TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

======*======**

ĐỒ ÁN TỐT NGHIỆP

TÍCH HỢP HỆ THỐNG ĐỀ XUẤT SÁCH MLOPS CHO WEBSITE BÁN SÁCH BOOKIES

GVHD	ThS. Lê Thị Thủy
Lớp	KHMT01
Sinh viên thực hiện	Nguyễn Văn Việt - 2021608149

Hà Nội, năm 2025

LỜI CẢM ƠN

Trong quá trình học tập và rèn luyện tại Đại Học Công Nghiệp Hà Nội và trong thời gian thực hiện thực tập tốt nghiệp đề tài "Tích hợp hệ thống đề xuất sách MLOps cho Bookies", em xin gửi lời cảm ơn về sự giúp đỡ của các thầy cô, giảng viên, cán bộ trong Khoa Công nghệ thông tin đã giúp em có được kiến thức và hoàn thiện đề tài.

Đặc biệt, em xin gửi lời cảm ơn tới cô Lê Thị Thủy, cô đã trực tiếp hướng dẫn và chỉ bảo, trang bị kiến thức cho em trong suốt quá trình tìm hiểu và hoàn thành đề tài.

Cuối cùng, với sự cố gắng của bản thân nhưng với vốn kiến thức kỹ năng và kinh nghiệm xây dựng trang web còn yếu, đề tài của em không thể tránh khỏi những thiếu sót trong quá trình hoàn thiện. Vì vậy em rất mong nhận được nhận được sự đánh giá và nhận xét của thầy cô để em có thể khắc phục những thiếu sót của đề tài, giúp em nâng cao kiến thức của bản thân và hơn nữa là phục vụ cho công việc sau này.

Em xin chân thành cảm ơn

MỤC LỤC

DANH MỤC CÁC THUẬT NGỮ, KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT

Viết tắt	Ý nghĩa
MLOPS	Machine Learning Operations
API	Application Programming Interface
FM	Factorization Machine
DNN	Deep Neuron Network
TPR	Machine Learning Operations
FPR	Application Programming Interface
ROC	Factorization Machine
AUC	Deep Neuron Network
DevOps	Machine Learning Operations
ML	Application Programming Interface
ReLU	Factorization Machine
JSON	Deep Neuron Network
CI/CD	Continuous Integration/Continuous Deployment

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

MỞ ĐẦU

Trong thời đại công nghệ số, nhu cầu cá nhân hóa trải nghiệm người dùng ngày càng trở nên quan trọng, đặc biệt trong lĩnh vực thương mại điện tử và nội dung số. Các hệ thống đề xuất (Recommendation Systems) đã trở thành một trong những công nghệ cốt lõi giúp nâng cao mức độ hài lòng của người dùng và tăng hiệu quả kinh doanh. Website Bookies, một nền tảng bán sách trực tuyến, đang hướng đến việc ứng dụng trí tuệ nhân tạo để cải thiện khả năng gợi ý sách phù hợp cho từng người dùng, từ đó gia tăng thời gian tương tác, tỷ lệ mua hàng và sự trung thành với nền tảng.

Tuy nhiên, việc xây dựng một mô hình đề xuất hiệu quả không chỉ dừng lại ở khâu huấn luyện và triển khai mô hình. Hệ thống cần được giám sát, cập nhật liên tục và tích hợp chặt chẽ với quy trình vận hành thực tế. Do đó, việc áp dụng MLOps (Machine Learning Operations) – một tập hợp các quy trình và công cụ nhằm tự động hóa toàn bộ vòng đời của mô hình học máy – là một hướng tiếp cận hiện đại, giúp đảm bảo hiệu suất, tính ổn định và khả năng mở rộng của hệ thống đề xuất.

Đề tài "Xây dựng hệ thống MLOps đề xuất sách cho website Bookies" nhằm mục tiêu thiết kế và phát triển một hệ thống đề xuất tích hợp MLOps, bao gồm các bước từ thu thập dữ liệu, huấn luyện mô hình, triển khai, giám sát đến cập nhật liên tục. Qua đó, đề tài không chỉ mang lại giá trị thực tiễn cho Bookies mà còn góp phần nghiên cứu và ứng dụng các xu hướng công nghệ tiên tiến trong lĩnh vực trí tuệ nhân tạo và kỹ thuật phần mềm.

CHƯƠNG 1 KHẢO SÁT HỆ THỐNG

1.1 Tổng quan về đề tài.

1.1.1 Giới thiệu chung.

Trong thời đại công nghệ số, khối lượng thông tin và sản phẩm trên các nền tảng thương mại điện tử ngày càng gia tăng, dẫn đến một thách thức lớn đối với người dùng: tìm kiếm sản phẩm phù hợp một cách nhanh chóng và hiệu quả. Chính vì vậy, các hệ thống gợi ý (Recommendation Systems) đã trở thành một thành phần quan trọng không thể thiếu trong việc nâng cao trải nghiệm người dùng và tối ưu hóa hiệu suất kinh doanh.

1.1.2 Tính cấp thiết của đề tài

Đề tài "Tích hợp hệ thống đề xuất cho website Bookies" hướng đến việc nghiên cứu, xây dựng và tích hợp một hệ thống gợi ý sách thông minh vào nền tảng thương mại điện tử Bookies – một website chuyên cung cấp sách cho nhiều đối tượng người dùng.

Thông qua việc áp dụng các thuật toán học máy, kết hợp với quy trình triển khai MLOps nhằm đảm bảo khả năng mở rộng, cập nhật và giám sát mô hình hiệu quả, hệ thống đề xuất sẽ giúp người dùng:

- Khám phá những cuốn sách phù hợp với sở thích cá nhân.
- Tiết kiệm thời gian tìm kiếm.
- Và nâng cao sự hài lòng trong quá trình trải nghiệm.

Bên cạnh đó, từ góc độ doanh nghiệp, hệ thống đề xuất sách giúp website Bookies:

- Tăng tỷ lệ tương tác và chuyển đổi.
- Thúc đẩy doanh thu.
- Giữ chân người dùng trung thành.
- Tạo lợi thế cạnh tranh so với các nền tảng cùng lĩnh vực.

Đề tài không chỉ tập trung vào thuật toán đề xuất mà còn chú trọng vào việc triển khai hệ thống trong môi trường thực tế, bao gồm: kiến trúc hệ thống,

pipeline xử lý dữ liệu, tích hợp frontend/backend, và đo lường hiệu quả hệ thống.

1.1.3 Mục tiêu nghiên cứu.

Đề tài hướng đến việc phát triển một hệ thống đề xuất thông minh, từ đó mang lại lợi ích cho cả người dùng lẫn nền tảng thương mại điện tử Bookies. Các mục tiêu cụ thể của nghiên cứu bao gồm:

- Cá nhân hóa trải nghiệm người dùng, giúp khách hàng tiếp cận dễ dàng với những nội dung sách yêu thích.
- Tăng doanh số bán hàng khi gợi ý những cuốn sách phù hợp với khách hàng
- Tối ưu hóa quy trình vận hành của hệ thống bằng cách tích hợp hệ thống MLOPS tự động
- Giúp của hàng Bookies quản lý danh mục sản phẩm tốt hơn, phân loại sách dựa trên nhu cầu, và dự đoán được các xu hướng sách sắp tới, từ đó giúp tối ưu hóa việc nhập hàng và phân phối.

1.1.4 Đối tượng và phạm vi nghiên cứu.

Đối tượng nghiên cứu của đề tài là hệ thống đề xuất sách dựa trên các kỹ thuật học máy và khai thác dữ liệu người dùng. Cụ thể:

- Mô hình gợi ý gợi ý: DeepFM.
- Quy trình triển khai hệ thống học máy trong môi trường thực tế, bao gồm cả MLOps.
- Tập dữ liệu người dùng và sản phẩm từ website Bookies: bao gồm thông tin sách, hành vi người dùng (đánh giá của người dùng)
- Kiến trúc hệ thống tích hợp giữa mô hình đề xuất và nền tảng website Bookies.

Đề tài tập trung vào các nội dung sau:

- Xây dựng hệ thống gợi ý sách dựa trên dữ liệu có sẵn của website Bookies.

- Tích hợp hệ thống đề xuất với website Bookies thông qua API, đảm bảo hoạt động ổn định và phản hồi nhanh.
- Triển khai và quản lý mô hình học máy sử dụng quy trình MLOps, bao gồm: huấn luyện, kiểm thử, triển khai, giám sát và cập nhật mô hình.
- Đánh giá hiệu quả mô hình đề xuất thông qua các chỉ số như: Precision, Recall, F1-score...

1.1.5 Phương pháp nghiên cứu.

Để đạt được các mục tiêu nghiên cứu đã đề ra, đề tài sử dụng các phương pháp nghiên cứu sau:

- Phương pháp thu thập và phân tích dữ liệu: sử dụng bộ dữ liệu tương tác của người dùng trên Kaggle.
- Phương pháp xây dựng mô hình đề xuất: nghiên cứu các kỹ thuật xây dựng hệ thống gợi ý như lọc cộng tác, lọc theo nội dung và lọc kết hợp.
- Phương pháp triển khai và tích hợp hệ thống: sử dụng kiến trúc microservices để triển khai mô hình gợi ý độc lập với hệ thống chính của website Bookies. Xây dựng RESTful API để kết nối mô hình với frontend. Đồng thời, áp dụng quy trình MLOps để tự động hóa pipeline gồm các bước: tiền xử lý, huấn luyện, triển khai, giám sát và cập nhật mô hình.
- Phương pháp đánh giá mô hình: Sử dụng các chỉ số phổ biến để đánh giá độ chính xác của mô hình như Precision, Recall, F1-score, AUC, Logloss, Accuracy

1.1.6 Tính thực tiễn của đề tài

Dự án "Tích hợp hệ thống đề xuất cho website Bookies" mang lại nhiều lợi ích thực tiễn quan trọng trong bối cảnh thương mại điện tử và bán sách trực tuyến hiện nay. Một số lợi ích chính có thể kể đến như sau:

- Hệ thống đề xuất giúp người dùng giải quyết vấn đề quá tải thông tin khi phải đối mặt với một loạt các sản phẩm trên các nền tảng thương mại điện tử. Bằng cách cung cấp các gợi ý sách phù hợp với sở thích cá nhân,

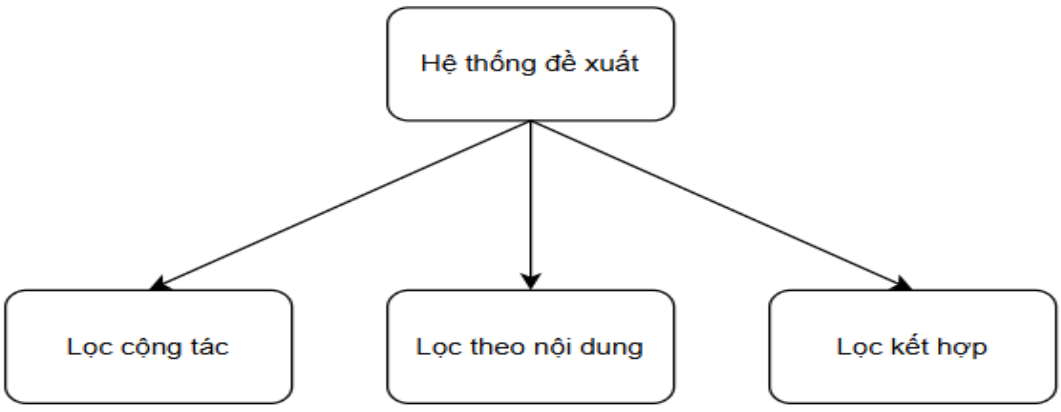
hệ thống không chỉ tiết kiệm thời gian mà còn nâng cao trải nghiệm người dùng, đặc biệt là đối với những người yêu sách có gu cá nhân rõ rệt.

- Từ góc độ doanh nghiệp, việc áp dụng hệ thống đề xuất chính xác có thể thúc đẩy doanh số bán hàng thông qua việc khuyến khích khách hàng mua thêm sách. Hơn nữa, hệ thống này còn cung cấp dữ liệu phân tích hành vi người dùng, hỗ trợ các bộ phận kinh doanh trong việc xây dựng chiến lược marketing hiệu quả, quản lý tồn kho và nhập hàng phù hợp với xu hướng tiêu dùng.
- Hệ thống có khả năng mở rộng và áp dụng cho nhiều lĩnh vực khác như thời trang, mỹ phẩm, thiết bị điện tử nhờ vào kiến trúc modular và ứng dụng MLOps. Điều này giúp doanh nghiệp dễ dàng nâng cấp và triển khai hệ thống mà không gây gián đoạn hoạt động kinh doanh.

1.2 Giới thiệu chung về hệ thống đề xuất và tổng quan về MLOps

1.2.1 Hệ thống đề xuất

Hệ thống đề xuất (Recommendation System) là một công cụ quan trọng trong việc cá nhân hóa trải nghiệm người dùng trên các nền tảng thương mại điện tử và nội dung số. Với sự hỗ trợ của trí tuệ nhân tạo và học máy, các hệ thống này có thể đưa ra các đề xuất dựa trên hành vi và sở thích của người dùng.



Hình 1.1 Các thuật toán đề xuất

- Các phương pháp phổ biến:
  - Hệ thống đề xuất dựa trên nội dung: phương pháp đề xuất nội dung dựa trên sự tương tự giữa các mục nội dung. Thuật toán này sử dụng thông tin về nội dung của các mục để đề xuất các mục tương tự. Các phương pháp lọc nội dung bao gồm sử dụng thông tin từ người dùng (user-based content filtering) và sử dụng thông tin từ sản phẩm (item-based content filtering).
  - Hệ thống đề xuất dựa trên lọc cộng tác: là phương pháp sử dụng thông tin từ người dùng khác để đề xuất nội dung tương tự. Thuật toán này dựa trên nguyên lý rằng những người dùng có sở thích tương tự sẽ thích những nội dung tương tự. Các phương pháp lọc cộng tác bao gồm lọc cộng tác dựa trên người dùng (user-based collaborative filtering) và lọc cộng tác dựa trên sản phẩm (item-based collaborative filtering).
  - Hệ thống đề xuất kết hợp: phương pháp kết hợp cả hai phương pháp trên để đề xuất nội dung. Thuật toán này sử dụng cả thông tin từ người dùng và thông tin về nội dung của các mục để đề xuất nội dung phù hợp nhất. Các phương pháp lọc kết hợp bao gồm việc sử dụng kỹ thuật học máy để kết hợp cả hai phương pháp.
- Vai trò của hệ thống đề xuất:
  - Tăng cường trải nghiệm người dùng: Gợi ý những cuốn sách phù hợp giúp người dùng tiết kiệm thời gian tìm kiếm và khám phá được những đầu sách giá trị.
  - Tăng tỷ lệ tương tác và doanh thu: Những đề xuất chính xác thúc đẩy người dùng mua hoặc đọc nhiều sách hơn.
  - Cá nhân hóa nội dung: Hệ thống hiểu người dùng hơn qua dữ liệu lịch sử và hành vi, từ đó tạo ra trải nghiệm cá nhân hóa.

1.2.2 Tổng quan về MLOps

MLOps (Machine Learning Operations) là tập hợp các thực hành, quy trình và công cụ nhằm tối ưu hóa việc triển khai, giám sát, và duy trì các mô hình học máy trong môi trường thực tế.

- Vai trò của MLOps:
  - Tự động hóa quy trình huấn luyện và triển khai mô hình: Giảm sự phụ thuộc vào thao tác thủ công.
  - Đảm bảo tái sử dụng và khả năng mở rộng: Cho phép dễ dàng nâng cấp mô hình hoặc thay đổi dữ liệu đầu vào.
  - Giám sát hiệu suất mô hình theo thời gian: Theo dõi độ chính xác của mô hình gợi ý sau khi được tích hợp trên website Bookies.
- Các thành phần chính của quy trình MLOps:
  - Data Pipline: Thu thập, xử lý, lưu trữ dữ liệu người dùng và dữ liệu sách.
  - Training Pipline: Huấn luyện các mô hình đề xuất với các thuật toán như lọc cộng tác, lọc theo nội dung và lọc kết hợp.
  - Serving Pipline: Triển khai mô hình dưới dạng API để tích hợp với website Bookies.
  - Monitoring and Retraining: Theo dõi hiệu suất, đánh giá độ chính xác và tái huấn luyện mô hình khi cần thiết để duy trì độ chính xác.

1.2.3 Sơ đồ hệ thống đề xuất sách.

Hệ thống MLOps đề xuất sách gồm có 4 thành phần chính bao gồm: Data pipeline (Luồng dữ liệu), Trainning Pipeline (Luồng huấn luyện mô hình), Serving Pipeline (Luồng triển khai), Model Registry (Đăng ký mô hình), Observability (Giám sát hệ thống)

Bảng 1.1 Cấu trúc hệ thống MLOps

STT	Thành phần	Mục đích	Quy trình
1	Data Pipeline	Chuyển đổi dữ liệu hành vi người	

		dùng thành các đặc trưng cho mô hình máy học và lưu trữ chúng	
2	Trainning Pipeline	Huấn luyện mô hình học máy	1. Kéo dữ liệu đã đã được làm sạch để sử dụng trong huấn luyện. 2. Huấn luyện mô hình học máy 3. Lưu mô hình và số liệu liên quan vào bằng Mlflow
3	Serving Pipeline		1. Tải mô hình từ MLflow đề thực hiện dự đoán 2. Gửi đi API chứa dữ liệu đề xuất.

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

2.1 Tổng quan về thuật toán đề xuất.

Badrul Sarwar, nhà nghiên cứu tại GroupLens Research, đã nhấn mạnh rằng: “Hệ thống đề xuất là một công nghệ được thiết kế để cá nhân hóa các trải nghiệm trực tuyến bằng cách tự động dự đoán những gì người dùng có thể quan tâm, dựa trên hành vi của họ hoặc của những người dùng tương tự.”. Yehuda Koren, nhà khoa học tại Google và từng làm việc tại Netflix, bổ sung rằng: “Hệ thống đề xuất không chỉ là công cụ để gợi ý, mà còn là cách giúp người dùng khám phá nội dung hoặc sản phẩm mới mà họ có thể không nhận ra là mình cần”. Paul Resnick, giáo sư tại Đại học Michigan, đã chỉ ra rằng:”Hệ thống đề xuất là một giải pháp cho vấn đề quá tải thông tin, bằng cách chọn lọc và gợi ý những nội dung phù hợp với sở thích và nhu cầu của người dùng”. Joseph A. Konstan, đồng sáng lập GroupLens Research, nhấn mạnh rằng: “Hệ thống đề xuất là một công cụ để tối ưu hóa sự tương tác giữa người dùng và nội dung, bằng cách cung cấp các gợi ý dựa trên dữ liệu lịch sử và mô hình hành vi”. Cuối cùng, Andreas Hotho, một chuyên gia về khai phá dữ liệu, cho rằng: “Hệ thống đề xuất là một phần quan trọng của các hệ thống thông tin, đóng vai trò quyết định trong việc hỗ trợ quyết định của người dùng và cá nhân hóa trải nghiệm”.

Tóm lại, hệ thống đề xuất (Recommendation System) là một công nghệ thông minh được thiết kế để cá nhân hóa trải nghiệm người dùng thông qua việc tự động gợi ý các sản phẩm, dịch vụ, hoặc nội dung dựa trên sở thích và hành vi của họ. Mục tiêu chính của hệ thống này là dự đoán những gì người dùng có thể quan tâm, giúp họ khám phá những tùy chọn mới mà có thể họ chưa nhận ra.

2.1.1 Phân loại lọc cộng tác

Lọc cộng tác là phương pháp sử dụng thông tin từ người dùng khác để đề xuất nội dung tương tự. Thuật toán này dựa trên nguyên lý rằng những người dùng có sở thích tương tự sẽ thích những nội dung tương tự. Các phương pháp lọc cộng tác bao gồm lọc cộng tác dựa trên người dùng (user-based

collaborative filtering) và lọc cộng tác dựa trên sản phẩm (item-based collaborative filtering).

Lọc nội dung là phương pháp đề xuất nội dung dựa trên sự tương tự giữa các mục nội dung. Thuật toán này sử dụng thông tin về nội dung của các mục để đề xuất các mục tương tự. Các phương pháp lọc nội dung bao gồm sử dụng thông tin từ người dùng (user-based content filtering) và sử dụng thông tin từ sản phẩm (item-based content filtering).

Lọc kết hợp là phương pháp kết hợp cả hai phương pháp trên để đề xuất nội dung. Thuật toán này sử dụng cả thông tin từ người dùng và thông tin về nội dung của các mục để đề xuất nội dung phù hợp nhất. Các phương pháp lọc kết hợp bao gồm việc sử dụng kỹ thuật học máy để kết hợp cả hai phương pháp.

2.1.2 Các kỹ thuật đánh giá hiệu quả hệ thống đề xuất.

2.1.2.1 *Accuracy (Độ chính xác)*

Độ chính xác là tỷ lệ phần trăm các mẫu dữ liệu được phân loại chính xác bởi mô hình. Accuracy được tính bằng công thức như sau**:**

$$Accuracy = (TP + TN)/ (TP + TN + FP + FN)$$

Trong đó:

- TP (True Positive): Dự đoán đúng là dương (1)
- TN (True Negative): Dự đoán đúng là âm (0)
- FP (False Positive): Dự đoán sai là dương
- FN (False Negative): Dự đoán sai là âm

Độ chính xác là thước đo trực quan và dễ hiểu nhất về hiệu suất phân loại của mô hình. Tuy nhiên, nó có thể bị ảnh hưởng bởi tỷ lệ phần trăm các lớp trong tập dữ liệu. Ví dụ, nếu tập dữ liệu có 90% là mẫu thuộc lớp A và 10% là mẫu thuộc lớp B, mô hình chỉ cần dự đoán tất cả các mẫu là A cũng đạt độ chính xác 90%, nhưng hiệu quả phân loại thực tế không tốt.

2.1.2.2 *Log Loss (Binary Cross-Entropy)*

Được dùng làm hàm mất mát trong quá trình huấn luyện mô hình để đo lường độ sai lệch giữa xác suất dự đoán và nhãn thực tế. Giá trị log loss càng

nhỏ chứng tỏ mô hình càng chính xác trong việc dự đoán xác suất đúng với nhãn thật. Log Loss được tính bằng công thức như sau:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right]$$

Trong đó:

- Yi ∈{0,1}: Nhãn thực tế
- $y^i \in (0,1)$: Xác suất dự đoán
- N: Tổng số mẫu

2.1.2.3 *AUC-ROC*

AUC - ROC là một phương pháp đánh giá hiệu suất của mô hình phân loại, đặc biệt trong việc phân biệt giữa hai lớp.

Giá trị AUC:

- 1: Dự đoán hoàn hảo
- 0.5: Dự đoán ngẫu nhiên (kém)
- < 0.5: Dự đoán tệ hơn ngẫu nhiên

Đường cong ROC (Receiver Operating Characteristic) là biểu đồ thể hiện mối quan hệ giữa TPR (True Positive Rate - tỷ lệ dương tính đúng) và FPR (False Positive Rate - tỷ lệ dương tính sai) khi thay đổi ngưỡng phân loại.

Một điểm đáng chú ý là AUC không bị ảnh hưởng bởi sự mất cân bằng dữ liệu, điều này giúp nó trở thành một chỉ số tin cậy trong nhiều trường hợp khi các lớp không đồng đều về số lượng. Việc sử dụng AUC - ROC hỗ trợ các nhà nghiên cứu và kỹ sư dữ liệu đánh giá chính xác khả năng phân loại của mô hình, từ đó đưa ra các cải thiện phù hợp.

2.1.2.4 *Precision (Độ chính xác của lớp dương)*

Độ chính xác là tỷ lệ phần trăm các mẫu được dự đoán là thuộc lớp nào đó thực sự thuộc lớp đó. Precision được tính bằng công thức như sau:

$$Precision = TP / (TP + FP)$$

Trong đó:

- TP (True Positive): Số lượng mẫu thuộc lớp A được dự đoán đúng là A.
- FP (False Positive): Số lượng mẫu thuộc lớp B được dự đoán sai là A.

Độ chính xác cho biết tỷ lệ tin cậy của các dự đoán dương tính (dự đoán là A) của mô hình. Giá trị Precision cao nghĩa là mô hình ít dự đoán sai âm tính (FP). Tuy nhiên, Precision cao không đồng nghĩa với độ chính xác cao, vì nó phụ thuộc vào tỷ lệ FP.

- Ưu điểm:
  - Cho biết tỷ lệ tin cậy của các dự đoán dương tính.
  - Phù hợp cho các bài toán cần ít sai sót dương tính (ví dụ: chẩn đoán bệnh).
- Nhược điểm:
  - Phụ thuộc vào tỷ lệ FP.
  - Không cung cấp thông tin về hiệu suất dự đoán cho các mẫu âm tính.

2.1.2.5 *Recall (Độ bao phủ)*

Độ nhạy là tỷ lệ phần trăm các mẫu thực sự thuộc lớp nào đó được dự đoán đúng là thuộc lớp đó. Recall được tính bằng công thức như sau**:**

$$Recall = TP / (TP + FN)$$

Trong đó:

- TP (True Positive): Số lượng mẫu thuộc lớp A được dự đoán đúng là A.
- FP (False Positive): Số lượng mẫu thuộc lớp B được dự đoán sai là A.

Độ nhạy cho biết tỷ lệ bao quát của các dự đoán dương tính của mô hình.

Giá trị Recall cao nghĩa là mô hình ít dự đoán sai dương tính (FN). Recall cao thường đi kèm với Precision thấp và ngược lại, do sự đánh đổi giữa việc giảm thiểu FP và FN.

2.1.2.6 *F1-Score*

F1-Score Là trung bình điều hòa giữa Precision và Recall. F1-score được tính bằng công thức như sau**:**

$$F1 = 2 * (Precision * Recall) / (Precision + Recall).$$

F1-Score là thước đo cân bằng giữa Precision và Recall, giúp đánh giá tổng thể hiệu suất của mô hình cho một lớp. Giá trị F1 cao cho thấy mô hình có hiệu suất tốt cho cả việc giảm thiểu FP và FN.

- Ưu điểm: Đánh giá tổng thể hiệu suất phân loại cho một lớp và cân bằng giữa Precision và Recall.
- Nhược điểm: Không cung cấp thông tin chi tiết về hiệu suất cho từng loại sai sót.

2.2 Mô hình Factorization Machine (FM)

2.2.1 Giời thiệu về mô hình Factorization Machine

Factorization Machine – FM được đề xuất bởi Rendle (2010), là một thuật toán học có giám sát, có thể sử dụng cho các nhiệm vụ phân loại, hồi quy và xếp hạng. Đây là một phương pháp phổ biến và có ảnh hưởng lớn trong việc dự đoán và đưa ra gợi ý, đặc biệt phù hợp cho mục đích đề xuất sản phẩm sách của đề tài. Điểm mạnh của FM so với hồi quy tuyến tính và phân tích ma trận là:

- Khả năng mô hình hóa tương tác giữa các biến ở bậc cao, thường được thiết lập ở bậc hai
- Thuật toán tối ưu nhanh của FM giúp giảm thời gian tính toán đa thức xuống độ phức tạp tuyến tính, rất hiệu quả đối với dữ liệu đầu vào thưa và có chiều cao.
- FM khắc phục dữ liệu thư (sparse): Bằng cách học các vector tiềm ẩn riêng biệt cho từng đặc trưng

Nhờ những ưu điểm này, FM được áp dụng rộng rãi trong các hệ thống đề xuất hiện đại.

2.2.2 Cách hoạt động.

Xét một cách hình thức, giả sử X là vector đặc trưng đại diện cho một mẫu dữ liệu, và y là nhãn tương ứng, có thể là một giá trị thực hoặc là nhãn phân loại. Trong ngữ cảnh hệ thống đề xuất sách, X bao gồm các đặc trưng như người dùng, sách, tác giả, và các thông tin bổ sung (như thể loại, số lượng đánh

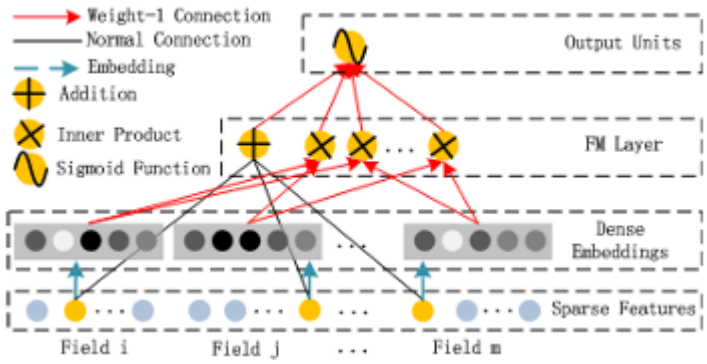
giá, đánh giá trung bình), còn y là điểm đánh giá do người dùng gán cho cuốn sách.

Mô hình Factorization Machine (FM) bậc hai được định nghĩa như sau:

$$\hat{y}(x) = \mathbf{w}_0 + \sum_{i=1}^{d} \mathbf{w}_i x_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

Trong đó:

- Wo là hệ số thiên lệch toàn cục (global bias);
- Wi là trọng số của biến đặc trưng thứ iii;
- $v_i \in R^k$ là vector nhúng (embedding) của đặc trưng i;
- ⟨vi,vj⟩ tích vô hướng giữa hai vector nhúng;
- k là số chiều của không gian tiềm ẩn.



Hình 2.1 Cấu trúc model Factorization Machine

Trong hệ thống đề xuất sách, điều này cho phép mô hình tự động học được mối quan hệ giữa người dùng và sách, hoặc giữa tác giả và thể loại, mà không cần phải thiết kế thủ công các đặc trưng tương tác. Trong trường hợp đặc trưng xi biểu diễn sách và xj biểu diễn người dùng, thì thuật ngữ tương tác chính là tích vô hướng giữa vector nhúng người dùng và vector nhúng sách - tương tự như trong mô hình phân rã ma trận (matrix factorization).

Dễ dàng nhận thấy rằng hai thành phần đầu tiên trong biểu thức tương ứng với mô hình hồi quy tuyến tính, trong khi thành phần thứ ba mở rộng mô

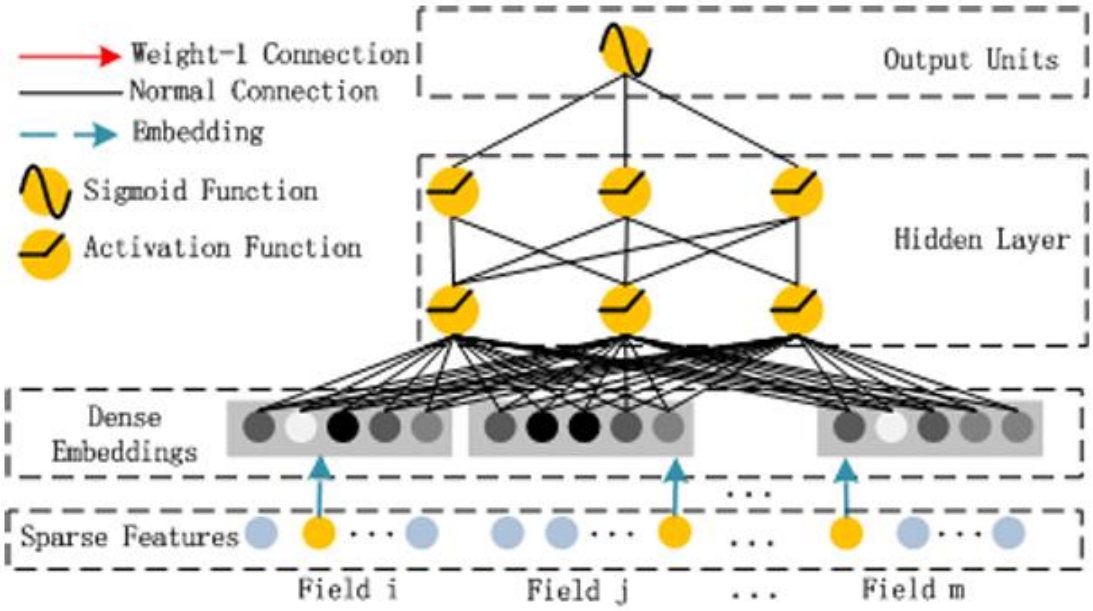
hình phân rã ma trận bằng cách mô hình hóa tự động các tương tác phi tuyến giữa mọi cặp đặc trưng.

2.3 Mô hình mạng neuron sâu(Deep Neuron Network – DNN)

2.3.1 Giới thiệu về mô hình học sâu.

Mạng neuron học sâu (Deep Neuron Network – DNN) là một phương pháp tiên tiến trong trí tuệ nhân tạo, được thiết kế dựa trên mạng nơ-ron nhân tạo truyền thống nhưng mở rộng với nhiều lớp ẩn, cho phép xử lý các mối quan hệ phức tạp và phi tuyến trong dữ liệu.

Nhờ khả năng tự động trích xuất đặc trưng từ dữ liệu đầu vào, DNN đã trở thành công cụ mạnh mẽ trong các lĩnh vực như thị giác máy tính, xử lý ngôn ngữ tự nhiên và hệ thống gợi ý. Đặc biệt, trong hệ thống gợi ý, DNN giúp học biểu diễn phi tuyến giữa người dùng và sản phẩm, đồng thời tích hợp các yếu tố ngữ cảnh như nội dung, địa điểm và hành vi. Điều này không chỉ tăng cường độ chính xác của dự đoán mà còn giảm thiểu sự phụ thuộc vào các kỹ thuật xử lý thủ công và mang lại hiệu quả cao.



Hình 2.2 Hình minh họa Neuron Network

Mô hình DNN gồm có 3 thành phần chính:

- Lớp đầu vào (Input Layer): Tiếp nhận dữ liệu đầu vào (vector đặc trưng của người dùng và sản phẩm (sau khi mã hóa hoặc embedding)
- Lớp ẩn (Hidden Layers): Diễn ra các phép biến đổi phi tuyến tính của dữ liệu. Mỗi lớp bao gồm nhều neuron hay node, mỗi node tính toán giá trị đầu ra thông qua hàm kích hoạt (Sigmoid).
- Lớp đầu ra (Output Layer): Sẽ đưa ra kết quả cuối cùng

2.3.2 Cách hoạt động của mô hình.

Quá trình hoạt động của 1 mô hình mạng học sâu (DNN) gồm 3 bước chính như sau:

Bước 1: Thực hiện lan truyền tiến (Forward Propagoation). Tại mỗi node, đầu vào được nhân với trọng số, cộng với bias và áp dụng hàm kích hoạt như sau

$$z = \mathbf{w}^T \mathbf{x} + b, \quad a = \sigma(z)$$

Trong đó:

- x là vector đầu vào,
- w là vector trọng số,
- b là hệ số bias,
- σ là hàm kích hoạt (activation function),
- a là đầu ra của nơ-ron.

Bước 2: Tính toán hàm mất mát (Loss Function). Một hàm mất mát như (MSE) hoặc Binary Cross-Entropy được dùng để đo lường sai lệnh giữa dự đoán và thực tế.

Bước 3 Lan truyền người và tối ưu hóa: sử dụng thuật toán lan truyền ngược (backpropagation) để tính đạo hàm của hàm mất mát theo các tham số trong mạng, các tham số được cập nhật thông qua thuật toán tối ưu như Gradient Descent hoặc Adam để giảm hàm mất mát.

2.4 Mô hình DeepFM

2.4.1 Giới thiệu về mô hình.

DeepFM (Deep Factorization Machine) là một mô hình lai (hybrid model) kết hợp giữa Factorization Machines (FM) và Deep Neural Networks (DNN)**,** được thiết kế để học các tương tác đặc trưng (feature interactions) ở cả bậc thấp và bậc cao.

Mô hình này được được sử dụng trong đề tài nhằm giải quyết các bài toán dự đoán đánh giá của người tiêu dùng với các sản phẩm, với khả năng tự động trích xuất đặc trưng phức tạp mà không cần kỹ thuật chọn đặc trưng thủ công.

2.4.2 Kiến trúc mô hình.

Giả sử tập dữ liệu huấn luyện bao gồm nnn mẫu dữ liệu có dạng (x,y), trong đó X là một bản ghi dữ liệu gồm mmm trường thông tin, thường biểu diễn mối quan hệ giữa người dùng và sách, còn y∈R là giá trị đánh giá (rating) mà người dùng gán cho cuốn sách tương ứng. Các trường trong bản ghi X có thể bao gồm các thuộc tính định danh như mã người dùng, mã sách, tên tác giả và các trường số như số lượt đánh giá trung bình của sách, hoặc số lượt đánh giá mà người dùng đã thực hiện.

Các trường định danh (ví dụ: người dùng, sách, tác giả) được mã hóa dưới dạng vector one-hot, trong khi các trường số có thể được giữ nguyên hoặc được rời rạc hóa và mã hóa tương tự. Sau quá trình tiền xử lý, mỗi mẫu dữ liệu được biểu diễn dưới dạng cặp (x,y), trong đó:

$$x = [x_{field1}, x_{field2},,\ldots, x_{fieldm},]$$

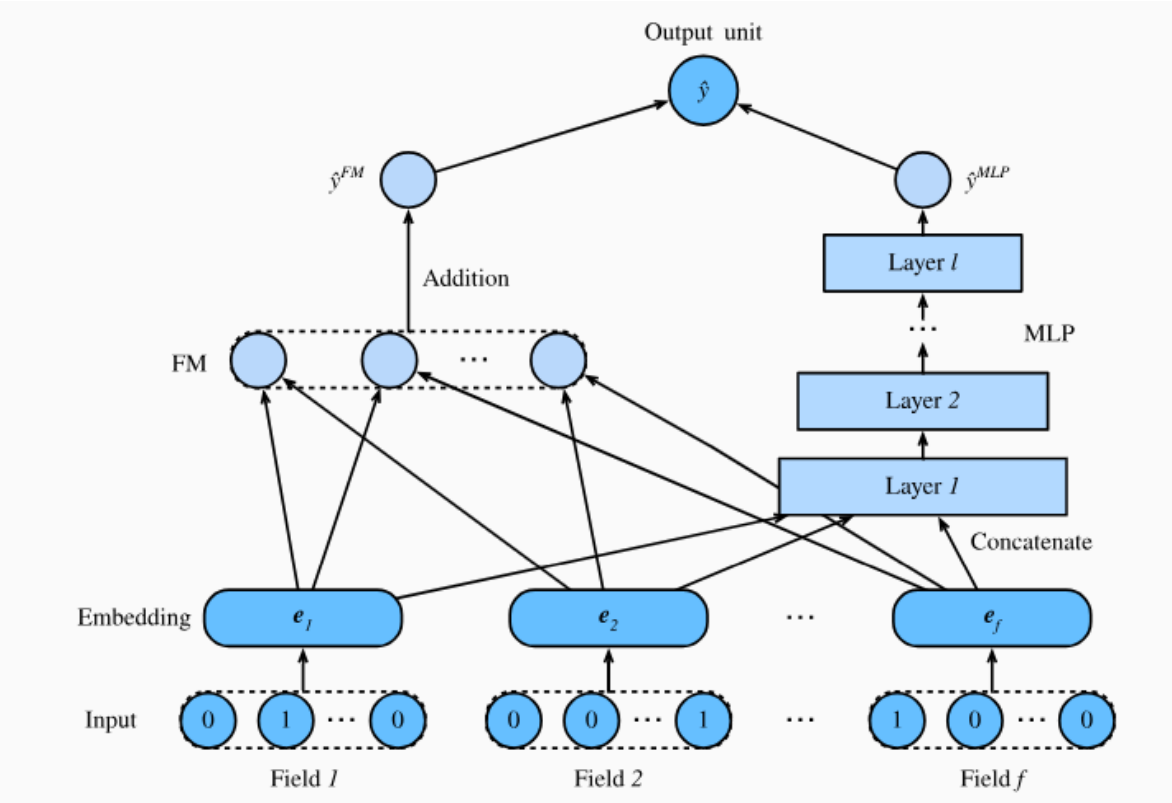
là một vector m chiều, với mỗi x_{fieldj} là biểu diễn vector của trường thứ j trong bản ghi dữ liệu.

Thông thường, vector đặc trưng x có số chiều lớn và rất thưa (sparse), đặc biệt trong trường hợp có số lượng người dùng và sách lớn. Nhiệm vụ của mô hình dự đoán đánh giá là xây dựng một hàm hồi quy:

$$Y = RatingModel(x)$$

nhằm ước lượng giá trị đánh giá mà người dùng có thể gán cho một cuốn sách nhất định trong một ngữ cảnh cụ thể, từ đó hỗ trợ hệ thống đề xuất cá nhân hóa hiệu quả hơn.

Trong đề tài này, dữ liệu đầu vào bao gồm các trường như người dùng (user), cuốn sách (book), tác giả (author), và có thể bao gồm các đặc trưng liên tục hoặc phân loại bổ sung khác.



Hình 2.3 Hình minh họa cấu trúc của DeepFM

Như minh họa trong mô hình, DeepFM bao gồm hai thành phần chính:

- Thành phần FM: dùng để học tương tác đặc trưng bậc hai (ví dụ: tương tác giữa người dùng và sách, hoặc giữa sách và tác giả).
- Thành phần Deep (mạng nơ-ron sâu): dùng để học tương tác đặc trưng bậc cao hơn, ví dụ như các mối quan hệ phức tạp giữa người dùng, thể loại sách, và phong cách của tác giả.

Cả hai thành phần đều dùng chung đầu vào X, bao gồm các biểu diễn vector nhúng (embedding vectors) của từng trường.

Đối với mỗi đặc trưng i, mô hình sử dụng:

- Một trọng số vô hướng w_i để biểu diễn tầm quan trọng bậc một của đặc trưng đó (ví dụ, mức ảnh hưởng trực tiếp của người dùng đến rating),
- Một vector tiềm ẩn V_i để biểu diễn ảnh hưởng tương tác của đặc trưng i với các đặc trưng khác (ví dụ: tương tác giữa người dùng và tác giả, hoặc giữa sách và người dùng).

Vector V_i được sử dụng đồng thời trong:

- Thành phần FM: để mô hình hóa tương tác đặc trưng bậc hai;
- Thành phần Deep: để học các tương tác đặc trưng bậc cao hơn.

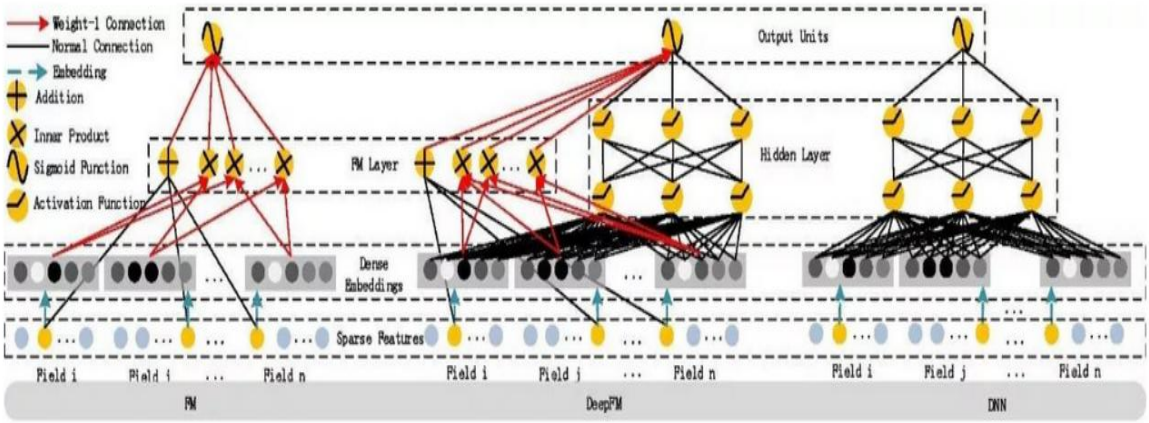
Tất cả các tham số, bao gồm w_i, V_i, và các tham số mạng như trọng số lớp $W^{(l)}$, bias $b^{(l)}$ đều được huấn luyện đồng thời trong quá trình tối ưu mô hình, nhằm dự đoán giá trị rating đầu ra như sau:

$$\hat{y} = \sigma(y_{FM} + y_{DNN})$$

Trong đó:

- Y-hat ∈ [0,1] là giá trị rating được chuẩn hóa hoặc xác suất suy ra từ rating,
- y_{FM} là đầu ra của thành phần Factorization Machine,
- y_{DNN} là đầu ra của thành phần mạng nơ-ron sâu (DNN).

2.4.3 Luồng dữ liệu trong mô hình DeepFM.



Hình 2.4 Hình minh họa luồng hoạt động của DeepFM

Mô hình DeepFM xử lý dữ liệu đầu vào thông qua một luồng dữ liệu cụ thể nhằm khai thác các tương tác đặc trưng một cách hiệu quả. Luồng dữ liệu này được chia thành các bước chính như sau:

Bước 1: Tiền xử lý dữ liệu đầu vào (Input Features). Dữ liệu đầu vào gồm các đặc trưng sparse (user_id, book_id, isbn, authors, title, location) . Các đặc trưng rời rạc được ánh xạ sang dạng one-hot encoding, sau đó sẽ được nhúng.

Bước 2: Embedding Layer (Lớp nhúng). Tất cả đặc trưng rời rạc được đưa vào lớp embedding, nơi mỗi giá trị được ánh xạ thành một vector nhúng có kích thước cố định. Các vector nhúng này được sử dụng chung cho cả hai thành phần: FM và DeepFM

Bước 3: Nhánh FM – Học tương tác bậc hai. Các vector nhúng từ embedding layer được sử dụng để tính toán các tương tác đặc trưng bậc hai bằng công thức của Factorization Machines. Kết quả đầu ra là một giá trị đơn thể hiện sự kết hợp tuyến tính và phi tuyến bậc hai của các đặc trưng.

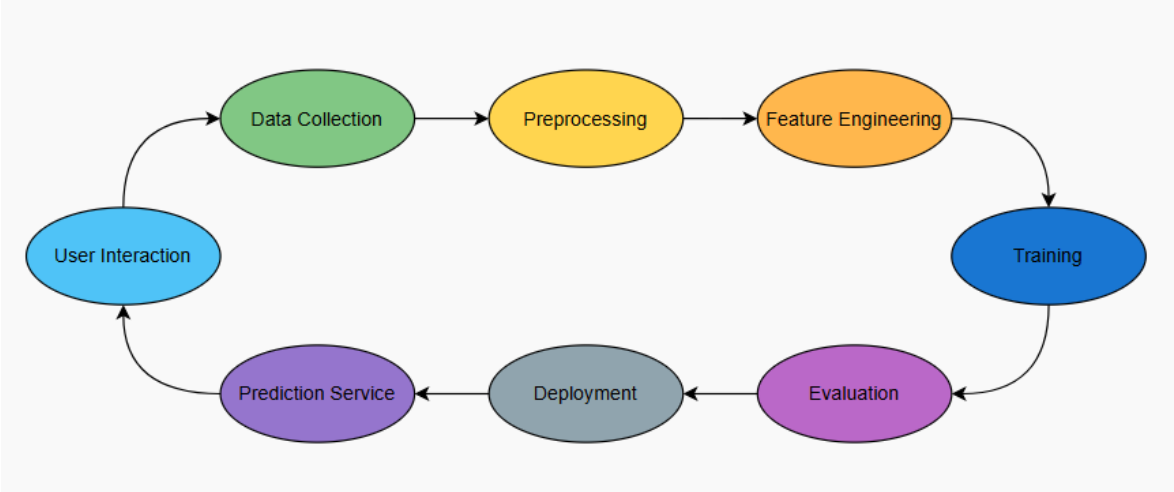
Bước 4: Nhánh Deep – Học tương tác bậc cao. Các vector nhúng được nối lại thành một vector duy nhất, sau đó được đưa qua các lớp ẩn của mạng neuron truyền thẳng (DNN). Mỗi lớp thực hiện lan truyền tiến thông qua hàm kích hoạt phi tuyến sigmoid.

Bước 5: Kết hợp kết quả hai nhánh (Fusion). Đầu ra của hai nhánh (FM + Deep) được cộng lại (hoặc ghép nối và đưa qua lớp fully connected cuối cùng). Kết quả sau cùng được đưa vào lớp đầu ra (output layer), sử dụng hàm Sigmoid

2.5 Ứng dụng của MLOPs trong hệ thống đề xuất sách.

MLOPs (Machine Learning Operations) là một tập hợp các nguyên tắc và thực hành nhằm tự động hóa và chuẩn hóa quy trình triển khai và vận hành các mô hình học máy trong môi trường sản xuất. MLOps kết hợp giữa Khoa học dữ liệu (Data Science) và Kỹ thuật phần mềm (DevOps) để đảm bảo rằng mô hình ML có thể hoạt động ổn định, dễ quản lý, và có thể mở rộng.

2.5.1 Vòng đời của một hệ thống MLOPs.



Hình 2.5 Quy trình MLOps trong hệ thống đề xuất sách

2.5.1.1 *Thu thập và xử lý dữ liệu (Data Collection & Processing)*

Để xây dựng một mô hình học máy hiệu quả, việc thu thập và xử lý dữ liệu là bước không thể thiếu và đòi hỏi sự chú ý đặc biệt. Trước tiên, dữ liệu về sách, đánh giá của người dùng được thu thập từ Kaggle với định dạng csv. Sau khi thu thập, dữ liệu cần được làm sạch để đảm bảo chất lượng, bao gồm xử lý các giá trị thiếu, loại bỏ giá trị dữ liệu nhiễu và định dạng lại giá trị của các trường dữ liệu. Sau đó, thực hiện lưu dữ liệu vừa được làm sạch vào 1 file csv. Tiếp theo, quá trình trích xuất đặc trưng (feature engineering) là bước quan trọng giúp chuyển đổi dữ liệu từ dạng rời rạc sang vector thông qua phương pháp label encoding. Cuối cùng, dữ liệu cần được phân chia thành các tập huấn luyện, kiểm tra và xác thực (train/test) theo tỉ lện 80:20 để đảm bảo mô hình được đánh giá một cách toàn diện và khách quan. Tất cả các bước này đều cần được thực hiện một cách cẩn thận và có hệ thống để đảm bảo rằng mô hình có thể học hỏi và dự đoán chính xác từ dữ liệu đã được chuẩn bị kỹ lưỡng.

2.5.1.2 *Huấn luyện mô hình (Model Training)*

Quá trình này bao gồm một số bước quan trọng. Đầu tiên, thiết kế model phục vụ DeepFM (Deep Factorization Machine) và thiết kế kiến trúc mô hình bao gồm xác định số lượng lớp, loại hàm kích hoạt. Tiếp theo, quá trình huấn luyện mô hình được thực hiện bằng cách đưa dữ liệu vào, tính toán hàm mất

mát và cập nhật trọng số thông qua các thuật toán tối ưu Adam. Tất cả các bước này cần được thực hiện một cách cẩn thận và có hệ thống để đảm bảo mô hình đạt được hiệu quả tốt nhất.

2.5.1.3 *Đánh giá và kiểm thử mô hình (Evaluation & Testing)*

Đối với các hệ thống đề xuất, các chỉ số như Recall, Loss và Accuracy được áp dụng để đo lường mức độ phù hợp và chất lượng gợi ý. Việc lựa chọn đúng tiêu chí đánh giá không chỉ giúp tối ưu hóa hiệu suất mà còn đảm bảo mô hình đáp ứng tốt yêu cầu thực tế.

2.5.1.4 *Triển khai mô hình (Model Deployment)*

Triển khai dưới dạng API, nơi mô hình FastAPI, giúp dễ dàng tích hợp với các hệ thống khác. Ngoài ra, đối với các tác vụ xử lý dữ liệu lớn định kỳ, dạng batch là lựa chọn tối ưu, ví dụ như việc chạy các tác vụ hàng đêm để tạo gợi ý sản phẩm hoặc phân tích dữ liệu. Để hỗ trợ quá trình triển khai, việc đóng gói mô hình bằng Docker giúp đảm bảo tính nhất quán khi chạy trên các môi trường khác nhau. Đồng thời, quản lý phiên bản mô hình thông qua model registry giúp theo dõi và kiểm soát các thay đổi.

2.5.1.5 *Tái huấn luyện (Retraining)*

Tái huấn luyện và cập nhật mô hình là một quy trình quan trọng nhằm đảm bảo mô hình luôn phù hợp với dữ liệu mới và đáp ứng tốt các yêu cầu thực tế. Quy trình này có thể được thực hiện theo lịch định kỳ (ví dụ: hàng tuần, dựa trên cấu hình trong Mlflow_tracking.yaml). Kết quả của mô hình mới sẽ được đánh giá kỹ lưỡng và nếu hiệu suất được cải thiện so với phiên bản hiện tại, mô hình mới sẽ được triển khai vào môi trường sản xuất. Quy trình này không chỉ giúp duy trì hiệu quả của hệ thống mà còn đảm bảo khả năng thích nghi với những thay đổi trong môi trường dữ liệu.

2.5.2 Lợi ích của MLOPs.

MLOps mang lại nhiều lợi ích quan trọng, không chỉ giúp quá trình phát triển mô hình học máy trở nên hiệu quả hơn, mà còn đảm bảo rằng mô hình có

thể được vận hành ổn định trong môi trường thực tế. Dưới đây là các lợi ích chính như sau:

- Tự động hóa quy trình phát triển và triển khai mô hình giúp giảm thiểu thao tác thủ công trong các giai đoạn như xử lý dữ liệu, huấn luyện, đánh giá và triển khai. Điều này không chỉ tối ưu hóa thời gian mà còn giảm thiểu rủi ro sai sót do con người gây ra. Nhờ tích hợp các quy trình CI/CD tự động, tốc độ đưa mô hình vào môi trường sản xuất được cải thiện đáng kể, đảm bảo tính liên tục và hiệu quả trong vận hành. Bên cạnh đó, việc thiết lập lịch tái huấn luyện định kỳ giúp mô hình luôn được cập nhật với dữ liệu mới, duy trì hiệu suất và độ chính xác cao trong các ứng dụng thực tế.
- Đảm bảo mọi bước trong quá trình phát triển và triển khai mô hình đều được ghi nhận và có thể lặp lại. Từ việc thu thập và xử lý dữ liệu, mã nguồn sử dụng, các tham số huấn luyện cho đến kết quả cuối cùng của mô hình, tất cả đều được lưu trữ một cách có hệ thống.
- Quản lý hiệu quả mô hình và dữ liệu đóng vai trò quan trọng trong việc đảm bảo tính nhất quán và tối ưu hóa quy trình làm việc. Việc theo dõi phiên bản của mô hình, dữ liệu và các đặc trưng không chỉ giúp ghi nhận lịch sử thay đổi mà còn cung cấp khả năng quay lại trạng thái trước đó khi cần thiết, từ đó giảm thiểu rủi ro và sai sót. Đồng thời, việc lưu trữ và tái sử dụng các mô hình đã được chứng minh hiệu quả cùng với các tập đặc trưng tối ưu giúp tiết kiệm đáng kể chi phí và thời gian phát triển.
- Hệ thống giám sát tự động giúp đảm bảo hiệu suất hoạt động của các mô hình trong thực tế. Bằng cách liên tục theo dõi và phân tích dữ liệu, hệ thống có khả năng phát hiện sớm các dấu hiệu bất thường hoặc suy giảm hiệu quả. Điều này không chỉ giúp tránh được các rủi ro tiềm ẩn mà còn tạo điều kiện để điều chỉnh và tối ưu hóa mô hình kịp thời.
- Mở rộng hiệu quả các mô hình học máy để xử lý các tập dữ liệu lớn hơn và đáp ứng khối lượng công việc tăng cao. Bằng cách tối ưu hoá quy

trình và sử dụng tài nguyên một cách hợp lý, các mô hình học máy có thể được triển khai trên quy mô lớn mà vẫn đảm bảo hiệu suất và độ chính xác.

CHƯƠNG 3 XÂY DỰNG ỨNG DỤNG.

3.1 Thiết kế và triển khai module

3.1.1 Xử lý dữ liệu.

3.1.1.1 *Nguồn dữ liệu*

Dữ liệu được sử dụng trong bài này gồm bảng Books - thông tin về sách gồm tiêu đề, mã sách, tác giả, Ratings - đánh giá của người dùng gồm tiêu đề, mã người dùng, đánh giá, và Users – thông tin người dùng bao gồm ID người dùng, địa chỉ . Bảng Users chứa 278858 bản ghi, Books gồm 271360 đầu sách và Ratings là 1148780 bản ghi.

3.1.1.2 *Làm sạch dữ liệu.*

Để xử lý dữ liệu thô và chuẩn bị cho việc phân tích, chúng ta thực hiện các bước sau: Đầu tiên, đọc dữ liệu từ các file Books.csv, Users.csv và Ratings.csv với các cột tương ứng là ISBN, Book-Title, Book-Author, User-ID, Location, và User-ID, ISBN, Book-Rating, Age, Year.

```
import numpy as np
import pandas as pd

# Read data
df_book = pd.read_csv('Books.csv', usecols=['ISBN','Book-Title', 'Book-Author'])
df_rating = pd.read_csv('Ratings.csv', usecols=['User-ID', 'ISBN', 'Book-Rating'])
df_user = pd.read_csv('Users.csv', usecols=['User-ID', 'Location'])
```

Hình 3.1 Code đọc dữ liệu

Tiếp theo, chuẩn hóa tên cột để dễ sử dụng hơn: đổi Book-Title thành Title, Book-Author thành Author, Book-Rating thành Rating và User-ID thành User. Sau đó, lọc bỏ các dữ liệu không hợp lệ như các đánh giá có Rating bằng 0 (vì một số dữ liệu thường quy định ngầm sách có đánh giá bằng 0 là khách hàng không thực hiện đánh giá) và người dùng có Location là 'n/a, n/a, n/a'.

```
df_user = df_user[df_user['Location'] != 'n/a, n/a, n/a']
df_rating = df_rating[df_rating['Book-Rating'] > 0]
df_rating['Book-Rating'] = df_rating['Book-Rating'].apply(lambda x: 1 if x > 5 else 0)
```

Hình 3.2 Loại bỏ các rating nhỏ hơn 0

Đồng thời, tiến hành xử lý và chuẩn hóa thông tin Location của người dùng. Đối với dữ liệu thiếu, loại bỏ các dòng chứa giá trị NA trong tất cả các bảng dữ liệu.

```
def process_location(location):
  # Ensure the input is a string before processing
  location_str = str(location)
  if location_str == 'n/a, n/a, n/a':
    return None
  parts = [part.strip() for part in location_str.split(',')]
  valid_parts = [part for part in parts if part != 'n/a']
  if not valid_parts:
    return None
  elif len(valid_parts) == 1:
    return valid_parts[0]
  else:
    return valid_parts[-1]

# Convert the 'Location' column to string type before applying the function
df_user['Location'] = df_user['Location'].astype(str)

df_user['Location'] = df_user['Location'].apply(process_location)
```

Hình 3.3 Đưa dữ liệu bảng địa chỉ về đúng định dạng

Tiếp theo, xử lý dữ liệu trùng lặp bằng cách loại bỏ các đánh giá trùng lặp giữa User và Title để đảm bảo mỗi người dùng chỉ đánh giá một cuốn sách một lần.

3.1.1.3 *Embedding các trường dữ liệu cho huấn luyện*

```
def transform(self, data: pd.DataFrame, features: list[str]) -> pd.DataFrame:
    data = data.copy()
    for feat in features:
        if feat in self.encoders:
            data[feat] = self.encoders[feat].transform(data[feat].astype(str))
    return data
```

Hình 3.4 Sử dụng labelEncoding cho dữ liệu rời rạc

Cuối cùng, tiến hành mã hóa dữ liệu bằng cách sử dụng LabelEncoder cho các cột dạng categorical như User, ISBN, Author và Location để chuẩn bị cho các bước phân tích tiếp theo. Các bước này đảm bảo dữ liệu được làm sạch và tối ưu hóa cho quá trình xử lý và phân tích.

3.1.1.4 *Chuẩn bị dữ liệu cho mô hình*

Để xây dựng mô hình DeepFM, trước tiên cần tạo nhãn (label) dựa trên ngưỡng rating, với giá trị mặc định là 5. Các giá trị rating lớn hơn hoặc bằng 5 sẽ được gán nhãn là 1 (positive), trong khi các giá trị nhỏ hơn 5 sẽ được gán nhãn là 0 (negative).

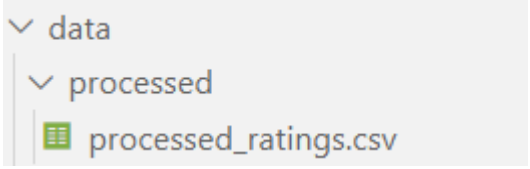
Sau khi xử lý nhãn, dữ liệu cần được chia thành hai tập riêng biệt: tập train và tập test, với tỷ lệ phân chia thường sử dụng là 80:20 nhằm đảm bảo đủ dữ liệu để huấn luyện và kiểm tra mô hình. Tiếp theo, cần chuẩn bị các feature columns để làm đầu vào cho mô hình DeepFM. Quá trình này bao gồm việc xác định các đặc trưng dạng sparse (thưa) và dense (dày), sau đó mã hóa các đặc trưng này bằng các phương pháp phù hợp như lable encoding hoặc embedding. Điều này giúp mô hình có khả năng xử lý cả thông tin tuyến tính và phi tuyến tính từ dữ liệu, đảm bảo hiệu quả trong việc dự đoán và phân loại.

```
sparse_features = ['User', 'ISBN', 'Title', 'Author', 'Location']
data = encode_data(data, sparse_features)

data['label'] = (data['Rating'] >= rating_threshold).astype(int)

sparse_feature_columns = create_feature_columns(data, sparse_features, embed_dim)
dense_feature_columns = []  # nếu có, bạn có thể thêm

feature_columns = [dense_feature_columns, sparse_feature_columns]

X_train, X_test, y_train, y_test = split_dataset(data, sparse_features, label_column='label'
                                                 test_size=test_size)

return feature_columns, (X_train, y_train), (X_test, y_test)
```

Hình 3.5 Chuẩn bị dữ liệu train và test

3.1.1.5 *Lưu dữ liệu đã được xử lý.*

Sau khi làm sạch dữ liệu thô, dữ liệu đã qua xử lý được lưu trữ vào thư mục "processed" với định dạng CSV một cách có tổ chức. Việc này không chỉ giúp dễ dàng truy xuất mà còn đảm bảo tính nhất quán trong quy trình làm việc.



Hình 3.6 Thư mục chứa dữ liệu được làm sạch

3.1.2 Huấn luyện mô hình DeepFM

3.1.2.1 *Thiết kế cấu trúc model của DeepFM.*

Lớp Embedding có nhiệm vụ chuyển đổi các đặc trưng phân loại (categorical features) thành các vector liên tục (dense vectors) có thể học được. Điều này giúp giảm chiều dữ liệu và học được mối quan hệ ngữ nghĩa giữa các giá trị rời rạc. Trong mô hình đề xuất, các đặc trưng được embedding bao gồm: User ID**,** ISBN (mã sách)**,** Title (tên sách)**,** Author (tác giả) và Location (vị trí người dùng). Mỗi đặc trưng sau khi được ánh xạ sẽ có một vector embedding tương ứng, được cập nhật trong quá trình huấn luyện mô hình.

```
model = DeepFM(
    feature_columns=(dense_feature_columns, sparse_feature_columns),
    k=6,
    w_reg=1e-3,
    v_reg=1e-3,
    hidden_units=[32, 16],
    output_dim=1,
    activation='relu'
)

# Biên dịch mô hình
model.compile(
    loss='binary_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)
```

Hình 3.7 Cấu hình mô hình DeepFM

Lớp FM chịu trách nhiệm học các tương tác bậc hai giữa các đặc trưng trong dữ liệu. Lớp này bao gồm hai phần: phần tuyến tính (linear part) giúp mô hình học trọng số cho từng đặc trưng riêng lẻ; và phần tương tác (interaction part) giúp mô hình học được mối quan hệ giữa từng cặp đặc trưng với nhau thông qua việc nhân tích vô hướng giữa các vector embedding. Để tránh hiện tượng quá khớp (overfitting), mô hình sử dụng hệ số regularization là w_reg = 1e-3 cho phần tuyến tính và v_reg = 1e-3 cho phần tương tác.

Lớp Deep là một mạng nơ-ron sâu (deep neural network) dùng để học các tương tác phi tuyến và phức tạp hơn giữa các đặc trưng. Cấu trúc của mạng gồm hai lớp ẩn với số lượng neuron lần lượt là 32 và 16, sử dụng hàm kích hoạt ReLU nhằm tăng khả năng biểu diễn phi tuyến của mô hình. Cuối cùng, lớp đầu ra (output layer) gồm một neuron duy nhất và sử dụng hàm kích hoạt sigmoid để đưa ra xác suất dự đoán đầu ra, đặc biệt phù hợp với các bài toán phân loại nhị phân (binary classification) như dự đoán đánh giá người dùng.

```
# Huấn Luyện
model.fit(
    X_train,
    y_train,
    batch_size=32,
    epochs=5,
    validation_split=0.2
)
```

Hình 3.8 Cấu hình quá trình huấn luyện của mô hình DeepFM

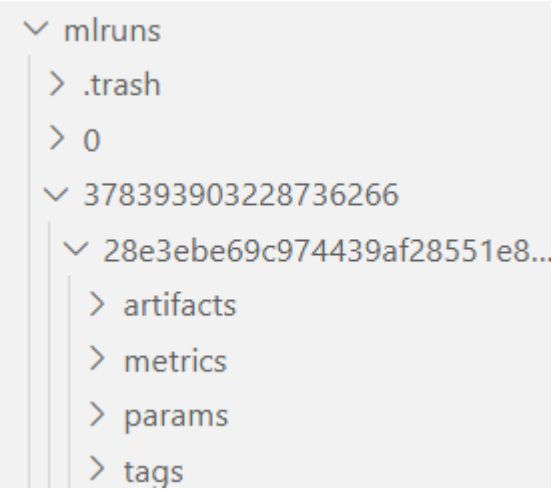
Trong quá trình huấn luyện mô hình, mỗi epoch được thiết kế để mô hình đi qua toàn bộ tập dữ liệu huấn luyện một lần, và tổng cộng có 5 epoch được thực hiện. Mỗi epoch được chia thành nhiều batch, mỗi batch chứa 32 mẫu dữ liệu. Trong từng batch, quy trình huấn luyện bao gồm bước forward pass để tính toán dự đoán từ mô hình, sau đó là bước tính toán loss và các metrics để đánh giá hiệu suất. Tiếp theo, backward pass được thực hiện nhằm cập nhật trọng số của mô hình dựa trên gradient, với sự hỗ trợ của thuật toán tối ưu Adam, giúp tự động điều chỉnh learning rate để tăng hiệu quả huấn luyện.

Sau khi hoàn thành mỗi epoch, mô hình sẽ được đánh giá trên tập validation để kiểm tra hiệu suất trên dữ liệu chưa được huấn luyện. Các chỉ số quan trọng như độ chính xác Accuracy được theo dõi nhằm đánh giá khả năng phân loại và hiệu quả tổng thể của mô hình trong suốt quá trình huấn luyện.

3.1.2.2 *Quản lý model và theo dõi phiên bản.*

Mỗi lần huấn luyện mô hình được ghi lại dưới dạng một experiment riêng biệt trong MLflow. Thông tin được ghi nhận bao gồm:

- Hyperparameters: ví dụ k=10, w_reg=1e-3, v_reg=1e-3, hidden_units=[32, 16]
- Metrics đánh giá: accuracy, AUC và loss trên tập train/validation
- Thông số huấn luyện: số epoch = 5, batch_size = 32
- Thời gian huấn luyện và tên mô hình được lưu theo timestamp



Hình 3.9 Các phiên bản model được lưu bằng MLflow

Quản lý phiên bản mô hình là một bước quan trọng trong quá trình phát triển và cải thiện hiệu suất của mô hình DeepFM. Sau mỗi lần huấn luyện, mô hình được lưu trữ dưới dạng một phiên bản riêng biệt trong thư mục mlruns.

3.1.2.3 *Triển khai mô hình.*

Sau khi huấn luyện thành công mô hình gợi ý sách (DeepFM) và lưu trữ phiên bản mới nhất dưới định dạng .pkl, hệ thống đã được triển khai thông qua một API đơn giản sử dụng FastAPI.

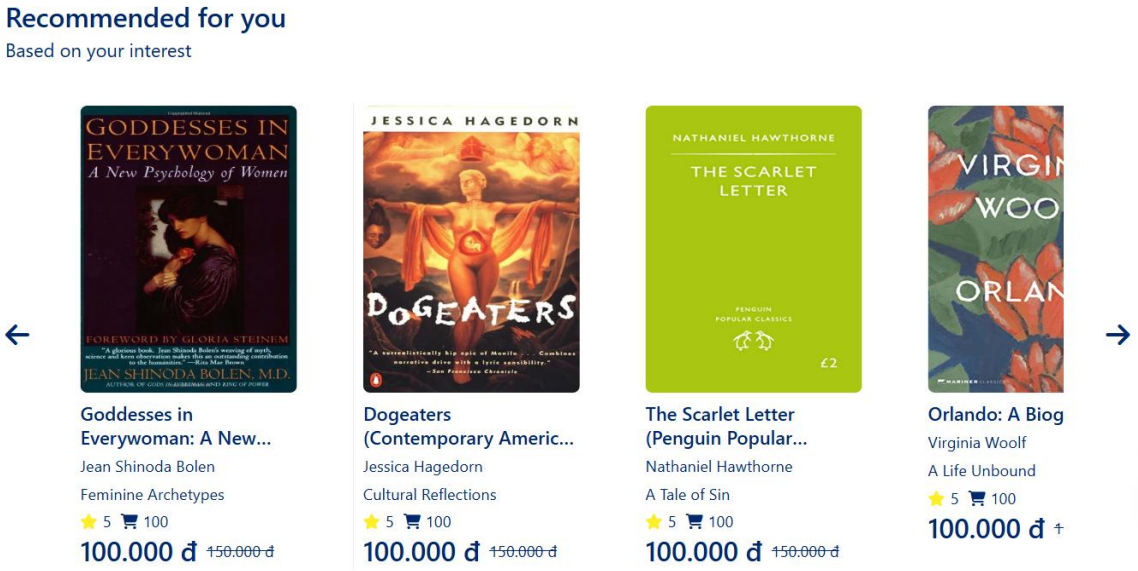
```
# Load model
MODEL_DIR = "models"
latest_model = max([os.path.join(MODEL_DIR, f) for f in os.listdir(MODEL_DIR) if f.endswith('.pkl')],
                   key=os.path.getctime)
model = joblib.load(latest_model)
```

Hình 3.10 Lấy model trong MLflow

Cụ thể, một ứng dụng FastAPI đã được xây dựng để phục vụ mô hình huấn luyện, cho phép nhận đầu vào là các đặc trưng của sách (đã qua xử lý) và trả về dự đoán từ mô hình. Khi khởi động, API sẽ tự động tìm và nạp mô hình mới nhất từ thư mục models. Dữ liệu đầu vào được định nghĩa rõ ràng thông qua Pydantic giúp đảm bảo kiểm tra kiểu dữ liệu và định dạng JSON chuẩn xác.

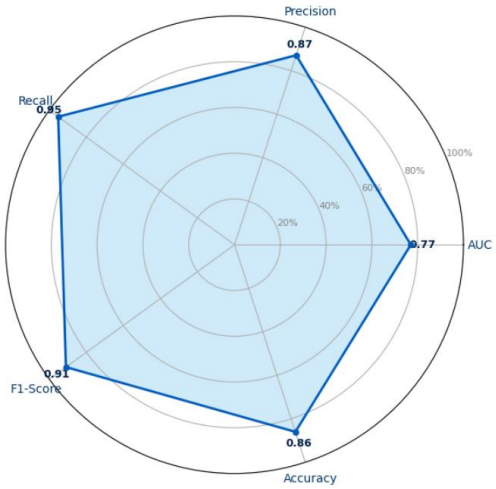
Endpoint /predict được sử dụng để nhận dữ liệu đầu vào dưới dạng danh sách các đặc trưng, sau đó được chuyển đổi thành mảng numpy để mô hình xử lý. Dự đoán kết quả được trả về dưới dạng JSON cho phép dễ dàng tích hợp vào các hệ thống frontend hoặc hệ thống đề xuất phía client.

Khi người dùng truy cập vào phía client, một yêu cầu (request) sẽ được gửi đến endpoint /predict kèm theo dữ liệu bao gồm ID người dùng. Hệ thống sẽ tiến hành phân tích dữ liệu và thực hiện dự đoán mức độ yêu thích của người dùng đối với tất cả các cuốn sách trong cơ sở dữ liệu. Dựa trên kết quả phân tích, hệ thống sẽ đề xuất danh sách 10 cuốn sách hàng đầu phù hợp nhất với sở thích của người dùng, từ đó nâng cao trải nghiệm cá nhân hóa và đáp ứng tốt hơn nhu cầu của họ.



Hình 3.11 Lấy model trong MLflow

3.2 Kết quả thực nghiệm và đánh giá



Hình 3.12 Chỉ số đánh giá model DeepFM cho hệ thống đề xuất sách

Mô hình DeepFM đạt hiệu suất tổng thể ở mức khá, đặc biệt nổi bật ở chỉ số Recall đạt 0.952, cho thấy khả năng bao phủ tương đối các đầu sách ưu thích trong hệ thống đề xuất. F1-score đạt 0.91, phản ánh sự cân bằng tốt giữa Precision và Recall, trong khi Precision đạt 0.871, thể hiện mô hình có khả năng dự đoán chính xác phần lớn các đề xuất phù hợp. Tuy nhiên, chỉ số AUC ở mức 0.770 cho thấy khả năng phân biệt giữa các đề xuất đúng và sai vẫn còn chưa được tốt. Tỷ lệ Accuracy đạt 0.860, chứng minh hiệu suất dự đoán tổng thể của mô hình là đáng tin cậy. Với hiệu suất Recall vượt trội, mô hình này đặc biệt phù hợp cho các ứng dụng yêu cầu không bỏ sót các mục tiêu quan trọng, tuy nhiên model cần tối ưu thêm Precision và AUC để giảm thiểu các đề xuất sai lệch và nâng cao hiệu quả phân loại.

KẾT LUẬN

Hệ thống đã tích hợp các kỹ thuật hiện đại như embedding, Factorization Machine và mạng nơ-ron sâu để khai thác các mối quan hệ tiềm ẩn giữa người dùng và sản phẩm, từ đó cải thiện độ chính xác trong gợi ý. Bên cạnh đó, việc đưa MLflow vào pipeline huấn luyện và triển khai giúp theo dõi hiệu suất, quản lý phiên bản mô hình và hỗ trợ triển khai thực tế một cách linh hoạt và có hệ thống.

Tuy nhiên, hệ thống vẫn còn tồn tại một số điểm hạn chế, đặc biệt là sự phụ thuộc lớn vào chất lượng dữ liệu đầu vào — bao gồm thông tin người dùng và dữ liệu mô tả sách. Trong các tình huống cold-start, như người dùng hoặc sách mới chưa có lịch sử tương tác, độ chính xác của hệ thống gợi ý chưa đạt tối ưu.

Trong tương lai, để khắc phục các điểm yếu trên và nâng cao hiệu suất hệ thống, cần tích hợp thêm dữ liệu nội dung như phân tích văn bản mô tả sách, đánh giá của người dùng, cũng như tri thức chuyên ngành. Định hướng phát triển tiếp theo là xây dựng mô hình đề xuất lai (hybrid recommendation), kết hợp các phương pháp collaborative filtering, content-based filtering và tri thức ngữ nghĩa. Điều này sẽ giúp hệ thống phục vụ hiệu quả hơn cho nhiều nhóm người dùng, đồng thời đóng góp vào việc nâng cao doanh thu, cải thiện trải nghiệm khách hàng và tăng lợi thế cạnh tranh cho Bookies trong hành trình chuyển đổi số toàn diện.

TÀI LIỆU THAM KHẢO

[1] Aggarwal, C. C. (2016). Recommender Systems: The Textbook. Springer.

[2] Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender Systems Handbook. Springer.

[3] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer

[4] Kim Falk. (2019). Practical Recommendation Systems. Manning.