

Customer Support Chatbot

Nhóm 4

Thành viên

20127206 - Vũ Đình Duy Khánh

20127384 - Văn Khánh Tường

20127443 - Nguyễn Hồ Hữu Bằng

20127652 - Hoàng Minh Triết

Giới thiệu

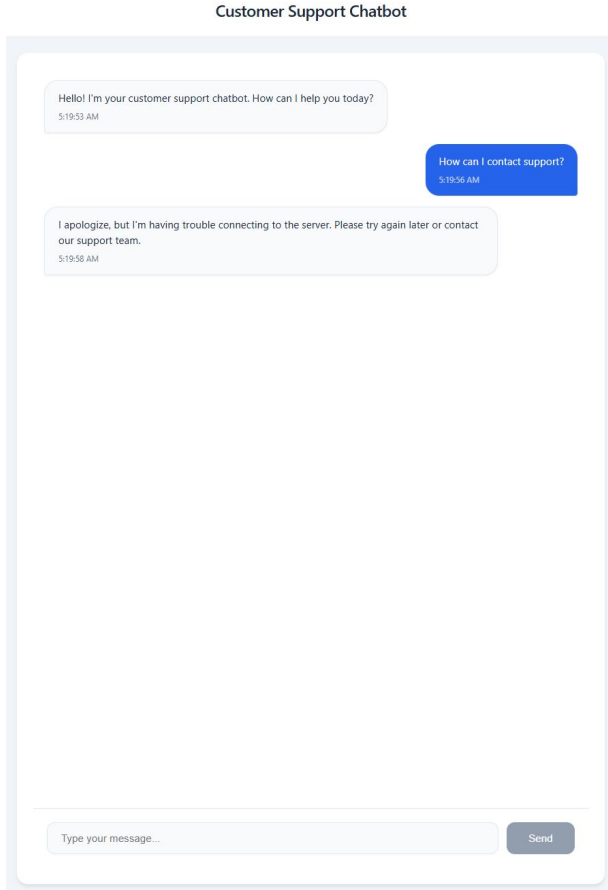
LangChain - Xử lý Ngôn ngữ & Truy xuất Dữ liệu

♦ **LangChain** là framework giúp xây dựng ứng dụng AI sử dụng mô hình ngôn ngữ lớn (LLM) với khả năng truy xuất dữ liệu và tạo phản hồi một cách thông minh.

Cách LangChain hỗ trợ chatbot:

- **Tích hợp LLM:** Kết nối với OpenAI GPT, LLaMA, Claude,... để tạo câu trả lời.
- **Truy xuất dữ liệu (Retrieval Augmented Generation - RAG):** Tìm kiếm thông tin từ cơ sở dữ liệu hoặc Vector Store (FAISS, Chroma).
- **Memory (Bộ nhớ hội thoại):** Giữ ngữ cảnh của cuộc trò chuyện để phản hồi liên tục.
- **Chain (Chuỗi xử lý):** Xây dựng quy trình xử lý từng bước, ví dụ: hiểu câu hỏi → tìm dữ liệu → tạo câu trả lời.

FrontEnd



- Sử dụng React.js cho Front End
- Giao tiếp với Back End bằng FastAPI

Giới thiệu LangChain/LangGraph

LangChain là framework giúp xây dựng ứng dụng AI sử dụng mô hình ngôn ngữ lớn (LLM) với khả năng truy xuất dữ liệu và tạo phản hồi một cách thông minh.

- **Tích hợp LLM:** Kết nối với OpenAI GPT, LLaMA, Claude,... để tạo câu trả lời.
- **Truy xuất dữ liệu (Retrieval Augmented Generation - RAG):** Tìm kiếm thông tin từ cơ sở dữ liệu hoặc Vector Store (FAISS, Chroma).
- **Memory (Bộ nhớ hội thoại):** Giữ ngữ cảnh của cuộc trò chuyện để phản hồi liên tục.
- **Chain (Chuỗi xử lý):** Xây dựng quy trình xử lý từng bước, ví dụ: hiểu câu hỏi → tìm dữ liệu → tạo câu trả lời.

LangGraph là một thư viện mở rộng cho LangChain giúp xây dựng chatbot theo hướng đồ thị (Graph-based).

- **Quản lý hội thoại phức tạp hơn:** Nếu chatbot chỉ dựa trên LangChain, nó có thể bị giới hạn trong việc xử lý hội thoại không tuyến tính.
- **Xử lý đa nhiệm (Branching Logic):** Ví dụ, chatbot có thể **điều hướng** người dùng đến bộ phận phù hợp dựa trên câu hỏi của họ.
- **Tích hợp quy trình xử lý (Workflow Orchestration):** Khi một truy vấn yêu cầu nhiều bước (kiểm tra đơn hàng, hủy đơn, liên hệ nhân viên,...), LangGraph giúp tổ chức luồng xử lý rõ ràng.

LangChain + LangGraph trong Chatbot

- **LangChain:** Xử lý truy vấn, truy xuất thông tin, và tạo phản hồi từ LLM.
- **LangGraph:** Điều hướng hội thoại, xử lý nhiều bước phức tạp theo dạng luồng đồ thị.

BackEnd - Thu thập dữ liệu

Dữ liệu có **27 intent khác nhau**, bao gồm:

- **Xử lý đơn hàng:** `place_order`, `cancel_order`, `change_order`
- **Vận chuyển:** `change_shipping_address`, `set_up_shipping_address`, `delivery_options`, `track_order`
- **Thanh toán:** `check_payment_methods`, `payment_issue`
- **Hoàn tiền:** `check_refund_policy`, `get_refund`, `track_refund`
- **Tài khoản:** `create_account`, `edit_account`, `delete_account`, `switch_account`, `recover_password`
- **Liên hệ:** `contact_customer_service`, `contact_human_agent`
- **Khác:** `review`, `complaint`, `newsletter_subscription`, `registration_problems`

-> Dữ liệu hỗ trợ nhiều mục đích khác nhau trong chăm sóc khách hàng.

-> Có thể sử dụng để huấn luyện chatbot AI nhằm hỗ trợ người dùng trong thương mại điện tử.

BackEnd - Thu thập dữ liệu

Tập dữ liệu có **11 danh mục chính** liên quan đến giao dịch thương mại điện tử:

order (đơn hàng), **shipping** (vận chuyển), **cancel** (hủy đơn), **invoice** (hóa đơn), **payment** (thanh toán), **refund** (hoàn tiền), **feedback** (phản hồi), **contact** (liên hệ), **account** (tài khoản), **delivery** (giao hàng), **subscription** (đăng ký dịch vụ)

-> Dữ liệu bao phủ nhiều lĩnh vực liên quan đến dịch vụ khách hàng trong thương mại điện tử.

-> Mô hình có thể hỗ trợ người dùng xử lý các vấn đề phổ biến như đơn hàng, thanh toán, tài khoản, giao hàng,...

Độ dài trung bình của instruction: 45.72 từ

Độ dài trung bình của response: 621.97 từ

-> Instruction ngắn gọn (~46 từ), phù hợp với câu hỏi của người dùng.

-> Response dài (~622 từ), chúng tôi phản hồi chi tiết, có thể là giải thích, hướng dẫn hoặc quy trình xử lý.

-> Mô hình AI cần có bộ nhớ cảnh lớn để xử lý response dài.

Back End - Phân tích, xử lý dữ liệu

Phân tích và xử lý giá trị thiếu (`analyze_missing_values`)

- Kiểm tra các cột có giá trị thiếu.
Ghi log các cột có trên 0% giá trị thiếu, sau đó loại bỏ
- Nếu một cột có hơn 50% giá trị bị thiếu, cột đó sẽ bị loại bỏ.
Với các cột số, giá trị thiếu được thay bằng giá trị trung vị (`median`).
 - Trung vị giúp giảm ảnh hưởng của outliers
 - Trung vị (`median`) là giá trị nằm ở giữa khi sắp xếp dữ liệu theo thứ tự tăng dần.
 - So với trung bình cộng (`mean`), trung vị ít bị ảnh hưởng bởi các giá trị cực đoan (`outliers`).
- Với các cột phân loại (`categorical`), giá trị thiếu được thay bằng giá trị xuất hiện nhiều nhất (`mode`).

Xử lý dữ liệu trùng lặp (`handle_duplicates`)

Xử lý outliers (ngoại lệ) (`handle_outliers`)

- Tính toán khoảng tứ phân vị (`IQR` – Interquartile Range) để phát hiện outliers.
- Các giá trị nằm ngoài phạm vi (`Q1 - 1.5*IQR, Q3 + 1.5*IQR`) sẽ bị clip lại.

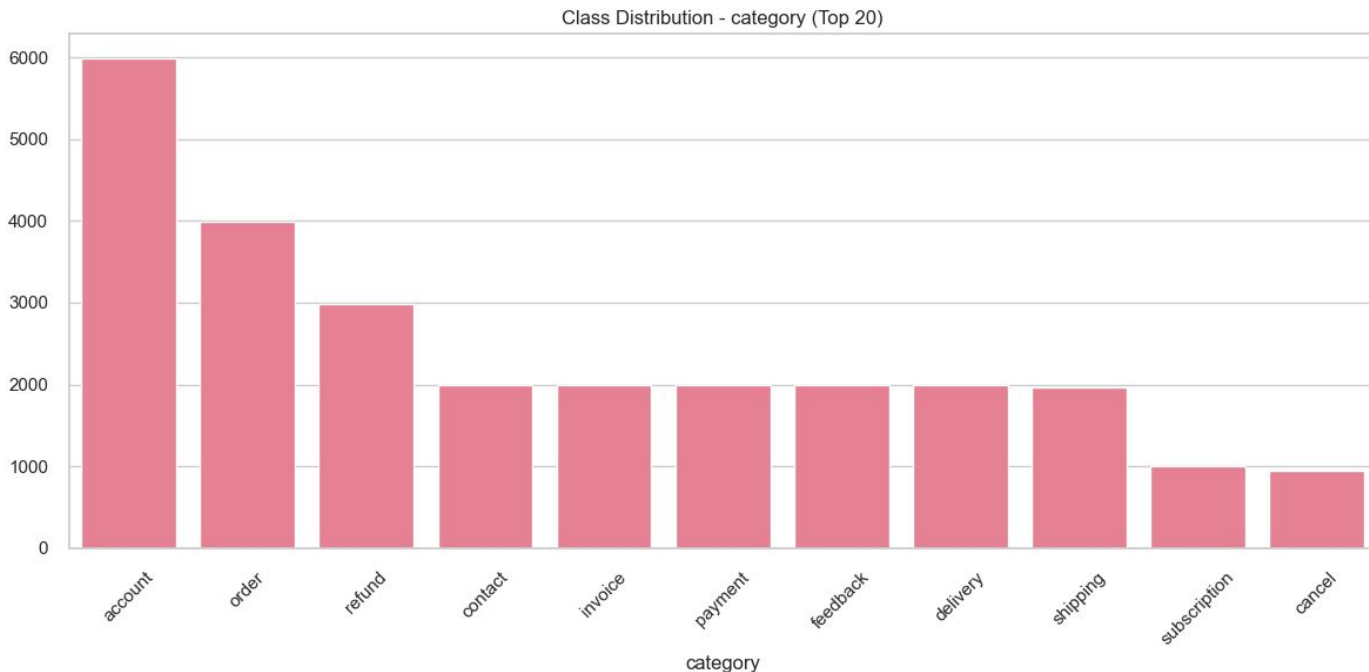
Chuẩn hóa dữ liệu (Normalization) (`normalize_data`)

- Chuẩn hóa dữ liệu số bằng **StandardScaler** (z-score normalization).
 - Giữ được phân phối dữ liệu, hiệu quả cho mô hình tuyến tính
 - Chuẩn hóa dữ liệu số bằng **MinMaxScaler** (đưa về khoảng [0,1]).

Back End - Phân tích, xử lý dữ liệu

Phân tích cân bằng dữ liệu trong các cột phân loại (`analyze_class_balance`)

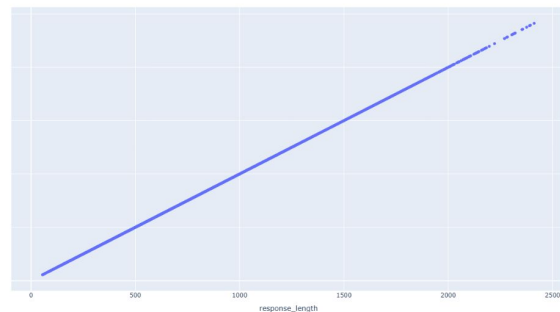
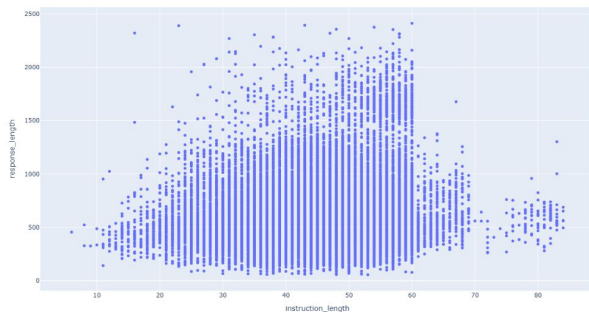
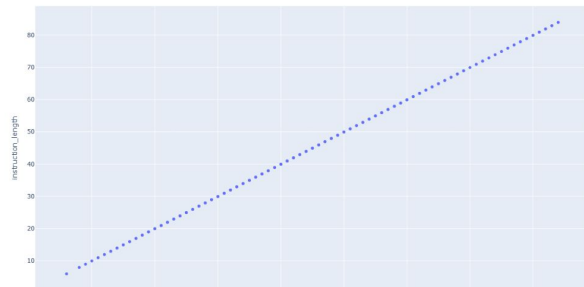
- Xác định tỉ lệ phân bố giữa các nhóm trong từng cột phân loại.
- Tạo biểu đồ phân bố (`barplot`).



Back End - Phân tích, xử lý dữ liệu

Tạo biểu đồ tương quan giữa các biến số (Pairplot) (`create_pairplot`)

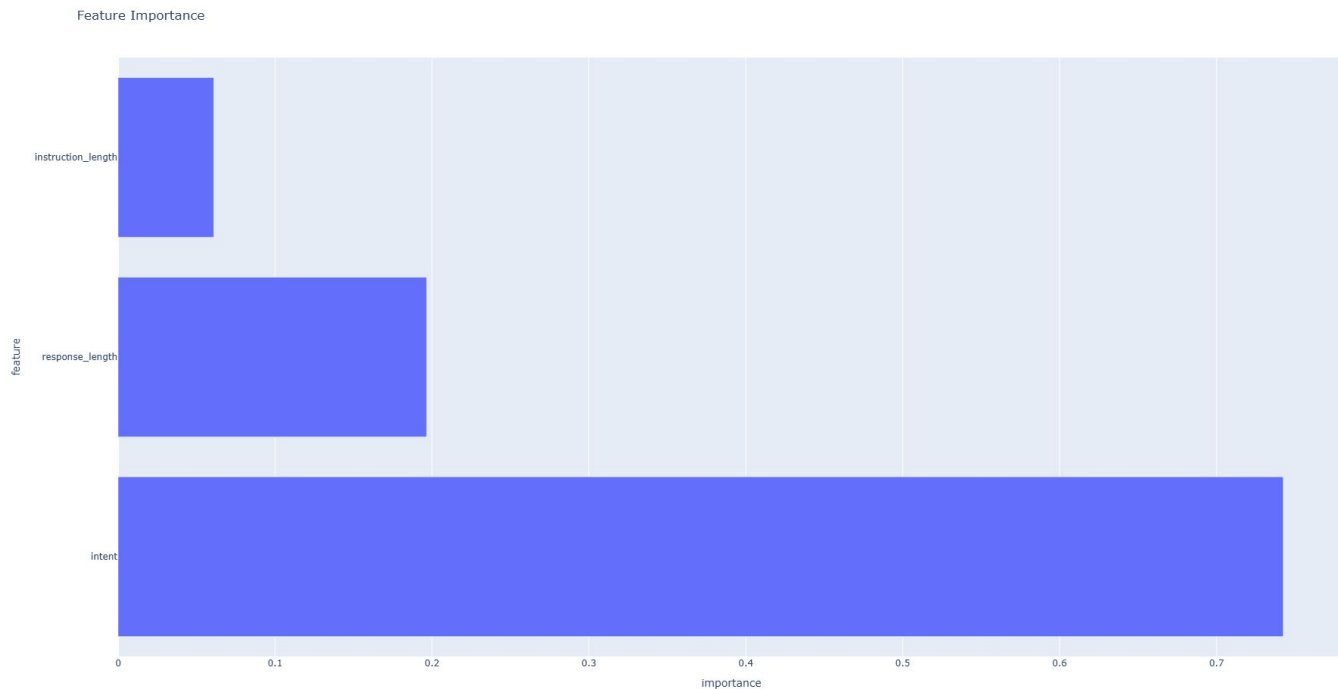
- Tạo biểu đồ `pairplot` giữa các cột số để quan sát mối quan hệ giữa các đặc trưng.
 - Mối quan hệ giữa `instruction_length` và `response_length` có xu hướng dần trải, nhưng vẫn thể hiện một số cụm dữ liệu rõ ràng.
 - Dữ liệu có độ phân tán cao, cho thấy không phải lúc nào `instruction_length` cũng tỷ lệ thuận với `response_length`.



Back End - Phân tích, xử lý dữ liệu

Phân tích tầm quan trọng của đặc trưng (`analyze_feature_importance`)

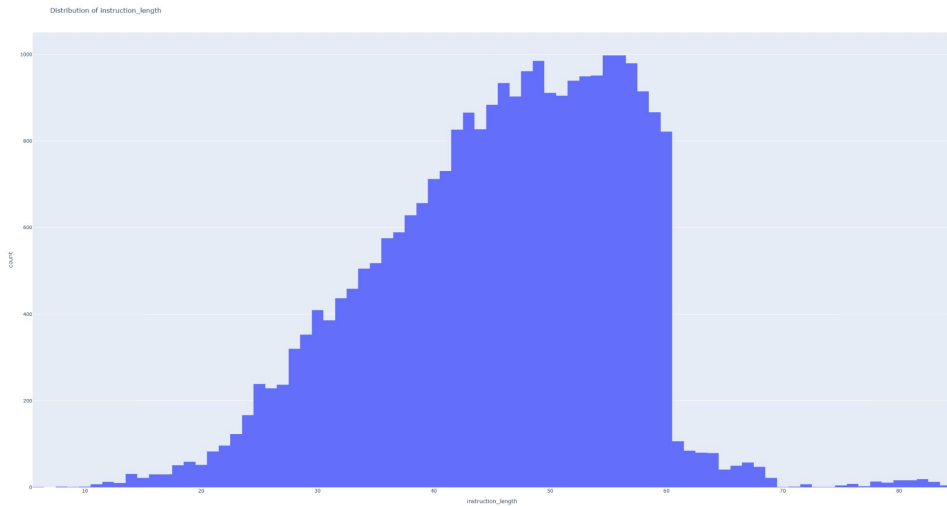
- Tính toán ma trận tương quan giữa các cột số.
- Vẽ biểu đồ của ma trận tương quan.



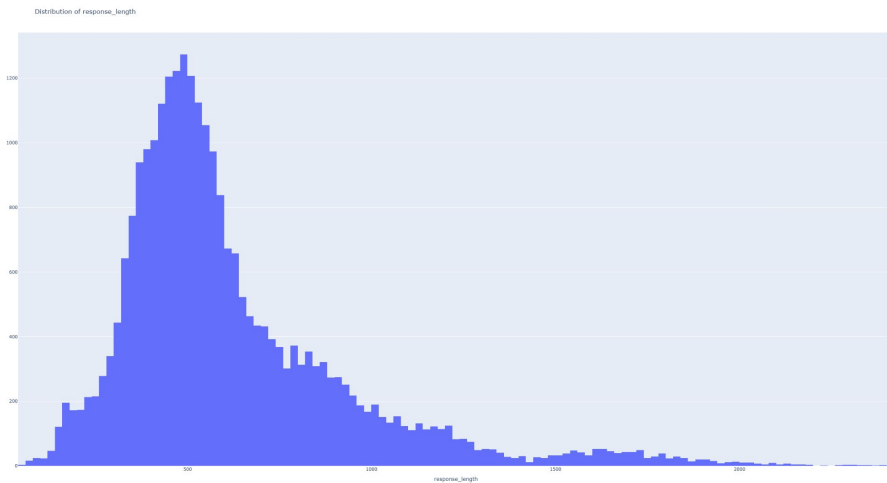
Back End - Trực quan hóa dữ liệu

Tạo biểu đồ phân phối (`create_distribution_plots`)

- Tạo biểu đồ histogram kèm đường KDE cho các cột số.
 - Hầu hết `instruction_length` tập trung trong khoảng 30-55, sau đó giảm dần. Những giá trị cực đoan (>70) có thể là outliers hoặc các câu lệnh rất dài.
 - Hầu hết các `response_length` có độ dài trung bình khoảng 500 từ, nhưng vẫn có một số response rất dài (>1500). Những giá trị này có thể là các câu trả lời phức tạp hoặc cần nhiều thông tin.



Instruction_length

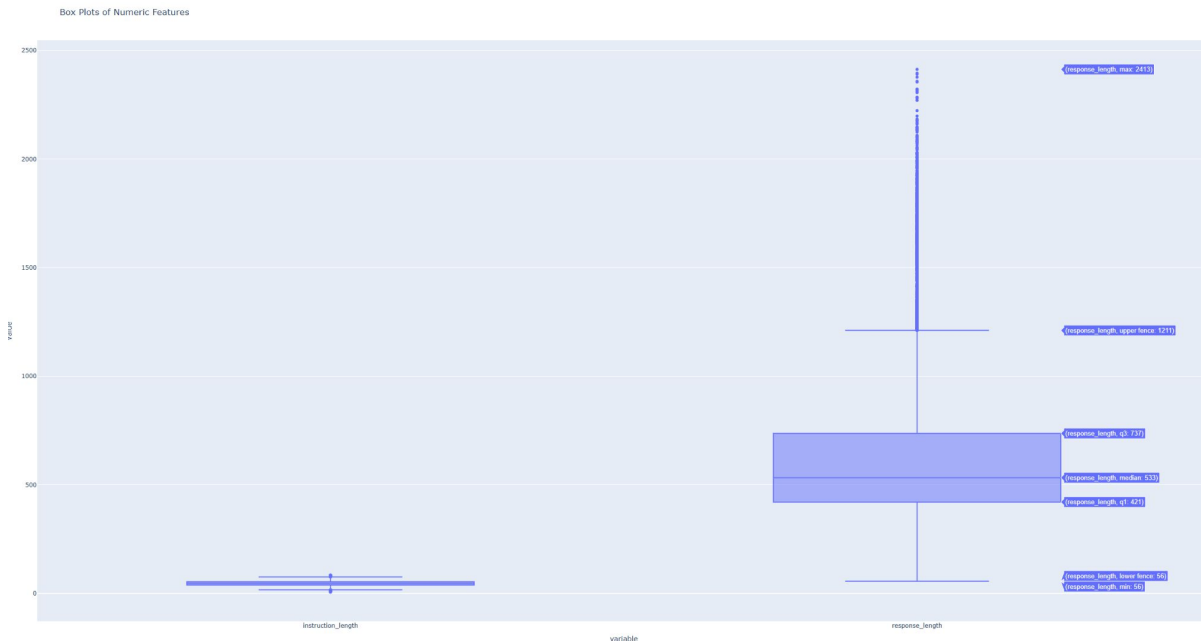


Response_length

Back End - Trực quan hóa dữ liệu

Tạo box plot (`create_box_plots`)

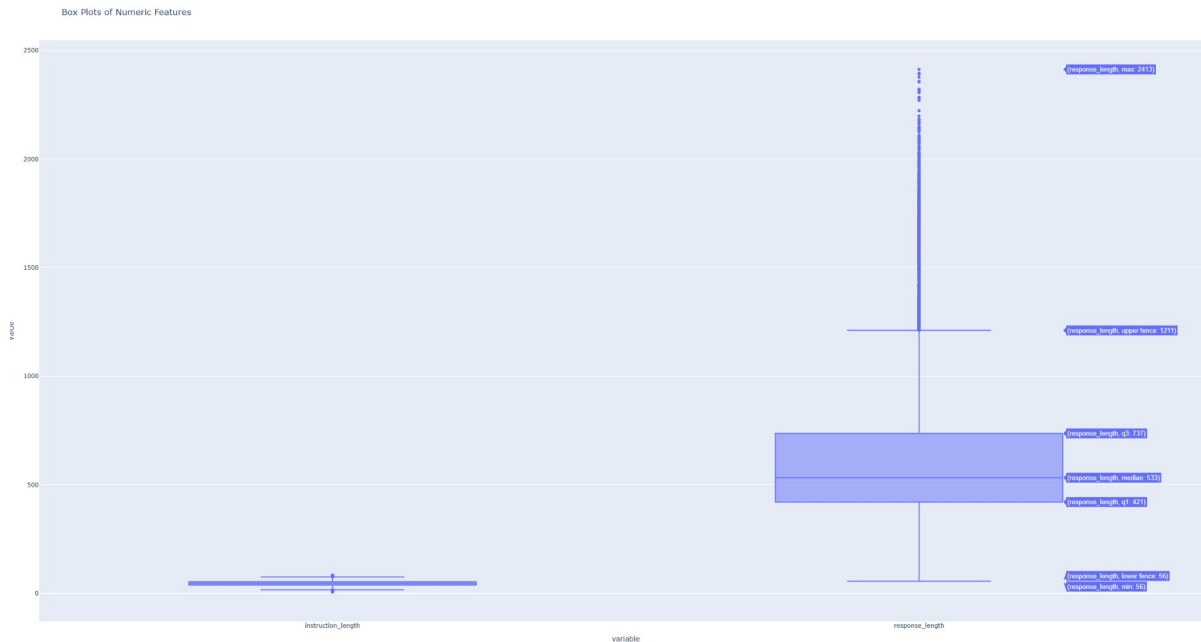
- Tạo box plot cho các đặc trưng số để phân tích phân bố dữ liệu và phát hiện outliers.
 - `response_length` có phạm vi biến động rộng hơn rất nhiều so với `instruction_length`.
 - Có nhiều response dài bất thường (>1211), có thể do nội dung yêu cầu chi tiết hơn hoặc lỗi tạo dữ liệu.
 - Instruction length khá ổn định, không có outliers lớn.



Back End - Trực quan hóa dữ liệu

Tạo heatmap tương quan (`create_correlation_heatmap`)

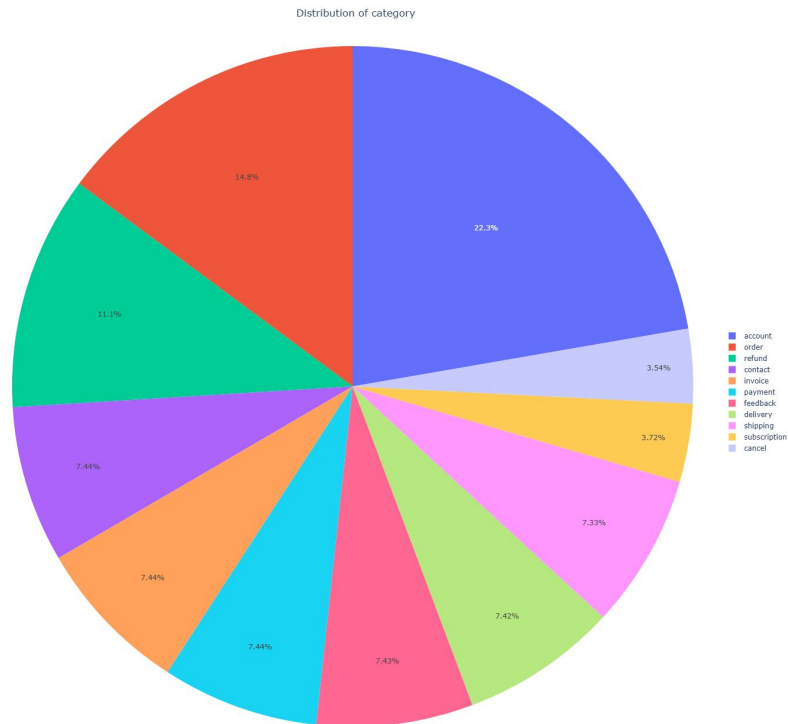
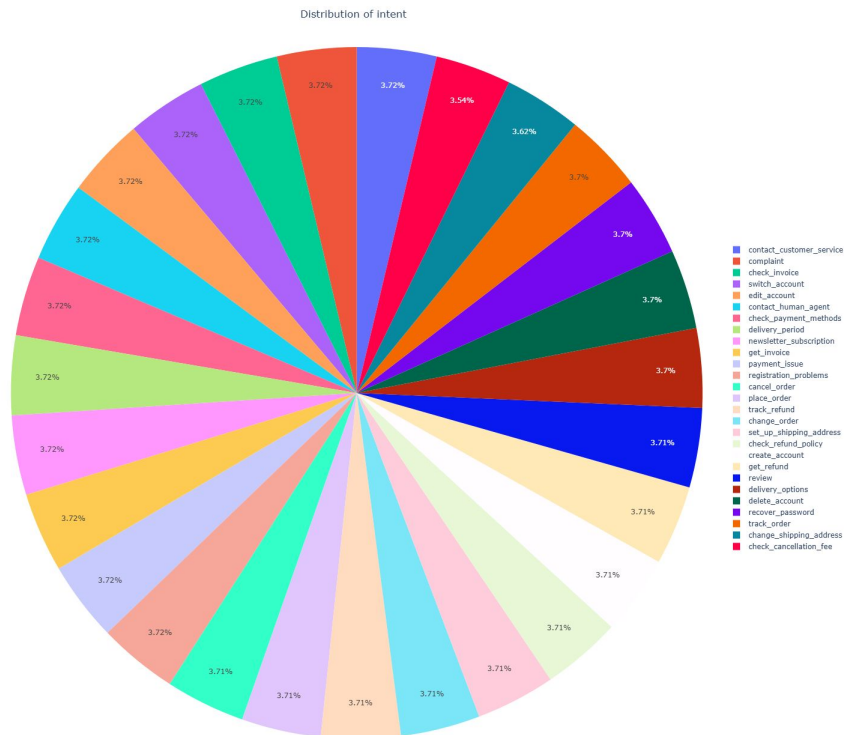
- Tính ma trận tương quan giữa các đặc trưng số.



Back End - Trực quan hóa dữ liệu

6. Trực quan hóa dữ liệu phân loại (`create_categorical_plots`)

- Vẽ biểu đồ bar và pie cho các cột dữ liệu phân loại.



Back End

Phân tích dữ liệu sau tiền xử lý và trực quan

- **Dữ liệu sạch:** Không có giá trị thiếu hoặc trùng lặp.
- **Dữ liệu chuẩn hóa:** Giá trị trung bình gần 0 và độ lệch chuẩn ≈ 1 giúp mô hình hoạt động tốt hơn.
- **Cấu trúc phù hợp cho AI:** Có thể sử dụng để huấn luyện mô hình NLP, đặc biệt trong tìm kiếm vector.

-> Dữ liệu này đã sẵn sàng để huấn luyện mô hình AI hoặc lưu vào Vector Store (FAISS/Chroma) để tìm kiếm hiệu quả.

Back End - Chuẩn bị dữ liệu LangChain

Chuyển đổi dữ liệu thành tài liệu LangChain

- Mỗi dòng dữ liệu được chuyển thành một **Document** có:
 - Dữ liệu thô sau khi tiền xử lý cần được chuẩn bị dưới dạng tài liệu (**Document**) để sử dụng trong các ứng dụng truy vấn thông minh hoặc tìm kiếm thông tin.
 - Nội dung: Instruction, response, category, intent.
 - Metadata: Thông tin bổ sung như category, intent, flags
- Tạo và lưu Vector Store
 - Tạo **FAISS** vector store (tìm kiếm nhanh).
 - Hoạt động trên Ram để tìm kiếm thông tin nhanh chóng
 - Tạo **Chroma** vector store (có thể lưu lại trên ổ đĩa).
 - Lưu trữ lâu dài trên ổ cứng

Công việc tiếp theo

- Xây dựng Model phù hợp với dữ liệu đã thu thập
 - Sử dụng mô hình từ Hugging Face do LangChain hỗ trợ
 - Xây dựng và tích hợp mô hình Transformer tự huấn luyện với LangChain
 - Sử dụng BERT tốt cho phân loại intent
 - Không cần sử dụng API bên thứ 3
 - Miễn phí
- Training Model dựa trên dữ liệu đã chuẩn bị
- Đánh giá Model dựa trên các điểm số hiệu suất như :
 - Accuracy
 - Precision
 - Recall
 - F1 Score
- Tối ưu hóa Model
- Tối ưu hóa FrontEnd

Cảm ơn thầy đã theo dõi