

PCI Query Flow with LangGraph

Project Overview

This project implements a **LangGraph workflow** that simulates a Predictive Customer Intelligence (PCI) pipeline. It captures customer queries, segments them based on intent, and generates personalized suggestions — while maintaining context across conversations using memory.

Features

- **LangGraph with 3 processing nodes**
 - **Node 1:** Captures user query + retrieves memory context
 - **Node 2:** Applies mock customer segmentation logic
 - **Node 3:** Returns suggestions based on customer segment
- **Memory persistence** using `ConversationBufferMemory`
- **Mock Predictive Customer Intelligence logic**
- Written in **Python** and supports both **Jupyter Notebook** and standalone script execution

Project Structure

`task2/` Main project directory

`app.py` Main application script

`task2.ipynb` Jupyter Notebook with full code and visualization

`PCI_LangGraph_Workflow_Diagram.png` Visual architecture of the workflow

`README.pdf` This file

How It Works

1. User Query Input

A customer query is passed to the workflow (e.g., "I want to buy a product" or "I have an urgent issue").

2. Context Retrieval

Memory stores and retrieves previous interactions using `ConversationBufferMemory`.

3. Customer Segmentation

Mock logic segments customers as:

- High-Priority Customer (e.g., queries with "urgent" or "problem")
- Potential Buyer (e.g., queries with "buy" or "purchase")
- General Inquirer (all other queries)

4. Suggestion Generation

Based on the segment, the system returns an appropriate response, such as offering support escalation or product discounts.

Requirements

- Python 3.8+
- Jupyter Notebook (optional, for running `task2.ipynb`)
- Required packages:
 - `langchain`
 - `langgraph`
 - `pydantic`

Setup Instructions

Refer to the `setup.pdf` document for detailed instructions on setting up the environment, installing dependencies, and running the application.

Quick Start

1. Create and activate a virtual environment:

```
python -m venv venv
source venv/bin/activate    % On Windows: venv\Scripts\activate
```

2. Install dependencies:

```
pip install langchain langgraph pydantic
```

3. Run the application:

```
python app.py
```

Or open `task2.ipynb` in Jupyter Notebook.

Running the Application

- **Script Mode:** Run `python app.py` to start the interactive chatbot. Enter queries and type `exit` to quit.
- **Notebook Mode:** Open `task2.ipynb` in Jupyter Notebook and execute the cells to explore the workflow interactively.

Example Usage

PCI Query Flow Chatbot
Type 'exit' to quit.

```
Enter your query: I have an urgent issue
Context:
User Query: I have an urgent issue
Segment: High-Priority Customer
Suggestion: Offer immediate support ticket escalation.
```

Future Work

- Integrate a large language model (LLM) for advanced segmentation and suggestion generation.
- Add support for external data sources to enhance PCI logic.
- Implement a graphical user interface for better user interaction.