Movie Assist AI

Objective: The primary objective of the Movie Recommendation Assist AI Project is to provide personalized movie recommendations to the users. The system aims to understand user preferences and suggest movies that align with their tastes.

Design: The model is designed using the following stages and different functions present in different stages.

1. **Stage 1: Intent Clarity and Intent Confirmation Layer**
   This layer helps in identifying the user's requirements and passing it on the product mapping layer.
   The functions that are used in this layer are:
   1. **initialize_conversation():** This function initializes the variable conversation with the system messages. Using prompt engineering and chain of thought reasoning, the function will enable the chatbot to keep asking the questions until the user requirements have been captured in a dictionary.
   2. **get_chat_completions():** This function takes the ongoing conversations as the input and returns the response by the assistant.
   3. **iterate_response():** We will create a small helper function to ensure the model's response is consistent.
   4. **moderation_check():** This checks if the user's or the assistant's response is inappropriate. If any of these is inappropriate then we can end the conversation.
   5. **intent_confirmation_layer():** This function takes the assistants's response and evaluates if the chatbot has captured the user's profile clearly. Specifically it checks if the following properties have been captured or not: Duration, Title Year, IMDb Score, Aspect Ratio, Genre.
   6. **dictionary_present():** This function checks if the final profile created by the chatbot is in the form of a dictionary or not.

2. **Stage 2: Product Mapping and Information Extraction Layer**
   In this layer we take the output of the previous layer/stage, that is, user requirements , which is in the form of a python dictionary. Next we will extract the top 3 movies as per the need.
   The function that is used in this layer are:

1. **product_map_layer():** This function is responsible for extracting key features and criteria from movie description.
2. **compare_movie_with_user():** this function will compare the user's profile with the different movies and come back with the recommendations.
3. **product_validation_layer():** This function verifies that the movie recommendations are good, has a score greater than 2 and matches the user's requirements.

3. **Stage 3: Product Recommendation Layer**
   Finally we come to the product recommendation layer. It takes the output from compare_movie_with_user() function in the previous layer and provides the recommendation to the user.

We then combine all the stages defined above.
Stage 1 + Stage 2 + Stage 3

We create a **dialogue_mgmt_system()** function that contains the logic of how the different layers would interact with each other. This will be the function that we will use to initialize the chatbot.

Implementation:

Technologies used:

1. **Programming Language:** Python
2. **Libraries Used:** Pandas, Openai, os, json, ast
3. **API:** IMDb
4. **Open AI Model Used:** gpt-3.5-turbo

Steps:

1. Data Collection: Collect the movie dataset from external APIs and store it in google drive.
2. Model Training: Train the model to understand the user requirements and categorize user inputs. Develop and train the model using historical data.
3. Recommendations: Give recommendations to the user based on their requirements.

Challenges:

Following challenges were faced during the creation of the model.

1. Data Quality and Availability: Ensuring the availability and accuracy of the movie data from external sources was a challenge. Handling inconsistencies and missing data was another.

2. Personalization: Balancing the need for personalized recommendations with user privacy. Implementing secure data handling and storage practices.

Lessons Learned:

Following lessons were learned as part of this project.

1. User Centric Design: Importance of understanding user behavior and preferences to provide relevant recommendations. Continuous user feedback is crucial for improving recommendation accuracy.
2. Iterative Development: Adopting an iterative approach to model training and system improvement. Regularly updating the model with new data to maintain relevance.
3. Performance Optimization: Necessity of optimizing algorithms for real time performance.