

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

-----🙞🕮🙜-----

A yellow and red logo

Description automatically generated

**BÀI TẬP LỚN**

**MÔN:TRÍ TUỆ NHÂN TẠO**

**ĐỀ TÀI:XÂY DỰNG HỆ THỐNG PHÂN LOẠI**

**THƯ RÁC DỰA TRÊN GIẢI THUẬT NAIVE BAYES**

**Giảng viên hướng dẫn: Trần Thanh Huân**

**Lớp học phần: 2024IT6094001**

**Nhóm: 11**

**Sinh viên thực hiện: Nguyễn Đức Trường 2023605862**

**Dương Kim Vũ 2023605127**

**Đào Xuân Thắng 2023605466**

**Nguyễn Cường Thịnh 2023606918**

***Hà Nội, 05/20***

# LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi lời cảm ơn trân thành đến thầy **Trần Thanh Huân**. Trong quá trình học tập và thực hiện đề tài, chúng em đã nhận được sự quan tâm, hướng dẫn tận tình từ thầy. Những gì chúng em nhận được không chỉ dừng lại ở kiến thức môn học mà nhiều hơn thế đó là những lời khuyên, chia sẻ thực tế từ thầy.

Để hoàn thành được đề tài này, chúng em đã nghiên cứu, áp dụng những kiến thức được học trên lớp cùng với các nguồn tài liệu trên Internet và cả những trải nghiệm của bản thân đối với đề tài lần này. Chúng em rất mong sẽ nhận được những lời nhận xét, góp ý từ thầy, cô và các bạn đọc để đề tài của chúng em để có thể hoàn thiện hơn nữa.

Vì thời gian còn hạn chế chúng em chưa có cơ hội cụ thể hóa hết các ý tưởng của mình với đề tài này, em rất mong sẽ được quý thầy cô và các bạn đón nhận cũng như đưa ra những lời góp ý để đề tài của chúng em hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc200645515)

[DANH MỤC HÌNH ẢNH 3](#_Toc200645516)

[MỞ ĐẦU 4](#_Toc200645517)

[1.Lí do chọn đề tài: 4](#_Toc200645518)

[2. Mục đích nghiên cứu: 5](#_Toc200645519)

[3. Phương pháp nghiên cứu: 6](#_Toc200645520)

[4.Bố cục của bài báo cáo: 6](#_Toc200645521)

[CHƯƠNG I. TỔNG QUAN VỀ KĨ THUẬT PHÂN LOẠI THƯ RÁC 6](#_Toc200645522)

[1.1 . Khái niệm thư rác 6](#_Toc200645523)

[1.2. Tác hại của thư rác 7](#_Toc200645524)

[1.3. Tổng quan về một số kĩ thuật phân loại thư rác: 7](#_Toc200645525)

[1.4. Những thách thức trong việc phân loại thư rác: 8](#_Toc200645526)

[1.5. Tổng kết chương 1 9](#_Toc200645527)

[CHƯƠNG II : PHÂN TÍCH THUẬT TOÁN NAIVE BAYES TRONG PHÂN LOẠI THƯ RÁC 9](#_Toc200645528)

[2.1. Định lí Bayes: 10](#_Toc200645529)

[2.2. Mô tả thuật toán Naive Bayes: 10](#_Toc200645530)

[2.3.Các bước triển khai thuật toán Naïve Bayes: 11](#_Toc200645531)

[2.3.1. Thu thập và tiền xử lý dữ liệu 11](#_Toc200645532)

[2.3.2. Xây dựng từ điển và tính xác suất 12](#_Toc200645533)

[2.3.3 Tính xác suất điều kiện 13](#_Toc200645534)

[2.3.4 Áp dụng định lý Bayes 13](#_Toc200645535)

[2.3.5 Quyết định phân loại 15](#_Toc200645536)

[2.3.6. Đánh giá và tinh chỉnh mô hình 15](#_Toc200645537)

[2.4. Phân loại 16](#_Toc200645538)

[2.5. Ưu điểm và hạn chế của thuật toán Naive Bayes 18](#_Toc200645539)

[2.5.1. Ưu điểm 18](#_Toc200645540)

[2.5.2. Hạn chế 18](#_Toc200645541)

[CHƯƠNG 3: THIẾT KẾ , TRIỂN KHAI HỆ THỐNG PHÂN LOẠI THƯ RÁC 19](#_Toc200645542)

[3.1. Ngôn ngữ và công cụ triển khai 19](#_Toc200645543)

[3.1.1. Python 19](#_Toc200645544)

[3.1.2. Môi trường Visual Studio Code (VS Code) 20](#_Toc200645545)

[3.1.3.Các thư viện Python sử dụng chính: 20](#_Toc200645546)

[3.2. Chuẩn bị dữ liệu huấn luyện 20](#_Toc200645547)

[3.3. Xây dựng mô hình Naive Bayes và triển khai chương trình (modules.py) 20](#_Toc200645548)

[3.3.1. Hàm huấn luyện train\_model 20](#_Toc200645549)

[3.3.2. Hàm phân loại 1 tin nhắn classify\_test 23](#_Toc200645550)

[3.4. Xây dựng giao diện người dùng (giaodien.py) 25](#_Toc200645551)

[3.4.1. Mô tả giao diện người dùng 25](#_Toc200645552)

[3.4.2. Thiết kế giao diện 25](#_Toc200645553)

[3.4.3. Kết nối giữa giao diện và modules 27](#_Toc200645554)

[3.5. Kết quả thực nghiệm 29](#_Toc200645555)

[3.6 Kết luận chương 3 30](#_Toc200645556)

[TỔNG KẾT 32](#_Toc200645557)

[1. Kết quả đạt được: 32](#_Toc200645558)

[2. Hạn chế 32](#_Toc200645559)

[3. Đóng góp thực tiễn 32](#_Toc200645560)

[TÀI LIỆU THAM KHẢO 33](#_Toc200645561)

# DANH MỤC HÌNH ẢNH

[Hình 1.1 Tỉ lệ thư rác của các khu vực……………………….……..……….7](#_heading=h.2jxsxqh)  
[Hình 3.1 Hàm huấn luyện train\_model……………………………..………..21](#_heading=h.2jxsxqh)

[Hình 3.2 Hàm classify\_test …………………………………………………. 24](#_heading=h.z337ya)

[Hình 3.3 Hàm thiết kế giao diện ……………………………………………. 25](#_heading=h.1y810tw)

[Hình 3.4 Hàm run filter …………………………………………………….. 28](#_heading=h.1y810tw)

[Hình 3.5. Giao diện hệ thống phân loại thư rác…………………………..…. 29](#_heading=h.2jxsxqh)

[Hình 3.6 Kết quả của tin nhắn spam ………………….…………………….. 30](#_heading=h.z337ya)

[Hình 3.7 Kết quả của tin nhắn ham ………………….…………………….. 30](#_heading=h.z337ya)

**MỞ ĐẦU**

**1.Lí do chọn đề tài:**

Việc sử dụng Internet ngày càng phổ biến đã mang lại nhiều tiện ích cho người dùng, nhưng đồng thời cũng kéo theo sự gia tăng của các vấn nạn như tin nhắn rác và cuộc gọi rác. Theo thống kê từ Bộ Thông tin và Truyền thông, trong 10 tháng đầu năm 2024, tổng đài phản ánh 156/5656 đã ghi nhận gần 850.000 lượt phản ánh từ người dùng, trong đó có khoảng 185.000 lượt liên quan đến tin nhắn rác, chiếm 22% tổng số phản ánh . Tình trạng này không chỉ gây phiền toái cho người dùng mà còn đặt ra những thách thức lớn về an ninh mạng và quản lý thông tin.

Mới đây, thuật toán máy học (machine learning) đã được áp dụng rộng rãi trong các lĩnh vực khác nhau như xử lý ngôn ngữ tự nhiên, nhận diện hình ảnh, khai thác dữ liệu, phân tích tín hiệu, v.v. Thuật toán máy học là một phương pháp mạnh mẽ trong việc phân loại thư rác vì khả năng tự học từ dữ liệu và cải thiện hiệu suất qua thời gian khi có thêm dữ liệu có sẵn. Những mô hình máy học này có thể phân loại email dựa trên các đặc trưng của email và huấn luyện mô hình trên các tập dữ liệu có sẵn.

Trong số các thuật toán máy học được sử dụng để phân loại thư rác, thuật toán naive bayes (NB) là một trong những thuật toán đơn giản nhất và hiệu quả nhất. Naive Bayes là một thuật toán học máy dựa trên định lý Bayes, sử dụng xác suất có điều kiện để dự đoán khả năng một tin nhắn thuộc vào một lớp cụ thể (ví dụ: spam hoặc ham). Với giả định rằng các đặc trưng (từ ngữ) trong một tin nhắn là độc lập với nhau, thuật toán này cho phép xử lý, phân loại văn bản một cách nhanh chóng.  
 Vì những lý do trên, tôi đã chọn đề tài “Xây dựng hệ thống phân loại thư rác bằng thuật toán Naïve Bayes” để nghiên cứu và thực hiện. Mục tiêu của đề tài là xây dựng một hệ thống phân loại thư rác sử dụng thuật toán NB và đánh giá hiệu quả của hệ thống so với các phương pháp khác. Đề tài này có ý nghĩa thực tiễn, giúp người dùng xử lí dễ dàng các thư rác. Đồng thời, đề tài này cũng có ý nghĩa khoa học, giúp tôi áp dụng kiến thức về máy học vào bài toán thực tế, tìm hiểu về thuật toán NB và vai trò của nó trong phân loại thư rác.

**2. Mục đích nghiên cứu:**

* Nắm vững cơ sở lý thuyết về thuật toán Naive Bayes, bao gồm nguyên lý hoạt động, giả định độc lập và các biến thể phổ biến.
* Phân tích khả năng ứng dụng của Naive Bayes trong lĩnh vực xử lý ngôn ngữ tự nhiên, đặc biệt là trong việc phân loại văn bản.
* Xây dựng một hệ thống có khả năng lọc và phát hiện tin nhắn rác (spam) dựa trên thuật toán Naive Bayes với độ chính xác cao.
* Đánh giá hiệu quả của hệ thống thông qua các chỉ số đo lường như độ chính xác (accuracy), độ nhạy (recall) và độ đặc hiệu (precision).
* So sánh hiệu quả thuật toán Naive Bayes với một số phương pháp phân loại khác (nếu có thời gian mở rộng).

**3. Phương pháp nghiên cứu:**

Phương pháp luận được sử dụng trong đề tài này bao gồm phương pháp nghiên cứu và tổng hợp lý thuyết và phương pháp thực nghiệm. Mục lý thuyết được nghiên cứu và thu thập từ trên mạng,các cuốn sách, bài báo tạp chí chuyên ngành. Sau khi có được nền tảng lý thuyết và mô hình thực nghiệm sẽ tiến hành. Mô hình thực nghiệm được xây dựng, hỗ trợ bởi một số nền tảng lý thuyết. Các kết quả thực nghiệm sẽ được đối chiếu với kết quả lý thuyết để chứng minh sự thành công của thực nghiệm.

**4.Bố cục của bài báo cáo:**

Đề tài thực hiện gồm 3 chương cơ bản như sau:

* **Chương 1**: Tổng quan về kĩ thuật phân loại thư rác
* **Chương 2**: Phân Tích thuật toán Naïve Bayes trong phân loại thư rác
* **Chương 3**:Thiết kế,triển khai hệ thống phân loại thư rác

# CHƯƠNG I. TỔNG QUAN VỀ KĨ THUẬT PHÂN LOẠI THƯ RÁC

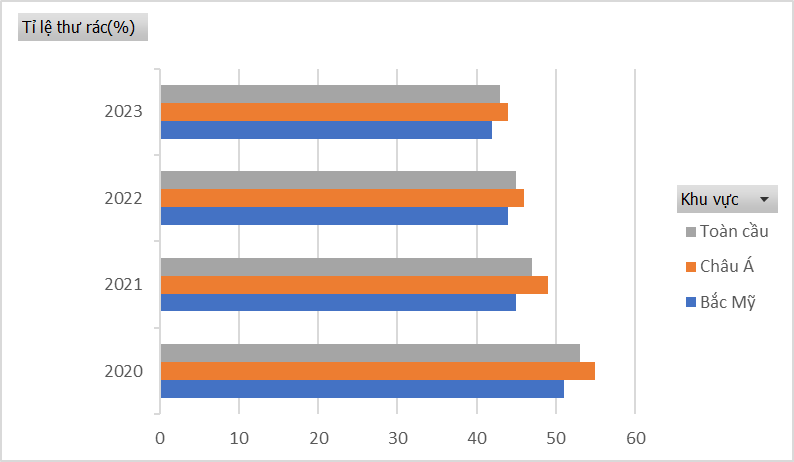
* 1. **. Khái niệm thư rác**

Thư rác (spam mail) là các thông điệp không mong muốn và không được yêu cầu gửi đến người nhận. Thư rác thường chứa các thông tin quảng cáo, thông tin không chính xác, lừa đảo hoặc liên kết độc hại. Nội dung của thư là căn cứ chính quyết định xem bức thư đó có phải là thư rác hay không. Đây cũng là cơ sở cho giải pháp phân loại thư rác bằng nội dung thư.

## 1.2. Tác hại của thư rác

Các tác hại mà thư rác gây ra đối với người dùng:

* Phiền toái và quản lý email:
* Thư rác làm cho hòm thư điện tử trở nên quá tải và khó quản lý.
* Tìm kiếm email quan trọng trở nên khó khăn.
* Nguy cơ an ninh thông tin:
* Thư rác có thể chứa các liên kết độc hại hoặc phần mềm độc hại.
* Virus, spyware, ransomware và các mối đe dọa khác có thể được lây lanqua email thư rác.
* Lừa đảo và đánh cắp thông tin:
* Thư rác thường mang tính lừa đảo, cố gắng lừa người nhận để tiết lộ thông tin cá nhân hoặc tài chính.
* Người gửi thư rác có thể cố gắng lừa người nhận thực hiện các giao dịch gian lận.



Hình 1.1. Tỉ lệ thư rác của các khu vực

## 1.3. Tổng quan về một số kĩ thuật phân loại thư rác:

Phân loại thư rác (spam email classification) là một trong những ứng dụng quan trọng của lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) và trí tuệ nhân tạo (AI). Nhiều phương pháp đã được nghiên cứu và áp dụng nhằm nâng cao hiệu quả phát hiện và ngăn chặn thư rác. Nhìn chung, các phương pháp này có thể được chia thành bốn nhóm chính: **dựa trên luật định (rule-based), dựa trên thống kê (statistical methods), dựa trên học máy (machine learning),** và **dựa trên học sâu (deep learning).**

* **Phương pháp dựa trên luật định (Rule-based Filtering)**: là cách tiếp cận truyền thống, trong đó các quy tắc được định nghĩa thủ công để phát hiện thư rác.
* **Phương pháp dựa trên thống kê** : sử dụng các mô hình xác suất để ước lượng khả năng một email là thư rác dựa trên tần suất xuất hiện của các từ hoặc cụm từ trong nội dung email. Trong đó, thuật toán **Naive Bayes** là một trong những kỹ thuật phổ biến nhất nhờ tính đơn giản và hiệu quả cao trong nhiều tình huống.
* **Phương pháp dựa trên học máy (Machine Learning):** sử dụng các thuật toán học từ dữ liệu đã gán nhãn để xây dựng mô hình phân loại. Các thuật toán thường được sử dụng gồm **Support Vector Machine (SVM), Decision Tree, Random Forest,** và **K-Nearest Neighbors (KNN).** Những thuật toán này có khả năng học các đặc trưng phức tạp từ dữ liệu, cho độ chính xác cao trong việc nhận diện thư rác.
* **Phương pháp dựa trên học sâu (Deep Learning):** là xu hướng hiện đại trong phân loại thư rác, sử dụng các mô hình mạng nơ-ron sâu như **Convolutional Neural Network (CNN), Recurrent Neural Network (RNN),** hoặc **Long Short-Term Memory (LSTM).** Các mô hình này có khả năng tự động trích xuất đặc trưng từ dữ liệu văn bản và học các mẫu ngữ nghĩa phức tạp mà các mô hình truyền thống khó nắm bắt được.

## 1.4. Những thách thức trong việc phân loại thư rác:

Việc phân loại tin nhắn rác, đặc biệt là trong thời đại thông tin số và mạng xã hội phát triển mạnh, đang đối mặt với nhiều thách thức. Các khó khăn này không chỉ đến từ **sự đa dạng nội dung** mà còn từ **sự thay đổi liên tục của kỹ thuật gửi spam.** Dưới đây là những thách thức tiêu biểu:

* Ngôn ngữ tự nhiên và phức tạp: Tin nhắn rác thường được viết bằng **ngôn ngữ không chuẩn,** sử dụng **viết tắt, ký tự đặc biệt, dấu câu thay thế,** gây khó khăn cho hệ thống phân tích văn bản.
* Spam thay đổi liên tục: Người gửi thư rác thường xuyên **thay đổi nội dung, cách diễn đạt để né tránh bộ lọc, điều này khiến các mô hình phân loại phải liên tục cập nhật để tránh bị lỗi thời.** Ví dụ, một bộ lọc hoạt động tốt hôm nay có thể trở nên vô dụng chỉ sau vài tuần nếu nội dung spam đã thay đổi hoàn toàn về cấu trúc và ngữ nghĩa.
* Thiếu dữ liệu huấn luyện chất lượng cao: Phân loại tin nhắn rác thường sử dụng các mô hình học máy hoặc học sâu, đòi hỏi một lượng lớn dữ liệu đầu vào được gán nhãn chính xác. Tuy nhiên, việc thu thập và tạo lập tập dữ liệu đủ lớn, cân bằng giữa tin nhắn rác và tin nhắn hợp lệ, lại là một công việc tốn thời gian và công sức. Ngoài ra, dữ liệu còn phải được cập nhật liên tục để phản ánh đúng thực tế.

**1.5. Tổng kết chương 1**

Trong Chương I, chúng ta đã tìm hiểu tổng quan về khái niệm thư rác, các tác hại mà thư rác gây ra, cũng như các phương pháp kỹ thuật hiện đang được sử dụng để phân loại và xử lý thư rác. Đây là những nền tảng lý thuyết quan trọng, tạo tiền đề cho các chương tiếp theo đi sâu vào kỹ thuật cụ thể, quy trình triển khai mô hình phân loại cũng như đánh giá hiệu quả của hệ thống.

# CHƯƠNG II : PHÂN TÍCH THUẬT TOÁN NAIVE BAYES TRONG PHÂN LOẠI THƯ RÁC

Trong chương này, chúng ta đã tìm hiểu chi tiết về cách áp dụng thuật toán Naive Bayes trong bài toán phân loại thư rác, bao gồm lý thuyết nền tảng, các bước triển khai và những ưu nhược điểm của phương pháp.Trước hết, định lý Bayes được giới thiệu như là cơ sở lý thuyết chính, cung cấp công thức tính xác suất hậu nghiệm dựa trên xác suất tiên nghiệm và xác suất có điều kiện. Naive Bayes dựa trên giả định độc lập giữa các đặc trưng, cho phép đơn giản hoá bài toán tính toán xác suất và đưa ra quyết định phân loại dựa trên công thức tối đa hoá xác suất hậu nghiệm.

## 2.1. Định lí Bayes:

Định lý Bayes là một quy tắc trong xác suất thống kê, cung cấp phương pháp tính toán xác xuất của một sự kiện dựa vào thông tin trước đó:

P(h | D) = P(D | h).P( h ) / P(D) (1)

* P(h): Xác suất trước của giả thiết (phân loại) h • P(D): Xác suất trước của việc quan sát được dữ liệu D
* P(D|h): Xác suất (có điều kiện) của việc quan sát được dữ liệu D, nếu biết giả thiết h là đúng
* P(h|D): Xác suất (có điều kiện) của giả thiết h là đúng, nếu quan sát được dữ liệu D

## 2.2. Mô tả thuật toán Naive Bayes:

* + Naive Bayes là một phương pháp học máy dựa trên định lý Bayes, thường dùng trong bài toán phân loại như xử lý văn bản (phân loại email spam, phân tích cảm xúc).
  + Naive Bayes được gọi là "naive" (ngây thơ) vì nó giả định rằng các đặc trưng (features) của dữ liệu là độc lập và có ảnh hưởng riêng biệt lên kết quả dự đoán. Mặc dù giả định này thường không chính xác trong thực tế, Naive Bayes vẫn được sử dụng phổ biến do tính đơn giản và khả năng xử lý hiệu quả với dữ liệu.
* Thuật toán Naive Bayes tính toán xác suất của các lớp (classes) dựa trên xác suất của các đặc trưng. Đầu tiên, dữ liệu huấn luyện được sử dụng để tính toán các xác suất có điều kiện. Sau đó, khi có một dữ liệu mới, thuật toán sử dụng định lý Bayes để tính toán xác suất xảy ra của các lớp và chọn lớp có xác suất cao nhất là kết quả dự đoán.
* Trong huấn luyện, Naive Bayes tính toán các xác suất P(X|C) ở công thức (1) bằng cách giả định rằng các đặc trưng (features) là độc lập nhau. Điều này có nghĩa là xác suất của một đặc trưng trong dữ liệu được tính riêng biệt cho từng lớp, và xác suất tổng hợp của dữ liệu X được tính bằng cách nhân các xác suất này lại với nhau.
* Naive Bayes có thể được áp dụng cho nhiều loại dữ liệu, bao gồm cả dữ liệu dạng văn bản. Trong trường hợp này, các đặc trưng của văn bản có thể là các từ và xác suất P(X|C) được tính dựa trên tần suất xuất hiện của từng từ trong các văn bản huấn luyện thuộc cùng một lớp.

## 2.3.Các bước triển khai thuật toán Naïve Bayes:

### 2.3.1. Thu thập và tiền xử lý dữ liệu

Bước đầu tiên là thu thập tập dữ liệu văn bản đã được gán nhãn (ví dụ: “spam” hoặc “ham”). Sau đó, dữ liệu được tiền xử lý bằng các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP), bao gồm:

* Chuyển văn bản về chữ thường: Giúp đồng nhất từ ngữ, tránh phân biệt giữa “Buy” và “buy”.
* Loại bỏ dấu câu, ký tự đặc biệt và số không cần thiết: Giữ lại các từ có ý nghĩa trong phân loại.
* Tách từ (Tokenization): Chia văn bản thành các đơn vị từ riêng lẻ.
* Loại bỏ từ dừng (Stop words): Loại bỏ những từ phổ biến như “the”, “is”, “at” vì ít mang ý nghĩa phân biệt.
* Chuẩn hóa từ (Stemming hoặc Lemmatization):
  + Stemming: Cắt bỏ hậu tố để đưa từ về gốc. Ví dụ: "running", "ran" → "run".
  + Lemmatization: Dùng từ điển ngôn ngữ để chuyển từ về dạng chuẩn (ví dụ: “better” → “good”).

### 2.3.2. Xây dựng từ điển và tính xác suất

Trong bước này, mục tiêu là tạo ra tập từ vựng (từ điển) và thống kê tần suất xuất hiện của từng từ trong từng lớp (ví dụ: “spam” và “ham”). Đây là nền tảng để tính toán xác suất có điều kiện trong các bước tiếp theo của thuật toán Naive Bayes.

**2.3.2.1. Xây dựng từ điển (Vocabulary)**

* Từ điển được tạo ra bằng cách trích xuất tất cả các từ (token) xuất hiện trong toàn bộ tập dữ liệu huấn luyện sau khi đã qua tiền xử lý (tokenization, loại bỏ stop words, chuẩn hóa).
* Mỗi từ trong từ điển sẽ trở thành một đặc trưng (feature) trong mô hình phân loại.
* Từ điển được xây dựng thống nhất và dùng chung cho tất cả các lớp để đảm bảo tính toàn vẹn và đồng nhất trong huấn luyện và dự đoán.
* Ví dụ với hai câu: "Buy now and get a discount", "Offer valid only today"
* Sau tiền xử lí, từ điển có thể là: → {buy, now, get, discount, offer, valid, today}

**2.3.2.2. Thống kê tần suất xuất hiện**

Sau khi có từ điển, bước tiếp theo là thống kê số lần xuất hiện của từng từ trong từng lớp dữ liệu để phục vụ cho việc tính xác suất có điều kiện.

Các chỉ số chính cần tính:

* count(w, c): Số lần từ www xuất hiện trong các văn bản thuộc lớp ccc (ví dụ: “spam” hoặc “ham”).
* total\_count(c): Tổng số từ (kể cả trùng lặp) xuất hiện trong tất cả văn bản thuộc lớp ccc.

Việc thống kê này giúp đánh giá mức độ liên quan của từ đối với từng lớp và được dùng trong công thức tính xác suất ở bước sau.

Ví dụ từ “discount” có thể:

* Xuất hiện 10 lần trong thư spam
* Xuất hiện 3 lần trong thư ham

### 2.3.3 Tính xác suất điều kiện

Dựa trên từ điển và xác suất từng từ đã tính được, tiến hành tính xác suất có điều kiện cho từng từ trong một thư thuộc về lớp "spam" hoặc "ham". Cụ thể, với mỗi từ trong tập từ vựng, ta xác định xác suất nó xuất hiện trong văn bản khi biết văn bản thuộc một lớp nhất định. Kỹ thuật làm trơn Laplace thường được sử dụng để tránh trường hợp xác suất bằng 0.

Trong thuật toán Naive Bayes, bước quan trọng là tính xác suất có điều kiện của từng từ (hoặc đặc trưng) trong văn bản khi biết trước nhãn lớp (ví dụ: "spam" hoặc "ham"). Công thức cơ như sau:

Trong đó:

count(w, c): Số lần từ w xuất hiện trong các thư thuộc lớp c.

total\_count(c): Tổng số từ (bao gồm cả lặp lại) trong toàn bộ thư thuộc lớp c

|V|: Số lượng từ duy nhất trong toàn bộ tập dữ liệu huấn luyện (kích thước từ điển)

## 2.3.4 Áp dụng định lý Bayes

Cơ sở lý thuyết xác suất

Gọi A, B là hai biến cố

Với P(B)>0:

P=

Suy ra:

P(AB) = PP(B) = P(B|A)P(A)

Công thức Bayes:

P(B|A) = = =

Công thức Bayes tổng quát:

Với P(A) và{B1, B2,…,Bn} là một hệ đầy đủ các biến cố:

Tổng sác suất của hệ bằng 1:

Từng đôi một xung khắc:

P

Khi đó ta có:

P =

=  
  Trong đó ta gọi A là một chứng cứ (evidence) (trong bài toán phân lớp A sẽ là một phần tử dữ liệu), B là một giả thiết nào để cho A thuộc về một lớp C nào đó. Trong bài toán phân lớp chúng ta muốn xác định giá trị P(B/A) là xác suất để giả thiết B là đúng với chứng cứ A thuộc vào lớp C với điều kiện ra đã biết các thông tin mô tả A. P(B|A) là một xác suất hậu nghiệm (posterior probability hay posteriori probability) của B với điều kiện A.

**Áp dụng vào hệ thống phân loại thư rác**

Trong mô hình Naive Bayes, ta muốn tính xác suất để một thư điện tử với nội dung cụ thể thuộc về một lớp nhất định (ví dụ: **thư rác** hoặc **thư hợp lệ**). Theo định lý Bayes, ta có:

Trong đó:

Ck: là một lớp (ví dụ: spam)

w1,w2,…wn​: là các từ trong thư

P(Ck): là xác suất tiên nghiệm của lớp Ck

P(w1,...,wn∣Ck):là xác suất các từ xuất hiện đồng thời khi biết thư thuộc lớp Ck

## 2.3.5 Quyết định phân loại

Sau khi tính toán được xác suất hậu nghiệm P(C∣w1,w2,...,wn) cho từng lớp Ck​ (ví dụ: “spam” hoặc “ham”) dựa trên danh sách các từ w1,w2,...,wn ​ xuất hiện trong thư điện tử, mô hình Naive Bayes sẽ thực hiện bước quyết định phân loại.

Cụ thể, hệ thống sẽ so sánh xác suất giữa các lớp và lựa chọn lớp có giá trị xác suất hậu nghiệm lớn nhất. Việc lựa chọn này đảm bảo rằng thư sẽ được gán vào lớp có khả năng xảy ra cao nhất theo mô hình thống kê đã học từ dữ liệu huấn luyện.

Công thức quyết định phân loại như sau:

Ĉ = arg max₍Cₖ₎ [ log P(Cₖ) + ∑₍i=1->n₎ log P(wᵢ | Cₖ) ]

Trong đó:

Ĉ: lớp dự đoán cuối cùng cho thư điện tử cần phân loại.

P(Ck): xác suất tiên nghiệm của lớp Ck

P(wi∣Ck): xác suất có điều kiện của từ wi trong lớp Ck

## 2.3.6. Đánh giá và tinh chỉnh mô hình

Sau khi huấn luyện mô hình Naïve Bayes, bước đánh giá và tinh chỉnh là cần thiết để đảm bảo hệ thống phân loại thư rác hoạt động chính xác và ổn định trên dữ liệu thực tế.

**2.3.6.1. Đánh giá hiệu suất mô hình**

Mô hình thường được đánh giá trên tập kiểm tra độc lập, sử dụng các chỉ số phổ biến trong học máy:

* Accuracy (Độ chính xác): Tỷ lệ dự đoán đúng trên tổng số mẫu.
* Precision (Độ chính xác theo lớp): Tỷ lệ thư được phân loại là “spam” thực sự là spam.
* Recall (Độ nhạy): Tỷ lệ thư “spam” được nhận diện đúng trên tổng số thư spam thực tế.
* F1-score: Trung bình điều hòa giữa Precision và Recall, phản ánh sự cân bằng giữa độ chính xác và độ bao phủ.
* Confusion Matrix (Ma trận nhầm lẫn): Biểu diễn trực quan số lượng mẫu được phân loại đúng và sai theo từng lớp.

Ngoài ra, kỹ thuật đánh giá chéo (k-fold cross-validation) cũng thường được sử dụng để kiểm tra độ ổn định và khả năng khái quát của mô hình, tránh hiện tượng quá khớp (overfitting) vào một tập dữ liệu cụ thể.

2.3.6.2. Tinh chỉnh mô hình

Để nâng cao hiệu quả mô hình, có thể áp dụng một số chiến lược tinh chỉnh sau:

* Cải thiện bước tiền xử lý: Loại bỏ từ dừng, chuẩn hóa văn bản, áp dụng stemming hoặc lemmatization giúp giảm nhiễu và giữ lại thông tin quan trọng.
* Chọn đặc trưng nâng cao: Thay vì chỉ sử dụng tần suất từ đơn giản, có thể áp dụng TF-IDF để tăng trọng số cho các từ mang tính phân biệt cao.
* Tối ưu kỹ thuật làm trơn: Có thể điều chỉnh hệ số Laplace hoặc áp dụng các phương pháp khác như m-estimate để cải thiện xử lý với các từ hiếm.
* Cân bằng dữ liệu: Nếu dữ liệu bị mất cân bằng (ví dụ: thư "ham" nhiều hơn thư "spam"), có thể sử dụng các kỹ thuật như oversampling (tăng mẫu lớp ít) hoặc undersampling (giảm mẫu lớp nhiều) để mô hình học được đầy đủ hơn và tránh thiên lệch.

Việc đánh giá và tinh chỉnh nên được thực hiện liên tục trong quá trình phát triển để đảm bảo mô hình đạt được hiệu suất tối ưu khi triển khai thực tế.

## 2.4. Phân loại

Thuật toán **Naive Bayes** là một họ các thuật toán phân loại dựa trên định lý Bayes với giả định mạnh mẽ (naive) rằng các đặc trưng (features) là độc lập với nhau. Đây là thuật toán phổ biến trong học máy (machine learning), đặc biệt trong các bài toán phân loại văn bản (như spam email, phân tích cảm xúc, v.v.).

Dưới đây là **phân loại các thuật toán Naive Bayes** chính:

* Gaussian Naive Bayes:

**Công thức:**

Với mỗi đặc trưng , mô hình giả định:

P(| ) = exp (- )

**Ứng dụng:** Phân loại dữ liệu thực như chiều cao, cân nặng, điểm số.

* Multinomial Naive Bayes:

**Mô tả:** Dùng cho dữ liệu rời rạc (discrete), đặc biệt trong phân loại văn bản (sử dụng bag-of-words hoặc TF-IDF).

Công thức:

P(| ) =

Với là số lần từ xuất hiện trong lớp , là hệ số làm trơn Laplace)

**Ứng dụng:** Phân loại email spam, cảm xúc review, chủ đề bài báo,...

* Bernoulli Naive Bayes:

**Mô tả:** Dùng cho dữ liệu nhị phân (0 hoặc 1). Mỗi đặc trưng chỉ biểu thị có hay không (vd: từ "good" có trong văn bản hay không).

**Ứng dụng:** Phân loại tài liệu ngắn, kiểm tra hiện diện từ, hệ thống lọc spam đơn giản.

* **Complement Naive Bayes:**

**Mô tả:** Là biến thể của MultinomialNB, phù hợp hơn với các tập dữ liệu mất cân bằng (class imbalance).

**Ứng dụng:** Phân loại văn bản với dữ liệu mất cân bằng.

* **Categorical Naive Bayes (Python 3.8+):**

**Mô tả:** Dùng cho đặc trưng phân loại (categorical, non-numeric). Thường áp dụng với dữ liệu có dạng bảng (tabular).

**Ứng dụng:** Phân tích khảo sát, dữ liệu khách hàng dạng phân loại.

## 2.5. Ưu điểm và hạn chế của thuật toán Naive Bayes

### 2.5.1. Ưu điểm

* **Đơn giản và dễ triển khai:** So với nhiều thuật toán phân loại khác, Naive Bayes được xem là đơn giản hơn do quá trình ước lượng tham số dễ dàng.
* **Tốc độ xử lý nhanh, dễ mở rộng:** So với hồi quy logistic, Naive Bayes có tốc độ xử lý nhanh, hoạt động hiệu quả khi giả định độc lập có điều kiện (conditional independence) được đảm bảo. Ngoài ra, nó yêu cầu ít dung lượng lưu trữ.
* **Xử lý tốt dữ liệu nhiều chiều:** Trong các bài toán như phân loại tài liệu, dữ liệu thường có số chiều rất lớn – điều mà nhiều thuật toán khác gặp khó khăn, nhưng Naive Bayes vẫn xử lý tốt.

### 2.5.2. Hạn chế

**- Vấn đề tần suất bằng 0 (Zero Frequency):** Xảy ra khi một giá trị hạng mục không tồn tại trong tập huấn luyện.

- **Giả định cốt lõi không thực tế:** Giả định rằng các đặc trưng độc lập với nhau có thể không đúng trong nhiều trường hợp thực tế, dẫn đến kết quả phân loại sai.

# 

# CHƯƠNG 3: THIẾT KẾ , TRIỂN KHAI HỆ THỐNG PHÂN LOẠI THƯ RÁC

## 3.1. Ngôn ngữ và công cụ triển khai

Để xây dựng mô hình phân loại thư rác bằng Naive Bayes và tích hợp giao diện người dùng, đề tài sử dụng **ngôn ngữ lập trình Python** kết hợp với **môi trường phát triển Visual Studio Code (VS Code).**

### 3.1.1. Python

Python là một ngôn ngữ lập trình bậc cao, thông dịch, có cú pháp rõ ràng và dễ đọc. Đây là ngôn ngữ rất phổ biến trong lĩnh vực khoa học dữ liệu, trí tuệ nhân tạo và học máy nhờ thư viện phong phú, dễ triển khai các thuật toán học máy như Naive Bayes.

Trong đề tài này, Python được sử dụng để:

Xử lý dữ liệu đầu vào (tiền xử lý ngôn ngữ tự nhiên)

Xây dựng và huấn luyện mô hình Naive Bayes

Tính xác suất và phân loại thư mới

Tạo giao diện ứng dụng với thư viện Tkinter

### 3.1.2. Môi trường Visual Studio Code (VS Code)

VS Code là một trình soạn thảo mã nguồn nhẹ, mạnh mẽ và có thể mở rộng nhờ vào hàng ngàn tiện ích mở rộng. Đây là công cụ phù hợp để phát triển ứng dụng Python nhờ các tính năng:

Tô màu cú pháp, hỗ trợ linting và auto-completion

Tích hợp terminal và quản lý môi trường ảo

Debug trực tiếp và theo dõi biến, breakpoint

Dễ dàng tổ chức và quản lý các file .py và dữ liệu

### 3.1.3.Các thư viện Python sử dụng chính:

pandas: đọc dữ liệu và thao tác với bảng dữ liệu

re: xử lý biểu thức chính quy để làm sạch văn bản

tkinter: tạo giao diện người dùng cơ bản

os, sys: hỗ trợ đường dẫn và đóng gói ứng dụng nếu cần

## 3.2. Chuẩn bị dữ liệu huấn luyện

Mô tả định dạng file Thuracspam (gồm 2 cột Label (spam/ham), SMS (nội dung))

Link tập tài liệu: <https://drive.google.com/file/d/1ASEMp4TO1z1FRCuKzlu4WLx_D1OI6Icw/view?usp=drive_link>

## 3.3. Xây dựng mô hình Naive Bayes và triển khai chương trình (modules.py)

### 3.3.1. Hàm huấn luyện train\_model

Hàm train\_model() thực hiện toàn bộ quá trình huấn luyện mô hình Naive Bayes từ dữ liệu văn bản. Mục tiêu là tính được các xác suất cần thiết để sử dụng cho việc phân loại tin nhắn sau này.

A screen shot of a computer program

AI-generated content may be incorrect.  
 Hình 3.1 Hàm huấn luyện train\_model

Dưới đây là giải thích chi tiết từng bước:  
**Bước 1: Đọc dữ liệu từ file vào DataFrame để chuẩn bị xử lý.**

**Bước 2: Làm sạch và chuẩn hóa dữ liệu**

huanluyen['SMS'] = huanluyen['SMS'].str.replace(r'\W', ' ', regex=True).str.lower().str.split()Loại bỏ ký tự không phải chữ (\W)

* Chuyển toàn bộ văn bản sang chữ thường
* Tách từng dòng thành danh sách các từ (split())

**Bước 3:** Tạo tập từ vựng (Vocabulary)

vocab = list(set(word for sms in huanluyen['SMS'] for word in sms))

Duyệt qua toàn bộ tin nhắn, gom tất cả các từ lại

Sử dụng set() để loại bỏ trùng lặp → ta được danh sách từ duy nhất (vocab)

Bước 4: Đếm số lần xuất hiện của mỗi từ

    number\_appear = {word: [0] \* len(huanluyen) for word in vocab}

    for idx, sms in enumerate(huanluyen['SMS']):

        for word in sms:

            number\_appear[word][idx] += 1

Tạo một dict lưu số lần xuất hiện của từng từ trong mỗi dòng tin nhắn

Ví dụ: "free" xuất hiện 2 lần trong dòng thứ 3 → number\_appear['free'][2] = 2

Bước 5: Kết hợp từ đếm với tập dữ liệu ban đầu

    word\_counts = pd.DataFrame(number\_appear)

    train\_extend = pd.concat([huanluyen, word\_counts], axis=1)  
Gộp cột đếm từ vào dataframe huấn luyện

Mỗi dòng giờ có thêm cột với số lần từng từ xuất hiện

**Bước 6:** Tính xác suất tiên nghiệm (prior probabilities)

    spam\_messages = train\_extend[train\_extend['Label'] == 'spam']

    ham\_messages = train\_extend[train\_extend['Label'] == 'ham']

    p\_spam = len(spam\_messages) / len(train\_extend)

    p\_ham = len(ham\_messages) / len(train\_extend) **P(spam)** = tỉ lệ thư rác trong tập huấn luyện

**P(ham)** = tỉ lệ thư thường

**Ví dụ kết quả thực tế (giả sử):**

Số lượng thư spam: 93, thư ham: 347 (trong 440 mẫu huấn luyện)  
→ P(spam) = 93/440 ≈ 0.211, P(ham) = 0.789

**Bước 7:** Tính xác suất có điều kiện với Laplace smoothing

    n\_spam = spam\_messages['SMS'].apply(len).sum()

    n\_ham = ham\_messages['SMS'].apply(len).sum()

    n\_vocab = len(vocab)

    parameters\_spam = {word: (spam\_messages[word].sum() + alpha) / (n\_spam + alpha \* n\_vocab) for word in vocab}

    parameters\_ham = {word: (ham\_messages[word].sum() + alpha) / (n\_ham + alpha \* n\_vocab) for word in vocab}

**n\_spam**: tổng số từ trong tất cả thư spam

**n\_ham**: tổng số từ trong tất cả thư ham

**n\_vocab**: số lượng từ trong vocab (ví dụ: 840 từ)

Dùng công thức Laplace smoothing:

P(

**Ví dụ kết quả (một số từ điển hình):**

| Từ | P(word|spam) | P(word|ham) |  
|------------|--------------|-------------|  
| free | 0.0012 | 0.0001 |  
| win | 0.0009 | 0.00005 |  
| ok | 0.0002 | 0.0007 |  
| love | 0.0003 | 0.0010 |

Như vậy, mô hình đã học rằng từ như free, win thường xuất hiện trong spam, còn từ như ok, love phổ biến trong ham.

**3.3.2. Hàm phân loại 1 tin nhắn classify\_test**

A computer screen with text

AI-generated content may be incorrect.Hàm classify\_test() là thành phần quan trọng để áp dụng mô hình Naive Bayes đã huấn luyện nhằm phân loại một tin nhắn văn bản đầu vào thành **spam** hoặc **ham**.

**Hình 3.2: Hàm classify\_test**

**Bước 1: Tiền xử lý văn bản đầu vào**:

Dùng re.sub() để loại bỏ ký tự đặc biệt, chỉ giữ lại chữ và số.

Đưa về chữ thường (lower()), sau đó tách từng từ (split()).

**Bước 2**: **Khởi tạo xác suất tiên nghiệm**:

Gán p\_spam\_given = P(spam) và p\_ham\_given = P(ham) – đây là xác suất tiên nghiệm từ quá trình huấn luyện.

**Bước 3: Tính xác suất hậu nghiệm**:

Duyệt qua từng từ trong tin nhắn.

Nếu từ có tồn tại trong tập từ vựng (đã tính xác suất có điều kiện):

Nhân lần lượt với P(word|spam) và P(word|ham) để cập nhật p\_spam\_given và p\_ham\_given.

**Bước 4: So sánh và phân loại**:

Nếu p\_spam\_given > p\_ham\_given → Dự đoán là **spam**

Ngược lại nếu p\_ham\_given > p\_spam\_given → Dự đoán là **ham**

Trường hợp xác suất bằng nhau → Không chắc chắn → Trả về 'needs human classification'

## 3.4. Xây dựng giao diện người dùng (giaodien.py)

### 3.4.1. Mô tả giao diện người dùng

Giao diện bao gồm các thành phần chính:

* **Ô nhập tin nhắn**: là một ScrolledText để người dùng nhập nội dung tin nhắn cần kiểm tra.
* **Nút "Lọc thư"**: khi nhấn vào, hệ thống sẽ gửi tin nhắn đến hàm phân loại và hiển thị kết quả.
* **Kết quả phân loại**:
* Nếu là **spam**, nội dung được hiển thị trong vùng "Spam".
* Nếu là **ham**, nội dung hiển thị trong vùng "Ham".
* Nếu không xác định được, hệ thống sẽ thông báo cần phân loại thủ công.

### A screen shot of a computer code AI-generated content may be incorrect.3.4.2. Thiết kế giao diện

Hình 3.3: Hàm thiết kế giao diện

- Cấu hình cửa sổ

        self.master.title("Phân loại thư rác")

        self.master.geometry("1050x900")

        self.master.resizable(False, False)  
Đặt tiêu đề cửa sổ.

* Thiết lập kích thước cố định: 1050x900 pixel.
* resizable(False, False): không cho người dùng thay đổi kích thước.

- Nhãn và ô nhập nội dung thư:

        tk.Label(master, text="Nhập nội dung thư:", font=("Arial", 12)).pack(anchor='w', padx=10, pady=(10, 0))

        self.email\_input = scrolledtext.ScrolledText(master, height=20, font=("Arial", 11))

        self.email\_input.pack(fill="x", padx=10, pady=5)

* Nhãn "Nhập nội dung thư:".
* Vùng nhập nội dung thư muốn phân loại.

- Nút bấm "Lọc thư":

        tk.Button(

            master,

            text="Lọc thư",

            font=("Arial", 12, "bold"),

            bg="#4CAF50",

            fg="white",

            height=2,

            width=15,

            command=self.run\_filter

        ).pack(pady=15)

* Nút có label là "Lọc thư", màu xanh, kích thước lớn.
* Khi nhấn sẽ gọi hàm run\_filter.

- Vùng hiển thị kết quả "Spam":

        tk.Label(master, text="Spam:", font=("Arial", 11)).pack(anchor='w', padx=10)

        self.spam\_words\_output = scrolledtext.ScrolledText(master, height=10, font=("Arial", 11), state='normal')

        self.spam\_words\_output.pack(fill="x", padx=10, pady=5)

* Nhãn "Spam".
* Vùng hiển thị kết quả nếu tin nhắn bị phân loại là spam

- Vùng hiển thị kết quả "Ham":

        tk.Label(master, text="Ham:", font=("Arial", 11)).pack(anchor='w', padx=10)

        self.result\_output = scrolledtext.ScrolledText(master, height=10, font=("Arial", 11), state='normal')

        self.result\_output.pack(fill="x", padx=10, pady=(5, 10))

* Vùng hiển thị kết quả nếu tin nhắn là ham (bình thường).

**3.4.3. Kết nối giữa giao diện và modules**

Giao diện sử dụng hàm classify\_test() từ module test.py đã huấn luyện trước đó (một phần của backend). Kết nối được thực hiện qua dòng:

from modules import classify\_test

Sau đó, hàm run\_filter() trong lớp SpamFilterGUI sẽ gọi classify\_test() để thực hiện phân loại nội dung mà người dùng nhập.

A computer screen shot of text

AI-generated content may be incorrect.

Hình 3.4 : Hàm run\_filter

**Phương thức** run\_filter**: Xử lý khi nhấn nút**

**Bước 1:** Lấy nội dung người dùng nhập

message = self.email\_input.get("1.0", tk.END).strip()

Lấy toàn bộ nội dung từ ScrolledText, từ dòng 1, cột 0 đến cuối (tk.END).

strip() loại bỏ khoảng trắng dư thừa.

**Bước 2:** Kiểm tra nội dung có rỗng không

        if not message:

            messagebox.showwarning("Cảnh báo", "Vui lòng nhập nội dung thư!")

            return

Nếu không nhập nội dung, hiển thị hộp thoại cảnh báo.

**Bước 3:** Gọi hàm phân loại

        result = classify\_test(message)

Hàm classify\_test xử lý nội dung, trả về "spam", "ham" hoặc "unknown".

Bước 4: Cập nhật giao diện theo kết quả

        self.spam\_words\_output.config(state='normal')

        self.result\_output.config(state='normal')

        self.spam\_words\_output.delete("1.0", tk.END)

        self.result\_output.delete("1.0", tk.END)

Mở khóa vùng hiển thị để xóa nội dung cũ.

        if result == "spam":

            self.spam\_words\_output.insert(tk.END, message)

        elif result == "ham":

            self.result\_output.insert(tk.END, message)

        else:

            messagebox.showinfo("Thông báo", "Không thể phân loại. Cần phân loại thủ công.")

Tùy theo kết quả, chèn tin nhắn vào vùng Spam hoặc Ham.

Nếu không phân loại được, hiển thị thông báo.

      self.spam\_words\_output.config(state='disabled')

        self.result\_output.config(state='disabled')

Đóng lại các vùng hiển thị để tránh người dùng chỉnh sửa.

**3.5. Kết quả thực nghiệm**

A screenshot of a computer

AI-generated content may be incorrect.Sau khi xây dựng thành công mô hình Naive Bayes và giao diện người dùng bằng Tkinter, chương trình đã được chạy thử nghiệm với một số tin nhắn mẫu để đánh giá khả năng phân loại thư rác (**spam**) và thư bình thường (**ham**).

Hình 3.5: Giao diện hệ thống phân loại thư rác

Tin nhắn mẫu dùng để kiểm thử:

**Spam**: Congratulations! You've won a $1000 Walmart gift card. Click here to claim.

**Ham**: Chào bạn, đây là giao diện trong báo cáo lọc thư rác của nhóm 11.

A screenshot of a computer

AI-generated content may be incorrect.  
 Hình 3.6: Kết quả của tin nhắn spam

A screenshot of a computer

AI-generated content may be incorrect.  
 Hình 3.7 : Kết quả của tin nhắn ham

## 3.6 Kết luận chương 3

Trong chương 3, hệ thống phân loại thư rác sử dụng thuật toán Naive Bayes đã được thiết kế và triển khai thành công. Ngôn ngữ lập trình Python kết hợp với môi trường Visual Studio Code đã hỗ trợ thuận lợi cho quá trình phát triển. Dữ liệu huấn luyện được xử lý tiền xử lý kỹ lưỡng, xây dựng mô hình thống kê xác suất với kỹ thuật làm mượt Laplace, đảm bảo mô hình có thể xử lý các từ chưa xuất hiện trong tập huấn luyện.

Song song với việc xây dựng mô hình, giao diện người dùng đơn giản nhưng hiệu quả đã được phát triển bằng thư viện Tkinter, giúp người dùng có thể nhập nội dung tin nhắn và nhận được kết quả phân loại ngay lập tức. Việc kết nối giữa giao diện và mô hình phía backend cũng được thực hiện thông qua cơ chế module hóa, đảm bảo tính dễ bảo trì và mở rộng cho hệ thống.

Kết quả thử nghiệm ban đầu trên một số tin nhắn mẫu cho thấy hệ thống có khả năng nhận diện chính xác thư rác và thư thông thường. Mặc dù vẫn còn một số trường hợp cần phân loại thủ công khi mô hình không đủ thông tin, nhưng nhìn chung hệ thống đã hoàn thành mục tiêu đặt ra ban đầu và sẵn sàng để tiếp tục nghiên cứu, cải thiện hiệu năng cũng như mở rộng tập dữ liệu huấn luyện trong các chương tiếp theo.

**TỔNG KẾT**

* + 1. **Kết quả đạt được:**

Thông qua quá trình học tập, nghiên cứu tài liệu, thực hành và thảo luận nhóm, chúng em đã từng bước tiếp cận, hiểu rõ và áp dụng hiệu quả kiến thức về thuật toán Naive Bayes vào việc xây dựng hệ thống phân loại thư rác. Đây là kết quả của sự nỗ lực nghiêm túc và phối hợp chặt chẽ giữa các thành viên trong nhóm dưới sự hướng dẫn tận tình của giảng viên.

Ngoài việc nắm vững kiến thức phân loại văn bản và thuật toán Naive Bayes, chúng em còn phát triển được nhiều kỹ năng mềm quan trọng như: kỹ năng tìm kiếm và chọn lọc thông tin từ nguồn tài liệu đáng tin cậy; kỹ năng tư duy logic trong việc phân tích và giải quyết vấn đề; kỹ năng làm việc nhóm hiệu quả; kỹ năng lập trình, triển khai mô hình học máy; cũng như khả năng trình bày và truyền đạt ý tưởng một cách rõ ràng, mạch lạc.

* + 1. **Hạn chế**

Mặc dù đã cố gắng đầu tư thời gian và công sức cho đề tài, nhưng do giới hạn về kinh nghiệm và kiến thức thực tiễn, mô hình vẫn còn đơn giản, chưa xử lý được các dạng thư rác phức tạp hay tinh vi. Ngoài ra, khả năng đánh giá phân loại còn hạn chế. Chúng em mong nhận được sự góp ý của thầy để hoàn thiện đề tài tốt hơn trong tương lai.

* + 1. **Đóng góp thực tiễn**

Hệ thống phân loại thư rác bằng giải thuật Naive Bayes khi được áp dụng vào thực tế sẽ góp phần hỗ trợ người dùng lọc email hiệu quả, hạn chế tình trạng nhận thư rác gây phiền toái hoặc có nguy cơ bảo mật. Việc tự động hóa quá trình kiểm duyệt thư đến giúp tiết kiệm thời gian, tăng hiệu suất làm việc, đồng thời giảm rủi ro từ các email chứa nội dung độc hại.

**TÀI LIỆU THAM KHẢO**

[1] : Nguyen Thi Hop , Thuật toán phân lớp Naïve Bayes , 2019 [truy cập vào ngày 27/5/2025] từ   
<https://viblo.asia/p/thuat-toan-phan-lop-naive-bayes-924lJWPm5PM>

[2] : Cao Hoàng Trụ, Trí tuệ Nhân tạo = Thông minh + Giải thuật. Nhà Xuất bản Đại học Quốc gia TP.HCM,2018 [truy cập vào ngày 2/6/2025]

[3] : Bùi Ngọc Lan , nghiên cứu mạng thư điện tử và ứng dụng trong lọc thư rác,2008, [truy cập ngày 4/6/2025] từ

<https://doan.edu.vn/do-an/luan-van-nghien-cuu-mang-thu-dien-tu-va-ung-dung-trong-loc-thu-rac-bui-ngoc-lan-22144/>

[4] : Nguyễn Văn Huy , Thuật toán bayes và ứng dụng ,2009 [ Truy cập vào ngày 7/6/2025] từ

https://www.thuvientailieu.vn/tai-lieu/khoa-luan-thuat-toan-bayes-va-ung-dung-4426/

[5] : ChatGPT : <https://chatgpt.com/>

# 