

ĐẠI HỌC BÁCH KHOA HÀ NỘI  
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN  
THÔNG

BÀI TẬP LỚN

MÔN: NHẬP MÔN HỌC MÁY VÀ KHAI PHÁ DỮ  
LIỆU

Đề tài: Hệ thống gợi ý phim sử dụng các  
phương pháp học máy



SOICT

Giáo viên hướng dẫn: PGS. TS. Thân Quang Khoát

Danh sách sinh viên thực hiện:

| STT | Họ và tên           | Mã sinh viên |
|-----|---------------------|--------------|
| 1   | Nguyễn Quốc Thái    | 20225083     |
| 2   | Nguyễn Hoàng Phương | 20225070     |
| 3   | Phạm Hùng Phong     | 20225060     |

# Mục lục

|                                                                                |           |
|--------------------------------------------------------------------------------|-----------|
| <b>Lời Mở Đầu</b>                                                              | <b>4</b>  |
| <b>1 Giới thiệu bài toán</b>                                                   | <b>5</b>  |
| <b>2 Thu thập và chuẩn bị dữ liệu</b>                                          | <b>5</b>  |
| <b>3 Phương pháp lọc dựa trên nội dung</b>                                     | <b>5</b>  |
| 3.1 Giới thiệu về Content-Based Recommendation . . . . .                       | 5         |
| 3.2 Dữ liệu và Tiền xử lý . . . . .                                            | 6         |
| 3.2.1 Tập dữ liệu sử dụng . . . . .                                            | 6         |
| 3.2.2 Quá trình tiền xử lý dữ liệu . . . . .                                   | 6         |
| 3.3 Phương pháp . . . . .                                                      | 7         |
| 3.3.1 TF-IDF và Cosine Similarity . . . . .                                    | 7         |
| 3.3.2 Word2Vec và Mô hình học sâu . . . . .                                    | 8         |
| 3.4 Kết quả và Dánh giá . . . . .                                              | 9         |
| 3.4.1 Các phương pháp thử nghiệm . . . . .                                     | 10        |
| 3.4.2 Kết quả Precision@K . . . . .                                            | 10        |
| 3.4.3 Kết quả NDCG và HR . . . . .                                             | 11        |
| 3.4.4 Dánh giá và Phân tích . . . . .                                          | 11        |
| <b>4 Mô hình Neural Collaborative Filtering (NCF)</b>                          | <b>12</b> |
| 4.1 Giới thiệu về Neural Collaborative Filtering (NCF) . . . . .               | 12        |
| 4.2 Generalized Matrix Factorization . . . . .                                 | 13        |
| 4.3 Multi-Layer Perceptron . . . . .                                           | 13        |
| 4.4 Tạo mô hình NCF bằng cách kết hợp GMF và MLP . . . . .                     | 14        |
| 4.5 Hàm mục tiêu và Phương pháp tối ưu . . . . .                               | 15        |
| 4.6 Dánh giá . . . . .                                                         | 15        |
| 4.6.1 Phương pháp đánh giá . . . . .                                           | 15        |
| 4.6.2 Các tham số trong mô hình NCF . . . . .                                  | 16        |
| 4.6.3 Kết quả thử nghiệm các tham số . . . . .                                 | 16        |
| 4.6.4 Kết luận . . . . .                                                       | 18        |
| 4.7 Dưa ra gợi ý với người dùng mới . . . . .                                  | 18        |
| 4.8 Weighted Pooling cho người dùng mới . . . . .                              | 18        |
| 4.9 Kết quả . . . . .                                                          | 19        |
| <b>5 User-based Collaborative Filtering sử dụng KNN</b>                        | <b>19</b> |
| 5.1 Giới thiệu về User-based Collaborative Filtering sử dụng KNN . . . . .     | 19        |
| 5.2 Công thức tính toán tương đồng bằng Cosine Similarity . . . . .            | 20        |
| 5.3 Quy trình áp dụng . . . . .                                                | 20        |
| 5.4 Dánh giá thuật toán KNN . . . . .                                          | 21        |
| 5.4.1 Phương pháp đánh giá . . . . .                                           | 21        |
| 5.4.2 Các tham số trong User-based Collaborative Filtering using KNN . . . . . | 22        |
| 5.4.3 Kết quả thử nghiệm các tham số . . . . .                                 | 22        |
| 5.5 Dưa ra gợi ý với người dùng mới . . . . .                                  | 22        |
| <b>6 Approximate Nearest Neighbor (ANN) trong bài toán hệ thống gợi ý phim</b> | <b>23</b> |
| 6.1 Nguyên lý hoạt động . . . . .                                              | 23        |
| 6.2 Giới thiệu về FAISS . . . . .                                              | 24        |
| 6.3 Quy trình áp dụng FAISS để tìm Top n-Phim gợi ý . . . . .                  | 24        |

|          |                                                                                                                                |           |
|----------|--------------------------------------------------------------------------------------------------------------------------------|-----------|
| 6.4      | Dánh giá . . . . .                                                                                                             | 25        |
| 6.4.1    | Phương pháp đánh giá . . . . .                                                                                                 | 25        |
| 6.4.2    | Các tham số trong User-based Collaborative Filtering using FAISS . . . . .                                                     | 26        |
| 6.4.3    | Kết quả thử nghiệm các tham số . . . . .                                                                                       | 26        |
| 6.5      | Dựa ra gợi ý với người dùng mới . . . . .                                                                                      | 27        |
| <b>7</b> | <b>Thử nghiệm hệ thống trên các kịch bản khác nhau</b>                                                                         | <b>28</b> |
| 7.1      | Mô tả các kịch bản thử nghiệm . . . . .                                                                                        | 28        |
| 7.2      | Quy trình thử nghiệm . . . . .                                                                                                 | 28        |
| 7.3      | Kết quả thử nghiệm với mô hình Content-based Filtering với phương pháp Same Metadata Only và Metadata-Based (SMO-MB) . . . . . | 28        |
| 7.4      | Kết quả thử nghiệm với mô hình NeuMF . . . . .                                                                                 | 30        |
| 7.5      | Kết quả thử nghiệm với phương pháp User-based Collaborative Filtering sử dụng KNN . . . . .                                    | 30        |
| 7.6      | Kết luận và nhận xét . . . . .                                                                                                 | 31        |
| <b>8</b> | <b>Ảnh minh họa và Hướng dẫn sử dụng Sản phẩm</b>                                                                              | <b>32</b> |
| 8.1      | Chọn Thể loại Phim . . . . .                                                                                                   | 32        |
| 8.2      | Màn Hình Chính . . . . .                                                                                                       | 33        |
| 8.3      | Thông Tin Chi Tiết Phim . . . . .                                                                                              | 34        |
| 8.4      | Lịch Sử Dánh Giá và Gợi Ý Mạng Người Dùng . . . . .                                                                            | 35        |
| <b>9</b> | <b>Tổng kết và Dự định tương lai</b>                                                                                           | <b>36</b> |
| 9.1      | Tổng kết . . . . .                                                                                                             | 36        |
| 9.2      | Dự định tương lai . . . . .                                                                                                    | 36        |

## Danh sách hình vẽ

|    |                                                                                        |    |
|----|----------------------------------------------------------------------------------------|----|
| 1  | Kết hợp mô hình GMF và MLP . . . . .                                                   | 15 |
| 2  | Hiệu suất của mô hình với các giá trị khác nhau của <code>n_factors</code> . . . . .   | 17 |
| 3  | Hiệu suất của mô hình với các giá trị khác nhau của <code>layer_sizes</code> . . . . . | 17 |
| 4  | Hiệu suất của mô hình với các giá trị khác nhau của <code>n_users</code> . . . . .     | 22 |
| 5  | Hiệu suất của mô hình với các giá trị khác nhau của <code>k</code> . . . . .           | 26 |
| 6  | Hiệu suất của phương pháp SMO (MB) với các số lượng phim đã xem khác nhau.             | 29 |
| 7  | Hiệu suất của mô hình NeuMF với các số lượng phim đã xem khác nhau. . . . .            | 30 |
| 8  | Hiệu suất của mô hình với các kịch bản khác nhau về số lượng phim đầu vào. .           | 31 |
| 9  | Trang chọn thể loại phim (chưa chọn thể loại). . . . .                                 | 32 |
| 10 | Trang chọn thể loại phim (đã chọn 3 thể loại). . . . .                                 | 33 |
| 11 | Màn hình chính của ứng dụng. . . . .                                                   | 33 |
| 12 | Gợi ý phim hot nhất theo thể loại. . . . .                                             | 34 |
| 13 | Thông tin chi tiết của bộ phim. . . . .                                                | 34 |
| 14 | Gợi ý phim tương tự sau khi đánh giá >3 sao. . . . .                                   | 35 |
| 15 | Lịch sử đánh giá 5 bộ phim. . . . .                                                    | 35 |
| 16 | Gợi ý dựa trên sở thích của người dùng khác. . . . .                                   | 36 |

## Danh sách bảng

|   |                                                            |    |
|---|------------------------------------------------------------|----|
| 1 | Hiệu suất của các phương pháp gợi ý (Precision@K). . . . . | 11 |
| 2 | Hiệu suất của các phương pháp gợi ý (NDCG và HR). . . . .  | 11 |

# Lời Mở Đầu

Bài báo cáo được chia làm 7 phần chính như sau:

- **Phần 1:** Giới thiệu bài toán
- **Phần 2:** Thu thập và chuẩn bị dữ liệu
- **Phần 3:** Phương pháp lọc dựa trên nội dung
- **Phần 4:** Phương pháp lọc cộng tác sử dụng mạng nơ-ron (Neural Collaborative Filtering)
- **Phần 5:** Phương pháp lọc cộng tác sử dụng k-Nearest Neighbour (KNN)
- **Phần 6:** Phương pháp lọc cộng tác sử dụng Approximate Nearest Neighbor (ANN) với FAISS
- **Phần 7:** Tổng kết và dự định tương lai

# 1 Giới thiệu bài toán

Trong bối cảnh sự phát triển nhanh chóng của công nghệ thông tin và lượng dữ liệu khổng lồ từ các nền tảng trực tuyến, hệ thống gợi ý đã trở thành một công cụ quan trọng giúp cải thiện trải nghiệm người dùng. Một trong những ứng dụng phổ biến nhất của hệ thống gợi ý là trong lĩnh vực giải trí, đặc biệt là các dịch vụ phát trực tuyến phim như Netflix, Amazon Prime và Disney+. Hệ thống gợi ý phim có vai trò quan trọng trong việc giúp người dùng nhanh chóng tìm thấy các nội dung phù hợp với sở thích cá nhân, từ đó tăng cường sự hài lòng và gắn kết của họ với nền tảng.

Bài toán gợi ý phim được định nghĩa là một bài toán dự đoán, trong đó mục tiêu là dự đoán các bộ phim mà một người dùng cụ thể có khả năng yêu thích dựa trên dữ liệu lịch sử về hành vi và sở thích của họ. Các phương pháp học máy được áp dụng rộng rãi để giải quyết bài toán này sẽ bao gồm lọc cộng tác (Collaborative Filtering), gợi ý dựa trên nội dung (Content-based Filtering) và mô hình lọc cộng tác sử dụng mạng nơ-ron (Neural Collaborative Filtering).

## 2 Thu thập và chuẩn bị dữ liệu

Dữ liệu đầu vào của nghiên cứu được lấy từ hai nguồn chính nhằm phục vụ cho việc phát triển và huấn luyện các hệ thống gợi ý phim:

- **Dữ liệu đánh giá người dùng (ratings.csv):**

- Bộ dữ liệu MovieLens 100K bao gồm khoảng 100.000 đánh giá phim được thực hiện bởi nhiều người dùng khác nhau.
- Đây là một bộ dữ liệu phổ biến trong lĩnh vực nghiên cứu hệ thống gợi ý và được sử dụng chủ yếu để huấn luyện các mô hình lọc cộng tác (Collaborative Filtering).
- Bộ dữ liệu cung cấp thông tin quan trọng về mối quan hệ giữa người dùng và các mục nội dung (phim) thông qua các đánh giá định lượng.

- **Dữ liệu đặc trưng nội dung (movies\_metadata.csv):**

- Bộ dữ liệu metadata của MovieLens chứa hơn 40.000 mục phim, đi kèm với các thông tin chi tiết như thể loại, từ khóa, tác giả, mô tả, dàn diễn viên, và các thuộc tính liên quan khác.
- Bộ dữ liệu này được sử dụng chủ yếu để xây dựng các mô hình gợi ý dựa trên nội dung (Content-Based Filtering).
- Việc tận dụng các thông tin chi tiết giúp hệ thống gợi ý có khả năng phân tích sâu hơn về đặc điểm nội dung của các bộ phim nhằm tăng cường khả năng cá nhân hóa.

## 3 Phương pháp lọc dựa trên nội dung

### 3.1 Giới thiệu về Content-Based Recommendation

Hệ thống gợi ý dựa trên nội dung (Content-Based Recommendation) là một kỹ thuật phổ biến trong lĩnh vực trí tuệ nhân tạo và học máy. Phương pháp này dựa trên việc phân tích các đặc điểm (*features*) của đối tượng để gợi ý các đối tượng tương tự với sở thích hoặc tương tác trước đó của người dùng.

Trong bối cảnh gợi ý phim, các đặc điểm như *thể loại (genres)*, *diễn viên chính (cast)*, *đạo diễn (director)*, *năm sản xuất (year)* và *từ khóa (keywords)* đóng vai trò quan trọng trong việc

xác định độ tương đồng giữa các bộ phim. Mỗi bộ phim được mô tả bởi một tập hợp các đặc điểm này, từ đó cho phép hệ thống gợi ý các bộ phim tương tự với các bộ phim mà người dùng đã quan tâm.

Phần này tập trung trình bày các phương pháp gợi ý dựa trên nội dung được áp dụng trong dự án, bao gồm:

- **TF-IDF và Cosine Similarity:** Mô hình hóa nội dung phim dưới dạng vector và tính toán độ tương đồng giữa các vector.
- **Word2Vec:** Áp dụng học sâu để biểu diễn đặc trưng phim và tính toán độ tương đồng giữa các phim dựa trên các đặc điểm chi tiết hơn.

## 3.2 Dữ liệu và Tiền xử lý

### 3.2.1 Tập dữ liệu sử dụng

Hệ thống gợi ý được xây dựng dựa trên các tập dữ liệu từ **MovieLens**, bao gồm:

- **movies\_metadata.csv:** Chứa thông tin chi tiết về phim, bao gồm tiêu đề (*title*), thể loại (*genres*), tóm tắt nội dung (*overview*), năm phát hành (*year*), và điểm đánh giá trung bình (*vote\_average*).
- **credits.csv:** Cung cấp danh sách diễn viên (*cast*) và đội ngũ sản xuất (*crew*) của từng phim.
- **keywords.csv:** Chứa các từ khóa (*keywords*) mô tả nội dung chính của từng phim.

### 3.2.2 Quá trình tiền xử lý dữ liệu

Tiền xử lý dữ liệu là bước quan trọng nhằm chuẩn bị các đặc điểm (*features*) để đưa vào mô hình gợi ý. Các bước thực hiện bao gồm:

**Xử lý JSON và chuẩn hóa** Các cột dữ liệu như *genres*, *cast*, và *keywords* được lưu trữ dưới dạng JSON và cần chuyển đổi thành danh sách các chuỗi (*list of strings*). Hàm `safe_list_parse` được sử dụng để đảm bảo tính an toàn khi xử lý:

- Chuyển đổi dữ liệu JSON thành danh sách (*list*).
- Loại bỏ khoảng trắng và chuyển tất cả chữ thành chữ thường để chuẩn hóa.

**Làm sạch thông tin đạo diễn** Cột *director* thường có dữ liệu thiếu (*missing values*). Để xử lý:

- Diền giá trị mặc định ('') nếu giá trị bị thiếu.
- Chuyển đổi chuỗi thành chữ thường và loại bỏ khoảng trắng.

**Tạo câu mô tả metadata** Để biểu diễn mỗi phim, các thuộc tính *director*, *genres*, *cast*, *keywords*, và *year* được kết hợp thành một danh sách duy nhất, gọi là **câu metadata (metadata sentence)**. Câu này sẽ được sử dụng trong các bước tiếp theo để tính toán độ tương đồng giữa các phim.

**Trọng số hóa dữ liệu dựa trên điểm đánh giá** Dựa vào cột *vote\_average*, mỗi câu metadata được nhân bản nhiều lần tương ứng với mức độ đánh giá của phim. Phương pháp này đảm bảo rằng các phim được đánh giá cao sẽ có trọng số lớn hơn trong mô hình.

**Kết quả tiền xử lý** Sau các bước trên, mỗi bộ phim được biểu diễn bằng một danh sách các đặc điểm chuẩn hóa, bao gồm: đạo diễn (*director*), thể loại (*genres*), diễn viên (*cast*), từ khóa (*keywords*), và năm phát hành (*year*). Các đặc điểm này sẽ được sử dụng làm đầu vào cho các thuật toán gợi ý.

### 3.3 Phương pháp

Hệ thống gợi ý dựa trên nội dung trong dự án này sử dụng hai phương pháp chính: **TF-IDF và Cosine Similarity** và **Word2Vec**. Mỗi phương pháp được áp dụng với các đặc điểm (*features*) được trích xuất từ dữ liệu phim để xác định độ tương đồng giữa các bộ phim.

#### 3.3.1 TF-IDF và Cosine Similarity

Phương pháp này biểu diễn nội dung phim dưới dạng vector dựa trên kỹ thuật **TF-IDF (Term Frequency-Inverse Document Frequency)**. Các bước thực hiện như sau:

**Tạo ma trận TF-IDF** TF-IDF là một kỹ thuật biểu diễn văn bản dưới dạng vector dựa trên tần suất xuất hiện của các từ. Công thức tính TF-IDF cho từ  $t$  trong phim  $d$  được định nghĩa như sau:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t) \quad (1)$$

Trong đó:

- $\text{TF}(t, d) = \frac{f(t, d)}{\sum_{t' \in d} f(t', d)}$ : Tần suất xuất hiện của từ  $t$  trong phim  $d$ .
- $\text{IDF}(t) = \log \frac{N}{1+n(t)}$ : Mức độ quan trọng của từ  $t$ , với:
  - $N$ : Tổng số phim trong tập dữ liệu.
  - $n(t)$ : Số lượng phim chứa từ  $t$ .

Kết quả là một ma trận TF-IDF, trong đó mỗi hàng tương ứng với một phim và mỗi cột tương ứng với một n-gram (các từ hoặc cụm từ).

**Tính độ tương đồng** Độ tương đồng giữa hai bộ phim  $d_1$  và  $d_2$  được tính toán bằng **Cosine Similarity**:

$$\text{Cosine Similarity}(d_1, d_2) = \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_1\| \|\mathbf{v}_2\|} \quad (2)$$

Trong đó:

- $\mathbf{v}_1, \mathbf{v}_2$ : Vector TF-IDF của hai bộ phim  $d_1$  và  $d_2$ .
- $\|\mathbf{v}\|$ : Độ dài của vector, được tính bằng:

$$\|\mathbf{v}\| = \sqrt{\sum_{i=1}^n v_i^2} \quad (3)$$

Giá trị cosine similarity nằm trong khoảng từ 0 đến 1, trong đó giá trị càng gần 1 thì hai bộ phim càng tương tự nhau.

**Thuật toán gợi ý** Dựa trên độ tương đồng cosine, các bộ phim được sắp xếp theo thứ tự giảm dần độ tương đồng với một phim đầu vào. Hàm `get_recommendations_tfidf(movie_id, top_k)` được triển khai để trả về danh sách các phim tương tự với một phim đầu vào:

$$\text{Ranked Movies} = \text{argsort}(-\text{Cosine Similarity}(d_{\text{input}}, d_i)) \quad \forall i \in \text{Movies} \quad (4)$$


---

### 3.3.2 Word2Vec và Mô hình học sâu

Phương pháp này sử dụng mô hình **Word2Vec CBOW (Continuous Bag of Words)** để học các biểu diễn vector cho các từ trong metadata. Các bước thực hiện như sau:

**Tạo câu trọng số (Weighted Sentences)** Mỗi bộ phim được biểu diễn bởi một danh sách từ, bao gồm: *director*, *genres*, *cast*, *keywords*, và *year*. Để phản ánh mức độ phổ biến của phim, mỗi câu được nhân bản dựa trên điểm đánh giá trung bình (*vote\_average*). Câu trọng số của phim  $d$  được biểu diễn như sau:

$$\text{Weighted Sentence}_d = \bigcup_{k=1}^{\text{round}(\text{vote\_average}_d)} \text{Metadata}_d \quad (5)$$

Trong đó:

- $\text{Metadata}_d = \{\text{director}, \text{genres}, \text{cast}, \text{keywords}, \text{year}\}$ : Danh sách từ của metadata phim  $d$ .
- $\text{vote\_average}_d$ : Điểm đánh giá trung bình của phim  $d$ .

**Lý do lựa chọn CBOW thay vì Skip-gram** Sử dụng mô hình **CBOW (Continuous Bag of Words)** thay vì **Skip-gram** để dự đoán từ hiện tại dựa trên ngữ cảnh xung quanh (tất cả các từ trong câu phim). Lý do như sau:

- Dự đoán từ hiện tại (ví dụ: *actor*) dựa trên ngữ cảnh (các từ khác trong metadata của bộ phim) hợp lý hơn và đơn giản hơn so với việc dự đoán các từ xung quanh dựa trên từ hiện tại.
- Bộ dữ liệu sử dụng, *Weighted Movie Sentence Dataset*, nhỏ hơn nhiều so với các bộ dữ liệu thông thường. Do không chắc chắn rằng Skip-gram sẽ hoạt động tốt với bộ dữ liệu nhỏ, vì vậy CBOW được lựa chọn để trích xuất đặc trưng.

**Sử dụng tham số mặc định trong huấn luyện** Hầu hết các tham số trong quá trình huấn luyện đều sử dụng giá trị mặc định, ngoại trừ *window size*. Giá trị *window size* được thiết lập tương đối lớn (= 50) để mô hình có thể xem xét toàn bộ câu phim như một ngữ cảnh khi huấn luyện biểu diễn vector cho từ. Cách tiếp cận này đảm bảo rằng tất cả các từ trong metadata của bộ phim đều được mô hình sử dụng để dự đoán ngữ nghĩa.

**Cấu hình huấn luyện mô hình Word2Vec** Mô hình **Word2Vec** được huấn luyện trên các câu trọng số để tạo ra vector biểu diễn cho từng từ. Các tham số chính:

- Kích thước vector (*vector\_size*) = 100.
- Kích thước của sổ ngữ cảnh (*window*) = 50.
- Phương pháp huấn luyện: **CBOW**.
- Số lần lặp (*epochs*) = 10.

**Tính toán độ tương đồng** Độ tương đồng giữa hai bộ phim  $d_1$  và  $d_2$  được tính dựa trên các danh mục metadata. Với mỗi danh mục  $c$ , độ tương đồng được tính như sau:

$$\text{Cosine Similarity}_c(d_1, d_2) = \frac{\sum_{i \in \mathbf{w}_1, j \in \mathbf{w}_2} \text{Cosine Similarity}(\mathbf{v}_{w_i}, \mathbf{v}_{w_j})}{|\mathbf{w}_1| \cdot |\mathbf{w}_2|} \quad (6)$$

Trong đó:

- $\mathbf{w}_1, \mathbf{w}_2$ : Các từ trong danh mục  $c$  của phim  $d_1$  và  $d_2$ .
- $\mathbf{v}_{w_i}$ : Vector biểu diễn của từ  $w_i$  trong mô hình Word2Vec.

Dộ tương đồng tổng thể giữa hai phim được tính bằng trung bình tổng hợp theo từng danh mục metadata:

$$\text{Similarity}(d_1, d_2) = \frac{\sum_{c \in \text{Metadata Categories}} \text{Cosine Similarity}_c(d_1, d_2)}{|\text{Metadata Categories}|} \quad (7)$$

**Thuật toán gợi ý** Hàm `get_recommendations_smo_mb(movie_id, top_k)` trả về danh sách các phim tương tự với một phim đầu vào dựa trên độ tương đồng tổng thể. Phim được sắp xếp theo thứ tự giảm dần độ tương đồng:

$$\text{Ranked Movies} = \text{argsort}(-\text{Similarity}(d_{\text{input}}, d_i)) \quad \forall i \in \text{Movies} \quad (8)$$

### 3.4 Kết quả và Đánh giá

Đánh giá hiệu quả của các phương pháp gợi ý sử dụng các thước đo chính là **Precision@K**, **NDCG**, và **HR**. Các thước đo này được định nghĩa như sau:

#### Precision@K

$$\text{Precision}@K = \frac{\text{Số lượng phim người dùng đã xem trong top K gợi ý}}{\text{Tổng số gợi ý trong top K}} \quad (9)$$

Precision@K đo lường độ chính xác của hệ thống gợi ý trong việc dự đoán các phim mà người dùng quan tâm. Một hệ thống tốt sẽ có giá trị Precision@K cao hơn, đặc biệt ở các mức K nhỏ.

**NDCG (Normalized Discounted Cumulative Gain)** NDCG đo lường mức độ liên quan của các gợi ý, trong đó các mục liên quan hơn sẽ được xếp hạng cao hơn. Thước đo này có tính đến thứ tự sắp xếp của các gợi ý, với công thức:

$$\text{NDCG} = \frac{\text{DCG}}{\text{IDCG}} \quad (10)$$

trong đó DCG là Cumulative Gain có trọng số và IDCG là giá trị DCG tối đa (tương ứng với thứ tự sắp xếp tối ưu).

**HR (Hit Ratio)** HR đo tỷ lệ các mục trong danh sách gợi ý được người dùng thực sự quan tâm:

$$\text{HR} = \frac{\text{Số lượng gợi ý trúng}}{\text{Tổng số mục trong danh sách gợi ý}} \quad (11)$$

Thước đo này đánh giá khả năng của hệ thống trong việc tìm đúng các mục mà người dùng quan tâm.

### 3.4.1 Các phương pháp thử nghiệm

Tiến hành thử nghiệm ba nhóm phương pháp:

- **TF-IDF (Term Frequency-Inverse Document Frequency):**

- **TF-IDF Overview:** Chỉ sử dụng nội dung tóm tắt (*overview*) của phim để tạo vector TF-IDF.
- **TF-IDF Metadata:** Sử dụng các đặc điểm *director*, *actor*, *keywords*, *genres* để tạo vector TF-IDF.

- **Word2Vec (CBOW - Continuous Bag of Words):**

- **Fully-Connected (FC):** Tính toán độ tương đồng giữa tất cả các từ trong metadata.
- **Same Metadata Only (SMO):** Tính toán độ tương đồng giữa các từ trong cùng một danh mục metadata.

- **Cách trung bình độ tương đồng:**

- **Total Average (TA):** Trung bình toàn bộ giá trị trong ma trận cosine similarity.
- **Metadata-Based (MB):** Trung bình độ tương đồng trong từng danh mục metadata, sau đó tổng hợp.

### 3.4.2 Kết quả Precision@K

Kết quả thử nghiệm được trình bày trong Bảng 1. Để tính toán **Precision@K**, thực hiện các bước như sau:

**Cách thực hiện:**

- Đầu tiên, hệ thống gợi ý một danh sách phim cho mỗi người dùng. Danh sách này được sắp xếp theo độ liên quan từ cao đến thấp (dựa trên điểm số dự đoán của từng phương pháp gợi ý).
- Với mỗi người dùng, so sánh danh sách  $K$  phim được gợi ý với danh sách các phim mà người dùng đã thực sự xem.
- Đếm số lượng phim trong danh sách gợi ý mà người dùng đã xem. Sau đó, chia số lượng này cho  $K$  để tính giá trị Precision@K.

**Ví dụ:** Nếu hệ thống gợi ý top 10 phim (Precision@10) và trong đó có 3 phim đã được người dùng xem, giá trị Precision@10 sẽ được tính là  $3/10 = 0.3 = 30\%$ .

**Kết quả:** Sau khi tính trung bình trên 943 người dùng trong tập dữ liệu MovieLens 100K, Bảng 1 thể hiện kết quả Precision@K của các phương pháp gợi ý. Trong đó, phương pháp **Word2Vec SMO (MB)** đạt hiệu quả cao nhất ở tất cả các mức  $K$ .

Bảng 1: Hiệu suất của các phương pháp gợi ý (Precision@K).

| Phương pháp       | Precision@1   | Precision@10 | Precision@100 | Precision@200 |
|-------------------|---------------|--------------|---------------|---------------|
| TF-IDF Overview   | 3.21%         | 2.89%        | 2.01%         | 1.85%         |
| TF-IDF Metadata   | 7.89%         | 5.45%        | 3.32%         | 3.01%         |
| Word2Vec FC (TA)  | 10.12%        | 6.78%        | 4.35%         | 3.98%         |
| Word2Vec FC (MB)  | 12.45%        | 7.12%        | 4.68%         | 4.21%         |
| Word2Vec SMO (TA) | 13.95%        | 7.89%        | 4.91%         | 4.33%         |
| Word2Vec SMO (MB) | <b>15.07%</b> | <b>8.12%</b> | <b>5.12%</b>  | <b>4.58%</b>  |

### 3.4.3 Kết quả NDCG và HR

Kết quả thử nghiệm NDCG và HR được trình bày trong Bảng 2. Dưới đây là cách tính và các bước thực hiện để đo lường hiệu suất của các phương pháp dựa trên hai chỉ số này:

Cách thực hiện:

- Đối với **HR**: Với mỗi người dùng, kiểm tra danh sách  $K$  phim gợi ý và xem có bao nhiêu phim trong danh sách đó đã thực sự được người dùng xem. Chia số lượng này cho tổng số phim trong danh sách để tính HR.
- Đối với **NDCG**: Thứ tự của các phim gợi ý được tính đến. Các phim được người dùng thực sự xem ở vị trí cao hơn trong danh sách gợi ý sẽ đóng góp nhiều hơn vào giá trị NDCG.

**Ví dụ:** Nếu một danh sách gợi ý gồm 10 phim có 5 phim thực sự được người dùng xem, thì HR@10 là  $5/10 = 0.5 = 50\%$ . Nếu trong 5 phim đó, các phim được xem nằm ở đầu danh sách, giá trị NDCG sẽ cao hơn, phản ánh rằng hệ thống không chỉ gợi ý đúng mà còn sắp xếp chính xác.

**Kết quả:** Sau khi tính trung bình trên 943 người dùng trong tập dữ liệu MovieLens 100K, Bảng 2 cho thấy phương pháp **Word2Vec SMO (MB)** tiếp tục đạt hiệu quả cao nhất ở cả hai thước đo, khẳng định tính hiệu quả vượt trội của phương pháp này.

Bảng 2: Hiệu suất của các phương pháp gợi ý (NDCG và HR).

| Phương pháp       | NDCG         | HR           |
|-------------------|--------------|--------------|
| TF-IDF Overview   | 2.8%         | 3.5%         |
| TF-IDF Metadata   | 7.5%         | 8.0%         |
| Word2Vec FC (TA)  | 12.0%        | 13.0%        |
| Word2Vec FC (MB)  | 14.0%        | 15.0%        |
| Word2Vec SMO (TA) | 15.5%        | 16.5%        |
| Word2Vec SMO (MB) | <b>16.5%</b> | <b>17.5%</b> |

### 3.4.4 Đánh giá và Phân tích

Dựa trên kết quả từ Bảng 1 và Bảng 2, chúng ta có một số nhận xét như sau:

## Hiệu quả của TF-IDF

- Phương pháp **TF-IDF Overview** đạt kết quả thấp nhất trong tất cả các thử nghiệm, với Precision@1 chỉ đạt 3.21% và NDCG đạt 2.8%. Điều này cho thấy rằng nội dung tóm tắt (*overview*) không cung cấp đủ thông tin để xác định độ tương đồng chính xác giữa các phim.
- Phương pháp **TF-IDF Metadata** cải thiện đáng kể hiệu suất, với Precision@1 đạt 7.89% và NDCG đạt 7.5%. Điều này khẳng định rằng việc sử dụng các đặc điểm như *director*, *actor*, *keywords*, *genres* mang lại nhiều thông tin hơn so với nội dung tóm tắt.

## Hiệu quả của Word2Vec

- Các phương pháp sử dụng Word2Vec đạt hiệu quả cao hơn đáng kể so với TF-IDF. Đặc biệt, cấu hình **Word2Vec SMO (MB)** cho kết quả Precision@1 cao nhất (15.07%), NDCG (16.5%), và HR (17.5%), vượt trội so với các phương pháp khác.
- Sự khác biệt giữa cách tính **Fully-Connected (FC)** và **Same Metadata Only (SMO)** cho thấy rằng việc tính toán độ tương đồng trong từng danh mục (*SMO*) mang lại kết quả tốt hơn, do giảm được nhiều từ các danh mục không liên quan.

## Hiệu quả của các cách trung bình

- Cách trung bình **Metadata-Based (MB)** vượt trội so với **Total Average (TA)** trong cả hai cấu hình Word2Vec. Điều này chứng tỏ rằng việc tính toán độ tương đồng riêng biệt trong từng danh mục metadata giúp cải thiện độ chính xác của hệ thống gợi ý.
- MB đặc biệt hiệu quả trong các mức Precision@100 và Precision@200, cũng như khi đo lường bằng NDCG và HR, cho thấy khả năng xử lý tốt khi mở rộng số lượng gợi ý.

**Phương pháp tốt nhất** Cấu hình **Word2Vec SMO (MB)** chứng tỏ là phương pháp tốt nhất trong thử nghiệm này, đạt hiệu quả cao nhất ở mọi thước đo. Phương pháp này tận dụng được lợi thế của Word2Vec để học ngữ nghĩa từ metadata, đồng thời tối ưu hóa tính toán độ tương đồng thông qua cách trung bình Metadata-Based.

## 4 Mô hình Neural Collaborative Filtering (NCF)

### 4.1 Giới thiệu về Neural Collaborative Filtering (NCF)

Neural Collaborative Filtering (NCF) là một mô hình gợi ý tiên tiến dựa trên mạng nơ-ron sâu, nhằm khắc phục những hạn chế của các phương pháp gợi ý truyền thống.

- **Mô hình NCF kết hợp:**
  - **Generalized Matrix Factorization (GMF):** Tận dụng tính tuyến tính để biểu diễn các mối quan hệ giữa người dùng và sản phẩm.
  - **Multi-Layer Perceptron (MLP):** Khai thác tính phi tuyến để học các đặc trưng tiềm ẩn phức tạp hơn.
- **Bằng cách kết hợp ưu điểm của GMF và MLP, NCF có khả năng:**
  - Học các mối quan hệ tiềm ẩn giữa người dùng và sản phẩm một cách hiệu quả.

- Cải thiện độ chính xác và chất lượng của các dự đoán gợi ý.
- **Phù hợp với dữ liệu lớn và không đồng nhất:** Cấu trúc mạng nơ-ron sâu giúp mô hình mở rộng và thích nghi tốt với các hệ thống gợi ý hiện đại.

## 4.2 Generalized Matrix Factorization

Generalized Matrix Factorization (GMF) là sự mở rộng của Matrix Factorization (MF) thông qua việc bổ sung một lớp Neural Collaborative Filtering (NCF) vào đầu ra. GMF tận dụng tính tuyến tính để biểu diễn mối quan hệ giữa người dùng và sản phẩm như sau:

$$\hat{r}_{u,i} = q_i^T p_u \quad (12)$$

Trong đó:

- $\hat{r}_{u,i}$ : Điểm dự đoán (rating) của người dùng  $u$  đối với sản phẩm  $i$
- $q_i$ : Vector đặc trưng của sản phẩm  $i$
- $p_u$ : Vector đặc trưng của người dùng  $u$

GMF thay thế tích vô hướng bằng phép nhân từng phần tử (element-wise product) và áp dụng hàm kích hoạt phi tuyến:

$$\hat{r}_{u,i} = a_{out} (h^T (q_i \odot p_u)) \quad (13)$$

Trong đó:

- $\odot$ : Phép nhân từng phần tử
- $h$ : Vector trọng số của lớp đầu ra
- $a_{out}$ : Hàm kích hoạt phi tuyến, ví dụ: ReLU hoặc sigmoid

Lợi ích của GMF:

- Tăng tính linh hoạt trong việc học các trọng số
- Cải thiện khả năng biểu diễn các mối quan hệ giữa người dùng và sản phẩm
- Cho phép tích hợp các hàm kích hoạt phi tuyến như ReLU, Sigmoid

## 4.3 Multi-Layer Perceptron

**Định nghĩa:** Multi-Layer Perceptron (MLP) là một mạng nơ-ron nhiều lớp, được thiết kế để học các tương tác phi tuyến giữa người dùng và sản phẩm. Đây là một phần quan trọng của mô hình NCF, giúp tăng cường khả năng biểu diễn và dự đoán các mối quan hệ phức tạp.

**Đầu vào:** Vector đặc trưng của người dùng ( $p_u$ ) và sản phẩm ( $q_i$ ) được ghép nối thành một vector đầu vào:

$$z_1 = \phi_1(p_u, q_i) = \begin{bmatrix} p_u \\ q_i \end{bmatrix}$$

**Cấu trúc mạng:** MLP bao gồm nhiều tầng ẩn, mỗi tầng áp dụng hàm kích hoạt phi tuyến để học các mối quan hệ phức tạp:

$$\phi_l(z_l) = a_{out}(W_l^T z_l + b_l), \quad l = 2, 3, \dots, L - 1$$

Trong đó:

- $W_l$ : Ma trận trọng số tại tầng  $l$ .
- $b_l$ : Vector bias tại tầng  $l$ .
- $z_l$ : Đầu vào của tầng  $l$ .
- $a_{out}$ : Hàm kích hoạt phi tuyến, ví dụ: ReLU, sigmoid hoặc tanh.

**Dự đoán đầu ra:** Tầng cuối cùng của MLP dự đoán xác suất tương tác giữa người dùng và sản phẩm:

$$\hat{r}_{u,i} = \sigma(h^T \phi(z_{L-1}))$$

Trong đó:

- $h$ : Vector trọng số của lớp đầu ra.
- $\sigma$ : Hàm kích hoạt sigmoid, giúp giới hạn đầu ra trong khoảng  $[0, 1]$ .

### Ưu điểm:

- MLP có khả năng học các tương tác phi tuyến, giúp mô hình hóa các mối quan hệ tiềm ẩn phức tạp hơn so với các phương pháp tuyến tính.
- Kết hợp với GMF, MLP mang lại sự toàn diện và hiệu quả trong các hệ thống gợi ý hiện đại.

## 4.4 Tạo mô hình NCF bằng cách kết hợp GMF và MLP

Neural Collaborative Filtering (NCF) kết hợp hai thành phần chính: Generalized Matrix Factorization (GMF) và Multi-Layer Perceptron (MLP), nhằm tận dụng cả tính tuyến tính lẫn phi tuyến để học các mối quan hệ tiềm ẩn giữa người dùng và sản phẩm.

- **GMF:** GMF thực hiện phép nhân từng phần tử giữa vector đặc trưng của người dùng ( $p_u^{GMF}$ ) và sản phẩm ( $q_i^{GMF}$ ):

$$\phi_{u,i}^{GMF} = p_u^{GMF} \odot q_i^{GMF}$$

- **MLP:** MLP thực hiện phép ghép nối vector người dùng và sản phẩm, sau đó truyền qua các tầng phi tuyến:

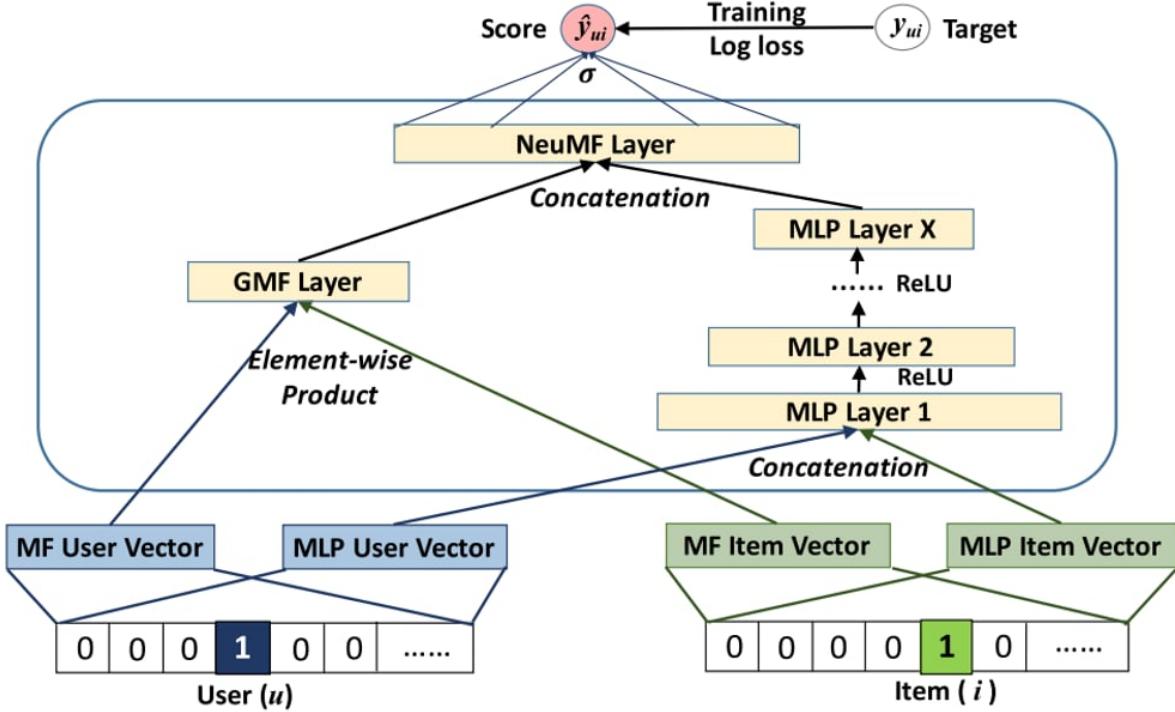
$$\phi_l(z_l) = a_{out}(W_l^T z_l + b_l), \quad l = 2, 3, \dots, L-1$$

- **Kết hợp GMF và MLP:** Đầu ra của GMF và MLP được ghép nối thành một vector hợp nhất, sau đó tính xác suất tương tác giữa người dùng và sản phẩm:

$$\hat{r}_{u,i} = \sigma\left(h^T \begin{bmatrix} \phi_{u,i}^{GMF} \\ \phi_{u,i}^{MLP} \end{bmatrix}\right)$$

Trong đó:

- $h$ : Vector trọng số của lớp đầu ra.
- $\sigma$ : Hàm kích hoạt sigmoid để chuẩn hóa đầu ra trong khoảng  $(0, 1)$ .



Hình 1: Kết hợp mô hình GMF và MLP

## 4.5 Hàm mục tiêu và Phương pháp tối ưu

Hàm mục tiêu của mô hình NeuMF được định nghĩa dựa trên hàm cross-entropy loss, thường được sử dụng để tối ưu hóa các bài toán dự đoán xác suất. Hàm loss được biểu diễn như sau:

$$L = - \sum (r_{u,i} \log \hat{r}_{u,i} + (1 - r_{u,i}) \log(1 - \hat{r}_{u,i}))$$

Trong đó:

- $r_{u,i} = 1$ : Nếu người dùng  $u$  đã tương tác (hoặc thích) sản phẩm  $i$ .
- $r_{u,i} = 0$ : Nếu người dùng  $u$  không tương tác (hoặc không thích) sản phẩm  $i$ .
- $\hat{r}_{u,i}$ : Xác suất được mô hình dự đoán cho việc người dùng  $u$  sẽ tương tác (hoặc thích) sản phẩm  $i$ .

Để tối ưu hóa hàm mục tiêu, NeuMF sử dụng thuật toán Adam Optimizer, một phương pháp hiện đại cho việc điều chỉnh gradient trong quá trình cập nhật tham số. Phương pháp này giúp tăng tốc độ hội tụ và cải thiện hiệu suất so với các phương pháp tối ưu hóa truyền thống như SGD.

## 4.6 Đánh giá

### 4.6.1 Phương pháp đánh giá

Quá trình đánh giá hiệu suất của mô hình NeuMF được thực hiện thông qua các bước và chỉ số sau:

- **Phương pháp leave-one-out:** Trong tập kiểm tra, mỗi người dùng được chọn một phim là tích cực (người dùng đã xem hoặc thích phim đó), còn lại được dùng để train.

**Quy trình đánh giá:** Các chỉ số được sử dụng để đánh giá hiệu suất mô hình bao gồm:

- **Hit Ratio at K (HR@K):** Kiểm tra xem phim tích cực có nằm trong danh sách top  $K$  phim được hệ thống gợi ý hay không:

$$HR = \begin{cases} 1, & \text{nếu phim tích cực nằm trong top } K \\ 0, & \text{nếu ngược lại.} \end{cases}$$

- **Normalized Discounted Cumulative Gain (NDCG@K):** Đánh giá độ quan trọng của vị trí phim tích cực trong danh sách top  $K$  phim gợi ý, sử dụng công thức:

$$NDCG@K = \frac{1}{\log_2(\text{rank} + 1)}$$

Trong đó:

- \* rank: Vị trí của phim tích cực trong danh sách top  $K$ .

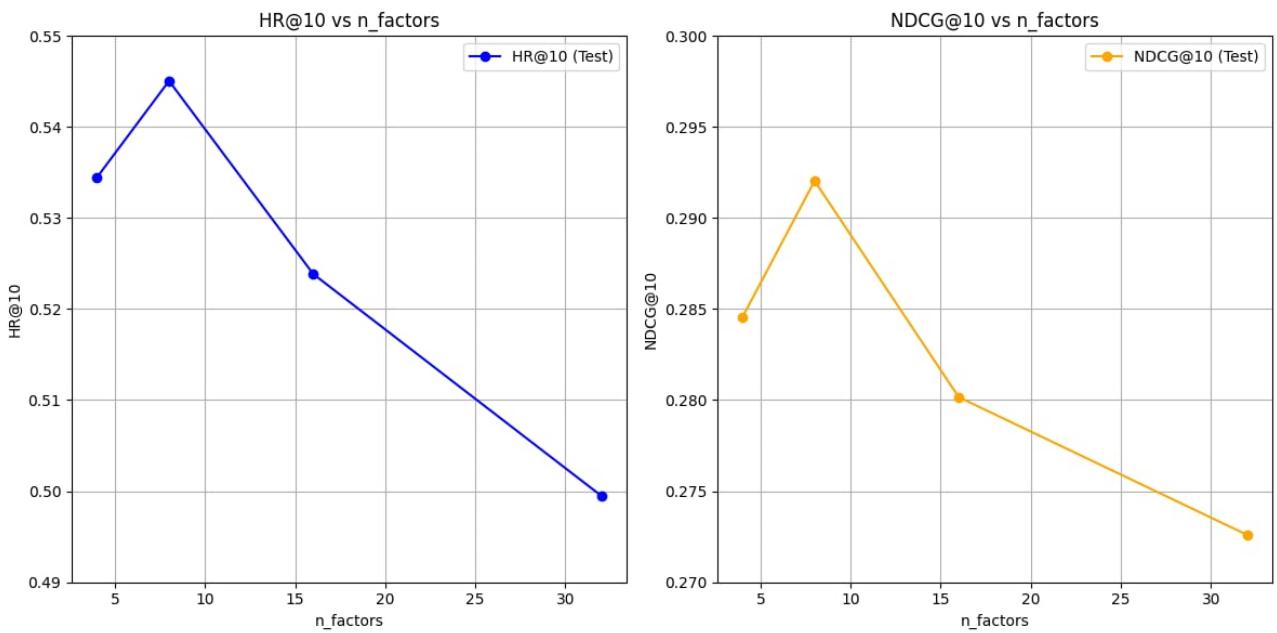
#### 4.6.2 Các tham số trong mô hình NCF

- **n\_factors:** Kích thước không gian tiềm ẩn, là số chiều của vector đặc trưng ẩn của người dùng và sản phẩm.
- **layer\_sizes:** Cấu trúc của MLP, bao gồm số tầng và số nút trong mỗi tầng. Ví dụ,  $[16, 8, 4]$  đại diện cho 3 tầng với số nút giảm dần từ 16, 8, đến 4.
- **n\_epochs:** Số vòng lặp qua toàn bộ dữ liệu trong quá trình huấn luyện, ví dụ 100 vòng.
- **batch\_size:** Số mẫu dữ liệu trong mỗi batch được sử dụng trong một lần cập nhật, ví dụ 256.
- **model\_type:** Loại mô hình được sử dụng, chẳng hạn "GMF"(Generalized Matrix Factorization), "MLP"(Multi-Layer Perceptron), hoặc "NeuMF"(Neural Matrix Factorization).
- **learning\_rate:** Tốc độ học, là tham số điều chỉnh mức độ thay đổi trọng số trong mỗi lần cập nhật.

#### 4.6.3 Kết quả thử nghiệm các tham số

Nhóm đã tiến hành thử nghiệm trên 3 tham số quan trọng trong mô hình NeuMF và đánh giá sự thay đổi kết quả khi thay đổi các tham số: `n_factors`, `layer_sizes`.

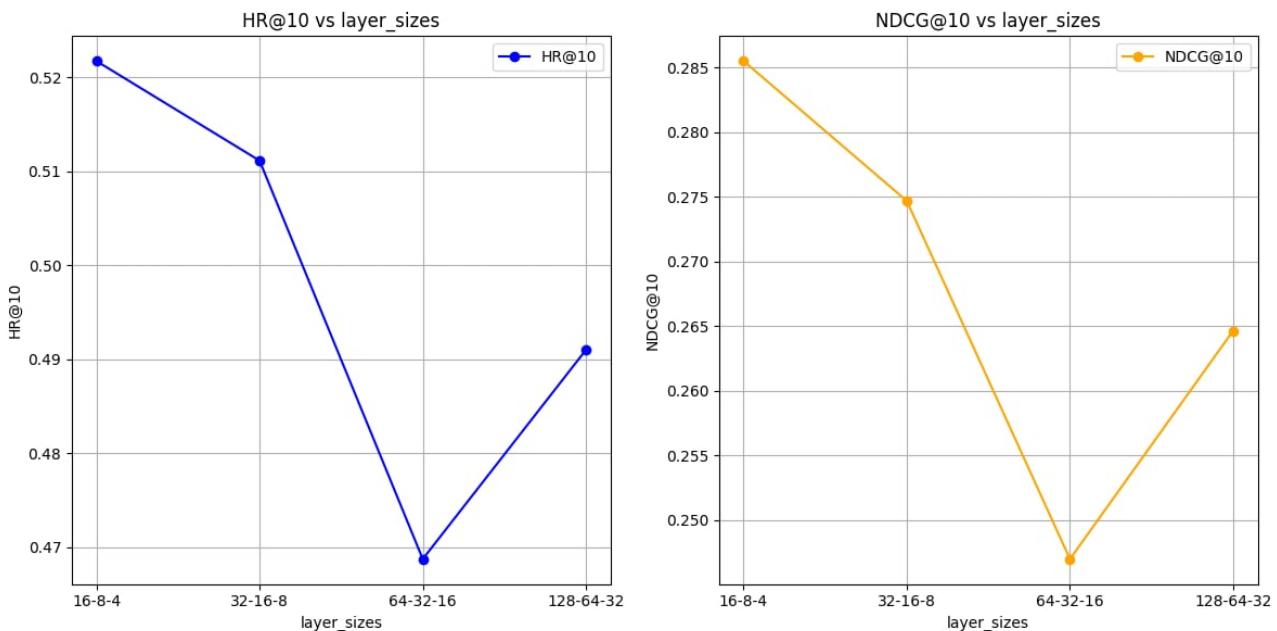
**Đối với tham số `n_factors`:**



Hình 2: Hiệu suất của mô hình với các giá trị khác nhau của `n_factors`.

Khi đánh giá trên tập test, HR@10 và NDCG@10 đạt giá trị cao nhất tại `n_factors` = 8. Khi `n_factors` tăng lên vượt quá 8, hiệu suất giảm dần. Điều này có thể do mô hình trở nên quá phức tạp, dẫn đến overfitting hoặc khó khăn trong việc học các đặc trưng hữu ích từ dữ liệu. Tương tự, tại `n_factors` thấp hơn 8 (ví dụ, 4), hiệu suất cũng thấp hơn, có thể vì không gian tiềm ẩn không đủ lớn để mô hình hóa tốt các mối quan hệ giữa người dùng và sản phẩm.

**Đối với tham số layer\_sizes:**



Hình 3: Hiệu suất của mô hình với các giá trị khác nhau của `layer_sizes`.

HR@10 và NDCG@10 đạt giá trị cao nhất khi cấu trúc `layer_sizes` là [16-8-4]. Khi `layer_sizes` tăng lên, hiệu suất giảm đáng kể, đặc biệt tại [64-32-16], hiệu suất đạt mức thấp nhất.

#### 4.6.4 Kết luận

- Tổng quan từ các kết quả đạt được, có thể nhận thấy rằng khi tăng độ phức tạp của mô hình thông qua các tham số như `n_factors`, `layer_sizes`, hoặc `batch_size`, hiệu suất của mô hình trên tập kiểm tra có xu hướng giảm đi.
- Điều này cho thấy rằng việc làm cho mô hình quá phức tạp không đồng nghĩa với việc cải thiện chất lượng gợi ý; ngược lại, nó có thể dẫn đến hiện tượng overfitting, khiến mô hình học quá mức vào dữ liệu huấn luyện và giảm khả năng tổng quát hóa trên dữ liệu kiểm tra.

### 4.7 Đưa ra gợi ý với người dùng mới

Khi một người dùng mới bắt đầu tương tác với nền tảng (ví dụ: xem phim), dữ liệu tương tác của họ hoàn toàn mới đối với mô hình NeuMF và chưa từng được mô hình thấy hay huấn luyện trước đây. Điều này dẫn đến các vấn đề sau:

- Nếu chỉ sử dụng hàm `predict`, mô hình sẽ không thể đưa ra dự đoán chính xác do không có bất kỳ thông tin nào về người dùng này.
- Mô hình không có bất kỳ *user embedding* nào được tạo sẵn cho người dùng mới.

**Cách tiếp cận cơ bản:** Một cách đơn giản để giải quyết vấn đề này là gọi lại hàm `fit` để huấn luyện lại toàn bộ mô hình từ đầu, tạo mới các embedding cho người dùng mới. Tuy nhiên:

- Quá trình này sẽ khởi tạo lại tất cả các trọng số của mô hình NeuMF.
- Việc huấn luyện lại toàn bộ mô hình gây tốn kém tài nguyên và thời gian đáng kể.

**Giải pháp hiệu quả hơn:** **Weighted Pooling** Để giải quyết vấn đề một cách hiệu quả, phương pháp *weighted pooling* được áp dụng

### 4.8 Weighted Pooling cho người dùng mới

Embedding cho người dùng mới (*user embedding*) được tính toán dựa trên phương pháp *weighted pooling*, sử dụng các embedding của sản phẩm đã được huấn luyện trước. Phương pháp này cho phép mô hình đưa ra khuyến nghị ngay cả khi không có thông tin trực tiếp về người dùng trong quá trình huấn luyện.

**Đầu vào:**

- `new_user_data`: Tập dữ liệu tương tác của người dùng mới, thường gồm khoảng 10 phim mà họ đã xem.
- `rating`: Trọng số được lấy từ đánh giá của người dùng mới đối với từng bộ phim họ đã xem.

**Quy trình Weighted Pooling:**

1. Ban đầu, một *user embedding* mới được tạo ngẫu nhiên.
2. Sử dụng phương pháp *weighted pooling* để tính toán embedding này:
  - Lấy trung bình có trọng số của các embedding phim mà người dùng đã tương tác.

- Trọng số được xác định bởi đánh giá (rating) của người dùng, với các phim được đánh giá cao hơn sẽ đóng góp nhiều hơn.

Công thức được định nghĩa như sau:

$$\text{user\_embedding} = \frac{\sum_{i=1}^N (\text{rating}_i \cdot \text{item\_embedding}_i)}{\sum_{i=1}^N \text{rating}_i}$$

Trong đó:

- $\text{rating}_i$ : Trọng số của sản phẩm  $i$  (giá trị đánh giá).
- $\text{item\_embedding}_i$ : Embedding của sản phẩm  $i$ .
- $N$ : Tổng số sản phẩm mà người dùng đã tương tác.

## 4.9 Kết quả

- **Khi chỉ gọi hàm predict mà chưa được train lại:** Mô hình không đưa ra bất kỳ gợi ý chính xác nào cho người dùng mới, do thiếu thông tin và embedding được huấn luyện trước.
- **Khi sử dụng phương pháp Weighted Pooling:** Phương pháp này đạt độ chính xác 30% trên tập người dùng mới hoàn toàn chưa được huấn luyện bởi NeuMF. Điều này chứng minh rằng phương pháp Weighted Pooling có hiệu quả đáng kể trong việc cung cấp các khuyến nghị chính xác mà không cần phải huấn luyện lại toàn bộ mô hình.

# 5 User-based Collaborative Filtering sử dụng KNN

## 5.1 Giới thiệu về User-based Collaborative Filtering sử dụng KNN

User-based Collaborative Filtering (UBCF) là một phương pháp truyền thống trong các hệ thống gợi ý, tập trung vào việc xác định mức độ tương đồng giữa người dùng dựa trên lịch sử đánh giá hoặc hành vi tương tác của họ.

**Mô hình UBCF sử dụng:**

- **Cosine Similarity:**

- Đo lường độ tương đồng giữa hai vector đặc trưng biểu diễn đánh giá của người dùng.
- Tính toán góc giữa hai vector để xác định mức độ tương tự giữa các người dùng trong không gian đặc trưng.

- **K-Nearest Neighbors (KNN):**

- Tìm kiếm  $k$  người dùng gần nhất dựa trên kết quả Cosine Similarity.
- Các đánh giá của  $k$  người dùng này được sử dụng để đưa ra gợi ý phù hợp cho người dùng hiện tại.

**Lợi ích của phương pháp:**

- Tận dụng được dữ liệu đánh giá của người dùng để đưa ra các đề xuất có tính cá nhân hóa.
- Dễ triển khai và trực quan khi giải thích trong các hệ thống gợi ý.

## 5.2 Công thức tính toán tương đồng bằng Cosine Similarity

Cosine Similarity giữa hai người dùng  $u$  và  $v$  được tính bằng công thức sau:

$$\text{sim}(u, v) = \frac{\sum_{i \in I_{uv}} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2} \cdot \sqrt{\sum_{i \in I_v} r_{vi}^2}}$$

Trong đó:

- $r_{ui}$ : Điểm đánh giá của người dùng  $u$  cho sản phẩm  $i$ .
- $I_u, I_v$ : Tập các sản phẩm được đánh giá bởi  $u$  và  $v$ .
- $I_{uv}$ : Tập giao các sản phẩm được cả hai người dùng đánh giá.

**Ưu điểm:**

- Đơn giản và dễ tính toán.
- Phù hợp với dữ liệu thưa thớt, nơi các đánh giá chủ yếu là 0.

## 5.3 Quy trình áp dụng

**Bước 1: Tìm người dùng tương đồng nhất**

- Sử dụng mô hình **K-Nearest Neighbors (KNN)** để tìm  $k$ -người dùng có độ tương đồng cao nhất với người dùng hiện tại (user\_id).
- Độ tương đồng giữa hai người dùng được tính toán bằng **Cosine Similarity**:

$$\text{similarity} = 1 - \text{distance}$$

Trong đó:

- $\text{distance}$ : Khoảng cách Cosine giữa vector đặc trưng của hai người dùng.

**Bước 2: Lấy đánh giá của những người dùng tương đồng**

- Dựa trên danh sách  $k$ -người dùng tương đồng đã tìm được, hệ thống thu thập các đánh giá phim của họ.
- Chỉ giữ lại các bộ phim thỏa mãn điều kiện sau:
  - Có điểm đánh giá lớn hơn giá trị tối thiểu  $\text{min\_rating}$ .
  - Chưa được người dùng hiện tại (user\_id) xem.

**Bước 3: Tính điểm trung bình có trọng số cho từng bộ phim**

- Điểm trung bình có trọng số được tính như sau:

$$\text{avg\_rating}(i) = \frac{\sum_{v \in N_k(u)} \text{sim}(u, v) \cdot r_{vi}}{\sum_{v \in N_k(u)} \text{sim}(u, v)}$$

Trong đó:

- $r_{vi}$ : Điểm đánh giá của người dùng  $v$  cho bộ phim  $i$ .

- sim( $u, v$ ): Độ tương đồng giữa người dùng hiện tại  $u$  và người dùng  $v$ .

#### Bước 4: Ưu tiên các bộ phim phổ biến

- Thêm chỉ số *viewer score* để đại diện cho độ phổ biến của bộ phim:

$$\text{viewer\_score}(i) = \frac{\text{viewer count for movie } i}{\max \text{ viewer count for all movies}} \times 5$$

#### Bước 5: Tính điểm tổng hợp cuối cùng

- Điểm tổng hợp cuối cùng cho mỗi bộ phim được tính bằng cách kết hợp giữa *viewer score* và *avg rating*:

$$\text{final\_score}(i) = \text{viewer\_score}(i) \times 0.51 + \text{avg\_rating}(i) \times 0.49$$

#### Bước 6: Lấy top $n$ -phim để gợi ý

- Các bộ phim được sắp xếp giảm dần theo *final\_score*.
- Top  $n$ -phim có điểm số cao nhất sẽ được chọn làm gợi ý cho người dùng.

### 5.4 Đánh giá thuật toán KNN

#### 5.4.1 Phương pháp đánh giá

Quá trình đánh giá hiệu suất của thuật toán K-Nearest Neighbors (KNN) được thực hiện thông qua các bước và chỉ số cụ thể như sau:

##### Tập dữ liệu và phương pháp đánh giá:

- Dữ liệu đánh giá từ người dùng được sử dụng toàn bộ để tính toán độ tương đồng giữa các người dùng và đưa ra gợi ý dựa trên kết quả của  $k$ -người dùng gần nhất.

**Quy trình đánh giá:** Các chỉ số được sử dụng để đánh giá hiệu suất bao gồm:

- **Hit Ratio at K (HR@K):** Kiểm tra xem phim tích cực có nằm trong danh sách top  $K$  phim được hệ thống gợi ý hay không;

$$\text{HR} = \begin{cases} 1, & \text{nếu phim tích cực nằm trong top } K \\ 0, & \text{nếu ngược lại.} \end{cases}$$

- **Normalized Discounted Cumulative Gain (NDCG@K):** Đánh giá độ quan trọng của vị trí phim tích cực trong danh sách top  $K$  phim gợi ý, sử dụng công thức:

$$\text{NDCG@K} = \frac{1}{\log_2(\text{rank} + 1)}$$

Trong đó:

- **rank:** Vị trí của phim tích cực trong danh sách top  $K$ .

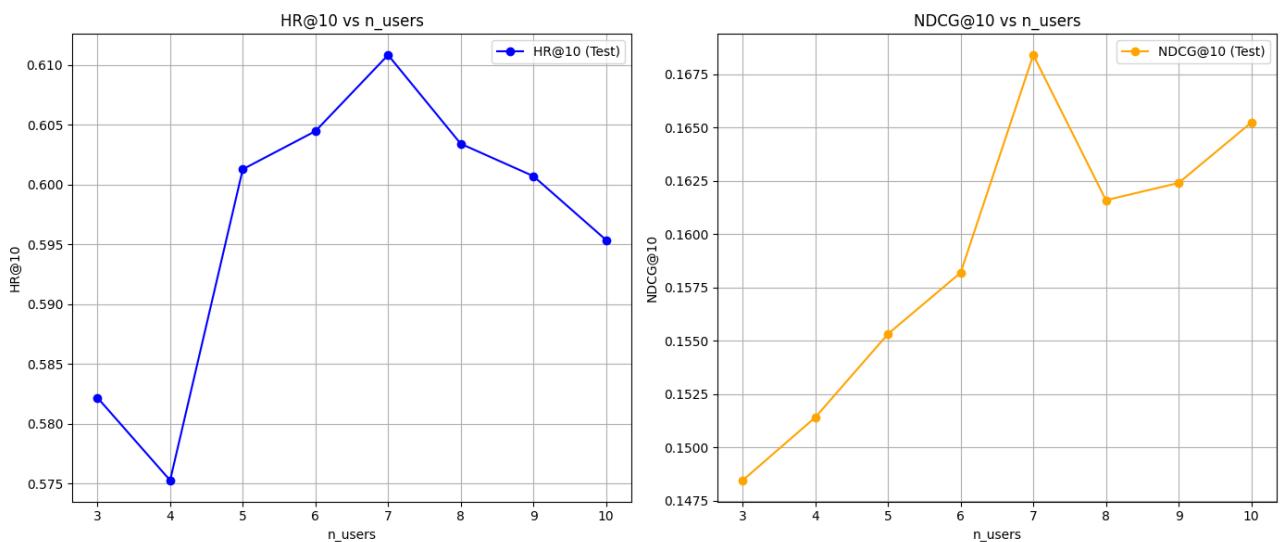
#### 5.4.2 Các tham số trong User-based Collaborative Filtering using KNN

- **n\_users:** Số lượng người dùng tương tự (nearest neighbors) mà hệ thống sẽ tìm kiếm để so sánh với người dùng hiện tại.
- **rec\_top\_n:** Số lượng phim tối đa mà hệ thống sẽ gợi ý cho người dùng hiện tại.
- **min\_rating:** Mức đánh giá tối thiểu để xem xét một bộ phim trong dữ liệu. Chỉ những bộ phim có đánh giá lớn hơn hoặc bằng giá trị này từ những người dùng tương tự mới được tính toán.

#### 5.4.3 Kết quả thử nghiệm các tham số

Nhóm đã tiến hành thử nghiệm trên tham số quan trọng trong bài toán và đánh giá sự thay đổi kết quả khi thay đổi các tham số: **n\_users**.

**Đối với tham số n\_users:**



Hình 4: Hiệu suất của mô hình với các giá trị khác nhau của **n\_users**.

Khi đánh giá trên tập test, HR@10 và NDCG@10 đạt giá trị cao nhất tại **n\_users = 7**. Đây là giá trị tối ưu trong số lượng người dùng tương tự để gợi ý phim. Khi **n\_users** tăng vượt quá 7, hiệu suất giảm dần. Điều này có thể do việc đưa quá nhiều người dùng tương đồng vào làm cho các gợi ý trở nên phân tán, giảm độ chính xác. Tương tự, tại **n\_users** thấp hơn 7 (ví dụ: 3 hoặc 4), hiệu suất cũng thấp hơn. Nguyên nhân có thể là do số lượng người dùng tương đồng quá ít không đủ để tìm ra các mẫu đánh giá đáng tin cậy.

#### 5.5 Đưa ra gợi ý với người dùng mới

**Vấn đề:**

- Khi có người dùng mới, hệ thống cần cập nhật người dùng vào dữ liệu và chạy lại quá trình tìm kiếm người dùng tương đồng.
- Quá trình này tiêu tốn thời gian và tài nguyên nếu dữ liệu lớn.

**Cách hoạt động:**

- Chạy lại hàm **KNN** để tính toán độ tương đồng giữa người dùng mới và các người dùng khác.

- Dưa ra gợi ý dựa trên các đánh giá mà người dùng mới đã cung cấp.
- Việc cập nhật này đảm bảo hệ thống sử dụng dữ liệu mới nhất để tạo ra các gợi ý phù hợp và chính xác.

#### **Ưu điểm:**

- Gợi ý được cá nhân hóa dựa trên dữ liệu thực tế của người dùng.
- Linh hoạt trong việc xử lý dữ liệu của người dùng mới.

#### **Nhược điểm:**

- Việc tính toán lại hàm KNN tiêu tốn thời gian nếu dữ liệu lớn.
- Hệ thống yêu cầu người dùng mới cung cấp đủ số lượng đánh giá để đảm bảo độ chính xác.

#### **Giải pháp khắc phục:**

- Sử dụng ANN (Approximate Nearest Neighbors) để tối ưu hóa quá trình tìm kiếm hàng xóm gần nhất.
- ANN giúp giảm đáng kể thời gian tính toán, đặc biệt hiệu quả trên các tập dữ liệu lớn và có độ đa chiều cao.

## **6 Approximate Nearest Neighbor (ANN) trong bài toán hệ thống gợi ý phim**

Approximate Nearest Neighbor (ANN) là một phương pháp hiệu quả được sử dụng để tìm kiếm các điểm gần nhất trong không gian nhiều chiều. Phương pháp này được áp dụng rộng rãi trong các hệ thống gợi ý nhằm tối ưu hóa tốc độ xử lý và tiết kiệm tài nguyên tính toán.

Trong bài toán gợi ý phim, ANN đóng vai trò quan trọng trong việc xác định nhanh chóng người dùng tương đồng hoặc các bộ phim có đặc điểm tương tự dựa trên vector đặc trưng được học từ dữ liệu. So với các phương pháp tìm kiếm chính xác, ANN cho phép đánh đổi độ chính xác nhỏ để đạt được tốc độ tìm kiếm nhanh hơn đáng kể.

### **6.1 Nguyên lý hoạt động**

- Thay vì thực hiện *tìm kiếm chính xác* các láng giềng gần nhất trong không gian vector, vốn yêu cầu chi phí tính toán lớn và tiêu tốn nhiều thời gian, ANN chấp nhận tìm kiếm gần đúng với một mức sai lệch nhỏ nhưng cải thiện đáng kể tốc độ xử lý.
- Trong các hệ thống gợi ý phim, **FAISS** (*Facebook AI Similarity Search*) là một thư viện phổ biến để triển khai ANN. FAISS cung cấp các thuật toán như:
  - **Chia cụm** (Clustering): Phân nhóm các vector để giảm không gian tìm kiếm.
  - **Lượng tử hóa** (Quantization): Nén và biểu diễn vector nhằm giảm kích thước bộ nhớ.
  - **Điều hướng dữ liệu** (Navigable Search): Tối ưu hóa quy trình tìm kiếm các vector gần nhất.

### **Ưu điểm của ANN:**

- Tốc độ tìm kiếm nhanh hơn đáng kể so với các phương pháp truyền thống.
- Khả năng mở rộng tốt cho các tập dữ liệu lớn và đa chiều.

### **Hạn chế của ANN:**

- Độ chính xác của kết quả tìm kiếm có thể bị giảm nhẹ so với phương pháp tìm kiếm chính xác.

## **6.2 Giới thiệu về FAISS**

**FAISS (Facebook AI Similarity Search)** là một thư viện được thiết kế để tìm kiếm và tính toán độ tương đồng giữa các vector lớn một cách hiệu quả và nhanh chóng. Trong hệ thống gợi ý, FAISS được sử dụng để tăng tốc độ xử lý và tìm kiếm các người dùng hoặc mục nội dung tương đồng khi áp dụng *Collaborative Filtering*.

**FAISS Index** là cấu trúc dữ liệu được sử dụng để tối ưu hóa việc tìm kiếm gần đúng (*Approximate Nearest Neighbor - ANN*) trên các vector lớn.

### **Ưu điểm của FAISS Index:**

- Tốc độ tìm kiếm nhanh với dữ liệu lớn.
- Tiết kiệm bộ nhớ và tài nguyên tính toán.
- Hỗ trợ các thuật toán Approximate Nearest Neighbor (ANN) như IVF, PQ và HNSW.

### **Các loại FAISS Index phổ biến:**

- **Flat Index:** Tìm kiếm chính xác nhưng tốn thời gian trên tập dữ liệu lớn.
- **IVF (Inverted File Index):** Chia không gian thành các cụm để tăng tốc độ tìm kiếm gần đúng.
- **PQ (Product Quantization):** Giảm kích thước vector để tiết kiệm bộ nhớ.
- **HNSW (Hierarchical Navigable Small World):** Tìm kiếm nhanh với đồ thị điều hướng.

## **6.3 Quy trình áp dụng FAISS để tìm Top n-Phim gợi ý**

### **Bước 1: Tạo FAISS Index và tìm k-người dùng tương tự**

- **Dữ liệu đầu vào:** Ma trận đặc trưng của người dùng (user x movie) dưới dạng *prediction\_array*.
- **Xây dựng FAISS Index:** FAISS sử dụng khoảng cách Euclidean (L2) để tìm kiếm người dùng tương tự:

$$\text{distance}(u, v) = \|\mathbf{p}_u - \mathbf{p}_v\|_2^2$$

- $\mathbf{p}_u, \mathbf{p}_v$ : Vector đặc trưng của người dùng  $u$  và  $v$ .
- Khoảng cách càng nhỏ thì người dùng càng tương đồng.

- **Chuyển đổi khoảng cách thành độ tương đồng:** Độ tương đồng giữa người dùng  $u$  và  $v$  được tính như sau:

$$\text{similarity}(u, v) = \frac{1}{1 + \text{distance}(u, v)}$$

### Bước 2: Lọc đánh giá của người dùng tương tự

- Lấy các đánh giá từ danh sách  $k$ -người dùng tương tự với điều kiện:

$$\text{rating} \geq \text{min\_rating}.$$

- Loại bỏ các bộ phim mà người dùng hiện tại đã xem để tránh gợi ý trùng lặp.

### Bước 3: Dự đoán điểm đánh giá cho từng bộ phim

Điểm dự đoán  $\hat{r}_{ui}$  của người dùng  $u$  cho bộ phim  $i$  được tính dựa trên trọng số tương đồng của người dùng tương tự:

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

Trong đó:

- $N_k(u)$ : Tập  $k$ -người dùng tương tự với người dùng  $u$ .
- $r_{vi}$ : Điểm đánh giá của người dùng  $v$  cho bộ phim  $i$ .
- $\text{sim}(u, v)$ : Độ tương đồng giữa người dùng  $u$  và  $v$ .

### Bước 4: Sắp xếp và trả về Top n-Phim gợi ý

- Sắp xếp các bộ phim theo điểm dự đoán giảm dần.
- Chọn ra Top  $n$ -phim có điểm số cao nhất để gợi ý.

**Kết quả cuối cùng:** Danh sách các bộ phim được gợi ý cho người dùng dưới dạng:

[movie\_id, final\_score]

Trong đó, `final_score` là điểm số dự đoán được tính toán dựa trên trọng số độ tương đồng và các đánh giá của người dùng tương tự.

## 6.4 Đánh giá

### 6.4.1 Phương pháp đánh giá

Quá trình đánh giá hiệu suất của thuật toán Facebook AI Similarity Search (FAISS) được thực hiện thông qua các bước và chỉ số cụ thể như sau:

#### Tập dữ liệu và phương pháp đánh giá:

- Dữ liệu đánh giá từ người dùng được sử dụng toàn bộ để tính toán độ tương đồng giữa các người dùng và đưa ra gợi ý dựa trên kết quả của  $k$ -người dùng gần nhất.

**Quy trình đánh giá:** Các chỉ số được sử dụng để đánh giá hiệu suất bao gồm:

- **Hit Ratio at K (HR@K):** Kiểm tra xem phim tích cực có nằm trong danh sách top  $K$  phim được hệ thống gợi ý hay không:

$$HR = \begin{cases} 1, & \text{nếu phim tích cực nằm trong top } K \\ 0, & \text{nếu ngược lại.} \end{cases}$$

- **Normalized Discounted Cumulative Gain (NDCG@K):** Dánh giá độ quan trọng của vị trí phim tích cực trong danh sách top  $K$  phim gợi ý, sử dụng công thức:

$$\text{NDCG@K} = \frac{1}{\log_2(\text{rank} + 1)}$$

Trong đó:

- **rank:** Vị trí của phim tích cực trong danh sách top  $K$ .

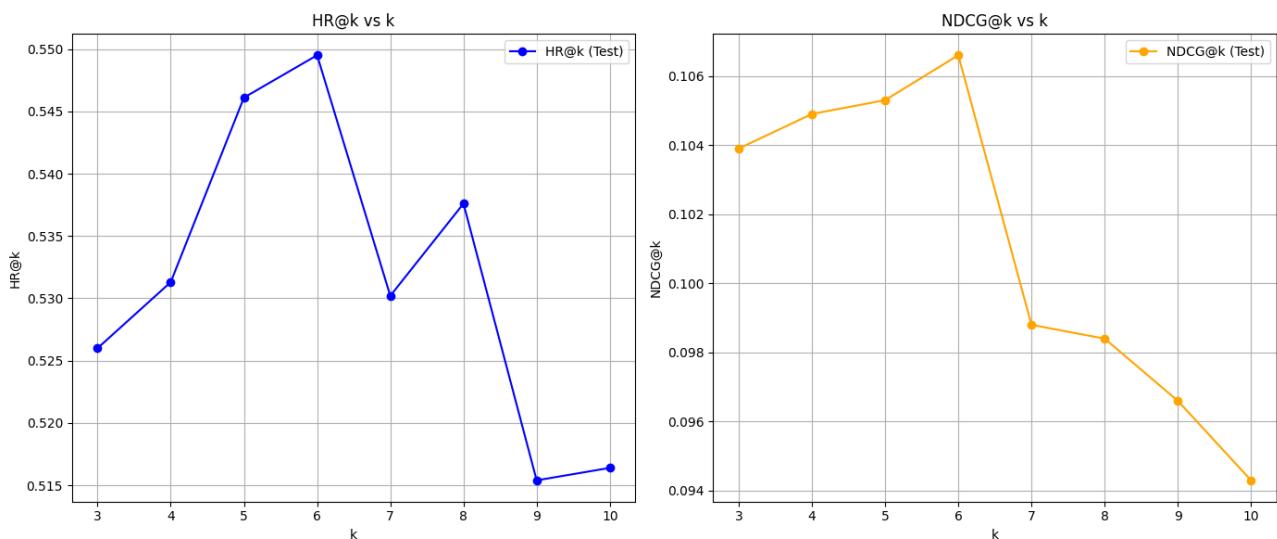
#### 6.4.2 Các tham số trong User-based Collaborative Filtering using FAISS

- **k:** Số lượng người dùng tương tự (nearest neighbors) mà hệ thống sẽ tìm kiếm để so sánh với người dùng hiện tại.
- **rec\_top\_n:** Số lượng phim tối đa mà hệ thống sẽ gợi ý cho người dùng hiện tại.
- **min\_rating:** Mức đánh giá tối thiểu để xem xét một bộ phim trong dữ liệu. Chỉ những bộ phim có đánh giá lớn hơn hoặc bằng giá trị này từ những người dùng tương tự mới được tính toán.

#### 6.4.3 Kết quả thử nghiệm các tham số

Nhóm đã tiến hành thử nghiệm trên tham số quan trọng trong bài toán và đánh giá sự thay đổi kết quả khi thay đổi các tham số:  $k$ .

**Đối với tham số k:**



Hình 5: Hiệu suất của mô hình với các giá trị khác nhau của  $k$ .

Khi đánh giá trên tập test, HR@K và NDCG@K đạt giá trị cao nhất tại  $k=6$ . Đây là giá trị tối ưu của số lượng người dùng tương đồng được xem xét để gợi ý phim. Khi  $k$  tăng vượt quá

6, hiệu suất bắt đầu giảm. Nguyên nhân có thể do việc xem xét quá nhiều người dùng tương đồng khiến gợi ý trở nên không tập trung, giảm tính chính xác của hệ thống. Ngược lại, tại các giá trị  $k$  nhỏ hơn (ví dụ:  $k=3$  hoặc  $k=4$ ), hiệu suất cũng thấp hơn do số lượng người dùng tương đồng không đủ để cung cấp thông tin đánh giá có giá trị cho gợi ý. Điều này chứng tỏ rằng việc chọn đúng số lượng người dùng tương đồng là rất quan trọng để tối ưu hóa hiệu quả của hệ thống gợi ý.

## 6.5 Đưa ra gợi ý với người dùng mới

### Đầu vào:

- **new\_user\_data:** Tập dữ liệu tương tác của người dùng mới, thường bao gồm khoảng 10 phim mà họ đã xem.
- **rating:** Trọng số lấy từ đánh giá của người dùng mới đối với từng bộ phim họ đã xem.

### Quy trình thêm người dùng mới:

- Tính toán vector đặc trưng cho người dùng mới dựa trên các đánh giá mà họ cung cấp.
- Thêm vector của người dùng mới vào FAISS Index bằng phương thức `add()` mà không cần xây dựng lại toàn bộ chỉ mục.
- Sử dụng FAISS để tìm *k*-người dùng tương đồng nhất với người dùng mới. FAISS tìm kiếm nhanh chóng các vector gần nhất trong chỉ mục đã cập nhật.
- Gợi ý các bộ phim mà những người dùng tương đồng đã đánh giá cao, đồng thời loại bỏ các bộ phim mà người dùng mới đã xem.

### Ưu điểm:

- **Cập nhật nhanh chóng:** FAISS cho phép thêm vector mới vào Index mà không cần tái xây dựng toàn bộ cấu trúc dữ liệu.
- **Tìm kiếm hiệu quả:** FAISS sử dụng tìm kiếm gần đúng (ANN) giúp tăng tốc độ xử lý, phù hợp với dữ liệu lớn.
- **Gợi ý cá nhân hóa:** Dựa trên dữ liệu thực tế của người dùng mới, đảm bảo độ chính xác và phù hợp với sở thích.
- **Linh hoạt:** Có thể áp dụng ngay cả khi người dùng mới chỉ cung cấp một số đánh giá nhỏ.

### Nhược điểm:

- **Độ chính xác xấp xỉ:** FAISS sử dụng tìm kiếm gần đúng (ANN), đôi khi có sai số nhỏ trong kết quả tương đồng.
- **Cold Start:** Nếu người dùng mới không cung cấp đủ đánh giá, hệ thống phải dựa vào gợi ý phổ biến, giảm tính cá nhân hóa.
- **Cần cập nhật định kỳ:** Dù việc thêm vector nhanh chóng, hệ thống vẫn cần làm mới FAISS Index định kỳ để đảm bảo hiệu quả tìm kiếm.

## 7 Thủ nghiệm hệ thống trên các kịch bản khác nhau

Nhóm đã tiến hành thử nghiệm trên 3 kịch bản khác nhau để đánh giá hiệu quả của các mô hình gợi ý với các mức độ dữ liệu đầu vào khác nhau từ người dùng. Mục tiêu của thử nghiệm này là xác định xem lượng dữ liệu tương tác đầu vào ảnh hưởng như thế nào đến hiệu suất của các mô hình (Content-based, KNN và NCF) dựa trên các chỉ số HR và NDCG.

### 7.1 Mô tả các kịch bản thử nghiệm

Các kịch bản thử nghiệm được định nghĩa dựa trên số lượng phim mà người dùng đã xem và được cung cấp làm đầu vào cho mô hình:

- **Kịch bản 1:** Người dùng đã xem 1 phim.
- **Kịch bản 2:** Người dùng đã xem 5 phim.
- **Kịch bản 3:** Người dùng đã xem 10 phim.

Trong mỗi kịch bản, dữ liệu đầu vào bao gồm lịch sử tương tác của người dùng (số phim đã xem) được sử dụng để huấn luyện và dự đoán. Hiệu suất của mô hình được đánh giá bằng cách tính toán HR và NDCG dựa trên khả năng gợi ý chính xác 5 phim tiếp theo mà người dùng thực sự đã xem.

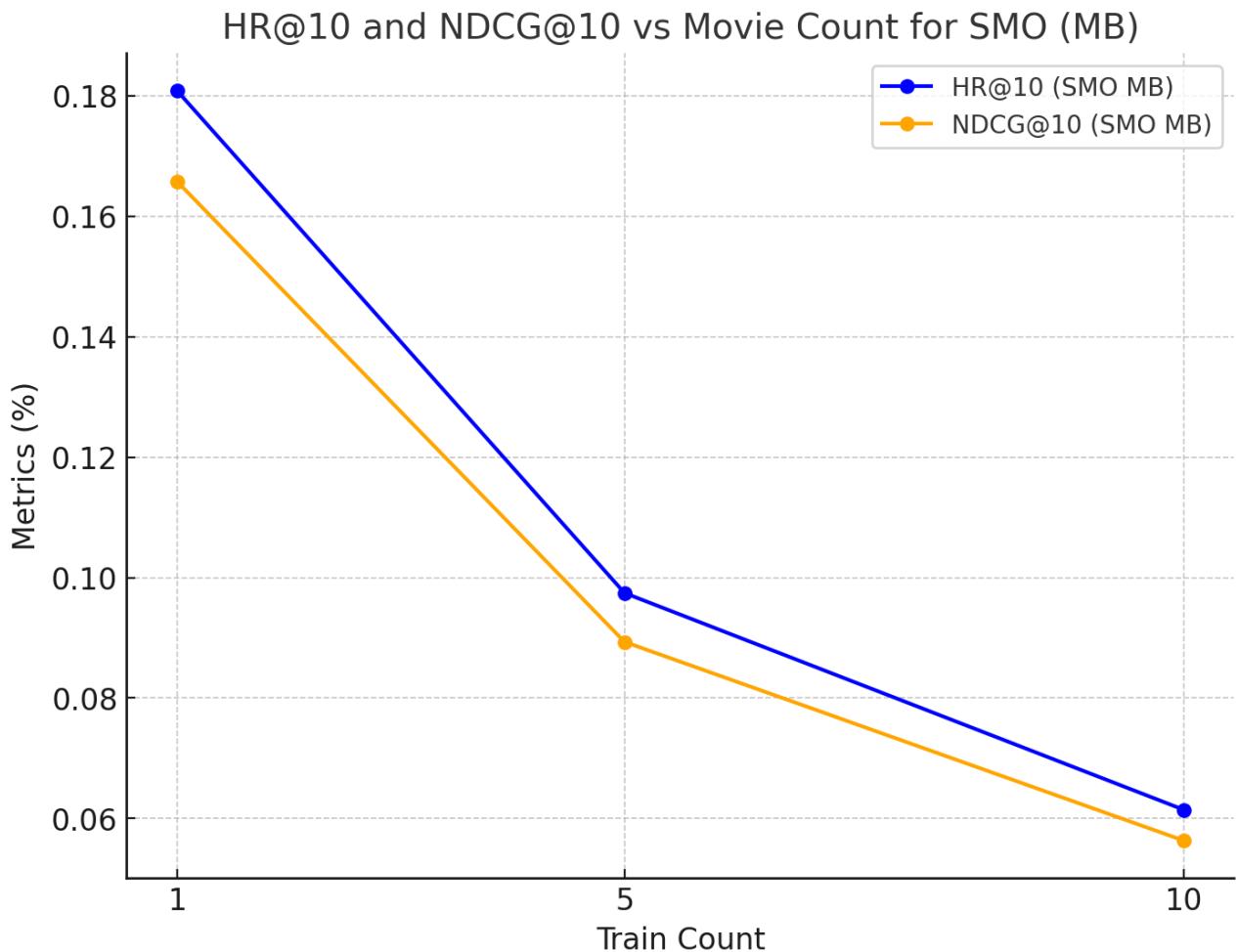
### 7.2 Quy trình thử nghiệm

Quy trình thử nghiệm được thực hiện như sau:

1. Thu thập dữ liệu tương tác từ tập dữ liệu đầu vào, chọn các người dùng có số lượng tương tác đủ để thử nghiệm cả 3 kịch bản.
2. Chia dữ liệu tương tác của mỗi người dùng thành hai phần:
  - Phần đầu: Sử dụng 1, 5, hoặc 10 phim làm dữ liệu đầu vào cho mô hình.
  - Phần sau: Sử dụng 5 phim tiếp theo làm tập kiểm tra (ground truth) để đánh giá hiệu suất.
3. Sử dụng lần lượt các mô hình đã kể trên để gợi ý danh sách các phim phù hợp cho mỗi người dùng.
4. Tính toán HR và NDCG để đánh giá độ chính xác của danh sách gợi ý so với tập ground truth.

### 7.3 Kết quả thử nghiệm với mô hình Content-based Filtering với phương pháp Same Metadata Only và Metadata-Based (SMO-MB)

Hình 6 minh họa kết quả của phương pháp SMO (MB) khi thử nghiệm trên các kịch bản khác nhau về số lượng phim đã xem được cung cấp làm đầu vào. Cụ thể, các kịch bản bao gồm người dùng đã xem 1 phim, 5 phim, và 10 phim (Phim đầu vào cho mô hình là phim có rating cao nhất của người dùng). Hiệu suất của phương pháp được đo lường bằng hai chỉ số HR@10 và NDCG@10.



Hình 6: Hiệu suất của phương pháp SMO (MB) với các số lượng phim đã xem khác nhau.

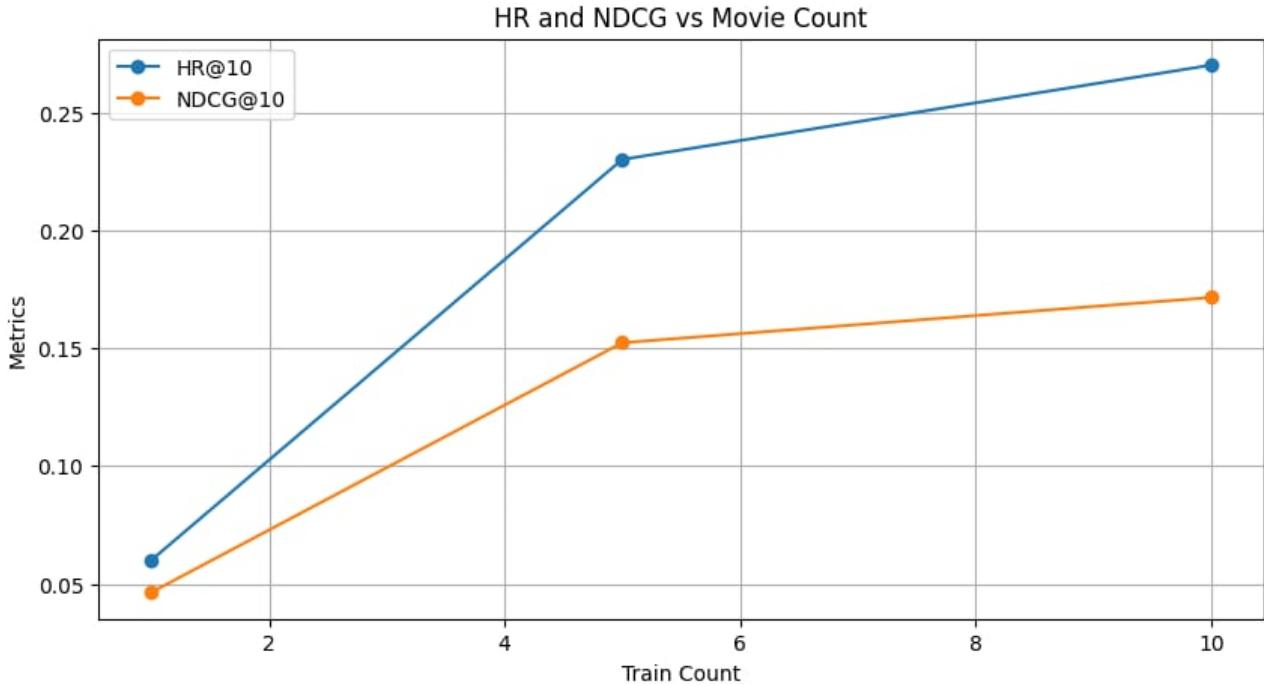
#### Nhận xét:

- Khi đầu vào là 1 phim:** HR@10 và NDCG@10 đều đạt giá trị cao nhất, lần lượt là 18% và 16.5%. Điều này cho thấy rằng với lượng thông tin ít nhưng cụ thể, phương pháp SMO (MB) có thể gợi ý các phim phù hợp dựa trên nội dung đã xem.
- Khi đầu vào là 5 phim:** Cả HR@10 và NDCG@10 đều giảm đáng kể, lần lượt xuống 12% và 10%. Điều này phản ánh rằng khi lượng dữ liệu đầu vào tăng lên, việc dự đoán các sở thích cụ thể trở nên khó khăn hơn do thông tin nội dung có thể không đủ để bao quát tất cả sở thích của người dùng.
- Khi đầu vào là 10 phim:** Hiệu suất tiếp tục giảm, với HR@10 đạt 5% và NDCG@10 đạt 3%. Điều này cho thấy rằng phương pháp Content-Based Filtering, như SMO (MB), không tận dụng được tốt dữ liệu đầu vào lớn do thiếu khả năng khai thác tương quan giữa người dùng.

Kết quả này chỉ ra rằng hiệu suất của phương pháp Content-Based Filtering, đặc biệt là SMO (MB), giảm dần khi lượng dữ liệu đầu vào tăng lên. Điều này có thể là do phương pháp này chỉ dựa vào thông tin nội dung của các phim đã xem, không tận dụng được các dữ liệu tương tác phức tạp hơn giữa các người dùng hoặc các mối quan hệ trong hệ thống.

## 7.4 Kết quả thử nghiệm với mô hình NeuMF

Hình 7 minh họa kết quả của mô hình NeuMF khi thử nghiệm trên các kịch bản khác nhau về số lượng phim đã xem được cung cấp làm đầu vào. Cụ thể, các kịch bản bao gồm người dùng đã xem 1 phim, 5 phim, và 10 phim. Hiệu suất của mô hình được đo lường bằng hai chỉ số HR@10 và NDCG@10.



Hình 7: Hiệu suất của mô hình NeuMF với các số lượng phim đã xem khác nhau.

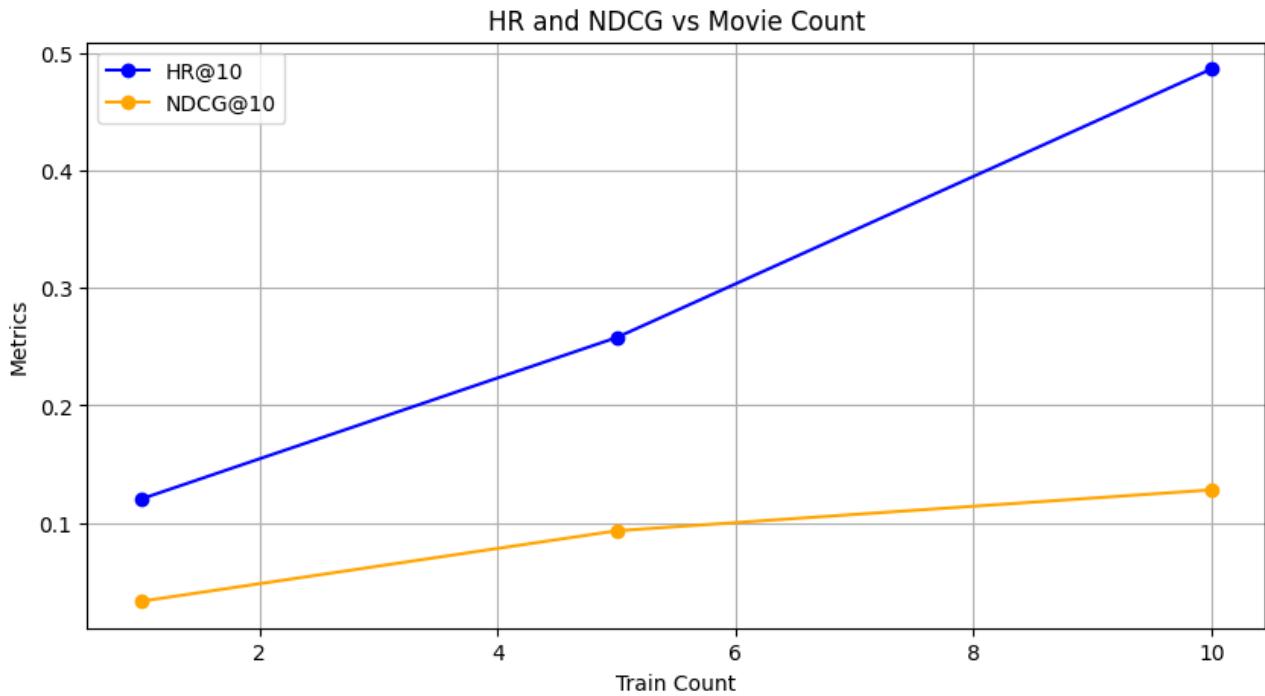
**Nhận xét:**

- **Khi đầu vào là 1 phim:** HR@10 và NDCG@10 đều đạt giá trị rất thấp, cho thấy rằng lượng dữ liệu đầu vào không đủ để mô hình học được các đặc trưng cần thiết để gợi ý chính xác.
- **Khi đầu vào là 5 phim:** Cả HR@10 và NDCG@10 đều tăng đáng kể so với trường hợp 1 phim, cho thấy việc tăng lượng dữ liệu đầu vào giúp cải thiện hiệu suất của mô hình.
- **Khi đầu vào là 10 phim:** Hiệu suất tiếp tục cải thiện, đạt giá trị cao nhất ở cả HR@10 và NDCG@10. Điều này khẳng định rằng việc cung cấp nhiều thông tin đầu vào hơn giúp mô hình học tốt hơn các sở thích của người dùng.

Kết quả này chứng minh rằng hiệu suất của mô hình NeuMF phụ thuộc mạnh mẽ vào lượng dữ liệu tương tác đầu vào. Việc cung cấp nhiều dữ liệu hơn cho mô hình là yếu tố quan trọng để cải thiện chất lượng gợi ý.

## 7.5 Kết quả thử nghiệm với phương pháp User-based Collaborative Filtering sử dụng KNN

Hình 8 minh họa kết quả của phương pháp User-based Collaborative Filtering sử dụng KNN khi thử nghiệm trên các kịch bản khác nhau về số lượng phim đã xem được cung cấp làm đầu vào. Cụ thể, các kịch bản bao gồm người dùng đã xem 1 phim, 5 phim, và 10 phim. Hiệu suất của mô hình được đo lường bằng hai chỉ số HR@10 và NDCG@1



Hình 8: Hiệu suất của mô hình với các kịch bản khác nhau về số lượng phim đầu vào.

#### Nhận xét:

- **Khi đầu vào là 1 phim:** HR@10 và NDCG@10 đều đạt giá trị rất thấp, điều này cho thấy rằng lượng dữ liệu đầu vào không đủ để mô hình học được các đặc trưng cần thiết để gợi ý chính xác. Điều này phản ánh việc thiếu thông tin tương tác giữa người dùng và sản phẩm.
- **Khi đầu vào là 5 phim:** Cả HR@10 và NDCG@10 đều tăng đáng kể so với trường hợp 1 phim, điều này cho thấy rằng việc tăng lượng dữ liệu đầu vào giúp cải thiện hiệu suất của mô hình. Lượng thông tin tương tác lớn hơn đã giúp mô hình nắm bắt tốt hơn sở thích của người dùng.
- **Khi đầu vào là 10 phim:** Hiệu suất tiếp tục cải thiện, đạt giá trị cao nhất ở cả HR@10 và NDCG@10. Điều này khẳng định rằng việc cung cấp dữ liệu đầu vào đầy đủ vào mô hình giúp mô hình học tốt hơn các đặc trưng về sở thích của người dùng, từ đó nâng cao chất lượng gợi ý.

## 7.6 Kết luận và nhận xét

Kết quả thử nghiệm cho thấy hai phương pháp gợi ý, Content-Based Filtering và Collaborative Filtering, có các đặc điểm và xu hướng hiệu suất khác nhau khi lượng dữ liệu đầu vào của người dùng thay đổi.

**Content-Based Filtering:** Hiệu suất của phương pháp Content-Based Filtering, cụ thể là SMO (MB), giảm dần khi lượng dữ liệu đầu vào tăng lên. Ở kịch bản với 1 phim đầu vào, phương pháp đạt hiệu suất cao nhất, nhờ khả năng dự đoán dựa trên nội dung cụ thể của các phim đã xem. Tuy nhiên, khi lượng phim đầu vào tăng lên, thông tin nội dung trở nên khó bao quát tất cả sở thích của người dùng, dẫn đến giảm hiệu suất gợi ý. Điều này chỉ ra rằng Content-Based Filtering hoạt động tốt hơn khi thông tin đầu vào ít nhưng có tính cụ thể cao (trường hợp cold-start).

**Collaborative Filtering:** Ngược lại, hiệu suất của phương pháp Collaborative Filtering, như NeuMF và KNN tăng dần khi lượng dữ liệu đầu vào tăng lên. Với dữ liệu tương tác phong phú hơn (kịch bản 5 và 10 phim), mô hình có thể khai thác tốt hơn mối quan hệ giữa người dùng và các mục được gợi ý, giúp cải thiện đáng kể hiệu suất. Điều này cho thấy Collaborative Filtering phụ thuộc mạnh mẽ vào dữ liệu tương tác và hoạt động hiệu quả hơn khi lượng dữ liệu đầu vào đủ lớn.

**Nhận xét tổng quan:** Kết quả thử nghiệm nhấn mạnh tầm quan trọng của việc lựa chọn phương pháp gợi ý phù hợp với lượng dữ liệu đầu vào:

- Với lượng dữ liệu ít, Content-Based Filtering là lựa chọn khả thi, nhờ khả năng gợi ý dựa trên nội dung cụ thể.
- Khi dữ liệu đầu vào phong phú hơn, Collaborative Filtering trở thành lựa chọn tối ưu, tận dụng được mối quan hệ giữa người dùng và các mục đã tương tác.

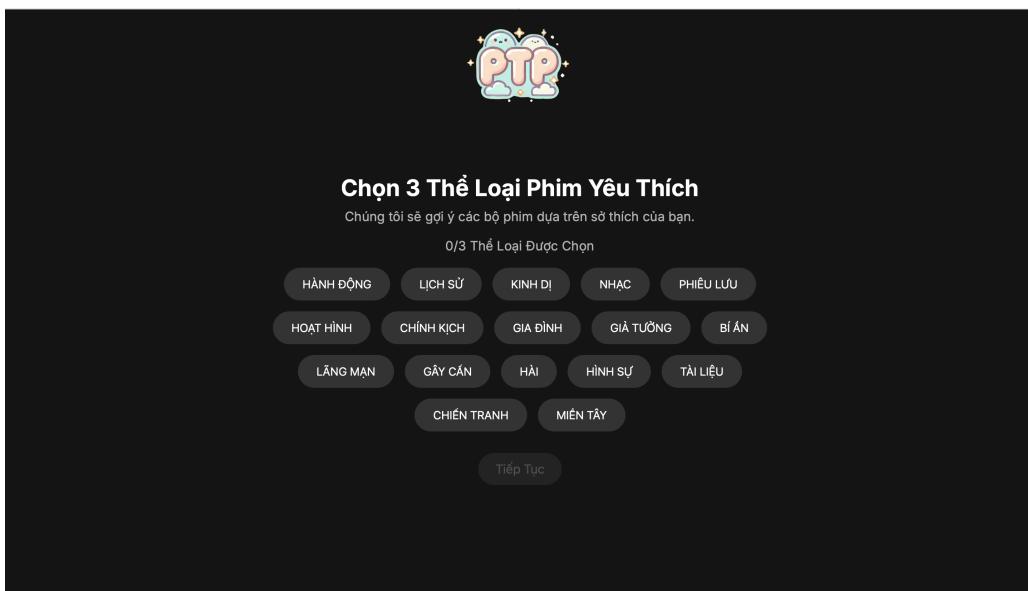
Tổng quan, việc kết hợp cả hai phương pháp trong một hệ thống lai (*hybrid system*) có thể mang lại hiệu quả cao hơn, tận dụng được ưu điểm của cả Content-Based và Collaborative Filtering để cải thiện chất lượng gợi ý cho các trường hợp đa dạng.

## 8 Ảnh minh họa và Hướng dẫn sử dụng Sản phẩm

Phần này giới thiệu các hình ảnh giao diện của sản phẩm để minh họa cho các tính năng chính. Dưới đây là các bước tương tác và giao diện của hệ thống gợi ý phim:

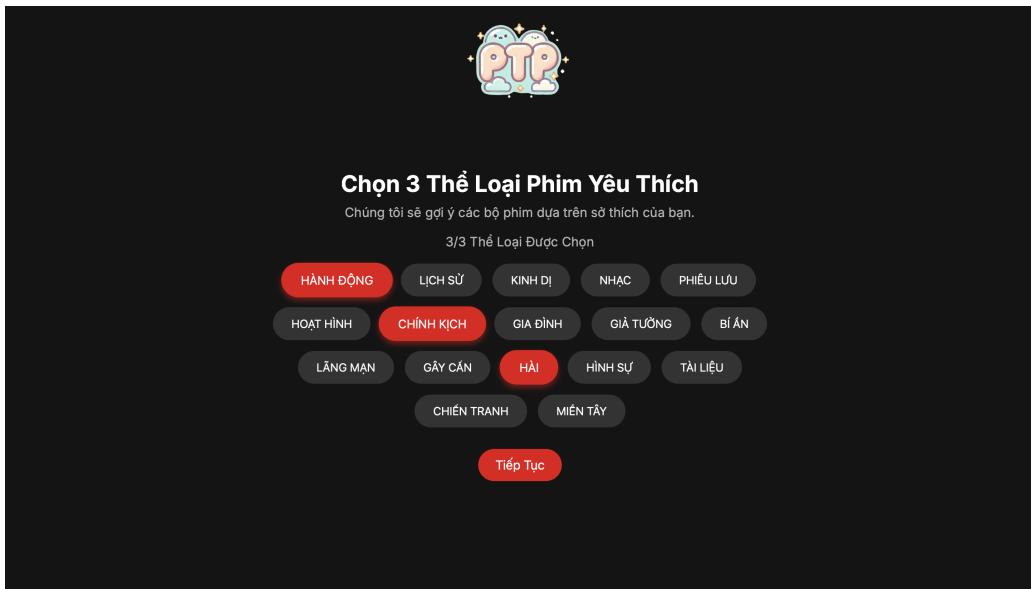
### 8.1 Chọn Thể loại Phim

Người dùng bắt đầu bằng việc chọn các thể loại phim yêu thích trên trang chọn thể loại. Đây là bước đầu tiên để cá nhân hóa hệ thống gợi ý.



Hình 9: Trang chọn thể loại phim (chưa chọn thể loại).

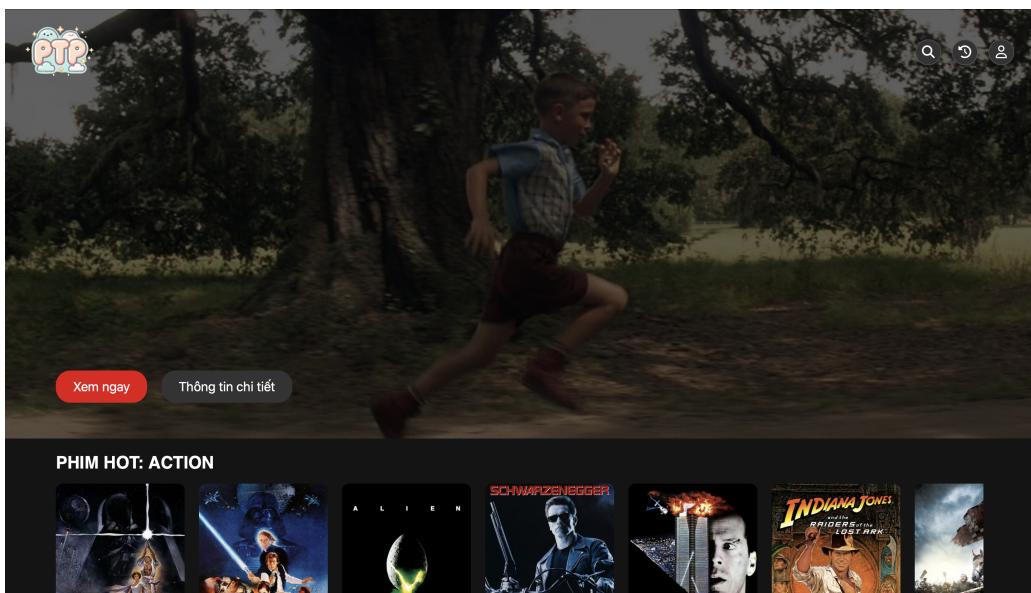
Sau khi chọn được 3 thể loại, giao diện sẽ cập nhật để phản ánh lựa chọn của người dùng.



Hình 10: Trang chọn thể loại phim (đã chọn 3 thể loại).

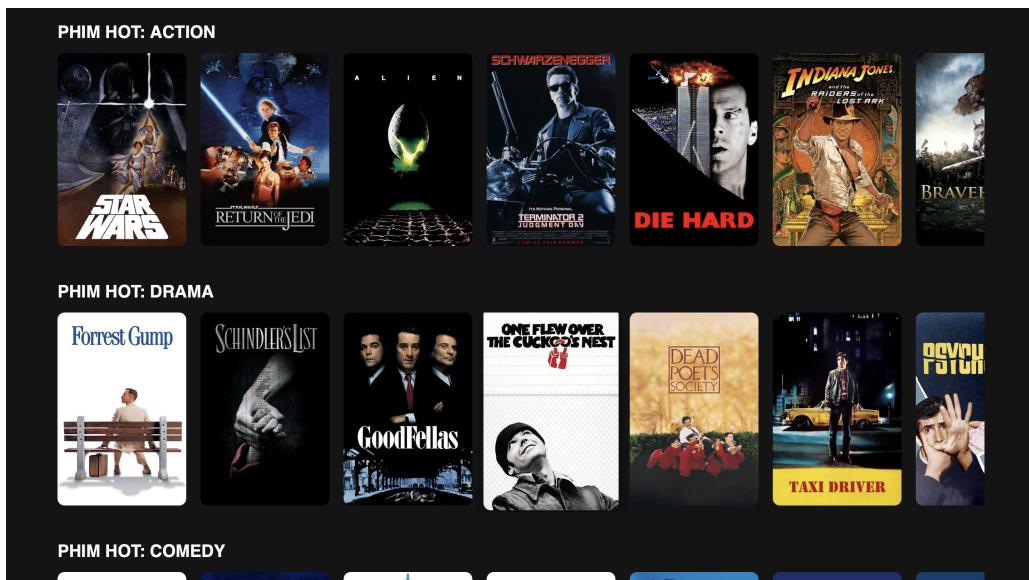
## 8.2 Màn Hình Chính

Giao diện màn hình chính hiển thị danh sách các bộ phim nổi bật và các mục gợi ý khác nhau.



Hình 11: Màn hình chính của ứng dụng.

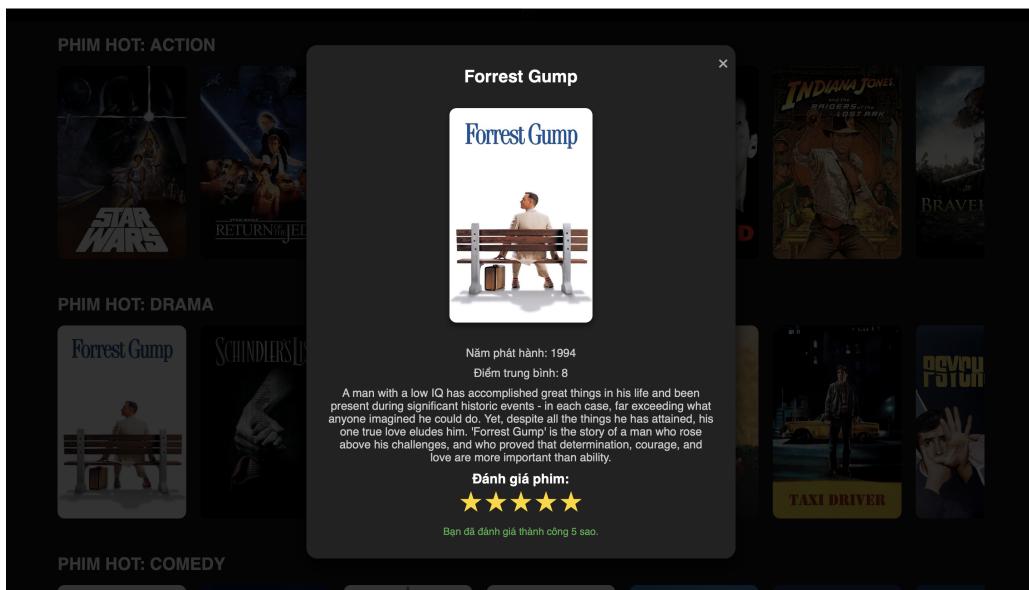
Khi người dùng cuộn xuống, mục gợi ý các bộ phim "Hot nhất theo thể loại" sẽ xuất hiện, cung cấp các đề xuất theo sở thích.



Hình 12: Gợi ý phim hot nhất theo thể loại.

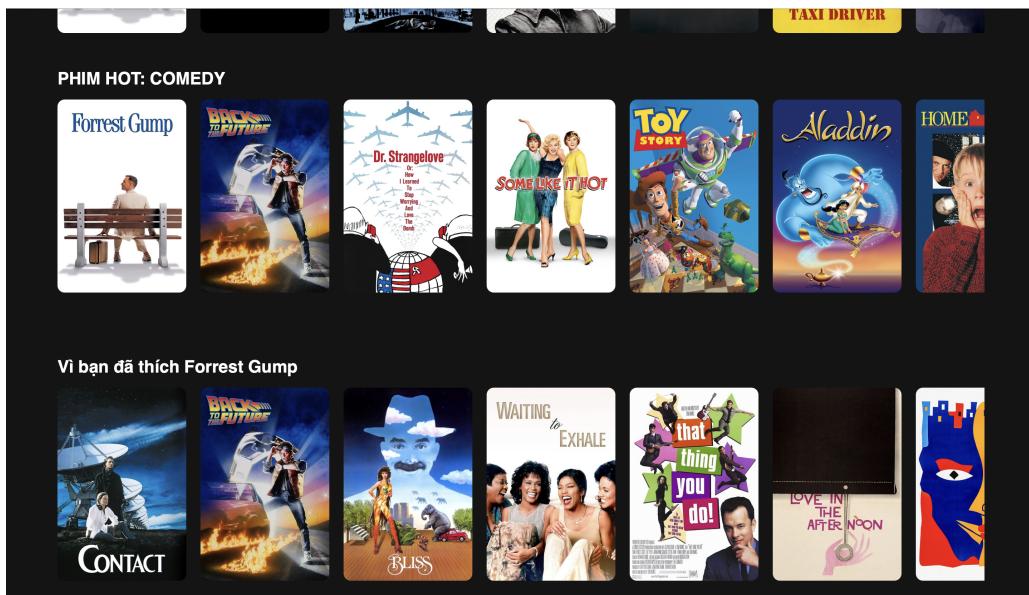
### 8.3 Thông Tin Chi Tiết Phim

Khi người dùng bấm vào một bộ phim, cửa sổ pop-up sẽ xuất hiện, hiển thị thông tin chi tiết về bộ phim đó.



Hình 13: Thông tin chi tiết của bộ phim.

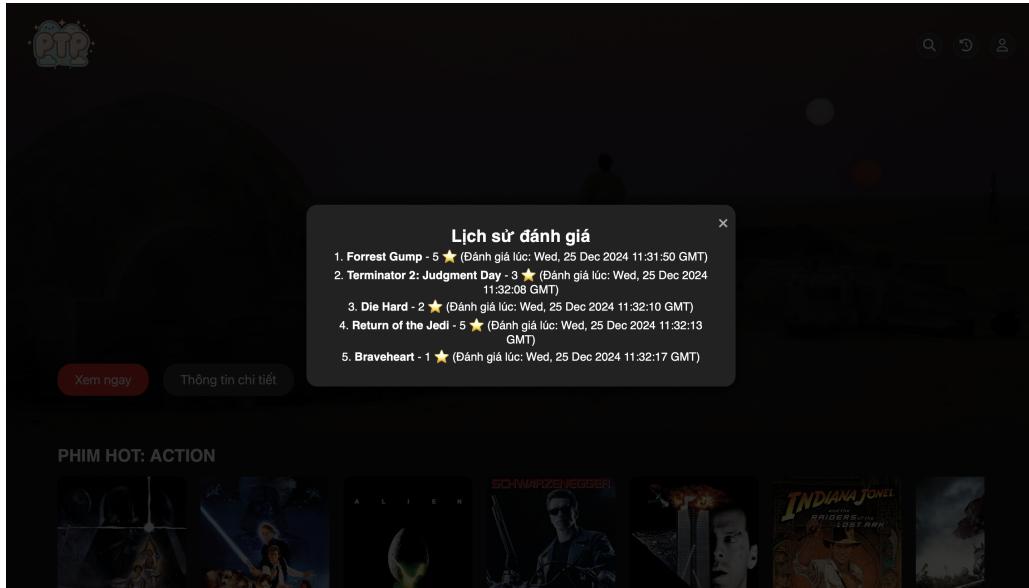
Nếu người dùng đánh giá bộ phim trên 3 sao, giao diện sẽ cập nhật để hiển thị các bộ phim tương tự dựa trên sở thích của họ (gợi ý dựa trên **Content-Based Filtering**).



Hình 14: Gợi ý phim tương tự sau khi đánh giá >3 sao.

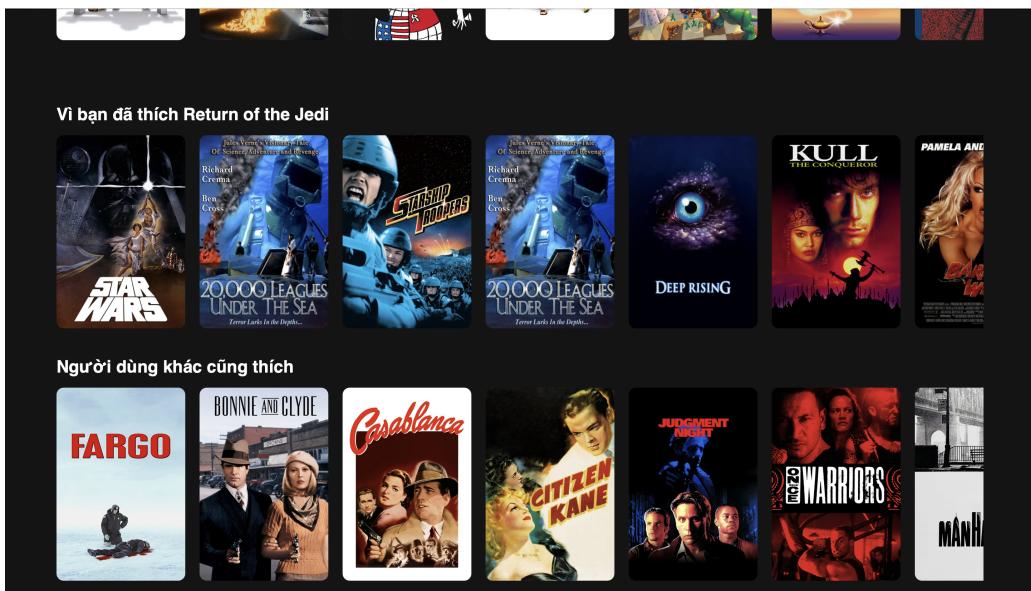
## 8.4 Lịch Sử Đánh Giá và Gợi Ý Mạng Người Dùng

Trang lịch sử hiển thị danh sách các bộ phim mà người dùng đã đánh giá, giúp họ dễ dàng quản lý và xem lại các phim yêu thích.



Hình 15: Lịch sử đánh giá 5 bộ phim.

Sau khi đánh giá 5 bộ phim, hệ thống sẽ hiển thị mục "Những người dùng khác đã thích", sử dụng phương pháp **Collaborative Filtering** để đề xuất các bộ phim mà những người dùng khác có sở thích tương tự đã đánh giá cao.



Hình 16: Gợi ý dựa trên sở thích của người dùng khác.

## 9 Tổng kết và Dự định tương lai

### 9.1 Tổng kết

Trong nghiên cứu này, các phương pháp gợi ý khác nhau như KNN, ANN và NCF đã được áp dụng và so sánh để cải thiện hệ thống gợi ý phim. Kết quả cho thấy:

- Phương pháp KNN và ANN đạt tỷ lệ HR cao hơn so với NCF, chứng minh hiệu quả trong việc gợi ý phim dựa trên người dùng tương đồng.
- Đã giải quyết được vấn đề gợi ý cho người dùng mới (chưa từng được huấn luyện hoặc chưa được thấy), tuy nhiên tỷ lệ gợi ý cho nhóm người dùng này vẫn chỉ đạt mức trung bình.

### 9.2 Dự định tương lai

Hướng phát triển tiếp theo của nhóm cho hệ thống gợi ý bao gồm:

- Tối ưu hóa thuật toán ANN bằng cách kết hợp với các kỹ thuật giảm chiều dữ liệu để tăng tốc độ xử lý.
- Nâng cao độ chính xác của NCF thông qua việc điều chỉnh siêu tham số và bổ sung các đặc trưng mới vào mô hình.
- Phát triển các phương pháp hybrid kết hợp giữa KNN, ANN và NCF để tận dụng ưu điểm của từng phương pháp.
- Tiếp tục mở rộng và thử nghiệm trên các tập dữ liệu thực tế lớn hơn nhằm đảm bảo tính khả thi và hiệu quả của hệ thống.