BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH

TRẦN HOÀI NAM

ĐÁNH GIÁ CẢM XÚC CỦA KHÁCH HÀNG DỰA TRÊN BÌNH LUẬN SỬ DỤNG NGÔN NGỮ TỰ NHIÊN

ĐỒ ÁN NGÀNH

NGÀNH KHOA HỌC MÁY TÍNH

TP. HỒ CHÍ MINH, 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH

TRẦN HOÀI NAM

ĐÁNH GIÁ CẢM XÚC CỦA KHÁCH HÀNG DỰA TRÊN BÌNH LUẬN SỬ DỤNG NGÔN NGỮ TỰ NHIÊN

Mã số sinh viên: 2251012097

ĐỒ ÁN NGÀNH

NGÀNH KHOA HỌC MÁY TÍNH

Giảng viên hướng dẫn: PHẠM CHÍ CÔNG

TP. HỒ CHÍ MINH, 2025

LỜI CẢM ƠN

Trước hết, em xin trân trọng cảm ơn Trường đại học Mở Thành phố Hồ Chí Minh đã cho em một môi trường học tập cởi mở, kỉ luật ,giàu tính thực tiễn và đầy đủ các trang thiết bị và cơ sở vật chất tạo điều kiện giúp em tập trung học tập và rèn luyện kiến thức lẫn kĩ năng.

Đồ án ngành này là kết quả mà em học tập được trong ba năm của đại học, đây cũng coi như là sản phẩm chứng minh quá trình học tập và kiến thức thêm của em. Và em cũng luôn nhận được sự giúp đỡ, chỉ bảo tận tình của các thầy cô cũng như sự ủng hộ của các bạn sinh viên và đó cũng là nguồn động lực để em tự tin bước đi trong tương lai.

Đăc biệt, em xin cảm ơn các thầy cô trong Khoa Công nghệ Thông tin cũng như các thầy cô trong trường đã giảng dạy em trong suốt thời gian học tập vừa qua , em xin chân thành và gửi lời cảm ơn đến thầyPhạm Chí Công đã luôn hỗ trợ, giúp đỡ và chỉ ra từng sai sót của em để em hoàn thiện trong quá trình làm đồ án ngành này.

Với những góp ý cụ thể của thầy giúp em trưởng thành hơn trong tư duy và cách thức làm việc. Và chắc chắn sản phẩm của em sẽ không tránh khỏi những thiếu sót, em mong nhận được nhiều ý kiến hơn để đồ án được hoàn thiện tốt hơn.

Em xin trân trọng cảm ơn.

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TÓM TẮT ĐỒ ÁN NGÀNH

Đồ án tập trung nghiên cứu và xây dựng hệ thống **phân tích cảm xúc khách hàng từ bình luận sản phẩm tiếng Việt**, nhằm giúp doanh nghiệp khai thác hiệu quả nguồn dữ liệu phản hồi ngày càng lớn từ thương mại điện tử. Dữ liệu bình luận được thu thập và xử lý qua các bước chuẩn hóa, tách từ, loại bỏ stopwords, xử lý teencode và từ cấm để đảm bảo tính sạch và nhất quán. Sau đó, văn bản được vector hóa bằng phương pháp TF-IDF và đưa vào nhiều mô hình học máy để huấn luyện và đánh giá.

Kết quả thực nghiệm cho thấy **Logistic Regression** là mô hình phù hợp nhất, đạt độ chính xác khoảng 93% trên tập dữ liệu thực tế, đồng thời đơn giản và dễ triển khai. Mô hình được đóng gói và tích hợp vào ứng dụng web Streamlit với các chức năng nổi bật như dự đoán cảm xúc từ bình luận đơn lẻ, xử lý batch dữ liệu từ nhiều định dạng file, trực quan hóa kết quả bằng dashboard và quản lý lịch sử dự đoán. Hệ thống đã chứng minh được tính ứng dụng thực tiễn, có khả năng hỗ trợ doanh nghiệp trong việc nắm bắt nhanh phản hồi của khách hàng và đưa ra quyết định kinh doanh kịp thời.

Mặc dù vẫn còn hạn chế về quy mô dữ liệu và phạm vi phân loại chỉ dừng ở hai nhãn cơ bản, đồ án đã hoàn thành tốt mục tiêu đề ra, đồng thời khẳng định năng lực vận dụng kiến thức về **Data Engineering, NLP và Machine Learning** vào giải quyết một bài toán thực tế. Trong tương lai, hệ thống có thể được mở rộng bằng cách bổ sung dữ liệu lớn hơn, áp dụng các phương pháp vector hóa tiên tiến như Word2Vec hoặc BERT và phát triển phân loại đa nhãn cảm xúc. Nhìn chung, đây là một đề tài mang tính cấp thiết, có giá trị học thuật và ứng dụng cao, thể hiện sự kết hợp chặt chẽ giữa lý thuyết và thực tiễn trong lĩnh vực khoa học dữ liệu.

ABSTRACT

This thesis focuses on the development of a system for **sentiment analysis of Vietnamese customer reviews**, aiming to help businesses effectively utilize the growing volume of feedback data from e-commerce platforms. Customer review data was collected and processed through several steps including text normalization, word segmentation, stopword removal, handling teencode and banned words to ensure data consistency and quality. The cleaned text was then vectorized using the TF-IDF method and evaluated on multiple machine learning models.

Experimental results demonstrate that **Logistic Regression** is the most suitable model, achieving approximately 93% accuracy on real-world datasets while remaining simple and easy to deploy. The trained model was packaged and integrated into a Streamlit-based web application that provides essential functions such as single comment prediction, batch processing of multi-format files, visualization through dashboards, and history management. The system has proven its practical applicability, enabling businesses to quickly capture customer feedback and make timely data-driven decisions.

Although the system still has limitations such as a relatively small dataset and binary sentiment classification, the project successfully achieved its objectives and showcased the ability to apply knowledge in **Data Engineering, NLP, and Machine Learning** to solve a real-world problem. In the future, the system can be enhanced by expanding the dataset, applying advanced representation methods such as Word2Vec or BERT, and extending to multi-class sentiment classification. Overall, this thesis is both academically valuable and practically relevant, reflecting a strong integration of theoretical knowledge and practical implementation in the field of data science.

MỤC LỤC

DANH MỤC TỪ VIẾT TẮT

Viết tắt	Đầy đủ
NLP	Natural Language Processing
AI	Artificial intelligence
ML	Machine Learning
SVM	Support Vector Machine

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG

MỞ ĐẦU

Ngày nay, các trang mạng xã hội, website thương mại điện tử và các nền tảng trực tuyến xuất hiện ngày càng nhiều, trở thành nơi mọi người dễ dàng đưa ra ý kiến, đánh giá hay bình luận về sản phẩm, dịch vụ. Mỗi ngày, hàng triệu bình luận được viết ra, phản ánh cảm xúc thật của người dùng, lúc thì hài lòng, lúc thì thất vọng, thậm chí là tức giận. Đối với doanh nghiệp, việc đọc và hiểu cảm xúc của khách hàng là rất quan trọng. Thông qua đó, họ có thể biết khách hàng có hài lòng hay không, từ đó cải thiện sản phẩm, dịch vụ và xây dựng hình ảnh thương hiệu tốt hơn. Tuy nhiên, số lượng bình luận quá lớn khiến việc đọc bằng tay là không thể. Vì vậy, công nghệ xử lý ngôn ngữ tự nhiên (NLP) ra đời để giúp máy tính tự động nhận diện và phân loại cảm xúc trong các bình luận, đánh giá hoặc bài viết. Nhờ đó, doanh nghiệp có thể đánh giá mức độ hài lòng của khách hàng, theo dõi hình ảnh thương hiệu và điều chỉnh sản phẩm, dịch vụ một cách nhanh chóng và hiệu quả.

Chương 1. GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

1.1. Giới thiệu

Trong bối cảnh thương mại điện tử phát triển mạnh mẻ, lượng dữ liệu từ bình luận và đánh giá các sản phẩm của người tiêu dùng ngày càng tăng. Đây như một kho báu thông tin quan trọng nhằm phản ảnh trải nghiệm, mức độ hài lòng của khách hàng. Tuy nhiên, do khối lượng dữ liệu quá lớn và tính chất phi cấu trúc của văn bản đặc biết là đối với ngôn ngữ tiếng Việt, việc phân tích thủ công là khó khăn và tốn rất nhiều thời gian.

Đề tài này hướng đến việc xây dựng mô hình tự động phân tích cảm xúc của khách hàng từ bình luận sản phẩm giúp phân loại cảm xúc thành các nhóm cơ bản như hài lòng và không hài lòng. Từ đó giúp doanh nghiệp nhanh chóng nắm bắt phản hồi, phát hiện vẫn đề cũng như cải thiện chất lượng sản phẩm và dịch vụ.

Ứng dụng các kỹ thuật trong xử lý ngôn ngữ tự nhiên (NLP) và Machine Learning, kết hợp việc thiết kế pipeline xử lý dữ liệu như thu nhập, làm sạch, trực quan hóa kết quả. Hệ thống không chỉ giúp tự động hóa việc phân tích dữ liệu văn bản mà còn là nơi mà phản hồi của người dùng là chìa khóa then chốt giúp doanh nghiệp phát triển.

1.2. Lí do chọn đề tài

Phản hồi sản phẩm trên các nền tảng thương mại điện tử là nguồn dữ liệu quan trọng phản ánh trải nghiệm của người dùng. Ý kiến, đánh giá của khách hàng ngày càng trở nên có giá trị thiết thực nên việc dự đoán nhận diện cảm xúc của người tiêu dùng là vô cùng quan trọng. Tuy nhiên, do số lượng lớn và dạng căn bản phi cấu trúc nên việc phân tích thủ công gặp nhiều khó khăn. Vì vậy cần có một hệ thống tự động để phân tích ý kiến phản hồi của khách hàng để nắm bắt nhu cầu và thị hiếu của khách hàng qua đó doanh nghiệp sẽ có chiến lược để nâng cao khả năng cạnh tranh với đối thủ và thích ứng với sự biến động không ngừng của thị trường.

Trong nghiên cứu học thuật, việc xây dựng hệ thống này là bước tiến lớn trong xử lý ngôn ngữ tự nhiên giúp giải quyết bài toàn phân tích cảm xúc người dùng. Cụ thể chia cảm xúc khách hàng thành hai trạng thái riêng biệt đối lập bằng phương pháp phân lớp. Mỗi ý kiến, phản hồi diễn đạt cảm xúc của người tiêu dùng được biểu diễn thành một vector để đưa vào huấn luyện.

1.3. Mục tiêu nghiên cứu

Tìm hiểu các lý thuyết cần thiết để xây dựng mô hình giải quyết bài toán nhận diện cảm xúc của người dùng qua bình luận tiếng Việt với 2 nhãn: tích cực và tiêu cực. Bên cạnh đó, mô hình giải quyết được tối ưu về độ chính xác cũng như hiệu suất thời gian thực hiện giúp giải quyết các vấn đề còn mắc phải trong nhận diện cảm xúc khách hàng nói riêng và xử lý ngôn ngữ tiếng Việt nói chung.

1.4. Phạm vi đề tài

Đề tài tập trung vào việc phân tích cảm xúc khách hàng từ bình luận sản phẩm bằng tiếng Việt với dữ liệu được thu thập từ Tiki.vn, phân loại cảm xúc với hai nhãn: tích cực và tiêu cực. Hệ thống được xây dựng theo pipeline end-to-end, bao gồm tiền xử lý văn bản, huấn luyện mô hình Logistic Regression và phát triển ứng dụng web bằng Streamlit để dự đoán cảm xúc và trực quan hóa kết quả.

1.5. Phương pháp nghiên cứu

Sử dụng phương pháp nghiên cứu thực nghiệm, bắt đầu từ thu thập và tiền xử lý dữ liệu. Áp dụng mô hình máy học Logistic Regression cùng với vectorizer và feature engineering để phân loại cảm xúc nhị phân. Được đánh giá trên dữ liệu thực tế nhằm kiểm tra độ chính xác. Cuối cùng, hệ thống được triển khai thành pipeline xử lý end-to-end, hỗ trợ batch processing cho dữ liệu đa định dạng và tích hợp trong ứng dụng web Streamlit với chức năng dự đoán và trực quan hóa kết quả.

Chương 2. CƠ SỞ LÝ THUYẾT

2.1. Khái niệm về phân tích cảm xúc (Sentiment Analysis)

2.1.1. Định nghĩa và ý nghĩa của phân tích cảm xúc

Phân tích cảm xúc là quá trình sử dụng các phương pháp tính toán để tìm cách xác định tâm trạng, thái độ, hoặc quan điểm được thể hiện trong tài liệu, văn bản. Phân tích cảm xúc đã trở thành một công cụ quan trọng giúp các tổ chức, doanh nghiệp biết được sở thích và nhu cầu của người khách hàng khi các phản hồi của khách hàng ngày càng trở nên quan trọng.

Phân tích cảm xúc là một nhánh của khai phá dữ liệu văn bản (Text Mining) và xử lý ngôn ngữ tự nhiên (NLP) tập trung vào việc xác định thái độ từ đó doanh nghiệp có thể đo lường tốt hơn mức hộ hài lòng của người tiêu dùng chẳng hạn như tích cực, tiêu cực hay trung lập. Bằng cách ứng dụng trí tuệ nhân tạo (AI), học máy (ML). Công cụ này có thể áp dụng trên nhiều nguồn thông tin như các bài đăng trên mạng xã hội, email, các bài viết blog, đánh giá trực tuyến, các bình luận đánh giá, phản hồi trên các sàn thương mại điện tử.

Phản hồi của khách hàng vô cùng quan trọng đối với doanh nghiệp vì nó cung cấp lượng lớn thông tin về mức độ hài lòng, sự yêu thích sản phẩm và những khó khăn của khách hàng khi trải nghiệm. Doanh nghiệp có thể cải thiện sản phẩm, dịch vụ cũng như có những chiến lược kinh doanh để cạnh tranh với đối thủ cho phép họ chủ động hơn trong việc giải quyết các vấn đề và tạo lòng tin cho khách hàng.

2.1.2. Vai trò của phân tích cảm xúc trong thương mại điện tử

Xác định và hiểu được cảm xúc được thể hiện trong dữ liệu văn bản trên các nền tảng thương mại điện tử là vai trò lớn nhất mà phân tích cảm xúc đem lại như:

- Xác định mức độ hài lòng của khách hàng đối với sản phẩm.
- Phát hiện sớm các vấn đề liên quan đến chất lượng hoặc dịch vụ.
- Hỗ trợ ra quyết định trong marketing, chăm sóc khách hàng.
- Triển khai chiến lược kinh doanh hợp lý để tăng sự cạnh tranh.
- Dự báo nhu cầu kinh doanh, nghiên cứu thị trường.

Ví dụ:

- “Sản phẩm gia nhanh, chất lượng tuyệt vời” -> nhãn “Hài lòng”
- “Hàng giao không đúng hẹn, bị vỡ một phần” -> nhãn “Không hài lòng”
- “Sản phẩm không có gì đặc sắc, cảm thấy bình thường” -> nhãn “Trung lập”

2.1.3. Các hướng tiếp cận phổ biến

- Dựa trên từ điển cảm xúc

- Sử dụng một tập hợp các từ ngữ gán nhãn tích cực/ tiêu cực sẵn có để tính toán điểm cảm xúc.
- Ví dụ: Từ “tốt”, “tuyệt vời”, “xuất sắc” mang giá trị +1

  Từ “tệ”, “xấu”, “thất vọng” mang giá trị 0
- Ưu điểm: dễ triển khai, không cần dữ liệu huấn luyện lớn
- Nhược điểm: thiếu linh hoạt, không có khả năng mở rộng mô hình, không xử lý tốt các ngữ cảnh phức tạp.

- Dựa trên học máy (Machine Learning)

- Sử dụng dữ liệu lớn đã gán nhãn để huấn luyện cho các mô hình phân loại.
- Mô hình học cách phân biệt cảm xúc dựa trên đặc trưng văn bản.
- Các thuật toán thường dùng: Logistic Regression, Navie Bayes, SVM, Deep Learning,...
- Ưu điểm: linh hoạt, độ chính xác cao nếu dữ liệu đủ lớn.
- Nhược điểm: cần lượng lớn dữ liệu đã gán nhãn.

- Kết hợp: Kết hợp từ điển cảm xúc với học máy để tận dụng tối đa ưu điểm của hai phương pháp.

2.2. Ngôn ngữ tự nhiên và NLP (Natural Language Processing)

2.2.1. Khái niệm ngôn ngữ tự nhiên và NLP

Trong ngôn ngữ học, một ngôn ngữ tự nhiên là bất kỳ ngôn ngữ nào phát sinh, không suy nghĩ trước trong bộ nào con người. Điển hình là một số ngôn ngữ mà con người dùng để giao tiếp với nhau dù là ngôn ngữ nói, ngôn ngữ ký hiệu hay chữ viết khác hoàn toàn với ngôn ngữ nhân tạo như ngôn ngữ máy tính (C, C++, Python,...) hay mã Morse, Braile,...

Theo thống kê, trên thế giới hiện tại có khoảng 5600 ngôn ngữ được phân bố đồng đều và chỉ có một số ít là các ngôn ngữ chữ viết. Ngôn ngữ tự nhiên là một trong những phương tiện giao tiếp quan trọng nhất đối với von người và là một hệ thống các tín hiệu đặc biệt.

Các ngôn ngữ được phân loại:

- Phân loại theo nguồn gốc lịch sử
- Phân loại theo trật tự từ
- Phân loại theo loại hình: được sử dụng phổ biến nhất

Xử lý ngôn ngữ tự nhiên là một lĩnh vực con của khoa học máy tính và trí tuệ nhân tạo (AI) giúp máy tính có thể hiểu, xử lý và tạo ra ngôn ngữ tự nhiên dưới dạng văn bản viết hoặc lời nói. NLP kết hợp Computational Linguistics (ngôn ngữ học tính toán), Rule-based với các phương pháp thống kê, học máy và học sâu để giúp máy tính không chỉ xử lý được ngôn ngữ con người mà còn nắm bắt được ý nghĩa đầy đủ, bao gồm cả ý định và cảm xúc của người nói hoặc người viết.

Xử lý ngôn ngữ tự nhiên là hướng dẫn máy tính thay thế và giúp đỡ con người thực hiện các công việc về xử lý ngôn ngữ như:

- Dịch ngôn ngữ tự động (Google Dịch)
- Trả lời câu hỏi (trợ lí ảo như Siri, Alexa)
- Tóm tắt văn bản
- Phân loại cảm xúc
- Lọc thư rác

- Gợi ý từ tiếp theo khi gõ văn bản (dự đoán văn bản)
- Kiểm tra chính tả
- Nhận diện giọng nói
- Chabot (GPT, Gemini,...)

2.2.2. Đặc thù xử lý văn bản tiếng Việt

So với tiếng Anh hay các ngôn ngữ tự nhiên khác, việc xử lý tiếng Việt gặp nhiều khó khăn, thách thức hơn do đặc điểm ngữ pháp ngôn ngữ của nó:

1. Ngôn ngữ đơn âm tiết có thanh điệu

- Tiếng Việt có 6 thanh điệu (ngang, huyền, sắc, hỏi, ngã, nặng) ảnh hưởng rất nhiều đến ý nghĩa của từ

- Chỉ cần thay đổi dấu đã làm nghĩa của từ thay đổi hoàn toàn

Ví dụ: “má” -> mẹ

“ma” -> hồn ma

“mà” -> liên từ

“mả” -> mộ

Tuy nhiên vẫn có các từ ngữ dù thay đổi dấu nhưng vẫn giữ nguyên nghĩa mặc dù số lượng này rất ít như tứ = tư, lùi = lui,...

2. Thiếu ranh giới từ rõ ràng: Khác với tiếng Anh, tiếng Việt không phải lúc nào cũng phân tách các từ bằng dấu cách, đòi hỏi các kỹ thuật phân đoạn từ tinh vi. Điều này càng phức tạp hơn bơi việc sử dụng phổ biến các từ ghép

Ví dụ: “học sinh” là một từ, nhưng nếu tách thành “học” và “sinh” sẽ mất đi ý nghĩa hoàng toàn

“xe máy” là một từ chỉ phương tiện, không thể tách từ “xe” và “máy”

3. Biến thể phương ngữ: Các phương ngữ vùng miền tạo thêm sự phức tạp, với sự khác biệt về từ vựng và cách phát âm.

4. Từ viết tắt, teencode, biến thể không chuẩn: Người dùng trên mạng thường sử dụng các ký hiệu thay cho chữ viết chuẩn: “k” = “không”, “hok” = “không”, “j” = “gì”, “dc” = “được”, “z” = “vậy”

Ví dụ: “A k mún cta như z đâu” -> Anh không muốn chúng ta như vậy đâu

5. Từ đa nghĩa, ngữ cảnh phụ thuộc rất mạnh: Một từ có thể mang nhiều nghĩa khác nhau tùy vào ngữ cảnh.

Ví dụ: “nho xanh” hiểu đơn giản là chùm nho có màu xanh hoặc là chỉ nho còn non hoặc chỉ người còn kém kinh nghiệm.

6. Sử dụng nhiều từ cảm thán, tục tĩu: Trong bình luận, khách hàng thường dùng cảm thán mạnh hoặc từ ngữ tục tĩu để nhấn mạnh cảm xúc

7. Sử dụng dấu câu: một bình luận nếu không đặt đúng dấu câu sẽ mang lại rất nhiều khó khăn và nghĩa sẽ khác đi hoàn toàn

Ví dụ: “Rắn hổ mang bò lên núi”. Thử đặt dấu câu vào vị trí các từ

“Rắn/ hổ/ mang/ bò/ lên núi” -> con rắn, con hổ, con mang và con bò cùng đi lên núi.

“Rắn hổ mang / bò lên núi” -> con rắn hổ mang đang bò lên núi.

“Rắn hổ/ mang bò lên núi” -> con rắn và con hổ đang mang con bò lên núi.

8. Tài nguyên hạn chế: So với ngôn ngữ như tiếng Anh, các tài nguyên NLP tiếng Việt (bộ dữ liệu, mô hình đào tạo sẵn) còn tương đối khan hiếm, đòi hỏi các cách tiếp cận sáng tạo trong việc tăng cương dữ liệu và đào tạo mô hình

2.2.3. Các kỹ thuật tiền xử lý dữ liệu văn bản

1. Chuẩn hóa văn bản: Văn bản được làm sạch trong giai đoạn này bằng cách xóa các ký hiệu thừa, ký tự dễ nhớ và các ký tự đặc biệt. Ngoài ra, giai đoạn này còn bao gồm việc chuyển toàn bộ văn bản sang chữ thường và loại bỏ bất kỳ số hoặc thành phần phi văn bản nào khác. Chuẩn hóa dữ liệu đảm bảo tính đồng nhất bằng cách chuẩn hóa chính tả, từ viết tắt và các khác biệt khác.

Ví dụ: “Sản pHẩM TốTTTTTT@@@!!!” -> “sản phẩm tốt”

2. Tách từ:

Ví dụ “Mình rất hài lòng với chất lượng của sản phẩm này” -> [“Mình”, “rất”, “hài_lòng”, “với”, “chất_lượng”, “của”, “sản_phẩm”, “này”

3. Loại bỏ Stopwords: Stopwords là những từ xuất hiện nhiều nhưng không mang nhiều ý nghĩa khi phân tích như “những”, “các”, “là”, “có”, “của”,...

Ví dụ: “Những sản phẩm này rất đẹp” ->”sản phẩm rất đẹp”

4. Xử lý các từ phủ định là từ tăng cường cảm xúc: Các từ phủ định như "không", "chưa", "chẳng", "chả" có thể thay đổi ngữ điệu của cả một cụm từ cũng như các từ "siêu", "cực", "rất", "quá", "khá" giúp nhẫn mạnh tăng mức độ cảm xúc nên cần có thẻ riêng biệt để thêm nhằm duy trì ảnh hưởng của chúng trong quá trình phân tích cảm xúc.

Ví dụ: “Tôi không hài lòng với dịch vụ này” mang cảm xúc tiêu cực, và nếu loại bỏ từ “không”, mô hình có thể hiểu sai ý nghĩa cả câu. Hoặc trong câu “Sản phẩm rất tốt và tôi rất hài lòng”, từ “rất” giúp nhấn mạnh, tăng mức độ cảm xúc tích cực.

5. Chuẩn hóa các từ viết tắt: thay thế teencode bằng dạng chuẩn.

6. Xử lý từ cấm, tục tiểu: phát hiện và lọc bỏ, ra thông báo cho người dùng.

2.3. Các phương pháp vectorize dữ liệu

Vector hóa dữ liệu là một trong những bước quan trọng khi xây dựng bài toán phân tích cảm xúc trong văn bản tiếng Việt bằng mô hình máy học. Một lý do cơ bản mà chúng ta cần phải vector hóa văn bản là máy tính không thể hiểu được nghĩa của các từ. Như vậy để xử lý ngôn ngữ tự nhiên chúng ta cần có một phương pháp để biểu 24 diễn văn bản dưới dạng mà máy tính có thể hiểu được. Phương pháp tiêu chuẩn để biểu diễn các văn bản thành các vector. Khi đó các từ hay các cụm từ được ánh xạ thành những vector trong không gian số thực. Quá trình này gọi là **vector hóa dữ liệu văn bản (text vectorization)**. Hai phương pháp phổ biến và cơ bản nhất là **CountVectorizer** và **TF-IDF**.

2.3.1. Phương pháp CountVectorizer

CountVectorizer của Scikit-learn được sử dụng để chuyển đổi văn bản thành ma trận số nguyên, trong đó mỗi cột biểu diễn một từ trong từ điển (vocabulary), và giá trị trong ô thể hiện số lần từ đó xuất hiện trong văn bản.. Nó cũng cung cấp khả năng xử

lý trước dữ liệu văn bản trước khi tạo biểu diễn vectơ, khiến nó trở thành một mô-đun biểu diễn tính năng cực kỳ linh hoạt cho văn bản. Đây là cách tiếp cận đơn giản, dựa trên mô hình Bag of Words (BoW).

Ví dụ: Tập văn bản gồm 3 câu:

1. “Sản phẩm tốt, chất lượng tốt”

2. “Sản phẩm kém, không hài lòng”

3. “Chất lượng tuyệt vời”

Từ điển (vocabulary) sau khi tách từ: {“sản”, “phẩm”, “tốt”, “chất”, “lượng”, “kém”, “không”, “hài”, “lòng”, “tuyệt”, “vời”}

Ma trận CountVectorizer:

Văn bản	sản	phẩm	tốt	chất	lượng	kém	không
1	1	1	2	1	1	0	0
2	1	1	0	0	0	1	1
3	0	0	0	1	1	0	0

Hình 2.3.1.1. Bảng ví dụ ma trận CountVectorizer

Ưu điểm:

- Đơn giản, dễ hiểu, dễ triển khai
- Hiệu quả trong các bài toán nhỏ, dữ liệu không quá phức tạp

Nhược điểm:

- Không phản ánh tầm quan trọng của từ (từ xuất hiện nhiêu như “sản phẩm” có thể chiếm ưu thế).
- Không thể hiện ngữ cảnh hay mối quan hệ giữa các từ.
- Ma trận thưa, gây tốn bộ nhớ nếu dữ liệu lớn.

2.3.2. Phương pháp TF-IDF (Term Frequency – Inverse Document Frequency)

TF-IDF là trọng số của một từ trong văn bản thu được qua thống kê thể hiện mức độ quan trọng của từ này trong một văn bản, mà bản thân văn bản đang xét nằm trong một

tập hợp các văn bản. Thuật toán này thường được sử dụng vì: trong ngôn ngữ luôn có những từ xảy ra thường xuyên với các từ khác.

Tần suất xuất hiện từ (TF): từ nào xuất hiện nhiều thì quan trọng hơn.

IDF: nhưng nếu từ xuất hiện ở mọi văn bản thì nó ít quan trọng (vd: "sản phẩm", "mua").

Kết hợp TF và IDF → mô hình tập trung vào những từ đặc trưng, phân biệt nhãn.
Công thức:

$$TF(t,d) = \frac{\text{số lần xuất hiện của từ } t \text{ trong văn bản } d}{\text{tổng số từ trong văn bản } d}$$

$$IDF(t) = \log \frac{N}{1 + n_t}$$

Trong đó:

- N: tổng số văn bản
- n_t : số văn bản chứ từ t

Ví dụ:

1. “Sản phẩm tốt, chất lượng tốt”

2. “Sản phẩm kém, không hài lòng”

3. “Chất lượng tuyệt vời”

Từ “sản” xuất hiện trong 2/3 văn bản -> IDF thấp.

Từ “tuyệt” xuất hiện duy nhất 1 lần -> IDF cao.

=> Kết quả: TF-IDF gán trọng số cao cho “tuyệt”, “vời”, vì chúng giúp phân biệt văn bản số 3 với các văn bản còn lại.

Ưu điểm:

- Phản ánh được tầm quan trọng của từ trong tập dữ liệu.
- Giảm ảnh hưởng của các từ phổ biến (như “sản phẩm”, “chất lượng”).

Nhược điểm:

- Vẫn không thể hiện mối quan hệ ngữ nghĩa giữa các từ.

- Có thể kém hiệu quả khi dữ liệu có nhiều từ đồng nghĩa hoặc ngữ cảnh phức tạp.

2.4. Mô hình học máy

2.4.1. Navie Bayes

Naive Bayes là một thuật toán phân loại xác suất đơn giản nhưng hiệu quả, dựa trên định lý Bayes và giả định "ngây thơ" rằng các thuộc tính (features) là độc lập có điều kiện với nhau khi biết lớp (class). Mặc dù giả định này không phải lúc nào cũng đúng trong thực tế, Naive Bayes vẫn có thể cho kết quả đáng ngạc nhiên trên nhiều bài toán phân loại. Cơ sở lý thuyết của Naive Bayes dựa vào định lý Bayes, cho phép tính xác suất hậu nghiệm của lớp C khi biết thuộc tính X, thông qua các xác suất tiên nghiệm và khả năng của dữ liệu quan sát được (Murphy, 2006). Công thức định lý Bayes được thể hiện dưới dạng:

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

Trong đó:

- C : nhãn (ví dụ: Hài lòng / Không hài lòng)
- X: dữ liệu đầu vào (chuỗi từ trong văn bản)
- P(C|X) : xác suất hậu nghiệm của lớp C khi biết thuộc tính X
- P(C): xác suất tiên nghiệm của lớp C
- P(X): xác suất của thuộc tính X

Ví dụ: Nếu từ “tốt” thường xuất hiện trong các văn bản nhãn Hài lòng, khi gặp bình luận có từ “tốt”, xác suất gán nhãn Hài lòng sẽ cao.

Ưu điểm:

- Nhanh, dễ huấn luyện, phù hợp với dữ liệu lớn.
- Hiệu quả cao với bài toán phân loại văn bản.

Nhược điểm:

- Giả định độc lập giữa các đặc trưng thường không đúng.
- Không linh hoạt khi dữ liệu phức tạp.

2.4.2. Random Forest

Random Forest là một phương pháp học máy mạnh mẽ, chủ yếu được sử dụng trong các bài toán phân loại và hồi quy. Phương pháp này kết hợp nhiều cây quyết định được tạo ra ngẫu nhiên, giúp xây dựng một hệ thống phân loại đa dạng và chính xác. Cơ sở lý thuyết của Random Forest dựa trên ba thành phần chính. Đầu tiên, kỹ thuật bagging (Bootstrap Aggregating) được sử dụng để tạo ra các tập con dữ liệu huấn luyện cho mỗi cây, bằng cách lấy mẫu ngẫu nhiên có hoàn lại từ tập dữ liệu ban đầu. Điều này giúp giảm thiểu sự thay đổi giữa các cây và gia tăng tính đa dạng của mô hình. Thứ hai, lựa chọn thuộc tính ngẫu nhiên được áp dụng tại mỗi nút của cây quyết định. Thay vì sử dụng tất cả các thuộc tính để phân chia, một tập con ngẫu nhiên các thuộc tính được chọn, giúp ngăn ngừa việc một số thuộc tính chiếm ưu thế trong quá trình xây dựng cây và làm tăng sự đa dạng giữa các cây. Cuối cùng, các dự đoán từ các cây quyết định được kết hợp lại, thường bằng phương pháp bình chọn đa số đối với phân loại hoặc trung bình đối với hồi quy, để đưa ra dự đoán cuối cùng.
Một trong những ưu điểm lớn của Random Forest là độ chính xác cao nhờ việc kết hợp nhiều cây quyết định, giúp giảm thiểu sai số và cải thiện khả năng dự đoán. Phương pháp này cũng thể hiện sự mạnh mẽ với nhiễu, có khả năng xử lý tốt các dữ liệu ngoại lai và giảm thiểu tác động của chúng đối với mô hình. Thêm vào đó, Random Forest có khả năng xử lý hiệu quả dữ liệu lớn với nhiều thuộc tính, giúp mô hình có thể làm việc với các tập dữ liệu phức tạp. Một ưu điểm khác là khả năng cung cấp ước lượng tầm quan trọng của các thuộc tính, cho phép đánh giá mức độ ảnh hưởng của từng thuộc tính đối với kết quả dự đoán, từ đó giúp người sử dụng đưa ra những quyết định chính xác hơn trong việc chọn lựa các yếu tố quan trọng.
Tóm lại, Random Forest là một phương pháp học máy hiệu quả, nổi bật nhờ khả năng kết hợp nhiều cây quyết định ngẫu nhiên để tạo ra một mô hình mạnh mẽ và linh hoạt. Với độ chính xác cao, khả năng xử lý nhiễu và dữ liệu lớn, cùng với khả năng đánh giá tầm quan trọng của các thuộc tính, Random Forest đã trở thành một công cụ không thể thiếu trong việc phân tích và dự đoán dữ liệu phức tạp

2.4.3. Logistic Regression

Hồi quy logistic là một kỹ thuật thống kê được sử dụng rộng rãi trong phân tích dữ liệu nhị phân, nơi biến kết quả chỉ có hai trạng thái, thường được mã hóa là 0 và 1. Phương pháp này dựa trên việc mô hình hóa mối quan hệ giữa biến kết quả và một hoặc nhiều biến dự đoán thông qua hàm logistic, một hàm sigmoid giúp ánh xạ giá trị thực thành xác suất nằm trong khoảng từ 0 đến 1. Thay vì trực tiếp dự đoán xác suất, hồi quy logistic tập trung vào mô hình hóa logarit tự nhiên của tỷ lệ chênh lệch (odds ratio), tức tỷ lệ giữa xác suất xảy ra sự kiện và xác suất không xảy ra sự kiện. (LaValley, 2008).

Các tham số của mô hình được ước tính thông qua phương pháp khả năng tối đa (MLE), đảm bảo xác suất quan sát dữ liệu thực tế là cao nhất. Hệ số hồi quy không chỉ cho biết ảnh hưởng của mỗi biến dự đoán đối với kết quả mà còn có thể được diễn giải dưới dạng tỷ lệ chênh lệch, làm sáng tỏ mức độ thay đổi trong khả năng xảy ra sự kiện khi biến dự đoán thay đổi một đơn vị. Một ưu điểm nổi bật của hồi quy logistic là khả năng dự đoán chính xác xác suất xảy ra sự kiện, đồng thời kiểm soát được nhiều biến nhiễu, giúp cải thiện chất lượng phân tích.

Hơn nữa, phương pháp này không yêu cầu biến kết quả phải tuân theo phân phối chuẩn hay giả định quan hệ tuyến tính giữa các biến, điều này làm cho hồi quy logistic trở thành công cụ linh hoạt trong nhiều tình huống dữ liệu khác nhau. Với khả năng xử lý dữ liệu nhị phân một cách chính xác và dễ giải thích, hồi quy logistic đã khẳng định vai trò quan trọng của mình không chỉ trong lĩnh vực thống kê mà còn trong học máy và phân tích dữ liệu thực tế

2.4.4. Support Vector Machine (SVM)

SVM là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc đệ quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta

vẽ đồi thị dữ liệu là các điểm trong n chiều (ở đây n là số lượng các tính năng bạn có) với giá trị của mỗi tính năng sẽ là một phần liên kết. Sau đó chúng ta thực hiện tìm "đường bay" (*hyper-plane*) phân chia các lớp. Hyper-plane nó chỉ hiểu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt.

Ví dụ: Với dữ liệu phân loại Hài lòng và Không hài lòng, SVM sẽ tìm ra đường ranh giới tối ưu sao cho phân tách được hai nhóm.

Ưu điểm:

- Hoạt động tốt với dữ liệu có số chiều cao (như văn bản).
- Hiệu quả với dữ liệu phức tạp và không tuyến tính.

Nhược điểm:

- Tốn nhiều thời gian khi dữ liệu lớn.
- Khó giải quyết kết quả trực quan.

Chương 3. XÂY DỰNG MÔ HÌNH

3.1. Thu thập dữ liệu

Nguồn dữ liệu: Bình luận sản phẩm được thu thập từ sàn thương mại điện tử Tiki. Đây là dữ liệu phản ánh trực tiếp trải nghiệm của khách hàng.

Phương pháp: Sử dụng kỹ thuật web crawling thông qua API. Cụ thể, dùng thư viện requests trong Python để gửi các truy vấn đến APi của Tiki. Với tổng 26 danh mục và tất cả sản phẩm thuộc danh mục đó, nên tập dữ liệu ban đầu được thu thập là 55950 dữ liệu và được lưu dưới dạng file .csv để xử lý.

Đặc điểm dữ liệu:

- Văn bản ngắn, nhiều từ cảm thán, biểu tượng, ký tự đặc biệt.
- Có sự xuất hiện của các từ viết tắt, teencode.
- Số lượng nhãn không cân bằng.

Bảng Tiki_Comments.csv

- Gồm 6 thuộc tính:

+ title: Mức độ đánh giá của khách hàng

+ content: Nội dung đánh giá

+ thank_count: Lượt thích bình luận

+ customer_id: Mã khách hàng bình luận

+ rating: Sao đánh giá sản phẩm

+ customer_name: Tên khách hàng bình luận

id	title	content	thank_count	customer_id	rating	customer_name
20149845	Hài lòng	m tốt nhất cho người đọc.	0	30366571	4	Hải Đăng
20135451	Cực kì hài lòng	tiếp tục đọc cuốn thứ hai!	1	6726888	5	Ngô Huyền Trang
20094432	Cực kì hài lòng	trọng hơn là khó chia sẻ.	1	30314118	5	Lê Nguyễn Gia Hân
20045798	Cực kì hài lòng	ighe bảo dắk hơn nữa :)))	2	12851810	5	Trâm Lê
20114385	Hài lòng	n phẩm của mình gặp vde	0	30361640	4	Phương Anh
20088010	Hài lòng	lỗi chính tả hơi đáng tiếc.	0	8408778	4	Hồng Thắm
20122647	Cực kì hài lòng	ối. Và hơn hết, là bài học	0	30348459	5	Ngọc Ánh
20157398	Hài lòng	ih thì đọc tiếp xem kết ntn	0	1364437	4	Vũ Đức Hiếu
19952333	Bình thường	có trải nghiệm không tốt.	0	6707062	3	Nguyễn Phúc Cường
19598719	Bình thường	i vẻ hay giải trí thì fail, tbh	1	19594971	3	Dat Thai
20047487	Cực kì hài lòng	àn về độ nổi tiếng và hay.	0	1424478	5	Phạm Lâm Kim Ngân
20090974	Cực kì hài lòng	nhà. cảm ơn tiki rất nhiều	0	7794129	5	Đỗ Khánh Linh
20139841	Hài lòng	c sách bị cong 1 chút, ổn.	0	30309633	4	Điệp Tạ
19963208	Cực kì hài lòng	hấy TIKI đang freeship	0	25072819	5	Sunny Nguyễn
20107370	Cực kì hài lòng	kĩ, in ấn đẹp, bìa xinh xỉu	0	27646304	5	Hoàng Yến
20153707	Cực kì hài lòng	ió phải gọi là siêu hayyyy	0	27218281	5	mynameis Jocasta
20139315	Cực kì hài lòng	ng gói cũng ok, hơi móp tí	0	30025341	5	Nguyễn Quốc Định
20124003	Cực kì hài lòng	nh, còn nguyên màng bọc	0	642183	5	hait huy
20064642	Cực kì hài lòng	ầy xước gì hết, 10đ	0	28753348	5	Leona Nil

Hình 3.1.1 Bảng Tiki_Comment.csv

3.2. Tiền xử lý dữ liệu

3.2.1. Làm sạch dữ liệu

- Chuẩn hóa kiểu dữ liệu: Nếu text không phải là chuỗi hoặc là chuỗi rỗng -> trả về “”

- Đầu vào: None -> Output: “”
- Đầu ra: “Xin chào” -> “Xin chào”

- Chuyển toàn bộ về chữ thường: thống nhất dữ liệu

- Đầu vào: “SảN pHẩM NÀY rẤT tốt”
- Đầu ra: “sản phẩm này rất tốt”

- Rút gọn ký tự lặp: Biểu thức “re.sub(r'(.)\1{2,}', r'\1', text” loại bỏ ký tự lặp nhiều hơn 2 lần để xử lý từ được gõ kéo dài nhằm nhấn mạnh cảm xúc

- Đầu vào: “ngonnnnnn quáaaa”
- Đầu ra: “ngon quá”

- Loại bỏ biểu tượng cảm xúc: Biểu thức “emoji_pattern.sub(r'', text)” để xóa các ký tự emoji không cần thiết để phân tích.

- Loại bỏ ký tự đặc biệt: Biểu thức “re.sub(r'[^\w\s]', ' ', text)” thay thế ký tự không phải chữ số hoặc số bằng dấu cách.

Đầu vào: “Sản phẩm!!! quá_tốt$$^^”

Đầu ra: “sản phẩm quá_tốt”

- Chuẩn hóa viết tắt: Dùng từ điển “short_word_dict” để thay thế các từ viết tắt thành từ đầy đủ.

Đầu vào “sp này okela lắm”

(short_word_dict = {"sp": "sản_phẩm", "okela": "tốt"})

Đầu vào: "sản_phẩm này tốt lắm"

- Tách từ: sử dụng “word_tokenize(text) để phân tách câu thanh từng từ / tokens. Là bước quan trọng trong xử lý tiếng Việt vì từ có thể gồm nhiều tiếng.

Đầu vào: "sản phẩm này rất tốt"

Đầu ra: ["sản", "phẩm", "này", "rất", "tốt"]

- Ghép từ với tiền tố (prefix): Nếu một từ nằm trong prefix_words (phủ định, tăng cường, trạng thái) và ngay sau nó có một từ khác thì ghép lại thành 1 token giúp giữ nguyên ý nghĩa ngữ cảnh.

Đầu vào: ["không", "hài_lòng", "về", "sản_phẩm"]

Đầu ra: ["không_hài_lòng", "về", "sản_phẩm"]

- Kết hợp tokens thành câu hoàn chỉnh: Ghép các tokens lại thành chuỗi văn bản sạch.

Đầu vào: ["không_hài_lòng", "về", "sản_phẩm"]

Đầu ra: "không_hài_lòng về sản_phẩm"

```python
from underthesea import word_tokenize, text_normalize
import re

prefix_words = {
    "không", "chưa", "chẳng", "chả", "đâu", "khỏ",
    "chả_hề", "chẳng_hề", "không_hề", "chưa_hề",
    "chẳng_bao_giờ", "không_bao_giờ",

    "siêu", "cực", "rất", "quá", "khá", "hơi", "thật",
    "hết_sức", "cực_kỳ", "cực_kì", "vô_cùng",
    "khủng_khiếp", "cực_đỉnh", "siêu_cấp",
    "rất_chi", "quá_đỗi", "cực_thích",

    "bị", "mới", "cũ", "to", "nhỏ", "nặng", "nhẹ",
    "cao", "thấp", "ốn", "tạm", "gần", "xa",
    "già", "trẻ", "kém", "xịn", "dở",
    "xấu", "tốt", "chậm", "nhanh"
}


def clean_data(text):
    if not isinstance(text, str) or text.strip() == "":
        return ""
    text = text.lower()
    text = re.sub(r'(.)\1{2,}', r'\1', text)
    text = emoji_pattern.sub(r'', text)
    text = re.sub(r'[^\w\s]', ' ', text)
    text = text_normalize(text)

    words = text.split()
    words = [short_word_dict.get(w, w) for w in words]
    text = ' '.join(words)

    tokens = word_tokenize(text)

    merged_tokens = []
    skip = False
    for i, w in enumerate(tokens):
        if skip:
            skip = False
            continue
        if w in prefix_words and i+1 < len(tokens):
            merged_tokens.append(w + "_" + tokens[i+1])
            skip = True
        else:
            merged_tokens.append(w)

    return " ".join(merged_tokens)
```

Hình 3.2.1. Code làm sạch dữ liệu

3.2.2. Chuẩn hóa dữ liệu

Trong quá trình xử lý dữ liệu văn bản, ngoài việc xây dựng hàm clean_data() để chuẩn hóa nội dung, dữ liệu gốc còn được lưu trữ trong các cột của DataFrame. Để đảm bảo tính thống nhất và thuận tiện cho việc phân tích, nhóm tiến hành chuẩn hóa hai cột chính: content (nội dung đánh giá) và title (tiêu đề đánh giá). Cụ thể:

Bước 1: Chuẩn hóa cột content

Đối với cột content, toàn bộ văn bản được đưa vào hàm clean_data() để loại bỏ nhiễu và chuẩn hóa theo các bước đã trình bày ở mục trước (chuyển chữ thường, loại bỏ emoji, ký tự đặc biệt, chuẩn hóa viết tắt, tách từ, ghép tiền tố).
Kết quả sau khi xử lý được lưu vào một cột mới tên là clean_content, giúp vẫn giữ lại dữ liệu gốc để đối chiếu khi cần thiết.

Bước 2: Chuẩn hóa cột title

Tương tự như với cột content, toàn bộ dữ liệu trong cột title cũng được đưa vào hàm clean_data() để làm sạch. Tuy nhiên, nhằm tăng tính trực quan khi hiển thị tiêu đề, tiến hành thay thế ký tự gạch dưới (_) bằng dấu cách ( ).

Bước 3: Kiểm tra kết quả

Sau khi chuẩn hóa, tiến hành kiểm tra trực quan bằng cách hiển thị 10 dòng dữ liệu đầu tiên.

Kết quả thu được cho thấy dữ liệu đã được chuẩn hóa thành công. Cột clean_content giữ nguyên dấu gạch dưới để phục vụ cho phân tích học máy (giúp phân biệt ngữ cảnh như "không_tốt" ≠ "tốt"), trong khi cột title hiển thị dạng dễ đọc phục vụ trực quan.

Chuẩn hóa dữ liệu

```
df['clean_content'] = df['content'].map(lambda text: clean_data(text))

df['title'] = df['title'].map(lambda text: clean_data(text))
df['title'] = df['title'].str.replace('_', ' ')
df.head(10)
```

	title	content	rating	clean_content
0	hài lòng	Sách hay, nội dung khá cuốn\r\nKhi sách về ngu...	4	sách hay nội dung khá_cuốn khi sách về nguyên ...
1	cực kì hài lòng	Beartown: Khi thể thao không chỉ là thể thao\n...	5	beartown khi thể thao không_chỉ là thể thao cố...
2	cực kì hài lòng	Beartown như những miêu tả là nơi dị thường. X...	5	beartown như những miêu tả là nơi dị thường xu...
3	cực kì hài lòng	Mình đã nghĩ quyển này sẽ không quá lôi cuốn, ...	5	mình đã nghĩ quyển này sẽ không_quá lôi cuốn n...
4	hài lòng	Hàng đẹp giá ổn\r\nGiao nhanh\r\nTiki hỗ trợ n...	4	hàng đẹp giá ổn_giao nhanh_tiki hỗ trợ nhiệt t...
5	hài lòng	Nội dung sách hay, lời văn lôi cuốn, thích nhấ...	4	nội dung sách hay lời văn lôi cuốn thích nhất ...
6	cực kì hài lòng	Cuốn sách sẽ mang lại cho bạn biết rất nhiều đ...	5	cuốn sách sẽ mang lại cho bạn biết rất_nhiều đ...
7	hài lòng	Cuốn này tạm ổn. Ko đọc được đến cuối do đoạn ...	4	cuốn này tạm_ổn không_đọc được đến cuối do đoạ...
8	bình thường	Sách có vết mực ở nhiều trang, một số trang sá...	3	sách có vết mực ở nhiều trang một số trang sác...
9	bình thường	Mình đã hơi ẩu khi mua cuốn này với suy nghĩ/t...	3	mình đã hơi_ẩu khi mua cuốn này với suy nghĩ t...

Hình 3.2.2. Kết quả chuẩn hóa dữ liệu trong DataFrame

3.2.3. Phân tích dữ liệu

Sau khi thu thập, làm sạch và chuẩn hóa dữ liệu. Tiến hành phân tích sơ bộ để hiểu rõ hơn về đặc điểm nội dung các bình luận của khách hàng.

1. Mức độ tập trung của các từ.

Để phân tích sơ bộ dữ liệu văn bản sau khi chuẩn hóa, tiến hành xây dựng **Bag-of-Words** và trực quan hóa bằng **Word Cloud**.

- **Tạo Bag-of-Words**: Tất cả các từ trong cột *clean_content* được gom lại thành một tập hợp (corpus). Sau đó, nhóm sử dụng nltk.FreqDist() để thống kê tần suất xuất hiện của từng từ.

- **Thống kê cơ bản**: Xuất ra tổng số từ trong corpus và 15 từ xuất hiện nhiều nhất, nhằm nhận diện sơ bộ các từ khóa thường gặp trong dữ liệu.

-**Trực quan hóa bằng Word Cloud**: Dùng thư viện WordCloud để biểu diễn trực quan. Các từ có tần suất xuất hiện cao sẽ được hiển thị với kích thước lớn hơn. Hình ảnh được thiết lập với nền trắng, kích thước 2000x1000, và bảng màu nổi bật (colormap="tab10").

Hình 3.2.3.1. Word Cloud thể hiện tần suất xuất hiện từ trong tập dữ liệu

2. Phân phối độ dài câu trong các đánh giá của khách hàng.

Để hiểu rõ hơn về đặc điểm dữ liệu văn bản, tiến hành thống kê **độ dài câu** (tính theo số lượng từ) trong các đánh giá của khách hàng.

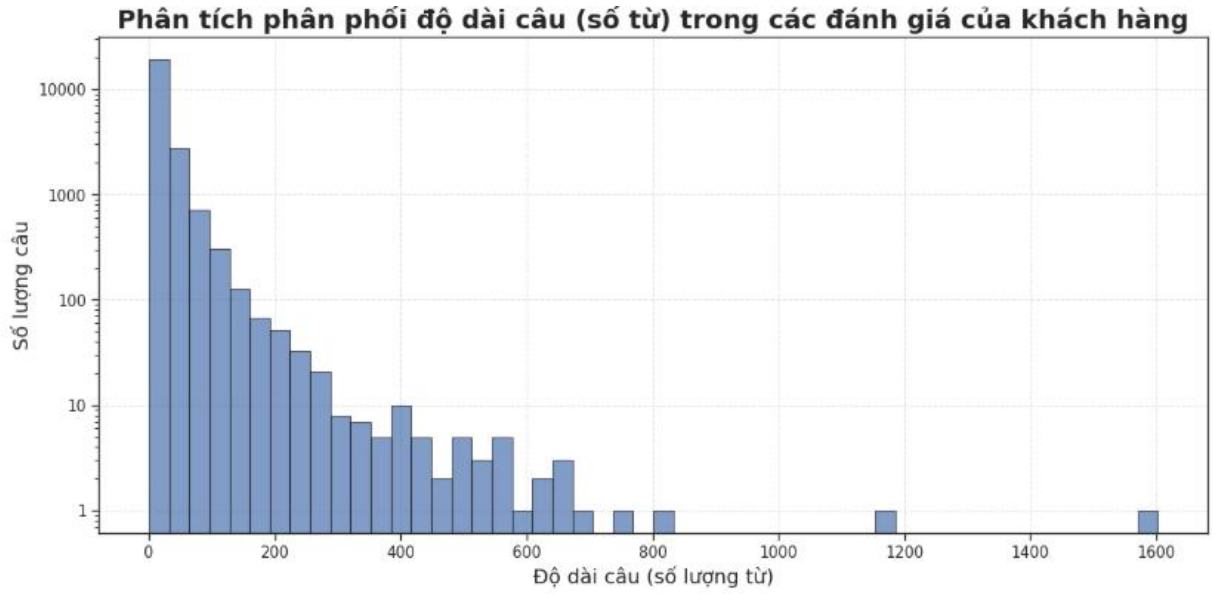
- Tính toán độ dài câu: Với mỗi văn bản trong cột *clean_content*, độ dài được xác định bằng số lượng từ (sử dụng phép tách bằng khoảng trắng).

- Trực quan hóa bằng biểu đồ histogram: Dữ liệu được hiển thị dưới dạng biểu đồ tần suất, với:

+ Trục hoành: số từ trong câu (độ dài câu).

+ Trục tung: số lượng câu có độ dài tương ứng.

+Thang đo trục tung được chuyển sang **logarit** nhằm giúp biểu đồ cân đối hơn, dễ quan sát cả những giá trị lớn và nhỏ.



Hình 3.2.3.2. Phân phối độ dài câu trong tập dữ liệu đánh giá khách hàng

3. Sự phân bố giữa label và rating

Để có cái nhìn tổng quan về dữ liệu đánh giá, tiến hành trực quan hóa theo hai khía cạnh: (i) nhãn đánh giá phổ biến và (ii) phân phối điểm số rating.

- Thống kê nhãn đánh giá:

+ Sử dụng value_counts() để đếm tần suất xuất hiện của từng nhãn (title).

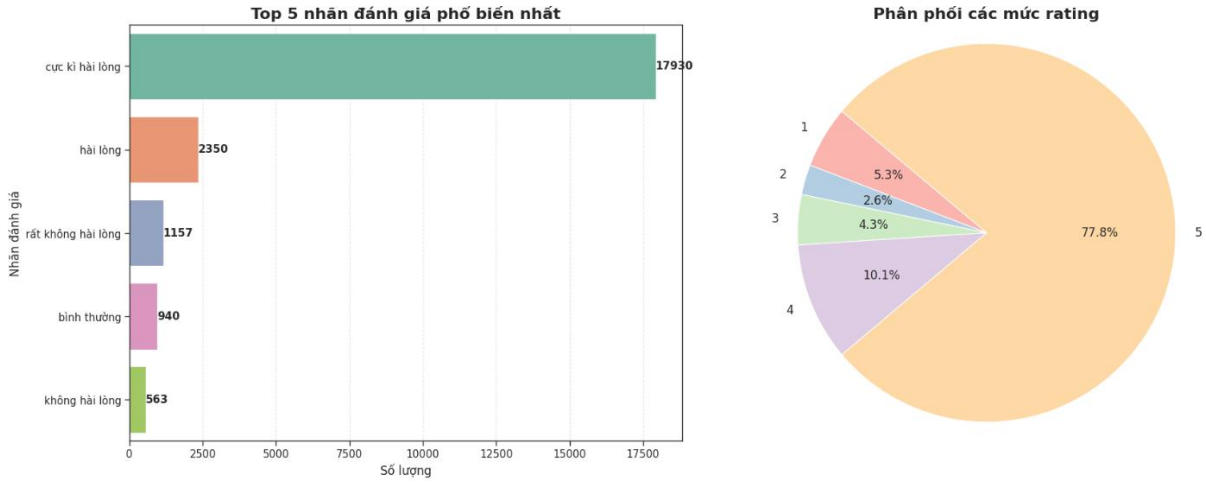
+ Lấy Top 5 nhãn xuất hiện nhiều nhất để trực quan bằng biểu đồ cột ngang (barplot).

Đồng thời, annotate (hiển thị số lượng cụ thể) ngay tại cuối mỗi cột để dễ quan sát.

- Thống kê điểm số rating:

+ Tính tần suất các mức rating (từ 1 đến 5 sao).

+ Trực quan bằng biểu đồ tròn (pie chart) để biểu diễn tỷ lệ phần trăm của từng mức điểm.



Hình 3.2.3.3. Biểu đồ thể hiện Top 5 nhãn đánh giá phổ biến và phân phối các mức rating trong tập dữ liệu

3.2.4. Cân bằng dữ liệu

Tiến hành gôm nhóm từ 5 nhãn tương ứng với rating thành 2 nhãn chính là nhãn "cực kỳ hài lòng" và "không hài lòng" .

Hình 3.2.4.1. Word Cloud thể hiện tần suất xuất hiện từ trong 2 nhãn chính

Vì số lượng tập dữ liệu có nhãn "cực kỳ hài lòng" chiếm số lượng lớn hơn nhãn "không hài lòng" (20547 > 2838), dẫn đến hiện tượng lệch mẫu, gây mất cân bằng ảnh hưởng đến kết quả mô hình. Thực hiện cân bằng mẫu bắt đầu từ việc triển khai kỹ thuật tăng cường văn bản (EDA - Easy Data Augmentation) cho nhãn "không hài lòng" bằng 4 phương pháp cơ bản:

+ Synonym Replacement (Thay thế từ đồng nghĩa)

+ Random Insertion (Chèn từ ngẫu nhiên)

+ Random swap (Hoán đổi từ ngẫu nhiên)

+ Random Deletion (Xóa từ ngẫu nhiên)

Phương pháp trên giúp tạo thêm dữ liệu mới từ tập gốc mà không làm thay đổi nhiều ý nghĩa nội dung. Vì vậy, kỹ thuật này cực kỳ hữu ích khi tập dữ liệu ban đầu nhỏ hoặc mất cân bằng giữ liệu các nhãn. Nhờ đó, mô hình học máy có thể giảm overfitting và cải thiện khả năng khái quát hóa.

```
Phân phối ban đầu:
title
cực kỳ hài lòng    20547
không hài lòng      2838
Name: count, dtype: int64

Phân phối sau khi EDA:
title
cực kỳ hài lòng    20547
không hài lòng     14190
Name: count, dtype: int64
```

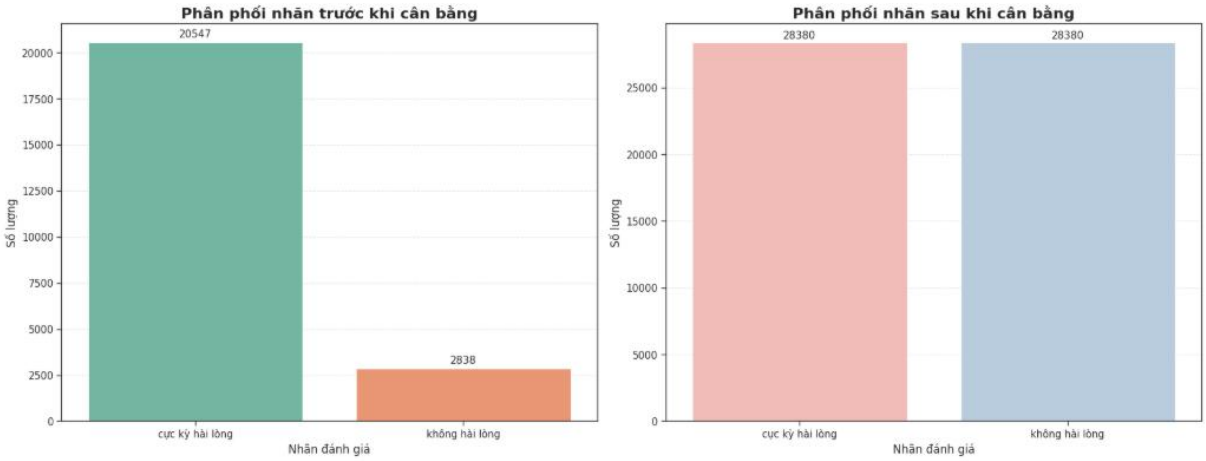
Cân bằng dữ liệu bằng phương pháp Undersampling

```
Phân phối cuối cùng cân bằng theo nhãn 'không hài lòng':
title
không hài lòng     14190
cực kỳ hài lòng    14190
Name: count, dtype: int64
```

Để tăng cường dữ liệu huấn luyện, tạo thêm phiên bản văn bản không dấu từ dữ liệu gốc. Phương pháp này giúp mô hình học được cả hai dạng dữ liệu (có dấu và không dấu), đồng thời phản ánh thực tế khi người dùng có thể nhập đánh giá không dấu. Kết quả là tập dữ liệu được nhân đôi, cải thiện độ đa dạng và khả năng tổng quát của mô hình.

```
Phân phối sau khi thêm dữ liệu không dấu:
title
cực kỳ hài lòng    28380
không hài lòng     28380
Name: count, dtype: int64
```

Kết quả thực hiện:



Hình 3.2.4.2. Biểu đồ cột trước và sau khi nhãn được cân bằng

3.3. Chia dữ liệu để huấn luyện và kiểm tra (80 - 20)

Tập dữ liệu sau khi xử lý gồm 56.760 mẫu, cân bằng giữa hai nhãn: cực kỳ hài lòng (28380) và không hài lòng (28380)

Dữ liệu chia theo tỷ lệ 80/20 (train/test) với phương pháp stratified sampling để đảm bảo cân bằng nhãn:

```
Tổng số mẫu 2 nhãn: 56760
Phân phối nhãn trước khi chia:
title
cực kỳ hài lòng    28380
không hài lòng     28380
Name: count, dtype: int64

Số mẫu tập train: 45408
Số mẫu tập test: 11352

Phân phối nhãn trong tập train:
title
không hài lòng     22704
cực kỳ hài lòng    22704
Name: count, dtype: int64

Phân phối nhãn trong tập test:
title
cực kỳ hài lòng    5676
không hài lòng     5676
Name: count, dtype: int64
```

Lý do chọn tỷ lệ 80/20 vì đây là tỷ lệ phổ biến trong huấn luyện mô hình học máy. Với 80% dữ liệu dùng để huấn luyện, mô hình có đủ thông tin để học đặc trưng của dữ liệu. 20% còn lại được giữ lại cho quá trình kiểm tra , giúp đánh giá khách quan khả năng tổng quát của mô hình đối với dữ liệu chưa từng thấy.

3.4. Vector hóa dữ liệu

1. Biểu diễn dữ liệu văn bản bằng Bag-of-Words

Sau khi hoàn tất các bước tiền xử lý và chia dữ liệu thành tập huấn luyện và tập kiểm tra, văn bản cần được chuyển đổi sang dạng số để mô hình học máy có thể xử lý. Trong đề tài này, phương pháp Bag-of-Words (CountVectorizer) được lựa chọn để trích xuất đặc trưng từ văn bản.

Quy trình thực hiện:

1. Xử lý dữ liệu rỗng (NaN): Trước khi vector hóa, dữ liệu văn bản ở cả tập train và test được kiểm tra và loại bỏ các giá trị rỗng nhằm đảm bảo tính toàn vẹn. Đồng thời, nhãn (label) cũng được lọc tương ứng để không bị sai lệch index giữa dữ liệu và nhãn.

2. Khởi tạo CountVectorizer: Đây là công cụ trong thư viện scikit-learn, thực hiện biến đổi văn bản thành ma trận đặc trưng. Mỗi cột của ma trận ứng với một từ trong từ vựng (được học từ tập train), và mỗi hàng tương ứng với một văn bản trong tập dữ liệu.

3. Vector hóa dữ liệu:

- Tập huấn luyện (train_sentences) được fit để xây dựng từ vựng và transform thành ma trận số.
- Tập kiểm tra (test_sentences) chỉ thực hiện transform dựa trên từ vựng đã học từ tập train, đảm bảo tính đồng nhất.

Kết quả: Ma trận đặc trưng thu được có kích thước

- X_train shape: (45376, 13241)
- X_test shape: (11340, 13241)

2. Biểu diễn dữ liệu văn bản bằng TF_IDF

Bên cạnh Bag-of-Words, nghiên cứu này sử dụng thêm phương pháp TF-IDF (Term Frequency – Inverse Document Frequency) để trích xuất đặc trưng từ văn bản. TF-IDF không chỉ tính số lần xuất hiện của từ, mà còn cân nhắc mức độ quan trọng của từ trong toàn bộ tập dữ liệu.

Quy trình thực hiện:

1. Khởi tạo TF-IDF Vectorizer**:** Sử dụng TfidfVectorizer trong thư viện scikit-learn.

2. Huấn luyện và biến đổi dữ liệu::

- Tập train được fit để học từ vựng và đồng thời transform thành ma trận TF-IDF.
- Tập test chỉ thực hiện transform dựa trên từ vựng đã học từ tập train. Điều này giúp đảm bảo tính nhất quán khi đánh giá mô hình.

Kết quả: Ma trận đặc trưng thu được có kích thước

- X_train shape: (45376, 13241)
- X_test shape: (11340, 13241)

```python
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
import pandas as pd

# Tập câu mẫu
sentences = [
    "Sản phẩm này rất tốt",
    "Sản phẩm tốt",
    "Dịch vụ chưa hài lòng",
    "Sản phẩm rất tốt và dịch vụ tốt"
]

# --- CountVectorizer ---
count_vect = CountVectorizer()
X_count = count_vect.fit_transform(sentences)
df_count = pd.DataFrame(X_count.toarray(), columns=count_vect.get_feature_names_out())
print("=== CountVectorizer ===")
print(df_count)

# --- TfidfVectorizer ---
tfidf_vect = TfidfVectorizer()
X_tfidf = tfidf_vect.fit_transform(sentences)
df_tfidf = pd.DataFrame(X_tfidf.toarray(), columns=tfidf_vect.get_feature_names_out())
print("\n=== TfidfVectorizer ===")
print(df_tfidf)
```

Hình 3.4.1 Minh họa kết quả biến đổi văn bản bằng CountVectorizer và TfidfVectorizer

```
=== CountVectorizer ===
   chưa  dịch  hài  lòng  này  phẩm  rất  sản  tốt  và  vụ
0     0     0    0     0    1     1    1    1    1   0   0
1     0     0    0     0    0     1    0    1    1   0   0
2     1     1    1     1    0     0    0    0    0   0   1
3     0     1    0     0    0     1    1    1    2   1   1

=== TfidfVectorizer ===
       chưa      dịch       hài      lòng       này      phẩm       rất
0  0.000000  0.000000  0.000000  0.000000  0.592992  0.378499  0.467522
1  0.000000  0.000000  0.000000  0.000000  0.000000  0.577350  0.000000
2  0.485461  0.382743  0.485461  0.485461  0.000000  0.000000  0.000000
3  0.000000  0.342166  0.000000  0.000000  0.000000  0.277013  0.342166

        sản       tốt        và        vụ
0  0.378499  0.378499  0.000000  0.000000
1  0.577350  0.577350  0.000000  0.000000
2  0.000000  0.000000  0.000000  0.382743
3  0.277013  0.554026  0.433994  0.342166
```

Hình 3.4.2 Kết quả chuyển đổi văn bản

Minh họa kết quả chuyển đổi văn bản sang đặc trưng số bằng hai phương pháp phổ biến là CountVectorizer và TfidfVectorizer. Ở phần trên, CountVectorizer thể hiện số lần xuất hiện của từng từ trong mỗi câu. Ví dụ, từ “tốt” xuất hiện một lần trong câu thứ

nhất, một lần trong câu thứ hai và hai lần trong câu thứ tư. Trong khi đó, ở phần dưới, TfidfVectorizer biểu diễn trọng số TF-IDF, phản ánh cả tần suất xuất hiện và mức độ quan trọng của từ trong toàn bộ tập dữ liệu. Chẳng hạn, từ "sản" có trọng số TF-IDF cao hơn ở câu thứ hai (0.577350) so với câu thứ nhất (0.378499). Sự khác biệt này cho thấy CountVectorizer chỉ đơn thuần dựa trên số lần lặp lại, còn TfidfVectorizer giúp giảm ảnh hưởng của các từ phổ biến, đồng thời nhấn mạnh những từ có giá trị phân loại cao. Vì vậy, trong thực tiễn, TF-IDF thường được sử dụng nhiều hơn để trích chọn đặc trưng trong các bài toán phân loại văn bản.

3.5. Huấn luyện và đánh giá mô hình

Trong nghiên cứu này,xây dựng và huấn luyện các mô hình học máy khác nhau nhằm giải quyết bài toán phân loại cảm xúc văn bản với **2 nhãn** (hài lòng và không hài lòng). Việc so sánh nhiều mô hình giúp đánh giá mức độ phù hợp và hiệu quả của từng thuật toán, từ đó lựa chọn mô hình tối ưu cho ứng dụng thực tế.

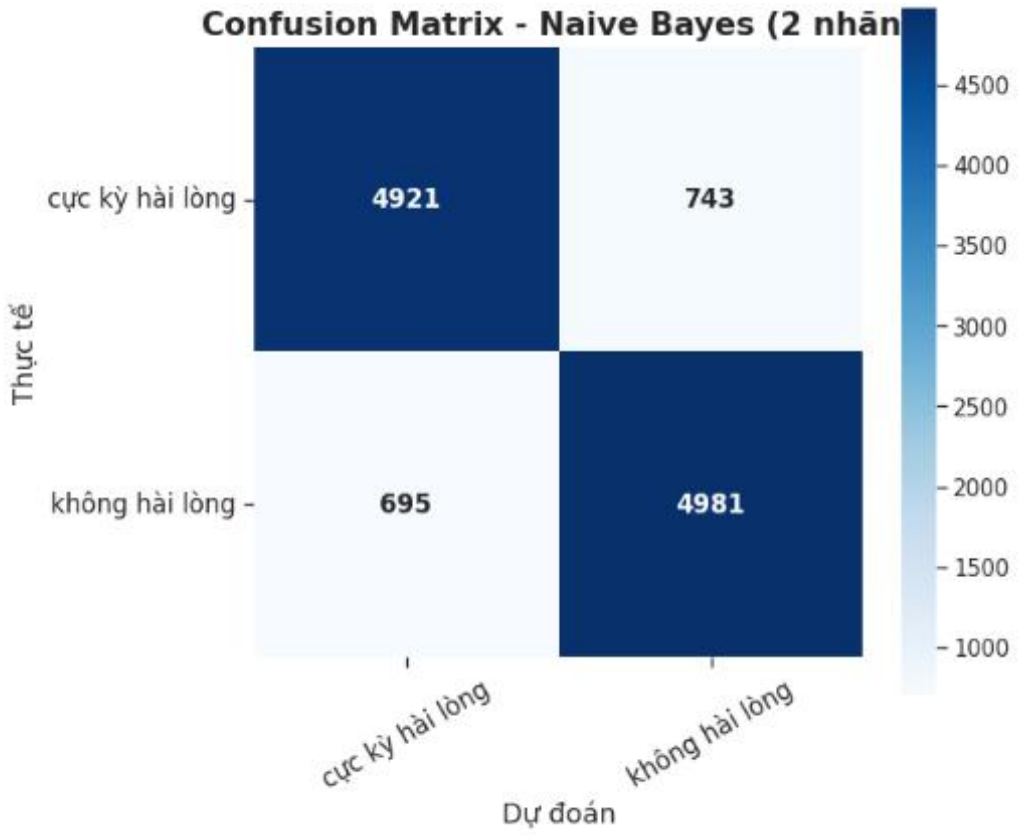
3.5.1. Navie Bayes

```
===== NAIVE BAYES (2 nhãn) =====
Accuracy: 0.8731922398589065
Classification Report:
                precision    recall  f1-score   support

cực kỳ hài lòng       0.88      0.87      0.87      5664
 không hài lòng       0.87      0.88      0.87      5676

       accuracy                           0.87     11340
      macro avg       0.87      0.87      0.87     11340
   weighted avg       0.87      0.87      0.87     11340
```

Hình 3.5.1.1: Kết quả đánh giá mô hình Naive Bayes với báo cáo phân loại



Hình 3.5.1.2: Ma trận nhầm lẫn của mô hình Naive Bayes trên tập dữ liệu 2 nhãn

Trong quá trình huấn luyện và đánh giá, mô hình Naive Bayes cho kết quả với độ chính xác đạt 87,3%, đồng thời các chỉ số Precision, Recall và F1-score đều ổn định quanh mức 0,87–0,88. Điều này cho thấy mô hình duy trì được sự cân bằng giữa hai nhãn "cực kỳ hài lòng" và "không hài lòng", không thiên lệch đáng kể về phía nào. Tuy nhiên, ma trận nhầm lẫn chỉ ra rằng vẫn có khoảng 700 mẫu ở mỗi lớp bị dự đoán sai, phản ánh hạn chế của Naive Bayes trong việc xử lý ngữ nghĩa phức tạp và các trường hợp có sự chồng lấn từ vựng. Nhìn chung, Naive Bayes phù hợp đóng vai trò mô hình cơ sở (baseline) cho bài toán phân loại cảm xúc, nhưng để nâng cao hiệu quả, cần xem xét áp dụng các mô hình mạnh hơn như Logistic Regression, SVM hoặc Random Forest nhằm cải thiện độ chính xác và giảm thiểu lỗi phân loại.

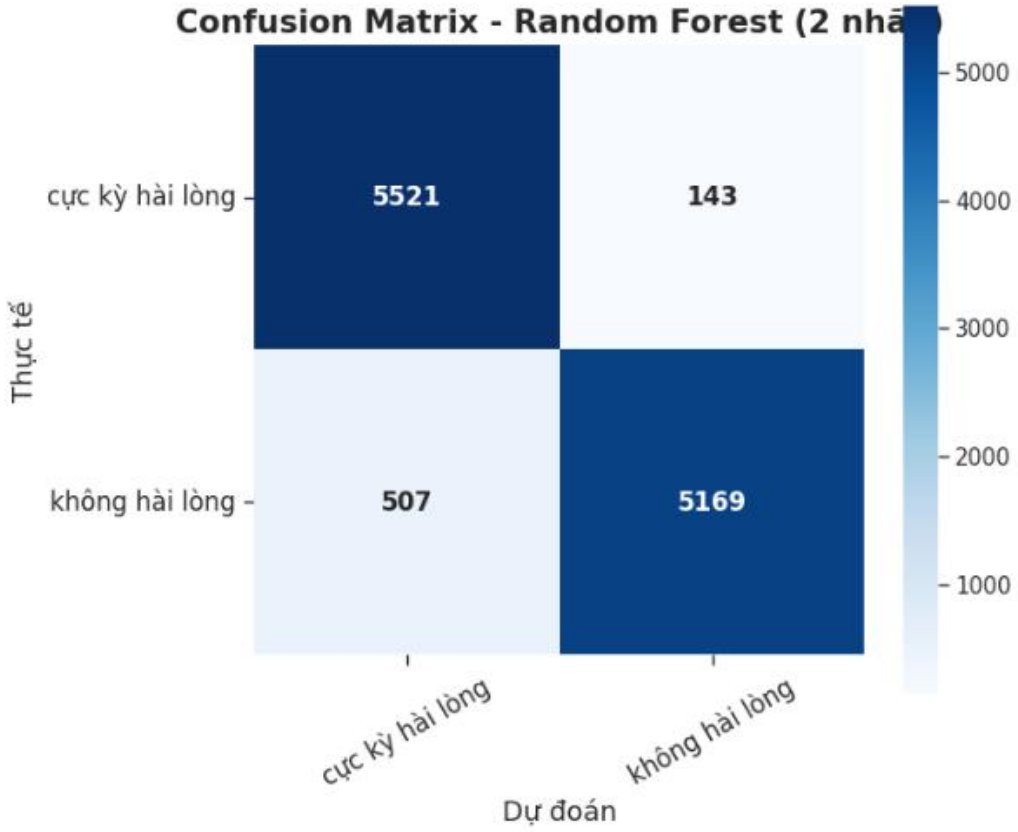
3.5.2. Random Forest

```
===== RANDOM FOREST (2 nhãn) =====
Accuracy: 0.9426807760141094
Classification Report:
                    precision    recall  f1-score   support

cực kỳ hài lòng         0.92      0.97      0.94      5664
 không hài lòng         0.97      0.91      0.94      5676

          accuracy                           0.94     11340
         macro avg       0.94      0.94      0.94     11340
      weighted avg       0.94      0.94      0.94     11340
```

Hình 3.5.2.1: Kết quả đánh giá mô hình Random Forest với báo cáo phân loại



Hình 3.5.2.2: Ma trận nhầm lẫn của mô hình Random Forest trên tập dữ liệu 2 nhãn

Kết quả mô hình Random Forest (2 nhãn) cho thấy độ chính xác đạt 94.27%, cao hơn đáng kể so với Naive Bayes. Ở lớp "cực kỳ hài lòng", mô hình có precision = 0.92 và recall = 0.97, điều này chứng tỏ khả năng nhận diện tốt và ít bỏ sót các trường hợp thuộc lớp này. Trong khi đó, ở lớp "không hài lòng", precision đạt 0.97 nhưng recall chỉ đạt 0.91, tức là vẫn còn một số trường hợp "không hài lòng" bị nhầm sang "cực kỳ

hài lòng". Giá trị F1-score của cả hai lớp đều đạt 0.94, cho thấy sự cân bằng tốt giữa precision và recall. Quan sát ma trận nhầm lẫn, mô hình phân loại đúng 10,690 mẫu trên tổng số 11,340 mẫu, trong đó chỉ có 143 trường hợp "cực kỳ hài lòng" bị dự đoán nhầm thành "không hài lòng" và 507 trường hợp "không hài lòng" bị dự đoán nhầm thành "cực kỳ hài lòng". Như vậy, mô hình Random Forest thể hiện hiệu quả vượt trội hơn Naive Bayes trong bài toán phân loại cảm xúc 2 nhãn, đặc biệt mạnh trong việc nhận diện các phản hồi "cực kỳ hài lòng".

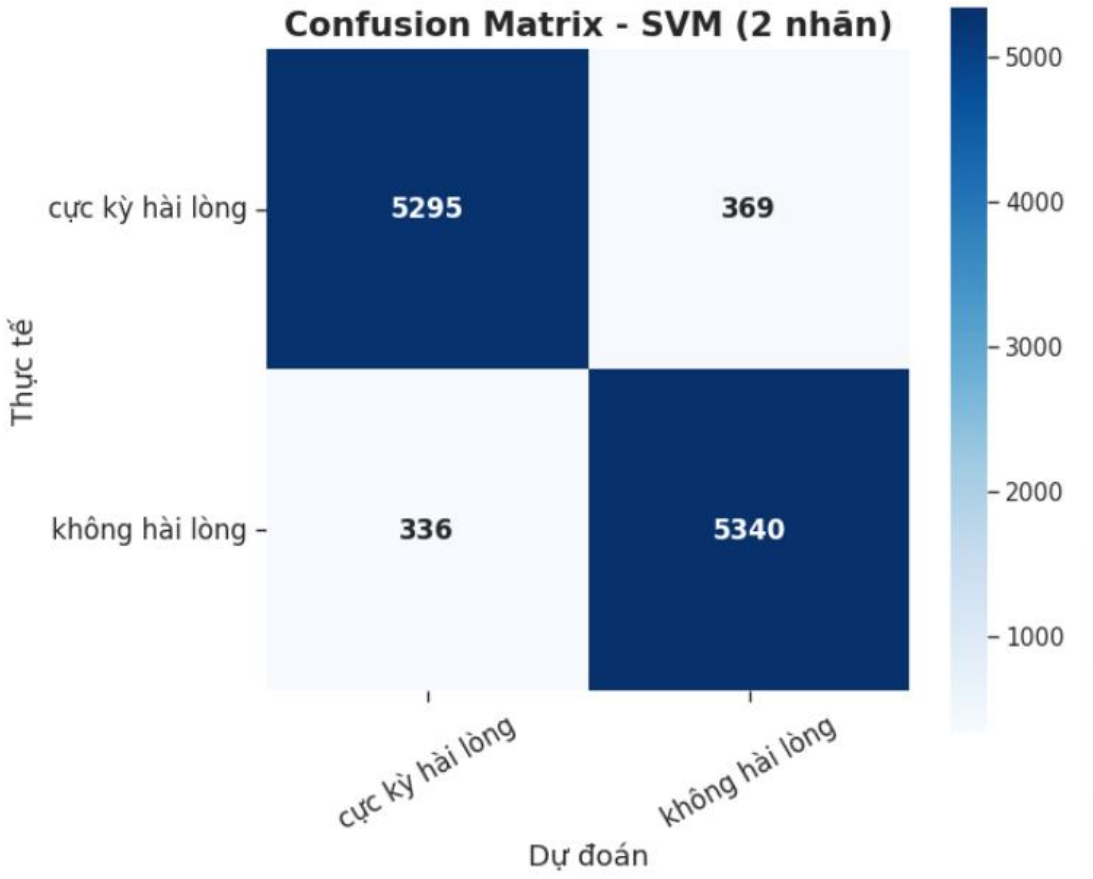
3.5.3. SVM

```
 ===== SVM (2 nhãn) =====
Accuracy: 0.9378306878306878
Classification Report:
                precision    recall  f1-score   support

cực kỳ hài lòng       0.94      0.93      0.94      5664
 không hài lòng       0.94      0.94      0.94      5676

       accuracy                           0.94     11340
      macro avg       0.94      0.94      0.94     11340
   weighted avg       0.94      0.94      0.94     11340
```

Hình 3.5.3.1: Kết quả đánh giá mô hình SVM với báo cáo phân loại



Hình 3.5.3.2: Ma trận nhầm lẫn của mô hình SVM trên tập dữ liệu 2 nhãn

Kết quả mô hình SVM (2 nhãn) đạt độ chính xác 93.78%, cho thấy hiệu suất khá cao và gần tương đương với Random Forest. Ở lớp "cực kỳ hài lòng", precision = 0.94 và recall = 0.93, chứng tỏ mô hình dự đoán đúng hầu hết các phản hồi thuộc lớp này nhưng vẫn có một số bị nhầm sang lớp còn lại. Với lớp "không hài lòng", precision và recall đều ở mức 0.94, thể hiện sự cân bằng tốt giữa khả năng phân loại chính xác và khả năng bao quát dữ liệu. Điểm F1-score của cả hai lớp đều đạt 0.94, phản ánh sự ổn định và cân bằng trong hiệu quả mô hình. Quan sát ma trận nhầm lẫn, mô hình phân loại đúng 10,635 mẫu trên tổng số 11,340 mẫu, trong đó có 369 trường hợp "cực kỳ hài lòng" bị nhầm thành "không hài lòng" và 336 trường hợp "không hài lòng" bị nhầm thành "cực kỳ hài lòng". Nhìn chung, mô hình SVM đạt hiệu quả cao, cho kết quả đồng đều giữa hai lớp và ít thiên lệch, là một lựa chọn mạnh mẽ cho bài toán phân loại cảm xúc với 2 nhãn.

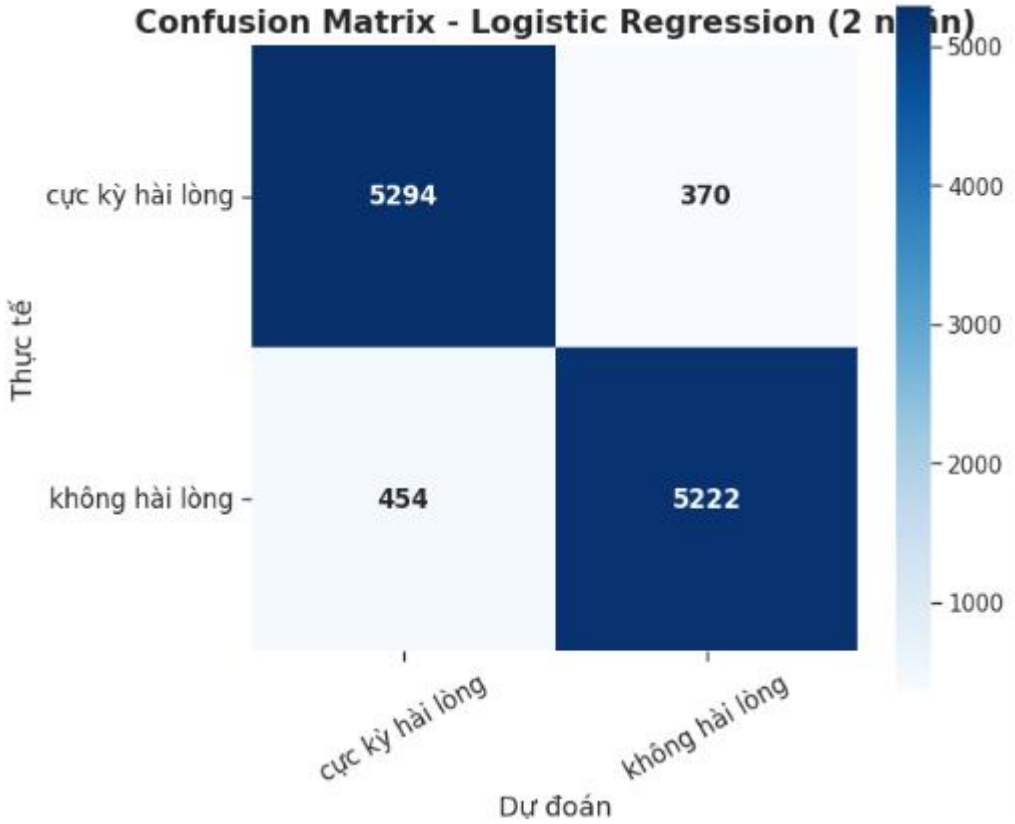
3.5.4. Logistic Regression

```
 ===== LOGISTIC REGRESSION (2 nhãn) =====
Accuracy: 0.9273368606701940
Classification Report:
                  precision    recall  f1-score   support

cực kỳ hài lòng       0.92      0.93      0.93      5664
 không hài lòng       0.93      0.92      0.93      5676

       accuracy                           0.93     11340
      macro avg       0.93      0.93      0.93     11340
   weighted avg       0.93      0.93      0.93     11340
```

Hình 3.5.4.1: Kết quả đánh giá mô hình Logistic Regression với báo cáo phân loại



Hình 3.5.4.2: Ma trận nhầm lẫn của mô hình Logistic Regression trên tập dữ liệu 2 nhãn

Kết quả mô hình Logistic Regression (2 nhãn) đạt độ chính xác 92.73%, phản ánh hiệu suất phân loại tốt và khá ổn định. Với lớp "cực kỳ hài lòng", precision = 0.92 và recall = 0.93, cho thấy mô hình nhận diện khá chính xác và ít bỏ sót dữ liệu. Ở lớp "không

hài lòng", precision = 0.93 và recall = 0.92, chứng tỏ khả năng phân loại cân bằng, không bị thiên lệch nhiều giữa hai lớp. Điểm F1-score của cả hai lớp đều đạt 0.93, thể hiện sự đồng đều và cân đối giữa precision và recall. Quan sát ma trận nhầm lẫn, mô hình dự đoán chính xác 10,516 mẫu trên tổng số 11,340 mẫu, trong đó có 370 phản hồi "cực kỳ hài lòng" bị nhầm thành "không hài lòng" và 454 phản hồi "không hài lòng" bị nhầm thành "cực kỳ hài lòng". Nhìn chung, Logistic Regression mang lại hiệu quả cao, mô hình khá đơn giản nhưng vẫn cho kết quả đáng tin cậy, phù hợp khi cần một giải pháp nhanh, ổn định và dễ triển khai trong bài toán phân loại cảm xúc với 2 nhãn.

3.6. So sánh và kết luận

Mô hình	Đặc điểm	Accuracy	Ưu điểm	Nhược điểm
Logistic Regression	Mô hình tuyến tính, hiệu quả cao trên dữ liệu đã vector hóa.	92.7%	- Accuracy cao, cân bằng precision & recall. - Dễ áp dụng và triển khai.	- Hạn chế nếu dữ liệu có quan hệ phi tuyến. - Phụ thuộc vào vector hóa.
SVM	Tìm siêu phẳng tối ưu phân tách dữ liệu, mạnh với dữ liệu có chiều cao.	93.7%	- Cân bằng tốt giữa precision và recall. - F1-score cao, ổn định.	- Huấn luyện lâu với tập dữ liệu lớn. - Nhạy cảm với lựa chọn tham số.
Random Forest	Tổ hợp nhiều cây quyết định, xử lý tốt quan hệ phi tuyến.	94.2%	- Accuracy cao nhất. - Ổn định, ít overfitting. - Khả năng giải thích đặc	- Mô hình phức tạp, khó giải thích chi tiết. - Sai nhãn tiêu cực còn xuất

			trưng tốt.	hiện.
Naive Bayes	Mô hình xác suất đơn giản, giả định đặc trưng độc lập nhau.	87.3%	- Huấn luyện nhanh, dễ triển khai. - Precision & recall tương đối cân bằng.	- Giả định độc lập không thực tế với dữ liệu text.- Độ chính xác thấp hơn so với mô hình khác.

Bảng 3.6.1. Bảng so sánh các mô hình phân loại cảm xúc (2 nhãn)

Kết luận:

Qua quá trình so sánh bốn mô hình Naïve Bayes, Random Forest, SVM và Logistic Regression, có thể thấy rằng **Random Forest đạt accuracy cao nhất (94.2%)**, tuy nhiên mô hình phức tạp, tốn nhiều tài nguyên và khó triển khai trực quan trên ứng dụng web. **SVM** tuy cho kết quả tốt (93.7%) nhưng thời gian huấn luyện lâu và nhạy cảm với tham số, khó mở rộng với dữ liệu lớn. **Naïve Bayes** có tốc độ nhanh nhưng độ chính xác thấp hơn (87.3%), chưa phù hợp để đảm bảo yêu cầu chất lượng trong ứng dụng thực tế.

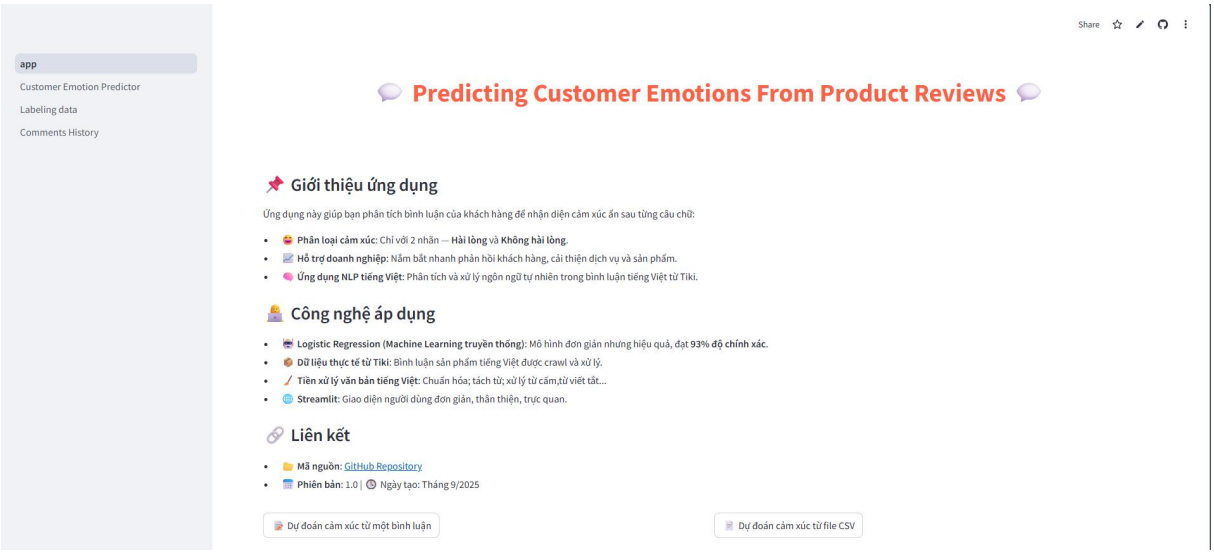
Trong khi đó, **Logistic Regression đạt accuracy 92.7%, cân bằng tốt giữa precision và recall, mô hình đơn giản, dễ huấn luyện và dễ triển khai**. Đặc biệt, Logistic Regression hoạt động hiệu quả với dữ liệu văn bản đã được vector hóa, giúp tiết kiệm tài nguyên và phù hợp cho mục tiêu của đồ án tốt nghiệp.

Vì vậy, mô hình **Logistic Regression** được lựa chọn để triển khai trên ứng dụng Streamlit trong đề tài đồ án tốt nghiệp.

3.7. Triển khai mô hình

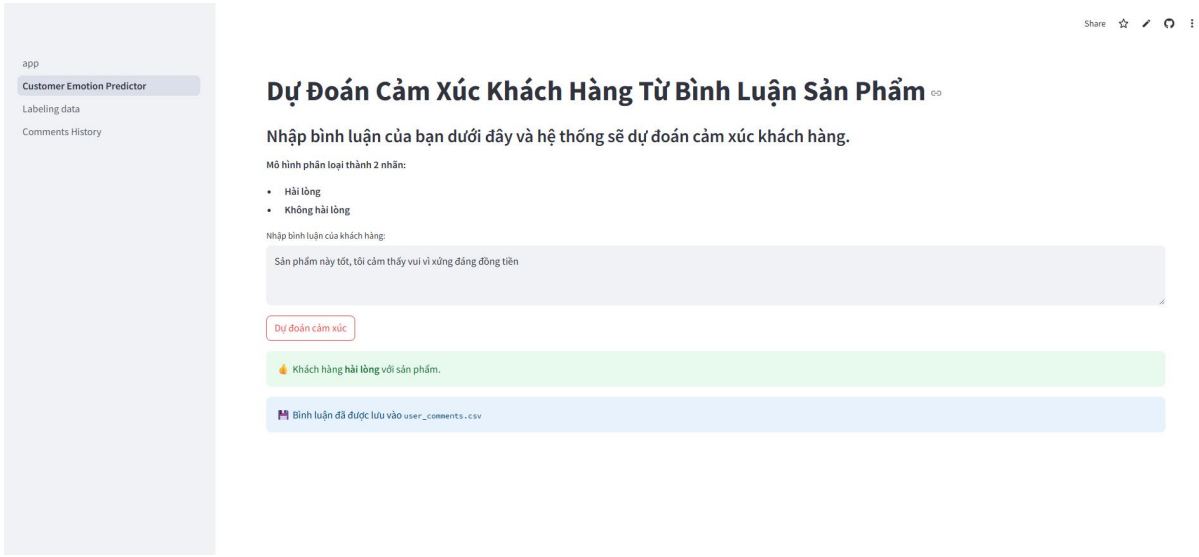
Sau khi hoàn thành quá trình huấn luyện, mô hình Logistic Regression cùng với bộ vectorizer được lưu trữ bằng thư viện joblib dưới dạng các tệp .pkl, bao gồm lr_model_2label.pkl và count_2label.pkl. Cách lưu trữ này giúp mô hình có thể được nạp lại và sử dụng nhiều lần mà không cần huấn luyện lại, đảm bảo tiết kiệm thời gian và giữ tính nhất quán trong triển khai. Trong trường hợp các tệp này chưa tồn tại, hệ thống sẽ tự động gọi hàm huấn luyện từ module utils.func, sau đó sinh ra mô hình mới và ghi đè vào bộ nhớ lưu trữ. Ngoài ra, hệ thống còn sử dụng một tệp văn bản bad_words.txt để kiểm soát và phát hiện các từ ngữ cấm trong quá trình dự đoán, nhằm đảm bảo tính hợp lệ của dữ liệu đầu vào.

Việc triển khai mô hình được thực hiện thông qua một ứng dụng web xây dựng bằng **Streamlit**. Ứng dụng này được thiết kế gồm ba thành phần chính và trang chủ giới thiệu.

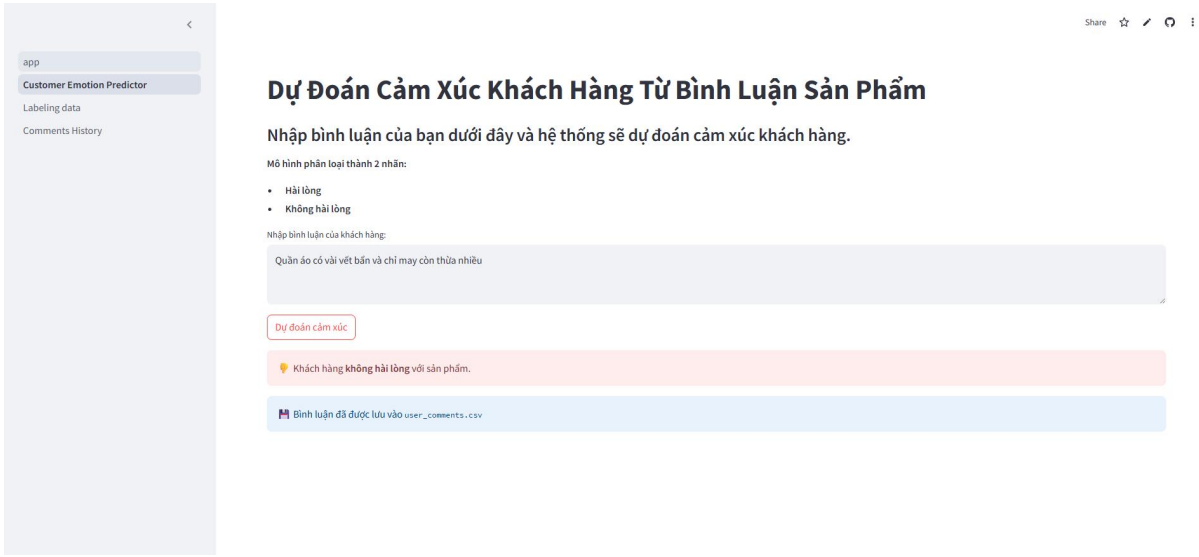


Hình 3.7.1 Giao diện trang chủ của ứng dụng

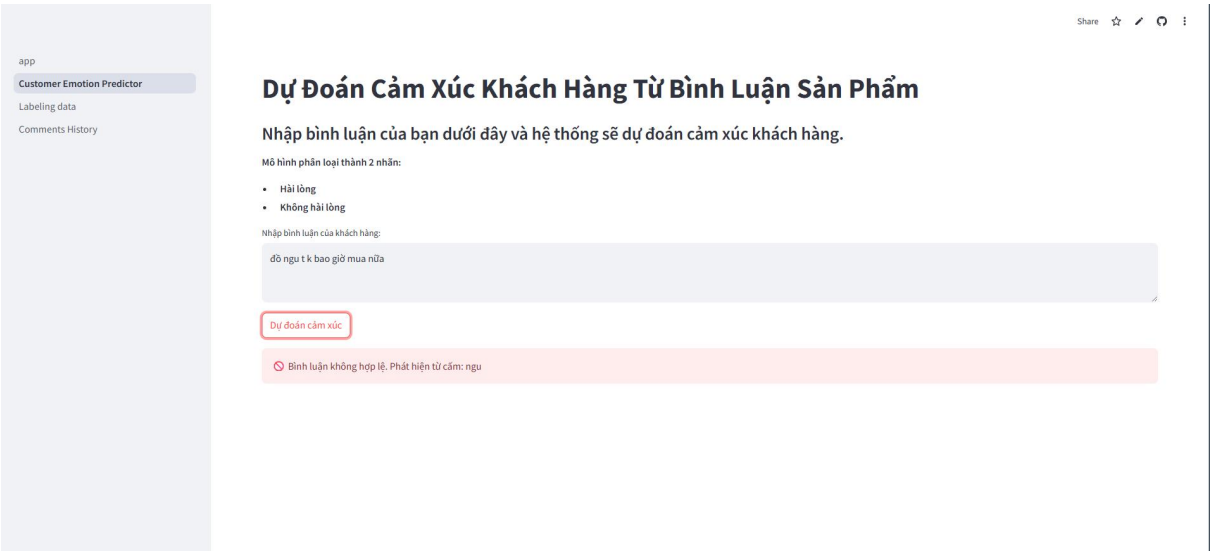
Trang thứ nhất là chức năng dự đoán cảm xúc từ một bình luận đơn lẻ, nơi người dùng nhập văn bản vào khung nhập liệu, hệ thống tiến hành chuẩn hóa, tách từ, loại bỏ từ dừng và kiểm tra từ cấm trước khi đưa vào vectorizer và mô hình Logistic Regression để phân loại. Kết quả dự đoán được hiển thị trực quan với hai trạng thái "Hài lòng" hoặc "Không hài lòng". Đồng thời, bình luận và nhãn dự đoán sẽ được ghi vào file "user_comments.csv" để phục vụ việc lưu vết và phân tích sau này.



Hình 3.7.2 Đánh giá sản phẩm với nhãn Hài lòng

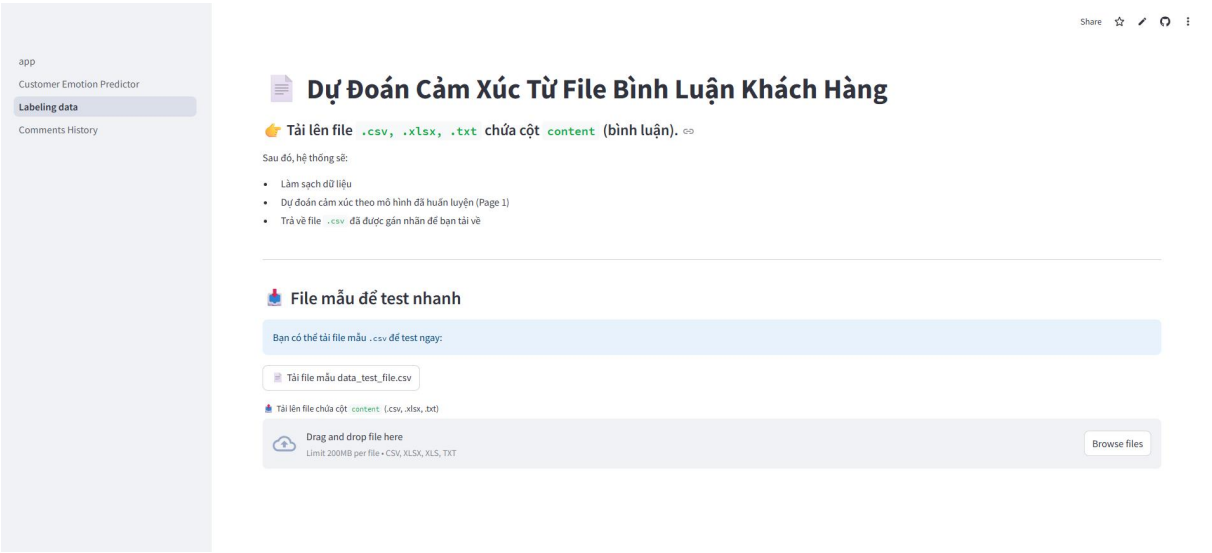


Hình 3.7.3 Đánh giá sản phẩm với nhãn Không hài lòng

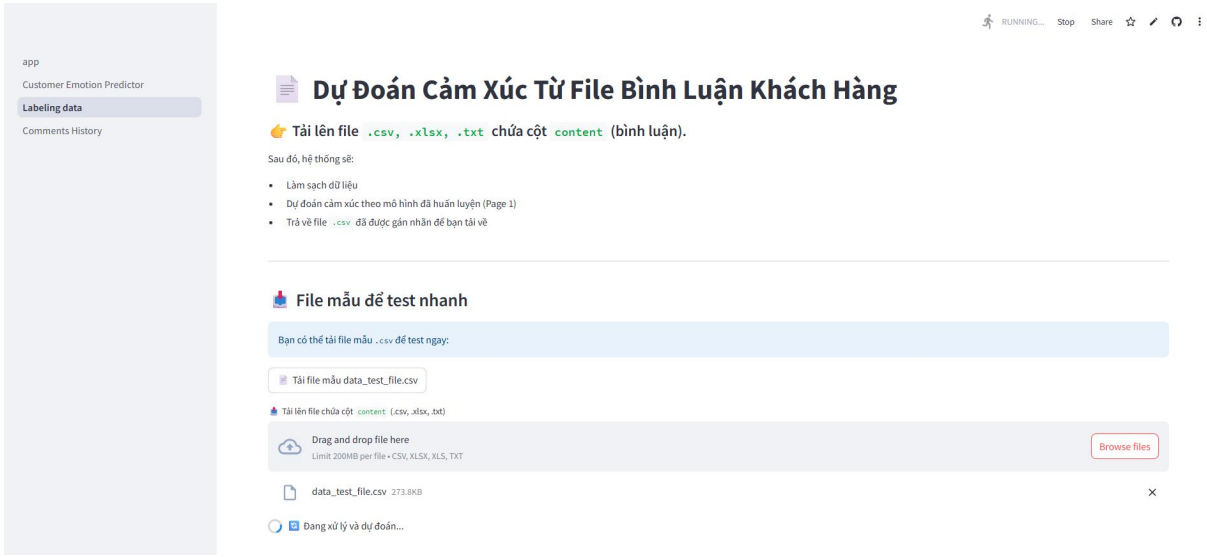


Hình 3.7.4 Bình luận không hợp lệ

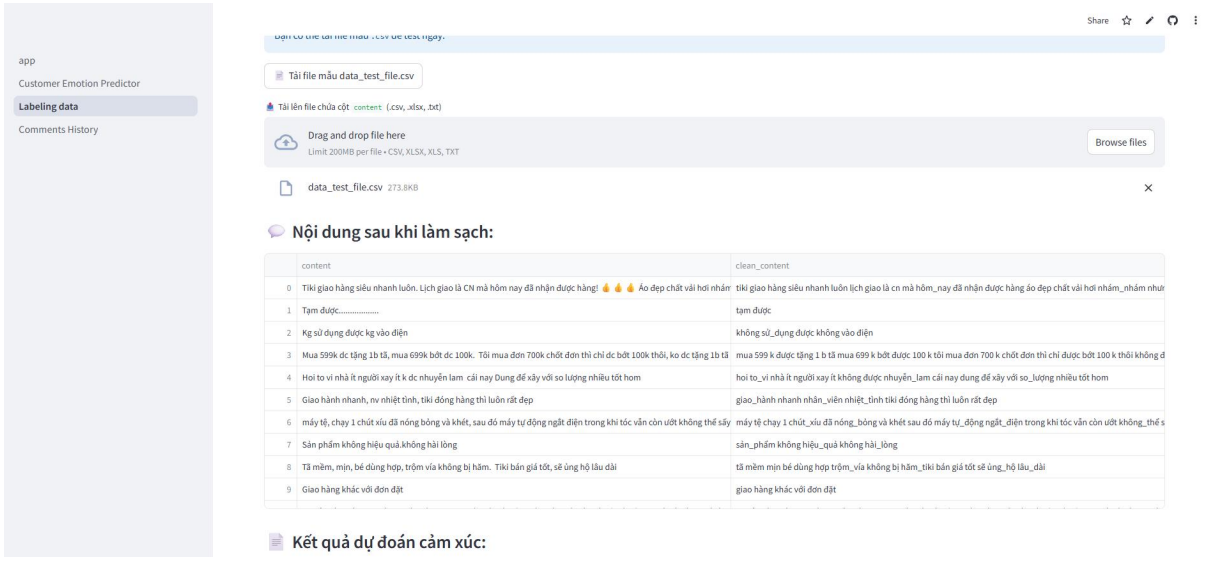
Trang thứ hai là chức năng dự đoán cảm xúc từ một tập dữ liệu, cho phép người dùng tải lên file .csv, .xlsx hoặc .txt có chứa cột content. Hệ thống sẽ tự động kiểm tra cấu trúc file, thực hiện các bước tiền xử lý văn bản, vector hóa và dự đoán nhãn cho toàn bộ tập dữ liệu. Kết quả dự đoán không chỉ được hiển thị trực tiếp trên giao diện dưới dạng bảng dữ liệu mà còn có thể được tải về dưới dạng file CSV mới. Đồng thời, ứng dụng còn cung cấp biểu đồ tròn thể hiện tỷ lệ các nhãn "Hài lòng" và "Không hài lòng", giúp người dùng có cái nhìn trực quan về phân bố cảm xúc trong tập dữ liệu.

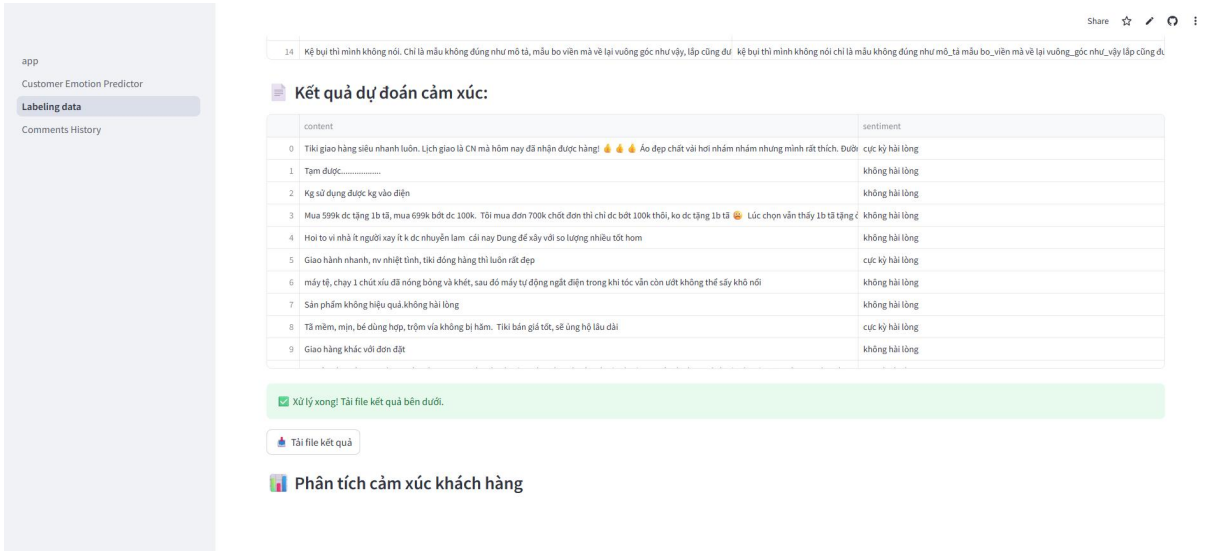


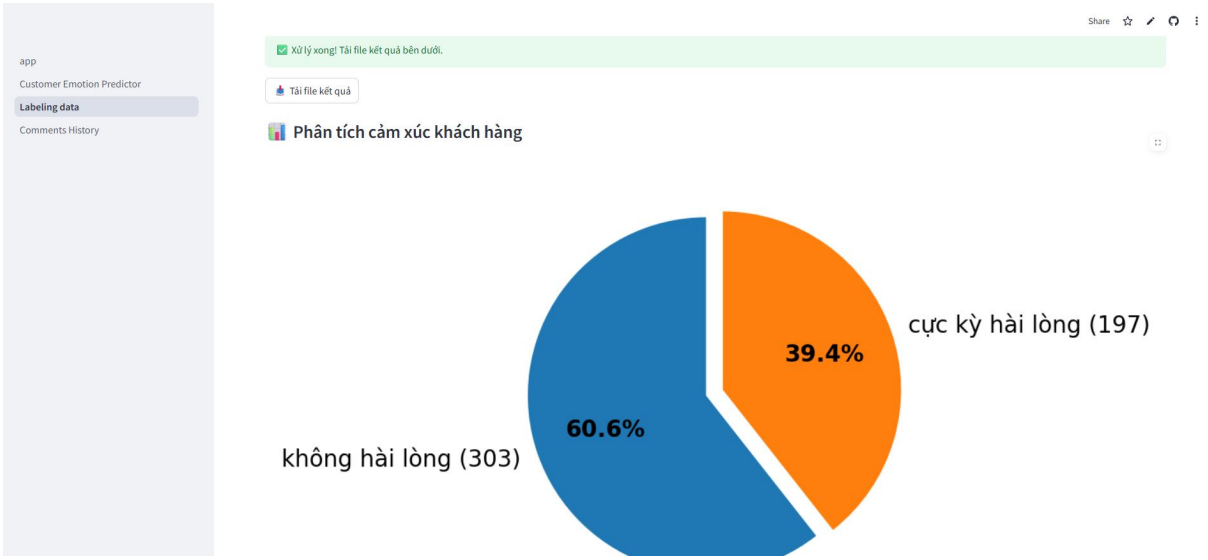
Hình 3.7.5 Giao diện dự đoán cảm xúc từ file



Hình 3.7.6 Giao diện khi thêm file vào để dự đoán

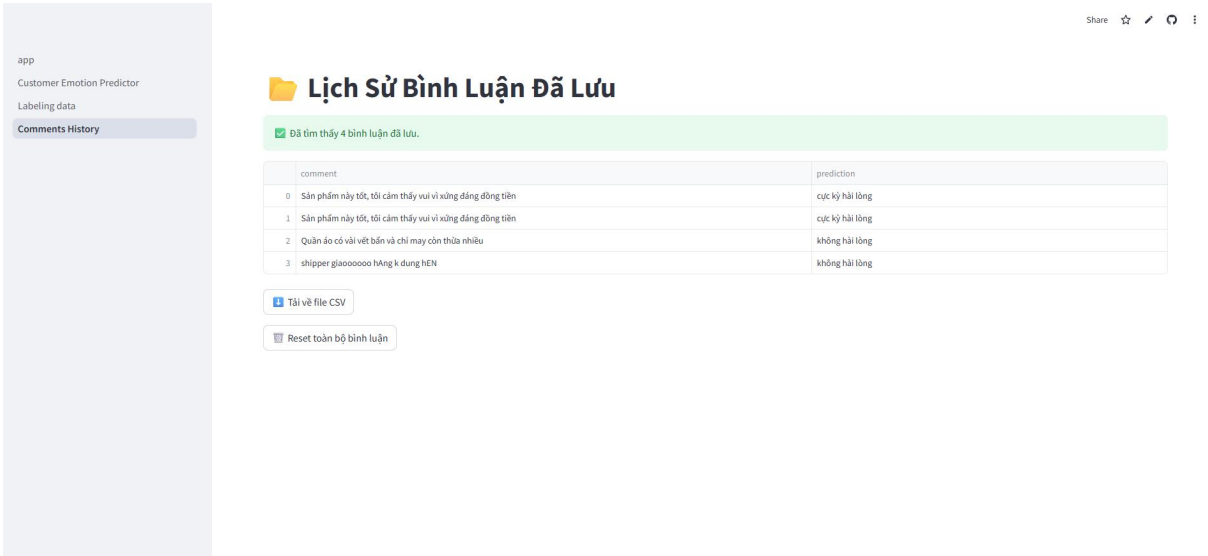




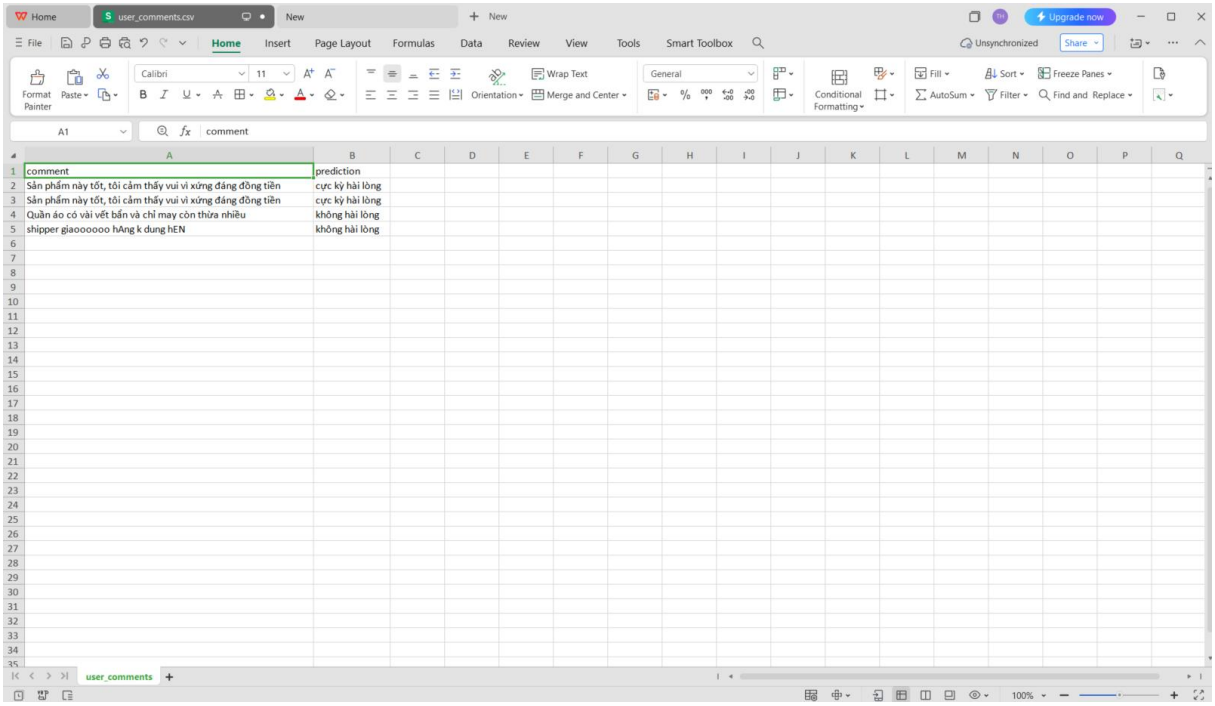


Hình 3.7.7 Kết quả sau khi dự đoán từ file

Cuối cùng, ứng dụng có thêm một trang quản lý lịch sử bình luận đã dự đoán. Trang này đọc dữ liệu từ file "user_comments.csv", hiển thị toàn bộ danh sách các bình luận kèm nhãn dự đoán, đồng thời cho phép người dùng tải file xuống để lưu trữ hoặc sử dụng trong các phân tích tiếp theo. Người dùng cũng có thể xóa toàn bộ lịch sử thông qua nút "Reset", lúc này hệ thống sẽ xóa file CSV và khởi động lại phiên làm việc.



Hình 3.7.8 Giao diện kết quả của trang lịch sử bình luận đã dự đoán



	A	B
1	comment	prediction
2	Sản phẩm này tốt, tôi cảm thấy vui vì xứng đáng đồng tiền	cực kỳ hài lòng
3	Sản phẩm này tốt, tôi cảm thấy vui vì xứng đáng đồng tiền	cực kỳ hài lòng
4	Quần áo có vài vết bẩn và chỉ may còn thừa nhiều	không hài lòng
5	shipper giaooooooo hAng k dung hEN	không hài lòng

Hình 3.7.9 File lịch sử bình luận sau khi được tải về

Mô hình sau khi huấn luyện đã được đóng gói và tích hợp thành công trong ứng dụng web Streamlit, với đầy đủ các chức năng như dự đoán cảm xúc từ một bình luận, dự đoán theo tập dữ liệu và quản lý lịch sử dự đoán. Nhờ vậy, pipeline xử lý dữ liệu end-to-end từ khâu thu thập, tiền xử lý, huấn luyện, đánh giá đến triển khai đã được hiện thực hóa một cách trọn vẹn. Điều này không chỉ chứng minh tính khả thi của giải pháp trong môi trường thực tế mà còn đặt nền móng cho việc mở rộng trong tương lai, như phát triển phân loại đa nhãn cảm xúc, tích hợp các phương pháp vector hóa hiện đại hơn hoặc triển khai trên môi trường dữ liệu lớn.

* Ứng dụng đã triển khai: https://customer-emotion-prediction.streamlit.app/

Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1. Thành tựu

Đề tài đã hoàn thành mục tiêu xây dựng một hệ thống phân tích cảm xúc khách hàng từ bình luận sản phẩm tiếng Việt với pipeline xử lý dữ liệu end-to-end. Cụ thể, thực hiện thành công các bước từ thu thập dữ liệu, tiền xử lý văn bản tiếng Việt (chuẩn hóa, tách từ, loại bỏ stopwords, xử lý từ viết tắt và từ cấm), vector hóa dữ liệu bằng TF-IDF, đến huấn luyện và đánh giá nhiều mô hình học máy. Trong số các mô hình thử nghiệm, Logistic Regression được lựa chọn vì đạt hiệu suất cao nhất với độ chính xác khoảng 93% trên tập dữ liệu thực tế. Ngoài ra, đề tài đã triển khai mô hình vào một ứng dụng web xây dựng bằng Streamlit, hỗ trợ dự đoán cảm xúc từ một bình luận, dự đoán theo tập dữ liệu, và trực quan hóa kết quả qua biểu đồ. Ứng dụng còn cho phép lưu trữ và quản lý lịch sử dự đoán, tạo tiền đề để tích hợp vào các hệ thống phân tích dữ liệu doanh nghiệp.

4.2. Hạn chế

Mặc dù đạt được những kết quả khả quan, đề tài vẫn tồn tại một số hạn chế nhất định. Thứ nhất, dữ liệu huấn luyện còn hạn chế về quy mô và mất cân bằng nhãn, dẫn đến độ chính xác của mô hình đối với các bình luận tiêu cực chưa thực sự cao. Thứ hai, hệ thống hiện tại mới chỉ phân loại cảm xúc theo hai nhãn cơ bản là “Hài lòng” và “Không hài lòng”, chưa phản ánh được mức độ cảm xúc đa dạng hơn như “rất hài lòng”, “bình thường” hay “tức giận”. Thứ ba, phương pháp vector hóa TF-IDF tuy đơn giản và hiệu quả, nhưng chưa khai thác được ngữ cảnh sâu và mối quan hệ ngữ nghĩa giữa các từ. Ngoài ra, ứng dụng hiện vẫn lưu trữ dữ liệu lịch sử dưới dạng CSV, chưa tích hợp cơ sở dữ liệu chuyên dụng để quản lý hiệu quả hơn khi quy mô hệ thống tăng lên.

4.3. Hướng khắc phục và phát triển

Mặc dù đạt được những kết quả khả quan, đề tài vẫn tồn tại một số hạn chế nhất định. Thứ nhất, dữ liệu huấn luyện còn hạn chế về quy mô và mất cân bằng nhãn, dẫn đến độ chính xác của mô hình đối với các bình luận tiêu cực chưa thực sự cao. Thứ hai, hệ

thống hiện tại mới chỉ phân loại cảm xúc theo hai nhãn cơ bản là “Hài lòng” và “Không hài lòng”, chưa phản ánh được mức độ cảm xúc đa dạng hơn như “rất hài lòng”, “bình thường” hay “tức giận”. Thứ ba, phương pháp vector hóa TF-IDF tuy đơn giản và hiệu quả, nhưng chưa khai thác được ngữ cảnh sâu và mối quan hệ ngữ nghĩa giữa các từ. Ngoài ra, ứng dụng hiện vẫn lưu trữ dữ liệu lịch sử dưới dạng CSV, chưa tích hợp cơ sở dữ liệu chuyên dụng để quản lý hiệu quả hơn khi quy mô hệ thống tăng lên.

TÀI LIỆU THAM KHẢO

[1] J. Kaur, "NLP for Sentiment Analysis in Customer Feedback," *XenonStack Blog*, Aug. 23, 2024. [Trực tuyến]. Available: https://www.xenonstack.com/blog/nlp-for-sentiment-analysis. [Truy cập: 04/09/2025]

[2] "Sentiment Analysis là gì? Tất tần tật về AI phân tích cảm xúc," *InterData*. [Trực tuyến]. Available: https://interdata.vn/blog/sentiment-analysis-la-gi/. [Truy cập: 04/09/2025]

[3] "Giải Mã Dữ Liệu: Kỹ Thuật Xử Lý Ngôn Ngữ Tiếng Việt (Vietnamese NLP)," *Evotek Careers*. [Trực tuyến]. Available: https://tuyendung.evotek.vn/giai-ma-du-lieu-ky-thuat-xu-ly-ngon-ngu-tieng-viet-vietnamese-nlp/. [Truy cập: 04/09/2025]

[4] "Mô hình phân lớp Naive Bayes," *Viblo*. [Trực tuyến]. Available: https://viblo.asia/p/mo-hinh-phan-lop-naive-bayes-vyDZO0A7lwj. [Truy cập: 04/09/2025]

[5] "Thuật toán Random Forest: Giải thích chi tiết và ứng dụng," *AIcandy*. [Trực tuyến]. Available: https://aicandy.vn/thuat-toan-random-forest-giai-thich-chi-tiet-va-ung-dung/. [Truy cập: 04/09/2025]

[6] "Support Vector Machine là gì?," *InterData*. [Trực tuyến]. Available: https://interdata.vn/blog/support-vector-machine-la-gi/. [Truy cập: 04/09/2025]

[7] "Logistic Regression - Bài toán cơ bản trong Machine Learning," *Viblo*. [Trực tuyến]. Available: https://viblo.asia/p/logistic-regression-bai-toan-co-ban-trong-machine-learning-924lJ4rzKPM. [Truy cập: 04/09/2025]

[8] "Using CountVectorizer for NLP feature extraction," *IBM Documentation*. [Trực tuyến]. Available: https://www.ibm.com/reference/python/countvectorizer. [Truy cập: 04/09/2025]

[9] "TF–IDF (term frequency – inverse document frequency)," *Viblo*. [Trực tuyến]. Available: https://viblo.asia/p/tf-idf-term-frequency-inverse-document-frequency-JQVkVZgKkyd. [Truy cập: 04/09/2025]

[10] N. T. Huy, *Luận văn Thạc sĩ: Hệ thống Thông tin*. Học viện Công nghệ Bưu chính Viễn thông, TP.HCM, 2020. [Trực tuyến]. Available: https://ptithcm.edu.vn/wp-content/uploads/2023/07/2020_HTTT_NguyenThanhHuy_LV.pdf. [Truy cập: 04/09/2025]