ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN - CƠ - TIN HỌC

Phạm Như Khoa - 20002062
Nguyễn Đình Kiên - 20002063

HỆ THỐNG ĐỀ XUẤT PHIM

Seminar một số vấn đề
chọn lọc về Khoa học dữ liệu

Ngành: Khoa học dữ liệu

(Chương trình đào tạo chuẩn)

Hà Nội - 2023

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA TOÁN - CƠ - TIN HỌC

Phạm Như Khoa - 20002062
Nguyễn Đình Kiên - 20002063

HỆ THỐNG ĐỀ XUẤT PHIM

Seminar một số vấn đề
chọn lọc về Khoa học dữ liệu

Ngành: Khoa học dữ liệu

(Chương trình đào tạo chuẩn)

Người hướng dẫn:
TS.Nguyễn Thị Bích Thuy

Hà Nội - 2023

Lời cảm ơn

Chúng em xin chân thành cảm ơn ***TS.Nguyễn Thị Bích Thủy*** đã tạo điều kiện và giúp đỡ chúng tôi hoàn thành bài tiểu luận với đề tài ***"Hệ thống đề xuất phim"***.

Mặc dù đã có nhiều cố gắng trong quá trình nghiên cứu, song do khả năng và kinh nghiệm của bản thân có hạn, nên bài tiểu luận của chúng em không thể tránh khỏi những tồn tại, hạn chế và thiếu sót. Vì vậy, rất mong nhận được sự góp ý chân thành của cô nhằm bổ sung hoàn thiện trong quá trình nghiên cứu tiếp theo.

Xin chân thành cảm ơn!

Hà Nội, ngày 21 tháng 04 năm 2023

Tóm tắt nội dung

Trong bài báo cáo này, chúng em sẽ xây dựng một hệ thống đề xuất phim sử dụng hai thuật toán là **Content-Based Filtering** và **Collaborative Filtering**. Sau khi thiết lập mô hình của hệ thống đề xuất, chúng em sẽ sử dựng Streamlit để thực hiện đề xuất phim bằng giao diện.

Mục lục

Danh sách hình vẽ

Chương 1

Tổng quan

1.1 Tổng quan

Với sự phát triển mạnh mẽ của khoa học kỹ thuật và Internet dẫn đến sự tăng trưởng bùng nổ của thông tin có sẵn trong những năm gần đây. Các hệ thống đề xuất (Recommendation systems) [2] là một trong những ứng dụng lọc thông tin hiệu quả và dễ dàng nhất. Mục đích của các hệ thống đề xuất này là tự động tạo ra các mục gợi ý như: phim, sách, tin tức, nhạc, ... cho người dùng tùy theo sở thích, lịch sử sử dụng của họ và tiết kiệm thời gian tìm kiếm trực tiếp bằng các thông thường.

Hệ thống đề xuất phim [2] là một trong những hệ thống được sự dụng rộng rãi nhất trên các nền đa phương tiện trực tuyến nhằm giúp khách hàng có thể truy cập các bộ phim ưa thích một cách thông minh từ kho phim khổng lồ. Ngày nay rất nhiều nghiên cứu đã được thực hiện để phát triển thuật toán và mở rộng việc đề xuất phim. Đã có nhiều nền tảng đa phương tiện trực tuyến nổi tiếng sử dụng hệ thống đề xuất tương tực như là: Youtube, Netflix, Douban, ...

Do sự quan tâm và nhu cầu của hệ thống đề xuất như trên, đề tài

"Hệ thống đề xuất phim" đã được chúng em lựa chọn. Mục tiêu chính của hệ thống đề xuất phim [4] là lọc và đưa ra dự đoán về phim mà người dùng có nhiều khả năng muốn xem nhất. Trong bài báo cáo này, chúng em sẽ nói về các bước triển khai để xây dựng một hệ thống đề xuất phim cơ bản sử dụng hai thuật toán chính là *Content-Based Filtering* và *Collaborative Filtering* [8] sau đó chúng em sẽ tạo ra một giao diện để có thể thực hiện việc đề xuất dễ dàng hơn cuối cùng là tổng kết đánh giá mô hình của hệ thống đề xuất và một số dự định cải tiến trong tương lai.

Chương 2

Tiếp cận vấn đề và hướng nghiên cứu

2.1 Tiếp cận vấn đề

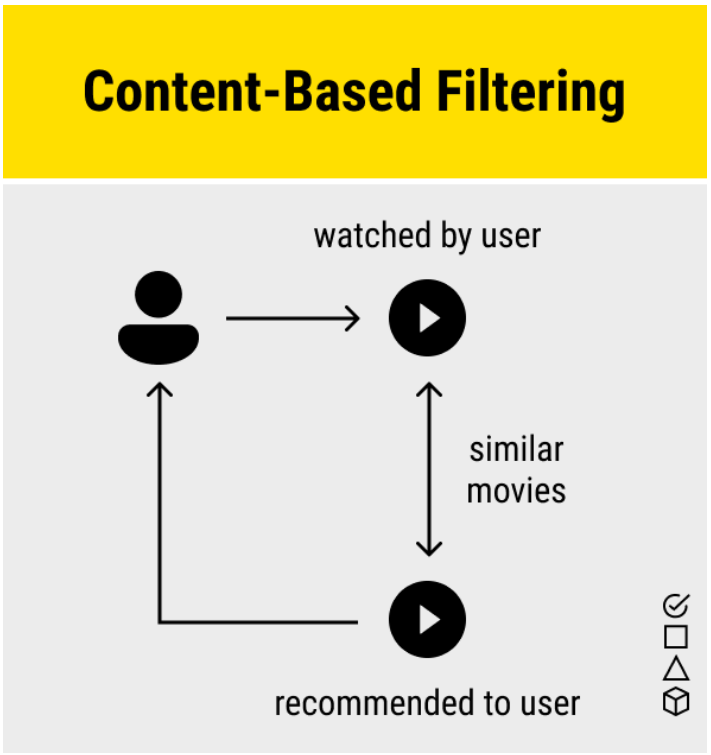
2.1.1 Thuật toán

Hệ thống đề xuất phim này chúng em sẽ triển khai bằng hai thuật toán là *Content-Based Filtering* [7] và *Collaborative Filtering* [5] .

Content-Based Filtering

Trong hệ thống đề xuất này, nội dung của phim (tổng quan, dàn diễn viên, đoàn làm phim, từ khóa, dòng giới thiệu, ...) được sử dụng để tìm điểm tương đồng của nó với các phim khác. Sau đó, những bộ phim có nhiều khả năng giống nhau nhất được đề xuất.

Chúng em sẽ tính điểm tương đồng theo cặp cho tất cả các phim dựa trên mô tả cốt truyện của chúng và đề xuất phim dựa trên điểm tương đồng đó. Đầu tiên chúng em sẽ chuyển nội dung phim từ văn bản sang



Hình 2.1: Content-Based Filtering

thành dạng vector từ bằng cách tính chỉ số *Term Frequency - Inverse Document Frequency* (TF-IDF) [1] .

Trong đó *Term Frequency* (TF) của một từ trong tài liệu và được đưa ra dưới dạng *(số trường hợp thuật ngữ/tổng số trường hợp)*. *Inverse Document Frequency* (IDF) là số lượng tương đối của các tài liệu chứa thuật ngữ được đưa ra dưới dạng: *log(số lượng tài liệu/tài liệu có thuật ngữ)*. Tầm quan trọng tổng thể của mỗi từ đối với các tài liệu mà chúng xuất hiện bằng $TF * IDF$.

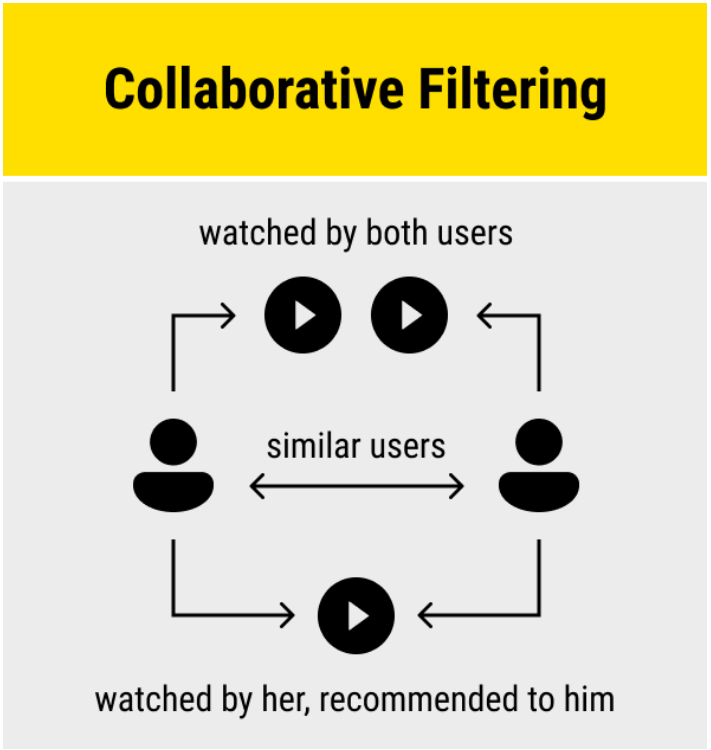
Điều này sẽ cung cấp cho bạn một ma trận trong đó mỗi cột đại diện cho một từ trong từ vựng tổng quan (tất cả các từ xuất hiện trong ít nhất một tài liệu) và mỗi hàng đại diện cho một bộ phim, như trước đây. Điều này được thực hiện để giảm tầm quan trọng của các từ xuất hiện thường xuyên trong tổng quan về cốt truyện và do đó, tầm quan trọng của chúng trong việc tính toán điểm tương đồng cuối cùng.

Sau đó chúng em sẽ sử dụng công thức *cosine* để tính ra chỉ số biểu thị độ giống nhau giữa các phim.

$$cos(\eta) = \frac{A.B}{||A||.||B||} = \frac{\sum_{i=1}^{n} A_i.B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2}.\sum_{i=1}^{n}(B_i)^2}$$

Collaborative Filtering

Với thuật toán *Content-Based Filtering* thì nó chỉ gợi ý những bộ phim gần với một bộ phim nhất định. Nghĩa là, nó không có khả năng nắm bắt thị hiếu và đưa ra đề xuất giữa các thể loại. Ngoài ra, công cụ mà chúng tôi xây dựng không thực sự mang tính cá nhân ở chỗ nó không nắm bắt được thị hiếu và thành kiến cá nhân của người dùng. Bất kỳ ai truy vấn công cụ của chúng tôi để tìm các đề xuất dựa trên một bộ phim sẽ nhận được các đề xuất tương tự cho bộ phim đó, bất kể người dùng là ai.



Hình 2.2: Collaborative Filtering

Vì vậy chúng em sử dụng thêm thuật toán *Collaborative Filtering* để đưa ra đề xuất cho người dùng. Thuật toán này bao gồm 2 loại chính đó là:

User based filtering [5]

Hệ thống này giới thiệu sản phẩm cho người dùng mà những người dùng tương tự đã thích. Để đo độ tương tự giữa hai người dùng, cụ thể sử dụng độ giống nhau cosine.

Item Based Collaborative Filtering [5]

Hệ thống sẽ dựa trên mặt hàng đề xuất các mặt hàng dựa trên sự tương đồng của chúng với các mặt hàng mà người dùng mục tiêu đã xếp hạng.

2.1.2 Công cụ

Đề xây dựng hệ thống đề xuất phim này chúng em sử dụng ngôn ngữ lập trình Python [6]và Streamlit [3] để xây dựng giao diện.

Python

Hình 2.3: Nhiệm vụ có kích thước lớn nhất

Python là một ngôn ngữ lập trình thông dịch, đa nền tảng và đa mục đích. Nó được phát triển vào năm 1980 bởi Guido van Rossum và hiện tại là một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới.

Python có cú pháp đơn giản và dễ đọc, giúp cho việc phát triển ứng dụng nhanh chóng và dễ dàng. Nó cũng cung cấp nhiều thư viện và framework hữu ích cho các lĩnh vực khác nhau như khoa học dữ liệu, trí tuệ nhân tạo, web development và nhiều hơn nữa. Python cũng được sử dụng trong nhiều lĩnh vực khác nhau như game development, đồ họa máy tính và hệ thống nhúng. Với sự phát triển của cộng đồng, Python đang ngày càng trở thành một ngôn ngữ lập trình tất cả trong mọi lĩnh vực, được sử dụng bởi các công ty lớn như Google, Facebook, Netflix và Dropbox.

Streamlit

Streamlit là một framework mã nguồn mở miễn phí được sử dụng để xây dựng các ứng dụng web dữ liệu và trực quan với tốc độ nhanh và độ linh hoạt cao. Streamlit được phát triển trên Python và cung cấp cho người dùng một cách dễ dàng để tạo ra các ứng dụng web bằng cách sử dụng các thư viện và các công cụ mà họ đã quen thuộc trong việc phân tích dữ liệu và trực quan hóa dữ liệu.

Hình 2.4: Nhiệm vụ có kích thước lớn nhất

Với Streamlit, người dùng có thể tạo các ứng dụng web trực quan một cách dễ dàng và nhanh chóng, bao gồm các biểu đồ, bảng, đồ thị và nhiều hơn nữa mà không cần phải có kiến thức chuyên sâu về web development. Streamlit cũng cung cấp cho người dùng khả năng tương tác với dữ liệu trực tiếp trong ứng dụng của mình, giúp cho người dùng

có thể thay đổi dữ liệu đầu vào và xem kết quả ngay lập tức.

Nhờ tính năng dễ sử dụng và linh hoạt, Streamlit đã trở thành một trong những công cụ được yêu thích nhất cho các nhà phân tích dữ liệu và nhà khoa học dữ liệu khi muốn tạo ra các ứng dụng web trực quan và đẹp mắt.

2.2 Phương pháp nghiên cứu

2.2.1 Dữ liệu

Dữ liệu dùng để xây dựng hệ thống sử dụng thuật toán *Content-Based Filtering* là bộ dữ liệu **TMDB 5000 Movie Dataset** trên trang Kaggle. Và chúng em sử dụng bộ dữ liệu **The Movies Dataset** trên Kaggle trong thuật toán *Collaborative Filtering*.

TMDB 5000 Movie Dataset

Bộ dữ liệu này gồm 2 tệp là *tmdb_5000_credits.csv* và *tmdb_5000_movies.csv*.

tmdb_5000_credits.csv

Tệp này bao gồm 4 trường và 4803 dòng. Chi tiết các trường dữ liệu như sau:

- movie_id - ID của phim.
- title - Tiêu đề phim
- cast - Tên của diễn viên.
- crew - Tên của đạo diễn, biên kịch, ...

tmdb_5000_movies.csv

Tệp này bao gồm 20 trường và 4803 dòng. Chi tiết các trường dữ liệu như sau:

- budget - Ngân sách của bộ phim.
- genre - Thể loại phim.
- homepage - Trang chủ của bộ phim.
- id - Giống movie_id ở tệp trên.
- keywords - Từ khóa liên quan đến bộ phim.
- original_language - Ngôn ngữ của phim.
- original_title - Tên phim gốc trước khi dịch và chuyển thể
- overview - Mô tả ngắn gọn về phim.
- popularity - Mức độ phổ biến của phim.
- production_companies - Nhà sản xuất phim.
- production_countries - Quốc gia sản xuất phim.
- release_date - Ngày sản xuất.
- revenue - Doanh thu toàn cầu.
- runtime - Thời lượng phim.
- status - Trạng thái "Đã phát hành" hay là "Được đồn đại".
- tagline - Khẩu hiểu của phim.
- title - Tiêu đề phim.
- vote_average - Số lượt vote trung bình mà bộ phim nhận được.
- vote_count - Số vote nhận được.

Movie Lens Small Latest Dataset

Tập dữ liệu này (ml-mới nhất-nhỏ) mô tả xếp hạng 5 sao. Nó chứa 100836 đánh giá và 3683 nhãn trên 9742 phim. Những dữ liệu này được tạo bởi 610 người dùng trong khoảng thời gian từ ngày 29 tháng 3 năm 1996 đến ngày 24 tháng 9 năm 2018. Bộ dữ liệu này được tạo vào ngày 26 tháng 9 năm 2018. Chi tiết các cột trong các tệp chúng em sử dụng là:

tag.csv

Chứa thông tin về nhãn cho người dùng gán.

- userId - ID người dùng.
- movieId - ID phim.
- tag - Nhãn phim.
- timestamp - Mốc thời gian.

movie.csv

Chứa thông tin về phim.

- movieId - ID phim.
- title - Tiêu đề phim.
- genres - Thể loại phim.

rating.csv

Chứa thông tin về đánh giá của người dùng.

- userId - ID người dùng.

- movieId - ID phim.
- ratings - Đánh giá điểm.
- timestamp - Mốc thời gian.

link.csv

Chứa số nhận dạng sử dụng để liên kết với một số nguồn khác cụ thể là IMDB và TMBD.

- movieId - ID phim.
- imdbId - IMDB ID.
- tmbdId - TMBD ID.

2.2.2 Tiền xử lý dữ liệu

Xử lý với bộ dữ liệu TMDB 5000 Movie Dataset

Sau khi loại bỏ các các hàng trùng lặp, tệp movie chúng em sẽ lấy ra các trường quan trọng là **genres, movie_id, keywords, title, overview, cast, crew** Sau đó thực hiện các bước sau:

- Kiểm tra các giá trị null và loại bỏ.
- Xử lý các trường dữ liệu có các bản ghi ở dạng JSON là cột *genres, keywords, cast, crew*. Chuyển dạng JSON sang dạng Dictionary.
- Xử lý phần nội dung ở trường *overview* bằng cách xóa các khoảng trắng, giảm sự trùng lặp dữ liệu.
- Tạo cột *tag* gộp các cột *overview, genres, keywords, cast, crew*

2.2.3 Triển khai

Content-Based Filtering

Sử dụng *CountVectorizer* để tạo ra một ma trận với mỗi hàng là một bộ phim, mỗi cột là một từ xuất hiện trong các tag của các bộ phim. Số cột được giới hạn bằng *max_features*, và ta sẽ loại bỏ các *stop_words* trong tiếng Anh thông qua tham số *stop_words*.

Sử dụng *cosine_similarity* của *scikit-learn* để tính toán ma trận độ tương đồng cosine giữa các bộ phim dựa trên ma trận vector từ *CountVectorizer*.

Để đưa ra các đề xuất bộ phim, chúng em sẽ chọn một bộ phim đầu vào và tìm các bộ phim có độ tương đồng cosine cao nhất với nó. Sau đó sắp xếp các bộ phim đó theo thứ tự giảm dần và trả về n bộ phim đầu tiên.

Công thức tính toán *cosine similarity* giữa hai vector x và y được cho bởi công thức:

$$cosinesimilarity(x, y) = \frac{x.y}{||x||.||y||}$$

Trong đó, x và y là hai vector cần so sánh, là phép nhân vector, và |||| là độ dài Euclid của vector đó. Kết quả của cosine similarity là một số thực trong khoảng [-1,1], cho biết mức độ tương đồng giữa hai vector. Khi kết quả gần với 1, hai vector tương đồng cao và gần với -1 thì hai vector tương đối khác biệt.

Collaborative Filtering

User-based (user-user) collaborative filtering

Chia tập đánh giá ratings thành tập train và test tỉ lệ 7:3. Tập train gồm 70585 dòng và 4 cột. Tập test gồm 30251 dòng và 4 cột.

Chúng em sẽ thực hiện đánh giá và dự đoán. Tập train để dự đoán những bộ phim chưa được user rate. Để bỏ qua những bộ phim được user rate, chúng em sẽ đánh dấu nó là 0 trong quá trình dự đoán. Những bộ phim không được user rate được đánh dấu là 1 để dự đoán. Tập test sẽ được sử dụng để đánh giá. Để đánh giá, chúng em sẽ chỉ đưa ra dự đoán về những bộ phim được user đã rate. Vì vậy, nó được đánh dấu là 1, ngược lại với tập train.

Tiếp đó, chúng em sẽ tạo ma trận độ tương đồng bằng cách sử dụng độ tương đồng cosine giữa các user giống phần trên. Sau đó sẽ thực hiện đánh giá và dự đoán.

Item-based collaborative filtering

Thực hiện tạo ma trận độ tương đồng từ tập Train đã có từ phương pháp user - user. Sau đó thực hiện đánh giá và dự đoán.

Chương 3

Kết quả và kết luận

3.1 Kết Quả

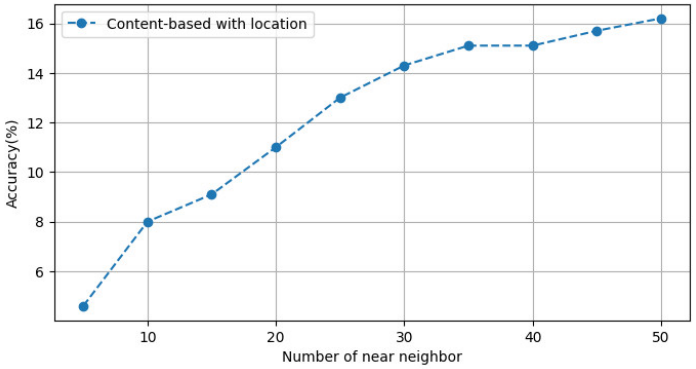
Đề xuất

Kết quả thu được khi đề xuất phim bằng phương pháp **Content-Based Filtering**.

Hình 3.1: Đề xuất phim bằng phương pháp Content-Based Filtering

Với đầu vào là bộ phim *"Batman Begins"* là 1 bộ phim về siêu anh hùng, hệ thống đã đề xuất ra cho ta các phần khác nhau bộ phim Batman. Khi đầu vào là bộ phim hoạt hình *"Kung Fu Panda 2"*, kết quả nhận được ngoài các phần phim còn lại của Kung Fu Panda, hệ thống còn đề xuất thêm phim *"Legend of a Rabbit"* một bộ phim hoạt hình khác cũng về nội dung võ thuật, hay phim *"Kickboxer: Vengeance"* nội dung liên

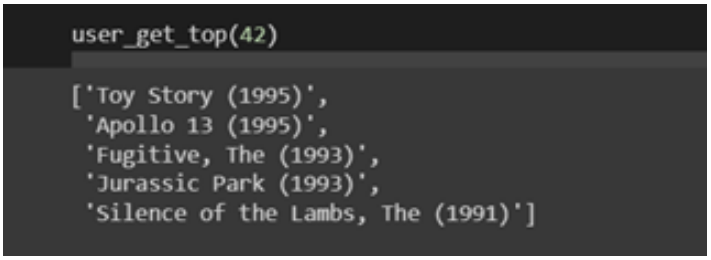
quan tới bộ môn boxing... Để làm rõ hơn về kết quả của phương pháp *content based filtering* chúng em giới hạn số lượng lân cận tương đồng từ 5-50 với các bước nhảy là 5 để tìm ra số lân cận tốt nhất.



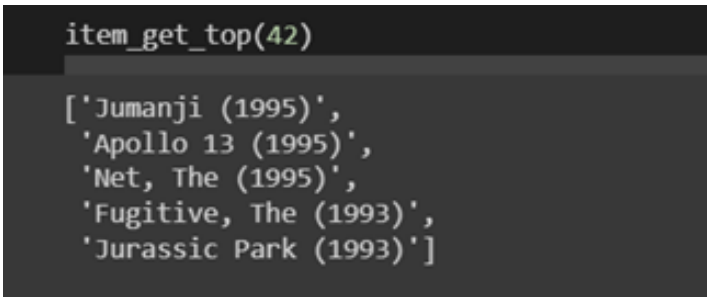
Hình 3.2: Độ tin cậy theo số lượng phim đề xuất bằng phương pháp Content-Based Filtering

Hình 3.2 trình bày kết quả accuracy theo số lượng bộ phim được đề xuất của content-based filtering, với số lượng 25 lân cận thì độ chính xác bắt đầu tăng chậm. Vì vậy, chúng em nhận thấy với hệ thống này, 25 là số lân cận tốt nhất dựa vào các tiêu chí đã đặt ra (giảm chi phí tính toán và giảm thời gian người dùng tìm kiếm phim nhưng vẫn đảm bảo kết quả đề xuất) với độ chính xác là 11.02%.

Kết quả thu được khi đề xuất phim bằng phương pháp **Collaborative Filtering**.



Hình 3.3: Đề xuất phim cho user 42 bằng User - User



Hình 3.4: Đề xuất phim cho user 42 bằng Item - Item

Bảng dưới đây thể hiện kết quả so sánh trong việc đánh giá độ chính xác của hai phương pháp User - User và Item-Item. Dựa trên kết quả RMSE và MAE của phương pháp user-user và Item - Item, có thể nhận xét rằng phương pháp User-User cho kết quả tốt hơn so với phương pháp Item - Item.

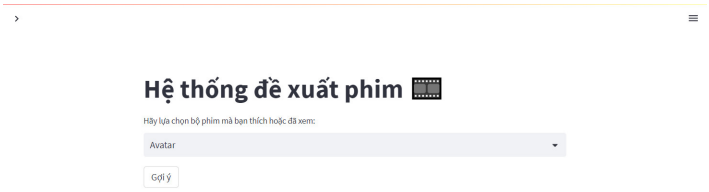
Mô hình	RMSE	MAE
User - User	1.564	1.212
Item - Item	2.512	2.215

Cụ thể, phương pháp user-user có giá trị RMSE là 1.564 và MAE là 1.212, trong khi đó phương pháp Item-Item có giá trị RMSE là 2.512 và MAE là 2.215. Điều này cho thấy rằng phương pháp User - User đã đưa

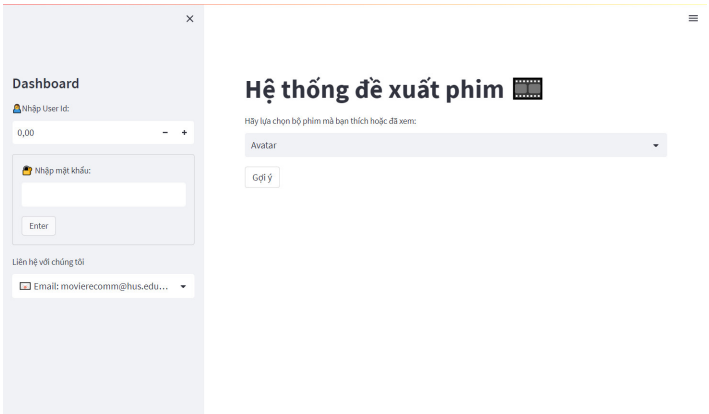
ra các recommendation về đánh giá của người dùng gần hơn với đánh giá của người dùng theo phương pháp Item - Item.

Triển khai giao diện

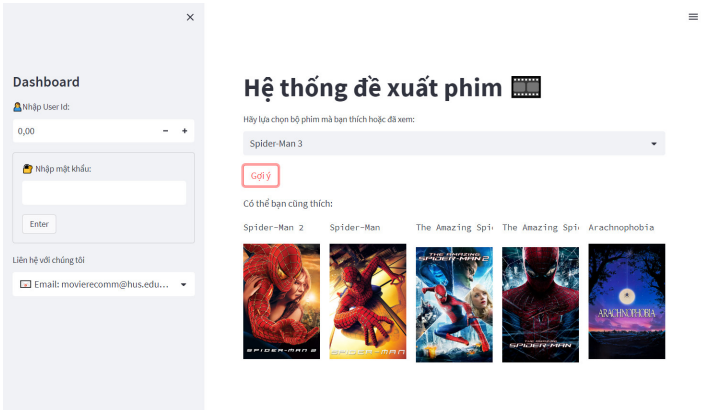
Sau đây là một số kết quả chúng em thực hiện tạo giao diện gợi ý đề xuất phim cho người dùng thông qua Streamlit.



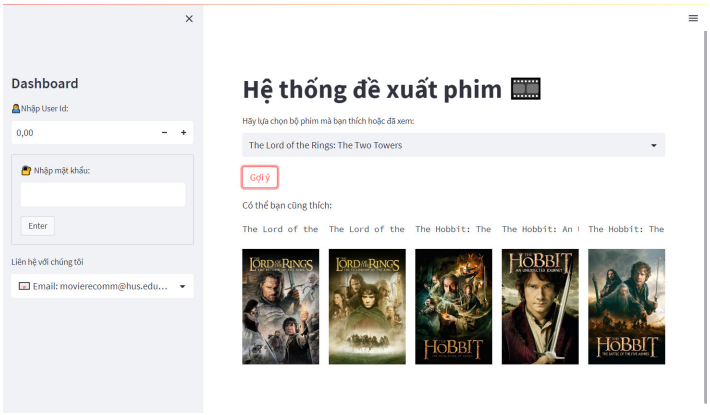
Hình 3.5: Thanh tìm kiếm phim và đưa ra đề xuất



Hình 3.6: Bên trái là phần đăng nhập theo ID của user

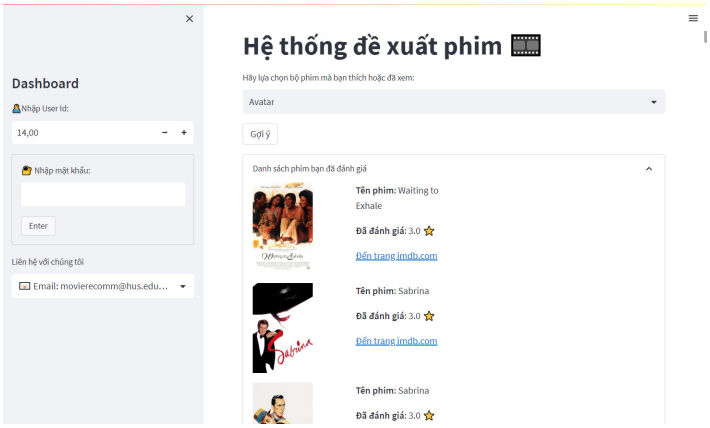


Hình 3.7: Đề xuất khi tìm kiếm theo phim "Spider-Man 3"

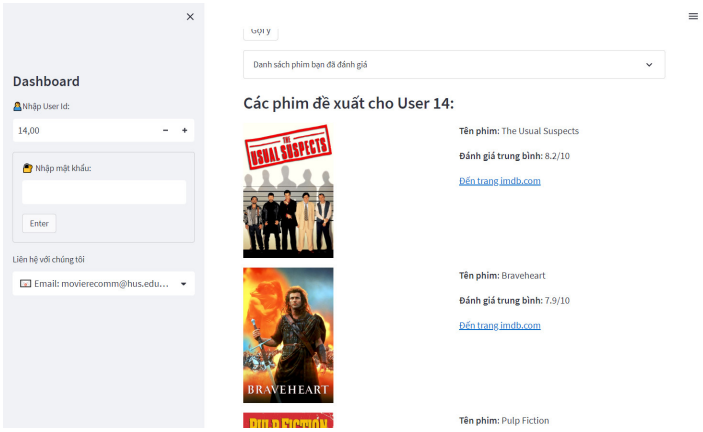


Hình 3.8: Đề xuất khi tìm kiếm theo phim "The Lord of the Rings: The Two Towers"

Sau khi đăng nhập theo User ID 14 chúng em hệ thống sẽ đưa ra một số đề xuất theo cá nhân User ID 14.



Hình 3.9: Xem các phim đã dánh giá của người dùng có user ID 14



Hình 3.10: Đề xuất phim cho người dùng user ID 14

3.2 Thảo luận

Hệ thống đề xuất đã gợi ý theo cả hai phương pháp đều đưa ra khá tốt thể loại và nội dung phim khá tương đồng với nội dung người dùng muốn tìm kiếm và theo sở thích. Thế nhưng bên cạnh đó còn một số hạn chế khi số lượng đề xuất tăng lên.

Chương 4

Kết luận

4.1 Kết luận

Chúng em đã xây dựng được một hệ thống đề xuất phim sử dụng hai thuật toán **Content-Based Filtering** và **Collaborative Filtering**. Hệ thống giao diện có thể thực hiện gợi ý cho người dùng và tìm kiếm.

Sau khi thực hiện project, chúng em rút ra một vài ưu nhược điểm của hai thuật toán trên đã thực hiện:

Đối với thuật toán Content-based Filtering:

- Ưu điểm: Không yêu cầu nhiều dữ liệu người dùng, có thể dễ dàng triển khai và hiệu quả đối với các sản phẩm ít được đánh giá.
- Nhược điểm: Giới hạn bởi tính chất của sản phẩm, chỉ có thể đề xuất các sản phẩm tương tự về nội dung với các sản phẩm đã được đánh giá, không phù hợp cho việc đề xuất sản phẩm đa dạng và phức tạp.

Đối với thuật toán Collaborative Filtering:

- Ưu điểm: Cho kết quả chính xác, phù hợp với sở thích cá nhân của

người dùng, có khả năng đề xuất sản phẩm mới và khác biệt so với các sản phẩm đã được đánh giá. Ngoài ra phù hợp với việc đề xuất các sản phẩm đa dạng và phức tạp

- Nhược điểm: Yêu cầu lượng dữ liệu lớn và đa dạng để có kết quả tốt, khó xử lý với những sản phẩm mới chưa được đánh giá. Dễ bị ảnh hưởng bởi spam, các đánh giá sai hoặc các người dùng có sở thích đặc biệt.

4.2 Đề xuất

Hướng phát triển trong tương lai:

Bộ dữ liệu:

Thu thập thêm dữ liệu từ các trang web xem phim trực tuyến, cùng với đó là thu thập thêm các thuộc tính mới như: bình luận của người đánh giá, rating cho từng khía cạnh,. . . để cho ra bộ dữ liệu đầy đủ thông tin hơn. Ngoài ra, xử lý hiện tượng thưa thớt trong các rating matrix cũng là một vấn đề rất quan trọng.

Mô hình:

Áp dụng các phương pháp, kỹ thuật đề xuất khác như: Collaborative Filtering dùng model-based, Knowledge-Based Recommender Systems, Demographic Recommender Systems, Hybrid and Ensemble-Based Recommender Systems . . . để cải thiện kết quả dự đoán tốt hơn nữa.

Tài liệu tham khảo

[1] S. Chawla, S. Gupta, and R. Majumdar. Movie recommendation models using machine learning. In *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*, pages 1–6, Mathura, India, 2021. IEEE.

[2] Jose Immanuvel, A Sheelavathi, M Priyadharshan, S Vignesh, and K Elango. Movie recommendation system. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, 10(6):2611, June 2022.

[3] Streamlit Inc. Streamlit: The fastest way to build data apps, 2021.

[4] Sambandam Jayalakshmi, Narayanan Ganesh, Robert Čep, and Janakiraman Senthil Murugan. Movie recommender systems: Concepts, methods, challenges, and future directions. *Sensors*, 22(13):4904, 2022.

[5] R. C K and K. C. Srikantaiah. Similarity based collaborative filtering model for movie recommendation systems. In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 1143–1147, Madurai, India, 2021. IEEE.

[6] Python Software Foundation. *Python Documentation*. Python Software Foundation, 3.10 edition, 10 2021.

[7] Poonam Sharma and Lokesh Yadav Yadav. Movie recommendation system using item based collaborative filtering. *International Journal of Innovative Research in Computer Science & Technology (IJIRCST)*, 8(4), July 2020.

[8] N. Soni, K. Kumar, A. Sharma, S. Kukreja, and A. Yadav. Machine learning based movie recommendation system. In *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, pages 1–4, Dehradun, India, 2021. IEEE.