

Các phương pháp học máy kết hợp

Bagging và Random Forests

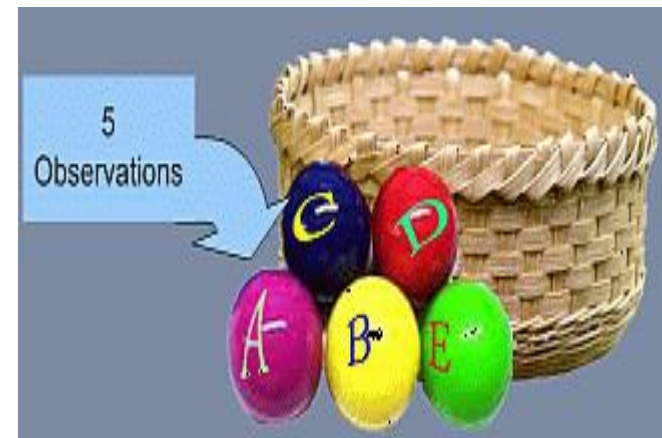


- Bootstrap
- Bagging
- Random forest

Ví dụ về Bootstrap



- Giả sử ta có 5 quả bóng gắn nhãn A,B,C,D, E và bỏ tất cả chúng vào trong 1 cái giỏ.
- Lấy ra ngẫu nhiên 1 quả từ giỏ và ghi lại nhãn, sau đó bỏ lại quả bóng vừa bốc được vào giỏ.
- Tiếp tục lấy ra ngẫu nhiên một quả bóng và lặp lại quá trình trên cho đến khi việc lấy mẫu kết thúc. Việc lấy mẫu này gọi là **lấy mẫu có hoàn lại**.



Nguồn: bis.net.vn/forums

- **Kết quả** của việc lấy mẫu như trên có thể như sau (giả sử kích thước mẫu là 10):

C, D, E, E, A, B, C, B, A, E



Bootstrap là gì?



- Bootstrap là kỹ thuật rất quan trọng trong thống kê
- Bootstrap là phương pháp lấy mẫu có hoàn lại (sampling with replacement). Khi đó, một mẫu có thể xuất hiện nhiều lần trong một lần lấy mẫu



Các phương pháp kết hợp

Ensemble Methods

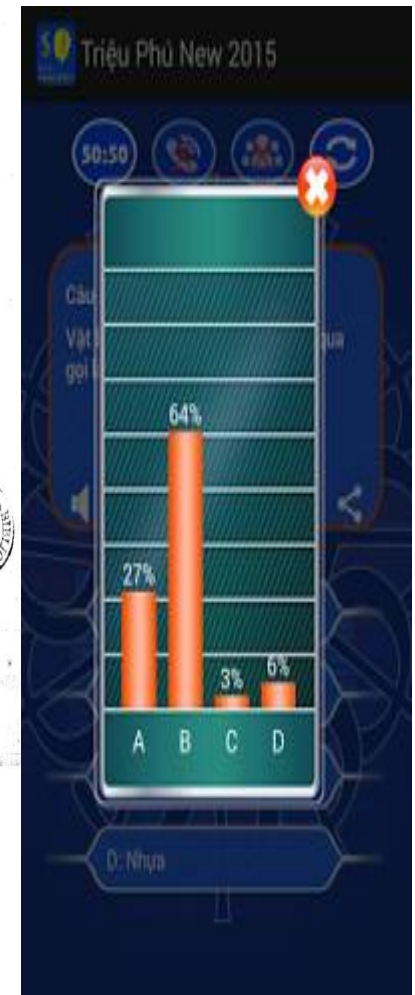
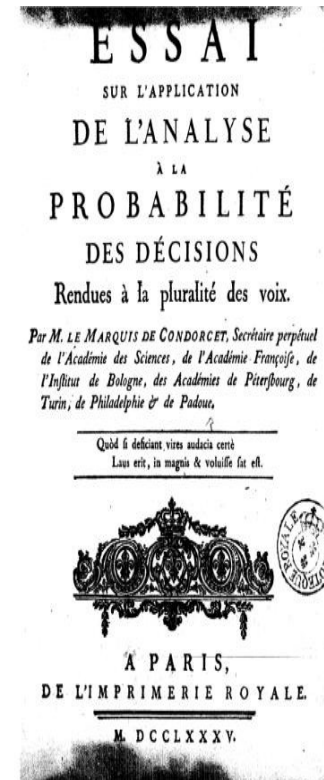
Condorcet's Jury Theorem – Nếu p lớn hơn $1/2$ (mỗi mỗi người tham gia bỏ phiếu đúng mong muốn của họ), thì càng thêm nhiều tham gia sẽ tăng xác suất theo quyết định số đông sẽ chính xác.

Trong giới hạn, xác suất bầu chọn theo số đông tiến đến 1 khi số cử tri tăng lên.

Sức mạnh của các bộ phân lớp yếu



Condorcet's Jury Theorem – Nếu p lớn hơn $1/2$ (mỗi mỗi người tham gia bỏ phiếu đúng mong muốn của họ), thì càng thêm nhiều tham gia sẽ tăng xác suất theo quyết định số đông sẽ chính xác. Trong giới hạn, xác suất bầu chọn theo số đông tiến đến 1 khi số cử tri tăng lên.



Source: gallica.bnf.fr / Bibliothèque nationale de France



- ❖ Việc lấy trung bình làm giảm phương sai và không làm tăng bias (bias vẫn được giữ nguyên)

$$\text{Var}[\bar{Y}] = \sigma^2/n$$

- ❖ Sự tương quan trong các phiếu bầu của các bộ phân lớp không trợ giúp được nhiều

$$\alpha \times \{ \mathbf{CART} \} + (1 - \alpha) \times \{ \mathbf{LinearModel} \}$$

Các phương pháp kết hợp:

- Bagging
- Random Forest

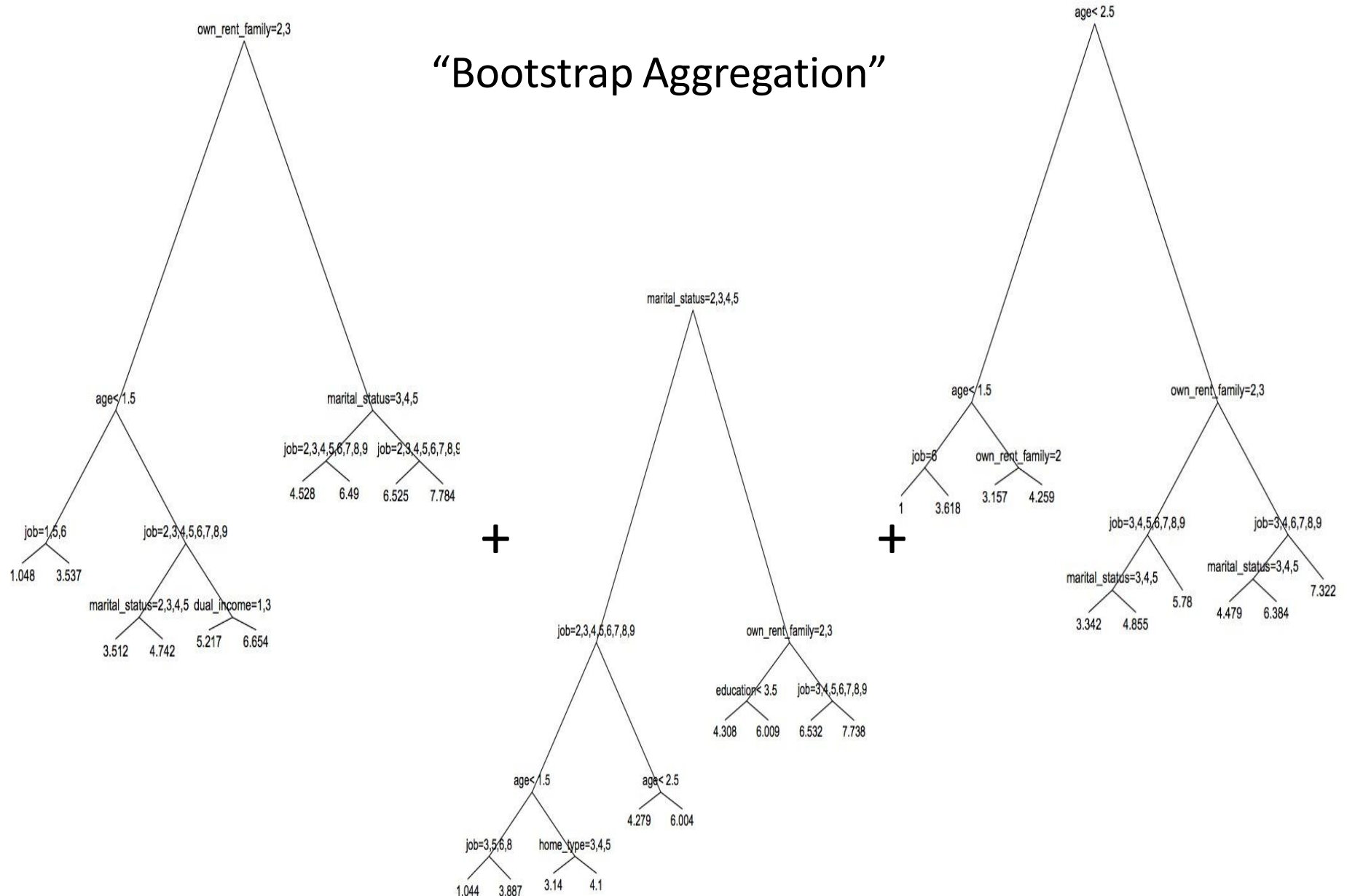
PHƯƠNG PHÁP BAGGING

- Có bias thấp và phương sai cao
- Nhạy với dữ liệu huấn luyện (các bộ huấn luyện khác nhau sẽ cho các cây hoàn toàn khác nhau)
→ Bagging được sử dụng để giảm phương sai của các phương pháp học máy (đặc biệt là cây quyết định)

Bagging là gì?



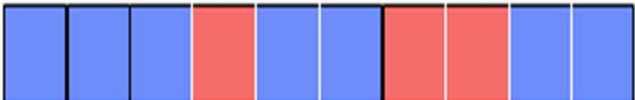
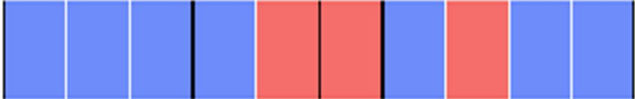

“Bootstrap Aggregation”



Bagging là gì?

“Bootstrap Aggregation”

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^{*b}(x)$$

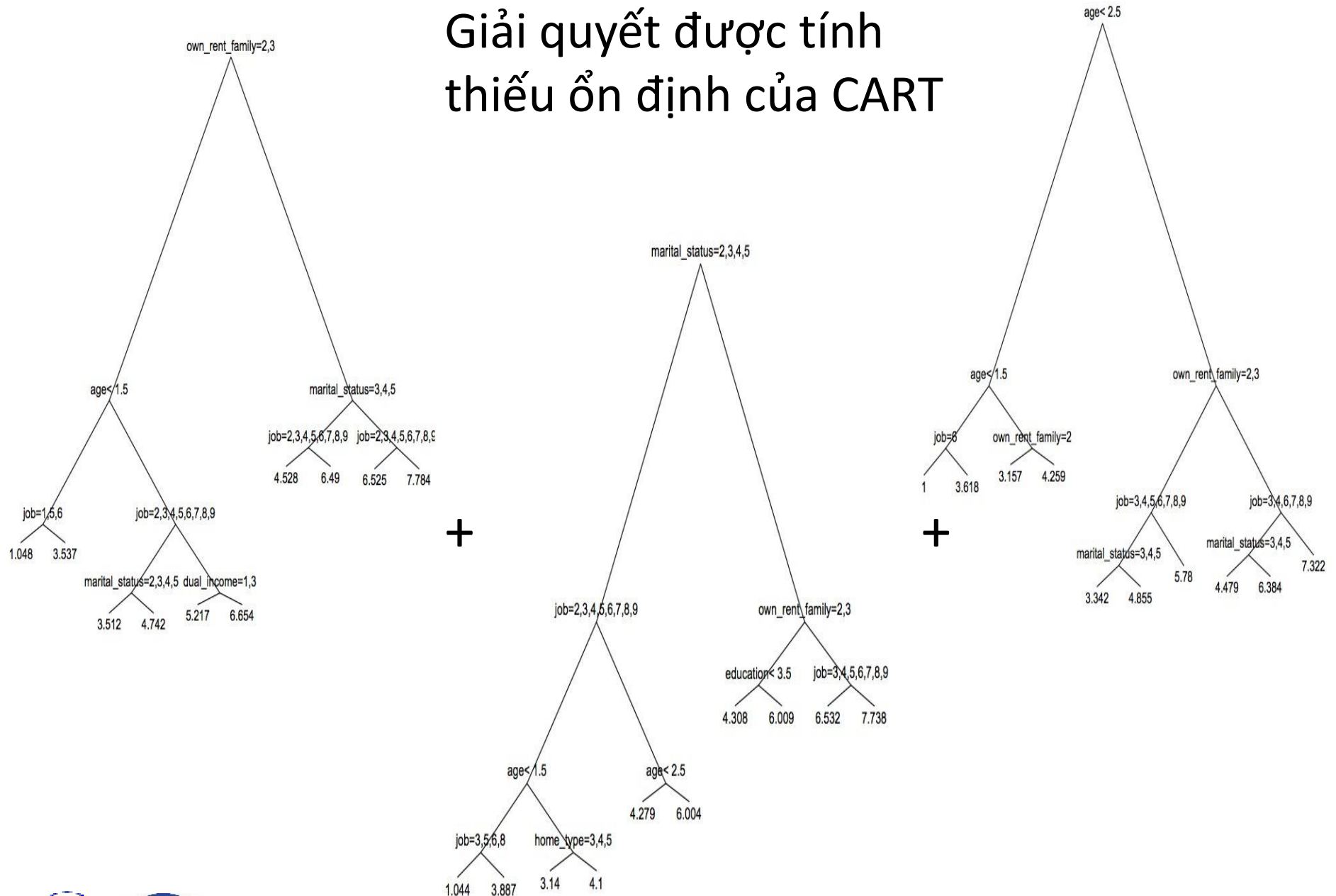
Model 1		7/10 correct
Model 2		7/10 correct
Model 3		6/10 correct

Ensemble Model
(Majority Voting)

1 or more tutorials: algorithms.wtf

Bagging

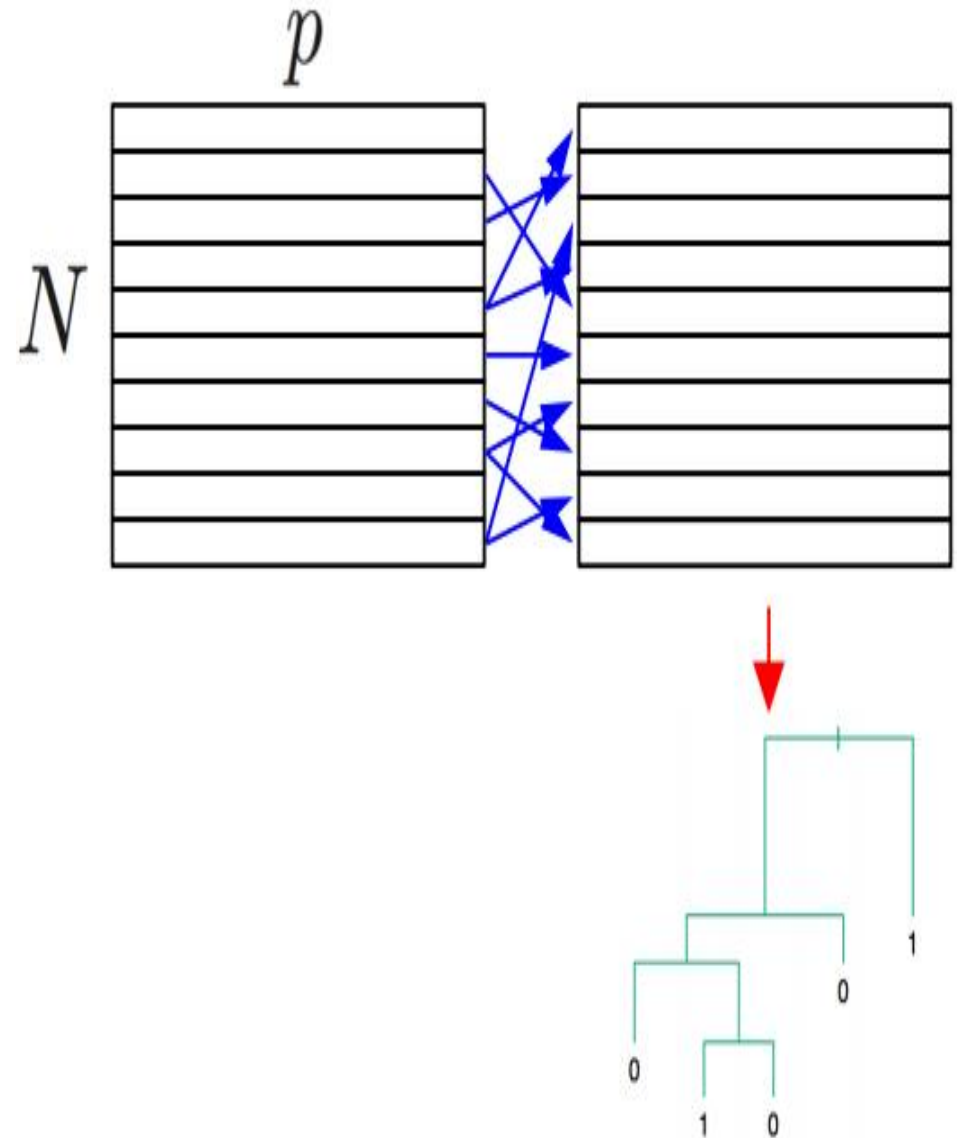
Giải quyết được tính
thiếu ổn định của CART



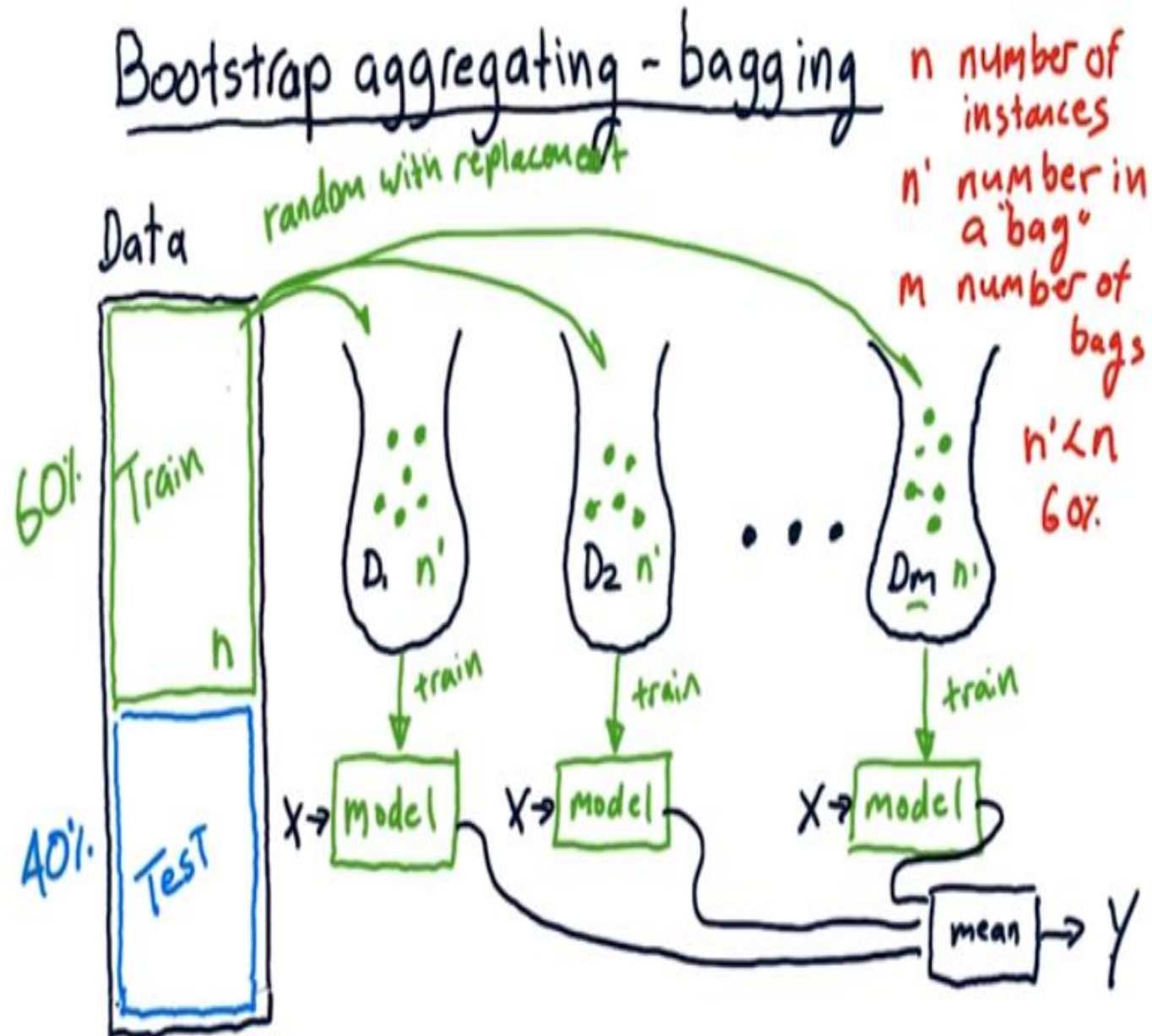
Bagging



- Lấy mẫu tập dữ liệu huấn luyện theo Bootstrap để tạo ra tập hợp các dự đoán.



Bagging



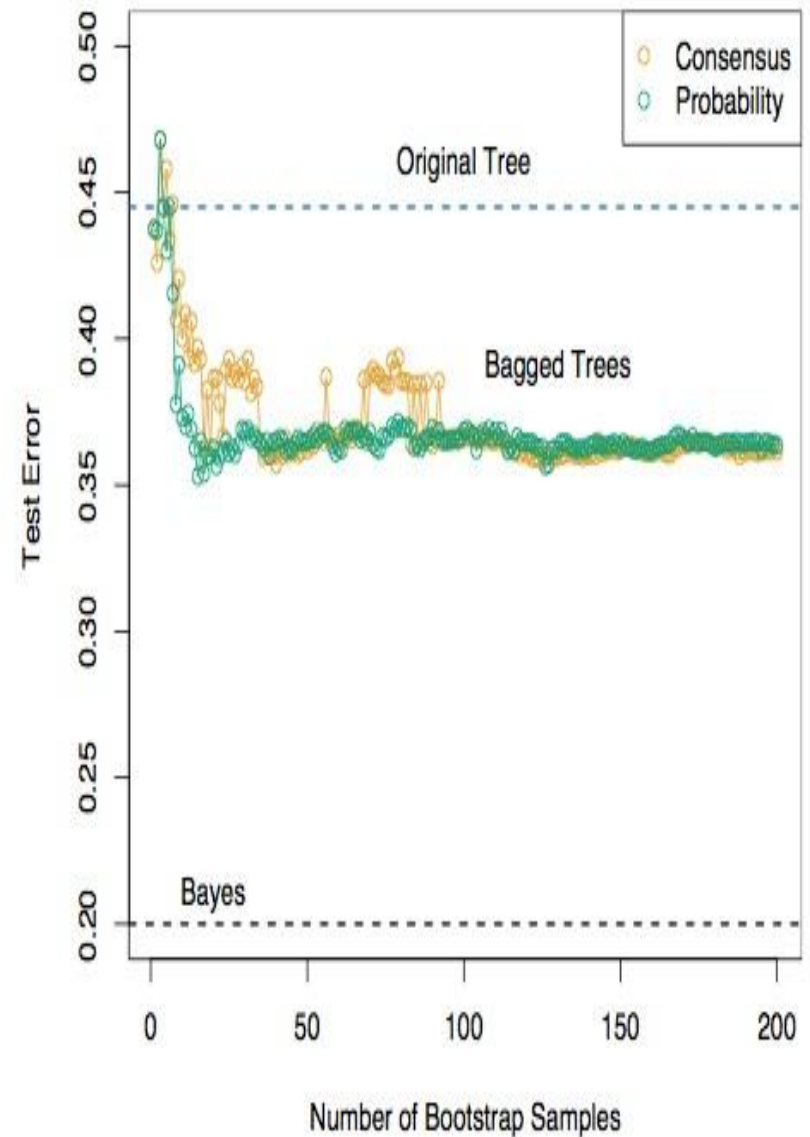
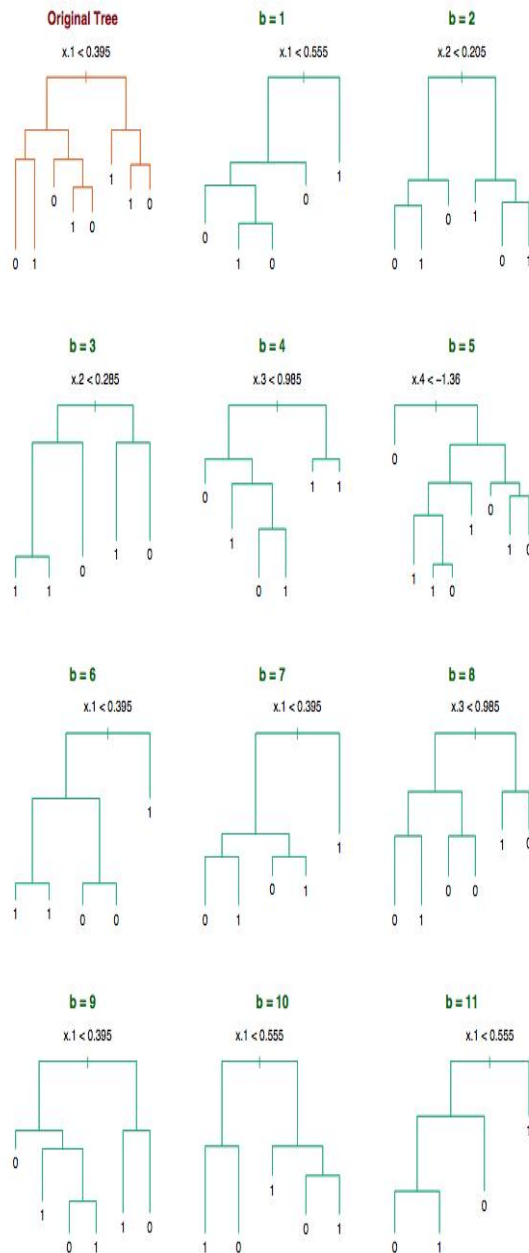
Bagging với cây quyết định và CART



- Tạo nhiều tập mẫu con một cách ngẫu nhiên
- Xây dựng mô hình CART trên mỗi tập mẫu.
- Cho 1 tập dữ liệu mới, xác định dự báo trên mỗi mô hình nhận được ở bước trên
- Ra quyết định: dựa trên trung bình hoặc lấy theo số đông từ các kết quả nhận được

- Tham số trong quá trình xây dựng là số tập mẫu được chọn, tham số này ảnh hưởng đến số cây được tạo ra.
- Số cây càng lớn, thời gian thực hiện càng lâu nhưng hiện tượng overfitting không xảy ra.

Bagging



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

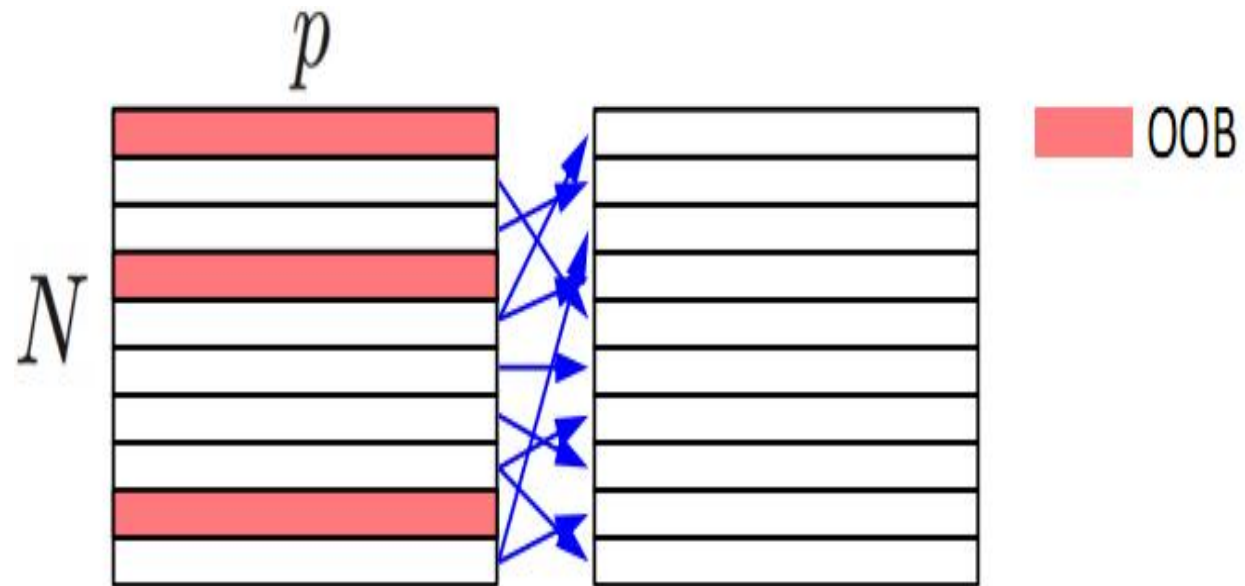
FIGURE 8.9. Bagging trees on simulated dataset. The top left panel shows the original tree. Eleven trees grown on bootstrap samples are shown. For each tree, the top split is annotated.

Out-of-bag cross-validation

Các mẫu Out-of-bag (OOB)



- Quá trình Bootstrapping:



- Mỗi cây chỉ sử dụng một tập con các mẫu huấn luyện (trung bình số mẫu $\sim 2/3$).
- Số mẫu cho OOB khoảng $\sim 1/3$ của cây quyết định.

PHƯƠNG PHÁP RỪNG NGẪU NHIÊN

Random Forests (RF)

Động lực để có Random forest



- Mô hình dựa trên cây phân loại và hồi quy (CART).
- Các mô hình cây có lỗi bias thấp, tuy nhiên phương sai lại cao (high variance).
- Phương pháp Bagging dùng để giảm phương sai giữ bias.
- Random forest là 1 cải tiến của Bagging

Động lực để có Random forest



- Các thuật toán cây quyết định sử dụng giải thuật tham lam
- Các cây quyết định có độ tương quan cao trong các dự đoán.
- Việc dự đoán sẽ tốt nếu các cây con không tương quan hoặc tương quan yếu.
- Random forest thay đổi thuật toán để tạo ra các mô hình con có tương quan yếu với nhau



Bagged trees vs. random forests



- **Bagging**

- Biểu thị sự biến thiên (variability) giữa các cây bởi việc chọn mẫu ngẫu nhiên từ dữ liệu huấn luyện.
- Cây được sinh ra từ phương pháp Bagging vẫn có tương quan lẫn nhau, do đó hạn chế trong việc giảm phương sai.

- **Random forests**

- Đưa ra thêm tính ngẫu nhiên (randomness)
- Làm giảm mối tương quan giữa các cây bằng cách lấy ngẫu nhiên các biến khi tách nút của cây.



CART vs Random forest



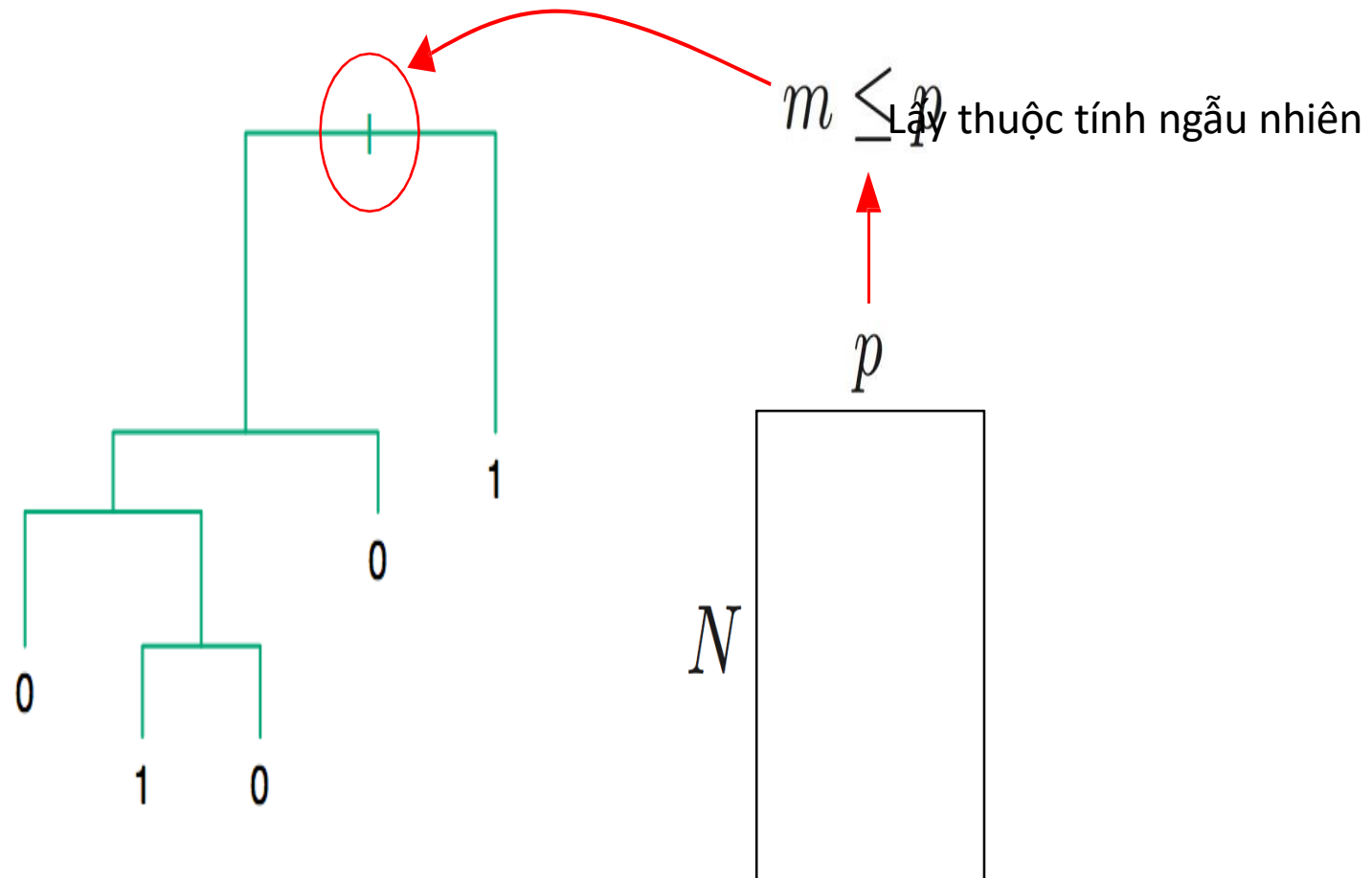
- Trong mô hình CART: việc lựa chọn nút tách phải xét đến tất cả các thuộc tính và tất cả các giá trị của thuộc tính để tối ưu hóa việc tách nút
- Random forest chỉ giới hạn một tập ngẫu nhiên các đặc trưng được tìm kiếm (số đặc trưng này ký hiệu là ***m*** và được gọi là ***số biến khả tách***)
- Thực hiện với các giá trị khác nhau của m và hiệu chỉnh bằng cách sử dụng cross validation



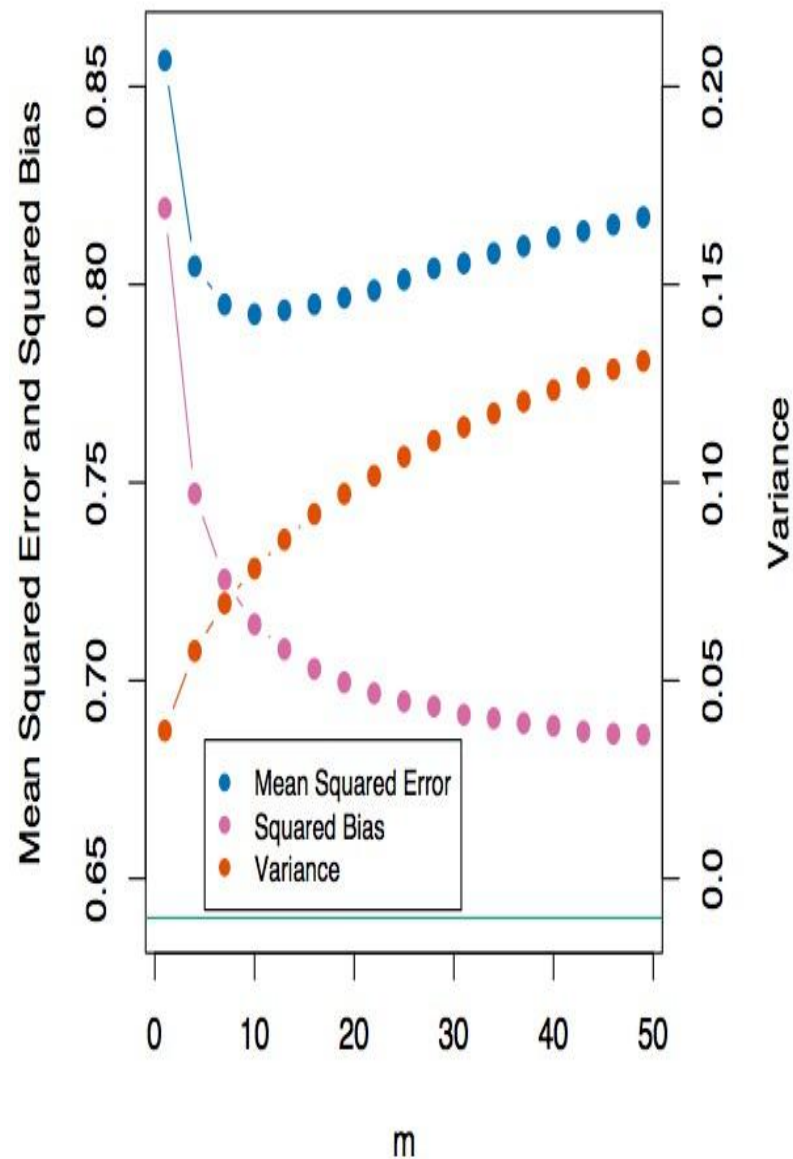
Các biến dùng cho tách nút



Số lượng biến dùng để tách nút



Các biến dùng cho tách nút



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



Bước 1:

Tập dữ liệu huấn luyện

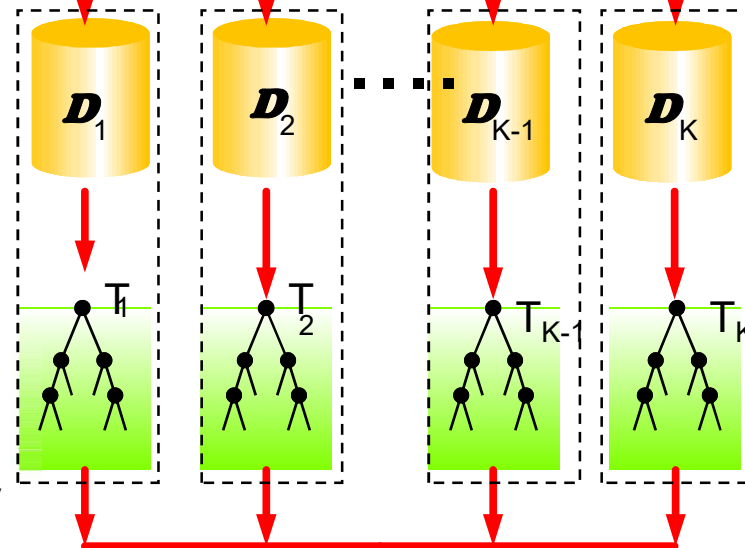


Tạo dữ liệu ngẫu nhiên
(mẫu bootstrap)

$\mathbf{D} = (\mathbf{X}_i, Y_i), i=1..p$
p: #chiều, N: #mẫu

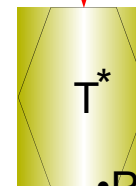
Bước 2:

Sử dụng các tập con dữ liệu lấy mẫu ngẫu nhiên để xây dựng cây



Bước 3:

Kết hợp các cây



- Phân lớp: Bình chọn theo số đông
- Hồi quy: Lấy trung bình giá trị dự đoán từ các cây $T_i (i=1..K)$

Các tham số chính

Các tham số quan trọng của Rừng ngẫu nhiên:

- Số lượng biến khả tách tại mỗi nút (m)
- Độ sâu của từng cây trong rừng (số lượng mẫu tối thiểu tại mỗi nút của cây-minimum node size)
- Số lượng cây trong rừng

Với p là số biến đầu vào của dữ liệu, giá trị của m tốt nhất là :

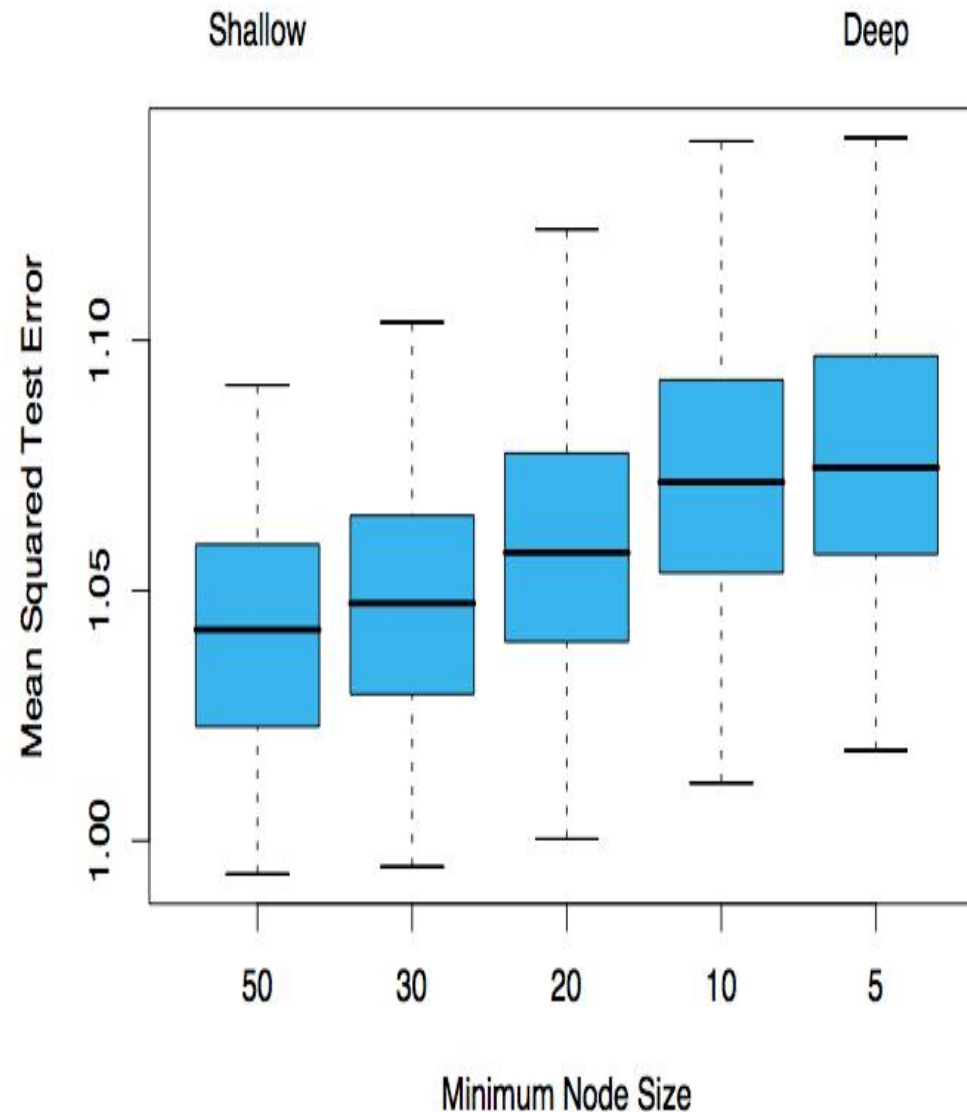
- Bài toán Phân lớp: $m = \sqrt{p}$,
- Bài toán Hồi quy: $m = p/3$

Ví dụ: Nếu mô hình phân lớp gồm 25 biến đầu vào
→ số biến khả tách là $\sqrt{25} = 5$

gói `randomForest` trong R dùng `mtry`

Độ sâu của từng cây

(số lượng mẫu tối thiểu tại mỗi nút của cây)



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

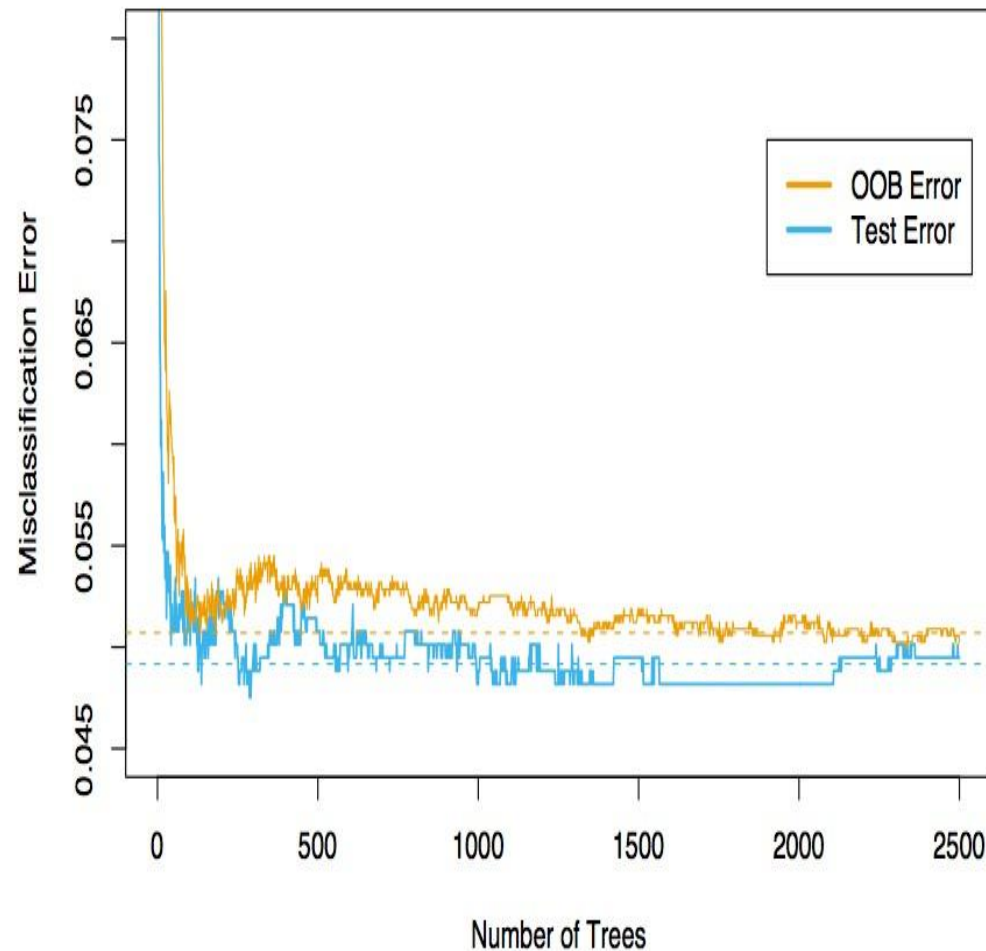
Độ sâu của cây

Giá trị mặc định

Bài toán phân lớp 1

Bài toán hồi quy 5

Số lượng cây trong rừng



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

- Thêm nhiều cây không gây ra overfitting.



Các tính năng khác của RF



- Các mẫu Out-of-bag (OOB)
- Độ quan trọng của biến (Variable importance measurements)



Độ quan trọng của biến



- Trong các cây quyết định (DT hoặc CART), điểm rơi của hàm sai số tại mỗi biến tại điểm tách nút là có thể tính được.
- Trong các bài toán hồi quy, giá trị này có thể đánh giá bằng tổng bình phương sai số và trong bài toán phân lớp, giá trị này có thể đo được bằng chỉ số Gini.
- Các điểm này có thể được tính trung bình trên các cây quyết định và cung cấp một ước lượng cho mức độ quan trọng của mỗi biến đầu vào. Giá trị điểm rơi càng lớn khi các biến được chọn thì biến đó càng quan trọng.



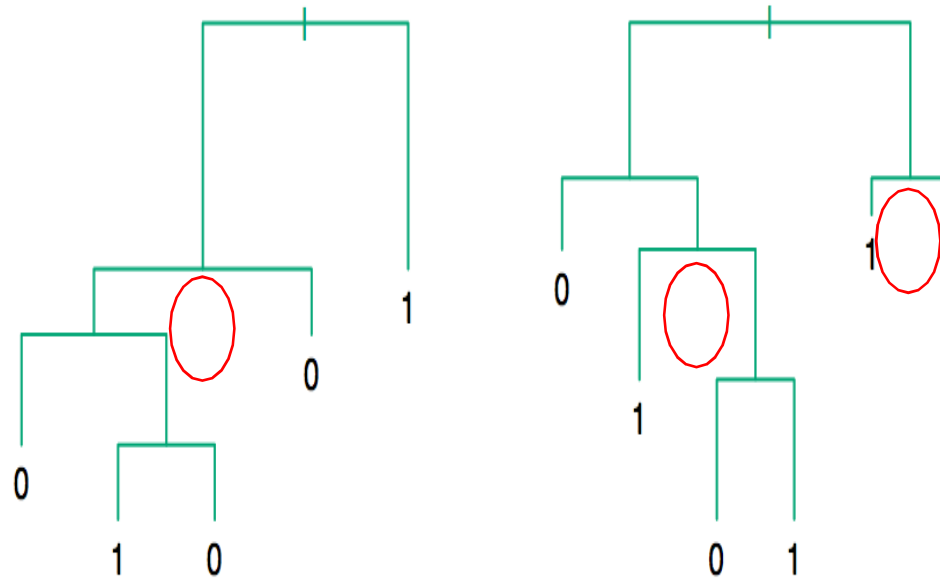
Độ quan trọng của biến



- Dựa vào kết quả này, các tập con của các biến đầu vào có thể được xác định là quan trọng nhất hoặc ít quan trọng nhất với bài toán đặt ra.
- Trong trường hợp này, việc rút gọn biến có thể được thực hiện

Độ quan trọng của biển

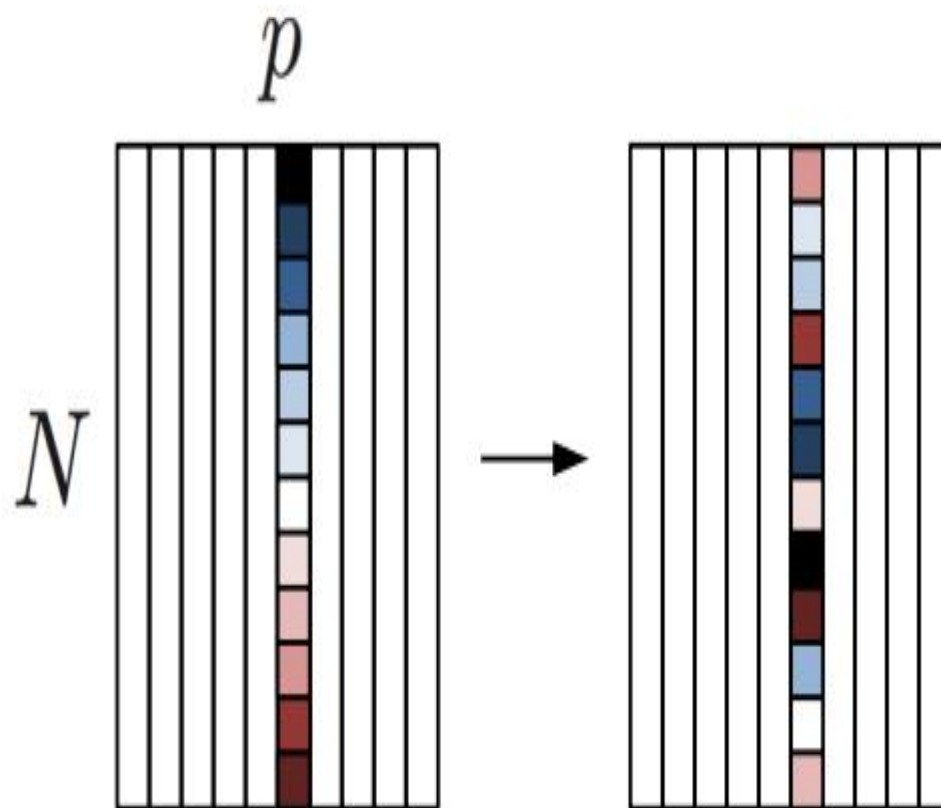
- **Dạng 1:**
- Độ giảm của lỗi dự đoán hoặc impurity từ các điểm tách nút liên quan đến các biến đó, cuối cùng lấy trung bình trên các cây trong rừng.



Độ quan trọng của biến



- **Dạng 2:**
- Độ tăng lỗi dự đoán tổng thể khi **các giá trị của biến được hoán vị ngẫu nhiên giữa các mẫu.**

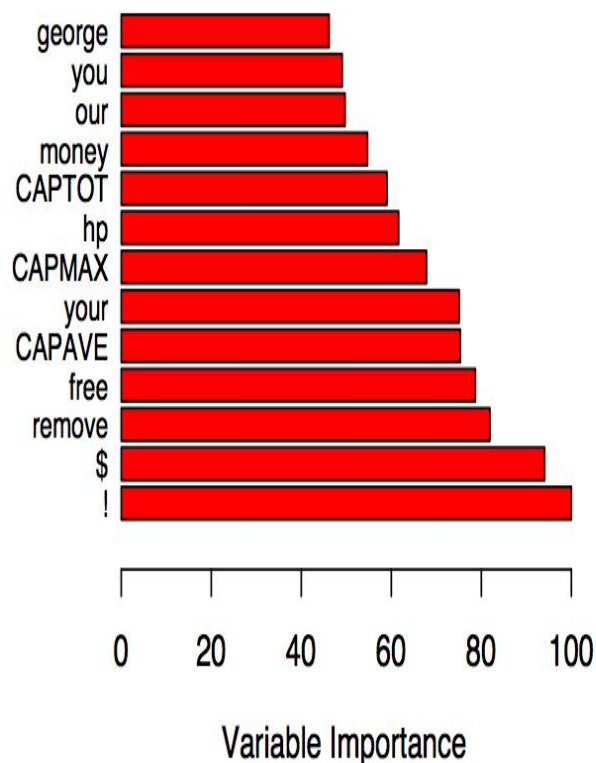


Ví dụ về độ quan trọng của biến

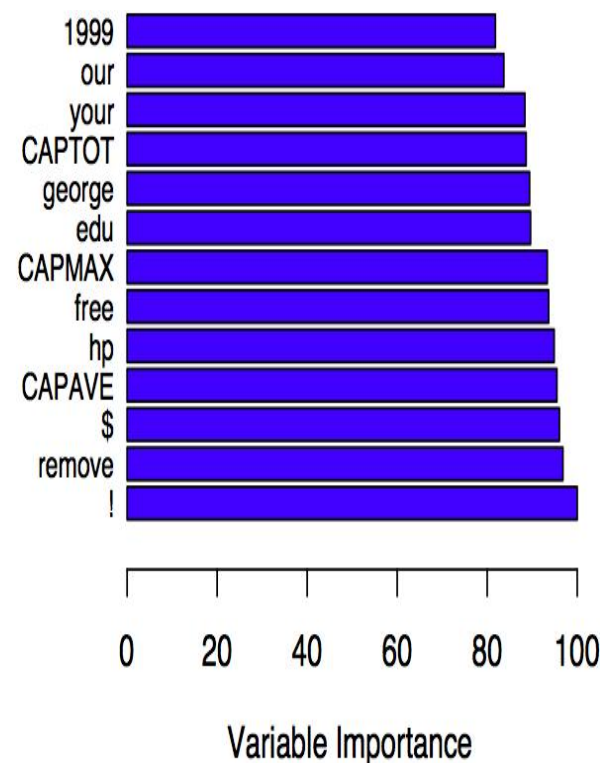


- Cả 2 dạng biểu thị gần giống nhau, tuy nhiên có sự khác biệt về xếp hạng các biến:

Dạng 1



Dạng 2



Tương tự như CART:

- RF tương đối mạnh trong việc xử lý **biến rác** (non-informative variable)
- RF xử lý (nắm bắt) được độ tương tác bậc cao giữa các biến (Capture high-order interactions between variables)
- RF có lỗi bias thấp
- RF dễ xử lý các biến hỗn hợp (biến rời rạc, phân loại)

Ưu điểm vượt trội CART:

- Lỗi phương sai thấp hơn (mạnh hơn vì sử dụng phương pháp bootstrapping lấy mẫu từ tập huấn luyện)
- Ít bị overfitting hơn
- Không cần tỉa cây (No need for pruning)
- Kiểm tra chéo được tích hợp sẵn trong mô hình (dùng các mẫu OOB)

Nhược điểm của RF



Tương tự như CART:

- Khó nắm bắt độ cộng tính

Nhược điểm so với CART:

- Khó diễn giải/giải thích mô hình dự đoán