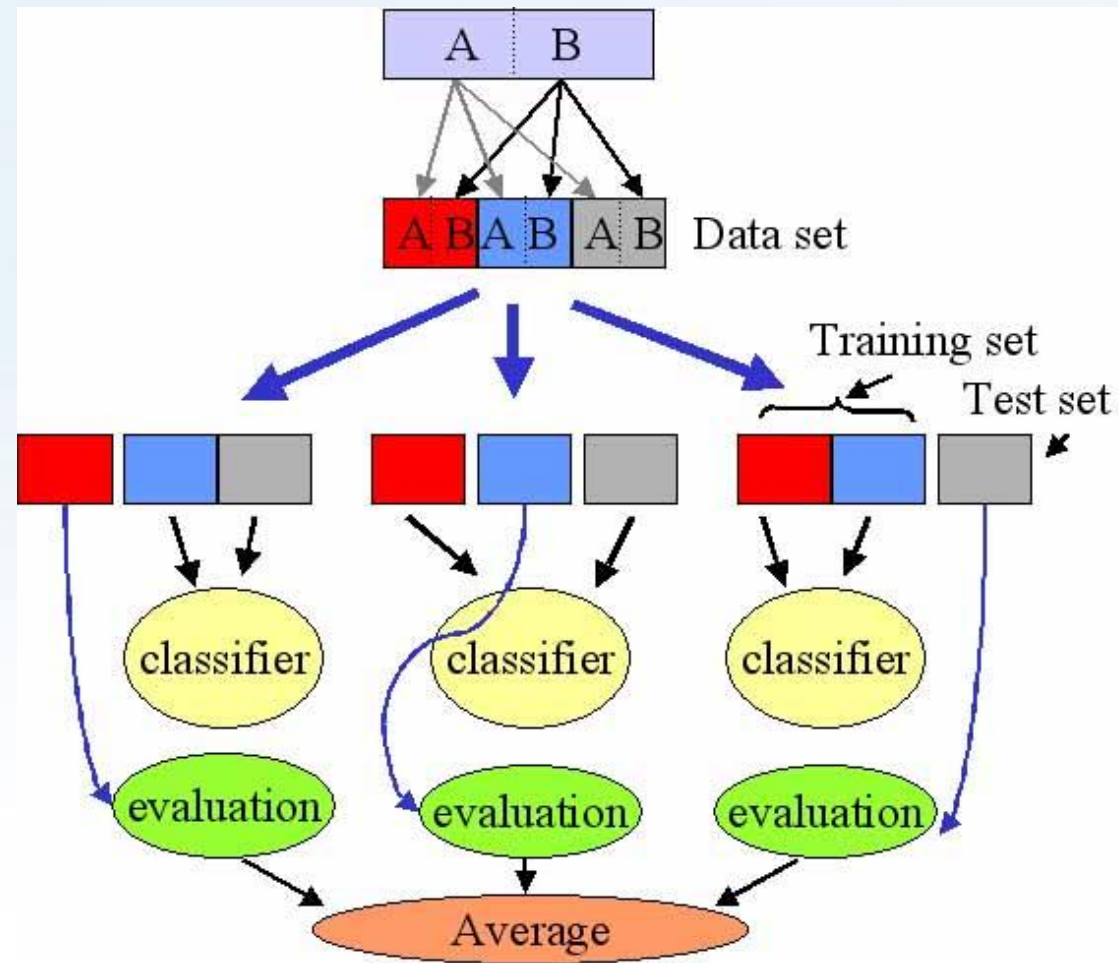


# Tỉa nhánh (1/2)

---

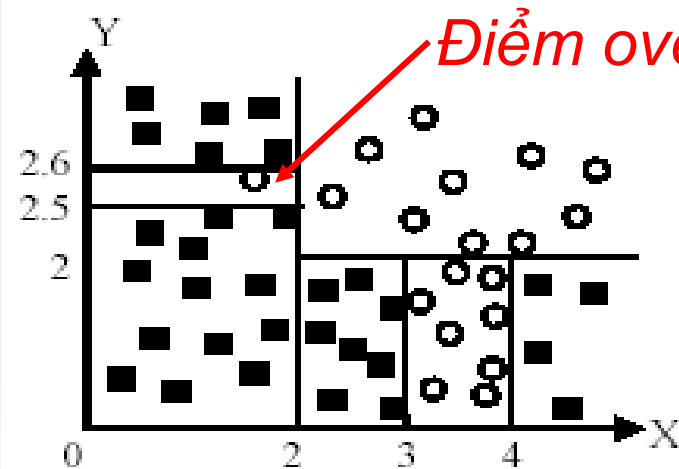
- Hai phương pháp để tránh *overfitting*:
  - Tỉa nhánh trước: dừng tạo nhánh sớm; không chia node nếu có một độ đo dưới ngưỡng
    - Khó để chọn một ngưỡng thích hợp
  - Tỉa nhánh sau: bỏ đi một số nhánh khi cây đã hoàn thành
    - Sử dụng tập dữ liệu khác nhau lấy từ dữ liệu huấn luyện để quyết định cây tỉa nhánh tốt nhất.

# Tỉa nhánh (2/2)

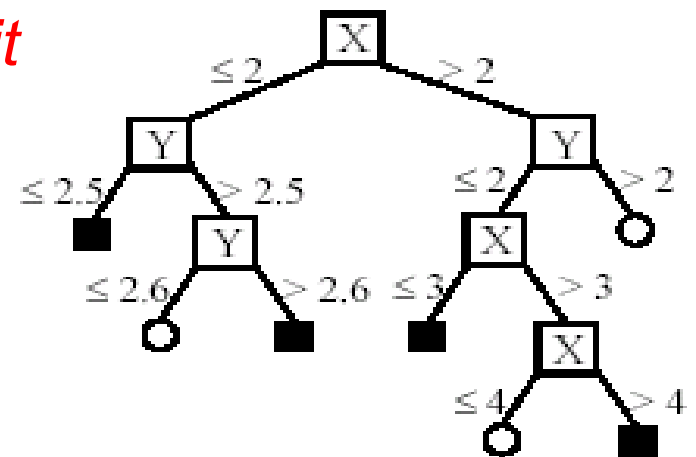


*Cross Validation*

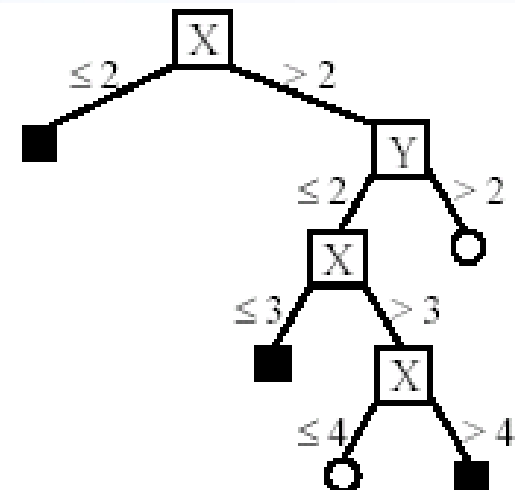
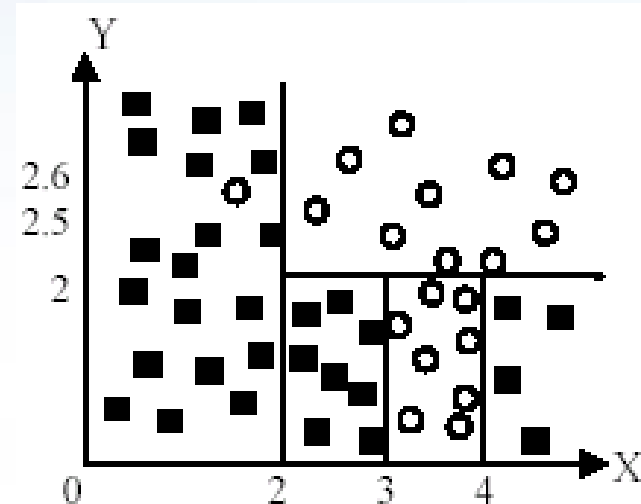
# Ví dụ tỉa nhánh



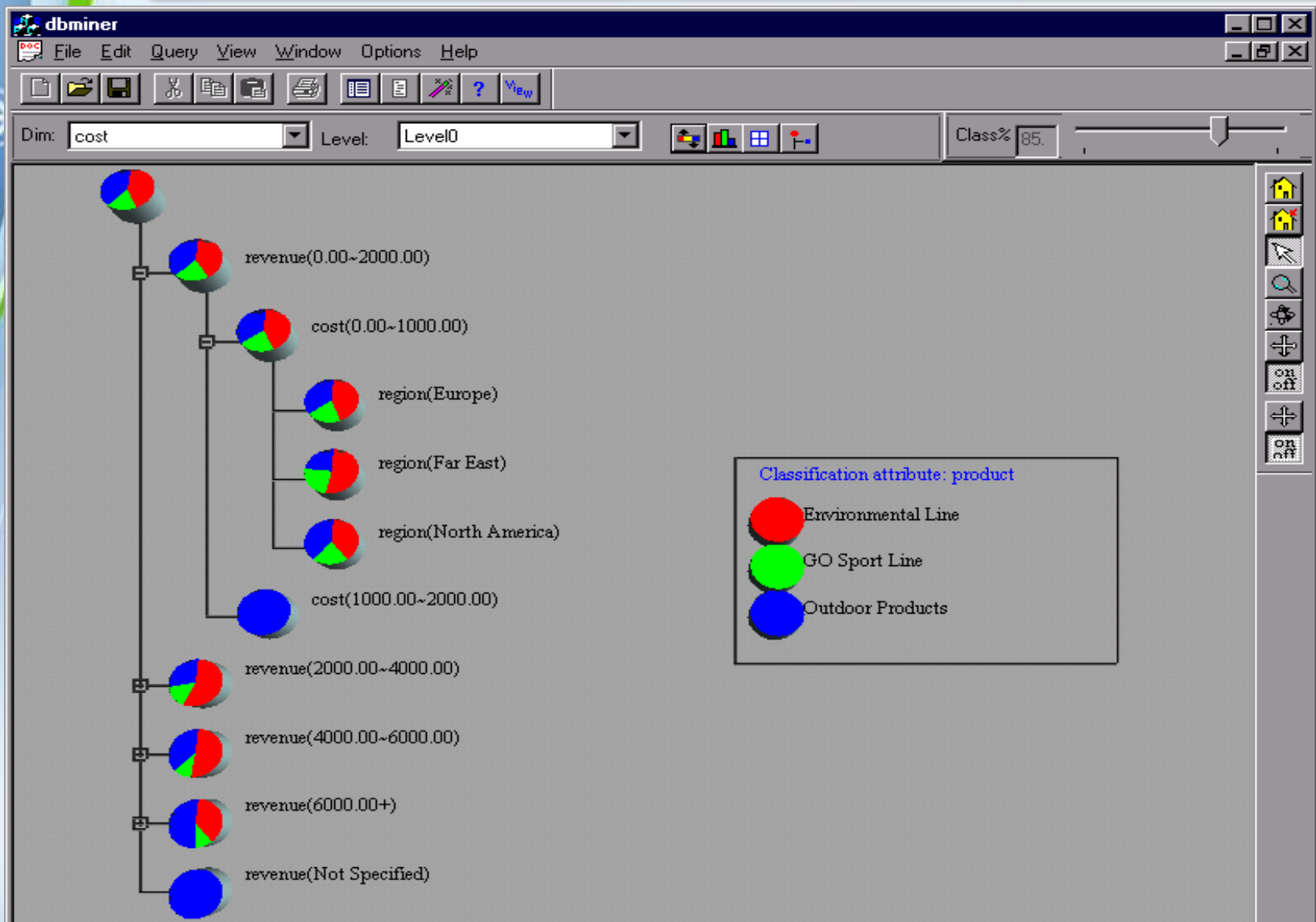
(A) A partition of the data space



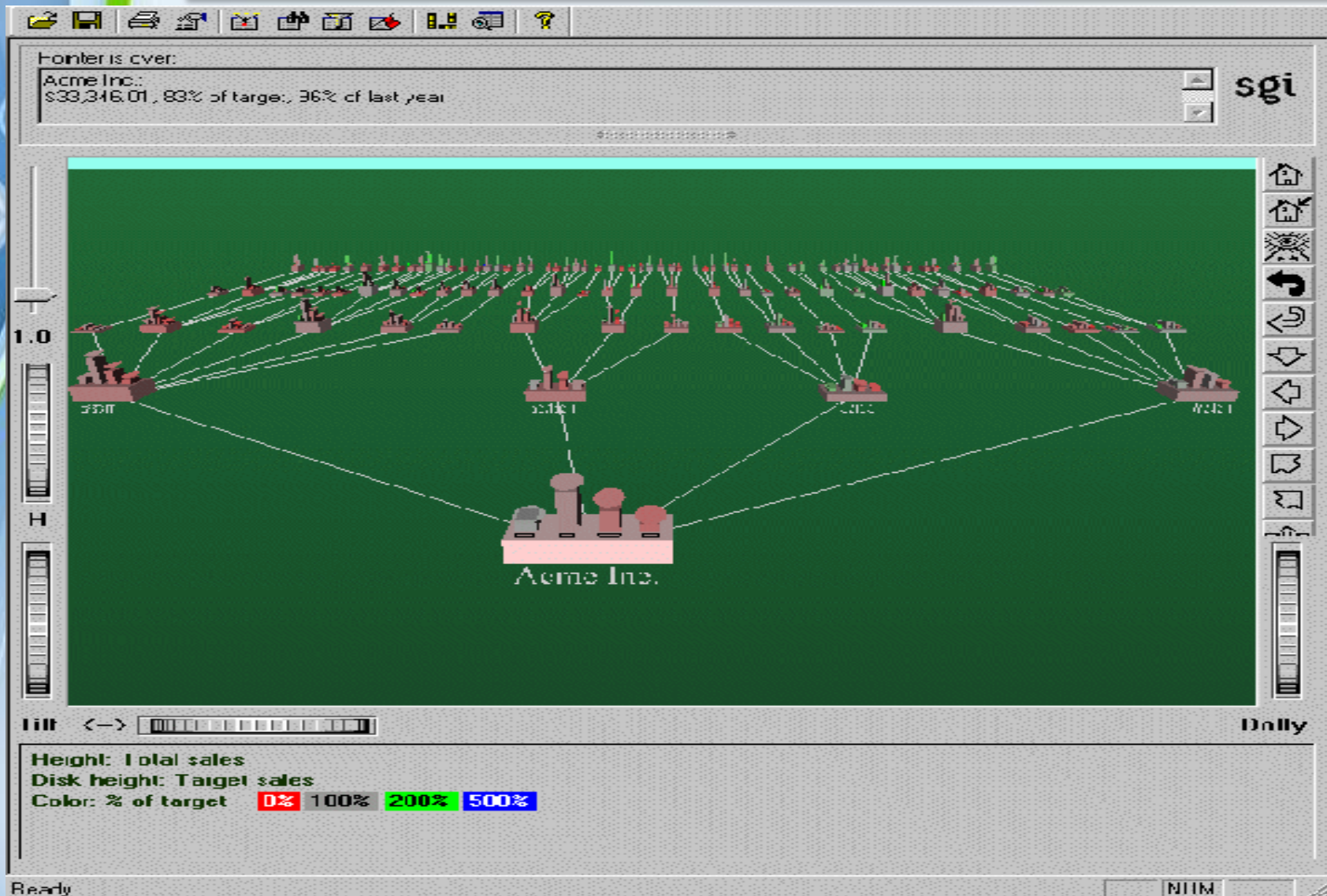
(B). The decision tree



# Thẻ hiện kết quả phân lớp(DBMiner)



# Minh họa cho cây quyết định trong dữ liệu SGI/MineSet 3.0



# Nội dung

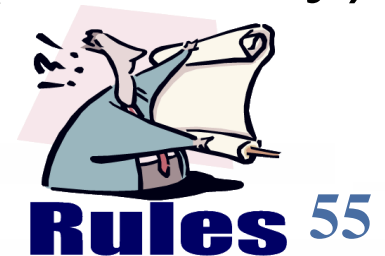
---

- Khái niệm cơ sở về phân lớp
- Phân lớp dựa trên cây quyết định
- **Phân lớp dựa trên luật**
  - Luật IF-THEN
  - Độ phủ và độ chính xác
  - Xây dựng luật
  - Đánh giá luật
  - Thuật toán ILA



# Phân lớp dùng luật IF-THEN

- Thể hiện tri thức ở dạng luật IF-THEN  
Ví dụ: IF *age* = youth AND *student* = yes  
THEN *buys\_computer* = yes
- Nếu một dòng dữ liệu thỏa điều kiện của luật thì người ta nói luật đó *phủ* (cover) được dòng dữ liệu
- Đánh giá luật dựa trên: độ phủ (coverage) và độ chính xác (accuracy)



# Độ phủ vs Độ chính xác

---

- Độ phủ của luật :  $\text{coverage}(R)$ 
  - Tỷ lệ các mẫu được phủ bởi luật

$$\text{coverage}(R) = \frac{n_{\text{covers}}}{|D|}$$

- Độ chính xác của luật :  $\text{accuracy}(R)$ 
  - Tỷ lệ mẫu được phân lớp đúng theo luật trong số các mẫu được phủ

$$\text{accuracy}(R) = \frac{n_{\text{correct}}}{n_{\text{covers}}}$$



# Ví dụ 1 về độ phủ và độ chính xác

<i>Tid</i>	<i>Refund</i>	<i>Marital Status</i>	<i>Taxable Income</i>	<i>Class</i>
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

R: IF *Marital Status*=Single  
THEN No

$$Coverage(R) = \frac{4}{10} = 40\%$$

$$Accuracy(R) = \frac{2}{5} = 50\%$$

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds
cat	warm	yes	no	no	mammals
leopard shark	cold	yes	no	yes	fishes
turtle	cold	no	no	sometimes	reptiles
penguin	warm	no	no	sometimes	birds
porcupine	warm	yes	no	no	mammals
eel	cold	no	no	yes	fishes
salamander	cold	no	no	sometimes	amphibians
gila monster	cold	no	no	no	reptiles
platypus	warm	no	no	no	mammals
owl	warm	no	yes	no	birds
dolphin	warm	yes	no	yes	mammals
eagle	warm	no	yes	no	birds

**R1: (Give Birth = no)  $\wedge$  (Can Fly = yes)  $\rightarrow$  Birds**

**R2: (Give Birth = no)  $\wedge$  (Live in Water = yes)  $\rightarrow$  Fishes**

**R3: (Give Birth = yes)  $\wedge$  (Blood Type = warm)  $\rightarrow$  Mammals**

**R4: (Give Birth = no)  $\wedge$  (Can Fly = no)  $\rightarrow$  Reptiles**

**R5: (Live in Water = sometimes)  $\rightarrow$  Amphibians**

*Tính độ phủ  
và độ chính  
xác cho  
từng luật.*

## Ví dụ 2 (tt)

**R1: (Give Birth = no)  $\wedge$  (Can Fly = yes)  $\rightarrow$  Birds;**

**R2: (Give Birth = no)  $\wedge$  (Live in Water = yes)  $\rightarrow$  Fishes;**

**R3: (Give Birth = yes)  $\wedge$  (Blood Type = warm)  $\rightarrow$  Mammals;**

**R4: (Give Birth = no)  $\wedge$  (Can Fly = no)  $\rightarrow$  Reptiles;**

**R5: (Live in Water = sometimes)  $\rightarrow$  Amphibians**

Dùng luật trên để  
phân lớp cho các  
mẫu mới sau

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

*Nhận xét?*

# Nhận xét ví dụ 2

**R1:** (Give Birth = no)  $\wedge$  (Can Fly = yes)  $\rightarrow$  Birds;

**R2:** (Give Birth = no)  $\wedge$  (Live in Water = yes)  $\rightarrow$  Fishes;

**R3:** (Give Birth = yes)  $\wedge$  (Blood Type = warm)  $\rightarrow$  Mammals;

**R4:** (Give Birth = no)  $\wedge$  (Can Fly = no)  $\rightarrow$  Reptiles;

**R5:** (Live in Water = sometimes)  $\rightarrow$  Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

- Mẫu “*lemur*” phủ bởi luật R3, nên được phân vào lớp “Mammals”
- Mẫu “*turtle*” phủ bởi cả luật R4 và R5 (vấn đề độ chồng)
- Mẫu “*dogfish shark*” không được phủ bởi bất kỳ luật nào.

***Cách giải quyết?***

# Phương pháp giải quyết (1/2)

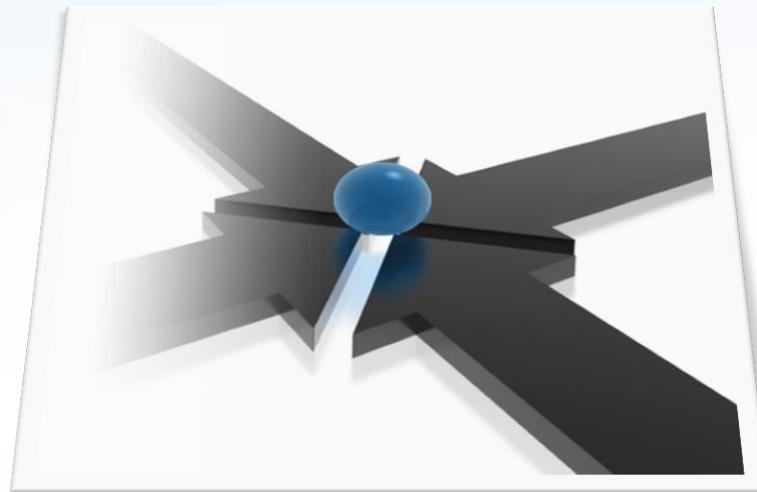
- Vấn đề độ:
  - Dựa trên kích thước của luật: các luật có *tập điều kiện* nhiều hơn sẽ có độ ưu tiên cao hơn
  - Dựa trên lớp: các lớp được xếp theo *độ phổ biến* hay theo *chi phí khi phân lớp sai*, các luật sẽ theo thứ tự ưu tiên của các lớp này.
  - Dựa trên luật: các luật được xếp hạng theo độ đo *chất lượng luật* (độ chính xác, độ phủ, ...) hoặc theo *ý kiến chuyên gia*



# Phương pháp giải quyết (2/2)

---

Nếu mẫu không được phủ  
bởi bất kỳ luật nào thì gán  
vào lớp mặc định



# Xây dựng luật



# Xây dựng luật phân lớp

---

- Phương pháp gián tiếp: rút luật từ các mô hình phân lớp khác
  - Ví dụ như cây quyết định, mạng nơron, ...
- Phương pháp trực tiếp: rút các luật trực tiếp từ dữ liệu
  - Một số thuật toán: RIPPER, CN2, ILA, FOIL, AQ, ...

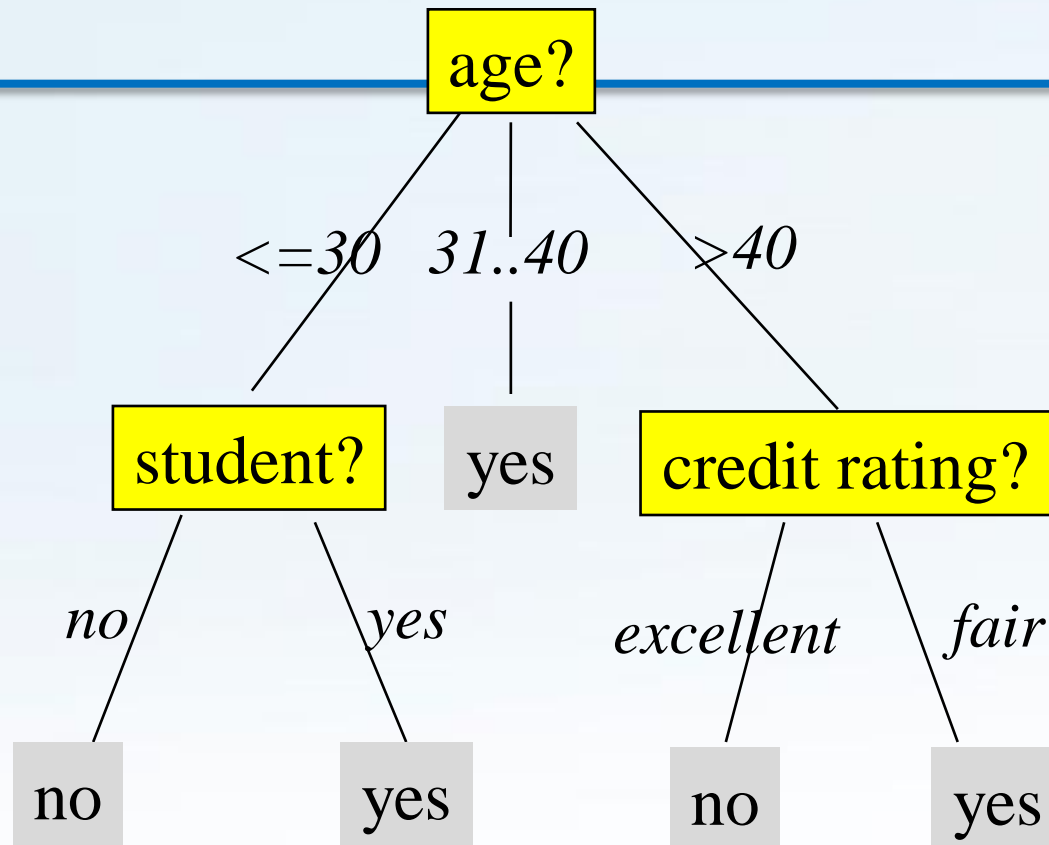


# Rút luật từ cây quyết định

- Luật dễ hiểu hơn cây quyết định lớn
- Mỗi luật được tạo ra từ mỗi nhánh từ gốc đến lá
- Mỗi cặp thuộc tính-giá trị dọc theo đường dẫn tạo nên phép kết
- Node lá là lớp dự đoán
- Luật mang tính toàn diện và loại trừ lẫn nhau



# Ví dụ rút luật từ cây



IF *age* = young AND *student* = *no*

IF *age* = young AND *student* = *yes*

IF *age* = mid-age

IF *age* = old AND *credit\_rating* = *excellent* THEN *buys\_computer* = *no*

IF *age* = old AND *credit\_rating* = *fair* THEN *buys\_computer* = *yes*

THEN *buys\_computer* = *no*

THEN *buys\_computer* = *yes*

THEN *buys\_computer* = *yes*

# Phương pháp trực tiếp

---

- Thuật toán *phủ tuần tự*. Các luật sẽ được học tuần tự.
- Mỗi luật trong lớp  $c_i$  sẽ phủ nhiều mẫu của  $c_i$  nhưng không phủ (hoặc phủ ít) mẫu của các lớp khác.
- Ưu điểm so với cây quyết định: các luật có thể rút ra đồng thời

# Thuật toán phủ tuần tự (1/2)

---

**B0:** Bắt đầu từ luật rỗng

**B1:** Với mỗi lớp  $c_i$

**B1.1:** Sử dụng hàm *Learn-One-Rule* để tìm ra luật “*tốt nhất*” cho lớp hiện tại

**B1.2:** Loại các mẫu bị phủ bởi luật ra khỏi DL

**B1.3:** Lặp lại quá trình từ B1.1 cho đến khi gặp điều kiện dừng (ví dụ như không còn mẫu hoặc độ đo chất lượng thấp hơn ngưỡng do người dùng xác định)

# Thuật toán phủ tuần tự (2/2)

**Algorithm: Sequential covering.** Learn a set of IF-THEN rules for classification.

**Input:**

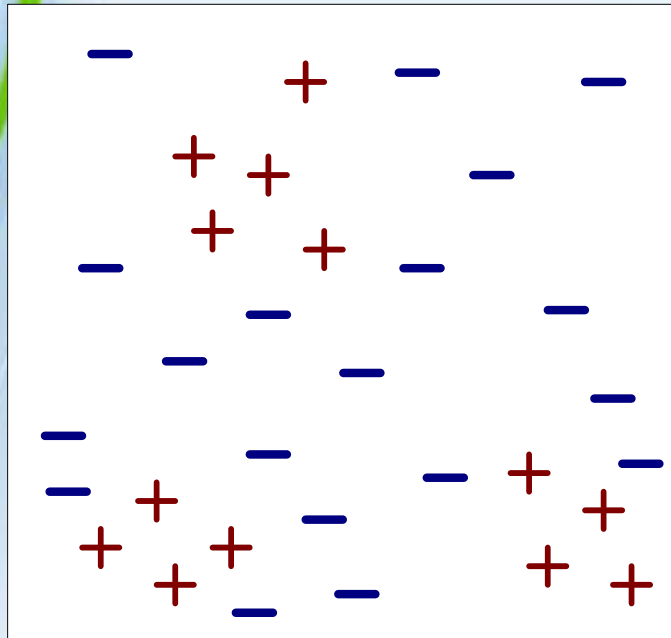
- $D$ , a data set of class-labeled tuples;
- $Att\_vals$ , the set of all attributes and their possible values.

**Output:** A set of IF-THEN rules.

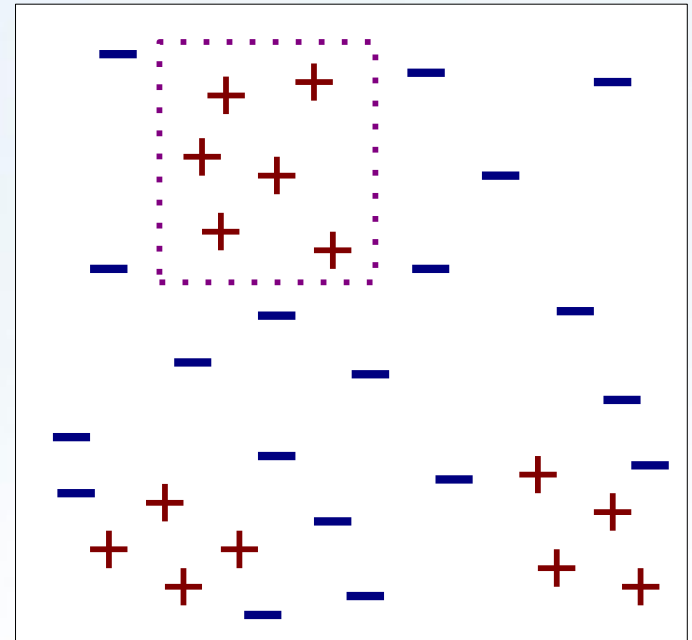
**Method:**

- (1)  $Rule\_set = \{\}$ ; // initial set of rules learned is empty
- (2) **for each** class  $c$  **do**
- (3)     **repeat**
- (4)          $Rule = \text{Learn\_One\_Rule}(D, Att\_vals, c)$ ;
- (5)         remove tuples covered by  $Rule$  from  $D$ ;
- (6)          $Rule\_set = Rule\_set + Rule$ ; // add new rule to rule set
- (7)     **until** terminating condition;
- (8) **endfor**
- (9) return  $Rule\_Set$ ;

# Ví dụ thuật toán phủ tuần tự



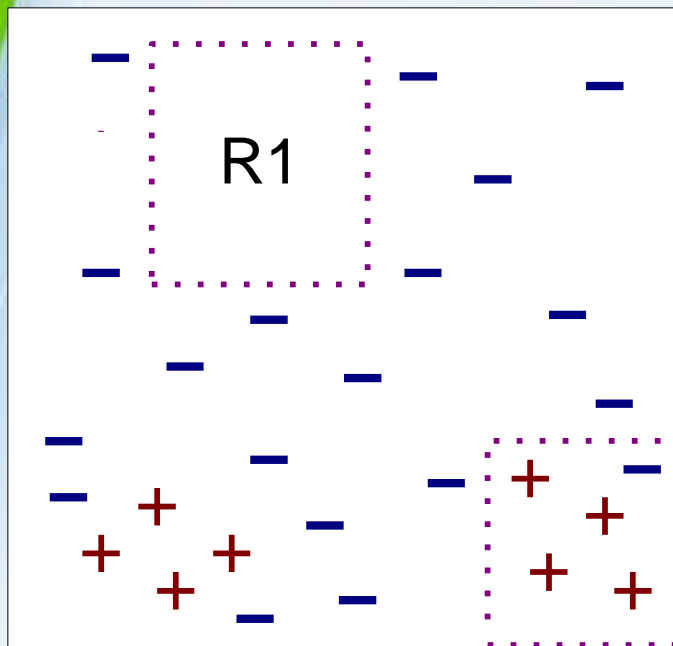
(i) Original Data



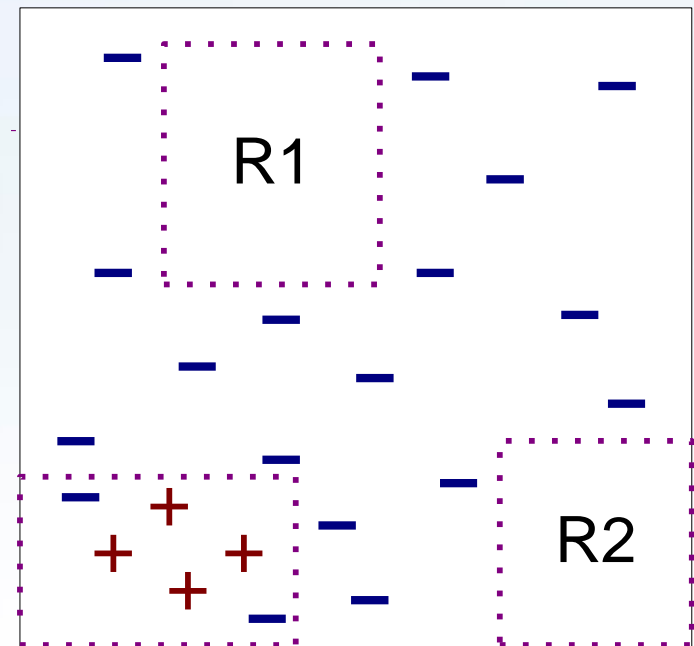
(ii) Step 1

*Mẫu dương (+) là các mẫu được phân vào lớp  $c_i$  đang xét.  
Các mẫu thuộc lớp khác là mẫu âm (-)*

# Ví dụ thuật toán phủ tuần tự (tt)



(iii) Step 2



(iv) Step 3

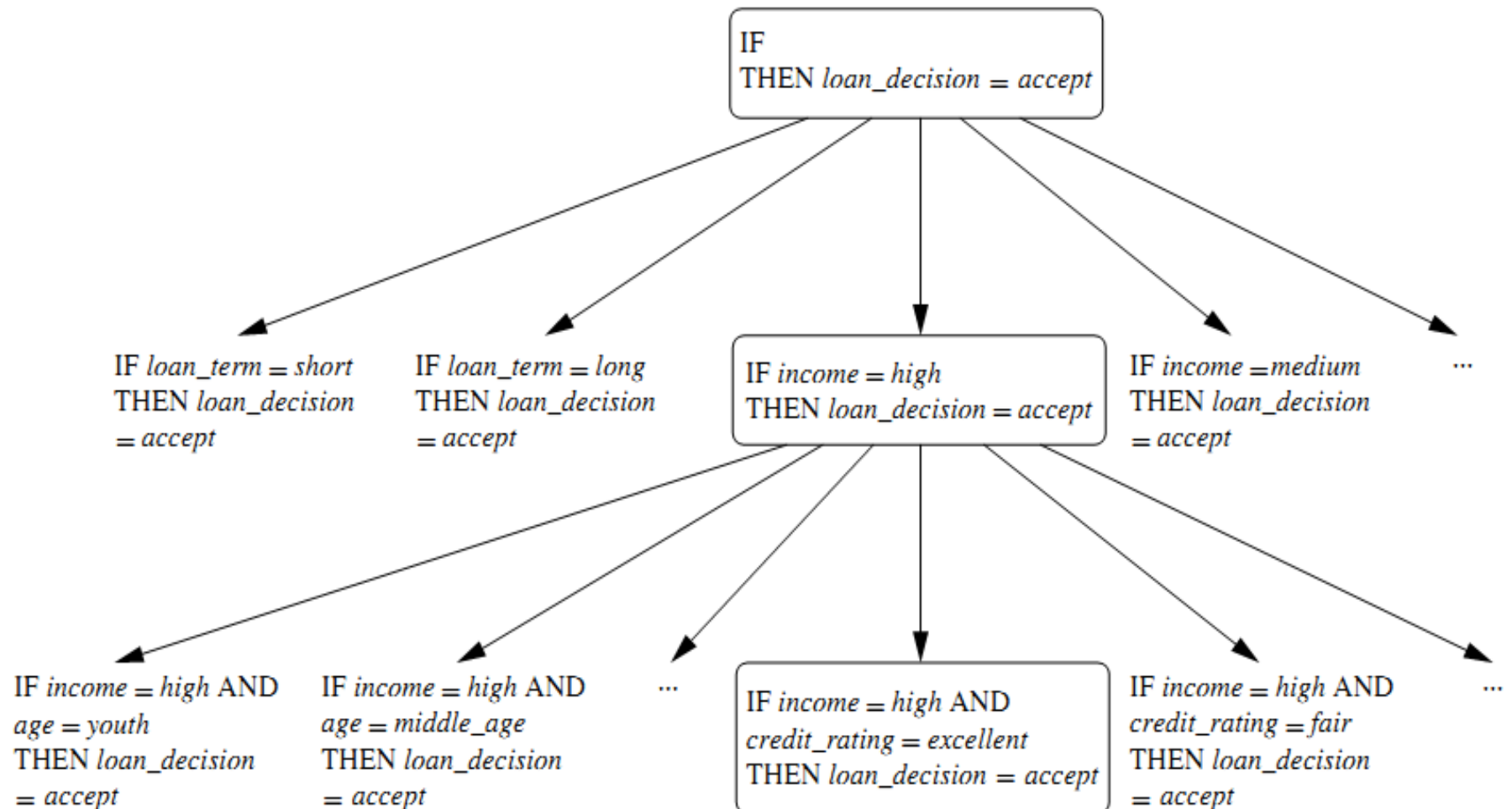
# Hàm Learn-One-Rule

---

- Bắt đầu với luật chung nhất: thuộc tính rỗng
  - IF THEN  $c_i$
- Lần lượt, thêm các thuộc tính mới sử dụng chiến lược tìm kiếm tham lam theo độ sâu
  - Chọn một thuộc tính cải thiện chất lượng của luật tốt nhất



# Ví dụ hàm Learn-One-Rule



# Độ đo chất lượng luật

- Một số độ đo có thể:
  - Độ bao phủ
  - Độ chính xác
  - FOIL (First Order Inductive Learner)
  - ...
- Độ đo FOIL dựa trên *Information Gain*. Nó hướng đến các luật có độ chính xác cao và bao phủ rất nhiều mẫu dương

# Độ đo chất lượng luật

- Gọi  $R$  là luật đang có hiện tại
  - Ví dụ: IF  $đk$  THEN  $c_i$
- $R'$  là luật được mở rộng từ  $R$ 
  - Ví dụ: IF  $đk \wedge (att_j = val_k)$  THEN  $c_i$
- Gọi  $pos$  là số mẫu dương,  $neg$  là số mẫu âm được phủ bởi luật  $R$
- $pos'$  là số mẫu dương,  $neg'$  là số mẫu âm được phủ bởi luật  $R'$

# Độ đo chất lượng luật

---

- FOIL đánh giá độ tăng cường thông tin (information gain) khi mở rộng luật:

$$FOIL\_Gain = pos' \times \left( \log_2 \frac{pos'}{pos' + neg'} - \log_2 \frac{pos}{pos + neg} \right)$$

- Luật có độ tăng cường lớn nhất sẽ được giữ lại

# Tỉa luật

- Để tránh Overfitting, sử dụng một tập dữ liệu test để tỉa bớt luật (rule pruning):

$$FOIL\_Prune(R) = \frac{pos - neg}{pos + neg}$$

pos (neg) là số mẫu dương (âm) phủ bởi R trong tập test

- Một luật bị tỉa bằng cách bớt đi một thuộc tính trong luật.
- Nếu phiên bản R sau khi tỉa có chất lượng tốt hơn (FOIL\_Prune nhỏ hơn) thì R sẽ bị tỉa.

# Nhận xét rút luật trực tiếp

- Độ chính xác: giống với cây quyết định
- Hiệu quả: chạy chậm hơn so với cây quyết định vì:
  - Để phát sinh mỗi luật, tất cả các luật có thể đều phải thử trên dữ liệu (không hoàn toàn nhưng vẫn nhiều)
  - Khi dữ liệu lớn và/hay số lượng thuộc tính-giá trị nhiều, thuật toán chạy rất chậm.
- Tính chặt chẽ của luật: mỗi luật có thể không độc lập với luật khác bởi vì luật được tìm thấy sau khi dữ liệu phủ bị luật trước đó bỏ đi.



# Thuật toán học trực tiếp ILA

# ILA – Học Quy Nạp

---

- M.Tolun, 1998, ILA – Inductive Learning Algorithm
- Xác định các luật IF-THEN trực tiếp từ tập huấn luyện (phát triển luật theo hướng từ tổng quát -> cụ thể)
- Chia tập huấn luyện thành các bảng con theo từng giá trị của lớp.
- Thực hiện việc so sánh các giá trị của thuộc tính trong từng bảng con và tính số lần xuất hiện.
- Thuộc tính có dạng phi số, giá trị rời rạc



# Thuật toán Học Quy Nạp (ILA)

---

B1: Chia tập mẫu thành các tập con ứng với từng phân lớp

B2: Với mỗi bảng con

B3:        Với mỗi tổ hợp thuộc tính có thể (bắt đầu với số lượng = 1)

B4:                Tìm các giá trị chỉ xuất hiện ở bảng con này mà không xuất hiện ở các bảng con khác

B5:                (Nếu có nhiều tổ hợp thì chọn tổ hợp có số lượng mẫu tin nhiều nhất)

B6:                Sử dụng tổ hợp thuộc tính, giá trị vừa tìm được để tạo luật

B7:                Bỏ đi các dòng phủ bởi luật

B8:        Nếu còn dòng chưa xét, lặp lại B3

B9: Lặp lại B2 với các bảng con

# Ví dụ ILA

- Cho bảng dữ liệu sau:

STT	Kích cỡ	Màu sắc	Hình dáng	Quyết định
1	Vừa	Xanh dương	Hộp	Mua
2	Nhỏ	Đỏ	Nón	Không mua
3	Nhỏ	Đỏ	Cầu	Mua
4	Lớn	Đỏ	Nón	Không mua
5	Lớn	Xanh lá	Trụ	Mua
6	Lớn	Đỏ	Trụ	Không mua
7	Lớn	Xanh lá	Cầu	Mua

# Ví dụ ILA (tt)

- Chia bảng thành các bảng con ứng với từng phân lớp:

STT	Kích cỡ	Màu sắc	Hình dáng	Quyết định
1	Vừa	Xanh dương	Hộp	Mua
3	Nhỏ	Đỏ	Cầu	Mua
5	Lớn	Xanh lá	Trụ	Mua
7	Lớn	Xanh lá	Cầu	Mua

STT	Kích cỡ	Màu sắc	Hình dáng	Quyết định
2	Nhỏ	Đỏ	Nón	Không mua
4	Lớn	Đỏ	Nón	Không mua
6	Lớn	Đỏ	Trụ	Không mua

# Ví dụ ILA (tt)

- Chọn tổ hợp thuộc tính (từ 1) có nhiều giá trị xuất hiện ở bảng này nhất mà không xuất hiện các bảng khác

STT	Kích cỡ	Màu sắc	Hình dáng	Quyết định
1	Vừa	Xanh dương	Hộp	Mua
3	Nhỏ	Đỏ	Cầu	Mua
5	Lớn	Xanh lá	Trụ	Mua
7	Lớn	Xanh lá	Cầu	Mua

STT	Kích cỡ	Màu sắc	Hình dáng	Quyết định
2	Nhỏ	Xanh lá	Hộp	Không mua
4	Lớn	Xanh lá	Cầu	Không mua
6	Lớn	Xanh lá	Trụ	Không mua

Chọn thuộc tính Màu sắc  
với giá trị Xanh lá

# Ví dụ ILA (tt)

- Xây dựng luật từ tổ hợp thuộc tính đó và xóa các mẫu phủ bởi luật.

STT	Kích cỡ	Màu sắc	Hình dáng	Quyết định
1	Vừa	Xanh dương	Hộp	Mua
3	Nhỏ	Đỏ	Cầu	Mua

IF Màu sắc = Xanh lá THEN Quyết định = Mua

STT	Kích cỡ	Màu sắc	Hình dáng	Quyết định
2	Nhỏ	Đỏ	Nón	Không mua
4	Lớn	Đỏ	Nón	Không mua
6	Lớn	Đỏ	Trụ	Không mua

# Ví dụ ILA (tt)

STT	Kích cỡ	Màu sắc	Hình dáng	Quyết định
3	Nhỏ	Đỏ	Cầu	Mua

IF Màu sắc = Xanh lá

THEN Quyết định = Mua

IF Kích cỡ = Vừa

THEN Quyết định = Mua

STT	Kích cỡ	Màu sắc	Hình dáng	Quyết định
2	Nhỏ	Đỏ	Nón	Không mua
4	Lớn	Đỏ	Nón	Không mua
6	Lớn	Đỏ	Trụ	Không mua

# Ví dụ ILA (tt)

IF Màu sắc = Xanh lá THEN Quyết định = Mua  
IF Kích cỡ = Vừa THEN Quyết định = Mua  
IF Hình dáng = Cầu THEN Quyết định = Mua

STT	Kích cỡ	Màu sắc	Hình dáng	Quyết định
2	Nhỏ	Đỏ	Nón	Không mua
4	Lớn	Đỏ	Nón	Không mua
6	Lớn	Đỏ	Trụ	Không mua

# Ví dụ ILA (tt)

STT	Kích cỡ	Màu sắc	Hình dáng	Quyết định
1	Vừa	Xanh dương	Hộp	Mua
3	Nhỏ	Đỏ	Cầu	Mua
5	Lớn	Xanh lá	Trụ	Mua
7	Lớn	Xanh lá	Cầu	Mua

STT	Kích cỡ	Màu sắc	Hình dáng	Quyết định
2	Nhỏ	Đỏ	Nón	Không mua
4	Lớn	Đỏ	Nón	Không mua
6	Lớn	Đỏ	Trụ	Không mua

IF Hình dáng = Nón THEN Quyết định = Không mua



# Ví dụ ILA (tt)

STT	Kích cỡ	Màu sắc	Hình dáng	Quyết định
1	Vừa	Xanh dương	Hộp	Mua
3	Nhỏ	Đỏ	Cầu	Mua
5	Lớn	Xanh lá	Trụ	Mua
7	Lớn	Xanh lá	Cầu	Mua

STT	Kích cỡ	Màu sắc	Hình dáng	Quyết định
6	Lớn	Đỏ	Trụ	Không mua

IF Hình dáng = Nón THEN Quyết định = Không mua

# Ví dụ ILA (tt)

STT	Kích cỡ	Màu sắc	Hình dáng	Quyết định
1	Vừa	Xanh dương	Hộp	Mua
3	Nhỏ	Đỏ	Cầu	Mua
5	Lớn	Xanh lá	Trụ	Mua
7	Lớn	Xanh lá	Cầu	Mua

STT	Kích cỡ	Màu sắc	Hình dáng	Quyết định
6	Lớn	Đỏ	Trụ	Không mua

IF Hình dáng = Nón THEN Quyết định = Không mua

IF Kích cỡ = Lớn AND Màu sắc = Đỏ THEN Quyết định  
= Không mua

# Bài tập

- Cho tập huấn luyện sau. Giả sử “Chơi Tennis” là thuộc tính phân lớp.

Quang cảnh	Nhiệt độ	Độ ẩm	Sức gió	Chơi tennis
Nắng	Nóng	Cao	Yêu	Không
Nắng	Nóng	Cao	Mạnh	Không
Mây	Nóng	Cao	Yêu	Có
Mưa	TB	Cao	Yêu	Có
Mưa	Lạnh	BT	Yêu	Có
Mưa	Lạnh	BT	Mạnh	Không
Mây	Lạnh	BT	Mạnh	Có
Nắng	TB	Cao	Yêu	Không
Nắng	Lạnh	BT	Yêu	Có
Mưa	TB	BT	Yêu	Có
Nắng	TB	BT	Mạnh	Có
Mây	TB	Cao	Mạnh	Có
Mây	Nóng	BT	Yêu	Có
Mưa	TB	Cao	Mạnh	Không

# Bài tập (tt)

- a) Sử dụng lần lượt độ đo Gain, chỉ mục gini để xây dựng cây quyết định. Biến đổi cây thành luật.
- b) Sử dụng phương pháp ILA để xác định luật.
- c) Sử dụng lần lượt các tập luật thu được từ câu (a), (b) để xác định lớp cho mẫu mới.

<b>Quang cảnh</b>	<b>Nhiệt độ</b>	<b>Độ ẩm</b>	<b>Sức gió</b>	<b>Chơi Tennis</b>
<b>Mưa</b>	<b>TB</b>	<b>BT</b>	<b>Mạnh</b>	<b>?</b>
<b>Nắng</b>	<b>TB</b>	<b>Cao</b>	<b>Mạnh</b>	<b>?</b>

# Tóm tắt

---

- Phân lớp là quá trình gán nhãn cho các mẫu.
- Bộ phân lớp được học dựa trên các mẫu đã được gán nhãn sẵn.
- Phương pháp phân lớp dựa trên cây quyết định tìm kiếm thuộc tính “tốt nhất” để đưa vào cây bằng độ đo như Information Gain, Gain Ratio, Gini Index. Vấn đề tỉa cây để vượt qua vấn đề Overfitting
- Phương pháp phân lớp dựa trên luật tập trung vào việc phát sinh luật trực tiếp/gián tiếp từ dữ liệu. Trực tiếp sử dụng hàm Learn-One-Rule và độ đánh giá chất lượng luật FOIL. Gián tiếp sử dụng cây quyết định,...

# Tài liệu tham khảo

---

1. J.Han, M.Kamber, Chương 8 – Classification: Basic Concepts và Chương 9 – Classification: Advanced Methods, cuốn “*Data mining: Basic Concepts and Methods*”, 3<sup>rd</sup> edition
2. J.Han, M.Kamber, J.Pei, Chapter 8, [http://www.cs.uiuc.edu/homes/hanj/cs412/bk3\\_slides/08ClassBasic.ppt](http://www.cs.uiuc.edu/homes/hanj/cs412/bk3_slides/08ClassBasic.ppt)
3. Bing Liu, Chapter 3 – Supervised Learning, <http://www.cs.uic.edu/~liub/teach/cs583-fall-06/CS583-supervised-learning.ppt>
4. Mehmet R. Tolun, Saleh M. Abu-Soud. ***ILA, an inductive learning algorithm for rule extraction.*** ESA 14(3), 4/1998, 361-370

# Hỏi & Đáp

---

