RANDOM FOREST AND ADABOOST
Prepare for lessons

Nguyễn Thảo Nhi

Random forest và AdaBoost là một trong những phương pháp giải quyết các nhược điểm của Decision Tree

1. **Random Forest:**

- Trong xác suất thống kê, việc khai thác thông tin từ 2 cột trở lên là điều rất là khó khăn và phức tạp. Machine Learning ra đời nhằm mục đích giải quyết nhiều vấn đề tồn đọng mà xác suất thống kê không giải quyết được.

- Random forest là một mô hình học máy hợp nhất nhiều cây quyết đinh để đưa ra dự đoán chính xác hơn. Điều này giúp mô hình tổng thể chắc chắn và chính xác hơn.

- Random forest là một cách lấy trung bình nhiều cây quyết định, được huấn luyện trên các tập con khác nhau của cùng một tập dữ liệu **mục tiêu làm giảm phương sai**
  Ví dụ: Khi cần dự đoán người này thấp hay cao, thì ta chỉ cần xem xét chiều cao của đối tượng đó.

Hình 1: Dự đoán chiều cao của con người

Trong trường hợp bài toán phức tạp như dự đoán chứng khoán hoặc dự đoán động vật, kết quả cần phải được dự đoán dựa vào rất nhiều thông tin.

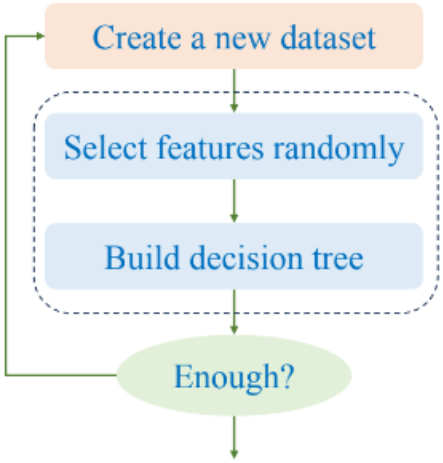
(a) Dự đoán động vật

(b) Dự đoán chứng khoán

Hình 2: Dự đoán các bài toán phức tạp

Type	Concepts
Classification	`GINI Impurity, Information Gain (Entropy)`
Regression	`MSE or SSR`

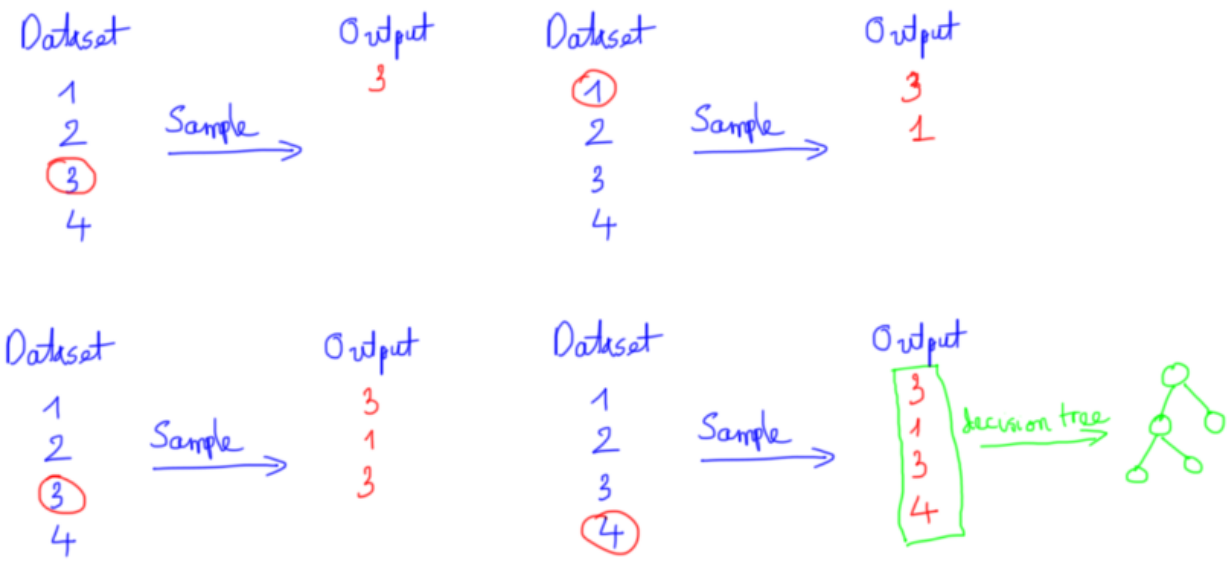
2. **Random Forset - Classification**



Hình 3: Các bước cho bài toán RF-Classification

"Random forest with random features is formed by selecting at random, at each node, a small group of input variables to split on." *Breiman writes*

- **Step 1:** Tạo một dataset mới lấy từ tập dataset ban đầu để huấn luyện mô hình và cho phép lặp lại sample (boostrapping)



Hình 4: Kỹ thuật Boostrapping

- **Step 2:** Sau đấy chọn ra ngẫu nhiên ở k thuộc tính (k< số lượng thuộc tính của dataset ban đầu). Đến đây mình có được bộ dữ liệu mới có k thuộc tính.
- **Step 3:** Dùng thuật toán Decision Tree dể xây dựng cây quyết định dự trên dữ liệu mới ở step 3
- **Step 4:** Quay lại step 1 để tiếp tục

Ví dụ: Trong bài toán Iris

Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1
1.8	0.9	1
1.2	1.3	1

(a) Dataset ban đầu

Petal_Length	**Petal_Width**	**Label**
1	0.2	0
1.3	0.6	0
1	0.2	0
1.8	0.9	1
1.8	0.9	1
1.2	1.3	1

(b) Tạo dataset mới ứng dụng kỹ thuật Boostrapping

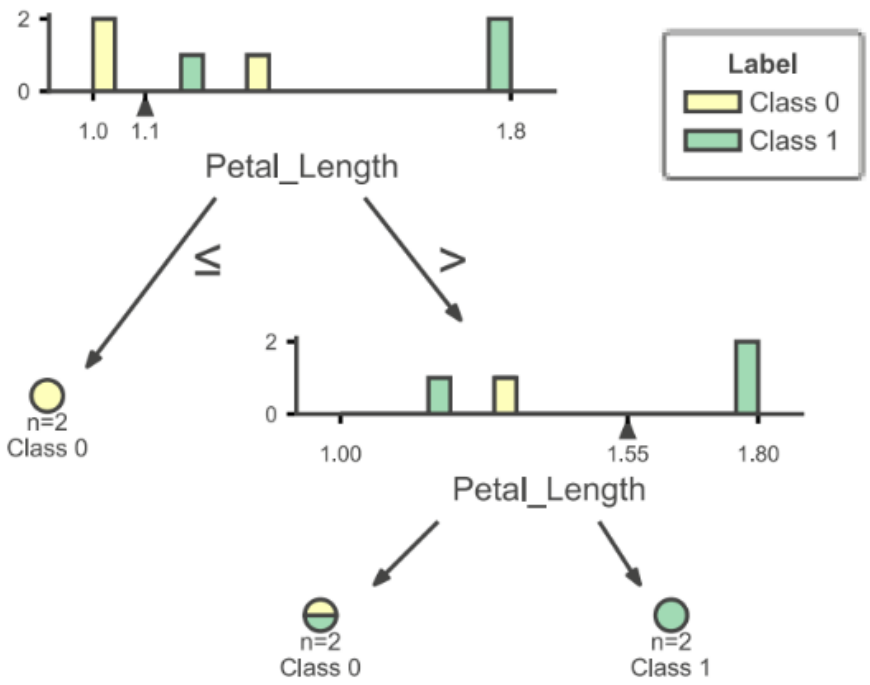
Petal_Length	**Label**
1	0
1.3	0
1	0
1.8	1
1.8	1
1.2	1

(c) Chọn k thuộc tính

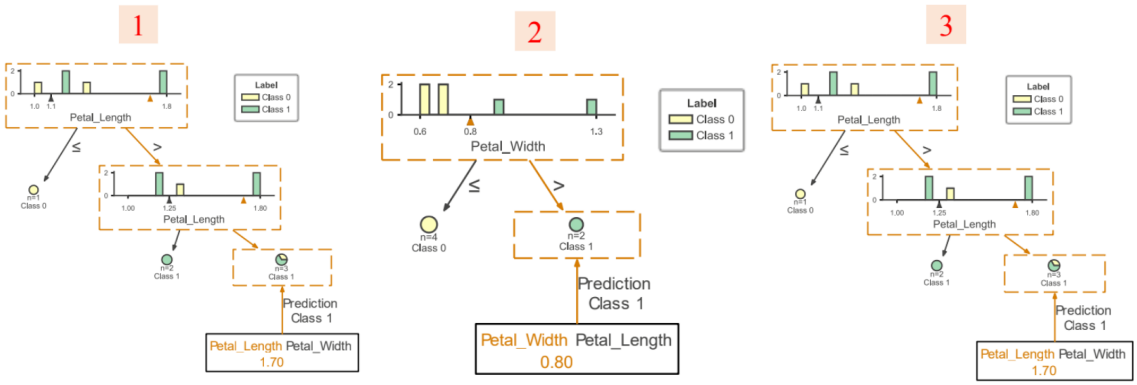
Hình 5: Step 01: Tạo dataset mới

Step 3: Xây dựng cây với tập dữ liệu vừa mới tạo

Step 4: Quay lại step 1 để tiếp tục Giả sử bài toán muốn xây dựng Random Forest với số lượng cây là 3 thì chúng ta sẽ xây dựng được 3 cây quyết định như sau:



Hình 6: Xây dựng cây quyết định với tập dữ liệu mới



(a) Cây quyết định 01 (b) Cây quyết định 02 (c) Cây quyết định 03

Hình 7: Kết quả 3 cây quyết định

Sau đấy dựa vào sự đoán của 3 cây quyết định trên để Voting kết quả cuối cùng.

3. **Random Forset - Regression**

- **Step 1:** Tạo một dataset mới lấy từ tập dataset ban đầu để huấn luyện mô hình và cho phép lặp lại sample (boostrapping)

Experience	Salary
1	0
1.5	0
2	0
2.5	0
3	60
3.5	64
4	55
4.5	61
5	66
5.5	83
6	93
6.5	91
7	98
7.5	101

(a) Dữ liệu ban đầu

Experience	Salary
1.5	0
2	0
2.5	0
3	60
3.5	64
4	55
4.5	61
5	66
5.5	83
6	93
6.5	91
7	98
7.5	101

(b) Dữ liệu tạo mới

Hình 8: Tạo dữ liệu mới trong bài toán Regression

- **Step 2:** Tuy nhiên ở bài toán Regression chúng mình sẽ có cách tách khác hơn so với bài toán Classification)

Experience	Salary	
1	0	$a_{mse} = 1184.07$
1.5	0	$a_{mse} = 911.19$
2	0	$a_{mse} = 588.68$
2.5	0	$a_{mse} = 201.68$
3	60	$a_{mse} = 383.92$
3.5	64	$a_{mse} = 526.52$
4	55	$a_{mse} = 543.51$
4.5	61	$a_{mse} = 575.09$
5	66	$a_{mse} = 613.34$
5.5	83	$a_{mse} = 758.4$
6	93	$a_{mse} = 947.73$
6.5	91	$a_{mse} = 1090.05$
7	98	$a_{mse} = 1256.21$
7.5	101	

Hình 9: Dùng MSE trong bài toán Regression

- **Step 3:** Dùng thuật toán Decision Tree để xây dựng cây quyết định dự trên dữ liệu mới ở step 3
- **Step 4:** Quay lại step 1 để tiếp tục **Lưu ý:** Với bài toán Regression mình sẽ trả về kết quả cuối cùng bằng cách tính Average/Mean/Median kết quả của các cây trả về.

4. **Bernoulli Random Variables**

- Cách thức mô tả một sự kiện nào đó xảy ra, sự kiện này chỉ có giá trị 1 hoặc 0.

$$0 \leq p \leq 1 \begin{cases} q = 1 - p & \text{if } k = 0 \\ p & \text{if } k = 1 \end{cases}$$

`Giả sử có 10 viên bi, trong đó có 9 viên đi đỏ và 1 viên bi vàng. Xác suất lấy bi đỏ là` $\frac{1}{N}$ `và xác suất KHÔNG lấy bi vàng là` $1 - \frac{1}{N}$`. Sau khi chọn xong ta lại quay lại ban đầu để chọn tiếp, từ đây ta có phép nhân giữa các lần chọn.`

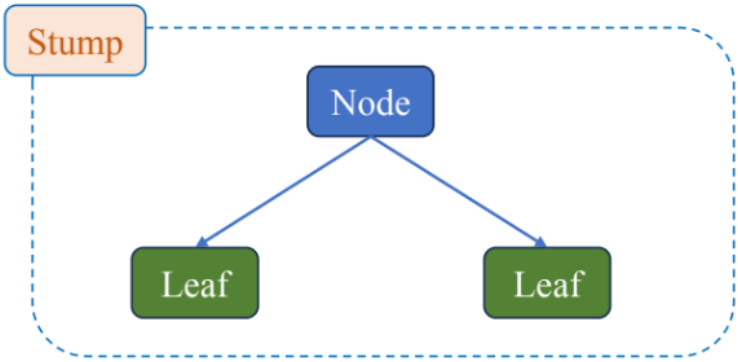
$\Rightarrow$ **Số lượng dữ liệu tham gia trong tập con mới là 2/3 trên tập ban đầu.**

$$p(x, n, k) = \binom{n}{k} \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k}$$

5. **Adaptive Boosting**

- Vấn đề overfitting trong cây không chỉ mỗi phụ thuộc và đặc trưng dữ liệu mà còn là độ sâu của cây quyết định dẫn đến bài toán overfitting.

⇒ Xây dựng cây quyết định có chiều sâu bằng 1



Hình 10: Stump

Dữ liệu của stump sẽ được xây dựng từ những dữ liệu được mô hình dự đoán sai trước đó

Petal_Length	Petal_Width	Label	Evaluation
1	0.2	0	T
1.3	0.6	0	F
0.9	0.7	0	T
1.7	0.5	1	T
1.8	0.9	1	F
1.2	1.3	1	T

(a) Kết quả dự đoán mô hình trước

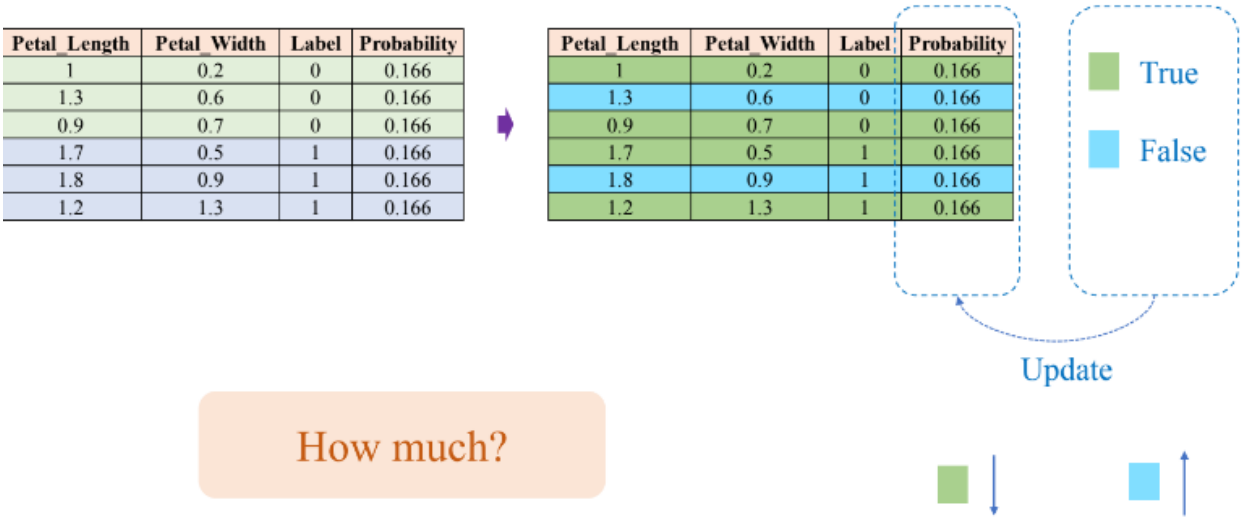
Petal_Length	Petal_Width	Label	Evaluation
1.3	0.6	0	F
1.3	0.6	0	F
1.8	0.9	1	F
1.8	0.9	1	F

(b) Những sample dự đoán sai ta khuếch đại số lượng để học

Hình 11: Tạo dữ liệu mới

Petal_Length	Petal_Width	Label	Evaluation	Score	Probability
1	0.2	0	T	1	0.125
1.3	0.6	0	F	2	0.25
0.9	0.7	0	T	1	0.125
1.7	0.5	1	T	1	0.125
1.8	0.9	1	F	2	0.25
1.2	1.3	1	T	1	0.125

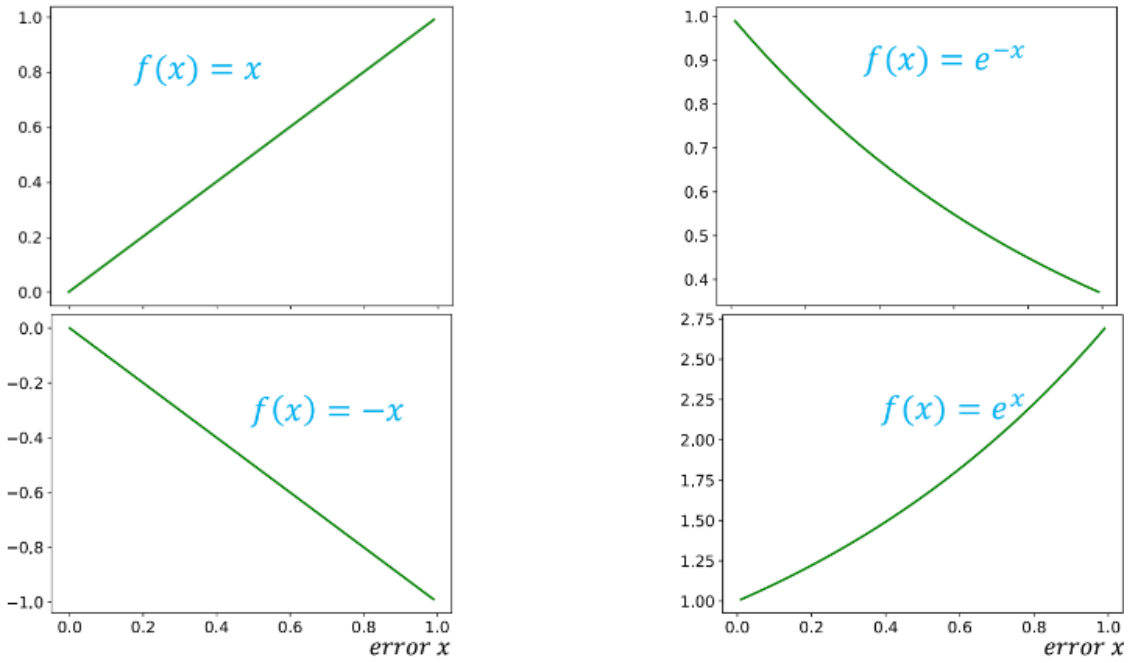
Hình 12: Normalize - quy về xác suất để sample đó được chọn cho bước tiếp theo



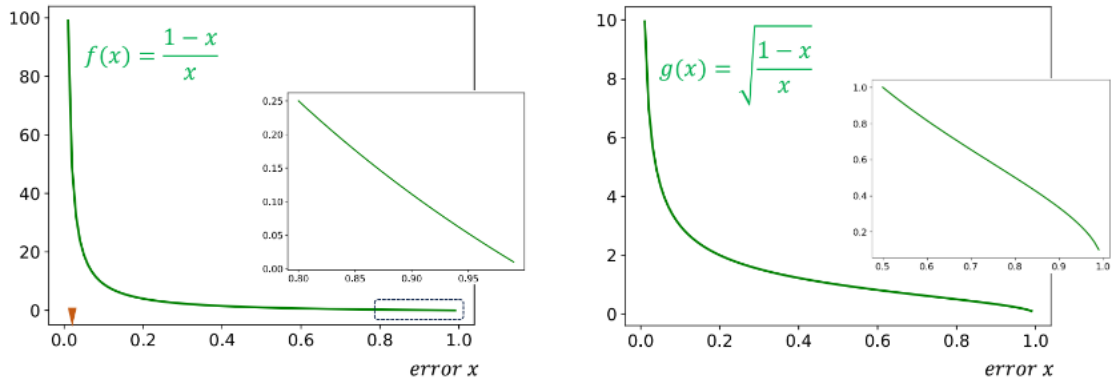
Hình 13: Mỗi lần xây dựng stump ta sẽ xem xét liệu mô hình đang cải thiện/sai bao nhiêu?

Cách tăng/giảm xác suất chọn sample cho tập con tiếp theo.

- **Với incorrect samples**

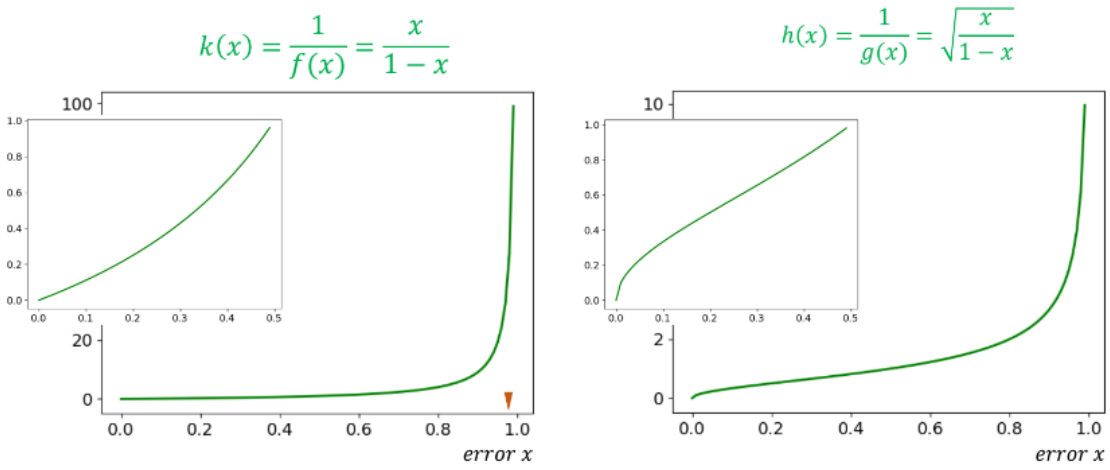


Hình 14: Khi mô hình tốt (lỗi nhỏ), trọng lượng được chia tỷ lệ sẽ tăng/giảm nhẹ/đáng kể?

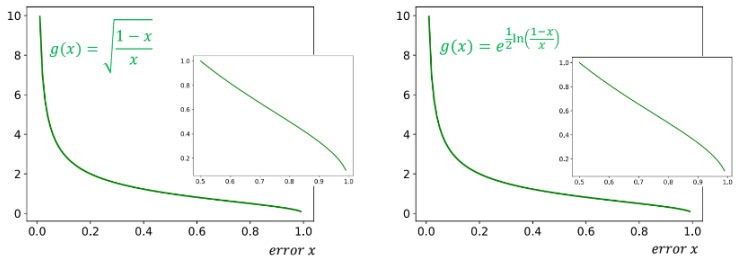


Hình 15: Khi mô hình tốt (lỗi nhỏ), tăng đáng kể

- **Với correct samples**



Hình 16: Giảm đáng kể



Hình 17: Tăng đáng kể

6. **Sử dụng Sklearn**

```
dt_classifier = AdaBoostClassifier(n_estimators=3)
dt_classifier.fit(x_data, y_train)
```

Hình 18: Dùng thư viện sklearn với lệnh sau

-END-