ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA

BÁO CÁO BÀI TẬP LỚN

KHO DỮ LIỆU VÀ HỆ HỖ TRỢ QUYẾT ĐỊNH - CO4031

XÂY DỰNG HỆ THỐNG ĐỀ XUẤT PHIM HOẠT HÌNH ANIME SỬ DỤNG COLLABORATIVE FILTERING

Giảng viên hướng dẫn: ThS. Bùi Tiến Đức
Sinh viên thực hiện: Đỗ Văn Bâng - 2110813
Nguyễn Minh Lộc - 2110342
Thạch Đinh Hoàng Việt - 2115287
Võ Văn Dũng - 2110102
Hồ Trọng Nhân - 2111899
Giản Đình Thái - 2112278

Tp. Hồ Chí Minh, Tháng 11/2024

Mục lục

Danh sách hình ảnh

Danh sách bảng biểu

1 Giới thiệu đề tài

1.1 Đặt vấn đề

Trong thời đại công nghệ số, với sự phát triển không ngừng của ngành công nghiệp giải trí, số lượng nội dung số, bao gồm anime, ngày càng tăng nhanh. Điều này mang đến một thách thức lớn cho người dùng trong việc tìm kiếm và lựa chọn nội dung phù hợp với sở thích của mình. Các nền tảng xem anime như Crunchyroll, Funimation hay Netflix đã và đang sử dụng hệ thống gợi ý để cải thiện trải nghiệm người dùng. Tuy nhiên, việc xây dựng một hệ thống gợi ý chính xác và cá nhân hóa vẫn là một bài toán phức tạp do:

- **Sự đa dạng của nội dung:** Anime có nhiều thể loại khác nhau như hành động, phiêu lưu, tình cảm, khoa học viễn tưởng, mỗi thể loại lại phù hợp với một nhóm đối tượng cụ thể.
- **Dữ liệu lớn và không đồng nhất:** Các hệ thống cần xử lý khối lượng dữ liệu lớn từ đánh giá của người dùng, danh sách phim, và các thông tin liên quan khác.
- **Sở thích người dùng thay đổi theo thời gian:** Người dùng không phải lúc nào cũng yêu thích một thể loại cố định, điều này đòi hỏi hệ thống phải linh hoạt và thích nghi với những thay đổi trong hành vi của họ.

Trước thực tế đó, việc nghiên cứu và phát triển một hệ thống gợi ý anime dựa trên mô hình học sâu là rất cần thiết. Không chỉ giúp người dùng dễ dàng tiếp cận nội dung phù hợp mà còn nâng cao hiệu quả hoạt động kinh doanh của các nền tảng dịch vụ.

Bài toán này hướng tới việc sử dụng các phương pháp hiện đại như học sâu (deep learning), gợi ý dựa trên nội dung (content-based recommendation) và gợi ý dựa trên cộng đồng (collaborative filtering) để phân tích dữ liệu đánh giá và hành vi của người dùng. Qua đó, tạo ra một hệ thống gợi ý có khả năng cá nhân hóa cao, không chỉ phản ánh sở thích hiện tại mà còn có thể thích nghi với sự thay đổi và đa dạng trong thị hiếu của người dùng.

1.2 Mục tiêu nghiên cứu

Mục tiêu chính của nghiên cứu là xây dựng một hệ thống gợi ý anime cá nhân hóa dựa trên mô hình học sâu, với các mục tiêu cụ thể như sau:

- **Tối ưu hóa việc gợi ý:** Cung cấp danh sách các anime phù hợp nhất với sở thích cá nhân của từng người dùng, dựa trên lịch sử đánh giá và hành vi của họ.
- **Tìm kiếm tương tự:**
  - Xác định các anime có nội dung tương tự dựa trên khoảng cách vector trong không gian nhúng (embedding space).
  - Xác định những người dùng có hành vi đánh giá tương đồng, từ đó cải thiện khả năng gợi ý thông qua chia sẻ sở thích.
- **Dự đoán chính xác đánh giá:** Phát triển mô hình dự đoán điểm đánh giá của người dùng cho những anime mà họ chưa từng xem, từ đó hỗ trợ việc gợi ý các lựa chọn tiềm năng.
- **Cải thiện trải nghiệm người dùng:** Đảm bảo các gợi ý được đưa ra không chỉ chính xác mà còn đa dạng, giúp người dùng khám phá thêm các nội dung mới và mở rộng sở thích của họ.

1.3 Phương pháp nghiên cứu

Để đạt được các mục tiêu trên, nghiên cứu áp dụng quy trình tiếp cận bao gồm các phương pháp cụ thể sau:

- **Thu thập và xử lý dữ liệu:**
  - Thu thập dữ liệu đánh giá từ người dùng và thông tin về các anime từ các nguồn dữ liệu công khai như MyAnimeList.
  - Tiền xử lý dữ liệu để làm sạch, chuẩn hóa và mã hóa thông tin (embedding) nhằm chuẩn bị dữ liệu đầu vào cho mô hình.

- **Xây dựng mô hình gợi ý:**
  - Áp dụng kỹ thuật embedding để biểu diễn người dùng và anime trong không gian vector.
  - Thiết kế mô hình học sâu với các thành phần như lớp nhúng (embedding layers), tích vô hướng (dot product), và lớp dày (dense layer).
  - Sử dụng hàm kích hoạt sigmoid để biểu diễn xác suất người dùng yêu thích một anime.
- **Huấn luyện và tối ưu hóa mô hình:**
  - Huấn luyện mô hình trên tập dữ liệu đánh giá với hàm mất mát binary cross-entropy.
  - Áp dụng các kỹ thuật tối ưu như giảm tốc độ học theo chu kỳ (learning rate scheduler), lưu trữ trạng thái tốt nhất của mô hình (model checkpoint), và dừng sớm (early stopping) để đảm bảo quá trình hội tụ hiệu quả.
- **Các chiến lược gợi ý:**
  - **Tìm kiếm anime tương tự:** Dựa trên khoảng cách cosine giữa vector nhúng của các anime để xác định mức độ tương tự.
  - **Tìm kiếm người dùng tương tự:** Sử dụng vector nhúng của người dùng để tìm những người có hành vi đánh giá tương đồng.
  - **Dự đoán đánh giá anime chưa xem:** Áp dụng mô hình đã huấn luyện để dự đoán điểm đánh giá của người dùng cho những anime chưa từng được xem, từ đó xếp hạng và gợi ý.

2 Cơ sở lý thuyết

2.1 Kho dữ liệu

2.1.1 Định nghĩa

Kho dữ liệu (data warehouse) là một hệ thống lưu trữ dữ liệu quy mô lớn được thiết kế để hỗ trợ việc phân tích dữ liệu và lập báo cáo. Dữ liệu trong kho dữ liệu được thu thập từ nhiều nguồn khác nhau, ví dụ như hệ thống giao dịch, hệ thống bán hàng, hệ thống nhân sự, v.v., và được lưu trữ một cách có tổ chức để có thể truy cập và phân tích một cách hiệu quả.

2.1.2 Các đặc trưng của kho dữ liệu

- **Tính chủ đề:** Dữ liệu trong kho dữ liệu được tập trung vào một chủ đề cụ thể, ví dụ như bán hàng, khách hàng, sản phẩm, v.v.
- **Tính tích hợp:** Dữ liệu từ nhiều nguồn khác nhau được tích hợp thành một định dạng thống nhất, giúp loại bỏ sự trùng lặp và mâu thuẫn dữ liệu.
- **Tính thời gian:**Dữ liệu trong kho dữ liệu được lưu trữ theo thời gian, cho phép theo dõi xu hướng và thay đổi dữ liệu theo thời gian.
- **Tính phi biến động:** Dữ liệu trong kho dữ liệu không được cập nhật thường xuyên như dữ liệu trong hệ thống giao dịch. Dữ liệu chỉ được cập nhật định kỳ, ví dụ như hàng ngày, hàng tuần hoặc hàng tháng.
- **Tính truy cập:**Dữ liệu trong kho dữ liệu được truy cập bởi các nhà phân tích dữ liệu và chuyên gia kinh doanh để thực hiện phân tích và báo cáo.

2.1.3 Kiến trúc kho dữ liệu

Một kho dữ liệu thông thường sẽ có kiến trúc cơ bản tương đối giống nhau. Dưới đây là phân tích về các thành phần trong kiến trúc của một dataware house tiêu chuẩn: **Dữ liệu nguồn:** Bao gồm dữ liệu hoạt động từ các nguồn khác nhau trong tổ chức, chẳng hạn như:

- Cơ sở dữ liệu giao dịch (bán hàng, hàng tồn kho, khách hàng).
- Hệ thống CRM (Quản lý quan hệ khách hàng).
- Hệ thống ERP (Kế hoạch hóa nguồn lực doanh nghiệp).
- Các file log (nhật ký)

Khu vực trung gian (Staging Area): Đây là khu vực lưu trữ tạm thời, nơi dữ liệu thô từ các hệ thống nguồn được trích xuất, chuyển đổi và tải trước khi được tích hợp vào data warehouse.

- **Trích xuất:** Dữ liệu được lấy từ các hệ thống nguồn.
- **Chuyển đổi:** Dữ liệu được làm sạch, định dạng và có thể tổng hợp để đáp ứng các nhu cầu của data warehouse. Điều này có thể liên quan đến việc xử lý các giá trị thiếu, sửa lỗi không nhất quán và chuyển đổi kiểu dữ liệu.
- **Tải:** Dữ liệu đã được chuyển đổi được dàn dựng trong data warehouse.

Data Warehouse: Đây là kho lưu trữ trung tâm lưu trữ dữ liệu lịch sử được tích hợp. Dữ liệu thường được tổ chức bằng cách sử dụng các lược đồ được tối ưu hóa cho khối lượng công việc phân tích, chẳng hạn như lược đồ ngôi sao (star schema) hoặc lược đồ bông tuyết (snowflake schema).

- **Lược đồ ngôi sao:** Một lược đồ đơn giản hóa với bảng dữ liệu thực tế (fact table) trung tâm được bao quanh bởi các bảng chiều (dimension table). Bảng dữ liệu thực tế lưu trữ các phép đo (ví dụ: số tiền bán hàng), trong khi các bảng chiều chứa các thuộc tính mô tả (ví dụ: tên sản phẩm, vị trí khách hàng).
- **Lược đồ bông tuyết:** Một lược đồ được chuẩn hóa nhiều hơn, trong đó các bảng chiều được chia nhỏ thành các chiều phụ để có độ chi tiết tốt hơn và giảm trùng lặp dữ liệu.

Kho lưu trữ siêu dữ liệu (Metadata Repository): Lưu trữ thông tin về dữ liệu trong data warehouse, bao gồm định nghĩa, nguồn gốc dữ liệu (data lineage) và quyền truy cập. Nó giúp người dùng hiểu ý nghĩa và cấu trúc của dữ liệu. Cung cấp thông tin: Lớp này cung cấp cho người dùng quyền truy cập vào dữ liệu trong data warehouse để phân tích và báo cáo. Các công cụ và công nghệ được sử dụng bao gồm:

- Công cụ xử lý phân tích trực tuyến đa chiều (OLAP) để phân tích dữ liệu đa chiều.
- Công cụ trực quan hóa dữ liệu để tạo biểu đồ và bảng điều khiển.
- Công cụ khai thác dữ liệu để khám phá các mẫu và thông tin chi tiết.
- Công cụ báo cáo để tạo báo cáo tùy chỉnh.

2.2 Collaborative Filtering trong Machine learning

2.2.1 Định nghĩa

Collaborative Filtering (CF) là một kỹ thuật phổ biến trong lĩnh vực **Machine Learning** được sử dụng để xây dựng các hệ thống gợi ý (*Recommendation Systems*). Thay vì phân tích các đặc điểm của sản phẩm hoặc dịch vụ, CF tập trung vào dữ liệu tương tác giữa người dùng và các đối tượng (sản phẩm, dịch vụ) để đưa ra các gợi ý phù hợp. Phương pháp này dựa trên nguyên tắc: "Những người có hành vi hoặc sở thích giống nhau trong quá khứ có thể có lựa chọn tương tự trong tương lai."

2.2.2 Phân loại Collaborative Filtering

F thường được chia thành hai dạng chính, mỗi dạng có cách tiếp cận và ưu điểm riêng.

User-based Collaborative Filtering

- **Ý tưởng cốt lõi:** Dựa trên sự tương đồng giữa *người dùng*. Nếu hai người dùng có cùng sở thích hoặc hành vi tương tự nhau trong quá khứ, họ có khả năng sẽ tiếp tục chọn các sản phẩm hoặc dịch vụ giống nhau.
- **Cách hoạt động:**
  1. Tìm kiếm nhóm người dùng có sở thích gần giống nhau.
  2. Dựa trên nhóm này, đưa ra dự đoán và gợi ý các sản phẩm mà người dùng có thể thích.
- **Ví dụ thực tế:** Nếu người dùng A và người dùng B đều thích phim hành động, và B đã xem một bộ phim hành động mới, thì A có thể được gợi ý xem bộ phim đó.

Item-based Collaborative Filtering

- **Ý tưởng cốt lõi:** Dựa trên sự tương đồng giữa các *đối tượng (item)*. Nếu hai sản phẩm thường được chọn hoặc đánh giá cao bởi cùng một nhóm người, thì chúng có khả năng tương tự về sở thích và chức năng.
- **Cách hoạt động:**
  1. Tìm các cặp sản phẩm có điểm tương đồng cao dựa trên hành vi của người dùng.
  2. Gợi ý sản phẩm liên quan cho người dùng dựa trên các mặt hàng họ đã tương tác.
- **Ví dụ thực tế:** Trên Amazon, nếu nhiều người mua máy ảnh cũng mua thẻ nhớ SD, thì thẻ nhớ SD sẽ được gợi ý cho những người đang xem máy ảnh.

2.2.3 Quy trình Collaborative Filtering

1. **Thu thập dữ liệu:** CF dựa vào dữ liệu lịch sử về tương tác giữa người dùng và các sản phẩm. Loại dữ liệu này thường được biểu diễn dưới dạng **User-Item Matrix**, với các hàng đại diện cho người dùng và các cột đại diện cho sản phẩm. Giá trị trong ma trận có thể là:
   - **Định lượng:** Rating, số lần mua hàng.
   - **Nhị phân:** Có hoặc không tương tác.
   - **Định tính:** Thích, không thích.

2. **Tính toán sự tương đồng:** Dựa trên dữ liệu thu thập, hệ thống sẽ tính toán mức độ tương đồng giữa người dùng hoặc sản phẩm. Một số phương pháp phổ biến để đo lường sự tương đồng bao gồm:
   - **Cosine Similarity:** Đo góc giữa hai vector người dùng hoặc sản phẩm.
   - **Pearson Correlation Coefficient:** Đo mối quan hệ tuyến tính giữa hai tập dữ liệu.
   - **Jaccard Similarity:** Đo mức độ giao nhau giữa hai tập hợp.
3. **Dự đoán giá trị còn thiếu:** Hệ thống sử dụng thông tin tương đồng để dự đoán điểm số cho các mục trong ma trận mà người dùng chưa tương tác. Ví dụ, nếu người dùng 1 chưa đánh giá *Item C*, hệ thống sẽ dự đoán dựa trên đánh giá của những người dùng tương tự.
4. **Đưa ra gợi ý:** Dựa trên điểm số dự đoán, hệ thống sắp xếp các sản phẩm và đưa ra danh sách gợi ý cho người dùng. Thông thường, các sản phẩm có điểm cao nhất sẽ được ưu tiên hiển thị.

2.3 Một số khái niệm liên quan đến mô hình mạng đề xuất của nhóm

2.3.1 Embedding Layer

Lớp nhúng (Embedding Layer) ánh xạ các giá trị phân loại hoặc rời rạc (như ID) vào không gian vector liên tục thấp chiều hơn.

Ví dụ: Một danh sách 1.000.000 sản phẩm có thể được nhúng vào không gian 128 chiều:

$$\text{Embedding} : \mathbb{R}^N \to \mathbb{R}^d, \quad N \gg d$$

2.3.2 Dot Product Layer

Tính tích vô hướng giữa hai vector:

$$\text{Dot Product} = \vec{u} \cdot \vec{v} = \sum_{i=1}^{d} u_i v_i$$

Ý nghĩa: Đo lường mức độ tương đồng giữa các vector.

2.3.3 Flatten Layer

Flatten Layer: chuyển đổi tensor nhiều chiều thành vector 1 chiều. Ví dụ, một tensor kích thước 4×4 được làm phẳng thành một vector:

$$[4 \times 4] \to [1 \times 16]$$

2.3.4 Fully Connected Layer: Dense

Dense: Kết nối tất cả các nơ-ron trong lớp trước với lớp hiện tại:

$$y = f(Wx + b)$$

Trong đó:

- W: Trọng số (weight matrix)
- x: Đầu vào
- b: Hệ số bias
- f: Hàm kích hoạt (activation function)

2.3.5 Hàm mất mát: Binary Crossentropy

Hàm mất mát đo lường sự khác biệt giữa nhãn thực tế và xác suất dự đoán:

$$\text{Binary Crossentropy} = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

Trong đó:

- y_i: Nhãn thực tế (0 hoặc 1)
- $\hat{y}_i$: Xác suất dự đoán

2.3.6 Tối ưu hóa mô hình: Adam

Adam là thuật toán tối ưu hóa dựa trên Gradient Descent. Công thức:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\theta_t = \theta_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

Trong đó:

- m_t, v_t: Trung bình động của gradient và bình phương gradient
- $\hat{m}_t, \hat{v}_t$: Giá trị hiệu chỉnh
- η: Tốc độ học (learning rate)

2.3.7 Metrics: MAE & MSE

MAE (Mean Absolute Error):

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|$$

MSE (Mean Squared Error):

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

Ý nghĩa:

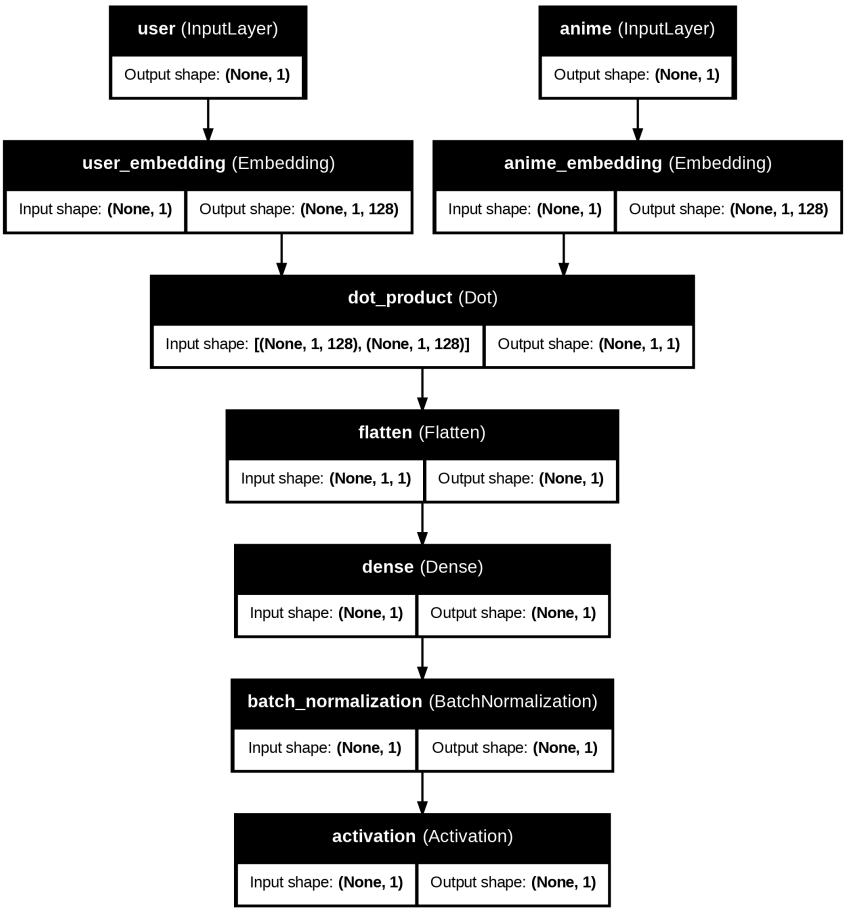
- MAE: Đo sai số trung bình tuyệt đối.
- MSE: Đo sai số trung bình bình phương, nhấn mạnh các sai số lớn.

3 Thực nghiệm

Các tài nguyên sử dụng:

- Ngôn ngữ lập trình: Python
- Môi trường thiết lập: GPU
- Thư viện sử dụng: TensorFlow, Keras, Pandas, Matplotlib.

Sơ đồ về kiến trúc mô hình mạng đề xuất:



Hình 1: *Sơ đồ về kiến trúc mô hình mạng đề xuất*

Hệ thống sẽ sử dụng mô hình mạng đề xuất để xử lí 4 nhiệm vụ chính như sau:

- Tìm kiếm anime tương tự
- Tìm kiếm người dùng tương tự
- Đề xuất anime cho người dùng
- Dự đoán đánh giá của người dùng với những anime chưa xem

3.1 Tổng quan về kho dữ liệu

Bộ dữ liệu "MyAnimeList Database 2020" chứa thông tin về 17.562 anime từ 325.772 người dùng khác nhau. Cụ thể, bộ dữ liệu này chứa những thông tin:

- Danh sách anime theo từng người dùng, bao gồm: đã bỏ (dropped), hoàn thành (complete), dự định coi, coi thường xuyên và đang coi/đang tạm dừng.
- Xếp hạng do người dùng đưa ra cho các anime mà họ đã xem hết.
- Thông tin về bộ phim anime như thể loại, số liệu thống kê, hãng phim, v.v,...

- Thông tin về đánh giá, tóm tắt, thông tin về đội ngũ, số lượng thống kê về anime, thể loại, v.v,...

Bộ dữ liệu bao gồm các tập dữ liệu chính sau:

- **animelist.csv:** có danh sách tất cả các anime được người dùng đăng ký với điểm tương ứng, trạng thái xem và số tập đã xem. Tập dữ liệu này chứa 109 triệu hàng, 17.562 anime khác nhau và 325.772 người dùng khác nhau, bao gồm các cột sau:
  - user_id: ID người dùng không xác định được tạo ngẫu nhiên.
  - anime_id: ID MyAnimeList của anime
  - score: điểm từ 1 đến 10 do người dùng cung cấp. 0 nếu người dùng không chỉ định điểm
  - watching_status: ID trạng thái của anime này trong danh sách anime của người dùng này.
  - watched_episodes: số tập phim mà người dùng đã xem
- **watching_status.csv:** mô tả mọi trạng thái có thể có của cột: "watching_status" trong animelist.csv
- **rating_complete.csv:** là một tập dữ liệu con của animelist.csv. Tập dữ liệu này chỉ xem xét các anime mà người dùng đã xem hoàn toàn (watching_status==2) và cho điểm (score!=0). Bộ dữ liệu này chứa 57 triệu xếp hạng được áp dụng cho 16.872 anime của 310.059 người dùng. Tệp này có các cột sau:
  - user_id: ID người dùng được tạo ngẫu nhiên không xác định được.
  - anime_id: - ID MyAnimelist của anime mà người dùng này đã đánh giá.
  - rating: xếp hạng mà người dùng này đã chỉ định.
- **anime.csv:** chứa thông tin chung của mọi anime (17.562 anime khác nhau) như thể loại, số liệu thống kê, hãng phim, v.v. Tập dữ liệu này gồm 35 trường:
  - Các trường thông tin cơ bản: MAL_ID, Name, Score, Genres, Type, Episodes, Studios, Source, Rating.
  - Các trường thông tin về xếp hạng và độ phổ biến: Ranked, Popularity, Members, Favorites.
  - Các trường thông tin về trạng thái xem của ngươi dùng: Watching, Completed, On-Hold, Dropped, Plan to Watch
  - Các trường thông tin đánh giá điểm cho bộ phim anime: Score-10 đến Score-1
- **anime_with_synopsis.csv:** tập dữ liệu này chứa thông tin bản tóm tắt nội dụng phim của các bộ phim anime tương ứng, gồm các trường: MAL_ID, Name, Score, Genres, sysnopsis.

Nhóm nghiên cứu sử dụng 3 tập dữ liệu phục vụ cho việc xây dựng hệ thống: "animelist.csv", "anime.csv" và "anime_with_synopsis.csv"

3.2 Tiền xử lí dữ liệu

Quy trình tiền xử lý dữ liệu cho hệ thống đề xuất anime được thực hiện qua các bước sau:

1. **Nhập dữ liệu:** Dữ liệu được tải từ file `animelist.csv` và chỉ giữ lại các cột quan trọng bao gồm:
   - `user_id`: ID của người dùng.
   - `anime_id`: ID của anime.
   - `rating`: Đánh giá của người dùng đối với anime.

   Những thông tin này là cơ sở để xây dựng hệ thống đề xuất.
2. **Lọc người dùng hoạt động:** Chỉ giữ lại những người dùng đã đánh giá ít nhất 400 bộ anime. Việc lọc này đảm bảo rằng dữ liệu đủ phong phú và các người dùng trong tập dữ liệu có ảnh hưởng đáng kể đến hệ thống.

3. **Chuẩn hóa giá trị đánh giá:** Giá trị `rating` được chuẩn hóa về khoảng $[0, 1]$ bằng phương pháp *Min-Max Scaling*. Công thức chuẩn hóa:

$$x' = \frac{x - \text{min\_rating}}{\text{max\_rating} - \text{min\_rating}}$$

   Trong đó:

   - x' là giá trị chuẩn hóa.
   - x là giá trị ban đầu.
   - min_rating và max_rating lần lượt là giá trị nhỏ nhất và lớn nhất của tập dữ liệu.

   Sau chuẩn hóa, giá trị trung bình là 0.4048, phản ánh dữ liệu phân phối khá đều.

4. **Loại bỏ dữ liệu trùng lặp:** Các hàng dữ liệu trùng lặp được kiểm tra và loại bỏ hoàn toàn để đảm bảo tính toàn vẹn của dữ liệu và tránh việc mô hình học bị ảnh hưởng bởi các bản ghi lặp lại.

5. **Phân tích người dùng và anime phổ biến:** Chúng tôi xác định:

   - Những người dùng có nhiều đánh giá nhất (top users).
   - Những anime phổ biến nhất dựa trên số lượng đánh giá (top animes).

   Kết quả phân tích này giúp hệ thống hiểu được sở thích và xu hướng đánh giá của các nhóm người dùng lớn.

6. **Mã hóa dữ liệu:** Các giá trị phân loại (`user_id`, `anime_id`) được mã hóa thành số nguyên. Sau khi mã hóa:

   - Có tổng cộng 91,641 người dùng và 17,560 anime.
   - Một từ điển ánh xạ được tạo để chuyển đổi giữa các giá trị ban đầu và giá trị được mã hóa.

7. **Xáo trộn dữ liệu:** Dữ liệu được xáo trộn ngẫu nhiên để giảm thiểu nguy cơ thiên lệch. Việc này đảm bảo rằng mô hình không học theo bất kỳ thứ tự nào từ dữ liệu gốc.

8. **Chia tập dữ liệu:** Dữ liệu được chia thành:

   - **Tập huấn luyện (train set):** Gồm 71,408,113 bản ghi, dùng để huấn luyện mô hình.
   - **Tập kiểm tra (test set):** Gồm 10,000 bản ghi, dùng để đánh giá hiệu suất mô hình trên dữ liệu mới.

3.3 Xây dựng mô hình mạng học

Mô hình gợi ý anime được xây dựng với kiến trúc mạng học sâu, bao gồm các thành phần chính như sau:

- **Input Layers:**
  - Lớp đầu vào nhận thông tin của người dùng (`user`) và anime (`anime`) với kích thước `(None, 1)`.
- **Embedding Layers:**
  - Lớp nhúng (`Embedding`) ánh xạ người dùng và anime thành các vector đặc trưng kích thước cố định (`(None, 1, 128)`).
  - Tổng số tham số: 23,460,096, với mỗi lớp nhúng có `11,730,048` tham số.
- **Dot Product Layer:**
  - Tính tích vô hướng giữa vector nhúng của người dùng và anime, tạo đầu ra với kích thước `(None, 1, 1)`.
- **Flatten Layer:**
  - Chuyển tensor từ nhiều chiều (`(None, 1, 1)`) về một chiều (`(None, 1)`).
- **Dense Layer:**

  - Fully connected layer với một đầu ra (`(None, 1)`), sử dụng kernel initializer `he_normal`.
  - Số tham số: 2.

- **Batch Normalization:**
  - Chuẩn hóa đầu ra của lớp Dense, với 4 tham số.
- **Activation Layer:**
  - Sử dụng hàm kích hoạt sigmoid để chuyển đầu ra thành xác suất, biểu thị khả năng người dùng yêu thích anime.

Tổng số tham số của mô hình: 23,460,102, trong đó 23,460,100 tham số có thể huấn luyện và 2 tham số không huấn luyện.

Để tối ưu hóa quá trình huấn luyện, các callback sau được áp dụng:

- **LearningRateScheduler:** Điều chỉnh tốc độ học để mô hình hội tụ nhanh và chính xác hơn, giảm rủi ro "nhảy quá xa" (overshooting) hoặc "đi quá chậm" (underfitting).
- **ModelCheckpoint:** Lưu lại trọng số mô hình tốt nhất nhằm giảm nguy cơ mất trạng thái tốt nhất do overfitting.
- **EarlyStopping:** Dừng sớm khi mô hình không cải thiện, tiết kiệm tài nguyên và tránh overfitting.

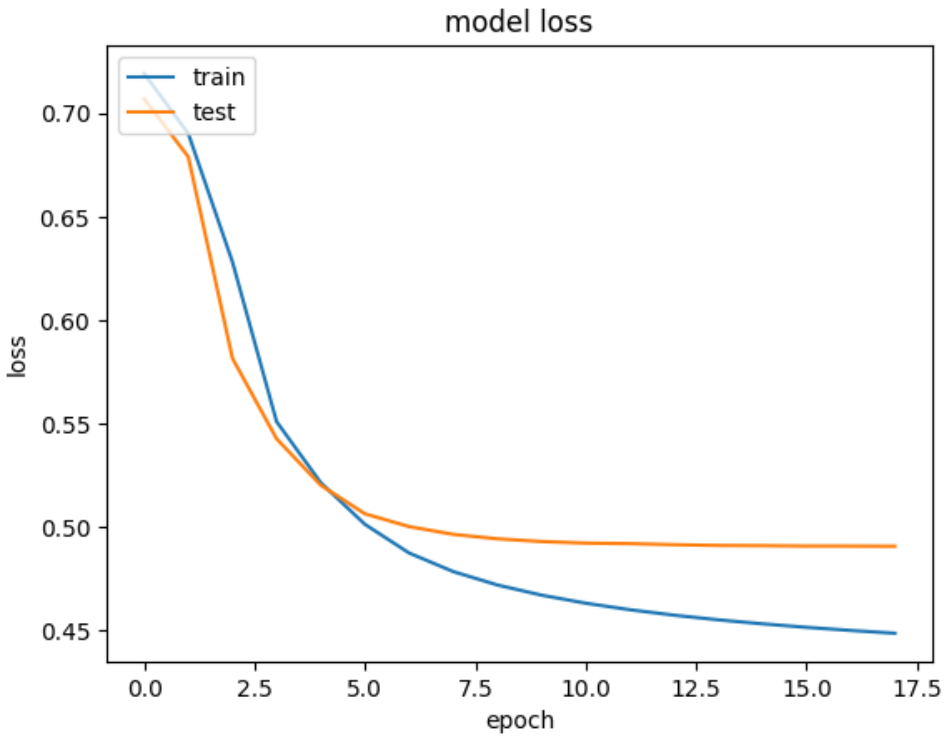
Bộ callback này phối hợp để đảm bảo quá trình huấn luyện diễn ra hiệu quả, ổn định và tiết kiệm.

Mô hình được huấn luyện trên tập dữ liệu đào tạo `X_train_array` và nhãn `y_train` với các thông số sau:

- **Batch size:** `10000`
- **Epochs:** `20`
- **Validation data:** Tập `X_test_array` và `y_test`.
- **Callbacks:** Sử dụng bộ callback bao gồm `LearningRateScheduler`, `ModelCheckpoint`, và `EarlyStopping`.

Quá trình huấn luyện được thực hiện với hàm `model.fit()` và trọng số tốt nhất của mô hình được tải lại sau khi huấn luyện, đảm bảo hiệu suất cao nhất.

Kết quả huấn luyện: Biểu đồ dưới đây minh họa giá trị hàm mất mát (*loss*) trên tập huấn luyện và tập kiểm tra qua các epoch.



Hình 2: *Giá trị hàm mất mát trên tập huấn luyện và tập kiểm tra.*

Biểu đồ cho thấy sự giảm dần của hàm mất mát trên cả tập huấn luyện và tập kiểm tra, cho thấy mô hình học tốt hơn qua từng epoch. Đồng thời, sự hội tụ giữa `loss` và `val_loss` phản ánh khả năng tổng quát hóa tốt của mô hình.

Trích xuất trọng số embedding: Lớp embedding của mô hình lưu trữ các vector đặc trưng đại diện cho người dùng và anime. Để chuẩn hóa các vector embedding, ta sử dụng hàm `extract_weights()`, thực hiện các bước sau:

1. Truy xuất trọng số từ lớp embedding dựa trên tên lớp (`name`).
2. Tính chuẩn Euclidean (L2 norm) cho mỗi vector embedding.
3. Chuẩn hóa vector bằng cách chia từng phần tử của vector cho chuẩn L2 tương ứng.

Hàm `extract_weights()` được sử dụng để lấy vector embedding của anime (`anime_weights`) và người dùng (`user_weights`).

Xử lý thông tin anime: Tệp dữ liệu `anime.csv` chứa thông tin về các anime, bao gồm *ID*, tên tiếng Anh, điểm đánh giá, thể loại, số tập, và các thông tin khác. Dữ liệu được xử lý như sau:

1. Đổi tên các cột cho phù hợp, ví dụ: `MAL_ID` thành `anime_id`, `English name` thành `eng_version`.
2. Tự động sửa lỗi các tên anime thiếu hoặc không rõ bằng cách thay thế bằng tên gốc nếu tên tiếng Anh không tồn tại.
3. Sắp xếp anime theo điểm đánh giá từ cao đến thấp.
4. Lọc các cột cần thiết, bao gồm: `anime_id`, `eng_version`, `Score`, `Genres`, `Episodes`, `Type`, `Premiered`, và `Members`.

Hàm `getAnimeFrame()` cho phép truy vấn thông tin chi tiết của một anime dựa trên `anime_id` hoặc tên tiếng Anh (`eng_version`).

Truy xuất synopsis: Tệp dữ liệu `anime_with_synopsis.csv` được sử dụng để cung cấp mô tả nội dung (*synopsis*) của anime. Các bước thực hiện:

1. Lọc các cột liên quan: `MAL_ID`, `Name`, `Genres`, và `sypnopsis`.

2. Sử dụng hàm `getSypnopsis()` để lấy nội dung mô tả của anime dựa trên `MAL_ID` hoặc tên gốc (`Name`).

3.4 Tìm kiếm anime tương tự

Hàm `find_similar_animes()` được sử dụng để tìm kiếm các anime tương tự dựa trên độ tương đồng cosine giữa các vector embedding trong không gian đặc trưng. Cách hoạt động và kết quả minh họa như sau:

Cách hoạt động

1. **Lấy thông tin đầu vào:**
   - `name`: Tên hoặc mã số (ID) của anime cần tìm kiếm tương tự.
   - `n`: Số lượng anime tương tự cần tìm kiếm (mặc định là 10).
   - `return_dist`: Nếu đặt `True`, trả về cả khoảng cách cosine và các anime tương tự.
   - `neg`: Nếu đặt `True`, tìm kiếm các anime không liên quan nhất.
2. **Xử lý dữ liệu:**
   - Truy xuất ID của anime từ dataframe thông qua hàm `getAnimeFrame()`.
   - Sử dụng embedding để mã hóa ID thành không gian đặc trưng (`encoded_index`).
   - Tính toán độ tương đồng cosine giữa embedding của anime đầu vào và các anime khác thông qua tích vô hướng.
3. **Sắp xếp và chọn lọc:**
   - Các điểm tương đồng được sắp xếp để tìm ra danh sách các anime gần nhất (`neg=False`) hoặc xa nhất (`neg=True`).
4. **Trả về kết quả:**
   - Tạo dataframe chứa thông tin về anime tương tự: tên, độ tương đồng, thể loại, và phần mô tả nội dung (`sypnopsis`).

Ví dụ minh họa

Tìm kiếm các anime gần giống với *Dragon Ball Z* (`n=5`):

	name	similarity	genre	sypnopsis
4	Dragon Ball	0.980308	Adventure, Comedy, Fantasy, Martial Arts, Shounen, Super Power	Gokuu Son is a young boy who lives in the woods all alone—that is, until a girl named Bulma runs into him in her search for a set of magical objects called the "Dragon Balls." Since the artifacts are said to grant one wish to whoever collects all seven, Bulma hopes to gather them and wish for a perfect boyfriend. Gokuu happens to be in possession of a dragon ball, but unfortunately for Bulma, he refuses to part ways with it, so she makes him a deal: he can tag along on her journey if he lets her borrow the dragon ball's power. With that, the two set off on the journey of a lifetime. They don't go on the journey alone. On the way, they meet the old Muten-Roshi and wannabe disciple Kuririn, with whom Gokuu trains to become a stronger martial artist for the upcoming World Martial Arts Tournament. However, it's not all fun and games; the ability to make any wish come true is a powerful one, and there are others who would do much worse than just wishing for a boyfriend. To stop those who would try to abuse the legendary power, they train to become stronger fighters, using their newfound strength to help the people around them along the way.
3	Dragon Ball GT	0.925363	Action, Sci-Fi, Adventure, Comedy, Super Power, Magic, Fantasy, Shounen	Emperor Pilaf finally has his hands on the Black Star Dragon Balls after years of searching, which are said to be twice as powerful as Earth's normal ones. Pilaf is about to make his wish for world domination when he is interrupted by Gokuu Son. As a result, Pilaf flubs his wish and accidentally turns Gokuu back into a child. After the wish is granted, the Black Star Dragon Balls scatter across the galaxy. However, Gokuu discovers that they will cause the Earth to explode unless they are all brought back within a year. Uniting with his granddaughter Pan and a young adult Trunks, Gokuu sets off on an adventure through the universe to find the Black Star Dragon Balls and save his planet from destruction!
2	Naruto	0.820448	Action, Adventure, Comedy, Super Power, Martial Arts, Shounen	oments prior to Naruto Uzumaki's birth, a huge demon known as the Kyuubi, the Nine-Tailed Fox, attacked Konohagakure, the Hidden Leaf Village, and wreaked havoc. In order to put an end to the Kyuubi's rampage, the leader of the village, the Fourth Hokage, sacrificed his life and sealed the monstrous beast inside the newborn Naruto. Now, Naruto is a hyperactive and knuckle-headed ninja still living in Konohagakure. Shunned because of the Kyuubi inside him, Naruto struggles to find his place in the village, while his burning desire to become the Hokage of Konohagakure leads him not only to some great new friends, but also some deadly foes.
1	Dragon Ball Z Kai	0.818219	Action, Adventure, Comedy, Fantasy, Martial Arts, Shounen, Super Power	Five years after the events of Dragon Ball, martial arts expert Gokuu is now a grown man married to his wife Chi-Chi, with a four-year old son named Gohan. While attending a reunion on Turtle Island with his old friends Master Roshi, Krillin, Bulma and others, the festivities are interrupted when a humanoid alien named Raditz not only reveals the truth behind Gokuu's past, but kidnaps Gohan as well. With Raditz displaying power beyond anything Gokuu has seen before, he is forced to team up with his old nemesis Piccolo, in order to rescue his son. But when Gokuu and Piccolo reveal the secret of the seven mystical wish-granting Dragon Balls to Raditz, he informs the duo that there is more of his race, the Saiyans, and they won't pass up an opportunity to seize the power of the Dragon Balls for themselves. These events begin the saga of Dragon Ball Kai, a story that finds Gokuu and his friends and family constantly defending the galaxy from increasingly more powerful threats. Bizarre, comical, heartwarming and threatening characters come together in a series of battles that push the powers and abilities of Gokuu and his friends beyond anything they have ever experienced.
0	Yu-Gi-Oh!	0.807040	Adventure, Game, Shounen	gend says that the enigmatic Millennium Puzzle will grant one wish to whoever deciphers its ancient secrets. Upon solving it, high school student Yuugi Mutou unleashes "another Yuugi," a peculiar presence contained inside. Now, whenever he is faced with a dilemma, this mysterious alter ego makes an appearance and aids him in his troubles. Wishing to unravel the mystery behind this strange spirit, Yuugi and his companions find themselves competing with several opponents in "Duel Monsters," a challenging card game used by people seeking to steal the Millennium Puzzle in a desperate attempt to harness the great power within. As the questions pile on, it is not long before they figure out that there is more than pride on the line in these duels.

Hình 3: *Tìm kiếm các anime gần giống với Dragon Ball Z*

Nhận xét

- Các anime có độ tương đồng cao như *Dragon Ball*, *Dragon Ball GT*, hoặc *Dragon Ball Z Kai* thường thuộc cùng thể loại (Shounen, Super Power, Adventure).
- Anime như *Naruto* hay *Yu-Gi-Oh!* tuy không thuộc cùng series nhưng có nội dung và thể loại tương đồng.
- Phần mô tả (*sypnopsis*) giúp người dùng hiểu rõ hơn về nội dung của từng anime được đề xuất.

Hàm `find_similar_animes()` cung cấp cách tiếp cận hiệu quả để gợi ý anime dựa trên sở thích cá nhân, phù hợp cho hệ thống đề xuất nội dung.

3.5 Tìm kiếm người dùng tương tự

3.5.1 Tìm người dùng có sở thích tương đồng

Hàm `find_similar_users()` được sử dụng để tìm kiếm các người dùng có sở thích tương đồng dựa trên độ tương đồng cosine giữa các vector embedding của người dùng. Dưới đây là chi tiết cách hoạt động và ví dụ minh họa.

Cách hoạt động

1. **Lấy thông tin đầu vào:**
   - `item_input`: ID của người dùng cần tìm kiếm người dùng tương tự.
   - `n`: Số lượng người dùng tương tự cần tìm kiếm (mặc định là 10).
   - `return_dist`: Nếu đặt `True`, trả về cả khoảng cách cosine và các người dùng tương tự.
   - `neg`: Nếu đặt `True`, tìm kiếm các người dùng không liên quan nhất.
2. **Xử lý dữ liệu:**
   - Truy xuất ID của người dùng từ dataframe `rating_df`.
   - Sử dụng embedding để mã hóa ID thành không gian đặc trưng (`encoded_index`).
   - Tính toán độ tương đồng cosine giữa embedding của người dùng đầu vào và các người dùng khác thông qua tích vô hướng.
3. **Sắp xếp và chọn lọc:**
   - Các điểm tương đồng được sắp xếp để tìm ra danh sách người dùng gần nhất (`neg=False`) hoặc xa nhất (`neg=True`).
   - Loại bỏ các người dùng có độ tương đồng thấp hơn ngưỡng 0.4 và loại bỏ người dùng gốc khỏi danh sách kết quả.
4. **Trả về kết quả:**
   - Tạo dataframe chứa thông tin về người dùng tương tự: ID người dùng và độ tương đồng.

Ví dụ minh họa

Tìm kiếm người dùng tương tự với người dùng ngẫu nhiên `user_id = 213102` (`n=5`):

STT	ID Người dùng tương tự	Độ tương đồng
1	113673	0.5056
2	262053	0.4749
3	44207	0.4565
4	198391	0.4475
5	157278	0.4427

Bảng 1: *Danh sách người dùng tương tự với người dùng* `user_id = 213102`

Nhận xét

- Các người dùng có độ tương đồng cao (`similarity > 0.4`) thường có sở thích đánh giá anime tương tự với người dùng gốc.
- Hàm `find_similar_users()` có thể hỗ trợ trong việc xây dựng hệ thống gợi ý, giúp đề xuất nội dung dựa trên những người dùng có thị hiếu tương đồng.

3.5.2 Phân tích sở thích của người dùng

Hàm `get_user_preferences()` được sử dụng để phân tích các sở thích anime của người dùng dựa trên các đánh giá cao nhất mà người dùng đã thực hiện. Ngoài ra, hàm `getFavGenre()` tạo một biểu đồ từ khóa (*word cloud*) để minh họa các thể loại yêu thích của người dùng.

Cách hoạt động

1. **Xác định các anime được đánh giá cao nhất:**
   - Lấy các anime mà người dùng đã đánh giá từ dataframe `rating_df`.
   - Chỉ giữ lại những anime được đánh giá cao hơn 75% mức đánh giá trung bình của người dùng.
   - Sắp xếp các anime này theo thứ tự giảm dần dựa trên điểm số đánh giá.
2. **Lọc và phân tích:**
   - Lọc thông tin từ dataframe `df` để lấy danh sách các thể loại (*Genres*) và tên anime (*eng_version*) được người dùng ưa thích.
   - Tạo danh sách các thể loại yêu thích bằng cách đếm tần suất xuất hiện của các thể loại.
3. **Hiển thị thông tin:**
   - Tạo biểu đồ từ khóa (*word cloud*) minh họa tần suất xuất hiện của các thể loại yêu thích.
   - Hiển thị danh sách các anime mà người dùng đánh giá cao nhất.

Ví dụ minh họa

Dưới đây là danh sách các anime được người dùng ngẫu nhiên `user_id = 213102` đánh giá cao nhất cùng với các thể loại tương ứng:

Tên Anime	Thể loại
Fullmetal Alchemist: Brotherhood	Action, Military, Adventure, Comedy, Drama, Magic, Fantasy, Shounen
Steins;Gate	Thriller, Sci-Fi
A Silent Voice	Drama, School, Shounen
Your Name.	Romance, Supernatural, School, Drama
Kaguya-sama: Love is War Season 2	Comedy, Psychological, Romance, School, Seinen

Bảng 2: *Danh sách anime được đánh giá cao bởi người dùng* `user_id = 213102`

Hình 4: *Biểu đồ từ khóa thể hiện thể loại anime yêu thích của người dùng `user_id = 213102`*

Nhận xét

- Người dùng có xu hướng ưa thích các thể loại hành động, phiêu lưu và drama.
- Biểu đồ từ khóa cung cấp cái nhìn trực quan về tần suất các thể loại xuất hiện trong danh sách anime yêu thích.
- Thông tin này có thể được sử dụng để gợi ý các anime phù hợp với sở thích cá nhân của người dùng.

3.6 Đề xuất anime cho người dùng

Hàm `get_recommended_animes` được thiết kế để đưa ra danh sách anime đề xuất cho người dùng dựa trên sở thích của các người dùng tương tự. Quy trình hoạt động được chia thành các bước chính sau:

Thu thập danh sách anime từ người dùng tương tự

- **Thu thập sở thích của các người dùng tương tự**:
  - Với mỗi `user_id` trong `similar_users`, lấy danh sách anime mà họ đánh giá cao bằng hàm `get_user_preferences`.
- **Loại bỏ anime đã xem**:
  - Các anime mà người dùng hiện tại (`user_pref`) đã xem sẽ bị loại bỏ khỏi danh sách đề xuất.

Tính tần suất xuất hiện của anime

- Tất cả anime từ danh sách sở thích của người dùng tương tự được tổng hợp thành một danh sách lớn (`anime_list`).
- Sử dụng pandas để đếm số lần xuất hiện của từng anime trong danh sách, từ đó sắp xếp theo tần suất giảm dần.

Xây dựng thông tin chi tiết cho anime được đề xuất

- Với mỗi anime trong danh sách đã sắp xếp, hàm cố gắng thu thập các thông tin sau:
  - Tên anime (`anime_name`).
  - Số người dùng tương tự yêu thích anime đó (`n_user_pref`).
  - Thể loại (`Genres`) của anime.
  - Nội dung tóm tắt (`sypnopsis`) của anime.

- Thông tin được lưu trữ vào danh sách `recommended_animes`, sau đó chuyển đổi thành một DataFrame để hiển thị.

Hiển thị kết quả

Danh sách anime được đề xuất cho người dùng cụ thể sẽ được hiển thị, kèm theo số lần xuất hiện trong danh sách sở thích của các người dùng tương tự. Đồng thời, một biểu đồ word cloud cũng được tạo để minh họa các thể loại phổ biến trong danh sách đề xuất.

Ví dụ: Đề xuất cho người dùng `random_user`:

Top recommendations for user: 213102

n	anime_name	Genres	sypnopsis
5	Prison School	Comedy, Ecchi, Romance, School, Seinen	ocated on the outskirts of Tokyo, Hachimitsu Private Academy is a prestigious all-girls boarding school, famous for its high-quality education and disciplined students. However, this is all about to change due to the revision of the school's most iconic policy, as boys are now able to enroll as well. At the start of the first semester under this new decree, a mere five boys have been accepted, effectively splitting the student body into a ratio of two hundred girls to one boy. Kiyoshi, Gakuto, Shingo, Andre, and Jo are quickly cast away without having a chance to make any kind of a first impression. Unable to communicate with their fellow female students, the eager boys set their sights on a far more dangerous task: peeping into the girls' bath! It's only after their plan is thoroughly decimated by the infamous Underground Student Council that the motley crew find their freedom abruptly taken from them, as they are thrown into the school's prison with the sentence of an entire month as punishment. Thus begins the tale of the boys' harsh lives in Prison School, a righteous struggle that will ultimately test the bonds of friendship and perverted brotherhood.
4	Is It Wrong to Try to Pick Up Girls in a Dungeon?	Action, Adventure, Comedy, Romance, Fantasy	fe in the bustling city of Orario is never dull, especially for Bell Cranel, a naïve young man who hopes to become the greatest adventurer in the land. After a chance encounter with the lonely goddess, Hestia, his dreams become a little closer to reality. With her support, Bell embarks on a fantastic quest as he ventures deep within the city's monster-filled catacombs, known only as the "Dungeon." Death lurks around every corner in the cavernous depths of this terrifying labyrinth, and a mysterious power moves amidst the shadows. Even on the surface, survival is a hard-earned privilege. Indeed, nothing is ever certain in a world where gods and humans live and work together, especially when they often struggle to get along. One thing is for sure, though: a myriad of blunders, triumphs and friendships awaits the dauntlessly optimistic protagonist of this herculean tale.
4	Naruto	Action, Adventure, Comedy, Super Power, Martial Arts, Shounen	oments prior to Naruto Uzumaki's birth, a huge demon known as the Kyuubi, the Nine-Tailed Fox, attacked Konohagakure, the Hidden Leaf Village, and wreaked havoc. In order to put an end to the Kyuubi's rampage, the leader of the village, the Fourth Hokage, sacrificed his life and sealed the monstrous beast inside the newborn Naruto. Now, Naruto is a hyperactive and knuckle-headed ninja still living in Konohagakure. Shunned because of the Kyuubi inside him, Naruto struggles to find his place in the village, while his burning desire to become the Hokage of Konohagakure leads him not only to some great new friends, but also some deadly foes.
4	That Time I Got Reincarnated as a Slime	Fantasy, Shounen	Thirty-seven-year-old Satoru Mikami is a typical corporate worker, who is perfectly content with his monotonous lifestyle in Tokyo, other than failing to nail down a girlfriend even once throughout his life. In the midst of a casual encounter with his colleague, he falls victim to a random assailant on the streets and is stabbed. However, while succumbing to his injuries, a peculiar voice echoes in his mind, and recites a bunch of commands which the dying man cannot make sense of. When Satoru regains consciousness, he discovers that he has reincarnated as a goop of slime in an unfamiliar realm. In doing so, he acquires newfound skills—notably, the power to devour anything and mimic its appearance and abilities. He then stumbles upon the sealed Catastrophe-level monster "Storm Dragon" Veldora who had been sealed away for the past 300 years for devastating a town to ashes. Sympathetic to his predicament, Satoru befriends him, promising to assist in destroying the seal. In return, Veldora bestows upon him the name Rimuru Tempest to grant him divine protection. Now, liberated from the mundanities of his past life, Rimuru embarks on a fresh journey with a distinct goal in mind. As he grows accustomed to his new physique, his gooey antics ripple throughout the world, gradually altering his fate.

Hình 5: *Đề xuất cho người dùng `user_id = 213102`*

3.7 Dự đoán đánh giá của người dùng với những anime chưa xem

Cách hoạt động

1. **Lấy danh sách anime chưa được xem bởi người dùng:**
   - Sử dụng dataframe `rating_df` để xác định các anime đã được người dùng `user_id` đánh giá.
   - Lọc danh sách các anime trong dataframe `df` để chỉ lấy những anime chưa được người dùng đánh giá.
2. **Xử lý dữ liệu:**
   - Mã hóa `anime_id` từ danh sách anime chưa xem bằng từ điển `anime2anime_encoded`.
   - Tạo mảng `user_anime_array`, chứa cặp `(user_id, anime_id)` để dự đoán đánh giá.
3. **Dự đoán:**
   - Sử dụng mô hình để dự đoán điểm đánh giá cho tất cả các anime chưa xem.
   - Lọc ra 10 anime có điểm đánh giá dự đoán cao nhất.
4. **Tạo danh sách kết quả:**
   - Với mỗi anime được gợi ý, lấy thông tin như tên (*name*), thể loại (*genre*), và mô tả ngắn gọn (*sypnopsis*).

Kết quả ví dụ

Dưới đây là 10 anime được gợi ý cho người dùng ngẫu nhiên `user_id = 213102`:

Tên Anime	Đánh Giá Dự Đoán	Thể Loại
Afro Samurai the Movie	0.972	Action, Adventure, Samurai
Dogtanian and the Three Muskehounds	0.966	Adventure, Historical
Yamada-kun to 7-nin no Majo	0.949	Comedy, Romance, School, Shounen
Dramaturgy	0.943	Music
The Secret About That Girl	0.943	Music, Romance
Doraemon	0.942	Adventure, Comedy, Fantasy, Kids, Shounen
Ramayana: The Legend of Prince Rama	0.941	Adventure
Outlanders	0.940	Sci-Fi, Adventure, Comedy, Ecchi, Shounen
Dokidoki Dream!!!	0.929	Music
Clock Lock Works	0.928	Music

Bảng 3: *Danh sách top 10 anime được gợi ý cho người dùng* `user_id = 213102`

Nhận xét

- Danh sách anime gợi ý có sự đa dạng về thể loại, phù hợp với sở thích tiềm năng của người dùng.
- Các điểm đánh giá dự đoán khá cao, cho thấy khả năng người dùng sẽ thích các anime này.
- Hệ thống có thể cải thiện bằng cách bổ sung các thông tin như độ phổ biến hoặc đánh giá từ người dùng khác để tăng độ tin cậy của gợi ý.

4 Đánh giá kết quả và kết luận

4.1 Đánh giá kết quả

Hệ thống gợi ý anime dựa trên mô hình học sâu đã được triển khai và thử nghiệm trên các nhiệm vụ khác nhau như tìm kiếm anime tương tự, tìm kiếm người dùng tương tự, đề xuất anime cho người dùng, và dự đoán đánh giá của người dùng đối với các anime chưa xem. Dưới đây là một số nhận xét quan trọng dựa trên kết quả thu được:

1. **Hiệu quả của mô hình:**
   - Mô hình thể hiện khả năng gợi ý các anime tương tự với độ chính xác cao, thể hiện qua việc tìm thấy các anime có nội dung và thể loại liên quan chặt chẽ.
   - Các đề xuất anime cho người dùng dựa trên lịch sử và sở thích của các người dùng tương tự mang tính cá nhân hóa cao.
2. **Sự phù hợp của kết quả:**
   - Các anime được gợi ý trong phần *Dự đoán đánh giá của người dùng với những anime chưa xem* có điểm đánh giá dự đoán cao, phù hợp với sở thích của người dùng ngẫu nhiên.
   - Các kết quả trong tìm kiếm người dùng tương tự và phân tích sở thích cho thấy mối quan hệ rõ ràng giữa những người dùng có hành vi đánh giá tương tự.
3. **Đánh giá trực quan:**
   - Các biểu đồ từ WordCloud cung cấp một cái nhìn trực quan về sở thích thể loại anime phổ biến của từng nhóm người dùng, hỗ trợ việc hiểu sâu hơn về các yếu tố ảnh hưởng đến gợi ý.
4. **Hạn chế:**
   - Hệ thống hiện tại chưa tính đến các yếu tố thời gian (ví dụ: anime mới ra mắt).
   - Dữ liệu đầu vào có thể chưa đầy đủ hoặc chứa thông tin lỗi, ảnh hưởng đến chất lượng gợi ý.

4.2 Kết luận

Hệ thống gợi ý anime sử dụng mô hình học sâu đã thể hiện tiềm năng lớn trong việc cá nhân hóa trải nghiệm người dùng. Những kết quả đạt được không chỉ cung cấp các gợi ý anime phù hợp mà còn thể hiện khả năng dự đoán điểm đánh giá với độ chính xác cao. Một số điểm nổi bật:

- Hệ thống đã xây dựng được cơ sở dữ liệu đáng tin cậy từ các nguồn dữ liệu có sẵn, xử lý hiệu quả việc mã hóa và ánh xạ dữ liệu.
- Kết quả từ các thử nghiệm cho thấy mô hình có thể áp dụng trong thực tế, đặc biệt trong các nền tảng streaming anime hoặc các hệ thống gợi ý sản phẩm tương tự.
- Với các cải tiến trong tương lai như bổ sung dữ liệu, tích hợp thông tin ngữ cảnh, và nâng cao thuật toán, hệ thống có thể đạt được hiệu quả cao hơn, đáp ứng nhu cầu ngày càng cao của người dùng.

Nhìn chung, hệ thống gợi ý này đã đạt được các mục tiêu đề ra, đồng thời mở ra hướng phát triển mới cho các nghiên cứu trong lĩnh vực gợi ý cá nhân hóa.

5 Tài liệu tham khảo.

Tài liệu

[1] Ph.D. DJacob Murel & Eda Kavlakoglu – *What is collaborative filtering?* March 21, 2024.

[2] GeeksforGeeks - *Collaborative Filtering in Machine Learning*. June 26, 2024.

[3] Abhinav Ajitsaria - *Build a Recommendation Engine With Collaborative Filtering*. 2024.

[4] BadrulSarwar, GeorgeKarypis, JosephKonstan, & JohnRiedl - *Item-Based Collaborative Filtering Recommendation Algorithms*.

[5] Maciej Kula - *Metadata Embeddings for User and Item Cold-start Recommendations*.