|  |  |
| --- | --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

TRƯỜNG ĐẠI HỌC THỦY LỢI



VŨ THỊ NỤ

Phân TÍCH VÀ ỨNG DỤNG CÁC PHƯƠNG PHÁP HỌC MÁY TRONG BÀI TOÁN ĐỀ XUẤT PHIM

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2024

|  |  |
| --- | --- |
| **bỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

TRƯỜNG ĐẠI HỌC THỦY LỢI

VŨ THỊ NỤ

Phân TÍCH VÀ ỨNG DỤNG CÁC PHƯƠNG PHÁP HỌC MÁY TRONG BÀI TOÁN ĐỀ XUẤT PHIM

|  |  |  |
| --- | --- | --- |
| Ngành | : | Hệ thống thông tin |
| Mã ngành | : | 7480104 |

|  |  |  |
| --- | --- | --- |
| NGƯỜI HƯỚNG DẪN | : | TS. TẠ QUANG CHIỂU |

HÀ NỘI, NĂM 2024

|  |  |
| --- | --- |
|  | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  **Độc lập - Tự do - Hạnh phúc**  ----------★----------  **NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP** |

**Họ tên sinh viên:** Vũ Thị Nụ **Hệ đào tạo:** Đại học chính quy

**Lớp:** 61HT **Ngành:** Hệ thống thông tin

**Khoa:** Công nghệ thông tin

1. **TÊN ĐỀ TÀI**

PHÂN TÍCH VÀ ỨNG DỤNG CÁC PHƯƠNG PHÁP HỌC MÁY TRONG BÀI TOÁN ĐỀ XUẤT PHIM.

1. **CÁC TÀI LIỆU CƠ BẢN**

[1] Tìm hiểu về hệ thống đề xuất phim sử dụng các phương pháp học máy <https://www.shiksha.com/online-courses/articles/movie-recommendation-system-using-machine-learning/>

[2] Hướng dẫn xây dựng hệ thống đề xuất bằng Python <https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-python/>

[3] Xây dựng Collaborative Filtering RS <https://viblo.asia/p/xay-dung-collaborative-filtering-rs-recommender-system-co-ban-phan-3-Az45bMqolxY>

1. **NỘI DUNG CÁC PHẦN THUYẾT MINH VÀ TÍNH TOÁN**

|  |  |
| --- | --- |
| **Nội dung trình bày** | **Tỷ lệ %** |
| Chương 1: Giới thiệu bài toán, giới thiệu tổng quan | 10% |
| Chương 2: Tổng quan về hệ đề xuất | 30% |
| Chương 3: Ứng dụng thuật toán KNN và SVD đề xuất phim | 40% |
| Chương 4: Kết quả và đánh giá mô hình | 20% |

1. **GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN**

|  |  |
| --- | --- |
| **Phần** | **Giáo viên hướng dẫn** |
| Chương 1: Giới thiệu bài toán, giới thiệu tổng quan | TS. Tạ Quang Chiểu |
| Chương 2: Tổng quan về hệ đề xuất | TS. Tạ Quang Chiểu |
| Chương 3: Ứng dụng thuật toán KNN và SVD đề xuất phim | TS. Tạ Quang Chiểu |
| Chương 4: Kết quả và đánh giá mô hình | TS. Tạ Quang Chiểu |

1. **NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Ngày ............ tháng ......... năm 202…

|  |  |
| --- | --- |
| **Trưởng Bộ môn**  *(Ký và ghi rõ Họ tên)* | **Giáo viên hướng dẫn chính**  *(Ký và ghi rõ Họ tên)* |

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua

Ngày … tháng … năm 202… **Chủ tịch Hội đồng**

*(Ký và ghi rõ Họ tên)*

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi ngày... tháng...

năm 202…

**Sinh viên làm Đồ án tốt nghiệp**

(Ký và ghi rõ Họ tên)

|  |  |
| --- | --- |
| logo | TRƯỜNG ĐẠI HỌC THỦY LỢI  **KHOA CÔNG NGHỆ THÔNG TIN**  BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP |

**Tên đề tài:** **Phân tích và ứng dụng các phương pháp học máy trong bài toán đề xuất phim.**

*Sinh viên thực hiện* : **Vũ Thị Nụ**

*Lớp* : **61HT**

*Mã sinh viên* : **1951060908**

*Số điện thoại* : **0334495434**

*Email* : **vuthinu27@gmail.com**

*Giáo viên hướng dẫn* : **TS. Tạ Quang Chiểu**

TÓM TẮT ĐỀ TÀI

Thế kỷ 21 là thời đại của dữ liệu và thông tin, một lượng lớn dữ liệu mang đến nhiều cơ hội và kéo theo đó là vấn đề quá tải thông tin. Trong lĩnh vực phim ảnh, người dùng không thể tìm thấy những bộ phim họ quan tâm một cách nhanh chóng và chính xác trước lượng dữ liệu phim khổng lồ.

Bài toán đề xuất phim là một phương tiện hữu hiệu để giải quyết vấn đề quá tải thông tin phim. Mục tiêu của bài toán là xây dựng hệ thống có khả năng gợi ý các bộ phim phù hợp với sở thích, yêu cầu của người dùng. Để giải quyết bài toán này, em đề xuất phát triển một mô hình học máy sử dụng bộ dữ liệu phim MovieLens đưa ra đề xuất phim. Mô hình sẽ học từ các đặc trưng của người dùng như lịch sử xem phim, đánh giá, sở thích, và từ các đặc trưng của phim như thể loại, diễn viên, đạo diễn, và đánh giá của người dùng khác. Kết quả của đề tài này có thể được áp dụng trong các hệ thống gợi ý phim trực tuyến, giúp người dùng tìm kiếm và khám phá các bộ phim mới một cách dễ dàng và thuận tiện.

Từ những vấn đề, cùng với sự phát triển của công nghệ hiện đại và tầm quan trọng của đề xuất phim. Vì vậy, em lựa chọn đề tài “**Phân tích và ứng dụng các phương pháp học máy trong bài toán đề xuất phim**”.

**CÁC MỤC TIÊU CHÍNH**

Mục tiêu 1: Tìm hiểu bài toán đề xuất phim.

Mục tiêu 2: Tìm hiểu, nghiên cứu một số mô hình học máy.

Mục tiêu 3: Áp dụng các mô hình học máy trong bài toán đề xuất phim.

Mục tiêu 4: Đánh giá mô hình.

Mục tiêu 5: Kết luận.

KẾT QUẢ DỰ KIẾN

Hoàn thành các mục tiêu đề ra.

Xây dựng được mô hình học máy cho bài toán đề xuất phim.

Báo cáo tổng kết ĐATN.

**LỜI CAM ĐOAN**

Tác giả xin cam đoan rằng Đồ án tốt nghiệp này là công trình nghiên cứu độc lập của bản thân tác giả. Các kết quả nghiên cứu trong Đồ án được thực hiện một cách trung thực và không sao chép từ bất kỳ nguồn tài liệu nào. Tất cả các nguồn tài liệu tham khảo đã được trích dẫn và ghi rõ nguồn gốc theo quy định.

|  |  |
| --- | --- |
|  | **Tác giả ĐATN**    **Vũ Thị Nụ** |
|  |  |

LỜI CÁM ƠN

Tôi xin chân thành cảm ơn Ban Giám Hiệu trường Đại Học Thủy Lợi đã tạo điều kiện cho tôi được học tập trong môi trường tốt, cảm ơn các thầy cô giảng viên trong khoa Công Nghệ Thông Tin đã dìu dắt, dạy cho tôi những kiến thức bổ ích về ngành, những kiến thức không chỉ có trong sách vở mà còn trong công việc và trong đời sống giúp tôi vững bước trên chặng đường mà tôi đang hướng đến.

Tôi xin chân thành cảm ơn TS. Tạ Quang Chiểu, người đã tận tâm và tận tình hướng dẫn đồ án tốt nghiệp của tôi. Nhờ sự chỉ bảo và giúp đỡ của thầy, tôi đã học hỏi được nhiều kiến thức mới và có thể hoàn thành đồ án tốt nghiệp kỳ này. Tôi rất mong trong tương lai tới, dù không còn ngồi trên ghế giảng đường nữa, tôi vẫn mong muốn được học tập và tiếp thu thêm nhiều kiến thức mới từ thầy.

Với hạn chế về thời gian và kiến thức chuyên ngành, trong quá trình thực hiện đồ án, tôi không tránh khỏi việc mắc phải những sai sót. Do đó, tôi rất mong nhận được sự đóng góp, góp ý và đánh giá từ các thầy cô để có thể hoàn thiện đồ án của mình một cách tốt nhất.

Tôi xin chân thành cảm ơn!

MỤC LỤC

[DANH SÁCH CÁC HÌNH ẢNH v](#_Toc156802243)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ vi](#_Toc156802244)

[CHƯƠNG 1 GIỚI THIỆU BÀI TOÁN, GIỚI THIỆU TỔNG QUAN 1](#_Toc156802245)

[1.1 Lý do chọn đề tài 1](#_Toc156802246)

[1.2 Mục tiêu, phạm vi 1](#_Toc156802247)

[1.3 Đối tượng, phạm vi nghiên cứu 2](#_Toc156802248)

[1.4 Khu vực nghiên cứu 3](#_Toc156802249)

[CHƯƠNG 2 TỔNG QUAN VỀ HỆ ĐỀ XUẤT (RECOMMENDER SYSTEM) 4](#_Toc156802250)

[2.1 Khái niệm Hệ đề xuất (Recommender system – RS) 4](#_Toc156802251)

[2.2 Các lĩnh vực ứng dụng của Hệ đề xuất 4](#_Toc156802252)

[2.3 Phát biểu bài toán Hệ đề xuất 5](#_Toc156802253)

[2.4 Các phương pháp xây dựng bài toán Hệ đề xuất 7](#_Toc156802254)

[2.4.1 Hệ đề xuất sử dụng lọc theo nội dung 7](#_Toc156802255)

[2.4.2 Hệ đề xuất sử dụng lọc cộng tác 9](#_Toc156802256)

[2.4.3 Hệ đề xuất sử dụng lọc kết hợp/lai ghép (Hybrid Filtering) 13](#_Toc156802257)

[2.5 Thuật toán K-Nearest Neighbors (KNN) 14](#_Toc156802258)

[2.5.1 Định nghĩa 14](#_Toc156802259)

[2.5.2 Nguyên lý hoạt động 14](#_Toc156802260)

[2.5.3 Thuật toán KNN trong bài toán đề xuất phim 17](#_Toc156802261)

[2.5.4 Ưu, nhược điểm của thuật toán KNN 17](#_Toc156802262)

[2.6 Thuật toán Singular Value Decomposition (SVD) 18](#_Toc156802263)

[2.6.1 Định nghĩa 18](#_Toc156802264)

[2.6.2 Nguyên lý hoạt động 18](#_Toc156802265)

[2.6.3 Thuật toán SVD trong bài toán đề xuất phim 19](#_Toc156802266)

[2.6.4 Ưu, nhược điểm của thuật toán SVD 20](#_Toc156802267)

[2.7 Độ đo đánh giá 20](#_Toc156802268)

[2.7.1 Độ đo tương tự Cosine similarity 20](#_Toc156802269)

[CHƯƠNG 3 ỨNG DỤNG THUẬT TOÁN KNN VÀ SVD XÂY DỰNG ĐỀ XUẤT PHIM 22](#_Toc156802270)

[3.1 Các công cụ và thư viện sử dụng 22](#_Toc156802271)

[3.1.1 Ngôn ngữ lập trình Python 22](#_Toc156802272)

[3.1.2 Thư viện sử dụng 23](#_Toc156802273)

[3.2 Phân tích khám phá dữ liệu (EDA) 28](#_Toc156802274)

[3.2.1 Khám phá dữ liệu 28](#_Toc156802275)

[3.2.2 Phân tích dữ liệu 31](#_Toc156802276)

[3.2.3 Tiền xử lý dữ liệu 34](#_Toc156802277)

[3.2.4 Xây dựng mô hình KNN 35](#_Toc156802278)

[3.2.5 Xây dựng mô hình SVD 38](#_Toc156802279)

[CHƯƠNG 4 KẾT QUẢ VÀ ĐÁNH GIÁ MÔ HÌNH 42](#_Toc156802280)

[4.1 Kết quả xây dựng mô hình KNN đề xuất phim 42](#_Toc156802281)

[4.2 Kết quả xây dựng mô hình SVD đề xuất phim 43](#_Toc156802282)

[4.3 Đánh giá kết quả xây dựng mô hình KNN và SVD đề xuất phim 43](#_Toc156802283)

[KẾT LUẬN 45](#_Toc156802284)

[TÀI LIỆU THAM KHẢO 46](#_Toc156802285)

DANH SÁCH CÁC HÌNH ẢNH

[Hình 2.1 Hệ thống đề xuất phim Netflix 4](#_Toc156721355)

[Hình 2.2 Ví dụ ma trận đánh giá tổng quát 6](#_Toc156721356)

[Hình 2.3 Hệ đề xuất sử dụng lọc theo nội dung 7](#_Toc156721357)

[Hình 2.4 Hệ đề xuất sử dụng lọc cộng tác 10](#_Toc156721358)

[Hình 2.5 Ví dụ minh họa thuật toán KNN 15](#_Toc156721359)

[Hình 2.6 Các bước đưa ra đề xuất phim bằng thuật toán KNN 17](#_Toc156721360)

[Hình 2.7 SVD cho ma trận A khi: m < n và m > n 19](#_Toc156721361)

[Hình 2.8 Ma trận SVD giữa item và users 20](#_Toc156721362)

[Hình 3.1 Ngôn ngữ lập trình Python 23](#_Toc156721363)

[Hình 3.2 Dữ liệu MovieLens 100K từ trang web Grouplens 28](#_Toc156721364)

[Hình 3.3 Dữ liệu phim file U-data 29](#_Toc156721365)

[Hình 3.4 Dữ liệu phim file U-item 30](#_Toc156721366)

[Hình 3.5 Đếm số hàng, cột của file dữ liệu 31](#_Toc156721367)

[Hình 3.6 Các cột dữ liệu sử dụng cho bài toán 32](#_Toc156721368)

[Hình 3.7 Phân tích số lượng đánh giá 32](#_Toc156721369)

[Hình 3.8 Phân phối số lượng đánh giá theo phim 33](#_Toc156721370)

[Hình 3.9 Phân phối số lượng đánh giá theo 100 người dùng đầu tiên 33](#_Toc156721371)

[Hình 3.10 Kết quả các thông số tốt nhất 36](#_Toc156721372)

[Hình 3.11 Kết quả tham số tốt nhất 39](#_Toc156721373)

[Hình 4.1 Kết quả mô hình KNN đề xuất phim 42](#_Toc156721374)

[Hình 4.2 Kết quả mô hình SVD đề xuất phim 43](#_Toc156721375)

[Hình 4.3 Kết quả độ tương tự đề xuất phim của mô hình KNN và SVD 44](#_Toc156721376)

DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ

|  |  |
| --- | --- |
| Từ viết tắt | Viết đầy đủ |
| CBF | Content-based recommend |
| CF | Collaborative Filtering |
| GUI | Graphical User Interface |
| HF | Hybrid Filtering |
| KNN | **K-Nearest Neighbors** |
| MAE | Mean Absolute Error |
| ML | Machine Learning |
| RS | Recommender System |
| RMSE | Root Mean Square Error |
| SVD | Singular Value Decomposition |

# GIỚI THIỆU BÀI TOÁN, GIỚI THIỆU TỔNG QUAN

## Lý do chọn đề tài

Ngày nay, nhu cầu giải trí của con người ngày càng cao, trong đó phim ảnh là một trong những hình thức giải trí phổ biến nhất. Việc tìm kiếm một bộ phim phù hợp với sở thích của bản thân là một vấn đề không hề đơn giản, đặc biệt là với số lượng phim ảnh khổng lồ hiện nay. Cùng với sự phát triển của học máy, đã mở ra nhiều cơ hội và ứng dụng mới trong lĩnh vực phim ảnh với khả năng học từ dữ liệu và tạo ra những dự đoán chính xác là một phương tiện hữu hiệu để giải quyết vấn đề này.

Đề xuất phim là một vấn đề thực tế và có tính ứng dụng cao trong ngành công nghiệp điện ảnh và giải trí giúp người dùng tìm kiếm và khám phá những bộ phim phù hợp với sở thích cá nhân. Việc sử dụng học máy để đề xuất phim có thể giúp cải thiện trải nghiệm người dùng, tăng doanh số bán vé và tối ưu hóa quy trình sản xuất phim.

Từ những vấn đề, cùng với sự phát triển của học máy và tầm quan trọng của đề xuất phim. Vì vậy, em lựa chọn đề tài “**Phân tích và ứng dụng các phương pháp học máy trong bài toán đề xuất phim**”.

## Mục tiêu, phạm vi

Tìm hiểu bài toán đề xuất phim.

Tìm hiểu, nghiên cứu một số mô hình học máy sử dụng trong bài toán đề xuất: Content-based recommend (CBF), Collaborative Filtering (CF) và Hybrid Filtering (HF)

Áp dụng các mô hình các học máy của phương pháp Collaborative Filtering là K-Nearest Neighbors (KNN) và Singular Value Decomposition (SVD) xây dựng bài toán đề xuất phim đưa ra đề xuất phim chính xác nhất phù hợp với sở thích cá nhân người dùng giúp cải thiện trải nghiệm người dùng, tăng cường tương tác và khám phá những bộ phim mới nhanh chóng tiết kiệm thời gian, chi phí.

## Đối tượng, phạm vi nghiên cứu

Đối tượng nghiên cứu:

Bộ dữ liệu Movielens: Movielens 100K là một bộ dữ liệu phổ biến được sử dụng trong nghiên cứu khoa học, bao gồm 100.000 xếp hạng từ 943 người dùng cho 1682 bộ phim, với mỗi xếp hạng từ 1 đến 5. Dữ liệu được thu thập thông qua trang web MovieLens (movielens.umn.edu) trong khoảng thời gian bảy tháng từ ngày 19 tháng 9 năm 1997 đến ngày 22 tháng 4 năm 1998. Dữ liệu này đã được lọc để loại bỏ các người dùng có ít hơn 20 xếp hạng hoặc thiếu thông tin cá nhân đầy đủ. Bộ dữ liệu lớn và nhiều thông tin không cần thiết để sử dụng trong bài toán của mình, nên em sẽ sử dụng 2 file dữ liệu của bộ dữ liệu cho bài toán của mình là file u.data và u.item.

U-data là tập dữ liệu đầy đủ gồm 100000 xếp hạng của 943 người dùng trên 1682 mục. User và item được đánh số liên tục từ 1. Bao gồm các cột dữ liệu: user id (id người dùng), item id (id mục), đánh giá (đánh giá) và timestamp (dấu thời gian).

U-item là file dữ liệu thông tin về các item (phim) bao gồm các cột dữ liệu: movie id (id phim), movie title (tiêu đề phim), release date (ngày phát hành), video release date (ngày phát hành video), URL IMDb, unknown, Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western. 19 trường cuối cùng là thể loại, số 1 biểu thị phim thuộc thể loại đó, số 0 biểu thị nó không thuộc thể loại đó.

Trải nghiệm người dùng tại các hệ thống đề xuất phim của Netflix, Amazon Prime Video, Youtube…

*Phạm vi nghiên cứu:*

Nghiên cứu tìm hiểu về các phương pháp học máy được sử dụng trong bài toán đề xuất phim: Content-based recommend (CBF), Collaborative Filtering (CF) và Hybrid Filtering (HF).

Nghiên cứu sâu hơn về phương pháp Collaborative Filtering với các phương pháp học máy Singular Value Decomposition (SVD), K-nearest neighbors (KNN) và áp dụng vào xây dựng bài toán đề xuất phim.

Nghiên cứu việc đánh giá và đo lường hiệu quả của các phương pháp học máy để đưa ra đề xuất phim.

## Khu vực nghiên cứu

Hệ thống đề xuất phim trên các website lớn trên thế giới như Netflix, Amazon Prime Video, Youtube…để kiểm tra tính hiệu quả và đánh giá của người dùng về hệ thống đề xuất phim.

Nghiên cứu hệ thống đề xuất phim tại thị trường phim ảnh online của Việt Nam như FPT Play, Vieon…

Các trung tâm thương mại với các rạp chiếu phim: đề xuất những sản phẩm phim mới nhất, được đánh giá cao, phù hợp với thị hiếu người xem phim tại các màn hình lớn tại trung tâm thương mại.

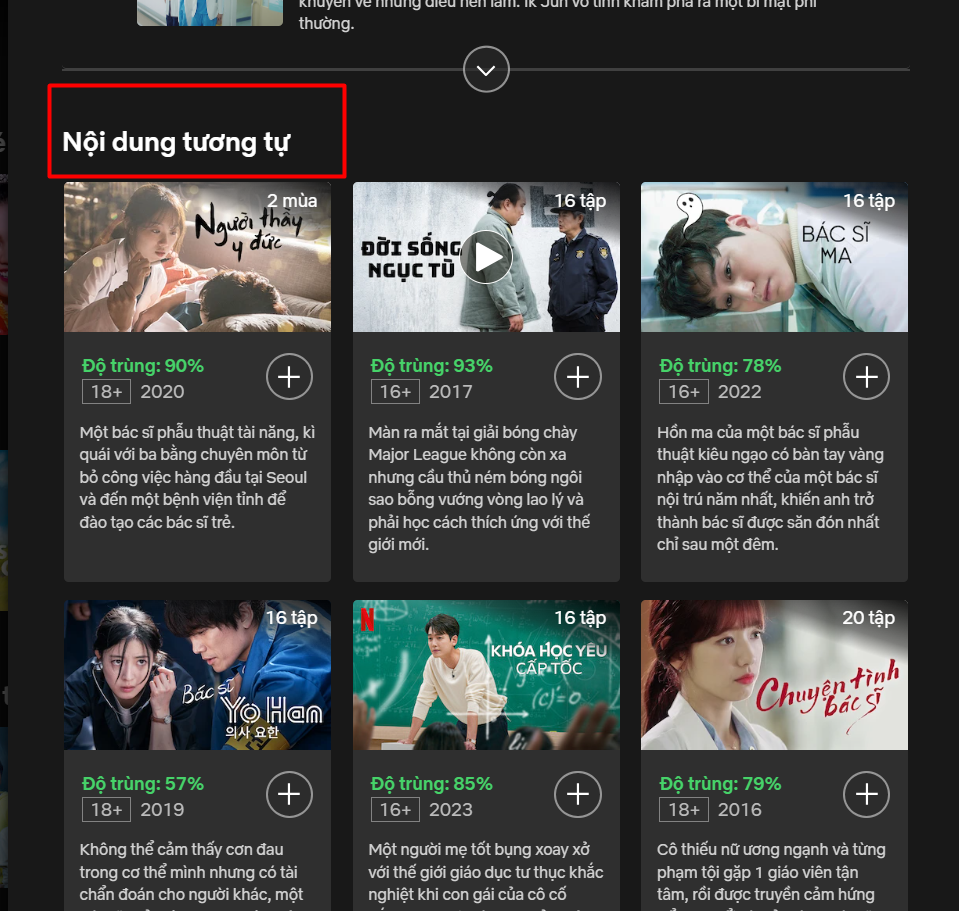
# TỔNG QUAN VỀ HỆ ĐỀ XUẤT (RECOMMENDER SYSTEM)

## Khái niệm Hệ đề xuất (Recommender system – RS)

Hệ đề xuất (Recommender System viết tắt là RS) là hệ thống sử dụng các kỹ thuật và công cụ phần mềm xử lý dữ liệu người dùng và dữ liệu về sản phẩm nhằm đáp ứng nhu cầu của người dùng về một sản phẩm hoặc dịch vụ nào đó [1]. Dựa trên cơ sở dữ liệu về hành vi tiêu dùng trong quá khứ, hệ thống thực hiện việc phân tích thông tin về sở thích và thị hiếu của người dùng. Từ đó, hệ thống cung cấp các gợi ý để hỗ trợ người dùng trong việc lựa chọn sản phẩm và dịch vụ phù hợp với sở thích của họ.

Hệ thống đề xuất là một công cụ phần mềm kỹ thuật, cung cấp các đề xuất về các đối tượng có thể hữu ích với người dùng, như việc gợi ý bộ phim nên xem, sản phẩm nên mua, bài hát nên nghe, hoặc tin tức nên đọc dựa trên quyết định của người dùng.

Ví dụ, hệ đề xuất phim của Netflix:



Hình . Hệ thống đề xuất phim Netflix

## Các lĩnh vực ứng dụng của Hệ đề xuất

Hiện tại hệ đề xuất được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau, ví dụ như:

* Thương mại điện tử: gợi ý sản phẩm tương tự hoặc phù hợp với sở thích của người dùng dựa trên lịch sử mua hàng, đánh giá và thông tin cá nhân như Amazon, Ebay.
* Giáo dục: gợi ý nguồn tài nguyên học tập như sách, bài báo, khóa học, trang web học tập cho người dùng thông qua các hệ thống của Foxtrot, InfoFinder.
* Giải trí:

+ Gợi ý các bộ phim và chương trình truyền hình dựa trên lịch sử xem phim, thể loại yêu thích và đánh giá từ người dùng như Netflix, MovieLens, EachMovie.

+ Gợi ý các bài hát, album và danh sách phát dựa trên thể loại yêu thích, lịch sử nghe nhạc và thông tin cá nhân của người dùng như LastFM, Stopify.

+ Gợi ý các video tương tự hoặc phù hợp với sở thích của người dùng dựa trên lịch sử xem video, đánh giá và các thông tin khác như YouTube.

+ Gợi ý các việc làm phù hợp với hồ sơ, kỹ năng và sở thích của người dùng như LinkedIn, TopCV.

## Phát biểu bài toán Hệ đề xuất

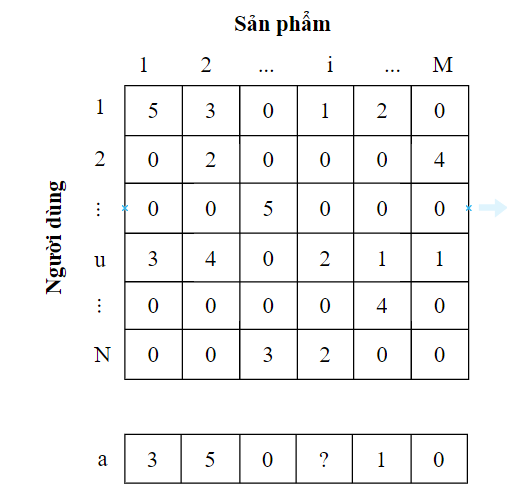
Hai đối tượng được sử dụng trong bài toán là người dùng (user) và sản phẩm (item). Người dùng (user) là những cá nhân sử dụng hệ thống để thực hiện các thao tác mua bán, giao dịch, xem, đánh giá. Sản phẩm (item) đề cập đến các mặt hàng được giao dịch như hàng hóa, phim ảnh, sách, báo, nhạc…

Bài toán được đặt ra như sau: Cho tập hợp hữu hạn gồm N người dùng U = {u1, u2, …, uN} và M sản phẩm P = {p1, p2, …, pM}.

Mỗi người dùng ui ∈ U (với i = 1, 2, …, N) được biểu diễn bởi |T| đặc trưng nội dung T = {t1, t2, …, t|T|}. Các đặc trưng tq ∈ T thông thường là thông tin cá nhân của người dùng. Ví dụ ui ∈U là một người dùng thì các đặc trưng nội dung biểu diễn người dùng ui có thể là T = {giới tính, tuổi, nghề nghiệp, …}.

Mỗi sản phẩm px ∈ P (với x = 1, 2, …, M) có thể là phim, ảnh, sách, báo, hàng hóa hoặc bất kỳ thông tin nào mà người dùng cần. Mỗi sản phẩm px ∈ P được biểu diễn thông qua |C| = {c1, c2, …, c|C|}. Các đặc trưng cs ∈ C nhận được từ các phương pháp trích chọn đặc trưng trong lĩnh vực truy vấn thông tin. Ví dụ px ∈ P là một phim thì các đặc trưng nội dung biểu diễn phim px có thể là C = {thể loại phim, tiêu đề, nước sản xuất, diễn viên, đạo diễn, …}.

Mối quan hệ giữa tập sản phẩm P và tập người dùng U được biểu diễn thông qua trận đánh giá R = [rix] với i = 1, 2, …, N; x = 1, 2, …, M.



Hình . Ví dụ ma trận đánh giá tổng quát

Giá trị rix thể hiện đánh giá của người dùng ui ∈ U cho một số sản phẩm px ∈ P. Thông thường giá trị rix nhận một số giá trị thuộc miền F = {1, 2, …, g} được thu thập trực tiếp bằng cách hỏi ý kiến người dùng hoặc thu thập gián tiếp thông qua cơ chế phản hồi của người dùng [2]. Những giá trị rix = 0 được hiểu là người dùng ui ∈ U chưa biết đến hoặc không đánh giá sản phẩm px ∈ P, những ô điền ký tự “?” là giá trị cần hệ đề xuất đưa ra dự đoán đánh giá. Tiếp đến, ta ký hiệu Pi ⊆ P là tập các sản phẩm px ∈ P được đánh giá bởi người dùng ui ∈ U và ua ∈ U được gọi là người dùng hiện tại, người dùng cần được tư vấn hay người dùng tích cực.

Khi đó, tồn tại hai dạng bài toán điển hình của hệ đề xuất là:

* Đưa ra dự đoán về đánh giá của người dùng ua cho những sản phẩm chưa được đánh giá.
* Đưa ra gợi ý danh sách một số các sản phẩm phù hợp với sở thích của người dùng hiện tại. Cụ thể đối với người dùng ua, hệ tư vấn sẽ đưa ra K sản phẩm mới px ∈ (P\Pa) phù hợp với người dùng ua để đề xuất cho họ.

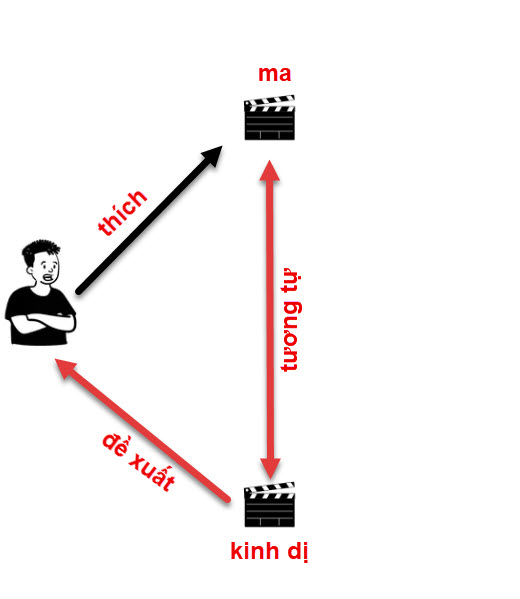
## Các phương pháp xây dựng bài toán Hệ đề xuất

Các giải thuật trong RS có rất nhiều, nhưng ta có thể phân thành các nhóm chính cơ bản sau đây [3]:

* Nhóm giải thuật lọc theo nội dung (content-based filtering - CBF)
* Nhóm giải thuật lọc cộng tác (collaborative filtering - CF)
* Nhóm giải thuật lọc kết hợp (Hybrid filtering - HF)

### Hệ đề xuất sử dụng lọc theo nội dung

Lọc nội dung là một kỹ thuật được sử dụng để gợi ý sản phẩm cho khách hàng của một hệ thống theo mức độ tương đồng của các mặt hàng. Nó thường được áp dụng trong hệ thống gợi ý để tìm các mục tương tự với những mục mà người dùng đã thể hiện sự quan tâm. Nội dung được sử dụng để xác định mức độ giống nhau của các mục có thể là bất kỳ thứ gì từ văn bản, hình ảnh đến xếp hạng. Mục tiêu của người dùng là hệ thống đề xuất dựa trên các đề xuất sàng lọc nội dung cho các sản phẩm tương tự với những sản phẩm mà họ đã bày tỏ sự quan tâm.



Hình . Hệ đề xuất sử dụng lọc theo nội dung

*Các bước thực hiện lọc theo nội dung:*

Bước 1.Biểu diễn nội dung của đối tượng khuyến nghị p ∈ P [4], được ký hiệu là Content(p), thông qua tập |C| đặc trưng nội dung của p. Tập các đặc trưng của sản phẩm p được xây dựng bằng các kỹ thuật truy vấn thông tin.

Bước 2.Mô hình hóa sở thích người dùng u ∈ U, gọi là hồ sơ người dùng, ký hiệu là UserProfile (u) [4]. Hồ sơ của người dùng u thực chất là lịch sử truy cập hoặc đánh giá của người đó đối với các đặc trưng nội dung sản phẩm. UserProfile (u) được xây dựng bằng cách phân tích nội dung các sản phẩm mà người dùng đã từng truy nhập hoặc đánh giá dựa trên các kỹ thuật truy vấn thông tin. Do vậy, UserProfile (u) là một véc tơ biểu diễn thông qua |C| đặc trưng nội dung của P.

Bước 3.Dự đoán đánh giá của người dùng u với sản phẩm p dựa trên độ tương tự nội dung của p với hồ sơ người dùng u. Hệ thống sẽ ưu tiên tư vấn những đối tượng p có nội dung tương tự cao nhất với hồ sơ người dùng u.

Để đánh giá độ tương đồng của của đối tượng p với hồ sơ người dùng u, hệ thống so sánh bằng consine góc lệch giữa 2 vector u và p:

Similarity = cos (θ) = = (2-1)

*Ưu điểm:*

* Hệ thống không yêu cầu có quá nhiều dữ liệu từ những người dùng khác để đạt được độ chính xác đề xuất.
* Có thể đề xuất các nội dung mới, phù hợp với sở thích của người dùng, ngay cả khi người dùng chưa từng tương tác với nội dung đó trước đây. Điều này là do hệ thống đề xuất lọc nội dung có thể phân tích các đặc trưng của nội dung để xác định các nội dung có thể phù hợp với người dùng.
* Có thể đề xuất các nội dung đa dạng, không bị giới hạn bởi các nội dung mà người dùng đã tương tác trước đây. Điều này là do hệ thống đề xuất lọc nội dung có thể phân tích các đặc trưng của nội dung một cách toàn diện, không chỉ dựa trên các tương tác của người dùng.
* Có thể được sử dụng cho nhiều loại nội dung khác nhau, bao gồm văn bản, hình ảnh, video, âm nhạc,... Điều này là do hệ thống đề xuất lọc nội dung có thể phân tích các đặc trưng của nội dung một cách linh hoạt, không phụ thuộc vào loại nội dung cụ thể.

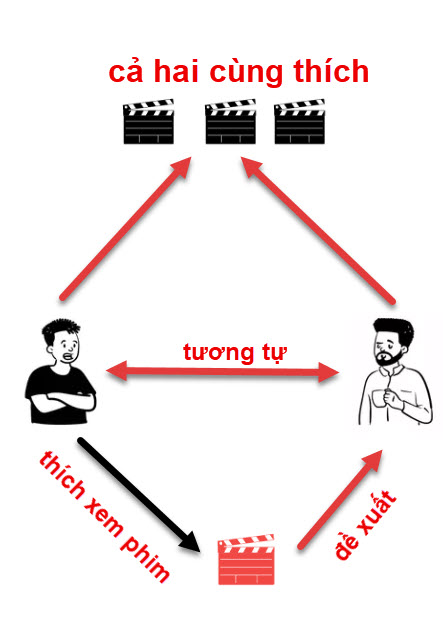
*Nhược điểm:*

* Có thể đề xuất các nội dung không phù hợp với sở thích của người dùng, đặc biệt là đối với các nội dung mới, chưa có nhiều thông tin. Điều này là do hệ thống đề xuất lọc nội dung chỉ dựa trên thông tin về nội dung của các đối tượng được đề xuất. Tuy nhiên, thông tin về nội dung có thể không chính xác hoặc không đầy đủ, dẫn đến việc đưa ra các gợi ý không phù hợp.
* Có thể đề xuất các nội dung quá giống nhau, dẫn đến sự nhàm chán cho người dùng. Điều này là do hệ thống đề xuất lọc nội dung thường dựa trên các đặc trưng của nội dung để xác định các nội dung có thể phù hợp với người dùng. Nếu các đặc trưng của nội dung tương tự nhau, thì hệ thống có thể đề xuất các nội dung quá giống nhau cho người dùng.
* Có thể khó khăn trong việc xác định các đặc trưng của nội dung, đặc biệt là đối với các loại nội dung phức tạp hoặc mới. Điều này có thể làm giảm độ chính xác của các gợi ý do hệ thống đưa ra.

### Hệ đề xuất sử dụng lọc cộng tác

Hệ thống lọc cộng tác dựa trên việc thu thập và phân tích thông tin về hành vi, hoạt động hoặc sở thích của người dùng để dự đoán những gì họ sẽ thích dựa trên sự tương đồng với người dùng khác.

Trong hệ thống gợi ý, người ta thường biểu diễn các đánh giá của người dùng cho các sản chúng ta có một hệ thống dự đoán các đánh giá của người dùng lên các sản phẩm mới, và dựa vào đó ta có thể lấy ra N sản phẩm được dự đoán có trọng số cao nhất. Cuối cùng, hai vấn đề quan trọng nhất đó là làm thế nào để tìm được tập K người dùng tương tự với người dùng u nhất dựa trên thị hiếu về sản phẩm hay làm thế nào để tìm ra tập K sản phẩm tương tự với sản phẩm i nhất để có thể đưa ra các dự đoán trọng số đánh giá của người dùng và sản phẩm tương ứng.



Hình . Hệ đề xuất sử dụng lọc cộng tác

#### Lọc cộng tác dựa trên bộ nhớ (Memory-based Collaborative Filtering)

Thuật toán lọc cộng tác dựa trên bộ nhớ là dựa trên giá trị xếp hạng của người dùng trong ma trận Người dùng – Sản phẩm, hệ thống tính toán độ tương đồng giữa người dùng hiện tại với những người dùng tương tự để đưa ra đề xuất sản phẩm cho người dùng [1].

Lọc cộng tác theo bộ nhớ được chia thành hai loại chính: Lọc cộng tác dựa trên người dùng và Lọc cộng tác dựa trên nội dung.

*Ưu điểm của lọc cộng tác theo bộ nhớ:*

* Độ chính xác cao: Lọc cộng tác theo bộ nhớ có thể đạt được độ chính xác cao hơn so với các phương pháp đề xuất khác, chẳng hạn như lọc nội dung. Điều này là do lọc cộng tác theo bộ nhớ sử dụng dữ liệu tương tác của người dùng để tìm ra các quy luật chung về sở thích của người dùng. Các quy luật này có thể được sử dụng để đưa ra các gợi ý chính xác hơn cho người dùng.
* Dễ hiểu: Lọc cộng tác theo bộ nhớ là một phương pháp đề xuất khá đơn giản và dễ hiểu. Các nhà phát triển có thể dễ dàng triển khai và bảo trì các hệ thống lọc cộng tác theo bộ nhớ.
* Hiệu quả: Lọc cộng tác theo bộ nhớ có thể hoạt động hiệu quả với một lượng dữ liệu tương tác tương đối nhỏ. Điều này làm cho lọc cộng tác theo bộ nhớ phù hợp với các ứng dụng có lượng người dùng lớn hoặc có lượng dữ liệu tương tác hạn chế.

*Nhược điểm của lọc cộng tác theo bộ nhớ*

* Có thể đề xuất các nội dung không phù hợp với sở thích của người dùng mới, đặc biệt là khi người dùng này chưa tương tác với bất kỳ nội dung nào trong hệ thống. Điều này là do lọc cộng tác theo bộ nhớ dựa trên dữ liệu tương tác của người dùng để tìm ra các quy luật chung về sở thích của người dùng. Nếu người dùng mới chưa tương tác với bất kỳ nội dung nào trong hệ thống, thì hệ thống sẽ không có dữ liệu để tìm ra các quy luật về sở thích của người dùng này.
* Có thể đề xuất các nội dung quá giống nhau, dẫn đến sự nhàm chán cho người dùng. Điều này là do lọc cộng tác theo bộ nhớ thường dựa trên các điểm tương đồng giữa các người dùng và các nội dung để đưa ra gợi ý. Nếu các người dùng có sở thích tương tự nhau, thì hệ thống có thể đề xuất các nội dung quá giống nhau cho các người dùng này.
* Có thể khó khăn trong việc xử lý các dữ liệu tương tác lớn, dữ liệu tương tác quá thưa thớt hoặc phức tạp. Điều này có thể làm giảm độ chính xác của các gợi ý do hệ thống đưa ra.

a, Lọc cộng tác dựa trên người dùng

Phương pháp lọc cộng tác theo bộ nhớ dựa vào người dùng sử dụng toàn bộ ma trận đánh giá để chọn ra một tập người dùng tương tự nhất với người dùng hiện tại, sau đó kết hợp các đánh giá của tập những người dùng tương tự nhất này để dự đoán đánh giá cho người dùng hiện tại với sản phẩm chưa biết. Phương pháp được thực hiện theo bốn bước:

Bước 1. Tính toán mức độ tương đồng của tất cả người dùng trong hệ thống với người dùng hiện tại.

Bước 2. Xác định tập người dùng láng giềng gần bằng việc lựa chọn những người dùng có mức độ tương tự cao nhất với người dùng hiện tại.

Bước 3. Đưa ra dự đoán của người dùng hiện tại với các sản phẩm chưa đánh giá bằng việc kết hợp tập đánh giá của các người dùng trong tập láng giềng.

Bước 4. Tư vấn tập sản phẩm mới có mức độ phù hợp nhất cho người dùng hiện tại.

b, Lọc cộng tác dựa trên nội dung

Phương pháp này thực hiện bằng cách tính toán mức độ tương tự giữa sản phẩm cần dự đoán đánh giá bởi người dùng với các sản phẩm đã được đánh giá thay vì tính mức độ tương tự giữa các người dùng trong hệ thống với người dùng hiện tại. Việc tính toán mức độ tương tự giữa hai sản phẩm dựa vào tập người dùng cùng đánh giá cả hai sản phẩm đó. Sau đó, chọn ra một tập sản phẩm láng giềng với sản phẩm cần dự đoán đánh giá. Kết hợp các đánh giá của tập sản phẩm láng giềng này để đưa ra dự đoán đánh giá của sản phẩm cần dự đoán.

#### *Lọc* cộng tác dựa trên mô hình (Model-based Collaborative Filtering)

Để khắc phục nhược điểm về thời gian tính toán và yêu cầu dung lượng bộ nhớ lớn, người ta đã phân biệt rõ ràng hai tác vụ chính trong phương pháp lọc cộng tác: một là tác vụ xây dựng mô hình từ dữ liệu huấn luyện, hai là tác vụ tính toán kết quả khuyến nghị từ mô hình nhận được.

Cụ thể, hệ thống RS sử dụng các thuật toán học máy có giám sát hoặc không có giám sát để huấn luyện mô hình để rút gọn mô phỏng sở thích của người dùng. Các thuật toán học máy có thể là cây quyết định, bộ phân loại Bayes, hồi quy, máy vector hỗ trợ (support vector machines), mạng noron... Sau khi nhận được mô hình từ bước huấn luyện, hệ thống RS sử dụng trực tiếp mô hình này để đưa ra kết quả khuyến nghị. [1]

*Ưu điểm của lọc theo mô hình so với lọc theo bộ nhớ:*

* Độ chính xác cao hơn đối với người dùng mới: Lọc theo mô hình có thể học được các quy luật về sở thích của người dùng, ngay cả khi người dùng này chưa tương tác với bất kỳ nội dung nào trong hệ thống. Điều này làm cho lọc theo mô hình phù hợp hơn với các ứng dụng có lượng người dùng mới lớn, chẳng hạn như các nền tảng thương mại điện tử hoặc các nền tảng phát trực tuyến.
* Đề xuất các nội dung đa dạng hơn: Lọc theo mô hình có thể học được các mối quan hệ phức tạp giữa các người dùng và các nội dung. Điều này làm cho lọc theo mô hình có thể đề xuất các nội dung đa dạng hơn cho người dùng, giúp người dùng khám phá được các nội dung mới và thú vị.
* Khả năng xử lý dữ liệu lớn và phức tạp: Lọc theo mô hình có thể xử lý các dữ liệu lớn và phức tạp một cách hiệu quả. Điều này làm cho lọc theo mô hình phù hợp với các ứng dụng có lượng dữ liệu tương tác lớn hoặc phức tạp.

### Hệ đề xuất sử dụng lọc kết hợp/lai ghép (Hybrid Filtering)

Lọc theo nội dung khai thác những khía cạnh liên quan đến các đặc trưng nội dung thông tin của những đối tượng cần lọc. Trái lại, lọc cộng tác khai thác những khía cạnh liên quan đến thói quen sử dụng các loại thông tin khác nhau của mỗi người dùng.

Mỗi phương pháp đều có những thế mạnh và hạn chế nhất định, do vậy để phát huy điểm mạnh và hạn chế những điểm yếu của từng kỹ thuật tư vấn riêng lẻ, các phương pháp lọc kết hợp được đưa ra nhằm cải thiện hiệu quả tư vấn sản phẩm mới phù hợp tới người dùng [2].

Nhiều nghiên cứu so sánh đã chứng minh rằng phương pháp kết hợp lọc cho kết quả dự đoán tốt hơn so với phương pháp lọc cộng tác và lọc nội dung thuần túy. Đặc biệt, lọc kết hợp giảm thiểu tác động của vấn đề dữ liệu thưa và người dùng mới.

*Một số phương pháp lai ghép:*

* Có trọng số (Weighted): Các điểm số của các thành phần đề xuất khác nhau được kết hợp dựa trên số lượng.
* Chuyển đổi (Switching): Hệ thống chọn giữa các thành phần đề xuất và áp dụng hệ thống đã chọn.
* Hỗn hợp (Mixed): Các đề xuất từ những người giới thiệu khác nhau được kết hợp để đưa ra đề xuất.
* Kết hợp tính năng (Feature Combination): Các tính năng từ các nguồn tri thức khác nhau được kết hợp và đưa ra cho một thuật toán gợi ý duy nhất.
* Tính năng tăng cường (Feature Augmentation): Một kỹ thuật gợi ý được sử dụng để tính toán một tính năng hoặc tập hợp các tính năng, sau đó trở thành một phần của đầu vào cho kỹ thuật tiếp theo.
* Cascade: Các khuyến nghị được ưu tiên nghiêm ngặt, với những ưu tiên thấp hơn phá vỡ các mối quan hệ trong việc tính điểm của những người cao hơn.
* Cấp độ meta (Meta-level): Một kỹ thuật đề xuất được áp dụng và tạo ra một số loại mô hình, sau đó trở thành đầu vào được sử dụng bởi kỹ thuật tiếp theo.

## Thuật toán ****K-Nearest Neighbors (KNN)****

### Định nghĩa

K-Nearest Neighbors (KNN) là một thuật toán học máy được sử dụng chủ yếu cho các nhiệm vụ phân loại và dự đoán. Ý tưởng cơ bản của KNN là dự đoán nhãn của một điểm dữ liệu mới dựa trên các điểm dữ liệu gần nhất trong tập dữ liệu huấn luyện.

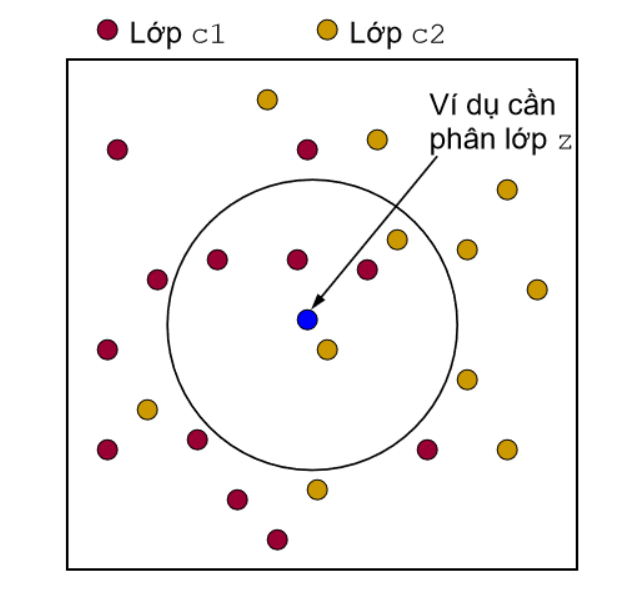
Cụ thể, trong quá trình huấn luyện, KNN giữ lại toàn bộ tập dữ liệu. Khi có một điểm dữ liệu mới cần phân loại, thuật toán sẽ tìm ra K điểm dữ liệu gần nhất dựa trên một khoảng cách được đo lường, thường là khoảng cách Euclidean.

Sau đó, nhãn của điểm dữ liệu mới sẽ được quyết định bằng cách sử dụng đa số hoặc trung bình của những nhãn của K điểm dữ liệu gần nhất đó.

### Nguyên lý hoạt động

Nguyên lý hoạt động của thuật toán K-Nearest Neighbors (KNN) là dựa trên giả định rằng các điểm dữ liệu có tính chất tương đồng với nhau thường nằm gần nhau trong không gian đặc trưng.

Khi có một điểm dữ liệu mới, KNN sẽ tìm ra các điểm dữ liệu gần nó nhất trong không gian đặc trưng, và sử dụng nhãn của các điểm dữ liệu này để dự đoán nhãn của điểm dữ liệu mới.



Hình . Ví dụ minh họa thuật toán KNN

*Các bước xây dựng mô hình KNN và đánh giá hiệu quả của mô hình:*

Bước 1: Chuẩn bị bộ dữ liệu để huấn luyện và rút trích đặc trưng của nó

Giai đoạn này được coi là giai đoạn quan trọng trong các bài toán về Machine Learning, đây là input cho việc xây dựng mô hình bài toán. Lựa chọn ra những đặc trưng tốt của bộ dữ liệu và lược bỏ những đặc trưng không tốt của dữ liệu gây nhiễu. Các dữ liệu không cần thiết gây khó khăn cho việc tính toán thì có thể lược bớt nhưng vẫn giữ được độ chính xác của dữ liệu.

Chuẩn bị bộ dữ liệu để test trên mô hình. Thông thường sẽ sử dụng cross-validation để chia tập dataset thành 2 phần, một phần dữ liệu phục vụ cho quá trình training và phần dữ liệu còn lại sử dụng cho mục đích test mô hình.

Bước 2: Xây dựng mô hình phân lớp

Mục đích của mô hình huấn luyện là tìm ra hàm F (x). Hàm F tìm được để gán nhãn cho dữ liệu. Bước này gọi là training hay học.

(2-2)

Trong đó: x là các input dữ liệu đầu vào

y là nhãn dán lớp hay đầu ra

Bước 3: Kiểm tra dữ liệu với mô hình

Sau khi training đã tìm ra được mô hình phân lớp, đến bước này sẽ đưa phần dữ liệu testing còn lại vào để kiểm tra trên mô hình phân lớp.

Bước 4: Đánh giá mô hình phân lớp xây dựng và lựa chọn mô hình tốt nhất

Cuối cùng, chúng ta sẽ đánh giá mô hình bằng cách kiểm tra mức độ lỗi của dữ liệu testing và dữ liệu training thông qua mô hình đã tìm được. Nếu không đạt kết quả mong muốn, chúng ta cần thay đổi các tham số của thuật toán học để tìm ra các mô hình tốt hơn, sau đó kiểm tra và đánh giá lại mô hình phân lớp và cuối cùng chọn ra mô hình phân lớp tốt nhất cho bài toán của chúng ta.

Bản chất của KNN chính là tính toán khoảng cách từ điểm cần xét đến k điểm lân cận, nên nếu các features có scale khác nhau, hoặc có đơn vị khác nhau, ta có thể thực hiện chuẩn hoá (normalize) dữ liệu để có thể đẩy mạnh hiệu suất của thuật toán. Tùy thuộc vào mục đích phân loại hay hồi quy mà đầu ra của KNN sẽ khác nhau:

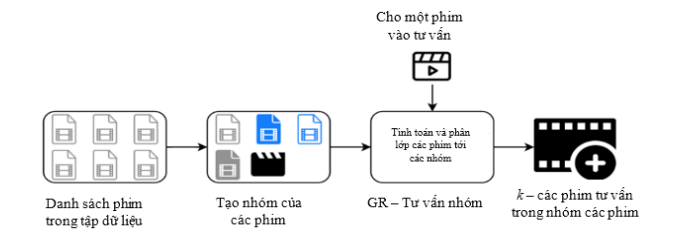
* Với bài toán phân loại, đầu ra của KNN là lớp của bản ghi cần phân loại. Một bản ghi được xác định lớp theo nguyên tắc biểu quyết đa số. Theo đó, lớp của bản ghi được xác định theo k bản ghi gần nhất với bản ghi 16 đang được xét theo nguyên tắc số đông với k đảm bảo điều kiện là số nguyên dương, thường đủ nhỏ.
* Với bài toán hồi quy, đầu ra của KNN chính là giá trị dự đoán của của hàm mất mát tại bản ghi đang được xét. Giá trị dự đoán này là trung bình nhãn của k mẫu gần nhất. Dữ liệu huấn luyện là tập hợp những điểm dữ liệu đã được gắn nhãn (với bài toán phân lớp), và tập các điểm dữ liệu với giá trị tương ứng (với bài toán hồi quy). Một đặc trưng của thuật toán KNN chính là sự nhạy cảm với hình dạng của dữ liệu.

### Thuật toán KNN trong bài toán đề xuất phim

Bước 1. Thực hiện nhập danh sách phim đã được đánh giá để tạo nhóm các phim.

Bước 2. Tạo GR, trong đó thực chất là tính toán và phân lớp các phim về các nhóm phim.

Bước 3. Sử dụng phương pháp k-NN để lựa chọn k phim tư vấn từ nhóm các phim [5].



Hình . Các bước đưa ra đề xuất phim bằng thuật toán KNN

### Ưu, nhược điểm của thuật toán KNN

*Ưu điểm:*

* Đơn giản và dễ hiểu: KNN là một thuật toán đơn giản và dễ hiểu, không cần phải tinh chỉnh nhiều siêu tham số.
* Không cần huấn luyện: KNN không cần quá trình huấn luyện phức tạp, chỉ cần lưu trữ dữ liệu huấn luyện và tính toán khoảng cách khi cần dự đoán.
* Hiệu suất tốt với dữ liệu lớn: KNN thường hoạt động tốt với dữ liệu lớn và không cần phải tạo mô hình phức tạp.

*Nhược điểm:*

* Tính toán phức tạp: KNN cần tính toán khoảng cách giữa điểm dữ liệu và tất cả các điểm dữ liệu huấn luyện, điều này có thể trở nên phức tạp khi dữ liệu lớn.
* Nhạy cảm với nhiễu: KNN có thể bị ảnh hưởng bởi dữ liệu nhiễu hoặc dữ liệu không chuẩn, dẫn đến dự đoán không chính xác.
* Cần lưu trữ toàn bộ dữ liệu huấn luyện: KNN cần lưu trữ toàn bộ dữ liệu huấn luyện trong bộ nhớ, điều này có thể tạo ra vấn đề khi xử lý dữ liệu lớn.

## Thuật toán Singular Value Decomposition (SVD)

### Định nghĩa

Singular Value Decomposition (SVD) một phương pháp từ đại số tuyến tính thường được sử dụng như một kỹ thuật giảm kích thước trong học máy, là một trong những phương pháp thuộc nhóm matrix factorization được phát triển lần đầu bởi những nhà hình học vi phân. Ban đầu mục đích của phương pháp này là tìm ra một phép xoay không gian sao cho tích vô hướng của các vector không thay đổi. Từ mối liên hệ này, khái niệm về ma trận trực giao đã hình thành để tạo ra các phép xoay đặc biệt.

Phương pháp SVD đã được phát triển dựa trên những tính chất của ma trận trực giao và ma trận đường chéo để tìm ra một ma trận xấp xỉ với ma trận gốc. Phương pháp này sau đó đã được ứng dụng rộng rãi trong các lĩnh vực như hình học vi phân, hồi qui tuyến tính, xử lý hình ảnh, clustering, các thuật toán nén và giảm chiều dữ liệu, và đặc biệt đặc biệt hiệu quả trong các bài toán recommendation. Nó cũng tăng tốc độ tạo ra kết quả và mang lại kết quả đáng tin cậy và hiệu quả. [6]

### Nguyên lý hoạt động

SVD là phương pháp giúp phân rã bất cứ ma trận nào thành tích của 3 ma trận với tính chất đặc biệt [7].

(2-3)

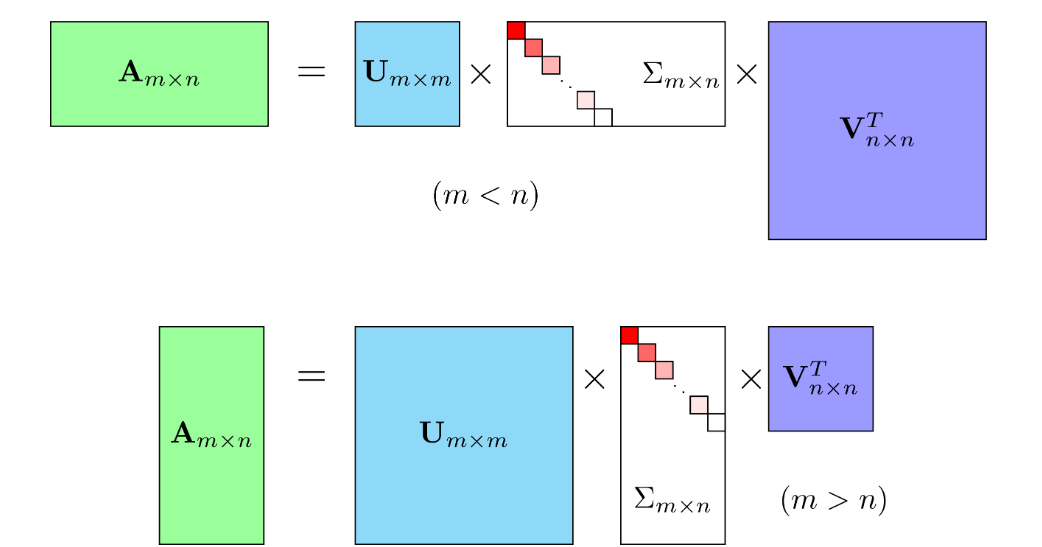
*Trong đó:*

U là ma trận m xm vec-tơ riêng phải và U là ma trận trực giao.

Σ là ma trận đường chéo n x m chứa các giá trị riêng của ma trận A.

V là ma trận n x n vec-tơ riêng trái và V cũng là ma trận trực giao.

Hình mô tả SVD của ma trận Amxn trong hai trường hợp: m < n và m > n. Trường hợp m = n có thể xếp vào một trong hai trường hợp trên.



Hình . SVD cho ma trận A khi: m < n và m > n

Σ là một ma trận đường chéo với các phần tử trên đó giảm dần và không âm. Màu đỏ càng đậm thể hiện giá trị càng cao. Các ô màu trắng trên ma trận này thể hiện giá trị 0.

### Thuật toán SVD trong bài toán đề xuất phim

SVD xây dựng một ma trận với cột là user và hàng là item (phim) và các phần tử đưa ra bởi xếp hạng của người dùng. SVD phân tích một ma trận thành ba ma trận khác và trích xuất các yếu tố từ việc phân tích nhân tử của ma trận cao cấp (user-item-đánh giá)

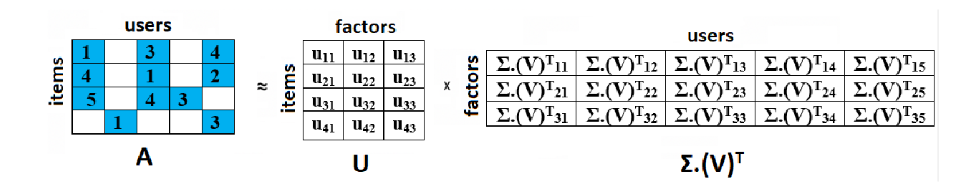
(2-4)

Ma trận U: ma trận suy biến của item x

Ma trận Σ: ma trận đường chéo

Ma trận VT: ma trận suy biến của user x

Sử dụng SVD trên ma trận xếp hạng, với các hàng đại diện cho người dùng và các cột đại diện cho phim. Ma trận thể hiện xếp hạng do người dùng đưa ra đối với phim. SVD tìm các yếu tố ma trận từ ma trận xếp hạng phim của người dùng [8]. Nó mang lại ma trận đánh giá tiện ích. Từ việc phân tích nhân tử ma trận, các biến tiềm ẩn cho thấy đặc điểm của item. Cuối cùng, ma trận uitility A được tạo ra với hình dạng m x n. Đầu ra cuối cùng của ma trận A cho thấy mối liên hệ giữa user và item bằng cách ánh xạ user và item vào không gian tiềm ẩn r chiều



Hình . Ma trận SVD giữa item và users

### Ưu, nhược điểm của thuật toán SVD

*Ưu điểm:*

* SVD giúp phân tích và tìm ra các thông tin quan trọng trong dữ liệu, giúp hiểu rõ hơn về các mối quan hệ và tương tác giữa các biến.
* SVD có thể giảm chiều dữ liệu một cách hiệu quả, giúp giảm thiểu sự phức tạp và tăng tốc độ tính toán.
* SVD có thể được áp dụng trong nhiều lĩnh vực khác nhau như xử lý ảnh, xử lý ngôn ngữ tự nhiên và hệ thống gợi ý.

*Nhược điểm:*

* SVD có thể gây ra mất thông tin do việc giảm chiều dữ liệu, dẫn đến việc mất đi sự chi tiết và chính xác trong dữ liệu.
* SVD yêu cầu tính toán phức tạp, đặc biệt khi áp dụng cho các ma trận lớn, có thể tạo ra áp lực lớn đối với hệ thống tính toán.
* SVD có thể gặp vấn đề khi xử lý dữ liệu thưa (sparse data), do đó cần phải sử dụng các phương pháp xử lý dữ liệu thưa trước khi áp dụng SVD.

## Độ đo đánh giá

### Độ đo tương tự Cosine similarity

Độ tương tự Cosine similarity là một cách [đo độ tương tự](https://vi.wikipedia.org/w/index.php?title=%C4%90o_%C4%91%E1%BB%99_t%C6%B0%C6%A1ng_t%E1%BB%B1&action=edit&redlink=1) (measure of similarity) giữa hai vectơ khác không của một [không gian tích vô hướng](https://vi.wikipedia.org/wiki/Kh%C3%B4ng_gian_t%C3%ADch_v%C3%B4_h%C6%B0%E1%BB%9Bng). Độ tương tự này được định nghĩa bằng giá trị [cosine](https://vi.wikipedia.org/wiki/Cosine) của góc giữa hai vectơ, và cũng là tích vô hướng của cùng các vectơ [đơn vị](https://vi.wikipedia.org/wiki/Vect%C6%A1_%C4%91%C6%A1n_v%E1%BB%8B) để cả hai đều có chiều dài 1. Giá trị cosine của 0° là 1, và bé hơn 1 với bất kỳ góc nào trong khoảng các [radian](https://vi.wikipedia.org/wiki/Radian) (0, π].

Công thức tính độ đo tương tự cosine:

Similarity = cos (θ) = (2-5)

Similarity có giá trị nằm trong khoảng [-1, -1]:

Giá trị −1 biểu thị không có mối quan hệ

Giá trị 0 biểu thị mối quan hệ độc lập tuyến tính

Giá trị 1 biểu thị mối quan hệ tương đồng

Từ đó, suy ra giá trị tương tự càng gần 1 thì độ tương tự giữa hai vec-to càng cao.

# ỨNG DỤNG THUẬT TOÁN KNN VÀ SVD XÂY DỰNG ĐỀ XUẤT PHIM

## Các công cụ và thư viện sử dụng

### Ngôn ngữ lập trình Python

Python là [ngôn ngữ lập trình bậc cao](https://topdev.vn/blog/ngon-ngu-lap-trinh-bac-cao/), mã nguồn mở và đa nền tảng. Python được Guido van Rossum giới thiệu vào năm 1991 và đã trải qua 3 giai đoạn phát triển khác nhau tương ứng với các phiên bản, mới nhất hiện nay là phiên bản Python 3.11.4.

*Những đặc điểm và tính năng nổi bật của python:*

Python được thiết kế với ưu điểm giúp người học dễ đọc, dễ hiểu và dễ nhớ; vì nó được xây dựng với mục tiêu loại bỏ sự phức tạp và chỉ giữ lại những gì cần thiết. Cấu trúc của Python cho phép người sử dụng viết mã lệnh với số lần nhập phím tối thiểu. Vì vậy so với các ngôn ngữ lập trình khác, chúng ta có thể sử dụng ít dòng mã hơn để viết ra một chương trình.

Python là một ngôn ngữ lập trình có tính linh hoạt và đa dạng ngôn ngữ, nó hỗ trợ cho phép tạo nguyên mẫu nhanh.

Python có sẵn trên nhiều nền tảng, bao gồm Windows, MacOS, Linux và các hệ thống UNIX khác nhau. Khả năng tương thích đa nền tảng này cho phép các nhà phát triển viết mã có thể được thực thi trên các hệ điều hành khác nhau mà không cần sửa đổi đáng kể. Nhờ vậy mà Python có thể làm được ứng dụng trong nhiều lĩnh vực khác nhau.

*Ngoài những tính năng nổi bật như trên python còn có một số hạn chế đáng kể:*

Tốc độ tính toán, thực thi chậm.

Sử dụng nhiều không gian bộ nhớ.

Hạn chế trong việc phát triển ứng dụng trên thiết bị di động.

Hạn chế trong việc thiết kế ứng dụng.



Hình . Ngôn ngữ lập trình Python

### Thư viện sử dụng

#### Thư viện Pandas

Pandas là một thư viện Python mã nguồn mở phổ biến để thao tác và phân tích dữ liệu. Nó cung cấp các cấu trúc dữ liệu và công cụ phân tích dữ liệu mạnh mẽ, làm cho nó trở thành một công cụ có giá trị để làm việc với dữ liệu có cấu trúc.

*Các cấu trúc dữ liệu chính được cung cấp bởi Pandas là:*

Series: Một đối tượng giống như mảng một chiều có thể chứa bất kỳ loại dữ liệu nào.

DataFrame: Một bảng dữ liệu hai chiều với các hàng và cột. Nó tương tự như bảng tính hoặc bảng SQL.

*Pandas cung cấp một loạt các chức năng thao tác và phân tích dữ liệu:*

Tải và lưu dữ liệu: Pandas có thể đọc dữ liệu từ nhiều định dạng tệp khác nhau, chẳng hạn như cơ sở dữ liệu CSV, Excel, SQL, v.v.

Làm sạch và tiền xử lý dữ liệu: Pandas cung cấp các công cụ để xử lý dữ liệu bị thiếu, loại bỏ các dữ liệu trùng lặp, chuyển đổi loại dữ liệu và xử lý các giá trị ngoại lai.

Thao tác dữ liệu: Pandas cho phép lọc, sắp xếp, nhóm và tổng hợp dữ liệu. Nó cũng cung cấp khả năng lập chỉ mục và cắt linh hoạt để truy cập và sửa đổi dữ liệu.

Trực quan hóa dữ liệu: Mặc dù Pandas chủ yếu là một thư viện thao tác dữ liệu, nhưng nó tích hợp tốt với các thư viện khác như Matplotlib và Seaborn.

Phân tích chuỗi thời gian: Pandas hỗ trợ mạnh mẽ để làm việc với dữ liệu chuỗi thời gian, bao gồm xử lý lập chỉ mục dựa trên thời gian, lấy mẫu lại và chuyển đổi tần suất.

#### Thư viện Numpy

NumPy (Python số) là một thư viện Python cơ bản cho tính toán khoa học. Nó cung cấp một đối tượng mảng N chiều mạnh mẽ, cùng với một tập hợp các hàm để thao tác và vận hành trên các mảng này một cách hiệu quả. NumPy tạo nền tảng cho nhiều thư viện Python liên quan đến dữ liệu và khoa học khác.

*Một số chức năng chính của thư viện NumPy:*

Mảng N chiều: Đối tượng ndarray trong NumPy là một cấu trúc dữ liệu mạnh mẽ có thể biểu diễn các mảng có bất kỳ chiều nào.

Các hàm toán học: NumPy cung cấp một loạt các hàm toán học hoạt động theo từng phần tử trên các mảng, giúp dễ dàng thực hiện các phép tính toán học trên các mảng dữ liệu.

Lập chỉ mục và cắt mảng: NumPy cung cấp khả năng lập chỉ mục và cắt linh hoạt để truy cập và sửa đổi các phần tử, mảng con hoặc các phần cụ thể của một mảng.

Tích hợp với các thư viện khác: NumPy tích hợp liền mạch với các thư viện Python liên quan đến dữ liệu và khoa học khác

#### Thư viện SciPy

SciPy là phần mềm nguồn mở miễn phí của Python cho toán học, khoa học và kỹ thuật. Thư viện SciPy được xây dựng dựa trên thư viện NumPy, cung cấp thao tác mảng N chiều thuận tiện và nhanh chóng. SciPy gồm các gói con (submodule) cho đại số tuyến tính, tối ưu hóa, tích hợp và thống kê. NumPy và SciPy rất dễ sử dụng, mạnh mẽ và được nhiều nhà khoa học và kỹ sư hàng đầu thế giới lựa chọn.

SciPy là thư viện cơ sở. Nó được xây dựng trên phần mở rộng NumPy. Không cần nhập NumPy nếu bạn đã nhập SciPy. SciPy tương thích với đối tượng mảng N chiều của NumPy. SciPy bao gồm mã cho hoạt động của các hàm NumPy. SciPy và NumPy cùng là sự lựa chọn tốt nhất cho các hoạt động khoa học.

*Một số chức năng chính của thư viện Scipy:*

Trong ngữ cảnh của việc tạo ma trận thưa thớt của bài toán, SciPy cung cấp các công cụ để thực hiện ma trận thưa thớt (sparse matrix) thông qua module scipy.sparse. Ma trận thưa thớt là một cách lưu trữ ma trận mà chỉ lưu trữ các phần tử khác không (non-zero elements), giúp tiết kiệm bộ nhớ và tăng tốc độ tính toán trong các ứng dụng lớn. SciPy cung cấp các công cụ để tạo, lưu trữ, và thực hiện các phép toán trên ma trận thưa thớt (sparse matrix).

Xử lý ảnh: SciPy cung cấp các hàm và thuật toán để xử lý ảnh, chẳng hạn như lọc ảnh, chuyển đổi ảnh, ...

Xử lý tín hiệu: SciPy cung cấp các hàm và thuật toán để xử lý tín hiệu, chẳng hạn như lọc tín hiệu, biến đổi Fourier, ...

Xử lý dữ liệu: SciPy cung cấp các hàm và thuật toán để xử lý dữ liệu, chẳng hạn như phân cụm, phân loại, hồi quy, ...

#### Thư viện Scikit-learn

Scikit-learn là một thư viện học máy phổ biến và mạnh mẽ, hỗ trợ nhiều loại thuật toán cũng như các công cụ để trực quan hóa [Machine Learning](https://www.thegioimaychu.vn/blog/thuat-ngu/machine-learning/?utm_source=tmblog&utm_medium=inlinekwd), tiền xử lý, điều chỉnh, lựa chọn và đánh giá mô hình. Dựa trên NumPy, SciPy và Matplotlib, Scikit-learn có một số thuật toán hiệu quả để phân loại, phân tích hồi quy và phân cụm.

Bao gồm các vector machines, rain forests, gradient boosting, k-means, và DBSCAN. Scikit-learn được biết đến với tính dễ phát triển tương đối nhờ các API được thiết kế nhất quán và hiệu quả, tài liệu mở rộng cho hầu hết các thuật toán và nhiều hướng dẫn online.

*Scikit-learn cung cấp các mô-đun để xây dựng, điều chỉnh và đánh giá mô hình ML:*

Preprocessing đề cập đến các công cụ Scikit-learn hữu ích trong việc trích xuất và chuẩn hóa features trong quá trình phân tích dữ liệu.

Classification đề cập đến một bộ công cụ xác định danh mục liên kết với dữ liệu trong mô hình học máy. Ví dụ: những công cụ này được sử dụng để phân loại email hợp lệ hoặc thư rác. Về cơ bản, Classification xác định đối tượng thuộc về loại nào.

Regression (Phân tích hồi quy) đề cập đến việc tạo ra một mô hình học máy để tìm hiểu mối quan hệ giữa dữ liệu đầu vào và đầu ra, chẳng hạn như hành vi hoặc giá cổ phiếu. Hồi quy dự đoán một thuộc tính có giá trị liên tục được liên kết với một đối tượng.

Các công cụ Clustering trong Scikit-learn tự động nhóm dữ liệu có đặc điểm tương tự thành các bộ (sets), chẳng hạn như dữ liệu khách hàng được sắp xếp theo bộ dựa trên vị trí thực tế.

Dimensionality Reduction làm giảm số lượng biến ngẫu nhiên để phân tích. Ví dụ: để tăng hiệu quả trực quan hóa, dữ liệu bên ngoài có thể bị loại bỏ.

Model selection đề cập đến các thuật toán và khả năng cung cấp các công cụ so sánh, xác thực và chọn tham số tối ưu để sử dụng trong các dự án học máy khoa học dữ liệu.

Pipeline đề cập đến các tiện ích để xây dựng quy trình làm việc mô hình.

Visualization cho học máy cho phép vẽ đồ thị và điều chỉnh trực quan nhanh chóng.

#### Thư viện Tkinter

Thư viện tkinter là một thư viện trong Python được sử dụng để tạo giao diện người dùng đồ họa (GUI). Tkinter cung cấp các công cụ và phương pháp để tạo và quản lý các thành phần giao diện như cửa sổ, nút, hộp thoại, v.v.

Tác dụng chính của thư viện tkinter là giúp người lập trình tạo ra các ứng dụng có giao diện người dùng thân thiện và dễ sử dụng. Tkinter cung cấp các công cụ để thiết kế và tương tác với các thành phần giao diện, đồng thời hỗ trợ các sự kiện và hành vi tương tác của người dùng.

Thư viện này được sử dụng rộng rãi trong việc phát triển ứng dụng desktop và các dự án liên quan đến giao diện người dùng trong Python.

#### Thư viện Surprise

Thư viện Surprise python là một thư viện sử dụng trong lĩnh vực học máy và khám phá dữ liệu. Nó cung cấp các thuật toán và công cụ để xây dựng và đánh giá các mô hình dự đoán và khám phá cấu trúc trong dữ liệu.

*Các chức năng của thư viện Surprise:*

Cung cấp cho người dùng quyền kiểm soát hoàn hảo các thử nghiệm của họ.

Giảm bớt sự khó khăn khi xử lý Tập dữ liệu. Người dùng có thể sử dụng cả bộ dữ liệu tích hợp (Movielens, Jester) và bộ dữ liệu tùy chỉnh của riêng họ.

Cung cấp nhiều thuật toán dự đoán sẵn sàng sử dụng khác nhau như thuật toán cơ sở, phương pháp lân cận, dựa trên hệ số ma trận (SVD, PMF, SVD++, NMF) và nhiều thuật toán khác. Ngoài ra, nhiều thước đo tương tự khác nhau (cosine, MSD, Pearson…) cũng được tích hợp sẵn.

Giúp việc thực hiện các ý tưởng thuật toán mới trở nên dễ dàng.

Cung cấp các công cụ để đánh giá, phân tích và so sánh hiệu quả của các thuật toán. Các quy trình xác thực chéo có thể được chạy rất dễ dàng bằng cách sử dụng các trình vòng lặp CV mạnh mẽ (lấy cảm hứng từ các công cụ tuyệt vời của scikit-learn), cũng như tìm kiếm toàn diện trên một tập hợp các tham số.

#### Môi trường VSCode

**Visual Studio Code chính là ứng dụng cho phép biên tập, soạn thảo các đoạn code để hỗ trợ trong quá trình thực hiện xây dựng, thiết kế website một cách nhanh chóng. Visual Studio Code hay còn được viết tắt là VS Code.**

**Trình soạn thảo này vận hành mượt mà trên các nền tảng như Windows, macOS, Linux. Hơn thế nữa, VS Code còn cho khả năng tương thích với những thiết bị máy tính có cấu hình tầm trung vẫn có thể sử dụng dễ dàng. Visual Studio Code hỗ trợ đa dạng các chức năng Debug, đi kèm với Git, có Syntax Highlighting. Đặc biệt là tự hoàn thành mã thông minh, Snippets, và khả năng cải tiến mã nguồn.**

***Trong đó có thể kể đến những ưu điểm sau:***

**Đa dạng ngôn ngữ lập trình giúp người dùng thỏa sức sáng tạo và sử dụng như HTML, CSS, JavaScript, C++, …**

**Ngôn ngữ, giao diện tối giản, thân thiện, giúp các lập trình viên dễ dàng định hình nội dung. Các tiện ích mở rộng rất đa dạng và phong phú.**

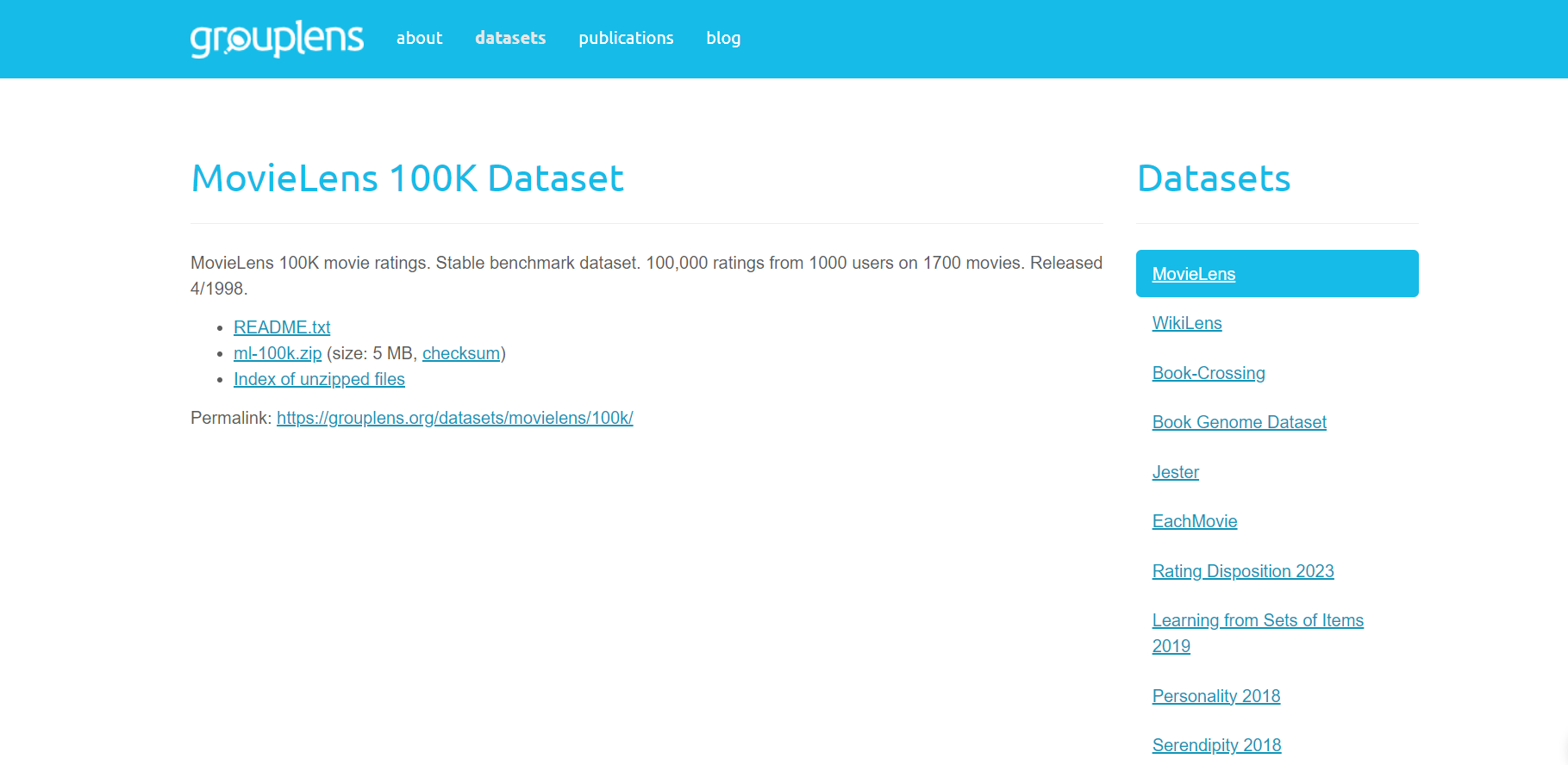
**Tích hợp các tính năng quan trọng như tính năng bảo mật (Git), khả năng tăng tốc xử lý vòng lặp (Debug), …**

**Đơn giản hóa việc tìm quản lý hết tất cả các Code có trên hệ thống.**

## Phân tích khám phá dữ liệu (EDA)

### Khám phá dữ liệu

Bộ dữ liệu Movielens 100K là một bộ dữ liệu phổ biến được sử dụng trong nghiên cứu khoa học, bao gồm 100.000 xếp hạng từ 943 người dùng cho 1682 bộ phim, với mỗi xếp hạng từ 1 đến 5. Dữ liệu được thu thập thông qua trang web MovieLens (movielens.umn.edu) trong khoảng thời gian bảy tháng từ ngày 19 tháng 9 năm 1997 đến ngày 22 tháng 4 năm 1998.

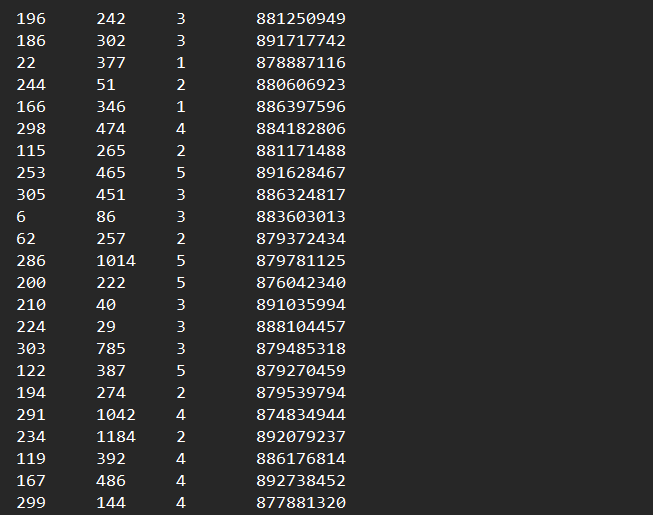


Hình . Dữ liệu MovieLens 100K từ trang web Grouplens

Dữ liệu này đã được lọc để loại bỏ các người dùng có ít hơn 20 xếp hạng hoặc thiếu thông tin cá nhân đầy đủ.

Bộ dữ liệu lớn em sẽ sử dụng 2 file dữ liệu của bộ dữ liệu cho bài toán của mình là file u.data và u.item:

* U-data là tập dữ liệu đầy đủ gồm 100000 xếp hạng của 943 người dùng trên 1682 mục. User và item được đánh số liên tục từ 1.



Hình . Dữ liệu phim file U-data

*Bao gồm các cột dữ liệu:*

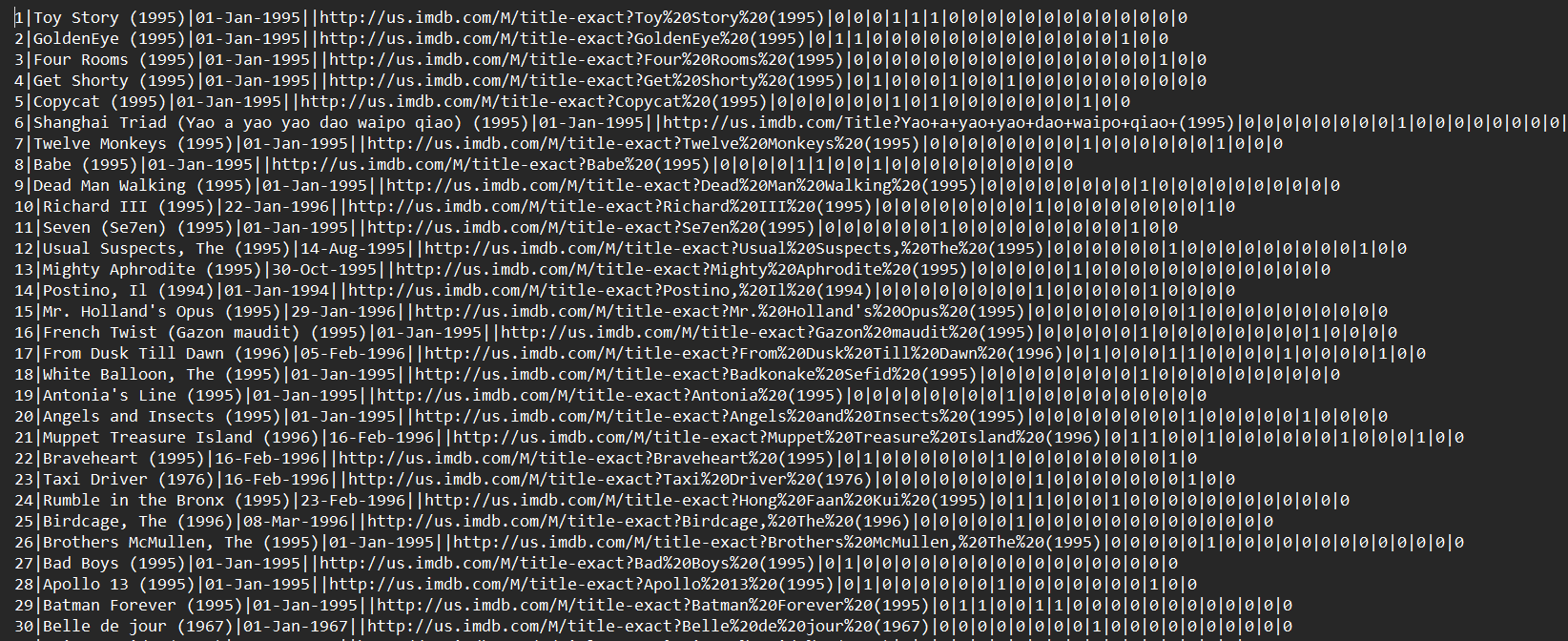
user id (id người dùng)

item id (id mục)

đánh giá (đánh giá)

timestamp (dấu thời gian).

* U-item là file dữ liệu thông tin về các item (phim).



Hình . Dữ liệu phim file U-item

*Dữ liệu bao gồm các cột dữ liệu:*

movie id: id của phim

movie title: tiêu đề phim

release date: ngày phát hành phim

video release date: ngày phát hành video

URL IMDb: link phim trên trang web phim IMDb

Unknown

Action: thể loại phim hành động

Adventure: thể loại phim phiêu lưu

Animation: thể loại phim hoạt hình

Children's: thể loại phim dành cho trẻ em

Comedy: thể loại phim hài

Crime: thể loại phim tội phạm

Documentary: thể loại phim tài liệu

Drama: thể loại phim kịch

Fantasy: thể loại phim tưởng tượng

Film-Noir: thể loại phim noir

Horror: thể loại phim kinh dị

Musical: thể loại phim âm nhạc

Mystery: thể loại phim bí ẩn

Romance: thể loại phim lãng mạn

Sci-Fi: thể loại phim khoa học viễn tưởng

Thriller: thể loại phim kinh dị

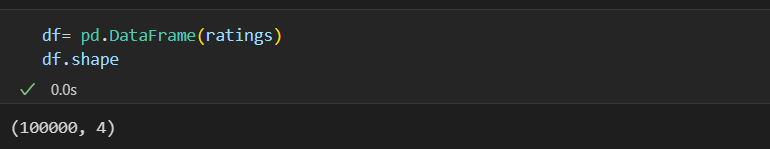
War: thể loại phim chiến tranh

Western: thể loại phim phương Tây

19 trường cuối cùng là thể loại, số 1 biểu thị phim thuộc thể loại đó, số 0 biểu thị nó không thuộc thể loại đó.

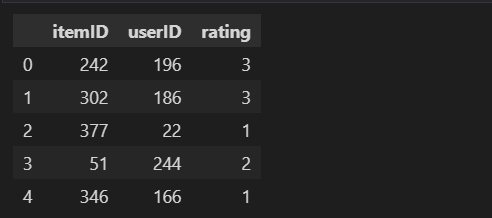
### Phân tích dữ liệu

Bộ dữ liệu sử dụng cho bài toán sẽ sử dụng trên file U-data có tổng 100000 hàng và 4 cột:



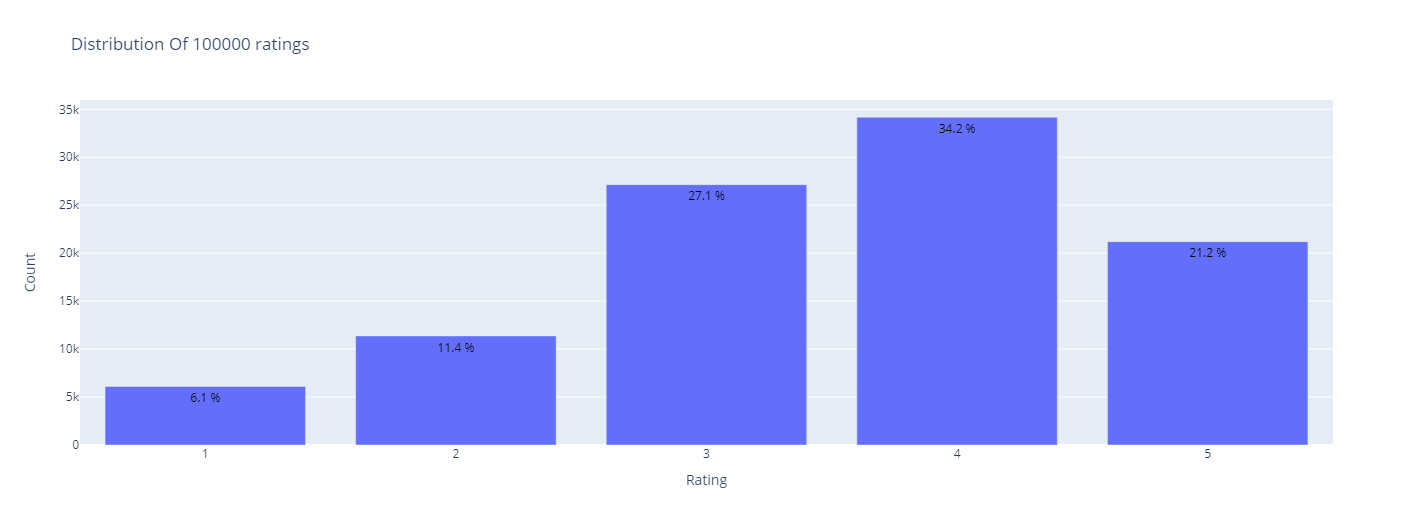
Hình . Đếm số hàng, cột của file dữ liệu

Sử dụng 3 cột chính phục vụ cho bài toán là itemID, userID và đánh giá:



Hình . Các cột dữ liệu sử dụng cho bài toán

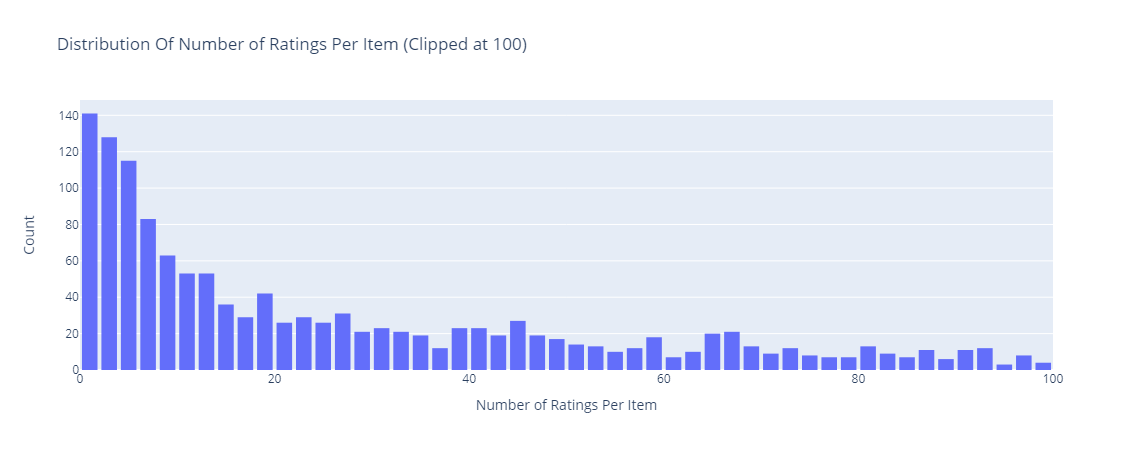
Đếm số lượng đánh giá:



Hình . Phân tích số lượng đánh giá

Từ biểu đồ, ta thấy các bộ phim chủ yếu có đánh giá từ 3 đến 5 sao, bộ phim được đánh giá 4 sao chiếm phần trăm cao nhất lên đến 34,2%.

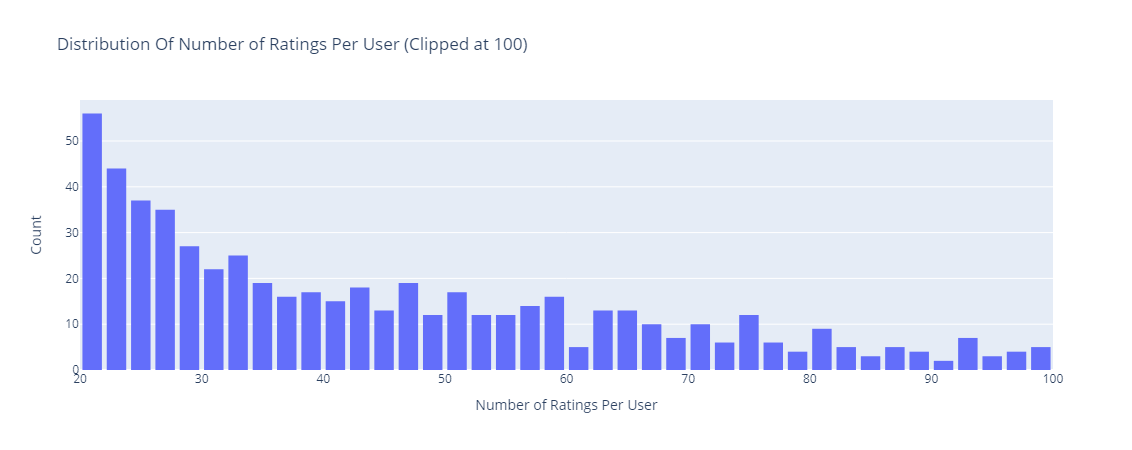
Phân phối số lượng đánh giá theo tổng số lượng đánh giá phim (item), phân phối theo phim đạt số lượng đánh giá từ 0 đến 100.



Hình . Phân phối số lượng đánh giá theo phim

Từ biểu đồ, ta thấy số lượt đánh giá của các bộ phim khá nhiều chủ yếu chiếm từ 1 đến 100 lượt đánh giá và bộ phim có tổng số lượt đánh giá nhiều từ 0 đến 12 số lượt đánh giá. Trong đó, phim có số lượt đánh giá bằng 0 chiếm số lượng cao nhất là 140 phim, tiếp theo là số lượt đánh giá bằng 1 là 128 bộ phim và thấp nhất là bộ phim có 94-95 lượt đánh giá có số lượng là 3 bộ phim, điều này có thể chỉ ra rằng chỉ một số ít bộ phim trong tập dữ liệu nhận được sự quan tâm lớn từ cộng đồng người xem.

Phân phối số lượng đánh giá theo 100 người dùng đầu tiên:



Hình . Phân phối số lượng đánh giá theo 100 người dùng đầu tiên

Từ biểu đồ, cho thấy mỗi người dùng đều có số lượt đánh giá dành cho các bộ phim, phân bố đồng đều từ 20 – 32 lượt đánh giá. Người dùng có số lượt đánh giá từ 20-21 lượt là nhiều nhất chiếm 56 người dùng, người dùng có số lượng đánh giá thấp nhất 90-91 lượt đánh giá là thấp nhất chiếm 2 người dùng.

### Tiền xử lý dữ liệu

Đầu tiên sẽ thực hiện đọc dữ liệu đầu vào:

column\_names1 = ['user id','movie id','rating','timestamp']

dataset = pd.read\_csv('Data Movielens/u.data', sep='\t',header=None,names=column\_names1)

d = 'movie id | movie title | release date | video release date | IMDb URL | unknown | Action | Adventure | Animation | Children | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western'

column\_names2 = d.split(' | ')

items\_dataset = pd.read\_csv('Data Movielens/u.item', sep='|',header=None,names=column\_names2,encoding='latin-1')

movie\_dataset = items\_dataset[['movie id','movie title']]

Hai file dữ liệu được sử dụng cho bài toán là 2 file khác nhau được liên kết với nhau bằng thuộc tính movie id. Nên sẽ hợp nhất 2 file lại một:

merged\_dataset = pd.merge(dataset, movie\_dataset, how='inner', on='movie id')

Sử dụng groupby và agg để tính điểm trung bình đánh giá của mỗi người dùng cho mỗi bộ phim:

refined\_dataset = merged\_dataset.groupby(by=['user id','movie title'], as\_index=False).agg({"rating":"mean"})

Tạo một DataFrame mới chứa số lượng đánh giá cho mỗi điểm đánh giá:

rating\_count\_df = pd.DataFrame(refined\_dataset.groupby(['rating']).size(), columns=['count'])

Trong bộ dữ liệu sẽ có các đánh giá bị bỏ trống (người dùng không đánh giá phim này), ta sẽ tính số lượng đánh giá thiếu này và bổ sung các đánh giá = 0 vào các đánh giá bị bỏ trống đó:

num\_users = len(refined\_dataset['user id'].value\_counts())

num\_items = len(refined\_dataset['movie title'].value\_counts())

total\_count = num\_items \* num\_users

zero\_count = total\_count-refined\_dataset.shape[0]

rating\_count\_df = rating\_count\_df.append(

    pd.DataFrame({'count': zero\_count}, index=[0.0]),

    verify\_integrity=True,

).sort\_index()

Vì giá trị của số lượng đánh giá lớn, do đó ta giảm đi bằng cách lấy logarit của nó:

rating\_count\_df['log\_count'] = np.log(rating\_count\_df['count'])

rating\_count\_df = rating\_count\_df.reset\_index().rename(columns={'index': 'rating score'})

Tạo một ma trận bảng đánh giá của người dùng cho các bộ phim, với các giá trị là điểm đánh giá. Chuyển đổi ma trận thành dạng ma trận thưa thớt để giảm lượng dữ liệu không cần thiết:

movie\_to\_user\_sparse\_df = csr\_matrix(movie\_to\_user\_df.values)

movies\_list = list(movie\_to\_user\_df.index)

Tạo một từ điển với tên phim là khóa và chỉ số của nó từ movies\_list:

movie\_dict = {movie : index for index, movie in enumerate(movies\_list)}

case\_insensitive\_movies\_list = [i.lower() for i in movies\_list]

### Xây dựng mô hình KNN

Bước 1: Điều chỉnh tham số của thuật toán với GridSearchCV để tìm ra tham số tốt nhất.

param\_grid = {

    'n\_neighbors': [5, 10, 15],

    'radius': [1, 2, 3],

    'metric': ['cosine', 'euclidean'],

    'algorithm': ['brute']

}

# Tạo mô hình NearestNeighbors

knn\_movie\_model = NearestNeighbors()

# Tạo đối tượng GridSearchCV

grid\_search = GridSearchCV(knn\_movie\_model, param\_grid, scoring='neg\_mean\_squared\_error', cv=5, n\_jobs=-1)

#Điều chỉnh tìm kiếm dạng lưới cho phù hợp với dữ liệu

grid\_search.fit(movie\_to\_user\_sparse\_df)

# In các thông số tốt nhất được tìm thấy

print("Best Parameters: ", grid\_search.best\_params\_

Kết quả các thông số tốt nhất:



Hình . Kết quả các thông số tốt nhất

Bước 2: Sử dụng thư viện scikit-learn để tạo một mô hình KNN dựa trên ma trận thưa thớt.

knn\_movie\_model = NearestNeighbors(n\_neighbors= my\_n\_neighbors, radius= my\_radius, metric=my\_metric, algorithm=my\_algorithm)

knn\_movie\_model.fit(movie\_to\_user\_sparse\_df)

Sử dụng thuật toán K-Nearest Neighbors (KNN) để tìm kiếm các hàng xóm gần nhất dựa trên độ tương đồng cosine giữa các vector biểu diễn của các bộ phim trong ma trận đánh giá của người dùng (movie\_to\_user\_sparse\_df).

Lớp NearestNeighbors trong thư viện sklearn.neighbors cung cấp các phương thức để tìm kiếm các hàng xóm gần nhất dựa trên các phép đo khác nhau. Trong đoạn mã trên, chúng ta sử dụng độ đo cosine để tính toán độ tương đồng giữa các vector. Tham số algorithm='brute' cho biết rằng thuật toán sẽ sử dụng phương pháp tìm kiếm brute-force để tìm kiếm các hàng xóm gần nhất.

Sau khi tạo đối tượng knn\_movie\_model, chúng ta sử dụng phương thức fit() để huấn luyện mô hình trên ma trận đánh giá của người dùng. Khi huấn luyện mô hình, thuật toán sẽ tính toán các hàng xóm gần nhất cho mỗi bộ phim trong ma trận đánh giá của người dùng và lưu trữ chúng để sử dụng cho việc dự đoán đánh giá của người dùng cho các bộ phim mới.

Bước 3: Xây dựng hàm hàm lấy các chuỗi kí tự trong tên phim được nhập vào, so sánh với danh sách phim để tìm các phim mà có thể bạn  đang tìm kiếm

Hàm nhận đầu vào là một chuỗi movie và trả về một danh sách các phim có thể tương ứng với chuỗi movie dựa trên việc so sánh không phân biệt chữ hoa/chữ thường.

Bước 4: Xây dựng hàm đề xuất phim tương tự dựa vào độ đo cosine:

def get\_similar\_movies(movie, n=10):

    index = movie\_dict[movie]

    knn\_input = np.asarray([movie\_to\_user\_df.values[index]])

    n = min(len(movies\_list)-1, n)

    distances, indices = knn\_movie\_model.kneighbors(knn\_input, n\_neighbors=n+1)

    # Tính toán độ tương đồng cosine từ khoảng cách cosine.

    cosine\_similarities = 1 - distances.flatten()

    recommendations = [

        f"{movies\_list[indices[0][i]]} - Similarity: {cosine\_similarities[i]:.2f}"

        for i in range(1, len(cosine\_similarities))

    ]

    return recommendations

Định nghĩa hàm get\_similar\_movies() để tìm kiếm các bộ phim tương tự với một bộ phim đã cho.

Hàm này nhận đầu vào là tên của bộ phim movie và số lượng bộ phim tương tự cần tìm n (mặc định là 10). Đầu tiên, nó tìm kiếm chỉ mục của bộ phim trong từ điển movie\_dict. Sau đó, nó tạo một mảng numpy knn\_input chứa vector biểu diễn của bộ phim đã cho trong ma trận đánh giá của người dùng movie\_to\_user\_df.

Tiếp theo, nó sử dụng mô hình KNN đã huấn luyện knn\_movie\_model để tìm kiếm n+1 hàng xóm gần nhất của bộ phim đã cho. Tham số n\_neighbors của phương thức kneighbors() cho biết số lượng hàng xóm gần nhất cần tìm.

Sau khi tìm kiếm được các hàng xóm gần nhất, hàm tính toán độ tương đồng cosine từ khoảng cách cosine và lưu trữ chúng trong mảng cosine\_similarities. Cuối cùng, hàm trả về danh sách các bộ phim tương tự với bộ phim đã cho, sắp xếp theo độ tương đồng cosine giảm dần. Mỗi bộ phim được trả về dưới dạng một chuỗi có định dạng "<tên bộ phim> - Similarity: <độ tương đồng cosine>".

Bước 5: Xây dựng hàm trả về các bộ phim đề xuất dựa vào từ khóa

def recommend():

    try:

        movie\_name = entry\_movie.get()

        movie\_name\_lower = movie\_name.lower()

        if movie\_name\_lower not in case\_insensitive\_movies\_list:

            raise InvalidInput

        else:

            num\_recom = int(entry\_num\_recom)

            recommendations = get\_similar\_movies(movies\_list[case\_insensitive\_movies\_list.index(movie\_name\_lower)], num\_recom)

            result\_label.config(text="\n".join(recommendations))

    except InvalidInput:

        possible\_movies = get\_possible\_movies(movie\_name\_lower)

        if len(possible\_movies) == len(movies\_list):

            result\_label.config(text="Movie name entered does not exist in the list.")

        else:

            indices = [case\_insensitive\_movies\_list.index(i) for i in possible\_movies]

            suggestions = '\n'.join([movies\_list[i] for i in indices])

            result\_label.config(text=suggestions)

Định nghĩa hàm recommend() để thực hiện việc đưa ra các đề xuất bộ phim cho người dùng dựa trên bộ phim mà họ đã chọn. Hàm này được gọi khi người dùng bấm nút "Đề xuất".

Đầu tiên, nó lấy tên bộ phim mà người dùng đã nhập từ ô văn bản entry\_movie. Nếu tên bộ phim này không tồn tại trong danh sách bộ phim, nó sẽ ném ra ngoại lệ InvalidInput. Nếu tên bộ phim hợp lệ, nó sẽ lấy số lượng bộ phim đề xuất mà người dùng muốn từ ô văn bản entry\_num\_recom.tượng

Sau đó, hàm gọi hàm get\_similar\_movies() để tìm kiếm các bộ phim tương tự và lưu trữ kết quả trong biến recommendations.

Cuối cùng hàm sử dụng phương thức config() của đối tượng Label result\_label để hiển thị các đề xuất bộ phim cho người dùng. Nếu ngoại lệ InvalidInput được ném ra, hàm sẽ gọi hàm get\_possible\_movies() để tìm kiếm các bộ phim có thể liên quan đến tên bộ phim đã nhập và hiển thị chúng cho người dùng.

Bước 6: Sử dụng thư viện tkinter để tạo giao diện người dùng

### Xây dựng mô hình SVD

Bước 1: Điều chỉnh tham số của thuật toán với GridSearchCV để tìm ra tham số tốt nhất.

# Xác định lưới tham số

param\_grid = {

    'n\_components': [10, 20, 30, 40, 50],

    'n\_iter': [5, 10, 15, 20, 25],

    'random\_state': [42]  # You can add more values if needed

}

# Tạo mô hình TruncatedSVD

svd\_model = TruncatedSVD()

# Tạo đối tượng GridSearchCV với phương sai được giải thích làm thước đo tính điểm

grid\_search = GridSearchCV(svd\_model, param\_grid, scoring='explained\_variance', cv=5, n\_jobs=-1)

# Điều chỉnh tìm kiếm dạng lưới cho phù hợp với dữ liệu

grid\_search.fit(movie\_to\_user\_df)

#In các thông số tốt nhất được tìm thấy

print("Best Parameters: ", grid\_search.best\_params\_)

Kết quả tham số tốt nhất:



Hình . Kết quả tham số tốt nhất

Bước 2: Sử dụng thư viện scikit-learn để tạo một mô hình SVD dựa trên ma trận thưa thớt

svd\_model = TruncatedSVD(n\_components=my\_n\_components,n\_iter = my\_n\_iters, random\_state=my\_random\_state)

svd\_matrix = svd\_model.fit\_transform(movie\_to\_user\_df)

TruncatedSVD là một phương pháp giảm chiều dữ liệu sử dụng Singular Value Decomposition (SVD) với tham số n\_components=40 chỉ ra rằng số lượng thành phần chính giữ lại sau khi giảm chiều dữ liệu, n\_iter=25 là số lần lặp để tiến hành SVD, random\_state=42 là seed cho việc tạo số ngẫu nhiên để đảm bảo kết quả có thể tái tạo được.

Sử dụng mô hình svd\_model đã được khởi tạo trước đó để giảm chiều dữ liệu của movie\_to\_user\_df (một ma trận biểu diễn điểm đánh giá của người dùng cho các bộ phim) thành một ma trận mới với số chiều được giảm xuống là n\_components =40 (được chỉ định trong đoạn mã trước đó). Kết quả là svd\_matrix, một ma trận mới có số hàng bằng số bộ phim và số cột bằng n\_components. Ma trận này có thể được sử dụng để tính toán độ tương đồng giữa các bộ phim và đưa ra các dự đoán đánh giá của người dùng cho các bộ phim mà họ chưa xem.

Bước 3: Xây dựng hàm lấy các chuỗi kí tự trong tên phim được nhập vào, so sánh với danh sách phim để tìm các phim mà có thể bạn  đang tìm kiếm

Hàm nhận đầu vào là một chuỗi movie và trả về một danh sách các phim có thể tương ứng với chuỗi movie dựa trên việc so sánh không phân biệt chữ hoa/chữ thường.

Bước 4: Xây dựng hàm tạo ra danh sách các phim được đề xuất dựa trên sự tương đồng với một phim cụ thể.

def recommend():

    try:

        movie\_name = entry\_movie.get()

        movie\_name\_lower = movie\_name.lower()

        if movie\_name\_lower not in case\_insensitive\_movies\_list:

            raise InvalidInput

        else:

            num\_recom = int(entry\_num\_recom)

            recommendations = get\_svd\_similar\_movies(movies\_list[case\_insensitive\_movies\_list.index(movie\_name\_lower)], num\_recom)

            result\_label.config(text="\n".join(recommendations))

    except InvalidInput:

        possible\_movies = get\_possible\_movies(movie\_name\_lower)

        if len(possible\_movies) == len(movies\_list):

            result\_label.config(text="Movie name entered does not exist in the list.")

        else:

            indices = [case\_insensitive\_movies\_list.index(i) for i in possible\_movies]

            suggestions = '\n'.join([movies\_list[i] for i in indices])

            result\_label.config(text=suggestions)

Cụ thể, hàm get\_svd\_similar\_movies nhận đầu vào là tên của phim movie\_name và số lượng phim đề xuất num\_recom.

Đầu tiên, nó sử dụng movie\_dict để lấy chỉ mục của phim trong ma trận SVD. Sau đó, nó lấy vector biểu diễn của phim đó từ svd\_matrix.

Tiếp theo, hàm tính toán độ tương đồng cosine giữa phim này và tất cả các phim khác trong svd\_matrix sử dụng phép nhân ma trận.

Sau đó, hàm thực hiện chuẩn hóa giá trị tương đồng cosine về khoảng [0, 1] và lấy ra chỉ mục của các phim có độ tương đồng cao nhất.

Cuối cùng, hàm tạo ra danh sách các phim được đề xuất với thông tin về tên phim và độ tương đồng đã được chuẩn hóa.

Bước 5: Xây dựng hàm trả về tên các bộ phim dựa vào từ khóa

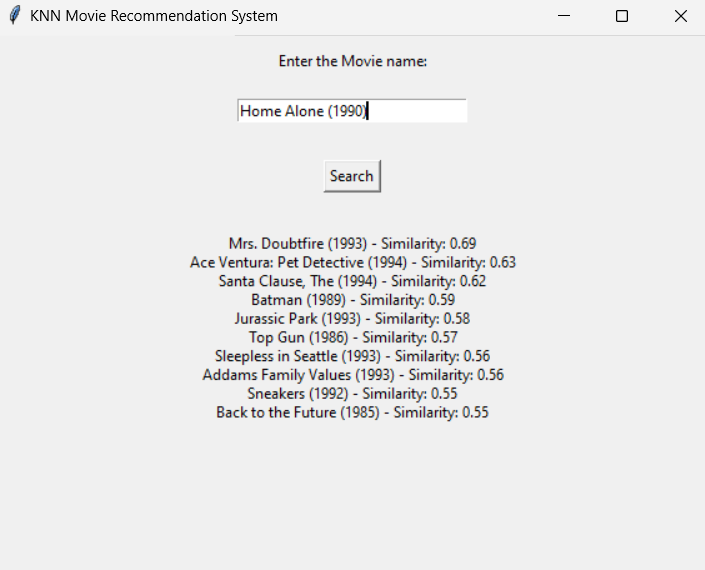
Hàm recommend() nhận đầu vào từ một trường nhập entry\_movie và một trường nhập khác entry\_num\_recom. Đầu tiên, nó kiểm tra xem từ khóa nhập vào có tồn tại trong danh sách case\_insensitive\_movies\_list hay không. Nếu không tồn tại, nó sẽ gọi hàm get\_possible\_movies() để lấy danh sách các phim có thể tương ứng với từ khóa nhập vào và hiển thị danh sách đó. Nếu từ khóa nhập vào tồn tại, nó sẽ gọi hàm get\_svd\_similar\_movies() để lấy danh sách các phim được đề xuất dựa trên tên phim và số lượng phim đề xuất, sau đó hiển thị danh sách đó.

Bước 6: Sử dụng thư viện tkinter tạo giao diện người dùng

# KẾT QUẢ VÀ ĐÁNH GIÁ MÔ HÌNH

## Kết quả xây dựng mô hình KNN đề xuất phim

Khi nhập tên một bộ phim hệ thống sẽ đưa ra kết quả 10 bộ phim tương tự cùng với độ đo cosine đánh giá kết quả tương tự của từng bộ phim với bộ phim nhập vào:



Hình . Kết quả mô hình KNN đề xuất phim

Kết quả trả về 10 bộ phim tương tự với bộ phim "Home Alone" dựa theo độ đo đánh giá cosine:

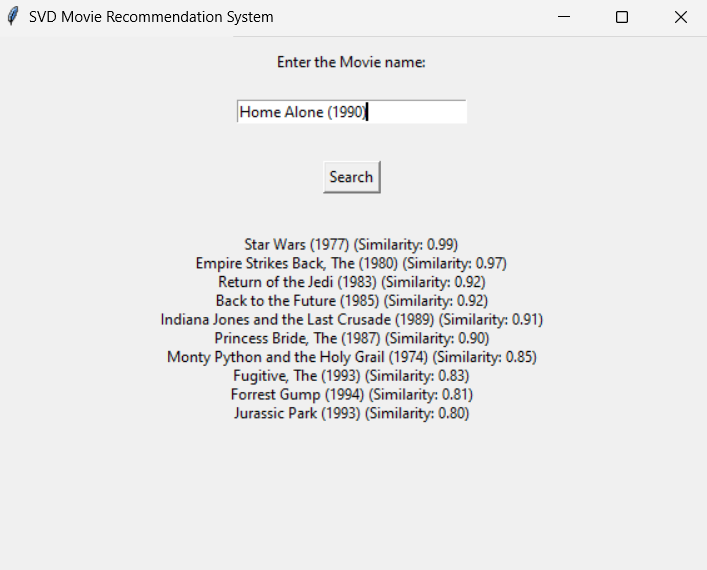
Bộ phim "Mrs.Doubtfire" đạt giá trị cao nhất là 0.69.

Thấp nhất là "Back to the Feature" với kết quả là 0.55.

Cho thấy kết quả của đề xuất phim tìm ra các phim có mức độ tương đồng khá cao và có thể phù hợp với sở thích của người dùng.

## Kết quả xây dựng mô hình SVD đề xuất phim

Khi nhập tên một bộ phim hệ thống sẽ đưa ra kết quả 10 bộ phim tương tự cùng với độ đo cosine đánh giá kết quả tương tự của từng bộ phim với bộ phim nhập vào:



Hình . Kết quả mô hình SVD đề xuất phim

Kết quả trả về 10 bộ phim tương tự với bộ phim "Home Alone" dựa theo độ đo đánh giá cosine:

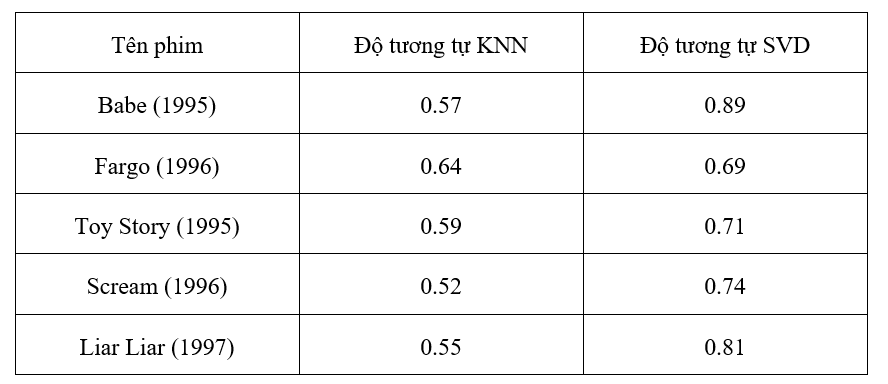
Với bộ phim "Star Wars" đạt giá trị tương tự cao nhất là 0.99.

Thấp nhất là bộ phim "Jurassic Park" với kết quả là 0.80l

Các điểm tương đồng cao cho thấy hệ thống tin rằng có mối tương quan mạnh mẽ giữa bộ phim được chọn và các bộ phim được đề xuất, cho thấy rằng người xem thích phim "Home Alone" có khả năng sẽ thích các bộ phim được đề xuất.

## Đánh giá kết quả xây dựng mô hình KNN và SVD đề xuất phim

Chọn ra 5 bộ phim bất kỳ và tính độ trung bình tương tự của 10 đề xuất phim được đưa ra:



Hình . Kết quả độ tương tự đề xuất phim của mô hình KNN và SVD

Từ kết quả đánh giá trên cho thấy, mô hình SVD đưa ra những đề xuất phim có độ tương tự cao hơn với mô hình KNN, cho ra kết quả tốt hơn. Cho thấy, mô hình SVD đưa ra kết quả chính xác cao, hoạt động tốt hơn trong bài toán đề xuất phim dựa vào rating phim.

# KẾT LUẬN

1. Kết quả đạt được

Sau khi tìm hiểu và hoàn thành đồ án về ứng dụng các phương pháp học máy trong bài toán đề xuất phim, tôi đã đạt được một số thành quả nhất định sau:

Nắm bắt được những kiến thức cơ bản về hệ thống đề xuất.

Hiểu biết thêm về một số phương pháp xây dựng hệ thống đề xuất.

Nắm vững kiến thức 2 thuật toán KNN và SVD, ứng dụng vào xây dựng hệ thống đưa ra đề xuất phim phù hợp và có độ chính xác cao.

2. Hạn chế

Do thời gian tìm hiểu còn hạn chế, chưa có kinh nghiệm và hiểu biết nhiều về các thuật toán KNN và SVD để đưa ra đề xuất phim nên kiến thức còn khá mới mẻ dẫn đến việc không thể tránh khỏi có những sai sót khi thực hiện và hoàn thành đồ án.

3. Hướng phát triển

Ứng dụng một số mô hình học máy khác và tìm hiểu một số phương pháp học sâu như LSTM, RNN, CNN cho bài toán đề xuất phim. Cải thiện, tăng cường độ chính xác của các mô hình bằng cách thử nghiệm với bộ tham số khác nhau cho mô hình.

Sử dụng thêm các tiêu chí khác để đánh giá mô hình.

Ứng dụng mô hình hiệu quả cho các bộ dữ liệu phim khác với nhiều dữ liệu đầu vào hơn để có thể ứng dụng vào các bài toán đề xuất phim thực tế.

TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | T. A. Dũng, "http://lib.uet.vnu.edu.vn/," 2020. [Online]. |
| [2] | Đỗ Thị Liên et al, "Một mô hình đồ thị cho hệ thống lai," 2015. |
| [3] | N.T.Nghe, "Chương 2: Hệ thống gợi ý: Kỹ thuật và ứng dụng, Cần Thơ: Khoa công nghệ thông tin và truyền thông, Đại học Cần Thơ," [Online]. |
| [4] | H. N. Tín, "Phát triển một số phương pháp khuyến nghị hỗ trợ tìm kiếm thông tin học thuật dựa trên tiếp cận phân tích mạng xã hội," 2016. |
| [5] | P. T. Huyền, "Đề xuất kỹ thuật tư vấn nhóm cho dữ liệu phim dựa vào phân cụm thô - mờ," 2021. |
| [6] | Ramni Harbir Singh et al, "Movie Recommendation System using Cosine Similarity and KNN," 2020. |
| [7] | P. H. Trương, "Nghiên cứu và ứng dụng kỹ thuật SVD vào hệ thống gợi ý," 2018. |
| [8] | Xun Zhou et al, "SVD-based incremental approaches for recommender systems," 2015. |