Restaurant Recommender AI Assistant

Objective:

The number of restaurants has increased significantly. And each restaurant has its own specialities in terms of the type of restaurants, cuisines available etc. With these many options, choosing a restaurant becomes a very tedious task for a user. The objective of this project is to build an AI based assistant to help the users simplify the decision-making process. The Restaurant Recommender AI system is designed to interact with the user and find his requirements (location, nature of restaurant, cuisines of interest etc.). The AI assistant looks at the data of available restaurants, picks the best restaurants by ratings as per the user's preferences. And recommends the top restaurants to the user so that the user can make his choice instantly. The dataset used for this project is a slightly altered data from the public dataset available at https://www.kaggle.com/datasets/abhijitdahatonde/zomato-restaurants-dataset .

System Design:

The project is implemented using Python programming language. The core functionality is powered by the OpenAI gpt-4o-mini model, which is utilized for natural human style conversations and for generating recommendations based on user input.

The Restaurant Recommender AI system has the following key components or layers.

a) **Conversation and Information Gathering:** In this layer, the AI assistant will interact with the user in natural language. Through a conversational flow, it will ask relevant questions to gather information about the user's requirements for the restaurants. In this stage, the assistant will fetch the finalized user preferences.

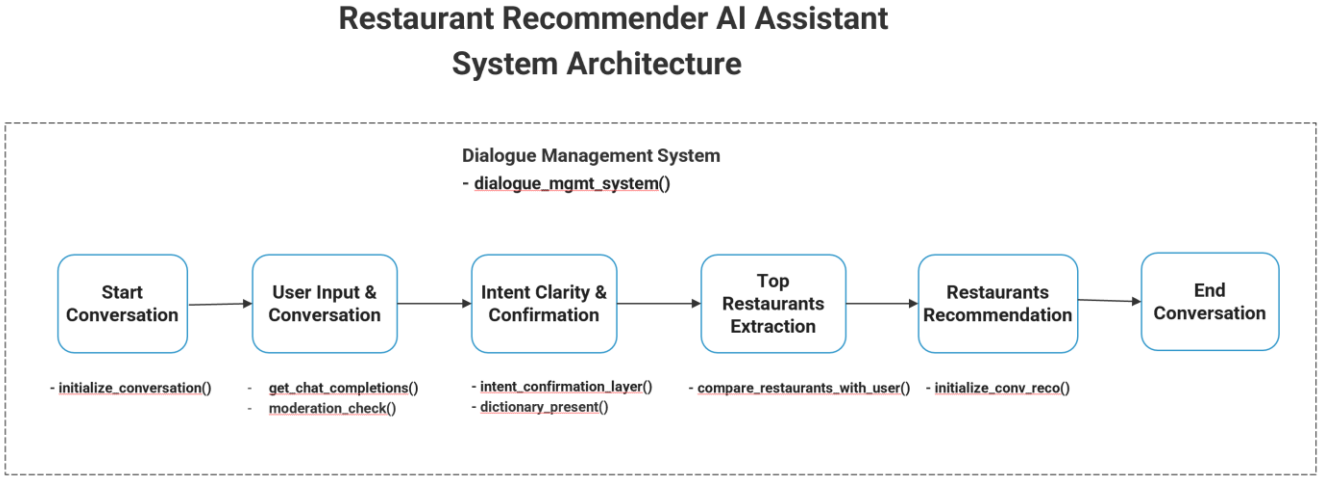
b) **User Profile Extraction:** In this layer, the AI assistant extracts key information from the conversation to build a user profile that reflects the user's restaurant preferences (restaurant type, cuisine, location, budget etc.).

c) **Personalized Recommendation:** In this layer, the AI assistant looks at the restaurants data available. And picks the top restaurants (by rating) based on the user's preferences. The personalized recommendations are provided to the user and the AI assistant helps the users further in any additional queries against these.

d) **Moderation:** Content moderation layer is implemented using OpenAI's moderation API. Moderation is applied to user input and model response to user, to ensure a safe and secure conversation.

e) **Dialogue Management System:** This layer is the core of the Recommendation system. This layer carries out the orchestration across the different layers and

modules present. This simulates a conversational flow between the user and the system and helps the user in fulfilling the task of finding the restaurant of his choice based on his preferences.



Major functions implemented in the Restaurant Recommendation AI System:

- initialize_conversation(): Initializes the variable conversation with the system message.
- get_chat_completions(): Takes the ongoing conversation as the input and returns the response by the assistant.
- moderation_check(): Checks if the user's or the assistant's message is inappropriate. If any of these is inappropriate, it ends the conversation.
- intent_confirmation_layer(): Evaluates if the assistant has captured the user's profile clearly.
- dictionary_present(): Checks if the final understanding of the user's profile is returned by the assistant as a Python dictionary.
- compare_restaurants_with_user(): Compares user profile with restaurants dataset and comes back with the top 5 recommendations.
- initialize_conv_reco(): Initializes the recommendations conversation.
- dialogue_mgmt_system(): The final orchestration function that manages the whole conversation with the user using the other functions.

Steps of Project Execution:

The pre-requisites for executing this project are Python and OpenAI API key.

To get started with the Restaurant Recommender AI assistant, execute the jupyter notebook submitted for this project. And follow the input prompts generated by the dialogue management system. To end the conversation at any point type “exit” in the input prompt.

Conclusion:

The Restaurant Recommender AI project demonstrates the capabilities of natural language processing and AI in providing personalized recommendations in the restaurant domain. Some of the challenges encountered were around getting structured output from the model response. This is currently handled with effective prompt tuning, applying techniques like few-shot prompting etc. It can be enhanced further by making use of model features like function calling to get structured output seamlessly.With further refinement and enhancements, the system can offer even more accurate and tailored recommendations, thereby enhancing the dining experience for users.

Though the current project was a demonstration of restaurant recommendations, similar AI assistants can be built across other domains as well by applying same system architecture and workflows after adding the domain relevant customizations.