CÁC PHƯƠNG PHÁP DỰA TRÊN CÂY
(Tree based models)

CÂY QUYẾT ĐỊNH
(Decision tree)

Cây quyết định là gì?

- Cây quyết định là cây mà mỗi nút biểu diễn một đặc trưng (tính chất), mỗi nhánh (branch) biểu diễn một quy luật (rule) và mỗi lá biểu biễn một kết quả (giá trị cụ thể hay một nhánh tiếp tục)
- Cây quyết định bắt chước mức độ suy nghĩ của con người nên nó đơn giản để hiểu và chuẩn bị dữ liệu.
- Cây quyết định giúp bạn thấy được logic từ dữ liệu

nguồn: Nguyễn Nhật Quang-Học máy

Cây quyết định là gì?

- Học cây quyết định (Decision tree –DT– learning)
  - Để học (xấp xỉ) một hàm mục tiêu có giá trị rời rạc (*discrete- valued target function*) – hàm phân lớp
  - Hàm phân lớp được biểu diễn bởi một cây quyết định

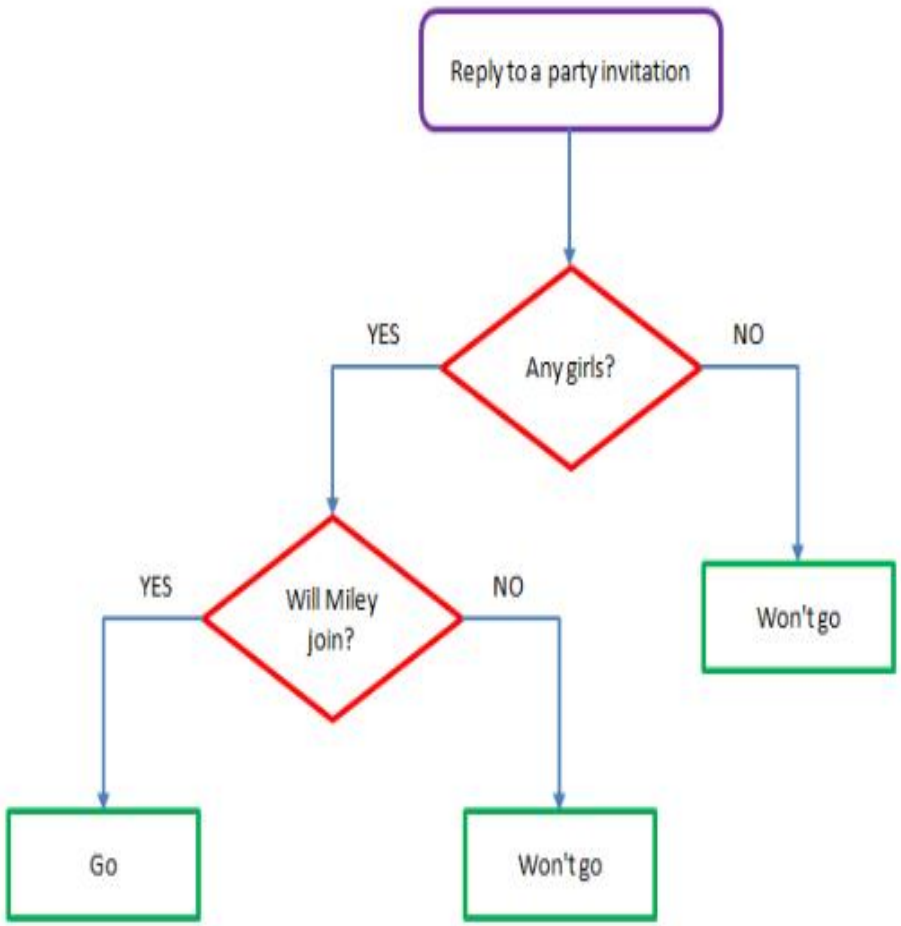
nguồn: Nguyễn Nhật Quang-Học máy

Cây quyết định là gì?

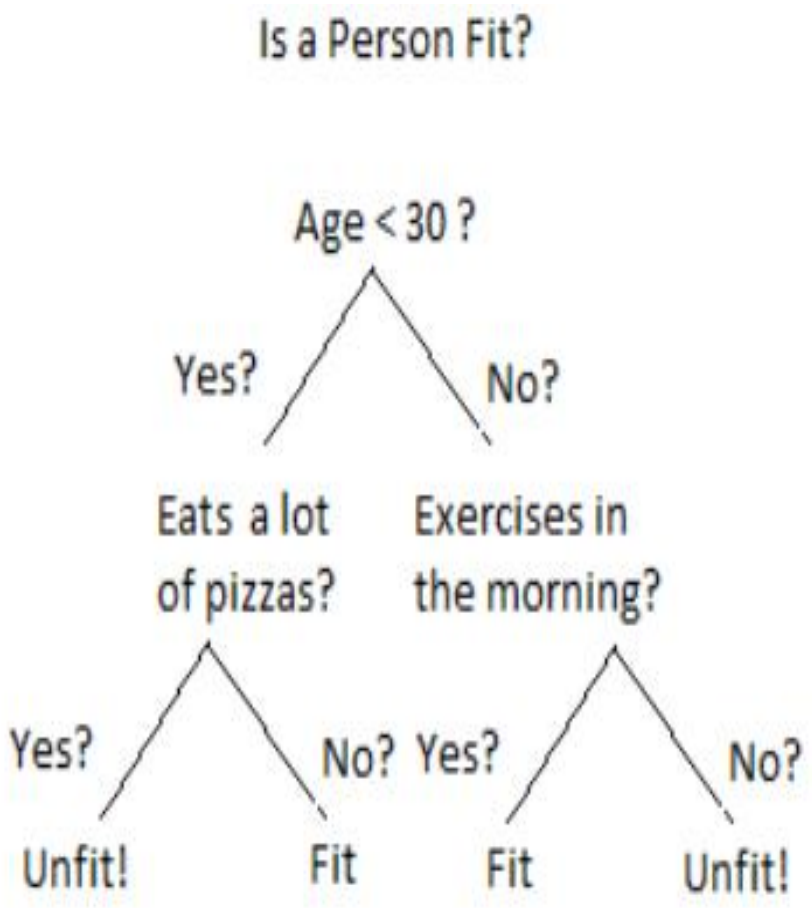
- Một cây quyết định có thể được biểu diễn (diễn giải) bằng một tập các luật IF-THEN (dễ đọc và dễ hiểu)
- Học cây quyết định có thể thực hiện ngay cả với các dữ liệu có chứa nhiễu/lỗi (noisy data)
- Được áp dụng thành công trong rất nhiều các bài toán ứng dụng thực tế

nguồn: Nguyễn Nhật Quang-Học máy

Cây quyết định là gì? – Ví dụ



Cây quyết định là gì? – Ví dụ



Biểu diễn cây quyết định

- Mỗi **nút trong** (*internal node)* biểu diễn một biến cần kiểm tra giá trị (*a variable to be tested*) đối với các mẫu
- Mỗi **nhánh** (*branch*) từ một nút sẽ tương ứng với một giá trị có thể của biến gắn với nút đó
- Mỗi **nút lá** (leaf node) biểu diễn một phân lớp (a classification)
- Một cây quyết định học được sẽ phân lớp đối với một mẫu, bằng cách duyệt cây từ nút gốc đến một nút lá → ***Nhãn lớp gắn với nút lá đó sẽ được gán cho mẫu cần phân lớp***

Biểu diễn cây quyết định

- Một cây quyết định biểu diễn một phép tuyển (disjunction) của các kết hợp (conjunctions) của các ràng buộc đối với các giá trị thuộc tính của các mẫu
- Mỗi đường đi (path) từ nút gốc đến một nút lá sẽ tương ứng với một kết hợp (conjunction) của các kiểm tra giá trị biến (variable tests)
- Cây quyết định (bản thân nó) chính là một phép tuyển của các kết hợp này

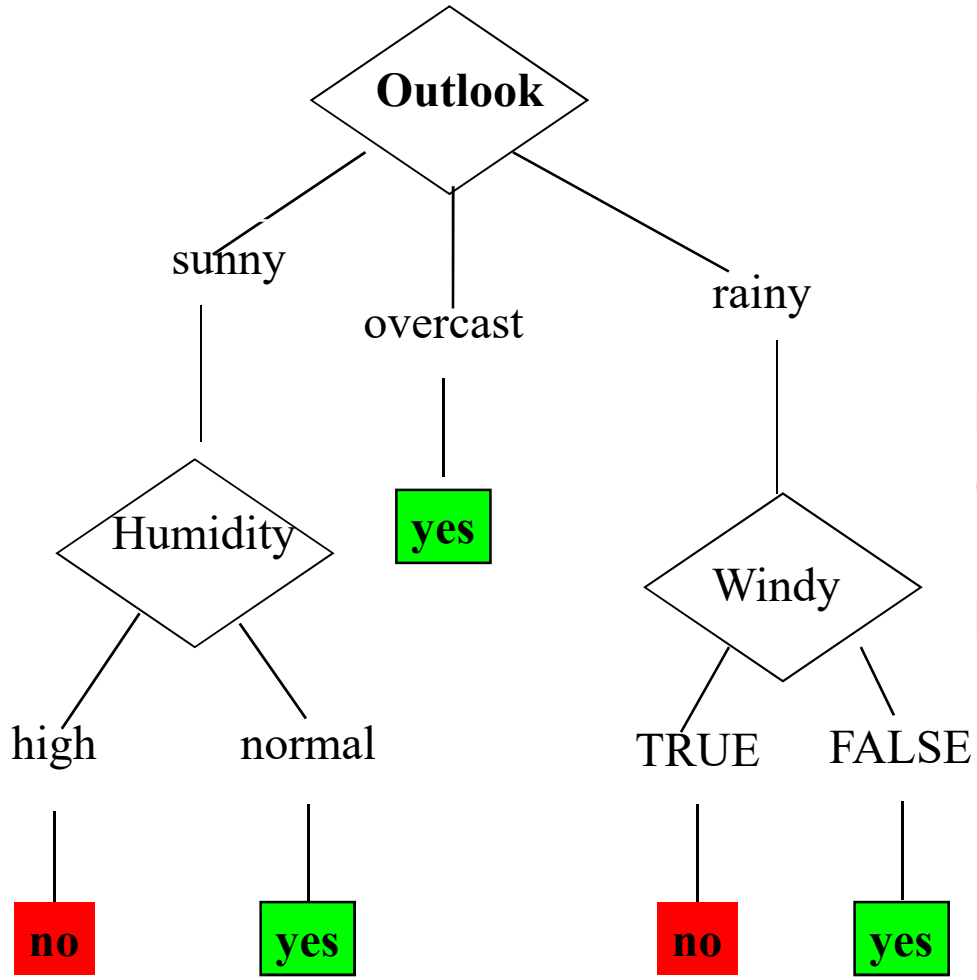
Tập dữ liệu Weather

Xét tập dữ liệu Weather ghi lại những ngày mà một người chơi (không chơi) tennis:

Day	Outlook	Temperature	Humidity	Windy	Play Tennis
D1	Sunny	Hot	High	FALSE	No
D2	Sunny	Hot	High	TRUE	No
D3	Overcast	Hot	High	FALSE	Yes
D4	Rain	Mild	High	FALSE	Yes
D5	Rain	Cool	Normal	FALSE	Yes
D6	Rain	Cool	Normal	TRUE	No
D7	Overcast	Cool	Normal	TRUE	Yes
D8	Sunny	Mild	High	FALSE	No
D9	Sunny	Cool	Normal	FALSE	Yes
D10	Rain	Mild	Normal	FALSE	Yes
D11	Sunny	Mild	Normal	TRUE	Yes
D12	Overcast	Mild	High	TRUE	Yes
D13	Overcast	Hot	Normal	FALSE	Yes
D14	Rain	Mild	High	TRUE	No

[Mitchell, 1997]

Mô hình cây QĐ có (không) chơi tennis



[(Outlook=`Sunny`) ∧ (Humidity=`Normal`)] ∨
(Outlook=`Overcast`) ∨

[(Outlook=`Rain`) ∧ (Windy=`False`)]

Xây dựng cây QĐ thế nào?

- **Phương pháp dựng cây theo Top-down**
  - Ban đầu, tất cả các mẫu trong tập huấn luyện đều đặt tại nút gốc.
  - Tách các mẫu theo đệ quy bằng cách chọn 1 thuộc tính trong mỗi lần tách cho đến khi gặp điều kiện dừng.
- **Phương pháp tỉa cây theo Bottom-up**
  - Ban đầu dựng cây lớn nhất có thể
  - Chuyển phần cây con hoặc nhánh từ phần đáy của cây lên nhằm cải thiện tính chính xác khi dự đoán mẫu mới

Giải thuật ID3

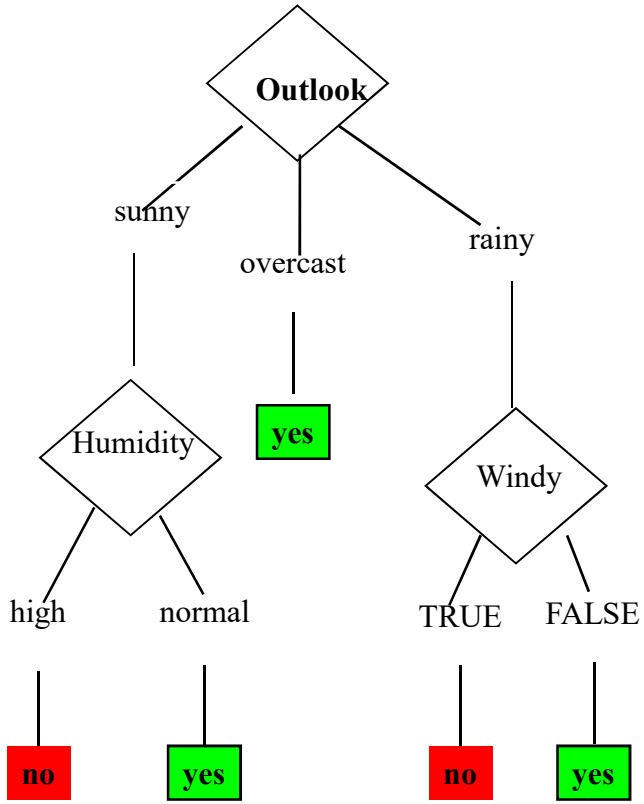
- Thực hiện giải thuật tìm kiếm tham lam (greedy search) đối với không gian các cây quyết định có thể
- Xây dựng (học) một cây quyết định theo chiến lược top-down, bắt đầu từ nút gốc
- Ở mỗi nút, biến kiểm tra (test variable) là biến có khả năng phân loại tốt nhất đối với các mẫu gắn với nút đó

Giải thuật ID3

- Tạo mới một cây con (sub-tree) của nút hiện tại cho mỗi giá trị có thể của biến kiểm tra, và tập huấn luyện sẽ được tách ra (thành các tập con) tương ứng với cây con vừa tạo
- Mỗi biến chỉ được phép xuất hiện tối đa 1 lần đối với bất kỳ một đường đi nào trong cây
- Quá trình phát triển (học) cây quyết định sẽ tiếp tục cho đến khi: Cây quyết định phân loại hoàn toàn (perfectly classifies) các mẫu, hoặc tất cả các thuộc tính đã được sử dụng

Lựa chọn biến kiểm tra

- Tại mỗi nút, chọn biến kiểm tra như thế nào?
- Chọn biến **quan trọng nhất** cho việc phân lớp các mẫu gắn với nút đó
- Làm thế nào để đánh giá khả năng của một biến đối với việc phân tách các mẫu theo nhãn lớp của chúng?

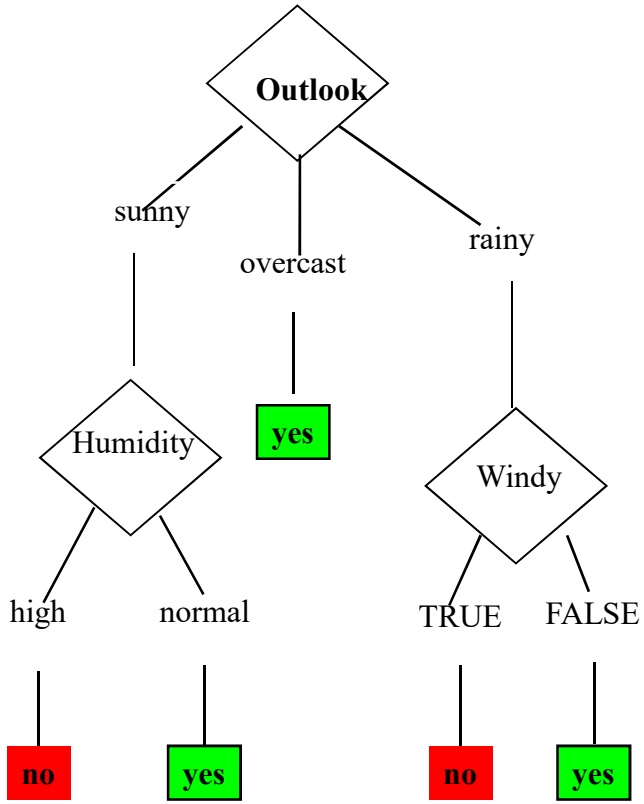


Lựa chọn biến kiểm tra

→ Sử dụng một đánh giá thống kê.

Một số cách đánh giá trên cây quyết định:

- Information gain ratio (C4.5)
- Gini index (CART)



Entropy

- Một đánh giá thường được sử dụng trong lĩnh vực lý thuyết thông tin (Information Theory)
- Để đánh giá mức độ hỗn tạp (impurity/inhomogeneity) của một tập
- Entropy của tập S đối với việc phân lớp có c lớp

$$Entropy(S) = \sum_{i=1}^{c} - p_i . \log_2 p_i$$

trong đó p_i là tỷ lệ các mẫu trong tập S thuộc vào lớp i, và quy ước **$0.\log_2 0 = 0$**

Entropy

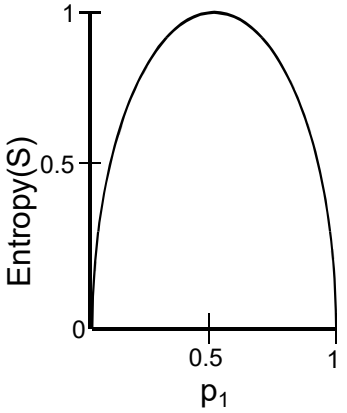
- Entropy của tập S đối với việc phân lớp có 2 lớp

$$Entropy(S) = -p_1.log_2p_1 – p_2.log_2p_2$$

- Ý nghĩa của entropy trong lĩnh vực lý thuyết thông tin: *Entropy của tập S chỉ ra số lượng bits cần thiết để mã hóa lớp của một phần tử được lấy ra ngẫu nhiên từ tập S*

Entropy

- Entropy = 0, nếu tất cả các mẫu thuộc cùng một lớp (c_1 hoặc c_2)
- Entropy =1, số lượng các mẫu thuộc về lớp c_1 bằng số lượng các mẫu thuộc về lớp c_2
- Entropy = một giá trị trong khoảng (0,1), nếu như số lượng các mẫu thuộc về lớp c_1 khác với số lượng các mẫu thuộc về lớp c_2



Entropy – Ví dụ với 2 lớp

- S gồm 14 mẫu, trong đó 9 mẫu thuộc về lớp c_1 và 5 mẫu thuộc về lớp c_2
- Entropy của tập S đối với phân lớp có 2 lớp này là:

$$\text{Entropy}(S) = -(9/14).\log_2(9/14) - (5/14).\log_2(5/14) = 0.94$$

Information gain (IG)

- Information Gain của một biến đối với một tập các mẫu:
  - Là mức độ giảm về Entropy
  - Bởi việc phân tách (partitioning) các mẫu theo các giá trị của biến đó

  Ý nghĩa của Gain(S, A): Số lượng bits giảm được (reduced) đối với việc mã hóa lớp của một phần tử được lấy ra ngẫu nhiên từ tập S, khi biết giá trị của biến A

Information gain

- Information Gain của biến A đối với tập S:

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

trong đó Values(A) là tập các giá trị có thể của biến A, và

$$S_v = \{ x \mid x \in S, x_A = v \}$$

- Trong công thức trên, thành phần thứ 2 thể hiện giá trị Entropy sau khi tập S được phân chia bởi các giá trị của biến A

Tập dữ liệu Weather

Xét tập dữ liệu Weather ghi lại những ngày mà một người chơi (không chơi) tennis:

Day	Outlook	Temperature	Humidity	Windy	Play Tennis
D1	Sunny	Hot	High	FALSE	No
D2	Sunny	Hot	High	TRUE	No
D3	Overcast	Hot	High	FALSE	Yes
D4	Rain	Mild	High	FALSE	Yes
D5	Rain	Cool	Normal	FALSE	Yes
D6	Rain	Cool	Normal	TRUE	No
D7	Overcast	Cool	Normal	TRUE	Yes
D8	Sunny	Mild	High	FALSE	No
D9	Sunny	Cool	Normal	FALSE	Yes
D10	Rain	Mild	Normal	FALSE	Yes
D11	Sunny	Mild	Normal	TRUE	Yes
D12	Overcast	Mild	High	TRUE	Yes
D13	Overcast	Hot	Normal	FALSE	Yes
D14	Rain	Mild	High	TRUE	No

[Mitchell, 1997]

Các bước thực hiện

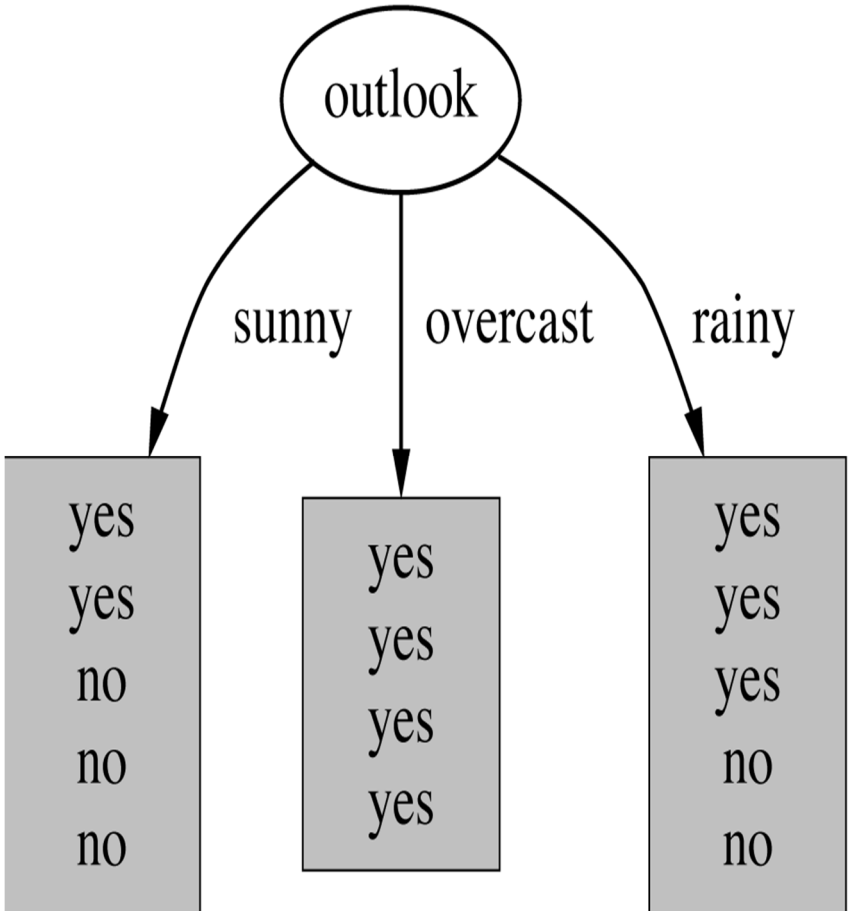
1. Tính toán entropy cho tập dữ liệu.
2. Với tất cả các thuộc tính:
   - Tính toán entropy của tất cả giá trị.
   - Tính entropy trung bình cho thuộc tính đang thực hiện.
   - Chọn đặc trưng có IG cao nhất.
   - Lặp lại cho đến khi thu được cây như mong muốn.

Tính toán entropy cho tập dữ liệu

Tập dữ liệu Weather gồm 14 mẫu, trong đó 9 mẫu thuộc về lớp Yes và 5 mẫu thuộc về lớp No

$$\text{Entropy}(S) = -(9/14).\log_2(9/14) - (5/14).\log_2(5/14) = \mathbf{0.94}$$

Biến Outlook



Entropy của mỗi tập con bị tách do biến Outlook

- "Outlook" = "Sunny"

$$\text{info}([2,3]) = \text{entropy}(2/5,3/5) = -2/5\log(2/5) - 3/5\log(3/5) = 0.971 \text{ bits}$$

- "Outlook" = "Overcast"

$$\text{info}([4,0]) = \text{entropy}(1, 0) = -1\log(1) - 0\log(0) = 0 \text{ bits}$$

Chú ý: log(0) không xác định, tuy nhiên ta tính quy ước 0*log(0) là 0

- "Outlook" = "Rainy"

$$\text{info}([3,2]) = \text{entropy}(3/5,2/5) = -3/5\log(3/5) - 2/5\log(2/5) = 0.971 \text{ bits}$$

- Thông tin kỳ vọng của biến Outlook:

$$\text{info}([3,2],[4,0],[3,2]) = (5/14)\times 0.971 + (4/14)\times 0 + (5/14)\times 0.971 = 0.693 \text{ bits}$$

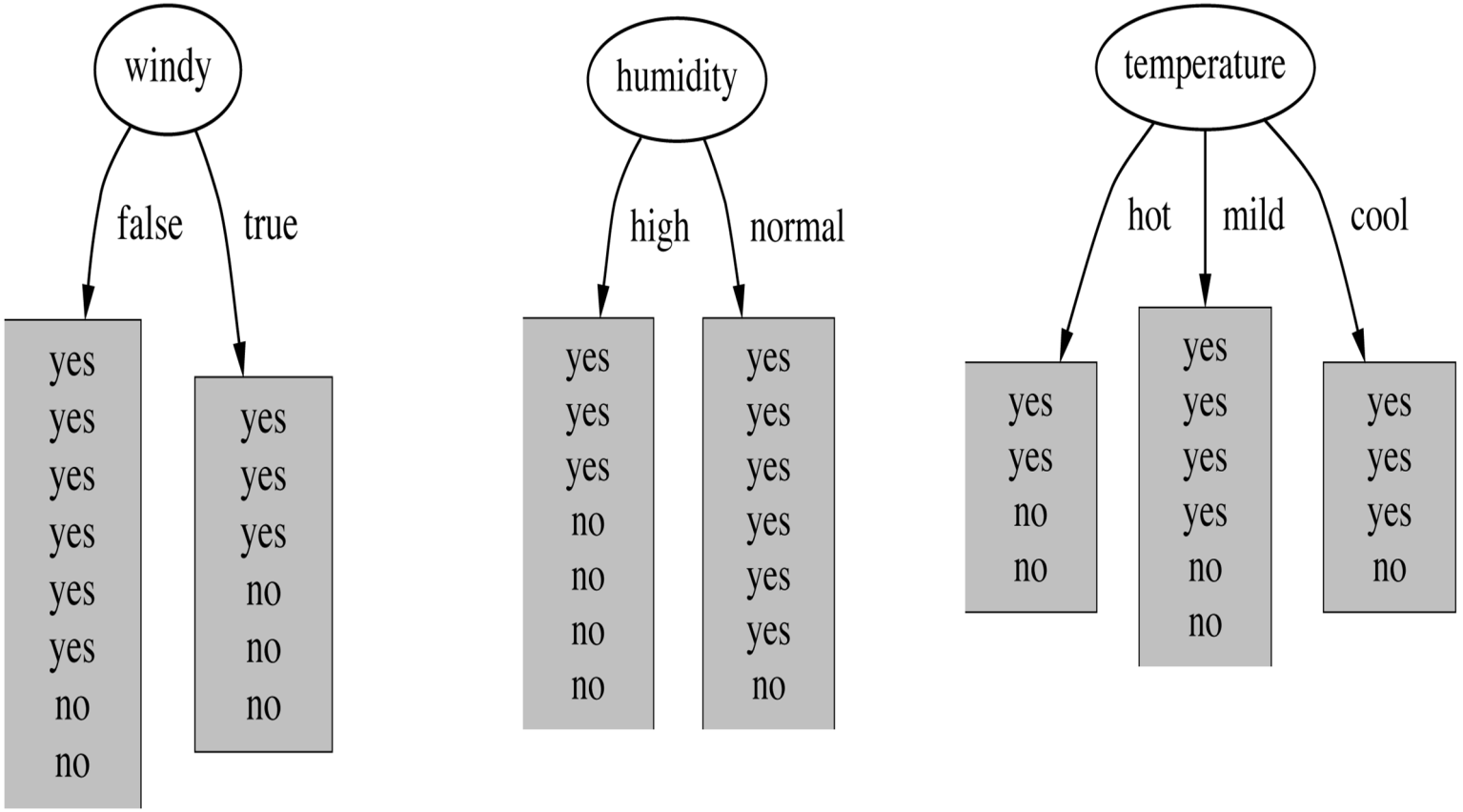
Tính Information Gain

- Information gain=(thông tin trước khi tách) – (thông tin sau khi tách)

$$\text{Gain(S, Outlook)} = \text{info([9,5])-info([2,3],[4,0],[3,2])} = 0.940\text{-}0.693$$
$$= 0.247 \text{ bits}$$

$$\textit{Entropy(S)} = -(9/14).\log_2(9/14) - (5/14).\log_2(5/14) \approx 0.94$$

Weather-Tìm các khả năng tách



Biến Windy

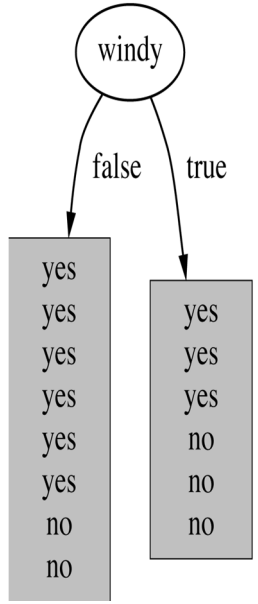
Hãy tính giá trị Information Gain của biến `Windy` đối với tập học `S` – `Gain(S,Windy)`?

Biến `Windy` có 2 giá trị có thể: `False` và `True`

`S` = {9 mẫu lớp `Yes` và 5 mẫu lớp `No`}

S_{False} = {6 mẫu lớp `Yes` và 2 mẫu lớp `No` có giá trị `Windy=False`}

S_{True} = {3 mẫu lớp `Yes` và 3 mẫu lớp `No` có giá trị `Windy=True`}



$$Gain(S,Windy) = Entropy(S) - \sum_{v \in \{False,True\}} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$= Entropy(S) - (8/14).Entropy(S_{False}) - (6/14).Entropy(S_{True})$$

$$= 0.94 - (8/14).(0.81) - (6/14).(1) = \underline{\mathbf{0.048}} \text{ bits}$$

$Entropy(S) = -(9/14).\log_2(9/14) - (5/14).\log_2(5/14) \approx 0.94$

Tính Information Gain

- Tương tự, ta tính được Information gain cho các biến trong tập dữ liệu weather:

$$\text{Gain}(S, \text{Outlook}) = 0.247 \text{ bits}$$

$$\text{Gain}(S, \text{Humidity}) = 0.152 \text{ bits}$$

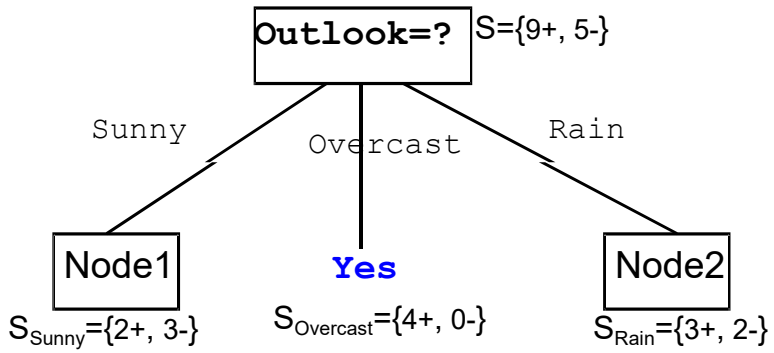
$$\text{Gain}(S, \text{Temperature}) = 0.029 \text{ bits}$$

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

- Vậy Outlook là biến được chọn để kiểm tra cho nút gốc vì có Information Gain cao nhất

Xây dựng cây

→`Outlook` được chọn là biến kiểm tra tại nút gốc!



Tiếp tục tách nút

- Tại nút Node1, biến nào trong số {Temperature, Humidity, Windy} nên được chọn là biến kiểm tra?

Lưu ý: Biến Outlook bị loại ra, bởi vì nó đã được sử dụng bởi cha của nút Node1 (là nút gốc)

Tập dữ liệu Weather

Outlook = Sunny

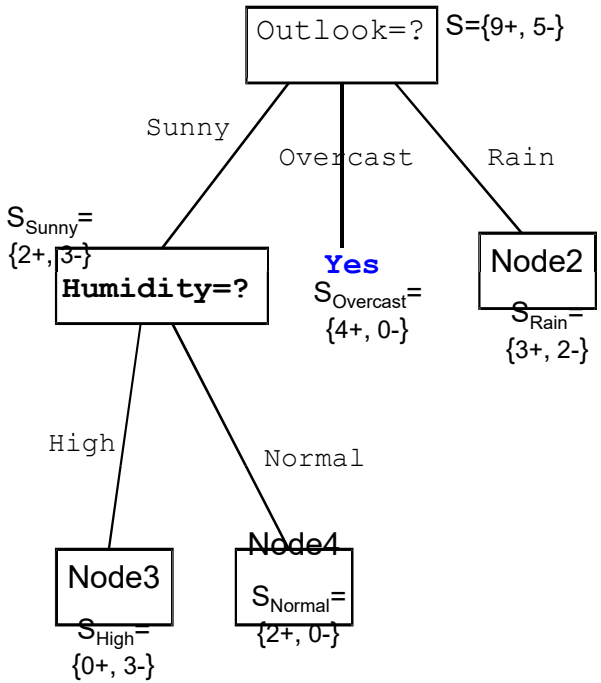
Day	Outlook	Temperature	Humidity	Windy	Play Tennis
D1	Sunny	Hot	High	FALSE	No
D2	Sunny	Hot	High	TRUE	No
D8	Sunny	Mild	High	FALSE	No
D9	Sunny	Cool	Normal	FALSE	Yes
D11	Sunny	Mild	Normal	TRUE	Yes

[Mitchell, 1997]

Tiếp tục tách nút

- Tại nút Node1, biến nào trong số {Temperature,Humidity, Windy} nên được chọn là biến kiểm tra?
  - **Lưu ý!** Biến Outlook bị loại ra, bởi vì nó đã được sử dụng bởi cha của nút Node1 (là nút gốc)
  - Gain(S_{Sunny}, Temperature) =...= 0.57
  - Gain(S_{Sunny}, **Humidity**) = ... = **0.97**
  - Gain(S_{Sunny}, Windy) = ... = 0.019

→Vì vậy, Humidity được chọn là biến kiểm tra cho nút Node1!

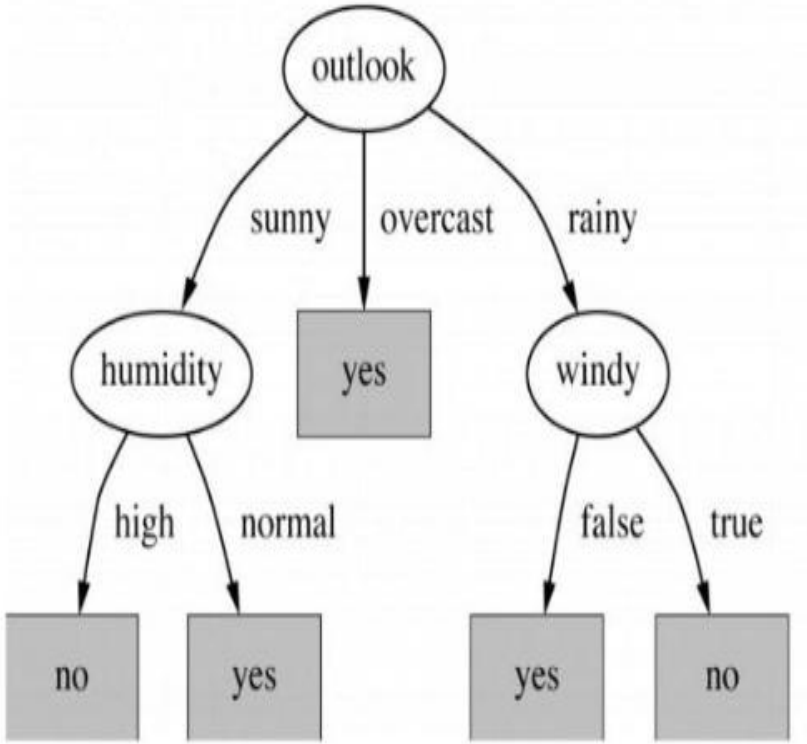


Điều kiện dừng

- Lượng dữ liệu của 1 nút được gán hầu hết vào 1 lớp
  - **vd: >90%**
- Số lượng mẫu trong tập con tại nút nhỏ hơn 1 giá trị cho trước – ngưỡng (threshold)
- Giảm được Information gain
- Các biến đều đã được kiểm tra

Cây quyết định dựng được

Final decision tree



Vấn đề trong ID3

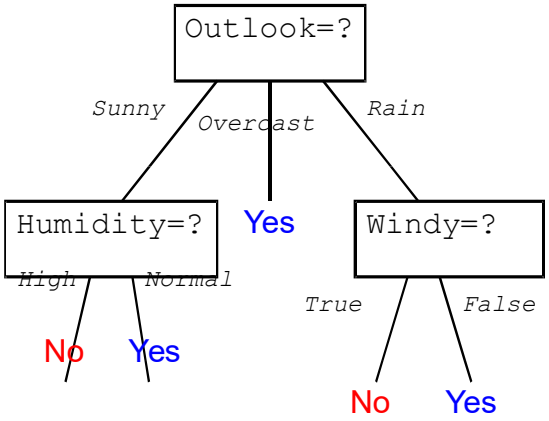
- Cây quyết định học được quá khớp (over-fit) với các mẫu
- Chưa xử lý các biến có kiểu giá trị liên tục (kiểu số thực)

Over-fitting trong học cây quyết định

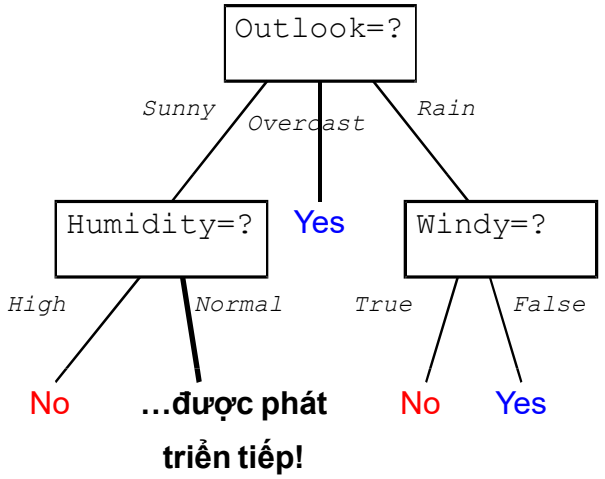
- Một cây quyết định phù hợp hoàn hảo đối với tập huấn luyện có phải là giải pháp tối ưu?
- Nếu như tập huấn luyện có nhiễu/lỗi…?

Vd: Một mẫu nhiễu/lỗi (Mẫu thực sự mang nhãn Yes, nhưng *bị gán nhãn nhầm là No*):

(Outlook=Sunny, Temperature=Hot, Humidity=Normal, Windy=True, PlayTennis=**No**)

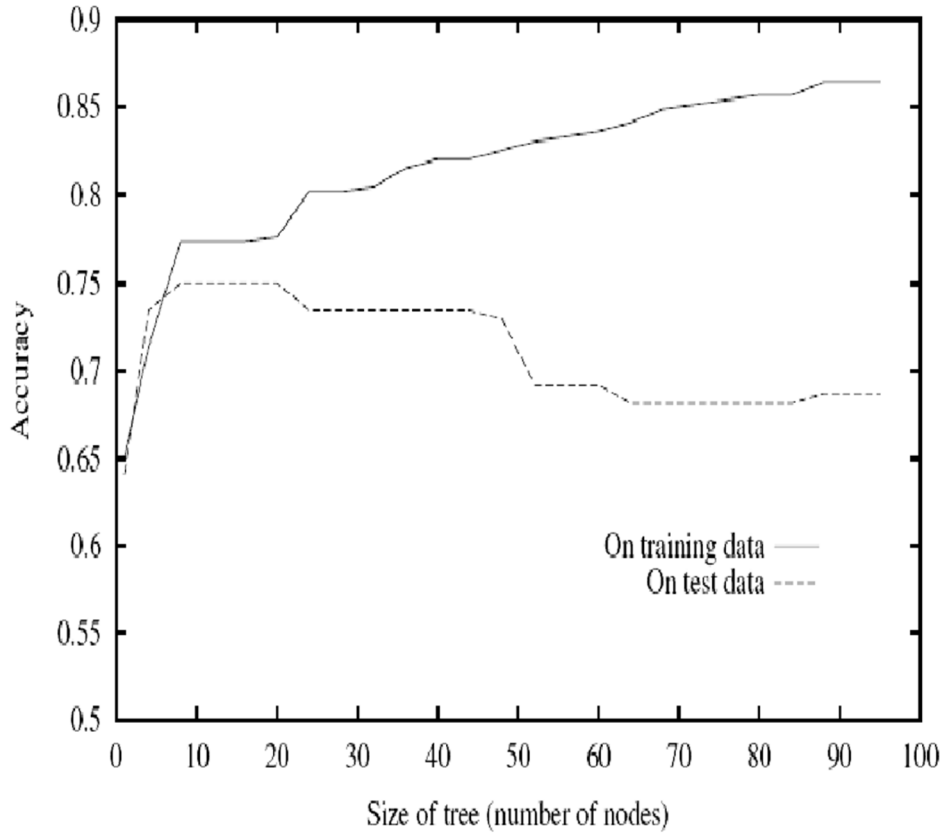


Học được một cây quyết định *phức tạp hơn*!
(chỉ bởi vì mẫu nhiễu/lỗi)



Over-fitting trong học cây quyết định

Tiếp tục quá trình học cây quyết định sẽ làm giảm độ chính xác đối với tập thử nghiệm mặc dù tăng độ chính xác đối với tập huấn luyện



[*Mitchell, 1997*]

Giải quyết vấn đề over-fitting

Hai chiến lược

- Ngừng việc học (phát triển) cây quyết định sớm hơn, trước khi nó đạt tới cấu trúc cây cho phép phân loại (khớp) hoàn hảo tập huấn luyện
- Học (phát triển) cây đầy đủ (tương ứng với cấu trúc cây hoàn toàn phù hợp đối với tập huấn luyện), và sau đó thực hiện quá trình tỉa (to post-prune) cây

Giải quyết vấn đề over-fitting

- Chiến lược tỉa cây đầy đủ (Post-pruning over-fit trees) thường cho hiệu quả tốt hơn trong thực tế

  → Lý do: Chiến lược "ngừng sớm" việc học cây cần phải đánh giá chính xác được *khi nào nên ngừng việc học* (phát triển) cây – Khó xác định!

Các thuộc tính có giá trị liên tục

- Cần xác định (chuyển đổi thành) các thuộc tính có giá trị rời rạc, bằng cách chia khoảng giá trị liên tục thành một tập các khoảng (intervals) không giao nhau
- Đối với thuộc tính (có giá trị liên tục) A, tạo một thuộc tính mới kiểu nhị phân A_v sao cho: A_v là đúng nếu A>v, và là sai nếu ngược lại
- Làm thế nào để xác định giá trị ngưỡng v "tốt nhất"?

  → Chọn giá trị ngưỡng v giúp sinh ra giá trị *Information Gain* cao nhất

Các thuộc tính có giá trị liên tục

Ví dụ: Giả sử thuộc tính Temperature là liên tục

- Sắp xếp các mẫu theo giá trị tăng dần đối với thuộc tính Temperature
- Xác định các mẫu liền kề nhưng khác phân lớp
- Có 2 giá trị ngưỡng có thể: $Temperature_{54}$ và $Temperature_{85}$
- Thuộc tính mới kiểu nhị phân $Temperature_{54}$ được chọn, bởi vì:

$Gain(S, Temperature_{54}) > Gain(S, Temperature_{85})$

Temperature	40	48	60	72	80	90
PlayTennis	No	No	Yes	Yes	Yes	No

Ví dụ khác về DT

Attributes				Classes
Gender	Car ownership	Travel Cost ($)/km	Income Level	**Transportation mode**
Male	0	Cheap	Low	**Bus**
Male	1	Cheap	Medium	**Bus**
Female	1	Cheap	Medium	**Train**
Female	0	Cheap	Low	**Bus**
Male	1	Cheap	Medium	**Bus**
Male	0	Standard	Medium	**Train**
Female	1	Standard	Medium	**Train**
Female	1	Expensive	High	**Car**
Male	2	Expensive	Medium	**Car**
Female	2	Expensive	High	**Car**

Information gain

- Information Gain của biến A đối với tập S:

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

trong đó

$$Entropy(S) = \sum_{i=1}^{c} - p_i . \log_2 p_i$$

Values(A) là tập các giá trị có thể của biến A, và

$$S_v = \{ x \mid x \in S, x_A = v \}$$

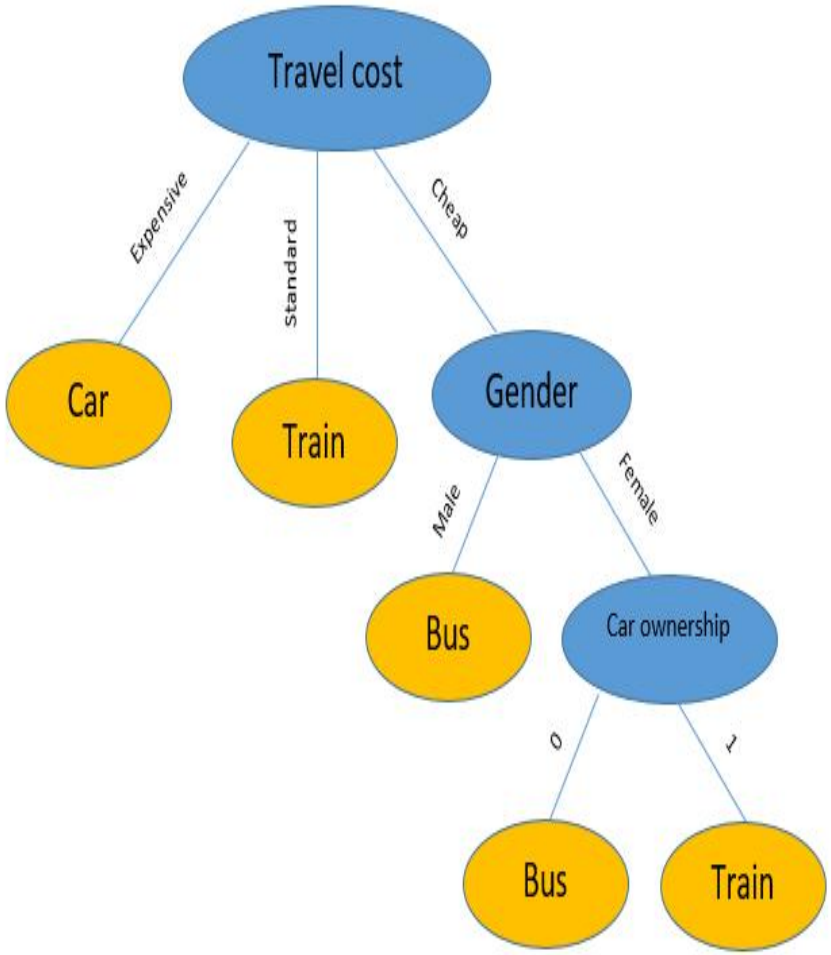
- Trong công thức trên, thành phần thứ 2 thể hiện giá trị Entropy sau khi tập S được phân chia bởi các giá trị của biến A

Xác định nút gốc

Gain(Income)	**0.70**
Gain(Gender)	**0.13**
Gain(CarOwner)	**0.54**
Gain(TravelCost)	**1.218**

CÂY QUYẾT ĐỊNH NHẬN ĐƯỢC

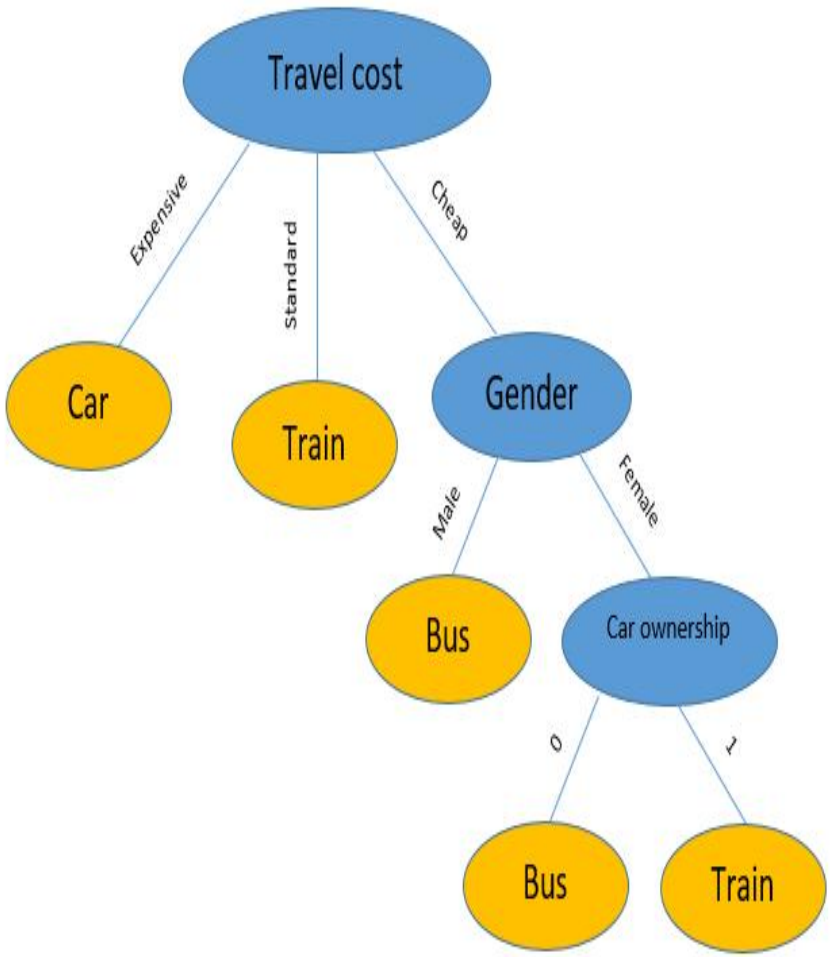
Thực hiện các bước tương tự như ví dụ trên ta nhận được cây quyết định



Sử dụng cây quyết định trong dự đoán lớp của các dữ liệu chưa biết

Person name	Gender	Car ownership	Travel Cost ($)/km	Income Level	Transportation Mode
Alex	Male	1	Standard	High	?
Buddy	Male	0	Cheap	Medium	?
Cherry	Female	1	Cheap	High	?

Trích rút luật từ cây quyết định



Rule 1 *: If Travel cost/km is expensive then mode = car*

Rule 2 *: If Travel cost/km is standard then mode = train*

Rule 3 *: If Travel cost/km is cheap and gender is male then mode = bus*

Rule 4 *: If Travel cost/km is cheap and gender is female and she owns no car then mode = bus*

Rule 5 *: If Travel cost/km is cheap and gender is female and she owns 1 car then mode = train*

Trích rút luật từ cây quyết định

Rule 1 *: If Travel cost/km is expensive then mode = car*

Rule 2 *: If Travel cost/km is standard then mode = train*

Rule 3 *: If Travel cost/km is cheap and gender is male then mode = bus*

Rule 4 *: If Travel cost/km is cheap and gender is female and she owns no car then mode = bus*

Rule 5 *: If Travel cost/km is cheap and gender is female and she owns 1 car then mode = train*

Sử dụng cây quyết định trong dự đoán lớp của các dữ liệu chưa biết

Person name	Gender	Car ownership	Travel Cost ($)/km	Income Level	Transportation Mode
Alex	Male	1	Standard	High	?
Buddy	Male	0	Cheap	Medium	?
Cherry	Female	1	Cheap	High	?

Áp dụng các luật để ra quyết định

- Alex có giá trị của thuộc tính **Travel Cost/Km** là **Standard** nên sẽ chọn phương tiện là **Train (Rule 2)**
- Buddy có giá trị của thuộc tính **Travel Cost/Km** là **Cheap** nên phải xét thêm thuộc tính Gender. **Gender** của anh ta là **Male** nên anh ta sẽ chọn **Bus (Rule 3).**
- Cheery cũng có giá trị thuộc tính **Travel Cost/Km** là **Cheap** nhưng **Gender** là **Female nên phải xét thêm thuộc tính Car Ownership và giá trị thuộc tính này là 1** nên theo **Rule 5** cô ta sẽ chọn phương tiện là **Train.**

Kết quả

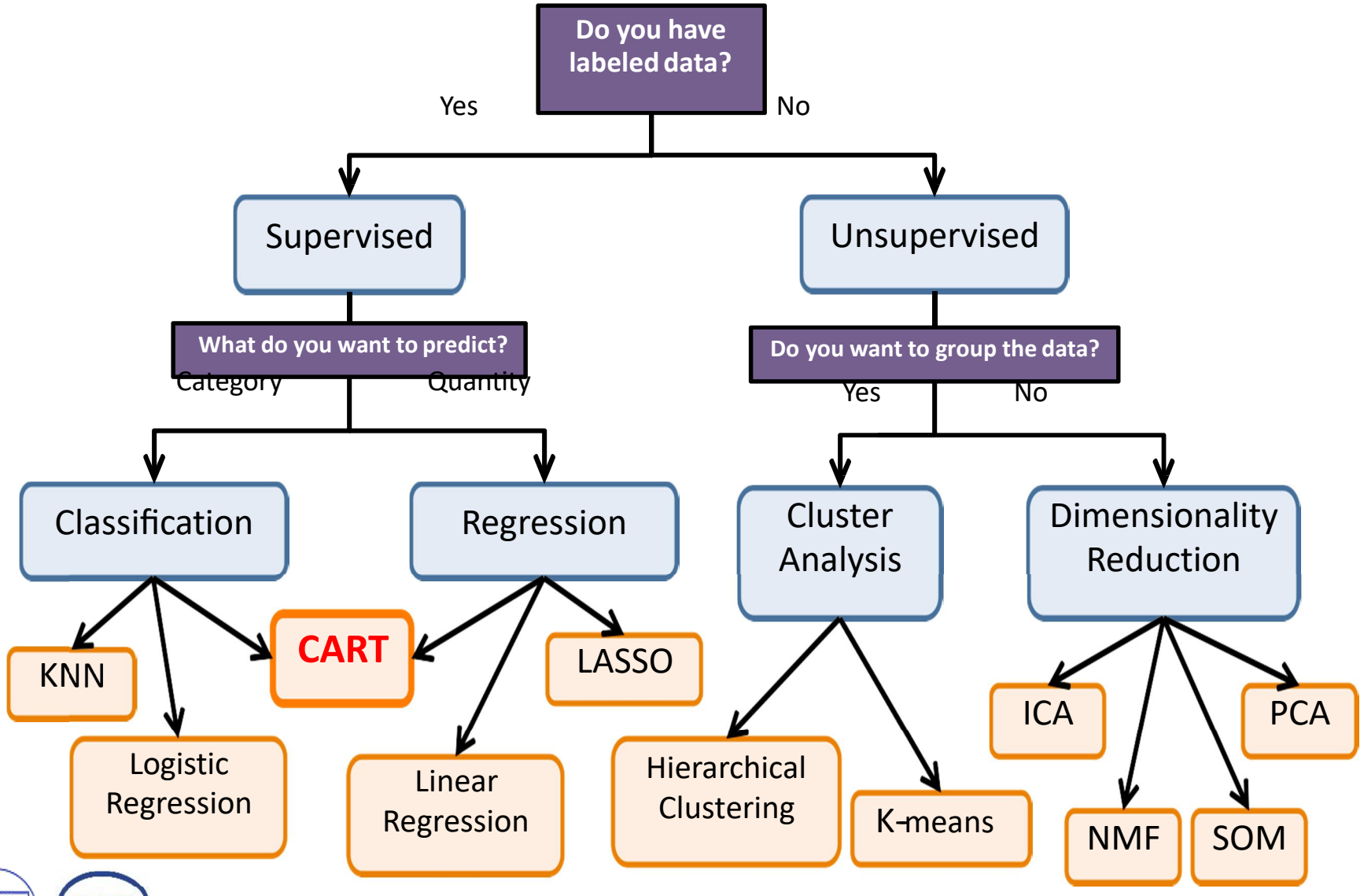
Person name	Gender	Car ownership	Travel Cost ($)/km	Income Level	Transportation Mode
Alex	Male	1	Standard	High	**Train**
Buddy	Male	0	Cheap	Medium	**Bus**
Cherry	Female	1	Cheap	High	**Train**

Cây phân loại và hồi quy

Classification **a**nd **R**egression **T**rees

(CART)

Các giải thuật Học máy



Mô hình cây (tree)

Xây dựng cây CART thế nào?

Có 2 dạng:

1. Hồi quy
2. Phân loại (lớp)

Cây phân loại (phân lớp) và hồi quy

- Cây **hồi quy** được dùng cho các biến đầu ra **liên tục**
- Cây **phân lớp** dùng cho các biến đầu ra **rời rạc**
- Cả hai loại cây này đều **chia biến dự đoán thành các vùng phân biệt không giao nhau**
- Quá trình tách được thực hiện cho đến khi **thỏa mãn điều kiện dừng**

CÂY PHÂN LỚP

$$\text{class } k(m) = \arg\max_k \hat{p}_{mk}$$

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

Mô hình liên tục từng đoạn

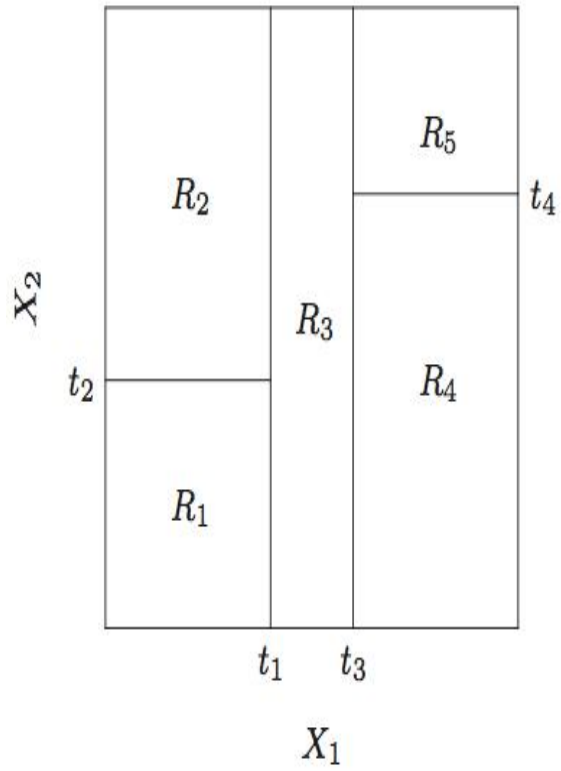
(piecewise continuous)

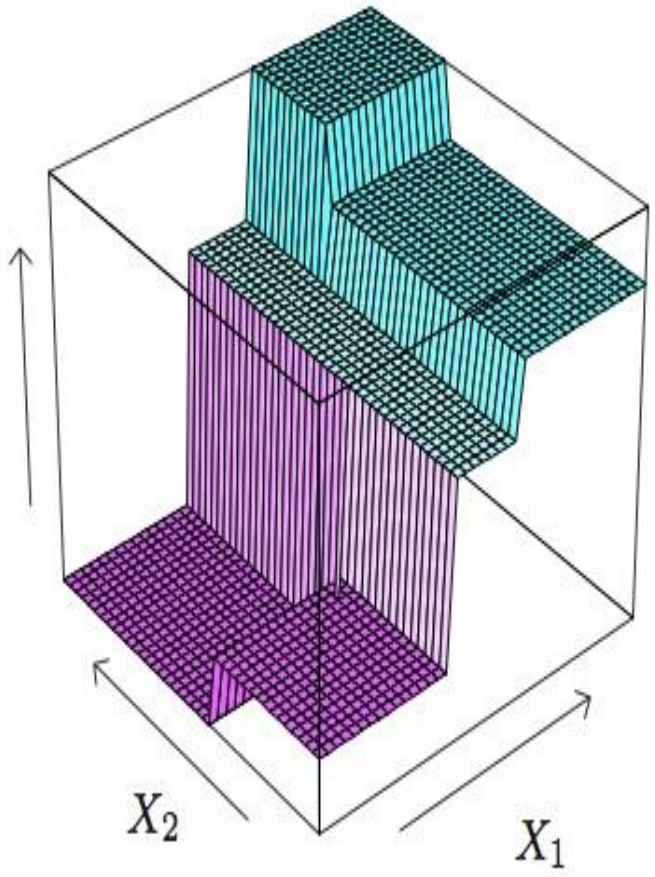
- Dự đoán liên tục trong mỗi vùng

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

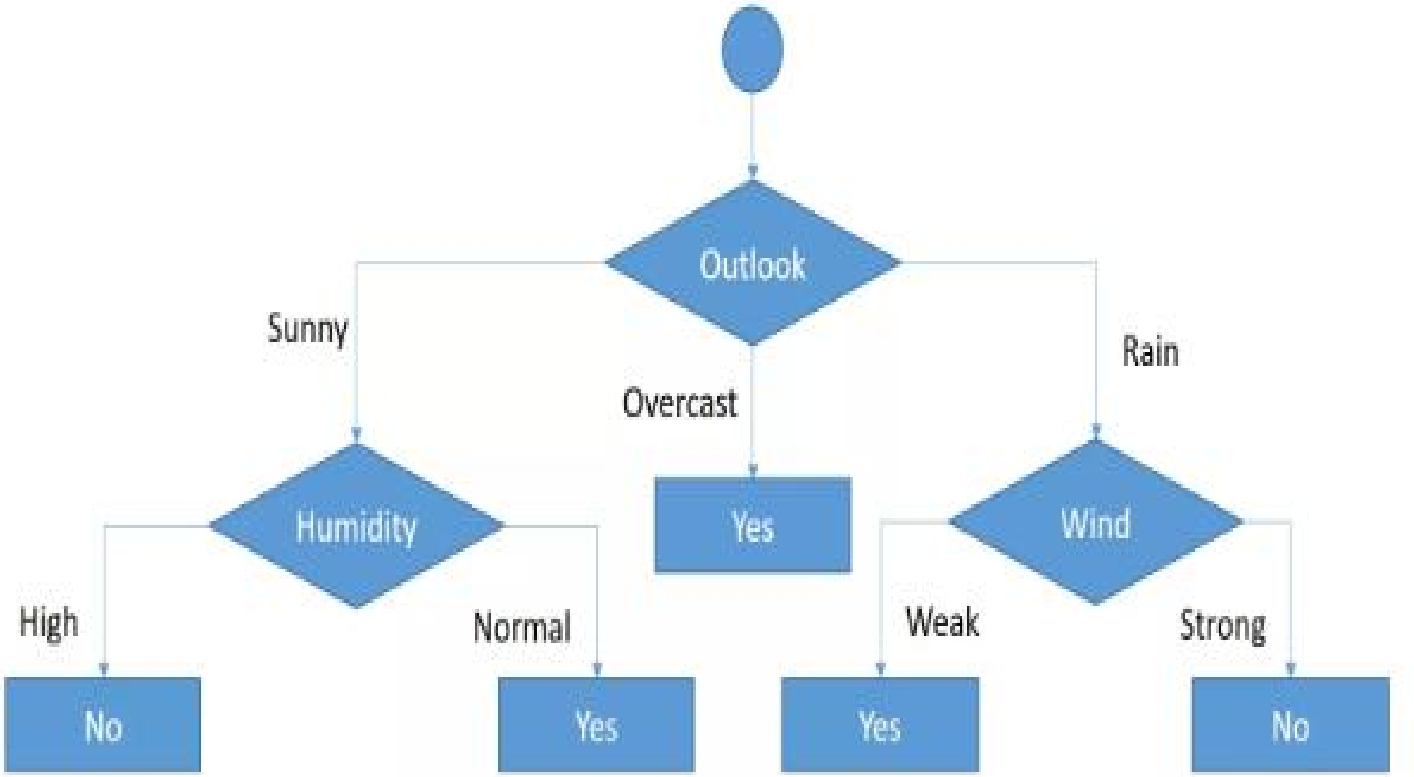
Mô hình liên tục từng đoạn





Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

Minh họa cây CART



Các bước xây dựng cây CART

- **Chỉ số Gini**: được dùng để xác định thuộc tính phân tách

$$\text{Gini}(D_i) = 1 - \sum_{i=1}^{m} (P_i)^2, \; i = 1, \ldots, m$$

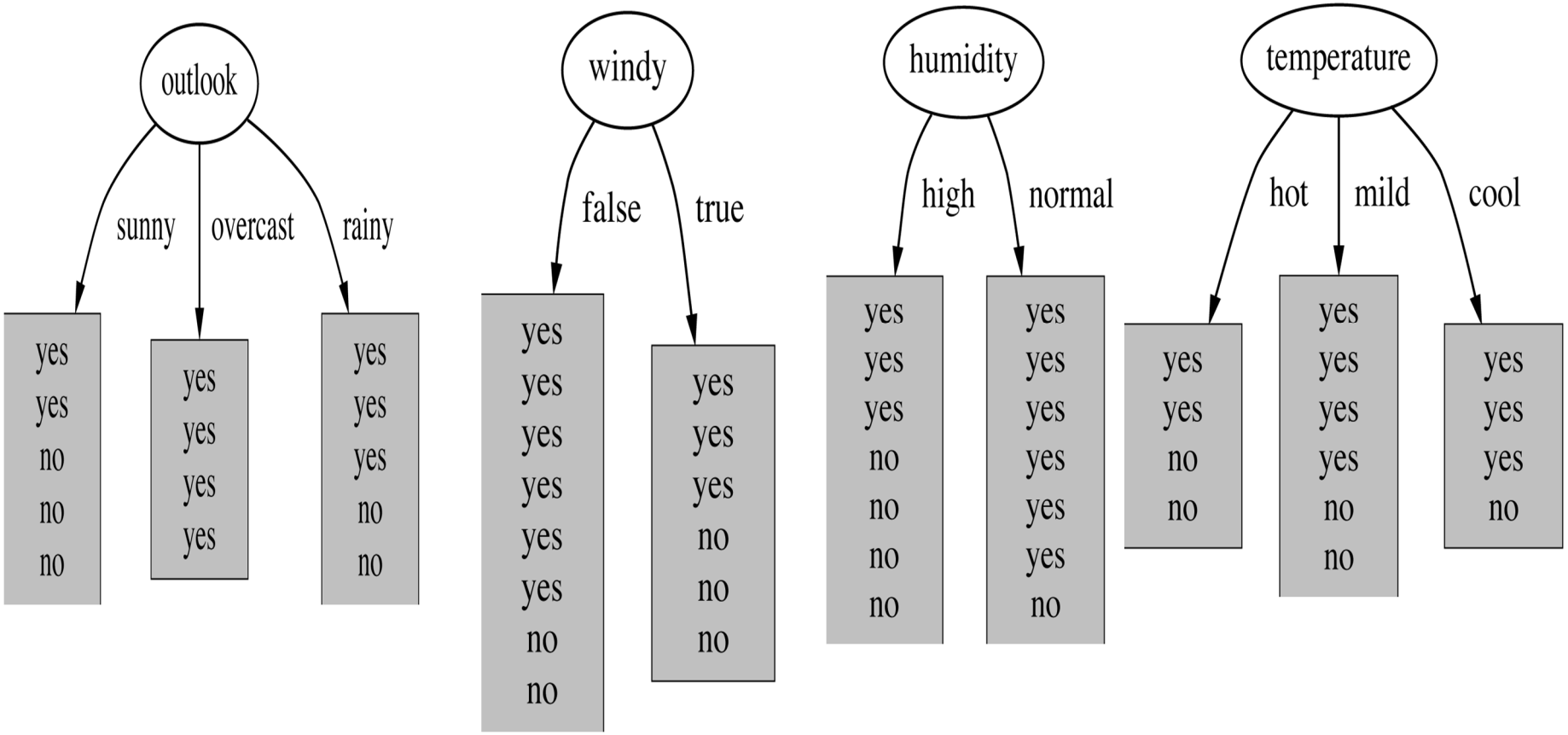
$$\mathbf{Gini_A(D)} = \frac{|D_1| * \text{Gini}(D_1)}{|D|} + \frac{|D_2| * \text{Gini}(D_2)}{|D|} + \cdots, \forall A$$

Ví dụ: Tập dữ liệu Weather

Day	Outlook	Temperature	Humidity	Windy	Play Tennis
D1	Sunny	Hot	High	FALSE	No
D2	Sunny	Hot	High	TRUE	No
D3	Overcast	Hot	High	FALSE	Yes
D4	Rain	Mild	High	FALSE	Yes
D5	Rain	Cool	Normal	FALSE	Yes
D6	Rain	Cool	Normal	TRUE	No
D7	Overcast	Cool	Normal	TRUE	Yes
D8	Sunny	Mild	High	FALSE	No
D9	Sunny	Cool	Normal	FALSE	Yes
D10	Rain	Mild	Normal	FALSE	Yes
D11	Sunny	Mild	Normal	TRUE	Yes
D12	Overcast	Mild	High	TRUE	Yes
D13	Overcast	Hot	Normal	FALSE	Yes
D14	Rain	Mild	High	TRUE	No

[Mitchell, 1997]

Weather-Tìm các khả năng tách



Xét thuộc tính Outlook

	Yes	No	Number of instances	$\|D_i\|/\|D\|$
D_1=Sunny	2	3	5	5/14
D_2=Overcast	4	0	4	4/14
D_3=Rain	3	2	5	5/14

Gini(Outlook) = 5/14*Gini(Outlook=Sunny) + 4/14*Gini(Outlook=Overcast) + 5/14*Gini(Outlook=Rain)

Xét thuộc tính Outlook

	Yes	No	Number of instances	$\|D_i\|/\|D\|$
Sunny	**2**	**3**	**5**	5/14
Overcast	4	0	4	4/14
Rain	3	2	5	5/14

Gini(**Outlook=Sunny**) = $1 - (2/5)^2 - (3/5)^2 = 1 - 0.16 - 0.36 = 0.48$

Gini(**Outlook=Overcast**) = $1 - (4/4)^2 - (0/4)^2 = 0$

Gini(**Outlook=Rain**) = $1 - (3/5)^2 - (2/5)^2 = 1 - 0.36 - 0.16 = 0.48$

Gini(Outlook) = (5/14) x **0.48** + (4/14) x **0** + (5/14) x **0.48** = 0.171 + 0 + 0.171 = **0.342**

Xét thuộc tính Temperature

	Yes	No	Number of instances	$\|D_i\|/\|D\|$
Hot	2	2	4	4/14
Cool	3	1	4	4/14
Mild	4	2	6	6/14

Gini(Temperature) =
4/14*Gini(Temperature=Hot) +
4/14*Gini(Temperature=Cold) +
5/14*Gini(Temperature=Mild)

Xét thuộc tính Temperature

	Yes	No	Number of instances	$\|D_i\|/\|D\|$
Hot	2	2	4	4/14
Cool	3	1	4	4/14
Mild	4	2	6	6/14

Gini(Temp=Hot) = $1 - (2/4)^2 - (2/4)^2 = 0.5$

Gini(Temp=Cool) = $1 - (3/4)^2 - (1/4)^2 = 1 - 0.5625 - 0.0625 = 0.375$

Gini(Temp=Mild) = $1 - (4/6)^2 - (2/6)^2 = 1 - 0.444 - 0.111 = 0.445$

Gini(Temp) = (4/14) x 0.5 + (4/14) x 0.375 + (6/14) x 0.445 = 0.439

Tương tự với các thuộc tính Humidity và Wind

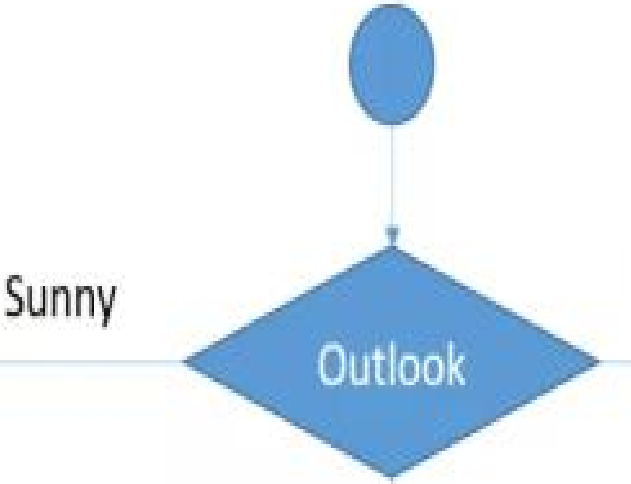
Gini(Humidity) = (7/14) x 0.489 + (7/14) x 0.244 = **0.367**

Gini(Wind) = (8/14) x 0.375 + (6/14) x 0.5 = **0.428**

Chọn thuộc tính có chỉ số Gini bé nhất làm nút gốc

Feature	Gini index
Outlook	**0.342**
Temperature	0.439
Humidity	0.367
Windy	0.428

Cây nhận được sau bước thứ nhất

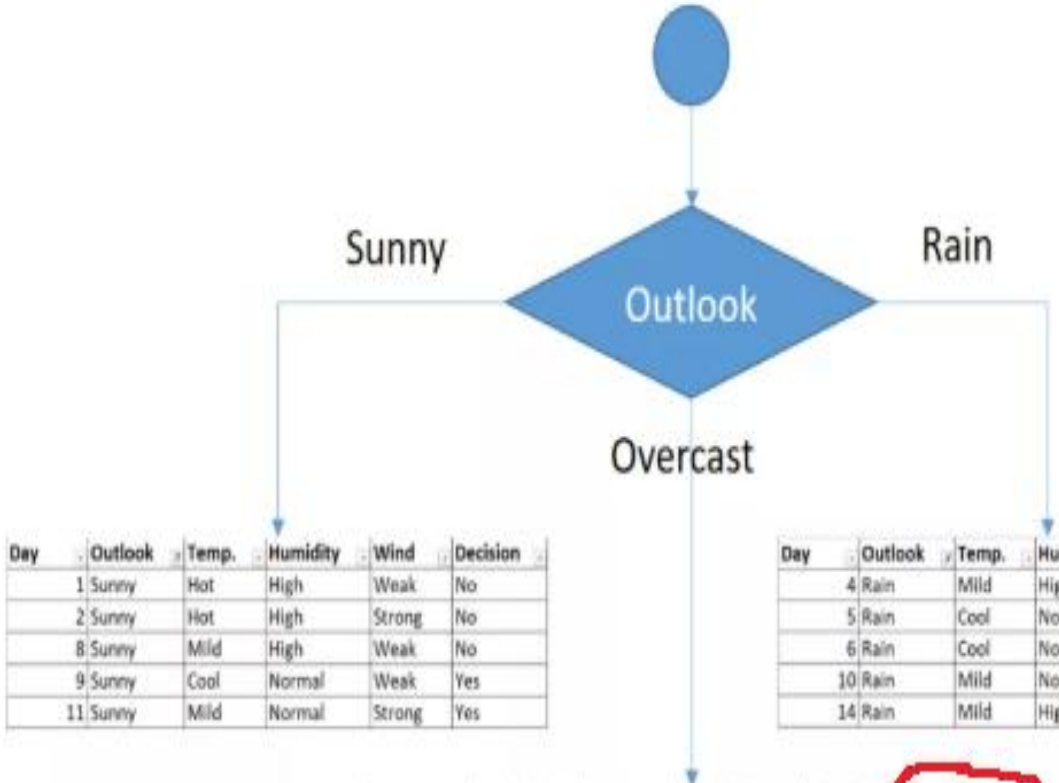


Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

Cây nhận được sau bước thứ nhất

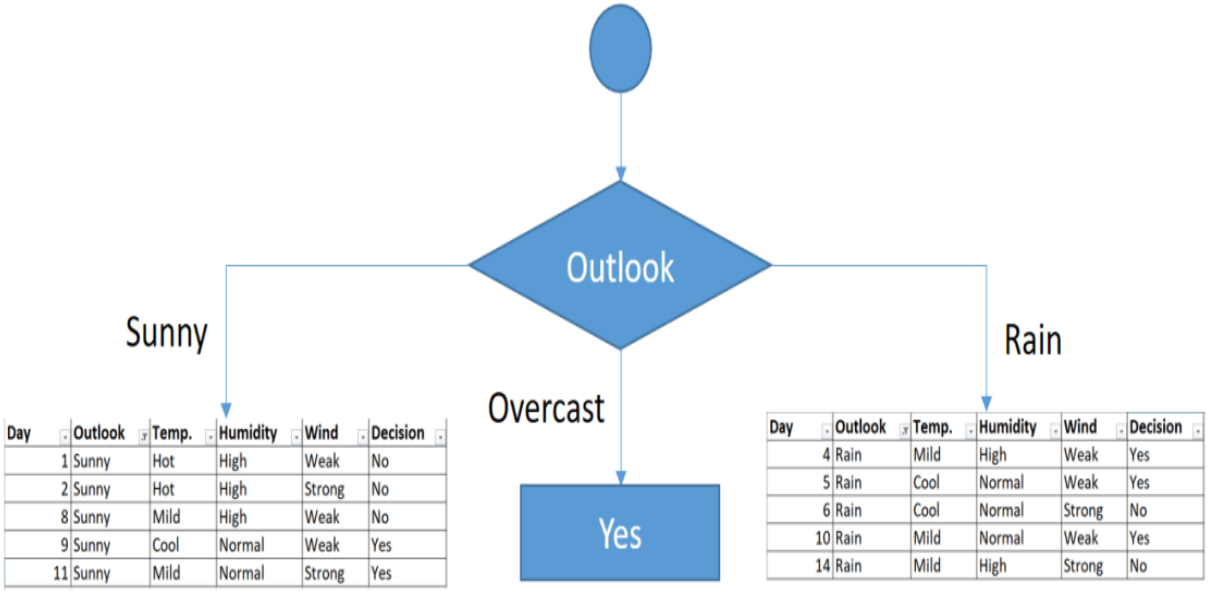


Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

Cây nhận được sau bước thứ nhất



Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Tính chỉ số Gini cho các nhánh

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Sunny Outlook

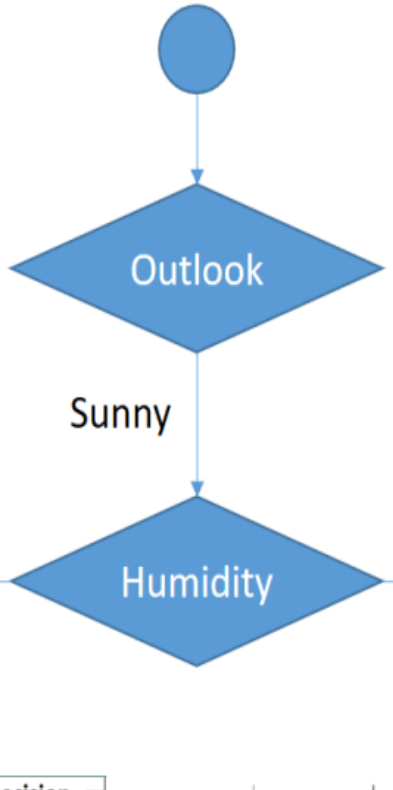
Day	Outlook	Temp.	Humidity	Windy	Decision
1	**Sunny**	Hot	High	Weak	No
2	**Sunny**	Hot	High	Strong	No
8	**Sunny**	Mild	High	Weak	No
9	**Sunny**	Cool	Normal	Weak	Yes
11	**Sunny**	Mild	Normal	Strong	Yes

Sunny Outlook

Temp.	Humidity	Windy
- Gini(Outlook=Sunny and Temp.=Hot) = $1 - (0/2)^2 - (2/2)^2 = 0$ - Gini(Outlook=Sunny and Temp.=Cool) = $1 - (1/1)^2 - (0/1)^2 = 0$ - Gini(Outlook=Sunny and Temp.=Mild) = $1 - (1/2)^2 - (1/2)^2 = 1 - 0.25 - 0.25 = 0.5$	- Gini(Outlook=Sunny and Humidity=High) = $1 - (0/3)^2 - (3/3)^2 = 0$ - Gini(Outlook=Sunny and Humidity=Normal) = $1 - (2/2)^2 - (0/2)^2 = 0$	- Gini(Outlook=Sunny and Wind=Weak) = $1 - (1/3)^2 - (2/3)^2 = 0.266$ - Gini(Outlook=Sunny and Wind=Strong) = $1-(1/2)^2 - (1/2)^2 = 0.2$

Gini(Outlook=Sunny and Temp.) = (2/5)x0 + (1/5)x0 + (2/5)x0.5 =**0.2**
Gini(Outlook=Sunny and **Humidity**) = (3/5)x0 + (2/5)x0 = **0**
Gini(Outlook=Sunny and Windy) = (3/5)x0.266 + (2/5)x0.2 = **0.466**

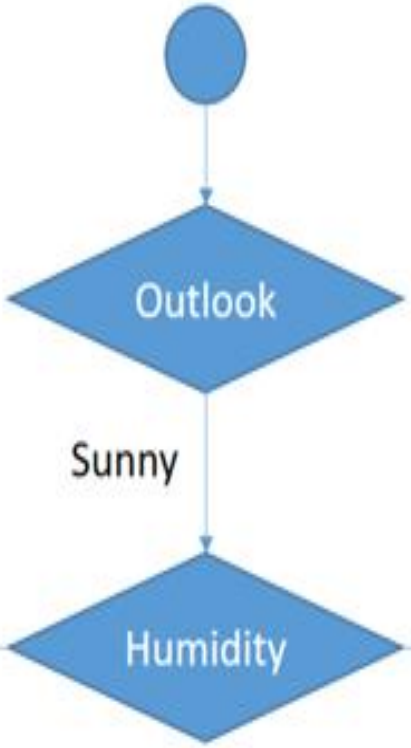
Sunny Outlook



Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No

Day	Outlook	Temp.	Humidity	Wind	Decision
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

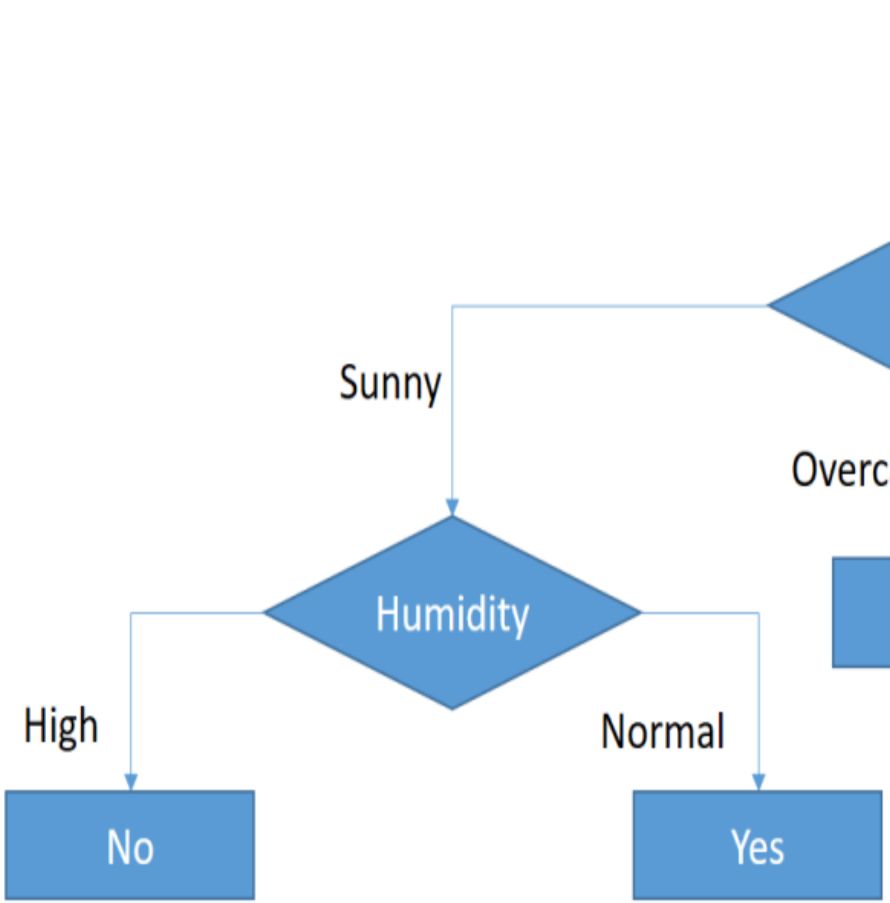
Sunny Outlook



Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No

Day	Outlook	Temp.	Humidity	Wind	Decision
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Kết quả



Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Rain Outlook

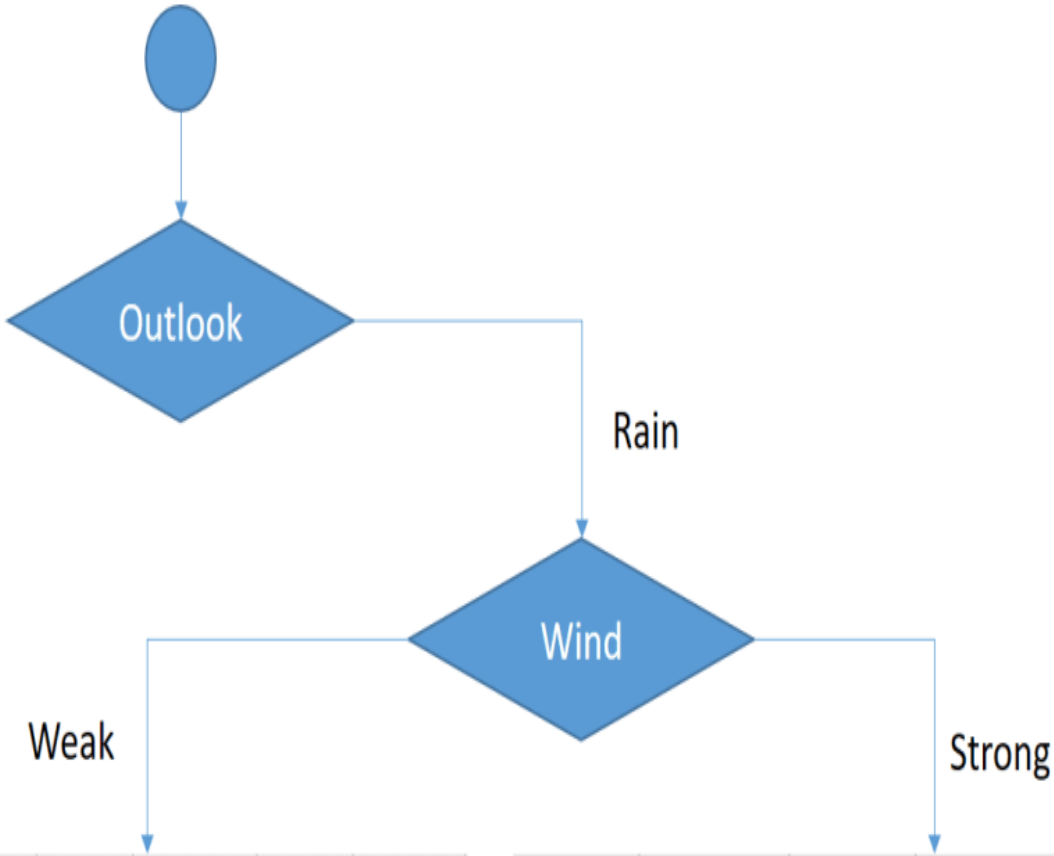
Day	Outlook	Temp.	Humidity	Windy	Decision
4	**Rain**	***Mild***	High	Weak	Yes
5	**Rain**	***Cool***	Normal	Weak	Yes
6	**Rain**	***Cool***	Normal	Strong	No
10	**Rain**	***Mild***	Normal	Weak	Yes
14	**Rain**	***Mild***	High	Strong	No

Rain Outlook

Temp.	Humidity	Windy
- Gini(Outlook=Rain and Temp.=Cool) = $1 - (1/2)^2 - (1/2)^2 = 0.5$ - Gini(Outlook=Rain and Temp.=Mild) = $1 - (2/3)^2 - (1/3)^2 = 0.444$	- Gini(Outlook=Rain and Humidity=High) = $1 - (1/2)^2 - (1/2)^2 = 0.5$ - Gini(Outlook=Rain and Humidity=Normal) = $1 - (2/3)^2 - (1/3)^2 = 0.444$	- Gini(Outlook=Rain and Wind=Weak) = $1 - (3/3)^2 - (0/3)^2 = 0$ - Gini(Outlook=Rain and Wind=Strong) = $1 - (0/2)^2 - (2/2)^2 = 0$

Gini(Outlook=Rain and Temp.) = (2/5)x0.5 + (3/5)x0.444 = 0.466
Gini(Outlook=Rain and Humidity) = (2/5)x0.5 + (3/5)x0.444 = 0.466
Gini(Outlook=Rain and **Windy**) = (3/5)x0 + (2/5)x0 = **0**

Kết quả trên Rain Outlook



Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes

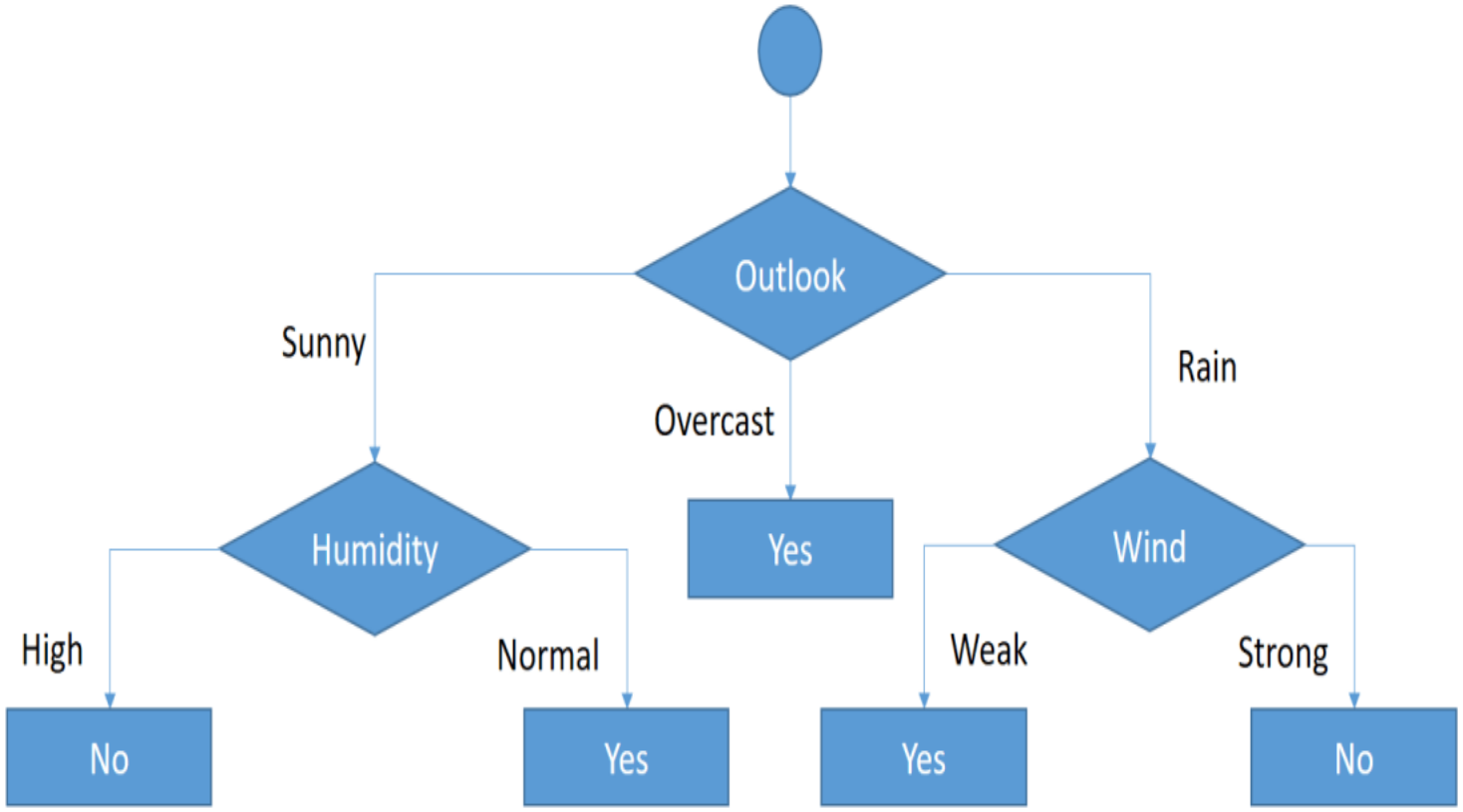
Day	Outlook	Temp.	Humidity	Wind	Decision
6	Rain	Cool	Normal	Strong	No
14	Rain	Mild	High	Strong	No

Kết quả trên Rain Outlook

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
6	Rain	Cool	Normal	Strong	No
14	Rain	Mild	High	Strong	No

Cây phân loại nhận được



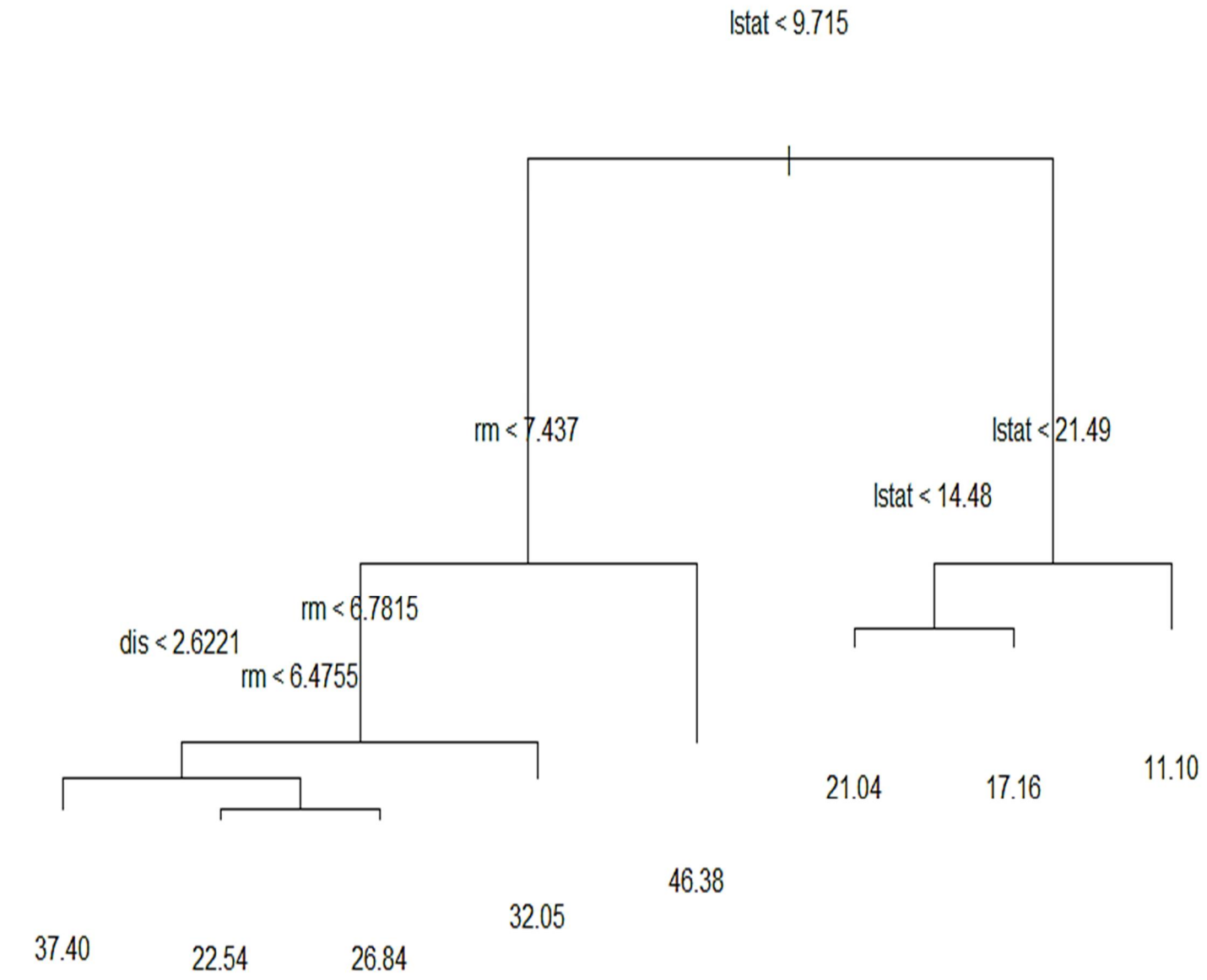
CÂY HỒI QUY

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

$$\hat{c}_m = \text{ave}(y_i | x_i \in R_m)$$

Giá trị dự đoán được lưu tại lá của cây hồi quy. Nó được tính bằng giá trị trung bình của tất cả các mẫu (bản ghi) tại lá đó.

Cây hồi quy


lstat < 9.715
rm < 7.437
lstat < 21.49
lstat < 14.48
rm < 6.7815
dis < 2.6221
rm < 6.4755
21.04
17.16
11.10
46.38
32.05
37.40
22.54
26.84

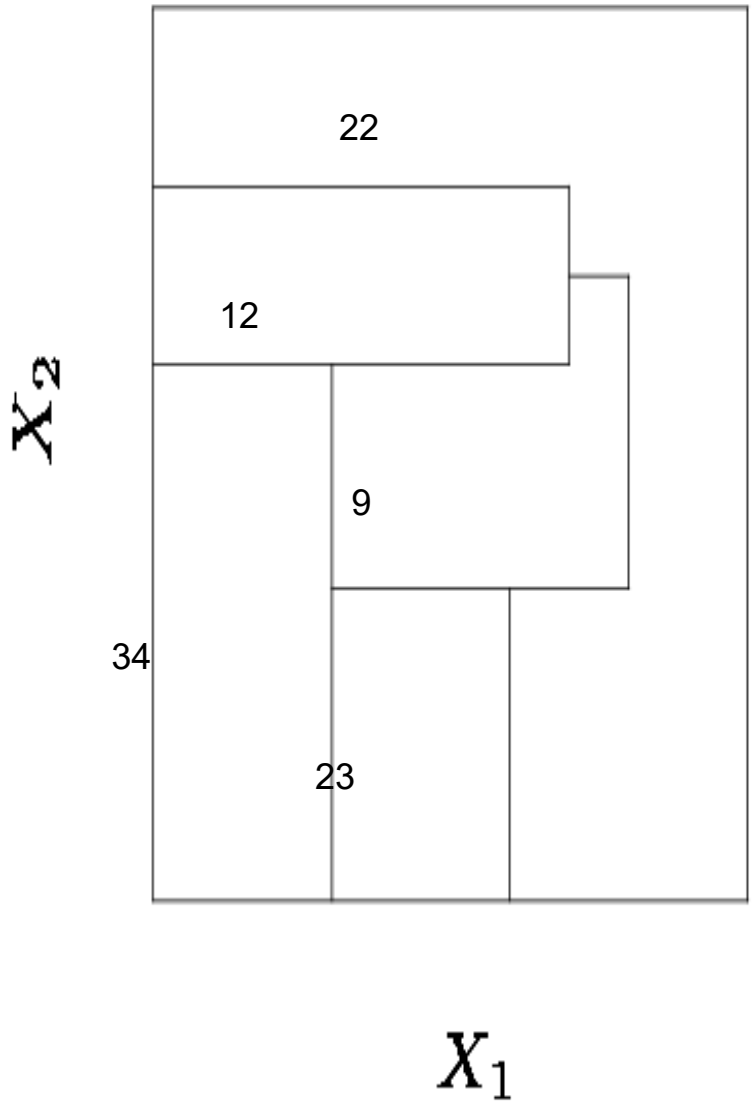
Cây hồi quy

- Giả sử ta có 2 vùng R_1 và R_2 với $\hat{Y}_1 = 10, \hat{Y}_2 = 20$

- Với các giá trị của X mà $X \in R_1$ ta sẽ có giá trị dự đoán là 10, ngược lại $X \in R_2$ ta có kết quả dự đoán là 20.

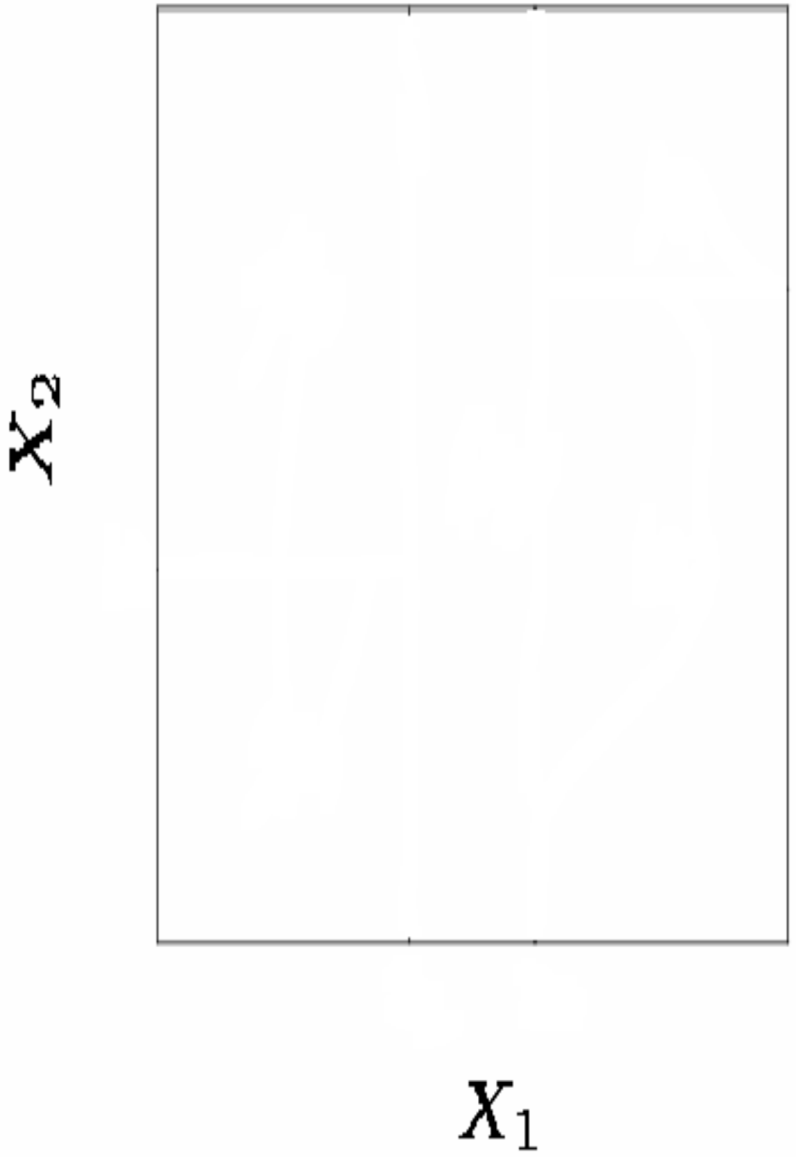
Cây hồi quy

- Cho 2 biến đầu vào và 5 vùng
- Tùy theo từng vùng của giá trị mới X ta sẽ có dự đoán 1 trong 5 giá trị cho Y.



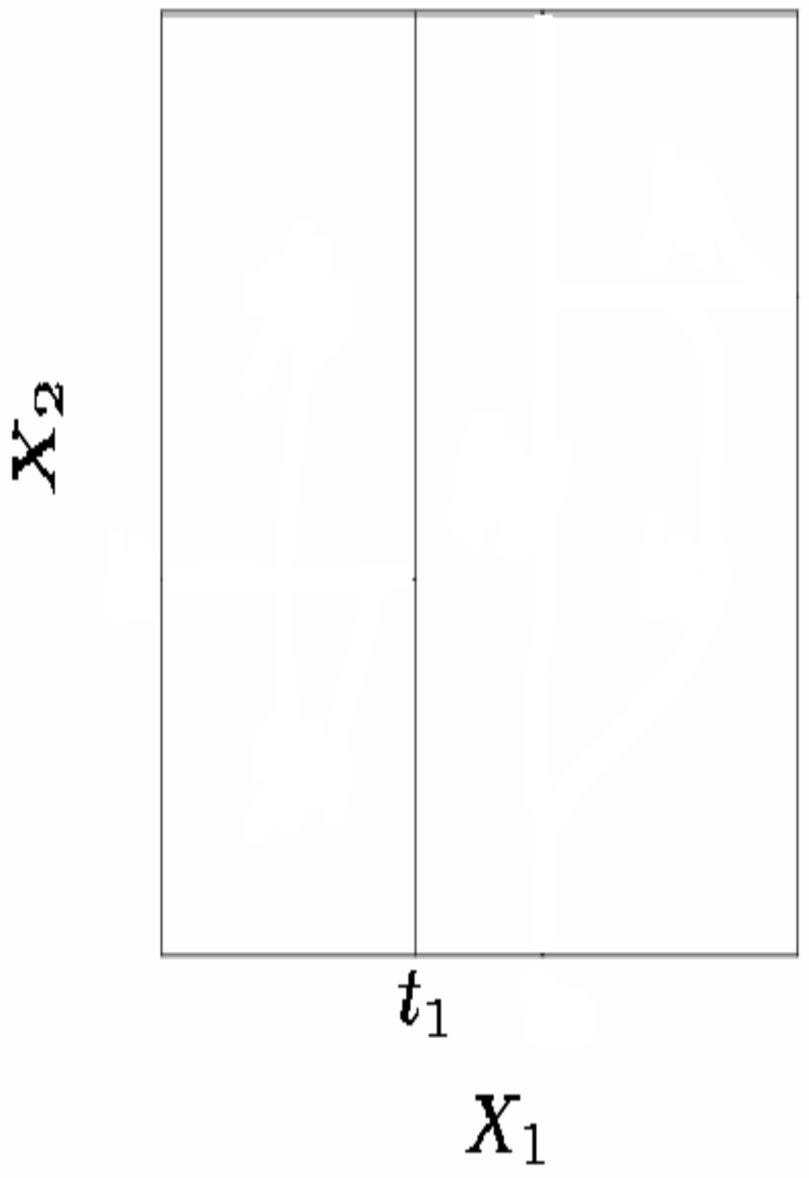
Tách các biến X

- Ta tạo ra các phân vùng bằng cách tách lặp đi lặp lại một trong các biến X thành hai vùng



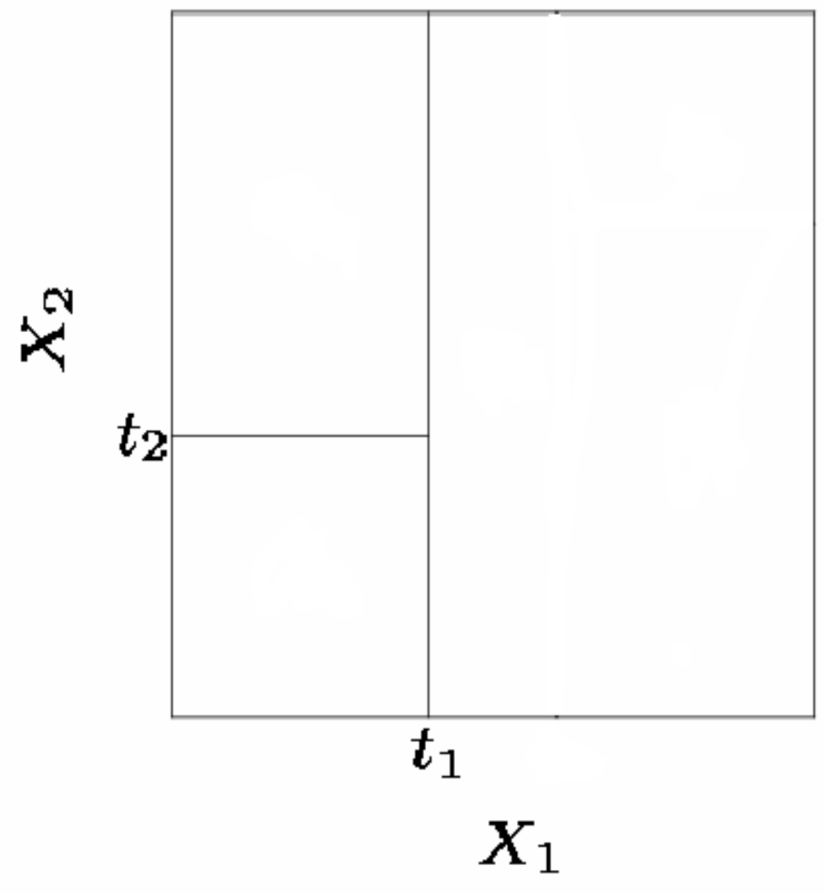
Tách các biến X

1. Đầu tiên tách trên $X_1=t_1$



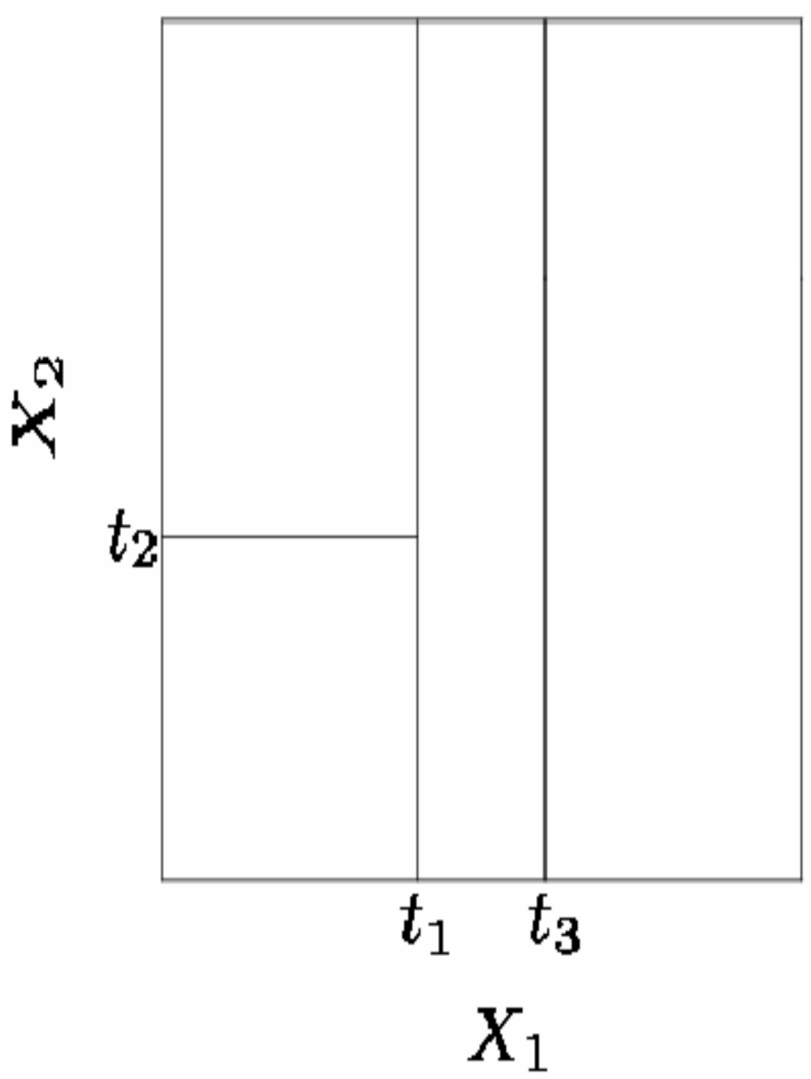
Tách các biến X

1. Đầu tiên tách trên $X_1=t_1$
2. Nếu $X_1<t_1$, tách trên $X_2=t_2$



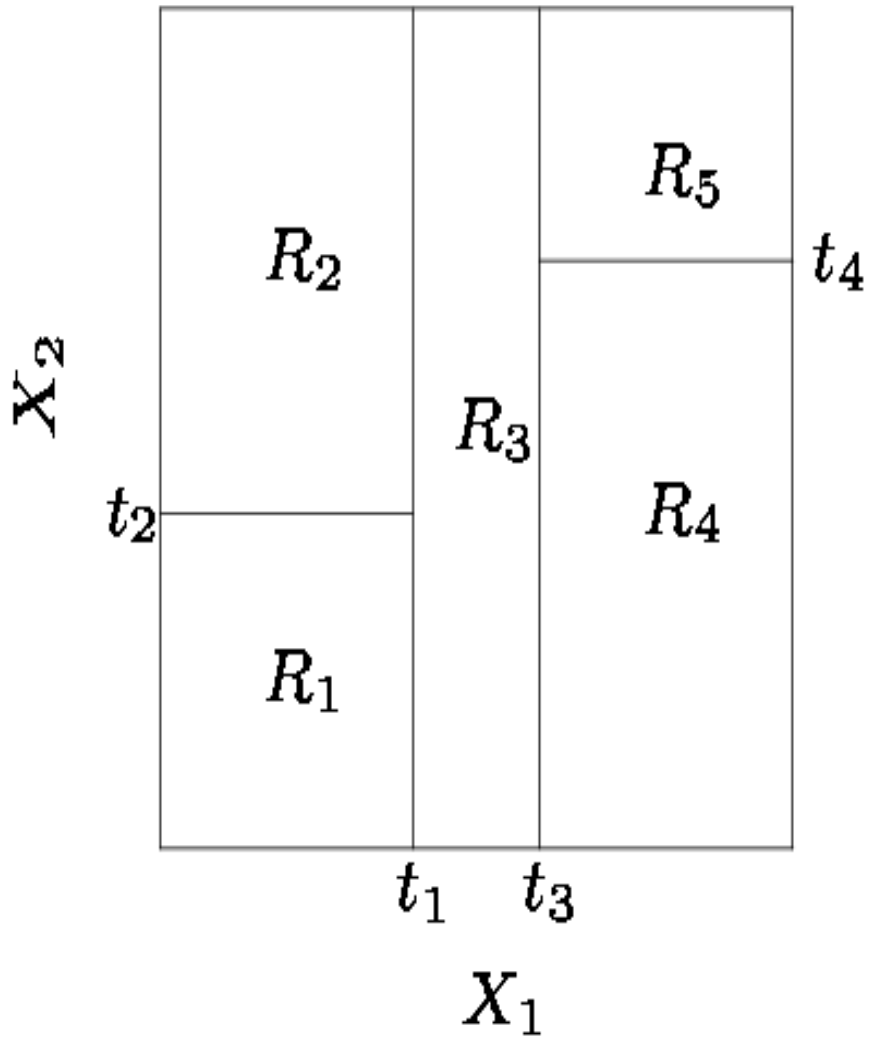
Tách các biến X

1. Đầu tiên tách trên $X_1=t_1$
2. Nếu $X_1<t_1$, tách trên $X_2=t_2$
3. Nếu $X_1>t_1$, tách trên $X_1=t_3$

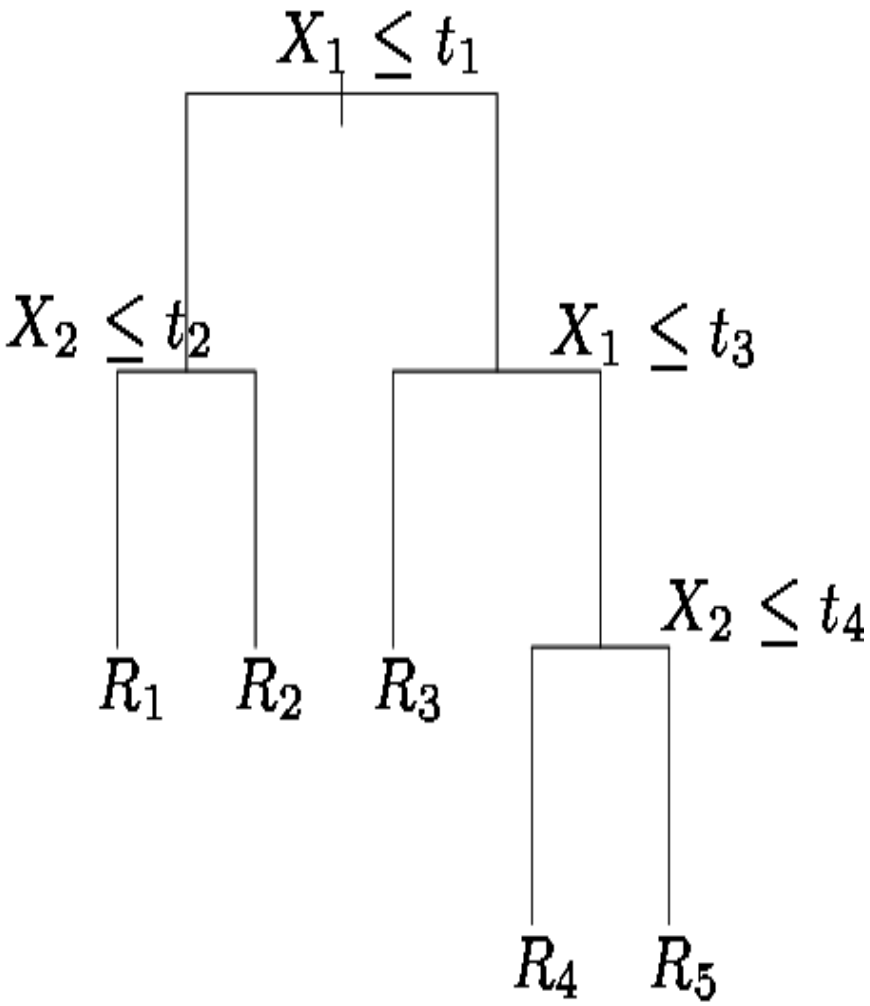


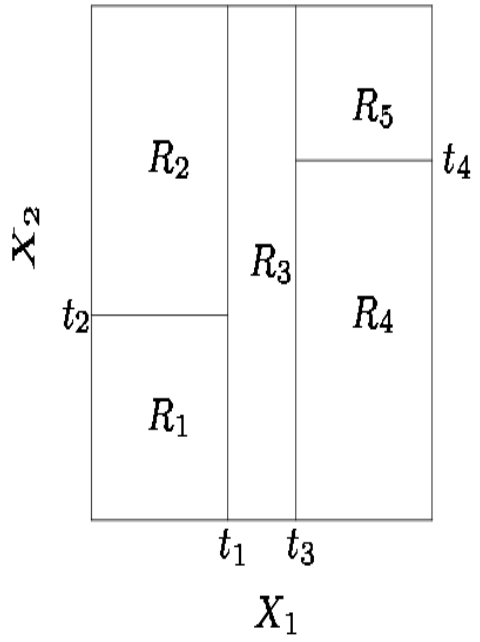
Tách các biến X

1. Đầu tiên tách trên $X_1=t_1$
2. Nếu $X_1<t_1$, tách trên $X_2=t_2$
3. Nếu $X_1>t_1$, tách trên $X_1=t_3$
4. Nếu $X_1>t_3$, tách $X_2=t_4$



Tách các biến X





- Khi ta tạo các vùng theo phương pháp này, ta có thể biểu diễn chúng dùng cấu trúc cây.
- Phương pháp này dễ diễn giải mô hình dự đoán, dễ diễn giải kết quả

Giải thuật tham lam: hồi quy

- Tìm thuộc tính tách j và điểm tách s mà nó cực tiểu lỗi dự đoán

$$\min_{j,\,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right]$$

CÂY PHÂN LỚP

$$\text{class } k(m) = \arg\max_k \hat{p}_{mk}$$

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

Giải thuật tham lam: phân lớp

- Nhiều độ đo cho lỗi dự đoán (độ hỗn tạp của nút-node impurity)

Misclassification error: $\frac{1}{N_m}\sum_{i\in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk(m)}$

$$\hat{p}_{mk} = \frac{1}{N_m}\sum_{x_i \in R_m} I(y_i = k)$$

Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

Giải thuật tham lam: phân lớp

- Nhiều độ đo cho lỗi dự đoán (độ hỗn tạp của nút-node impurity)

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

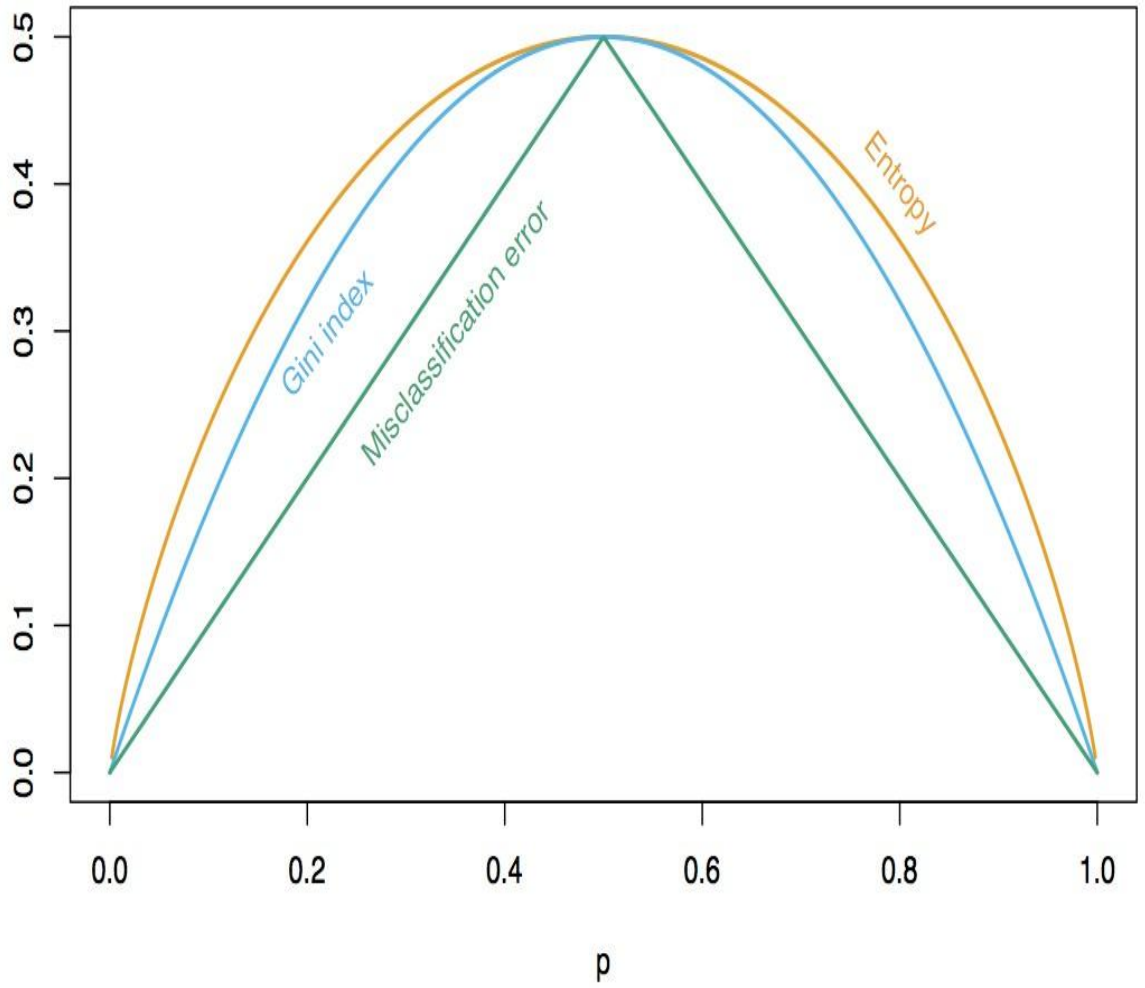
Gini index:

$$\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

Độ hỗn tạp của nút khi phân lớp

Classification node impurity

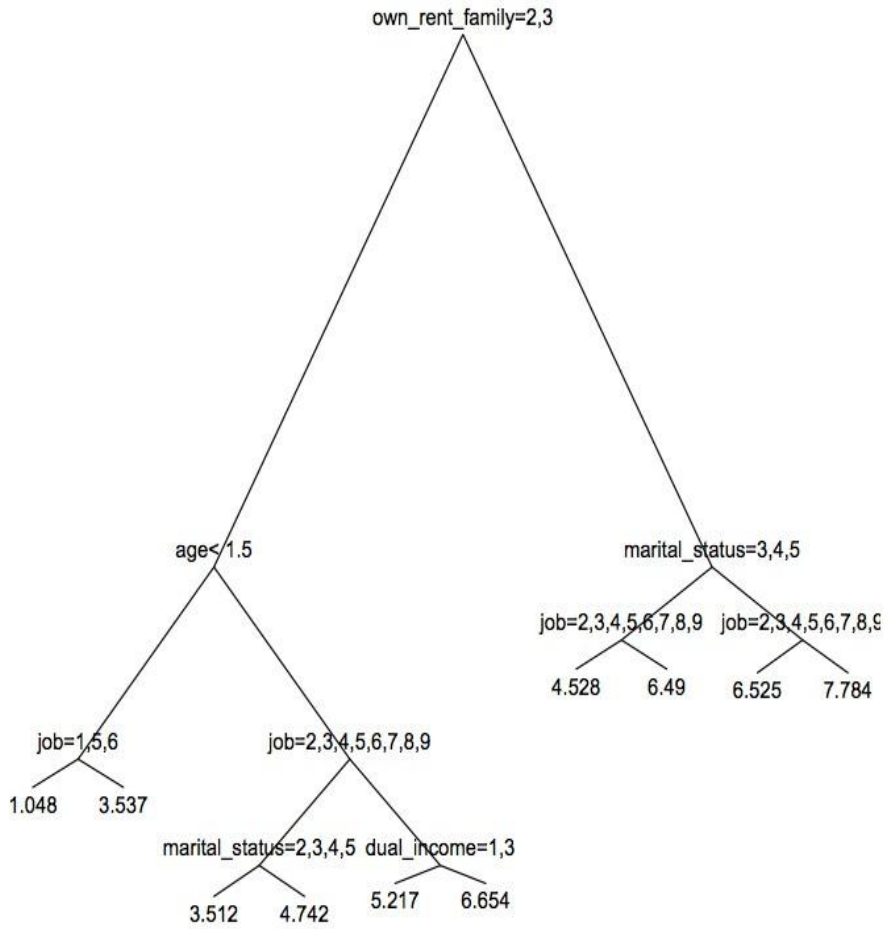


Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

Ưu điểm của CART

Ưu điểm của CART

Dễ giải thích, lý tưởng để lý giải "tại sao" cho người ra quyết định



Ưu điểm của CART

Xử lý được tính tương tác cao giữa các thuộc tính

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \ \ldots$$

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \theta_1 x_1 x_2 + \theta_2 x_1 x_3 + \theta_3 x_2 x_3 + \lambda_1 x_1 x_2 x_3 \ldots$$

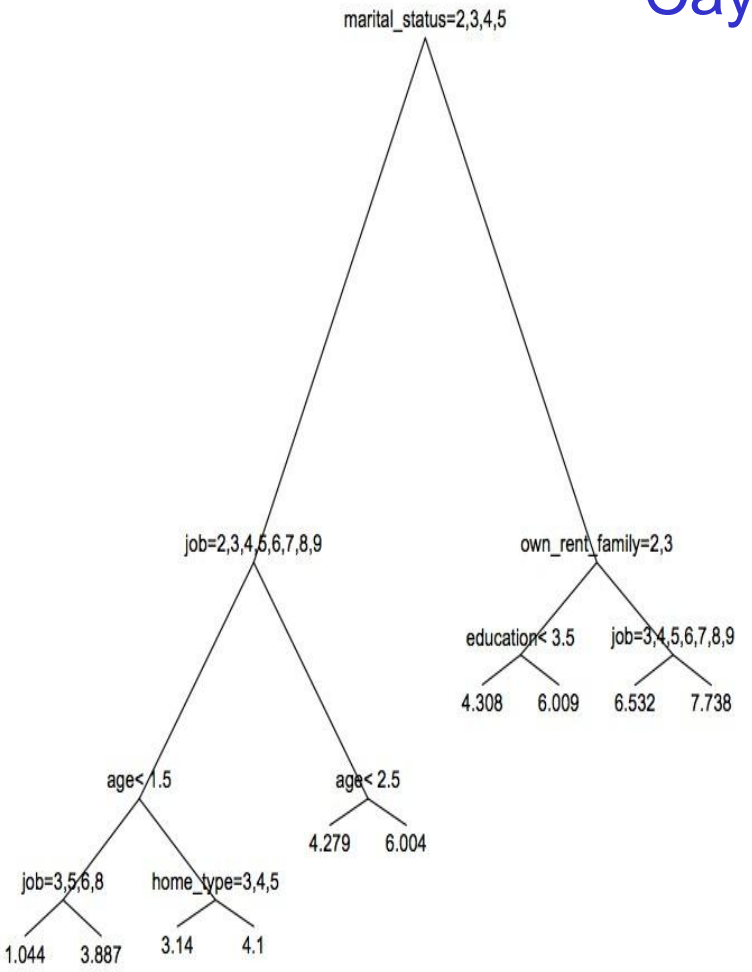
Y = 3.5 if ((1<marital_status<6) AND (1<job<9)) AND (age<1.5) OR …

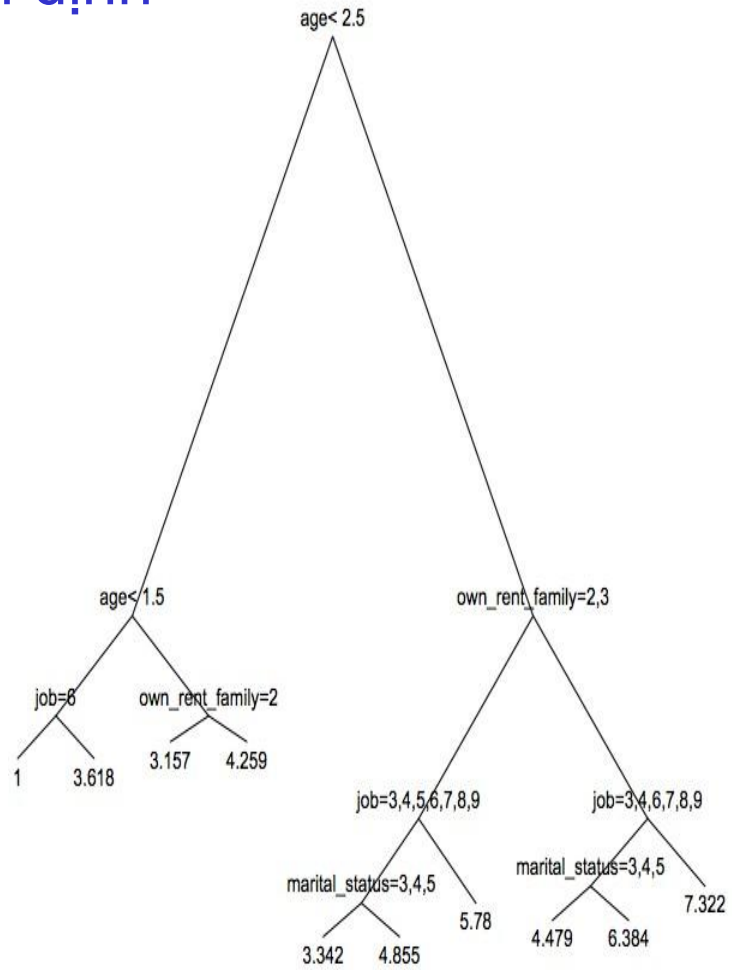
Nhược điểm của CART

- Cây không ổn định (Instability of trees)
- Thiếu tính trơn (Lack of smoothness)
- Khó nắm bắt độ cộng tính (Hard to capture additivity)

Nhược điểm của CART

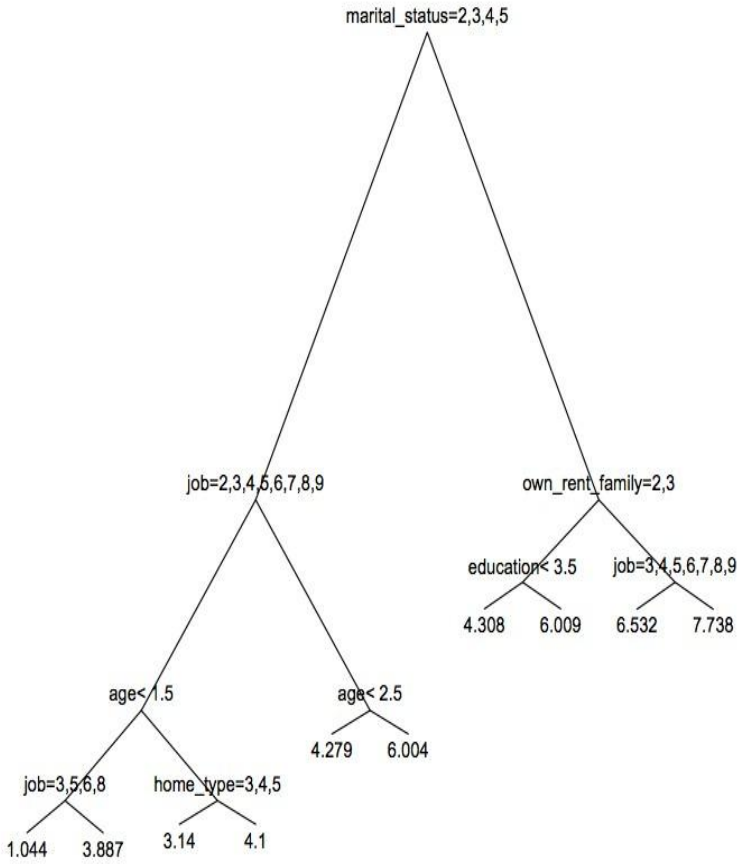
Cây không ổn định

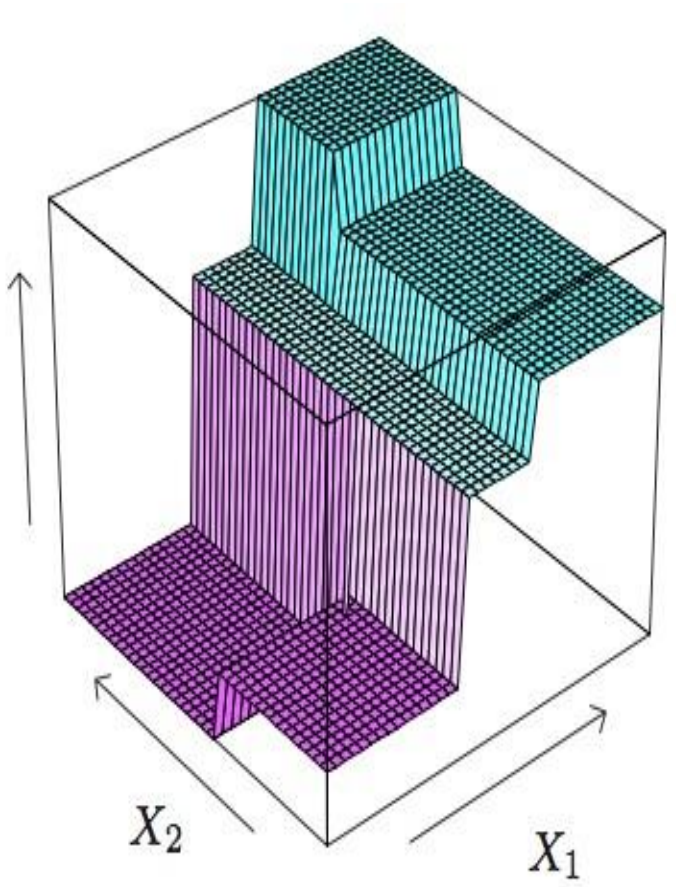




Nhược điểm của CART

Thiếu tính trơn (Smoothness)





Khắc phục nhược điểm của CART

1. Cây không ổn định
   - *Giải pháp –* ***Random Forests***
2. Thiếu tính trơn
   - *Giải pháp – MARS*
3. Khó nắm bắt độ cộng tính (additivity)
   - *Giải pháp – MART* or *MARS*

MARS – "Multivariate Adaptive Regression Splines"

MART – "Multiple Additive Regression Trees"