

Classification

Trịnh Thành

thanh.trinh@phenikaa-uni.edu.vn

Phenikaa

Nội dung

1. **Classification**
2. Cây quyết định
3. Randomforest
4. Sampling
5. Bayesian Decision Theory
6. Naïve Bayes Classification



biết bay



biết bơi



Biết bò

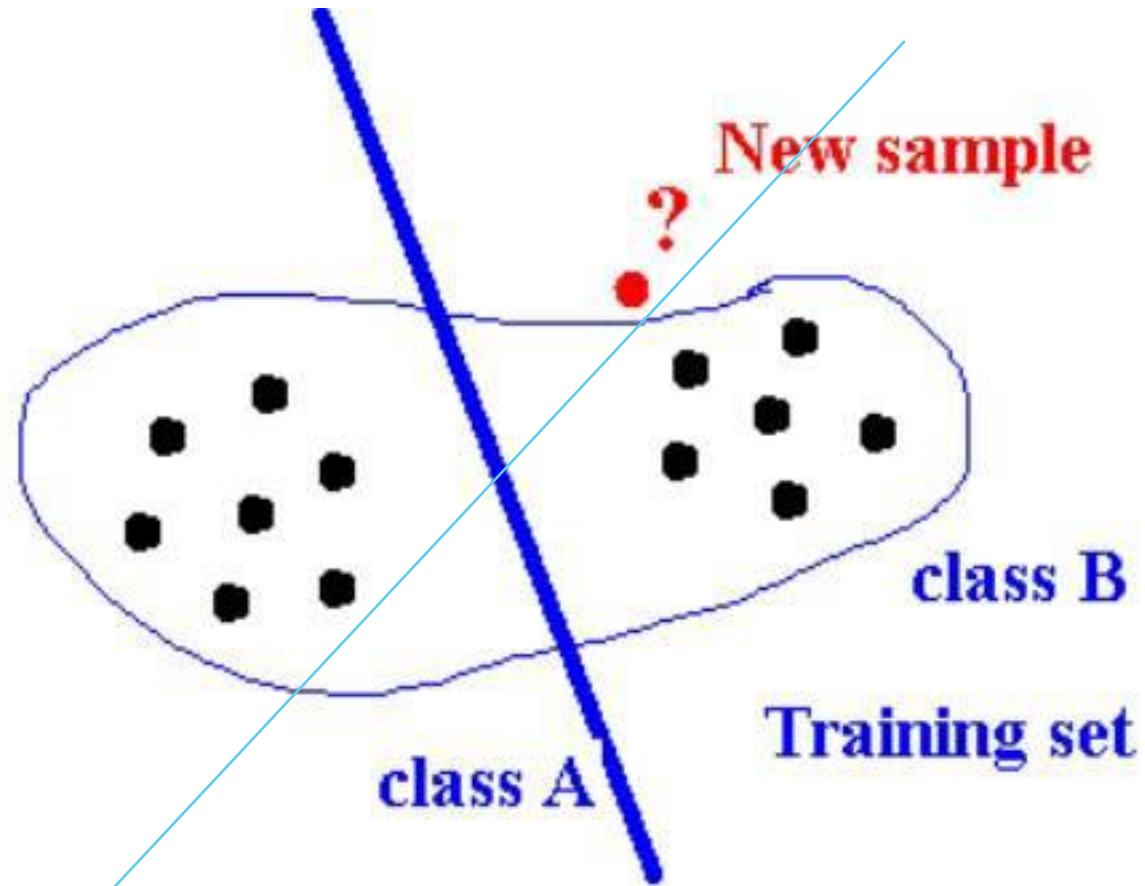


Biết....?

► Vấn đề?

Khóa	MãSV	MônHọc1	MônHọc2	...	TốtNghiep
2004	1	9.0	8.5	...	Có
2004	2	6.5	8.0	...	Có
2004	3	4.0	2.5	...	Không
2004	8	5.5	3.5	...	Không
2004	14	5.0	5.5	...	Có
...
2005	90	7.0	6.0	...	Có
2006	24	9.5	7.5	...	Có
2007	82	5.5	4.5	...	Không
2008	47	2.0	3.0	...	Không
...

Làm sao xác định liệu sinh viên A sẽ tốt nghiệp?



Cho trước tập huấn luyện (training set), dẫn ra mô tả về class A và class B?
Cho trước mẫu/đối tượng mới, làm sao xác định class cho mẫu/đối tượng đó?
Liệu class đó có thực sự phù hợp/đúng cho mẫu/đối tượng đó?



BỆNH VIỆN ĐẠI HỌC Y HÀ NỘI
HANOI MEDICAL UNIVERSITY HOSPITAL

PHÂN BIỆT CÁC TRIỆU CHỨNG NHIỄM COVID-19

BIỂU HIỆN	COVID-19	CẢM LẠNH	DỊ ỨNG THỜI TIẾT	CÚM MÙA
Ho	Thường gặp (ho khan)	Thường gặp	Thỉnh thoảng có	Thường gặp
Đau cơ	Thường gặp	Thỉnh thoảng có	Không có	Thường gặp
Mệt mỏi	Thường gặp	Thỉnh thoảng có	Thỉnh thoảng	Thường gặp
Hắt hơi	Hiếm khi có	Thỉnh thoảng có	Thường gặp	
Đau họng	Thường gặp	Thường gặp	Hiếm gặp	Thường gặp
Chảy mũi/ Nghẹt mũi	Thường gặp	Thường gặp	Thường gặp	Thường gặp
Sốt	Thường gặp	Thỉnh thoảng có	Không có	Thường gặp, không phải mọi lúc
Tiêu chảy	Thỉnh thoảng có	Không có	Không có	Thỉnh thoảng có, thường gặp hơn ở trẻ nhỏ
Nôn, buồn nôn	Thỉnh thoảng có	Không có	Không có	Thỉnh thoảng có, thường gặp hơn ở trẻ nhỏ
Mới xuất hiện mất vị giác, khứu giác	Thường có (xảy ra sớm, thường không kèm theo sổ mũi hay nghẹt mũi)	Thỉnh thoảng có (đặc biệt nếu có nghẹt mũi kèm theo)	Thỉnh thoảng có	Hiếm có
Mắt đỏ (viêm kết mạc)	Thỉnh thoảng có		Thỉnh thoảng có	
Khó thở	Thường gặp			Thường gặp
Ngứa mắt, mũi, miệng	Không có		Thường có	

Phân lớp là gì?

- ▶ Là một quá trình của việc chia các lớp dữ liệu thành các nhóm hay loại khác nhau bằng việc gắn nhãn.
- ▶ Là kỹ thuật của việc phân loại các quan sát (mẫu) thành các loại khác nhau. Về cơ bản, chúng ta xử lý dữ liệu, phân tích dữ liệu dựa trên một số điều kiện và cuối cùng chúng ta phân chia dữ liệu đấy thành các loại hay nhóm đã được gắn nhãn trước.

Name	Egg-laying	Scales	Poisonous	Cold-Blooded	#legs	Reptile
Rắn mang bành	True	True	True	True	0	YES
Rắn đuôi chuông	True	True	True	True	0	YES
Trăn nhiệt đới	False	True	False	True	0	YES
Gà	True	True	False	False	2	NO
Cá chép	False	True	False	False	0	No
Ếch độc	True	False	True	False	4	No
Ngựa vằn	False	False	False	False	4	No
Trăn	True	True	False	True	0	Yes
Cá sấu	True	True	False	True	4	Yes

Mô hình phân loại: scales, cold-blooded

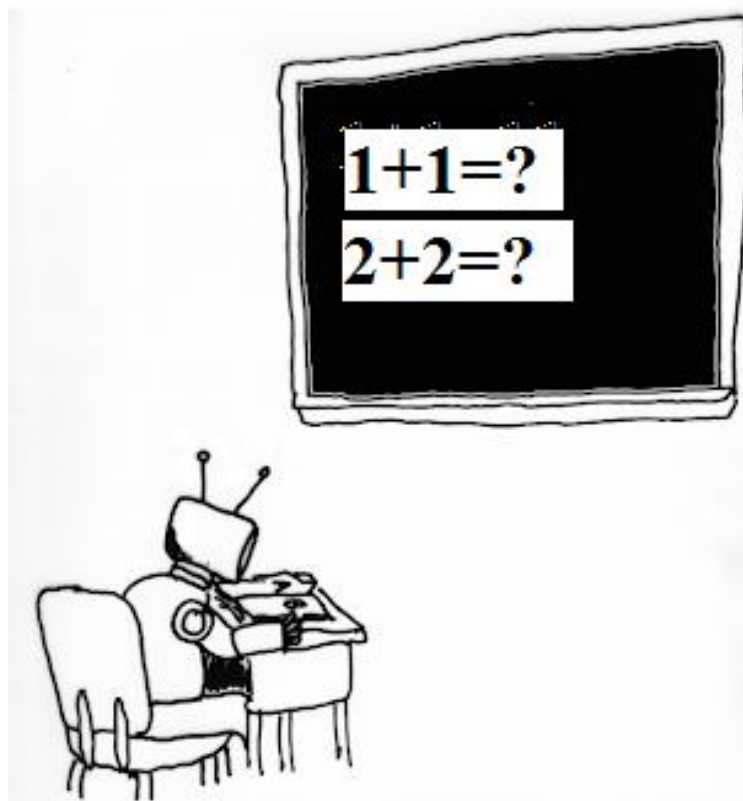
Name	Egg-laying	Scales	Poisonous	Cold-Blooded	#legs	Reptile
Rắn mang bành	True	True	True	True	0	YES
Rắn đuôi chuông	True	True	True	True	0	YES
Gà	True	True	False	False	2	NO
Cá chép	False	True	False	False	0	No
Ếch độc	True	False	True	False	4	No
Ngựa vằn	False	False	False	False	4	No
Trăn	True	True	False	True	0	Yes
Cá sấu	True	True	False	True	4	Yes
Con rùa	True	True	False	True	4	?
Cá hồi	True	True	False	True	0	?

Phân lớp -Classification

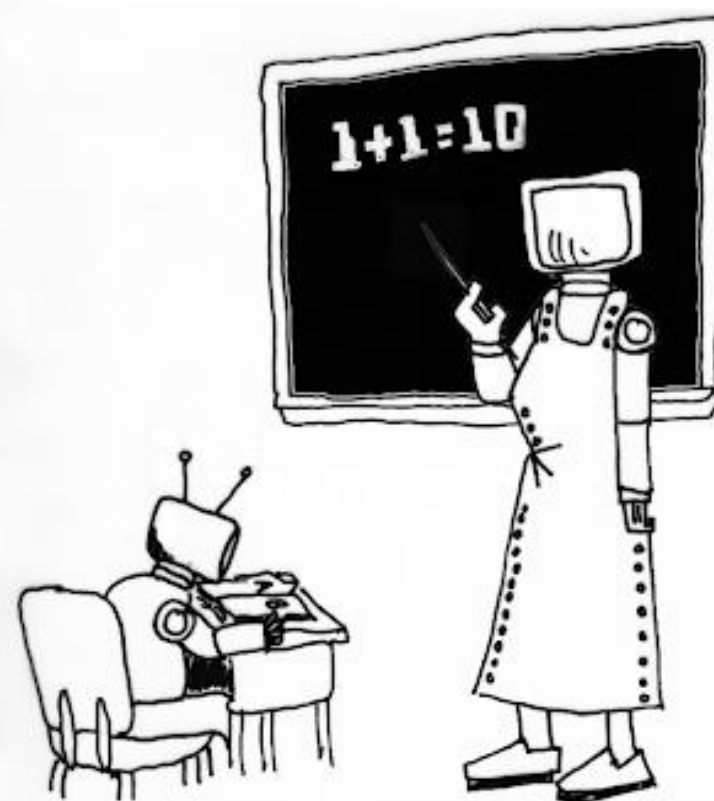
- ▶ Phân lớp là một trong các tác vụ phổ biến của KHDL (data mining)
- ▶ Phân lớp sử dụng một phương pháp (mô hình) để gán một đối tượng một giá trị cụ thể trong nhóm các giá trị đã được định nghĩa trước (gọi là các labels - nhãn).
- ▶ Một mô hình phân lớp được xây dựng từ những dữ liệu đã biết. (Gọi là training data).

Thuật toán (mô hình) phân lớp

UNSUPERVISED MACHINE LEARNING

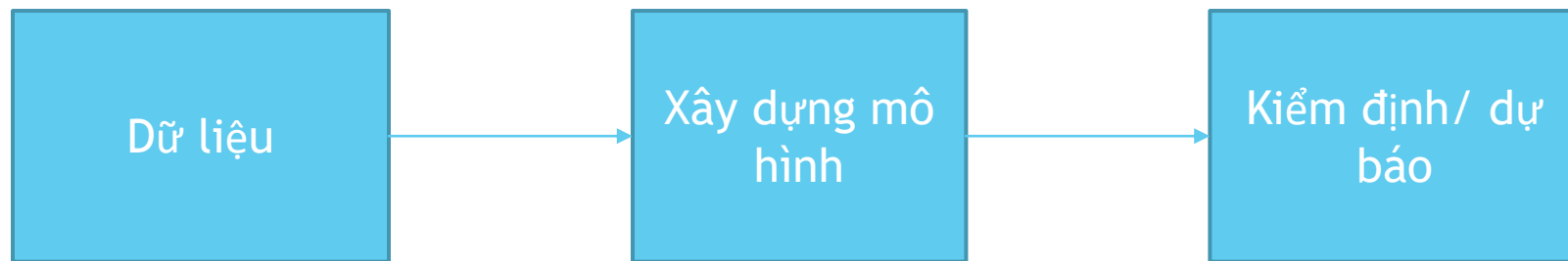


SUPERVISED MACHINE LEARNING



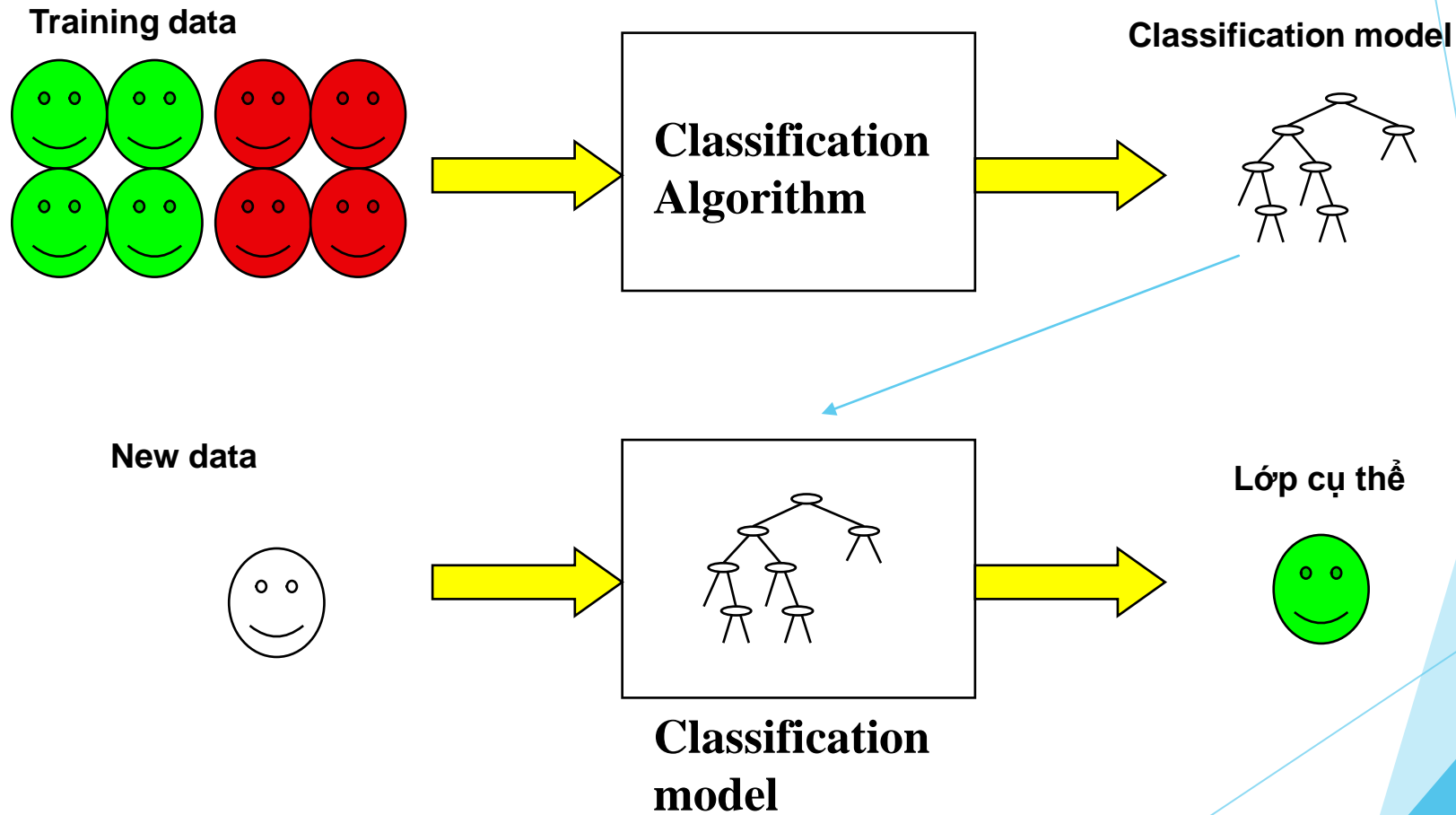
Thuật toán phân lớp

- ▶ Phân lớp dữ liệu: Là xếp các đối tượng dữ liệu vào trong một lớp đã được xác định trước.
- ▶ Phân lớp gồm 2 bước:
Bước 1: Xây dựng mô hình
- ▶ Bước 2: Vận hành mô hình.



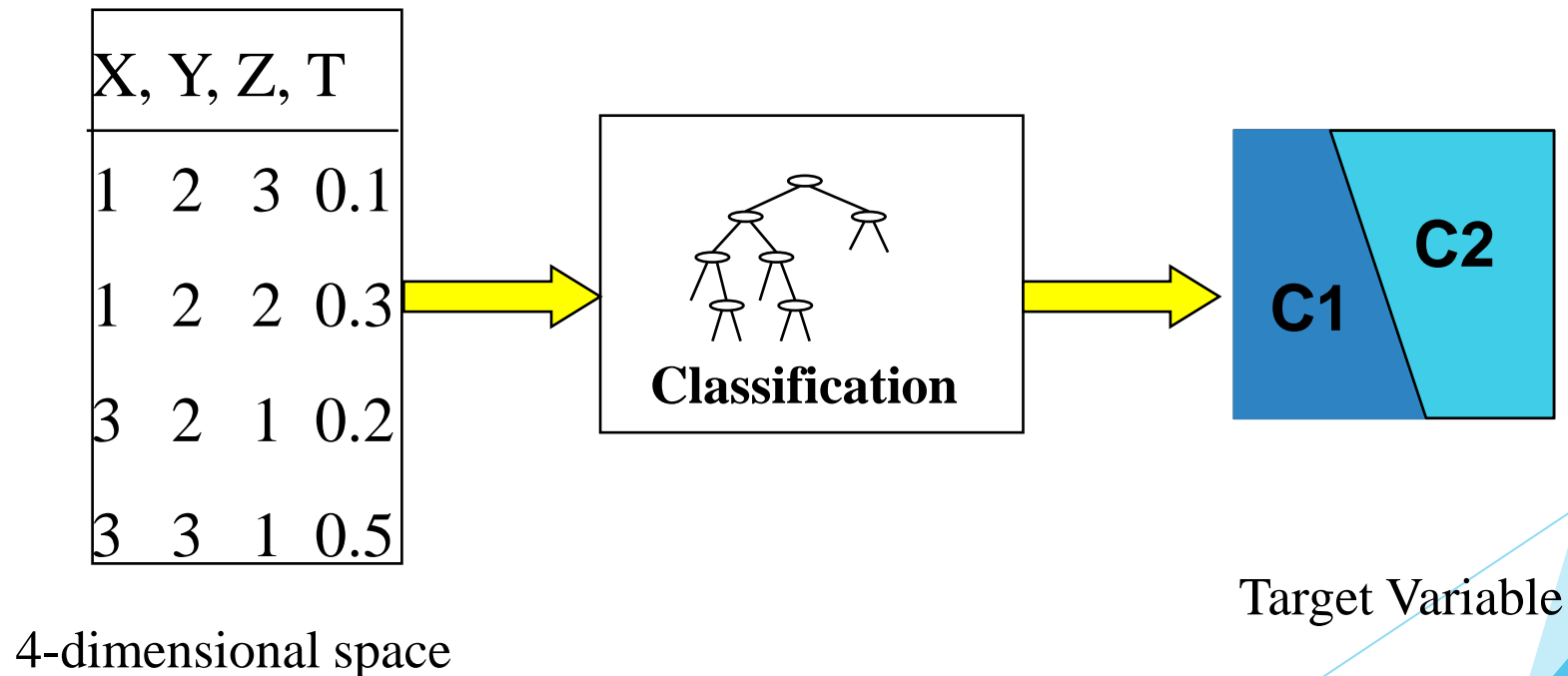
- ▶ Bước 1: Xây dựng mô hình
 - ▶ Mô tả dữ liệu đã biết
 - ▶ Mỗi một mẫu thuộc về một lớp cụ thể
 - ▶ Tìm luật phân lớp để xây dựng mô hình
 - ▶ Bayesian classifiers
 - ▶ Decision trees
 - ▶ Neural networks
 - ▶ Genetic algorithms
 - ▶ K-nearest neighbors
 - ▶ Support Vector Machines
- ▶ Bước 2: Vận hành
 - ▶ Phân lớp đối tượng chưa biết
 - ▶ Kiểm thử độ chính xác của mô hình

Mô hình phân lớp

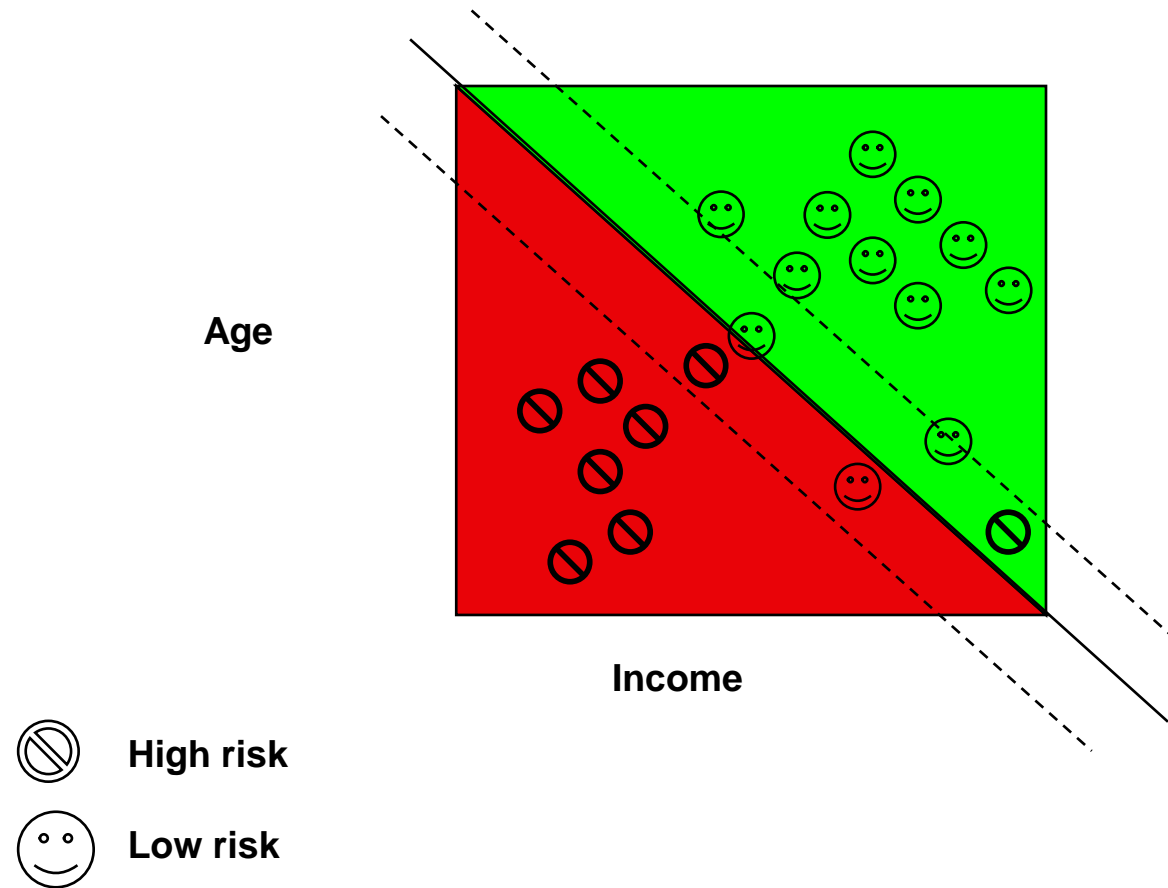


Không gian mô tả và biến dự đoán (target variable)

- ▶ Trong việc, phân lớp: Thông thường, một tập hợp các đối tượng (object) được mô tả là một tập hợp d thuộc tính (hoặc biến hoặc đặc trưng), d - tạo ra không gian d -chiều. Mỗi một đối tượng là một điểm (point) trong không gian và được diễn tả như là một bản ghi hoặc một vector.



Xác định các vùng quyết định tối ưu và phân lớp nhằm



Một số vấn đề trong phân lớp

- ▶ Xác định biến mục tiêu (target class) ???????
- ▶ Không gian mô tả: Sự lựa chọn các đặc trưng (thuộc tính)
- ▶ Lựa chọn thuật toán
- ▶ Kích thước của training data
- ▶ Đánh giá mô hình

Nội dung

1. Classification
2. Cây quyết định
3. Randomforest
4. Sampling
5. Bayesian Decision Theory
6. Naïve Bayes Classification

Cây quyết định

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

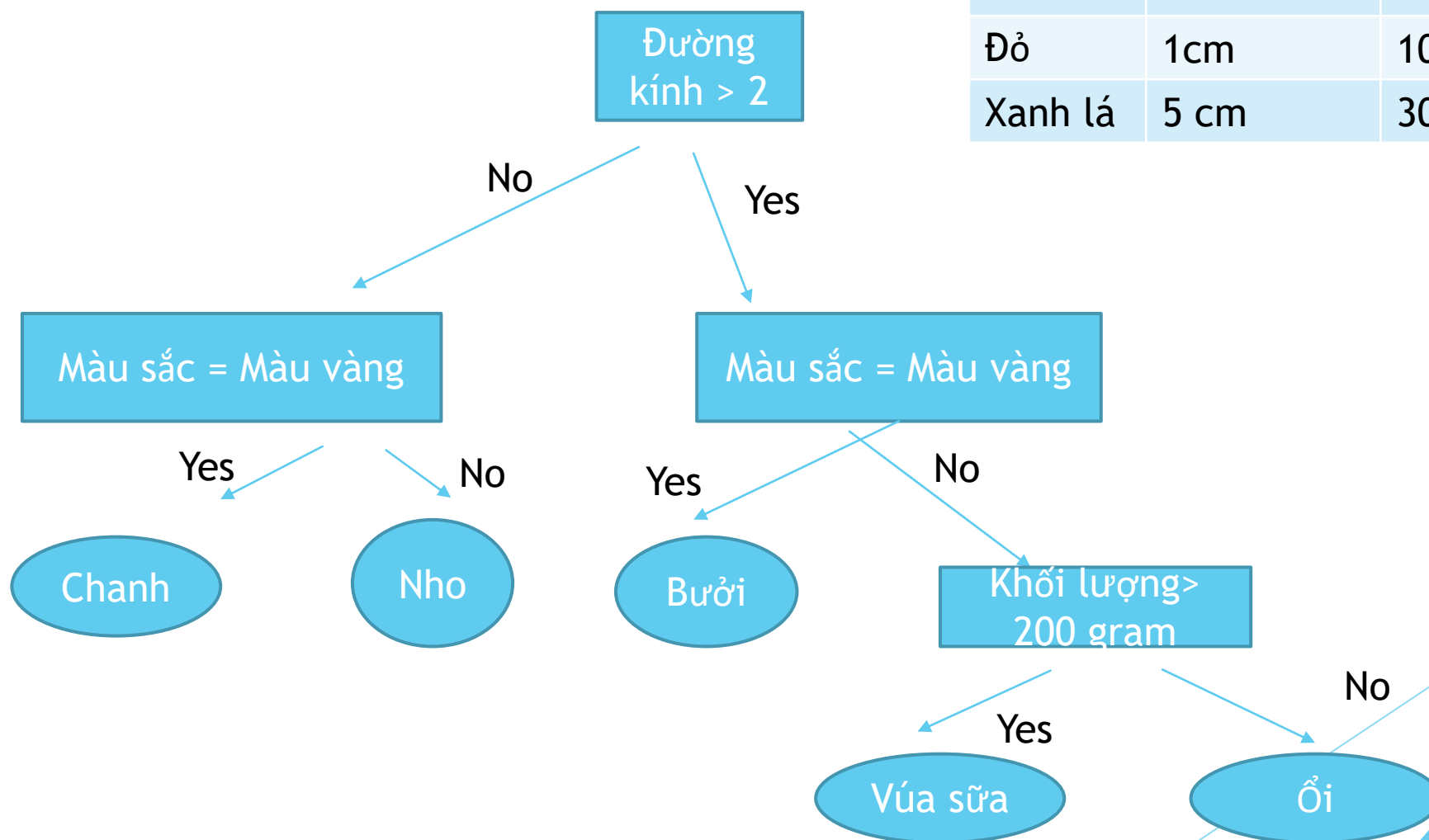
Thuật toán cây quyết định

Thuật toán cây quyết định

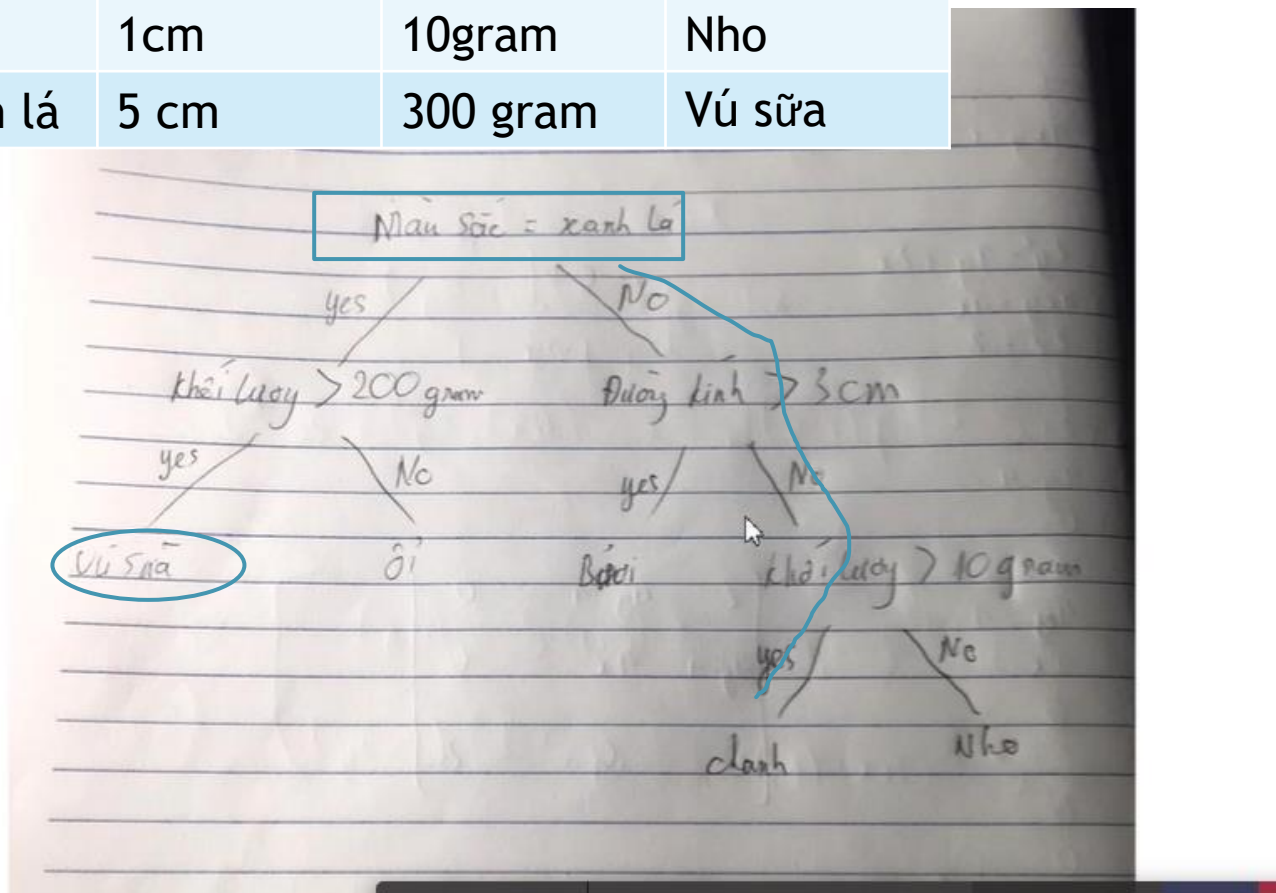
- ▶ Sử dụng rộng rãi trong lĩnh vực data mining.
- ▶ Được phát triển trong lĩnh vực học máy và thống kê
- ▶ Sử dụng để tạo ra các mô hình về hồi quy, dự đoán và phân lớp.

Ví dụ: Phân loại hoa quả.

Màu sắc	Đường kính	Khối lượng	Loại quả
Vàng	2 cm	50 gram	Chanh
Vàng	15 cm	500 gram	Bưởi
Xanh lá	5cm	200 gram	Ổi
Đỏ	1cm	10gram	Nho
Xanh lá	5 cm	300 gram	Vú sữa



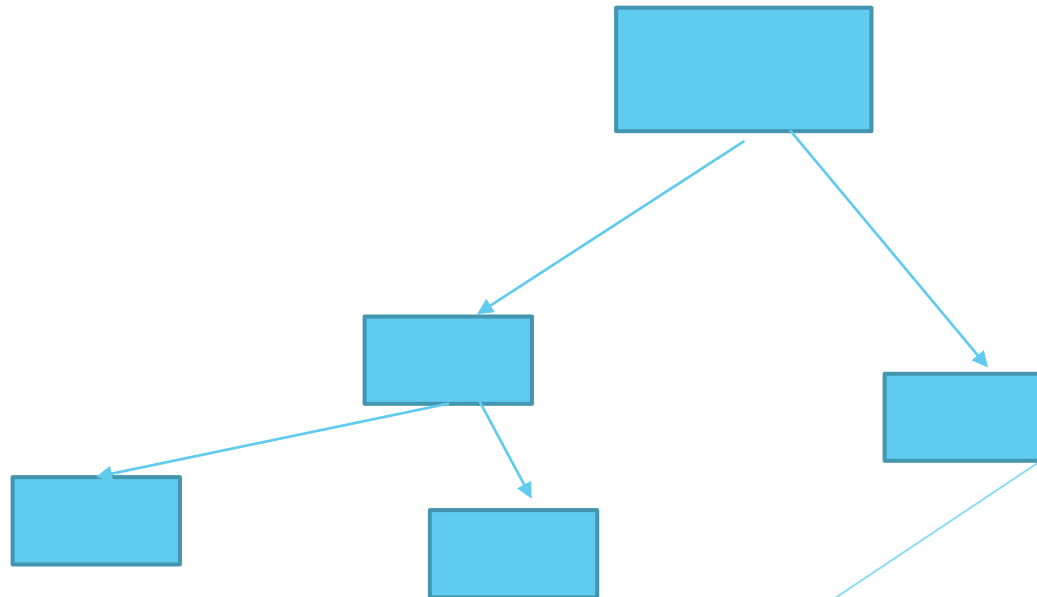
Màu sắc	Đường kính	Khối lượng	Loại quả
Vàng	2 cm	50 gram	Chanh
Vàng	15 cm	500 gram	Bưởi
Xanh lá	5cm	200 gram	Ổi
Đỏ	1cm	10gram	Nho
Xanh lá	5 cm	300 gram	Vú sữa



- ▶ Cây quyết định có thể được minh họa bằng?
- ▶ Một tập hợp các quy luật (set of rules):
 - ▶ Quả bưởi = {Đường kính >2cm, màu vàng}
 - ▶ Quả chanh = {Đường kính không lớn hơn 2cm, màu vàng}

Mô hình cây quyết định

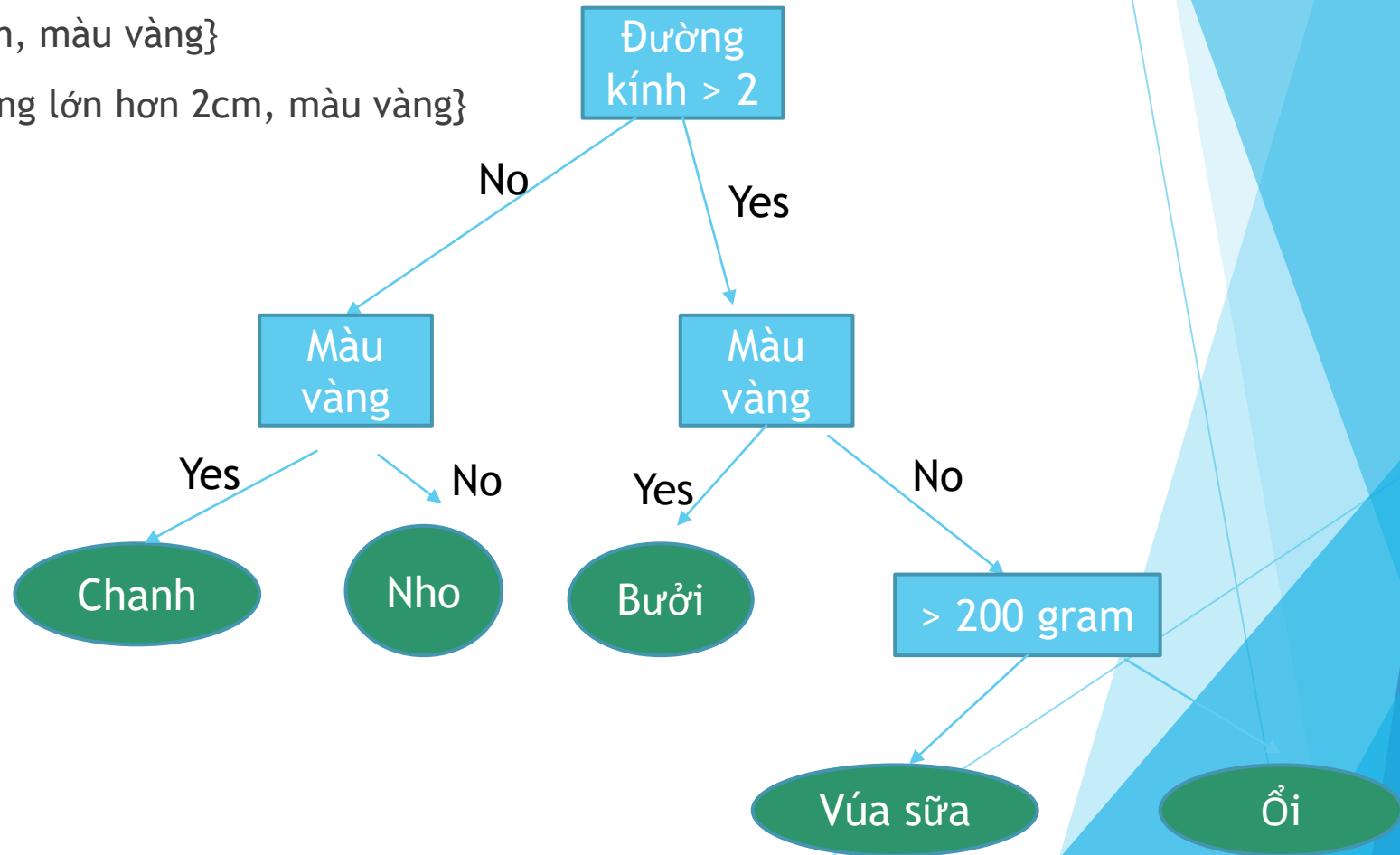
- ▶ Một node thể hiện một đặc trưng (thuộc tính, tính chất của dữ liệu)
- ▶ Một nhánh mô tả một quy luật của dữ liệu
- ▶ Mỗi lá biểu diễn một kết quả phân lớp.
- ▶ Tại mỗi node, thuộc tính (đặc trưng) được chọn dùng để chia dữ liệu thành các quy luật.



Từ cây tạo ra một tập hợp các quy luật

- ▶ Một tập hợp các quy luật (set of rules):

- ▶ Quả bưởi = {Đường kính > 2cm, màu vàng}
- ▶ Quả chanh = {Đường kính không lớn hơn 2cm, màu vàng}



Tri thức rút ra từ cây quyết định

- ▶ Cây quyết định và tập hợp các luật
- ▶ Dữ liệu training nằm hoàn toàn thể hiện trong cây (lá hoặc các luật)
- ▶ Độ tin cậy của phân loại thể hiện ở các lá (hoặc rule)

Điểm mạnh của cây quyết định

- ▶ Cây và luật là dễ hiểu
- ▶ Lựa chọn các thuộc tính quan trọng để phân lớp
- ▶ Dùng cho cả thuộc tính dạng số và dạng nhóm (category)
- ▶ Hiệu quả xử lý dữ liệu lớn.

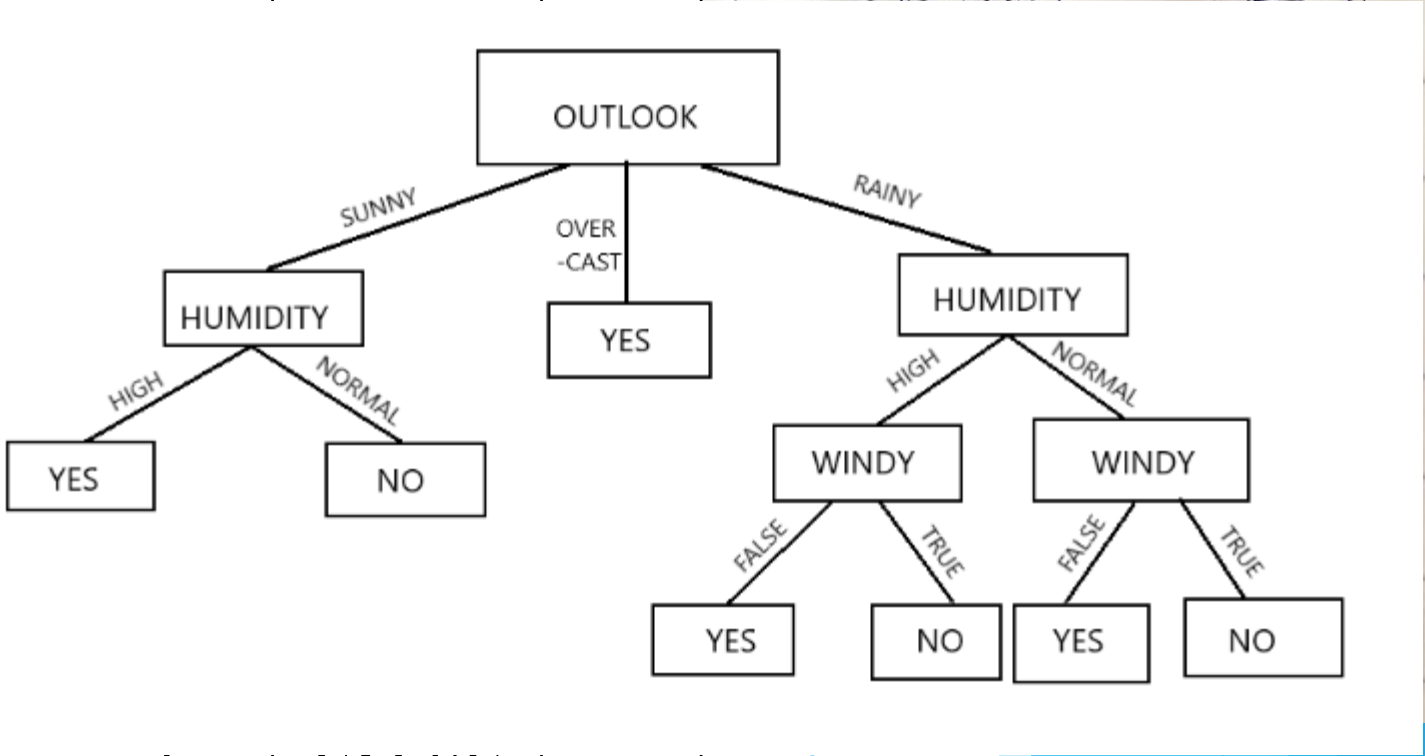
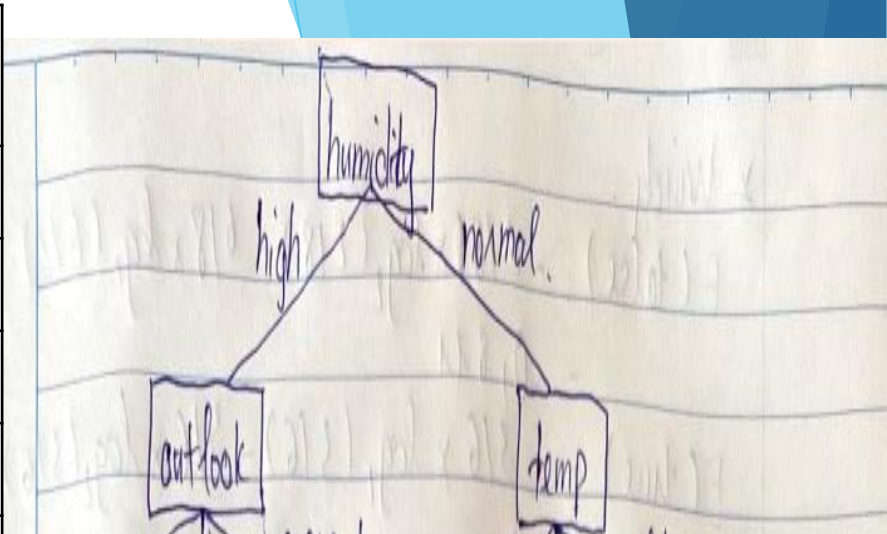
Các thuật toán phổ biến xây dựng cây quyết định

- ▶ ID3, C4.5, C5.0 (Ross Quinlan 1986,1993)
- ▶ CART (Leo Briemen, et al 1984)
- ▶ CHAID (J. A. Hartigan, 1975)

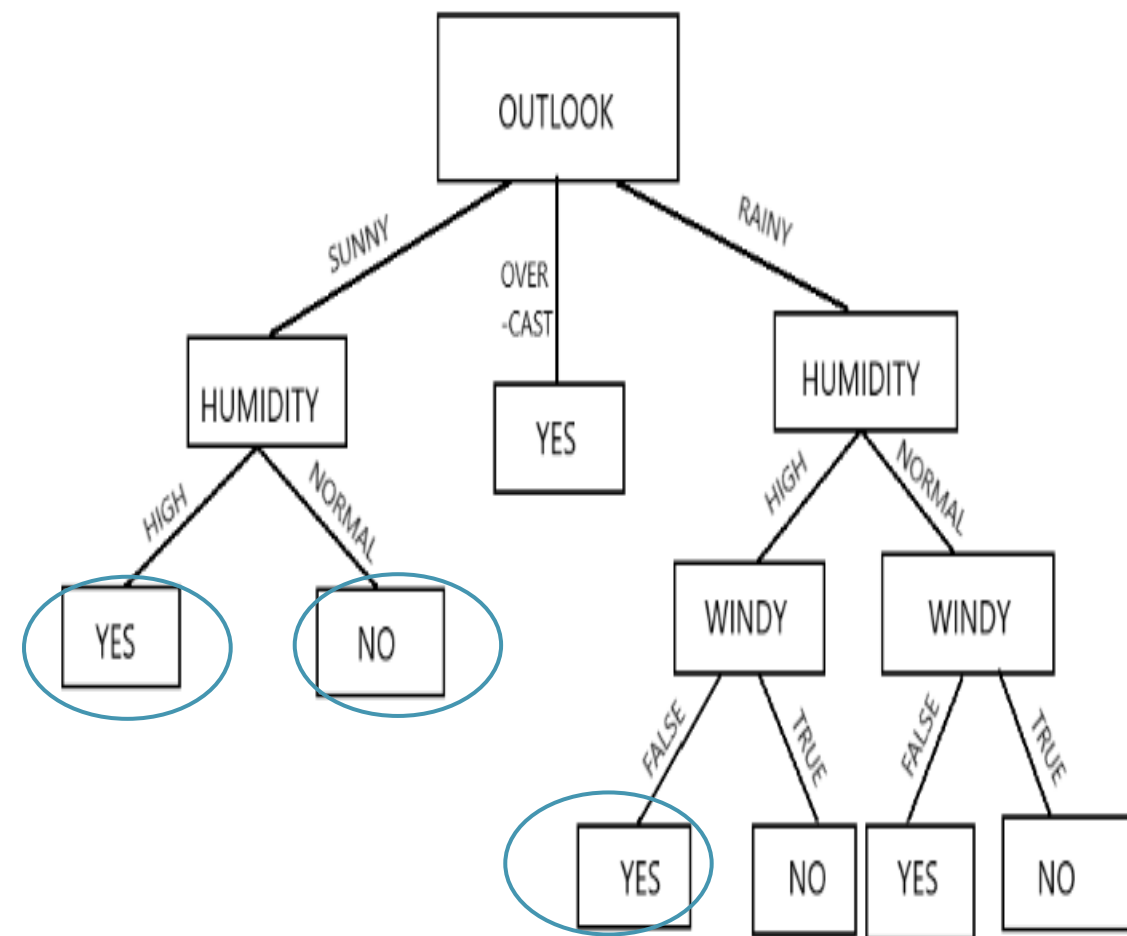
Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

Thời tiết = {rainy, hot, high, false}
Play = YES or NO ???

	Outlook	Temperature	Humidity	Windy	Play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool			
6	rainy	cool			
7	overcast	cool			
8	sunny	mild			
9	sunny	cool			
10	rainy	mild			
11	sunny	mild			
12	overcast	mild			
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no



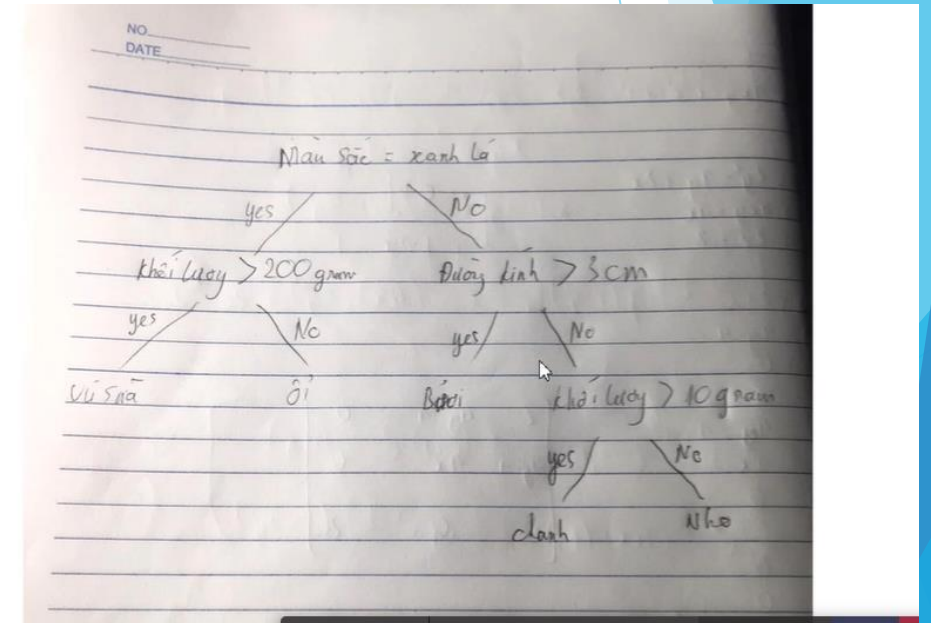
	Outlook	Temperature	Humidity	Windy	Play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no



Xây dựng cây quyết định:

► Xây dựng cây quyết định:

- Phát triển cây quyết định: đi từ gốc, đến các nhánh, phát triển quy nạp theo hình thức chia để trị.
 1. Chọn thuộc tính “tốt” nhất bằng một độ đo đã định trước
 2. Phát triển cây bằng việc thêm các nhánh tương ứng với từng giá trị của thuộc tính đã chọn
 3. Sắp xếp, phân chia tập dữ liệu đào tạo tới node con
 4. Nếu các samples được phân lớp rõ ràng thì dừng.
 5. Ngược lại: lặp lại bước 1 tới bước 4 cho từng node con



Xây dựng cây quyết định:

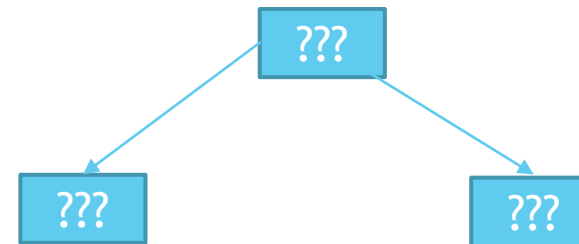
► Xây dựng cây quyết định:

- Phát triển cây quyết định: đi từ gốc, đến các nhánh, phát triển quy nạp theo hình thức chia để trị.
 1. Chọn thuộc tính “tốt” nhất bằng một độ đo đã định trước
 2. Phát triển cây bằng việc thêm các nhánh tương ứng với từng giá trị của thuộc tính đã chọn
 3. Sắp xếp, phân chia tập dữ liệu đào tạo tới node con
 4. Nếu các samples được phân lớp rõ ràng thì dừng.
 5. Ngược lại: lặp lại bước 1 tới bước 4 cho từng node con

	Outlook	Temperature	Humidity	Windy	Play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Chọn một thuộc tính để chia

- ▶ Tại mỗi node, các thuộc tính sẽ được đánh giá dựa trên việc chia các lớp mục tiêu (target class) của toàn bộ dữ liệu training.
- ▶ Một đơn vị đo sẽ được sử dụng: ví dụ impurity (hỗn tạp)
- ▶ Một số kiểu dùng để đo impurity.
 - ▶ Information gain (ID3/C4.5)
 - ▶ Information gain ratio (C4.5)
 - ▶ Gini index (CART)
 - ▶ χ^2 test (CHAID)



Tiêu chuẩn để chọn thuộc tính

- ▶ Thuộc tính nào là tốt nhất?
 - ▶ Đưa ra cây nhỏ nhất (tối ưu nhất)
 - ▶ Kinh nghiệm: Chọn thuộc tính mà nó tạo ra các node trong suốt nhất (purest)
- ▶ Đơn vị đo impurity phổ thông: information gain
 - ▶ Information gain tăng lên theo độ trong suốt trung bình của các tập con mà thuộc tính tạo ra
- ▶ Chiến lược: Chọn một thuộc tính mà tạo ra information gain lớn nhất.

Entropy

- ▶ $\text{Entropy}(S) = \sum_{(i=1 \text{ to } C)} -|S_i| / |S| * \log_2(|S_i| / |S|)$
 - ▶ S = tập mẫu
 - ▶ S_i = Tập con S_i
 - ▶ C = Số lượng các lớp;

Information gain

$$IG(S, A) = Entropy(S) - \sum_{v \in A} \frac{|S_v|}{|S|} * Entropy(A_v)$$

Entropy (S): Thông tin entropy trước khi chia.

A: là một đặc trưng

v: là một giá trị của đặc trưng A

|S_v|: là số mẫu khi đặc trưng A mang giá trị v;

|S|: là số tổng số mẫu trước khi chia

Entropy (A_v) là thông tin Entropy của đặc trưng A mang giá trị v

Outlook

sunny

sunny

overcast

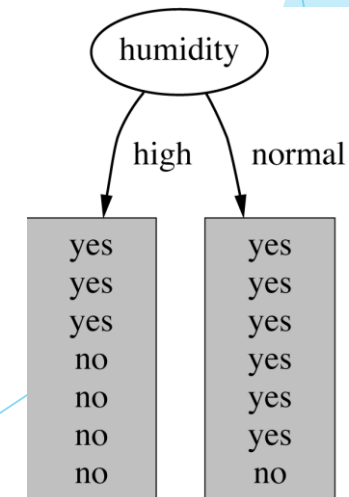
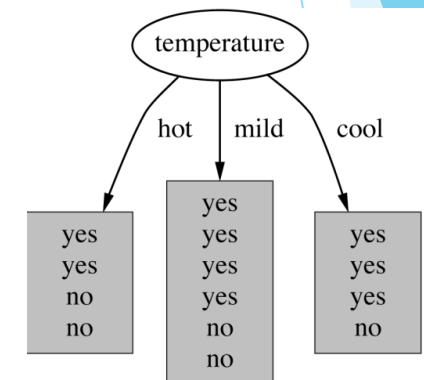
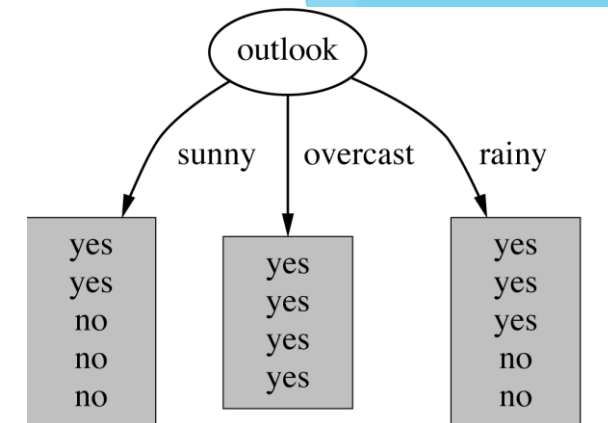
rainy

rainy

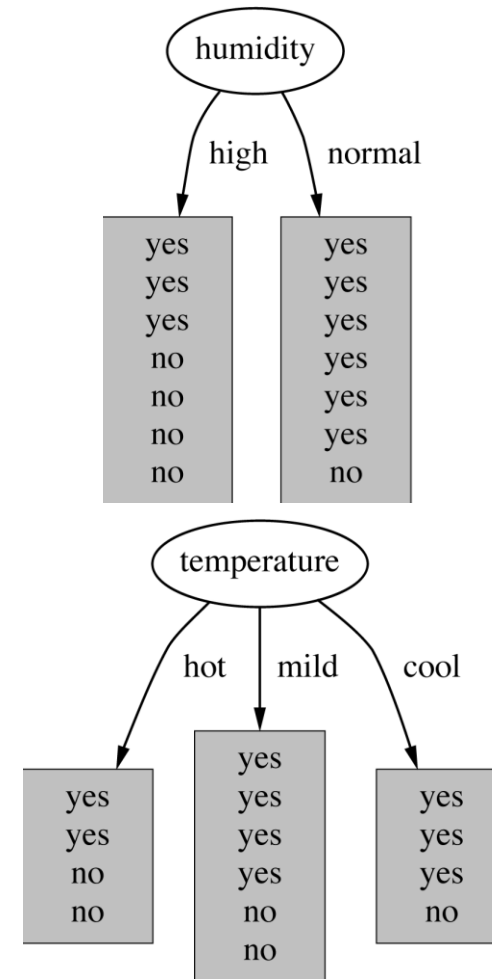
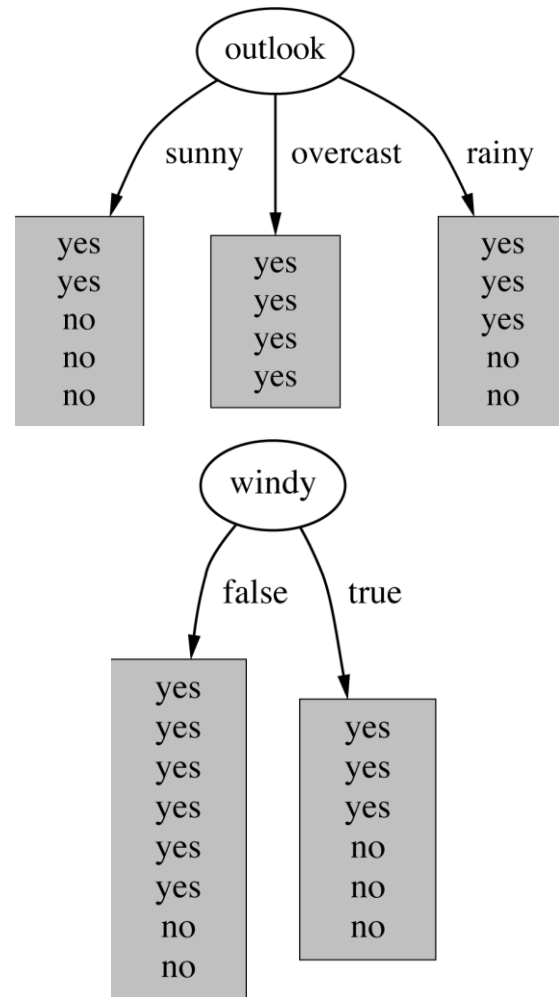
rainy

overcast

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no



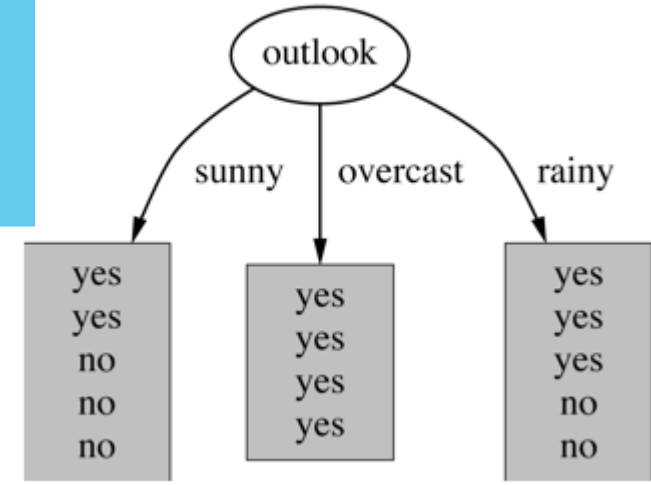
Các khả năng chọn thuộc tính



Tính Inform

$$\text{Entropy}(S) = \sum_{(i=1 \text{ to } C)} -|S_i|/|S| * \log_2(|S_i|/|S|)$$

- ▶ S = tập mẫu
- ▶ S_i = Tập con S_i
- ▶ C = Số lượng các lớp;



- ▶ Tính Entropy (S): trước khi chia nhánh.

- ▶ $\text{Entropy}(S) = - (5/14) * \log_2(5/14) - (9/14) * \log_2(9/14) = 0.940$

- ▶ Chọn đặc trưng Outlook để chia:

- ▶ $E(\text{outlook} = \text{sunny}) = -3/5 * \log_2(3/5) - 2/5 * \log_2(2/5) = 0.971$

- ▶ $E(\text{outlook} = \text{overcast}) = -0/4 * \log_2(0/4) - 4/4 * \log_2(4/4) = 0$

- ▶ $E(\text{outlook} = \text{rainy}) = -2/5 * \log_2(2/5) - 3/5 * \log_2(3/5) = 0.971$

Coi là giá trị là 0

- ▶ Thông tin trung bình Entropy(Outlook) =

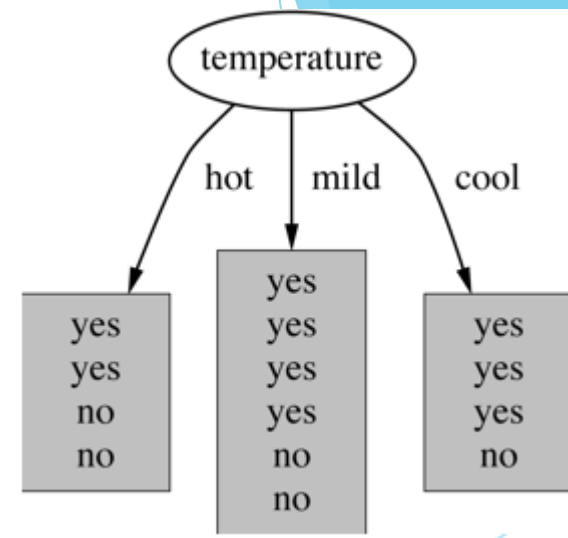
$$= 5/14 * E(\text{outlook} = \text{sunny}) + 4/14 * E(\text{outlook} = \text{overcast}) + 5/14 * E(\text{outlook} = \text{rainy}) = 0.693$$

- ▶ $\text{IG}(S, \text{outlook}) = \text{Entropy}(S) - \text{Entropy}(\text{outlook}) = 0.940 - 0.693 = 0.247$

Tính Information Gain (Temp.)

- ▶ Chọn đặc trưng Temp. để chia

- ▶ $E(\text{temp.} = \text{hot}) = -2/4 * \log_2(2/4) - 2/4 * \log_2(2/4) = 1$
- ▶ $E(\text{temp.} = \text{mild}) = -2/6 * \log_2(2/6) - 4/6 * \log_2(4/6) = 0.918$
- ▶ $E(\text{temp.} = \text{cool}) = -1/4 * \log_2(1/4) - 3/4 * \log_2(3/4) = 0.811$



- ▶ Thông tin trung bình Entropy(temp.):

- ▶ $E(\text{temp.}) = 4/14 * E(\text{temp.} = \text{hot}) + 6/14 * E(\text{temp.} = \text{mild}) + 4/14 * E(\text{temp.} = \text{cool}) = 0.911$

- ▶ $IG(S, \text{temp.}) = 0.940 - 0.911 = 0.029$

Tính Information Gain (humidity)

- ▶ Chọn đặc trưng humidity để chia

- ▶ $E(\text{humidity}) = \frac{7}{14} * (\frac{4}{7} * \log_2 (\frac{4}{7}) + \frac{3}{7} * \log_2 (\frac{3}{7})) + \frac{7}{14} * (\frac{1}{7} * \log_2 (\frac{1}{7}) + \frac{6}{7} * \log_2 (\frac{6}{7})) = 0.788$

- ▶ $IG(S, \text{humidity}) = 0.940 - 0.788 = 0.152$

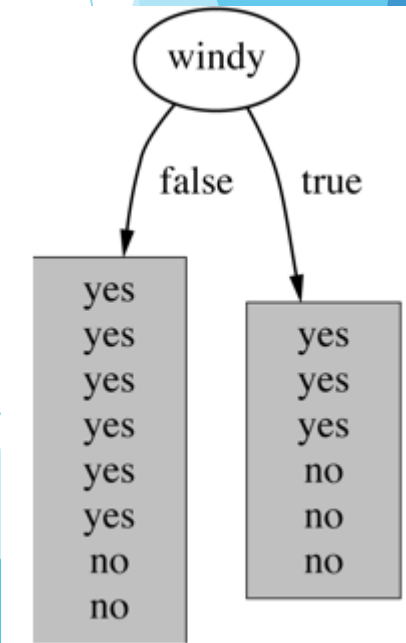
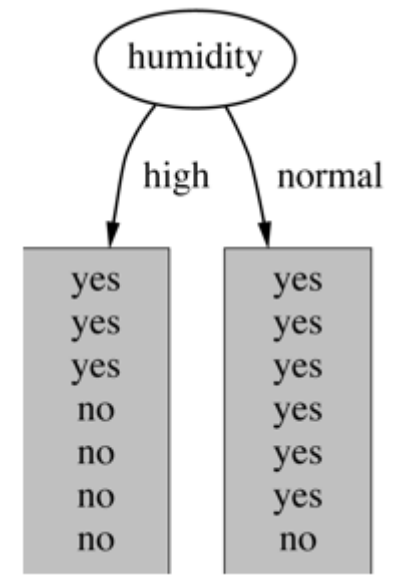
- ▶ Chọn đặc trưng Windy để chia

- ▶ $E(\text{windy}) = \frac{8}{14} * (\frac{2}{8} * \log_2 (\frac{2}{8}) + \frac{6}{8} * \log_2 (\frac{6}{8})) + \frac{6}{14} * (\frac{3}{6} * \log_2 (\frac{3}{6}) + \frac{3}{6} * \log_2 (\frac{3}{6})) = 0.892$

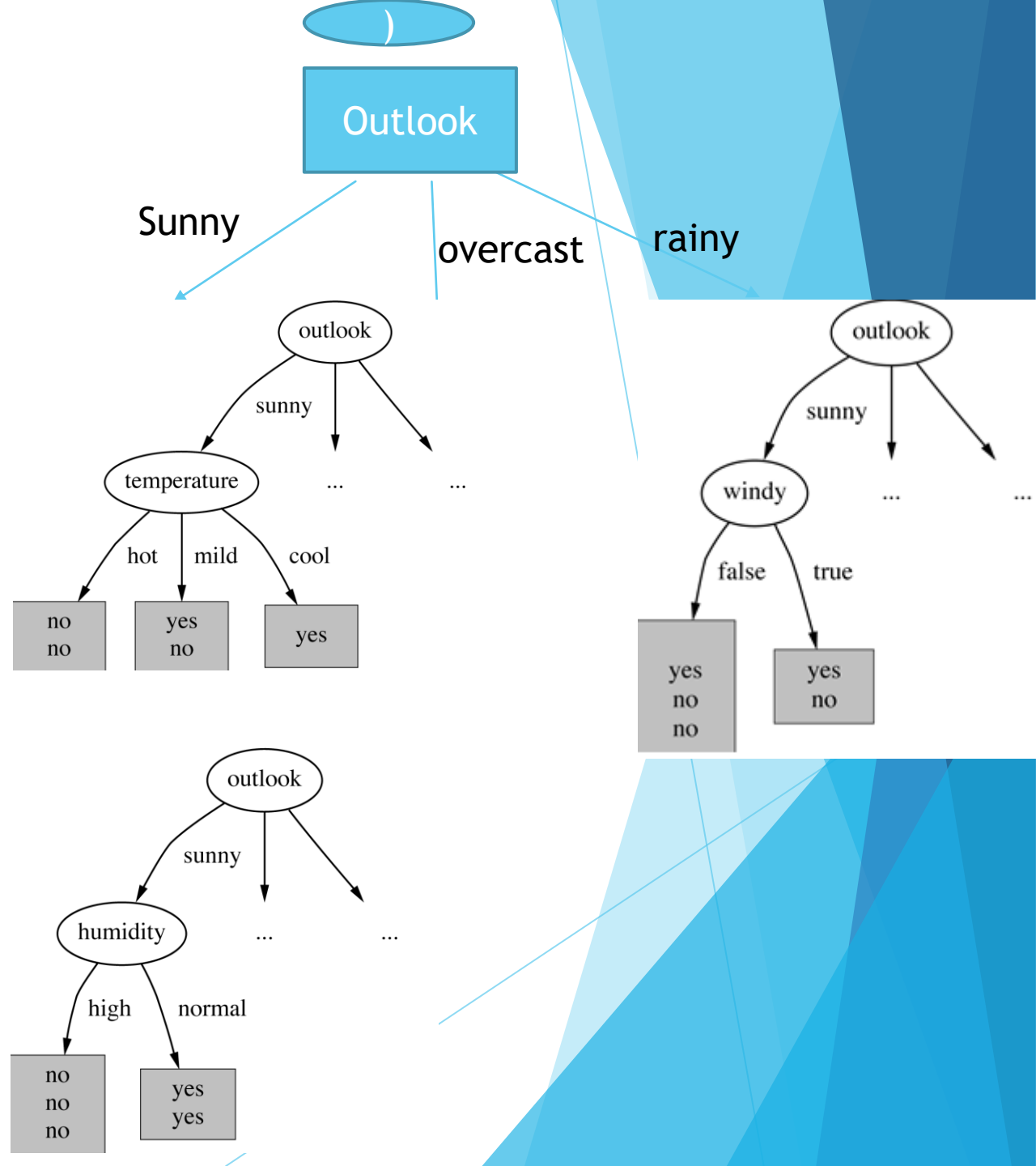
- ▶ $IG(S, \text{windy}) = 0.940 - 0.892 = 0.048$

- ▶ $IG(S, \text{outlook}) = 0.247$; $IG(S, \text{temp.}) = 0.029$; $IG(S, \text{humidity}) = 0.152$; $IG(S, \text{windy}) = 0.048$;

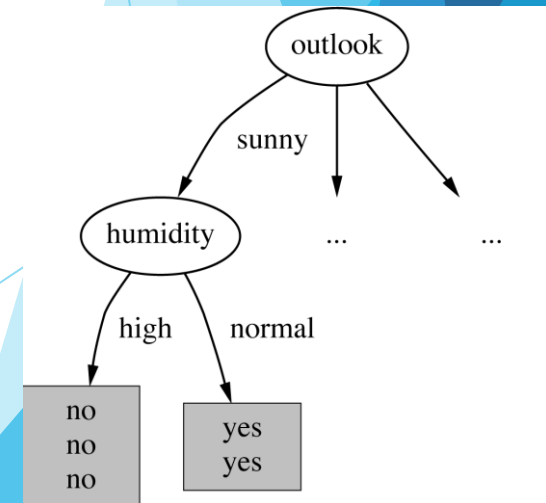
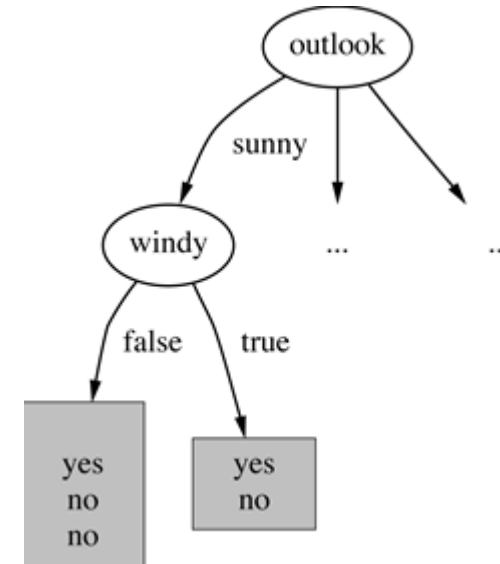
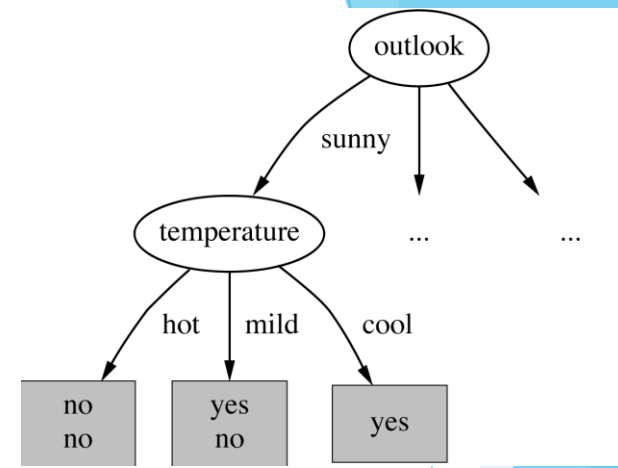
- ▶ Vậy chọn đặc trưng **outlook** là nốt chia đầu tiên (nốt gốc)



Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no



Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no



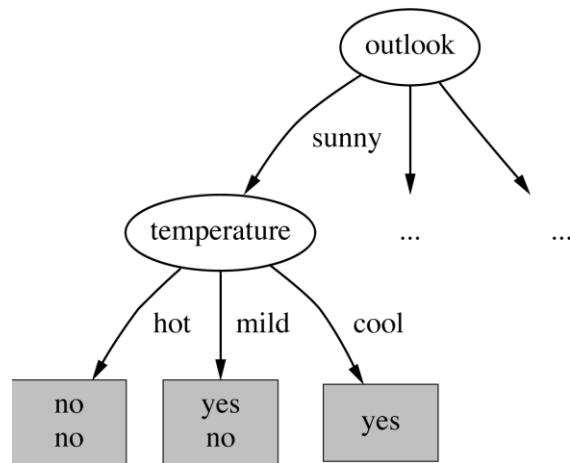
- ▶ $E(\text{outlook}, \text{Temp} = \text{mild}) = -1/2 * \log_2(1/2) - 1/2 * \log_2(1/2)$
- ▶ $E(\text{outlook}, \text{Temp} = \text{cool}) = -0/1 * \log_2(0/1) - 1/2 * \log_2(1/1) = 0$

Tiếp tục chia nốt

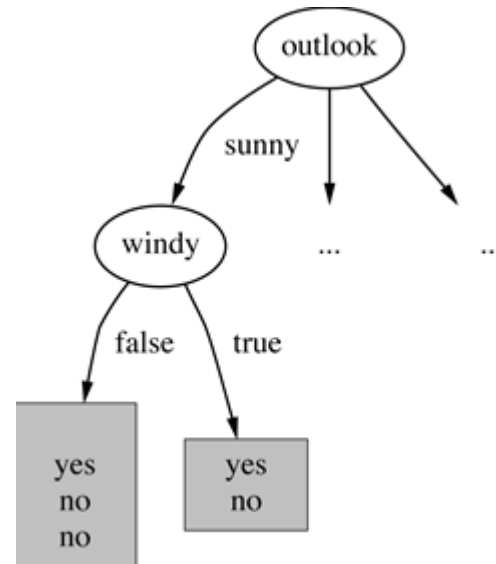
- ▶ $E(S) = E(\text{outlook} = \text{sunny}) = 0.971$
- ▶ $E(\text{humidity} = \text{high}) = -3/3 * \log(3/3) = 0$
- ▶ $E(\text{humidity} = \text{normal}) = -2/2 * \log(2/2) = 0$
- ▶ $IG(\text{outlook} = \text{sunny}, \text{humidity}) = 0.971 - 0 = 0.971$
- ▶ $E(\text{temp.} = \text{hot}) = -2/2 * \log(2/2) = 0$
- ▶ $E(\text{temp.} = \text{mild}) = -1/2 * \log(1/2) - 1/2 * \log(1/2) = 1$
- ▶ $E(\text{temp.} = \text{cool}) = -1/1 * \log(1/1) = 0;$
- ▶ $E(\text{temp.}) = 2/5 * 0 + 2/5 * 1 + 1/5 * 0 = 0.4$
- ▶ $IG(\text{outlook} = \text{sunny}, \text{temp.}) = E(\text{outlook} = \text{sunny}) - E(\text{temp.}) = 0.971 - 0.4 = 0.571$

- ▶ $E(\text{windy} = \text{false}) = -2/3 * \log(2/3) - 1/3 * \log(1/3) = 0.918$
- ▶ $E(\text{windy} = \text{true}) = -1/2 * \log(1/2) - 1/2 * \log(1/2) = 1$
- ▶ $E(\text{windy}) = 3/5 * 0.918 + 2/5 * 1 = 0.951$
- ▶ $IG(\text{outlook} = \text{sunny}, \text{windy}) = E(\text{outlook} = \text{sunny}) - E(\text{windy}) = 0.971 - 0.951 = 0.02$

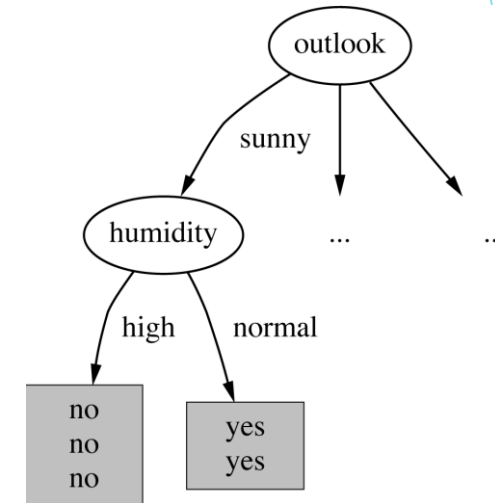
- ▶ $IG(\text{outlook} = \text{sunny}, \text{humidity}) = \mathbf{0.971}$
- ▶ $IG(\text{outlook} = \text{sunny}, \text{temp.}) = \mathbf{0.571}$
- ▶ $IG(\text{outlook} = \text{sunny}, \text{windy}) = 0.02$



$\text{gain}(\text{"Temperature"}) = 0.571 \text{ bits}$



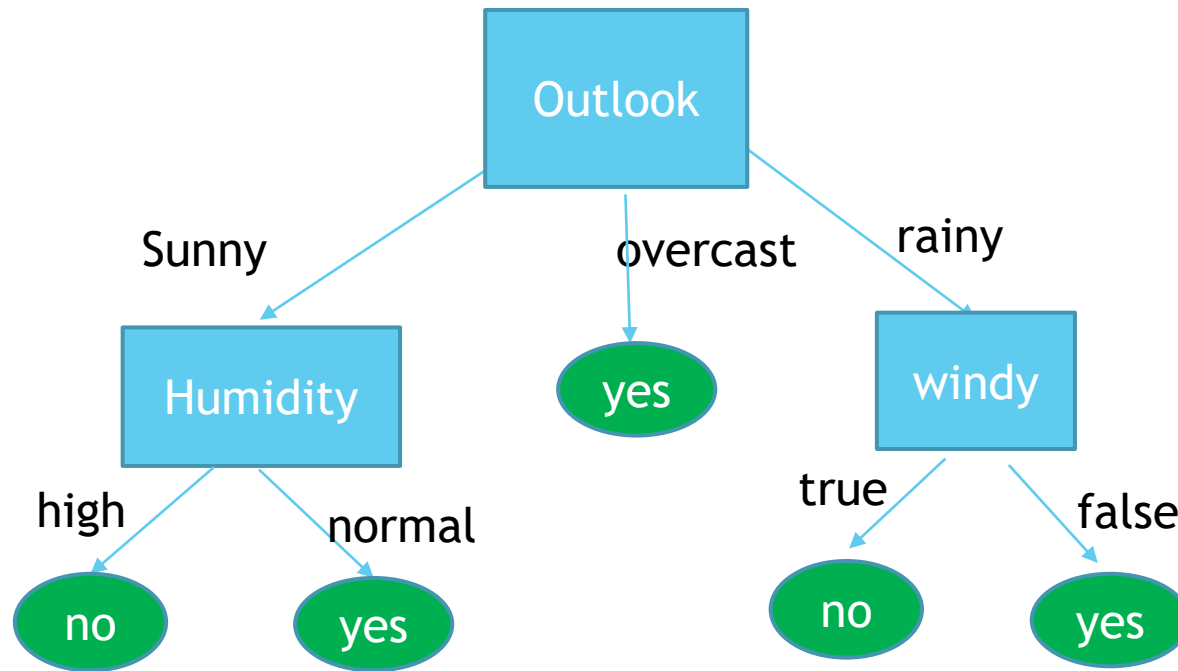
$\text{gain}(\text{"Windy"}) = 0.020 \text{ bits}$



$\text{gain}(\text{"Humidity"}) = 0.971 \text{ bits}$

Điều kiện dừng

- ▶ Một lớp mà có lượng một lớp mục tiêu quá nhiều so với các lớp mục tiêu khác
 - ▶ e.g., >90%
- ▶ Số lượng các đối tượng trong các tập con tại một node nhỏ hơn nhiều giá trị ngưỡng (threshold)
- ▶ Giảm sút trong giá trị IG



{Sunny, Cool, High, True}

Thời tiết = {rainy, hot, high, false}
Play = YES or NO ???

Outlook Temperature Humidity Windy

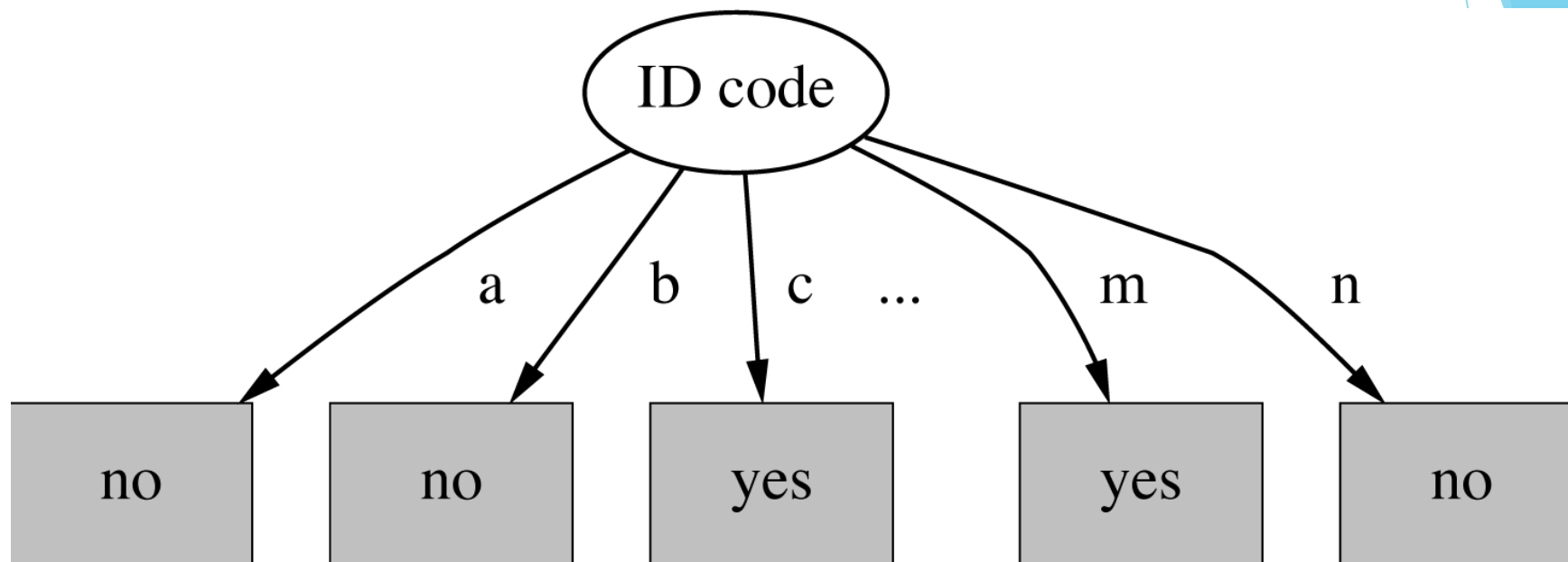
Một số vấn đề khi xây dựng cây

- ▶ Các thuộc tính có nhiều giá trị (trường hợp cực đoan: mã ID)
 - ▶ IG sẽ bị bias khi chọn những thuộc tính các giá trị lớn.
 - ▶ Điều này có thể dẫn đến kết quả là overfitting

Weather Data with ID code

ID	Outlook	Temperature	Humidity	Windy	Play?
A	sunny	hot	high	false	No
B	sunny	hot	high	true	No
C	overcast	hot	high	false	Yes
D	rain	mild	high	false	Yes
E	rain	cool	normal	false	Yes
F	rain	cool	normal	true	No
G	overcast	cool	normal	true	Yes
H	sunny	mild	high	false	No
I	sunny	cool	normal	false	Yes
J	rain	mild	normal	false	Yes
K	sunny	mild	normal	true	Yes
L	overcast	mild	high	true	Yes
M	overcast	hot	normal	false	Yes
N	rain	mild	high	true	No

Chia cho thuộc tính ID



Entropy of split = 0; mỗi lá là một trường hợp cụ thể và là pure
ID code sẽ có giá trị IG cao nhất

Overfitting : FAIL

Gain Ratio and Intrinsic Information

- ▶ C4.5 Dùng gain ratio để chọn ra đặc trưng tốt nhất
- ▶ Intrinsic information: sự phân bố của các mẫu vào các nhánh

$$\textit{IntrinsicInfo}(S, A) \equiv -\sum \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}.$$

- ▶ *Gain ratio* (Quinlan'86) :

$$\textit{GainRatio}(S, A) = \frac{\textit{Gain}(S, A)}{\textit{IntrinsicInfo}(S, A)}.$$

Gain Ratios cho các đặc trưng

$$IntrinsicInfo(S,A) \equiv -\sum \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}.$$

Intrinsic infor (S, outlook)= 5/14 * log (5/14) + 4/14 *log (4/14)+ 5/14 *log(5/14) = 1.577

Outlook		Temperature	
Info:	0.693	Info:	0.911
Gain: 0.940-0.693	0.247	Gain: 0.940-0.911	0.029
Split info: info([5,4,5])	1.577	Split info: info([4,6,4])	1.362
Gain ratio: 0.247/1.577	0.156	Gain ratio: 0.029/1.362	0.021

Humidity		Windy	
Info:	0.788	Info:	0.892
Gain: 0.940-0.788	0.152	Gain: 0.940-0.892	0.048
Split info: info([7,7])	1.000	Split info: info([8,6])	0.985
Gain ratio: 0.152/1	0.152	Gain ratio: 0.048/0.985	0.049

CART Decision Tree Algorithm

- ▶ Phát triển Breiman, Friedman, Olshen, Stone (1984), *Classification and Decision Trees*
- ▶ Sử dụng Gini để chia
- ▶ Tạo ra cây nhị phân

CART Split Criterion: Gini Index

- ▶ Cho một dataset T^i chứa n lớp, gini index (T^i) được tính như sau:

$$gini(T^i) = 1 - \sum_{j=1}^n p_j^2$$

- ▶ p_j là tần xuất của một đối tượng được phân loại cụ thể trong dataset T^i .

Gini Index

- ▶ Sau khi chia Dataset T thành 2 lớp con T_1 và T_2 với size N_1 và size N_2 , chỉ số Gini index của việc chia Dataset T được định nghĩa như sau:

$$gini_{split}(T) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

- ▶ Thuộc tính mà đưa ra giá trị $Gini_{split}$ nhỏ nhất, sẽ được chọn để chia node

$$gini_{split}(T) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

$$gini(T^i) = 1 - \sum_{j=1}^n p_j^2$$

► Ví dụ: Outlook

		Play golf			
		=====			
		yes no			

Rainy	yes	3	2	5	
	no	6	3	9	

		9	5		

			Play golf	
			=====	
			yes no	

Sunny	yes	2	3	5
	no	7	2	9

		9	5	

		Play golf			
		=====			
		yes no			

overcast	yes	4		0	4
	no	5		5	10

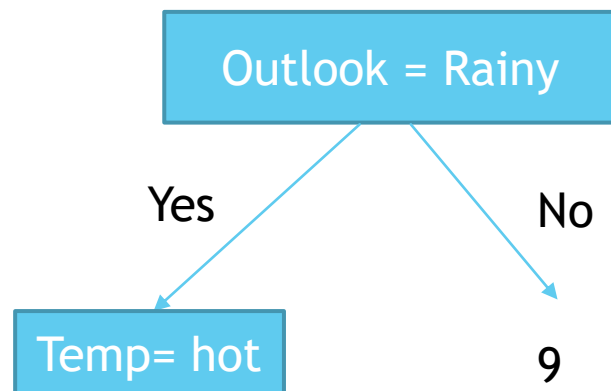
		9		5	

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

Play golf				
=====				
yes no				

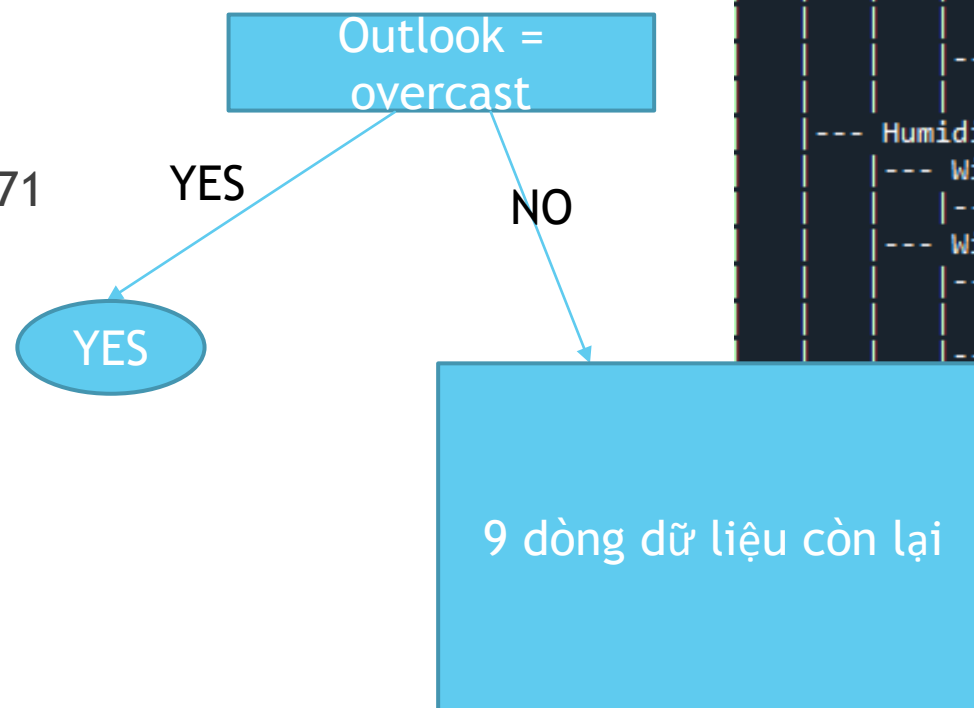
Rainy	yes	3	2	5
	no	6	3	9

		9	5	



Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

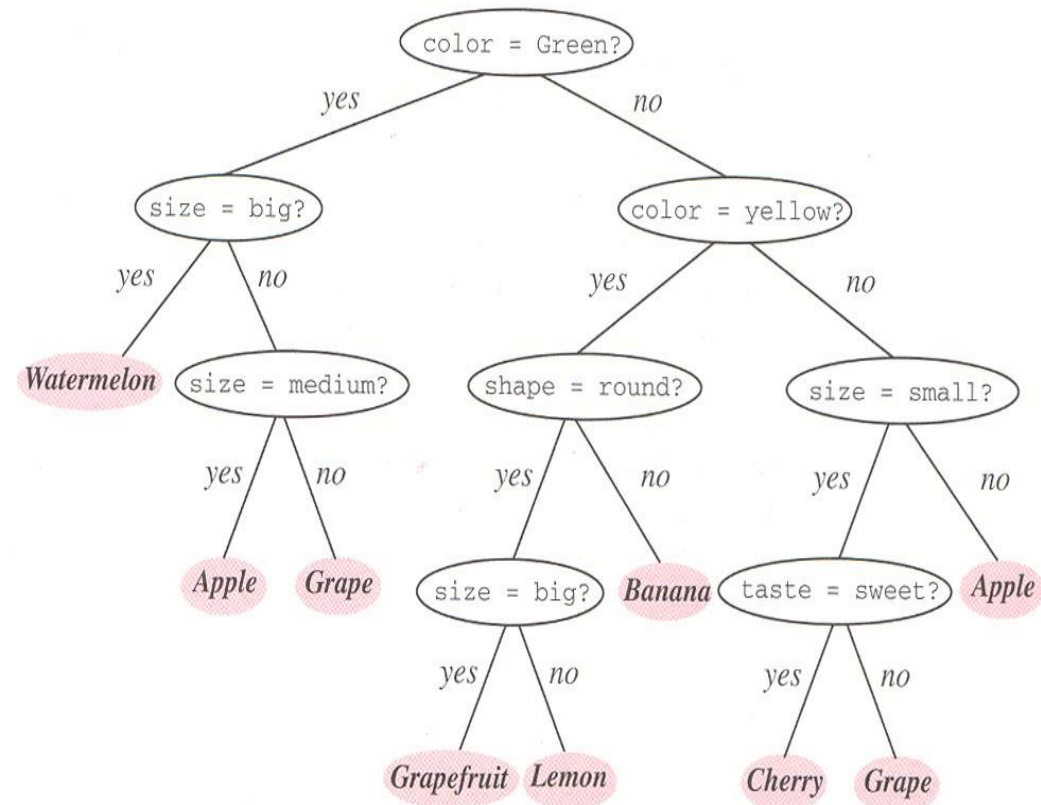
- ▶ $\text{Gini}(\text{rainy}) = 0.3936$
- ▶ $\text{Gini}(\text{sunny}) = 0.4571$
- ▶ $\text{Gini}(\text{overcast}) = 0.3571$

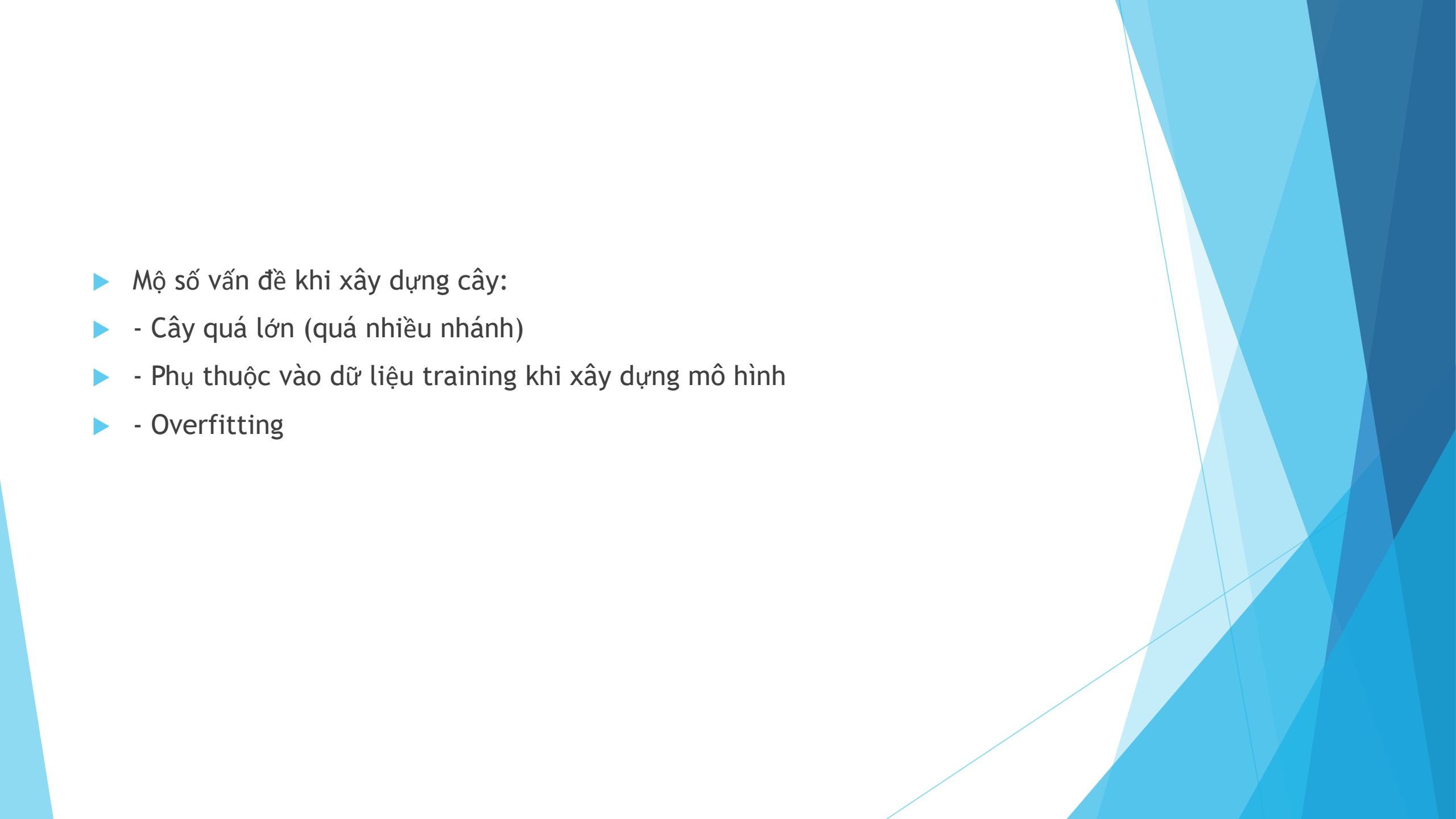


```
(IPdb [40]): print (r)
--- Outlook <= 0.50
|--- class: 1
--- Outlook > 0.50
|--- Humidity <= 0.50
|   |--- Outlook <= 1.50
|   |   |--- class: 0
|   |   |--- Outlook > 1.50
|   |       |--- Windy <= 0.50
|   |       |   |--- class: 1
|   |       |   |--- Windy > 0.50
|   |           |--- class: 0
|   |--- Humidity > 0.50
|   |   |--- Windy <= 0.50
|   |   |   |--- class: 1
|   |   |   |--- Windy > 0.50
|   |       |--- Outlook <= 1.50
|   |       |   |--- class: 1
|   |       |   |--- Outlook > 1.50
|   |           |--- class: 0
```

CART Algorithm

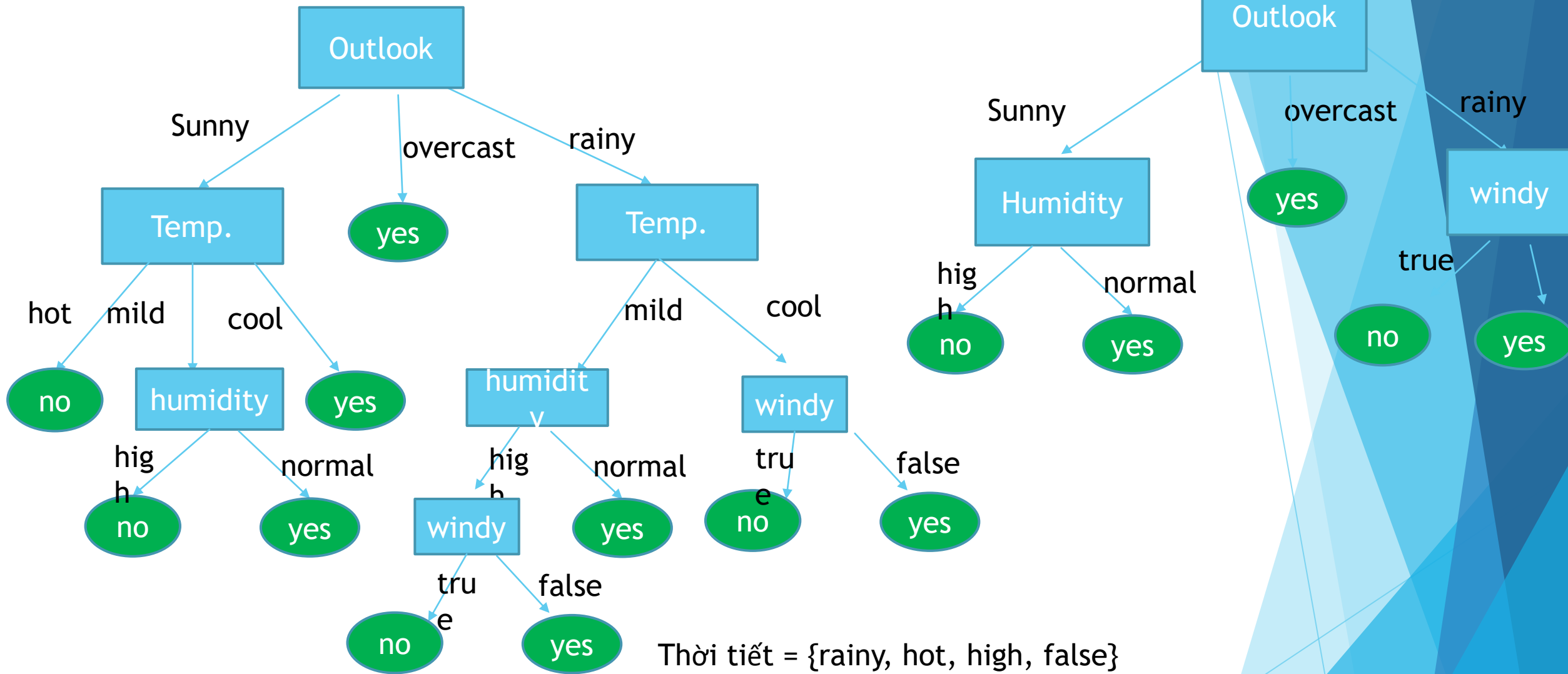
- ▶ Số lần chia
 - ▶ Chia theo nhánh
 - ▶ Cây nhị phân
- ▶ Thuộc tính được chọn
 - ▶ Đơn giản
 - ▶ Cây rút gọn đơn giản với vài node
 - ▶ Thuộc tính được chọn tạo các lớp con gần nhau nhất có thể.



- 
- ▶ Một số vấn đề khi xây dựng cây:
 - ▶ - Cây quá lớn (quá nhiều nhánh)
 - ▶ - Phụ thuộc vào dữ liệu training khi xây dựng mô hình
 - ▶ - Overfitting

Kết luận:

- ▶ Được sử dụng rộng rãi trong lĩnh vực khai thác dữ liệu
- ▶ Được phát triển trong các mô hình thống kê và học máy
- ▶ Được sử dụng để xây dựng các mô hình phân lớp, dự báo và hồi quy
- ▶ Điểm mạnh:
 - ▶ Dễ hiểu, dễ giải thích, dễ minh họa.
 - ▶ Dùng cho cả dữ liệu: Category, và dạng số
 - ▶ Không có tham số
- ▶ Điểm yếu:
 - ▶ Overfitting
 - ▶ High variance
 - ▶ Low bias



Outlook Temperature Humidity Windy

Ensemble methods

- ▶ Ensemble methods:
- ▶ Sampling
- ▶ Variance và bias
- ▶ Boostrops
- ▶ Bagging và Random forest
- ▶ Boosting
- ▶ Stacking

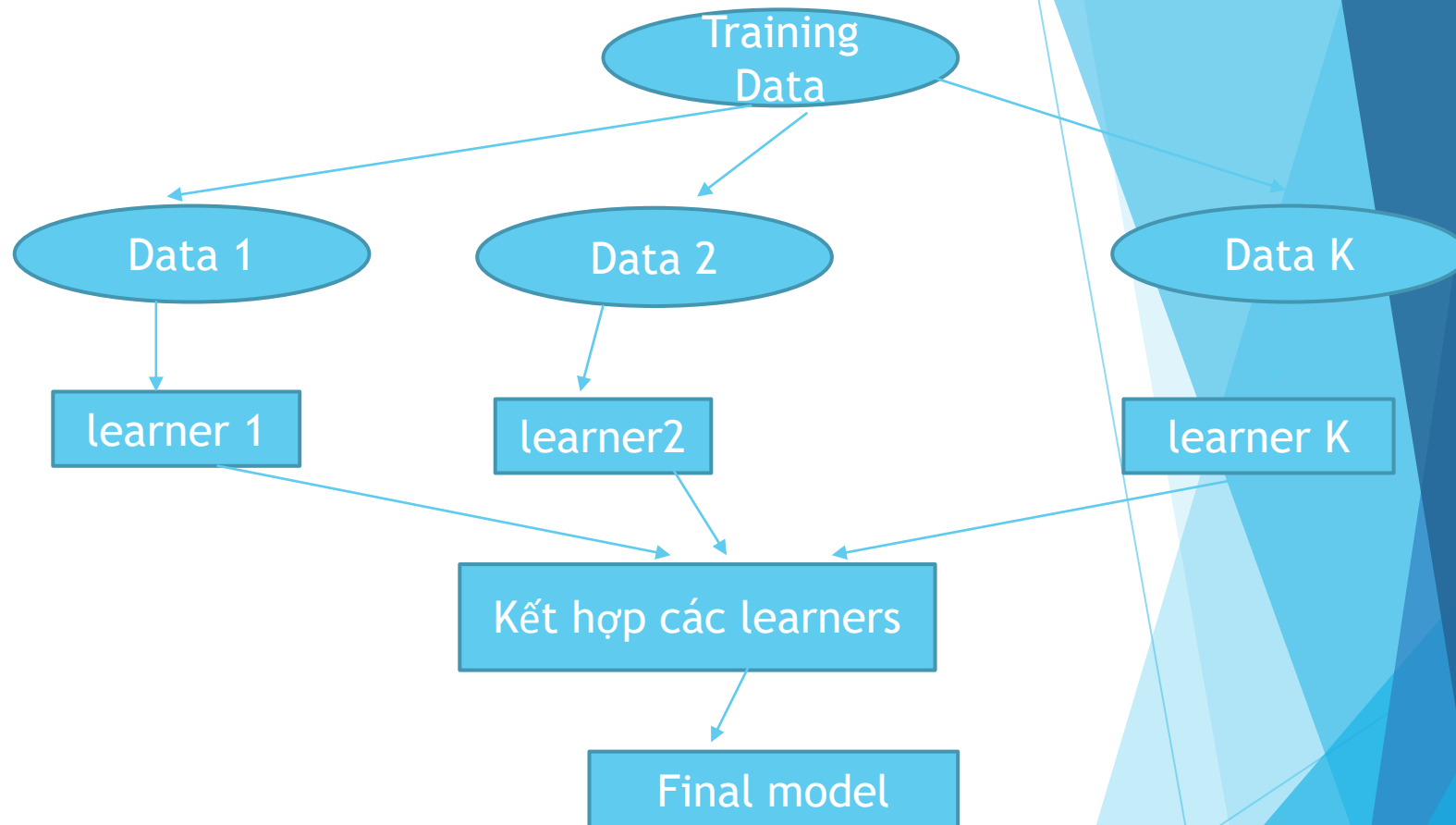
Ensembles

- ▶ Định nghĩa:
 - ▶ Một tập hợp các phương học máy (cùng loại hay khác loại), trong đấy mỗi phương pháp học máy có một kết quả riêng được kết hợp với nhau theo một số cách (bằng trọng số, bằng voting) để phân loại các lớp mẫu mới, gọi là phương pháp Ensemble methods.
- ▶ Các phương pháp Ensemble được tạo nên bởi các phương pháp học máy đơn lẻ và thông thường có độ chính xác cao hơn các phương pháp đấy.
- ▶ Ensembles dùng để giải quyết các vấn đề trong data mining như là Phân loại, hồi quy, phân nhóm.

Ensemble

Ý tưởng:

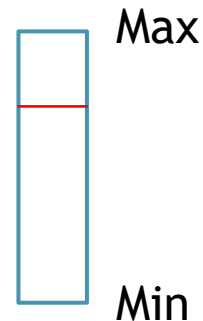
- ▶ Từ dữ liệu ban đầu:
 - ▶ Tạo thành K dataset
 - ▶ Mỗi Data dùng một phương pháp học máy
- ▶ Kết hợp các phương pháp học máy:
 - ▶ Kết hợp theo trọng số
 - ▶ Theo voting
- ▶ Final model



Ensembles

- ▶ Xuất phát từ ý tưởng:
- ▶ Ensemble method:
 - ▶ Chính xác hơn: Ensemble đưa ra dự đoán tốt hơn khi có dữ liệu mới
 - ▶ Đa dạng hơn: Ensemble đưa ra dự báo với phạm vi chính xác khá rộng.

Ví dụ: Độ chính xác sau 100 lần của một ensemble



- ▶ Theo lý thuyết: Marquis de Condorcet (1785)([Link](#)) về xác suất của các quyết định theo số đông.
- ▶ Giả sử: Các bộ học máy là độc lập
- ▶ Thì xác suất để phương pháp Ensemble dự đoán sai là:

$$\sum_{k \geq \lceil \frac{M+1}{2} \rceil} \binom{M}{k} \epsilon^k (1 - \epsilon)^{M-k}$$

Ensembles

- ▶ Ensembles đồng nhất: Mọi model (learner) trong Ensemble dùng chung một thuật toán, khác nhau về dataset
 - ▶ Từ một Training Data, nhân bản thành nhiều dataset, Data1, ..., Data k
 - ▶ Mỗi dataset dùng cho một learner
- ▶ Các phương pháp:
 - ▶ Bagging: Phương pháp lấy lại mẫu.
 - ▶ Boosting: Gán trọng số lại training data
- ▶ Ensemble hỗn tạp: Các model (hay learner) sử dụng một số thuật toán khác nhau.
 - ▶ Ví dụ: Stacking và Blending.

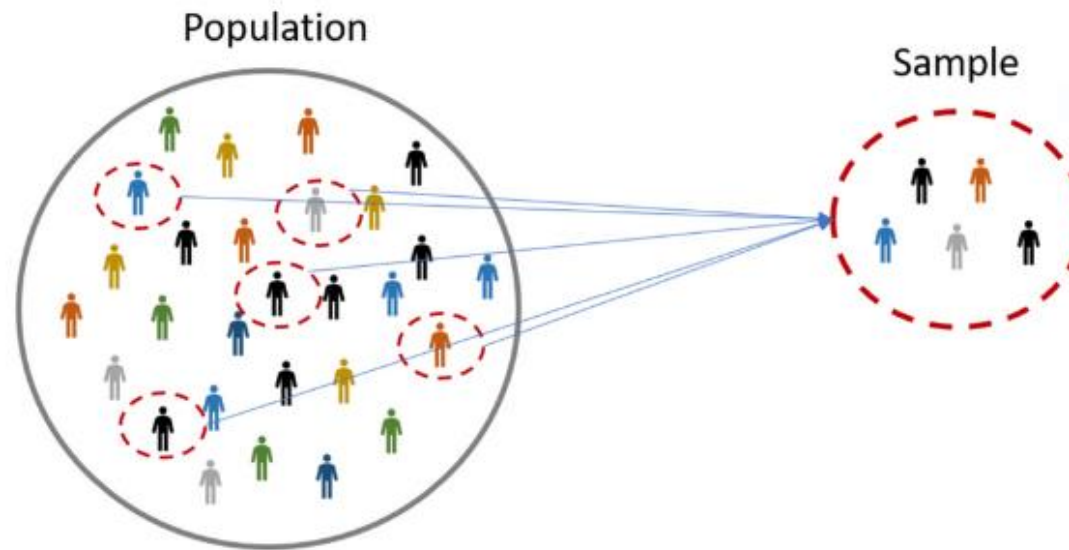
Các phương pháp tạo dữ liệu -lấy mẫu (sampling)

Sampling?

- ▶ Lấy mẫu là gì ???
- ▶ Là cách thức lấy mẫu từ tập dữ liệu gốc:
 - ▶ Lấy theo tỷ lệ
 - ▶ Stratified sampling
 - ▶ Lấy ngẫu nhiên
 - ▶ K-folds

Lợi ích và thách thức của Data Sampling

- ▶ Lấy mẫu rất hữu ích khi mà datasets là quá lớn để có thể phân tích đầy đủ.
 - ▶ Ví dụ: Phân tích dữ liệu lớn
- ▶ Tiết kiệm chi phí và thời gian
- ▶



Sampling framework

- ▶ **Mục tiêu lấy mẫu (Sample Goal).** Đặc tính của mẫu để chọn lựa.
- ▶ **Toàn bộ dữ liệu (Population).** Phạm vi hoặc vùng dữ liệu
- ▶ **Tiêu chí lựa chọn (Selection Criteria).** Phương pháp sẽ sử dụng để lựa chọn các samples trong việc lấy mẫu.
- ▶ **Kích thước của mẫu (Sample size).** Số lượng các quan sát sẽ lấy.
- ▶



▶ 1. Simple Random Sampling

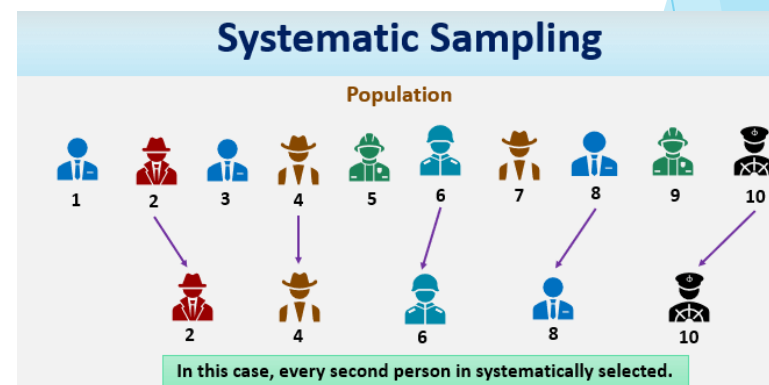
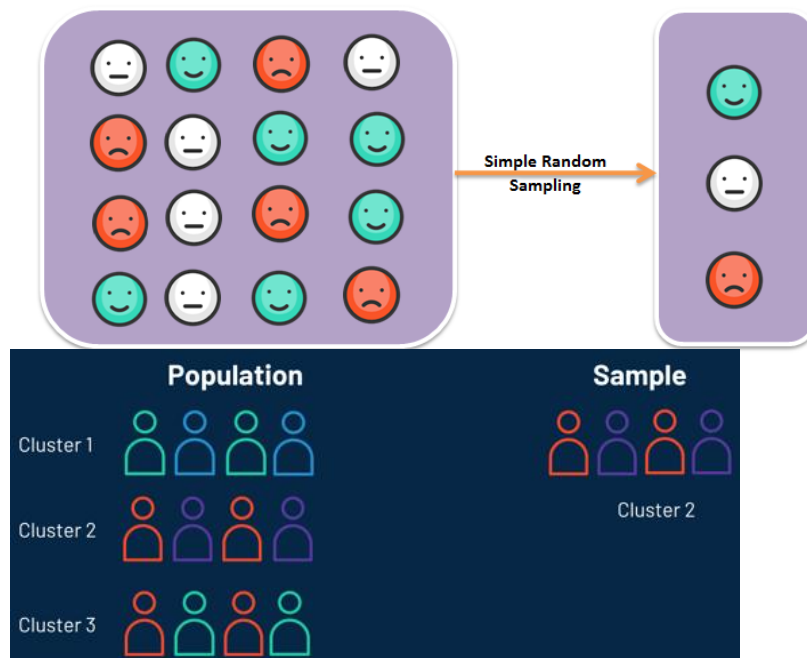
- ▶ Mỗi đại diện lấy một mẫu

▶ 2. Cluster Sampling

- ▶ Lấy mẫu theo nhóm/cụm
- ▶ Ví dụ: Tuổi, vị trí.

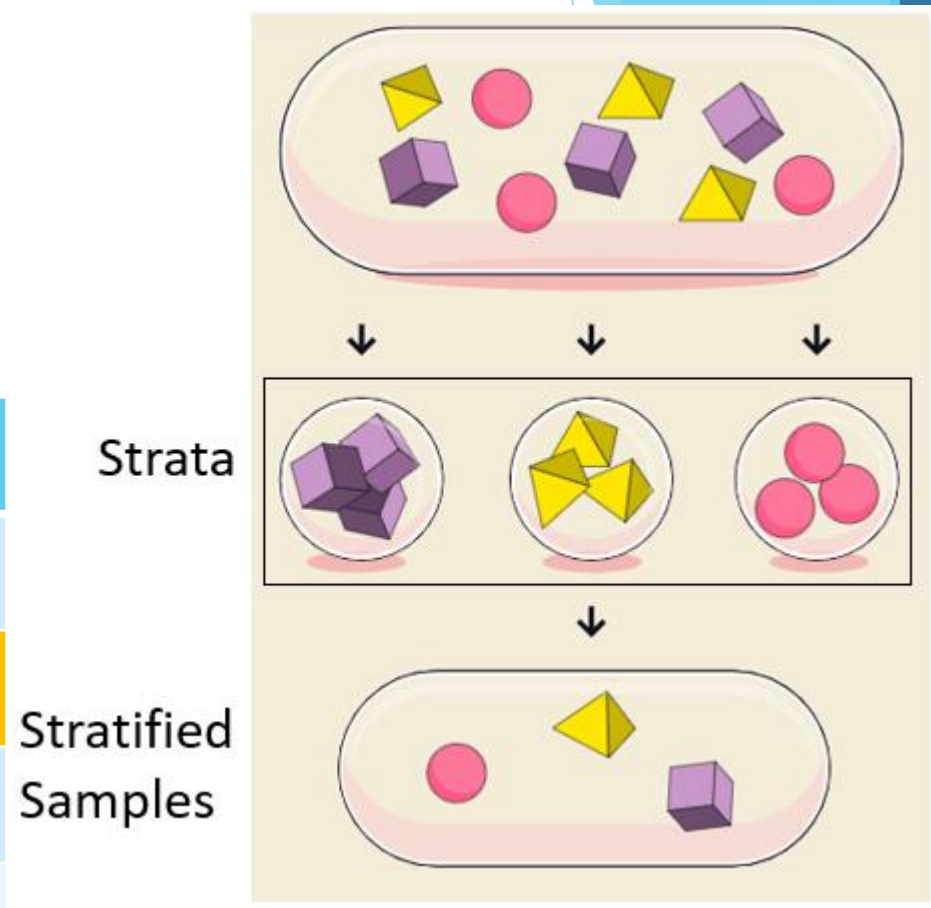
▶ 3. Systematic Sampling

- ▶ Lấy mẫu có hệ thống
- ▶ Có thể theo một thứ tự xuất hiện
 - ▶ Ví dụ: Cứ cách 2 vị trí, lấy một mẫu.



► Stratified random Sampling

- Chia thành các nhóm cụ thể
- Mỗi nhóm lấy một hoặc nhiều mẫu.

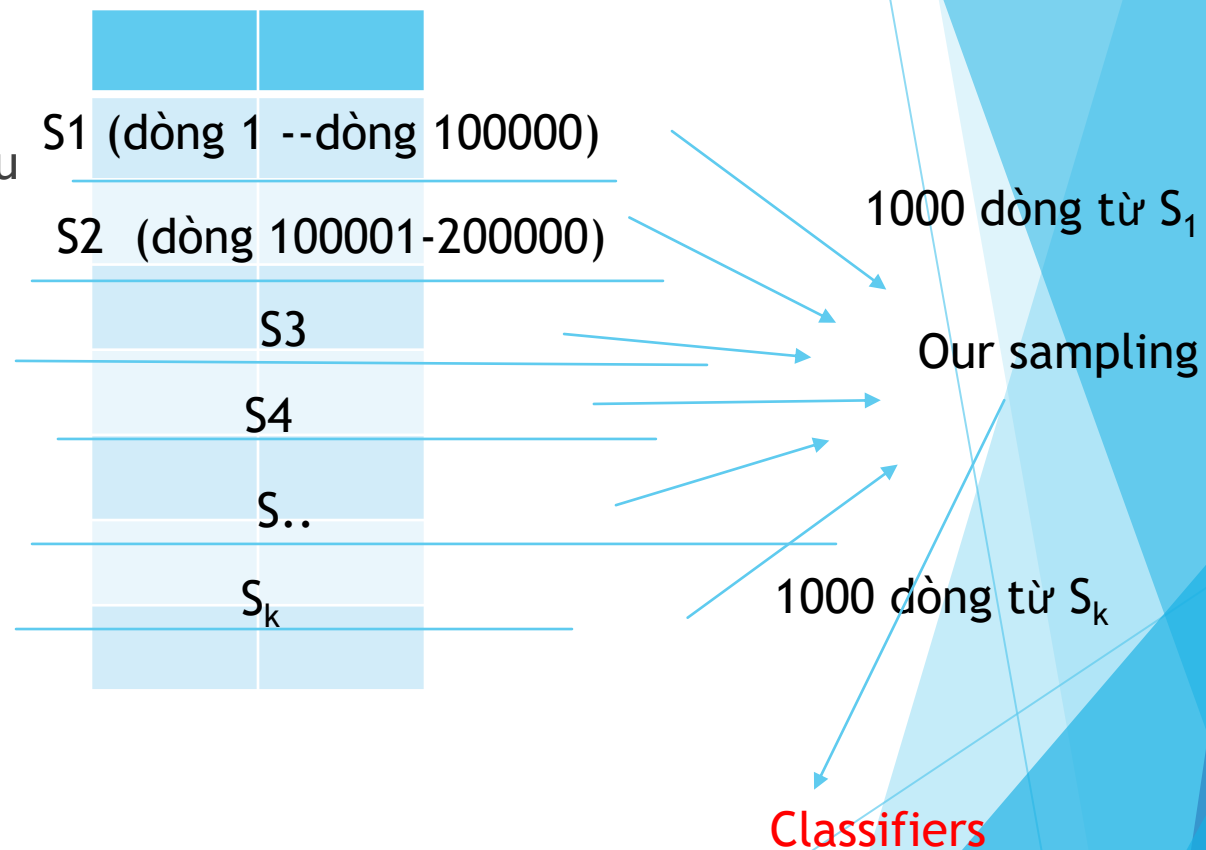


- ▶ Stratified sampling:
- ▶ $N = 14$;
- ▶ Tạo một dataset = 5;
- ▶ Bao nhiêu Yes và No?
- ▶ $(YES + No) = 5$;
- ▶ $YES = 9/14 = p(YES)$;
- ▶ $No = 5/14 = p(NO)$;
- ▶ $YES(D1) = p(YES) * 5$;
- ▶ $NO(D1) = p(NO) * 5$;
- ▶ D1 đảm bảo?
- ▶ Chạy 80-00 lần

	Outlook	Temperature	Humidity	Windy	Play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
5	rainy	cool	normal	FALSE	yes
7	overcast	cool	normal	TRUE	yes
13	overcast	hot	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

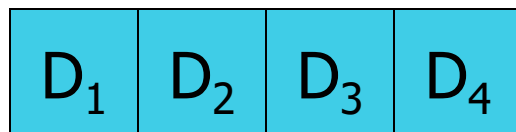


- ▶ % các lớp đích (class) trong toàn bộ dữ liệu
- ▶ % lớp 0 + % lớp 1 = 100% số lượng mẫu
- ▶ $N + M = 100\%$ số lượng mẫu S ;
- ▶ Giả sử $S < 1000$; $S < 5000$;
- ▶ $S = 11\ 000\ 000$;
- ▶ $S = S_1 + S_2 + S_3 + S_4 + S_5 + S_6 + \dots + S_k$;
- ▶ $S_1 = N_1(0) + M_1(1)$
- ▶ Ví dụ $S = 10000$;

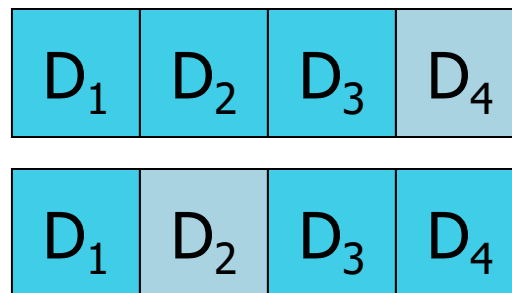
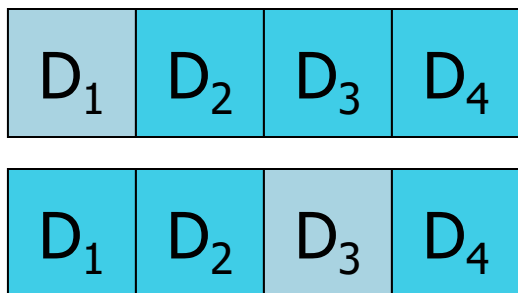


Cross-Validation

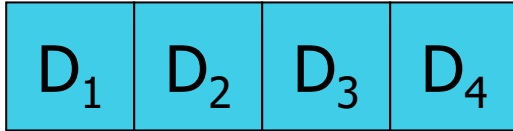
- ▶ K-Fold cross-validation chia dữ liệu D thành k phần bằng nhau. Ví dụ: D_1 - D_4



- ▶ Thuật toán sẽ kiểm thử k lần, mỗi lần training set sẽ là $D \setminus D_i$ và testing là D_i



- ▶ Trường hợp đặc biệt là: leave-one-out (chỉ test trên 1 sample)



D1 : Testing

Training set →
build model

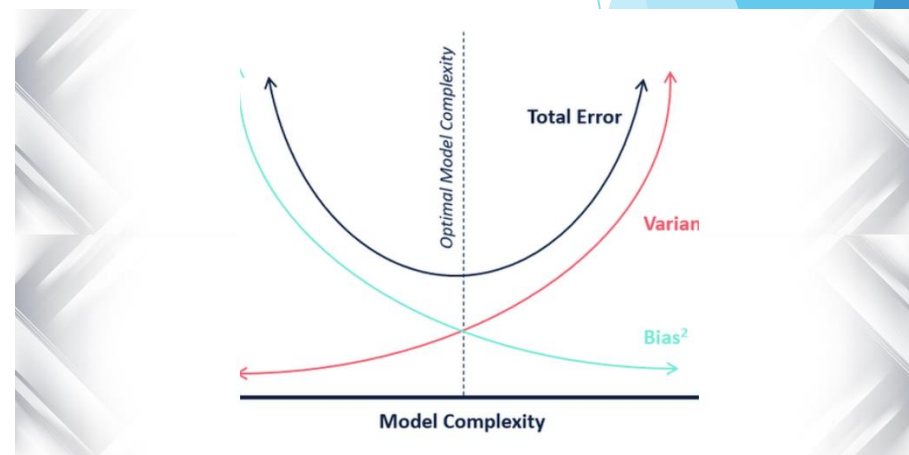
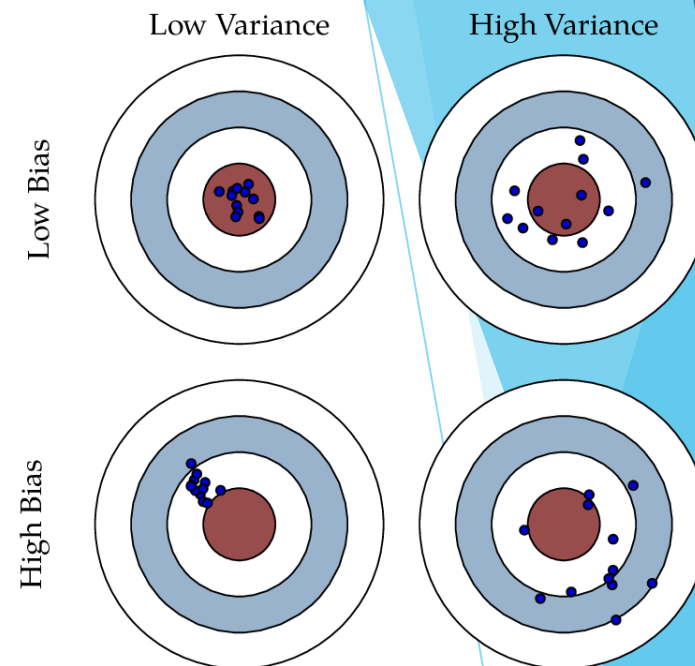
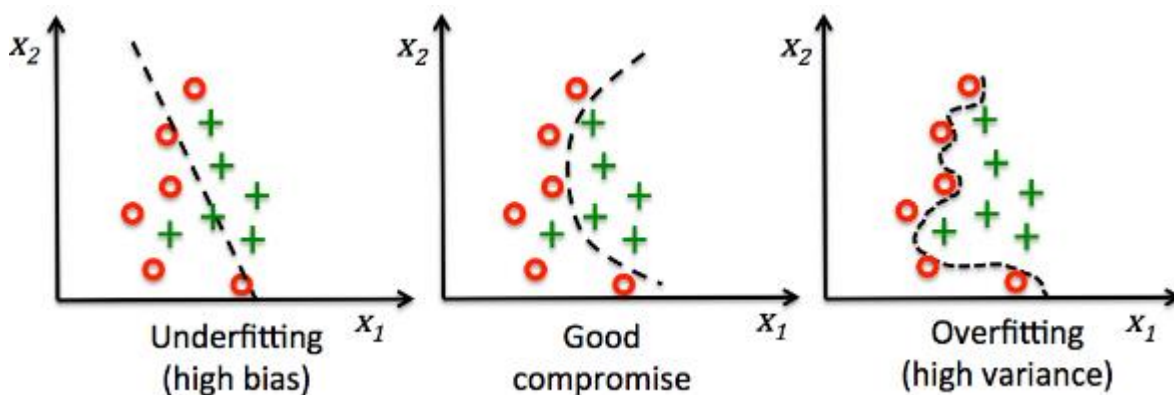
	Outlook	Temperature	Humidity	Windy	Play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	YES

Stratified Cross-Validation

- ▶ Đây là trường hợp đặc biệt của stratified sampling.
 - ▶ Các folds sẽ có tỷ lệ phân trăm của lớp đích (classes) tương tự như dữ liệu gốc.

Variance và Bias

- ▶ Hiệu quả các mô hình
- ▶ Bias : thành phần lỗi độc lập với mẫu dữ liệu học
- ▶ Variance : thành phần lỗi do biến động liên quan đến sự ngẫu nhiên của tập học

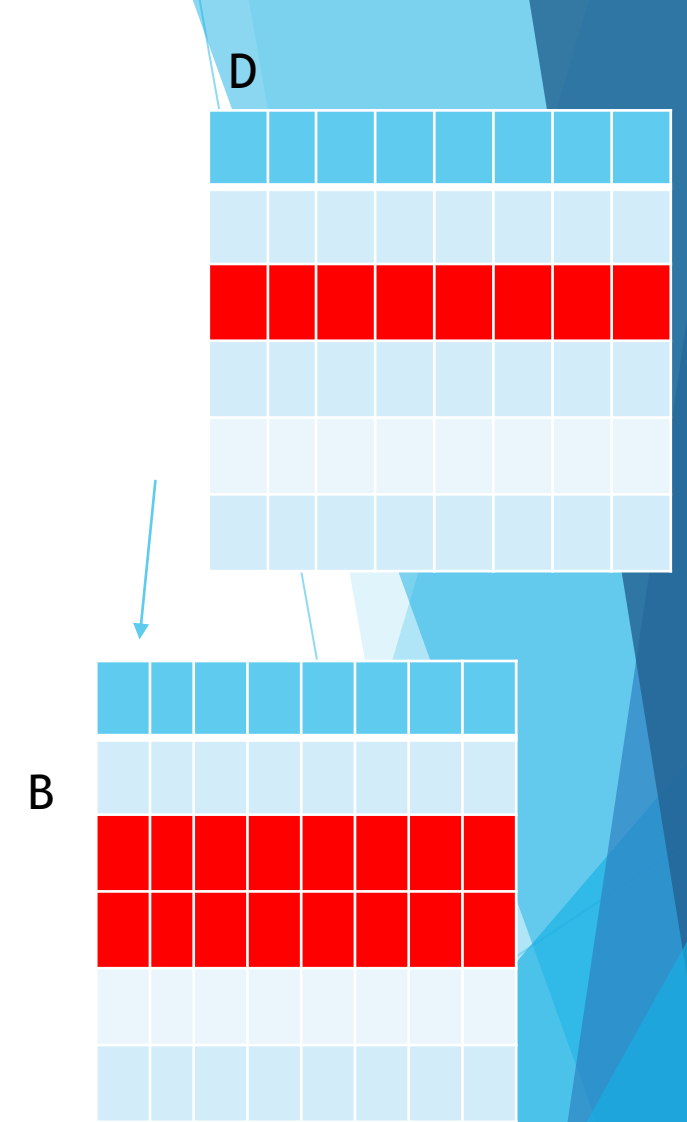


Bootstraps

- ▶ Là một sample của dữ liệu gốc D có lặp lại:
- ▶ Mỗi bootstraps khi lấy từ dữ liệu gốc: chứa khoảng 63.2% số dòng (samples) là không lặp lại, còn lại số dòng (samples) bị lặp lại.

$$(1 - 1/n)^n = e^{-1} = 0.632$$

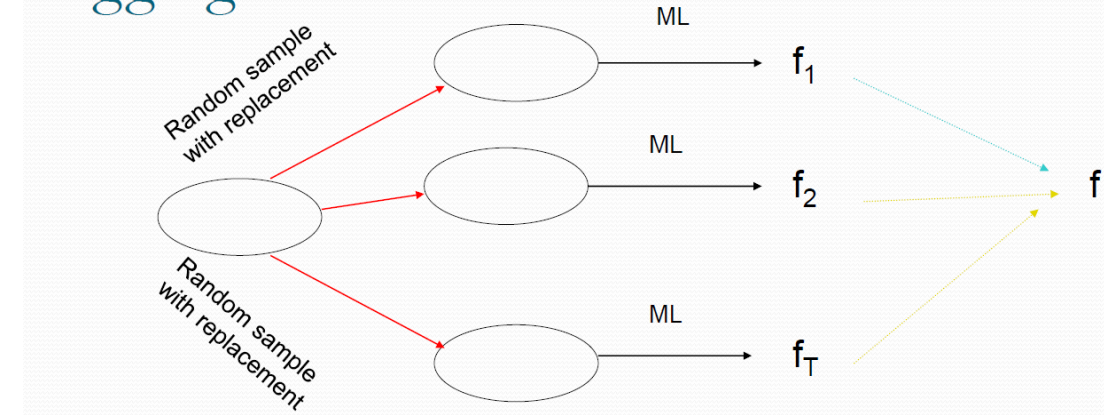
- ▶ Bootstrap sample B^i được dùng cho training và phần còn lại của dataset dùng cho testing:
 - ▶ $D \setminus B^i$



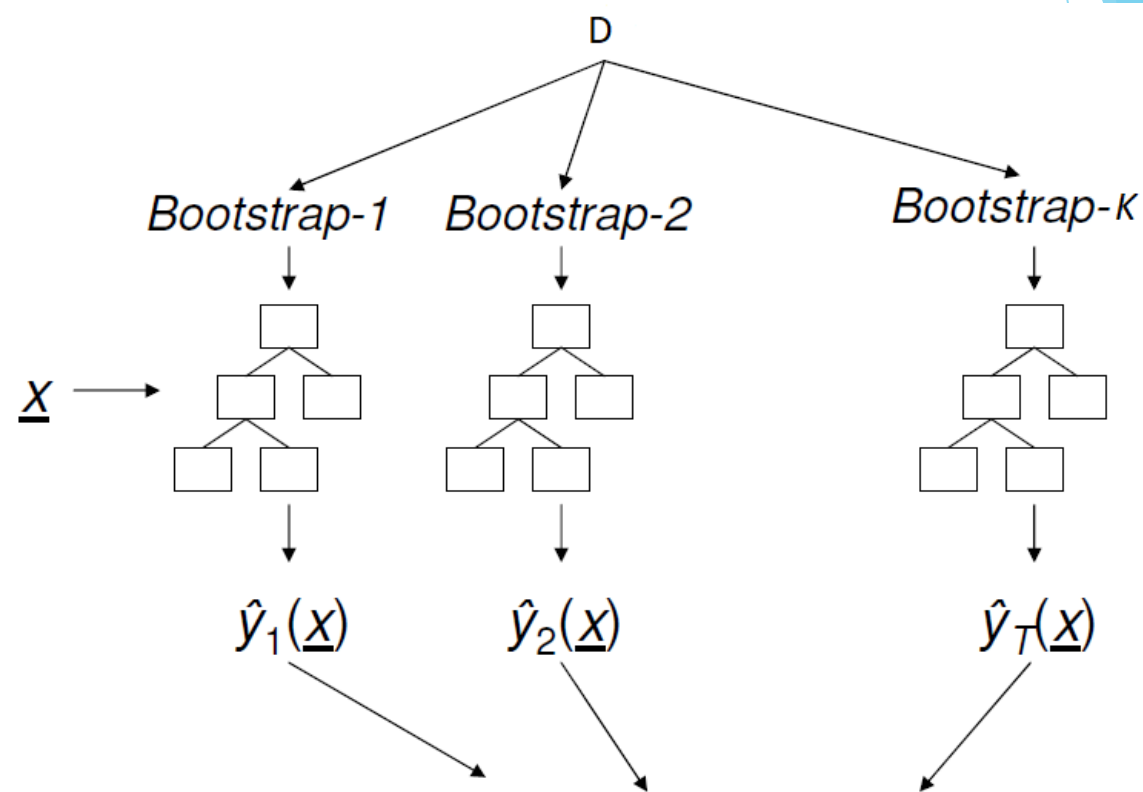
Bagging

- ▶ Bagging(bootstrap aggregation) (Brieman, 1996)
- ▶ Tư tưởng:
 - ▶ Tạo một tập ensemble bằng cách lặp lại cách lấy mẫu ngẫu nhiên trên một tập training data.
- ▶ Thuật toán:
 - ▶ Cho trước một tập dữ liệu D
 - ▶ Tạo K bootstraps (B) (resample có lặp lại) cũng kích cỡ với dữ liệu training.
 - ▶ Phần dữ liệu không được chọn $D \setminus B^i$ sẽ được chọn làm testingⁱ
 - ▶ Ứng với bootstrap B^i xây dựng một mô hình phân loại (hồi quy).
- ▶ Kết hợp K model lại và dùng voting để phân loại
- ▶ Hồi quy thì chia trung bình

Bagging



- ▶ Giảm được Variance;



hồi quy : $\hat{y}(\underline{x}) = (\hat{y}_1(\underline{x}) + \hat{y}_2(\underline{x}) + \dots + \hat{y}_T(\underline{x})) / \kappa$

phân loại : $\hat{y}(\underline{x}) = \text{bình chọn số đông } \{\hat{y}_1(\underline{x}), \dots, \hat{y}_T(\underline{x})\}$

Ví dụ:

Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

- ▶ $D = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$
- ▶ Bootstrap $B^1: \{7, 8, 10, 8, 2, 5, 10, 10, 5, 9\}$
- ▶ Bootstrap $B^2: \{1, 4, 9, 1, 2, 3, 2, 7, 3, 2\}$
- ▶ Bootstrap $B^3: \{1, 8, 5, 10, 5, 5, 9, 6, 3, 7\}$

- ▶ Dùng để testing $B^1 : \{\}$?????
- ▶ Dùng để testing $B^2 : \{, \}$
- ▶ Dùng để testing $B^3 : \{, \}$

- ▶ Bagging (bootstrap aggregation) (Brieman, 1996)
- ▶ Tư tưởng:
 - ▶ Tạo một tập ensemble bằng cách lặp lại cách lấy mẫu ngẫu nhiên trên một tập training data.
- ▶ Thuật toán:
 - ▶ Cho trước một tập dữ liệu D
 - ▶ Tạo K bootstraps (B) (resample có lặp lại) cũng kích cỡ với dữ liệu training.
 - ▶ Phần dữ liệu không được chọn $D \setminus B^i$ sẽ được chọn làm testingⁱ
 - ▶ Ứng với bootstrap B^i xây dựng một mô hình phân loại (hồi quy).
- ▶ Kết hợp K model lại và dùng voting để phân loại

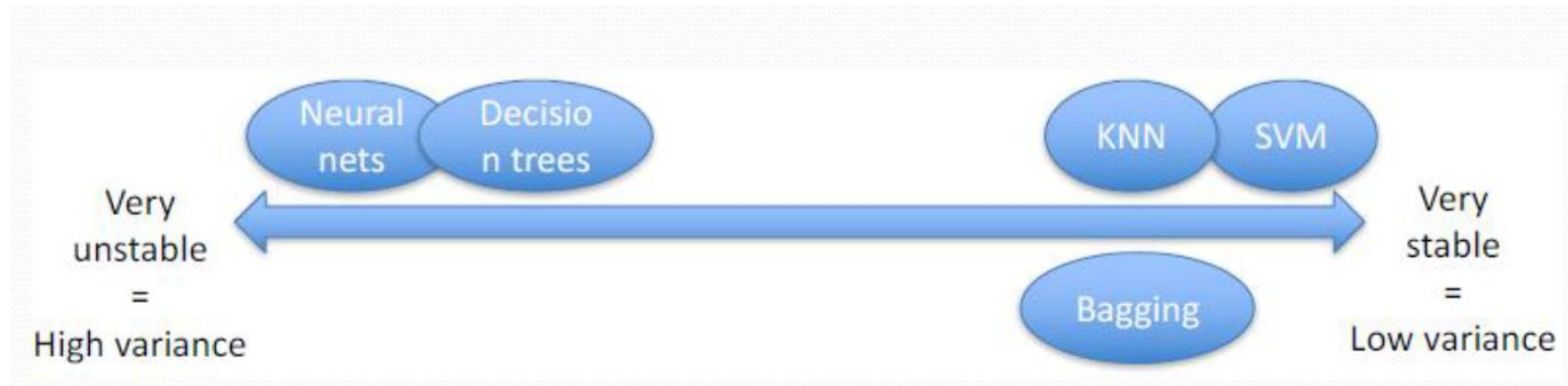
Bag

Bagging

- ▶ Điểm mạnh
 - ▶ Hiệu quả khi bộ phân lớp là không ổn định
 - ▶ Tăng độ chính xác bởi vì giảm được Variance của các bộ phân lớp đơn
 - ▶ Không phụ thuộc vào bất kỳ một mẫu (một dòng) nào của dữ liệu training
 - ▶ Giảm được vấn đề overfitting

Bagging

- ▶ Điểm bất lợi của Bagging
 - ▶ Nếu các bộ học là ổn định thì Bagging đưa ra kết quả không quá khác biệt



Rừng ngẫu nhiên - Random forest



#236946514

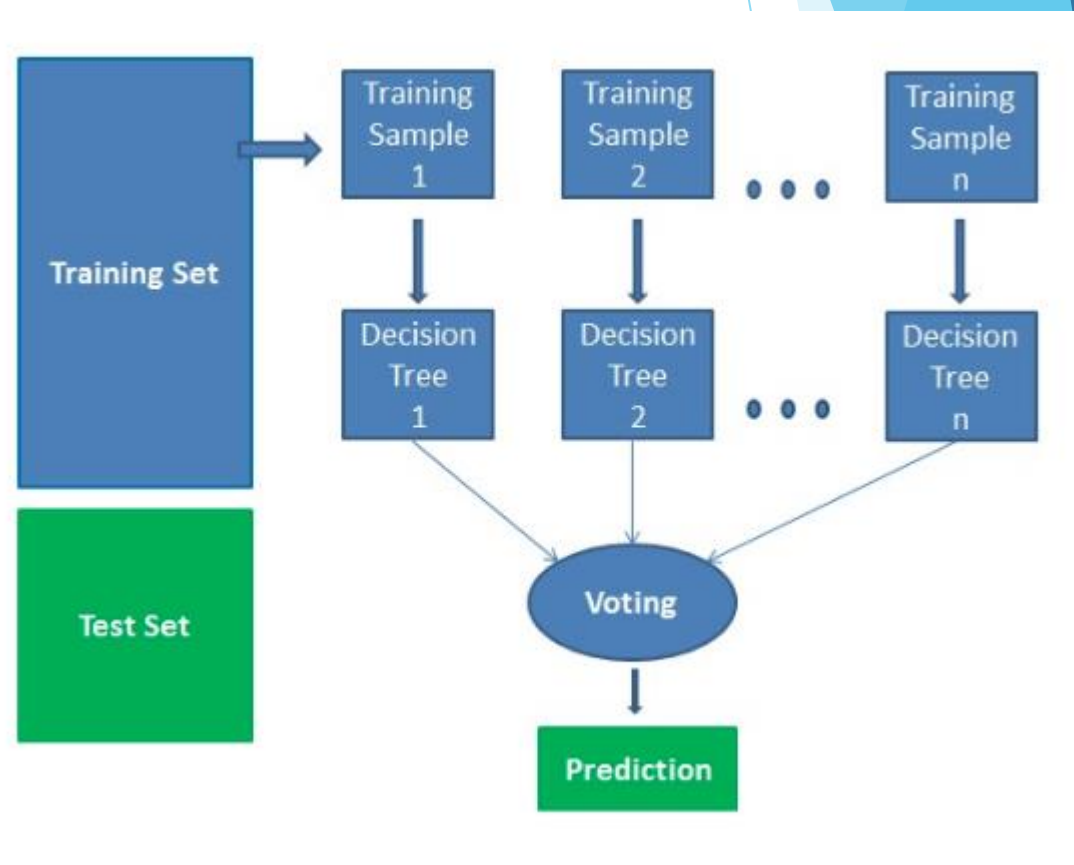
Rừng ngẫu nhiên - Random forest

- Beriman(2001): đưa ra thuật toán random forest.

Ý tưởng:

Bước 1: Từ tập dữ liệu ban đầu, dùng phương pháp lấy **mẫu bagging** tạo thành một K tập dữ liệu con.

Bước 2: Với mỗi dữ liệu con tạo một cây quyết định. Tại mỗi nốt chia, chọn **ngẫu nhiên** số lượng **x** feature ($x = \sqrt{m}$), m là số lượng các đặc trưng). Với K cây quyết định thì tạo thành một Rừng.



Boosting

▶ Lịch sử:

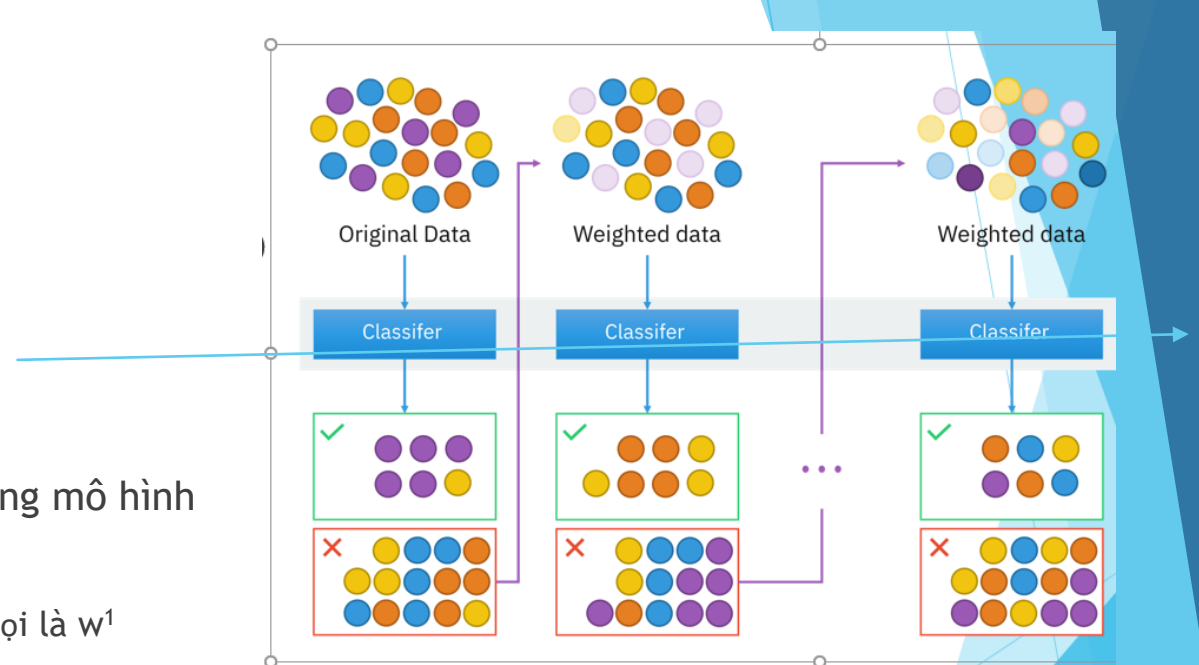
- ▶ Xuất phát từ câu hỏi: "Can a set of weak learners create a single strong learner?" của [Kearns](#) and [Valiant](#) (1988, 1989)
 - ▶ Một tập học các learners yếu có thể tạo nên một learner đơn lẻ hiệu năng?
- ▶ [Robert Schapire](#) đã trả lời là [yes](#): Và đưa ra khái niệm là boosting
- ▶ Freund and Schapire (1996) phát triển thuật toán Adaboost

▶ Ý tưởng:

- ▶ Tương tự như Bagging, tạo ra một loạt các learners, tuy nhiên sự khác biệt là hoạt động theo sequence, theo chuỗi. Bagging các learner làm việc song song.

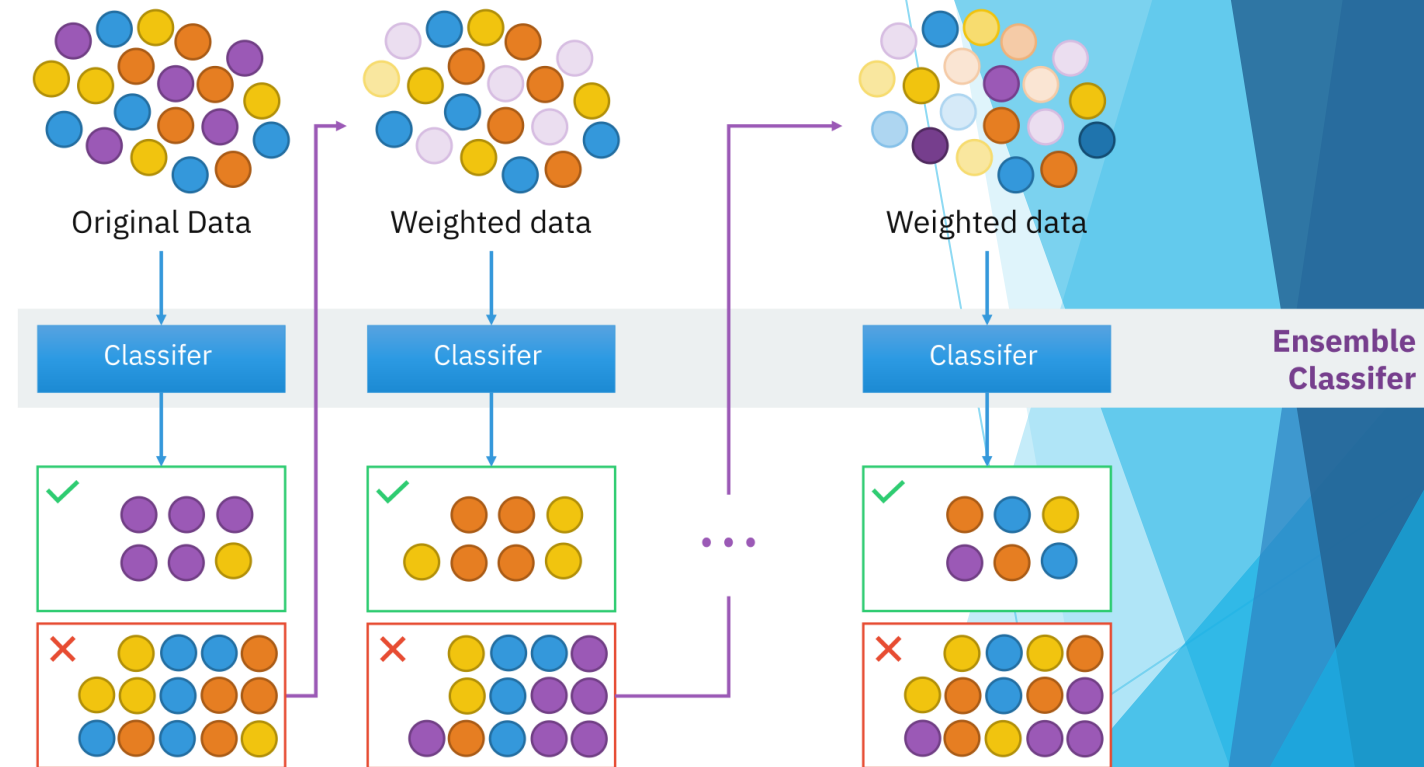
Boosting

- ▶ Cho bộ dữ liệu Training set D, và testing T
- ▶ Bước 1:
 - ▶ Từ dữ liệu Dataset D, tạo một tập dữ liệu B^1
- ▶ Bước 2:
 - ▶ Cho B^1 vào một learner (decision tree) để xây dựng mô hình
 - ▶ Kiểm tra mô hình bằng chính dữ liệu D
 - ▶ Xác định được chính xác của dữ liệu của mô hình: gọi là w^1
 - ▶ Xác định được những mẫu (sample) chưa được phân loại chính xác
- ▶ Bước 3: Tạo tập dữ liệu mới B^2 bằng cách: Những mẫu chưa chính xác ở mô hình trước + sample dữ liệu từ D; sao cho kích thước của $B^2 = D$ (Ví dụ đều là 100 mẫu)
 - ▶ Cho B^2 vào để xây dựng mô hình
 - ▶ Kiểm tra độ chính xác bằng dữ liệu D
 - ▶ Xác định được chính xác của dữ liệu của mô hình: gọi là w^2
 - ▶ Xác định được những mẫu (sample) chưa được phân loại chính xác
- ▶ Bước K: Lặp lại tiến trình đến khi tạo ra K models, hoặc độ sai số của mô hình thứ K và K-1 là ổn định thì dừng.



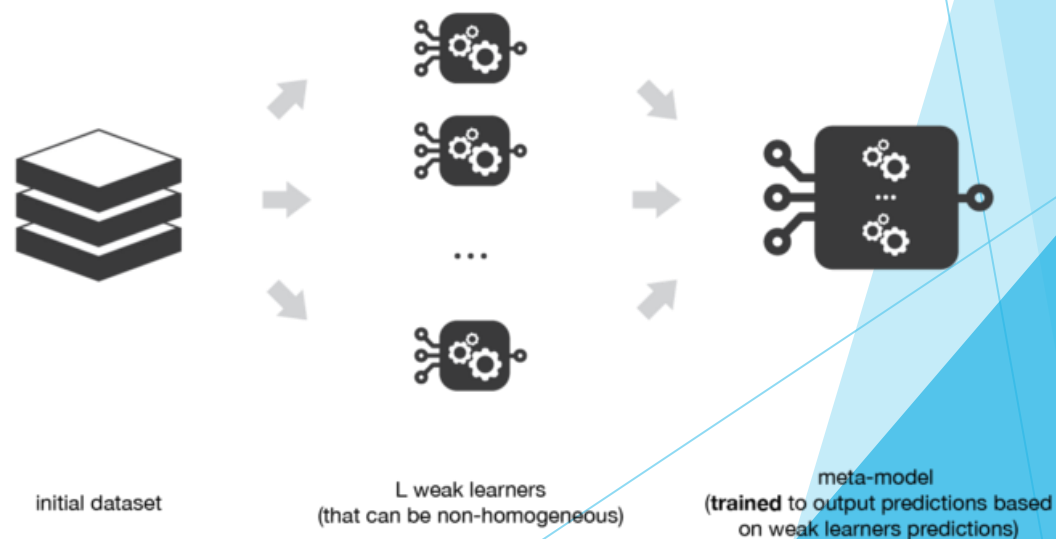
Boosting

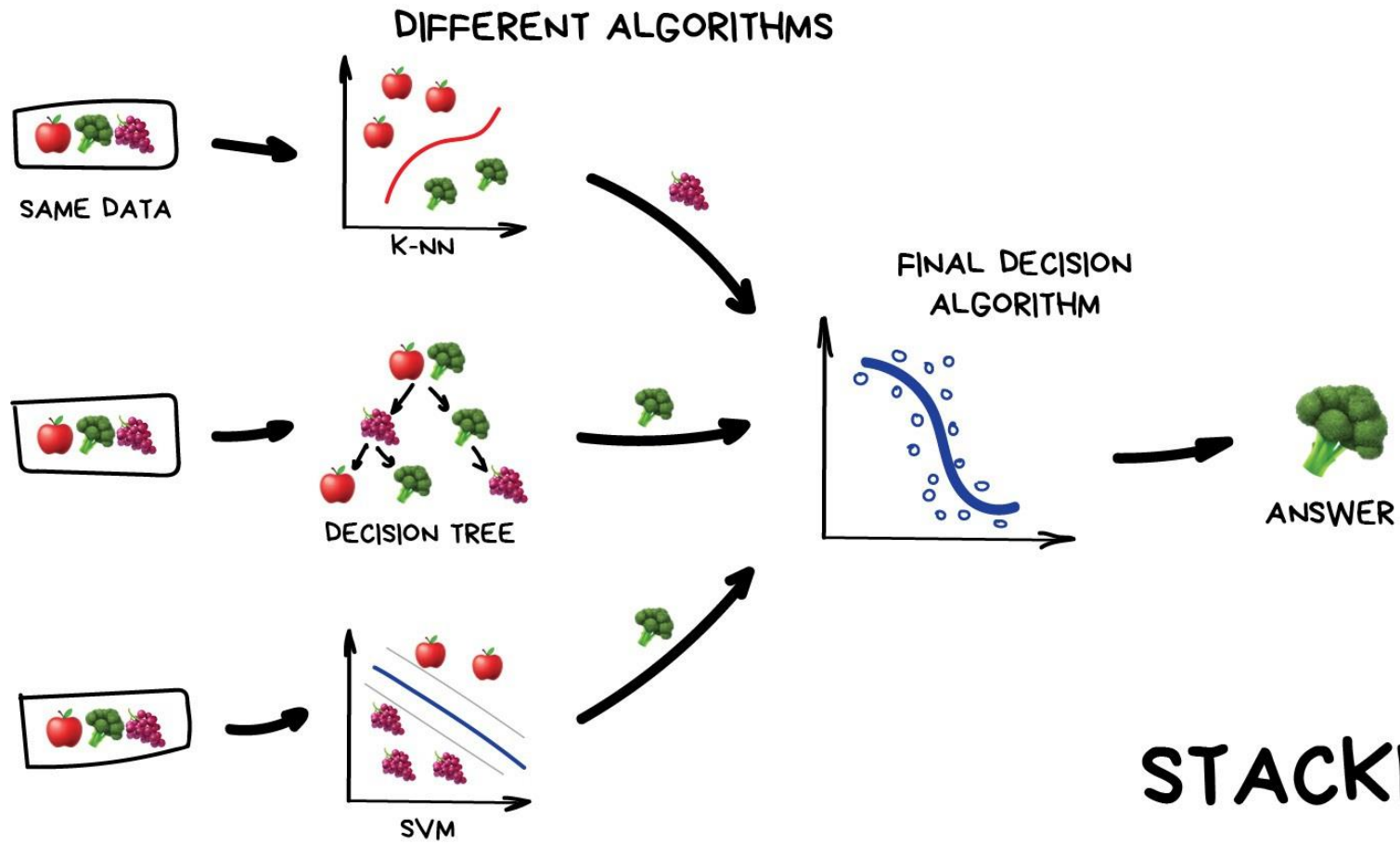
- ▶ Với dữ liệu mới để kiểm tra mô hình:
- ▶ Bằng voting
- ▶ $(\text{class}) = \{w^1(\text{model1}), w^2(\text{model2}), \dots, w^3(\text{model } k)\}$



Stacking

- ▶ Khác với Bagging, Boosting, stacking dùng nhiều mô hình trên một tập dataset
- ▶ Ý tưởng:
 - ▶ Với một dataset dùng nhiều mô hình để dự đoán
 - ▶ Các mô hình này gọi là meta-model
 - ▶ Dữ liệu mới vào sẽ qua meta-model này đưa ra kết quả: Chọn theo voting hoặc model đơn nào đúng nhất





STACKING

Biến thể của stacking



▶ 1,2,4,6,7,8,8,**8**,8,8,8,9,10,11,12

▶ 1,2,4,**6**,7,8,8

▶ 1, 5,6,7, 10

▶ 8,8,8,**9**,10,11,12