ĐẠI HỌC QUỐC GIA TPHCM

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

AI23 (23TNT1), FIT@HCMUS-VNUHCM

Báo cáo Lab 3

Đề tài: Phân loại thư rác

Môn học: Phương pháp Toán cho Trí tuệ nhân tạo

Sinh viên thực hiện:
Nguyễn Đình Hà Dương (23122002)
Nguyễn Lê Hoàng Trung (23122004)
Đinh Đức Tài (23122013)
Hoàng Minh Trung (23122014)

Giáo viên hướng dẫn:
TS. Cấn Trần Thành Trung
ThS. Nguyễn Ngọc Toàn

Ngày 17 tháng 6 năm 2025

Mục lục

1 Giới thiệu

Đây là bài báo cáo cho **Lab 2 - Phân loại thư rác**, môn Phương pháp toán cho Trí tuệ nhân tạo, lớp Trí tuệ nhân tạo Khóa 2023 (23TNT1), Khoa Công nghệ thông tin, Trường Đại học Khoa học tự nhiên - Đại học Quốc gia TP.HCM.

Trong bài báo cáo này, chúng tôi sẽ trình bày phương pháp **phân loại thư rác** dựa trên **Naive bayes classifier**, **MLE**, **MAP** và **Bag-of-Words**.

Báo cáo được thực hiện bởi nhóm các thành viên:

- Nguyễn Đình Hà Dương (23122002)
- Nguyễn Lê Hoàng Trung (23122004)
- Đinh Đức Tài (23122013)
- Hoàng Minh Trung (23122014)

Đường dẫn repository Github của báo cáo: https://github.com/ductai05/Math-For-AI [1]

Bảng phân công nhiệm vụ cho từng thành viên:

Họ và tên	MSSV	Nhiệm vụ
Nguyễn Đình Hà Dương	23122002	- Báo cáo Naive Bayes Classifier. - Review report.
Nguyễn Lê Hoàng Trung	23122004	- Code Naive Bayes Classifier. - Đánh giá và nhận xét kết quả mô hình.
Đinh Đức Tài	23122013	- Báo cáo Bag-of-Words. - Báo cáo tổng quan pipeline xử lý. Review report.
Hoàng Minh Trung	23122014	- Báo cáo MLE, MAP. - Kiểm tra code Naive Bayes Classifier.

Các thư viện và công nghệ sử dụng:

- Numpy, Pandas: Thư viện Python để xử lý số học, thao tác và xử lý dữ liệu.
- collections.Counter: Đếm tần suất xuất hiện của các phần tử trong một iterable hoặc từ một mapping.
- matplotlib.pyplot: Dùng để tạo ra các loại biểu đồ và hình ảnh trực quan hóa dữ liệu một cách dễ dàng và nhanh chóng.
- scipy.sparse.csr_matrix: Dùng để lưu trữ và thao tác hiệu quả với các ma trận thưa, giúp tiết kiệm bộ nhớ và tăng tốc độ tính toán.

2 MLE và MAP

2.1 MLE - Ước lượng hợp lý cực đại

Ước lượng hợp lý cực đại (Maximum Likelihood Estimation - MLE) là phương pháp ước lượng tham số của một phân phối xác suất dựa trên dữ liệu quan sát. Cho họ phân phối p_a với tham số $a \in D$ (miền giá trị) và tập dữ liệu $\Theta = \{x_1, \ldots, x_n\}$ được lấy mẫu độc lập từ p_a, hàm hợp lý được định nghĩa:

$$\mathcal{L}(a \mid \Theta) = \prod_{i=1}^{n} p(x_i; a).$$

Mục tiêu của MLE là tìm a_{MLE} sao cho tối đa hóa hàm hợp lý:

$$a_{\text{MLE}} = \arg\max_{a \in D} \mathcal{L}(a \mid \Theta).$$

Để đơn giản hóa tính toán, ta thường sử dụng log-hàm hợp lý, vì log là hàm đơn điệu:

$$\log \mathcal{L}(a \mid \Theta) = \sum_{i=1}^{n} \log p(x_i; a)$$

Khi đó:

$$a_{\text{MLE}} = \arg\max_{a \in D} \log \mathcal{L}(a \mid \Theta).$$

2.2 MAP - Ước lượng cực đại hậu nghiệm

Ước lượng cực đại hậu nghiệm (Maximum A Posterior Estimation - MAP) là phương pháp ước lượng tham số dựa trên việc kết hợp dữ liệu quan sát với thông tin tiền nghiệm. Cho họ phân phối p_a với tham số $a \in D$ và tập dữ liệu $\Theta = \{x_1, \ldots, x_n\}$ được lấy mẫu độc lập từ p_a, cùng với phân phối tiên nghiệm $\mu \sim p_{a_0}$. Hàm hợp lý có điều kiện được định nghĩa:

$$\mathcal{L}(a \mid a_0, \Theta) = \prod_{i=1}^{n} p_a(x_i \mid a_0).$$

Mục tiêu của MAP là tìm a_{MAP} tối đa hóa hàm hợp lý kết hợp với tiền nghiệm:

$$a_{\text{MAP}} = \arg\max_{a \in D} \mathcal{L}(a \mid a_0, \Theta).$$

Trong thực tế, ta thường tối ưu hóa log-hàm hợp lý để đơn giản hóa:

$$\log \mathcal{L}(a \mid a_0, \Theta) = \sum_{i=1}^{n} \log p_a(x_i \mid a_0).$$

MAP hữu ích khi dữ liệu quan sát hạn chế và thông tin tiền nghiệm từ p_{a_0} có thể cải thiện độ chính xác của ước lượng.

3 Naive bayes classifier

Naive Bayes Classifier (NBC) là một mô hình phân loại dựa trên lý thuyết xác suất, đặc biệt là định lý Bayes. Naive Bayes là một mô hình phân loại đơn giản nhưng mạnh mẽ, đặc biệt trong các ứng dụng NLP như phân loại email, phân tích cảm xúc,... Sự hiệu quả đến từ khả năng tổng quát tốt với dữ liệu lớn và tính toán nhanh chóng.

3.1 Định lý Bayes

Giả sử ta có một nhóm đầy đủ $A_1, A_2, \ldots, A_n$, và xảy ra một sự kiện H nào đó. Khi đó, xác suất có điều kiện của A_i khi biết H được xác định như sau:

$$P(A_i \mid H) = \frac{P(A_i)P(H \mid A_i)}{P(H)} \tag{*}$$

Trong đó, theo định lý xác suất toàn phần, ta có:

$$P(H) = \sum_{i=1}^{n} P(A_i)P(H \mid A_i) \tag{**}$$

Thay (**) vào (*), ta được công thức Bayes tổng quát:

$$P(A_i \mid H) = \frac{P(A_i)P(H \mid A_i)}{\sum\limits_{i=1}^{n} P(A_i)P(H \mid A_i)} \tag{***}$$

3.2 Naive Bayes

Xét bài toán phân loại (classification) với C lớp $1, 2, \ldots, C$. Giả sử ta có một điểm dữ liệu mới $\mathbf{x} \in \mathbb{R}^d$, mục tiêu là xác định xác suất để điểm này thuộc vào lớp c, tức là tính $p(y = c \mid \mathbf{x})$, hoặc viết gọn là $p(c \mid \mathbf{x})$. Ta sẽ chọn nhãn có xác suất lớn nhất theo công thức:

$$c = \arg\max_{c \in \{1,\ldots,C\}} p(c \mid \mathbf{x}) \tag{1}$$

Áp dụng định lý Bayes:
Xác suất $p(c \mid \mathbf{x})$ có thể khó tính trực tiếp. Ta áp dụng định lý Bayes:

$$p(c \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid c)p(c)}{p(\mathbf{x})} \tag{2}$$

Do $p(\mathbf{x})$ là biểu thức không phụ thuộc c, ta có thể viết:

$$c = \arg\max_{c} p(\mathbf{x} \mid c)p(c) \tag{3}$$

Xác suất p(c) là xác suất để một điểm dữ liệu rơi vào lớp c. Giá trị này có thể tính bằng MLE, tức tỉ lệ số điểm dữ liệu trong tập training rơi vào class này chia cho tổng số lượng dữ liệu trong tập training.

3.3 Giả định Naive (Độc lập)

Xác suất $p(\mathbf{x} \mid c)$, tức là phân phối của các điểm dữ liệu trong nhãn y, thường rất khó tính toán vì $\mathbf{x}$ là một biến ngẫu nhiên nhiều chiều. Để giúp cho việc tính toán trở nên đơn giản hơn, Giả định Naive thường đưa rằng các thành phần của biến ngẫu nhiên $\mathbf{x}$ là độc lập với nhau nếu biết y. Tức là:

$$p(\mathbf{x} \mid y) = p(x_1, x_2, \ldots, x_d \mid y) = \prod_{i=1}^{d} p(x_i \mid y)$$

Giả định này giúp đơn giản hóa việc ước lượng xác suất, đặc biệt khi số lượng đặc trưng rất lớn như trong văn bản.

3.4 Multinomial Naive Bayes (cho dữ liệu rời rạc)

Trong mô hình Multinomial Naive Bayes, ta tính xác suất có điều kiện của một đặc trưng x_i (thường là từ trong văn bản) xuất hiện trong lớp c như sau:

$$p(x_i \mid c) = \frac{\text{count}(x_i, c)}{\sum_j \text{count}(x_j, c)}$$

- x_i: đặc trưng thứ i (trong ngữ cảnh xử lý văn bản, đây thường là một từ trong từ điển); c: một lớp cụ thể trong bài toán phân loại.
- $\text{count}(x_i, c)$: số lần đặc trưng x_i xuất hiện trong tất cả văn bản thuộc lớp c; $\sum_j \text{count}(x_j, c)$: tổng số lần xuất hiện của tất cả đặc trưng x_j trong các văn bản thuộc lớp c.

Khi đó, $p(x_i \mid c)$ tỉ lệ với tần suất từ thứ i (hay đặc trưng thứ i trong trường hợp tổng quát) xuất hiện trong các văn bản thuộc lớp c.

Tuy nhiên, nếu một đặc trưng x_i không xuất hiện trong bất kỳ văn bản nào thuộc lớp c trong tập huấn luyện, thì $\text{count}(x_i, c) = 0$, dẫn đến xác suất $p(x_i \mid c) = 0$. Điều này khiến cho toàn bộ xác suất hậu nghiệm $P(c \mid \mathbf{x})$ trở thành 0 gây ảnh hưởng nghiêm trọng đến quá trình phân loại. Để khắc phục, ta sử dụng kỹ thuật **làm trơn Laplace**. Công thức xác suất sau khi làm trơn là:

$$p(x_i \mid c) = \frac{\text{count}(x_i, c) + \alpha}{\sum_j \text{count}(x_j, c) + \alpha \cdot V}$$

Trong đó:

- $\alpha > 0$ là hệ số làm trơn Laplace, giúp tránh xác suất bằng 0 (thường chọn $\alpha = 1$),
- V là kích thước từ vựng (số lượng đặc trưng khác nhau).

Ứng dụng: Kỹ thuật này rất phổ biến trong bài toán phân loại văn bản như lọc thư rác, gán nhãn chủ đề, phân tích cảm xúc, nơi dữ liệu thường được biểu diễn dưới dạng mô hình bag-of-words hoặc n-grams.

4 Mô hình Bag-of-Words

Bag-of-Words (BoW) [2] là một dạng biểu diễn đơn giản được sử dụng trong xử lý ngôn ngữ tự nhiên và truy vấn thông tin. BoW xem một đoạn văn bản (một câu hoặc một tài liệu) như một "túi" (bag) chứa các từ, hoàn toàn bỏ qua ngữ pháp và thứ tự của các từ, mà chỉ quan tâm đến tần suất xuất hiện của chúng.

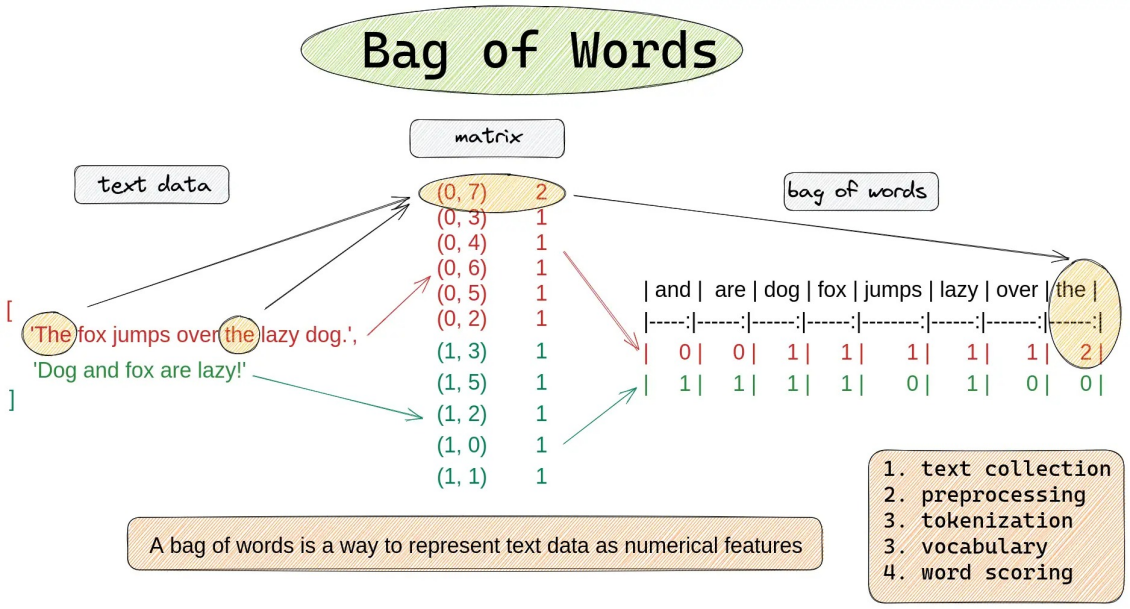
Nói một cách đơn giản, **BoW biến một đoạn văn bản thành một danh sách thống kê số lượng của từng từ**.

Mô hình BoW chủ yếu được dùng trong các phương pháp phân loại văn bản. Với mô hình này, các đặc trưng dùng để huấn luyện bộ phân loại được tạo ra dựa trên sự xuất hiện hoặc tần suất của mỗi từ trong văn bản.

Để áp dụng mô hình Bag-of-Words cho một tập hợp nhiều văn bản (gọi là corpus), chúng ta thực hiện các bước sau:

1. **Tách từ** (Tokenization): Tách các câu thành các từ riêng lẻ (token). Thường thì các dấu câu sẽ bị loại bỏ và các từ được chuyển thành chữ thường.
2. **Xây dựng từ điển**: Tạo ra một bộ từ điển chứa tất cả các từ duy nhất xuất hiện trong toàn bộ corpus.
3. **Vector hóa** (Vectorization): Với mỗi tài liệu, tạo một vector số. Độ dài của vector này bằng với kích thước của từ điển. Mỗi phần tử trong vector biểu diễn số lần xuất hiện của một từ tương ứng trong từ điển.

Kết quả là chúng ta đã chuyển đổi thành công văn bản thành các vector số mà máy tính có thể hiểu và xử lý.



Hình 1: Minh họa mô hình Bag-of-Words

5 Phân loại thư rác trên tập dữ liệu Enron-Spam

5.1 Tổng quan

Bộ dữ liệu **Enron-Spam** là một nguồn tài liệu tuyệt vời được thu thập bởi V. Metsis, I. Androutsopoulos và G. Paliouras và được mô tả trong ấn phẩm của họ "Spam Filtering with Naive Bayes - Which Naive Bayes?" [3]. Bộ dữ liệu chứa tổng cộng 17.171 thư rác và 16.545 thư không phải thư rác ("ham") (tổng cộng 33.716 thư điện tử).

Dựa trên kiến thức lý thuyết về Maximum Likelihood Estimation (MLE), Maximum A Posteriori Estimation (MAP) và mô hình Bag-of-Words, chúng tôi sẽ sử dụng mô hình thống kê **Naive Bayes Classifier** để **phân loại thư rác** trên bộ dữ liệu Enron-Spam. Quy trình các bước như sau:

1. **Tải, khám phá và tiền xử lý dữ liệu.**
2. **Vector hóa văn bản dựa trên mô hình Bag-of-Words.**
3. **Phân loại thư rác bằng Multinomial Naive Bayes.**
4. **Đánh giá mô hình bằng độ chính xác, ma trận nhầm lẫn.**

5.2 Bộ dữ liệu Enron-Spam và tiền xử lý dữ liệu

Bộ dữ liệu **Enron-Spam** được cung cấp dưới dạng 2 file: `train.csv` và `val.csv`, ứng với 2 tập **training** và **validation**. Trong đó gồm:

- `train.csv`: 27284 dòng; 4 cột: `Message ID`, `Subject`, `Message`, `Spam/Ham`.
- `val.csv`: 3084 dòng; 4 cột: `Message ID`, `Subject`, `Message`, `Spam/Ham`.
  - `Message ID`: Chỉ số của thư.
  - `Subject`: Tiêu đề thư
  - `Message`: Nội dung thư.
  - `Spam/Ham`: Biến phân loại thư. `Ham` là thư bình thường, `Spam` là thư rác.

Đây lần lượt là 5 dòng đầu tiên của tập training (`train.csv`) và tập validation (`val.csv`):

	Message ID	Subject	Message	Spam/Ham	split
0	0	christmas tree farm pictures	NaN	ham	0.038415
1	1	vastar resources , inc .	gary , production from the high island larger ...	ham	0.696509
2	2	calpine daily gas nomination	- calpine daily gas nomination 1 . doc	ham	0.587792
3	3	re : issue	fyi - see note below - already done .\nstella\...	ham	-0.055438
5	5	mcmullen gas for 11 / 99	jackie ,\nsince the inlet to 3 river plant is ...	ham	-0.419658

Hình 2: 5 dòng đầu tiên của tập train

	Message ID	Subject	Message	Spam/Ham	split
23	23	miscellaneous	- fo...	ham	-0.351998
24	24	re : purge of old contract _ event _ status	fyi - what do you all think ?\n- - - - - - - -...	ham	0.257704
32	32	valero 8018 and 1394	it is my understanding the outages valero incu...	ham	0.091200
37	37	01 / 00 natural gas nomination	enron methanol company nominates the following...	ham	-1.745133
43	43	re : misc . questions	- fo...	ham	-1.911987

Hình 3: 5 dòng đầu tiên của tập validation

Với các bước tiền xử lý dữ liệu, chúng tôi lần lượt áp dụng các kĩ thuật sau:

1. **Thay thế các ô NaN** (Not-a-Number) bằng chuỗi kí tự rỗng. Chuyển các kí tự chữ thành chữ thường.
2. **Gộp chuỗi kí tự** trong hai cột `Subject` và `Message` thành một chuỗi kí tự duy nhất và tạo cột mới: `full_text`.
3. **Chuyển đổi nhãn** của cột `ham/spam` thành số: `ham -> 0`, `spam -> 1`.

5.3 Vector hóa văn bản bằng Bag-of-Words

Chúng tôi áp dụng phương pháp Bag-of-Words được nhắc đến trong phần 4 để chuyển đổi các chuỗi kí tự trong cột `full_text` (được hợp bởi `Subject` và `Message`) thành vector. Các vector này có đặc điểm:

- Mỗi vector ứng với một `full_text` của một thư. Các vector này biểu diễn tần suất xuất hiện của các từ có trong `full_text`.
- Kích thước các vector là bằng nhau và bằng với số lượng từ của từ điển được xây dựng trên toàn bộ cột `full_text`, theo phương pháp Bag-of-Words.
- Mỗi vector có kích thước là `(1, 37830)`, ứng với số lượng từ có trong từ điển là 37830.

5.4 Multinomial Naive Bayes

Mô hình được sử dụng trong quá trình huấn luyện là **Multinomial Naive Bayes**, một mô hình xác suất dựa trên *định lý Bayes* với giả định "naive" rằng các đặc trưng là *độc lập có điều kiện* khi biết nhãn lớp. Đây là lựa chọn đặc biệt phù hợp trong các bài toán phân loại văn bản, nơi mà đặc trưng của dữ liệu là *tần suất xuất hiện của từ* trong tài liệu.

5.4.1 Huấn luyện mô hình (fit)

Mục tiêu: Học các tham số của mô hình từ tập dữ liệu huấn luyện.

Quy trình huấn luyện:

1. **Xây dựng từ điển**: Thu thập tất cả các từ xuất hiện trong tập dữ liệu huấn luyện để tạo không gian đặc trưng.
2. **Vector hóa dữ liệu**: Biểu diễn các văn bản dưới dạng ma trận BoW (*Bag-of-Words*).
3. **Tính log xác suất tiên nghiệm** $P(c)$:
$$P(c) = \frac{N_c}{N}$$
trong đó N_c là số lượng mẫu thuộc lớp c, và N là tổng số mẫu trong tập huấn luyện.
4. **Tính tổng số từ xuất hiện trong mỗi lớp**.
5. **Tính log xác suất có điều kiện (likelihood)** $P(\text{word}_i \mid c)$ cho mỗi từ và mỗi lớp.

5.4.2 Dự đoán log xác suất hậu nghiệm cho một vector BoW thưa

Mục tiêu: Tính toán *log của tử số xác suất hậu nghiệm* cho một văn bản (biểu diễn dưới dạng vector BoW thưa) tương ứng với từng lớp.

Với mỗi văn bản $x^{(i)}$, mô hình tính:

$$\log P(c) + \sum_j x_j^{(i)} \cdot \log P(\text{word}_j \mid c)$$

trong đó $x_j^{(i)}$ là số lần từ thứ j xuất hiện trong văn bản $x^{(i)}$.

Không áp dụng phép chuẩn hoá (tức không chia cho mẫu số), vì ta chỉ cần so sánh tương đối giữa các lớp.

Nhãn dự đoán $\hat{y}^{(i)}$ được xác định theo nguyên lý **Maximum A Posteriori (MAP)**:

$$\hat{y}^{(i)} = \arg\max_{c \in \mathcal{C}} \left(\log P(c) + \sum_j x_j^{(i)} \cdot \log P(\text{word}_j \mid c) \right)$$

Quá trình này được áp dụng cho mọi $i = 1, \ldots, n$, tạo thành vector dự đoán:

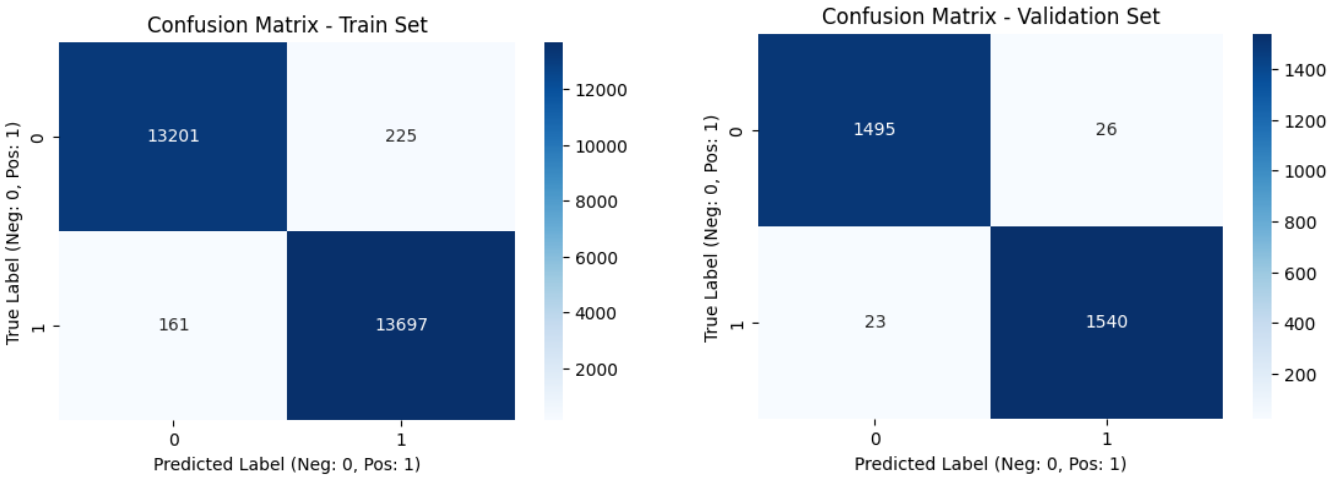
$$\hat{y} = [\hat{y}^{(1)}, \hat{y}^{(2)}, \ldots, \hat{y}^{(n)}]$$

5.5 Đánh giá mô hình

Ở bước đánh giá mô hình, chúng tôi sử dụng 2 phương pháp chính là **độ chính xác** (accuracy) và **ma trận nhầm lẫn** (confusion matrix).

- **Độ chính xác:** Được tính bằng số trường hợp mô hình phân loại `ham/spam` đúng trên tổng số trường hợp cần phân loại.
- **Ma trận nhầm lẫn**: Phương pháp đánh giá hiệu suất của bài toán phân loại, trong đó:
  - TP (True Positive): Số lượng thư `spam` được phân loại đúng.
  - TN (True Negative): Số lượng thư `ham` được phân loại đúng.

– FP (False Positive - Type 1 Error): Số lượng thư `ham` bị phân loại sai thành `spam`.
– FN (False Negative - Type 2 Error): Số lượng thư `spam` bị phân loại sai thành `ham`.



(a) Độ chính xác trên tập Training: 0.9893 (b) Độ chính xác trên tập Validation: 0.9857

Hình 4: Ma trận nhầm lẫn của mô hình trên tập training và validation

6 Đánh giá, kết luận

Qua quá trình thực hiện báo cáo *Lab 3 - Phân loại thư rác*, nhóm đã thành công trong việc tìm hiểu và tự lập trình xây dựng một mô hình *Naive Bayes Classifier*. Mô hình được xây dựng dựa trên các kiến thức nền tảng về lý thuyết xác suất, *Định lý Bayes*, cùng với các kỹ thuật ước lượng tham số như *Maximum Likelihood Estimation (MLE)* cho xác suất tiên nghiệm của lớp và *Maximum A Posteriori (MAP) Estimation* (thông qua *Laplace Smoothing*) cho xác suất có điều kiện của từ. Nguyên tắc *Bag-of-Words* đã được áp dụng hiệu quả để biểu diễn dữ liệu văn bản, tập trung vào tần suất xuất hiện của từ mà bỏ qua cấu trúc ngữ pháp phức tạp.

Khi được huấn luyện và đánh giá trên bộ dữ liệu *Enron-Spam*, mô hình của nhóm đã đạt được độ chính xác tổng thể trên 98.9% trên tập huấn luyện và 98.5% trên tập kiểm định, cho thấy khả năng học tốt các đặc điểm của dữ liệu và khả năng tổng quát hóa cao trên dữ liệu mới chưa từng thấy. Các chỉ số đánh giá chi tiết như Precision, Recall, và F1-Score cho cả hai lớp ‘‘`ham`’’ và ‘‘`spam`’’ đều ở mức rất cao (trên 0.98), phản ánh sự cân bằng và hiệu quả của mô hình trong việc vừa phát hiện chính xác thư rác, vừa hạn chế tối đa việc phân loại nhầm thư hợp lệ. Ma trận nhầm lẫn cũng cho thấy số lượng False Positives (23 trên tập validation) và False Negatives (26 trên tập validation) là tương đối thấp, khẳng định tính thực tiễn của giải pháp.

Việc tự lập trình mô hình *Naive Bayes Classifier* đã giúp nhóm hiểu sâu hơn và có thể áp dụng thực tiễn kiến thức nền tảng về lý thuyết xác suất, cùng các kỹ thuật MLE, MAP được học ở trên lớp để đảm bảo xây dựng mô hình hoạt động ổn định và hiệu quả.

Tài liệu

[1] Ha Duong, Hoang Trung, Duc Tai, and Minh Trung. Math for AI - MTH00056, AI23@HCMUS. https://github.com/ductai05/Math-For-AI/, 2025.

[2] Wisam A Qader, Musa M Ameen, and Bilal I Ahmed. An overview of bag of words; importance, implementation, applications, and challenges. In *2019 international engineering conference (IEC)*, pages 200–204. IEEE, 2019.

[3] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. Spam filtering with naive bayes-which naive bayes? In *CEAS*, volume 17, pages 28–69. Mountain View, CA, 2006.