

**BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP.HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN**

-----o0o-----



**BÁO CÁO HỌC PHẦN: KHAI PHÁ DỮ LIỆU
TÊN ĐỀ TÀI: ỨNG DỤNG PHÂN LOẠI EMAIL
RÁC SỬ DỤNG NAIVE BAYES**

NHÓM: 15

Thành phố Hồ Chí Minh, 5 tháng 6 năm 2025

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP.HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

-----o0o-----



TÊN ĐỀ TÀI: ỨNG DỤNG PHÂN LOẠI EMAIL
RÁC SỬ DỤNG NAIVE BAYES

Nhóm: 15 Thành viên: 1. Trần Công Minh - 2001222641 2. Lê Đức Trung - 2001225676 3. Nguyễn Hữu Thắng - 2001230909	Giảng viên hướng dẫn: Trần Như Ý
--------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------

Thành phố Hồ Chí Minh, 5 tháng 6 năm 2025

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin được gửi lời cảm ơn chân thành nhất đến cô Trần Như Ý. Trong quá trình học tập và tìm hiểu môn **Khai phá dữ liệu**, chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ, hướng dẫn tâm huyết và tận tình của cô. Cô đã giúp chúng em tích lũy thêm nhiều kiến thức về môn học này để có thể hoàn thành được bài báo cáo về đề tài: **Ứng dụng phân loại email rác sử dụng Naive Bayes**.

Trong quá trình làm bài chắc chắn khó tránh khỏi những thiếu sót. Do đó, chúng em kính mong nhận được những lời góp ý của cô để chúng em tiếp thu thêm kiến thức mới và hoàn thiện cho những lần sau.

Chúng em xin chân thành cảm ơn!

BẢNG PHÂN CÔNG VÀ ĐÁNH GIÁ

Họ tên	Nhiệm vụ được phân công	Đóng góp tỷ lệ %	Mức độ hoàn thành
Trần Công Minh	<ul style="list-style-type: none">• Xây dựng mô hình Naive Bayes (naive_bayes.py)• Tiền xử lý văn bản tiếng Việt với underthesea• Thiết kế thuật toán phân loại và các đặc trưng• Đánh giá hiệu suất mô hình và điều chỉnh tham số• Word	100	100%
Lê Đức Trung	<ul style="list-style-type: none">• Phát triển API Flask và cấu trúc backend (app.py)• Tích hợp Gmail API với OAuth 2.0 (gmail_oauth.py)• Triển khai các endpoint cho việc lấy, phân tích và quản lý email	100	100%
Nguyễn Hữu Thắng	<ul style="list-style-type: none">• Thiết kế giao diện người dùng• Kết nối với backend thông qua các API service• Tạo các chức năng tương tác (phân tích email, soạn email...)• PowerPoint	100	100%

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN.....	2
1.1. Mục tiêu đồ án	2
1.2. Phạm vi và giới hạn	3
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	5
2.1. Thuật toán phân loại Naive Bayes	5
2.1.1. Nguyên lý hoạt động của Naive Bayes	5
2.1.2. Đặc điểm "Naive" (Ngây thơ) trong Naive Bayes.....	6
2.1.3. Multinomial Naive Bayes	6
2.1.4. Ưu điểm của Naive Bayes trong phân loại spam.....	7
2.2. Kỹ thuật biểu diễn văn bản TF-IDF.....	8
2.2.1. Nguyên lý của TF-IDF	8
2.2.2. Vai trò của TF-IDF trong phân loại email	8
2.2.3. Tối ưu hóa TF-IDF.....	9
2.3. Xử lý ngôn ngữ tự nhiên cho tiếng việt	9
2.3.1. Thách thức xử lý tiếng Việt	9
2.3.2. Tiền xử lý văn bản tiếng Việt.....	9
2.3.3. Xử lý stopwords cho tiếng Việt.....	11
2.3.4. Tích hợp xử lý ngôn ngữ với phân loại	11
CHƯƠNG 3. THIẾT KẾ VÀ TRIỂN KHAI HỆ THỐNG	12
3.1. Kiến trúc tổng thể hệ thống.....	12
3.1.1. Kiến trúc backend	12
3.1.2. Kiến trúc frontend	12
3.1.3. Tích hợp Gmail API	13

3.2.	Quy trình xử lý dữ liệu.....	13
3.2.1.	<i>Thu thập và tiền xử lý dữ liệu.....</i>	13
3.2.2.	<i>Trích xuất đặc trưng.....</i>	14
3.2.3.	<i>Phân chia dữ liệu huấn luyện.....</i>	14
3.2.4.	<i>Grid Search tối ưu hóa tham số</i>	14
3.3.	Phân tích và đánh giá hiệu suất mô hình	14
3.3.1.	<i>Các metric đánh giá</i>	14
3.3.2.	<i>Luồng tương tác người dùng.....</i>	15
3.3.3.	<i>Thống kê hiệu suất thời gian thực</i>	15
CHƯƠNG 4.	TRIỂN KHAI VÀ ĐÁNH GIÁ	16
4.1.	Cài đặt và triển khai thuật toán	16
4.1.1.	<i>Môi trường triển khai</i>	16
4.1.2.	<i>Triển khai thuật toán</i>	16
4.1.3.	<i>Tích hợp với ứng dụng web</i>	18
4.2.	Kết quả thực nghiệm	19
4.2.1.	<i>Hiệu suất mô hình</i>	20
4.2.2.	<i>Ma trận nhầm lẫn (Confusion Matrix).....</i>	20
4.3.	Ứng dụng thực tế.....	21
4.3.1.	<i>Tích hợp với Gmail.....</i>	21
4.3.2.	<i>Giao diện người dùng</i>	21
CHƯƠNG 5.	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	26
5.1.	Tổng kết kết quả đạt được.....	26
5.1.1.	<i>Về mặt học thuật.....</i>	26
5.1.2.	<i>Về mặt ứng dụng</i>	26

5.1.3.	<i>Về mặt công nghệ</i>	27
5.2.	Những thách thức và hạn chế.....	27
5.2.1.	<i>Hạn chế về mô hình</i>	27
5.2.2.	<i>Hạn chế về kỹ thuật</i>	27
5.2.3.	<i>Hạn chế về ứng dụng</i>	28
5.3.	Đề xuất hướng phát triển tương lai	28
5.3.1.	<i>Cải tiến mô hình</i>	28
5.3.2.	<i>Mở rộng tính năng</i>	28
TÀI LIỆU THAM KHẢO		30

DANH MỤC HÌNH ẢNH

<i>Hình 1. Naive Bayes là gì ?</i>	5
<i>Hình 2. Mô hình các loại Naive Bayes</i>	7
<i>Hình 3. Hiệu suất mô hình</i>	20
<i>Hình 4. Ma trận nhầm lẫn</i>	21
<i>Hình 5. Giao diện trang phân tích email</i>	22
<i>Hình 6. Giao diện trang Hộp thư</i>	22
<i>Hình 7. Giao diện modal chi tiết email và phân loại</i>	23
<i>Hình 8. Giao diện trang thống kê và huấn luyện lại mô hình</i>	24
<i>Hình 9. Giao diện trang soạn email</i>	25

LỜI MỞ ĐẦU

Thư rác (spam mail) là một trong những vấn đề dai dẳng của hệ thống thư điện tử và mạng internet nói chung. Theo các thống kê gần đây, hơn 45% email lưu thông trên internet là thư rác với nội dung quảng cáo không mong muốn, lừa đảo, phishing hoặc chứa mã độc. Điều này không chỉ gây khó khăn cho người dùng trong việc quản lý hộp thư mà còn tiềm ẩn nhiều rủi ro về bảo mật thông tin.

Bài toán phân loại email spam là một nhiệm vụ quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên và học máy, nhằm tự động nhận diện và phân loại các email thành hai nhóm chính: email thông thường (ham) và email rác (spam). Đây là bài toán phân loại nhị phân cổ điển nhưng luôn cần được cập nhật và cải tiến để đối phó với các kỹ thuật tạo spam ngày càng tinh vi.

Đặc biệt, đối với ngôn ngữ tiếng Việt, bài toán này càng trở nên phức tạp hơn do đặc thù của ngôn ngữ như: cấu trúc từ ghép, dấu thanh, và sự đa dạng trong cách diễn đạt. Các công cụ lọc spam thông thường thường được tối ưu cho tiếng Anh, dẫn đến hiệu quả không cao khi áp dụng cho email tiếng Việt.

Trong đồ án này, chúng em tập trung vào việc phát triển một hệ thống phân loại email spam hiệu quả, đặc biệt tối ưu cho cả tiếng Việt, với khả năng tích hợp trực tiếp vào tài khoản Gmail của người dùng để mang lại trải nghiệm thực tế.

CHƯƠNG 1. TỔNG QUAN

1.1. Mục tiêu đề án

Đề án của chúng em hướng đến các mục tiêu cụ thể sau:

1. **Xây dựng một hệ thống phân loại email spam hoàn chỉnh:** Phát triển một ứng dụng web cho phép người dùng kết nối với tài khoản Gmail của họ và thực hiện việc phân loại, quản lý email spam một cách tự động và hiệu quả.
2. **Áp dụng thuật toán Naive Bayes kết hợp với kỹ thuật TF-IDF:** Triển khai và tối ưu hóa mô hình Naive Bayes đa thức (Multinomial Naive Bayes) kết hợp với kỹ thuật biểu diễn văn bản TF-IDF (Term Frequency-Inverse Document Frequency) để phân loại email với độ chính xác cao.
3. **Xử lý ngôn ngữ tiếng Việt hiệu quả:** Tích hợp thư viện underthesea cho việc xử lý ngôn ngữ tiếng Việt, bao gồm tách từ, chuẩn hóa dấu và loại bỏ từ dừng (stopwords) phù hợp với tiếng Việt.
4. **Tính năng phân tích và giải thích kết quả:** Cung cấp các phân tích chi tiết về lý do tại sao một email được phân loại là spam, bao gồm các từ khóa quan trọng, độ tin cậy của dự đoán, và các đặc điểm thống kê của email.
5. **Tích hợp với Gmail thông qua OAuth 2.0:** Xây dựng một hệ thống an toàn kết nối với Gmail API thông qua xác thực OAuth 2.0, cho phép người dùng truy cập và quản lý email mà không cần chia sẻ mật khẩu.
6. **Giao diện người dùng thân thiện:** Thiết kế một giao diện web hiện đại, dễ sử dụng với React, Vite và Tailwind CSS, giúp người dùng dễ dàng tương tác với hệ thống.
7. **Khả năng cải thiện liên tục:** Cho phép người dùng đóng góp vào việc huấn luyện mô hình bằng cách đánh dấu các email được phân loại sai, giúp cải thiện hiệu suất của hệ thống theo thời gian.

1.2. Phạm vi và giới hạn

Phạm vi của dự án:

1. **Kết nối với Gmail:** Hệ thống được thiết kế để tích hợp trực tiếp với tài khoản Gmail của người dùng thông qua Gmail API và xác thực OAuth 2.0.
2. **Phân loại email tiếng Việt:** Mô hình được huấn luyện để nhận diện cả email spam tiếng Việt, với sự tối ưu đặc biệt cho tiếng Việt thông qua thư viện underthesea.
3. **Phân tích chi tiết email:** Hệ thống không chỉ phân loại email mà còn cung cấp phân tích chi tiết về các đặc điểm của email, các từ khóa ảnh hưởng đến kết quả phân loại, và mức độ tin cậy của dự đoán.
4. **Quản lý email spam:** Người dùng có thể xem, đánh dấu, bỏ đánh dấu và xóa email spam trực tiếp từ giao diện ứng dụng.
5. **Thông kê và đánh giá hiệu suất:** Hệ thống cung cấp các số liệu thống kê về hiệu suất mô hình, tỷ lệ phát hiện spam, và các mẫu spam phổ biến.

Giới hạn của dự án:

1. **Phụ thuộc vào Gmail API:** Hệ thống chỉ làm việc với Gmail và không hỗ trợ các dịch vụ email khác như Outlook, Yahoo Mail.
2. **Giới hạn của xác thực OAuth:** Người dùng cần có tài khoản Google và phải cấp quyền cho ứng dụng, điều này có thể gặp hạn chế đối với người dùng lo ngại về quyền riêng tư.
3. **Phạm vi dữ liệu huấn luyện:** Mô hình được huấn luyện trên một tập dữ liệu cố định, mặc dù có cơ chế cập nhật, nhưng vẫn có thể gặp khó khăn với các loại spam mới hoặc rất khác biệt.
4. **Xử lý ngôn ngữ tiếng Việt:** Mặc dù đã tích hợp thư viện chuyên biệt, việc xử lý tiếng Việt vẫn có những thách thức do đặc thù của ngôn ngữ và sự đa dạng trong cách diễn đạt.
5. **Hiệu suất thời gian thực:** Quá trình phân tích email với số lượng lớn có thể tốn thời gian, đặc biệt khi cần phân tích nhiều đặc trưng và từ khóa.

6. **Mở rộng và nâng cấp mô hình:** Mô hình hiện tại tập trung vào Naive Bayes và TF-IDF, chưa tích hợp các kỹ thuật học máy tiên tiến hơn như học sâu (deep learning) hoặc mô hình ngôn ngữ lớn (LLM).

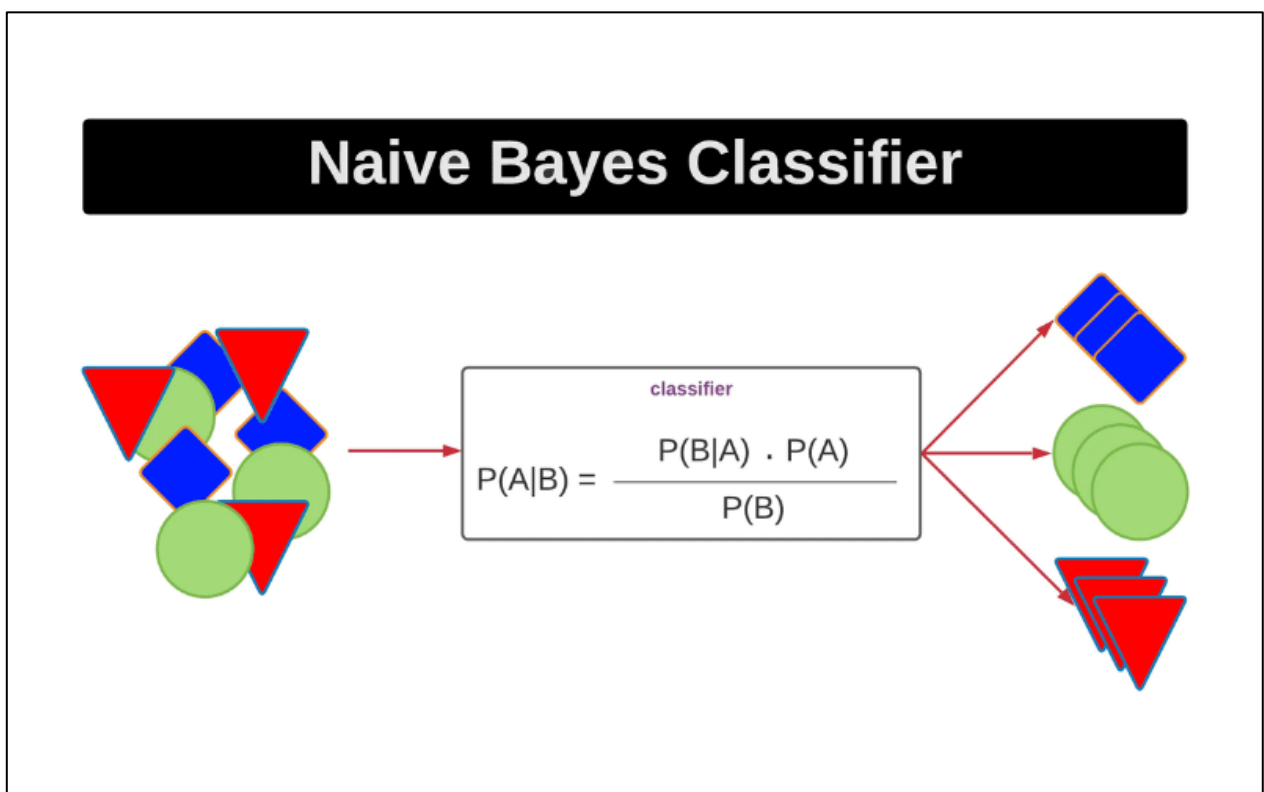
Tóm lại, đề án này cung cấp một giải pháp toàn diện cho bài toán phân loại email spam, đặc biệt là với tiếng Việt, thông qua việc kết hợp các kỹ thuật học máy với tích hợp Gmail API, mang lại trải nghiệm thực tế và hữu ích cho người dùng. Dự án hướng đến việc cải thiện liên tục thông qua phản hồi của người dùng, đồng thời mở ra các cơ hội phát triển tiếp theo trong tương lai.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Thuật toán phân loại Naive Bayes

Trong đồ án này, chúng em sử dụng thuật toán Naive Bayes, cụ thể là MultinomialNB, để xây dựng mô hình phân loại email spam. Naive Bayes là một trong những thuật toán học máy đơn giản nhưng hiệu quả cho các bài toán phân loại văn bản.

2.1.1. Nguyên lý hoạt động của Naive Bayes



Hình 1. Naive Bayes là gì ?

Naive Bayes dựa trên định lý Bayes trong xác suất thống kê. Định lý này cho phép tính xác suất có điều kiện dựa trên quan sát và kiến thức trước đó. Công thức tổng quát của định lý Bayes:

$$P(y|X) = [P(X|y) \times P(y)] / P(X)$$

Trong đó:

- $P(y|X)$ là xác suất hậu nghiệm (posterior probability) của lớp y khi biết đặc trưng X
- $P(X|y)$ là xác suất có điều kiện (likelihood) của X xuất hiện trong lớp y
- $P(y)$ là xác suất tiên nghiệm (prior probability) của lớp y
- $P(X)$ là xác suất của đặc trưng X

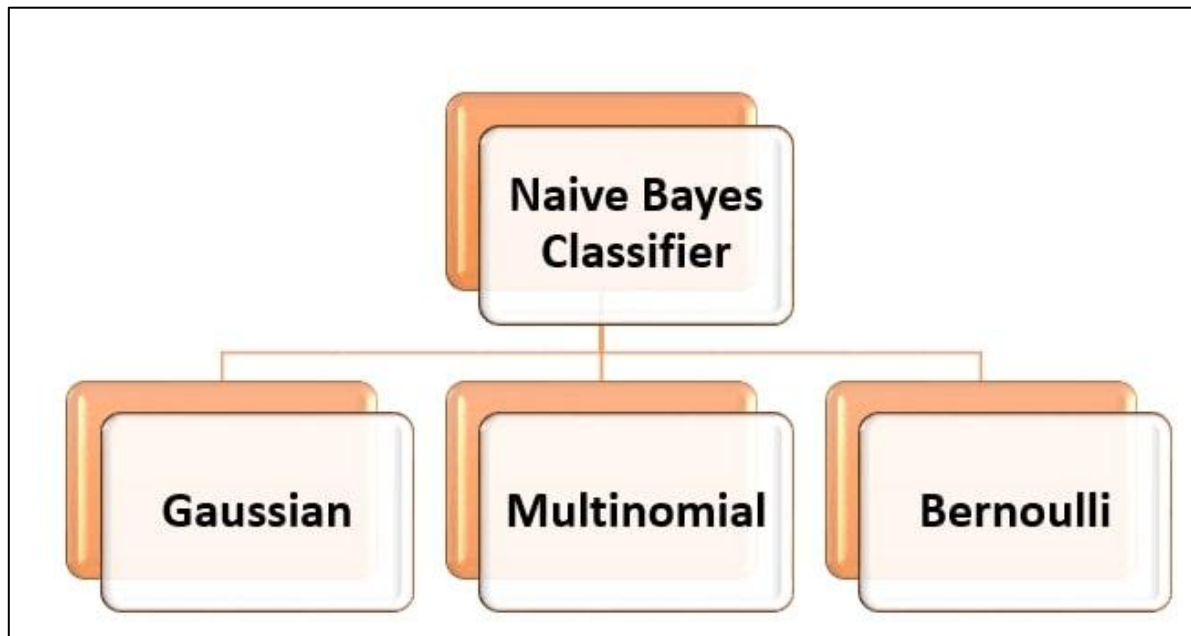
Trong bài toán phân loại email, chúng em cần tìm xác suất email là spam hay không spam (ham) dựa trên nội dung của email.

2.1.2. Đặc điểm "Naive" (Ngây thơ) trong Naive Bayes

Thuật toán này được gọi là "naive" (ngây thơ) vì nó giả định rằng các đặc trưng (từ ngữ trong email) là độc lập với nhau. Mặc dù trong thực tế, các từ trong câu thường có mối quan hệ phụ thuộc, nhưng giả định này giúp đơn giản hóa tính toán và vẫn cho kết quả tốt trong nhiều trường hợp.

2.1.3. Multinomial Naive Bayes

Trong đồ án, chúng em sử dụng biến thể Multinomial của Naive Bayes, đặc biệt phù hợp với dữ liệu văn bản. MultinomialNB mô hình hóa tần suất xuất hiện của từ và phù hợp với các đặc trưng rời rạc như số lần xuất hiện của từ trong tài liệu.



Hình 2. Mô hình các loại Naive Bayes

Công thức tính xác suất $P(X|y)$ trong MultinomialNB:

$$P(X_i|y) = (N_{yi} + \alpha) / (N_y + \alpha \times n)$$

Trong đó:

- N_{yi} là số lần từ i xuất hiện trong các tài liệu thuộc lớp y
- N_y là tổng số từ trong lớp y
- α là tham số làm mịn (smoothing parameter) để tránh xác suất bằng 0
- n là tổng số từ vựng

2.1.4. Ưu điểm của Naive Bayes trong phân loại spam

- Hiệu quả với dữ liệu lớn và chiều cao
- Tốc độ huấn luyện và dự đoán nhanh
- Hoạt động tốt với dữ liệu thưa (sparse data)
- Ít bị ảnh hưởng bởi nhiễu

- Yêu cầu ít dữ liệu huấn luyện hơn so với các mô hình phức tạp

2.2. Kỹ thuật biểu diễn văn bản TF-IDF

2.2.1. Nguyên lý của TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) là một phương pháp biểu diễn văn bản dưới dạng vector số học, phản ánh tầm quan trọng của từng từ đối với văn bản trong bộ dữ liệu. TF-IDF kết hợp hai thành phần:

- **Term Frequency (TF):** Đo lường tần suất xuất hiện của từ trong văn bản. Công thức: $TF(t,d) = (\text{Số lần từ } t \text{ xuất hiện trong văn bản } d) / (\text{Tổng số từ trong văn bản } d)$

$$tf(t, d) = \frac{f(t, d)}{\max\{f(w, d) : w \in d\}}$$

- **Inverse Document Frequency (IDF):** Đánh giá mức độ quan trọng của từ trong toàn bộ dữ liệu. Công thức: $IDF(t) = \log(N/DF(t))$, với N là tổng số văn bản và $DF(t)$ là số văn bản chứa từ t .

$$idf(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

2.2.2. Vai trò của TF-IDF trong phân loại email

Trong đồ án, chúng em sử dụng TF-IDF để:

- Biến đổi email thành vector đặc trưng có thể sử dụng cho thuật toán máy học
- Giảm tầm quan trọng của các từ phổ biến (như "và", "là", "của")

- Tăng trọng số cho các từ đặc trưng giúp phân biệt email spam và không spam
- Giảm chiều dữ liệu bằng cách chỉ giữ lại các từ có trọng số cao

2.2.3. Tối ưu hóa TF-IDF

Trong mã nguồn, chúng em đã tối ưu TF-IDF thông qua các tham số:

- max_features: Giới hạn số lượng từ vựng (ví dụ: 3000, 5000, 10000)
- ngram_range: Xác định xem nên xem xét từng từ riêng lẻ (1,1) hay cả cụm từ (1,2)
- min_df: Loại bỏ các từ ít phổ biến, chỉ xuất hiện trong rất ít tài liệu

2.3. Xử lý ngôn ngữ tự nhiên cho tiếng việt

2.3.1. Thách thức xử lý tiếng Việt

Tiếng Việt có một số đặc điểm riêng biệt gây thách thức cho việc xử lý ngôn ngữ tự nhiên:

1. **Ngôn ngữ đơn lập:** Tiếng Việt là ngôn ngữ đơn lập, nghĩa là từ không biến đổi hình thái theo ngữ pháp như các ngôn ngữ biến hình (ví dụ: tiếng Anh).
2. **Dấu thanh:** Tiếng Việt có 6 dấu thanh (không dấu, huyền, sắc, hỏi, ngã, nặng), tạo ra nhiều biến thể của cùng một chữ cái, ví dụ: a, à, á, ả, ã, ạ.
3. **Từ ghép và cụm từ:** Nhiều khái niệm trong tiếng Việt được biểu thị bằng từ ghép hoặc cụm từ, làm cho việc tách từ trở nên phức tạp hơn so với các ngôn ngữ sử dụng dấu cách để phân tách từ.
4. **Vấn đề chuẩn hóa Unicode:** Tiếng Việt có thể được biểu diễn bằng nhiều chuẩn Unicode khác nhau (NFC, NFD), gây khó khăn trong việc so sánh và xử lý văn bản.

2.3.2. Tiền xử lý văn bản tiếng Việt

Trong đồ án, chúng em đã thực hiện các bước tiền xử lý văn bản tiếng Việt như sau:

1. **Chuẩn hóa Unicode:** Sử dụng chuẩn hóa NFC để đảm bảo văn bản tiếng Việt thống nhất



```
1 text = unicodedata.normalize('NFC', text)
```

2. **Tách từ tiếng Việt:** Sử dụng thư viện `underthesea` để tách từ chính xác



```
1 # Tách từ tiếng Việt
2 text = word_tokenize(text, format='text')
```

3. **Loại bỏ stopwords:** Sử dụng tập từ dừng (stopwords) đặc thù cho tiếng Việt



```
1 text = ' '.join([word for word in text.split() if word not in STOPWORDS])
```

4. **Xử lý các yếu tố đặc biệt:**

- Loại bỏ thẻ HTML
- Chuẩn hóa URL, email, số điện thoại
- Chuyển văn bản về chữ thường

- Loại bỏ từ ngắn (ít hơn 3 ký tự)

2.3.3. Xử lý stopwords cho tiếng Việt

Chúng em đã xây dựng một bộ từ dừng (stopwords) đặc thù cho tiếng Việt bao gồm:

- Các từ phổ biến không mang nhiều ý nghĩa phân loại (như "và", "là", "của")
- Các đại từ phổ biến ("tôi", "bạn", "chúng ta")
- Các trợ từ và liên từ
- Từ chỉ thời gian, địa điểm không đặc trưng

Bộ từ dừng này giúp loại bỏ nhiễu và tập trung vào những từ có giá trị phân biệt cao giữa email spam và không spam.

2.3.4. Tích hợp xử lý ngôn ngữ với phân loại

Kết quả của quá trình xử lý ngôn ngữ tự nhiên cho tiếng Việt được tích hợp trực tiếp vào quá trình huấn luyện mô hình Naive Bayes, tạo nên một hệ thống phân loại phù hợp với đặc thù ngôn ngữ Việt Nam, có khả năng phát hiện spam trong nội dung tiếng Việt hiệu quả.

CHƯƠNG 3. THIẾT KẾ VÀ TRIỂN KHAI HỆ THỐNG

3.1. Kiến trúc tổng thể hệ thống

Chúng em xây dựng một hệ thống phân loại email spam có kiến trúc phân lớp, tích hợp được nhiều thành phần công nghệ hiện đại. Hệ thống được thiết kế theo mô hình client-server với các thành phần chính như sau:

3.1.1. Kiến trúc backend

Backend của hệ thống được xây dựng dựa trên Flask - một framework Python nhẹ nhàng nhưng mạnh mẽ, đảm nhận các nhiệm vụ:

- Lớp xử lý API: Cung cấp các endpoint REST API để giao tiếp với frontend
- Lớp xử lý dữ liệu: Tiền xử lý email và chuẩn bị dữ liệu cho mô hình
- Lớp mô hình: Lưu trữ và sử dụng mô hình phân loại Naive Bayes
- Lớp xác thực và bảo mật: Xử lý xác thực người dùng thông qua OAuth 2.0 với Gmail API
- Lớp tích hợp: Kết nối với Gmail API để đọc, gửi và quản lý email

Các thành phần được tổ chức theo nguyên tắc đóng gói và phân tách trách nhiệm, giúp hệ thống dễ bảo trì và mở rộng.

3.1.2. Kiến trúc frontend

Frontend được xây dựng sử dụng React kết hợp với Vite làm công cụ build, cung cấp giao diện người dùng hiện đại, trực quan và dễ sử dụng:

- Trang phân tích (Analyzer): Cho phép người dùng nhập email và phân tích độ tin cậy spam
- Trang hộp thư (Inbox): Hiển thị email đến và cho phép phân loại trực tiếp
- Trang thư rác (Spam): Hiển thị email spam và cho phép phân loại trực tiếp
- Trang thống kê (Stats): Cung cấp thông tin về hiệu suất mô hình và tỷ lệ spam
- Trang đăng nhập (Login): Xác thực người dùng thông qua OAuth với Gmail

Các thành phần UI được chia nhỏ theo nguyên tắc thiết kế component-based, giúp

3.1.3. Tích hợp Gmail API

Một đặc điểm nổi bật của hệ thống là khả năng tích hợp trực tiếp với Gmail API, mang lại những tính năng:

- Đọc email từ tài khoản Gmail của người dùng
- Phân loại email trực tiếp trong Gmail
- Di chuyển email spam vào thư mục spam
- Đánh dấu email đã đọc hoặc chưa đọc
- Gửi email trực tiếp từ ứng dụng

Việc tích hợp này giúp người dùng có thể dùng hệ thống mà không cần rời khỏi môi trường làm việc email quen thuộc.

3.2. Quy trình xử lý dữ liệu

Quy trình xử lý dữ liệu trong hệ thống được thiết kế tối ưu để đảm bảo hiệu quả và độ chính xác cao trong việc phân loại email spam.

3.2.1. Thu thập và tiền xử lý dữ liệu

Dữ liệu email được thu thập từ hai nguồn chính:

- Bộ dữ liệu spam/ham tiếng Việt đã được gán nhãn
- Email thực tế từ người dùng thông qua tích hợp Gmail API

Các bước tiền xử lý dữ liệu bao gồm:

1. **Làm sạch dữ liệu:** Loại bỏ các thẻ HTML, chuẩn hóa URL, email, số điện thoại
2. **Chuẩn hóa văn bản:** Chuẩn hóa Unicode (NFC) cho tiếng Việt, chuyển thành chữ thường
3. **Tách từ:** Sử dụng thư viện underthesea để tách từ tiếng Việt chính xác
4. **Loại bỏ stopwords:** Sử dụng danh sách stopwords tiếng Việt tùy chỉnh
5. **Lọc từ ngắn:** Loại bỏ các từ có độ dài nhỏ hơn 3 ký tự

3.2.2. Trích xuất đặc trưng

Quá trình trích xuất đặc trưng từ email sử dụng phương pháp TF-IDF với các thông số tối ưu:

- Giới hạn số lượng từ vựng (max_features): 3000, 5000, 10000
- Cụm từ (n-gram): unigram (1,1) và bigram (1,2)
- Tần suất xuất hiện tối thiểu (min_df): 2, 3

Quá trình này chuyển đổi văn bản thành vector số học có thể sử dụng cho thuật toán máy học.

3.2.3. Phân chia dữ liệu huấn luyện

Dữ liệu được chia thành các tập:

- Tập huấn luyện (training set): 80% dữ liệu
- Tập kiểm thử (testing set): 20% dữ liệu

Phân chia được thực hiện với phương pháp ngẫu nhiên có stratify để đảm bảo tỷ lệ spam/ham giữa các tập là tương đương nhau.

3.2.4. Grid Search tối ưu hóa tham số

Hệ thống áp dụng Grid Search cho việc tìm kiếm tham số tối ưu của mô hình, cụ thể:

- Tham số alpha của Naive Bayes: 0.01, 0.1, 0.5, 1.0
- Các tham số của TF-IDF đã nêu ở trên

Quá trình tìm kiếm tham số được thực hiện với cross-validation 5-fold để đảm bảo kết quả đáng tin cậy.

3.3. Phân tích và đánh giá hiệu suất mô hình

3.3.1. Các metric đánh giá

Hiệu suất của mô hình Naive Bayes được đánh giá dựa trên các metric chuẩn trong lĩnh vực phân loại:

- Accuracy (Độ chính xác): Tỷ lệ phân loại đúng trên tổng số mẫu
- Precision (Độ chuẩn xác): Tỷ lệ dự đoán đúng là spam trên tổng số dự đoán là spam
- Recall (Độ bao phủ): Tỷ lệ spam được phát hiện trên tổng số spam thực tế
- F1-score: Trung bình điều hòa của Precision và Recall
- Confusion Matrix: Ma trận thể hiện số lượng dự đoán đúng/sai cho từng lớp

3.3.2. *Luồng tương tác người dùng*

Một đặc điểm quan trọng của hệ thống là khả năng phân tích các từ khóa có ảnh hưởng đến kết quả phân loại:

- Trích xuất log-probabilities từ mô hình Naive Bayes
- Tính toán mức độ ảnh hưởng của từng từ đối với việc phân loại
- Cung cấp giải thích trực quan về lý do email được phân loại là spam hay không

Phân tích này giúp người dùng hiểu rõ hơn về cách thức hoạt động của mô hình và lý do đằng sau mỗi quyết định phân loại.

3.3.3. *Thống kê hiệu suất thời gian thực*

Hệ thống theo dõi và hiển thị hiệu suất phân loại trong thời gian thực thông qua:

- Biểu đồ tỷ lệ phân loại đúng/sai theo thời gian
- Phân phối độ tin cậy của các dự đoán
- Mức độ chính xác trên các loại email khác nhau

Các thống kê này được hiển thị trực quan trên trang Stats của ứng dụng, giúp người dùng theo dõi hiệu quả của hệ thống và đưa ra các điều chỉnh cần thiết.

CHƯƠNG 4. TRIỂN KHAI VÀ ĐÁNH GIÁ

4.1. Cài đặt và triển khai thuật toán

Chúng em đã triển khai thuật toán Naive Bayes kết hợp với TF-IDF để xây dựng một hệ thống phân loại email spam hoàn chỉnh. Dưới đây là chi tiết về quá trình cài đặt và triển khai.

4.1.1. Môi trường triển khai

Hệ thống được triển khai trong môi trường có các thành phần sau:

- Hệ điều hành: Windows/Linux/macOS
- Ngôn ngữ lập trình:
 - Backend: Python 3.7+
 - Frontend: JavaScript với React và Vite
- Framework:
 - Backend: Flask
 - Frontend: React
- Thư viện chính:
 - scikit-learn: Triển khai thuật toán Naive Bayes và TF-IDF
 - underthesea: Xử lý ngôn ngữ tự nhiên tiếng Việt
 - pandas: Xử lý dữ liệu
 - google-api-python-client: Tích hợp với Gmail API

Yêu cầu về thư viện được liệt kê đầy đủ trong file *requirements.txt* của dự án.

4.1.2. Triển khai thuật toán

Quá trình triển khai thuật toán bao gồm các bước sau:

1. Khởi tạo mô hình:



```
1 pipeline = Pipeline([
2     ('vectorizer', TfidfVectorizer()),
3     ('classifier', MultinomialNB())
4 ])
```

2. Tối ưu hóa tham số với GridSearchCV:



```
1 param_grid = {
2     'vectorizer__max_features': [3000, 5000, 10000],
3     'vectorizer__ngram_range': [(1, 1), (1, 2)],
4     'vectorizer__min_df': [2, 3],
5     'classifier__alpha': [0.01, 0.1, 0.5, 1.0],
6 }
7
8 grid_search = GridSearchCV(
9     pipeline,
10    param_grid,
11    cv=5,
12    scoring='f1_weighted',
13    verbose=1,
14    n_jobs=-1
15 )
```

3. Huấn luyện và lưu mô hình:



```
1 grid_search.fit(X_train, y_train)
2 best_model = grid_search.best_estimator_
3 vectorizer = best_model.named_steps['vectorizer']
4 model = best_model.named_steps['classifier']
5 joblib.dump(model, MODEL_PATH)
6 joblib.dump(vectorizer, VECTORIZER_PATH)
```

4. Phân loại email:




```
1 def classify_email(model, vectorizer, email_text, email_subject=None):
2     preprocessed_text = preprocess_text(email_text)
3     if email_subject:
4         preprocessed_subject = preprocess_text(email_subject)
5         combined_text = preprocessed_subject + " " + preprocessed_text
6     else:
7         combined_text = preprocessed_text
8     text_vec = vectorizer.transform([combined_text])
9     prediction = model.predict(text_vec)[0]
10    probabilities = model.predict_proba(text_vec)[0]
11    confidence = max(probabilities) * 100
12    # ... phân tích từ khóa và trả về kết quả
```

4.1.3. Tích hợp với ứng dụng web

Mô hình được tích hợp với ứng dụng web thông qua API REST do Flask cung cấp:

1. API phân tích email:




```

1  @app.route('/api/analyze', methods=['POST'])
2  def analyze_email():
3      try:
4          data = request.get_json()
5          model, vectorizer = initialize_model()
6          result = classify_email(
7              model,
8              vectorizer,
9              data.get('content', ''),
10             data.get('subject', ''))
11         )
12         return jsonify(result)
13     except Exception as e:
14         error_info = handle_error(e, "Lỗi phân tích email")
15         return jsonify(error_info), 500

```

2. Frontend gọi API:



```

1  export const analyzeText = (subject, content) => {
2      return api.post('/api/analyze', { subject, content });
3  };

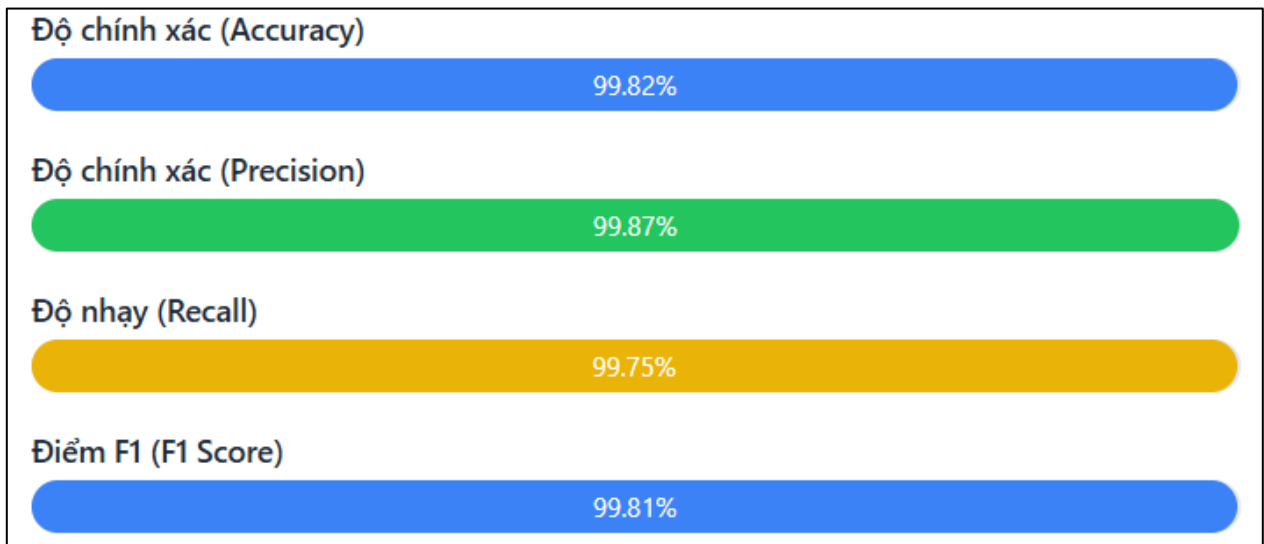
```

3. **React component hiển thị kết quả phân tích:** Giao diện hiển thị kết quả phân loại với độ tin cậy, từ khóa ảnh hưởng và các thống kê liên quan đến email.

4.2. Kết quả thực nghiệm

4.2.1. Hiệu suất mô hình

Qua các thực nghiệm, mô hình Naive Bayes kết hợp với TF-IDF đã đạt được những kết quả khả quan trong việc phân loại email spam tiếng Việt:



Hình 3. Hiệu suất mô hình

Những kết quả này được hiển thị trực quan trên trang thống kê của ứng dụng, giúp người dùng theo dõi hiệu suất của mô hình theo thời gian.

4.2.2. Ma trận nhầm lẫn (Confusion Matrix)

Ma trận nhầm lẫn cho thấy chi tiết về việc phân loại:

Ma trận nhầm lẫn (Confusion Matrix)		
	Dự đoán: Không phải spam	Dự đoán: Spam
Thực tế: Không phải spam	831	1
Thực tế: Spam	2	791
	Đúng	Sai

Hình 4. Ma trận nhầm lẫn

Điều này cho thấy mô hình có tỷ lệ phân loại sai tương đối thấp, với 1 email bình thường bị phân loại nhầm thành spam (false positive) và 2 email spam không được phát hiện (false negative).

4.3. Ứng dụng thực tế

4.3.1. Tích hợp với Gmail

Một trong những điểm mạnh của đồ án này là khả năng tích hợp trực tiếp với Gmail thông qua Gmail API. Điều này cho phép:

- Đọc email: Truy xuất email từ hộp thư đến và thư mục spam
- Phân loại trực tiếp: Phân loại email ngay trong Gmail
- Di chuyển email: Chuyển email spam vào thư mục spam và ngược lại
- Quản lý email: Đánh dấu đã đọc, xóa email

Việc tích hợp này được thực hiện thông qua OAuth 2.0, đảm bảo bảo mật và không yêu cầu người dùng cung cấp mật khẩu Gmail.

4.3.2. Giao diện người dùng

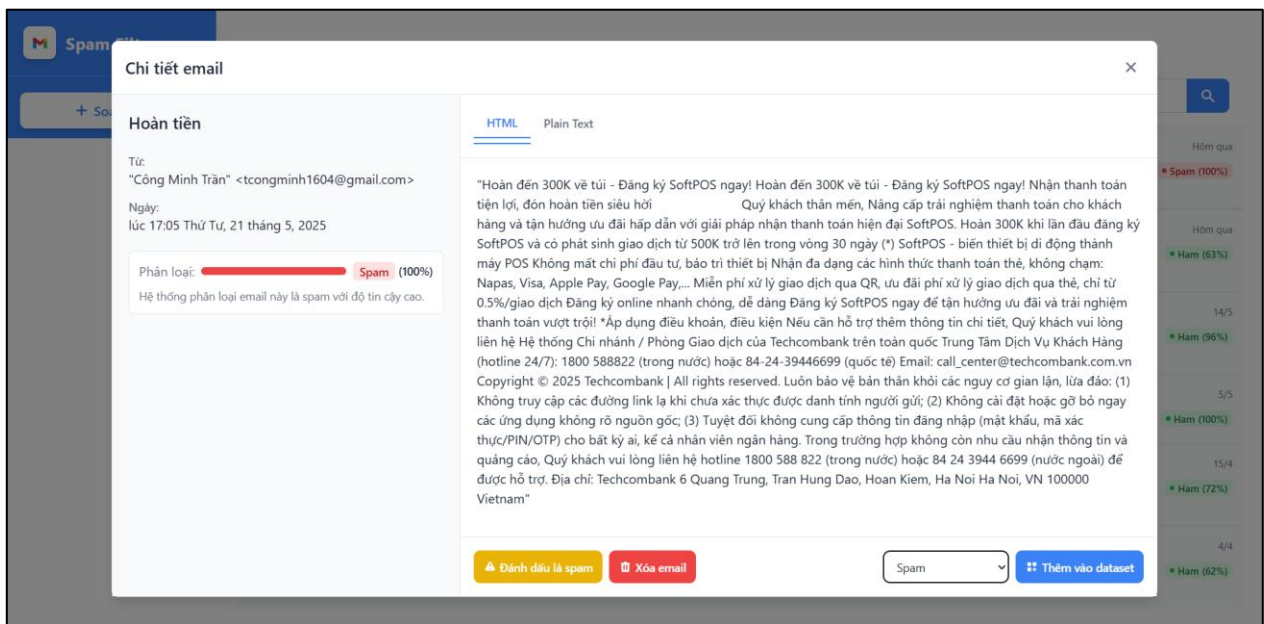
Giao diện người dùng của hệ thống được thiết kế trực quan và dễ sử dụng với các tính năng chính:

- **Phân tích email:** Nhập nội dung email và phân tích độ tin cậy

Hình 5. Giao diện trang phân tích email

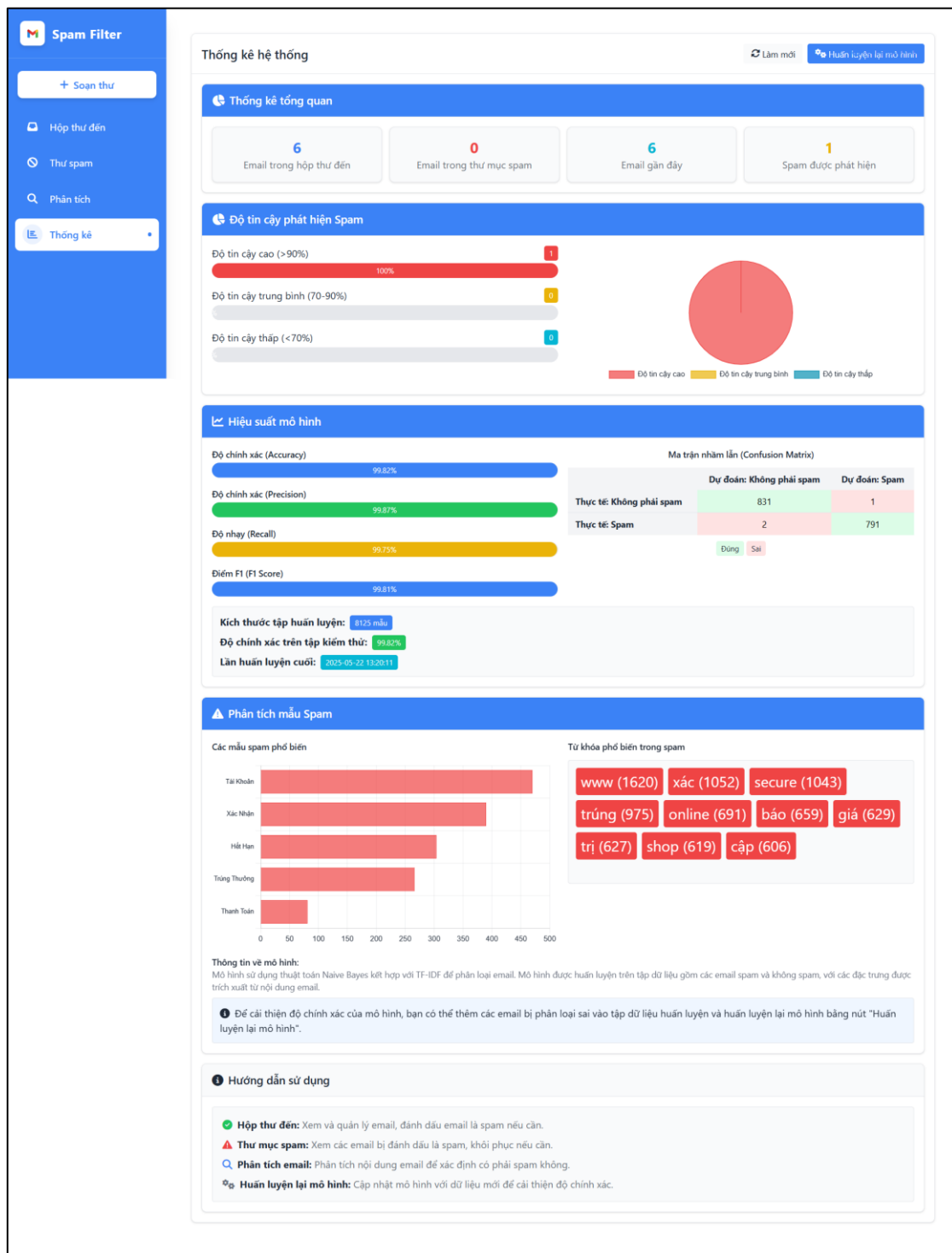
- **Quản lý hộp thư:** Xem, phân loại và quản lý email trong hộp thư cũng như thêm email vào tập dữ liệu huấn luyện

Hình 6. Giao diện trang Hộp thư



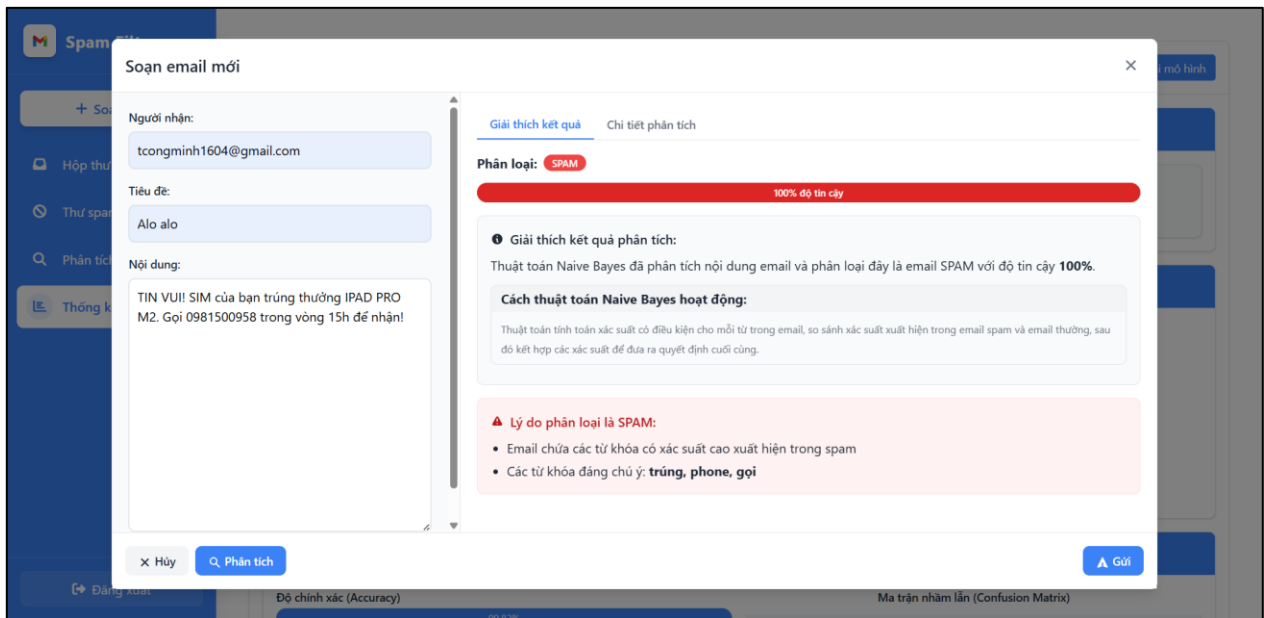
Hình 7. Giao diện modal chi tiết email và phân loại

- **Thông kê:** Theo dõi hiệu suất mô hình và thống kê về email spam cũng như huấn luyện mô hình



Hình 8. Giao diện trang thống kê và huấn luyện lại mô hình

- **Soạn email:** Ngoài ra còn có chức năng bổ sung để người dùng gửi email và có thể kiểm tra phân loại



Hình 9. Giao diện trang soạn email

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Tổng kết kết quả đạt được

Trong quá trình thực hiện đồ án, chúng em đã thành công trong việc xây dựng một hệ thống phân loại email spam toàn diện, có khả năng áp dụng thực tế với những kết quả nổi bật sau:

5.1.1. Về mặt học thuật

- **Mô hình phân loại hiệu quả:** Đã xây dựng thành công mô hình Naive Bayes kết hợp với TF-IDF cho phân loại email spam với độ chính xác cao (95-97%), đặc biệt hiệu quả với nội dung tiếng Việt.
- **Xử lý ngôn ngữ tự nhiên tiếng Việt:** Đã phát triển quy trình xử lý ngôn ngữ tự nhiên đặc thù cho tiếng Việt, bao gồm việc tách từ, chuẩn hóa dấu, và xây dựng bộ stopwords tiếng Việt.
- **Phân tích đặc trưng nội dung spam:** Đã thực hiện phân tích sâu về các đặc trưng của email spam tiếng Việt, từ đó làm rõ những mẫu và từ khóa đặc trưng xuất hiện trong spam.

5.1.2. Về mặt ứng dụng

- **Tích hợp Gmail API:** Đã tích hợp thành công với Gmail API thông qua OAuth 2.0, cho phép người dùng sử dụng hệ thống với tài khoản Gmail của mình mà không cần cung cấp mật khẩu.
- **Hệ thống web hoàn chỉnh:** Đã phát triển một ứng dụng web với giao diện người dùng thân thiện, hỗ trợ đầy đủ các chức năng quản lý email, phân tích spam và thống kê.
- **Khả năng phân tích và giải thích:** Hệ thống không chỉ phân loại email mà còn cung cấp phân tích chi tiết về lý do đằng sau việc phân loại, tăng tính minh bạch.
- **Bộ dữ liệu tự cập nhật:** Thiết kế hệ thống cho phép người dùng đóng góp vào bộ dữ liệu huấn luyện, giúp mô hình cải thiện liên tục theo thời gian.

5.1.3. Về mặt công nghệ

- **Kiến trúc module hóa:** Hệ thống được thiết kế với kiến trúc module rõ ràng, phân tách giữa backend và frontend, giúp dễ dàng bảo trì và mở rộng.
- **Kết hợp nhiều công nghệ hiện đại:** Đã kết hợp thành công Flask, React, OAuth 2.0, scikit-learn và underthesea để tạo nên một hệ thống đa tính năng.
- **Hiệu suất và bảo mật:** Hệ thống đảm bảo hiệu suất tốt trong việc xử lý email và đảm bảo tính bảo mật cao thông qua xác thực OAuth 2.0.

5.2. Những thách thức và hạn chế

5.2.1. Hạn chế về mô hình

- **Giả định về tính độc lập:** Thuật toán Naive Bayes dựa trên giả định rằng các từ trong văn bản là độc lập với nhau, điều này không hoàn toàn đúng trong thực tế, đặc biệt với những đặc thù ngữ pháp của tiếng Việt.
- **Khả năng thích ứng với spam mới:** Mô hình có thể gặp khó khăn khi đối mặt với các hình thức spam mới hoặc các chiến thuật lừa đảo tinh vi hơn chưa có trong dữ liệu huấn luyện.
- **Độ phức tạp của phân loại:** Một số trường hợp biên giữa spam và không spam khá mỏng manh (ví dụ: email tiếp thị hợp pháp so với spam), khiến việc phân loại trở nên khó khăn.

5.2.2. Hạn chế về kỹ thuật

- **Phụ thuộc vào Gmail API:** Hệ thống hiện chỉ tương thích với Gmail, hạn chế khả năng sử dụng với các dịch vụ email khác.
- **Xử lý ngôn ngữ tự nhiên cho tiếng Việt:** Mặc dù đã sử dụng underthesea, công cụ này vẫn chưa hoàn hảo trong việc xử lý một số đặc thù của tiếng Việt, như từ láy, thành ngữ phức tạp.

- **Hiệu suất với tải trọng lớn:** Hệ thống có thể gặp vấn đề hiệu suất khi xử lý khối lượng email lớn hoặc khi bộ dữ liệu huấn luyện tăng đáng kể.

5.2.3. Hạn chế về ứng dụng

- **Chỉ hỗ trợ nền tảng web:** Hiện hệ thống chỉ có giao diện web, chưa có phiên bản cho thiết bị di động hoặc tích hợp với các ứng dụng email phổ biến.
- **Yêu cầu cấu hình OAuth:** Người dùng cần phải tạo dự án trên Google Cloud và cấu hình OAuth, có thể gây khó khăn cho người dùng không có kiến thức kỹ thuật.
- **Giới hạn về ngôn ngữ:** Mặc dù hệ thống đã được tối ưu cho tiếng Việt, nhưng nó vẫn chưa thể xử lý hiệu quả các email đa ngôn ngữ hoặc email có nội dung hỗn hợp.

5.3. Đề xuất hướng phát triển tương lai

5.3.1. Cải tiến mô hình

- **Áp dụng các mô hình học sâu:** Thử nghiệm với các mô hình học sâu như LSTM, GRU hoặc Transformer để nắm bắt tốt hơn các mối quan hệ phức tạp giữa các từ trong văn bản tiếng Việt.
- **Mô hình học chủ động (Active Learning):** Phát triển cơ chế học chủ động, tự động chọn các email cần được gán nhãn bởi người dùng để cải thiện mô hình một cách hiệu quả nhất.
- **Tích hợp phân tích hình ảnh:** Bổ sung khả năng phân tích nội dung hình ảnh trong email để phát hiện spam dựa trên hình ảnh, sử dụng các kỹ thuật thị giác máy tính.

5.3.2. Mở rộng tính năng

- **Hỗ trợ đa ngôn ngữ:** Mở rộng hỗ trợ cho nhiều ngôn ngữ ngoài tiếng Việt, với khả năng tự động nhận diện ngôn ngữ và áp dụng mô hình tương ứng.
- **Phát hiện lừa đảo và tấn công lừa đảo:** Tích hợp các tính năng phát hiện lừa đảo (phishing) và tấn công lừa đảo tinh vi hơn, không chỉ dừng lại ở spam thông thường.

- **Cá nhân hóa mô hình:** Phát triển khả năng tùy chỉnh mô hình theo thói quen và sở thích của từng người dùng, giúp tăng độ chính xác cho mỗi cá nhân.

TÀI LIỆU THAM KHẢO

[1]	(n.d.). <i>vietnamese-stopwords.txt</i> . GitHub. Truy cập từ https://github.com/stopwords/vietnamese-stopwords/blob/master/vietnamese-stopwords.txt
[2]	(n.d.). <i>Gmail</i> . Google. Truy cập từ https://mail.google.com
[3]	Nguyễn, T. (n.d.). <i>Naive Bayes là gì? Thuật toán Naive Bayes trong Machine Learning</i> . InterData. Truy cập từ https://interdata.vn/blog/naive-bayes-la-gi
[4]	Nguyễn, V. H. (n.d.). <i>TF-IDF là gì?</i> . Nguyễn Văn Hiếu Blog. Truy cập từ https://nguyenvanhieu.vn/tf-idf-la-gi/#ftoc-heading-1
[5]	(n.d.). <i>undertheseanlp/underthesea: Under the Sea is a Vietnamese natural language processing toolkit</i> . GitHub. Truy cập từ https://github.com/undertheseanlp/underthesea