**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**VIỆN ĐÀO TẠO CÔNG NGHỆ THÔNG TIN, CHUYỂN ĐỔI SỐ**

**\*\*\*\*\*\*\*\*\*\*\***



**BÁO CÁO TIỂU LUẬN**

**XÂY DỰNG VÀ TÍCH HỢP API NHẬN DIỆN BÌNH LUẬN TIÊU CỰC**

**Sinh viên thực hiện : Nguyễn Duy Dương**

**Lớp : D21KTPM02**

**Khoá : 2024-2025**

**Ngành : Kỹ thuật phần mềm**

**Sinh viên thực hiện : Nguyễn Minh Khôi**

**Lớp : D21KTPM02**

**Khoá : 2024-2025**

**Ngành : Kỹ thuật phần mềm**

**Giảng viên hướng dẫn : TS. Huỳnh Nguyễn Thành Luân**

**Bình Dương, tháng 10/2024**

TÓM TẮT

API nhận diện bình luận tiêu cực được thực hiện sau quá trình thu thập và khảo sát thông tin từ các nguồn dữ liệu bình luận trực tuyến và bộ dữ liệu IMDB. API giúp người sử dụng phân loại và phát hiện nhanh các bình luận tiêu cực để quản lý nội dung hiệu quả hơn, góp phần xây dựng môi trường giao tiếp lành mạnh.

Dựa trên nền tảng Flask song song là ngôn ngữ Python cùng với công cụ lưu trữ cơ sở dữ liệu Pickle cho việc lưu mô hình và vectorizer.

Cấu trúc bài báo cáo được chia thành 6 phần:

Chương 1: Giới thiệu tổng quan về đề tài, lý do chọn đề tài và các mục tiêu nghiên cứu.

Chương 2: Cơ sở lý thuyết liên quan đến nhận diện cảm xúc, xử lý ngôn ngữ tự nhiên (NLP), và thuật toán Naive Bayes.

Chương 3: Phân tích và thiết kế hệ thống, bao gồm kiến trúc API và cách thức hoạt động của hệ thống nhận diện.

Chương 4: Triển khai và hiện thực hệ thống, từ xử lý dữ liệu, huấn luyện mô hình đến xây dựng API.

Chương 5: Đánh giá kết quả của hệ thống, phân tích độ chính xác và hiệu quả của API khi tích hợp vào website.

Chương 6: Kết luận và đề xuất hướng phát triển trong tương lai.

.

MỞ ĐẦU

Trong thời đại số hiện nay, việc duy trì môi trường tương tác lành mạnh trên các nền tảng trực tuyến là vô cùng quan trọng. Để đảm bảo trải nghiệm người dùng tích cực và giảm thiểu các bình luận tiêu cực, chúng tôi đã phát triển một API nhận diện cảm xúc cho bình luận, đặc biệt là nhằm xác định các bình luận tiêu cực. API này sẽ được tích hợp vào website, giúp tự động hóa quá trình phân loại cảm xúc, từ đó hỗ trợ đội ngũ quản trị viên quản lý bình luận hiệu quả hơn.

MỤC LỤC

[TÓM TẮT 1](#_Toc11341)

[MỞ ĐẦU 2](#_Toc17046)

[MỤC LỤC 3](#_Toc732)

[DANH MỤC HÌNH 4](#_Toc23194)

[CHƯƠNG 1. TỔNG QUAN 1](#_Toc18438)

[1.1. Giới thiệu đề tài 1](#_Toc18267)

[1.2. Công nghệ sử dụng 1](#_Toc748)

[1.2.1. Giới thiệu về Visual studio code 1](#_Toc11448)

[1.2.2. Giới thiệu về Python 1](#_Toc12132)

[1.2.3. Giới thiệu về Flask 1](#_Toc12947)

[CHƯƠNG 2. TIỀN XỬ LÝ DỮ LIỆU 2](#_Toc3523)

[2.1. THU THẬP DỮ LIỆU 2](#_Toc21873)

[2.2. Mô tả dữ liệu 2](#_Toc7257)

[2.2.1. Review (Bình luận phim) 2](#_Toc12469)

[2.2.2. Sentiment (Cảm xúc): 3](#_Toc21072)

[2.3. TRỰC QUAN HÓA DỮ LIỆU 3](#_Toc12508)

[CHƯƠNG 3. THIẾT KẾ MÔ HÌNH 4](#_Toc28044)

[3.1. giới thiệu mô hình 4](#_Toc31967)

[3.1.1. Mô hình Logistic 4](#_Toc19819)

[3.1.2. Mô hình Naive Bayes 6](#_Toc13356)

[3.2. GIAI ĐOẠN HUẤN LUYỆN (training) 9](#_Toc15086)

[3.2.1. Phân chia tập huấn luyện 9](#_Toc26886)

[3.2.2. Huấn luyện mô hình 10](#_Toc32084)

[3.2.2.1. Mô hình Naive Bayes 10](#_Toc29590)

[3.2.2.2. Mô hình Logistic 12](#_Toc2688)

[3.3. giai đoạn đánh giá (testing) 13](#_Toc24357)

[CHƯƠNG 4. ĐÁNH GIÁ THỰC NGHIỆM 17](#_Toc16829)

[4.1. kết quả thực nghiệm 18](#_Toc12883)

[4.1. Chuẩn hóa dữ liệu 18](#_Toc10659)

[4.2. Kết quả dự đoán mô hình Logistic 20](#_Toc29161)

[4.3. Kết quả dự đoán mô hình Naive Bayes 22](#_Toc17436)

[4.4. Kết quả sau khi Training 22](#_Toc29089)

[4.5. Triển khai website nhận diện bình luận tiêu cực bằng Naive Bayes 23](#_Toc8006)

[KếT LUẬN 24](#_Toc11074)

[TÀI LIỆU THAM KHẢO 26](#_Toc31916)

DANH MỤC HÌNH

[DANH MỤC HÌNH 4](#_Toc28015)

[*Hình 2.1 -* Biểu đồ phân phối độ dài bình luận 3](#_Toc6359)

[*Hình 2.2 -* Boxplot độ dài bình luận theo cảm xúc 3](#_Toc6561)

[*Hình 3.1 -* Cách hoạt động của mô hình Logistic 6](#_Toc5681)

[*Hình 3.2 -* Cách hoạt động của mô hình Naive Bayes 9](#_Toc19435)

[*Hình 4.1 -* Lọc top 20 từ phổ biến nhất 18](#_Toc18470)

[*Hình 4.2 -* Phân bố độ dài văn bản 18](#_Toc25477)

[*Hình 4.3 -* Tỷ lệ nhãn trong dữ liệu huấn luyện 19](#_Toc24301)

[*Hình 4.4 -* Dữ liệu sau khi giảm chiều với PCA 20](#_Toc29912)

[*Hình 4.5 -* Biểu đồ hàm Loss mô hình Logistic 20](#_Toc2206)

[*Hình 4.6 -* Biểu đồ Accuracy mô hình Logistic 21](#_Toc4279)

[*Hình 4.7 -* Ma trận nhầm lẫn mô hình Logistic 21](#_Toc22783)

[*Hình 4.8 -* Ma trận nhầm lẫn mô hình Naive Bayes 22](#_Toc13280)

[*Hình 4.9 -* Hình ảnh website nhận diện bình luận tiêu cực bằng Naive Bayes 23](#_Toc21902)

DANH SÁCH CÁC KÝ TỰ, CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| **Từ viết tắt** | **Giải thích** |
| PHP | Hypertext Preprocessor |
| MVC | Model – View – Controller |
| CSDL | Cơ sở dữ liệu |
| CLB | Câu lạc bộ |
| HLV | Huấn luyện viên |
| ND | Nội dung |

2. TỔNG QUAN
   1. **Giới thiệu đề tài**

Trong thời đại công nghệ phát triển vượt bậc, tương tác giữa người dùng và các nền tảng trực tuyến ngày càng trở nên phổ biến. Các trang mạng xã hội, diễn đàn, và website cung cấp nhiều tính năng để người dùng thể hiện cảm xúc và ý kiến. Tuy nhiên, việc kiểm soát nội dung tiêu cực trong phần bình luận là một thách thức lớn đối với các quản trị viên. Đề tài này tập trung vào việc xây dựng và tích hợp một API có khả năng tự động nhận diện và phân loại cảm xúc bình luận, đặc biệt là xác định các bình luận tiêu cực để hỗ trợ quản trị viên quản lý nội dung hiệu quả hơn.

* 1. **Công nghệ sử dụng**

Đề tài sử dụng các công nghệ và phương pháp chính sau đây:

Visual studio code: môi trường phát triển

Python: Ngôn ngữ lập trình chính cho việc xây dựng hệ thống.

Flask: Framework để xây dựng và triển khai API.

* + 1. Giới thiệu về Visual studio code

Visual Studio Code (VS Code) là một môi trường phát triển tích hợp (IDE) mã nguồn mở và miễn phí, được phát triển bởi Microsoft. Nó được thiết kế để hỗ trợ lập trình viên làm việc với mã nguồn hiệu quả hơn bằng cách cung cấp các tính năng hiện đại, tiện dụng.

* + 1. *Giới thiệu về Python*

Python là ngôn ngữ lập trình chính được chọn để xây dựng API nhận diện bình luận tiêu cực. Đây là lựa chọn phù hợp vì:

* Có các thư viện mạnh mẽ cho xử lý ngôn ngữ tự nhiên như NLTK, TextBlob
* Hỗ trợ tốt cho việc xây dựng các mô hình học máy phân loại văn bản
* Dễ dàng tích hợp với các framework web để xây dựng API
* Cộng đồng lớn và nhiều tài liệu tham khảo về xử lý ngôn ngữ tự nhiên
  + 1. *Giới thiệu về Flask*

• Framework Python nhẹ và linh hoạt để xây dựng REST API

• Dễ dàng tạo các endpoint để tiếp nhận và xử lý request

• Hỗ trợ tốt việc xử lý request/response dạng JSON

• Có thể mở rộng với các middleware và extension

1. TIỀN XỬ LÝ DỮ LIỆU
   1. **THU THẬP DỮ LIỆU**

Dữ liệu được sử dụng trong dự án này được lấy từ bộ dữ liệu IMDB – một trong những tập dữ liệu phổ biến và đáng tin cậy nhất về các đánh giá phim. Bộ dữ liệu IMDB đã được sử dụng rộng rãi trong nghiên cứu và ứng dụng về nhận diện cảm xúc, xử lý ngôn ngữ tự nhiên (NLP) và học máy, do vậy độ tin cậy của nó đã được kiểm chứng bởi cộng đồng nghiên cứu.

Lý do chọn bộ dữ liệu IMDB thay vì các bộ dữ liệu khác

Độ phong phú và tính cân bằng: Bộ dữ liệu IMDB bao gồm một lượng lớn các bình luận phim được phân loại rõ ràng thành hai cảm xúc chính: tích cực và tiêu cực. Nhờ sự phong phú và cân bằng này, mô hình học máy có thể huấn luyện một cách hiệu quả mà không bị thiên lệch.

Phù hợp với bài toán phân loại cảm xúc: Vì các bình luận trên IMDB thường chứa nhiều từ ngữ thể hiện cảm xúc và ý kiến cá nhân rõ ràng, bộ dữ liệu này rất phù hợp để phát triển một hệ thống phân loại bình luận.

Được chuẩn hóa và dễ tiếp cận: Bộ dữ liệu IMDB đã qua xử lý sơ bộ và phân chia thành các tập huấn luyện và kiểm tra tiêu chuẩn. Điều này giúp tiết kiệm thời gian tiền xử lý dữ liệu và giúp mô hình có thể đạt hiệu quả cao hơn.

Vì những lý do trên, việc sử dụng bộ dữ liệu IMDB là lựa chọn hợp lý cho dự án này, giúp tối ưu hóa độ chính xác của API nhận diện bình luận tiêu cực đồng thời đảm bảo độ tin cậy của mô hình.

* 1. **Mô tả dữ liệu**
     1. *Review (Bình luận phim)*

Đây là trường chứa nội dung văn bản của các bình luận phim từ người dùng, mỗi bình luận bao gồm thông tin chi tiết về trải nghiệm hoặc ý kiến cá nhân của người dùng về bộ phim.

Ý nghĩa: Trường này cung cấp các đặc điểm ngôn ngữ thể hiện cảm xúc, đánh giá hoặc phê bình, giúp mô hình có đủ thông tin để phân tích và nhận diện cảm xúc.

Lý do sử dụng: review là trường trọng tâm của bài toán nhận diện cảm xúc vì nó bao gồm toàn bộ thông tin cần thiết để phân loại bình luận thành tích cực hay tiêu cực.

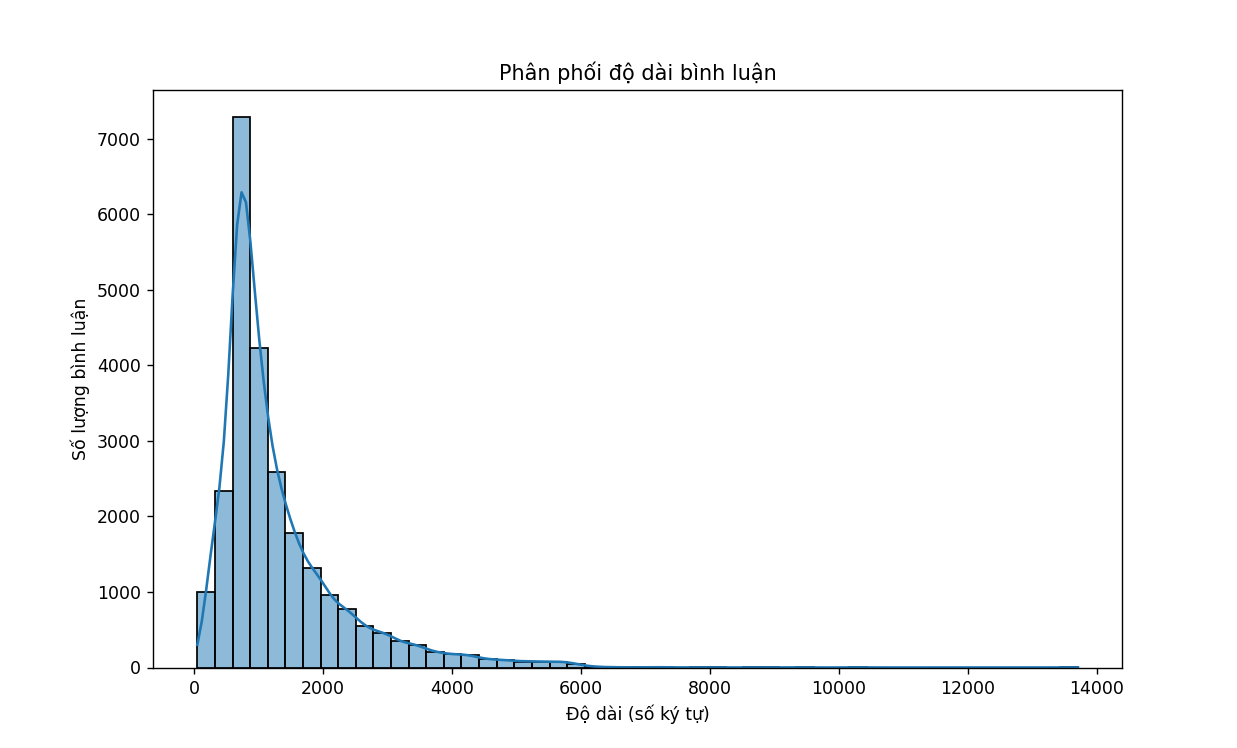
* + 1. *Sentiment (Cảm xúc):*

Trường này cho biết nhãn cảm xúc của bình luận, được gán giá trị là 1 cho cảm xúc tích cực và 0 cho cảm xúc tiêu cực.

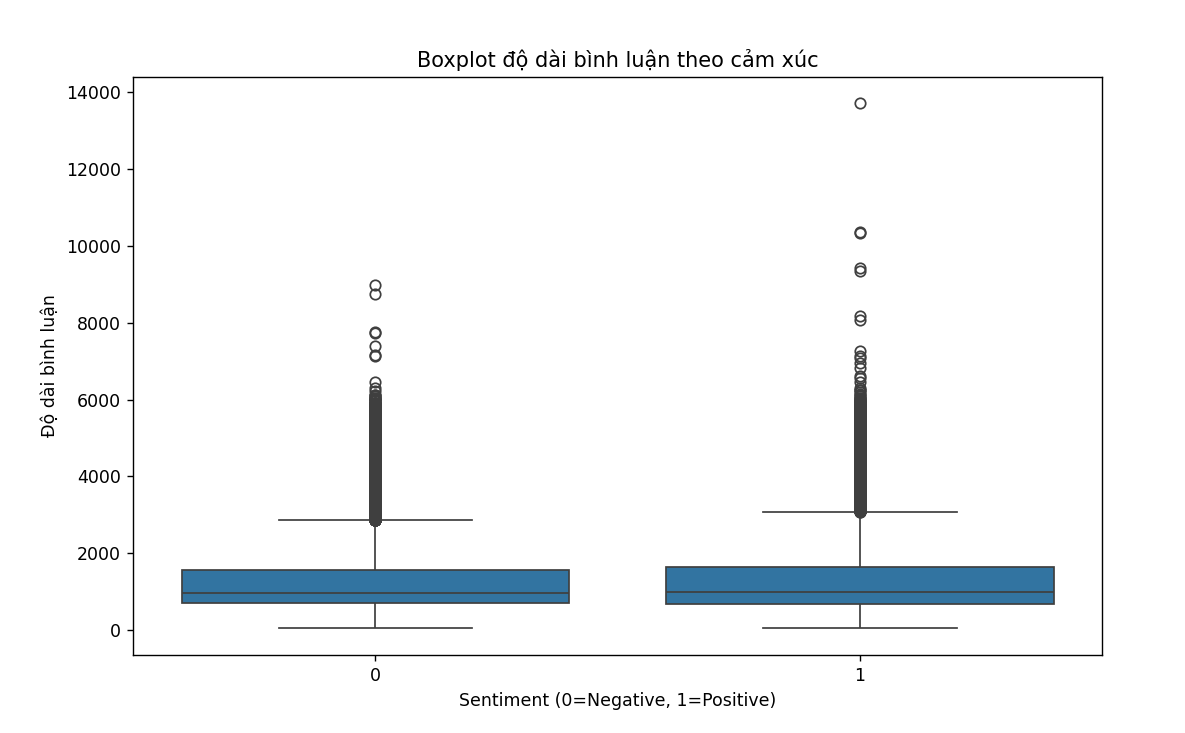
Ý nghĩa: Trường sentiment đóng vai trò là biến mục tiêu, cho phép mô hình học cách dự đoán cảm xúc dựa trên các đặc điểm ngôn ngữ từ trường review.

Lý do sử dụng: Trường này rất cần thiết trong quá trình huấn luyện mô hình. Nó giúp mô hình học từ các bình luận được gán nhãn sẵn, qua đó dự đoán chính xác cảm xúc của các bình luận mới.

2.3. TRỰC QUAN HÓA DỮ LIỆU



* + - 1. Biểu đồ phân phối độ dài bình luận

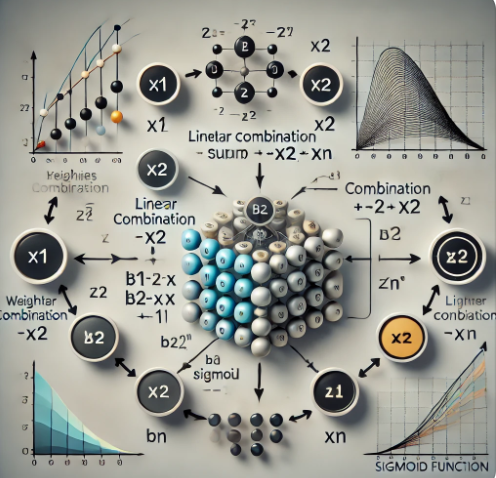


* + - 1. Boxplot độ dài bình luận theo cảm xúc

1. THIẾT KẾ MÔ HÌNH
   1. **giới thiệu mô hình**

***3.1.1. Mô hình Logistic***

Mô hình Logistic (Logistic Regression) là một phương pháp phân loại (classification) thuộc lĩnh vực học máy (machine learning) và thống kê. Nó được sử dụng để dự đoán xác suất của một biến phụ thuộc nhị phân (binary dependent variable), tức là một biến chỉ có hai giá trị (thường là 0 và 1, hoặc True và False).



Đầu vào (Input Layer):

Đây là nơi các biến số đầu vào (features) được đưa vào mô hình.

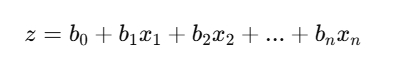
Các biến đầu vào được biểu diễn bởi các ký hiệu như x1, x2 x3,...xn đại diện cho các yếu tố ảnh hưởng đến kết quả cần dự đoán.

Đầu ra (Output): Xác suất (giá trị từ 0 đến 1), thường được sử dụng để phân loại thành hai nhóm (ví dụ: "có" hoặc "không").

Kết hợp tuyến tính (Linear Combination):

Các biến đầu vào được kết hợp với trọng số b1, b2, b3,...bn và hệ số tự do b0 để tạo ra giá trị z

Công thức tính



Ý nghĩa: Giá trị z là một kết quả trung gian, biểu thị mức độ tác động của các yếu tố đầu vào lên biến mục tiêu (target variable).

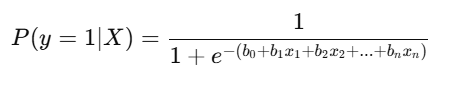
Hàm kích hoạt (Activation Function): Sử dụng hàm sigmoid

​

để biến đổi giá trị đầu ra của mô hình thành xác suất.

Biến đầu vào (Input): Có thể là liên tục hoặc phân loại (categorical), và thường được chuẩn hóa hoặc xử lý trước khi đưa vào mô hình.

Phương trình hồi quy logistic được biểu diễn như sau:

​

Trong đó:

P(y=1∣X): Xác suất biến mục tiêu y có giá trị bằng 1.

​

b : Các hệ số hồi quy cần được ước lượng từ dữ liệu.

x : Hàm tuyến tính kết hợp các đầu vào.

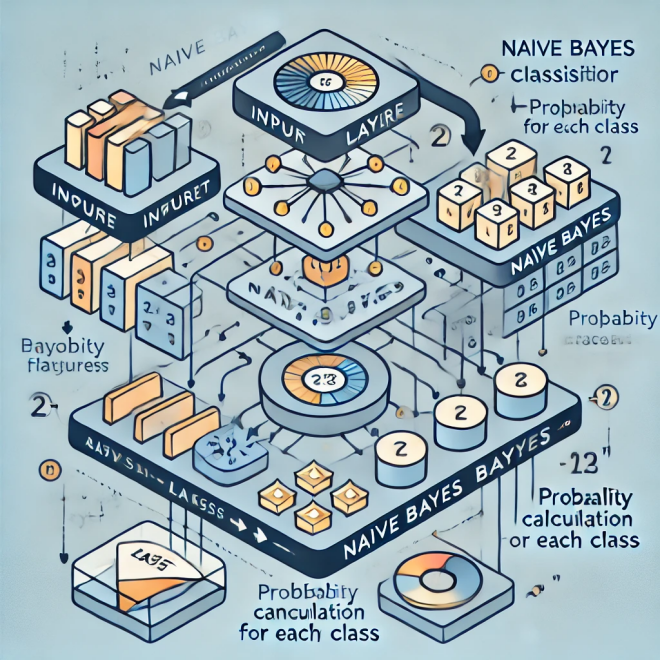
A diagram of a process

Description automatically generated

* + - 1. Cách hoạt động của mô hình Logistic

***3.1.2. Mô hình Naive Bayes***

Mô hình sử dụng trong dự án này là Multinomial Naive Bayes (MNB), một thuật toán học máy dựa trên lý thuyết xác suất Bayes, thường được dùng cho các bài toán phân loại văn bản như phân loại cảm xúc trong bình luận. MNB hoạt động hiệu quả với dữ liệu có đặc trưng rời rạc và tần suất xuất hiện của từ, phù hợp với dữ liệu phân loại bình luận thành "tích cực" hoặc "tiêu cực."



**Input Layer (Tầng đầu vào)**

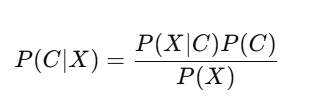
Dữ liệu đầu vào: Các đặc trưng (features) của dữ liệu được đưa vào hệ thống.

Ví dụ: Nếu áp dụng Naive Bayes vào phân loại email, các đặc trưng có thể là từ khóa trong email, số từ, hoặc tần suất xuất hiện của các từ nhất định.

Mỗi đặc trưng sẽ được xử lý độc lập theo giả định độc lập có điều kiện của Naive Bayes.

**Central Bayesian Classifier (Bộ phân loại Bayesian trung tâm)**

Tính xác suất hậu nghiệm: Thuật toán sử dụng công thức Bayes để tính toán xác suất của từng lớp P(C∣X), trong đó:



P(C): Xác suất tiên nghiệm của lớp C (dựa trên dữ liệu huấn luyện).

P(X∣C): Xác suất có điều kiện của dữ liệu đầu vào X thuộc lớp

P(X): Xác suất của đầu vào X, được tính như hằng số khi so sánh các lớp.

**Output Layer (Tầng đầu ra)**

Phân loại: Sau khi tính toán xác suất hậu nghiệm cho mỗi lớp, lớp có xác suất cao nhất sẽ được chọn làm kết quả đầu ra.

Ví dụ: Nếu bạn phân loại email, đầu ra có thể là Spam hoặc Không phải Spam.

**Luồng dữ liệu (Flow of data)**

Mũi tên: Thể hiện quá trình dữ liệu đi từ đầu vào qua các bước xử lý trong bộ phân loại Bayesian và đến tầng đầu ra.

Các bước được thực hiện tuần tự, từ việc tính xác suất, so sánh, và cuối cùng là phân loại.

Lý do chọn Multinomial Naive Bayes

Đơn giản và hiệu quả: MNB dễ triển khai và nhanh chóng khi xử lý dữ liệu lớn, do các tính toán được tối giản qua giả định độc lập giữa các từ trong bình luận.

Phù hợp với văn bản: Mô hình xử lý tốt khi dữ liệu có tần suất từ và biến rời rạc, giúp tối ưu hóa cho việc phân loại văn bản.

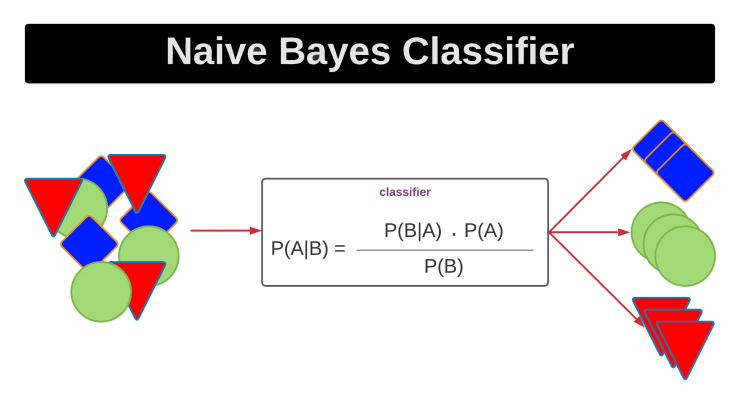
Giảm thiểu overfitting: Với cách tính xác suất có điều kiện và thêm kỹ thuật Laplace smoothing, MNB hạn chế overfitting, đặc biệt khi có từ hiếm gặp.

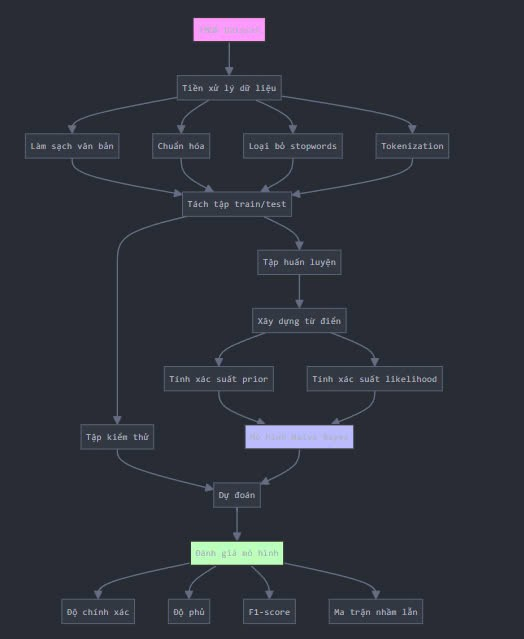
Cách mô hình hoạt động

Tiền xử lý văn bản: Bình luận sẽ được làm sạch, loại bỏ từ ngữ không quan trọng (stop words) và chuyển thành dạng số bằng phương pháp Bag of Words với CountVectorizer.

Huấn luyện mô hình: MNB sử dụng dữ liệu đã được vector hóa để tính toán xác suất cho mỗi loại cảm xúc (tích cực hoặc tiêu cực) dựa trên tần suất xuất hiện của từ trong từng lớp.

Dự đoán: Khi có bình luận mới, mô hình tính toán xác suất nó thuộc về từng lớp và chọn lớp có xác suất cao nhất làm kết quả dự đoán.





* + - 1. Cách hoạt động của mô hình Naive Bayes

* 1. **GIAI ĐOẠN HUẤN LUYỆN (training)**
     1. Phân chia tập huấn luyện

Quá trình truyền dữ liệu đầu vào

Dữ liệu đầu vào cho mô hình là các bình luận dạng văn bản thu thập từ tập dữ liệu IMDB, chứa các bình luận với nhãn “positive” hoặc “negative.” Dữ liệu này được truyền vào hàm xử lý clean\_text để làm sạch và chuyển đổi thành dạng mà mô hình có thể sử dụng, bao gồm:

Loại bỏ ký tự đặc biệt và HTML tags: Các ký tự không phải chữ cái và các thẻ HTML được loại bỏ để tránh nhiễu trong dữ liệu.

Tokenization và Lowercasing: Các từ được tách riêng và chuyển thành chữ thường, giúp việc phân loại không bị ảnh hưởng bởi viết hoa hay thường.

Loại bỏ Stop Words: Các từ không mang ý nghĩa như “is,” “and,” “the” được loại bỏ để giữ lại thông tin cốt lõi của bình luận.

Sau khi xử lý, mỗi bình luận trở thành một chuỗi các từ không chứa stop words và sẵn sàng cho quá trình vector hóa.

Vector hóa dữ liệu

Dữ liệu văn bản sau khi làm sạch được truyền vào CountVectorizer để chuyển đổi thành ma trận số (Bag of Words). Ma trận này có kích thước (n\_samples, n\_features) trong đó:

**n\_samples** là số bình luận trong tập dữ liệu.

**n\_features** là số lượng từ (hoặc từ ngữ) đặc trưng được chọn, ở đây là 5000.

Lưu trữ dữ liệu vector hóa

Dữ liệu vector hóa được lưu trong các biến X\_train\_vect và X\_test\_vect để sử dụng trong quá trình huấn luyện và đánh giá. Cụ thể:

**X\_train\_vect**: Chứa ma trận vector hóa của tập huấn luyện.

**X\_test\_vect**: Chứa ma trận vector hóa của tập kiểm tra.

Phân chia dữ liệu

Dữ liệu IMDB sau khi được vector hóa được chia thành hai tập:

80% cho tập huấn luyện (X\_train và y\_train), tương đương với khoảng 40,000 bình luận, dùng để huấn luyện mô hình.

20% còn lại cho tập kiểm tra (X\_test và y\_test), tương đương với 10,000 bình luận, dùng để đánh giá độ chính xác của mô hình trên dữ liệu mới.

Việc chia tỷ lệ này giúp mô hình học đủ từ tập dữ liệu và cũng đảm bảo rằng mô hình được kiểm tra trên một phần dữ liệu chưa từng thấy để xác định tính hiệu quả của mô hình trên dữ liệu thực tế.

* + 1. Huấn luyện mô hình

*3.2.2.1. Mô hình Naive Bayes*

Trong mô hình phân loại cảm xúc này, ta sử dụng mô hình Naive Bayes đa thức (Multinomial Naive Bayes). Mô hình này không sử dụng trọng số giống như các mạng nơ-ron hay mô hình hồi quy, nhưng sẽ tính toán xác suất cho từng từ xuất hiện trong mỗi lớp (positive hoặc negative). Sau đây là chi tiết về cách mô hình hoạt động, bao gồm các trọng số ban đầu, hàm truyền và các tính toán liên quan.

Trọng số ban đầu

Trong mô hình Naive Bayes, thay vì các trọng số cụ thể, ta sử dụng xác suất từ xuất hiện trong từng lớp dựa trên các lần xuất hiện trong dữ liệu huấn luyện. Các bước bao gồm:

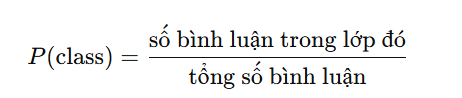
Tần suất từ trong từng lớp: Tạo ra bảng đếm số lần từ xuất hiện trong mỗi lớp (positive hoặc negative).

Xác suất từ xuất hiện trong lớp: Áp dụng Laplace Smoothing để tính toán xác suất điều kiện cho từng từ xuất hiện trong một lớp. Mỗi xác suất này là "trọng số" đặc trưng của từ trong lớp đó.

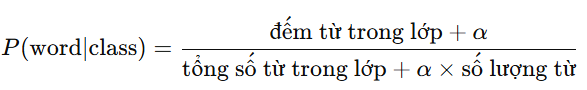
Hàm truyền sử dụng

Mô hình Naive Bayes không có hàm truyền dạng phi tuyến tính như trong mạng nơ-ron, mà sử dụng công thức xác suất để truyền dữ liệu qua các bước tính toán:

Xác suất tiên nghiệm (Prior Probability): Tính xác suất của từng lớp (positive và negative) trong tập huấn luyện. Được tính bằng:



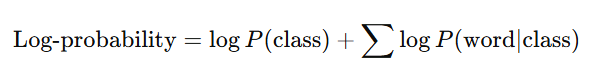
Xác suất điều kiện (Conditional Probability): Xác suất từ xuất hiện trong mỗi lớp được tính dựa trên số lần từ đó xuất hiện trong lớp chia cho tổng số từ trong lớp. Để tránh xác suất bằng 0, áp dụng Laplace Smoothing:



Với hệ số điều chỉnh α (alpha) được chọn là 1 để tránh số không trong tính toán.

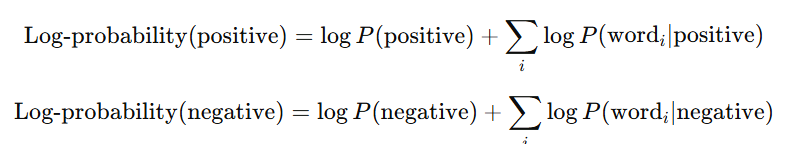
Đạo hàm

Trong Naive Bayes, việc tính toán không cần đạo hàm như các mô hình tối ưu hóa khác. Tuy nhiên, logarithmic transformation (phép biến đổi log) được sử dụng để tránh vấn đề "underflow" khi tính toán tích của các xác suất nhỏ. Công thức tính xác suất cho từng lớp được biến đổi thành:



Kết hợp các xác suất (Tính Log-probability)

Khi dự đoán cảm xúc cho một bình luận, mô hình tính tổng log-probability của tất cả các từ trong bình luận đó cho mỗi lớp:



Mô hình chọn lớp có log-probability lớn nhất làm kết quả dự đoán cho bình luận.

Tổng kết

Quá trình tính toán trong mô hình Naive Bayes bao gồm các bước đếm tần suất, tính xác suất, và chọn lớp dựa trên log-probability. Việc sử dụng Laplace Smoothing giúp tăng độ chính xác và tránh các vấn đề liên quan đến từ không xuất hiện trong tập huấn luyện.

### ***3.2.2.2. Mô hình Logistic***

Mô hình logistic regression (hồi quy logistic) là một thuật toán học máy phổ biến, được sử dụng để giải quyết các bài toán phân loại. Mặc dù tên gọi "hồi quy" có thể gây nhầm lẫn, logistic regression thực chất là một mô hình phân loại, không phải mô hình hồi quy dự đoán giá trị liên tục.

Hồi quy logistic dựa trên mô hình hồi quy tuyến tính, nhưng thay vì dự đoán một giá trị liên tục, nó sử dụng hàm logistic (hàm sigmoid) để biến đổi đầu ra thành giá trị xác suất nằm trong khoảng [0,1][0, 1][0,1].

Công thức hàm logistic:

A mathematical equation with numbers and symbols

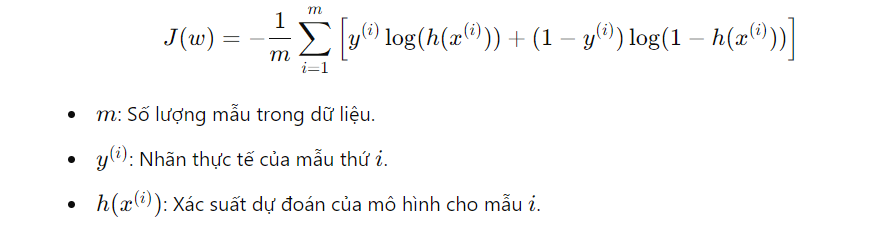
Description automatically generated

A white background with black text

Description automatically generated

Các bước huấn luyện:

**Xây dựng hàm mất mát (loss function):** Hồi quy logistic sử dụng hàm mất mát **Cross-Entropy** để đánh giá mức độ khác biệt giữa dự đoán và giá trị thực:



**Tối ưu hóa hàm mất mát:** Sử dụng các thuật toán tối ưu hóa như Gradient Descent để tìm bộ tham số www tối ưu, sao cho hàm mất mát đạt giá trị nhỏ nhất.

**Dự đoán:** Sau khi huấn luyện, mô hình có thể được sử dụng để dự đoán xác suất cho các mẫu mới và gán nhãn phân loại.

* 1. **giai đoạn đánh giá (testing)**

Trong quá trình đánh giá mô hình phân loại cảm xúc sử dụng Naive Bayes, có một số độ đo (hàm tổn thất) phổ biến mà chúng ta có thể áp dụng để xác định hiệu suất của mô hình. Dưới đây là các độ đo mà bạn có thể sử dụng, bao gồm công thức tính toán và lý do lựa chọn:

***Val loss and accuracy:***

Định nghĩa: Val loss (Validation Loss): Đây là một chỉ số đo lường mức độ sai lệch giữa dự đoán của mô hình và giá trị thực tế trên tập dữ liệu xác thực (validation set). Val loss cho thấy khả năng mô hình tổng quát hóa dữ liệu mà nó chưa từng thấy trong quá trình huấn luyện. Mục tiêu là giảm val loss trong suốt quá trình huấn luyện, vì điều này thường cho thấy mô hình học được tốt hơn.

Accuracy (Độ chính xác): Đây là tỷ lệ các dự đoán đúng của mô hình so với tổng số dự đoán. Accuracy thường được tính trên cả tập huấn luyện và tập xác thực để đánh giá hiệu suất mô hình. Trong ngữ cảnh của tập xác thực, val accuracy phản ánh khả năng của mô hình trong việc đưa ra dự đoán chính xác trên dữ liệu chưa được học trước đó.

Công thức tính:

A white background with black text

Description automatically generated

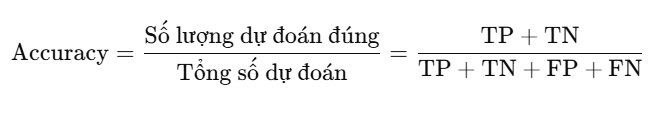
***Confusion Matrix:***

Định nghĩa: là một bảng tóm tắt kết quả hoạt động của một mô hình phân loại. Nó so sánh giữa giá trị **thực tế** và **dự đoán** của mô hình, giúp chúng ta hiểu được hiệu suất mô hình qua từng loại dự đoán

***Accuracy Score(Độ chính xác):***

Định nghĩa: Độ chính xác là tỷ lệ số dự đoán đúng so với tổng số dự đoán. Đây là một trong những độ đo đơn giản và phổ biến nhất để đánh giá mô hình phân loại.

Công thức tính:



Trong đó:

TP (True Positive): Số lượng dự đoán đúng là tích cực.

TN (True Negative): Số lượng dự đoán đúng là tiêu cực.

FP (False Positive): Số lượng dự đoán sai là tích cực.

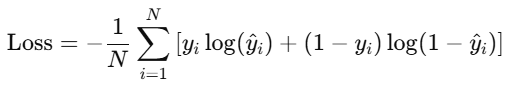
FN (False Negative): Số lượng dự đoán sai là tiêu cực.

Lý do sử dụng: Accuracy là độ đo đơn giản và dễ hiểu. Nó cho phép chúng ta nhanh chóng xác định mức độ chính xác của mô hình.

***Loss(độ mất mát):***

Định nghĩa: Độ Mất Mát (Loss) là một chỉ số dùng để đánh giá hiệu suất của một mô hình trong học máy, đặc biệt trong các bài toán phân loại và hồi quy. Độ mất mát cho biết mức độ sai lệch giữa giá trị dự đoán của mô hình và giá trị thực tế. Một giá trị mất mát thấp cho thấy mô hình hoạt động tốt hơn trong việc dự đoán đúng kết quả.

Công thức:



Trong đó:

* N là số mẫu,
* yi​ là nhãn thực tế (0 hoặc 1),
* y^i​ là xác suất dự đoán của mẫu thứ i.

Lý do sử dụng:

Đánh Giá Hiệu Suất Mô Hình: Độ mất mát giúp xác định xem mô hình có học tốt hay không. Mô hình sẽ cố gắng tối thiểu hóa độ mất mát trong quá trình huấn luyện.

Hướng Dẫn Quá Trình Huấn Luyện: Độ mất mát được sử dụng để tính toán gradient trong thuật toán tối ưu (như Gradient Descent). Việc này cho phép mô hình cập nhật trọng số sao cho giảm độ mất mát.

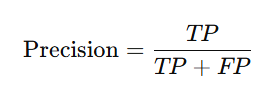
So Sánh Các Mô Hình Khác Nhau: Giúp dễ dàng so sánh hiệu suất của các mô hình khác nhau dựa trên độ mất mát. Mô hình có độ mất mát thấp hơn thường được chọn làm mô hình tốt hơn.

Phân Tích Các Vấn Đề Mô Hình: Độ mất mát có thể giúp phát hiện các vấn đề như overfitting (quá khớp) hoặc underfitting (thiếu khớp) trong mô hình.

***Precision (Độ chính xác):***

Định nghĩa: Precision đo lường tỷ lệ dự đoán đúng là tích cực trong tổng số dự đoán tích cực.

Công thức tính:

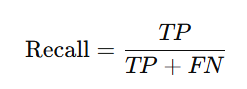


Lý do sử dụng: Precision hữu ích khi bạn muốn giảm thiểu số lượng sai lầm loại I (false positives). Ví dụ, trong trường hợp phát hiện bình luận tiêu cực, một phát hiện sai là tích cực có thể dẫn đến các phản hồi không mong muốn từ người dùng.

***Recall:***

Định nghĩa: Recall đo lường tỷ lệ dự đoán đúng là tích cực trong tổng số thực tế tích cực.

Công thức tính:

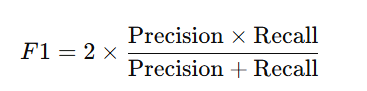


Lý do sử dụng: Recall quan trọng trong các trường hợp mà bạn muốn đảm bảo rằng tất cả các trường hợp tích cực được phát hiện. Ví dụ, trong phân loại cảm xúc, bạn không muốn bỏ lỡ bất kỳ bình luận tiêu cực nào.

***F1 Score:***

Định nghĩa: F1 Score là trung bình điều hòa giữa precision và recall. Nó cung cấp một cái nhìn tổng thể hơn về sự cân bằng giữa độ chính xác và độ nhắc lại.

Công thức:

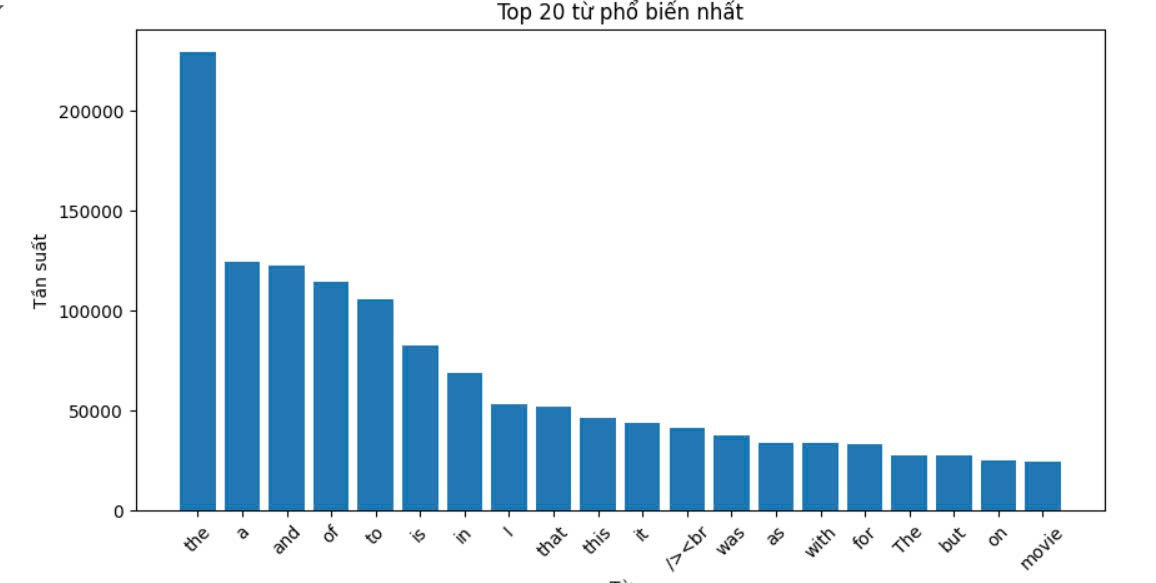


Lý do sử dụng: F1 Score là một độ đo quan trọng khi có sự không cân bằng giữa các lớp. Nó giúp đánh giá mô hình một cách công bằng hơn so với chỉ sử dụng độ chính xác đơn giản.

1. ĐÁNH GIÁ THỰC NGHIỆM

4.1. **kết quả thực nghiệm**

### ***4.1. Chuẩn hóa dữ liệu***



* + - 1. Lọc top 20 từ phổ biến nhất

***A graph of a person

Description automatically generated with medium confidence***

* + - 1. Phân bố độ dài văn bản

***A red and green circle with white text

Description automatically generated***

* + - 1. Tỷ lệ nhãn trong dữ liệu huấn luyện

***A diagram of a graph

Description automatically generated with medium confidence***

* + - 1. Dữ liệu sau khi giảm chiều với PCA

*4.2. Kết quả dự đoán mô hình Logistic*

A graph with a line

Description automatically generated

* + - 1. Biểu đồ hàm Loss mô hình Logistic

A graph showing a curve

Description automatically generated with medium confidence

* + - 1. Biểu đồ Accuracy mô hình Logistic

A blue squares with white text

Description automatically generated

* + - 1. Ma trận nhầm lẫn mô hình Logistic

*4.3. Kết quả dự đoán mô hình Naive Bayes*

A diagram of a positive and negative

Description automatically generated with medium confidence

* + - 1. Ma trận nhầm lẫn mô hình Naive Bayes

**4.4. Kết quả sau khi Training**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy | Precision | Recall | F1 score |
| Logistic | 0.8150 | 0.7931 | 0.8553 | 0.8230 |
| Naive Bayes | 0.8454 | 0.8448 | 0.8448 | 0.8467 |

* + - * 1. Bảng so sánh kết quả sau khi tranning 2 mô hình Naive Bayes và Logistics

**4.5. Triển khai website nhận diện bình luận tiêu cực bằng Naive Bayes**

**A screenshot of a computer

Description automatically generated**

* + - 1. Hình ảnh website nhận diện bình luận tiêu cực bằng Naive Bayes

**KếT LUẬN**

Đề tài đã triển khai thành công mô hình để nhận diện bình luận tiêu cực và tích hợp vào API phục vụ bài toán thực tế phân loại cảm xúc trong văn bản.

**Kết quả đạt được:**  
 Sau khi hoàn thành đề tài "Xây dựng và tích hợp API nhận diện bình luận tiêu cực", nhóm thực hiện đã đạt được nhiều cải thiện cả về kiến thức lẫn kỹ năng. Về mặt chuyên môn, nhóm đã nắm vững quy trình xây dựng và triển khai một mô hình machine learning, từ khâu thu thập, xử lý dữ liệu hình ảnh đến huấn luyện và đánh giá mô hình Mô hình đã đạt độ chính xác trung bình khoảng 84% trên tập kiểm thử. Với ưu điểm đơn giản và nhanh chóng, mô hình rất phù hợp để tích hợp trong các hệ thống yêu cầu thời gian xử lý nhanh như API thời gian thực. Độ đo F1-score được sử dụng để đánh giá hiệu quả phân loại của mô hình, trong đó F1-score cho lớp bình luận tiêu cực đạt 87% và cho lớp bình luận không tiêu cực là 83%.

**Hạn chế:**  
Dù đạt được hiệu quả tốt trên tập dữ liệu huấn luyện, mô hình vẫn bộc lộ một số hạn chế khi áp dụng trên tập dữ liệu thực tế:

1. Gặp khó khăn khi xử lý các câu bình luận mơ hồ, dài hoặc có chứa từ đồng nghĩa không xuất hiện trong tập huấn luyện.
2. Sai số tăng lên khi kiểm thử trên dữ liệu thực tế, thể hiện hiện tượng overfitting nhẹ.
3. Bộ dữ liệu đầu vào chủ yếu là văn bản ngắn, chưa đủ đa dạng về ngữ cảnh và biểu cảm.

**Nguyên nhân:**  
Các hạn chế trên có thể xuất phát từ một số nguyên nhân chính như sau:

* **Dữ liệu**: Bộ dữ liệu huấn luyện còn hạn chế về quy mô và tính đa dạng.
* **Tiền xử lý**: Các kỹ thuật tiền xử lý như loại bỏ từ dừng, chuẩn hóa từ đồng nghĩa, stemming hoặc lemmatization chưa được tối ưu.
* **Phân phối dữ liệu**: Tập huấn luyện và tập kiểm thử chưa cân đối, dẫn đến sai số tăng khi thử nghiệm.

**Định hướng cải tiến:**  
Để nâng cao hiệu quả mô hình, nhóm nghiên cứu có thể thực hiện các bước cải tiến như sau:

1. **Mở rộng dữ liệu**: Thu thập thêm bình luận từ nhiều nguồn khác nhau để tăng sự đa dạng về ngữ cảnh và biểu cảm.
2. **Cải tiến tiền xử lý**: Áp dụng các kỹ thuật nâng cao như stemming, lemmatization và mở rộng từ vựng thông qua embedding hoặc từ điển đồng nghĩa.
3. **Kết hợp mô hình nâng cao**: Sử dụng thêm các mô hình như SVM hoặc các mô hình học sâu (BERT) để xử lý các trường hợp phức tạp.
4. **Tối ưu API**: Đảm bảo API xử lý dữ liệu đầu vào linh hoạt, giảm thiểu thời gian phản hồi và tăng cường tính ứng dụng trong thực tế.

TÀI LIỆU THAM KHẢO

[1] Nada, AM Abu, et al. "Age and gender prediction and validation through single user images using CNN." Int. J. Acad. Eng. Res.(IJAER) 4 (2020): 21-24.

[2] Jana, Ranjan, Debaleena Datta, and Rituparna Saha. "Age estimation from face image using wrinkle features." Procedia Computer Science 46 (2015): 1754-1761.

[3] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.

[4] Antipov, Grigory, et al. "Effective training of convolutional neural networks for face-based gender and age prediction." Pattern Recognition 72 (2017): 15-26.

[5] Liu, Hao, et al. "Group-aware deep feature learning for facial age estimation." Pattern Recognition 66 (2017): 82-94.

[6] Webb, G. I., Keogh, E., & Miikkulainen, R. (2010). Naïve Bayes. *Encyclopedia of machine learning*, *15*(1), 713-714.