KHOA KỸ THUẬT VÀ CÔNG NGHỆ

BỘ MÔN CÔNG NGHỆ THÔNG TIN

THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH

HỌC KỲ 1 , NĂM HỌC 2023 – 2024

Tên đề tài:

NGHIÊN CỨU THUẬT TOÁN GỢI Ý THEO PHƯƠNG PHÁP LỌC THEO NỘI DUNG VÀ XÂY DỰNG ỨNG DỤNG MINH HOẠ

Giảng viên hướng dẫn:

Họ tên: Phan Thị Phương Nam

Sinh viên thực hiện:

Họ tên: Kim Thanh Ái Nhân

MSSV: 110120146

Lớp: DA20TTB

Trà Vinh, tháng 01 năm 2024

KHOA KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN

THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH
HỌC KỲ 1 , NĂM HỌC 2023 – 2024

Tên đề tài:

NGHIÊN CỨU THUẬT TOÁN GỢI Ý THEO PHƯƠNG PHÁP LỌC THEO NỘI DUNG VÀ XÂY DỰNG ỨNG DỤNG MINH HOẠ

Giảng viên hướng dẫn:
Họ tên: Phan Thị Phương Nam

Sinh viên thực hiện:
Họ tên: Kim Thanh Ái Nhân
MSSV: 110120146
Lớp: DA20TTB

Trà Vinh, tháng 01 năm 2024

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Trà Vinh, ngày ..... tháng …… năm ……

Giáo viên hướng dẫn
(Ký tên và ghi rõ họ tên)

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Trà Vinh, ngày ..... tháng …… năm ……

Thành viên hội đồng
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Tôi xin được gửi lời cảm ơn đến quý thầy cô Khoa Kỹ thuật và Công nghệ vì những kiến thức đã được học tập từ quý thầy cô, cũng như các kiến thức trong quá trình tự tìm hiểu và đặc biệt với sự hướng dẫn nhiệt tình của cô Phan Thị Phương Nam, tôi đã hoàn thành được đề tài “Nghiên cứu thuật toán gợi ý theo phương pháp lọc theo nội dung và xây dựng ứng dụng minh hoạ.”. Tôi rất mong nhận được sự quan tâm, góp ý của các thầy cô để đề tài của tôi được hoàn chỉnh và đầy đủ hơn.

Tôi xin chân thành cảm ơn.

Sinh viên thực hiện

Kim Thanh Ái Nhân

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN 10
1.1 Tổng quan về phương pháp gợi ý. 10
1.1.1 Tương Đồng Jaccard: Đo Lường Sự Tương Đồng giữa Các Tập Hợp 11
1.1.2 Shingling: Mô Hình Hóa Văn Bản và Đo Lường Sự Tương Đồng 12
1.2 Tầm quan trọng và ứng dụng của hệ thống đề xuất. 13
1.2.1 Ứng dụng trong thực tế. 13
1.2.2 Vai trò trong kinh doanh. 13
1.3 Các loại phương pháp gợi ý phổ biến. 14
1.3.1 Gợi ý dựa trên nội dung (Content-Based Recommendation). 14
1.3.2 Gợi ý dựa trên hành vi người dùng (Collaborative Filtering). 15
1.3.3 Phương pháp kết hợp (Hybrid Methods). 16
1.4 Thách thức và cơ hội trong phát triển phương pháp gợi ý 16
1.4.1 Thách thức về đa dạng và mất mát thông tin: 16
1.4.2 Cơ hội với sự phát triển của học máy và học sâu: 16
1.4.3 Tương tác người dùng tăng cường: 17
1.4.4 Bảo mật và quyền riêng tư: 17
1.4.5 Học máy gia Reinforcement (Học máy tăng cường): 17
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT 18
2.1 Định nghĩa và nguyên tắc hoạt động cơ bản về lọc theo nội dung. 18
2.1.1 Định nghĩa. 18
2.1.2 Nguyên tắc hoạt động. 18
2.1.3 Ưu điểm và Ứng dụng: 19
2.2 Lịch sử và phát triển của phương pháp: 20
2.2.1 Xuất hiện ban đầu: 20
2.2.2 Phát triển qua các giai đoạn: 20
2.2.3 Ứng dụng và sự lan rộng: 20
2.2.4 Thách thức và Hướng Phát Triển: 20
2.3 Nguyên lý hoạt động của phương pháp lọc theo nội dung. 21
2.3.1 Nguyên tắc cơ bản của phương pháp lọc theo nội dung. 21
2.3.1.1 Hiểu biết về sản phẩm. 21
2.3.1.2 Xây dựng hồ sơ người dùng. 21
2.3.1.3 So khớp nội dung và hồ sơ người dùng. 21
2.3.1.4 Tạo ra đề xuất cá nhân hóa. 21
2.3.1.5 Bài toán ví dụ về nguyên tắc hoạt động của hệ thống gợi ý theo nội dung. 22
2.3.2 Cơ sở toán học và thuật toán. 25
2.3.2.1 Vector hóa sản phẩm và người dùng. 25
2.3.2.2 Sử dụng các thuật toán. 25
2.3.2.3 Mô hình học sâu. 25

2.4 Các hệ thống và ứng dụng thực tế. 25
2.4.1 Áp dụng trong thương mại điện tử:.....25
2.4.2 Nền tảng bán lẻ trực tuyến: 25
2.4.3 Tích hợp trong streaming âm nhạc: 26
2.4.5 Ứng dụng di động và cửa hàng ứng dụng:.....26
2.4.6 Giáo dục trực tuyến:.....26
2.4.7 Du lịch và giải trí: 27
2.5 Một số thuật toán tiêu biểu trong phương pháp lọc theo nội dung. 27
2.5.1 Thuật toán TF-IDF (Term Frequency-Inverse Document Frequency):.....27
2.5.1.1 Mô tả thuật toán TF-IDF: 27
2.5.1.2 Nguyên lý hoạt động thuật toán TF-IDF:.....27
2.5.1.3 Mục tiêu thuật toán TF-IDF: 27
2.5.1.4 Ưu và nhược điểm thuật toán TF-IDF:.....28
2.5.2 Thuật toán Word Embeddings: 28
2.5.2.1 Mô tả thuật toán Word Embeddings: 28
2.5.2.2 Nguyên lý hoạt động thuật toán Word Embeddings: 28
2.5.2.3 Mục tiêu thuật toán Word Embeddings:.....29
2.5.2.4 Ưu và nhược điểm thuật toán Word Embeddings:.....29
2.6 Thuật toán TF-IDF và bài toán ví dụ. 29
2.6.1 TF-IDF trong xử lý ngôn ngữ tự nhiên và thông tin truy xuất: 29
2.6.2 Bài toán ví dụ về thuật toán TF-IDF. 30
2.7 Ứng dụng và ngôn ngữ thực hiện đề tài.....32
2.7.1 Anaconda Navigator.....32
2.7.2 Jupyter Notebook. 33
2.7.3 Ngôn ngữ lập trình Python. 33
2.7.4 Flask framework.....33
CHƯƠNG 3: ĐÁNH GIÁ KẾT QUẢ.....35
3.1 Mô tả cấu trúc dữ liệu minh họa của đề tài.....35
3.2 Xử lý dữ liệu 35
3.3 Xây dựng mô hình 37
3.4 Xây dựng ứng dụng minh họa. 41
3.5 Giao diện ứng dụng minh họa.....44
CHƯƠNG 4: KẾT LUẬN.....47
4.1 Kết quả đạt được. 47
4.2 Hạn chế. 47
CHƯƠNG 5: HƯỚNG PHÁT TRIỂN.....48
DANH MỤC TÀI LIỆU THAM KHẢO 49

DANH MỤC HÌNH ẢNH – BẢNG BIỂU

Hình 1. Minh họa quá trình đề xuất....10
Hình 2. Hai tập hợp A và B có bộ Jaccard là 3/8....11
Hình 3. Kỹ thuật lọc theo nội dung....14
Hình 4. Kỹ thuật lọc theo hành vi....15
Hình 5. Ví dụ về hệ thống đề xuất....19
Hình 6. Chọn thuộc tính của cơ sở dữ liệu....35
Hình 7. Thay chuỗi "unknown" vào giá trị khuyết....36
Hình 8. Xử lí dữ liệu trong cột thể loại và tiêu đề....36
Hình 9. Xử lí thuộc tính tiêu đề phim và lưu dữ liệu....37
Hình 10. Cài đặt WordNet và đọc dữ liệu....37
Hình 11. Kết hợp dữ liệu và gán giá trị cho biến X và Y....38
Hình 12. Số lượng mẫu trong biến Y....38
Hình 13. Xóa lớp ít mẫu và sử dụng EDA....39
Hình 14. Tăng cường từ đồng nghĩa và kết hợp dữ liệu....39
Hình 15. Sử dụng TfidfVectorizes chuyển đổi ma trận....40
Hình 16. Chia dữ liệu thành tập train và test....40
Hình 17. Đánh giá độ chính xác và lưu mô hình....41
Hình 18. Cài đặt thư viện là load mô hình....41
Hình 19. Tạo bảng từ khóa....42
Hình 20. Chuyển đổi thành ma trận TF-IDF....42
Hình 21. Tính ma trận tương đồng....43
Hình 22. Hàm gợi ý và kiểm tra chỉ số hợp lệ....43
Hình 23. Hàm xử lý tìm kiếm....44
Hình 24. Hàm lấy thông tin phim....44
Hình 25. Giao diện tìm kiếm....45
Hình 26. Giao diện tìm kiếm phim có tên "avatar"....46
Hình 27. Giao diện phim chưa có trên hệ thống....46

Bảng 1. Bộ dữ liệu phim ví dụ....22
Bảng 2. Xếp hạng người dùng....22
Bảng 3. Ma trận thể loại phim người dùng đã xem....23
Bảng 4. Ma trận trọng số thể loại....23
Bảng 5. Hồ sơ người dùng....23
Bảng 6. Ma trận thể loại phim người dùng chưa xem....24
Bảng 7. Trọng số ma trận đề xuất....24
Bảng 8. Chỉ số đề xuất....24
Bảng 9. Bảng tần xuất xuất hiện các từ trong câu (TF)....31
Bảng 10. Bảng tần xuất nghịch đảo của tài liệu (IDF)....31
Bảng 11. Tầm quan trọng của từ trong ngữ cảnh....32

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

1. **Vấn đề nghiên cứu:**
   - Tìm hiểu tổng quan về thuật toán lọc theo nội dung,
   - Tìm hiểu các phương pháp lọc theo nội dung,
   - Tìm hiểu và cài đặt thuật toán.
2. **Hướng tiếp cận:**
   - Tìm hoặc xây dựng tập dữ liệu thử nghiệm,
   - Xây dựng ứng dụng minh họa cho thuật toán.
3. **Hướng giải quyết:**
   - Nghiên cứu tài liệu về các thuật toán liên quan đến phương pháp gợi ý,
   - Nghiên cứu thuật toán TF – IDF,
   - Nghiên cứu tập dữ liệu thử nghiệm phù hợp để cài đặt thuật toán và tiến hành xây dựng ứng dụng minh họa.
4. **Kết quả đạt được:**
   - Hoàn thành các nội dung của đề tài "Nghiên cứu thuật toán gợi ý theo phương pháp lọc theo nội dung và xây dựng ứng dụng minh hoạ.".

MỞ ĐẦU

1. Lý do chọn đề tài:

Trải qua sự phát triển nhanh chóng của công nghệ và sự gia tăng đáng kể về lượng thông tin, việc tìm kiếm thông tin chính xác và sản phẩm phù hợp trở nên ngày càng khó khăn đối với người dùng. Đặc biệt, trong môi trường số ngày nay, khi người dùng đối mặt với sự dư thừa thông tin, một hệ thống gợi ý thông minh dựa trên nội dung trở thành một yếu tố quan trọng để giảm bớt gánh nặng tìm kiếm và cung cấp trải nghiệm cá nhân hóa.

Việc cải thiện trải nghiệm người dùng không chỉ là một mục tiêu cá nhân mà còn là chìa khóa mở cửa cho các doanh nghiệp tạo ra một môi trường thuận lợi và thuận tiện. Thuật toán gợi ý dựa trên nội dung không chỉ giúp người dùng tiết kiệm thời gian mà còn tăng cường sự tương tác và tìm thấy những thông tin hoặc sản phẩm mà họ có thể chưa biết đến, tạo nên một trải nghiệm dựa trên sự cá nhân hóa và chất lượng.

Thông qua đề tài "Nghiên cứu thuật toán gợi ý theo phương pháp lọc theo nội dung và xây dựng ứng dụng minh họa," tôi hi vọng hiểu sâu hơn về cách thiết kế, triển khai và tối ưu hóa các thuật toán gợi ý, đồng thời xây dựng một ứng dụng minh họa để thấy rõ ứng dụng thực tế của kiến thức đã nghiên cứu. Điều này không chỉ nâng cao kỹ năng kỹ thuật mà còn mang lại cái nhìn chi tiết về sức mạnh và tiềm năng ứng dụng của lọc nội dung trong các hệ thống thông tin hiện đại.

2. Mục đích nghiên cứu:

Nghiên cứu tài liệu về các thuật toán liên quan đến phương pháp gợi ý.

Nghiên cứu tài liệu về phương pháp lọc theo nội dung.

Cài đặt thuật toán với tập dữ liệu thử nghiệm.

Xây dựng ứng dụng minh họa cho thuật toán.

3. Đối tượng:

Các phương pháp lọc theo nội dung.

Tìm hiểu thuật toán TF - IDF.

Cài đặt thuật toán TF - IDF.

Tìm hiểu sử dụng thuật toán.

4. Phạm vi nghiên cứu:

Tìm hoặc xây dựng tập dữ liệu thử nghiệm phù hợp với thuật toán.

Cài đặt thuật toán với tập dữ liệu thử nghiệm.

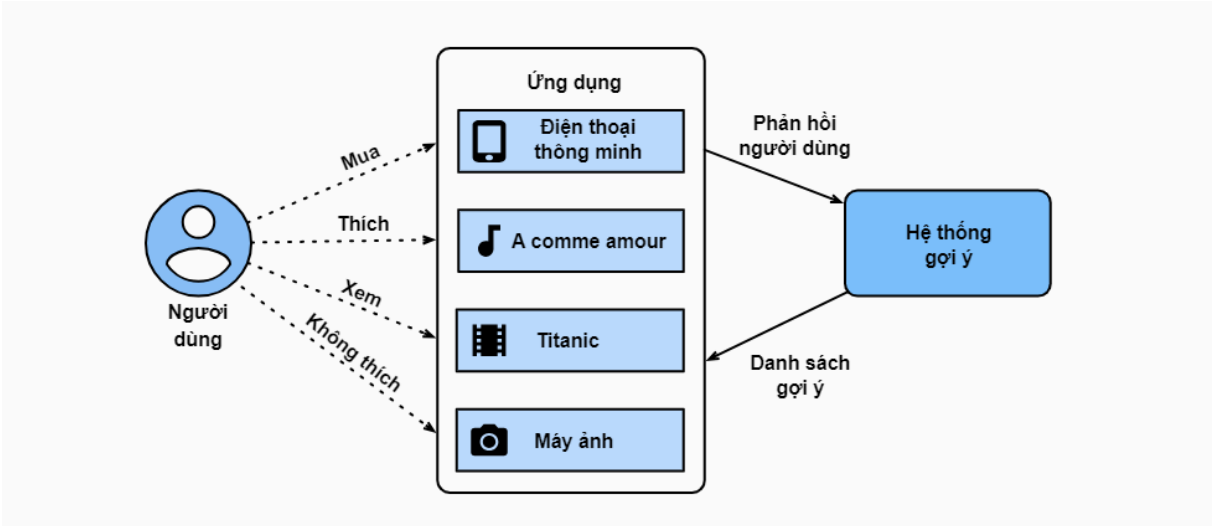
Xây dựng ứng dụng minh họa cho thuật toán với chức năng tìm kiếm cơ bản.

CHƯƠNG 1: TỔNG QUAN

1.1 Tổng quan về phương pháp gợi ý.

Hệ thống đề xuất được sử dụng một cách rộng rãi trong kinh doanh và luôn hiện diện trong cuộc sống hàng ngày của chúng ta. Những hệ thống này được tận dụng trong nhiều lĩnh vực như thương mại điện tử (như amazon.com), các dịch vụ âm nhạc / điện ảnh (như Netflix và Spotify), cửa hàng ứng dụng di động (như App Store và Google Play), quảng cáo trực tuyến, v.v.

Mục đích chính của các hệ thống đề xuất là giúp người dùng tìm ra những sản phẩm liên quan như phim để xem, văn bản để đọc hay hàng hóa để mua, nhằm tạo nên một trải nghiệm thú vị cho người dùng. Hơn nữa, hệ thống đề xuất là một trong những hệ thống máy học mạnh mẽ nhất mà các công ty bán lẻ áp dụng với mục đích tăng doanh thu. Hệ thống đề xuất là công cụ thay thế cho các công cụ tìm kiếm bằng cách giảm nỗ lực tìm kiếm chủ động và tăng cơ hội tiếp cận của người dùng với những đề xuất mà họ không bao giờ tìm đến. Rất nhiều công ty đã vượt lên trên các đối thủ nhờ có hệ thống đề xuất hiệu quả hơn. Do đó, hệ thống đề xuất đã trở thành trung tâm không chỉ trong cuộc sống hàng ngày của chúng ta mà còn có vai trò quan trọng trong một số lĩnh vực kinh doanh.



Hình 1. Minh họa quá trình đề xuất

1.1.1 Tương Đồng Jaccard: Đo Lường Sự Tương Đồng giữa Các Tập Hợp

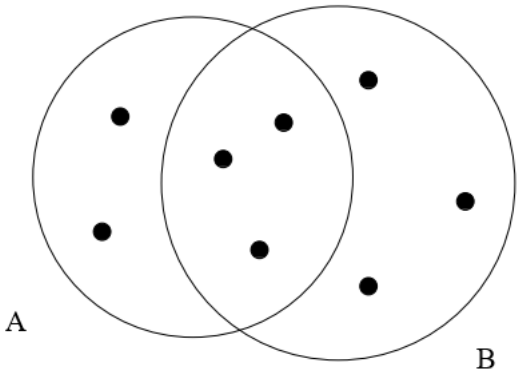
Tương đồng Jaccard là một khái niệm quan trọng trong việc đo lường sự tương đồng giữa các tập hợp dựa trên tỷ lệ giữa kích thước của giao điểm và kích thước của hợp của chúng. Công thức đơn giản của nó là một phép toán chia, nhưng ứng dụng của nó mạnh mẽ và đa dạng.

- Định Nghĩa:

Tương đồng Jaccard giữa hai tập hợp A và B được tính bằng cách lấy kích thước của giao điểm |A∩B| chia cho kích thước của hợp |A∪B|. Công thức tổng quát được biểu diễn như sau:

$$J(A,B)=\frac{\text{kích thước giao của hai tập hợp}}{\text{kích thước hợp của hai tập hợp}} \quad [2]$$

Đây là một phép đo đơn giản nhưng mạnh mẽ, thường xuyên được sử dụng trong nhiều ngữ cảnh để hiểu rõ mức độ tương đồng giữa các tập hợp.



Hình 2. Hai tập hợp A và B có bộ Jaccard là 3/8

- Ứng Dụng:

- Đo Lường Sự Tương Đồng Văn Bản: Tương đồng Jaccard là một công cụ hữu ích trong việc so sánh sự tương đồng giữa các văn bản. Thay vì dựa vào cấu trúc hay ngữ pháp, nó tập trung vào tập từ vựng xuất hiện trong các văn bản. Điều này giúp xác định mức độ tương đồng giữa các nội dung với chi phí tính toán thấp.

- Phân Loại Hành Vi Mua Sắm: Trong lĩnh vực phân loại hành vi mua sắm, tương đồng Jaccard được sử dụng để đo lường sự tương đồng giữa các hành vi mua sắm của các khách hàng. Các sản phẩm được coi là tương đồng nếu chúng xuất hiện trong các đơn đặt hàng của cùng một nhóm khách hàng.

Tương đồng Jaccard không chỉ là một khái niệm toán học, mà còn là một công cụ quan trọng trong việc hiểu và phân tích sự tương đồng giữa các tập hợp dữ liệu đa dạng.[2]

1.1.2 Shingling: Mô Hình Hóa Văn Bản và Đo Lường Sự Tương Đồng

Shingling là một phương pháp mô hình hóa văn bản trong lĩnh vực xử lý ngôn ngữ tự nhiên và đặc biệt hữu ích trong việc đo lường sự tương đồng giữa các văn bản. Phương pháp này tập trung vào việc chia nhỏ văn bản thành các đoạn văn bản con gọi là "shingles" để sau đó ánh xạ chúng thành các tập hợp. Dưới đây là một trình bày về Shingling và ứng dụng của nó:

- Định Nghĩa:

Shingling là quá trình chia một văn bản thành các phần nhỏ có kích thước cố định gọi là "shingles." Mỗi shingle là một chuỗi gồm k từ liên tiếp trong văn bản. Các shingles này sau đó được biểu diễn dưới dạng tập hợp, tạo thành biểu diễn tập hợp của văn bản.

- Ứng Dụng:

- Đo Lường Sự Tương Đồng Văn Bản Chi Tiết: Shingling cho phép đo lường sự tương đồng giữa các văn bản ở mức độ chi tiết hơn so với việc sử dụng từ vựng. Thay vì chỉ xem xét từng từ, shingling xem xét các đoạn nhỏ liên tục, giúp phát hiện sự tương đồng trong cấu trúc và diễn đạt của văn bản.

- Xử Lý Nội Dung Web và Tìm Kiếm Liên Quan: Trong việc xử lý nội dung web, shingling có thể được sử dụng để xác định sự tương đồng giữa các trang web. Điều này có ứng dụng trong việc tìm kiếm liên quan, phân loại nội dung, và theo dõi sự xuất hiện của thông tin trên mạng.

Shingling, tương tự như tương đồng Jaccard, là một công cụ mạnh mẽ trong việc mô hình hóa và đo lường sự tương đồng giữa các đoạn văn bản, mang lại thông tin chi tiết về cấu trúc và nội dung.[2]

1.2 Tầm quan trọng và ứng dụng của hệ thống đề xuất.

Hệ thống đề xuất không chỉ là một công nghệ mà còn là động lực mạnh mẽ đằng sau sự cá nhân hóa và tối ưu hóa trong cả thế giới kinh doanh và cuộc sống hàng ngày. Với sự đa dạng và linh hoạt, hệ thống đề xuất đã và đang ngày càng chứng minh vai trò quan trọng của mình trong nhiều lĩnh vực.

1.2.1 Ứng dụng trong thực tế.

- Thương mại điện tử: Hệ thống đề xuất trong thương mại điện tử không chỉ là công cụ quảng cáo mà còn là người hướng dẫn tận tâm của người tiêu dùng. Bằng cách theo dõi và hiểu biểu hiện mua sắm của người dùng, nó có thể tạo ra các đề xuất thông minh, thậm chí dự đoán nhu cầu tương lai. Điều này không chỉ tối ưu hóa trải nghiệm mua sắm mà còn giúp doanh nghiệp tạo ra chiến lược bán hàng hiệu quả.
- Dịch vụ âm nhạc/điện ảnh: Hệ thống đề xuất trong lĩnh vực giải trí không chỉ giúp người dùng khám phá nội dung mới mà còn tạo ra sự kết nối giữa các tác phẩm và người xem. Nó có thể dựa vào sở thích, lịch sử xem, thậm chí cảm xúc để đưa ra những đề xuất độc đáo, đem lại trải nghiệm giải trí không giới hạn.
- Cửa hàng ứng dụng di động: Trong thế giới ngày nay, hệ thống đề xuất không chỉ là công cụ giới thiệu ứng dụng mới mà còn là đối tác đáng tin cậy của nhà phát triển. Bằng cách giới thiệu ứng dụng dựa trên sở thích và nhu cầu cụ thể của người dùng, nó giúp tạo ra cộng đồng ứng dụng phong phú và đa dạng.
- Quảng cáo trực tuyến: Hệ thống đề xuất không chỉ là công cụ để tối ưu hóa hiệu suất quảng cáo mà còn là nguồn cảm hứng sáng tạo. Bằng cách hiểu rõ người xem và xu hướng ngành, nó giúp doanh nghiệp tạo ra những chiến dịch quảng cáo độc đáo, thu hút sự chú ý và tương tác tích cực.

1.2.2 Vai trò trong kinh doanh.

- Tăng doanh thu: Hệ thống đề xuất không chỉ giúp tăng doanh thu thông qua việc tối ưu hóa doanh số bán hàng mà còn làm gia tăng giá trị đơn hàng. Bằng cách đề xuất các sản phẩm hoặc dịch vụ bổ sung phù hợp, nó tạo ra cơ hội cho doanh nghiệp tăng cường doanh thu từ mỗi giao dịch.
- Thay thế công cụ tìm kiếm: Trong khi công cụ tìm kiếm yêu cầu người dùng có ý định cụ thể, hệ thống đề xuất mở rộng không gian khám phá. Điều này không chỉ

làm giảm áp lực tìm kiếm chủ động mà còn mở rộng khả năng tiếp cận đối tượng khách hàng mới, thúc đẩy sự đa dạng và phát triển.

- Phân tích xu hướng và hành vi người dùng: Khả năng thu thập và phân tích xu hướng và hành vi người dùng không chỉ giúp doanh nghiệp hiểu rõ người tiêu dùng mà còn là nguồn thông tin quý báu để dự đoán xu hướng thị trường. Điều này giúp doanh nghiệp không chỉ thích ứng nhanh chóng mà còn định hình sự phát triển dài hạn của họ.

1.3 Các loại phương pháp gợi ý phổ biến.

1.3.1 Gợi ý dựa trên nội dung (Content-Based Recommendation).

- Mô tả: Phương pháp này tập trung vào đặc điểm và nội dung của sản phẩm để đề xuất các mục tương tự. Thông qua việc phân tích thuộc tính của sản phẩm và sở thích của người dùng, hệ thống có thể tạo ra đề xuất cá nhân hóa.

Hình 3. Kỹ thuật lọc theo nội dung

- Ưu điểm:

- Hiệu suất tốt với những người dùng mới, khi chưa có đủ dữ liệu về hành vi của họ.
- Đề xuất dựa trên nội dung cụ thể, giúp tạo ra đề xuất chính xác và giải quyết vấn đề "cold start."

- Nhược điểm:

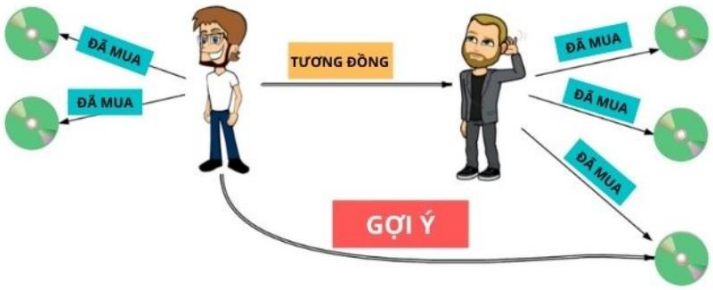
- Hạn chế trong việc khám phá sở thích mới của người dùng do sự hạn chế của thông tin nội dung.

- Không có khả năng đề xuất sản phẩm nếu chúng không tương tự với các sản phẩm đã được người dùng xem xét.

- Ví dụ ứng dụng thực tế: Netflix sử dụng gợi ý dựa trên nội dung để đề xuất phim và chương trình TV dựa trên thị hiếu của người xem, ví dụ như đề xuất phim hành động nếu người dùng thường xem thể loại này.[9]

1.3.2 Gợi ý dựa trên hành vi người dùng (Collaborative Filtering).

- Mô tả: Phương pháp này sử dụng thông tin về hành vi và sở thích của người dùng để đề xuất sản phẩm. Nó phát hiện mối quan hệ tương đồng giữa người dùng và đề xuất các mục mà người dùng có sở thích giống nhau đã thích.



Hình 4. Kỹ thuật lọc theo hành vi

- Ưu điểm:

- Có khả năng đề xuất sản phẩm mới mà người dùng có thể chưa biết đến dựa trên sự tương đồng với người dùng khác.

- Hiệu suất tốt với cộng đồng người dùng lớn và có sự chia sẻ sở thích.

- Nhược điểm:

- Gặp khó khăn khi đối mặt với vấn đề "cold start" với người dùng mới.

- Dựa vào dữ liệu người dùng, nếu thông tin không chính xác hoặc thay đổi thường xuyên, độ chính xác có thể giảm.

- Ví dụ ứng dụng thực tế: Amazon sử dụng collaborative filtering để gợi ý sản phẩm dựa trên lịch sử mua sắm và đánh giá của người dùng có sở thích tương tự. [10]

1.3.3 Phương pháp kết hợp (Hybrid Methods).

- Mô tả: Phương pháp này kết hợp các phương pháp truyền thống như gợi ý dựa trên nội dung và collaborative filtering để tận dụng ưu điểm của từng loại. Điều này có thể cải thiện độ chính xác và đa dạng của các đề xuất.
- Ưu điểm:

- Kết hợp ưu điểm của cả hai loại phương pháp, cung cấp đề xuất chính xác và đa dạng.
- Hạn chế nhược điểm của từng loại phương pháp, giúp tối ưu hóa trải nghiệm người dùng.

- Nhược điểm:

- Đòi hỏi nhiều công sức trong việc tích hợp và duy trì.
- Cần xử lý độ phức tạp của cả hai hệ thống.

- Ví dụ ứng dụng thực tế: Spotify sử dụng phương pháp kết hợp bằng cách tích hợp cả gợi ý dựa trên thị hiếu nhạc và dựa trên hành vi người dùng để tạo ra danh sách phát cá nhân hóa. [3]

1.4 Thách thức và cơ hội trong phát triển phương pháp gợi ý.

1.4.1 Thách thức về đa dạng và mất mát thông tin:

- Đa dạng trong gợi ý: Một trong những thách thức chính là đảm bảo độ đa dạng trong việc đề xuất sản phẩm. Hệ thống đề xuất cần phải hiểu rõ và đáp ứng đa dạng của sở thích và mong muốn của người dùng, tránh việc tạo ra nhóm hẹp và lặp lại.
- Mất mát thông tin: Khi xử lý lượng lớn dữ liệu, có khả năng mất mát thông tin quan trọng về sở thích và hành vi của người dùng. Điều này có thể dẫn đến đề xuất không chính xác và thiếu độ đa dạng.

1.4.2 Cơ hội với sự phát triển của học máy và học sâu:

- Học máy và học sâu: Sự tiến bộ trong lĩnh vực học máy và học sâu mở ra cơ hội lớn để cải thiện độ chính xác và khả năng đề xuất. Các mô hình phức tạp, như mạng nơ-ron sâu, có khả năng học các biểu diễn phức tạp của dữ liệu và cải thiện khả năng dự đoán.

- Tích hợp đa nguồn dữ liệu: Sự đa dạng của nguồn dữ liệu giúp cải thiện chất lượng đề xuất. Kết hợp thông tin từ nhiều nguồn như xã hội, đánh giá người dùng, và thông tin chi tiết về sản phẩm có thể tạo ra mô hình phong phú hơn.
- Xử lý vấn đề "Cold Start": Các phương pháp mới sử dụng học máy có thể giải quyết vấn đề "cold start" hiệu quả hơn. Điều này làm tăng khả năng đề xuất cho người dùng mới hoặc mục chưa được đánh giá.

1.4.3 Tương tác người dùng tăng cường:

Phản hồi người dùng liên tục:

- Học máy tăng cường cho cá nhân hóa: Triển khai học máy tăng cường để liên tục học từ phản hồi người dùng, xây dựng một mô hình người dùng ngày càng chính xác và linh hoạt, đáp ứng linh hoạt đối với sự thay đổi trong sở thích và ưu tiên của họ.
- Hệ thống hỏi đáp tự động: Mở rộng khả năng tương tác bằng cách tích hợp hệ thống hỏi đáp tự động, nâng cao khả năng hiểu biết về nhu cầu và yêu cầu của người dùng, và cung cấp đề xuất phản ánh chính xác.

1.4.4 Bảo mật và quyền riêng tư:

Quản lý an toàn và quyền riêng tư:

- Chính sách rõ ràng: Phát triển chính sách an toàn và quyền riêng tư rõ ràng, cung cấp thông tin chi tiết để giáo dục người dùng về cách thông tin của họ được sử dụng và bảo vệ.
- Công cụ quản lý cá nhân: Cung cấp công cụ quản lý để người dùng có thể kiểm soát và điều chỉnh thông tin cá nhân của mình, tăng sức mạnh và niềm tin từ phía họ.

1.4.5 Học máy gia Reinforcement (Học máy tăng cường):

Tối ưu hóa hiệu suất:

- Sử dụng học máy học nhanh: Áp dụng kỹ thuật học máy học nhanh để nhanh chóng tối ưu hóa hiệu suất mô hình dựa trên phản hồi người dùng, giảm thời gian cập nhật và tăng cường trải nghiệm người dùng.
- Điều chỉnh mô hình trực tiếp: Sử dụng học máy tăng cường để điều chỉnh mô hình ngay lập tức sau mỗi tương tác, tối ưu hóa đề xuất theo ngữ cảnh và nhu cầu người dùng hiện tại.

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1 Định nghĩa và nguyên tắc hoạt động cơ bản về lọc theo nội dung.

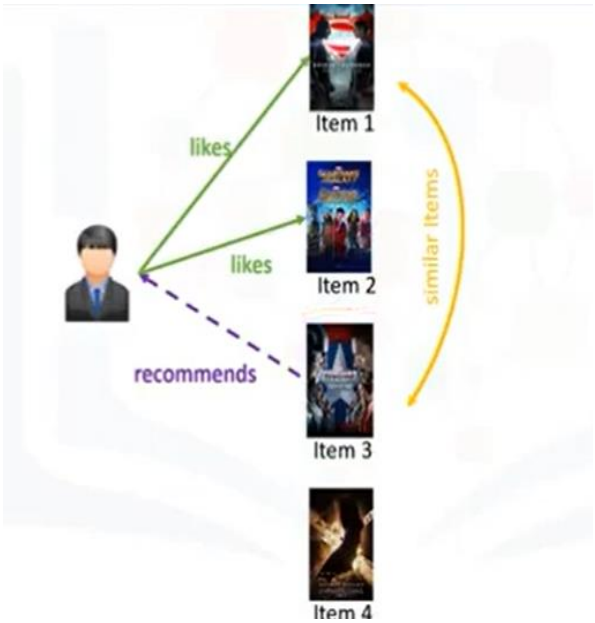
2.1.1 Định nghĩa.

Hệ thống đề xuất dựa trên nội dung cố gắng đề xuất các mục cho người dùng dựa trên hồ sơ của họ. Hồ sơ người dùng xoay quanh sở thích và thị hiếu của người dùng đó nó được định hình dựa trên xếp hạng của người dùng bao gồm số lần người dùng nhấp vào các nhau hoặc thậm chí có thể thích các mục đó. Quá trình đề xuất dựa trên sự giống nhau giữa các mục đó. Độ tương tự hay mức độ gần gũi của các mục được đo lường dựa trên sự tương đồng về nội dung của các mục đó. Khi nói về nội dung thì chúng ta đang nói về những thứ như danh mục, thẻ, thể loại,...[9]

Phương pháp lọc theo nội dung là một kỹ thuật trong hệ thống đề xuất, mà ở đó thông tin được gợi ý dựa trên tính chất và đặc điểm của sản phẩm hoặc nội dung, chứ không phải dựa trên hành vi của người dùng hoặc sự tương đồng với người dùng khác. Điều này có nghĩa là hệ thống tập trung vào những đặc điểm cụ thể của mỗi mục và sử dụng thông tin này để đề xuất các mục tương tự.[9]

2.1.2 Nguyên tắc hoạt động.

- Phân tích nội dung: Hệ thống sử dụng các phương pháp phân tích để đánh giá và hiểu các đặc điểm của sản phẩm. Điều này có thể bao gồm việc xác định từ khóa, thể loại, hoặc các thuộc tính khác của mục, tùy thuộc vào bối cảnh cụ thể.
- Xây dựng hồ sơ người dùng: Dựa trên lịch sử tương tác của người dùng và thông tin cá nhân, hệ thống xây dựng một hồ sơ người dùng chi tiết. Thông qua quá trình này, nó có thể hiểu rõ sở thích, ưa thích, và nhu cầu của người dùng, tạo ra một cơ sở để đề xuất các mục tương tự.[9]



Hình 5. Ví dụ về hệ thống đề xuất

- Ví dụ: Nếu chúng ta có bốn bộ phim nếu người dùng thích 2 bộ phim đầu tiên và nếu bộ phim thứ 3 giống thể loại với bộ phim thứ 1 thì hệ thống sẻ đưa ra đề xuất bộ phim thứ 3 cho người dùng.

2.1.3 Ưu điểm và Ứng dụng:

- Tích hợp hiệu quả với người dùng mới: Phương pháp lọc theo nội dung hoạt động tốt khi chưa có đủ dữ liệu về hành vi của người dùng. Với người dùng mới, nó có khả năng đưa ra các đề xuất dựa trên đặc điểm của sản phẩm, giảm thiểu vấn đề thiếu thông tin.
- Đối mặt tốt với vấn đề "cold start": Hệ thống có khả năng khắc phục khó khăn khi không có đủ thông tin về người dùng mới hoặc sản phẩm mới. Thay vì dựa vào lịch sử tương tác, nó tập trung vào các thuộc tính cụ thể của mục để đưa ra đề xuất.
- Ứng dụng linh hoạt: Phương pháp này thích hợp trong nhiều lĩnh vực ứng dụng như thương mại điện tử, giáo dục trực tuyến, và các nền tảng cung cấp nội dung trực tuyến, nơi sự linh hoạt và cá nhân hóa là quan trọng.[9]

2.2 Lịch sử và phát triển của phương pháp:

2.2.1 Xuất hiện ban đầu:

Đầu những năm 1990: Phương pháp lọc theo nội dung xuất hiện và đầu tiên được ứng dụng trong các hệ thống đề xuất đầu tiên, như các thư viện số. Những năm đầu tiên này là giai đoạn pionnier, nơi mà các nhà nghiên cứu và kỹ sư đầu tiên bắt đầu nhận ra tiềm năng của việc tập trung vào đặc điểm và nội dung của sản phẩm.

2.2.2 Phát triển qua các giai đoạn:

- Thập kỷ 2000: Sự gia tăng đáng kể về khả năng xử lý dữ liệu và sự phát triển của ngôn ngữ tự nhiên đã đưa đến cải tiến đáng kể trong phương pháp lọc theo nội dung. Khả năng tính toán mạnh mẽ hơn cùng với sự hiểu biết ngôn ngữ tự nhiên đã mở ra cánh cửa cho việc tối ưu hóa các thuật toán lọc theo nội dung.
- Hiện đại: Sự phát triển của học sâu và các mô hình ngôn ngữ tự nhiên tiếp tục nâng cao độ chính xác và hiệu suất của phương pháp này. Sự tiến triển nhanh chóng trong lĩnh vực trí tuệ nhân tạo đã mở ra những cánh cửa mới cho ứng dụng và tối ưu hóa phương pháp lọc theo nội dung.

2.2.3 Ứng dụng và sự lan rộng:

- Thương mại điện tử: Các trang web thương mại điện tử như Amazon và eBay chủ yếu sử dụng phương pháp lọc theo nội dung để đề xuất sản phẩm dựa trên mô tả và đặc điểm kỹ thuật. Nhờ vào việc tập trung vào nội dung, họ có thể đưa ra các đề xuất chính xác, đáp ứng mong muốn và nhu cầu cụ thể của người tiêu dùng.
- Giáo dục trực tuyến: Nền tảng học trực tuyến sử dụng phương pháp lọc theo nội dung để đề xuất nội dung học dựa trên nguyện vọng và kiến thức trước đó của người học. Điều này giúp cá nhân hóa trải nghiệm học tập và nâng cao khả năng tiếp thu thông tin của sinh viên.

2.2.4 Thách thức và Hướng Phát Triển:

- Thách thức: Hạn chế trong việc khám phá sở thích mới của người dùng là một thách thức đối với phương pháp lọc theo nội dung. Do sự tập trung vào thông tin cụ thể, có thể khó để đề xuất những sản phẩm có liên quan mà người dùng có thể chưa biết đến.

- Hướng phát triển: Hướng phát triển chính của phương pháp lọc theo nội dung là sự kết hợp với các phương pháp khác, đặc biệt là học sâu. Bằng cách này, có thể tối ưu hóa độ chính xác và đa dạng của đề xuất, giúp giải quyết vấn đề của "bubble effect" và mở rộng khả năng hiệu quả của hệ thống đề xuất.

2.3 Nguyên lý hoạt động của phương pháp lọc theo nội dung.

2.3.1 Nguyên tắc cơ bản của phương pháp lọc theo nội dung.

2.3.1.1 Hiểu biết về sản phẩm.

Phương pháp lọc theo nội dung không chỉ giới hạn ở việc xác định đặc điểm cơ bản của sản phẩm, bao gồm phân tích các đặc điểm như từ khóa, thể loại, hoặc thuộc tính kỹ thuật. Mà còn mở rộng việc hiểu biết đến các yếu tố không gian như ngữ cảnh và ý nghĩa sâu sắc của sản phẩm trong hệ thống. Việc này giúp tăng cường khả năng hiểu biết về tính chất đặc trưng và giúp mô hình đề xuất có sự chi tiết hóa cao.[9]

2.3.1.2 Xây dựng hồ sơ người dùng.

Xây dựng hồ sơ người dùng không chỉ đơn thuần là việc thu thập thông tin cá nhân và lịch sử mua sắm. Nó còn liên quan đến việc hiểu biết về tâm lý và mong đợi của người dùng, tạo ra một hồ sơ đa chiều để phản ánh sự đa dạng và thay đổi trong ưu tiên của họ.[9]

2.3.1.3 So khớp nội dung và hồ sơ người dùng.

Thuật toán so khớp không chỉ dựa vào các yếu tố cơ bản như từ khóa hay thể loại, mà còn tích hợp các phương tiện như xử lý ngôn ngữ tự nhiên để hiểu biết ý nghĩa sâu sắc của sản phẩm và sở thích người dùng. Việc này giúp tạo ra những đề xuất có sự kết hợp tinh tế giữa nội dung và người dùng.[9]

2.3.1.4 Tạo ra đề xuất cá nhân hóa.

Quá trình tạo đề xuất không chỉ dựa vào sự so khớp mà còn xem xét mức độ đa dạng của nội dung để đảm bảo rằng người dùng nhận được sự đề xuất không chỉ phản ánh sở thích hiện tại mà còn mở rộng tới các lĩnh vực có thể người dùng chưa khám phá.[9]

2.3.1.5 Bài toán ví dụ về nguyên tắc hoạt động của hệ thống gợi ý theo nội dung.

Giả sử chúng ta có bộ dữ liệu có 6 bộ phim và thể loại như sau:

Bảng 1. Bộ dữ liệu phim ví dụ

Tên phim	Thể loại
Batman v Superman	Advanture, super hero
Guardians of the Galaxy	Comedy, sci-fi, advanture, super hero
Captain America: Civil War	Comedy, super hero
Hitchhiker's guide to the galaxy	Advanture, comedy, sci-fi
Batman begins	Super hero
Spiderman	Comedy, super hero

Giả sử người dùng đã xem và đánh giá ba bộ phim đầu tiên như sau: Batman v Superman 2/10, Guardians of the Galaxy 10/10, Captain America: Civil War 8/10. Nhiệm vụ của công cụ đề xuất và giới thiệu một trong ba bộ phim ứng cử viên còn lại cho người dùng này, hay nói cách khác là dự đoán xếp hạng cho ba bộ phim còn lại nếu người dùng xem chúng.

Để đạt được điều này, chúng ta phải xây dựng hồ sơ người dùng. Đầu tiên chúng ta cần tạo một vecto để hiển thị xếp hạn của người dùng đối với những bộ phim đã xem gọi là xếp hạng người dùng đầu vào.

Bảng 2. Xếp hạng người dùng

Tên phim	Người dùng đánh giá
Batman v Superman	2
Guardians of the Galaxy	10
Captain America: Civil War	8

Sau đó mã hóa các thuộc tính phim qua phương pháp mã hóa một lần ta được:

Bảng 3. Ma trận thể loại phim người dùng đã xem

	Comedy	Adventure	Super hero	Sci-Fi
Batman v Superman	0	1	1	0
Guardians of the Galaxy	1	1	1	1
Captain America: Civil War	1	0	1	0

Nhân hai ma trận trên với nhau ta được:

Bảng 4. Ma trận trọng số thể loại

	Comedy	Adventure	Super hero	Sci-Fi
Batman v Superman	0	2	2	0
Guardians of the Galaxy	10	10	10	10
Captain America: Civil War	8	0	8	0

Ma trận này được gọi là ma trận trọng số thể loại và nó thể hiện sở thích của người dùng đối với từng thể loại dựa trên những bộ phim mà người dùng đã xem. Về cơ bản điểm số của các thể loại là: comedy 18, adventure 12, super hero 20, sci-fi 10. Sau khi chuẩn hóa chúng ta sẻ có được hồ sơ người dùng như sau:

Bảng 5. Hồ sơ người dùng

	Comedy	Adventure	Super hero	Sci-Fi
Chỉ số người dùng	0.3	0.2	0.33	0.16

Dựa trên hồ sơ người dùng ta thấy rõ ràng ngừ này thích phim thể loại Super hero hơn các thể loại khác. Chúng ta sẻ sử dụng hồ sơ này để tìm ra bộ phim phù hợp để giới thiệu cho người dùng này.

Bây giờ chúng ta sẻ tính xem hệ thống sẻ đề xuất phim nào trong ba bộ phim mà người dùng này chưa xem.

Bảng 6. Ma trận thể loại phim người dùng chưa xem

	Comedy	Adventure	Super hero	Sci-Fi
Hitchhiker's guide to the galaxy	1	1	0	1
Batman begins	0	0	1	0
Spiderman	1	0	1	0

Bây giờ chúng ta sẻ tìm ra bộ phim nào phù hợp nhất để đề xuất cho người dùng. Để làm được chúng ta sẻ nhân ma trận hồ sơ người dùng với ma trận phim đề xuất.

Bảng 7. Trọng số ma trận đề xuất

	Comedy	Adventure	Super hero	Sci-Fi
Hitchhiker's guide to the galaxy	0.3	0.2	0	0.16
Batman begins	0	0	0.33	0
Spiderman	0.33	0	0.33	0

Ma trận trên cho thấy mức độ quan trọng của từng thể loại đối với hồ sơ người dùng. Bây giờ chúng ta sẻ tổng hợp các xếp hạng trọng số và chúng ta sẻ biết được mức dộ quan tâm có thể có của người dùng đối với ba bộ phim này là:

Bảng 8. Chỉ số đề xuất

Tên phim	Trọng số
Hitchhiker's guide to the galaxy	0.66
Batman begins	0.33
Spiderman	0.63

Nhìn vào bảng chỉ số đề xuất của ba bộ phim trên chúng ta thấy có số điểm cao nhất trong danh sách là bộ phim Hitchhiker's guide to the galaxy nên chúng ta nói việc giới thiệu cho người dùng bộ phim này là điều đúng đắn.[9]

2.3.2 Cơ sở toán học và thuật toán.

2.3.2.1 Vector hóa sản phẩm và người dùng.

Trong việc vector hóa, mô hình không chỉ xem xét các đặc điểm cơ bản mà còn quan tâm đến việc biểu diễn không gian ngữ cảnh, giúp mô hình nắm bắt được sự tương tác phức tạp giữa các yếu tố.

2.3.2.2 Sử dụng các thuật toán.

- Cosine Similarity: Ngoài việc đo lường tương đồng giữa vectơ, còn có thể tích hợp trọng số để tăng cường độ chính xác của so sánh.
- TF-IDF (Term Frequency-Inverse Document Frequency): Đối với mỗi sản phẩm, mô hình có thể tự động xác định trọng số cho các thuộc tính dựa trên tần suất xuất hiện trong hồ sơ người dùng.

2.3.2.3 Mô hình học sâu.

Học sâu không chỉ liên quan đến việc học biểu diễn sâu sắc mà còn có khả năng chủ động hóa quá trình học thông qua tương tác với người dùng, tăng cường khả năng dự đoán chính xác theo thời gian.

2.4 Các hệ thống và ứng dụng thực tế.

2.4.1 Áp dụng trong thương mại điện tử:

- Mục tiêu: Cải thiện trải nghiệm mua sắm trực tuyến.
- Ví dụ ứng dụng 1: Amazon
- Amazon nhằm đến việc tối ưu hóa trải nghiệm mua sắm bằng cách sử dụng phương pháp lọc theo nội dung. Họ đề xuất sản phẩm dựa trên sở thích và lịch sử mua sắm của người dùng, nhằm mang đến trải nghiệm cá nhân hóa. Thách thức chủ yếu của họ là xử lý đa dạng sản phẩm và đảm bảo đánh giá chính xác để đáp ứng mong muốn của người dùng.

2.4.2 Nền tảng bán lẻ trực tuyến:

- Mục tiêu: Hỗ trợ người dùng khám phá sản phẩm mới và thú vị.
- Ví dụ ứng dụng 1: Shopee

- Shopee chủ yếu sử dụng lọc theo nội dung để giúp người dùng khám phá sản phẩm mới. Giao diện đa dạng và linh hoạt giúp họ đáp ứng nhu cầu người dùng đa dạng. Thách thức chính của Shopee là duy trì sự đa dạng của sản phẩm trên nền tảng của mình.
- Ví dụ ứng dụng 2: Alibaba
- Alibaba gợi ý sản phẩm và nhà cung cấp dựa trên lịch sử tìm kiếm và mua sắm của người dùng. Họ chú trọng vào việc tăng cường bảo mật và quyền riêng tư để xây dựng lòng tin từ phía người dùng.

2.4.3 Tích hợp trong streaming âm nhạc:

- Mục tiêu: Tạo danh sách phát cá nhân hóa.
- Ví dụ ứng dụng: Spotify
- Spotify sử dụng đề xuất dựa trên nội dung để cá nhân hóa danh sách phát cho người nghe. Họ không chỉ gợi ý bài hát dựa trên sở thích hiện tại mà còn liên tục cải thiện thông qua phản hồi liên tục từ người dùng.

2.4.5 Ứng dụng di động và cửa hàng ứng dụng:

- Mục tiêu: Tăng khả năng khám phá sản phẩm.
- Ví dụ ứng dụng 1: Apple App Store
- Apple App Store đề xuất ứng dụng dựa trên sở thích và ứng dụng trước đó của người dùng. Họ cung cấp công cụ quản lý và điều khiển thông tin cá nhân để người dùng có sự linh hoạt và an toàn.
- Ví dụ ứng dụng 2: Google Play Store
- Google Play Store gợi ý ứng dụng dựa trên tương tác và lịch sử tải về của người dùng. Họ tích hợp học máy tăng cường để cập nhật mô hình dựa trên thay đổi trong sở thích và hành vi.

2.4.6 Giáo dục trực tuyến:

- Mục tiêu: Đề xuất khóa học dựa trên lịch sử học tập và quan tâm của học viên.
- Ví dụ ứng dụng: Khan Academy
- Khan Academy gợi ý video và bài giảng dựa trên kết quả kiểm tra và tiến độ học tập. Họ đặc biệt chú trọng vào bảo mật thông tin học tập và tôn trọng quyền riêng tư học viên.

2.4.7 Du lịch và giải trí:

- Mục Tiêu: Gợi ý điểm đến và hoạt động dựa trên sở thích và kinh nghiệm trước đó của người dùng.
- Ví dụ ứng dụng: TripAdvisor
- TripAdvisor đề xuất điểm đến và hoạt động dựa trên sở thích và kinh nghiệm trước đó của người dùng. Họ tích hợp tương tác người dùng liên tục để cải thiện đề xuất.

2.5 Một số thuật toán tiêu biểu trong phương pháp lọc theo nội dung.

2.5.1 Thuật toán TF-IDF (Term Frequency-Inverse Document Frequency):

2.5.1.1 Mô tả thuật toán TF-IDF:

TF-IDF là một kỹ thuật trong xử lý ngôn ngữ tự nhiên (NLP) sử dụng để đo lường tầm quan trọng của một từ trong một tài liệu so với một tập hợp các tài liệu khác. Nó giúp xác định độ quan trọng của từng từ trong một văn bản, đặt nặng vào những từ xuất hiện nhiều trong một tài liệu cụ thể nhưng ít xuất hiện trong các tài liệu khác.

2.5.1.2 Nguyên lý hoạt động thuật toán TF-IDF:

- Tính Term Frequency (TF): Đo lường tần suất xuất hiện của từ trong một tài liệu. Điều này thường được tính theo công thức:

$$TF(t,d)=\frac{\text{số từ } \boldsymbol{t} \text{ xuất hiện trong tài liệu } \boldsymbol{d}}{\text{Tổng số từ trong } \boldsymbol{d}} \text{ [3]}$$

- Tính Inverse Document Frequency (IDF): Điều chỉnh tần suất của từ dựa trên tần suất xuất hiện của từ đó trong toàn bộ tập hợp các tài liệu. Công thức tính IDF là:

$$IDF(t)=\log=\frac{\text{tổng số tài liệu trong tập hợp}}{\text{số tài liệu chứa từ } \boldsymbol{t}} \text{ [3]}$$

- Tính TF-IDF Score: Kết hợp TF và IDF để đánh giá tầm quan trọng của từ trong tài liệu:

$$TF\text{-}IDF(t,d)=TF(t,d)\times IDF(t) \text{ [3]}$$

2.5.1.3 Mục tiêu thuật toán TF-IDF:

TF-IDF (Term Frequency-Inverse Document Frequency) có mục tiêu chính là xác định độ quan trọng của từng từ trong một tài liệu để làm nổi bật những từ có tầm quan trọng đặc biệt đối với nội dung của tài liệu đó. Điều này được thực hiện thông qua việc

tính toán tần suất xuất hiện của từ trong tài liệu (Term Frequency - TF) và điều chỉnh nó dựa trên tần suất của từ đó trong toàn bộ tập hợp các tài liệu (Inverse Document Frequency - IDF).

2.5.1.4 Ưu và nhược điểm thuật toán TF-IDF:

- Ưu điểm của TF-IDF bao gồm tính đơn giản và dễ triển khai, hiệu quả trong việc đo lường tầm quan trọng của từ, và khả năng hoạt động tốt trong nhiều ứng dụng xử lý ngôn ngữ tự nhiên (NLP).
- Tuy nhiên, cũng có nhược điểm, bao gồm việc không xem xét ngữ cảnh và cấu trúc ngôn ngữ, khó xử lý các từ đồng nghĩa, và hiệu suất giảm khi xử lý các văn bản ngắn.

2.5.2 Thuật toán Word Embeddings:

2.5.2.1 Mô tả thuật toán Word Embeddings:

Word Embeddings là một phương pháp biểu diễn từ dưới dạng vectơ số học trong không gian nhiều chiều. Thay vì đơn thuần là đại diện cho từng từ bằng một chỉ số, Word Embeddings tạo ra một không gian vectơ trong đó các từ có ý nghĩa tương đồng được biểu diễn gần nhau trong không gian vectơ.

2.5.2.2 Nguyên lý hoạt động thuật toán Word Embeddings:

- Word Embeddings, cụ thể là mô hình Word2Vec, hoạt động bằng cách ánh xạ từng từ trong từ điển thành các vectơ số học trong không gian nhiều chiều. Nguyên lý này dựa trên giả định rằng các từ có ý nghĩa tương đồng sẽ có biểu diễn số học gần nhau.
- Công Thức (Skip-gram):

Mô hình Skip-gram của Word2Vec sử dụng hàm softmax để dự đoán xác suất xuất hiện của các từ xung quanh (context) dựa trên từ đang xem xét. Công thức như sau:

$$P(context|word)=\frac{e^{v}context\cdot^{v}word}{\sum w\in Vocab^{e^{v}}\ w\cdot^{v}word} \quad [3]$$

Trong đó:

- Vcontext là vectơ biểu diễn của từ trong ngữ cảnh.
- Vword là vectơ biểu diễn của từ đang xem xét.

- Vocab là tập hợp tất cả các từ trong từ điển.

Công thức trên giúp mô hình cập nhật các vectơ biểu diễn để phản ánh tương đồng ngữ nghĩa giữa các từ trong không gian vectơ.

2.5.2.3 Mục tiêu thuật toán Word Embeddings:

Mục tiêu chính của Word Embeddings là tạo ra biểu diễn không gian của từ với sự tương đồng ngữ nghĩa được bảo toàn. Vectơ biểu diễn từng từ phản ánh mối quan hệ ngữ nghĩa và cú pháp giữa chúng.

2.5.2.4 Ưu và nhược điểm thuật toán Word Embeddings:

- Ưu Điểm:

- Word Embeddings có khả năng hiểu biết ngữ nghĩa của từ trong ngữ cảnh, giúp giảm thiểu hiện tượng "đồng nghĩa" và "đối nghĩa".
- Giải Quyết Vấn Đề Từ Đồng Nghĩa: Có thể giải quyết vấn đề từ đồng nghĩa bằng cách ánh xạ các từ tương đồng gần nhau trong không gian vectơ.

- Nhược Điểm:

- Đòi hỏi lượng dữ liệu lớn để đào tạo mô hình Word Embeddings đủ chất lượng.
- Khó Khăn Khi Xử Lý Các Từ Hiếm: Có khó khăn khi xử lý các từ hiếm, đặc biệt là khi không có đủ dữ liệu để biểu diễn chúng trong không gian vectơ.

2.6 Thuật toán TF-IDF và bài toán ví dụ.

2.6.1 TF-IDF trong xử lý ngôn ngữ tự nhiên và thông tin truy xuất:

- TF-IDF giúp xác định độ quan trọng của mỗi từ trong một tập văn bản dựa trên sự kết hợp giữa tần suất xuất hiện của từ đó trong văn bản cụ thể và tầm quan trọng của từng từ trong toàn bộ tập văn bản.

- Tính tần suất từ (TF): TF đo lường sự xuất hiện của mỗi từ trong một văn bản, giúp hiểu rõ ngữ cảnh và chủ đề chính của văn bản đó.

$$TF(t,d)=\frac{số\ từ\ \boldsymbol{t}\ xuất\ hiện\ trong\ tài\ liệu\ \boldsymbol{d}}{Tổng\ số\ từ\ trong\ \boldsymbol{d}}$$ [3]

Những từ xuất hiện nhiều lần có thể là các từ quan trọng đối với nội dung cụ thể đó.

- Nghịch đảo tần suất văn bản (IDF): IDF xác định độ độc đáo của từng từ trong tập văn bản.

$$IDF(t)=\log=\frac{tổng\ số\ tài\ liệu\ trong\ tập\ hợp}{số\ tài\ liệu\ chứa\ từ\ t}$$ [3]

Từ xuất hiện ít trong nhiều văn bản sẽ có giá trị IDF cao, vì chúng có khả năng đặc biệt và quan trọng cho văn bản cụ thể.

- Kết hợp tất cả (TF-IDF Score): Phản ánh tầm quan trọng toàn cảnh. TF-IDF Score kết hợp cả hai yếu tố trên để tạo ra một con số phản ánh tầm quan trọng của từ trong cả ngữ cảnh của văn bản cụ thể và trong toàn bộ tập văn bản.

$$TF\text{-}IDF(t,d)=TF(t,d)\times IDF(t)$$ [3]

Từ với điểm số cao thường là những từ quan trọng và có ý nghĩa đặc biệt trong ngữ cảnh đó.

- Ứng Dụng Thực Tế:

- Hệ Thống Tìm kiếm: Trong các công cụ tìm kiếm, TF-IDF giúp cải thiện hiệu suất tìm kiếm bằng cách xác định độ liên quan của mỗi văn bản đến một truy vấn.
- Phân Loại Văn bản: Tính hiệu quả trong Phân loại. Trong các mô hình máy học, TF-IDF thường được sử dụng để tạo ra vectơ đặc trưng, giúp phân loại văn bản vào các danh mục khác nhau.
- Tư vấn Thông tin: Xác định từ Khóa Quan trọng. TF-IDF là một công cụ hữu ích trong tư vấn thông tin, giúp xác định những từ khóa quan trọng trong một tập văn bản, làm nổi bật nội dung quan trọng.[8]

2.6.2 Bài toán ví dụ về thuật toán TF-IDF.

Chúng ta sẻ có 3 câu như như sau:

Câu 1: Good boy

Câu 2: Good gird

Câu 3: Boy gird good

Đầu tiên chúng ta sẻ đi tính tần số xuất hiện của thuật ngữ trong câu bằng công thức:

$$TF(t,d)=\frac{số\ từ\ t\ xuất\ hiện\ trong\ tài\ liệu\ d}{Tổng\ số\ từ\ trong\ d}$$ [3]

Ta sẻ được bảng giá trị như sau:

Bảng 9. Bảng tần xuất xuất hiện các từ trong câu (TF)

	Câu 1	Câu 2	Câu 3
Good	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{3}$
Boy	$\frac{1}{2}$	0	$\frac{1}{3}$
Gird	0	$\frac{1}{2}$	$\frac{1}{3}$

Bảng trên thể hiện rõ số lần xuất hiện của các từ trong câu. Chúng ta sẻ tiếp tục tính tần xuất nghịch đảo của tài liệu bằng công thức sau:

$$\text{IDF(t)=log=}\frac{\textit{tổng số tài liệu trong tập hợp}}{\textit{số tài liệu chứa từ } \boldsymbol{t}} \quad [3]$$

Ta sẻ được bảng giá trị:

Bảng 10. Bảng tần xuất nghịch đảo của tài liệu (IDF)

Từ	**IDF**
Good	$log\left(\frac{3}{3}\right) = 0$
Boy	$log(\frac{3}{2})$
Gird	$log(\frac{3}{2})$

Bảng trên thể hiện giá trị của tần xuất nghịch đảo của tài liệu. Sau đó chúng ta tiếng hành tính tầm quan trọng của từ trong ngữ cảnh bằng công thức sau:

$$\text{TF-IDF(t,d)=TF(t,d)}\times\text{IDF(t)} \quad [3]$$

Ta được bảng giá trị sau:

Bảng 11. Tầm quan trọng của từ trong ngữ cảnh

	Good	Boy	Gird
Câu 1	0	$\frac{1}{2} * log\left(\frac{3}{2}\right)$	0
Câu 2	0	0	$\frac{1}{2} * log\left(\frac{3}{2}\right)$
Câu 3	0	$\frac{1}{3} * log\left(\frac{3}{2}\right)$	$\frac{1}{3} * log\left(\frac{3}{2}\right)$

Như vậy chúng ta có thể thấy độ quan trọng của từ trong câu 1 nằm ở từ boy, câu 2 là gird và câu 3 là từ boy và gird.

2.7 Ứng dụng và ngôn ngữ thực hiện đề tài.

2.7.1 Anaconda Navigator.

- Anaconda Navigator là một ứng dụng quản lý môi trường và công cụ Python mạnh mẽ, được thiết kế để đơn giản hóa quá trình cài đặt, quản lý và tương tác với các môi trường Python và các công cụ khoa học dữ liệu. Đây là một thành phần quan trọng của bộ công cụ Anaconda, một nền tảng phổ biến trong cộng đồng khoa học dữ liệu và machine learning.
- Với giao diện đồ họa thân thiện, Anaconda Navigator cung cấp một trải nghiệm người dùng thuận lợi cho việc:

- Quản lý Môi trường: Tạo và chuyển đổi giữa các môi trường Python một cách dễ dàng, mỗi môi trường có thể chứa phiên bản Python và bộ thư viện riêng biệt.
- Quản lý Công cụ: Cài đặt, cập nhật và xóa các công cụ và gói thư viện Python, như Jupyter, Spyder, Pandas, NumPy, và nhiều công cụ khoa học dữ liệu khác.
- Quản lý Packages: Cài đặt và quản lý các gói Python thông qua giao diện người dùng trực quan.
- Khởi chạy Ứng dụng: Mở và chạy các ứng dụng như Jupyter Notebook, Spyder IDE và nhiều công cụ khoa học dữ liệu khác một cách thuận tiện.

- Với Anaconda Navigator, người dùng có thể tiếp cận môi trường Python và các công cụ khoa học dữ liệu một cách dễ dàng, giúp tối ưu hóa quá trình phát triển và nghiên cứu trong lĩnh vực khoa học dữ liệu và machine learning.

2.7.2 Jupyter Notebook.

Jupyter Notebook là một công cụ mạnh mẽ được sử dụng chủ yếu với ngôn ngữ lập trình Python, mang đến một môi trường tương tác và hiệu quả cho việc phân tích dữ liệu, thực hiện tính toán khoa học, và chia sẻ kiến thức. Bằng cách tích hợp mã nguồn Python và văn bản định dạng Markdown trong cùng một tệp, Jupyter Notebook cho phép người dùng tạo ra các tài liệu linh hoạt, trực quan, và dễ hiểu. Với khả năng chạy mã tương tác, hiển thị đồ họa, và hỗ trợ nhiều ngôn ngữ lập trình, Jupyter Notebook đã trở thành công cụ ưu tiên cho các nhà khoa học dữ liệu và lập trình viên Python trong quá trình phát triển và trình bày công việc của mình.

2.7.3 Ngôn ngữ lập trình Python.

- Python, một ngôn ngữ lập trình đa năng và mạnh mẽ, nổi tiếng với sự đơn giản, dễ đọc, và hiệu quả trong việc giải quyết nhiều vấn đề khác nhau. Khởi nguồn từ ý tưởng của Guido van Rossum và ra đời vào năm 1991, Python nhanh chóng trở thành lựa chọn hàng đầu của cộng đồng lập trình viên, từ người mới bắt đầu đến các chuyên gia trong nhiều lĩnh vực khác nhau.
- Một điểm đặc biệt của Python là cú pháp rõ ràng, giúp tạo ra mã nguồn dễ đọc và dễ hiểu. Sự đơn giản này không chỉ giúp tạo ra mã nhanh chóng mà còn giảm thiểu khả năng phạm lỗi và tăng hiệu suất lập trình.
- Điều quan trọng khác là cộng đồng lớn mạnh và tích cực. Python có một hệ sinh thái đa dạng với hàng ngàn thư viện và framework, giúp lập trình viên giải quyết nhanh chóng nhiều thách thức khác nhau. Sự linh hoạt và tích hợp của Python cũng làm cho nó trở thành ngôn ngữ lập trình linh hoạt, có thể tích hợp dễ dàng với các công nghệ và ngôn ngữ khác.

2.7.4 Flask framework.

Flask là một khung web siêu nhẹ và linh động dành cho ngôn ngữ lập trình Python, được thiết kế để xây dựng ứng dụng web nhanh chóng và hiệu quả. Với sự đơn giản và tính linh hoạt, Flask đã trở thành một trong những công cụ phổ biến cho

các nhà phát triển web. Dưới đây là một giới thiệu về Flask dựa trên những ứng dụng thực tế:

- Phát triển Ứng dụng Web Nhỏ:

• Flask thích hợp cho việc xây dựng các ứng dụng web nhỏ đến trung bình với codebase đơn giản.

• Đối với các dự án nhỏ, Flask giúp nhà phát triển triển khai nhanh và tập trung vào logic kinh doanh chính mà không phải lo lắng về quá trình cấu hình phức tạp.

- Ứng dụng Machine Learning và Data Science:

• Flask là một lựa chọn phổ biến để triển khai các mô hình học máy và ứng dụng data science.

• Kết hợp với các thư viện như scikit-learn, Flask giúp tạo ra các ứng dụng dự đoán và phân tích dữ liệu một cách linh hoạt.

Flask không chỉ giúp nhà phát triển tạo ra các ứng dụng web hiệu quả và dễ bảo trì mà còn thúc đẩy quá trình phát triển thông qua cộng đồng lớn và tài liệu phong phú. Đối với nhiều ứng dụng thực tế, Flask là một lựa chọn hợp lý với sự linh hoạt và độ dễ sử dụng của nó.

CHƯƠNG 3: ĐÁNH GIÁ KẾT QUẢ

3.1 Mô tả cấu trúc dữ liệu minh họa của đề tài

Trong phần cơ sở dữ liệu của đề tài dữ liệu này được thu thập từ [7] và tập dữ liệu cung cấp thông tin về các bộ phim, bao gồm tên đạo diễn, tên diễn viên chính, diễn viên phụ, thể loại và tiêu đề của từng tác phẩm. thông qua các thuộc tính này, dữ liệu giúp mô tả đa dạng và sự phong phú của ngành công nghiệp điện ảnh, với sự tham gia của nhiều đạo diễn và diễn viên trong các thể loại khác nhau như hành động, phiêu lưu, hài, gia đình, giả tưởng và nghệ thuật.

3.2 Xử lý dữ liệu

Cơ sở dữ liệu minh họa của chúng ta tập trung chủ yếu vào những thông tin quan trọng liên quan đến mỗi bộ phim. Dưới đây là một mô tả ngắn về các thuộc tính trong cơ sở dữ liệu:

- Tên đạo diễn (director_name): Chứa tên của đạo diễn của bộ phim.

- Tên diễn viên chính (actor_1_name): Chứa tên của diễn viên chính đóng vai trong bộ phim.

- Tên diễn viên phụ (actor_2_name, actor_3_name): Chứa tên của các diễn viên phụ tham gia trong bộ phim.

- Thể loại (genres): Chứa các thể loại mà bộ phim thuộc về, có thể là hành động, phiêu lưu, hài, gia đình, giả tưởng, v.v.

- Tiêu đề của bộ phim (movie_title): Chứa tiêu đề của bộ phim.

```
#tất cả columns
data.columns
```

```
Index(['color', 'director_name', 'num_critic_for_reviews', 'duration',
       'director_facebook_likes', 'actor_3_facebook_likes', 'actor_2_name',
       'actor_1_facebook_likes', 'gross', 'genres', 'actor_1_name',
       'movie_title', 'num_voted_users', 'cast_total_facebook_likes',
       'actor_3_name', 'facenumber_in_poster', 'plot_keywords',
       'movie_imdb_link', 'num_user_for_reviews', 'language', 'country',
       'content_rating', 'budget', 'title_year', 'actor_2_facebook_likes',
       'imdb_score', 'aspect_ratio', 'movie_facebook_likes'],
      dtype='object')
```

```
#sẽ chỉ dựa trên những tính năng này
data = data.loc[:,['director_name','actor_1_name','actor_2_name','actor_3_name','genres','movie_title']]
```

Hình 6. Chọn thuộc tính của cơ sở dữ liệu

Sau khi kiểm tra dữ liệu, tôi đã phát hiện nhiều giá trị khuyết trong các thuộc tính. Để xử lý vấn đề này một cách đồng đều, tôi đã quyết định thay thế những giá trị khuyết này bằng chuỗi "unknown". Bước này giúp duy trì tính toàn vẹn của dữ liệu và chuẩn bị cho các pha tiếp theo của quá trình phân tích hay xử lý dữ liệu một cách đồng nhất.

```
data.head(10)
```

	director_name	actor_1_name	actor_2_name	actor_3_name	genres	movie_title
0	James Cameron	CCH Pounder	Joel David Moore	Wes Studi	Action|Adventure|Fantasy|Sci-Fi	Avatar
1	Gore Verbinski	Johnny Depp	Orlando Bloom	Jack Davenport	Action|Adventure|Fantasy	Pirates of the Caribbean: At World's End
2	Sam Mendes	Christoph Waltz	Rory Kinnear	Stephanie Sigman	Action|Adventure|Thriller	Spectre
3	Christopher Nolan	Tom Hardy	Christian Bale	Joseph Gordon-Levitt	Action|Thriller	The Dark Knight Rises
4	Doug Walker	Doug Walker	Rob Walker	NaN	Documentary	Star Wars: Episode VII - The Force Awakens ...
5	Andrew Stanton	Daryl Sabara	Samantha Morton	Polly Walker	Action|Adventure|Sci-Fi	John Carter
6	Sam Raimi	J.K. Simmons	James Franco	Kirsten Dunst	Action|Adventure|Romance	Spider-Man 3
7	Nathan Greno	Brad Garrett	Donna Murphy	M.C. Gainey	Adventure|Animation|Comedy|Family|Fantasy|Musi...	Tangled
8	Joss Whedon	Chris Hemsworth	Robert Downey Jr.	Scarlett Johansson	Action|Adventure|Sci-Fi	Avengers: Age of Ultron
9	David Yates	Alan Rickman	Daniel Radcliffe	Rupert Grint	Adventure|Family|Fantasy|Mystery	Harry Potter and the Half-Blood Prince

```
# thay thế các giá trị NaN bằng chuỗi "unknown" trong các cột của DataFrame 'data'.
data['actor_1_name'] = data['actor_1_name'].replace(np.nan, 'unknown')
data['actor_2_name'] = data['actor_2_name'].replace(np.nan, 'unknown')
data['actor_3_name'] = data['actor_3_name'].replace(np.nan, 'unknown')
data['director_name'] = data['director_name'].replace(np.nan, 'unknown')
```

Hình 7. Thay chuỗi "unknown" vào giá trị khuyết

Sau khi kiểm tra dữ liệu, tôi đã nhận thấy rằng thông tin về thể loại có sự ngăn cách bằng dấu gạch xổ. Tôi đã quyết định thay thế dấu gạch xổ bằng khoảng trắng, giúp tạo ra một biểu diễn thể loại dễ đọc hơn. Tôi đã chuyển toàn bộ ký tự trong thuộc tính tên phim thành chữ thường. Điều này giúp tránh sự không nhất quán giữa chữ hoa và chữ thường, tạo ra một cơ sở dữ liệu mà mọi người có thể dễ dàng đọc và sử dụng. Mục đích làm cho dữ liệu trở nên đồng nhất và dễ quản lý hơn trong quá trình tiếp tục phân tích.

```
#thay thế ký tự "|" trong cột "genres" của DataFrame "data" bằng khoảng trắng (" ")
data['genres'] = data['genres'].str.replace('|', ' ')
```

```
#chuyển đổi tất cả các ký tự trong cột "movie_title" của DataFrame "data" thành chữ thường
data['movie_title'] = data['movie_title'].str.lower()
```

Hình 8. Xử lí dữ liệu trong cột thể loại và tiêu đề

Để làm sạch dữ liệu, tôi đã thực hiện việc cắt bớt ký tự cuối cùng của mỗi tiêu đề phim, giảm thiểu các ký tự vô nghĩa. Đồng thời, để tránh sự trùng lặp trong các tiêu đề phim, tôi đã loại bỏ các bản sao, chỉ giữ lại bản gốc xuất hiện đầu tiên. Điều này giúp đảm bảo tính chính xác và độ duy nhất của dữ liệu trong DataFrame. Cuối cùng là lưu dữ liệu đã xử lý và tệp data.csv

```
data['movie_title'][1]
```

```
"pirates of the caribbean: at world's end\xa0"
```

```
# Cắt bớt ký tự cuối cùng của mỗi chuỗi trong cột "movie_title"
data['movie_title'] = data['movie_title'].apply(lambda x: x[:-1])
```

```
# Loại bỏ bản sao dựa trên cột 'movie_title'
data = data.drop_duplicates(subset='movie_title', keep='first')
```

```
# Lưu dữ liệu đã xử lý vào tệp 'data.csv'
data.to_csv('data.csv', index=False)
```

Hình 9. Xử lí thuộc tính tiêu đề phim và lưu dữ liệu

3.3 Xây dựng mô hình

Bước đầu tiên của xây dựng mô hình tôi thực hiện tải và tích hợp WordNet vào dự án giúp tăng cường khả năng hiểu ngữ cảnh và xử lý ngôn ngữ trong môi trường tự nhiên. Điều này có thể hỗ trợ trong việc phân tích ý nghĩa của các từ, xây dựng mô hình ngữ nghĩa, hoặc thực hiện các tác vụ như đồng nghĩa hóa và phân loại từ vựng. Và thực hiện đọc file dữ liệu đã được xử lý.

```
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\AVITA\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

```
True
```

```
# Đọc dữ liệu từ file CSV đã được xử lý
data = pd.read_csv('data.csv')
```

Hình 10. Cài đặt WordNet và đọc dữ liệu

Tôi thực hiện kết hợp tên đạo diễn và diễn viên vào một trường văn bản duy nhất làm tăng sự đa dạng của dữ liệu và thuận lợi cho xử lý ngôn ngữ tự nhiên. Tạo điều kiện thuận lợi cho mô hình học máy hiểu rõ hơn về tương tác giữa các yếu tố. Kết quả là khả năng dự đoán của mô hình được cải thiện khi áp dụng cho dữ liệu mới. Sau đó gán giá trị cho hai biến X và Y.

```
# Kết hợp các cột để tạo một trường văn bản duy nhất
data['combined_features'] = data['director_name'] + ' ' + data['actor_1_name'] + ' ' +
data['actor_2_name'] + ' ' + data['actor_3_name']
```

```
# Gán giá trị cho nhãn
X = data['combined_features']
y = data['genres']
```

Hình 11. Kết hợp dữ liệu và gán giá trị cho biến X và Y

Sau khi tôi xác minh số lượng mẫu trong biến mục tiêu (y). Bằng cách sử dụng hàm value_counts(), chúng ta có thể đếm số lượng mẫu tương ứng với mỗi giá trị trong biến mục tiêu và hiển thị kết quả cho thấy đa dạng về thể loại thể loại xuất hiện phổ biến nhất như là: "Drama" và "Comedy." Tuy nhiên, cũng có các thể loại hiếm chỉ xuất hiện một hoặc vài lần.

```
# Xác minh số lượng mẫu trong biến y
class_counts = y.value_counts()
print(class_counts)
```

```
genres
Drama                                                 233
Comedy                                                205
Comedy Drama                                          189
Comedy Drama Romance                                  185
Comedy Romance                                        157
                                                     ...
Action Adventure Comedy Crime Family Romance Thriller   1
Biography Drama Family                                  1
Adventure Animation Family Western                      1
Action Biography Drama History Romance Western          1
Comedy Crime Horror                                     1
Name: count, Length: 914, dtype: int64
```

Hình 12. Số lượng mẫu trong biến Y

Để giảm sự không cân bằng giữa các lớp và làm cho dữ liệu trở nên cân bằng hơn, tôi thực hiện bỏ lớp có ít hơn 2 mẫu. điều này có thể cải thiện hiệu suất của mô hình học máy trong việc dự đoán thể loại của các bộ phim. Và tôi sử dụng EDA (Easy Data Augmentation) là một phương pháp tăng cường dữ liệu sử dụng các chiến lược như đồng nghĩa, chèn từ, xóa từ và đảo ngược câu. Mục tiêu là tạo ra các biến thể của

văn bản để cải thiện khả năng tổng quát hóa của mô hình học máy. Và khởi tạo hai biến mới để lưu trữ dữ liệu sau khi được tăng cường.

```
# Loại bỏ lớp có ít hơn 2 mẫu (nếu có)
y_filtered = y[y.isin(class_counts[class_counts >= 2].index)]
X_filtered = X[y.isin(class_counts[class_counts >= 2].index)]
```

```
# Tăng cường dữ liệu sử dụng EDA
eda = EDA()

X_augmented = []
y_augmented = []
```

Hình 13. Xóa lớp ít mẫu và sử dụng EDA

Thực hiện lặp qua từng mẫu trong tập dữ liệu đã lọc. Đối với mỗi mẫu, thực hiện tăng cường dữ liệu bằng cách thay thế từ bằng từ đồng nghĩa. Kết quả của tăng cường dữ liệu được thêm vào danh sách X_augmented, và nhãn tương ứng của mẫu từ tập dữ liệu đã lọc được thêm vào danh sách y_augmented. Sau khi vòng lặp hoàn thành, X_augmented và y_augmented sẽ chứa các phiên bản tăng cường của dữ liệu đầu vào và nhãn tương ứng của chúng. Sau đó sẻ được kết hợp chung với dữ liệu gốc.

```
 # Thực hiện tăng cường dữ liệu: thay thế từ bằng từ đồng nghĩa
for i in range(X_filtered.shape[0]):
    sentence = X_filtered.iloc[i]
    augmented_sentence = eda.synonym_replacement(sentence)
    X_augmented.append(augmented_sentence)
    y_augmented.append(y_filtered.iloc[i])
```

```
# Kết hợp dữ liệu mới với dữ liệu gốc
X_combined = pd.concat([X_filtered, pd.Series(X_augmented)], ignore_index=True)
y_combined = pd.concat([y_filtered, pd.Series(y_augmented)], ignore_index=True)
```

Hình 14. Tăng cường từ đồng nghĩa và kết hợp dữ liệu

Sử dụng IDF giúp đặt trọng số cao cho các từ hiếm xuất hiện trong toàn bộ tập dữ liệu, Chuyển đổi văn bản thành chữ thường, loại bỏ dấu và ký tự đặc biệt, sử dụng unigram và bigram điều này có lợi ích trong việc giữ lại thông tin về cả cấu trúc từ đơn và mối quan hệ giữa các từ trong văn bản. Loại bỏ các từ xuất hiện trong hơn 80% các văn bản và loại bỏ các từ xuất hiện trong ít hơn 2 văn bản. Sau đó chuyển đổi dữ liệu văn bản thành ma trận TF-IDF.

```
# Sử dụng TfidfVectorizer để chuyển đổi văn bản thành ma trận tf-idf
vectorizer = TfidfVectorizer(use_idf=True, lowercase=True, strip_accents='ascii',
                             ngram_range=(1, 2), max_df=0.8, min_df=2)
```

```
# Chuyển đổi dữ liệu văn bản thành ma trận tf-idf
X_combined_tfidf = vectorizer.fit_transform(X_combined)
```

Hình 15. Sử dụng TfidfVectorizes chuyển đổi ma trận

StratifiedShuffleSplit là một phương pháp chia dữ liệu dựa trên các lớp trong biến phụ thuộc, kết hợp giữa chia dữ liệu một cách ngẫu nhiên và bảo toàn cân bằng giữa các lớp. Tham số n_splits=1 chỉ định chỉ có một cách chia được tạo ra. test_size=0.2: Xác định tỷ lệ dữ liệu sử dụng cho tập kiểm tra là 20% của dữ liệu. random_state=42: Cung cấp seed để đảm bảo tái tạo ngẫu nhiên. Khi random_state được đặt với giá trị cố định, kết quả sẽ giữ nguyên qua các lần chạy mã. Cuối cùng, sau vòng lặp, các biến X_train, X_test, y_train, và y_test chứa dữ liệu đã được chia thành tập huấn luyện và tập kiểm tra theo tỷ lệ xác định. Tạo một đối tượng RandomForestClassifier với 100 cây quyết định và seed là 42. RandomForestClassifier là một mô hình ensemble dựa trên cây quyết định. Đào tạo mô hình trên tập dữ liệu huấn luyện. X_train là ma trận TF-IDF của tập huấn luyện, và y_train là các nhãn tương ứng. Mô hình sẽ học cách phân loại dữ liệu dựa trên mối quan hệ giữa đặc trưng và nhãn trong tập huấn luyện.

```
# Chia dữ liệu thành tập huấn luyện và tập kiểm tra sử dụng StratifiedShuffleSplit
sss = StratifiedShuffleSplit(n_splits=1, test_size=0.2, random_state=42)
for train_index, test_index in sss.split(X_combined_tfidf, y_combined):
    X_train, X_test = X_combined_tfidf[train_index], X_combined_tfidf[test_index]
    y_train, y_test = y_combined.iloc[train_index], y_combined.iloc[test_index]
```

```
# Sử dụng RandomForestClassifier
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)
```

Hình 16. Chia dữ liệu thành tập train và test

Tính độ chính xác trên tập kiểm tra bằng cách so sánh nhãn thực tế (y_test) với nhãn được dự đoán bởi mô hình trên tập kiểm tra (clf.predict(X_test)). Kết quả được hiển thị dưới dạng phần trăm để thể hiện hiệu suất của mô hình trên dữ liệu kiểm tra.

Sau đó, mô hình được lưu vào tệp 'nlp_model.joblib' và vectorizer được lưu vào tệp 'tfidf_vectorizer.joblib'. Việc lưu trữ mô hình và vectorizer là quan trọng để sau này có thể tái sử dụng mô hình đã được huấn luyện và vectorizer đã được tạo ra trước đó mà không cần phải huấn luyện lại từ đầu.

```
# Đánh giá độ chính xác trên tập kiểm tra
accuracy_test = accuracy_score(y_test, clf.predict(X_test)) * 100
print(f"Accuracy on Test Set: {accuracy_test:.2f}%")
```

```
Accuracy on Test Set: 82.99%
```

```
# Lưu mô hình vào tệp 'nlp_model.joblib'
filename = 'nlp_model.joblib'
dump(clf, filename)

# Lưu vectorizer để sử dụng trong quá trình dự đoán trên dữ liệu mới
vectorizer_filename = 'tfidf_vectorizer.joblib'
dump(vectorizer, vectorizer_filename)

print("Model và Vectorizer đã được lưu thành công.")
```

```
Model và Vectorizer đã được lưu thành công.
```

Hình 17. Đánh giá độ chính xác và lưu mô hình

3.4 Xây dựng ứng dụng minh họa.

Bắt đầu bằng việc nhập các thư viện cần thiết. Bao gồm Flask để xây dựng ứng dụng web, thư viện joblib để tải mô hình máy học, pandas để thao tác dữ liệu, và các thư viện khác như TfidfVectorizer và linear_kernel từ scikit-learn để xử lý văn bản và tính toán độ tương đồng cosin và tải các mô hình và dữ liệu đã được xử lý vào ứng dụng.

```
from joblib import load
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel
from flask import Flask, render_template, request, url_for

app = Flask(__name__)

# Load mô hình và vectorizer
clf = load('nlp_model.joblib')
vectorizer = load('tfidf_vectorizer.joblib')

# Load dữ liệu để hiển thị thông tin phim
data = pd.read_csv('data.csv')
```

Hình 18. Cài đặt thư viện là load mô hình

Tôi thực hiện tạo DataFrame rỗng để chứa thông tin. Có hai cột được thêm vào bảng là cột 'movie_title' để lưu trữ tựa đề của mỗi bộ phim và cột 'keywords' để chứa các từ khóa kết hợp từ tên đạo diễn, diễn viên chính và thể loại của phim. Việc kết hợp các thông tin trên thành một chuỗi từ khóa, điều này giúp tạo ra một biểu diễn đa chiều về mỗi bộ phim. Bảng từ khóa này sẽ được sử dụng để tạo ma trận TF-IDF để đo lường mức độ quan trọng của từng từ khóa trong toàn bộ tập dữ liệu phim.

```
# Tạo bảng từ khóa với 2 cột
keyword_table = pd.DataFrame()
keyword_table['movie_title'] = data['movie_title']
keyword_table['keywords'] = data['director_name'] + ' ' + data['actor_1_name']
+ ' ' + data['actor_2_name'] + ' ' + data['actor_3_name'] + ' ' + data['genres']
```

Hình 19. Tạo bảng từ khóa

Thực hiện việc sử dụng vectorizer (TF-IDF Vectorizer) để chuyển đổi các từ khóa trong bảng thành ma trận TF-IDF. Mỗi hàng của ma trận này tương ứng với một bộ phim, trong khi mỗi cột tương ứng với một từ khóa cụ thể. Giá trị tại mỗi ô của ma trận là giá trị TF-IDF của từ khóa đó trong bộ phim tương ứng. Ma trận TF-IDF này sẽ được sử dụng để đo lường sự tương đồng giữa các bộ phim dựa trên nội dung của chúng, giúp xác định các bộ phim có nội dung tương tự nhau trong quá trình đề xuất phim cho người dùng.

```
# Chuyển đổi từ khóa thành ma trận TF-IDF
tfidf_matrix = vectorizer.transform(keyword_table['keywords'])
```

Hình 20. Chuyển đổi thành ma trận TF-IDF

Trong bước này, độ tương đồng cosin giữa các bộ phim được tính toán để đo lường mức độ giống nhau giữa các từ khóa của chúng. Cụ thể, ma trận tương đồng cosin được tính bằng cách sử dụng hàm linear_kernel từ thư viện scikit-learn, nhận đầu vào là ma trận TF-IDF của các từ khóa. Trong ngữ cảnh này, vectơ đại diện cho mỗi bộ phim, và ma trận tương đồng cosin cuối cùng sẽ chứa các giá trị tương đồng giữa tất cả các cặp bộ phim trong tập dữ liệu. Ma trận tương đồng cosin này sẽ đóng vai trò quan trọng trong việc xác định các bộ phim có nội dung tương tự, là cơ sở để đề xuất các bộ phim tương đồng khi người dùng tìm kiếm hoặc xem một bộ phim cụ thể.

```
# Tính ma trận linear kernel
cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)
```

Hình 21. Tính ma trận tương đồng

Hàm này nhận vào tựa đề của bộ phim (movie_title), ma trận tương đồng cosin (cosine_sim), và một ngưỡng tương đồng (threshold) mặc định là 0.2. Trước hết, kiểm tra xem bộ phim có trong bảng từ khóa hay không. Nếu không có, in một thông báo và trả về một danh sách rỗng. Xác định chỉ số của bộ phim trong ma trận tương đồng cosin để lấy thông tin tương đồng với các bộ phim khác. Tính toán tất cả các tương đồng cosin giữa bộ phim đã chọn và tất cả các bộ phim khác. Lọc ra các bộ phim có tương đồng vượt qua ngưỡng được đặt ra. Sắp xếp các bộ phim theo tương đồng giảm dần. Lấy chỉ số của 10 bộ phim có tương đồng cao nhất (trừ bộ phim đầu tiên vì nó là chính bộ phim đang xem). Kiểm tra xem chỉ số có hợp lệ không (nếu chỉ số vượt quá số lượng bộ phim trong dữ liệu). Trả về danh sách các đối tượng phim được đề xuất với thông tin đầy đủ như tựa đề, đạo diễn, và thể loại.

```
# Hàm gợi ý với kiểm tra chỉ số hợp lệ
def recommend_movies_safe(movie_title, cosine_sim, threshold=0.2):
    movie_title = movie_title.strip()
    # Kiểm tra xem tên phim có trong bảng từ khóa không
    if movie_title not in keyword_table['movie_title'].values:
        print(f"Phim '{movie_title}' không có trong dữ liệu.")
        return []
    idx = keyword_table.index[keyword_table['movie_title'] == movie_title].tolist()[0]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = [(i, score) for i, score in sim_scores if score > threshold]
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    movie_indices = [i[0] for i in sim_scores[1:11]]
    # Kiểm tra xem chỉ số có hợp lệ không
    valid_indices = [i for i in movie_indices if i < len(data)]
    # Trả về danh sách các đối tượng phim với thông tin đầy đủ
    recommended_movies = []
    for i in valid_indices:
        movie_info = {
            'title': data['movie_title'].iloc[i],
            'director': data['director_name'].iloc[i],
            'genres': data['genres'].iloc[i]
        }
        recommended_movies.append(movie_info)
    return recommended_movies
```

Hình 22. Hàm gợi ý và kiểm tra chỉ số hợp lệ

Khi người dùng gửi một yêu cầu POST đến, hàm search() được gọi. Hàm này thực hiện các bước xử lý tìm kiếm. Hàm sẻ lấy dữ liệu từ form tìm kiếm và chuyển đổi nó thành chữ thường. Sau đó, nó gọi hàm get_movie_info để lấy thông tin chi tiết về

bộ phim được tìm kiếm. Nếu không tìm thấy thông tin về bộ phim, trang chủ được render lại với thông báo không tìm thấy. Nếu có thông tin về bộ phim, hàm recommend_movies_safe được gọi để đề xuất các bộ phim dựa trên mô hình. Cuối cùng, trang chủ được kết xuất lại với thông tin tìm kiếm và các đề xuất.

```
@app.route('/search', methods=['POST'])
def search():
    if request.method == 'POST':
        query = request.form['query'].lower()
        # Lấy thông tin của phim được tìm kiếm
        search_movie_info = get_movie_info(query)
        if not search_movie_info:
            return render_template('index.html', query=query, not_found=True)
        # Sử dụng mô hình để đề xuất phim
        recommended_movies = recommend_movies_safe(query, cosine_sim)
        return render_template('index.html', query=query, search_movie_info=search_movie_info,
                               recommended_movies=recommended_movies)
```

Hình 23. Hàm xử lý tìm kiếm

Hàm lấy thông tin phim nhận vào một tham số là movie_title, tức là tựa đề của bộ phim cần lấy thông tin. Nó kiểm tra xem tựa đề của bộ phim có trong cột 'movie_title' của tập dữ liệu không. Nếu tựa đề có trong dữ liệu, nó lấy chỉ số của bộ phim trong tập dữ liệu. Sau đó, nó tạo một đối tượng movie_info chứa thông tin chi tiết về bộ phim bao gồm 'title' (tựa đề), 'director' (đạo diễn), 'genres' (thể loại), và 'poster_url'. Cuối cùng, hàm trả về đối tượng movie_info chứa thông tin chi tiết về bộ phim. Nếu tựa đề không có trong dữ liệu, hàm trả về một đối tượng rỗng.

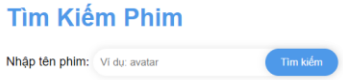
```
# Hàm để lấy thông tin của phim được tìm kiếm
def get_movie_info(movie_title):
    movie_info = {}
    if movie_title in data['movie_title'].values:
        idx = data.index[data['movie_title'] == movie_title].tolist()[0]
        movie_info = {
            'title': data['movie_title'].iloc[idx],
            'director': data['director_name'].iloc[idx],
            'genres': data['genres'].iloc[idx],
            'poster_url': url_for('static', filename=f'hinhanh/phim.jpg')
        }
    return movie_info
```

Hình 24. Hàm lấy thông tin phim

3.5 Giao diện ứng dụng minh họa.

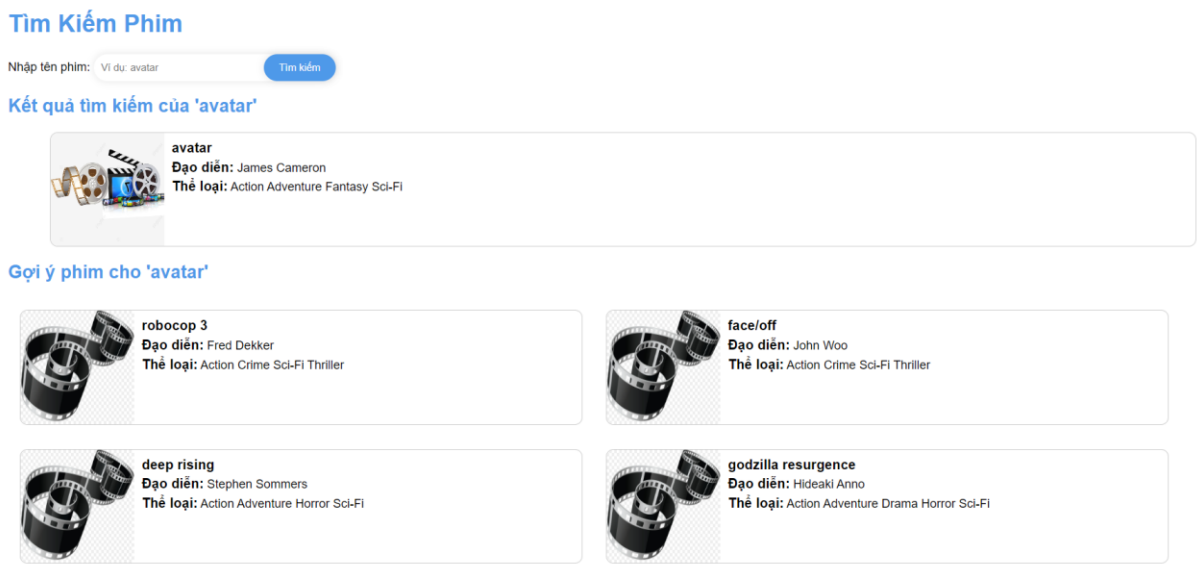
Trên trang web này cung cấp một giao diện dễ sử dụng cho người dùng. Người dùng chỉ cần nhập tên về bộ phim mà họ quan tâm vào ô tìm kiếm, ví dụ như "avatar", sau đó nhấn nút "Tìm kiếm". Hệ thống sẽ tiến hành xử lý yêu cầu tìm kiếm dựa trên thông tin từ người dùng và hiển thị kết quả tương ứng trên trang web. Điều này giúp

người dùng dễ dàng tìm kiếm và xem thông tin chi tiết về các bộ phim một cách thuận tiện, tạo trải nghiệm tìm kiếm tích cực và tương tác trên trang web.



Hình 25. Giao diện tìm kiếm

Nếu tìm kiếm của người dùng có kết quả sẽ hiển thị thông tin chi tiết về bộ phim được tìm kiếm. Tên bộ phim, đạo diễn, và thể loại sẽ được trình bày một cách trực quan. Giúp người dùng nhanh chóng có cái nhìn tổng quan về bộ phim mà họ quan tâm và có phần "Gợi ý phim" hiển thị danh sách các bộ phim được đề xuất dựa trên phim tìm kiếm của người dùng. Mỗi phim đề xuất đi kèm thông tin chi tiết như tên bộ phim, đạo diễn và thể loại. Mang lại cho người dùng cơ hội khám phá các tùy chọn phim mới và tương tự với sở thích của họ.



Hình 26. Giao diện tìm kiếm phim có tên “avatar”

Nếu không có kết quả cho tìm kiếm, người dùng sẽ nhận được thông báo rằng phim mà họ đang tìm kiếm chưa có trong hệ thống.

Hình 27. Giao diện phim chưa có trên hệ thống

CHƯƠNG 4: KẾT LUẬN

4.1 Kết quả đạt được.

Sau khi nghiên cứu và thực hiện đề tài, tôi đã tìm hiểu về các phương gợi ý và các thuật toán gợi ý cho đề tài thực tập đồ án chuyên ngành "Nghiên cứu thuật toán gợi ý theo phương pháp lọc theo nội dung và xây dựng ứng dụng minh họa". Và tôi xây dựng được một trang web tìm kiếm phim đơn giản dựa trên dữ liệu đã được xử lý và mô hình đã huấn luyện. Nó có thể giúp người dùng tìm kiếm các thông tin cần thiết giúp phục vụ cho việc truy vấn dữ liệu.

4.2 Hạn chế.

Với dữ liệu minh họa này chỉ phục vụ chính cho việc lưu trữ và truy xuất các dữ liệu đã nhập vào cơ sở dữ liệu sẵn có.

Do giao diện chủ yếu chỉ truy xuất nội dung đã được chuẩn bị sẵn nên giao diện chưa được tối ưu hóa và chưa thân thiện với người dùng.

CHƯƠNG 5: HƯỚNG PHÁT TRIỂN

Trong tương lai, tôi định hướng sẻ phát triển đề tài này thành ứng dụng web với các tính năng hoàn chỉnh như: Cập nhật đa dạng dữ liệu như: Âm thanh, hình ảnh, video,... Cập nhật thông tin người dùng và đánh giá người dùng. Mục đích giúp việc tìm kiếm các bộ phim của khách hàng dễ dàng hơn và nhiều thông tin tìm kiếm hơn, với giao diện ứng dụng trực quan và thân thiện với người dùng.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] Alberto Boschetti, Luca Massaron, Python Data Science Essentials, sencond edition, Packt Publishing, 2016.

[2] Jure Leskovec, Anand Rajaraman, Jeffrey David Ullman, Mining of Massive Datasets, 3rd edition, Rocketship VC, Stanford University, California, 2020.

[3] Hannes Hapke, Cole Howard, Hobson Lane, Natural Language Processing in Action, Manning, The United States of America, 2019.

[4] Trần Nguyễn Minh Thư và Phạm Xuân Hiền, 2016, Các phương pháp đánh giá hệ thống gợi ý, Tạp chí Khoa học Trường Đại học Cần Thơ, 42a: 18-27.

[5] Trần Nguyễn Minh Thư, Huỳnh Quang Nghi, Hệ thống gợi ý hỗ trợ tra cứu tài liệu, Tạp chí Khoa học Trường Đại học Cần Thơ, 43a: 126-134.

[6] < https://www.youtube.com/watch?v=D2V1okCEsiE>, xem 1/11/2023.

[7] <https://github.com/chaitanyaprasadnirujogi/JARVIS-MOVIE-SEARCH-AND-RECOMMENDATION-SYSTEM-USING-CONTENT-BASED-FILTERING>, xem 5/11/2023

[8] < https://vietnix.vn/tf-idf-la-gi/ >, xem 1/11/2023

[9] < https://www.youtube.com/watch?v=YMZmLx-AUvY >, xem 5/11/2023

[10] < https://www.youtube.com/watch?v=3oCtj29XeYY > , 6/11/2023