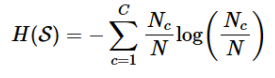
# **Decision tree** (cây quyết định)

1. **Khái niệm cây quyết định:**
   * Là mô hình học có giám sát
   * Cấu trúc cây như biểu đồ luồng
   * Mỗi nút trong thể hiện 1 sự kiểm tra trên 1 thuộc tính
   * Mỗi nhánh đại diện cho 1 kq của sự kiểm tra
   * Các nút lá đại diện cho các nhãn lớp hoặc phân phối lớp
2. **Đầu vào:**
3. **Đầu ra:**
4. **Phương thức:**

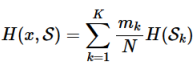
**B1:** tính entropy trên toàn bộ tập S

* + H(S) : entropy trên toàn bộ tập dữ liệu S
  + S: tập dữ liệu huấn luyện(gồm cả x và y)
  + N: tổng số mẫu dữ liệu huấn luyện
  + Nc: số các mẫu dữ liệu trong tập S đc gán nhãn c
  + C: tổng số nhãn dữ liệu khác nhau trong tập S
  + c: chỉ số của nhãn dữ liệu



**B2:** Tính entropy trên các mẫu dữ liệu trong tập S và có thuộc tính là x

* + x: là thuộc tính đc chọn
  + K: số giá trị của thuộc tính đó
  + mk: số mẫu có thuộc tính x và có giá trị là k
  + Sk: tập các mẫu sao cho thuộc tính x có giá trị là k
  + H(Sk): entropy trên tập SK



**B3:** tính *information gain* trên thuộc tính x



(G(x,S) càng lớn càng tốt -> H(x,S) càng nhỏ càng tốt)



**B4:** Chọn đc thuộc tính tốt nhất ở bc thứ i(thuộc tính tốt nhất là thuộc tính có H(x,S) nhỏ nhất)

**B5:** Chia dữ liệu vào các nút con tương ứng với các giá trị của thuộc tính:

* + - TH1: nếu tất cả các mẫu trong node con thuộc cùng 1 lớp C (có cùng nhãn C) thì node đó đc gán nhãn C

-> Dừng thuật toán

* + - TH2: nếu node con là rỗng (không có dữ liệu) thì node đó sẽ đc gán = nhãn phổ biến nhất trong tập S

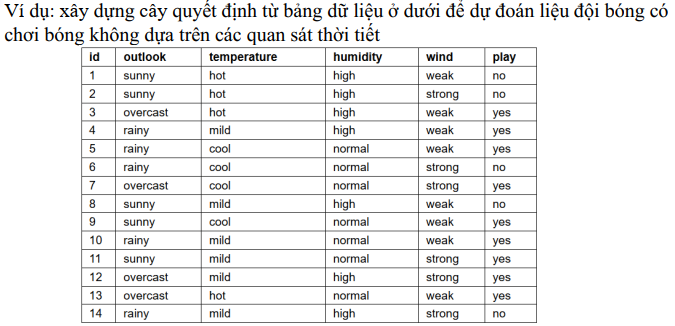
-> Dừng thuật toán

* + - TH3: nếu không còn thuộc tính nào để phân chia (các thuộc tính đều đc xét hết) -> node lá đó sẽ đc gán = nhãn phổ biến nhất

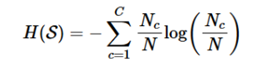
->Dừng thuật toán

* + - TH4: th còn lại quay lại bc 2, bc 3(xét tất cả các thuộc tính còn lại trừ thuộc tính đã đc xác định trên node cha) để tìm ra thuộc tính tốt nhất trên tập dữ liệu mới.

## 5. VD:



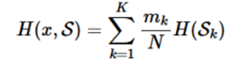
B1: tính entropy trên tập S: H(S) = ?



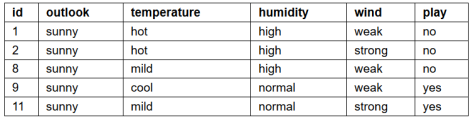
1. tổng số mẫu dữ trong tập huấn luyện S là: N = 14
2. số nhãn dữ liệu (khác nhau) trong tập S: C = 2 (no, yes)
3. chỉ sô của các nhãn: c = {1,2} (với c=1 là nhãn “no”, c=2 là nhãn “yes”)
4. Tổng số mẫu dữ liệu trong tập S có chỉ số nhãn là c{1,2}: Nc
   * c = 1 -> giá trị của nhãn là “no”: N1=5
   * c=2 -> giá trị của nhãn là “yes”: N2 = 9

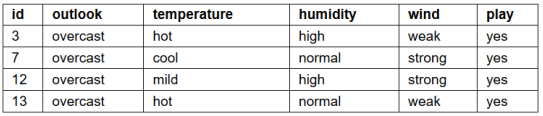
H(S) =

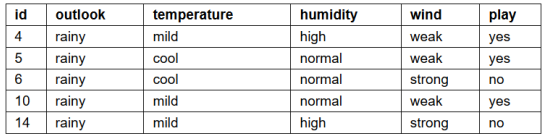
B2: Tính entropy của từng thuộc tính trong tập S



1. Thuộc tính outlook:



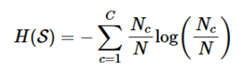




* + x = outlook
  + K = 3: tổng số giá trị của outlook(sunny, overcast, rainy)
  + mk: số mẫu có thuộc tính là outlook và có giá trị là k
    - k=1 <=> sunny: số mẫu có thuộc tính là outlook và giá trị sunny là: m1 = 5
    - k=2 <=> overcast: số mẫu có thuộc tính là outlook và giá trị overcast là: m2 = 4
    - k = 3 <=> rainy: số mẫu có thuộc tính là outlook và giá trị rainy là:

m3 = 5

* + tính H(Sk):

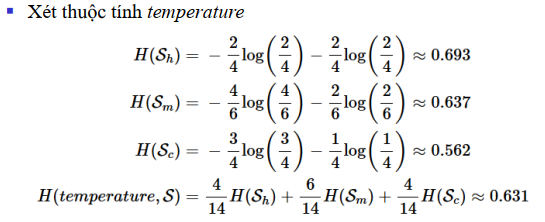


* + - H(S1) = H(Ssunny) =
    - H(S2) = H(Sovercast) =
    - H(S3) = H(Srainy) =
  + Entropy của thuộc tính outlook:

H(outlook, S) =

H(outlook, S) =

1. Thuộc tính Temperature



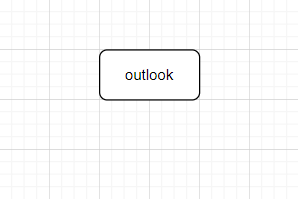
1. Thuộc tính humidity:



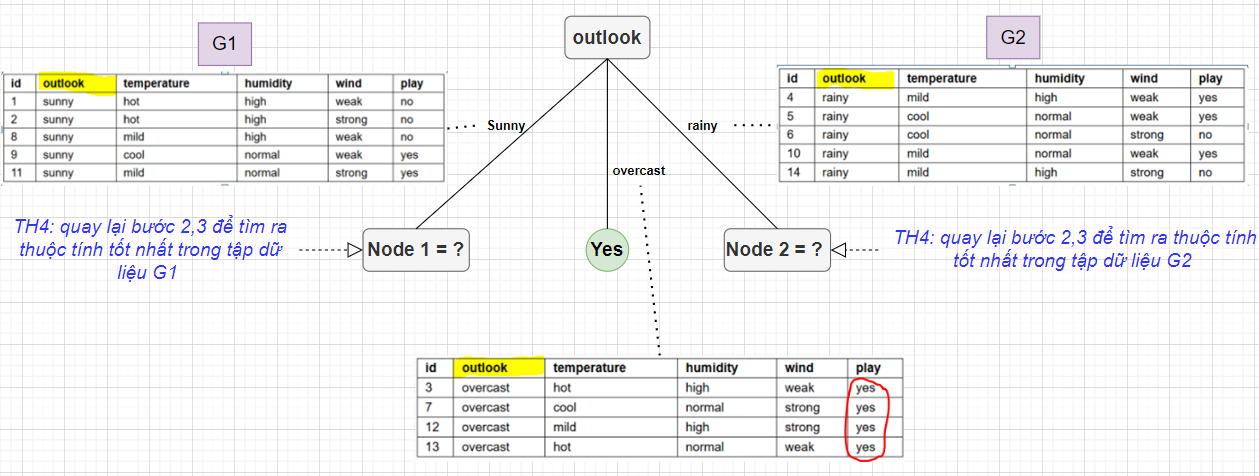
1. Thuộc tính wind:



B3,4: thuộc tính cần chọn là outlook vì H(outlook,S) đạt giá trị nhỏ nhất(information gain là lớn nhất)

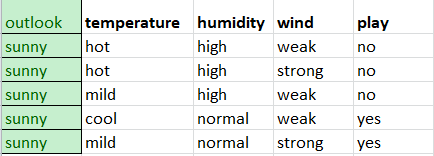


B5: Chia dữ liệu vào các nút con tương ứng với các giá trị của thuộc tính:

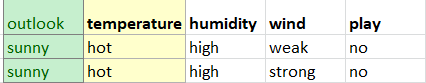


B6: (quay lại b2,3) xét trên tập dữ liệu:

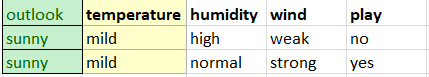
S =



* H(S) =
* Thuộc tính Temperature:



* + H(S1) = H(Shot) =



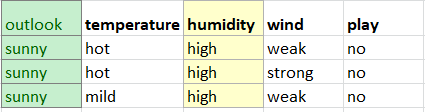
* + H(S2) = H(Smild) =



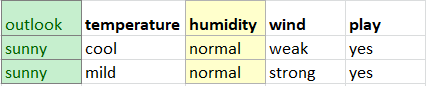
* + H(S3) = H(Scool) =
* H(temperature, S) = =

=

* Thuộc tính humidity



* + H(S1) = H(Shigh) =

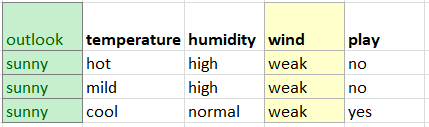


* + H(S2) = H(Snormal) =

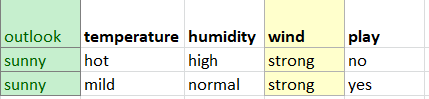
🡺H(humidity, S) = =

=

* Thuộc tính wind:



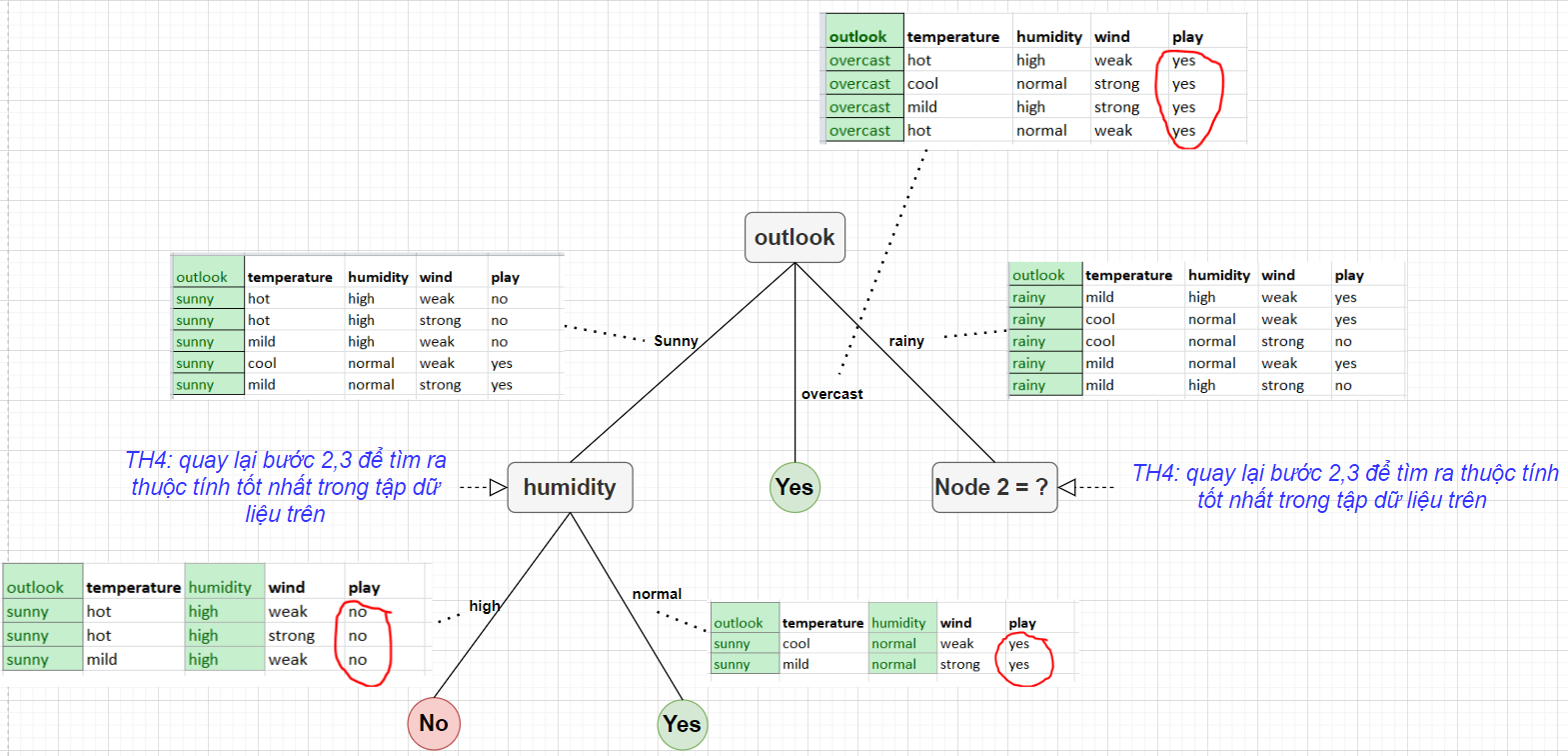
H(S1) = H(Sweak) =



H(S2) = H(Sstrong) =

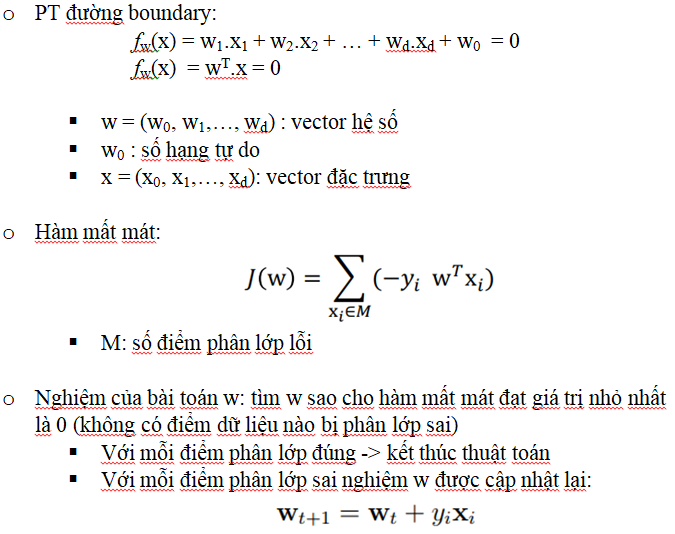
🡺H(wind,S) =

* Vậy thuộc tính cần chọn là humidity vì có H(huminity,S) = 0 là nhỏ nhất
* Chia dữ liệu vào các node con tương ứng với giá trị của thuộc tính:



1. Perceptron

– tìm w để hàm mất mát là nhỏ nhất:



* Chốt tìm w để min(||m||2 2)

|  |  |
| --- | --- |
|  |  |
| - Tìm ra siêu phẳng phân lớp 2 tập dữ liệu  - Tìm ra 1 đg tuyến tính | Tìm ra siêu phẳng có:  + margin của 2 tập dữ liệu là = nhau và lớn nhất(siêu phẳng cách đều 2 tập dữ liệu),  - *có nghiệm duy nhất(chỉ tìm ra đc siêu phẳng thỏa mãn cả 2 đk)*  - Tìm ra nhiều đg tuyến tính -> chọn đg tốt nhất |

* Giá trị của thuộc tính Kernel:

Nếu = Linear: sp tuyến tính

Nếu poLy: là mặt cong

Nếu rbf: tính theo công thức …

Nếu là xíc ma: thì sd hàm …

Logistic

* Input :
* Outout: là giá trị xác xuất (số thực trong khoảng [0, 1])