

Phân tích dữ liệu



# MOVIE RECOMMENDER SYSTEM



# Thành Viên

Bùi Dạ Lý - 2254052042

Võ Thị Ngọc Chi - 2254052008

Huỳnh Lê Giang - 2254050009



## 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).

# Nội dung



- Chương 1: Giới thiệu về dataset
- Chương 2: Tiền xử lý
- Chương 3: Trực Quan Hóa
- Chương 4: Xử lý outlier và Chuyển đổi dữ liệu
- Chương 5: Các mô hình thực hiện
- Chương 6: Ưu nhược điểm của các mô hình



# Giới Thiệu Về Dataset

**CHƯƠNG**  
**01**

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.csv 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.



# Tiền Xử Lý Dữ Liệu

# CHƯƠNG 02

# File movies.csv

movieId		title	genres	year
0	1	toy story	[adventure, animation, children, comedy, fantasy]	1995
1	2	jumanji	[adventure, children, fantasy]	1995
2	3	grumpier old men	[comedy, romance]	1995
3	4	waiting to exhale	[comedy, drama, romance]	1995
4	5	father of the bride part ii	[comedy]	1995
...	...	...	...	...
9737	193581	black butler: book of the atlantic	[action, animation, comedy, fantasy]	2017
9738	193583	no game no life: zero	[animation, comedy, fantasy]	2017
9739	193585	flint	[drama]	2017
9740	193587	bungo stray dogs: dead apple	[action, animation]	2018
9741	193609	andrew dice clay: dice rules	[comedy]	1991

9742 rows x 4 columns

- Chuyển đổi giá trị cột genres về chữ thường
- Chuyển đổi genres về dạng danh sách
- Chuyển đổi cột title về chữ thường
- Trích xuất thông tin năm (year) từ title
- Thay thế giá trị thiếu trong cột year bằng 'unknown'
- Xóa thông tin năm khỏi cột title
- Kiểm tra giá trị null

```
movies.isnull().sum()
movieId    0
title     0
genres    0
dtype: int64
```

# File ratings.csv

	userId	movieId	rating
0	1	1	4.0
1	1	3	4.0
2	1	6	4.0
3	1	47	5.0
4	1	50	5.0
...	...	...	...
100831	610	166534	4.0
100832	610	168248	5.0
100833	610	168250	5.0
100834	610	168252	5.0
100835	610	170875	3.0

```
0  
userId 0  
movieId 0  
rating 0  
timestamp 0  
dtype: int64
```

- Kiểm tra giá trị null
- Kiểm tra các hàng trùng lặp
- Bỏ cột Timestamp

100836 rows × 3 columns

# Kết hợp dữ liệu từ 2 file

	userId	movieId	rating	title	genres	year
0	1	1	4.0	toy story	[adventure, animation, children, comedy, fantasy]	1995
1	1	3	4.0	grumpier old men	[comedy, romance]	1995
2	1	6	4.0	heat	[action, crime, thriller]	1995
3	1	47	5.0	seven (a.k.a. se7en)	[mystery, thriller]	1995
4	1	50	5.0	usual suspects, the	[crime, mystery, thriller]	1995
...	...	...	...	...	...	...
100831	610	166534	4.0	split	[drama, horror, thriller]	2017
100832	610	168248	5.0	john wick: chapter two	[action, crime, thriller]	2017
100833	610	168250	5.0	get out	[horror]	2017
100834	610	168252	5.0	logan	[action, sci-fi]	2017
100835	610	170875	3.0	the fate of the furious	[action, crime, drama, thriller]	2017

100836 rows × 6 columns

	userId	movieId	rating
count	100836.000000	100836.000000	100836.000000
mean	326.127564	19435.295718	3.501557
std	182.618491	35530.987199	1.042529
min	1.000000	1.000000	0.500000
25%	177.000000	1199.000000	3.000000
50%	325.000000	2991.000000	3.500000
75%	477.000000	8122.000000	4.000000
max	610.000000	193609.000000	5.000000

- Kiểm tra giá trị null
- Kiểm tra các hàng trùng lặp

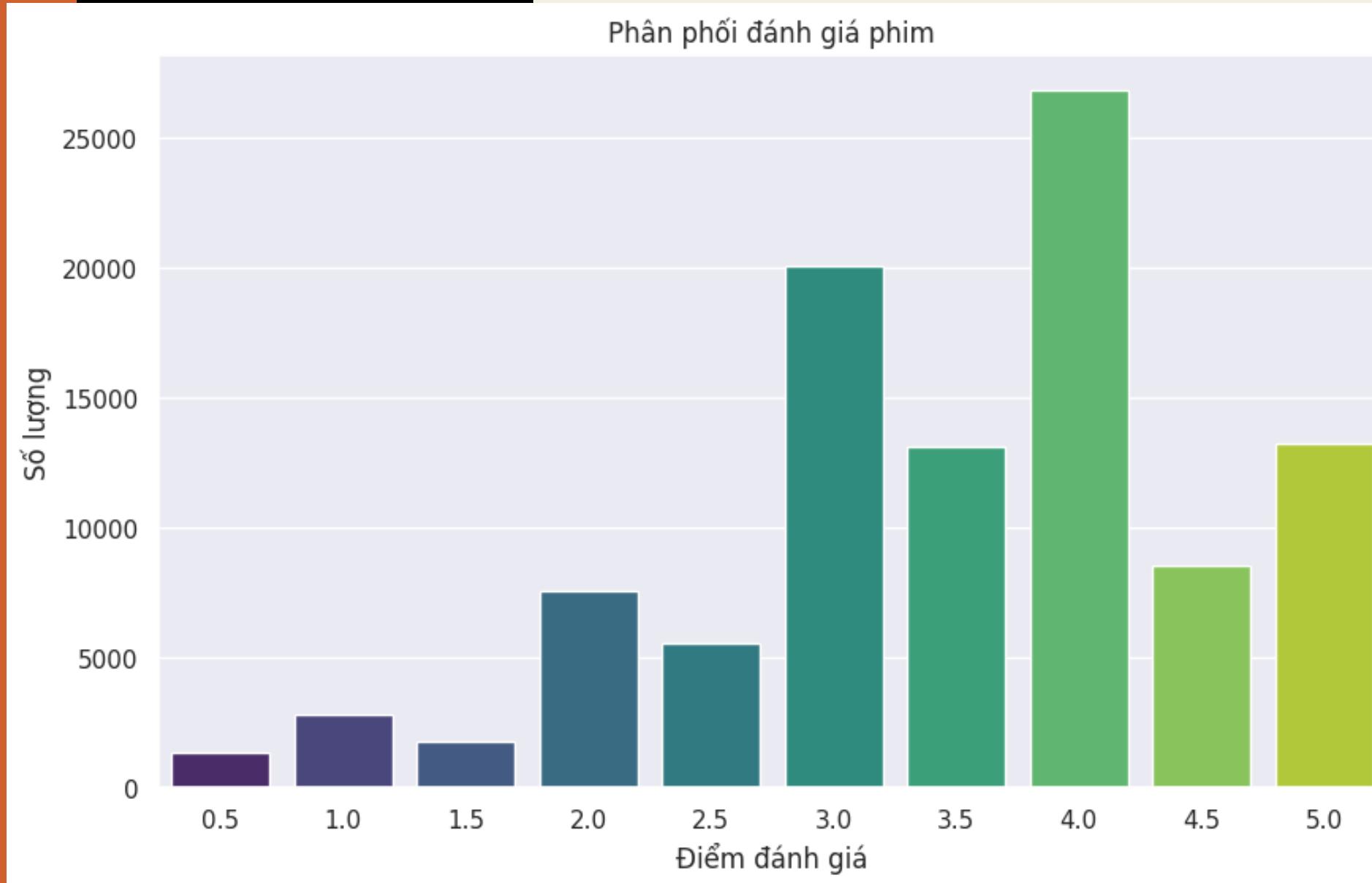


# Trực Quan Hóa Dữ Liệu

# CHƯƠNG 03

# Trực Quan Hóa Dữ Liệu

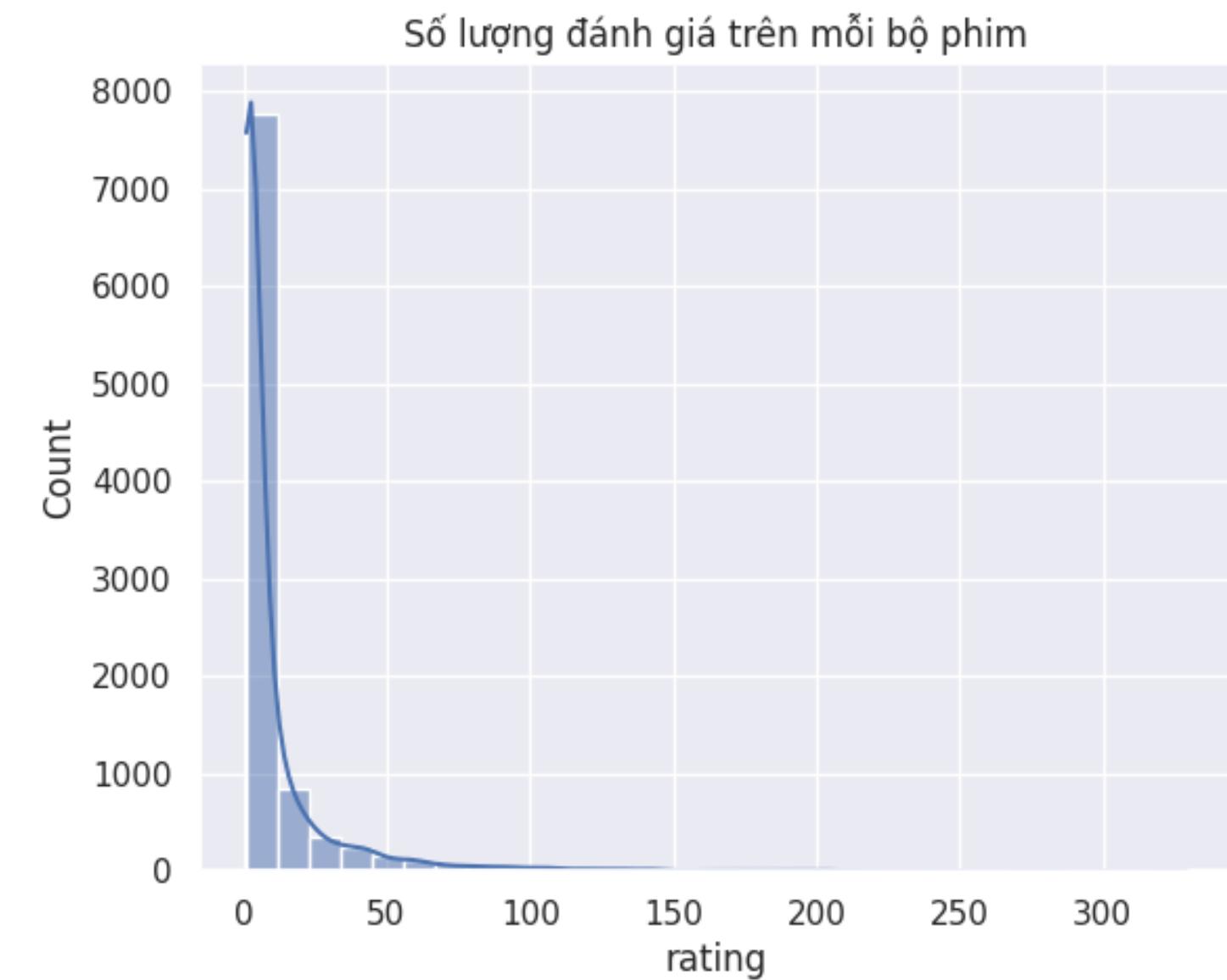
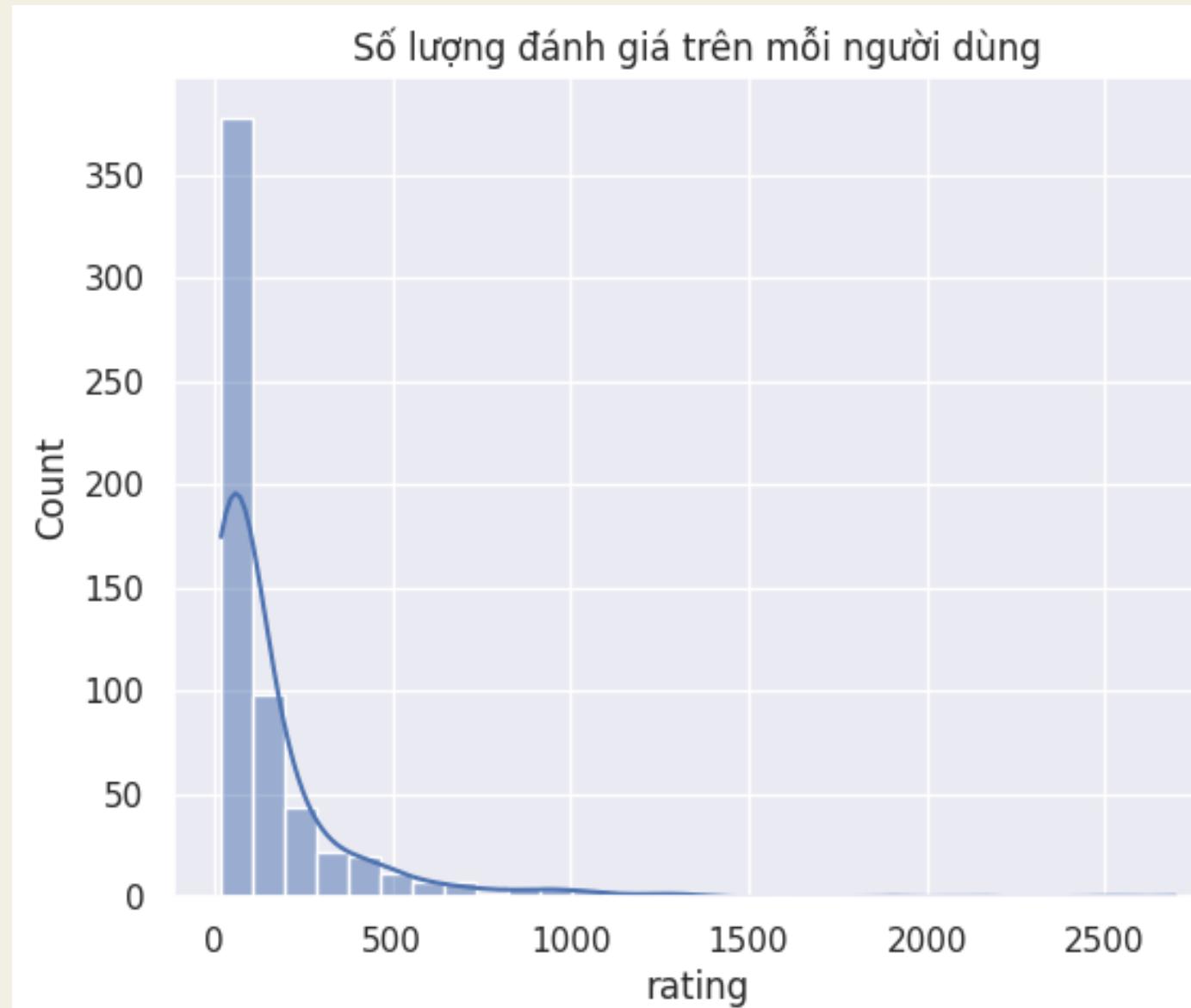
## Đặc điểm:



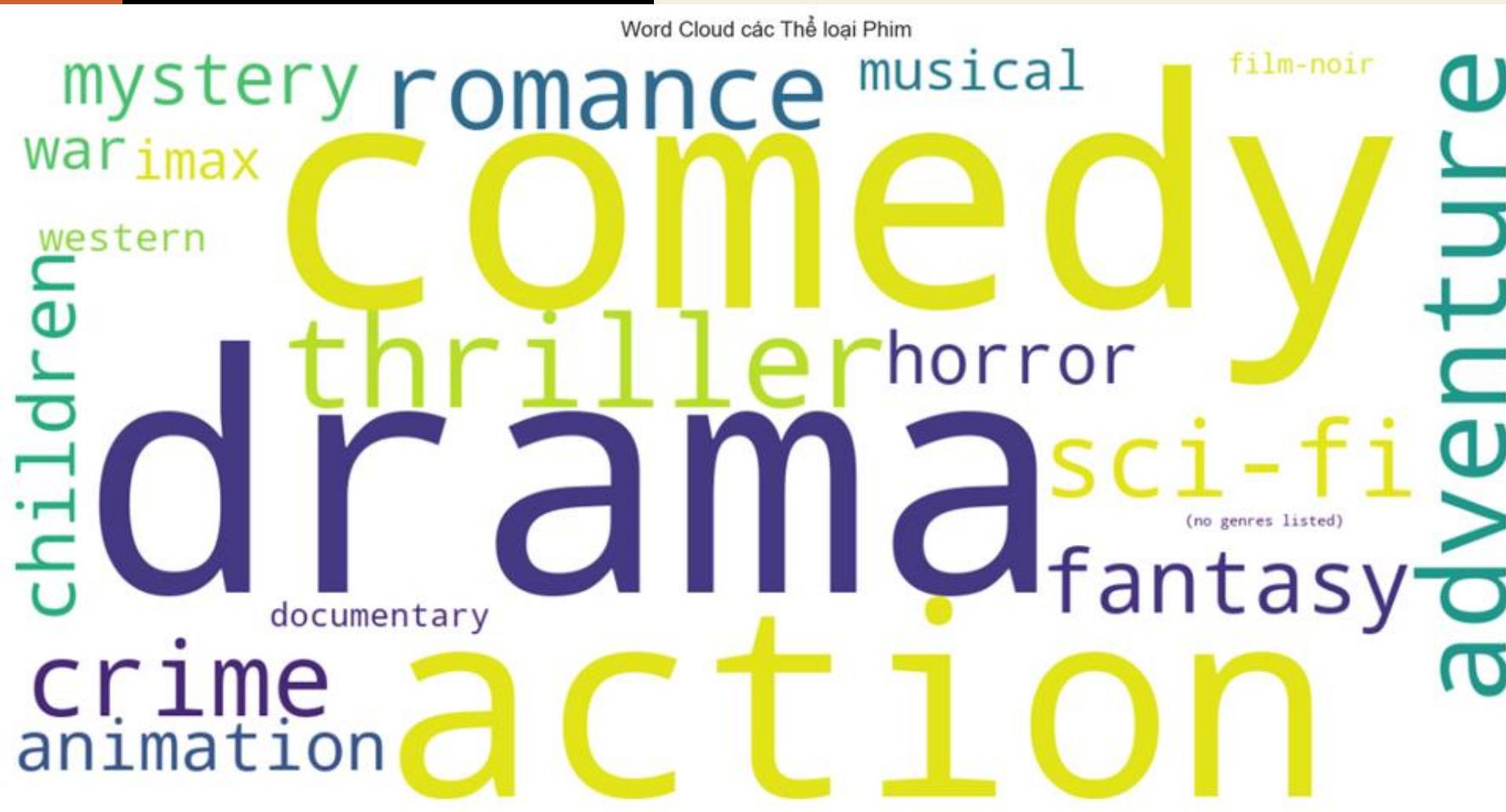
- Điểm đánh giá tập trung chủ yếu ở mức 3.0, 4.0, và 5.0.
- Rất ít người dùng đánh giá phim dưới mức 2.0.
- Điều này cho thấy xu hướng thiên lệch tích cực trong đánh giá.

- Số lượng đánh giá của mỗi người dùng phân phối không đồng đều, với đa số người dùng chỉ đánh giá rất ít phim (dưới 50 phim).
- Một số ít người dùng đánh giá rất nhiều phim (hơn 500 phim), gây ra sự bất thường.

- Số lượng đánh giá trên mỗi bộ phim cũng không đồng đều, với đa số các phim chỉ nhận được rất ít đánh giá (dưới 50 lượt).
- Một số ít phim được đánh giá rất nhiều (hơn 100 lượt).

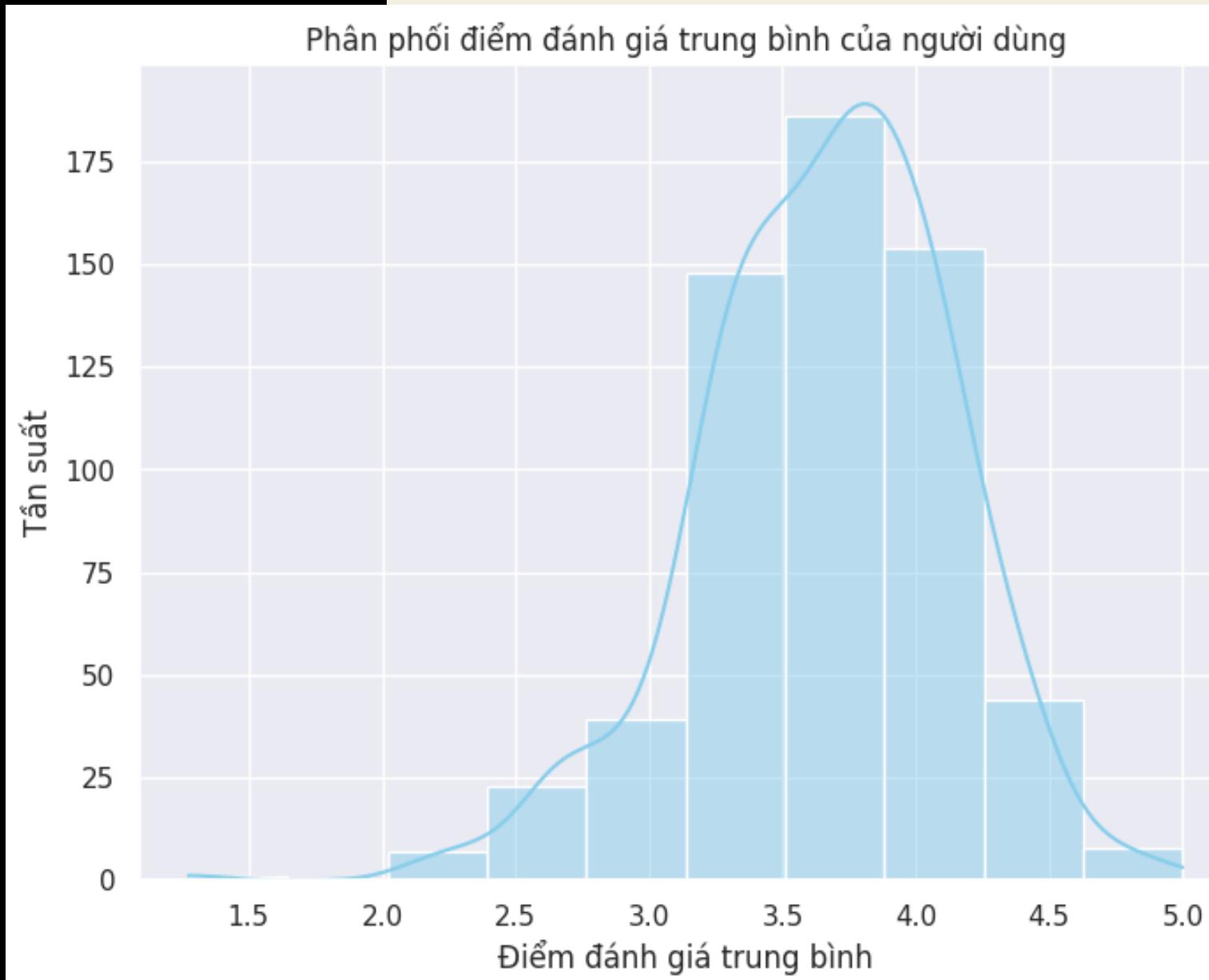


# Trực Quan Hóa Dữ Liệu



- **Thể loại phổ biến:** Các thể loại như Drama, Comedy, và Action chiếm ưu thế rõ rệt, phản ánh sở thích chung của phần lớn người xem.
- **Sự đa dạng:** Bên cạnh các thể loại chính, các thể loại khác như Romance, Thriller, và Adventure cũng được nhấn mạnh, cho thấy sự đa dạng trong sở thích của người dùng.
- **Thể loại ít phổ biến:** Một số thể loại như Film-Noir, Documentary, hoặc IMAX xuất hiện ít hơn, phản ánh nhóm khán giả đặc thù cho các thể loại này.

# Trực Quan Hóa Dữ Liệu



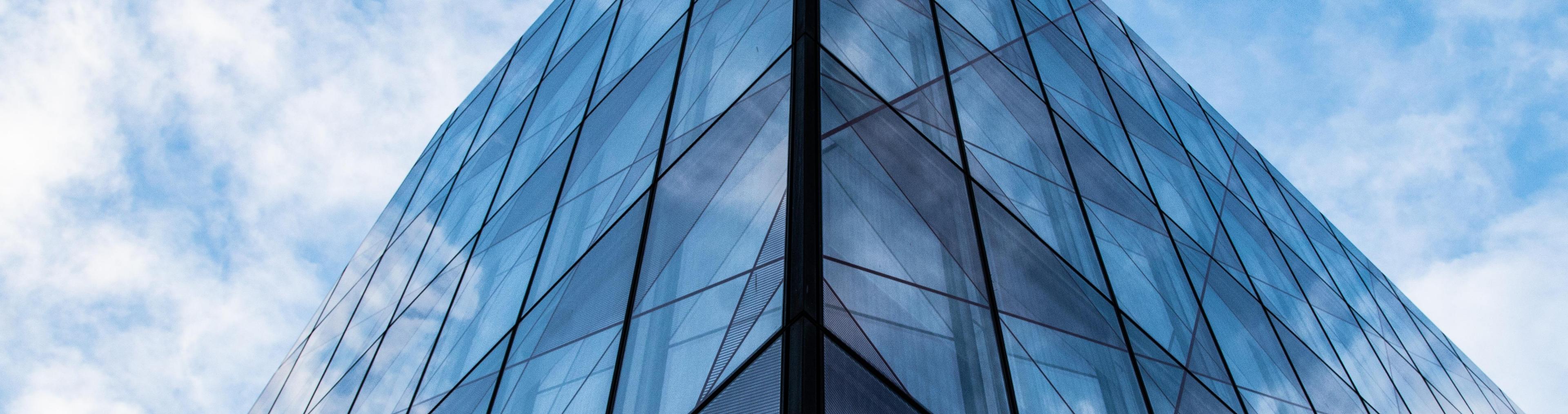
**Mục đích :** Xác định xu hướng đánh giá của người dùng (người khó tính, người dễ tính):

- Điểm đánh giá trung bình của người dùng có dạng phân phối gần chuẩn (Gaussian).
- Phần lớn người dùng có điểm đánh giá trung bình từ 3.0 đến 4.0.
- Tuy nhiên, vẫn có một số người dùng đánh giá cực kỳ thấp hoặc cao toàn bộ (tất cả điểm đều dưới 2.0 hoặc đều 5.0).

# Trực Quan Hóa Dữ Liệu

## 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).
- Dữ liệu rating rất thưa (sparsity > 95%).



# Xử lý outlier và Chuyển đổi dữ liệu

**CHƯƠNG**  
**04**

- 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)

=> Áp dụng một số phương pháp sau để giảm tác động của các thiên lệch đó, gồm:



## Bayesian Average



## Tỷ lệ nghịch

# Bayesian Average

$$\text{Adjusted Rating}_i = \frac{(\text{num\_ratings}_i \times \text{mean}_i) + (k \times \text{global\_mean})}{\text{num\_ratings}_i + k}$$

- $\text{num\_ratings}_i$ : Số lượng đánh giá của mục  $i$ .
- $\text{mean}_i$ : Điểm trung bình của mục  $i$ .
- $k$ : Tham số làm mượt (giả định thêm  $k$  đánh giá với điểm bằng  $\text{global\_mean}$ ).
- $\text{global\_mean}$ : Điểm trung bình toàn cục.

- **Mục đích:** Dùng Bayesian Average cho movieId để làm giảm ảnh hưởng của phim có quá ít hoặc quá nhiều đánh giá.
- **Hiệu quả:**
  - Những phim có nhiều đánh giá sẽ giữ được trung bình của riêng nó.
  - Những phim có ít đánh giá sẽ bị "kéo" về gần giá trị trung bình toàn cục.

# Trọng số tỷ lệ nghịch

- Người dùng thực hiện nhiều đánh giá ( $x$ ) sẽ có trọng số thấp.
- Người dùng thực hiện ít đánh giá ( $x$ ) sẽ có trọng số cao hơn.

Công thức:

$$\text{user\_weight} = \frac{\text{max\_count} - x}{\text{max\_count} - \text{min\_count}}$$

- **Mục đích:** Điều chỉnh ảnh hưởng của các người dùng dựa trên số lượng đánh giá mà họ thực hiện.
- **Logic:**
  - Người dùng đánh giá càng nhiều, trọng số của họ càng thấp (tỷ lệ nghịch).
  - Người dùng đánh giá ít, trọng số của họ cao hơn để cân bằng ảnh hưởng.
- **Hiệu quả:** Giảm tác động của người dùng có hành vi "spam" đánh giá (quá nhiều đánh giá với xu hướng không rõ ràng).



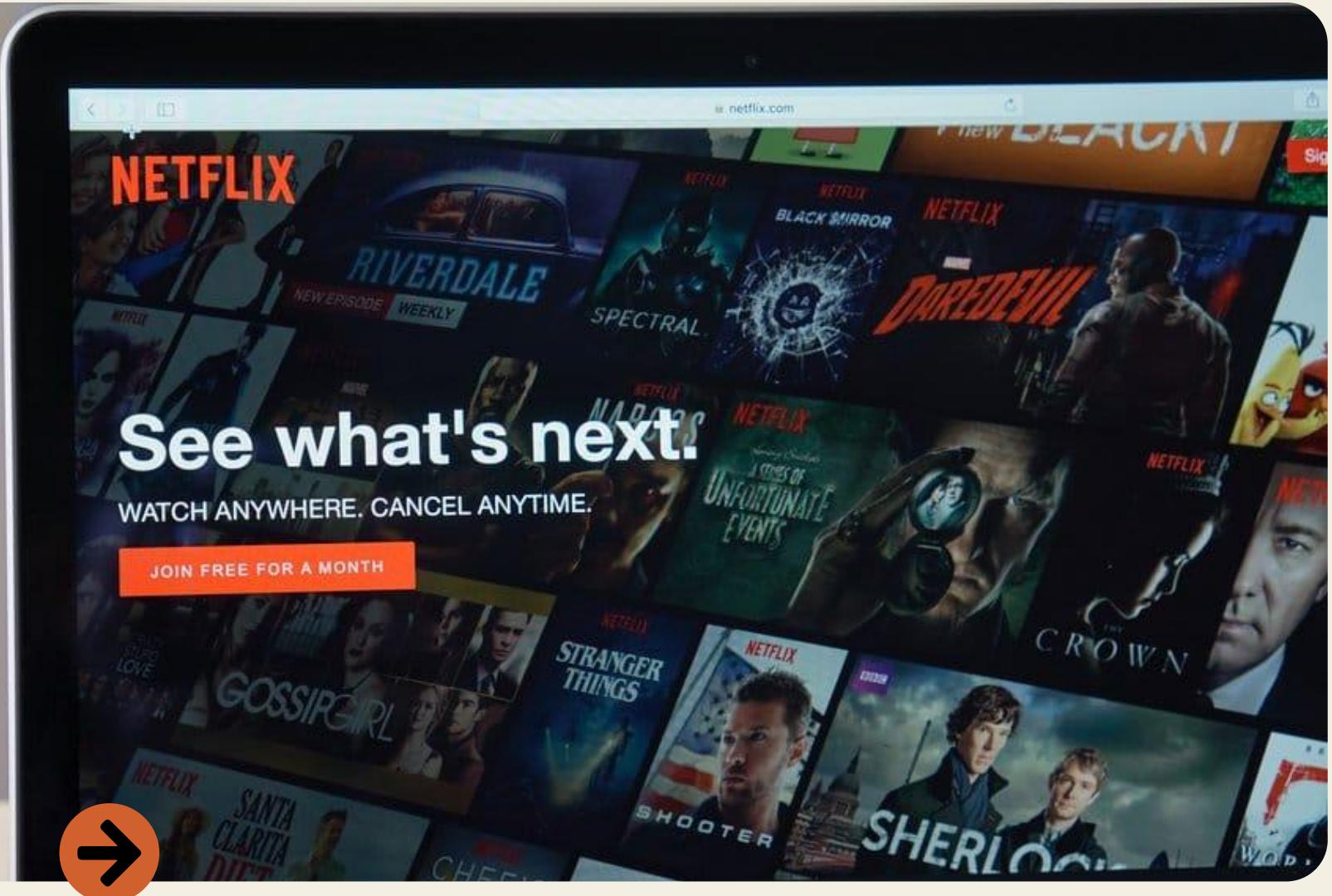
# Các Mô Hình Thực Hiện

**CHƯƠNG  
03**

# Xây dựng Mô Hình



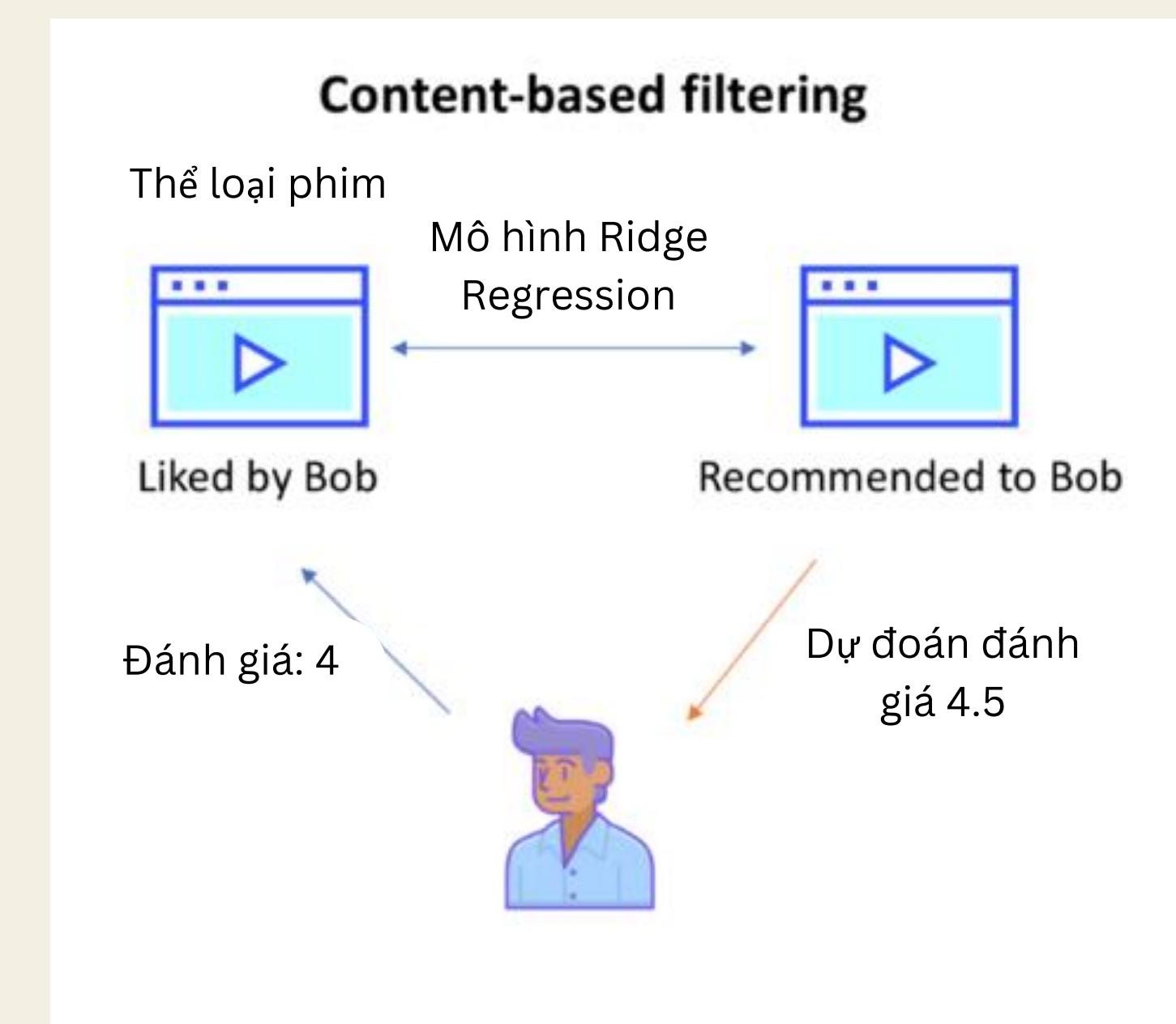
**Content-Based  
Filtering**



**Collaborative  
Filtering**

# 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 thể loại 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.



# Content-Based Filtering

## 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 lặp giữa các phim trong tập huấn 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.

Movield	Title	Genres	Dataset
1	Movie A	Action, Drama	Train
2	Movie B	Comedy, Drama	Train
3	Movie C	Action	Train
4	Movie D	Comedy	Test
5	Movie E	Drama	Test

# Content-Based Filtering

## 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 các 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

VD: Solo: A Star Wars Story thể loại: Action|Adventure|Children

	action	adventure	animation	children	comedy	crime
0	1	1	0	1	0	0

# Content-Based Filtering

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

- Sử dụng ma trận vừa tạo để chuyển đổi thành ma trận TF-IDF bằng TfidfTransformer
- Tính TF tính toán tần suất xuất hiện của một thể loại trong mỗi bộ phim. Nếu bộ phim có thể loại là "Drama, Mystery, Sci-Fi, Thriller" và có 4 thể loại, tần suất của mỗi thể loại  $\frac{1}{4} = 0.25$ .

$$TF(t, d) = \frac{\text{Số lần từ } t \text{ xuất hiện trong } d}{\text{Số từ tổng cộng trong } d}$$

- IDF (Inverse Document Frequency) tính toán mức độ quan trọng của một từ (hoặc thể loại) trong toàn bộ tập dữ liệu. Nếu có 10 bộ phim và từ "Drama" xuất hiện trong 7 bộ phim, IDF của "Drama" sẽ là:  $\log\left(\frac{10}{7}\right) = 0.155$

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

- TF-IDF bằng cách nhân TF và IDF

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

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

# Content-Based Filtering

## Bước 3: Xây dựng ma trận người dùng - thể loại

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

- Giả sử userId = 1 có các đánh giá sau cho thể loại Action:

- Đánh giá 1: 5.0
- Đánh giá 2: 3.0
- Đánh giá 3: 4.0

Trung bình của các đánh giá này là:

$$\text{Trung bình} = \frac{5.0 + 3.0 + 4.0}{3} = 4.0$$

genres	action	adventure	animation	children	comedy	crime
userId						
0	4.333333	4.345455	4.666667	4.518519	4.232143	4.266667
1	3.812500	3.750000	0.000000	0.000000	4.000000	3.714286
2	3.312500	3.000000	0.500000	0.500000	1.250000	0.500000
3	3.846154	3.736842	4.000000	3.666667	3.557377	3.722222
4	3.200000	3.400000	4.250000	4.000000	3.222222	4.000000

# Content-Based Filtering

## Bước 4: Áp dụng Ridge Regression 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 X huấn luyện

> 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.

- 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]

Kết hợp

- Sở thích thể loại của người dùng: [0.8, 0.2] (Action: 0.8, Drama: 0.2).



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

# Content-Based Filtering

Bước 4: Áp dụng Ridge Regression cho từng người dùng:

- 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

$$y_{\text{user}} = [4.0, 3.5, 5.0]$$

- 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: Mô hình sẽ lưu vector trọng số w đại diện cho mối quan hệ giữa các đặc trưng X và đánh giá y. Vector trọng số w này sẽ được sử dụng để dự đoán đánh giá cho các phim mới

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

Trong đó:

- $X$ : Ma trận đặc trưng (feature matrix).
- $y$ : Vector mục tiêu (ratings).
- $w$ : Trọng số cần tìm.
- $\alpha$ : Hệ số điều chỉnh. Giá trị lớn của  $\alpha$  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).

# Content-Based Filtering

Bước 4: Áp dụng Ridge Regression cho từng người dùng:

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

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

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

Trong đó:

- $X$ : Ma trận đặc trưng (feature matrix).
- $y$ : Vector mục tiêu (ratings).
- $w$ : Trọng số cần tìm.
- $\alpha$ : Hệ số điều chỉnh. Giá trị lớn của  $\alpha$  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).

# Content-Based Filtering

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 test

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 tích vô hướng giữa vector đặc trưng (X) và vector trọng số (w), cộng với hệ số chêch (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 chêch (bias), nếu có.

# Content-Based Filtering

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

- 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 người dùng sau đó đề xuất bộ phim đó cho người dùng

	userId	movieId	true_rating	predicted_rating
15	0	786	5.0	4.837474
38	0	1558	4.0	4.778394
33	0	1492	5.0	4.712084
40	0	1596	5.0	4.690045
19	0	913	5.0	4.661752
...	...	...	...	...
40165	609	7097	5.0	3.252782
40044	609	5523	4.5	3.213451
40079	609	6126	2.0	3.209167
39931	609	3057	5.0	3.160288
40061	609	5783	3.5	3.114322

# Đánh Giá Mô Hình Content-Based Filtering

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

- Tính toán RMSE dựa trên rating thực tế và rating dự đoán trong tập kiểm tra

Công thức tính RMSE là:

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

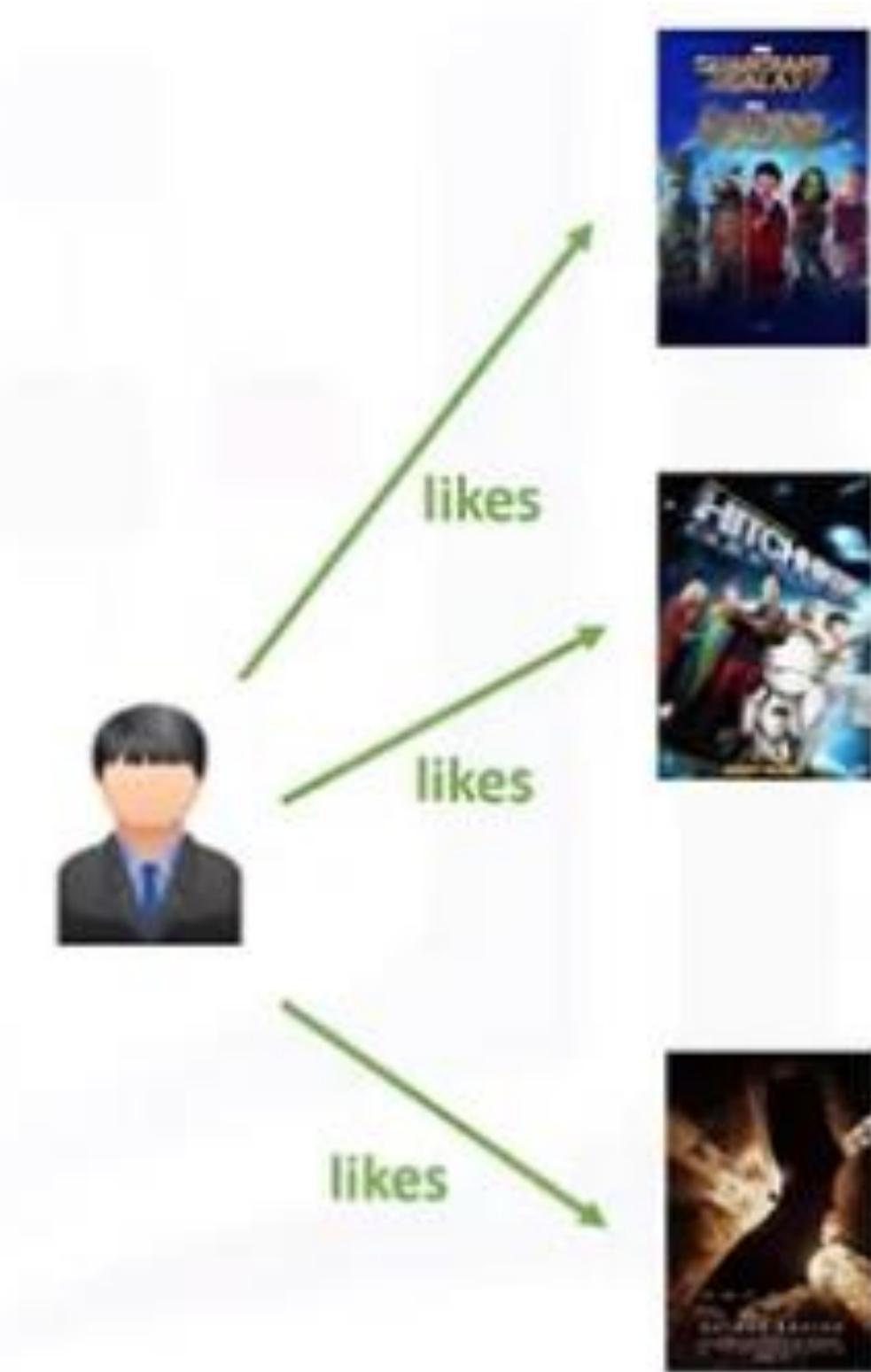
Trong đó:

- $n$  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  $i$ .

Root Mean Squared Error (RMSE): 0.9118882780482396

# Collaborative Filtering

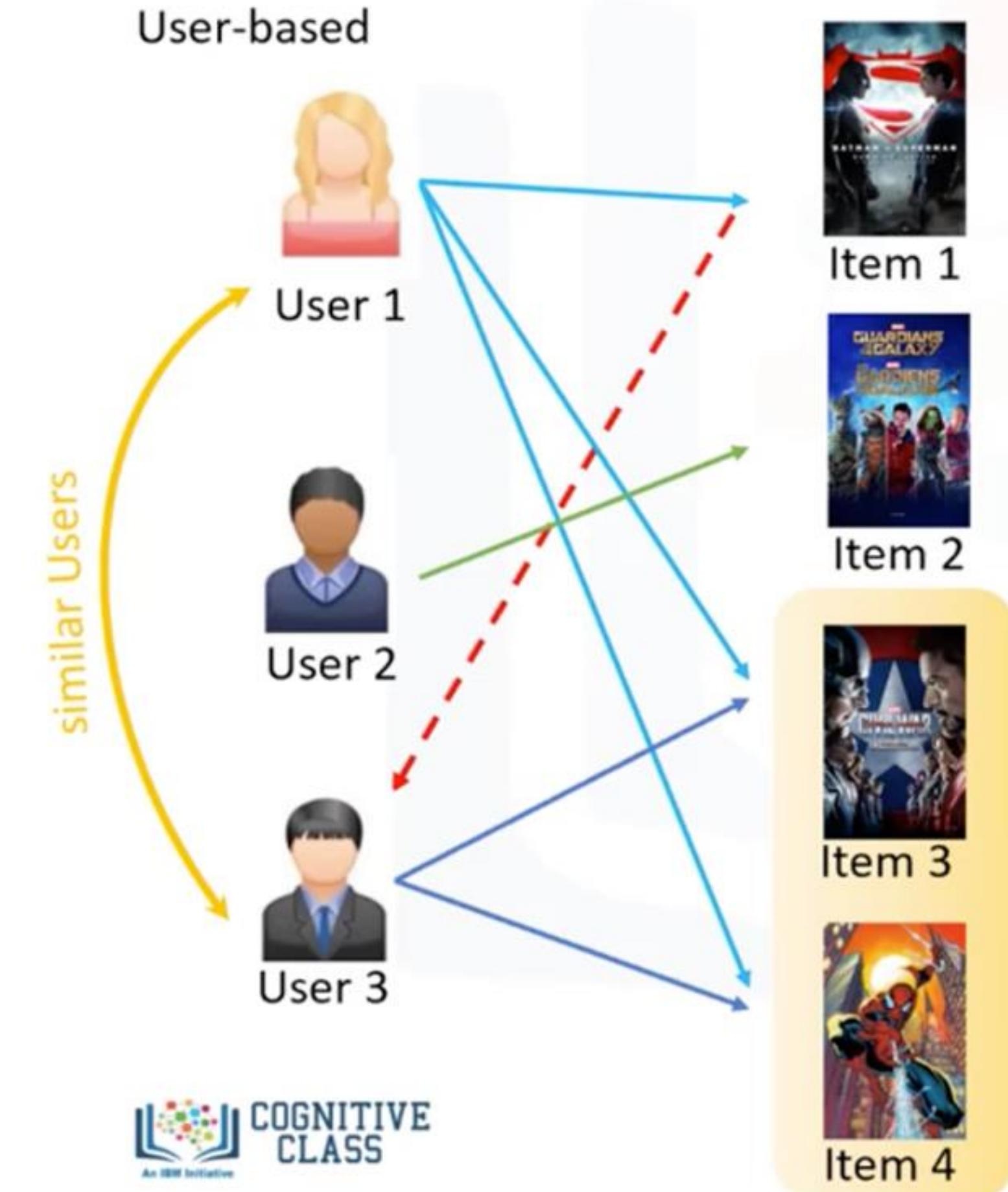
- **User-Based Collaborative Filtering:** các gợi ý được tạo ra dựa trên *sự tương đồng* giữa các người dùng. Ý tưởng chính là những người dùng tương tự thích các sản phẩm tương tự nhau.



# User-Based Collaborative Filtering

## Gồm các bước:

- Tìm người dùng tương tự.
- Xác định các mục chưa được người dùng quan tâm sử dụng.
- Tính điểm trung bình có trọng số cho mỗi mục.
- Xếp hạng và chọn n mục hàng đầu để đề xuất.



# User-Based Collaborative Filtering

	$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$
$i_0$	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

$\downarrow \quad \downarrow \quad | \quad \downarrow \quad \downarrow \quad | \quad \downarrow \quad \downarrow \quad \downarrow$

$\bar{u}_j$	3.25	2.75	2.5	1.33	2.5	1.5	3.33
-------------	------	------	-----	------	-----	-----	------

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

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

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

	$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$
$i_0$	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  $\bar{\mathbf{Y}}$ .

# User-Based Collaborative Filtering

	$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$
$u_0$	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 S.

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

- Tính toán độ tương đồng: Sử dụng hàm **Cosine Similarity**.

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

- Tạo ma trận tương đồng người dùng (**User similarity matrix**).

# User-Based Collaborative Filtering

	$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$
$i_0$	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)  $\hat{Y}$

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

- Xác định các users đã đánh giá cho item.
- Lấy users tương đồng nhất.
- 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
- $\bar{y}_{i,u_j}$ : 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  $u_j$

# User-Based Collaborative Filtering

## Bước 4: Dự đoán các ratings còn thiếu: (Ví Dụ)

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:  $N(u_1, i_1) = \{u_0, u_5\}$

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

	$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$
$u_0$	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

	$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$
$i_0$	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

# User-Based Collaborative Filtering

Movie recommended for User 50:

Item 277 -> Predicted Rating: 0.54

Item 2224 -> Predicted Rating: 0.45

Item 841 -> Predicted Rating: 0.44

Item 602 -> Predicted Rating: 0.42

Item 913 -> Predicted Rating: 0.42

Movie recommended for User 50:

Item 277 -> Predicted Rating: 3.54

Item 2224 -> Predicted Rating: 3.45

Item 841 -> Predicted Rating: 3.45

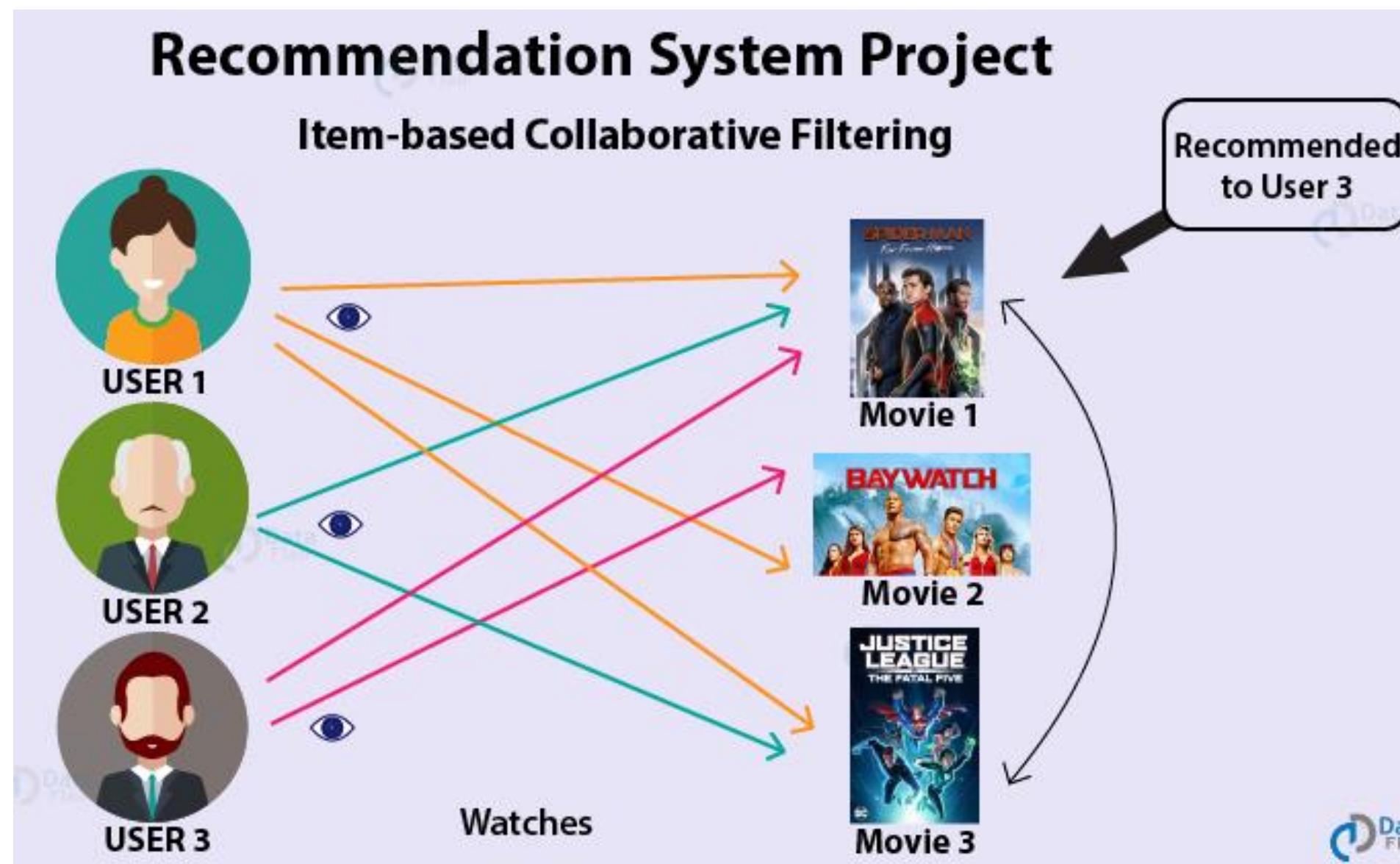
Item 602 -> Predicted Rating: 3.43

Item 913 -> Predicted Rating: 3.43

## Bước 5: Đề xuất items cho người dùng:

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

# Collaborative Filtering



- **Item-Based Collaborating Filtering:** phương pháp này xem xét các item đã được user đánh giá và xác định các item có xu hướng sẽ được đánh giá tương tự. Từ đó, gợi ý những items gần giống với những items mà user có mức độ quan tâm cao.

# Item-Based Collaborative Filtering

	$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$	
$i_0$	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.

**Bước 1: Tạo ma trận User-Item (Utility Matrix) (hình a)**  
**Bước 2: Chuẩn hóa điểm đánh giá: (hình b)**

	$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$	
$i_0$	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  $\bar{\mathbf{Y}}$ .

# Item-Based Collaborative Filtering

	$i_0$	$i_1$	$i_2$	$i_3$	$i_4$
$i_0$	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

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

- Tính toán độ tương đồng: Sử dụng hàm **Cosine Similarity**.

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

- Tạo ma trận tương đồng (**Item similarity matrix**).

# Item-Based Collaborative Filtering

	$u_0$	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$
$i_0$	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  $\bar{Y}$ .

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

- 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

# Item-Based Collaborative Filtering

MoviesID 5 recommended for User:

User 174 -> Predicted Rating: 0.94

User 52 -> Predicted Rating: 0.80

User 577 -> Predicted Rating: 0.79

User 546 -> Predicted Rating: 0.75

User 91 -> Predicted Rating: 0.75

MoviesID 5 recommended for User:

User 174 -> Predicted Rating: 4.34

User 52 -> Predicted Rating: 4.21

User 577 -> Predicted Rating: 4.20

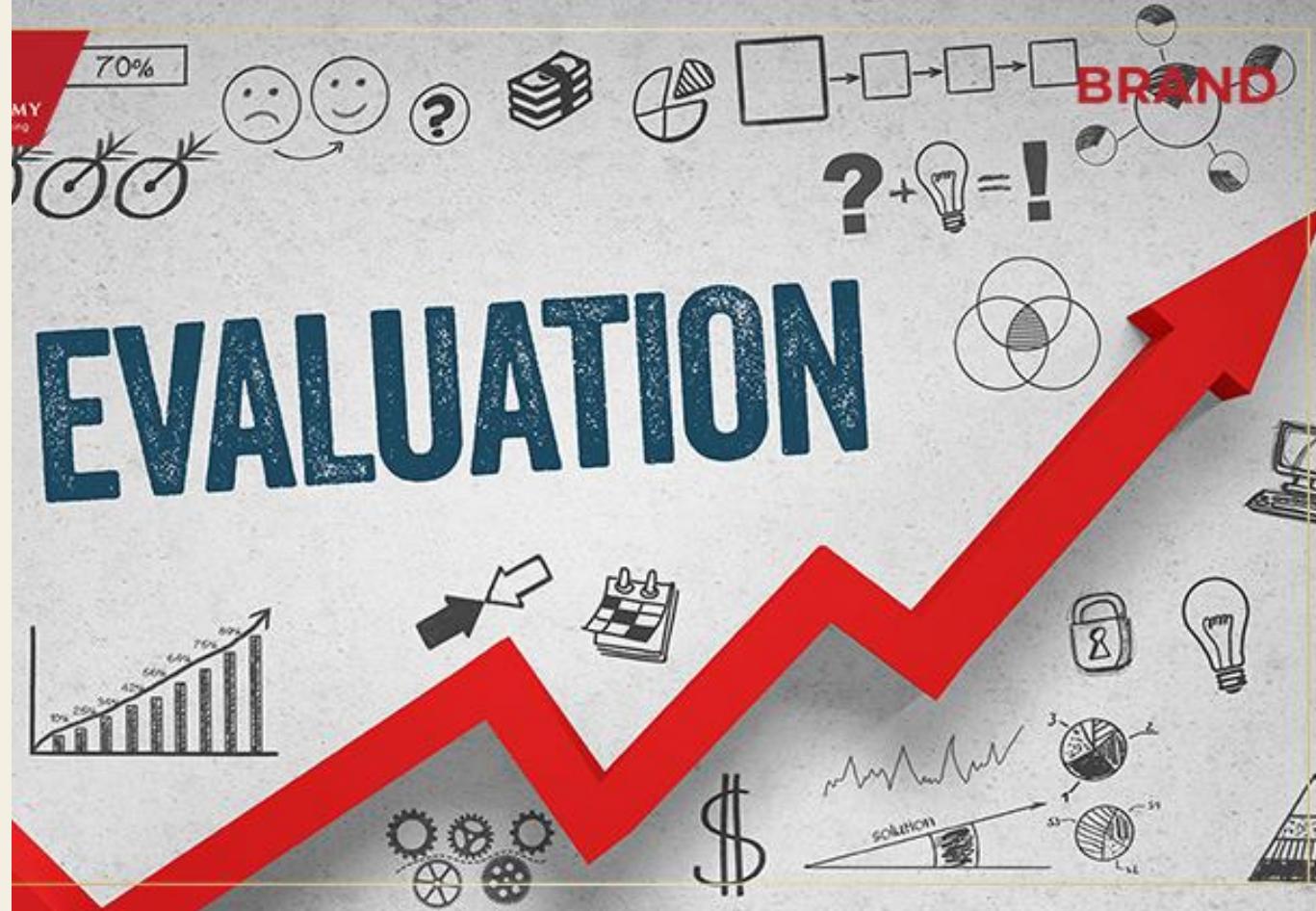
User 546 -> Predicted Rating: 4.16

User 91 -> Predicted Rating: 4.15

## 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: hệ thống gợi ý những items gần giống với những items mà user có mức độ quan tâm cao

# Đánh Giá Mô Hình Collaborative Filtering



- Dữ liệu được chia thành tập train và test theo tỷ lệ 80/20.
- Huấn luyện mô hình
- Sử dụng dữ liệu test để dự đoán điểm đánh giá.
- Tính RMSE (Root Mean Squared Error) để đo lường hiệu quả của mô hình.

RMSE (User-based CF): 0.1021  
RMSE (Item-based CF): 0.3893





**Ưu nhược điểm  
của các mô hình**

**CHƯƠNG  
04**

Mô hình	Ưu điểm	Nhược điểm
Content-based Filtering	<ul style="list-style-type: none"> <li>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</li> <li>Đơn giản</li> </ul>	<ul style="list-style-type: none"> <li>Cần nhiều dữ liệu về các mô tả phim, đạo diễn, diễn viên, .... Phụ thuộc vào việc phân tích nội dung chính xác</li> <li>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.</li> </ul>
User-user Collaborative Filtering	<ul style="list-style-type: none"> <li>Có tính cá nhân hóa cao</li> <li>Phù hợp với những hệ thống lớn có nhiều đánh giá từ người dùng.</li> <li>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.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> <li>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.</li> <li>Không thể gợi ý được những sản phẩm mới hoặc những sản phẩm chưa được ai đánh giá.</li> </ul>
Item-item Collaborative Filtering	<ul style="list-style-type: none"> <li>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</li> <li>Hiệu quả với các tập dữ liệu lớn.</li> <li>Dễ mở rộng và phù hợp với các hệ thống có số lượng users lớn hơn số lượng items.</li> </ul>	<ul style="list-style-type: none"> <li>Ít cá nhân hóa hơn, vì tập trung vào tương đồng giữa items thay vì từng user cụ thể.</li> <li>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.</li> </ul>

# Kết Luận

- Trong dữ liệu này, việc chỉ dựa vào thông tin nội dung không đủ mạnh để phân biệt sở thích phức tạp giữa các người dùng.
- Bên cạnh đó, với tập dữ liệu này, có thể thấy các người dùng có xu hướng đánh giá các bộ phim giống nhau, làm cho user-based CF rất hiệu quả.
- Mặc dù item-based CF hoạt động tốt, nhưng với số lượng items lớn hơn rất nhiều (9743 phim) so với số lượng users (610), dữ liệu thừa và nhiều phim chỉ có ít lượt đánh giá làm giảm độ chính xác khi tính tương đồng giữa các items.

**User-based CF là mô hình tốt nhất cho dữ liệu này, với hướng Content-based cần được cải thiện đặc trưng nội dung. Item-based CF có thể cải thiện nếu kết hợp trong một hệ thống hybrid hoặc với dữ liệu phong phú hơn.**

**THANK YOU!**