

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN

Tài liệu bài giảng:

KHAI THÁC DỮ LIỆU – IS252

Chương 6:

Phân lớp dữ liệu

ThS. Dương Phi Long – Email: longdp@uit.edu.vn

Supervised vs. Unsupervised Learning

- Supervised Learning

- Supervision: Dữ liệu huấn luyện (quan sát, đo lường, ...) được kèm theo nhãn lớp
- Dữ liệu mới được phân lớp dựa trên tập huấn luyện (classification)

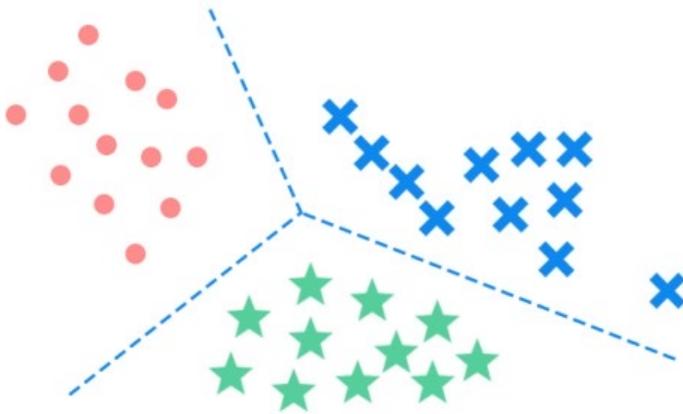
- Unsupervised Learning

- Nhãn lớp của dữ liệu huấn luyện không xác định
- Đưa ra một tập hợp các phép đo, quan sát, ... với mục đích thiết lập sự tồn tại của các lớp hoặc cụm trong dữ liệu (clustering)



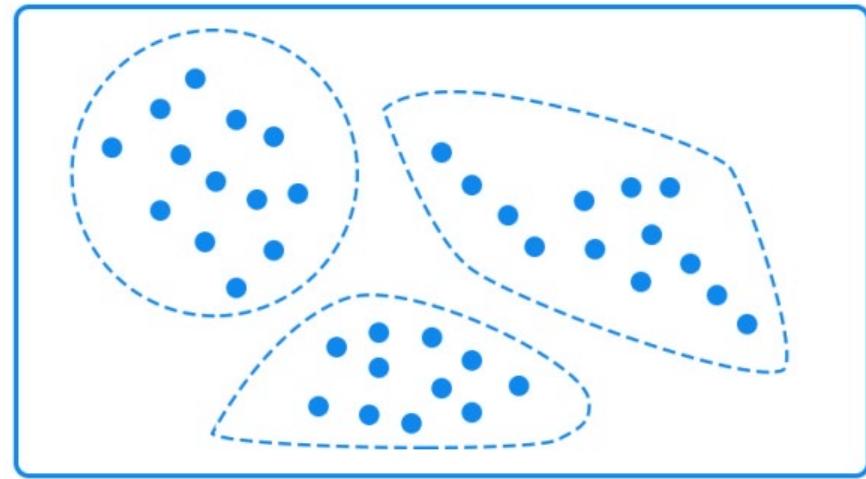
Supervised vs. Unsupervised Learning

Classification



Supervised learning

Clustering

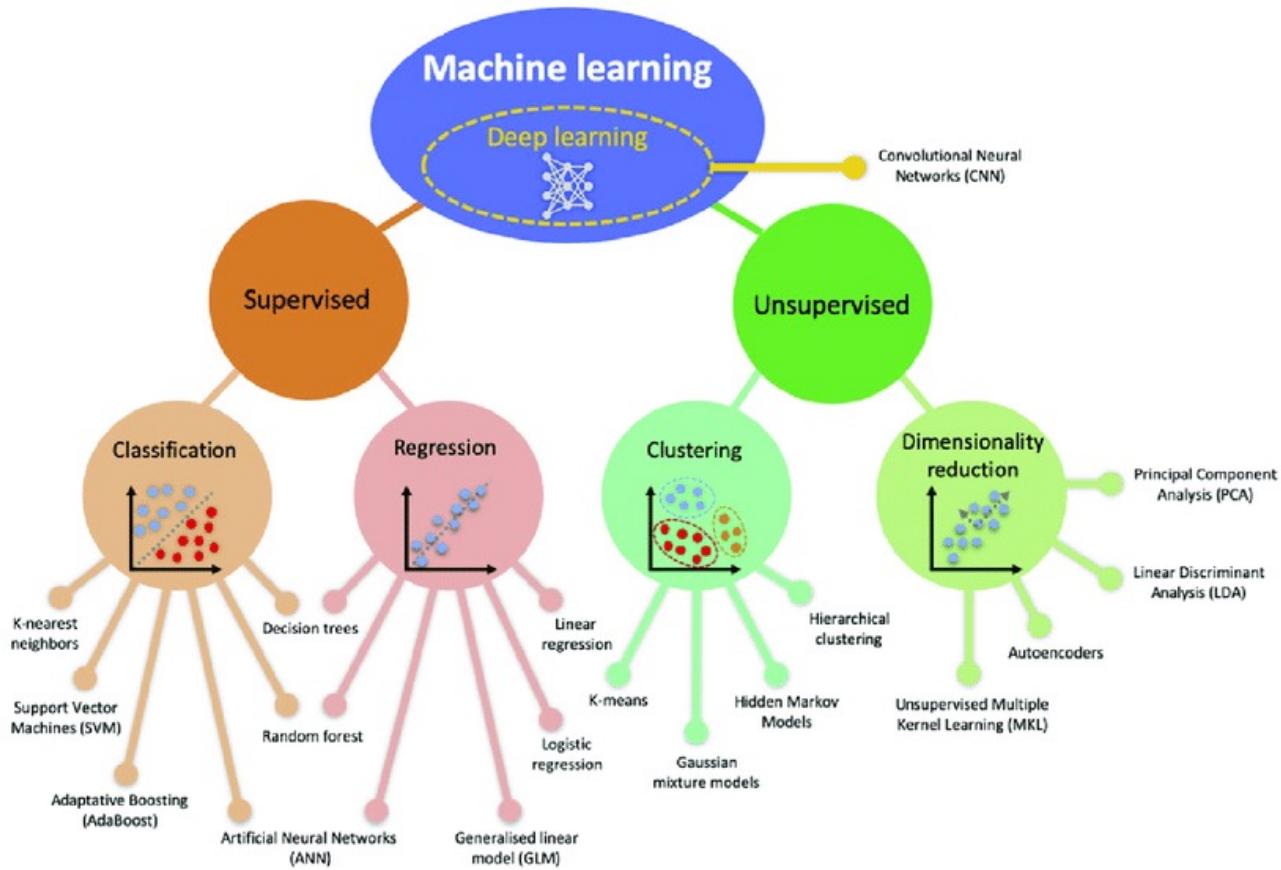


Unsupervised learning



Supervised vs Unsupervised vs Semi-Supervised learning

	Overview	Process	Subtypes	Examples
Supervised learning	Majority of algorithms. Machine is trained using well-labeled data; inputs and outputs are matched.	Mapping function takes inputs and matches to outputs, creating a target function	Classification Regression	Decision tree, Random forest, SVM, K-NN, Neural network, Linear regression, Logistic regression, ...
Unsupervised learning	Unlabeled data (inputs only) is analyzed. Learning happens without supervision	Inputs are used to create a model of the data	Clustering Association Dimensionality reduction	K-Means, C-Means, Hierarchy, Gaussian Mixture Apriori, FP-Growth PCA, LDA,...
Semi-Supervised learning	Some data is labeled, some not. Goal: better results than labeled data alone. Good for real world data.	Combination of above processes	All the above	Self-training, Mixture models, Semi-supervised SVM,...



NỘI DUNG BÀI HỌC

01

Tổng quan về phân lớp dữ liệu

02

Phương pháp dựa trên Cây quyết định

03

Phương pháp dựa trên Luật

04

Phương pháp Naïve Bayes

05

Phương pháp dựa trên thể hiện

NỘI DUNG BÀI HỌC

06

07

08

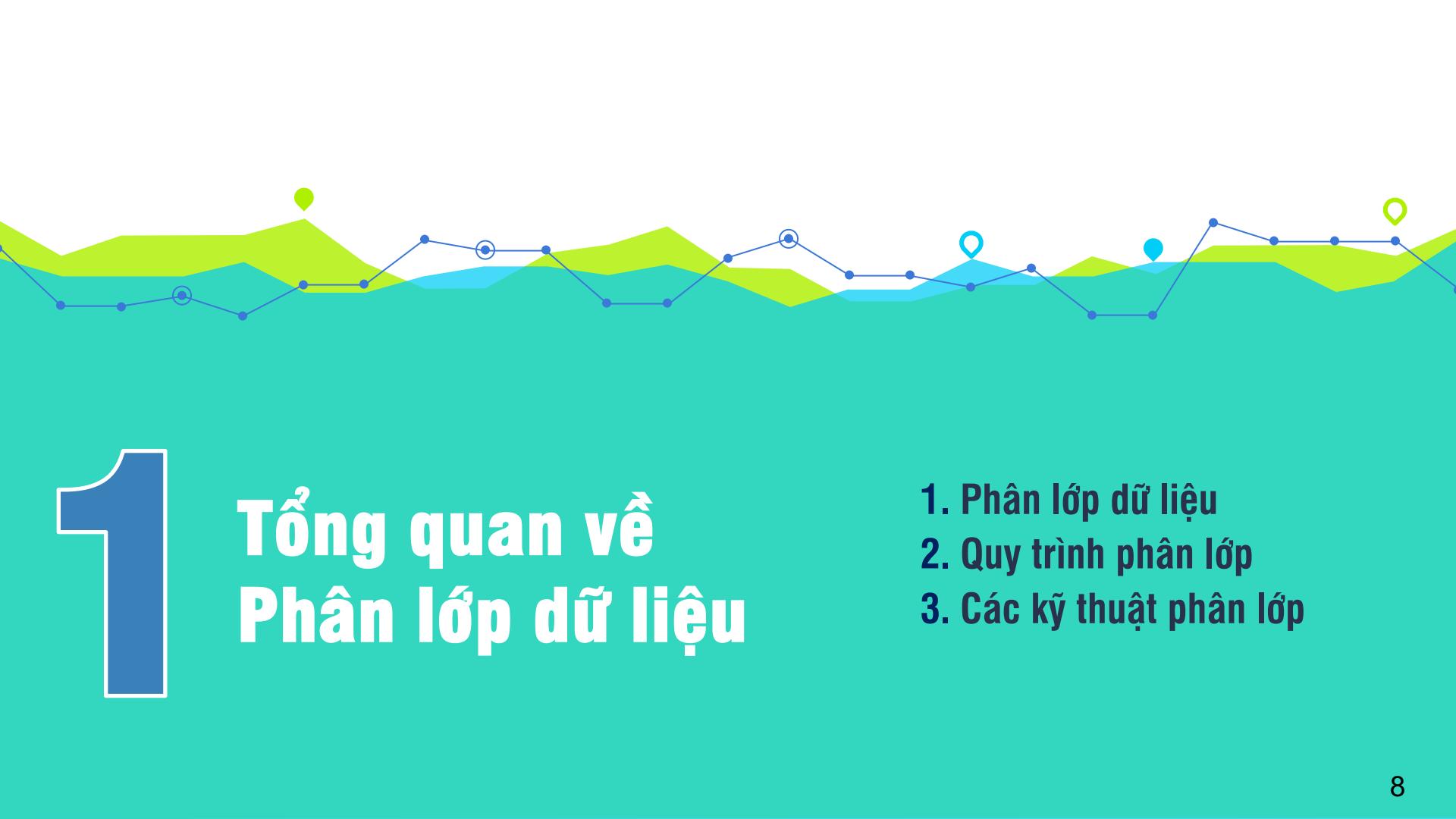
Mạng neural

Các phương pháp khác

Đánh giá mô hình

1

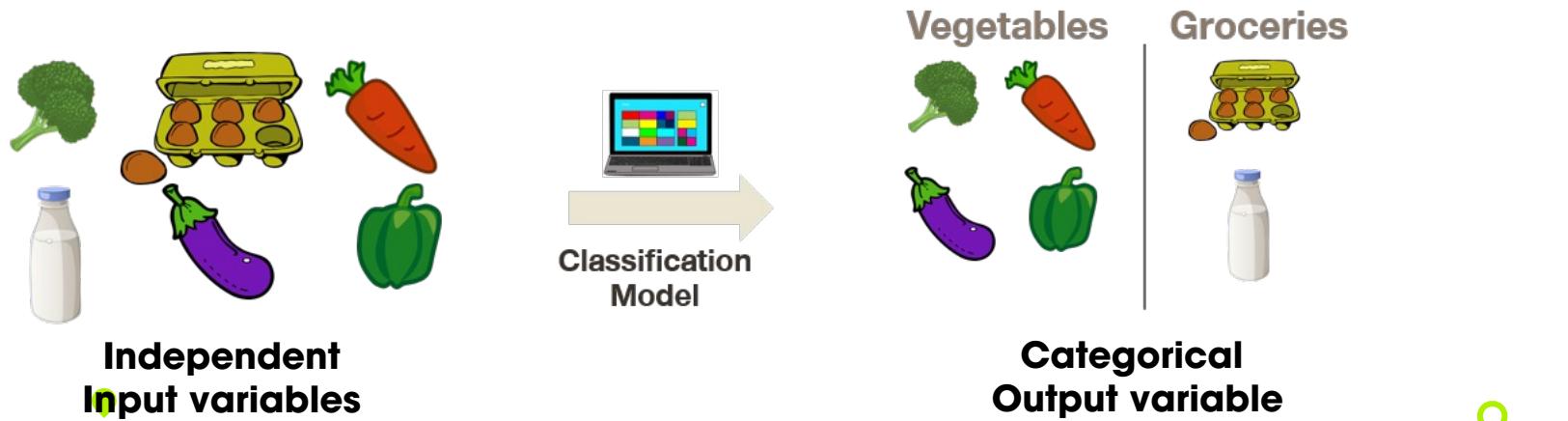
Tổng quan về Phân lớp dữ liệu

- 
1. Phân lớp dữ liệu
 2. Quy trình phân lớp
 3. Các kỹ thuật phân lớp

1. Phân lớp dữ liệu

- Phân lớp dữ liệu

- Dự đoán nhãn lớp (discrete hoặc nominal)
- Xây dựng mô hình phân lớp dựa trên tập huấn luyện và các nhãn lớp của thuộc tính phân lớp và sử dụng mô hình đó để phân lớp cho dữ liệu mới.



1. Phân lớp dữ liệu

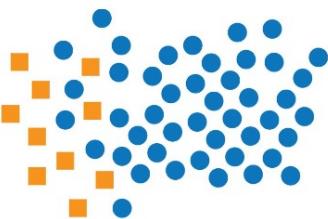
- **Some Applications of Machine Learning Classification Problems**
 - Image classification
 - Fraud detection
 - Document classification
 - Spam filtering
 - Facial recognition
 - Voice recognition
 - Medical diagnostic test
 - Customer behavior prediction
 - Product categorization
 - Malware classification



1. Phân lớp dữ liệu

- Types of Classification Tasks in Machine Learning

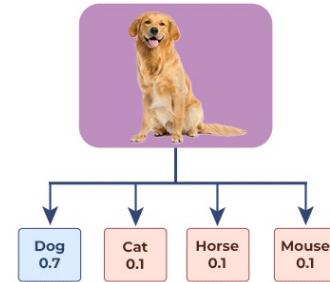
- Binary Classification
- Multi-Class Classification
- Multi-Label Classification
- Imbalanced Classification



Imbalanced data Classification



Multiclass Classification

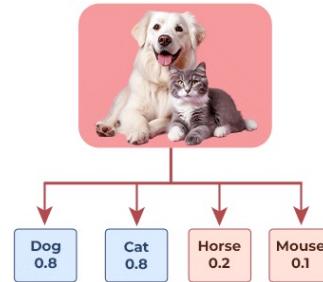


Classes

(pick one class)



Multilabel Classification

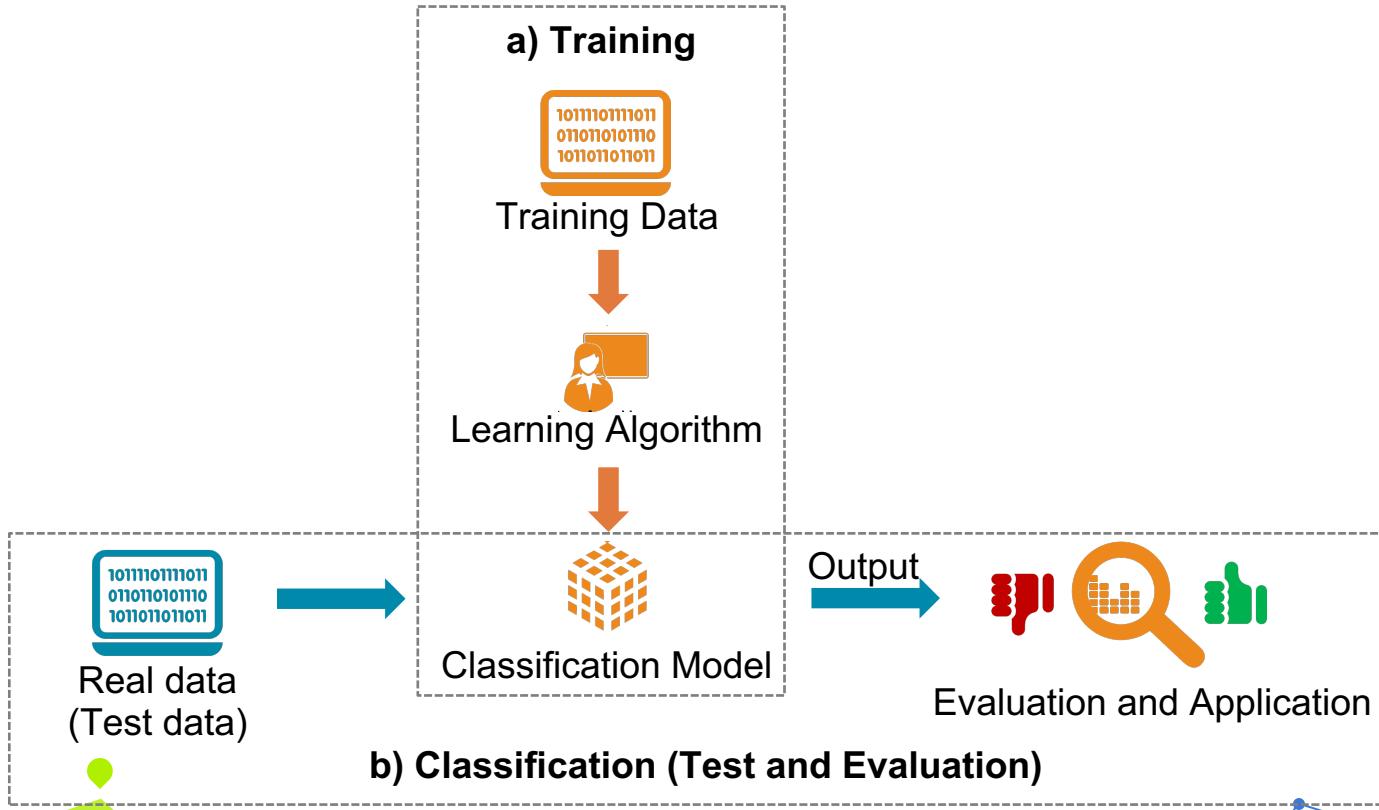


Classes

(pick all the labels present in the image)



2. Các bước trong phân lớp dữ liệu



2. Các bước trong phân lớp dữ liệu

- **Bước 1: Xây dựng mô hình – Học/ huấn luyện**

- Mỗi bộ dữ liệu được gán vào các lớp (nhãn) được xác định trước
- Tập huấn luyện (train set): Tập các bộ dữ liệu dùng để xây dựng mô hình
- Tìm ra các luật phân lớp, cây quyết định hoặc công thức toán học để mô tả mô hình.

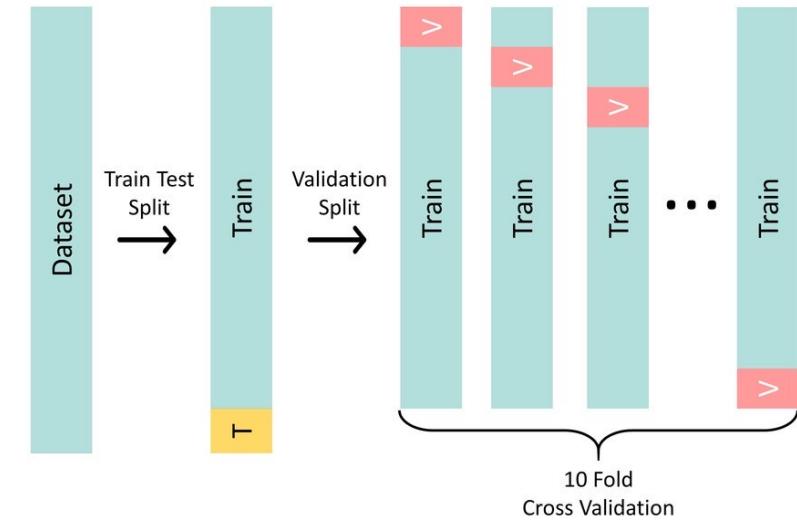
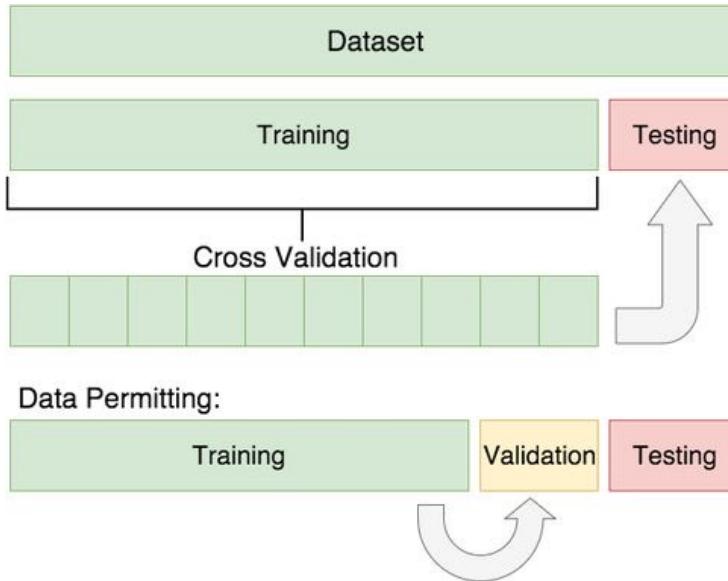


2. Các bước trong phân lớp dữ liệu

- **Bước 2: Sử dụng mô hình** – Phân lớp các đối tượng chưa biết

- Đánh giá độ chính xác của mô hình
 - So sánh nhãn mẫu test với kết quả phân lớp từ mô hình
 - Tỷ lệ chính xác: tỷ lệ mẫu thử được phân lớp chính xác
 - Tập kiểm thử (test set) độc lập với tập huấn luyện (training set)
- Sử dụng mô hình để phân lớp nếu độ chính xác chấp nhận được

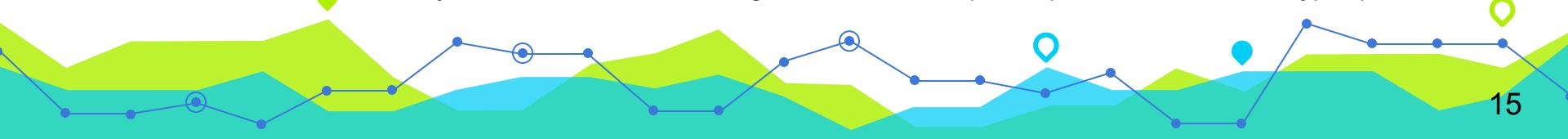
Training set – Testing set – Validation set



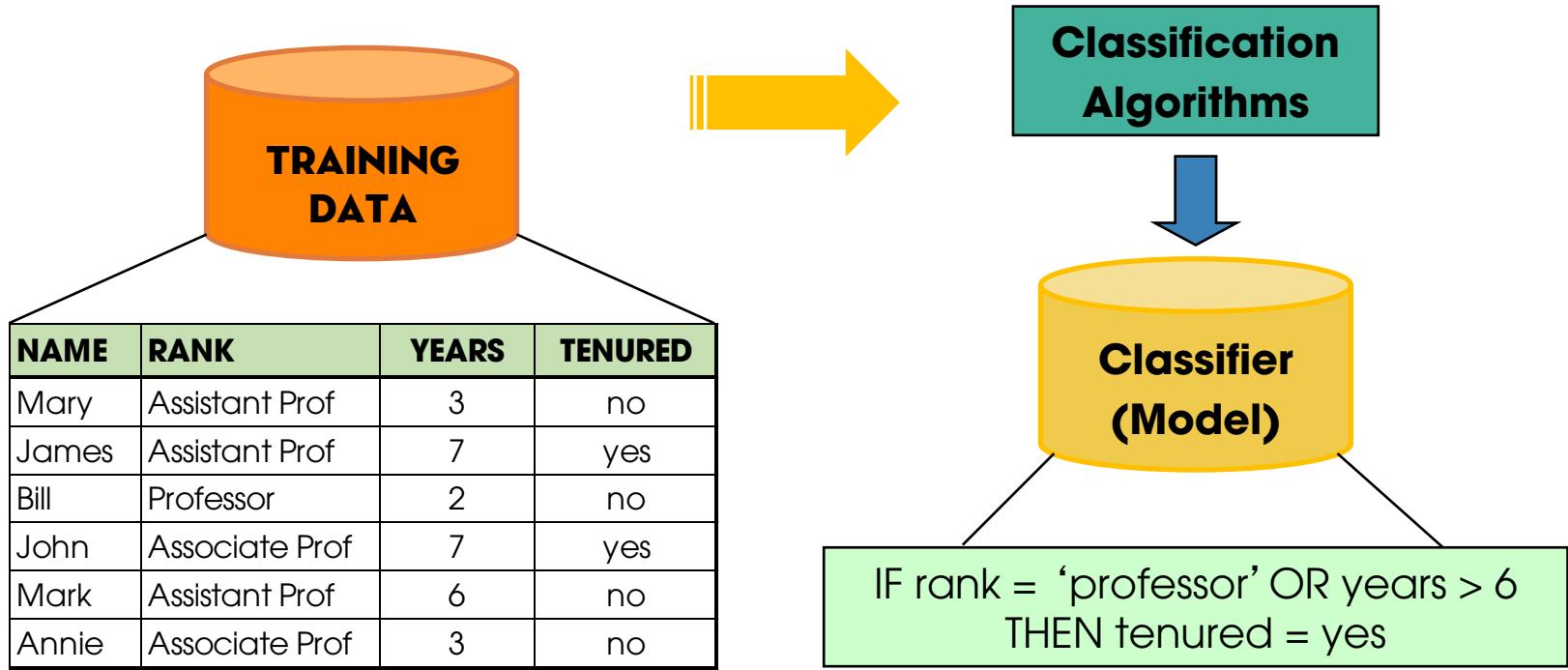
Training set: The subset of data used to train a machine learning model

Testing set: The subset of data used to evaluate the performance of a trained machine learning model on unseen examples, simulating real-world data

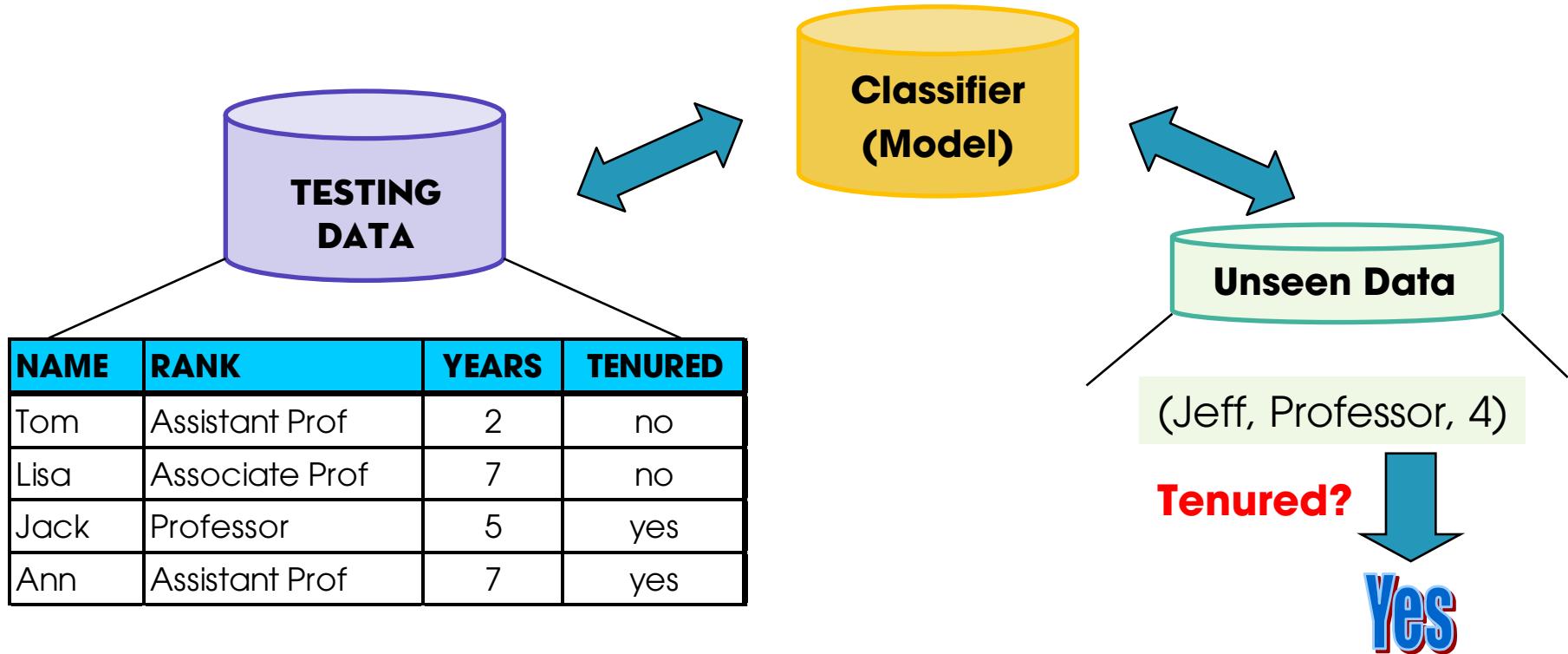
Validation set: The intermediary subset of data used during the model development process to fine-tune hyper-parameter



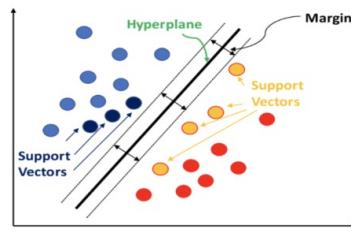
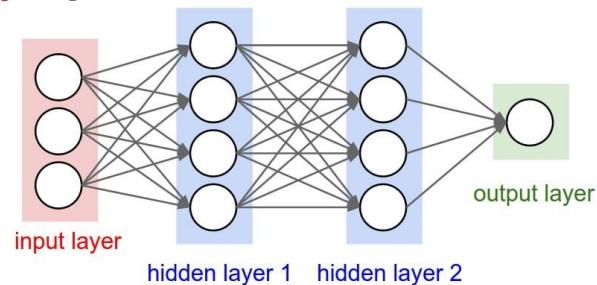
Quy trình phân lớp – B1. Xây dựng mô hình



Quy trình phân lớp – B2. Sử dụng mô hình



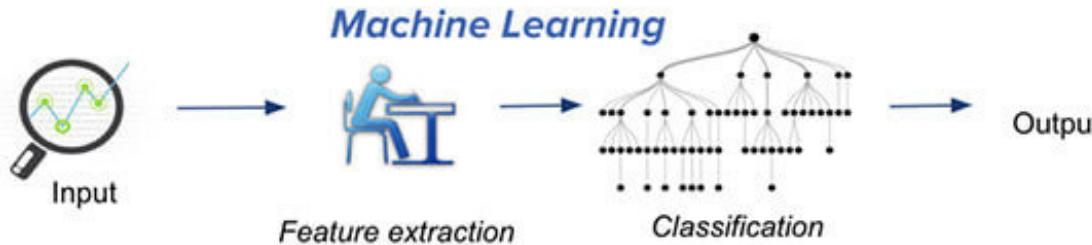
3. Các kỹ thuật phân lớp dữ liệu



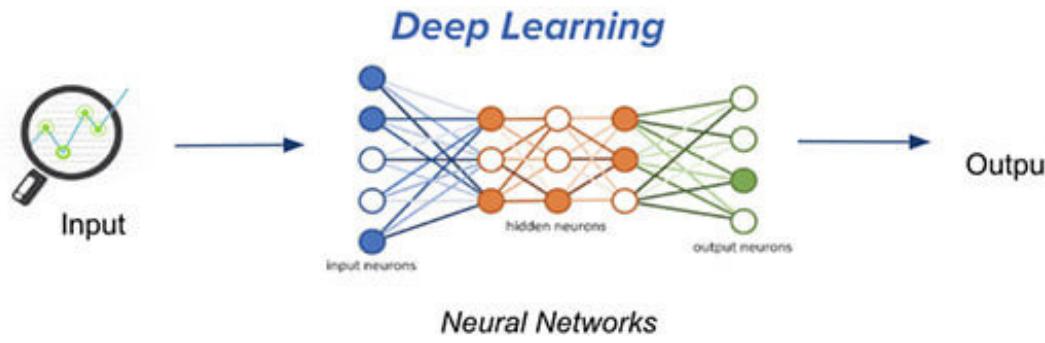
- Phương pháp dựa trên cây quyết định
- Phương pháp dựa trên luật
- Phương pháp Naïve Bayes
- Phương pháp dựa trên thể hiện
- Mạng neural / Deep Learning
- Tập thô
- SVM (Support Vector Machine) (*)
- Ensemble Methods (*)

(*): tìm hiểu, seminar

Bài toán phân lớp



Traditional machine learning uses hand-crafted features, which is tedious and costly to develop.

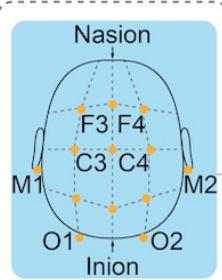


Deep learning learns hierarchical representation from the data itself, and scales with more data.

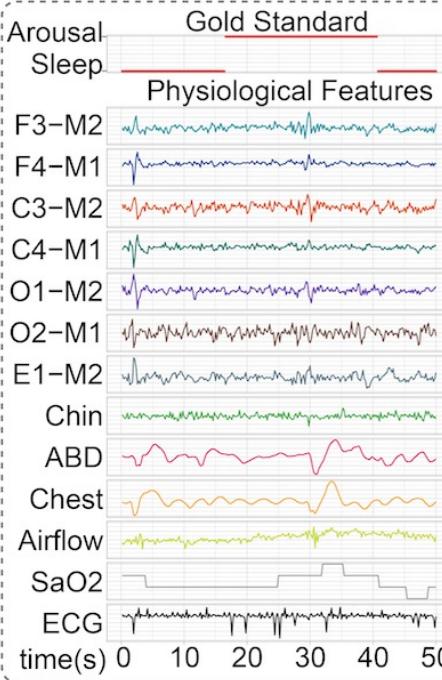


State sleep Classification

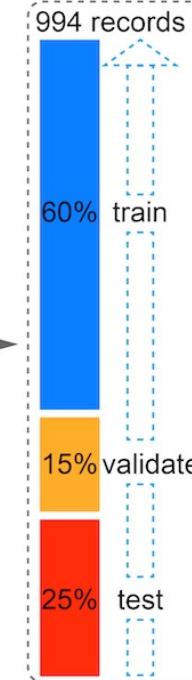
Location



Data

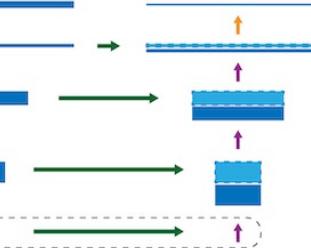


Cross-validation

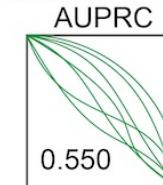
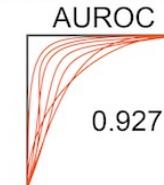


Model

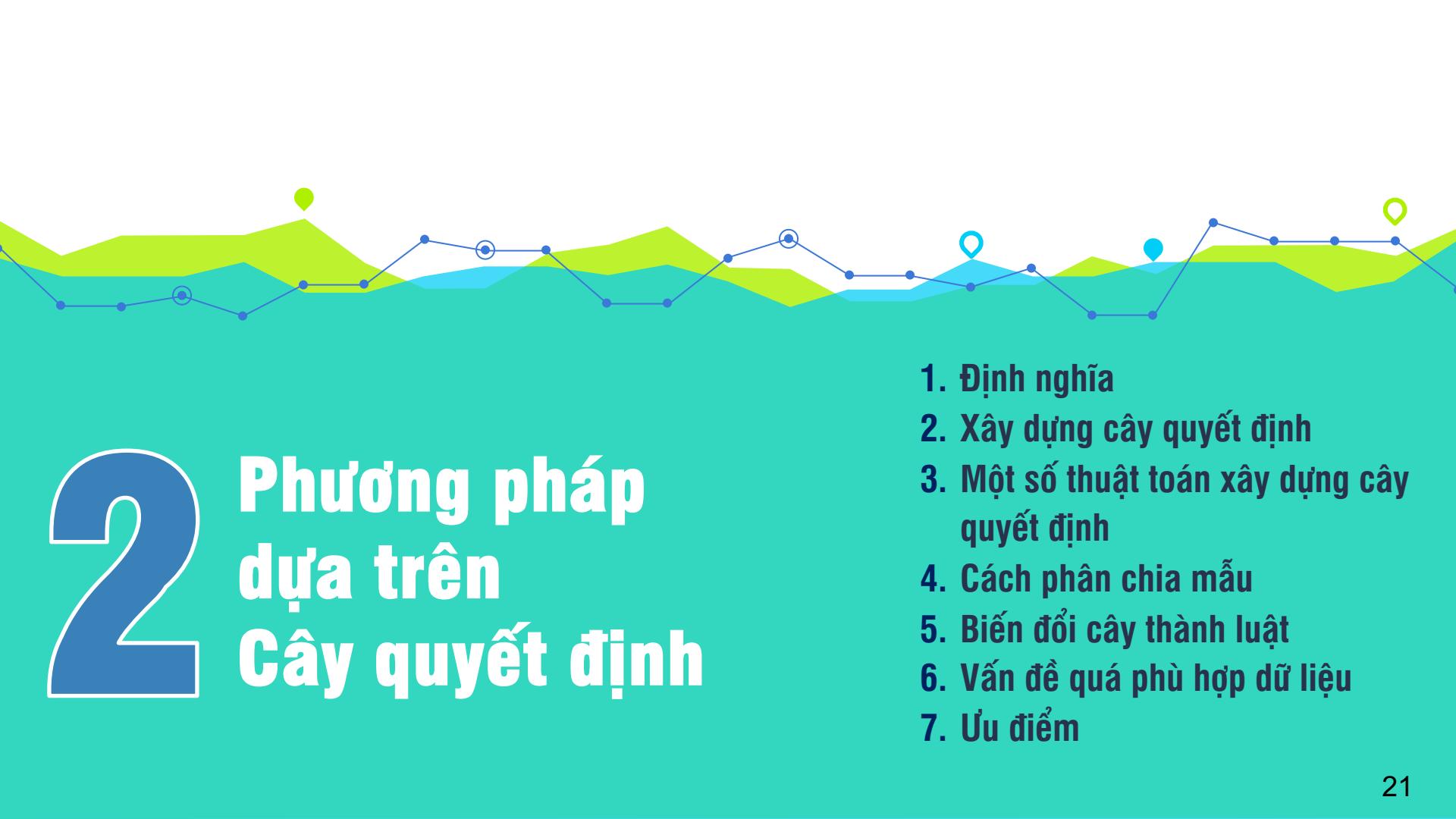
Deep U-Net



Evaluation



2 Phương pháp dựa trên Cây quyết định

- 
1. Định nghĩa
 2. Xây dựng cây quyết định
 3. Một số thuật toán xây dựng cây quyết định
 4. Cách phân chia mẫu
 5. Biến đổi cây thành luật
 6. Vấn đề quá phù hợp dữ liệu
 7. Ưu điểm

1. Định nghĩa Cây quyết định

- Là một cấu trúc phân cấp của các node và các nhánh

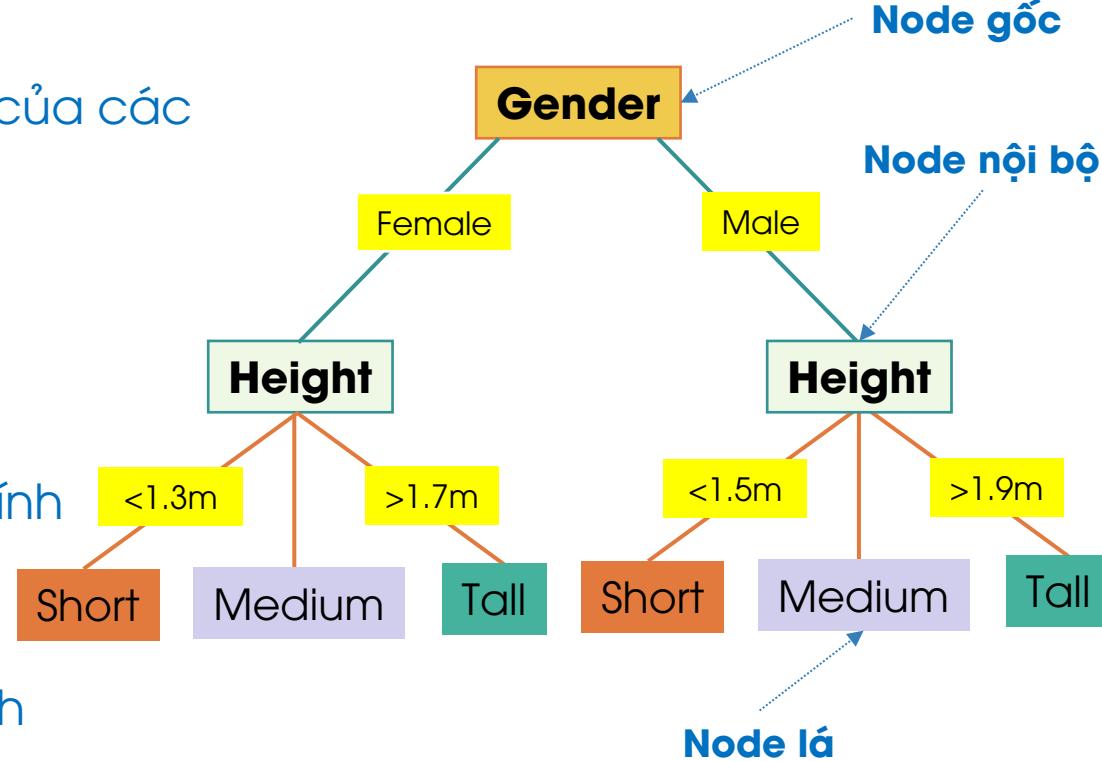
- 3 loại node

- Node gốc

- Node nội bộ: tên thuộc tính

- Node lá: tên lớp

- Nhánh: giá trị của thuộc tính



2. Xây dựng cây quyết định

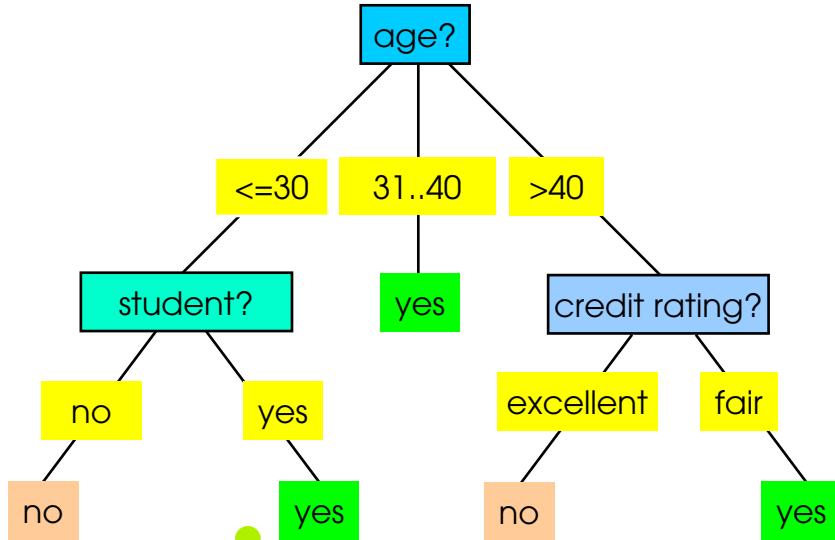
- **Bước 1: Thiết lập cây quyết định**
 - Bắt đầu từ gốc
 - Kiểm tra các giá trị của thuộc tính và phân chia các mẫu để quy
- **Bước 2: Tỉa bới cây**
 - Xác định và loại bỏ bớt các nhánh không ổn định hoặc cá biệt



2. Xây dựng cây quyết định

VD1: Minh họa xây dựng cây quyết định

- Training data set: Buys_computer
- The data set follows an example of Quinlan's ID3
- Resulting tree:



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

3. Thuật toán xây dựng cây quyết định

- **Một số thuật toán xây dựng cây quyết định:**

- Hunt's Algorithm
- CART
- ID3, C4.5
- SLIQ, SPRINT
- ...



3. Thuật toán xây dựng cây quyết định

- **Ý tưởng chính:**

- Phương pháp tham lam (greedy)
- Phân chia tập mẫu dựa trên thuộc tính cho kết quả tối ưu hóa tiêu chuẩn



3. Thuật toán xây dựng cây quyết định

- Vấn đề:

- Xác định cách phân chia mẫu: Dựa trên độ đo sự đồng nhất của dữ liệu
- Điều kiện dừng:
 - Tất cả các mẫu rơi vào một node thuộc về cùng một lớp
 - Tất cả các mẫu rơi vào một node có cùng giá trị thuộc tính



4. Cách phân chia mẫu dữ liệu

- Tiêu chuẩn phân chia: tạo ra các nhóm sao cho một lớp chiếm ưu thế trong từng nhóm
- Thuộc tính được chọn là thuộc tính cho độ đo tốt nhất, lợi nhất trong quá trình phân lớp
- Độ đo sự đồng nhất: đánh giá chất lượng phân chia:
 - Entropy (Information Gain)
 - Information Gain Ratio
 - Gini Index



4.1. Information Gain (Entropy)

- **Độ lợi thông tin.** Áp dụng trong thuật toán **ID3, C4.5**
- Chọn thuộc tính có **độ lợi thông tin cao nhất**
- **Giả sử:**
 - **D**: tập huấn luyện
 - $C_{i,D}$: tập các mẫu của D thuộc lớp C_i với $i = \{1, \dots, m\}$, **m** số lớp
 - $|C_{i,D}|, |D|$: lực lượng của tập $C_{i,D}$ và D
- p_i : xác suất để một mẫu bất kỳ của D thuộc lớp C_i
- **Info(D)**: Thông tin kỳ vọng để phân lớp một mẫu trong D:

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

$$p_i = \frac{|C_{i,D}|}{|D|} \quad (2)$$



4.1. Information Gain (Entropy)

- Độ lợi thông tin dựa trên phân chia theo 1 thuộc tính (A)

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

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} * Info(D_j) \quad (3)$$

- $Gain(A)$: Độ lợi thông tin dựa trên phân chia theo thuộc tính A

$$Gain(A) = Info(D) - Info_A(D) \quad (4)$$



4.1. Information Gain (Entropy)

VD2: Cho tập dữ liệu với thuộc tính quyết định là buy_computer.

Giả sử:

- Lớp P: buy_computer = "yes"
- Lớp N: buy_computer = "no"

Tính độ lợi thông tin dựa trên phân chia theo thuộc tính "age"

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

4.1. Information Gain (Entropy)

VD2:

- Thông tin kỳ vọng để phân lớp một mẫu trong D:

Ta có: $|D| = 14$; $|P| = 9$; $|N| = 5$

$$\Rightarrow \text{Info}(D) = I(9,5)$$

$$= -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.940$$

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

4.1. Information Gain (Entropy)

VD2:

- Thông tin cần thiết để phân chia D theo Age:

age	p _j	n _j	I(p _j , n _j)
<=30	2	3	0.971
31...40	4	0	0
>40	3	2	0.971

$$I(2,3) = -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.971$$

$$I(4,0) = -\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4} = 0$$

$$I(3,2) = -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} = 0.971$$

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



4.1. Information Gain (Entropy)

VD2:

- Thông tin cần thiết để phân chia D theo Age:

$$Info_{age}(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} * Info(D_j)$$

$$= \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2)$$

$$= \frac{5}{14} * 0.971 + \frac{4}{14} * 0 + \frac{5}{14} * 0.971 = 0.694$$

- Độ lợi thông tin khi phân chia D theo Age:

$$Gain(age) = Info(D) - Info_{age}(D)$$
$$= 0.94 - 0.694 = 0.246$$

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

4.1. Information Gain (Entropy)

BT10

Tiếp tục tính độ lợi thông tin dựa trên phân chia theo thuộc tính:

- “income”
- “student”
- “credit_rating”

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

4.1. Information Gain (Entropy)

VD2: Cho tập dữ liệu với thuộc tính quyết định là buy_computer.

Xây dựng cây quyết định (sử dụng độ lợi thông tin để chọn thuộc tính)

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

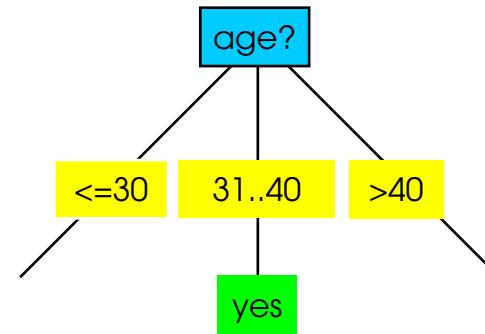
4.1. Information Gain (Entropy)

VD2: Xây dựng cây quyết định

Từ VD1 và BT10:

- $\text{Gain}(\text{age}) = 0.246$
- $\text{Gain}(\text{income}) = 0.029$
- $\text{Gain}(\text{student}) = 0.151$
- $\text{Gain}(\text{credit_rating}) = 0.048$

=> Chọn thuộc tính **age**

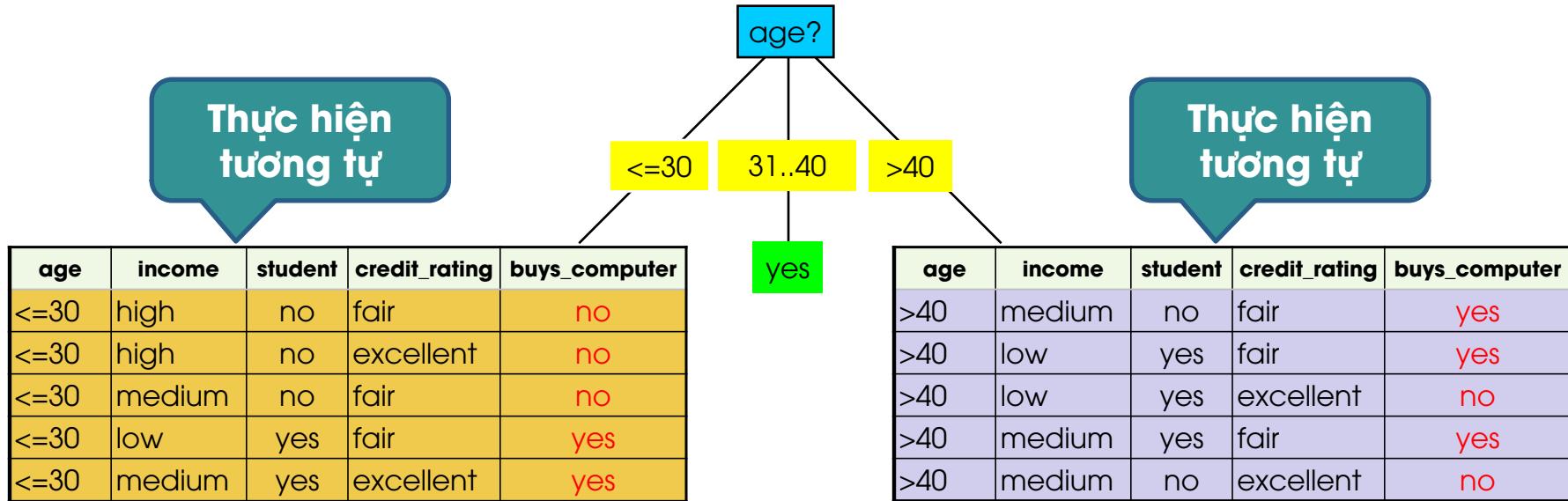


Với $\text{age} = \text{"31..40"}$, $I(p_j, n_j) = 0$
=> $\text{buy_computer} = \text{"yes"}$



4.1. Information Gain (Entropy)

VD2: Xây dựng cây quyết định



4.1. Information Gain (Entropy)

VD2: Xây dựng cây quyết định

- **Xét nhánh Age <=30:** thực hiện tương tự, tính:
 - Gain(income)
 - Gain(student)
 - Gain(credit_rating)

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

4.1. Information Gain (Entropy)

VD2: Xây dựng cây quyết định

- **Xét nhánh Age <=30:** thực hiện tương tự, tính:

- Gain(income): 0.571
- Gain(student): 0.971
- Gain(credit_rating): 0.952

=> Chọn thuộc tính **student**

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
<=30	medium	yes	excellent	yes



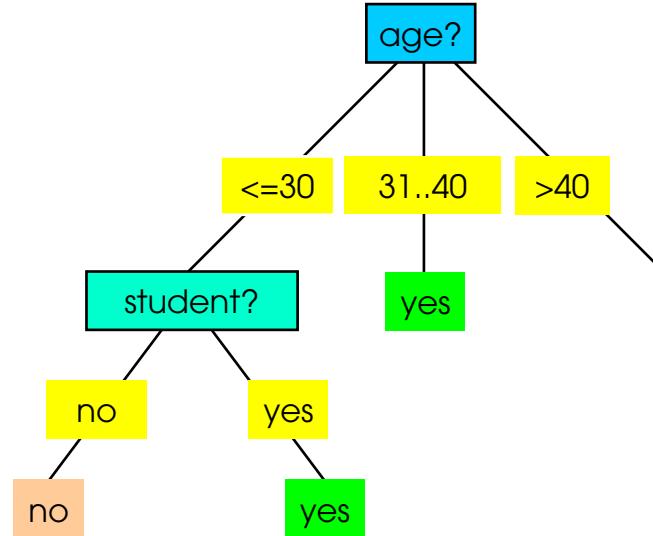
4.1. Information Gain (Entropy)

VD2: Xây dựng cây quyết định

- **Xét nhánh Age ≤ 30 :** thực hiện tương tự, tính:

- Gain(income): 0.571
- Gain(student): 0.971
- Gain(credit_rating): 0.952

=> Chọn thuộc tính **student**



4.1. Information Gain (Entropy)

VD2: Xây dựng cây quyết định

- **Xét nhánh Age >40:** thực hiện tương tự, tính:
 - Gain(income)
 - Gain(student)
 - Gain(credit_rating)

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

4.1. Information Gain (Entropy)

VD2: Xây dựng cây quyết định

- **Xét nhánh Age >40:** thực hiện tương tự, tính:

- Gain(income): 0.019
- Gain(student): 0.003
- Gain(credit_rating): 0.971

=> Chọn thuộc tính **credit_rating**

age	income	student	credit_rating	buys_computer
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
>40	medium	yes	fair	yes
>40	medium	no	excellent	no



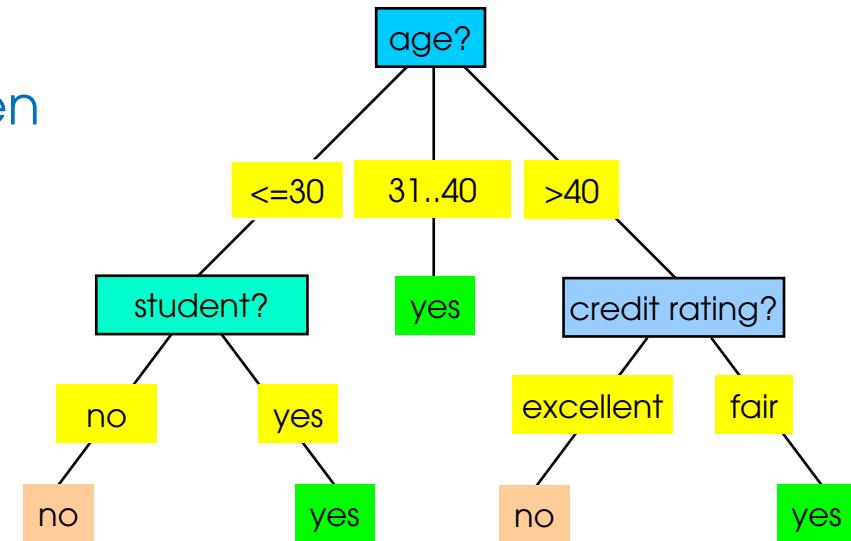
4.1. Information Gain (Entropy)

VD2: Xây dựng cây quyết định

- **Xét nhánh Age >40:** thực hiện tương tự, tính:

- Gain(income): 0.019
- Gain(student): 0.003
- Gain(credit_rating): 0.971

=> Chọn thuộc tính **credit_rating**



4.2. Gain Ratio

- Áp dụng trong thuật toán **C4.5** (cải tiến từ ID3)
- Độ lợi thông tin (Gain) có xu hướng thiên vị chọn các thuộc tính có nhiều giá trị → cần chuẩn hóa độ đo Gain
- Chọn thuộc tính có độ đo **Gain Ratio lớn nhất**

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)} \quad (5)$$

$$SplitInfo_A(D) = - \sum_{j=1}^v \frac{|D_j|}{|D|} * \log_2\left(\frac{|D_j|}{|D|}\right) \quad (6)$$



4.2. Gain Ratio

VD3: Tính Gain Ratio dựa trên phân chia theo thuộc tính “income”

Từ BT10: **Gain(income) = 0.029**

$$\begin{aligned} \text{SplitInfo}_A(D) &= -\sum_{j=1}^v \frac{|D_j|}{|D|} * \log_2 \left(\frac{|D_j|}{|D|} \right) \\ &= -\frac{4}{14} \log_2 \frac{4}{14} - \frac{6}{14} \log_2 \frac{6}{14} - \frac{4}{14} \log_2 \frac{4}{14} \\ &= 1.557 \end{aligned}$$

$$\begin{aligned} \text{GainRatio}(A) &= \frac{\text{Gain}(A)}{\text{SplitInfo}_A(D)} \\ &= \frac{0.029}{1.557} = 0.019 \end{aligned}$$

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

4.3. Gini index

- Áp dụng trong thuật toán **CART, SLIQ, SPRINT**
- Chọn thuộc tính có **Gini index nhỏ nhất**
- **Giả sử:**
 - D : tập huấn luyện
 - $C_{i,D}$: tập các mẫu của D thuộc lớp C_i với $i = \{1, \dots, m\}$, m số lớp
 - $|C_{i,D}|, |D|$: lực lượng của tập $C_{i,D}$ và D
- p_i : xác suất để một mẫu bất kỳ của D thuộc lớp C_i
- **Gini(D)**: Gini index của tập D :

$$gini(D) = 1 - \sum_{i=1}^m (p_i)^2 \quad (7)$$

$$p_i = \frac{|C_{i,D}|}{|D|} \quad (2)$$



4.3. Gini index

- Gini index dựa trên phân chia theo 1 thuộc tính (A)

- Thuộc tính A có v các giá trị: $\{a_1, a_2, \dots, a_v\}$
- Dùng A để phân chia tập train D thành v tập con $\{D_1, D_2, \dots, D_v\}$
- Gini index của phân chia D theo thuộc tính A:

$$gini_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} * gini(D_j) \quad (8)$$



4.3. Gini index

VD4: Tính Gini index của tập D

Ta có: $|D| = 14$; $|P| = 9$; $|N| = 5$

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

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

4.3. Gini index

VD4: Tính Gini index cho thuộc tính "age":

age	p _j	n _j	Gini (p _j , n _j)
<=30	2	3	0.48
31...40	4	0	0
>40	3	2	0.48

$$Gini(2,3) = 1 - \left[\left(\frac{2}{5}\right)^2 + \left(\frac{3}{5}\right)^2 \right] = 0.48$$

$$Gini(4,0) = 1 - \left[\left(\frac{4}{4}\right)^2 + \left(\frac{0}{4}\right)^2 \right] = 0$$

$$Gini(3,2) = 1 - \left[\left(\frac{3}{5}\right)^2 + \left(\frac{2}{5}\right)^2 \right] = 0.48$$

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



4.3. Gini index

VD4: Tính Gini index cho thuộc tính "age":

$$\begin{aligned}gini_{age}(D) &= \sum_{j=1}^v \frac{|D_j|}{|D|} * gini(D_j) \\&= \frac{5}{14} gini(2,3) + \frac{4}{14} gini(4,0) + \frac{5}{14} gini(3,2) \\&= \frac{5}{14} * 0.48 + \frac{4}{14} * 0 + \frac{5}{14} * 0.48 = \mathbf{0.343}\end{aligned}$$

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

4.3. Gini index

VD5: Cho tập dữ liệu với thuộc tính quyết định là buy_computer.

Xây dựng cây quyết định (sử dụng Gini index để chọn thuộc tính)

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

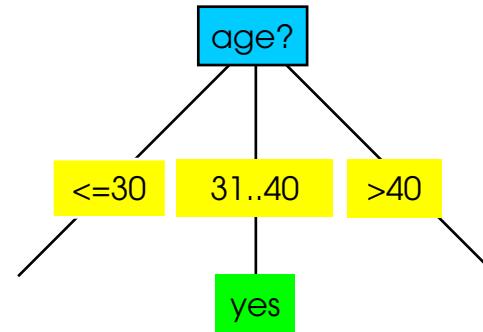
4.3. Gini index

VD5: Xây dựng cây quyết định

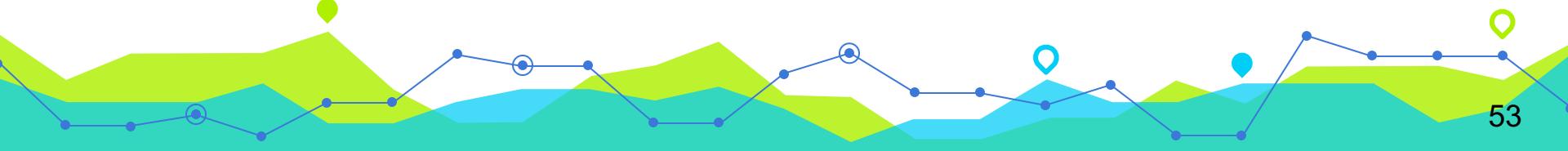
Thực hiện tương tự VD4, ta có:

- $\text{Gini}(\text{age}) = 0.343$
- $\text{Gini}(\text{income}) = 0.44$
- $\text{Gini}(\text{student}) = 0.367$
- $\text{Gini}(\text{credit_rating}) = 0.429$

=> Chọn thuộc tính **age**

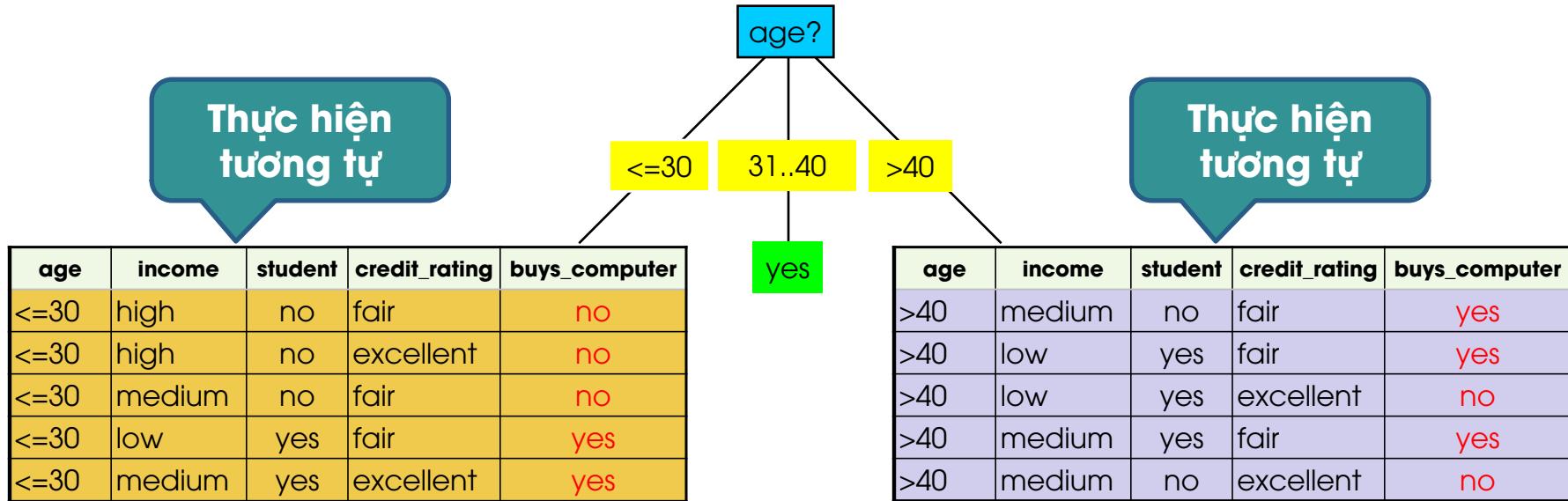


Với $\text{age} = \text{"31..40"}$, $I(p_j, n_j) = 0$
=> $\text{buy_computer} = \text{"yes"}$



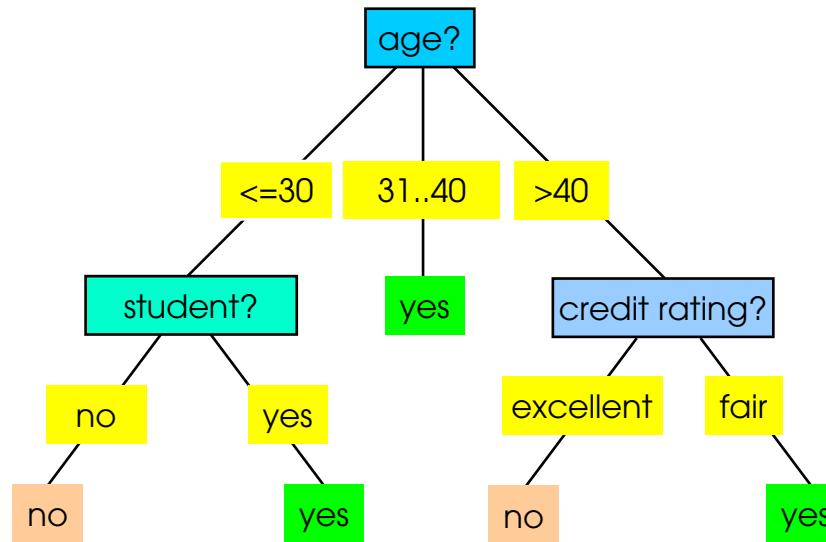
4.3. Gini index

VD5: Xây dựng cây quyết định



4.3. Gini index

VD5: Xây dựng cây quyết định



So sánh các độ đo chọn thuộc tính

Information Gain

- Thiên vị thuộc tính có nhiều giá trị

Gain Ratio

- Thiên về việc phân chia không cân bằng giữa các vùng

Gini index

- Thiên vị thuộc tính có nhiều giá trị
- Gặp vấn đề khi số lớp lớn
- Có xu hướng ưu tiên cho các thử nghiệm tạo ra kết quả phân chia có cùng kích thước và cùng sự đồng nhất



Những độ đo khác để chọn thuộc tính

- CHAID: a popular decision tree algorithm, measure based on χ^2 test for independence
- C-SEP: performs better than info. gain and gini index in certain cases
- G-statistic: has a close approximation to χ^2 distribution
- MDL (Minimal Description Length) principle (i.e., the simplest solution is preferred):
 - The best tree as the one that requires the fewest # of bits to both (1) encode the tree, and (2) encode the exceptions to the tree
- Multivariate splits (partition based on multiple variable combinations)
 - CART: finds multivariate splits based on a linear comb. of attrs.
- Which attribute selection measure is the best?
 - Most give good results, none is significantly superior than others

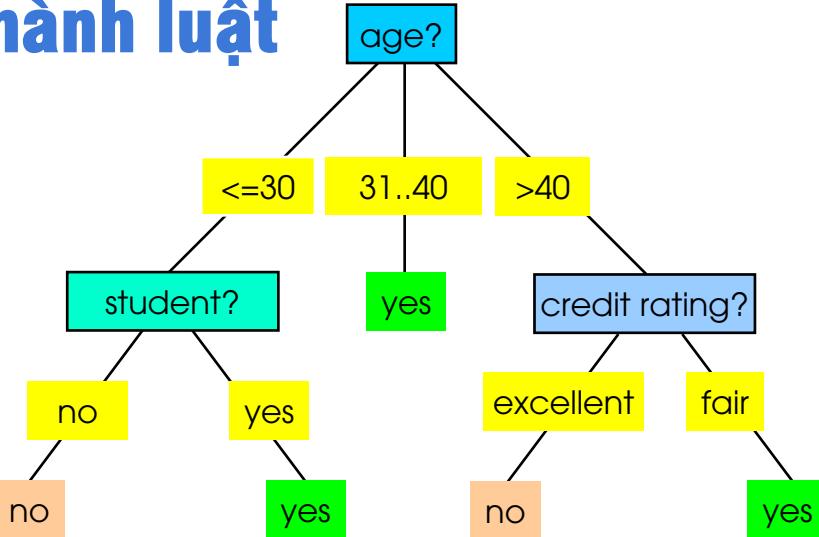
5. Biến đổi cây thành luật

- Biểu diễn tri thức dưới dạng luật IF-THEN
- Mỗi luật tạo ra từ mỗi đường dẫn từ gốc đến lá
- Mỗi cặp giá trị thuộc tính dọc theo đường dẫn tạo nên phép kết (phép AND)
- Các nút lá mang tên của lớp



5. Biến đổi cây thành luật

VD6: Xác định các luật



R₁: If (age <= 30) \wedge (student = No)

R₂: If (age <= 30) \wedge (student = Yes)

R₃: If (age = 31..40)

R₄: If (age > 40) \wedge (credit_rating = Excellent)

R₅: If (age > 40) \wedge (credit_rating = Fair)

Then buy_computer = No

Then buy_computer = Yes

Then buy_computer = Yes

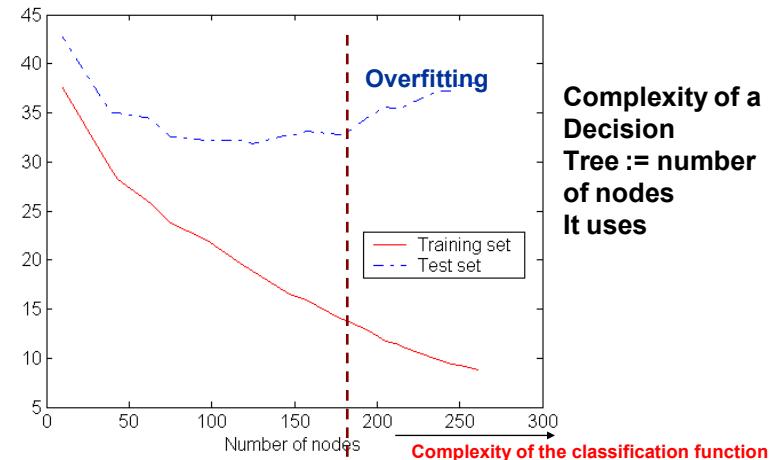
Then buy_computer = No

Then buy_computer = Yes



6. Vấn đề quá phù hợp với dữ liệu (Overfitting)

- Overfitting: Mô hình tạo ra có thể quá phù hợp với dữ liệu train nhưng độ chính xác kém trên dữ liệu test, unseen data.
- Cây quá phù hợp với dữ liệu train:
 - Quá nhiều nhánh do dữ liệu nhiều hoặc cá biệt
 - Do thiếu mẫu. Thiếu dữ liệu ở 1 vùng nào đó gây khó khăn cho việc dự đoán lớp chính xác của vùng này



6. Vấn đề quá phù hợp với dữ liệu (Overfitting)

- Để tránh quá phù hợp dữ liệu: Tỉa cây
 - Tỉa trước: Dừng thêm nhánh cây sớm, ngay khi nó có thể tạo ra độ d dưới ngưỡng nào đó. Tuy nhiên, rất khó chọn ngưỡng thích hợp
 - Tỉa sau: Loại bỏ nhánh từ cây hoàn chỉnh. Sử dụng tập dữ liệu độc lập để kiểm thử (validation) và loại bỏ



7. Ưu điểm

- Dễ dàng xây dựng cây
- Phân lớp mẫu mới nhanh
- Dễ dàng diễn giải cho các cây có kích thước nhỏ
- Độ chính xác chấp nhận được so với các kỹ thuật phân lớp khác trên nhiều tập dữ liệu đơn

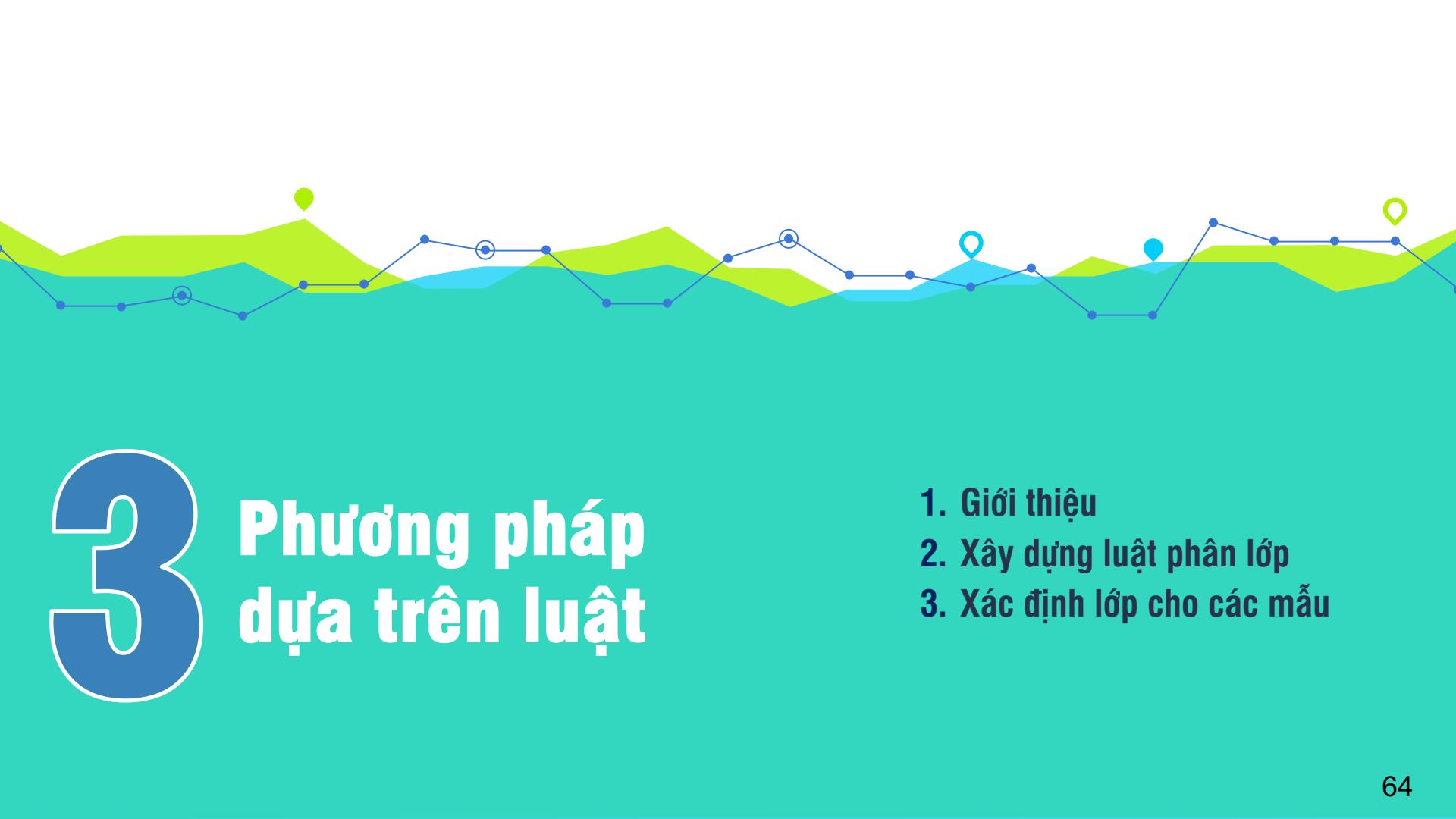


Phân lớp trong CSDL lớn

- **Khả năng mở rộng:** Phân lớp tập dữ liệu với hàng triệu mẫu và hàng trăm thuộc tính với tốc độ hợp lý
- Tại sao cây quyết định lại phổ biến?
 - Tốc độ học tương đối nhanh (so với các phương pháp phân lớp khác)
 - Chuyển đổi thành các luật phân lớp đơn giản và dễ hiểu
 - Có thể sử dụng truy vấn SQL để truy cập CDSL
 - Độ chính xác phân lớp có thể so sánh với các phương pháp khác
- **RainForest** (VLDB'98 — Gehrke, Ramakrishnan & Ganti) Xây dựng danh sách AVC (thuộc tính, giá trị, nhãn lớp)
- **BOAT** (SIGMOD'99) Sử dụng một kỹ thuật thống kê được gọi là bootstrapping để tạo một số mẫu nhỏ hơn (tập hợp con)

3

Phương pháp dựa trên luật

- 
1. Giới thiệu
 2. Xây dựng luật phân lớp
 3. Xác định lớp cho các mẫu

1. Giới thiệu

- Sử dụng các luật IF <điều kiện> THEN <kết luận> để phân lớp
- VD: IF (age \leq 30) \wedge (student = Yes) THEN buy_computer = Yes
- Luật R phủ một mẫu x nếu các thuộc tính của mẫu thỏa mãn điều kiện của luật



1. Giới thiệu

- **Coverage (R) – Độ phủ của luật:**
 - Tỷ lệ các mẫu thỏa điều kiện (về trái) của luật
- **Accuracy (R) – Độ chính xác của luật:**
 - Tỷ lệ các mẫu thỏa mãn cả điều kiện (về trái) và kết luận (về phải) của luật



1. Giới thiệu

VD7:

R: IF (Marital status = Single) → No

Coverage (R) = 4/10 = 40%

Accuracy (R) = 2/4 = 50%

Tid	Refund	Marital status	Taxable income	Cheat
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



2. Xây dựng luật phân lớp

- Phương pháp trực tiếp

- Rút các luật trực tiếp từ dữ liệu
- VD: RIPPER, CN2, ILA, FOIL, AQ, ... (*)

- Phương pháp gián tiếp

- Rút luật từ các mô hình phân lớp khác: cây quyết định, mạng neural
- VD: luật C4.5, ID3, ...

(*): tìm hiểu và seminar



3. Xác định lớp cho mẫu mới

VD7: Cho tập dữ liệu train và tập luật sau:

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	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
salar	cold	No	No	Sometimes	amphibians
gila	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

3. Xác định lớp cho mẫu mới

VD7: Cho tập dữ liệu train và tập luật sau:

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

Sử dụng tập luật để xác định 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	?
shark	cold	Yes	No	Yes	?



3. Xác định lớp cho mẫu mới

VD7:

Name	Blood type	Give birth	Can fly	Live in water	Class
lemur	warm	Yes	No	No	?
turtle	cold	No	No	Sometimes	?
shark	cold	Yes	No	Yes	?

- Mẫu “lemur” phủ bởi R3: phân lớp “Mammals”

R₃: (Give birth = yes) \wedge (Blood type = warm) \rightarrow Mammals

- Mẫu “turtle” phủ bởi R4 và R5: phân lớp ??

R₄: (Give birth = no) \wedge (Can fly = no) \rightarrow Reptiles

R₅: (Live in water = sometimes) \rightarrow Amphibians

- Mẫu “shark” không được phủ bởi bất kỳ luật nào: phân lớp ???

3. Xác định lớp cho mẫu mới

- **Xếp hạng các luật theo độ ưu tiên**
 - Theo kích thước của luật: các luật có tập điều kiện lớn hơn sẽ có độ ưu tiên cao hơn
 - Theo luật: các luật được xếp hạng theo độ đo chất lượng luật hoặc theo ý kiến chuyên gia
- Nếu một mẫu được phủ bởi nhiều luật: chọn luật thứ hạng cao nhất
- Nếu không phủ bởi bất kỳ luật nào thì gán vào lớp mặc định



Cây quyết định: Bài tập

BT11

Cho tập dữ liệu huấn luyện

Day	Outlook	Temp	Humidity	Windy	Play
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Cây quyết định: Bài tập

BT11

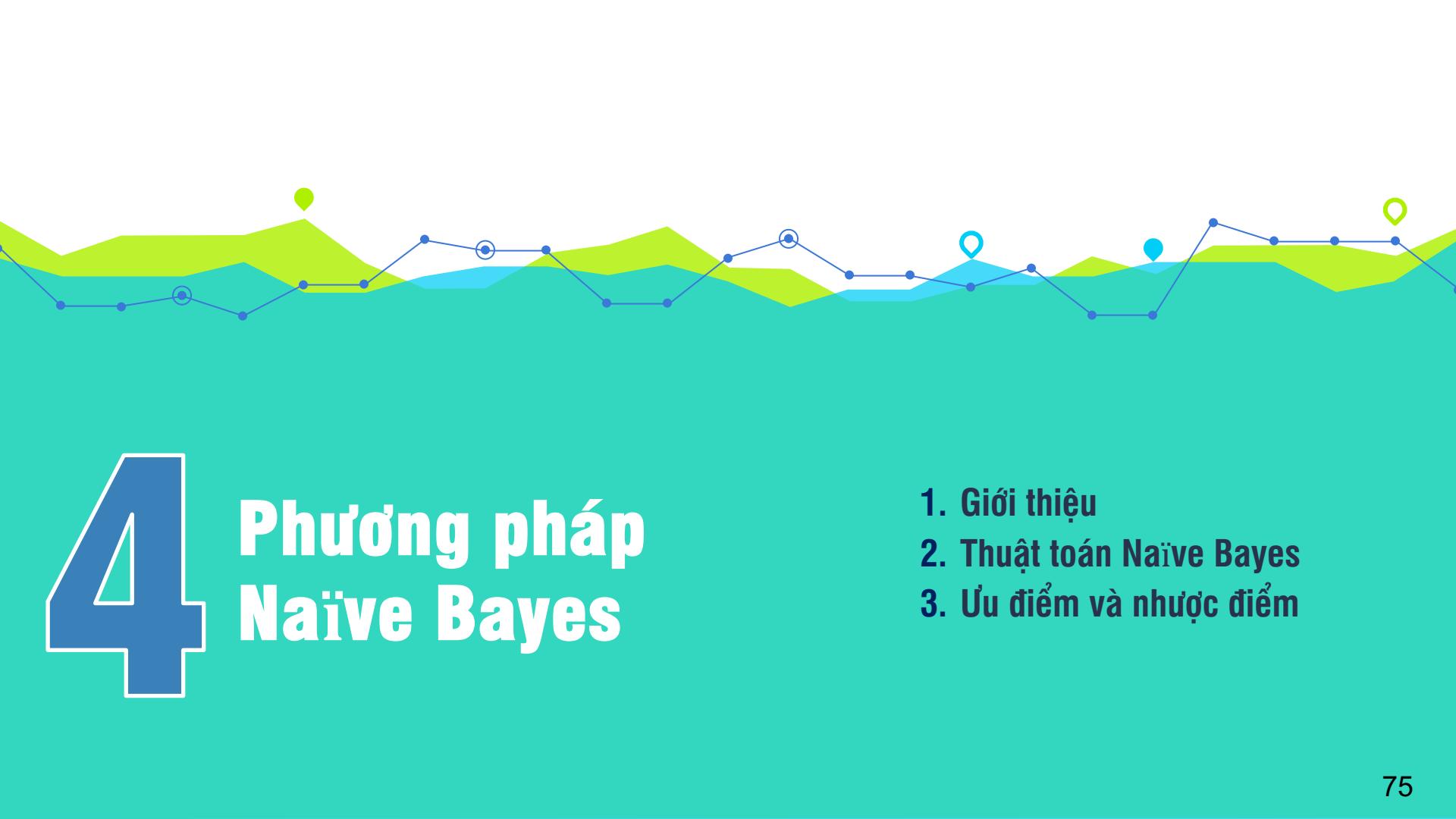
Các yêu cầu:

1. Tính Gain, Gain Ratio, Gini index cho các thuộc tính
2. Xây dựng cây quyết định (sử dụng Gini index)
3. Sử dụng luật ở câu 2. để xác định lớp:

Outlook	Temp	Humidity	Wind	Play
Rain	Mild	Normal	Strong	?
Sunny	Mild	High	Strong	?

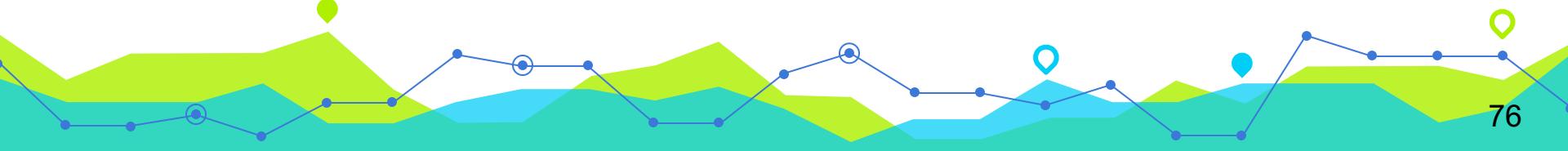


4 Phương pháp Naïve Bayes

- 
1. Giới thiệu
 2. Thuật toán Naïve Bayes
 3. Ưu điểm và nhược điểm

1. Giới thiệu

- **Phân lớp theo mô hình xác suất**
- Dự đoán xác suất mẫu là thành viên của lớp
- Dựa trên định lý Bayes (1673)
 - Cho X, Y: các biến bất kỳ (rời rạc, số, cấu trúc, ...)
 - Dự đoán Y từ X
- Lượng giá các tham số của $P(X | Y)$, $P(Y)$ từ tập train



1.1. Định lý Bayes

$$P(Y|X) = \frac{P(X|Y) P(Y)}{P(X)} \quad (9)$$

Cụ thể:

$$P(Y = y_i | X = x_j) = \frac{P(X = x_j | Y = y_i) P(Y = y_i)}{P(X = x_j)}$$

Biến bất kỳ

Giá trị thứ i



1.1. Định lý Bayes

$$P(Y = y_i | X = x_j) = \frac{P(X = x_j | Y = y_i) P(Y = y_i)}{P(X = x_j)}$$

Tương đương:

$$(\forall i, j) P(Y = y_i | X = x_j) = \frac{P(X = x_j | Y = y_i) P(Y = y_i)}{\sum_k P(X = x_j | Y = y_k) P(Y = y_k)}$$



1.2. Phân lớp Bayes

Xây dựng mô hình: **Lượng giá $P(X | Y)$, $P(Y)$**

Phân lớp: Dùng định lý Bayes để tính $P(Y | X^{new})$

Tập huấn luyện

	X					Y
	Day	Outlook	Temp	Humidity	Wind	Play
1	Sunny	Hot	High	Weak	No	
2	Sunny	Hot	High	Strong	No	
3	Overcast	Hot	High	Weak	Yes	
4	Rain	Mild	High	Weak	Yes	

1.3. Độc lập điều kiện

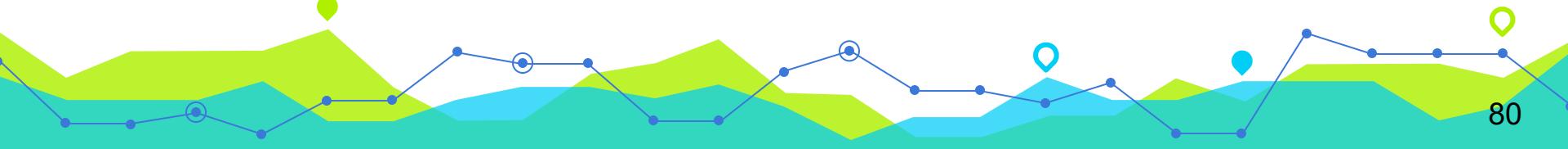
- X độc lập điều kiện với Y khi cho Z nếu phân bố xác suất trên X độc lập với các giá trị của Y khi cho các giá trị của Z

$$(\forall i, j, k) P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

Có thể viết:

$$P(X|Y, Z) = P(X|Z)$$

- VD: $P(\text{Sấm sét} | \text{Mưa, Chớp}) = P(\text{Sấm sét} | \text{Chớp})$



2. Thuật toán Naïve Bayes

- Giả sử:
 - **D:** tập train gồm các mẫu biến diễn dưới dạng $X = \langle x_1, \dots, x_n \rangle$
 - **C_i, D:** tập các mẫu của D thuộc lớp C_i với i = {1, 2, ..., m}
 - Các thuộc tính x₁, ..., x_n độc lập điều kiện đôi một với nhau khi cho Lớp C
- Cần xác định xác suất **P(C_i|X) lớn nhất**

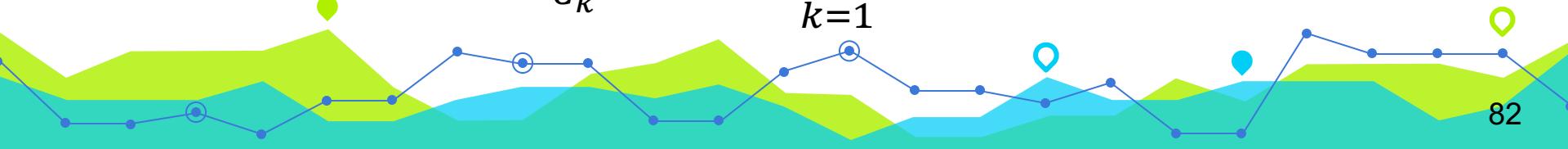
2. Thuật toán Naïve Bayes

- **Theo định lý Bayes:** $P(C_i|X) = \frac{P(X|C_i) P(C_i)}{P(X)}$ (10)
- **Theo tính chất độc lập điều kiện:**

$$P(X|C_i) = \prod_{k=1}^n P(x_k|C_i) = P(x_1|C_i) * P(x_2|C_i) * \dots * P(x_n|C_i) \quad (11)$$

- **Luật phân lớp cho $X^{\text{new}} = \{x_1, \dots, x_n\}$ là:**

$$\operatorname{argmax}_{C_k} P(C_i) \prod_{k=1}^n P(x_k|C_i)$$



2. Thuật toán Naïve Bayes

- **B1. Huấn luyện Naïve Bayes trên tập train**
 - Tính $P(C_i)$
 - Tính $P(X|C_i)$
- **B2: X^{new} được gán vào lớp cho giá trị theo công thức lớn nhất:**

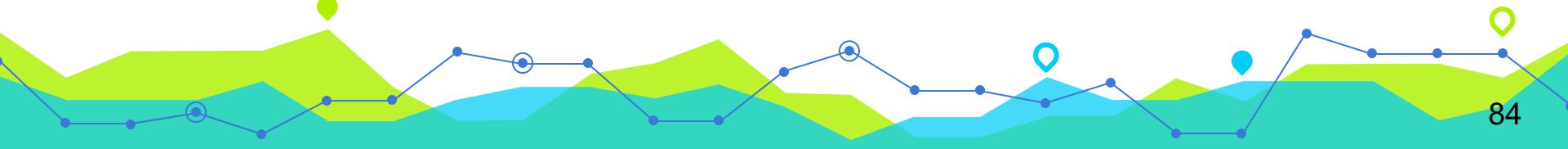
$$\operatorname{argmax}_{C_k} P(C_i) \prod_{k=1}^n P(x_k | C_i) \quad (11)$$



2. Thuật toán Naïve Bayes

- **Chia 2 trường hợp:**

- Tập X - tập các giá trị rời rạc
- Tập X – tập các giá trị liên tục



2.1. Tập X – Tập giá trị rời rạc

- Giả sử:
 - Tập $X = \langle x_1, \dots, x_n \rangle$
 - x_i nhận các giá trị rời rạc

- Khi đó:
$$P(C_i) \approx \frac{|C_{i,D}|}{|D|} = \frac{\text{Số mẫu của } C_i \text{ trong } D}{\text{Số mẫu của } D} \quad (12)$$

$$P(x_k | C_i) \approx \frac{\#C_{i,D}(x_k)}{|C_{i,D}|} = \frac{\text{Số mẫu của } C_i \text{ trong } D \text{ có giá trị } x_k}{\text{Số mẫu của } C_i \text{ trong } D} \quad (13)$$



2.1. Tập X – Tập giá trị rời rạc

VD8: Cho tập dữ liệu huấn luyện

Phân lớp cho X^{new}

= <Outlook = sunny,

Temp = cool,

Humidity = high,

Windy = strong>

Day	Outlook	Temp	Humidity	Windy	Play
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

2.1. Tập X – Tập giá trị rời rạc

VD8: Bước 1: Ước lượng $P(C_i)$ với $C_1 = \text{yes}$, $C_2 = \text{no}$ và $P(X|C_i)$

$$P(\text{Play}=\text{yes}) = 9/14 = 0.643$$

$$P(\text{Play}=\text{no}) = 5/14 = 0.357$$

- **Với thuộc tính Outlook:**

$$P(\text{Outlook}=\text{sunny} | \text{Play}=\text{yes}) = 2/9$$

$$P(\text{Outlook}=\text{sunny} | \text{Play}=\text{no}) = 3/5$$

Day	Outlook	Temp	Humidity	Windy	Play
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

2.1. Tập X – Tập giá trị rời rạc

VD8: Bước 1: Ước lượng $P(C_i)$ với $C_1 = \text{yes}$, $C_2 = \text{no}$ và $P(X|C_i)$

- **Với thuộc tính Temp:**

$$P(\text{Temp}=\text{cool} | \text{Play}=\text{yes}) = 3/9$$

$$P(\text{Temp}=\text{cool} | \text{Play}=\text{no}) = 1/5$$

- **Với thuộc tính Humidity:**

$$P(\text{Humidity}=\text{high} | \text{Play}=\text{yes}) = 3/9$$

$$P(\text{Humidity}=\text{high} | \text{Play}=\text{no}) = 4/5$$

Day	Outlook	Temp	Humidity	Windy	Play
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

2.1. Tập X – Tập giá trị rời rạc

VD8: Bước 1: Ước lượng $P(C_i)$ với $C_1 = \text{yes}$, $C_2 = \text{no}$ và $P(X|C_i)$

- **Với thuộc tính Wind:**

$$P(\text{Windy=strong} | \text{Play=yes}) = 3/9$$

$$P(\text{Windy=strong} | \text{Play=no}) = 3/5$$

Day	Outlook	Temp	Humidity	Windy	Play
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

2.1. Tập X – Tập giá trị rời rạc

VD8: Bước 2: $X^{\text{new}} = \langle \text{Outlook} = \text{sunny}, \text{Temp} = \text{cool}, \text{Humidity} = \text{high}, \text{Windy} = \text{strong} \rangle$

Tính $P(C_i) * P(X|C_i)$ với $C_1 = \text{yes}$, $C_2 = \text{no}$

$$\begin{aligned} P(\text{Play=yes}) * P(X|\text{Play=yes}) &= P(\text{Play=yes}) * P(\text{Outlook=sunny} | \text{Play=yes}) \\ &\quad * P(\text{Temp=cool} | \text{Play=yes}) * P(\text{Humidity=high} | \text{Play=yes}) \\ &\quad * P(\text{Windy=strong} | \text{Play=yes}) = 9/14 * 2/9 * 3/9 * 3/9 = \mathbf{0.0053} \end{aligned}$$

$$\begin{aligned} P(\text{Play=no}) * P(X|\text{Play=no}) &= P(\text{Play=no}) * P(\text{Outlook=sunny} | \text{Play=no}) \\ &\quad * P(\text{Temp=cool} | \text{Play=no}) * P(\text{Humidity=high} | \text{Play=no}) \\ &\quad * P(\text{Windy=strong} | \text{Play=no}) = 5/14 * 3/5 * 1/5 * 4/5 * 3/5 = \mathbf{0.021} \end{aligned}$$

$\Rightarrow X^{\text{new}}$ thuộc lớp C_2 ($\text{Play} = \text{no}$)



2.1. Tập X – Tập giá trị rời rạc

Làm trơn theo Laplace

- Để tránh trường hợp $P(x_k|C_i) = 0$ do không có mẫu nào trong tập train thỏa mãn tần số, ta làm trơn bằng cách một số mẫu ảo

$$P(C_i) \approx \frac{|C_{i,D}| + 1}{|D| + m}$$

$$P(x_k|C_i) \approx \frac{\#C_{i,D}(x_k) + 1}{|C_{i,D}| + r}$$

Với:

m: số lớp

r: số giá trị rời rạc của thuộc tính x



2.1. Tập X – Tập giá trị rời rạc

VD9: Cho tập dữ liệu huấn luyện. Sử dụng công thức làm tròn laplace, hãy phân lớp cho X^{new}

= <Outlook = overcast,

Temp = cool,

Humidity = high,

Windy = strong>

Day	Outlook	Temp	Humidity	Windy	Play
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

2.1. Tập X – Tập giá trị rời rạc

VD9: Bước 1: Ước lượng $P(C_i)$ với $C_1 = \text{yes}$, $C_2 = \text{no}$ và $P(X|C_i)$ theo công thức

làm lớn Laplace

$$P(\text{Play}=\text{yes}) = (9+1)/(14+2) = 10/16$$
$$P(\text{Play}=\text{no}) = (5+1)/(14+2) = 6/16$$

Outlook		
$P(\text{Outlook}=\text{sunny} \text{Play}=\text{yes}) = 3/12$		$P(\text{Outlook}=\text{sunny} \text{Play}=\text{no}) = 4/8$
$P(\text{Outlook}=\text{overcast} \text{Play}=\text{yes}) = 5/12$		$P(\text{Outlook}=\text{overcast} \text{Play}=\text{no}) = 1/8$
$P(\text{Outlook}=\text{rain} \text{Play}=\text{yes}) = 4/12$		$P(\text{Outlook}=\text{rain} \text{Play}=\text{no}) = 3/8$
Temperature		
$P(\text{Temperature}=\text{hot} \text{Play}=\text{yes}) = 3/12$		$P(\text{Temperature}=\text{hot} \text{Play}=\text{no}) = 3/8$
$P(\text{Temperature}=\text{mild} \text{Play}=\text{yes}) = 5/12$		$P(\text{Temperature}=\text{mild} \text{Play}=\text{no}) = 3/8$
$P(\text{Temperature}=\text{cool} \text{Play}=\text{yes}) = 4/12$		$P(\text{Temperature}=\text{cool} \text{Play}=\text{no}) = 2/8$



2.1. Tập X – Tập giá trị rời rạc

VD9: Bước 1: Ước lượng $P(C_i)$ với $C_1 = \text{yes}$, $C_2 = \text{no}$ và $P(X|C_i)$ theo công thức làm lớn Laplace

Humidity		
$P(\text{Humidity}=\text{high} \text{Play}=\text{yes}) = 4/11$		$P(\text{Humidity}=\text{high} \text{Play}=\text{no}) = 5/7$
$P(\text{Humidity}=\text{normal} \text{Play}=\text{yes}) = 7/11$		$P(\text{Humidity}=\text{normal} \text{Play}=\text{no}) = 2/7$
Windy		
$P(\text{Windy}=\text{strong} \text{Play}=\text{yes}) = 4/11$		$P(\text{Windy}=\text{strong} \text{Play}=\text{no}) = 4/7$
$P(\text{Windy}=\text{weak} \text{Play}=\text{yes}) = 7/11$		$P(\text{Windy}=\text{weak} \text{Play}=\text{no}) = 3/7$



2.1. Tập X – Tập giá trị rời rạc

VD9: Bước 2: $X^{\text{new}} = \langle \text{Outlook} = \text{overcast}, \text{Temp} = \text{cool}, \text{Humidity} = \text{high}, \text{Windy} = \text{strong} \rangle$

Tính $P(C_i) * P(X|C_i)$ với $C_1 = \text{yes}$, $C_2 = \text{no}$

$$\begin{aligned} P(\text{Play=yes}) * P(X|\text{Play=yes}) &= P(\text{Play=yes}) * P(\text{Outlook=overcast} | \text{Play=yes}) \\ &\quad * P(\text{Temp=cool} | \text{Play=yes}) * P(\text{Humidity=high} | \text{Play=yes}) * \\ &\quad P(\text{Windy=strong} | \text{Play=yes}) = 10/16 * 5/12 * 4/12 * 4/11 * 4/11 = \textcolor{red}{0.011} \end{aligned}$$

$$\begin{aligned} P(\text{Play=no}) * P(X|\text{Play=no}) &= P(\text{Play=no}) * P(\text{Outlook= overcast} | \text{Play=no}) \\ &\quad * P(\text{Temp=cool} | \text{Play=no}) * P(\text{Humidity=high} | \text{Play=no}) \\ &\quad * P(\text{Windy=strong} | \text{Play=no}) = 6/16 * 1/8 * 2/8 * 5/7 * 5/7 = \textcolor{red}{0.005} \end{aligned}$$

$\Rightarrow X^{\text{new}}$ thuộc lớp C_1 ($\text{Play} = \text{yes}$)



2.2. Tập X – Tập giá trị liên tục

- Tập $X = \langle x_1, \dots, x_n \rangle$ và x_i nhận các giá trị liên tục thì $P(x_k | C_i)$ được tính dựa trên phân bố Gauss với giá trị trung bình μ và độ lệch σ

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (14)$$

- Khi đó:

$$P(x_k | C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) \quad (15)$$



3. Ưu điểm và nhược điểm

- **Ưu điểm:**

- Dễ dàng cài đặt
- Thời gian thi hành nhanh
- Đạt kết quả tốt trong phần lớn các trường hợp

- **Nhược điểm:**

- Giả thiết về tính độc lập điều kiện của thuộc tính làm giảm độ chính xác



Naïve Bayes: Bài tập

BT12

Cho tập dữ liệu với thuộc tính quyết định là buy_computer.

Sử dụng Naïve Bayes để xác định lớp cho X = (age ≤ 30 ,

Income = medium,

Student = yes,

Credit_rating = fair)

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

5

Phương pháp dựa trên thể hiện

1. Giới thiệu
2. Thuật toán K-NN



1. Giới thiệu

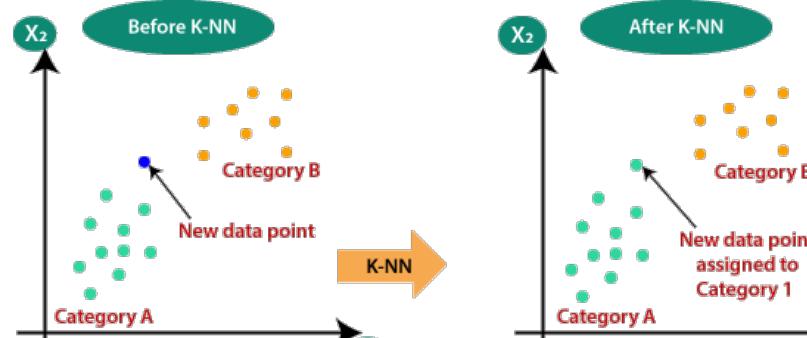
- **Phương pháp phân lớp dựa trên thể hiện (Instance-based)**
 - Lưu trữ các mẫu/ đối tượng huấn luyện và chỉ xử lý khi có yêu cầu phân lớp mẫu/ đối tượng mới
 - Đưa mẫu/ đối tượng mới vào lớp mà gần với chúng nhất



1. Giới thiệu

- Một số phương pháp

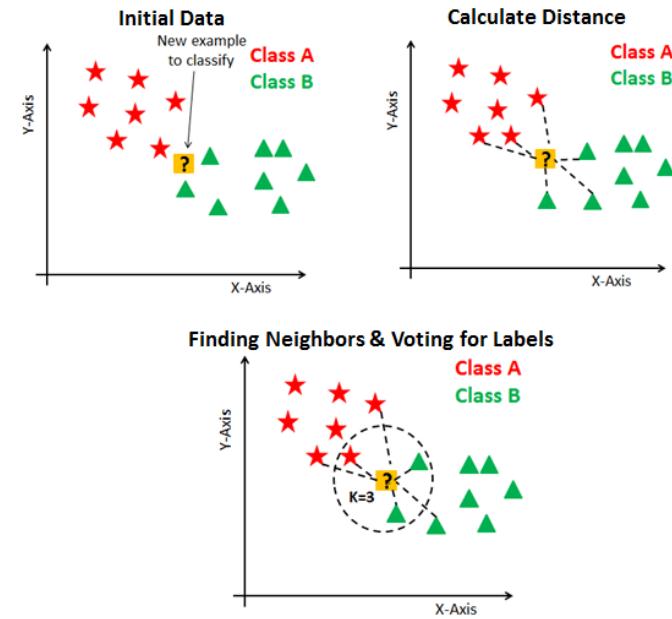
- Thuật toán K-NN (K-Nearest Neighbor)
- Hồi qui với trọng số cục bộ(Locally weighted regression)
- Suy luận dựa trên trường hợp (Case-based reasoning)



2. K-NN (K-Nearest Neighbor)

- Thuật toán xác định lớp cho mẫu mới E:

- Tính khoảng cách giữa E và tất cả các mẫu trong tập train
- Chọn k mẫu gần với E nhất (k mẫu láng giềng)
- Gán E vào lớp có nhiều mẫu nhất trong số k mẫu láng giềng đó (hoặc E nhận giá trị trung bình của k mẫu)



2. K-NN (K-Nearest Neighbor)

- **Tính khoảng cách giữa 2 mẫu**

- Mỗi mẫu: tập thuộc tính số
- Khoảng cách Euclide giữa $X = (x_1, x_2, \dots, x_n)$ và $Y = (y_1, y_2, \dots, y_n)$:

$$D(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (16)$$

- Khi thực hiện so sánh, có thể bỏ qua căn bậc 2



2. K-NN (K-Nearest Neighbor)

VD10: Tính khoảng cách giữa Ricky và Steven



Steven:

Age: 35

Income: 95K

No. of credit card: 3



Ricky:

Age: 41

Income: 215K

No. of credit card: 2

$$D(\text{Steven}, \text{Ricky}) = \sqrt{(35 - 41)^2 + (95 - 215)^2 + (3 - 2)^2}$$

- Các thuộc tính có giá trị lớn sẽ ảnh hưởng nhiều đến khoảng cách. VD: income
- Các thuộc tính có miền giá trị khác nhau

Cần chuẩn hóa giá trị
thuộc tính



2. K-NN (K-Nearest Neighbor)

- **Chuẩn hóa dữ liệu:** Ánh xạ các giá trị vào đoạn [0. 1] theo công thức:

$$a_i = \frac{v_i - \min v_i}{\max v_i - \min v_i} \quad (17)$$

- v_i : giá trị thực tế của thuộc tính i
- a_i : giá trị đã chuẩn hóa của thuộc tính i



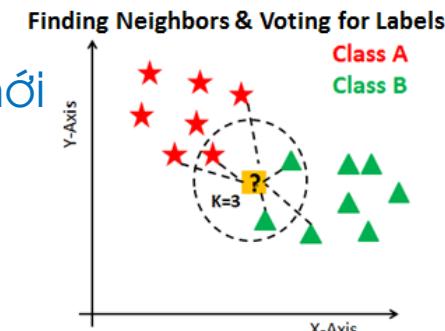
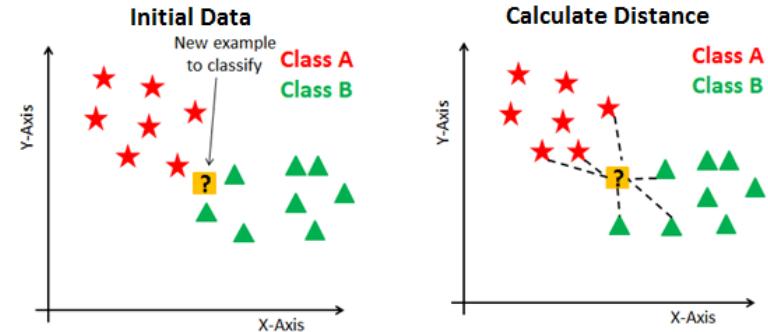
2. K-NN (K-Nearest Neighbor)

- Ưu điểm

- Dễ sử dụng và cài đặt
- Xử lý tốt với dữ liệu nhiễu

- Nhược điểm

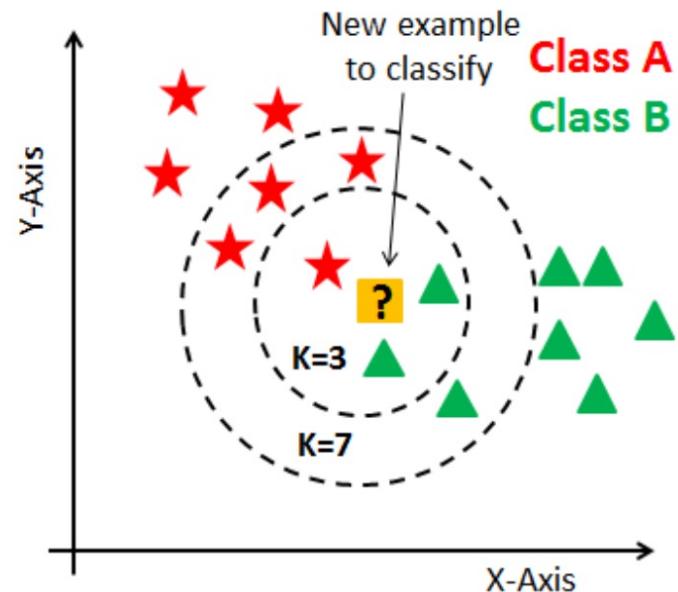
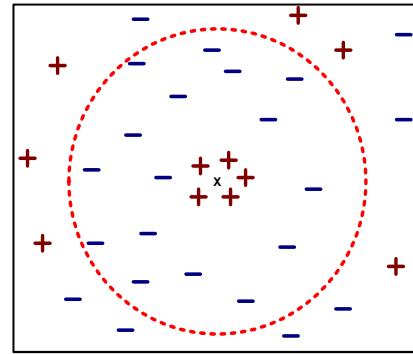
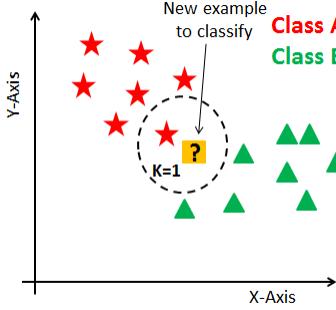
- Cần lưu tất cả các mẫu
- Cần nhiều thời gian để xác định lớp cho mẫu mới
- Phụ thuộc vào giá trị k do người dùng chọn
- Thuộc tính phi số?



2. K-NN (K-Nearest Neighbor)

- Phụ thuộc vào giá trị k do người dùng chọn

- k quá nhỏ: nhạy cảm với nhiễu
- k quá lớn: vùng lân cận có thể chứa các điểm của lớp khác



K-NN: Bài tập

BT13

Chuẩn hóa dữ liệu và sử dụng thuật toán K-NN để xác định lớp cho Minh

1. Với $k = 3$
2. Với $k = 5$

Customer	Age	Income (K)	No. cards	Response
Lâm	35	35	3	Yes
Hưng	22	50	2	No
Mai	28	40	1	Yes
Lan	45	100	2	No
Thúy	20	30	3	Yes
Tuấn	34	55	2	No
Ninh	63	200	1	No
Vân	55	140	2	No
Thiện	59	170	1	No
Ngọc	25	40	4	Yes
Minh	39	41	2	???



6

Mạng neural nhân tạo

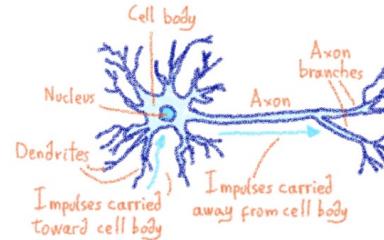


1. Giới thiệu
2. Cấu trúc một neural nhân tạo
3. Perceptron
4. Kiến trúc ANN
5. Huấn luyện trong ANN
6. Thuật toán lan truyền ngược
7. Siêu tham số
8. Ưu và nhược điểm của ANN

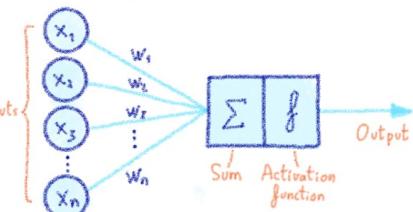
1. Giới thiệu

- **Artificial neural network(ANN) (mạng neural nhân tạo)**
 - Mô phỏng các hệ thần kinh sinh học (não người)
 - Một cấu trúc/mạng được tạo thành từ sự kết nối của các neuron nhân tạo, mỗi kết nối có một trọng số được liên kết với nó
- **Neuron**
 - Có input, output
 - Thực hiện những tính toán nội bộ

Biological Neuron

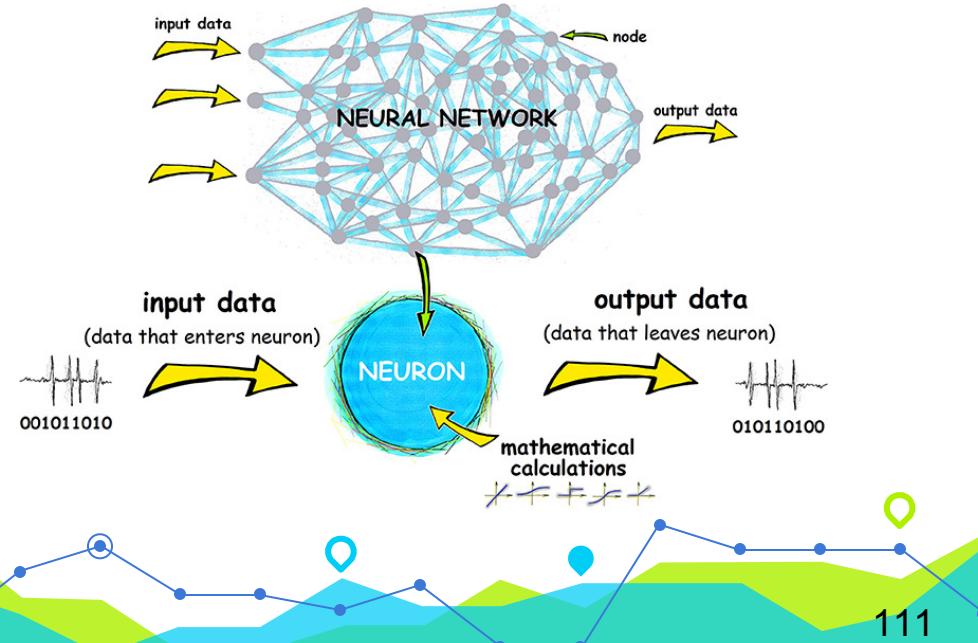


Artificial Neuron



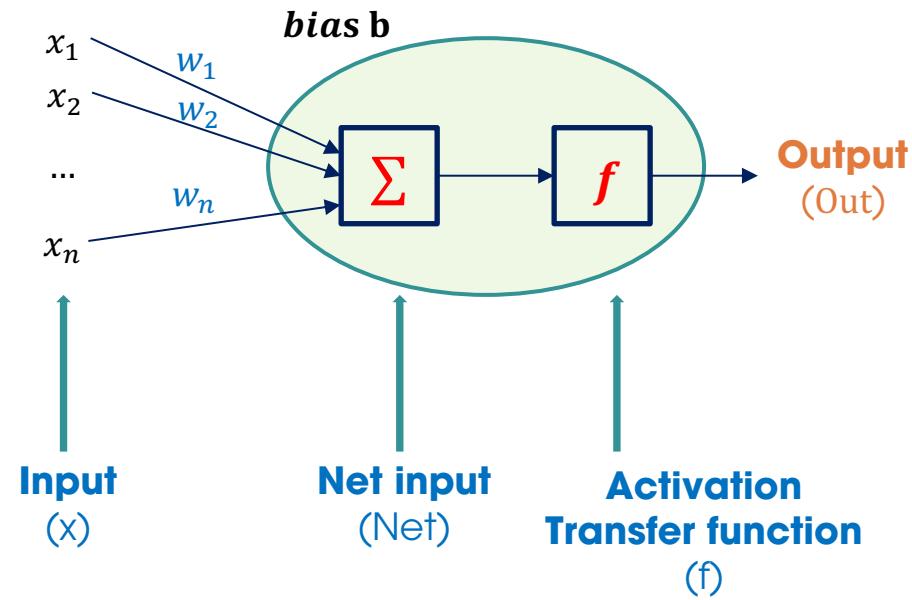
1. Giới thiệu

- ANN: một cấu trúc xử lý thông tin song song và phi tập trung cao
- ANN có khả năng học, nhớ và tổng quát hóa từ dữ liệu huấn luyện
- Khả năng của một ANN phụ thuộc vào:
 - Kiến trúc mạng
 - Đặc trưng input/ output
 - Thuật toán huấn luyện
 - Dữ liệu huấn luyện



2. Cấu trúc một neuron nhân tạo

- Input: $\{x_i, i = 1 \dots n\}$
- Mỗi x_i có 1 trọng số w_i
- Bias b
- Net input: tổng hợp các input
Net (w, x)
- Hàm kích hoạt f : tính toán output của 1 neuron
- Output: **Out** = $f(\text{Net } (w, x))$



Có thể xem bias b là w_0 với $x_0 = 1$

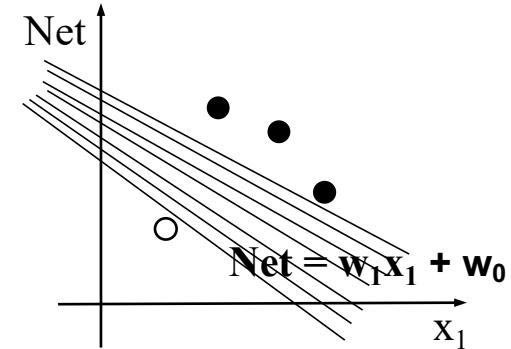
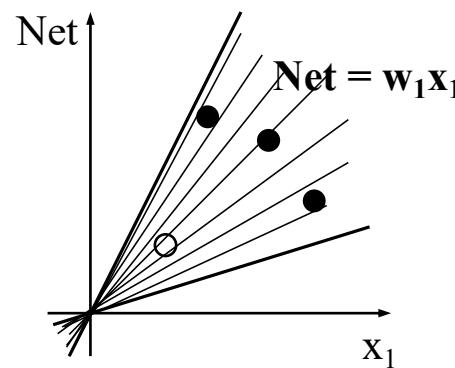


2. Cấu trúc một neural nhân tạo

- **Net Input:** thường được tính bằng 1 công thức tuyến tính

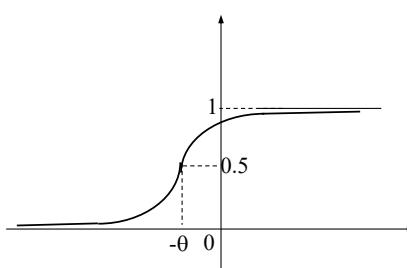
$$\text{Net}(w, x) = \sum_{i=1}^n w_i x_i + b = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$

Bias b : có thể phân lớp,
tách biệt lớp tốt hơn

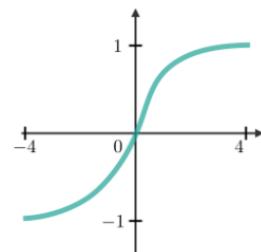


2. Cấu trúc một neural nhân tạo

- **Activation function:** thường được tính bằng 1 hàm phi tuyến



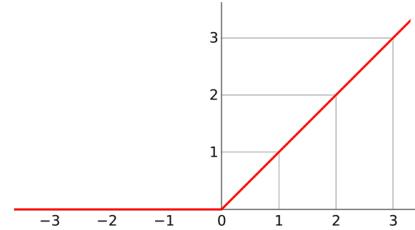
Sigmoid



Tanh Function
(Hyperbolic)

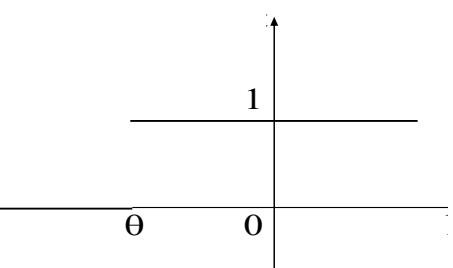
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$Tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$



Rectified linear unit
(ReLU)

$$f(z) = \max(0, z)$$



Binary hard-limiter

$$f(z) = \begin{cases} 0, & \text{if } z < \theta \\ 1, & \text{if } z \geq \theta \end{cases}$$

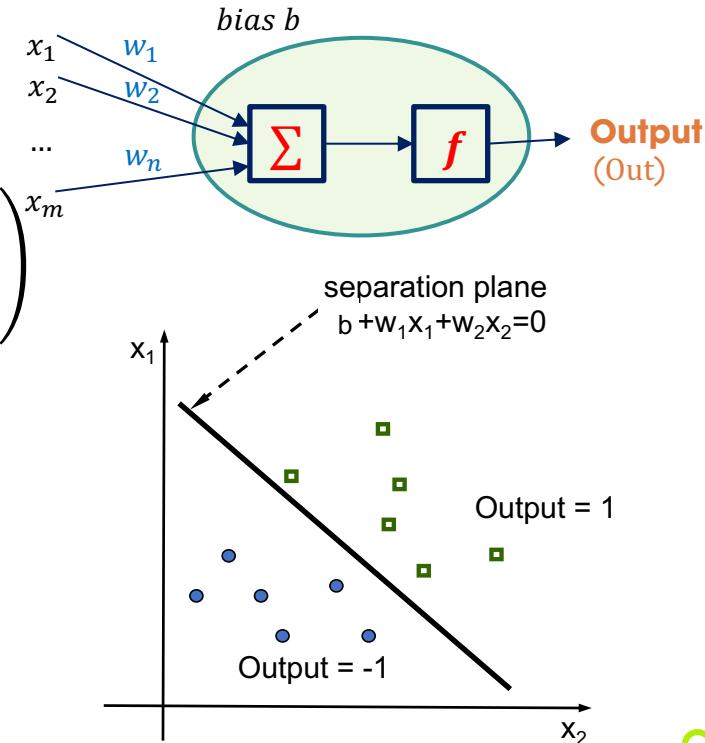


3. Perceptron

- ANN đơn giản nhất (chỉ gồm 1 neuron)
- Sử dụng hard-limited activation function

$$Out = \text{sign}(\text{Net}(w, x)) = \text{sign} \left(\sum_{i=1}^n w_j x_j + b \right)$$

- Với mỗi input x :
 - Nếu $\text{Net}(w, x) > 0$, output = 1
 - Ngược lại, output = -1



3. Perceptron

- Tập dữ liệu huấn luyện $D = \{(x, d)\}$
 - x : input vector
 - d : target output (1 hoặc -1)
- Mục tiêu của quá trình huấn luyện: xác định một vector trọng số (w) cho phép tạo ra predict output (-1 hoặc 1) cho mỗi điểm dữ liệu
- Đối với điểm dữ liệu x được phân lớp chính xác, w không thay đổi
- Nếu $d = 1$, mà predict output = -1: thay đổi w để $\text{Net}(w, x)$ tăng lên
- Nếu $d = -1$, mà predict output = 1: thay đổi w để $\text{Net}(w, x)$ giảm đi

3. Perceptron

Perceptron_batch (\mathbf{D} , η)

Initialize \mathbf{w} ($w_i \leftarrow$ an initial (small) random value)

do

$\Delta\mathbf{w} \leftarrow 0$

for each training instance $(\mathbf{x}, d) \in \mathbf{D}$

Compute the real output value Out

if ($Out \neq d$)

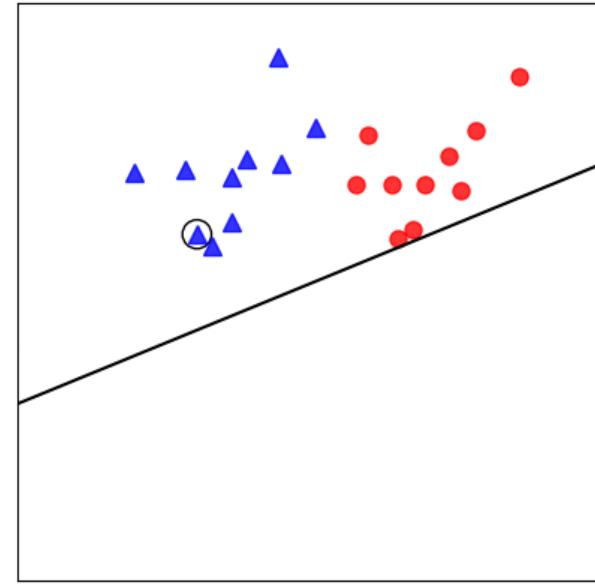
$\Delta\mathbf{w} \leftarrow \Delta\mathbf{w} + \eta(d - Out)\mathbf{x}$

end for

$\mathbf{w} \leftarrow \mathbf{w} + \Delta\mathbf{w}$

until all the training instances in \mathbf{D} are correctly classified

return \mathbf{w}



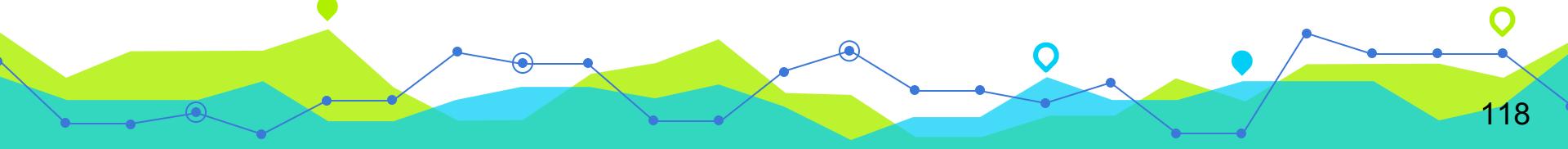
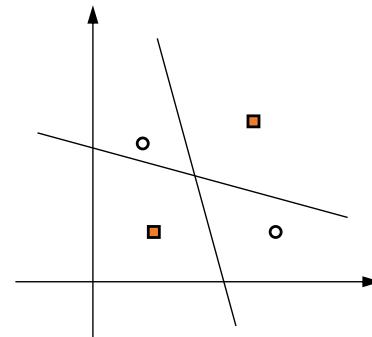
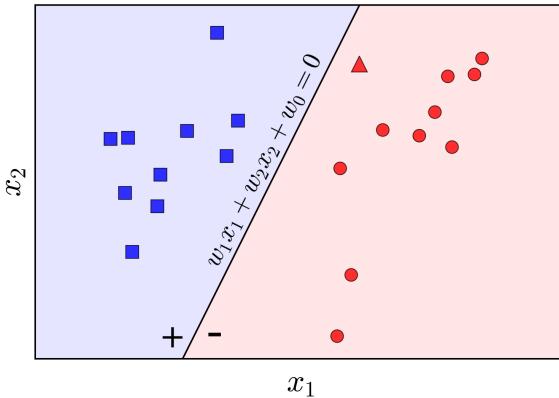
PLA: iter 0/18



117

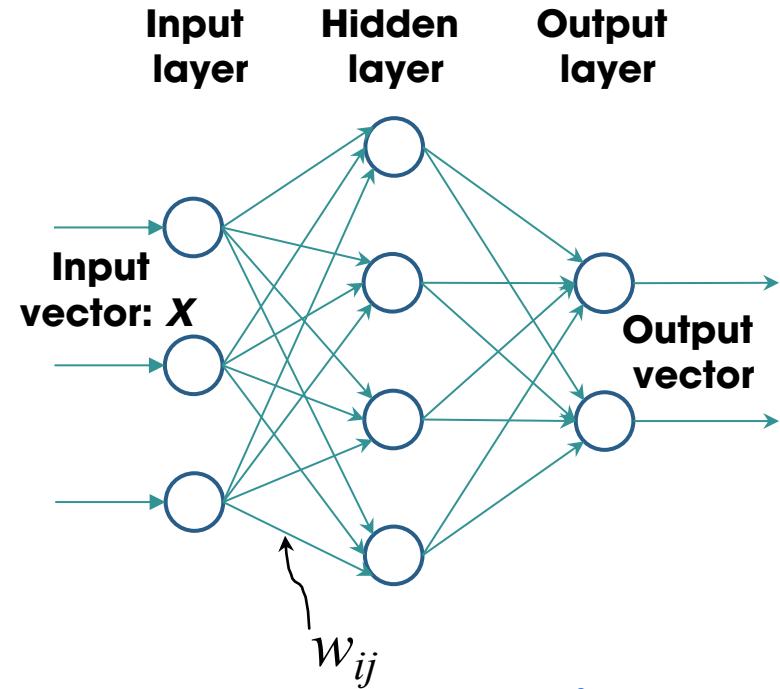
3. Perceptron

- Thuật toán huấn luyện perceptron
được chứng minh là hội tụ nếu:
 - Điểm dữ liệu có thể phân tách tuyến tính
 - Sử dụng tốc độ học η đủ nhỏ
- Thuật toán huấn luyện perceptron có thể không hội tụ nếu các điểm dữ liệu không thể phân tách tuyến tính



4. Kiến trúc ANN

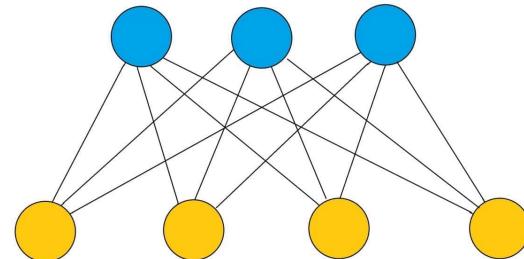
- **Kiến trúc của ANN** xác định bởi:
 - Số lượng input, output
 - Số tầng (layer)
 - Số neuron trong mỗi layer
 - Số lượng kết nối của mỗi neuron
 - Cách kết nối giữa các neuron (cùng 1 layer, giữa các layer)
- **ANN có:**
 - 1 input layer
 - 1 output layer
 - 0, 1, hoặc nhiều hidden layer



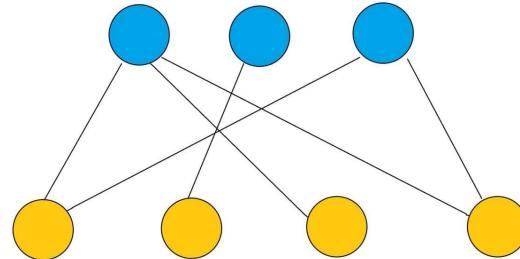
4. Kiến trúc ANN

- **Fully connection (Dense)**: output của các neuron của layer trước đó kết nối với **tất cả** các neuron của layer kế tiếp.
- **Not Fully connection (Sparse)**: output của các neuron của layer trước đó chỉ kết nối với **một số** neuron của layer kế tiếp.

Densely Connected



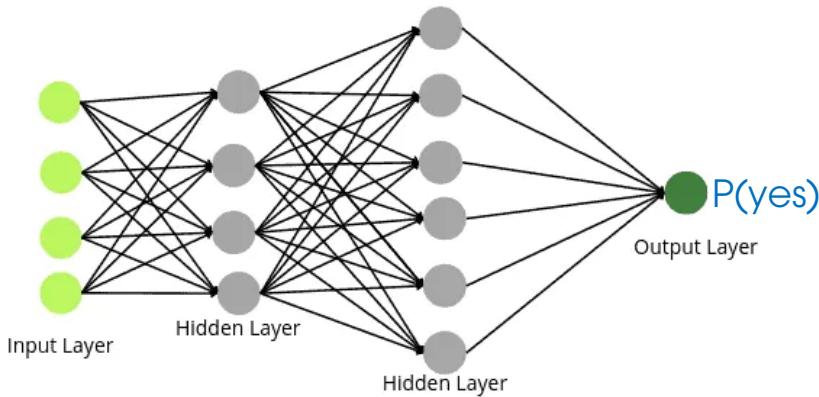
Sparingly Connected



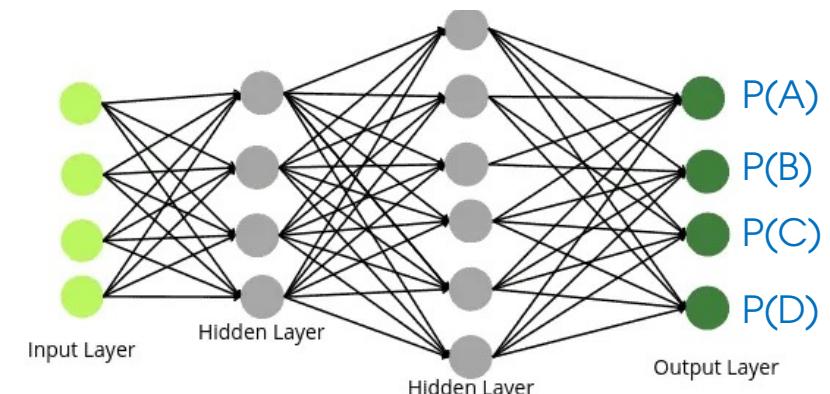
4. Kiến trúc ANN

- **Với bài toán phân lớp: số lượng neuron của output layer**

Neural network for Binary Classification



Neural network for Multi-classification



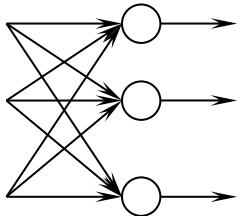
4. Kiến trúc ANN

- **Feed-forward network (mạng lan truyền tiến)**: nếu không có bất kỳ output nào của 1 neuron là input của 1 neuron khác trong cùng layer hoặc layer trước đó
- **Feedback network (mạng phản hồi)**: output của 1 neuron là input của neuron cùng layer hoặc layer trước đó.
 - Nếu feedback kết nối với input của các neuron của cùng một layer, được gọi là **lateral feedback** (mạng phản hồi bên).
- **Recurrent network (mạng truy hồi)**: feedback network với các vòng khép kín

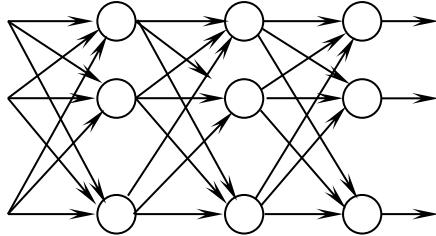


4. Kiến trúc ANN

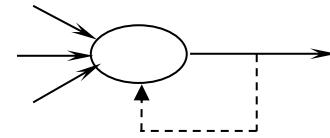
Feed-forward network



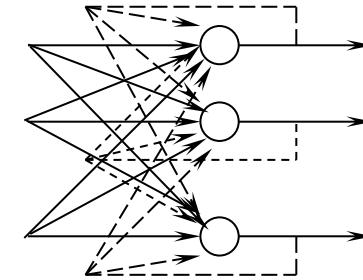
Feed-forward network with multiple layers



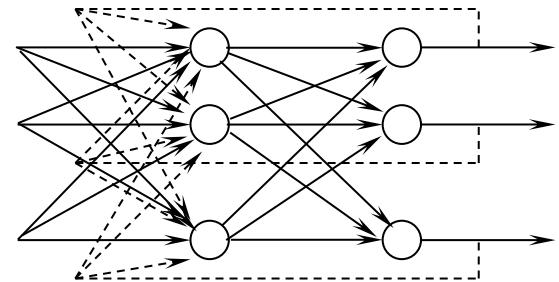
A neural with feedback to itself



Recurrent network with single layer



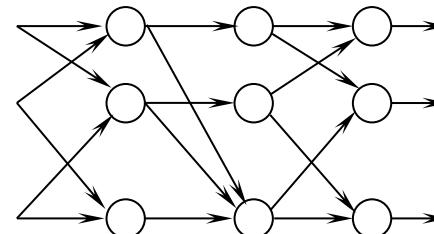
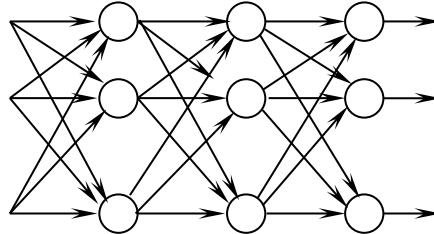
Recurrent network with multiple layers



5. Huấn luyện trong ANN

- 2 hình thức học trong ANN

- **Parameter learning:** Mục tiêu là **học và điều chỉnh trọng số** của các kết nối trong ANN, với cấu trúc mạng cố định. Được thực hiện bằng cách cực tiểu Loss function
- **Structure learning:** Mục tiêu là **học cấu trúc mạng**, bao gồm số lượng neuron và các loại kết nối giữa các neuron và trọng số



5. Huấn luyện trong ANN

- Huấn luyện ANN: **học và điều chỉnh các trọng số w, bias b** của mạng từ tập dữ liệu huấn luyện D
- Tối ưu hóa hàm lỗi, hàm mất mát (loss function) để giảm thiểu sai lệch giữa giá trị dự đoán và giá trị thực tế

$$\text{Loss}(w, b) = \frac{1}{|D|} \sum_{x \in D} L(d_x, \text{out}(x))$$

- $\text{Out}(x)$: predict output của ANN
- d_x : target output của input x
- $L(\cdot)$: hàm mất mát bất kỳ phù hợp với bài toán
- Các cách ký hiệu: Loss , \mathcal{L} , J



5. Huấn luyện trong ANN

Một số loss function phổ biến:

- Regression:

- Mean Squared Error (MSE)

$$MSE_x = (y_x - \hat{y}_x)^2$$

$$\mathcal{L}_{MSE} = \frac{1}{|D|} \sum_{x \in D} MSE_x$$

- Mean Squared Logarithmic Error (MSLE)

$$MSLE_x = [\log(1 + y_x) - \log(1 + \hat{y}_x)]^2$$

$$\mathcal{L}_{MSLE} = \frac{1}{|D|} \sum_{x \in D} MSLE_x$$

- Mean Absolute Error (MAE)

$$MAE_x = |y_x - \hat{y}_x|$$

$$\mathcal{L}_{MAE} = \frac{1}{|D|} \sum_{x \in D} MAE_x$$

y_x : giá trị thực tế của điểm dữ liệu x
 \hat{y}_x : giá trị mô hình dự đoán cho điểm dữ liệu x



5. Huấn luyện trong ANN

Một số loss function phổ biến:

- Binary Classification:

- Binary Cross-Entropy (Log Loss) $y_x = 0$ hoặc 1 ; $\hat{y}_x \in [0,1]$

$$BCE_x = -(y_x \log(\hat{y}_x) + (1 - y_x) \log(1 - \hat{y}_x))$$

$$\mathcal{L}_{BCE} = \frac{1}{|D|} \sum_{x \in D} BCE_x$$

- Hinge Loss $y_x = -1$ hoặc 1

$$Hinge_x = \max(0, 1 - y_x \cdot \hat{y}_x)$$

$$\mathcal{L}_{Hinge} = \frac{1}{|D|} \sum_{x \in D} Hinge_x$$

- Squared Hinge Loss $y_x = -1$ hoặc 1

$$SquaredHinge_x = (\max(0, 1 - y_x \cdot \hat{y}_x))^2$$

$$\mathcal{L}_{SquaredHinge} = \frac{1}{|D|} \sum_{x \in D} SquaredHinge_x$$

y_x : nhãn thực tế của điểm dữ liệu x

\hat{y}_x : xác suất dự đoán của mô hình cho điểm dữ liệu x



5. Huấn luyện trong ANN

Một số loss function phổ biến:

- Multi-class Classification:
 - Multi-class Cross-Entropy (Categorical Cross-Entropy)

$$CE_x = \sum_{j=1}^C y_{xj} \log(\hat{y}_{xj})$$

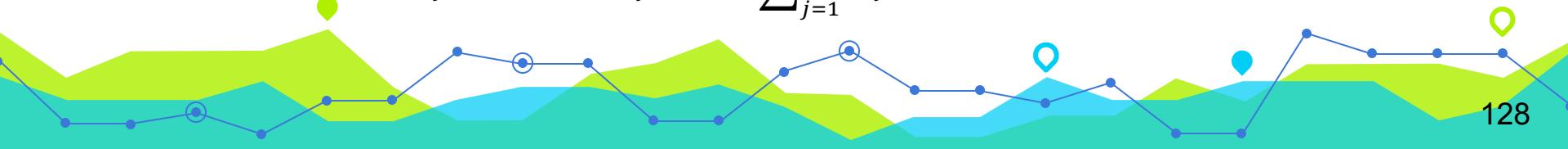
$$\mathcal{L}_{CE} = \frac{1}{|D|} \sum_{x \in D} CE_x$$

C: số lớp

y_{xj}: nhãn thực tế của điểm dữ liệu x cho lớp j

ŷ_{xj}: xác suất dự đoán của mô hình cho điểm dữ liệu x cho lớp j

$$y_{xj} = 0 \text{ hoặc } 1; \hat{y}_{xj} \in [0,1]; \sum_{j=1}^C \hat{y}_{xj} = 1$$



5. Huấn luyện trong ANN

Một số loss function phổ biến:

- Multi-class Classification:
 - Sparse Multi-class Cross-Entropy

$$SCE_x = - \log(\hat{y}_{y_x})$$

$$\mathcal{L}_{SCE} = \frac{1}{|D|} \sum_{x \in D} SCE_x$$

\hat{y}_{y_x} : xác suất dự đoán của mô hình cho lớp thực sự của điểm dữ liệu x
 $\hat{y}_{y_x} \in [0,1]$



5. Huấn luyện trong ANN

Một số loss function phổ biến:

- Multi-class Classification:
 - Kullback Leibler Divergence

$$D_{KL}(P\|Q) = \sum_{i=1}^c P(i) \log \left(\frac{P(i)}{Q(i)} \right)$$

$P(i)$: phân bố xác suất các nhãn thực tế của dữ liệu

$Q(i)$: phân bố xác suất các nhãn được mô hình dự đoán



5. Huấn luyện trong ANN

Minimize Loss with gradients

- Vector Gradient của Loss ($\nabla \mathcal{L}$)

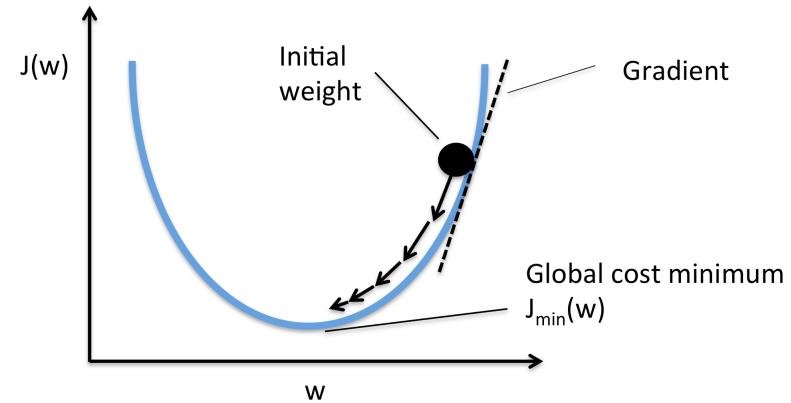
$$\nabla \mathcal{L}(w) = \left(\frac{\partial \mathcal{L}}{\partial w_1}, \frac{\partial \mathcal{L}}{\partial w_2}, \dots, \frac{\partial \mathcal{L}}{\partial w_N} \right)$$

N : số lượng các trọng số w

- Cập nhật w theo ∇w

$$\Delta w = -\eta \cdot \nabla \mathcal{L}(w); \quad \Delta w_i = -\eta \frac{\partial \mathcal{L}}{\partial w_i} \quad (i = 1, \dots, N)$$

η : learning rate



5. Huấn luyện trong ANN

Gradient_descent_incremental (\mathbf{D} , η)

Initialize \mathbf{w} ($w_i \leftarrow$ an initial (small) random value)

do

for each training instance $(x, d) \in \mathbf{D}$

 Compute the network output
 for each weight component w_i

$$w_i \leftarrow w_i - \eta \frac{\partial \mathcal{L}}{\partial w_i}$$

end for

end for

until (stopping criterion satisfied)

return \mathbf{w}

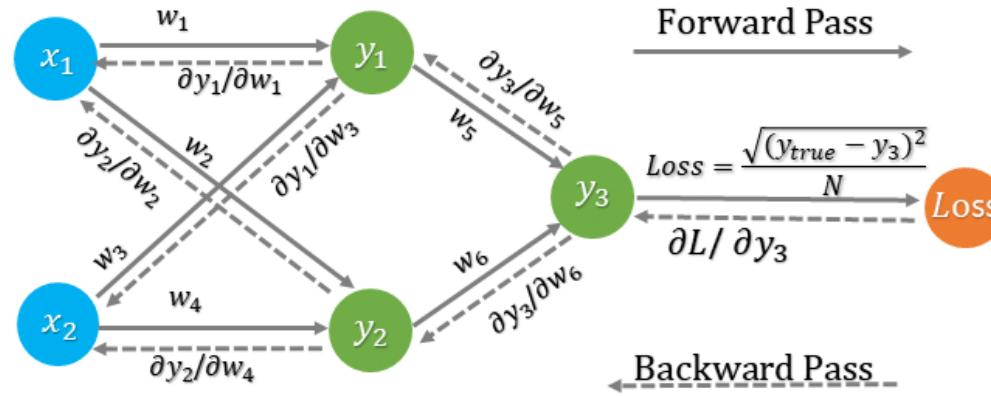
Stopping criterion: epochs, threshold error, ...

If we take a small subset (mini-batch) randomly from \mathbf{D} to update the weights, we will have mini-batch training



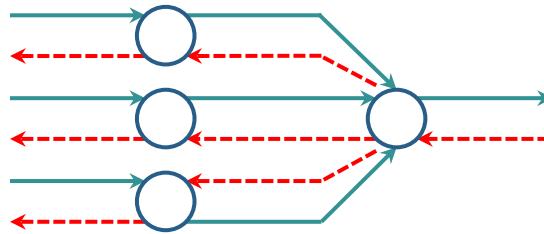
6. Thuật toán lan truyền ngược

- **Backpropagation algorithm (BP)**
- BP được sử dụng để học các trọng số của ANN với cấu trúc mạng cố định
- BP áp dụng gradient descent để cập nhật trọng số, để **cực tiểu Loss** giữa output thực tế và output mong muốn (đối với dữ liệu huấn luyện)



6. Thuật toán lan truyền ngược

- BP bao gồm 2 phase :
 - **Signal Forward Propagation:** Các input được tính toán, chuyển tiếp từ input layer đến output layer (qua các hidden layer).
 - **Error Backward Propagation:** tính toán Loss (khác biệt giữa output thực tế với output mong muốn), truyền Loss ngược lại qua mạng bắt đầu từ lớp output layer, từ layer này sang layer trước, đến input layer để cập nhật các trọng số



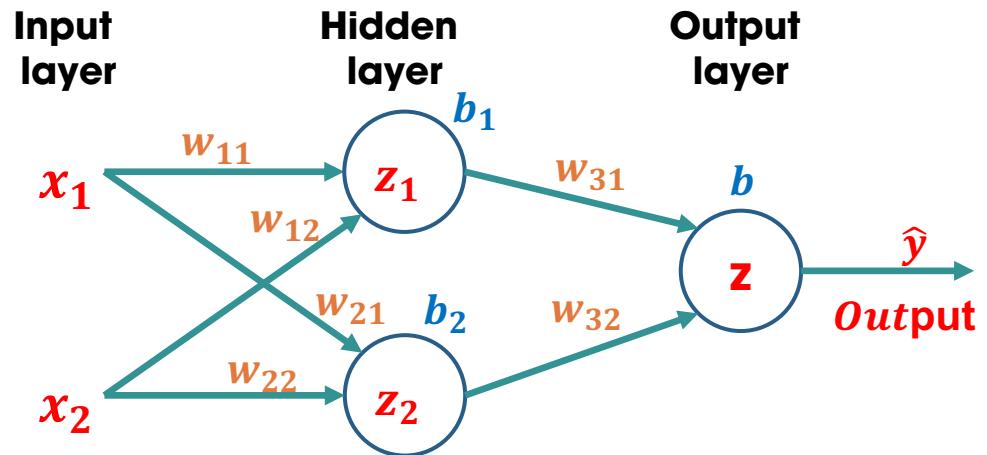
- **Signal forward phase:**
 - Forward signals via the network
- ← **Error forward phase:**
 - Calculate error at the output
 - Error back-propagation



6. Thuật toán lan truyền ngược

VD11: Xét kiến trúc ANN mô phỏng XOR

x1	x2	Target output
0	0	0
0	1	1
1	1	0
1	0	1

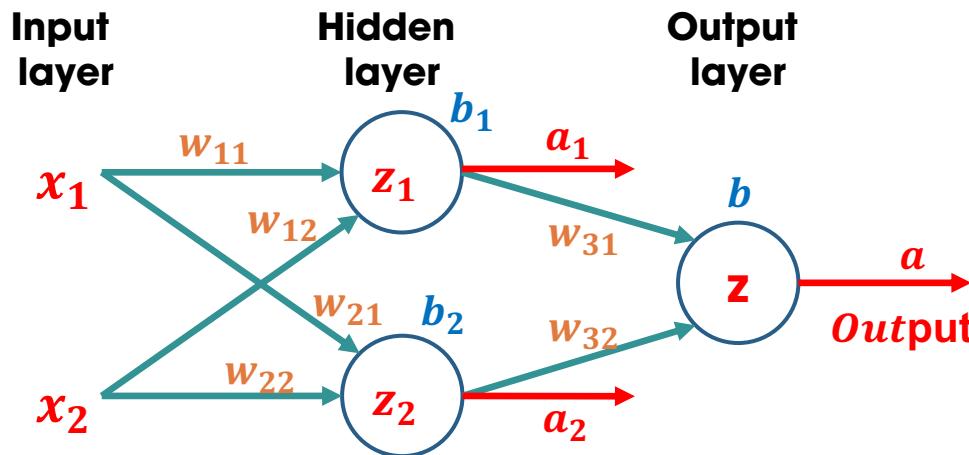


Activation function: Sigmoid $\sigma(z) = \frac{1}{1 + e^{-z}}$



6. Thuật toán lan truyền ngược

VD11: Forward Propagation



Activation function: Sigmoid $\sigma(z) = \frac{1}{1 + e^{-z}}$

$$z_1 = w_{11}x_1 + w_{12}x_2 + b_1$$

$$a_1 = \sigma(z_1) = \frac{1}{1 + e^{-z_1}}$$

$$z_2 = w_{21}x_1 + w_{22}x_2 + b_2$$

$$a_2 = \sigma(z_2) = \frac{1}{1 + e^{-z_2}}$$

$$z = w_{31}a_1 + w_{32}a_2 + b$$

$$a = \sigma(z) = \frac{1}{1 + e^{-z}}$$



6. Thuật toán lan truyền ngược

VD11: Forward Propagation

Giả sử khởi tạo các trọng số và có kết quả của quá trình Propagation như sau:

x1	x2	w11	w12	b1	w21	w22	b2	z1	a1	z2	a2	w31	w32	b	z	a
0	0	0.16	-0.07	0.37	-0.66	0.63	-0.98	0.37	0.59	-0.98	0.27	0.41	-0.90	0.35	0.35	0.59
0	1	0.16	-0.07	0.37	-0.66	0.63	-0.98	0.30	0.57	-0.35	0.41	0.41	-0.90	0.35	0.21	0.55
1	1	0.16	-0.07	0.37	-0.66	0.63	-0.98	0.53	0.63	-1.64	0.16	0.41	-0.90	0.35	0.46	0.61
1	0	0.16	-0.07	0.37	-0.66	0.63	-0.98	0.46	0.61	-1.01	0.27	0.41	-0.90	0.35	0.36	0.59



6. Thuật toán lan truyền ngược

VD11: Forward Propagation

a	y	BCE_x
0.59	0	0.881
0.55	1	0.592
0.61	1	0.488
0.59	0	0.889
$\mathcal{L}_{BCE}(w, b)$		0.713

Tính Loss: sử dụng Binary Cross-Entropy (Log Loss)

$$BCE_x = L(y, a) = -(y \log(a) + (1 - y) \log(1 - a))$$

Nếu $y = 0$: $L(y, a) = -(\log(1 - \hat{y}_x))$

Nếu $y = 1$: $L(y, a) = -(\log(\hat{y}_x))$

$$\mathcal{L}_{BCE}(w, b) = \frac{1}{|D|} \sum_{x \in D} BCE_x$$



6. Thuật toán lan truyền ngược

VD11: Backward Propagation

Điều chỉnh w, b sao cho cực tiểu Loss

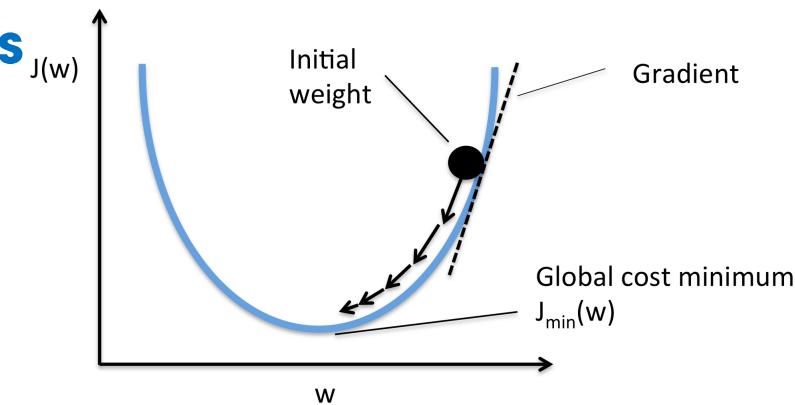
$$w := w - \eta * \delta w \quad \eta \ll 1: learning rate$$

$$b := b - \eta * \delta b$$

$$\delta w = \frac{d\mathcal{L}}{dw} = \frac{1}{|D|} \sum_{x \in D} \delta w_x$$

$$\delta b = \frac{d\mathcal{L}}{db} = \frac{1}{|D|} \sum_{x \in D} \delta b_x$$

w, b: đại diện cho tất cả weight và bias



6. Thuật toán lan truyền ngược

VD11: Backward Propagation

Xét điểm dữ liệu x: $L(y, a) = -(y \log(a) + (1 - y) \log(1 - a))$

$$a = \sigma(z) = \frac{1}{1 + e^{-z}}$$

Áp dụng quy tắc chuỗi đạo hàm và L phụ thuộc vào a, a phụ thuộc vào z

$$\delta_w = \frac{dL}{dw} = \frac{dL}{da} \cdot \frac{da}{dz} \cdot \frac{dz}{dw}$$

$$\delta_b = \frac{dL}{db} = \frac{dL}{da} \cdot \frac{da}{dz} \cdot \frac{dz}{db}$$

Tính đạo hàm từng phần

$$\frac{dL}{da} = -\left(\frac{y}{a} + \frac{1-y}{1-a}\right)$$

$$\delta_w = \delta_z \cdot \frac{dz}{dw}$$

$$\frac{da}{dz} = \frac{d}{dz} \left(\frac{1}{1 + e^{-z}} \right) = a(1 - a)$$

Suy ra:

$$\delta_b = \delta_z \cdot \frac{dz}{db}$$

$$\delta_z = \frac{dL}{da} \cdot \frac{da}{dz} = -\left(\frac{y}{a} + \frac{1-y}{1-a}\right) a(1 - a) = a - y$$

$$\text{Với } \delta_z = a - y$$



6. Thuật toán lan truyền ngược

VD11: Backward Propagation

Tính các $\frac{dz}{dw}$ và $\frac{dz}{db}$

Với $z = w_{31}a_1 + w_{32}a_2 + b$

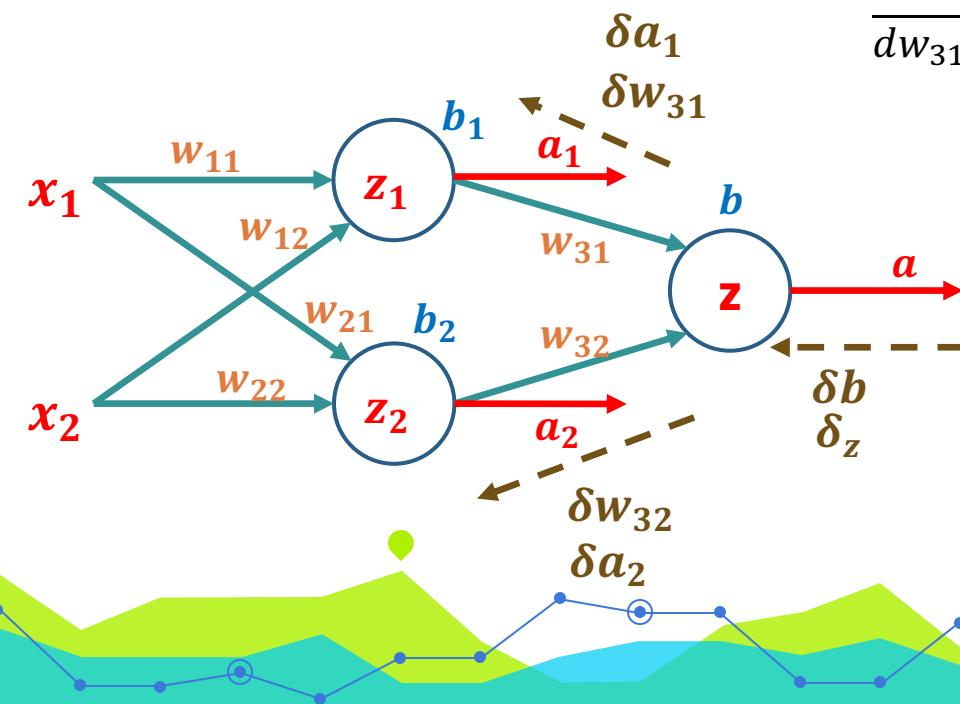
$$\frac{dz}{dw_{31}} = a_1; \frac{dz}{da_1} = w_{31}; \frac{dz}{dw_{32}} = a_2; \frac{dz}{da_2} = w_{32}; \frac{dz}{db} = 1$$

Suy ra:

$$\delta w_{31} = \delta z \cdot \frac{dz}{dw_{31}} = \delta z \cdot a_1$$

$$\delta w_{32} = \delta z \cdot \frac{dz}{dw_{32}} = \delta z \cdot a_2$$

$$\delta b = \delta z \cdot \frac{dz}{db} = \delta z$$



6. Thuật toán lan truyền ngược

VD11: Backward Propagation

Với neuron z_1 : $z_1 = w_{11}x_1 + w_{12}x_2 + b_1$

$$\delta w = \frac{dL}{dw} = \frac{dL}{da} \cdot \frac{da}{dz} \cdot \frac{dz}{da_1} \cdot \frac{da_1}{dz_1} \cdot \frac{dz_1}{dw}$$

Tính đạo hàm từng phần

$$\delta a_1 = \frac{dL}{da} \cdot \frac{da}{dz} \cdot \frac{dz}{da_1} = \delta z \cdot \frac{dz}{da_1} = \delta z \cdot w_{31}$$

$$\frac{da_1}{dz_1} = \frac{d}{dz_1} \left(\frac{1}{1 + e^{-z_1}} \right) = a_1(1 - a_1)$$

$$\delta z_1 = \frac{dL}{da} \cdot \frac{da}{dz} \cdot \frac{dz}{da_1} \cdot \frac{da_1}{dz_1} = a_1(1 - a_1) \cdot \delta a_1$$

$$\frac{dz_1}{dw_{11}} = x_1; \frac{dz_1}{dw_{12}} = x_2; \frac{dz_1}{db_1} = 1$$

$$a_1 = \sigma(z_1) = \frac{1}{1 + e^{-z_1}}$$

$$\delta b = \frac{dL}{db} = \frac{dL}{da} \cdot \frac{da}{dz} \cdot \frac{dz}{da_1} \cdot \frac{da_1}{dz_1} \cdot \frac{dz_1}{db}$$

Suy ra:

$$\delta w_{11} = \delta z_1 \cdot x_1$$

$$\delta w_{12} = \delta z_1 \cdot x_2$$

$$\delta b_1 = \delta z_1$$



6. Thuật toán lan truyền ngược

VD11: Backward Propagation

Với neuron z_2 : $z_2 = w_{21}x_1 + w_{22}x_2 + b_2$

$$\delta w = \frac{dL}{dw} = \frac{dL}{da} \cdot \frac{da}{dz} \cdot \frac{dz}{da_2} \cdot \frac{da_2}{dz_2} \cdot \frac{dz_2}{dw}$$

Tính đạo hàm từng phần

$$\delta a_2 = \frac{dL}{da} \cdot \frac{da}{dz} \cdot \frac{dz}{da_2} = \delta z \cdot \frac{dz}{da_2} = \delta z \cdot w_{32}$$

$$\frac{da_2}{dz_2} = \frac{d}{dz_2} \left(\frac{1}{1 + e^{-z_2}} \right) = a_2(1 - a_2)$$

$$\delta z_2 = \frac{dL}{da} \cdot \frac{da}{dz} \cdot \frac{dz}{da_2} \cdot \frac{da_2}{dz_2} = a_2(1 - a_2) \cdot \delta a_2$$

$$\frac{dz_2}{dw_{21}} = x_1; \frac{dz_2}{dw_{22}} = x_2; \frac{dz_2}{db_2} = 1$$

$$a_2 = \sigma(z_2) = \frac{1}{1 + e^{-z_2}}$$
$$\delta b = \frac{dL}{db} = \frac{dL}{da} \cdot \frac{da}{dz} \cdot \frac{dz}{da_2} \cdot \frac{da_2}{dz_2} \cdot \frac{dz_2}{db}$$

Suy ra:

$$\delta w_{21} = \delta z_2 \cdot x_1$$

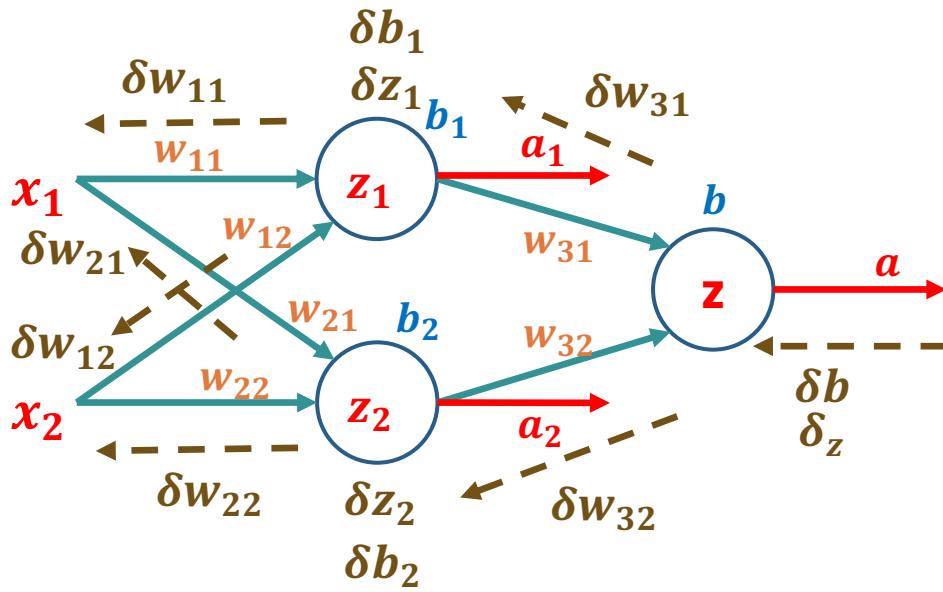
$$\delta w_{22} = \delta z_2 \cdot x_2$$

$$\delta b_2 = \delta z_2$$



6. Thuật toán lan truyền ngược

VD11: Backward Propagation



Gradient	Value
δz	$a - y$
δw_{31}	$\delta z \cdot a_1$
δw_{32}	$\delta z \cdot a_2$
δa_1	$\delta z \cdot w_{31}$
δz_1	$\delta a_1 \cdot a_1 (1 - a_1)$
δw_{11}	$\delta z_1 \cdot x_1$
δw_{12}	$\delta z_1 \cdot x_2$
δb_1	δz_1
δa_2	$\delta z \cdot w_{32}$
δz_2	$\delta a_2 \cdot a_2 (1 - a_2)$
δw_{21}	$\delta z_2 \cdot x_1$
δw_{22}	$\delta z_2 \cdot x_2$
δb_2	δz_2

6. Thuật toán lan truyền ngược

VD11: Backward Propagation

Tính Gradient cho tất cả điểm dữ liệu

$$\delta w = \frac{d\mathcal{L}}{dw} = \frac{1}{|D|} \sum_{x \in D} \delta w_x$$

$$\delta b = \frac{d\mathcal{L}}{db} = \frac{1}{|D|} \sum_{x \in D} \delta b_x$$

w, b: đại diện cho tất cả weight và bias



6. Thuật toán lan truyền ngược

VD11: Backward Propagation

Áp dụng Gradient Descent cho tất cả weight, bias

$$w := w - \eta * \delta w$$
$$b := b - \eta * \delta b$$

Suy ra:

$$w_{11} := w_{11} - \eta * \delta w_{11}$$

$$w_{12} := w_{12} - \eta * \delta w_{12}$$

$$w_{21} := w_{21} - \eta * \delta w_{21}$$

$$w_{22} := w_{22} - \eta * \delta w_{22}$$

$$w_{31} := w_{31} - \eta * \delta w_{31}$$

$$w_{32} := w_{32} - \eta * \delta w_{32}$$

$$b_1 := b_1 - \eta * \delta b_1$$

$$b_2 := b_2 - \eta * \delta b_2$$

$$b := b - \eta * \delta b$$



6. Thuật toán lan truyền ngược

VD11:

Với learning rate $\eta=0.1$, thực hiện propagation và back propagation với số lần lặp epoch = 100.000, sự thay đổi Loss như sau:

step	w11	w12	b1	w21	w22	b2	w31	w32	b	Loss
initial	0.16	-0.07	0.37	-0.66	0.63	-0.98	0.41	-0.90	0.35	0.713
#1	0.16	-0.07	0.37	-0.66	0.63	-0.98	0.4	-0.90	0.34	0.712
#2	0.16	-0.07	0.37	-0.66	0.63	-0.98	0.4	-0.90	0.33	0.711
#3	0.16	-0.07	0.37	-0.66	0.63	-0.98	0.39	-0.90	0.33	0.710
#4	0.16	-0.07	0.37	-0.65	0.63	-0.97	0.39	-0.91	0.32	0.709
#5	0.16	-0.07	0.37	-0.65	0.63	-0.97	0.39	-0.91	0.31	0.708
...
#1000000	7.69	7.69	-3.55	6.34	6.34	-9.68	14.74	-15.39	-7.02	0.001



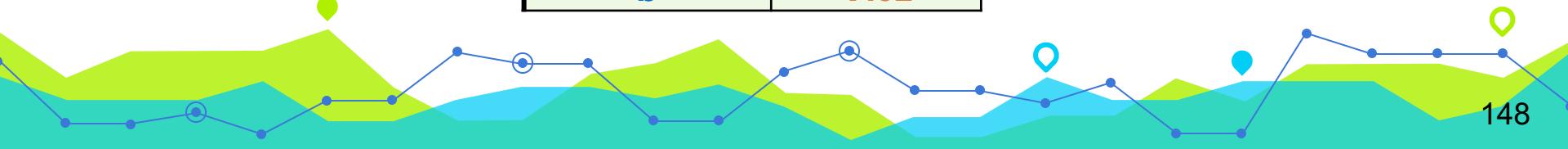
147

6. Thuật toán lan truyền ngược

VD11:

Sau 100.000 bước, giá trị của các tham số sau quá trình học

Parameter	Value
w11	7.69
w12	7.69
b1	-3.55
w21	6.34
w22	6.34
b	-9.68
w31	14.74
w31	-15.39
b	-7.02



6. Thuật toán lan truyền ngược

VD11:

Áp dụng mô hình đã huấn luyện được cho các input

x1	x2	w11	w12	b1	w21	w22	b2	z1	a1	z2	a2	w31	w32	b	z	a	y	BCE_x
0	0	7.69	7.69	-3.55	6.34	6.34	-9.68	-3.55	0.03	-9.68	0.00	14.74	-15.39	-7.02	-6.61	0.001346	0	0.0006
0	1	7.69	7.69	-3.55	6.34	6.34	-9.68	4.14	0.98	-3.34	0.03	14.74	-15.39	-7.02	6.96	0.999054	1	0.0004
1	1	7.69	7.69	-3.55	6.34	6.34	-9.68	4.14	0.98	-3.34	0.03	14.74	-15.39	-7.02	6.96	0.999054	1	0.0004
1	0	7.69	7.69	-3.55	6.34	6.34	-9.68	11.83	1.00	3.00	0.95	14.74	-15.39	-7.02	-6.94	0.000967	0	0.0004
$\mathcal{L}_{BCE}(w, b)$																		0.0005



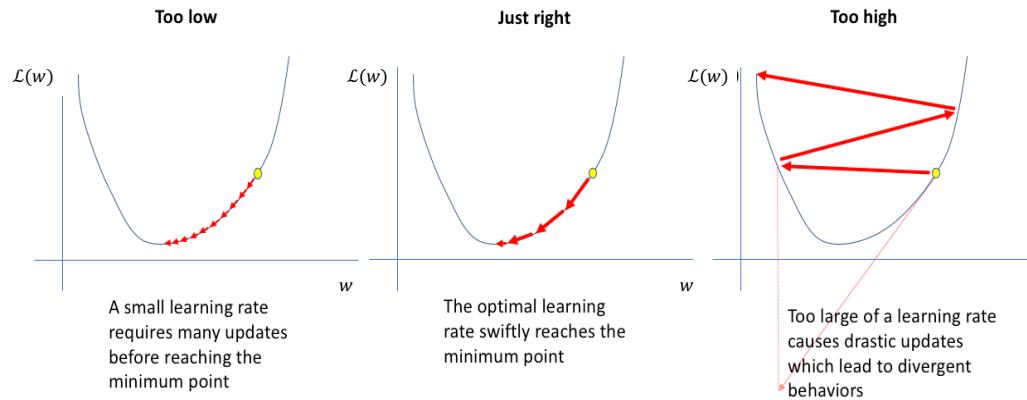
7. Hyper-parameter (Siêu tham số)

- Là những tham số hằng số, những cấu hình không thay đổi trong quá trình huấn luyện.
- Chỉ đánh giá lại các siêu tham số sau quá trình huấn luyện
- Một số siêu tham số
 - Learning rate
 - Số lượng hidden layers
 - Batch size: số lượng mẫu cho 1 lần huấn luyện
 - Activation function
 - Các giá trị, lựa chọn làm thay đổi cách thức học của mạng



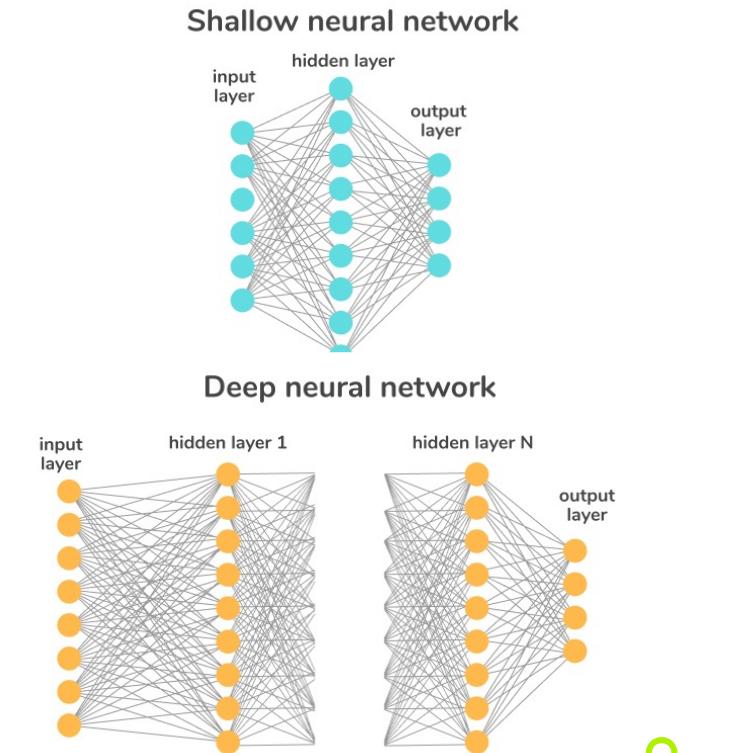
7.1 Learning rate

- Ảnh hưởng quan trọng đến hiệu quả và sự hội tụ của thuật toán
- η lớn: đẩy nhanh sự hội tụ của quá trình học, nhưng có thể bỏ qua điểm tối ưu hoặc tập trung vào điểm xấu (điểm yên ngựa).
- η nhỏ: có thể khiến quá trình học mất nhiều thời gian
- Thường chọn η theo kinh nghiệm.



7.2. Số lượng hidden layer

- Số lượng hidden layer, số lượng neuron trong từng hidden layer quan trọng cho việc ứng dụng ANN nhiều lớp để giải quyết các bài toán thực tế.
- Khó để xác định chính xác số lượng hidden layer, số lượng neuron của từng hidden layer để đạt độ chính xác mong muốn
- Thường được xác định thông qua các thực nghiệm



7.3. Activation Function

- Activation function: các hàm phi tuyến tính
- Lựa chọn Activation function thông qua thực nghiệm

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	

Activation function	Equation	Example	1D Graph
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	Multi-layer Neural Networks	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = \max(0, z)$	Multi-layer Neural Networks	
Rectifier, softplus	$\phi(z) = \ln(1 + e^z)$	Multi-layer Neural Networks	

Copyright © Sebastian Raschka 2016
(<http://sebastianraschka.com>)



7.4. Khởi tạo các tham số

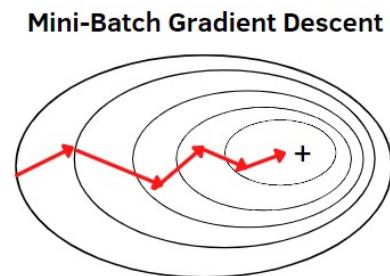
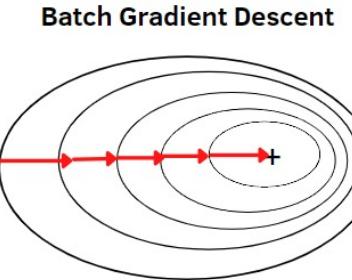
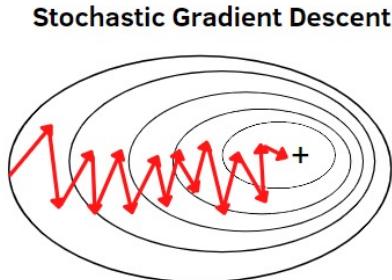
- Khởi tạo trọng số weight, bias
- Thông thường, các trọng số được khởi tạo với các giá trị nhỏ ngẫu nhiên.
 - Weight: thường khởi tạo ngẫu nhiên khác 0
 - Bias: thường khởi tạo bằng 0
- Nếu trọng số có giá trị ban đầu lớn: Activation function sẽ sớm đạt đến mức bão hòa (không thay đổi nhiều) hoặc có thể sẽ bế tắc, đứng yên tại một điểm nào đó (điểm yên ngựa)



7.5. Learning modes

03 modes of learning:

- Stochastic Gradient Descent (SGD)
- Batch Gradient Descent
- Mini-batch Gradient Descent

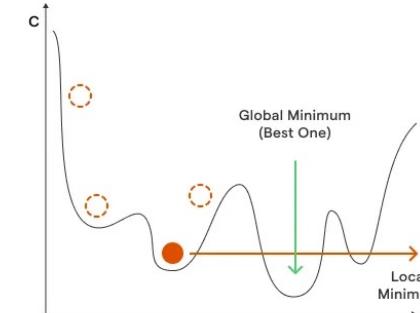
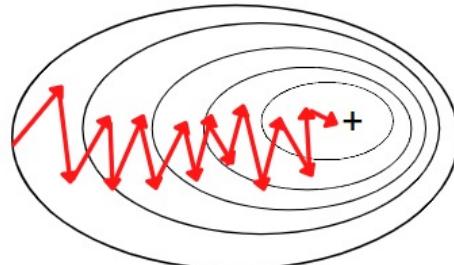


7.5. Learning modes

- Stochastic Gradient Descent (SGD):

- Mỗi lần cập nhật trọng số, chỉ một điểm dữ liệu (một cặp dữ liệu đầu vào và nhãn) ngẫu nhiên được sử dụng để tính gradient và cập nhật trọng số.
- Tính toán gradient nhanh hơn, không ổn định trong hướng cập nhật. Giúp vượt qua các điểm local minimum

Stochastic Gradient Descent

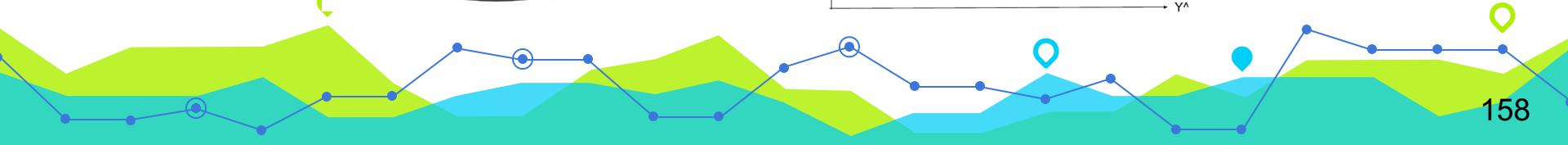
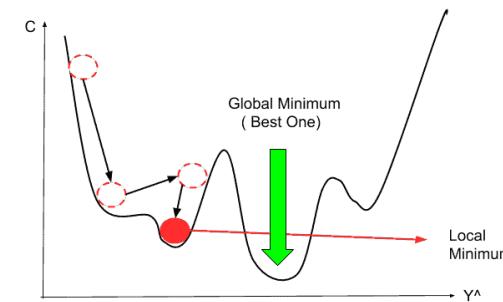
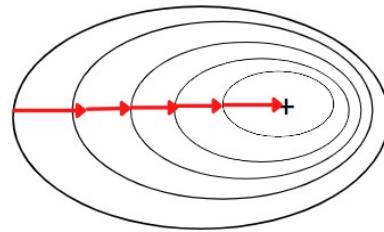


7.5. Learning modes

- Batch Gradient Descent:

- Toàn bộ tập dữ liệu huấn luyện được sử dụng để tính gradient và cập nhật trọng số trong mỗi lần huấn luyện.
- Tính toán gradient chính xác hơn, nhưng cũng làm tăng đáng kể chi phí tính toán nếu tập dữ liệu lớn. Có thể bị dừng lại tại 1 điểm local minimum

Batch Gradient Descent



7.5. Learning modes

- **Mini-batch Gradient Descent:**

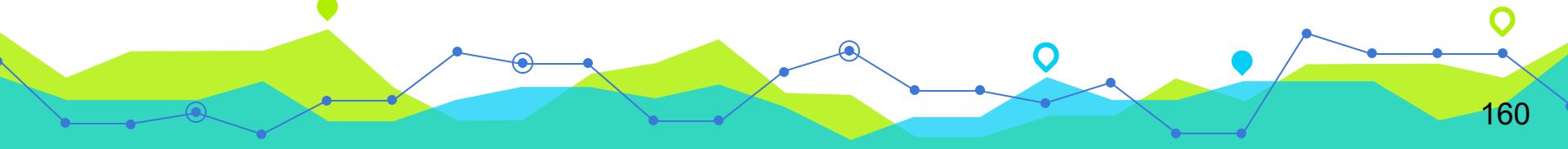
- Tập dữ liệu huấn luyện được chia thành các mini-batches nhỏ.
- Trong mỗi epoch, các mini-batches này được lần lượt sử dụng để tính gradient và cập nhật trọng số.
- Giảm bớt sự không ổn định trong quá trình cập nhật trọng số so với SGD, giảm chi phí tính toán so với batch gradient descent. Hạn chế bị dừng lại tại local minimum



8. Ưu điểm và nhược điểm của ANN

- **Ưu điểm**

- Hỗ trợ tính toán song song rất cao
- Đạt độ chính xác cao trong nhiều bài toán (ảnh, video, âm thanh, văn bản, thư viết tay, ...)
- Kiến trúc mạng rất linh hoạt
- Khả năng thích ứng tốt với dữ liệu nhiễu



8. Ưu điểm và nhược điểm của ANN

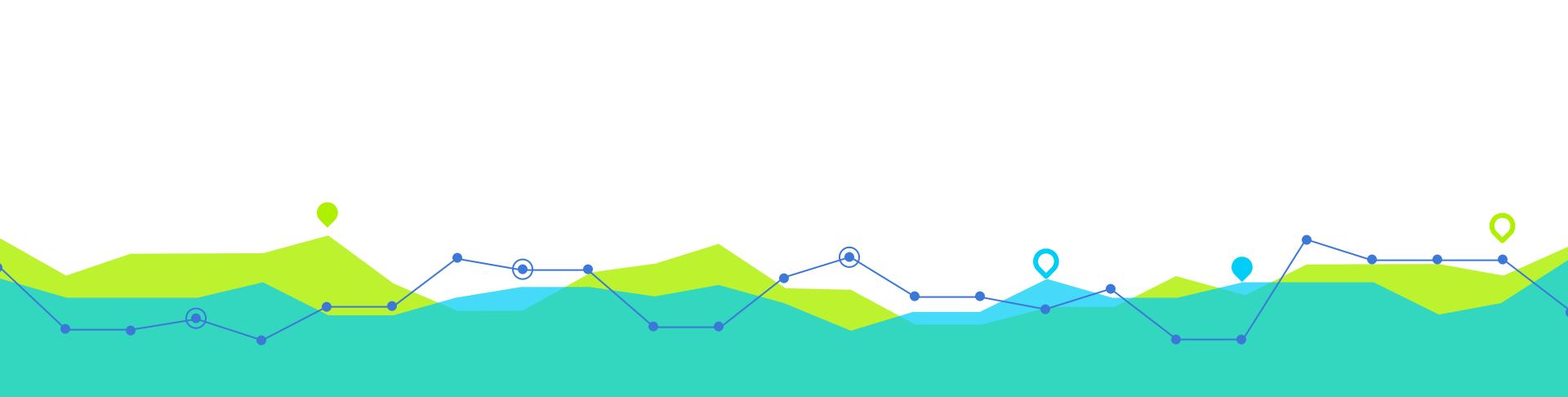
- Nhược điểm

- Thời gian huấn luyện dài
- Không có quy tắc chung để xác định cấu trúc mạng và các tham số tối ưu cho một vấn đề nhất định. Thường dựa trên kinh nghiệm
- Khả năng diễn giải kém: Khó diễn giải ý nghĩa tượng trưng đằng sau các trọng số đã học và các “node ẩn” trong mạng



7

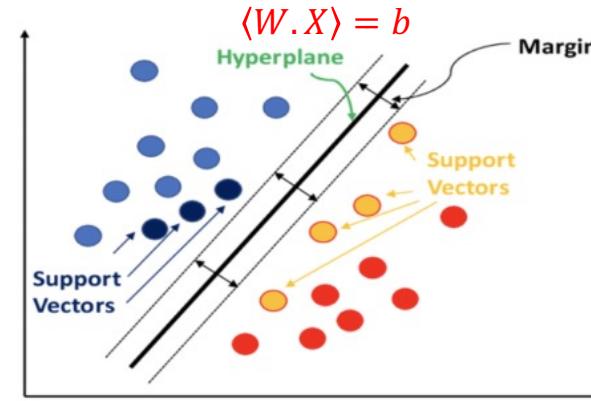
Một số phương pháp khác

- 
1. Support Vector Machines
 2. Ensemble Methods

1. Support Vector Machines (SVM)

- Thuật toán Supervised learning phổ biến trong bài toán Classification và Regression
- Ý tưởng: tìm một siêu phẳng (hyper plane) $\langle W \cdot X \rangle = b$ trong không gian N chiều (ứng với N đặc trưng) để phân tách các điểm dữ liệu, phân chia không gian thành các miền khác nhau và mỗi miền sẽ chứa một lớp dữ liệu.

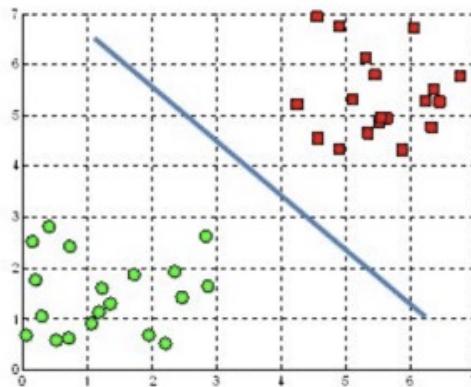
WHAT IS A
**SUPPORT
VECTOR
MACHINE?**



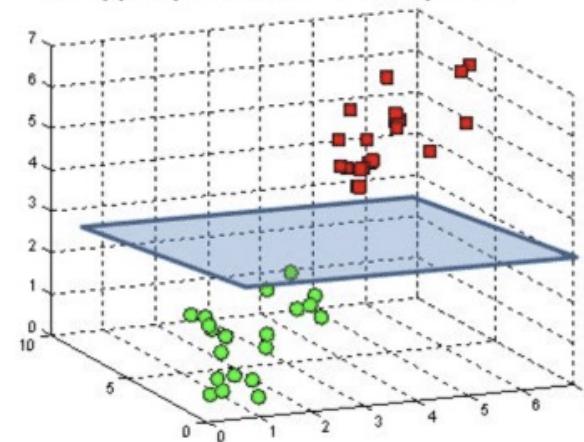
1. Support Vector Machines (SVM)

- Không gian N chiều, siêu phẳng là một không gian con N-1 chiều

A hyperplane in \mathbb{R}^2 is a line

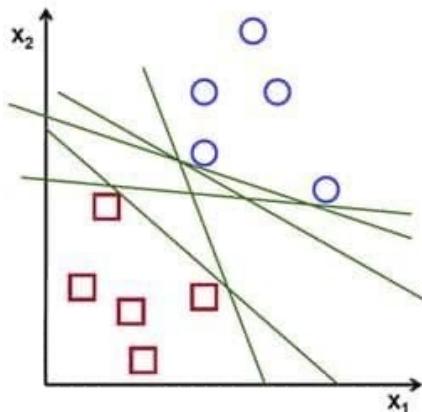


A hyperplane in \mathbb{R}^3 is a plane

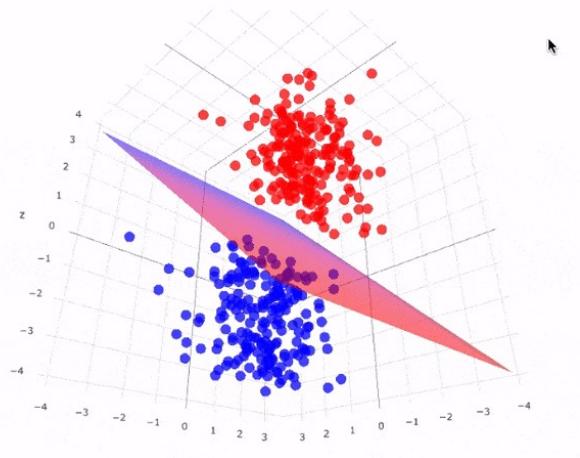
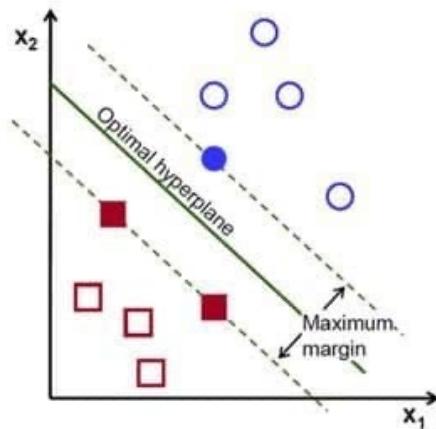


1. Support Vector Machines (SVM)

- Có rất nhiều siêu phẳng có thể phân chia các lớp dữ liệu => tìm ra **siêu phẳng có lề (margin) rộng nhất**, tức là có khoảng cách tới các điểm của hai lớp là lớn nhất.



Possible hyperplanes



1. Support Vector Machines (SVM)

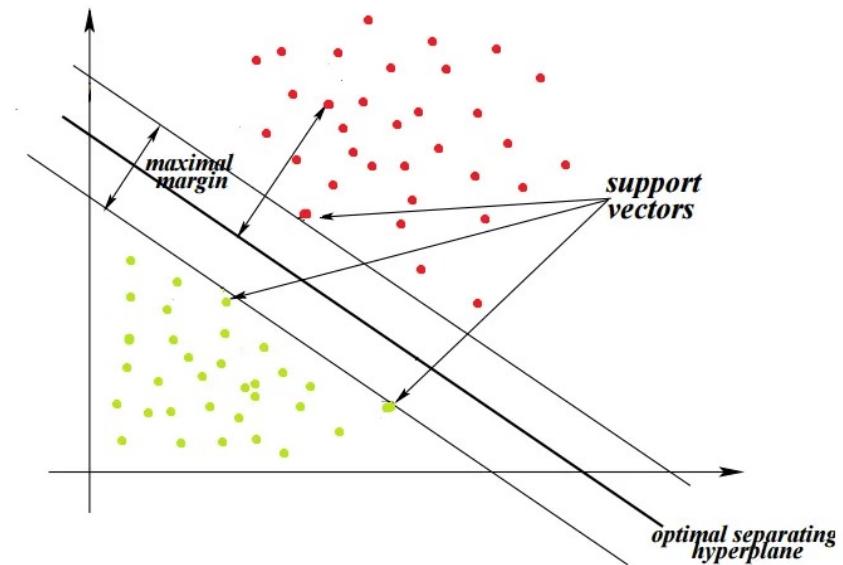
- **Vector:** một điểm trong không gian

- **Vector hỗ trợ:**

- Điểm dữ liệu nằm trên hoặc gần nhất với siêu phẳng,

- Ảnh hưởng đến vị trí và hướng của siêu phẳng, được sử dụng để tối ưu hóa margin. Nếu xóa các điểm này, vị trí của siêu phẳng sẽ thay đổi.

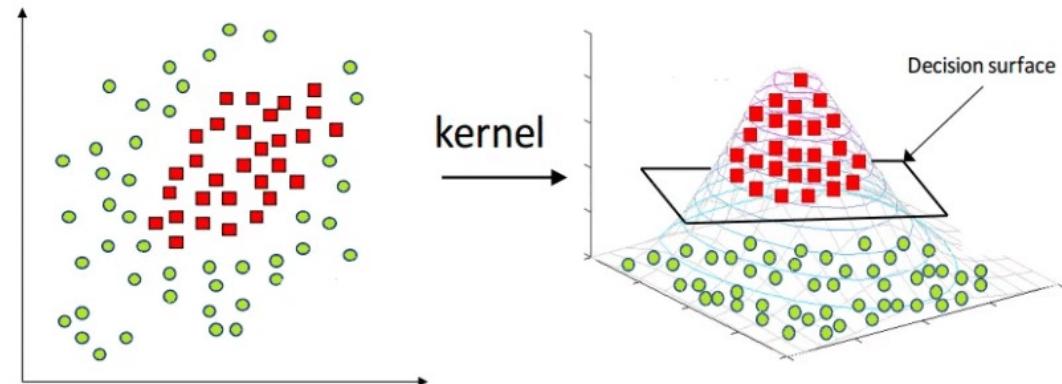
- Các vector hỗ trợ phải cách đều siêu phẳng.



1. Support Vector Machines (SVM)

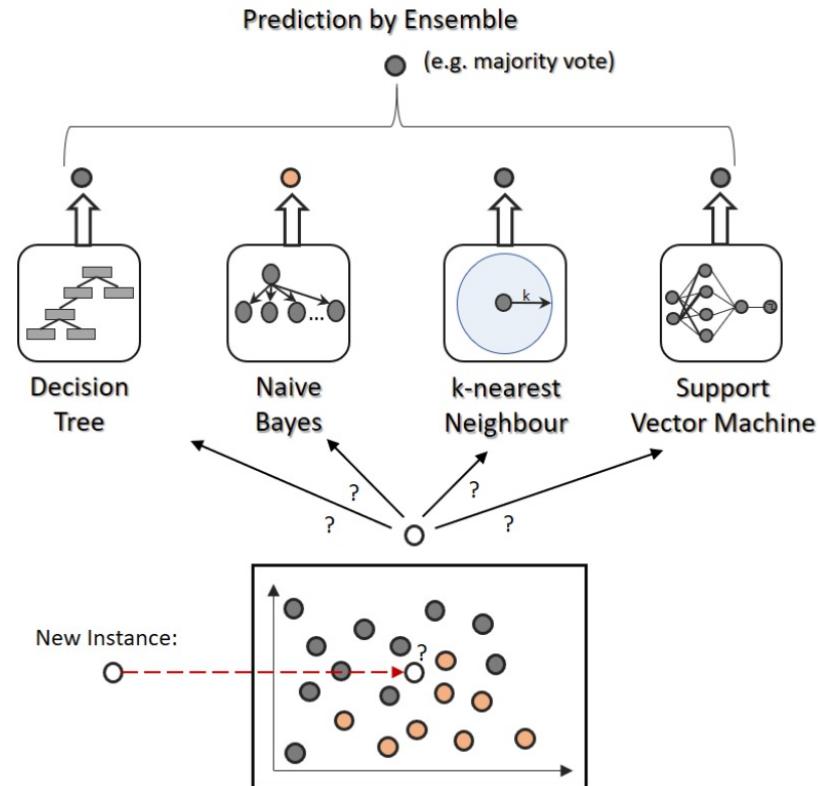
- Với dữ liệu phân tách phi tuyến tính:

- Sử dụng kernel là một hàm ánh xạ dữ liệu từ không gian ít chiều hơn sang không gian nhiều chiều hơn, từ đó tìm được siêu phẳng phân tách dữ liệu.
- Một số kernel:
 - Linear (tuyến tính)
 - Polynomial (đa thức)
 - Radial Basic Function (RBF) hay Gaussian kernel
 - Sigmoid



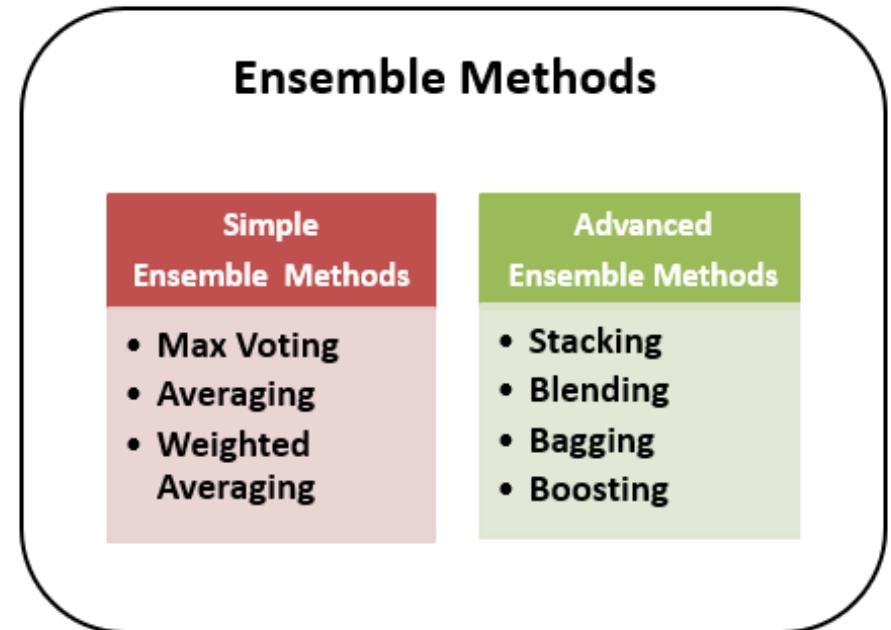
2. Ensemble Methods

- Là phương pháp kết hợp các dự đoán từ nhiều thuật toán máy học để có được dự đoán chính xác hơn từng mô hình riêng biệt.
- Kết quả của các mô hình học máy có thể được cải thiện bằng cách kết hợp kết quả của các mô hình khác nhau.



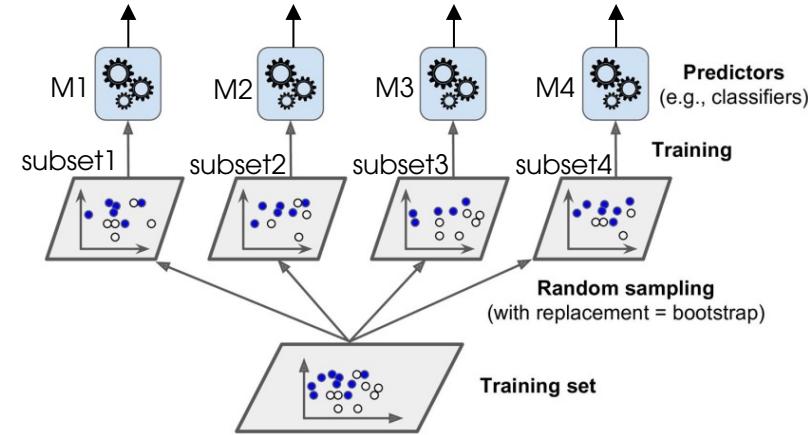
2. Ensemble Methods

- Phân loại:
 - Simple ensemble methods
 - Max voting
 - Averaging
 - Weighted Average
 - Advanced ensemble methods
 - Bagging
 - Boosting
 - Stacking
 - Blending



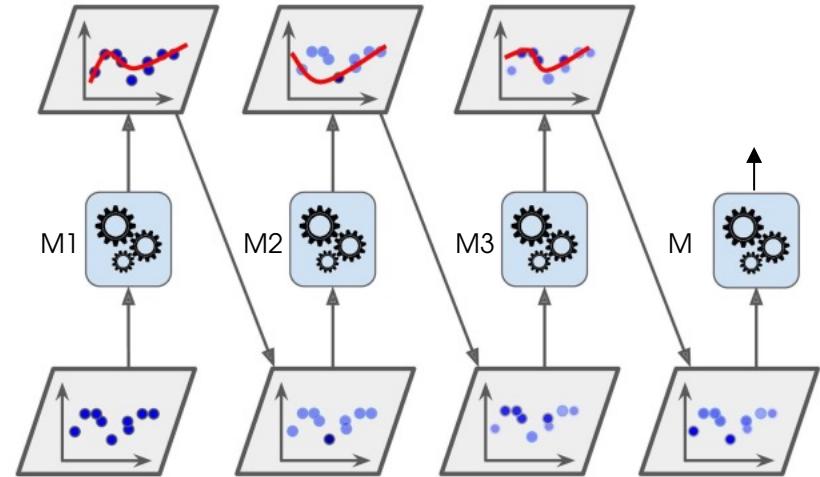
Ensemble: Bagging

- Chia tập train thành các subset, huấn luyện các bộ weak learner trên các subset (song song, độc lập).
- Chung 1 thuật toán cho các model
- Output:
 - Hard Voting: Lớp được nhiều bộ learner dự đoán nhất
 - Soft Voting: Nếu output của các learner là số thực (xác suất của class trong classification hoặc giá trị trong regression) thì có thể chọn class có xác suất lớn nhất hoặc lấy trung bình các regression
- VD: Random forest, Bagging meta-estimator



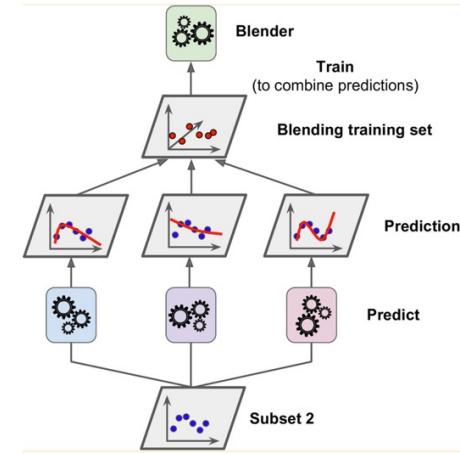
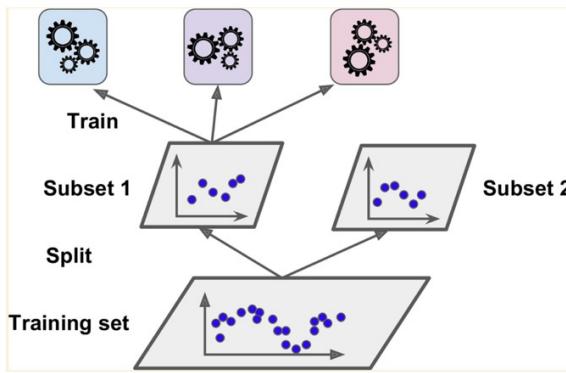
Ensemble: Boosting

- Cố gắng xây dựng một strong classifier từ những weak classifier
- Đầu tiên, một mô hình được xây dựng từ dữ liệu huấn luyện. Sau đó, mô hình thứ hai được xây dựng để cố gắng sửa các lỗi có trong mô hình đầu tiên.
- Quy trình này được tiếp tục và các mô hình được thêm vào cho đến khi tập dữ liệu huấn luyện hoàn chỉnh được dự đoán chính xác hoặc số lượng mô hình tối đa được thêm vào.
- VD: AdaBoost, Gradient Boosting (GBM), XGBoost (XGBM), Light GBM, CatBoost



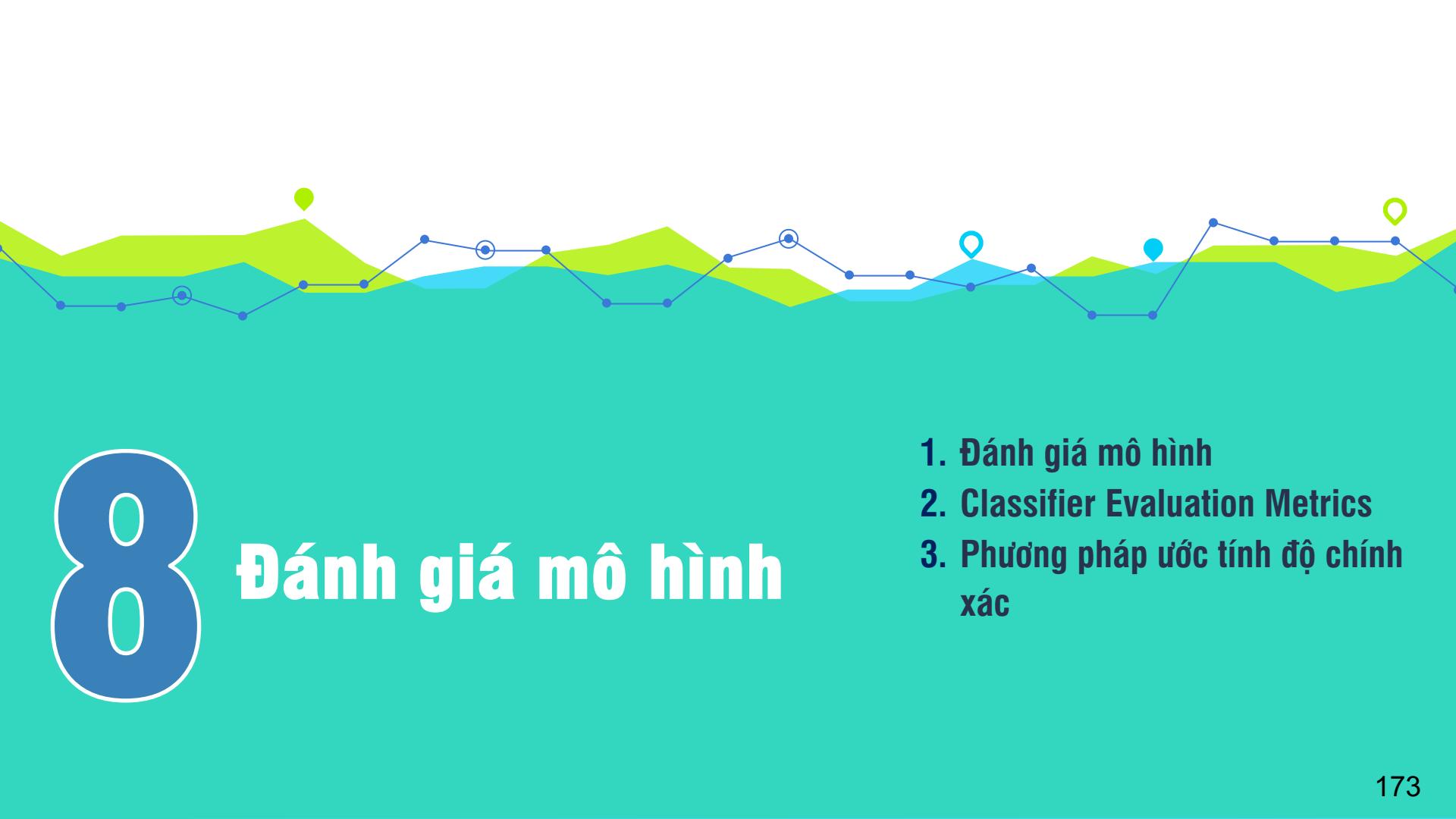
Ensemble: Stacking

- Chia dữ liệu: subset1 và subset2
- Huấn luận các base model từ subset1. Sau đó, các base model dự đoán cho subset2.
- Xây dựng Blender huấn luyện từ những dự đoán của các base model, giá trị thực tế của subset2



8

Đánh giá mô hình

- 
1. Đánh giá mô hình
 2. Classifier Evaluation Metrics
 3. Phương pháp ước tính độ chính xác

1. Đánh giá mô hình

- Sử dụng tập validation, test của các bộ dữ liệu được gắn nhãn lớp để đánh giá thay vì sử dụng tập train
- Sử dụng các độ đo đánh giá (evaluation metrics)
- Sử dụng các phương pháp ước tính độ chính xác:
 - Cross-validation
 - Bootstrap
 - Comparing classifiers
 - Confidence intervals
 - Cost-benefit analysis and ROC Curves



2. Classifier Evaluation Metrics

- Tập trung vào **khả năng dự đoán** của mô hình hơn là tốc độ phân lớp hay xây dựng mô hình, khả năng mở rộng,
- Một số Evaluation metrics:
 - Confusion Matrix
 - Accuracy, Error Rate, Sensitivity, Specificity
 - Precision, Recall, F-measures

2.1. Confusion matrix

↔ Lớp dự đoán →

Lớp thực tế \ Lớp dự đoán		
	C_1	$\neg C_1$
C_1	True Positives (TP)	False Negatives (FN)
	False Positives (FP)	True Negatives (TN)

Cho m classes, $CM_{i,j}$ số lượng bộ ở lớp i được dự đoán ở lớp j

2.2. Accuracy, Error Rate, Sensitivity and Specificity

A \ P	C	$\neg C$	sum
C	TP	FN	P
$\neg C$	FP	TN	N
sum	P'	N'	All

Accuracy: Tỷ lệ các bộ được phân lớp đúng

$$Accuracy = \frac{TP + TN}{All} \quad (19)$$

Error rate:

$$Error rate = 1 - Accuracy = \frac{FP + FN}{All} \quad (20)$$

Vấn đề mất cân bằng giữa các lớp: (một lớp có thể rất hiếm)

Sensitivity: Tỷ lệ phân lớp positive đúng

$$Sensitivity = \frac{TP}{P} \quad (21)$$

Specificity: Tỷ lệ phân lớp negative đúng

$$Specificity = \frac{TN}{N} \quad (22)$$



2.3. Precision and Recall, and F-measures

Binary Classification

A \ P	C	$\neg C$	
C	TP	FN	P
$\neg C$	FP	TN	N
	P'	N'	All

F-measure (F₁ hoặc F-score): trung bình hài hòa giữa Precision và Recall

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (25)$$

F1 càng cao classifier càng tốt

Precision: Tỷ lệ các bộ được gán positive thực sự là positive

$$Precision = \frac{TP}{TP + FP} \quad (23)$$

Recall: Tỷ lệ các bộ được gán positive trên các bộ positive thực sự

$$Recall = \frac{TP}{TP + FN} \quad (24)$$

F_β : weighted measure giữa Precision và Recall

$$F_\beta = \frac{(1 + \beta^2) * Precision * Recall}{\beta^2 * Precision + Recall} \quad (26)$$



2. Classifier Evaluation Metrics

VD12: Cho Confusion matrix. Tính Accuracy, Error rate, Sensitivity, Specificity, Precision, Recall và F1-score

Actual class \ Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	6954	46	7000
buy_computer = no	412	2588	3000
Total	7366	2634	10000



2. Classifier Evaluation Metrics

VD12:

Actual \ Predicted	no	no	Total
yes	6954	46	7000
no	412	2588	3000
Total	7366	2634	10000

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} = \frac{6954 + 2588}{10000} = 95.42\%$$

$$Errorrate = 1 - Accuracy = 4.58\%$$

$$Sensitivity = \frac{TP}{TP + FN} = \frac{6954}{7000} = 99.34\%$$

$$Specificity = \frac{TN}{FP + TN} = \frac{2588}{3000} = 86.27\%$$

$$Precision = \frac{TP}{TP + FP} = \frac{6954}{6954 + 412} = 94.41\%$$

$$Recall = \frac{TP}{TP + FN} = \frac{6594}{6594 + 46} = 99.34\%$$

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} = 96.81\%$$



180

2.3. Precision and Recall, and F-measures

Multi-class Classification

A \ P	C ₁	C ₂	...	C _n
C ₁				
C ₂				
...				
C _n				

Với từng class C_i: $Precision_{C_i}$, $Recall_{C_i}$, $F1 - score_{C_i}$

Trung bình cộng trên tất cả các class

$$Macro\ Precision = \frac{1}{n} \sum_{i=1}^n Precision_{C_i} \quad (27)$$

$$Macro\ Recall = \frac{1}{n} \sum_{i=1}^n Recall_{C_i} \quad (28)$$

$$Macro\ F1 - score = \frac{1}{n} \sum_{i=1}^n F1 - score_{C_i} \quad (29)$$



2.3. Precision and Recall, and F-measures

Multi-class Classification

A \ P	C ₁	C ₂	...	C _n
C ₁				
C ₂				
...				
C _n				

Với từng class C_i: $Precision_{C_i}$, $Recall_{C_i}$, $F1 - score_{C_i}$

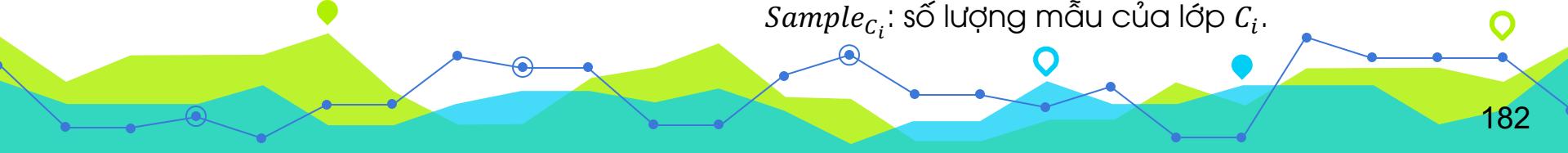
Trung bình cộng có trọng số trên tất cả các class

$$\text{Weighted Precision} = \frac{\sum_{i=1}^n (Precision_{C_i} * Sample_{C_i})}{\sum_{i=1}^n Sample_{C_i}} \quad (30)$$

$$\text{Weighted Recall} = \frac{\sum_{i=1}^n (Recall_{C_i} * Sample_{C_i})}{\sum_{i=1}^n Sample_{C_i}} \quad (31)$$

$$\text{Weighted F1-score} = \frac{\sum_{i=1}^n (F1 - score_{C_i} * Sample_{C_i})}{\sum_{i=1}^n Sample_{C_i}} \quad (32)$$

$Sample_{C_i}$: số lượng mẫu của lớp C_i.



2. Phương pháp ước tính độ chính xác

- Phương pháp Holdout:

- Phân chia ngẫu nhiên dữ liệu:
 - Tập train (2/3) để xây dựng mô hình
 - Tập test (1/3) để ước tính độ chính xác
- Thích hợp cho tập dữ liệu nhỏ
- Lấy mẫu sao cho mỗi lớp được phân bổ đều trong train và test
- Lấy mẫu ngẫu nhiên: Lặp lại holdout k lần, độ chính xác = trung bình của độ chính xác thu được.



2. Phương pháp ước tính độ chính xác

- **Phương pháp Cross-validation (k-fold):**

- Phân chia ngẫu nhiên dữ liệu thành k tập con loại trừ lẫn nhau, mỗi tập có kích thước xấp xỉ bằng nhau
- Tại mỗi vòng lặp, sử dụng một tập con làm tập test và các tập còn lại làm tập train
- Thường chọn $k = 10$
- Leave-one-out: k lần trong đó $k = \text{số mẫu}$ (đối với dữ liệu nhỏ)
- Stratified cross-validation: dùng phương pháp lấy mẫu để phân bố các lớp trong từng tập con giống như trên toàn bộ dữ liệu



2. Phương pháp ước tính độ chính xác

- **Phương pháp Bootstrap:**
 - Hoạt động tốt với các bộ dữ liệu nhỏ
 - Khi một bộ dữ liệu được chọn, nó có khả năng được chọn lại và thêm lại vào tập huấn luyện.
- Ngoài ra, còn nhiều phương pháp khác như:
 - Comparing classifiers
 - Confidence intervals
 - Cost-benefit analysis and ROC Curves



Các vấn đề ảnh hưởng đến việc lựa chọn mô hình

- **Accuracy:** Độ chính xác của bộ phân lớp, dự đoán nhãn lớp
- **Tốc độ (Speed):**
 - Thời gian xây dựng mô hình (thời gian huấn luyện)
 - Thời gian sử dụng mô hình (thời gian phân lớp/dự đoán)
- **Mạnh mẽ (Robustness):** xử lý dữ liệu noise và các giá trị bị thiếu
- **Khả năng mở rộng (Scalability):** hiệu quả với CSDL lớn
- **khả năng diễn giải (Interpretability):** sự hiểu biết và diễn giải mô hình
- Các vấn đề khác: mức độ tốt của các luật, chẳng hạn như kích thước cây quyết định hoặc độ chặt chẽ của các luật phân lớp

Đánh giá mô hình: Bài tập

BT14

Cho ma trận nhầm lẩn sau

Tính Accuracy, Weighted Precision, Weighted Recall và Weighted F1-score

Actual Class \ Predicted class	Iris-setosa	Iris-versicolor	iris-virginica
Iris-setosa	25	5	0
Iris-versicolor	3	20	5
iris-virginica	2	7	15



187

Tổng kết chương



Tổng quan về Phân lớp dữ liệu

1. Phân lớp dữ liệu
2. Quy trình phân lớp
3. Các kỹ thuật phân lớp



Phương pháp dựa trên Cây quyết định

1. Định nghĩa
2. Xây dựng cây quyết định
3. Một số thuật toán xây dựng cây quyết định
4. Cách phân chia mẫu
5. Biến đổi cây thành luật
6. Vấn đề quá phù hợp dữ liệu
7. Ưu điểm



Phương pháp dựa trên luật

1. Giới thiệu
2. Xây dựng luật phân lớp
3. Xác định lớp cho các mẫu



Tổng kết chương



Phương pháp Naïve Bayes

- 1.Giới thiệu
- 2.Thuật toán Naïve Bayes
- 3.Ưu điểm và nhược điểm



Phương pháp dựa trên thể hiện

- 1.Giới thiệu
- 2.Thuật toán K-NN



Mạng neural nhân tạo

- 1.Giới thiệu
- 2.Cấu trúc một neural nhân tạo
- 3.Kiến trúc ANN
- 4.Learning trong ANN
- 5.Thuật toán lan truyền ngược
- 6.Ưu và nhược điểm của ANN



Tổng kết chương



Một số phương pháp khác

1. Support Vector Machines
2. Ensemble Methods



Đánh giá mô hình

1. Đánh giá mô hình
2. Classifier Evaluation Metrics
3. Phương pháp ước tính độ chính xác



Bài tập chương 6

6.1. Cho dữ liệu huấn luyện:

1. Xây dựng cây quyết định
(sử dụng Gain)
2. Xây dựng cây quyết định
(sử dụng Gini index)

	Headache	Vomitting	Temperature	Viral illness
1	No	Yes	High	Yes
2	Yes	No	High	Yes
3	Yes	Yes	Very high	Yes
4	No	Yes	Normal	No
5	Yes	No	Very high	Yes
6	No	Yes	Very high	Yes
7	Yes	Yes	High	Yes
8	No	No	Very high	No
9	Yes	Yes	Normal	No



Bài tập chương 6

6.2. Cho dữ liệu huấn luyện:

Áp dụng Naïve Bayes để tính các xác suất $P(C_i)$ và $P(x_k | C_i)$ với $C_1 = \text{yes}$, $C_2 = \text{no}$.

Chuẩn hóa các xác suất bằng phương pháp làm tròn Laplace

	Size	Color	Shape	Decision
1	Vừa	Xanh dương	Hộp	Yes
2	Nhỏ	Đỏ	Nón	No
3	Nhỏ	Đỏ	Cầu	Yes
4	Lớn	Đỏ	Nón	No
5	Lớn	Xanh lá cây	Trụ	Yes
6	Lớn	Đỏ	Trụ	No
7	Lớn	Xanh lá cây	Cầu	Yes

Bài tập chương 6

6.3. Cho tập huấn luyện

1. Sử dụng K-NN để xác định lớp cho Tuyến với $k = 3, 5, 7$.
2. Chuẩn hóa dữ liệu và sử dụng K-NN để xác định lớp cho Tuyến

Customer	Age	Income (K)	No. cards	Response
Lâm	35	35	3	Yes
Hưng	22	50	2	No
Mai	28	40	1	Yes
Lan	45	100	2	No
Thúy	20	30	3	Yes
Tuấn	34	55	2	No
Ninh	63	200	1	No
Vân	55	140	2	No
Thiện	59	170	1	No
Ngọc	25	40	4	Yes
Tuyến	25	30	1	???

THANKS!

Any questions?

