|  |  |
| --- | --- |
|  |  |
| **PHẠM QUỐC HUY** | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
|  |
| ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC NGÀNH KHOA HỌC MÁY TÍNH |
|  |
| **XÂY DỰNG HỆ THỐNG GỢI Ý PHIM SỬ DỤNG THUẬT TOÁN HỌC MÁY** |
|  |
|  |
| **CBHD: *Ths.Phạm Thị Kim Phượng*** |
| **KHOA HỌC MÁY TÍNH** | **Sinh viên: *Phạm Quốc Huy*** |
| **Mã số sinh viên: *2019605757*** |
|  |
|  |
|  |
| Hà Nội – Năm 2024 |

**MỤC LỤC**

[DANH MỤC HÌNH ẢNH 1](#_Toc166424793)

[DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT 4](#_Toc166424794)

[LỜI CẢM ƠN 5](#_Toc166424795)

[LỜI MỞ ĐẦU 6](#_Toc166424796)

[CHƯƠNG 1: TỔNG QUAN VỀ BÀI TOÁN GỢI Ý 8](#_Toc166424797)

[1.1 Giới thiệu chung 8](#_Toc166424798)

[1.2 Khái niệm: 9](#_Toc166424799)

[1.3 Ý tưởng 9](#_Toc166424800)

[1.4 Bài toán Gợi ý 9](#_Toc166424801)

[CHƯƠNG 2: MỘT SỐ KỸ THUẬT GỢI Ý 12](#_Toc166424802)

[2.1 Các hướng tiếp cận bài toán gợi ý 12](#_Toc166424803)

[2.2 Content-based Filtering 12](#_Toc166424804)

[2.2.1 Hồ sơ sản phẩm 13](#_Toc166424805)

[2.2.2 Sử dụng tích vô hướng như một thước đo độ tương đồng (Similarity Measure) 14](#_Toc166424806)

[2.2.3 Ưu và nhược điểm 14](#_Toc166424807)

[2.3 Collaborative Filtering 14](#_Toc166424808)

[2.3.1 User-user collabrative Filtering vs Item-item collaborative Filtering 16](#_Toc166424809)

[2.3.2 Ưu và nhược điểm 17](#_Toc166424810)

[2.3.3 Matrix Factorization 17](#_Toc166424811)

[2.4 Hệ thống lai (Hybrid) 18](#_Toc166424812)

[2.5 Chức năng của hệ thống gợi ý phim 19](#_Toc166424813)

[2.6 Tác dụng của hệ thống gợi ý 20](#_Toc166424814)

[CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG 21](#_Toc166424815)

[3.1 Hoạt động của hệ thống 21](#_Toc166424816)

[3.2 Mô tả chức năng 21](#_Toc166424817)

[3.3 Mô tả use case 21](#_Toc166424818)

[3.4 Phân tích các use case 24](#_Toc166424819)

[3.5 Biểu đồ tổng hợp 26](#_Toc166424820)

[CHƯƠNG 4: THIẾT KẾ GIAO DIỆN, CÀI ĐẶT VÀ KIẾM THỬ 27](#_Toc166424821)

[4.1 Tổng quan hệ thống 27](#_Toc166424822)

[4.2 Một số khái niệm 27](#_Toc166424823)

[4.2.1 Thư viện Scikit-learn 27](#_Toc166424824)

[4.2.2 Thư viện Surprise 28](#_Toc166424825)

[4.2.3 Thư viện Streamlit 29](#_Toc166424826)

[4.2.4 Thư viện Pickle 29](#_Toc166424827)

[4.2.5 Xử lý ngôn ngữ tự nhiên (NLP) 29](#_Toc166424828)

[4.3 Thiết kế giao diện 30](#_Toc166424829)

[4.3.1 Biểu đồ các lớp phân tích 30](#_Toc166424830)

[4.3.2 Giao diện usecase “Đăng nhập” 31](#_Toc166424831)

[4.3.3 Giao diện usecase “Đăng kí” 32](#_Toc166424832)

[4.3.4 Giao diện usecase “Nhấn Show Recommendation” 33](#_Toc166424833)

[4.4 Bộ dữ liệu (Dataset) 34](#_Toc166424834)

[4.5 Xây dựng hệ thống 35](#_Toc166424835)

[4.5.1 Lựa chọn cơ sở dữ liệu 35](#_Toc166424836)

[4.5.2 Load và khám phá dữ liệu 36](#_Toc166424837)

[4.5.3 Tiếp cận theo Demographic Filtering 38](#_Toc166424838)

[4.5.4 Tiếp cận theo Content-based Filtering 42](#_Toc166424839)

[4.5.5 Tiếp cận theo Collaborative Filtering 51](#_Toc166424840)

[4.6 Xây dựng giao diện 53](#_Toc166424841)

[4.6.1 Chuẩn bị 53](#_Toc166424842)

[4.6.2 Xây dựng giao diện 55](#_Toc166424843)

[4.6.3 Giao diện ứng dụng 63](#_Toc166424844)

[4.7 Kiểm thử hệ thống 65](#_Toc166424845)

[4.7.1 Kiểm thử chức năng 65](#_Toc166424846)

[4.7.2 Kiểm thử giao diện 65](#_Toc166424847)

[4.7.3 Báo cáo kiểm thử 66](#_Toc166424848)

[KẾT LUẬN 67](#_Toc166424849)

[TÀI LIỆU THAM KHẢO 68](#_Toc166424850)

# DANH MỤC HÌNH ẢNH

[Hình 1.1: Minh họa hệ thống gợi ý 9](#_Toc167144096)

[Hình 2.1: Kỹ thuật Content-based Filtering 13](#_Toc167144097)

[Hình 2.2: Kỹ thuật Collaborative Filtering 16](#_Toc167144098)

[Hình 3.1: Biểu đồ Usecase chính 21](#_Toc167144099)

[Hình 3.2: Biểu đồ trình tự Usecase Đăng nhập 24](#_Toc167144100)

[Hình 3.3: Biểu đồ lớp phân tích usecase Đăng nhập 24](file:///D:\DATN\CODE_BTL\KHMT_2019605757_Pham-Quoc-Huy%20(1).docx#_Toc167144101)

[Hình 3.4: Biểu đồ trình tự usecase Đăng ký 25](#_Toc167144102)

[Hình 3.5: Biểu đồ lớp phân tích usecase Đăng ký 25](#_Toc167144103)

[Hình 3.6: Biểu đồ trình tự usecase Show Recommendation 25](#_Toc167144104)

[Hình 3.7: Biểu đồ lớp phân tích usecase Show Recommendation 26](#_Toc167144105)

[Hình 3.8: Biểu đồ các lớp Entity của hệ thống 26](#_Toc167144106)

[Hình 4.1: Biểu đồ lớp phân tích giao diện 30](#_Toc167144107)

[Hình 4.2: Hình dung màn hình usecase Đăng nhập 31](#_Toc167144108)

[Hình 4.3: Biểu đồ cộng tác màn hình usecase Đăng nhập 31](#_Toc167144109)

[Hình 4.4: Hình dung màn hình usecase Đăng ký 32](#_Toc167144110)

[Hình 4.5: Biểu đồ trình tự usecase Đăng ký 32](#_Toc167144111)

[Hình 4.6: Hình dung màn hình Show Recommendation 33](#_Toc167144112)

[Hình 4.7: Biểu đồ trình tự Show Recommendation 33](#_Toc167144113)

[Hình 4.8: The Movies Dataset 34](#_Toc167144114)

[Hình 4.9: TMDB 5000 Movie Dataset 34](#_Toc167144115)

[Hình 4.10: Lược đồ cấu trúc csdl user 36](#_Toc167144116)

[Hình 4.11: Import thư viện xử lý, tính toán dữ liệu 36](#_Toc167144117)

[Hình 4.12: Load và hiển thị tên các trường dữ liệu trong dataset 37](#_Toc167144118)

[Hình 4.13: Join 2 tệp dữ liệu dựa vào trường “Id” 38](#_Toc167144119)

[Hình 4.14: Giá trị trung bình vote (vote average) của cả tệp 39](#_Toc167144120)

[Hình 4.15: Số lượng vote tối thiểu cần có 40](#_Toc167144121)

[Hình 4.16: Số lượng phim đạt yêu cầu vote tối thiểu. 40](#_Toc167144122)

[Hình 4.17: Hàm tính Weighted Rating 40](#_Toc167144123)

[Hình 4.18: Áp dụng hàm Weighted\_rating() cho toàn bộ dữ liệu 41](file:///D:\DATN\CODE_BTL\KHMT_2019605757_Pham-Quoc-Huy%20(1).docx#_Toc167144124)

[Hình 4.19: Top những bộ phim có Weighted Rating cao nhất 41](#_Toc167144125)

[Hình 4.20: Khuyến nghị phim dựa trên độ phổ biến 42](file:///D:\DATN\CODE_BTL\KHMT_2019605757_Pham-Quoc-Huy%20(1).docx#_Toc167144126)

[Hình 4.21: Hiển thị dữ liệu cột Overview 43](#_Toc167144127)

[Hình 4.22: Kỹ thuật xử lý ngôn ngữ tự nhiên TF-IDF 43](#_Toc167144128)

[Hình 4.23: Tính Cosine Similarity 44](#_Toc167144129)

[Hình 4.24: Ánh xạ ngược index của phim 44](#_Toc167144130)

[Hình 4.25: Khai báo hàm get\_recommendation() 45](#_Toc167144131)

[Hình 4.26: Gợi ý phim The Dark Knight Rises 46](#_Toc167144132)

[Hình 4.27: Chuyển dữ liệu từ stringified list sang list 47](#_Toc167144133)

[Hình 4.28: Khai báo hàm get\_director() lấy danh sách đạo diễn 47](#_Toc167144134)

[Hình 4.29: Hàm get\_list() lấy giá trị của cột cast, keyword, genres 47](file:///D:\DATN\CODE_BTL\KHMT_2019605757_Pham-Quoc-Huy%20(1).docx#_Toc167144135)

[Hình 4.30: Áp dụng hàm get\_director() và get\_list() vào tệp dữ liệu 48](file:///D:\DATN\CODE_BTL\KHMT_2019605757_Pham-Quoc-Huy%20(1).docx#_Toc167144136)

[Hình 4.31: Dữ liệu sau khi định nghĩa lại 48](file:///D:\DATN\CODE_BTL\KHMT_2019605757_Pham-Quoc-Huy%20(1).docx#_Toc167144137)

[Hình 4.32: K hai báo hàm clean\_data() 48](#_Toc167144138)

[Hình 4.33: Áp dụng hàm clean\_data() vào tệp dữ liệu 49](#_Toc167144139)

[Hình 4.34: Khai báo và áp dụng hàm creat\_soup() vào tệp dữ liệu 49](file:///D:\DATN\CODE_BTL\KHMT_2019605757_Pham-Quoc-Huy%20(1).docx#_Toc167144140)

[Hình 4.35: Tính toán cosine similarity 2 50](#_Toc167144141)

[Hình 4.36: Đưa ra gợi ý phim tương tự The Dark Knight Rises 50](#_Toc167144142)

[Hình 4.37: Import thư viện Surprise và đọc dữ liệu 51](#_Toc167144143)

[Hình 4.38: Matrix Factorization Algorithms - SVD 52](#_Toc167144144)

[Hình 4.39: Xây dựng mô hình trên toàn bộ tệp dữ liệu 52](#_Toc167144145)

[Hình 4.40: Áp dụng dự đoán rating của userId = 1 53](#_Toc167144146)

[Hình 4.41: Dump movies.pkl 54](#_Toc167144147)

[Hình 4.42: Dump similarity.pkl 54](#_Toc167144148)

[Hình 4.43: Import thư viện 55](#_Toc167144149)

[Hình 4.44 Cấu hình giao diện 55](#_Toc167144150)

[Hình 4.45 Hàm load\_movies\_data() 55](#_Toc167144151)

[Hình 4.46 Hàm clean\_data() 56](#_Toc167144152)

[Hình 4.47 Hàm Load\_similary\_data() 56](#_Toc167144153)

[Hình 4.48 Hàm reload\_data() 57](#_Toc167144154)

[Hình 4.49: Kết nối và thao tác với database 57](#_Toc167144155)

[Hình 4.50: Tạo giao diện và validate chức năng đăng nhập 58](#_Toc167144156)

[Hình 4.51: Tạo giao diện và vadidate chức năng đăng kí 58](#_Toc167144157)

[Hình 4.52: Hàm fectch\_poster() 59](file:///D:\DATN\CODE_BTL\KHMT_2019605757_Pham-Quoc-Huy%20(1).docx#_Toc167144158)

[Hình 4.53 Hàm Main\_Screen() 59](#_Toc167144159)

[Hình 4.54: Hàm recommend() 60](file:///D:\DATN\CODE_BTL\KHMT_2019605757_Pham-Quoc-Huy%20(1).docx#_Toc167144160)

[Hình 4.55: Tạo nút bấm gợi ý phim và giao diện hiển thị 61](#_Toc167144161)

[Hình 4.56: Define variable and state session 62](#_Toc167144162)

[Hình 4.57: Giao diện đăng nhập 63](#_Toc167144163)

[Hình 4.58: Giao diện đăng ký 63](#_Toc167144164)

[Hình 4.59: Giao diện hệ thống gợi ý phim 64](#_Toc167144165)

[Hình 4.60: Giao diện sau khi đưa ra gợi ý 64](#_Toc167144166)

[Hình 4.61: Bảng kiểm thử chức năng 65](#_Toc167144167)

[Hình 4.62: Bảng kiểm thử giao diện 65](#_Toc167144168)

[Hình 4.63: Báo cáo kiểm thử 66](#_Toc167144169)

# DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ viết tắt, ký hiệu** | **Giải thích** |
| 1 | AI | Artificial Intelligence |
| 2 | ML | Machine Learning |
| 3 | WR | Weighted Rating |
| 4 | RS | Recommendation System |
| 5 | CSDL | Cơ sở dữ liệu |

# LỜI CẢM ƠN

Bài báo cáo đồ án tốt nghiệp là kết quả của quá trình học tập không ngừng nghỉ của em và nhận được sự giúp đỡ tận tình, động viên, khích lệ của thầy cô, doanh nghiệp, bạn bè. Qua đây, em xin gửi lời cảm ơn chân thành đến những người đã giúp đỡ em trong thời gian học tập, nghiên cứu và làm việc vừa qua.

Trước hết, em xin chân thành cảm ơn ban giám hiệu cùng toàn thể quý thầy, cô khoa Công nghệ thông tin trường Đại học Công nghiệp Hà Nội đã tạo điều kiện cho em hoàn thành tốt bài báo cáo này.

Em xin trân trọng gửi đến cô Phạm Thị Kim Phượng - người đã trực tiếp hướng dẫn cũng như cung cấp tài liệu, thông tin khoa học cần thiết để hoàn thành bài báo cáo này lời cảm ơn chân thành và sâu sắc nhất. Đó là những góp ý hết sức quý báu không chỉ trong quá trình thực hiện bài báo cáo này mà còn là hành trang tiếp bước trong quá trình học tập và lập nghiệp sau này.

Và cuối cùng, xin gửi lời cảm ơn đến gia đình, bạn bè, tập thể lớp 2019DHKHMT02, những người luôn sẵn sàng sẻ chia và giúp đỡ chúng em trong học tập và cuộc sống. Mong rằng, chúng ta sẽ mãi mãi gắn bó với nhau.

Xin chúc những điều tốt đẹp nhất sẽ luôn đồng hành cùng mọi người.

Em xin chân thành cảm ơn!

*Sinh viên thực hiện !*

# LỜI MỞ ĐẦU

Internet phát triển, việc tìm kiếm thông tin trở nên dễ dàng hơn rất nhiều. Tuy nhiên, đôi lúc người dùng các hệ thống thông tin, các trang thương mại điện tử, thường gặp khó khăn trong việc tìm kiếm các thông tin cần thiết và phù hợp, như việc chọn mua một sản phẩm phù hợp, hoặc việc lên kế hoạch cho một chuyến đi du lịch; vì người sử dụng có quá nhiều lựa chọn, nhưng không có đủ thời gian hoặc kiến thức để tự đánh giá những lựa chọn này và đưa ra quyết định hợp lý. Nhưng với trường hợp, người dùng không biết rõ sự ưu tiên của mình, công cụ tìm kiếm khó có thể phát huy tác dụng, lúc này bạn cần đến sự giúp đỡ của hệ thống gợi ý.

Hệ thống gợi ý phim sử dụng Machine Learning, Deep Learning để hiểu được sở thích của người dùng, nhờ vậy đưa ra những dự đoán và gợi ý những sản phẩm/dịch vụ/nội dung mà người dùng có thể quan tâm (hàng hoá, phim, sách, video, tin tức, bài hát, khách sạn, khoá học, v.v.).

Dựa trên đó, em thực hiện đề tài “Xây dựng hệ thống gợi ý phim sử dụng thuật toán học máy” bằng thuật toán của Machine Learning. Trong quá trình thực hiện đề tài sẽ không thể tránh những sai sót, vì vậy em mong nhận được những ý kiến đóng góp của thầy cô để đề tài có thể hoàn thiện hơn.

Trong phạm vi báo cáo, em sẽ trình bày 4 chương như sau:

**Chương 1:** *Tổng quan bài toán gợi ý phim.*  Trong chương này em sẽ trình bày sơ lược tổng quan bài toán gợi ý phim

**Chương 2:** *Các kĩ thuật gợi ý phim.* Em đưa ra những khái niệm, kỹ thuật và một vài thuật toán có thể được áp dụng trong việc xây dựng hệ thống gợi ý phim.

**Chương 3:** *Phân tích Hệ thống gợi ý phim.* Chương này, em trình bày những vấn đề liên quan tới việc xây dựng hệ thống gợi ý phim như các thư viện được sử dụng, Biểu đồ UC và phân tích UC.

**Chương 4:** *Thiết kế giao diện, cài đặt và kiểm thử hệ thống*: Chương này, em trình bày những vấn đề liên quan tới việc xây dựng hệ thống gợi ý phim như giao diện, cài đặt hệ thống, kiểm thử.

*Sinh viên thực hiện!*

# TỔNG QUAN VỀ BÀI TOÁN GỢI Ý

## Giới thiệu chung

Vào thời đại mà công nghệ thông tin phát triển bùng nổ, bùng nổ cả về sự phát triển của sản phẩm lẫn số lượng người dùng sử dụng công nghệ, thì gợi ý cho người dùng những thông tin liên quan cũng đang là một trong những bài toán được các nhà phát triển chú trọng, dựa trên những hành vi thu thập được từ quá trình sử dụng của người dùng đó hay dữ liệu trong quá trình sử dụng hệ thống mà người dùng đã tạo ra, các nhà phát triển đã và đang xây dựng những hệ thống gợi ý để gợi ý cho cá nhân người dùng với mục đích nhằm đưa họ đạt tới việc tối ưu các trải nghiệm khi họ sử dụng dịch vụ của hệ thống.

Lấy ví dụ cụ thể với những hệ thống lớn trong thời điểm hiện tại từ các gã khổng lồ công nghệ, trong quá trình sử dụng các sản phẩm của Facebook, Google, Amazon, Netflix… Chắc hẳn chúng ta vẫn đang gặp những trường hợp sau:

* Youtube tự động chuyển sang clip có liên quan đến clip đang xem. Youtube gợi ý những clip có thể bạn sẽ thích.
* Facebook hiển thị quảng cáo những sản phẩm có liên quan đến từ khoá bạn vừa tìm kiếm
* Facebook gợi ý kết bạn
* Netflix tự động gợi ý phim cho người dùng
* Khi mua một món hàng trên Amazon, hệ thống sẽ tự động gợi ý “Frequently bought together”, hoặc nó biết bạn có thể thích món hàng nào dựa trên lịch sử mua hàng của bạn.

Đó chính là kết quả làm việc của hệ thống gợi ý mà các công ty, tập đoàn công nghệ đã xây dựng để đưa ra những gợi ý cho khách hàng sử dụng dịch vụ của họ.

## Khái niệm:

***Hệ thống gợi ý hoặc hệ thống khuyến nghị***, là một lớp con của hệ thống lọc thông tin nhằm tìm cách dự đoán “xếp hạng” (rating) hoặc “sở thích” (preference), mối quan tâm, nhu cầu mà người dùng sẽ đưa ra cho một mặt hàng. Chúng chủ yếu được sử dụng trong các ứng dụng thương mại:

Diagram

Description automatically generated

Hình .1: Minh họa hệ thống gợi ý

## Ý tưởng

Ý tưởng của hệ thống gợi ý cũng xuất phát từ hành vi của người mua hàng: Người mua hàng thường sẽ hỏi bạn bè, chuyên gia, hay từ chính người bán hàng tư vấn cho mình về sản phẩm họ có ý định mua. Người được hỏi sẽ tiến hành thu thập thông tin từ người mua bao gồm: nhu cầu sử dụng, đặc điểm, sản phẩm, màu sắc, chức năng yêu thích…., kết hợp với kiến thức hiểu biết của mình về sản phẩm để đưa ra đề xuất, lời khuyên sản phẩm phù hợp nhất cho người mua. Ở một mức cao hơn, người được hỏi sẽ liên hệ, liên tưởng những người đã từng mua sản phẩm mà có đặc điểm tương đồng với người mua, từ đó họ dự đoán người mua sẽ có khả năng thích sản phẩm nào nhất để đưa ra gợi ý cho người mua.

## Bài toán Gợi ý

Phát biểu bài toán:

**Input:**

* Cho tập người dùng , mỗi người dùng thuộc có các đặc điểm .
* Một tập các sản phẩm, dịch vụ (gọi chung là sản phẩm ) P, mỗi sản phẩm có các đặc điểm đặc trưng
* Một ma trận với ; , thể hiện mối quan hệ giữa tập người dùng và tập sản phẩm . Trong đó là đánh giá của người dùng cho sản phẩm , và lần lượt là số người dùng và số sản phẩm.

**Output:**

Danh sách các sản phẩm thuộc có độ phù hợp với người dùng thuộc nhất.

Để giải bài toán này chúng ta cần xây dựng hàm để đo độ phù hợp sản phẩm đối với người dùng từ đó sẽ lấy ra được danh sách các sản phẩm/ dịch vụ phù hợp (có khả năng người dùng chọn) cao nhất.

Tuỳ thuộc vào phương pháp sử dụng ta có nhiều cách xây dựng hàm F, các cách xây dựng hàm F phụ thuộc chủ yếu bởi các yếu tố sau:

* Đặc điểm của người dùng : điều này được đánh giá chủ quan bởi các quy luật tự nhiên, hoặc các quy tắc cơ bản. Ví dụ là nam thì sẽ có xu hướng mua các sản phẩm của nam hơn các sản phẩm của nữ,…
* Đặc điểm của sản phẩm : giống như lọc theo nội dung của người dùng, các sản phẩm có đặc điểm giống nhau, thì cũng có khả năng được một người dùng đánh giá như nhau. Ví dụ về đặc điểm của các bộ phim có thể là thể loại phim, chủ đề phim,…
* Lịch sử giao dịch của người dùng: từ lịch sử giao dịch có thể suy ra ngành/vấn đề/ chủ đề mà người dùng quan tâm, do đó những sản phẩm có cùng lĩnh vực sẽ có độ liên quan cao hơn. Ví dụ một người đã từng mua áo và giầy đá bóng thì có thể dự đoán người này yêu bóng đá, thích thể thao. Từ đó suy ra người này sẽ có khả năng sử dụng dịch vụ hoặc mua các sản phẩm thể thao cao hơn.
* Người dùng khác nhau có cùng các đặc điểm giống nhau: với quan niệm rằng những người dùng giống nhau sẽ thích, đánh giá những sản phẩm giống nhau. Các đặc điểm của người dùng bao gồm tập đặc điểm ban đầu, kết hợp với các đặc điểm cộng tác như cùng mua mặt hàng nào đó, có các hành vi mua hàng giống nhau… Việc tìm hiểu những mặt hàng/ dịch vụ mà người dùng đã từng quan tâm sẽ đưa ra được những gợi ý phù hợp cho người dùng.

# MỘT SỐ KỸ THUẬT GỢI Ý

## Các hướng tiếp cận bài toán gợi ý

Có 2 hướng tiếp cận chính để xây dựng bài toán gợi ý. Cách 1 là dựa trên nội dung (Content-based filtering): Hệ thống dựa trên nội dung tập trung vào các thuộc tính của mặt hàng, tính tương tự của sản phẩm được xác định bằng cách đo sự tương tự trong các thuộc tính của chúng. Cách 2 là Lọc cộng tác (Collaborative filtering) tập trung vào các mối quan hệ giữa người sử dụng và các mặt hàng. Tính tương tự của các mặt hàng được xác định bởi sự tương tự của xếp hạng của những mặt hàng đó bởi những người dùng để đánh giá cả hai mặt hàng. Các hệ thống gợi ý này thường kết hợp cả 2 hướng tiếp cận trên gọi là hệ thống gợi ý lai (Hybrid).

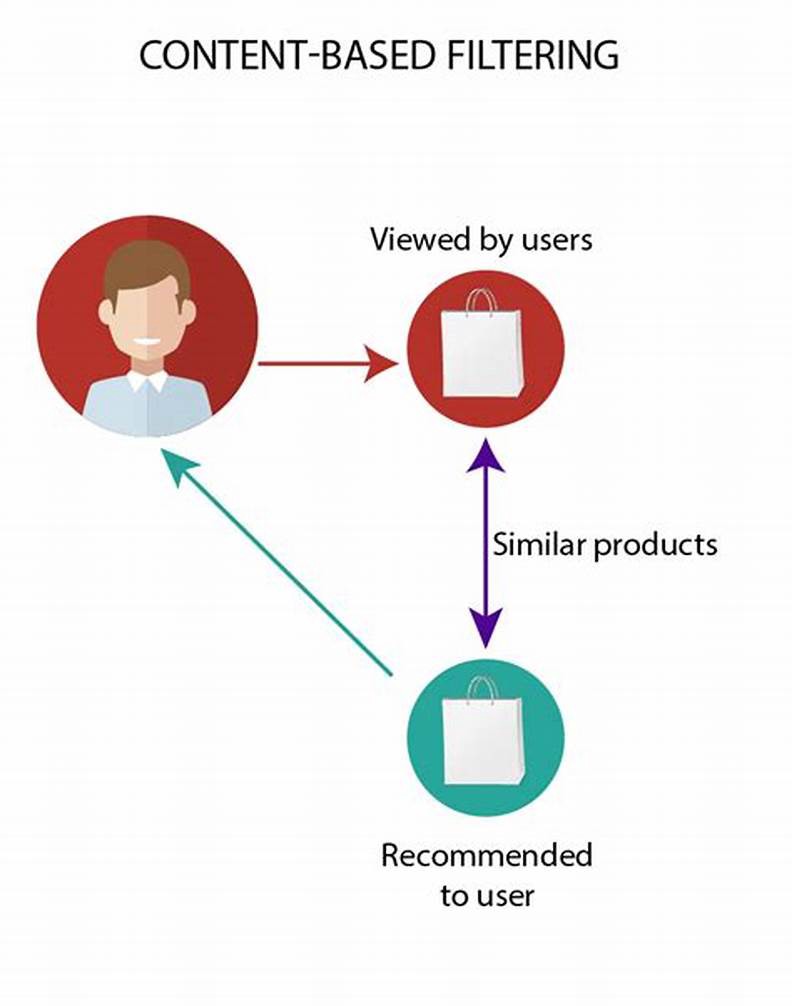
Tóm lại, hệ thống gợi ý thường được chia thành những loại sau:

* Content-based filtering
* Collaborative filtering
* Hybrid (hệ thống lai giữa hai loại trên)

## Content-based Filtering

Bộ lọc dựa trên nội dung là phương pháp sử dụng các thuộc tính của sản phẩm để đề xuất các sản phẩm khác tương tự như cái mà người dùng thích, dựa trên các hành động trước đây của họ hoặc những phản hồi mà họ đã gửi. Nói cách khác, các thuật toán này cố gắng đề xuất các mục tương tự như các mục mà người dùng thích trong quá khứ (hoặc đang kiểm tra trong hiện tại). Đặc biệt, các ứng viên khác nhau được so sánh với các mục được đánh giá trước đây bởi người dùng và các mục phù hợp nhất được khuyến khích. Vấn đề chính của phương pháp này là bị giới hạn bởi nội dung của sản phẩm, chỉ tư vấn được các sản phẩm tương tự trong cùng mục nội dung.

*Ví dụ:* Một người dùng xem rất nhiều các bộ phim về cảnh sát hình sự, vậy thì gợi ý một bộ phim trong cơ sở dữ liệu có chung đặc tính hình sự với *người dùng* này. Cách tiếp cận này yêu cầu việc sắp xếp các *sản phẩm* vào từng nhóm hoặc đi tìm các đặc trưng của từng *sản phẩm*. Tuy nhiên, có những *sản phẩm* không có nhóm cụ thể và việc xác định nhóm hoặc đặc trưng của từng item đôi khi là bất khả thi.



Hình .1: Kỹ thuật Content-based Filtering

### Hồ sơ sản phẩm

Trong các hệ thống content-based, tức dựa trên nội dung của mỗi sản phẩm, chúng ta cần xây dựng một bộ hồ sơ (profile) cho mỗi sản phẩm. Bộ hồ sơ này được biểu diễn dưới dạng toán học là một feature vector (vector thuộc tính).

### Sử dụng tích vô hướng như một thước đo độ tương đồng (Similarity Measure)

Giả sử người dùng được gán là x và ứng dụng được gán là y, cả hai là vector nhị phân. Ta có công thức:

Một thuộc tính (feature) xuất hiện trong cả x và y được tính là thêm 1 vào tổng. Hay nói cách khác, <x,y> là số thuộc tính đồng thời có trong cả hai vector x và y. Tích vô hướng cao hơn biểu thị rằng thuộc tính đó phổ biến hơn, do đó có mức độ tương đồng cao hơn

### Ưu và nhược điểm

* **Ưu điểm:**
  + Mô hình không cần bất kỳ dữ liệu nào về những người dùng khác, vì các đề xuất chỉ dành riêng cho người dùng này. Điều này giúp dễ dàng mở rộng quy mô tới lượng lớn người dùng.
  + Mô hình có thể nắm bắt sở thích cụ thể của người dùng và có thể đề xuất các item thích hợp mà rất ít người dùng khác quan tâm.
* **Nhược điểm:**
* Kỹ thuật này đòi hỏi nhiều kiến thức chuyên ngành về một lĩnh vực nào đó.
* Mô hình chỉ có thể đưa ra các đề xuất dựa trên sở thích hiện có của người dùng.

## Collaborative Filtering

Ý tưởng cơ bản của bộ lọc cộng tác là xác định mức độ quan tâm của một người dùng tới một sản phẩm dựa trên các người dùng khác gần giống với người dùng này. Việc gần giống nhau giữa các người dùng có thể được xác định thông qua mức độ quan tâm của các người dùng này tới các sản phẩm khác mà hệ thống đã biết.

Những câu hỏi quan trọng nhất trong một hệ thống Collaborative Filtering là:

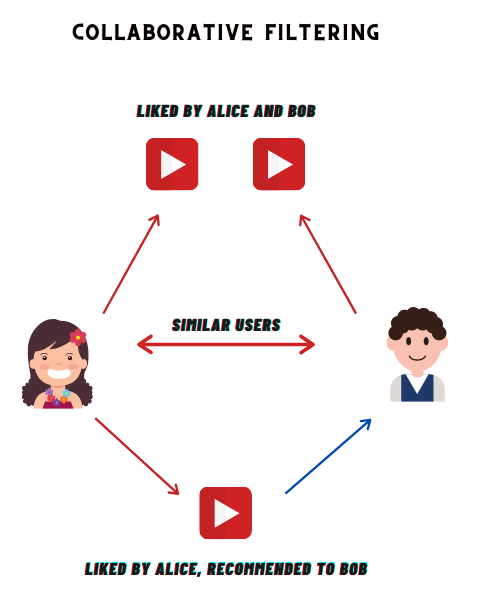
1. Làm thế nào để xác định người dùng hay sản phẩm nào tương đồng với những người dùng và sản phẩm khác?
2. Khi đã xác định được các người dùng gần giống nhau rồi, làm thế nào để dự đoán được mức độ quan tâm của một người dùng lên một sản phẩm?
3. Làm cách nào để bạn đo lường độ chính xác của xếp hạng mà bạn tính toán?

Việc xác định mức độ quan tâm của mỗi người dùng tới một item dựa trên mức độ quan tâm của người dùng tương tự tới sản phẩm đó còn được gọi là User-user collaborative filtering. Có một hướng tiếp cận khác được ho là làm việc hiệu quả hơn là Item-item collaborative filtering. Trong hướng tiếp cận này, thay vì xác định mức độ tương đồng của người dùng, hệ thống sẽ xác định mức độ tương đồng của sản phẩm. Từ đó hệ thống gợi ý những sản phẩm gần giống với những sản phẩm mà người dùng có mức độ quan tâm cao.

Một điều quan trọng cần lưu ý là trong một cách tiếp cận hoàn toàn dựa trên bộ lọc hợp tác, sự tương đồng không được tính bằng các yếu tố như tuổi của người dùng, thể loại các bộ phim hoặc bất kỳ dữ liệu nào khác về người dùng hoặc item. Nó chỉ tính toán trên cơ sở xếp hạng (rating) mà người dùng cung cấp cho sản phẩm.

*Ví dụ:* Hai người dùng được coi là tương đồng nếu họ cho những đánh giá giống nhau đối với 10 bộ phim mặc dù có sự khác biệt lớn trong tuổi tác của họ.

Về câu hỏi thứ 3, một trong những cách tiếp cận để đo độ chính xác của kết quả là Root Mean Square Error (RMSE), trong đó dự đoán xếp hạng cho bộ dữ liệu test của các cặp người dùng - sản phẩm mà đã được định sẵn giá trị. Sự khác nhau giữa giá trị đã biết và giá trị dự đoán sẽ là lỗi (error). Bình phương tất cả các giá trị lỗi sau đó lấy trung bình, tiếp theo căn bậc hai giá trị trung bình đó ta được RMSE.



Hình .2: Kỹ thuật Collaborative Filtering

### User-user collabrative Filtering vs Item-item collaborative Filtering

**User-based:** Đối với một người dùng **U**, với một tập hợp những người dùng tương tự được xác định dựa trên các vector đánh giá (rating) bao gồm các xếp hạng đánh giá item nhất định, đánh giá cho một item **I**, cái mà chưa được đánh giá, sẽ được lấy từ **N** người dùng trong danh sách tương đồng đã đánh giá item **I** và tính toán đánh giá dựa trên **N** đánh giá này.

**Item-based:** Đối với một item **I**, với một tập hợp các item tương tự được xác định dựa trên các vector đánh giá, đánh giá của người dùng **U**, người chưa đánh giá item **I**, sẽ được lấy dựa theo **N** items trong danh sách tương đồng, những item này đã được đánh giá bởi người dùng **U** và tính toán đánh giá dựa trên N đánh giá này

### Ưu và nhược điểm

* **Ưu điểm:**

Không quá cần thiết kiến thức chuyên ngành

Mô hình có thể giúp người dùng khám phá những sở thích mới. Hệ thống học máy có thể không biết người dùng quan tâm đến một mặt hàng nhất định, nhưng mô hình vẫn có thể đề xuất những mặt hàng đó vì những người dùng tương tự cũng quan tâm đến mặt hàng đó.

* **Nhược điểm:**

Khó để bao hàm cả những thuộc tính bên lề cho item. Ví dụ với các đề xuất phim, các thuộc tính bên lề có thể bao gồm quốc gia hoặc tuổi. Bao hàm các thuộc tính bên lề có sẵn giúp cải thiện chất lượng của mô hình

### Matrix Factorization

Matrix Factorization là một hướng tiếp cận khác của Collaborative Filtering, còn gọi là Matrix Decomposition, nghĩa là gợi ý bằng “kỹ thuật phân rã ma trận”.

Kỹ thuật phân rã ma trận là phương pháp chia một ma trận lớn X thành hai ma trận có kích thước nhỏ hơn là W và H, sao cho ta có thể xây dựng lại X từ hai ma trận nhỏ hơn này càng chính xác càng tốt.

Có thể hiểu rằng, ý tưởng chính của Matrix Factorization là đặt items và users vào trong cùng một không gian thuộc tính ẩn.

Áp dụng phương pháp này vào bài toán gợi ý, chúng ta có x là một vector của item profile.

Mục tiêu của chúng ta là tìm một vector w tương ứng với mỗi user sao cho ratings đã biết của user đó cho item (y) xấp xỉ với:

## Hệ thống lai (Hybrid)

Các phương pháp tiếp cận lai (hybrid) có thể được thực hiện bằng nhiều cách: bằng cách đưa ra các dự đoán dựa trên nội dung và dựa trên cộng tác một cách riêng biệt và sau đó kết hợp chúng; bằng cách thêm các tính năng dựa trên nội dung vào một cách tiếp cận hợp tác (và ngược lại). Một số nghiên cứu thực nghiệm so sánh hiệu suất của phương pháp lai với các phương pháp hợp tác và dựa trên nội dung thuần tuý và chứng minh rằng các phương pháp lai có thể đưa ra các gợi ý chính xác hơn các phương pháp thuần tuý. Một số cách kết hợp như sau:

* Sử dụng cả hai phương pháp, sau đó dùng hai kết quả thu được để quyết định:
  + Sử dụng kết quả của phương pháp nào tốt hơn tuỳ từng thời điểm
  + Dùng cả hai kết quả để đánh giá
* Xây dựng hệ thống lọc cộng tác có sử dụng các đặc trưng của phương pháp lọc dựa trên nội dung.
* Xây dựng hệ thống lọc dựa trên nội dung có sử dụng các đặc trưng của phương pháp lọc cộng tác.
* Xây dựng hệ thống két hợp cả lọc cộng tác và lọc dựa trên nội dung, chia làm nhiều pha, mỗi pha là lọc cộng tác hoặc lọc dựa trên nội dung.
* **Cosine Similarity**

Đây là hàm được sử dụng nhiều nhất để đo mức độ tương đồng.

Trong đó: là những vector tương ứng với người dùng 1,2.

Mức độ tương đồng của hai vector là 1 số trong đoạn [-1, 1]. Giá trị bằng 1 thể hiện hai vector hoàn toàn tương đồng nhau. Hàm số cos của một góc bằng 1 nghĩa là góc giữa hai vector bằng 0, tức một vector bằng tích của một số với với vector còn lại. Giá trị cos bằng -1 thể hiện hai vector này hoàn toàn trái ngược nhau. Điều này cũng hợp lí, tức khi hành vi của hai người dùng là hoàn toàn ngược nhau thì mức độ tương đồng giữa hai vector đó là thấp nhất.

## Chức năng của hệ thống gợi ý phim

*Thu thập dữ liệu*: Hệ thống thu thập dữ liệu về phim từ nhiều nguồn khác nhau như cơ sở dữ liệu phim, trang web đánh giá phim, hoặc dữ liệu người dùng cung cấp.

*Xây dựng hồ sơ người dùng*: Hệ thống tạo hồ sơ cho từng người dùng dựa trên lịch sử xem phim, đánh giá và sở thích cá nhân. Thông tin về thị hiếu phim, thể loại ưa thích, diễn viên, đạo diễn, và các yếu tố khác có thể được sử dụng để tạo hồ sơ này.

*Xây dựng hồ sơ phim*: Hệ thống phân tích các thông tin về phim như thể loại, diễn viên, đạo diễn, nhà sản xuất, và mô tả nội dung để tạo hồ sơ cho từng bộ phim.

*Tính toán độ tương tự giữa các phim*: Hệ thống tính toán độ tương tự giữa các phim dựa trên hồ sơ của chúng. Các phương pháp tính toán tương tự như cosine similarity hoặc Jaccard similarity thường được sử dụng.

*Gợi ý phim*: Khi một người dùng yêu cầu gợi ý phim, hệ thống sẽ sử dụng thông tin từ hồ sơ người dùng và hồ sơ phim để gợi ý các phim mà người dùng có thể thích. Các phim được đề xuất được xếp hạng dựa trên mức độ tương tự với sở thích của người dùng.

*Cập nhật hồ sơ*: Hệ thống liên tục cập nhật hồ sơ người dùng và hồ sơ phim khi có dữ liệu mới, đảm bảo rằng các gợi ý luôn phản ánh sở thích và xu hướng mới nhất của người dùng.

*Đánh giá và phản hồi*: Hệ thống cung cấp cơ chế cho người dùng đánh giá các gợi ý và cung cấp phản hồi, từ đó giúp cải thiện chất lượng của các gợi ý trong tương lai.

## Tác dụng của hệ thống gợi ý

*Tăng trải nghiệm người dùng*: Hệ thống gợi ý phim giúp người dùng khám phá các bộ phim mới và phong phú hơn, từ đó tăng trải nghiệm giải trí của họ trên các nền tảng phim.

*Tiết kiệm thời gian*: Thay vì phải tìm kiếm qua hàng ngàn tùy chọn phim, người dùng có thể dễ dàng nhận được các đề xuất phim phù hợp với sở thích cá nhân của họ, giúp họ tiết kiệm thời gian và nỗ lực.

*Tăng doanh số bán hàng hoặc doanh thu*: Bằng cách gợi ý các bộ phim tương tự với những bộ phim mà người dùng đã thích trước đó, hệ thống có thể tăng khả năng tiêu thụ của người dùng, từ đó tăng doanh số bán hàng hoặc doanh thu cho các dịch vụ phim.

*Tạo sự đa dạng và phong phú*: Hệ thống gợi ý phim có thể giúp mở rộng phạm vi xem phim của người dùng bằng cách giới thiệu họ với các thể loại, diễn viên, đạo diễn mới mà họ có thể chưa biết đến trước đó.

*Tăng tương tác và cam kết*: Bằng việc cung cấp các đề xuất phim chính xác và cá nhân hóa, hệ thống có thể tăng tương tác và cam kết của người dùng với nền tảng phim, từ đó tạo ra một môi trường giải trí thu hút hơn.

*Nâng cao trải nghiệm khách hàng*: Hệ thống gợi ý phim có thể giúp các dịch vụ phim tăng cường mối quan hệ với khách hàng bằng cách cung cấp các giải pháp giải trí cá nhân hóa và phù hợp.

# PHÂN TÍCH HỆ THỐNG

## Hoạt động của hệ thống

Là hệ thống được tích hợp vào các trang web xem phim, với chức năng gợi ý về các bộ phim cho người dùng, dựa trên nội dung liên quan với nội dung mà người dùng muốn tìm kiếm.

Các yêu cầu chức năng: Gợi ý trong khi gõ (các bộ phim có tên tương tự), Hiển thị các bộ phim trong dataset, tìm kiếm những bộ phim có tên giống tên tương tự người dùng nhập và poster liên quan.

## Mô tả chức năng

#### **Biểu đồ usecase chính (phần front-end).**



Hình .1: Biểu đồ Usecase chính

## Mô tả use case

* **Mô tả use case Đăng nhập**

Mô tả hệ thống: Ca này cho phép người dùng đăng nhập để sử dụng hệ thống.

* **Luồng sự kiện:**
* **Luồng cơ bản:**
  + Usecase này bắt đầu khi người dùng chọn “Log in” tại dropdown trong sidebar.
  + Người dùng nhập đầy đủ thông tin cá nhân (Tài khoản, mật khẩu) sau đó nhấn nút “Log in”, hệ thống sẽ so sánh thông tin người dùng nhập sau đó đưa ra các thông báo phù hợp, nếu chưa nhập gì thì sẽ thông báo “You haven’t typed anything to Log in”, sai mật khẩu sẽ báo sai mật khẩu, sai tài khoản sẽ báo sai tài khoản, bỏ trống 1 trong 2 ô trên sẽ thông báo về độ dài nhập liệu của từng ô.

.

* **Luồng rẽ nhánh:**

Trong quá trình truy vấn hệ thống cơ sở dữ liệu lỗi, hệ thống sẽ đưa ra thông báo lỗi.

* **Các yêu cầu đặc biệt:** Không
* **Tiền điều kiện:** Không
* **Hậu điều kiện:** Không
* **Các điểm mở rộng:** Không
* **Mô tả use case Đăng kí**

Mô tả hệ thống: Usecase này cho phép người dùng có thể xem hết tất cả các phim có trong dataset.

* **Luồng sự kiện:**
* **Luồng cơ bản:**
  + Usecase này bắt đầu khi người dùng chọn “Sign up” tại dropdown trong sidebar.
  + Người dùng nhập đầy đủ thông tin cá nhân (Tài khoản, mật khẩu) sau đó nhấn nút “Sign up”, hệ thống sẽ so sánh thông tin người dùng nhập sau đó đưa ra các thông báo phù hợp, nếu chưa nhập gì thì sẽ thông báo “You haven’t typed anything to Sign up, bỏ trống 1 trong 2 ô trên sẽ thông báo về độ dài nhập liệu của từng ô.
* **Luồng rẽ nhánh:**

Usecase này có thể kết thúc khi người dùng chọn Log in ở sidebar.

* **Các yêu cầu đặc biệt:** Không
* **Tiền điều kiện:** Không
* **Hậu điều kiện:** Không
* **Các điểm mở rộng:** Không
* **Mô tả use case Show Recommendation**

Mô tả hệ thống: Usecase này sẽ gợi ý cho người dùng những bộ phim liên quan và poster với tên phim mà người dùng đã nhập.

* **Luồng sự kiện:**
* **Luồng cơ bản:**
  + Usecase này bắt đầu khi người dùng nhấn vào “Show recommendation” ở trên màn hình.
* Hệ thống sẽ hiển thị các bộ phim liên quan và posters.
* **Luồng rẽ nhánh:**
  + Use case sẽ kết thúc khi người dùng chọn một bộ phim trong menu.
  + Nếu không có nội dung được nhập vào trong thanh tìm kiếm, thì hệ thống sẽ hiển thị phim dựa trên dataset (mặc định).
* **Các yêu cầu đặc biệt:** Không
* **Tiền điều kiện:** Người dùng đã đăng nhập hệ thống thành công
* **Hậu điều kiện:** Không
* **Các điểm mở rộng:** Không

## Phân tích các use case

* **Phân tích use case Đăng nhập**

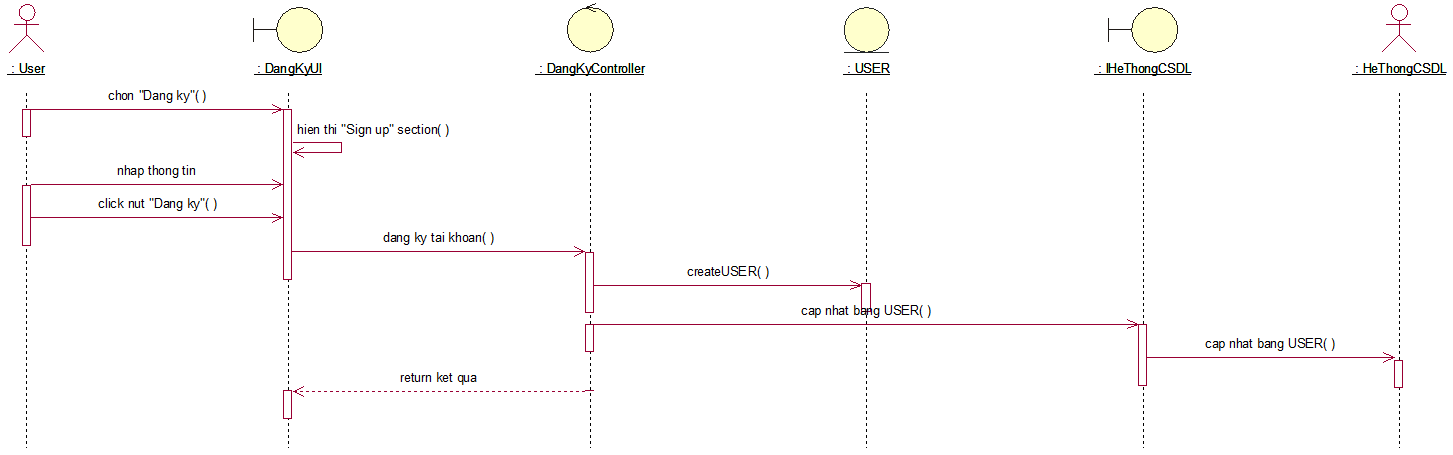


Hình .2: Biểu đồ trình tự Usecase Đăng nhập



Hình 3.3: Biểu đồ lớp phân tích usecase Đăng nhập

* **Phân tích usecase Đăng kí**



Hình .4: Biểu đồ trình tự usecase Đăng ký



Hình .5: Biểu đồ lớp phân tích usecase Đăng ký

* **Phân tích use case “Nhấn Show Recommendation”**



Hình .6: Biểu đồ trình tự usecase Show Recommendation



Hình .7: Biểu đồ lớp phân tích usecase Show Recommendation

## Biểu đồ tổng hợp



Hình .8: Biểu đồ các lớp Entity của hệ thống

# THIẾT KẾ GIAO DIỆN, CÀI ĐẶT VÀ KIẾM THỬ

## Tổng quan hệ thống

Jupyter Notebook là một nền tảng tính toán khoa học mã nguồn mở, có thể sử dụng để tạo và chia sẻ các tài liệu có chứa code trực tiếp, phương trình, trực quan hoá dữ liệu và văn bản tường thuật. Với Jupyter Notebook, người dùng có thể đưa dữ liệu, code, hình ảnh, công thức, video… vào trong cùng một file, gióp cho việc trình bày trở nên dễ dàng hơn. Em sử dụng công cụ này để trình bày thuật toán và các bước để xây dựng hệ thống.

Em sử dụng ngôn ngữ lập trình Python . Thư viện mạnh mẽ và giúp em có thể mang các thuật toán học máy (Machine learning) và trong hệ thống thống gợi ý đó chính là Scikit-learn (Sklearn). Thư viện giúp xử lý và trực quan hoá dữ liệu như Pandas, Numpy, Matplotlib. Và không thể thiếu Surprise, một thư viện Python dùng để xây dựng và phân tích hệ thống gới ý sử dụng dữ liệu đánh giá.

Hệ thống sẽ gợi ý phim dựa trên 3 cách tiếp cận:

1. Demographic filtering
2. Content-based filtering
3. Collaborative filtering

Để giao diện hoá bài toán gợi ý phim, Em sử dụng thư viện Streamlit để xây dựng giao diện cho bài toán. Ngoài ra còn thư viện Pickle, một thư viện phục vụ hỗ trợ cho Streamlit.

## Một số khái niệm

### Thư viện Scikit-learn

Scikit-learn ban đầu được đề xuất bởi David Cournapeau trong một dự án mùa hè của Google vào năm 2007.

Scikit-learn (Sklearn) là thư viện mạnh mẽ nhất dành cho các thuật toán học máy được viết trên ngôn ngữ Python. Thư viện cung cấp một tập các công cụ xử lý các bài toán machine learning và statistical modeling gồm: Classification, regression, clustering và dimensionality reduction.

Scikit-learn hỗ trợ mạnh mẽ trong việc xây dựng các sản phẩm. Nghĩa là thư viện này tập trung sâu vào việc xây dựng các yếu tố: dễ sử dụng, dễ code, dễ tham khảo, dễ làm việc, hiệu quả cao.

Mặc dù được viết cho Python nhưng thực ra các thư viện nền tảng của scikit-learn lại được viết dưới các thư viện của C để tăng hiệu suất làm việc.

### Thư viện Surprise

Thư viện Surprise đại diện cho Simple Python Recommendation System Engine.

Thư viện này được thiết kế cho những mục đích sau:

1. Cung cấp cho người dùng quyền kiểm soát hoàn hảo trong việc sử dụng. Với tài liệu hướng dẫn rõ ràng và chi tiết nhất có thể về các thuật toán có trong thư viện.
2. Giảm bớt khó khăn khi xử lý Dataset. Người dùng có thể sử dụng cả tập dữ liệu tích hợp (Movielens, Jester) và tập dữ liệu tuỳ chỉnh của riêng họ.
3. Cung cấp các thuật toán dự đoán sẵn sàng sử dụng khác nhau như thuật toán đường cơ sở, thuật toán lân cận, thừa số hoá ma trận (SVD, PMF, SVD++, NMF) và nhiều thuật toán khác. Ngoài ra còn có những phương pháp đo độ tương đồng như Cosine, MSD, pearson… được tích hợp sẵn.
4. Dễ dàng triển khai các ý tưởng thuật toán mới.
5. Cung cấp các công cụ để đánh giá, phân tích và so sánh hiệu suất của thuật toán.

### Thư viện Streamlit

Streamlit là một open-source python lib, giúp người sử dụng dễ dàng tạo ra một web app cho Machine Learning và Data Science. Đối với một người có rất ít kiến thức về HTML, CSS, JavaScript thì đây có lẽ là công cụ rất tuyệt vời phù hợp để demo các sản phẩm liên quan tới AI.

### Thư viện Pickle

Một nhà khoa học dữ liệu có thể sẽ sử dụng tập hợp dữ liệu ở dạng đối tượng kiểu từ điển (dictionary), DataFrames hoặc bất kỳ kiểu dữ liệu nào khác. Khi làm việc với chúng, bạn có thể muốn lưu chúng vào một tệp để có thể sử dụng sau này hoặc gửi cho người khác. Thư việc Pickle của Python được sử dụng cho công việc này. Pickle sẽ thực hiện chuyển đổi các đối tượng sang một dạng byte nhị phân để chúng có thể được lưu vào một tệp và được tải lại trong một chương trình sau này. Cụ thể ở bài toán lần này, tập pickle sẽ được tải lại vào chương trình sử dụng streamlit. Pickle giúp lưu thông tin giữa các phiên trong quá trình sử dụng web app xây dựng dựa trên streamlit.

### Xử lý ngôn ngữ tự nhiên (NLP)

Hệ thống gợi ý phim có sử dụng một số kỹ thuật xử lý ngôn ngữ tự nhiên.

Xử lý ngôn ngữ là một kĩ thuật quan trọng nhằm giúp máy tính hiểu được ngôn ngữ của con người, qua đó hướng dẫn máy tính thực hiện và giúp đỡ con người trong những công việc có liên quan đến ngôn ngữ như: dịch thuật, phân tích dữ liệu văn bản, nhận dạng tiếng nói, tìm kiếm thông tin, tóm tắt văn bản, và rất nhiều ứng dụng khác.

Một trong những kĩ thuật để xử lý ngôn ngữ tự nhiên là: **tf-idf**.

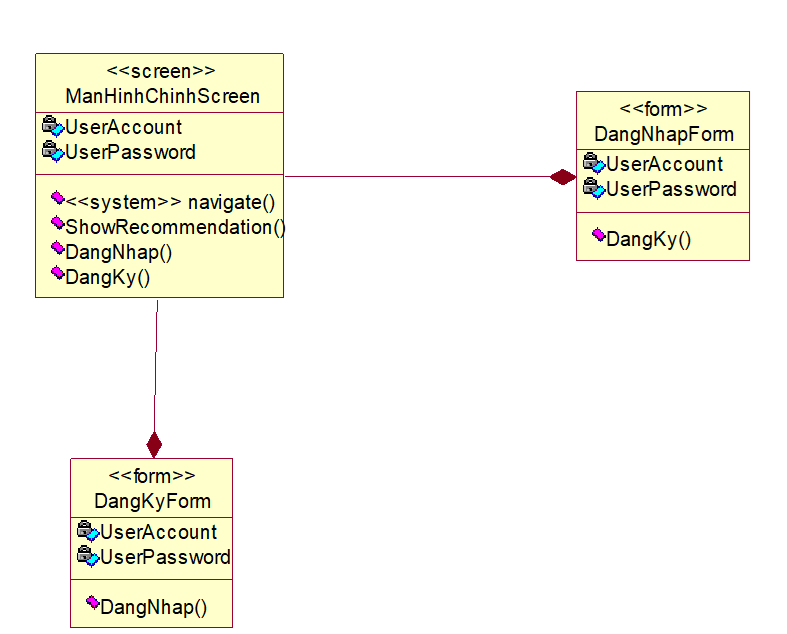
**TF-IDF**

Term frequency – inverse document ferquency là trọng số của một từ trong văn bản thu được qua thống kê thể hiện mức độ quan trọng của từ này trong một văn bản mà bản thân văn bản đang xét nằm trong một tập hợp các văn bản.

## Thiết kế giao diện

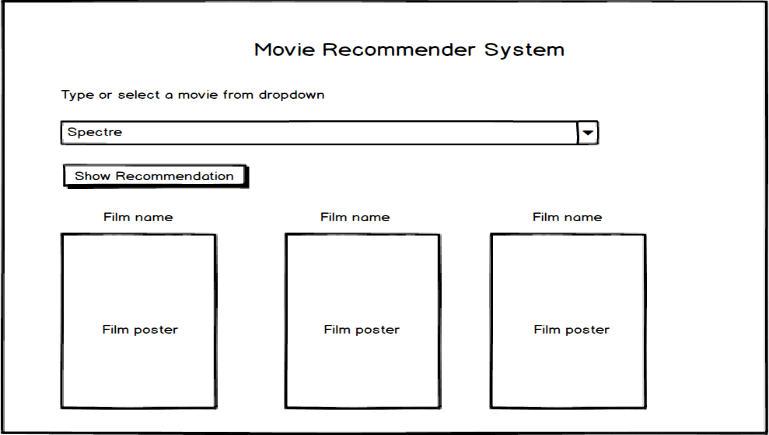
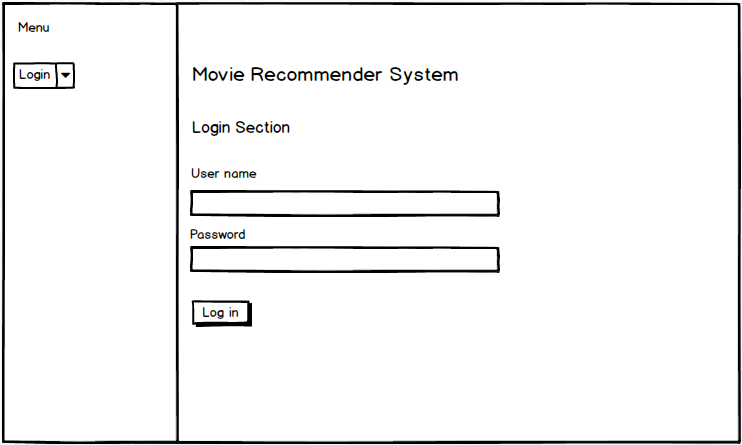
### Biểu đồ các lớp phân tích

Vì chỉ có một màn hình nên ta có biểu đồ lớp màn hình cho cả hệ thống như sau:



Hình .1: Biểu đồ lớp phân tích giao diện

### Giao diện usecase “Đăng nhập”



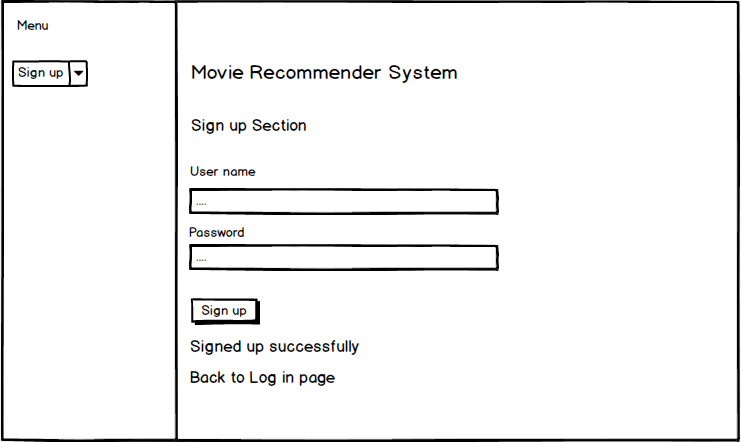
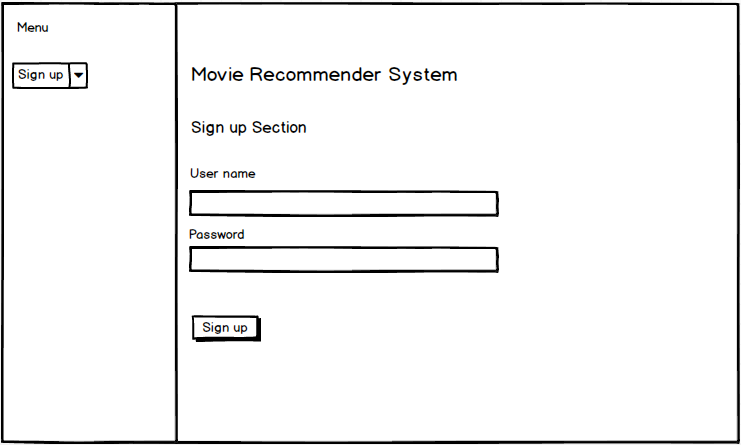
Log out

Hình .2: Hình dung màn hình usecase Đăng nhập



Hình .3: Biểu đồ cộng tác màn hình usecase Đăng nhập

### Giao diện usecase “Đăng kí”

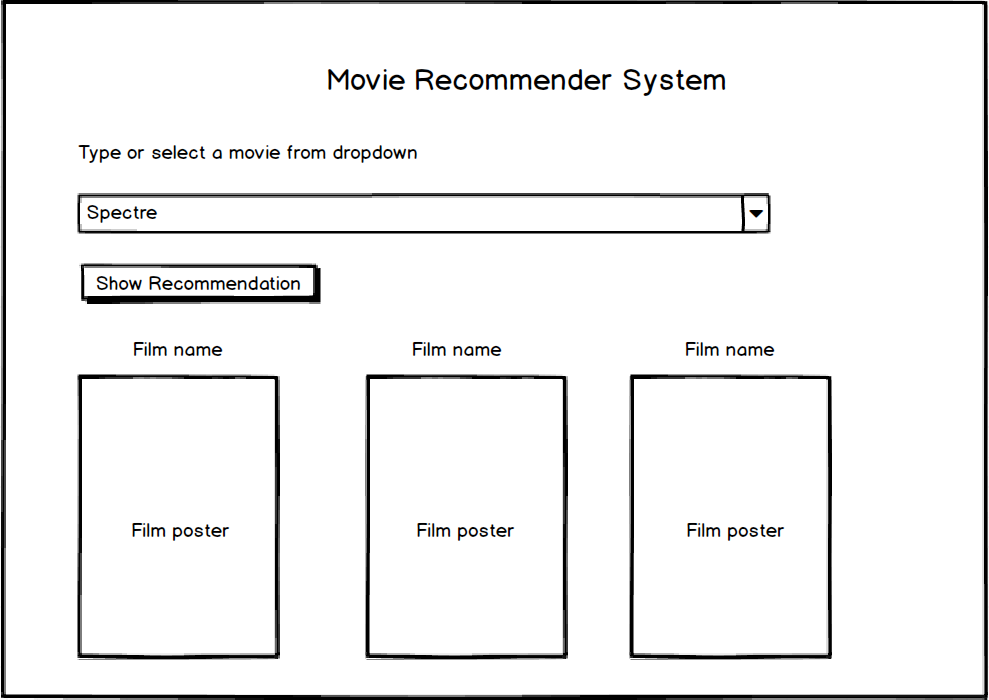


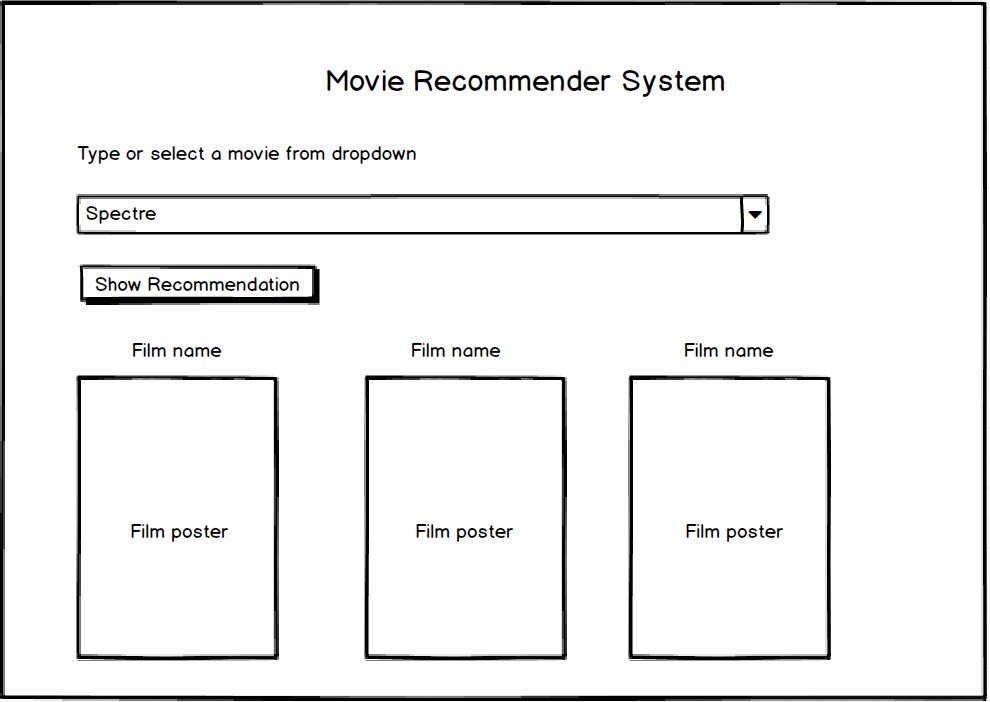
Hình .4: Hình dung màn hình usecase Đăng ký



Hình .5: Biểu đồ trình tự usecase Đăng ký

### Giao diện usecase “Nhấn Show Recommendation”





Hình .6: Hình dung màn hình Show Recommendation



Hình .7: Biểu đồ trình tự Show Recommendation

## Bộ dữ liệu (Dataset)

Hai bộ dữ liệu có tên **The Movies Dataset** và **TMDB 5000 Movie Dataset** được lấy trên trang Kaggle.com – một cộng đồng trực tuyến chủ yếu dành cho Data Scientist và mọi đối tượng thực hành Machine Learning.

*Link:*  [The Movies Dataset](https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset) | [TMDB 5000 Movie Dataset](https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata)

A picture containing graphical user interface

Description automatically generated

Hình .8: The Movies Dataset

Background pattern

Description automatically generated with low confidence

Hình .9: TMDB 5000 Movie Dataset

**The Movies Dataset**

* The Movies Dataset chứa metadata của tất cả 45,000 bộ phim được liệt kê trong Full MovieLens Dataset. Dataset chứa những bộ phim được khởi chiếu trước tháng 7 năm 2017.
* Dataset cũng chứa các files chứa dữ liệu của 26 triệu đánh giá từ 270,000 người dùng của tất cả 45,000 phim. Đánh giá trên thang điểm từ 1-5 và được lấy từ website chính thức của GroupLens.
* Tuy nhiên trong phạm vi đồ án tốt nghiệp, em chỉ sử dụng file *ratings\_small.csv* có trong dataset để dự đoán đánh giá của người dùng. File *ratings\_small.csv* là một phần nhỏ chứa 100,000 đánh giá từ 700 người dùng trên 9,000 phim.

**TMDB 5000 Movie Dataset**

* Metadata trên gần 5000 bộ phim từ TMDb. Dataset chứa thông tin chi tiết nhất của gần 5000 bộ phim được lấy từ TMDb.

## Xây dựng hệ thống

### Lựa chọn cơ sở dữ liệu

Đối với dữ liệu sử dụng cho việc gợi ý phim, em sử dụng cơ sở dữ liệu dạng file (flat file database) cụ thể là cơ sở dữ liệu được lưu trữ dưới định dạng file .csv .

Ngoài ra, để phục vụ cho chức năng đăng ký và đăng nhập vào hệ thống, em sử dụng thêm cơ sở dữ liệu dạng cấu trúc (structured database), cụ thể công cụ quản lý CSDL được sử dụng là SQLite -một engine cơ sở dữ liệu SQL khép kín, có độ tin cậy cao, được nhúng vào trong nhiều nền tảng, đầy đủ tính năng. Python có thư viện để truy cập đến các cơ sở dữ liệu SQLite, được gọi là sqlite3, cung cấp khả năng làm việc với cơ sở dữ liệu này. Vì sqlite3 được tích hợp sẵn trong python, nên có thể sử dụng thao tác thuận tiện mà không cần cài đặt thêm bất cứ phần mềm hỗ trợ nào.

Table

Description automatically generated

Hình .10: Lược đồ cấu trúc csdl user

### Load và khám phá dữ liệu

Import 2 thư viện dùng để xử lý và tính toán dữ liệu là pandas, numpy

Graphical user interface, text, application

Description automatically generated

Hình .11: Import thư viện xử lý, tính toán dữ liệu

Load 2 tệp dữ liệu chính được lấy từ TMDB dataset và in tên các cột có trong 2 tệp dữ liệu

Graphical user interface, text, application

Description automatically generated

Hình .12: Load và hiển thị tên các trường dữ liệu trong dataset

**tmdb\_5000\_credits.csv:**

* movie\_id: id của mỗi bộ phim
* cast: tên của các diễn viên và đóng thế
* crew: tên của đạo diễn, ban biên tập, ban kịch bản…

**tmbd\_5000\_movies.csv:**

* budget: kinh phí thực hiện bộ phim
* genre: thể loại phim
* homepage: đường dẫn tới trang chủ của phim
* id: thông tin như movie\_id trong tệp dữ liệu trên
* keywords: từ khoá hoặc tag liên quan đến bộ phim
* original\_title: tên phim trước khi dịch hoặc chuyển thể
* overview: bản mô tả vắn tắt về bộ phim
* popularity: một con số biểu thị độ phổ biến của bộ phim
* production\_countries: những nước sản xuất bộ phim
* release\_date: ngày khởi chiếu
* revenue: doanh thu trên toàn cầu của bộ phim
* runtime: thời lượng phim tính bằng phút
* status: “Released” hoặc “Rumored”
* tagline: khẩu hiệu của phim
* title: tên của bộ phim
* vote\_average: trung bình đánh giá mà bộ phim nhận được
* vote\_count: số lượng đánh giá

Tiếp theo, tiến hành nối (join) 2 tệp dữ liệu vào với nhau dựa trên trường ‘id’

Graphical user interface, text, application

Description automatically generated

Hình .13: Join 2 tệp dữ liệu dựa vào trường “Id”

### Tiếp cận theo Demographic Filtering

Trước khi bắt đầu chúng ta cần:

* Số liệu để chấm điểm hoặc đánh giá bộ phim
* Tính điểm cho mỗi bộ phim
* Sắp xếp điểm và gợi ý bộ phim có đánh giá tốt nhất cho người dùng

Có thể dùng trung bình đánh giá (average ratings) của bộ phim như là điểm để đánh giá. Tuy nhiên như vậy sẽ xảy ra hiện tượng bất công bằng bởi vì những bộ phim có trung bình đánh giá 8.9 nhưng chỉ có 3 lượt đánh giá không thể được coi là tốt hơn bộ phim chỉ có 7.8 điểm nhưng lại có 40 lượt đánh giá. Vì vậy phương pháp này sử dụng IMDB’s weighted rating (WR) để dùng làm số liệu để chấm điểm:

Trong đó:

* v là số vote của bộ phim
* m là số lượng vote tối thiểu cần có để được liệt kê trong biểu đồ
* R là đánh giá trung bình của bộ phim
* C trung bình vote của cả tệp

Giá trị v sử dụng trường dữ liệu *vote\_count*, giá trị R sử dụng trường dữ liệu *vote\_average*

Tính giá trị C:

Graphical user interface, text, application

Description automatically generated

Hình .14: Giá trị trung bình vote (vote average) của cả tệp

Tiếp theo, quyết định giá trị phù hợp cho m. Em chỉ sử dụng dữ liệu của những bộ phim có số lượng votes nhiều hơn ít nhất 90% bộ phim có trong danh sách.

Graphical user interface, text, application, Word, email

Description automatically generated

Hình .15: Số lượng vote tối thiểu cần có

m = 1838.4, vậy chúng ta chỉ lấy những bộ phim có số lượng vote lớn hơn giá trị này.

Graphical user interface, text, application

Description automatically generated

Hình .16: Số lượng phim đạt yêu cầu vote tối thiểu.

q\_movies.shape biểu thị có 481 phim đạt yêu cầu với 23 cột dữ liệu.

Tiếp theo, định nghĩa hàm *weighted\_rating()*

Text

Description automatically generated

Hình .17: Hàm tính Weighted Rating

Graphical user interface, text, application

Description automatically generatedSau đó áp dụng hàm *weighted\_rating()* cho toàn bộ dữ liệu.

Hình 4.18: Áp dụng hàm Weighted\_rating() cho toàn bộ dữ liệu

Cuối cùng, sắp xếp lại những bộ phim dựa theo số điểm vừa tính được

Graphical user interface, table

Description automatically generated with medium confidence

Hình .19: Top những bộ phim có Weighted Rating cao nhất

Ở trên là những bộ phim có WR cao nhất. Cũng sẽ là những bộ phim sẽ được gợi ý tới tất cả người dùng.

Chart

Description automatically generatedNgoài ra, hướng tiếp cận này có thể dựa trên mức độ phổ biến của bộ phim mà đưa ra gợi ý.

Hình 4.20: Khuyến nghị phim dựa trên độ phổ biến

**Kết luận:** Demographic filtering cung cấp một biểu đồ tổng quan về những bộ phim được gợi ý xem cho tất cả người dùng. Phương pháp này không chú trọng tới sở thích xem khác nhau của từng cá nhân người dùng. Đây là khi em chuyển sang hướng tiếp cận tốt hơn cho hệ thống gợi ý phim – Content Based Filtering.

### Tiếp cận theo Content-based Filtering

#### **Tính Similarity score dựa vào plot description**

Ở hướng tiếp cận này, em sẽ tính toán điểm tương đồng (similarity scores) dựa vào mô tả cốt truyện (plot description) của bộ phim và gợi ý phim dựa vào điểm tương đồng đó.

Ở đây plot description chính là cột overview trong tệp dữ liệu.

Graphical user interface, text

Description automatically generated

Hình .21: Hiển thị dữ liệu cột Overview

Em sử dụng kĩ thuật **TF-IDF** trong xử lý ngôn ngữ tự nhiên để chuyển đổi mỗi Overview thành vector từ.

Graphical user interface, text

Description automatically generated

Hình .22: Kỹ thuật xử lý ngôn ngữ tự nhiên TF-IDF

Số liệu trên biểu thị có gần 21,000 từ được dùng để mô tả cho hơn 4,800 phim. Đây là những từ có tần số xuất hiện nhiều nhất.

Tiếp theo sử dụng công thức *Cosine similarity*  để tính điểm tương đồng. Sử dụng hàm *linear\_kernel()* được cung cấp trong thư viện Sklearn.

Table

Description automatically generated

Hình .23: Tính Cosine Similarity

Sau khi đã có điểm tương đồng (similarity score), em sẽ khai báo một hàm nhận dữ liệu đầu vào là tên phim và đầu ra là 10 phim có độ tương đồng nhất. Trước tiên cần ánh xạ ngược lại tiêu đồ của phim và chỉ mục của phim đó. Nghĩa là cần một cơ chế để dựa vào tiêu đề của phim có thể xác định được chỉ mục của phim đó.

Text

Description automatically generated with medium confidence

Hình .24: Ánh xạ ngược index của phim

Tiếp theo khai báo hàm *get\_recommendations()*.

Text

Description automatically generated

Hình .25: Khai báo hàm get\_recommendation()

Hàm sẽ nhận dữ liệu đầu vào là tiêu đề phim và đầu ra là top 10 phim

liên quan tới phim đã cho nhất.

Áp dụng với phim ‘The Dark Knight Rises’.

Graphical user interface

Description automatically generated with low confidence

Hình .26: Gợi ý phim The Dark Knight Rises

**Kết luận:** Hệ thống gợi ý chưa tốt. Bởi vì ‘The Dark Knight Rises’ trả về hầu hết các bộ phim về batman. Trong khi đó những người yêu thích bộ phim này thường có xu hướng yêu thích những bộ phim khác do Christopher Nolan làm đạo diễn.

#### **Tính Similarity score dựa vào Credits, Genres, Keywords**

Nhằm cải thiện hệ thống gợi ý phim dựa trên bộ lọc nội dung. Em sử dụng những dữ liệu liên quan tới: top 3 diễn viên, đạo diễn, thể loại phim và từ khoá về bộ phim.

1. **Chuẩn hoá dữ liệu**

Dữ liệu ở các cột cast, crew, keywords, genres đang ở dạng stringified list, cần chuyển về list.

Graphical user interface, text, application

Description automatically generated

Hình .27: Chuyển dữ liệu từ stringified list sang list

Hàm *get\_director()* lấy ra tên đạo diễn của bộ phim.

Graphical user interface, text, application, Word

Description automatically generated

Hình .28: Khai báo hàm get\_director() lấy danh sách đạo diễn

Graphical user interface, text, application

Description automatically generatedHàm *get\_list()* lấy ra những 3 giá trị hàng đầu của cột cast, keywords, genres.

Hình 4.29: Hàm get\_list() lấy giá trị của cột cast, keyword, genres

Graphical user interface, text

Description automatically generatedÁp dụng hàm *get\_director()* và *get\_list()* vào tệp dữ liệu.

Hình 4.30: Áp dụng hàm get\_director() và get\_list() vào tệp dữ liệu

Dữ liệu sau khi đã định nghĩa lại.

Graphical user interface, application

Description automatically generated with medium confidence

Hình 4.31: Dữ liệu sau khi định nghĩa lại

Tiếp theo khai báo hàm *clean\_data()* nhằm format lại dữ liệu.

Text

Description automatically generated

Hình .32: K hai báo hàm clean\_data()

Áp dụng hàm *clean\_data()* vào tệp dữ liệu.

Graphical user interface, text, application

Description automatically generated

Hình .33: Áp dụng hàm clean\_data() vào tệp dữ liệu

Graphical user interface, text, application

Description automatically generatedTiếp theo, tạo và áp dụng hàm *create\_soup()* vào tệp dữ liệu.

Hình 4.34: Khai báo và áp dụng hàm creat\_soup() vào tệp dữ liệu

Hàm *create\_soup()* sẽ đưa ra một vector từ (word vectorizer) cần cho việc tính điểm tương đồng.

1. **Tính toán cosine similarity và đưa ra gợi ý**

Thực hiện ma trận hoá vector từ. Sau đó tính Cosine similarity và ánh xạ ngược chỉ mục của phim.

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình .35: Tính toán cosine similarity 2

Áp dụng với phim ‘The Dark Knight Rises’ và điểm tương đồng vừa tính được ở trên.

A picture containing table

Description automatically generated

Hình .36: Đưa ra gợi ý phim tương tự The Dark Knight Rises

**Kết luận:** Nhờ vào việc sử dụng nhiều dữ liệu hơn, hệ thống gợi ý phim của em đã đưa ra kết quả gợi ý tốt hơn. Dựa vào kết quả ở trên sẽ giống với kết luận rằng người dùng ngoài xem The Dark Kight Rise có thể xem những phim khác cùng thuộc DC comics.

### Tiếp cận theo Collaborative Filtering

Ở hướng tiếp cận này, Em sử dụng file dữ liệu *ratings\_small.csv* nhằm đưa ra đánh giá dự đoán của người dùng với id = x cho bộ phim index=y mà người dùng này chưa đánh giá, dựa vào những đánh giá của người dùng này đối với những bộ phim khác.

Import hàm cần thiết trong thư viện Surprise, đọc và hiển thị dữ liệu trong *ratings\_small.csv.*

A picture containing text

Description automatically generated

Hình .37: Import thư viện Surprise và đọc dữ liệu

Tiếp theo, chạy quy trình xác thực chéo cho thuật toán SVD (Matrix Factorization-based Algorithms), báo cáo phép đo độ chính xác, RMSE và thời gian tính toán.

Graphical user interface, text, application, email

Description automatically generated

Hình .38: Matrix Factorization Algorithms - SVD

Tiếp theo, xây dựng mô hình trên toàn bộ tệp dữ liệu.

Graphical user interface, text, application, email

Description automatically generated

Hình .39: Xây dựng mô hình trên toàn bộ tệp dữ liệu

Cuối cùng, áp dụng mô hình vừa xây dựng vào userId=1.

Graphical user interface, application, table

Description automatically generated

Hình .40: Áp dụng dự đoán rating của userId = 1

**Kết luận:** Đối với user có id=1, mô hình dự đoán user sẽ đánh giá số điểm gần 2.7 cho bộ phim có id=302 (Insomnia). Ở hệ thống gợi ý này, nó không quan tâm bộ phim là gì (hoặc nội dung của bộ phim đó như thế nào). Hệ thống hoạt động hoàn toàn dựa trên ID phim được chỉ định và cố gắng dự đoán rating dựa trên cách người dùng này dự đoán những phim khác.

## Xây dựng giao diện

### Chuẩn bị

Trước khi thực hiện xây dựng giao diện cho ứng dụng, em tiến hành tạo hai file *movies.pkl*, *similarity.pkl.*

Hai file *.pkl* được tạo bởi thư viện Pickle. Pickle trong Python cơ bản được dùng trong việc tuần tự và huỷ tuần tự hoá cấu trúc đối tượng Python. Hay nói cách khác, nó là quá trình chuyển đổi đối tượng Python sang dạng byte stream để lưu trữ thành một file, duy trì trạng thái của phần mềm qua các phiên, hoặc truyền tải dữ liệu qua mạng.

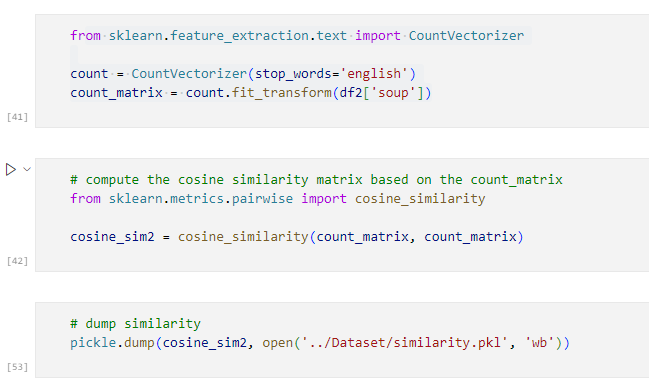
Tiến hành dump file .pkl



Hình .41: Dump movies.pkl

Dataframe có tên *df2*, là sự kết hợp giữa hai tệp dữ liệu *tmdb\_5000\_credits.csv* và *tmdb\_5000\_movies.csv*. Dump *df2* vào file có tên *movies.pkl.*

Sau khi tính được hàm cosine\_similarity, dump vào file *similarity.pkl*.



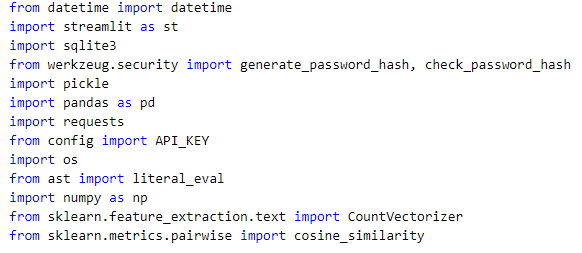
Hình .42: Dump similarity.pkl

Dữ liệu của hai file này sẽ được sử dụng để tính toán và đưa ra gợi ý phim cho người dùng.

### Xây dựng giao diện

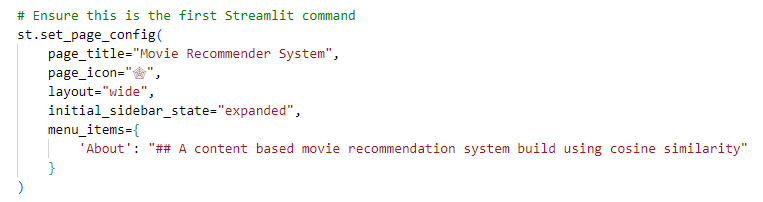
Để xây dựng giao diện cho ứng dụng, Em sử dụng thư viện Streamlit.

Import thư viện cần thiết.



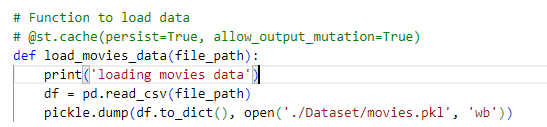
Hình .43: Import thư viện

Tạo cấu hình cho giao diện.



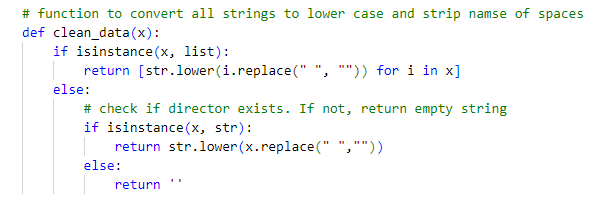
Hình .44: Cấu hình giao diện

Load dữ liệu file movie.pkl.



Hình .45: Hàm load\_movies\_data()

Loại bỏ khoảng trắng và chuyển đổi tất cả chữ thành chữ thường.



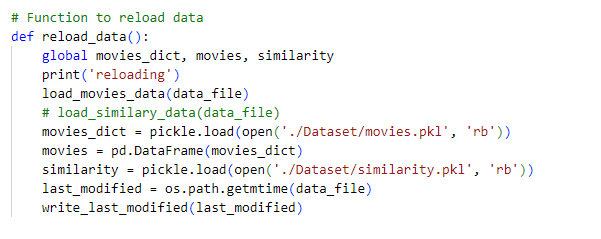
Hình .46: Hàm clean\_data()

Load dữ liệu file similarity.



Hình .47: Hàm Load\_similary\_data()

Load lại dữ liệu sau khi được thay đổi. Trong hàm reload\_data có hàm Load().



Hình .48: Hàm reload\_data()

Kết nối với cơ sở dữ liệu sqlite3, tạo hàm *create\_usertable()* để tạo bảng mới trong cơ sở dữ liệu và hàm *create\_usertable()* để thực hiện thêm user mới vào cơ sở dữ liệu.

Graphical user interface, text, application

Description automatically generated

Hình .49: Kết nối và thao tác với database

Tiếp theo tạo giao diện và validate cho chức năng đăng nhập và đăng kí của hệ thống.Text

Description automatically generated.

Hình .50: Tạo giao diện và validate chức năng đăng nhập

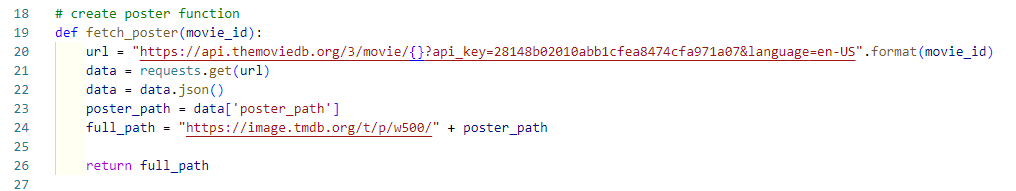
Text

Description automatically generated

Hình .51: Tạo giao diện và vadidate chức năng đăng kí

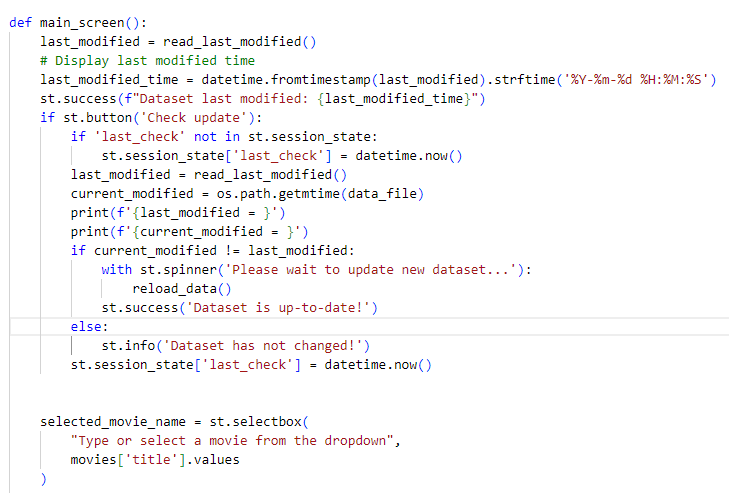
Trước khi tạo nút bấm để gợi ý phim. Chúng ta cần tạo hai hàm, một là *fetch\_poster()*  dùng để đẩy hình ảnh poster của phim từ api lên webapp, hai là *recommend()* dùng để đưa ra gợi ý phim.

Hàm này có chức năng lấy dữ liệu poster từ api được cung cấp bởi trang web *themoviedb.org*.

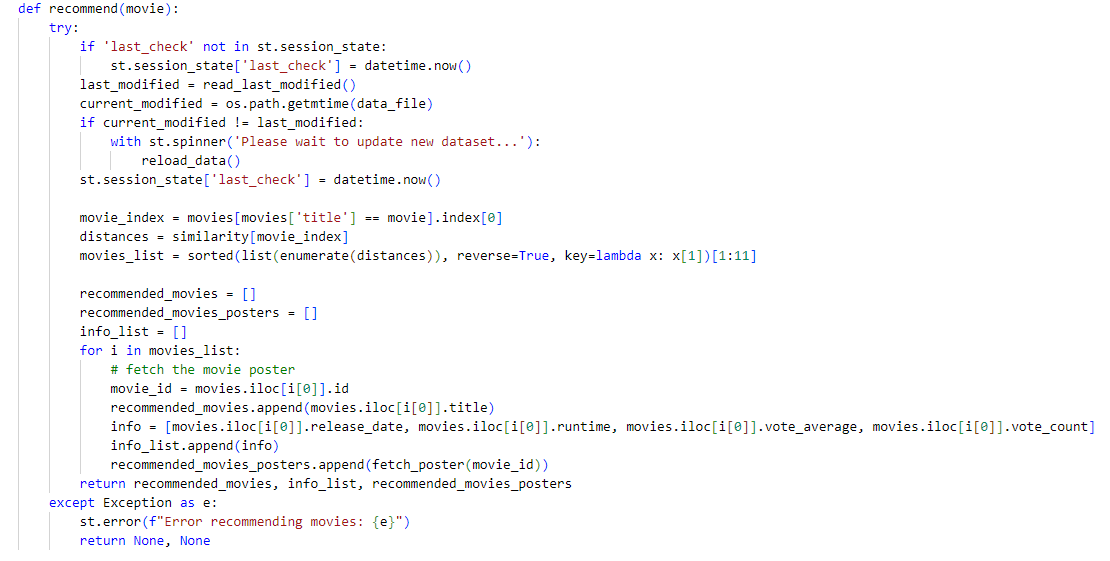


Hình 4.52: Hàm fectch\_poster()

Hàm này có chức năng hiển thị thời gian cuối cùng update bộ dữ liệu.

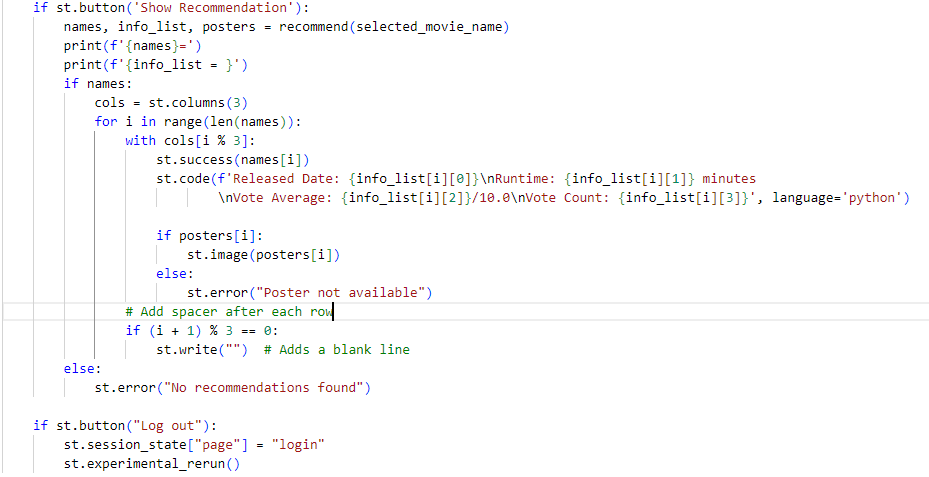


Hình .53: Hàm Main\_Screen()

Hàm *recommend()*  lấy tên phim được truyền vào, đưa ra vị trí, cũng như dùng *cosine\_similarity* đã được tính toán từ trước làm gợi ý, và đưa ra poster phim.

Hình 4.: Hàm recommend()

Tạo nút Show Recommendation sử dụng hàm *recommend()* để đưa ra gợi ý và hiển thị phim được gợi ý dưới dạng 3 cột, 3 hàng.



Hình .55: Tạo nút bấm gợi ý phim và giao diện hiển thị

Cuối cùng là phân chia trạng thái hoạt động (state session) của trang web và gán một vài dữ liệu cần thiết.



Hình .56: Define variable and state session

### Giao diện ứng dụng

Graphical user interface, application

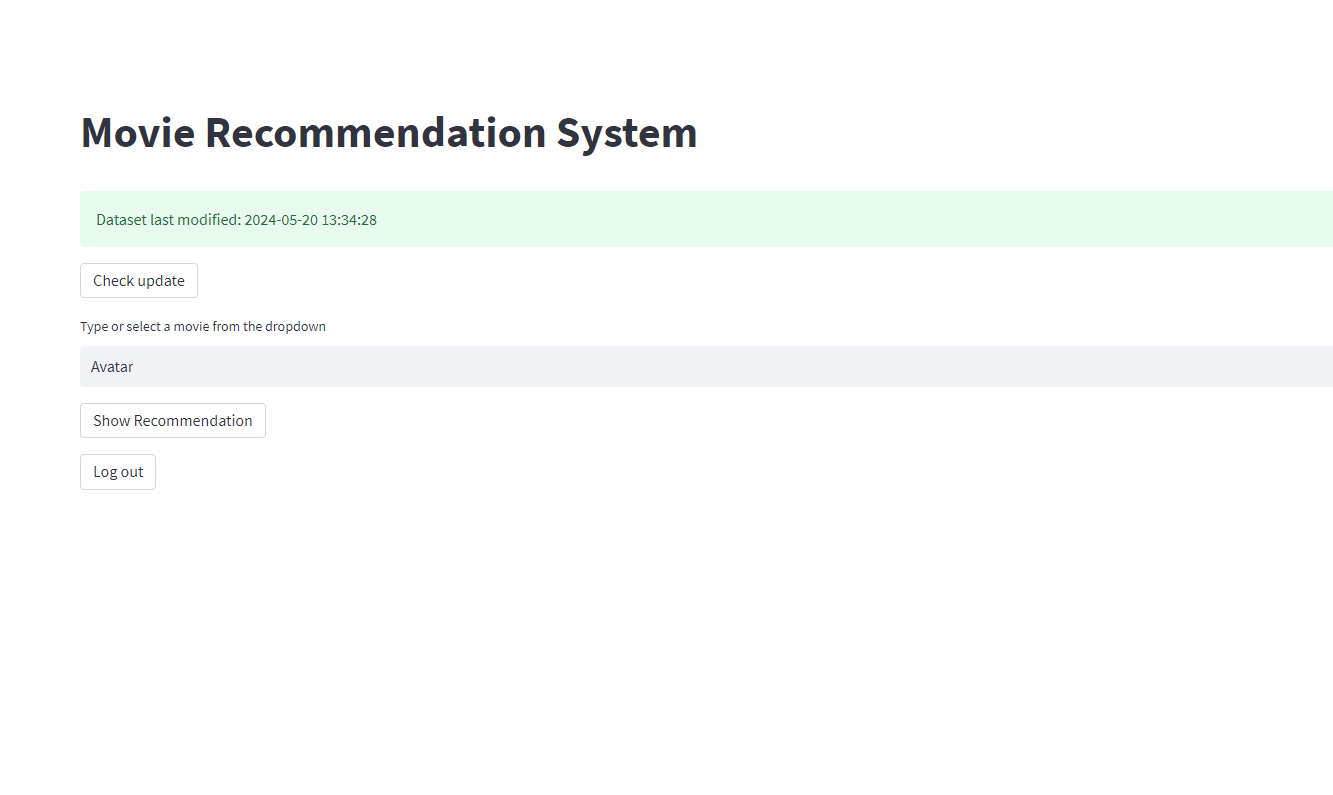
Description automatically generated

Hình .57: Giao diện đăng nhập

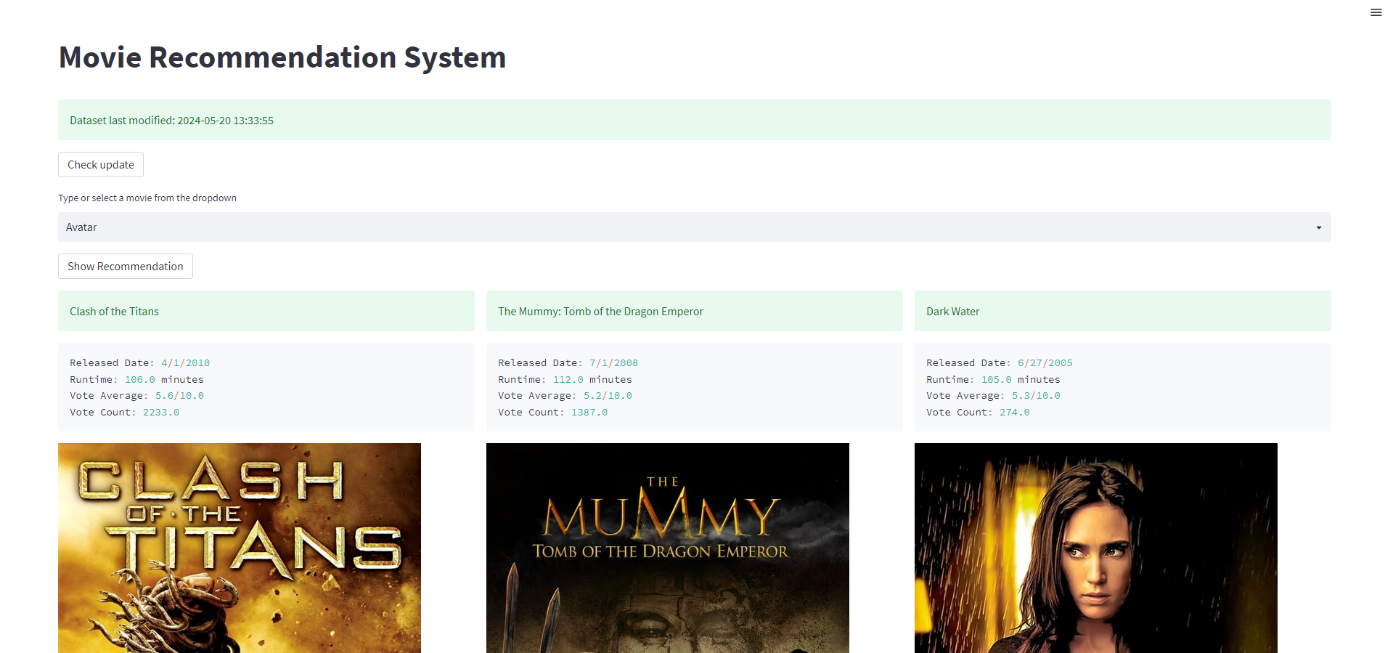
Graphical user interface, application

Description automatically generated

Hình .58: Giao diện đăng ký



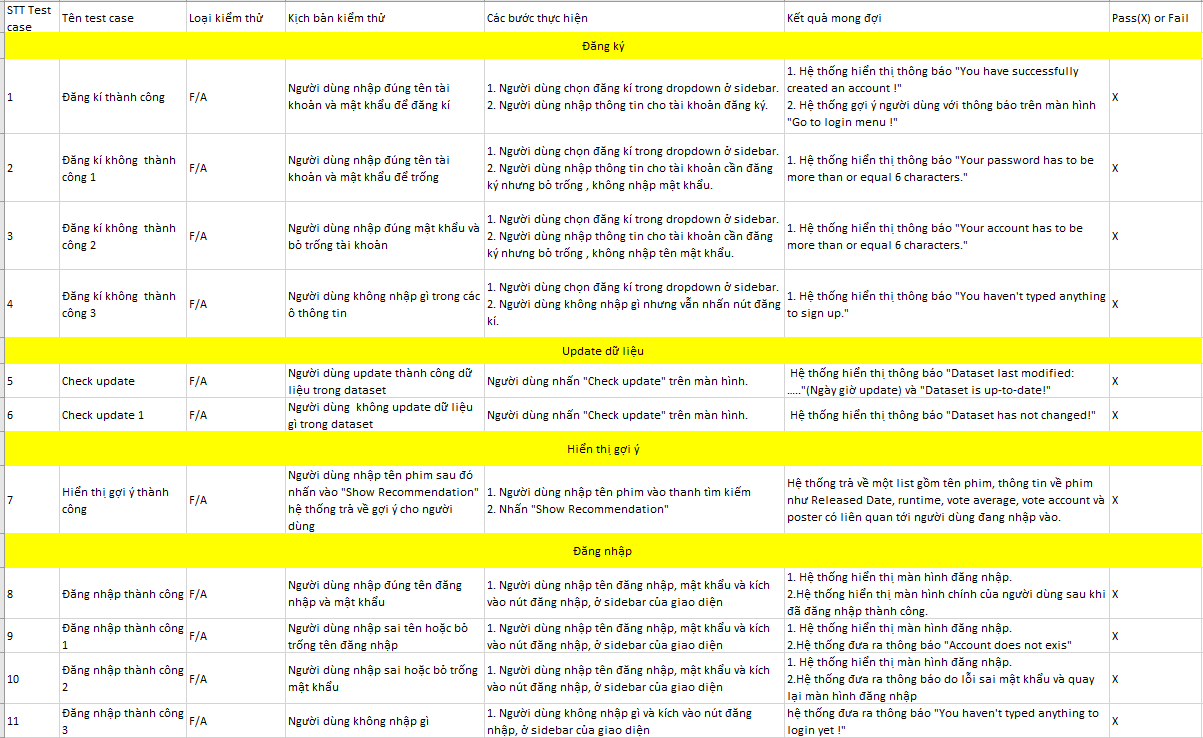
Hình .59: Giao diện hệ thống gợi ý phim



Hình .60: Giao diện sau khi đưa ra gợi ý

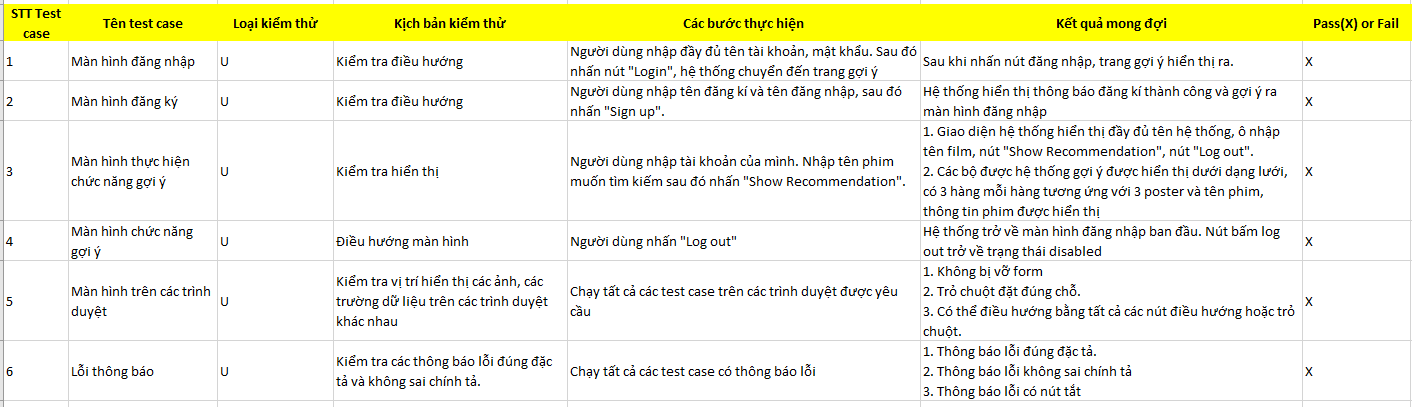
## Kiểm thử hệ thống

### Kiểm thử chức năng



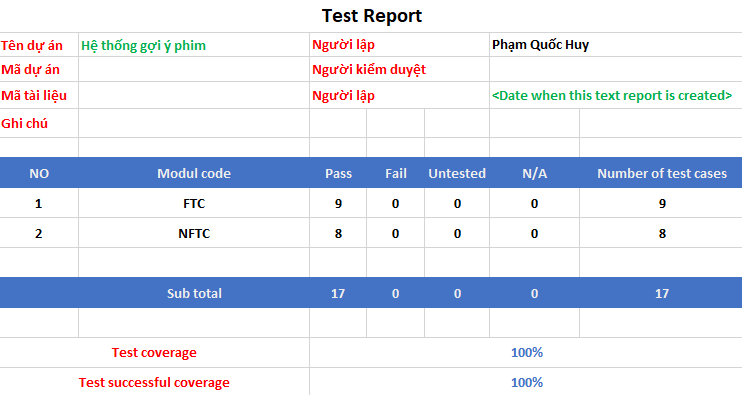
Hình .61: Bảng kiểm thử chức năng

### Kiểm thử giao diện



Hình .62: Bảng kiểm thử giao diện

### Báo cáo kiểm thử



Hình .63: Báo cáo kiểm thử

# KẾT LUẬN

Trong quá trình thực hiện đề tài cũng như hoàn thành báo cáo đồ án tốt nghiệp, em đã đúc kết và rút ra được nhiều kiến thức và kinh nghiệm quý báu cũng như những kỷ năng nền tảng, chẳng hạn như: Nắm bắt các kiến thức về hệ thống gợi ý, và những yếu tố cần thiết để xây dựng một hệ thống gợi ý. Cũng như việc áp dụng các kĩ thuật khác nhau để thực hiện đề tài. Tuy nhiên trong quá trình nghiên cứu, tìm hiểu và thực hiện, em vẫn còn những vấn đề cần cải thiện như: Tốc độ xử lý của hệ thống đang còn chậm, hệ thống chỉ gợi ý dựa trên một kĩ thuật,…

Trong tương lai, em sẽ tiếp tục tìm hiểu nhiều hơn để cải thiện những hạn chế cũng như phát triển thêm nhiều tính năng mới. Qua đó hy vọng có thể được áp dụng và phát triển những kĩ năng, kiến thức đã học, phát huy những điều ấy trong môi trường thực tế, phát triển hệ thống giúp ích cho cuộc sống.

Trong quá trình tìm hiểu và nghiên cứu, khó tránh những sai sót. Rất mong giảng viên và các bạn góp ý để đề tài được hoàn thiện hơn.

Em xin trân trọng cảm ơn!

*Sinh viên thực hiện!*

# TÀI LIỆU THAM KHẢO

[1] *Wenbo Shi*, [Recommendation Systems: A review](https://towardsdatascience.com/recommendation-systems-a-review-d4592b6caf4b#:~:text=A%20recommender%20system%2C%20or%20a,primarily%20used%20in%20commercial%20applications.) , Towards Data Science

[2] *Tiep Huu Vu*, [Content-based Recommendation Systems](https://machinelearningcoban.com/2017/05/17/contentbasedrecommendersys/), Machine Learning Co Ban

[3] *Tiep Huu Vu*, [Phân nhóm các thuật toán Machine Learning](https://khanhnv2901.github.io/2021-05-06/ML-categories), Machine Learning Co Ban

[4] *Abhinav Ajitsaria*, [Build a Recommendation Engine With Collaborative Filtering](https://realpython.com/build-recommendation-engine-collaborative-filtering/#:~:text=Collaborative%20filtering%20is%20a%20technique,similar%20to%20a%20particular%20user.), Real Python

[5] *Tommy*, [How Machine Learning Recommends Movies for You](https://towardsdatascience.com/how-machine-learning-recommends-movies-for-you-6f418f26bcb2), Medium

[6] *Rounak Banik*, [Movie Recommender Systems](https://www.kaggle.com/code/rounakbanik/movie-recommender-systems/notebook), Kaggle

[7] *Ibtesam Ahmed*, [Getting Started with a Movie Recommendation System](https://www.kaggle.com/code/ibtesama/getting-started-with-a-movie-recommendation-system/data), Kaggle

[8] *Bruno Stecanelle*, [Understanding TF-ID: A Simple Introduction](https://monkeylearn.com/blog/what-is-tf-idf/), Monkey Learn

[9] *Google*, [Recommendation Systems – Google Developers Course](https://developers.google.com/machine-learning/recommendation), Google Developers

[10] *Hai Ha*, [Matrix Factorization – Collaborative Filtering](https://viblo.asia/p/matrix-factorization-phuong-phap-goi-y-dua-tren-ky-thuat-phan-ra-ma-tran-p1-Az45bapNlxY), Viblo Asia