



ID3 Algorithm Research in Data Mining. Apply ID3 Algorithm in Some Machine Learning Problems, Making Decisions. Install Problems in Python Programming Language

Tran Thi Gia

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

June 28, 2023

NGHIÊN CỨU THUẬT TOÁN ID3 TRONG LĨNH VỰC KHAI PHÁ DỮ LIỆU. ÁP DỤNG THUẬT TOÁN ID3 TRONG MỘT SỐ BÀI TOÁN VỀ HỌC MÁY, ĐƯA RA QUYẾT ĐỊNH. CÀI ĐẶT BÀI TOÁN BẰNG NGÔN NGỮ LẬP TRÌNH PYTHON

Trần Thị Gia

Khoa Công nghệ thông tin, Trường Đại học Sư phạm Kỹ thuật Vinh

Tranthigia40@gmail.com

TÓM TẮT— Khai phá dữ liệu (data mining -DM) có thể nói là một lĩnh vực non trẻ nhưng đầy hứa hẹn của Công nghệ thông tin. Khai phá dữ liệu được hiểu như là một quá trình thống kê tính toán để tìm ra yếu tố quyết định trong các bộ dữ liệu lớn. Đây là một lĩnh vực liên quan đến thống kê, học máy, csdl,... Cây quyết định giống như các nhà chiến lược đưa ra các quyết định kịp thời và nhanh chóng dựa vào sự phân tích cây quyết định. Trong thời đại công nghiệp 4.0 ai có được thông tin và đưa ra được quyết định sớm và kịp thời thì người đó sẽ thành công. Ngày nay kỹ thuật khai phá dữ liệu bằng cây quyết định được sử dụng phổ biến trong học máy và xử lý dữ liệu lớn. Mấu chốt và ý nghĩa sâu xa trong các thuật toán xây dựng cây quyết định lại là việc tính toán "mức độ phân biệt của một thuộc tính so với thuộc tính quyết định trong dữ liệu". Trong lý thuyết thông tin đã chỉ ra công thức để thể hiện được mối quan hệ này, đây cũng là nền tảng xây dựng nên thuật toán ID3. Trong bài viết này, tôi xin phép được đề cập lại công thức về mối liên hệ giữa entropy và mức độ phân biệt của một thuộc tính trong khai phá dữ liệu, thuật toán ID3 và cài đặt thuật toán thông qua một vài ví dụ thực tiễn.

Từ khóa— Khai phá dữ liệu, cây quyết định, entropy, thuộc tính quyết định, học máy, python, ...

I. CÁC KHÁI NIỆM CƠ BẢN

1. Entropy

Entropy là độ đo mức nghi ngờ, mức ngẫu nhiên của một thông tin C. Entropy càng cao thì mức độ nghi ngờ về thông tin C càng lớn. Nói một cách khác, entropy là độ bất ngờ của một tin, về mặt ý nghĩa thì entropy tỷ lệ thuận với lượng tin, về mặt giá trị thì lượng tin của một tin C chính bằng entropy của chính tin đó:

$$\text{Entropy (C)} = \text{Lượng tin (C)}$$

2. Cây quyết định

Kỹ thuật cây quyết định là một công cụ mạnh và hiệu quả trong việc phân lớp và dự báo. Các đối tượng dữ liệu được phân thành các lớp. Các giá trị của đối tượng dữ liệu chưa biết sẽ được dự đoán, dự báo. Tri thức được rút ra trong kỹ thuật này thường được mô tả dưới dạng tường minh, đơn giản, trực quan, dễ hiểu đối với người sử dụng.[1]

3. Khai phá dữ liệu

Theo bách khoa toàn thư, *khai phá dữ liệu (DM)* là khâu chủ yếu trong quá trình phát hiện tri thức từ [dữ liệu](#) để trợ giúp cho việc làm quyết định trong quản lý. DM sử dụng nhiều phương pháp của [phân tích thống kê](#), của lý thuyết nhận dạng, của các hệ học máy, các mạng nơron nhân tạo... nhằm phát hiện các mẫu hình tri thức trực tiếp từ các kho dữ liệu. DM và phát hiện tri thức là những hướng nghiên cứu mới trong tổ chức và khai thác các [hệ thống thông tin](#) và trợ giúp quyết định.[1]

Thuật ngữ DM do Fayyad Smyth và Piatestky-Shapiro đề xuất năm 1989. Có rất nhiều định nghĩa khác nhau về DM đã được đưa ra. *Theo nghĩa đơn giản nhất, DM là việc trích lọc tri thức từ một lượng lớn dữ liệu.*[1]

4. Học máy

Học máy là môn khoa học (và cả nghệ thuật) về cách lập trình máy tính để chúng có thể học từ dữ liệu. Nói cách khác, học máy là lĩnh vực nghiên cứu nhằm giúp máy tính có khả năng học mà không cần phải lập trình tường minh.[2]

5. Ngôn ngữ lập trình python

Python là ngôn ngữ lập trình bậc cao hướng đối tượng được Guido van Rossum cùng các cộng sự tạo ra năm 1991 dành cho mục đích lập trình đa năng. Cấu trúc chương trình của python dễ đọc, rõ ràng, dễ viết hơn các ngôn ngữ lập trình hiện đại khác. Python là 1 ngôn ngữ lập trình mã nguồn mở, có một cộng đồng người sử dụng chiếm số lượng lớn nhất hiện nay.[3]

II. THUẬT TOÁN ID3[1]

ID là viết tắt của 'iterative dichotomiser' = chia đôi nhiều lần.

1. Lịch sử ra đời

Thuật toán học quy nạp lần đầu tiên được Hunt sử dụng trong hệ thống học khái niệm CLS – concept learning system, 1966.

Sau đó nó được Ross Quinlan phát triển thành ID3 năm 1979.

Được dùng trong hầu hết các hệ quy nạp

2. Bài toán

- Có một bảng gồm nhiều cột, mỗi cột là một **dấu hiệu** (thuộc tính ..),
- Một cột là **kết luận chỉ 2 khả năng “có” | “không”**.
- Mỗi dòng của bảng là một trường hợp (có được từ người chuyên gia, từ kinh nghiệm quá khứ ..).
- Cây quyết định biểu diễn tri thức từ bảng này.

Các nút = lựa chọn, rẽ thành nhiều nhánh, tùy theo giá trị của một dấu hiệu (thuộc tính..).

- Nút lá là một phương án quyết định: **có | không**.

* Ví dụ

Một người lạc trên hoang đảo. Để sống, cần phải thử xem loại quả nào ăn được loại nào độc. Sau nhiều lần thử, sẽ lập được **bảng thống kê** sau

Bảng 1. Identifying what's good to eat

Conclusion	Skin	Colour	Size	Flesh
Safe	Hairy	brown	large	Hard
Safe	Hairy	green	large	Hard
Dangerous	Smooth	red	large	Soft
Safe	Hairy	green	large	Soft
Safe	Hairy	red	small	Hard
Safe	Smooth	red	small	Hard
Safe	Smooth	brown	small	Hard
Dangerous	Hairy	green	small	Soft
Dangerous	Smooth	green	small	Hard
Safe	Hairy	red	large	Hard
Safe	Smooth	brown	large	Soft
Dangerous	Smooth	green	small	Soft
Safe	Hairy	red	small	Soft
Dangerous	Smooth	red	large	hard
Safe	Smooth	red	small	hard
Dangerous	Hairy	green	small	hard

Mỗi dòng trong bảng là một ví dụ học.

Bảng là tập ví dụ học

Mỗi dòng có thể coi là một **luật** dạng

```

IF  skin = hairy
    and colour = brown
    and size = large
    and flesh = hard
THEN conclusion = safe
  
```

Có thể tạo ra một **cây quyết định** để thay thế **tập hợp các luật**

Một nút lựa chọn của cây quyết định có dạng tổng quát như sau

IF thuộc tính = giá trị1 then <subtree 1>

else if thuộc tính = giá trị2 then <subtree 2>

else if ...

.....

else if thuộc tính = giá trịN then <subtree N>

Trên cây quyết định, một đường đi từ gốc đến nút lá sẽ ứng với một luật.

3. Thuật toán CLS

Xuất phát: cây rỗng. Bổ xung thêm dần các nút cho đến khi cây quyết định phân lớp được đúng tất cả các ví dụ trong tập học C.

1- nếu các ví dụ trong C **đều đúng** thì tạo nút lá **“có”** và kết thúc

nếu các ví dụ trong C **đều sai** thì tạo nút lá **“không”** và kết thúc. Trái lại

2- chọn một thuộc tính **A** có các giá trị **V₁, V₂ .. V_m**. Tạo nút quyết định **m** nhánh.

3- Chia tập ví dụ học thành **m** tập con **C₁, C₂ .. C_m** **tùy theo giá trị của thuộc tính đã chọn**

4- Quay lại từ bước 1.

Trong thuật toán CLS, việc **chọn** thuộc tính A ở bước 2 là **ngẫu nhiên**.

Thuật toán ID3 sẽ cải tiến để tăng hiệu quả bằng cách chọn thuộc tính có **độ phân biệt cao nhất**.

4. Entropy và mức độ phân biệt của một thuộc tính

Thế nào là độ phân biệt của một thuộc tính

Lí thuyết thông tin cho phép **lượng hóa** thông tin.

Một cách tổng quát, giả sử kết luận **C** có thể nhận một trong **n** giá trị **c₁, c₂, .. c_n**. Trong ví dụ trên, C nhận 2 giá trị **“ăn được”**, **“độc”**.

Giả sử **thuộc tính A** có thể nhận **m** giá trị **a₁, a₂, ... a_m**.

Kí hiệu xác suất điều kiện

$P(C = c_i | A = a_j)$ hay gọn hơn $P(c_i | a_j)$.

Ví dụ $P(C = \text{safe} | \text{Skin} = \text{hairy}) = 6/8 = 3/4$,

(8 dòng với Skin=hairy,

(trong đó 6 dòng kết luận C = safe.

Theo lí thuyết thông tin (xem Claude Shannon):

- $\text{Entropy}(C) = - \sum P(C = c_i) \log_2 P(C = c_i)$
- Entropy của thông tin $A = a_j$ đối với kết luận C

Biểu thức $- \log_2 P(c_i | a_j)$ là lượng tin mà $A = a_j$ mang lại cho kết luận $C = c_i$. Tổng theo $i = 1 \dots n$ là entropy của thông tin $A = a_j$ đối với kết luận C.

$\text{Entropy}(a_j) = - \sum P(c_i | a_j) \log_2 P(c_i | a_j)$

Entropy của thuộc tính A đối với C được định nghĩa là tổng

$\text{Entropy}(A) = - \sum P(A = a_j) * \text{Entropy}(a_j)$
 $= - \sum P(A = a_j) \sum P(c_i | a_j) \log_2 P(c_i | a_j)$

trong đoạn [0,1].

Entropy là độ đo mức nghi ngờ, mức ngẫu nhiên của kết luận.

Nó càng cao thì nghi ngờ về kết luận C càng lớn.

Entropy càng thấp thì a_j càng mang nhiều thông tin về kết luận C.

Entropy = 0 nghĩa là tất cả các ví dụ thuộc cùng 1 lớp, có cùng một kết luận.

Entropy = 1 nghĩa là hoàn toàn ngẫu nhiên

Ví dụ

Xét thuộc tính Size trong ví dụ trên. Từ bảng dữ liệu ta có

$$P(\text{safe} \mid \text{large}) = 5/7$$

$$P(\text{dangerous} \mid \text{large}) = 2/7$$

$$P(\text{large}) = 7/16$$

$$P(\text{safe} \mid \text{small}) = 5/7$$

$$P(\text{dangerous} \mid \text{small}) = 5/9$$

$$P(\text{small}) = 9/16$$

Như vậy entropy của thuộc tính Size đối với kết luận “safe | dangerous” là

$$7/16 \{5/7 \log 5/7 + 2/7 \log 2/7\} + 9/16 \{5/9 \log 5/9 + 4/9 \log 4/9\} = 0.9350955$$

Định nghĩa **lượng tin mà thuộc tính A mang lại đối với tập ví dụ**

$$\text{Gain}(C, A) = \text{Entropy}(C) - \text{Entropy}(A)$$

ở đây:

Thuộc tính có entropy thấp nhất chính là có độ phân biệt cao nhất (cho kết luận C).

Căn cứ của định nghĩa entropy

k khả năng có thể, xs tương ứng là $p_1, p_2, \dots, p_k$.

độ đo mức không chắc chắn phải là một hàm của các xác suất này

$$H(p_1, p_2, \dots, p_k).$$

H cần thỏa mãn các điều kiện sau:

1. là **hàm liên tục** của các đối
2. Nếu các xác suất là đồng đều $= 1/k$ thì **entropy phải tăng khi k tăng** (càng nhiều khả năng thì mức không chắc càng lớn)
3. độc lập với cách phân chia các khả năng: nếu tập hợp các khả năng lựa chọn được chia nhỏ dần nhiều mức thì entropy của tập ban đầu phải bằng tổng (có trọng số) của các entropy của các tập con.

Ví dụ, xét việc gieo xúc sắc có 6 khả năng có thể chia thành ba trường hợp: “bé hơn hay bằng 3”, “4 hoặc 5”, “6” hoặc chia dần thành 2 mức: “bé hơn hay bằng 3”, “lớn hơn 3” và sau đó phân chia tiếp “lớn hơn 3” thành “6” hoặc “trái lại (4 hoặc 5)”. Phải có $H(1/2, 1/3, 1/6) = H(1/2, 1/2) + \frac{1}{2} H(2/3, 1/3)$

5. Thuật toán ID3

5.1. Ý tưởng

ID3 là cải tiến từ thuật toán CLS

1- CLS làm việc với toàn bộ tập thí dụ học có sẵn từ đầu. ID3 **giảm số lượng thí dụ học**, dùng một **tập con** xuất phát.

2- ở mỗi bước ID3 chọn thuộc tính có mức phân biệt cao nhất để phân nhánh.

Mỗi thí dụ sẽ gán một thông báo “có” hay “+” và “không” hay “-”

Tính lượng thông tin của thông báo qua các xác suất (xấp xỉ bằng tần xuất) đối với tập thí dụ C.

$$M(C) = - \sum p^+ \log_2 p^+ - \sum p^- \log_2 p^-$$

Tổng thứ nhất ứng với các thí dụ trong lớp “có”,

Tổng thứ hai ứng với các thí dụ trong lớp “không”

Chọn thuộc tính A_j nào

- C được chia thành m tập con $C_1, C_2 \dots C_m$

ứng với các giá trị $a_{1j}, a_{2j} \dots a_{mj}$ của thuộc tính A_j .

- Kí hiệu $p_{ij} = P\{A_j = a_{ij}\}$

$$B(C, A_j) = \sum p_{ij} * M(C_i)$$

Thuật toán chọn A_j sao cho $M(C) - B(C, A_j)$ cực đại

5.2. Các bước của thuật toán ID3

- 1- chọn ngẫu nhiên một tập con W của tập các ví dụ học, gọi là cửa sổ
- 2- Áp dụng thuật toán CLS tạo cây (hay luật) quyết định cho W
- 3- Duyệt toàn bộ các ví dụ còn lại (trừ W) trên cây để phát hiện các ngoại lệ
- 4- Nếu có ví dụ là ngoại lệ, thêm vào W và lặp lại từ bước 2. Trái lại, kết thúc cho kết quả là cây nhận được

Chi tiết bước 2

1. Tính entropy của tất cả các thuộc tính (đối với kết luận cần quyết định);
2. Chọn thuộc tính (ví dụ A) có entropy thấp nhất
3. Chia tập ví dụ thành các tập con tùy theo giá trị của thuộc tính A. A nhận cùng một giá trị trên mỗi tập con.
4. Xây dựng cây phân nhánh theo giá trị của A:
if A=a1 then ... (subtree1)
if A=a2 then ... (subtree2)
...etc...
5. Lặp lại từ bước 1 với mỗi cây con.
6. Mỗi lần lặp xét được 1 thuộc tính. Quá trình dừng khi đã xét hết các thuộc tính, hoặc không cần phân nhánh nữa (vì mọi ví dụ trong một cây con đã có cùng kết luận).

III. CÁC VÍ DỤ MINH HỌA THUẬT TOÁN ID3

1. Bài toán xây dựng cây quyết định phân lớp mức lương

Mức lương của các đối tượng có các dữ kiện được cho trong bảng sau:

ID (Định danh), age (tuổi), salary (lương).

Phân lớp mức lương thành 2 lớp: tốt (good), xấu (bad).

Bảng tập dữ liệu:

Bảng 2: Tập dữ liệu huấn luyện quyết định phân lớp mức lương

ID	Age	Salary	Class
1	30	65	Good
2	23	15	Bad
3	40	70	Good
4	55	40	Bad
5	55	100	Good
6	45	60	Good

1.1. Xây dựng cây quyết định dựa vào lý thuyết thông tin entropy

a. Xác định nút gốc:

Để xác định nút gốc, theo lý thuyết thông tin, chúng ta sẽ đi tính Gain của tất cả các thuộc tính đối với thuộc tính quyết định: Ở đây chúng ta có 4 thuộc tính: Thứ tự, lương, tuổi và thuộc tính phân lớp, trong đó thuộc tính phân lớp (class) là thuộc tính quyết định. Ở đây là kí hiệu thuộc tính quyết định là C.

Ta có $C = 6$; $C_{\text{Good}} = 4$; $C_{\text{Bad}} = 2$

$$\text{Entropy}(C) = -4/6 * \log_2(4/6) - 2/6 * \log_2(2/6) = 0.918$$

Xét nút: Tuổi = {<=34, >34}: tuổi được chia làm 2 vùng giá trị (lớn hơn 34 tuổi và bé hơn hoặc bằng 34 tuổi)

$I_{<=34} = \{1, 2\} = I_1$ (có 2 người có tuổi thỏa mãn I_1 , trong đó 1 người có kết luận tốt, 1 người có kết luận xấu)

$I_{>34} = \{3, 4, 5, 6\} = I_2$ (có 4 người thuộc I_2 , trong đó có 3 người nhận kết luận tốt, 1 người nhận kết luận xấu)

$$\text{Entropy}(<=34) = I(1, 1) = -1/2 * \log_2(1/2) - 1/2 * \log_2(1/2) = 1$$

$$\text{Entropy}(>34) = I(3, 1) = -3/4 * \log_2(3/4) - 1/4 * \log_2(1/4) = 0.811$$

$$\text{Entropy}(\text{Tuổi}) = 2/6 * I_1 + 4/6 * I_2 = 2/6 * 1 + 4/6 * 0.811 = 0.874$$

$$\text{Gain}(C, \text{Túổi}) = 0.918 - 0.874 = 0.044$$

Xét nút: Lương (<=50, >50)

$$I_{<=50} = \{2, 4\}$$

$$I_{>50} = \{1, 3, 5, 6\}$$

$$\text{Entropy}(<=50) = I(0, 2) = 0 = -0 * \log_2(0) - 2 * \log_2(2/2) = 0$$

$$\text{Entropy}(>50) = I(4, 0) = 0$$

$$\text{Gain}(C, \text{Lương}) = 0.918 - 0 = 0.918$$

Kết quả:

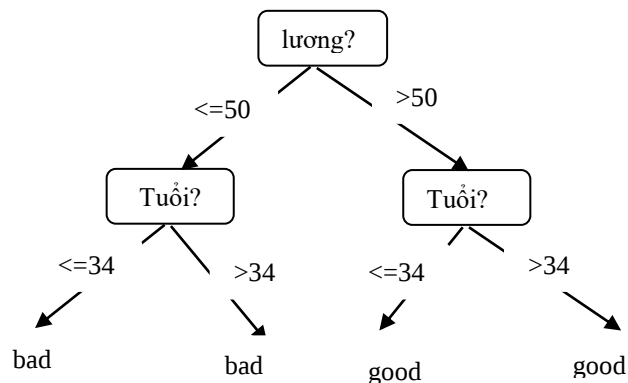
$$\text{Gain}(C, \text{Túổi}) = 0.918 - 0.874 = 0.044$$

$$\text{Gain}(C, \text{Lương}) = 0.918 - 0 = 0.918$$

Thuộc tính Lương có Gain cao nhất, do đó nó được dùng làm nút gốc của cây quyết định

b. Cây quyết định

Phân lớp mức lương ứng với dữ liệu trong bảng trên có dạng:



Hình 1. Cây quyết định phân lớp (bad/good) mức lương

1.2. Luật suy diễn:

If (Tuổi <= 34) and (Lương <= 50) then class = Bad

If (Tuổi <= 34) and (Lương > 50) then class = Good

If (Tuổi > 34) and (Lương <= 50) then class = Bad

If (Tuổi > 34) and (Lương > 50) then class = Good

2. Bài toán xây dựng cây quyết định cho tập dữ liệu huấn luyện quyết định chơi hay không chơi dựa vào yếu tố thời tiết

Bảng 3. Tập dữ liệu huấn luyện quyết định chơi hay không chơi dựa vào yếu tố thời tiết

Quang cảnh	Nhiệt độ	Độ ẩm	Gió	Chơi
Nắng	Nóng	Cao	Nhẹ	Không
Nắng	Nóng	Cao	Mạnh	Không
Âm u	Nóng	Cao	Nhẹ	Có
Mưa	Ấm áp	Cao	Nhẹ	Có
Mưa	Mát	Trung bình	Nhẹ	Có
Mưa	Mát	Trung bình	Mạnh	Không
Âm u	Mát	Trung bình	Mạnh	Có
Nắng	Ấm áp	Cao	Nhẹ	Không
Nắng	Mát	Trung bình	Nhẹ	Có
Mưa	Ấm áp	Trung bình	Nhẹ	Có
Nắng	Ấm áp	Trung bình	Mạnh	Có
Âm u	Ấm áp	Cao	Mạnh	Có
Âm u	Nóng	Trung bình	Nhẹ	Có
Mưa	Ấm áp	Cao	Mạnh	Không

2.1. Xây dựng cây quyết định

a. Xác định thuộc tính làm nút gốc cây.

Theo lý thuyết thông tin, chúng ta sẽ đi tính Gain của tất cả các thuộc tính

Gain(C, Gió)

$$\text{Gain}(C, \text{Gió}) = \text{Entropy}(C) - (8/14) * \text{Entropy}(\text{Nhẹ}) - (6/14) * \text{Entropy}(\text{Mạnh})$$

$$\text{Entropy}(\text{Nhẹ}) = - (6/8) * \log_2(6/8) - (2/8) * \log_2(2/8) = 0.811$$

$$\text{Entropy}(\text{Mạnh}) = - (3/6) * \log_2(3/6) - (3/6) * \log_2(3/6) = 1.00$$

Vậy

$$\text{Gain}(C, \text{Gió}) = 0.940 - (8/14) * 0.811 - (6/14) * 1.00 = 0.048$$

Gain(C, Quang cảnh)

$$\text{Gain}(C, \text{Quang cảnh}) = \text{Entropy}(C) - (5/14) * \text{Entropy}(\text{Nắng}) - (4/14) * \text{Entropy}(\text{Âm u}) - (5/14) * \text{Entropy}(\text{Mưa})$$

$$\text{Entropy}(\text{Nắng}) = - (3/5) * \log_2(3/5) - (2/5) * \log_2(2/5) = 0.97$$

$$\text{Entropy}(\text{Âm u}) = - (4/4) * \log_2(4/4) = 0$$

$$\text{Entropy}(\text{Mưa}) = - (3/5) * \log_2(3/5) - (2/5) * \log_2(2/5) = 0.97$$

Vậy

$$\text{Gain}(C, \text{Quang cảnh}) = 0.940 - 0.97 * 5/14 - 0 * 4/14 - 0.97 * 5/14 = 0.247$$

Gain(C, Nhiệt độ)

$$\text{Gain}(C, \text{Nhiệt độ}) = \text{Entropy}(C) - (4/14) * \text{Entropy}(\text{Nóng}) - (6/14) * \text{Entropy}(\text{Ấm áp}) - (4/14) * \text{Entropy}(\text{Mát})$$

$$\text{Entropy}(\text{Nóng}) = - (2/4) \cdot \log_2(2/4) - (2/4) \cdot \log_2(2/4) = 1$$

$$\text{Entropy}(\text{Ám áp}) = - (4/6) \cdot \log_2(4/6) - (2/6) \cdot \log_2(2/6) = 0.918$$

$$\text{Entropy}(\text{Mát}) = - (3/4) \cdot \log_2(3/4) - (1/4) \cdot \log_2(1/4) = 0.811$$

Vậy

$$\text{Gain}(\text{C}, \text{Nhiệt độ}) = 0.940 - 1 \cdot 4/14 - 0.918 \cdot 6/14 - 0.811 \cdot 4/14 = 0.0291$$

Gain(C, Độ ẩm)

$$\text{Gain}(\text{C}, \text{Độ ẩm}) = \text{Entropy}(\text{C}) - (7/14) \cdot \text{Entropy}(\text{Cao}) - (7/14) \cdot \text{Entropy}(\text{Trung bình})$$

$$\text{Entropy}(\text{Cao}) = - (3/7) \cdot \log_2(3/7) - (4/7) \cdot \log_2(4/7) = 0.985$$

$$\text{Entropy}(\text{Trung bình}) = - (6/7) \cdot \log_2(6/7) - (1/7) \cdot \log_2(1/7) = 0.591$$

Vậy

$$\text{Gain}(\text{C}, \text{Độ ẩm}) = 0.940 - 0.985 \cdot 7/14 - 0.591 \cdot 7/14 = 0.153$$

Thuộc tính **Quang cảnh (Outlook)** có **gain cao nhất**. Do đó, nó được dùng làm nút gốc.

b. Xây dựng cây quyết định

Outlook có 3 giá trị, cần phân 3 nhánh *Nắng, Âm u, Mưa*.

Tiếp tục phân nhánh.

Xét nút *Nắng*.

Chỉ còn các thuộc tính **Nhiệt độ, Độ Ẩm, Gió**.

$$C_{\text{Nắng}} = 5 \rightarrow \text{Entropy}(C_{\text{Nắng}}) = - (3/5) \cdot \log_2(3/5) - (2/5) \cdot \log_2(2/5) = 0.97$$

- Xét thuật tính Nhiệt độ

$$\text{Entropy}(\text{Nóng}) = - (2/2) \cdot \log_2(2/2) = 0$$

$$\text{Entropy}(\text{Ám áp}) = - (1/2) \cdot \log_2(1/2) - (1/2) \cdot \log_2(1/2) = 1$$

$$\text{Entropy}(\text{Mát}) = - (1/1) \cdot \log_2(1/1) = 0$$

$$\text{Gain}(C_{\text{Nắng}}, \text{Nhiệt độ}) = 0.57$$

- Xét thuật tính Độ ẩm

$$\text{Entropy}(\text{Cao}) = - (3/3) \cdot \log_2(3/3) = 0$$

$$\text{Entropy}(\text{Trung bình}) = - (2/2) \cdot \log_2(2/2) = 0$$

$$\text{Gain}(C_{\text{Nắng}}, \text{Độ ẩm}) = 0.97$$

- Xét thuật tính Gió

$$\text{Entropy}(\text{Nhẹ}) = - (2/3) \cdot \log_2(2/3) - (1/3) \cdot \log_2(1/3) = 0.918$$

$$\text{Entropy}(\text{Mạnh}) = - (1/2) \cdot \log_2(1/2) - (1/2) \cdot \log_2(1/2) = 1$$

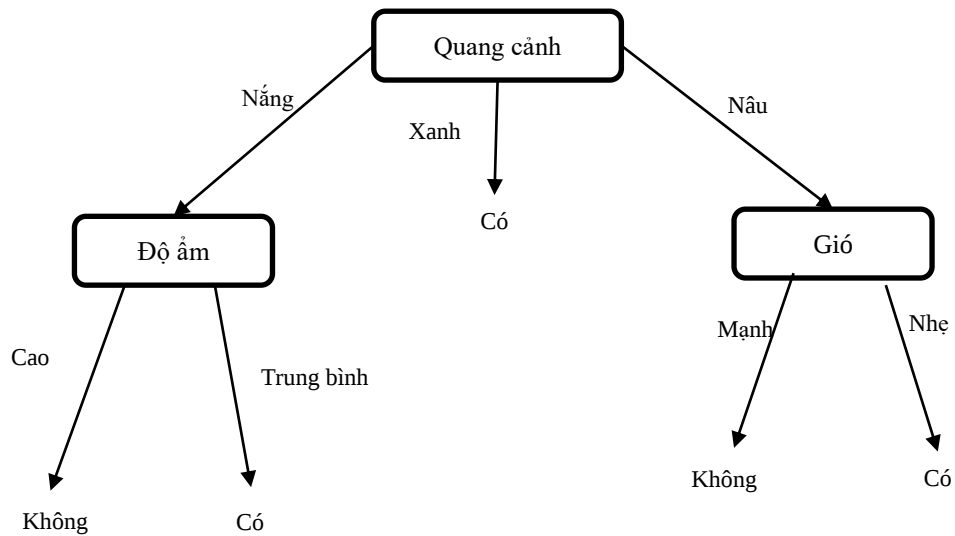
$$\text{Gain}(C_{\text{Nắng}}, \text{Gió}) = 0.019$$

Thuộc tính Độ ẩm có gain cao nhất. Lấy nó làm nút phân nhánh tiếp theo. Lặp lại quá trình

Độ ẩm = Cao $\rightarrow$ Kết luận = Không

Độ ẩm = Trung bình $\rightarrow$ Kết luận = Có

Tương tự với các nút **Âm u** , **Mưa** ta được:



Hình 2. Cây quyết định của bài thời tiết

2.2. Luật suy dẫn

Cây quyết định thể hiện các luật, là tri thức rút ra từ bảng trên.

IF Quang cảnh = Nắng AND Độ ẩm = Cao THEN Chơi = Không

IF Quang cảnh = Mưa AND Độ ẩm = Cao THEN Chơi = Không

IF Quang cảnh = Mưa AND Gió = Mạnh THEN Chơi = Có

IF Quang cảnh = Âm u THEN Chơi = Có

IF Quang cảnh = Mưa AND Gió = Nhẹ THEN Chơi = Có

IV. CÀI ĐẶT CHƯƠNG TRÌNH BẰNG NGÔN NGỮ LẬP TRÌNH PYTHON MINH HỌA THUẬT TOÁN ID3 XÂY DỰNG CÂY QUYẾT ĐỊNH CHO TẬP DỮ LIỆU CỦA BÀI TOÁN THỜI TIẾT

1. Code chương trình lập trình bằng python

```
import tkinter as tk
from tkinter import messagebox
from tkinter import ttk
from collections import Counter
from math import log
# Dữ liệu huấn luyện
training_data = [
    {'Quang cảnh': 'Nắng', 'Nhiệt độ': 'Nóng', 'Độ ẩm': 'Cao', 'Gió': 'Nhẹ', 'Chơi': 'Không'},
    {'Quang cảnh': 'Nắng', 'Nhiệt độ': 'Nóng', 'Độ ẩm': 'Cao', 'Gió': 'Mạnh', 'Chơi': 'Không'},
    {'Quang cảnh': 'Âm u', 'Nhiệt độ': 'Nóng', 'Độ ẩm': 'Cao', 'Gió': 'Nhẹ', 'Chơi': 'Có'},
    {'Quang cảnh': 'Mưa', 'Nhiệt độ': 'Ấm áp', 'Độ ẩm': 'Cao', 'Gió': 'Nhẹ', 'Chơi': 'Có'},
```

```
{'Quang cảnh': 'Mưa', 'Nhiệt độ': 'Mát', 'Độ ẩm': 'Trung bình', 'Gió': 'Nhẹ', 'Chơi': 'Có'},
{'Quang cảnh': 'Mưa', 'Nhiệt độ': 'Mát', 'Độ ẩm': 'Trung bình', 'Gió': 'Mạnh', 'Chơi': 'Không'},
{'Quang cảnh': 'Ám u', 'Nhiệt độ': 'Mát', 'Độ ẩm': 'Trung bình', 'Gió': 'Mạnh', 'Chơi': 'Có'},
{'Quang cảnh': 'Nắng', 'Nhiệt độ': 'Ám áp', 'Độ ẩm': 'Cao', 'Gió': 'Nhẹ', 'Chơi': 'Không'},
{'Quang cảnh': 'Nắng', 'Nhiệt độ': 'Mát', 'Độ ẩm': 'Trung bình', 'Gió': 'Nhẹ', 'Chơi': 'Có'},
{'Quang cảnh': 'Mưa', 'Nhiệt độ': 'Ám áp', 'Độ ẩm': 'Trung bình', 'Gió': 'Nhẹ', 'Chơi': 'Có'},
{'Quang cảnh': 'Nắng', 'Nhiệt độ': 'Ám áp', 'Độ ẩm': 'Trung bình', 'Gió': 'Mạnh', 'Chơi': 'Có'},
{'Quang cảnh': 'Ám u', 'Nhiệt độ': 'Ám áp', 'Độ ẩm': 'Cao', 'Gió': 'Mạnh', 'Chơi': 'Có'},
{'Quang cảnh': 'Ám u', 'Nhiệt độ': 'Nóng', 'Độ ẩm': 'Trung bình', 'Gió': 'Nhẹ', 'Chơi': 'Có'},
{'Quang cảnh': 'Mưa', 'Nhiệt độ': 'Ám áp', 'Độ ẩm': 'Cao', 'Gió': 'Mạnh', 'Chơi': 'Không'}
```

```
def entropy(data):
    class_labels = [record['Chơi'] for record in data]
    class_counts = Counter(class_labels)
    num_records = len(data)

    entropy = 0
    for count in class_counts.values():
        probability = count / num_records
        entropy -= probability * log(probability, 2)

    return entropy

def information_gain(data, attribute):
    attribute_values = set(record[attribute] for record in data)
    num_records = len(data)
    subset_entropy = 0

    for value in attribute_values:
        subset = [record for record in data if record[attribute] == value]
        subset_probability = len(subset) / num_records
        subset_entropy += subset_probability * entropy(subset)

    return entropy(data) - subset_entropy

def build_tree(data, attributes, target_attribute):
    class_labels = [record[target_attribute] for record in data]

    if len(set(class_labels)) == 1:
        return class_labels[0]
```

```
if len(attributes) == 0:
    majority_label = Counter(class_labels).most_common(1)[0][0]
    return majority_label

best_attribute = max(attributes, key=lambda attr: information_gain(data, attr))

tree = {best_attribute: {}}
remaining_attributes = attributes - {best_attribute}

for value in set(record[best_attribute] for record in data):
    subset = [record for record in data if record[best_attribute] == value]
    subtree = build_tree(subset, remaining_attributes, target_attribute)
    tree[best_attribute][value] = subtree

return tree

def classify(record, tree):
    if isinstance(tree, dict):
        attribute = next(iter(tree))
        value = record.get(attribute)
        subtree = tree[attribute].get(value)
        if subtree is None:
            return None
        return classify(record, subtree)
    else:
        return tree

def get_decision(record, tree):
    decision = classify(record, tree)
    if decision is None:
        return 'Unknown'
    return decision

def on_predict():
    record = {
        'Quang cảnh': combo_quangcanh.get(),
        'Nhiệt độ': combo_nhietdo.get(),
        'Độ ẩm': combo_doam.get(),
        'Gió': combo_gio.get()
    }
```

```
decision = get_decision(record, tree)
result_label.configure(text=f'Chơi: {decision}')
if decision == 'Có':
    result_label.configure(foreground="green")
elif decision == 'Không':
    result_label.configure(foreground="red")
else:
    result_label.configure(foreground="black")

# Tạo giao diện Tkinter
root = tk.Tk()
root.title("ID3 Decision Tree")

# Tạo frame chứa phần nhập liệu và phần kết quả
input_frame = ttk.Frame(root, padding=20)
input_frame.grid(row=0, column=0, padx=20, pady=10)

result_frame = ttk.Frame(root, padding=20)
result_frame.grid(row=1, column=0, padx=20, pady=10)

# Label và combobox cho từng thuộc tính
label_quangcanh = ttk.Label(input_frame, text="Quang cảnh:")
label_quangcanh.grid(row=0, column=0, sticky="W", padx=5, pady=5)
combo_quangcanh = ttk.Combobox(input_frame, values=['Nắng', 'Âm u', 'Mưa'])
combo_quangcanh.grid(row=0, column=1, padx=5, pady=5)

label_nhietdo = ttk.Label(input_frame, text="Nhiệt độ:")
label_nhietdo.grid(row=1, column=0, sticky="W", padx=5, pady=5)
combo_nhietdo = ttk.Combobox(input_frame, values=['Nóng', 'Ấm áp', 'Mát'])
combo_nhietdo.grid(row=1, column=1, padx=5, pady=5)

label_doam = ttk.Label(input_frame, text="Độ ẩm:")
label_doam.grid(row=2, column=0, sticky="W", padx=5, pady=5)
combo_doam = ttk.Combobox(input_frame, values=['Cao', 'Trung bình'])
combo_doam.grid(row=2, column=1, padx=5, pady=5)

label_gio = ttk.Label(input_frame, text="Gió:")
label_gio.grid(row=3, column=0, sticky="W", padx=5, pady=5)
combo_gio = ttk.Combobox(input_frame, values=['Nhẹ', 'Mạnh'])
combo_gio.grid(row=3, column=1, padx=5, pady=5)
```

```
# Button "Dự đoán"
```

```
button_predict = ttk.Button(input_frame, text="Dự đoán", command=on_predict)
```

```
button_predict.grid(row=4, columnspan=2, pady=10)
```

```
# Label hiển thị kết quả
```

```
result_label = ttk.Label(result_frame, text="Chơi: ")
```

```
result_label.pack()
```

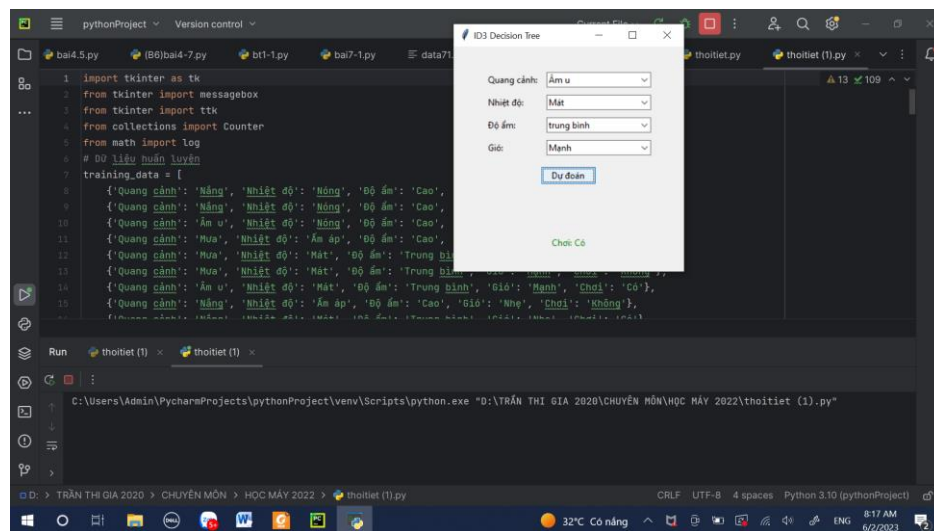
```
# Xây dựng cây quyết định ID3
```

```
attributes = {'Quang cảnh', 'Nhiệt độ', 'Độ ẩm', 'Gió'}
```

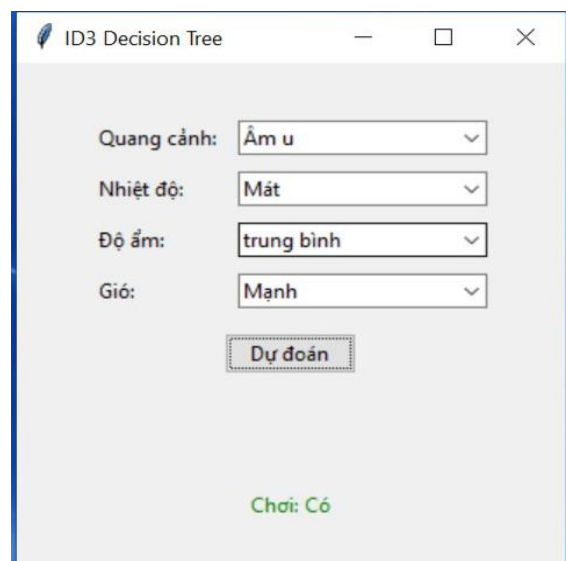
```
tree = build_tree(training_data, attributes, 'Chơi')
```

```
root.mainloop()
```

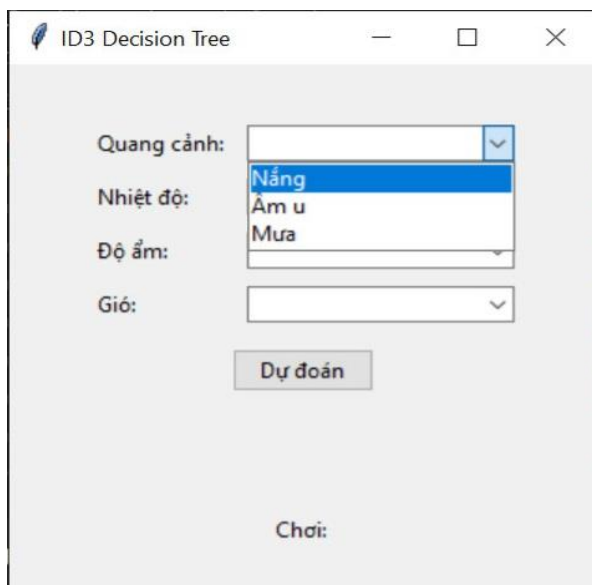
2. Demo chương trình



Hình 2. Demo toàn bộ chương trình



Hình 3. Demo chương trình



Hình 4. Demo chương trình thể hiện danh lục dữ liệu để chọn

V. TÀI LIỆU THAM KHẢO

- [1]. Bài giảng Data Mining, Trường Đại học kinh doanh và công nghệ Hà Nội, 2020
- [2]. Thực hành học máy với Scikit-Learn, Keras và TensonFlow, Tập 1- Kiến thức cơ bản về học máy, Aurélien Géron, Copyright@2019 Kiwisoft S.A.S all rights reserved ,Published by o'reilley Media, Inc,...1005 Gravenstein Highway North, Sebastopol, CA 95472, Vietnamese @2021 by Funix&MLBVN, nhóm dịch thuật Machine Learning cơ bản, NXB thông tin&truyền thông, dựa theo ấn bản lần thứ 2, 2021.
- [3]. Python cơ bản, TS. Nguyễn Văn Hậu -TS. Nguyễn Duy Tân- Th.s. Nguyễn Thị Hồng Đăng- Th.s. Nguyễn hoàng Hiệp, Nhà xuất bản đại học quốc gia Hà Nội, 2021.
- [4]. Slide bài giảng của Hà Quang Thụy, ĐH Công nghệ, ĐH Quốc Gia Hà Nội
Download tại: <http://www.coltech.vnu.edu.vn/~thuyhq/courses.htm>