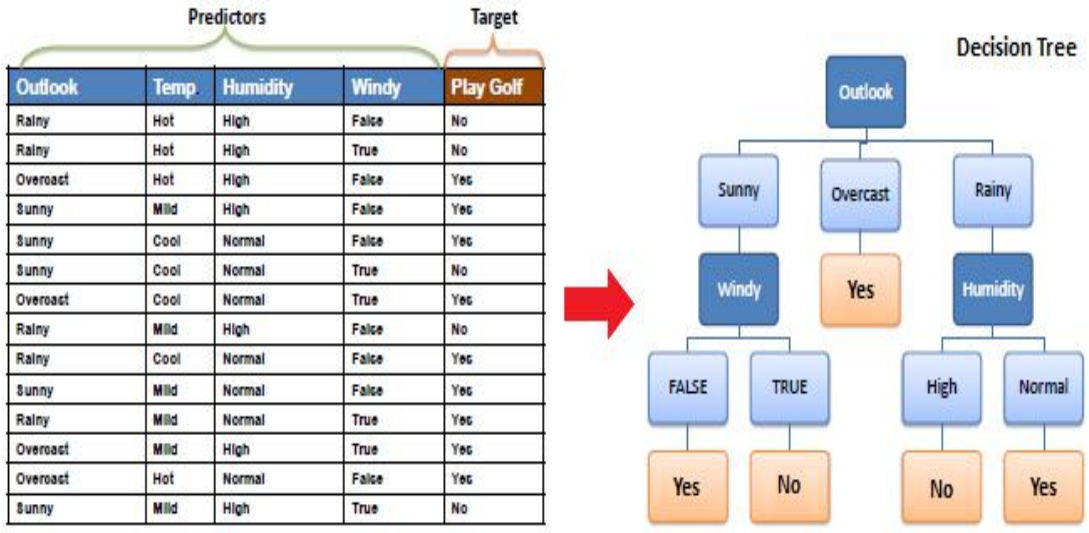
Tên sinh viên: Nguyễn Vũ Khánh Huy

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

2. 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.

- Hiệu quả trong việc xử lý các tập dữ liệu lớn , thêm vào đó là khả năng huấn luyện nhanh nếu so với các kỹ thuật phân loại khác điển hình là neural networks.
- 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.
- Quá trình phân tách nhánh ở mỗi node đều liên đến quá trình phân tách nhánh ở các node trước đó nên kết quả là sự tương quan giữa các thuộc tính, độ tin cậy của mô hình được thể hiện rõ ràng ở đầu ra.
- Ít cần đến hoạt động chuẩn hoá dữ liệu vì kỹ thuật cây quyết định không bị ảnh hưởng bởi các mẫu dữ liệu ngoại lệ.
- Kết quả trả về của cây quyết định có thể được thể hiện qua các ký hiệu để cung cấp cho phần phác thảo tổng quát tập dữ liệu.
- Quá trình trả về của cây quyết định có thể được chuyển đổi thành luật phân loại (Quinlan, 1993) và ngôn ngữ truy vấn (SQL).

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.
- Đặc điểm của cây quyết định là phân nhánh liên tục dựa trên các biểu thức logic hay toán học ở mỗi node cho đến khi thấy được kết quả sau cùng và không hỗ trợ khả năng truy vấn ngược, truy vấn ngược là kỹ thuật giúp truy tìm lỗi nếu có xảy ra.

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

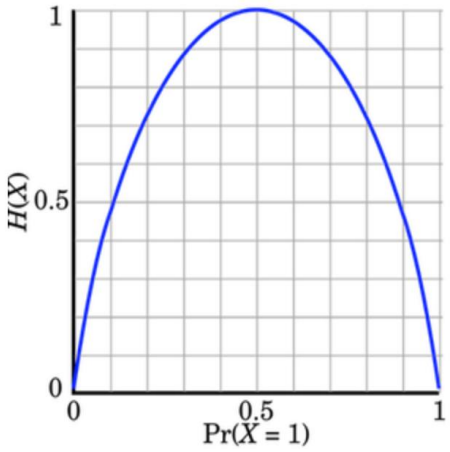
$x_1, x_2, ..., x_n$. . Giả sử rằng xác suất để x nhận các giá trị này là $p_i = p\ (x = x_i)$.

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

Entropy của phân phối này là:

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

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 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:

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

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 $\mathrm{S_1}$, S_2, ..., S_k với số điểm trong mỗi child node lần lượt là

$\mathrm{m_1}$, m_2, ..., m_k. Ta định nghĩa.

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

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:

$$x^* = \arg\max_x \mathrm{G}(x, \mathrm{S}) = \arg\min_x \mathrm{H}(x, S)$$

Tức thuộc tính khiến cho information gain đạt giá trị lớn nhất.