

Random forest

☑ Trắc nghiệm 12 ☑ Thực hành 3 🔗 Chạy trên Kaggle ⬇ PDF bài

Một cây quyết định rất nhạy với dữ liệu: đổi vài mẫu huấn luyện là ngưỡng ở các nút có thể đổi, và dự đoán đổi theo. Random forest huấn luyện nhiều cây, mỗi cây trên một bản lấy mẫu khác của dữ liệu và chỉ xét một phần đặc trưng ở mỗi nút, rồi gộp dự đoán của các cây. Bài này giải thích vì sao gộp nhiều cây giúp dự đoán ổn định hơn, và vì sao các cây phải khác nhau thì gộp mới có tác dụng.

MỤC TIÊU

- Hiểu bootstrap và việc chọn ngẫu nhiên đặc trưng ở mỗi nút.
- Tính điểm OOB.
- Chọn số cây, độ sâu; đọc feature importance.

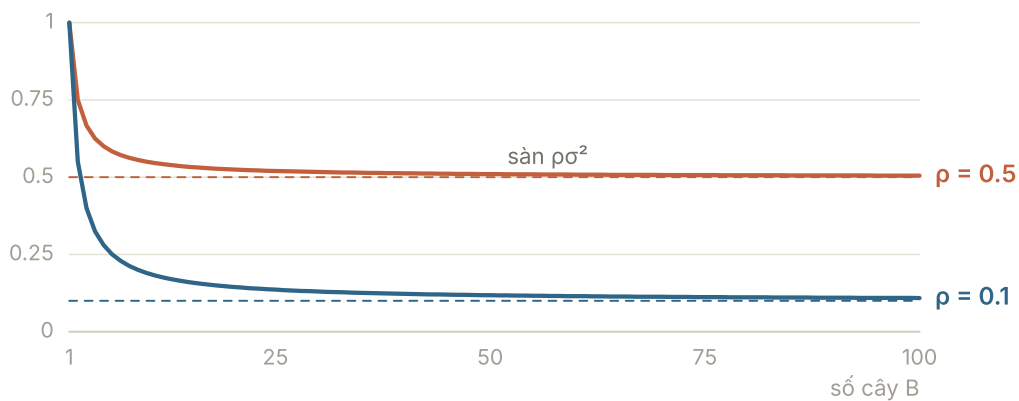
Vì sao trung bình giúp ích

Bài 2.5 kết luận cây có variance cao: đổi dữ liệu một chút là cây đổi hẳn. Giả sử có B mô hình, mỗi cái có phương sai σ^2 , và hai mô hình bất kỳ có hệ số tương quan ρ . Phương sai của trung bình là

$$\text{Var}\left(\frac{1}{B} \sum_{b=1}^B f_b\right) = \rho \sigma^2 + \frac{1-\rho}{B} \sigma^2.$$

Khi $B \rightarrow \infty$, số hạng thứ hai biến mất nhưng số hạng đầu thì còn nguyên. Nhiều cây hơn chỉ giúp đến một mức; muốn giảm tiếp thì phải làm các cây **ít giống nhau** hơn, tức giảm ρ .

Với $\sigma^2 = 1$ và $B = 100$: nếu $\rho = 0.5$ thì phương sai còn 0.505; nếu $\rho = 0.1$ thì còn 0.109. Giảm tương quan có lợi hơn nhiều so với thêm cây. Khi $\rho = 0$ thì phương sai chia đều cho B : chụp 8 tấm cùng một cảnh tối rồi lấy trung bình từng điểm ảnh, nhiễu ngẫu nhiên triệt tiêu bớt còn cảnh đứng yên được giữ nguyên.



Thêm cây chỉ kéo phương sai về mức sàn $\rho\sigma^2$; giảm tương quan mới hạ được sàn.

HỎI NHANH

Lấy $\sigma^2 = 1$. Rừng A có 50 cây, $\rho = 0.1$. Rừng B có 500 cây, $\rho = 0.4$. Theo công thức trên, rừng nào có phương sai của trung bình nhỏ hơn?

- A A, vì tương quan thấp quan trọng hơn số cây
- B B, vì nhiều cây gấp mười lần
- C Bằng nhau, vì cả hai đều đã đủ nhiều cây
- D Chưa so được khi chưa biết σ^2 thật

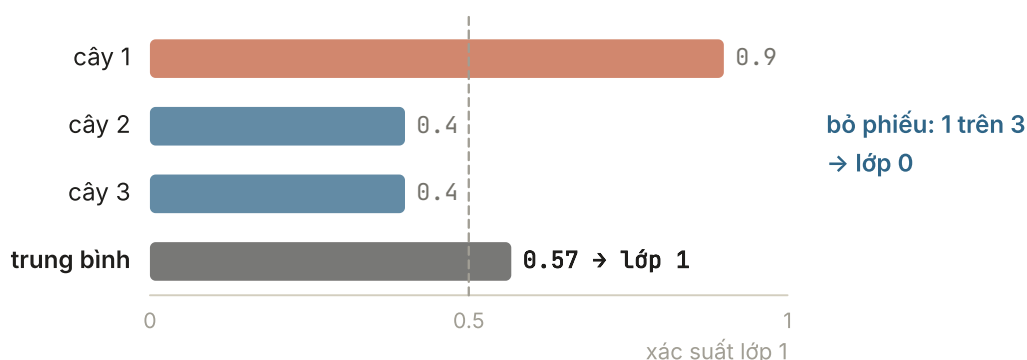
Hai nguồn ngẫu nhiên

Random forest (Breiman, 2001) làm các cây khác nhau bằng hai cách:

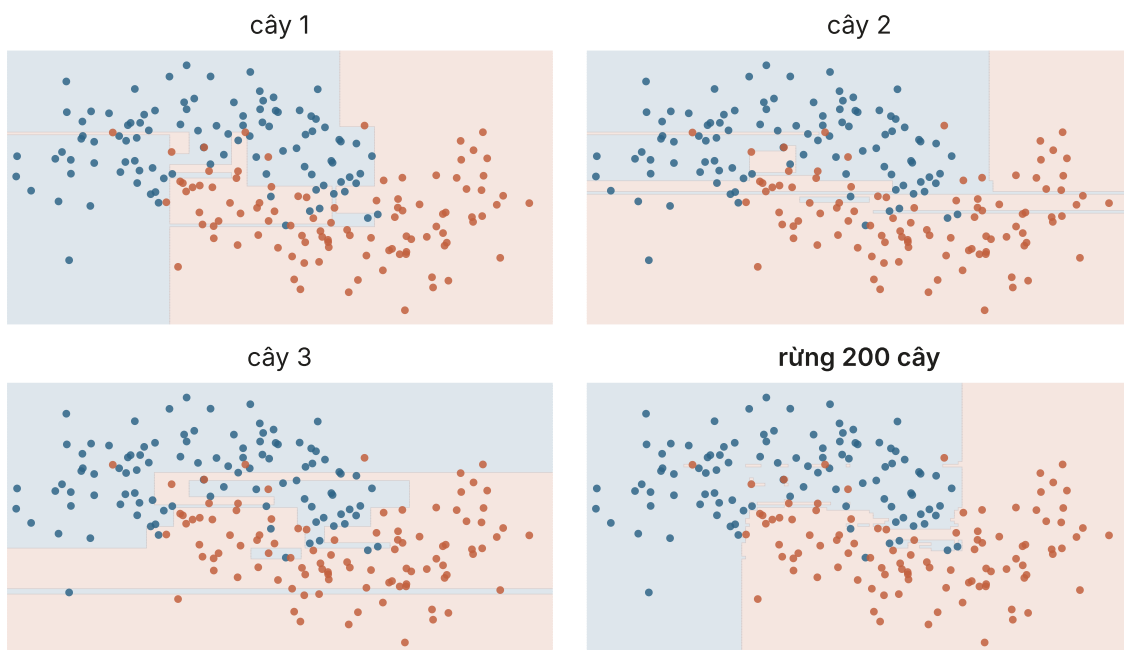
1. **Bagging** (bootstrap aggregating): mỗi cây học trên một tập bootstrap, tức n mẫu rút có hoàn lại từ n mẫu gốc.
2. **Chọn ngẫu nhiên đặc trưng** ở mỗi nút: chỉ xét một tập con ngẫu nhiên gồm max_features đặc trưng khi tìm phép chia tốt nhất. Nếu có một đặc trưng rất mạnh, bagging thuần túy vẫn cho mọi cây chia ở gốc bằng đặc trưng đó và các cây vẫn giống nhau. Che bớt đặc trưng buộc các cây phải tìm đường khác.

Mỗi cây được mọc sâu, gần như không cắt tỉa: bias thấp, variance cao, rồi trung bình hóa để kéo variance xuống. Phân loại thì lấy trung bình xác suất (hoặc bỏ phiếu); hồi quy thì lấy trung bình giá trị, $\frac{1}{B} \sum_b T_b(\mathbf{x})$. Các cây học độc lập, có thể học song song; boosting ở bài 2.7 thì dựng cây nối tiếp, cây sau sửa lỗi cây trước.

Hai cách gộp không luôn đồng ý. Ba cây cho xác suất lớp 1 của cùng một mẫu là 0.9, 0.4, 0.4. Trung bình xác suất là $(0.9 + 0.4 + 0.4)/3 \approx 0.57$, dự đoán lớp 1; bỏ phiếu thì chỉ 1 trên 3 cây chọn lớp 1, dự đoán lớp 0. Một cây rất chắc chắn thắng hai cây lưỡng lự. scikit-learn dùng cách thứ nhất, và khi cần xác suất để tính log loss hay AUC thì đó cũng là cách nên dùng.



Trung bình xác suất chọn lớp 1; bỏ phiếu chọn lớp 0.



Mỗi cây học một tập bootstrap nên ranh giới khác nhau; trung bình 200 cây mượt hơn.



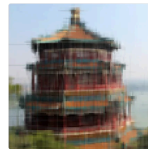
trung bình 1 cây: RMSE điểm chưa thấy 33.3 (một cây sâu không bootstrap: 25.5)



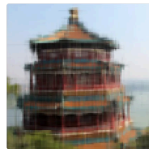
trung bình 5 cây: RMSE điểm chưa thấy 24.0



trung bình 20 cây: RMSE điểm chưa thấy 22.1



trung bình 100 cây: RMSE điểm chưa thấy 21.5



Random forest trên cùng bài đoán màu: mỗi cây học một mẫu bootstrap nên sai theo kiểu riêng; lấy trung bình nhiều cây thì ảnh mượt hơn và sai số trên điểm chưa thấy giảm.

HỎI NHANH

“Bootstrap n mẫu từ n mẫu gốc” nghĩa là gì?

- A Rút n lần không hoàn lại, tức là xáo trộn thứ tự
- B Rút n lần có hoàn lại từ n mẫu gốc
- C Giữ mỗi mẫu đúng một lần, chỉ đổi trọng số
- D Rút ngẫu nhiên $n/2$ mẫu, không lặp lại

Out-of-bag: validation miễn phí

Xác suất một mẫu không được chọn lần nào trong n lần rút có hoàn lại là

$$\left(1 - \frac{1}{n}\right)^n \xrightarrow{n \rightarrow \infty} e^{-1} \approx 0.368.$$

Vậy mỗi cây “chưa thấy” khoảng 37% dữ liệu. Dự đoán mỗi mẫu bằng các cây không dùng nó, ta được điểm out-of-bag (OOB): một ước lượng gần như cross-validation mà không tốn thêm lần huấn luyện nào. Mỗi mẫu nhận phiếu đa số của riêng các cây không thấy nó; lỗi OOB là tỉ lệ mẫu bị đoán sai, tính theo mẫu chứ không theo từng phiếu.

Ví dụ 5 mẫu, đánh số 1 đến 5, rừng có 3 cây:

Cây	Tập bootstrap	Mẫu out-of-bag
1	1, 1, 3, 4, 4	2, 5
2	2, 2, 3, 5, 5	1, 4
3	1, 2, 2, 4, 5	3

Điểm OOB của mẫu 1 chỉ do cây 2 chấm, mẫu 3 chỉ do cây 3 chấm, và cứ thế. Với $n = 5$, xác suất một mẫu nằm ngoài một tập bootstrap là $(4/5)^5 \approx 0.33$, đã khá gần 0.368.

	mẫu 1	mẫu 2	mẫu 3	mẫu 4	mẫu 5
cây 1	×2	OOB	×1	×2	OOB
cây 2	OOB	×2	×1	OOB	×2
cây 3	×1	×2	OOB	×1	×1

Ô nét đứt là mẫu cây đó chưa thấy, dùng để chấm OOB.

HỎI NHANH

Train có 1000 mẫu. Với một cây trong rừng, khoảng bao nhiêu mẫu nằm ngoài tập bootstrap của nó, và trong tập ấy có khoảng bao nhiêu mẫu **khác nhau**?

- A 0 mẫu nằm ngoài, 1000 mẫu khác nhau
- B 500 mẫu nằm ngoài, 500 mẫu khác nhau
- C Khoảng 632 mẫu nằm ngoài, khoảng 368 mẫu khác nhau
- D Khoảng 368 mẫu nằm ngoài, khoảng 632 mẫu khác nhau

Các tham số

Tham số	Ý nghĩa	Gợi ý
n_estimators	Số cây	Nhiều cây hơn thường bớt dao động; đo validation, RAM và thời gian trước
max_features	Số đặc trưng xét ở mỗi nút	Phân loại thường $\sqrt{d}$; nhỏ hơn thì cây đa dạng hơn
min_samples_leaf	Số mẫu tối thiểu ở lá	1-5; tăng lên khi dữ liệu nhiều
max_depth	Độ sâu	Thường để không giới hạn

Biến thể Extra Trees (extremely randomized trees) chọn luôn ngưỡng ngẫu nhiên thay vì tìm ngưỡng tốt nhất: nhanh hơn, cây còn khác nhau hơn nữa, đôi khi tốt hơn trên dữ liệu nhiễu.

HỎI NHANH

Muốn từng cây chính xác hơn, một bạn đặt max_features bằng toàn bộ d feature. Lỗi tổng quát của rừng có xu hướng thế nào?

- A Giảm, vì từng cây có lỗi tổng quát thấp hơn
- B Tăng, vì mỗi cây không có đủ dữ liệu train
- C Tăng, vì các cây trở nên rất giống nhau
- D Không đổi, vì bootstrap đã đủ tạo khác biệt

PHÒNG THEO CMU 10-601 · Exam 3 Practice · Q6.2(c)

Tự nhìn một lần bootstrap

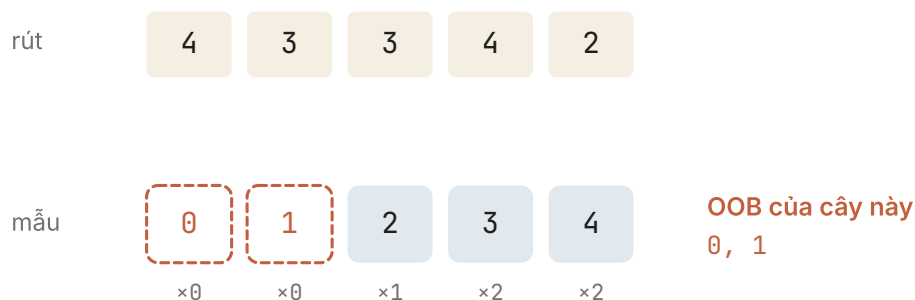
Đừng hiểu bootstrap là lấy một tập con không lặp. Với n mẫu, mỗi cây rút n lần **có hoàn lại**: một mẫu có thể xuất hiện nhiều lần, mẫu khác không xuất hiện lần nào. Ta nhìn ngay một lượt rút nhỏ trước khi dùng cả rừng.

Python

```
import numpy as np

rng_boot = np.random.default_rng(7)
draw = rng_boot.integers(0, 5, size=5)
seen = np.unique(draw)
oob = np.setdiff1d(np.arange(5), seen)
print("rút:", draw.tolist())
print("đã thấy:", seen.tolist(), "OOB:", oob.tolist())
probabilities = np.array([0.9, 0.4, 0.4])
print("trung bình:", round(float(probabilities.mean()), 3))
print("phiếu lớp 1:", int(np.sum(probabilities > 0.5)))
```

Đọc dòng “đã thấy”: số mẫu phân biệt có thể nhỏ hơn 5 dù đã rút đủ 5 lần. Với nhiều cây, mẫu 0 chỉ được chấm OOB bằng những cây không thấy mẫu 0. Có một cây không dùng nó không đủ để điểm OOB ổn định; cần nhiều cây và phải kiểm cảnh báo mẫu thiếu dự đoán OOB.



Một lượt bootstrap: mẫu 3, 4 lặp hai lần; mẫu 0, 1 không được rút nên là OOB.

HỎI NHANH

Một cây có tập bootstrap gồm các chỉ số [0, 0, 2, 3, 3], lấy từ dữ liệu có 5 mẫu. Những mẫu nào là OOB?

- A Mẫu 0 và 3, vì bị rút lặp
- B Mẫu 1 và 4
- C Chỉ mẫu 4, vì mẫu cuối không được rút
- D Không có, vì đã rút đủ 5 lần

Trung bình giảm nhiễu có điều kiện

Công thức phương sai ở trên giả sử các dự đoán có cùng phương sai σ^2 và mọi cặp có cùng tương quan ρ . Nó là một mô hình để thấy tác dụng của sự đa dạng, không phải cách dự đoán chính xác metric của bất kỳ rừng nào. Với các dự đoán thật, ta không được tự gán $\rho = 0$ chỉ vì các cây được train bằng seed khác nhau: chúng vẫn dùng chung một nguồn dữ liệu và cùng những đặc trưng mạnh.

Random forest bớt biến động nhờ trung bình hóa, nhưng không chữa được việc toàn bộ dữ liệu train sai phân phối hay một cột rò rỉ target. Tăng số cây chủ yếu làm dự đoán ổn định hơn; validation có thể dao động và một rừng lớn vẫn overfit. Tăng `min_samples_leaf` hoặc hạn chế độ sâu làm từng cây mượt hơn, còn giảm `max_features` làm các cây đa dạng hơn nhưng cũng có thể làm từng cây yếu đi. Ta chọn cân bằng bằng validation.

HỎI NHANH

100 cây được học với 100 seed khác nhau. Có thể coi dự đoán của chúng là độc lập, tức $\rho = 0$, để tính phương sai theo công thức trên không?

- A Được, vì seed khác nhau thì bootstrap khác nhau
- B Được, nếu max_features đủ nhỏ
- C Không được, vì seed không ảnh hưởng gì tới cây
- D Không được, vì các cây học chung dữ liệu và chung những tín hiệu mạnh

OOB rất tiện với các mẫu độc lập cùng phân phối. Nếu cùng một học sinh có nhiều bản ghi, cây không thấy một bản ghi nhưng có thể thấy các bản ghi khác của người đó. OOB theo hàng lúc này không thay thế group split. Với dự báo tương lai, bootstrap cũng không thay thế validation theo thời gian. Một lần chạy có OOB score cao chưa chứng minh mô hình dùng được ngoài lớp học.

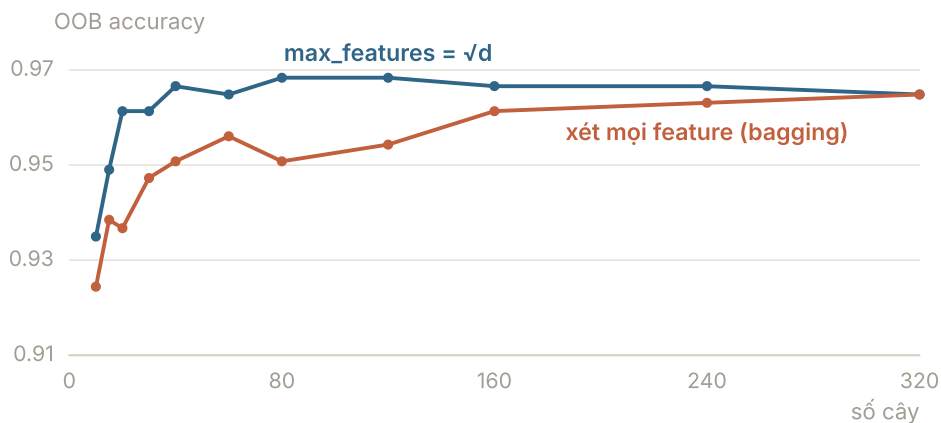
Dùng thư viện

Python

```
from sklearn.datasets import load_breast_cancer
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier

X, y = load_breast_cancer(return_X_y=True)
tree = DecisionTreeClassifier(random_state=0)
forest = RandomForestClassifier(n_estimators=80, oob_score=True, n_jobs=1, random_state=0)
print("một cây:", round(cross_val_score(tree, X, y, cv=5).mean(), 3))
print("rừng:   ", round(cross_val_score(forest, X, y, cv=5).mean(), 3))
print("OOB:    ", round(forest.fit(X, y).oob_score_, 3))
```

Rừng hơn một cây khoảng 4.6 điểm CV. OOB 0.968 và CV 0.963 là hai ước lượng khác nhau của cùng một thứ, và ở đây chúng gần nhau.



OOB accuracy theo số cây: sau khoảng 80 cây gần như đứng yên.

HỎI NHANH

Điểm OOB của rừng là 0.968, còn cross-validation 5 fold cho 0.963. Hiểu chênh lệch này thế nào?

- A OOB bị rò rỉ, vì cao hơn CV
- B CV sai, vì OOB dùng nhiều dữ liệu hơn
- C Hai cách lấy mẫu khác nhau, chênh nhỏ như vậy là bình thường
- D Phải chọn số cây sao cho hai con số bằng nhau

Khi đi thi

- Bài dữ liệu bảng thì chạy random forest sớm làm baseline: không cần chuẩn hóa, ít khi hỏng.
- Điểm OOB chỉ đáng tin khi các hàng độc lập; dữ liệu theo nhóm hay theo thời gian thì dùng cách chia riêng.

- Thêm cây mà validation đứng yên thì thử `max_features` hoặc feature mới, đừng tăng cây tiếp.

Tóm tắt

- Trung bình hóa nhiều mô hình giảm variance, nhưng chỉ khi chúng ít tương quan.
- Random forest = bagging + chọn ngẫu nhiên đặc trưng ở mỗi nút; mỗi cây sâu, cả rừng ổn định.
- Khoảng 37% dữ liệu nằm ngoài mỗi tập bootstrap, cho điểm OOB miễn phí.
- Thêm cây chủ yếu ổn định việc trung bình hóa, không thay thế kiểm validation; tham số quan trọng nhất là `max_features` và `min_samples_leaf`.

Kiểm tra cuối bài

Mười câu kiểm cả cách hiểu, phép tính và quy trình. Câu sai chỉ ra phần cần đọc lại; hãy làm trước khi mở lời giải.

CÂU 1

Có $n = 4$ mẫu train. Một cây rút 4 lần độc lập, đều xác suất, có hoàn lại. Xác suất một mẫu cụ thể không được rút lần nào là bao nhiêu?

A $(1/4)^4$

B 0

C $(3/4)^4$

D $1/4$

CÂU 2

Random forest chuẩn có dùng bootstrap. So với bagging các cây xét mọi feature ở mỗi nút, random forest giảm tương quan giữa các cây bằng cách nào?

- A Ở mỗi nút, chỉ xét một tập con feature chọn ngẫu nhiên
- B Bắt mọi cây học cùng một tập bootstrap
- C Lấy dự đoán của cây trước làm target cho cây sau
- D Chọn một feature duy nhất dùng chung cho cả rừng

CÂU 3

Giả sử B cây có phương sai dự đoán $\sigma^2 = 4$, và mọi cặp cây có cùng tương quan $\rho = 0.25$. Với $B = 20$, phương sai của trung bình dự đoán bằng bao nhiêu?

- A 0.20
- B 1.00
- C 4.00
- D 1.15

CÂU 4

Rừng 5 cây học trên 7 mẫu và gộp kết quả bằng đa số phiếu. Bảng ghi dự đoán của những cây không thấy từng mẫu.

Mẫu	Thật	Dự đoán OOB của các cây không thấy mẫu
1	Không	Không
2	Không	Không, Có, Có
3	Có	Có
4	Có	Có
5	Không	Không
6	Có	Không
7	Có	Không, Không

Lỗi OOB của rừng bằng bao nhiêu?

A 5/10

B 3/7

C 2/7

D 4/7

PHÒNG THEO CMU 10-601 · Exam 3 Practice · Q6.2(a)

CÂU 5

Trong bảng của câu trước, có một cây đúng mọi mẫu OOB của nó, và một cây khác sai mọi mẫu. Khi dự đoán một mẫu mới, random forest chuẩn cho cây nào trọng số cao nhất?

- A Cây đúng mọi mẫu OOB
- B Cây có nhiều mẫu OOB nhất
- C Không cây nào, các cây góp phần ngang nhau
- D Cây sai mọi mẫu, nhưng đảo ngược dự đoán của nó

PHÒNG THEO CMU 10-601 · Exam 3 Practice · Q6.2(b)

CÂU 6

Rừng hồi quy học giá nhà từ những căn có giá 1 đến 5 tỷ. Một căn mới rộng gấp đôi căn lớn nhất trong train. Rừng sẽ dự đoán ra sao?

- A Khoảng 10 tỷ, vì rừng kéo dài xu hướng theo diện tích
- B Không quá 5 tỷ, vì đó là trung bình của các lá
- C Số nào cũng có thể, vì mỗi cây kéo dài theo một hướng riêng
- D Báo lỗi, vì giá trị nằm ngoài vùng train

CÂU 7

Mỗi bệnh nhân có 4 bản ghi, và mục tiêu là dự đoán cho bệnh nhân mới. Nhưng OOB tính theo hàng vẫn để một bản ghi được chấm bởi những cây đã học các bản ghi khác của cùng người. Ta nên đánh giá bổ sung thế nào?

- A Để riêng theo bệnh nhân, rồi chấm cả quy trình
- B Chỉ cần tăng số cây, để hàng nào cũng có nhiều dự đoán OOB hơn
- C Chỉ cần tăng `min_samples_leaf`, để cấm trùng bệnh nhân
- D Dùng `StratifiedKFold` theo từng hàng thay cho chia theo nhóm

CÂU 8

Tăng `n_estimators` từ 100 lên 1000. Accuracy trên train tăng từ 99.2% lên 99.8%, trên validation vẫn đứng ở 87.3%, còn thời gian chạy tăng gấp 10. Nhận định nào phù hợp?

- A Validation đứng yên chứng tỏ phải tăng độ sâu của cây
- B Cứ tăng số cây tới khi train đạt 100%, rồi chọn theo train
- C Train tăng 0.6 điểm, nên sau này validation cũng sẽ tăng
- D Lợi ích của việc thêm cây đã bão hòa, nên thử yếu tố khác

PHÒNG THEO [NOAI 2026 Singapore](#) · [Chung kết](#) · [Câu 2](#)

CÂU 9

`classes_` của rừng là `["fraud", "normal"]`, và metric cần score của lớp fraud. Dòng nào lấy đúng score ấy?

- A `predict_proba(X)[:, 0]`
- B `predict_proba(X)[:, 1]`
- C `predict(X) == "fraud"`, để giữ đủ thứ tự score
- D Lấy cột có nhiều score trên 0.5 hơn

CÂU 10

Rừng tìm ra một cột chứa sẵn đáp án của validation, và điểm rất cao. Điểm cao ấy chứng tỏ điều gì?

- A Rừng đã học được quy luật tốt hơn mọi mô hình khác
- B Gần như chắc chắn có rò rỉ, phải sửa dữ liệu
- C Nên thêm cây cho điểm ổn định hơn nữa
- D Nên tăng `max_features` để cột ấy được dùng nhiều hơn

Đọc thêm

- James, Witten, Hastie, Tibshirani — [An Introduction to Statistical Learning](#), mục 8.2. Bagging, random forest và OOB với ví dụ đọc kết quả.
- Géron — *Hands-On Machine Learning*, chương 7. Ensemble, bagging, random forest và Extra Trees bằng scikit-learn.
- Galton (1907) — *Vox Populi*, Nature 75. Câu chuyện đám đông đoán cân nặng con bò: trung bình nhiều phỏng đoán độc lập chính xác hơn từng người.
- [Breiman — Random Forests](#). Bài gốc về random forest, sự đa dạng và out-of-bag.

- [scikit-learn — Forests of randomized trees](#). Tham số, phân loại bằng trung bình xác suất và chi phí song song.