Proposal: Enhancing Product Recommendations Using Large Language Models (LLMs)

1. Introduction

As a leading e-commerce platform, personalizing customer interactions is key to maintaining a competitive edge. Leveraging Large Language Models (LLMs) Like Gemma-7b-It(GROQ) allows us to enhance product recommendations, automate customer support, and optimize internal processes. This proposal outlines a strategy to integrate LLMs into our recommendation system, supported by a comprehensive codebase and deployment plan.

2. Objective

- **Enhance Product Recommendations(RAG Model):** Utilize LLMs to personalize product recommendations based on customer preferences, historical data, and real-time interactions.
- **Code Implementation(Fine Tuning):** Demonstrate the fine-tuning of a pre-trained LLM on a relevant dataset to improve recommendation relevance and conversion rates.

3. Technical Approach-1(Using RAG Model System)

3.1. Data Preprocessing

- **Dataset:** The provided dataset includes customer reviews,Summary, and interaction history. The preprocessing steps were executed using the following notebooks and scripts:
  - `Cleaning_data.ipynb`**:** This notebook handles the initial data cleaning steps:
    - **Missing Values:** Removed or imputed missing values.
    - **Text Normalization:** Removed Stop Words From Review Text.
  - `Grouped_Data.ipynb`**:** This notebook groups the cleaned data by customer ID, providing a consolidated view of each customer's interactions:
    - **Aggregation:** Combined customer reviews, for each product And Aggregating In One ReviewText.

3.2 Model Architecture

3.2.1 RAG (Retrieval-Augmented Generation) Model

- **Base Model:** Gemma-7b-It was chosen as the base model due to its advanced language understanding and generation capabilities.

  To effectively integrate customer reviews into the recommendation system, a Retrieval-Augmented Generation (RAG) approach will be employed. The system will consist of two RAG models:

- **RAG1:** Processes the detailed review text to generate insightful recommendations.

- **RAG2:** Utilizes the summary of the reviews for a more concise recommendation generation.

System Flow Diagram:

- **Chain Mechanism:** The outputs (O1 and O2) from RAG1 and RAG2 are combined in a chain, which integrates these insights with additional query information.
- **Final Model Integration:** The chained output is then passed to the final model, which synthesizes the recommendations for the end user.
- **Flow Diagram**: Rag_Chain_Flow.png

3.3. Integration with Recommendation Engine

- **RAG (Retrieval-Augmented Generation) System:** The Gemma-7b-It model was integrated into a RAG-based recommendation system:
  - **Vector Store Creation:** Implemented in `Vector_Store.ipynb`, the vector store enables efficient retrieval of relevant product information:
    - **Embedding Generation:** Converted product descriptions and customer queries into vector embeddings using Hugging Face Embeddings
    - **Real-Time Retrieval:** The system retrieves and ranks relevant products based on vector similarity, allowing for personalized recommendations.
  - **Integration with Final Engine:**
    - The RAG system is Intergrated With Final Prompt To Recommend Products
    - The integration ensures that recommendations are not only based on historical data but also on real-time customer interactions.

3.4. Code Implementation

- **Data Preprocessing Notebooks (`Cleaning_data.ipynb` & `Grouped_Data.ipynb`):**
  - These notebooks document the data preparation process, ensuring transparency and reproducibility.
  - Key steps are annotated with explanations to provide context and understanding of the data transformation process.
- **Vector Store Creation Implementation (`Vector_Store.ipynb`):**
  - This notebook outlines the creation and utilization of the vector store for efficient product retrieval.
  - The RAG approach is explained in detail, with code examples demonstrating how to integrate it into the recommendation pipeline.
- **Recommendation System Implementation**(`Recommendation_System.ipynb`)
  - This notebook utilizes the vector store created for reviewText and summary

    - Create the final RAG Chain And Get Recommendation.

3.5: Deployment Strategy:

- The Deployment Strategy Is Given In File aws_deployment.txt
- **Step 1**: Create an AWS Lambda function
- **Step 2**: Create an AWS API Gateway
    - **Resource** name: recommendation
    - **HTTP method**: POST
    - **Integration type**: Lambda function
    - **Lambda function**: lambda_function
- **Step 3**: Deploy the **API Gateway**

3.6. Model Evaluation On Query

1) `print(recommendation("LED Lights"))`

```
## Top Product Recommendations Based On Provided Reviews:

**1. LEDGLOW KIT Festoon LED Light Bulbs:**

- This product received the highest praise in the provided reviews, with one reviewer calling them "FREAKING AWESOME!"
- Customers praised their quality, effectiveness, and significant lighting upgrade.


**2. Led Bulbs Festoon LED Light Bulbs:**

- This product is praised for its quality, value, and high-quality LEDs.
- Reviewers consider them a must-have for anyone seeking high-quality LEDs.


**3. LED Stop Light Bulbs, 31mm Festoon LED Light Bulbs for Cars, Truck, SUV, White:**

- This product wasn't featured in the provided reviews, but it's likely a reliable option based on the positive feedback received by other Festoon LED light bulbs.

**Key Considerations:**

- Based on the provided reviews, all three recommended products offer significant upgrades in lighting quality and are considered reliable.
- Price and specific features (such as brightness and durability) should be considered when making your final decision.
- Ensure the selected bulbs are compatible with your vehicle or application.
```

2) `print(recommendation("Air Filter"))`

```
**Product Names From Top 3 Best Air Filter Recommendations:**
```

1. **K&N Air Filter**

2. **AEM Air Filter**

3. **Fram Air Filter**

3). `print(recommendation("Car Duster"))`

Product Name: California Car Duster

4. Technical Approach-2 (Using Fine Tune Model(Computationally Expensive Required Paid Collab For Training))

Data Preparation: Loaded and split the dataset into training and validation sets, ensuring balanced representation of customer preferences

Model Training: Fine-tuned t5(Like Llama) LLM Model using customer reviews and summary features to capture nuanced customer preferences.

Fine-Tuning Script (train.py):

The script details the entire process of fine-tuning Of T5_Model(Like LLAMA) on customer review data, from data loading to model training and evaluation.

Comments and explanations are included to clarify the rationale behind each step, including parameter choices and model adjustments.

5. Conclusion

This proposal presents a two comprehensive approach **(RAG-Model And Fine Tuning-Model)** to enhancing product recommendations using LLMs. The provided codebase, including detailed scripts and notebooks, supports a robust implementation that is ready for deployment. By integrating these advanced models into our recommendation engine, we aim to deliver a more personalized and engaging customer experience, ultimately driving higher conversions and customer satisfaction.

This proposal covers the entire workflow, from data preprocessing to model deployment, using the files and scripts you provided. Let me know if you need further adjustments or additions!