Tên sinh viên: Nguyễn Vũ Khánh Huy

Decision tree (cây quyết định)

1. Giới thiệu sơ lược

Cây quyết định thuộc dạng **supervised learning** (máy học có giám sát) có thể được áp dụng vào cả bài toán classification và hồi quy. Việc xây dựng một cây quyết định trên dữ liệu huấn luyện cho trước là việc đi xác định các câu hỏi và thứ tự của chúng. Một điểm đáng lưu ý của decision tree là nó có thể làm việc với các đặc trưng (trong các tài liệu về decision tree, các đặc trưng thường được gọi là thuộc tính - attribute) dạng categorial, thường là rời rạc và không có thứ tự. Ví dụ, mưa nắng hay xanh đỏ, ... Cây quyết định cũng làm việc với dữ liệu có vector đặc trưng bao gồm cả thuộc tính dạng categorial và liên tục (numeric). Một điểm đáng chú ý nữa là cây quyết định ít yêu cầu việc chuẩn hoá dữ liệu.

Cây quyết định được sử dụng cho các bài toán **phân loại dữ liệu**. Dễ xây dựng cây và dễ hiểu là thế mạnh của cách tiếp cận bài toán bằng cây quyết định. Một cây quyết định bao gồm **nodes** (điểm trên cây), **branches** (nhánh) và l**eaf nodes** (node lá). Mỗi node là một đại diện cho một phép thử **logic** hay **toán học** trên từng thuộc tính trong tập dữ liệu. Mục tiêu cần đạt được là phân tách tập dữ liệu một cách rõ ràng để chỉ ra được sự liên quan giữa các biến số. Kết quả của từng phép thử chính là hướng đi của từng node. Node cha có thể có hai hoặc nhiều node con, tuỳ thuộc vào thuật toán đã chọn. Node cha và các node con được liên kết với nhau thông qua các nhánh, mỗi nhánh là đại diện cho kết quả của mỗi phép thử ở node cha. Node lá thì không có node con và chính là đại diện cho một class.

Hình 1.1 Ví dụ xây dựng cây quyết định

1. Mục tiêu.

Bài luận văn này tập trung vào việc tìm hiểu các bài toán trong thực tế và áp dụng cây quyết định để phân loại các tập dữ liệu sau đó trả về kết quả phù hợp nhất. Đặc điểm cấu tạo của cây quyết định giúp truyền tải ý tưởng từ bài toán vào thuật toán một cách tự nhiên nhất, không những vậy cây quyết định thường xuyên được sử dụng vào các bài toán phân loại trong thực tế như kinh tế, tài chính, y tế, nông nghiệp, sinh học.

3. **Ưu điểm** và **nhược điểm** cây quyết định.

Dự đoán từ thuật toán đưa ra đem lại kết quả tốt kể cả trên các tập dữ liệu chứa các đặc trưng độc lập, đây chính là ưu điểm lớn khi mà số lượng lớn các bài toán trong thực tế chỉ có các tập dữ liệu là các đặc trưng độc lập.

Xử lý được trên tập dữ liệu categorical và numerical.

Tốn ít công sức cho hoạt động chuẩn hoá dữ liệu, tuy vậy cây quyết định không sử dụng được trên các điểm dữ liệu trống.

Khả năng giải các bài toán cho ra nhiều output.

Kiểm tra tính đúng đắn của mô hình dưa trên thống kê.

Tuy vậy, cây quyết định cũng có những hạn chế nhất định như sau:

Nếu tập dữ liệu có nhiều biến liên hệ với nhau thì cây quyết định không hoạt động được, cụ thể hơn là nếu training trên các bộ dữ liệu phức tạp, nhiều biến và thuộc tính khác nhau có thể dẫn đến mô hình bị overfit, quá khớp với dữ liệu training dẫn đến hậu quả là mô hình khi đem để thử trên mẫu dữ liệu mới sẽ không cho kết quả chính xác.

Khi tập dữ liệu được phân chia ra thành các đặc trưng và sự chênh lệch giữa các đặc trưng là nhiều thì mô hình từ thuật toán cây quyết định bị **bias,** phân nhánh đơn giản chỉ chú ý đến các giá trị tiêu biểu và không kiểm soát hết các khả năng phân loại dữ liệu.

Tối ưu việc học trên cây quyết định được liệt vào dạng các bài toán NP-complete, thuật toán tham lam và heuristic thường xuyên được sử dụng

trong việc giải quyết các bài toán khi việc tối ưu chỉ diễn ra tại mỗi node mà từ đó trả về các kết quả không tối ưu. Nhược điểm này có thể được cải thiện trên kỹ thuật random forest.

4. Thuật toán.

4.1 ID3

4.1.1 Định nghĩa về Entropy.

Thuật ngữ **entropy** được các nhà khoa học mượn từ lĩnh vực vật lý trong quá trình xây dựng các phương pháp phân loại trong khoa học máy tính. **Entropy** được dùng để đo độ vẩn đục của tập dữ liệu.

Cho một phân phối xác suất của một biến rời rạc x có thể nhận n giá trị khác nhau

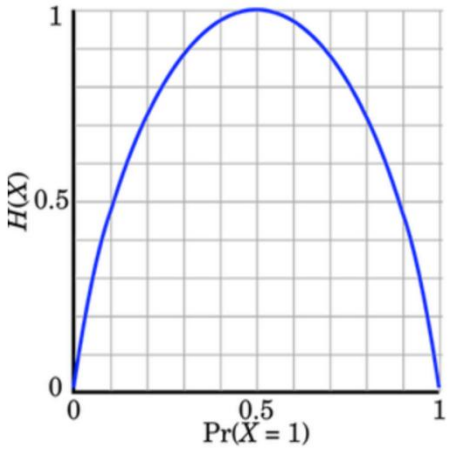
$x_1, x_2, ..., x_n$. . Giả sử rằng xác suất để x nhận các giá trị này là $p_i = p\ (x = x_i)$.

Ký hiệu phân phối này là $\boldsymbol{p} = (p_1, p_2, ..., p_n)$.

Entropy của phân phối này là:

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

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



Từ đồ thị ta thấy hàm Entropy với n > 2 thì

√ 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.

Tổng kết cho định nghĩa về hàm số Entropy.

Entropy biểu thị cho sự hỗn độn, độ bất định, độ phức tạp của thông tin.

Thông tin càng phức tạp thì entropy càng cao.

Entropy nhạy cảm với việc thay đổi sác xuất nhỏ, khi hai phân bố càng giống nhau thì entropy càng giống nhau và ngược lại.

Mục tiêu khi xây dựng cây quyết định là cho ta nhiều thông tin nhất tức là chọn entropy cao nhất.

Những tính chất này của hàm entropy khiến nó được sử dụng trong việc đo độ vẩn đục của một phép phân chia của ID3. Vì lý do này, ID3 còn được gọi là **entropy-based decision tree**.

4.1.2 Thuật toán ID3 (Iterative Dichotomiser 3)

Thuật toán ID3 lần đầu được công bố bởi Ross Quinlan vào năm 1986, thuật toán hoạt động dựa trên hàm số entropy.

Trong ID3, *tổng có trọng số của entropy tại các leaf-node* sau khi xây dựng decision tree được coi là hàm mất mát của decision tree đó .Các trọng số ở đây tỉ lệ với số điểm dữ liệu được phân vào mỗi node. Công việc của ID3 là tìm các cách phân chia hợp lý (thứ tự chọn thuộc tính hợp lý) sao cho hàm mất mát cuối cùng đạt giá trị càng nhỏ càng tốt. Như đã đề cập, việc này đạt được bằng cách chọn ra thuộc tính sao cho nếu dùng thuộc tính đó để phân chia, entropy tại mỗi bước giảm đi một lượng lớn nhất. Bài toán xây dựng một decision tree bằng ID3 có thể chia thành các bài toán nhỏ, trong mỗi bài toán, ta chỉ cần chọn ra thuộc tính giúp cho việc phân chia đạt kết quả tốt nhất. Mỗi bài toán nhỏ này tương ứng với việc phân chia dữ liệu trong một *non-leaf node*. Chúng ta sẽ xây dựng phương pháp tính toán dựa trên mỗi node này.

Xét một bài toán với **C** class khác nhau. Giả sử ta đang làm việc với một *non-leaf node* với các điểm dữ liệu tạo thành một tập S với số phần tử là $|S| = N$.
Giả sử thêm rằng trong số N điểm dữ liệu này N_c, c = 1 , 2 , ..., C điểm thuộc vào class **c**.
Xác suất để mỗi điểm dữ liệu rơi vào một class c được xấp xỉ bằng $\frac{N_c}{N}$ (maximum likelihood estimation).

Như vậy, entropy tại node này được tính bởi:

$$\mathrm{H(S)} = -\sum_{c=1}^{C} \frac{\mathrm{N_c}}{\mathrm{N}} \log \frac{\mathrm{N_c}}{\mathrm{N}} \qquad (\mathbf{2})$$

Tiếp theo, giả sử thuộc tính được chọn là x. Dựa trên x, các điểm dữ liệu trong S được phân ra thành K child node S_1, S_2, ..., S_k với số điểm trong mỗi child node lần lượt là

$m_1,\ m_2, ...,\ m_k$. Ta định nghĩa.

$$\mathrm{H}(x, \mathrm{S}) \ = \ \sum_{\mathrm{k}=1}^{\mathrm{K}} \frac{\mathrm{m_k}}{\mathrm{N}} \mathrm{H}(S_k) \qquad (\mathbf{3})$$

là tổng có trọng số entroy của mỗi child node–được tính tương tự như (2). Việc lấy trọng số này là quan trọng vì các node thường có số lượng điểm khác nhau.

Tiếp theo, ta định nghĩa i**nformation gain** dựa trên thuộc tính x:

$$x^* \ = \ \underset{x}{arg\ max}\ G(x,\ S) = \ \underset{x}{arg\ min}\ H(x,\ S)$$

Tức thuộc tính khiến cho information gain đạt giá trị lớn nhất.
4.1.3 Ví dụ thuật toán ID3.

Dưới đây là tập dữ liệu mô tả quan hệ thời tiết trong 14 ngày gồm bốn thuộc tính **outlook**, **temperature**, **humidity**, **wind.** Cột **play** chính là target mà ta phải dự đoán nếu đã biết giá trị của bốn cột còn lại.

id	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rainy	mild	high	weak	yes
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rainy	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rainy	mild	high	strong	no

Dữ liệu về thời tiết ảnh hưởng đến quyết định chơi bóng.

Sơ lược về tập dữ liệu:

Tập dữ liệu gồm bốn thuộc tính thời tiết:
1. **Outlook** có ba giá trị sunny, overcast, rainy.
2. **Temperature** nhận một trong ba giá trị hot, cool, mild.
3. **Humidity** nhận một trong hai giá trị high, normal.
4. **Wind** nhận một trong hai giá trị weak và strong.

Target chính là cột play, đây chính là cột phải đưa quyết định dựa trên các thuộc tính trên gồm có hai giá trị là yes và no.

Tập dữ liệu trên có **14** giá trị kết quả trong đó có **9** giá trị **yes** và **5** giá trị **no**. Entropy tại root node của bài toán tính theo **(1)** là:

$$\mathbf{H}(\mathbf{S}) = -\left(\frac{9}{14}\log\frac{9}{14} + \frac{5}{14}\log\frac{5}{14}\right) \approx 0.65$$

Tính tổng có trọng số entropy của các child node nếu chọn một trong các thuộc tính outlook, temperature, humidity, wind, play để phân chia dữ liệu.
***Lưu ý các phép tính trong machine learning, ngôn ngữ lập trình khi nói đến log là chỉ đến ln.**

- Xét thuộc tính **outlook**. Thuộc tính này có thể nhận một trong ba giá trị sunny, overcast, rainy. Mỗi một giá trị sẽ tương ứng một child node. Gọi **tập hợp các điểm** trong **mỗi child node** này lần lượt là S_s, S_o, S_r với tương ứng m_s, m_o, m_r .

Sắp xếp bảng theo thuộc tính **outlook** ta được.

id	outlook	temperature	humidity	wind	play
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

id	outlook	temperature	humidity	wind	play
3	overcast	hot	high	weak	yes
7	overcast	cool	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes

id	outlook	temperature	humidity	wind	play
4	rainy	mild	high	weak	yes
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
10	rainy	mild	normal	weak	yes
14	rainy	mild	high	strong	no

Theo công thức tính Entropy **(2)** ta có được các giá trị:

Tính **Entropy** cho child node **sunny** của thuộc tính **outlook** với **2** giá trị **yes, 3** giá trị **no** và m_s **= 5.**

$$\mathbf{H}(S_s) = -\left(\frac{2}{5}\log\frac{2}{5} + \frac{3}{5}\log\frac{3}{5}\right) \approx 0.673$$

Tính **Entropy** cho child node **overcast** của thuộc tính **outlook** với

4 giá trị **yes, 0** giá trị **no** và m_o **= 4.**

$$\mathbf{H}(S_o) = -\left(\frac{4}{4}\log\frac{4}{4} + \frac{0}{4}\log\frac{0}{4}\right) \approx 0$$

Tính **Entropy** cho child node **rainy** của thuộc tính **outlook** với **3** giá trị **yes, 2** giá trị **no** và m_r **= 5.**

$$\mathbf{H}(S_r) = -\left(\frac{3}{5}\log\frac{3}{5} + \frac{2}{5}\log\frac{2}{5}\right) \approx 0.673$$

Tính tổng trọng số Entropy của mỗi child node theo công thức **(3)** được.

$$\mathbf{H}(\mathbf{outlook}, \boldsymbol{S}) = \frac{5}{14}\mathrm{H}(S_s) + \frac{4}{14}\mathrm{H}(S_o) + \frac{5}{14}\mathrm{H}(S_r)$$

$$= \frac{5}{14}0.673 + \frac{4}{14}0 + \frac{5}{14}0.673 \approx 0.48$$

- Xét thuộc tính **temperature**. Thuộc tính này có thể nhận một trong ba giá trị **hot**, **mild**, **cool** . Mỗi một giá trị sẽ tương ứng một child node. Gọi **tập hợp các điểm** trong **mỗi child node** này lần lượt là S_h, S_m, S_c với tương ứng m_h, m_m, m_c .

Sắp xếp dữ liệu theo thuộc tính **temperature** ta được

id	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
13	overcast	hot	normal	weak	yes

id	outlook	temperature	humidity	wind	play
4	rainy	mild	high	weak	yes
8	sunny	mild	high	weak	no
10	rainy	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
14	rainy	mild	high	strong	no

id	outlook	temperature	humidity	wind	play
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
9	sunny	cool	normal	weak	yes

Theo công thức tính Entropy **(2)** ta có được các giá trị:

Tính **Entropy** cho child node **hot** của thuộc tính **temperature** với **2** giá trị **yes, 2** giá trị **no** và $\boldsymbol{m_h}$ **= 4.**

$$\mathbf{H}(S_h) = -\left(\frac{2}{4}\log\frac{2}{4} + \frac{2}{4}\log\frac{2}{4}\right) \approx 0.693$$

Tính **Entropy** cho child node **mild** của thuộc tính **temperature** với **4** giá trị **yes, 2** giá trị **no** và $\boldsymbol{m_m}$ **= 6.**

$$\mathbf{H}(S_m) = -\left(\frac{4}{6}\log\frac{4}{6} + \frac{2}{6}\log\frac{2}{6}\right) \approx 0.637$$

Tính **Entropy** cho child node **cool** của thuộc tính **temperature** với **3** giá trị **yes, 1**giá trị **no** và $\boldsymbol{m_c}$ **= 4.**

$$\mathbf{H}(S_c) = -\left(\frac{3}{4}\log\frac{3}{4} + \frac{1}{4}\log\frac{1}{4}\right) \approx 0.562$$

Tính tổng trọng số Entropy của mỗi child node theo công thức **(3)** được.

$$\mathbf{H}(\text{ temperature }, \boldsymbol{S}) = \frac{4}{14}\mathrm{H}(S_h) + \frac{6}{14}\mathrm{H}(S_m) + \frac{4}{14}\mathrm{H}(S_c)$$

$$= \frac{4}{14}0.693 + \frac{6}{14}0.637 + \frac{4}{14}0.562 \approx 0.631$$

- Xét thuộc tính **humidity**. Thuộc tính này có thể nhận một trong ba giá trị **high**, **normal** . Mỗi một giá trị sẽ tương ứng một child node. Gọi **tập hợp các điểm** trong **mỗi child node** này lần lượt là $\boldsymbol{S_h}$, $\boldsymbol{S_n}$ với tương ứng $\boldsymbol{m_h}$, $\boldsymbol{m_n}$.

Sắp xếp dữ liệu theo thuộc tính **humidity** ta được

id	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rainy	mild	high	weak	yes
8	sunny	mild	high	weak	no
12	overcast	mild	high	strong	yes
14	rainy	mild	high	strong	no

id	outlook	temperature	humidity	wind	play
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
9	sunny	cool	normal	weak	yes
10	rainy	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
13	overcast	hot	normal	weak	yes

Theo công thức tính Entropy **(2)** ta có được các giá trị:

Tính **Entropy** cho child node **high** của thuộc tính **humidity** với **3** giá trị **yes, 4** giá trị **no** và $\boldsymbol{m_h}$ **= 7.**

$$\mathbf{H}(S_h) = -\left(\frac{3}{7}\log\frac{3}{7} + \frac{4}{7}\log\frac{4}{7}\right) \approx 0.683$$

Tính **Entropy** cho child node **normal** của thuộc tính **humidity** với **6** giá trị **yes, 1** giá trị **no** và $\boldsymbol{m_n}$ **= 7.**

$$\mathbf{H}(S_n) = -\left(\frac{6}{7}\log\frac{6}{7} + \frac{1}{7}\log\frac{1}{7}\right) \approx 0.410$$

Tính tổng trọng số Entropy của mỗi child node theo công thức **(3)** được.

$$\mathbf{H}(\mathbf{humidity}, \boldsymbol{S}) = \frac{7}{14}\mathrm{H}(S_h) + \frac{7}{14}\mathrm{H}(S_n)$$
$$= \frac{7}{14}0.683 + \frac{7}{14}0.410 \approx 0.547$$

- Xét thuộc tính **wind**. Thuộc tính này có thể nhận một trong ba giá trị **strong**, **weak**. Mỗi một giá trị sẽ tương ứng một child node. Gọi **tập hợp các điểm** trong **mỗi child node** này lần lượt là $\boldsymbol{S_s}$, $\boldsymbol{S_w}$ với tương ứng $\boldsymbol{m_s}$, $\boldsymbol{m_w}$.

Sắp xếp tập dữ liệu theo thuộc tính wind ta được

id	outlook	temperature	humidity	wind	play
2	sunny	hot	high	strong	no
6	rainy	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
14	rainy	mild	high	strong	no

id	outlook	temperature	humidity	wind	play
1	sunny	hot	high	weak	no
3	overcast	hot	high	weak	yes
4	rainy	mild	high	weak	yes
5	rainy	cool	normal	weak	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rainy	mild	normal	weak	yes
13	overcast	hot	normal	weak	yes

Theo công thức tính Entropy **(2)** ta có được các giá trị:

Tính **Entropy** cho child node **strong** của thuộc tính **wind** với **3** giá trị **yes, 3** giá trị **no** và $\boldsymbol{m_s}$ **= 6.**

$$\mathbf{H}(S_s) = -\left(\frac{3}{6}\log\frac{3}{6} + \frac{3}{6}\log\frac{3}{6}\right) \approx 0.693$$

Tính **Entropy** cho child node **weak** của thuộc tính **wind** với **6** giá trị **yes, 2** giá trị **no** và $\boldsymbol{m_w}$ **= 8.**

$$\mathbf{H}(S_w) = -\left(\frac{6}{8}\log\frac{6}{8} + \frac{2}{8}\log\frac{2}{8}\right) \approx 0.562$$

Tính tổng trọng số Entropy của mỗi child node theo công thức **(3)** được.

$$\mathbf{H}(\text{ wind }, \boldsymbol{S}) = \frac{6}{14}\,\mathrm{H}(S_s) + \frac{8}{14}\,\mathrm{H}(S_w)$$

$$= \frac{6}{14}\,0.693 + \frac{8}{14}\,0.562 \approx 0.618$$

Tổng trọng số Entropy của bốn thuộc tính sau các phép tính trên là:

√ $\mathbf{H}(\text{ outlook}, \boldsymbol{S}) \approx 0.48$

√ $\mathbf{H}(\text{ temperature}, \boldsymbol{S}) \approx 0.631$

√ $\mathbf{H}(\text{ humidity}, \boldsymbol{S}) \approx 0.547$

√ $\mathbf{H}(\text{ wind}, \boldsymbol{S}) \approx 0.618$

Information gain lớn nhất là khi chọn thuộc tính có trọng số Entropy nhỏ nhất. Vậy ở bước đầu tiên ta chọn thuộc tính **outlook** vì $\mathbf{H}(\text{ outlook}, \boldsymbol{S})$ đạt giá trị nhỏ nhất.

Những thuộc tính của outlook là overcast, sunny, rainy.

√ Nếu thuộc tính là overcast thì sẽ dừng phân nhánh và cho kết quả là yes, vì entropy của thuộc tính overcast là bằng 0.

Nếu thuộc tính là sunny sẽ có entropy là:

$$\mathbf{H}(\mathbf{Sunny}) = -\left(\frac{2}{5}\log\frac{2}{5} + \frac{3}{5}\log\frac{3}{5}\right) \approx 0.673$$

Tập dữ liệu của thuộc tính **outlook** là **sunny** sắp xếp theo **temperature**

id	outlook	temperature	humidity	wind	play
9	sunny	cool	normal	weak	yes
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
8	sunny	mild	high	weak	no
11	sunny	mild	normal	strong	yes

➢ Tính entropy của thuộc tính **temperature** trong tập **outlook** là **sunny**:

$\mathbf{H(temperature, sunny)}$

$$= \frac{1}{5}\mathbf{H(cool, temperature)} + \frac{2}{5}\mathbf{H(hot, temperature)}$$

$$+ \frac{2}{5}\mathbf{H(mild, temperature)}$$

$$\mathbf{H(cool, temperature)} = -\left(\frac{1}{1}\log\frac{1}{1}\right) = 0$$

$$\mathbf{H(hot, temperature)} = -\left(\frac{0}{2}\log\frac{0}{2} + \frac{2}{2}\log\frac{2}{2}\right) = 0$$

$$\mathbf{H(mild, temperature)} = -\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right) \approx 0.693$$

√ Vậy $\mathbf{H(temperature, sunny)} \approx \mathbf{0.2772}$

Tập dữ liệu của thuộc tính **outlook** là **sunny** sắp xếp theo **humidity.**

id	outlook	temperature	humidity	wind	play
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

➢ Tính entropy của thuộc tính **humidity** trong tập **outlook** là **sunny**:

$$\mathbf{H(humidity, sunny)} = \frac{3}{5}\mathbf{H(high, humidity)} + \frac{2}{5}\mathbf{H(normal, humidity)}$$

$$\mathbf{H(high, humidity)} = -\left(\frac{0}{3}\log\frac{0}{3} + \frac{3}{3}\log\frac{3}{3}\right) = 0$$

$$\mathbf{H(normal, humidity)} = -\left(\frac{2}{2}\log\frac{2}{2} + \frac{0}{2}\log\frac{0}{2}\right) = 0$$

√ Vậy $\mathbf{H(humidity, sunny)} = \mathbf{0}$

Tập dữ liệu của thuộc tính **outlook** là **sunny** sắp xếp theo **wind.**

id	outlook	temperature	humidity	wind	play
2	sunny	hot	high	strong	no
11	sunny	mild	normal	strong	yes
1	sunny	hot	high	weak	no
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes

➢ Tính entropy của thuộc tính **wind** trong tập **outlook** là **sunny**:

$$\mathbf{H(wind, sunny)} = \frac{2}{5}\mathbf{H(strong, wind)} + \frac{3}{5}\mathbf{H(weak, wind)}$$

$$\mathbf{H(strong, wind)} = -\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right) \approx 0.693$$

$$\mathbf{H(weak, wind)} = -\left(\frac{1}{3}\log\frac{1}{3} + \frac{2}{3}\log\frac{2}{3}\right) \approx 0.637$$

√ Vậy $\mathbf{H(wind, sunny)} \approx \mathbf{0.66}$

√ Chọn thuộc tính **humidity** để tiếp tục phân nhánh vì tổng trọng số entropy bằng 0 với output là yes khi và chỉ khi humidity là **normal**.

Nếu thuộc tính là rainy sẽ có entropy là:

$$\mathbf{H(rainy)} = -\left(\frac{2}{5}\log\frac{2}{5} + \frac{3}{5}\log\frac{3}{5}\right) \approx 0.673$$

Tập dữ liệu của thuộc tính **outlook** là **rainy** sắp xếp theo **temperature**

id	outlook	temperature	humidity	wind	play
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
4	rainy	mild	high	weak	yes
10	rainy	mild	normal	weak	yes
14	rainy	mild	high	strong	no

➢ Tính entropy của thuộc tính **temperature** trong tập **outlook** là **rainy**:

$$\mathbf{H(temperature, rainy)}$$

$$= \frac{2}{5}\mathbf{H(cool, temperature)} + \frac{0}{5}\mathbf{H(hot, temperature)}$$

$$+ \frac{3}{5}\mathbf{H(mild, temperature)}$$

$$\mathbf{H(cool, temperature)} = -\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right) \approx 0.693$$

$$\mathbf{H(hot, temperature)} = 0$$

$$\mathbf{H(mild, temperature)} = -\left(\frac{2}{3}\log\frac{2}{3} + \frac{1}{3}\log\frac{1}{3}\right) \approx 0.636$$

√ Vậy $\mathbf{H(temperature, rainy)} \approx \mathbf{0.6588}$

Tập dữ liệu của thuộc tính **outlook** là **rainy** sắp xếp theo **humidity.**

id	outlook	temperature	humidity	wind	play
4	rainy	mild	high	weak	yes
14	rainy	mild	high	strong	no
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
10	rainy	mild	normal	weak	yes

➢ Tính entropy của thuộc tính **humidity** trong tập **outlook** là **rainy**:

$$\mathbf{H}(\text{humidity}, \mathbf{rainy}) = \frac{2}{5}\mathbf{H}(\text{high}, \mathbf{humidity}) + \frac{3}{5}\mathbf{H}(\text{normal}, \mathbf{humidity})$$

$$\mathbf{H}(\text{high}, \mathbf{humidity}) = -\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right) \approx 0.693$$

$$\mathbf{H}(\text{normal}, \mathbf{humidity}) = -\left(\frac{2}{3}\log\frac{2}{3} + \frac{1}{3}\log\frac{1}{3}\right) \approx 0.636$$

√ Vậy $\mathrm{H}(\text{humidity}, \mathbf{rainy}) \approx 0.659$

Tập dữ liệu của thuộc tính **outlook** là **rainy** sắp xếp theo **wind.**

id	outlook	temperature	humidity	wind	play
14	rainy	mild	high	strong	no
6	rainy	cool	normal	strong	no
4	rainy	mild	high	weak	yes
5	rainy	cool	normal	weak	yes
10	rainy	mild	normal	weak	yes

➢ Tính entropy của thuộc tính **wind** trong tập **outlook** là **sunny**:

$$\mathbf{H}(\mathbf{wind}, \mathbf{sunny}) = \frac{2}{5}\mathbf{H}(\mathbf{strong}, \mathbf{wind}) + \frac{3}{5}\mathbf{H}(\mathbf{weak}, \mathbf{wind})$$

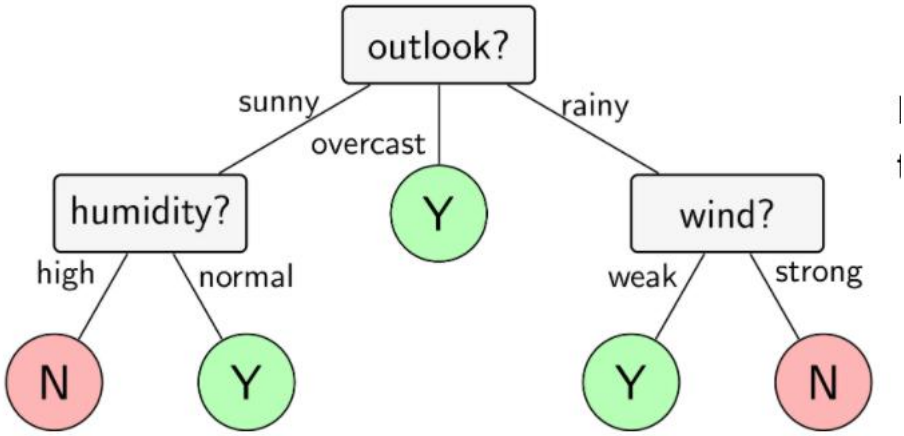
$$\mathbf{H}(\mathbf{strong}, \mathbf{wind}) = -\left(\frac{0}{2}\log\frac{0}{2} + \frac{2}{2}\log\frac{2}{2}\right) = 0$$

$$\mathbf{H}(\mathbf{weak}, \mathbf{wind}) = -\left(\frac{3}{3}\log\frac{3}{3} + \frac{0}{3}\log\frac{0}{3}\right) = 0$$

√ Vậy $\mathbf{H}(\mathbf{wind}, \mathbf{rainy}) = \mathbf{0}$

√ Chọn thuộc tính **wind** để tiếp tục phân nhánh vì tổng trọng số entropy bằng 0 với output là **yes** khi và chỉ khi rainy là **weak**.

Từ đó ta có cây quyết định được xây dựng như hình bên dưới



4.2 Random forest.

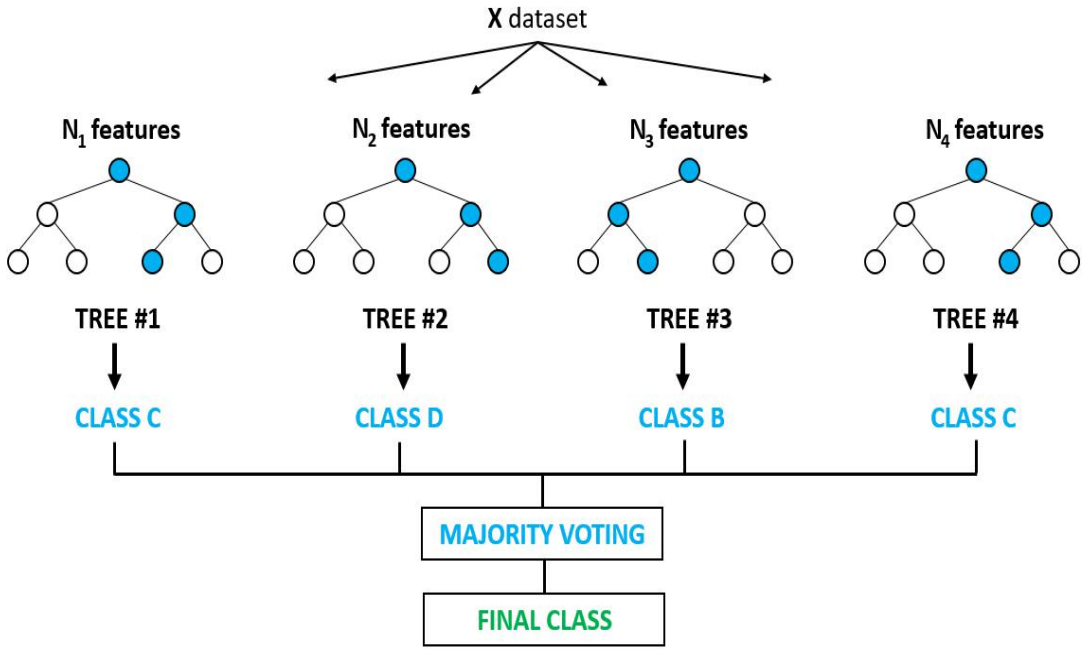
Random Forest là một thành viên trong họ thuật toán **decision tree** (cây quyết định).Ý tưởng phía sau *Random Forest* khá đơn giản.
Thuật toán này sinh một số cây quyết định (thường là vài trăm) và sử dụng chúng. Các câu hỏi của cây quyết định sẽ là câu hỏi về các thuộc tính.
Ví dụ: "Cánh hoa có dài hơn 1.7cm hay không?". Câu giá trị ở nút lá sẽ là các lớp (*class*). Sử dụng hàng trăm cây quyết định là bất khả thi với con người, nhưng máy tính có thể làm việc này tương đối dễ dàng.

Có hai giải pháp. Cách thứ nhất là hỏi chuyên gia (ví dụ như một nhà nhân chủng học với bài toán phân biệt chủng loại của con vật). Nhưng không phải khi nào cũng có thể tiếp cận được với chuyên gia trong bài toán của mình. Hơn nữa, ngay cả những chuyên gia giỏi nhất cũng gặp khó khăn trong việc viết ra những kiến thức của mình và ngay cả khi tìm được một chuyên gia có khả năng đó thì chắc chắn sẽ có những thứ mà họ không biết tới. Ví dụ, nhà nhân chủng học của chúng ta có thể quên mất rằng con đà điểu có thể nhỏ hơn 50kg.

Thay vì sử dụng chuyên gia, các nhà nghiên cứu sử dụng phương án thứ hai: tạo ra một thuật toán tự sinh cây quyết định. Điều kiện duy nhất là phải có vài ví dụ để máy tính có thể tham chiếu. Trong Iris dataset, những ví dụ này chính là những bông hoa mà chúng ta đã biết chủng loại.

Để tạo ra một cây quyết định, thuật toán *Random Forest* luôn bắt đầu bằng một cây rỗng. Một cây quyết định rỗng chỉ có một ô *Start* chỉ thẳng đến câu trả lời (ô xanh lá). Tiếp theo, thuật toán sẽ tìm câu hỏi đầu tiên và bắt đầu xây dựng cây quyết định. Mỗi lần thuật toán tìm được thêm một câu hỏi, nó tạo hai nhánh trên cây quyết định. Khi không còn câu hỏi nào nữa, thuật toán dừng lại và chúng ta có một cây quyết định hoàn chỉnh.

Làm thế nào để tìm ra những câu hỏi tốt nhất cho cây quyết định? Đây là một bước khá phức tạp nhưng ý tưởng đằng sau nó tương đối đơn giản: Ở thời điểm bắt đầu, thuật toán của chúng ta chưa biết phân biệt các chủng loại của các con vật. Nói cách khác, tất cả các con vật được cho chung vào một "cái túi". Để tìm ra câu hỏi tốt nhất, thuật toán thử đưa ra tất cả các câu hỏi có thể (có khi là hàng triệu câu hỏi). Ví dụ: "Nó có bao nhiêu chân?", "Nó có đuôi không?",... Sau đó, với mỗi câu hỏi, thuật toán sẽ đánh giá mức độ hiệu quả mà câu hỏi này giúp phân biệt các chủng loại, hay các *class*. Câu hỏi được chọn không cần thiết phải hoàn hảo, nhưng nó phải tốt hơn những câu hỏi khác. Để tính toán mức độ hiệu quả của câu hỏi, chúng ta sử dụng một độ đo có tên là **information gain**.
Có thể hiểu **information gain** như một cách để "cho điểm" các câu hỏi. Câu hỏi với *information gain* lớn nhất sẽ được chọn như là câu hỏi tốt nhất để xây dựng cây quyết định. Sau khi thuật toán xây dựng xong các cây quyết định, những cây này sẽ được sử dụng để trả lời câu hỏi (hay phân loại).



Random Forest coi mỗi cây quyết định như một cử tri bỏ phiếu độc lập (như một cuộc bầu cử thực sự). Ở cuối cuộc bầu cử, câu trả lời nhận được nhiều bầu chọn nhất từ các cây quyết định sẽ được lựa chọn.

Tuy nhiên, vẫn còn một vấn đề: Nếu như tất cả các cây được dựng theo cùng một cách, chúng sẽ cho những câu trả lời giống nhau. Như vậy chẳng khác gì chúng ta chỉ sử dụng một cây quyết định duy nhất cả. Ở đây, Random Forest có một cách làm rất hay: Để chắc chắn rằng không phải tất cả các cây quyết định cho cùng câu trả lời, thuật toán Random Forest chọn ngẫu nhiên các quan sát (observations). Chính xác hơn, Random Forest sẽ xoá một số quan sát và lặp lại một số khác một cách ngẫu nhiên. Xét toàn cục, những quan sát này vẫn rất gần với tập các quan sát ban đầu, nhưng những thay đổi nhỏ sẽ đảm bảo rằng mỗi cây quyết định sẽ có một chút khác biệt. Quá trình này gọi là **bootstrapping**

Thêm vào đó, để thực sự chắc chắn các cây quyết định là khác nhau, thuật toán Random Forest sẽ ngẫu nhiên bỏ qua một số câu hỏi khi xây dựng cây quyết định. Trong trường hợp này, nếu câu hỏi tốt nhất không được chọn, một câu hỏi kế tiếp sẽ được lựa chọn để dựng cây. Quá trình này được gọi là **attribute sampling**

Tạo ra một thuật toán phức tạp như vậy, ngẫu nhiên thay đổi các quan sát và bỏ qua một số câu hỏi. Câu trả lời rất đơn giản: Có thể các mẫu thử mà chúng ta đang sử dụng chưa hoàn hảo. Ví dụ, có thể mẫu thử của chúng ta chỉ có những con mèo có lông đuôi. Trong trường hợp này những con mèo thuộc loài sphynx (mèo không lông) có thể được phân loại là con chuột. Tuy nhiên, nếu câu hỏi về đuôi không được hỏi (bởi vì sự thay đổi ngẫu nhiên), thuật toán có thể sử dụng câu hỏi các câu hỏi khác (ví dụ: Con vật đó có kích thước như thế nào?). Việc có nhiều câu hỏi đa dạng (có thể không hoàn hảo) là một ý tưởng không tồi: nó có thể là cứu tinh khi thuật toán tham chiếu đến một quan sát mà nó chưa từng thấy trước đây.