Decision Tree: ID3

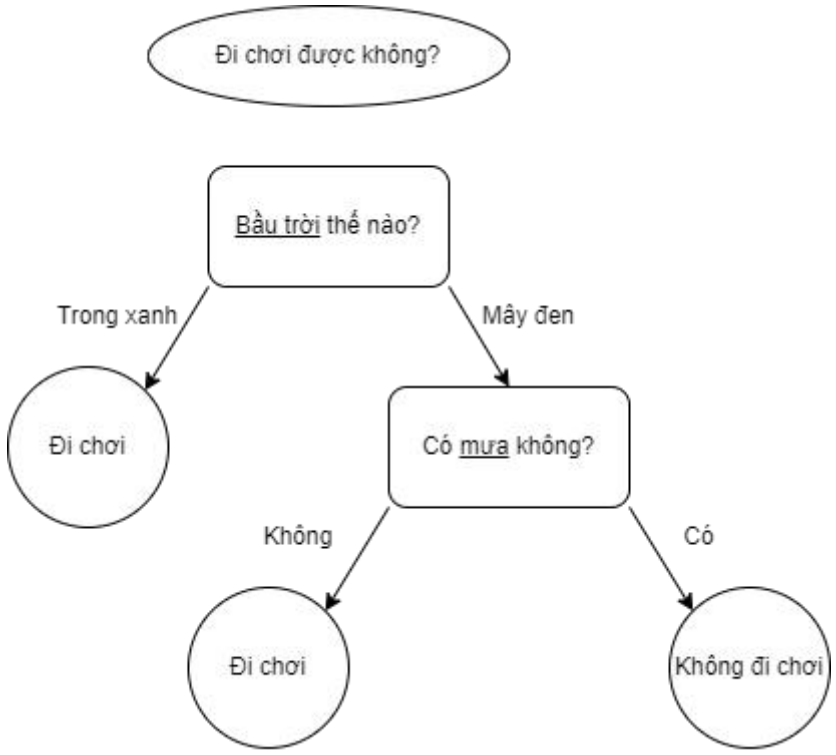
Thuật toán ID3 (Iterative Dichotomiser 3) là một kỹ thuật xây dựng cây quyết định dựa trên cơ chế chọn lọc các đặc trưng để tối ưu hóa *thông tin nhận vào* (Information Gain). Thuật toán này thường được dùng trong phân lớp (classification).

Ý tưởng của thuật toán là mình cần tìm ra đặc trưng tốt nhất làm nút gốc dựa trên một tiêu chuẩn nhất định. Với một đặc trưng được chọn, mình sẽ chia dữ liệu vào các nút con tương ứng với các giá trị của đặc trưng ấy, mà rồi tiếp tục áp dụng phương pháp này cho mỗi nút con. Việc chọn đặc trưng tốt nhất ở mỗi bước được gọi là một cách chọn tham lam. Sau tất cả thì mình muốn tìm đặc trưng có lượng *thông tin nhận vào* cao nhất làm nút gốc.

Trong cây quyết định, mỗi đặc trưng trong tập dữ liệu là một *câu hỏi*, và các dữ liệu được phân chia vào các nút con tượng trưng cho *các câu trả lời* cho *câu hỏi* ấy. Giả sử mình có 1 tập dữ liệu với đặc trưng "Bầu trời" quyết định sự kiện "Đi chơi" có diễn ra hay không dựa trên 2 lớp từ đặc trưng ấy.

Đặc trưng "Bầu trời" và 2 lớp của nó ("Trong xanh" và "Mây đen") dưới dạng cây quyết định đưa ra khả năng đi chơi.

Trong cây quyết định, một phép phân chia tốt nhất là khi dữ liệu trong mỗi nút con hoàn toàn phụ thuộc vào một lớp. Cùng lúc đó, nút con này có thể được gọi là một nút lá, tức là một nút không còn khả năng phân chia dữ liệu. Từ ví dụ ở sơ đồ trên, mình sẽ thêm vào một đặc trưng mới "Mưa" với 2 lớp của nó tương ứng với "Có mưa" và "Không mưa" để quyết định sự kiện "Đi chơi" rõ ràng hơn.



Đặc trưng "Bầu trời", và 1 đặc trưng mới, "Mưa", quyết định sự kiện đi chơi.

Theo sơ đồ trên dựa trên tập dữ liệu, trong đó:

- Đặc trưng "Bầu trời": nút gốc/nút cha (root node/parent node).
- Đặc trưng "Mưa": nút con (child node) của nút gốc.
- Các thuộc tính "Đi chơi" và "Không đi chơi" của biến mục tiêu "Đi chơi?": các nút lá (leaf node).
- Các mũi tên: Chỉ ra sự kiện sẽ xảy ra dựa trên các lớp của các đặc trưng.

Nếu vậy dữ liệu trong các nút con sẽ bị trộn lẫn với nhau theo một tỉ lệ lớn, mình coi rằng phép phân chia ấy thực sự chưa tốt.

Vì thế, mình cần có một hàm đo lường độ tinh khiết (purity), hoặc ngược lại, độ vẩn đục (impurity) của phép phân chia. Phép đo lường này cho giá trị thấp

nhất nếu dữ liệu trong mỗi nút con nằm trong cùng một lớp (tinh khiết nhất), và sẽ cho giá trị cao nếu mỗi nút con chứa dữ liệu của nhiều lớp khác nhau.

Vì dụ cơ bản về độ tinh khiết trong dữ liệu nằm ở hình thứ 2 trong tài liệu này:

- Sự kiện "Bầu trời trong xanh" hoàn toàn sẽ dẫn đến sự kiện "đi chơi".
- Sự kiện "Bầu trời mây đen" phải nhờ đến sự kiện "Có mưa" hay "Không mưa" thì mới có thể quyết định được khả năng đi chơi của sự kiện.

Nhưng để mà thực sự quyết định được đặc trưng nào trong tập dữ liệu làm nút gốc, nút con, mình sẽ sữ dụng một hàm số đặc biệt thường được dùng nhiều trong lý thuyết thông tin, hàm **Entropy**. hàm này đo lường độ vẩn đục trong một tập dữ liệu..Mình có công thức tính hàm Entropy như sau:

$$\mathrm{H(S)} \;=\; -\sum_{\mathrm{i}=1}^{\mathrm{n}} p_{\mathrm{i}} \log_2 (p_{\mathrm{i}})$$

Trong đó:

- H(S): Entropy của tập dữ liệu S.
- p_i : Tỉ lệ của lớp i trong 1 đặc trưng của tập dữ liệu. (với $0 \leq \mathrm{pi} \leq 1$)

Trong thuật toán ID3, tổng trọng số của Entropy tại các nút lá sau khi xây dựng thành công một cây quyết định được biết rằng là hàm mất mát của thuật toán. Các trọng số tỉ lệ với số mẫu dữ liệu được phân chia ra mỗi nút. Và nhiệm của ID3 là tìm ra cách phân chia tốt nhất để có được hàm mất mát cuối cùng nhỏ nhất.

Ví dụ một tập dữ liệu có $\boldsymbol{C}$ nhãn khác nhau từ biến mục tiêu, mình đang tính toán một nút lá với các mẫu dữ liệu tạo thành tập dữ liệu $\boldsymbol{S}$ với số phần tử $\boldsymbol{N}$. Giả sử rằng $\boldsymbol{N}$ mẫu dữ liệu, mình có $\boldsymbol{N_c}$ (với $\boldsymbol{c}$ = 1, 2, …. , $\boldsymbol{C}$) mẫu thuộc nhãn $\boldsymbol{c}$. Mình có xác suất để mỗi mẫu dữ liệu rơi vào nhãn $\boldsymbol{c}$ xấp xỉ bằng $\frac{\boldsymbol{N_c}}{\boldsymbol{N}}$ (maximum likelihood estimation). Từ đó, Entropy tại nút ấy được tính bằng:

$$\boldsymbol{H(S)} \;=\; -\sum_{\boldsymbol{i}=1}^{\boldsymbol{C}} \frac{\boldsymbol{N_c}}{\boldsymbol{N}} \log_2 \left(\frac{\boldsymbol{N_c}}{\boldsymbol{N}}\right)$$

Tiếp đó, hãy nghĩ rằng đặc trưng được chọn để tính là $\boldsymbol{x}$, các mẫu dữ liệu trong tập $\boldsymbol{S}$ được chia ra thành $\boldsymbol{K}$ các nút con/các lớp ($\boldsymbol{S_1}$, $\boldsymbol{S_2}$, … , $\boldsymbol{S_K}$) với số mẫu của mỗi nút con tương ứng là ($\boldsymbol{m_1}$, $\boldsymbol{m_2}$, … , $\boldsymbol{m_K}$). Mình có:

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

Công thức trên là tổng trọng số Entropy của mỗi nút con.

Sau đó, mình sẽ tính *thông tin nhận vào* dựa trên mỗi đặc trưng $\boldsymbol{x}$:

$$\boldsymbol{G(x, S)} = \boldsymbol{H(S)} - \boldsymbol{H(x, S)}$$

Trong thuật toán ID3, tại mỗi nút, đặc trưng được chọn thường dựa trên:

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

Nói cách khác, đặc trưng $\boldsymbol{x}$ được chọn thường có:

- *Thông tin nhận vào (information gain)* $\boldsymbol{G(x, S)}$ lớn nhất
- *Độ vẩn đục Entropy* $\boldsymbol{H(x, S)}$ nhỏ nhất

Đọc thêm tại:

https://machinelearningcoban.com/2018/01/14/id3/#-lap-trinh-python-cho-id