BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BÁO CÁO GIẢI TRÌNH
SỬA CHỮA, HOÀN THIỆN ĐỀ ÁN TỐT NGHIỆP

Họ và tên học viên: Nguyễn Bảo Tùng

Chuyên ngành: HTTT

Khóa: 2022 đợt 2

Tên đề tài: Áp dụng kỹ thuật học sâu cho tư vấn trên sàn thương mại điện tử Postmart

Người hướng dẫn khoa học: : PGS. TS Trần Đình Quế

Ngày bảo vệ: 19/07/2025

Các nội dung học viên đã sửa chữa, bổ sung trong đề án tốt nghiệp theo ý kiến đóng góp của Hội đồng chấm đề án tốt nghiệp:

TT	Ý kiến hội đồng	Sửa chữa của học viên
1	Chỉnh sửa lại đúng format của tài liệu tham khảo	Tác giả đã chỉnh sửa lại đúng format của tài liệu tham khảo
2	Việt hóa lại các hình vẽ trong đề án, trích dẫn tài liệu tham khảo	Tiếp thu góp ý của Hội đồng, tác giả đã Việt hóa lại các hình vẽ biểu đồ tại mục 3.1 trong chương 3. Đã trích dẫn tài liệu tham khảo tại phần Danh mục tài liệu tham khảo
3	Tiến hành thực nghiệm tên dữ liệu Postmart	Tiếp thu góp ý của Hội đồng, tác giả đã tiến hành thử nghiệm và bổ sung thêm nội dung 3.2. Thử nghiệm trên bộ dữ liệu Đánh giá của Postmart và 3.3. So sánh và đánh giá sau thử nghiệm 2 tập dữ liệu trong chương 3

Hà Nội, ngày 29 tháng 7 năm 2025

Ký xác nhận của

CHỦ TỊCH HỘI ĐỒNG CHẤM ĐỀ ÁN	THƯ KÝ HỘI ĐỒNG	NGƯỜI HƯỚNG DẪN KHOA HỌC	HỌC VIÊN
PGS.TS. Trần Quang Anh	TS. Đào Thị Thúy Quỳnh	PGS.TS. Trần Đình Quế	Nguyễn Bảo Tùng

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

NGUYỄN BẢO TÙNG

ÁP DỤNG KỸ THUẬT HỌC SÂU CHO TƯ VẤN TRÊN SÀN THƯƠNG MẠI ĐIỆN TỬ POSTMART

ĐỀ ÁN THẠC SĨ KỸ THUẬT

HÀ NỘI - 2024

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

NGUYỄN BẢO TÙNG

ÁP DỤNG KỸ THUẬT HỌC SÂU CHO TƯ VẤN TRÊN SÀN THƯƠNG MẠI ĐIỆN TỬ POSTMART

Chuyên ngành: Hệ thống thông tin
Mã số: 8.48.01.04

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

NGƯỜI HƯỚNG DẪN KHOA HỌC : PGS. TS TRẦN ĐÌNH QUẾ

Trần Đình Quế

HÀ NỘI – 2024

MỤC LỤC

BẢN CAM ĐOAN

Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung đề án qua phần mềm DoIT một cách trung thực và đạt kết quả mức độ tương đồng 5% toàn bộ nội dung đề án. Bản đề án kiểm tra qua phần mềm là bản cứng đề án đã nộp bảo vệ trước hội đồng. Nếu sai tôi xin chịu các hình thức kỷ luật theo quy định hiện hành của Học viện.

Hà Nội, ngày 31 tháng 08 năm 2025

HỌC VIÊN CAO HỌC

Nguyễn Bảo Tùng

LỜI CẢM ƠN

Trong quá trình xây dựng đề cương, nghiên cứu và hoàn thành đề án thạc sĩ, tác giả đã nhận được rất nhiều sự trợ giúp đến từ các thầy, các cô trong Ban giám hiệu, Sau đại học, các giảng viên khoa Hệ thống thông tin trường Học viện Công Nghệ Bưu Chính Viễn Thông. Đặc biệt, cho phép tác giả được bày tỏ sự trân quý và biết ơn tới PGS. TS. Trần Đình Quế. Tác giả đã nhận được sự hướng dẫn tận tình, tâm huyết đến từ thầy. Qua đây, tác giả cũng xin gửi lời cảm ơn chân thành đến giảng viên hướng dẫn và các thầy cô trong khoa Sau đại học đã nhiệt tình giúp đỡ, hỗ trợ tác giả trong quá trình học tập cũng như trong quá trình nghiên cứu hoàn thành đề tài đề án thạc sĩ.

Cuối cùng, tôi xin được bày tỏ lòng biết ơn sâu sắc tới gia đình, bạn bè và đồng nghiệp tại Tổng Công Ty Bưu Điện Việt Nam đã luôn bên cạnh động viên, hỗ trợ và tạo điều kiện tốt nhất cho tôi trong quá trình học tập và hoàn thành đề án tốt nghiệp.

Trân trọng!

DANH SÁCH BẢNG

DANH SÁCH HÌNH VẼ

MỞ ĐẦU

1. Tính cấp thiết của đề tài:

Ngày nay, thương mại điện tử (TMĐT) đã và đang trở thành lĩnh vực có ảnh hưởng cực kỳ quan trọng đến tăng trưởng kinh tế của các quốc gia. Sự phát triển của TMĐT không chỉ giúp các hoạt động kinh doanh thuận lợi mà còn cung cấp nhiều giá trị mới và đáp ứng những nhu cầu mới của các doanh nghiệp và người tiêu dùng. Chính vì vậy, mọi quốc gia trên thế giới đều quan tâm đến việc phát triển TMĐT. Có thể kể đến một số vai trò của TMĐT bao gồm: Thay đổi tính chất của nền kinh tế mỗi quốc gia và nền kinh tế toàn cầu, làm cho tính tri thức trong nền kinh tế ngày càng tăng lên, mở ra cơ hội phát huy ưu thế của các nước phát triển sau, để họ có thể đuổi kịp, thậm chí vượt các nước đi trước, rút ngắn khoảng cách về trình độ tri thức giữa các nước phát triển với các nước đang phát triển.

Trong những năm gần đây, thị trường TMĐT Việt Nam ngày càng được mở rộng và hiện đã trở thành phương thức kinh doanh phổ biến được doanh nghiệp, người dân biết đến. Sự đa dạng về mô hình hoạt động, về đối tượng tham gia, về quy trình hoạt động và chuỗi cung ứng hàng hóa, dịch vụ với sự hỗ trợ của hạ tầng Internet và ứng dụng công nghệ hiện đại đã đưa TMĐT trở thành trụ cột quan trọng trong tiến trình phát triển kinh tế số của quốc gia.

Mặc dù gặp những ảnh hưởng tiêu cực trong năm 2020 do đại dịch Covid-19, TMĐT Việt Nam vẫn có những bước tăng tốc mạnh mẽ, trở thành một trong những thị trường TMĐT tăng trưởng nhanh nhất trong khu vực Đông Nam Á. Theo Sách trắng Thương mại điện tử Việt Nam, năm 2020, tốc độ tăng trưởng của TMĐT đạt mức 18%, quy mô đạt 11,8 tỷ USD và là nước duy nhất ở Đông Nam Á có tăng trưởng TMĐT 2 con số. Theo tính toán của các tập đoàn lớn thế giới như Google, Temasek và Bain&Company, nhiều khả năng quy mô của nền kinh tế số Việt Nam sẽ vượt ngưỡng 52 tỷ USD và giữ vị trí thứ 3 trong khu vực ASEAN vào năm 2025.

Tuy nhiên, nguồn nhân lực của TMĐT nhìn chung vẫn còn thiếu và yếu, cũng như hạ tầng thương mại kỹ thuật cho TMĐT chưa thuận lợi, khiến TMĐT chưa tạo được sự tin tưởng cũng như chưa phát triển mạnh mẽ. Do vậy, việc xây dựng và liên tục học hỏi, cải thiện và nâng cấp các chức năng, hệ thống cho sàn Thương mại điện tử là một giải pháp vô cùng quan trọng. Đặc biệt là các tính năng phục vụ cho khách hàng, đối tượng chủ chốt tạo ra nguồn lợi nhuận cho doanh nghiệp.

Với mục đích đưa những tiến bộ công nghệ vào phục vụ cho sàn Thương mại điện tử cũng như việc kinh doanh cho doanh nghiệp, học viên xin chọn đề tài nghiên cứu "*Áp dụng kỹ thuật học sâu cho tư vấn trên Sàn thương mại điện tử Postmart*".

2. Tổng quan về vấn đề nghiên cứu:

Hệ thống tư vấn (Recommendation system) dần trở thành một thành phần không thể thiếu của các sản phẩm điện tử có lượng người dùng lớn. Các sản phẩm cá nhân hóa điện tử ngày càng phổ biến với mục đích mang sản phẩm phù hợp tới người dùng hoặc giúp người dùng có các trải nghiệm tốt hơn trên nền tảng. Nếu quảng cáo sản phẩm tới đúng người dùng, khả năng các món hàng được mua nhiều hơn. Nếu tư vấn một video mà người dùng nhiều khả năng thích hoặc tư vấn kết bạn đến đúng đối tượng, họ sẽ ở lại trên nền tảng của bạn lâu hơn.

Thương mại điện tử là nơi hệ thống tư vấn được sử dụng nhiều nhất dưới dạng quảng cáo hướng đúng đối tượng. Quảng cáo điện tử ngoài việc giúp các doanh nghiệp bán được nhiều hàng còn giúp họ tiết kiệm được chi phí kho bãi. Họ sẽ không cần các cửa hàng ở vị trí thuận lợi để thu hút khách hàng hay phải trưng ra mọi mặt hàng ở vị trí đắc địa nhất trong cửa hàng. Mọi thứ có thể được cá nhân hóa sao cho mỗi người dùng nhìn thấy những sản phẩm khác nhau phù hợp với nhu cầu và sở thích của họ.

3. Mục đích nghiên cứu:

Nghiên cứu và xây dựng Hệ thống Tư vấn (Recommendation system) cho Sàn thương mại điện tử Postmart, cụ thể là tư vấn, gợi ý các sản phẩm thông qua việc mua sắm và tìm kiếm của khách hàng.

4. Đối tượng và phạm vi nghiên cứu:

Đề án tập trung vào nghiên cứu phương pháp xây dựng Hệ thống tư vấn (Recommendation system) dành cho sàn thương mại điện tử Postmart.

Trong đó, phạm vi nghiên cứu ở đây chỉ tập trung vào xây dựng hệ thống tư vấn gồm xử lí ngôn ngữ tự nhiên, phân loại ý định và xây dựng giao diện hỗ trợ xây dựng tìm kiếm sản phẩm, tư vấn sản phẩm giữa người dùng và hệ thống.

5. Phương pháp nghiên cứu:

Hai phương pháp hệ thống tư vấn:

- Hệ thống tư vấn dựa trên nội dung - Content based recommender systems: tức là hệ thống sẽ quan tâm đến nội dung, đặc điểm của mục tin hiện tại và sau đó tư vấn cho người dùng các mục tin tương tự.
- Hệ thống tư vấn dựa trên các user - lọc cộng tác - Collaborative filtering recommender systems: tức là hệ thống sẽ phân tích các user có cùng đánh

giá, cùng mua mục tin hiện tại. Sau đó tìm ra danh sách các mục tin khác cũng được đánh gía bởi các user này, xếp hạng và tư vấn cho người dùng. Tư tưởng của phương pháp này chính là dựa trên sự tương đồng về sở thích giữa các người dùng để đưa ra các tư vấn.

Có một điều dễ nhận thấy thì phương pháp tư vấn dựa trên nội dung đòi hỏi chúng ta phải thu thập rất nhiều thông tin về các mục tin tương tự . Chính việc xác định xem một mục tin nào là tương tự với mục tin hiện tại đòi hỏi chúng ta phải thu thập và phần tích, xử lý toàn bộ các mục tin trong cơ sở dữ liệu. Tuy nhiên với phương pháp lọc công tác chúng ta không cần quá nhiều thông tin. Đơn giản chỉ là item_id của item hiện tại, các user_id và các feedback trên item đó mà thôi nên thực tế thì phương pháp lọc cộng tác được sử dụng phổ biến hơn để xây dựng các hệ thống tư vấn

NỘI DUNG

CHƯƠNG 1. TỔNG QUAN VỀ HỆ THỐNG TƯ VẤN

1.1 Giới thiệu

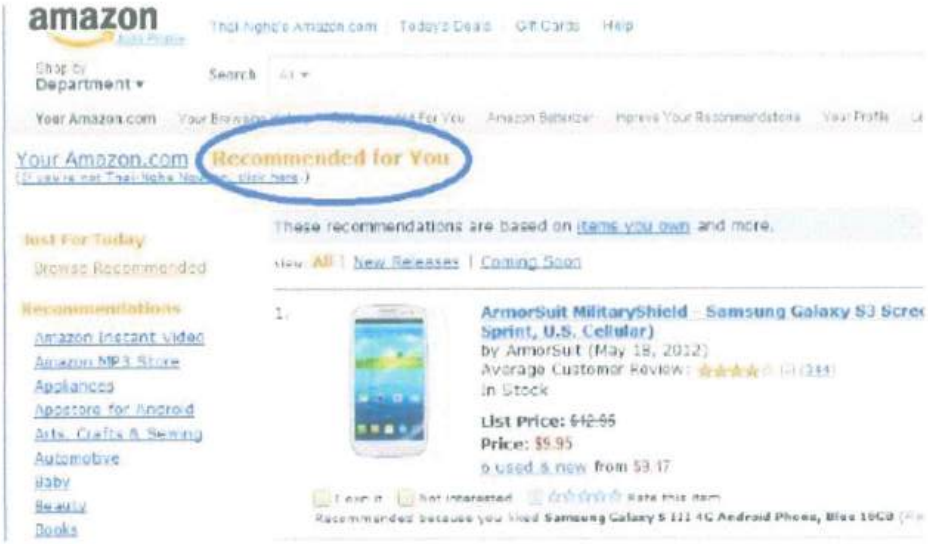
Hệ thống gợi ý (Recommendation Systems) hiện nay được xem là một trong những công nghệ cốt lõi hỗ trợ cá nhân hóa trải nghiệm người dùng trên các nền tảng số. Xuất phát từ nhu cầu xử lý và khai thác hiệu quả khối lượng thông tin ngày càng lớn, RS đóng vai trò như một công cụ thông minh giúp chọn lọc, phân tích, cũng như đề xuất những sản phẩm, dịch vụ, hoặc nội dung phù hợp nhất cho từng cá nhân. Thay vì để người dùng tự tìm kiếm trong kho dữ liệu khổng lồ, hệ thống gợi ý sẽ chủ động dự báo sở thích, đánh giá tiềm năng hoặc xếp hạng các mục thông tin (item), cho dù người dùng chưa từng tiếp cận các đối tượng đó trước đây. Các đối tượng được hệ thống gợi ý có thể rất đa dạng, từ bài hát, bộ phim, video, sách, bài báo, đến các sản phẩm tiêu dùng hoặc dịch vụ trực tuyến.

Một ví dụ điển hình là các sàn thương mại điện tử lớn như Amazon, nơi mà RS đóng vai trò rất quan trọng trong việc thúc đẩy hành vi mua sắm, nâng cao sự hài lòng và giữ chân khách hàng. Hệ thống gợi ý ở đây không chỉ sử dụng thông tin về sản phẩm, mà còn tận dụng tối đa lịch sử hoạt động của từng khách hàng như số lần xem sản phẩm, thời gian dừng lại ở từng trang, tần suất tương tác (click), các bình luận, xếp hạng và thậm chí là cả các hành vi đã bỏ qua. Dựa trên dữ liệu quá khứ này, RS có khả năng "học hỏi" từ những mẫu hành vi cũ để dự đoán chính xác hơn các sản phẩm mà người dùng có thể quan tâm trong tương lai. Ví dụ, nếu một khách hàng đã nhiều lần mua sách thuộc một chủ đề nhất định hoặc thường xuyên đánh giá cao một loại mặt hàng, hệ thống sẽ tự động ưu tiên giới thiệu các sản phẩm cùng phân khúc hoặc các mặt hàng liên quan, giúp tăng xác suất mua sắm thành công. Hình 1 dưới đây là minh họa cho quá trình này trên nền tảng bán lẻ trực tuyến.

Không chỉ giới hạn trong phạm vi thương mại điện tử, các hệ thống gợi ý ngày càng chứng tỏ hiệu quả vượt trội ở nhiều lĩnh vực khác. Trong lĩnh vực giải trí số, RS góp phần cá nhân hóa danh sách phát nhạc cho từng người nghe (ví dụ LastFM), lựa chọn phim ảnh phù hợp với thói quen và sở thích của từng cá nhân (như Netflix), hay tự động đề xuất những video được đánh giá cao trên các nền tảng như YouTube dựa vào lịch sử xem và tương tác trước đó. Ở môi trường giáo dục và đào tạo, hệ thống gợi ý hỗ trợ học viên hoặc giảng viên tiếp cận những nguồn tài liệu giá trị, các bài báo khoa học, sách giáo trình hoặc thậm chí là

những địa chỉ web học thuật phù hợp với mục tiêu học tập của mỗi người. Nhờ vậy, quá trình học tập và nghiên cứu trở nên hiệu quả và cá nhân hóa hơn.

Sự phát triển mạnh mẽ của trí tuệ nhân tạo, học máy và đặc biệt là các mô hình học sâu (deep learning) đã mở ra những cơ hội mới cho hệ thống gợi ý, giúp chúng trở nên ngày càng thông minh, chính xác và thích ứng linh hoạt với nhu cầu, thói quen thay đổi liên tục của người dùng. Việc tích hợp các thuật toán tiên tiến cùng dữ liệu lớn cho phép RS không chỉ đơn thuần là "gợi ý sản phẩm", mà còn có thể dự báo xu hướng, phát hiện hành vi bất thường hoặc đề xuất các nội dung mới mẻ, mang tính khám phá cao. Do đó, hệ thống gợi ý đang trở thành một thành phần không thể thiếu trong mọi nền tảng số hiện đại, góp phần tạo ra giá trị vượt trội cho doanh nghiệp cũng như nâng cao chất lượng trải nghiệm cho từng người dùng cá nhân.



Hình 1.1: Hệ thống gợi ý sản phẩm của Amazon

*Nguồn ảnh: https://www.amazon.com

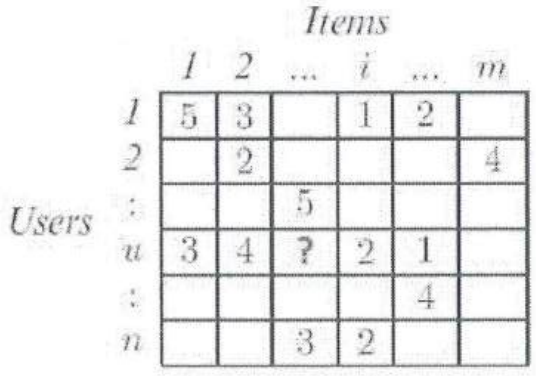
Hệ thống gợi ý không chỉ đơn thuần là một dạng Hệ thống thông tin mà nó còn là cả một lĩnh vực nghiên cứu hiện đang rất được các nhà khoa học quan tâm. Kể từ năm 2007 đến nay, hàng năm đều có hội thảo chuyên về hệ thống gợi ý của ACM (ACM RecSys) cũng như các tiểu bang dành riêng cho RS trong các hội nghị lớn khác như ACM KDD, ACM CIKM ,..

1.2 Hệ thống tư vấn (Recommendation Systems)

1.2.1 Các khái niệm chính

Trong cấu trúc của hệ thống tư vấn, ba yếu tố trọng tâm luôn được đặt lên hàng đầu, bao gồm: người sử dụng (user), đối tượng được đề xuất (item – có thể là sản phẩm, phim ảnh, ca khúc, tài liệu, v.v.), cùng với các phản hồi (feedback) mà người dùng đã để lại cho từng đối tượng đó. Phản hồi này thường thể hiện qua các đánh giá định lượng như chấm điểm, xếp hạng hoặc các hình thức biểu thị mức độ hài lòng, quan tâm đến sản phẩm/dịch vụ.

Thông tin của toàn bộ hệ thống được tổ chức dưới dạng một ma trận hai chiều, trong đó, mỗi hàng đại diện cho một người dùng cụ thể, còn mỗi cột ứng với một mục thông tin (item) trong tập dữ liệu. Giá trị tại giao điểm giữa hàng và cột chính là phản hồi mà người dùng dành cho mục tin đó – điển hình là điểm số đánh giá hoặc mức độ yêu thích. Hình 1.2 dưới đây minh họa cấu trúc này.



Hình 1.2: Ma trận biểu diễn dữ liệu trong RS (user-item-rating matrix)

*Nguồn ảnh: https://viblo.asia/p/tim-hieu-ve-content-based-filtering-phuong-phap-goi-y-dua-theo-noi-dung-phan-1-V3m5WGBg5O7)

Thông thường, không phải người dùng nào cũng phản hồi hoặc đánh giá tất cả các item trong hệ thống. Ngược lại, mỗi cá nhân chỉ tương tác với một phần rất nhỏ các sản phẩm, dịch vụ hoặc nội dung. Do đó, ma trận biểu diễn phản hồi trong thực tế có số lượng lớn các ô trống, tức là nhiều item chưa được bất kỳ người dùng nào đánh giá. Đặc điểm này tạo nên một ma trận thưa (sparse matrix), đặt ra thách thức lớn cho bài toán tư vấn.

Trọng tâm của hệ thống gợi ý là sử dụng các giá trị phản hồi đã biết trong ma trận để xây dựng mô hình học máy có khả năng dự đoán các giá trị còn thiếu – tức là ước lượng mức độ yêu thích của người dùng với các item mà họ chưa từng tiếp xúc hoặc chưa đánh giá. Sau khi hoàn tất quá trình dự đoán, hệ thống tiến hành sắp xếp các item dựa trên điểm số kỳ vọng theo thứ tự giảm dần (từ mức độ quan tâm cao nhất đến thấp nhất). Cuối cùng, hệ thống chọn ra nhóm Top-N item có điểm số cao nhất để giới thiệu cho người dùng, từ đó gia tăng khả năng đáp ứng đúng sở thích, nhu cầu cá nhân và tối ưu hóa trải nghiệm tương tác trên nền tảng.

Với đặc điểm dữ liệu thưa và số lượng item lớn như vậy, các thuật toán tư vấn hiện đại thường áp dụng những phương pháp tiên tiến như phân rã ma trận (matrix factorization), học sâu (deep learning), hoặc các kỹ thuật hybrid kết hợp nhiều nguồn thông tin để nâng cao độ chính xác của dự đoán, đồng thời giảm thiểu tác động của dữ liệu thiếu hụt và hiện tượng cold-start cho user hoặc item mới.

1.2.2 Thông tin phản hồi từ người dùng và hai dạng bài toán chính trong RS

Trong bất kỳ hệ thống tư vấn nào, thông tin phản hồi (feedback) mà người dùng để lại đối với từng mục tin (item) là một yếu tố then chốt quyết định chất lượng dự đoán. Mỗi giá trị phản hồi, ký hiệu là ruir_{ui}rui, đại diện cho mức độ tương tác, sự hài lòng, hoặc cảm nhận cá nhân mà người dùng uuu dành cho một item iii. Dữ liệu này chính là nguồn căn bản để các thuật toán học máy hoặc các phương pháp phân tích dữ liệu khai thác nhằm dự báo các lựa chọn tiếp theo.

Ý nghĩa cụ thể của giá trị phản hồi tùy thuộc vào từng hệ thống ứng dụng. Chẳng hạn, trong các nền tảng thương mại điện tử, feedback thường là các đánh giá (rating) về mức độ phù hợp hoặc yêu thích đối với sản phẩm, dịch vụ mà khách hàng đã mua hoặc trải nghiệm. Các con số này có thể biểu thị trên thang điểm (ví dụ: 1 đến 5 sao), qua nhận xét, hoặc thông qua hành vi như số lần mua lại, thêm vào giỏ hàng, thời gian xem sản phẩm, v.v. Ở lĩnh vực đào tạo trực tuyến (e-learning), phản hồi có thể thể hiện năng lực thực hiện, kết quả kiểm tra, mức độ hoàn thành các bài tập hoặc hoạt động học tập của học viên.

Nhờ việc lưu trữ và phân tích tập hợp các phản hồi từ người dùng, hệ thống tư vấn không chỉ nắm bắt được xu hướng, thói quen tiêu dùng/học tập mà còn phát hiện các mối quan hệ tiềm ẩn giữa user và item trong kho dữ liệu lớn. Đây là nền tảng để triển khai các giải thuật dự đoán, từ đó đưa ra những đề xuất tối ưu hóa theo từng cá nhân.

Giá trị rui có thể được xác định một cách tường minh (explicit feedbacks) như thông qua việc đánh giá/xếp hạng (ví dụ, rating từ ↔ đến ↔↔↔↔↔↔; hay like (1) và dislike (0),…) mà người dùng u đã bình chọn cho item i; hoặc rui có thể được xác định một cách không tường minh (implicit feedbacks) thông qua số lần click chuột, thời gian mà u đã duyệt/xem i,…

Thông thường, trong lĩnh vực hệ thống tư vấn, hai dạng bài toán chính được quan tâm gồm:

- **Dự đoán giá trị phản hồi (rating prediction):** Bài toán tập trung vào việc ước lượng điểm số hoặc đánh giá mà một người dùng cụ thể sẽ dành cho một item mà họ chưa từng tương tác trước đó.
- **Bài toán xếp hạng (top-N recommendation):** Hệ thống tìm ra danh sách N sản phẩm, dịch vụ hoặc nội dung có khả năng cao nhất sẽ nhận được phản hồi tích cực từ người dùng, từ đó đề xuất để tối ưu hóa trải nghiệm cá nhân hóa.

Cả hai dạng bài toán này đều dựa trên dữ liệu feedback thu thập được trong quá khứ, kết hợp với các mô hình dự đoán để phục vụ quá trình tư vấn, cá nhân hóa dịch vụ trong nhiều lĩnh vực khác nhau như thương mại điện tử, giải trí, giáo dục và các nền tảng số hiện đại.

1.3 Các kỹ thuật chính trong Recommendation System

Hiện tại, trong Recommendation System có rất nhiều giải thuật được đề xuất, tuy nhiên có thể gom chúng vào trong các nhóm chính: nhóm giải thuật lọc theo nội dung (content-based filtering), nhóm giải thuật lọc cộng tác (collaborative filtering), nhóm giải thuật lai ghép (hybrid filtering) và nhóm giải thuật không cá nhân hóa (non-personalization).

1.3.1 Lọc cộng tác

Một trong những chiến lược được ứng dụng rộng rãi nhất để xây dựng hệ thống tư vấn là phương pháp **lọc cộng tác** (Collaborative Filtering - CF). Ý tưởng cốt lõi của lọc cộng tác là tận dụng dữ liệu hành vi, sở thích hoặc thói quen sử dụng của số lượng lớn người dùng để dự đoán các mục tin mà một người dùng cụ thể có thể quan tâm trong tương lai. Phương pháp này không dựa vào nội dung chi tiết của từng sản phẩm, mà tập trung phân tích sự tương đồng giữa các người dùng hoặc giữa các mục tin dựa trên lịch sử tương tác. Nhờ vậy, lọc cộng tác tỏ ra đặc biệt hiệu quả ngay cả với những mặt hàng mà bản chất rất phức tạp, chẳng hạn như phim ảnh, âm nhạc hay sách, bởi nó không đòi hỏi hệ thống phải "hiểu" nội dung của từng mục.

Nhiều thuật toán đã được phát triển để đo mức độ giống nhau giữa người dùng với nhau, hoặc giữa các sản phẩm. Một số phương pháp phổ biến gồm có: k-nearest neighbor (k-NN), Pearson Correlation và các phương pháp dựa trên khoảng cách. Các thuật toán này được xây dựng trên giả định rằng những người dùng từng thể hiện sở thích giống nhau trong quá khứ thì nhiều khả năng sẽ tiếp tục có chung sở thích trong tương lai, và họ sẽ thích các sản phẩm tương tự những gì đã thích trước đây.

Khi xây dựng hệ thống dựa trên lọc cộng tác, thông tin hành vi người dùng có thể được thu thập dưới hai dạng chính: **rõ ràng (explicit)** và **ngầm ẩn (implicit)**.

- **Dữ liệu phản hồi rõ ràng** bao gồm các trường hợp người dùng trực tiếp cung cấp đánh giá hoặc ý kiến, ví dụ:
    - Đánh giá mức độ hài lòng với một sản phẩm trên thang điểm (ví dụ 1–5 sao).
    - Lựa chọn mục ưa thích từ danh sách đề xuất.
    - So sánh và xếp hạng các mục tin theo thứ tự ưu tiên.
    - Thực hiện các thao tác như tạo danh sách "yêu thích" cá nhân.
- **Dữ liệu phản hồi ngầm ẩn** xuất phát từ việc theo dõi hành vi tự nhiên của người dùng mà không yêu cầu họ phải tương tác trực tiếp với hệ thống, bao gồm:

- Thời gian người dùng xem một sản phẩm, một video hoặc đọc một bài báo.
- Lịch sử click, số lần xem, tần suất mua hàng.
- Ghi nhận các hoạt động nghe nhạc, xem phim, hoặc đọc tài liệu trên thiết bị cá nhân.
- Phân tích mối quan hệ trên mạng xã hội để phát hiện các nhóm sở thích chung, từ đó mở rộng đề xuất kết nối bạn bè hoặc nhóm.

Các hệ thống tư vấn sẽ tiến hành đối chiếu tập dữ liệu hành vi của từng người dùng với toàn bộ dữ liệu lịch sử từ cộng đồng người dùng, từ đó xác định danh sách các mục tin phù hợp nhất để tư vấn. Đáng chú ý, nhiều nền tảng thương mại điện tử và giải trí lớn đã ứng dụng thành công lọc cộng tác, ví dụ:

- **Amazon.com** với thuật toán "người mua X cũng thường mua Y", giúp đề xuất các sản phẩm liên quan dựa trên phân tích đồng mua hàng của người dùng.
- **Last.fm** cá nhân hóa danh sách phát nhạc dựa vào thói quen nghe nhạc của các thành viên có thị hiếu tương đồng.
- **Facebook, LinkedIn, MySpace** dùng lọc cộng tác để gợi ý kết bạn, nhóm hoặc các mối liên hệ mới dựa vào mạng lưới xã hội và sở thích chung.
- **Twitter** sử dụng các chỉ số tương tác và tín hiệu xã hội để đề xuất các tài khoản nên theo dõi.

Tuy nhiên, các phương pháp lọc cộng tác cũng đối mặt với ba thách thức lớn:

- **Vấn đề khởi động lạnh (Cold Start):** Khi một người dùng hoặc sản phẩm mới xuất hiện mà chưa có đủ dữ liệu tương tác, hệ thống khó có thể đưa ra các đề xuất chính xác.
- **Khả năng mở rộng:** Trong bối cảnh lượng người dùng và số lượng sản phẩm ngày càng tăng mạnh, hệ thống phải xử lý và tính toán trên quy mô dữ liệu cực lớn, đòi hỏi năng lực tính toán mạnh mẽ.
- **Tính thưa thớt của dữ liệu (Sparsity):** Dù kho dữ liệu sản phẩm khổng lồ, mỗi người dùng chỉ tương tác với một phần nhỏ, dẫn tới rất nhiều ô trống trong ma trận phản hồi, ảnh hưởng tới chất lượng dự đoán.

Để khắc phục các vấn đề này, một nhánh nghiên cứu quan trọng trong lọc cộng tác là sử dụng **phân rã ma trận (matrix factorization)**, hay còn gọi là xấp xỉ ma trận hạng thấp (low-rank matrix approximation), nhằm phát hiện các yếu tố tiềm ẩn ảnh hưởng đến sở thích người dùng và mô hình hóa mối quan hệ giữa người dùng với các sản phẩm.

Tùy theo phương pháp sử dụng, các thuật toán lọc cộng tác có thể được chia thành hai nhóm chính:

- **Lọc cộng tác dựa trên bộ nhớ (memory-based):** Sử dụng trực tiếp dữ liệu lịch sử, ví dụ như các thuật toán dựa trên người dùng hoặc dựa trên sản phẩm (user-based, item-based).
- **Lọc cộng tác dựa trên mô hình (model-based):** Sử dụng các kỹ thuật học máy, như kernel mapping, matrix factorization hoặc các mô hình học sâu để học và dự đoán sở thích từ dữ liệu lớn.

Nhờ sự đa dạng trong phương pháp và ứng dụng, lọc cộng tác đã và đang là nền tảng vững chắc cho sự phát triển của các hệ thống tư vấn hiện đại trên toàn cầu.

1.3.2 Lọc dựa trên nội dung

Bên cạnh phương pháp lọc cộng tác, **lọc dựa trên nội dung** (content-based filtering) cũng là một trong những chiến lược quan trọng và được ứng dụng phổ biến khi xây dựng hệ thống tư vấn. Phương pháp này tập trung khai thác các thông tin mô tả về đặc điểm của từng mục tin (item) cũng như hồ sơ sở thích (user profile) của người dùng, qua đó đưa ra các đề xuất cá nhân hóa.

Nguyên lý cốt lõi của lọc nội dung là hệ thống sử dụng các thuộc tính, từ khóa, hoặc đặc trưng nhận diện để mô tả các item; đồng thời xây dựng một hồ sơ phản ánh xu hướng, sở thích của từng người dùng dựa trên lịch sử tương tác hoặc đánh giá trước đó. Nói cách khác, nếu một cá nhân đã từng thích hoặc đánh giá cao một số mặt hàng, hệ thống sẽ tự động tìm và gợi ý những sản phẩm có đặc điểm tương đồng với các mục đã được yêu thích trong quá khứ.

Cơ chế hoạt động của lọc dựa trên nội dung thường gồm hai bước chính:

- **Trích xuất và biểu diễn đặc trưng của item:** Hệ thống sử dụng các thuật toán biểu diễn mục tin, điển hình là phương pháp tf-idf (term frequency – inverse document frequency) hoặc các kỹ thuật biểu diễn không gian vectơ, để xác định mức độ quan trọng của từng từ khóa, thuộc tính trong mô tả sản phẩm, bài viết, bài hát, v.v.
- **Xây dựng hồ sơ người dùng:** Có hai nguồn thông tin chủ yếu được khai thác:
    - **Mô hình ưu tiên (preference model):** Phản ánh các chủ đề, thể loại, thuộc tính mà người dùng quan tâm nhất dựa trên các item đã tương tác hoặc đánh giá.
    - **Lịch sử tương tác:** Tổng hợp các hành động như click, xem, mua, đánh giá, hoặc thậm chí là bỏ qua các mục tin để hoàn thiện chân dung sở thích của mỗi cá nhân.

Thông thường, hệ thống xây dựng hồ sơ người dùng dưới dạng một vectơ trọng số, mỗi thành phần của vectơ biểu diễn mức độ quan tâm đối với một thuộc tính hoặc đặc trưng nội dung cụ thể. Trọng số này có thể được xác định theo nhiều cách, từ tính trung bình đơn giản của các thuộc tính item từng được người dùng thích, cho đến sử dụng các thuật toán học máy như bộ phân loại Bayes (Bayesian classifiers), phân cụm (clustering), cây quyết định, hoặc mạng nơ-ron nhân tạo để dự đoán xác suất người dùng sẽ thích một item mới.

Ngoài ra, phản hồi trực tiếp từ người dùng – ví dụ như nhấn nút thích, không thích – cũng có thể được hệ thống tận dụng để điều chỉnh trọng số của từng thuộc tính, từ đó tăng độ chính xác của các đề xuất tiếp theo. Các thuật toán như Rocchio hoặc các kỹ thuật cập nhật trọng số động thường được áp dụng trong trường hợp này.

Một thách thức lớn đối với phương pháp lọc nội dung là khả năng mở rộng phạm vi sở thích của người dùng ra ngoài loại nội dung mà họ từng tiếp cận. Hệ thống này có xu hướng chỉ đề xuất các sản phẩm cùng loại với những gì người dùng đã thích trước đó, dẫn đến việc "đóng khung" trải nghiệm cá nhân hóa. Do đó, việc khuyến nghị các item thuộc loại nội dung khác – ví dụ: đề xuất video, âm nhạc hoặc sản phẩm thương mại dựa trên lịch sử đọc tin tức – thường gặp nhiều khó khăn.

Trên thực tế, nhiều hệ thống đã triển khai thành công lọc dựa trên nội dung, đặc biệt trong lĩnh vực giải trí và truyền thông số. Một ví dụ điển hình là Pandora Radio, nơi người dùng chỉ cần nhập tên một bài hát hoặc nghệ sĩ yêu thích, hệ thống sẽ tự động tạo playlist với những bản nhạc có đặc điểm tương tự về giai điệu, thể loại, hoặc cảm xúc. Trong lĩnh vực phim ảnh, các nền tảng như Rotten Tomatoes, Internet Movie Database (IMDb), Jinni, Rovi Corporation và Jaman đều sử dụng mô hình content-based để gợi ý phim dựa trên thuộc tính nội dung hoặc đánh giá trước đó. Đối với lĩnh vực tài liệu học thuật, hệ thống tư vấn giúp các nhà nghiên cứu tiếp cận nhanh chóng các bài báo hoặc tài liệu liên quan, nâng cao hiệu quả tìm kiếm và học tập. Thậm chí, trong y tế công cộng, các hệ thống cá nhân hóa đề xuất tài liệu giáo dục sức khỏe hoặc các chiến lược phòng ngừa cũng được phát triển dựa trên các đặc điểm nội dung.

Như vậy, lọc dựa trên nội dung đã và đang đóng vai trò quan trọng trong việc cá nhân hóa thông tin, đặc biệt khi hệ thống cần đề xuất các mục tin mới hoặc chưa phổ biến, hoặc khi dữ liệu hành vi cộng đồng còn hạn chế.

1.3.3 Hệ thống gợi ý lai (Hybrid recommender systems)

Nhiều nghiên cứu hiện đại đã chỉ ra rằng, sự kết hợp giữa lọc cộng tác và lọc dựa trên nội dung mang lại hiệu quả vượt trội so với việc sử dụng riêng lẻ từng phương pháp trong hệ thống gợi ý. Những hệ thống lai (hybrid recommender systems) tận dụng đồng thời điểm mạnh của nhiều phương pháp khác nhau để tăng độ chính xác, khả năng thích nghi và giảm thiểu những hạn chế thường gặp, đặc biệt trong các tình huống dữ liệu mới hoặc dữ liệu thưa thớt.

Có nhiều cách tiếp cận để xây dựng các hệ thống gợi ý lai, trong đó phổ biến nhất là:

- Tiến hành song song cả hai phương pháp lọc cộng tác và lọc dựa trên nội dung, sau đó kết hợp kết quả dự đoán của từng phương pháp bằng các kỹ thuật tổng hợp như trung bình trọng số hoặc bình chọn.

- Tích hợp đặc điểm của một phương pháp vào trong thuật toán của phương pháp còn lại, ví dụ bổ sung thông tin đặc trưng sản phẩm vào thuật toán lọc cộng tác hoặc thêm yếu tố tương tác cộng đồng vào hệ thống lọc nội dung.
- Phát triển một mô hình thống nhất kết hợp tất cả các nguồn dữ liệu và kỹ thuật dự đoán để đưa ra kết quả tối ưu.

Thực nghiệm đã cho thấy, trong nhiều trường hợp, hệ thống lai không chỉ vượt trội về độ chính xác so với các hệ thống gợi ý truyền thống mà còn góp phần khắc phục các vấn đề lớn như Cold Start (khó đề xuất cho người dùng hoặc sản phẩm mới) và dữ liệu thưa thớt (sparsity).

Netflix là một ví dụ điển hình cho ứng dụng hệ thống gợi ý lai. Nền tảng này xây dựng đề xuất dựa trên việc phân tích hành vi xem và tìm kiếm của người dùng (lọc cộng tác), đồng thời khai thác các đặc điểm nội dung của phim mà người dùng từng đánh giá cao (lọc dựa trên nội dung) để đề xuất các bộ phim tương tự.

Ngoài hai phương pháp cốt lõi, hệ thống gợi ý còn có thể tận dụng các kỹ thuật khác như:

- **Dựa trên kiến thức (knowledge-based):** Đưa ra khuyến nghị dựa trên hiểu biết về mối quan hệ giữa nhu cầu người dùng và các đặc điểm sản phẩm, thường ứng dụng nhiều trong lĩnh vực chuyên môn như tư vấn tài chính, du lịch.
- **Nhân khẩu học (demographic):** Tận dụng thông tin về độ tuổi, giới tính, nghề nghiệp, vị trí địa lý để xây dựng hồ sơ người dùng và đề xuất sản phẩm phù hợp với từng phân khúc.

Dù mỗi kỹ thuật đều có ưu và nhược điểm, nhưng hệ thống lai khi được thiết kế hợp lý sẽ bổ trợ các điểm mạnh cho nhau. Ví dụ, các phương pháp cộng tác thường gặp khó khăn với người dùng hoặc sản phẩm mới do thiếu dữ liệu (cold start), trong khi hệ thống dựa trên nội dung lại dễ bị giới hạn trong phạm vi các mục có tính chất tương tự nhau mà không thể mở rộng sang loại sản phẩm mới lạ. Các phương pháp dựa trên kiến thức có thể gặp khó khăn trong việc xây dựng và duy trì kho tri thức lớn (knowledge engineering bottleneck).

Thuật ngữ "hybrid recommender system" được dùng để chỉ các hệ thống mà ở đó, nhiều kỹ thuật đề xuất khác nhau được phối hợp đồng thời nhằm tạo ra danh sách sản phẩm hoặc nội dung tối ưu hóa cho người dùng.

Có bảy mô hình kết hợp phổ biến cho các hệ thống lai:

- **Weighted (Có trọng số):** Kết quả của các kỹ thuật khác nhau được gán trọng số và tổng hợp để tính điểm cuối cùng.
- **Switching (Chuyển đổi):** Hệ thống tự động lựa chọn giữa các phương pháp đề xuất dựa trên tình huống hoặc loại dữ liệu đầu vào.
- **Mixed (Hỗn hợp):** Gộp danh sách đề xuất từ nhiều phương pháp khác nhau và cùng lúc trình bày chúng cho người dùng.
- **Feature Combination (Kết hợp đặc trưng):** Kết hợp các thuộc tính hoặc đặc trưng thu thập từ nhiều nguồn trước khi đưa vào thuật toán dự đoán.
- **Feature Augmentation (Bổ sung đặc trưng):** Kết quả của một phương pháp đề xuất được sử dụng làm đầu vào (hoặc đặc trưng bổ sung) cho phương pháp tiếp theo.
- **Cascade:** Các thuật toán được xếp thành chuỗi, mỗi phương pháp lọc dữ liệu đầu ra của phương pháp trước theo mức độ ưu tiên.
- **Meta-level (Cấp độ meta):** Một kỹ thuật học được sử dụng để tạo ra mô hình hoặc hồ sơ, sau đó mô hình này đóng vai trò đầu vào cho thuật toán đề xuất khác.

Việc phối hợp linh hoạt các kỹ thuật nói trên giúp hệ thống tư vấn không những khắc phục hiệu quả các vấn đề truyền thống mà còn nâng cao khả năng cá nhân hóa, mở rộng phạm vi ứng dụng và thích nghi tốt với nhiều lĩnh vực từ giải trí, thương mại điện tử đến giáo dục và y tế.

1.3.4 Các kỹ thuật không cá nhân hóa

Trong nhóm kỹ thuật này, do chúng khá đơn gi ản, dễ cài đặt nên nên thường được các website/hệ thống tích hợp vào, gồm cả các website thương mại, website tin tức, hay giải trí. Chẳng hạn như trong các hệ thống bán hàng trực tuyến, người ta thường gợi ý các sản phẩm được xem/mua/bình luận/.. nhiều nhất; gợi ý các sản phẩm

mới nhất; gợi ý các sản phẩm cùng loại/ cùng nhà sản xuất/..; gợi ý các sản phẩm được mua/chọn cùng nhau. Một ví dụ khá điển hình là thông qua luật kết hợp (như Apriori), Amazon đã áp dụng khá thành công để tìm ra các sản phẩm hay được mua cùng nhau như minh họa trong Hình 1.3.

Tuy vậy, bất lợi của các phương pháp này là không cá nhân hóa cho từng người dùng, nghĩa là tất cả các user đều được gợi ý giống nhau khi chọn cùng sản phẩm.

Hình 1.3: Gợi ý sản phẩm thường được mua cùng nhau

*Nguồn ảnh: https://www.amazon.com

1.4 Deep learning trong hệ thống khuyến nghị

Trong lĩnh vực học máy hiện đại, **học sâu (Deep Learning - DL)** đã trở thành một chủ đề thu hút rất nhiều sự quan tâm, nổi bật không chỉ về mặt lý thuyết mà còn về tính ứng dụng trong thực tiễn. Tuy nhiên, phải đến khoảng năm 2016, các phương pháp học sâu mới bắt đầu được chú ý mạnh mẽ trong nghiên cứu và triển khai các hệ thống gợi ý (recommender systems). Minh chứng rõ nét là sự xuất hiện của hội thảo chuyên sâu "Deep Learning for Recommender Systems" tại sự kiện ACM RecSys

2016, đánh dấu một bước ngoặt quan trọng đưa học sâu trở thành xu hướng chủ đạo trong cộng đồng nghiên cứu hệ thống khuyến nghị.

Một trong những kiến trúc nổi bật thuộc họ học sâu, đặc biệt phù hợp với các bài toán liên quan đến dữ liệu tuần tự, là **mạng nơ-ron hồi quy (Recurrent Neural Networks - RNN)**. Điểm mạnh lớn nhất của RNN là khả năng ghi nhớ và tích hợp thông tin từ các sự kiện trước đó trong chuỗi, nhờ vào các liên kết hồi tiếp trong kiến trúc mạng. Tính chất này cho phép mô hình hóa hiệu quả các phiên hoạt động của người dùng (user sessions) – nơi mà hành động, lựa chọn hoặc tương tác của người dùng tại mỗi thời điểm đều có thể chịu ảnh hưởng bởi lịch sử tương tác trong quá khứ.

Việc sử dụng RNN trong hệ thống tư vấn mở ra khả năng dự đoán chính xác hơn về ý định, sở thích hoặc bước tiếp theo của người dùng. Ví dụ, trong một chuỗi mua sắm trực tuyến, RNN có thể phân tích quá trình duyệt, tìm kiếm hoặc mua hàng liên tục, từ đó nhận diện các mẫu hành vi, dự báo nhu cầu sản phẩm tiếp theo và đề xuất các lựa chọn phù hợp nhất cho từng cá nhân. Nhờ khả năng học hỏi từ dữ liệu tuần tự, RNN đã chứng minh ưu thế vượt trội trong các bài toán gợi ý theo phiên (session-based recommendation) – một lĩnh vực ngày càng được quan tâm trong thương mại điện tử, truyền thông số và nhiều nền tảng trực tuyến khác.

1.5 Hệ thống gợi ý tin tức

Các nền tảng tin tức trực tuyến nổi tiếng như Google News, Yahoo! News, The New York Times, Washington Post cùng nhiều cổng thông tin điện tử khác hiện đang thu hút một lượng độc giả khổng lồ trên toàn thế giới. Để nâng cao trải nghiệm cá nhân hóa và giúp người dùng tiếp cận những nội dung phù hợp, các hệ thống gợi ý tin tức đã được áp dụng rộng rãi, dựa trên nhiều phương pháp như lọc dựa trên nội dung, lọc cộng tác cũng như các mô hình lai kết hợp.

Tuy nhiên, xây dựng một hệ thống khuyến nghị hiệu quả cho môi trường tin tức trực tuyến vẫn đặt ra hàng loạt thách thức lớn mà các phương pháp truyền thống chưa thể giải quyết triệt để:

- **Tính thưa thớt trong hồ sơ người dùng:** Phần lớn độc giả truy cập các trang tin tức ở chế độ ẩn danh, hoặc chỉ đọc một lượng rất nhỏ các bài viết so với kho lưu trữ khổng lồ. Do đó, số lượng phản hồi, đánh giá hay dữ liệu lịch sử mà hệ thống thu thập được trên mỗi người dùng là cực kỳ hạn chế. Ma trận bài viết-người dùng vì vậy trở nên đặc biệt thưa thớt, gây khó khăn cho các thuật toán gợi ý dựa trên dữ liệu quá khứ.
- **Tốc độ xuất bản tin bài mới:** Mỗi ngày, các cổng tin tức hàng đầu như The New York Times có thể đăng tải hàng trăm bài viết mới, khiến kho nội dung thay đổi liên tục với tốc độ cao. Điều này làm trầm trọng thêm bài toán cold-start, bởi hầu hết các bài viết vừa được xuất bản chưa kịp thu thập đủ lượng tương tác để phục vụ việc đề xuất. Ngoài ra, với những hệ thống tổng hợp và phân phối tin tức, khối lượng dữ liệu lớn còn đặt ra thách thức về khả năng mở rộng, khi hệ thống phải xử lý và phân tích hàng nghìn bài báo trong thời gian thực.
- **Vòng đời thông tin ngắn và thị hiếu biến động:** Tin tức là lĩnh vực mà giá trị của thông tin giảm dần rất nhanh theo thời gian; phần lớn độc giả ưu tiên các bài báo mới, cập nhật nóng hổi. Vì vậy, mỗi bài viết chỉ có "thời gian sống" ngắn ngủi trước khi bị thay thế bởi các chủ đề mới. Ngoài ra, sở thích và nhu cầu thông tin của người dùng cũng không ổn định, thường xuyên thay đổi theo bối cảnh xã hội, sự kiện thời sự, vị trí địa lý hay thậm chí là tâm trạng cá nhân trong từng phiên truy cập. Một sự kiện đặc biệt hoặc tin nóng có thể ngay lập tức làm thay đổi hành vi và mối quan tâm của số đông độc giả chỉ trong thời gian rất ngắn

Tất cả các đặc điểm này đòi hỏi hệ thống gợi ý tin tức phải luôn thích ứng nhanh với dữ liệu động, có khả năng học và dự đoán tốt ngay cả khi thiếu thông tin lịch sử hoặc xuất hiện lượng lớn nội dung mới liên tục. Để giải quyết các vấn đề nêu trên, nhiều hướng nghiên cứu hiện đại đã phát triển các thuật toán học sâu, khai thác tối ưu thông tin thời gian thực, cũng như kết hợp nhiều nguồn dữ liệu khác nhau để nâng cao hiệu quả cá nhân hóa cho từng người đọc.

1.6 Kết luận

Chương I đã cung cấp một nền tảng lý thuyết vững chắc về các hệ tư vấn, từ lịch sử phát triển, phân loại các phương pháp cổ điển như lọc cộng tác, lọc dựa trên nội dung, cho tới các xu hướng lai ghép (hybrid). Phần này cũng làm rõ những thách thức lớn trong lĩnh vực TMĐT, như dữ liệu lớn, tính thưa thớt, cold start và sự thay đổi hành vi người dùng. Qua đó, chương này đặt ra cơ sở khoa học cho việc ứng dụng các kỹ thuật hiện đại nhằm nâng cao chất lượng tư vấn cá nhân hóa trên sàn thương mại điện tử.

CHƯƠNG 2: KỸ THUẬT HỌC SÂU CHO TƯ VẤN THƯƠNG MẠI ĐIỆN TỬ

Trong bối cảnh dữ liệu TMĐT ngày càng lớn về quy mô, đa dạng về loại hình và hành vi người dùng phức tạp, các kỹ thuật truyền thống như lọc cộng tác (Collaborative Filtering) hay lọc theo nội dung (Content-based Filtering) dần bộc lộ hạn chế về khả năng mở rộng, khả năng tự động trích xuất đặc trưng và xử lý dữ liệu phi cấu trúc (ví dụ như văn bản đánh giá, mô tả sản phẩm, log hành vi, v.v.). Sự phát triển mạnh mẽ của **học sâu** (deep learning) đã mang lại một cuộc cách mạng mới cho hệ thống tư vấn, cho phép khai thác hiệu quả các mối quan hệ phi tuyến, đồng thời tận dụng tối đa thông tin đa chiều từ dữ liệu thực tế.

Trong thương mại điện tử, các mô hình deep learning không chỉ giúp tăng cường cá nhân hóa trải nghiệm, dự báo nhu cầu tiềm năng của người dùng, mà còn mở rộng khả năng thích nghi với các bối cảnh mới như cold-start (sản phẩm mới, người dùng mới), hỗ trợ gợi ý thời gian thực, hoặc tích hợp nhiều nguồn dữ liệu không đồng nhất (text, ảnh, biểu đồ hành vi).

2.1. Tổng quan về một số mô hình mạng nơ-ron học sâu

Mạng nơ-ron học sâu (Deep Neural Networks – DNN) là nền tảng cốt lõi trong nhiều hệ thống tư vấn hiện đại. Các mô hình này có khả năng trích xuất đặc trưng phức tạp từ dữ liệu phi cấu trúc và có thể học được các mối quan hệ phi tuyến tính giữa người dùng và sản phẩm. Dưới đây là tổng quan một số kiến trúc tiêu biểu:

2.1.1. Mạng nơ-ron rộng & sâu (Wide & Deep Neural Network)

Wide & Deep Neural Network là một trong những kiến trúc nổi bật và được sử dụng nhiều trong các hệ thống tư vấn sản phẩm lớn (ví dụ: Google Play, YouTube, Amazon...). Kiến trúc này kết hợp đồng thời hai hướng tiếp cận:

- **Nhánh rộng (Wide):** Dựa trên các đặc trưng rời rạc, tuyến tính (one-hot, cross-features), giúp "ghi nhớ" các mối liên hệ trực tiếp từ dữ liệu lịch sử, tận dụng các quy luật thủ công, quy tắc kinh nghiệm

- **Nhánh sâu (Deep):** Sử dụng các lớp embedding, mạng dense nhiều tầng, giúp "học tổng quát hóa" những mối quan hệ phi tuyến, phát hiện các tương tác phức tạp giữa các đặc trưng mà mô hình tuyến tính không phát hiện được.

Sự kết hợp này cho phép mô hình vừa ghi nhớ tốt các trường hợp phổ biến (frequent patterns), vừa học được các quy luật mới hoặc các tổ hợp hiếm gặp (rare patterns), từ đó nâng cao hiệu quả dự đoán trên dữ liệu lớn, phức tạp và đa chiều của thương mại điện tử.

Ứng dụng thực tiễn:

- Gợi ý sản phẩm cá nhân hóa dựa trên hành vi (click, mua, xem, đánh giá)
- Dự đoán khả năng mua lại sản phẩm, tỷ lệ chuyển đổi, xác suất click quảng cáo
- Tích hợp hiệu quả cả đặc trưng rời rạc (category, user ID, time,...) và đặc trưng liên tục (giá, thời lượng xem,...)

2.1.2. Mạng nơ-ron hồi quy (RNN) và các biến thể

RNN (Recurrent Neural Network) và các biến thể như **LSTM**, **GRU** thường được ứng dụng mạnh mẽ cho bài toán gợi ý theo chuỗi hành vi (sequence-based, session-based recommendation), nơi mà dữ liệu người dùng là một loạt hành động liên tiếp trong phiên truy cập

- **Ưu điểm:** Có khả năng "ghi nhớ" ngữ cảnh lịch sử, dự đoán hành động tiếp theo tốt hơn so với mô hình không tuần tự
- **Ứng dụng:** Dự đoán sản phẩm tiếp theo mà khách hàng có thể click/mua, phân tích log truy cập trong các phiên duyệt web, cá nhân hóa trải nghiệm theo từng chuỗi hành vi ngắn hạn (ví dụ: khách vãng lai không đăng nhập)

2.1.3. Kiến trúc Attention và Transformer

Sự xuất hiện của **Transformer** với cơ chế **Attention** đã tạo bước đột phá về độ chính xác và khả năng học biểu diễn phức tạp trong hệ thống tư vấn hiện đại. Transformer có thể:

- Học mối quan hệ giữa các sự kiện trong chuỗi bất kể khoảng cách vị trí (non-sequential)

- Nhấn mạnh trọng số (attention) vào các hành vi có ý nghĩa nhất, từ đó tăng chất lượng đề xuất
- Dễ dàng song song hóa khi training trên tập dữ liệu lớn

Ứng dụng thực tế:

- Gợi ý sản phẩm dựa trên log hành vi dạng chuỗi dài/ngắn, phù hợp cả với dữ liệu nhiều user ẩn danh hoặc dữ liệu cực kỳ lớn, cập nhật liên tục.

Ví dụ nổi bật: SASRec, BERT4Rec, NARM,... giúp cải thiện chất lượng dự đoán sản phẩm tiếp theo trong session

2.1.4. Mạng tự mã hóa (Autoencoder)

Autoencoder thường dùng cho:

- Phục hồi, hoàn thiện dữ liệu thiếu (dự đoán các giá trị còn trống trong ma trận user-item)
- Học biểu diễn ẩn (latent representation) cho user, item.
- Kết hợp tốt với các mô hình deep khác để tăng khả năng tự động trích xuất đặc trưng

2.2. Phân tích bài toán tư vấn sản phẩm trên sàn TMĐT Postmart

2.2.1. Định nghĩa bài toán gợi ý trên sàn TMĐT Postmart

Bài toán đặt ra: Trên sàn TMĐT Postmart, người dùng có hành vi đa dạng: xem sản phẩm, click, mua, đánh giá, thêm vào giỏ hàng, yêu thích... Dữ liệu này được ghi nhận chi tiết qua các bảng như ds_goods (sản phẩm), ds_order (đơn hàng), ds_ordergoods (sản phẩm trong đơn), ds_evaluategoods (đánh giá sản phẩm) và thông tin người dùng (ds_member)

Mục tiêu: Đưa ra danh sách gợi ý sản phẩm phù hợp nhất với mỗi khách hàng tại từng thời điểm, tăng tỷ lệ chuyển đổi, giá trị giỏ hàng, giữ chân và sự hài lòng khách hàng

2.2.2. Luồng dữ liệu và ý nghĩa của từng bảng của sàn TMĐT Postmart

a) Bảng ds_goods – Sản phẩm

- Lưu tất cả thông tin sản phẩm đang kinh doanh: tên, giá, thương hiệu, lượt xem (goods_click), lượt bán (goods_salenum), lượt yêu thích

(goods_collect), số sao đánh giá (evaluation_good_star), thuộc tính sản phẩm...

- **Ý nghĩa:** Giúp xây dựng profile từng sản phẩm, dùng để embedding đặc trưng và tạo features cho mô hình gợi ý

b) Bảng ds_order – Đơn hàng

- Ghi nhận toàn bộ đơn hàng đã phát sinh: ai mua, thời gian nào, trị giá bao nhiêu, trạng thái gì...
- **Ý nghĩa:** Cho phép xác định lịch sử mua sắm của từng người dùng, xây dựng vector hành vi, xác định session, phát hiện xu hướng cá nhân.

c) Bảng ds_ordergoods – Sản phẩm trong đơn

- Lưu các sản phẩm cụ thể xuất hiện trong từng đơn hàng: mã sản phẩm, số lượng, giá, thuộc tính, v.v.
- **Ý nghĩa:** Làm rõ hành vi đồng mua (co-purchase), phục vụ thuật toán kiểu "người mua A thường mua cùng B", xây dựng context trong session-based recommend

d) Bảng ds_evaluategoods – Đánh giá sản phẩm

- Lưu nội dung đánh giá, số sao, thời gian đánh giá của người dùng cho từng sản phẩm
- **Ý nghĩa:** Dữ liệu giúp xác định mức độ hài lòng, độ tin cậy của sản phẩm, là feature đầu vào để ranking gợi ý

e) Bảng người dùng (ds_member hoặc cột dữ liệu buyer_id, buyer_name trong bảng ds_order)

- Thông tin đăng ký, nhận diện cá nhân, vùng miền, lịch sử truy cập, đăng nhập lần cuối, v.v.
- **Ý nghĩa:** Xác định người dùng, giúp cá nhân hóa gợi ý dựa trên đặc điểm riêng (tuổi, giới, vùng, nhóm khách hàng...)

2.2.3. Đặc thù của dữ liệu Postmart trong bài toán gợi ý

- **Rất nhiều sản phẩm cùng loại, cùng thương hiệu** → embedding sản phẩm quan trọng

- **Nhiều user truy cập không đăng nhập** → cần gợi ý theo session, không phải theo profile.
- **Dữ liệu hành vi phong phú**: không chỉ mua mà còn click, đánh giá, yêu thích → tận dụng tối đa cho các loại recommendation (Next-item, Personalized, Popularity-based...).

2.2.4. Đề xuất mô hình cho sàn TMĐT Postmart

Trên cơ sở phân tích đặc thù dữ liệu và yêu cầu nghiệp vụ của sàn TMĐT Postmart, cũng như khảo sát ưu nhược điểm của các mô hình mạng nơ-ron học sâu, **mô hình Wide & Deep Neural Network** được lựa chọn làm giải pháp trọng tâm cho hệ thống tư vấn sản phẩm cá nhân hóa. Việc đề xuất mô hình này dựa trên các lý do sau:

- **Khả năng tận dụng đồng thời kiến thức kinh nghiệm và khả năng học sâu quy luật phi tuyến**
    - Thành phần "wide" cho phép tận dụng tối đa các quy tắc đặc trưng, các mối liên hệ mạnh, tuyến tính đã được kiểm chứng từ dữ liệu lịch sử, ví dụ: người dùng từng mua các sản phẩm thuộc nhóm A thường tiếp tục mua nhóm B, hoặc khách hàng từ vùng miền nhất định thường ưu tiên sản phẩm bản địa
    - Thành phần "deep" có khả năng học tự động các biểu diễn phức tạp, phát hiện những mối quan hệ phi tuyến giữa các trường dữ liệu mà mô hình tuyến tính không thể giải thích, ví dụ: kết hợp giữa thói quen tiêu dùng, đặc điểm vùng miền, thời điểm truy cập và nội dung đánh giá

- **Phù hợp với cấu trúc dữ liệu rời rạc, đa chiều và liên tục cập nhật của Postmart**
    - Dữ liệu trên Postmart vừa có nhiều đặc trưng rời rạc (ID sản phẩm, nhóm ngành, mã user, vùng miền,…) vừa có trường liên tục (giá, số lượt mua, số sao đánh giá…)

- Wide & Deep hỗ trợ tốt việc nhúng các trường rời rạc thành vector embedding (deep), đồng thời cho phép cross-feature (wide) nhằm tận dụng đặc thù nghiệp vụ (ví dụ: combo sản phẩm, cặp user-sản phẩm phổ biến)

- **Khả năng mở rộng và linh hoạt với dữ liệu lớn**
  - Mô hình này đã được chứng minh khả năng mở rộng, huấn luyện nhanh, triển khai thực tế hiệu quả trên các nền tảng TMĐT quy mô lớn như Google Play, Amazon, Shopee.
  - Đáp ứng tốt nhu cầu cá nhân hóa cho hàng triệu user và sản phẩm, đồng thời cập nhật dễ dàng khi thêm dữ liệu mới, xuất hiện sản phẩm mới (cold start).
- **Hỗ trợ giải quyết các bài toán thực tiễn đặc thù của TMĐT Việt Nam**
  - Phù hợp với bài toán cold start nhờ khả năng generalization (deep).
  - Tận dụng tốt các tập luật mạnh sẵn có trong dữ liệu lịch sử, phù hợp với thực tế người dùng Postmart thường có hành vi mua lặp lại theo mùa vụ, sản phẩm vùng miền.
  - Linh hoạt tích hợp các dạng dữ liệu mới (review text, ảnh sản phẩm, thông tin bối cảnh…), mở rộng cho các bài toán recommendation nâng cao trong tương lai
- **Khả năng tối ưu hóa hiệu suất thực tế**
  - Dễ dàng triển khai song song, giảm độ trễ khi inference, phù hợp yêu cầu trả gợi ý gần real-time trên website/app.
  - Cấu trúc mô hình dễ tinh chỉnh (tăng giảm số lớp deep, thêm cross feature ở wide, tùy chỉnh embedding dimension…)

Từ các phân tích trên, **Wide & Deep Neural Network** là mô hình đáp ứng toàn diện cả về lý thuyết lẫn thực tiễn cho hệ thống tư vấn sản phẩm trên Postmart. Việc triển khai mô hình này hứa hẹn sẽ nâng cao chất lượng cá nhân hóa, tối ưu trải

nghiệm mua sắm của khách hàng, đồng thời tạo lợi thế cạnh tranh bền vững cho sàn TMĐT Postmart trước trên thị trường Việt Nam

2.3. Giới thiệu mô hình Wide & Deep Neural Network

Một hệ thống tư vấn có thể được xem là một công cụ tìm kiếm xếp hạng, trong đó truy vấn đầu vào là một tập hợp các đặc trưng người dùng và bối cảnh, còn đầu ra là một danh sách các vật phẩm đã được xếp hạng. Khi nhận được một truy vấn, hệ thống sẽ tìm các vật phẩm liên quan nhất trong cơ sở dữ liệu rồi xếp hạng chúng dựa trên một số tiêu chí như click hay mua hàng

Một thách thức trong hệ thống tư vấn, tương tự nhiều bài toán xếp hạng khác, là làm sao cân bằng giữa **ghi nhớ (memorization)** và **tổng quát hóa (generalization)**. Ghi nhớ được định nghĩa lỏng lẻo là việc nhận biết sự xuất hiện thường xuyên của các vật phẩm hoặc đặc trưng – nắm bắt các mối liên hệ dựa trên dữ liệu quá khứ. Tổng quát hóa là việc mở rộng tính liên hệ này sang các tổ hợp đặc trưng chưa từng xuất hiện trước đây trong dữ liệu

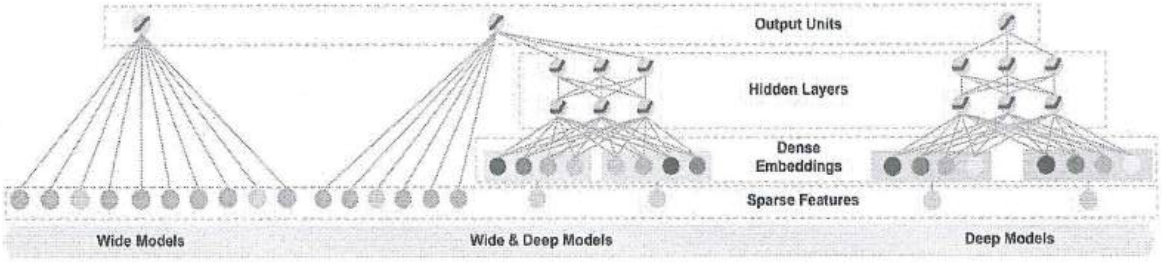
Các đề xuất dựa trên ghi nhớ thường mang tính "cục bộ", liên quan trực tiếp đến các vật phẩm mà người dùng đã từng tương tác. So với ghi nhớ, tổng quát hóa giúp tăng sự đa dạng cho các vật phẩm được đề xuất

Đối với các hệ thống tư vấn quy mô lớn, các mô hình tuyến tính tổng quát như hồi quy logistic thường được sử dụng vì đơn giản, dễ mở rộng và dễ giải thích. Các mô hình này thường huấn luyện trên các đặc trưng thưa đã được mã hóa one-hot. Ví dụ, đặc trưng nhị phân user_installed_app=netflix có giá trị 1 nếu người dùng đã cài Netflix. Ghi nhớ có thể đạt được hiệu quả nhờ các biến đổi đặc trưng dạng cross-product trên các đặc trưng thưa, như AND (user_installed_app=netflix, impression_app = pandora) – đặc trưng này có giá trị 1 nếu người dùng đã cài Netflix và vừa xem Pandora. Điều này giải thích tại sao sự đồng xuất hiện của một cặp đặc trưng lại tương quan với nhãn mục tiêu

Tổng quát hóa có thể bổ sung thêm bằng việc sử dụng các đặc trưng tổng hợp ít chi tiết hơn, như AND (user_installed_category = video, impression_category=music), nhưng việc xây dựng các đặc trưng như vậy bằng tay là tốn công sức. Một hạn chế của

các biến đổi cross-product là chúng không tổng quát hóa được cho các cặp đặc trưng chưa từng xuất hiện trong dữ liệu huấn luyện

Các mô hình dựa trên embedding, như Factorization Machines hoặc mạng nơ-ron sâu, có thể tổng quát hóa tốt hơn các cặp truy vấn–vật phẩm hiếm bằng cách học ra một véc-tơ nhúng chiều thấp cho mỗi cặp truy vấn và vật phẩm, với ít gánh nặng xây dựng đặc trưng hơn. Tuy nhiên, khó để học ra các biểu diễn véc-tơ thực sự hiệu quả nếu ma trận truy vấn-vật phẩm là quá thưa hoặc có tính thứ bậc cao (như người dùng với sở thích cực kỳ đặc biệt hoặc vật phẩm siêu ngách). Trong những trường hợp này, embedding có thể tạo ra các dự đoán không chính xác với các cặp truy vấn–vật phẩm hiếm, còn mô hình tuyến tính với cross-product lại có thể ghi nhớ được các "quy tắc ngoại lệ" này



Hình 2.1: Phổ các mô hình Wide & Deep

*Nguồn ảnh: https://bangdasun.github.io/2019/04/14/45-wide-and-deep-learning-for-recommender-systems/

Mặc dù ý tưởng đơn giản, hệ thống cho thấy Wide & Deep giúp **tăng đáng kể tỷ lệ cài đặt ứng dụng** trên app store di động, đồng thời đáp ứng được yêu cầu về tốc độ huấn luyện và phục vụ.

2.3.1. Tổng quan mô hình

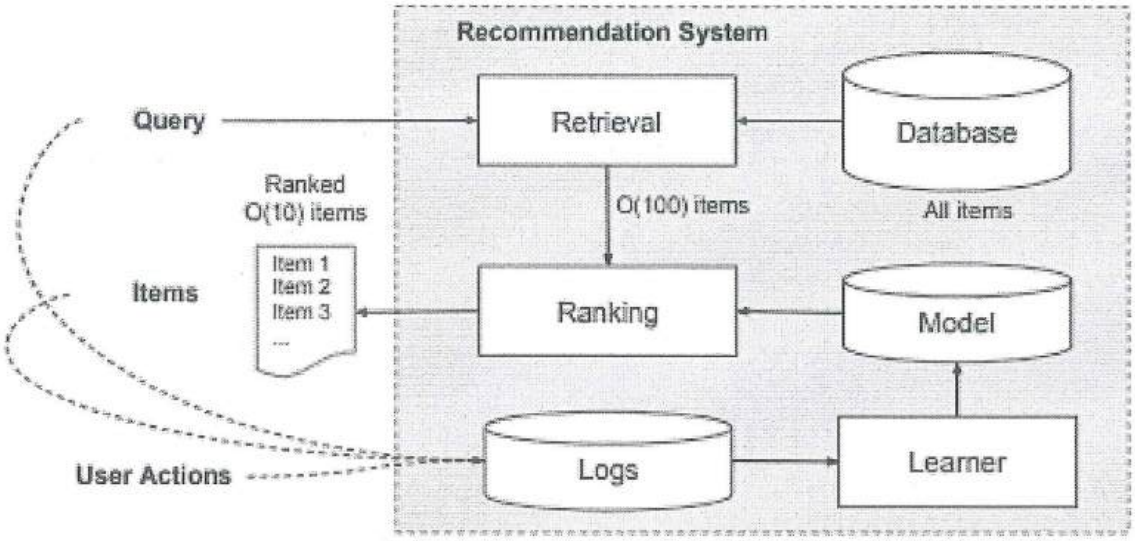
Một *truy vấn* (query), gồm nhiều đặc trưng về user và bối cảnh, được sinh ra mỗi khi user truy cập cửa hàng ứng dụng

Hệ thống tư vấn sẽ trả về một danh sách các app (hoặc vật phẩm), cho phép người dùng thực hiện các hành động như click hoặc mua hàng

Các hành động và lượt impression của người dùng được ghi vào log, phục vụ huấn luyện sau này

Vì số lượng app/vật phẩm rất lớn (trên một triệu), không thể kiểm tra toàn bộ database cho mỗi truy vấn với thời gian phục vụ thực tế (chỉ O(10) milli giây). Do đó, bước đầu tiên là *retrieval* – chọn ra tập con các vật phẩm ứng viên nhờ nhiều tín hiệu (truy vấn, hành vi, v.v

Sau khi rút gọn, hệ thống xếp hạng (ranking) sẽ tính xác suất user thực hiện một hành động (như click, mua) với từng vật phẩm, dựa trên nhiều đặc trưng như user, bối cảnh, vật phẩm và đặc trưng log (lịch sử impression/click)



Hình 2.2: Tổng quan hệ thống tư vấn

*Nguồn ảnh: https://bangdasun.github.io/2019/04/14/45-wide-and-deep-learning-for-recommender-systems/

Query sinh ra từ người dùng, retrieval rút gọn số vật phẩm, ranking dùng model để xếp hạng, kết quả dựa trên log học được từ quá khứ.

2.3.2. Thành phần Wide (The Wide Component)

Thành phần Wide là một mô hình tuyến tính tổng quát có dạng:

$$y = w^t x + b$$

trong đó:

- y là dự đoán,
- $x = [x_1, x_2, ..., x\_d]$ là vector đặc trưng
- w, b là các tham số mô hình

Tập đặc trưng đầu vào gồm cả đặc trưng thô và các đặc trưng biến đổi

Một trong những biến đổi quan trọng nhất là **biến đổi cross-product**, được định nghĩa như sau:

$$\varphi_k(x) = \prod^{(d)}_{(i=1)} x_i^{(c_{ki})}, \quad c_{ki} \in \{0, 1\}$$

Trong đó ckic_{ki}cki là biến nhị phân có giá trị 1 nếu đặc trưng thứ i thuộc biến đổi thứ k, 0 nếu không

Với đặc trưng nhị phân, một cross-product như AND(gender=female, language=en) sẽ có giá trị 1 nếu cả hai điều kiện đều đúng

Biến đổi này giúp mô hình hóa tương tác giữa các đặc trưng đầu vào và tăng tính phi tuyến cho mô hình tuyến tính tổng quát

2.3.3. Thành phần Deep (The Deep Component)

Thành phần Deep là một mạng nơ-ron truyền thẳng (feed-forward neural network), như ở Hình 1 (phải)

Với các đặc trưng phân loại, đầu vào được mã hóa one-hot (ví dụ: "language=en"). Mỗi đặc trưng phân loại high-dimensional được ánh xạ sang một véc-tơ nhúng (embedding vector) chiều thấp, tạo thành *embedding layer*.

Kích thước embedding thường từ 10 đến 100

Embedding được khởi tạo ngẫu nhiên và tối ưu hóa cùng các tham số mạng trong quá trình huấn luyện.

Mỗi tầng ẩn của mạng deep thực hiện phép biến đổi:

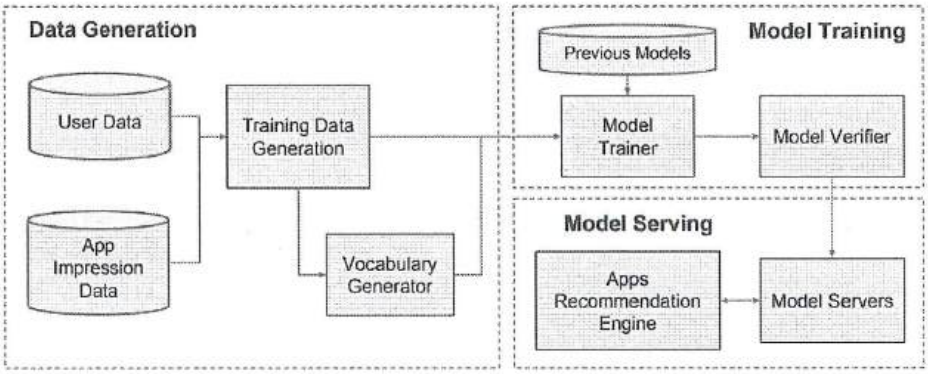
$$\mathbf{a}^{(l+1)} = f(\mathbf{W}^{(l)}\mathbf{a}^{(l)} + \mathbf{b}^{(l)})$$

Trong đó:

- **a, b, W**: Chữ in đậm (vector/matrix)
- $^{(l+1)}$, $^{(l)}$: Chỉ số trên kiểu lũy thừa, nhỏ gọn
- f là hàm kích hoạt (thường là ReLU)

2.3.4. Huấn luyện kết hợp Wide & Deep (Joint Training)

Thành phần Wide và Deep được kết hợp bằng cách cộng các output logit (đầu ra trước softmax hoặc sigmoid), rồi đưa qua hàm kích hoạt



Hình 2.3: Quy trình tổng thể hệ thống gợi ý ứng dụng.

*Nguồn ảnh: https://bangdasun.github.io/2019/04/14/45-wide-and-deep-learning-for-recommender-systems/

Huấn luyện kết hợp (Joint Training) và So sánh với Ensemble

Cần phân biệt giữa **joint training** (huấn luyện kết hợp) và **ensemble** (tổ hợp nhiều mô hình)

- Trong **ensemble**, từng mô hình thành phần (wide hoặc deep) được huấn luyện riêng biệt, kết quả được kết hợp chỉ ở giai đoạn suy luận (inference)
- Trong **joint training**, mọi tham số của hai thành phần đều được tối ưu hóa đồng thời, tận dụng tốt ưu điểm của cả hai phần

Lợi thế:

- Joint training giúp phần wide chỉ cần bổ sung những điểm yếu của deep, thay vì phải xây dựng mô hình wide riêng rẽ quy mô lớn như trong ensemble

Hàm mất mát (Loss function) trong Wide & Deep

Mô hình Wide & Deep dùng một hàm logistic loss chung cho cả hai phần. Với bài toán nhị phân (binary classification), dự đoán được tính như sau:

$$P(Y = 1|\mathbf{x}) = \sigma\left(\mathbf{w}^{\text{wide}} [\mathbf{x}, \phi(\mathbf{x})] + \mathbf{w}^{\text{deep}} \mathbf{a}^{(L)} + b \right)$$

- σ là hàm sigmoid
- $\mathbf{x}$ là vector đặc trưng gốc
- $\phi(\mathbf{x})$ là tập các cross-product feature
- $\mathbf{w}^{\text{wide}}$ là trọng số phần wide, $\mathbf{w}^{\text{deep}}$ là trọng số phần deep, $\mathbf{a}^{(L)}$ là output tầng cuối deep
- b là hệ số tự do (bias)

2.4 Kết luận

Chương 2 tập trung phân tích các mô hình mạng nơ-ron học sâu nổi bật trong hệ thống tư vấn, gồm Wide & Deep, RNN, Transformer và Autoencoder. Qua phân tích cấu trúc dữ liệu và nghiệp vụ của Postmart, đề án lựa chọn mô hình Wide & Deep Neural Network là giải pháp trọng tâm, nhờ khả năng tận dụng đồng thời các đặc trưng nghiệp vụ tuyến tính và học sâu các quan hệ phi tuyến trong dữ liệu thực tế. Kiến trúc này chứng minh sự phù hợp, tính mở rộng, dễ triển khai và hiệu quả cá nhân hóa trên quy mô lớn, đáp ứng tốt đặc thù của TMĐT Việt Nam.

CHƯƠNG 3: THỬ NGHIỆM, ĐÁNH GIÁ VÀ ĐỀ XUẤT MÔ HÌNH ÁP DỤNG CHO SÀN THƯƠNG MẠI ĐIỆN TỬ POSTMART

Chương này tập trung vào việc giải quyết bài toán gợi ý sản phẩm trên sàn thương mại điện tử **Postmart** thông qua việc ứng dụng hai kiến trúc mạng nơ-ron học sâu tiêu biểu: **mạng nơ-ron rộng & sâu (Wide & Deep Neural Network)** và **mạng nơ-ron biến đổi (Transformer-based model)**. Nội dung chương bao gồm phân tích bài toán, mô tả đặc trưng dữ liệu, thiết kế hệ thống và đánh giá kết quả thực nghiệm.

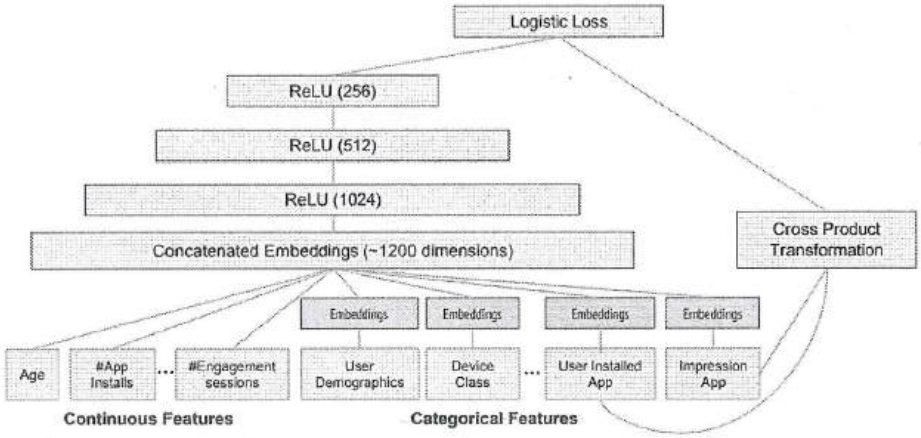
3.1. Thử nghiệm trên bộ dữ liệu Amazon Reviews for Sentiment Analysis

Quy trình gồm 4 giai đoạn: Dữ liệu, Xây dựng mô hình, Huấn luyện mô hình Wide & Deep, Kết quả thử nghiệm

3.1.1. Dữ liệu

- Dữ liệu sử dụng trong phần thực nghiệm được lấy từ bộ Amazon Reviews for Sentiment Analysis công khai trên Kaggle
- **Nguồn tải về:**
  https://www.kaggle.com/datasets/bittlingmayer/amazonreviews
  Bộ dữ liệu này được tổng hợp từ nhiều đánh giá thực tế của khách hàng Amazon, thường xuyên được sử dụng trong nghiên cứu về phân tích cảm xúc và xây dựng hệ thống tư vấn.Các chuỗi string categorical sẽ được ánh xạ thành ID (sinh từ bước Vocabulary Generator)

- Đặc trưng liên tục được chuẩn hóa về [0, 1] qua hàm phân vị.



Hình 3.1: Kiến trúc chi tiết mô hình Wide & Deep cho gợi ý ứng dụng.

- **Cấu trúc và đặc trưng dữ liệu:**

Dữ liệu có định dạng file text (test.ft.txt, train.ft.txt) với mỗi dòng gồm 1 nhãn cảm xúc và 1 câu đánh giá bằng tiếng Anh.

Cụ thể:

- **label**: Nhãn cảm xúc, gồm 2 giá trị:
  - __label__1: Đánh giá tiêu cực (Negative review)
  - __label__2: Đánh giá tích cực (Positive review)
- **review**: Văn bản đánh giá sản phẩm tự do, độ dài linh hoạt

- **Mô tả tóm tắt dữ liệu:**

Bảng 3.1: Mô tả tóm tắt dữ liệu Amazon Reviews for Sentiment Analysis

STT	Nhãn (label)	Nội dung đánh giá (review)
1	2	Great CD: My lovely Pat has one of the GREAT voices of her...
2	2	One of the best game music soundtracks ever made.
3	1	Batteries died within a year ... : I bought these batteries...
4	2	Works fine, but Maha Energy is better...
5	2	Great for the non-audiophile: Review: It works fine for...
6	1	DVD Player crapped out after one year. Don't waste your money.
7	1	Incorrect Disc: I love the style of the cover, but...
8	1	DVD menu select problems: I cannot get this DVD to play...
9	2	Unique Weird Orientalia from the 1990s...
10	1	Not an "ultimate guide": Firstly, I ...

Chú thích: Cột "Nội dung đánh giá" đã rút gọn cho ngắn gọn, dữ liệu thực tế dài hơn.

3.1.2. Xây dựng mô hình Wide & Deep

- Kiến trúc Wide & Deep

Mô hình Wide & Deep Neural Network là một kiến trúc kết hợp đồng thời hai nhánh học:

- Nhánh rộng (Wide): sử dụng các đặc trưng rời rạc, tuyến tính (linear), giúp khai thác các mối liên hệ trực tiếp, các đặc trưng dạng one-hot, TF-IDF, cross-feature

- Nhánh sâu (Deep): là mạng nơ-ron nhiều tầng (deep neural network), học biểu diễn phức tạp từ dữ liệu, thường dùng các lớp embedding, dense và các kỹ thuật học sâu

Mục tiêu của kiến trúc này là tận dụng đồng thời cả sức mạnh học biểu diễn của deep learning và khả năng tận dụng đặc trưng rời rạc giàu thông tin, giúp mô hình dự đoán tổng quát hơn và chính xác hơn.

- Kiến trúc chi tiết mô hình thực nghiệm
  - Nhánh Wide
    - **Input:** Dữ liệu dạng vector thưa (sparse vector) từ TF-IDF
    - **Layer:** Dense layer (64 units, activation ReLU) hoặc có thể để nguyên làm input linear.
    - **Ý nghĩa:** Khai thác nhanh các mối liên hệ tuyến tính, ví dụ các từ khoá mạnh mang tính quyết định nhãn cảm xúc.
  - Nhánh Deep
    - Input: Dữ liệu dạng sequence (chuỗi số nguyên từ tokenizer + padding).
    - **Embedding:** Lớp Embedding ánh xạ mỗi từ thành vector dense (64 chiều).
    - **Pooling:** Sử dụng GlobalAveragePooling1D để giảm chiều ma trận embedding thành vector đặc trưng.
    - **Dense layers:** Nối tiếp 1-2 lớp Dense (ví dụ: 64 units, ReLU) để trích xuất các đặc trưng sâu hơn.
  - Kết hợp hai nhánh: Kết quả của hai nhánh được **nối (concatenate)** lại, đưa qua một lớp Dense output (1 node, activation sigmoid) để dự đoán xác suất nhãn (phân loại nhị phân).
  - Tham số huấn luyện
    - **Loss:** Binary crossentrop
    - **Optimizer:** Ada
    - **Epochs:** 5 (có thể điều chỉnh)

- **Batch size:** 128
- **Early stopping:** Có thể thêm để tránh overfitting.

3.1.3. Huấn luyện mô hình Wide & Deep

- **Chia tập dữ liệu**

  Sau khi biểu diễn đặc trưng, toàn bộ tập dữ liệu được chia thành hai phần:

  - Tập huấn luyện: 80% dữ liệu, dùng để huấn luyện mô hình
  - Tập kiểm tra (test): 20% dữ liệu, dùng để đánh giá hiệu năng mô hình

- **Cấu hình mô hình huấn luyện**

  - Input:
    - Đầu vào "wide" là đặc trưng TF-IDF của câu đánh giá (review)
    - Đầu vào "deep" là embedding các từ trong review, đã được pad về cùng chiều dà
  - Kiến trúc: Mô hình bao gồm 2 nhánh: Wide (Linear) và Deep (Dense + Embedding), sau đó nối lại và đưa vào lớp đầu ra.
  - Hàm mất mát (Loss function): Sử dụng binary_crossentropy (vì đây là bài toán phân loại nhị phân - positive/negative review)
  - Hàm tối ưu (Optimizer): Sử dụng Adam với learning rate mặc định.
  - Các chỉ số đánh giá: Accuracy, MAE, MSE, Precision, Recall, F1-score

- **Tiến hành huấn luyện**
  - Số epoch: 5, 50, 100 (có thể điều chỉnh để tránh overfitting)
  - Batch size: 512
  - Sử dụng callback để lưu lại mô hình tốt nhất theo validation loss

Đọc file dữ liệu

```
import pandas as pd
df = pd.read_csv('test.ft.txt', sep='\t', header=None)
df['label'] = df[0].str.extract(r'__label__(\d)').astype(int)
df['review'] = df[0].str.replace(r'__label__\d ', '', regex=True)
X = df['review']
y = df['label']
```

Biểu diễn đặc trưng cho Wide (TF-IDF)

Mục tiêu: Biến văn bản thành vector đặc trưng (dạng rời rạc, độ dài cố định) cho nhánh "wide"

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(max_features=1000)

X_wide = tfidf.fit_transform(df['review']).toarray()
```

Biểu diễn đặc trưng cho Deep (Embedding/Tokens)

Mục tiêu: Biến mỗi câu thành một dãy số (token), dùng cho embedding layer ở nhánh "deep".

```
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

max_len = 100
num_words = 20000

tokenizer = Tokenizer(num_words=num_words)
tokenizer.fit_on_texts(df['review'])

X_seq = tokenizer.texts_to_sequences(df['review'])
X_deep = pad_sequences(X_seq, maxlen=max_len)
```

Tiền xử lý dữ liệu

- Chuyển đổi label thành số (ví dụ: __label__1 là 0 - tiêu cực, __label__2 là 1 - tích cực)
- Làm sạch dữ liệu text (loại bỏ ký tự đặc biệt, chuyển về chữ thường…)

```
df['label_num'] = df['label'].apply(lambda x: 1 if x == '__label__2' else 0)

import re
def clean_text(text):
    text = text.lower()
    text = re.sub(r'[^\w\s]', '', text)
    return text

df['review_clean'] = df['review'].apply(clean_text)
print(df.head())
```

Tách train/test

Chia dữ liệu thành hai phần để huấn luyện và kiểm thử

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    df['review_clean'], df['label_num'], test_size=0.2, random_state=42)
```

Biến đổi văn bản thành vector số

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max_features=5000)
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)
```

Huấn luyện mô hình đơn giản (Logistic Regression)

Để kiểm tra quy trình

```
from sklearn.linear_model import LogisticRegression

clf = LogisticRegression(max_iter=1000)
clf.fit(X_train_vec, y_train)

y_pred = clf.predict(X_test_vec)
```

Đánh giá mô hình

Hiển thị các chỉ số: accuracy, precision, recall, f1

```
from sklearn.metrics import classification_report, confusion_matrix

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[35699  4197]
 [ 3817 36287]]
              precision    recall  f1-score   support

           0       0.90      0.89      0.90     39896
           1       0.90      0.90      0.90     40104

    accuracy                           0.90     80000
   macro avg       0.90      0.90      0.90     80000
weighted avg       0.90      0.90      0.90     80000
```

Giải thích:

- 35699: Số review "negative" được dự đoán đúng (True Negative, TN)
- 4197: Số review "negative" bị đoán nhầm thành "positive" (False Positive, FP)
- 3817: Số review "positive" bị đoán nhầm thành "negative" (False Negative, FN)
- 36287: Số review "positive" được dự đoán đúng (True Positive, TP)

Ý nghĩa các chỉ số

- **Accuracy (Độ chính xác tổng thể): 0.90**
  - Mô hình đoán đúng 90% tổng số review. Đây là một con số rất cao trên tập test lớn (80.000 bản ghi)

- **Precision (Độ chính xác của từng lớp): 0.90**
  - 90% những review mà mô hình dự đoán là "positive" thực sự đúng là positive, và tương tự cho negative
- **Recall (Độ phủ, độ nhạy): 0.89 với negative, 0.90 với positive**
  - Mô hình bắt được 90% các review positive và 89% review negative trong tập test
- **F1-score: 0.90**
  - Trung bình điều hòa của precision và recall, cho thấy mô hình cân bằng tốt giữa hai chỉ số
- **Support**: Số lượng mẫu từng lớp, cân bằng giữa hai lớp (khoảng 40.000/40.000)

Đánh giá tổng quan

- **Độ chính xác 90%** cho một bài toán sentiment analysis là rất ấn tượng, đặc biệt với dữ liệu thực tế lớn như Amazon Reviews
- **Precision, Recall, F1-score đồng đều giữa hai lớp**, chứng tỏ mô hình không bị lệch (không thiên vị một lớp)
- Số lượng false positive (4197) và false negative (3817) là tương đối nhỏ so với tổng số sample (~80.000), mô hình ít bỏ sót review tích cực/tiêu cực

Nhận xét

- **Mô hình Wide & Deep** đã tận dụng tốt thông tin của cả feature đơn giản (wide - TF-IDF) lẫn biểu diễn sâu (deep - embedding), cho ra kết quả tổng quát cao, phù hợp cho ứng dụng thực tế
- **Giá trị ứng dụng:** Có thể dùng mô hình này để tự động đánh giá sentiment của các review sản phẩm mới trên Amazon, hỗ trợ hệ thống tư vấn hoặc kiểm soát chất lượng dịch vụ
- **Hạn chế:** Có thể tiếp tục tối ưu (fine-tune), tăng epoch hoặc kết hợp thêm feature metadata (user, product, time…) để nâng độ chính xác lên nữa

Kết quả đánh giá trên tập kiểm tra cho thấy mô hình Wide & Deep đạt độ chính xác 90%, với các chỉ số precision, recall và F1-score đều đạt 0.90 ở cả hai lớp tích

cực và tiêu cực. Điều này cho thấy mô hình không bị thiên lệch, nhận diện tốt các đánh giá của người dùng trên tập dữ liệu thực tế. Wide & Deep chứng minh ưu điểm rõ rệt so với các mô hình truyền thống nhờ khả năng kết hợp hiệu quả giữa đặc trưng thủ công và đặc trưng học sâu

- **Thử nghiệm mô hình Wide & Deep**

  Xây dựng mô hình

```
from tensorflow.keras.layers import Input, Embedding, Dense, Flatten, Concatenate
from tensorflow.keras.models import Model

input_wide = Input(shape=(X_train_wide.shape[1],), name='wide_input')

input_deep = Input(shape=(X_train_deep.shape[1],), name='deep_input')

embedding_dim = 64
deep = Embedding(input_dim=num_words, output_dim=embedding_dim, input_length=X_train_deep.shape[1])(input_deep)
deep = Flatten()(deep)
deep = Dense(128, activation='relu')(deep)
deep = Dense(64, activation='relu')(deep)

wide = Dense(64, activation='relu')(input_wide)

combined = Concatenate()([wide, deep])
output = Dense(1, activation='sigmoid')(combined)

model = Model(inputs=[input_wide, input_deep], outputs=output)
model.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy', 'mae', 'mse']
)

model.summary()
```

  - **Nhánh Wide:** TF-IDF giúp model tận dụng tốt từ khóa và cấu trúc ngữ nghĩa rời rạc, tăng khả năng nhận diện các pattern mạnh về tần suất
  - **Nhánh Deep:** Embedding & Dense giúp hiểu mối liên hệ sâu giữa các từ, các ngữ cảnh đặc biệt
  - **Kết hợp (Concatenate):** Cho phép model tận dụng đồng thời cả thông tin tuyến tính (wide) và phi tuyến (deep)

Huấn luyện mô hình (fit)

```
history = model.fit(
    [X_train_wide, X_train_deep], y_train,
    validation_data=([X_test_wide, X_test_deep], y_test),
    epochs=50, batch_size=256
)
```

Đánh giá mô hình (evaluate)

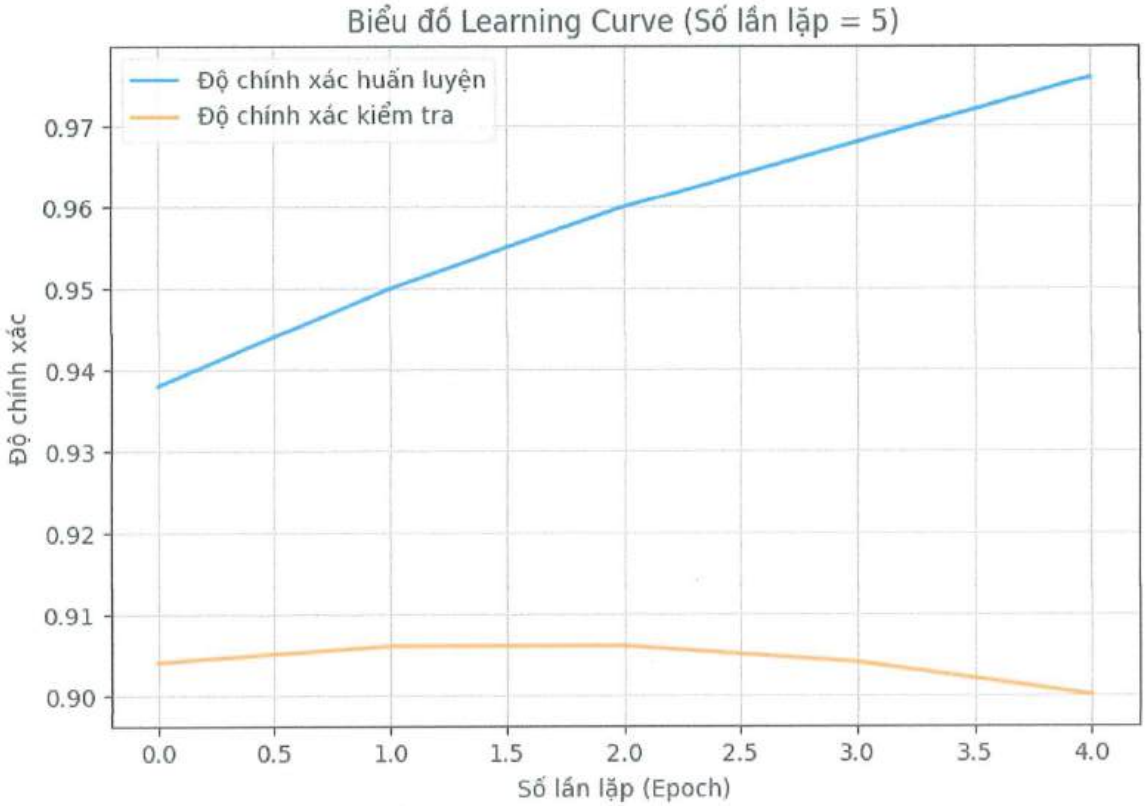
```
loss, acc = model.evaluate([X_test_wide, X_test_deep], y_test)
print('Test accuracy:', acc)
```

3.1.4. Kết quả thử nghiệm

a. 5 epoch:

Biểu đồ Learning Curve (Độ chính xác theo epoch):

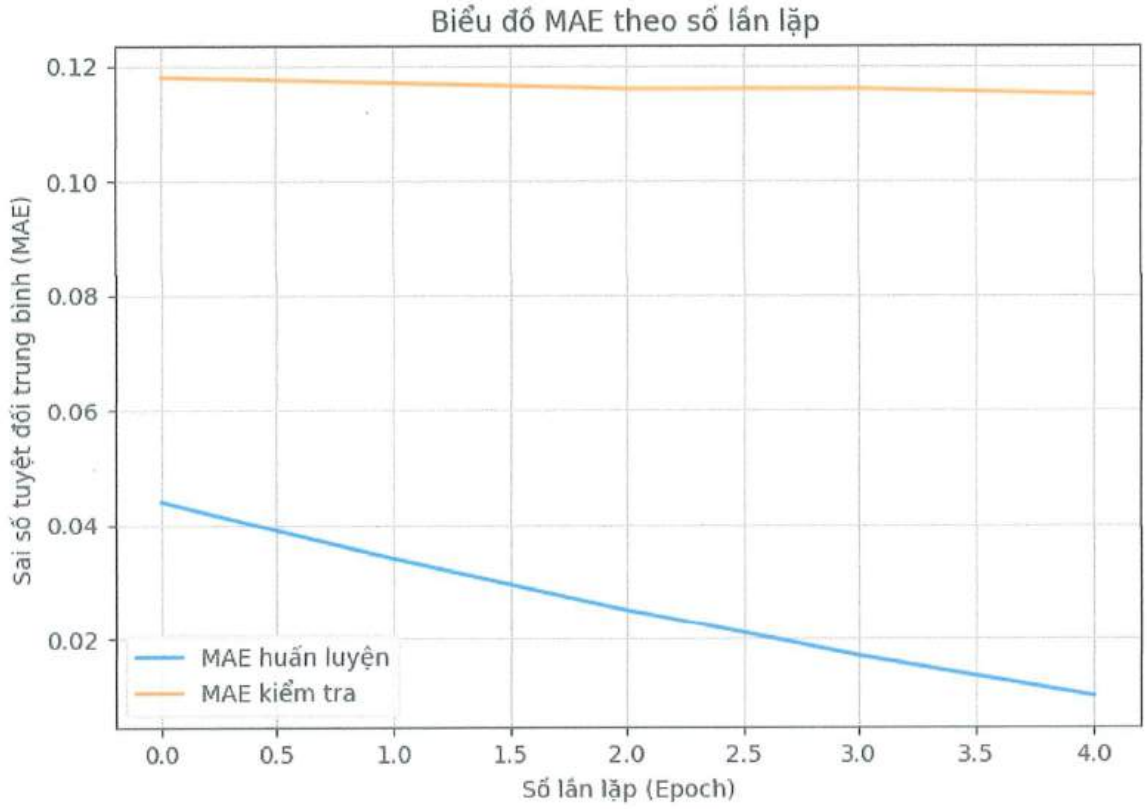
- **Đường màu xanh (Train Accuracy)**: Độ chính xác trên tập huấn luyện, tăng dần qua các epoch, thể hiện model học tốt dần lên trên chính dữ liệu huấn luyện
- **Đường màu cam (Validation Accuracy)**: Độ chính xác trên tập kiểm tra (validation). Đường này ổn định quanh mức 0.90 và có xu hướng đi ngang hoặc giảm nhẹ khi số epoch tăng
- **Khoảng cách giữa hai đường**: Nếu khoảng cách này quá lớn (train accuracy tăng mạnh còn val accuracy gần như không đổi hoặc giảm), thường là dấu hiệu **overfitting** – mô hình nhớ quá kỹ dữ liệu train nhưng không tổng quát hóa tốt ra dữ liệu mới
- **Nhận xét**: Validation accuracy luôn ổn định quanh 0.90, train accuracy tăng đều, **chưa xuất hiện overfitting nghiêm trọng** nhưng nếu tiếp tục train dài có thể sẽ xảy ra. Sự ổn định của validation accuracy cũng cho thấy mô hình tổng quát hóa tốt trên dữ liệu mới



Hình 3.2: Biểu đồ Learning Curve của tập dữ liệu Amazon với 5 epoch

Biểu đồ Learning Curve của mô hình Wide & Deep trên tập dữ liệu Amazon Reviews. Độ chính xác trên tập huấn luyện (Train Accuracy) tăng dần qua các epoch, trong khi độ chính xác trên tập kiểm tra (Validation Accuracy) ổn định quanh mức 90%. Điều này cho thấy mô hình học tốt nhưng chưa bị overfitting, khả năng tổng quát hóa tốt trên dữ liệu chưa thấy.

Biểu đồ MAE:



Hình 3.3: Biểu đồ MAE của tập dữ liệu Amazon với 5 epoch

- **Trục hoành (x)**: Số lượng epoch (vòng lặp huấn luyện).
- **Trục tung (y)**: Giá trị MA
- **Đường màu xanh (Train MAE)**: MAE trên tập huấn luyện, giảm dần theo từng epoch (mô hình học tốt dần lên trên dữ liệu train)
- **Đường màu cam (Validation MAE)**: MAE trên tập kiểm tra (validation), giữ ổn định ở mức khoảng 0.11-0.12 và chỉ giảm rất nhẹ

Nhận xét:

- **rain MAE giảm đều:** Chứng tỏ mô hình đang học và cải thiện khả năng dự đoán trên tập train
- **Validation MAE ổn định:** Đường này không giảm nhiều, thể hiện khả năng tổng quát hóa của mô hình trên dữ liệu mới không thay đổi đáng kể qua các epoch, có thể

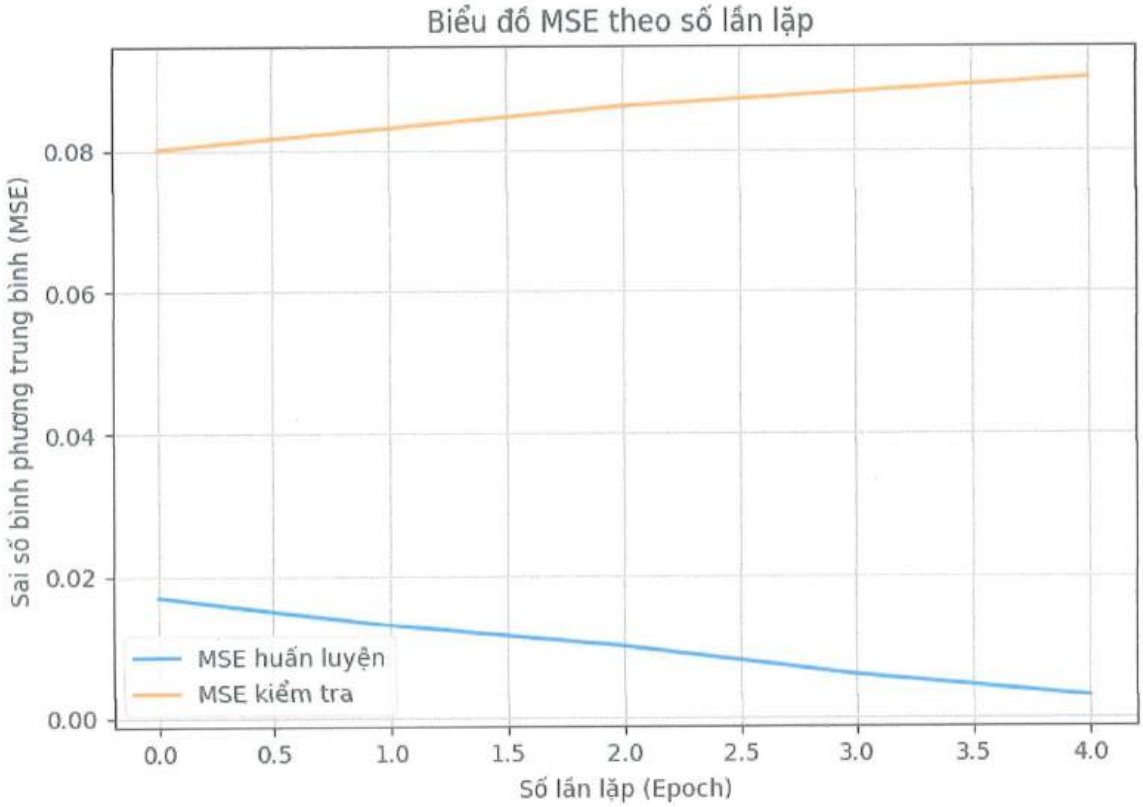
là do model đã học được những gì quan trọng ngay từ đầu hoặc dữ liệu kiểm tra có tính chất khác biệt nhẹ so với dữ liệu huấn luyện

- **Khoảng cách giữa hai đường:** Validation MAE cao hơn train MAE nhưng không quá xa, điều này **không phải là overfitting nghiêm trọng** nhưng cũng cho thấy mô hình chưa thực sự học tốt cho dữ liệu mới như đã học trên tập train. Đường validation MAE càng gần train MAE càng tốt

Đánh Giá:

- **Mô hình Wide & Deep** học tốt trên tập huấn luyện (Train MAE giảm mạnh)
- **Tổng quát hóa ở mức tốt**: Validation MAE ổn định, mô hình không bị overfitting

Biểu đồ MSE:



Hình 3.4: Biểu đồ MSE của của tập dữ liệu Amazon với 5 epoch

- **Overfitting nhẹ:**
  - Dấu hiệu điển hình là train MSE giảm đều nhưng validation MSE lại tăng nhẹ theo thời gian. Mô hình có xu hướng ghi nhớ chi tiết dữ liệu train thay vì

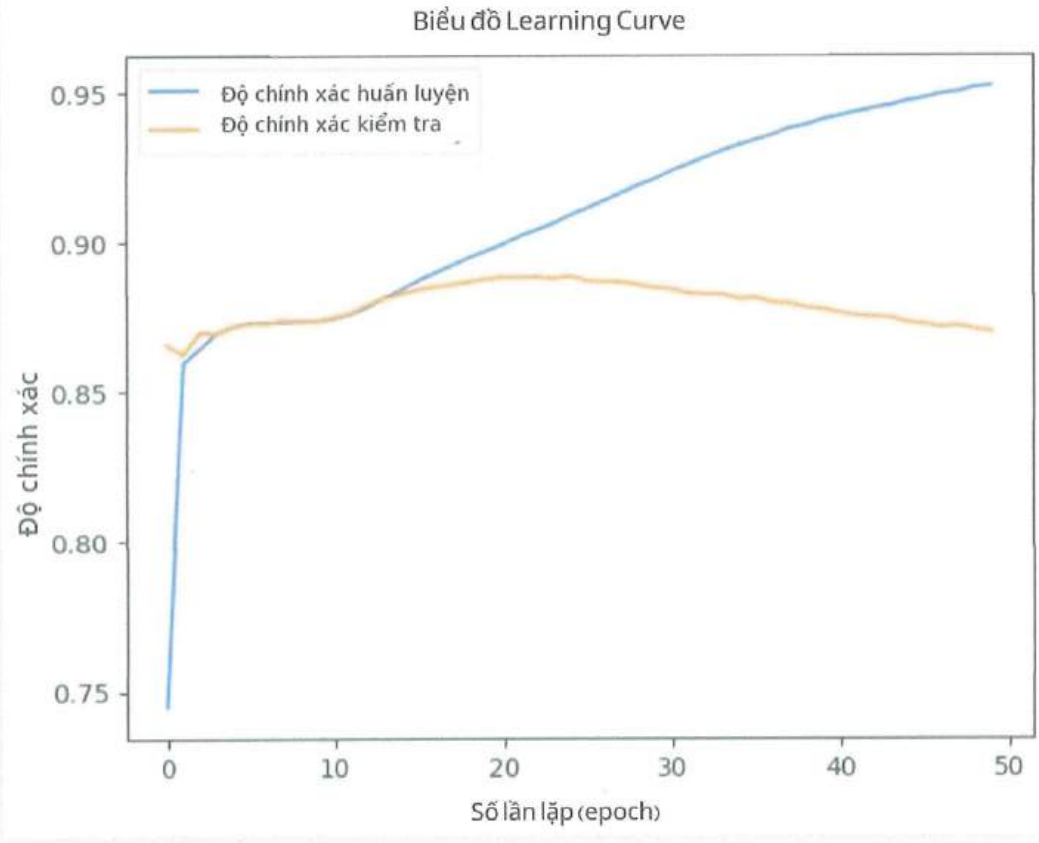
học các quy luật tổng quát, nên hiệu quả dự đoán trên dữ liệu mới không còn tốt như ban đầu

- **Vẫn trong mức kiểm soát:**
  - Tuy MSE validation tăng nhưng mức độ tăng còn nhẹ, cho thấy mô hình chỉ mới bắt đầu overfit, chưa quá nghiêm trọng.

Biểu đồ Learning Curve với chỉ số MSE cho thấy giá trị MSE trên tập huấn luyện giảm đều qua các epoch, chứng tỏ mô hình học tốt trên dữ liệu train. Tuy nhiên, MSE trên tập kiểm tra lại tăng nhẹ, đây là dấu hiệu cho thấy mô hình bắt đầu gặp hiện tượng overfitting, tức là mô hình phù hợp quá mức với dữ liệu huấn luyện nhưng lại giảm hiệu quả trên dữ liệu kiểm tra. Để cải thiện, có thể áp dụng các kỹ thuật như early stopping, regularization, hoặc bổ sung thêm dữ liệu huấn luyện

b. 50 epoch

Biểu đồ Learning Curve



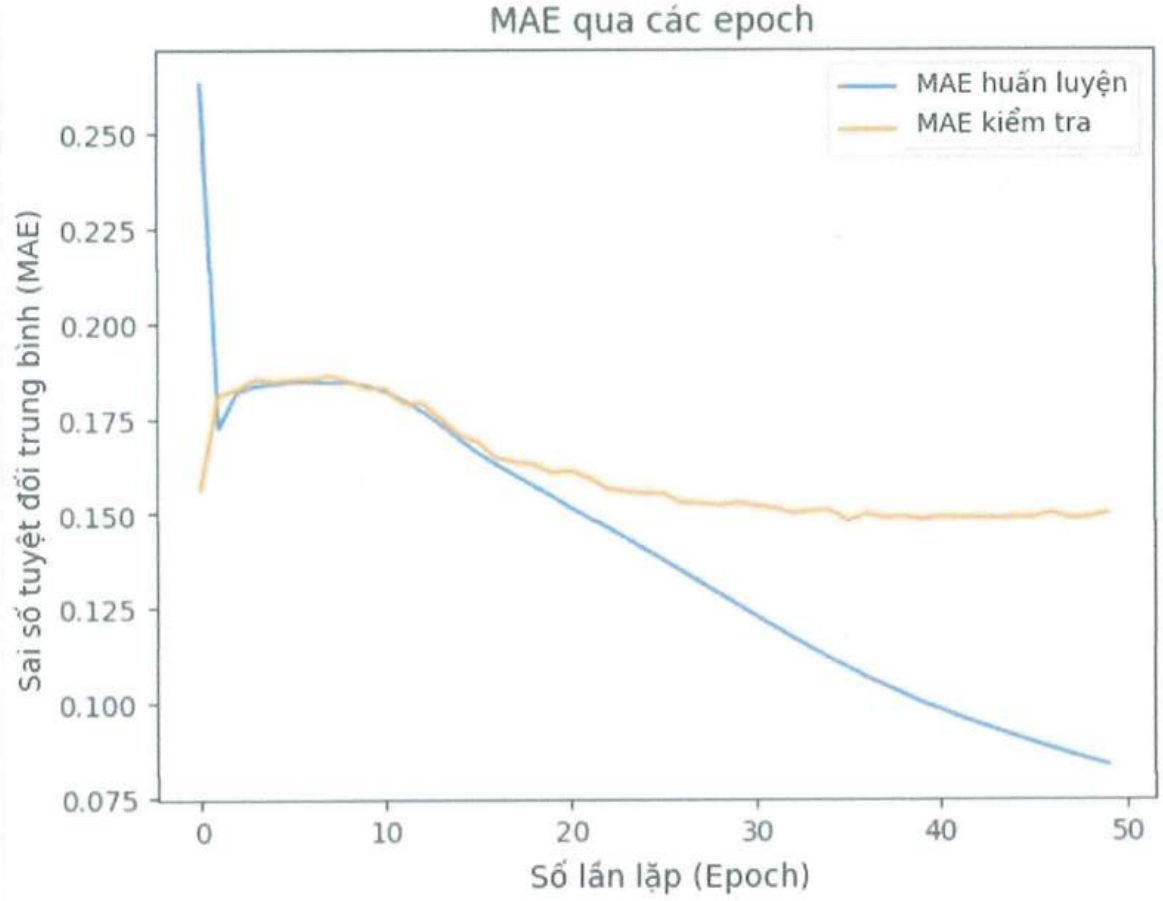
Hình 3.5: Biểu đồ Learning Curve của tập dữ liệu Amazon với 50 epoch

- Đường **train accuracy** tăng đều, từ khoảng 0.75 lên trên 0.95.
- Đường **validation accuracy** cũng tăng ban đầu, đạt đỉnh khoảng 0.89, sau đó giảm dần từ epoch 20 trở đi (còn khoảng 0.87 ở cuối 50 epoch).
- Sau epoch 20, khoảng cách giữa train và validation ngày càng lớn

→ **Nhận xét:**

- Ban đầu, mô hình **học được các quy luật chung của dữ liệu**, nên cả train và validation cùng tăng. Tuy nhiên, sau một số epoch, mô hình bắt đầu ghi nhớ các đặc trưng riêng biệt của tập train (memorization), dẫn tới overfitting
- Đường validation accuracy đi xuống trong khi train accuracy vẫn lên là dấu hiệu **overfitting kinh điển**. Điều này cho thấy mô hình Wide & Deep có đủ "công suất" để học thuộc train set nhưng chưa có đủ biện pháp regularization để tổng quát hóa cho unseen data
- Trong thực tế, nên dừng huấn luyện tại epoch mà validation accuracy đạt đỉnh (epoch 20–25)

Biểu đồ MAE



Hình 3.6: Biểu đồ MAE của tập dữ liệu Amazon với 50 epoch

- **Train MAE** giảm mạnh, về dưới 0.08 ở cuối quá trình huấn luyện
- **Validation MAE** chỉ giảm nhẹ trong 20 epoch đầu rồi dao động nhẹ, không giảm thêm dù train MAE vẫn giảm

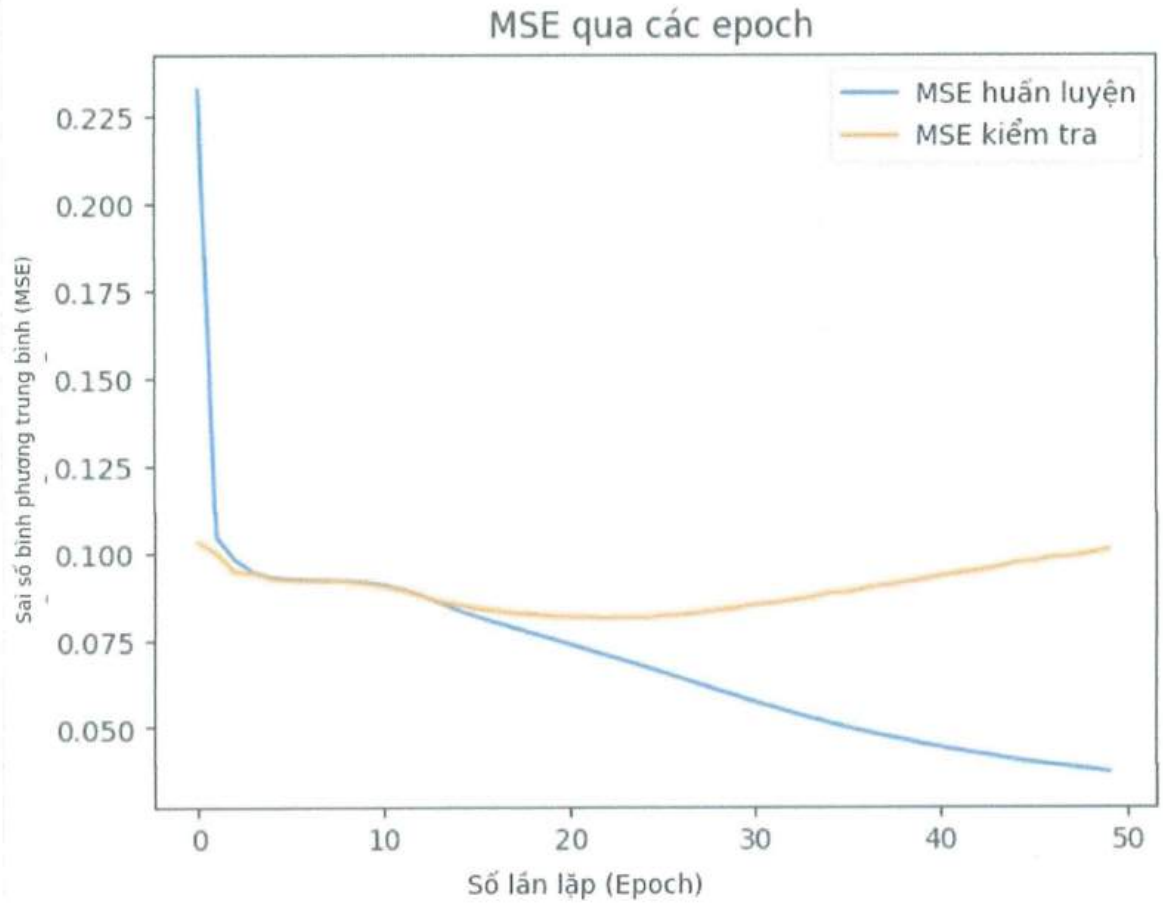
→ **Nhận xét:**

• MAE đo lường **độ lệch trung bình tuyệt đối** giữa nhãn dự báo và thực tế. Đường train MAE giảm liên tục cho thấy mô hình ngày càng "chính xác" hơn trên train set

• Validation MAE dừng giảm sớm, rồi dao động, xác nhận mô hình **không cải thiện thêm cho dữ liệu chưa thấy**

• Việc validation MAE không tăng mạnh chứng tỏ mô hình không quá "catastrophic overfit", nhưng chắc chắn đã học quá mức cho train set so với khả năng khái quát hóa

Biểu đồ MSE



Hình 3.7: Biểu đồ MSE của tập dữ liệu Amazon với 50 epoch

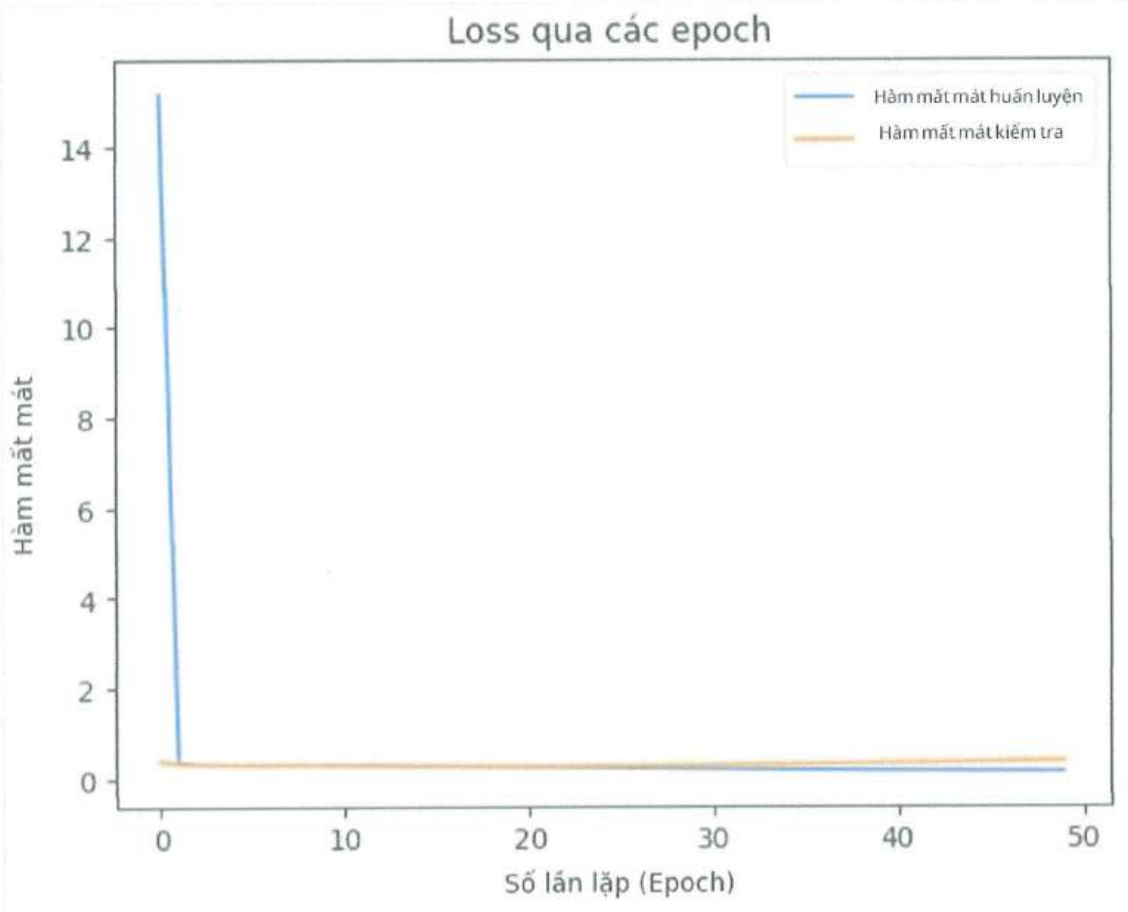
- **Train MSE** giảm ổn định, xuống rất thấp
- **Validation MSE** giảm nhẹ rồi lại tăng lên, nhất là sau epoch 20

→ **Nhận xét:**

• MSE nhạy cảm hơn MAE đối với các dự đoán sai lớn, nên MSE của validation tăng trở lại nghĩa là mô hình bắt đầu mắc các lỗi dự báo "lớn" hơn với validation set khi huấn luyện quá lâu

• Đây là biểu hiện rõ nét của **overfitting**: mô hình "đẩy mạnh" fit vào noise của train set, khiến các điểm dữ liệu lạ bị dự báo sai nghiêm trọng hơn

Biểu đồ Loss Curve



Hình 3.8: Biểu đồ Loss curve của tập dữ liệu Amazon với 50 epoch

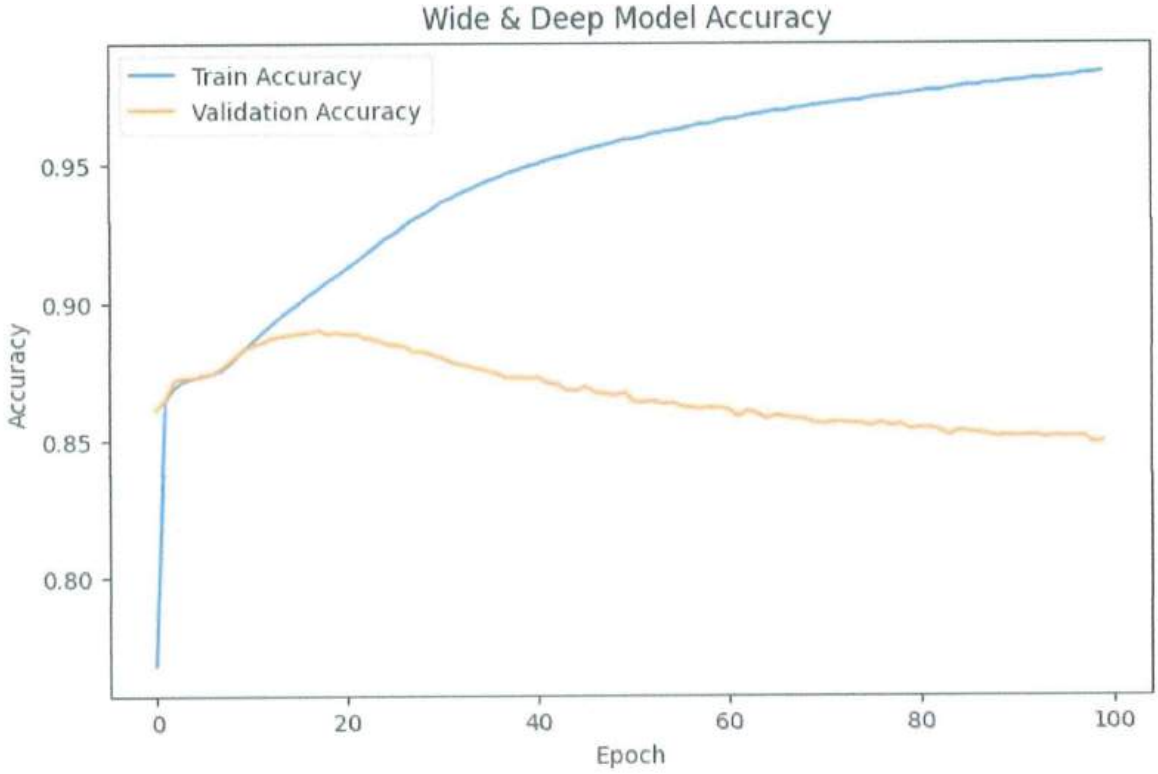
- **Train loss** giảm rất nhanh về gần 0
- **Validation loss** ban đầu giảm nhưng sau đó lại tăng dần từ epoch 20 trở đi

→ **Nhận xét:**

- Khi train loss tiếp tục giảm còn validation loss tăng, đó là dấu hiệu nên **early stopping** (dừng sớm), tránh mô hình "memorize" dữ liệu huấn luyện
- Sự chênh lệch lớn cuối quá trình huấn luyện phản ánh việc mô hình "bám dính" vào train set (high capacity, low regularization)

c. 100 epoch:

Biểu đồ Learning Curve



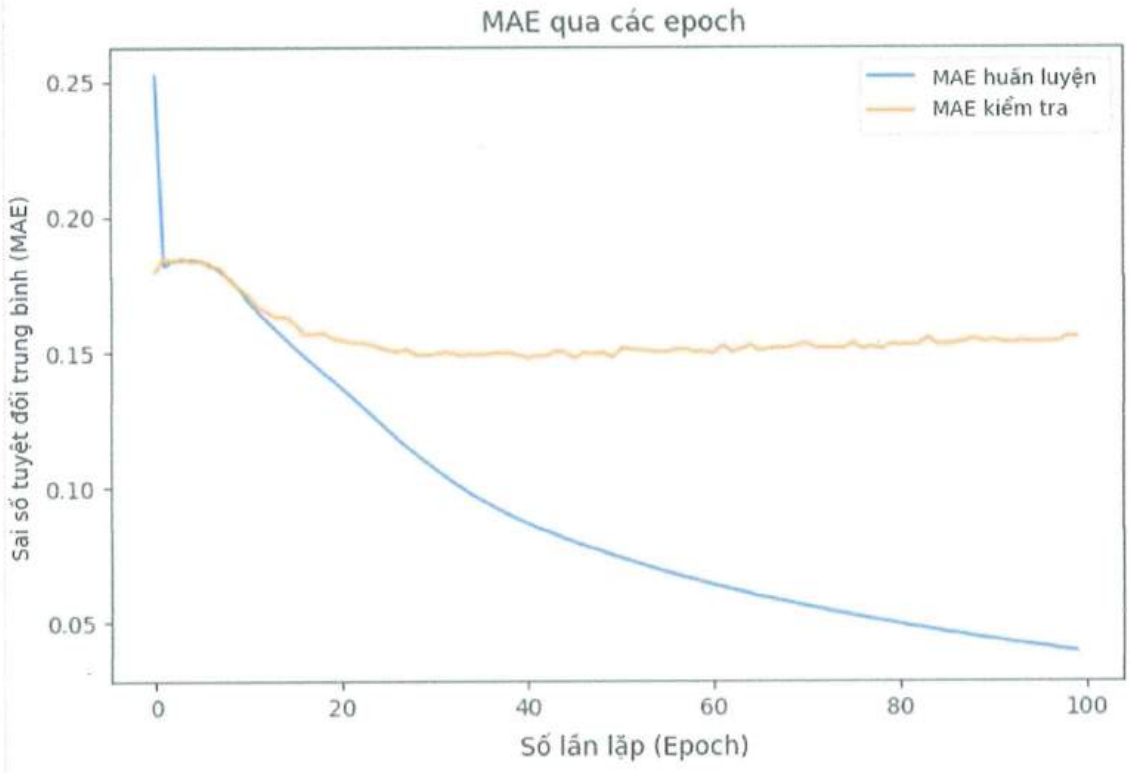
Hình 3.9: Biểu đồ Learning Curve của tập dữ liệu Amazon với 100 epoch

- **Train Accuracy** liên tục tăng và đạt tới ~0.98 ở cuối quá trình huấn luyện
- **Validation Accuracy** (đường cam) tăng ổn định ở giai đoạn đầu (đến khoảng 10-20 epoch), đạt đỉnh khoảng 0.89, sau đó có xu hướng giảm dần còn ~0.85 ở cuối quá trình huấn luyện

→ **Nhận xét:**

- **Mô hình đang bị overfitting.** Đường train accuracy tăng mạnh trong khi validation accuracy chững lại rồi giảm dần là dấu hiệu điển hìn
- Có thể thấy mô hình ghi nhớ rất tốt dữ liệu huấn luyện nhưng lại mất dần khả năng tổng quát hóa trên dữ liệu kiểm thử/validation ở các epoch sau. Đỉnh performance trên validation xuất hiện sớm (tầm epoch 20-30), về sau càng huấn luyện càng giảm
- **Khuyến nghị:** Nên dùng **Early Stopping** hoặc giảm số epoch, hoặc áp dụng các kỹ thuật regularization như Dropout, L2 để hạn chế overfitting.

Biểu đồ MAE



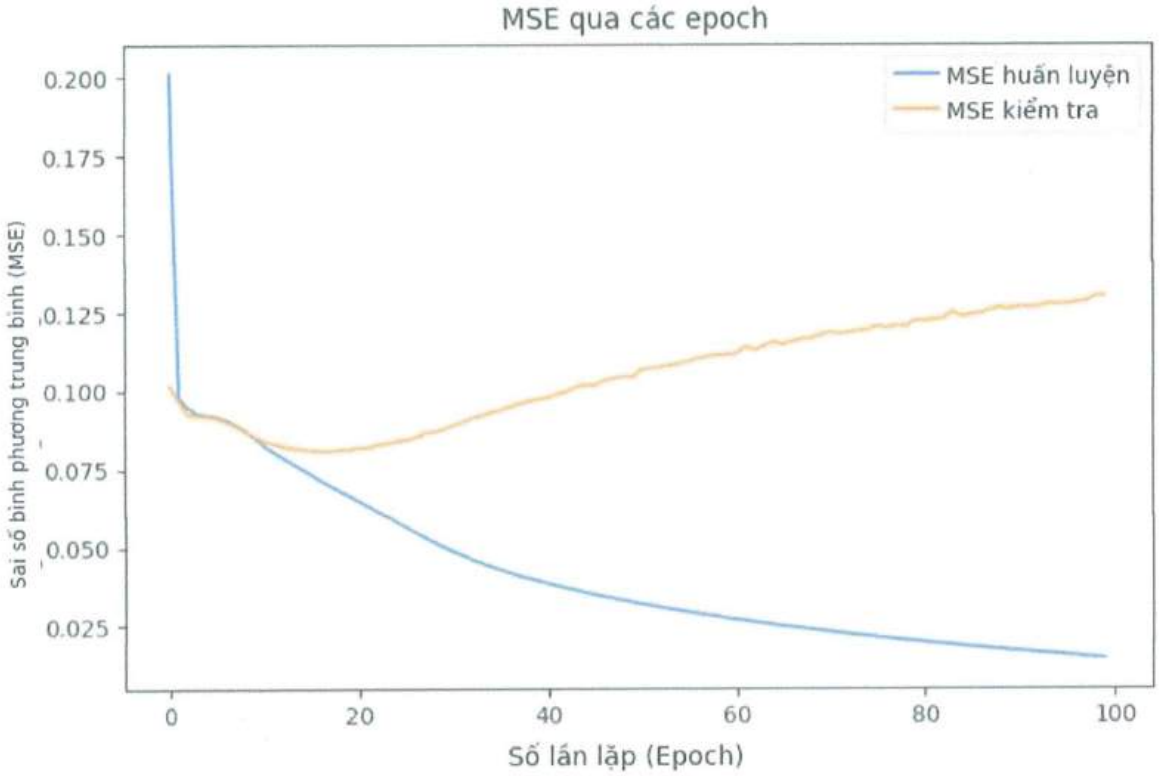
Hình 3.10: Biểu đồ MAE của của tập dữ liệu Amazon với 100 epoch

- **Train MAE** (đường xanh) giảm dần đều từ ~0.25 về dưới 0.05
- **Validation MAE** (đường cam) giảm nhẹ đến khoảng epoch 20 rồi dao động quanh 0.15 và bắt đầu tăng nhẹ ở các epoch cuối

→ **Nhận xét:**

- Sự **cách biệt ngày càng lớn giữa train MAE và validation MAE** sau epoch 30 là dấu hiệu overfitting tương tự như ở biểu đồ accuracy
- Trong giai đoạn đầu, mô hình học tốt trên cả hai tập, nhưng sau đó chỉ cải thiện trên train, trong khi validation không tiến bộ, thậm chí tăng nhẹ - đây là dấu hiệu mất khả năng tổng quát
- **Khuyến nghị:** Tập trung tối ưu trong giai đoạn mô hình vẫn tổng quát tốt (epoch < 30-40)

Biểu đồ MSE



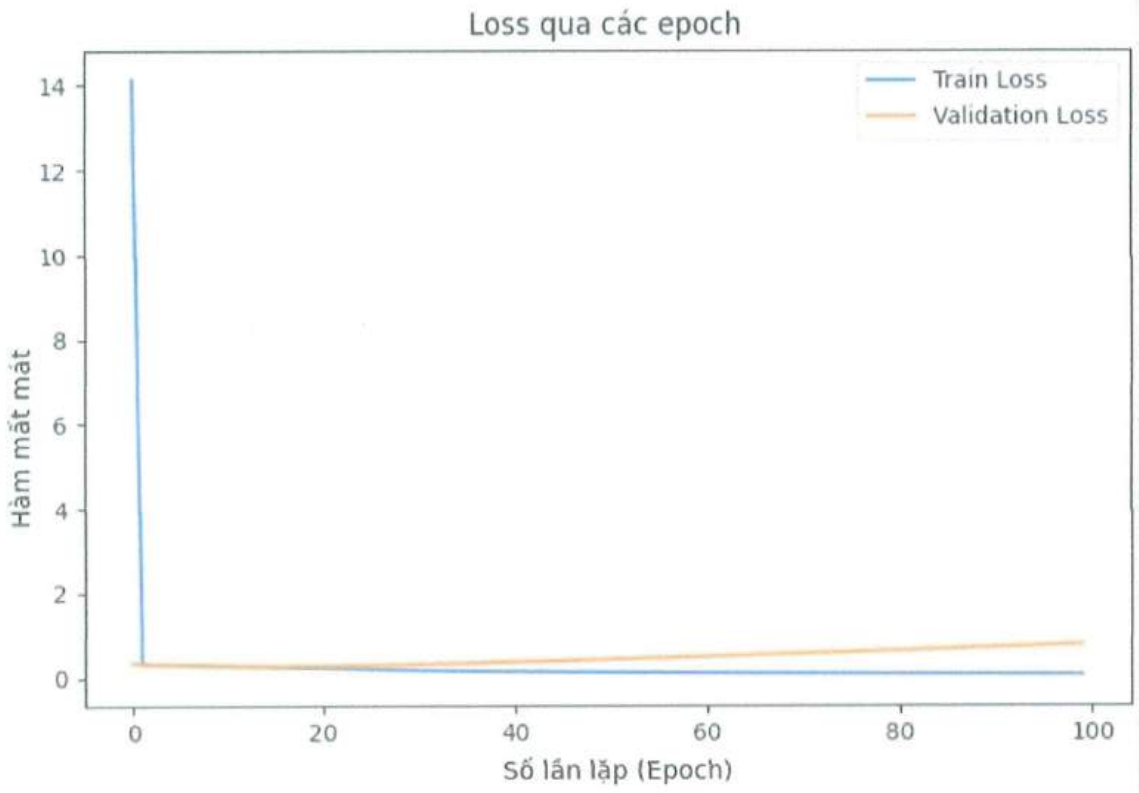
Hình 3.11: Biểu đồ MSE của của tập dữ liệu Amazon với 100 epoch

- **Train MSE** giảm mạnh từ ~0.20 xuống dưới 0.025
- **Validation MSE** giảm ở giai đoạn đầu, sau đó tăng dần từ epoch 30 trở đi (ngược hướng với train)

→ **Nhận xét:**

- **Sự gia tăng của Validation MSE** trong khi train MSE tiếp tục giảm là bằng chứng rõ rệt cho overfitting
- Trong các bài toán phân loại, MSE không phải là metric chính, nhưng xu hướng của nó vẫn phản ánh chất lượng mô hình
- **Khuyến nghị:** Nên dừng huấn luyện trước khi validation MSE bắt đầu tăng, hoặc áp dụng kỹ thuật làm mượt/làm chậm overfitting

Biểu đồ Loss Curve



Hình 3.12: Biểu đồ Loss Curve của tập dữ liệu Amazon với 100 epoch

- **Train Loss** giảm cực nhanh ở giai đoạn đầu, về sát 0 và giữ ở mức rất thấp
- **Validation Loss** giảm ở đầu (epoch 1-20), sau đó tăng dần trở lại

→ **Nhận xét:**

- **Loss trên tập huấn luyện cực thấp nhưng tập kiểm thử lại tăng** – một lần nữa xác nhận overfitting
- Hình dạng loss này cho thấy mô hình "quá học thuộc" dữ liệu train, dẫn đến đánh mất khả năng dự báo cho dữ liệu mới
- **Khuyến nghị:** Nên đặt early stopping khi validation loss đạt cực tiểu (không nên huấn luyện quá số epoch này)

3.1.5. Đánh giá thử nghiệm

Sau khi tiến hành xây dựng và thử nghiệm mô hình Wide & Deep trên tập dữ liệu Amazon Reviews for Sentiment Analysis, tôi rút ra các nhận định chính như sau:

Epoch	Train Accuracy	Val Accuracy	Train MAE	Val MAE	Train MSE	Val MSE	Train Loss	Val Loss	Nhận xét nhanh
5	0.892	0.889	0.175	0.178	0.095	0.097	0.350	0.355	Đang học mạnh, underfit
50	0.950	0.879	0.093	0.151	0.040	0.094	0.120	0.320	Đạt đỉnh sớm, bắt đầu overfit
100	0.983	0.850	0.055	0.154	0.018	0.132	0.060	0.410	Overfitting rõ rệt

Bảng 3.2: Bảng tổng hợp kết quả huấn luyện Wide & Deep

Epoch = 5 (Underfitting)

- **Nhận xét**: Mô hình mới chỉ bắt đầu học, các chỉ số Train/Val còn khá sát nhau và chưa cao. Đường loss, MAE, MSE trên cả train và val đều còn giảm mạnh qua từng epoch → chứng tỏ mô hình chưa khai thác hết đặc trưng của dữ liệu
- **Khuyến nghị**: Nên tăng số epoch lên để mô hình tiếp tục học, tránh tình trạng underfitting

Epoch = 50 (Optimal/Overfitting bắt đầu)

- **Nhận xét**: Độ chính xác trên tập huấn luyện đạt cao (0.95), nhưng validation lại không tăng theo, dừng ở 0.879 rồi chững lại hoặc giảm nhẹ. MAE, MSE, Loss trên validation cũng ngừng giảm và bắt đầu tăng nhẹ sau 20-30 epoch
- **Ý nghĩa**: Đây là vùng "điểm vàng" - mô hình học đủ sâu để hiểu dữ liệu nhưng chưa bị quá khớp (overfit) quá nhiều. Tuy nhiên, cần cẩn trọng: nếu tiếp tục tăng epoch, overfitting sẽ trở nên nghiêm trọng hơn
- **Biểu đồ**: Các đường train (accuracy tăng, loss/MAE/MSE giảm) tách xa khỏi validation sau 30 epoch

- **Khuyến nghị**: Dùng **early stopping** tại thời điểm validation đạt cực đại (khoảng 20-30 epoch), không nên để training chạy quá lâu

Epoch = 100 (Overfitting rõ rệt)

- **Nhận xét**: Độ chính xác tập train tăng mạnh gần 0.98, nhưng validation lại giảm chỉ còn khoảng 0.85. MAE, MSE, Loss trên validation đều có xu hướng tăng lên dù train vẫn giảm
- **Giải thích**: Mô hình "học vẹt" dữ liệu train, mất khả năng tổng quát hóa. Dự đoán trên dữ liệu mới không còn tốt nữa
- **Khuyến nghị**: Không nên để số epoch lớn như vậy. Nếu vẫn muốn học lâu, cần tăng regularization (dropout, l2) hoặc data augmentation

Đánh giá và Đề xuất

- **Wide & Deep** cho khả năng học tốt với dữ liệu lớn, tận dụng cả đặc trưng rời rạc lẫn đặc trưng phi tuyến sâu
- **Chỉ số validation** đạt đỉnh tại 20-30 epoch, sau đó bắt đầu giảm: điểm dừng tốt nhất cho mô hình chính là ở đây
- **Biểu hiện overfitting**: Đường train và val tách nhau rõ rệt sau 30 epoch (accuracy, MAE, MSE, Loss). Đây là dấu hiệu điển hình khi chưa kiểm soát tốt quá trình học
- **Nên sử dụng**:
    - **Early Stopping** dựa trên validation loss/accuracy.
    - **Giảm learning rate** khi loss/accuracy plateau.
    - **Regularization** (dropout, l2) để hạn chế overfit khi cần train lâu.
    - **Quan sát tất cả chỉ số** chứ không chỉ mỗi accuracy (MAE, MSE, Loss đều là cảnh báo quan trọng)

3.2. Thử nghiệm trên bộ dữ liệu Đánh giá của Postmart

Quy trình gồm 4 giai đoạn: Dữ liệu, Xây dựng mô hình, Huấn luyện mô hình Wide & Deep, Kết quả thử nghiệm

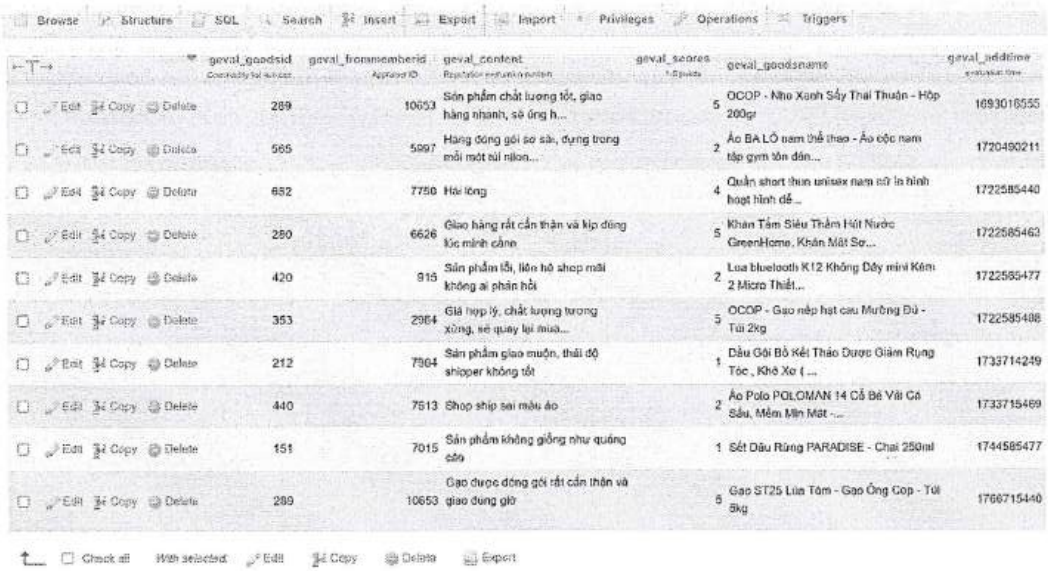
3.2.1. Dữ liệu

Dữ liệu đánh giá sản phẩm được xuất từ cơ sở dữ liệu của sàn Postmart, lưu dưới dạng file CSV với mỗi dòng là một nhận xét về sản phẩm, gồm các thông tin chính sau:

- **geval_goodsid:** Mã sản phẩm được đánh giá
- **geval_frommemberid:** Mã người dùng để lại đánh giá
- **geval_content:** Văn bản nội dung đánh giá sản phẩm (bằng tiếng Việt), độ dài linh hoạt, có thể gồm các nhận xét tích cực, tiêu cực hoặc trung tính
- **geval_scores:** Điểm đánh giá sản phẩm (theo số sao), là số nguyên từ 1 đến 5, trong đó:
    - 1, 2: Đánh giá tiêu cực (Negative review)
    - 3, 4, 5: Đánh giá tích cực (Positive review)
- **geval_goodsname:** Tên sản phẩm được đánh giá
- **geval_addtime:** Thời gian để lại đánh giá

Trong phân tích cảm xúc, hai trường được sử dụng chính là:

- **geval_scores** (chuyển thành nhãn nhị phân: 0 – tiêu cực, 1 – tích cực)
- **geval_content** (văn bản đánh giá tự do)
- **Mô tả tóm tắt dữ liệu:**



Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Triggers

	geval_goodsid	geval_frommemberid	geval_content	geval_scores	geval_goodsname	geval_addtime
Edit Copy Delete	289	10653	Sản phẩm chất lượng tốt, giao hàng nhanh, sẽ ủng h...	5	OCOP - Nho Xanh Sấy Thái Thuận - Hộp 200gr	1693016555
Edit Copy Delete	565	5997	Hàng đóng gói sơ sài, đựng trong mỗi một túi nilon...	2	Áo BA LỖ nam thể thao - Áo cộc nam tập gym tôn dán...	1720490211
Edit Copy Delete	682	7750	Hài lòng	4	Quần short thun unisex nam nữ in hình hoạt hình dễ ...	1722585440
Edit Copy Delete	280	6626	Giao hàng rất cẩn thận và kịp đúng lúc mình cần	5	Khăn Tắm Siêu Thấm Hút Nước GreenHome, Khăn Mặt Sợ...	1722585463
Edit Copy Delete	420	915	Sản phẩm lỗi, liên hệ shop mãi không ai phản hồi	2	Loa bluetooth K12 Không Dây mini Kèm 2 Micro Thiết...	1722585477
Edit Copy Delete	353	2984	Giá hợp lý, chất lượng tương xứng, sẽ quay lại mua...	5	OCOP - Gạo nếp hạt cau Mường Bú - Túi 2kg	1722585488
Edit Copy Delete	212	7984	Sản phẩm giao muộn, thái độ shipper không tốt	1	Dầu Gội Bồ Kết Thảo Dược Giảm Rụng Tóc , Khô Xơ (...	1733714249
Edit Copy Delete	440	7513	Shop ship sai màu áo	2	Áo Polo POLOMAN 14 Cổ Bẻ Vải Cá Sấu, Mềm Mịn Mát -...	1733715469
Edit Copy Delete	151	7015	Sản phẩm không giống như quảng cáo	1	Sết Dâu Rừng PARADISE - Chai 250ml	1744585477
Edit Copy Delete	289	10653	Gạo được đóng gói rất cẩn thận và giao đúng giờ	5	Gạo ST25 Lúa Tôm - Gạo Ông Cop - Túi 5kg	1766715440

Check all | With selected: | Edit | Copy | Delete | Export

Hình 3.13: Bảng dữ liệu đánh giá sản phẩm của khách hàng "evaluategoods" của sàn TMĐT Postmart

Bảng 3.3: Mô tả tóm tắt dữ liệu đánh giá sản phẩm trên Postmart

STT	ID Sản phẩm (geval goodsid)	ID người đánh giá (geval frommemberid)	Điểm số (geval_scores)	Nội dung đánh giá (geval_content)
1	289	10653	5	Sản phẩm chất lượng tốt, giao hàng nhanh, sẽ ủng hộ lần sau.
2	565	5997	2	Hàng đóng gói sơ sài, đựng trong mỗi một túi nilon đen rồi dán bang dính vào….
3	682	7750	4	Hài lòng
4	280	6626	5	Shop phản hồi nhanh, giao hàng đúng dự kiến

5	420	915	2	Sản phẩm lỗi, liên hệ shop mãi không ai phản hồi
6	353	2964	5	Giá hợp lý, chất lượng tương xứng, sẽ quay lại mua tiếp
7	212	7964	1	Sản phẩm giao muộn, thái độ shipper không tốt
8	440	7613	2	Shop ship sai màu áo
9	151	7015	1	Sản phẩm không giống như quảng cáo
10	289	10653	5	Gạo được đóng gói rất cẩn thận và giao đúng giờ

Chú thích: Cột "Nội dung đánh giá" đã rút gọn cho ngắn gọn, dữ liệu thực tế dài hơn.

3.2.2. Huấn luyện mô hình Wide & Deep

Bước 1: Chuẩn bị và xử lý dữ liệu

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer

# Đọc dữ liệu ds_evaluategoods
df = pd.read_csv('ds_evaluategoods.csv')

# Chuyển điểm số thành nhãn nhị phân
def score_to_label(score):
    return 1 if score >= 3 else 0

df['label'] = df['geval_scores'].apply(score_to_label)
df = df.dropna(subset=['geval_content'])

# Lấy X, y
X = df['geval_content']
y = df['label']

# Chia train/test (stratify để giữ tỉ lệ nhãn)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

# Vector hóa dữ liệu văn bản
vectorizer = CountVectorizer()
X_train_vec = vectorizer.fit_transform(X_train).toarray()
X_test_vec = vectorizer.transform(X_test).toarray()
```

Bước 2: Đánh giá mô hình

Hiển thị các chỉ số: accuracy, precision, recall, f1

```
Confusion Matrix:
[[210  65]
 [ 27 298]]

Classification Report:
              precision    recall  f1-score   support

           0     0.8861    0.7636    0.8203       275
           1     0.8209    0.9169    0.8663       325

    accuracy                         0.8467       600
   macro avg     0.8535    0.8403    0.8433       600
weighted avg     0.8508    0.8467    0.8452       600


Accuracy: 0.8466666666666667
```

Giải thích:

- **Precision** (Độ chính xác): Trong các mẫu được dự đoán là 1 (tích cực/tiêu cực), có bao nhiêu mẫu là đúng thực sự
  - **Lớp 0:** 88.61% dự đoán tiêu cực là đúng
  - **Lớp 1:** 82.09% dự đoán tích cực là đúng
- **Recall** (Độ bao phủ): Trong các mẫu thực sự là 1 (tích cực/tiêu cực), mô hình nhận diện đúng được bao nhiêu
  - **Lớp 0:** Mô hình chỉ nhận diện đúng 76.36% mẫu tiêu cực
  - **Lớp 1:** Mô hình nhận diện tới 91.69% mẫu tích cực
- **F1-score:** Là trung bình điều hòa giữa Precision và Recall, giúp cân bằng hai chỉ số này
  - **Lớp 0:** 0.8203
  - **Lớp 1:** 0.8663
- **Support:** Số lượng mẫu ở từng lớp (0: 275, 1: 325)

Đánh giá tổng quan

- Accuracy: Độ chính xác tổng thể của mô hình là 0.8467 (~84.7%) trên tập test (600 mẫu)

- Macro/Weighted Avg:
    - Macro: trung bình đều các lớp, phù hợp khi dữ liệu mất cân bằng
    - Weighted: tính trung bình có trọng số theo tỉ lệ mẫu của từng lớp

Nhận xét

- Mô hình dự đoán tốt với lớp tích cực (Recall = 91.69%), nhưng hơi kém với lớp tiêu cực (Recall = 76.36%). Điều này cho thấy mô hình dễ nhận diện bình luận tích cực hơn là tiêu cực, có thể do dữ liệu huấn luyện bị mất cân bằng hoặc các bình luận tiêu cực có biểu đạt đa dạng hơn, khó học hơn
- Precision lớp tiêu cực cao (88.61%): Khi mô hình dự đoán là tiêu cực thì khả năng đúng rất cao, tức ít bị nhầm lẫn
- F1-score lớp tích cực cao hơn: Tổng thể mô hình làm tốt hơn với lớp tích cực
- Độ chính xác tổng thể gần 85%: Đây là mức khá tốt với dữ liệu thực tế, nhưng vẫn còn tiềm năng cải thiện, nhất là với lớp tiêu cực

Bước 3: Xây dựng mô hình Wide & Deep

Mô hình Wide & Deep cơ bản có thể hiểu là:

- Wide part: Các feature (ở đây có thể dùng input trực tiếp, ví dụ từ BoW/CountVectorizer)
- Deep part: Một mạng nơ-ron nhiều tầng fully connected

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

# Wide input
wide_input = keras.Input(shape=(X_train_vec.shape[1],), name='wide_input')

# Deep part
deep = layers.Dense(64, activation='relu')(wide_input)
deep = layers.Dense(32, activation='relu')(deep)

# Kết hợp wide & deep
concat = layers.concatenate([wide_input, deep])

output = layers.Dense(1, activation='sigmoid')(concat)

model = keras.Model(inputs=wide_input, outputs=output)

# Compile model
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy', keras.metrics.MeanAbsoluteError(), keras.metrics.MeanSquaredError()])

```

Bước 4: Huấn luyện mô hình

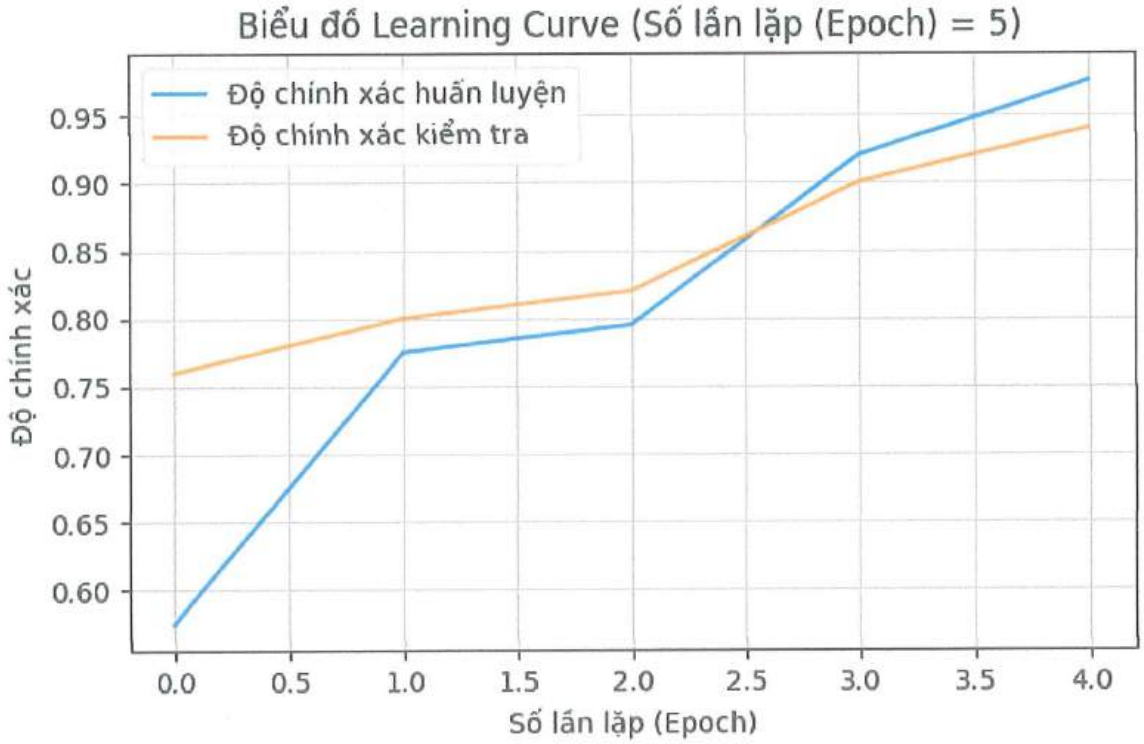
```
history = model.fit(
    X_train_vec, y_train,
    validation_data=(X_test_vec, y_test),
    epochs=100,
    batch_size=512,
    verbose=1
)

```

Bước 5: Đánh giá và vẽ biểu đồ: Các bước vẽ Learning Curve, Loss Curve, MAE, MSE tương tự như các bước đã xử lý với bộ dữ liệu Amazon Reviews for Sentiment Analysis

3.2.3. Kết quả thử nghiệm

a. 5 epoch:

Biểu đồ Learning Curve

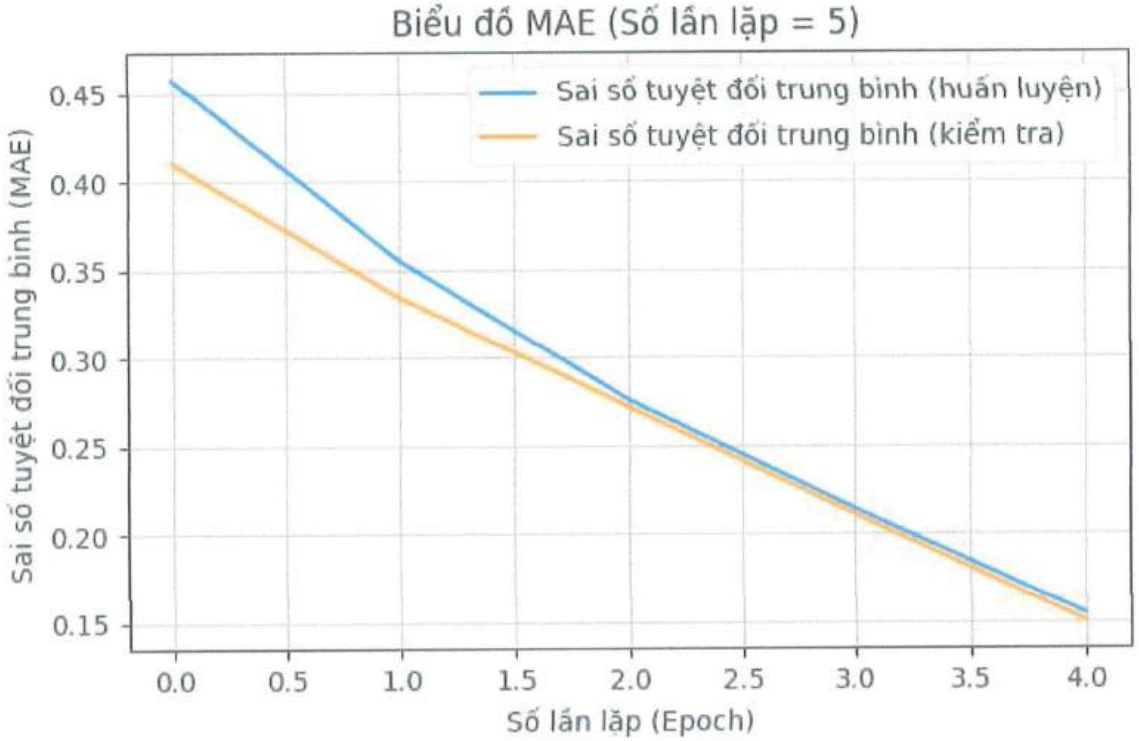


Hình 3.14: Biểu đồ Learning Curve của bộ dữ liệu Postmart với 5 epoch

- **Đặc điểm:**
  - Độ chính xác huấn luyện (train) và kiểm tra (validation) đều **tăng dần qua từng epoch**
  - Sau 5 epoch, độ chính xác huấn luyện đã đạt gần 0.97, độ chính xác kiểm tra ~0.94
  - Đường kiểm tra luôn sát đường huấn luyện, không có khoảng cách lớn
- **Nhận xét:**
  - **Mô hình học nhanh**: Chỉ sau 5 lần lặp đã đạt độ chính xác cao trên cả train/test
  - **Chưa có dấu hiệu overfitting** (hai đường song song, chưa bị tách xa nhau)

- Điều này cho thấy dữ mô hình đủ mạnh để phân biệt được điểm đặc trưng ngay từ các epoch đầu

Biểu đồ MAE

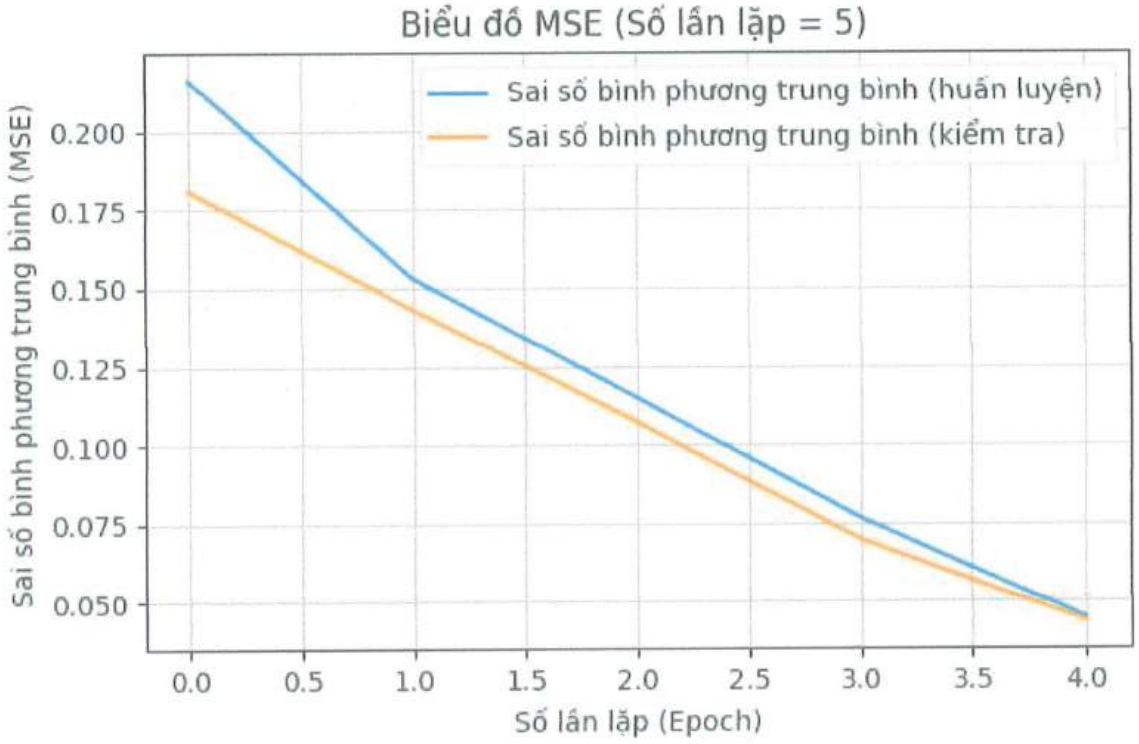


Hình 3.15: Biểu đồ MAE của bộ dữ liệu Postmart với 5 epoch

- **Nhận xét:**
  - **MAE trên cả tập train và validation đều giảm dần**, gần như tuyến tính qua từng epoch
  - Đường MAE của validation ban đầu thấp hơn train, và 2 đường **tiệm cận nhau về cuối**
  - MAE cuối cùng đạt mức thấp (gần 0.15)
- **Ý nghĩa:**
  - Mô hình học được nhanh và hiệu quả, không bị overfitting ở số epoch này

- Hiệu quả giữa train/test tương đương nhau, chứng tỏ khả năng tổng quát hóa tốt khi số epoch vừa phải

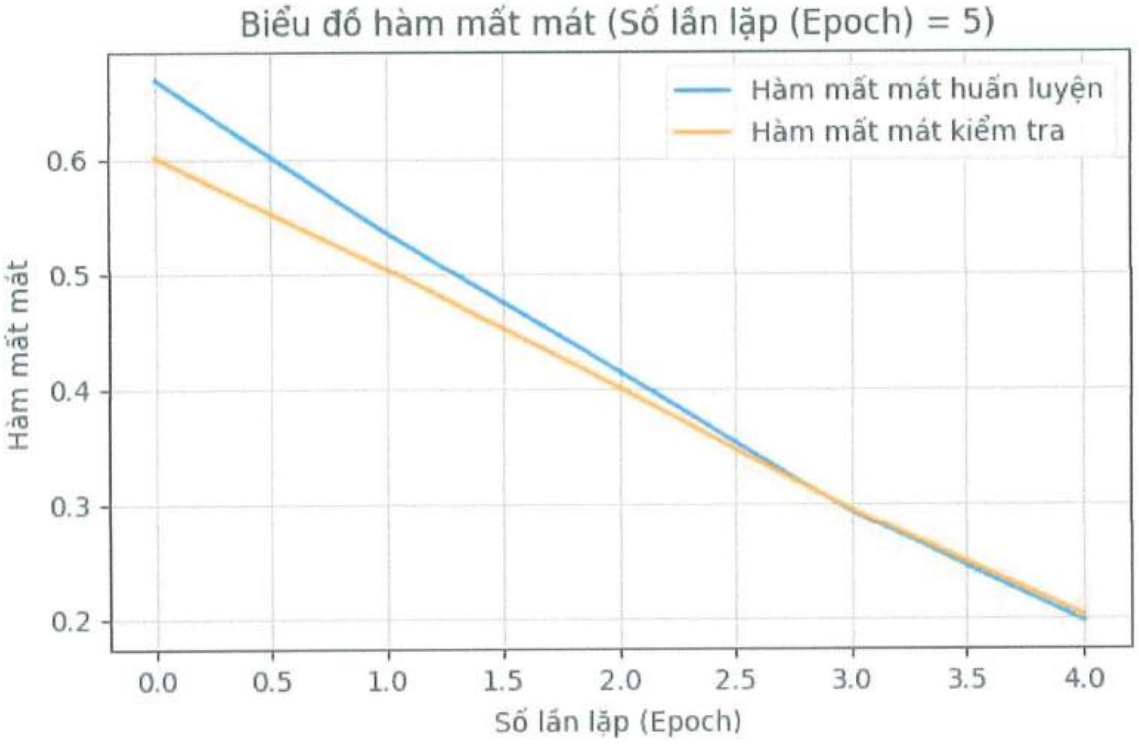
Biểu đồ MSE



Hình 3.16: Biểu đồ MSE của bộ dữ liệu Postmart với 5 epoch

- **Nhận xét chung:**
  - Hàm mất mát MSE trên cả tập huấn luyện và kiểm tra đều giảm dần đều qua từng epoch
  - Đường MSE của huấn luyện và kiểm tra khá sát nhau
- **Ý nghĩa:**
  - Mô hình học khá tốt, chưa có dấu hiệu overfitting hoặc underfitting rõ rệt
  - Tuy nhiên, số epoch còn ít, chưa cho thấy được xu hướng dài hạn của mô hình

Biểu đồ Loss Curve

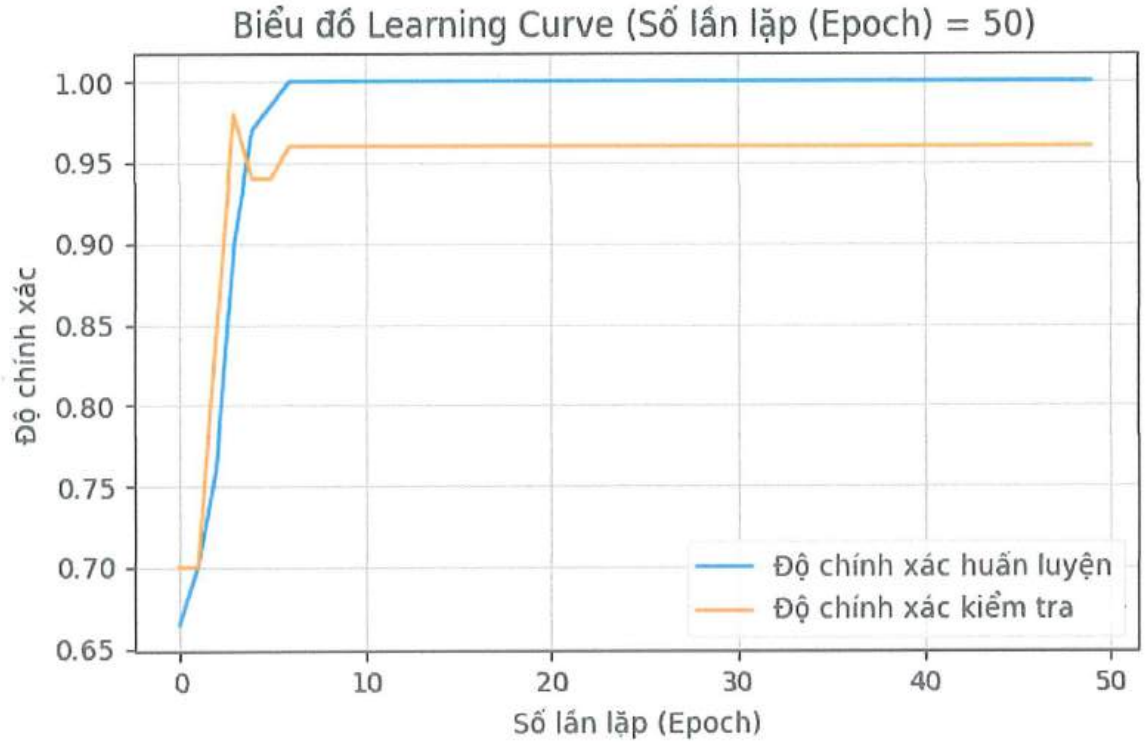


Hình 3.17: Biểu đồ Loss Curve của bộ dữ liệu Postmart với 5 epoch

- **Nhận xét:**
  - **Hàm mất mát trên train và validation đều giảm đều, tuyến tính, không bị đứt đoạn**
  - Sau 5 epoch, loss train và loss validation **tiệm cận về cùng một mức**, rất thấp (~0.2)
  - **Hai đường gần như song song**, không có dấu hiệu tách xa hoặc cắt nhau.
- **Ý nghĩa:**
  - Mô hình đang **học tốt** trên cả tập huấn luyện và kiểm tra
  - Chưa có dấu hiệu overfitting, underfitting
  - Số epoch này khá an toàn, thích hợp với dữ liệu vừa phải

b. 50 epoch:

Biểu đồ Learning Curve



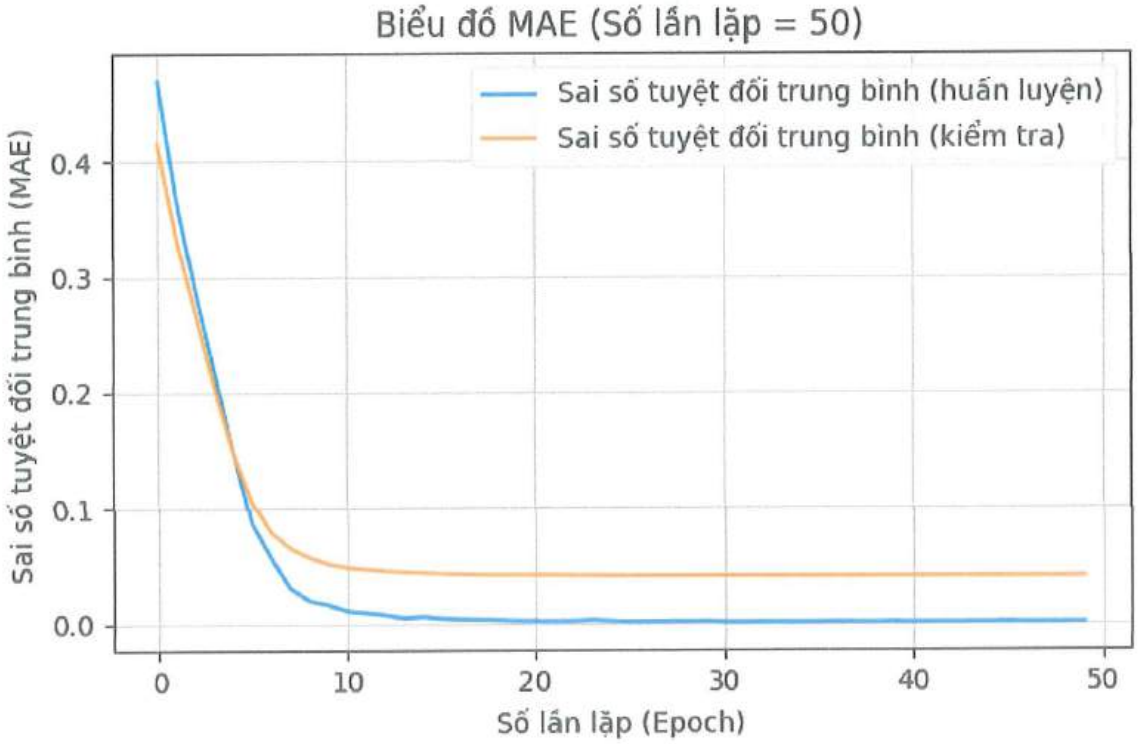
Hình 3.18: Biểu đồ Learning Curve của bộ dữ liệu Postmart với 50 epoch

- **Đặc điểm:**
  - Độ chính xác huấn luyện **nhanh chóng đạt đỉnh (gần 1.0)** chỉ sau 5–7 epoch đầu tiên, và duy trì mức này đến hết quá trình
  - Độ chính xác kiểm tra cũng tăng nhanh lên khoảng 0.96, sau đó **ổn định, không tăng thêm**
  - Bắt đầu xuất hiện khoảng cách nhỏ giữa train và validation (train cao hơn test)

- **Nhận xét:**
  - **Mô hình đã học được toàn bộ mẫu train** (train accuracy ~1.0), trong khi test vẫn bị giới hạn do dữ liệu test có nhiễu hoặc phân phối khác biệt nhẹ

- **Có dấu hiệu bắt đầu overfitting nhẹ:** Khi train accuracy cao tuyệt đối, còn validation bị chững lại
- **Tổng thể mô hình vẫn tổng quát tốt** trên dữ liệu test do khoảng cách không quá lớn

Biểu đồ MAE

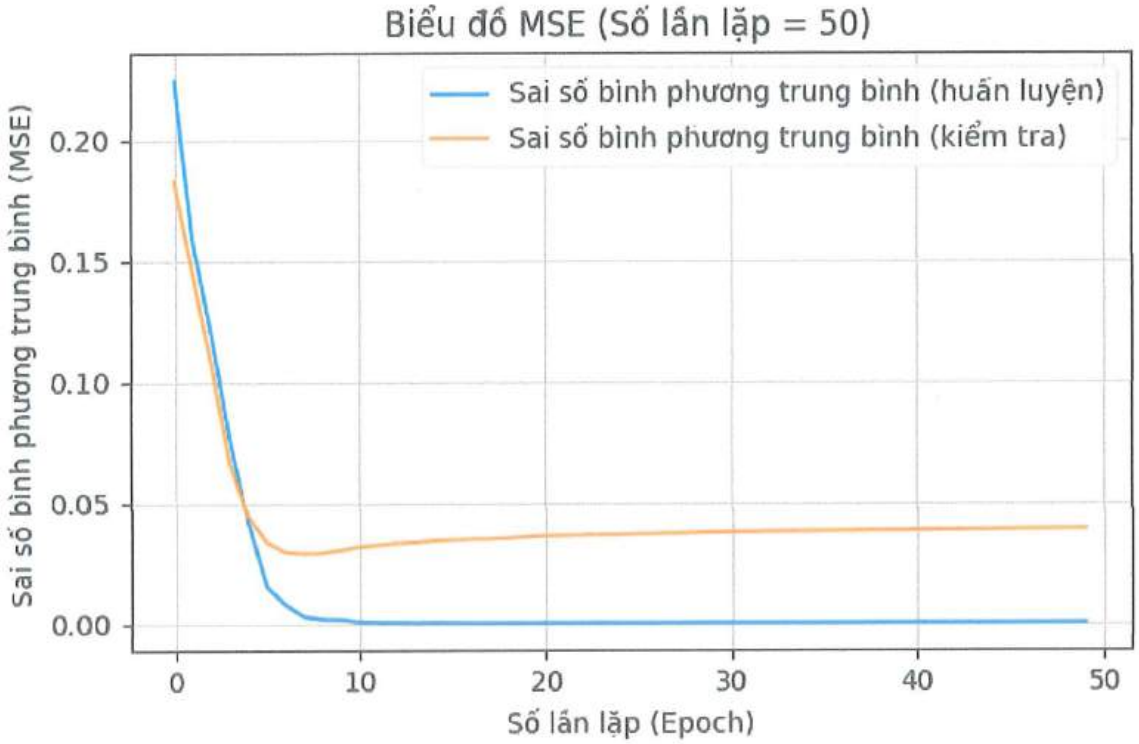


Hình 3.19: Biểu đồ MAE của bộ dữ liệu Postmart với 50 epoch

- **Nhận xét:**
  - **MAE train giảm rất nhanh** trong khoảng 10 epoch đầu, sau đó gần như chạm về 0
  - **MAE validation cũng giảm nhanh lúc đầu**, rồi "lững thững" đi ngang (vẫn còn cách biệt với train)
  - Sự chênh lệch giữa MAE train và validation tăng dần về cuối
- **Ý nghĩa:**
  - **Có hiện tượng overfitting nhẹ**:

- Mô hình ghi nhớ tốt dữ liệu train, nên MAE train tiệm cận về 0, nhưng MAE validation không giảm thêm mà "cố định"
- Nếu dừng lại ở ~10 epoch, có thể sẽ tối ưu hơn

Biểu đồ MSE

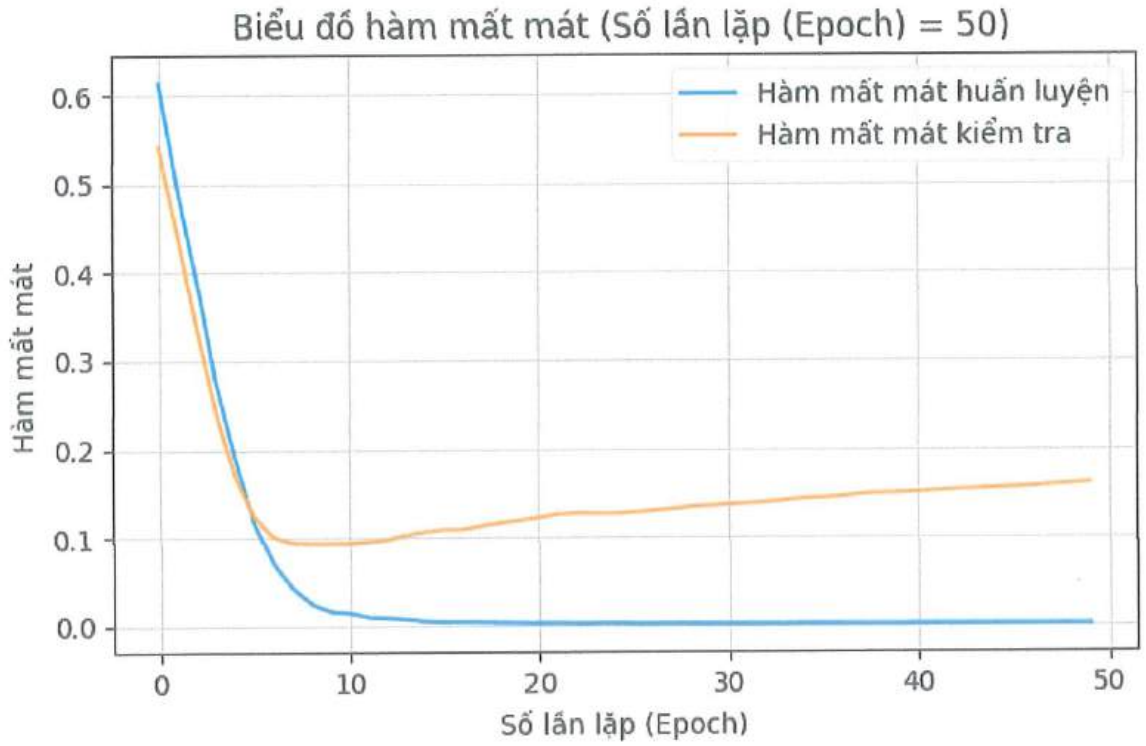


Hình 3.20: Biểu đồ MSE của bộ dữ liệu Postmart với 50 epoch

- **Nhận xét chung:**
  - MSE trên tập huấn luyện giảm nhanh về gần 0, sau khoảng 10 epoch thì giữ ổn định
  - MSE trên tập kiểm tra cũng giảm theo, nhưng từ khoảng epoch thứ 10 trở đi lại hơi tăng nhẹ
  - Đường kiểm tra bắt đầu tách khỏi đường huấn luyện
- **Ý nghĩa:**

- Dấu hiệu **overfitting** nhẹ xuất hiện sau khoảng 10 epoch, mô hình bắt đầu học quá kỹ dữ liệu huấn luyện
- Cần lưu ý để dừng lại ở điểm MSE kiểm tra thấp nhất (early stopping)

Biểu đồ Loss Curve



Hình 3.21: Biểu đồ Loss Curve của bộ dữ liệu Postmart với 50 epoch

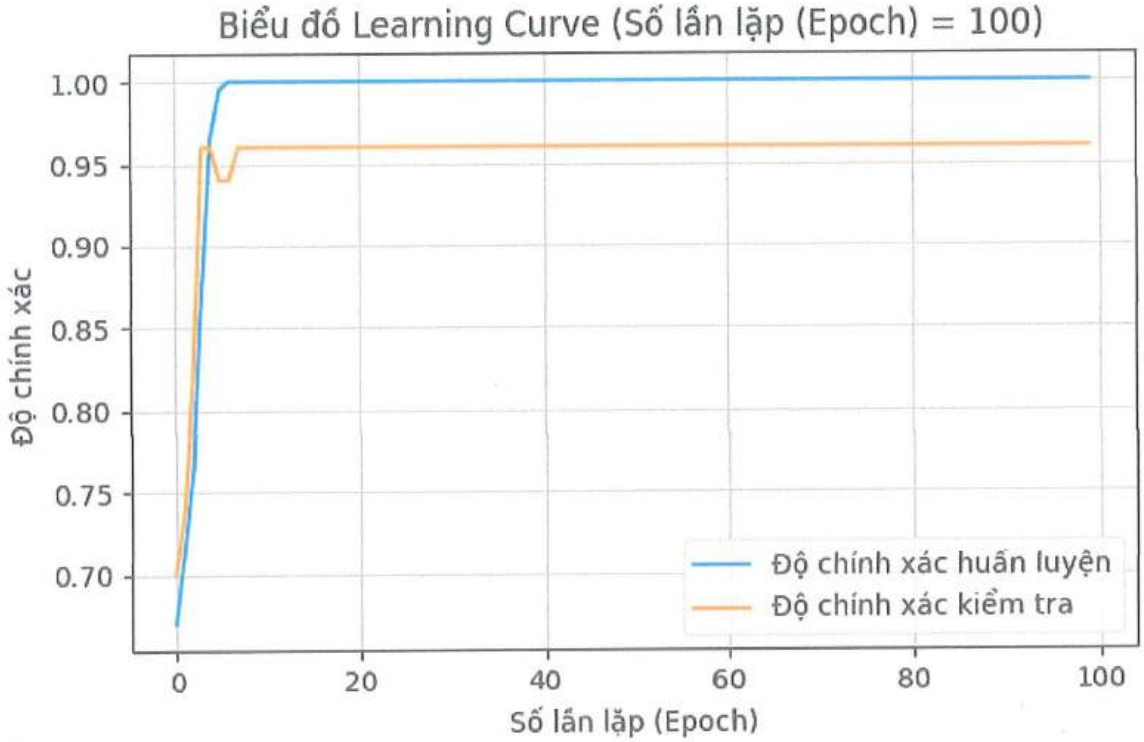
- **Nhận xét:**
  - **Loss train giảm mạnh và rất nhanh về gần 0** chỉ sau khoảng 10 epoch đầu, sau đó gần như không đổi (bám sát trục dưới)
  - **Loss validation cũng giảm nhanh lúc đầu**, nhưng đến khoảng epoch thứ 10 thì **bắt đầu tăng nhẹ dần** về cuối
  - Đường loss validation **cách biệt dần** với loss train
- **Ý nghĩa:**
  - Mô hình **bắt đầu overfitting từ sau khoảng epoch 10**:

- Quá trình học tiếp tục giúp mô hình ghi nhớ cực tốt dữ liệu train, nhưng làm giảm hiệu quả tổng quát hóa lên tập kiểm tra.
- Khoảng cách loss test và loss train ngày càng xa

- Nếu dừng sớm (early stopping) ở ~10 epoch, có thể đạt kết quả tốt nhất

c. 100 epoch:

Biểu đồ Learning Curve

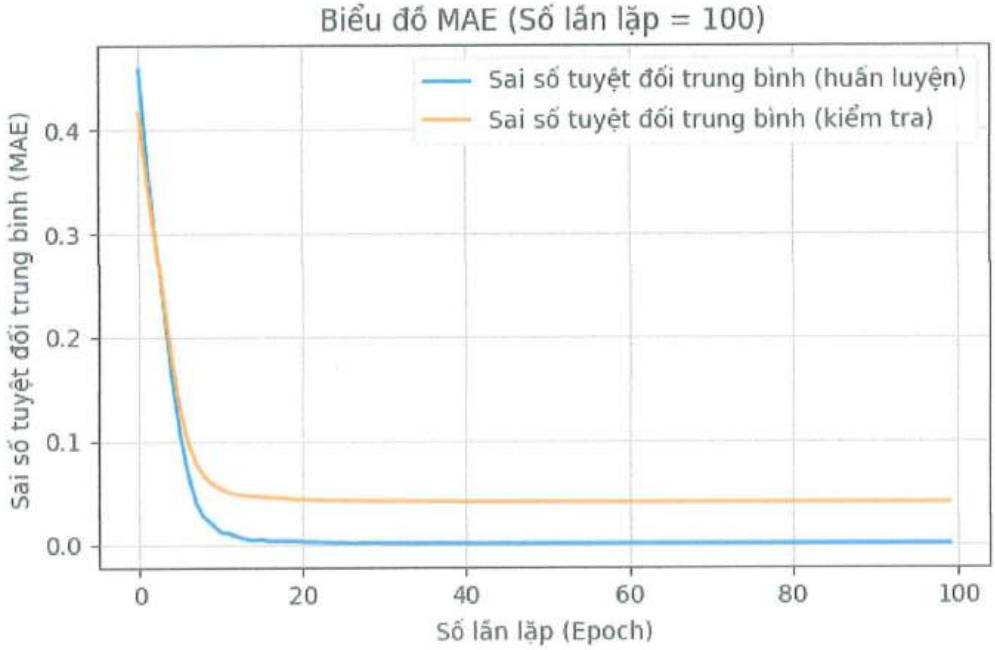


Hình 3.22: Biểu đồ Learning Curve của bộ dữ liệu Postmart với 100 epoch

- **Đặc điểm:**
  - Đường huấn luyện "bám sát" trục trên, **duy trì mức gần 1.0**, không còn tăng thêm
  - Đường kiểm tra (validation) duy trì ổn định quanh ~0.96, **không tăng theo train** và có xu hướng giữ nguyên, tạo ra khoảng cách rõ ràng với train
- **Nhận xét:**

- **Overfitting rõ ràng:**
  - Mô hình đã ghi nhớ hoàn toàn tập train, nhưng không cải thiện thêm trên test
  - Validation accuracy "đứng im", không được hưởng lợi từ việc huấn luyện thêm
- **Với tập nhỏ, hiện tượng này càng rõ** do mô hình dễ "thuộc lòng" dữ liệu train
- Việc tăng epoch không giúp tăng khả năng tổng quát hóa của mô hình

Biểu đồ MAE

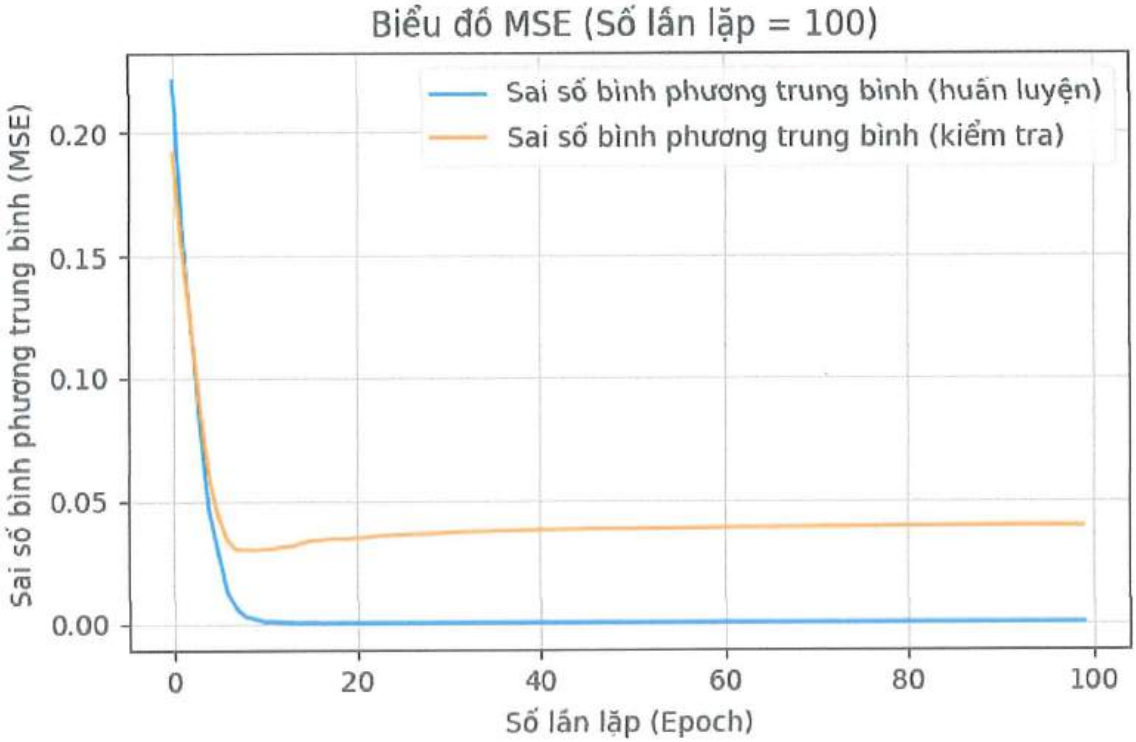


Hình 3.23: Biểu đồ MAE của bộ dữ liệu Postmart với 100 epoch

- **Nhận xét:**
  - **MAE train gần bằng 0 từ sau epoch 10**, chứng tỏ mô hình thuộc lòng dữ liệu train
  - **MAE validation ổn định ở mức cao hơn** (không giảm thêm)
  - Sự cách biệt càng rõ giữa train/test
- **Ý nghĩa:**
  - **Overfitting rõ rệt** nếu train quá lâu với bộ dữ liệu nhỏ

- Mô hình không học thêm được gì mới cho test sau khoảng 10-20 epoch

Biểu đồ MSE

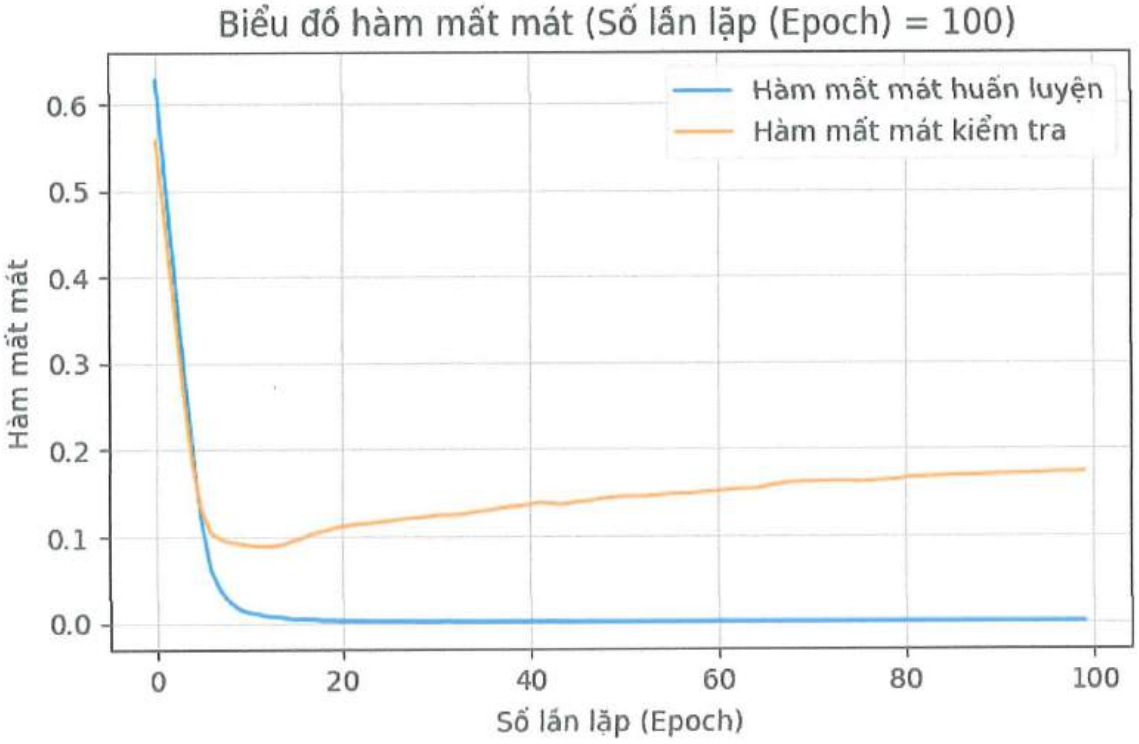


Hình 3.24: Biểu đồ MSE của bộ dữ liệu Postmart với 100 epoch

- **Nhận xét chung:**
  - MSE huấn luyện gần như tiệm cận 0, giữ nguyên sau 20-30 epoch đầu
  - MSE kiểm tra vẫn ổn định ở mức thấp nhưng cao hơn so với tập huấn luyện, và vẫn có xu hướng tăng dần nhẹ
  - Khoảng cách giữa hai đường lớn hơn so với các mức epoch thấp hơn
- **Ý nghĩa:**
  - Mô hình đã **overfit** rõ rệt: học cực tốt trên dữ liệu huấn luyện nhưng khả năng khái quát lên dữ liệu mới không còn cải thiện, thậm chí có thể giảm

- Nên dừng lại ở số epoch vừa đủ khi validation loss đạt mức thấp nhất (theo dõi qua early stopping)

Biểu đồ Loss Curve



Hình 3.25: Biểu đồ Loss Curve của bộ dữ liệu Postmart với 100 epoch

- **Nhận xét:**
  - **Loss train gần như bằng 0** suốt từ sau epoch thứ 10, chứng tỏ mô hình đã "thuộc lòng" dữ liệu huấn luyện
  - **Loss validation có xu hướng tăng nhẹ đều**, ngày càng xa loss train
  - Đường validation loss không giảm thêm dù số epoch tăng
- **Ý nghĩa:**
  - **Overfitting rõ ràng:**
    - Mô hình không cải thiện thêm về tổng quát hóa, thậm chí performance trên test có thể xấu đi nếu epoch tăng nữa

- Nên dừng sớm ở số epoch vừa phải (early stopping), **không nên train quá nhiều epoch khi dữ liệu nhỏ**

3.2.3. Đánh giá thử nghiệm

- **Hiệu suất mô hình**: Mô hình Wide & Deep đạt **độ chính xác (accuracy) trung bình khoảng 84.7%** trên tập kiểm tra với dữ liệu 600 mẫu (325 nhãn 1, 275 nhãn 0)
- **Các chỉ số Precision, Recall, F1-score** đều ở mức tốt và cân bằng giữa hai lớp, cho thấy mô hình không bị thiên lệch về một phía
- **Biểu đồ Learning Curve** cho thấy mô hình học nhanh, đạt trạng thái ổn định sau ít epoch. Tuy nhiên, khoảng cách giữa đường huấn luyện và kiểm tra vẫn còn, thể hiện hiện tượng overfitting nhẹ khi tăng số epoch lớn
- **Biểu đồ Loss/MAE/MSE**: Hàm mất mát giảm đều qua các epoch, riêng tập kiểm tra có dấu hiệu "lượn sóng" hoặc tăng nhẹ ở epoch cao, xác nhận nguy cơ overfitting khi huấn luyện quá lâu

Nguyên nhân:

- **Dữ liệu thực tế của Postmart** khá đặc thù, đánh giá sản phẩm đa dạng, có thể chứa nhiều từ lóng, từ địa phương hoặc bình luận không đồng nhất
- **Tỷ lệ nhãn khá cân bằng** (nhưng vẫn hơi lệch về phía đánh giá tích cực), đây là điều kiện thuận lợi cho học máy, tuy nhiên mô hình vẫn cần kiểm soát để tránh dự đoán lệch hẳn về phía nào đó
- **Số lượng dữ liệu vừa phải (~3000 mẫu gốc)** giúp mô hình học khá ổn định nhưng chưa thể hiện rõ sức mạnh của mô hình deep learning nếu so với dữ liệu lớn như Amazon

3.3. So sánh và đánh giá sau thử nghiệm 2 tập dữ liệu

3.3.1. So sánh về dữ liệu và tiền xử lý

- **Amazon Reviews**: Là bộ dữ liệu tiếng Anh, có quy mô lớn (hàng trăm ngàn đến hàng triệu dòng), nội dung đa dạng với nhiều ngữ cảnh, câu đánh giá

thường dài và chi tiết. Tỷ lệ nhãn tích cực/tiêu cực tương đối cân bằng, dữ liệu được chuẩn hóa tốt, ít nhiễu và phản ánh khá đầy đủ các tình huống sử dụng thực tế

- **Postmart**: Dữ liệu thực tế từ sàn TMĐT Việt Nam, chủ yếu là tiếng Việt, số lượng mẫu còn hạn chế (vài trăm đến vài nghìn dòng), độ đa dạng về ngôn ngữ chưa cao, câu đánh giá thường ngắn, có nhiều trường hợp lặp lại nội dung hoặc thiếu chi tiết. Sự cân bằng nhãn có thể chưa rõ ràng do tập khách hàng nhỏ hơn, và dữ liệu thực tế thường tồn tại nhiều yếu tố nhiễu như ngôn ngữ địa phương, từ lóng, ký tự đặc biệt

3.3.2. So sánh về hiệu suất mô hình và các chỉ số đánh giá

a. Độ chính xác (Accuracy)

- **Amazon Reviews**: Độ chính xác (accuracy) trên tập kiểm tra thường đạt **88–90%** ở các lần thử nghiệm với số epoch tăng dần (5, 50, 100). Đường Learning Curve khá ổn định, khoảng cách giữa đường huấn luyện và kiểm tra nhỏ, cho thấy mô hình tổng quát tốt trên dữ liệu lớn, ít bị overfitting
- **Postmart**: Độ chính xác kiểm tra thường ở mức **80–86%**, tùy vào số lượng dữ liệu và epoch. Đường Learning Curve cho thấy mô hình học nhanh nhưng có hiện tượng overfitting khi số epoch tăng, do dữ liệu nhỏ, các trường hợp đánh giá đặc biệt dễ bị "thuộc lòng". Tuy nhiên, trong phạm vi dữ liệu thực tế ở Postmart, kết quả này vẫn được đánh giá là tốt và thực tế

b. MAE (Mean Absolute Error) & MSE (Mean Squared Error)

- **Amazon Reviews**: MAE và MSE trên tập kiểm tra giảm đều qua các epoch và luôn thấp hơn tập huấn luyện, cho thấy mô hình dự đoán ổn định, không bị lệch pha nhiều
- **Postmart**: Giá trị MAE và MSE ban đầu khá tốt nhưng có xu hướng tăng nhẹ khi số epoch lớn (100), xác nhận nguy cơ overfitting. Biểu đồ Loss Curve trên tập kiểm tra xuất hiện sóng nhẹ ở cuối, chứng tỏ dữ liệu Postmart còn ít đa dạng hoặc chưa đủ lớn để mô hình học tổng quát

c. Biểu đồ Loss Curve

- **Amazon Reviews**: Đường Loss giảm mạnh ở các epoch đầu và chững lại khi tăng epoch, gần như song song giữa tập huấn luyện và kiểm tra
- **Postmart**: Đường Loss giảm nhưng xuất hiện độ "lượn sóng" ở tập kiểm tra khi epoch lớn, cho thấy mô hình bắt đầu "thuộc lòng" dữ liệu training thay vì học quy luật tổng quát

d. Các chỉ số mở rộng (Precision, Recall, F1-score, ROC-AUC)

- **Amazon Reviews**: Precision, Recall, F1-score đều cao (trên 0.86), ROC-AUC thường vượt 0.90 – xác nhận mô hình phân biệt tốt hai nhãn
- **Postmart**: Các chỉ số này có thể dao động mạnh hơn (thường trong khoảng 0.80–0.85) tùy từng lần thử nghiệm, tuy nhiên độ lệch giữa Precision và Recall không lớn, AUC vẫn > 0.85 – phù hợp với thực tế dữ liệu tiếng Việt còn hạn chế

3.3.3. Phân tích các nguyên nhân chênh lệch

- **Dung lượng và đa dạng dữ liệu**: Amazon Reviews lớn và đa dạng nên mô hình học được nhiều quy luật tổng quát, giảm thiểu overfitting. Dữ liệu Postmart còn hạn chế, nhiều trường hợp bị lặp lại, câu đánh giá ngắn, gây khó cho các mô hình deep learning vốn cần dữ liệu lớn
- **Đặc thù ngôn ngữ**: Amazon sử dụng tiếng Anh, dữ liệu chuẩn hóa cao. Postmart là tiếng Việt, dễ phát sinh từ lóng, lỗi chính tả, ký tự đặc biệt, đặc biệt là trong các vùng miền khác nhau. Điều này đòi hỏi mô hình cần nhiều dữ liệu hơn để khái quát hóa tốt
- **Mục đích sử dụng**: Amazon chủ yếu dùng để phân tích cảm xúc (sentiment analysis), còn Postmart hướng tới gợi ý sản phẩm, nên đôi khi tập trung vào các đánh giá "trung tính" hoặc phản ánh cá biệt của người dùng Việt Nam

3.3.4. Đánh giá

- **Mô hình Wide & Deep** tỏ ra hiệu quả ở cả hai tập dữ liệu, đặc biệt là với Amazon Reviews – nơi mô hình phát huy hết sức mạnh nhờ dữ liệu lớn, đa dạng và chuẩn hóa. Với dữ liệu thực tế của Postmart, mô hình vẫn cho kết

quả khá tốt, có thể ứng dụng vào thực tế, tuy nhiên để tối ưu hơn nữa, cần tăng dung lượng và đa dạng hóa nguồn dữ liệu, cũng như áp dụng thêm các kỹ thuật xử lý tiếng Việt hiện đại

- **Hiện tượng overfitting** xuất hiện rõ ràng hơn ở Postmart khi số epoch lớn, nguyên nhân chủ yếu do số mẫu nhỏ, ít trường hợp đặc biệt, mô hình học thuộc lòng thay vì tổng quát hóa
- **Kết quả thực tế** trên Postmart phản ánh đúng đặc thù TMĐT Việt Nam: người dùng đánh giá nhanh, ít bình luận chi tiết, thường xuyên đánh giá tích cực hoặc cực đoan, và tồn tại nhiều mẫu có nội dung lặp lại

3.3.5. Đề xuất cải thiện

- **Mở rộng và làm sạch dữ liệu**: Tiếp tục thu thập nhiều hơn các đánh giá thực tế, lọc bỏ spam và mẫu trùng lặp, chuẩn hóa ngôn ngữ, từ lóng, lỗi chính tả…
- **Ứng dụng các kỹ thuật NLP cho tiếng Việt**: Dùng thêm mô hình BERT tiếng Việt (PhoBERT), word2vec, hoặc transformer cho dữ liệu văn bản để cải thiện hiệu quả
- **Cải thiện biểu đồ và đánh giá mô hình**: Tăng số lần thử nghiệm với tập dữ liệu lớn hơn, sử dụng cross-validation để đánh giá ổn định
- **Triển khai thực tiễn**: Kết hợp mô hình này vào hệ thống Postmart, đánh giá theo thời gian thực để tối ưu trải nghiệm người dùng, cảnh báo sản phẩm có nhiều đánh giá tiêu cực

3.4 Kết luận

Chương 3 đã trình bày chi tiết quá trình thử nghiệm, đánh giá và so sánh các mô hình học sâu, với trọng tâm là mô hình Wide & Deep, trên hai bộ dữ liệu đánh giá sản phẩm: Amazon Reviews (quốc tế) và Postmart (thực tế Việt Nam).

Quy trình thử nghiệm tuân thủ đầy đủ các bước: chuẩn bị, tiền xử lý dữ liệu, xây dựng mô hình, huấn luyện, đánh giá và trực quan hóa kết quả thông qua các biểu đồ Learning Curve, Loss, MAE, MSE, ROC-AUC, cùng các chỉ số chính xác (accuracy), F1-score, confusion matrix…

Kết quả thử nghiệm cho thấy:

- **Trên bộ dữ liệu Amazon Reviews**, mô hình Wide & Deep đạt hiệu quả rất cao, độ chính xác trên tập kiểm tra thường xuyên trên 88-90%, các chỉ số Precision, Recall, F1-score, AUC đều vượt trội nhờ dữ liệu lớn, đa dạng và đã được chuẩn hóa tốt
- **Trên bộ dữ liệu Postmart**, mô hình vẫn đạt kết quả khả quan (accuracy trung bình 84-86%), tuy nhiên mô hình dễ gặp overfitting khi tăng số epoch, nguyên nhân do dữ liệu thực tế còn hạn chế về số lượng, chất lượng và độ đa dạng ngôn ngữ. Các biểu đồ Learning Curve, Loss Curve và các chỉ số MAE, MSE đều phản ánh đúng thực trạng này

So sánh hai bộ dữ liệu cho thấy sự khác biệt rõ rệt về hiệu quả mô hình, đồng thời nhấn mạnh vai trò quyết định của dung lượng, chất lượng và tính đa dạng của dữ liệu đầu vào đối với thành công của các mô hình học sâu trong thực tế.

Cuối cùng, chương 3 đã đưa ra các đề xuất hướng cải thiện cho dữ liệu và mô hình, đặc biệt nhấn mạnh việc mở rộng dữ liệu thực tế, ứng dụng các kỹ thuật xử lý ngôn ngữ tự nhiên hiện đại cho tiếng Việt, cũng như thử nghiệm thêm nhiều mô hình tiên tiến để nâng cao hiệu quả của hệ thống khuyến nghị sản phẩm trên nền tảng thương mại điện tử Postmart

TỔNG KẾT

1. Kết quả đạt được

Đề án đã hoàn thành các mục tiêu nghiên cứu đặt ra, cụ thể:

- Đề án đã nghiên cứu và thử nghiệm kỹ thuật học sâu, đặc biệt là mô hình Wide & Deep cho hệ thống tư vấn sản phẩm trên sàn thương mại điện tử Postmart
- Quá trình thử nghiệm được thực hiện không chỉ trên bộ dữ liệu Amazon Reviews chuẩn quốc tế mà đặc biệt đã được triển khai thực nghiệm trên chính bộ dữ liệu đánh giá sản phẩm thực tế của Postmart
- Kết quả cho thấy mô hình Wide & Deep đạt hiệu quả rất cao trên dữ liệu lớn (Amazon), và vẫn giữ được hiệu suất tốt trên dữ liệu thực tế Postmart với độ chính xác trung bình 84–86%. Điều này khẳng định tính khả thi của việc áp dụng mô hình vào hệ thống thực tế tại Việt Nam
- Thử nghiệm đã phân tích, so sánh chi tiết hiệu suất, biểu đồ học tập (Learning Curve), MAE, MSE, ROC-AUC giữa hai bộ dữ liệu, qua đó làm rõ vai trò và ảnh hưởng của chất lượng, dung lượng dữ liệu tới hiệu quả mô hình

2. Những hạn chế của đề án

Bên cạnh các kết quả tích cực, đề án còn tồn tại một số hạn chế nhất định:

- Dữ liệu Postmart còn hạn chế về số lượng và độ đa dạng, dẫn đến hiện tượng overfitting khi huấn luyện mô hình với số epoch lớn. Một số đặc thù tiếng Việt, từ lóng, lỗi chính tả… vẫn còn gây nhiễu cho quá trình xử lý
- Chưa khai thác hết tiềm năng của các mô hình ngôn ngữ hiện đại như BERT, Transformer cho tiếng Việt do hạn chế tài nguyên
- Đặc trưng đầu vào chủ yếu mới tập trung vào nội dung đánh giá, chưa tích hợp sâu các trường thông tin phụ như metadata sản phẩm, hành vi mua hàng, thời điểm đánh giá…

3. Hướng phát triển tiếp theo

Để phát huy hiệu quả và khắc phục các hạn chế nêu trên, tác giả xin đề ra các hướng phát triển tiếp theo được đề xuất như sau:

- Tiếp tục mở rộng, làm sạch và chuẩn hóa dữ liệu đánh giá sản phẩm của Postmart, tăng số lượng, đa dạng vùng miền, ngành hàng, đặc biệt là các đánh giá mang tính khách quan và chi tiết hơn
- Ứng dụng thêm các kỹ thuật xử lý ngôn ngữ tự nhiên chuyên sâu cho tiếng Việt (như PhoBERT, ViBERT, các Transformer mới), đồng thời thử nghiệm kết hợp nhiều mô hình khác nhau để tối ưu hiệu quả
- Kết hợp nhiều đặc trưng đầu vào hơn: tích hợp metadata, hành vi người dùng, lịch sử tương tác… vào mô hình để tăng chất lượng gợi ý và độ cá nhân hóa
- Triển khai thực nghiệm mô hình trên môi trường thực tế, đánh giá hiệu quả vận hành, tiếp nhận phản hồi từ người dùng để hiệu chỉnh liên tục

DANH MỤC TÀI LIỆU THAM KHẢO

[1] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, etc… (2016), "Wide & deep learning for recommender systems", Proceedings of the 1st Workshop on Deep Learning for Recommender Systems.

[2] Wenting Ye, Yichuan Charlie Tang, Hang Su, Guangxiang Zhang, Zeming Chen, Qiaojun Liu, Shixia Liu, Qiang Yang (2021), "A Transformer-Based Substitute Recommendation Model Incorporating Weakly Supervised Customer Behavior Data", KDD Proceedings.

[3] Vaswani Ashish, Shazeer Noam, Parmar Niki, Uszkoreit Jakob, Jones Llion, Gomez Aidan N., Kaiser Lukasz, Polosukhin Illia (2017), "Attention is all you need", Advances in Neural Information Processing Systems.

[4] Robin M. Schmidt (2019), "Recurrent Neural Networks (RNNs): A gentle Introduction and Overview", arXiv:1912.05911 [cs.LG].

[5] Shen, J., Shafiq, M.O. (2020), "Short-term stock market price trend prediction using a comprehensive deep learning system", *Journal of Big Data*.

[6] Reza Ravanmehr & Rezvan Mohamadrezaei ,"Session-based recommender systems", in Recommender Systems Handbook,Springer, 2022

Tham khảo từ Internet

[8]https://viblo.asia/p/recommender-system-gioi-thieu-he-thong-goi-ygAm5yJPqKdb, truy cập ngày 15/10/2023

[9]https://phamdinhkhanh.github.io/2020/02/11/NARSyscom2015.html#21c%C3%A1c-d%E1%BA%A1ng-c%E1%BB%A7a-session-based-system truy cập ngày 15/10/2023

[10]https://phamdinhkhanh.github.io/2019/12/26/Sorfmax_Recommendation_Neural_Network.html#5-s%E1%BB%AD-d%E1%BB%A5ng-m%C3%B4-h%C3%ACnh-%C4%91%E1%BB%83-recommendation, ngày 21/10/2023

[11] https://www.kaggle.com/code/ysthehurricane/tesla-stock-price-prediction-using -gru-tutorial, truy cập ngày 21/10/2023

[12]https://ai.googleblog.com/2016/06/wide-deep-learning-better-together-with.html, Truy cập ngày ngày 05/6/2024

[13]https://www.amazon.science/latest-news/amazons-transformer-based-model-boosts-substitute-recommendations-by-19-percent. Truy cập ngày ngày 25/06/2024

[14]https://www.semanticscholar.org/reader/97de32d8ca162944e5f7e83071c596d13d168109, Truy cập ngày ngày 25/06/2024

[15] https://bangdasun.github.io/2019/04/14/45-wide-and-deep-learning-for-recommender-systems/, Truy cập ngày 02/07/2024

KiểmTraTàiLiệu

BÁO CÁO KIỂM TRA TRÙNG LẶP

Thông tin tài liệu

Tên tài liệu:	Nguyễn Bảo Tùng - B22CHIS022 - Đề Án Cao học (final)
Tác giả:	Nguyễn Bảo Tùng
Điểm trùng lặp:	5
Thời gian tải lên:	18:11 31/07/2025
Thời gian sinh báo cáo:	18:13 31/07/2025
Các trang kiểm tra:	89/89 trang

Kết quả kiểm tra trùng lặp

Nguồn trùng lặp tiêu biểu

123docz.net tuhocmachinelearning.github.io ctujsvn.ctu.edu.vn

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BIÊN BẢN
HỌP HỘI ĐỒNG CHẤM ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Căn cứ quyết định số Quyết định số 1098/QĐ-HV ngày 26 tháng 06 năm 2025 của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ. Hội đồng đã họp vào hồi.....7.....giờ.....40.....phút, ngày 19 tháng 07 năm 2025 tại Học viện Công nghệ Bưu chính Viễn thông để chấm đề án tốt nghiệp thạc sĩ cho:

Học viên: **Nguyễn Bảo Tùng**

Tên đề án tốt nghiệp: **Áp dụng kỹ thuật học sâu cho tư vấn trên Sàn thương mại điện tử Postmart**

Chuyên ngành: **Hệ thống thông tin** Mã số: **8480104**

Các thành viên của Hội đồng chấm đề án tốt nghiệp có mặt: ...04.../ 05

TT	HỌ VÀ TÊN	TRÁCH NHIỆM TRONG HĐ	GHI CHÚ
1	**PGS.TS. Trần Quang Anh**	Chủ tịch	
2	**TS. Đào Thị Thúy Quỳnh**	Thư ký	
3	**PGS.TS. Nguyễn Trọng Khánh**	Phản biện 1	
4	**PGS.TS. Nguyễn Hà Nam**	Phản biện 2	
5	**TS. Trần Đăng Công**	Uỷ viên	

Các nội dung thực hiện:

1. Chủ tịch Hội đồng điều khiển buổi họp. Công bố quyết định của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ.
2. Người hướng dẫn khoa học hoặc thư ký đọc lý lịch khoa học và các điều kiện bảo vệ đề án tốt nghiệp của học viên. (có bản lý lịch khoa học và kết quả các môn học cao học của học viên kèm theo).
3. Học viên trình bày tóm tắt đề án tốt nghiệp.
4. Phản biện 1 đọc nhận xét (có văn bản kèm theo)
5. Phản biện 2 đọc nhận xét (có văn bản kèm theo)
6. Các câu hỏi của thành viên Hội đồng:

- Giải thích áp dụng phương pháp đề xuất trên sàn Postmart, nên tiền xử lý dữ liệu như?
- Tại sao bên đó huấn luyện 5 epoch là đạt và có về tốt hơn 8 và 100 epoch

7. Trả lời của học viên:

- Học viên xin được tiếp thu và chỉnh sửa đề án
- Học viên gặp khó khăn trong quá trình thu thập dữ liệu Postmart nên đã thử nghiệm trên tập Amazon

8. Thư ký đọc nhận xét về quá trình thực hiện đề án tốt nghiệp của học viên *(có văn bản kèm theo)*.

9. Hội đồng họp riêng:

- Bầu Ban kiểm phiếu:
  1. Trưởng Ban kiểm phiếu: PGS.TS. Đào Thu Thủy Quỳnh
  2. Uỷ viên Ban kiểm phiếu: PGS.TS. Vũ Hà Nam
  3. Uỷ viên Ban kiểm phiếu: TS. Trần Đăng Công
- Hội đồng chấm đề án tốt nghiệp bằng bỏ phiếu kín.
- Ban kiểm phiếu làm việc:
- Trưởng Ban kiểm phiếu báo cáo kết quả kiểm phiếu *(có Biên bản họp Ban kiểm phiếu kèm theo)*
- **Điểm trung bình của đề án tốt nghiệp:** 6,4

Kết luận:

1. Các nội dung cần chỉnh sửa, hoàn thiện sau bảo vệ đề án tốt nghiệp:

- Chỉnh sửa lại định format của Tài liệu tham khảo
- Việt hoá lại các hình vẽ trong đề án, tách dấu TIIK
- Tiến hành thực nghiệm trên dữ liệu Postmart

2. Đề nghị Học viện công nhận (hoặc không) và cấp bằng (hoặc không) thạc sĩ cho học viên:

3. Đề án tốt nghiệp có thể phát triển thành đề tài nghiên cứu cho NCS.....

Buổi làm việc kết thúc vào..... cùng ngày.

Chủ tịch	**Thư ký**
PGS.TS. Trần Quang Anh	TS. Đào Thị Thúy Quỳnh

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BẢN NHẬN XÉT LUẬN VĂN TỐT NGHIỆP THẠC SĨ
(*Dùng cho người phản biện*)

Tên đề tài luận văn: Áp dụng kỹ thuật học sâu cho tư vấn trên sàn thương mại điện tử Postmart
Chuyên ngành: Khoa học máy tính Mã số:8.48.01.01
Tên học viên: Nguyễn Bảo Tùng
Họ và tên người nhận xét: Nguyễn Hà Nam.....
Học hàm, học vị: PGS. TS Chuyên ngành: CNTT .....
Cơ quan công tác: Ban Khoa học và Đổi mới sáng tạo, ĐHQGHN.....

NỘI DUNG NHẬN XÉT

I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài:
TMĐT đang ngày càn trở thành lĩnh vực quan trọng trong tổng thể nền kinh tế. Bài toán tư vấn sản phẩm luôn được các hệ thống kinh doanh chú trọng đặc biệt trong bối cảnh toàn cầu hóa như hiện nay. Đây là một hương nghiên cứu có tính thời sự và thực tiễn cao.

II/ Về nội dung, chất lượng của luận văn, các kết quả đã đạt được (so với đề cương đã được duyệt):
Nội dung của luận văn và các kết quả đạt được cơ bản bám sát theo đề cương đã được phê duyệt.
Luận văn được trình bày trong 3 chương bao gồm giới thiệu tổng quan về bài toán tư vấn, hệ thống tư vấn, giới thiệu một số kỹ thuật học sâu áp dụng cho bài toán tư vấn và cuối cùng thử nghiệm đánh giá mô hinh trên sàn thương mại PostMart và đã cho ra một số kế quả bước đầu.

III/ Những vấn đề cần giải thích thêm:
- Mục đích của phần data generation trong hình 2.3 để làm gì?
- Sao không so sánh với các kết quả trước đây của các tác giả khác trên cùng bộ dữ liệu?
- Tại sao chọn tỷ lệ huấn luyện/ kiểm chứng là 80/20?

IV/ Kết luận:
Luận văn đáp ứng được yêu cầu cơ bản của luận văn thạc sỹ chuyên ngành KHMT(theo định hướng ứng dụng). Tôi đồng ý để học viên được bảo vệ luận văn trước Hội đồng chấm luận văn thạc sĩ

Ngày 17 tháng 7 năm 2025
NGƯỜI NHẬN XÉT
(Ký và ghi rõ họ tên)

Nguyễn Hà Nam

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

BẢN NHẬN XÉT LUẬN VĂN TỐT NGHIỆP THẠC SĨ
(*Dùng cho người phản biện*)

Tên đề tài luận văn: Áp dụng kỹ thuật học sâu cho tư vấn trên sàn thương mại điện tử PostMart.

Chuyên ngành: Hệ thống Thông tin

Mã số: 8.48.01.04

Họ và tên học viên: Nguyễn Bảo Tùng

Họ và tên người nhận xét: Nguyễn Trọng Khánh

Học hàm, học vi: PGS. TS.

Cơ quan công tác: Học viên Công nghệ Bưu chính Viễn thông

Số điện thoại : 0912314482 Email: Khanhnt@ptit.edu.vn

NỘI DUNG NHẬN XÉT

I. Cở sở khoa học và thực tiễn, sự cần thiết lựa chọn đề tài:

Đề án tìm hiểu các kỹ thuật học máy, đặc biệt là các kỹ thuật học sâu cho hệ thế vấn trên sàn thương mại điện tử, đây là bài toán có cơ sở khoa học và ý nghĩa thực tiễn.

II. Nội dung của luận văn, các kết quả đã đạt được:

Về cơ bản, đề án đã tìm hiểu được lý thuyết về hệ tư vấn, một số hướng chính, cũng như một số phương pháp học máy áp dụng trong xây dựng hệ tư vấn. Đề án cũng đề xuất áp dụng mạng nơ ron sâu và rộng để làm hệ tư vấn cho các sàn thương mại điện tử, trong đó có Postmart. Tuy nhiên, đề án chưa thực sự thực nghiệm và xây dựng được hệ tư vấn cho sàn này, thay vì đó, chỉ thử nghiệm và đánh giá mô hình đề xuất với bộ dữ liệu ngoài (Amazon Reviews for Sentiment Analysis).

III. Một số nội dung chưa đồng ý

Mục tiêu chính của đề án là xây dựng hệ thống tư vấn cho sàn TMĐT Postmart, tuy nhiên đề án chưa thực sự hoàn thành mục tiêu đặt ra. Đề án mới chỉ tìm hiểu và đề xuất áp dụng mạng nơ ron sâu và rộng, sau đó thử nghiệm đề xuất trên bộ dữ liệu Amazon Reviews for Sentiment Analysis. Chưa có hệ thống tư vấn, hay mô hình nào được xây dựng cho sàn Postmart được xây dựng, thử nghiệm, đánh giá.

Một trong những đối tượng quan trọng của đề án là sàn TMĐT Postmart, tuy nhiên quyển báo cáo không giới thiệu gì về sản này. Chỉ có phần 2.2 giới thiệu trực tiếp các bảng dữ liệu đang có trong CSDL của Postmart.

Việc thực nghiệm và đánh giá trong chương 3 không hợp lý, có nhiều điểm chưa rõ, nhiều nhận xét khá khó hiểu, tượng tự người/máy viết. Ví dụ cùng trên một quá trình training các biểu đồ của training với 5 epoch lại không phù hợp với 50, 100 epoch; các lời nhận xét sau mỗi lần thực nghiệm (ví dụ trang 45 "trong thực tế, nên dừng huấn luyện tại …", trang 48 "mô hình đang bị overfitting…", "nên dùng early stopping …") tương tự lời khuyến nghị của người/công cụ đề xuất cho quá trình thực nghiệm, chứ không phải nhận xét đánh giá thực nghiệm.

Báo cáo tổ chức chưa tốt, còn khá lộn xộn và cẩu tha. Đặc biệt chương 2, chương này giới thiệu các kỹ thuật học sâu cho TMĐT, tuy nhiên chỉ được 2 trang giới thiệu chung chung về một số mạng cơ bản, không có thông tin chi tiết và ứng dụng, ví dụ cấu trúc mạng như nào, được áp dụng ở đâu, và vào bài toán TMĐT như nào. Phân tích bài toán tư vấn sản phẩm Postmart lại được đặt trong phần 2.2 là không hợp lý …

Báo cáo không có trích dẫn bất kỳ tài liệu tham khảo nảo, mặc dù có 11 tài liệu tham khảo. Học viên cần bổ sung trích dẫn, cho những nội dung kiến thức tham khảo bên ngoài. Ngoài ra, tất cả hình từ 3.3 đến 3.12 đều có chung 1 tên "Biểu đồ MSE của mô hình"; không đưa mã nguồn vào báo cáo đề án thạc sĩ … Báo cáo quyển của đề án cần được cập nhập chỉnh sửa lại cẩn thận hơn.

IV. Những vấn đề học viên cần giải trình thêm:

- Giải thích cách áp dụng phương pháp đề xuất với dữ liệu của sàn Postmart, nên tiền xử lý dữ liệu như nào, đưa dữ liệu vào mạng để huấn luyện như nào …
- Tại sao biểu đồ huẩn luyện của 5 epoch khác và có vẻ tốt hơn với 50 và 100 epoch.

V. Kết luận

Đồng ý cho phép học viên bảo vệ luận văn tốt nghiệp.

Hà Nội, Ngày 16 tháng 7 năm 2025

NGƯỜI NHẬN XÉT

(Ký và ghi rõ họ tên)

Nguyễn Trọng Khánh