Thuật toán CART

1. Mục đích ý nghĩa

Cây quyết định là một kỹ thuật mô hình hóa máy học hiệu quả không tham số và được sử dụng chủ yếu cho các vấn đề hồi quy và phân loại. Để tìm giải pháp, cây quyết định đưa ra quyết định tuần tự, phân cấp về kết quả dựa trên dữ liệu dự đoán.

Cây quyết định xây dựng mô hình hồi quy hoặc phân loại dưới dạng cấu trúc cây. Nó chia tập dữ liệu thành các tập con nhỏ hơn và nhỏ hơn đồng thời một cây quyết định liên quan được phát triển dần dần. Kết quả cuối cùng là một cây với nodes quyết định và các nodes lá.

1. Ứng dụng của thuật toán

* Lựa chọn dự phòng: Cart giúp các tổ chức xem xét các lựa chọn thay thế cho các sự kiện có thể xảy ra. Ví dụ, một tổ chức đang có kế hoạch xây dựng một sản phẩm mới. Rất nhiều nguồn lực và tiền bạc sẽ được đầu tư vào nó. Cây quyết định sẽ giúp tổ chức có cái nhìn về tất cả các lựa chọn thay thế khác nếu họ không muốn xây dựng sản phẩm từ đầu. Các lựa chọn thay thế có thể bao gồm không xây dựng sản phẩm mới hoặc mang lại sửa đổi cho các sản phẩm hiện có để kết hợp các tính năng mới trong đó
* Event: Một sự kiện ở đây là giả định cho 1 sự kiện mà xảy ra nằm ngoài tầm kiểm soát trực tiếp của bạn và là kết quả của hành động của bạn. Trong ví dụ về việc có nên thay đổi ngăn xếp kỹ thuật của nền tảng hay không, kết quả có thể có từ quyết định tiến hành sẽ là sự phát triển của một sản phẩm thành công hoặc lỗi sản phẩm.
* Kết quả: Các kết quả là kết quả của việc đưa ra quyết định, có trọng số với xác suất của các sự kiện cụ thể xảy ra. Ví dụ về phát triển sản phẩm, bạn có thể phải đầu tư 50.000 đô la để phát triển sản phẩm mới, hy vọng nó sẽ tạo ra tổng lợi nhuận 100.000 đô la. Xác suất phát triển một sản phẩm thành công có thể là 80%. Kết quả là, bạn có 80% cơ hội kiếm được 100.000 đô la, 20% cơ hội kiếm được bất cứ thứ gì nếu sản phẩm thất bại và 100% cơ hội kiếm được thêm lợi nhuận nếu bạn không phát triển sản phẩm.
* Quyết định: Bạn đang cố gắng quyết định có phê duyệt ngân sách phát triển cho một sản phẩm được cải thiện hay không. Bạn được khuyến khích làm như vậy với lý do rằng sự phát triển, nếu thành công, sẽ mang lại cho bạn lợi thế cạnh tranh, nhưng nếu bạn không phát triển sản phẩm, đối thủ cạnh tranh của bạn có thể - và có thể làm tổn hại nghiêm trọng thị phần của bạn. Cây quyết định giúp các tổ chức đưa ra các quyết định cần thiết bằng cách hình dung tất cả các hành động có thể có thể theo một quyết định cụ thể và kết quả cuối cùng cho quyết định đã chọn.

1. Ý tưởng của thuật toán

Một thuật toán có thể được minh bạch chỉ khi mọi người có thể đọc và hiểu rõ các quyết định của nó. Mặc dù deep learning là siêu sao của học máy hiện nay, nó là một thuật toán không rõ ràng và chúng ta không biết lý do của quyết định. Ở đây, các thuật toán cây quyết định vẫn giữ được sự phổ biến của chúng vì chúng có thể tạo ra các quyết định minh bạch. ID3 sử dụng information gain trong khi C4.5 sử dụng gain ratio for splitting. Ở đây, Cart là một thuật toán xây dựng cây quyết định thay thế. Nó có thể xử lý cả nhiệm vụ phân loại và hồi quy. Thuật toán này sử dụng một số liệu mới có tên là chỉ số gini để tạo các điểm quyết định cho các nhiệm vụ phân loại.

Công thức toán học của gini index có dạng là

Gini = 1 – Σ (Pi)2

Với I là số lượng của các lớp

1. Ví dụ minh họa

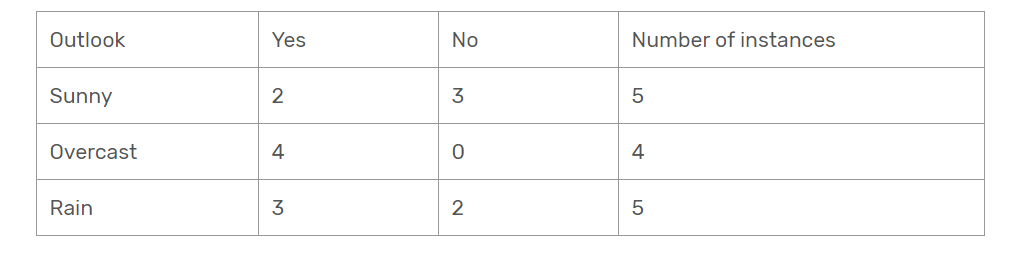
Ta có bảng số liệu dưới đây



Chúng ta sẽ xây dựng cây quyết định để biết với những điều kiện thời tiết như này thì có chơi hay không

B1: Chọn root node

* Outlook: Outlook là một tính năng danh nghĩa. Có Sunny, Overcast hoặc rainy. Tôi sẽ tóm tắt các quyết định cuối cùng cho outlook



Gini(Outlook=Sunny) = 1 – (2/5)2 – (3/5)2 = 1 – 0.16 – 0.36 = 0.48

Gini(Outlook=Overcast) = 1 – (4/4)2 – (0/4)2 = 0

Gini(Outlook=Rain) = 1 – (3/5)2 – (2/5)2 = 1 – 0.36 – 0.16 = 0.48

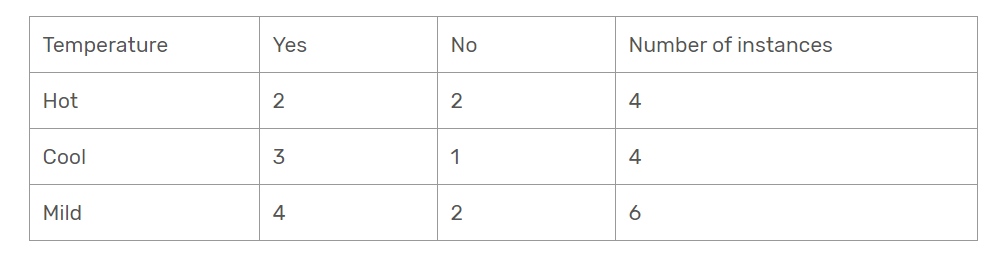
Tiếp theo chỉ số Gini là:

Gini(Outlook) = (5/14) x 0.48 + (4/14) x 0 + (5/14) x 0.48 = 0.171 + 0 + 0.171

= 0.342

* Temparature

Tương tự với outlook ta có



Gini(Temp=Hot) = 1 – (2/4)2 – (2/4)2 = 0.5

Gini(Temp=Cool) = 1 – (3/4)2 – (1/4)2 = 1 – 0.5625 – 0.0625 = 0.375

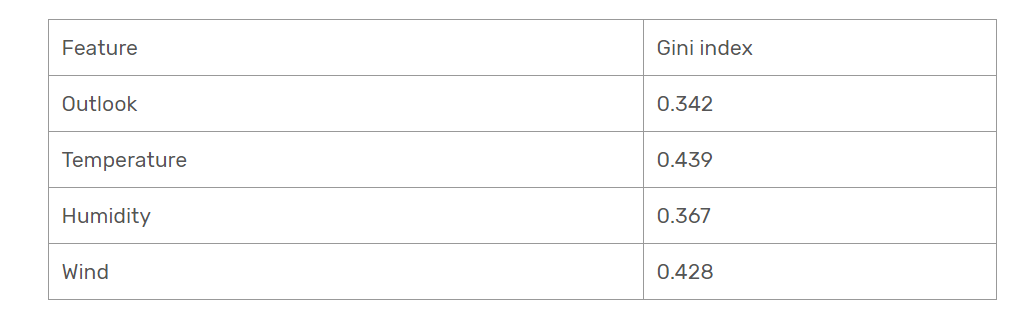
Gini(Temp=Mild) = 1 – (4/6)2 – (2/6)2 = 1 – 0.444 – 0.111 = 0.445

Và chỉ số gini của index là:

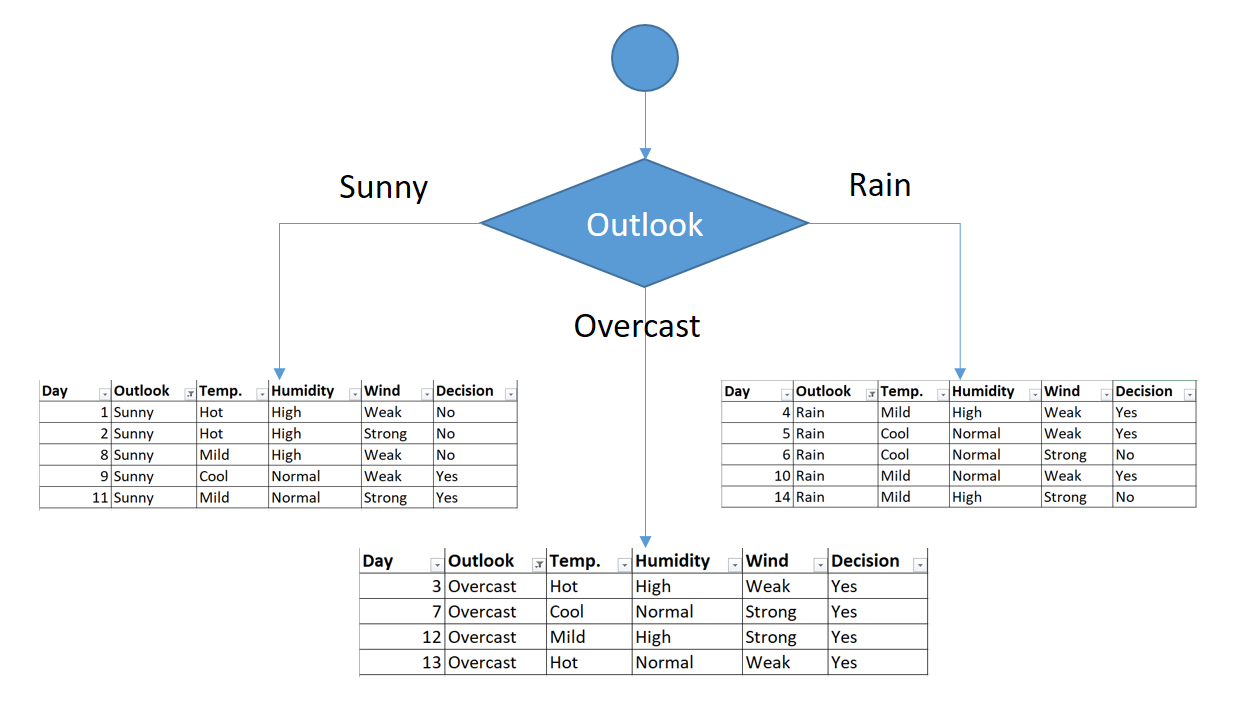
Gini(Temp) = (4/14) x 0.5 + (4/14) x 0.375 + (6/14) x 0.445 = 0.142 + 0.107 + 0.190 = 0.439

Tương tự ta có chỉ số gini của humidity = 0.367 và wind = 0.428

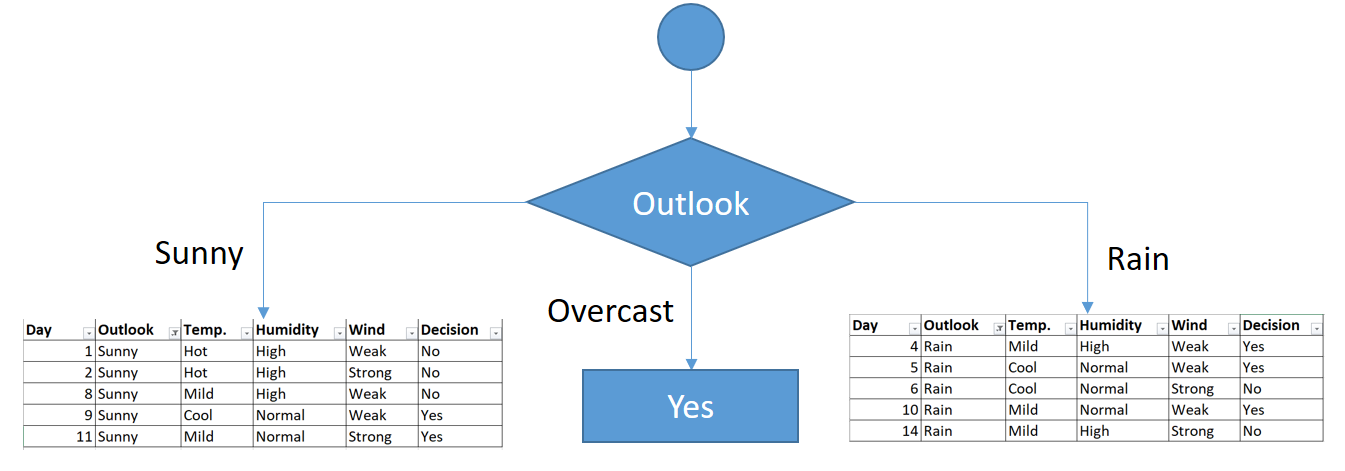
Ta có bảng tổng hợp sau đây



Ta sẽ chọn outlook vì nó có chỉ số gini thấp nhất



Chúng ta thấy overcast chỉ có yes chứ không có no nên



Bước 2: Chúng ta sẽ tập trung vào tìm nhánh cho sunny

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Day | Outlook | Temp. | Humidity | Wind | Decision |
| 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 |

* **Temperature cho sunny**

|  |  |  |  |
| --- | --- | --- | --- |
| Temperature | Yes | No | Number of instances |
| Hot | 0 | 2 | 2 |
| Cool | 1 | 0 | 1 |
| Mild | 1 | 1 | 2 |

Gini(Outlook=Sunny and Temp.=Hot) = 1 – (0/2)2 – (2/2)2 = 0

Gini(Outlook=Sunny and Temp.=Cool) = 1 – (1/1)2 – (0/1)2 = 0

Gini(Outlook=Sunny and Temp.=Mild) = 1 – (1/2)2 – (1/2)2 = 1 – 0.25 – 0.25 = 0.5

Gini(Outlook=Sunny and Temp.) = (2/5)x0 + (1/5)x0 + (2/5)x0.5 = 0.2

* **Humidity cho sunny**

|  |  |  |  |
| --- | --- | --- | --- |
| Humidity | Yes | No | Number of instances |
| High | 0 | 3 | 3 |
| Normal | 2 | 0 | 2 |

Gini(Outlook=Sunny and Humidity=High) = 1 – (0/3)2 – (3/3)2 = 0

Gini(Outlook=Sunny and Humidity=Normal) = 1 – (2/2)2 – (0/2)2 = 0

Gini(Outlook=Sunny and Humidity) = (3/5)x0 + (2/5)x0 = 0

* **Wind cho sunny**

|  |  |  |  |
| --- | --- | --- | --- |
| Wind | Yes | No | Number of instances |
| Weak | 1 | 2 | 3 |
| Strong | 1 | 1 | 2 |

Gini(Outlook=Sunny and Wind=Weak) = 1 – (1/3)2 – (2/3)2 = 0.266

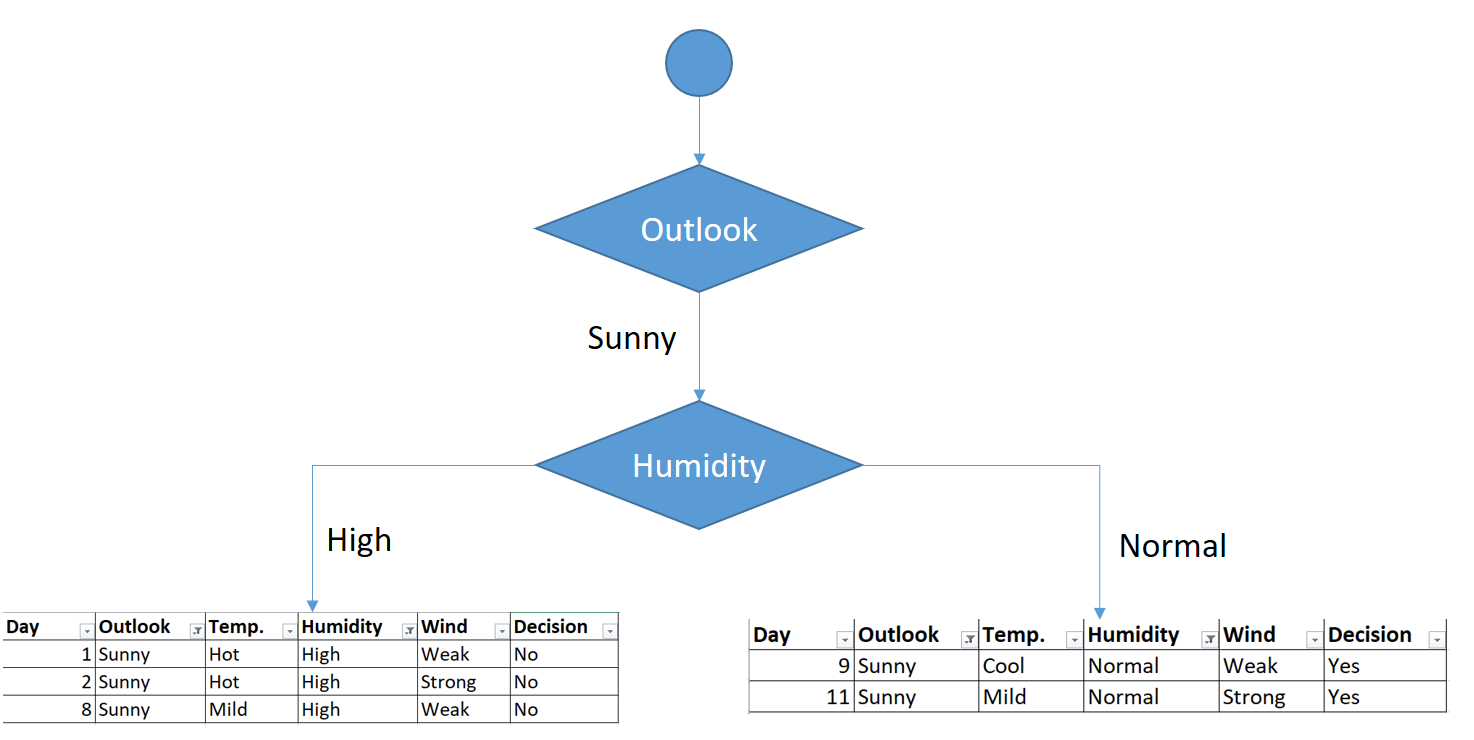
Gini(Outlook=Sunny and Wind=Strong) = 1- (1/2)2 – (1/2)2 = 0.2

Gini(Outlook=Sunny and Wind) = (3/5)x0.266 + (2/5)x0.2 = 0.466

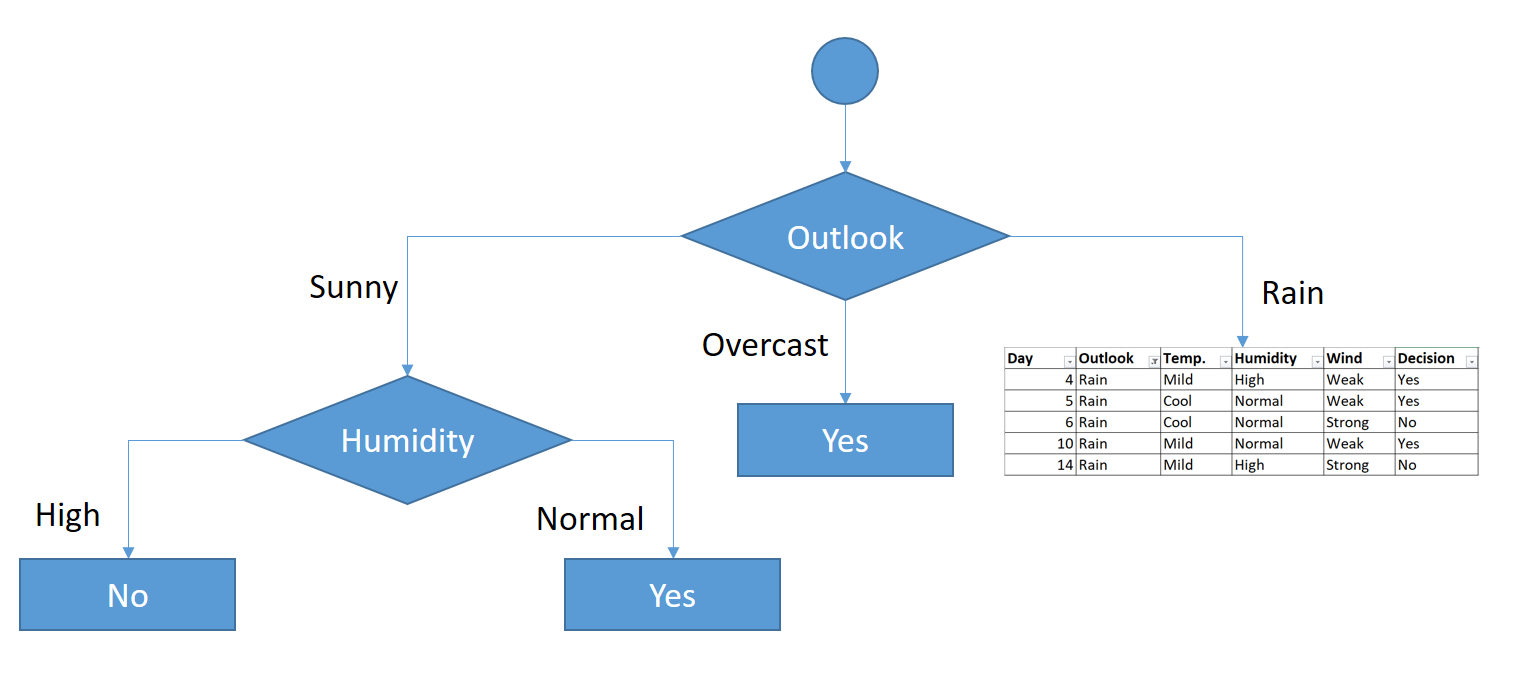
Tổng kết lại ta có bảng

|  |  |
| --- | --- |
| Feature | Gini index |
| Temperature | 0.2 |
| Humidity | 0 |
| Wind | 0.466 |

Humidity được chọn vì nó có chỉ số gini bé nhất



Kết hợp với bảng outlook ban đầu ta có



Bước 3: Rain Outlook

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Day | Outlook | Temp. | Humidity | Wind | Decision |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 6 | Rain | Cool | Normal | Strong | No |
| 10 | Rain | Mild | Normal | Weak | Yes |
| 14 | Rain | Mild | High | Strong | No |

Chúng ta sẽ tính toán chỉ số gini của temperature, wind, humidity

**Gini of temprature for rain outlook**

|  |  |  |  |
| --- | --- | --- | --- |
| Temperature | Yes | No | Number of instances |
| Cool | 1 | 1 | 2 |
| Mild | 2 | 1 | 3 |

Gini(Outlook=Rain and Temp.=Cool) = 1 – (1/2)2 – (1/2)2 = 0.5

Gini(Outlook=Rain and Temp.=Mild) = 1 – (2/3)2 – (1/3)2 = 0.444

Gini(Outlook=Rain and Temp.) = (2/5)x0.5 + (3/5)x0.444 = 0.466

**Gini of humidity for rain outlook**

|  |  |  |  |
| --- | --- | --- | --- |
| Humidity | Yes | No | Number of instances |
| High | 1 | 1 | 2 |
| Normal | 2 | 1 | 3 |

Gini(Outlook=Rain and Humidity=High) = 1 – (1/2)2 – (1/2)2 = 0.5

Gini(Outlook=Rain and Humidity=Normal) = 1 – (2/3)2 – (1/3)2 = 0.444

Gini(Outlook=Rain and Humidity) = (2/5)x0.5 + (3/5)x0.444 = 0.466

**Gini of wind for rain outlook**

|  |  |  |  |
| --- | --- | --- | --- |
| Wind | Yes | No | Number of instances |
| Weak | 3 | 0 | 3 |
| Strong | 0 | 2 | 2 |

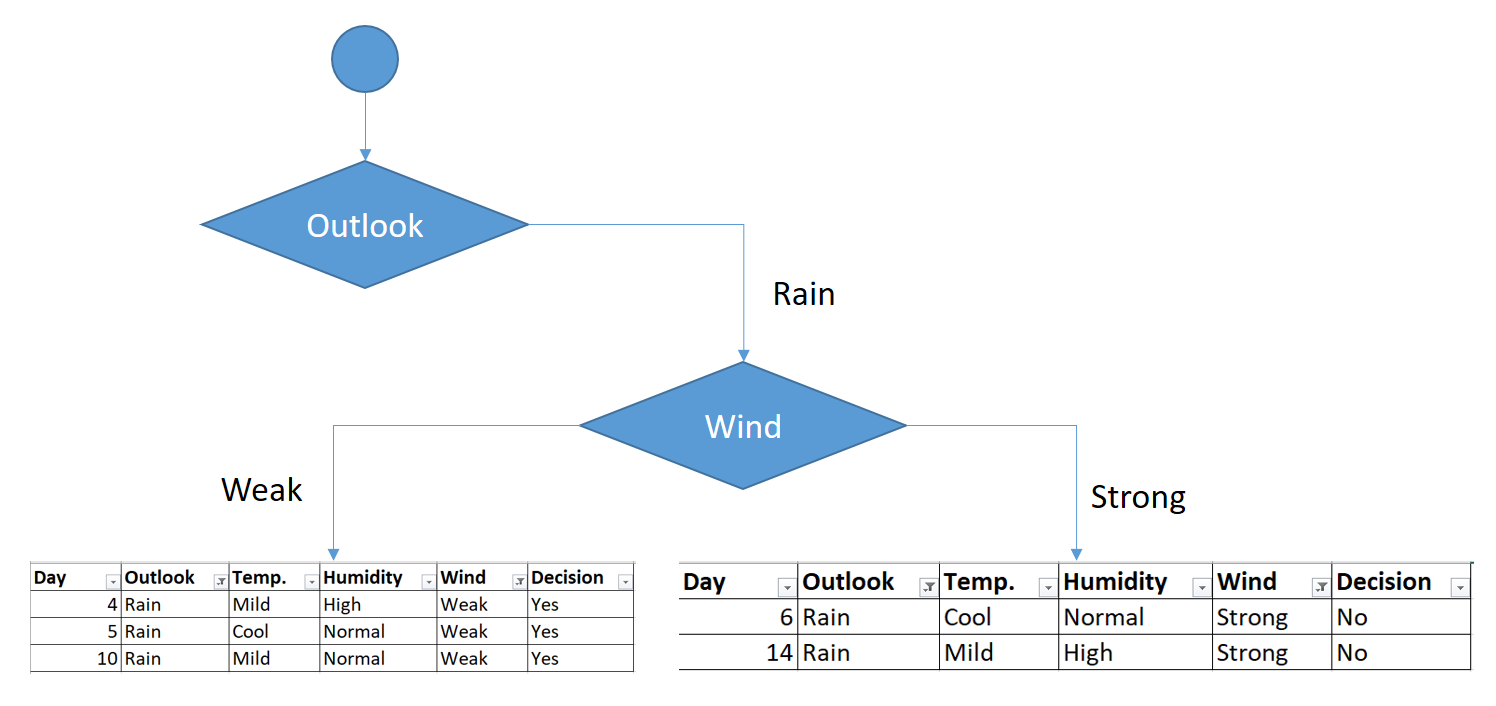
Gini(Outlook=Rain and Wind=Weak) = 1 – (3/3)2 – (0/3)2 = 0

Gini(Outlook=Rain and Wind=Strong) = 1 – (0/2)2 – (2/2)2 = 0

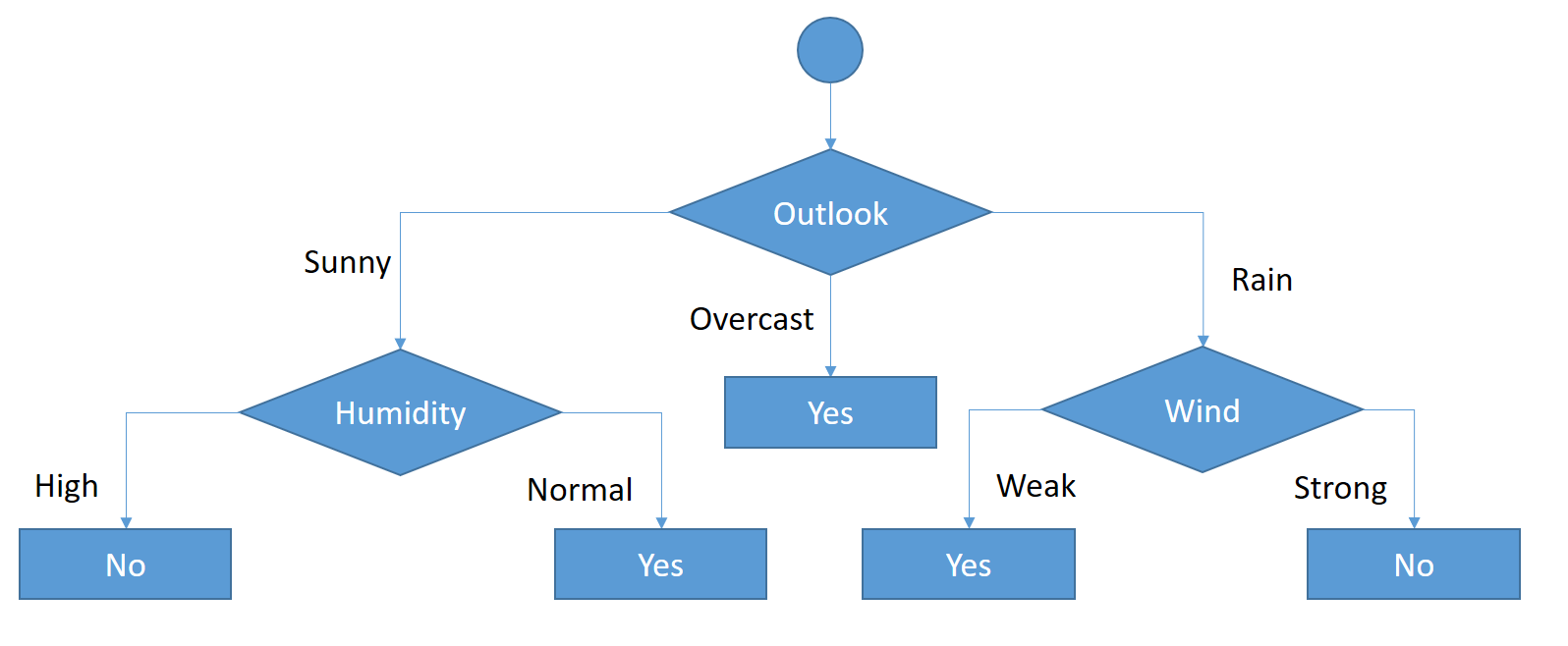
Gini(Outlook=Rain and Wind) = (3/5)x0 + (2/5)x0 = 0

Kết quả cuối cùng ta có

|  |  |
| --- | --- |
| Feature | Gini index |
| Temperature | 0.466 |
| Humidity | 0.466 |
| Wind | 0 |

[](https://sefiks.com/wp-content/uploads/2018/08/cart-step-5.png)wind mà bé thì luôn chơi còn wind mà to thì luôn nghỉ

Dưới đây là bảng tổng hợp gini index

[](https://sefiks.com/wp-content/uploads/2018/08/cart-step-6.png)

1. Ưu nhược điểm
2. Ưu điểm

* CART là không đối xứng. Do đó phương pháp này không yêu cầu cụ thể bất kỳ hình thức chức năng nào
* CART không yêu cầu các biến được chọn trước
* GIỎI rất linh động và có khả năng điều chỉnh kịp thời

1. Nhược điểm

* Có thể xuất hiện cây quyết định không ổn định

Thuật toán ID3

1. Mục đích ý nghĩa của thuật toán

- ID3 là thuật toán nổi tiếng để xây dựng cây quyết định, áp dụng cho bài toán phân loại (Classification) mà tất các các thuộc tính để ở dạng category

- Trong thuật toán ID3, các thuộc tính được đánh giá dựa trên Hàm số Entropy ( là hàm số đo sự hỗn loạn, không chắc chắc, độ bất định, độ bất ngờ, lượng thông tin hay những từ tương tự thế )

1. Một vài ứng dụng của thuật toán

- Dùng cho các bài dự đoán

1. Ý tưởng thuật toán

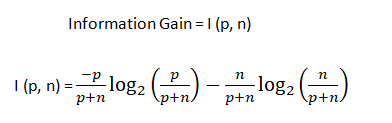
Trong thuật toán cây quyết định, vấn đề được giải quyết bằng việc biểu diễn cây.

* Mỗi nút bên trong của cây tương ứng với một thuộc tính.
* Mỗi nút lá tương ứng với một Nhãn lớp.

Trong cây quyết định để dự đoán nhãn lớp cho một bản ghi, chúng ta bắt đầu từ gốc của cây. Chúng ta so sánh giá trị của thuộc tính gốc với thuộc tính bản ghi. Chúng ta theo nhánh tương ứng với các giá trị đó và nhảy đến nút tiếp theo. Chúng ta tiếp tục so sánh giá trị thuộc tính bản ghi của chúng ta với các nút bên trong khác của cây cho đến khi chúng tôi đạt được một nút lá.

Trong cây quyết định, chúng ta sử dụng

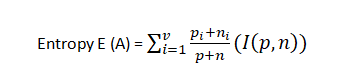
* Information Gain:



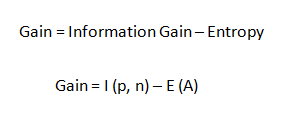
Ở đây P và n là gì? Để tìm p và n, chúng ta kiểm tra thuộc tính lớp hoặc kết quả dưới dạng nhị phân (0, 1). Vì vậy, với p, chúng ta lấy giá trị đúng là 1 (trong trường hợp nhị phân) và với n giá trị sai là 0 (giá trị nhị phân).

* Entropy

Entropy về cơ bản được sử dụng để tạo cây. Chúng ta tìm entropy từ thuộc tính hoặc lớp.



* Gain

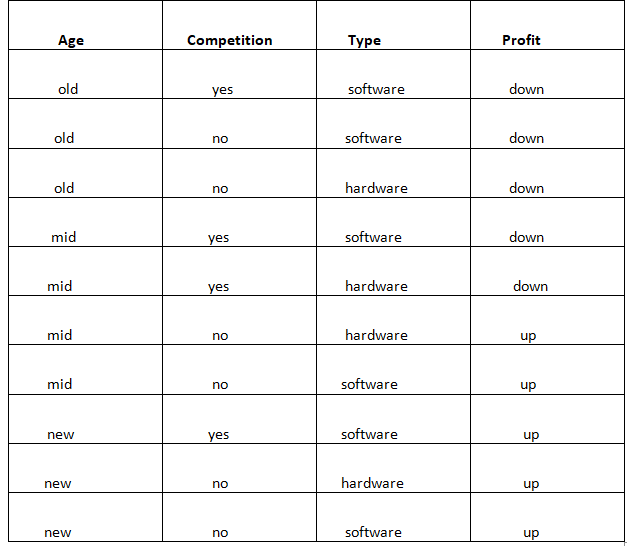


Gain về cơ bản được sử dụng để tìm từng thuộc tính của tập huấn luyện

Chúng ta sẽ sử dụng 3 công thức này để tính toán và vẽ ra cây quyết định. Để hiểu rõ hơn chúng ta sẽ đi vào phần ví dụ minh họa

1. Ví dụ minh họa

Bộ dữ liệu này chứa thông tin về công ty có lãi và lỗ. Bộ dữ liệu này có bốn thuộc tính là Tuổi, Cạnh tranh, Loại và Lợi nhuận. Lợi nhuận là thuộc tính mục tiêu của chúng tôi. Là một nhà khoa học dữ liệu, mục tiêu của chúng tôi là tạo ra cây bằng cách sử dụng phương pháp cây quyết định và đưa ra quyết định đúng đắn mà công ty nào có được lợi nhuận tối đa và cơ hội bị mất là tối thiểu. Vậy hãy bắt đầu



Trước hết tôi tìm thấy thông tin đạt được từ thuộc tính Target lên xuống

P: lên = 5

N: xuống = 5

* Information Gain của profit

Theo công thức ta có: I(5,5) = 1

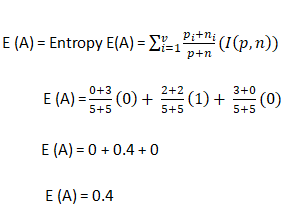
Vì vậy, tôi tính toán Information Gain của tập dữ liệu bằng 1.

Sau khi tìm được Information Gain, chúng ta sẽ tính toán Entropy của từng thuộc tính

|  |  |  |  |
| --- | --- | --- | --- |
| Age | Pi | Ni | I(Pi, Ni ) |
| Old | 0 | 3 | I(0,3) = 0 |
| Mid | 2 | 2 | I(2,2) = 1 |
| New | 3 | 0 | I(3,0) = 1 |
|  |  |  |  |

Trong thuộc tính age, chúng ta có dữ liệu phân loại cây old, mid, new và tìm Information Gain của từng loại như bảng trên.

Sau khi tìm thấy thông tin của từng (old, mid, new), hãy đặt giá trị vào bên dưới phương trình entropy để tìm entropy của tuổi.



Vậy entropy của age là 0,4

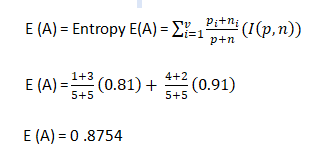
Sau khi tìm thấy entropy chúng ta có thể dễ dàng tính được gain.

Gain(age) = Information gain – Entropy(age) = 1 – 0.4 = 0.6

Tiếp theo là competition

|  |  |  |  |
| --- | --- | --- | --- |
| competition | Pi | Ni | I(Pi, Ni ) |
| Yes | 1 | 3 | I(1,3) = 0.81 |
| No | 4 | 2 | I(4,2) = 0.91 |
|  |  |  |  |

Entropy của competition



Vậy entropy = 0.8754

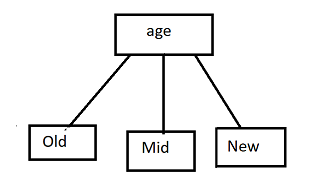
Ta có Gain = 1- 0.8754 = 0.1245.

Tính tương tự như 2 thuộc tính trên ta có entropy = 1 và gain = 0

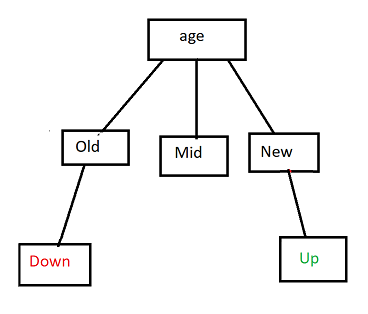
Bây giờ chúng ta có

* Gain của age là 0.6
* Gain của competition là 0.12
* Gain của type là 0

Vậy Root node sẽ là age

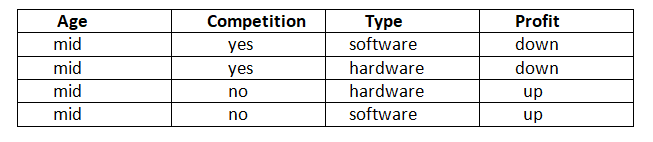


Cả 3 trường hợp công ty có age là old thì đều có profit là down và các công ty có age là new đều có profit là up nên ta có



Với mid có 4 trường hợp là 2 down và 2 up.

Bây giờ chúng ta sẽ thu nhỏ data lại chỉ tính toán trong khung 4 trường hợp này thôi để tính competition và type xem ai sẽ là con của node mid này



Information gain = I(2,2) = 1

Sau khi có được information gain chúng ta lại tiếp tục tính toán entropy và gain của competion.

Tính như cũ ta có

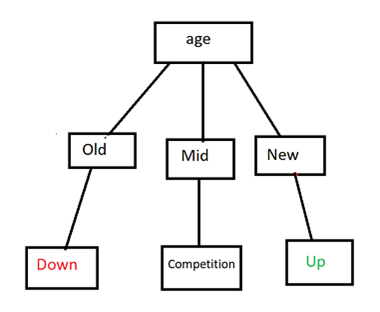
Entropy của competion = 0

Gain của competition = 1

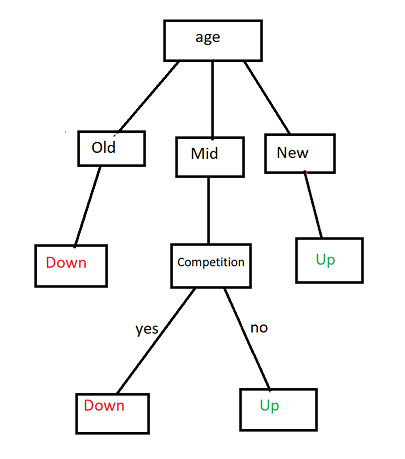
Entropy của type = 1

Gain của type = 0

Bây giờ chúng ta chọn được competition sẽ là con của mid



Với competition là yes profit là down còn với no profit là up nên ta có



Vậy là cây của chúng ta đã hoàn thành, chúng ta sử dụng Gain từ tất cả các thuộc tính và tạo ra cây. Theo cây quyết định, chúng ta chỉ có thể đưa ra những quyết định mà công ty cần có để đạt được lợi nhuận tối đa

1: Vì vậy, lợi nhuận sẽ tăng nếu tuổi là new.

Thứ 2: lợi nhuận sẽ tăng nếu tuổi là mid và không có cạnh tranh.

Vì vậy, chúng ta sử dụng thông tin này để mang lại lợi nhuận cho công ty.

1. Ưu nhược điểm

Ưu điểm

* Tạo ra tree nhanh nhất
* Tạo ra 1 cây lùn
* Quét qua cả database
* Độ chính xác hợp lí

Nhược điểm

* Cây quyết định không hoạt động tốt nếu bạn có ranh giới trơn tru. tức là chúng hoạt động tốt nhất khi bạn có mô hình hằng số khôn ngoan không liên tục. Nếu bạn thực sự có một cây quyết định chức năng tuyến tính, cây không phải là tốt nhất.
* Cây quyết định không hoạt động tốt nhất nếu bạn có nhiều biến không tương quan. Công việc của cây quyết định là bằng cách tìm các tương tác giữa các biến. nếu bạn có một tình huống trong đó không có tương tác giữa các biến tiếp cận tuyến tính có thể là tốt nhất.
* Phân mảnh dữ liệu: Mỗi phân chia trong một cây dẫn đến một bộ dữ liệu giảm đang được xem xét. Và, do đó mô hình được tạo ra ở phần tách sẽ có khả năng giới thiệu sai lệch.
* Phương sai cao và không ổn định: Kết quả của chiến lược tham lam được áp dụng bởi phương sai của cây quyết định trong việc tìm điểm xuất phát đúng của cây có thể ảnh hưởng lớn đến kết quả cuối cùng. tức là những thay đổi nhỏ sớm có thể có tác động lớn về sau. Vì vậy, nếu bạn lấy hai mẫu khác nhau từ vũ trụ của mình, điểm bắt đầu của cả hai mẫu có thể rất khác nhau (và thậm chí có thể là các biến khác nhau), điều này có thể dẫn đến kết quả hoàn toàn khác nhau.