**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**
**KHOA CÔNG NGHỆ THÔNG TIN 1**

**_____ o0o _____**


# BÀI TẬP LỚN NHẬP MÔN TRÍ TUỆ NHÂN TẠO

## Tên đề tài: Thuật toán lọc công tác ứng dụng trong bài toán gợi ý phim

**LỚP : N09**
**Số thứ tự nhóm: 16**

| | |
|---|---|
| **Trần Việt Anh** | **MSSV: B21DCCN162** |
| **Nguyễn Bá Hải Long** | **MSSV: B21DCAT119** |
| **Tô Quang Huy** | **MSSV: B21DCAT104** |
| **Lê Trần Hiếu** | **MSSV: B21DCAT088** |
| **Lưu Đức Hải** | **MSSV: B21DCAT081** |

**Giảng viên hướng dẫn: Ths. Vũ Hoài Thư**

**HÀ NỘI, 05/2023**

# LỜI CẢM ƠN

Lời đầu tiên, nhóm tác giả xin trân trọng cảm ơn giảng viên **Vũ Hoài Thư** - người đã trực tiếp chỉ bảo, hướng dẫn trong quá trình hoàn thành bài báo cáo này.
Mặc dù đã có những đầu tư nhất định trong quá trình làm bài song cũng khó có thể tránh khỏi những sai sót, kính mong nhận được ý kiến đóng góp của cô để bài báo cáo được hoàn thiện hơn.
**Xin chân thành cảm ơn!**

# MỤC LỤC

# DANH MỤC HÌNH VẼ

# DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

| Thuật ngữ | Ý nghĩa |
|---|---|
| RMSE | Root Mean Square Error - Căn bậc hai mức trung bình sai số bình phương. Được sử dụng trong đánh giá mức độ hiệu quả của mô hình |

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

## 1.1 Đặt vấn đề

Trong thời đại số hóa ngày nay, việc xem phim trực tuyến trở nên phổ biến hơn bao giờ hết. Người dùng thường xem phim trên các nền tảng streaming như Netflix, Amazon Prime và Disney+. Tuy nhiên, một thách thức lớn mà họ thường gặp phải là không biết chọn bộ phim nào để xem tiếp theo.

Hiện nay, các nền tảng streaming chưa thực sự tận dụng được dữ liệu về lịch sử xem phim và sở thích của người dùng để cung cấp các gợi ý phim cá nhân hóa. Việc này không chỉ làm giảm trải nghiệm xem phim mà còn ảnh hưởng đến sự hài lòng và trung thành của người dùng với nền tảng.

Vấn đề này rất cấp thiết và quan trọng đối với các nền tảng streaming. Việc cung cấp các gợi ý phim cá nhân hóa không chỉ tăng cường trải nghiệm xem phim của người dùng mà còn có thể tăng doanh số bán hàng và độ trung thành của họ với nền tảng.

Nếu vấn đề này được giải quyết thành công, các nền tảng streaming sẽ có thể tận dụng tối đa dữ liệu xem phim của người dùng để cung cấp các gợi ý phim cá nhân hóa, từ đó cải thiện trải nghiệm xem phim và tăng doanh số bán hàng. Ngoài ra, phương pháp này cũng có thể áp dụng vào các lĩnh vực khác như các trang web tin tức, ứng dụng âm nhạc trực tuyến, mua hàng trực tuyến và các nền tảng thương mại điện tử khác để cung cấp các gợi ý cá nhân hóa cho người dùng.

## 1.2 Mục tiêu và phạm vi đề tài

Mục tiêu của đề tài là phát triển và triển khai một hệ thống gợi ý phim dựa trên phương pháp lọc cộng tác, nhằm cải thiện trải nghiệm xem phim của người dùng trên các nền tảng streaming trực tuyến.

Các nghiên cứu hiện nay về hệ thống gợi ý phim tập trung vào sử dụng các phương pháp máy học và học sâu để dự đoán sở thích của người dùng và gợi ý các phim tương tự. Các hệ thống này thường sử dụng các phương pháp như lọc cộng tác, lọc dựa trên nội dung và kết hợp giữa hai phương pháp này để cung cấp các gợi ý phim cá nhân hóa.

Các phương pháp và hệ thống gợi ý phim hiện nay thường gặp phải các hạn chế như:

- Thiếu khả năng cá nhân hóa đầy đủ.
- Không đủ chính xác trong việc gợi ý các phim mới và mang tính đa dạng cao.

- Không thể tự động thích nghi với sở thích mới của người dùng.

Nhóm tác giả sẽ giải quyết vấn đề trên bằng cách phát triển một hệ thống gợi ý phim dựa trên phương pháp lọc cộng tác, nhằm cải thiện tính cá nhân hóa và độ chính xác của các gợi ý phim cho người dùng. Đồng thời, hệ thống cũng sẽ được thiết kế để tự động thích nghi với sở thích mới của người dùng để cung cấp trải nghiệm xem phim ngày càng tốt hơn.

### 1.3 Định hướng giải pháp

Để giải quyết vấn đề, nhóm tác giả sẽ tập trung vào việc phát triển một hệ thống gợi ý sản phẩm dựa trên phương pháp lọc cộng tác. Phương pháp này sẽ sử dụng dữ liệu lịch sử hoạt động của người dùng, bao gồm lịch sử xem, đánh giá, thời gian tương tác và thói quen sử dụng, để tạo ra các gợi ý sản phẩm cá nhân hóa. Đồng thời, nhóm tác giả cũng sẽ xem xét việc kết hợp lọc cộng tác với lọc dựa trên nội dung để tăng cường độ đa dạng và tính phong phú của các gợi ý. Nhóm tác giả lựa chọn phương pháp lọc cộng tác là do nó dựa trên hành vi của cả người dùng và sản phẩm để tạo ra các gợi ý cá nhân hóa. Phương pháp này tự nhiên và hiệu quả trong việc đề xuất các sản phẩm dựa trên sự tương đồng giữa người dùng và giữa các sản phẩm. Điều này giúp cải thiện tính chính xác và tính cá nhân hóa của hệ thống gợi ý sản phẩm, làm tăng sự hài lòng và trung thành từ phía người dùng.

Hệ thống gợi ý sản phẩm sẽ bao gồm các bước chính sau:

- Thu thập dữ liệu: Thu thập và tiền xử lý dữ liệu từ các nguồn khác nhau như lịch sử xem, đánh giá và thói quen sử dụng của người dùng.
- Xây dựng ma trận Tiện ích: Xây dựng ma trận tiện ích để biểu diễn mối quan hệ giữa người dùng và sản phẩm.
- Tính độ tương đồng: Tính toán độ tương đồng giữa người dùng hoặc sản phẩm dựa trên ma trận tiện ích đã xây dựng.
- Dự đoán sở thích: Dựa trên độ tương đồng, dự đoán sở thích của người dùng cho các sản phẩm chưa được xem hoặc đánh giá.
- Tạo gợi ý sản phẩm: Tạo ra danh sách các gợi ý sản phẩm cá nhân hóa cho mỗi người dùng dựa trên dự đoán sở thích.

Đóng góp chính của bài tập lớn của nhóm tác giả là phát triển và triển khai một hệ thống gợi ý sản phẩm sử dụng phương pháp lọc cộng tác. Qua quá trình nghiên cứu và thử nghiệm, nhóm tác giả đã đạt được một số kết quả quan trọng như sau:

- Cải thiện tính chính xác: Bằng cách áp dụng phương pháp lọc cộng tác và kết hợp với lọc dựa trên nội dung, hệ thống gợi ý sản phẩm đã đạt được một mức độ chính xác cao hơn so với các phương pháp gợi ý truyền thống. Điều này

giúp tăng cường sự hài lòng từ phía người dùng và cải thiện trải nghiệm của họ khi sử dụng nền tảng.

- Tính cá nhân hóa tăng cường: Hệ thống gợi ý đã được tối ưu hóa để tạo ra các gợi ý sản phẩm cá nhân hóa dựa trên sở thích và hành vi của từng người dùng cụ thể. Điều này giúp tạo ra một trải nghiệm người dùng cá nhân hóa, giúp họ khám phá và tận hưởng nội dung một cách tối ưu.
- Tạo ra một hệ thống tối ưu: Hệ thống gợi ý sản phẩm mới đã đem lại một trải nghiệm tốt hơn cho người dùng, từ việc cung cấp các gợi ý chính xác đến việc tạo ra một môi trường tương tác và khám phá sản phẩm trên nền tảng.

## 1.4 Bố cục bài tập lớn

Phần còn lại của báo cáo bài tập lớn này được tổ chức như sau:

Chương 2 sẽ giới thiệu về bài toán lọc cộng tác và trình bày về cơ sở lý thuyết của bài toán. Bài toán lọc cộng tác là một trong những kỹ thuật quan trọng nhất trong việc xây dựng hệ thống đề xuất thông minh. Phương pháp này tập trung vào việc dự đoán sở thích của người dùng dựa trên sự tương tự giữa họ và những người dùng khác, hoặc giữa các mục dựa trên sự tương tự của chúng. Dữ liệu được sử dụng thường là ma trận phản ứng giữa người dùng và mục, trong đó mỗi phần tử là xếp hạng hoặc phản ứng một người dùng cho một mục cụ thể. Thuật toán lọc cộng tác có hai dạng chính: dựa trên người dùng và dựa trên mục. Dựa trên người dùng tìm kiếm những người dùng có sở thích tương tự và dựa trên xếp hạng của họ để đưa ra các đề xuất. Trong khi đó, dựa trên mục tìm kiếm các mục có sự tương tự và dựa trên xếp hạng của người dùng để đưa ra đề xuất. Lọc cộng tác hoạt động tốt trong nhiều trường hợp, đặc biệt là khi không có nhiều dữ liệu về mục hoặc người dùng, và không cần phải biết trước các tính năng cụ thể về mục hoặc người dùng. Tuy nhiên, nó cũng gặp phải một số thách thức như vấn đề khởi động nguội cho các mục mới và sự thưa thớt dữ liệu. Kết luận, bài toán lọc cộng tác là một công cụ mạnh mẽ để xây dựng hệ thống đề xuất, và việc hiểu rõ về cách hoạt động của nó cũng như những ưu và nhược điểm của nó có thể giúp chọn lựa phương pháp phù hợp nhất cho một bài toán cụ thể.

Chương 3 sẽ trình bày về thực nghiệm của bài toán. Trong thực nghiệm của bài toán lọc cộng tác, dữ liệu về sở thích của người dùng và phản ứng của họ đối với các mục được sử dụng để xây dựng một hệ thống đề xuất. Qua việc phân tích và so sánh các mẫu tương tự giữa người dùng hoặc giữa các mục, thuật toán dự đoán xếp hạng cho các mục mà người dùng chưa tương tác. Thực nghiệm này thường đo lường độ chính xác của các dự đoán bằng các phép đánh giá như RMSE trên tập dữ liệu kiểm tra. Kết quả từ thực nghiệm cung cấp thông tin danh sách similarity

của bộ phim ta cần tính dự đoán, danh sách kết quả dự đoán của các bộ phim, danh sách các bộ phim thuộc top 3 để gợi ý cho người xem, giá trị RMSE.

Chương 4 trình bày về phần kết luận của bài toán. Từ việc thử nghiệm và đánh giá các thuật toán trên tập dữ liệu, nhóm tác giả có thể nhận thấy tính hiệu quả của lọc cộng tác trong việc đưa ra các đề xuất phù hợp cho người dùng. Kết quả của các độ đo như RMSE giúp đánh giá chính xác mức độ sai lệch giữa xếp hạng dự đoán và xếp hạng thực tế, từ đó cung cấp thông tin cần thiết để điều chỉnh và tối ưu hóa thuật toán. Mặt khác, các thách thức như khởi động nguội, sự thưa thớt dữ liệu và mở rộng quy mô cũng được đưa ra để người ta hiểu rõ hơn về những hạn chế của phương pháp này. Tuy nhiên, với sự phát triển của công nghệ và các phương pháp tiếp cận tiên tiến, lọc cộng tác vẫn là một công cụ mạnh mẽ trong hệ thống đề xuất. Điều quan trọng là phải kết hợp các phương pháp và thuật toán khác nhau để tối ưu hóa hiệu suất của hệ thống. Trong khi lọc cộng tác tập trung vào sự tương tác giữa người dùng và các mục, các phương pháp khác như lọc dựa trên nội dung có thể được sử dụng để bổ sung và cải thiện chất lượng của đề xuất. Điều này cho thấy rằng, mặc dù lọc cộng tác không phải là giải pháp hoàn hảo, nhưng khi được kết hợp và tinh chỉnh một cách linh hoạt, nó có thể cung cấp những đề xuất hữu ích và phù hợp cho người dùng, giúp cải thiện trải nghiệm người dùng và tăng cường sự tương tác trên các nền tảng trực tuyến.

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## 2.1 Giới thiệu về bài toán lọc cộng tác

Lọc cộng tác là kỹ thuật phổ biến nhất được sử dụng khi xây dựng hệ thống đề xuất thông minh có thể học cách đưa ra đề xuất tốt hơn khi thu thập được nhiều thông tin hơn về người dùng.

Hầu hết các trang web như Amazon, YouTube và Netflix đều sử dụng tính năng lọc cộng tác như một phần của hệ thống đề xuất phức tạp của họ. Kỹ thuật này được sử dụng để xây dựng những công cụ đề xuất đưa ra đề xuất cho người dùng dựa trên lượt thích và không thích của những người dùng tương tự.

## 2.2 Lọc cộng tác là gì?

Lọc cộng tác là một kỹ thuật có thể lọc ra các mục mà người dùng có thể thích dựa trên phản ứng của những người dùng tương tự.

Nó hoạt động bằng cách tìm kiếm một nhóm lớn người và tìm một nhóm người dùng nhỏ hơn có sở thích tương tự như một người dùng cụ thể. Nó xem xét các mục họ thích và kết hợp chúng để tạo ra danh sách gợi ý được xếp hạng.

Có nhiều cách để quyết định những người dùng nào giống nhau và kết hợp các lựa chọn của họ để tạo ra danh sách đề xuất.

## 2.3 Bộ dữ liệu

Để thử nghiệm các thuật toán đề xuất, cần dữ liệu chứa một tập hợp các mục và một tập hợp người dùng đã phản ứng với một số mục.

Phản ứng có thể rõ ràng (đánh giá theo thang điểm từ 1 đến 5, thích hoặc không thích) hoặc ngầm định (xem một mục, thêm nó vào danh sách mong muốn, thời gian dành cho một bài viết).

Khi làm việc với những dữ liệu đó, hầu như nó ở dạng ma trận bao gồm các phản ứng do một nhóm người dùng đưa ra đối với một số mục trong một tập hợp các mục. Mỗi hàng sẽ chứa xếp hạng do người dùng đưa ra và mỗi cột sẽ chứa xếp hạng mà một mục nhận được. Một ma trận có năm người dùng và năm mục có thể trông như thế này:

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ | $i_5$ |
|---|---|---|---|---|---|
| $u_1$ | 5 | | 4 | 1 | |
| $u_2$ | | 3 | | 3 | |
| $u_3$ | | 2 | 4 | 4 | 1 |
| $u_4$ | 4 | 4 | 5 | | |
| $u_5$ | 2 | 4 | | 5 | 2 |

**Hình 2.1:** Ma trận đánh giá giữa người dùng và mục

Ma trận hiển thị năm người dùng đã xếp hạng một số mục theo thang điểm từ 1 đến 5. Ví dụ: người dùng đầu tiên đã xếp hạng 4 cho mục thứ ba.

Trong hầu hết các trường hợp, các ô trong ma trận đều trống vì người dùng chỉ xếp hạng một vài mục. Rất khó có khả năng mọi người dùng đều xếp hạng hoặc phản ứng với mọi mặt hàng có sẵn. Một ma trận có hầu hết các ô trống được gọi là thưa thớt, và ngược lại (ma trận hầu hết được lấp đầy) được gọi là dày đặc.

Cách tốt nhất để bắt đầu là bộ dữ liệu MovieLens do GroupLens Research thu thập. Đặc biệt, tập dữ liệu MovieLens 100k là tập dữ liệu điểm chuẩn ổn định với 100.000 xếp hạng được đưa ra bởi 943 người dùng cho 1682 phim, trong đó mỗi người dùng đã xếp hạng ít nhất 20 phim.

Tập dữ liệu này bao gồm nhiều tệp chứa thông tin về phim, người dùng và xếp hạng do người dùng đưa ra đối với phim họ đã xem. Những điều được quan tâm là như sau:

u.item: Danh sách phim.

u.data: Danh sách xếp hạng do người dùng đưa ra.

Tệp u.data chứa xếp hạng là danh sách ID người dùng, ID mục, xếp hạng và dấu thời gian được phân tách bằng tab. Một vài dòng đầu tiên của tập tin trông như thế này:

| user_id | item_id | rating | timestamp |
|---|---|---|---|
| 196 | 242 | 3 | 881250949 |
| 186 | 302 | 3 | 891717742 |
| 22 | 377 | 1 | 878887116 |
| 244 | 51 | 2 | 880606923 |
| 166 | 346 | 1 | 886397596 |

**Hình 2.2:** Dữ liệu đánh giá giữa người dùng và mục trong CSDL

Như được hiển thị ở trên, tệp cho biết xếp hạng mà người dùng đã đưa ra cho một bộ phim cụ thể. Tệp này chứa 100.000 xếp hạng như vậy, sẽ được sử dụng để dự đoán xếp hạng của những bộ phim mà người dùng không xem.

## 2.4 Các bước liên quan đến lọc cộng tác

Để xây dựng một hệ thống có thể tự động giới thiệu các mặt hàng cho người dùng dựa trên sở thích của những người dùng khác, bước đầu tiên là tìm những người dùng hoặc mặt hàng tương tự. Bước thứ hai là dự đoán xếp hạng của các mục chưa được người dùng xếp hạng.

Làm thế nào để xác định những người dùng hoặc mục nào giống nhau? Khi biết những người dùng nào tương tự nhau, làm cách nào để xác định xếp hạng mà người dùng sẽ đưa ra cho một mặt hàng dựa trên xếp hạng của những người dùng tương tự? Làm thế nào để đo lường độ chính xác của xếp hạng?

Hai câu hỏi đầu tiên không có câu trả lời duy nhất. Lọc cộng tác là một nhóm thuật toán trong đó có nhiều cách để tìm người dùng hoặc mục tương tự và nhiều cách để tính xếp hạng dựa trên xếp hạng của những người dùng tương tự.

Một điều quan trọng cần lưu ý là trong cách tiếp cận hoàn toàn dựa trên lọc cộng tác, độ tương tự không được tính toán bằng cách sử dụng các yếu tố như độ tuổi của người dùng, thể loại phim hoặc bất kỳ dữ liệu nào khác về người dùng hoặc mục. Nó chỉ được tính toán dựa trên xếp hạng (rõ ràng hoặc ngầm định) mà người dùng đưa ra cho một mục. Ví dụ: hai người dùng có thể được coi là giống nhau nếu họ đưa ra cùng xếp hạng cho mười bộ phim mặc dù có sự khác biệt lớn về tuổi tác của họ.

Câu hỏi thứ ba về cách đo lường độ chính xác trong dự đoán cũng có nhiều câu trả lời, bao gồm các kỹ thuật tính toán lỗi có thể được sử dụng ở nhiều nơi chứ không chỉ những lời khuyên dựa trên lọc cộng tác.

Một trong những phương pháp để đo lường độ chính xác của kết quả là lỗi bình phương trung bình gốc (RMSE), trong đó dự đoán xếp hạng cho tập dữ liệu thử nghiệm gồm các cặp mục người dùng có giá trị xếp hạng đã biết. Sự khác biệt giữa giá trị đã biết và giá trị dự đoán sẽ là sai số. Bình phương tất cả các giá trị lỗi cho tập kiểm tra, tìm giá trị trung bình (hoặc giá trị trung bình), sau đó lấy căn bậc hai của giá trị trung bình đó để nhận RMSE.

Một số liệu khác để đo độ chính xác là Sai số tuyệt đối trung bình (MAE), trong đó tìm mức độ sai số bằng cách tìm giá trị tuyệt đối của nó và sau đó lấy trung bình của tất cả các giá trị lỗi.

## 2.5 Các loại thuật toán trong lọc cộng tác

### 2.5.1 Lọc cộng tác dựa trên người dùng (User-based Collaborative Filtering)

Thuật toán đầu tiên là lọc cộng tác dựa trên người dùng, trong đó các kỹ thuật thống kê được áp dụng cho toàn bộ tập dữ liệu để tính toán dự đoán. Để tìm xếp hạng R mà người dùng U sẽ dành cho mặt hàng I, cách tiếp cận bao gồm: Tìm những người dùng tương tự đã đánh giá sản phẩm rồi tính xếp hạng R dựa trên xếp hạng của người dùng tìm thấy ở bước trước.

Để hiểu khái niệm về sự tương đồng, trước tiên hãy tạo một tập dữ liệu đơn giản. Dữ liệu bao gồm bốn người dùng A, B, C và D, mỗi người xếp hạng hai bộ phim. Xếp hạng được lưu trữ trong danh sách và mỗi danh sách chứa hai số cho biết xếp hạng của từng phim:

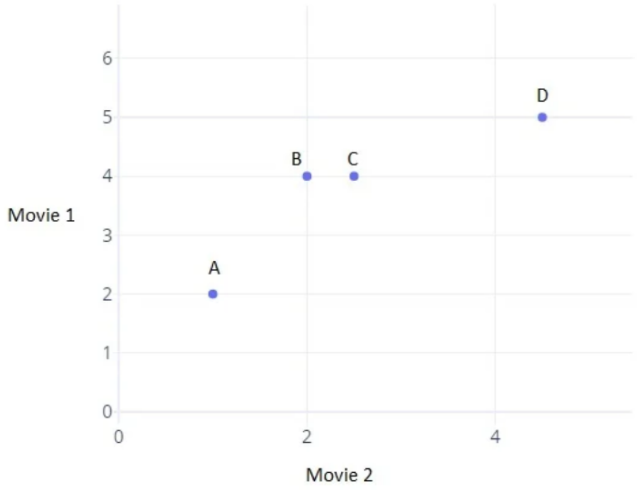
Xếp hạng của A là [1.0, 2.0].

Xếp hạng của B là [2.0, 4.0].

Xếp hạng của C là [2,5, 4,0].

Xếp hạng của D là [4,5, 5,0].

Để bắt đầu bằng manh mối trực quan, hãy vẽ biểu đồ xếp hạng của hai bộ phim do người dùng đưa ra và tìm kiếm một mẫu. Biểu đồ trông như thế này:




**Hình 2.3:** Biểu đồ xếp hạng 2 bộ phim mà các người dùng đưa ra

Trong biểu đồ trên, mỗi điểm đại diện cho một người dùng và được thể hiện dựa trên xếp hạng mà họ đưa ra cho hai bộ phim.

Khoảng cách giữa các điểm có vẻ là một cách hay để ước tính độ tương tự. Ta có thể tìm khoảng cách bằng cách sử dụng công thức tính khoảng cách Euclide giữa hai điểm. Ta có thể sử dụng chức năng có sẵn trong scipy như trong chương trình sau:

```
Python

>>> from scipy import spatial

>>> a = [1, 2]
>>> b = [2, 4]
>>> c = [2.5, 4]
>>> d = [4.5, 5]

>>> spatial.distance.euclidean(c, a)
2.5
>>> spatial.distance.euclidean(c, b)
0.5
>>> spatial.distance.euclidean(c, d)
2.23606797749979
```

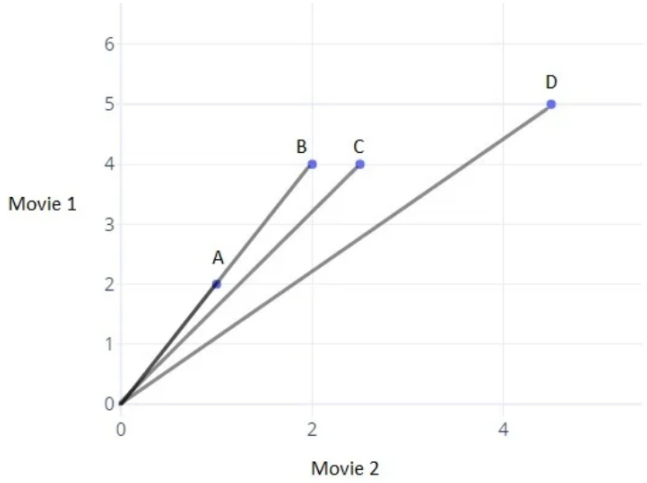
**Hình 2.4:** Chương trình tính khoảng cách euclide giữa 2 điểm

Như được hiển thị ở trên, ta có thể sử dụng scipy.spatial.distance.euclidean để tính khoảng cách giữa hai điểm. Sử dụng nó để tính khoảng cách giữa xếp hạng của A, B và D với xếp hạng của C cho chúng ta thấy rằng xét về khoảng cách, xếp hạng của C gần nhất với xếp hạng của B.

Có thể thấy người dùng C gần gũi nhất với B ngay cả khi nhìn vào biểu đồ. Nhưng chỉ trong A và D thì C thân với ai hơn?

Có thể nói C gần D hơn về khoảng cách. Nhưng nhìn vào bảng xếp hạng, có vẻ như lựa chọn của C sẽ phù hợp với lựa chọn của A hơn D vì cả A và C đều thích bộ phim thứ hai gần gấp đôi so với bộ phim đầu tiên, nhưng D thích cả hai bộ phim. bằng nhau.

Vì vậy, có thể sử dụng cái gì để xác định những mô hình mà khoảng cách Euclide không thể làm được? Góc giữa các đường nối các điểm với gốc có thể được sử dụng để đưa ra quyết định không? Có thể quan sát góc giữa các đường nối gốc của đồ thị với các điểm tương ứng như hình:




**Hình 2.5:** Biểu đồ xếp hạng 2 bộ phim mà các người dùng đưa ra sau khi thêm đoạn nối

Biểu đồ hiển thị bốn đường nối mỗi điểm với điểm gốc. Hai đường thẳng A và B trùng nhau nên góc giữa chúng bằng 0.

Có thể cân nhắc rằng, nếu góc giữa các đường tăng lên thì độ tương tự sẽ giảm và nếu góc bằng 0 thì người dùng rất giống nhau.

Để tính toán độ tương tự bằng cách sử dụng góc, cần một hàm trả về độ tương tự cao hơn hoặc khoảng cách nhỏ hơn cho góc thấp hơn và độ tương tự thấp hơn hoặc khoảng cách lớn hơn cho góc cao hơn. Cosin của một góc là hàm giảm từ 1 đến -1 khi góc tăng từ 0 lên 180.

Có thể sử dụng cosin của góc để tìm điểm tương đồng giữa hai người dùng. Góc càng cao thì cosin càng thấp và do đó độ tương đồng của người dùng sẽ càng thấp.

Độ tương tự cosine được tính bằng công thức sau:

$$Sim(\vec{q},\vec{d}) = \frac{\vec{q}.\vec{d}}{|\vec{q}||\vec{d}|} = \frac{\sum_{k=1}^{t} w_{qk} \times w_{dk}}{\sqrt{\sum_{k=1}^{t} (w_{qk})^2} \cdot \sqrt{\sum_{k=1}^{t} (w_{dk})^2}}$$

**Hình 2.6:** Công thức tính độ tương tự cosin

Lưu ý rằng người dùng A và B được coi là hoàn toàn giống nhau về số liệu độ tương tự cosine mặc dù có xếp hạng khác nhau. Đây thực sự là một điều thường xảy ra trong thế giới thực và những người dùng như người dùng A là những người mà có thể gọi là những người đánh giá khó tính. Một ví dụ là một nhà phê bình

phim luôn đưa ra xếp hạng thấp hơn mức trung bình, nhưng thứ hạng của các mục trong danh sách của họ sẽ tương tự như những người xếp hạng trung bình như B.

Để tính đến các sở thích của từng người dùng như vậy, cần phải đưa tất cả người dùng về cùng một cấp độ bằng cách loại bỏ những thành kiến của họ. Có thể thực hiện việc này bằng cách trừ đi xếp hạng trung bình do người dùng đó đưa ra cho tất cả các mục từ mỗi mục do người dùng đó xếp hạng. Đây là giao diện của nó:

Đối với người dùng A, vectơ xếp hạng [1, 2] có giá trị trung bình là 1,5. Trừ 1,5 cho mỗi xếp hạng sẽ cho vectơ [-0,5, 0,5].

Đối với người dùng B, vectơ xếp hạng [2, 4] có giá trị trung bình là 3. Trừ 3 từ mỗi xếp hạng sẽ cho vectơ [-1, 1].

Bằng cách này, ta đã thay đổi giá trị xếp hạng trung bình do mọi người dùng đưa ra thành 0. Làm tương tự cho người dùng C và D sẽ thấy rằng xếp hạng hiện được điều chỉnh để đưa ra mức trung bình là 0 cho tất cả người dùng , điều này đưa tất cả họ đến cùng một cấp độ và loại bỏ những thành kiến của họ.

Cosin của góc giữa các vectơ đã điều chỉnh được gọi là cosin tâm. Cách tiếp cận này thường được sử dụng khi có nhiều giá trị bị thiếu trong vectơ và cần đặt một giá trị chung để lấp đầy các giá trị còn thiếu.

Việc điền các giá trị còn thiếu vào ma trận xếp hạng bằng một giá trị ngẫu nhiên có thể dẫn đến kết quả không chính xác. Một lựa chọn tốt để điền vào các giá trị còn thiếu có thể là xếp hạng trung bình của mỗi người dùng, nhưng mức trung bình ban đầu của người dùng A và B lần lượt là 1,5 và 3, đồng thời điền tất cả các giá trị trống của A bằng 1,5 và của B bằng 3 sẽ làm cho họ trở thành những người dùng khác nhau.

Nhưng sau khi điều chỉnh các giá trị, mức trung bình ở giữa của cả hai người dùng là 0, điều này cho phép nắm bắt ý tưởng về mục ở trên hoặc dưới mức trung bình chính xác hơn cho cả hai người dùng với tất cả các giá trị bị thiếu trong vectơ của cả hai người dùng có cùng giá trị 0.

Khoảng cách Euclide và độ tương tự cosine là một số phương pháp mà có thể sử dụng để tìm những người dùng tương tự nhau và thậm chí cả các mục tương tự nhau.

Lưu ý: Công thức tính cosin trung tâm giống với công thức tính hệ số tương quan Pearson. Nhiều tài nguyên và thư viện về người giới thiệu đề cập đến việc triển khai cosin trung tâm dưới dạng Tương quan Pearson.

Sau khi xác định danh sách người dùng tương tự như người dùng U, cần tính xếp

hạng R mà U sẽ đưa ra cho một mục I nhất định. Một lần nữa, giống như độ tương tự, có thể thực hiện việc này theo nhiều cách.

Có thể dự đoán rằng xếp hạng R của người dùng cho một mặt hàng I sẽ gần bằng mức trung bình của các xếp hạng mà 5 hoặc 10 người dùng hàng đầu giống với U nhất đưa ra cho I. Công thức toán học cho xếp hạng trung bình do n người dùng đưa ra sẽ xem xét như thế này:

$$R_U = (\sum_{u=1}^{n} R_u)/n$$

**Hình 2.7:** Công thức tính xếp hạng trung bình dựa trên n người dùng tương tự

Công thức này cho thấy xếp hạng trung bình do n người dùng tương tự đưa ra bằng tổng xếp hạng do họ đưa ra chia cho số lượng người dùng tương tự, là n.

Sẽ có những tình huống trong đó n người dùng tương tự mà không giống với người dùng mục tiêu U. 3 người dùng hàng đầu trong số họ có thể rất giống nhau và những người còn lại có thể không giống U như 3 người dùng hàng đầu, có thể xem xét một cách tiếp cận trong đó xếp hạng của người dùng giống nhau nhất quan trọng hơn người dùng giống nhau thứ hai, v.v. Trung bình có trọng số có thể giúp chúng ta đạt được điều đó.

Theo cách tiếp cận trung bình có trọng số, nhân mỗi xếp hạng với một hệ số tương tự (hệ số này cho biết mức độ giống nhau của người dùng). Bằng cách nhân với hệ số tương tự, thêm trọng số vào xếp hạng. Trọng lượng càng nặng thì đánh giá càng quan trọng.

Hệ số tương tự, đóng vai trò như trọng số, phải là nghịch đảo của khoảng cách được thảo luận ở trên vì khoảng cách càng nhỏ thì độ tương tự càng cao.

Với hệ số tương tự S cho mỗi người dùng tương tự với người dùng mục tiêu U, có thể tính trung bình có trọng số bằng công thức sau:

$$R_U = (\sum_{u=1}^{n} R_u * S_u)/(\sum_{u=1}^{n} S_u)$$

**Hình 2.8:** Công thức tính xếp hạng trung bình có trọng số dựa trên n người dùng tương tự

Trong công thức trên, mỗi xếp hạng được nhân với hệ số tương tự của người dùng đã đưa ra xếp hạng. Xếp hạng dự đoán cuối cùng của người dùng U sẽ bằng tổng xếp hạng có trọng số chia cho tổng trọng số.

Lưu ý: Tại sao tổng xếp hạng có trọng số lại được chia cho tổng các trọng số

chứ không phải cho n? Hãy xem xét điều này: trong công thức trung bình trước đó, trong đó chia cho n, giá trị của trọng số là 1. Mẫu số luôn là tổng của các trọng số khi tìm giá trị trung bình và trong trường hợp trung bình thường, trọng số bằng 1 có nghĩa là mẫu số sẽ bằng n. Với mức trung bình có trọng số, cần cân nhắc nhiều hơn đến xếp hạng của những người dùng tương tự theo thứ tự tương đồng của họ.

Bây giờ, khi đã biết cách tìm những người dùng tương tự và cách tính xếp hạng dựa trên xếp hạng của họ, ta có thể phát triển một biến thể của lọc cộng tác trong đó ta dự đoán xếp hạng bằng cách tìm các mục tương tự nhau thay vì người dùng và tính toán xếp hạng.

### 2.5.2 Lọc cộng tác dựa trên mục (Item-based Collaborative Filtering)

Kỹ thuật trong các ví dụ được giải thích ở trên, trong đó ma trận xếp hạng được sử dụng để tìm những người dùng tương tự dựa trên xếp hạng mà họ đưa ra, được gọi là lọc cộng tác dựa trên người dùng hoặc người dùng - người dùng. Nếu sử dụng ma trận xếp hạng để tìm các mục tương tự dựa trên xếp hạng do người dùng đưa ra thì phương pháp này được gọi là lọc cộng tác dựa trên mục hoặc mục - mục. Hai cách tiếp cận này khá giống nhau về mặt toán học, nhưng có sự khác biệt về mặt khái niệm giữa hai cách tiếp cận này. Đây là cách so sánh cả hai:

- Dựa trên người dùng: Đối với người dùng U, với một tập hợp người dùng tương tự được xác định dựa trên vectơ xếp hạng bao gồm các xếp hạng mục nhất định, xếp hạng cho một mục I, chưa được xếp hạng, được tìm thấy bằng cách chọn ra N người dùng từ điểm tương đồng danh sách những người đã xếp hạng mục I và tính xếp hạng dựa trên N xếp hạng này.
- Dựa trên mục: Đối với một mục I, với một tập hợp các mục tương tự được xác định dựa trên vectơ xếp hạng bao gồm xếp hạng của người dùng nhận được, xếp hạng của người dùng U, người chưa xếp hạng nó, được tìm thấy bằng cách chọn ra N mục từ điểm giống nhau danh sách đã được U xếp hạng và tính xếp hạng dựa trên N xếp hạng này.

Lọc cộng tác dựa trên vật phẩm được phát triển bởi Amazon. Trong một hệ thống có nhiều người dùng hơn mục, việc lọc dựa trên mục sẽ nhanh hơn và ổn định hơn so với lọc dựa trên người dùng. Nó hiệu quả vì thông thường, xếp hạng trung bình mà một mặt hàng nhận được không thay đổi nhanh như xếp hạng trung bình mà người dùng đưa ra cho các mặt hàng khác nhau. Nó cũng được biết là hoạt động tốt hơn phương pháp dựa trên người dùng khi ma trận xếp hạng thưa thớt.

Mặc dù, cách tiếp cận dựa trên mục hoạt động kém đối với các tập dữ liệu có các mục liên quan đến duyệt web hoặc giải trí như MovieLens, trong đó các đề xuất mà nó đưa ra dường như rất rõ ràng đối với người dùng mục tiêu. Những bộ

dữ liệu như vậy sẽ mang lại kết quả tốt hơn với các kỹ thuật phân tích nhân tử ma trận trong phần tiếp theo hoặc với các công cụ đề xuất kết hợp cũng tính đến nội dung của dữ liệu như thể loại bằng cách sử dụng tính năng lọc dựa trên nội dung.

## 2.6 Khi nào có thể sử dụng lọc cộng tác?

Lọc cộng tác hoạt động xung quanh các tương tác mà người dùng có với các mục. Những tương tác này có thể giúp tìm ra các mẫu mà dữ liệu về các mặt hàng hoặc bản thân người dùng không thể tìm thấy. Dưới đây là một số điểm có thể giúp quyết định xem có thể sử dụng tính năng lọc cộng tác hay không:

- Lọc cộng tác không yêu cầu các tính năng về các mục hoặc người dùng phải được biết đến. Nó phù hợp với một tập hợp các loại mặt hàng khác nhau. Ví dụ: kho hàng của siêu thị, nơi có thể thêm các mặt hàng thuộc nhiều danh mục khác nhau. Tuy nhiên, trong một tập hợp các mặt hàng tương tự như của một hiệu sách, các tính năng đã biết như tác giả và thể loại có thể hữu ích và có thể được hưởng lợi từ các phương pháp tiếp cận dựa trên nội dung hoặc kết hợp.
- Lọc cộng tác có thể giúp hệ thống giới thiệu không quá chuyên môn hóa hồ sơ của người dùng và đề xuất các mục hoàn toàn khác với những gì họ đã thấy trước đây. Nếu muốn hệ thống giới thiệu không đề xuất một đôi giày thể thao cho người vừa mua một đôi giày thể thao tương tự khác thì hãy thử thêm tính năng lọc cộng tác vào hệ thống giới thiệu.

Mặc dù lọc cộng tác được sử dụng rất phổ biến trong những hệ thống đề xuất, nhưng một số thách thức gặp phải khi sử dụng nó như sau:

- Lọc cộng tác có thể dẫn đến một số vấn đề như khởi động nguội cho các mục mới được thêm vào danh sách. Cho đến khi ai đó đánh giá chúng, chúng sẽ không được đề xuất.
- Sự thưa thớt dữ liệu có thể ảnh hưởng đến chất lượng của công cụ đề xuất dựa trên người dùng và cũng làm tăng thêm vấn đề khởi đầu nguội đã đề cập ở trên.
- Việc mở rộng quy mô có thể là một thách thức đối với việc phát triển các tập dữ liệu vì độ phức tạp có thể trở nên quá lớn. Đề xuất dựa trên mục nhanh hơn dựa trên người dùng khi tập dữ liệu lớn.
- Với cách triển khai đơn giản, có thể nhận thấy rằng các đề xuất có xu hướng phổ biến và các mục ít phổ biến hơn có thể bị bỏ qua.

Với mỗi loại thuật toán đề xuất đều có danh sách ưu và nhược điểm riêng. Lợi ích của việc nhiều thuật toán hoạt động cùng nhau hoặc theo một quy trình có thể giúp thiết lập các đề xuất chính xác hơn. Trên thực tế, giải pháp của Netflix cũng là sự kết hợp phức tạp của nhiều thuật toán.

## 2.7 Kết luận

Giờ đây, khi đã biết những phép tính nào được đưa vào công cụ đề xuất loại lọc cộng tác và cách thử nhanh các loại thuật toán khác nhau trên tập dữ liệu của mình để xem liệu lọc cộng tác có phải là giải pháp phù hợp hay không. Ngay cả khi nó có vẻ không phù hợp với dữ liệu với độ chính xác cao, một số trường hợp sử dụng được thảo luận có thể giúp lập kế hoạch mọi thứ theo cách kết hợp lâu dài.

# CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

## 3.1 Code xử lý dữ liệu và gợi ý phim cho người dùng

### 3.1.1 Dữ liệu đầu vào

Dữ liệu đầu vào sẽ là bảng gồm User-id, Movie-id và Rating (từ 1 sao đến 5 sao).

Tại đây nhóm tác giả xử lý chuỗi nhập vào có cấu trúc (user-id, movie-id, rating) và đưa vào list lưu tạm "a". Các biến "m" và "n" chứa giá trị tương ứng là số bộ phim và số người xem. Từ dữ liệu có được ở trên, nhóm tác giả tổng hợp vào một ma trận m*n với hàng là bộ phim còn cột là người xem, với những khoảng trống do người xem không thực hiện đánh giá thì sẽ đem giá trị -1.

```
def getData():
    m = 0
    n = 0
    a = list()
    tmp = list()
    while tmp != [-1,-1,-1]:
        tmp = [int(i) for i in input().split()]
        m = max(m, tmp[1])
        n = max(n, tmp[0])
        a.append(tmp)
    a.pop()
    print(a)
    m += 1
    n += 1
    matrix = [[-1]*n for i in range(m)]
    for i in range(len(a)):
        matrix[a[i][1]][a[i][0]] = a[i][2]
    return matrix, m, n
```

**Hình 3.1:** Code tiền xử lý dữ liệu

### 3.1.2 Thuật toán tìm kiếm các bộ phim tương đồng (similarity)

Để có thể tính toán độ tương đồng giữa các bộ phim thì nhóm tác giả phải biết đánh giá của toàn bộ người dùng lên từng bộ phim. Trên thực tế, có những người xem dù đã trải nghiệm nhưng không thực hiện đánh giá, hay có người xem khó tính, họ luôn đánh giá các bộ phim họ trải nghiệm với mức điểm thấp, hoặc người xem dễ dãi, với mọi bộ phim họ trải nghiệm đều đánh giá chúng ở mức điểm trung bình đến cao. Để đối phó với các vấn đề kể trên, có giải pháp được áp dụng có tên centered. Giải pháp centered sẽ đưa mức điểm trung bình của tất cả người dùng về 0 tức với các bộ phim mà họ không đánh giá thì nhóm tác giả đều coi là ở mức trung bình (bằng 0) và các bộ phim họ đánh giá sẽ lấy điểm đó trừ đi trung bình cộng của tất cả điểm đánh giá cho bộ phim đó.

```python
def centered(matrix, n, m):
    data = [[-1]*n for i in range(m)]

    for i in range(m):
        for j in range(n):
            data[i][j] = matrix[i][j]

    avg = list()
    for i in range(m):
        sum = 0
        cnt = 0
        for t in range(n):
            if data[i][t] != -1:
                sum += data[i][t]
                cnt+=1
        avg.append(sum/cnt)
```

```python
    for i in range(m):
        for t in range(n):
            if data[i][t] != -1:
                data[i][t] = data[i][t] - avg[i]
            else:
                data[i][t] = 0

    return data
```

**Hình 3.2:** Hàm centered

Từ ma trận dữ liệu được centered, nhóm tác giả sử dụng công thức cosine similarity. Với kết quả trả ra càng cao thì độ tương đồng của bộ phim này với bộ phim nhóm tác giả đang xét M càng lớn và ngược lại với kết quả trả ra càng thấp thì độ tương đồng giữa bộ phim này với bộ phim M nhóm tác giả đang xét càng thấp.

$$similarity(X, Y) = \frac{\sum_{i=0}^{n} X_i Y_i}{\sqrt{\sum_{i=0}^{n} X_i^2}\sqrt{\sum_{i=0}^{n} Y_i^2}}$$

Trong đó:

$X_i$: Đánh giá của người xem i với bộ phim X

$Y_i$: Đánh giá của người xem i với bộ phim Y

$n$: Số lượng người xem

```
def sim(x, y, n, data):
    a = 0
    b = 0
    c = 0
    for i in range(n):
        a += data[x][i] * data[y][i]
        b += math.pow(data[x][i], _y: 2)
        c += math.pow(data[y][i], _y: 2)

    result = "{:.2f}".format(a/(math.sqrt(b)*math.sqrt(c)))

    return float(result)
```

**Hình 3.3:** Hàm sim tính độ tương đồng giữa hai bộ phim x và y

### 3.1.3 Tính toán dự đoán đánh giá từ người xem

Trong danh sách các sản phẩm thông qua thuật toán tìm kiếm bộ phim tương đồng, nhóm tác giả thực hiện lọc các bộ phim có độ tương đồng cao (similarity > 0), đã được đối tượng U thực hiện đánh giá. Sau đó nhóm tác giả tính trung bình cộng các điểm đánh giá của các sản phẩm lọc được. Kết quả cuối cùng là kết quả dự đoán người dùng U đánh giá bộ phim nhóm tác giả đang xét M.

Trên thực tế, trong danh sách các bộ phim tương đồng thì không phải tất cả các bộ phim đều có độ tương đồng cao với sản phẩm nhóm tác giả đang xét M. Ví dụ trong 5 bộ phim lọc được thì chỉ có các bộ phim thuộc top 2 là có độ tương đồng cao với bộ phim M, còn lại thì không có độ tương đồng đủ cao giống các bộ phim thuộc top 2. Nhằm tăng độ chính xác của dự đoán nhóm tác giả thêm trọng số cho các bộ phim để tính toán dự đoán đánh giá của người dùng U với bộ phim M theo công thức dưới:

$$predict(U, M) = \frac{\sum_{i=0}^{n} X_i * similarity(X_i, M)}{\sum_{i=0}^{n} similarity(X_i, M)}$$

Trong đó:

$X_i$: Điểm đánh giá của người xem U với bộ phim $X_i$.

$similarity(X_i, M)$: Độ tương đồng giữa bộ phim $X_i$ và bộ phim M.

Trong trường hợp không có bộ phim nào thỏa mãn điều kiện (độ tương đồng lớn hơn 0 và bộ phim đó phải được đánh giá bởi người xem U) thì kết quả dự đoán sẽ được trả về là -1 (không dự đoán được).

```
def getPredict(matrix, m, n, target, movie):
    data = centered(matrix, n, m)
    similarity = list()
    for i in range(m):
        if i != movie:
            tmp = sim(x=movie, y=i, n=n, data=data)
            similarity.append((i, tmp))
    similarity.sort(key=lambda x: x[1], reverse=True)

    x = 0
    y = 0
    predict = 0
    for i in range(len(similarity)):
        if matrix[similarity[i][0]][target] != -1 and similarity[i][1] > 0:
            x += matrix[similarity[i][0]][target] * similarity[i][1]
            y += similarity[i][1]
```

```
    # Trả về kết quả dự đoán:
    if x != 0 and y != 0:
        predict = x / y
    else:
        predict = -1

    return predict, similarity
```

**Hình 3.4:** Hàm dự đoán

### 3.1.4 Tổng hợp

Để có thể đưa ra list danh sách các bộ phim gợi ý cho người xem, nhóm tác giả phải thực hiện dự đoán đánh giá người xem với các bộ phim mà đối tượng chưa đánh giá hay chưa xem. Từ danh sách tìm được, nhóm tác giả thực hiện gợi ý cho người xem top 3 bộ phim có thể người xem sẽ thích.

```
#Code này đang xét toàn bộ movie chưa được rated bởi target:
for movie in range(m):
    if matrix[movie][target] == -1:
        predict, similarity = getPredict(matrix, m, n, target, movie)
        result_all.append((movie, "{:.2f}".format(predict)))
        print("Movie " + str(movie) + " with:", end=" ")
        print(similarity) # Trả danh sách các bộ phim tương đồng
result_all.sort(key=lambda x: x[1], reverse=True)
print("All predict rating from user {} (movie_id, predict):".format(target))
print(result_all)
print("Recommend to user {} top 3 movies:".format(target))
cnt = 0
for i in range(len(result_all)):
    if cnt == 3:
        break
    if result_all[i][1] != "{:.2f}".format(-1):
        print(result_all[i], end=" ")
        cnt+=1
```

**Hình 3.5:** Code tổng hợp

### 3.1.5 Đánh giá mô hình bằng RMSE

Root Mean Square Error (RMSE) hoặc Root Mean Square Deviation (RMSD) là căn bậc hai mức trung bình của các sai số bình phương. RMSE là độ lệch chuẩn của các phần dư (sai số dự đoán).

Phần dư là thước đo khoảng cách từ các điểm dữ liệu đường hồi quy; RMSE là thước đo mức độ dàn trải của những phần dư này, nói cách khác, nó cho bạn biết mức độ tập trung của dữ liệu xung quanh đường phù hợp nhất.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(M_i - predict(U, M_i))^2}$$

Trong đó:

$M_i$: Điểm đánh giá chính xác người xem U với bộ phim $M_i$

$predict(U, M_i)$ : Điểm đánh giá dự đoán người xem U với bộ phim $M_i$

So với các mô hình học máy khác, để đánh giá mức độ tốt của một mô hình, từ dữ liệu, nhóm tác giả có sẽ chia nửa để train và nửa còn lại thực hiện test, nhưng đối với lọc cộng tác nhóm tác giả không thể làm vậy.

Vậy để đánh giá độ tốt của mô hình lọc cộng tác, nhóm tác giả vờ như tại matrix[M][U] (đánh giá của người dùng U lên bộ phim M) không tồn tại để rồi đi tính dự đoán của U đối với bộ phim M đó. Từ dữ liệu dự đoán và dữ liệu thực đã

có, nhóm tác giả áp dụng RMSE đã nêu ở trên để đánh giá mô hình.

Giá trị RMSE trả về -1 khi mọi giả định nhóm tác giả không tính được dự đoán.

```
def rmse(matrix, m, n, target):
    cnt = 0
    x = 0
    for movie in range(m):
        if matrix[movie][target] != -1:
            test = [[-1] * n for i in range(m)]
            for i in range(m):
                for j in range(n):
                    test[i][j] = matrix[i][j]
            test[movie][target] = -1
            predict, similarity = CollaberativeFiltering.getPredict(test, m,
                                                                     n, target, movie)
            if predict != -1:
                x += math.pow(matrix[movie][target] - predict, _y: 2)
                cnt += 1
```

```
    if cnt != 0:
        result = math.sqrt(x/cnt)
    else:
        result = -1

    return result
```

**Hình 3.6:** Hàm RMSE

## 3.2 Kết quả thực nghiệm

- Tại thử nghiệm này, nhóm tác giả sẽ trả ra danh sách kết quả gồm:
  - Danh sách các bộ phim tương đồng với bộ phim nhóm tác giả cần tính dự đoán.
  - Danh sách kết quả dự đoán của các bộ phim.
  - Danh sách các bộ phim thuộc top 3 để gợi ý cho người xem.
  - Giá trị RMSE.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | | 3 | | | 5 | | | 5 | | 4 | |
| 1 | | | 5 | 4 | | | 4 | | | 2 | 1 | 3 |
| 2 | 2 | 4 | | 1 | 2 | | 3 | | 4 | 3 | 5 | |
| 3 | | 2 | 4 | | 5 | | | 4 | | | 2 | |
| 4 | | | 4 | 3 | 4 | 2 | | | | | 2 | 5 |
| 5 | 1 | | 3 | | 3 | | | 2 | | | 4 | |

**Hình 3.7:** Ma trận dữ liệu được thử nghiệm (hàng user-id, cột movie-id)

- Thử nghiệm và đánh giá các trường hợp:

```
Give a target id: 0
Movie 1 with: [(3, 0.47), (4, 0.4), (0, -0.18), (5, -0.31), (2, -0.53)]
Movie 3 with: [(1, 0.47), (4, 0.46), (0, -0.1), (5, -0.24), (2, -0.62)]
Movie 4 with: [(3, 0.46), (1, 0.4), (5, -0.22), (2, -0.28), (0, -0.31)]
All predict rating from user 0 (movie_id, predict):
[(1, '-1.00'), (3, '-1.00'), (4, '-1.00')]
Recommend to user 0 top 3 movies:

RMSE value: 0.95
```

**Hình 3.8:** Kết quả dự đoán đánh giá của người xem id 0

Thử nghiệm 3.8 là trường hợp không thể dự đoán đánh giá của người xem id 0 với các bộ phim.

Với bộ phim 1, nhóm tác giả thấy các bộ phim có độ tương đồng cao là bộ phim 3 và 4 nhưng trong ma trận dữ liệu nhóm tác giả thấy bộ phim 3 và 4 đều chưa được đánh giá bởi người xem id 0. Bởi vậy bộ phim 1 không thể dự đoán đánh giá.

Các bộ phim 3 và 4 cũng không thể dự đoán đánh giá lý do tương tự với bộ phim 1.

RMSE = 0.95 với việc mức đánh giá người xem từ 1 - 5 thì nhóm tác giả thấy mô hình này đạt hiệu quả.

```
Give a target id: 1
Movie 0 with: [(5, 0.59), (2, 0.41), (3, -0.1), (1, -0.18), (4, -0.31)]
Movie 1 with: [(3, 0.47), (4, 0.4), (0, -0.18), (5, -0.31), (2, -0.53)]
Movie 4 with: [(3, 0.46), (1, 0.4), (5, -0.22), (2, -0.28), (0, -0.31)]
Movie 5 with: [(0, 0.59), (2, 0.51), (4, -0.22), (3, -0.24), (1, -0.31)]
All predict rating from user 1 (movie_id, predict):
[(0, '4.00'), (5, '4.00'), (1, '2.00'), (4, '2.00')]
Recommend to user 1 top 3 movies:
(0, '4.00') (5, '4.00') (1, '2.00')
RMSE value: Can't evaluate
```

**Hình 3.9:** Kết quả dự đoán đánh giá của người xem id 1

Thử nghiệm 3.9 là trường hợp không trả ra giá trị RMSE.

Trường hợp này xảy ra khi tất cả trường hợp giả sử người xem id 1 không đánh giá bộ phim 2 hoặc 3 mô hình không dự đoán được đánh giá của người xem id 1.

```
Movie 2 with: [(5, 0.54), (0, 0.42), (4, -0.31), (3, -0.49), (1, -0.58)]
Movie 3 with: [(1, 0.59), (4, 0.56), (0, -0.12), (5, -0.4), (2, -0.63)]
RMSE value: Can't evaluate
```

**Hình 3.10:** Kết quả đánh giá mô hình trong trường hợp người xem id 1

Tại hình 3.10 nhóm tác giả thấy, với trường hợp giả sử người xem id 1 không đánh giá bộ phim 2, các bộ phim có độ tương đồng cao với bộ phim 2 là bộ phim 5 và 0 nhưng trong dữ liệu thì cả hai bộ phim này đều chưa được đánh giá bởi người xem id 1. Bởi vậy mô hình không thể dự đoán đánh giá người xem id 1 với bộ phim 2 dẫn đến giá trị RMSE không xác định. Tương tự với bộ phim 3.

```
Give a target id: 2
Movie 2 with: [(5, 0.51), (0, 0.41), (4, -0.28), (1, -0.53), (3, -0.62)]
All predict rating from user 2 (movie_id, predict):
[(2, '3.00')]
Recommend to user 2 top 3 movies:
(2, '3.00')
RMSE value: 0.52
```

**Hình 3.11:** Kết quả dự đoán đánh giá của người xem id 2

Thử nghiệm 3.11 là trường hợp mong đợi khi mô hình dự đoán được đánh giá của người xem id 2 lên bộ phim còn trống là bộ phim 2 và gợi ý người xem nên xem bộ phim này cùng với kết quả đánh giá RMSE.

RMSE = 0.52 với mức đánh giá người xem từ 1 – 5 nhóm tác giả thấy mô hình này có sai số bình phường trung bình gốc thấp, đạt hiệu quả.

```
Give a target id: 10
All predict rating from user 10 (movie_id, predict):
[]
Recommend to user 10 top 3 movies:

RMSE value: 1.19
```

**Hình 3.12:** Kết quả dự đoán đánh giá của người xem id 10

Thử nghiệm 3.12 là trường hợp khi người xem đã đánh giá hết tất cả các bộ phim hay tất cả các bộ phim đều được xem bởi người xem. Bởi vậy, nhóm tác giả không gợi ý thêm bộ phim nào cho người xem id 10.

RMSE = 1.19 với mức đánh giá người xem từ 1 – 5 cho nhóm tác giả thấy mô hình này có sai số bình phường trung bình gốc thấp, đạt hiệu quả.

# CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 4.1 So sánh với sản phẩm tuong tự

Bên trên, nhóm tác giả đã trình bày mô hình láng giềng trong lọc cộng tác – mô hình tư vấn dựa trên độ tương tự trực tiếp giữa hai người dùng hoặc sản phẩm. Trong mô hình này, nhóm tác giả tính toán độ tương tự giữa hai sản phẩm, từ đó đưa ra dự đoán đánh giá của người dùng với sản phẩm mới. Nhóm tác giả sẽ so sánh với "Hệ thống gợi ý sản phẩm trong bán hàng trực tuyến sử dụng kỹ thuật lọc cộng tác" của Đại học Cần Thơ. Nhóm tác giả sẽ so sánh về thuận toán, bộ dữ liệu và kết quả thực nghiệm.

### 4.1.1 Thuật toán

Hệ thống của Đại học Cần Thơ sử dụng mô hình lọc cộng tác dựa trên người dùng. Họ sẽ xác định sự tương tự giữa hai người dùng thông qua việc so sánh các đánh giá của họ trên cùng sản phẩm, sau đó dự đoán đánh giá sản phẩm i bởi người dùng u, hay chính là đánh giá trung bình của những người dùng tương tự với người dùng u.

Độ tương tự giữa 2 người dùng sẽ được tính bằng công thức Pearson thay vì công thức cosine similarity được trình bày ở chương 2:

$$sim_{pearson}(u,u') = \frac{\sum_{i \in I_{uu'}} (r_{ui} - \bar{r}_u)(r_{u'i} - \bar{r}_{u'})}{\sqrt{\sum_{i \in I_{uu'}} (r_{ui} - \bar{r}_u)^2 \sum_{i \in I_{uu'}} (r_{u'i} - \bar{r}_{u'})^2}} \quad \text{(CT 1)}$$

**Hình 4.1:** Công thức pearson

Do công thức Pearson sẽ bỏ qua các mục không được đánh giá nên sẽ không cần hàm centered sẽ đưa mức điểm trung bình của tất cả người dùng về 0

Sau khi tính toán độ tương tự giữa các người dùng có thể dự đoán đánh giá của người dùng u trên sản phẩm i theo công thức như sau:

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{u' \in K_u} sim(u,u') \cdot (r_{u'i} - \bar{r}_{u'})}{\sum_{u' \in K_u} |sim(u,u')|} \quad \text{(CT 2)}$$

**Hình 4.2:** Công thức tính đánh giá dự đoán

Công thức trên tương tự với công thức tính toán dự đoán đánh giá từ người xem đã trình bày. Chỗ khác ở đây là các đối tượng được so sánh. Các công thức trên có

đối tượng là người dùng.

Giải thuật lọc cộng tác dựa trên người dùng lân cận gần nhất (User-KNN) sử dụng độ tương tự Pearson bằng ngôn ngữ giả để dự đoán độ thích cho người dùng u trên sản phẩm i được biểu diễn như sau:

1: **procedure** USERKNN-CF ($\overline{r_u}$ ,$r$,$D^{train}$)
2: **for** $u$=1 to $N$ **do**
3: Tính Sim_uu' , sử dụng công thức (CT 1)
4: **end for**
5: **Sort** Sim_uu' // sắp xếp giảm dần độ tương tự
6: **for** $k$=1 to $K$ **do**
7: $K_u \leftarrow k$ // Các người dùng k gần nhất của u
8: **end for**
9: **for** $i$ = 1 to $M$ **do**
10: Tính $\widehat{r_{ui}}$ , sử dụng công thức (CT 2)
11: **end for**
12: end procedure

**Hình 4.3:** Giả ngôn ngữ mô hình dự đoán đánh giá người dùng bên Đại học Cần Thơ

Về cơ bản thì giải thuật sẽ gồm có 2 phần là tính độ tương đồng giữa 2 người dùng và dùng dữ liệu đó để dự đoán đánh giá.

### 4.1.2 Bộ dữ liệu

Trong hệ thống gợi ý thường biểu diễn các đánh giá của người dùng cho các sản phẩm qua ma trận gồm một tập người dùng U và tập sản phẩm I. Và hệ thống lọc cộng tác bên Đại học Cần Thơ cũng sử dụng phương pháp đó.

Thay vì bộ dữ liệu tự tạo, họ sử dụng tập dữ liệu MovieLens 100K. Đây là một bộ dữ liệu lớn giúp làm tăng độ chính xác của thuật toán.

Tập dữ liệu này được chia làm 2 phần, 1 phần làm đầu vào và 1 phần để kiểm tra. Điều cho kiểm tra được xem thuật toán có chạy đúng như dự tính không.

### 4.1.3 Kết quả thực nghiệm

Phương pháp đánh giá: vì thuật toán của cả 2 là tương đồng nhau nên để đánh giá hiệu quả của thuật toán thì vẫn sẽ sử dụng Root Mean Squared Error (RMSE). Ngoài ra, họ còn đo thêm thời gian tính toán để đưa ra hiệu năng. Số liệu trung

bình của hệ thống này là RMSE = 0.900996. Trong khi đó khi lấy trung bình các ví dụ trong chương 3 ta đươc RMSE = 0.886667. Có thể thấy kết quả của sản phẩm của nhóm tác giả tốt hơn so với hệ thống trên, mặc dù không nhiều. Độ chênh lệch này có thể là do công thức tính độ tương đồng gây ra. Công thức cosine similarity sẽ không chỉ xem xét được dữ kiệu tương đồng mà còn xem xét dữ liệu khác nhau, từ đó sẽ cho ta kết quả chính xác hơn.

## 4.2 Hướng phát triển

Hệ thống sẽ bao gồm các bước chính sau

- Thu thập dữ liệu
- Xây dựng ma trận Tiện ích
- Tính độ tương đồng
- Dự đoán sở thích
- Tạo gợi ý sản phẩm

Ở bước thu thập dữ liệu, đối với người dùng cũ, ngoài việc thu thập những thông tin như lịch sử xem, đánh giá và thói quen sử dụng thì có thể thu thập thêm dữ liệu về hành vi người dùng như là tìm kiếm hoặc đọc mô tả một bộ phim nào đó rồi từ đó suy ra sở thích và đánh giá. Đối với những người dùng mới sẽ cần thu thập thêm một số thông tin như là: sở thích, tính cách, nghề nghiệp,... . Việc này sẽ giúp tư vấn tốt hơn cho các người dùng mới.

Ở bước tính độ tương đồng, sau khi tính xong có thể gộp những sản phẩm có độ tương thích cao vào một nhóm. Khi thêm sản phẩm mới vào sẽ chỉ cần tính độ tương đồng với đại diện của mỗi nhóm. Điều này giúp giảm thời gian tính toán.

Ở bước tạo gợi ý sản phẩm, trong các trường hợp không thể dự đoán được đánh giá của người dùng thì sẽ gợi ý cho họ những bộ phim được đánh giá cao nhất.

## 4.3 Đóng góp

Trần Việt Anh - B21DCCN162: Ứng dụng mô hình vào trang web gợi ý phim.

Nguyễn Bá Hải Long - B21DCAT119: Viết báo cáo chương 3 và lập trình mô hình lọc cộng tác.

Tô Quang Huy - B21DCAT104: Viết báo cáo chương 4 và làm slide.

Lê Trần Hiếu - B21DCAT088: Viết báo cáo chương 1.

Lưu Đức Hải - B21DCAT081: Viết báo cáo chương 2.

# TÀI LIỆU THAM KHẢO

[1] Toby Segaran, *Programming Collective Intelligence: Building Smart Web 2.0 Applicationsh*. O'Reilly Media, 2007.

[2] Jure Leskovec, Anand Rajaraman, Jeffrey David Ullman, *Mining of Massive Datasets*. Cambridge University Press, 2014.

[3] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, *Item-Based Collaborative Filtering Recommendation Algorithms*. [Online]. Available: `https://files.grouplens.org/papers/www10_sarwar.pdf`).

[4] Nguyễn Hùng Dũng và Nguyễn Thái Nghe, Hệ thống gợi ý sản phẩm trong bán hàng trực tuyến sử dụng kỹ thuật lọc cộng tác, *Tạp chí Khoa học Trường Đại học Cần Thơ*, 2014.