BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG

PHÂN HIỆU TRƯỜNG ĐẠI HỌC THỦY LỢI

BỘ MÔN CÔNG NGHỆ THÔNG TIN

MÔN HỌC: KHAI PHÁ DỮ LIỆU

TÊN ĐỀ TÀI: HỆ THỐNG GỢI Ý SÁCH

Giảng viên:	ThS. Vũ Thị Hạnh
Sinh viên thực hiện:	Phạm Thành Doanh – 2351267258
Lớp:	S26-65TTNT

TP. Hồ Chí Minh, ngày 14 tháng 1 năm 2025

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TP. Hồ Chí Minh, ngày … tháng … năm 2025

Chữ ký của giảng viên

Lời cảm ơn

Trước tiên, với tình cảm sâu sắc và chân thành nhất, nhóm em xin được bày tỏ lòng biết ơn đến tất cả các cá nhân và tổ chức đã tạo điều kiện, hỗ trợ và giúp đỡ nhóm em trong suốt quá trình học tập và thực hiện đề tài này. Trong suốt thời gian học tập tại trường, nhóm em đã nhận được rất nhiều sự quan tâm, chỉ bảo tận tình của quý thầy cô và sự giúp đỡ của bạn bè.

Với lòng biết ơn sâu sắc nhất, nhóm em xin gửi lời cảm ơn chân thành đến quý thầy cô Bộ môn Công nghệ Thông tin – Phân hiệu Trường Đại học Thủy Lợi, những người đã truyền đạt cho nhóm em những kiến thức quý báu trong suốt thời gian học tập. Nhờ có sự giảng dạy, hướng dẫn và động viên của quý thầy cô mà nhóm em có thể hoàn thành tốt đề tài này.

Đặc biệt, nhóm em xin gửi lời cảm ơn sâu sắc đến Cô Vũ Thị Hạnh, người đã trực tiếp hướng dẫn, tận tình giúp đỡ và định hướng cho nhóm trong suốt quá trình thực hiện bài báo cáo.

Do thời gian có hạn và kiến thức còn nhiều hạn chế, bài báo cáo chắc chắn không tránh khỏi những thiếu sót. Nhóm em rất mong nhận được những góp ý quý báu của quý thầy cô để có thể hoàn thiện hơn kiến thức và kinh nghiệm của mình trong lĩnh vực này.

Nhóm em xin chân thành cảm ơn!

MỤC LỤC

Lời cảm ơn 2
CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI 7
1.1. Lý do chọn đề tài 7
1.2. Mục tiêu nghiên cứu 7
1.3. Đối tượng và phạm vi nghiên cứu 7
1.4. Ý nghĩa khoa học và thực tiễn 8
CHƯƠNG 2: MỤC TIÊU VÀ BÀI TOÁN ĐẶT RA 9
2.1. Mô tả bài toán 9
2.2. Input – Output 9
2.3. Yêu cầu đặt ra 9
CHƯƠNG 3: MÔ TẢ DỮ LIỆU VÀ TIỀN XỬ LÝ 10
3.1. Mô tả bộ dữ liệu 10
3.1.1. Books.csv 10
3.1.2. Ratings.csv 10
3.2. Các vấn đề dữ liệu thường gặp và cách xử lý 10
3.3. Quy trình tiền xử lý chi tiết 11
3.3.1. Đọc dữ liệu và chuẩn hóa định dạng 11
3.3.2. Làm sạch dữ liệu (Data Cleaning) 11
3.3.3. Lọc explicit feedback 11
3.3.4. Giảm độ thưa (Sparsity Reduction) bằng lọc user/item 12
3.3.5. Mã hóa user và item (Encoding) 12
3.3.6. Chia tập train/test theo user (User-based split) 12
3.4. Kết quả sau tiền xử lý 13
CHƯƠNG 4: PHƯƠNG PHÁP KHAI PHÁ DỮ LIỆU / MÔ HÌNH ML 14
4.1. Tổng quan hệ thống gợi ý 14

4.2. Baseline: Popularity-based Recommendation 14
4.3. Collaborative Filtering và Matrix Factorization 14
4.3.1. Bài toán ma trận user–item 14
4.3.2. Ý tưởng Matrix Factorization 15
4.3.3. Hàm mất mát và regularization 15
4.3.4. Vì sao MF phù hợp bài toán này 15
4.4. Chiến lược huấn luyện và lựa chọn siêu tham số (giải thích, không code) ...15
4.5. Sinh gợi ý Top-N (Recommendation Generation) 16
CHƯƠNG 5: KẾT QUẢ VÀ ĐÁNH GIÁ MÔ HÌNH 17
5.1. Mục tiêu đánh giá 17
5.2. Kết quả EDA và các nhận xét quan trọng 17
5.3. Baseline Popularity: kết quả và phân tích 18
5.4. Đánh giá dự đoán rating: RMSE và MAE 19
5.5. Đánh giá gợi ý Top-K: Precision@K và Recall@K 20
5.6. Demo gợi ý cho người dùng và phân tích định tính 21
5.7. Tổng hợp kết quả và hạn chế 21
CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 24
6.1. Kết luận 24
6.2. Hướng phát triển 24
CHƯƠNG 7: TÀI LIỆU THAM KHẢO 25

DANH MỤC HÌNH ẢNH

Hình 1 Kết quả EDA sau tiền xử lý (phân phối rating, số rating theo user/sách, phân phối điểm trung bình theo sách). 18

Hình 2 Lịch sử huấn luyện mô hình Matrix Factorization (Loss/RMSE/MAE theo epoch). 20

DANH MỤC BẢNG

Bảng 1 Top 10 sách theo baseline popularity. 19

Bảng 2 Kết quả RMSE và MAE trên tập test. 20

Bảng 3 Top 10 sách đề xuất cho User 21

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1.1. Lý do chọn đề tài

Trong thời đại bùng nổ thông tin số, lượng đầu sách trên các nền tảng trực tuyến tăng nhanh khiến người đọc gặp khó khăn khi lựa chọn. Việc "tìm đúng sách" trở thành bài toán vừa tốn thời gian vừa dễ bị nhiễu bởi quảng cáo, xu hướng, hoặc sở thích nhất thời. Trong khi đó, dữ liệu đánh giá (rating) của người dùng lại chứa tín hiệu rất mạnh phản ánh mức độ yêu thích thực tế.

Hệ thống gợi ý (Recommender System) là giải pháp quan trọng để cá nhân hóa trải nghiệm: dựa vào lịch sử đánh giá của người dùng, hệ thống có thể dự đoán mức độ phù hợp của mỗi cuốn sách và đưa ra danh sách đề xuất. Đây là hướng ứng dụng phổ biến trong thương mại điện tử và truyền thông số (Amazon, Netflix, Goodreads…), đồng thời phù hợp với yêu cầu môn khai phá dữ liệu vì có đầy đủ bài toán xử lý dữ liệu, mô hình hóa, và đánh giá.

1.2. Mục tiêu nghiên cứu

Đề tài hướng đến các mục tiêu cụ thể:

- Xây dựng pipeline khai phá dữ liệu từ 2 tệp Books.csv và Ratings.csv: đọc dữ liệu, làm sạch, chuẩn hóa và chuẩn bị dữ liệu cho mô hình.
- Phân tích dữ liệu (EDA) để hiểu phân phối rating, độ thưa (sparsity), mức độ phổ biến của sách và hành vi đánh giá của người dùng.
- Xây dựng hệ thống gợi ý theo hướng Collaborative Filtering, có baseline và mô hình học máy (Machine Learning).
- Đánh giá mô hình bằng các thước đo phù hợp: RMSE/MAE cho dự đoán rating, và Precision@K/Recall@K cho gợi ý Top-K.
- Xây dựng demo: nhập User-ID và trả về Top-N sách đề xuất (loại bỏ sách đã đánh giá).

1.3. Đối tượng và phạm vi nghiên cứu

- Đối tượng nghiên cứu: dữ liệu đánh giá của người dùng (User-ID, ISBN, Book-Rating) và mô hình gợi ý dựa trên Collaborative Filtering.

- Phạm vi: sử dụng dữ liệu rating dạng explicit (điểm số), không đi sâu vào xử lý ngôn ngữ tự nhiên nội dung sách. Mô hình chính tập trung vào tương tác user–item; metadata sách chỉ dùng để hiển thị thông tin sách (tên, tác giả, năm…).

1.4. Ý nghĩa khoa học và thực tiễn

- Ý nghĩa khoa học: minh họa quy trình khai phá dữ liệu đầy đủ cho bài toán gợi ý: từ dữ liệu thô đến mô hình hóa và đánh giá theo nhiều góc nhìn.
- Ý nghĩa thực tiễn: có thể áp dụng làm nền cho hệ thống gợi ý sách đơn giản trong thư viện, nhà sách, hoặc các nền tảng nội dung; dễ mở rộng sang hybrid recommendation.

CHƯƠNG 2: MỤC TIÊU VÀ BÀI TOÁN ĐẶT RA

2.1. Mô tả bài toán

Bài toán gợi ý sách có thể nhìn theo 2 hướng:

1. Dự đoán rating (Rating Prediction): dự đoán điểm số mà người dùng u sẽ đánh giá cho sách i.
2. Gợi ý Top-N (Top-N Recommendation): chọn ra N cuốn sách mà user chưa đánh giá nhưng có khả năng thích cao nhất.

Trong đề tài này, hệ thống được xây theo hướng kết hợp: mô hình học dự đoán rating, sau đó chuyển rating dự đoán thành danh sách Top-N.

2.2. Input – Output

Input:

- Books.csv: ISBN, Book-Title, Book-Author, Year-Of-Publication, Publisher, Image URLs…
- Ratings.csv: User-ID, ISBN, Book-Rating.

Output:

- Với mỗi User-ID: danh sách Top-N sách đề xuất, gồm ISBN, tiêu đề, tác giả… và điểm dự đoán (Predicted Rating).
- Các file kết quả phục vụ báo cáo: bảng baseline, bảng metrics, danh sách recommendation mẫu.

2.3. Yêu cầu đặt ra

- Dữ liệu rating rất thưa: đa số user chỉ đánh giá một số ít sách, số lượng sách lớn dẫn đến ma trận user–item sparse.
- Cần cơ chế lọc dữ liệu để mô hình ổn định, tránh nhiễu do user/item quá ít tương tác.
- Đánh giá phải phản ánh cả:
    - độ chính xác dự đoán rating
    - chất lượng danh sách gợi ý Top-K

CHƯƠNG 3: MÔ TẢ DỮ LIỆU VÀ TIỀN XỬ LÝ

3.1. Mô tả bộ dữ liệu

Bộ dữ liệu gồm 2 bảng chính:

3.1.1. Books.csv

Chứa thông tin mô tả sách:

- ISBN: mã định danh sách
- Book-Title: tên sách
- Book-Author: tác giả
- Year-Of-Publication: năm xuất bản
- Publisher: nhà xuất bản
- Image-URL-*: đường dẫn ảnh

Vai trò trong bài toán:

- dùng để join với danh sách gợi ý nhằm hiển thị thông tin sách cho người dùng.
- có thể dùng mở rộng sang content-based trong hướng phát triển.

3.1.2. Ratings.csv

Chứa dữ liệu tương tác:

- User-ID: mã người dùng
- ISBN: mã sách
- Book-Rating: điểm rating (thang 0–10)

Đặc trưng quan trọng:

- Dữ liệu có thể có rating = 0 (nhiều bộ dữ liệu coi 0 là implicit/không đánh giá). Trong hệ thống gợi ý dựa trên explicit feedback, rating = 0 thường được xem như không thể hiện sở thích, nên cần cân nhắc loại bỏ.

3.2. Các vấn đề dữ liệu thường gặp và cách xử lý

Trong thực tế khi đọc dữ liệu CSV, có thể xuất hiện:

- Sai delimiter: file dùng dấu ; thay vì ,.
- Encoding: ký tự đặc biệt gây lỗi đọc file (thường dùng latin-1).
- Dòng lỗi/thiếu: thiếu User-ID/ISBN/Book-Rating.

- Dữ liệu “dính 1 cột”: đọc CSV sai khiến toàn bộ dòng nằm trong một cột, cần tách lại.

Tiền xử lý phải giải quyết các vấn đề này để đảm bảo dữ liệu sạch trước khi mô hình hóa.

3.3. Quy trình tiền xử lý chi tiết

3.3.1. Đọc dữ liệu và chuẩn hóa định dạng

- Đọc file với encoding phù hợp (ví dụ latin-1) để hạn chế lỗi ký tự.
- Kiểm tra số cột: nếu dataset bị đọc thành 1 cột, tiến hành tách lại theo delimiter đúng.
- Chuẩn hóa tên cột thống nhất giữa các file.

Ý nghĩa: đảm bảo dữ liệu vào pipeline có schema đúng, tránh sai lệch khi join/feature engineering.

3.3.2. Làm sạch dữ liệu (Data Cleaning)

- Loại bỏ dấu nháy " " dư thừa trong các cột text.
- Trim khoảng trắng đầu cuối.
- Chuyển kiểu dữ liệu:
    - User-ID sang số (int)
    - Book-Rating sang số (int/float)
    - Year-Of-Publication sang số (nếu cần dùng)
- Xóa các dòng bị thiếu giá trị quan trọng: User-ID, ISBN, Book-Rating.

Ý nghĩa: tránh lỗi khi tính thống kê, train model, và tránh tạo mapping sai.

3.3.3. Lọc explicit feedback

Vì mục tiêu là học từ điểm đánh giá thực:

- Giữ lại các rating > 0.
- Rating = 0 được xem như “không đánh giá/không thể hiện mức độ thích” (tùy dataset), nếu giữ lại sẽ làm nhiễu vì mô hình học rằng người dùng “ghét” nhiều sách mà thực tế chỉ là không đánh giá.

Ý nghĩa: làm tín hiệu học rõ ràng hơn, mô hình tập trung vào sở thích thực.

3.3.4. Giảm độ thưa (Sparsity Reduction) bằng lọc user/item

Dữ liệu gợi ý thường cực thưa. Nếu đưa toàn bộ vào, mô hình dễ:

- overfit vào user ít dữ liệu
- không học đủ thông tin cho item hiếm
- đánh giá không ổn định

Giải pháp:

- Chỉ giữ user có số lượng rating ≥ ngưỡng (ví dụ 10).
- Chỉ giữ sách có số lượng rating ≥ ngưỡng (ví dụ 10).

Ý nghĩa:

- Tăng "mật độ" dữ liệu hữu ích.
- Đảm bảo mỗi user/item đủ tín hiệu để mô hình học embedding ổn định.

3.3.5. Mã hóa user và item (Encoding)

Mô hình MF cần index liên tục:

- user_idx: map mỗi User-ID thành 0…n_users-1
- item_idx: map mỗi ISBN thành 0…n_items-1

Đồng thời lưu mapping ngược để hiển thị kết quả:

- idx_to_user
- idx_to_isbn

Ý nghĩa: giảm kích thước biểu diễn, tối ưu tốc độ và tương thích với embedding layer.

3.3.6. Chia tập train/test theo user (User-based split)

Nếu chia ngẫu nhiên toàn bộ tương tác, có thể xảy ra:

- một user chỉ xuất hiện ở test nhưng không có lịch sử ở train → không thể recommend
- hoặc test chứa tương tác "quá dễ" do trùng ngữ cảnh

Vì vậy dùng cách chia theo user:

- với mỗi user: giữ lại một phần rating làm test, phần còn lại làm train
- đảm bảo user nào cũng có lịch sử train

Ý nghĩa: phản ánh đúng bài toán thực tế: dự đoán hành vi tương lai dựa trên lịch sử quá khứ.

3.4. Kết quả sau tiền xử lý

Sau tiền xử lý, thu được:

- Tập ratings explicit đã lọc (giảm nhiễu)
- Tập filtered_ratings (giảm sparsity)
- Train set và Test set theo user
- Mapping user/item cho mô hình
- Dữ liệu sẵn sàng cho EDA và mô hình hóa

CHƯƠNG 4: PHƯƠNG PHÁP KHAI PHÁ DỮ LIỆU / MÔ HÌNH ML

4.1. Tổng quan hệ thống gợi ý

Hệ thống gợi ý sách thường có các nhóm phương pháp:

1. Popularity-based: gợi ý sách phổ biến nhất (baseline).
2. Content-based: gợi ý dựa trên nội dung/metadata (tác giả, thể loại, từ khóa).
3. Collaborative Filtering: gợi ý dựa trên hành vi cộng đồng (ai giống ai, sách nào hay được đánh giá chung).
4. Hybrid: kết hợp nhiều hướng.

Đề tài chọn Collaborative Filtering vì phù hợp với dữ liệu rating và thể hiện đúng tinh thần khai phá dữ liệu tương tác.

4.2. Baseline: Popularity-based Recommendation

Baseline là "mốc kiểm chứng". Dù không cá nhân hóa, nó giúp:

- kiểm tra dữ liệu đã sạch và hợp lý
- xem sách nào có rating cao và nhiều lượt đánh giá
- so sánh với mô hình cá nhân hóa

Cách tính độ phổ biến thường kết hợp:

- số lượt rating (count)
- điểm trung bình (mean)

Một scoring đơn giản:

- score = avg_rating × log(1 + rating_count)
  Log giúp tránh việc sách cực phổ biến áp đảo hoàn toàn, đồng thời vẫn ưu tiên các sách có đủ lượt đánh giá.

Ưu điểm: đơn giản, nhanh, chạy tốt với user mới.
Nhược điểm: không cá nhân hóa.

4.3. Collaborative Filtering và Matrix Factorization

4.3.1. Bài toán ma trận user–item

Ta có ma trận R kích thước (n_users × n_items), trong đó:

- R(u,i) = rating nếu user u đánh giá item i

- phần lớn ô trống vì user không thể đánh giá hết mọi sách

Mục tiêu: ước lượng các ô trống, đặc biệt là các item user chưa đánh giá.

4.3.2. Ý tưởng Matrix Factorization

MF giả định rằng "sở thích" có thể biểu diễn trong một không gian tiềm ẩn kích thước k (embedding dimension).

- Mỗi user u có vector $p_u \in R^k$
- Mỗi item i có vector $q_i \in R^k$

Dự đoán rating:

- $\hat{y}(u,i) = p_u \cdot q_i + b_u + b_i + b$
  Trong đó:
- b_u: thiên lệch user (user khó tính/dễ tính)
- b_i: thiên lệch item (sách thường được đánh giá cao/thấp)
- b: bias toàn cục

4.3.3. Hàm mất mát và regularization

Huấn luyện bằng tối ưu sai số dự đoán:

- MSELoss: trung bình bình phương $(y - \hat{y})$

Để tránh overfitting:

- weight decay (L2 regularization) giúp embedding không phình quá lớn và mô hình tổng quát hóa tốt hơn.

4.3.4. Vì sao MF phù hợp bài toán này

- Học được "taste" tiềm ẩn của user và "chất" của sách mà không cần metadata.
- Hiệu quả tốt trên dữ liệu rating thưa nếu đã lọc hợp lý.
- Dễ mở rộng: tăng embedding_dim, thêm regularization, thêm implicit feedback.

4.4. Chiến lược huấn luyện và lựa chọn siêu tham số (giải thích, không code)

- Embedding dimension (k):
  k nhỏ quá → mô hình thiếu khả năng biểu diễn; k lớn quá → dễ overfit và tốn tài nguyên. Thực nghiệm thường chọn 50–200 tùy dữ liệu.
- Batch size:
  batch lớn giúp GPU chạy nhanh hơn, nhưng cần cân bằng RAM/VRAM.

- Learning rate:
  LR quá lớn gây dao động, quá nhỏ học chậm. Thường dùng Adam/AdamW để ổn định.
- Epochs:
  Không nên quá nhiều nếu dữ liệu không lớn; có thể kết hợp early stopping (nếu làm nâng cao).
- Clipping rating:
  Sau dự đoán, có thể cắt về [1,10] để hợp thang điểm.

4.5. Sinh gợi ý Top-N (Recommendation Generation)

Quy trình gợi ý cho một user:

1. Lấy danh sách sách user đã đánh giá → tập "seen"
2. Lấy các sách còn lại → tập "candidate"
3. Dự đoán điểm ŷ cho candidate
4. Sắp xếp giảm dần theo ŷ
5. Trả về Top-N và join Books.csv để hiển thị title/author/year…

CHƯƠNG 5: KẾT QUẢ VÀ ĐÁNH GIÁ MÔ HÌNH

5.1. Mục tiêu đánh giá

Đánh giá hệ thống gợi ý không chỉ dừng ở việc mô hình dự đoán điểm số gần đúng với thực tế, mà còn cần phản ánh chất lượng danh sách gợi ý (Top-K/Top-N) khi triển khai cho người dùng. Vì vậy, đề tài sử dụng đồng thời hai nhóm chỉ số:

- Nhóm đánh giá dự đoán rating: RMSE và MAE, đo độ lệch giữa điểm dự đoán và điểm đánh giá thật trên tập test.
- Nhóm đánh giá gợi ý Top-K: Precision@K và Recall@K, đo mức độ "đúng" của danh sách gợi ý khi chỉ xem top K sản phẩm đầu tiên.

Cách đánh giá này giúp nhìn hệ thống ở cả 2 góc độ: (1) đoán đúng điểm, và (2) gợi ý đúng thứ tự ưu tiên.

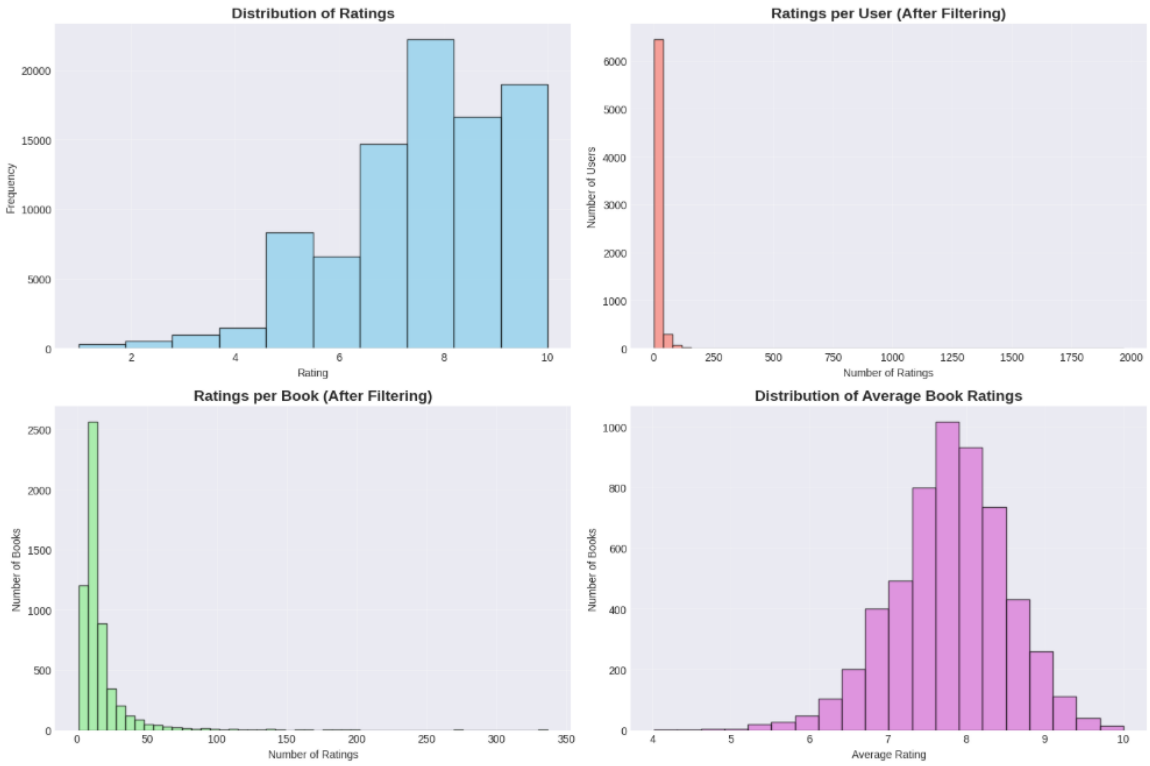
5.2. Kết quả EDA và các nhận xét quan trọng

Sau khi thực hiện tiền xử lý gồm: làm sạch dữ liệu, lọc explicit feedback (rating > 0), lọc độ thưa bằng ngưỡng số lượt đánh giá tối thiểu cho user và sách, dữ liệu thu được có chất lượng tốt hơn để đưa vào huấn luyện mô hình.

Các kết quả EDA cho thấy:

- Phân phối rating: thể hiện xu hướng chấm điểm của người dùng. Nếu phân phối tập trung ở các mức cao (ví dụ 7–10), dữ liệu thể hiện rõ thiên hướng "thích" hơn "không thích"; điều này ảnh hưởng đến cách chọn threshold cho Top-K.
- Số lượt đánh giá theo user: thường lệch phải (đa số user đánh giá ít, một số user đánh giá rất nhiều). Những user đánh giá quá ít dễ gây nhiễu, vì mô hình khó học được sở thích.
- Số lượt đánh giá theo sách: cũng lệch phải (một số sách cực phổ biến có nhiều rating). Điều này dẫn tới hiện tượng "thiên lệch phổ biến" (popularity bias), mô hình dễ ưu tiên sách nổi tiếng nếu không kiểm soát.
- Phân phối điểm trung bình theo ISBN: giúp nhận biết mức độ "dễ được chấm cao" của sách, đồng thời cho thấy nếu sách có ít rating thì điểm trung bình có thể thiếu ổn định.

Kết luận từ EDA: việc lọc user/item có số rating tối thiểu là cần thiết để (1) giảm nhiễu, (2) tăng mật độ dữ liệu hữu ích, và (3) giúp mô hình embedding học ổn định hơn.



Hình 1 Kết quả EDA sau tiền xử lý (phân phối rating, số rating theo user/sách, phân phối điểm trung bình theo sách).

5.3. Baseline Popularity: kết quả và phân tích

Để có mốc so sánh, đề tài xây dựng baseline theo độ phổ biến (Popularity-based). Ý tưởng là xếp hạng sách theo mức độ được đánh giá nhiều và được đánh giá cao. Baseline giúp:

- kiểm tra nhanh dữ liệu có hợp lý hay không,
- tạo "chuẩn tham chiếu" để so sánh với mô hình cá nhân hóa,
- hoạt động tương đối tốt trong trường hợp cold-start (user mới).

Trong notebook, danh sách baseline được lưu thành file baseline_popular_books.csv, có thể trích ra Top 10 để đưa vào báo cáo.

Nhận xét: baseline thường gợi ý các sách nổi tiếng, phù hợp với số đông nhưng không phản ánh đúng từng cá nhân. Vì vậy baseline dùng để so sánh chứ không phải mục tiêu cuối.

Bảng 1 Top 10 sách theo baseline popularity.

	ISBN	rating_count	avg_rating	popularity	Book-Title	Book-Author	Year-Of-Publication
2	385504209	276	8.565217391	48.17093	The Da Vinci Code	Dan Brown	2003
3	316666343	337	8.246290801	48.01853	The Lovely Bones: A Novel	Alice Sebold	2002
4	059035342X	184	8.994565217	46.95483	Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))	J. K. Rowling	1999
5	043935806X	158	9.056962025	45.90887	Harry Potter and the Order of the Phoenix (Book 5)	J. K. Rowling	2003
6	679781587	179	8.636871508	44.8509			
7	142001740	179	8.581005587	44.56079	The Secret Life of Bees	Sue Monk Kidd	2003
8	446310786	136	9.051470588	44.53306	To Kill a Mockingbird	Harper Lee	1988
9	312195516	196	8.382653061	44.28726	The Red Tent (Bestselling Backlist)	Anita Diamant	1998
10	439139597	105	9.295238095	43.34778	Harry Potter and the Goblet of Fire (Book 4)	J. K. Rowling	2000
11	439136350	110	9.2	43.32768	Harry Potter and the Prisoner of Azkaban (Book 3)	J. K. Rowling	1999
12	446672211	162	8.296296296	42.25926	Where the Heart Is (Oprah's Book Club (Paperback))	Billie Letts	1998
13	439064872	116	8.862068966	42.20271	Harry Potter and the Chamber of Secrets (Book 2)	J. K. Rowling	2000
14	439064864	106	8.962264151	41.87913	Harry Potter and the Chamber of Secrets (Book 2)	J. K. Rowling	1999
15	671027360	157	8.222929936	41.62936	Angels & Demons	Dan Brown	2001
16	60928336	167	8.101796407	41.51331	Divine Secrets of the Ya-Ya Sisterhood: A Novel	Rebecca Wells	1997
17	385484518	115	8.695652174	41.33557	Tuesdays with Morrie: An Old Man, a Young Man, and Life's Greatest Lesson	MITCH ALBOM	1997
18	590353403	93	9.010752688	40.93851	Harry Potter and the Sorcerer's Stone (Book 1)	J. K. Rowling	1998
19	345342968	101	8.772277228	40.57154	Fahrenheit 451	RAY BRADBURY	1987
20	156027321	139	8.201438849	40.52858	Life of Pi	Yann Martel	2003
21	439136369	87	9.011494253	40.34749	Harry Potter and the Prisoner of Azkaban (Book 3)	J. K. Rowling	2001

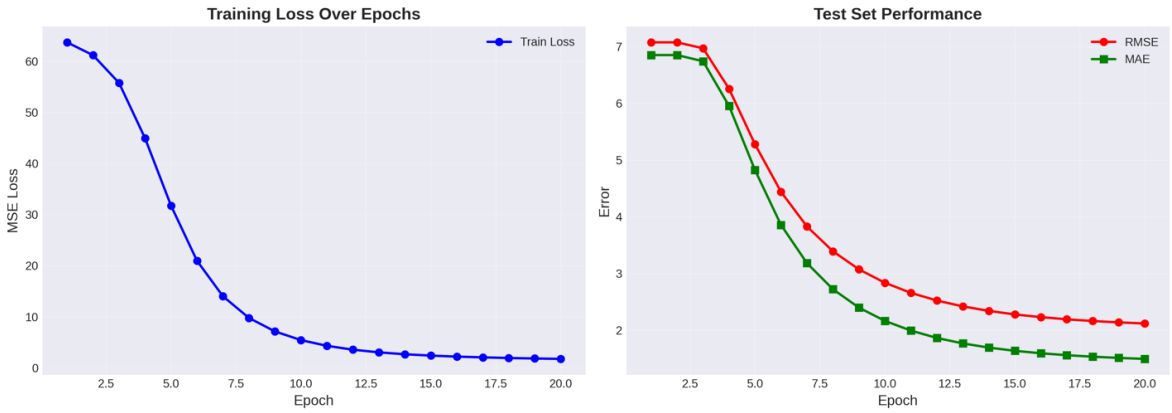
5.4. Đánh giá dự đoán rating: RMSE và MAE

Đề tài dùng hai chỉ số phổ biến cho dự đoán rating:

- MAE (Mean Absolute Error): trung bình sai số tuyệt đối $|y - \hat{y}|$. MAE dễ diễn giải vì biểu thị mức lệch trung bình trực tiếp theo thang điểm rating.
- RMSE (Root Mean Squared Error): căn bậc hai của trung bình bình phương sai số. RMSE nhạy với sai số lớn, nên phản ánh rõ hơn các dự đoán "lệch mạnh".

Ý nghĩa:

- RMSE/MAE càng nhỏ càng tốt.
- Nếu RMSE giảm dần theo epoch rồi chững lại, có thể mô hình đã gần hội tụ.
- Nếu RMSE/MAE ở train tốt nhưng test xấu đi, có thể xảy ra overfitting (cần regularization hoặc giảm epochs).



Hình 2 Lịch sử huấn luyện mô hình Matrix Factorization (Loss/RMSE/MAE theo epoch).

Bảng 2 Kết quả RMSE và MAE trên tập test.

	A	B	C
1	Metric	Value	
2	RMSE	2.118246342	
3	MAE	1.494322062	
4	Precision@5	0.864516129	
5	Recall@5	0.693695213	
6	Precision@10	0.871014493	
7	Recall@10	0.759526171	

5.5. Đánh giá gợi ý Top-K: Precision@K và Recall@K

Ngoài dự đoán rating, hệ gợi ý quan trọng nhất là khả năng đưa ra danh sách Top-K phù hợp. Để tính Precision@K và Recall@K, cần định nghĩa item "relevant" trong tập test. Với dữ liệu explicit rating, đề tài xem một sách là relevant nếu rating thật ≥ threshold (ví dụ threshold = 7).

- Precision@K: trong K sách mô hình gợi ý, có bao nhiêu sách thật sự relevant.
- Recall@K: trong tất cả sách relevant của user ở test, mô hình gợi ý được bao nhiêu sách.

Nhận xét quan trọng:

- Precision@K cao nghĩa là danh sách gợi ý "ít rác", user ít gặp đề xuất không phù hợp.
- Recall@K cao nghĩa là mô hình "không bỏ sót" nhiều sách mà user thích.

- Trong dữ liệu thưa, mỗi user có thể có ít item relevant ở test, nên Recall@K có thể dao động, cần ghi chú ngắn về hạn chế này.

Phần báo cáo nên trình bày:

- Precision@5, Recall@5
- Precision@10, Recall@10
  và nhận xét so sánh: khi K tăng, recall thường tăng nhưng precision có thể giảm.

5.6. Demo gợi ý cho người dùng và phân tích định tính

Để minh họa trực quan cho hệ thống, cần trình bày ít nhất 1 trường hợp demo:

- Chọn một User-ID có số lượng rating đủ lớn (sau khi lọc).
- Sinh danh sách Top 10 sách đề xuất (loại bỏ sách user đã đánh giá).
- Join với Books.csv để hiển thị thông tin: tên sách, tác giả, năm xuất bản, nhà xuất bản…
- Đưa ra nhận xét định tính: các sách gợi ý có xu hướng cùng tác giả, cùng nhóm chủ đề, hoặc tương đồng về mức phổ biến/điểm số dự đoán.

Bảng 3 Top 10 sách đề xuất cho User

	ISBN	Predicted_Rat	Book-Title	Book-Author	Year-Of-Publication	Publisher
2	380788624	10.313593	Cryptonomicon	Neal Stephenson	2000	Perennial
3	345310020	9.961602	Chronicle of a Death Foretold	GABRIEL GARCIA MARQUEZ	1984	Ballantine Books
4	2070360024	9.917837	L'Etranger (Collection Folio, 2)	Albert Camus	1990	Gallimard Jeunesse
5	3522128001	9.793398	Die unendliche Geschichte: Von A bis Z	Michael Ende	1979	Thienemann
6	811802981	9.57119	The Golden Mean: In Which the Extraordinary Correspondence of Griffin & Sabine Concludes	Nick Bantock	1993	Chronicle Books
7	425109720	9.569666	Patriot Games (Jack Ryan Novels)	Tom Clancy	1992	Berkley Publishing Group
8	618002235	9.533818	The Two Towers (The Lord of the Rings, Part 2)	J. R. R. Tolkien	1999	Houghton Mifflin Company
9	8433969978	9.522337	El Libro de Las Ilusiones	Paul Auster	2003	Anagrama
10	8445071416	9.501603	El Hobbit	J. R. R. Tolkien	1991	Minotauro
11	684826976	9.483718	Undaunted Courage: Meriwether Lewis Thomas Jefferson and the Opening of the American West	Stephen Ambrose	1997	Simon & Schuster

5.7. Tổng hợp kết quả và hạn chế

Tổng hợp kết quả:

- Baseline popularity tạo mốc tham chiếu và hoạt động ổn với cold-start, nhưng không cá nhân hóa.
- Mô hình Matrix Factorization cho khả năng cá nhân hóa tốt hơn khi user có đủ lịch sử rating, thể hiện qua RMSE/MAE và Top-K metrics.
- Hệ thống có thể xuất kết quả ra file và demo gợi ý Top-N theo User-ID.
-

Hạn chế:

- Cold-start: user mới hoặc sách mới ít dữ liệu khiến mô hình khó gợi ý tốt.
- Thiên lệch phổ biến: sách nổi tiếng nhiều rating dễ được ưu tiên.
- Chỉ dùng ratings: chưa khai thác sâu nội dung sách (tiêu đề, tác giả, publisher) để tăng chất lượng gợi ý

CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. Kết luận

Đề tài đã xây dựng được hệ thống gợi ý sách dựa trên dữ liệu đánh giá người dùng, đáp ứng pipeline chuẩn:

- làm sạch dữ liệu và xử lý các lỗi định dạng CSV
- lọc explicit feedback và giảm sparsity bằng ngưỡng user/item
- EDA mô tả dữ liệu sau tiền xử lý
- xây dựng baseline popularity và mô hình MF
- đánh giá bằng RMSE/MAE và Precision/Recall@K
- demo gợi ý Top-N sách cho một User-ID

Hệ thống cho thấy mô hình MF có khả năng cá nhân hóa tốt hơn baseline khi người dùng có đủ lịch sử đánh giá.

6.2. Hướng phát triển

- Hybrid recommender: kết hợp MF với content-based từ Book-Title/Author/Publisher (TF-IDF hoặc embedding).
- Xử lý cold-start: dùng metadata để gợi ý cho user mới hoặc item mới.
- Bổ sung chỉ số xếp hạng: NDCG@K, MAP@K để đo chất lượng ranking sâu hơn.
- Đánh giá theo thời gian: nếu có timestamp, dùng temporal split để phản ánh thực tế tốt hơn.
- Triển khai giao diện: Streamlit/Gradio cho phép nhập User-ID, hiển thị top-N và đồ thị EDA.

CHƯƠNG 7: TÀI LIỆU THAM KHẢO

1. Kaggle Dataset: Books Dataset (saurabhbagchi/books-dataset).
2. Ricci, F., Rokach, L., Shapira, B. *Recommender Systems Handbook*.
3. Koren, Y., Bell, R., Volinsky, C. (2009). *Matrix Factorization Techniques for Recommender Systems*.
4. Aggarwal, C. C. (2016). *Recommender Systems: The Textbook*.