BỘ GIÁO DỤC ĐÀO TẠO

TRƯỜNG ĐẠI HỌC ĐẠI NAM

--- 🕮 ---

ĐỒ ÁN TỐT NGHIỆP

XÂY DỰNG ỨNG DỤNG PHÂN LOẠI TIẾNG VIỆT

SINH VIÊN THỰC HIỆN : Bùi Thị Nhiên

MÃ SINH VIÊN : 1351020080

KHOA : CÔNG NGHỆ THÔNG TIN

Hà Nội, Năm 2024

BỘ GIÁO DỤC ĐÀO TẠO

TRƯỜNG ĐẠI HỌC ĐẠI NAM

BÙI THỊ NHIÊN

XÂY DỰNG ỨNG DỤNG PHÂN LOẠI TIẾNG VIỆT

CHUYÊN NGÀNH : CÔNG NGHỆ THÔNG TIN

MÃ SỐ : 74.80.201

GIẢNG VIÊN HƯỚNG DẪN: TS. TRẦN ĐỨC MINH

Hà Nội, năm 2024

LỜI CAM ĐOAN

Em cam đoan rằng thành quả đạt được trong bài khóa luận này là kết quả của sự nỗ lực nghiên cứu và tìm hiểu chân thành từ phía em. Em đã dành thời gian và công sức lớn để thu thập dữ liệu, phân tích thông tin và trình bày các khái niệm, ý kiến và kết quả một cách cẩn thận và khách quan.

Bên cạnh việc trình bày quan điểm và công trình của riêng mình, bài khóa luận cũng dựa trên việc tham khảo và sử dụng tài liệu từ các nguồn có liên quan đến đề tài. Tất cả các nguồn tài liệu này đã được xác định và trích dẫn một cách rõ ràng và tuân thủ các quy tắc và quy định về bản quyền và tài liệu tham khảo.

Em cam kết hoàn toàn chịu trách nhiệm về tính chính xác và đáng tin cậy của kết quả nghiên cứu và tài liệu trong bài khóa luận. Em hiểu rõ rằng em phải tuân thủ tất cả các quy định, quy tắc và quy chế kỷ luật của nhà trường liên quan đến thực hiện và trình bày bài khóa luận này.

Em đồng ý chấp nhận mọi hình phạt kỷ luật, hậu quả và trách nhiệm pháp lý nếu phát hiện bất kỳ vi phạm hoặc vi phạm bản quyền nào trong quá trình thực hiện bài khóa luận này. Em sẽ hoàn toàn hợp tác với các cơ quan quản lý của trường và người hướng dẫn để giải quyết mọi vấn đề phát sinh.

Em hy vọng rằng bài khóa luận này đáp ứng được các tiêu chí và yêu cầu của nhà trường và góp phần vào lĩnh vực nghiên cứu tương ứng.

Hà Nội, ngày 24 tháng 04 năm 2024

Sinh viên thực hiện

Bùi Thị Nhiên

LỜI CẢM ƠN

Trong quá trình thực hiện đồ án, với sự hỗ trợ và tạo điều kiện từ trường Đại Học Đại Nam, những đóng góp và gợi ý từ các bạn cùng với sự hướng dẫn tận tâm của giảng viên khoa Công Nghệ Thông Tin và người hướng dẫn chính là giảng viên Trần Thu Trang, em đã hoàn thành đề tài và báo cáo theo đúng thời gian quy định.

Dù với khả năng và thời gian có hạn, không tránh khỏi những thiếu sót, em mong nhận được sự quan tâm và giúp đỡ, tạo điều kiện từ thầy cô giáo để em hoàn thiện đề tài nghiên cứu.

Một lần nữa, em xin chân thành cảm ơn tất cả các thầy cô giáo trong trường Đại Học Đại Nam đã dạy dỗ và chỉ bảo chúng em trong suốt thời gian học. Đặc biệt, chúng em muốn gửi lời cảm ơn sâu sắc tới giảng viên Trần Đức Minh đã hướng dẫn em suốt quá trình làm báo cáo.

Em xin chân thành cảm ơn!

LỜI NÓI ĐẦU

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, việc ứng dụng các công nghệ mới vào cuộc sống ngày càng trở nên phổ biến và cần thiết. Một trong những lĩnh vực đang thu hút nhiều sự chú ý và nghiên cứu là xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP). NLP là một nhánh của trí tuệ nhân tạo (AI) tập trung vào việc tương tác giữa máy tính và ngôn ngữ tự nhiên của con người. Vì thế em đã chọn đề tài ***"Xây Dựng Ứng Dụng Phân Loại Tiếng Việt"*** làm đề tài nghiên cứu, với mục tiêu phát triển một hệ thống có khả năng tự động phân loại văn bản tiếng Việt.

Việt Nam là một quốc gia có dân số đông đảo với một ngôn ngữ phong phú và đa dạng. Tuy nhiên, so với tiếng Anh và các ngôn ngữ phổ biến khác, tiếng Việt chưa được nghiên cứu và ứng dụng rộng rãi trong lĩnh vực NLP. Việc xây dựng một ứng dụng phân loại văn bản tiếng Việt không chỉ đóng góp vào việc phát triển công nghệ mà còn mang lại nhiều lợi ích thiết thực cho xã hội, như hỗ trợ tìm kiếm thông tin, phân tích dư luận xã hội, và cải thiện chất lượng dịch vụ khách hàng.

Mục tiêu chính của đề tài này là xây dựng một ứng dụng phân loại văn bản tiếng Việt sử dụng các kỹ thuật học máy (machine learning). Ứng dụng sẽ có khả năng nhận diện và phân loại các đoạn văn bản vào các chủ đề cụ thể. Để đạt được mục tiêu này, nghiên cứu sẽ tập trung vào các khía cạnh sau: thu thập và xử lý dữ liệu, tạo ra một bộ dữ liệu huấn luyện và kiểm thử chất lượng cao từ các nguồn tin cậy; phân tích và tiền xử lý ngôn ngữ, sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên để chuẩn hóa và trích xuất đặc trưng từ văn bản; xây dựng mô hình học máy, lựa chọn và tối ưu hóa các thuật toán học máy phù hợp để phân loại văn bản; đánh giá và cải thiện mô hình, sử dụng các chỉ số đánh giá hiệu quả để đo lường và cải thiện độ chính xác của mô hình.

Để thực hiện đề tài này, nghiên cứu sẽ áp dụng một số phương pháp chính bao gồm: thu thập dữ liệu, sử dụng các nguồn dữ liệu trực tuyến như báo chí, diễn đàn, và mạng xã hội để xây dựng bộ dữ liệu đa dạng và phong phú; tiền xử lý dữ liệu, sử dụng các kỹ thuật như tách từ, loại bỏ từ dừng, và chuẩn hóa văn bản để chuẩn bị dữ liệu cho quá trình học máy; xây dựng mô hình học máy, áp dụng các thuật toán như Naive Bayes; đánh giá và tối ưu hóa, sử dụng các chỉ số như độ chính xác, độ nhạy, và độ đặc hiệu để đánh giá hiệu quả của mô hình, từ đó điều chỉnh và tối ưu hóa các tham số.

Đề tài *"**Xây Dựng Ứng Dụng Phân Loại Tiếng Việt***" không chỉ có ý nghĩa khoa học mà còn có ứng dụng thực tiễn cao. Kết quả của nghiên cứu sẽ góp phần nâng cao năng lực xử lý ngôn ngữ tự nhiên cho tiếng Việt, mở ra nhiều cơ hội mới trong các lĩnh vực như truyền thông, thương mại điện tử, và dịch vụ khách hàng. Với những tiến bộ trong công nghệ học máy và NLP, tương lai của việc ứng dụng các hệ thống phân loại văn bản tiếng Việt hứa hẹn sẽ đem lại nhiều đột phá và tiện ích cho xã hội.

NHẬN XÉT

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ký và ghi họ tên

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Chú thích
API	**Giao diện chương trình ứng dụng.**
BERT	**Bidirectional Encoder Representation from Transformer**
CORS	**Cross Origin Resource Sharing**
CSS	**Cascading Style Sheets**
GPT	**Generative Pre-trained Transformer**
HTTP	**Hyper Text Transfer Protocol**
HTML	**HyperText Markup Language**
NLP	**Ngôn ngữ tự nhiên**
OCR	**Optical Character Recognition**
SVM	**Support Vector Machines**
Transformer	**Bộ chuyển đổi**
TF-IDF	**term frequency – inverse document frequency**

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

1.1. Lý do chọn đề tài

1.1.1. Bối cảnh chung về công nghệ

Trong thời đại cách mạng công nghiệp 4.0, công nghệ thông tin và truyền thông đã phát triển vượt bậc, tạo ra một lượng dữ liệu khổng lồ hàng ngày. Việc xử lý và phân loại dữ liệu, đặc biệt là dữ liệu văn bản, trở nên vô cùng quan trọng trong nhiều lĩnh vực như truyền thông, thương mại điện tử, quản lý dữ liệu và nghiên cứu khoa học. Công nghệ xử lý ngôn ngữ tự nhiên (NLP) và học máy (Machine Learning) là hai lĩnh vực đang được chú trọng phát triển để giải quyết các vấn đề này.

Các công nghệ tiên tiến như học sâu (Deep Learning) với các mô hình nổi bật như BERT, GPT-3 đã cho thấy khả năng xử lý ngôn ngữ tự nhiên vượt trội, đặc biệt là trong các ngôn ngữ phổ biến như tiếng Anh. Tuy nhiên, đối với tiếng Việt, các nghiên cứu và ứng dụng vẫn đang ở giai đoạn phát triển và gặp nhiều thách thức như đặc thù ngôn ngữ, sự thiếu hụt dữ liệu chuẩn và tài nguyên tính toán.

1.1.2. Lý do chọn mô hình Naive Bayes để phân loại tiếng Việt

Mặc dù có nhiều mô hình phức tạp và hiện đại có thể áp dụng cho việc phân loại văn bản, nhưng em vẫn chọn mô hình Naive Bayes dùng để phân loại cho đề tài của em vì:

- **Đơn giản và hiệu quả**: Naive Bayes là một thuật toán đơn giản, dễ hiểu và dễ triển khai. Nó không yêu cầu nhiều tài nguyên tính toán, phù hợp với các hệ thống có hạn chế về phần cứng và thời gian xử lý.
- **Hiệu suất tốt trên các bài toán phân loại văn bản**: Naive Bayes đã được chứng minh là hiệu quả trong các bài toán phân loại văn bản, đặc biệt là khi dữ liệu không quá phức tạp và có số lượng từ vựng lớn. Đối với các bài toán như phân loại email spam, phân loại tin tức, thuật toán này vẫn cho kết quả tốt.
- **Phù hợp với ngôn ngữ tiếng Việt**: Với đặc thù ngôn ngữ của tiếng Việt, bao gồm cú pháp phức tạp và sự phong phú về từ vựng, việc sử dụng Naive Bayes có thể giảm

bớt một số khó khăn so với các mô hình phức tạp khác. Giả định độc lập giữa các từ của Naive Bayes tuy không hoàn toàn chính xác nhưng lại giúp đơn giản hóa bài toán và đạt được độ chính xác chấp nhận được.

– **Khả năng mở rộng và tích hợp**: Naive Bayes dễ dàng tích hợp vào các hệ thống hiện có và có khả năng mở rộng khi cần thiết. Điều này giúp dễ dàng triển khai và sử dụng trong các ứng dụng thực tế như hệ thống quản lý tài liệu, hệ thống tìm kiếm thông tin và các dịch vụ phân tích dữ liệu.

1.2. Tình hình nghiên cứu

Hiện nay, có nhiều nghiên cứu về phân loại văn bản sử dụng các phương pháp khác nhau như SVM, k-NN, và đặc biệt là các mô hình học sâu như RNN, CNN và Transformer. Tuy nhiên, Naive Bayes vẫn là một phương pháp được ưa chuộng do tính đơn giản, hiệu quả và khả năng xử lý tốt với các tập dữ liệu nhỏ và trung bình. Đối với tiếng Việt, đã có một số nghiên cứu về phân loại văn bản nhưng chưa có nhiều ứng dụng cụ thể và thực tế. Đề tài này sẽ tiếp tục phát triển và ứng dụng các nghiên cứu đó vào việc xây dựng một ứng dụng thực tiễn, nhằm kiểm chứng và nâng cao hiệu quả của phương pháp này đối với tiếng Việt.

1.3. Đối tượng và phạm vi nghiên cứu

1.3.1. Đối tượng nghiên cứu:

Thuật toán Naive Bayes và các kỹ thuật xử lý ngôn ngữ tự nhiên.

1.3.2. Phạm vi nghiên cứu:

– Nghiên cứu các khái niệm và kỹ thuật học máy cơ bản, bao gồm các loại học máy, thuật toán và ứng dụng.

– Phân tích và xử lý dữ liệu tiếng Việt để chuẩn bị cho việc phân loại.

– Đánh giá và lựa chọn các mô hình học máy phù hợp cho nhiệm vụ phân loại tiếng Việt, Naive Bayes

– Tối ưu hóa các mô hình học máy đã chọn để đạt được độ chính xác và hiệu suất cao nhất.

– Xây dựng một API back-end bằng ngôn ngữ Python để cung cấp dịch vụ phân loại tiếng Việt.

– Phát triển một ứng dụng dựa trên API back-end để người dùng có thể sử dụng dịch vụ phân loại tiếng Việt một cách dễ dàng.

1.4. Mục đích và nhiệm vụ nghiên cứu nghiên cứu

1.4.1. Mục đích nghiên cứu:

Mục đích của nghiên cứu đề tài này là xây dựng và tối ưu hóa các mô hình học máy để phân loại văn bản tiếng Việt. Qua việc áp dụng mô hình Naive Bayes, mục tiêu là tạo ra một ứng dụng hoặc hệ thống có khả năng tự động phân loại văn bản một cách hiệu quả và chính xác. Đồng thời, nghiên cứu cũng nhằm vào việc nâng cao khả năng xử lý ngôn ngữ tự nhiên (NLP) trong tiếng Việt và đóng góp vào việc phát triển lĩnh vực này. Mục đích cuối cùng là áp dụng các kết quả và kiến thức thu được vào thực tế, giúp cải thiện quá trình quản lý và xử lý dữ liệu văn bản trong nhiều lĩnh vực như tin tức, email, phân loại tài liệu và các ứng dụng khác.

1.4.2. Nhiệm vụ nghiên cứu:

– Tìm hiểu lý thuyết về thuật toán Naive Bayes và các kỹ thuật xử lý ngôn ngữ tự nhiên.

– Thu thập và tiền xử lý dữ liệu văn bản tiếng Việt.

– Xây dựng và triển khai mô hình Naive Bayes.

– Đánh giá hiệu quả của mô hình và cải tiến nếu cần thiết.

– Phát triển ứng dụng thực tế và kiểm thử trên các tập dữ liệu khác nhau.

1.5. Phương pháp nghiên cứu

– Thu thập dữ liệu: Bước đầu tiên là thu thập dữ liệu văn bản tiếng Việt từ các nguồn đáng tin cậy và đại diện. Điều này có thể bao gồm việc sử dụng các bộ dữ liệu có sẵn trên internet, nhưng cũng cần đảm bảo tính đa dạng và đại diện của dữ liệu.

– Tiền xử lý dữ liệu: Dữ liệu thu thập được sẽ được tiền xử lý để loại bỏ các yếu tố không cần thiết như stop words, dấu câu, và chuyển đổi các từ về dạng chuẩn hóa.

- Phân chia dữ liệu huấn luyện và kiểm tra: Dữ liệu sẽ được chia thành hai phần: một phần dành cho việc huấn luyện mô hình và một phần dành cho việc kiểm tra và đánh giá hiệu suất của mô hình.
- Xây dựng các mô hình học máy Naive Bayes và huấn luyện trên dữ liệu huấn luyện.
- Tối ưu hóa mô hình: Mô hình sẽ được tối ưu hóa thông qua việc điều chỉnh các tham số và siêu tham số của từng mô hình, sử dụng các kỹ thuật như lưới tìm kiếm (grid search) hoặc tối ưu hóa thông qua thuật toán tối ưu hóa.
- Đánh giá hiệu suất của mô hình: Mô hình sau khi được tối ưu hóa sẽ được đánh giá hiệu suất trên tập dữ liệu kiểm tra, bằng cách đo lường các độ đo như độ chính xác, độ phủ, và độ đo F1-score.
- So sánh và phân tích kết quả: Kết quả từ các mô hình sẽ được so sánh và phân tích để đánh giá ưu và nhược điểm của từng phương pháp, cũng như đưa ra những kết luận và khuyến nghị cho việc triển khai trong thực tế.
- Triển khia ứng dụng:
    + Sử dụng Flask để xây dựng giao diện người dùng và triển khai ứng dụng phân loại văn bản trên web.
    + Sử dụng HTML, CSS và JavaScript để tạo giao diện người dùng thân thiện và tương tác.
    + Kiểm thử và tối ưu hóa:
    + Sử dụng pytest hoặc unittest để thực hiện kiểm thử đơn vị và đảm bảo tính đúng đắn của mã nguồn.
    + Sử dụng kỹ thuật tối ưu hóa mã nguồn và mô hình để cải thiện hiệu suất và tốc độ của ứng dụng.

1.6. Đóng góp đề tài

Đề tài "Xây dựng ứng dụng phân loại tiếng Việt dùng thuật toán Naive Bayes" không chỉ mang lại giá trị thực tiễn mà còn đóng góp vào kho tàng nghiên cứu về xử lý ngôn ngữ tự nhiên cho tiếng Việt. Qua quá trình nghiên cứu và triển khai, đề tài sẽ cung

cấp một công cụ hữu ích, hỗ trợ đắc lực cho việc xử lý và phân tích văn bản tiếng Việt, đồng thời mở ra hướng đi mới cho các nghiên cứu tiếp theo.

1.7. Bố cục của đề tài

– Đề tài được chia thành các phần:

+ Chương 1: Tổng quan về đề tài
+ Chương 2: Cơ sở lí thuyết
+ Chương 3: Phân tích và xử lí dữ liệu
+ Kết luận:
+ Danh mục tài liệu tham khảo

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Tổng quan về phân loại văn bản tiếng Việt

2.1.1. Khái niệm phân loại văn bản tiếng Việt

Phân loại văn bản tiếng Việt là một nhiệm vụ trong Xử lý ngôn ngữ tự nhiên (NLP) nhằm tự động phân chia các văn bản tiếng Việt thành các nhóm được xác định trước dựa trên nội dung và đặc điểm của chúng. Quá trình này sử dụng các kỹ thuật học máy và trí tuệ nhân tạo để phân tích văn bản và xác định các chủ đề, thể loại hoặc ý định chính của nó.

Phân loại văn bản (Text Classification) là bài toán thuộc nhóm học có giám sát (Supervised learning) trong học máy. Bài toán này yêu cầu dữ liệu cần có nhãn (label). Mô hình sẽ học từ dữ liệu có nhãn đó, sau đó được dùng để dự đoán nhãn cho các dữ liệu mới mà mô hình chưa gặp.

2.1.2. Ứng dụng của phân loại văn bản tiếng Việt

Phân loại văn bản tiếng Việt có nhiều ứng dụng quan trọng trong nhiều lĩnh vực khác nhau. Dưới đây là một số ứng dụng phổ biến của việc phân loại văn bản tiếng Việt:

Tìm kiếm thông tin: Phân loại văn bản giúp tổ chức và tìm kiếm thông tin một cách hiệu quả trên các nền tảng tìm kiếm và thư viện số. Việc này giúp người dùng dễ dàng truy cập thông tin cần thiết dựa trên chủ đề, mục đích, hoặc ngữ cảnh.

Phân tích ý kiến và cảm xúc: Phân loại văn bản có thể được sử dụng để phân tích ý kiến và cảm xúc trong các bình luận trên mạng xã hội, diễn đàn, hoặc các bài đánh giá sản phẩm. Điều này giúp doanh nghiệp và tổ chức hiểu rõ hơn về ý kiến của khách hàng và phản hồi từ cộng đồng.

Tóm tắt văn bản: Phân loại văn bản có thể được sử dụng để tạo ra tóm tắt tự động cho các văn bản dài. Điều này hữu ích trong việc tiết kiệm thời gian đọc và hiểu nội dung của văn bản, đặc biệt là khi phải xử lý một lượng lớn các tài liệu.

Phân loại thư rác và lọc email: Trong email và các hệ thống tin nhắn, phân loại văn bản giúp tự động phân loại thư rác và lọc email dựa trên nội dung và ngữ cảnh của các email, giúp người dùng tiết kiệm thời gian và tăng hiệu suất làm việc.

Tư vấn và đề xuất sản phẩm: Trong lĩnh vực thương mại điện tử, phân loại văn bản có thể được sử dụng để phân tích hành vi mua hàng của khách hàng và đề xuất sản phẩm phù hợp dựa trên sở thích và nhu cầu của họ.

Phân loại tin tức và bài báo: Phân loại văn bản cũng được sử dụng để tổ chức và phân loại các tin tức và bài báo trên các trang web tin tức và các nền tảng truyền thông trực tuyến khác.

2.1.3. Các thách thức trong phân loại văn bản tiếng Việt

Trong quá trình phân loại văn bản tiếng Việt, phải đối mặt với nhiều thách thức đặc biệt. Một số thách thức quan trọng bao gồm từ vựng đa nghĩa, sự phong phú của từ ngữ và biến thể ngôn ngữ, thiếu dữ liệu lớn và đa dạng, đối mặt với văn bản ngắn và không chuẩn, khả năng xử lý tiếng Việt của các công cụ và thư viện, cũng như sự phân cấp và cấu trúc ngôn ngữ phức tạp của tiếng Việt.

Trong tiếng Việt, từ vựng thường có nhiều ý nghĩa khác nhau, làm cho việc xác định ý nghĩa của một câu hoặc đoạn văn trở nên khó khăn. Đồng thời, sự phong phú của từ ngữ và biến thể ngôn ngữ (như từ ngữ địa phương, từ lóng) cũng làm cho quá trình xử lý văn bản trở nên phức tạp hơn.

Thêm vào đó, việc thu thập dữ liệu đủ lớn và đa dạng để huấn luyện các mô hình phân loại là một thách thức khác, đặc biệt là khi nguồn dữ liệu phong phú không có sẵn. Trong các mạng xã hội và các ứng dụng trực tuyến, văn bản thường ngắn và không chuẩn, làm tăng khó khăn trong việc trích xuất đặc trưng và phân loại.

Ngoài ra, khả năng của các công cụ và thư viện xử lý ngôn ngữ tự nhiên trong việc xử lý tiếng Việt cũng có thể gặp hạn chế so với các ngôn ngữ khác. Và cuối cùng, cấu trúc ngôn ngữ phân cấp và phức tạp của tiếng Việt cũng làm tăng độ phức tạp trong việc xử lý và phân loại văn bản

2.2. Các phương pháp phân loại văn bản tiếng Việt

2.2.1. Phân loại dựa trên từ khóa

- Nguyên lý:

Phương pháp này sử dụng các từ khóa cụ thể xuất hiện trong văn bản để xác định chủ đề hoặc thể loại của văn bản. Từ khóa có thể được xác định trước dựa trên kiến thức chuyên môn hoặc được rút trích từ các tập dữ liệu mẫu.

Trong một tập hợp các bài báo, từ khóa như "giáo dục", "học sinh", "trường học" có thể được sử dụng để phân loại các bài viết thuộc chủ đề giáo dục.

- Ưu điểm

Đơn giản và dễ hiểu: Phương pháp này dễ triển khai và không đòi hỏi nhiều về kiến thức kỹ thuật.

Hiệu quả trong các ngữ cảnh cụ thể: Khi các từ khóa được lựa chọn cẩn thận, phương pháp này có thể rất hiệu quả trong các ngữ cảnh cụ thể.

- Nhược điểm:

Khó xử lý các văn bản phức tạp: Các văn bản có cấu trúc phức tạp, chứa nhiều nghĩa và đa ngữ cảnh thường khó phân loại chính xác chỉ dựa vào từ khóa.

Không linh hoạt: Khi chủ đề hoặc từ khóa thay đổi, hệ thống cần được cập nhật thủ công.

2.2.2. Phân loại dựa trên học máy

- Nguyên lý:

Phương pháp này sử dụng các thuật toán học máy để phân tích văn bản và học các mẫu từ dữ liệu huấn luyện để thực hiện phân loại. Các thuật toán phổ biến bao gồm Naive Bayes, Support Vector Machines (SVM), và Decision Trees.

Sử dụng thuật toán Naive Bayes để phân loại email thành thư rác hoặc không phải thư rác dựa trên các đặc trưng của nội dung email.

- Ưu điểm:

Hiệu quả cao: Có khả năng xử lý các văn bản phức tạp và đa ngữ cảnh bằng cách học từ dữ liệu thực tế.

Tự động hóa: Hệ thống có thể tự động cải thiện khi được cung cấp thêm dữ liệu huấn luyện.

- Nhược điểm:

Cần nhiều dữ liệu huấn luyện: Để đạt hiệu quả cao, hệ thống cần được huấn luyện trên một lượng lớn dữ liệu.

Đòi hỏi kiến thức kỹ thuật: Việc triển khai và tối ưu hóa các thuật toán học máy đòi hỏi kiến thức sâu về lĩnh vực này.

2.2.3. Phân loại dựa trên mô hình ngôn ngữ

- Nguyên lý:

Sử dụng các mô hình ngôn ngữ, bao gồm mô hình ngôn ngữ thống kê (như n-gram models) và mô hình ngôn ngữ thần kinh (như Transformer, BERT, GPT) để phân tích ngữ nghĩa và ngữ cảnh của văn bản nhằm thực hiện phân loại.

Ví dụ:

Sử dụng mô hình BERT để phân loại bài viết thành các thể loại như thể thao, chính trị, giải trí dựa trên nội dung và ngữ cảnh của văn bản.

- Ưu điểm:

Hiểu ngữ nghĩa và ngữ cảnh: Mô hình ngôn ngữ có khả năng hiểu rõ hơn về ý nghĩa và ngữ cảnh của văn bản, từ đó cải thiện độ chính xác của việc phân loại.

Linh hoạt: Có thể áp dụng cho nhiều loại văn bản và chủ đề khác nhau mà không cần nhiều thay đổi về cấu trúc mô hình.

- Nhược điểm:

Cần nhiều dữ liệu huấn luyện: Để mô hình ngôn ngữ hoạt động hiệu quả, cần một lượng lớn dữ liệu huấn luyện chất lượng cao.

Đòi hỏi kiến thức chuyên môn: Việc triển khai và điều chỉnh các mô hình ngôn ngữ đòi hỏi kiến thức sâu rộng về lĩnh vực học máy và xử lý ngôn ngữ tự nhiên.

2.3. Học máy

2.3.1. Khái niệm về học máy

Học máy là một nhánh của trí tuệ nhân tạo, nó là một lĩnh vực nghiên cứu cho phép máy tính có khả năng cải thiện chính bản thân chúng dựa trên dữ liệu mẫu (training data) hoặc dựa vào kinh nghiệm (những gì đã được học). Machine learning có thể tự dự đoán hoặc đưa ra quyết định mà không cần được lập trình cụ thể.

2.3.2. Phân loại học máy

Học máy có thể được phân loại dựa trên hai tiêu chí chính: loại bài toán và phương pháp học.

2.3.2.1 Phân loại thuật toán học máy dựa trên bài toán và nhiệm vụ cần giải quyết

Học có giám sát (Supervised Learning): Đây là phương pháp học máy trong đó mô hình được huấn luyện trên một tập dữ liệu có gán nhãn trước. Trong bài toán phân loại văn bản tiếng Việt, học có giám sát là phương pháp phổ biến nhất. Thuật toán Naive Bayes là một ví dụ điển hình của học có giám sát, nơi mô hình học từ các văn bản đã được phân loại trước đó để dự đoán nhãn của các văn bản mới.

Học không giám sát (Unsupervised Learning): Phương pháp này sử dụng dữ liệu không có gán nhãn. Các thuật toán học không giám sát tìm kiếm cấu trúc ẩn hoặc phân cụm dữ liệu mà không cần thông tin nhãn trước. Phân cụm văn bản là một ví dụ về học không giám sát, tuy nhiên, nó ít được sử dụng trong bài toán phân loại cụ thể.

Học bán giám sát (Semi-Supervised Learning): Kết hợp cả dữ liệu có nhãn và không có nhãn để cải thiện độ chính xác của mô hình. Điều này hữu ích khi việc gán nhãn dữ liệu tốn nhiều thời gian và chi phí.

Học tăng cường/củng cố (Reinforcement Learning): Học tăng cường là phương pháp trong đó một tác nhân (agent) học cách thực hiện các hành động trong một môi trường để tối ưu hóa một phần thưởng (reward) chung. Thuật toán sẽ tự học dựa trên việc tính điểm thưởng và phạt cho các kết quả thực hiện nhiệm vụ.

AlphaGo: chương trình máy tính này học cách chơi và thắng trong các trò chơi như cờ vây và cờ vua thông qua việc tối ưu hóa điểm thưởng từ các ván chơi.

Tự động hóa, robot, quản lý chuỗi cung ứng, trò chơi, hệ thống đề xuất, v.v.

2.3.2.2 Phân loại thuật toán học máy dựa trên cách máy tính học

Học dựa trên ví dụ (Instance-based Learning): Các thuật toán như K-nearest neighbors (KNN) lưu trữ các ví dụ của dữ liệu huấn luyện và đưa ra dự đoán dựa trên sự tương đồng với các ví dụ đã biết. Phương pháp này thường yêu cầu nhiều bộ nhớ và tính toán, không phù hợp cho các tập dữ liệu lớn.

Học dựa trên mô hình (Model-based Learning): Các thuật toán như Naive Bayes, Logistic Regression, và Support Vector Machines (SVM) xây dựng một mô hình toán học từ dữ liệu huấn luyện để đưa ra dự đoán. Phương pháp này thường hiệu quả hơn về mặt tính toán và có thể mở rộng cho các tập dữ liệu lớn.

Học sâu (Deep Learning): Sử dụng các mạng nơ-ron nhiều lớp để học các đặc trưng phức tạp từ dữ liệu. Các mô hình học sâu như LSTM và Transformer đã đạt được thành công lớn trong xử lý ngôn ngữ tự nhiên, nhưng yêu cầu tài nguyên tính toán lớn và dữ liệu huấn luyện khổng lồ.

2.3.3. Quy trình của học máy phân loại văn bản tiếng Việt

Quy trình xây dựng mô hình học máy để phân loại văn bản tiếng Việt bao gồm các bước sau:

1. Thu thập dữ liệu: Thu thập một lượng lớn văn bản tiếng Việt từ các nguồn như báo chí, blog, và mạng xã hội. Dữ liệu này cần được đa dạng và đại diện cho nhiều chủ đề khác nhau.
2. Tiền xử lý dữ liệu: Bao gồm các bước làm sạch dữ liệu, chuyển đổi văn bản thành dạng có thể xử lý được bởi máy tính. Các bước tiền xử lý bao gồm:
   + Loại bỏ các ký tự đặc biệt và dấu câu.
   + Chuyển đổi tất cả các từ thành chữ thường.
   + Tách từ (tokenization).
   + Loại bỏ các từ dừng (stopwords).
3. Chia dữ liệu: Dữ liệu được chia thành hai tập: tập huấn luyện (training set) và tập kiểm tra (test set). Tập huấn luyện được sử dụng để huấn luyện mô hình, trong khi tập kiểm tra được sử dụng để đánh giá hiệu quả của mô hình.

4. Xây dựng mô hình: Sử dụng thuật toán Naive Bayes để huấn luyện mô hình trên tập huấn luyện. Naive Bayes dựa trên định lý Bayes và giả định rằng các đặc trưng (từ) trong văn bản là độc lập với nhau.
5. Đánh giá mô hình: Đánh giá mô hình trên tập kiểm tra bằng các chỉ số như độ chính xác (accuracy), độ nhạy (recall), độ đặc hiệu (precision), và điểm F1 (F1 score).
6. Triển khai mô hình: Sau khi đạt được hiệu quả mong muốn, mô hình được triển khai vào hệ thống thực tế để phân loại các văn bản tiếng Việt mới.

2.3.4. Ứng dụng của học máy:

Học máy đã và đang được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau. Dưới đây là một số ví dụ cụ thể về các ứng dụng phổ biến của học máy:

- Xử lý ảnh

Xử lý ảnh (Image Processing) giải quyết các vấn đề liên quan đến phân tích thông tin từ hình ảnh hoặc thực hiện một số phép biến đổi trên hình ảnh. Một số ví dụ bao gồm:

- Gắn thẻ hình ảnh (Image Tagging):

Facebook sử dụng thuật toán tự động phát hiện và gắn thẻ khuôn mặt của bạn và bạn bè trong các bức ảnh. Thuật toán này học từ những bức ảnh mà bạn đã tự gắn thẻ trước đó.

- Nhận dạng ký tự (Optical Character Recognition - OCR):

OCR chuyển dữ liệu trên giấy tờ, văn bản thành dữ liệu số hóa. Thuật toán học cách nhận biết các ký tự trong ảnh chụp và chuyển chúng thành văn bản.

- Ô tô tự lái (Self-driving cars):

Một phần của công nghệ ô tô tự lái sử dụng xử lý ảnh. Thuật toán học máy giúp phát hiện các mép đường, biển báo, và chướng ngại vật bằng cách phân tích các khung hình video từ camera.

- Phân tích văn bản

Phân tích văn bản (Text Analysis) liên quan đến việc trích xuất hoặc phân loại thông tin từ văn bản. Các văn bản có thể bao gồm các bài đăng trên mạng xã hội, email, đoạn chat, và tài liệu. Một số ví dụ phổ biến là:

- Lọc spam (Spam filtering):

Phân loại email để xác định xem nó có phải là thư rác hay không dựa trên nội dung và tiêu đề email.

- Phân tích ngữ nghĩa (Sentiment Analysis):

Phân loại ý kiến là tích cực, trung tính hay tiêu cực dựa trên nội dung văn bản của người viết.

- Khai thác thông tin (Information Extraction):

Trích xuất các thông tin hữu ích từ văn bản, chẳng hạn như địa chỉ, tên người, từ khóa.

- Khai phá dữ liệu

Khai phá dữ liệu (Data Mining) là quá trình khám phá ra các thông tin có giá trị hoặc đưa ra các dự đoán từ dữ liệu. Đây là một lĩnh vực bao quát với nhiều ứng dụng cụ thể, bao gồm:

- Phát hiện bất thường (Anomaly Detection):

Phát hiện các ngoại lệ, ví dụ như phát hiện gian lận thẻ tín dụng. Thuật toán có thể phát hiện một giao dịch khả nghi dựa trên các giao dịch thông thường của người dùng đó.

- Phát hiện các quy luật (Association Rules):

Trong các siêu thị hoặc trang thương mại điện tử, học máy có thể khám phá ra các mặt hàng thường được mua cùng nhau. Ví dụ, nếu khách hàng mua món hàng A, họ thường mua kèm món hàng B. Thông tin này rất hữu ích cho việc tiếp thị sản phẩm.

- Các ứng dụng khác

Ngoài các ứng dụng trên, học máy còn được sử dụng rộng rãi trong nhiều lĩnh vực khác:

Nhận diện giọng nói (Speech Recognition):

Sử dụng trong các trợ lý ảo như Siri hay Google Assistant để tìm kiếm bằng giọng nói.

Hệ thống khuyến nghị (Recommendation Systems):

Khi bạn tìm kiếm một sản phẩm trên Google, các quảng cáo liên quan đến sản phẩm đó sẽ xuất hiện trên các nền tảng xã hội mà bạn sử dụng.

- Dịch thuật (Translation):

Các công cụ dịch thuật như Google Translate sử dụng học máy để dịch văn bản giữa các ngôn ngữ.

- Lọc email:

Phân loại email, phát hiện và lọc các email spam hoặc không quan trọng.

- Xây dựng xe tự hành (Autonomous Vehicles):

Học máy giúp các phương tiện tự hành phát hiện và phản ứng với các điều kiện giao thông và môi trường xung quanh.

Học máy đang ngày càng trở nên quan trọng và không thể thiếu trong nhiều lĩnh vực công nghệ hiện đại, đóng góp vào sự phát triển và tối ưu hóa nhiều dịch vụ và sản phẩm trong cuộc sống hàng ngày.

2.3.5. Lý do chọn thuật toán Naive Bayes

Thuật toán Naive Bayes được chọn cho bài toán phân loại văn bản tiếng Việt vì những lý do sau:

Đơn giản và hiệu quả: Naive Bayes là một trong những thuật toán học máy đơn giản và dễ triển khai. Nó yêu cầu ít tài nguyên tính toán và thời gian huấn luyện nhanh chóng, phù hợp với các bài toán phân loại văn bản có dữ liệu lớn.

Hiệu suất tốt trên các bài toán văn bản: Naive Bayes thường cho kết quả tốt trong các bài toán phân loại văn bản, đặc biệt là với các tập dữ liệu có kích thước lớn và nhiều đặc trưng. Điều này là do giả định độc lập giữa các đặc trưng, mặc dù đơn giản, lại phù hợp với tính chất của dữ liệu văn bản.

Khả năng mở rộng: Naive Bayes dễ dàng mở rộng cho các bài toán đa lớp, tức là có thể phân loại văn bản thành nhiều nhãn khác nhau mà không cần thay đổi cấu trúc thuật toán.

Xử lý tốt với dữ liệu nhiễu: Naive Bayes có khả năng xử lý dữ liệu nhiễu và không yêu cầu nhiều tinh chỉnh. Điều này rất hữu ích khi làm việc với dữ liệu thực tế, nơi mà chất lượng dữ liệu có thể không đồng nhất.

Dễ hiểu và giải thích: Kết quả của Naive Bayes dễ dàng giải thích, giúp cho việc phân tích và cải thiện mô hình trở nên đơn giản hơn. Điều này đặc biệt quan trọng khi làm việc trong các dự án cần sự minh bạch và dễ hiểu cho người dùng cuối.

So với các thuật toán khác như SVM, KNN hay các mô hình học sâu phức tạp, Naive Bayes cung cấp một giải pháp đơn giản nhưng hiệu quả cho bài toán phân loại văn bản tiếng Việt, đáp ứng tốt các yêu cầu về tốc độ, độ chính xác và khả năng triển khai.

2.4. Thuật toán Naive Baye s

Thuật toán Naive Bayes là một phương pháp phân loại trong Machine Learning dựa trên việc áp dụng định lý Bayes với giả định "ngây thơ" (naive) rằng các đặc trưng đầu vào độc lập với nhau. Mặc dù giả định này thường không hoàn toàn đúng trong thực tế, thuật toán này vẫn hoạt động hiệu quả trong nhiều bài toán thực tế.

2.4.1. Toán học về thuật toán Naive Bayes

Thuật toán Naive Bayes bao gồm các bước sau:

Xây dựng mô hình với dữ liệu huấn luyện.

Ước lượng xác suất có điều kiện của mỗi nhãn dựa trên dữ liệu huấn luyện và giả định độc lập.

Dự đoán nhãn mới cho các mẫu dữ liệu thử nghiệm bằng cách chọn nhãn có xác suất cao nhất.

Cụ thể, trong thuật toán Naive Bayes, ta tính xác suất của một lớp (class) cụ thể dựa trên các đặc trưng (features) của một mẫu dữ liệu. Công thức cơ bản của định lý Bayes được sử dụng như sau:

$$P(Y|X)=P(X|Y)\cdot P(Y)P(X)\text{P(Y|X)=P(X)P(X|Y)}\cdot\text{P(Y)}$$

Trong đó:

YY là lớp cần dự đoán.

XX là tập hợp các đặc trưng của mẫu dữ liệu.

$P(Y|X)$ P(Y|X) là xác suất của lớp YY khi biết các đặc trưng XX.

$P(X|Y)$ P(X|Y) là xác suất của các đặc trưng XX khi biết lớp YY.

$P(Y)$P(Y) là xác suất tiên nghiệm của lớp YY.

$P(X)$P(X) là xác suất tiên nghiệm của các đặc trưng XX.

Với giả định "ngây thơ", ta giả định rằng các đặc trưng là độc lập với nhau, từ đó ta có thể viết lại $P(X|Y)$ P(X|Y) như một tích của các xác suất của từng đặc trưng:

$$P(X|Y) =\prod i=1nP(xi|Y)\text{ P(X|Y) =}\prod\text{i=1nP(xi|Y)}$$

Thuật toán Naive Bayes được ứng dụng phổ biến trong các bài toán phân loại văn bản, lọc thư rác (spam), phân loại chủ đề, và nhiều ứng dụng Machine Learning khác.

2.4.2. Điểm mạnh của thuật toán Naive Bayes:

Đơn giản và nhanh chóng: Naive Bayes là một trong những thuật toán học máy dễ hiểu và dễ triển khai nhất. Nó yêu cầu ít tài nguyên tính toán và có thể huấn luyện nhanh chóng trên các tập dữ liệu lớn.

Hiệu quả với các tập dữ liệu lớn: Nhờ vào tính đơn giản và khả năng mở rộng tốt, Naive Bayes có thể xử lý hiệu quả các tập dữ liệu lớn và nhiều đặc trưng.

Xử lý tốt với dữ liệu nhiễu: Naive Bayes có khả năng xử lý dữ liệu nhiễu tốt, do nó dựa trên các xác suất điều kiện độc lập, giúp giảm thiểu ảnh hưởng của nhiễu trong dữ liệu.

Khả năng mở rộng: Naive Bayes có thể dễ dàng mở rộng để phân loại đa lớp, tức là có thể phân loại văn bản thành nhiều nhãn khác nhau mà không cần thay đổi cấu trúc thuật toán.

Dễ dàng cập nhật mô hình: Khi có thêm dữ liệu mới, mô hình Naive Bayes có thể được cập nhật dễ dàng mà không cần phải huấn luyện lại từ đầu.

2.4.3. Điểm yếu của thuật toán Naive Bayes

Giả định độc lập giữa các đặc trưng: Giả định rằng các từ trong văn bản là độc lập với nhau không phải lúc nào cũng đúng. Trong thực tế, các từ thường có quan hệ với nhau và ảnh hưởng lẫn nhau, điều này có thể làm giảm độ chính xác của mô hình.

Không xử lý tốt với dữ liệu hiếm gặp: Naive Bayes có thể gặp khó khăn với các từ hiếm gặp trong tập huấn luyện, do xác suất của những từ này có thể rất nhỏ hoặc bằng không, dẫn đến vấn đề số học.

Hiệu quả kém với dữ liệu không cân bằng: Nếu tập dữ liệu huấn luyện không cân bằng, tức là một số lớp xuất hiện nhiều hơn hẳn so với các lớp khác, Naive Bayes có thể bị thiên vị về phía các lớp xuất hiện nhiều hơn.

2.4.4. Ứng dụng của thuật toán Naive Bayes trong xử lý ngôn ngữ tự nhiên

Thuật toán Naive Bayes được sử dụng rộng rãi trong xử lý ngôn ngữ tự nhiên (NLP) và các bài toán phân loại do tính hiệu quả và đơn giản của nó. Dưới đây là một số ứng dụng phổ biến:

- Lọc thư rác (Spam Filtering)

Mô tả: Naive Bayes là thuật toán phổ biến nhất trong việc lọc thư rác. Hệ thống sẽ học từ các email đã được gán nhãn là spam hoặc không phải spam để xác định các email mới có phải là spam không.

Cách hoạt động: Mỗi email được phân tích để tìm các từ khóa và dựa trên tần suất xuất hiện của chúng trong các email spam và không spam trước đó, hệ thống sẽ tính toán xác suất để xác định loại của email.

- Phân loại văn bản (Text Classification)

Mô tả: Naive Bayes thường được sử dụng để phân loại các tài liệu văn bản vào các danh mục khác nhau như chủ đề báo chí, phân loại blog, và phân loại sản phẩm.

Cách hoạt động: Thuật toán tính xác suất của mỗi danh mục dựa trên từ ngữ xuất hiện trong văn bản và chọn danh mục có xác suất cao nhất.

- Phân tích ngữ nghĩa (Sentiment Analysis)

Mô tả: Naive Bayes có thể được sử dụng để xác định cảm xúc (tích cực, tiêu cực, trung tính) của văn bản như đánh giá sản phẩm, nhận xét phim, và phản hồi khách hàng.

Cách hoạt động: Thuật toán học từ các văn bản đã được gán nhãn cảm xúc để dự đoán cảm xúc của các văn bản mới.

- Phân loại ngôn ngữ (Language Detection)

Mô tả: Naive Bayes cũng được áp dụng để phát hiện ngôn ngữ của một đoạn văn bản.

Cách hoạt động: Bằng cách sử dụng các mẫu từ và cấu trúc câu từ các ngôn ngữ khác nhau, thuật toán có thể dự đoán ngôn ngữ của văn bản mới.

Tiểu kết:

Chương 2 đã trình bày các khái niệm cơ bản và cơ sở lý thuyết liên quan đến phân loại văn bản tiếng Việt. Trước hết, em đã giới thiệu khái niệm, ứng dụng và các thách thức trong phân loại văn bản tiếng Việt. Tiếp theo, các phương pháp phân loại, bao gồm dựa trên từ khóa, học máy và mô hình ngôn ngữ, đã được phân tích chi tiết.

Phần tiếp theo tập trung vào học máy, bao gồm các khái niệm cơ bản, phân loại học máy theo bài toán và nhiệm vụ, cũng như phương pháp học. Quy trình học máy trong phân loại văn bản tiếng Việt được chi tiết hóa từ thu thập và tiền xử lý dữ liệu đến xây dựng, đánh giá và triển khai mô hình. Ngoài ra, em cũng nêu bật các ứng dụng của học máy trong phân loại văn bản và lý do chọn thuật toán Naive Bayes cho bài toán này.

Cuối cùng, chương này đi sâu vào thuật toán Naive Bayes, bao gồm nền tảng toán học, điểm mạnh và yếu của thuật toán, cùng những ứng dụng cụ thể trong xử lý ngôn ngữ tự nhiên. Naive Bayes đã chứng tỏ là một công cụ đơn giản và hiệu quả cho bài toán phân loại văn bản.

Chương 2 đã cung cấp nền tảng lý thuyết vững chắc, chuẩn bị cho việc áp dụng thực tế trong các chương tiếp theo.

CHƯƠNG 3. PHÂN TÍCH VÀ XỬ LÝ DỮ LIỆU

3.1. Nguồn gốc và phân tích dữ liệu

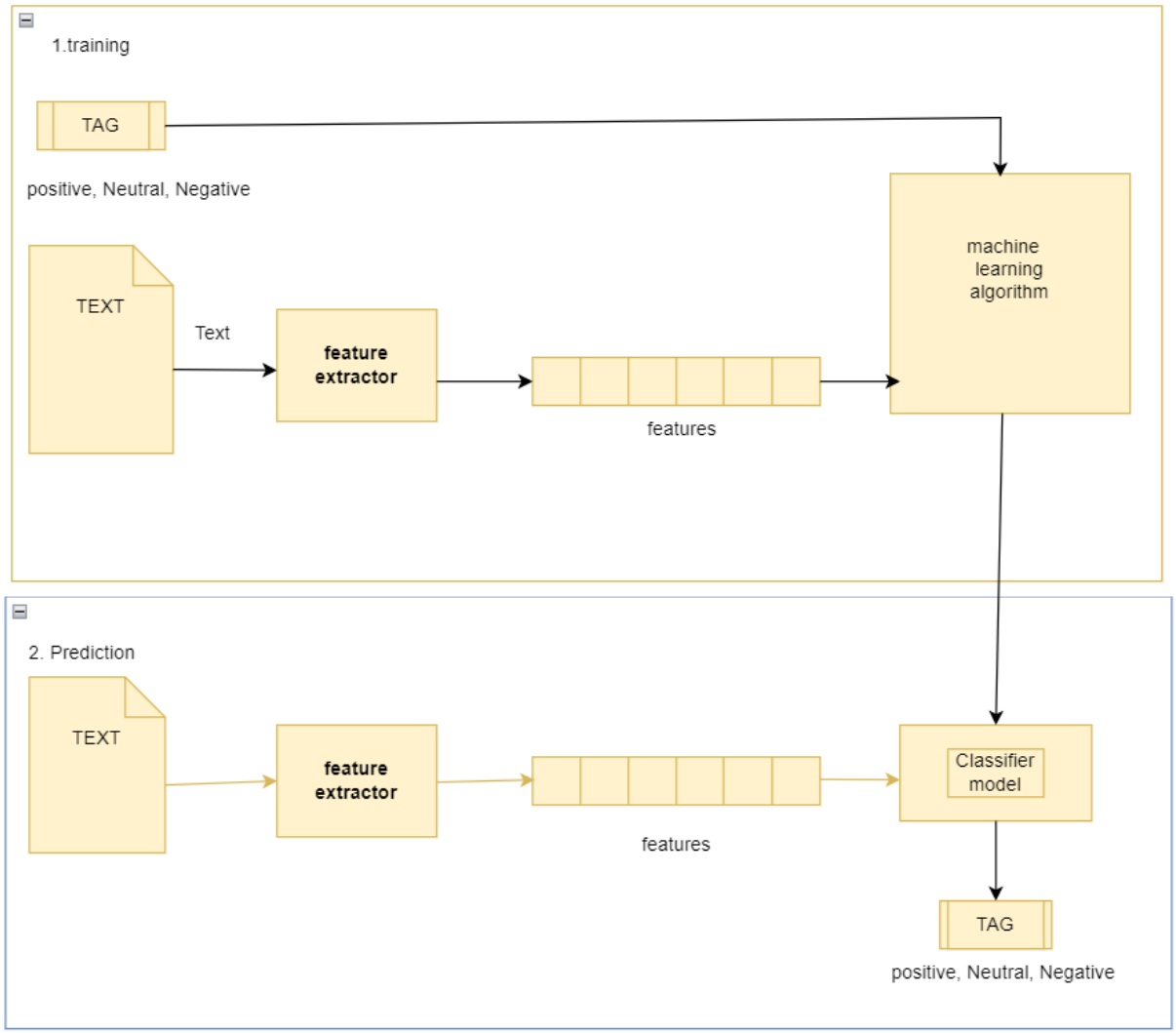
3.1.1. Nguồn gốc dữ liệu

Nguồn gốc của dữ liệu này được thu thập bằng cách lên các trang báo để lấy dữ liệu xuất phát từ các bài viết, tin tức, và nội dung khác được công bố trên các trang báo điện tử uy tín và phổ biến tại Việt Nam. Các trang báo này bao gồm VnExpress, Tuổi Trẻ. Các bài viết trên các trang này bao phủ nhiều lĩnh vực và chủ đề như Chính trị, Kinh tế, Xã hội, Giáo dục, Thể thao, Giải trí, Công nghệ, Khoa học, Y tế, v.v. mỗi bài viết sẽ được gán nhãn vào các chủ đề tương ứng. Và được lưu trữ trong file txt

3.1.2. Phân tích dữ liệu

3.1.2.1 Bài toán phân loại văn bản

Bài toán phân loại văn bản là bài toán xác định nhãn (label) của một văn bản dựa trên nội dung của nó. Mục tiêu là xây dựng một mô hình có thể dự đoán nhãn của văn bản mới dựa trên các mẫu đã được gán nhãn trong tập huấn luyện.



Hình 3.1. Mô hình phân loại văn bản tiếng Việt tự động với Machine learning (nguồn: MonkeyLearn.com)

Giai đoạn 1: Huấn luyện (training) là giai đoạn học tập của mô hình phân loại văn bản. Ở bước này, mô hình sẽ học từ dữ liệu có nhãn (trong ảnh trên nhãn là Possitive, Negative, Neutral). Dữ liệu văn bản sẽ được số hóa thông qua bộ trích xuất đặc trưng (feature extractor) để mỗi mẫu dữ liệu trong tập huấn luyện trở thành 1 vector nhiều chiều (đặc trưng). Thuật toán máy học sẽ học và tối ưu các tham số để đạt được kết quả tốt trên tập dữ liệu này. Nhãn của dữ liệu được dùng để đánh giá việc mô hình học tốt không và dựa vào đó để tối ưu.

Giai đoạn 2: Dự đoán (prediction), là giai đoạn sử dụng mô hình học máy sau khi nó đã học xong. Ở giai đoạn này, dữ liệu cần dự đoán cũng vẫn thực hiện các bước trích xuất đặc trưng. Mô hình đã học sau đó nhận đầu vào là đặc trưng đó và đưa ra kết quả dự đoán.

3.1.2.2 Tiền xử lí dữ liệu văn bản

Việc tiền sử lý dữ liệu là quá trình chuẩn hóa dữ liệu và loại bỏ các thành phần không có ý nghĩa cho việc phân loại văn bản.

Tiền xử lý dữ liệu tiếng Việt cho bài toán phân loại văn bản thường gồm các việc sau:

+ Xóa HTML code (nếu có)
+ Chuẩn hóa bảng mã Unicode (đưa về Unicode tổ hợp dựng sẵn)
+ Chuẩn hóa kiểu gõ dấu tiếng Việt (dùng òa úy thay cho oà uý)
+ Thực hiện tách từ tiếng Việt (sử dụng thư viện tách từ như pyvi, underthesea, vncorenlp,…)
+ đưa về văn bản lower (viết thường)
+ Xóa các ký tự đặc biệt: “.”, “,”, “;”, “)”, …

- Xóa html code trong dữ liệu

Dữ liệu được thu thập từ các website đôi khi vẫn còn sót lại các đoạn mã HTML. Các mã HTML code này là rác, chẳng những không có tác dụng cho việc phân loại mà còn làm kết quả phân loại văn bản bị kém đi.

Xóa HTML code bằng cách sử dụng regex trong python:

Bảng 3.1. Xóa HTMT code trong dữ liệu

```
def remove_html(txt):
return re.sub(r'< [^>]*>', '', txt)
txt = "<p class="par">Diên viên Quan Hiểu Đồng thành tâm điểm chú ý</p>"
remove_html(txt)
```

```
>>>kết quả:  'Diên viên Quan Hiểu Đồng thành tâm điểm chú ý'
```

- Giải thích code:
  - Hàm **remove_html** nhận một chuỗi **txt** làm đầu vào.

- Nó sử dụng hàm **re.sub** từ module **re** để thực hiện thay thế dựa trên biểu thức chính quy.
- Mẫu biểu thức chính quy là **r'< [^>]*>'**.

- Chuẩn hóa unicode tiếng việt

Bảng 3.2. Chuẩn hóa unicode tiếng việt

```
import regex as re

uniChars =
"àáảãạâầấẩẫậăằắẳẵặèéẻẽẹêềếểễệđìíỉĩịòóỏõọôồốổỗộơờớởỡợùúủũụưừứửữựỳýỷỹỵ
ÀÁẢÃẠÂẦẤẨẪẬĂẰẮẲẴẶÈÉẺẼẸÊỀẾỂỄỆĐÌÍỈĨỊÒÓỎÕỌÔỒỐỔỖỘƠỜỚỞ
ỠỢÙÚỦŨỤƯỪỨỬỮỰỲÝỶỸỴÂĂĐÔƠƯ"
unsignChars =
"aaaaaaaaaaaaaaaaaeeeeeeeeeeeediiiiioooooooooooooooooouuuuuuuuuuuyyyyyA
AAAAAAAAAAAAAAAAAEEEEEEEEEEEDIIIOOOOOOOOOOOOOOOOO
OOUUUUUUUUUUUYYYYYAADOOU"


def loaddicchar():
    dic = {}
    char1252 =
'à|á|ả|ã|ạ|ầ|ấ|ẩ|ẫ|ậ|ằ|ắ|ẳ|ẵ|ặ|è|é|ẻ|ẽ|ẹ|ề|ế|ể|ễ|ệ|ì|í|ỉ|ĩ|ị|ò|ó|ỏ|õ|ọ|ồ|ố|ổ|ỗ|ộ|ờ|ớ|ở|ỡ|ợ|ù|ú|ủ|ũ|
ụ|ừ|ứ|ử|ữ|ự|ỳ|ý|ỷ|ỹ|ỵ|À|Á|Ả|Ã|Ạ|Ầ|Ấ|Ẩ|Ẫ|Ậ|Ằ|Ắ|Ẳ|Ẵ|Ặ|È|É|Ẻ|Ẽ|Ẹ|Ề|Ế|Ể|Ễ|Ệ|Ì|Í|Ỉ|Ĩ|
Ị|Ò|Ó|Ỏ|Õ|Ọ|Ồ|Ố|Ổ|Ỗ|Ộ|Ờ|Ớ|Ở|Õ|Ợ|Ù|Ú|Ủ|Ũ|Ụ|Ừ|Ứ|Ử|Ữ|Ự|Ỳ|Ý|Ỷ|Ỹ|Ỵ'.split(
        '|')
    charutf8 =
"à|á|ả|ã|ạ|ầ|ấ|ẩ|ẫ|ậ|ằ|ắ|ẳ|ẵ|ặ|è|é|ẻ|ẽ|ẹ|ề|ế|ể|ễ|ệ|ì|í|ỉ|ĩ|ị|ò|ó|ỏ|õ|ọ|ồ|ố|ổ|ỗ|ộ|ờ|ớ|ở|ỡ|ợ|ù|ú|ủ|ũ
|ụ|ừ|ứ|ử|ữ|ự|ỳ|ý|ỷ|ỹ|ỵ|À|Á|Ả|Ã|Ạ|Ầ|Ấ|Ẩ|Ẫ|Ậ|Ằ|Ắ|Ẳ|Ẵ|Ặ|È|É|Ẻ|Ẽ|Ẹ|Ề|Ế|Ể|Ễ|Ệ|Ì|Í|Ỉ|Ĩ|
Ị|Ò|Ó|Ỏ|Õ|Ọ|Ồ|Ố|Ổ|Ỗ|Ộ|Ờ|Ớ|Ở|Õ|Ợ|Ù|Ú|Ủ|Ũ|Ụ|Ừ|Ứ|Ử|Ữ|Ự|Ỳ|Ý|Ỷ|Ỹ|Ỵ".split(
        '|')
    for i in range(len(char1252)):
```

```
            dic[char1252[i]] = charutf8[i]
        return dic



    dicchar = loaddicchar()

    # Đưa toàn bộ dữ liệu qua hàm này để chuẩn hóa lại
    def covert_unicode(txt):
        return re.sub(

r'à|á|ả|ã|ạ|ầ|ấ|ẩ|ẫ|ậ|ằ|ắ|ẳ|ẵ|ặ|è|é|ẻ|ẽ|ẹ|ề|ế|ể|ễ|ệ|ì|í|ỉ|ĩ|ị|ò|ó|ỏ|õ|ọ|ồ|ố|ổ|ỗ|ộ|ờ|ớ|ở|ỡ|ợ|ù|ú|ủ|ũ
|ụ|ừ|ứ|ử|ữ|ự|ỳ|ý|ỷ|ỹ|ỵ|À|Á|Ả|Ã|Ạ|Ầ|Ấ|Ẩ|Ẫ|Ậ|Ằ|Ắ|Ẳ|Ẵ|Ặ|È|É|Ẻ|Ẽ|Ẹ|Ề|Ế|Ể|Ễ|Ệ|Ì|Í|Ỉ|Ĩ|
Ị|Ò|Ó|Ỏ|Õ|Ọ|Ồ|Ố|Ổ|Ỗ|Ộ|Ờ|Ớ|Ở|Õ|Ợ|Ù|Ú|Ủ|Ũ|Ụ|Ừ|Ứ|Ử|Ữ|Ự|Ỳ|Ý|Ỷ|Ỹ|Ỵ',
            lambda x: dicchar[x.group()], txt)
```

- Giải thích code:
    - uniChars: Chuỗi chứa tất cả các ký tự Tiếng Việt có dấu.
    - unsignChars: Chuỗi chứa các ký tự Tiếng Việt tương ứng không dấu.
    - hàm loaddicchar để tạo từ điển chuyển đổi các ký tự từ định dạng TCVN3/CP1252 sang Unicode
    - char1252: Danh sách các ký tự Tiếng Việt có dấu trong định dạng TCVN3/CP1252, được phân tách bởi ký tự '|'.
    - charutf8: Danh sách các ký tự Tiếng Việt tương ứng trong định dạng Unicode, cũng được phân tách bởi ký tự '|'.
    - Tạo từ điển dic để ánh xạ từ các ký tự trong char1252 sang các ký tự tương ứng trong charutf8.
    - Sử dụng hàm re.sub để thay thế tất cả các ký tự Tiếng Việt có dấu (trong định dạng TCVN3/CP1252) trong chuỗi txt bằng các ký tự tương ứng trong định dạng Unicode.
    - Biểu thức chính quy
      `r'à|á|ả|ã|ạ|ầ|ấ|ẩ|ẫ|ậ|ằ|ắ|ẳ|ẵ|ặ|è|é|ẻ|ẽ|ẹ|ề|ế|ể|ễ|ệ|ì|í|ỉ|ĩ|ị|ò|ó|ỏ|õ|ọ|ồ|ố|ổ|ỗ|ộ|ờ|ớ|ở|ỡ|ợ|

ù|ú|ủ|ũ|ụ|ừ|ứ|ử|ữ|ự|ỳ|ý|ỷ|ỹ|ỵ|À|Á|Ả|Ã|Ạ|Ầ|Ấ|Ẩ|Ẫ|Ậ|Ằ|Ắ|Ẳ|Ẵ|Ặ|È|É|Ẻ|Ẽ|Ẹ|Ề|Ế|Ể|Ễ|Ệ|Ì|Í|Ỉ|Ĩ|Ị|Ò|Ó|Ỏ|Õ|Ọ|Ồ|Ố|Ổ|Ỗ|Ộ|Ờ|Ớ|Ở|Ỡ|Ợ|Ù|Ú|Ủ|Ũ|Ụ|Ừ|Ứ|Ử|Ữ|Ự|Ỳ|Ý|Ỷ|Ỹ|Ỵ' liệt kê tất cả các ký tự

- Chuẩn hóa kiểu gõ dấu

Kiểu gõ dấu khác nhau thì bạn nhìn mắt thường cũng sẽ thấy được sự khác nhau: òa với oà lần lượt là kiểu gõ cũ (phổ biến hơn) và kiểu gõ mới.

Bảng 3.3. Bảng chuẩn hóa kiểu gõ dấu trong unicode

```
def chuan_hoa_dau_tu_tieng_viet(word):
    if not is_valid_vietnam_word(word):
        return word

    chars = list(word)
    dau_cau = 0
    nguyen_am_index = []
    qu_or_gi = False
    for index, char in enumerate(chars):
        x, y = nguyen_am_to_ids.get(char, (-1, -1))
        if x == -1:
            continue
        elif x == 9:  # check qu
            if index != 0 and chars[index - 1] == 'q':
                chars[index] = 'u'
                qu_or_gi = True
        elif x == 5:  # check gi
            if index != 0 and chars[index - 1] == 'g':
                chars[index] = 'i'
                qu_or_gi = True
        if y != 0:
            dau_cau = y
            chars[index] = bang_nguyen_am[x][0]
        if not qu_or_gi or index != 1:
```

```
            nguyen_am_index.append(index)
        if len(nguyen_am_index) < 2:
            if qu_or_gi:
                if len(chars) == 2:
                    x, y = nguyen_am_to_ids.get(chars[1])
                    chars[1] = bang_nguyen_am[x][dau_cau]
                else:
                    x, y = nguyen_am_to_ids.get(chars[2], (-1, -1))
                    if x != -1:
                        chars[2] = bang_nguyen_am[x][dau_cau]
                    else:
                        chars[1] = bang_nguyen_am[5][dau_cau] if chars[1] == 'i' else
bang_nguyen_am[9][dau_cau]
                return ''.join(chars)
            return word

        for index in nguyen_am_index:
            x, y = nguyen_am_to_ids[chars[index]]
            if x == 4 or x == 8:  # ê, ơ
                chars[index] = bang_nguyen_am[x][dau_cau]
                # for index2 in nguyen_am_index:
                #     if index2 != index:
                #         x, y = nguyen_am_to_ids[chars[index]]
                #         chars[index2] = bang_nguyen_am[x][0]
                return ''.join(chars)

        if len(nguyen_am_index) == 2:
            if nguyen_am_index[-1] == len(chars) - 1:
                x, y = nguyen_am_to_ids[chars[nguyen_am_index[0]]]
                chars[nguyen_am_index[0]] = bang_nguyen_am[x][dau_cau]
                # x, y = nguyen_am_to_ids[chars[nguyen_am_index[1]]]
                # chars[nguyen_am_index[1]] = bang_nguyen_am[x][0]
```

```
            else:
                # x, y = nguyen_am_to_ids[chars[nguyen_am_index[0]]]
                # chars[nguyen_am_index[0]] = bang_nguyen_am[x][0]
                x, y = nguyen_am_to_ids[chars[nguyen_am_index[1]]]
                chars[nguyen_am_index[1]] = bang_nguyen_am[x][dau_cau]
        else:
            # x, y = nguyen_am_to_ids[chars[nguyen_am_index[0]]]
            # chars[nguyen_am_index[0]] = bang_nguyen_am[x][0]
            x, y = nguyen_am_to_ids[chars[nguyen_am_index[1]]]
            chars[nguyen_am_index[1]] = bang_nguyen_am[x][dau_cau]
            # x, y = nguyen_am_to_ids[chars[nguyen_am_index[2]]]
            # chars[nguyen_am_index[2]] = bang_nguyen_am[x][0]
        return ''.join(chars)


    def is_valid_vietnam_word(word):
        chars = list(word)
        nguyen_am_index = -1
        for index, char in enumerate(chars):
            x, y = nguyen_am_to_ids.get(char, (-1, -1))
            if x != -1:
                if nguyen_am_index == -1:
                    nguyen_am_index = index
                else:
                    if index - nguyen_am_index != 1:
                        return False
                    nguyen_am_index = index
        return True


    def chuan_hoa_dau_cau_tieng_viet(sentence):
        """
```

```
        Chuyển câu tiếng việt về chuẩn gõ dấu kiểu cũ.
        :param sentence:
        :return:
        """
    sentence = sentence.lower()
    words = sentence.split()
    for index, word in enumerate(words):
        cw = re.sub(r'(^p{P}*)([p{L}.]*p{L}+)(p{P}*$)', r'1/2/3', word).split('/')
        # print(cw)
        if len(cw) == 3:
            cw[1] = chuan_hoa_dau_tu_tieng_viet(cw[1])
        words[index] = ''.join(cw)
    return ' '.join(words)


"""
    End section: Chuyển câu văn về cách gõ dấu kiểu cũ: dùng òa úy thay oà uý
    Xem tại đây: https://vi.wikipedia.org/wiki/Quy_tắc_đặt_dấu_thanh_trong_chữ_quốc_ngữ
"""
if __name__ == '__main__':
    print(chuan_hoa_dau_cau_tieng_viet('anh hoà, đang làm.. gì'))
```

- Giải thích code:
    - bang_nguyen_am: Bảng chứa các nguyên âm tiếng Việt và các dạng dấu khác nhau của chúng.
    - nguyen_am_to_ids: Từ điển ánh xạ mỗi ký tự nguyên âm và dấu của nó tới vị trí tương ứng trong bang_nguyen_am.
    - is_valid_vietnam_word(word): Hàm kiểm tra tính hợp lệ của từ tiếng Việt bằng cách kiểm tra vị trí các nguyên âm.

- def chuan_hoa_dau_tu_tieng_viet(word): hàm chuẩn hóa tiếng việt và Xử lý các trường hợp đặc biệt với 'qu' và 'gi'.
- def chuan_hoa_dau_cau_tieng_viet(sentence): hàm chuẩn hóa tiếng việt trong một câuSử dụng biểu thức chính quy để tách từ và dấu câu, sau đó chuẩn hóa từng từ.

❖ Tách từ tiếng việt

Đơn vị từ trong tiếng Việt bao gồm từ đơn (yêu) và từ ghép (học sinh). Nên chúng ta cần phải nói cho mô hình học máy biết đâu là từ đơn, đâu là từ ghép. Nếu không thì từ nào cũng sẽ là từ đơn hết.

Bởi vì mô hình của chúng ta sẽ coi các từ là đặc trưng, tách nhau theo dấu cách. Do đó, chúng ta phải nối các từ ghép lại thành một từ để không bị tách sai:

Thường vụ Quốc hội đồng ý đề nghị của Viện trưởng =>Thường_vụ_Quốc_hội đồng_ý đề_nghị của Viện_trưởng

Bài toán này là một bài toán cơ sở trong NLP – bài toán tách từ (word tokenize)

Bảng 3.4. Bảng tách từ sử dụng thư viện word_tokenize

from underthesea import word_tokenize sentence = 'Chàng trai 9X Quảng Trị khởi nghiệp từ nấm sò' word_tokenize(sentence) word_tokenize(sentence, format="text")
– Kết quả của(word_tokenize(sentence)): ['Chàng trai', '9X', 'Quảng Trị', 'khởi nghiệp', 'từ', 'nấm', 'sò'] – Kết quả của word_tokenize(sentence, format="text"): 'Chàng_trai 9X Quảng_Trị khởi_nghiệp từ nấm sò'

- Giải thích code: Hàm **word_tokenize(sentence)** tách câu **sentence** thành danh sách các từ và cụm từ.
- Kết quả là danh sách các từ và cụm từ: **['Ca sĩ', 'Taylor Swift', 'chiếm', '14', 'vị**

❖ Đưa về viết thường (lowercase)

Việc đưa dữ liệu về chữ viết thường là rất cần thiết, Đưa về chữ viết thường giúp giảm số lượng đặc trưng (vì máy tính hiểu hoa thường là 2 từ khác nhau) và tăng độ chính xác hơn cho mô hình.

❖ Xóa các ký tự không cần thiết

Lợi ích:

Giảm số chiều đặc trưng, tăng tốc độ học và xử lý

Tránh làm ảnh hưởng xấu tới kết quả của mô hình

Các dấu ngắt câu, số đếm và các ký tự đặc biệt khác không giúp phân loại một văn bản thuộc chuyên mục nào. Do đó, chúng ta nên loại bỏ nó đi.

Riêng với số đếm, ngày tháng, email (Các token đặc biệt). Nếu có thể bạn nên đưa nó về các token chung như: <number>, <date>, <email>, … Việc này có thể không giúp ích cho mô hình học tốt hơn nhưng sẽ giúp ích trong việc giữ được mạch của dữ liệu.

Bảng 3.5. Bảng xóa các kí tự không cần thiết trong văn bản

```
def text_preprocess(document):
    # xóa html code
    document = remove_html(document)
    # chuẩn hóa unicode
    document = convert_unicode(document)
    # chuẩn hóa cách gõ dấu tiếng Việt
    document = chuan_hoa_dau_cau_tieng_viet(document)
    # tách từ
    document = word_tokenize(document, format="text")
    # đưa về lower
    document = document.lower()
    # xóa các ký tự không cần thiết
    document =
re.sub(r'[^swáàảãạăắằẳẵặâấầẩẫậéèẻẽẹêếềểễệóòỏõọôốồổỗộơớờởỡợíìỉĩịúùủũụưứừửữựý
ỳỷỹỵđ_]',' ',document)
```

```
    # xóa khoảng trắng thừa
    document = re.sub(r's+', ' ', document).strip()
    return document
```

- Giải thích code: đoạn code này định nghĩa hàm **text_preprocess** để tiền xử lý văn bản tiếng Việt. Hàm này thực hiện các bước sau: xóa mã HTML, chuẩn hóa Unicode, chuẩn hóa cách gõ dấu tiếng Việt, tách từ, chuyển thành chữ thường, xóa các ký tự không cần thiết và xóa khoảng trắng thừa.

  Ví dụ đoạn văn bản:

Vì thế, Bộ Công Thương khi xây dựng chính sách cần bảo đảm mục tiêu khuyến khích người dân sử dụng năng lượng tái tạo có sẵn, hài hòa lợi ích. Cơ quan này cũng phải rà soát để không có sơ hở dẫn đến trục lợi chính sách.

"Cần có quy định về hệ thống tích điện để nguồn tự sản, tự tiêu sử dụng không hết được bán thế nào, giá bán trên nguyên tắc nào?", thông báo nêu, thêm rằng các chính sách "nên khuyến khích bán điện dư thừa, nhưng có điều kiện".

Trước đó, theo dự thảo Bộ Công Thương, điện mặt trời mái nhà tự sản tự tiêu nối lưới hay không sẽ không được giao dịch mua bán. Tức là, điện dư thừa chọn phát lên lưới sẽ chỉ được ghi nhận sản lượng với giá 0 đồng.

Bởi, theo nhà chức trách, loại hình này được xây dựng với nhiều cơ chế ưu đãi như được nối lưới, miễn giấy phép hoạt động điện lực, không phải điều chỉnh công năng đất. Do đó, nếu cho mua bán sẽ gây ra tình trạng phát triển ồ ạt, khó kiểm soát dẫn tới tình trạng mất cân đối nguồn, ảnh hưởng an toàn hệ thống điện quốc gia.

Và dữ liệu sau khi xử lý:

vì_thế bộ công_thương khi xây_dựng chính_sách cần bảo_đảm mục_tiêu khuyến_khích người_dân sử_dụng năng_lượng tái_tạo có sẵn hài_hòa lợi_ích cơ_quan này cũng phải rà_soát để không có sơ_hở dẫn đến trục_lợi chính_sách cần có quy_định về hệ_thống tích_điện để nguồn tự_sản tự_tiêu sử_dụng không hết được bán thế_nào giá bán trên nguyên_tắc nào thông_báo nêu thêm rằng các chính_sách nên khuyến_khích bán điện dư_thừa nhưng có điều_kiện trước đó theo dự_thảo bộ

công_thương điện mặt_trời mái nhà tự_sản tự_tiêu nối lưới hay không sẽ không được giao_dịch mua_bán tức_là điện dư_thừa chọn phát lên lưới sẽ chỉ được ghi_nhận sản_lượng với giá 0 đồng bởi theo nhà_chức_trách loại_hình này được xây_dựng với nhiều cơ_chế ưu_đãi như được nối lưới miễn giấy_phép hoạt_động điện_lực không phải điều_chỉnh công_năng đất do_đó nếu cho mua_bán sẽ gây ra tình_trạng phát_triển ồ_ạt khó kiểm_soát dẫn tới tình_trạng mất cân_đối nguồn ảnh_hưởng an_toàn hệ_thống điện quốc_gia

3.1.2.3 Loại bỏ các stopword tiếng việt

Stopword là các từ xuất hiện nhiều ở tất cả các chuyên mục cần phân loại. Do đó, chúng là các đặc trưng không có tác dùng cho việc phân loại văn bản.

Các stopword thường là các từ nối (của, là, có, được, những, …) và các từ đặc trưng của dữ liệu

Bảng 3.6. Bảng loại bỏ stopword trong file

```
stopword = set()


def remove_stopwords(line):
    words = []
    for word in line.strip().split():
        if word not in stopword:
            words.append(word)
    return ' '.join(words)
```

- Giải thích code: đoạn code này định nghĩa tập hợp các từ dừng (stopwords) từ tệp **stopwords.txt** và sau đó loại bỏ các từ dừng này khỏi một chuỗi văn bản.

3.2. Xây dựng mô hình phân loại tiếng việt

Trước khi huấn luyện mô hình phân loại văn bản, ta cần xây dựng tập huấn luyện và tập kiểm thử. Việc này là cần thiết để đánh giá kết quả huấn luyện, lựa chọn mô hình cũng như tinh chỉnh để mô hình cho tốt hơn.

3.2.1. Xây dựng tập

Dữ liệu dùng cho bài toán phân loại văn bản của mình sau khi tiền xử lý được lưu thành 1 file duy nhất. Mỗi dòng là một bài báo kèm theo thông tin danh mục của nó.

1.__label__thời_sự Riêng, toàn_quốc xác_lập 110 kỷ_lục nhiệt_độ, 2023 gấp 10 so kỳ 4/2023. Trung_tâm Dự_báo Khí_tượng Thủy_văn quốc_gia xảy ba đợt nắng_nóng nắng_nóng gay_gắt tỉnh Tây_Bắc_Bộ, khu_vực Thanh_Hóa - Phú_Yên . Đợt thứ 1-4 / , đợt 12-17 / , đợt ba 19-30 / . Trong 26-30 / nắng_nóng xuất_hiện toàn_bộ Bắc_Bộ Trung_Bộ . Các tỉnh Tây_Bắc Bắc_Bộ Thanh_Hóa - Phú_Yên nắng_nóng gay_gắt nền nhiệt phổ_biến 39-42_độ , riêng Trung_Bộ nơi 43 độ C. Nhiệt_độ trung_bình Bắc_Bộ , Bắc Trung_Trung_Bộ 2-4_độ , nơi độ C so trung_bình kỳ . Các khu_vực phổ_biến 1-3_độ , riêng Tây_Nguyên nơi độ C.

2.__label__thế_giới Quan_chức cấp Mỹ kêu_gọi Nga Trung_Quốc con_người , thay_vì AI , đưa quyết_định triển_khai vũ_khí hạt_nhân . Paul_Dean , quan_chức Cục Kiểm_soát Vũ_khí , Răn_đe Ổn_định thuộc Bộ Ngoại_giao Mỹ , cuộc họp 1/5 Washington đưa " cam_kết mạnh_mẽ rõ_ràng " con_người toàn_quyền kiểm_soát đối_với vũ_khí hạt_nhân , bao_giờ trao quyền định_đoạt trí_tuệ nhân_tạo (AI) . Theo ông , Pháp Anh đưa cam_kết như_vậy . " Chúng_tôi hoan_nghênh tuyên_bố tương_tự Nga Trung_Quốc . Chúng_tôi chuẩn_mực cực_kỳ quan_trọng hành_vi trách_nhiệm nhóm P5 hoan_nghênh " , quan_chức , đề_cập 5 thành_viên thường_trực Hội_đồng Bảo_an Liên_Hợp_Quốc . Nga Trung_Quốc hiện chưa phản_hồi đề_nghị .

3.__label__phim Nhà_sản_xuất phát_hành first-look (video sơ_lược phim điện_ảnh) tối_29/4 , mở_đầu cảnh quân địch càn_quét miền Nam . Sau tiếng bom nổ bối_cảnh căn_cứ dưới lòng đất chiến_sĩ Mặt_trận Dân_tộc Giải_phóng miền Nam Việt_Nam . Video đoạn nhân_vật : " Mỹ kêu ' Củ_Chi , Sài_Gòn mất ' " . Đoạn trailer khép cảnh căn hầm rung tiếng bom , tưởng_chừng sụp_đổ . Dự_án kỷ_niệm 50 thống_nhất đất_nước , phim chiến_tranh ngân_sách_xã_hội hóa . Ngoài Thái_Hòa , dàn diễn_viên Quang_Tuấn , Hồ_Thu_Anh , Cao_Minh , Diễm_Hằng Lamoon , Hoàng_Minh_Triết , Khánh_Ly . Đạo_diễn Bùi_Thạc_Chuyên mất 10 chuẩn dự_án , nhằm tập_trung tái_hiện tinh_thần yêu

nhân_dân miền Nam . Tác_phẩm ghi_hình , cảnh thực_hiện Củ_Chi . " Khó tưởng_tượng lính , dân_quân kiên_cường dùng dụng_cụ thô_sơ xây_dựng hệ_thống phòng_thủ dày_đặc phức_tạp ngay dưới lòng đất " , Bùi_Thạc_Chuyên .

4.__label__âm_nhạc Ca_sĩ Taylor_Swift chiếm 14 vị_trí bảng xếp_hạng top ca_khúc ăn_khách Mỹ , bán 2,6 triệu bản tuần . Hôm_30/4 , tạp_chí Billboard xác_nhận album Taylor_Swift - The_Tortured_Poets Department - 14 ca_khúc chiếm top Hot 100 , bảng xếp_hạng bài hát yêu_thích tuần Mỹ . Giọng ca 35 tuổi phá kỷ_lục chính cô 2022 album Midnights (10 bài) . Đĩa đơn_Fortnight - hát Post_Malone - trở_thành ca_khúc thứ 12 đứng Hot 100 Taylor_Swift . Ra_mắt tuần , đĩa cô thống_trị Billboard 200 - bảng xếp_hạng album bán_chạy - 2,6 triệu đơn_vị , 1,9 triệu đĩa nhạc , quy_đổi lượt nghe trực_tuyến , tải nhạc số . Cô bán 859.000_đĩa than , phá kỷ_lục doanh_số tuần loại đĩa suốt 30 Mỹ . Taylor_Swift Jay-Z hiện nghệ_sĩ solo album đứng Billboard 200 - 12 đĩa . Cô chiếm_lĩnh trang nghe nhạc trực_tuyến . Tại Spotify , album đạt 1,18 tỷ lượt nghe toàn_cầu bảy , đĩa nhạc vượt mốc tuần ra_mắt . Apple_Music xác_nhận The_Tortured_Poets Department album nghe lịch_sử hệ_thống nhạc số . Trên trang cá_nhân , Taylor_Swift dám tin thành_tích album . " Cảm_ơn bạn nghe chào_đón The_Tortured_Poets cuộc_sống bạn . Tôi thật_sự choáng_ngợp " , cô .

5.__label__thời_trang Các bộ_đồ tính biểu_tượng Công_nương_Diana trưng_bày triển_lãm 30 bà . Theo Lifestyle_Asia , hãng Julien's_Auction giới_thiệu hiện_vật Công_nương_Diana (1961 - 1997) , 18-29 / . Đây triển_lãm phong_cách Vương_phi kể buổi đấu_giá từ_thiện 79 bộ váy Christie's tổ_chức New_York 1997 . Khách tham_quan dịp chiêm_ngưỡng di_sản thời_trang_Diana tư_cách thành_viên hoàng_gia_Anh công_chúng yêu_mến Điểm nhấn triển_lãm chiếc đầm xanh đính họa_tiết ngôi_sao Diana_diện buổi ra_mắt nhạc_kịch Phantom_of the_Opera 1986 chiếc váy lụa đen thiết_kế Victor_Edelstein . Ban tổ_chức ước_tính giá_cả khoảng 200.000 400.000_USD. Bên_cạnh , sự_kiện trưng_bày bộ suit gam vàng chân váy màu navy Catherine_Walker - thiết_kế riêng Công_nương Diana 16 - thực_hiện .

Diana mặc chiếc váy Hong_Kong 1989 . Thiết_kế dự_kiến giá 30.000 50.000 USD._

6.__label__thể_thao Loạt trận tứ_kết diễn dịp nghỉ lễ vừa_qua , ba số bốn trận đấu giải_quyết đá luân_lưu . Bình_Dương đội bóng duy_nhất sót miền Nam . Họ tưởng tạo bất_ngờ hòa chủ Nam_Định 1-1 120 phút thi_đấu . Nhưng , loạt luân_lưu , tiền_đạo U23 Việt_Nam Bùi_Vĩ_Hào hoàn_thành nhiệm_vụ , Bình_Dương thua_3-4 . Đương_kim giữ Cup_Thanh_Hóa trải trận đấu vất_vả Hải_Phòng . Đội bóng xứ Thanh_dẫn phút 18 nhờ pha phản_lưới Đặng_Văn_Tới . Nhưng 10 phút , đội khách gỡ hòa_1-1 công_Lương Hoàng_Nam . Trong loạt sút luân_lưu , Thanh_Hóa hút chết tiền đạo trụ_cột Rimario hỏng quả . Nhưng , nhờ xuất_sắc thủ_môn Xuân_Hoàng , cản_phá thành_công , thầy_trò HLV Popov bán_kết tỷ_số 4-2 . Trận đấu cuối_cùng cần loạt luân_lưu diễn giữa đội hạng Nhất_là PVF-CAND Thể_Công . Sau vượt gỡ hòa , cầu_thủ Thể_Công thực_hiện chính_xác lượt_sút 11 m , giành chiến_thắng 5-3 . Ở trận đấu , Đà_Nẵng thi_đấu kiên_cường không_thể tạo bất_ngờ khách Hà_Nội FC._Trận , đội bóng thủ_đô vươn dẫn cú đúp Nguyễn_Hai_Long phút 35 55 , đội khách rút ngắn phút 76 công Liễu_Quang_Vinh . Tại bán_kết , Thanh_Hóa gặp hàng_xóm Nam_Định sân_nhà . Mùa , Nam_Định bay V-League dẫn_đầu 32 điểm , Thanh_Hóa xếp thứ 10 điểm . Tuy_nhiên , đội bóng thành Nam hòa 1-1 khách sân Thanh_Hóa V-League . Do_đó , lợi_thế sân_nhà , Thanh_Hóa lợi_thế đường bảo_vệ danh_hiệu Cup_Quốc_gia . Trận derby thủ_đô giữa Hà_Nội Thể_Công . Ở lượt V-League_mùa , Hà_Nội dễ_dàng thắng 2-0 . Dù đội phong_độ , tính_chất trận đấu_Cup hứa_hẹn kịch_tính . Mùa , Thể_Công_lọt chung_kết thua Thanh_Hóa loạt sút luân_lưu .

7.__label__giáo_dục Sau đánh học_sinh lớp bầm vai , cô_giáo huyện Tân_Thạnh mẹ em tát ngay trường . Sự_việc xảy hôm 2/4 trường Tiểu_học THCS Tân_Bình (điểm Cây_Sao) . Thấy học vết bầm vai tay , phụ_huynh liền gặng hỏi . Bé cô_giáo đánh . Gia_đình đưa bệnh_viện kiểm_tra , chẩn_đoán chấn_thương phần_mềm , xây_xát vùng da_tay , vai . Do bức_xúc , mẹ học_sinh trường gặp giáo_viên tát cô cái . Phòng Giáo_dục_và_Đào_tạo huyện Tân_Thạnh yêu_cầu cô_giáo viết tường_trình . Nữ giáo_viên cho_hay dùng thước đánh bé chịu bài_tập giờ Toán . Chiều 19/4 , ông Lê_Thanh_Đông ,

Chủ_tịch UBND huyện Tân_Thạnh , nữ giáo_viên 35 tuổi chịu hình_thức kỷ_luật khiển_trách điều_chuyển trường tiểu_học xã . " Phía gia_đình đồng_ý hình_thức kỷ_luật , tuy_nhiên yêu_cầu cô_giáo trực_tiếp nói_chuyện " , ông Đông cho_hay . Do cô_giáo ốm ngày_mai nhà_chức_trách gặp gia_đình . Luật_Giáo_dục quy_định giáo_viên xúc_phạm danh_dự , thân_thể học . Tùy mức_độ , bốn hình_thức kỷ_luật giáo_viên nếu vi_phạm , gồm khiển_trách , cảnh_cáo , cách_chức hoặc buộc thôi_việc . Về phụ_huynh tát giáo_viên , dùng điện_thoại quay video đăng mạng xã_hội , Chủ_tịch UBND huyện Tân_Thạnh chỉ_đạo công_an vào_cuộc . " Việc cô_giáo đánh học_sinh sai phụ_huynh trường đánh xem_xét xử_lý " , ông Đông .

8.__label__sức_khỏe Bệnh_viện Đa_khoa khu_vực Long_Khánh tiếp_nhận 209 nghi_ngộ_độc ăn bánh_mì , trường_hợp nặng chuyển viện tuyến . Số bệnh_nhân lần_lượt nhập_viện 1-2 / 5 , 160 nằm viện điều_trị , 5 khám rồi xin , 43 trường_hợp nhẹ xuất_viện ngay , lãnh_đạo Bệnh_viện Đa_khoa Long_Khánh sáng 2/5 . Bệnh_nhân xuất_hiện triệu_chứng bất_thường ăn bánh_mì thịt mua cơ_sở kinh_doanh phường Xuân_Bình , Long_Khánh , 15 h 19 h 30/4 . Dấu_hiệu chung nôn , tiêu_chảy , sốt , đau bụng . Bác_sĩ chẩn_đoán bệnh_nhân nhiễm_trùng đường ruột , nghi_ngộ_độc thực_phẩm . " Đa_số bệnh_nhân hiện ổn_định sức_khỏe " , đại_diện bệnh_viện cho_hay . Dự_kiến hôm_nay Chi_cục An_toàn Vệ_sinh thực_phẩm tỉnh Đồng_Nai làm_việc Trung_tâm Y_tế cơ_quan liên_quan TP Long_Khánh điều_tra nguyên_nhân vụ_việc . Ngoài_ra , Sở Y_tế chỉ_đạo Bệnh_viện Đa_khoa khu_vực Long_Khánh đảm_bảo điều_kiện tốt điều_trị bệnh_nhân . Cơ_sở bánh_mì hoạt_động 10 nay . Kiểm_tra cơ_sở hôm_qua , cơ_quan_chức_năng ghi_nhận nơi giấy chứng_nhận đủ điều_kiện an_toàn vệ_sinh thực_phẩm . Cơ_sở tạm ngưng kinh_doanh kết_luận cơ_quan_chức_năng .

9.__label__du_lịch Thấy hình_ảnh Y_Tý mùa vàng đẹp mơ mạng xã_hội , chị Thu_Anh , Hà_Nội , ngay nhóm chat bạn lập kế_hoạch " lên_đường ngay luôn " . Chỉ quyết_định Y_Tý , Lào_Cai , chị Thu_Anh đặt xong dịch_vụ chuyến . Thu_Anh chốt ngay homestay đêm 5 " chill " đặt dịch_vụ ăn_uống tại_chỗ . Quyết_định du_lịch lướt_mạng xã_hội ảnh quá đẹp . Chuyến nữ du_khách như_ý thời_tiết xấu , " quyết_định quá nhanh vội " , cô khẳng_định " quay dịp

phục_thù nơi trải_nghiệm " . Chị Hương_Chi , TP HCM , chuyến khám_phá hang Sơn_Đoòng " tuyệt_vời " hồi . Ba , đọc bài viết " xem xem_lại " hình du_khách nước_ngoài mạng xã_hội , chị nhủ " nhất_định nào Sơn_Đoòng " . " Mong_ước ấy thành sự_thật món quà sinh_nhật tuổi 40 ý_nghĩa " , chị Chi . Trong cuộc hội_thảo mới_đây TP HCM thay_đổi xu_hướng du_lịch thế_giới , bà Lê_Khánh_Linh , Giám_đốc Entravision_Vietnam , đối_tác độc_quyền nền_tảng_mạng xã_hội Việt_Nam , cho_hay gặp nhóm bạn trẻ Áo chuyến Sơn_Đoòng . Họ sang Việt_Nam thứ bỏ 3.000 USD (70 triệu đồng) mỗi tour_Sơn_Đoòng tình_cờ nhìn hình_ảnh nơi mạng xã_hội . Bà Linh_dẫn thống_kê Statista 10/2023 thế_giới hiện 4,95 tỷ dùng mạng xã_hội , phổ_biến Facebook , Tiktok , Instagram , X (Twitter) , Snapchat , Pinterest . 36,5 % khách du_lịch sử_dụng mạng xã_hội tìm ý_tưởng cảm_hứng du_lịch , xếp thứ ba tìm_kiếm thông_tin hình_ảnh . Xu_hướng phổ_biến du_khách trẻ tuổi thuộc thế_hệ Gen_Z (sinh 1997 2012) , đối_tượng khách_hàng chính ngành du_lịch hiện_tại . Đây chính sinh giai_đoạn bùng_nổ mạng xã_hội , sống tự_do tự đưa quyết_định nhanh_chóng . " Khách du_lịch tìm_kiếm thông_tin chuyến , điều xảy ý_tưởng , ý_tưởng cảm_hứng mạng xã_hội " , đại_diện Entravision cho_hay .

10.__label__xe Trung_Quốc đơn_thuần quê_hương hãng xe , Nhật , Hàn , nó phân_khúc khách_hàng riêng_biệt . Xe Trung_Quốc cụm_từ khá nhạy_cảm Việt_Nam , sẵn_sàng đưa nhận_xét kiểu " xe Trung_Quốc thì thế_nọ , thì thế kia " chưa bao_giờ sử_dụng . Chúng_ta dễ_dàng mua thiết_bị Xiaomi , đánh_giá kiểu_dáng , chất_lượng chúng , dè_bỉu chiếc ôtô đất_nước . Tất_nhiên , định_kiến khá dễ giải_thích , trước_đây , sản_phẩm xe_máy Trung_Quốc Việt_Nam nhanh xuống_cấp , kể_cả một_số hãng ôtô thời . Nhưng cần công_bằng , sản_phẩm xuất_xứ Trung_Quốc chất_lượng kém " thương_lái Việt_Nam đặt_hàng , nhu_cầu chất_lượng , giá_thành tối_thiểu " . Điểm giỏi Trung_Quốc sản_phẩm thiết_kế tương_tự nhau , mức giá . Vì_vậy , chính Việt_Nam Việt_Nam mất niềm tin sản_phẩm Trung_Quốc , chứ sản_phẩm nào kém .

11.__label__kinh_doanh bữa ăn 8 triệu khách bàng_hoàng đắt đắt kinh_doanh zing vn quản_lý nhà_hàng phục_vụ khách bữa ăn 8 triệu hà_nội dịch_vụ

khách_hàng hưởng số tiền hóa đơn thực_chất chưa lợi_nhuận 12 11 cư_dân mạng đồng_loạt chia_sẻ hình_ảnh hóa đơn nhà_hàng dành nhóm khách nam mức chi_trả 7 7 triệu đồng vợ khách dùng_bữa nhà_hàng hóa đơn thể_hiện dùng hết 15 chiếc khăn bông 6 bình đỗ đen loại 5 lít đĩa thịt chim giá 44 triệu đĩa rau sống 200 000 đồng quá đắt_đỏ trao_đổi phóng_viên zing vn ông trường quản_lý nhà_hàng a_bunadh mức giá quá nếu tính dịch_vụ khách_hàng hưởng khách tự rượu nhà_hàng phục_vụ bởi 6 nhân_viên bar_trưởng nhiệm_vụ rót rượu loại rượu khách đắt_đỏ chúng_tôi sử_dụng ly chuyên_dụng đồng_thời tặng suối nguồn thử rượu loại đắt_đỏ ông trường hóa đơn 7 7 triệu đồng khách_hàng thắc_mắc quá đắt_đỏ chi_tiết vô_lý ảnh fbnv giải_thích thắc_mắc khách_hàng uống hết 10 lít khoảng ngắn quản_lý số_lượng cần_thiết trung_hòa loại rượu mạnh khách_hàng sử_dụng đồ uống khách nồng_độ cồn 50 độ mức đốt cháy 18 h30 23 h30 uống hết chai rượu thì lượng dùng thế quá quản_lý nhà_hàng khẳng_định ông trường bổ_sung đỗ đen nhà_hàng sử_dụng nấu loại ruột_xanh chuyên_dùng giã rượu khách giá mỗi ly đỗ đen so cam khách_hàng thoải_mái trả_giá cam 80 000 đồng tại_sao thắc_mắc bình đỗ đen_giá tương_tự riêng lời phàn_nàn khách món ăn gồm đĩa chim giá 44 triệu đồng rau sống 200 000 đồng_đĩa hoa_quả giá 540 000 đồng phí phục_vụ rượu đại_diện a_bunadh nhận_định cảm_tính mỗi chim cắt miếng chúng_tôi đủ thì khách nhận ngay riêng phí phục_vụ bình_thường nhà_hàng tính 30 40 giá chai rượu khách chi_phí cơ_hội nhà_hàng tính nếu phục_vụ rượu khách tuy_nhiên đặt bàn_là khách quen chúng_tôi lấy chi_phí tượng_trưng 500 000 đồng đại_diện nhà_hàng khẳng_định người_quản_lý nhấn_mạnh nhà_hàng diện_tích 200 m2 thuê phố_lý thường_kiệt đắt_đỏ 2007 nay 6 phòng ăn mức giá dịch_vụ kèm thực_ra nhà_hàng phục_vụ nhóm khách hề lãi trường bình_luận hóa đơn bữa tối 8 triệu đồng lương nguyễn_định_phương quản_lý au manoir de khai nhà_hàng cao_cấp tùy chất_lượng nhà_hàng mục_tiêu khách mức giá hợp_lý ngoài_ra mức giá tùy thuộc tỷ_lệ lợi_nhuận dự_kiến nhà_hàng trừ chi_phí nguyên_liệu nhân_công thuê mặt_bằng phí phục_vụ rượu phương cho_hay thông_thường mức tính giá_trị sản_phẩm khách thực_tế nhà_hàng chọn yêu_cầu khách rượu hoặc trả mức phí tối_thiểu 30 100 usd nếu khách muốn dùng đồ uống tự tuy_nhiên trường_hợp nhà_hàng phục_vụ miễn_phí rượu

khách yêu_cầu dùng đạt hạn_mức chi_trả bữa ăn tối_thiểu quy_định nhà_hàng miễn_phí phục_vụ rượu nếu khách chi dùng số tiền tối_thiểu hoặc sử_dụng menu nào quy_định nói_chung khách nhà_hàng cần thống_nhất phục_vụ tránh rủi_ro nảy_sinh thanh_toán phương giám_sát nhà_hàng sao super hotel vũ_văn_thủ tổ_hợp nhà_hàng phục_vụ khăn ăn mỗi chiếc nếu khách yêu_cầu chứ tự_ý liên_tục thay_khăn bẩn trường_hợp nhà_hàng phục_vụ bữa ăn 8 triệu giá đĩa rau sống như_vậy hơi quá lố_song tùy thuộc loại rau gồm món cao_cấp như_thế_nào hiện_tại tổ_hợp nhà_hàng thường tính rau kèm món ăn nếu khách nhu_cầu gọi giá dưới 100 000 đồng đĩa thủ chia_sẻ khảo_sát giá thuê địa_điểm nhà_hàng diện_tích sàn 200 m2 khu_vực đường_lý thường kiệt_hoàn kiếm hà_nội hiện dao_động 200 250 triệu đồng khách bàng_hoàng bữa ăn 8 triệu nhà_hàng sang_trọng bạn nhà_hàng dùng_bữa tối tổng_số tiền 7 77 triệu tiền khăn bông 120 000 đồng chim nướng 440 triệu đồng

__label__giải_trí britney spears khoe vòng ngấn mỡ sao hollywood zing vn công_chúa nhạc thả_dáng vui_đùa sóng biển hawaii trai kỳ_nghỉ mới_đây hôm 24 7 britney spears trai jayden 8 tuổi sean 9 tuổi bờ biển hawaii vui_chơi nữ ca_sĩ diện bộ bikini mảnh khoe hình_thể săn chắc nữ ca_sĩ 33 tuổi gây chú_ý vòng ngấn_mỡ hình_ảnh thân_hình xuống_cấp britney spears chủ_đề bàn_tán nay chìm_đắm tiệc_tùng giọng ca toxic giữ vẻ thanh_xuân vài nay britney nỗ_lực lấy vóc_dáng ít cô bắt_gặp tình_trạng ăn_mặc lôi_thôi vòng sồ sề công_chúa nhạc pop tận_hưởng buổi chơi trai cô tung_tăng chạy nhảy_đùa nghịch dưới biển britney hỏi_thăm cậu trai sean dường_như thấm_mệt nằm cát nghỉ_ngơi ba mẹ_con nữ ca_sĩ thường_xuyên quấn_quýt mỗi kỳ_nghỉ chia_tay bạn trai charlie ebersol britney buồn_phiền bởi cô chàng trai nhỏ giọng ca tóc vàng cầm phao tiếp_tục xuống_nước vui_chơi britney vài nghỉ trở_lại sân_khấu piece of_me planet hollywood las vegas hợp_đồng brit khu resort sòng bạc kéo_dài 2016

- Trong tập dữ liệu này gồm có 11 nhãn bao gồm các nhãn: Thời sự, thế giới, phim, xe, thể thao, thời trang, du lịch, giải trí, kinh doanh, giáo dục, sức khỏe. Đi kèm với mỗi nhãn là nội dung của nó

❖ Thống kê số lượng data theo nhãn

Bảng 3.7. Bảng thống kê lượng data theo nhãn

Đoạn code	Kết quả chạy

count = {}
for line in open('file_text.txt',
encoding='utf-8'):
  key = line.split()[0]
  count[key] = count.get(key, 0) + 1
 	
for key in count:
  print(key, count[key])
	__label__thời_sự 192
	__label__thế_giới 134
	__label__phim 95
	__label__âm_nhạc 127
	__label__thời_trang 81
	__label__thể_thao 196
	__label__giáo_dục 104
	__label__sức_khỏe 101
	__label__du_lịch 73
	__label__xe 74
	__label__giải_trí 86

- Giải thích code: đoạn code này đọc tệp văn bản **file_text.txt**, đếm số lần xuất hiện của label trong mỗi dòng, và sau đó in ra label cùng với số lần xuất hiện của nó.

❖ Thống kê tất cả các từ xuất hiện trong nhãn

Bảng 3.8. Bảng thống kê 100 xuất hiện trong các nhãn

Đoạn code	Kết quả chạy

total_label = 11
vocab = {}
label_vocab = {}
for line in open('file_text.txt',
encoding='utf-8'):
  words = line.split()
  label = words[0]
  if label not in label_vocab:
  label_vocab[label] = {}
  for word in words[1:]:
	và 471
	của 413
	là 329
	có 301
	với 292
	trong 281
	được 280
	một 264
	cho 259
	những 226

label_vocab[label][word] = label_vocab[label].get(word, 0) + 1
if word not in vocab:
vocab[word] = set()
vocab[word].add(label)
 	
count = {}
for word in vocab:
if len(vocab[word]) == total_label:
count[word] = min([label_vocab[x][word] for x in label_vocab])
 	
sorted_count = sorted(count, key=count.get, reverse=True)
for word in sorted_count[:100]:
print(word, count[word])
	không 210
	khi 208
	các 204
	người 195
	này 153
	đã 140
	năm 138
	đến 138
	nhiều 136
	trên 135
	để 135
	từ 129
	như 122
	cũng 118
	ở 117
	...
	gần 24
	đặc_biệt 24
	con 24
	vì 23

- Giải thích code:
  - Đọc dòng từ tệp văn bản file_text.txt.
  - Tách từng dòng thành các từ và đếm tần suất xuất hiện của từng từ trong mỗi nhãn.
  - Xây dựng từ điển vocab lưu trữ tất cả các từ và nhãn mà chúng xuất hiện.
  - Xây dựng từ điển label_vocab lưu trữ tần suất xuất hiện của từng từ trong từng nhãn.
  - Tìm các từ xuất hiện trong tất cả các nhãn (tổng số nhãn là total_label).
  - In ra 100 từ có tần suất xuất hiện cao nhất trong tất cả các nhãn.

- Sử dụng thư viện sklearn trong python giúp tách dữ liệu làm 2 tập train/ test riêng biệt

Bảng 3.9. Bảng tách 2 tệp riêng biệt để huấn luyện

```
# chia tập train/test
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
test_percent = 0.3

text = []
label = []

for line in open('file_text.prep', encoding='utf-8'):
    words = line.strip().split()
    label.append(words[0])
    text.append(' '.join(words[1:]))

X_train, X_test, y_train, y_test = train_test_split(text, label, test_size=test_percent, random_state=42)
with open('train.txt', 'w', encoding='utf-8') as fp:
    for x, y in zip(X_train, y_train):
        fp.write('{} {}\n'.format(y, x))

with open('test1.txt', 'w', encoding='utf-8') as fp:
    for x, y in zip(X_test, y_test):
        fp.write('{} {}\n'.format(y, x))
label_encoder = LabelEncoder()
label_encoder.fit(y_train)
print(list(label_encoder.classes_), '\n')
y_train = label_encoder.transform(y_train)
y_test = label_encoder.transform(y_test)
```

```
print(X_train[0], y_train[0], '\n')
print(X_test[0], y_test[0])
```

- Giải thích code:
    - Đọc dữ liệu từ file và tách làm 2 list text (dữ liệu) và label (nhãn). Dữ liệu text[i] sẽ có nhãn là label[i].
    - Chia làm 2 tập train (X_train, y_train) và test (X_test, y_test) theo tỉ lệ 80% train, 20% test.
    - Lưu train/test data ra file để sử dụng cho việc train với thư viện Fasttext.
    - Đưa label về dạng vector để tiện cho tính toán sử dụng LabelEncoder.

3.2.2. Phân loại văn bản với naive bayes

Bảng 3.10. Bảng phân loại tiếng việt với thuật toán naive bayes

```
import pickle
import time
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline

start_time = time.time()
text_clf = Pipeline([('vect', CountVectorizer(ngram_range=(1,1),
                                max_df=0.7,
                                max_features=None)),
                     ('tfidf', TfidfTransformer()),
                     ('clf', MultinomialNB())
                    ])
text_clf = text_clf.fit(X_train, y_train)

train_time = time.time() - start_time
print('Done training Naive Bayes in', train_time, 'seconds.')
```

```
pickle.dump(text_clf, open(os.path.join(MODEL_PATH, "naive_bayes.pkl"), 'wb'))
```

- Giải thích code:
    - Thư viện pickle: Dùng để lưu trữ mô hình sau khi huấn luyện.
    - Thư viện time: Dùng để tính thời gian huấn luyện.
    - Thư viện CountVectorizer: Dùng để chuyển đổi văn bản thành ma trận đếm các từ.
    - Thư viện TfidfTransformer: Dùng để chuyển đổi ma trận đếm thành ma trận TF-IDF.
    - Thư viện MultinomialNB: Thuật toán Naive Bayes đa thức.
    - Thư viện: Dùng để kết hợp nhiều bước xử lý dữ liệu và mô hình vào một luồng.

3.2.3. Đánh giá mô hình

- Đánh giá mô hình với thuật toán Naïve Bayes

Bảng 3.11. Bảng đánh giá mô hình tổng quan với thuật toán Naïve Bayes

Đoạn code	Kết quả

import numpy as np
# Naive Bayes 	
model = pickle.load(open(os.path.join(MODEL_PATH,"naive_bayes.pkl"), 'rb'))
y_pred = model.predict(X_test)
print('Naive Bayes, Accuracy =', np.mean(y_pred == y_test))
	Naive Bayes, Accuracy = 0.6956521739130435

➢ Kết quả: Naive Bayes, Accuracy = 0.6150943396226415

- Giải thích code:
    - LabelEncoder() dùng để chuyển đổi các nhãn (categorical labels) thành các số nguyên.
    - fit() được sử dụng để tìm các nhãn duy nhất trong y_train.

- transform() dùng để chuyển đổi các nhãn thành các số nguyên dựa trên bộ dữ liệu đã được fit trước đó.
- text_preprocess() và remove_stopwords() được sử dụng để tiền xử lý văn bản mới trước khi dự đoán nhãn.
- pickle.load() được sử dụng để tải mô hình đã lưu trữ từ tệp đã được chỉ định.
- predict() dùng để dự đoán nhãn cho văn bản mới.
- inverse_transform() dùng để chuyển đổi các số nguyên (dự đoán được) thành các nhãn tương ứng.

- Xem kết quả trên từng nhãn

Bảng 3.12. Đánh giá mô hình của từng nhãn

	PRECISION	**RECALL**	**F1-SCORE**	**SUPPORT**
__label__du_lịch	1.00	0.10	0.18	20
__label__giáo_dục	1.00	0.93	0.96	14
__label__giải_trí	1.00	0.13	0.24	15
__label__phim	0.93	0.81	0.87	16
__label__sức_khỏe	1.00	0.67	0.80	24
__label__thế_giới	0.93	0.54	0.68	26
__label__thể_thao	0.91	0.98	0.94	50
__label__thời_sự	0.38	1.00	0.55	34
__label__thời_trang	1.00	0.45	0.62	11
__label__xe	1.00	0.28	0.43	18
__label__âm_nhạc	0.62	0.74	0.74	25
accuracy			0.70	253
macro avg	0.89	0.62	0.64	253
weighted avg	0.85	0.70	0.67	253

- Giải thích code:
  - nb_model=pickle.load(open(os.path.join(MODEL_PATH,"naive_bayes.pkl"), 'rb')) Load mô hình Naive Bayes đã được huấn luyện từ tệp đã lưu trữ.

- y_pred = nb_model.predict(X_test) : Sử dụng mô hình đã được load để dự đoán nhãn trên tập dữ liệu kiểm tra (**X_test**).
- print(classification_report(y_test,y_pred,target_names=list(label_encoder.classes_))) :
- classification_report là một hàm từ sklearn.metrics được sử dụng để tạo báo cáo đánh giá của mô hình phân loại.
- y_test là nhãn thực tế trên tập dữ liệu kiểm tra.
- y_pred là nhãn dự đoán được từ mô hình.
- target_names được sử dụng để chỉ định tên của các nhãn. Trong trường hợp này, nó được lấy từ list(label_encoder.classes_), tức là danh sách các nhãn đã được chuyển đổi thành dạng nhãn.

❖ Nhận xét:

- Đánh giá chi tiết hiệu suất mô hình

Precision: Chỉ số Precision thể hiện tỷ lệ các dự đoán chính xác trong số các dự đoán mà mô hình cho rằng thuộc về một lớp cụ thể. Các lớp như __label__du_lịch, __label__giáo_dục, __label__giải_trí, và __label__xe có chỉ số precision cao, cho thấy mô hình ít khi dự đoán sai khi xác định một mẫu thuộc các lớp này. Điều này cho thấy mô hình khá chắc chắn khi đưa ra các dự đoán cho những lớp này.

Recall: chỉ số Recall cho biết tỷ lệ các mẫu thực sự thuộc về một lớp cụ thể mà mô hình có thể nhận diện chính xác. Những lớp như __label__thể_thao, __label__thời_sự, và __label__âm_nhạc có recall cao, biểu thị rằng mô hình có khả năng nhận diện tốt các mẫu thuộc các lớp này. Tuy nhiên, những lớp như __label__du_lịch, __label__giải_trí, và __label__xe có chỉ số recall thấp, nghĩa là nhiều mẫu thực sự thuộc các lớp này không được mô hình nhận diện đúng.

F1-score: chỉ số F1-score là trung bình điều hòa của precision và recall, cung cấp cái nhìn cân bằng về hiệu suất của mô hình. Các lớp như __label__giáo_dục, __label__thể_thao, và __label__âm_nhạc có f1-score cao, cho thấy mô hình hoạt động tốt trong việc nhận diện và dự đoán chính xác các lớp này. Ngược lại, các lớp như __label__du_lịch, __label__giải_trí, và __label__xe có f1-score thấp, biểu thị rằng mô hình hoạt động kém hiệu quả đối với các lớp này.

Accuracy: Độ chính xác tổng quan của mô hình là 0.69, tức là mô hình dự đoán đúng 69% trong tổng số các mẫu. Mặc dù đây là một con số tương đối tốt, nhưng vẫn có thể cải thiện thêm để nâng cao hiệu suất của mô hình.

Macro avg và Weighted avg:

Macro avg (trung bình số học) cung cấp cái nhìn cân bằng về các chỉ số cho từng lớp mà không xét đến số lượng mẫu trong mỗi lớp, với các giá trị precision là 0.88, recall là 0.62 và f1-score là 0.63.

Weighted avg (trung bình có trọng số) tính toán trung bình các chỉ số dựa trên số lượng mẫu của từng lớp, mang lại cái nhìn toàn diện hơn về hiệu suất của mô hình, với các giá trị precision là 0.85, recall là 0.69 và f1-score là 0.66.

- Đánh giá tổng quát

Mô hình hoạt động tốt trong việc dự đoán các lớp có số lượng mẫu lớn như __label__thể_thao và __label__thời_sự. Tuy nhiên, hiệu suất kém trong việc dự đoán các lớp như __label__du_lịch và __label__xe có thể do số lượng mẫu trong các lớp này ít hoặc do sự phân bố dữ liệu không đồng đều. Việc cải thiện recall cho các lớp có chỉ số recall thấp là cần thiết để mô hình có thể nhận diện đầy đủ hơn các mẫu thuộc các lớp này.

mặc dù mô hình hiện tại đạt được một số thành công nhất định, nhưng vẫn cần tiếp tục cải thiện để đạt hiệu suất cao hơn, đặc biệt là đối với các lớp có hiệu suất hiện tại chưa tốt.

3.3. Triển khai xây dựng ứng dụng web

3.3.1. Xây dựng ứng dụng bằng Flask

Bảng 3.13.Bảng xây dựng ứng dụng bằng thư viện flask

```
from flask import Flask
from flask import request
from flask_cors import CORS
```

```
app = Flask(__name__)
CORS(app)
```

- Flask:
    - ‘from flask import Flask, from flask import request’: Import Flask và request từ thư viện Flask. Flask là một micro web framework dùng để xây dựng các ứng dụng web.
    - ‘app = Flask(__name__)’: Tạo một ứng dụng Flask. ‘__name__’ là tên module hiện tại, giúp Flask xác định đường dẫn đến các file tĩnh và template.
- Cors:
    - from flask_cors import CORS: Import CORS từ thư viện flask_cors.
    - CORS(app): Kích hoạt Cross-Origin Resource Sharing (CORS) cho ứng dụng Flask. CORS cho phép các tài nguyên trên một trang web được yêu cầu từ một tên miền khác với tên miền mà tài nguyên đó đã được phục vụ, giúp ứng dụng có thể nhận yêu cầu từ các nguồn gốc khác nhau (cross-origin).

3.3.2. Triển khai API dự đoán nhãn của văn bản đầu vào

Bảng 3.14. Bảng triển khai API dự đoán nhãn văn bản đầu vào

```
@app.route("/")
def hello_world():
  return "<p>Hello, World!</p>"


@app.route("/classify/predict", methods=["POST"])
def classify_predict():
  data = request.json
  document = data.get("text")
  document = text_preprocess(document)
  document = remove_stopwords(document)
  nb_model = pickle.load(open(os.path.join(MODEL_PATH,"naive_bayes.pkl"),
'rb'))
  label = nb_model.predict([document])
```

```
print("Label:", label)
print('Predict label:', label_encoder.inverse_transform(label))
# return label_encoder.inverse_transform(label)
resp = []
for item in label_encoder.inverse_transform(label):
  item = item.replace("__label__","")
  item = item.replace("_"," ")
  resp.append(item)

return resp
```

❖ **Định nghĩa route cơ bản**:

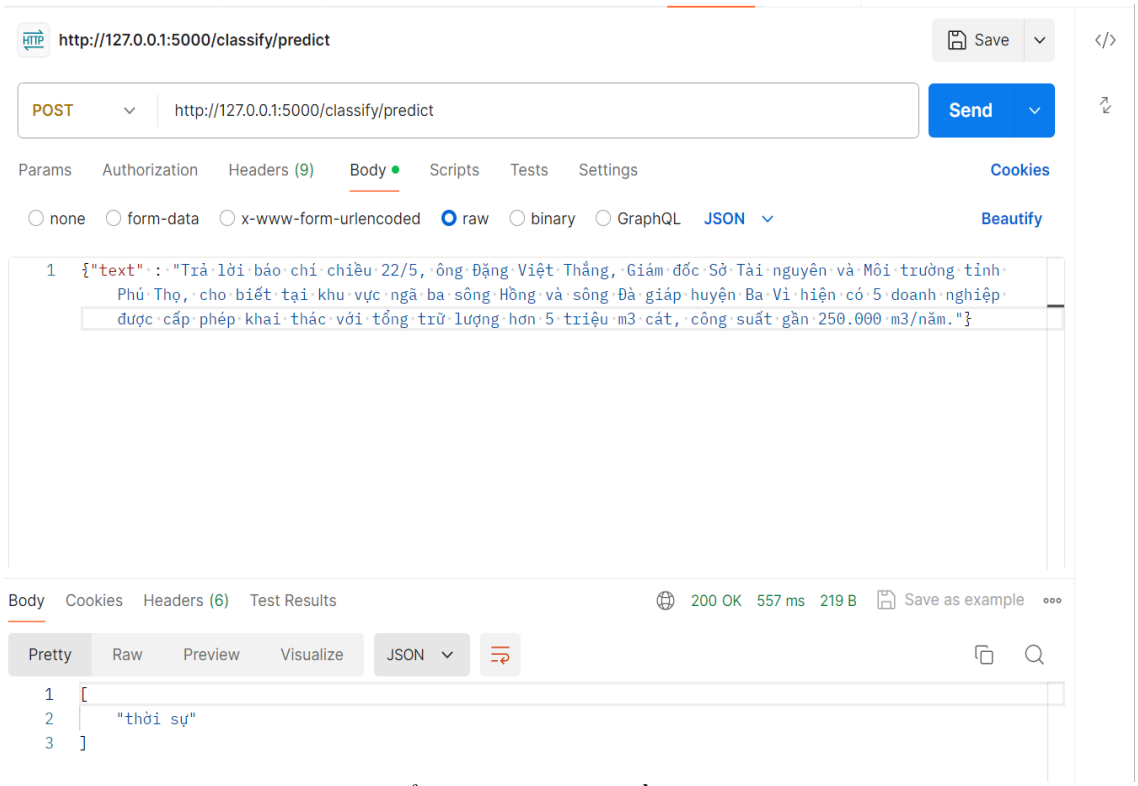
- **@app.route ("/")**: Định nghĩa một route tại đường dẫn gốc ("/").
- **def hello_world ()**: Hàm xử lý yêu cầu đến đường dẫn gốc.
- **return "<p>Hello, World! < /p >"**: Trả về một chuỗi HTML đơn giản khi truy cập đường dẫn gốc.

❖ **Định nghĩa API dự đoán nhãn**:

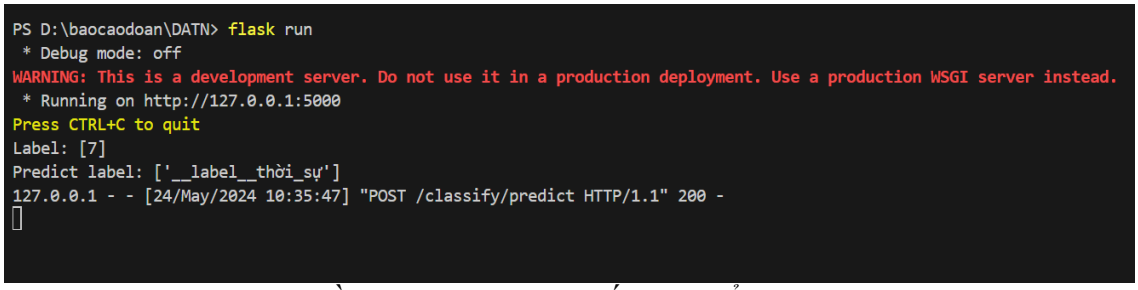
- **@app.route ("/classify/predict", methods=["POST"])**: Định nghĩa một route tại đường dẫn "/classify/predict" cho các yêu cầu POST.
- **def classify_predict()**: Hàm xử lý yêu cầu POST để phân loại văn bản.
- **data = request.json**: Lấy dữ liệu JSON từ yêu cầu.
- **document = data.get("text")**: Lấy văn bản cần phân loại từ dữ liệu JSON.
- **document = text_preprocess(document)**: Tiền xử lý văn bản (chuẩn hóa Unicode, chuẩn hóa dấu, token hóa, chuyển về chữ thường, loại bỏ ký tự không cần thiết).
- **document = remove_stopwords(document)**: Loại bỏ stopwords khỏi văn bản.
- **nb_model**
- **pickle.load(open(os.path.join(MODEL_PATH,"naive_bayes.pkl"), 'rb'))**: Tải mô hình Naive Bayes đã huấn luyện từ file.

- **label = nb_model.predict([document])**: Dự đoán nhãn của văn bản đã tiền xử lý.
- **Print ("Label:", label)**: In nhãn dự đoán ra console.
- **Print ('Predict label:', label_encoder.inverse_transform(label))**: In nhãn dự đoán sau khi chuyển từ mã hóa số sang nhãn gốc.
- **resp = []**: Khởi tạo danh sách để lưu kết quả phản hồi.
- **for item in label_encoder.inverse_transform(label)**: Lặp qua các nhãn dự đoán gốc.
- **item = item.replace("__label__","")**: Loại bỏ tiền tố "**label**" khỏi nhãn.
- **item = item.replace("_"," ")**: Thay thế dấu gạch dưới bằng dấu cách trong nhãn.
- **resp.append(item)**: Thêm nhãn đã chỉnh sửa vào danh sách phản hồi.
- **return resp**: Trả về danh sách nhãn dự đoán dưới dạng phản hồi JSON.

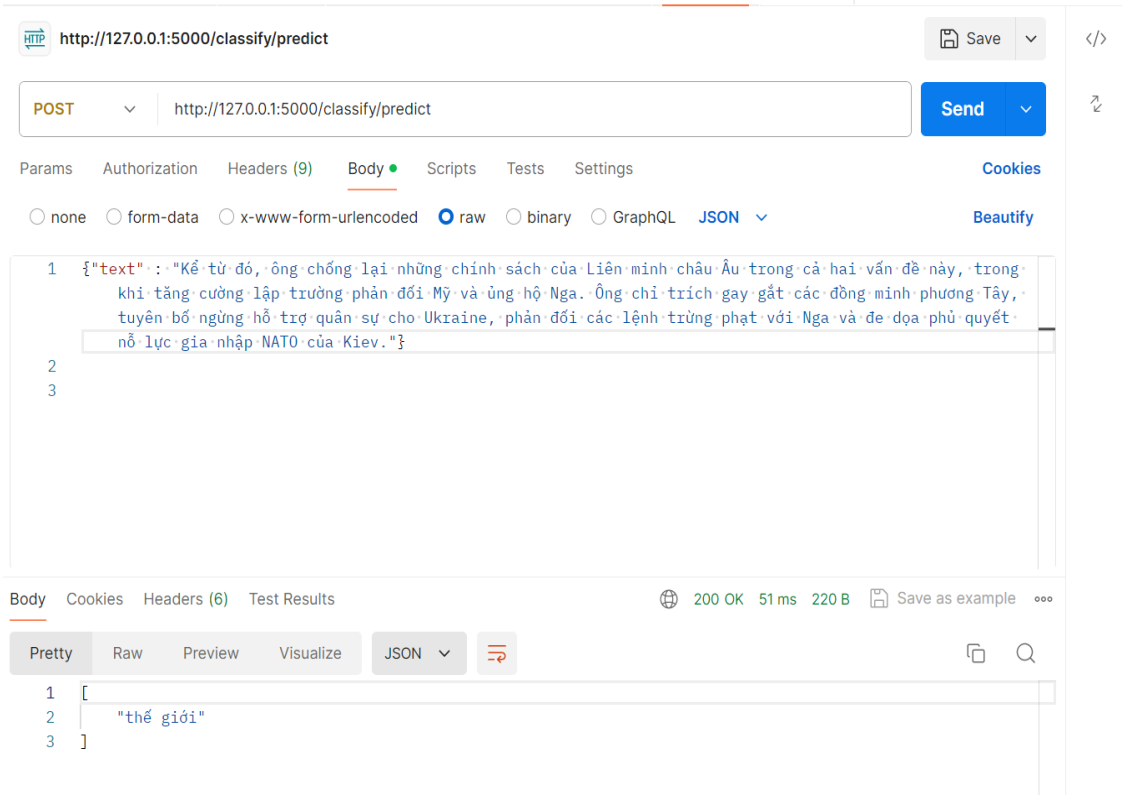
3.3.3. Triển khai API dự đoán văn bản trên Postman



Hình 3.2. Kiểm thử api chủ đề thời sự trên postman



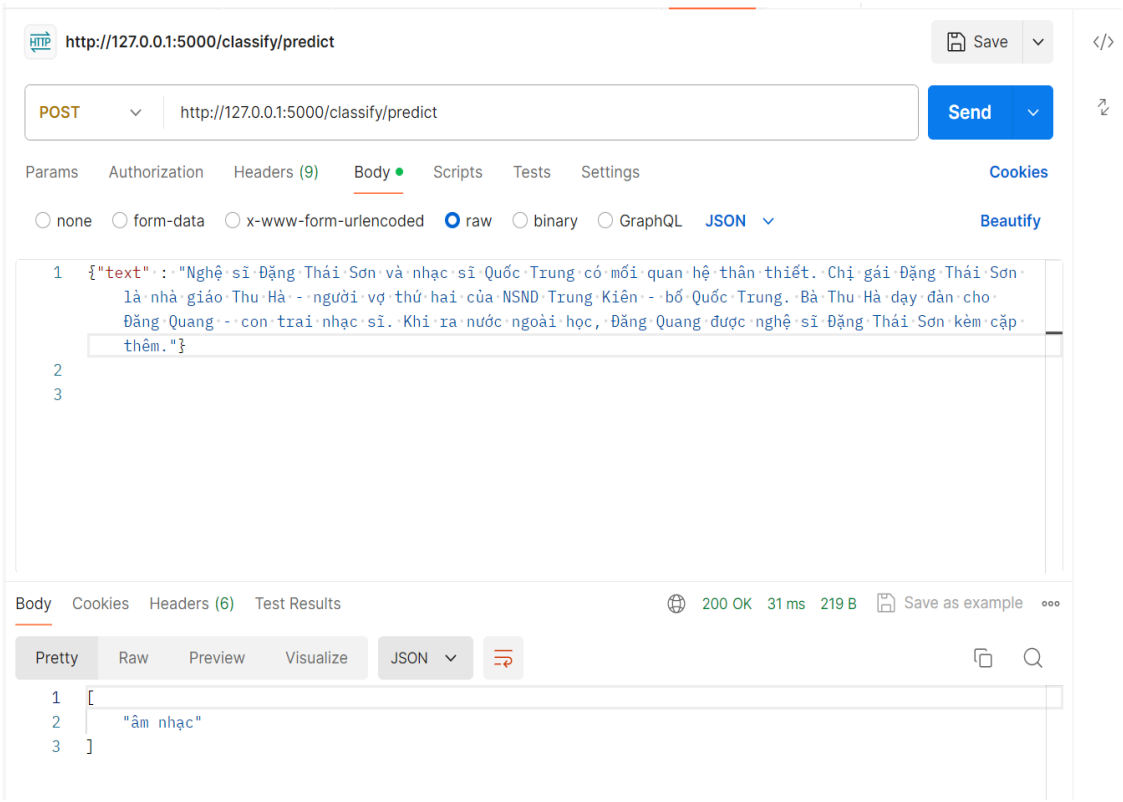
Hình 3.3. HTTP trả về từ máy chủ flask kết quả kiểm thử label thời sự trong quá trình kiểm tra trên Postman



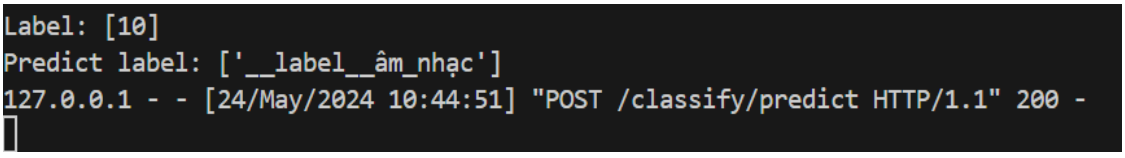
Hình 3.4. Kiểm thử api chủ đề thế giới trên postman

```
Label: [5]
Predict label: ['__label__thế_giới']
127.0.0.1 - - [24/May/2024 10:38:53] "POST /classify/predict HTTP/1.1" 200 -
```

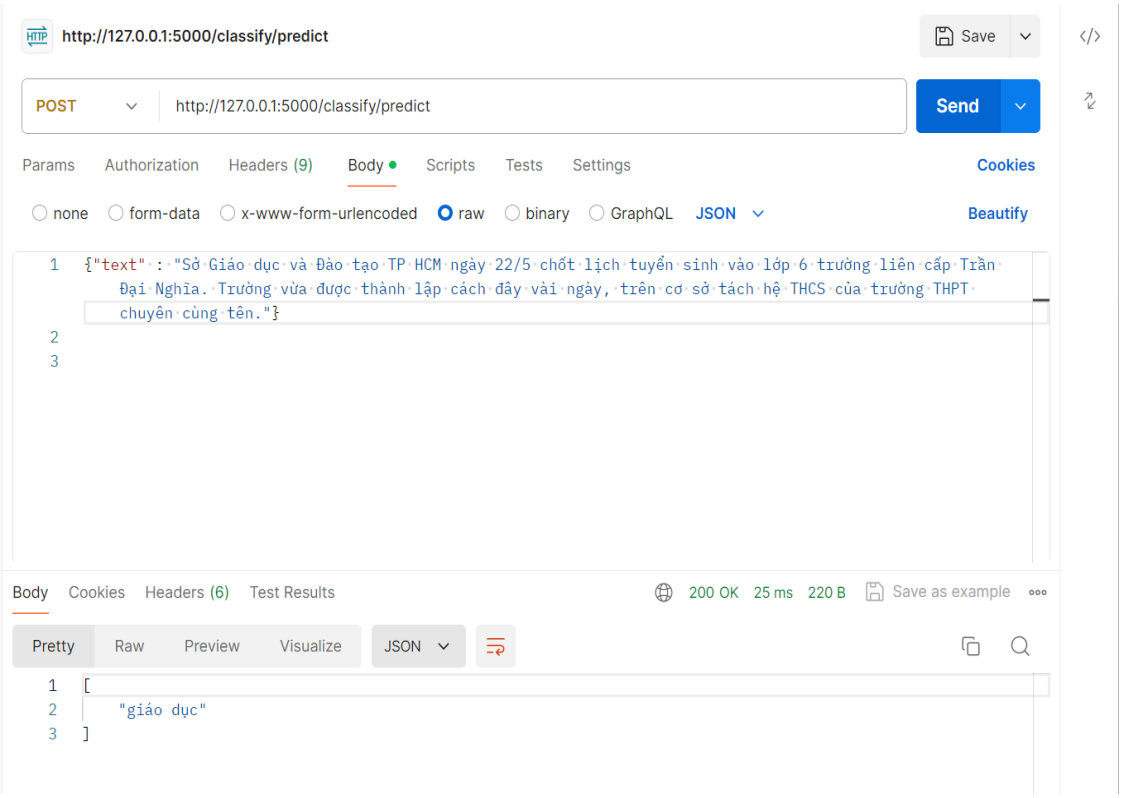
Hình 3.5. HTTP trả về từ máy chủ flask kết quả kiểm thử label thế giới trong quá trình kiểm tra trên Postman



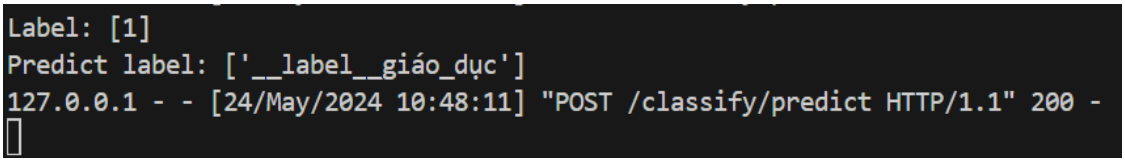
Hình 3.6 Kiểm thử api chủ đề âm nhạc trên postman



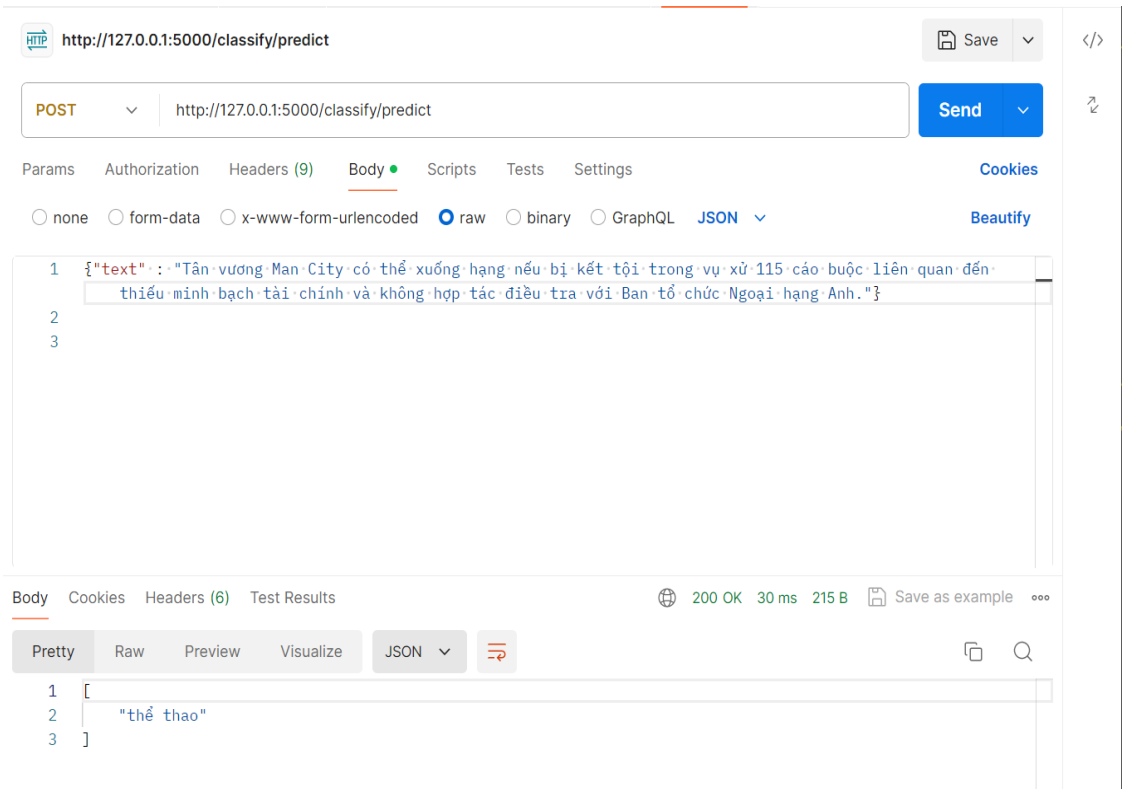
Hình 3.7. HTTP trả về từ máy chủ flask kết quả kiểm thử label âm nhạc trong quá trình kiểm tra trên Postman



Hình 3.8. Kiểm thử api chủ đề giáo dục trên postman

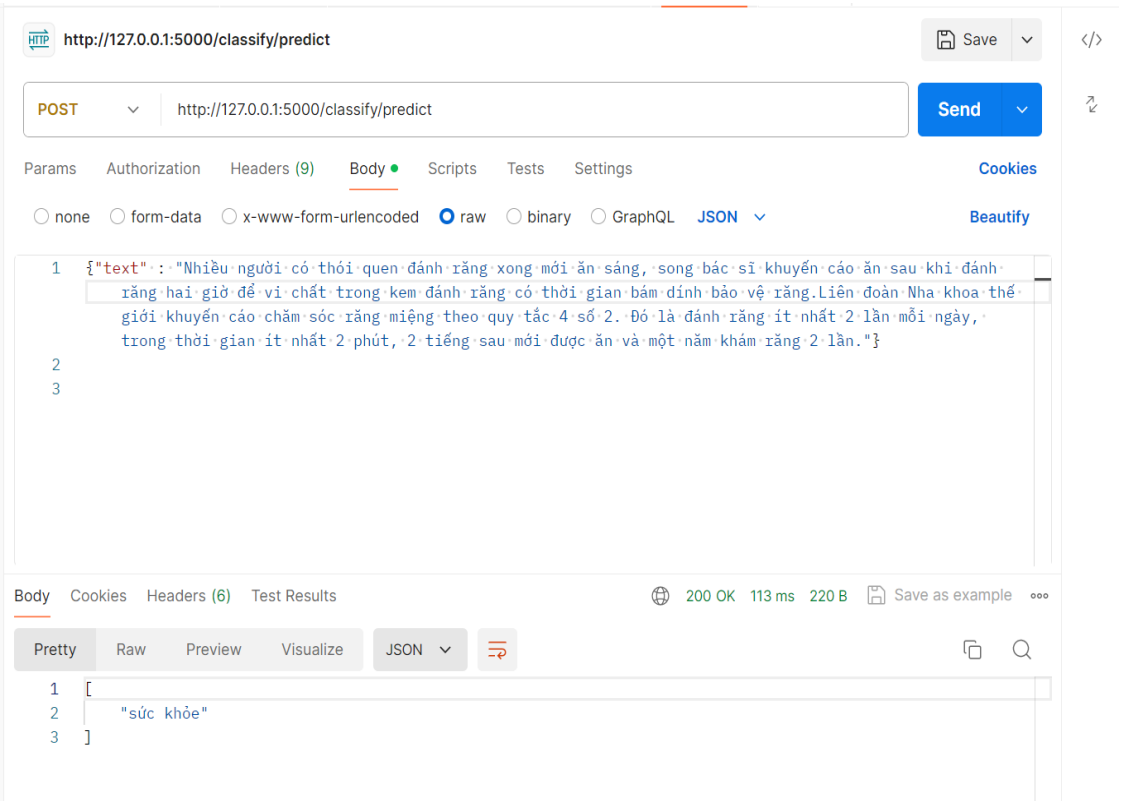


Hình 3.9. HTTP trả về từ máy chủ flask kết quả kiểm thử label giáo dục trong quá trình kiểm tra trên Postman

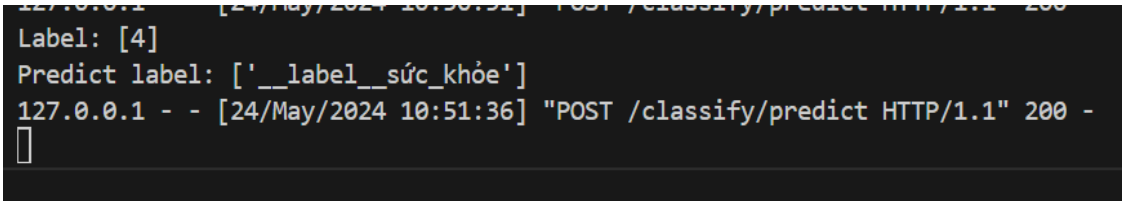


Hình 3.10. Kiểm thử api chủ đề thể thao trên postman

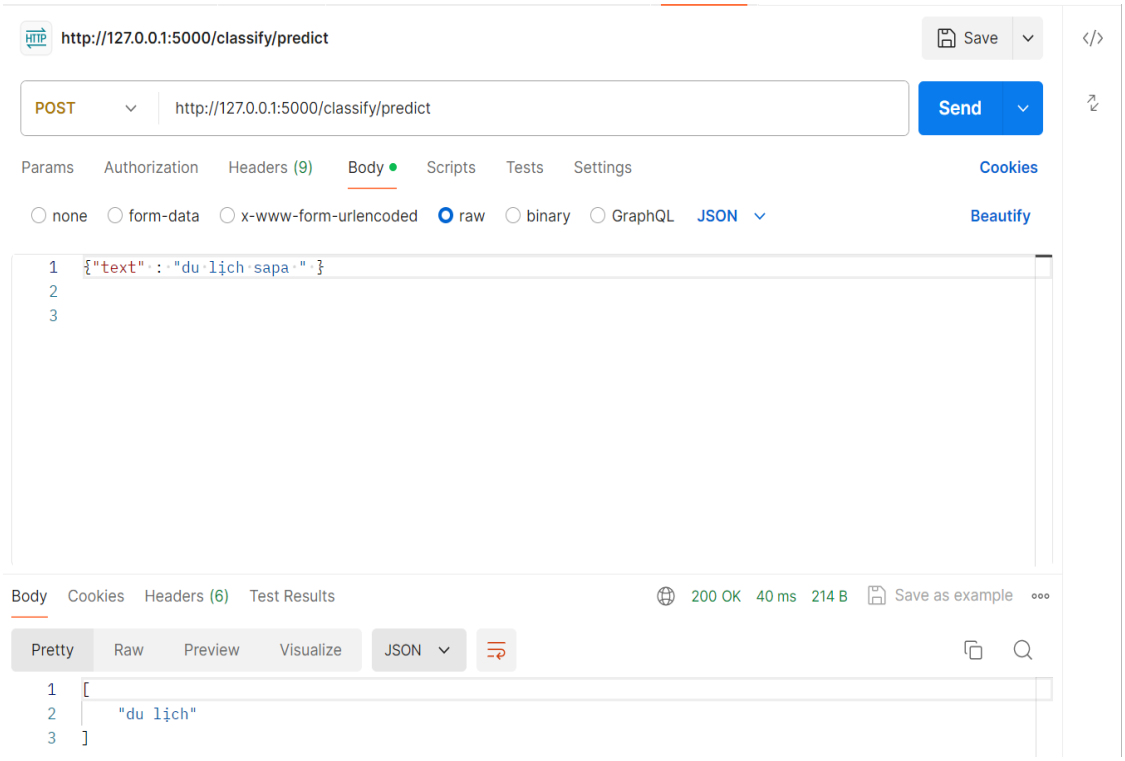
Hình 3.11. HTTP trả về từ máy chủ flask kết quả kiểm thử label thể thao trong quá trình kiểm tra trên Postman



Hình 3.12. Kiểm thử api chủ đề sức khỏe trên postman



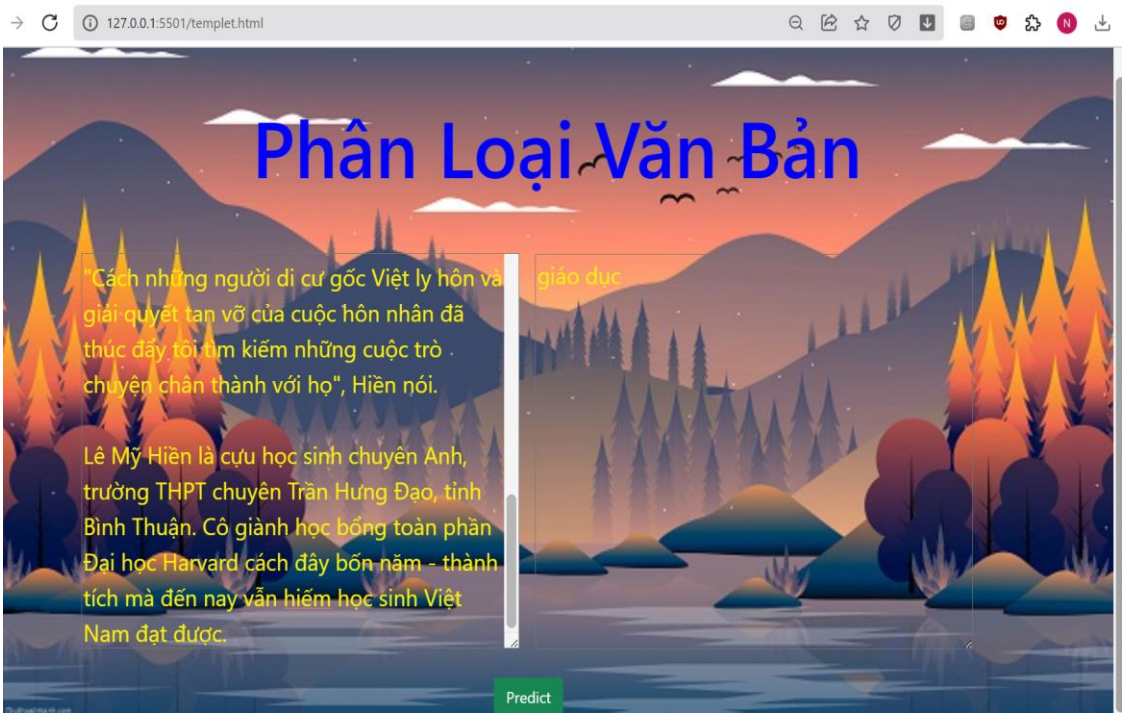
Hình 3.13. HTTP trả về từ máy chủ flask kết quả kiểm thử label sức khỏe trong quá trình kiểm tra trên Postman



Hình 3.14. Kiểm thử api chủ đề du lịch trên postman

Hình 3.15. HTTP trả về từ máy chủ flask kết quả kiểm thử label du lịch trong quá trình kiểm tra trên Postman

3.3.4. Triển khai trên giao diện web



Hình 3.16. Giao diện web phân loại tiếng việt

Dựa trên mô hình thuật toán phân loại tiếng việt đã được triển khai trước đó, em đã phát triển một ứng dụng hoàn chỉnh sử dụng ngôn ngữ Python và framework Flask API để xây dựng API phân loại tiếng việt khi, sau khi nhập đoạn văn bản vào ô nhập nội dung và bấm nút Predict thì sẽ hiện ra chủ đề tương ứng với nội dung nhập vào. Đồng thời, em cũng đã sử dụng HTML, CSS, và JavaScript để tạo giao diện người dùng và gọi API từ phía máy khách.

Ứng dụng được phát triển bằng Python và sử dụng Flask API để xây dựng các endpoint API cho việc dự đoán kết quả. Giao diện người dùng được xây dựng bằng HTML, CSS và JavaScript để tương tác với người dùng cuối.

Công việc phía sau (back-end) bao gồm:

- Xây dựng các mô hình học máy với các tham số tối ưu, dựa trên các mô hình đã được xác định trước đó.
- Tạo các endpoint API để truy cập và xử lý dữ liệu từ giao diện người dùng.

- Xây dựng các hàm dự đoán kết quả từ các mô hình, sử dụng dữ liệu được nhập từ API.

Phía giao diện người dùng (front-end) bao gồm:

- Thiết kế và phát triển giao diện người dùng bằng HTML, CSS và JavaScript để cho phép người dùng nhập thông tin cần thiết.
- Sử dụng JavaScript để gọi các API từ phía máy khách, gửi dữ liệu đến các endpoint API và hiển thị kết quả trả về cho người dùng.

3.3.5. Kết luận về xây dựng phần mềm

Xây dựng phần mềm của đồ án đã tập trung vào triển khai các mô hình thuật toán trên ứng dụng Windows app bằng Python, sử dụng Flask API để tạo API dự đoán phân loại tiếng việt theo chủ đề tương úng và xây dựng giao diện người dùng bằng HTML, CSS, và JavaScript. Phần mềm này nhằm mục đích cung cấp dự đoán kết quả phân loại tiếng việt theo chủ đề nhanh chóng bằng cách chỉ cần đưa đoạn văn bản vào, Việc sử dụng các mô hình học máy như Naive Bayes đảm bảo tính nhất quán và chính xác cao hơn so với việc phân loại bằng tay, nhờ vào khả năng học hỏi từ dữ liệu mẫu và áp dụng các quy tắc một cách không thiên vị.

Đồng thời, việc tích hợp các thành phần back-end và front-end này cũng giúp tạo nên một trải nghiệm ứng dụng toàn diện và thuận tiện cho người dùng cuối.

Tiểu kết:

Trong chương 3 này em đã trình bày chi tiết quá trình phân tích và xử lý dữ liệu, xây dựng mô hình phân loại tiếng Việt và triển khai ứng dụng web để phân loại văn bản theo chủ đề.

KẾT LUẬN

- **Kết quả đạt được**

Trong nghiên cứu này, em đã thu thập tập dữ liệu về các bài viết trên Internet được gán nhãn theo chủ đề, bao gồm các lĩnh vực như thể thao, công nghệ, giáo dục và văn hóa, du lịch, thời sự, giải trí, âm nhạc, thế giới, Xe. Dữ liệu này đã được làm sạch và tiền xử lý một cách kỹ lưỡng, bao gồm loại bỏ stopwords, chuẩn hóa văn bản.

Sau đó, em đã huấn luyện một mô hình Naive Bayes chia tập dữ liệu thành 2 tập (train/test), để tối ưu hóa hiệu suất của mô hình. Tiếp theo, em đã sử dụng kiểm định chéo để đánh giá hiệu suất của mô hình trên các tập dữ liệu kiểm tra khác nhau.

Kết quả cho thấy rằng mô hình Naive Bayes đã đạt được độ chính xác trung bình khoảng 69%, với độ chính xác cân bằng tốt trên các lớp. Ngoài ra, em đã đánh giá kết quả bằng các phép đo precision, recall và F1-score, và thấy rằng mô hình có khả năng phân loại tương đối chính xác các bài viết theo chủ đề một cách hiệu quả.

- **Ưu điểm và nhược điểm**
    - Ưu điểm:

Naive Bayes là tính đơn giản và dễ triển khai. Thuật toán này không đòi hỏi nhiều tài nguyên tính toán và có thể được triển khai nhanh chóng trên các dự án thực tế. Việc tính toán xác suất có điều kiện dựa trên giả định "ngây thơ" giúp giảm bớt độ phức tạp của thuật toán.

Thường cho kết quả tốt trên các tập dữ liệu lớn. Điều này là do thuật toán có thể xử lý các bộ dữ liệu có kích thước lớn mà không gặp vấn đề về hiệu suất. Đồng thời, nó cũng không nhạy cảm với độ dài của văn bản, giúp cho việc phân loại trở nên linh hoạt hơn.

    - Nhược điểm:

Giả định về sự độc lập giữa các đặc trưng có thể không phù hợp với một số tình huống, dẫn đến sự giảm chính xác của mô hình. Ngoài ra, khi dữ liệu thiếu thông tin về một từ hoặc đặc trưng, Naive Bayes có thể gặp phải vấn đề về sự phân loại không chính xác.

Tóm lại, Naive Bayes là một lựa chọn hợp lý cho việc phân loại tiếng Việt theo chủ đề nhờ tính đơn giản và hiệu suất tốt. Tuy nhiên, để đạt được kết quả tốt nhất, việc điều chỉnh và tinh chỉnh tham số cũng như xử lý dữ liệu một cách cẩn thận là cần thiết.

- **Hướng phát triển:**

– Tiếp tục cải thiện hiệu quả mô hình phân loại tiếng Việt

Nghiên cứu áp dụng các phương pháp học máy mới và tối ưu hóa mô hình để nâng cao độ chính xác và khả năng xử lý các ngữ liệu phức tạp. Một số mô hình học máy tiên tiến như Random Forest, Support Vector Machine (SVM), và các mạng nơ-ron sâu (Deep Neural Networks) có thể được xem xét và so sánh với Naive Bayes. Các mô hình này có thể cung cấp độ chính xác cao hơn và khả năng xử lý tốt hơn cho các văn bản có cấu trúc phức tạp và đa dạng.

– Mở rộng ứng dụng phân loại tiếng Việt sang các lĩnh vực khác

Nghiên cứu ứng dụng phân loại tiếng Việt vào các lĩnh vực như phân tích tình cảm, tóm tắt văn bản, chatbot, và các hệ thống gợi ý. Các mô hình học máy như Bi-LSTM, Transformer, và BERT có thể được sử dụng để cải thiện hiệu suất trong các lĩnh vực này. Đặc biệt, mô hình BERT (Bidirectional Encoder Representations from Transformers) đã chứng minh hiệu quả cao trong nhiều bài toán xử lý ngôn ngữ tự nhiên và có thể áp dụng để nâng cao khả năng phân loại văn bản tiếng Việt.

– Tăng cường tính tương tác và giao tiếp với người dùng

Phát triển giao diện người dùng trực quan và dễ sử dụng để người dùng có thể tương tác dễ dàng với hệ thống. Thu thập phản hồi từ người dùng để cải thiện trải nghiệm người dùng và tùy chỉnh các tính năng theo nhu cầu cụ thể của họ. Các công nghệ như ReactJS, AngularJS, và VueJS có thể được sử dụng để xây dựng các giao diện người dùng hiện đại và phản hồi nhanh.

– Phát triển kho dữ liệu tiếng Việt chuẩn

Góp phần xây dựng kho dữ liệu tiếng Việt chuẩn và phong phú để hỗ trợ cho nghiên cứu và phát triển các ứng dụng xử lý ngôn ngữ tiếng Việt. Việc xây dựng và duy trì một kho dữ liệu lớn và đa dạng sẽ giúp cải thiện chất lượng của các mô hình học máy và hỗ trợ cho các nghiên cứu tương lai. Kho dữ liệu này cần bao gồm nhiều thể loại văn bản khác nhau từ các lĩnh vực như báo chí, văn học, pháp luật, và hội thoại hàng ngày.

– Tính ứng dụng và so sánh một số mô hình học máy

Các mô hình học máy như Random Forest, SVM, và các mạng nơ-ron sâu có thể được so sánh với Naive Bayes để tìm ra mô hình phù hợp nhất cho từng ứng dụng cụ thể.

Random Forest: Là một mô hình ensemble, Random Forest kết hợp nhiều cây quyết định để tăng độ chính xác và giảm overfitting. Nó có thể xử lý tốt các dữ liệu không cân bằng và tương tác phức tạp giữa các đặc trưng.

Support Vector Machine (SVM): SVM là một thuật toán mạnh mẽ trong việc phân loại các văn bản có ranh giới rõ ràng giữa các lớp. Tuy nhiên, nó có thể trở nên phức tạp và tốn kém tài nguyên khi xử lý với các tập dữ liệu lớn.

Mạng nơ-ron sâu (Deep Neural Networks): Các mạng nơ-ron sâu như LSTM, GRU, và Transformer đã chứng minh được hiệu quả vượt trội trong nhiều nhiệm vụ xử lý ngôn ngữ tự nhiên. Đặc biệt, mô hình BERT có khả năng hiểu ngữ cảnh sâu sắc và có thể được huấn luyện tiếp tục với dữ liệu tiếng Việt để cải thiện hiệu suất.

DANH MỤC TÀI LIỆU THAM KHẢO

Tiếng Việt:

[1] Slide bài giảng Thầy Trần Đức Minh

[2] Vũ Hữu Tiệp, Machine Learning cơ bản, Vũ Hữu Tiệp, 2020

Tiếng Anh:

[1] Peter Harrington, Machine Learning in Action, Manning, 2012

[2] Tom M.Mitchell, Machine Learning, McGraw-Hill, 1997

Danh mục các Website tham khảo:

[1] https://machinelearningcoban.com/

[2] https://monkeylearn.com/blog/sentiment-analysis-machine-learning/

[3] https://viblo.asia/p/phan-loai-van-ban-tu-dong-bang-machine-learning-nhu-the- nao-4P856Pa1ZY3

[4] https://vinbigdata.com/chatbot/tokenization-la-gi-cac-ky-thuat-tach-tu-trong-xu-ly-ngon-ngu-tu-nhien.html

[5] https://viblo.asia/p/dung-python-3-tinh-xac-suat-thoat-fa-aWj53VRbl6m

[6] https://www.geeksforgeeks.org/naive-bayes-classifiers/

[7] https://www.datacamp.com/tutorial/naive-bayes-scikit-learn