**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHÊ CẦN THƠ**

KHOA CÔNG NGHỆ THÔNG TIN

**CHUYÊN NGÀNH KHOA HỌC MÁY TÍNH KHÓA 21**

**★★★🕮★★★**

**A white circle with blue text and a book and a symbol

Description automatically generated**

ĐỒ ÁN 2

**XÂY DỰNG HỆ THỐNG PHÂN LOẠI CÁC VĂN BẢN THEO CHỦ ĐỀ**

***Cần Thơ, tháng 6, năm 2024***

**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHÊ CẦN THƠ**

KHOA CÔNG NGHỆ THÔNG TIN

**CHUYÊN NGÀNH KHOA HỌC MÁY TÍNH KHÓA 21**

**★★★🕮★★★**

**A white circle with blue text and a book and a symbol

Description automatically generated**

ĐỒ ÁN 2

**XÂY DỰNG HỆ THỐNG PHÂN LOẠI VĂN BẢN THEO CHỦ ĐỀ**

**Giảng viên hướng dẫn**: ThS. NGUYỄN NHỰT QUỲNH

**Sinh viên thực hiện**: LÊ PHƯỚC HỮU 2100450, VÕ VĂN THẠCH 2101389

***Cần Thơ, tháng 6, năm 2024***

**NHẬN XÉT CỦA GIẢNG VIÊN**

**----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

**LỜI CẢM ƠN**

Đầu tiên, em xin gửi lời cảm ơn chân thành đến Trường Đại Học Kỹ Thuật - Công Nghệ Cần Thơ đã tạo điều kiện đưa các nghành công nghệ vào trương trình giảng dạy. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn cô Nguyễn Nhựt Quỳnh đã giúp đỡ, truyền đạt những kiến thức quý báu cho em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia học tập chúng em đã được tiếp thu những kiến thức quan trọng, mới mẻ của nghành công nghệ, tinh thần học tập hiệu quả, nghiêm túc. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để em có thể vững bước sau này.

Đề tài “Xây dựng hệ thống phân loại văn bản theo chủ đề ” là đề tài vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên. Tuy nhiên, do vốn kiến thức còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều bỡ ngỡ. Mặc dù em đã cố gắng hết sức nhưng chắc chắn bài tiểu luận khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong cô xem xét và góp ý để bài tiểu luận của em được hoàn thiện hơn.

**LỜI CAM ĐOAN**

Nhóm chúng em xin giới thiệu với cô và mọi người đề tài “Xây dựng hệ thống phân loại văn bản theo chủ đề”. Chúng em chọn đề tài này vì sự thiết thực và bổ ích cho các bạn sinh viên trong học tập cũng như trong môi trường làm việc với máy tính trở nên thông dụng như hiện nay.

Trong quá trình thực hiện đề tài này còn có nhiều thiếu sót do kiến thức còn sơ sài nhưng những nội dung trình bày trong quyển báo cáo này là những biểu hiện kết quả của chúng em đạt được dưới sự hướng dẫn của Cô Nguyễn Nhựt Quỳnh.

Chúng em xin cam đoan rằng: Những nội dung trình bày trong quyển báo cáo tiểu luận này không phải là bản sao chép từ bất kì tiểu luận nào có trước. Nếu không đúng sự thật, chúng em xin chịu mọi trách nhiệm trước hội đồng cố vấn.

Giảng viên hướng dẫn Sinh viên thực hiện

*A close up of a sign

Description automatically generated* *(Ký tên)*  *(Ký tên)*

**MỤC LỤC**

[**PHẦN GIỚI THIỆU 1**](#_Toc169207258)

[**1. GIỚI THIỆU TỔNG QUAN VÀ LÝ DO CHỌN ĐỀ TÀI 1**](#_Toc169207259)

[**2. NGHIÊN CỨU LIÊN QUAN 2**](#_Toc169207260)

[**3. MỤC TIÊU VÀ PHẠM VI NGHIÊN CỨU 4**](#_Toc169207261)

[**4. ĐỐI TƯỢNG NGHIÊN CỨU 4**](#_Toc169207262)

[**5. PHƯƠNG PHÁP NGHIÊN CỨU 5**](#_Toc169207263)

[**6. BỐ CỤC ĐỒ ÁN 5**](#_Toc169207264)

[**PHẦN NỘI DUNG 6**](#_Toc169207265)

[**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 6**](#_Toc169207266)

[**1.1 HỌC MÁY 6**](#_Toc169207267)

[**1.2 XỬ LÝ NGÔN NGỮ TỰ NHIÊN 19**](#_Toc169207268)

[**1.3 PHÂN LOẠI VĂN BẢN 25**](#_Toc169207269)

[**1.4 KỸ THUẬT TF – IDF 30**](#_Toc169207270)

[**1.5 MÔ HÌNH NAIVE BAYES 33**](#_Toc169207271)

[**CHƯƠNG 2: PHƯƠNG PHÁP THỰC HIỆN 37**](#_Toc169207272)

[**2.1 GIỚI THIỆU DEMO 37**](#_Toc169207273)

[**2.2 MÔI TRƯỜNG VÀ CÁC ĐỐI TƯỢNG SỬ DỤNG XÂY DỰNG DEMO 37**](#_Toc169207274)

[**2.3 CÁC BƯỚC THỰC HIỆN 38**](#_Toc169207275)

[**CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ 47**](#_Toc169207276)

[**3.1 KẾT QUẢ THỰC NGHIỆM 47**](#_Toc169207277)

[**3.2 ĐÁNH GIÁ 49**](#_Toc169207278)

[**PHẦN KẾT LUẬN 50**](#_Toc169207279)

[**1. KẾT LUẬN 50**](#_Toc169207280)

[**2. HƯỚNG PHÁT TRIỂN 50**](#_Toc169207281)

[**TÀI LIỆU KHAM KHẢO 51**](#_Toc169207282)

**DANH MỤC HÌNH ẢNH**

[**Hình 1. 1 Ứng dụng học máy vào lĩnh vực chăm sóc sức khỏe 7**](#_Toc169208314)

[**Hình 1. 2 Tiến trình phát triển mô hình học máy cơ bản 13**](#_Toc169208315)

[**Hình 1. 3 Mô phỏng cơ bản về mô hình mạng nơ-ron 13**](#_Toc169208316)

[**Hình 1. 4 Minh họa hồi quy tuyến tính 14**](#_Toc169208317)

[**Hình 1. 5 Mô phỏng ứng dụng Desision tree 16**](#_Toc169208318)

[**Hình 1. 6 Mô hình cơ bản của kỹ thuật Naive Beyes 18**](#_Toc169208319)

[**Hình 1. 7 Mô hình tổng quát phân loại văn bản 29**](#_Toc169208320)

**TÓM TẮT**

Trong những năm gần đây các hệ thống giúp phân loại văn bản ngày càng phát triển để phục vụ nhu cầu cần thiết cho người dùng, với độ chính xác cao cùng với tính ổn định đã giúp cho các doanh nghiệp tiết kiệm được thời gian rất lớn cho việc quản lí và phân loại dữ liệu phục vụ cho công việc.

Hệ thống phân loại văn bản theo chủ đề là một công nghệ được thiết kế để tự động phân loại văn bản thành các chủ đề khác nhau dựa trên nội dung giúp người dùng có được dữ liệu cần thiết mà không tốn quá nhiều thời gian.

Hệ thống phân loại văn bản theo chủ đề có vai trò quan trọng trong việc tổ chức, tìm kiếm và trích xuất thông tin từ các tài liệu văn bản lớn giúp tối ưu quá trình làm việc và quản lí dữ liệu trong nhiêu lĩnh vực khác nhau. Nhận thấy đây là một đề tài ứng dụng thực tiễn cao cùng với mong muốn tìm tòi học hỏi nên nhóm chúng em quyết định chọn đề tài “Xây dựng hệ thống phân loại văn bản theo chủ đề” cho đồ án lần này.

**ABSTRACT**

In recent years, document classification systems have been increasingly developed to meet the growing needs of users. With high accuracy and stability, these systems have helped us and businesses save a significant amount of time in data management and classification for work purposes.

A text classification system is designed to automatically categorize documents into different topics based on their content, enabling users to access necessary data without consuming excessive time.

Text classification systems play a crucial role in organizing, searching, and extracting information from large document repositories, optimizing work processes and data management in various fields. Recognizing its practical application and with a desire for exploration and learning, our group has decided to choose the topic "Building a text classification system by topic" for this project.

# **PHẦN GIỚI THIỆU**

1. **GIỚI THIỆU TỔNG QUAN VÀ LÝ DO CHỌN ĐỀ TÀI**

Với xu hướng phát triển của thế giới các ngành công nghệ đã có những thay đổi bước tiến vượt bật và là một trong những lĩnh vực được ưu tiên phát triển hàng đầu trong đó trí tuệ nhân tạo đóng góp một phần rất lớn tạo nên cuộc thành công về chuyển đổi số hóa trong xã hội giúp con người dễ dàng tiếp cận với những thông tin trong đời sống, ngoài ra trong doanh nghiệp, y học, giáo dục, truyền thông.... các công nghệ dược ứng dụng một cách rất tiến bộ.

Trong thời đại internet ngày nay, dữ liệu ngày càng trở nên phong phú và đa dạng, đặc biệt trong bối cảnh của cuộc cách mạng công nghiệp 4.0. Cùng với sự bùng nổ thông tin, nhu cầu phân loại và khai thác thông tin trở nên ngày càng quan trọng trong mọi tổ chức và quy mô hoạt động. Khi dữ liệu văn bản được tạo ra và chia sẻ một cách rộng rãi. Các công ty, tổ chức và cá nhân thường phải đối mặt với một lượng lớn các văn bản, bao gồm email, tài liệu, bài báo, tin tức, và nhiều loại tài liệu khác. Việc phân loại tự động các văn bản có thể giúp tối ưu hóa quy trình làm việc và quản lý thông tin.

Phân loại văn bản theo chủ đề đóng vai trò quan trọng trong việc tổ chức và xử lý thông tin trong thế giới số ngày nay. Trong một thế giới mà thông tin phát triển không ngừng và dữ liệu văn bản ngày càng lớn, việc tự động gán nhãn cho các văn bản dựa trên nội dung của chúng là một yếu tố không thể phủ nhận. Qua quá trình này, có thể tổ chức và tìm kiếm thông tin một cách hiệu quả hơn, từ việc phân loại email, quản lý tài liệu đến việc tổ chức dữ liệu trên internet.

Trong lĩnh vực thương mại, việc phân loại tin tức, phản hồi khách hàng và phân tích thị trường dựa trên nội dung văn bản có thể giúp các doanh nghiệp hiểu rõ hơn về nguyên nhân và kỳ vọng của khách hàng. Trong lĩnh vực y tế, phân loại văn bản có thể được sử dụng để phát hiện các triệu chứng, phân loại bệnh lý và dự đoán xu hướng dịch bệnh. Trong lĩnh vực tin tức và truyền thông, việc phân loại tin tức theo chủ đề giúp người đọc dễ dàng truy cập và tiếp cận thông tin một cách nhanh chóng và thuận tiện.

Nhận thấy đây là một chủ đề thực tế và có tính ứng dụng cao nên chúng em quyết định tìm hiểu về đề tài “XÂY DỰNG HỆ THỐNG PHÂN LOẠI VĂN BẢN THEO CHỦ ĐỀ” và sử dụng mô hình Naive Bayes để hổ trợ cho việc phân loại văn bản một cách nhanh chóng cũng như độ ổn định cao.

1. **NGHIÊN CỨU LIÊN QUAN**
   1. **Công trình nghiên cứu "Attention is All You Need"**

Công trình nghiên cứu "Attention is All You Need" của Vaswani et al., được công bố vào năm 2017, giới thiệu một kiến trúc mạng nơ-ron mới gọi là Transformer, làm thay đổi cách xử lý dữ liệu chuỗi trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) và nhiều lĩnh vực khác.

Transformer giải quyết vấn đề của các kiến trúc trước đây như LSTM (Long Short-Term Memory) và GRU (Gated Recurrent Unit) bằng cách thay đổi cách mà mạng nơ-ron xử lý thông tin trên dữ liệu chuỗi. Thay vì sử dụng các lớp RNN (Recurrent Neural Network) để duy trì trạng thái ẩn qua các thời điểm, Transformer sử dụng các cơ chế attention để tập trung vào các phần quan trọng của dữ liệu đầu vào. Transformer sử dụng cơ chế chú ý tự (self-attention) để tính toán sự quan trọng của từng cặp từ trong câu. Điều này cho phép mô hình "nhìn" vào tất cả các từ đầu vào cùng một lúc, giúp nắm bắt được các mối quan hệ phi tuyến tính phức tạp giữa các từ. Transformer sử dụng nhiều "đầu" attention song song. Mỗi đầu attention tạo ra một biểu diễn khác nhau của dữ liệu đầu vào, sau đó kết hợp lại để tạo ra đầu ra cuối cùng. Điều này giúp mô hình tập trung vào các khía cạnh khác nhau của dữ liệu đầu vào.

* 1. **BERT (Bidirectional Encoder Representations from Transformers)**

Là một công trình nghiên cứu quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), được giới thiệu bởi Jacob Devlin và các đồng nghiệp tại Google AI Language vào năm 2018. BERT đã đạt được những tiến bộ đáng kể trong việc hiểu ngôn ngữ tự nhiên và thực hiện nhiều tác vụ NLP với hiệu suất cao.

Một trong những điểm mạnh của BERT là khả năng hiểu bản chất của từ ngữ trong ngữ cảnh. Không giống như các mô hình trước đó như GPT (Generative Pre-trained Transformer) chỉ sử dụng thông tin từ bên trái, BERT sử dụng cả hai hướng (trái và phải) của câu để tạo ra biểu diễn từ hiệu quả hơn.

BERT sử dụng kiến trúc Transformer, nhưng thay vì sử dụng cho nhiệm vụ sinh văn bản như GPT, BERT được huấn luyện để tạo ra biểu diễn từ cho các nhiệm vụ như phân loại văn bản, trích xuất thông tin, và nhiều tác vụ NLP khác.

BERT được huấn luyện trước trên một lượng lớn dữ liệu văn bản không có giám sát từ các nguồn như Wikipedia. Sau đó, có thể được tinh chỉnh (fine-tuning) trên tập dữ liệu nhỏ hơn cho các tác vụ cụ thể. Quá trình này giúp cải thiện khả năng tổng quát hóa của mô hình.

* 1. **GPT (Generative Pre-trained Transformer)**

Là một dòng mô hình học sâu được huấn luyện trước dựa trên kiến trúc Transformer trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). GPT được giới thiệu bởi OpenAI, với phiên bản đầu tiên GPT-1 được công bố vào năm 2018, sau đó là GPT-2 và GPT-3. Mỗi phiên bản cải tiến và mở rộng khả năng của mô hình.

GPT sử dụng kiến trúc mạng nơ-ron Transformer, cho phép tập trung vào các phần quan trọng của dữ liệu đầu vào thông qua cơ chế attention. Trước khi được sử dụng cho các nhiệm vụ cụ thể, GPT được huấn luyện trước (pre-trained) trên một lượng lớn dữ liệu văn bản không giám sát từ Internet. Quá trình này giúp mô hình hiểu được ngữ cảnh của từng từ và cấu trúc của văn bản.

GPT có khả năng sinh ra văn bản tự nhiên, có ý nghĩa và logic dựa trên ngữ cảnh. Điều này làm cho GPT trở thành một công cụ mạnh mẽ cho các ứng dụng như tạo văn bản tự động, phản hồi tự động trong chatbot và nhiều ứng dụng sáng tạo khác.

Phiên bản GPT-3, trong đó số lượng tham số lên đến hàng triệu, là một trong những mô hình lớn nhất được công bố cho đến nay trong lĩnh vực NLP. Sự lớn mạnh này cung cấp khả năng xử lý ngôn ngữ tự nhiên với độ phức tạp cao và có thể giải quyết nhiều nhiệm vụ NLP khác nhau.

* 1. **Text Classification Algorithms: A Survey" (2018)**

Là một công trình nghiên cứu quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) và học máy, được viết bởi Ahmad Majid Tavana và Masoud Mohammadian. Bài báo này là một tổng quan toàn diện về các phương pháp và thuật toán phổ biến nhất được sử dụng trong phân loại văn bản, một trong những tác vụ cơ bản nhất của NLP.

Bài báo cung cấp một tổng quan về khái niệm và ý nghĩa của phân loại văn bản, một tác vụ quan trọng trong xử lý ngôn ngữ tự nhiên. Nói về ứng dụng của phân loại văn bản trong các lĩnh vực như phân tích cảm xúc, phát hiện spam, phân loại tin tức, và nhiều ứng dụng khác. Bài báo đề cập đến các phương pháp và thuật toán phổ biến được sử dụng trong phân loại văn bản, bao gồm các phương pháp cổ điển như Naive Bayes và SVM (Support Vector Machines), cũng như các phương pháp hiện đại như Random Forest, Gradient Boosting, và Deep Learning. Cuối cùng, bài báo đề xuất các hướng phát triển tiềm năng trong lĩnh vực phân loại văn bản, bao gồm việc kết hợp các phương pháp và thuật toán khác nhau, nghiên cứu về các kỹ thuật mới như học sâu và học tăng cường, và nghiên cứu về việc áp dụng phân loại văn bản trong các lĩnh vực mới.

1. **MỤC TIÊU VÀ PHẠM VI NGHIÊN CỨU**
2. **Mục tiêu**

Mục tiêu nghiên cứu là xây dựng được hệ thống phân loại văn bản theo chủ đề sử dụng mô hình Naive Bayes.

Năm vững các lý thuyết về mô hình Naive Bayes, trình bày được cách hoạt động của mô hình.

1. **Phạm vi nghiên cứu**

Tập trung nghiên cứu sâu về định lý Bayes và mô hình phân loại Naive Bayes. Nghiên cứu cách trích xuất đặc trưng từ bộ dữ liệu, vector hóa các đặc trưng đó bằng kỹ thuật TF – IDF.

Thực hiện thu thập dữ liệu phù hợp với đề tài nghiên cứu, bộ dữ liệu sẽ giới hạn trong năm chủ đề, cụ thể là các chủ đề sau:

* Entertainment
* Sport
* Business
* Politics
* Tech

1. **ĐỐI TƯỢNG NGHIÊN CỨU**
   * Các lí thuyết cơ bản về Machine Learning, NLP, Naive Bayes, TF-IDF.
   * Quy trình máy học áp dụng vào hệ thống phân loại văn bản theo chủ đề.
   * Cách xây dựng mô hình cũng như các hoạt động của Naive Bayes.
2. **PHƯƠNG PHÁP NGHIÊN CỨU**

Các phương pháp nghiên cứu được áp dụng trong hệ thống phân loại văn bản theo chủ đề:

* Thu thập tài liệu từ các nguồn chính thống để tìm hiểu về Machine Learning, NLP, TF-IDF.
* Mô hình xử lý trong máy học cũng như mô hình Naive Bayes.
* Phân tích, xử lí dữ liệu và xử lí ngôn ngữ tự nhiên.
* Phương pháp đánh giá mô hình đã xây dựng.

1. **BỐ CỤC ĐỒ ÁN**

**Chương 1**: Giới thiệu tổng quan và lý do chọn đề tài, mục tiêu và phạm vi nghiên cứu, đối tượng và phương pháp nghiên cứu, bố cục của đồ án.

**Chương 2**: Giới thiệu về NLP, Machine Learning, Naïve Bayes, TF-IDF.

**Chương 3**: Thu thập dữ liệu, xử lý dữ liệu, huấn luyện , đánh giá mô hình.

**Chương 4**: Tổng quan kết quả đạt được và kết quả thực tế.

**Chương 5**: Kết luận và hướng phát triển.

# **PHẦN NỘI DUNG**

## **CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

1. **HỌC MÁY**
2. **Định nghĩa về học máy**

Học máy là thuật ngữ chỉ các hành động dạy máy tính để cải thiện một nhiệm vụ mà đang thực hiện. Cụ thể khả năng của học máy là sử dụng các thuật toán để phân tích những thông tin sẵn có, sau đó học hỏi và đưa ra những quyết định, dự đoán về những thứ có liên quan. Thay vì phải lập trình ra một phần mềm với các hành động, hướng dẫn chi tiết để thực hiện một nhiệm vụ thì máy tính sẽ được học cách thực hiện các nhiệm vụ đó thông qua một lượng dữ liệu và các thuật toán.

Học máy đang trở thành một yếu tố quan trọng trong lĩnh vực khoa học máy tính đang phát triển. Bằng cách sử dụng các phương pháp thống kê và đào tạo các thuật toán, học máy có khả năng cung cấp các phân loại hoặc dự đoán và khám phá thông tin chi tiết từ những dự án khai thác dữ liệu.

Sử dụng thông tin chuyên sâu để thúc đẩy các quyết định về ứng dụng và kinh doanh cũng như tác động đến các chỉ số tăng trưởng. Khi dữ liệu lớn mở rộng và phát triển, nhu cầu về các nhà khoa học máy tính cũng tăng theo. Họ được yêu cầu giúp xác định các câu hỏi kinh doanh chính và thông tin cần thiết để trả lời chúng.

Thường thì các bài toán trong học máy được phân thành hai loại chính: dự đoán (prediction) và phân loại (classification). Các bài toán dự đoán thường liên quan đến việc ước lượng giá trị, như giá nhà, giá xe, trong khi các bài toán phân loại thường liên quan đến việc xác định danh tính hoặc loại của một đối tượng, như nhận diện chữ viết tay, đồ vật, và các nhiệm vụ tương tự.

A diagram of a machine learning application

Description automatically generated

Hình 1. 1 Ứng dụng học máy vào lĩnh vực chăm sóc sức khỏe

1. **Lịch sử hình thành của học máy**

Lịch sử hình thành của học máy là một hành trình kéo dài từ những năm đầu của thế kỷ 20 đến hiện tại, với nhiều bước tiến và tiến bộ quan trọng. Mỗi giai đoạn đều có những bước phát triển khác nhau nhưng chung quy lại là phát triển nền tảng lý thuyết và thực tiễn vững chắc.

* **Thập niên 1950 – 1960**

Trong thập niên 1950 đến 1960, các ý tưởng đầu tiên về học máy bắt đầu hình thành, đặc biệt là trong lĩnh vực trí tuệ nhân tạo (*AI*). Alan Turing, một nhà toán học và nhà máy tính nổi tiếng, đã đề xuất "máy Turing", một khái niệm cho máy tính có khả năng học. Ông đặt ra câu hỏi liệu một máy tính có thể giả lập được trí tuệ của con người hay không, thông qua trò chơi Imitation Game, sau này được biết đến như là test Turing.

Trong cùng thời kỳ, Marvin Minsky và các nhà khoa học khác đã đề xuất các mô hình mới gọi là "mạng nơ-ron nhân tạo". Các mạng nơ-ron này được lấy cảm hứng từ cách mà não người hoạt động, trong đó các "nơ-ron" nhân tạo được kết nối với nhau để xử lý thông tin. Mặc dù các mô hình ban đầu rất đơn giản và hạn chế, nhưng chúng đã tạo ra nền tảng cho sự phát triển sau này của học máy.

Những ý tưởng và nghiên cứu trong thập niên này đã đặt nền móng cho sự tiến bộ vượt bậc trong lĩnh vực học máy trong các thập kỷ tiếp theo. Đồng thời, cũng đã mở ra cánh cửa cho những khám phá và ứng dụng mới trong trí tuệ nhân tạo và các lĩnh vực liên quan.

* **Thập niên 1970 – 1980**

Sau những bước đầu tiên trong thập kỷ 1950 và 1960, sự phát triển của học máy và các thuật toán liên quan tiếp tục diễn ra mạnh mẽ. Các nhà nghiên cứu và nhà khoa học đã tiếp tục khám phá và phát triển các phương pháp mới như học tăng cường (*reinforcement learning*) và cây quyết định (*decision trees*).

Học tăng cường là một phương pháp trong học máy mà một hệ thống tự động học qua thời gian từ những tương tác với môi trường bằng cách thử và sai. Trong khi đó, cây quyết định là một phương pháp học có giám sát, trong đó một loạt các quyết định dựa trên các đặc trưng của dữ liệu được sử dụng để dự đoán hoặc phân loại. Cả hai phương pháp này đều đã được áp dụng rộng rãi trong các lĩnh vực như xử lý ngôn ngữ tự nhiên và thị giác máy tính.

Trong xử lý ngôn ngữ tự nhiên, học máy được sử dụng để xây dựng các hệ thống nhận diện và tự động xử lý văn bản, từ việc phân loại email vào hòm thư rác đến dịch tự động và phân tích cảm xúc trong văn bản. Các thuật toán như học tăng cường có thể được sử dụng để cải thiện hiệu suất của các hệ thống dịch máy thông qua việc tối ưu hóa các chiến lược dịch.

Những tiến bộ trong học máy và các thuật toán liên quan không chỉ mở ra những cơ hội mới trong nhiều lĩnh vực ứng dụng, mà còn đóng góp vào sự tiến bộ chung của trí tuệ nhân tạo và công nghệ máy tính.

* **Thập niên 1980 – 1990**

Sự phát triển của các thuật toán học máy trở nên phổ biến hơn, đặc biệt là trong lĩnh vực học tăng cường và học sâu (*deep learning*). Một số công nghệ và thuật toán như mạng nơ-ron nhân tạo lan truyền ngược (*backpropagation*) và máy vector hỗ trợ (*support vector machines*) được phát triển.

* **Thập niên 1990 – 2000**

Các nhà nghiên cứu tập trung vào việc tối ưu hóa và cải tiến các thuật toán học máy hiện có, cũng như sự phát triển của các phương pháp mới như học tập tập trung (*ensemble learning*).

* **Thập niên 2000 – 2010**

Sự phát triển nhanh chóng của internet và dữ liệu kỹ thuật số đã tạo ra điều kiện thuận lợi cho sự bùng nổ của học máy. Với sự gia tăng đáng kể về lượng dữ liệu được tạo ra và thu thập từ các nguồn như mạng xã hội, thiết bị di động, cảm biến và các hệ thống thông minh, khối lượng lớn dữ liệu được gọi là Big Data đã trở thành một nguồn tài nguyên quý giá cho học máy.

Big Data không chỉ đem lại lượng dữ liệu lớn, mà còn đặc biệt về độ phức tạp và đa dạng. Điều này thúc đẩy sự phát triển của các phương pháp xử lý dữ liệu lớn như Hadoop, Spark và các công nghệ lưu trữ dữ liệu phân tán khác. Những công nghệ này cung cấp khả năng xử lý và phân tích dữ liệu với tốc độ và quy mô mà trước đây là không thể.

Sự kết hợp giữa Big Data và các phương pháp học máy đã tạo ra nền tảng cho việc áp dụng học máy vào nhiều lĩnh vực khác nhau. Trong tài chính, học máy được sử dụng để phân tích dữ liệu tài chính và dự đoán xu hướng thị trường. Trong y tế, mô hình học máy có thể được sử dụng để dự đoán và chuẩn đoán bệnh, cải thiện chăm sóc sức khỏe và dự đoán kết quả điều trị. Trong marketing, học máy giúp tối ưu hóa chiến lược quảng cáo, phân tích hành vi người tiêu dùng và tăng cường trải nghiệm khách hàng.

Với sức mạnh của Big Data và học máy, các tổ chức và doanh nghiệp có thể nắm bắt và phân tích thông tin từ dữ liệu để đưa ra các quyết định chiến lược thông minh và tạo ra giá trị cho doanh nghiệp. Sự kết hợp này không chỉ mang lại lợi ích kinh tế mà còn góp phần vào sự phát triển toàn diện của xã hội và kinh tế số.

* **Thập niên 2010 - hiện tại**

Sự phát triển của deep learning đánh dấu một bước tiến lớn trong lĩnh vực học máy. Công nghệ này đã mang lại những tiến bộ đáng kể trong việc xử lý dữ liệu phức tạp và tạo ra các mô hình dự đoán chính xác hơn. Deep learning, dựa trên các mạng nơ-ron sâu, đã trở thành một trong những công nghệ quan trọng nhất trong lĩnh vực học máy và trí tuệ nhân tạo.

Công nghệ deep learning cho phép mạng nơ-ron có khả năng học từ dữ liệu không có cấu trúc hoặc dữ liệu phức tạp mà không cần định nghĩa rõ ràng các đặc trưng. Điều này làm cho các mô hình học máy dựa trên deep learning trở nên mạnh mẽ hơn và có khả năng xử lý các nhiệm vụ phức tạp như nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, và dự đoán.

Các công ty công nghệ hàng đầu như Google, Facebook, và Microsoft đã nhận ra tiềm năng của deep learning và tiếp tục đầu tư mạnh mẽ vào nghiên cứu và phát triển các ứng dụng của học máy và trí tuệ nhân tạo. Các ứng dụng của deep learning không chỉ giúp cải thiện trải nghiệm người dùng trên các nền tảng trực tuyến mà còn có ứng dụng rộng rãi trong nhiều lĩnh vực như y tế, tài chính, ô tô tự hành, và nhiều lĩnh vực khác.

Với sự phát triển nhanh chóng của deep learning, dự đoán và ứng dụng của học máy và trí tuệ nhân tạo sẽ tiếp tục mở ra những cơ hội mới và tạo ra những tiến bộ đáng kể trong tương lai. Đây là một phần quan trọng của cuộc cách mạng số và sẽ tiếp tục đóng góp vào sự phát triển toàn diện của xã hội và kinh tế.

1. **Các mô hình học máy cơ bản**
2. **Học có giám sát**

Học máy có giám sát, hay còn gọi là học có giám sát, là phương pháp sử dụng các tập dữ liệu được gắn nhãn để huấn luyện các thuật toán phân loại dữ liệu hoặc dự đoán kết quả một cách chính xác. Mô hình được điều chỉnh trọng lượng khi dữ liệu đầu vào được đưa vào, trong quá trình xác nhận chéo để tránh tình trạng quá mức hoặc thiếu thông tin.

Học máy có giám sát hỗ trợ tổ chức giải quyết nhiều vấn đề thực tế quy mô lớn, như phân loại thư rác trong hộp thư đến cá nhân.

Các phương pháp thường được sử dụng trong học máy có giám sát bao gồm: logistic regression, neural networks, linear regression, naive bayes, random forest và support vector machine.

1. **Học không giám sát**

Học máy không giám sát là phương pháp sử dụng các thuật toán máy học để phân tích và phân cụm các tập dữ liệu không được gắn nhãn. Không yêu cầu sự can thiệp của con người, các thuật toán này có khả năng phát hiện mẫu hoặc nhóm dữ liệu ẩn, làm cho chúng lý tưởng cho việc phân tích dữ liệu khám phá và nhận diện hình ảnh và mẫu.

Học máy không giám sát cũng được sử dụng để giảm số lượng tính năng trong một mô hình thông qua quá trình giảm kích thước, với phương pháp như phân tích thành phần chính và phân tích giá trị đơn lẻ.

Các thuật toán thường được sử dụng trong học máy không giám sát bao gồm: k-means clustering, neural networks và các phương pháp phân cụm xác suất.

1. **Học bán giám sát**

Học máy bán giám sát cung cấp một phương pháp hiệu quả giữa học máy có giám sát và không giám sát. Trong quá trình đào tạo, sử dụng một tập dữ liệu có nhãn nhỏ hơn để hướng dẫn phân loại và trích xuất tính năng từ một tập dữ liệu lớn hơn, không được gắn nhãn. Phương pháp này có thể giải quyết vấn đề thiếu dữ liệu được gắn nhãn cho thuật toán học có giám sát và làm giảm chi phí đánh nhãn đối với dữ liệu.

1. **Học tăng cường**

Học tăng cường là kỹ thuật máy học giúp đào tạo phần mềm đưa ra quyết định nhằm thu về kết quả tối ưu nhất. Kỹ thuật này bắt chước quy trình học thử và sai mà con người sử dụng để đạt được mục tiêu đã đặt ra. Học tăng cường giúp phần mềm tăng cường các hành động hướng tới mục tiêu, đồng thời bỏ qua các hành động làm xao lãng mục tiêu.

Thuật toán học tăng cường sử dụng mô hình khen thưởng và trừng phạt trong quy trình xử lý dữ liệu. Các thuật toán này tiếp thu ý kiến phản hồi của từng hành động và tự khám phá ra con đường xử lý tốt nhất để thu về kết quả cuối cùng. Thuật toán học tăng cường còn có khả năng trì hoãn khen thưởng. Chiến lược tổng thể tốt nhất có thể đòi hỏi phải đánh đổi một vài lợi ích trước mắt, vì vậy cách tiếp cận tốt nhất mà học tăng cường khám phá ra có thể bao gồm một số trừng phạt hoặc giai đoạn quay lui. Mô hình học tăng cường là phương pháp hiệu suất cao giúp hệ thống trí tuệ nhân tạo đạt kết quả tối ưu trong môi trường chưa biết.

1. **Quy trình hoạt động của học máy**

Hiện nay có khá nhiều quy trình học máy được nghiên cứu và phát triển, để có một mô hình hoạt động không gặp nhiều vấn đề thì cần có một quy trình được xác định trước đó, nhưng chung quy lại có các bước chính không thể trách tách rời trong quy trình xây dựng và phát triển mô hình, các bước được xác định như sau: Data collection**,** Preparing data, Training model**,** Evaluating model**,** Improving model.

* **Data collection**

Để máy tính có thể học được cần có một bộ dữ liệu (dataset), có thể tự thu thập chúng hoặc lấy các bộ dữ liệu đã được công bố trước đó. Phải thu thập từ nguồn chính thống, có như vậy dữ liệu mới chính xác và máy có thể học một cách đúng đắng và đạt hiệu quả cao hơn.

* Preparing data

Bước này dùng để chuẩn hóa dữ liệu, loại bỏ các thuộc tính không cần thiết, gán nhãn dữ liệu, mã hóa một số đặc trưng, trích xuất đặc trưng, rút gọn dữ liệu nhưng vẫn đảm bảo kết quả… Bước này tốn thời gian nhất tỉ lệ thuận với số lượng dữ liệu. Bước 1 và 2 thường chiếm hơn 70% tổng thời gian thực hiện.

* **Training model**

Bước này là bước huấn luyện cho mô hình hay chính là cho máy tính học trên dữ liệu đã thu thập và xử lý ở hai bước đầu.

* **Evaluating model**

Sau khi đã huấn luyện mô hình xong, cần dùng các độ đo để đánh giá mô hình, tùy vào từng độ đo khác nhau mà mô hình cũng được đánh giá tốt hay không khác nhau. Độ chính xác của mô hình đạt trên 80% được cho là tốt.

* Improving model

Sau khi đã đánh giá mô hình, các mô hình đạt độ chính xác không tốt thì cần được train lại, mô hình sẽ lặp lại từ bước 3, cho đến khi đạt độ chính xác như kỳ vọng. Tổng thời gian của 3 bước cuối rơi vào khoảng 30% tổng thời gian thực A blue squares with white text

Description automatically generatedhiện

Hình 1. 2 Tiến trình phát triển mô hình học máy cơ bản

1. **Các thuật toán học máy cơ bản**
2. **Mạng nơ – ron nhân tạo**

Mạng nơ – ron, còn được gọi là mạng thần kinh nhân tạo (ANN) hoặc mạng thần kinh mô phỏng (SNN), là một tập hợp con của học máy.

Mạng nơ – ron nhân tạo là một chuỗi các thuật toán cố gắng nhận ra các mối quan hệ cơ bản trong một tập hợp dữ liệu thông qua một quá trình bắt chước cách thức hoạt động của bộ não con người. Theo nghĩa này, mạng lưới thần kinh đề cập đến các hệ thống tế bào thần kinh, có bản chất hữu cơ hoặc nhân tạo.

A diagram of a triangle with blue circles and black lines

Description automatically generatedNeural networks là thuật toán được dùng trong việc nhận dạng các mẫu và đóng một vai trò quan trọng trong các ứng dụng bao gồm dịch ngôn ngữ tự nhiên, nhận dạng hình ảnh, nhận dạng giọng nói và tạo hình ảnh.

Hình 1. 3 Mô phỏng cơ bản về mô hình mạng nơ-ron

1. **Hồi quy tuyến tính**

Hồi quy tuyến tính là một loại phân tích thống kê được sử dụng để dự đoán mối quan hệ giữa hai biến. Giả định mối quan hệ tuyến tính giữa biến độc lập và biến phụ thuộc và nhằm mục đích tìm ra đường phù hợp nhất mô tả mối quan hệ. Đường này được xác định bằng cách giảm thiểu tổng bình phương chênh lệch giữa giá trị dự đoán và giá trị thực tế.

A diagram of a line of data points

Description automatically generatedHồi quy tuyến tính thường được sử dụng trong nhiều lĩnh vực, bao gồm kinh tế, tài chính và khoa học xã hội, để phân tích và dự đoán xu hướng trong dữ liệu. Cũng có thể được mở rộng thành hồi quy tuyến tính bội, trong đó có nhiều biến độc lập và hồi quy logistic, được sử dụng cho các vấn đề phân loại nhị phân.

Hình 1. 4 Minh họa hồi quy tuyến tính

1. **Hồi quy logistic**

Hồi quy logistic là một kỹ thuật phân tích dữ liệu sử dụng toán học để tìm ra mối quan hệ giữa hai yếu tố dữ liệu. Sau đó, kỹ thuật này sử dụng mối quan hệ đã tìm được để dự đoán giá trị của những yếu tố đó dựa trên yếu tố còn lại. Dự đoán thường cho ra một số kết quả hữu hạn, như có hoặc không.

Giả sử muốn đoán xem khách truy cập trang web sẽ nhấp vào nút thanh toán trong giỏ hàng của họ hay không. Phân tích hồi quy logistic xem xét hành vi của khách truy cập trước đây, chẳng hạn như thời gian dành cho trang web và số lượng các mặt hàng trong giỏ hàng. Quá trình phân tích này xác định rằng, trước đây, nếu khách truy cập dành hơn năm phút trên trang web và thêm hơn ba mặt hàng vào giỏ hàng, họ sẽ nhấp vào nút thanh toán. Nhờ vào thông tin này, sau đó, hàm hồi quy logistic có thể dự đoán hành vi của một khách mới truy cập trang web.

1. **Clustering**

Clustering trong học máy là một phương pháp phân loại dữ liệu không được gán nhãn vào các nhóm khác nhau dựa trên sự tương đồng giữa chúng. Mục tiêu của việc clustering là tìm ra cấu trúc ẩn trong dữ liệu mà không cần có thông tin gắn nhãn trước.

Quá trình clustering bắt đầu bằng việc chọn một số lượng nhóm hoặc các trung tâm khởi đầu, sau đó dữ liệu sẽ được gán vào các nhóm tương ứng dựa trên sự tương đồng giữa chúng. Sự tương đồng thường được đo lường bằng các phép đo như khoảng cách Euclid, khoảng cách Mahalanobis, hoặc hàm tương tự như cosine similarity, tùy thuộc vào bối cảnh và đặc điểm của dữ liệu. Có nhiều phương pháp clustering khác nhau có thể kể đến như: K-means, Hierarchical Clustering, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), Mean Shift.

1. **Decision tree**

Decision tree hay được gọi là cây quyết định, trong học máy là một phương pháp học có giám sát được sử dụng để phân loại và dự đoán các giá trị của biến mục tiêu bằng cách học các quy tắc quyết định đơn giản được rút ra từ dữ liệu. Cây quyết định dựa trên việc tạo ra một cấu trúc cây có các nút và các nhánh, mỗi nút trong cây đại diện cho một thuộc tính (hoặc biến độc lập), và các nhánh của nút đại diện cho các giá trị có thể của thuộc tính đó. Các lá của cây đại diện cho các lớp hoặc giá trị của biến mục tiêu.

Quá trình xây dựng cây quyết định bắt đầu từ nút gốc và phân tách dữ liệu dựa trên các thuộc tính sao cho sự không đồng nhất của các lớp trong các nhóm con là nhỏ nhất. Quá trình này tiếp tục đệ quy cho đến khi một điều kiện dừng được đáp ứng, chẳng hạn như khi tất cả các lá đều thuộc về cùng một lớp hoặc khi không còn thêm thuộc tính nào để phân tách.

A diagram of a triangle

Description automatically generated

Hình 1. 5 Mô phỏng ứng dụng Desision tree

1. **Thuật toán SVM**

Support Vector Machines (SVM) là một thuật toán máy học phổ biến được áp dụng cho cả bài toán phân loại và hồi quy. Mục tiêu của SVM là tìm ra ranh giới phân chia tối ưu giữa các nhóm dữ liệu.

Trong bài toán phân loại, SVM cố gắng tạo ra một ranh giới phân chia (gọi là siêu phẳng) sao cho khoảng cách từ các điểm dữ liệu gần nhất tới ranh giới này (còn gọi là margin) là lớn nhất. Các điểm dữ liệu gần nhất này được gọi là "vector hỗ trợ" (support vectors), từ đó xuất phát tên gọi của thuật toán.

SVM hoạt động hiệu quả trong các không gian chiều cao (high-dimensional space), có khả năng xử lý các tập dữ liệu có số lượng biến lớn. Cũng thích hợp cho các tập dữ liệu lớn, nhưng cần cân nhắc kỹ lưỡng khi lựa chọn các tham số quan trọng như loại kernel (ví dụ: linear, polynomial, radial basis function) để phù hợp với dữ liệu cụ thể và tránh overfitting.

1. **K-Nearest Neighbors**

K-Nearest Neighbors (K-NN) là một thuật toán máy học không giám sát được sử dụng cho các bài toán phân loại và hồi quy. Trong K-NN, việc phân loại hoặc dự đoán giá trị mới được thực hiện dựa trên việc xem xét các điểm dữ liệu gần nhất trong tập huấn luyện.

Cụ thể, khi phải phân loại một điểm dữ liệu mới, K-NN tính toán khoảng cách giữa điểm dữ liệu này với tất cả các điểm trong tập huấn luyện. Sau đó, chọn ra K điểm gần nhất và dùng đa số phiếu (ví dụ: phân loại dựa trên lớp phổ biến nhất trong K điểm) để quyết định lớp của điểm dữ liệu mới.

Thuật toán K-NN không yêu cầu việc huấn luyện mô hình trước, mà chỉ cần lưu trữ tất cả các điểm dữ liệu trong tập huấn luyện để sau đó sử dụng cho việc phân loại hoặc dự đoán. Tuy nhiên, việc tính toán khoảng cách với mỗi điểm dữ liệu trong tập huấn luyện có thể làm tăng độ phức tạp tính toán của thuật toán khi số lượng điểm dữ liệu lớn.

1. **Naive Bayes** **Classifier**

Naive Bayes Classifier, hay còn được gọi là phân loại Bayes đơn giản, là một mô hình phân loại trong máy học, được xây dựng dựa trên định lý Bayes với giả định "naive" (ngây thơ). Điều này ngụ ý rằng các biến độc lập với nhau, mặc dù trong thực tế thường không phải như vậy. Tuy nhiên, mặc dù có giả định này, mô hình Naive Bayes vẫn có hiệu suất tốt trong nhiều tình huống thực tế.

Cách hoạt động của Naive Bayes Classifier dựa trên tính toán xác suất để đánh giá xác suất của một điểm dữ liệu thuộc vào từng lớp phân loại. Dựa trên các xác suất này, điểm dữ liệu được phân loại vào lớp có xác suất cao nhất.

Mô hình Naive Bayes thường được áp dụng trong các bài toán như phân loại văn bản, phân loại email spam, phát hiện cảm xúc trong văn bản, và nhiều ứng dụng khác trong xử lý ngôn ngữ tự nhiên và phân loại dữ liệu. Mặc dù đơn giản, nhưng Naive Bayes có thể hoạt động hiệu quả và đem lại kết quả tốt cho nhiều loại dữ liệu.

1. A diagram of a software model

   Description automatically generated**Ứng dụng của học máy**

Hình 1. 6 Mô hình cơ bản của kỹ thuật Naive Beyes

Như đã trình bay ở phần trên máy học đã xuất hiện từ rất lâu, nhưng trong những thời điểm đầu tiên phát triển học máy còn nhiều hạn chế vì thế những ứng dụng thực tiễn còn chưa cao. Nhưng những năm 1990 đến nay đã có rất nhiều ứng dụng thực tiễn, có những ứng dụng đã đi vào lịch sử, tạo bước đệm cho những ứng dụng vĩ đại sau này. Một số ứng dụng có thế nhắc đến sau đây.

*Nhận diện giọng nói*: Nhận dạng giọng nói cho phép máy tính , ứng dụng và phần mềm hiểu và dịch dữ liệu giọng nói của con người thành văn bản cho các giải pháp kinh doanh . Mô hình nhận dạng giọng nói hoạt động bằng cách sử dụng trí tuệ nhân tạo (AI) để phân tích giọng nói và ngôn ngữ , xác định bằng cách học các từ đang nói, sau đó xuất những từ đó với độ chính xác phiên âm dưới dạng nội dung mô hình hoặc dữ liệu văn bản trên màn hình.

*Chăm sóc khách hàng*: Chatbots trực tuyến đang thay thế các tác nhân con người trong hành trình của khách hàng, thay đổi cách nghĩ về sự tương tác của khách hàng trên website và nền tảng xã hội.

*Thị giác máy tính*: Công nghệ AI này cho phép máy tính lấy thông tin có ý nghĩa từ video, hình ảnh kỹ thuật số và các đầu vào trực quan khác, sau đó thực thi hành động thích hợp.

*Công cụ gợi ý*: Sử dụng dữ liệu hành vi tiêu dùng trong quá khứ, các thuật toán có thể giúp khám phá các xu hướng dữ liệu có thể được sử dụng để phát triển các chiến lược cross-sell hiệu quả hơn.

*Phát hiện gian lận*: Các ngân hàng và các tổ chức tài chính có thể sử dụng máy học để phát hiện các giao dịch đáng ngờ.

Và còn rất rất nhiều lĩnh vực mà machine learning có thể được áp dụng, machine learning tỏ ra cực kỳ hiệu quả, hơn hẳn con người trong cụ thể các lĩnh vực mà chúng được áp dụng.

Một ví dụ đơn giản về sự ứng dụng của Machine Learning là trong lĩnh vực dự báo thời tiết. Trong quá trình dự báo thời tiết, các chuyên gia sẽ sử dụng các phép tính và quan sát từ các thông tin về thời tiết trong quá khứ để dự đoán các điều kiện thời tiết trong tương lai. Tuy nhiên, khi có một lượng lớn quan sát đến hàng triệu hoặc hàng tỉ, con người sẽ gặp khó khăn trong việc xử lý và tính toán trên dữ liệu lớn đó. Hơn nữa, việc xử lý dữ liệu lớn như vậy có thể dẫn đến các sai sót trong dự đoán.

Ứng dụng Machine Learning trong việc dự báo thời tiết giúp máy tính học từ các quan sát được thu thập trong quá khứ. Nhờ vào khả năng xử lý lượng lớn dữ liệu một cách nhanh chóng và hiệu quả, máy tính có thể tạo ra dự đoán về thời tiết trong tương lai với độ chính xác cao hơn rất nhiều so với con người.

Điều này làm nổi bật sự quan trọng và hiệu quả của Machine Learning trong thời đại công nghệ 4.0. Việc hiểu và áp dụng Machine Learning không chỉ mang lại lợi thế lớn cho cá nhân mà còn đóng góp tích cực vào sự phát triển toàn cầu trong lĩnh vực công nghệ.

1. **XỬ LÝ NGÔN NGỮ TỰ NHIÊN**
2. **Định nghĩa về xử lý ngôn ngữ tự nhiên**

Xử lý ngôn ngữ tự nhiên là việc sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) để phân tích và thao tác ngôn ngữ của con người trong bối cảnh tính toán. Điều này có thể bao gồm các nhiệm vụ như dịch ngôn ngữ, nhận dạng giọng nói, tạo văn bản, phân tích tình cảm và hơn thế nữa. NLP là một lĩnh vực đang phát triển nhanh chóng, với một loạt các ứng dụng trong các lĩnh vực như trí tuệ nhân tạo, truy xuất thông tin và tương tác giữa con người và máy tính.

Các tổ chức ngày nay có khối lượng lớn dữ liệu thoại và văn bản từ nhiều kênh liên lạc khác nhau như email, tin nhắn văn bản, bảng tin trên mạng xã hội, tệp video, tệp âm thanh và nhiều hơn nữa. Họ sử dụng phần mềm NLP để tự động xử lý dữ liệu này, phân tích ý định hoặc cảm xúc trong tin nhắn và phản hồi bằng người thật theo thời gian thực.

Xử lý ngôn ngữ tự nhiên (NLP) đóng một vai trò quan trọng trong việc hiệu quả hóa phân tích dữ liệu văn bản và giọng nói. Công nghệ này có khả năng xử lý sự đa dạng trong phương ngữ, tiếng lóng và những biến thường về ngôn ngữ thường gặp trong các cuộc trò chuyện hàng ngày. Các công ty tích hợp NLP vào các quá trình tự động, bao gồm:

* Xử lý, phân tích và lưu trữ lượng lớn tài liệu văn bản.
* Phân tích phản hồi từ khách hàng hoặc ghi âm cuộc gọi tới tổng đài.
* Triển khai chatbot cho dịch vụ khách hàng tự động.
* Trả lời các câu hỏi về người, sự vật, thời gian và địa điểm.
* Phân loại và trích xuất thông tin từ văn bản.

Ngoài ra, còn có thể tích hợp NLP trực tiếp vào các ứng dụng tương tác với khách hàng để tăng cường khả năng giao tiếp hiệu quả. Ví dụ, một Chatbot có thể phân tích và phân loại các truy vấn từ khách hàng, tự động trả lời các câu hỏi thường gặp và chuyển hướng các truy vấn phức tạp đến bộ phận hỗ trợ khách hàng. Sự tự động hóa này không chỉ giảm chi phí mà còn giúp nhân viên tránh mất thời gian vào các truy vấn lặp lại, từ đó cải thiện mức độ hài lòng của khách hàng.

1. **Các bước cơ bản của xử lý ngôn ngữ tự nhiên**

Trong quá để xử lý ngôn ngữ tự nhiên sẽ trải qua các bước thự hiện, các bước này nhầm giúp cho dữ liệu về ngôn ngữ được rõ ràng, minh bạch hơn, giảm thiểu dữ liệu bị mất mát, dựa trên dữ liệu ngôn ngữ được xử lý đó, việc các huấn luyện mô hình học máy sẽ cho ra kết quả tối ưu nhất. Sẽ có khá nhiều bước xử lý ngôn ngữ tự nhiên khác nhau, tùy vào yêu cầu thì sẽ chọn các bước nào phù hợp nhất với từng loại ngôn ngữ, nhưng chung quy lại sẽ bao gồm các bước cơ bản như sau: Tokenization, Named Entity Recognition, Part-of-Speech, Lemmatization, Dependency Parsing, Feature Extraction,... Với các ví dụ sau sẽ được thực hiện trên thư viên spaCy và NLTK.

* **Tokenization**

Tokenization là quá trình chia một chuỗi văn bản thành các phần tử nhỏ hơn gọi là "token". Một token có thể là một từ, một dấu câu hoặc một phần từ của từ, nhưng chúng thường là các đơn vị ngữ cảnh như từ hoặc số. Ví dụ trong câu “*Người đàn ông đi bộ*” quá trình tokenization sẽ chia thành các token như sau: ("Người", "đàn", "ông", "đi", "bộ", "đến", "công", "viên").

Việc tách các câu thành những từ đơn lẽ sẽ hỗ trợ rất nhiều trong việc xử lý văn bản, làm sạch văn bản hoặc sâu hơn là trích xuất các đặc trưng từ văn bản, có thể nói tokennization là bước không thể thiếu trong xử lý ngôn ngữ tự nhiên nói chung và xử lý văn bản nói riêng.

**import** nltk  
**from** nltk.tokenize **import** word\_tokenize  
  
text = "Tokenization is a process of dividing a text into smaller units called tokens. It can be done at various levels such as word level or sentence level."  
  
word\_tokens = word\_tokenize(text)  
print("Word Tokens:")  
print(word\_tokens)

>>>Word Tokens:  
>>>['Tokenization', 'is', 'a', 'process', 'of', 'dividing', 'a', 'text', 'into', 'smaller', 'units', 'called', 'tokens', '.', 'It', 'can', 'be', 'done', 'at', 'various', 'levels', 'such', 'as', 'word', 'level', 'or', 'sentence', 'level', '.']

* **Named Entity Recognition (NER)**

Named Entity Recognition (NER) là một phần quan trọng của xử lý ngôn ngữ tự nhiên (NLP) dùng để xác định và phân loại các thực thể có tên (named entities) trong văn bản thành các loại nhất định như tên riêng, địa điểm, ngày tháng, số lượng, tổ chức, và các loại khác.

NER thường được sử dụng như một bước tiền xử lý trong nhiều ứng dụng NLP như: “*Tóm tắt văn bản*” để xác định các thông tin quan trọng trong văn bản, “*Trích xuất thông tin*” trích xuất thông tin cụ thể từ văn bản, “*Tạo đồng bộ hóa*” đồng bộ hóa thông tin giữa các tài liệu hoặc ngôn ngữ khác nhau, “*Phân loại văn bản*” xác định loại văn bản, chẳng hạn liệu văn bản đó là tin tức, bài viết y tế hay bình luận sản phẩm.

import spacy  
nlp = spacy.load("en\_core\_web\_sm")  
  
**text** = "Apple was founded by Steve Jobs and Steve Wozniak in April 1976. The company's headquarters is in Cupertino, California."  
  
doc = nlp(**text**)  
  
print("Named Entities:")  
**for** ent **in** doc.ents:  
 print(ent.**text**, "-", ent.label\_)

>>>Named Entities:  
>>>Apple - ORG  
>>>Steve Jobs - PERSON  
>>>Steve Wozniak - PERSON  
>>>April 1976 - DATE  
>>>Cupertino - GPE  
>>>California - GPE

* **Part-of-Speech**

Part-of-Speech dùng để xác định loại từ của mỗi từ trong một câu, như danh từ, động từ, tính từ, trạng từ, chủ ngữ, tân ngữ, và các loại khác. Mục tiêu của POS là phân loại từng từ trong một câu văn thành các loại từ cơ bản để hiểu cấu trúc ngữ pháp và ý nghĩa của câu.

POS rất quan trọng trong NLP vì cung cấp thông tin về cấu trúc ngữ pháp và ý nghĩa của câu, giúp cho các ứng dụng như parsing, semantic analysis, machine translation, và information retrieval hoạt động hiệu quả.

import spacy  
nlp = spacy.load("en\_core\_web\_sm")  
  
**text** = "I love natural language processing."  
  
doc = nlp(**text**)  
  
**for** **token** **in** doc:  
 print(**token**.**text**, "-", **token**.pos\_)

>>>I - PRON  
>>>love - VERB  
>>>natural - ADJ  
>>>language - NOUN  
>>>processing - NOUN  
>>>. - PUNCT

* **Lemmatization**

Lemmatization là quá trình chuyển đổi một từ về dạng gốc, được gọi là "lemma", dựa trên từ điển ngôn ngữ và các quy tắc ngữ pháp. Mục tiêu của lemmatization là chuyển đổi các từ về dạng gốc của chúng để giảm thiểu sự biến thể và đồng bộ hóa các từ có cùng nguồn gốc.

So với stemming, lemmatization cung cấp kết quả chính xác hơn vì sử dụng từ điển ngôn ngữ để xác định dạng gốc của từ, trong khi stemming chỉ cắt bớt các phần cuối của từ mà có thể không tạo ra các từ hợp lệ.

Lemmatization thường được sử dụng trong các ứng dụng NLP để chuẩn hóa văn bản trước khi thực hiện các phân tích văn bản khác như phân loại văn bản, phân tích ý kiến, hoặc tạo đặc trưng cho các mô hình học máy.

import spacy   
nlp = spacy.load("en\_core\_web\_sm")  
  
**text** = "The cats are running and jumping around the house."  
  
doc = nlp(**text**)  
  
**for** **token** **in** doc:  
 print(**token**.**text**, "-", **token**.lemma\_)

>>>cats - cat  
>>>are - be  
>>>running - run  
>>>and - and  
>>>jumping - jump  
>>>around - around  
>>>the - the  
>>>house - house  
>>>. - .

* **Dependency Parsing**

Dependency Parsing là quá trình phân tích cấu trúc ngữ pháp của một câu để xác định các quan hệ phụ thuộc (dependency relationships) giữa các từ trong câu. Mục tiêu của Dependency Parsing là biểu diễn câu văn dưới dạng một cây phụ thuộc (dependency tree), trong đó mỗi từ là một nút và các quan hệ phụ thuộc là các cạnh.

Trong một cây phụ thuộc, mỗi từ có thể là một "trụ" (head) hoặc "chỉ phụ thuộc" (dependent) của một hoặc nhiều từ khác. Quan hệ giữa một từ và trụ thường là một trong các loại quan hệ như chủ ngữ, tân ngữ, động từ, trạng từ,...Ví dụ trong câu "*John eats apples*", từ "*eats*" phụ thuộc vào từ "*John*" với quan hệ chủ ngữ (subject) và "*apples*" với quan hệ tân ngữ (object).

**import** spacy  
nlp = spacy.load("en\_core\_web\_sm")  
  
text = "The cat is sleeping on the mat."  
  
doc = nlp(text)  
  
**for** token **in** doc:  
 print(token.text, "-->", token.dep\_, "-->", token.head.text)

>>>The --> det --> cat  
>>>cat --> nsubj --> sleeping  
>>>is --> aux --> sleeping  
>>>sleeping --> ROOT --> sleeping  
>>>on --> prep --> sleeping  
>>>the --> det --> mat  
>>>mat --> pobj --> on  
>>>. --> punct --> sleeping

* **Feature Extraction**

Feature extraction trong xử lý ngôn ngữ tự nhiên (NLP) là quá trình chuyển đổi văn bản thành các đặc trưng số học mà mô hình máy học có thể hiểu được. Mục đích của việc này là tạo ra một biểu diễn số học của dữ liệu văn bản để có thể sử dụng cho các mô hình học máy, như hồi quy, phân loại hoặc gom cụm. một số phương pháp phổ biến trong feature extraction: Bag-of-Words (BoW), TF-IDF (Term Frequency-Inverse Document Frequency), Word Embeddings, N-grams.

1. **Ứng dụng của xử lý ngôn ngữ tự nhiên**

Nhận dạng chữ viết có hai loại nhận dạng, ví dụ như loại đầu tiên là nhận dạng bản in: nhận dạng và chuyển đổi văn bản sách giáo khoa. Khó hơn nữa là nhận dạng chữ viết tay, vốn khó vì chữ viết tay không có mẫu rõ ràng và thay đổi tùy theo từng người. Với chương trình nhận dạng xuất bản, hàng ngàn cuốn sách thư viện có thể được chuyển đổi thành tài liệu điện tử trong thời gian ngắn. Nhận dạng chữ viết tay của con người có ứng dụng trong tội phạm học và bảo mật thông tin (nhận dạng chữ ký điện tử).

Nhận dạng tiếng nói nhận biết âm thanh và chuyển đổi chúng thành văn bản tương ứng. Giúp mọi người sử dụng thiết bị nhanh hơn và dễ dàng hơn. Đây cũng là bước đầu tiên cần thực hiện để hiện thực hóa giao tiếp giữa con người và robot. Một ví dụ về ứng dụng nhận dạng giọng nói đang giúp đỡ người khiếm thị.

Tổng hợp tiếng nói từ văn bản được tự động tổng hợp thành giọng nói. Thay vì phải tự đọc nội dung của một cuốn sách hoặc trang web, sẽ tự động đọc. Giống như nhận dạng giọng nói, tổng hợp giọng nói là trợ thủ đắc lực cho người mù nhưng ngược lại, là bước cuối cùng trong quá trình giao tiếp giữa robot và con người.

Dịch tự động (Machine translate) là một chương trình dịch tự động từ ngôn ngữ này sang ngôn ngữ khác. Nghiên cứu và ứng dụng dịch máy hiện nay. Phát triển nhanh chóng, ứng dụng có độ chính xác cao.

Tìm kiếm thông tin (Information retrieval)hãy hỏi và chương trình sẽ tự động tìm nội dung phù hợp nhất. Thông tin đang bùng nổ, đặc biệt là với sự phát triển của Internet khiến việc lấy thông tin trở nên dễ dàng hơn bao giờ hết. Vấn đề là tìm được thông tin phù hợp và đặc biệt thông tin đó phải đáng tin cậy. Các công cụ tìm kiếm trên Internet như Google hay Yahoo hiện chỉ phân tích nội dung rất đơn giản dựa trên tần suất từ khóa, thứ hạng trang và nhiều tiêu chí đánh giá khác để đưa ra kết luận. Kết quả là nhiều lượt tìm kiếm không nhận được câu trả lời phù hợp, thậm chí dẫn đến các liên kết không liên quan do các website dùng thủ đoạn lừa đảo để giới thiệu sản phẩm (kỹ thuật SEO - Search Engine Optimization). Trên thực tế, cho đến nay chưa có công cụ tìm kiếm nào hiểu được ngôn ngữ tự nhiên của con người, ngoại trừ www.ask.com, được coi là “hiểu biết”; trả lời các câu hỏi có cấu trúc đơn giản nhất.

Tóm tắt văn bản trong xử lý ngôn ngữ tự nhiên (NLP) là quá trình tạo ra một phiên bản ngắn gọn và hợp lý của một văn bản dài hơn, bằng cách giữ lại những thông tin quan trọng và loại bỏ thông tin không cần thiết. Mục tiêu của tóm tắt văn bản là cung cấp một bản tóm tắt thú vị và dễ đọc, giúp người đọc nắm bắt được ý chính của văn bản mà không cần đọc toàn bộ nội dung.

Khai phá dữ liệu và phát hiện tri thức là tìm kiếm thông tin mới trong nhiều tài liệu khác nhau. Khai thác dữ liệu mô phỏng việc học tập của con người và tìm thấy thông tin hữu ích. Ở mức độ đơn giản với các công cụ tìm kiếm, cho phép đặt câu hỏi theo cách mà công cụ tìm kiếm tìm ra câu trả lời phù hợp nhất. Ví dụ ứng dụng của NLP.

1. **PHÂN LOẠI VĂN BẢN**
2. **Định nghĩa về văn bản**

Khái niệm văn bản là một khái niệm cơ bản trong lĩnh vực ngôn ngữ học và trí tuệ nhân tạo, đề cập đến một đoạn văn bản hoặc một tập hợp các từ, cụm từ và câu được sắp xếp theo một trật tự nhất định để truyền đạt ý nghĩa hoặc thông điệp. Dưới dạng dữ liệu, văn bản thường được biểu diễn bằng các chuỗi ký tự hoặc các token, trong đó mỗi token có thể là một từ, một cụm từ, một ký tự hoặc một số ký tự đặc biệt. Chi tiết hơn, một văn bản có thể bao gồm các thành phần sau.

*Từ và Cụm Từ*: Là các đơn vị cơ bản của văn bản, bao gồm các từ đơn (như "nhà", "đi", "là") hoặc các cụm từ (như "nhà hàng", "điều kiện thời tiết", "trong khi").

*Câu*: Là một chuỗi các từ và cụm từ được sắp xếp theo một cách cú pháp nhất định để truyền đạt một ý nghĩa hoặc một thông điệp. Mỗi câu thường kết thúc bằng dấu chấm câu, dấu phẩy, hoặc các dấu câu khác.

*Đoạn*: Là một nhóm các câu liên quan nhau, thường chứa một ý chính hoặc một phần của thông điệp được truyền đạt trong văn bản.

*Tài liệu Văn Bản*: Là một tập hợp các đoạn văn bản và các phần khác nhau của văn bản, có thể là một bài báo, một trang web, một email, một cuốn sách, v.v.

*Ngôn Ngữ và Cú Pháp*: Văn bản được viết bằng một ngôn ngữ cụ thể và tuân thủ các quy tắc cú pháp của ngôn ngữ đó, bao gồm cú pháp ngữ pháp, cú pháp từ vựng, và cú pháp ngữ cảnh. Văn bản không chỉ là một nguồn thông tin quan trọng mà còn là đối tượng nghiên cứu và xử lý quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), máy học, khai phá dữ liệu, và nhiều lĩnh vực khác.

1. **Định nghĩa về phân loại văn bản**

Phân loại văn bản là quá trình tự động gán nhãn (tên lớp / nhãn lớp) cho các văn bản ngôn ngữ tự nhiên vào một hoặc nhiều lớp được xác định trước. Đây là một bước quan trọng trong khai phá dữ liệu văn bản, nơi các văn bản được phân loại dựa trên nội dung của chúng vào các chủ đề đã được xác định trước.

Phân loại văn bản có nhiều ứng dụng, bao gồm hỗ trợ trong quá trình tìm kiếm thông tin, chiết lọc thông tin, lọc văn bản hoặc tự động dẫn đường cho các văn bản tới các chủ đề xác định trước. Quá trình này có thể được thực hiện thủ công hoặc tự động bằng cách sử dụng các kỹ thuật học máy có giám sát.

Các hệ thống phân loại văn bản có thể được áp dụng trong nhiều ngữ cảnh, như phân loại tài liệu trong thư viện điện tử, phân loại các bài báo trên các trang tin điện tử, và nhiều ứng dụng khác. Khi được triển khai hiệu quả, các hệ thống này có thể cung cấp kết quả đáng kể, giúp tối ưu hóa công việc và tạo ra lợi ích đáng kể cho người sử dụng.

1. **Mô hình phân loại văn bản**

Phân loại văn bản là một bài toán trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), trong đó mục tiêu là tự động gán các nhãn hoặc danh mục cho các văn bản dựa trên nội dung của chúng. Đây là một bước quan trọng trong khai phá dữ liệu văn bản, giúp tổ chức và hiểu được các tài liệu văn bản một cách tự động và hiệu quả. Dưới đây là các bước cơ bản của bài toán phân loại văn bản.

* **Dataset / Text – Label**

Ở bước này nhiệm vụ là chọn một tập dữ liệu theo nhu cầu phân loại văn bản, các loại văn bản có thể thu thập từ nhiều nguồn khác nhau, từ các trang báo, các blog, status hoặc comment trên các nền tảng xã hộ, đơn giản hơn hiện nay đã có nhiều tập dữ liệu văn bản đã được chọn lọc sẵn từ nhiều nhà nghiên cứu trước đó đã được sử dụng ở các nền tảng như Github, Kaggle, Zenodo,...

* **Data preprocessing**

Data preprocessing sẽ là bước xử lý những dữ liệu đã được thu thập từ trước đó, bước này đồi hỏi phải trải qua nhiều bước nhỏ hơn để có thể trong ra một bộ dữ liệu sạch sẽ, rõ ràng. Cụ thể bước này sẽ thực hiện các bước nhỏ khác như: Tách từ, tách các câu, loại bỏ stopword, loại bỏ các ký tự đặc biệc, loại bỏ khoảng trắng dư thừa, loại bỏ các chữ bị kéo dài. Hiện nay đã có nhiều thư viện hỗ trợ cho bước Data preprocessing này, giúp giảm chi phí cho việc xử lý làm sạch dữ liệu ,có thể kể đến: NLTK(Natural Language Toolkit), spaCy, Scikit – learn, Gensim.

* **Feature Extraction**

Là quá trình chuyển đổi văn bản từ dạng không cấu trúc thành các đặc trưng (features) có thể được sử dụng để huấn luyện các mô hình học máy. Mục tiêu của feature extraction là biến đổi văn bản thành các biểu diễn số học mà mô hình có thể hiểu và xử lý. Các phương pháp trich xuất đặc trưng thường được sử là Bag of Words (BoW), F-IDF (Term Frequency-Inverse Document Frequency), Word Embeddings, N-gram.

* **Machine Learning Model**

Trong xử lý ngôn ngữ tự nhiên (NLP), Machine Learning Model là một mô hình máy học được huấn luyện để thực hiện các tác vụ như phân loại văn bản, dự đoán từ tiếp theo, dịch máy, phân tích ý kiến, tóm tắt văn bản và nhiều tác vụ khác liên quan đến ngôn ngữ.

Để có một kết quả tối ưu nhất đều này phụ thuộc vào việc chọn một mô hình phù hợp với dữ liệu và cấu trúc của dữ liệu, những mô hình thường được sử dụng để huấn luyện mô hình như: Naive Bayes, Logistic Regression, K-Nearest Neighbors, Gradient Boosting Machines, Support Vector Machine.

* **Evaluation Model**

Evaluation Model là quá trình đánh giá hiệu suất của một mô hình hoặc hệ thống NLP dựa trên các tiêu chí cụ thể. Mục tiêu của việc đánh giá mô hình là đo lường khả năng của trong việc thực hiện các tác vụ NLP một cách chính xác và hiệu quả. Chuẩn bị tập dữ liệu kiểm tra hoặc tập dữ liệu đánh giá riêng biệt từ tập dữ liệu huấn luyện. Tập dữ liệu này thường chứa các mẫu dữ liệu không được mô hình nhìn thấy trong quá trình huấn luyện để đảm bảo tính khách quan của đánh giá.

Xác định các tiêu chí để đánh giá hiệu suất của mô hình, phụ thuộc vào loại tác vụ NLP cụ thể. Các tiêu chí đánh giá có thể bao gồm độ chính xác, độ phủ, F1-score, perplexity, BLEU score, ROUGE score, và nhiều tiêu chí khác. Áp dụng mô hình đã huấn luyện lên tập dữ liệu đánh giá và đo lường các tiêu chí đã chọn. Mỗi mô hình sẽ được đánh giá trên một tập dữ liệu kiểm tra riêng biệt hoặc trên nhiều tập dữ liệu kiểm tra để đảm bảo tính đáng tin cậy của kết quả. So sánh kết quả của mô hình đánh giá với các mô hình khác để xác định xem mô hình nào hoạt động tốt nhất cho tác vụ cụ thể đó.

* **Improving Model**

Là bước tối ưu hóa và cải thiện hiệu suất của mô hình NLP sau khi đã đánh giá và phân tích kết quả. Mục tiêu của bước này là tăng cường khả năng dự đoán và hiệu suất tổng thể của mô hình. một số bước thường được thực hiện trong quá trình cải thiện mô hình: Tinh chỉnh tham số (Parameter Tuning), Tăng cường dữ liệu (Data Augmentation), Feature Engineering.

1. A diagram of a data processing process

   Description automatically generated**Ứng dụng phân loại phân bản**

Hình 1. 7 Mô hình tổng quát phân loại văn bản

Việc phân loại văn bản có rất nhiều ứng dụng trong thực tế, từ các lĩnh vực công nghệ thông tin đến ngành y tế và tài chính. Dưới đây là một số ứng dụng cụ thể của việc phân loại văn bản trong thực tế:

* *Phân loại email*: Gmail, Outlook và các dịch vụ email khác sử dụng phân loại văn bản để tự động phân loại thư đến vào các hộp thư như hộp thư đến chính, hộp thư quan trọng hoặc hộp thư rác.
* *Phân loại tài liệu và tài nguyên công ty*: Công ty sử dụng phân loại văn bản để tổ chức và quản lý tài liệu nội bộ, từ hợp đồng và báo cáo đến chính sách và quy trình.
* *Phân loại sản phẩm và dịch vụ*: Các trang web thương mại điện tử sử dụng phân loại văn bản để tự động phân loại và gắn nhãn cho các sản phẩm, giúp người dùng dễ dàng tìm kiếm và mua hàng.
* *Phân loại dữ liệu y tế*: Trong lĩnh vực y tế, phân loại văn bản được sử dụng để tổ chức và phân loại thông tin y tế, bao gồm bệnh án, báo cáo y khoa và thông tin liên quan đến bệnh nhân.
* *Phân loại tài chính*: Các tổ chức tài chính sử dụng phân loại văn bản để phân loại và phân tích thông tin tài chính, bao gồm báo cáo tài chính, dự báo xu hướng thị trường và phân tích dữ liệu tài chính.
* *Phân loại tin tức và bài viết*: Các trang web tin tức và blog sử dụng phân loại văn bản để tự động gắn nhãn và phân loại các tin tức và bài viết theo chủ đề hoặc danh mục.
* *Phân loại ý kiến và phản hồi khách hàng*: Các doanh nghiệp sử dụng phân loại văn bản để phân tích ý kiến và phản hồi của khách hàng, từ đánh giá sản phẩm đến bài đánh giá trên mạng xã hội.
* *Phân loại dữ liệu trong ngành sản xuất*: Trong ngành sản xuất, phân loại văn bản có thể được sử dụng để tổ chức và quản lý dữ liệu liên quan đến quy trình sản xuất, bảo trì và kiểm soát chất lượng.

1. **KỸ THUẬT TF – IDF**
2. **Định nghĩa**
3. **Tần số từ (TF - Term Frequency)**

Term Frequency (TF), hay Tần suất xuất hiện của từ, là số lần một từ xuất hiện trong một văn bản. Do các văn bản có thể có độ dài khác nhau, một số từ có thể xuất hiện nhiều lần trong một văn bản dài hơn so với một văn bản ngắn. Do đó, để chuẩn hóa, term frequency thường được chia cho độ dài của văn bản, tức là tổng số từ trong một văn bản.

Ví dụ: Nếu từ "machine" xuất hiện 10 lần trong một văn bản có tổng cộng 1000 từ, thì TF của "machine" trong văn bản đó là 10/1000=0,01.

Trong đó:

* t là từ cần tính TF.
* d là văn bản.
* Số lần từ t xuất hiện trong văn bản d sẽ được đếm.
* Tổng số từ trong văn bản d sẽ được đếm.

1. **Tần số văn bản nghịch đảo (IDF - Inverse Document Frequency)**

IDF là viết tắt của "Inverse Document Frequency" (Tần số văn bản nghịch đảo). Đây là một khái niệm quan trọng trong xử lý ngôn ngữ tự nhiên và khai phá dữ liệu văn bản, được sử dụng để đo lường mức độ quan trọng của một từ trong một tập hợp các văn bản.

Cụ thể, IDF đo lường tần suất xuất hiện của một từ trong toàn bộ tập hợp các văn bản và định lượng mức độ quan trọng của từ đó. Mục đích của IDF là tìm ra những từ xuất hiện hiếm trong tập hợp các văn bản, tức là những từ đặc biệt có khả năng đặc trưng và có thể phân biệt các văn bản với nhau.

Công thức tính IDF cho một từ t thường là số lượng văn bản trong tập dữ liệu chia cho Document Frequency (DF) của từ t. Để tránh việc chia cho 0, có thể thêm một hằng số vào mẫu số. Một công thức phổ biến cho IDF là:

Trong đó:

* N là tổng số văn bản trong tập dữ liệu D.
* n(t) là số văn bản trong tập hợp D mà từ t xuất hiện

Nếu trong một tập dữ liệu có tổng cộng 10,000 văn bản, và từ "machine" xuất hiện trong 1,000 văn bản, thì IDF của "machine" là log (10000/1000) = log (10) = 1.

Hàm logarithm được sử dụng để giảm thiểu ảnh hưởng của các từ phổ biến mà không cung cấp nhiều thông tin quan trọng. Càng cao giá trị IDF, tức là càng hiếm khi từ xuất hiện trong các văn bản, càng được coi là quan trọng và có khả năng phân biệt cao giữa các văn bản.

1. **Tần số từ - Tần số văn bản nghịch đảo (TF-IDF)**

TF-IDF, viết tắt của "Term Frequency – Inverse Document Frequency", là một chỉ số thống kê thể hiện mức độ quan trọng của một từ trong một văn bản so với một tập hợp các văn bản. Giá trị TF-IDF tăng tương ứng với số lần một từ xuất hiện trong văn bản, nhưng thường được điều chỉnh bằng tần số của từ trong toàn bộ tập hợp các văn bản, giúp điều chỉnh thực tế là một số từ xuất hiện thường xuyên hơn nói chung.

Ví dụ: Nếu TF của từ "machine" trong một văn bản là 0.01 và IDF của "machine" trong toàn bộ tập dữ liệu là 1, thì TF-IDF của "machine" trong văn bản đó là 0,01x1= 0,01.

1. **Vai trò TF – IDF**

TF-IDF (Term Frequency-Inverse Document Frequency) có vai trò quan trọng trong xử lý ngôn ngữ tự nhiên và khai phá dữ liệu văn bản, và được sử dụng rộng rãi trong nhiều ứng dụng như tìm kiếm thông tin, phân loại văn bản, và trích xuất thông tin. TF-IDF giúp xác định những từ khóa quan trọng trong một văn bản bằng cách tăng cường trọng số cho những từ xuất hiện ít trong văn bản đó nhưng xuất hiện nhiều trong các văn bản khác. Cải thiện hiệu suất tìm kiếm trong các hệ thống tìm kiếm, TF-IDF được sử dụng để đánh giá độ phù hợp của một văn bản với một truy vấn tìm kiếm. Các từ có TF-IDF cao hơn được coi là những từ khóa quan trọng và giúp cải thiện chất lượng kết quả tìm kiếm.

Các bài toán phân loại văn bản, TF-IDF được sử dụng để biểu diễn các văn bản dưới dạng vector có kích thước cố định. Các mô hình học máy sau đó có thể được huấn luyện trên các vector này để phân loại các văn bản vào các lớp khác nhau. TF-IDF có thể được sử dụng để trích xuất thông tin quan trọng từ các văn bản, như trích xuất từ khóa, phát hiện chủ đề hoặc phân loại văn bản vào các danh mục cụ thể. Các từ có TF-IDF thấp có thể được loại bỏ hoặc được coi là không quan trọng trong các ứng dụng xử lý ngôn ngữ tự nhiên như làm sạch văn bản hoặc giảm kích thước của dữ liệu.

Tóm lại, TF-IDF đóng vai trò quan trọng trong việc biểu diễn và đánh giá mức độ quan trọng của các từ trong văn bản, giúp cải thiện hiệu suất của các hệ thống tìm kiếm và phân loại văn bản, và hỗ trợ trong quá trình trích xuất thông tin từ dữ liệu văn bản.

1. **Ưu điểm và nhược điểm của TF – IDF**

* **Ưu điểm**

*Tính tương đối*: TF-IDF không chỉ đơn giản là đếm số lần xuất hiện của một từ trong một tài liệu mà còn xem xét tần suất của từ đó so với tần suất trong tập hợp các tài liệu khác. Điều này giúp làm nổi bật những từ quan trọng trong một tài liệu cụ thể so với các từ thông thường.

*Giảm nhiễu*: Những từ phổ biến như "và", "hoặc" có thể xuất hiện nhiều lần trong các tài liệu, nhưng chúng thường không mang lại nhiều ý nghĩa. TF-IDF giúp giảm nhiễu bằng cách giảm tầm quan trọng của những từ này trong việc phân tích và tạo ra biểu đồ từ khóa.

*Đa dạng phương pháp*: TF-IDF có thể được sử dụng cho nhiều mục đích trong xử lý ngôn ngữ tự nhiên, bao gồm phân loại văn bản, trích xuất từ khóa, và tìm kiếm thông tin.

*Dễ hiểu và triển khai*: Phương pháp này khá đơn giản để hiểu và triển khai, không cần nhiều kiến thức chuyên sâu về toán học hay machine learning.

* **Nhược điểm**

*Không xử lý ngữ cảnh*: TF-IDF không xem xét ngữ cảnh của từ trong văn bản. Điều này có nghĩa là có thể bỏ qua mối quan hệ giữa các từ hoặc không xử lý tốt trong các trường hợp cụ thể.

*Không xử lý đồng nghĩa và từ đồng nghĩa*: TF-IDF không xem xét các từ đồng nghĩa (synonyms) hoặc từ đồng nghĩa (homographs), điều này có thể làm mất đi một số thông tin quan trọng trong văn bản.

*Nhược điểm với văn bản ngắn*: Trong trường hợp các văn bản ngắn, TF-IDF có thể không hiệu quả vì sự hiện diện và tần suất của từ không đủ để phản ánh tầm quan trọng của chúng.

1. **MÔ HÌNH NAIVE BAYES**
2. **Định nghĩa về Naive Baye**

Naive Bayes Classification (NBC) là một phương pháp dựa trên định lý Bayes về xác suất, được sử dụng để đưa ra dự đoán và phân loại dữ liệu dựa trên thông tin quan sát và thống kê từ dữ liệu. NBC là một trong những thuật toán phổ biến trong lĩnh vực Machine Learning, được áp dụng rộng rãi để tạo ra các dự đoán chính xác từ tập dữ liệu thu thập được. Thuật toán này được đánh giá cao vì tính dễ hiểu và độ chính xác cao. NBC thuộc nhóm thuật toán Học có giám sát (Supervised Machine Learning Algorithms), nghĩa là thuật toán học từ các ví dụ được cung cấp từ các mẫu dữ liệu đã biết trước.

Định lý Bayes cho phép tính xác suất xảy ra của một sự kiện ngẫu nhiên A khi biết sự kiện liên quan B đã xảy ra. Xác suất này được ký hiệu là P(A|B), và đọc là “xác suất của A nếu có B”. Đại lượng này được gọi xác suất có điều kiện hay xác suất hậu nghiệm vì được rút ra từ giá trị được cho của B hoặc phụ thuộc vào giá trị đó.

Theo định lí Bayes, P(A|B) sẽ phụ thuộc vào 3 yếu tố:

* Xác suất xảy ra A, không quan tâm đến B. Kí hiệu là P(A).
* Xác suất xảy ra B, không quan tâm đến A. Kí hiệu là P(B).
* Xác suất xảy ra B khi biết A xảy ra. Kí hiệu là P(B|A). Đại lượng này gọi là khả năng (likelihood) xảy ra B khi biết A đã xảy ra.

1. **Các bước hoạt động của Naive Bayes**

Cách hoạt động của thuật toán Naive Bayes dựa trên định lý Bayes về xác suất. Thuật toán này giả định rằng các đặc trưng đầu vào độc lập với nhau, tức là sự xuất hiện của một đặc trưng không phụ thuộc vào sự xuất hiện của các đặc trưng khác. Dựa trên giả định này, Naive Bayes tính toán xác suất của một lớp dựa trên các đặc trưng của mẫu dữ liệu.

* + **Xác định các đặc trưng**

Đầu tiên, thuật toán cần xác định các đặc trưng hoặc thuộc tính của dữ liệu. Ví dụ, trong bài toán phân loại email, các đặc trưng có thể bao gồm từ vựng xuất hiện trong email, số lượng từ trong email, hoặc các đặc điểm của email như độ dài, tỷ lệ các ký tự in hoa, v.v.

* + **Tính toán xác suất của mỗi lớp**

Naive Bayes tính toán xác suất của mỗi lớp dựa trên các đặc trưng của mẫu dữ liệu. Đối với mỗi lớp, tính toán xác suất có điều kiện của mỗi đặc trưng dựa trên dữ liệu huấn luyện.

* + **Áp dụng định lý Bayes**

Sau khi tính toán xác suất có điều kiện cho mỗi lớp, Naive Bayes áp dụng định lý Bayes để tính toán xác suất của mỗi lớp dựa trên các đặc trưng của mẫu dữ liệu. Điều này giúp dự đoán xác suất của mỗi lớp cho mẫu dữ liệu mới.

* + **Chọn lớp có xác suất cao nhất**

Naive Bayes chọn lớp có xác suất cao nhất là lớp dự đoán cho mẫu dữ liệu. Xác định lớp mà mẫu dữ liệu có xác suất cao nhất thuộc vào và gán lớp đó cho dự đoán cuối cùng.

* + **Đưa ra dự đoán**

Cuối cùng, thuật toán đưa ra dự đoán bằng cách gán lớp có xác suất cao nhất cho mẫu dữ liệu. Điều này có nghĩa là xác định lớp mà mẫu dữ liệu được phân loại vào và đưa ra dự đoán tương ứng.

1. **Ưu điểm và nhược điểm của Naive Bayes**

* **Ưu điểm**

Thuật toán Naive Bayes thực hiện phân loại một cách dễ dàng và nhanh chóng. Vì vậy, thích hợp cho các bài toán có thời gian đáp ứng nhanh hoặc cần sự đơn giản. Naive Bayes thường hội tụ nhanh hơn các mô hình phân biệt khác như hồi quy logistic, giúp giảm thời gian huấn luyện. Thuật toán này yêu cầu ít dữ liệu đào tạo hơn so với một số mô hình khác, điều này có thể rất hữu ích khi dữ liệu huấn luyện có hạn.

Naive Bayes có khả năng mở rộng tốt và chia tỷ lệ tuyến tính với số lượng đặc trưng dự đoán và điểm dữ liệu. Có thể đưa ra dự đoán xác suất, giúp người dùng hiểu rõ hơn về độ tin cậy của kết quả phân loại. Ngoài ra, cũng có khả năng xử lý cả dữ liệu liên tục và rời rạc. Được sử dụng cho cả hai loại bài toán phân loại: nhị phân (hai lớp) và đa lớp (nhiều lớp). Điều này trở thành một lựa chọn linh hoạt cho nhiều loại bài toán.

* **Nhược điểm**

Một trong những giả định chính của Naive Bayes là giả định về độc lập giữa các đặc trưng khi biết lớp. Trong thực tế, các đặc trưng thường không độc lập hoàn toàn, điều này có thể dẫn đến các dự đoán không chính xác nếu có sự tương quan giữa các đặc trưng. Nếu một đặc trưng trong dữ liệu test không xuất hiện trong tập dữ liệu huấn luyện, Naive Bayes sẽ gán xác suất bằng 0, dẫn đến việc mô hình không thể tạo ra dự đoán cho những trường hợp này.

Naive Bayes thường hoạt động tốt trên dữ liệu đơn giản và có cấu trúc rõ ràng. Tuy nhiên, khi dữ liệu phức tạp và có tương quan cao giữa các đặc trưng, Naive Bayes có thể tạo ra các dự đoán không chính xác. có thể bị ảnh hưởng bởi dữ liệu nhiễu hoặc các đặc trưng không quan trọng. Điều này có thể dẫn đến việc mô hình không hoạt động tốt trên dữ liệu thực tế.

1. **Ứng dụng của Naive Bayes**

Naive Bayes thường được sử dụng trong phân loại văn bản, chẳng hạn như phân loại email vào thư rác và thư không phải thư rác. Có thể được áp dụng trong việc phân loại văn bản theo chủ đề hoặc ngôn ngữ. Thuật toán Naive Bayes cũng được sử dụng để phân tích cảm xúc từ văn bản, như xác định liệu một bình luận trên mạng xã hội là tích cực, tiêu cực hay trung tính.

Với khả năng phân loại hiệu quả và tính nhanh nhẹn, Naive Bayes thường được sử dụng trong các hệ thống lọc thư rác để phân biệt giữa email spam và email hợp lệ. Naive Bayes có thể được tích hợp vào hệ thống gợi ý để đề xuất sản phẩm hoặc nội dung dựa trên lịch sử hoạt động của người dùng.

Trong lĩnh vực y học, Naive Bayes có thể được sử dụng để phân loại dữ liệu y tế, như xác định liệu một bệnh nhân có mắc một loại bệnh nhất định hay không dựa trên các dữ liệu y tế. Naive Bayes cũng có thể được sử dụng trong các hệ thống phát hiện gian lận, chẳng hạn như phát hiện gian lận trong giao dịch tài chính hoặc giao dịch trực tuyến. Thuật toán này có thể được áp dụng trong việc phân loại tin tức theo chủ đề, như phân loại tin tức thể thao, tin tức chính trị, tin tức khoa học.

## **CHƯƠNG 2: PHƯƠNG PHÁP THỰC HIỆN**

1. **GIỚI THIỆU DEMO**

“Xây dựng hệ thống phân loại các văn bản theo chủ đề” là việc thu thập dữ liệu dạng văn bản, sau đó trải qua bước tiền xử lý dữ liệu để được một tập dữ liệu sạch và được gán nhãn cho mỗi bộ dữ liệu, trích xuất các đặc trưng từ tập dữ liệu, những đặc trưng này sẽ được sử trong mô hình học máy huấn luyện phân loại văn bản và kết quả cuối cùng là một văn bản được gán nhãn đúng với chủ đề hoặc một đoạn văn bản.

Trong đề tài này sử dụng bộ dữ liệu được thu thập qua trang báo BBC nổi tiếng, với năm chủ đề cụ thể là: Business, Entertainment, Politics, Sport, Tech. Với trung bình mỗi bộ dữ liệu con sẽ chứa 445 tệp với đuôi tệp là .txt, trong mỗi tệp .txt chứ các đoạn văn bản chưa được qua bước tiền xử lý dữ liệu cho dạng văn bản. Nhiệm vụ của đề tài này là xác định một đoạn văn bản đầu vào với không nhãn và đầu ra là đoạn văn bản đã được gán nhãn.

1. **MÔI TRƯỜNG VÀ CÁC ĐỐI TƯỢNG SỬ DỤNG XÂY DỰNG DEMO**
2. **Môi trường**

Môi trường được sử dụng để xây dựng hệ thống là Visual Studio Code cùng vơi extension Juputer Notebook.

Ngôn ngữ được sử dụng là Python version 3.12.2.

1. **Các đối tượng sử dụng xây dựng Demo**

import **os**

import **random**

import **string**

import **nltk**

**nltk**.download('punkt')

**nltk**.download('stopwords')

from **nltk** import **word\_tokenize**

from **collections** import **defaultdict**

from **nltk** import **FreqDist**

from **nltk**.**corpus** import stopwords

from **sklearn**.**feature\_extraction**.**text** import **TfidfVectorizer**

from **sklearn**.**feature\_extraction**.**text** import **CountVectorizer**

from **sklearn**.**naive\_bayes** import **MultinomialNB**

from **sklearn** import **metrics**

import **pickle**

Đầu tiên cần import các thư viện cần thiết như os, random, string, nltk và sklearn. Thư viện os được sử dụng để tương tác với hệ điều hành, trong khi random được sử dụng để tạo số ngẫu nhiên. Thư viện nltk là một thư viện phổ biến trong xử lý ngôn ngữ tự nhiên, và trong đoạn mã này, sử dụng để tải xuống các tài nguyên cần thiết như tokenizer và danh sách từ dừng. Thư viện sklearn cung cấp các công cụ để xây dựng và đánh giá các mô hình học máy.

Sau đó, mã tiến hành tiền xử lý dữ liệu văn bản. Các bước tiền xử lý này bao gồm việc tải xuống và tách từ, loại bỏ các từ dừng và các ký tự đặc biệt, và chuẩn hóa văn bản thành dạng thống nhất. Điều này giúp làm sạch và chuẩn hóa dữ liệu để sử dụng cho việc huấn luyện mô hình.

Tiếp theo, mã sử dụng các công cụ từ thư viện sklearn để biến đổi văn bản thành các biểu diễn số hóa. Cụ thể, sử dụng TfidfVectorizer và CountVectorizer để tạo ma trận TF-IDF và ma trận đếm từ từ các văn bản. Các biểu diễn số hóa này sẽ được sử dụng để huấn luyện mô hình phân loại.

Cuối cùng, mã sử dụng mô hình Multinomial Naive Bayes từ sklearn để huấn luyện và phân loại văn bản vào các chủ đề khác nhau. Sau đó, đánh giá hiệu suất của mô hình bằng cách sử dụng các phương pháp đánh giá như accuracy, precision và recall.

1. **CÁC BƯỚC THỰC HIỆN**
2. **Bộ stopword cho văn bản**

Hàm words() từ thư viện stopwords của nltk được sử dụng để tải danh sách các từ dừng trong tiếng Anh. Các từ dừng là các từ phổ biến không mang ý nghĩa và thường bị loại bỏ trong quá trình xử lý văn bản.

Hàm set() được sử dụng để chuyển danh sách các từ dừng thành một tập hợp (set) để tối ưu hóa việc truy xuất và kiểm tra các phần tử. Tập hợp này chứa các từ dừng tiếng Anh.

Hai dòng cuối được sử dụng để thêm các từ dừng bổ sung vào tập hợp stop\_words. Trong trường hợp này, các từ "said" và "mr" được thêm vào danh sách các từ dừng. Điều này giúp loại bỏ các từ này khỏi văn bản trong quá trình tiền xử lý để cải thiện chất lượng của dữ liệu đầu vào cho mô hình phân loại văn bản.

stop\_words = **set**(stopwords.**words**('english'))

stop\_words.**add**('said')

stop\_words.**add**('mr')

1. **Tạo đường dẫn đến bộ dữ liệu và các nhãn cho chủ đề**

LABELS chứa một danh sách các nhãn đại diện cho các chủ đề hoặc loại văn bản. Danh sách này bao gồm các nhãn: 'business', 'entertainment', 'politics', 'sport', và 'tech'. Các nhãn này đại diện cho các chủ đề hoặc loại văn bản mà mô hình phân loại được huấn luyện để nhận diện.

BASE\_DIR ='D:\\Learning\\School\\Semester\_\_6\\COMPUTER\_SCIENCE\_PROJECT\_2\\DEMO\\bbc'

LABELS = ['business', 'entertainment', 'politics', 'sport', 'tech']

1. **Tạo tập dữ liệu với các nhãn**

Hàm create\_data\_set() này sẽ tạo ra một tệp 'data.txt' chứa dữ liệu từ tất cả các tệp tin văn bản trong các thư mục của các nhãn được xác định, với mỗi bản ghi bao gồm nhãn, tên tệp và nội dung của văn bản.

def **create\_data\_set**():

    with **open**('data.txt', 'w', encoding='utf8') as outfile:

        for label in LABELS:

**dir** = '%s/%s' % (BASE\_DIR, label)

            for filename in **os**.**listdir**(**dir**):

                fullfilename = '%s/%s' % (**dir**, filename)

**print**(fullfilename)

                with **open**(fullfilename, 'rb') as file:

                    text = file.**read**().**decode**(errors= 'replace').**replace**('\n', '')

                    outfile.**write**('%s\t%s\t%s\n' % (label, filename, text))

1. **Đọc bộ dữ liệu**

Hàm setup\_docs() để đọc dữ liệu từ tệp data.txt và chuyển đổi thành một danh sách các bộ dữ liệu gồm nhãn và văn bản tương ứng. Hàm này hoạt động bằng cách duyệt qua từng dòng trong tệp data.txt, tách dòng thành các phần bằng dấu phân cách tab \t, và sau đó tạo một bộ dữ liệu mới từ nhãn và văn bản trong mỗi dòng. Cuối cùng, danh sách các bộ dữ liệu này được trả về để sử dụng trong việc huấn luyện và kiểm tra mô hình phân loại văn bản.

Hàm này là một phần quan trọng trong quá trình tiền xử lý dữ liệu trước khi huấn luyện mô hình, vì cho phép tải dữ liệu từ tệp văn bản đã được chuẩn bị trước và biến đổi thành một định dạng phù hợp để sử dụng cho các bước tiếp theo trong quy trình xây dựng mô hình phân loại văn bản.

def **setup\_docs**():

    docs = []

    with **open**('data.txt', 'r', encoding='utf8') as datafile:

        for row **in** datafile:

            parts = row.**split**('\t')

            doc = ( parts[0], parts[2].**strip**() )

            docs.**append**(doc)

        return docs

1. **Làm sạch văn bản**

Đoạn mã trên định nghĩa một hàm clean\_text() để thực hiện việc làm sạch văn bản đầu vào trước khi tiến hành tiền xử lý. Hàm này thực hiện hai bước chính để chuẩn bị dữ liệu cho việc xử lý tiếp theo.

Đầu tiên là loại bỏ các dấu câu sử dụng phương thức translate() để loại bỏ các dấu câu từ văn bản. Các dấu câu được thay thế bằng khoảng trắng, giúp làm sạch văn bản và loại bỏ yếu tố không cần thiết trong quá trình xử lý.

Kế tiếp chuyển đổi văn bản thành chữ thường hàm lower() được sử dụng để chuyển đổi tất cả các ký tự trong văn bản thành chữ thường. Điều này giúp đồng nhất hóa văn bản, loại bỏ sự phân biệt giữa chữ hoa và chữ thường trong quá trình xử lý dữ liệu.

Hàm clean\_text() trả về văn bản sau khi đã được làm sạch và chuẩn hóa, sẵn sàng cho các bước tiền xử lý tiếp theo như tách từ và loại bỏ từ dừng. Điều này giúp cải thiện chất lượng của dữ liệu và làm cho quá trình xây dựng mô hình phân loại văn bản trở nên hiệu quả hơn.

def **clean\_text**(text):

    text = text.translate(**str**.**maketrans**('', '', **string**.punctuation))

    text = text.lower()

    return text

1. **Danh sách tần suất xuất hiện của các từ trong từ điển mỗi danh mục văn bản**

Xây dựng hàm print\_frequency\_dist(docs) để thực hiện việc in ra tần suất xuất hiện của các từ trong từ điển của mỗi danh mục văn bản. Quá trình này giúp phân phối từ vựng trong từng danh mục văn bản và có cái nhìn tổng quan về các từ phổ biến trong mỗi danh mục.

Khởi tạo từ điển và xây dựng danh sách từ cho mỗi danh mục, khởi tạo một từ điển mặc định defaultdict với các giá trị ban đầu là danh sách rỗng. Trong từ điển này, mỗi danh mục văn bản sẽ có một danh sách các từ xuất hiện trong các văn bản thuộc danh mục đó.

Tiếp theo, với mỗi văn bản trong danh sách docs, lấy nhãn của văn bản và làm sạch văn bản bằng cách loại bỏ dấu câu và chuẩn hóa chữ viết. Sau đó tách văn bản đã làm sạch thành các từ và thêm chúng vào danh sách từ tương ứng với nhãn của văn bản trong từ điển.

In ra tần suất xuất hiện của từng từ trong từ điển, duyệt qua từng cặp nhãn và danh sách từ trong từ điển. Sử dụng FreqDist từ thư viện nltk để tính toán tần suất xuất hiện của mỗi từ trong danh sách từ. In ra số lượng từ phổ biến nhất và tần suất xuất hiện của các từ trong danh mục văn bản tương ứng.

Hàm print\_frequency\_dist() là một phần quan trọng trong quá trình phân tích dữ liệu văn bản, giúp hiểu được cấu trúc và nội dung của từng danh mục văn bản. Điều này cung cấp thông tin hữu ích cho quá trình tiền xử lý và xây dựng mô hình phân loại văn bản.

def **print\_frequency\_dist**(docs):

    tokens = **defaultdict**(**list**)

    for doc in docs:

        doc\_label = doc[0]

        doc\_text = **clean\_text**(doc[1])

        doc\_tokens = **get\_tokens**(doc\_text)

        tokens[doc\_label].**extend**(doc\_tokens)

    for category\_label, category\_tokens **in** tokens.**items**():

**print**(category\_label)

        fd = **FreqDist**(category\_tokens)

**print**(fd.**most\_common**(5))

1. **Tách từ và loại bỏ stopword**

Hàm get\_tokens() để thực hiện hai công việc chính là tách từ và loại bỏ các stopwords từ văn bản đầu vào. Quá trình tiền xử lý này là một phần quan trọng trong quá trình xây dựng mô hình phân loại văn bản, nhằm làm sạch và chuẩn bị dữ liệu để có thể được sử dụng hiệu quả trong việc huấn luyện mô hình.

Hàm sử dụng word\_tokenize() từ thư viện nltk để tách văn bản thành các từ hoặc token riêng biệt, tạo ra một danh sách các token.

Sau đó, hàm sử dụng một biểu thức danh sách để lọc và loại bỏ các từ dừng từ danh sách các token đã tách ra từ văn bản. Việc này giúp loại bỏ các từ không mang ý nghĩa và thường không cần thiết trong việc phân loại văn bản.

Kết quả là hàm trả về danh sách các token đã được làm sạch và loại bỏ các từ dừng, sẵn sàng cho các bước tiền xử lý tiếp theo như vectơ hóa hoặc huấn luyện mô hình.

Hàm get\_tokens() là một bước quan trọng trong quy trình tiền xử lý dữ liệu văn bản, đảm bảo rằng dữ liệu được chuẩn bị một cách chính xác và hiệu quả cho việc xây dựng mô hình phân loại văn bản.

def **get\_tokens**(text):

    tokens = **word\_tokenize**(text)

    tokens = [t for t in tokens if not t in stop\_words]

    return tokens

1. **Chia tập dữ liệu**

Hàm get\_splits(docs) để chia tập dữ liệu thành tập huấn luyện và tập kiểm tra, sao cho 80% dữ liệu được sử dụng cho việc huấn luyện và 20% dữ liệu được sử dụng cho việc kiểm tra.

Trộn dữ liệu và khởi tạo danh sách huấn luyện và kiểm tra, sử dụng hàm random.shuffle() để trộn ngẫu nhiên danh sách dữ liệu docs, đảm bảo tính ngẫu nhiên khi chia tập dữ liệu. Sau đó khởi tạo các danh sách rỗng để lưu trữ dữ liệu và nhãn tương ứng cho tập huấn luyện và tập kiểm tra.

Chia tập dữ liệu, sử dụng chỉ mục pivot để xác định điểm chia tách giữa tập huấn luyện và tập kiểm tra, với 80% dữ liệu được sử dụng cho việc huấn luyện.

Duyệt qua danh sách dữ liệu docs và thêm dữ liệu và nhãn tương ứng vào tập huấn luyện cho các chỉ mục từ 0 đến pivot - 1.

Thêm dữ liệu và nhãn tương ứng vào tập kiểm tra cho các chỉ mục từ pivot đến cuối danh sách. Trả về các danh sách X\_train, X\_test, y\_train, y\_test chứa dữ liệu và nhãn tương ứng của tập huấn luyện và tập kiểm tra.

Hàm get\_splits() là một bước quan trọng trong quy trình chuẩn bị dữ liệu cho việc huấn luyện và kiểm tra mô hình phân loại văn bản. Điều này có thể đánh giá hiệu suất của mô hình trên tập dữ liệu độc lập để đảm bảo tính khách quan và chính xác của mô hình.

def **get\_splits**(docs):

**random**.shuffle(docs)

    X\_train = []

    y\_train = []

    X\_test = []

    y\_test = []

    pivot = **int**(.80 \* **len**(docs))

    for i in **range**(0, pivot):

        X\_train.**append**(docs[i][1])

        y\_train.**append**(docs[i][0])

    for i in **range**(pivot, **len**(docs)):

        X\_test.**append**(docs[i][1])

        y\_test.**append**(docs[i][0])

    return X\_train, X\_test, y\_train, y\_test

1. **Tạo hàm đánh giá hiệu suất**

Hàm evaluate\_classifier(title, classifier, vectorizer, X\_test, y\_test) để đánh giá hiệu suất của một mô hình phân loại trên tập dữ liệu kiểm tra.

Biến đổi dữ liệu kiểm tra, hàm vectorizer đã được huấn luyện để biến đổi tập dữ liệu kiểm tra X\_test thành ma trận TF-IDF. Điều này đảm bảo rằng đầu vào cho mô hình có cùng định dạng với dữ liệu mà mô hình đã được huấn luyện trên.

Sau khi có ma trận TF-IDF của dữ liệu kiểm tra, hàm sử dụng mô hình phân loại đã được huấn luyện classifier để dự đoán nhãn cho từng mẫu dữ liệu trong tập kiểm tra.

Đánh giá hiệu suất của mô hình, hàm tính toán các chỉ số đánh giá bao gồm độ chính xác (precision), chỉ số recall và F1-score bằng cách so sánh nhãn dự đoán với nhãn thực tế trên tập dữ liệu kiểm tra.

Các chỉ số này được tính toán thông qua hàm `precision\_score`, `recall\_score`, và `f1\_score` từ thư viện sklearn.metrics. Cuối cùng hàm in ra tiêu đề của phân loại cùng với các chỉ số đánh giá precision, recall và F1-score. Giúp đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra.

Hàm evaluate\_classifier() là một công cụ quan trọng để đánh giá hiệu suất của mô hình phân loại văn bản trên tập dữ liệu kiểm tra và đưa ra thông tin về độ chính xác, độ phủ và F1-score của mô hình. Điều này giúp có cái nhìn tổng quan về khả năng phân loại của mô hình trên dữ liệu mới mà chưa từng nhìn thấy.

def **evaluate\_classifier**(title, classifier, vectorizer, X\_test, y\_test):

    X\_test\_tfidf = vectorizer.transform(X\_test)

    y\_pred = classifier.predict(X\_test\_tfidf)

    precision = **metrics**.**precision\_score**(y\_test, y\_pred, average='macro')

    recall = **metrics**.**recall\_score**(y\_test, y\_pred, average='macro')

    f1 = **metrics**.**f1\_score**(y\_test, y\_pred, average='macro')

**print**("%s\t%f\t%f\t%f\n" % (title, precision, recall, f1))

1. **Xây dựng mô hình huấn huyện Naive Bayes**

Hàm train\_classifier(docs) để huấn luyện một mô hình phân loại trên tập dữ liệu được cung cấp và lưu trữ mô hình và vectorizer để sử dụng cho dữ liệu mới.

Hàm get\_splits() để chia tập dữ liệu docs thành tập huấn luyện và tập kiểm tra. Biến đổi dữ liệu và huấn luyện mô hình, sử dụng CountVectorizer để biến đổi các văn bản thành các vectơ đặc trưng. Tạo ma trận văn bản đặc trưng từ dữ liệu huấn luyện bằng cách sử dụng phương thức fit\_transform() của vectorizer.

Khởi tạo và huấn luyện một mô hình phân loại Naive Bayes trên ma trận văn bản đặc trưng. Đánh giá hiệu suất mô hình sử dụng hàm evaluate\_classifier() để đánh giá hiệu suất của mô hình trên cả tập huấn luyện và tập kiểm tra. Kết quả được in ra để đánh giá độ chính xác, độ phủ và F1-score của mô hình.

Lưu trữ mô hình và vectorizer sử dụng pickle.dump() để lưu trữ mô hình phân loại đã được huấn luyện vào một tệp. Lưu trữ vectorizer cũng được thực hiện tương tự.

Hàm train\_classifier() là một bước quan trọng trong quá trình xây dựng và triển khai mô hình phân loại văn bản. Việc này giúp huấn luyện mô hình trên dữ liệu có sẵn và lưu trữ mô hình và vectorizer để có thể sử dụng lại trong tương lai cho việc phân loại dữ liệu mới.

def **train\_classifier**(docs):

    X\_train, X\_test, y\_train, y\_test = **get\_splits**(docs)

    vectorizer = **CountVectorizer**(stop\_words='english', ngram\_range=(1, 3), min\_df=3, analyzer='word')

    dtm = vectorizer.**fit\_transform**(X\_train)

    naive\_bayes\_classifier = **MultinomialNB**().**fit**(dtm, y\_train)

**evaluate\_classifier**("Naive Bayes\tTRAIN\t", naive\_bayes\_classifier, vectorizer, X\_train, y\_train)

**evaluate\_classifier**("Naive Bayes\tTEST\t", naive\_bayes\_classifier, vectorizer, X\_test, y\_test)

    clf\_filename = 'naive\_bayes\_classifier.pkl'

**pickle**.**dump**(naive\_bayes\_classifier, **open**(clf\_filename, 'wb'))

    vec\_filename = 'count\_vectorizer.pkl'

**pickle**.**dump**(vectorizer, **open**(vec\_filename, 'wb'))

1. **Hàm phân loại cho văn bản đầu vào**

Hàm classify(text) để phân loại một đoạn văn bản mới vào một trong các nhãn của mô hình phân loại đã được huấn luyện.

Load mô hình và vectorizer sử dụng pickle.load() để load mô hình phân loại đã được lưu trữ từ tệp và gán cho biến nb\_clf.

Tương tự, cũng load vectorizer từ tệp và gán cho biến vectorizer. Chuyển đổi đoạn văn bản mới thành vectơ đặc trưng và dự đoán nhãn. Sử dụng vectorizer để chuyển đổi đoạn văn bản mới thành vectơ đặc trưng, bằng cách sử dụng phương thức transform() của vectorizer.

Sau đó, sử dụng mô hình phân loại để dự đoán nhãn của đoạn văn bản này bằng cách sử dụng phương thức predict().

Kết quả phân loại được in ra màn hình với chuỗi " Văn bản được phân loại: " cùng với nhãn được dự đoán cho đoạn văn bản. Hàm classify() giúp mô hình phân loại đã huấn luyện để dự đoán nhãn của một đoạn văn bản mới và in ra kết quả phân loại.

def **classify**(text):

    clf\_filename = 'D:\Learning\School\Semester\_\_6\COMPUTER\_SCIENCE\_PROJECT\_2\DEMO\naive\_bayes\_classifier.pkl'

    nb\_clf = **pickle**.**load**(**open**(clf\_filename, 'rb'))

    vec\_filename = 'D:\Learning\School\Semester\_\_6\COMPUTER\_SCIENCE\_PROJECT\_2\DEMO\count\_vectorizer.pkl'

    vectorizer = **pickle**.**load**(**open**(vec\_filename, 'rb'))

    pred = nb\_clf.predict(vectorizer.transform([text]))

**print**(f"\n\nVăn bản được phân loại: {**str**(pred[0]).**upper**()}\n\n")

1. **Hàm Main**

if \_\_name\_\_ == '\_\_main\_\_':

**create\_data\_set**()

    docs = **setup\_docs**()

**print\_frequency\_dist**(docs)

**train\_classifier**(docs)

    new\_doc = """ """

**classify**(new\_doc)

## **CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ**

1. **KẾT QUẢ THỰC NGHIỆM**
2. **Bộ dữ liệu được tạo**

* **Tập văn bản “business”**

**A close-up of a text

Description automatically generated**

* **Tập văn bản “entertainment”**

**A screenshot of a computer

Description automatically generated**

* **Tập văn bản “politics”**

**A screenshot of a computer

Description automatically generated**

* **Tập văn bản “sport”**

**A screenshot of a computer screen

Description automatically generated**

* **Tập văn bản “tech”**

**A screenshot of a computer

Description automatically generated**

1. **Các từ được tính tần suất xuất hiện**

**A screenshot of a computer code

Description automatically generated**

1. **Kết quả huấn luyện và đánh giá mô hình**

**A close-up of numbers

Description automatically generated**

1. **ĐÁNH GIÁ**

Hệ thống phân loại văn bản theo chủ đề sử dụng mô hình Naive Bayes và kỹ thuật TF-IDF đã đạt được kết quả rất khả quan với năm chủ đề. Kết quả huấn luyện đạt 0.99 và kết quả đánh giá mô hình đạt 0.967 chứng tỏ hệ thống có khả năng phân loại chính xác và hiệu quả.

Việc áp dụng Naive Bayes và TF-IDF đã cho thấy tính ưu việt trong việc xử lý và phân loại văn bản, đảm bảo độ tin cậy và tính tổng quát hóa cao. Những kết quả này khẳng định tiềm năng và ứng dụng thực tiễn của hệ thống trong lĩnh vực xử lý ngôn ngữ tự nhiên, đồng thời mở ra hướng nghiên cứu và cải tiến tiếp theo nhằm nâng cao hiệu suất và mở rộng phạm vi ứng dụng.

# **PHẦN KẾT LUẬN**

1. **KẾT LUẬN**

Sau khi tìm hiểu và thực hiện đề tài “XÂY DỰNG HỆ THỐNG PHÂN LOẠI VĂN BẢN THEO CHỦ ĐỀ” chúng em đã hoàn thành một hệ thống phân loại văn bản theo một số chủ đề cho trước như thể thao, doanh nghiệp,...

Hiểu được những lý thuyết cơ bản trong quá trình thực hiện như:

* Nắm được kiến thức cơ bản về hệ thống phân loại.
* Kiến thức về NLP, Machine Learning, TF-IDF, mô hình Naïve Bayes.
* Cách sử lý dữ liệu cũng như huấn luyện mô hình phân loại.

1. **HƯỚNG PHÁT TRIỂN**

Trong phạm vi đồ án lần này chúng em đã trình bày được những thành phần cơ bản nhất về hệ thống phân loại văn bản theo chủ đề. Tuy việc nghiên cứu chưa ứng dụng vào thực tiễn một cách chuyên sâu và bài bản nhưng cơ bản hệ thống có thể phân loại được dữ liệu với độ chính xác khá ổn định theo một số chủ đề đã được huấn luyện từ trước.

So với kết quả hiện tại thì hệ thống phân loại văn bản của chúng em vẫn còn nhiều hạn chế nhưng với việc tìm tòi học hỏi thì chương trình này vẫn có thể phát triển ổn định hơn thông qua các hướng phát triển như sau:

* Mở rộng thêm dữ liệu trong bộ dữ liệu huấn luyện để tăng thêm độ chính xác trong quá trình phân loại.
* Thêm nhiều chủ đề ở nhiều lĩnh vực khác nhau làm đa dạng nguồn dữ liệu hơn.
* Xây dựng thêm giao diện thân thiện với người dùng để dễ dàng thao tác trong quá trình sử dụng.

# **TÀI LIỆU KHAM KHẢO**

|  |  |
| --- | --- |
| [1] | A. Hodges, "Alan Turing (Stanford Encyclopedia Of Philosophy)," Jun. 3, 2002. [Online]. Available: https://plato.stanford.edu/Entries/turing/. [Accessed: Nov. 14, 2023]. |
| [2] | P. T. H. Q. Thụy, P. T. P. X. Hiếu, P. T. N. T. Thành, and G. T. N. T. Thủy, "Trí tuệ nhân tạo trong thời đại số: Bối cảnh thế giới và liên hệ với Việt Nam," 2018. [Online]. Available: https://uet.vnu.edu.vn/~thuyhq/PPNCKH/AI.pdf. [Accessed: Nov. 14, 2023]. |
| [3] | T. Bayes, "Naive Bayes classifier," Article Sources and Contributors, 1968, pp. 1-9. |
| [4] | I. Rish, "An empirical study of the naive Bayes classifier," in IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, vol. 3, no. 22, pp. 41-46, Aug. 4, 2001. |
| [5] | G. I. Webb, E. Keogh, and R. Miikkulainen, "Naïve Bayes," Encyclopedia of Machine Learning, vol. 15, no. 1, pp. 713-714, 2010. |
| [6] | J. Ramos, "Using tf-idf to determine word relevance in document queries," in Proceedings of the First Instructional Conference on Machine Learning, vol. 242, no. 1, pp. 29-48, Dec. 3, 2003. |
| [7] | K. Chowdhary and K. R. Chowdhary, "Natural language processing," Fundamentals of Artificial Intelligence, pp. 603-649, 2020. |