AI VIET NAM – COURSE 2024
Decision Tree

Trung-Trực Trần

Ngày 17 tháng 2 năm 2024

1 Giới thiệu.

Cây quyết định (Decision Tree) là một trong những thuật toán máy học phổ biến nhất. Nó là một công cụ mạnh mẽ được sử dụng cho việc phân loại và dự đoán trong học máy và khai phá dữ liệu.

Cây quyết định thực hiện quá trình học bằng cách phân chia tập dữ liệu thành các phần nhỏ hơn và xây dựng một cây quyết định dựa trên các quy tắc. Mỗi nút trên cây biểu diễn một thuộc tính (hoặc biến độc lập), mỗi cạnh đi từ nút cha đến các nút con biểu diễn các quy tắc quyết định, và mỗi lá trên cây biểu diễn một lớp hoặc một giá trị dự đoán.

2 Công thức cây quyết định

2.1 Gini Impurity

Công thức Gini Impurity được sử dụng trong thuật toán cây quyết định để đo lường độ không chính xác của một dự đoán khi phân loại một tập dữ liệu.

cần lưu ý là Gini Impurity càng nhỏ (gần 0) thì tập dữ liệu đó càng "thuần khiết", nghĩa là các mẫu trong cùng một nhóm có xu hướng thuộc vào cùng một lớp. Ngược lại, nếu Gini Impurity cao (gần 1), thì việc phân loại các mẫu trong nhóm đó trở nên không chắc chắn.

Giả sử bạn đang xem xét một tập dữ liệu chia thành K nhóm, mỗi nhóm chứa một phần tỷ lệ p_i với i=1,2,...,K.

Công thức được biểu diễn như sau:

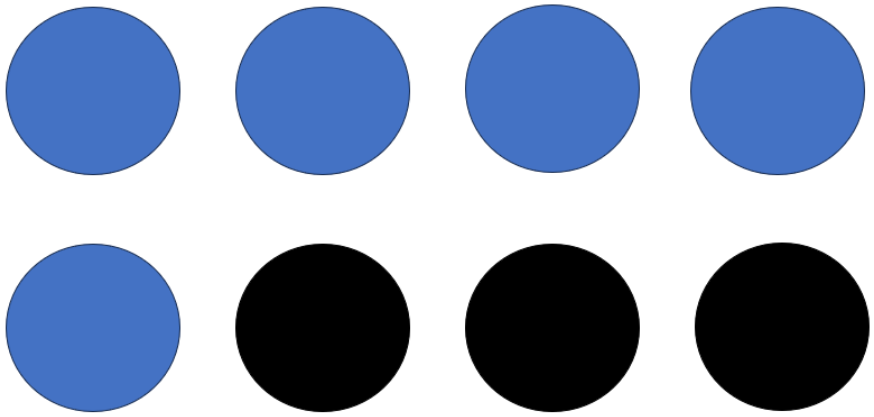
$$I_G = 1 - \sum_{i=1}^{K} p_i^2 \tag{1}$$

Trong đó:

- I_G là Gini Impurity.
- p_i là tỷ lệ các mẫu thuộc vào lớp i.

Khi xây dựng cây quyết định, chúng ta cần chọn thuộc tính và giá trị phân chia sao cho Gini Impurity sau phân chia là nhỏ nhất, tức là mức độ "thuần khiết" cho dữ liệu thuộc về từng nhóm con được tạo ra.

Đối với tệp dữ liệu gồm 2 labels (i=2) thì chỉ số I_G sẽ đạt ngưỡng lớn nhất là bằng 0.5 (Dữ liệu sẽ bị nhiễu lớn nhất).



Hình 1: Ví dụ 1 với 2 label xanh và đen

Với ví dụ ở hình 1, ta có 2 label là xanh và đen. Với 5 viên bi thuộc xanh và 3 viên bi thuộc đen ta có thể tính được xác suất suất hiện của lần lượt 2 tập là $\frac{5}{8}$ và $\frac{3}{8}$. Áp dụng công thức Gini Impurity ta sẽ tính được:

$$I_G = 1 - \frac{5}{8}^2 - \frac{3}{8}^2 = 0.46875.$$

Vì ví dụ chỉ có 2 label là xanh và đen nên kết quả 0.46 có thể coi là tính không thuần khiết tiệm cận mức cao nhất (Max=0.5).

2.2 Entropy

Entropy trong cây quyết định là một khái niệm được sử dụng để đo lường sự không chắc chắn trong dữ liệu (Trung bình surprise). Trong ngữ cảnh của cây quyết định, entropy thường được sử dụng để đo lường mức độ không chắc chắn của phân phối lớp trong tập dữ liệu.

Entropy được tính bằng công thức sau:

$$Entropy(S) = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{2}$$

Trong đó:

- S là tập dữ liệu.
- c là số lớp trong tập dữ liệu.

- p_i là tỷ lệ của lớp i trong tập dữ liệu.

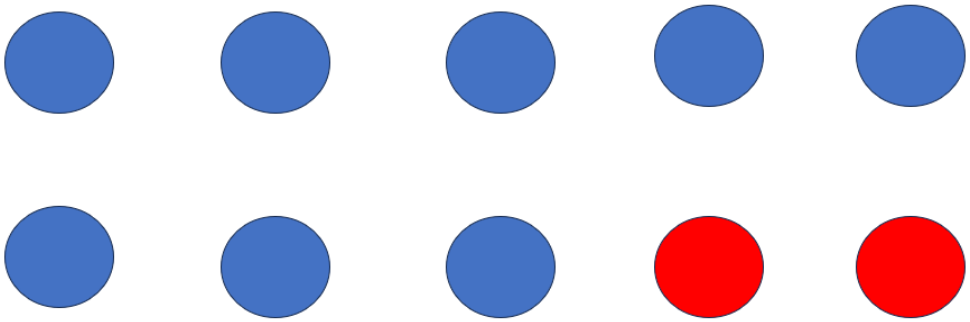
Entropy càng cao khi tỷ lệ của các lớp trong tập dữ liệu gần bằng nhau, và càng thấp khi một lớp chiếm đa số.

Khi xây dựng cây quyết định, chúng ta cố gắng chia tập dữ liệu sao cho entropy sau khi chia là thấp nhất có thể. Điều này giúp cây quyết định có thể học được các quy tắc quyết định hiệu quả từ dữ liệu.

Quyết định về cách chia tập dữ liệu dựa trên entropy thường được thực hiện bằng cách so sánh entropy trước và sau khi chia, và chọn cách chia mà giảm entropy nhiều nhất.

2.2.1 Diễn giải công thức Entropy

Để hiểu rõ hơn về công thức Entropy chúng ta hãy cùng sơ lược lại cái yếu tố chính và cách công thức hình thành.



Hình 2: Ví dụ 2 gồm 10 viên bi xanh và đỏ

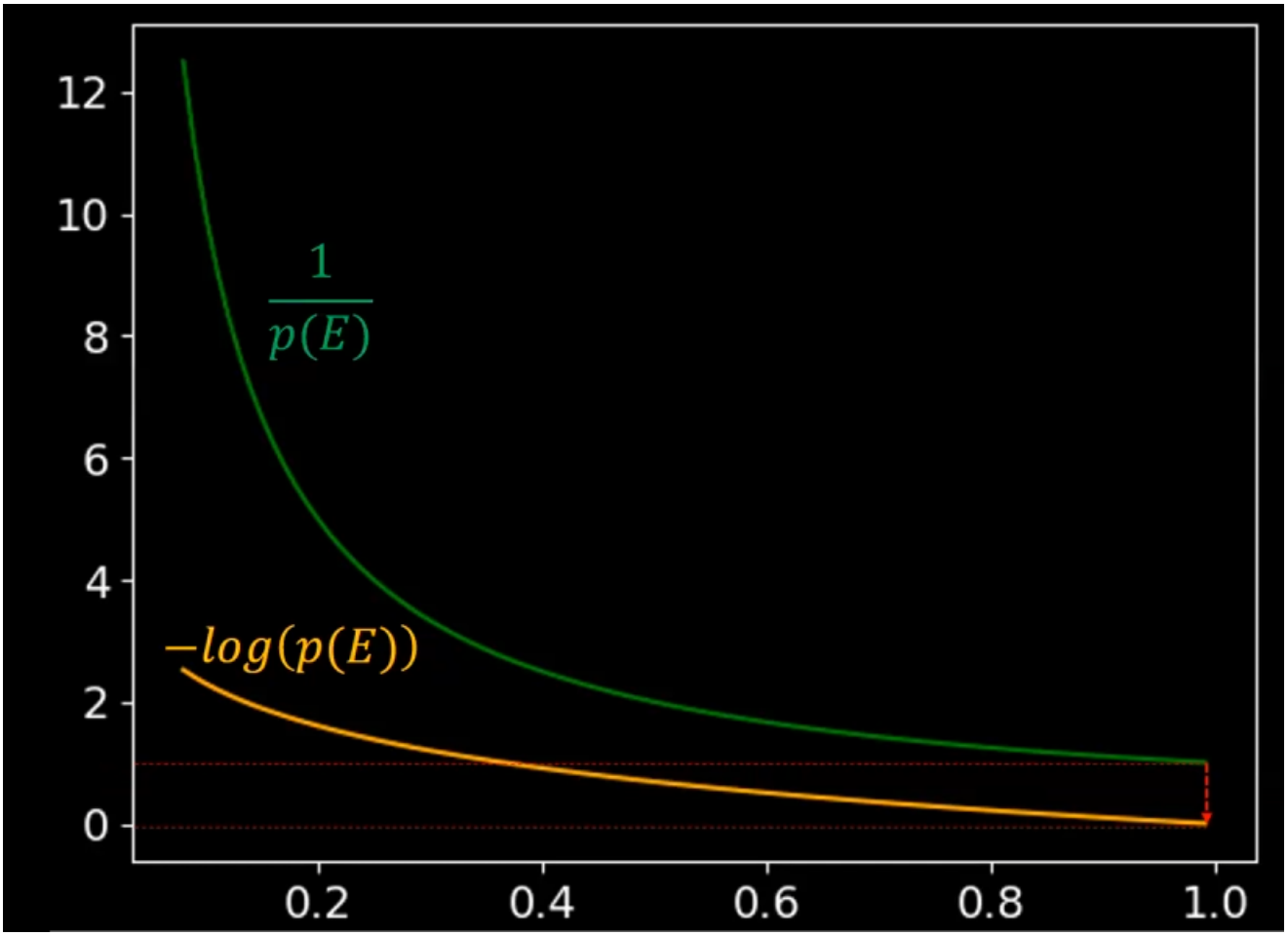
Với ví dụ ở hình 2 chúng ta có thể thấy đây là một tập dataset gồm 10 viên bi và có 2 label (xanh-đỏ). Với 8 viên bi là xanh và 2 viên bi là đỏ ta có thể tính xác suất của từng trường hợp là P(A)=$\frac{8}{10}$=$\frac{4}{5}$ và P(B)$\frac{2}{10}$=$\frac{1}{5}$

Từ xác suất trên chúng ta có thể tính được độ ngạc nhiên khi xảy ra trường hợp đó surprise=$\frac{1}{P}$. Vậy độ ngạc nhiên của trường hợp bi xanh và bi đỏ là surprise(A)=$\frac{1}{\frac{4}{5}}$=1.25 và surprise(B)=$\frac{1}{\frac{1}{5}}$=5.

Công thức surprise:

$$surprise = \frac{1}{P(E)} \tag{3}$$

- Với P(E) là xác suất xảy ra sự kiện E.



Hình 3: Minh hoạ đồ thị tương quan giữa surprise và log

Như ta có thể thấy ở hình 3, với surprise=$\frac{1}{P(E)}$ thì surprise $\in$ [1,∞]. Nếu xác suất là 0% thì mức độ ngạc nhiên sẽ là dương vô cùng và với xác suất là 100% thì kết quả lại trả về mức độ ngạc nhiên là 1. Điều này khác vô lý ở vế 2 vì với xác suất 100% thì mức độ ngạc nhiên nên trả về 0 (Không bất ngờ).

Vì vậy nên chúng ta sử dụng hàm Log để có thể đổi cận của surprise $\in$ [1,∞] thành surprise $\in$ [0,∞].

Công thức hàm log khi áp dụng vào surprise:

$$log(\frac{1}{P(E)}) = -log(P(E)) \tag{4}$$

Điều này đã giúp cho công thức trở nên hợp lý hơn khi xác suất xảy ra bằng 100% thì mức độ ngạc nhiên sẽ trả về 0.

Từ các suy luận trên ta có công thức trung bình của mức độ ngạc nhiên:

$$Entropy = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{5}$$

2.3 Ví dụ thực tế về Decision Tree dùng Entropy (Classify)

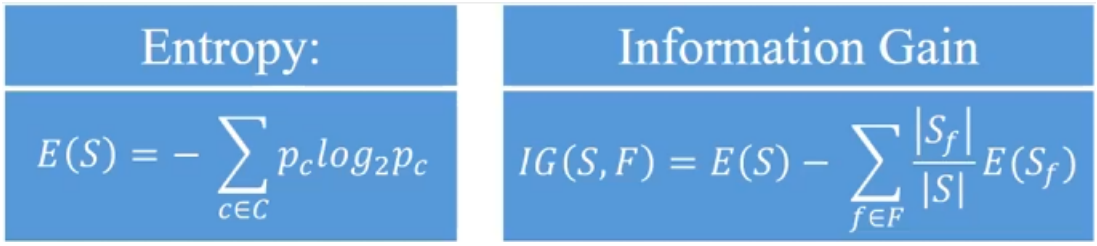
2.3.1 Data có biến rời rạc

Ta có một dataset về chơi tenis, với 2 **labels** là yes (có chơi tenis) và no(không chơi tenis) với 14 samples và các cột (outlook, temperature, humidity, wind) là các **Features**.

Bảng 1: Dữ liệu

STT	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rainy	Mild	High	Weak	Yes
5	Rainy	Cool	Normal	Weak	Yes
6	Rainy	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rainy	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rainy	Mild	High	Strong	No

Chúng ta cần tính Entropy của các Features và tính information gain theo công thức sau.



Hình 4: Công thức tính entropy và gain

Nếu entropy nhánh giảm càng nhiều chúng ta sẽ nhận được Gain càng lớn nên ta ưu tiên chọn Features có Gain lớn nhất.

Với data có [9 yes, 5 no]: Entropy(data)= $-\frac{5}{14}.log_2(\frac{5}{14}) - \frac{9}{14}.log_2(\frac{9}{14}) = 0.94$.

Tính gain cho feature Outlook:

- Entropy(Sunny)= $-\frac{2}{5}.log_2(\frac{2}{5}) - \frac{3}{5}.log_2(\frac{3}{5}) = 0.971$.
- Entropy(Rainy)= $-\frac{3}{5}.log_2(\frac{3}{5}) - \frac{2}{5}.log_2(\frac{2}{5}) = 0.971$.
- Entropy(Overcast)= $-\frac{4}{4}.log_2(\frac{4}{4}) - \frac{0}{4}.log_2(\frac{0}{4}) = 0$.

- Gain(Outlook)= $0.94 - 0.971.\frac{5}{14} - 0.971.\frac{5}{14} - 0 = 0.246$.

Tính gain cho feature Temperature:

- Entropy(Hot)= $-\frac{2}{4}.log_2(\frac{2}{4}) - \frac{2}{4}.log_2(\frac{2}{4}) = 1$.
- Entropy(Mild)= $-\frac{4}{6}.log_2(\frac{4}{6}) - \frac{2}{6}.log_2(\frac{2}{6}) = 0.918$.
- Entropy(Cool)= $-\frac{3}{4}.log_2(\frac{3}{4}) - \frac{1}{4}.log_2(\frac{1}{4}) = 0.811$.
- Gain(Outlook)= $0.94 - 1.\frac{4}{14} - 0.918.\frac{6}{14} - 0.811.\frac{4}{14} = 0.029$.

Tính gain cho feature Humidity:

- Entropy(High)= $-\frac{3}{7}.log_2(\frac{3}{7}) - \frac{4}{7}.log_2(\frac{4}{7}) = 0.985$.
- Entropy(Normal)= $-\frac{6}{7}.log_2(\frac{6}{7}) - \frac{1}{7}.log_2(\frac{1}{7}) = 0.592$.
- Gain(Humidity)= $0.94 - 0.985.\frac{7}{14} - 0.592.\frac{7}{14} = 0.151$.

Tính gain cho feature Wind:

- Entropy(Weak)= $-\frac{6}{8}.log_2(\frac{6}{8}) - \frac{2}{8}.log_2(\frac{2}{6}) = 0.811$.
- Entropy(Strong)= $-\frac{3}{6}.log_2(\frac{3}{6}) - \frac{3}{6}.log_2(\frac{3}{6}) = 1$.
- Gain(Wind)= $0.94 - 0.811.\frac{8}{14} - 1.\frac{6}{14} = 0.048$.

Sau khi tính Gain, ta chọn Outlook làm gốc vì Outlook có chỉ số Gain lớn nhất (Nghĩa là entropy giảm nhiều nhất) ta được sơ đồ như sau:

Hình 5: Nút gốc của cây

ta có thể thấy với Outlook(Overcast) thì tất cả samples đều là yes thì Entropy=0 ta có thể trả về yes cho Overcast.

Đối với 2 thuộc tính Sunny [2 yes, 3 no], Rainy [3 yes, 2 no] ta phải tính thêm các điều kiện của các features. Bằng cách tính gain lần lượt của Temperature, Humidity và Wind. Nhưng với điều kiện Outlook phải là Sunny hoặc Rainy. Từ dữ liệu trên ta có được dataset ở bảng 2.

Bảng 2: Dữ liệu thu gọn

STT	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Rainy	Mild	High	Weak	Yes
4	Rainy	Cool	Normal	Weak	Yes
5	Rainy	Cool	Normal	Strong	No
6	Sunny	Mild	High	Weak	No
7	Sunny	Cool	Normal	Weak	Yes
8	Rainy	Mild	Normal	Weak	Yes
9	Sunny	Mild	Normal	Strong	Yes
10	Rainy	Mild	High	Strong	No

Bây giờ chúng ta lặp lại các bước tính gain của các Features với điều kiện là Outlook=Sunny và Outlook=Rainy.

Đối với Outlook là Sunny

Bảng 3: Dữ liệu Outlook(Sunny)

STT	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Sunny	Mild	High	Weak	No
4	Sunny	Cool	Normal	Weak	Yes
5	Sunny	Mild	Normal	Strong	Yes

Ở đây Entropy tổng sẽ là Entropy của Sunny:

$$\text{Entropy(Sunny)}= -\tfrac{2}{5}.log_2(\tfrac{2}{5}) - \tfrac{3}{5}.log_2(\tfrac{3}{5}) = 0.971.$$

Tính Gain cho Temperature

- Entropy(Cool)= 0.
- Entropy(Mild)= 1.
- Entropy(Hot)= 0.

- Gain(Temperature)= 0.571.

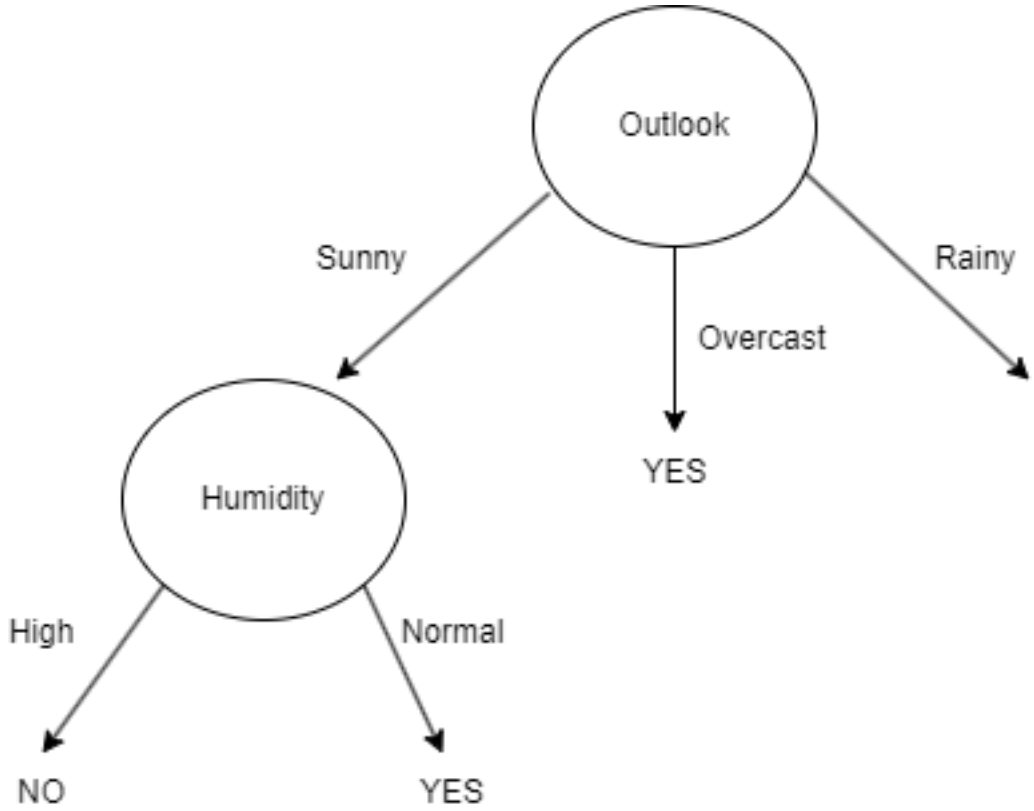
Tính Gain cho Humidity

- Entropy(High)= 0.
- Entropy(Normal)= 0.
- Gain(Humidity)= 0.971.

Tính Gain cho Wind

- Entropy(Weak)= 0.918.
- Entropy(Strong)= 1.
- Gain(Wind)= 0.1977.

Sau khi tính Gain, ta chọn Humidity làm gốc vì Humidity có chỉ số Gain lớn nhất (Nghĩa là entropy giảm nhiều nhất).



Hình 6: Tree

ta có thể thấy với Humidity(High) thì tất cả samples đều là no thì Entropy=0 ta có thể trả về no cho High. Tương tự với Normal sẽ trả về yes.

Bảng 4: Dữ liệu Outlook(Rainy)

STT	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Rainy	Mild	High	Weak	Yes
2	Rainy	Cool	Normal	Weak	Yes
3	Rainy	Cool	Normal	Strong	No
4	Rainy	Mild	Normal	Weak	Yes
5	Rainy	Mild	High	Strong	No

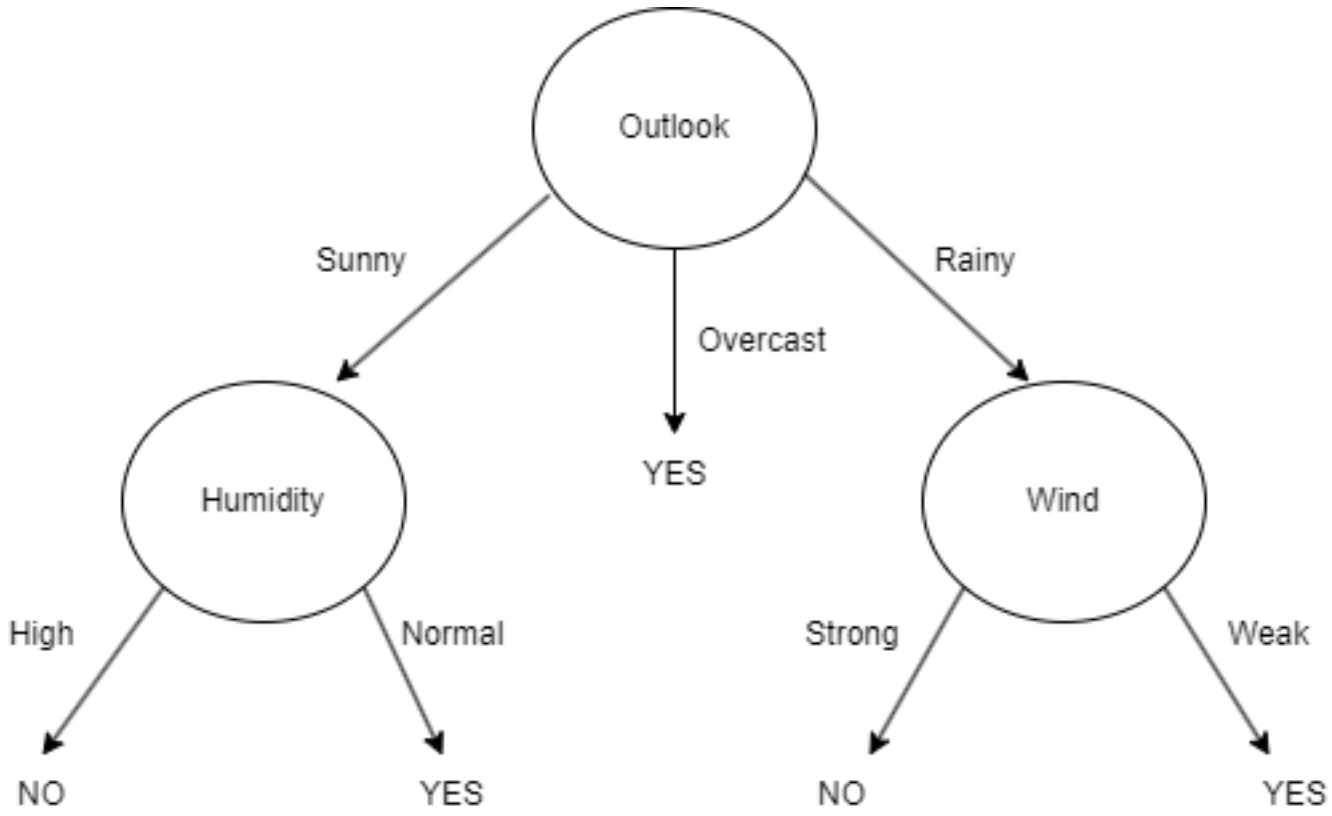
Đối với Outlook là Rainy

Ở đây Entropy tổng sẽ là Entropy của Rainy:

$$\text{Entropy(Rainy)} = -\frac{3}{5}.log_2(\frac{3}{5}) - \frac{2}{5}.log_2(\frac{2}{5}) = 0.971.$$

Tương tự như trên, chúng ta tính được gain lần lượt cho Temperature, Humidity và Wind là: 0.0202, 0.0202, 0.971

Sau khi tính Gain, ta chọn Wind làm gốc vì Wind có chỉ số Gain lớn nhất (Nghĩa là entropy giảm nhiều nhất).



Hình 7: Final Tree

ta có thể thấy với Wind(Strong) thì tất cả samples đều là no thì Entropy=0 ta có thể trả về no cho Strong. Tương tự với Weak sẽ trả về yes.

2.3.2 Data có biến liên tục

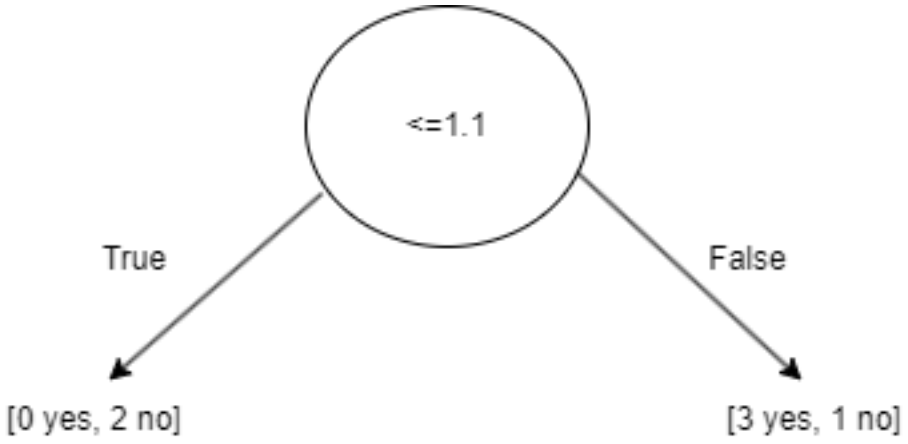
Ta có 1 tệp data với các biến là độ dài của cách hoa với 2 label là 0 và 1.

Bảng 5: Dữ liệu liên tục

STT	Petal_Length	Label
1	1	0
2	1.3	0
3	0.9	0
4	1.7	1
5	1.8	1
6	1.2	1

Đầu tiên chúng ta phải sắp xếp dữ liệu của tệp data trên từ bé đến lớn hoặc từ lớn đến bé, từ đó chúng ta có 1 dataset mới như sau. Sau đó chúng ta tiếp tục tính trung bình cộng của từng đôi.

Ta tính gain của các thuộc tính Trung bình để tìm gain lớn nhất, với cách tính là xét các giá trị bé hơn trung bình đang xét và các giá trị lớn hơn trung bình.



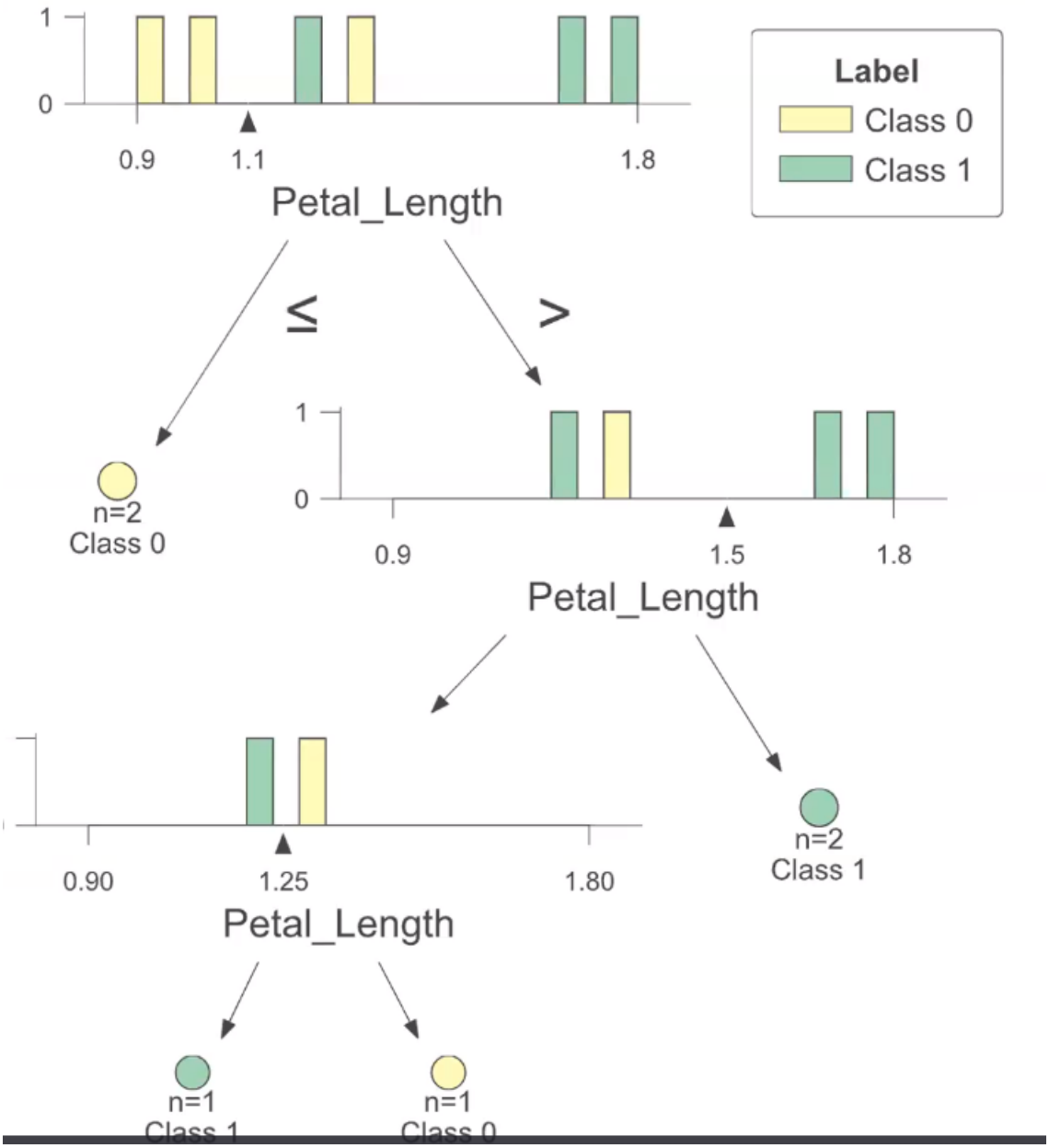
Hình 8: Minh hoạ Tree Data liên tục

Như vậy ta tính gain lần lượt của 0.95, 1.1, 1.25, 1.5, 1.75 và có kết quả là gain của 1.1 sẽ là lớn nhất, ta chọn điều kiện <=1.1 làm nút gốc.

Tính tương tự như ví dụ tính cây bằng biến rời rạc, ta được một sơ đồ cây cho biến liên tục như sau.

Bảng 6: Dữ liệu liên tục (sắp xếp từ bé đến lớn theo Petal_Length) và trung bình cộng

STT	Petal_Length	Label	Trung bình
3	0.9	0	
1	1	0	0.95
2	1.2	1	1.1
6	1.3	0	1.25
4	1.7	1	1.5
5	1.8	1	1.75



Hình 9: Minh hoạ Decison Tree với Data liên tục

2.4 Ví dụ về Decision Tree dùng Mean và Variance (Regression)

Chúng ta có một table thể hiện sự tương quan giữa năm kinh nghiệm (Exp) và mức lương tương ứng (Salary) với năm kinh nghiệm là feature và lương là label.

Bảng 7: Dữ liệu liên tục

STT	Experience	Salary
1	1	0
2	1.5	0
3	2	0
4	2.5	0
5	3	60
6	3.5	64
7	4	55
8	4.5	61
9	5	66
10	5.5	83
11	6	93
12	6.5	91
13	7	98
14	7.5	101

Để có thể hoàn thành bài toán dự đoán Decision Tree, chúng ta cần tính lần lượt Mean(data), Variance(Data).

Với Mean và Variance được biểu diễn như sau:

$$Mean = \frac{1}{S}\sum_{i=1}^{n} S_i \tag{6}$$

- Mean là giá trị trung bình.
- S là số lượng các phần tử trong tập hợp dữ liệu.
- S_i là số lượng các phần tử trong tập hợp dữ liệu.

$$mse = \frac{1}{S}\sum_{i=1}^{n}(S_i - Mean)^2 \tag{7}$$

- mse là phương sai.
- Mean là giá trị trung bình.
- S là số lượng các phần tử trong tập hợp dữ liệu.
- S_i là số lượng các phần tử trong tập hợp dữ liệu.

Chúng ta lần lượt tách dataset theo hàng và phân chia thành dataset-right và dataset-left để có thể tính a_mse Mean và mse của left và right, ưu tiên lựa chọn a_mse có giá trị nhỏ nhất.

Sau khi áp dụng công thức ta có được dataset-left và dataset-right với chỉ số a_mse nhỏ nhất như sau.

STT	Experience	Salary
1	1	0
2	1.5	0
3	2	0
4	2.5	0

Bảng 8: Dataset-left

STT	Experience	Salary
1	3	60
2	3.5	64
3	4	55
4	4.5	61
5	5	66
6	5.5	83
7	6	93
8	6.5	91
9	7	98
10	7.5	101

Bảng 9: Dataset-right

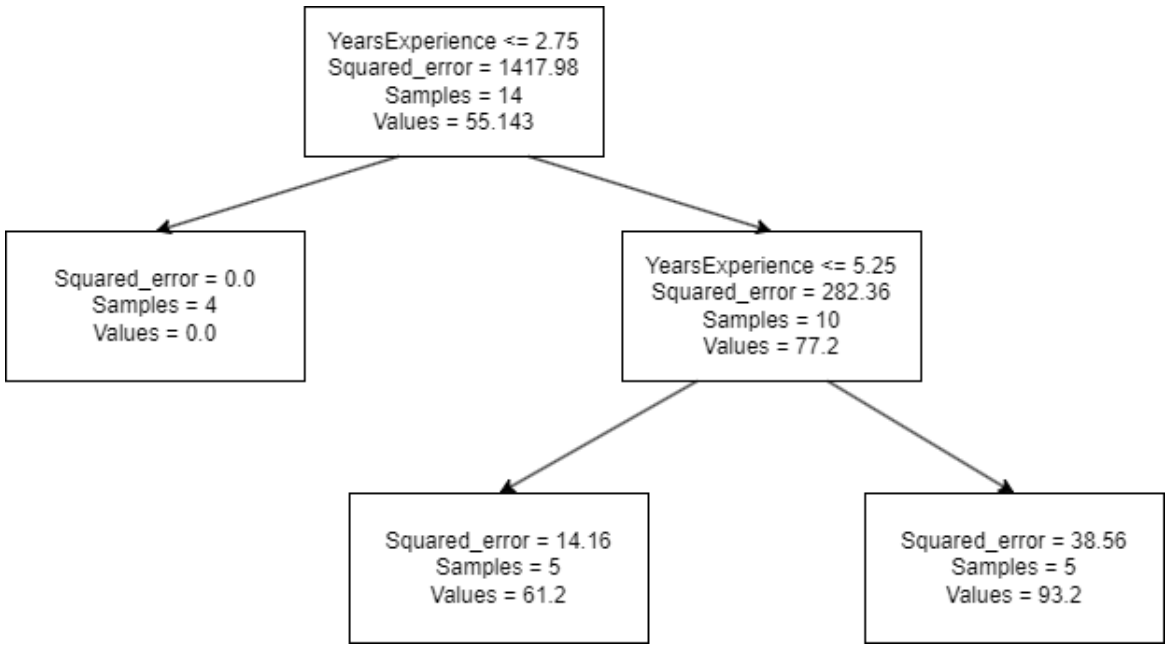
Lần lượt các chỉ số của Mean và Variance của data-left và data-right là: 0,0 và 77.2, 282.35.

$$a_{mse} = \frac{|L|}{|S|} mse_L + \frac{|R|}{|S|} mse_R$$
$$= \frac{4}{14} * 0 + \frac{10}{14} * 282.35$$
$$= 201.68$$

Hình 10: Minh hoạ cách tính a_mse

- với L là số lượng data ở data-left.
- với R là số lượng data ở data-right.
- với S là số lượng data ở data gốc.

Lặp lại việc tách và tìm min(a_mse) chúng ta có được một sơ đồ cây với Depth=3 như sau:



Hình 11: Ví dụ Decision Tree Regression

Chúng ta có thể tiếp tục tách data đến khi Squared_error đạt 0.0 nhưng vì để tránh bị Overfitting nên mô hình dừng ở Depth=3 với chỉ số Squared_error ở mức chấp nhận được.

THAT'S IT, THANK YOU FOR READING ♡♡♡