Cây quyết định (Decision Tree) là gì? Tìm hiểu Thuật toán ID3

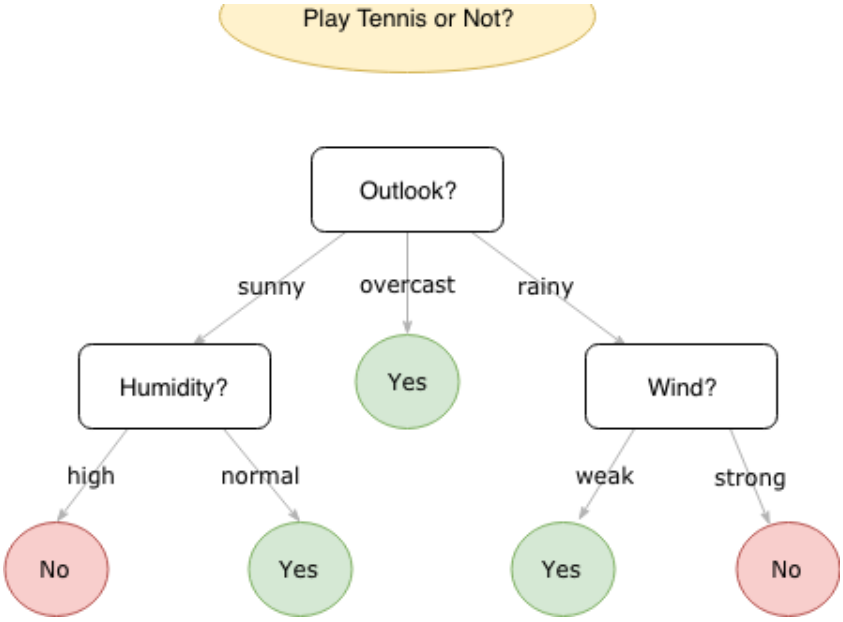
Mục lục

Định nghĩa

Cây quyết định (*Decision Tree*) là một mô hình thuộc nhóm thuật toán Học có giám sát (*Supervised Learning*).
Tìm hiểu thêm về phân loại các thuật toán Machine Learning tại đây.

Cây quyết định là gì?

Cây quyết định (gọi tắt là *DT*) là mô hình đưa ra quyết định dựa trên các câu hỏi.
Dưới đây là mô hình DT về một ví dụ kinh điển.
Câu hỏi có chơi tennis hay không? Quyết định đưa ra dựa trên các yếu tố về thời tiết: outlook, humidity, wind.



DT được áp dụng vào cả 2 bài toán: Phân loại (*Classification*) và Hồi quy (*Regression*). Tuy nhiên bài toán phân loại được sử dụng nhiều hơn.

Có nhiều thuật toán để xây dựng DT, trong bài này ta tìm hiểu một thuật toán nổi tiếng và cơ bản nhất của DT là thuật toán ID3.

Thuật toán ID3

Iterative Dichotomiser 3 (ID3) là thuật toán nổi tiếng để xây dựng Decision Tree, áp dụng cho bài toán Phân loại (*Classification*) mà tất các các thuộc tính để ở dạng category.

Để dễ hiểu ta cùng tìm hiểu thuật toán này qua ví dụ.

Tìm hiểu qua ví dụ

Ta có tập Training Data như bảng dưới:

ID	Engine	Type	Color	4WD	Want?
1	2000cc	SUV	Silver	Yes	Yes
2	1000cc	Sedan	Silver	Yes	Yes
3	2000cc	Sport	Blue	No	No
4	1000cc	SUV	Blue	No	Yes
5	2000cc	Sedan	Silver	Yes	No
ID	Engine	Type	Color	4WD	Want?

6	2000cc	Sport	Blue	Yes	Yes
7	1000cc	Sedan	Blue	No	Yes
8	1000cc	SUV	Silver	No	Yes

Data của ta có 4 thuộc tính: Engine, Type, Color, 4WD.
Để tính toán được DT, ta cần phân chia các thuộc tính vào các node đánh giá. Vậy làm sao để biết được thuộc tính nào quan trọng, nên đặt ở gốc, thuộc tính nào ở nhánh...

Trong thuật toán ID3, các thuộc tính được đánh giá dựa trên Hàm số Entropy, hàm số phổ biến trong toán học xác suất.

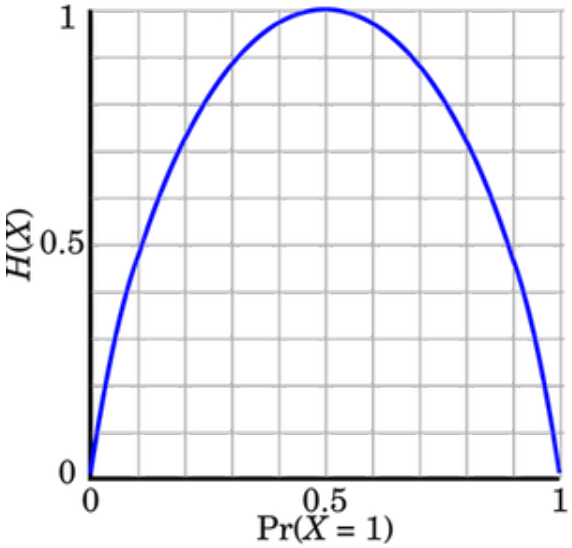
Hàm số Entropy

Cho một phân phối xác suất của một biến rời rạc x có thể nhận n giá trị khác nhau $x_1, x_2, \ldots, x_n$.
Giả sử rằng xác suất để x nhận các giá trị này là $p_i = p(x = x_i)$

Ký hiệu phân phối này là $\mathbf{p} = (p_1, p_2, \ldots, p_n)$.
Entropy của phân phối này là:

$$H(\mathbf{p}) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

Hàm Entropy được biểu diễn dưới dạng đồ thị như sau:



Từ đồ thị ta thấy, hàm Entropy sẽ đạt giá trị nhỏ nhất nếu có một giá trị $p_i = 1$, đạt giá trị lớn nhất

nếu tất cả các p_i bằng nhau.
Hàm Entropy càng lớn thì độ ngẫu nhiên của các biến rời rạc càng cao (càng không tinh khiết).

Với cây quyết định, ta cần tạo cây như thế nào để cho ta nhiều thông tin nhất, tức là Entropy là cao nhất.

Information Gain

Bài toán của ta trở thành, tại mỗi tầng của cây, cần chọn thuộc tính nào để độ giảm Entropy là thấp nhất.
Người ta có khái niệm **Information Gain** được tính bằng

$$Gain(S, f) = H(S) - H(f, S)$$

trong đó:
$H(S)$ là Entropy tổng của toàn bộ tập data set S.
$H(f, S)$ là Entropy được tính trên thuộc tính f.

Do $H(S)$ là không đổi với mỗi tầng, ta chọn thuộc tính f có Entropy nhỏ nhất để thu được $Gain(S, f)$ lớn nhất.

Tính Entropy của các thuộc tính

Xét thuộc tính Engine

Thuộc tính này có thể nhận 1 trong 2 giá trị 1000cc, 2000cc, tương ứng với 2 child node.
Gọi tập hợp các điểm trong mỗi child node này lần lượt là S_1, S_2.

Sắp xếp lại theo thuộc tính **Engine** ta có 2 bảng nhỏ.

Engine 1000cc (S_1)

ID	Engine	Type	Color	4WD	Want?
2	1000cc	Sedan	Silver	Yes	Yes
4	1000cc	SUV	Blue	No	Yes
7	1000cc	Sedan	Blue	No	Yes
8	1000cc	SUV	Silver	No	Yes

Engine 2000cc (S_2)

ID	Engine	Type	Color	4WD	Want?
1	2000cc	SUV	Silver	Yes	Yes
3	2000cc	Sport	Blue	No	No
5	2000cc	Sedan	Silver	Yes	No
6	2000cc	Sport	Blue	Yes	Yes

Child node ứng với Engine 1000cc sẽ có Entropy = 0 do tất cả các giá trị đều là **Yes**.
Ta chỉ việc tính Entropy với Engine 2000cc. Sau đó tính Entropy trung bình.
Cụ thể như sau:

$$
\begin{aligned}
H(S_1) &= 0 \\
H(S_2) &= -\frac{2}{4}log_2\left(\frac{2}{4}\right) - \frac{2}{4}log_2\left(\frac{2}{4}\right) = 1 \\
H(engine, S) &= \frac{4}{8}H(S_1) + \frac{4}{8}H(S_2) = 0.5
\end{aligned}
$$

Xét thuộc tính Type

Thuộc tính này có thể nhận 1 trong 3 giá trị SUV, Senda, Sport tương ứng với 3 child node.
Gọi tập hợp các điểm trong mỗi child node này lần lượt là S_u, S_e, S_p.

Sắp xếp lại theo thuộc tính **Type** ta có 3 bảng nhỏ.

Type SUV (S_u)

ID	Engine	Type	Color	4WD	Want?
1	2000cc	SUV	Silver	Yes	Yes
4	1000cc	SUV	Blue	No	Yes
8	1000cc	SUV	Silver	No	Yes

Type Sedan (S_e)

ID	Engine	Type	Color	4WD	Want?
2	1000cc	Sedan	Silver	Yes	Yes

2	1000cc	Sedan	Silver	Yes	Yes
5	2000cc	Sedan	Silver	Yes	No
7	1000cc	Sedan	Blue	No	Yes

Type Sport (S_p)

ID	Engine	Type	Color	4WD	Want?
3	2000cc	Sport	Blue	No	No
6	2000cc	Sport	Blue	Yes	Yes

Tương tự, ta lần lượt tính Entropy như bên dưới:

$$
\begin{aligned}
H(S_u) &= 0 \\
H(S_e) &= -\frac{2}{3}log_2\left(\frac{2}{3}\right) - \frac{1}{3}log_2\left(\frac{1}{3}\right) \approx 0.918 \\
H(S_p) &= -\frac{1}{2}log_2\left(\frac{1}{2}\right) - \frac{1}{2}log_2\left(\frac{1}{2}\right) = 1 \\
H(type, S) &= \frac{3}{8}H(S_u) + \frac{3}{8}H(S_e) + \frac{2}{8}H(S_p) \approx 0.594
\end{aligned}
$$

Xét thuộc tính Color

Thuộc tính này có thể nhận 1 trong 2 giá trị Silver, Blue tương ứng với 2 child node.
Gọi tập hợp các điểm trong mỗi child node này lần lượt là S_s, S_b.

Sắp xếp lại theo thuộc tính **Color** ta có 2 bảng nhỏ.

Color Silver (S_s)

ID	Engine	Type	Color	4WD	Want?
1	2000cc	SUV	Silver	Yes	Yes
2	1000cc	Sedan	Silver	Yes	Yes
5	2000cc	Sedan	Silver	Yes	No
8	1000cc	SUV	Silver	No	Yes

Color Blue (S_b)

ID	Engine	Type	Color	4WD	Want?

ID	Engine	Type	Color	4WD	Want?
3	2000cc	Sport	Blue	No	No
4	1000cc	SUV	Blue	No	Yes
6	2000cc	Sport	Blue	Yes	Yes
7	1000cc	Sedan	Blue	No	Yes

Dễ thấy, 2 giá trị Silver và Blue đều có tỉ lệ Yes, No như nhau là ¾ và ¼.

Do đó ta tính luôn Entropy trung bình:

$$H(color, S) = \quad -\frac{3}{4}log_2\left(\frac{3}{4}\right) - \frac{1}{4}log_2\left(\frac{1}{4}\right) \approx 0.811$$

Xét thuộc tính 4WD

Thuộc tính này có thể nhận 1 trong 2 giá trị Yes, No tương ứng với 2 child node.
Gọi tập hợp các điểm trong mỗi child node này lần lượt là S_y, S_n.

Sắp xếp lại theo thuộc tính **4WD** ta có 2 bảng nhỏ.

4WD Yes (S_y)

ID	Engine	Type	Color	4WD	Want?
1	2000cc	SUV	Silver	Yes	Yes
2	1000cc	Sedan	Silver	Yes	Yes
5	2000cc	Sedan	Silver	Yes	No
6	2000cc	Sport	Blue	Yes	Yes

4WD No (S_n)

ID	Engine	Type	Color	4WD	Want?
3	2000cc	Sport	Blue	No	No
4	1000cc	SUV	Blue	No	Yes
7	1000cc	Sedan	Blue	No	Yes
8	1000cc	SUV	Silver	No	Yes

Tương tự Color, ta tính Entropy trung bình:

$$H(4wd, S) = -\frac{3}{4}log_2\left(\frac{3}{4}\right) - \frac{1}{4}log_2\left(\frac{1}{4}\right) \approx 0.811$$

Chọn thuộc tính có Entropy nhỏ nhất

Sau khi tính Entropy trung bình của 4 thuộc tính ta thu được:
$H(engine, S) = 0.5$
$H(type, S) \approx 0.594$
$H(color, S) \approx 0.811$
$H(4wd, S) \approx 0.811$

Thuộc tính **Engine** có giá trị Entropy nhỏ nhất nên ta chọn là node đánh giá đầu tiên.
Với Engine 1000cc, tất cả các data đều có giá trị Yes, vì vậy ta thu được node là Yes ở nhánh 1000cc.
Ta tiếp tục tính cho nhánh Engine 2000cc với tập data nhỏ hơn là

ID	Engine	Type	Color	4WD	Want?
1	2000cc	SUV	Silver	Yes	Yes
3	2000cc	Sport	Blue	No	No
5	2000cc	Sedan	Silver	Yes	No
6	2000cc	Sport	Blue	Yes	Yes

Tương tự ta lần lượt tính Entropy cho 3 thuộc tính là: Type, Color, 4WD

Với thuộc tính **Type**:

$$\begin{aligned} H(S_u) &= 0 \\ H(S_e) &= 0 \\ H(S_p) &= -\frac{1}{2}log_2\left(\frac{1}{2}\right) - \frac{1}{2}log_2\left(\frac{1}{2}\right) = 1 \\ H(type, S) &= \frac{1}{4}H(S_u) + \frac{1}{4}H(S_e) + \frac{2}{4}H(S_p) = 0.5 \end{aligned}$$

Với thuộc tính **Color**:

Do 2 giá trị Silver và Blue có cùng tỉ lệ Yes, No là ½ và ½.

$$H(color, S) = -\frac{1}{2}log_2\left(\frac{1}{2}\right) - \frac{1}{2}log_2\left(\frac{1}{2}\right) = 1$$

Với thuộc tính **4WD**:

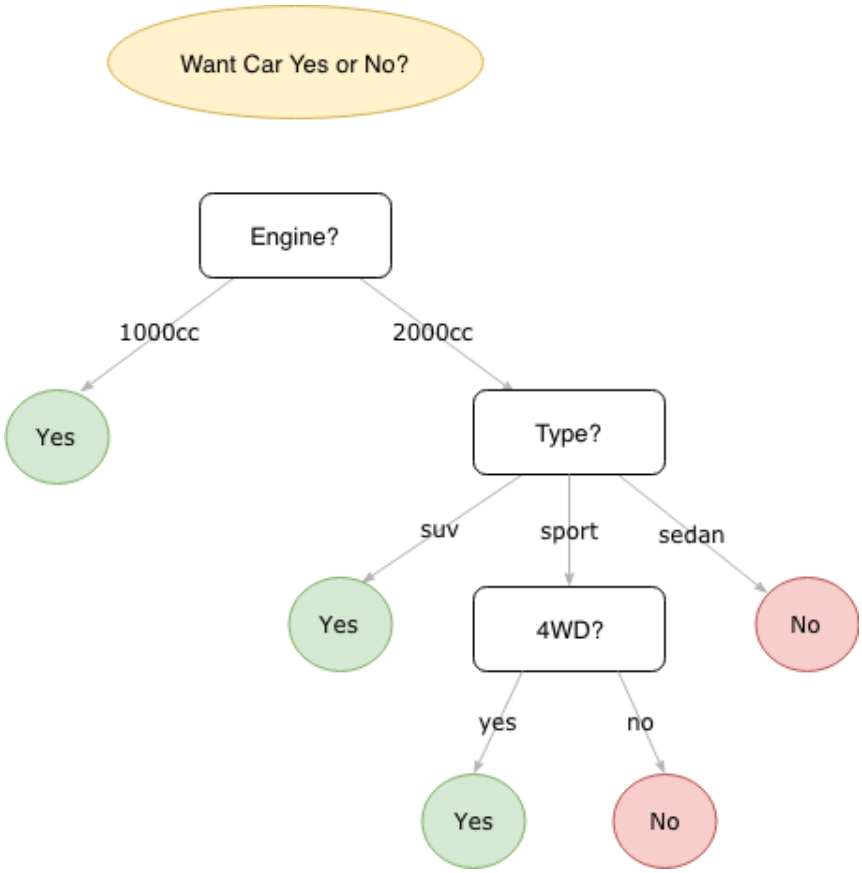
$$\begin{aligned} H(S_y) &= -\frac{2}{3}log_2\left(\frac{2}{3}\right) - \frac{1}{3}log_2\left(\frac{1}{3}\right) \approx 0.918 \\ H(S_n) &= 0 \\ H(4wd, S) &= \frac{3}{4}H(S_y) + \frac{1}{4}H(S_n) \approx 0.688 \end{aligned}$$

Vậy ta chọn **Type** là node đánh giá tiếp theo.

Với trường hợp Type là SUV hoặc Sedan, ta có ngay node lá vì chỉ có một kết quả.
Với trường hợp Type là Sport, do thuộc tính Color là giống nhau với tất cả data, ta chọn node đánh giá tiếp theo là **4WD**.

Kết quả

Ta thu được Decision Tree như hình bên dưới.



Kiểm tra (Validation)

Ta sẽ tiến hành kiểm tra mô hình DT ta vừa tạo được bằng tập Test Data như bên dưới:

ID	Engine	Type	Color	4WD	Want?
9	2000cc	Sedan	Silver	Yes	Yes
10	2000cc	Sport	Silver	No	No
11	1000cc	SUV	Blue	Yes	Yes
12	2000cc	Sedan	Blue	No	Yes

Ta có bảng mapping đánh giá kết quả như sau:

Actual Result	Estimate Result	Validation
Yes	Yes	True Positive (TP)
No	No	True Negative (TN)
Yes	No	False Negative (FN)
No	Yes	False Positive (FP)

Dựa vào DT ta vừa tạo được, ta tiến hành đánh giá như sau:

ID	Actual Result	Estimate Result	Validation
9	Yes	No	False Negative (FN)
10	No	No	True Negative (TN)
11	Yes	Yes	True Positive (TP)
12	Yes	No	False Negative (FN)

Các thông số áp dụng để đánh giá được tính như sau:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} = 0.5$$

$$Recall = \frac{TP}{TP + FN} = 0.5$$

$$Precision = \frac{TP}{TP + FP} = 1$$

$$F - Measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \approx 0.667$$

Recall + Precision

Nhìn chung Decision Tree tìm được có độ chính xác không cao khi chạy thử với Test Data. Nguyên nhân chính có lẽ là do tập Training Data quá ít.