BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC THĂNG LONG**

**A black text on a white background

Description automatically generated**

**BÁO CÁO BÀI TẬP LỚN  
MÔN: TRÍ TUỆ NHÂN TẠO & CÔNG NGHỆ TRI THỨC**

**XÂY DỰNG HỆ THỐNG ĐỀ XUẤT PHIM**

**GIẢNG VIÊN HƯỚNG DẪN: NGÔ MẠNH CƯỜNG**

**LỚP HỌC PHẦN: MI322\_07**

**NHÓM THỰC HIỆN: NHÓM 9**

**THÀNH VIÊN: A45512- NGUYỄN KHẮC LỘC**

**A45179- LÊ HUY HOÀNG**

**A41737- NGUYỄN KHÔI NGUYÊN**

**HÀ NỘI – 2025**

**LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành nhất đến thầy Ngô Mạnh Cường. Trong quá trình học tập, nghiên cứu và xây dựng bài tập lớn “**Xây dựng hệ thống đề xuất phim**”, nhóm em đã nhận được rất nhiều sự quan tâm, góp ý, hướng dẫn tận tình của thầy. Thầy đã giúp nhóm tích lũy thêm nhiều kiến thức bổ ích để không chỉ hoàn thành được bài tập lớn, mà còn học được nhiều kinh nghiệm về lĩnh vực học máy và xử lý dữ liệu. Tất cả những kiến thức này đều rất hữu ích sau này cho công việc của chúng em. Có lẽ kiến thức là vô hạn mà sự tiếp nhận kiến thức của bản thân mỗi người luôn tồn tại những hạn chế nhất định. Do đó, trong quá trình hoàn thành bài tập lớn, chắc chắn không tránh khỏi những thiếu sót. Nhóm em rất mong nhận được những góp ý đến từ thầy để bài tiểu luận của nhóm được hoàn thiện hơn.

Kính chúc thầy sức khỏe, hạnh phúc và thành công trên con đường sự nghiệp giảng dạy.

**MỤC LỤC**

[CHƯƠNG 1 Giới thiệu chung 6](#_Toc200375671)

[1.1. Giới thiệu bài toán 6](#_Toc200375672)

[1.2. Ý tưởng và mục tiêu nghiên cứu 6](#_Toc200375673)

[1.2.1 Ý tưởng 6](#_Toc200375674)

[1.2.2 Mục tiêu nghiên cứu 7](#_Toc200375675)

[CHƯƠNG 2 Cơ sở lý thuyết 8](#_Toc200375676)

[2.1. Hệ thống gợi ý (Recommendation System) 8](#_Toc200375677)

[2.2. Thuật toán lọc dựa trên nội dung (Content-based Filtering) 9](#_Toc200375678)

[2.2.1 Khái niệm về thuật toán Content-based Filtering 9](#_Toc200375679)

[2.2.2 Cách xây dựng vector đặc trưng cho item. 10](#_Toc200375680)

[2.2.3 Cách đo độ tương đồng 11](#_Toc200375681)

[2.3. Thuật toán lọc cộng tác (Collaborative Filtering) 11](#_Toc200375682)

[2.3.1 Khái niệm về thuật toán Colaborative Filtering 11](#_Toc200375683)

[2.3.2 Cách xây dựng ma trận người dùng – sản phẩm 12](#_Toc200375684)

[2.3.3 Cách đo độ tương đồng 13](#_Toc200375685)

[2.4. Các công nghệ và framework sử dụng 13](#_Toc200375686)

[2.4.1 Ngôn ngữ lập trình chính- Python 13](#_Toc200375687)

[2.4.2 Xử lý dữ liệu- Pandas: 13](#_Toc200375688)

[2.4.3 Xử lý văn bản 14](#_Toc200375689)

[2.4.4 Mô hình học máy 15](#_Toc200375690)

[2.4.5 Giao diện người dùng 15](#_Toc200375691)

[CHƯƠNG 3 Thực nghiệm 17](#_Toc200375692)

[3.1. Mô tả bài toán nghiệp vụ 17](#_Toc200375693)

[3.1.1 Bối cảnh và thách thức gặp phải 17](#_Toc200375694)

[3.1.2 Nhu cầu và chiến lược tiếp cận 17](#_Toc200375695)

[3.1.3 Giải pháp công nghệ gắn với mục tiêu kinh doanh 17](#_Toc200375696)

[3.2. Yêu cầu và tiêu chí đánh giá hệ thống 18](#_Toc200375697)

[3.3. Thu thập và tiền xử lý dữ liệu 19](#_Toc200375698)

[3.3.1 Nguồn dữ liệu sử dụng. 19](#_Toc200375699)

[3.3.2 Làm sạch và chuẩn hóa dữ liệu. 20](#_Toc200375700)

[3.4. Xây dựng thuật toán gợi ý 21](#_Toc200375701)

[3.4.1 Mô hình Content-based Filtering 21](#_Toc200375702)

[3.4.2 Mô hình Collaborative Filtering 23](#_Toc200375703)

[3.5. Ứng dụng hệ thống vào thực tế 24](#_Toc200375704)

[3.5.1 Mô tả hệ thống 24](#_Toc200375705)

[3.5.2 Kiến trúc tổng quan 24](#_Toc200375706)

[3.5.3 Đặc tả Use Case 25](#_Toc200375707)

[3.5.4 Luồng hoạt động 26](#_Toc200375708)

[3.5.5 Hình ảnh hệ thống 28](#_Toc200375709)

[3.6. Đánh giá kết quả và tác động 28](#_Toc200375710)

[CHƯƠNG 4 Kết luận và định hướng trong tương lai 30](#_Toc200375711)

[4.1. Kết luận 30](#_Toc200375712)

[4.2. Định hướng trong tương lai 30](#_Toc200375713)

[Bảng phân công nhiệm vụ 32](#_Toc200375714)

[Tài liệu tham khảo 33](#_Toc200375715)

**DANH MỤC ẢNH MINH HỌA**

[Ảnh 1. Thuật toán trong hệ thống gợi ý 9](#_Toc200376491)

[Ảnh 2. Mô tả thuật toán Content- based Filtering 9](#_Toc200376492)

[Ảnh 3. Mô tả thuật toán Colaborative Filtering 11](#_Toc200376493)

[Ảnh 4. Ma trận người dùng- sản phẩm 12](#_Toc200376494)

[Ảnh 5. Quy trình xử lý bài toán 18](#_Toc200376495)

[Ảnh 6. Luồng dữ liệu của hệ thống 21](#_Toc200376496)

[Ảnh 7. Kiến trúc hệ thống 25](#_Toc200376497)

[Ảnh 8. Cách người dùng tương tác, hoạt động với hệ thống 27](#_Toc200376498)

[Ảnh 9. Hình ảnh giao diện hệ thống với thuật toán Content- based Filtering 28](#_Toc200376499)

[Ảnh 10. Hình ảnh giao diện hệ thống với thuật toán Collaborative Filtering 28](#_Toc200376500)

**DANH MỤC BẢNG MINH HỌA**

[Bảng 1. Mô tả chức năng chính của hệ thống 24](#_Toc200376559)

[Bảng 2. Use case tìm kiếm phim 26](#_Toc200376560)

[Bảng 3. Use case hiển thị phim được đề xuất 26](#_Toc200376561)

# Giới thiệu chung

## Giới thiệu bài toán

Trong thời đại số hiện nay, nhu cầu giải trí qua các nền tảng xem phim trực tuyến như Netflix, Disney+, Amazon Prime, v.v... ngày càng gia tăng mạnh mẽ. Sự bùng nổ về số lượng phim được sản xuất và phát hành mỗi năm khiến người dùng đứng trước một kho nội dung khổng lồ và gặp khó khăn trong việc lựa chọn bộ phim phù hợp với sở thích cá nhân. Việc phải tìm kiếm thủ công những bộ phim yêu thích không chỉ tốn thời gian mà còn dễ dẫn đến trải nghiệm không như mong đợi.

Bài toán đặt ra là làm sao để xây dựng một hệ thống thông minh có khả năng hiểu được sở thích và hành vi xem phim của từng người dùng, từ đó đưa ra những gợi ý phù hợp và mang tính cá nhân hóa cao. Đây chính là vai trò của **hệ thống đề xuất phim** – một ứng dụng thực tiễn của các kỹ thuật học máy (machine learning), học sâu (deep learning), và khai phá dữ liệu người dùng (user data mining).

Hệ thống đề xuất có thể phân tích lịch sử xem phim, đánh giá (rating), hành vi tương tác, và các thông tin liên quan của người dùng để xây dựng mô hình dự đoán. Mô hình này giúp gợi ý các bộ phim tương đồng với những gì người dùng yêu thích, từ đó tối ưu hóa trải nghiệm cá nhân, tiết kiệm thời gian tìm kiếm và giữ chân người dùng lâu hơn trên nền tảng.

Ngoài ra, từ góc độ doanh nghiệp, hệ thống đề xuất còn góp phần quan trọng trong việc tăng doanh thu, cải thiện tỷ lệ giữ chân khách hàng và nâng cao mức độ hài lòng của người dùng. Nhờ đó, các nền tảng xem phim có thể thu hút thêm khách hàng tiềm năng và tối ưu hóa chiến lược phát triển nội dung.

## Ý tưởng và mục tiêu nghiên cứu

### Ý tưởng

Trong bối cảnh nhu cầu giải trí bằng phim ảnh ngày càng gia tăng, người dùng phải đối mặt với một "rừng" lựa chọn trên các nền tảng số như Netflix, Disney+, FPT Play,... Việc chọn ra một bộ phim phù hợp với sở thích cá nhân trở thành một thách thức lớn, gây mất thời gian và dễ bỏ lỡ những nội dung phù hợp. Trước thực trạng đó, việc xây dựng hệ thống đề xuất phim thông minh là một giải pháp cần thiết để hỗ trợ người dùng cá nhân hóa trải nghiệm xem phim của mình.

Xuất phát từ nhu cầu này, đề tài hướng đến việc xây dựng một hệ thống đề xuất phim bằng cách ứng dụng hai kỹ thuật phổ biến nhất trong hệ thống gợi ý hiện nay, đó là:

* Content-based Filtering: Đề xuất các bộ phim có đặc điểm tương đồng với những phim mà người dùng đã từng thích hoặc đánh giá cao (về thể loại, nội dung, diễn viên, đạo diễn,...).
* Collaborative Filtering: Dựa vào hành vi và đánh giá của những người dùng có sở thích tương tự để gợi ý phim cho người dùng hiện tại.

Thông qua việc kết hợp cả hai phương pháp, hệ thống không chỉ đảm bảo được độ chính xác mà còn mở rộng tính đa dạng và khả năng cá nhân hóa, mang lại trải nghiệm tối ưu cho người dùng.

### Mục tiêu nghiên cứu

Dựa trên ý tưởng xây dựng hệ thống đề xuất phim thông minh, mục tiêu của nghiên cứu này là:

* Xây dựng và phát triển một hệ thống đề xuất phim có khả năng cá nhân hóa trải nghiệm xem phim của người dùng, giúp họ dễ dàng tìm thấy những nội dung phù hợp với sở thích cá nhân.
* Ứng dụng và tích hợp hai kỹ thuật chính:
  + Content-based Filtering: Phát triển module đề xuất dựa trên đặc điểm nội dung của phim (thể loại, diễn viên, đạo diễn, từ khóa, cốt truyện) để gợi ý những bộ phim tương đồng với lịch sử xem và đánh giá của người dùng.
  + Collaborative Filtering: Xây dựng module đề xuất dựa trên hành vi và sở thích của cộng đồng người dùng, xác định những người dùng có thị hiếu tương tự để đưa ra gợi ý phim phù hợp.
* Nghiên cứu và đánh giá hiệu quả của việc kết hợp hai phương pháp Content-based Filtering và Collaborative Filtering trong việc cải thiện độ chính xác và đa dạng của các đề xuất phim.
* Phát triển một giao diện người dùng thân thiện (nếu có) để người dùng có thể tương tác dễ dàng với hệ thống, nhận các đề xuất và cung cấp phản hồi.

# Cơ sở lý thuyết

Để xây dựng một hệ thống đề xuất phim hiệu quả, cần phải kết hợp nhiều kiến thức và công nghệ trong lĩnh vực khoa học dữ liệu, trí tuệ nhân tạo và lập trình phần mềm. Trong chương này, chúng ta sẽ đi sâu vào các thuật toán, mô hình và công nghệ nền tảng đã được sử dụng trong quá trình phát triển hệ thống, bao gồm cả các khái niệm về hệ thống gợi ý, hai kỹ thuật Content-based Filtering và Collaborative Filtering, cũng như các thư viện và framework hỗ trợ triển khai mô hình trong thực tế.

Những cơ sở lý thuyết này không chỉ đóng vai trò nền tảng trong việc thiết kế thuật toán gợi ý, mà còn giúp đảm bảo tính chính xác, hiệu quả và khả năng mở rộng của hệ thống.

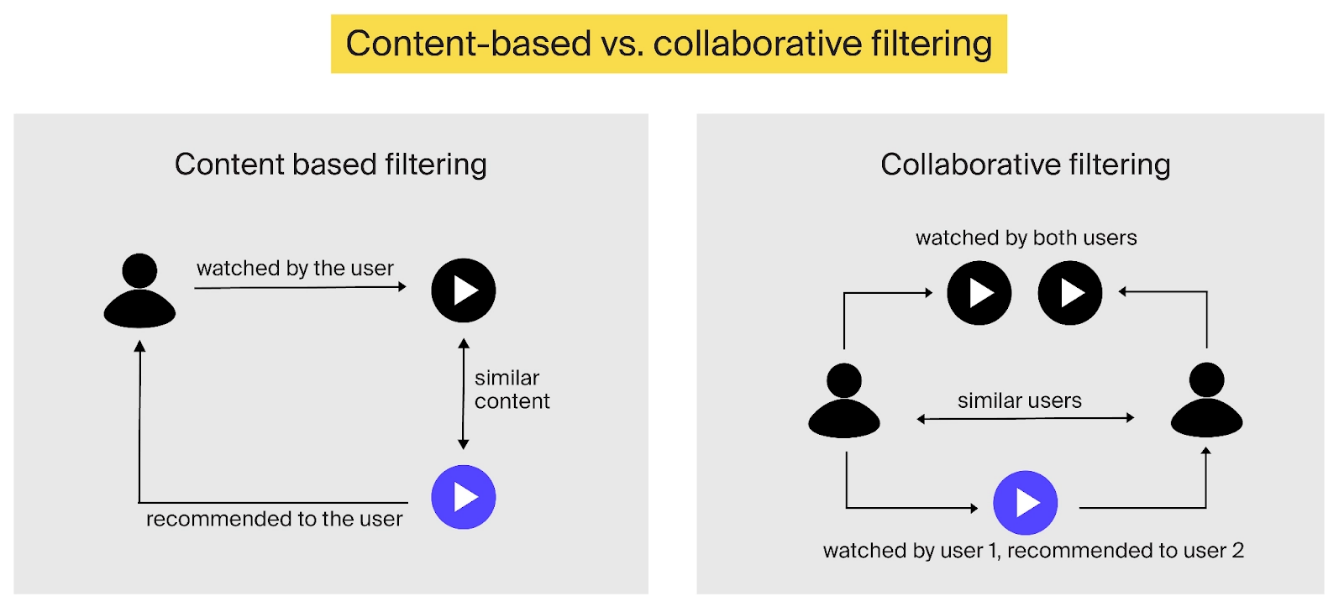
## Hệ thống gợi ý (Recommendation System)

Hệ thống gợi ý (Recommender systems hoặc Recommendation systems) là một dạng của hệ hỗ trợ ra quyết định, cung cấp giải pháp mang tính cá nhân hóa mà không phải trải qua quá trình tìm kiếm phức tạp. Hệ gợi ý học từ người dùng và gợi ý các sản phẩm tốt nhất trong số các sản phẩm phù hợp.

Hệ thống gợi ý sử dụng các tri thức về sản phẩm, các tri thức của chuyên gia hay tri thức khai phá học được từ hành vi con người dùng để đưa ra các gợi ý về sản phẩm mà họ thích trong hàng ngàn hàng vạn sản phẩm có trong hệ thống. Các website thương mại điện tử, ví dụ như sách, phim, nhạc, báo...sử dụng hệ thống gợi ý để cung cấp các thông tin giúp cho người sử dụng quyết định sẽ lựa chọn sản phẩm nào. Các sản phẩm được gợi ý dựa trên số lượng sản phẩm đó đã được bán, dựa trên các thông tin cá nhân của người sử dụng, dựa trên sự phân tích hành vi mua hàng trước đó của người sử dụng để đưa ra các dự đoán về hành vi mua hàng trong tương lai của chính khách hàng đó. Các dạng gợi ý bao gồm: gợi ý các sản phẩm tới người tiêu dùng, các thông tin sản phẩm mang tính cá nhân hóa, tổng kết các ý kiến cộng đồng, và cung cấp các chia sẻ, các phê bình, đánh giá mang tính cộng đồng liên quan tới yêu cầu, mục đích của người sử dụng đó.

Có 3 phương pháp gợi ý chính thường được sử dụng trong lĩnh vực xây dựng hệ thống gợi ý, đó là:

* Hệ thống gợi ý dựa theo lọc cộng tác (Collaborative recommendation systems): là phương pháp gợi ý được triển khai rộng rãi nhất và thành công nhất trong thực tế.
* Hệ thống gợi ý dựa theo nội dung (Content-based recommendation systems): là sự kế thừa và mở rộng của lĩnh vực nghiên cứu lọc thông tin.
* Hệ thống gợi ý dựa trên cơ sở tri thức (Knowledge-based recommenders systems): gợi ý các đối tượng dựa trên các suy luận về nhu cầu và sở thích của người dùng.

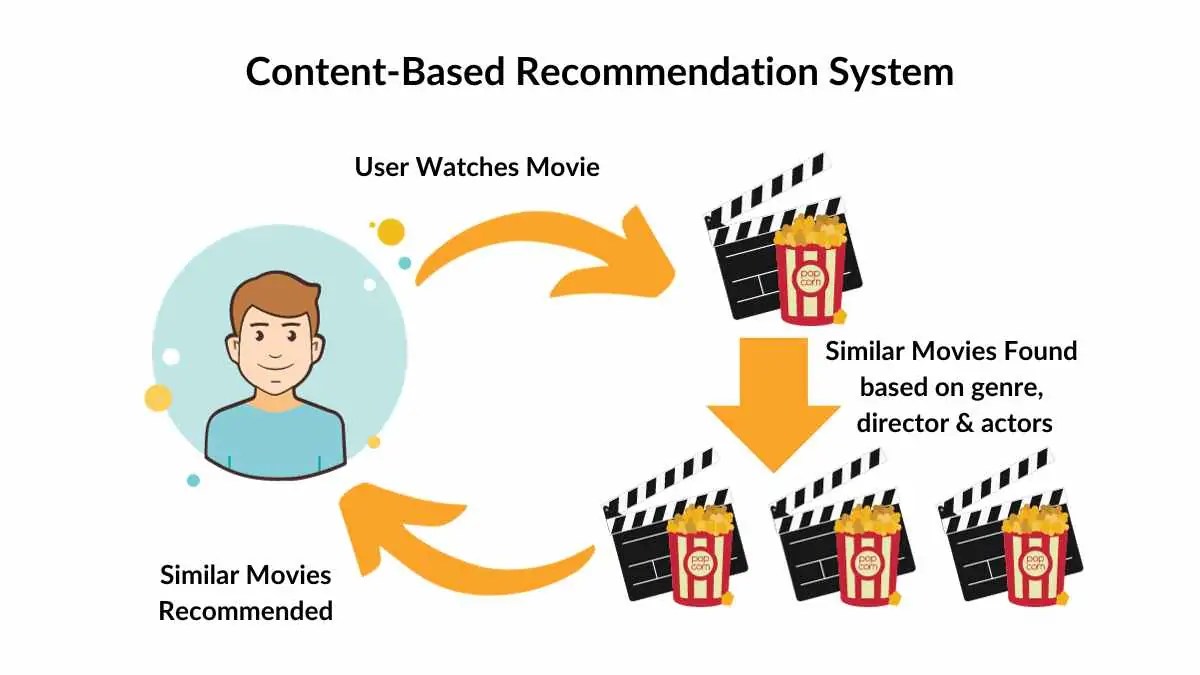


Ảnh 1. Thuật toán trong hệ thống gợi ý

## Thuật toán lọc dựa trên nội dung (Content-based Filtering)

### Khái niệm về thuật toán Content-based Filtering

Content-Based Filtering là một phương pháp phổ biến trong hệ thống gợi ý (recommender systems), sử dụng thông tin mô tả của các mục (item) để đề xuất các mục tương tự với những gì người dùng đã thích hoặc tương tác trước đó. Phương pháp này dựa trên giả định rằng người dùng có xu hướng thích những nội dung có đặc điểm giống với những gì họ từng đánh giá cao.



Ảnh 2. Mô tả thuật toán Content- based Filtering

Cách thức hoạt động của Content-based Filtering gồm các bước chính như sau:

* Bước 1: Hệ thống phân tích các mục (ví dụ: phim) dựa trên các đặc trưng như thể loại, đạo diễn, diễn viên, mô tả nội dung, từ khóa, v.v. để xây dựng một vector đặc trưng cho từng mục.
* Bước 2: Từ lịch sử người dùng (những phim đã xem, đã đánh giá cao), hệ thống tổng hợp đặc trưng của các mục đó để xây dựng hồ sơ người dùng (user profile).
* Bước 3: Hệ thống tính độ tương đồng giữa các mục mới và hồ sơ người dùng, thường sử dụng độ tương đồng cosine.
* Bước 4: Những mục có độ tương đồng cao sẽ được gợi ý cho người dùng.

Ưu điểm của phương pháp này là không phụ thuộc vào dữ liệu của người dùng khác, phù hợp khi hệ thống còn ít người dùng (cold start với user mới). Tuy nhiên, hạn chế là có thể bị giới hạn về tính đa dạng do chỉ đề xuất những gì tương tự với sở thích trước đó.

### Cách xây dựng vector đặc trưng cho item.

Để hệ thống có thể so sánh và gợi ý các bộ phim tương tự nhau, mỗi item (ở đây là phim) cần được biểu diễn dưới dạng vector đặc trưng. Các bước cụ thể như sau:

* Thu thập đặc trưng: Thu thập các thuộc tính mô tả bộ phim như: thể loại (genre), đạo diễn (director), diễn viên chính (main actors), từ khóa mô tả (keywords), tóm tắt nội dung (description), v.v.
* Tiền xử lý văn bản (nếu dùng mô tả): Bao gồm chuyển văn bản về chữ thường, loại bỏ dấu câu, stopwords, stemming, v.v.
* Chuyển đổi về vector số:
  + Nếu dữ liệu dạng văn bản (như mô tả phim hoặc từ khóa): sử dụng TF-IDF vectorizer (hoặc CountVectorizer).
    - Sử dụng TF- IDF:

Cho một tài liệu ddd trong tập tài liệu DDD, và một từ khóa ttt, thì TF-IDF được tính như sau:

* + - * Term Frequency (TF) - Tần suất xuất hiện của từ t trong tài liệu d:
      * Inverse Document Frequency (IDF) - Độ đặc hiệu của từ t trong tập tài liệu D:   
        [(trong đó ∣D| là tổng số tài liệu, ∣{ d ∈ D : t ∈ d}∣ là số tài liệu chứa từ t, và 1 được cộng vào mẫu số để tránh chia cho 0)]
      * TF- IDF:

Kết quả thu được là một ma trận đặc trưng, trong đó mỗi hàng là một phim, và mỗi cột là một thuộc tính đặc trưng đã mã hóa.

### Cách đo độ tương đồng

Sau khi xây dựng được các vector đặc trưng cho các bộ phim, ta có thể sử dụng các phép đo độ tương đồng để tính xem hai phim giống nhau tới mức nào, ta sử dụng ma trận Cosine Similarity:

A, B: là hai vecto đánh giá

→ Kết quả là một giá trị từ 0 đến 1, càng gần 1 thì càng tương đồng.

## Thuật toán lọc cộng tác (Collaborative Filtering)

### Khái niệm về thuật toán Colaborative Filtering

Thuật toán Collaborative Filtering (lọc cộng tác) là một trong những phương pháp phổ biến và hiệu quả nhất trong hệ thống gợi ý. Phương pháp này đưa ra các đề xuất dựa trên hành vi, đánh giá hoặc sở thích của cộng đồng người dùng.

A diagram of a person with a movie clapper board

AI-generated content may be incorrect.

Ảnh 3. Mô tả thuật toán Colaborative Filtering

Ý tưởng cơ bản của thuật toán này là dự đoán mức độ yêu thích của một user đối với một item dựa trên các users khác “gần giống” với user đang xét. Việc xác định độ “giống nhau” giữa các users có thể dựa vào mức độ quan tâm (rating) của các users này với các items khác mà hệ thống đã biết trong quá khứ

Collaborative Filtering được chia thành hai loại chính:

* User-based Collaborative Filtering: Đề xuất dựa trên người dùng có hành vi tương tự. Ví dụ: “Người dùng giống bạn đã xem...”
* Item-based Collaborative Filtering: Đề xuất dựa trên các sản phẩm có mẫu đánh giá tương tự. Ví dụ: “Những người đã xem phim này cũng thích các phim sau...”

Khác với các phương pháp dựa trên nội dung (content-based), lọc cộng tác không yêu cầu hiểu rõ đặc điểm của sản phẩm, mà chỉ cần dựa trên tương quan giữa người dùng và các sản phẩm đã tương tác.

### Cách xây dựng ma trận người dùng – sản phẩm

Cốt lõi của collaborative filtering là một ma trận đánh giá (user-item rating matrix), trong đó mỗi dòng đại diện cho một người dùng, mỗi cột đại diện cho một item, và mỗi ô thể hiện mức độ yêu thích (ví dụ: số sao, số lượt xem, số lượt mua) của người dùng đó đối với item tương ứng.

Do người dùng thường chỉ tương tác với một phần nhỏ các item, nên ma trận này thường thưa (sparse matrix), và nhiều giá trị bị thiếu. Hệ thống cần ước lượng (predict) các giá trị chưa biết này để đưa ra gợi ý.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Người dùng | | | |
| Sản phẩm |  | Phim 1 | Phim 2 | Phim 3 |
| User A | 5 | 3 | ? |
| User B | 2 | ? | 4 |
| User C | 1 | ? | 3 |

Ảnh 4. Ma trận người dùng- sản phẩm

Để dự đoán đánh giá của người dùng u với item i dựa trên đánh giá của các người dùng tương tự với u đã đánh giá item i ta sử dụng công thức:

A screenshot of a math problem

AI-generated content may be incorrect.

### Cách đo độ tương đồng

Sau khi xây dựng được các vector đặc trưng cho các bộ phim, ta có thể sử dụng các phép đo độ tương đồng để tính xem hai phim giống nhau tới mức nào, ta sử dụng ma trận Cosine Similarity:

A, B: là hai vecto đánh giá

→ Kết quả là một giá trị từ 0 đến 1, càng gần 1 thì càng tương đồng.

## Các công nghệ và framework sử dụng

### Ngôn ngữ lập trình chính- Python

Python là ngôn ngữ lập trình máy tính bậc cao thường được sử dụng để xây dựng trang web và phần mềm, tự động hóa các tác vụ và tiến hành phân tích dữ liệu. Python là ngôn ngữ có mục đích chung, nghĩa là nó có thể được sử dụng để tạo nhiều chương trình khác nhau và không chuyên biệt cho bất kỳ vấn đề cụ thể nào. Tính linh hoạt này, cùng với sự thân thiện với người mới bắt đầu, đã khiến nó trở thành một trong những ngôn ngữ lập trình được sử dụng nhiều nhất hiện nay. Một cuộc khảo sát được thực hiện bởi công ty phân tích ngành RedMonk cho thấy rằng đây là ngôn ngữ lập trình phổ biến thứ hai đối với các nhà phát triển vào năm 2021.

### Xử lý dữ liệu- Pandas:

Pandas là một thư viện cung cấp các cấu trúc dữ liệu và công cụ phân tích dữ liệu mạnh mẽ cho Python. Được xây dựng trên nền tảng của NumPy, Pandas cho phép thao tác dữ liệu dạng bảng (tương tự như bảng tính Excel) một cách dễ dàng. Các cấu trúc dữ liệu chính của Pandas là DataFrame và Series, cho phép người dùng thực hiện các thao tác như lọc, nhóm, và biến đổi dữ liệu một cách linh hoạt và hiệu quả. Ví dụ, nếu chúng ta có một bảng dữ liệu về nhân viên và muốn lọc ra những người có tuổi trên 30, Pandas có thể giúp chúng ta thực hiện điều này một cách dễ dàng.

### Xử lý văn bản

1. Xử lý chuỗi và biểu thức chính quy:

* re: Thư viện chuẩn trong Python hỗ trợ xử lý các biểu thức chính quy (regex), giúp tìm kiếm, tách chuỗi, thay thế và phân tích cú pháp văn bản hiệu quả.
* ast: Dùng để chuyển đổi biểu diễn dạng chuỗi của các cấu trúc dữ liệu như danh sách, tuple hoặc dictionary thành các đối tượng Python tương ứng. Điều này hữu ích khi làm việc với dữ liệu được lưu dưới dạng chuỗi trong tập tin hoặc cơ sở dữ liệu.

1. Xử lý ngôn ngữ tự nhiên (NLP):

* nltk (Natural Language Toolkit): Một thư viện mã nguồn mở phổ biến trong lĩnh vực xử lý ngôn ngữ tự nhiên.
* stopwords: Dùng để loại bỏ các từ dừng – những từ phổ biến không mang nhiều giá trị ngữ nghĩa như “the”, “is”, “in”,...
* WordNetLemmatizer: Chuẩn hóa từ về dạng gốc (lemmatization), giúp giảm nhiễu và tăng tính chính xác khi so sánh văn bản.
* RegexpTokenizer: Công cụ tách văn bản thành các token (từ) dựa trên biểu thức chính quy, linh hoạt trong việc xử lý các kiểu định dạng đầu vào.

1. Biểu diễn và so sánh văn bản:

* TfidfVectorizer (thuộc sklearn.feature\_extraction.text): Dùng để chuyển đổi văn bản thành vector số bằng kỹ thuật TF-IDF (Term Frequency - Inverse Document Frequency), giúp lượng hóa tầm quan trọng của từ trong văn bản.
* cosine\_similarity (thuộc sklearn.metrics.pairwise): Tính độ tương đồng giữa hai vector văn bản thông qua hệ số cosine, rất phù hợp để so sánh mức độ giống nhau giữa các mô tả phim hoặc phản hồi người dùng.

1. So khớp gần đúng (Approximate Matching):

* difflib.get\_close\_matches: Hàm tìm kiếm các chuỗi gần giống nhất trong một danh sách đầu vào, rất hữu ích khi thực hiện so khớp mờ (fuzzy matching), đặc biệt khi người dùng nhập sai chính tả hoặc nhập không chính xác tên phim.

### Mô hình học máy

Hệ thống đề xuất sử dụng mô hình dựa trên kỹ thuật học máy không giám sát để tính toán độ tương đồng giữa các phim, qua đó đưa ra các đề xuất phù hợp. Các công cụ hỗ trợ bao gồm:

* + - **Scikit-learn** (sklearn): Đây là thư viện học máy phổ biến trong Python, cung cấp các công cụ phục vụ xây dựng và đánh giá mô hình. Trong đề tài này, các thành phần chính được sử dụng là:
    - **TfidfVectorizer**: Biến đổi dữ liệu văn bản thành vector đặc trưng.
    - **Cosine Similarity**: Tính toán độ tương đồng giữa các vector văn bản, nhằm xác định mức độ liên quan giữa các phim dựa trên mô tả nội dung.

Mô hình này không yêu cầu dữ liệu đánh giá từ người dùng mà hoạt động dựa trên đặc điểm nội dung, do đó phù hợp cho cả người dùng mới và các phim chưa có nhiều phản hồi.

### Giao diện người dùng

Trong đề tài này, giao diện người dùng được xây dựng bằng các công nghệ cơ bản của phát triển web, bao gồm:

* + HTML (HyperText Markup Language): Ngôn ngữ đánh dấu dùng để xây dựng cấu trúc và bố cục của trang web. HTML giúp hiển thị các thành phần như ô nhập dữ liệu, nút bấm và kết quả gợi ý phim.
  + CSS (Cascading Style Sheets): Ngôn ngữ dùng để định dạng và thiết kế giao diện trang web, bao gồm màu sắc, kiểu chữ, bố cục và hiệu ứng hiển thị nhằm nâng cao trải nghiệm người dùng.
  + Flash: Công nghệ được sử dụng để xây dựng các hiệu ứng động hoặc tương tác nâng cao trên giao diện. Giao diện này cho phép người dùng:
* Nhập vào tên phim muốn tìm kiếm gợi ý.
* Xem danh sách các phim được hệ thống đề xuất.
* Hiển thị một số thông tin mô tả ngắn gọn về các phim liên quan.

Tuy không sử dụng các framework hiện đại như React, Angular hay Streamlit, giao diện vẫn đảm bảo sự đơn giản, dễ sử dụng và đáp ứng đúng chức năng cốt lõi của hệ thống gợi ý.

# Thực nghiệm

## Mô tả bài toán nghiệp vụ

### Bối cảnh và thách thức gặp phải

Ngành công nghiệp giải trí trực tuyến đang phát triển mạnh mẽ, với sự cạnh tranh ngày càng cao giữa các nền tảng phát video theo yêu cầu (Video on demand). Doanh nghiệp phải đối mặt với các thách thức lớn:

* Lượng nội dung khổng lồ: Người dùng có quá nhiều lựa chọn, dẫn đến khó khăn trong việc tìm kiếm nội dung phù hợp.
* Tỷ lệ rời bỏ cao: Nếu nền tảng không giúp người dùng khám phá phim dễ dàng, họ sẽ chuyển sang dịch vụ khác.
* Tối ưu hóa doanh thu: Các đề xuất phim không chỉ ảnh hưởng đến trải nghiệm mà còn tác động đến chiến lược quảng cáo, đăng ký thuê bao và doanh thu từ nội dung trả phí.

### Nhu cầu và chiến lược tiếp cận

Để duy trì sự cạnh tranh và tăng trưởng, doanh nghiệp cần một hệ thống gợi ý phim thông minh nhằm:

* Cá nhân hóa nội dung: Hiển thị đúng bộ phim phù hợp với sở thích và thói quen của từng người dùng.
* Giữ chân khách hàng: Cải thiện trải nghiệm giúp tăng tỷ lệ tương tác và duy trì thuê bao dài hạn.
* Tăng hiệu quả quảng bá: Hệ thống gợi ý có thể hỗ trợ chiến lược tiếp thị, giúp quảng bá nội dung phù hợp với từng phân khúc khách hàng.

### Giải pháp công nghệ gắn với mục tiêu kinh doanh

Một hệ thống gợi ý hiệu quả có thể kết hợp các mô hình trí tuệ nhân tạo như:

* Content-Based Filtering: Phân tích nội dung phim để đề xuất dựa trên đặc điểm giống nhau.
* Collaborative Filtering: Học từ hành vi của người dùng khác để tạo đề xuất phù hợp.

Việc ứng dụng hệ thống này không chỉ nâng cao trải nghiệm người dùng mà còn mang lại nhiều lợi ích kinh doanh rõ rệt. Cụ thể, nó giúp tăng tỷ lệ giữ chân khách hàng, từ đó giảm thiểu việc người dùng rời bỏ dịch vụ. Doanh thu cũng được tối ưu hóa thông qua việc thúc đẩy lượt xem nội dung trả phí và quảng cáo. Đồng thời, chiến lược marketing trở nên hiệu quả hơn nhờ khả năng cá nhân hóa cách tiếp cận từng khách hàng tiềm năng.

## Yêu cầu và tiêu chí đánh giá hệ thống

Hệ thống gợi ý phim cần đáp ứng các yêu cầu sau nhằm phục vụ mục tiêu nghiệp vụ trong việc nâng cao trải nghiệm người dùng và tối ưu hóa khả năng khám phá nội dung:

* Dễ sử dụng: Giao diện trực quan, thao tác tìm kiếm đơn giản, phù hợp với mọi đối tượng người dùng.
* Không yêu cầu đăng nhập để sử dụng cơ bản: Người dùng có thể tìm kiếm phim tương tự mà không cần tạo tài khoản.
* Cá nhân hóa khi đăng nhập: Với người dùng đã đăng nhập, hệ thống cần cung cấp gợi ý cá nhân hóa dựa trên lịch sử tương tác hoặc mô hình cộng tác.
* Tốc độ phản hồi nhanh: Hệ thống phải trả kết quả gợi ý trong thời gian ngắn (dưới 2 giây với truy vấn phổ biến).
* Khả năng mở rộng: Có thể tích hợp thêm các loại dữ liệu khác như thể loại, đạo diễn, năm phát hành,... trong tương lai.
* Dễ triển khai: Có thể cài đặt và chạy được trên máy tính cá nhân hoặc server nhẹ.

Sau khi xác định các yêu cầu và tiêu chí đánh giá hệ thống, ta cần hiểu rõ quy trình hoạt động của hệ thống gợi ý. Sơ đồ dưới đây là flowchart, minh họa từng bước trong quá trình hệ thống xử lý dữ liệu, từ khi tiếp nhận thông tin đầu vào, áp dụng thuật toán gợi ý, đến khi hiển thị kết quả cho người dùng. Flowchart giúp mô tả chính xác cách hệ thống vận hành, đảm bảo hiệu quả và khả năng mở rộng.

A black background with white rectangles

AI-generated content may be incorrect.

Ảnh 5. Quy trình xử lý bài toán

## Thu thập và tiền xử lý dữ liệu

### Nguồn dữ liệu sử dụng.

1. TMDB 5000 Movie Dataset

* Đặc điểm dữ liệu:
  + tmdb\_5000\_credits.csv: 4 trường và 4803 dòng
  + tmdb\_5000\_movies.csv: 20 trường và 4803 dòng
* Các trường thông tin chính:
  + movie\_id: Mã định danh phim
  + title: Tên phim
  + overview: Mô tả nội dung
  + genres: Thể loại phim
  + keywords: Từ khóa liên quan
  + cast: Diễn viên
  + crew: Đội ngũ sản xuất
* Mục đích sử dụng: Cung cấp thông tin chi tiết về nội dung phim, phục vụ cho thuật toán Content-based Filtering.

1. Movie Lens Small Latest Dataset

* Quy mô dữ liệu:
  + 100,836 đánh giá
  + 9,742 bộ phim
  + 610 người dùng
* Thời gian thu thập:
  + Từ 29/03/1996 đến 24/09/2018
* Đặc điểm dữ liệu:
  + Ratings: Điểm đánh giá của user cho phim
  + Tags: Nhãn người dùng gán cho phim
  + User demographics: Thông tin người dùng
  + Movie metadata: Thông tin chi tiết phim
* Mục đích sử dụng: Cung cấp dữ liệu tương tác giữa người dùng và phim, hỗ trợ xây dựng hệ thống gợi ý dựa trên cộng tác (Collaborative Filtering).

### Làm sạch và chuẩn hóa dữ liệu.

a. TMDB 5000 Movie Dataset

Quá trình tiền xử lý dữ liệu từ TMDB 5000 bao gồm các bước sau:

* + Chuyển đổi các trường dữ liệu dạng JSON:  
    Một số cột như genres, keywords, cast, crew có định dạng chuỗi JSON. Các trường này được chuyển đổi sang cấu trúc danh sách để dễ dàng truy xuất thông tin.
  + Xóa các khoảng trắng thừa:  
    Các chuỗi văn bản trong cột như overview, keywords được loại bỏ khoảng trắng dư thừa nhằm đồng nhất định dạng dữ liệu.
  + Chuyển đổi về chữ thường (lowercase):  
    Tất cả văn bản được chuẩn hóa bằng cách chuyển về chữ thường để tránh sự khác biệt không cần thiết khi xử lý ngôn ngữ tự nhiên.
  + Loại bỏ từ dừng (stopwords):  
    Các từ phổ biến không mang nhiều ý nghĩa như “the”, “is”, “and”... được loại bỏ khỏi văn bản để tập trung vào các từ mang nội dung quan trọng.
  + Lemmatization (Chuẩn hóa từ gốc):  
    Từ ngữ được đưa về dạng gốc để giảm thiểu sự đa dạng về hình thái từ. Ví dụ, “running”, “ran” đều được đưa về “run”.
  + Vector hóa văn bản:  
    Sau khi xử lý văn bản, các mô tả phim và từ khóa được chuyển thành vector số sử dụng kỹ thuật TF-IDF (đã trình bày ở phần 2: Cơ sở lý thuyết), giúp mô hình có thể xử lý và tính toán mức độ tương đồng giữa các phim.

b. Movie Lens Small Latest Dataset

Quá trình xử lý dữ liệu từ tập MovieLens Small Latest Dataset tập trung vào việc chuẩn hóa thông tin người dùng và tương tác giữa người dùng với phim, gồm các bước sau:

* Lọc dữ liệu cần thiết: Từ toàn bộ tập dữ liệu, chỉ giữ lại các cột chính gồm userId, movieId, rating từ bảng ratings và movieId, title, genres từ bảng movies.
* Tách thể loại phim (genres): Cột genres chứa nhiều thể loại trong một chuỗi, được tách thành danh sách các thể loại để phục vụ cho việc xây dựng hệ thống gợi ý dựa trên thể loại hoặc phân tích thống kê.
* Xử lý tên phim: Một số tựa phim chứa năm phát hành trong ngoặc, ví dụ: "Toy Story (1995)", được tách riêng thành tên.
* Tạo ma trận tương tác người dùng – phim: Sử dụng dữ liệu từ bảng ratings, xây dựng ma trận người dùng – phim (user-item matrix), là nền tảng cho mô hình Collaborative Filtering hoạt động. Với user-based Collaborative Filtering, hệ thống tính tương đồng giữa các người dùng dựa trên đánh giá của họ.

## Xây dựng thuật toán gợi ý

Hệ thống gợi ý phim được triển khai dựa trên hai phương pháp chính: Content-Based Filtering và Collaborative Filtering. Mỗi phương pháp có cách tiếp cận riêng để tìm ra các bộ phim phù hợp với người dùng, dựa trên nội dung phim hoặc hành vi xem phim của những người dùng khác.

Sơ đồ dưới đây minh họa quy trình hoạt động của hệ thống gợi ý, từ việc tiếp nhận dữ liệu đầu vào, xử lý thông tin, đến khi hiển thị kết quả đề xuất cho người dùng. Flowchart này giúp trực quan hóa cách hệ thống phân tích và đưa ra danh sách phim phù hợp với từng người dùng.

A close-up of a black and white sign

AI-generated content may be incorrect.

Ảnh 6. Luồng dữ liệu của hệ thống

Trong các phần tiếp theo, bài viết sẽ trình bày chi tiết cách xây dựng từng mô hình gợi ý, bao gồm việc vector hóa dữ liệu phim, tính toán độ tương đồng, và đề xuất danh sách phim dựa trên thuật toán.

### Mô hình Content-based Filtering

* Bước 1: Tổng hợp nội dung đặc trưng

Trước khi xây dựng mô hình gợi ý, hệ thống cần xây dựng một biểu diễn chung của từng bộ phim dựa trên các thuộc tính quan trọng. Điều này được thực hiện bằng cách:

* Kết hợp nhiều nguồn thông tin: Dữ liệu ban đầu thường chứa mô tả phim (overview), thể loại (genres), từ khóa (keywords), diễn viên chính (cast), đạo diễn (crew). Các thành phần này được ghép thành một trường dữ liệu duy nhất gọi là tags, giúp hệ thống có cái nhìn tổng quan về nội dung phim.
* Chuẩn hóa dữ liệu văn bản: Chuyển tất cả văn bản về chữ thường, loại bỏ dấu câu và dư thừa để tạo một bộ từ chuẩn hóa.
* Xóa từ không quan trọng: Một số từ không mang giá trị nội dung, như "the", "and", "of" sẽ được loại bỏ để giữ lại các từ quan trọng liên quan đến nội dung phim.
* Bước 2. Vector hóa nội dung phim
* Sau khi có trường tags, hệ thống tiến hành chuyển đổi nội dung văn bản thành dạng số để có thể tính toán trên máy.
* Áp dụng kỹ thuật TF-IDF (Term Frequency-Inverse Document Frequency) để chuyển nội dung thành vector.
* TF-IDF giúp xác định mức độ quan trọng của từng từ trong tập dữ liệu, bằng cách đánh trọng số cao hơn cho các từ xuất hiện nhiều trong một phim nhưng ít xuất hiện trong toàn bộ kho phim.
* Vector hóa giúp hệ thống so sánh nội dung phim mà không cần phải xử lý trực tiếp chuỗi văn bản.
* Bước 3. Tính toán độ tương đồng
* Sau khi có vector hóa nội dung phim, hệ thống cần so sánh mức độ tương đồng giữa các phim để đưa ra đề xuất.
* Áp dụng công thức cosine similarity để đo mức độ giống nhau giữa hai phim.
* Công thức cosine similarity tính toán góc giữa hai vector, giúp xác định mức độ tương đồng mà không bị ảnh hưởng bởi độ dài của văn bản.
* Kết quả của phép đo cosine similarity là một ma trận tương đồng, trong đó mỗi bộ phim có danh sách các phim khác có nội dung gần giống nhất.
* Bước 4. Gợi ý phim
* Dựa trên ma trận tương đồng, hệ thống có thể trả về danh sách các phim liên quan khi người dùng nhập tên phim.
* Khi người dùng nhập tên một bộ phim, hệ thống lấy vector của phim đó, sau đó tìm các phim có điểm tương đồng cao nhất trong ma trận cosine similarity.
* Kết quả trả về là danh sách top N bộ phim có nội dung gần giống nhất, kèm theo hình ảnh poster để tăng trải nghiệm cho người dùng.

### Mô hình Collaborative Filtering

* Bước 1: Xây dựng ma trận người dùng – phim. Thuật toán bắt đầu bằng cách xây dựng một ma trận người dùng – phim, trong đó:
* Mỗi hàng tương ứng với một người dùng.
* Mỗi cột đại diện cho một bộ phim.
* Giá trị trong ô là điểm đánh giá mà người dùng đã chấm cho bộ phim tương ứng.
* Để đảm bảo chất lượng dữ liệu, hệ thống chỉ giữ lại những người dùng đã đánh giá đủ nhiều phim. Việc lọc này giúp:
  + Tránh các trường hợp người dùng có quá ít dữ liệu, gây nhiễu cho mô hình.
  + Nâng cao độ tin cậy khi tính toán tương đồng giữa người dùng.
* Các giá trị trống (phim chưa được người dùng xem hoặc đánh giá) được điền bằng 0, tạo thành một ma trận thưa (sparse matrix) đặc trưng cho dữ liệu đánh giá.
* Bước 2: Tính toán độ tương đồng giữa người dùng
* Hệ thống sử dụng cosine similarity để đo lường mức độ tương đồng giữa hai người dùng dựa trên cách họ đánh giá các bộ phim.
* Cosine similarity giúp xác định người dùng nào có hành vi đánh giá tương tự nhau, bất kể thang điểm tuyệt đối.
* Kết quả là một ma trận tương đồng giữa người dùng, trong đó mỗi phần tử thể hiện mức độ giống nhau giữa hai người dùng bất kỳ.
* Bước 3: Dự đoán điểm đánh giá cho phim chưa xem
* Với mỗi phim mà người dùng chưa đánh giá, hệ thống dự đoán điểm gợi ý dựa trên đánh giá của những người dùng tương đồng.
* Cách tính điểm dự đoán sử dụng công thức trung bình có trọng số, trong đó trọng số là mức độ tương đồng giúp hệ thống ước lượng độ yêu thích của người dùng đối với những bộ phim chưa từng xem.

A close-up of a word

AI-generated content may be incorrect.

* Bước 4: Gợi ý danh sách phim phù hợp
* Sau khi tính điểm dự đoán cho tất cả các phim chưa xem, hệ thống sắp xếp các phim theo điểm dự đoán giảm dần.
* Từ đó, chọn ra top N bộ phim có điểm dự đoán cao nhất để đề xuất cho người dùng.
* Gợi ý dựa hoàn toàn vào hành vi đánh giá của những người dùng có sở thích tương tự, không sử dụng đặc trưng nội dung phim.

## Ứng dụng hệ thống vào thực tế

### Mô tả hệ thống

Hệ thống được xây dựng dưới dạng một website đơn giản, cho phép người dùng:

* Tìm kiếm các bộ phim tương tự với một phim đã chọn (không cần đăng nhập).
* Nhận gợi ý phim cá nhân hóa khi đăng nhập (dựa trên lịch sử người dùng hoặc mô hình collaborative filtering).

Chức năng chính của hệ thống:

| Chức năng | Mô tả |
| --- | --- |
| Tìm kiếm phim tương tự | Gợi ý các bộ phim có nội dung gần giống với phim được nhập (dựa trên TF-IDF + cosine similarity). |
| Đăng nhập người dùng | Cho phép người dùng đăng nhập để nhận gợi ý cá nhân. |
| Gợi ý phim cá nhân hóa | Khi đã đăng nhập, hệ thống hiển thị các phim được đề xuất riêng theo lịch sử tương tác. |

Bảng 1. Mô tả chức năng chính của hệ thống

### Kiến trúc tổng quan

Hệ thống áp dụng mô hình kết hợp giữa collaborative filtering và content-based filtering. Sơ đồ dưới đây mô tả quy trình xử lý dữ liệu đầu vào và tạo ra kết quả gợi ý tối ưu:

A diagram of a model

AI-generated content may be incorrect.

Ảnh 7. Kiến trúc hệ thống

* Frontend: Giao diện người dùng được xây dựng bằng HTML/CSS.
* Backend: Xử lý truy vấn tìm kiếm và sinh gợi ý sử dụng Flask (Python), tích hợp các mô hình học máy (TF-IDF + cosine similarity, matrix factorization...).

Để có thể đảm bảo hiệu suất tốt khi xử lý dữ liệu và chạy thuật toán gợi ý, hệ thống yêu cầu cấu hình như sau: Bộ vi xử lý (CPU): Intel Core i5 hoặc tương đương. Bộ nhớ RAM: Tối thiểu 8GB để xử lý dữ liệu nhanh chóng. Bộ xử lý đồ họa (GPU): Nvidia 2GB (tùy chọn, giúp tăng tốc tính toán). Dung lượng lưu trữ: SSD 256GB để tăng tốc độ đọc/ghi dữ liệu. Hệ điều hành: Windows 10, Linux hoặc macOS.

### Đặc tả Use Case

* Use Case 1: Tìm kiếm phim tương tự

|  |  |
| --- | --- |
| Tên Use- Case | Tìm kiếm phim tương tự |
| Tác nhân | Người dùng chưa đăng nhập |
| Mô tả | Gợi ý danh sách phim có nội dung tương tự với phim mà người dùng đã tìm kiếm. |
| Điều kiện tiên quyết | Thiết bị của người dùng đã được kết nối internet  Người dùng truy cập trang web. Không cần đăng nhập. |
| Luồng chính | 1. Người dùng truy cập website. 2. Hệ thống hiển thị ô tìm kiếm phim. 3. Người dùng nhập tên phim cần tìm. 4. Hệ thống xử lý truy vấn và tìm phim phù hợp. 5. Hệ thống tính toán độ tương đồng nội dung (content-based). 6. Hệ thống hiển thị danh sách phim tương tự. |
| Điều kiện sau | Hệ thống hiển thị danh sách phim có nội dung tương tự |

Bảng 2. Use case tìm kiếm phim

* Use Case 2: Hiển thị phim được đề xuất

|  |  |
| --- | --- |
| Tên Use- Case | Hiển thị phim được đề xuất |
| Tác nhân | Người dùng đã đăng nhập |
| Mô tả | Đưa ra danh sách phim được cá nhân hóa theo mô hình collaborative filtering. |
| Điều kiện tiên quyết | Người dùng đã đăng nhập vào hệ thống.  Thiết bị của người dùng có kết nối internet |
| Luồng chính | 1. Người dùng đăng nhập vào hệ thống. 2. Hệ thống xác định ID người dùng. 3. Hệ thống truy xuất lịch sử hoặc thông tin liên quan. 4. Hệ thống tính toán gợi ý bằng mô hình collaborative filtering 5. Hệ thống hiển thị danh sách phim được đề xuất. |
| Điều kiện sau | Danh sách phim đề xuất được hiển thị cho người dùng đã đăng nhập. |

Bảng 3. Use case hiển thị phim được đề xuất

### Luồng hoạt động

Sơ đồ hoạt động dưới đây mô tả quy trình tương tác chính của người dùng với hệ thống gợi ý phim: Nếu chưa đăng nhập, người dùng có thể sử dụng chức năng tìm kiếm để nhập tên một bộ phim. Hệ thống sẽ xử lý và hiển thị danh sách các phim có nội dung tương tự. Nếu đã đăng nhập, hệ thống sẽ tự động hiển thị danh sách các bộ phim được cá nhân hóa dựa trên lịch sử đánh giá hoặc hành vi người dùng.

A screenshot of a diagram

AI-generated content may be incorrect.

Ảnh 8. Cách người dùng tương tác, hoạt động với hệ thống

### Hình ảnh hệ thống

A screenshot of a computer

AI-generated content may be incorrect.

Ảnh 9. Hình ảnh giao diện hệ thống với thuật toán Content- based Filtering

A screenshot of a computer

AI-generated content may be incorrect.

Ảnh 10. Hình ảnh giao diện hệ thống với thuật toán Collaborative Filtering

## Đánh giá kết quả và tác động

Hệ thống gợi ý phim được xây dựng và kiểm thử ở quy mô dữ liệu trung bình (~5.000 phim, ~600 người dùng). Dù chưa áp dụng các chỉ số đánh giá định lượng (như Precision, Recall, RMSE...), nhưng hiệu quả bước đầu có thể đánh giá qua các khía cạnh sau:

* Tính hợp lý của kết quả gợi ý: Các phim được đề xuất thường có nội dung tương đồng với phim đầu vào (về thể loại, từ khóa, diễn viên…), cho thấy mô hình content-based hoạt động phù hợp với mục tiêu đề ra.
* Trải nghiệm người dùng: Hệ thống cho phép người dùng không cần đăng nhập vẫn có thể tìm kiếm và nhận kết quả nhanh chóng. Với người dùng đăng nhập, danh sách gợi ý mang tính cá nhân hơn.
* Hiệu năng: Hệ thống phản hồi nhanh với tập dữ liệu thử nghiệm. Điều này đảm bảo trải nghiệm liền mạch cho người dùng cuối.
* Tính mở rộng: Cấu trúc hệ thống đơn giản, dễ mở rộng thêm dữ liệu, chức năng, hoặc nâng cấp mô hình đề xuất.

# Kết luận và định hướng trong tương lai

## Kết luận

Trong bài báo cáo này, nhóm đã xây dựng thành công một hệ thống gợi ý phim kết hợp giữa hai phương pháp phổ biến: Content-based Filtering và Collaborative Filtering.

* Với Content-based Filtering, hệ thống phân tích nội dung mô tả của từng bộ phim như thể loại, từ khóa, diễn viên, đạo diễn,... để gợi ý những bộ phim có nội dung tương đồng.
* Với Collaborative Filtering, hệ thống dựa trên hành vi đánh giá của người dùng để xác định các sở thích tương tự và đưa ra đề xuất phù hợp.

Việc kết hợp hai mô hình giúp hệ thống linh hoạt hơn, có thể phục vụ tốt cả người dùng mới (ít dữ liệu) và người dùng có lịch sử đánh giá rõ ràng. Trong quá trình triển khai, nhóm cũng đã xử lý dữ liệu từ các tập dữ liệu TMDB 5000 và MovieLens Small Latest, thực hiện tiền xử lý, vector hóa, tính toán độ tương đồng và xây dựng cơ chế gợi ý.

Kết quả thu được cho thấy hệ thống hoạt động ổn định, có khả năng đưa ra gợi ý hợp lý theo ngữ cảnh, góp phần cải thiện trải nghiệm người dùng trong việc tìm kiếm và lựa chọn phim.

Mặc dù hệ thống đã đạt được một số kết quả tích cực, vẫn còn tồn tại những hạn chế nhất định:

* Dữ liệu còn hạn chế: Tập dữ liệu được sử dụng mang tính chất thử nghiệm, chưa đủ lớn hoặc đa dạng để mô phỏng tốt hành vi người dùng trong thực tế.
* Vấn đề cold-start: Dù đã lọc người dùng có nhiều đánh giá để xây mô hình Collaborative Filtering, hệ thống vẫn chưa xử lý tốt các trường hợp người dùng hoàn toàn mới hoặc phim mới chưa có đánh giá.
* Gợi ý còn đơn chiều: Content-based Filtering thường chỉ gợi ý những phim rất giống với phim đã xem, dẫn đến thiếu tính đa dạng và khám phá.
* Không cập nhật theo thời gian thực: Hệ thống hiện tại chưa hỗ trợ học hoặc cập nhật theo phản hồi người dùng khi gợi ý được sử dụng trong môi trường thật.

## Định hướng trong tương lai

Dựa trên các hạn chế đã nêu, hệ thống có thể được cải thiện theo các hướng sau trong tương lai:

* Tích hợp mô hình lai (Hybrid Model): Kết hợp ưu điểm của cả hai mô hình để giảm ảnh hưởng của cold-start và tăng tính cá nhân hóa.
* Áp dụng các mô hình hiện đại hơn: Sử dụng các thuật toán nâng cao như Matrix Factorization, SVD++, hoặc mô hình học sâu (Deep Learning) để tăng độ chính xác và khả năng khái quát.
* Mở rộng dữ liệu và cá nhân hóa sâu hơn: Thu thập thêm thông tin hành vi người dùng như thời gian xem, lượt tìm kiếm, click, để tăng khả năng gợi ý sát với sở thích thật.
* Tối ưu hóa hiệu suất và xử lý thời gian thực: Xây dựng hệ thống hỗ trợ cập nhật mô hình liên tục hoặc gợi ý trong thời gian thực với tập dữ liệu lớn.

# Bảng phân công nhiệm vụ

|  |  |
| --- | --- |
| **Công việc** | **Người thực hiện** |
| Phân tích nghiệp vụ tổng thể | Nguyễn Khắc Lộc |
| Vẽ luồng hoạt động, đặc tả chức năng | Nguyễn Khắc Lộc |
| Viết báo cáo | Nguyễn Khắc Lộc |
| Lập trình web | Lê Huy Hoàng |
| Lập trình 2 thuật toán | Lê Huy Hoàng |
| Làm silde | Nguyễn Khôi Nguyên |
| Thuyết trình | Nguyễn Khôi Nguyên |

# Tài liệu tham khảo

1. [Xây dựng một hệ thống gợi ý phim đơn giản với Python](https://viblo.asia/p/xay-dung-mot-he-thong-goi-y-phim-don-gian-voi-python-eW65Ge1PZDO)
2. [http://www.ir.juit.ac.in:8080/jspui/bitstream/123456789/9988/1/Movie%20Recommendation%20System%20using%20Machine%20Learning.pdf](http://www.ir.juit.ac.in:8080/jspui/bitstream/123456789/9988/1/Movie%20Recommendation%20System%20using%20Machine%20Learning.pdf#:~:text=In%20this%20paper%2C%20to%20improve%20the%20quality%20of,than%20the%20pure%20approaches%20in%20three%20different%20datasets.)
3. [Machine Learning cơ bản](https://machinelearningcoban.com/2017/05/17/contentbasedrecommendersys/#:~:text=T%E1%BA%ADp%20h%E1%BB%A3p%20t%E1%BA%A5t%20c%E1%BA%A3%20c%C3%A1c%20ratings%2C%20bao%20g%E1%BB%93m,theo%20m%E1%BB%A9c%20%C4%91%E1%BB%99%20t%E1%BB%AB%200%20%C4%91%E1%BA%BFn%205%20sao.)
4. [Xây dựng Collaborative Filtering - Recommender System cơ bản](https://viblo.asia/p/xay-dung-collaborative-filtering-rs-recommender-system-co-ban-phan-3-Az45bMqolxY)
5. [Tìm hiểu về Content-based Filtering - Phương pháp gợi ý dựa theo nội dung (Phần 1)](https://viblo.asia/p/tim-hieu-ve-content-based-filtering-phuong-phap-goi-y-dua-theo-noi-dung-phan-1-V3m5WGBg5O7)
6. [TMDB 5000 Movie Dataset](https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata/data)
7. [Movie Lens Small Latest Dataset](https://www.kaggle.com/datasets/shubhammehta21/movie-lens-small-latest-dataset)
8. [Seminar - Movie Recommendation System - Hệ thống đề xuất phim - Phạm Như Khoa, Nguyễn Đình Kiên - YouTube](https://www.youtube.com/watch?v=-L6skyohFbI&t=821s)
9. [Understanding TF-IDF (Term Frequency-Inverse Document Frequency) - GeeksforGeeks](https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/)