

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**



**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH KHOA HỌC MÁY TÍNH**

Đề tài

**XÂY DỰNG WEBSITE KINH DOANH
CÁC MẶT HÀNG NỘI THẤT CÓ TÍCH HỢP CHATBOT
HỖ TRỢ KHÁCH HÀNG**

**DEVELOPING A FURNITURE E-COMMERCE
WEBSITE WITH INTEGRATED
CUSTOMER SUPPORT CHATBOT**

**Sinh viên: Đinh Hoàng Nhân
Mã số: B2007253
Khóa: K46**

Cần Thơ, 12/2024

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
KHOA KHOA HỌC MÁY TÍNH**



**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
NGÀNH KHOA HỌC MÁY TÍNH**

Đề tài

**XÂY DỰNG WEBSITE KINH DOANH
CÁC MẶT HÀNG NỘI THẤT CÓ TÍCH HỢP CHATBOT
HỖ TRỢ KHÁCH HÀNG**

**DEVELOPING A FURNITURE E-COMMERCE
WEBSITE WITH INTEGRATED
CUSTOMER SUPPORT CHATBOT**

**Người hướng dẫn
TS. Lưu Tiến Đạo**

**Sinh viên thực hiện
Đình Hoàng Nhân
Mã số: B2007253
Khóa: K46**

Cần Thơ, 12/2024

NHẬN XÉT CỦA GIẢNG VIÊN

[illegible]

Cần Thơ, ngày.... tháng.... năm 2024

(GVHD ký và ghi rõ họ tên)

TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN
VÀ TRUYỀN THÔNG

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

XÁC NHẬN CHỈNH SỬA LUẬN VĂN THEO YÊU CẦU CỦA HỘI ĐỒNG

Tên luận văn (tiếng Việt và tiếng Anh):

Xây dựng website kinh doanh các mặt hàng nội thất có tích hợp chatbot hỗ trợ khách hàng (Developing a furniture e-commerce website with integrated customer support chatbot)

Họ tên sinh viên: Đinh Hoàng Nhân

MASV: B2007253

Mã lớp: DI20Z6A3

Đã báo cáo tại hội đồng ngành: Khoa học máy tính

Ngày báo cáo: 10/12/2024

Luận văn đã được chỉnh sửa theo góp ý của Hội đồng.

Cần Thơ, ngày tháng năm 20...

Giáo viên hướng dẫn

(Ký và ghi họ tên)

LỜI CẢM ƠN

Trong quá trình nghiên cứu và viết luận văn, em đã nhận được sự quan tâm, hướng dẫn, giúp đỡ của nhiều cá nhân trong và ngoài trường.

Em xin chân thành cảm ơn sự quan tâm và chỉ bảo, hướng dẫn chuyên môn của quý Thầy Cô Khoa công nghệ thông tin và Truyền Thông-Trường Đại học Cần Thơ trong suốt quá trình học tập.

Đặc biệt, Em xin được gửi lời cảm ơn sâu sắc tới thầy TS. Lưu Tiến Đạo đã trực tiếp hướng dẫn Em hoàn thành tốt luận văn này.

Cuối cùng, Em xin gửi lời cảm ơn đến gia đình, những người thân và bạn bè đã thường xuyên quan tâm, truyền động lực, động viên, chia sẻ kinh nghiệm và cung cấp tài liệu hữu ích trong thời gian học tập, cũng như trong suốt quá trình thực hiện luận văn.

Cần Thơ, ngày tháng 12 năm 2024

Người viết

Đinh Hoàng Nhân

MỤC LỤC

PHẦN GIỚI THIỆU	1
1. Đặt vấn đề	1
2. Những nghiên cứu liên quan	1
3. Mục tiêu đề tài	2
4. Đối tượng và phạm vi nghiên cứu	2
5. Phương pháp nghiên cứu	3
6. Nội dung nghiên cứu	3
7. Bố cục quyền luận văn	4
PHẦN NỘI DUNG	5
CHƯƠNG I: MÔ TẢ BÀI TOÁN	5
1. Mục tiêu của đề tài	5
2. Các chức năng mong đợi.	5
2.1. Các chức năng dành cho phân hệ người dùng chung.	5
2.2. Các chức năng dành cho phân hệ chủ cửa hàng	6
3. Kiến trúc tổng thể	6
4. Chức năng từng phần trong hệ thống.	7
4.1. Frontend	7
4.2. Backend	7
4.3. MongoDB Database	7
4.4. Classify Comment Service	7
4.5. Chatbot	7
CHƯƠNG II: VẤN ĐỀ VÀ GIẢI PHÁP	8
1. Vấn đề đặt ra	8
2. Giải pháp liên quan	8
2.2. NestJS Framework	8
2.3. ReduxToolkit	9
2.4. FastAPI	9
2.5. Tiền xử lý dữ liệu	9
2.6. PhoBert	11
2.7. Logistic Regression	11
2.8. SVM (Support Vector Machine)	12
2.9. Random Forest	12
2.10. Chatbot hỗ trợ khách hàng	12
2.11. Các chỉ số đánh giá	14
CHƯƠNG III: CÀI ĐẶT GIẢI PHÁP	17
1. Thiết kế cơ sở dữ liệu và xây dựng bộ API cho website.	17
1.1. Sơ đồ Usecase tổng quát	17
1.2. Mô hình dữ liệu mức quan niệm (CDM)	18
1.3. Xây dựng thực thể	18
1.4. Thiết kế mô hình chức năng	22
1.4.1. Các chức năng dành cho nhóm người dùng chung	22
1.5. Xây dựng API cho website	23
2. Xây dựng hệ thống phân loại ý kiến	27
2.1. Chuẩn bị dữ liệu	27
2.2. Vector hóa văn bản.	28
2.3. Huấn luyện mô hình phân loại	29

2.4. Kết quả huấn luyện	33
3. Xây dựng chatbot với Rasa NLU	34
3.1. Xây dựng mô hình chatbot	34
3.2. Huấn luyện chatbot	37
3.3. Kết quả huấn luyện và nhận xét	39
CHƯƠNG IV: KIỂM THỬ ĐÁNH GIÁ	43
1. Giao diện website	43
1.1. Giao diện trang chủ	43
1.2. Giao diện tất cả sản phẩm	44
1.3. Giao diện thông tin người dùng	45
1.4. Giao diện giỏ hàng	45
1.5. Giao diện thanh toán	46
1.7. Giao diện hội thoại	47
1.8. Giao diện cửa sổ chatbot	48
1.9. Giao diện thống kê sản phẩm	48
1. Kết quả kiểm thử	49
1.1. Mục đích kiểm thử	49
1.2. Phạm vi kiểm thử	49
1.3. Cách tiếp cận	49
1.4. Tiêu chí đánh giá	49
1.5. Môi trường kiểm thử	49
1.6. Các chức năng kiểm thử	49
2. Đánh giá.	51
2.1. Website sàn thương mại điện tử kinh doanh nội thất	51
2.2. Hệ thống tự động phân loại đánh giá	51
2.3. Chatbot	51
PHẦN KẾT LUẬN	52
1. Kết quả đạt được	52
2. Hạn chế	52
3. Hướng phát triển	52
TÀI LIỆU THAM KHẢO	53

DANH MỤC BẢNG

Bảng 3.1	Bảng mô tả thực thể User	19
Bảng 3.2	Bảng mô tả thực thể Shop	19
Bảng 3.3	Bảng mô tả thực thể Product	19
Bảng 3.4	Bảng mô tả thực thể Order	20
Bảng 3.5	Bảng mô tả thực thể Category	20
Bảng 3.6	Bảng mô tả thực thể Chat	20
Bảng 3.7	Bảng mô tả thực thể Comment	21
Bảng 3.8	Bảng mô tả thực thể Message	21
Bảng 3.9	API chức năng lấy danh sách tất cả sản phẩm	23
Bảng 3.10	Kết quả trả về của API lấy danh sách tất cả sản phẩm	23
Bảng 3.11	API chức năng lấy thông tin người dùng	24
Bảng 3.12	Kết quả trả về của API lấy thông tin người dùng	24
Bảng 3.13	API chức năng lấy danh sách đơn hàng của người dùng	24
Bảng 3.14	Kết quả trả về của API lấy danh sách đơn hàng người dùng	24
Bảng 3.15	API chức năng lấy danh sách quảng cáo của cửa hàng	24
Bảng 3.16	Kết quả trả về của API lấy danh sách quảng cáo cửa hàng	24
Bảng 3.17	API chức năng tạo sản phẩm mới	25
Bảng 3.18	Kết quả trả về của API tạo sản phẩm mới	25
Bảng 3.19	API chức năng tạo danh mục sản phẩm	25
Bảng 3.20	Kết quả trả về của API tạo danh mục sản phẩm mới	26
Bảng 3.21	API tạo đơn đặt hàng	26
Bảng 3.22	Kết quả trả về của API tạo đơn đặt hàng	26
Bảng 3.23	Bảng so sánh độ chính xác của ba mô hình huấn luyện	33
Bảng 3.24	Bảng bộ ý định (intent)	34
Bảng 3.25	Bảng cấu hình pipeline	38
Bảng 4.1	Bảng kiểm thử chức năng đặt hàng	50
Bảng 4.2	Bảng kiểm thử chức năng tạo sản phẩm mới	50
Bảng 4.3	Bảng kiểm thử chức năng tạo danh mục mới	51

DANH MỤC HÌNH ẢNH

Hình 1.1	Hình mô tả kiến trúc tổng thể của hệ thống.....	6
Hình 2.1	Cơ chế hoạt động của Redux	9
Hình 2.2	Mô tả quy trình tiền xử lý dữ liệu.	10
Hình 2.3	Vai trò của PhoBERT trong hệ thống.	11
Hình 2.4	Mô tả các bước xây dựng chatbot với Rasa NLU	13
Hình 2.5	Cấu trúc thư mục dự án Rasa NLU	14
Hình 2.6	Ma trận nhầm lẫn cho bài toán phân loại hai lớp	14
Hình 3.1	Mô tả sơ đồ Usecase tổng quát	17
Hình 3.2	Sơ đồ mô hình dữ liệu mức quan niệm	18
Hình 3.3	Mô tả sơ đồ chức năng dành cho nhóm người dùng chung	22
Hình 3.4	Mô tả sơ đồ chức năng dành cho chủ cửa hàng	23
Hình 3.5	Hình minh họa biểu đồ phân bố số lượng mẫu của tập dữ liệu	27
Hình 3.6	Hình minh họa biểu đồ phân bố số lượng mẫu	28
Hình 3.7	Hình minh họa biểu đồ phân bố số lượng mẫu	28
Hình 3.8	Ma trận nhầm lẫn của mô hình Random Forest	30
Hình 3.9	Ma trận nhầm lẫn của mô hình SVM	31
Hình 3.10	Ma trận nhầm lẫn của mô hình Logistic Regression	32
Hình 3.11	Hình mô tả cách đưa ra câu trả lời	36
Hình 3.12	Độ chính xác của chatbot dựa trên chỉ số Accuracy	39
Hình 3.13	Độ chính xác tính theo chỉ số F1-Score	40
Hình 3.14	Ma trận nhầm lẫn của chatbot sau huấn luyện	41
Hình 3.15	Biểu đồ minh họa độ tin cậy của chatbot.....	42
Hình 4.1	Giao diện trang chủ	43
Hình 4.2	Giao diện trang hiển thị tất cả sản phẩm	44
Hình 4.3	Giao diện trang thông tin người dùng	45
Hình 4.4	Giao diện trang giỏ hàng	45
Hình 4.5	Giao diện trang thanh toán	46
Hình 4.6	Giao diện trang lịch sử đặt hàng	47
Hình 4.7	Giao diện trang hội thoại	47
Hình 4.8	Giao diện khung trò chuyện với chatbot	48
Hình 4.9	Giao diện thống kê sản phẩm	48

DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt	Từ viết đầy đủ
1	API	Application Programing Interface
2	FP	False Positive
3	FN	False Negative
4	NLU	Natural Language Understanding
5	NLP	Natural Language Processing
6	SVM	Support Vector Machine
7	TP	True Positive
8	TN	True Negative

ABSTRACT

Nowadays, with the development of information technology and the emergence of e-commerce platforms, shopping has become easier and more convenient.

The topic "Building a website to sell furniture products with integrated customer support chatbot" focuses on developing an online shopping platform for furniture products with customer support features, improving user experience. The website allows stores to post furniture products for sale, integrates a customer support chatbot built on the open source chatbot framework Rasa NLU (Natural Language Understanding), an automatic review classification system that helps reduce product review operations and automatically classifies reviews through customer text reviews. The automatic classification of customer reviews is solved by the PhoBert model in word separation and text vectorization and the basic model is pre-trained specifically for Vietnamese, then uses the Logistic Regression machine learning algorithm to train the classification model.

The website is developed in JavaScript using NextJS for the interactive interface, NestJS provides APIs for tasks related to retrieving products and user information, Python FastAPI is responsible for automatically classifying customer reviews and Server Rasa serves the communication between chatbot and customers.

TÓM TẮT

Hiện nay, với sự phát triển của lĩnh vực công nghệ thông tin và sự xuất hiện của các sàn thương mại điện tử, việc mua sắm trở nên dễ dàng và thuận lợi.

Đề tài “xây dựng website kinh doanh các mặt hàng nội thất có tích hợp chatbot hỗ trợ khách hàng” tập trung vào việc phát triển một nền tảng mua sắm trực tuyến với các mặt hàng nội thất với các tính năng hỗ trợ khách hàng, nâng cao trải nghiệm người dùng. Website cho phép các cửa hàng đăng bán sản phẩm nội thất, tích hợp chatbot hỗ trợ khách hàng được xây dựng trên khung chatbot mã nguồn mở Rasa NLU (Natural Language Understanding), hệ thống phân loại đánh giá tự động giúp giảm bớt thao tác đánh giá sản phẩm và tự động phân loại đánh giá thông qua đoạn đánh giá bằng văn bản của khách hàng. Việc phân loại tự động đánh giá của khách hàng được giải quyết bởi mô hình PhoBert trong việc tách từ và vector hóa văn bản và mô hình cơ bản được huấn luyện trước dành riêng cho tiếng Việt, sau đó sử dụng thuật toán máy học Logistic Regression để huấn luyện mô hình phân loại.

Website được phát triển trên ngôn ngữ JavaScript sử dụng NextJS dùng cho phần giao diện tương tác, NestJS đảm nhận cung cấp các API cho các tác vụ liên quan đến truy xuất các sản phẩm, thông tin người dùng, Python FastAPI đảm nhận việc phân loại tự động đánh giá của khách hàng và Server Rasa phục vụ việc giao tiếp giữa chatbot và khách hàng.

PHẦN GIỚI THIỆU

1. Đặt vấn đề

Hiện nay, có rất nhiều doanh nghiệp hoặc cá nhân đã và đang hoạt động trong lĩnh vực kinh doanh các mặt hàng nội thất. Đa số các doanh nghiệp đó thường có một trang web để quảng bá, buôn bán các mặt hàng đang kinh doanh, nhưng vẫn còn những cá nhân, doanh nghiệp nhỏ lẻ chưa có trang web để buôn bán riêng hoặc giữ lối kinh doanh truyền thống là mua bán tại cửa hàng, điều này có thể khiến cho doanh số của họ không được cao và còn hạn chế trong việc tiếp cận với khách hàng từ xa. Vì vậy, việc có một sàn thương mại điện tử giúp kết nối các khách hàng với nhà kinh doanh, đơn giản hóa quá trình mua bán và tụ hội các nhà kinh doanh lại với nhau trong lĩnh vực nội thất là vô cùng hữu ích.

Với mục tiêu trên, website kinh doanh các mặt hàng nội thất được xây dựng để giải quyết các vấn đề về việc giúp các nhà kinh doanh nội thất tiếp cận với khách hàng từ xa, trực tuyến, tổng hợp các nhà kinh doanh lại trên một sàn thương mại điện tử chuyên biệt về lĩnh vực nội thất giúp khách hàng dễ dàng tìm kiếm và có nhiều sự lựa chọn khi mua sắm. Ngoài ra, website còn tích hợp một trợ lý ảo (chatbot) để chăm sóc và hỗ trợ khách hàng trong việc mua sắm, hỏi đáp về các mặt hàng và tra cứu đơn hàng. Bên cạnh đó, website cũng có hệ thống tự động phân loại đánh giá của khách hàng theo năm cấp độ, giúp khách hàng giảm lược quá trình đánh giá và giúp cho các nhà buôn bán tổng hợp, phân tích ý kiến khách hàng về những sản phẩm mà họ đang kinh doanh, kịp thời cải thiện chất lượng sản phẩm theo thị hiếu của khách hàng.

2. Những nghiên cứu liên quan

Trong nước, theo Phòng Thương Mại và Công Nghiệp Việt Nam về Báo Cáo Nghiên Cứu “Thương mại điện tử trên mạng xã hội Việt Nam: Một số vấn đề pháp lý” [1]. Hoạt động thương mại điện tử (TMĐT) tại Việt Nam đang trong giai đoạn đang trong giai đoạn bùng nổ với tăng trưởng hàng năm với mức rất cao. Ngoài các website và sàn giao dịch TMĐT thì hoạt động giao dịch, mua bán hàng hóa, dịch vụ thông qua phương tiện là mạng xã hội hiện cũng đang thu hút số lượng lớn các cá nhân, doanh nghiệp. Xu hướng sử dụng mạng xã hội làm kênh tiếp thị, phân phối đang có chiều hướng gia tăng trong thời gian trở lại đây.

Theo kết quả nghiên cứu từ bài báo “Nghiên cứu các yếu tố ảnh hưởng đến ý định sử dụng chatbot của khách hàng: Góc nhìn tại Việt Nam” [2]. Công nghệ mới đã và đang mang lại hiệu quả kinh doanh và tương tác giữa doanh nghiệp và khách hàng. Trí tuệ nhân tạo và chatbot là ví dụ điển hình cho xu hướng ứng dụng công nghệ mới vào thương mại điện tử. Chúng thay thế con người nhưng vẫn thể hiện ưu thế vượt trội như hoạt động thông suốt, hạn chế rủi ro, và tiết kiệm chi phí.

Chatbot được tích hợp trên nhiều phương tiện khác nhau như website, ứng dụng di động, và mạng xã hội. Chatbot cho phép doanh nghiệp tăng cường trao đổi với khách hàng trong thời gian thực. Chatbot mang đến nhiều lợi ích như sự thuận tiện, linh hoạt, và khả năng truy cập. Đồng thời, chatbot có thể thay thế cho nhân viên để thực hiện trao đổi và tư vấn với khách hàng, thậm chí giải quyết vấn đề mà con người không thực hiện được. Vì thế, doanh nghiệp có xu hướng sử dụng chatbot như một công cụ hỗ trợ kinh doanh trực tuyến.

Với các sàn thương mại điện tử lớn đang có mặt hiện nay như Lazada¹, Shopee², Tiki³,... đã mang lại nhiều sự thuận tiện cho khách hàng và sự đầu tư mạnh mẽ trong lĩnh vực trí tuệ nhân tạo phát triển các chatbot hỗ trợ khách hàng của các ngân hàng như Vietcombank (Ngân hàng thương mại cổ phần ngoại thương Việt Nam), VPBank (Ngân hàng thương mại cổ phần Việt Nam Thịnh Vượng), TPBank (Ngân hàng thương mại cổ phần Tiên Phong), nhưng việc kết hợp chatbot vào các sàn thương mại điện tử đặc biệt là trong lĩnh vực nội thất và sàn thương mại điện tử chuyên biệt cho các doanh nghiệp kinh doanh nội thất còn chưa được áp dụng rộng rãi.

3. Mục tiêu đề tài

Mục tiêu của đề tài là xây dựng một sàn thương mại điện tử chuyên biệt cho các cá nhân, doanh nghiệp đang hoạt động trong lĩnh vực kinh doanh các mặt hàng nội thất kết hợp trợ lý ảo (chatbot) hỗ trợ khách hàng và hệ thống tự động phân loại đánh giá của khách hàng trên trình duyệt web với giao diện đơn giản, dễ sử dụng và thân thiện với người dùng.

4. Đối tượng và phạm vi nghiên cứu

4.1. Đối tượng nghiên cứu

- Các đại lý, nhà kinh doanh nội thất: Là những người đăng bán các mặt hàng nội thất trên sàn thương mại điện tử.
- Người dùng cuối: Là những khách hàng có nhu cầu mua các mặt hàng nội thất.

4.2. Phạm vi nghiên cứu

- Nghiên cứu về Framework NextJS: Xây dựng giao diện tương tác với người dùng.
- Nghiên cứu về Framework NestJS: Xây dựng máy chủ, APIs cho các tác vụ của người dùng.
- Nghiên cứu về Python FastAPI: Xây dựng máy chủ cho việc phân loại đánh giá của khách hàng
- Nghiên cứu về Rasa NLU: Xây dựng chatbot hỗ trợ khách hàng
- Nghiên cứu về PhoBERT và mô hình máy học Logistic Regression: Xây dựng mô hình phân loại.
- Nghiên cứu về MongoDB: Cơ sở dữ liệu lưu trữ thông tin.

¹ Sàn thương mại điện tử Lazada - <https://www.lazada.vn/>

² Sàn thương mại điện tử Shopee - <https://shopee.vn/>

³ Sàn thương mại điện tử Tiki - <https://tiki.vn/>

5. Phương pháp nghiên cứu

5.1. Tìm hiểu và phân tích yêu cầu

Nghiên cứu nhu cầu mua sắm của người dùng với các mặt hàng nội thất, bao gồm các yếu tố như giá cả, chất lượng, thương hiệu và xu hướng tiêu dùng. Phân tích tình hình kinh doanh và nhu cầu buôn bán của các doanh nghiệp trong lĩnh vực nội thất, đặc biệt trong môi trường thương mại điện tử.

5.2. Tìm hiểu công nghệ

Nghiên cứu các công nghệ xây dựng ứng dụng web như ReactJS, NextJS, NestJS và các giải pháp lưu trữ dữ liệu như MongoDB. Tìm hiểu các mô hình máy học và kỹ thuật xử lý ngôn ngữ tự nhiên liên quan đến văn bản tiếng Việt. Nghiên cứu Framework Rasa NLU để xây dựng chatbot hỗ trợ khách hàng. Xác định công cụ và nền tảng phù hợp để triển khai website.

5.3. Tìm hiểu về UX/UI

Nghiên cứu các nguyên tắc thiết kế giao diện người dùng (UX/UI) để đảm bảo website thân thiện, trực quan và dễ sử dụng. Tìm hiểu, tham khảo bố cục của các sản phẩm thương mại điện tử có sẵn.

5.4. Kiểm thử và đánh giá

Tiến hành kiểm thử website bằng phương pháp thủ công. Thực hiện các trường hợp kiểm thử cụ thể để đảm bảo tính chính xác và độ ổn định của website.

6. Nội dung nghiên cứu

6.1. Phân tích nhu cầu mua sắm của người dùng

Thu thập, quan sát dữ liệu từ các kênh thương mại điện tử như Lazada, Shopee, Tiki, để hiểu hành vi tiêu dùng và nhu cầu của khách hàng đối với các mặt hàng nội thất.

6.2. Phát triển website thương mại điện tử

Thiết kế giao diện website. Thiết kế cơ sở dữ liệu lưu trữ thông tin khách hàng và thông tin sản phẩm. Xây dựng các chức năng cho website như: Hiển thị sản phẩm, giỏ hàng, danh sách yêu thích, đơn hàng, thanh toán, quản lý sản phẩm, quản lý danh mục, đánh giá sản phẩm, v.v... Tích hợp phương thức đăng nhập, đăng ký bằng tài khoản Email.

6.3. Xây dựng chatbot hỗ trợ khách hàng

Áp dụng Rasa NLU để xây dựng chatbot hỗ trợ người dùng tìm kiếm sản phẩm, đề xuất sản phẩm, giải đáp thắc mắc. Tích hợp chatbot vào website để nâng cao trải nghiệm người dùng.

6.4. Xây dựng mô hình phân loại đánh giá của khách hàng

Thu thập ý kiến đánh giá của khách hàng trên các sản phẩm thương mại điện tử. Tiến hành xử lý dữ liệu đã thu thập được. Áp dụng PhoBert Embeddings để vector hóa các câu văn bản với mô hình đã được huấn luyện trước của VinAI cho ngôn ngữ tiếng Việt. Huấn luyện mô hình Logistic Regression cho bài toán phân loại văn bản.

6.5. Tối ưu hóa trải nghiệm người dùng (UX/UI)

Xây dựng giao diện trực quan, dễ sử dụng. Cải thiện tốc độ tải trang và tối ưu hóa các tương tác trên website.

6.6. Đánh giá và kiểm thử sản phẩm

Kiểm tra tính ổn định, hiệu suất và khả năng mở rộng của website. Thu thập ý kiến để cải thiện sản phẩm.

7. Bố cục quyền luận văn

Phần giới thiệu

Giới thiệu tổng quát về đề tài

Phần nội dung

Chương I: Mô tả bài toán

Chương II: Thiết kế giải pháp

Chương III: Cài đặt giải pháp

Chương IV: Đánh giá và kiểm thử

Phần kết luận

Trình bày kết quả đạt được và hướng phát triển của sản phẩm.

Tài liệu tham khảo

PHẦN NỘI DUNG

CHƯƠNG I: MÔ TẢ BÀI TOÁN

1. Mục tiêu của đề tài

Xây dựng hoàn thiện một sàn thương mại điện tử tập trung vào lĩnh vực nội thất tích hợp chatbot nhằm tạo ra một nền tảng kinh doanh hiện đại, tiện ích, giúp kết nối các doanh nghiệp kinh doanh nội thất với người tiêu dùng một cách hiệu quả. Cung cấp đầy đủ thông tin về sản phẩm, hỗ trợ người dùng tìm kiếm, lựa chọn và mua sắm dễ dàng, tích hợp chatbot thông minh giúp giải đáp thắc mắc, tư vấn sản phẩm và nâng cao trải nghiệm người dùng. Đồng thời, hệ thống phân loại ý kiến khách hàng sử dụng mô hình máy học tự động phân tích và đánh giá phản hồi của người dùng, giúp doanh nghiệp nắm bắt kịp thời nhu cầu, cải thiện chất lượng dịch vụ và tối ưu hóa chiến lược kinh doanh. Qua đó, đề tài không chỉ hỗ trợ doanh nghiệp nâng cao năng lực cạnh tranh mà còn thúc đẩy sự phát triển của ngành nội thất trong môi trường thương mại điện tử.

2. Các chức năng mong đợi.

Các chức năng của website được chia theo hai thành phần chính bao gồm các chức năng chung cho thành phần người dùng cuối là những khách hàng mua sắm sản phẩm và các chức năng liên quan đến việc quản lý cửa hàng dành cho thành phần người dùng là các doanh nghiệp đăng bán các mặt hàng trên website này.

2.1. Các chức năng dành cho phân hệ người dùng chung.

- **Tìm kiếm sản phẩm:** Khách hàng có thể tìm kiếm các sản phẩm có trên website bằng tên sản phẩm hoặc mã của sản phẩm, tìm kiếm sản phẩm theo danh mục hoặc tìm kiếm cửa hàng theo tên của cửa hàng.
- **Chức năng đăng nhập/đăng ký:** Để có thể sử dụng thêm nhiều chức năng của website, khách hàng cần có tài khoản trên hệ thống. Vì vậy, khách hàng có thể đăng ký tài khoản mới hoặc đăng nhập vào website bằng tài khoản đã tạo để sử dụng các dịch vụ cần xác thực tài khoản của website. Ngoài ra, website còn tích hợp đăng nhập hoặc đăng ký tài khoản bằng tài khoản Email.
- **Chức năng xem chi tiết sản phẩm:** Khách hàng có thể xem chi tiết thông tin của sản phẩm để có thể đưa ra các lựa chọn mua sắm.
- **Chức năng thêm sản phẩm vào giỏ hàng:** Khách hàng có thể thêm sản phẩm vào giỏ hàng để lưu các sản phẩm muốn mua và thanh toán cùng lúc nhiều sản phẩm khác nhau.
- **Chức năng thêm sản phẩm yêu thích:** Khách hàng có thể thêm sản phẩm vào mục yêu thích để có thể truy cập nhanh vào sản phẩm, giảm bớt thời gian tìm kiếm sản phẩm và có thể thanh toán bất cứ khi nào.
- **Chức năng thanh toán:** Khách hàng có thể thanh toán các sản phẩm trong đơn hàng với các hình thức thanh toán.
- **Chức năng quản lý thông tin cá nhân:** Quản lý thông tin cá nhân của khách hàng.
- **Chức năng xem lịch sử đặt hàng:** Khách hàng có thể xem lịch sử mua hàng đã mua những gì và đồng thời xem được trạng thái đơn hàng đang đặt.

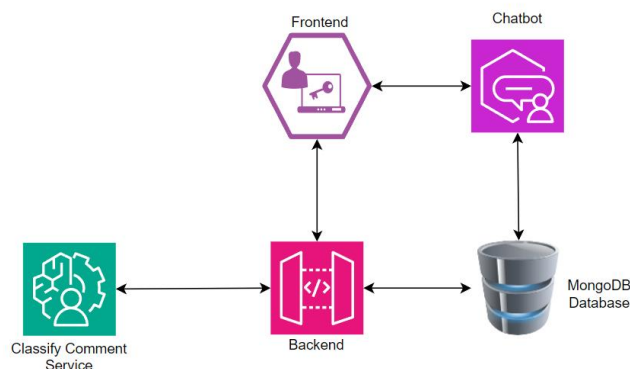
- **Chức năng trò chuyện thời gian thực:** Khách hàng có thể trao đổi trực tiếp với các nhà buôn bán trên sàn về các vấn đề về sản phẩm, nhu cầu, yêu cầu hoặc các vấn đề liên quan trong thời gian thực.
- **Chức năng trợ lý ảo (chatbot):** Khách hàng có thể trò chuyện với chatbot để giải đáp thắc mắc hoặc tra cứu, tìm kiếm, hỏi đáp về sản phẩm.
- **Chức năng đánh giá:** Sau khi khách hàng đã mua và thanh toán thành công, khách hàng có thể để lại đánh giá về sản phẩm. Các đánh giá này sẽ được phân loại tự động thông qua đoạn đánh giá của khách hàng qua 5 cấp độ.
- **Chức năng đăng ký tài khoản doanh nghiệp:** Khách hàng không chỉ là những người mua sắm đơn thuần, mà có thể là doanh nghiệp trong tương lai, vì vậy khách hàng có thể đăng ký cửa hàng để bắt đầu kinh doanh các mặt hàng nội thất.

2.2. Các chức năng dành cho phân hệ chủ cửa hàng

- **Quản lý sản phẩm:** Doanh nghiệp có thể dễ dàng quản lý thông tin các sản phẩm đang kinh doanh, thay đổi, cập nhật thông tin sản phẩm, thêm sản phẩm mới hoặc xóa sản phẩm khỏi cửa hàng, phân tích một sản phẩm cụ thể về số lượng đã bán, tồn kho, các đánh giá của khách hàng,...
- **Chức năng quản lý danh mục:** Doanh nghiệp có thể quản lý các danh mục sản phẩm, thêm, xóa các sản phẩm trong danh mục hoặc thêm danh mục sản phẩm mới.
- **Chức năng quản lý quảng cáo:** Doanh nghiệp có thể tạo các chiến dịch quảng cáo hoặc chỉnh sửa thông tin quảng cáo để đưa ra các chiến lược kinh doanh.
- **Chức năng trò chuyện thời gian thực:** Doanh nghiệp có thể trao đổi với khách hàng trực tiếp trên website.

3. Kiến trúc tổng thể

Hệ thống được chia thành 5 thành phần được thể hiện bằng Hình 1.1 bên dưới như sau:



Hình 1.1 Hình mô tả kiến trúc tổng thể của hệ thống.

- **Frontend:** Xây dựng giao diện người dùng bằng NextJS, sử dụng ReduxToolkit để quản lý trạng thái người dùng, sử dụng TailwindCSS để tạo bố cục và tạo kiểu cho giao diện.
- **Backend:** Sử dụng NestJS để triển khai RESTful API⁴ xử lý logic hệ thống, Websocket cho tính năng trò chuyện thời gian thực.

⁴ Tiêu chuẩn thiết kế API cho ứng dụng web

- **MongoDB Database:** Lưu trữ thông tin sản phẩm, thông tin người dùng, thông tin đơn hàng, thông tin cửa hàng, hội thoại.
- **Classify Comment Service:** Hệ thống phân loại ý kiến khách hàng sử dụng mô hình đã được huấn luyện với PhoBERT và Logistic Regression và được phục vụ như một máy chủ bằng FastAPI.
- **Chatbot:** Sử dụng Rasa NLU để giao tiếp và hỗ trợ giữa chatbot và người dùng.

4. Chức năng từng phần trong hệ thống.

4.1. Frontend

- Hiển thị danh sách sản phẩm, thông tin chi tiết sản phẩm
- Hiển thị danh sách danh mục, các sản phẩm trong danh mục,
- Hiển thị giao diện giỏ hàng, đơn hàng, danh sách sản phẩm yêu thích, giao diện thanh toán, lịch sử đơn hàng.
- Hiển thị giao diện quản lý sản phẩm, quản lý đơn hàng, quản lý danh mục, quản lý quảng cáo cho chủ doanh nghiệp.
- Hiển thị giao diện trò chuyện giữa khách hàng và chủ cửa hàng.
- Hiển thị giao diện tương tác giữa khách hàng và chatbot.

4.2. Backend

- Xử lý yêu cầu của người dùng, quản lý dữ liệu sản phẩm, người dùng, đơn hàng, hội thoại.
- Tích hợp hệ thống phân loại ý kiến khách hàng.

4.3. MongoDB Database

- Lưu trữ thông tin của người dùng, sản phẩm, đơn hàng, hội thoại.

4.4. Classify Comment Service

- Xử lý phân loại ý kiến từ người dùng.

4.5. Chatbot

- Tư vấn sản phẩm, trả lời các câu hỏi liên quan đến sản phẩm, chính sách mua hàng, vận chuyển, đơn hàng.

CHƯƠNG II: VẤN ĐỀ VÀ GIẢI PHÁP

1. Vấn đề đặt ra

Làm thế nào để xây dựng được website sàn thương mại điện tử kinh doanh các mặt hàng nội thất gồm các chức năng: Đăng nhập, đăng ký, quản lý thông tin cá nhân, quản lý đơn hàng, quản lý giỏ hàng và mục sản phẩm yêu thích, mua hàng trực tuyến, quản lý sản phẩm, quản lý đơn hàng dành cho người bán, quản lý danh mục, quản lý quảng cáo, trò chuyện thời gian thực, đánh giá ý kiến sản phẩm đã mua.

- Làm thế nào để xây dựng chatbot hỗ trợ khách hàng.
- Làm thế nào để tích hợp chatbot vào website.
- Chuẩn bị dữ liệu học cho mô hình phân loại ý kiến khách hàng.
- Làm thế nào để xây dựng mô hình phân loại văn bản tiếng Việt
- Làm thế nào để tích hợp mô hình phân loại vào website.
- Làm thế nào để lưu trữ thông tin người dùng và tiết kiệm bộ nhớ, giảm tải cho máy chủ.

2. Giải pháp liên quan

Là một Framework mã nguồn mở cho React, được phát triển bởi Vercel, giúp cho việc xây dựng ứng dụng web hiệu quả hơn, đơn giản hơn, cung cấp nhiều tính năng như Serverside Rendering (SSR), Client side rendering (CSR) và Static Site Generation (SSG) và có hỗ trợ Search Engine Optimization (SEO) giúp trang web hiển thị cao hơn trên các trang kết quả của công cụ tìm kiếm.

2.1. NextJS Framework

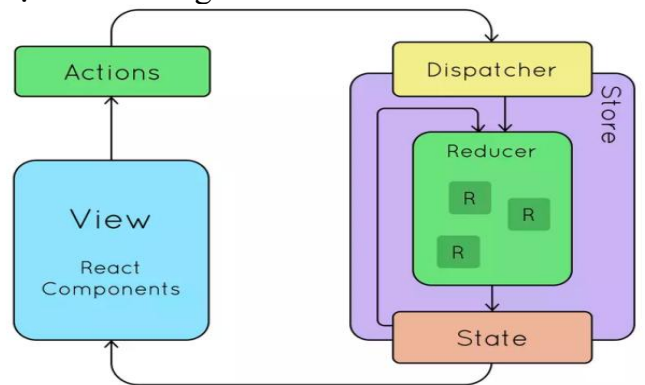
NextJS còn cung cấp cơ chế routing dựa trên cấu trúc thư mục dự án, tăng sự linh hoạt trong việc định tuyến đường dẫn của các trang và có thể chia nhiều bố cục (layout) để chia sẻ cho các route sử dụng giúp giảm thiểu thời gian phát triển và tái sử dụng lại các bố cục.

2.2. NestJS Framework

NestJS là một Framework để xây dựng các ứng dụng serverside dựa trên nền tảng NodeJS, được xây dựng và hỗ trợ ngôn ngữ Typescript, kết hợp các tính năng của lập trình hướng đối tượng OOP (Object Oriented Programming). Có kiến trúc cụ thể buộc phải tuân theo là module, provider và controller, đảm bảo tính mở rộng và tách biệt các thành phần của dự án, vì vậy NestJS cung cấp một kiến trúc thống nhất, dễ mở rộng, dễ bảo trì.

2.3. ReduxToolkit

Redux-Toolkit là một thư viện chính thức của Redux, được xem như một công cụ quản lý trạng thái phổ biến trong các ứng dụng sử dụng JavaScript, đặc biệt là ReactJS và các thư viện, Framework của ReactJS. Redux-Toolkit cung cấp các API và công cụ giúp giảm thiểu và lặp lại mã (code), hỗ trợ cho các tác vụ bất đồng bộ, cơ chế hoạt động của Redux được mô tả bằng Hình 2.1 bên dưới.



Hình 2.1 Cơ chế hoạt động của Redux

Ban đầu, các trạng thái được khai báo và lưu trữ trong một kho gọi là store, các trạng thái này được hiển thị ra giao diện. Khi có hành động (Actions) làm thay đổi giá trị của trạng thái, Redux-Toolkit cập nhật lại trạng thái thông qua các Dispatcher đã được định nghĩa, dựa vào sự kiện được kích hoạt mà các hàm xử lý (Reducers) sẽ được thực hiện. Kết quả cuối cùng là trạng thái đã được xử lý và lưu lại vào kho, sau đó hiển thị ra giao diện. Điểm mạnh ở đây là có thể truy cập vào trạng thái ở toàn cục và có thể tái thiết lập lại trạng thái đồng thời với phiên hoạt động của trình duyệt hoặc lưu trữ lâu dài.

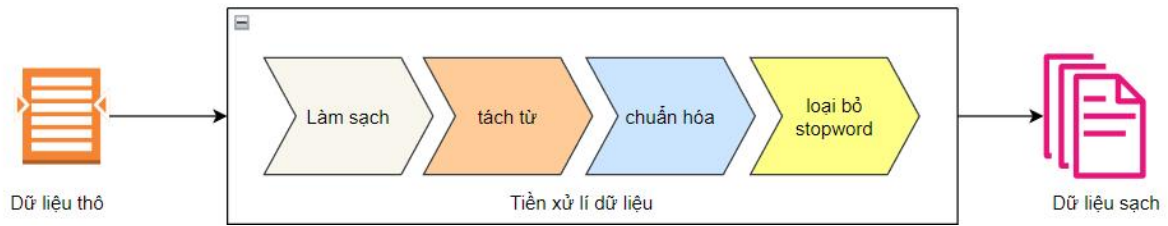
2.4. FastAPI

FastAPI là một Framework hiện đại, hiệu năng cao để xây dựng Web APIs với ngôn ngữ lập trình Python dựa trên tiêu chuẩn Python type hints.

Trong đề tài này, FastAPI đảm nhận vai trò làm máy chủ phục vụ cho hệ thống phân loại ý kiến của khách hàng. Khi khách hàng thêm ý kiến đánh giá sản phẩm, ý kiến này sẽ được gửi đến FastAPI Server thông qua giao thức HTTP, sau đó ý kiến sẽ được đưa vào mô hình máy học đã huấn luyện để phân lớp, đầu ra là ý kiến cùng nhãn của ý kiến, kết quả này được lưu vào cơ sở dữ liệu, đồng thời gửi về client để hiển thị kết quả.

2.5. Tiền xử lý dữ liệu

Xử lý ngôn ngữ tự nhiên (NLP) là một nhánh của trí tuệ nhân tạo tập trung vào các ứng dụng dựa trên ngôn ngữ tự nhiên của con người. Bước đầu trong xử lý ngôn ngữ tự nhiên là tiền xử lý dữ liệu, vì văn bản trong ngôn ngữ tự nhiên không tuân theo một quy chuẩn cố định nào, gây cản trở cho việc nhận biết của các chương trình đầu là dữ liệu có giá trị, đâu là dữ liệu vô nghĩa, không giá trị. Quy trình tiền xử lý dữ liệu được thực hiện qua các bước là làm sạch dữ liệu, tách từ theo các từ đơn hoặc cặp từ có ý nghĩa, chuẩn hóa từ theo quy chuẩn đồng nhất, loại bỏ từ dừng (stopword) được thể hiện qua Hình 2.2 dưới đây.



Hình 2.2 Mô tả quy trình tiền xử lý dữ liệu.

- **Làm sạch dữ liệu:** Dữ liệu thô ban đầu sau khi được thu thập không tránh khỏi việc chứa các ký tự không mang ý nghĩa hoặc các dòng bị thiếu, không có giá trị. Vì vậy bước đầu tiên trong tiền xử lý dữ liệu là làm sạch dữ liệu thô này để loại bỏ các dòng trống, các ký tự vô nghĩa.

Ví dụ: một vài dòng dữ liệu thô trước khi được làm sạch.

Văn bản	nhãn
Sau 1 tháng sd. Tốt : màn hình đẹp, mượt. Âm thanh ok. Camera đẹp.	4
Máy đẹp pin trâu camera đẹp nhân viên tư vấn nhiệt tình quá tuyệt vời	5
Cam xấu, cấu hình tạm đc Ko bang may con oppo reamel xiaomi, noi chung chỉ mua đt này cho người già thôi	(không có)

Sau khi làm sạch, văn bản được làm sạch các ký tự đặc biệt, đối với dòng dữ liệu bị thiếu nhãn đã được loại bỏ, cuối cùng thu được kết quả dưới đây.

Văn bản	nhãn
Sau tháng sd Tốt màn hình đẹp mượt Âm thanh ok Camera đẹp	4
Máy đẹp pin trâu camera đẹp nhân viên tư vấn nhiệt tình quá tuyệt vời 5	5

- **Tách từ:** Ngôn ngữ tiếng việt bao gồm từ đơn và từ ghép, vì thế cần phải tách những từ, cụm từ này để cho mô hình máy học hiểu được đâu là từ đơn, đâu là từ ghép có nghĩa.

Ví dụ: câu gốc: “Điều quý giá nhất không phải là thời gian, mà là cách bạn sử dụng nó”

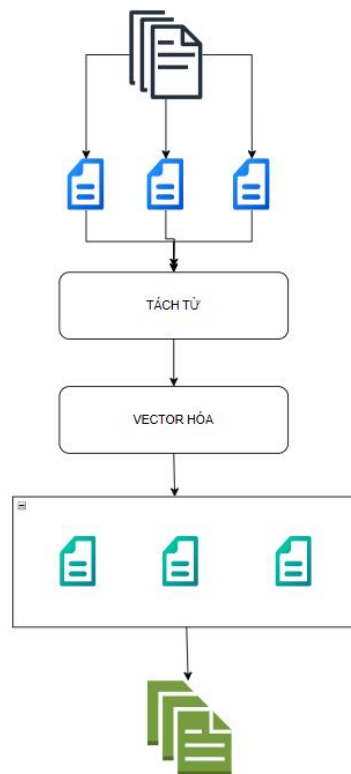
Kết quả tách từ: Điều quý_giá nhất không_phải là thời_gian, mà là cách bạn sử_dụng nó

- **Chuẩn hóa dữ liệu:** Vì máy tính hiểu chữ viết hoa và chữ viết thường là hai từ khác nhau, nên cần đưa dữ liệu về đồng nhất một dạng và thông thường được đưa về dạng dữ liệu viết thường. Mục đích chuẩn hóa dữ liệu là để giảm số lượng đặc trưng và tăng độ chính xác cho mô hình máy học.
- **Loại bỏ từ dừng (stopword):** Từ dừng (stopword) là những từ xuất hiện nhiều trong ngôn ngữ tự nhiên nhưng các từ này không mang nhiều ý nghĩa. Trong tiếng Việt, từ dừng là các từ thường được sử dụng như: “là”, “kìa”,

“ơ”,... Và trong đề tài này, việc loại bỏ từ dừng được thực hiện bằng cách lọc văn bản qua một bộ từ điển được xây dựng riêng cho chủ đề đánh giá ý kiến của khách hàng với các mặt hàng nội thất.

2.6. PhoBert

PhoBert (Vietnamese Bidirection Encoder Representation for Transformers) là một mô hình ngôn ngữ được huấn luyện trước (pre-trained language model) dành riêng cho tiếng Việt. Nó dựa trên kiến trúc RoBERTa, một phiên bản cải tiến của BERT được nghiên cứu và phát triển tại VinAI Research. PhoBERT được thiết kế để hiểu và xử lý ngôn ngữ tiếng Việt tốt hơn, hỗ trợ nhiều bài toán trong xử lý ngôn ngữ tự nhiên (NLP). PhoBERT có hai phiên bản chính là PhoBERT-base dành cho các bài toán yêu cầu nhanh, nhẹ, hiệu suất cao và PhoBERT-large có mô hình lớn hơn, mạnh mẽ hơn, phù hợp cho các bài toán yêu cầu phức tạp. Đề tài này sử dụng phiên bản PhoBERT-base, dùng để vector hóa các văn bản trong tập dữ liệu được mô tả qua sơ đồ Hình 2.3 dưới đây.



Hình 2.3 Vai trò của PhoBert trong hệ thống.

Tập dữ liệu sẽ được chia nhỏ thành nhiều lô (batches), mỗi lô sẽ được đưa vào xử lý riêng để tránh gặp các vấn đề về bộ nhớ. Mỗi đoạn văn bản có trong lô sẽ được tách từ bằng tokenizer của PhoBERT-base. Sau khi tách từ xong, đoạn văn bản đã được tách từ sẽ được chuẩn hóa thành các vector thông qua mô hình được huấn luyện sẵn của PhoBERT-base. Khi hoàn thành tất cả các lô, các lô đã được xử lý sẽ được ghép nối lại như ban đầu và trả về kết quả là tập dữ liệu đã được vector hóa.

2.7. Logistic Regression

Mô hình Logistic Regression [3] thường được sử dụng cho các bài toán phân loại nhị phân (2 lớp) nhưng vẫn được mở rộng cho các bài toán phân loại đa lớp thông

qua các chiến lược “**One-vs-Rest**” (OvR) hoặc “**Softmax Regresstion**” ánh xạ xác suất của từng lớp.

2.8. SVM (Support Vector Machine)

SVM [4] (Support Vector Machine) là 1 thuật toán học máy thuộc nhóm Supervised Learning (học có giám sát) được sử dụng trong các bài toán phân lớp dữ liệu (classification) hay hồi quy (Regression). SVM là mô hình xây dựng 1 siêu phẳng hoặc 1 tập hợp các siêu phẳng trong 1 không gian nhiều chiều hoặc vô hạn chiều, có thể được sử dụng cho phân loại, hồi quy, hoặc các nhiệm vụ khác. Để phân loại tốt nhất thì phải xác định siêu phẳng (Optimal hyperplane) nằm ở càng xa các điểm dữ liệu của tất cả các lớp (Hàm lề) càng tốt, vì nói chung lề càng lớn thì sai số tổng quát hóa của thuật toán phân loại càng bé.

2.9. Random Forest

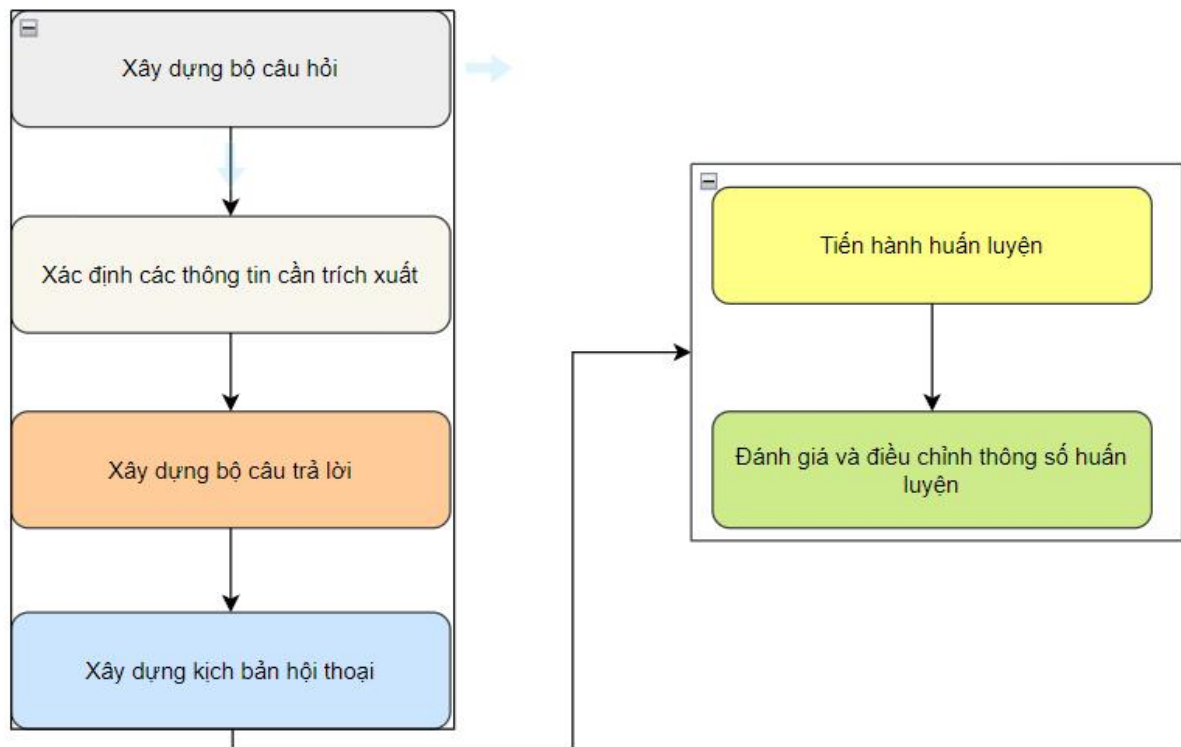
Random Forest [5] là một thuật toán học máy mạnh mẽ thuộc nhóm ensemble learning, tức là kết hợp nhiều mô hình học máy đơn lẻ (các cây quyết định) để tạo ra một mô hình mạnh mẽ hơn. Random Forest thường được sử dụng cho các bài toán phân loại và hồi quy. Cách tiếp cận của nó giúp giảm thiểu hiện tượng overfitting (quá khớp) mà một cây quyết định đơn lẻ có thể gặp phải.

2.10. Chatbot hỗ trợ khách hàng

Rasa là một framework mã nguồn mở dùng để xây dựng chatbot hội thoại với nhiều sự hỗ trợ mạnh mẽ ngay cả khi dữ liệu cung cấp ít. Rasa gồm hai thành phần chính là Rasa NLU [6] (Natural Language Understanding) và Rasa Core dùng để quản lý luồng hội thoại.

- **Rasa NLU:** là một module đóng vai trò xử lý đầu vào câu hội thoại của người dùng (tokenize, features), xác định ý định người dùng (intent classification) và trích chọn thực thể (entity extraction). Ngoài ra, Rasa NLU còn có các tính năng khác như Regular Expression, Lookup table,...
- **Rasa Core:** Là nơi quản lý luồng hội thoại, sau khi Rasa NLU xác định được các ý định (intent) của người dùng, Rasa Core sẽ tiếp nhận phân thông tin đó để quyết định các hành động tiếp theo cũng như thông điệp phản hồi.

Trong đề tài này, chatbot hỗ trợ khách hàng trong việc mua sắm các sản phẩm trên website được xây dựng với Rasa NLU thực hiện thông qua các bước được mô tả qua Hình 2.4.

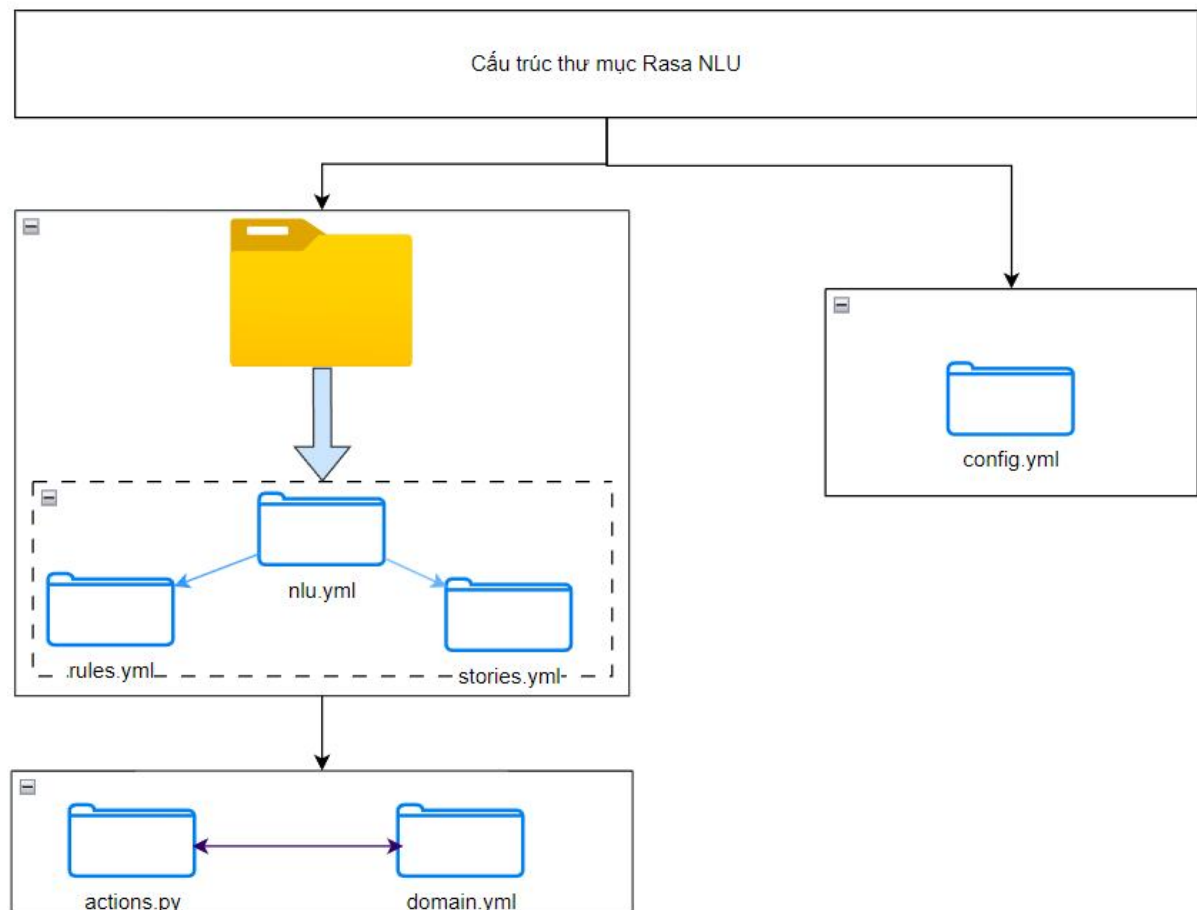


Hình 2.4 Mô tả các bước xây dựng chatbot với Rasa NLU

- **Xây dựng bộ câu hỏi:** Bộ câu hỏi được xây dựng dựa trên một phạm vi, chủ đề mà chatbot hướng tới để giải quyết các yêu cầu của người dùng. Cụ thể trong đề tài này, bộ câu hỏi xoay quanh các câu hỏi của khách hàng về các mặt hàng nội thất và các yêu cầu hỗ trợ liên quan.
- **Xác định thông tin cần trích xuất:** Trích xuất thông tin từ câu hỏi của người dùng.
- **Xây dựng bộ câu trả lời:** Dựa trên câu hỏi và các thông tin trích xuất được mà từ đó, chatbot có thể xác định được ý định của người dùng, qua đó chatbot đưa ra các câu trả lời phù hợp theo ý định đã xác định được.
- **Xây dựng kịch bản hội thoại:** Để chatbot có thể giao tiếp gần gũi và phù hợp với nhiều trường giao tiếp với nhiều kịch bản khác nhau.
- **Tiến hành huấn luyện:** Huấn luyện chatbot với pipeline của Rasa NLU.
- **Đánh giá và điều chỉnh thông số huấn luyện:** Sau khi huấn luyện, thu được kết quả huấn luyện, qua đó tiến hành điều chỉnh thông số cho phù hợp.

Cấu trúc thư mục dự án của Rasa NLU được xây dựng sẵn để hỗ trợ việc xây dựng chatbot dễ dàng hơn được mô tả bằng Hình 2.5 bên dưới. Vai trò của từng tập tin được xác định như sau:

- **nlu.yml:** khai báo các câu hỏi và ý định tương ứng
- **rules.yml:** quy định các hành động theo ý định
- **stories.yml:** quy định các kịch bản hội thoại
- **actions.py:** các câu trả lời được tùy chỉnh theo trường hợp cụ thể và cần truy vấn.
- **domain.yml:** khai báo các ý định, thực thể, câu trả lời.
- **config.yml:** định nghĩa các tham số huấn luyện trong pipeline của Rasa NLU.



Hình 2.5 Cấu trúc thư mục dự án Rasa NLU

2.11. Các chỉ số đánh giá

2.11.1. Confusion Matrix

Là một phương pháp đánh giá kết quả của những bài toán phân loại với việc xem xét cả những chỉ số về độ chính xác và độ bao quát của các dự đoán cho từng lớp. Một confusion maxtrix cho bài toán phân loại hai lớp gồm bốn chỉ số với mỗi lớp phân loại được mô tả qua Hình 2.6 bên dưới.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Hình 2.6 Ma trận nhầm lẫn cho bài toán phân loại hai lớp

- **TP (True Positive):** số lượng dự đoán chính xác một mẫu thuộc lớp Positive.
- **FP (False Positive):** số lượng dự đoán sai một mẫu thuộc lớp Negative là Positive.

- **FN (False Negative):** số lượng dự đoán sai một mẫu thuộc lớp Positive là Negative.
- **TN (True Negative):** số lượng dự đoán đúng một mẫu thuộc lớp Negative.

Confusion maxtrix (C) cho k lớp

Dự đoán	1	...	K
1			
...			
k			

- $C[i,j]$: số phần tử lớp i (dòng) được giải thuật dự đoán là lớp j (cột)
- $C[i,i]$: số phần tử phân lớp đúng
- Độ chính xác lớp i: $C[i,i] / C[i,]$
- Độ chính xác tổng thể: $\sum C[i,i] / C$

2.11.2. Precision và Recall

Precision được định nghĩa là tỉ lệ số mẫu Positive mô hình dự đoán đúng trên tổng số mẫu mô hình dự đoán là Positive, Precision được tính theo công thức (1).

$$\text{Precision} = \frac{TP}{(TP+FP)} \quad (1)$$

Precision càng cao, tức là số mẫu mô hình dự đoán là positive đều là positive càng nhiều. Precision = 1, tức là tất cả số mẫu mô hình dự đoán là Positive đều đúng, hay không có mẫu nào có nhãn là Negative mà mô hình dự đoán nhầm là Positive.

Recall được định nghĩa là tỉ lệ số điểm Positive mô hình dự đoán đúng trên tổng Số mẫu thật sự là Positive (hay tổng số điểm được gán nhãn là Positive ban đầu). Recall được tính theo công thức (2).

$$\text{Recall} = \frac{TP}{(TP+FN)} \quad (2)$$

Recall càng cao, tức là số mẫu là positive bị bỏ sót càng ít. Recall = 1, tức là tất cả số mẫu có nhãn là Positive đều được mô hình nhận ra.

2.11.3. F1 Score

F1-score là trung bình điều hòa (harmonic mean) của precision và recall (giả sử hai đại lượng này khác 0). F1-score được tính theo công thức (3).

$$\text{F1} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

2.11.4. Accuracy

Accuracy là độ chính xác được tính bằng tỉ lệ giữa số mẫu được dự đoán đúng trên tổng số mẫu trong tập dữ liệu. Accuracy được tính theo công thức (4).

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

2.11.5. Cross entropy loss

Đánh giá độ mất mát của mô hình Rasa.

- Hàm lỗi của phân loại ý định.

$$L_{intent} = - \sum_{i=1}^N \sum_{j=1}^C y_{ij} \log(y'_{ij}) \quad (5)$$

- Hàm lỗi cho nhận dạng thực thể.

$$L_{entity} = - \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^T y_{ijk} \log(y'_{ijk}) \quad (6)$$

Công thức tổng hợp:

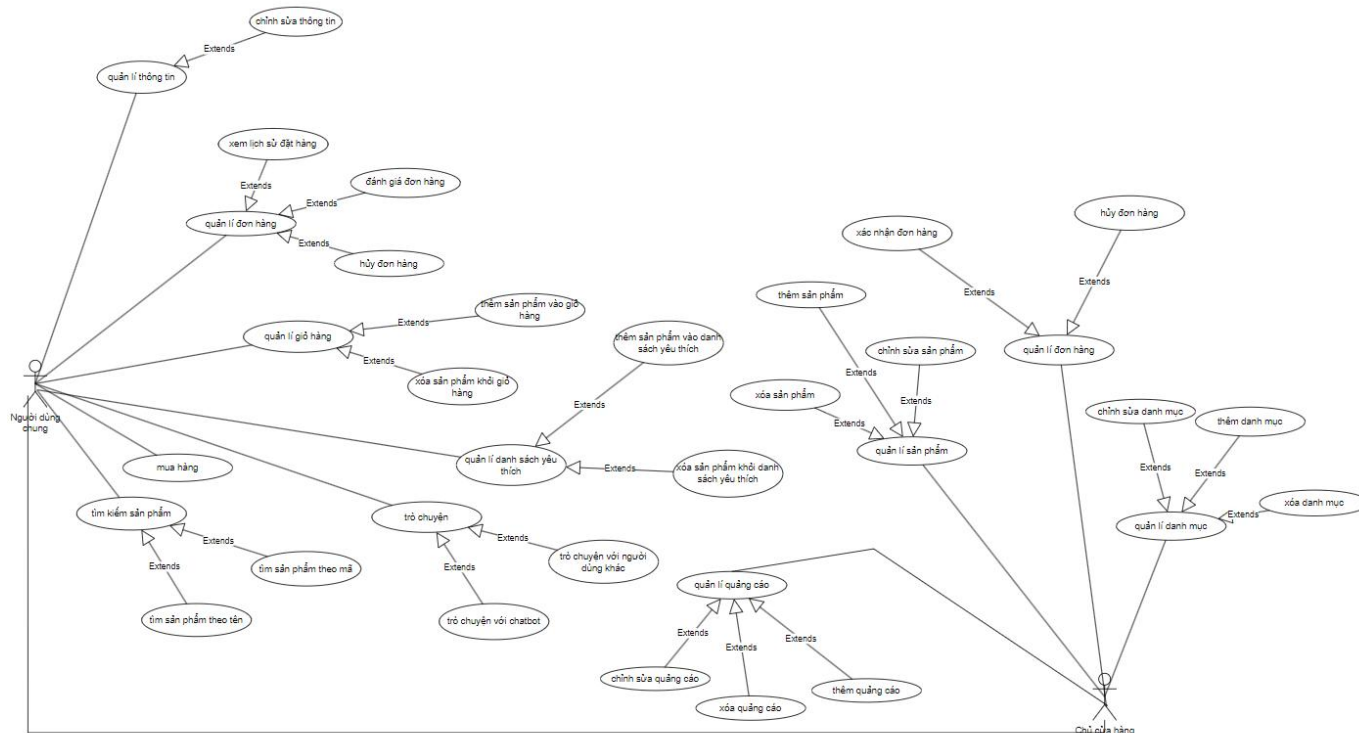
$$L_{total} = L_{intent} + L_{entity} \quad (7)$$

CHƯƠNG III: CÀI ĐẶT GIẢI PHÁP

1. Thiết kế cơ sở dữ liệu và xây dựng bộ API cho website.

1.1. Sơ đồ Usecase tổng quát

Sơ đồ Usecase tổng quát (xem Hình 3.1) được mô tả như sau:

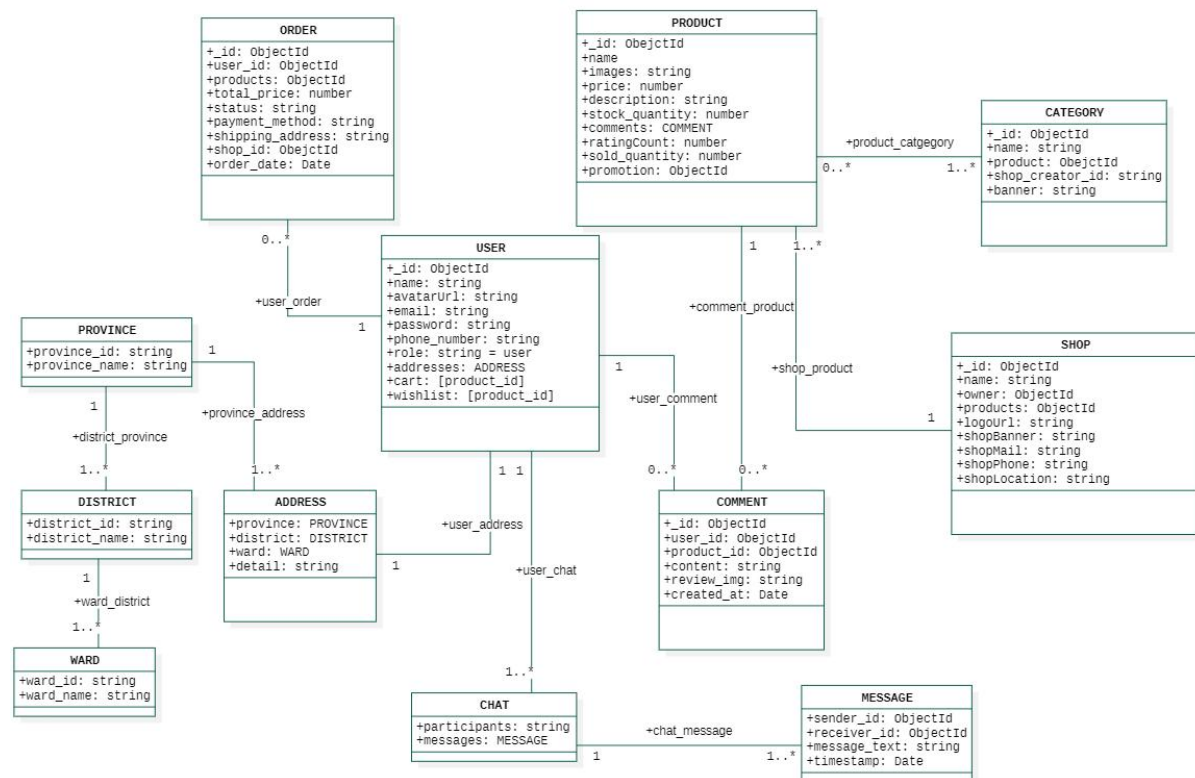


Hình 3.1 Mô tả sơ đồ Usecase tổng quát

Diễn giải sơ đồ Usecase tổng quát

- ❖ Đối với người dùng chưa đăng nhập hoặc chưa có tài khoản: Có thể xem sản phẩm, tìm kiếm sản phẩm.
- ❖ Đối với người dùng đã đăng nhập vào website:
 - **Khách hàng thông thường:** Có thể thêm sản phẩm vào giỏ hàng, thêm sản phẩm vào danh sách yêu thích, mua sản phẩm, xem lịch sử đặt hàng, trạng thái đơn hàng, trò chuyện với chủ cửa hàng, tương tác với trợ lý ảo chatbot.
 - **Chủ cửa hàng:** Bao gồm các chức năng của người dùng thông thường, quản lý cửa hàng, quản lý sản phẩm, quản lý danh mục sản phẩm, quản lý chương trình quảng cáo.

1.2. Mô hình dữ liệu mức quan niệm (CDM)



Hình 3.2 Sơ đồ mô hình dữ liệu mức quan niệm

Diễn giải mô hình luận lý CDM

- Một người dùng có thể có không hoặc nhiều đơn hàng, một đơn hàng duy nhất một người dùng.
- Một người dùng có duy nhất một địa chỉ, một địa chỉ có thể có một hoặc nhiều người dùng.
- Một người dùng có một hoặc nhiều cuộc hội thoại (chat), một hội thoại chỉ duy nhất một người dùng.
- Một người dùng có không hoặc nhiều đánh giá, một đánh giá chỉ duy nhất thuộc về một người dùng.
- Một sản phẩm có thể không hoặc nhiều đánh giá, một đánh giá chỉ thuộc về duy nhất một sản phẩm.
- Một danh mục có thể không có hoặc nhiều sản phẩm, Một sản phẩm có thể nằm trong một hoặc nhiều danh mục.
- Một cửa hàng có thể có nhiều sản phẩm, một sản phẩm chỉ duy nhất một cửa hàng.

1.3. Xây dựng thực thể

Vì cơ sở dữ liệu sử dụng hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL) MongoDB, nên các thuộc tính sẽ có các kiểu dữ liệu thuộc MongoDB, đồng thời bản thân thực thể cũng là một kiểu dữ liệu tham chiếu của một hoặc nhiều thuộc tính trong các thực thể khác.

1.3.1. Thực thể User (người dùng)

Bảng 3.1 Bảng mô tả thực thể User

Tên thuộc tính	Kiểu dữ liệu	Diễn giải
id	ObjectId	Định danh duy nhất của tài liệu
name	String	Tên user
avatarUrl	String	Địa chỉ ảnh đại diện của user
password	String	Mật khẩu của user
role	String	Vai trò của user bao gồm “user” là mặc định, “owner” đối với user là chủ cửa hàng.
address	Address	Địa chỉ của user, kiểu dữ liệu tham chiếu với thực thể Address
wishlist	[Product]	Danh sách sản phẩm yêu thích, kiểu dữ liệu tham chiếu đến thuộc tính _id của thực thể Product
cart	[Product]	Danh sách sản phẩm đã thêm vào giỏ hàng, kiểu dữ liệu tham chiếu đến thuộc tính _id thực thể Product
Shop_id	ObejctId	Cửa hàng mà user sở hữu, kiểu dữ liệu tham chiếu đến thuộc tính _id của thực thể Shop

1.3.2. Thực thể Shop (cửa hàng)

Bảng 3.2 Bảng mô tả thực thể Shop

Tên thuộc tính	Kiểu dữ liệu	Diễn giải
id	ObejctId	Định danh duy nhất của tài liệu
Name	String	Tên Shop
Owner	ObejctId	Chủ shop, tham chiếu đến thuộc tính _id của thực thể User
logoUrl	String	Đường dẫn hình ảnh logo của Shop
shopBanner	String	Đường dẫn hình banner của Shop
shopMail	String	Địa chỉ email
shopPhone	String	Số điện thoại Shop
shopLocation	String	Địa chỉ Shop
createdAt	Date	Ngày thành lập Shop

1.3.3. Thực thể Product (sản phẩm)

Bảng 3.3 Bảng mô tả thực thể Product

Tên thuộc tính	Kiểu dữ liệu	Diễn giải
id	ObejctId	Định danh duy nhất của tài liệu
Name	String	Tên sản phẩm
Sku	String	Mã sản phẩm
Images	[string]	Danh sách đường dẫn hình ảnh sản phẩm
Price	Number	Giá sản phẩm
Description	String	Mô tả sản phẩm
Stock_quantity	Number	Số lượng trong kho
Comments	[Comment]	Các đánh giá của sản phẩm

ratingCount	Number	Số lượt đánh giá
Sold_quantity	Number	Số lượng đã bán
Shop_owner_id	ObjectId	Tham chiếu đến thuộc tính _id của Shop, thể hiện sản phẩm thuộc một Shop cụ thể
Promotion	[Promotion]	Các chương trình quảng cáo của sản phẩm

1.3.4. Thực thể Order (đơn hàng)

Bảng 3.4 Bảng mô tả thực thể Order

Tên thuộc tính		Kiểu dữ liệu	Diễn giải
id		ObejectId	Định danh duy nhất của tài liệu
User_id		ObjectId	Id của người mua, tham chiếu đến _id của thực thể User
Products		[Product]	Danh sách sản phẩm trong đơn hàng, tham chiếu đến id của thực thể Product
Total_price		Number	Tổng giá trị đơn hàng
Status		String	Trạng thái đơn hàng
Payment_status		String	Trạng thái thanh toán
Payment_method		String	Phương thức thanh toán
Shipping_adress		String	Địa chỉ nhận hàng
Shop_id		ObejectId	Thể hiện đơn hàng mua của Shop nào, kiểu dữ liệu tham chiếu đến _id của thực thể Shop
Order_date		Date	Thời gian đặt hàng

1.3.5. Thực thể Category (danh mục)

Bảng 3.5 Bảng mô tả thực thể Category

Tên thuộc tính	Kiểu dữ liệu	Diễn giải
id	ObejectId	Định danh duy nhất của tài liệu
Name	String	Tên danh mục
Product	[Product]	Danh sách sản phẩm của danh mục, kiểu dữ liệu tham chiếu tới thuộc tính _id của thực thể Product
Shop_creator_id	ObejectId	Cửa hàng đã tạo danh mục, kiểu dữ liệu tham chiếu đến id của thực thể Shop
Banner	String	Đường dẫn hình ảnh banner của danh mục

1.3.6. Thực thể Chat (hội thoại)

Bảng 3.6 Bảng mô tả thực thể Chat

Tên thuộc tính	Kiểu dữ liệu	Diễn giải
id	ObjectId	Định danh duy nhất của tài liệu
Participants	[ObjectId]	Người tham gia hội thoại, tham chiếu đến _id của thực thể User
Message	[Message]	Nội dung hội thoại, tham chiếu đến _id của thực thể Message

1.3.7. Thực thể Comment (đánh giá)

Bảng 3.7 Bảng mô tả thực thể Comment

Tên thuộc tính	Kiểu dữ liệu	Diễn giải
_id	ObjectId	Định danh duy nhất của tài liệu
User_id	ObejctId	Người đánh giá, tham chiếu đến _id của thực thể User
Product_id	ObejctId	Sản phẩm được đánh giá, tham chiếu đến _id của thực thể Product
Content	String	Nội dung đánh giá
Review_img	[string]	Danh sách đường dẫn hình ảnh kèm theo nếu có khi đánh giá
Created_at	Date	Thời gian đánh giá

1.3.8. Thực thể Message (thông điệp)

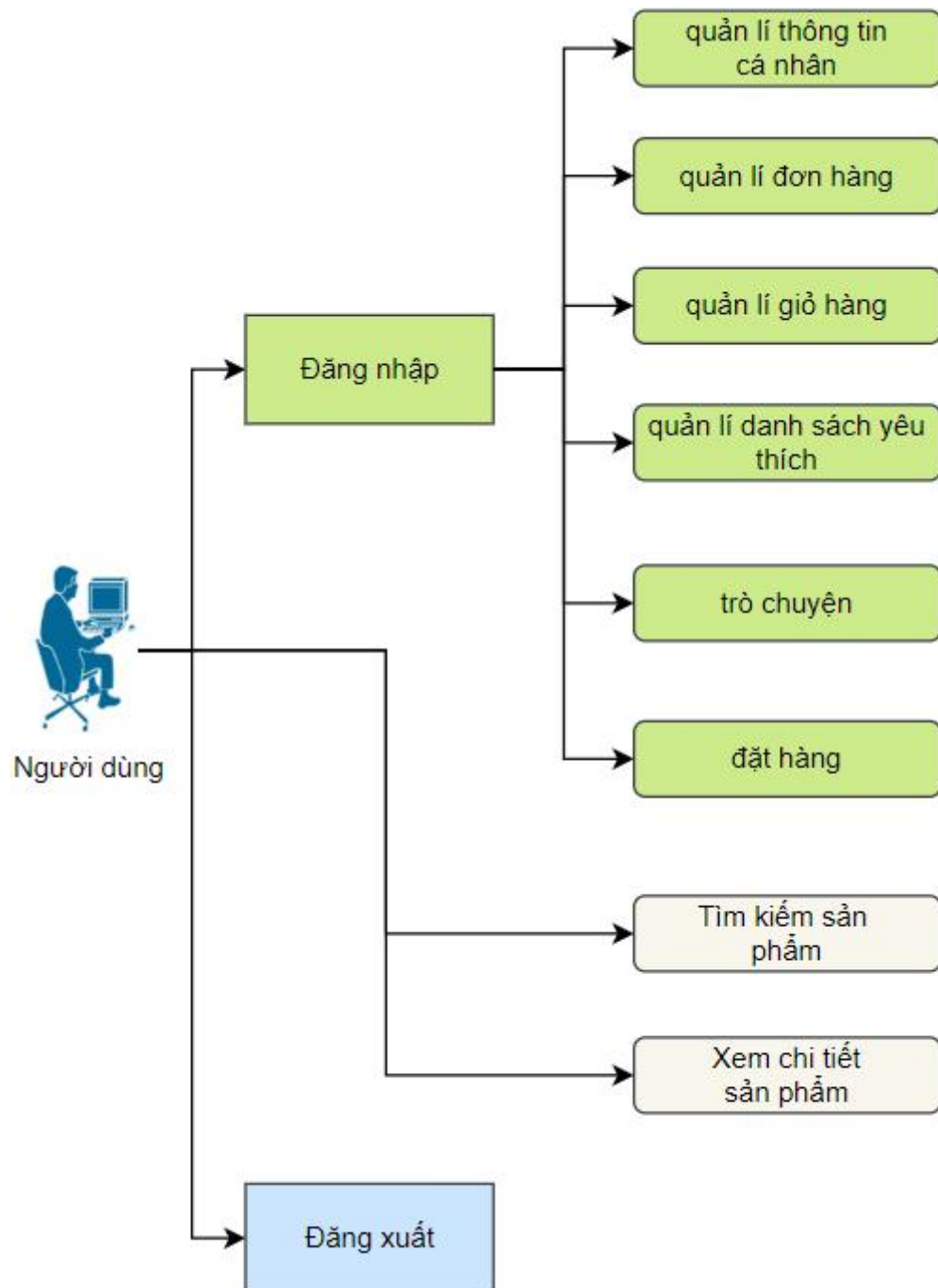
Bảng 3.8 Bảng mô tả thực thể Message

Tên thuộc tính	Kiểu dữ liệu	Diễn giải
_id	ObejctId	Định danh duy nhất của tài liệu
Sender_id	ObejctId	Id của người gửi, tham chiếu đến _id của thực thể User
Receiver_id	ObejctId	Id của người
message_text	String	Thông điệp
Timestamp	Date	Thời gian thông điệp được gửi đi

1.4. Thiết kế mô hình chức năng

1.4.1. Các chức năng dành cho nhóm người dùng chung

Các chức năng dành cho nhóm người dùng chung được mô tả qua Hình 3.3 dưới đây.

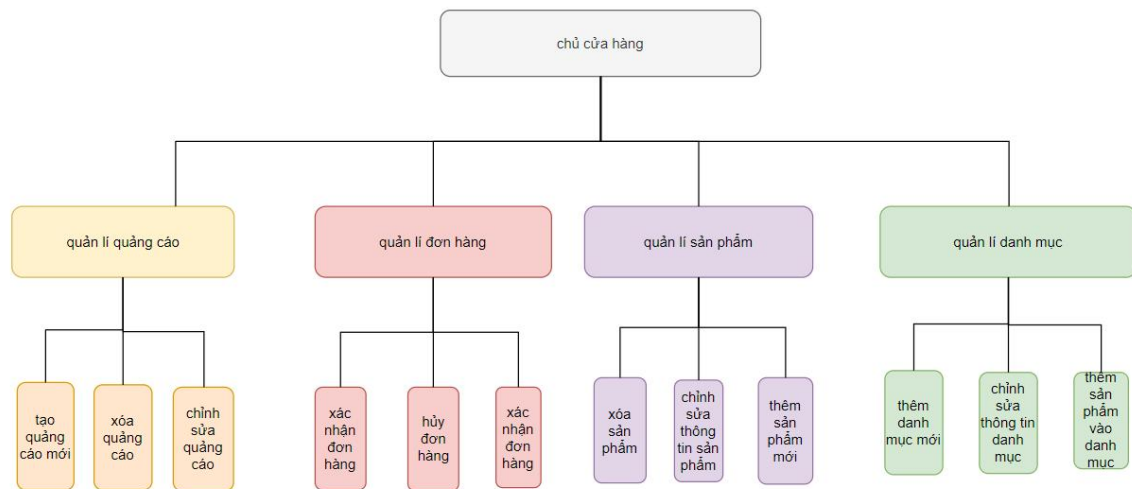


Hình 3.3 Mô tả sơ đồ chức năng dành cho nhóm người dùng chung

- Mỗi người dùng khi vào website đều có thể tìm kiếm sản phẩm và xem sản phẩm chi tiết.
- Với người dùng đã đăng nhập, người dùng có thể tiến hành đặt hàng, trò chuyện với chủ cửa hàng, quản lý thông tin cá nhân, quản lý giỏ hàng và danh sách yêu thích.

1.4.2. Chức năng dành cho nhóm người dùng là chủ cửa hàng.

Đối với chủ cửa hàng, bao gồm các chức năng của nhóm người dùng chung và các chức năng dùng để quản lý cửa hàng được miêu tả qua Hình 3.4 bên dưới.



Hình 3.4 Mô tả sơ đồ chức năng dành cho chủ cửa hàng

- Quản lý quảng cáo: Thêm, xóa, sửa thông tin quảng cáo.
- Quản lý đơn hàng: Xác nhận, hủy, thay đổi trạng thái đơn hàng.
- Quản lý sản phẩm: Thêm, xóa, chỉnh sửa thông tin sản phẩm.
- Quản lý danh mục: Thêm, xóa, chỉnh sửa thông tin danh mục.

1.5. Xây dựng API cho website

Xây dựng bộ API để cung cấp thông tin cho website sàn thương mại điện tử cung cấp các thông tin về sản phẩm, thông tin người dùng, đơn hàng.

1.5.1. Hàm lấy danh sách tất cả sản phẩm

Mô tả: Lấy danh sách thông tin tất cả sản phẩm hiện có trong cơ sở dữ liệu.

Bảng 3.9 API chức năng lấy danh sách tất cả sản phẩm

Request URL	http:serverdomain/product/getAll
Request Method	GET
Request Header	Content-Type: Application/json;
Paramer	

Thông điệp phản hồi: Chuỗi JSON chứa danh sách sản phẩm và thông tin chi tiết của sản phẩm.

Bảng 3.10 Kết quả trả về của API lấy danh sách tất cả sản phẩm

Trường	Kết quả trả về
Header status	200
Header ContentType	Application/json; charset=UTF-8
Json data	Danh sách sản phẩm và thông tin chi tiết từng sản phẩm

1.5.2. Hàm lấy thông tin người dùng

Mô tả: Lấy thông tin người dùng cụ thể được xác định bằng id của người dùng.

Bảng 3.11 API chức năng lấy thông tin người dùng

Request URL	http:serverdomain/getInfo/:userId
Request Method	GET
Request Header	Content-Type: Application/json;
Paramer	userId - id của người dùng

Thông điệp phản hồi: chuỗi JSON chứa thông tin của người dùng.

Bảng 3.12 Kết quả trả về của API lấy thông tin người dùng

Trường	Kết quả trả về
Header status	200
Header ContentType	Application/json; charset=UTF-8
Json data	Thông tin của người dùng

1.5.3. Hàm lấy danh sách đơn hàng của người dùng

Mô tả: Lấy danh sách đơn hàng của một người dùng được xác định bằng id của người dùng.

Bảng 3.13 API chức năng lấy danh sách đơn hàng của người dùng

Request URL	http:serverdomain/order/get/:userId
Request Method	GET
Request Header	Content-Type: Application/json;
Paramer	userId - id của người dùng

Thông điệp phản hồi: Chuỗi JSON chứa danh sách đơn hàng của người dùng.

Bảng 3.14 Kết quả trả về của API lấy danh sách đơn hàng người dùng

Trường	Kết quả trả về
Header status	200
Header ContentType	Application/json; charset=UTF-8
Json data	Danh sách đơn hàng của người dùng

1.5.4. Hàm lấy danh sách quảng cáo của cửa hàng

Mô tả: lấy danh sách các quảng cáo hiện có của một cửa hàng được xác định bằng id của cửa hàng.

Bảng 3.15 API chức năng lấy danh sách quảng cáo của cửa hàng

Request URL	http:serverdomain/promotion/shop/:shopId
Request Method	GET
Request Header	Content-Type: Application/json;
Paramer	shopId - id của cửa hàng

Thông điệp phản hồi: Chuỗi JSON chứa danh sách các chương trình quảng cáo của cửa hàng.

Bảng 3.16 Kết quả trả về của API lấy danh sách quảng cáo cửa hàng

Trường	Kết quả trả về
Header status	200

Header ContentType	Application/json; charset=UTF-8
Json data	Danh sách các chương trình quảng cáo của cửa hàng

1.5.5. Hàm tạo sản phẩm mới

Mô tả: Cửa hàng tạo một sản phẩm mới.

Bảng 3.17 API chức năng tạo sản phẩm mới

Request URL	http:serverdomain/product/create
Request Method	POST
Request Header	Content-Type: Application/json;
formData	Product_name Sku Product_images Description Product_quantity Shop_owner_id

Thông điệp phản hồi: Chuỗi JSON đoạn thông báo thông điệp tạo sản phẩm mới thành công.

Bảng 3.18 Kết quả trả về của API tạo sản phẩm mới

Trường	Kết quả trả về
Header status	200
Header ContentType	Application/json; charset=UTF-8
Json data	thông điệp thông báo tạo sản phẩm mới thành công

1.5.6. Hàm tạo danh mục sản phẩm mới

Mô tả: Cửa hàng tạo danh mục sản phẩm mới cho cửa hàng đó.

Bảng 3.19 API chức năng tạo danh mục sản phẩm

Request URL	http:serverdomain/category/create
Request Method	POST
Request Header	Content-Type: Application/json;
Form Data	Category_name Banner_url [Product_id] Shop_id

Thông điệp phản hồi: Chuỗi JSON chứa thông điệp thông báo tạo danh mục sản phẩm mới thành công.

Bảng 3.20 Kết quả trả về của API tạo danh mục sản phẩm mới

Trường	Kết quả trả về
Header status	200
Header ContentType	Application/json; charset=UTF-8
Json data	Thông điệp thông báo tạo danh mục sản phẩm mới thành công

1.5.7. Hàm tạo đơn đặt hàng

Mô tả: người dùng tạo đơn đặt hàng với các sản phẩm đã chọn.

Bảng 3.21 API tạo đơn đặt hàng

Request URL	http:serverdomain/order/create
Request Method	POST
Request Header	Content-Type: Application/json;
Form Data	User_id Shipping_address Product_id Quantity Payment_method

Thông điệp phản hồi: Chuỗi JSON chứa thông điệp thông báo đặt hàng thành công.

Bảng 3.22 Kết quả trả về của API tạo đơn đặt hàng

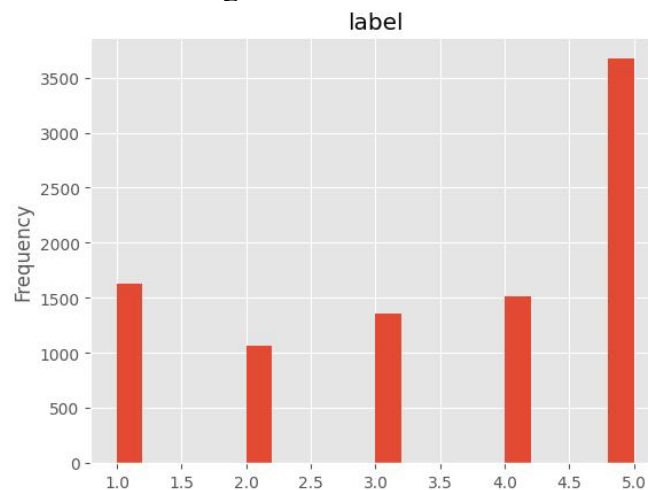
Trường	Kết quả trả về
Header status	200
Header ContentType	Application/json; charset=UTF-8
Json data	Thông báo đặt hàng thành công

2. Xây dựng hệ thống phân loại ý kiến

2.1. Chuẩn bị dữ liệu

Dữ liệu là yếu tố quan trọng cũng là vấn đề cần quan tâm trong quá trình xây dựng một hệ thống phân lớp văn bản.

- Tập dữ liệu được thu thập từ nhiều nguồn bao gồm các đánh giá sản phẩm về mặt hàng nội thất trên sàn thương mại điện tử tiki.vn và lazada.vn sử dụng selenium và BeautifulSoup để cào dữ liệu từ các đánh giá của người dùng và gán nhãn theo số sao tương ứng, kết hợp với tập dữ liệu được đăng tải công khai tại HuggingFace⁵ về phân tích đánh giá của người Việt.
- Kích thước: Tập dữ liệu có tổng cộng 9.238 mẫu dữ liệu.
- Cấu trúc mẫu dữ liệu: bao gồm 2 cột là cột “**comment**” gồm các ý kiến của người dùng và cột “**label**” là nhãn của các ý kiến tương ứng được đánh số từ 1 đến 5 tương ứng với 5 lớp thể hiện 5 cấp độ của khách hàng:
 - ♦ **Nhãn 1:** Rất không hài lòng - 1.628 mẫu.
 - ♦ **Nhãn 2:** Không hài lòng - 1.067 mẫu.
 - ♦ **Nhãn 3:** Bình thường - 1.359 mẫu.
 - ♦ **Nhãn 4:** Khá hài lòng - 1.513 mẫu.
 - ♦ **Nhãn 5:** Rất hài lòng - 3.670 mẫu.

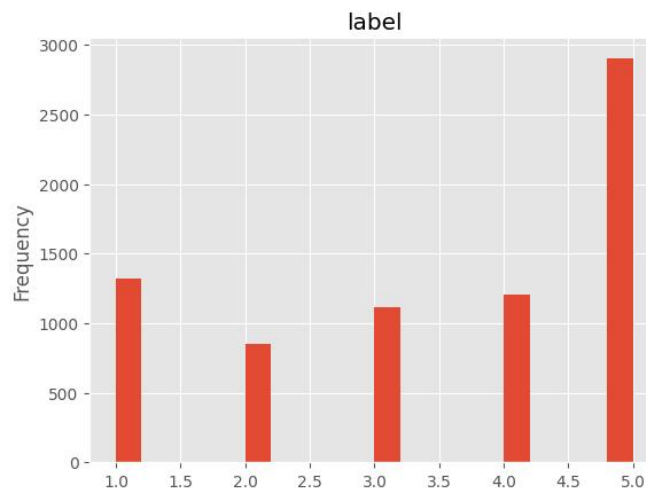


Hình 3.5 Hình minh họa biểu đồ phân bố số lượng mẫu của tập dữ liệu

Sau đó, tập dữ liệu được chia làm 2 phần là tập dữ liệu huấn luyện (80%) và tập dữ liệu dùng để kiểm tra (20%).

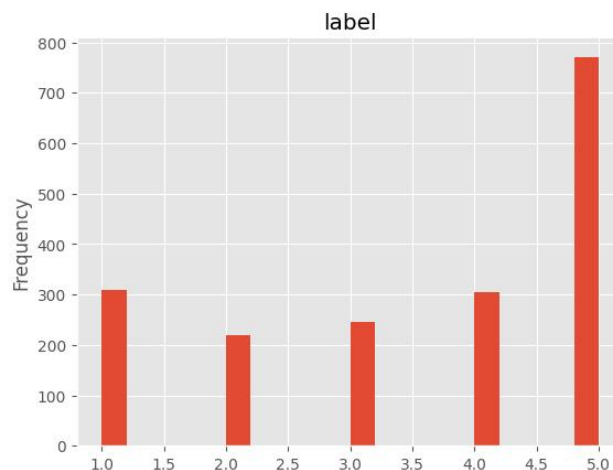
Tập dữ liệu huấn luyện: có 7.390 mẫu dữ liệu, trong đó nhãn 1 có 1.320 mẫu, nhãn 2 có 847 mẫu, nhãn 3 có 114 mẫu, nhãn 4 có 1.209 mẫu, nhãn 5 có 2.900 mẫu.

⁵ Nền tảng chia sẻ, lưu trữ các mô hình trí tuệ nhân tạo - <https://huggingface.co/>



Hình 3.6 Hình minh họa biểu đồ phân bố số lượng mẫu của tập huấn luyện (train)

Tập dữ liệu kiểm tra: có 1.848 mẫu dữ liệu, trong đó nhãn 1 có 308 mẫu, nhãn 2 có 220 mẫu, nhãn 3 có 245 mẫu, nhãn 4 có 304 mẫu, nhãn 5 có 770 mẫu.



Hình 3.7 Hình minh họa biểu đồ phân bố số lượng mẫu của tập kiểm tra (test)

2.2. Vector hóa văn bản.

Trong đề tài này, việc vector hóa văn bản được thực hiện bằng phương pháp text embedding sử dụng PhoBERT (một mô hình BERT được huấn luyện riêng cho tiếng Việt). Phương pháp này giúp biến đổi các đoạn văn bản hoặc các câu thành các vector (mảng số học) với các đặc trưng ngữ nghĩa của nó, từ đó cho phép các mô hình máy học xử lý và phân tích văn bản hiệu quả hơn.

Với PhoBERT, quá trình này thường sử dụng lớp [CLS] token (một token đặc biệt trong BERT) để tạo ra vector biểu diễn cho toàn bộ văn bản được thực hiện qua các bước:

- **Tokenization:** chia nhỏ văn bản thành các token bằng bộ chia từ riêng của PhoBERT.
- **Encoding:** Biến các token thành các chỉ số (IDs) trong từ điển của PhoBERT.
- **Embedding:** Dùng mô hình PhoBERT để tính toán các embeddings cho các token.
- **Vector hóa:** Sử dụng embedding của token [CLS] hoặc tính toán các embeddings từ các lớp khác của PhoBERT để tạo ra một vector cho toàn bộ văn bản.

2.3. Huấn luyện mô hình phân loại

Tập dữ liệu huấn luyện: **X_train_phobert** (tập đặc trưng đầu vào đã được vector hóa, mỗi vector có kích thước 768 chiều).

y_train: tập nhãn tương ứng với mỗi mẫu dữ liệu trong tập huấn luyện.
y_test: tập nhãn tương ứng với mỗi mẫu dữ liệu trong tập kiểm tra.

Tập dữ liệu kiểm tra: **X_test_phobert** (tập đặc trưng đầu vào đã được vector hóa, mỗi vector có kích thước 768 chiều).

2.3.1. Mô hình rừng ngẫu nhiên (Random Forest)

❖ Khởi tạo mô hình rừng ngẫu nhiên

random_forest_model = RandomForestClassifier(random_state=42)

- **RandomForestClassifier** là một thuật toán máy học thuộc họ ensemble methods.
- **RandomForest** tạo ra nhiều cây quyết định (decision tree) từ các tập con ngẫu nhiên của dữ liệu và kết hợp chúng lại thông qua cơ chế bỏ phiếu đa số đối với phân loại.
- **Random_state=42** đảm bảo tính tái lập mô hình bằng cách cố định seed cho việc chọn dữ liệu và tạo cây quyết định ngẫu nhiên.

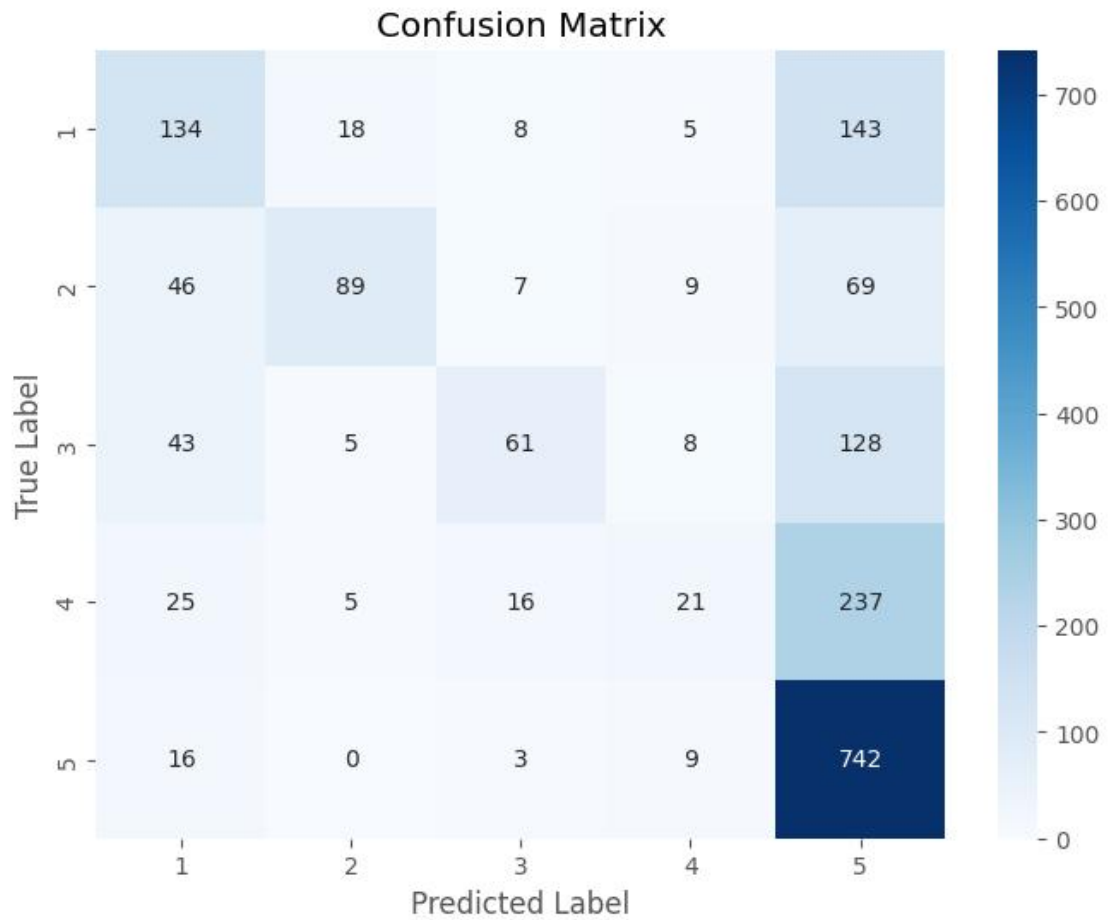
❖ Huấn luyện mô hình trên tập huấn luyện

random_forest_model.fit(X_train_phobert, y_train)

- Chọn ngẫu nhiên các mẫu và đặc trưng: RandomForest tạo ra các tập dữ liệu con (bootstrapping) từ tập dữ liệu ban đầu. Các tập này có thể trùng lặp các mẫu dữ liệu. Mỗi cây chỉ xem xét số lượng nhỏ các đặc trưng (random feature selection), giúp tăng tính ngẫu nhiên và giảm overfitting.
- Xây dựng cây quyết định: Với mỗi dữ liệu con, một cây quyết định được xây dựng bằng cách tìm các điểm cắt tốt nhất trên các đặc trưng (dựa vào giảm impurity như Gini hoặc entropy). Quá trình dừng lại khi đạt độ sâu tối đa hoặc khi không còn dữ liệu để tiếp tục chia.

❖ **Kết quả dự đoán trên tập kiểm tra**

Kết quả dự đoán trên tập kiểm tra được thể hiện bằng ma trận nhầm lẫn (xem hình 3.8)



Hình 3.8 Ma trận nhầm lẫn của mô hình Random Forest

2.3.2. Mô hình SVM (Support Vector Machine)

❖ Khởi tạo mô hình SVM

```
svm_model = SVC(kernel="linear", random_state=42)
```

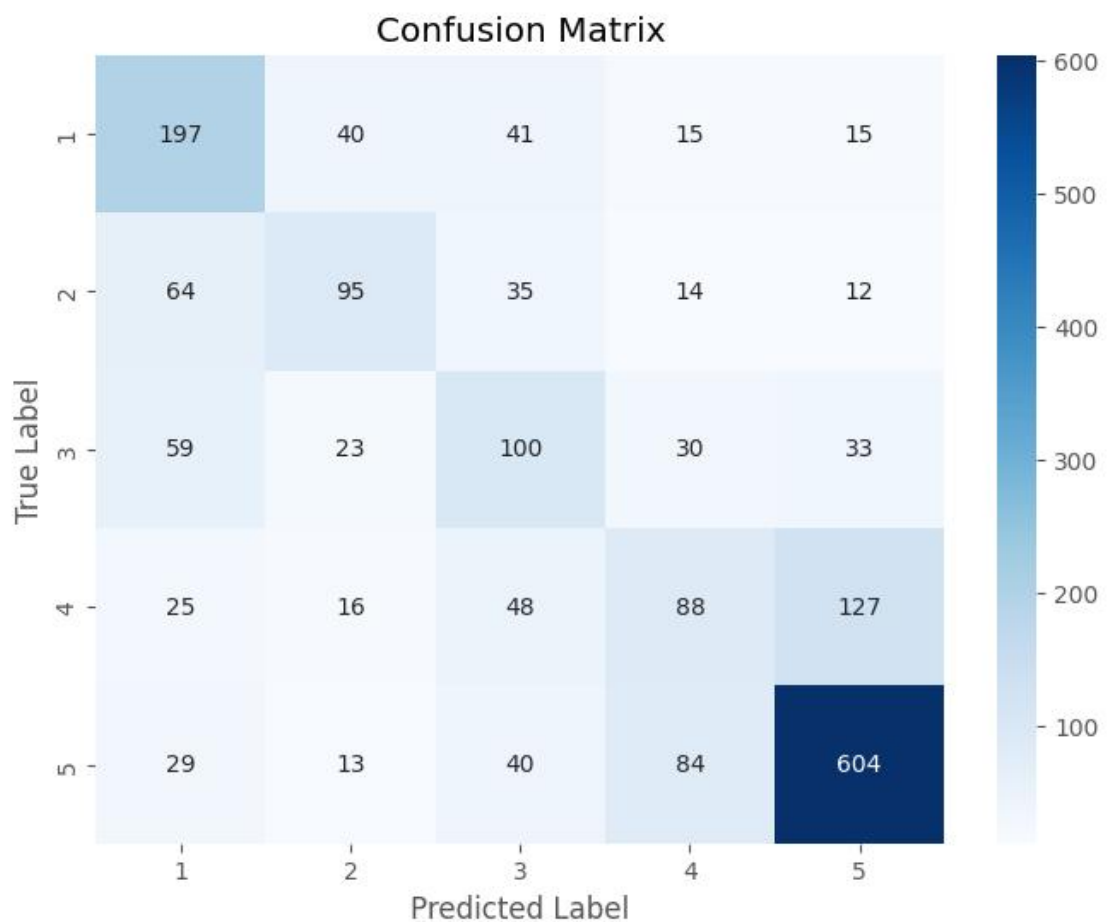
- **SVC:** Hỗ trợ phân loại bằng thuật toán SVC (Support Vector Classifier).
- **Kernel=linear:** sử dụng kernel tuyến tính.
- **random_state=42:** Đảm bảo tính tái lập mô hình.

❖ Huấn luyện mô hình SVM trên tập huấn luyện

```
svm_model.fit(X_train_phobert, y_train)
```

❖ Kết quả dự đoán trên tập kiểm tra

Kết quả dự đoán trên tập kiểm tra được thể hiện bằng ma trận nhầm lẫn (xem hình 3.9)



Hình 3.9 Ma trận nhầm lẫn của mô hình SVM

2.3.3. Mô hình Logistic Regression

❖ Khởi tạo mô hình Logistic Regression

`logistic_model=LogisticRegression(max_iter=1000,random_state=42,
multi_class='multinomial',solver='lbfgs')`

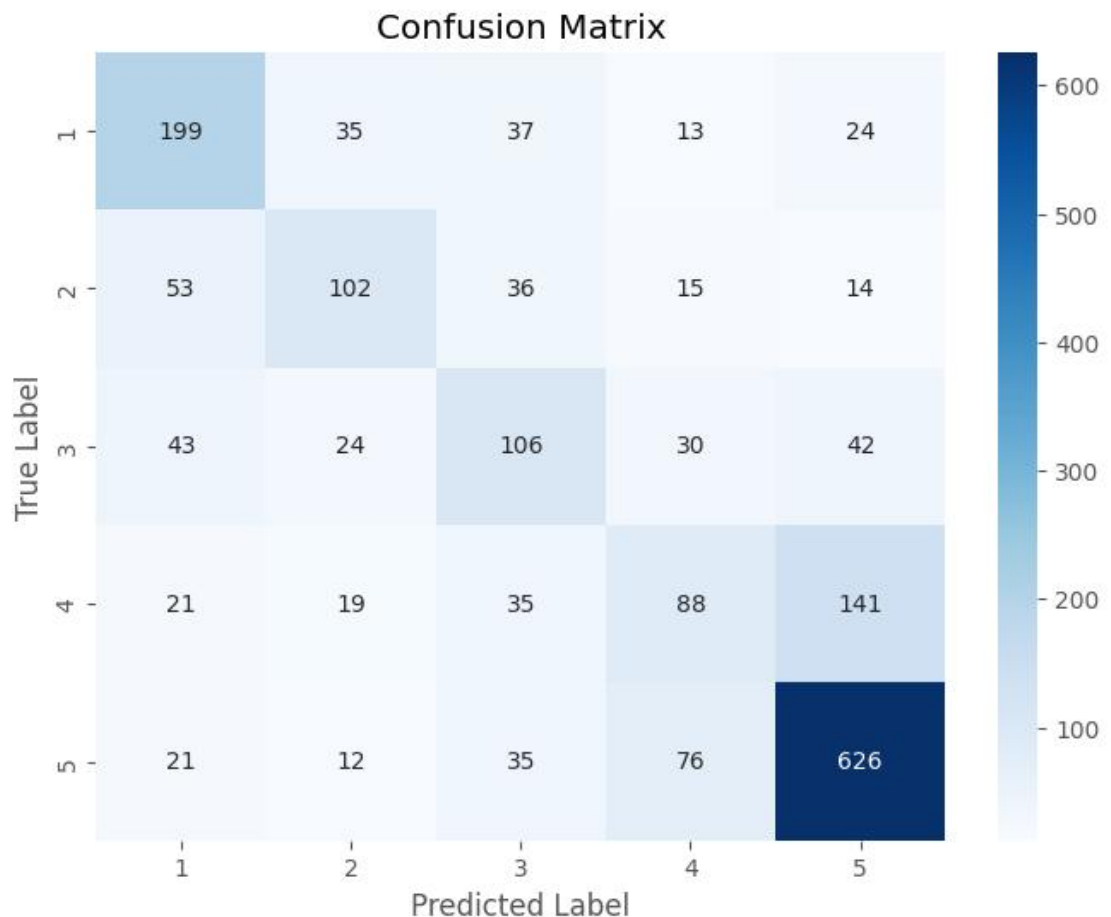
- **max_iter=1000**: số lần lặp tối đa cho thuật toán tối ưu.
- **Logistic Regression** sử dụng thuật toán tối ưu lbfgs đảm bảo đủ số lần lặp để hội tụ
- **multi_class='multinomial'**: sử dụng phân phối xác suất đa thức cho các nhãn.

❖ Huấn luyện mô hình Logistic Regression trên tập huấn luyện

`logistic_model.fit(X_train_phobert, y_train)`

❖ Kết quả dự đoán trên tập kiểm tra

Kết quả dự đoán trên tập kiểm tra được thể hiện bằng ma trận nhầm lẫn (xem hình 3.10)



Hình 3.10 Ma trận nhầm lẫn của mô hình Logistic Regression

2.4. Kết quả huấn luyện

Bảng 3.23 Bảng so sánh độ chính xác của ba mô hình huấn luyện

Mô hình	Accuracy (%)
SVM	58,68
Random Forest	56,68
Logistic Regression	60,69

Nhận xét và đánh giá:

- ❖ Cả 3 mô hình đều có độ chính xác tương đối bằng nhau, tuy nhiên ở mô hình RandomForest, dựa theo ma trận nhầm lẫn có thể thấy được các lớp bị dự đoán nhầm cao hơn 2 mô hình còn lại.
- ❖ So sánh mô hình SVM và Logistic, cả hai mô hình này có tỷ lệ dự đoán đúng và tỷ lệ dự đoán sai tương đối bằng nhau, độ chính xác của Logistic Regression nhỉnh hơn mô hình SVM một chút. Vì vậy, hệ thống phân loại ý kiến sử dụng mô hình Logistic Regression để phân loại ý kiến.

3. Xây dựng chatbot với Rasa NLU

3.1. Xây dựng mô hình chatbot

3.1.1. Xây dựng bộ câu hỏi

Bộ câu hỏi được xây dựng dựa trên một vấn đề và phạm vi cụ thể trong thực tế. Trong đề tài này, bộ câu hỏi được xoay quanh việc hỗ trợ khách hàng là những người mua các mặt hàng nội thất trên website về các vấn đề liên quan đến tìm kiếm, truy vấn sản phẩm, hỏi đáp các vấn đề về bảo hành, chính sách, mua hàng, giao hàng, đơn hàng nhằm hỗ trợ cho khách hàng khi sử dụng website để mua sắm. Với mỗi ý định của khách hàng cần hỗ trợ sẽ xây dựng bộ câu hỏi liên quan đến các ý định đó.

Bảng 3.24 Bảng bộ ý định (intent)

STT	Ý định	Diễn giải	Số lượng câu hỏi
1	greet	Chào hỏi	17
2	goodbye	Tạm biệt, kết thúc trò chuyện	11
3	warranty policy	Chính sách bảo hành	5
4	buying guide	Hướng dẫn mua hàng	17
5	ask_product_type	Hỏi về các loại sản phẩm có trên trang web	61
6	ask_product_made	Nguồn gốc sản phẩm	16
7	ask_product_material	Chất liệu sản phẩm	20
8	ask_product_brand	Thương hiệu sản phẩm	40
9	ask_predict_delivery	Thời gian giao hàng dự kiến	20
10	ask_payment_methods	Các phương thức thanh toán	20
11	ask_payment_guide	Hướng dẫn thanh toán	15
12	ask_return_policy	Quy định đổi trả	15
13	ask_cart_quantity	Số lượng sản phẩm trong giỏ	50
14	ask_wishlist_quantity	Số lượng sản phẩm đã thích	25
15	ask_cart_products	Thông tin chi tiết sản phẩm trong giỏ hàng	39
16	ask_wishlist_product	Thông tin chi tiết sản phẩm đã thích	10
17	ask_order_quantity	Số lượng đơn hàng	29
18	ask_order_by_status_pending_quantity	Số lượng đơn hàng chờ xác nhận	30
19	ask_order_by_status_delivery_quantity	Số lượng đơn hàng chờ giao	30
20	ask_order_by_status_shipped_quantity	Số lượng đơn hàng đã giao thành công	30
21	ask_about_categories	Số lượng danh mục và thông tin các danh	30

		mục	
22	find_products_by_category	Tìm các sản phẩm theo tên danh mục cụ thể	50
23	find_products_by_price_range	Tìm sản phẩm bằng tên sản phẩm với khoảng giá cụ thể	40
24	find_products_by_name	Tìm sản phẩm bằng tên	141
25	suggest_product_large_area	Đề xuất sản phẩm cho không gian lớn	15
26	suggest_product_medium_area	Đề xuất sản phẩm cho không gian vừa	15
27	suggest_product_small_area	Đề xuất sản phẩm cho không gian nhỏ	51
Tổng cộng			806

Ví dụ khi hỏi về “các danh mục của website” có thể hỏi bằng nhiều cách khác nhau.

Ý định (intent)	Ask about categories
Các mẫu câu hỏi (examples)	Cửa hàng có bao nhiêu mục sản phẩm
	Có bao nhiêu danh mục trong cửa hàng hiện tại
	Cửa hàng có bao nhiêu danh mục
	Số lượng danh mục sản phẩm trong cửa hàng là bao nhiêu
	Cửa hàng có tất cả bao nhiêu danh mục

Hoặc các ý định có câu hỏi chứa thực thể (entity) như:

Ý định (intent)	Find_products_by_category
Các mẫu câu hỏi (examples)	Tôi muốn xem các sản phẩm trong danh mục [sofas](category_name)
	Bạn có sản phẩm trong danh mục [ghế sofa dài](category_name) hay không
	Có sản phẩm trong danh mục [bàn nhỏ](category_name) không ạ
	Tôi cần tìm sản phẩm trong danh mục [bàn cà phê](category_name)
	Cho tôi xem các ghế trong danh mục [ghế](category_name)

3.1.2. Xác định các thông tin cần trích xuất

Trích xuất thông tin từ tin nhắn người dùng còn gọi là trích xuất thực thể (entity). Các thực thể này được định nghĩa trong các câu hỏi dựng sẵn, qua đó giúp chatbot có thể sử dụng các thực thể này để đưa ra các hành động với bộ câu trả lời được xây dựng.

Thực thể (entities)	Diễn giải
Product name	Tên sản phẩm

Category name	Tên danh mục
From_price	Khoảng giá tìm kiếm nhỏ nhất
To_price	Khoảng giá tìm kiếm lớn nhất

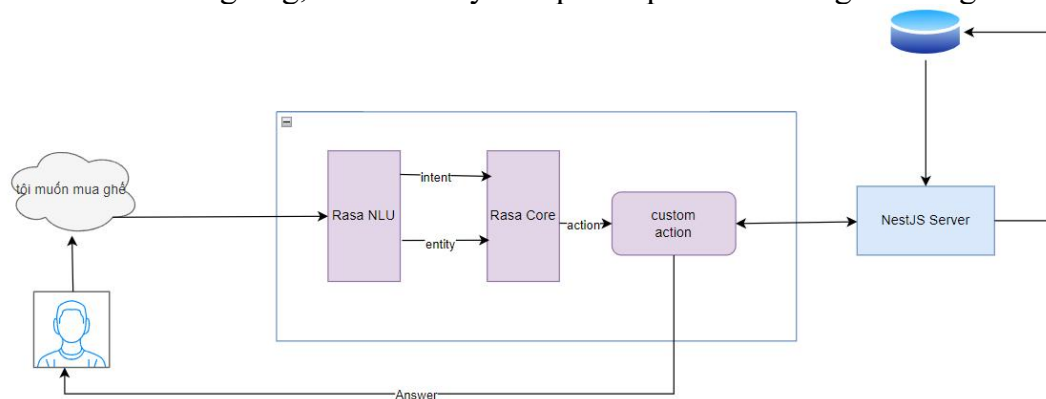
3.1.3. Xây dựng bộ câu trả lời

Sau khi xác định được ý định của người dùng và trích xuất thông tin từ các thực thể trong câu hỏi (nếu có), công việc tiếp theo chính là đưa ra các câu trả lời tương ứng (xem Hình 3.11). Trong Rasa có các dạng câu trả lời là : Default actions, utter actions và custom actions.

- **Default actions:** Là các câu trả lời mặc định được tích hợp sẵn vào trình quản lý hội thoại khi người dùng khởi động phiên hội thoại.
- **Utter actions:** Là các câu trả lời cố định được định nghĩa cho chatbot. Thông thường, các utter actions sẽ không mất thời gian xử lý các truy vấn và tốc độ phản hồi nhanh. Ví dụ khi khách hàng chào hỏi, chatbot sẽ phản hồi thông điệp đó với utter actions đã được định nghĩa.

```
utter_greet:
- text: "Dạ! Louis xin chào ạ!"
```

- **Custom actions:** Là các câu trả lời cần tính linh hoạt, cập nhật liên tục. Các hành động này sẽ thực hiện các thao tác xử lý, truy vấn dữ liệu trước khi đưa ra phản hồi về người dùng. Ví dụ như tìm sản phẩm theo khoảng giá bằng tên sản phẩm "tôi muốn xem các ghế ăn giá từ 2000 đến 6000" thì hệ thống cần phải truy vấn thông tin từ cơ sở dữ liệu với điều kiện truy vấn tương ứng, sau đó xử lý kết quả và phản hồi về người dùng.



Hình 3.11 Hình mô tả cách đưa ra câu trả lời

3.1.4. Xây dựng các kịch bản hội thoại

Sau khi xây dựng được bộ câu hỏi phân loại các ý định của người dùng. Bước tiếp theo là tiến hành xây dựng các kịch bản hội thoại bằng các quy định và trình tự thực hiện các hành động. Trong Rasa, các định nghĩa hành vi được thực hiện bằng hai cách:

- **Rule:** Là các quy tắc xác định các tình huống phản hồi nhất định của bot. Mỗi rule là một chuỗi các hành động mà bot cần phải thực hiện trong một

tình huống cụ thể, thường sử dụng cho các hội thoại đơn giản, xác định rõ ràng có tính dự đoán cao và dễ kiểm soát luồng hội thoại.

- rule: chính sách mua hàng
steps:
- intent: warranty_policy
- action: utter_warranty_policy

Với quy tắc “chính sách mua hàng” trên, khi người dùng hỏi các câu hỏi liên quan đến ý định về chính sách mua hàng thì bot sẽ thực thi utter action để trả lời tương ứng.

- **Stories:** Là các kịch bản mô phỏng cuộc hội thoại đầy đủ mà bot sẽ theo dõi để hiểu các người dùng có thể tương tác với bot qua nhiều bước, tạo thành một tình huống hội thoại hoàn chỉnh và phức tạp.

- story: Tìm sản phẩm theo loại
steps:
- intent: ask_product_type
- action: action_ask_product_type
- intent: find_products_by_category
- action: action_find_products_by_category

Ví dụ với câu chuyện “tìm sản phẩm theo loại”, khi ý định ban đầu của người dùng là “hỏi loại sản phẩm”, lúc này bot sẽ phản hồi lại bằng cách hỏi lại người dùng đang hỏi về “loại sản phẩm nào”, khi người dùng trả lời câu tiếp theo, bot sẽ dựa vào câu hỏi của người dùng để xác định các “sản phẩm” thuộc “loại” mà người dùng cung cấp.

3.2. Huấn luyện chatbot

Quá trình huấn luyện được thực hiện sau khi hoàn tất các bước xây dựng các bộ câu hỏi, phân tích ý định, bộ câu trả lời và kịch bản hội thoại.

3.2.1. Phân chia dữ liệu huấn luyện

Phân chia dữ liệu huấn luyện được thực hiện trên dữ liệu NLU, với tỷ lệ phân chia mặc định của Rasa là **8:2**, với 80% dữ liệu cho việc huấn luyện và 20% cho việc kiểm thử. Chi tiết phân chia dữ liệu như sau:

- Tổng số câu của tập dữ liệu: 806.
- Tập huấn luyện: 645.
- Tập kiểm thử: 161.

3.2.2. Cấu hình pipeline

Cấu hình pipeline với Rasa mặc định được xây dựng chi tiết theo Bảng 3.25 bên dưới.

Bảng 3.25 Bảng cấu hình pipeline

language	vi	
Pipeline	name	WhiteSpaceTokenizer
	name	RegexFeaturizer
	name	LexicalSyntacticFeaturizer
	name	CounVectorsFeaturizer
	name	CounVectorsFeaturizer
	name	DIETClassifier
	epochs	100
	constrain_similarities	epoch
	name	EntitySynonymMapper
	name	FallbackClassifier
	threshold	0.4
	ambiguity_threshold	0.1
Policies	name	RulePolicy
	core fallback threshold	0.4
	core fallback action_name	action_fallback
	enable fallback prediction	true

Chi tiết các cấu hình trong pipeline như sau:

- **WhiteSpaceTokenizer**: Bộ phân tách từ sử dụng khoảng trắng để tách các từ trong văn bản, lý do sử dụng bộ tách từ này vì đang sử dụng ngôn ngữ tiếng Việt, đa số các từ trong tiếng Việt được cách nhau bởi khoảng trắng và thói quen gõ của người Việt.
- **RegexFeaturizer**: Tạo các đặc trưng dựa trên biểu thức chính quy để nhận diện các mẫu trong văn bản.
- **LexicalSyntacticFeaturizer**: Tạo các đặc trưng từ các tính chất hình thức và cú pháp của từ vựng trong văn bản.
- **CounterVectorsFeaturizer**: Tạo các đặc trưng số học từ văn bản bằng cách đếm số lần xuất hiện của các từ.
- **DIETClassifier**: Đây là classifier chính trong mô hình Rasa, giúp phân loại các intent và trích xuất các thực thể từ văn bản.
- **epochs**: Được đặt thành 100, có nghĩa là mô hình sẽ được huấn luyện 100 lần.

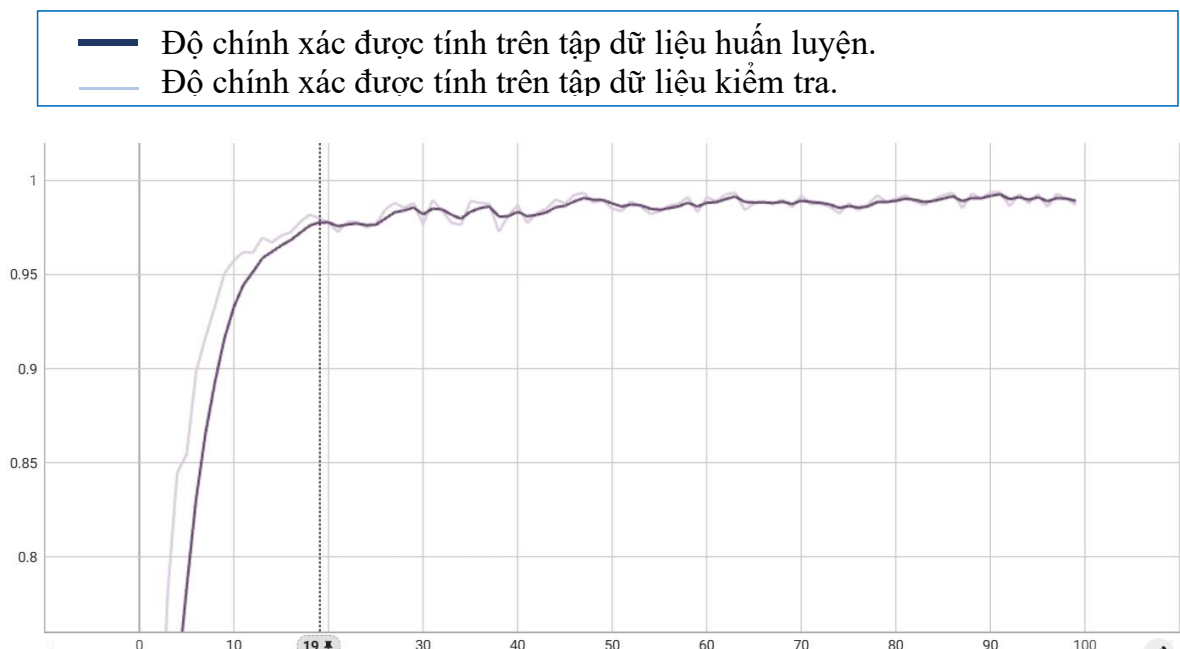
- **constrain_similarities**: Giúp mô hình tránh sự tương đồng quá mức giữa các câu có ý nghĩa khác nhau.
- **EntitySynonymMapper**: Dùng để ánh xạ các từ đồng nghĩa cho các thực thể, giúp mô hình nhận diện các biến thể khác nhau của thực thể.
- **threshold**: Đặt ngưỡng cho việc quyết định khi nào nên gọi action fallback nếu không có intent phù hợp.
- **ambiguity_threshold**: Ngưỡng cho việc xác định khi có sự mơ hồ cao trong lựa chọn intent.
- **RulePolicy**: Là chính sách quy tắc, giúp mô hình quyết định hành động dựa trên các quy tắc đã định sẵn.

3.3. Kết quả huấn luyện và nhận xét

3.3.1. Kết quả

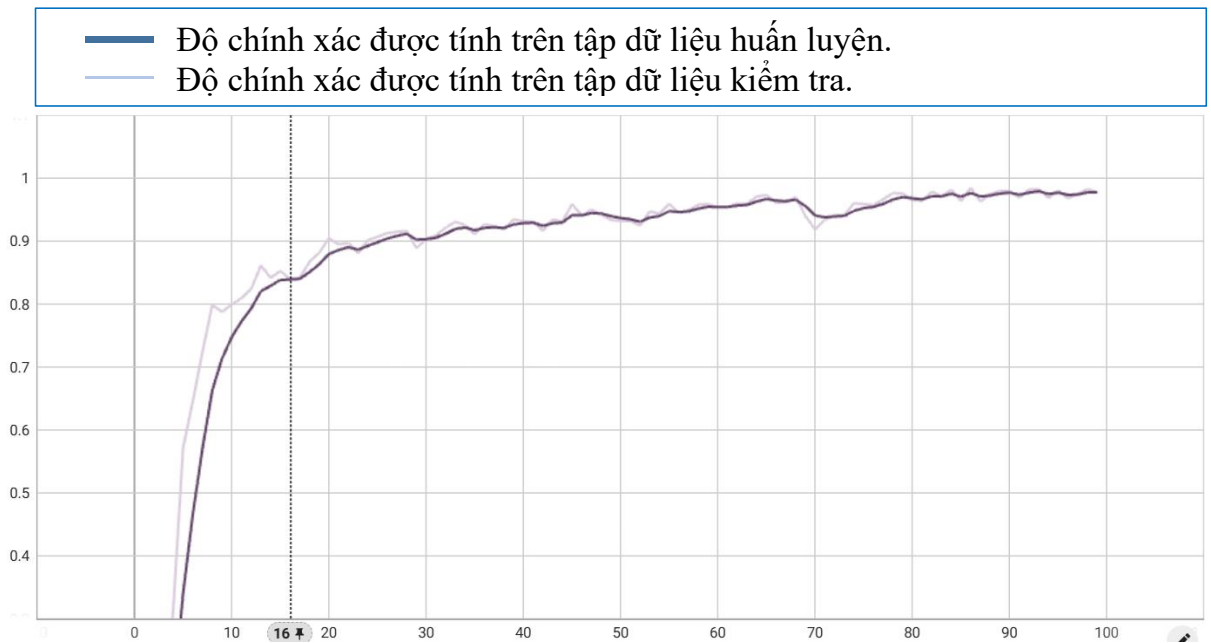
Quá trình huấn luyện được thực hiện qua 100 epoch. Rasa sử dụng chỉ số đánh giá Accuracy và F1-Score làm hai chỉ số mặc định để đánh giá độ chính xác và chỉ số Cross Entropy Loss để tính toán độ chính xác.

- ♦ Độ chính xác được tính theo chỉ số Accuracy (xem Hình 3.12): Giá trị độ chính xác được khởi đầu khoảng 0.8 (80%), sau đó nhanh chóng tăng lên trong 15 epoch đầu và hội tụ sau khoảng 20 epoch, nhưng nhanh chóng tách ra sau đó. Lúc này độ chính xác xấp xỉ 0.97 - 0.98 (97-98%).



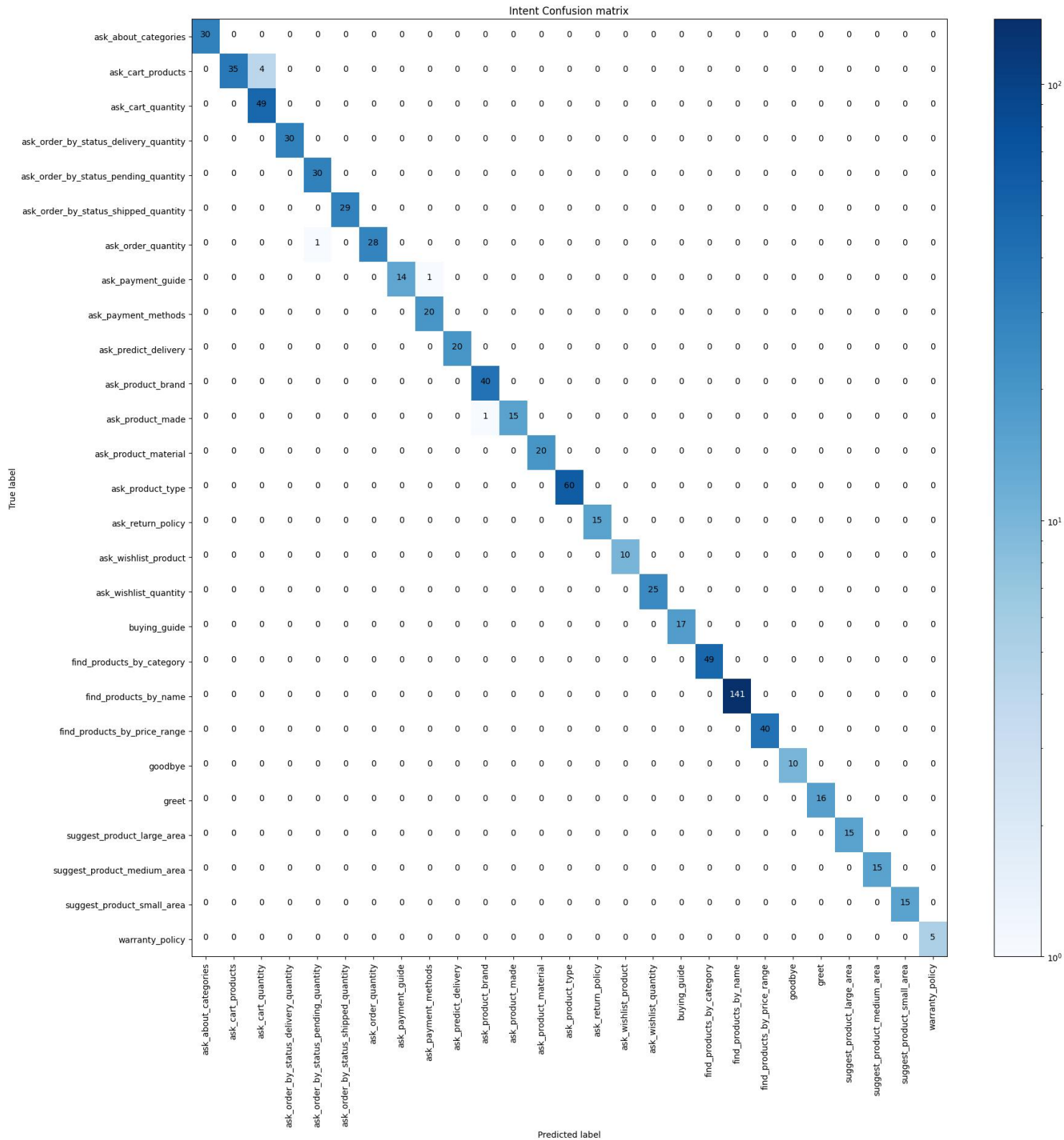
Hình 3.12 Độ chính xác của chatbot dựa trên chỉ số Accuracy

- ♦ Với độ chính xác được tính toán theo chỉ số F1-Score (xem Hình 3.13) cho thấy xu hướng hội tụ nhanh hơn so với Accuracy.

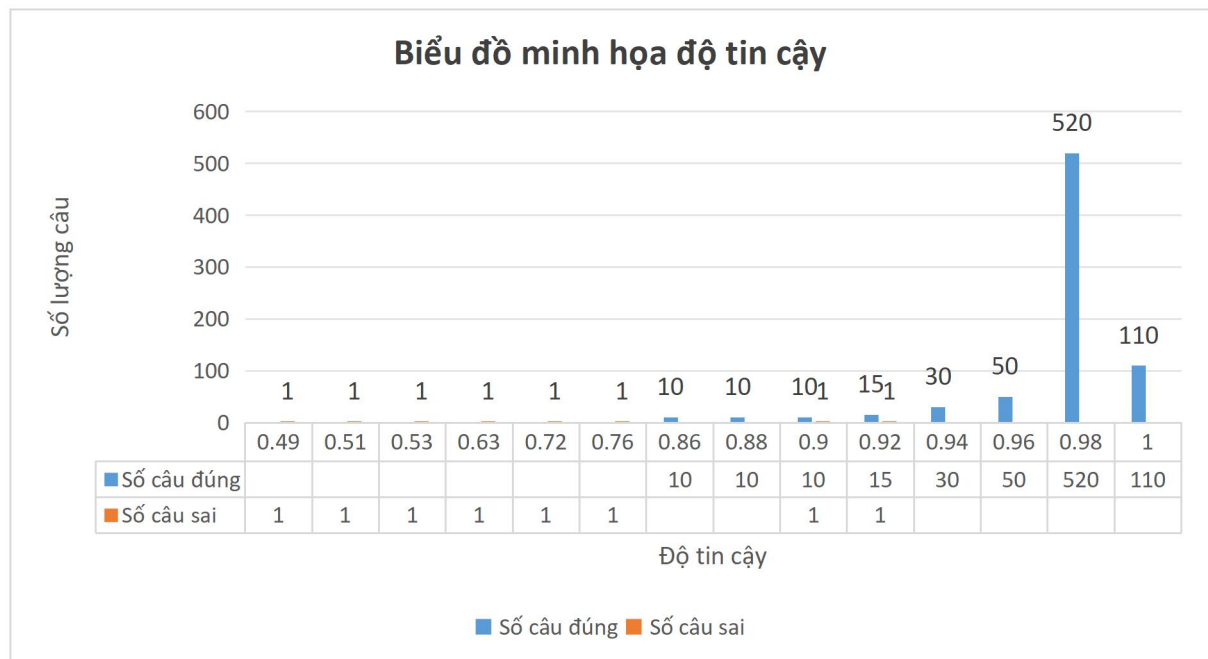


Hình 3.13 Độ chính xác tính theo chỉ số F1-Score

- Ma trận nhầm lẫn: Kết quả ma trận nhầm lẫn cho thấy hầu hết các ý định đều được xác định chính xác, mô hình có sự phân biệt tốt, chỉ có ý định “ask_product_cart” là còn nhầm lẫn với ý định khác.



Hình 3.14 Ma trận nhầm lẫn của chatbot sau huấn luyện



Hình 3.15 Biểu đồ minh họa độ tin cậy của chatbot trong phân loại ý định

Trong tổng 161 mẫu dữ liệu kiểm tra, khi đi qua mô hình cho kết quả sai là 8 câu. Hầu hết các mẫu đều cho dự đoán chính xác với độ tự tin từ 83% - 95%.

3.3.2. Nhận xét

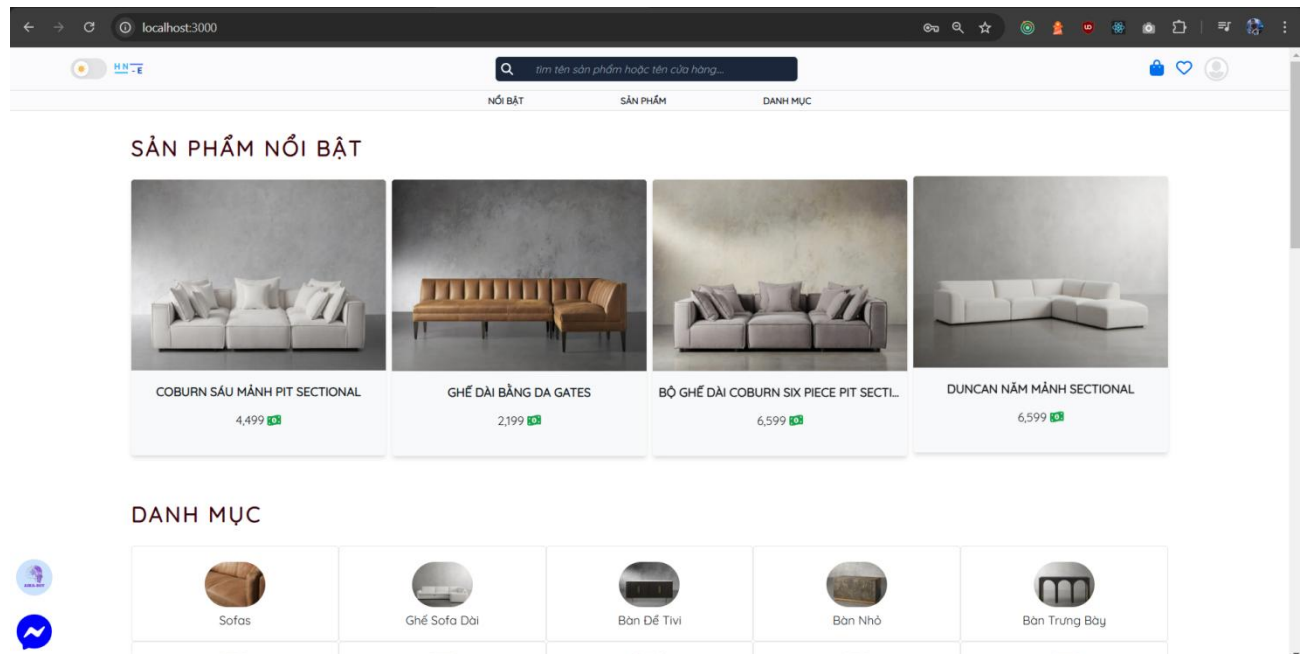
Qua kết quả tổng quát sau khi huấn luyện và kết quả dự đoán cho thấy Rasa đã phân loại ý định với DIETClassifier cho kết quả rất tốt. Kết thúc quá trình huấn luyện chatbot với pipeline mặc định của Rasa cho thấy mô hình rất phù hợp để tích hợp vào website “sàn thương mại điện tử kinh doanh các mặt hàng nội thất tích hợp chatbot hỗ trợ khách hàng”.

CHƯƠNG IV: KIỂM THỬ ĐÁNH GIÁ

1. Giao diện website

1.1. Giao diện trang chủ

Người dùng có thể xem các sản phẩm nổi bật, danh mục sản phẩm và một số sản phẩm được hiển thị, có thể tìm kiếm sản phẩm trên thanh tìm kiếm.

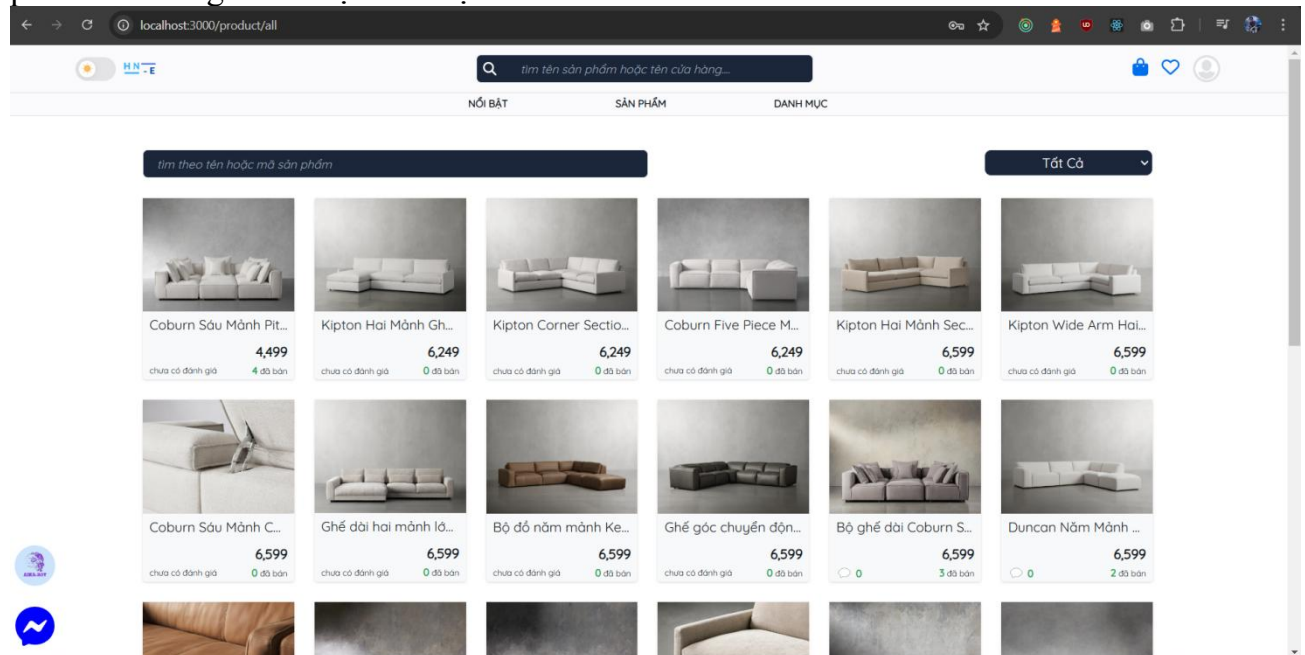


Hình 4.1. Giao diện trang chủ

Ở trang chủ, xem Hình 4.1, người dùng có thể sử dụng thanh tìm kiếm để tìm kiếm sản phẩm theo tên, truy cập nhanh vào giỏ hàng và danh sách sản phẩm yêu thích bằng cách nhấn vào biểu tượng “giỏ hàng” và “yêu thích”. Bên cạnh đó, người dùng có thể đi đến trang khi nhấn vào biểu tượng “người dùng” bên góc phải trên cùng.

1.2. Giao diện tất cả sản phẩm

Hiện thị tất cả sản phẩm của trang web và được phân trang. Người dùng có thể tìm kiếm sản phẩm và bấm vào sản phẩm để xem chi tiết, giao diện trang tất cả sản phẩm của trang web được thể hiện như Hình 4.2.



Hình 4.2. Giao diện trang hiển thị tất cả sản phẩm

Tại trang hiển thị tất cả sản phẩm, người dùng có thể xem danh sách sản phẩm, tìm kiếm sản phẩm bằng cách điền tên sản phẩm cần tìm trên thanh tìm kiếm, sắp xếp thứ tự xem của sản phẩm theo thứ tự bản chữ cái từ “a-z” hoặc từ “z-a” bằng bộ sắp xếp phía bên phải màn hình.

1.3. Giao diện thông tin người dùng

Hiển thị thông tin của người dùng. Người dùng có thể thay đổi thông tin cá nhân bằng cách nhấn vào nút “chỉnh sửa”, điền các mục cần thay đổi và nhấn nút “lưu” để lưu thông tin, giao diện thông tin người dùng được thể hiện như Hình 4.3.

Hình 4.3. Giao diện trang thông tin người dùng

Khi người dùng muốn thay đổi thông tin, đầu tiên cần nhấn nút “chỉnh sửa” để mở khóa các ô thông tin, sau đó người dùng có thể thay đổi các thông tin như tên, số điện thoại, nếu thay đổi địa chỉ khác, người dùng phải điền theo thứ tự Tỉnh/Thành phố-Quận/Huyện-Xã. Sau đó nhấn nút “Lưu” để cập nhật thông tin mới.

1.4. Giao diện giỏ hàng

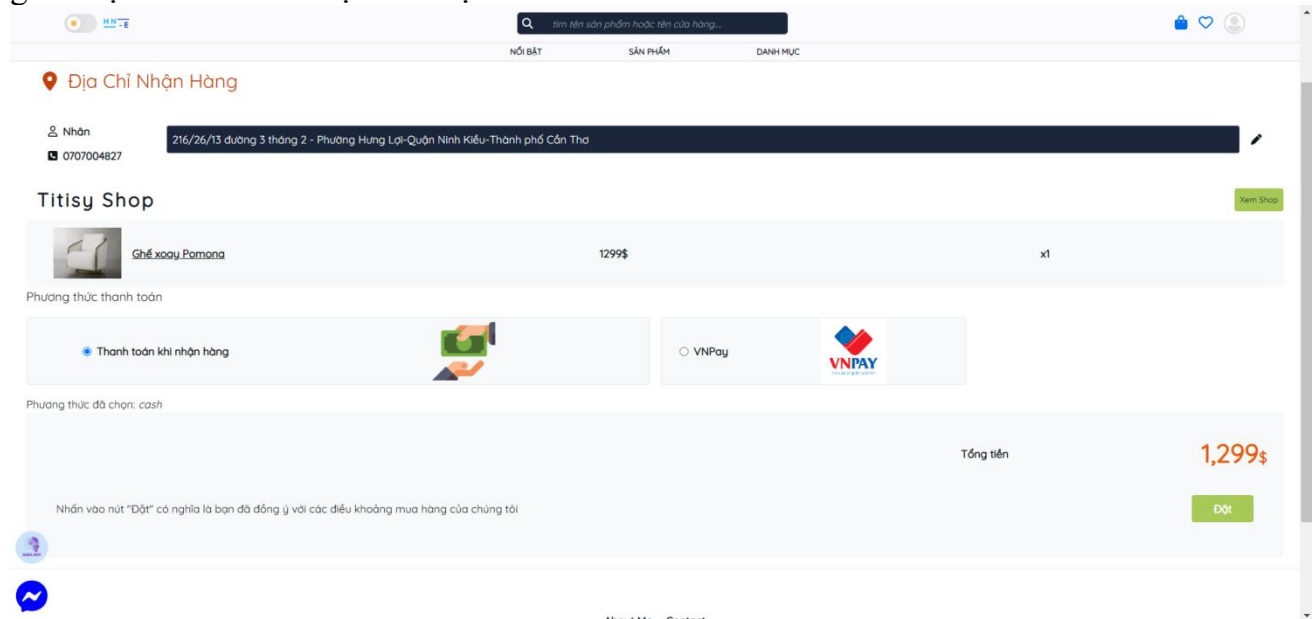
Hiển thị các sản phẩm mà người dùng đã thêm vào giỏ hàng, người dùng có thể chọn một hoặc nhiều sản phẩm và số lượng sản phẩm để thanh toán, giao diện giỏ hàng được thể hiện như Hình 4.4.

Hình 4.4. Giao diện trang giỏ hàng

Tại trang giao diện giỏ hàng, người dùng có thể chọn một hoặc nhiều sản phẩm để mua bằng cách tích vào các ô bên cạnh sản phẩm, sau đó nhấn nút “Mua” để đến trang thanh toán, có thể thay đổi số lượng sản phẩm bằng cách nhấn vào biểu tượng “+” hoặc “-” ở ô số lượng sản phẩm. Nếu người dùng muốn xóa sản phẩm ra khỏi giỏ hàng, có thể nhấn nút “X” ở sản phẩm cần xóa.

1.5. Giao diện thanh toán

Hiện thị các sản phẩm mà người dùng đã chọn để thanh toán, người dùng có thể thay đổi địa chỉ nhận hàng, chọn phương thức thanh toán và tiến hành đặt hàng, giao diện thanh toán được thể hiện như Hình 4.5.



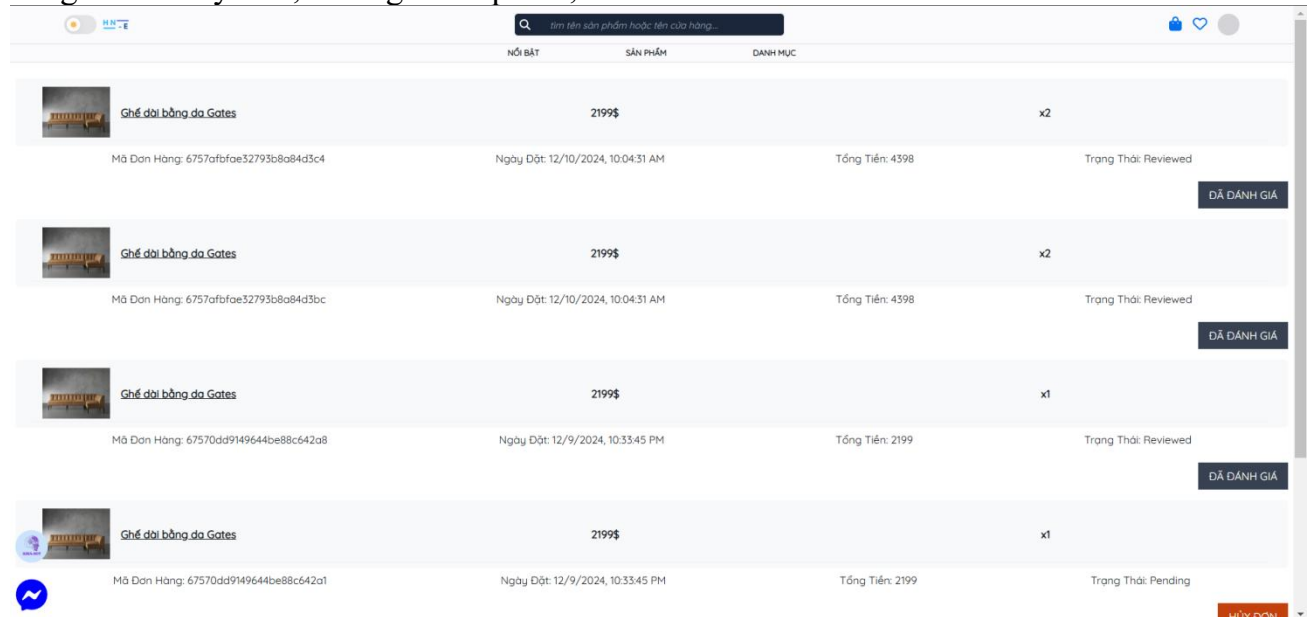
The screenshot displays the checkout interface of the Titisy Shop. At the top, there's a navigation bar with a search bar and icons for cart, heart, and user profile. Below this, the user's address is shown: "Địa Chỉ Nhận Hàng" (Delivery Address) as "Nhân" (Name), "216/26/13 đường 3 tháng 2 - Phường Hưng Lợi-Quận Ninh Kiều-Thành phố Cần Thơ" (Address), and "0707004827" (Phone). The selected item is "Ghế xoay Pomona" priced at 1299\$ with a quantity of x1. Under "Phương thức thanh toán" (Payment Method), "Thanh toán khi nhận hàng" (Cash on delivery) is selected. The "Phương thức đã chọn: cash" (Selected method: cash) is confirmed. The total amount "Tổng tiền" is 1,299\$. A green "Đặt" (Place order) button is at the bottom right. A note at the bottom states: "Nhấn vào nút 'Đặt' có nghĩa là bạn đã đồng ý với các điều khoản mua hàng của chúng tôi" (Pressing the 'Place order' button means you agree with our purchase terms).

Hình 4.5. Giao diện trang thanh toán

Địa chỉ nhận hàng mặc định là địa chỉ hiện tại của người dùng ở phần thông tin, nếu người dùng muốn thay đổi địa chỉ, người dùng có thể điền thông tin địa chỉ mới ở ô địa chỉ, có thể chọn phương thức thanh toán bằng cách tích vào phương thức muốn chọn, sau đó tiến hành nhấn nút “Đặt” để đặt hàng.

1.6. Giao diện lịch sử đặt hàng

Hiển thị các sản phẩm mà người dùng đã đặt và trạng thái đơn hàng, khách hàng có thể hủy đơn, đánh giá sản phẩm, xem Hình 4.6.

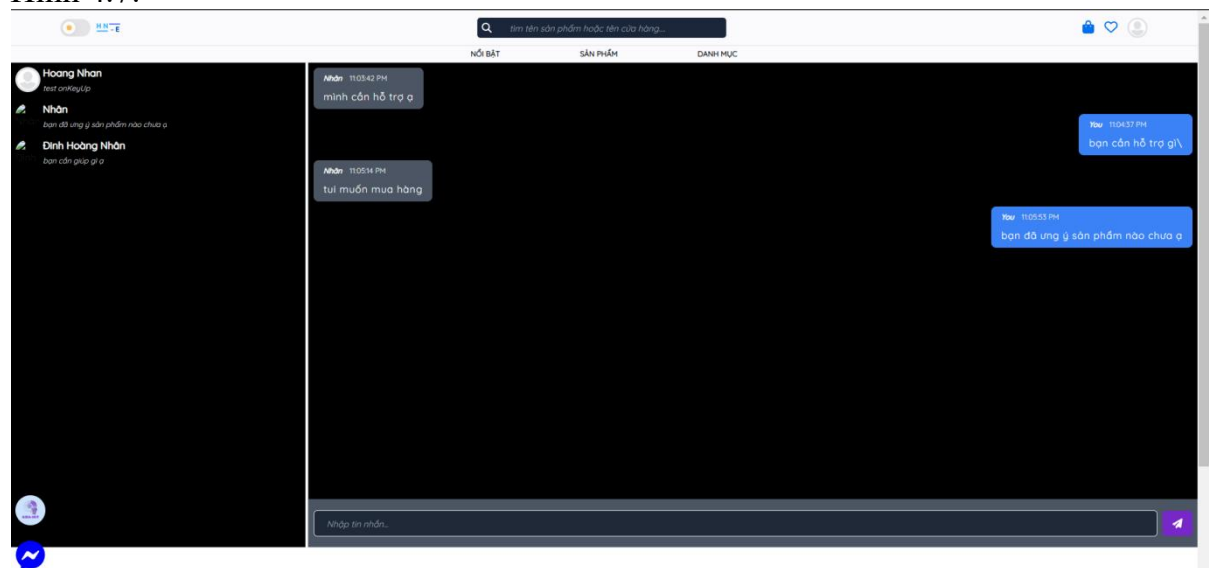


Hình 4.6. Giao diện trang lịch sử đặt hàng

Để hủy hoặc đánh giá sản phẩm, người dùng có thể nhấn nút “Hủy” hoặc “Đánh giá” ở mỗi sản phẩm

1.7. Giao diện hội thoại

Hiển thị giao diện hội thoại của người dùng với các người dùng khác, người dùng có thể chọn cuộc hội thoại với ai đó và bắt đầu trò chuyện. Tính năng trò chuyện của website là trò chuyện thời gian thực (chat realtime), giao diện trò chuyện như Hình 4.7.

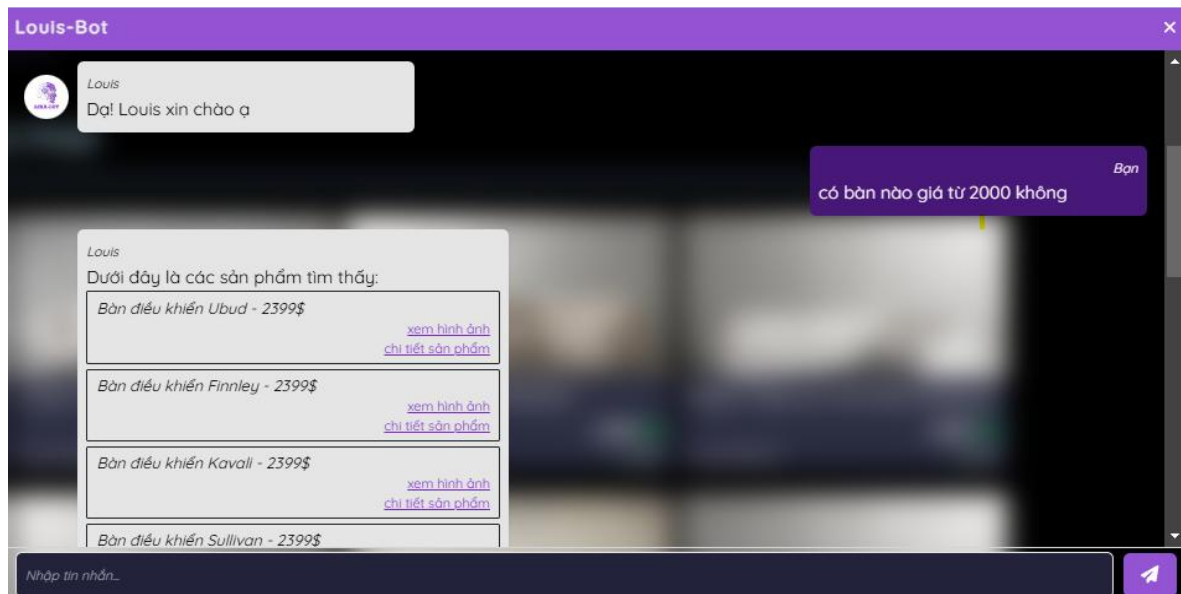


Hình 4.7. Giao diện trang hội thoại

Để bắt đầu cuộc trò chuyện, người dùng chọn đối tượng để trò chuyện bên trái, sau đó nhập tin nhắn ở ô “nhập tin nhắn” và nhấn “Enter” để gửi.

1.8. Giao diện cửa sổ chatbot

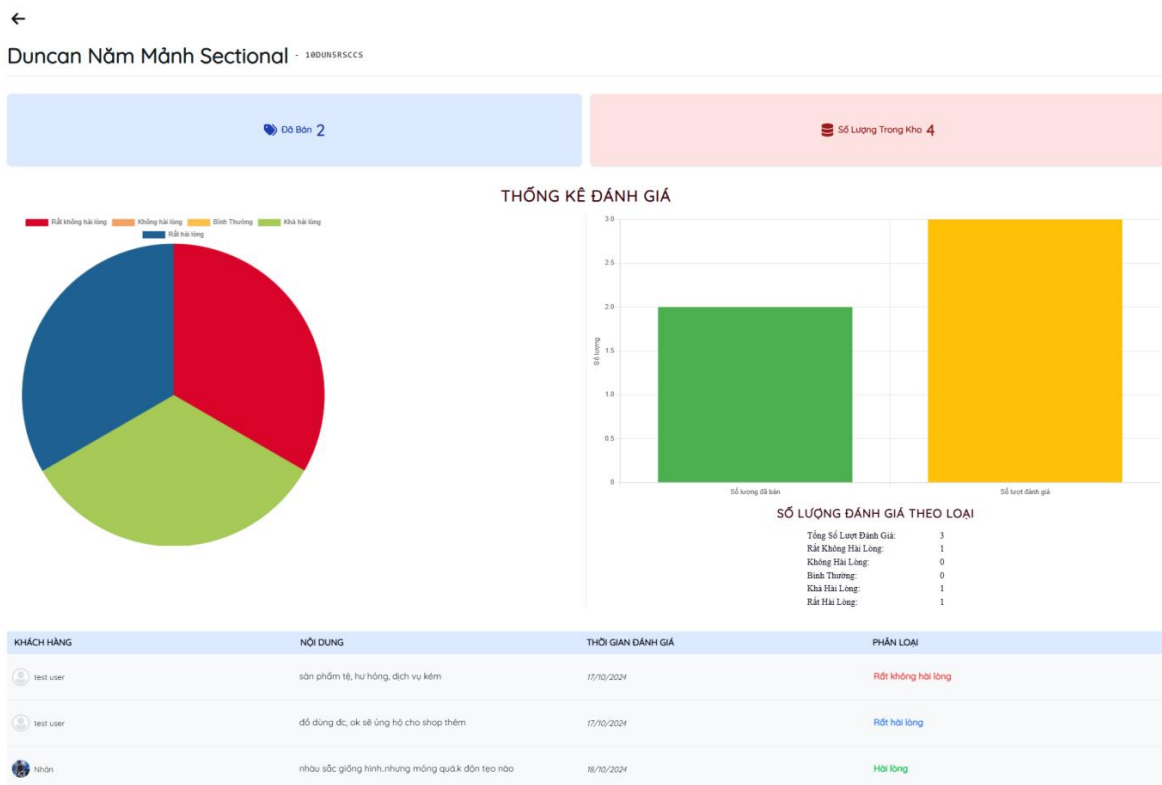
Hiện thị cửa sổ chatbot, người dùng có thể tương tác với chatbot bằng văn bản, xem Hình 4.8.



Hình 4.8. Giao diện khung trò chuyện với chatbot

1.9. Giao diện thống kê sản phẩm

Hiện thị các thống kê của sản phẩm cụ thể về các thông tin như số lượng đã bán, số lượng tồn kho, lượt đánh giá, mức độ hài lòng của các đánh giá đã được hệ thống tự động phân loại, xem Hình 4.9.



Hình 4.9. Giao diện thống kê sản phẩm

1. Kết quả kiểm thử

1.1. Mục đích kiểm thử

- Tìm các lỗi, sai sót trong ứng dụng, giúp nhận diện và ghi nhận các lỗi này để có thể sửa chữa trước khi triển khai.
- Giúp đánh giá mức độ chất lượng ứng dụng và ngăn ngừa các lỗi có thể xảy ra trong tương lai.
- Kiểm tra ứng dụng hoạt động đúng với thiết kế.
- Làm tài liệu cho giai đoạn bảo trì.

1.2. Phạm vi kiểm thử

- Kiểm thử một vài chức năng chính của trang web nhằm đảm bảo ứng dụng đáp ứng được các yêu cầu tối thiểu đặt ra.

1.3. Cách tiếp cận

- Kiểm thử thủ công.

1.4. Tiêu chí đánh giá

- Kiểm thử thành công khi kết quả kiểm thử giống hoặc tương tự với kết quả mong muốn.
- Kiểm thử thất bại khi kết quả kiểm thử khác hoàn toàn với kết quả mong muốn.

1.5. Môi trường kiểm thử

- Trình duyệt sử dụng: Trình duyệt Chrome phiên bản 130.0.6723.117 (Official Build) (64-bit)
- Phần cứng (máy tính): Laptop Acer, Hệ điều hành Window 11, 512GB Rom, 16GB Ram, Card đồ họa GTX 1650.
- Phần mềm sử dụng: Visual Studio Code 2022, Window Terminal.
- Môi trường cài đặt: môi trường ảo sử dụng Python 3.10, CUDA 12.4

1.6. Các chức năng kiểm thử

- Chức năng đặt hàng.
- Chức năng tạo sản phẩm mới.
- Chức năng tạo danh mục mới.

1.6.1. Chức năng đặt hàng

Bảng 4.1 Bảng kiểm thử chức năng đặt hàng

STT	Tên trường hợp thử nghiệm	Dữ liệu đầu vào	Kết quả mong muốn
1	Đặt hàng thành công	Id người dùng (mặc định được thêm vào), sản phẩm, số lượng, đơn giá, phương thức thanh toán, địa chỉ nhận hàng	Thông báo đặt hàng thành công cho người dùng
2	Đặt hàng thất bại khi không chọn phương thức thanh toán	Đầy đủ các trường nhưng không chọn phương thức thanh toán	Hiện thị thông báo “vui lòng chọn phương thức thanh toán”
3	Đặt hàng thất bại khi địa chỉ nhận hàng để trống	Đầy đủ các trường nhưng để trống địa chỉ nhận hàng	Thông báo yêu cầu người dùng nhập địa chỉ nhận hàng

1.6.2. Chức năng tạo sản phẩm mới

Bảng 4.2 Bảng kiểm thử chức năng tạo sản phẩm mới

STT	Tên trường hợp thử nghiệm	Dữ liệu đầu vào	Kết quả mong muốn
1	Tạo sản phẩm mới thành công	Tên sản phẩm, hình ảnh của sản phẩm, mô tả, số lượng (mặc định 1), đơn giá, id danh mục.	Thông báo tạo sản phẩm thành công, hiển thị ra danh sách sản phẩm mới với sản phẩm vừa được tạo.
2	Tạo sản phẩm thất bại với id danh mục để trống	Đầy đủ các trường nhưng không chọn danh mục	Hiện thị cảnh báo yêu cầu người dùng chọn danh mục
3	Tạo sản phẩm thất bại với tên sản phẩm để trống	Đầy đủ các trường nhưng không có tên sản phẩm	Hiện thị cảnh báo yêu cầu người dùng nhập tên sản phẩm
4	Tạo sản phẩm thất bại khi không chọn hình ảnh sản phẩm	Đầy đủ các trường nhưng không chọn ảnh của sản phẩm	Hiện thị cảnh báo yêu cầu người dùng chọn ít nhất 1 ảnh
5	Tạo sản phẩm thất bại khi để trống giá sản phẩm	Đầy đủ các trường nhưng không điền giá sản phẩm	Hiện thị yêu cầu người dùng nhập giá sản phẩm

1.6.3. Chức năng tạo danh mục mới

Bảng 4.3 Bảng kiểm thử chức năng tạo danh mục mới

STT	Tên trường hợp thử nghiệm	Kết quả đầu vào	Kết quả mong muốn
1	Tạo danh mục mới thành công	Tên danh mục, hình ảnh danh mục, mô tả	Thông báo tạo danh mục thành công, tải lại danh sách danh mục mới với danh mục đã tạo
2	Tạo danh mục thất bại khi để trống tên danh mục	Đầy đủ các trường nhưng không điền tên danh mục	Hiện thị thông báo yêu cầu người dùng nhập tên danh mục
3	Tạo danh mục thành công với mô tả danh mục để trống	Đầy đủ các trường nhưng không cần điền mô tả	Thông báo tạo danh mục thành công, danh mục mới sẽ không có mô tả
4	Tạo danh mục thành công khi không chọn ảnh của danh mục	Đầy đủ các trường nhưng không chọn ảnh danh mục	Thông báo tạo danh mục thành công, hiển thị danh mục mới.

2. Đánh giá.

2.1. Website sàn thương mại điện tử kinh doanh nội thất

- **Ưu điểm:** Giao diện đơn giản, dễ sử dụng, có hỗ trợ chế độ nền tối, bố cục các trang rõ ràng, tường minh.
- **Nhược điểm:** Bảo mật chưa cao, màu sắc giao diện chưa được hài hòa.

2.2. Hệ thống tự động phân loại đánh giá

- **Ưu điểm:** Tốc độ phân loại đánh giá nhanh, tốn ít tài nguyên.
- **Nhược điểm:** Độ chính xác phân lớp còn thấp, chưa thể đạt 100%. Dữ liệu cho mô hình học còn ít.

2.3. Chatbot

- **Ưu điểm:** Phản hồi nhanh, chính xác. Giao diện dễ sử dụng. Lời thoại chatbot thân thiện, gần gũi với giao tiếp hằng ngày.
- **Nhược điểm:** Còn phân định sai ý định, độ chính xác chưa thể đạt 100%. Kịch bản hội thoại còn đơn giản, chưa đáp ứng đủ tất cả các trường hợp giao tiếp.

PHẦN KẾT LUẬN

1. Kết quả đạt được

Sau nghiên cứu “Xây dựng sàn thương mại điện tử kinh doanh các mặt hàng nội thất tích hợp chatbot hỗ trợ khách hàng”. Đáp ứng được yêu cầu cơ bản .

Xây dựng trang web kinh doanh nội thất với các tính năng như người dùng có thể đăng ký, quản lý thông tin cá nhân, mua sắm sản phẩm, tra cứu lịch sử đơn hàng, trò chuyện với người bán, tương tác với chatbot hỗ trợ. Người kinh doanh có thể quản lý cửa hàng của mình, quản lý sản phẩm, quản lý đơn đặt hàng, quản lý danh mục, quản lý quảng cáo, trò chuyện với khách hàng theo thời gian thực, và thực hiện các chức năng như một người dùng bình thường. Giao diện đơn giản, dễ sử dụng.

Xây dựng được chatbot hỗ trợ khách hàng với các đặc điểm như trả lời nhanh, độ chính xác cao, ngôn ngữ thân thiện, giao diện dễ sử dụng. Xây dựng được hệ thống tự động phân loại ý kiến khách hàng với các đặc điểm như dự đoán nhanh, khá chính xác, ít tốn tài nguyên máy chủ.

2. Hạn chế

Trong quá trình thực hiện công đoạn thu thập dữ liệu để huấn luyện mô hình phân loại ý kiến, dữ liệu thu thập bị rời rạc, số lượng còn ít, có sự mất cân bằng của lớp có nhãn là 5 so với các lớp còn lại. Kịch bản chatbot đơn giản, chưa linh hoạt. Bộ câu hỏi xác định ý định còn ít dẫn đến việc chatbot có thể nhận định sai các ý định mong muốn.

3. Hướng phát triển

Xây dựng thêm mô hình phân loại ý kiến với mô hình ngôn ngữ lớn. Thu thập bổ sung thêm các bình luận trên các sàn thương mại điện tử lớn và cân bằng lại tập dữ liệu. Xây dựng thêm các kịch bản chi tiết cho chatbot. Bổ sung các câu hỏi cho việc phân loại ý định. Xây dựng thêm ứng dụng cho di động, tablet sử dụng ReactNative. Xây dựng thêm tính năng voice chat và gọi thoại bằng video. Xây dựng thêm chức năng nhận diện giọng nói và chuyển văn bản thành giọng nói để hỗ trợ người dùng trong việc trò chuyện, trao đổi.

TÀI LIỆU THAM KHẢO

- [1] Huỳnh, Thị Kim Hà. "Nghiên cứu tình hình phát triển thương mại điện tử tại Việt Nam." (2013).
- [2] Cù, Lê Xuân. "Tạp chí khoa học Đại học mở Thành Phố Hồ Chí Minh-Kinh tế và quản trị kinh doanh", no. 5 (2023): 32-46.
- [3] Nick, Todd G., and Kathleen M. Campbell. "Logistic regression." *Topics in biostatistics* (2007): 273-301.
- [4] Pisner, Derek A., and David M. Schnyer. "Support vector machine." In *Machine learning*, pp. 101-121. Academic Press, 2020.
- [5] Breiman, Leo. "Random forests." *Machine learning* 45 (2001): 5-32.
- [6] Pham, Bao. "Rasa NLU Chatbot." PhD diss., Texas Wesleyan University, Fort Worth, TX, 2020.
- [7] Awad, Neveen F., and Arik Ragowsky. "Establishing trust in electronic commerce through online word of mouth: An examination across genders." *Journal of management information systems* 24, no. 4 (2008): 101-121.
- [8] Quốc, Anh Nguyễn. "Hành vi sử dụng Chatbot của thế hệ gen Z tại các ngân hàng thương mại Việt Nam." *Tạp chí Kinh tế và Phát triển* 309 (2) (2023): 2-11.
- [9] Bình, Nguyễn Ngọc. "Dùng lý thuyết tập thô và các kỹ thuật khác để phân loại, phân cụm văn bản tiếng Việt." *Kỷ yếu hội thảo ICT. rda* 4 (2004).
- [10] Sharma, Rakesh Kumar, and Manoj Joshi. "An analytical study and review of open source chatbot framework, rasa." *Int. J. Eng. Res* 9, no. 06 (2020): 1011-1014.
- [11] Mai, Trang Nguyen Thi, and Shcherbakov Maxim. "Enhancing Rasa NLU model for Vietnamese chatbot." *International Journal of Open Information Technologies* 9, no. 1 (2021): 31-36.