KHAI PHÁ DỮ LIỆU

VO DUC QUANG – VINH UNIVERSITY

QUANGVD@VINHUNI.EDU.VN

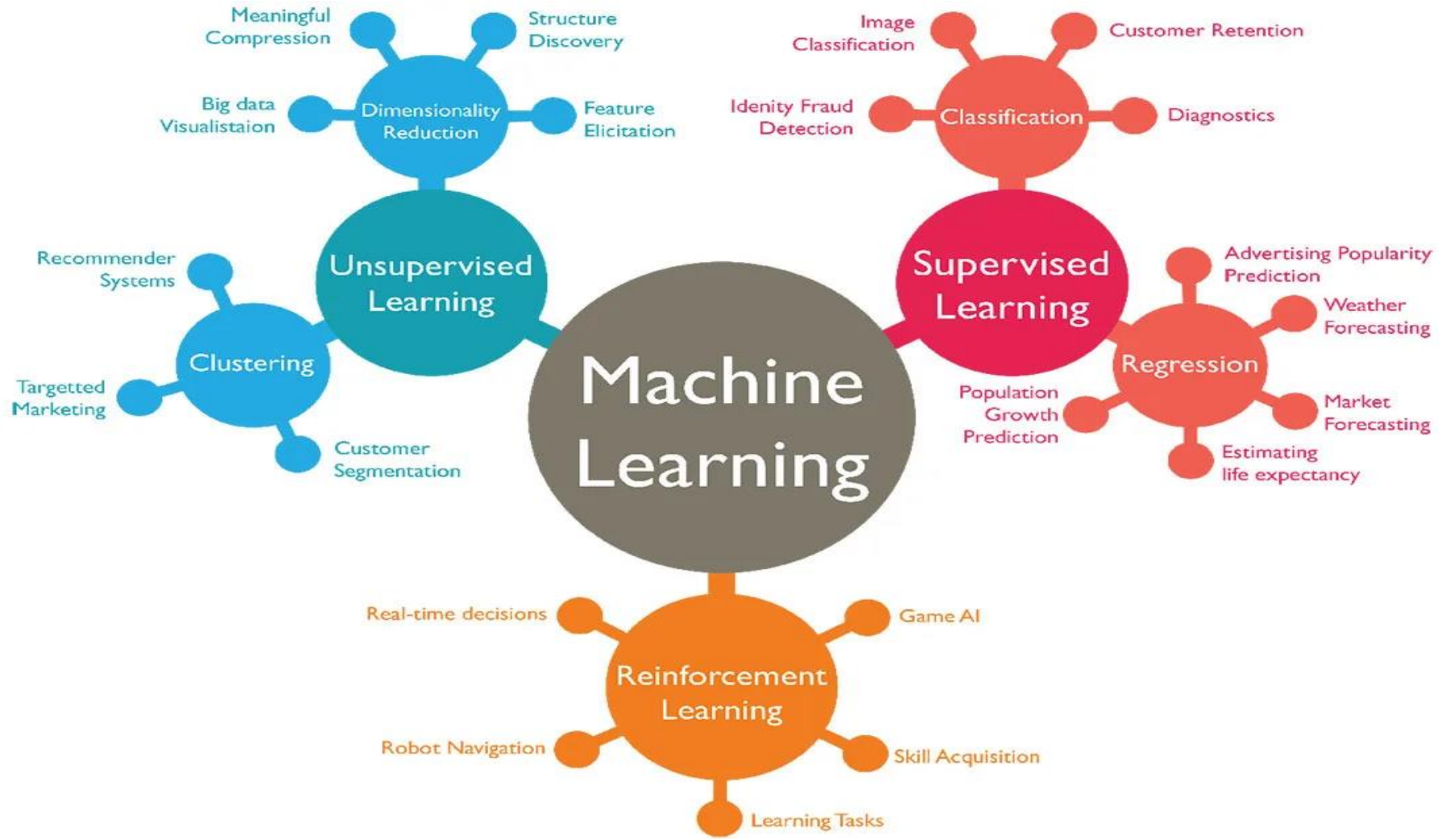
Nội dung

- Chương 1: Tổng quan về Data Mining
- Chương 2: Dữ liệu và tiền xử lý dữ liệu
- Chương 3: Bài toán phân lớp dữ liệu
- Chương 4: Bài toán phân cụm dữ liệu
- Chương 5: Khai phá luật kết hợp

Chương 3 – Phân lớp (Classification)

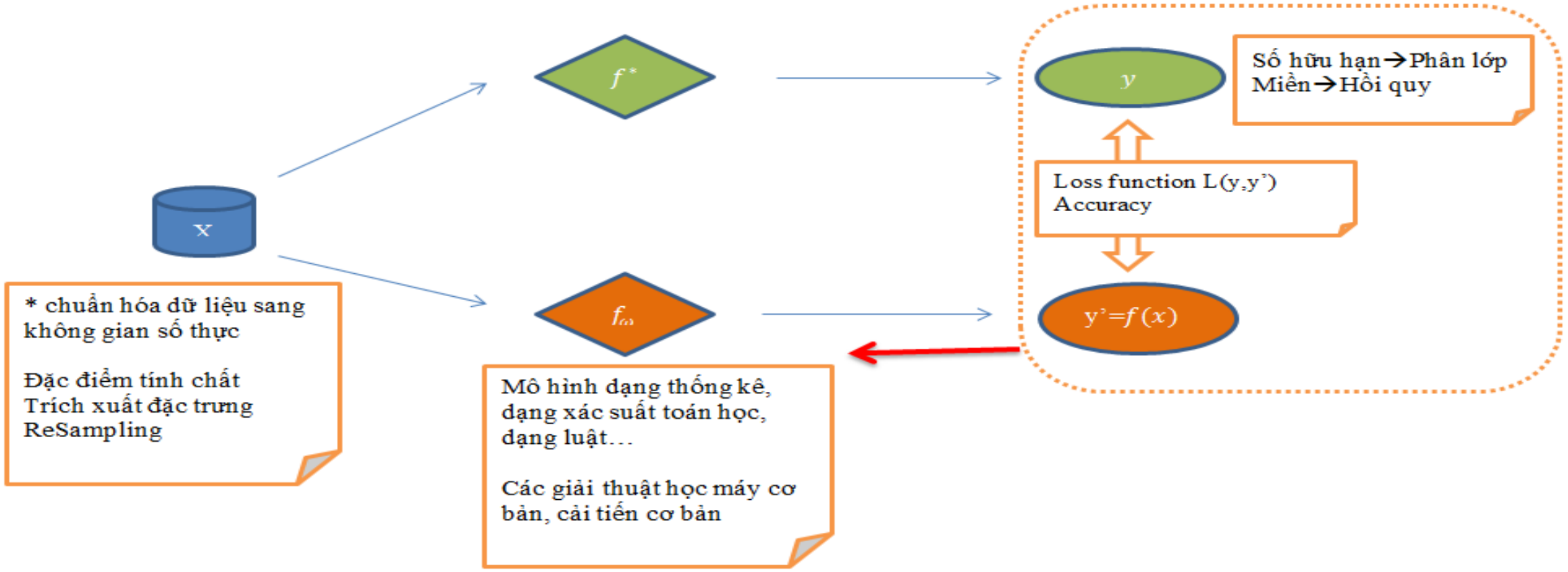
- Bài toán Phân lớp
- Phân lớp sử dụng Cây quyết định
- Một số giải thuật khác
  - Naïve Bayes
  - KNN
- Đánh giá mô hình phân lớp

Tổng quan Machine Learning



Mô hình Phân lớp

- **Training set** (Tập mẫu huấn luyện): {(x1,y1); (x2,y2), …, (xN,yM)}
- M: Số nhãn lớp, N: số mẫu dữ liệu

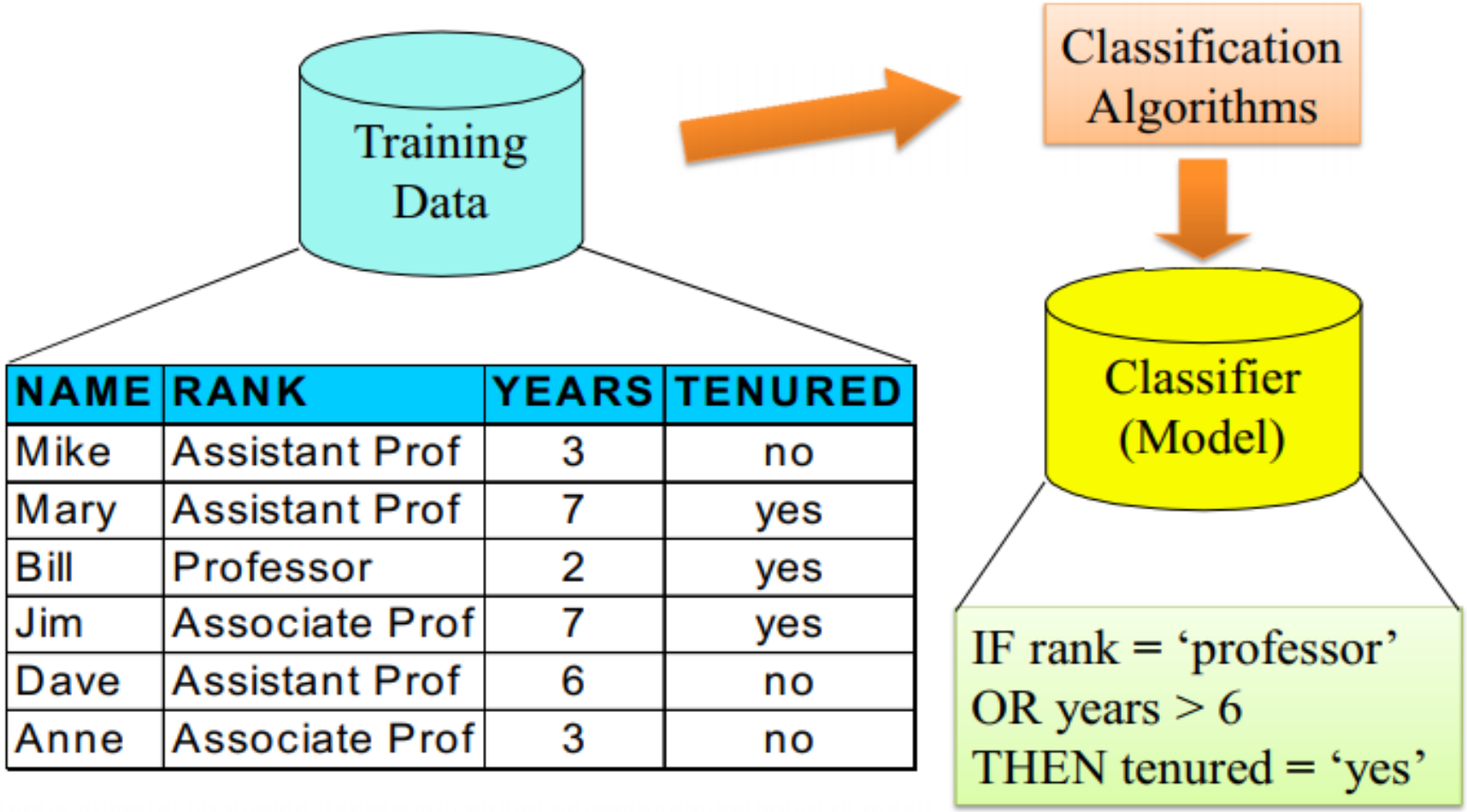


Phân lớp

- Tập dữ liệu
  - Tập dữ liệu huấn luyện? S = {S1, S2, …, Sm} → Tập dữ liệu S có m mẫu dữ liệu
  - Nhãn lớp là gì? C = {C1, C2, …, Cn} → Tập dữ liệu S có n nhãn lớp
  - Thuộc tính là gì? Si = {Xi1, Xi2, …, Xik, Cj} → Mẫu dữ liệu Si có k thuộc tính và có nhãn lớp Cj
- Quá trình huấn luyện (Training)
  - Tập dữ liệu huấn luyện
  - Giải thuật huấn luyện
  - Tạo ra mô hình phân lớp (model) F
- Quá trình kiểm tra (Testing)
  - F(Xi) ??? Cj
- Đánh giá mô hình

Phân lớp

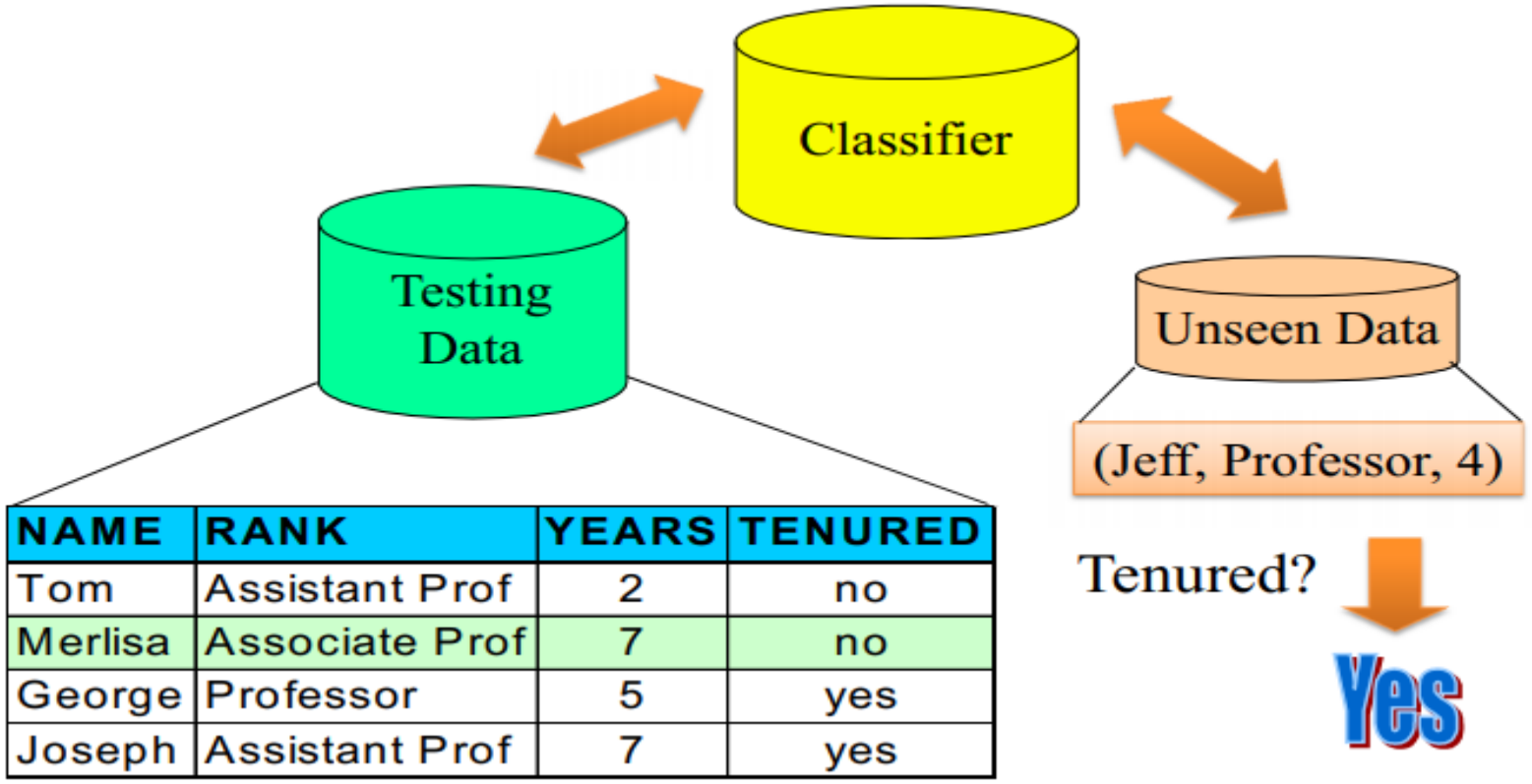
- Ví dụ: Huấn luyện mô hình phân lớp dạng luật



NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

Phân lớp

- Ví dụ: Sử dụng mô hình phân lớp (đã huấn luyện)



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

Phân lớp

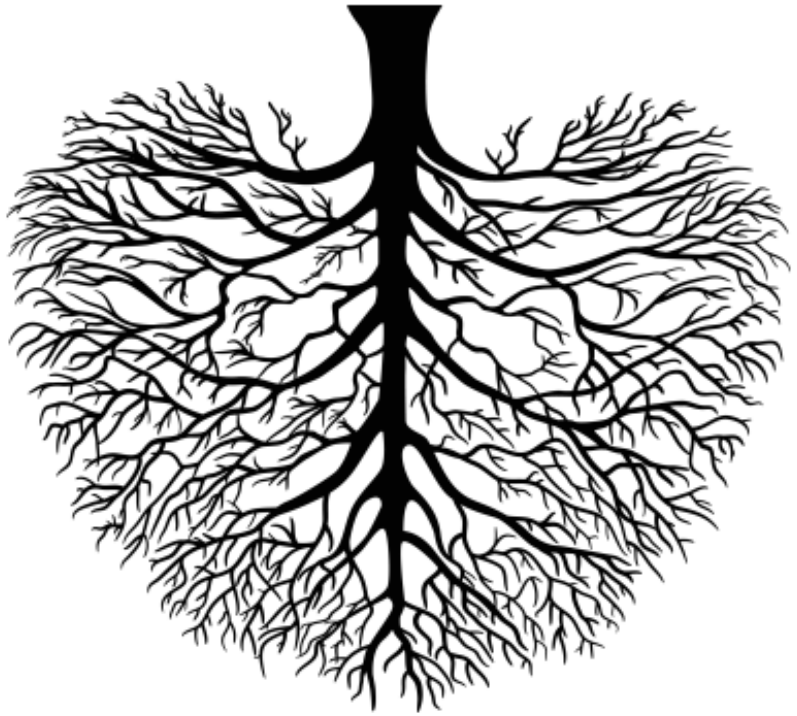
- Phân biệt Phân lớp và Dự báo
  - Phân lớp:
    - Sử dụng tập dữ liệu có “đáp án” để huấn luyện mô hình
    - Mô hình này có thể dự đoán các nhãn lớp dựa trên các thuộc tính của dữ liệu
  - Dự báo:
    - Đầu ra mô hình phân lớp là các giá trị liên tục
    - Dự đoán các thông tin giá trị chưa được biết tới

Phân lớp - Một số giải thuật

- Cây quyết định (Decision Tree)
- Mô hình dạng luật (Rule-based methods)
- Mạng nơ-ron (Neural Networks)
- SVM (Support Vector Machine)
- KNN
- Naïve Bayes
- Boosting: ADA Boost, XGBoost…
- Bagging: Rừng cây ngẫu nhiên (Random forest)
- …

Cây quyết định

- Dùng một mô hình cây để biểu diễn một hàm chức năng
- Mô hình dạng luật (rule) IF-THEN



Cây quyết định

- Ví dụ:

CustomerID	Debt	Income	Employment	Risk
1	high	high	self-employed	high
2	high	high	employed	high
3	high	low	employed	high
4	low	low	employed	low
5	low	low	self-employed	high
6	low	high	employed	low

Mô hình đơn giản:

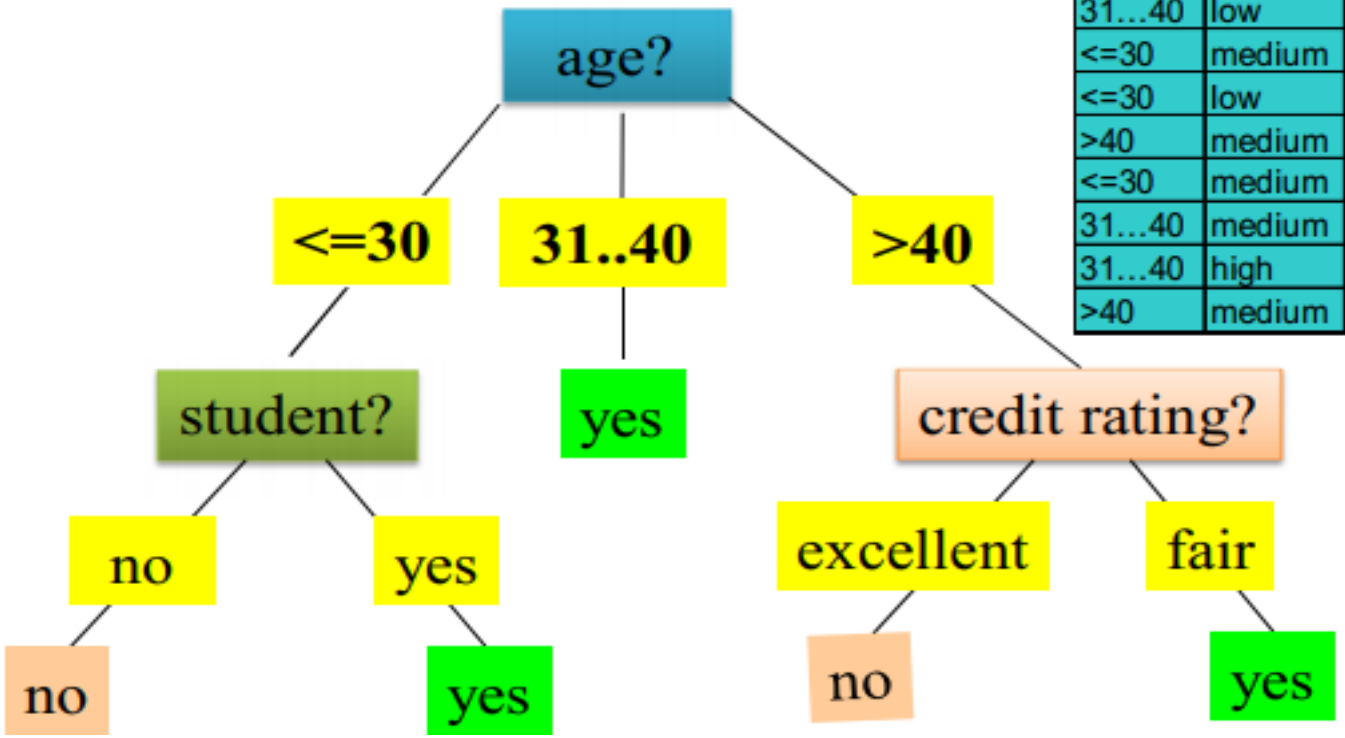
```
if Debt = high then Risk = high
if Debt = low && Employment = employed then
   Risk = low
if Debt = low && Employment = self-employed
then Risk = high
```

Cây quyết định

- Ví dụ
  - Tập dữ liệu huấn luyện:

    buys_computer
  - Xây dựng cây:

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



Cây quyết định

- Ví dụ:
- Nhận xét bộ dữ liệu
  - Thuộc tính?
  - Nhãn lớp?
  - Bắt đầu từ thuộc tính nào?

Day	Outlook	Temperature	Humidity	Wind	Playing_Tennis
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	cloudy	hot	high	weak	yes
4	rainy	mild	high	weak	yes
5	rainy	cold	normal	weak	yes
6	rainy	cold	normal	strong	no
7	cloudy	cold	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cold	normal	weak	yes
10	rainy	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	cloudy	mild	high	strong	yes
13	cloudy	hot	normal	weak	yes
14	rainy	mild	high	strong	no

Cây quyết định

- 04 thuộc tính
- 02 nhãn lớp (yes/no)
- 14 dòng dữ liệu (số mẫu dữ liệu): 09 {yes} và 05 {no}
- Phân tích từng thuộc tính để xem thuộc tính nào là quan trọng, có ý nghĩa hơn trong việc phân lớp (đưa ra kết quả dự đoán)?
  - Wind {Strong, weak}
  - Temperature {cold, mild, hot}
  - Humidity {high, normal}
  - Outlook {sunny, cloudy, rainy}

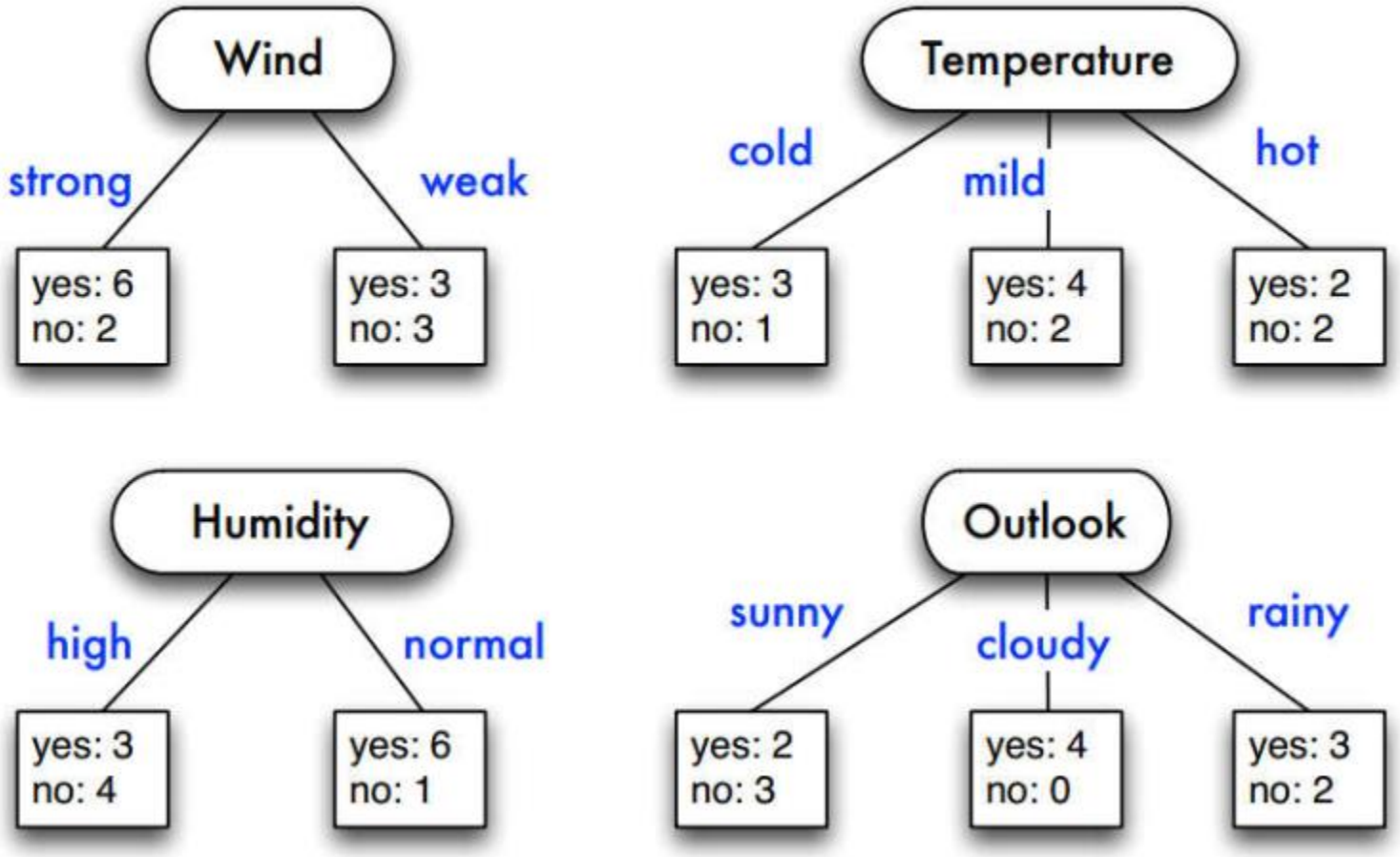
Cây quyết định

(Outlook=Overcast, Temperature=Hot, Humidity=High, Wind=Weak)
→ Yes

(Outlook=Rain, Temperature=Mild, Humidity=High, Wind=Strong)
→ No

(Outlook=Sunny, Temperature=Hot, Humidity=High, Wind=Strong)
→ No

Cây quyết định



Day	Outlook	Temperature	Humidity	Wind	Playing_Tennis
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	cloudy	hot	high	weak	yes
4	rainy	mild	high	weak	yes
5	rainy	cold	normal	weak	yes
6	rainy	cold	normal	strong	no
7	cloudy	cold	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cold	normal	weak	yes
10	rainy	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	cloudy	mild	high	strong	yes
13	cloudy	hot	normal	weak	yes
14	rainy	mild	high	strong	no

Cây quyết định

- Làm thế nào?

Cây quyết định

- Tìm một độ đo để đánh giá mức độ phân hoạch rõ ràng nhãn của các mẫu dữ liệu
- Lựa chọn thuộc tính để có một phân hoạch đồng nhất cực đại
- **Entropy!**
- **Information Gain, GINI index**

Cây quyết định

- Các định nghĩa
  - D : Tập dữ liệu huấn luyện
  - C = {C1, C2, … Cm} Tập nhãn lớp
  - $D = D_1 \cup D_2 \cup \cdots \cup D_t$. D_i là các phân hoạch trên D. $D_i \cap D_j = \emptyset$
  - **Entropy của tập dữ liệu**, ***Info(D):*** là lượng thông tin cần để phân loại một phần tử trong tập dữ liệu D
  - p_i : xác suất của một phần tử bất kỳ trong D thuộc về lớp Ci

$$p_i = \frac{|D_i|}{|D|}$$

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

Cây quyết định

- Entropy của tập dữ liệu ứng với thộc tính
  - Là lượng thông tin cần để phân loại một phần tử trong tập dữ liệu D dựa trên thuộc tính A, ký hiệu $\boldsymbol{Info_A(D)}$
  - Thuộc tính A dùng để tách D thành **t** phân hoạch là D1, D2,…, Dt
  - Mỗi phân hoạch Dj có |Dj| phần tử, với 1<= j<=t
  - Lượng thông tin này cho biết mức độ trùng lặp giữa các phân hoạch

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

  - **Mong đợi $\boldsymbol{Info_A(D)}$càng nhỏ càng tốt**

Cây quyết định

- Độ lợi thông tin (Information Gain)
  - Tối thiểu hóa lượng thông tin cần thiết để phân lớp các mẫu dữ liệu
  - Độ lợi thông tin ứng với thuộc tính A, Ký hiệu **Gain(A),** là độ sai lệch giữa Entropy ban đầu của tập dữ liệu (trước khi phân hoạch) và Entropy của dữ liệu với thuộc tính A (sau khi phân hoạch bởi A)

$$\boldsymbol{Gain(A) = Info(D) - Info_A(D)}$$

Cần chọn thuộc tính có độ lợi thông tin **Gain(A)** lớn nhất → **Nút gốc**

Cây quyết định

- Ví dụ 2

  - $|D| = 14$;
  - $C = \{yes, no\}$
  - $|C_{yes}| = 9$; $|C_{no}| = 5$

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Cây quyết định

- Tính Entropy

$$Info(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940 \text{ bits.}$$

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

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

- Xét thuộc tính **age**
  - youth
    - yes: 2; no: 3
  - middle_age
    - yes: 4; no: 0
  - senior
    - yes: 3; no: 2

$$Info_{age}(D) = \frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right) + \frac{4}{14} \times \left(-\frac{4}{4}\log_2\frac{4}{4}\right) + \frac{5}{14} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right) = 0.694 \text{ bits.}$$

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246 \text{ bits}$$

Cây quyết định

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

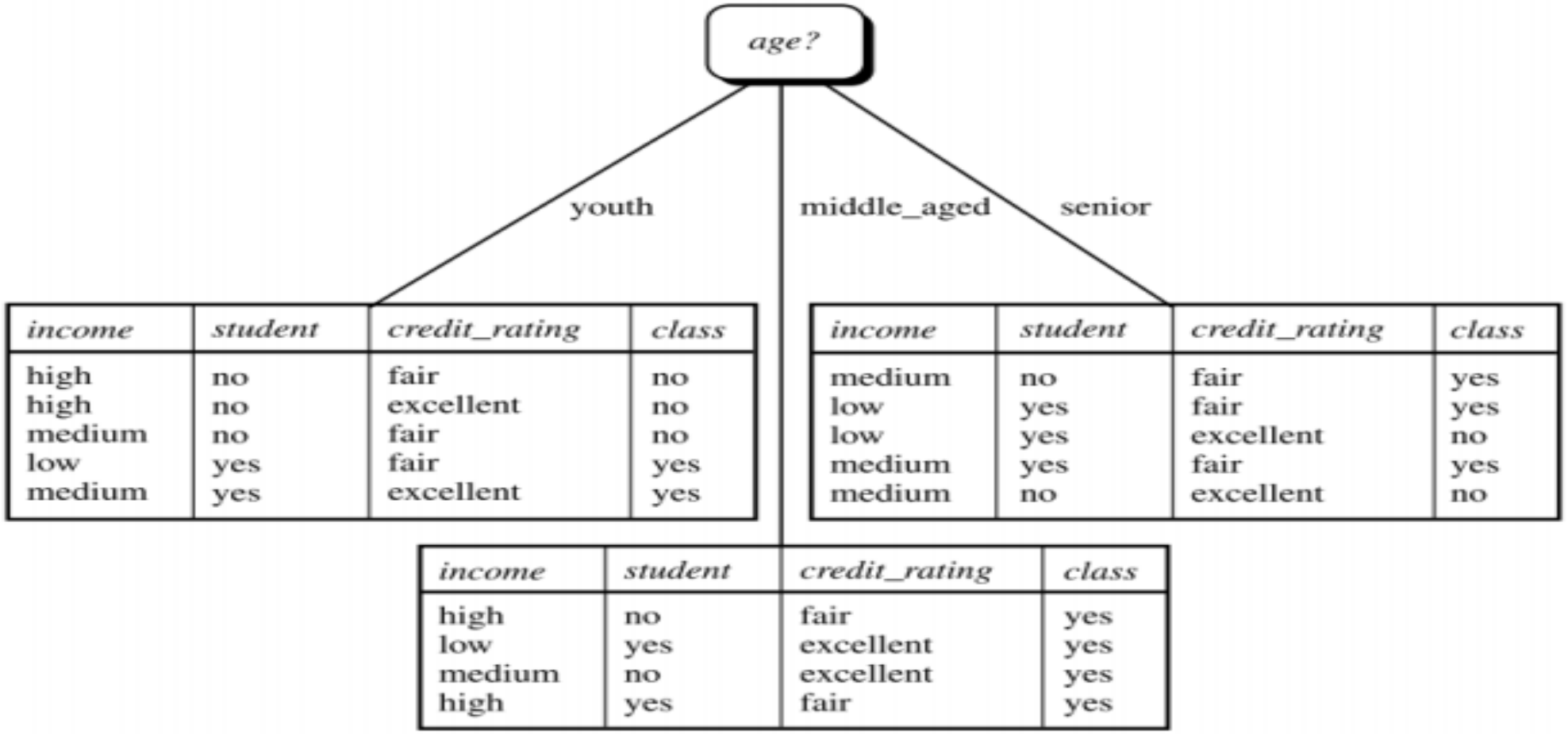
Tính toán tương tự ta có

- Gain(income) = 0.029 bits
- Gain(student) = 0.151 bits
- Gain(credit_rating) = 0.048 bits

Vì thuộc tính *age* có độ lợi thông tin lớn nhất (0.246 bits) nên *age* là thuộc tính được chọn để phân tách (rẽ nhánh)

Cây quyết định

- Chọn nút gốc



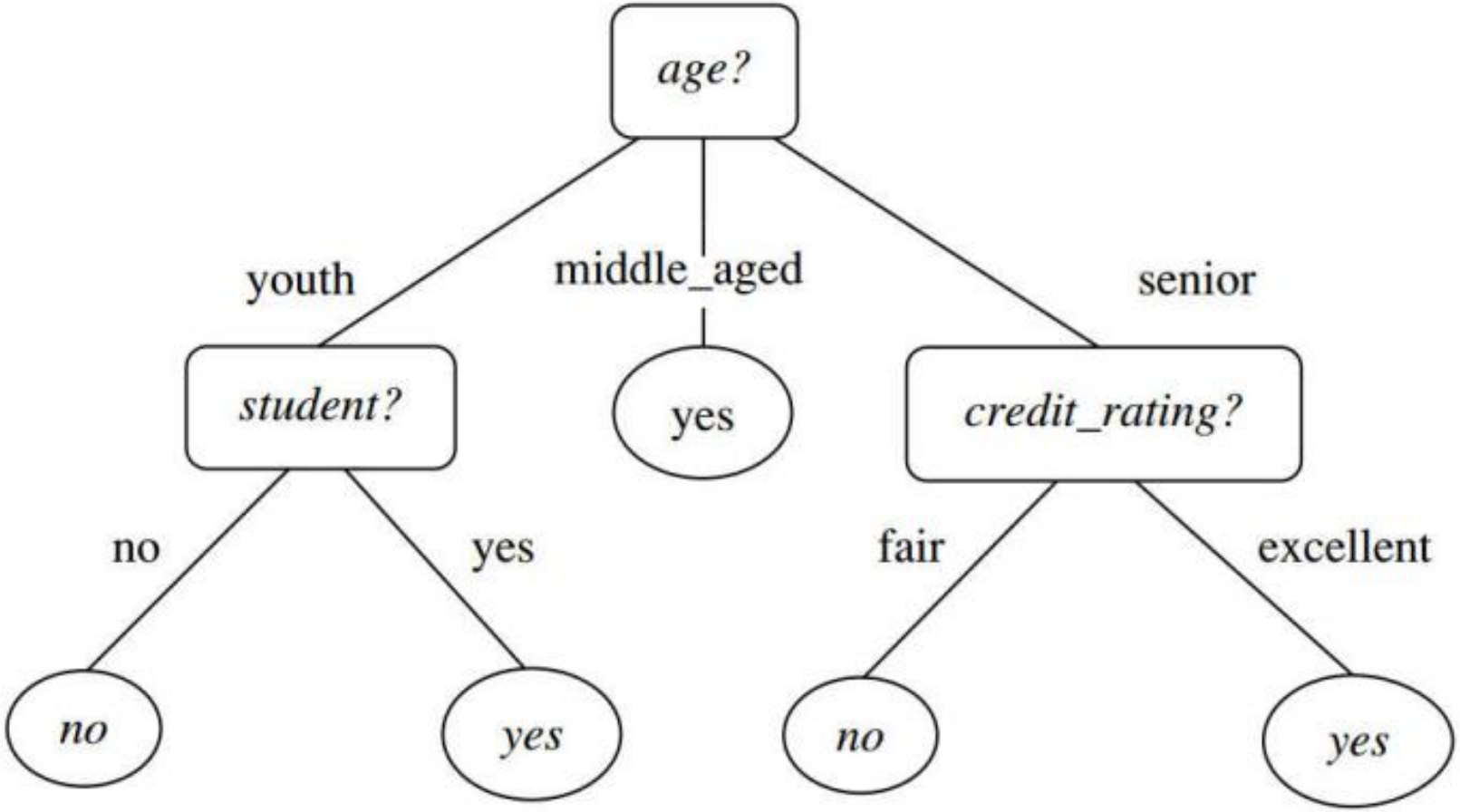
income	*student*	*credit_rating*	*class*
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	*student*	*credit_rating*	*class*
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

income	*student*	*credit_rating*	*class*
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

Cây quyết định

- Hình thành cây



Bài tập

- 1. Dựng cây với dữ liệu Ví dụ 1 (Slide x)
- 2. **Bài tập**: Cho tập dữ liệu *D* như sau, nếu tính độ lợi thông tin dựa trên Entropy thì thuật toán cây quyết định sẽ chọn thuộc tính nào để rẽ nhánh?

TT	Màu tóc	Chiều cao	Cân nặng	Dùng thuốc	Kết quả
1	Đen	Tầm thước	Nhẹ	Không	Bị rám
2	Đen	Cao	Vừa phải	Có	Không
3	Râm	Thấp	Vừa phải	Có	Không
4	Đen	Thấp	Vừa phải	Không	Bị rám
5	Bạc	Tầm thước	Nặng	Không	Bị rám
6	Râm	Cao	Nặng	Không	Không
7	Râm	Tầm thước	Nặng	Không	Không
8	Đen	Thấp	Nhẹ	Có	Không

Thảo luận

- Demo code

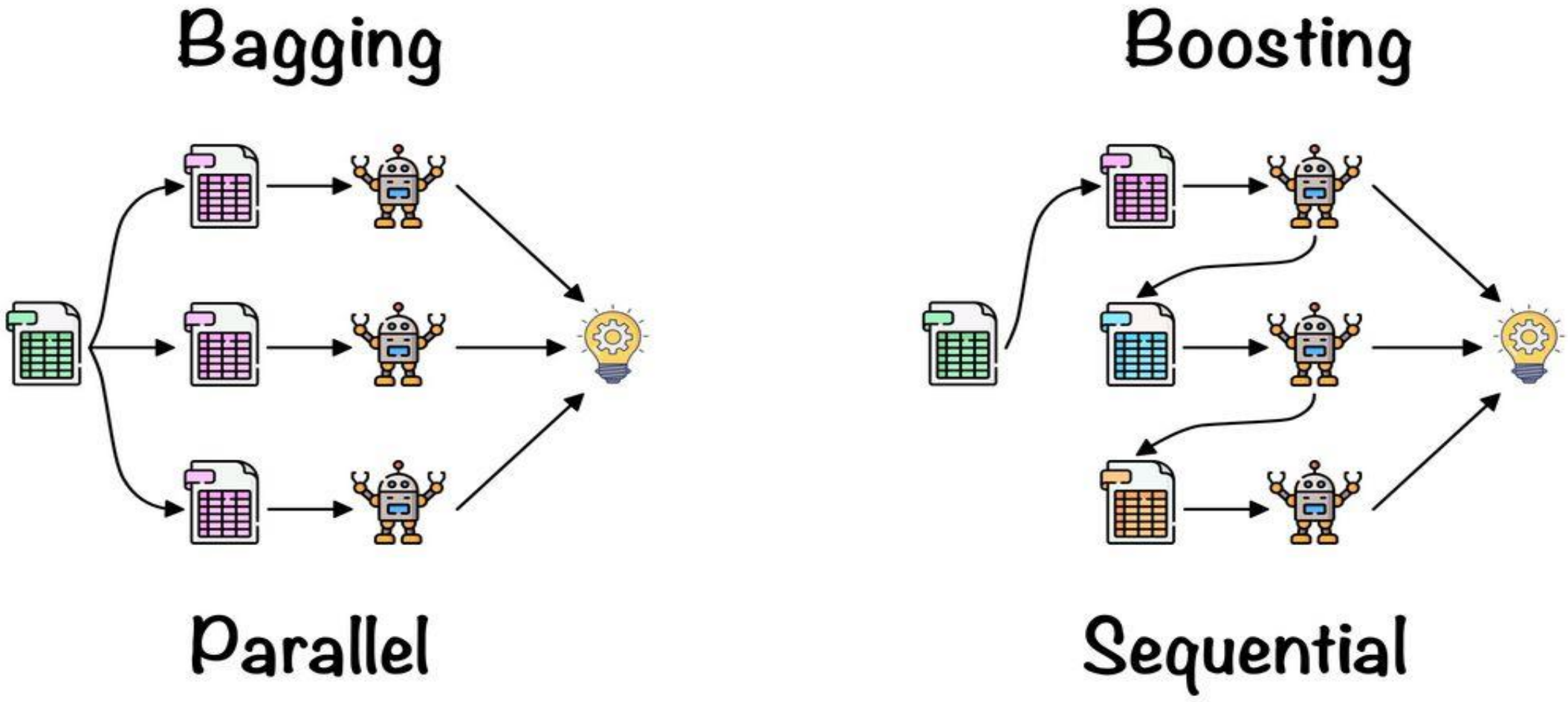
Mở rộng vấn đề

- "Một cây làm chẳng lên non,Ba cây chụm lại lên hòn núi cao“ ????
- Hỏi ý kiến khán giả? Hỏi tổ tư vấn tại chỗ trong “Ai là triệu phú”
- …
- → Học kết hợp (tổng hợp)

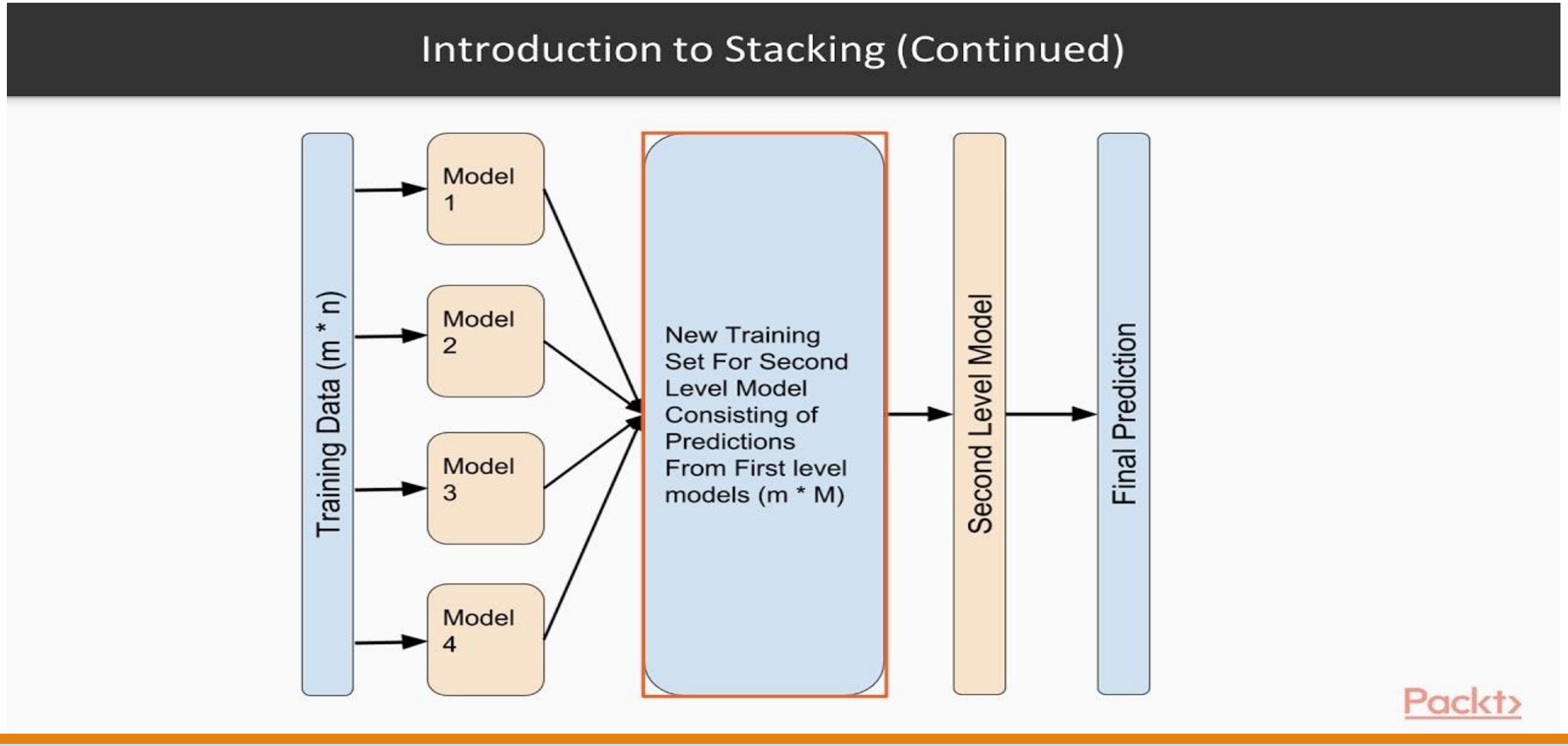
Ensemble Learning

- Ensemble giống như một cuộc bỏ phiếu giữa các model với nhau cho một vấn đề nào đó
- Ensemble Learning là phương pháp tổng hợp kết quả dự đoán của nhiều model thành thành model cuối cùng. Giúp nâng cao tính tổng quát của model Machine Learning
- Các model trong ensemble learning càng độc lập thì độ chính xác của model tổng càng tốt
- Có 3 kiểu kết hợp chính:
  - Boosting (tuần tự)
  - Bagging (song song)
  - Stacking (xếp chồng)

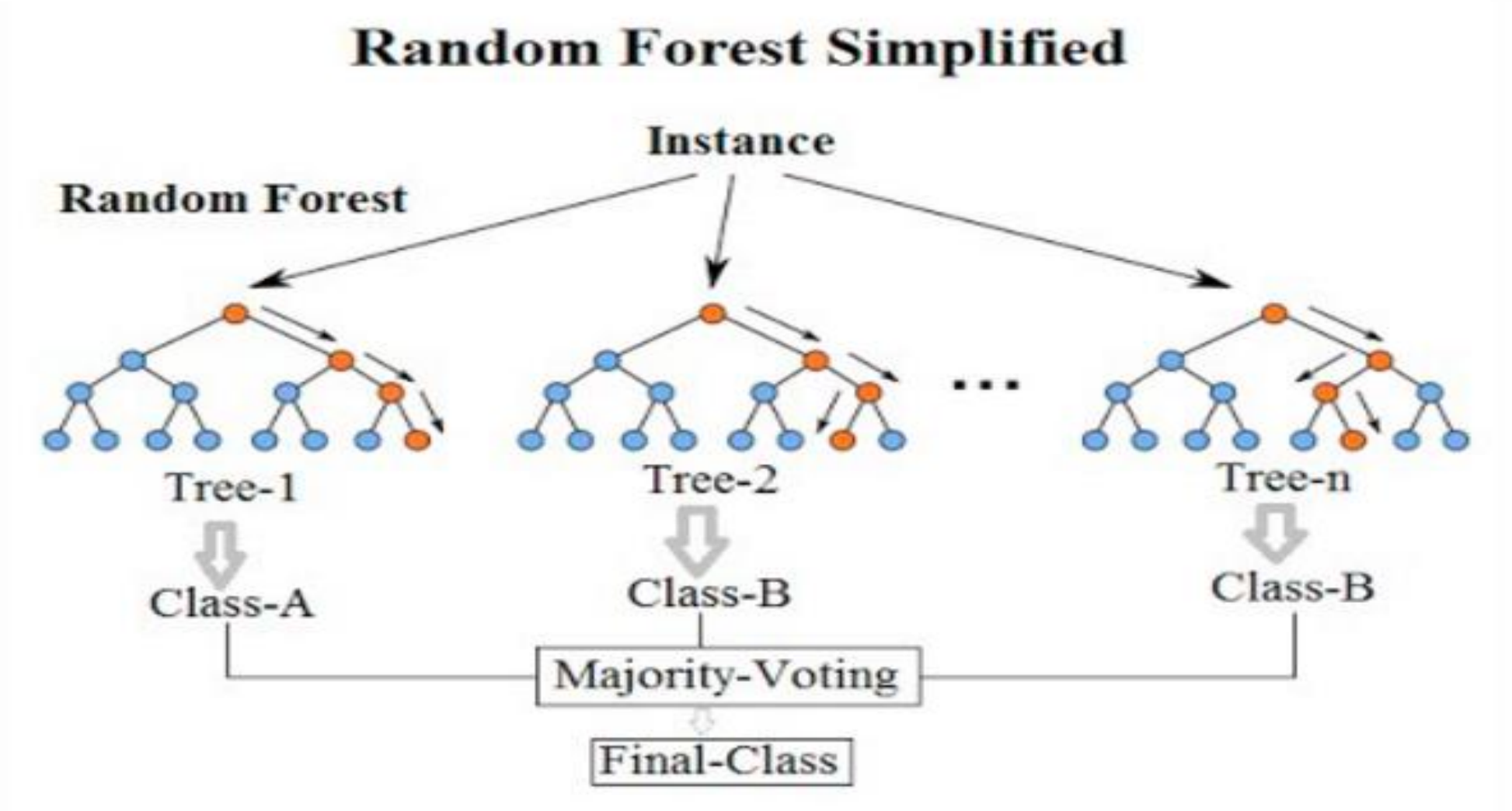
Ensemble Learning



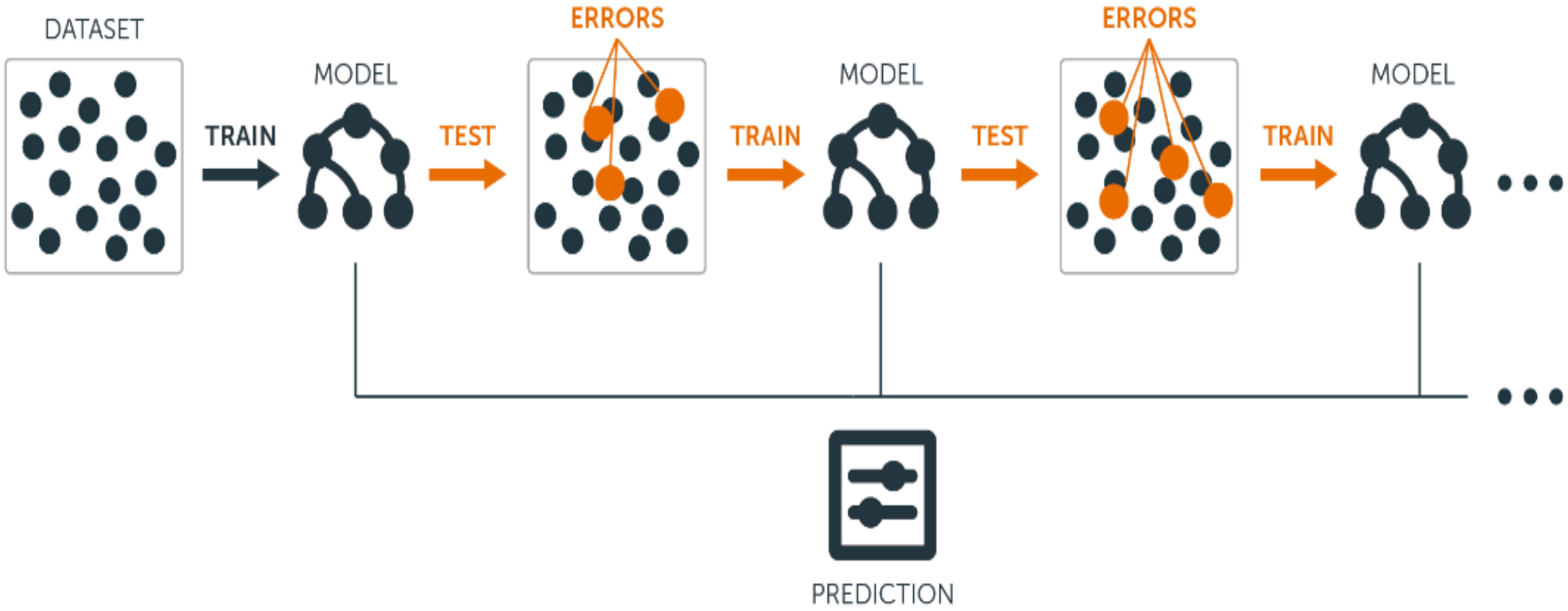
Ensemble Learning



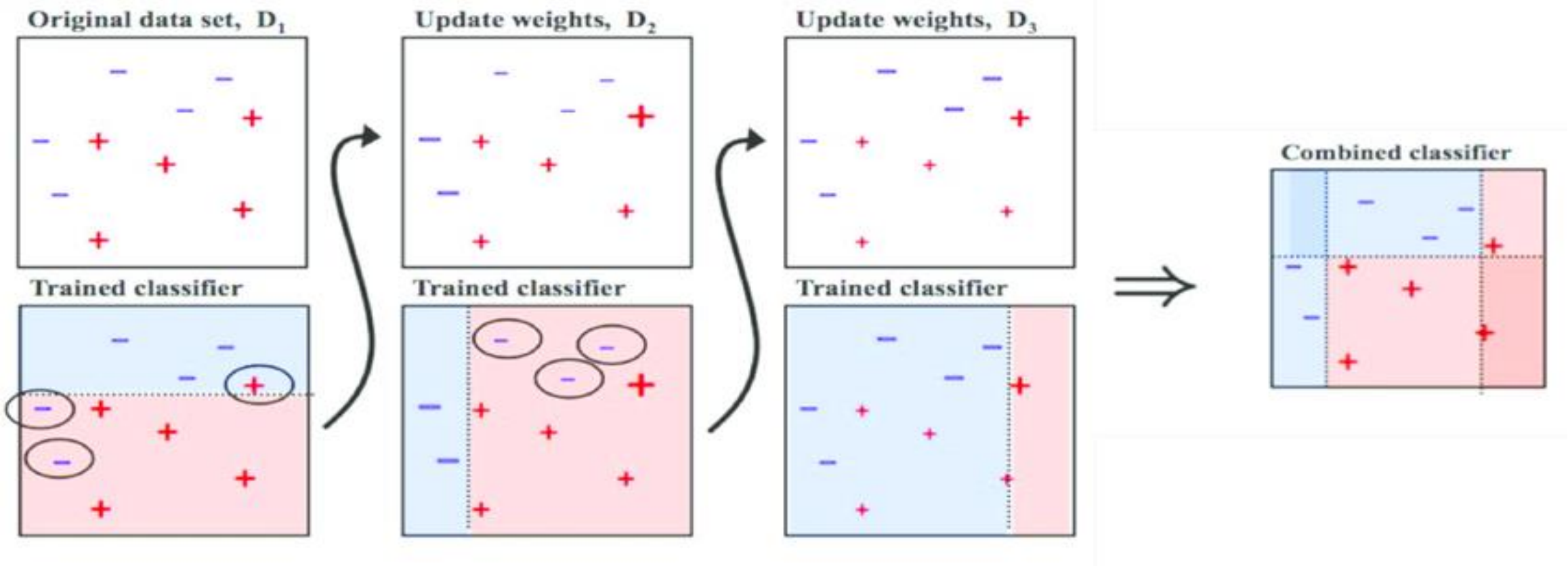
Bagging


Random Forest Simplified
Instance
Random Forest
Tree-1
Tree-2
Tree-n
Class-A
Class-B
Class-B
Majority-Voting
Final-Class

BOOSTING


DATASET
TRAIN
MODEL
TEST
ERRORS
TRAIN
MODEL
TEST
ERRORS
TRAIN
MODEL
PREDICTION

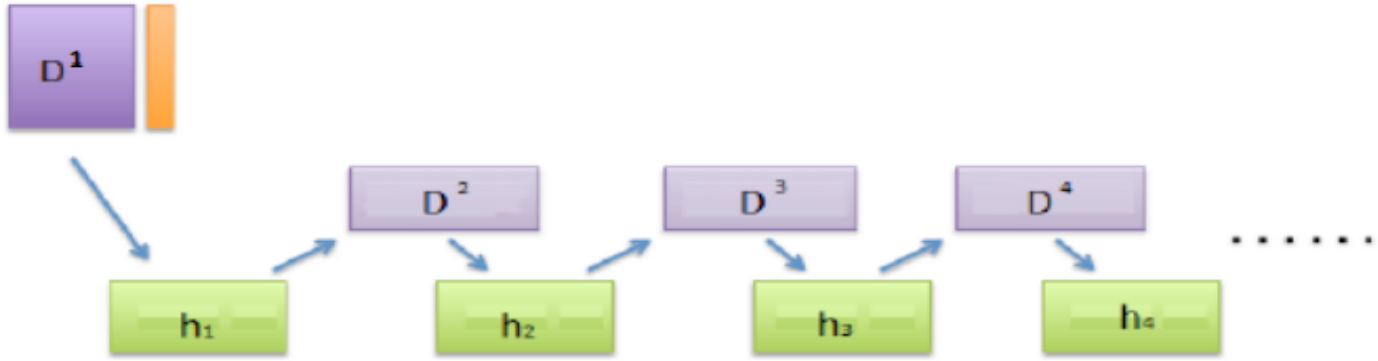
BOOSTING


Original data set, D_1
Update weights, D_2
Update weights, D_3
Trained classifier
Trained classifier
Trained classifier
Combined classifier

AdaBoost: Algorithm Outline

Algorithm 1: AdaBoost Sketch

Input: Training Data $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, Number of rounds M.
Training:
Define a weight distribution over the examples $D_i^1 = \frac{1}{N}$, for $i = 1, 2, \ldots, N$.
for round $j = 1$ to M **do**
    Build a model h_j from the training set using distribution D^j.
    Update D^{j+1} from D^j:
        Increase weights of examples misclasified by h_j.
        Decrease weights of examples correctly classified by h_j.
end for
Prediction: For a new example x', output the weighted (confidence-rated) majority vote of the models $\{h_1, h_2, \ldots, h_M\}$.



AdaBoost: Algorithm

Algorithm 2 AdaBoost

Input: Training Data $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$, Number of rounds M.
Training:
$D_i^1 = \frac{1}{N}$, for $i = 1, 2, \ldots, N$.
for $j = 1$ to M **do**
Define $\epsilon_j = \sum_{i:h_j(x_i) \neq y_i} D_i^j$.

ε_j: weighted error of the j-th model

Obtain a hypothesis h_j that minimizes ϵ_j and satisfies the condition $\epsilon_j < \frac{1}{2}$.
$\alpha_j = \frac{1}{2} \log\left(\frac{1-\epsilon_j}{\epsilon_j}\right)$.

α_j: ``confidence'' of the j-th model

$D_i^{j+1} = e^{-y_i h_j(x_i) \alpha_j} D_i^j$.
$D_i^{j+1} = \frac{D_i^{j+1}}{\sum_{i=1}^N D_i^{j+1}}$.

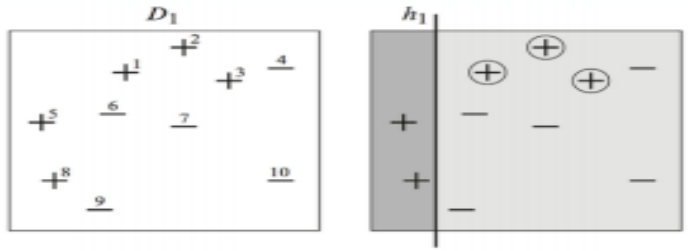
Update weights for next iteration

After normalization we have: $\sum_{i=1}^{N} D_i^{j+1} = 1$

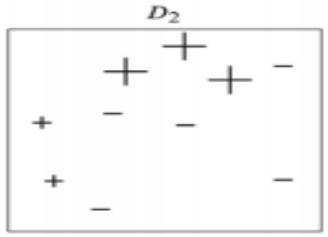
end for
Prediction: $H(\boldsymbol{x}') = sign\left[\sum_{j=1}^M \alpha_j h_j(\boldsymbol{x}')\right]$.

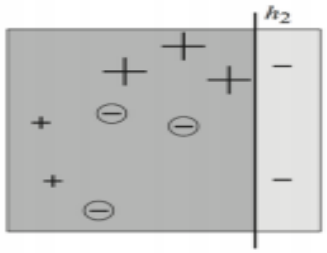
$y \in \{-1, 1\}$

Adaboost: an example

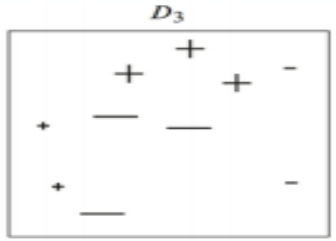


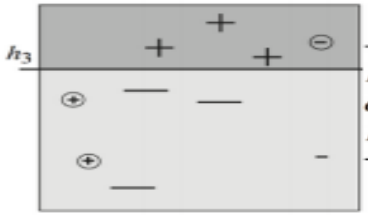
	1	2	3	4	5	6	7	8	9	10	
$D_1(i)$	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	$\epsilon_1 = 0.30, \alpha_1 \approx 0.42$
$e^{-\alpha_1 y_i h_1(x_i)}$	1.53	1.53	1.53	0.65	0.65	0.65	0.65	0.65	0.65	0.65	
$D_1(i)\, e^{-\alpha_1 y_i h_1(x_i)}$	0.15	0.15	0.15	0.07	0.07	0.07	0.07	0.07	0.07	0.07	$Z_1 \approx 0.92$



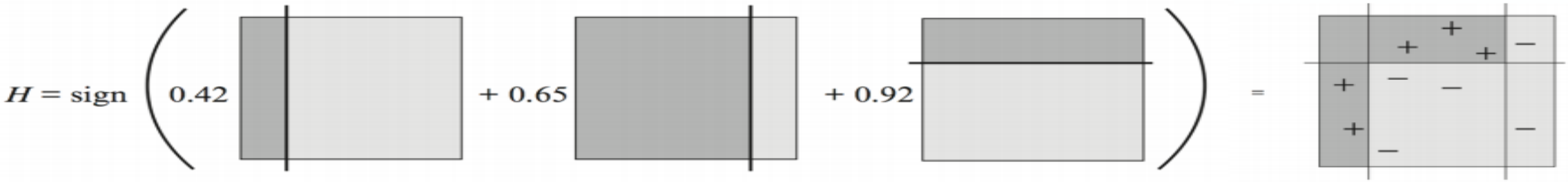


$D_2(i)$	0.17	0.17	0.17	0.07	0.07	0.07	0.07	0.07	0.07	0.07	$\epsilon_2 \approx 0.21, \alpha_2 \approx 0.65$
$e^{-\alpha_2 y_i h_2(x_i)}$	0.52	0.52	0.52	0.52	0.52	1.91	1.91	0.52	1.91	0.52	
$D_2(i)\, e^{-\alpha_2 y_i h_2(x_i)}$	0.09	0.09	0.09	0.04	0.04	0.14	0.14	0.04	0.14	0.04	$Z_2 \approx 0.82$





$D_3(i)$	0.11	0.11	0.11	0.05	0.05	0.17	0.17	0.05	0.17	0.05	$\epsilon_3 \approx 0.14, \alpha_3 \approx 0.92$
$e^{-\alpha_3 y_i h_3(x_i)}$	0.40	0.40	0.40	2.52	2.52	0.40	0.40	2.52	0.40	0.40	
$D_3(i)\, e^{-\alpha_3 y_i h_3(x_i)}$	0.04	0.04	0.04	0.11	0.11	0.07	0.07	0.11	0.07	0.02	$Z_3 \approx 0.69$



Thảo luận

- Demo code