TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

---o0o---

BÁO CÁO MÔN HỌC

Movie Recommendation System
(Hệ thống đề xuất phim)

Môn học : **Phân tích dữ liệu**

Giáo viên hướng dẫn : **Th Hồ Hướng Thiên**

Sinh viên thực hiện: : **Bùi Dạ Lý - 2254052042**

: **Huỳnh Lệ Giang - 2254050009**

: **Võ Thị Ngọc Chi - 2254052008**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024

MỤC LỤC

Lời Mở Đầu 2

1. Lý do chọn đề tài 2

2. Mục tiêu nghiên cứu 3

Chương 1: Giới thiệu về Dataset 3

Chương 2: Tiền xử lý dữ liệu 3

Chương 3: Trực quan hóa dữ liệu 4

Chương 4. Xử lý outlier và Chuyển đổi dữ liệu 6

Chương 5: Các Mô Hình Thực Hiện 6

5.1 Content-based Filtering 6

5.2 User-user Collaborative Filtering 10

5.3 Item-item Collaborative Filtering 12

Chương 6: Đánh giá mô hình 14

Chương 7: Ưu Nhược Điểm Của Các Mô Hình 15

Kết Luận 16

Tài Liệu Tham Khảo 16

Lời Mở Đầu

1. Lý do chọn đề tài

Trong thời đại kỹ thuật số, ngành công nghiệp giải trí đang phát triển mạnh mẽ, đặc biệt là các nền tảng phát trực tuyến (streaming) như Netflix, Amazon Prime, hay Disney+.

Với hàng triệu bộ phim và chương trình truyền hình, người dùng dễ dàng bị choáng ngợp khi lựa chọn nội dung phù hợp với sở thích cá nhân.

Hệ thống đề xuất (Recommendation System) là giải pháp thiết yếu, giúp cá nhân hóa trải nghiệm người dùng, tăng mức độ hài lòng và giữ chân khách hàng. Ngoài ra, đây cũng là một ứng dụng quan trọng trong lĩnh vực khoa học dữ liệu, học máy và trí tuệ nhân tạo, mở ra cơ hội nghiên cứu và phát triển các thuật toán xử lý dữ liệu lớn.

2. Mục tiêu nghiên cứu

Nhằm áp dụng kiến thức lý thuyết vào thực tiễn, đồng thời tìm hiểu cách xử lý và phân tích dữ liệu trong bài toán thực tế, chúng tôi chọn đề tài "Hệ thống đề xuất phim" để nghiên cứu. Đây là một lĩnh vực có ứng dụng cao trong thực tế, giúp phát triển tư duy phân tích dữ liệu và giải quyết vấn đề.

Mục tiêu của hệ thống đề xuất phim là dự đoán đánh giá của người dùng đối với các bộ phim sau đó đề xuất các phim có đánh giá cao dựa trên sự tương đồng của người dùng hoặc các bộ phim (User-based/ Item-based Collaborative Filtering) và dựa trên đặc tính các bộ phim và lịch sử xem của người dùng (Content-based Filtering).

Chương 1: Giới thiệu về Dataset

Hệ thống sử dụng dữ liệu MovieLens (ml-latest-small) là một tập dữ liệu ghi lại hoạt động đánh giá phim theo thang điểm 5 sao và gắn thẻ tự do của người dùng từ dịch vụ MovieLens, một hệ thống gợi ý phim trực tuyến tại MovieLens. Với 2 tập dữ liệu là movies.cvs và ratings.csv, có **610 người dùng** đã đóng góp đánh giá từ ngày **29 tháng 3 năm 1996** đến **24 tháng 9 năm 2018**.

- Trong movies.csv bao gồm thông tin của 9743 bộ phim chứa thông tin về mã phim, tên phim, và các thể loại trong phim.
- Trong ratings.csv bao gồm 100837 lượt đánh giá của 610 người dùng cho các mã phim với các rating từ 1 đến 5 cùng với thời gian đánh giá.

Chương 2: Tiền xử lý dữ liệu

- Xử lý dữ liệu file movies:
    - Kiểm tra giá trị null, na

- Kiểm tra trùng lắp
- Xử lý cột genres và title thành chữ thường
- Điền giá trị thiếu cho cột year bằng ‘unknown’

- Xử lý dữ liệu file ratings:
  - Kiểm tra giá trị null, na
  - Kiểm tra trùng lắp
  - Bỏ cột timestamp không cần thiết
- Merging dữ liệu từ 2 file movies và rating lại thành tập dữ liệu cuối cùng df dùng để xây dựng mô hình

Chương 3: Trực quan hóa dữ liệu

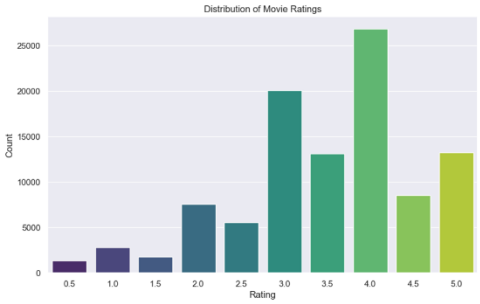
1. **Mục đích trực quan hóa:**

✓ Là để biến các tập dữ liệu phức tạp trở nên dễ hiểu hơn thông qua các hình ảnh trực quan như biểu đồ, đồ thị và cấc biểu đồ khác.

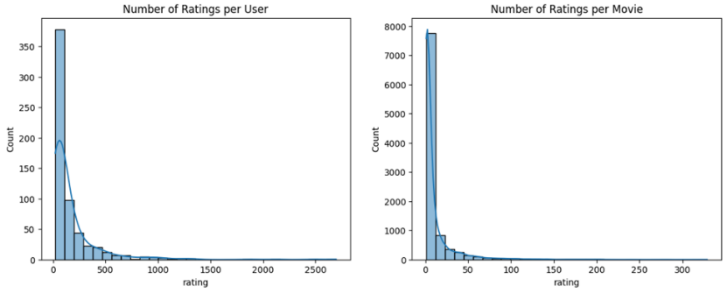
✓ Giúp cho ta có thể dễ dàng nhận ra được mối quan hệ, xu hướng và biến động trong dữ liệu.

✓ Từ đó phát hiện ra các vấn đề và đưa ra các insight ban đầu để định hướng các bước tiếp theo.

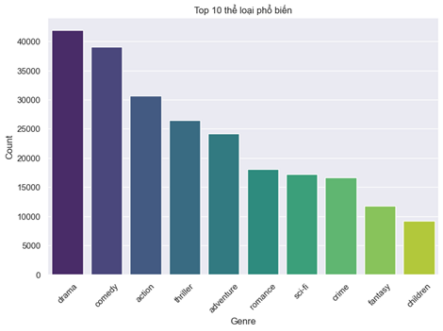
2. **Tổng quan dữ liệu:**
   - Biểu đồ thể hiện tần suất rating.

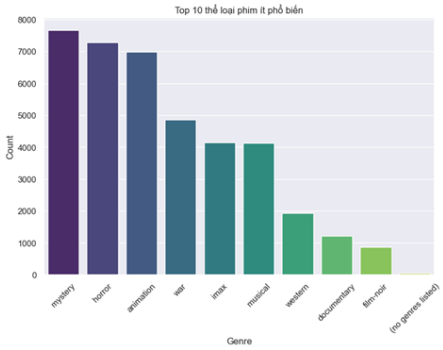


   - Biểu đố phân tích số lượng rating theo user và item.

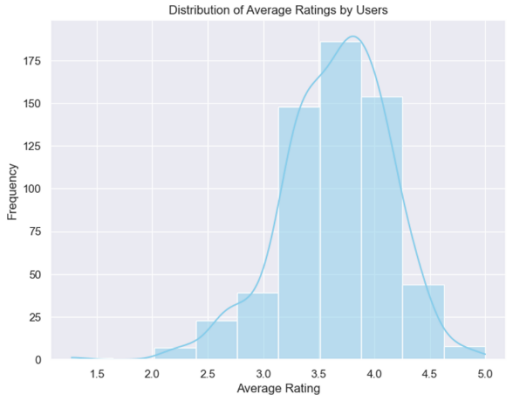


   - Biểu đồ thể hiện phân phối thể loại phim.





- Biểu đồ đánh giá mối quan hệ giữa mức độ phổ biến của phim và xếp hạng trung bình.



3. Insight từ trực quan hóa:

- Người dùng có xu hướng đánh giá cao hơn mức trung bình. Đa phần các đánh giá nằm trong khoảng từ **3.0 đến 4.5.**
- Phần lớn người dùng chỉ thực hiện rất ít đánh giá (cụ thể, dưới 100 đánh giá).
- Phần lớn các bộ phim nhận được rất ít đánh giá (dưới 50 đánh giá).

➔Ảnh hưởng:

- Mô hình có thể dễ bị thiên vị, dự đoán điểm đánh giá cao cho hầu hết các phim.
- Có thể có lỗi trong việc ghi nhận dữ liệu với các mức thấp này.

➔Cải thiện:

- Nên chuẩn hóa dữ liệu trước khi xây dựng mô hình

- Có thể xem xét sử dụng các phương pháp để cân bằng trọng số giữa các điểm đánh giá thấp và cao.

Chương 4. Xử lý outlier và Chuyển đổi dữ liệu

- Từ việc thống kê, trực quan hóa dữ liệu, ta nhận thấy sự thiên lệch của dữ liệu (Phân phối điểm đánh giá phim thiên về tích cực, trong khi số lượng đánh giá của người dùng và phim không đồng đều), để giải quyết vấn đề này chúng tôi quyết định áp dụng một số phương pháp sau để giảm tác động của các thiên lệch đó, gồm:

- Dùng Bayesian Average cho movieId để làm giảm ảnh hưởng của phim có quá ít hoặc quá nhiều đánh giá.
- Dùng tỷ lệ nghịch để giảm trọng số của userId với số lượng đánh giá bất thường.
- Nhân kết quả của hai phương pháp trên với nhau để tạo ra cột giá trị weighted_rating

- Dữ liệu ban đầu userId và movieId bắt đầu từ 1, trong khi đó mô hình Collaborative Filtering, chỉ số người dùng và phim thường được sử dụng làm index trong các ma trận thưa (sparse matrix). Các index này cần liên tục và bắt đầu từ 0 để phù hợp với cách lưu trữ và truy xuất dữ liệu trong ma trận. Vì vậy sử dụng LabelEncoder để chuẩn hóa các giá trị trong cột userId và movieId thành các số nguyên liên tục bắt đầu từ 0.

Chương 5: Các Mô Hình Thực Hiện

5.1 Content-based Filtering

Content-Based Filtering được sử dụng trong bài này là xây dựng một hệ thống gợi ý, trong đó các gợi ý được tạo ra dựa trên các đặc trưng của các phim và lịch sử đánh giá của người dùng. Sau đó sử dụng các đặc trưng đó dùng mô hình Ridge Regression để dự đoán rating cho các phim.

Thuật toán Content-Based Filtering kết hợp với Ridge Regression có các bước sau:

Bước 1: Xây dựng tập huấn luyện và tập kiểm tra

- Chia dữ liệu thành tập huấn luyện và kiểm tra (60-40)
  - Lấy danh sách tất cả các thể loại
  - Đối với từng thể loại trong danh sách, xác định các phim chưa thể loại đó và chia vào 2 phần: 60% vào tập huấn luyện (train) và 40% vào tập kiểm tra (test)

- Đảm bảo mỗi thể loại có mặt trong cả tập huấn luyện và tập kiểm tra và không có sự trùng lắp giwuax các phim trong tập hhuaans luyện và tập kiểm tra. Dữ liệu được chia dựa trên thể loại để đảm bảo sự phân phối hợp lý. Tránh trường hợp trong tập huấn luyện có thể loại mà trong tập kiểm tra không có khiến mô hình học sai.

Bước 2: Xây dựng ma trận TFIDF (Cho cả tập Train và tập Test)

- Tạo danh sách thể loại dưới dạng chuỗi và sử dụng CountVectorizer để mã hóa cac thể loại sang 1 và 0. Với các cột là các thể loại được tách ra từ Genres, hàng là các bộ phim trong danh sách
- Sử dụng ma trận vừa tạo để chuyển đổi thành ma trận TF-IDF bằng TfidfTransformer
  - **TF** (Term Frequency) tính toán tần suất xuất hiện của một từ (hoặc thể loại) trong mỗi bộ phim.

$$\text{TF}(t, d) = \frac{\text{Số lần từ } t \text{ xuất hiện trong tài liệu } d}{\text{Tổng số từ trong tài liệu } d}$$

  - **IDF** (Inverse Document Frequency) tính toán mức độ quan trọng của một từ (hoặc thể loại) trong toàn bộ tập dữ liệu

$$\text{IDF}(t, D) = \log\left(\frac{\text{Số tài liệu trong tập dữ liệu } D}{\text{Số tài liệu chứa từ } t}\right)$$

  - **TF-IDF** bằng cách nhân TF và IDF

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D)$$

Mục đích là TF-IDF cho thể loại phim giúp đánh giá chính xác hơn sự quan trọng của từng thể loại trong bộ dữ liệu phim.

Bước 3: Xây dựng ma trận người dùng - thể loại (Tập Train)

Xây dựng từ dữ liệu huấn luyện, trong đó các cột là thể loại phim, các hàng là người dùng, mỗi giá trị trong ma trận là **trung bình đánh giá** của người dùng đối với từng thể loại phim

Bước 4: Áp dụng Ridge Regression cho từng người dùng: xây dựng một mô hình học máy cho mỗi người dùng nhằm dự đoán các đánh giá phim của họ, dựa trên các thể loại của các bộ phim mà họ đã xem và đánh giá trung bình của người dùng đó với thê loại phim. Mô hình này được huấn luyện bằng **Ridge Regression**, một loại hồi quy tuyến tính

với cơ chế regularization (điều chỉnh độ phức tạp của mô hình) giúp giảm thiểu overfitting.

- Xây dựng hàm huấn luyện cho từng người dùng:
  - Lấy vector thể loại của người dùng trong Ma trận người dùng - thể loại
  - Lấy các chỉ số của phim đã được người dùng đánh giá trong tập huấn luyện
  - Lấy các vector của các phim đó từ Ma trận TF-IDF
  - Kết hợp các vector phim và vector thể loại của người dùng tạo thành một ma trận đặc trưng (features matrix). Đây là dữ liệu đầu vào huấn luyện

    - Người dùng U_1 đã đánh giá 2 phim:
      - **Phim 1 TF-IDF**: [0.1, 0.5, 0.4]
      - **Phim 2 TF-IDF**: [0.3, 0.7, 0.2]
    - **Sở thích thể loại của người dùng**: [0.8, 0.2] (Action: 0.8, Drama: 0.2).

    Kết quả sau khi kết hợp để tạo nhãn cho mô hình:

$$\begin{bmatrix} 0.1 & 0.5 & 0.4 & 0.8 & 0.2 \\ 0.3 & 0.7 & 0.2 & 0.8 & 0.2 \end{bmatrix}$$

Mục đích: Việc thêm thông tin về **sở thích thể loại của người dùng** giúp mô hình hiểu rõ hơn về mối quan hệ giữa người dùng và phim. Giảm thiểu hụt dữ liệu.

  - Lấy các đánh giá của người dùng đối với các phim đã đánh giá. Đây là dữ liệu nhãn mà mô hình sẽ học để dự đoán
  - Huấn luyện mô hình hồi quy Ridge với tham số điều chỉnh alpha. Sau đó mô hình được huấn luyện với dữ liệu đặc trưng và nhãn
  - Quá trình học

$$J(w) = \sum_{i=1}^{n} (y_i - X_i w)^2 + \alpha \sum_{j=1}^{p} w_j^2$$

Trong đó:

- X: Ma trận đặc trưng (feature matrix).
- y: Vector mục tiêu (ratings).
- w: Trọng số cần tìm.
- α: Hệ số điều chuẩn. Giá trị lớn của α làm giảm overfitting nhưng có thể dẫn đến underfitting.
- n: Số lượng mẫu (movies đã đánh giá bởi user).
- p: Số lượng đặc trưng (bao gồm các đặc trưng phim và vector thể loại của người dùng).

Mô hình sẽ lưu vector trọng số w đại diện cho mối quan hệ giữa các đặc trưng X và đánnhgiá y .

Vector trọng số này sẽ được sử dụng để dự đoán đánh giá cho các phim mới:

- Huấn luyện mô hình Ridge cho tất cả người dùng từ hàm xây dựng cho từng người dùng.
  - Lặp qua tất cả các người dùng ấy danh sách tất cả người dùng trong tập huấn luyện.
  - Áp dụng hàm train_user_ridge_model cho từng người dùng: Sử dụng danh sách người dùng để huấn luyện mô hình riêng biệt cho mỗi người dùng.

Sau khi học xong sẽ lưu dictionary user_models chứa mô hình Ridge cho tất cả người dùng trong tập huấn luyện. Bạn có thể sử dụng các mô hình này để dự đoán điểm đánh giá cho từng người dùng với các bộ phim chưa được đánh giá.

Bước 5: Dự đoán đánh giá của người dùng, đề xuất bộ phim cho người dùng trên tập kiểm tra và tính toán RMSE

- Dự đoán đánh giá cho các phim trong tập kiểm tra
  - Lấy chỉ số bộ phim trong tập dữ liệu test
  - Lấy vector đặc trưng của bộ phim từ ma trận TFIDF
  - Lấy vector thể loại của người dùng từ ma trận người dùng - thể loại
  - Kết hợp vector bộ phim và vector thể loại của người dùng
  - Dự đoán rating cho bộ phim của người dùng bằng mô hình Ridge cho tất cả người dùng: Mô hình Ridge Regression sẽ dự đoán điểm đánh giá bằng cách tính toán tích vô hướng giữa vector đặc trưng (X) và vector trọng số (w), cộng với hệ số chệch (bias) (nếu có). Công thức dự đoán có thể được viết dưới dạng:

$$\hat{y} = Xw + b$$

- $\hat{y}$ là điểm đánh giá dự đoán.
- X là vector đặc trưng (bao gồm cả TF-IDF của phim và thể loại của người dùng).
- w là vector trọng số của mô hình Ridge.
- b là hệ số chệch (bias), nếu có.

- Đề xuất bộ phim cho người dùng trên tập kiểm tra
  - Gọi hàm dự đoán đánh giá cho các phim đã tạo
  - Sắp xếp điểm đánh giá từ cao đến thấp cho 610 đề xuất bộ phim đó cho người dùng
- Tính toán RMSE dựa trên rating thực tế và rating dự đoán trong tập kiểm tra
  - Duyệt qua tập kiểm tra lấy giá trị thực tế rating, lấy giá tị dự đoán từ hàm xây dựng dự đoán rating từ mô hình. Sử dụng công thức sau để tính:

Công thức tính RMSE là:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{\text{true},i} - y_{\text{pred},i})^2}$$

Trong đó:

- n là số lượng điểm dữ liệu.
- $y_{\text{true},i}$ là giá trị thực tế tại chỉ số i.
- $y_{\text{pred},i}$ là giá trị dự đoán tại chỉ số i.

5.2 User-user Collaborative Filtering

Lọc cộng tác dựa trên người dùng đưa ra các đề xuất (gợi ý) dựa trên tương tác giữa người dùng và sản phẩm trong quá khứ. Giả định đằng sau thuật toán là những người dùng tương tự thích các sản phẩm tương tự.

Thuật toán lọc cộng tác dựa trên người dùng thường có các bước sau:

1. Tìm người dùng tương tự dựa trên tương tác với các mục chung.
2. Xác định các mục được người dùng tương tự đánh giá cao nhưng chưa được người dùng quan tâm sử dụng.
3. Tính điểm trung bình có trọng số cho mỗi mục.
4. Xếp hạng các mục dựa trên điểm số và chọn k mục hàng đầu để đề xuất.

Triển khai:

Bước 1: Tạo ma trận User-Item (hình a)

- Hàng đại diện cho các mục (items).

- Cột đại diện cho người dùng (users).
- Giá trị trong ô là điểm đánh giá mà người dùng u dành cho mục i.
- Ô trống (nếu người dùng chưa đánh giá) sẽ được điền bằng '?'

Bước 2: Chuẩn hóa điểm đánh giá: (hình b)

- Tính điểm trung bình của mỗi người dùng .
- Chuẩn hóa ma trận ban đầu User-Item:
  - Mỗi rating ban đầu trừ đi điểm trung bình của người dùng tương ứng.
  - Các giá trị chưa biết (?) được thay bằng 0
  - Lưu trữ dưới dạng ma trận thưa (sparse matrix) để tối ưu tính toán.

Bước 3: Tính ma trận độ tương đồng (User Similarity Matrix): (hình c)

- Tính toán độ tương đồng:

  Sử dụng hàm Cosine Similarity để tính độ tương đồng giữa các vectors (các vectors đã được chuẩn hóa tương ứng với mỗi user)

$$\text{cosine\_similarity}(\mathbf{u}_1, \mathbf{u}_2) = \cos(\mathbf{u}_1, \mathbf{u}_2) = \frac{\mathbf{u}_1^T \mathbf{u}_2}{||\mathbf{u}_1||_2 . ||\mathbf{u}_2||_2} \quad (1)$$

- Tạo ma trận tương đồng người dùng (User similarity matrix), với giá trị [u,v]biểu thị mức độ tương đồng giữa u và v.

Bước 4: Dự đoán các ratings còn thiếu: (hình d)

- Xác định các users đã đánh giá cho item.
- Lấy users tương đồng nhất: Xác định k users gần nhất (nearest neighbors) với user cần dự đoán, dựa trên độ tương đồng.

- Tính rating dự đoán:

$$\hat{y}_{i,u} = \frac{\sum_{u_j \in \mathcal{N}(u,i)} \bar{y}_{i,u_j} \text{sim}(u, u_j)}{\sum_{u_j \in \mathcal{N}(u,i)} |\text{sim}(u, u_j)|}$$

Trong đó:

- $\hat{y}_{i,u}$: Điểm dự đoán của user u cho item i
- $\mathcal{N}(u,i)$ là tập hợp k users có độ tương đồng (*similarity*) cao nhất với u mà **đã rating** i.
- $\bar{y}_{i,u_j}$: là rating của users u_j cho item i đã được chuẩn hóa
- $\text{sim}(u,u_j)$: Độ tương đồng giữa user u và các users u_j

Bước 5: Đề xuất items cho người dùng:

- Xác định các items mà người dùng mục tiêu user u chưa đánh giá.
- Sắp xếp items theo điểm dự đoán cao nhất.
- Đề xuất.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	5	5	2	0	1	?	?
i_1	4	?	?	0	?	2	?
i_2	?	4	1	?	?	1	1
i_3	2	2	3	4	4	?	4
i_4	2	0	4	?	?	?	5
u_j	3.25	2.75	2.5	1.33	2.5	1.5	3.33

a) Original utility matrix $\mathbf{Y}$ and mean user ratings.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	1.75	2.25	-0.5	-1.33	-1.5	0	0
i_1	0.75	0	0	-1.33	0	0.5	0
i_2	0	1.25	-1.5	0	0	-0.5	-2.33
i_3	-1.25	-0.75	0.5	2.67	1.5	0	0.67
i_4	-1.25	-2.75	1.5	0	0	0	1.67

b) Normalized utility matrix $\bar{\mathbf{Y}}$.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
u_0	1	0.83	-0.58	-0.79	-0.82	0.2	-0.38
u_1	0.83	1	-0.87	-0.40	-0.55	-0.23	-0.71
u_2	-0.58	-0.87	1	0.27	0.32	0.47	0.96
u_3	-0.79	-0.40	0.27	1	0.87	-0.29	0.18
u_4	-0.82	-0.55	0.32	0.87	1	0	0.16
u_5	0.2	-0.23	0.47	-0.29	0	1	0.56
u_6	-0.38	-0.71	0.96	0.18	0.16	0.56	1

c) User similarity matrix $\mathbf{S}$.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	1.75	2.25	-0.5	-1.33	-1.5	0.18	-0.63
i_1	0.75	0.48	-0.17	-1.33	-1.33	0.5	0.05
i_2	0.91	1.25	-1.5	-1.84	-1.78	-0.5	-2.33
i_3	-1.25	-0.75	0.5	2.67	1.5	0.59	0.67
i_4	-1.25	-2.75	1.5	1.57	1.56	1.59	1.67

d) $\hat{\mathbf{Y}}$

Predict normalized rating of u_1 on i_1 with $k = 2$

Users who rated i_1 : $\{u_0, u_3, u_5\}$

Corresponding similarities: {0.83, -0.40, -0.23}

$\Rightarrow$ most similar users: $\mathcal{N}(u_1, i_1) = \{u_0, u_5\}$

with **normalized ratings** {**0.75**, **0.5**}

$$\Rightarrow \hat{y}_{i_1,u_1} = \frac{0.83*0.75+(-0.23)*0.5}{0.83+|-0.23|} \approx 0.48$$

e) Example

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	5	5	2	0	1	1.68	2.70
i_1	4	3.23	2.33	0	1.67	2	3.38
i_2	4.15	4	1	-0.5	0.71	1	1
i_3	2	2	3	4	4	2.10	4
i_4	2	0	4	2.9	4.06	3.10	5

f) Full $\mathbf{Y}$

5.3 Item-item Collaborative Filtering

Item-Item Based Collaborative Filtering là một phương pháp trong hệ thống gợi ý, tập trung vào việc tìm kiếm sự tương đồng giữa các mục (items) dựa trên hành vi đánh giá của người dùng. Thay vì so sánh người dùng với nhau như User-User Based, phương pháp này xem xét các mục đã được người dùng đánh giá và xác định các mục có xu hướng được đánh giá tương tự. Kết quả là hệ thống có thể gợi ý các mục mà người dùng có thể quan tâm dựa trên sự tương tự với các mục họ đã thích hoặc sử dụng trước đó.

Cách hoạt động:

Bước 1: Tạo ma trận Utility Matrix (hình a)

- Tạo ma trận từ dữ liệu đánh giá, trong đó hàng là các items, cột là các users, và giá trị là ratings của user với item
- Các giá trị ratings chưa biết (?) được để trống

Bước 2: Chuẩn hóa dữ liệu (hình b)

- Tính trung bình rating của mỗi user
- Chuẩn hóa ma trận Utility:
  - Trừ mỗi rating của user với trung bình của user đó
  - Các giá trị chưa biết (?) được thay bằng 0 để tối ưu tính toán và lưu trữ dưới dạng ma trận thưa (sparse matrix).

Bước 3: Tính ma trận độ tương đồng (Item Similarity Matrix): (hình c)

- Tính toán độ tương đồng: Sử dụng hàm Cosine Similarity để tính độ tương đồng giữa các vectors (các vectors đã được chuẩn hóa tương ứng với mỗi item)

$$\text{cosine\_similarity}(\mathbf{u}_1, \mathbf{u}_2) = \cos(\mathbf{u}_1, \mathbf{u}_2) = \frac{\mathbf{u}_1^T \mathbf{u}_2}{||\mathbf{u}_1||_2 . ||\mathbf{u}_2||_2} \quad (1)$$

- Tạo ma trận đối xứng, trong đó mỗi ô (i, j) là độ tương đồng giữa item i và item j

Bước 4: Dự đoán các ratings còn thiếu: (hình d)

- Xác định các items mà user đã đánh giá
- Tìm các items tương đồng nhất: Xác định k items gần nhất (nearest neighbors) với item cần dự đoán, dựa trên độ tương đồng.
- Tính rating dự đoán:

$$\hat{y}_{i,u} = \frac{\sum_{i_j \in N(i,u)} y_{i_j,u} \cdot sim(i, i_j)}{\sum_{i_j \in N(i,u)} |sim(i, i_j)|}$$

Trong đó:

- $\hat{y}_{i,u}$: Điểm dự đoán của user u với item i
- $N(i,u)$: Tập hợp k items gần nhất đã được user u đánh giá

- $y_{i_j,u}$: Rating của user u cho item ij
- $sim(i, i_j)$: Độ tương đồng giữa item i và ij

Bước 5: Quyết định gợi ý items:

- Xác định các items chưa được đánh giá với mỗi user
- Sắp xếp items theo điểm dự đoán
- Gợi ý

	u_0	u_1	u_2	u_3	u_4	u_5	u_6		
i_0	5	5	2	0	1	?	?	→	2.6
i_1	4	?	?	0	?	2	?	→	2
i_2	?	4	1	?	?	1	1	→	1.75
i_3	2	2	3	4	4	?	4	→	3.17
i_4	2	0	4	?	?	?	5	→	2.75

a) Original utility matrix $\mathbf{Y}$ and mean item ratings.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	2.4	2.4	-.6	-2.6	-1.6	0	0
i_1	2	0	0	-2	0	0	0
i_2	0	2.25	-0.75	0	0	-0.75	-0.75
i_3	-1.17	-1.17	-0.17	0.83	0.83	0	0.83
i_4	-0.75	-2.75	1.25	0	0	0	2.25

b) Normalized utility matrix $\bar{\mathbf{Y}}$.

	i_0	i_1	i_2	i_3	i_4
i_0	1	0.77	0.49	-0.89	-0.52
i_1	0.77	1	0	-0.64	-0.14
i_2	0.49	0	1	-0.55	-0.88
i_3	-0.89	-0.64	-0.55	1	0.68
i_4	-0.52	-0.14	-0.88	0.68	1

c) Item similarity matrix $\mathbf{S}$.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	2.4	2.4	-.6	-2.6	-1.6	-0.29	-1.52
i_1	2	2.4	-0.6	-2	-1.25	0	-2.25
i_2	2.4	2.25	-0.75	-2.6	-1.20	-0.75	-0.75
i_3	-1.17	-1.17	-0.17	0.83	0.83	0.34	0.83
i_4	-0.75	-2.75	1.25	1.03	1.16	0.65	2.25

d) Normalized utility matrix $\bar{\mathbf{Y}}$.

Chương 6: Đánh giá mô hình

Sử dụng chỉ số RMSE để đánh giá mô hình, kết quả 3 mô hình như sau:

- **Content-based Filtering**: RMSE = 0.9119.

Có chỉ số RMSE cao nhất trong các mô hình, cho thấy thể loại phim không đủ chi tiết để phản ánh đúng sở thích thực sự của người dùng.

- **User-based Collaborative Filtering:** RMSE = 0.1021.

RMSE thấp nhất, chứng tỏ rằng hành vi đánh giá của người dùng trong dữ liệu có tính nhất quán cao, và việc khai thác mối quan hệ giữa các người dùng trở nên hiệu quả hơn so với các mô hình còn lại.

- **Item-based Collaborative Filtering:** RMSE = 0.3893.

Hiệu suất tốt hơn content-based nhưng kém hơn user-based, có thể do số lượng items lớn và dữ liệu đánh giá thưa khiến việc tính toán tương đồng giữa các bộ phim kém chính xác.

Chương 7: Ưu Nhược Điểm Của Các Mô Hình

Mô hình	Ưu điểm	Nhược điểm
Content-based Filtering	• Không phụ thuộc vào dữ liệu người dùng khác: không cần phải có dữ liệu từ các người dùng khác để đưa ra gợi ý. Hệ thống chỉ dựa vào các đặc điểm của nội dung phim, lịch sử đánh giá người dùng theo thể loại... để đưa ra gợi ý • Đơn giản	• Cần nhiều dữ liệu về các mô tả phim, đạo diễn, diễn viên, ... Phụ thuộc vào việc phân tích nội dung chính xác • Có xu hướng gợi ý những mục tương tự với những gì người dùng đã xem hoặc đánh giá. Điều này có thể dẫn đến việc thiếu tính đa dạng trong các gợi ý, khiến người dùng không được khám phá các lựa chọn mới hoặc khác biệt.
User-user Collaborative Filtering	• Gợi ý dựa trên trải nghiệm của người dùng tương tự khác nên có thể gợi ý được những sản phẩm phù hợp với sở thích. • Phù hợp với những hệ thống lớn có nhiều đánh giá từ người dùng. • Có khả năng dự đoán được sở thích và nhu cầu của người dùng để đưa ra gợi ý mà không cần hiểu sản phẩm.	• Trên thực tế số lượng users lớn hơn số lượng items rất nhiều. Similarity matrix là rất lớn nên việc lưu trữ tốn nhiều tài nguyên. • Ma trận Utility matrix Y thường rất trống vì users thường lười rating nên khi user đó thay đổi rating hay thêm rating thì giá trị chuẩn hóa sẽ bị thay đổi nhiều. • Không thể gợi ý được những sản phẩm mới hoặc những sản phẩm chưa được ai đánh giá.

Item-item Collaborative Filtering	• Tập trung vào độ tương đồng giữa các items, giúp giảm tác động dữ liệu của từng user riêng lẻ, hạn chế sự thiên lệch của user cá biệt • Hiệu quả với các tập dữ liệu lớn. • Dễ mở rộng và phù hợp với các hệ thống có số lượng users lớn hơn số lượng items.	• Ít cá nhân hóa hơn, vì tập trung vào tương đồng giữa items thay vì từng user cụ thể. • Cold start problem: Gặp khó khăn khi có item mới, vì không có đủ thông tin để xác định tương đồng với các items khác.

Kết Luận

User-based CF là lựa chọn tối ưu nhờ khai thác hiệu quả mối quan hệ giữa người dùng. Trong khi đó, content-based hoạt động kém hiệu quả hơn do bị giới hạn bởi các đặc trưng nội dung chưa đủ chi tiết. Item-based CF đạt kết quả khá nhưng vẫn thấp hơn user-based do dữ liệu thưa và số lượng phim lớn, tuy nhiên có thể cải thiện nếu kết hợp trong một hệ thống hybrid hoặc với dữ liệu phong phú hơn.

Tài Liệu Tham Khảo

Nguyễn, T. H. (2017, May 17). *Học máy cơ bản - Content-based Recommender System*. Machine Learning Cơ Bản. Retrieved from https://machinelearningcoban.com/2017/05/17/contentbasedrecommendersys/

Stanford University. (n.d.). *Mining of Massive Datasets - Chapter 9: Recommender Systems*. Retrieved from http://infolab.stanford.edu/~ullman/mmds/ch9.pdf

Simplilearn. (2020, June 27). *Content-Based Recommendation System | Recommendation Engine Machine Learning Tutorial* [Video]. YouTube. Retrieved from https://www.youtube.com/watch?v=2uxXPzm-7FY

Stanford University. (n.d.). *Mining of Massive Datasets - Chapter 9: Recommender Systems*. Retrieved from http://infolab.stanford.edu/~ullman/mmds/ch9.pdf

Simplilearn. (2020, June 27). *What is Collaborative Filtering? Collaborative Filtering Explained with Examples | Machine Learning Tutorial* [Video]. YouTube. Retrieved from https://www.youtube.com/watch?v=h9gpufJFF-0&t=436s

Nguyễn, T. H. (2017, May 24). Học máy cơ bản - Collaborative Filtering. Machine Learning Cơ Bản. Retrieved from https://machinelearningcoban.com/2017/05/24/collaborativefiltering/