TRƯỜNG ĐẠI HỌC CÔNG NGHỆ GIAO THÔNG VẬN TẢI
KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN
HỆ TRỢ GIÚP QUYẾT ĐỊNH

ĐỀ TÀI:
THUẬT TOÁN NAIVES BAYES
VÀ ỨNG DỤNG TRONG LỌC EMAIL RÁC

GIẢNG VIÊN HƯỚNG DẪN: ĐOÀN THỊ THANH HẰNG
SINH VIÊN THỰC HIỆN: PHẠM ANH ĐỨC
PHÙNG MINH TRƯỜNG
ĐỖ THÁI HẢI ANH
ĐINH GIA BẢO
NGUYỄN VĂN ANH
LỚP: 74DCHT21

HÀ NỘI, 9/2025

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ GIAO THÔNG VẬN TẢI
KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN
HỆ TRỢ GIÚP QUYẾT ĐỊNH

ĐỀ TÀI:
THUẬT TOÁN NAIVES BAYES
VÀ ỨNG DỤNG TRONG LỌC EMAIL RÁC

GIẢNG VIÊN HƯỚNG DẪN: ĐOÀN THỊ THANH HẰNG
SINH VIÊN THỰC HIỆN: PHẠM ANH ĐỨC
PHÙNG MINH TRƯỜNG
ĐỖ THÁI HẢI ANH
ĐINH GIA BẢO
NGUYỄN VĂN ANH
LỚP: 74DCHT21

HÀ NỘI, 9/2025

NHẬN XÉT – ĐÁNH GIÁ CỦA GIÁO VIÊN HƯỚNG DẪN

ĐÁNH GIÁ HOẠT ĐỘNG THÀNH VIÊN

STT	MÃ SV	Thành viên	Công việc	Đánh giá	Ghi chú
1	74DCHT22041	Phạm Anh Đức	Chương 3: Thực nghiệm và ứng dụng Thu thập dataset Triển khai thực nghiệm và cài đặt ứng dụng	25%	TN
2	74DCHT22234	Phùng Minh Trường	Chương 2: Cơ sở lý thuyết	20%	
3	74DCHT22207	Đinh Gia Bảo	Chương 3: Thực nghiệm và ứng dụng Thu thập dataset	22%	
4	74DCHT21183	Nguyễn Văn Anh	Chương 1: Tổng quan về Hệ trợ giúp quyết định	16.5%	
5	74DCHT22024	Đỗ Thái Hải Anh	Chương 2: Cơ sở lý thuyết	16.5%	

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC CÁC TỪ VIẾT TẮT

Viết tắt	Thuật ngữ tiếng Anh	Thuật ngữ tiếng Việt
DSS	Decision Support System	Hệ trợ giúp quyết định
AI	Artificial Intelligence	Trí tuệ nhân tạo
ML	Machine Learning	Học máy
NB	Naive Bayes	Thuật toán Naive Bayes
MNB	Multinomial Naive Bayes	Naive Bayes đa thức
BNB	Bernoulli Naive Bayes	Naive Bayes Bernoulli
GNB	Gaussian Naive Bayes	Naive Bayes Gaussian
CNB	Complement Naive Bayes	Naive Bayes bổ sung
TF-IDF	Term Frequency – Inverse Document Frequency	Trọng số tần suất – nghịch đảo tần suất văn bản
CSV	Comma-Separated Values	Tập tin dữ liệu dạng bảng phân cách bằng dấu phẩy

LỜI NÓI ĐẦU

Trong bối cảnh khối lượng thông tin ngày càng gia tăng, việc xây dựng các hệ thống hỗ trợ ra quyết định trở thành nhu cầu cấp thiết, đặc biệt trong lĩnh vực xử lý và phân loại dữ liệu văn bản. Với mục tiêu học tập và vận dụng kiến thức đã được trang bị trong môn Hệ trợ giúp quyết định, nhóm đã thực hiện đề tài "Ứng dụng thuật toán Naive Bayes trong phân loại thư điện tử".

Bài báo cáo không chỉ tập trung vào việc tìm hiểu cơ sở lý thuyết của thuật toán Naive Bayes mà còn triển khai thực nghiệm trên dữ liệu thực tế để kiểm chứng hiệu quả. Thông qua quá trình thực hiện, nhóm đã củng cố kiến thức về mô hình xác suất, xử lý ngôn ngữ tự nhiên và quy trình xây dựng một hệ thống hỗ trợ quyết định.

Nhóm xin chân thành cảm ơn cô ***Đoàn Thị Thanh Hằng*** đã tận tình hướng dẫn trong suốt quá trình học tập, giúp nhóm có cơ hội tiếp cận và ứng dụng các kiến thức lý thuyết vào bài toán thực tiễn. Mặc dù đã có nhiều cố gắng, báo cáo khó tránh khỏi những thiếu sót, nhóm rất mong nhận được ý kiến đóng góp của thầy/cô để hoàn thiện hơn.

CHƯƠNG 1. TỔNG QUAN VỀ HỆ TRỢ GIÚP QUYẾT ĐỊNH

1.1 Khái niệm và đặc điểm

a) Khái niệm

Hệ thống hỗ trợ quyết định (Decision Support System - DSS) là một hệ thống phần mềm máy tính được thiết kế đặc biệt nhằm hỗ trợ quá trình ra quyết định trong các tổ chức. Nó cung cấp một nền tảng tích hợp để thu thập, xử lý và phân tích dữ liệu, từ đó đưa ra những gợi ý, mô hình hóa và đánh giá các phương án lựa chọn.

Điểm đặc biệt của DSS là khả năng xử lý thông tin trong những tình huống phức tạp, khi mà thông tin có sẵn không đầy đủ hoặc khi có quá nhiều lựa chọn cần cân nhắc. Hệ thống này không chỉ đơn thuần cung cấp thông tin, mà còn hỗ trợ người dùng trong việc đánh giá và so sánh các phương án, giúp họ đưa ra quyết định sáng suốt hơn.

Ngoài ra, DSS thường có tính tương tác cao, cho phép người dùng dễ dàng thử nghiệm các kịch bản "nếu – thì" (what-if analysis). DSS cũng khác với các hệ thống thông tin quản lý (MIS) ở chỗ nó không chỉ dừng lại ở việc báo cáo dữ liệu, mà còn khai thác các mô hình phân tích, thống kê, hoặc trí tuệ nhân tạo để tạo ra giải pháp linh hoạt và thích ứng với từng hoàn cảnh ra quyết định.

b) Đặc điểm

Hệ thống hỗ trợ ra quyết định (DSS) mang trong mình nhiều đặc điểm nổi bật, giúp nó trở thành công cụ hữu ích trong quá trình phân tích và lựa chọn phương án cho các nhà quản lý.

- ***Tính tương tác cao:***

DSS cho phép người dùng tương tác trực tiếp với hệ thống một cách linh hoạt và thuận tiện. Người dùng có thể dễ dàng điều chỉnh các tham số đầu vào, thay đổi các giả định trong mô hình, và ngay lập tức xem được kết quả của những thay đổi này. Tính tương tác này giúp người ra quyết định có thể khám phá nhiều kịch bản khác nhau, từ đó đưa ra quyết định phù hợp nhất.

- ***Tính linh hoạt và khả năng thích ứng:***

DSS có khả năng thích ứng cao với các vấn đề kinh doanh cụ thể và đa dạng. Hệ thống có thể được tùy chỉnh để phù hợp với nhu cầu đặc thù của từng doanh nghiệp, từng ngành nghề. Điều này cho phép DSS có thể áp dụng trong nhiều lĩnh vực khác nhau, từ tài chính, marketing cho đến quản lý chuỗi cung ứng.

- ***Hỗ trợ ra quyết định:***

Một đặc điểm quan trọng của DSS là nó được thiết kế để hỗ trợ, chứ không phải thay thế hoàn toàn vai trò của con người trong quá trình ra quyết định. DSS cung cấp thông tin, phân tích và gợi ý, nhưng quyết định cuối cùng vẫn thuộc về người sử dụng.

Tóm lại, nhờ những đặc điểm trên, DSS vừa đảm bảo tính khoa học trong phân tích, vừa duy trì được sự chủ động của con người, từ đó trở thành một công cụ hỗ trợ đắc lực cho việc ra quyết định trong tổ chức.

1.2 Thành phần và cách hoạt động

a) Thành phần

Một hệ thống hỗ trợ ra quyết định (DSS) thường bao gồm ba thành phần cốt lõi, đảm bảo sự phối hợp chặt chẽ giữa dữ liệu, mô hình phân tích và khả năng tương tác với người dùng:

- ***Cơ sở dữ liệu:***

Cơ sở dữ liệu là nơi lưu trữ tất cả thông tin liên quan đến vấn đề cần giải quyết. Đây có thể là dữ liệu nội bộ của doanh nghiệp (như doanh số bán hàng, thông tin khách hàng, dữ liệu sản xuất) hoặc dữ liệu từ các nguồn bên ngoài (như dữ liệu thị trường, xu hướng ngành).

- ***Hệ thống quản lý mô hình:***

Cho phép người dùng tạo ra, lưu trữ, và sử dụng các mô hình khác nhau để phân tích dữ liệu và dự báo kết quả.

- ***Giao diện người dùng:***

Giao diện người dùng cho phép điều hướng hệ thống dễ dàng. Mục tiêu chính của giao diện người dùng DSS là giúp người dùng dễ dàng thao tác dữ liệu được lưu trữ trên đó. Doanh nghiệp có thể sử dụng giao diện này để đánh giá hiệu quả của các giao dịch DSS đối với người dùng cuối. Giao diện DSS bao gồm các cửa sổ đơn giản, giao diện điều khiển menu phức tạp và giao diện dòng lệnh.

Ba thành phần này kết hợp với nhau tạo nên một hệ thống DSS hoàn chỉnh, vừa có khả năng xử lý dữ liệu, vừa cung cấp công cụ phân tích mạnh mẽ, đồng thời duy trì tính thân thiện với người sử dụng.

b) Cách hoạt động

Quy trình hoạt động của một hệ thống hỗ trợ ra quyết định (DSS) thường được triển khai theo các bước cơ bản sau:

- ***Thu thập dữ liệu:***

Hệ thống lấy dữ liệu từ nhiều nguồn khác nhau, bao gồm cơ sở dữ liệu nội bộ, dữ liệu theo thời gian thực và cả dữ liệu bên ngoài như thị trường, ngành nghề. Mục tiêu là xây dựng một nền tảng thông tin đầy đủ và có cấu trúc để phục vụ phân tích.

- ***Xử lý và phân tích dữ liệu:***

Các dữ liệu được tổng hợp, làm sạch và chuẩn hóa trước khi đưa vào phân tích. DSS sử dụng nhiều công cụ và mô hình khác nhau (tài chính, thống kê, mô phỏng) để nhận diện mối quan hệ, xu hướng và các yếu tố tiềm ẩn.

- ***Cung cấp thông tin và mô hình hóa:***

Sau khi phân tích, DSS trình bày kết quả dưới dạng báo cáo, bảng biểu hoặc trực quan hóa dữ liệu. Đồng thời, hệ thống cho phép mô hình hóa các kịch bản "nếu – thì", giúp người dùng dự báo và so sánh kết quả của từng phương án.

- ***Hỗ trợ đưa ra quyết định:***

Cuối cùng, DSS cung cấp các gợi ý hoặc phương án khả thi để người dùng tham khảo. Tuy nhiên, quyết định cuối cùng vẫn thuộc về con người; hệ thống chỉ đóng vai trò là công cụ hỗ trợ, mang đến cơ sở dữ liệu và lập luận logic để tăng tính chính xác.

DSS hoạt động như một chu trình khép kín, bắt đầu từ việc thu thập dữ liệu, xử lý – phân tích, mô hình hóa tình huống và kết thúc ở việc hỗ trợ con người đưa ra quyết định tối ưu.

1.3 Vai trò và lợi ích và ưu nhược điểm

1.3.1 Vai trò và lợi ích

Hệ thống hỗ trợ ra quyết định (DSS) đóng vai trò quan trọng trong việc nâng cao chất lượng và hiệu quả của quá trình ra quyết định trong tổ chức. Những vai trò và lợi ích chính có thể kể đến như sau:

- ***Cải thiện chất lượng quyết định:***

DSS cung cấp dữ liệu chính xác, phân tích logic và các mô hình dự báo, từ đó giúp người dùng đưa ra quyết định sáng suốt và khoa học hơn.

- ***Tiết kiệm thời gian:***

Thay vì mất nhiều công sức tổng hợp thủ công, DSS tự động hóa quá trình thu thập và xử lý dữ liệu, giúp nhà quản lý tập trung vào việc phân tích và lựa chọn phương án.

- ***Tăng tính linh hoạt:***

DSS cho phép mô hình hóa nhiều kịch bản "nếu – thì", từ đó hỗ trợ người dùng nhanh chóng thích ứng với sự thay đổi của môi trường kinh doanh.

- ***Nâng cao khả năng cạnh tranh:***

Doanh nghiệp sử dụng DSS có thể nắm bắt cơ hội thị trường sớm hơn, đưa ra quyết định chiến lược chính xác hơn và hạn chế rủi ro so với đối thủ.

- ***Hỗ trợ hợp tác và ra quyết định nhóm:***

Nhiều hệ thống DSS hiện nay có chức năng cộng tác, giúp các nhà quản lý ở nhiều cấp độ cùng tham gia đánh giá phương án và đi đến thống nhất.

Nhìn chung, DSS không chỉ là một công cụ công nghệ, mà còn là một giải pháp chiến lược giúp các tổ chức tối ưu hóa quy trình ra quyết định, tăng cường năng lực quản lý và nâng cao hiệu quả hoạt động.

1.3.2 Ưu, nhược điểm của DSS

a) Ưu điểm

Hệ thống hỗ trợ ra quyết định (DSS) mang đến nhiều lợi ích rõ rệt cho các tổ chức:

- ***Đưa ra quyết định sáng suốt hơn:***

DSS tổng hợp và phân tích dữ liệu từ nhiều nguồn khác nhau, nhờ đó cung cấp thông tin đáng tin cậy để người dùng lựa chọn phương án tối ưu.

- ***Xem xét các kịch bản khác nhau:***

Thông qua việc dựa trên dữ liệu hiện tại và lịch sử, hệ thống cho phép mô phỏng nhiều kết quả kinh doanh khác nhau.

- ***Tăng hiệu quả xử lý:***

DSS tự động hóa quá trình phân tích các tập dữ liệu lớn, giúp tiết kiệm thời gian và công sức cho nhà quản lý.

- ***Hỗ trợ cộng tác:***

Nhiều công cụ DSS tích hợp chức năng giao tiếp, giúp các nhóm hoặc phòng ban cùng tham gia thảo luận và đánh giá.

- ***Tính linh hoạt cao:***

Có thể ứng dụng trong nhiều ngành công nghiệp khác nhau như tài chính, marketing, logistics, y tế…

- ***Giải quyết các vấn đề phức tạp:***

DSS có khả năng xử lý những tình huống nhiều biến số, nhiều ràng buộc và mối quan hệ chồng chéo.

Những ưu điểm này cho thấy DSS là một công cụ mạnh mẽ, góp phần nâng cao năng lực phân tích và hiệu quả ra quyết định trong tổ chức.

b) Nhược điểm

Bên cạnh những lợi ích mang lại, DSS cũng tồn tại một số hạn chế đáng lưu ý:

- ***Chi phí cao:***

Việc phát triển, triển khai và duy trì DSS đòi hỏi nguồn lực tài chính lớn, gây khó khăn cho các tổ chức vừa và nhỏ.

- ***Nguy cơ phụ thuộc:***

Nếu quá dựa dẫm vào hệ thống, nhà quản lý có thể đánh mất tính chủ quan và kinh nghiệm cá nhân trong việc ra quyết định.

- ***Độ phức tạp trong thiết kế và triển khai:***

DSS cần lượng dữ liệu lớn, đa chiều, khiến việc xây dựng và vận hành trở nên khó khăn.

- ***Rủi ro bảo mật:***

Do DSS thường xử lý dữ liệu nhạy cảm, yêu cầu bảo mật và quản lý quyền truy cập phải được đảm bảo chặt chẽ.

Có thể thấy, dù DSS mang lại nhiều lợi ích thiết thực, nhưng để khai thác tối đa giá trị, tổ chức cần cân nhắc kỹ giữa chi phí, khả năng triển khai và mức độ sẵn sàng của nguồn nhân lực.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Thuật toán Naive Bayes đóng vai trò quan trọng trong lĩnh vực học máy, đặc biệt là các bài toán phân loại. Với nền tảng từ định lý Bayes, thuật toán này cho phép xây dựng mô hình dự đoán nhanh chóng, hiệu quả và có khả năng áp dụng rộng rãi. Sự phát triển của các biến thể như Multinomial, Bernoulli, Gaussian hay Complement giúp Naive Bayes thích ứng với nhiều dạng dữ liệu khác nhau, từ văn bản rời rạc cho đến dữ liệu liên tục. Trong chương này, chúng ta sẽ lần lượt tìm hiểu cơ sở lý thuyết của Naive Bayes và các biến thể tiêu biểu của nó.

2.1 Tổng quan về thuật toán Naive Bayes

Naive Bayes là một trong những thuật toán phân loại dựa trên lý thuyết xác suất, được xây dựng từ định lý Bayes. Ý tưởng cơ bản của thuật toán là dự đoán lớp của một mẫu dữ liệu mới bằng cách tính toán xác suất hậu nghiệm của từng lớp, rồi chọn lớp có xác suất cao nhất.

Định lý Bayes được phát biểu như sau:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

Trong đó:

- $\boldsymbol{P(C|X)}$ **- Xác suất hậu nghiệm** (posterior probability): xác suất mà giả thuyết hoặc lớp C là đúng sau khi quan sát dữ liệu X. Đây chính là thứ ta muốn tính trong phân loại.
- $\boldsymbol{P(X|C)}$ **– Độ khả dĩ** (likelihood): xác suất quan sát được dữ liệu X nếu giả thuyết C đúng. Nó phản ánh mức độ dữ liệu phù hợp với từng lớp.
- $\boldsymbol{P(C)}$ **– Xác suất tiên nghiệm** (prior probability): xác suất ban đầu của lớp C trước khi có dữ liệu mới. Thường được ước lượng từ tần suất xuất hiện của lớp trong tập huấn luyện.
- $\boldsymbol{P(X)}$ **- Bằng chứng** (evidence): xác suất quan sát dữ liệu X trong toàn bộ không gian, đóng vai trò là hằng số chuẩn hóa để đảm bảo tổng xác suất bằng 1.

Ý nghĩa trực quan: Định lý Bayes cho phép kết hợp kiến thức đã biết trước (prior) với bằng chứng quan sát mới (likelihood) để tính ra niềm tin cập nhật (posterior).

Nhờ sự đơn giản, tốc độ xử lý nhanh và hiệu quả trên các tập dữ liệu lớn, thuật toán Naive Bayes được ứng dụng rộng rãi trong nhiều bài toán thực tế, tiêu biểu như:

- Phân loại văn bản: Đây là lĩnh vực ứng dụng mạnh mẽ nhất của Naive Bayes. Ngoài lọc email rác (spam filtering) là trọng tâm của đề tài này, nó còn được dùng để phân tích cảm xúc (sentiment analysis) của bình luận, đánh giá sản phẩm, hoặc tự động phân loại tin tức theo chủ đề (thể thao, chính trị, công nghệ).
- Chẩn đoán y khoa: Naive Bayes có thể được dùng để xây dựng các mô hình dự đoán nguy cơ mắc một loại bệnh (ví dụ: tiểu đường, bệnh tim) dựa trên các triệu chứng và tiền sử của bệnh nhân.

- Hệ thống đề xuất (Recommendation Systems): Thuật toán có thể được sử dụng để gợi ý sản phẩm cho người dùng dựa trên lịch sử mua hàng hoặc các mặt hàng họ đã xem.
- Nhận dạng khuôn mặt: Mặc dù các phương pháp hiện đại hơn đã xuất hiện, Naive Bayes vẫn có thể được dùng trong các bước đầu của bài toán nhận dạng hình ảnh.

Naive Bayes có nhiều biến thể để phù hợp với từng loại dữ liệu khác nhau. Bốn biến thể phổ biến nhất là: **Multinomial Naive Bayes, Bernoulli Naive Bayes, Gaussian Naive Bayes** và **Complement Naive Bayes**. Các biến thể này sẽ được trình bày chi tiết trong các mục tiếp theo.

2.2 Multinomial Naive Bayes

2.2.1 Định nghĩa

Multinomial Naive Bayes (MNB) là một biến thể của thuật toán Naive Bayes, thường được sử dụng trong bài toán phân loại văn bản. Ý tưởng chính của mô hình là:

- Mỗi văn bản được coi như một "túi từ" (bag-of-words), nghĩa là ta chỉ quan tâm từ gì xuất hiện và xuất hiện bao nhiêu lần, không xét đến thứ tự sắp xếp.
- Mỗi từ trong từ vựng (vocabulary) được xem như một đặc trưng (feature). Văn bản sẽ được biểu diễn bằng một vector đặc trưng, trong đó giá trị của mỗi phần tử là số lần từ đó xuất hiện.
- Giả định quan trọng của Naive Bayes là tính độc lập có điều kiện: sự xuất hiện của một từ không ảnh hưởng đến sự xuất hiện của các từ khác, khi đã biết nhãn lớp của văn bản.

Chính nhờ giả định đơn giản này, Multinomial Naive Bayes có thể tính toán nhanh chóng, hiệu quả ngay cả với tập dữ liệu rất lớn.

2.2.2 Công thức Multinomial Naive Bayes

a) Áp dụng định lý Bayes trong Multinomial Naive Bayes

Công thức phân loại trong Naive Bayes được biểu diễn như sau:

$$P(C = c_j \mid d) = \frac{P(C = c_j \mid d) \cdot P(d \mid C = c_j)}{P(d)}$$

Trong đó:

- $C = c_j$ là một lớp cụ thể, ví dụ trong bài toán lọc thư lừa đảo có thể là *Spam* hoặc *Phishing*.
- d là văn bản cần phân loại.
- $P(C = c_j \mid d)$ là xác suất hậu nghiệm, cho biết khả năng văn bản d thuộc về lớp c_j.
- $P(C = c_j)$ là xác suất tiên nghiệm của lớp c_j, thường được ước lượng từ tỉ lệ số văn bản trong lớp c_j so với toàn bộ tập huấn luyện.
- $P(d \mid C = c_j)$ là độ khả dĩ (likelihood), thể hiện xác suất quan sát được văn bản d khi biết rằng nó thuộc lớp c_j.

- $P(d)$ là hằng số chuẩn hóa (evidence), đảm bảo tổng xác suất bằng 1. Trong thực tế khi so sánh giữa các lớp, $P(d)$ giống nhau nên có thể bỏ qua.

Như vậy, để xác định lớp cho một văn bản mới, ta chỉ cần tính giá trị $P(C = c_j \mid d) \cdot P(d \mid C = c_j)$ cho từng lớp, rồi chọn lớp có giá trị lớn nhất. Đây chính là nguyên tắc ***Maximum A Posteriori (MAP)*** trong phân loại Bayes.

b) Biểu diễn văn bản bằng mô hình túi từ (Bag-of-Words)

Một văn bản d có thể được biểu diễn dưới dạng vector đặc trưng:

$$d = (x_1, x_2, \dots, x_n)$$

Trong đó:

- n là kích thước từ vựng (*vocabulary size*) của toàn bộ tập dữ liệu.
- x_i là số lần xuất hiện của từ thứ i trong văn bản d (tần suất).

Cách biểu diễn này chính là nền tảng của **Multinomial Naive Bayes**, thường được gọi là **mô hình Túi từ (Bag-of-Words)** dựa trên số đếm. Theo đó, mỗi văn bản được mã hóa thành một vector, trong đó mỗi phần tử x_i phản ánh mức độ xuất hiện của từ tương ứng trong từ vựng. Nhờ đó, các văn bản khác nhau có thể được so sánh và phân loại dựa trên cấu trúc tần suất từ.

c) Xác suất có điều kiện cho các đặc trưng trong Multinomial Naive Bayes

Trong **Multinomial Naive Bayes**, xác suất một văn bản d thuộc lớp $C = c_j$ được tính dựa trên giả định rằng các từ trong văn bản độc lập có điều kiện khi biết lớp. Theo mô hình này, xác suất của văn bản được tính bằng tích xác suất của các từ thành phần, đồng thời phản ánh tần suất xuất hiện của từng từ:

$$P(d \mid C = c_j) = \prod_{i=1}^{n} P(w_i \mid C = c_j)^{x_i}$$

Trong đó:

- n: kích thước từ vựng (số lượng từ duy nhất trong toàn bộ tập dữ liệu).
- w_i: từ thứ i trong từ vựng.
- x_i: số lần từ w_i xuất hiện trong văn bản n.
- $P(w_i \mid C = c_j)$: xác suất từ w_i xuất hiện trong lớp c_j .

Ý nghĩa:

- Số mũ x_i nhấn mạnh tần suất từ: từ xuất hiện nhiều lần ảnh hưởng lớn hơn đến xác suất của văn bản.
- Công thức này phản ánh chính xác cách hoạt động của mô hình **Multinomial**, nơi tần suất từ là yếu tố quan trọng trong phân loại văn bản.
- Những từ xuất hiện thường xuyên (có x_i lớn) sẽ ảnh hưởng mạnh hơn đến xác suất cuối cùng $P(d \mid C = c_j)$.

Ngoài ra, công thức trên còn được biểu diễn dưới dạng tích xác suất của các từ xuất hiện trong văn bản:

$$P(d \mid C = c_j) \approx \prod_{w \in d} P(w \mid C = c_j)$$

Kết quả: Những từ xuất hiện thường xuyên (có x_i lớn) sẽ có ảnh hưởng mạnh mẽ hơn đến xác suất cuối cùng $P(d \mid C = c_j)$. Nếu $P(w_i \mid C = c_j)$ cao, nó sẽ làm tăng xác suất của lớp c_j ; nếu $P(w_i \mid C = c_j)$ thấp, nó sẽ làm giảm xác suất.

Để tính xác suất có điều kiện của từng từ w_i trong lớp c_j , Multinomial Naive Bayes áp dụng **Laplace smoothing**:

$$P(w_i \mid C = c_j) = \frac{count(w_i\ trong\ lớp\ c_j) + \alpha}{count(tất\ cả\ trong\ lớp\ c_j) + \alpha \cdot n}$$

Trong đó:

- $count(w_i\ trong\ lớp\ c_j)$: số lần từ w_i xuất hiện trong tất cả văn bản thuộc lớp c_j. Ví dụ, nếu "spam" xuất hiện 5 lần trong các email spam, thì giá trị này là 5.
- **α (alpha)**: hệ số làm trơn Laplace. Thường α = 1. Mục đích là **tránh xác suất bằng 0** cho các từ chưa xuất hiện trong lớp. Nếu một từ chưa bao giờ xuất hiện trong lớp spam, nhờ α, xác suất vẫn > 0.
- **Tổng số từ trong lớp c_j**: tổng tất cả từ (theo tần suất) trong các văn bản thuộc lớp c_j. Ví dụ, nếu tất cả email spam có tổng cộng 1000 từ, thì đây là 1000.
- **α · n**: phần bù cho tất cả từ trong từ vựng (n = số từ duy nhất trong tập dữ liệu). Phần này đảm bảo **làm trơn tổng quát**, tránh các từ hiếm làm lệch xác suất.

Nhìn chung, việc cộng thêm α vào tử số đảm bảo rằng mọi từ, kể cả những từ chưa từng xuất hiện trong lớp, vẫn có xác suất dương. Đồng thời, việc cộng α nhân với kích thước từ vựng vào mẫu số giúp tổng xác suất sau khi làm trơn vẫn bằng 1. Nhờ cách làm này, mô hình trở nên ổn định hơn, tránh tình trạng một từ hiếm làm "vô hiệu hóa" toàn bộ xác suất văn bản, đồng thời cải thiện khả năng tổng quát hóa, đặc biệt khi phân loại các văn bản chứa nhiều từ lạ hoặc ít gặp.

d) Quy tắc dự đoán lớp dựa trên xác suất hậu nghiệm

Khi phân loại văn bản mới d, thuật toán chọn lớp $\hat{C}$ có **xác suất hậu nghiệm** $P(C = c_j \mid d)$ lớn nhất.

Vì $P(d)$là hằng số cho tất cả các lớp và có thể bỏ qua, quy tắc phân loại trở thành việc tìm lớp cực đại hóa tử số:

$$\hat{C} = \underset{c_j}{arg\max} \left[P(C = c_j) \cdot \prod_{i=1}^{n} P(w_i \mid C = c_j)^{x_i} \right]$$

Trong thực tế, khi nhân quá nhiều xác suất nhỏ lại với nhau (đặc biệt với n lớn, tức từ vựng lớn), kết quả có thể rất gần 0, gây ra lỗi **tràn số dưới (underflow)** trong tính toán máy tính. Để khắc phục, người ta chuyển sang làm việc với **logarithm** (log-space). Việc chuyển từ phép nhân sang phép cộng không làm thay đổi thứ tự lớn nhỏ của kết quả (vì log là hàm đơn điệu tăng), nên lớp được chọn vẫn giữ nguyên:

$$\hat{C} = \underset{c_j}{arg\max} \left[\log P(C = c_j) + \sum_{i=1}^{n} x_i \cdot \log P(w_i \mid C = c_j) \right]$$

Giải thích:

- $\log P(C = c_j)$: log-xác suất tiên nghiệm của lớp c_j (ví dụ: tỉ lệ email spam trong toàn bộ dữ liệu).
- $\sum_{i=1}^{n} x_i \cdot \log P(w_i \mid C = c_j)$: tổng log-xác suất có trọng số.
  - x_i: số lần từ w_i xuất hiện trong văn bản.
  - $P(w_i \mid C = c_j)$: xác suất xuất hiện từ w_i khi văn bản thuộc lớp c_j.

Việc chuyển đổi công thức sang log-space trước hết giúp khắc phục hiện tượng **underflow**, từ đó đảm bảo tính ổn định trong quá trình tính toán số học. Nhờ tính chất $\boldsymbol{log(AB) = logA + logB}$, các phép nhân giữa nhiều xác suất nhỏ được thay thế bằng phép cộng giữa các log-xác suất, giúp công thức trở nên đơn giản và dễ xử lý hơn. Đồng thời, theo $\boldsymbol{log(A^x) = x \cdot logA}$, các số mũ thể hiện tần suất từ được chuyển thành hệ số nhân với log-xác suất, làm giảm độ phức tạp trong tính toán. Bên cạnh đó, vì logarit là một hàm đơn điệu tăng, thứ tự so sánh giữa các lớp không bị thay đổi, do đó kết quả phân loại vẫn được giữ nguyên so với công thức gốc. Cách biểu diễn trong log-space không chỉ nâng cao độ chính xác mà còn góp phần đơn giản hóa và tăng tốc quá trình xử lý, đặc biệt hữu ích khi áp dụng cho các tập dữ liệu văn bản có kích thước lớn.

2.2.3 Quy trình huấn luyện Multinomial Naive Bayes

Quá trình huấn luyện Multinomial Naive Bayes được tiến hành theo một chuỗi bước có tính hệ thống, trong đó mỗi bước đều giữ vai trò nền tảng cho việc xây dựng một mô hình phân loại hiệu quả. Trước hết, dữ liệu huấn luyện cần được thu thập và gán nhãn đầy đủ, sau đó trải qua các công đoạn tiền xử lý như loại bỏ ký tự đặc biệt, chuẩn hóa định dạng văn bản hay loại bỏ các từ dừng ít mang giá trị ngữ nghĩa. Các thao tác này nhằm đảm bảo dữ liệu đầu vào có chất lượng tốt, phản ánh chính xác đặc trưng của từng lớp.

Trên cơ sở đó, hệ thống tiến hành xây dựng tập từ vựng và biểu diễn dữ liệu văn bản dưới dạng vector Bag-of-Words. Phương pháp này tuy giản lược cấu trúc ngữ pháp nhưng lại cho phép mô hình tập trung vào tần suất xuất hiện của từ, yếu tố được coi là quan trọng trong việc phân biệt nội dung giữa các lớp.

Bước tiếp theo là ước lượng xác suất tiên nghiệm của từng lớp dựa trên tỷ lệ phân bố của dữ liệu huấn luyện. Thông tin này cung cấp cho mô hình một bức tranh tổng quan về mức độ phổ biến của các lớp, từ đó định hướng quá trình phân loại. Song song với đó, xác suất có điều kiện của từng từ trong từ vựng đối với mỗi lớp cũng được ước lượng. Kỹ thuật làm trơn Laplace được áp dụng ở giai đoạn này nhằm khắc phục tình trạng một số từ chưa từng xuất hiện trong lớp dẫn đến xác suất bằng không, đồng thời nâng cao khả năng khái quát hóa của mô hình.

Cuối cùng, mô hình kết hợp xác suất tiên nghiệm và xác suất có điều kiện để hình thành công thức phân loại hoàn chỉnh. Khi một văn bản mới được đưa vào, mô hình sẽ tính toán xác suất hậu nghiệm cho từng lớp và gán nhãn văn bản dựa trên lớp có xác suất cao nhất. Quy trình này tạo nên một cơ chế phân loại vừa đơn giản vừa hiệu quả, đặc biệt thích hợp với các tập dữ liệu văn bản có quy mô lớn.

2.3 Bernoulli Naive Bayes

2.3.1 Định nghĩa

Bernoulli Naive Bayes (BNB) là một thuật toán học máy thuộc họ Naive Bayes, được xây dựng dựa trên nguyên lý Bayes và giả định độc lập có điều kiện giữa các đặc trưng. Điểm khác biệt chính của BNB nằm ở chỗ mô hình được thiết kế để xử lý dữ liệu nhị phân (binary features), nơi mà mỗi đặc trưng chỉ có hai trạng thái: xuất hiện (1) hoặc không xuất hiện (0). Do đó, Bernoulli NB đặc biệt phù hợp cho các bài toán mà sự xuất hiện đơn thuần của một yếu tố đã mang tính phân loại mạnh, thay vì phải đếm tần suất.

Trong lĩnh vực phân loại văn bản, Multinomial Naive Bayes thường được sử dụng khi tần suất xuất hiện của từ quan trọng. Ngược lại, Bernoulli Naive Bayes chỉ quan tâm đến việc từ đó có xuất hiện hay không. Ví dụ, trong phân loại email spam, sự xuất hiện duy nhất của từ 'viagra' có thể đủ để kết luận email là spam, bất kể nó xuất hiện bao nhiêu lần. So với Gaussian NB (dành cho dữ liệu liên tục) và Multinomial NB (dành cho dữ liệu đếm tần suất), Bernoulli NB lấp đầy một khoảng trống quan trọng: dữ liệu dạng nhị phân.

2.3.2 Công thức Bernoulli Naive Bayes

Đối với BNB, vì các đặc trưng x_i là nhị phân (0 hoặc 1), xác suất điều kiện $P(x_i \mid y)$ được tính bằng công thức phân phối Bernoulli:

$$P(x_i \mid y) = P(i \mid y)^{x_i} + \left(1 - P(i \mid y)\right)^{1-x_i}$$

Công thức này bao quát đồng thời cả hai trạng thái của đặc trưng, cho phép mô hình vừa xem xét khả năng xuất hiện, vừa khai thác thông tin từ sự vắng mặt của đặc trưng trong dữ liệu. Nhờ đó, Bernoulli Naive Bayes trở thành lựa chọn phù hợp cho các tập dữ liệu nhị phân, điển hình như phân loại văn bản dựa trên sự có mặt của từ khóa. Khi $\boldsymbol{x_i = 1}$, ta có $P(x_i \mid y) = P(i \mid y)$, còn khi $\boldsymbol{x_i = 0}$, công thức trở thành $P(x_i \mid y) = 1 - P(i \mid y)$.

Quy tắc quyết định cuối cùng là chọn lớp $\hat{y}$ có xác suất hậu nghiệm lớn nhất:

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

2.3.3 Cách thức hoạt động và quy trình học

Quy trình huấn luyện Bernoulli Naive Bayes được triển khai thông qua việc ước lượng các tham số xác suất từ tập dữ liệu huấn luyện. Quá trình này bao gồm ba thành phần cơ bản:

a) Ước lượng xác suất tiên nghiệm (Prior Probability)

Xác suất tiên nghiệm $\mathbf{P(y)}$ biểu thị tỷ lệ mẫu thuộc lớp **y** trong toàn bộ tập dữ liệu, được xác định theo công thức:

$$\mathrm{P(y)} = \frac{N_y}{N}$$

với N_y là số lượng mẫu thuộc lớp y, N là tổng số mẫu. Tham số này phản ánh mức độ phổ biến của từng lớp và đóng vai trò nền tảng trong việc xây dựng phân bố xác suất tổng thể của mô hình.

b) Ước lượng xác suất đặc trưng có điều kiện (Conditional Feature Probability)

Đối với mỗi đặc trưng i, mô hình tiến hành ước lượng xác suất đặc trưng đó xuất hiện khi mẫu thuộc lớp y, ký hiệu là:

$$\mathrm{P}(i \mid \mathrm{y}) = \frac{N_{i,y}}{N_y}$$

trong đó $N_{i,y}$ là số lượng mẫu thuộc lớp y mà đặc trưng i xuất hiện. Cách tiếp cận này cho phép mô hình thiết lập mối quan hệ giữa sự xuất hiện của đặc trưng và nhãn lớp, từ đó định hình nên cấu trúc phân bố của toàn bộ không gian đặc trưng.

c) Làm mịn Laplace (Laplace Smoothing)

Trong trường hợp một đặc trưng không xuất hiện trong bất kỳ mẫu nào của lớp y, việc ước lượng trực tiếp dẫn đến xác suất bằng 0, gây ảnh hưởng nghiêm trọng đến quá trình tính toán hậu nghiệm. Để khắc phục hiện tượng này, kỹ thuật làm mịn Laplace được sử dụng:

$$\mathrm{P}(i \mid \mathrm{y})_{smooth} = \frac{N_{i,y} + \alpha}{N_y + 2\alpha}$$

Trong đó, α là hệ số làm mịn (thông thường chọn $\alpha = 1$), đảm bảo mọi xác suất đều dương và đồng thời cải thiện khả năng tổng quát hóa của mô hình.

Trên cơ sở các tham số xác suất được ước lượng ở các bước trên, Bernoulli Naive Bayes tiến hành tính toán xác suất hậu nghiệm cho từng lớp và xác định nhãn phân loại dựa trên nguyên tắc Maximum A Posteriori (MAP). Cơ chế này cho phép mô hình tận dụng không chỉ thông tin từ sự xuất hiện của đặc trưng mà còn cả giá trị phân loại từ sự vắng mặt của nó, qua đó nâng cao độ chính xác trong các bài toán xử lý dữ liệu nhị phân.

2.4 Gaussian Naive Bayes

Gaussian Naive Bayes là một biến thể của thuật toán Naive Bayes được thiết kế để xử lý dữ liệu có đặc trưng dạng liên tục thay vì rời rạc. Trong khi Multinomial Naive Bayes thường được sử dụng cho các bài toán phân loại văn bản dựa trên tần suất từ, Gaussian Naive Bayes lại giả định rằng các giá trị đặc trưng tuân theo phân phối chuẩn (Gaussian distribution) trong từng lớp. Cụ thể, với một đặc trưng liên tục x, xác suất có điều kiện được ước lượng theo công thức:

$$P(x \mid C = c_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} exp\left(-\frac{(x - \mu_j)^2}{2\sigma_j^2}\right)$$

Trong đó, μ_j và σ_j^2 lần lượt là trung bình và phương sai của đặc trưng x trong lớp $\boldsymbol{c_j}$. Việc sử dụng mô hình phân phối chuẩn giúp thuật toán có thể mô tả tốt các dữ liệu liên tục như chiều cao, cân nặng, nhiệt độ hay các chỉ số sinh trắc

học khác. Khi dự đoán, Gaussian Naive Bayes kết hợp các xác suất có điều kiện này với xác suất tiên nghiệm của mỗi lớp, sau đó áp dụng quy tắc Bayes để tìm lớp có xác suất hậu nghiệm lớn nhất.

Ưu điểm nổi bật của Gaussian Naive Bayes là tốc độ huấn luyện nhanh, do chỉ cần ước lượng giá trị trung bình và phương sai cho từng đặc trưng trong mỗi lớp, đồng thời mô hình có khả năng hoạt động hiệu quả ngay cả khi số lượng dữ liệu huấn luyện không lớn. Thuật toán này đặc biệt hữu ích trong các tình huống giả định phân phối chuẩn là hợp lý, ví dụ trong y học (phân loại bệnh dựa trên chỉ số xét nghiệm), trong sinh học (phân tích dữ liệu gene), hoặc trong các ứng dụng IoT khi dữ liệu cảm biến thường có dạng liên tục. Tuy nhiên, nhược điểm là giả định về phân phối chuẩn đôi khi không đúng với dữ liệu thực tế, dẫn đến độ chính xác giảm khi phân phối thực sự lệch hoặc có nhiều đỉnh.

Nhìn chung, Gaussian Naive Bayes là một mô hình đơn giản nhưng mạnh mẽ, kết hợp được ưu điểm của tính "ngây thơ" trong giả định độc lập đặc trưng với sự linh hoạt trong xử lý dữ liệu liên tục, từ đó trở thành một công cụ cơ bản và hữu ích trong nhiều bài toán phân loại thực tiễn.

2.5 Complement Naive Bayes

Complement Naive Bayes (CNB) là một biến thể của Multinomial Naive Bayes (MNB), được Rennie và cộng sự (2003) đề xuất nhằm cải thiện hiệu quả phân loại văn bản trong trường hợp dữ liệu mất cân bằng lớp. Điểm khác biệt cốt lõi của CNB là thay vì ước lượng phân phối xác suất từ trong chính lớp c_j, mô hình sử dụng phần bù (complement) của lớp đó, tức toàn bộ dữ liệu ngoại trừ c_j.

Công thức ước lượng xác suất có điều kiện của một từ w_i trong lớp c_j được xác định như sau:

$$\hat{P}(w_i \mid C = c_j) = \frac{count(w_i \; trong \; \overline{c_j}) + \alpha}{\sum_k count(w_i \; trong \; \overline{c_j}) + a \cdot n}$$

Trong đó:

- $\overline{c_j}$: phần bù của lớp c_j.
- n: kích thước từ vựng.
- α: hệ số làm trơn Laplace.

Khi phân loại một tài liệu mới $\boldsymbol{d}$, Complement Naive Bayes (CNB) không trực tiếp ước lượng xác suất hậu nghiệm như trong MNB mà thay vào đó tính toán **hàm điểm số** cho từng lớp. Điểm số của lớp c_j được xác định bởi:

$$Score(c_j) = \sum_{w_i \in d} x_i . \log \hat{P}(w_i \mid C = c_j)$$

trong đó x_i là số lần xuất hiện của từ w_i trong tài liệu d, còn $\hat{P}(w_i \mid C = c_j)$ được ước lượng từ phần bù của lớp x_i. Mô hình sẽ gán tài liệu d cho lớp có điểm số **thấp nhất**, phản ánh rằng các từ trong văn bản ít "tương đồng" nhất với phần bù và do đó có khả năng thuộc về chính lớp c_i.

Ý nghĩa của cơ chế này nằm ở chỗ CNB nhấn mạnh khả năng **phân biệt giữa các lớp**, thay vì chỉ đo mức độ phù hợp nội tại của văn bản với một lớp. Nhờ đó, CNB có khả

năng giảm thiểu thiên lệch do sự mất cân bằng dữ liệu và cải thiện độ chính xác trong các bài toán phân loại văn bản quy mô lớn. Đồng thời, CNB vẫn giữ được những đặc tính quan trọng của họ Naive Bayes như: mô hình đơn giản, dễ triển khai, tốc độ huấn luyện và suy luận nhanh. Tuy nhiên, cũng giống như MNB, CNB bị giới hạn ở dữ liệu rời rạc và ít phù hợp trong bối cảnh dữ liệu liên tục.

CHƯƠNG 3. THỰC NGHIỆM VÀ ỨNG DỤNG

Chương này trình bày các thực nghiệm và ứng dụng của mô hình Naive Bayes trong bài toán phân loại văn bản. Thực nghiệm được tiến hành theo hai hướng: ***triển khai trực tiếp thuật toán*** *nhằm minh họa cơ chế hoạt động, và* ***ứng dụng thực tế*** *áp dụng thư viện hỗ trợ với mô hình Multinomial Naive Bayes – biến thể phổ biến và phù hợp nhất cho dữ liệu văn bản. Qua đó, chương này không chỉ kiểm chứng tính đúng đắn của cơ sở lý thuyết mà còn đánh giá hiệu quả ứng dụng mô hình trong bối cảnh thực tế.*

3.1 Thực nghiệm

a) Dữ liệu thực nghiệm

Trong quá trình nghiên cứu và thực nghiệm, việc lựa chọn tập dữ liệu nhỏ gọn có ý nghĩa quan trọng để làm rõ các bước tính toán của mô hình. Thay vì sử dụng các kho dữ liệu lớn và phức tạp, nghiên cứu này khai thác một tập văn bản ngắn được gán nhãn "spam" và "ham".

Cụ thể, tập dữ liệu bao gồm sáu câu tin nhắn với nội dung đa dạng, trong đó ba câu thuộc nhóm "spam" (liên quan đến quảng cáo, khuyến mãi hoặc kêu gọi truy cập liên kết) và ba câu thuộc nhóm "ham" (phản ánh thông tin công việc hoặc trao đổi hằng ngày). Ví dụ:

- Spam: "Bạn đã trúng thưởng 100 triệu! Nhấp vào link nhận ngay!", "Giảm giá 50% cho đơn hàng hôm nay", "Nhận quà tặng miễn phí, click để nhận".
- Ham: "Mời bạn tham dự hội thảo về AI tuần tới", "Cuộc họp nhóm lúc 14h chiều nay", "Báo cáo tài chính đã được gửi qua email".

```
bayes.ipynb
data = [
    ("Bạn đã trúng thưởng 100 triệu! Nhấp vào link nhận ngay!", "spam"),
    ("Giảm giá 50% cho đơn hàng hôm nay", "spam"),
    ("Nhận quà tặng miễn phí, click để nhận", "spam"),
    ("Mời bạn tham dự hội thảo về AI tuần tới", "ham"),
    ("Cuộc họp nhóm lúc 14h chiều nay", "ham"),
    ("Báo cáo tài chính đã được gửi qua email", "ham")
]
```

Hình 3.1.Tập dữ liệu thực nghiệm

Việc sử dụng tập dữ liệu nhỏ như trên giúp thuận tiện trong việc minh họa các thao tác tiền xử lý, tính toán xác suất trong mô hình Multinomial Naive Bayes, cũng như kiểm chứng khả năng phân loại ở mức cơ bản.

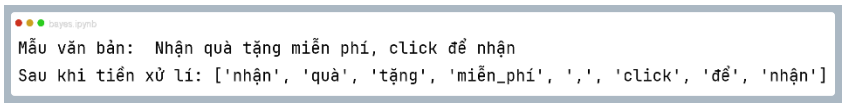
b) Quy trình thực hiện

Quy trình thực nghiệm theo thuật toán Naive Bayes được tiến hành qua các bước sau:

- *Bước 1: Tiền xử lý dữ liệu*

Văn bản đầu vào thường chứa các yếu tố không cần thiết như khoảng trắng thừa, sự khác biệt về chữ hoa/chữ thường hoặc các ký tự đặc biệt. Do đó, trước

tiên văn bản được chuẩn hóa: loại bỏ khoảng trắng dư thừa, đưa toàn bộ về chữ thường, và tách thành các từ riêng biệt.

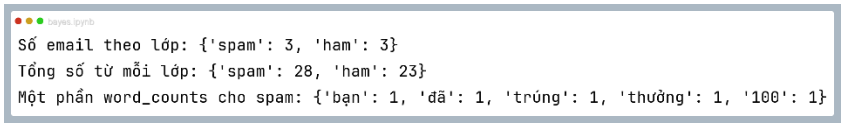


Hình 3.2. Kết quả sau tiền xử lí thực nghiệm

Quá trình tiền xử lý đã chuyển đổi văn bản gốc thành một danh sách các từ (tokens). Có thể quan sát thấy một số đặc điểm đáng chú ý: Toàn bộ chữ đã được đưa về dạng **chữ thường** (ví dụ: "Nhận" → "nhận"), giúp thống nhất dữ liệu. Cụm từ "miễn phí" được ghép lại thành **token duy nhất "miễn_phí"**, phản ánh khả năng của công cụ tách từ tiếng Việt trong việc nhận diện từ ghép. Các dấu câu như dấu phẩy (,) vẫn được giữ lại dưới dạng token, cho thấy mô hình vẫn coi đây là một đặc trưng tiềm năng. Các từ mang ý nghĩa chính như "quà", "tặng", "nhận" được giữ nguyên, giúp mô hình có thể nhận diện thông điệp liên quan đến khuyến mãi hay quảng cáo.

- *Bước 2: Xác định tập đặc trưng và gán nhãn dữ liệu*

Sau khi tiền xử lý, mỗi văn bản trong tập dữ liệu được gán nhãn thủ công là **spam** (thư rác) hoặc **ham** (thư hợp lệ). Tập dữ liệu này được sử dụng để thống kê đặc trưng (các từ xuất hiện) cho từng lớp.



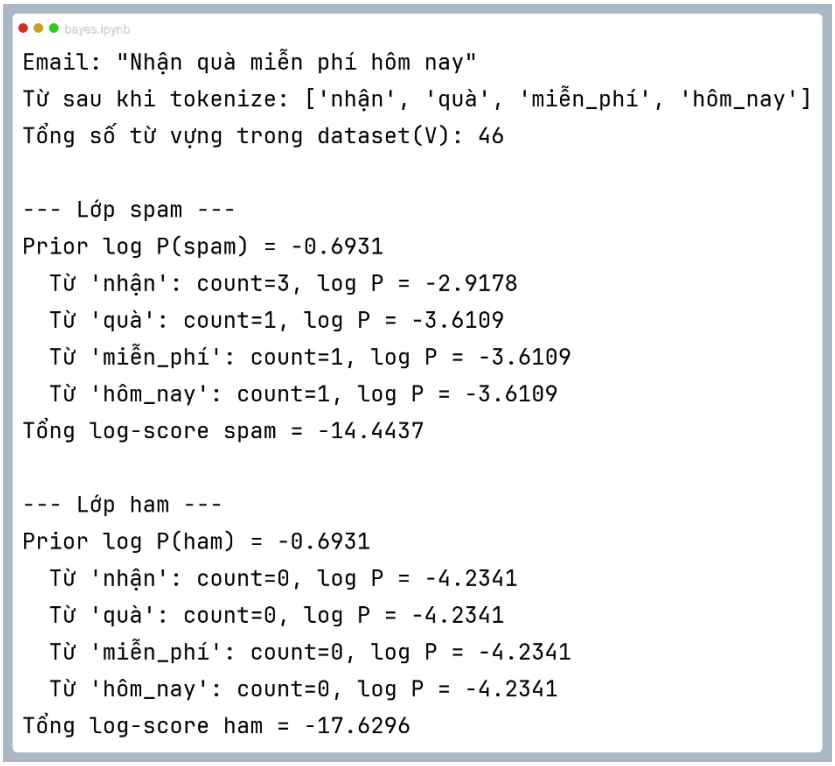
Hình 3.3. Kết quả sau gán nhãn

Kết quả cho thấy dữ liệu huấn luyện được phân bổ cân bằng giữa hai lớp, với 3 văn bản thuộc lớp **spam** và 3 văn bản thuộc lớp **ham**. Việc phân bố cân đối này giúp mô hình tránh tình trạng thiên lệch về một lớp khi học.

Ở cấp độ từ vựng, hệ thống đã ghi nhận tổng cộng **28 từ xuất hiện trong các email spam** và **23 từ trong các email ham**. Các từ điển word_counts phản ánh tần suất của từng từ theo từng lớp. Ví dụ, trong lớp spam, các từ đặc trưng thường liên quan đến quảng cáo hoặc khuyến mãi như "trúng", "thưởng", "100", "giảm giá". Trong khi đó, lớp ham thường chứa từ vựng mang tính thông báo, trao đổi công việc hoặc hội họp. Như vậy, kết quả ở bước này đã hình thành cơ sở dữ liệu thống kê ban đầu cho mô hình, đảm bảo rằng mỗi lớp đều có đặc trưng ngôn ngữ riêng, từ đó hỗ trợ cho các bước tính xác suất và phân loại sau này.

- *Bước 3: Tính toán điểm số xác suất*

Sau khi email được tiền xử lý và tách từ, mô hình Multinomial Naive Bayes tính toán prior log P(C) và log-xác suất có điều kiện của từng từ trong email dựa trên thống kê từ dữ liệu huấn luyện.

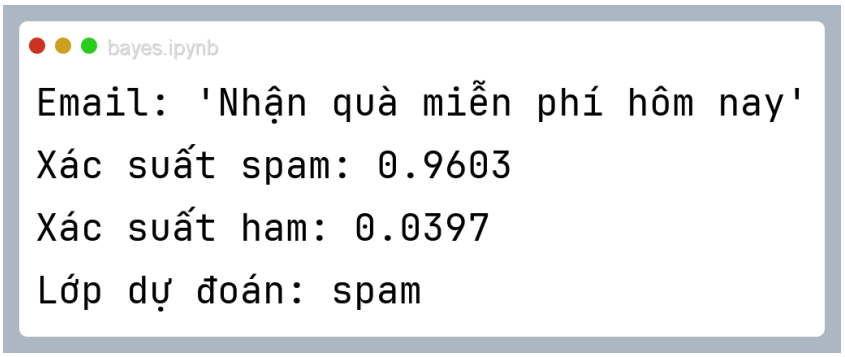


Hình 3.4. Kết quả tính toán thực nghiệm với mô hình Multinomial Naive Bayes

- *Bước 4: Xác định lớp dự đoán*

Sau khi tổng log-score cho từng lớp được tính xong, bước tiếp theo là chuẩn hóa các giá trị này sang xác suất hậu nghiệm để so sánh trực tiếp. Quá trình chuẩn hóa thường sử dụng **softmax**, giúp biến tổng log-score thành xác suất thực có giá trị từ 0 đến 1, thể hiện mức độ phù hợp của email với từng lớp.

Với email mẫu đã được tính toán, kết quả thu được như sau:



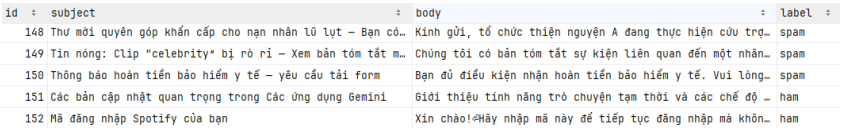
Hình 3.5. Kết quả thực nghiệm

3.2 Ứng dụng thuật toán Naive Bayes trong nhận dạng email spam

Thay vì triển khai thủ công các phép tính xác suất và xử lý văn bản, phần này áp dụng thuật toán Naive Bayes để phân loại email spam và ham, sử dụng các công cụ sẵn có của thư viện **scikit-learn** nhằm tăng tốc quá trình huấn luyện và đánh giá hiệu quả mô hình.

a) Dữ liệu thực nghiệm

Dữ liệu thực nghiệm được lưu dưới dạng CSV, gồm các trường subject, body và label. Trước khi đưa vào mô hình, các trường văn bản được kết hợp thành một trường duy nhất text và các giá trị thiếu được thay bằng chuỗi rỗng, giúp thuận tiện cho quá trình tiền xử lý và huấn luyện mô hình Naive Bayes.



id	subject	body	label
148	Thư mời quyên góp khẩn cấp cho nạn nhân lũ lụt – Bạn có…	Kính gửi, tổ chức thiện nguyện A đang thực hiện cứu trợ…	spam
149	Tin nóng: Clip "celebrity" bị rò rỉ – Xem bản tóm tắt m…	Chúng tôi có bản tóm tắt sự kiện liên quan đến một nhân…	spam
150	Thông báo hoàn tiền bảo hiểm y tế – yêu cầu tải form	Bạn đủ điều kiện nhận hoàn tiền bảo hiểm y tế. Vui lòng…	spam
151	Các bản cập nhật quan trọng trong Các ứng dụng Gemini	Giới thiệu tính năng trò chuyện tạm thời và các chế độ …	ham
152	Mã đăng nhập Spotify của bạn	Xin chào!↵Hãy nhập mã này để tiếp tục đăng nhập mà khôn…	ham

Hình 3.6. Một phần dữ liệu trong dataset

b) Quy trình thực hiện

- *Bước 1: Đọc và chuẩn hóa dữ liệu*

Dữ liệu từ file CSV được nhập vào Python và chuẩn hóa để loại bỏ giá trị trống. Các trường subject và body được kết hợp thành một trường duy nhất, đảm bảo tính nhất quán và sẵn sàng cho các bước tiền xử lý, token hóa và vector hóa tiếp theo.

- *Bước 2: Tiền xử lý và tách từ*

Văn bản được chuẩn hóa bằng cách loại bỏ khoảng trắng thừa và ghép các dòng thành một chuỗi duy nhất. Sau đó, sử dụng thư viện ***underthesea*** để tách văn bản thành các từ (token) tiếng Việt và loại bỏ các token chỉ chứa dấu câu.

- *Bước 3: Tách dữ liệu huấn luyện và kiểm thử*

Tập dữ liệu sau khi tiền xử lý được chia thành hai phần: tập huấn luyện và tập kiểm thử. Việc này được thực hiện bằng hàm ***train_test_split*** của ***scikit-learn***, trong đó một phần dữ liệu được dùng để huấn luyện mô hình Naive Bayes, phần còn lại dùng để đánh giá hiệu suất. Việc tách dữ liệu giúp kiểm soát khả năng tổng quát hóa của mô hình và đánh giá độ chính xác trên dữ liệu chưa từng thấy.

Với tập dữ liệu gốc gồm 250 mẫu, sau khi tách, tập huấn luyện gồm 200 mẫu và tập kiểm thử gồm 50 mẫu, đảm bảo tỉ lệ dữ liệu đủ cho huấn luyện và đánh giá mô hình.

- *Bước 4: Vector hóa văn bản và loại bỏ stopwords*

Sau khi tiền xử lý, văn bản được chuyển thành ma trận đặc trưng số bằng ***TfidfVectorizer***. Mỗi từ trong văn bản được biểu diễn bằng trọng số ***TF-IDF***, phản ánh tầm quan trọng của từ đó trong tập dữ liệu. Đồng thời, các từ dừng (stopwords) tiếng Việt được loại bỏ, nhằm giữ lại những đặc trưng có khả năng phân loại cao và giảm nhiễu cho mô hình.

- *Bước 5: Huấn luyện và kiểm thử mô hình Naive Bayes*

Sau khi văn bản đã được vector hóa, mô hình **Multinomial Naive Bayes** được huấn luyện trên tập dữ liệu huấn luyện. Tham số alpha=0.5 được sử dụng để thực hiện **Laplace smoothing**, tránh trường hợp xác suất bằng 0 đối với các từ chưa xuất hiện trong lớp.

input.ipynb

	precision	recall	f1-score	support
ham	0.94	0.85	0.89	20
spam	0.91	0.97	0.94	30
accuracy			0.92	50
macro avg	0.93	0.91	0.92	50
weighted avg	0.92	0.92	0.92	50

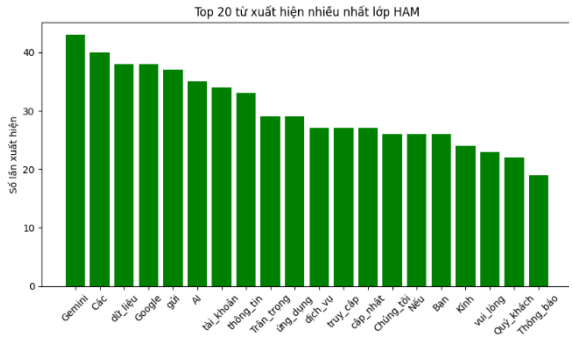
Hình 3.7. Kết quả huấn luyện dữ liệu

Kết quả đánh giá mô hình Naive Bayes trên tập dữ liệu cho thấy hiệu năng khá cao và cân bằng giữa hai lớp. Cụ thể, với lớp **ham**, mô hình đạt precision 0.94, recall 0.85 và f1-score 0.89, phản ánh khả năng nhận diện tin nhắn hợp lệ tốt nhưng vẫn tồn tại tỷ lệ bỏ sót nhất định. Ngược lại, với lớp **spam**, precision đạt 0.91, recall lên tới 0.97 và f1-score 0.94, chứng tỏ mô hình đặc biệt mạnh trong việc phát hiện thư rác, giảm thiểu nguy cơ spam lọt vào hộp thư. Độ chính xác tổng thể (**accuracy**) đạt 92%, đồng thời macro average f1-score ở mức 0.92, cho thấy hiệu năng ổn định và phù hợp cho ứng dụng thực tế.

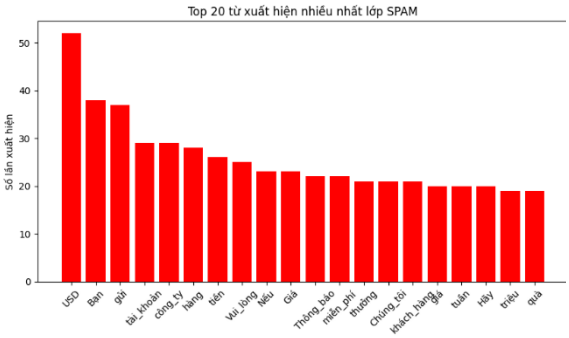
Mô hình cho thấy khả năng nhận diện spam tốt, recall cao đảm bảo ít bỏ sót tin nhắn rác. Tuy nhiên với lớp ham, recall thấp hơn (0.85), đồng nghĩa vẫn có nguy cơ một số tin nhắn hợp lệ bị phân loại nhầm thành spam (false positive).

- *Bước 6: Phân tích từ khóa nổi bật*

Để hiểu rõ hơn cách mô hình phân loại dựa trên đặc trưng ngôn ngữ, ta tiến hành phân tích tần suất xuất hiện từ vựng trong hai lớp **ham** và **spam**. Kết quả thống kê top 20 từ xuất hiện nhiều nhất cho thấy sự khác biệt rõ rệt giữa hai nhóm:



Hình 3.8. Phân tích từ khóa nổi bật lớp Ham

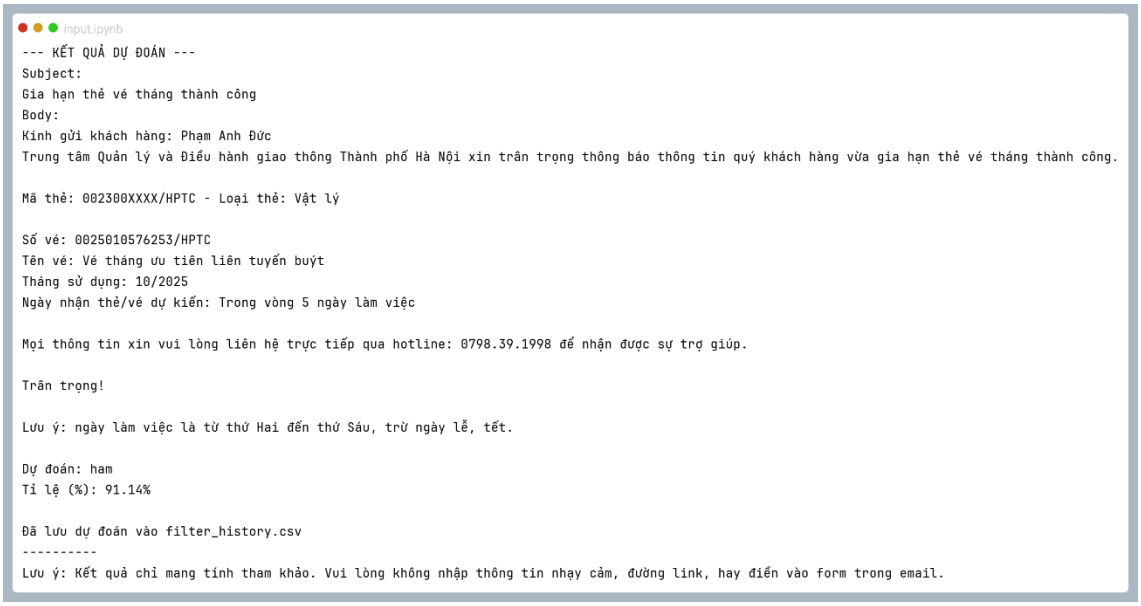


Hình 3.9. Phân tích từ khóa nổi bật lớp Spam

Trong tập dữ liệu đã huấn luyện, với lớp **SPAM**, từ khóa tập trung vào tài chính và dụ dỗ: "USD" (52 lần), "tài_khoản" (29), "công_ty" (28), "tiền" (27), cùng các cụm như "miễn_phí", "khách_hàng". Điều này phản ánh đặc trưng lừa đảo, hứa hẹn lợi ích để lôi kéo. Trong khi đó, lớp **HAM** lại thiên về nội dung công

việc và trao đổi thông tin: “Gemini” (43), “dữ_liệu” (38), “Google” (37), “AI” (35). Các từ như “thông_tin”, “truy_cập” xuất hiện nhiều nhưng mang tính minh bạch. Sự khác biệt này giúp mô hình nhận diện rõ: spam gắn với ngôn ngữ tài chính – khuyến dụ, còn ham thiên về kỹ thuật và trao đổi thực.

- *Bước 7: Dự đoán email mới và lưu kết quả*



Hình 3.10. Kết quả nhận định email mới

Trong ví dụ, email có tiêu đề *“Gia hạn thẻ vé tháng thành công”* với nội dung chi tiết về khách hàng, mã thẻ, ngày sử dụng, cùng thông tin hỗ trợ. Mô hình đã phân loại email này vào lớp **ham** (thư hợp lệ) với độ tin cậy rất cao, lên đến **91.14%**. Điều này cho thấy mô hình nhận diện tốt những thư hành chính – dịch vụ, tránh gán nhầm sang spam.

Ngoài ra, kết quả dự đoán được lưu tự động vào file *filter_history.csv*, đảm bảo khả năng truy vết và đối chiếu khi cần thiết. Cách tiếp cận này giúp việc quản lý kết quả phân loại trở nên hệ thống hơn, đồng thời tạo cơ sở dữ liệu phục vụ phân tích, đánh giá và cải thiện mô hình trong các lần huấn luyện tiếp theo.

	subject	body	pred	probability (%)	time
1	Đăng nhập trên một thiết bị mới	Hi Phạm Anh Đức,↵Tài khoản của bạn đã được sử dụng để đăng nhập trên một thi…	spam	55.20062144333156	2025-10-04 00:50:18
2	Lời nhắc: hãy cập nhật thông tin thanh toán của bạn	Chúng tôi vẫn không thể xử lý khoản thanh toán của bạn.↵Đã xảy ra sự cố với …	spam	64.20687972653994	2025-10-04 00:51:36
3	Gia hạn thẻ vé tháng thành công	Kính gửi khách hàng: Phạm Anh Đức↵Trung tâm Quản lý và Điều hành giao thông …	ham	91.13943763358586	2025-10-04 00:54:39
4	[Bitmart] 🔥 DÀNH RIÊNG CHO【VIỆT NAM】 - Hơn 90% đồng bào	Kính gửi Người dùng BitMart,↵↵Sự kiện ưu đãi siêu hấp dẫn sắp kết thúc! Đừn…	spam	88.3862127197562	2025-10-04 01:04:09

Hình 3.11. Cấu trúc file lịch sử nhận diện

KẾT LUẬN

Trong báo cáo này, nhóm đã tiến hành nghiên cứu và triển khai mô hình Naive Bayes để giải quyết bài toán phân loại thư điện tử. Quá trình thực hiện bao gồm các bước: tiền xử lý dữ liệu, huấn luyện mô hình, đánh giá trên tập kiểm thử và dự đoán trên dữ liệu mới. Kết quả thu được cho thấy mô hình có khả năng phân loại tương đối chính xác giữa thư hợp lệ (ham) và thư rác (spam), đồng thời đảm bảo tốc độ xử lý nhanh và dễ triển khai trên dữ liệu lớn.

Từ góc độ lý thuyết, Naive Bayes dựa trên xác suất hậu nghiệm cùng giả định độc lập giữa các đặc trưng. Nhờ vậy, công thức tính toán trở nên đơn giản, tránh được vấn đề phức tạp khi xử lý dữ liệu văn bản nhiều chiều. Về mặt ứng dụng, mô hình đã chứng minh hiệu quả trong việc hỗ trợ quá trình ra quyết định, đặc biệt là trong các hệ thống lọc thư điện tử và phân tích văn bản tự động.

Tuy nhiên, mô hình vẫn tồn tại những hạn chế nhất định. Giả định độc lập giữa các đặc trưng thường không phản ánh chính xác mối quan hệ thực tế giữa các từ trong văn bản. Điều này có thể ảnh hưởng đến độ chính xác trong một số tình huống. Ngoài ra, chất lượng kết quả còn phụ thuộc nhiều vào khâu tiền xử lý, đặc biệt là bước loại bỏ stopwords và chuẩn hóa dữ liệu.

Trong tương lai, có thể mở rộng nghiên cứu bằng cách thử nghiệm với các biến thể khác của Naive Bayes (như Complement Naive Bayes hoặc Bernoulli Naive Bayes), hoặc kết hợp với các phương pháp học máy khác như SVM hay Logistic Regression. Bên cạnh đó, việc tối ưu hóa tập dữ liệu huấn luyện và bổ sung kỹ thuật xử lý ngôn ngữ tự nhiên nâng cao sẽ giúp cải thiện hiệu năng mô hình, mở ra khả năng ứng dụng rộng rãi hơn trong các hệ trợ giúp quyết định và các bài toán phân tích văn bản khác.

TÀI LIỆU THAM KHẢO

- Decision Support System (DSS): What It Is and How Businesses Use Them
https://www.investopedia.com/terms/d/decision-support-system.asp
- Multinomial Naive Bayes – GeeksforGeeks
https://www.geeksforgeeks.org/machine-learning/multinomial-naive-bayes/
- Multinomial Naive Bayes áp dụng trong classification
https://viblo.asia/p/multinomial-naive-bayes-ap-dung-trong-classification-XL6lAgOpKek
- MultinomialNB
https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html
- Van-Duyet Le. Vietnamese-stopwords
https://github.com/stopwords/vietnamese-stopwords
- Underthesea - Vietnamese NLP Toolkit
https://github.com/undertheseanlp/underthesea
- Naive Bayes classifier - Wikipedia
https://en.wikipedia.org/wiki/Naive_Bayes_classifier