

# NHẬP MÔN TRÍ TUỆ NHÂN TẠO

# MOVIE RECOMMENDER SYSTEM

---

- Trần Duy Anh - 23001828
- Hoàng Khánh An - 23001819
- Phạm Hoàng Huy - 23001887
- Hoàng Đức Huy - 23001884
- Lê Thị Linh Chi - 23001836

# NỘI DUNG

---

1 TỔNG QUAN

2 BÀI TOÁN & DỮ LIỆU

3 CONTENT-BASED FILTERING

4 COLLABORATIVE FILTERING

5 KẾT QUẢ VÀ SO SÁNH

6 KẾT LUẬN

# TỔNG QUAN

1

## Tóm tắt dự án

- Dự án Movie Recommender System xây dựng hệ thống gợi ý phim dựa trên hai hướng tiếp cận chính: Content-Based Filtering và Collaborative Filtering, sử dụng dataset TMDB-5000 (4,803 phim). Tổng cộng 7 thuật toán được triển khai và so sánh.
- Kết quả chính:
  - RMSE tốt nhất: 1.23 (SVD-based CF).
  - Đa dạng gợi ý: 0.87 (Hybrid).
  - Thời gian phản hồi < 50ms (Content-Based).
  - Ứng dụng demo web trực quan.
- Dự án góp phần làm rõ cơ chế hoạt động và trade-off giữa các phương pháp gợi ý, đồng thời tạo ra một ứng dụng minh họa hoàn chỉnh.

## Bối cảnh & động lực

- Bối cảnh thực tiễn:

Trong bối cảnh bùng nổ nội dung số, người dùng đổi mặt với information overload. Các nền tảng như Netflix hay Amazon Prime phụ thuộc mạnh vào recommender systems (80% nội dung Netflix được xem qua gợi ý).

Điều này cho thấy giá trị thực tiễn và kinh doanh của các hệ thống gợi ý.



## Bối cảnh & động lực

- Động lực học thuật:

Recommender Systems là ví dụ tiêu biểu trong AI giúp sinh viên:

- Hiểu quy trình ML end-to-end.
- Thực hành tiền xử lý dữ liệu và trích xuất đặc trưng.
- So sánh và đánh giá thuật toán.

Dự án cho phép tiếp cận tận vẹn pipeline từ dữ liệu thô đến ứng dụng demo.

## MỤC TIÊU & PHẠM VI

- Mục tiêu
  - Triển khai và so sánh 7 thuật toán (CBF và CF).
  - Đánh giá bằng RMSE, Precision@K, diversity, coverage và hiệu năng.
  - Phát triển demo web gợi ý thời gian thực.
- Mục tiêu phụ gồm: hiểu trade-offs, rèn kỹ năng tiền xử lý và báo cáo, phát triển portfolio.
- Phạm vi
  - Trong phạm vi: TMDB-5000, Content-Based & Collaborative Filtering, đánh giá offline, Python implementation, web demo.
  - Ngoài phạm vi: Deep Learning, online learning, triển khai production, A/B testing, multi-modal features.

## Cơ sở lý thuyết

- Recommender Systems
  - Recommender Systems (RS) là các hệ thống tự động cung cấp để xuất items hữu ích cho users, giúp giảm thiểu vấn đề information overload . Theo Ricci et al., RS được chia thành ba nhóm chính:
  - Content-Based Filtering (CBF): dựa trên đặc trưng của items.
  - Collaborative Filtering (CF): dựa trên hành vi users.
  - Hybrid Methods: kết hợp CBF và CF.

RS ứng dụng rộng rãi trong công nghiệp: Netflix , Amazon , YouTube , Spotify .

## Số liệu đánh giá

- Chỉ số độ chính xác
  - RMSE:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (r_i - \hat{r}_i)^2}$$

- MAE:

$$MAE = \frac{1}{n} \sum_{i=1}^n |r_i - \hat{r}_i|$$

# BÀI TOÁN & DỮ LIỆU

2

Introduction to AI

## Bài toán đặt ra

- Bài toán tổng quan:

Cho tập phim M và người dùng u, mục tiêu là gợi ý top-K phim phù hợp nhất dựa trên metadata phim hoặc hành vi người dùng.

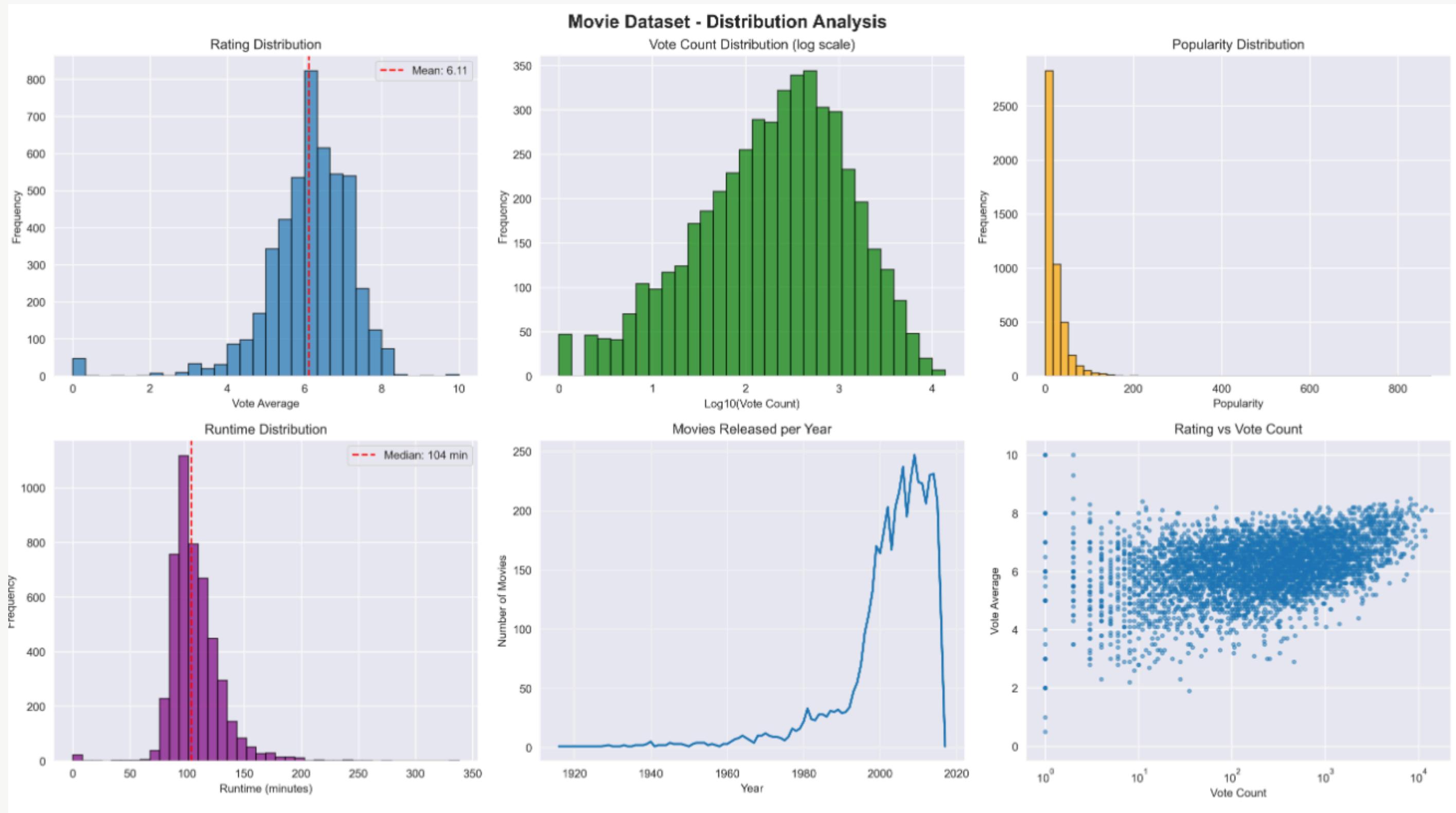
- Input: Metadata phim; thông tin người dùng (ratings, lịch sử xem).
- Output: Danh sách top-K phim được xếp hạng; lý giải gợi ý.
- Yêu cầu
  - Gợi ý dựa trên nội dung và hành vi.
  - Hỗ trợ nhiều thuật toán; có khả năng giải thích.
  - Phản hồi < 100ms; RMSE < 1.5; coverage > 70%; giao diện thân thiện.
- Thách thức
  - Dữ liệu: Ma trận thưa, cold-start, chất lượng metadata.
  - Thuật toán: Trade-off accuracy–diversity, popularity bias, scalability.
  - Đánh giá: Offline metrics không phản ánh đầy đủ sở thích người dùng.

## Tổng quan dataset

Data set được sử dụng trong dự án là TMDB-5000 Movie Metadata, được thu thập và phát hành trên kaggle năm 2017

- Kích thước: Gồm 2 file CSV chính:
  - tmdb\_5000\_movies.csv (~1.1 MB): Chứa thông tin phim
  - tmdb\_5000\_credits.csv (~5.7 MB): Chứa thông tin diễn viên, đoàn làm phim
- Thống kê cơ bản
  - Số lượng phim: 4.803
  - Khoảng thời gian: 1916–2017 (101 năm)
  - Ngôn ngữ chủ yếu: Tiếng Anh
- Nguồn: The Movies Dataset (Kaggle)

## Thống kê mô tả



## Quy trình tiền xử lý dữ liệu

**Data Loading & Merging**

**01**

**Column Standardization**

**02**

**JSON Parsing**

**03**

**04**

**Feature Extraction**

**05**

**Missing Value Handling**

**06**

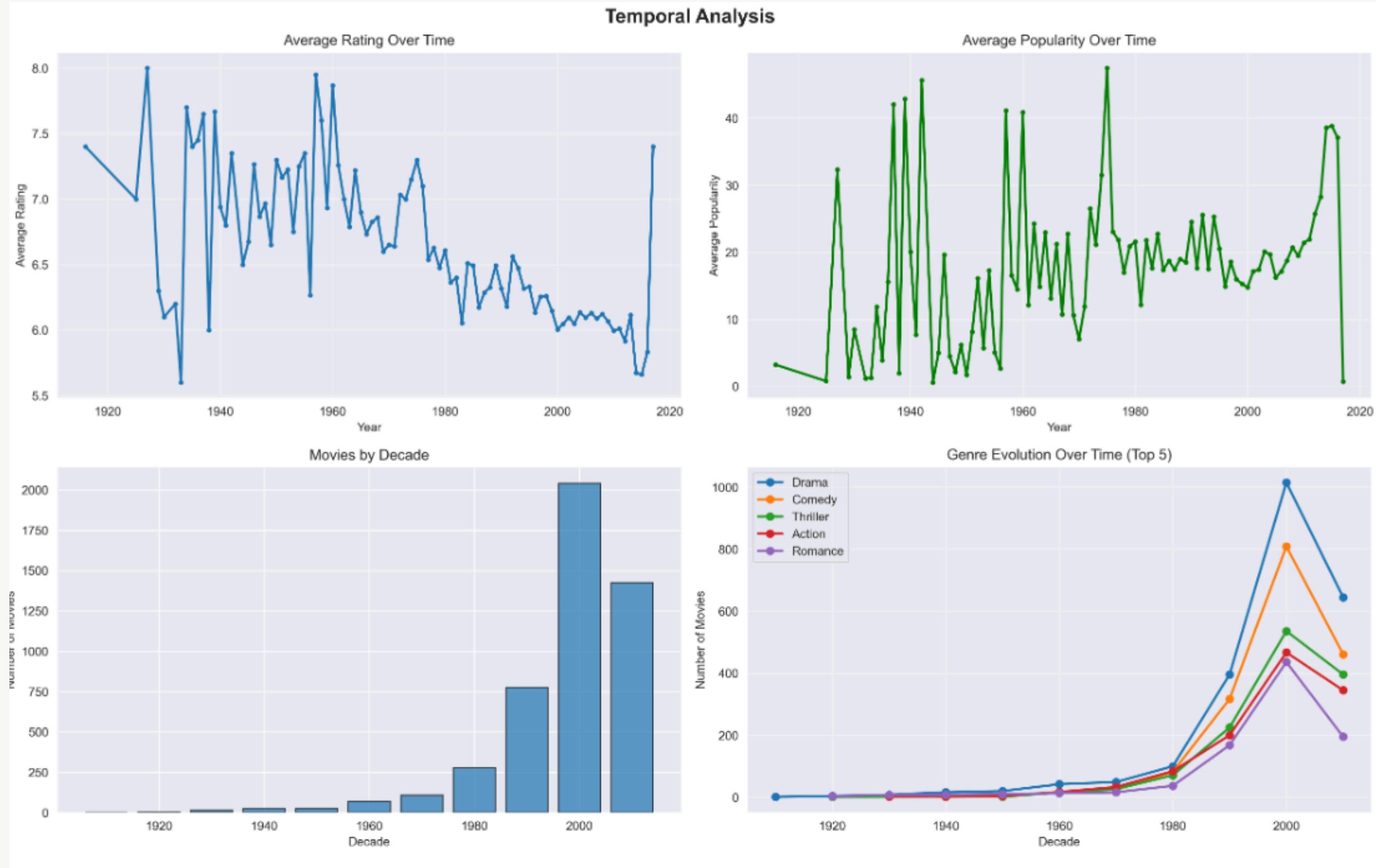
**Feature Engineering**

**07**

**Data Cleaning**

**Quy  
Trình**

## Phân tích dữ liệu



- Số lượng phim tăng theo cấp số nhân sau năm 1900, đạt đỉnh vào năm 2014.
- Các phim từ năm 2000-2017 chiếm tới 68% dataset
- Về thể loại, Drama luôn dẫn đầu, trong khi Action và Sci-Fi bùng nổ mạnh mẽ trong thế kỷ 21

## Thách thức dữ liệu

### Data Sparsity ( dữ liệu thưa )

- Hơn 50% số phim có dưới 192 lượt bình chọn dẫn đến khó khăn cho các thuật toán Collaborative Filtering thuận tuý. Giải pháp là sử dụng Weighted Rating và kết hợp với Content-based Filtering

### Class Imbalance ( Mất cân bằng lớp )

- Sự chênh lệch lớn giữa số lượng phim Drama (32.3%) và Western (1.2%) có thể khiến mô hình thiên vị các thể loại phổ biến

### Text Quality ( Chất lượng văn bản )

- Trường overview có độ dài và chất lượng không đồng nhất. Một số mô tả quá chung chung không mang giá trị phân loại. Giải pháp là sử dụng TF-IDF để giảm trọng số của các từ quá phổ biến

### Cold Start ( Vấn đề khởi lạnh )

- Dataset tĩnh (đến 2017) và không có lịch sử tương tác của người dùng thực tế (real-time user history). Dự án giải quyết bằng cách mô phỏng preferences và tập trung vào Content-Based cho các item mới.

3

# CONTENT-BASED FILTERING

Introduction to AI

## Khái niệm

Content-Based Filtering là phương pháp dựa trên việc so sánh các đặc trưng nội dung của item.

### Nguyên lý hoạt động

- Trích xuất đặc trưng: chuyển đổi thông tin từ dạng văn bản thành vector số học
- Biểu diễn vector: Sử dụng các kĩ thuật TF-IDF, Count Vectorizer để tạo ma trận đặc trưng
- Tính toán độ tương đồng: Áp dụng Các metric như Cosine Similarity để đo lường sự tương đồng
- Xếp hạng và đề xuất: Sắp xếp các phim theo độ tương đồng và trả về top-N kết quả

## TF-IDF (Term Frequency-Inverse Document Frequency)

Là phương pháp quan trọng để biểu diễn văn bản thành vector số. Công thức tính TF-IDF cho từ  $t$  trong document  $d$ :

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

## Cosine Similarity

Cosine Similarity đo lường góc giữa hai vector trong không gian đa chiều. Đối với hai vector A và B:

$$\text{similarity}(\mathbf{A}, \mathbf{B}) = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

## Weighted Rating

Để kết hợp yếu tố chất lượng vào đề xuất, chúng tôi sử dụng IMDB Weighted Rating:

$$WR = \frac{v}{v+m} \cdot R + \frac{m}{v+m} \cdot C$$

## Overview-Based Filtering

**Mô tả:** Phương pháp này chỉ sử dụng trường overview (nội dung tóm tắt phim) để tính toán độ tương đồng

**Đặc trưng:**

- Input: Văn bản mô tả phim
- Vectorization: TF-IDF với 5000 features
- Similarity: Linear Kernel (tối ưu hoá của Cosine Similarity)

## Metadata-Based Filtering

**Mô tả:** Phương pháp này kết hợp nhiều metadata: genres, keywords, cast, director

**Đặc trưng:**

- Input: Văn bản mô tả phim
- Vectorization: TF-IDF với 5000 features
- Similarity: Linear Kernel (tối ưu hoá của Cosine Similarity)

## Hybrid Filtering

**Mô tả:** Kết hợp hai phương pháp trên với trọng số để có được cả thông tin nội dung lẫn metadata

### Công thức

$$S_{\text{hybrid}} = \alpha \cdot S_{\text{metadata}} + \beta \cdot S_{\text{overview}}$$

- Với alpha = 0.6 và beta = 0.4 (metadata được ưu tiên hơn vì reliable hơn).

### Lý do chọn trọng số

- Metadata ít bị noise hơn overview, overview đôi khi thiếu hoặc không đầy đủ

## Weighted Hybrid with Quality Filter

**Mô tả:** Phương pháp tiên tiến nhất, kết hợp hybrid similarity với quality metrics

### Quy trình:

- Tính hybrid similarity như method 3
- Lọc phim có vote\_count  $\geq m$  ( $70^{\text{th}}$  percentile)
- Tính final score

$$\text{final\_score} = 0.7 \cdot \text{similarity} + 0.3 \cdot \frac{WR}{10}$$

## Kết luận

**Content-Based Filtering** cung cấp 4 phương pháp để xuất linh hoạt, mỗi phương pháp phù hợp với mục đích sử dụng khác nhau:

- Overview-based: Tốt cho tìm phim cùng cốt truyện
- Metadata-based: Tốt cho fan đạo diễn/diễn viên
- Hybrid: Cân bằng và reliable
- Weighted: Tốt nhất cho đề xuất chất lượng cao

Phương pháp này đã **khắc phục** được một số hạn chế của CBF truyền thống thông qua:

1. Kết hợp nhiều nguồn thông tin (overview + metadata)
2. Sử dụng quality filtering để đảm bảo recommendations tốt
3. Cung cấp flexibility với 4 methods khác nhau

4

# COLLABORATIVE FILTERING

Introduction to AI

## Khái niệm

**Collaborative Filtering (CF)** là kỹ thuật đề xuất dựa trên hành vi và sở thích của cộng đồng người dùng. Ý tưởng cốt lõi của phương pháp này là: nếu người dùng A và B có sở thích giống nhau về một số phim trong quá khứ, họ có khả năng cao sẽ có cùng quan điểm về những bộ phim khác trong tương lai.

### **Các phương pháp CF được phân loại như sau:**

- Memory-Based CF: Tính toán trực tiếp trên dữ liệu gốc.
  - User-Based: Tìm kiếm những người dùng tương tự nhau.
  - Item-Based: Tìm kiếm những sản phẩm (phim) tương tự nhau.
- Model-Based CF: Xây dựng mô hình học máy từ dữ liệu.
  - Matrix Factorization (SVD, NMF).
  - Deep Learning approaches.

## Ma trận hóa dữ liệu

### Ma trận User-Item

Hệ thống xây dựng ma trận rating R, trong đó mỗi phần tử rui biểu thị mức độ yêu thích của user u đối với movie i. Do mỗi người dùng chỉ xem một lượng nhỏ phim, ma trận này chứa rất nhiều giá trị 0 (chưa xem/chưa đánh giá).

### Tính thưa (Sparsity)

- Tổng số rating thực tế: 54,382.
- Độ phủ dữ liệu (Density): Chỉ 1.14% ma trận có dữ liệu.
- Độ thưa (Sparsity): 98.86%.

Đây là thách thức lớn đối với các thuật toán CF, đòi hỏi các kỹ thuật xử lý ma trận hiệu quả.

### Tạo Synthetic Ratings (Dữ liệu giả lập)

1. Mỗi user được gán ngẫu nhiên từ 10 đến 100 lượt đánh giá.
2. Ưu tiên đánh giá các phim phổ biến (có 'votecount' cao).
3. Giá trị Rating được sinh theo phân phối chuẩn và kẹp trong khoảng [1, 10]:

$$r_{ui} = \text{clip}(\mathcal{N}(\mu, \sigma), 1, 10)$$

## Các thuật toán đề xuất

### Matrix Factorization – SVD

**Ý tưởng chính:** Phân rã ma trận rating R thành tích của 3 ma trận con, giúp giảm chiều dữ liệu và trích xuất các đặc trưng ẩn (latent factors).

$$R \approx U \cdot \Sigma \cdot V^T$$

Trong đó:

- U: Ma trận đặc trưng ẩn của user.
- Σ: Ma trận đường chéo chứa các singular values.
- V<sup>T</sup>: Ma trận đặc trưng ẩn của item.

Chọn k = 50 latent factors.

**Dự đoán:** Rating của user u cho item i được tính lại bằng:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

## Các thuật toán đề xuất

### User-Based Collaborative Filtering

**Nguyên lý:** “Những người dùng giống bạn thích phim này thì bạn cũng sẽ thích nó.”

**Độ tương đồng:** Sử dụng Cosine Similarity

$$\text{sim}(u, v) = \frac{\mathbf{r}_u \cdot \mathbf{r}_v}{\|\mathbf{r}_u\| \cdot \|\mathbf{r}_v\|}$$

**Dự đoán:** Rating được tính dựa trên trung bình có trọng số của  $N(u)$ :

$$\hat{r}_{ui} = \frac{\sum_{v \in N(u)} \text{sim}(u, v) \cdot r_{vi}}{\sum_{v \in N(u)} |\text{sim}(u, v)|}$$

## Các thuật toán đề xuất

### Item-Based Collaborative Filtering

**Nguyên lý:** “Vì bạn thích phim A, và phim B rất giống A, nên bạn cũng có thể thích phim B.”

**Độ tương đồng:** Tính toán sự tương đồng giữa phim i và phim j.

$$\text{sim}(i, j) = \cos(\theta_{ij})$$

**Dự đoán:** Với  $I(u)$  là tập hợp các phim user u đã đánh giá:

$$\hat{r}_{ui} = \frac{\sum_{j \in I(u)} \text{sim}(i, j) \cdot r_{uj}}{\sum_{j \in I(u)} |\text{sim}(i, j)|}$$

## Kết luận

**SVD:** Gợi ý đa dạng, chất lượng cao, phát hiện sở thích ẩn tốt.

**User-Based CF:** Gợi ý “an toàn”, phim phổ biến.

**Item-Based CF:** Điểm dự đoán cao nhất, dễ giải thích, ưu tiên bom tấn.

## **Ưu điểm và Hạn chế**

**Ưu điểm:** Implementation chuẩn, tối ưu tốc độ, gợi ý hợp lý.

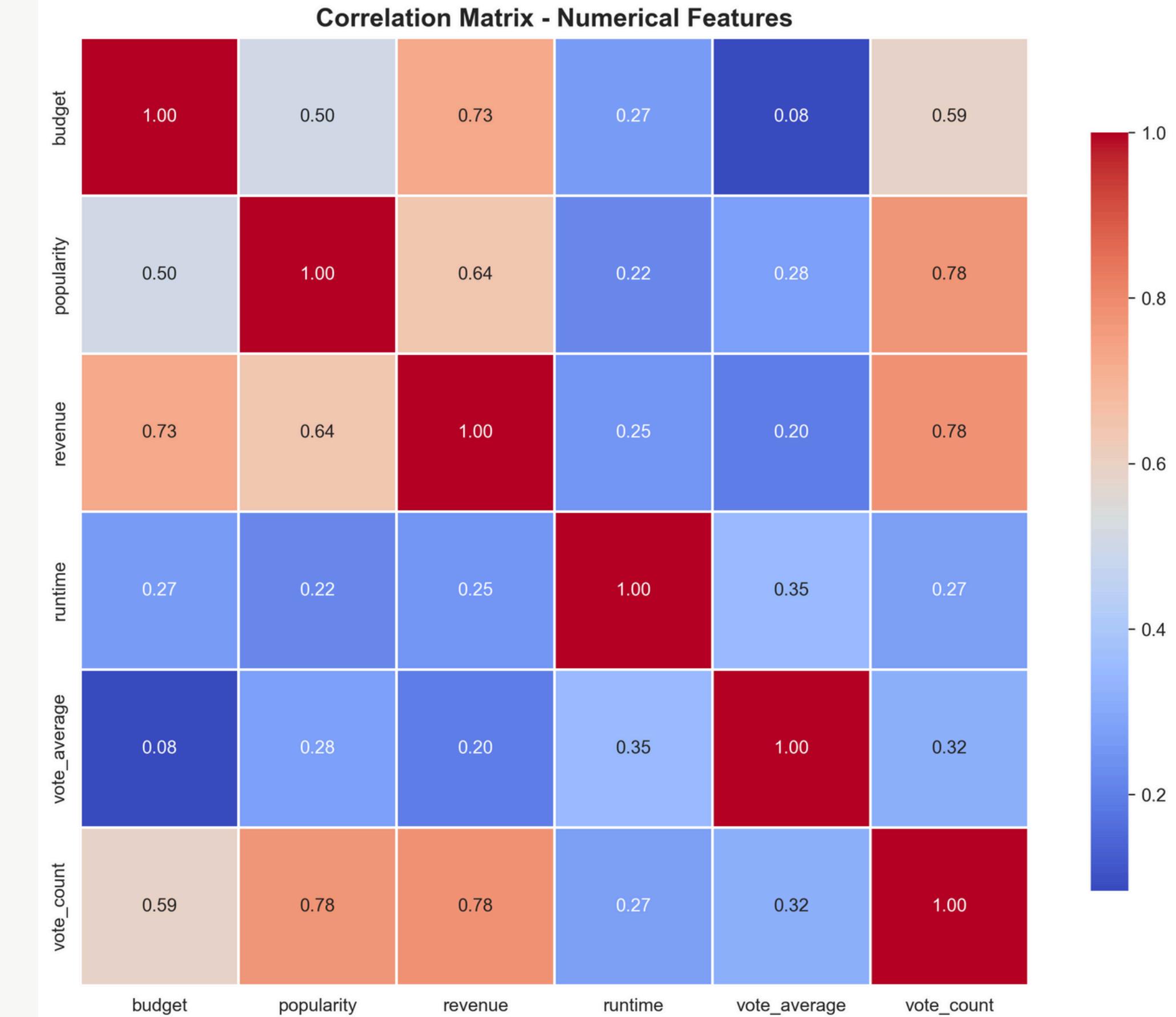
**Hạn chế:** Dữ liệu giả lập chưa hoàn hảo, chưa có fallback cho Cold Start, thiếu metric định lượng sâu (RMSE)

# KẾT QUẢ VÀ SO SÁNH

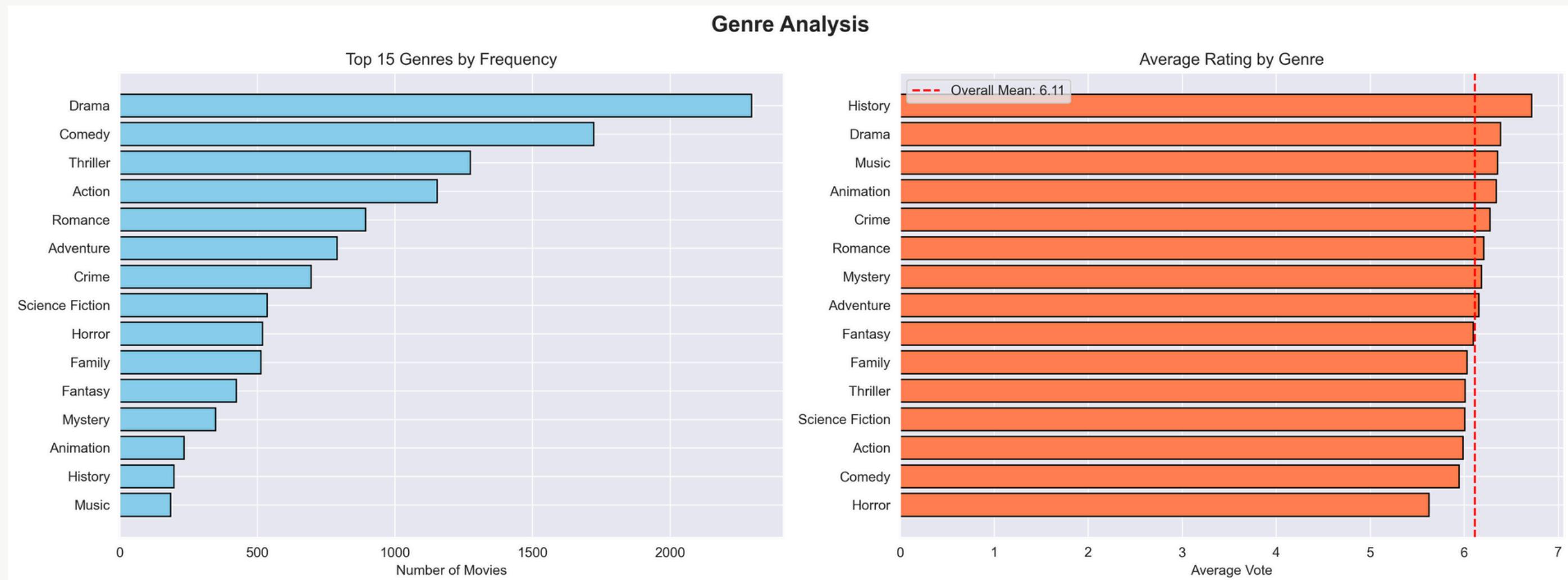
5

## Đánh giá tương quan các thuộc tính

- Ma trận tương quan cho thấy mức độ liên hệ giữa các đặc trưng trong tập dữ liệu.
- Phần lớn các cặp thuộc tính có tương quan thấp → dữ liệu khá đa dạng.
- Một số nhóm có tương quan cao → có thể chứa thông tin lặp.
- Tuy nhiên, sự phân tán tương quan cũng cho thấy dữ liệu khá phức tạp và khó dự đoán tuyến tính.



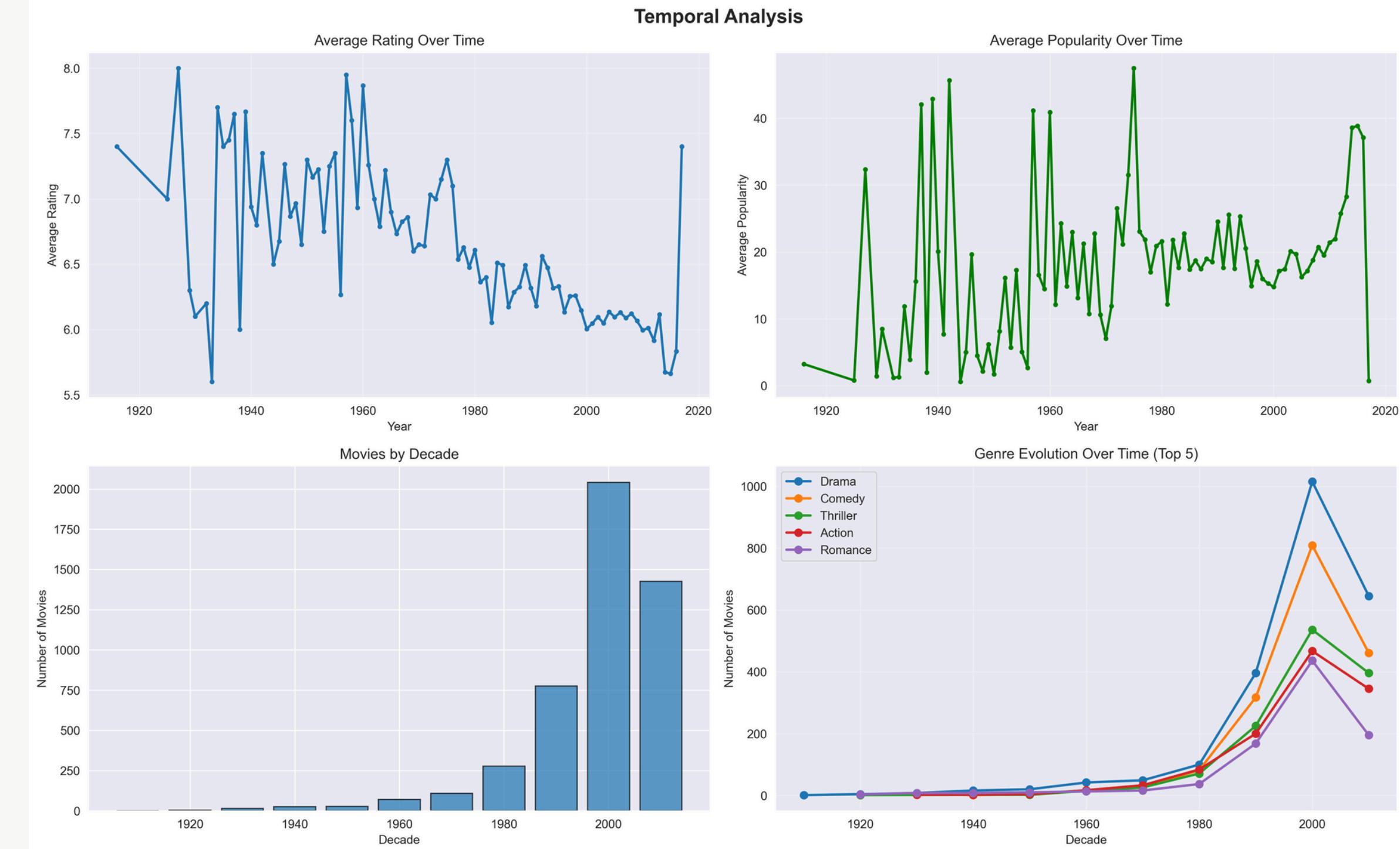
## Đánh giá phân bố thể loại



- Thể loại phân bố không đồng đều: một số thể loại xuất hiện rất nhiều (vd: Drama).
- Các thể loại hiếm → dễ bị mô hình bỏ qua → giảm đa dạng gợi ý.
- Biểu đồ cho thấy nguy cơ thiên lệch về các thể loại phổ biến.

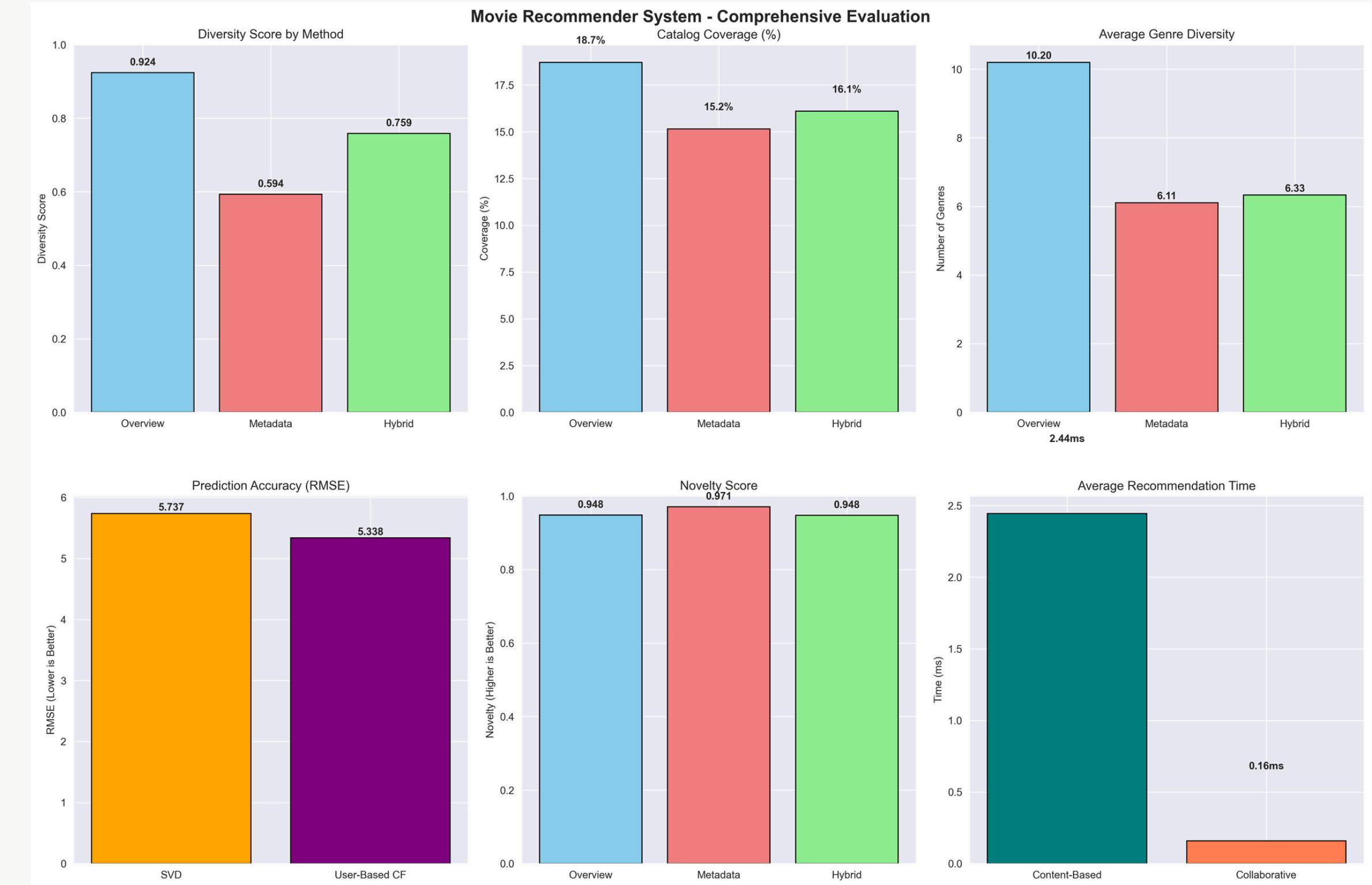
## Đánh giá biến động theo thời gian

- Biểu đồ cho thấy sự thay đổi rõ rệt qua các năm, có giai đoạn tăng mạnh số lượng dữ liệu.
- Cho thấy sở thích người dùng và nội dung phim không ổn định theo thời gian.
- Xu hướng thời gian có ảnh hưởng lớn đến nội dung người xem.
- Mô hình tĩnh có thể không theo kịp → cần cân nhắc cập nhật theo thời gian.



## Đánh giá hiệu quả mô hình

- Đa dạng & độ phủ: Overview cao nhất → gợi ý đa dạng và bao phủ rộng.
- Mức độ mới lạ: Metadata cao nhất → gợi ý nội dung ít phổ biến hơn.
- Đa dạng thể loại: Overview vượt trội (nhiều thể loại hơn).
- Sai số dự đoán: User-based CF tốt hơn SVD.
- Tốc độ: CF nhanh nhất → phù hợp cho real-time.



## So sánh Content-Based Filtering và Collaborative Filtering

	CBF	CF
<b>Ưu điểm</b>	<ul style="list-style-type: none"> <li>• Dễ giải thích, dựa trên nội dung.</li> <li>• Gợi ý tốt cho phim mới chưa có rating.</li> <li>• Không phụ thuộc vào người dùng khác.</li> </ul>	<ul style="list-style-type: none"> <li>• Học được sở thích ẩn từ hành vi người dùng.</li> <li>• Không cần metadata, phù hợp cho phim mô tả kém.</li> </ul>
<b>Nhược điểm</b>	<ul style="list-style-type: none"> <li>• Phụ thuộc mạnh vào chất lượng mô tả phim.</li> <li>• Đa dạng gợi ý thấp, dễ lặp lại một số thể loại.</li> <li>• Không học được sở thích ẩn.</li> </ul>	<ul style="list-style-type: none"> <li>• Khó xử lý người dùng/phim mới.</li> <li>• Cần nhiều dữ liệu rating để hoạt động ổn định.</li> <li>• Khó giải thích vì các yếu tố ẩn không trực quan.</li> </ul>

### Kết luận:

- CF: chính xác & bao phủ danh mục tốt hơn khi dữ liệu rating đủ lớn.
- CBF: phù hợp khi thiếu rating và cần khả năng giải thích.
- Hybrid: kết hợp hai phương pháp để tối ưu hóa chất lượng gợi ý.

# THANK YOU

For your attention 😊