Tuần 2 - Tổng hợp kiến thức Buổi học số 1 và 2

Time-Series Team

Ngày 15 tháng 7 năm 2025

Buổi học số 1 (Thứ 3, 8/07/2025) và buổi học số 2 (Thứ 4, 9/07/2025) có nhiều nội dung tương đồng và kế thừa nhau nên nhóm mình sẽ tổng hợp lại thành 4 nội dung chính:

- ***Phần 1: Events***
- ***Phần 2: Probability***
- ***Phần 3: Bayes' Theorem***
- ***Phần 4: Mở rộng: Unigram trong bài toán phân loại email spam***

Phần 1: Events

1. Sự kiện (Events)

1.1. Một số khái niệm cơ bản

- **Phép thử (Experiment):** Việc thực hiện một tập hợp các điều kiện nhất định để quan sát một hiện tượng. *Ví dụ: Tung một con xúc xắc một lần.*
- **Kết quả (Outcome):** Kết quả đơn lẻ thu được từ một lần thực hiện phép thử. *Ví dụ: Xuất hiện mặt "4" khi tung xúc xắc.*
- **Không gian mẫu (Sample Space), ký hiệu S hoặc Ω:** Tập hợp tất cả các kết quả có thể xảy ra của một phép thử. *Ví dụ: Tung xúc xắc:*

$$S = \{1, 2, 3, 4, 5, 6\}$$

- **Sự kiện (Event):** Một tập con của không gian mẫu. *Ví dụ: "Xuất hiện số chẵn khi tung xúc xắc"* $\rightarrow A = \{2, 4, 6\}$.

1.2. Các loại sự kiện

- **Sự kiện chắc chắn (Certain Event):** Luôn xảy ra trong phép thử, ký hiệu Ω. *Ví dụ (tung xúc xắc):* $\Omega =$ "số chấm từ 1 đến 6" $= \{1, 2, 3, 4, 5, 6\}$.
- **Sự kiện không thể xảy ra (Impossible Event):** Không bao giờ xảy ra khi thực hiện phép thử, ký hiệu $\varnothing$. *Ví dụ: "Xuất hiện 7 chấm khi tung xúc xắc"* $\rightarrow \varnothing = \{\}$.
- **Sự kiện ngẫu nhiên (Random Event):** Có thể xảy ra hoặc không xảy ra khi thực hiện phép thử. *Ví dụ: "Xuất hiện số chẵn khi tung xúc xắc"* $\rightarrow A = \{2, 4, 6\}$.
- **Phép thử ngẫu nhiên (Random Experiment):** phép thử mà kết quả thu được là các sự kiện ngẫu nhiên. *Ví dụ: Tung xúc xắc, tung đồng xu.*
- **Ký hiệu:** Thông thường các sự kiện được ký hiệu bằng chữ cái in hoa: $A, B, C, \ldots$

1.3. Ví dụ minh họa

Ví dụ với xúc xắc

Tung một con xúc xắc:

- **Sự kiện chắc chắn:** $\Omega =$ {số chấm từ 1 đến 6} $= \{1, 2, 3, 4, 5, 6\}$
- **Sự kiện không thể xảy ra:** $\varnothing =$ {7 chấm}
- **Sự kiện ngẫu nhiên:** $A =$ {số chẵn} $= \{2, 4, 6\}$

1.4. Ví dụ thực tế khác

Ví dụ thực tế

Ví dụ 1: Một gia đình có 2 đứa trẻ

- A = "Gia đình có 1 trai 1 gái" → **Sự kiện ngẫu nhiên**
- B = "Gia đình có 3 đứa trẻ" → **Sự kiện không thể xảy ra**
- C = "Gia đình có 2 đứa trẻ" → **Sự kiện chắc chắn**

Ví dụ 2: Một hộp có 8 quả bóng (6 xanh, 2 đỏ), bốc ngẫu nhiên 3 quả

- A = "Bốc được 3 quả xanh" → **Sự kiện ngẫu nhiên**
- B = "Bốc được 3 quả đỏ" → **Sự kiện không thể xảy ra**
- C = "Bốc được 3 quả bóng" → **Sự kiện chắc chắn**

2. Các phép toán trên sự kiện (Operations with Events)

2.1. Giao của các sự kiện (Intersection)

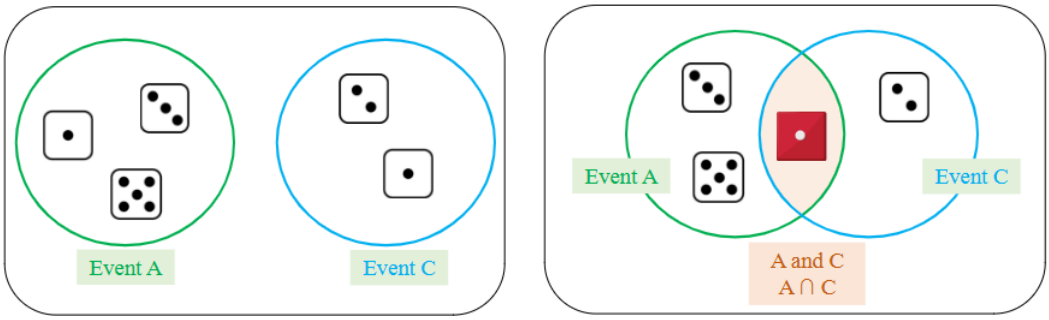
Định nghĩa: Giao của hai sự kiện A và B, ký hiệu $A \cap B$, là tập hợp các kết quả **đồng thời** thuộc cả A và B.

Ví dụ (tung xúc xắc)

- A = "Xuất hiện số lẻ" → $A = \{1, 3, 5\}$
- B = "Xuất hiện số chia hết cho 2" → $B = \{2, 4, 6\}$
- $A \cap B = \{\}$ → **Không có giao nhau**

Ví dụ khác:

- A = "Xuất hiện số lẻ" → $A = \{1, 3, 5\}$
- C = "Xuất hiện số nhỏ hơn 3" → $C = \{1, 2\}$
- $A \cap C = \{1\}$



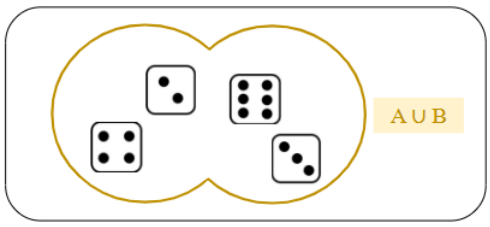
Trái: 2 sự kiện không giao nhau
Phải: 2 sự kiện giao nhau

2.2. Hợp của các sự kiện (Union)

Định nghĩa: Hợp của hai sự kiện A và B, ký hiệu $A \cup B$, là tập hợp các kết quả thuộc A, hoặc B, hoặc cả hai.

Ví dụ (tung xúc xắc)

- $A =$ "Xuất hiện số chẵn" $\rightarrow A = \{2, 4, 6\}$
- $B =$ "Xuất hiện số chia hết cho 3" $\rightarrow B = \{3, 6\}$
- $A \cup B = \{2, 3, 4, 6\}$



Hình 1: Hợp của 2 sự kiện A và B

2.3. Phủ định của sự kiện (Complement)

Định nghĩa: Phủ định của một sự kiện A, ký hiệu A' (hoặc A^c), là tập hợp tất cả các kết quả trong Ω **không thuộc** A. Công thức: $A \cup A' = \Omega$

Ví dụ

- $A =$ "Xuất hiện số lớn hơn 4" $\rightarrow A = \{5, 6\}$
- $A' =$ "Không xuất hiện số lớn hơn 4" $\rightarrow A' = \{1, 2, 3, 4\}$

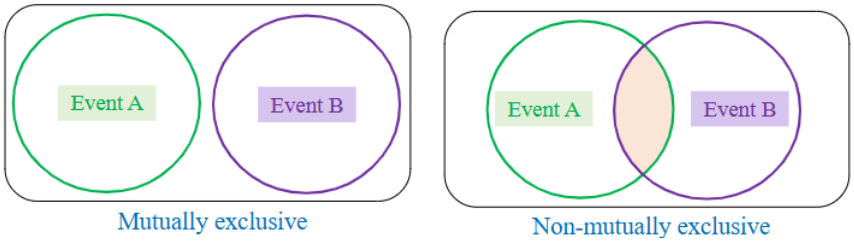
2.4. Sự kiện xung khắc (Mutually Exclusive Events)

Định nghĩa: Hai sự kiện A và B được gọi là **xung khắc** (mutually exclusive) nếu không thể xảy ra đồng thời trong một lần phép thử, tức là:

$$A \cap B = \varnothing$$

Ví dụ

- $A =$ "Xuất hiện số chẵn" $\rightarrow A = \{2, 4, 6\}$
- $B =$ "Xuất hiện số lẻ" $\rightarrow B = \{1, 3, 5\}$
- $A \cap B = \varnothing$



Phần 2: Probability

2. Xác suất (Probability)

2.1. Khái niệm xác suất (Probability)

- **Định nghĩa:** Xác suất của một sự kiện A, ký hiệu $P(A)$, là một số nằm trong khoảng $0 \leq P(A) \leq 1$, thể hiện mức độ xảy ra của sự kiện đó:
  - $P(A) \to 0$: Rất khó xảy ra.
  - $P(A) \to 1$: Gần như chắc chắn xảy ra.
- **Tính chất cơ bản:**

$$0 \leq P(A) \leq 1, \quad P(\Omega) = 1, \quad P(\varnothing) = 0$$

2.1.1. Định nghĩa xác suất cổ điển (Classical Probability)

Định nghĩa: Với phép thử có n kết quả đồng khả năng, xác suất của một sự kiện A là:

$$P(A) = \frac{\text{số kết quả thuận lợi } (\mathrm{n}_A)}{\text{tổng số kết quả có thể } (\mathrm{n}_\Omega)}$$

Ví dụ:

Ví dụ 1: Tung xúc xắc

- $A =$ "Xuất hiện số lẻ" $\to A = \{1, 3, 5\}$
- $n_\Omega = 6, \quad n_A = 3$
- $P(A) = 3/6 = 0.5$

Ví dụ 2: Rút bài từ bộ bài 52 lá

- $A =$ "Rút được quân K (King)" $\to n_A = 4$
- $n_\Omega = 52$
- $P(A) = 4/52 = 1/13$

2.1.2. Định nghĩa xác suất theo quan điểm hình học (Geometric Probability)

Định nghĩa: Quan điểm hình học được sử dụng khi không thể đếm số kết quả rời rạc mà phải dựa trên **độ đo hình học** như:

- Chiều dài (1D – đoạn thẳng),
- Diện tích (2D – mặt phẳng),
- Thể tích (3D – khối).

Khi đó, xác suất của một sự kiện A được tính bằng:

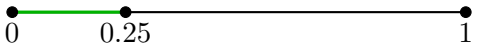
$$P(A) = \frac{\text{Độ đo của miền (vùng) A}}{\text{Độ đo của toàn bộ không gian mẫu } \Omega}$$

với:

- Ω: không gian mẫu (tập hợp tất cả kết quả có thể xảy ra),
- $A \subseteq \Omega$: miền quan tâm (sự kiện cần tính xác suất).

Ví dụ 1 (1D – đoạn thẳng):
Giả sử X được chọn ngẫu nhiên trên đoạn $[0, 1]$. Tìm xác suất $A = \{0 < X < 0.25\}$.

$$P(A) = \frac{\text{Chiều dài của đoạn } (0, 0.25)}{\text{Chiều dài của đoạn } (0, 1)} = \frac{0.25}{1} = 0.25$$



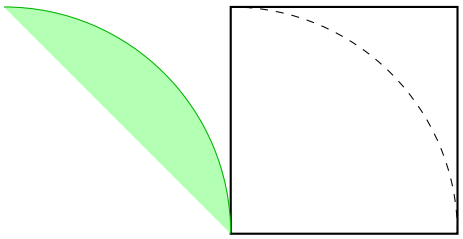
Ví dụ 2 (2D – hình phẳng):
Một điểm được chọn ngẫu nhiên trong hình vuông có cạnh dài 1 (tọa độ (x, y) được phân bố đều trên $[0, 1] \times [0, 1]$). Tìm xác suất điểm đó rơi vào miền $A = \{x^2 + y^2 \leq 1\}$ (góc phần tư hình tròn bán kính 1).

Phân tích:

- Toàn bộ không gian mẫu: hình vuông cạnh 1, diện tích $= 1^2 = 1$.
- Miền quan tâm: 1/4 hình tròn bán kính 1, diện tích $= \frac{\pi r^2}{4} = \frac{\pi}{4}$.

$$P(A) = \frac{\text{Diện tích miền A}}{\text{Diện tích hình vuông}} = \frac{\pi/4}{1} \approx 0.785$$

Kết luận: Xác suất điểm rơi vào góc tròn này khoảng 78.5%.



2.1.3. Mô phỏng (Simulation)

Ví dụ: Mô phỏng tung đồng xu

- Tung đồng xu nhiều lần, đếm số lần xuất hiện mặt sấp (tails) và mặt ngửa (heads).
- Khi số lần thử lớn, tần suất xuất hiện $\approx$ xác suất lý thuyết: $P(\text{Heads}) \approx 0.5$

Ví dụ: Xấp xỉ số π

Ý tưởng: Sinh ngẫu nhiên N điểm trong hình vuông cạnh $s = 2$, đếm số điểm nằm trong hình tròn bán kính $r = 1$. Khi đó:

$$\pi \approx \frac{s^2 \times N_{\text{trong tròn}}}{N_{\text{tổng}}}$$

2.2. Các quy tắc xác suất (Rules of Probability)

2.2.1. Xác suất thực nghiệm (Empirical Probability)

Định nghĩa: Xác suất được ước lượng từ số lần xảy ra của sự kiện trong thực nghiệm:

$$P(A) = \frac{\text{số lần A xảy ra}}{\text{tổng số lần thực hiện phép thử}}$$

Ví dụ: Trong 6 lần kiểm tra:

$$P(\text{Pass}) = 3/6 = 0.5, \quad P(\text{Fail}) = 3/6 = 0.5$$

2.2.2. Quy tắc cộng (Additive Rule)

Quy tắc:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

Nếu A và B xung khắc:

$$P(A \cup B) = P(A) + P(B)$$

Ví dụ:

Ví dụ (xúc xắc)

$A = \{3, 4\} \rightarrow P(A) = P(3) + P(4) = 2/6 = 1/3$

Ví dụ (quảng cáo)

- $P(A =$ "Thăm Hà Nội") $= 0.7$P(B="$ThmTP.HCM$") $= 0.6$
- $P(A \cap B) = 0.4$
- $\Rightarrow P(A \cup B) = 0.7 + 0.6 - 0.4 = 0.9$

2.3. Xác suất có điều kiện (Conditional Probability)

Định nghĩa: Xác suất có điều kiện mô tả khả năng xảy ra của một sự kiện A khi biết chắc rằng một sự kiện B đã xảy ra.

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \quad P(B) \neq 0$$

Trong đó:

- $P(A|B)$: Xác suất A xảy ra **dưới điều kiện** B đã xảy ra.
- $P(A \cap B)$: Xác suất cả A và B xảy ra đồng thời.
- $P(B)$: Xác suất B xảy ra.

Ví dụ: Tung xúc xắc (dice example)
Giả sử ta tung một con xúc xắc công bằng, không gian mẫu $S = \{1, 2, 3, 4, 5, 6\}$.

- A: "Ra số 3" $\rightarrow A = \{3\}$, $P(A) = 1/6$.
- B: "Ra số lẻ" $\rightarrow B = \{1, 3, 5\}$, $P(B) = 3/6 = 1/2$.
- $A \cap B$: "Vừa ra số 3 và số đó lẻ" $\rightarrow A \cap B = \{3\}$, $P(A \cap B) = 1/6$.

Áp dụng công thức:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{1/6}{1/2} = \frac{1}{3}$$

Nếu chỉ xét trong tập B (tức chỉ nhìn các số lẻ $\{1, 3, 5\}$), ta có 3 kết quả khả dĩ như nhau và chỉ có 1 kết quả thỏa mãn A (số 3). Vậy:

$$P(A|B) = \frac{\text{số kết quả thỏa mãn cả A và B}}{\text{tổng kết quả trong B}} = \frac{1}{3}$$

Minh họa bằng bảng:

	Lẻ (B)	**Chẵn ($\overline{B}$)**	**Tổng**
A = 3	1 kết quả	0	1
Không phải 3	2 kết quả	3 kết quả	5
Tổng	3	3	6

Trong 3 kết quả thuộc B, có đúng 1 kết quả đồng thời thuộc A.

Tính $P(B|A)$ (xác suất ra số lẻ khi biết chắc là ra số 3):

$$P(B|A) = \frac{P(A \cap B)}{P(A)} = \frac{1/6}{1/6} = 1$$

Rõ ràng nếu đã biết kết quả là số 3 thì chắc chắn đó là số lẻ.

2.4. Quy tắc nhân (Multiplication Rule)

Định nghĩa: Quy tắc nhân cho phép tính xác suất xảy ra đồng thời của nhiều biến cố phụ thuộc lẫn nhau:

$$P(A \cap B) = P(A) \cdot P(B|A) = P(B) \cdot P(A|B)$$

Tổng quát cho n biến cố:

$$P(A_1A_2 \dots A_n) = P(A_1) \cdot P(A_2|A_1) \cdot P(A_3|A_1A_2) \dots P(A_n|A_1A_2 \dots A_{n-1})$$

Trong đó:

- $P(A_1)$: xác suất xảy ra biến cố đầu tiên,
- $P(A_2|A_1)$: xác suất A_2 xảy ra khi A_1 đã xảy ra,
- ...

Ví dụ: Chọn chìa khóa mở cửa

Có 10 chìa giống hệt nhau, trong đó chỉ có 2 chìa mở được cửa. Ta chọn và thử từng chìa một, bỏ chìa đã thử.
Câu hỏi: Xác suất mở được cửa đúng vào lần thứ 3?

Phân tích biến cố:
Ký hiệu:

- A_1: Chìa đúng ngay lần đầu.
- A_2: Chìa đúng ở lần thứ 2.
- A_3: Chìa đúng ở lần thứ 3.
- $\overline{A_1}$, $\overline{A_2}$: Thử sai ở lần 1, 2.

Ta cần tính:

$$P(\overline{A_1}\ \overline{A_2}\ A_3) = P(\overline{A_1}) \cdot P(\overline{A_2}|\overline{A_1}) \cdot P(A_3|\overline{A_1}\ \overline{A_2})$$

Tính từng bước:

1. **Lần 1 sai:** Có 10 chìa, 2 đúng $\rightarrow$ 8 sai:
$$P(\overline{A_1}) = \frac{8}{10}$$
2. **Lần 2 sai, biết lần 1 sai:** Còn 9 chìa (7 sai, 2 đúng):
$$P(\overline{A_2}|\overline{A_1}) = \frac{7}{9}$$
3. **Lần 3 đúng, biết 2 lần trước sai:** Còn 8 chìa (6 sai, 2 đúng):
$$P(A_3|\overline{A_1}\ \overline{A_2}) = \frac{2}{8}$$

Kết quả:

$$P(\overline{A_1}\ \overline{A_2}\ A_3) = \frac{8}{10} \cdot \frac{7}{9} \cdot \frac{2}{8} \approx 0.155$$

→ Xác suất mở được cửa đúng vào lần thử thứ 3 là khoảng 15.5%.

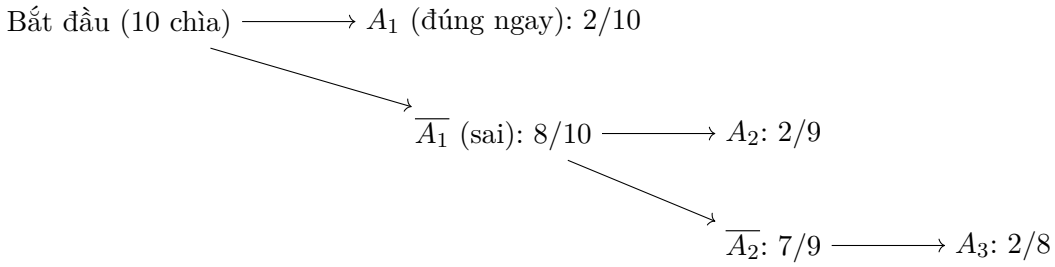
Câu hỏi mở rộng: Xác suất mở được cửa trong tối đa 3 lần thử?
Ta cần cộng ba trường hợp: mở đúng ngay lần đầu, lần 2 hoặc lần 3:

$$\begin{aligned} P(\text{at most 3 attempts}) &= P(A_1) + P(\overline{A_1}A_2) + P(\overline{A_1A_2}A_3) \\ &= \frac{2}{10} + \frac{8}{10} \cdot \frac{2}{9} + \frac{8}{10} \cdot \frac{7}{9} \cdot \frac{2}{8} \\ &= \frac{2}{10} + \frac{16}{90} + \frac{14}{90} \\ &= \frac{8}{15} \approx 0.533 \end{aligned}$$

→ Xác suất mở được cửa trong tối đa 3 lần thử là khoảng 53.3%.

Nhận xét:

- Quy tắc nhân cho phép ta tính xác suất cho các chuỗi sự kiện phụ thuộc lẫn nhau.
- Càng nhiều lần thử, xác suất thành công tích lũy càng tăng.



2.5. Sự kiện độc lập (Independent Events)

Định nghĩa: Hai sự kiện A và B độc lập nếu:

$$P(A \cap B) = P(A) \cdot P(B)$$

Ví dụ (xúc xắc): $A = \{4\},\ B = \{2, 4, 6\}$:

$$P(A) = 1/6, \quad P(B) = 1/2, \quad P(A \cap B) = 1/6$$

$$P(A)P(B) = (1/6)(1/2) = 1/12 \neq P(A \cap B)$$

→ A và B **không độc lập**.

Phần 3: Bayes' Theorem

3. Định lý Bayes (Bayes' Theorem)

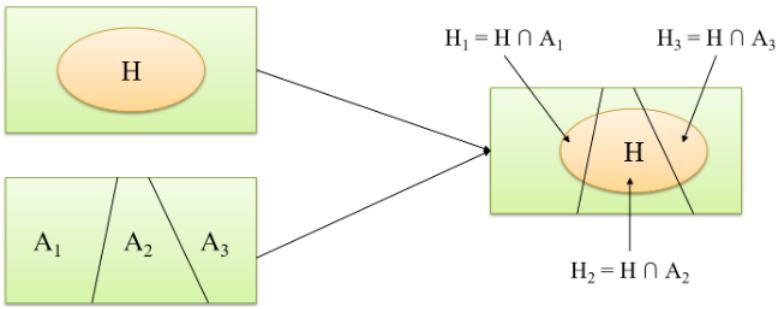
3.1. Định lý xác suất toàn phần (Total Probability Theorem)

Phát biểu: Giả sử $A_1, A_2, \ldots, A_n$ là một **hệ đầy đủ các sự kiện** (complete system of events), tức là:

$$A_1 \cup A_2 \cup \cdots \cup A_n = \Omega \quad \text{và} \quad A_i \cap A_j = \varnothing \ \forall i \neq j$$

Với một sự kiện H, ta có:

$$P(H) = \sum_{i=1}^{n} P(A_i) \cdot P(H|A_i)$$



Minh họa hệ đầy đủ gồm 3 sự kiện A_1, A_2, A_3 và sự kiện H

Ví dụ: Chọn bi trong 3 túi

Ví dụ: Xác suất toàn phần với bi màu

- Có 3 túi, mỗi túi chứa 100 viên bi:
  - Túi 1: 75 bi đỏ, 25 bi xanh
  - Túi 2: 60 bi đỏ, 40 bi xanh
  - Túi 3: 45 bi đỏ, 55 bi xanh
- Chọn ngẫu nhiên một túi, sau đó bốc ngẫu nhiên một viên bi.
- Tìm xác suất để viên bi chọn được là bi đỏ.

Lời giải: Gọi:

$$H: \text{"Chọn được bi đỏ"}$$
$$A_1: \text{"Chọn túi 1"}, \quad P(A_1) = 1/3$$
$$A_2: \text{"Chọn túi 2"}, \quad P(A_2) = 1/3$$
$$A_3: \text{"Chọn túi 3"}, \quad P(A_3) = 1/3$$

Ta có:

$$P(H|A_1) = 0.75, \quad P(H|A_2) = 0.6, \quad P(H|A_3) = 0.45$$

Áp dụng định lý xác suất toàn phần:

$$\begin{aligned} P(H) &= P(A_1)P(H|A_1) + P(A_2)P(H|A_2) + P(A_3)P(H|A_3) \\ &= \frac{1}{3}(0.75 + 0.6 + 0.45) = 0.60 \end{aligned}$$

3.2. Định lý Bayes (Bayes' Rule)

Phát biểu: Với hai sự kiện A và B, $P(A) \neq 0$:

$$P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)}$$

Tổng quát, với hệ đầy đủ các sự kiện $C_1, C_2, \ldots, C_n$:

$$P(C_i|X) = \frac{P(C_i) \cdot P(X|C_i)}{\sum_{j=1}^{n} P(C_j) \cdot P(X|C_j)}, \quad i = 1, 2, \ldots, n$$

Ý nghĩa:

- $P(C_i)$: **Prior** (Xác suất tiên nghiệm)
- $P(X|C_i)$: **Likelihood** (Khả năng xảy ra dữ liệu X khi C_i đúng)
- $P(C_i|X)$: **Posterior** (Xác suất hậu nghiệm, điều ta muốn tìm)
- $P(X)$: **Marginal** (Xác suất tổng, dùng để chuẩn hóa)

Ví dụ: Phát hiện email spam

Ví dụ: Phát hiện email spam với từ "offer"

- C_1: Email spam, C_2: Email không spam (Not spam)
- $P(C_1) = 0.3, \quad P(C_2) = 0.7$
- X: Email chứa từ `"offer"`
- $P(X|C_1) = 0.8, \quad P(X|C_2) = 0.1$

Tìm: $P(C_1|X)$ = Xác suất email là spam khi có từ "offer".

$$\begin{aligned} P(X) &= P(C_1)P(X|C_1) + P(C_2)P(X|C_2) \\ &= 0.3 \cdot 0.8 + 0.7 \cdot 0.1 = 0.31 \end{aligned}$$

Áp dụng định lý Bayes:

$$P(C_1|X) = \frac{P(C_1)P(X|C_1)}{P(X)} = \frac{0.3 \times 0.8}{0.31} \approx 0.774$$

Kết luận: Xác suất email này là spam lên tới 77.4% → nên được đánh dấu là spam.

Ví dụ: Phát hiện email spam bằng định lý Bayes

Ví dụ: Phát hiện email spam với từ "offer"

Đề bài: Giả sử trong hộp thư:

- 30% email là **spam** (C_1), 70% email còn lại là **không spam** (C_2).
- Từ khóa ``offer'' xuất hiện trong:
  - 80% các email spam: $P(X|C_1) = 0.8$
  - 10% các email không spam: $P(X|C_2) = 0.1$

Một email mới tới và có chứa từ ``offer''. Hãy tính xác suất email đó là spam.

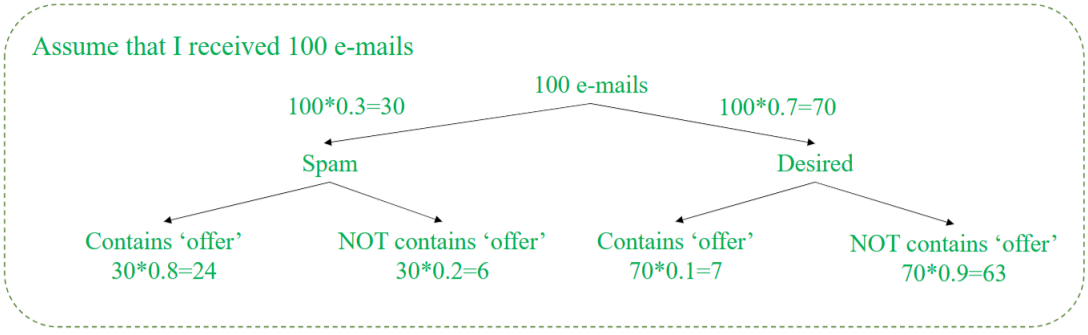
Lời giải:

Bước 1: Gọi sự kiện

$$C_1 : \text{Email spam}, \quad P(C_1) = 0.3$$
$$C_2 : \text{Email không spam}, \quad P(C_2) = 0.7$$
$$X : \text{"Email chứa từ offer"}$$

Bước 2: Vẽ sơ đồ cây để dễ hình dung
Giả sử ta có **100 email**:

- Chọn ngẫu nhiên: 30 email là spam, 70 email không spam.
- Trong số 30 email spam:
  - Có từ ``offer'': $30 \times 0.8 = 24$ email
  - Không có từ ``offer'': $30 \times 0.2 = 6$ email
- Trong số 70 email không spam:
  - Có từ ``offer'': $70 \times 0.1 = 7$ email
  - Không có từ ``offer'': $70 \times 0.9 = 63$ email



Sơ đồ cây

Bảng tóm tắt:

Loại email	**Có "offer"**	**Không "offer"**	**Tổng**
Spam (C_1)	24	6	30
Không spam (C_2)	7	63	70
Tổng	31	69	100

Bước 3: Tính xác suất tổng của sự kiện X (email chứa "offer")
Dựa vào bảng trên:

$$P(X) = \frac{31}{100} = 0.31$$

hoặc theo công thức xác suất toàn phần:

$$P(X) = P(C_1)P(X|C_1) + P(C_2)P(X|C_2) = 0.3 \times 0.8 + 0.7 \times 0.1 = 0.31$$

Bước 4: Áp dụng định lý Bayes

$$P(C_1|X) = \frac{P(C_1)P(X|C_1)}{P(X)} = \frac{0.3 \times 0.8}{0.31} = \frac{0.24}{0.31} \approx 0.774$$

Bước 5: Kết luận
Nếu một email có chứa từ ``offer'', xác suất nó là **spam** lên tới **77.4%**. → **Nên được đánh dấu là spam.**

Từ bài toán trên, chúng ta có thể rút ra được phương pháp giải chung cho dạng bài toán này:

Cách giải bài toán Bayes theo 5 bước

Bước 1: Xác định các biến cố và ký hiệu

- Chọn ra các biến cố $C_1, C_2, \ldots, C_n$ tạo thành **hệ đầy đủ** (phân chia toàn bộ không gian mẫu). Ví dụ: "Spam" và "Không spam"; hoặc "Bệnh" và "Không bệnh".
- Gọi X là **dữ kiện quan sát được** (ví dụ: "có từ offer" hoặc "test dương tính").
- Ghi rõ các xác suất đã cho:

  $P(C_i)$ (xác suất tiên nghiệm) $P(X|C_i)$ (khả năng xảy ra dữ kiện khi C_i đúng)

Bước 2: Vẽ sơ đồ cây xác suất (gợi ý nên dùng khi mới học)

- Tầng 1: các nhánh $C_1, C_2, \ldots, C_n$ với xác suất $P(C_i)$.
- Tầng 2: mỗi nhánh C_i tiếp tục chia thành 2 (hoặc nhiều) nhánh: X và $\overline{X}$, với xác suất $P(X|C_i)$ và $1 - P(X|C_i)$.
- Tính số phần tử giả định trên 100 (hoặc 1000) đối tượng để dễ hình dung.

Bước 3: Tính xác suất tổng của dữ kiện X (xác suất toàn phần)

$$P(X) = \sum_{i=1}^{n} P(C_i)\, P(X|C_i)$$

(hoặc lấy tổng số phần tử thuộc nhánh X chia cho tổng).

Bước 4: Áp dụng định lý Bayes để tìm xác suất hậu nghiệm

$$P(C_k|X) = \frac{P(C_k)\, P(X|C_k)}{P(X)}$$

Chỉ cần thay k bằng biến cố cần hỏi (thường là "Bệnh" hoặc "Spam").

Bước 5: Kết luận và giải thích

- So sánh xác suất hậu nghiệm với ngưỡng (ví dụ >50% → dự đoán Spam).
- Diễn giải bằng cách đếm số phần tử trong 100 (hoặc 1000): Ví dụ: "Trong 100 trường hợp có dữ kiện X, có bao nhiêu thuộc C_k?"

Mẹo

Nếu thấy khó nhớ công thức, hãy luôn:

1. Vẽ cây → điền từng nhánh.
2. Đếm phần tử → trực giác chính là định lý Bayes!

4. Simple Classification

Ví dụ: Dự đoán kết quả thi dựa trên việc có học hay không

Đề bài: Ta có dữ liệu về 6 sinh viên với kết quả thi (**Result**) và việc có học bài (**Studied**):

Result (Kết quả)	Studied (Học bài?)
Pass	Yes
Pass	Yes
Pass	No
Fail	Yes
Fail	No
Fail	No

Mục tiêu: Dự đoán xác suất một sinh viên sẽ "Pass" nếu biết rằng bạn ấy có học bài (Studied = Yes).

Bước 1: Xác định các biến cố và ký hiệu

$$C_1 : \text{Result} = \text{Pass}, \quad P(C_1) = P(\text{Pass})$$
$$C_2 : \text{Result} = \text{Fail}, \quad P(C_2) = P(\text{Fail})$$
$$X : \text{Studied} = \text{Yes}$$

Từ bảng dữ liệu:

$$P(\text{Pass}) = \frac{3}{6} = 0.5 P(\text{Fail}) = \frac{3}{6} = 0.5$$

Bước 2: Tính các xác suất có điều kiện $P(X|C_i)$

- Trong 3 người "Pass": có 2 người học bài

$$P(\text{Studied} = \text{Yes}|\text{Pass}) = \frac{2}{3}$$

- Trong 3 người "Fail": chỉ 1 người học bài

$$P(\text{Studied} = \text{Yes}|\text{Fail}) = \frac{1}{3}$$

Bước 3: Tính xác suất tổng của X (Total Probability)

$$\begin{aligned} P(X) &= P(\text{Studied} = \text{Yes}) \\ &= P(C_1)P(X|C_1) + P(C_2)P(X|C_2) \\ &= 0.5 \times \frac{2}{3} + 0.5 \times \frac{1}{3} \\ &= \frac{1}{2} \end{aligned}$$

Bước 4: Áp dụng định lý Bayes

$$\begin{aligned} P(\text{Pass}|\text{Studied} = \text{Yes}) &= \frac{P(C_1)P(X|C_1)}{P(X)} \\ &= \frac{0.5 \times \frac{2}{3}}{\frac{1}{2}} \\ &= \frac{\frac{1}{3}}{\frac{1}{2}} = \frac{2}{3} \approx 0.67 \end{aligned}$$

Tương tự:

$$\begin{aligned} P(\text{Fail}|\text{Studied} = \text{Yes}) &= 1 - P(\text{Pass}|\text{Studied} = \text{Yes}) \\ &= 1 - \frac{2}{3} = \frac{1}{3} \approx 0.33 \end{aligned}$$

Bước 5: Kết luận
Nếu một sinh viên có học bài, xác suất bạn ấy sẽ **Pass** là khoảng 67%, cao gấp đôi xác suất bị trượt (33%).

→ Nên khuyên học bài để tăng cơ hội đỗ!

Phần 4: Mở rộng: Unigram trong bài toán phân loại email spam

> *"Unigram nghe có vẻ học thuật, nhưng thực ra nó chính là cách đơn giản nhất để dạy máy tính hiểu một đoạn văn."*

4.1. Giới thiệu

Trong thời đại AI, việc **máy tính có thể đọc email và nhận ra thư rác (spam)** không còn là chuyện mới mẻ. Nhưng chúng ta đã bao giờ tự hỏi:

> Máy làm điều đó như thế nào, chỉ dựa vào những dòng chữ?

Câu trả lời nằm ở một mô hình xác suất đơn giản nhưng cực kỳ hiệu quả: **Unigram model**.

4.2. Bài toán: Phân loại email spam

Đặt vấn đề

Giả sử ta có 2 email sau:

- **Email A**:

```
Congratulations! You won a FREE iPhone. Click now.

```

- **Email B**:

```
Dear student, your class schedule is updated for tomorrow.

```

Câu hỏi: Email nào là thư rác?
Với một hệ thống AI, việc này tương đương với **phân loại văn bản vào hai nhóm**:

- **Spam** (rác)
- **Ham** (bình thường)

Đây chính là bài toán **phân loại văn bản** – một ứng dụng điển hình của **Naive Bayes Classifier** và **Unigram model**.

4.3. Unigram model là gì?

Unigram (hay **1-gram**) là mô hình đơn giản nhất trong các **mô hình ngôn ngữ n-gram** – nơi một câu hoặc đoạn văn bản được coi như một chuỗi các token (thường là từ), và **mỗi token được giả định là độc lập với các token khác**.

Uni-gram là gì?

Unigram model giả định rằng xác suất xuất hiện của từng từ trong câu là độc lập với các từ khác:

$$P(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} P(w_i)$$

→ Đây là một mô hình **naive (ngây thơ)**, nhưng cực kỳ nhanh và dễ triển khai.

Nói cách khác: Từ "FREE" xuất hiện không ảnh hưởng gì đến khả năng từ "iPhone" xuất hiện sau đó.

4.4. Mô hình xác suất Naive Bayes + Unigram

Để phân loại một email là **Spam** hay **Ham**, chúng ta áp dụng mô hình **Naive Bayes**, dựa trên định lý Bayes và giả định đơn giản (naive) rằng: *các từ trong email là độc lập có điều kiện, tức là không phụ thuộc nhau khi biết nhãn*. Đây chính là nơi mô hình **Unigram** phát huy tác dụng.

Cụ thể, ta tính:

$$P(\text{Spam} \mid \text{email}) \propto P(\text{Spam}) \times \prod_{i=1}^{n} P(w_i \mid \text{Spam})$$

$$P(\text{Ham} \mid \text{email}) \propto P(\text{Ham}) \times \prod_{i=1}^{n} P(w_i \mid \text{Ham})$$

Trong đó:

- $w_1, w_2, \ldots, w_n$ là các từ trong email cần phân loại.
- $P(\text{Spam})$ và $P(\text{Ham})$ là xác suất tiên nghiệm (prior probability), ước lượng từ dữ liệu huấn luyện:

$$P(\text{Spam}) = \frac{\text{số email spam}}{\text{tổng số email}}, \quad P(\text{Ham}) = \frac{\text{số email ham}}{\text{tổng số email}}$$

- $P(w_i \mid \text{Spam})$ là xác suất có điều kiện: từ w_i xuất hiện trong email spam (tương tự với Ham). Đây chính là **trọng tâm** của mô hình ngôn ngữ Unigram:

$$P(w \mid \text{Spam}) = \frac{\text{số lần từ } w \text{ xuất hiện trong email spam} + 1}{\text{tổng số từ trong email spam} + V}$$

- V là kích thước từ điển (corpus) — số lượng từ phân biệt trong toàn bộ tập huấn luyện (chúng ta có thể áp dụng **Laplace smoothing** (giới thiệu sau) để xử lý những từ chưa từng thấy).

Cuối cùng, ta chọn nhãn có xác suất hậu nghiệm (posterior) cao hơn:

$$\text{label}^* = \arg\max_{\text{label} \in \{\text{Spam}, \text{Ham}\}} P(\text{label}) \times \prod_{i=1}^{n} P(w_i \mid \text{label})$$

Lưu ý

Mô hình Unigram chính là phần "$P(w_i \mid \textbf{label})$" trong công thức trên. Nó giả định rằng các từ trong email không ảnh hưởng lẫn nhau, và xác suất của toàn bộ email là tích các xác suất từ riêng lẻ.

Ví dụ: Giả sử ta có email: `"click here to get FREE iPhone"`
Với mỗi từ trong email, ta tra:

- P("click" | Spam), P("click" | Ham)
- P("FREE" | Spam), ...
- ...

Tính tích xác suất theo từng nhãn, rồi so sánh:

$$P(\text{Spam} \mid \text{email}) \quad \text{vs} \quad P(\text{Ham} \mid \text{email})$$

Nhãn nào cao hơn thì ta dự đoán đó là loại của email.

Tóm lại: Mô hình **Naive Bayes + Unigram** là một trong những mô hình đơn giản và hiệu quả nhất trong phân loại văn bản. Nó đặc biệt phù hợp cho bài toán phân loại email spam — nơi một số từ như `"free"`, `"buy"`, `"click"` có xác suất cao trong lớp spam, và rất thấp trong lớp ham.

4.5. Smoothing – xử lý từ chưa từng thấy

Trong thực tế, khi áp dụng mô hình Naive Bayes + Unigram để phân loại văn bản, ta thường gặp một vấn đề nghiêm trọng:

Nếu một từ trong email test chưa từng xuất hiện trong tập huấn luyện, xác suất $P(w_i \mid \textbf{label})$ của nó sẽ bằng 0.
Ví dụ, từ `"FREEdom"` chưa từng xuất hiện trong bất kỳ email nào bạn huấn luyện. Khi đó:

$$P(\text{"FREEdom"} \mid \text{Spam}) = 0$$

Dù các từ khác có xác suất cao, thì:

$$P(\text{Spam} \mid \text{email}) = P(\text{Spam}) \times \underbrace{0}_{\text{"FREEdom"}} \times \cdots = 0$$

→ Toàn bộ xác suất trở thành 0, làm hỏng dự đoán.

Giải pháp:

Ta sử dụng kỹ thuật **Laplace smoothing** (hay còn gọi là "add-one smoothing") để đảm bảo rằng mọi từ, kể cả từ chưa từng thấy, đều có xác suất khác 0.

$$P(w \mid \text{label}) = \frac{\text{Count}(w) + 1}{\text{Total words (label)} + V}$$

Trong đó:

- Count(w) là số lần từ w xuất hiện trong văn bản thuộc nhãn đó
- V là số lượng từ phân biệt (kích thước từ điển)
- Total words (label) là tổng số từ (có lặp lại) trong tất cả văn bản thuộc nhãn đó

Ý nghĩa của Laplace smoothing:

- Mỗi từ, dù xuất hiện hay không, đều được cộng thêm 1 vào số lần xuất hiện.
- Đồng thời, mẫu số (tổng từ) cũng được cộng thêm V (vì ta cộng thêm 1 cho mỗi từ trong từ điển).
- Kết quả là xác suất nhỏ nhưng khác 0 cho những từ chưa từng xuất hiện.

Ví dụ

Giả sử:

- Tổng số từ trong email spam là 5000
- Số lượng từ phân biệt $V = 1000$
- Từ "FREEdom" chưa từng xuất hiện → Count = 0

Khi đó:

$$P(\text{"FREEdom"} \mid \text{Spam}) = \frac{0+1}{5000+1000} = \frac{1}{6000} \approx 0.000167$$

→ Thay vì bằng 0, ta vẫn có một xác suất nhỏ cho từ chưa từng thấy.

Tóm lại: Laplace smoothing là một bước **bắt buộc** trong hầu hết các mô hình Naive Bayes để đảm bảo mô hình không bị "vỡ" khi gặp từ mới. Nó giúp đảm bảo tính ổn định và khả năng tổng quát của mô hình khi áp dụng vào dữ liệu thực tế.

4.6. Huấn luyện mô hình từ dữ liệu

Để sử dụng mô hình Naive Bayes + Unigram, ta cần huấn luyện từ một tập dữ liệu văn bản đã được **gán nhãn**. Giả sử ta có tập dữ liệu gồm 1000 email, trong đó:

- 600 email là **Spam**
- 400 email là **Ham (không phải spam)**

Bước 1: Tính xác suất tiên nghiệm (prior)
Xác suất một email bất kỳ là spam hay ham (ước lượng từ dữ liệu):

$$P(\text{Spam}) = \frac{600}{1000} = 0.6$$
$$P(\text{Ham}) = \frac{400}{1000} = 0.4$$

Bước 2: Tính xác suất có điều kiện $P(w_i \mid \text{label})$ cho từng từ

Giả sử trong toàn bộ các email spam, bạn đếm được:

- "FREE": xuất hiện 200 lần
- "iPhone": 120 lần
- "Click": 180 lần
- Tổng số từ (token) trong toàn bộ email spam: 5000 từ

Tương tự, bạn đếm được trong email ham:

- "FREE": xuất hiện 5 lần
- "Tổng số từ trong email ham": 4000 từ

Khi đó:

$$P(\text{"FREE"} \mid \text{Spam}) = \frac{200}{5000} = 0.04$$
$$P(\text{"FREE"} \mid \text{Ham}) = \frac{5}{4000} = 0.00125$$

Tổng quát: Với mỗi từ w, ta tính:

$$P(w \mid \text{label}) = \frac{\text{số lần } w \text{ xuất hiện trong email của label}}{\text{tổng số từ trong label}}$$

Bước 3: Thêm smoothing
Để tránh xác suất bằng 0 khi gặp từ mới trong dữ liệu test, ta dùng **Laplace smoothing**:

$$P(w \mid \text{label}) = \frac{\text{count}(w) + 1}{\text{total words} + V}$$

Trong đó V là số lượng từ khác nhau (kích thước từ điển).

Code

```python
from collections import Counter, defaultdict

# Simple data simulation
spam_emails = [
    "FREE iPhone click now",
    "Click to get your FREE prize",
    "FREE FREE FREE click click iPhone"
]
```

```

ham_emails = [
    "Meeting schedule for tomorrow",
    "Your class has been updated",
    "Reminder: team meeting at 10AM"
]

# Preprocessing: split words, lowercase
def tokenize(email):
    return email.lower().split()

# Count words for each class
spam_words = []
for email in spam_emails:
    spam_words.extend(tokenize(email))

ham_words = []
for email in ham_emails:
    ham_words.extend(tokenize(email))

# Total words and count times
spam_counts = Counter(spam_words)
ham_counts = Counter(ham_words)
spam_total = sum(spam_counts.values())
ham_total = sum(ham_counts.values())

# General dictionary
vocab = set(spam_counts) | set(ham_counts)
V = len(vocab)

# The function P(w | label) has smoothing
def compute_prob(w, label):
    if label == 'spam':
        return (spam_counts[w] + 1) / (spam_total + V)
    else:
        return (ham_counts[w] + 1) / (ham_total + V)

# Example: calculate the probability of the word "free"
print("P('free' | spam):", compute_prob("free", "spam"))
print("P('free' | ham): ", compute_prob("free", "ham"))
```

Output:

```
P('free' | spam): 0.16666666666666666
P('free' | ham):  0.029411764705882353
```

Ta thấy rằng từ `"free"` có xác suất cao trong nhóm spam, và thấp trong nhóm ham. Điều này phản ánh đúng trực giác và là lý do vì sao từ này là dấu hiệu mạnh cho spam.

Ghi chú: Việc đếm từ và tính xác suất có thể được thực hiện trên tập dữ liệu lớn hơn (như Enron Spam Dataset hoặc SMS Spam Collection). Với dữ liệu đủ lớn, mô hình Unigram + Naive Bayes có thể đạt độ chính xác trên 95% trong bài toán phát hiện email spam.

4.7. Làm dự đoán cho một email mới

Sau khi đã huấn luyện xong mô hình (tính được $P(w \mid \text{label})$ cho mọi từ), ta có thể sử dụng nó để phân loại một email mới.

Giả sử bạn nhận được email mới:

```
Click now to get a FREE iPhone
```

Mục tiêu: Tính:

$$P(\text{Spam} \mid \text{email}) \propto P(\text{Spam}) \times \prod_i P(w_i \mid \text{Spam})$$

$$P(\text{Ham} \mid \text{email}) \propto P(\text{Ham}) \times \prod_i P(w_i \mid \text{Ham})$$

Trong đó:

- $P(\text{Spam}) = 0.6$, $P(\text{Ham}) = 0.4$
- Các từ w_i gồm: `"click", "now", "to", "get", "a", "free", "iphone"`
- Các xác suất $P(w_i \mid \text{label})$ được tính bằng Laplace smoothing ở bước trước

Lưu ý: Nếu ta nhân tất cả các xác suất lại với nhau, sẽ rất dễ bị **tràn số (underflow)** do tích của các số rất nhỏ. Vì vậy, ta sẽ tính **logarithm** của xác suất (log-probability) để ổn định hơn:

$$\log P(\text{Spam} \mid \text{email}) = \log(0.6) + \sum_i \log P(w_i \mid \text{Spam})$$

$$\log P(\text{Ham} \mid \text{email}) = \log(0.4) + \sum_i \log P(w_i \mid \text{Ham})$$

Sau đó, ta so sánh hai giá trị này. Nhãn nào có giá trị log lớn hơn → mô hình dự đoán nhãn đó.

Code

```python
import math

# New email needs sorting
new_email = "Click now to get a FREE iPhone"

# Tokenization as training step
words = tokenize(new_email)

# Function to calculate the overall log-probability of an email
def predict_log_prob(words, label):
    if label == 'spam':
        log_prob = math.log(0.6)  # prior
    else:
        log_prob = math.log(0.4)

    for w in words:
        prob = compute_prob(w.lower(), label)
        log_prob += math.log(prob)
```

```
19
20      return log_prob
21
22  # Calculate the log probability for both labels.
23  log_spam = predict_log_prob(words, 'spam')
24  log_ham = predict_log_prob(words, 'ham')
25
26  print("Log P(Spam | email):", log_spam)
27  print("Log P(Ham  | email):", log_ham)
28
29  # Predict
30  predicted_label = "Spam" if log_spam > log_ham else "Ham"
31  print("==> Predict:", predicted_label)
```

Output:

```
1 Log P(Spam | email): -19.016206980948663
2 Log P(Ham  | email): -25.60081440418729
3 ==> Predict: Spam
```

Giải thích:

- Vì xác suất xuất hiện các từ như `"free"`, `"click"`, `"iphone"` trong email spam cao hơn trong ham.
- Tổng log-xác suất của email rơi vào spam cao hơn → hệ thống dự đoán đó là thư rác.

Tóm lại: Mô hình Naive Bayes + Unigram dựa trên nguyên lý đơn giản nhưng hiệu quả:

- Ước lượng xác suất từ → nhân/tính log → cộng dồn lại
- Chọn nhãn có log-xác suất cao nhất

4.8. Ưu và nhược điểm của Unigram trong phân loại spam

Sau khi đã hiểu rõ cách huấn luyện và dự đoán bằng mô hình Naive Bayes + Unigram, ta hãy cùng phân tích sâu hơn về điểm mạnh và điểm yếu của mô hình này.

Ưu điểm

- **Đơn giản, dễ triển khai:** Mô hình Unigram chỉ cần đếm tần suất từ và tính xác suất. Không cần học đặc trưng phức tạp hay dùng mạng nơ-ron. Thậm chí có thể viết từ đầu chỉ với vài chục dòng Python.
- **Tính toán nhanh:** Mọi bước chỉ gồm cộng, chia, và log → tính toán nhanh và có thể huấn luyện/training trên máy yếu hoặc xử lý thời gian thực. Thích hợp với email server, hệ thống spam đơn giản, hoặc thiết bị IoT.
- **Hiệu quả bất ngờ với dữ liệu đủ lớn:** Khi tập huấn luyện đủ phong phú (từ hàng ngàn đến hàng triệu email), mô hình có thể học ra rất nhiều từ khóa "nhạy cảm" với

spam như: `"free"`, `"win"`, `"offer"`, `"urgent"`,...
Trong thực tế, mô hình Unigram + Naive Bayes thường đạt độ chính xác 90–95% trong bài toán phát hiện email spam, dù đơn giản.

=> Có thể dùng để khởi tạo mô hình hoặc baseline.

Nhược điểm

- **Không xét ngữ cảnh giữa các từ:** Unigram giả định rằng các từ là độc lập → mô hình không phân biệt được giữa các câu có cùng từ nhưng khác nghĩa. Ví dụ: `"I want to buy a phone"` và `"Buy I want phone a to"` sẽ được gán cùng xác suất, vì chỉ xét từng từ riêng lẻ, không xét thứ tự.
- **Không phát hiện được các mẫu theo cấu trúc:** Nhiều thư rác sử dụng cấu trúc đặc biệt như "Click here to claim", "Act now and win",... nhưng mô hình Unigram không phát hiện ra cụm từ nào đang lặp đi lặp lại vì nó tách từng từ rời rạc.
- **Dễ bị sai khi gặp từ mới hoặc lừa đảo từ vựng:** Ví dụ: nếu kẻ gửi spam viết `"fr33 iph0ne cl1ck"` thay vì `"free iphone click"`, thì mô hình Unigram sẽ không nhận ra vì đây là từ chưa từng thấy → xác suất thấp → không phát hiện spam. Ngoài ra, nếu email test chứa quá nhiều từ mới (không có trong từ điển huấn luyện), các xác suất trở nên thiếu ổn định → mô hình dễ đoán sai.

=> Không phù hợp cho các tác vụ cần hiểu ngữ pháp, ngữ nghĩa.

Tóm lại:

Ưu điểm	**Nhược điểm**
Đơn giản, dễ triển khai và dễ hiểu	Không xét ngữ cảnh giữa các từ
Tính toán nhanh, dùng được cả trên máy yếu	Không phát hiện được các mẫu theo cụm từ hoặc ngữ pháp
Hiệu quả tốt khi có đủ dữ liệu huấn luyện	Dễ bị sai nếu từ mới xuất hiện quá nhiều hoặc bị đánh lừa cú pháp

4.9. Kết luận

Unigram là mô hình cực kỳ đơn giản nhưng lại rất hữu dụng cho bài toán phân loại email spam. Dù không xét đến ngữ cảnh, nó **vẫn đạt hiệu quả rất tốt khi kết hợp với Naive Bayes** và một chút kỹ thuật như **smoothing**.