
2102470 Học máy

Bài giảng: 2.3 Bài toán phân lớp

Chương 2: Xấp xỉ và phân lớp

Ôn lại bài học trước

- Bạn có nhớ ? % ?

Nội dung chính

- 2.3.1 Bài toán phân lớp
- 2.3.2 Hàm mục tiêu
- 2.3.3 K-nearest neighbor
- 2.3.4 Support vector machine
- **2.3.5 Decision tree**
- **2.3.6 Ví dụ về bài toán phân lớp**

2.3 Bài toán phân lớp

2.3.5 Decision tree

- Chơi trò chơi Đoán số

Trang chủ » Lớp 3 » Tin học

CÂU HỎI:

🕒 12/07/2024 👁 929

Trò chơi Đoán số

Hai bạn An và Khoa cùng chơi trò đoán số. Bạn An nghĩ ra một con số trong khoảng từ 1 đến 100. An viết con số này ra mảnh giấy và cất đi, không để Khoa nhìn thấy. An yêu cầu Khoa đoán xem con số An đã viết ra là số nào. Khi Khoa đưa ra một con số dự đoán, An sẽ trả lời Khoa là “Quá thấp”, “Quá cao” hoặc “Hoan hô! Bạn đã đoán đúng” tùy vào kết quả so sánh con số mà Khoa đưa ra với con số mà An đã viết.

Mẫu: Nếu con số Khoa dự đoán lớn hơn con số An nghĩ thì An nói “Quá cao”.

b) Điều kiện để An nói “Hoan hô! Bạn đã đoán đúng” là gì?

Một cách tiếp cận khác cho việc giải quyết bài toán phân lớp?

<https://khoahoc.vietjack.com/question/1032136/tro-choi-doan-so-hai-ban-an-va-khoa-cung-choi-tro-doan-so-ban-an-nghi-ra-mot-con-so-trong-khoang-tu-pqdil>

2.3.5 Decision tree

- Không dựa trên loại hàm dự đoán trước
- Học từ các câu hỏi if/else phân cấp, từ đó đưa ra quyết định
- Cây quyết định gồm các node và các nhánh
 - Node
 - Nút gốc, nút con, nút lá
 - Biểu diễn một câu hỏi/**thuộc tính** cần quan tâm
 - Nút cuối (nút lá), kết thúc (không phân chia thêm), chứa câu trả lời (nhãn lớp hoặc giá trị)
 - Nhánh

Decision tree

- Quyết định
 - Duyệt đường đi từ nút gốc đến một nút lá
 - Nhãn lớp gắn với nút lá được dùng cho việc phân lớp

- Chú ý: cây hồi quy khi các nút lá chứa các giá trị liên tục

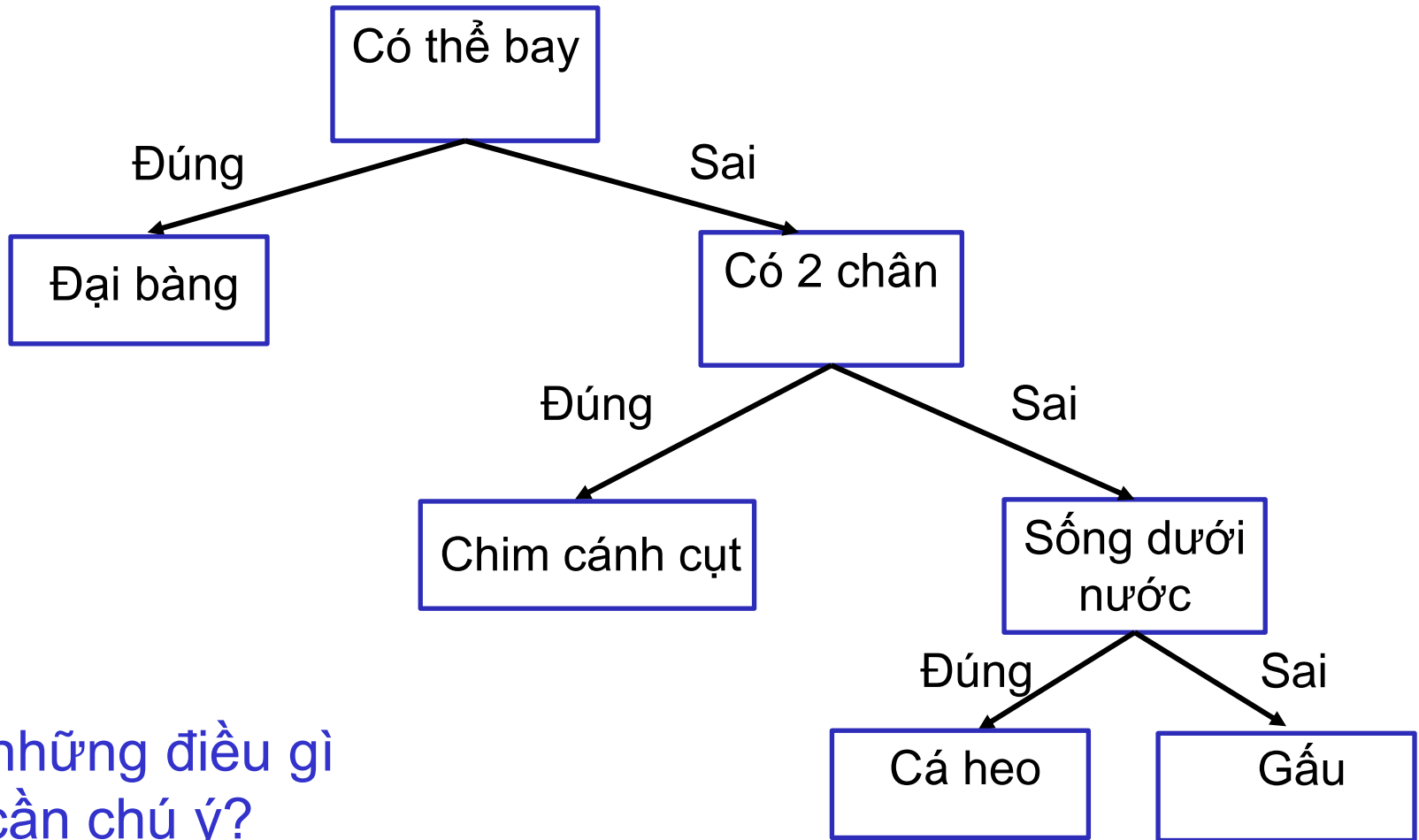
Ví dụ

Bộ dữ liệu, trong đó gồm 4 loài động vật: đại bàng, chim cánh cụt, cá heo, gấu

Xây dựng 1 cây quyết định để phân biệt được 4 lớp động vật, sử dụng 3 đặc trưng.

Ví dụ

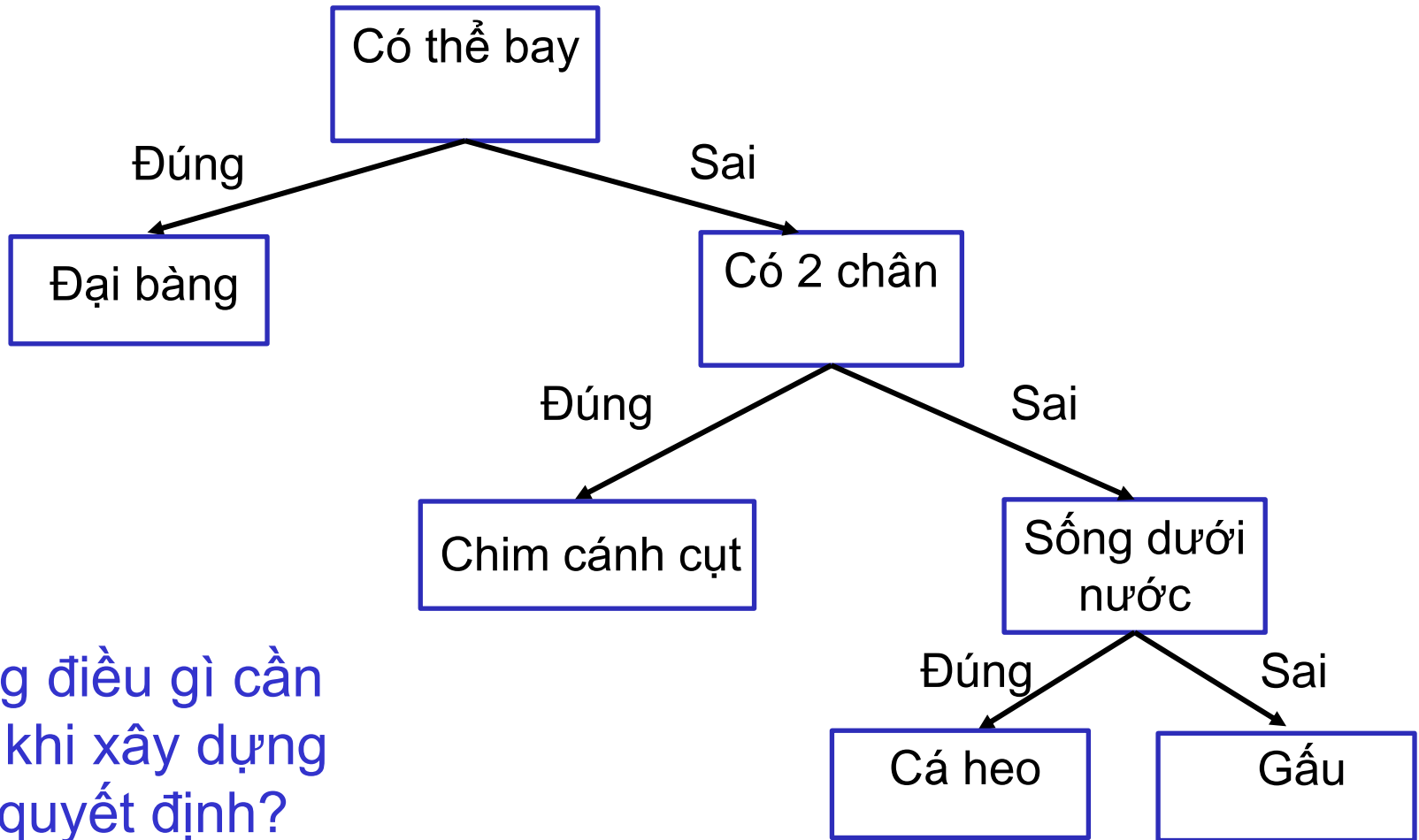
Sử dụng 3 đặc trưng: có thể bay, có 2 chân, sống dưới nước



Có những điều gì cần chú ý?

Ví dụ

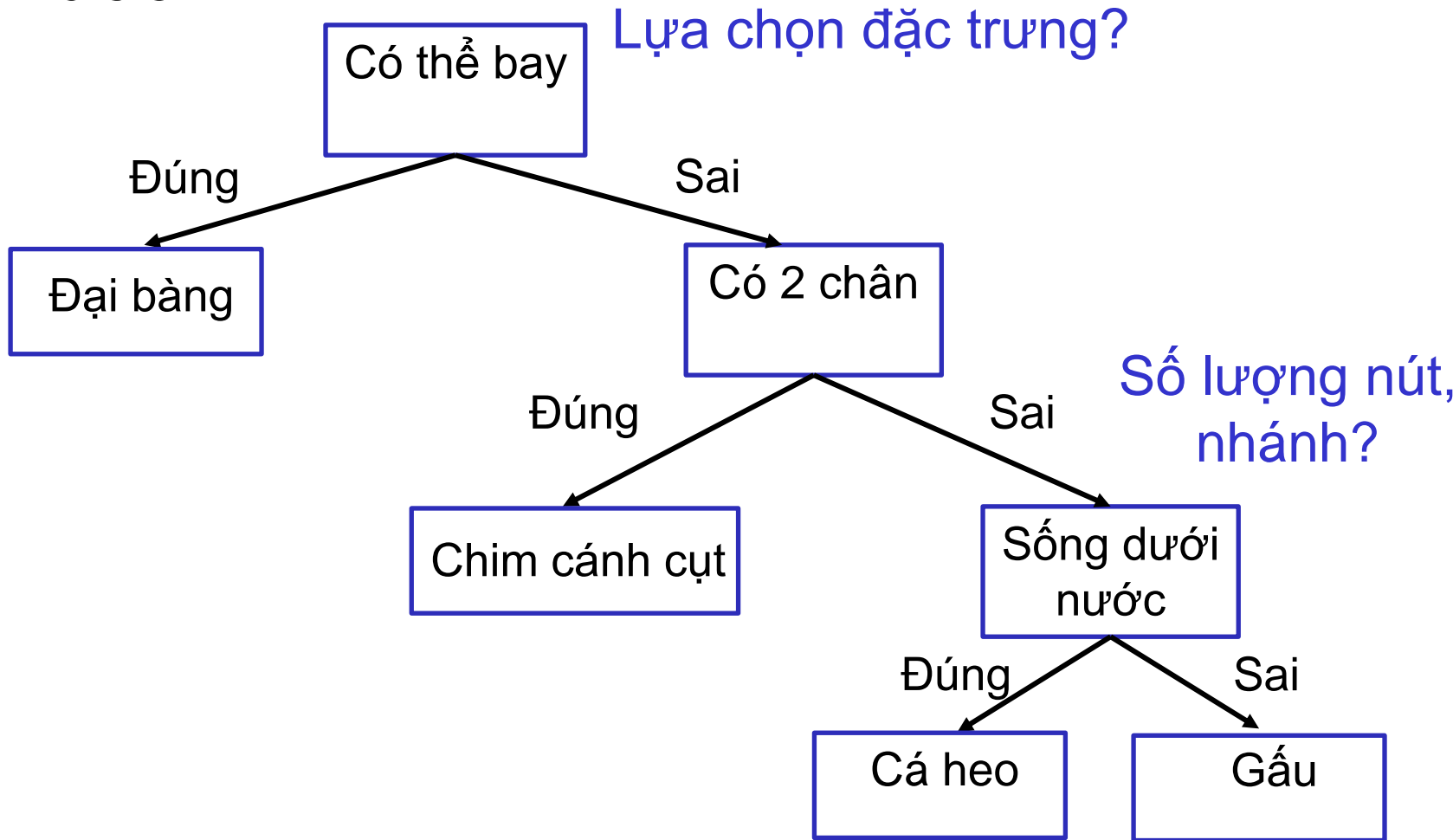
Sử dụng 3 đặc trưng: có thể bay, có 2 chân, sống dưới nước



Những điều gì cần chú ý khi xây dựng cây quyết định?

Ví dụ

Sử dụng 3 đặc trưng: có thể bay, có 2 chân, sống dưới nước



Entropy

- Ôn lại

- Khái niệm entropy thông tin được Claude Shannon giới thiệu trong bài báo “A Mathematical Theory of Communication” năm 1948 => “Shannon entropy” .

Quantities of the form $H = -\sum p_i \log p_i$ (the constant K merely amounts to a choice of a unit of measure) play a central role in information theory as measures of information, choice and uncertainty. The form of H will be recognized as that of entropy as defined in certain formulations of statistical mechanics⁸ where p_i is the probability of a system being in cell i of its phase space. H is then, for example, the H in Boltzmann's famous H theorem. We shall call $H = -\sum p_i \log p_i$ the entropy of the set of probabilities p_1, \dots, p_n . If x is a chance variable we will write $H(x)$ for its entropy; thus x is not an argument of a function but a label for a number, to differentiate it from $H(y)$ say, the entropy of the chance variable y .

The entropy in the case of two possibilities with probabilities p and $q = 1 - p$, namely

$$H = -(p \log p + q \log q)$$

Entropy

- Ôn lại
 - Khái niệm entropy thông tin được Claude Shannon giới thiệu trong bài báo “A Mathematical Theory of Communication” năm 1948 => “Shannon entropy” .

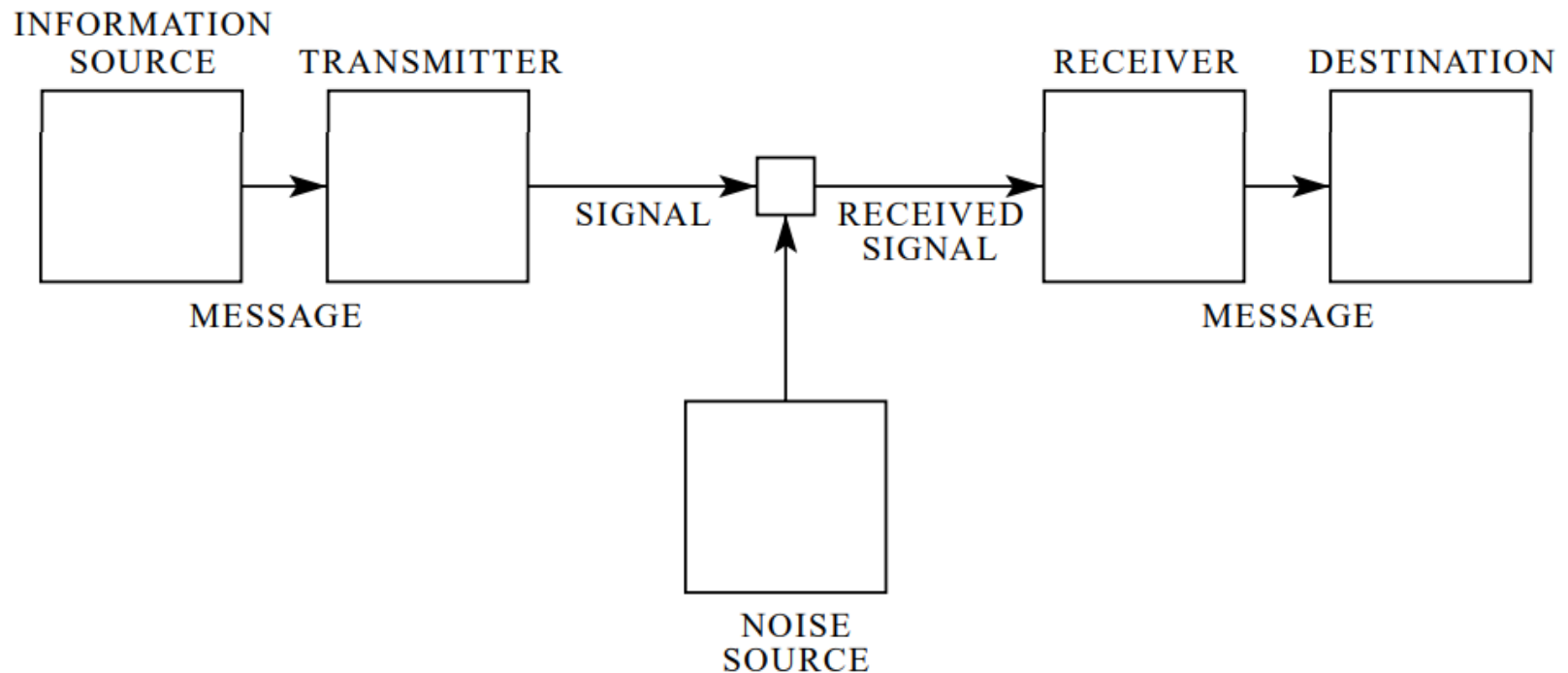


Fig. 1—Schematic diagram of a general communication system.

Entropy

- Có thể sử dụng để đo mức độ hỗn tạp của bộ dữ liệu
- Tất cả các mẫu thuộc cùng một lớp => Entropy = 0

$$\text{Entropy}(S) = \sum_{i=1}^K -p(C_i) \log_2(p(C_i))$$

S : tập dữ liệu (N mẫu)

K : số lượng lớp

N_{C_i} : số lượng mẫu thuộc lớp C_i


$p(C_i)$: tỷ lệ các mẫu thuộc lớp

$$p(C_i) = \frac{N_{C_i}}{N}$$

Entropy

- Ví dụ: Cho tập dữ liệu S , có $N = 100$ mẫu thuộc 2 lớp $\{C_1, C_2\}$ tính Entropy cho 5 trường hợp sau và nhận xét

Case	$\in C_1$	$\in C_2$	$p(C_1)$	$p(C_2)$	Entropy
1	100	0			
2	80	20			
3	60	40			
4	50	50			
5	40	60			


$$\text{Entropy}(S) = -p(C_1)\log_2(p(C_1)) - p(C_2)\log_2(p(C_2))$$



Entropy

- Ví dụ: Cho tập dữ liệu S , có $N = 100$ mẫu thuộc 2 lớp $\{C_1, C_2\}$ tính Entropy cho 5 trường hợp sau và nhận xét

Case	$\in C_1$	$\in C_2$	$p(C_1)$	$p(C_2)$	Entropy
1	100	0	100/100	0/100	0
2	80	20	80/100	20/100	0.722
3	60	40	60/100	40/100	0.971
4	50	50	50/100	50/100	1
5	40	60	40/100	60/100	0.971

- Tất cả các mẫu thuộc cùng 1 lớp \Rightarrow Entropy = 0
- Dữ liệu càng hỗn tạp \Rightarrow giá trị Entropy càng lớn

Information Gain

- Information Gain của thuộc tính/đặc trưng A đối với tập S

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

$\text{Values}(A)$: Tập các giá trị có thể có của thuộc tính A

S_v : Tập các mẫu với thuộc tính A có giá trị bằng v
(là một tập con của S)

ID3 (Iterative Dichotomiser 3)

- ID3 được Ross Quinlan dùng để tạo ra cây quyết định từ tập dữ liệu
 - Xây dựng từ nút gốc, kiểu từ trên xuống
 - Mỗi nút: sử dụng thuộc tính “tốt nhất” (~ có khả năng hỗ trợ phân loại tốt nhất ~ có **entropy** nhỏ nhất hay có **information gain** lớn nhất)
 - Xây dựng tiếp cây quyết định cho tới khi
 - Tất cả các đặc trưng đều đã sử dụng, hoặc
 - Có thể phân lớp tất cả các mẫu

Ví dụ

- Xét tập dữ liệu

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Ví dụ

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Xác định đặc trưng nào dùng ở nút gốc?



PlayTennis: Yes (+), No (-)

Xét đặc trưng Wind

Values(Wind) = Weak, Strong

$S : [9+, 5-]$ 14 mẫu

$S_{\text{Weak}} \leftarrow [6+, 2-]$ 8 mẫu

$S_{\text{Strong}} \leftarrow [3+, 3-]$ 6 mẫu

$$\text{Entropy}(S) = -\frac{9}{14} \log_2 \left(\frac{9}{14} \right) - \frac{5}{14} \log_2 \left(\frac{5}{14} \right) = 0.94$$

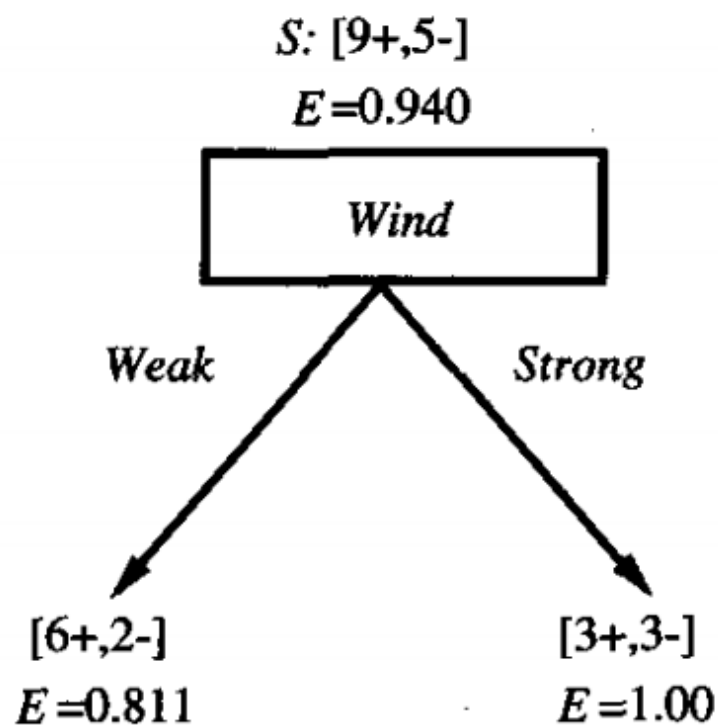
$$\text{Entropy}(S_{\text{Weak}}) = -\frac{6}{8} \log_2 \left(\frac{6}{8} \right) - \frac{2}{8} \log_2 \left(\frac{2}{8} \right) = 0.811$$

$$\text{Entropy}(S_{\text{Strong}}) = -\frac{3}{6} \log_2 \left(\frac{3}{6} \right) - \frac{3}{6} \log_2 \left(\frac{3}{6} \right) = 1$$

$$\text{Gain}(S, \text{Wind}) = \text{Entropy}(S) - \frac{8}{14} \text{Entropy}(S_{\text{Weak}}) - \frac{6}{14} \text{Entropy}(S_{\text{Strong}}) = 0.048$$

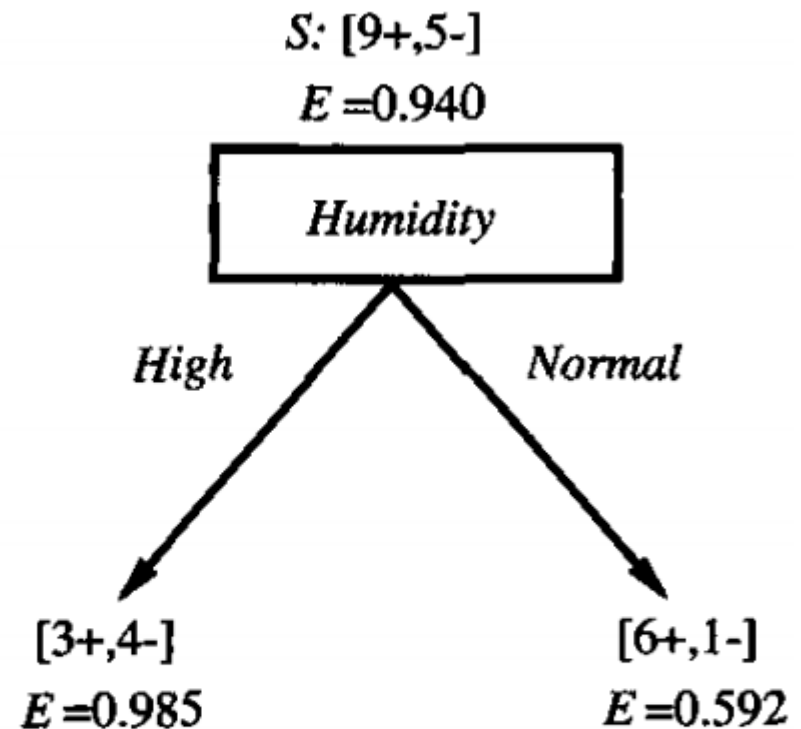
Ví dụ

- Tính toán tương tự cho các đặc trưng khác



$Gain(S, Wind)$

$$\begin{aligned} &= .940 - (8/14).811 - (6/14)1.0 \\ &= .048 \end{aligned}$$



$Gain(S, Humidity)$

$$\begin{aligned} &= .940 - (7/14).985 - (7/14).592 \\ &= .151 \end{aligned}$$

Ví dụ

- Tính toán tương tự cho tất cả các đặc trưng, ta có

$$\text{Gain}(S, \text{Outlook}) = 0.246$$

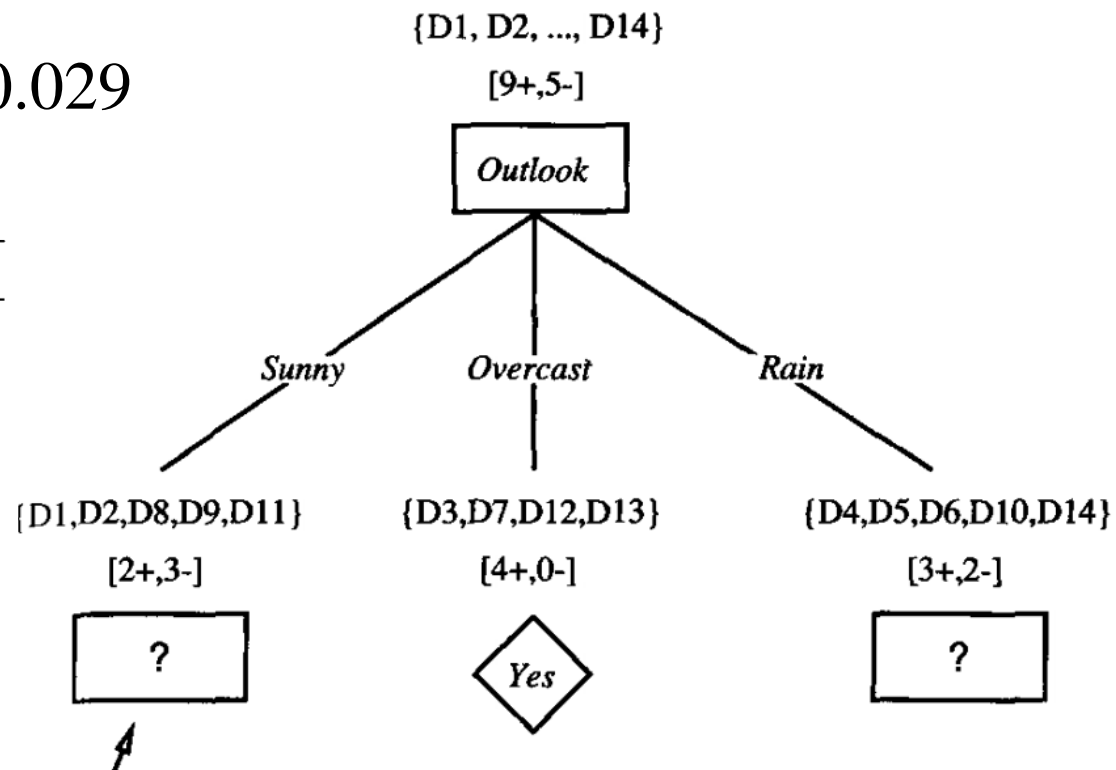
=> chọn là đặc trưng dùng ở nút gốc

$$\text{Gain}(S, \text{Humidity}) = 0.151$$

$$\text{Gain}(S, \text{Wind}) = 0.048$$

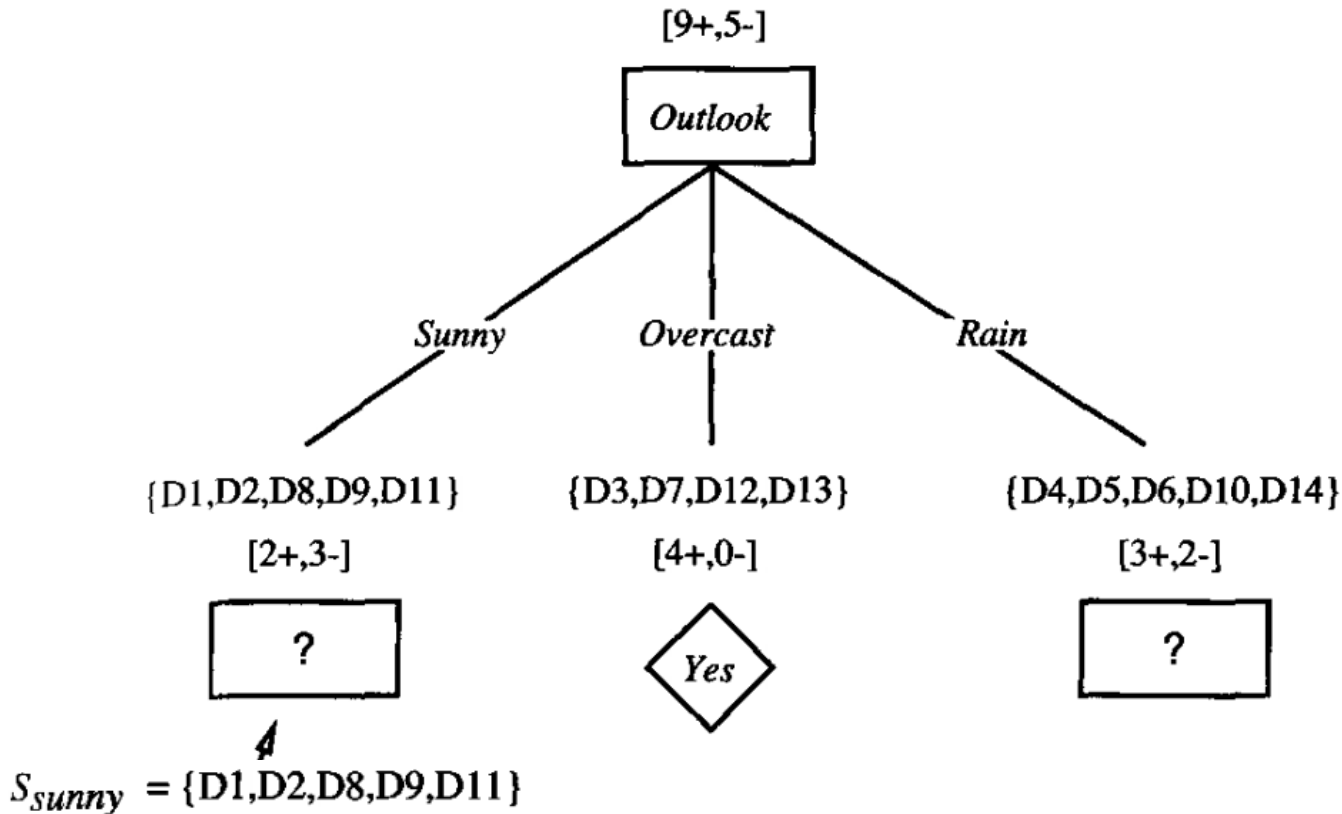
$$\text{Gain}(S, \text{Temperature}) = 0.029$$

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



Ví dụ

- Lặp lại quá trình để xây dựng cây
 $\{D1, D2, \dots, D14\}$

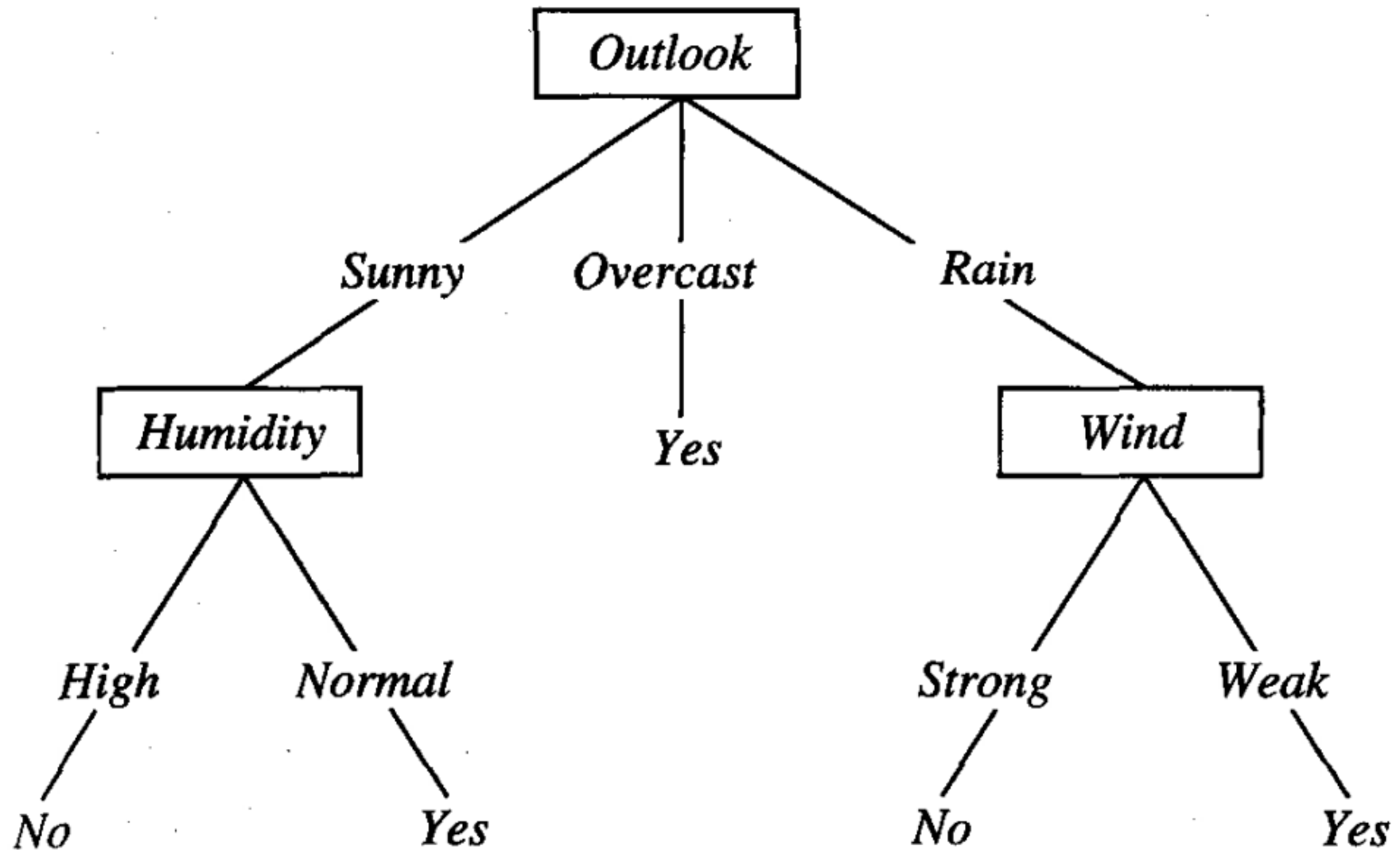


$$\text{Gain}(S_{\text{sunny}}, \text{Humidity}) = .970 - (3/5) 0.0 - (2/5) 0.0 = .970 \Rightarrow \text{Chọn đặc trưng Humidity}$$

$$\text{Gain}(S_{\text{sunny}}, \text{Temperature}) = .970 - (2/5) 0.0 - (2/5) 1.0 - (1/5) 0.0 = .570$$

$$\text{Gain}(S_{\text{sunny}}, \text{Wind}) = .970 - (2/5) 1.0 - (3/5) .918 = .019$$

Ví dụ

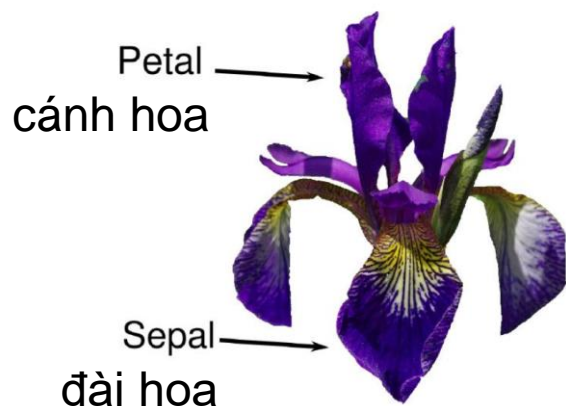


ID3 (Iterative Dichotomiser 3)

- Chọn ra đặc trưng có khả năng phân loại “tốt nhất” tại mỗi bước
 - Tìm kiếm tham lam (greedy)
 - Có thể không phải là tối ưu
- ID3 chỉ đảm bảo tìm được lời giải tối ưu cục bộ
- Khi cây trở nên phức tạp (nhiều nút, nhiều nhánh, nhiều nút lá chỉ có một số ít điểm dữ liệu) => vấn đề quá khớp (overfitting)

Ví dụ

- Phân loại các loài hoa diên vĩ (iris)

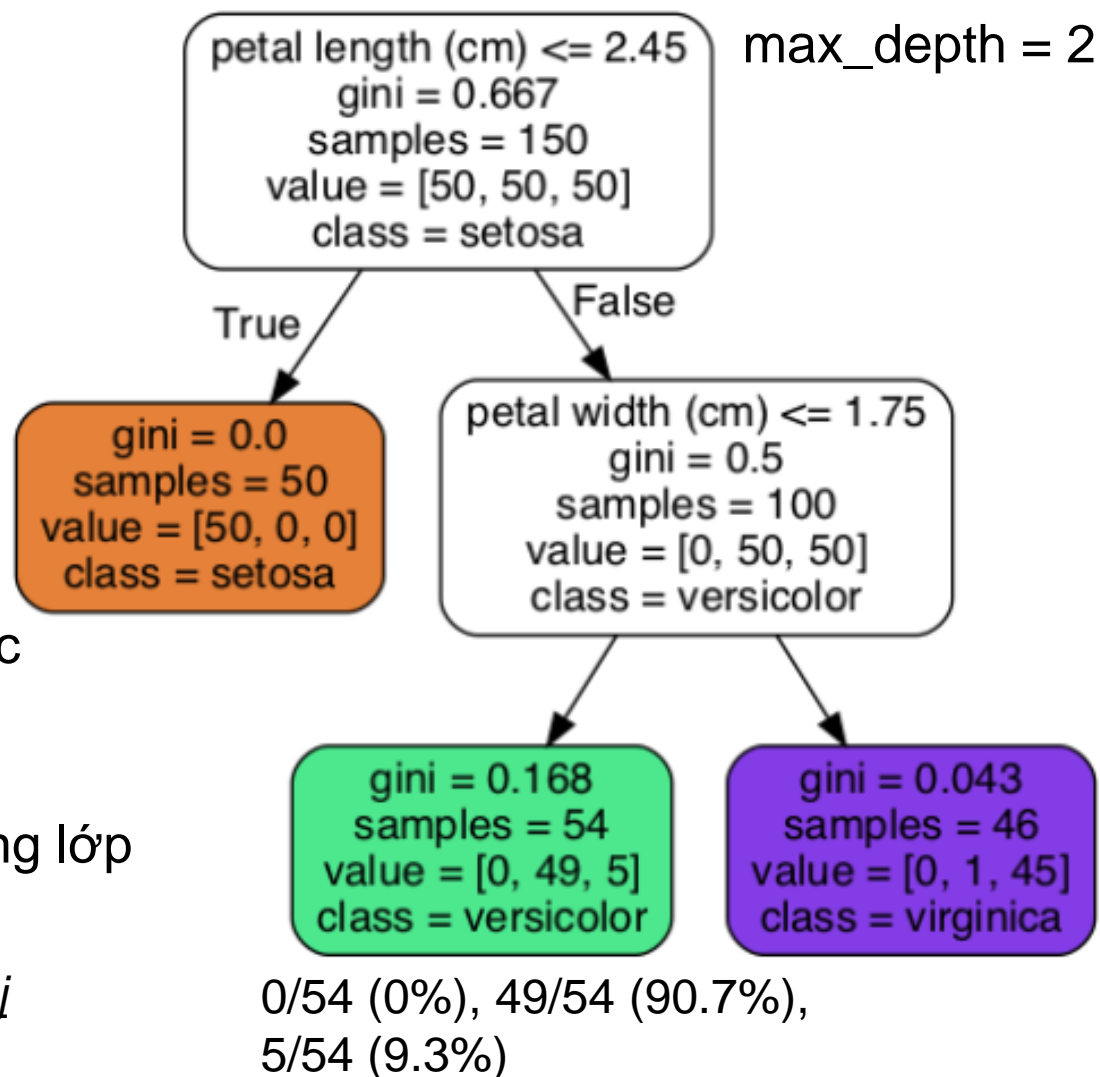


- **gini**: đo lường sự vẩn đục (impurity) không tinh khiết

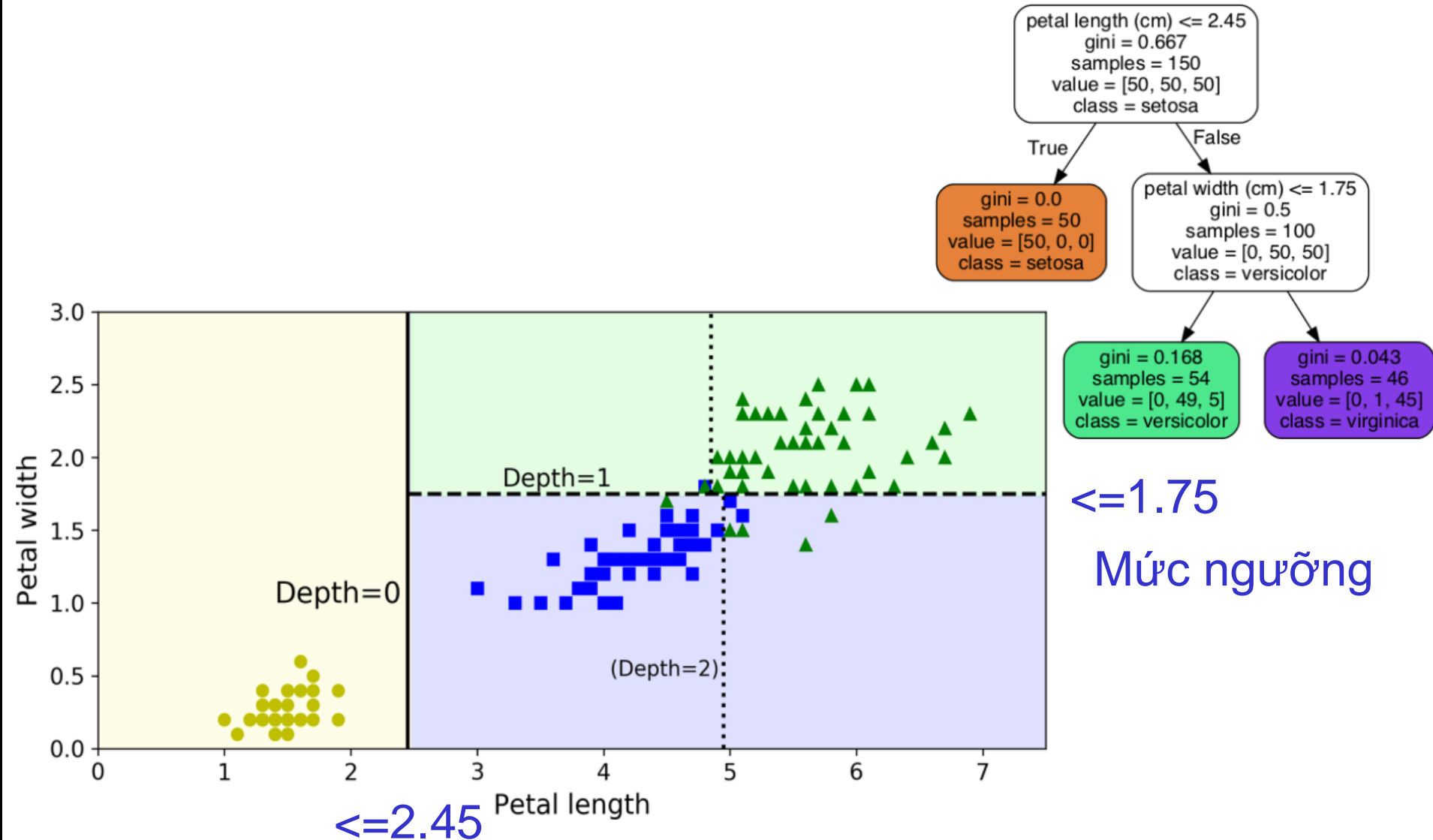
gini = 0 => tất cả các mẫu thuộc cùng 1 lớp

$$G_i = 1 - \sum_{j=1}^K p_{i,j}^2 \quad K: \text{số lượng lớp}$$

$$p_{i,j} = \frac{\text{số lượng mẫu thuộc lớp } j}{\text{số lượng mẫu tại nút } i}$$



Ví dụ



Thuật toán huấn luyện CART

- Classification and Regression Tree (CART) in sklearn
 - Chia tập huấn luyện thành 2 tập con (bên trái, bên phải) sử dụng 1 thuộc tính và 1 mức ngưỡng (k, t_k)
 - + Tìm (k, t_k) để tạo ra các tập con tinh khiết nhất ~ hàm mục tiêu nhỏ nhất

$$J(k, t_k) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}}$$

$G_{\text{left/right}}$: Đo độ tinh khiết của tập con bên trái, bên phải

$m_{\text{left/right}}$: Số lượng mẫu trong tập con bên trái, bên phải

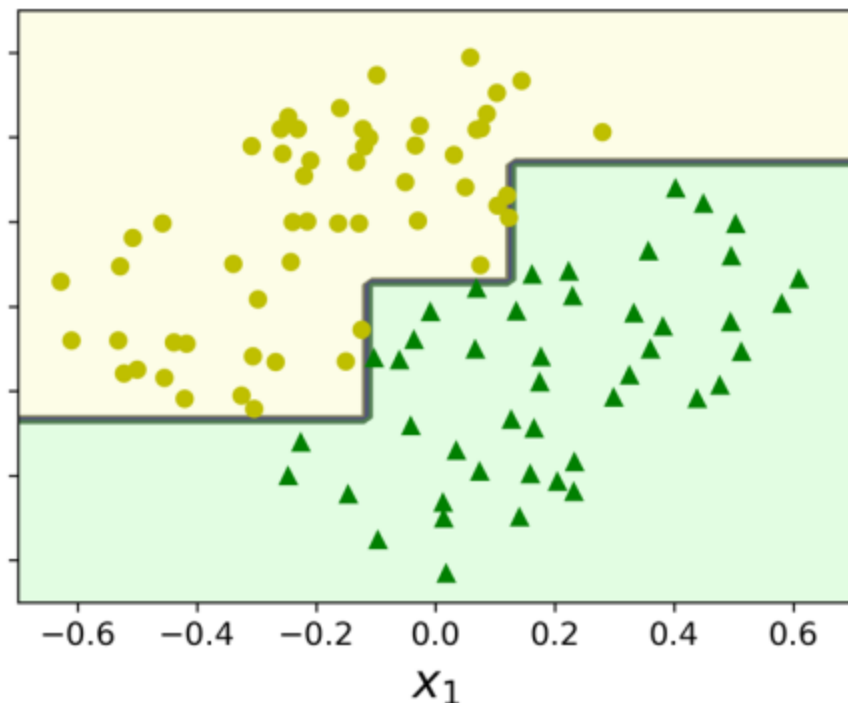
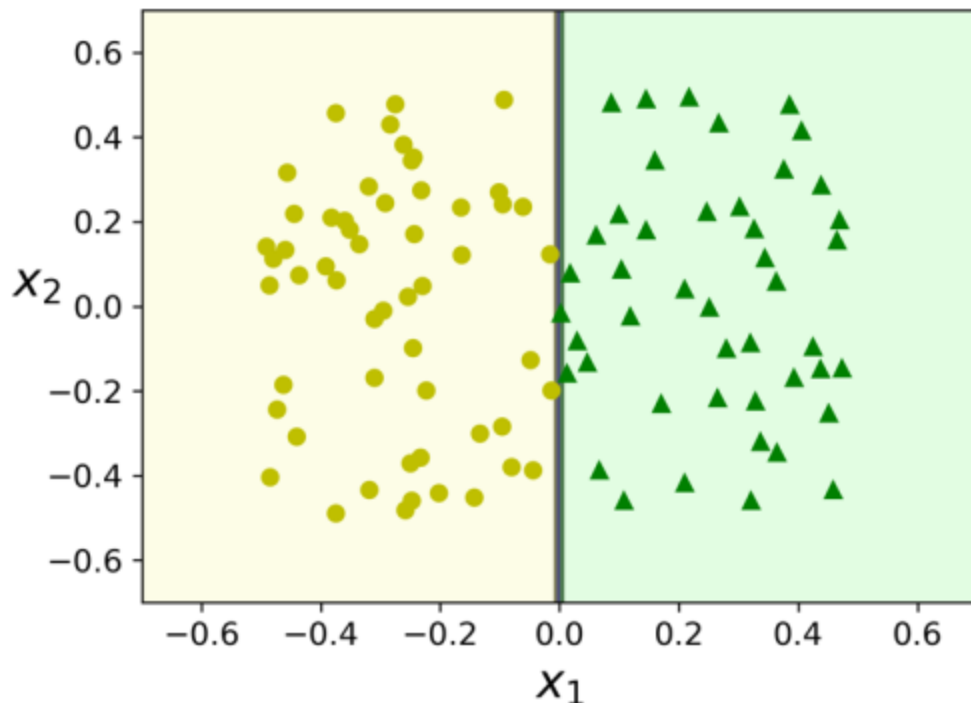
- Lặp lại việc chia các tập con như trên cho từng tập con cho tới khi không thể chia tiếp được do không thể giảm được độ không tinh khiết, hoặc tới độ sâu tối đa đã thiết lập (max_depth)

Điều chuẩn

- Cây quyết định dùng rất ít giả định về dữ liệu huấn luyện => dễ xảy ra overfitting nếu không sử dụng các ràng buộc
- Để tránh overfitting
 - Hạn chế
 - Độ sâu lớn nhất (max_depth)
 - Hình dạng của cây quyết định: số lượng lớn nhất các nút lá (max_leaf_node), số lượng mẫu nhỏ nhất mà 1 nút lá cần có (min_samples_leaf), số lượng mẫu nhỏ nhất mà 1 nút cần phải có để được tiến hành phân chia (min_samples_split), ...
 - Pruning: cắt tỉa/xóa bớt những nút không cần thiết

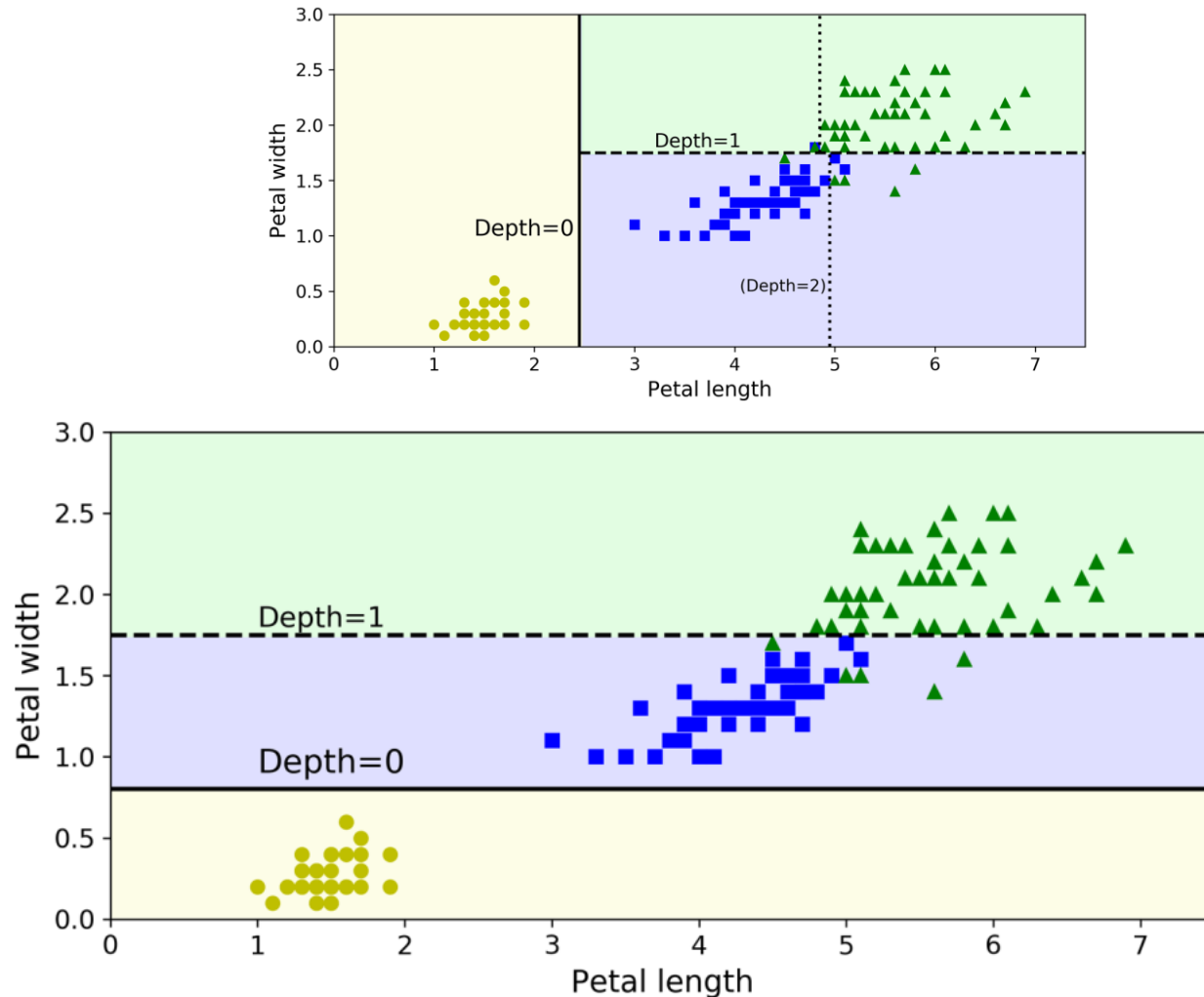
Giới hạn

- Nhạy cảm với việc xoay dữ liệu huấn luyện do
 - Cây quyết định phù hợp với các ranh giới quyết định trục giao (tất cả các phần phân chia đều vuông góc với một trục)



Giới hạn

- Cây quyết định rất nhạy cảm với những thay đổi nhỏ trong dữ liệu huấn luyện



Ưu và nhược điểm

Ưu điểm

Dễ dàng trực quan hóa và dễ hiểu cho những người không phải là chuyên gia

Thuật toán không thay đổi khi thay đổi tỷ lệ dữ liệu do: mỗi thuộc tính được xử lý một cách riêng biệt; các thuộc tính không cần phải tiền xử lý

Nhược điểm

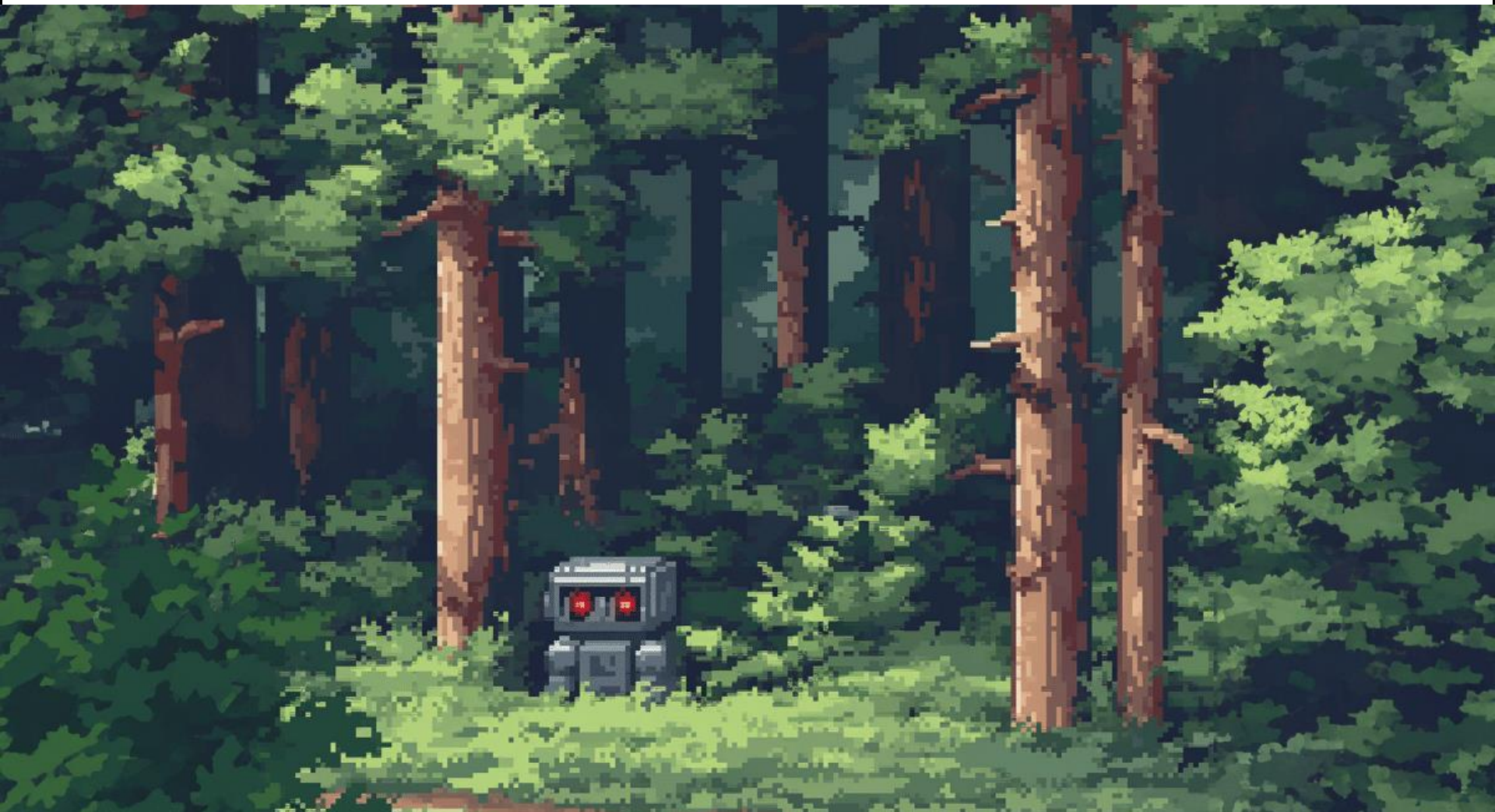
Có xu hướng quá phù hợp và khó tổng quát hóa ngay cả khi sử dụng các kỹ thuật như pre-pruning

Nhạy cảm với việc xoay bộ dữ liệu huấn luyện

Rất nhạy cảm với những thay đổi nhỏ trong dữ liệu huấn luyện

Random forests

- Chương 7 [TLHT02]: Ensemble Learning and Random



<https://shelf.io/blog/random-forests-in-machine-learning/>

Ensemble learning

- Học tập tập thể/tập hợp
 - Trí tuệ đám đông
 - Tổng hợp kết quả của một nhóm các mô hình dự đoán (ví dụ mô hình phân lớp hay hồi quy)
=> thường sẽ có kết quả tốt hơn kết quả thu được khi sử dụng một mô hình đơn lẻ
 - Trong thực tế
 - Khi đã xây dựng được một số ít mô hình tốt => có thể kết hợp để có một mô hình tốt hơn
 - Linh hoạt, mạnh mẽ và khá đơn giản để sử dụng

Ensemble learning

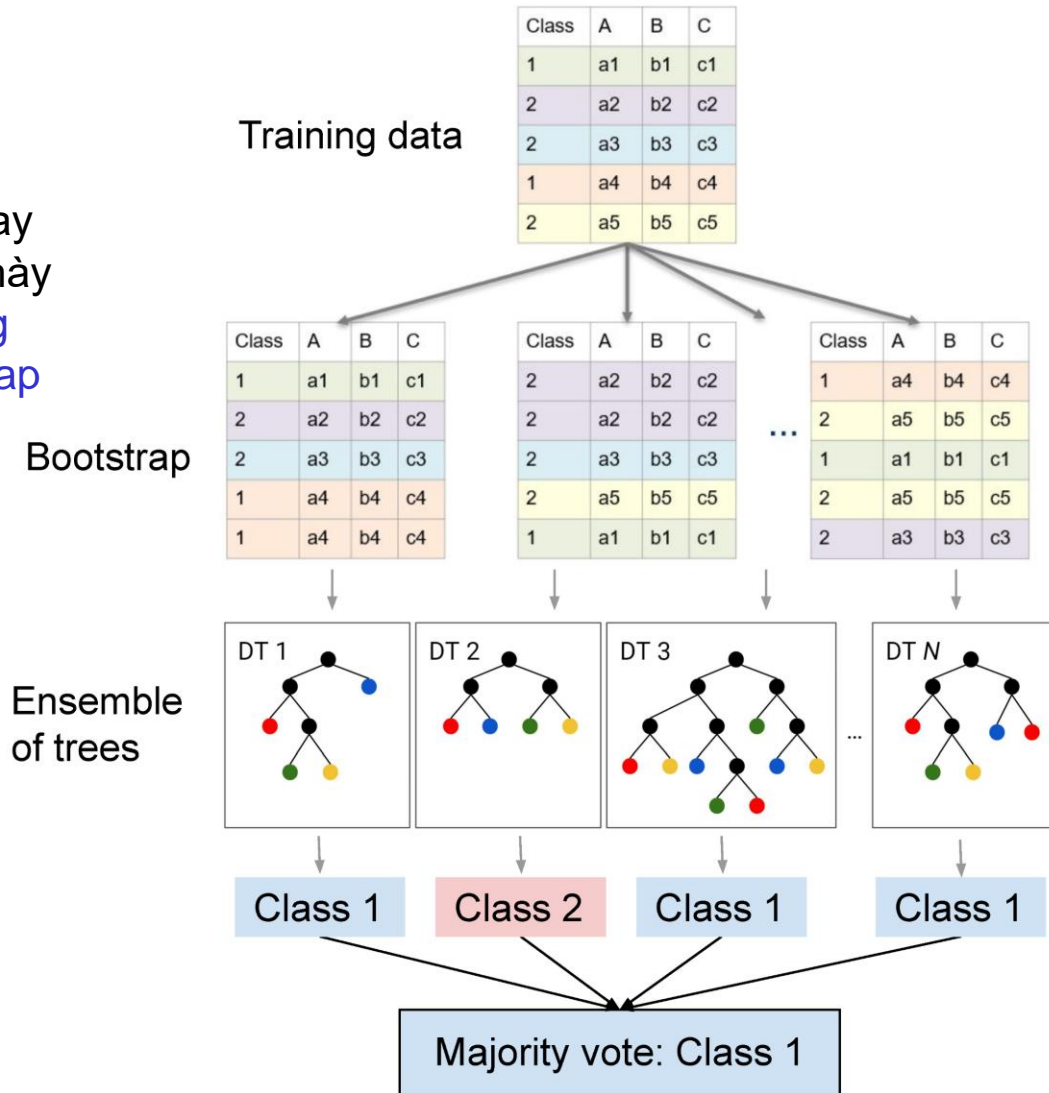
- Các phương pháp phổ biến nhất
 - Voting classifiers
 - Hard voting
 - Soft voting
 - Scikit-learn: VotingClassifier
 - Bagging (bootstrap aggregating) and pasting ensembles
 - Scikit-learn: BaggingClassifier
 - Random forest
 - Boosting and stacking ensembles
 - AdaBoost (adaptive boosting), gradient boosting, histogram-based gradient boosting
 - Scikit-learn: AdaBoostClassifier, GradientBoostingClassifier, HistGradientBoostingClassifier, StackingClassifier

Random forests

- Có thể huấn luyện một nhóm các mô hình phân lớp Decision Tree
 - Mỗi mô hình được huấn luyện trên một tập hợp con ngẫu nhiên khác nhau của bộ dữ liệu huấn luyện
- Để đưa ra dự đoán, lấy các dự đoán của tất cả các cây riêng lẻ, sau đó dự đoán lớp nhận được nhiều phiếu bầu nhất
- Một tập hợp các Decision Tree như vậy được gọi là **Random Forest**
 - Mặc dù đơn giản, đây là một trong những thuật toán ML mạnh mẽ nhất hiện nay

Random forests

Khi lấy mẫu được thực hiện với sự thay thế, phương pháp này được gọi là **bagging** (viết tắt của **bootstrap aggregating**)



Random forests

- Scikit-learn: RandomForestClassifier

```
from sklearn.ensemble import RandomForestClassifier
```

```
rnd_clf = RandomForestClassifier(n_estimators=500, max_leaf_nodes=16, n_jobs=-1)  
rnd_clf.fit(X_train, y_train)
```

```
y_pred_rf = rnd_clf.predict(X_test)
```

n_estimators=500

Số lượng cây: 500

max_leaf_nodes=16

Số lượng nút lá tại mỗi cây: 16

Có thể huấn luyện một cách song song, sử dụng các core CPU khác nhau, hoặc thậm chí các server khác nhau ?

Đánh giá mô hình

- Ma trận nhầm lẫn (confusion matrix)

Để minh họa, xét cho trường hợp phân lớp nhị phân

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

độ chính xác

$$\text{Precision} = \frac{TP}{TP+FP}$$

độ chuẩn xác

$$\text{Recall} = \frac{TP}{TP+FN}$$

độ tin cậy

$$F_1 = 2 \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Dự đoán

negative class

positive class

negative class

Thực sự

positive class

negative class	TN True Negative	FP False Positive
	FN False Negative	TP True Positive

Đánh giá mô hình

- Ma trận nhầm lẫn giúp ta hiểu rõ việc phân loại đúng hay sai của các điểm dữ liệu
 - Số lượng mẫu trên đường chéo chính = số mẫu được phân loại đúng vào các lớp dữ liệu
- TP (True Positive): Tổng số mẫu được phân lớp đúng là mẫu dương tính
- TN (True Negative): Tổng số mẫu được phân lớp đúng là mẫu âm tính
- FP (False Positive): Tổng số mẫu âm tính bị phân lớp sai thành mẫu dương tính
- FN (False Negative): Tổng số mẫu dương tính bị phân lớp sai thành mẫu âm tính

Đánh giá mô hình

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

độ chính xác: tổng số các mẫu được phân lớp đúng chia cho tổng số mẫu => chỉ cho biết tỷ lệ mẫu được dự báo đúng, không chỉ ra được cụ thể mỗi lớp được phân loại như thế nào

Từ TP, TN; FP, FN: Ta tính được độ chuẩn xác và độ tin cậy cho từng lớp

Ví dụ: Ta đang quan tâm tới các mẫu thuộc positive class (dương tính)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

độ chuẩn xác: tổng số các mẫu thuộc một lớp (dương tính) được phân loại đúng chia cho tổng số các mẫu được phân lớp vào lớp đó (dương tính)

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

độ tin cậy: tổng số các mẫu thuộc một lớp (dương tính) được phân loại đúng chia cho tổng số các mẫu thuộc lớp đó (dương tính)

$$F_1 = 2 \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Kết hợp cả precision và recall

Ví dụ

- Tập huấn luyện có 1000 mẫu là 1000 bức ảnh các con rắn:
 - 990 bức ảnh: những con rắn bình thường (negative class)
 - 10 bức ảnh: những con rắn thuộc vào loại hiếm/mới (positive class)
- Xây dựng một bộ phân lớp: rắn bình thường và rắn hiếm
 - Phân lớp: 1000 mẫu đều là rắn bình thường
 - ? Đánh giá bộ phân lớp đã xây dựng này

TN = 990 FP = 0

FN = 10 TP = 0

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{990}{1000} = 0.99$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = 0$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = 0$$

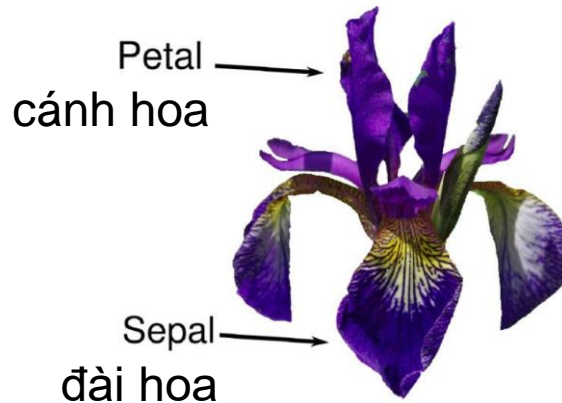


Đánh giá mô hình

- Bên cạnh đó có thể dùng
 - Đường Receiver Operating Characteristic (ROC)
 - Area Under the Curve (AUC)

2.3.6 Ví dụ về bài toán phân lớp

- Ví dụ 1: Phân loại các loài hoa diên vĩ (iris)



iris setosa



petal sepal

iris versicolor



petal sepal

iris virginica



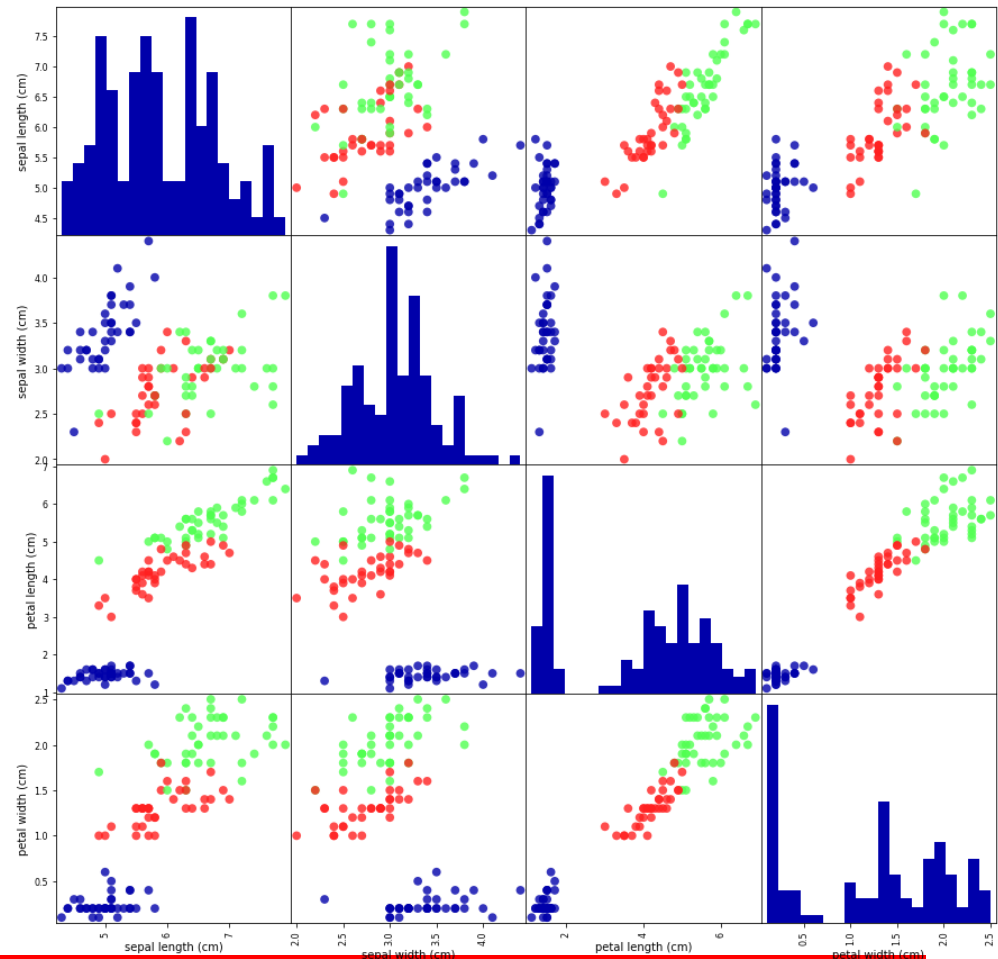
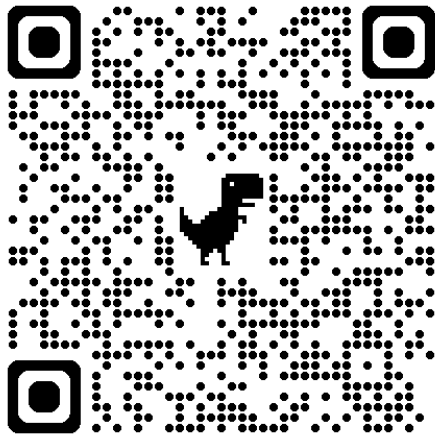
petal sepal

<https://editor.analyticsvidhya.com/uploads/51518iris%20img1.png>

Ví dụ 1

- Ví dụ 1: Sinh viên chạy từng bước trong ví dụ để hiểu ví dụ để hiểu cách phân lớp hoa

https://github.com/amueller/introduction_to_ml_with_python/blob/main/01-introduction.ipynb



Ví dụ 2

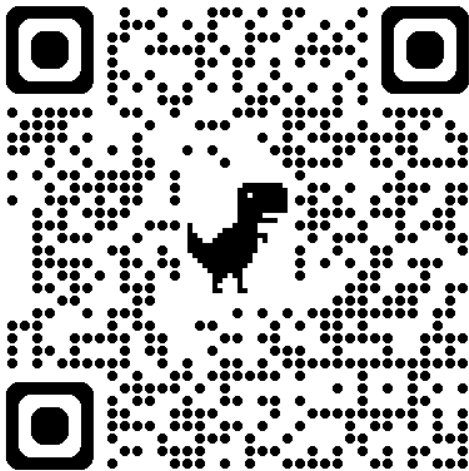
- Ví dụ 2: Phân loại chữ số viết tay sử dụng bộ dữ liệu MNIST



Ví dụ 2

- Sinh viên chạy từng bước trong ví dụ để hiểu cách sử dụng các hàm

https://github.com/ageron/handson-ml3/blob/main/03_classification.ipynb



số ?

784 features ?

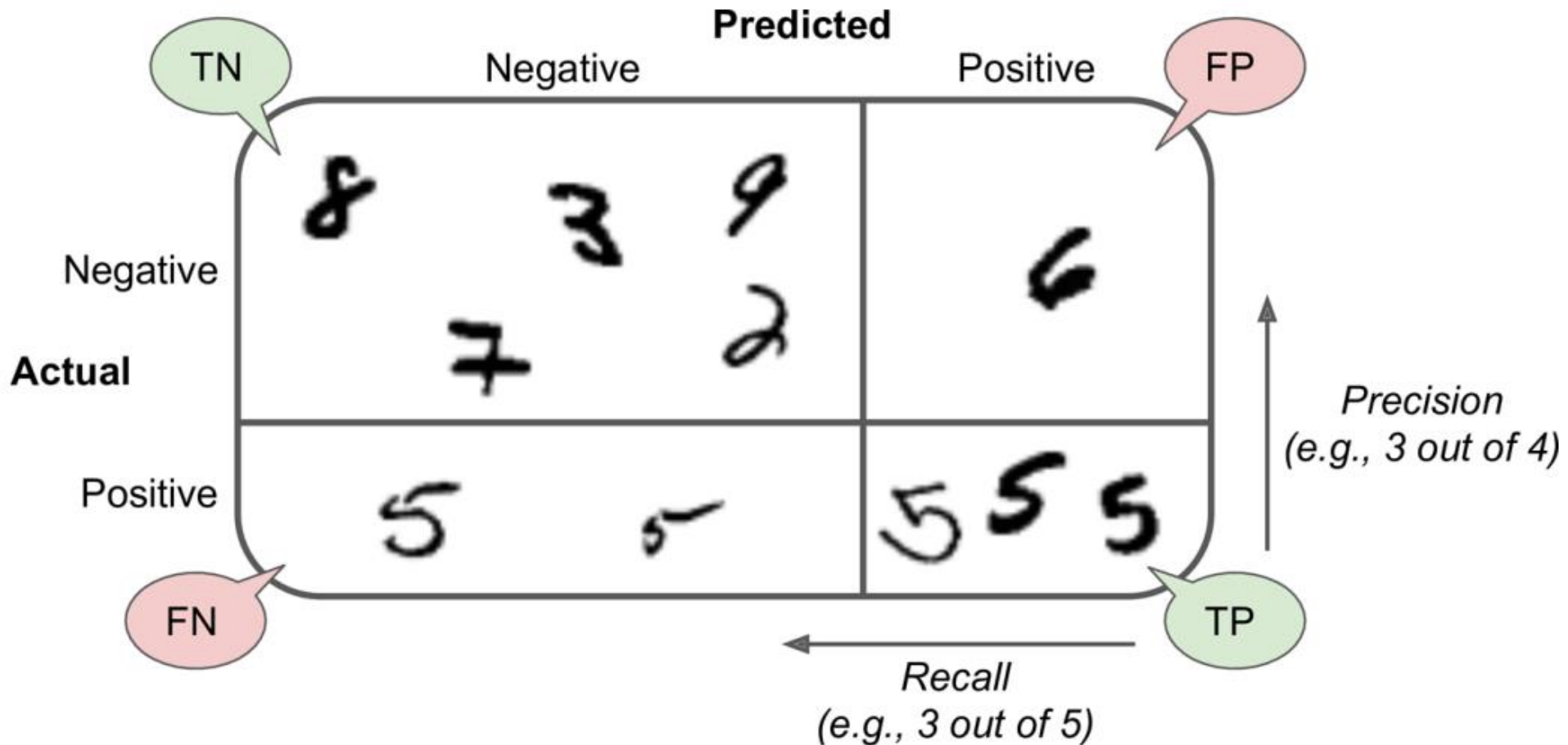
$784 = 28 \times 28$ pixels

```
**Author**: Yann LeCun, Corinna Cortes, Christopher J.C. Burges  
**Source**: [MNIST Website](http://yann.lecun.com/exdb/mnist/) - Date unknown  
**Please cite**:
```

The MNIST database of handwritten digits with 784 features, raw data available at: <http://yann.lecun.com/exdb/mnist/>. It can be split in a training set of the first 60,000 examples, and a test set of 10,000 examples

Ví dụ 2

- Chú ý tới cách đánh giá mô hình



Ví dụ 2

- Sử dụng: confusion_matrix, precision_score, recall_score, f1_score

```
>>> from sklearn.metrics import confusion_matrix
>>> confusion_matrix(y_train_5, y_train_pred)
array([[53057, 1522],
       [ 1325, 4096]])
```

```
>>> from sklearn.metrics import precision_score, recall_score
>>> precision_score(y_train_5, y_train_pred) # == 4096 / (4096 + 1522)
0.7290850836596654
>>> recall_score(y_train_5, y_train_pred) # == 4096 / (4096 + 1325)
0.7555801512636044
```

```
>>> from sklearn.metrics import f1_score
>>> f1_score(y_train_5, y_train_pred)
0.7420962043663375
```

Tổng kết

- Nắm được cách xây dựng cây quyết định
- Biết các sử dụng thuật toán ID3
- Hiểu và áp dụng được các giải thuật dựa trên cây quyết định, đặc biệt là random forest

Hoạt động sau buổi học

- Làm bài tập về nhà
- Ôn lại các vấn đề liên quan đến cây quyết định

Chuẩn bị cho buổi học tiếp theo

- Ôn lại về mạng neuron

Tài liệu tham khảo

- Chương 2 và chương 3 cuốn sách Machine Learning, tác giả Tom M. Mitchell
- Bài báo quan trọng của Shannon:
<http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf>